

TESIS

PERAMALAN IHSG DENGAN METODE ARIMA-GARCH DAN LSTM PADA PERIODE SEBELUM, MASA DAN SESUDAH COVID-19

untuk memenuhi sebagian persyaratan mencapai derajat Pascasarjana S-2

Program Studi Statistika Program Magister



diajukan oleh

Yoel Christopher Tjen

23928001

Kepada

**PROGRAM STUDI STATISTIKA PROGRAM MAGISTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
FEBUARI 2025**

**PERAMALAN IHSG DENGAN METODE ARIMA-GARCH DAN
LSTM PADA PERIODE SEBELUM, MASA DAN SESUDAH COVID-19**

TESIS

diajukan oleh:

Yoel Christopher Tjen

23928001

Telah dipertahankan di hadapan Panitia Penguji Tesis
Program Studi Statistika Program Magister Jurusan Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia

Tanggal : 16 Mei 2025

Dewan Penguji

1. Priyonggo Suseno S.E., M.Sc. Ph.D
2. Dr. Edy Widodo, S.Si., M.Si.
3. Dr. Raden Bagus Fajriya Hakim, S.Si., M.Si.
4. Prof. Drs. Agus Widarjono, MA., Ph.D.
5. Dr. techn. Rohmatul Fajriyah
6. Dr. Atina Ahdika, S.Si., M.Si.

Tanda Tangan



Mengetahui,
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia



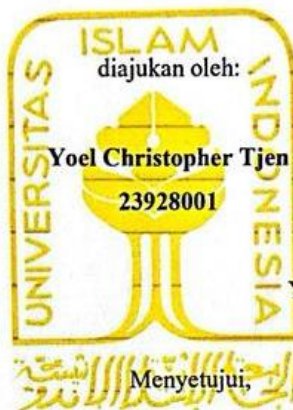
Prof. Riyanto, S.Pd., M.Si., Ph.D..



LEMBAR PENGESAHAN PEMBIMBING

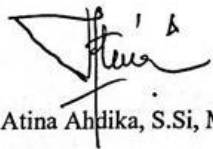
TESIS

**Peramalan IHSG dengan metode ARIMA-GARCH dan LSTM
pada periode sebelum, masa dan sesudah Covid-19**



Yogyakarta, 21 Febuari 2025

Dosen Pembimbing I


Dr. Atina Ahdika, S.Si, M.Si.

Dosen Pembimbing II


Dr.techn. Rohmatul Fajriyah, S.Si,M.Si.

Mengetahui,
Ketua Program Magister


Dr.techn. Rohmatul Fajriyah, S.Si, M.Si.

PERNYATAAN BEBAS PLAGIASI

Saya yang bertandatangan di bawah ini :

Nama : Yoel Christopher Tjen
NIM : 23928001
Tahun Terdaftar : 2023
Program Studi : Statistika Program Magister
Fakultas : Matematika dan Ilmu Pengetahuan Alam

Menyatakan bahwa dalam dokumen ilmiah Tesis ini tidak terdapat bagian karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara lengkap tertulis dalam daftar pustaka. Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Tesis ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 22 Febuari 2025



Yoel Christopher Tjen

23928001

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tesis ini yang berjudul " PERAMALAN IHSG DENGAN METODE ARIMA-GARCH DAN LSTM PADA PERIODE SEBELUM, MASA DAN SESUDAH COVID-19". Tesis ini diajukan sebagai salah satu syarat untuk memperoleh gelar M.Stat di Program Studi Statistika Program Magister, Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.

Penulisan tesis ini tidak lepas dari dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D selaku Rektor Universitas Islam Indonesia.
2. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia
3. Bapak Dr. Edy Widodo, S.Si., M.Si. selaku Ketua Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia.
4. Ibu Dr.techn. Rohmatul Fajriyah, S.Si, M.Si., selaku Ketua Program Studi Statistika Program Magister dan Dosen Pembimbing Kedua yang telah memberikan bimbingan, arahan, dan motivasi selama penyusunan Tugas Akhir.
5. Ibu Dr. Atina Ahdika, S.Si., M.Si. selaku Dosen Pembimbing Pertama yang telah memberikan bimbingan, arahan, dan motivasi selama penyusunan Tugas Akhir.
6. Seluruh staff pengajar Program Studi Statistika Program Magister Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia yang telah memberikan ilmu ilmu yang bermanfaat bagi penulis
7. Seluruh staf akademik di Program Studi Statistika Program Magister, yang telah memberikan dukungan selama masa studi penulis.

8. Imtikhanah Anis Mahmudiati, Alumni Magister Statistika Universitas Islam Indonesia, yang membantu dalam proses pembuatan sintaks R untuk penyusunan Tugas Akhir.
9. Kedua Orang tua penulis, Bapak Yohanes dan Ibu Tjhin Cui Dyun serta seluruh keluarga besar yang selalu memberikan kasih sayang, dukungan, motivasi, dan doa untuk kelancaran segala urusan penulis

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun guna menyempurnakan karya ini di masa mendatang. Akhir kata, penulis berharap semoga thesis ini dapat memberikan manfaat bagi semua pihak yang membutuhkan.

Yogyakarta, 22 Febuari 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI.....	4
DAFTAR TABEL.....	7
DAFTAR GAMBAR	8
DAFTAR ISTILAH/SINGKATAN.....	10
DAFTAR LAMPIRAN.....	11
BAB 1 PENDAHULUAN	14
1.1 Latar Belakang.....	14
1.2 Rumusan Masalah	20
1.3 Tujuan Penelitian	21
1.4 Manfaat Penelitian	22
1.5 Batasan Masalah	23
1.6 Kebaharuan Penelitian	25
1.7 Sistematika Penulisan	25
1.8 Kaitan permasalahan yang diteliti dengan Al-Qur'an	27
BAB 2 TINJAUAN PUSTAKA	28
2.1 Tinjauan Hasil Penelitian.....	28
2.1.1 Penelitian Terdahulu	28
2.2 Konsep Dasar	33
2.2.1 Pasar Modal	33
2.2.2 Saham.....	35
2.2.3 Indeks Saham	38
2.2.4 Indeks Harga Saham Gabungan.....	40
2.2.5 Covid-19	42
2.2.6 Deret Waktu (<i>Time Series</i>).....	44
2.2.7 <i>Forecasting</i>	46
2.2.8 <i>Autocorrelation Function</i> (ACF)	47
2.2.9 Stasioneritas	49
2.3 Landasan Teori ARIMA-GARCH.....	53

2.3.1 Model ARIMA.....	53
2.3.2 Model ARCH-GARCH.....	57
2.4 Landasan Teori LSTM.....	61
2.4.1 <i>Artificial Neural Network</i> (ANN)	61
2.4.2 <i>Recurrent Neural Network</i> (RNN).....	64
2.4.3 <i>Long Short-Term Memory</i> (LSTM)	65
2.4.4 <i>Optimizer Adaptive Moment Estimation Optimization</i> (Adam).....	72
2.4.5 Normalisasi	74
2.4.6 Denormalisasi	75
2.5 Kriteria Pemilihan Model Terbaik	75
2.5.1 <i>Mean Squared Error</i> (MSE)	76
2.5.2 <i>Mean Absolute Percentage Error</i> (MAPE)	77
2.5.3 <i>Akaike Information Criterion</i> (AIC)	78
2.5.4 <i>Root Mean Square Error</i> (RMSE).....	79
BAB 3 METODOLOGI PENELITIAN	80
3.1 Pendekatan Penelitian	80
3.2 Jenis dan Sumber Data.....	80
3.3 Variabel Penelitian.....	81
3.4 Tahapan Pengelolaan Data untuk Metode ARIMA-GARCH.....	81
3.5 Tahapan Pengolahan Data untuk Metode LSTM.....	85
BAB 4 HASIL DAN PEMBAHASAN	93
4.1 Analisis Deskriptif Data IHSG	93
4.1.1 Data IHSG Sebelum Covid-19.....	95
4.1.2 Data IHSG pada Masa Covid-19	98
4.1.3 Data IHSG Setelah Covid-19.....	102
4.2 Hasil Pengujian dan Pembahasan ARIMA-GARCH.....	106
4.2.1 Pembagian Data Training dan Testing.....	106
4.2.2 Uji Stasioneritas Data	107
4.2.3 Transformasi Data dengan Differencing.....	116
4.2.4 Penentuan Nilai p dan q Berdasarkan ACF, PACF, dan Uji ARIMA	124
4.2.5 Pemilihan Model ARIMA Terbaik	129

4.2.6 Uji Normalitas Residual.....	137
4.2.7 Uji Autokorelasi Residual.....	138
4.2.8 Uji Rerata Residual Sama dengan Nol.....	139
4.2.9 Uji Homoskedastisitas Residual dengan White Test	140
4.2.10 Uji Efek ARCH pada Residual dan Pemilihan Model GARCH.....	141
4.2.11 Pemilihan Model GARCH Terbaik	142
4.2.12 Peramalan Model ARIMA-GARCH.....	148
4.3 Hasil Pengujian dan Pembahasan LSTM.....	157
4.3.1 Normalisasi Data.....	157
4.3.2 Pembagian Data Training dan Testing.....	161
4.3.3 Penentuan Jumlah <i>Neuron</i> dan <i>Epoch</i> LSTM.....	162
4.3.4 Hasil Prediksi IHSG Menggunakan Metode LSTM.....	166
4.4 Perbandingan Model ARIMA-GARCH dengan Model LSTM.....	181
4.4.1 Perbandingan Pada Periode Sebelum COVID-19.....	182
4.4.2 Perbandingan Pada Periode Selama COVID-19.....	182
4.4.3 Periode Setelah COVID-19.....	183
BAB 5 KESIMPULAN DAN SARAN	185
5.1 Kesimpulan	185
5.2 Saran	187
DAFTAR PUSTAKA	190

DAFTAR TABEL

Tabel 2.1. Penelitian-penelitian Terdahulu	28
Tabel 4.1. Analisis Data Deskriptif IHSG Sebelum Covid-19	96
Tabel 4.2. Analisis Data Deskriptif IHSG pada masa Covid-19.....	100
Tabel 4.3. Analisis Data Deskriptif IHSG setelah Covid-19	104
Tabel 4.4. Hasil Pengujian ADF test pada Data Sebelum Covid-19.....	109
Tabel 4.5. Hasil Pengujian ADF test pada Data pada Masa Covid-19	111
Tabel 4.6. Hasil Pengujian ADF test pada Data Setelah Covid-19.....	114
Tabel 4.7. Hasil Pengujian ADF test pada Data Sebelum Covid-19 Setelah Differencing	117
Tabel 4.8. Hasil Pengujian ADF test pada Data pada Masa Covid-19 Setelah Differencing	120
Tabel 4.9. Hasil Pengujian ADF test pada Data Setelah Covid-19 Setelah Differencing.....	122
Tabel 4.10. Ringkasan Hasil Pengujian ARIMA sebelum Covid-19.....	129
Tabel 4.11. Ringkasan Hasil Pengujian ARIMA pada Masa Covid-19.....	131
Tabel 4.12. Ringkasan Hasil Pengujian ARIMA Setelah Covid-19	135
Tabel 4.13. Ringkasan Hasil Pengujian GARCH Sebelum Covid-19	144
Tabel 4.14. Ringkasan Hasil Pengujian GARCH pada Masa Covid-19	145
Tabel 4.15. Ringkasan Hasil Pengujian GARCH Setelah Covid-19	146
Tabel 4.16. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19	149
Tabel 4.17. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19	152
Tabel 4.18. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19	154
Tabel 4.19. Normalisasi Data IHSG sebelum Covid-19	158
Tabel 4.20. Normalisasi Data IHSG pada Covid-19.....	159
Tabel 4.21. Normalisasi Data IHSG setelah Covid-19	160
Tabel 4.22. Hasil Percobaan <i>Neuron</i> dan <i>Epoch</i> sebelum Covid-19	163
Tabel 4.23. Hasil Percobaan <i>Neuron</i> dan <i>Epoch</i> pada masa Covid-19.....	164
Tabel 4.24. Hasil Percobaan <i>Neuron</i> dan <i>Epoch</i> setelah Covid-19	165
Tabel 4.25. Perhitungan Prediksi IHSG Sebelum Pandemi Covid-19.....	171
Tabel 4.26. Perhitungan Prediksi IHSG Pada Masa Pandemi Covid-19	175
Tabel 4.27. Perhitungan Prediksi IHSG Setelah Pandemi Covid-19	179
Tabel 4.28. Perbandingan Akurasi Model ARIMA-GARCH dan LSTM.....	181

DAFTAR GAMBAR

Gambar 2.1. Jenis-jenis Pola Data Times Series.....	50
Gambar 2.2. Struktur <i>neuron</i> Jaringan Syaraf Tiruan, Widagda (2016).....	61
Gambar 2.3. Arsitektur Artificial neural network (ANN), Bre, Gimenez dan Fachinotti (2018).....	63
Gambar 2.4. Recurrent Neural Network (RNN), Olah (2016).....	65
Gambar 2.5. Perulangan pada RNN dengan satu layer, Olah (2016).....	66
Gambar 2.6. Perulangan pada LSTM, Olah (2016)	67
Gambar 2.7. <i>Cell State</i> pada LSTM, Olah (2016).....	67
Gambar 2.8. Alur <i>Forget Gate</i> pada LSTM, Olah (2016)	68
Gambar 2.9. Alur input gate pada LSTM, Olah (2016)	69
Gambar 2.10. Alur Pembaruan <i>Cell State</i> pada LSTM, Olah (2016)	70
Gambar 2.11. Alur output pada LSTM, Olah (2016).....	71
Gambar 3.1. <i>Flowchart</i> Analisis Data.....	84
Gambar 3.2. Diagram Alur metode <i>forecasting</i> LSTM	85
Gambar 3.3. Alur preprocessing.....	86
Gambar 3.4. Alur pelatihan prediksi harga saham menggunakan LSTM.....	88
Gambar 3.5. Alur pengujian prediksi harga saham menggunakan LSTM.....	89
Gambar 4.1. Pergerakan IHSG dari 1 Januari 2018 sampai 5 September 2024. Sumber: Data diolah.....	93
Gambar 4.2. Pergerakan IHSG Sebelum Covid.....	97
Gambar 4.3. Pergerakan IHSG pada masa Covid. Sumber: Data diolah	101
Gambar 4.4. Pergerakan IHSG setelah Covid. Sumber: Data diolah.....	105
Gambar 4.5. ACF dan PACF pada Data Sebelum Covid-19 sebelum Differencing.....	108
Gambar 4.6. ACF dan PACF pada Data pada Masa Covid-19 sebelum Differencing	111
Gambar 4.7. ACF dan PACF pada Data Setelah Covid-19 sebelum Differencing.....	114
Gambar 4.8. ACF dan PACF Sebelum Covid-19 Setelah Differencing	117
Gambar 4.9. ACF dan PACF pada Masa Covid-19 Setelah Differencing.....	120
Gambar 4.10. ACF dan PACF Setelah Covid-19 Setelah Differencing	122
Gambar 4.11. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19	149
Gambar 4.12. Perbandingan Data Aktual dan <i>Forecast</i> pada Periode Masa Covid-19	151
Gambar 4.13. Perbandingan Data Aktual dan <i>Forecast</i> pada Periode Setelah Covid-19.....	154
Gambar 4.14. Prediksi IHSG Menggunakan LSTM pada Periode Sebelum Pandemi COVID-19	167
Gambar 4.15. Analisis Perbandingan Prediksi dan Aktual IHSG Sebelum Pandemi COVID-19	169
Gambar 4.16. Prediksi IHSG Menggunakan LSTM pada Periode pada Masa Pandemi COVID-19.....	172
Gambar 4.17. Analisis Perbandingan Prediksi dan Aktual IHSG pada Masa Pandemi COVID-19	173

Gambar 4.18. Prediksi IHSG Menggunakan LSTM pada Periode Setelah Pandemi COVID-19	176
Gambar 4.19. Analisis Perbandingan Prediksi dan Aktual IHSG Setelah Pandemi COVID-19	178

DAFTAR ISTILAH/SINGKATAN

IHSG	: Indeks Harga Saham Gabungan
AR	: Autoregressive
MA	: Moving Average
ARMA	: Autoregressive Moving Average
ARIMA	: Autoregressive Integrated Moving Average
ARCH	: Autoregressive Conditional Heteroskedasticity
GARCH	: Generalized Autoregressive Conditional Heteroskedasticity
ANN	: Artificial Neural Network
RNN	: Recurrent Neural Network
LSTM	: Long Short-Term Memory
MSE	: Mean Squared Error
RMSE	: Root Mean Squared Error
MAPE	: Mean Absolute Percentage Error
SSE	: Sum of Squared Errors
AIC	: Akaike Information Criterion
BIC	: Bayesian Information Criterion
Covid-19	: Coronavirus Disease 2019

DAFTAR LAMPIRAN

Lampiran 1. Data Penelitian (Sebelum, Masa dan Seteleh).....	205
Lampiran 2. Sintaks R Model ARIMA GARCH Sebelum Covid-19.....	206
Lampiran 3. Sintaks R Model ARIMA GARCH Masa Covid-19	231
Lampiran 4. Sintaks R Model ARIMA GARCH SetelahCovid-19.....	261
Lampiran 5. Sintaks Python Model LSTM Sebelum Covid-19	284
Lampiran 6. Sintaks Python Model LSTM Masa Covid-19	297
Lampiran 7. Sintaks Python Model LSTM SetelahCovid-19	308

PERAMALAN IHSG DENGAN METODE ARIMA-GARCH DAN LSTM PADA PERIODE SEBELUM, MASA DAN SESUDAH COVID-19

Yoel Christopher Tjen

Dibawah Bimbingan Dr. Atina Ahdika, S.Si., M.Si dan Dr.techn. Rohmatul Fajriyah, S.Si, M.Si.

ABSTRAK

Pasar saham merupakan indikator utama yang mencerminkan kondisi ekonomi suatu negara dan sering kali dipengaruhi oleh berbagai faktor makroekonomi serta sentimen pasar. Oleh karena itu, metode peramalan yang akurat sangat diperlukan untuk membantu investor, regulator, dan pemangku kepentingan dalam mengambil keputusan yang lebih tepat. Penelitian ini bertujuan untuk membandingkan performa model ARIMA-GARCH dan LSTM dalam memprediksi pergerakan Indeks Harga Saham Gabungan (IHSG) pada tiga periode yang berbeda, yaitu sebelum, selama, dan setelah pandemi COVID-19. Hasil penelitian menunjukkan bahwa model ARIMA-GARCH lebih unggul dalam hal akurasi dibandingkan dengan LSTM, yang terlihat dari nilai MSE, RMSE, dan MAPE yang lebih rendah di semua periode. Model ARIMA-GARCH terbukti lebih efektif dalam menangkap pola volatilitas pasar, baik dalam kondisi stabil sebelum pandemi maupun saat terjadi gejolak selama dan setelah pandemi. Di sisi lain, meskipun model LSTM memiliki keunggulan dalam menangkap pola non-linear yang kompleks, model ini mengalami kesulitan dalam mengakomodasi perubahan pola data yang drastis akibat ketidakpastian ekonomi selama pandemi. Dengan demikian, hasil penelitian ini memberikan pengetahuan yang berharga bagi investor, akademisi, dan regulator dalam memilih model peramalan yang paling sesuai untuk memprediksi pergerakan pasar saham serta dalam merumuskan strategi investasi dan manajemen risiko yang lebih optimal.

Kata kunci: Peramalan, ARIMA-GARCH, LSTM, IHSG, COVID-19.

ABSTRACT

The stock market serves as a key indicator of a country's economic conditions and is often influenced by various macroeconomic factors and market sentiment. Therefore, accurate forecasting methods are essential to assist investors, regulators, and stakeholders in making informed decisions. This study aims to compare the performance of the ARIMA-GARCH and LSTM models in predicting the movement of the Indonesia Stock Exchange Composite Index (IHSG) across three different periods: before, during, and after the COVID-19 pandemic. The results indicate that the ARIMA-GARCH model outperforms LSTM in terms of accuracy, as evidenced by its lower MSE, RMSE, and MAPE across all periods. The ARIMA-GARCH model is more effective in capturing market volatility patterns, whether under stable conditions before the pandemic or during the market disruptions caused by the pandemic and its aftermath. On the other hand,

although the LSTM model excels at capturing complex non-linear patterns, it struggles to adapt to abrupt changes in data patterns caused by economic uncertainty during the pandemic. Thus, the findings of this study provide valuable insights for investors, academics, and regulators in selecting the most suitable forecasting model for predicting stock market movements, as well as in formulating more effective investment strategies and risk management practices.

Keywords: *Forecasting, ARIMA-GARCH, LSTM, IHSG, COVID-19.*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun kebelakang ekonomi Indonesia mengalami perubahan yang signifikan dimana kenaikan penurunannya cukup signifikan tetapi ini tidak hanya terjadi di Indonesia namun seluruh dunia dan negara tetangga kita seperti Malaysia, Singapore dan Thailand juga mengalaminya. Menurut data dari website bank dunia sebelum tahun 2020 pertumbuhan ekonomi Indonesia cenderung stabil di angka 5-6%, dan hal tersebut mengungguli pertumbuhan ekonomi negara tetangga seperti Thailand yang hanya bertumbuh sebesar 1% sampai 4.2%, Malaysia yang bertumbuh 4.4% sampai 6 % dan Singapore bertumbuh sebesar 1.3% sampai 4.8% serta perekonomian dunia yang bertumbuh sebesar 2.6% sampai 3.4%.

Perubahan tersebut menghasilkan ketidakpastian dan membuat dampak ke seluruh dunia. Menurut Jurado et al (2015), perubahan merupakan suatu tantangan tetapi juga bisa menjadi peluang untuk menciptakan perkembangan nilai tambah serta keunggulan kompetitif. Ketidakpastian akan sangat berdampak pada strategi dan analisis yang dilakukan untuk menghasilkan output yang berkualitas dikarenakan perubahan yang terjadi.

Perubahan ini dimulai pada tahun 2020 dimana terjadi pandemi Covid-19 yang sangat mematikan dan dapat menular sangat cepat. Hal itu menimbulkan terjadinya berbagai kebijakan pembatasan perekonomian yang dibuat oleh pemerintah di seluruh dunia untuk mengurangi dan mengatasi dampak penyebaran virus Covid-19 yang tentunya akan berdampak secara signifikan terhadap perekonomian, hal ini dapat terlihat pada pertumbuhan ekonomi dunia yang negatif sebesar 3.1% di tahun 2020, namun penurunan pertumbuhan ekonomi di Indonesia masih dibawah penurunan perekonomian di dunia dikarenakan penurunan perekonomian di Indonesia hanya sebesar 2.1% dan jika dibandingkan dengan negara tetangga penurunannya jauh lebih tinggi seperti Malaysia mengalami penurunan perekonomian sebesar 5.5%, Thailand mengalami penurunan sebesar 6.1% dan Singapore mengalami penurunan 3.9%.

Menurut Sharma (2020), pandemi COVID-19 membuat kehidupan manusia tidak lagi sama seperti sebelumnya. Ekonomi dan bisnis banyak yang lesu, bahkan mengalami kebangkrutan akibat

perubahan yang disebabkan oleh pandemi ini, sehingga diperlukan pendekatan yang berbeda dalam menghadapi krisis serupa. Adaptasi strategi dalam dunia bisnis, inovasi teknologi, serta kebijakan ekonomi yang responsif menjadi kunci dalam menjaga keberlanjutan di tengah ketidakpastian global.

Menurut Fatmawati dan Hendrayani (2020) Covid-19 bermula dari Wuhan di China pada akhir 2019 dan pada awal 2020 yang menyebar hampir ke seluruh dunia yaitu 213 negara, pada akhir April 2020 saja sudah mencatatkan lebih dari 3 juta orang terpapar virus tersebut sedangkan di Indonesia hampir 10 ribu orang terpapar virus tersebut dengan Tingkat kematian 784 orang. Pambudi (2023) berpendapat Pemerintah melakukan beberapa kebijakan untuk mengatasi Pandemi Covid-19 khususnya untuk membatasi interaksi antara manusia untuk mencegah penyebaran lebih lanjut, seperti pembatasan sosial berskala besar (PSBB) dan *Sosial Distancing*, tetapi kebijakan tersebut terdapat konsekuensi yaitu memperlambat dan menghambat perekonomian.

Masa pemulihan perekonomian dunia dan Indonesia dimulai dari tahun 2021 dan 2022, Dimana pertumbuhan ekonomi yang negatif di tahun 2020 mulai berbalik positif di tahun 2021, perekonomian dunia pada tahun 2021 bertumbuh 6.2% dan tahun 2022 bertumbuh 3.1%, sedangkan di Indonesia sendiri pertumbuhan perekonomian meningkat dari tahun ke tahun. Pada tahun 2021 perekonomian Indonesia bertumbuh sebesar 3.7% dan tahun 2022 bertumbuh sebesar 5.3%. Perekonomian di negara tetangga Indonesia juga mengalami pertumbuhan seperti Malaysia mengalami pertumbuhan sebesar 3.7% pada tahun 2021 dan 8.7% pada tahun 2022, Thailand mengalami pertumbuhan sebesar 1,5% pada tahun 2021 dan 2.6% di tahun 2022 sedangkan Singapore mengalami pertumbuhan sebesar 8.9% pada tahun 2021 dan 3.6% pada tahun 2022.

Dampak dari pandemi ini juga menghampiri pasar saham di dunia khususnya di Indonesia yang menyebabkan berbagai indeks dan saham di Indonesia mengalami penurunan yang signifikan. Pasar Modal di Indonesia sudah dimulai sejak zaman kolonial Belanda. Bursa efek pertama kali didirikan di Batavia (sekarang Jakarta) pada tahun 1912 oleh pemerintah Hindia Belanda untuk memperdagangkan saham dan obligasi perusahaan-perusahaan kolonial. Namun, aktivitas bursa sempat terhenti beberapa kali, terutama selama Perang Dunia II dan masa awal kemerdekaan Indonesia. Pada tahun 1977, pemerintah Indonesia menghidupkan kembali pasar modal dengan

mengaktifkan Bursa Efek Jakarta (BEJ) di bawah pengawasan Bapepam (Badan Pengawas Pasar Modal), sekaligus memperkenalkan berbagai insentif seperti deregulasi dan insentif pajak untuk menarik minat investor domestik (Rorizki et al., 2022).

Seiring berkembangnya ekonomi Indonesia, pasar modal mengalami transformasi besar. Pada tahun 1995, Bursa Efek Surabaya (BES) didirikan untuk memperdagangkan obligasi dan derivatif, tetapi akhirnya digabung dengan BEJ pada tahun 2007, membentuk Bursa Efek Indonesia (BEI) yang menjadi satu-satunya bursa saham di Indonesia. Regulasi dan infrastruktur terus diperkuat, termasuk penerapan sistem perdagangan elektronik dan kebijakan untuk meningkatkan transparansi serta perlindungan investor. Hingga kini, pasar modal Indonesia terus berkembang dengan meningkatnya jumlah emiten, investor, dan instrumen keuangan yang diperdagangkan, berperan penting dalam pembiayaan sektor riil dan pertumbuhan ekonomi nasional (Muklis, 2016).

Indeks Harga Saham Gabungan (IHSG) adalah indeks utama yang mencerminkan kinerja seluruh saham yang tercatat di Bursa Efek Indonesia (BEI). Diluncurkan pada 1 April 1983 dengan nilai dasar 100, IHSG digunakan sebagai indikator pergerakan pasar modal Indonesia secara keseluruhan. Pergerakan IHSG dipengaruhi oleh berbagai faktor, termasuk kondisi makroekonomi, kebijakan moneter, arus modal asing, serta sentimen global. Sebagai barometer pasar, IHSG sering dijadikan acuan oleh investor untuk menilai tren investasi di Indonesia, dengan sektor-sektor utama seperti perbankan, infrastruktur, dan komoditas memiliki pengaruh besar terhadap fluktuasi indeks (Fatmawati, Badriah dan Gunawan, 2024).

Junaedi dan Salistia (2020) menemukan Pandemi COVID-19 menyebabkan pasar modal di Indonesia, khususnya pada pergerakan Indeks Harga Saham Gabungan (IHSG) mengalami penurunan drastis akibat ketidakpastian investor dan kebijakan pembatasan sosial seperti PSBB dan WFH. Sebelum kasus pertama dikonfirmasi di Indonesia pada Maret 2020, IHSG sudah menunjukkan tren penurunan sebagai respons terhadap lonjakan kasus global. Faktor eksternal, seperti dinamika pasar saham di AS, China, dan Eropa, juga mempengaruhi volatilitas pasar Indonesia, mengingat keterkaitan pasar modal secara global. Penelitian ini menunjukkan bahwa kebijakan social distancing memberikan dampak negatif terhadap IHSG, sementara investor cenderung mengalihkan dana ke aset yang lebih aman. Meskipun IHSG mulai menunjukkan pemulihan setelah kebijakan stimulus ekonomi diterapkan, volatilitas masih tinggi dan belum

mencapai level sebelum pandemi, mencerminkan kompleksitas dampak COVID-19 terhadap pasar modal Indonesia

Anggraini (2021) berpendapat bahwa pandemi COVID-19 menyebabkan pasar saham di Indonesia mengalami penurunan yang signifikan. Hal ini terutama disebabkan oleh meningkatnya ketidakpastian di dunia usaha akibat lonjakan jumlah kasus COVID-19 yang terus bertambah. Ketidakpastian tersebut membuat para investor menjadi lebih berhati-hati dan cenderung menarik dana dari pasar saham, sehingga menimbulkan tekanan jual yang tinggi dan berdampak pada pelemahan indeks harga saham. Selain itu, penyebaran virus yang semakin luas mendorong pemerintah untuk memberlakukan berbagai kebijakan pembatasan aktivitas, seperti Pembatasan Sosial Berskala Besar (PSBB) dan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM). Kebijakan tersebut secara langsung menghambat berbagai sektor ekonomi, terutama sektor yang bergantung pada mobilitas dan aktivitas fisik seperti pariwisata, transportasi, dan ritel. Dampak negatif terhadap perekonomian ini semakin memperburuk sentimen pasar dan menyebabkan volatilitas tinggi dalam perdagangan saham. Dengan adanya pembatasan aktivitas dan ketidakpastian yang berkepanjangan, banyak perusahaan mengalami penurunan pendapatan dan keuntungan, yang pada akhirnya turut memperburuk kondisi pasar saham di Indonesia.

Sementara itu, penelitian oleh Caiado dan Lucio (2023) menunjukkan bahwa COVID-19 berdampak negatif pada sebagian besar sektor pasar saham di Amerika Serikat. Sektor pariwisata, perhotelan, transportasi, dan energi mengalami penurunan tajam akibat berkurangnya permintaan serta gangguan dalam rantai pasokan global. Penutupan bisnis dan ketidakpastian kebijakan ekonomi semakin memperburuk kondisi sektor-sektor tersebut, mendorong investor untuk lebih berhati-hati dalam menanamkan modalnya. Namun, tidak semua industri mengalami tekanan yang sama. Beberapa sektor justru mengalami pertumbuhan signifikan, seperti farmasi, teknologi internet, dan hiburan digital. Peningkatan permintaan obat-obatan, vaksin, serta alat kesehatan mendorong sektor farmasi, sementara pergeseran aktivitas masyarakat ke platform digital mempercepat pertumbuhan e-commerce, video streaming, dan kerja jarak jauh. Dengan adanya perubahan pola konsumsi selama pandemi, industri hiburan digital juga mengalami lonjakan popularitas. Temuan ini menunjukkan bahwa meskipun pandemi memberikan dampak negatif luas, beberapa sektor tetap berkembang karena adaptasi terhadap perubahan perilaku masyarakat dan kebutuhan pasar yang baru.

Di pasar modal, memprediksi harga aset keuangan di masa depan merupakan hal yang sangat sulit dilakukan karena adanya banyak faktor yang dapat mempengaruhi pergerakan pasar, baik yang bersifat ekonomi, politik, sosial, maupun lingkungan. Ketidakpastian yang tinggi dan volatilitas yang terjadi dalam jangka pendek menjadi tantangan utama bagi para investor dan analis dalam meramalkan tren pasar. Salah satu teknik yang sering digunakan untuk mencoba mengatasi ketidakpastian ini adalah *forecasting*, yang bertujuan untuk memberikan proyeksi harga atau nilai suatu aset berdasarkan data historis yang ada. Meskipun metode ini dapat memberikan gambaran mengenai kemungkinan pergerakan harga di masa depan, hasil dari *forecasting* sangat bergantung pada kualitas dan keakuratan input data yang digunakan serta kondisi pasar yang sedang berlangsung.

Penelitian ini menggunakan Indeks Harga Saham Gabungan (IHSG) sebagai objek penelitian dalam forecasting didasarkan pada karakteristiknya yang lebih komprehensif dan representatif dibandingkan dengan saham individu. IHSG mencerminkan kinerja keseluruhan pasar modal Indonesia, sehingga hasil peramalan dapat memberikan gambaran yang lebih luas mengenai tren ekonomi dan sentimen investor secara makro. Selain itu, IHSG memiliki volatilitas yang lebih terdiversifikasi, karena terdiri dari berbagai saham dari berbagai sektor, sehingga dapat mengurangi risiko bias yang mungkin terjadi jika hanya menggunakan satu saham tertentu yang kinerjanya dipengaruhi oleh faktor spesifik perusahaan. Penggunaan indeks dalam forecasting juga lebih relevan untuk kebijakan ekonomi dan pengambilan keputusan investasi, karena banyak investor institusional dan regulator menggunakan indeks sebagai acuan dalam strategi investasi dan analisis risiko pasar.

Penelitian ini akan menggunakan tiga periode data yang mencerminkan tahapan berbeda dalam perkembangan pandemi COVID-19 di Indonesia. Pembagian periode ini didasarkan pada peristiwa penting yang secara resmi diakui oleh pemerintah dan memiliki dampak signifikan terhadap kondisi ekonomi, sosial, serta kebijakan di Indonesia.

Pertama, periode sebelum COVID-19 akan mencakup data dari 1 Januari 2018 hingga 1 Maret 2020. Pemilihan batas akhir ini merujuk pada pengumuman resmi pertama kasus COVID-19 di Indonesia oleh Presiden Joko Widodo pada 2 Maret 2020 (Kompas, 2020) yang tertuang dalam Keputusan Presiden no 11 tahun 2020 tentang penetapan kedaruratan Covid-19 di Indonesia . Data

pada periode ini akan merepresentasikan kondisi normal sebelum adanya gangguan akibat pandemi.

Kedua, periode masa COVID-19 akan meliputi data dari 2 Maret 2020 hingga 30 Desember 2022. Pemilihan tanggal akhir ini didasarkan pada pencabutan kebijakan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) oleh pemerintah pada 30 Desember 2022, yang menandakan transisi dari kebijakan pembatasan pandemi menuju kondisi yang lebih fleksibel yang diberitakan oleh kantor kepresidenan pada akhir 2022 (Presiden RI, 2022). Periode ini akan mencerminkan dampak penuh pandemi terhadap berbagai indikator ekonomi dan sosial.

Ketiga, periode setelah COVID-19 akan menggunakan data dari 1 Januari 2023 hingga 5 September 2024. Pemilihan periode ini merujuk pada masa setelah pencabutan PPKM dan pengesahan Keputusan Presiden Nomor 17 Tahun 2023 pada 21 Juni 2023, yang secara resmi menyatakan bahwa status pandemi COVID-19 di Indonesia telah berakhir dan diubah menjadi endemi (BPK RI, 2023). Data pada periode ini akan digunakan untuk memahami pola pemulihan pasca-pandemi dan proyeksi tren ke depan.

Alasan utama penelitian ini menggunakan tiga periode tersebut adalah untuk menguji efektivitas teknik *forecasting* dalam berbagai kondisi ekonomi dan sosial. Setiap periode memiliki karakteristik unik yang dapat memengaruhi hasil prediksi, sehingga memungkinkan analisis mendalam mengenai keandalan dan ketepatan metode forecasting dalam situasi yang berbeda (Nabi, 2020).

Periode sebelum COVID-19 merepresentasikan keadaan ekonomi yang stabil tanpa adanya gangguan besar, sehingga dapat digunakan sebagai dasar untuk memahami pola historis sebelum adanya shock eksternal (Ivashka, Arsana dan Dewi, 2024). Selanjutnya Periode selama COVID-19 mencerminkan kondisi ekonomi dengan volatilitas tinggi akibat krisis global, kebijakan pembatasan, serta perubahan perilaku pasar yang ekstrem (Arianto, 2020). Menguji forecasting dalam kondisi ini dapat menunjukkan sejauh mana metode yang digunakan mampu beradaptasi terhadap perubahan mendadak.

Periode setelah COVID-19 menggambarkan fase pemulihan dan transisi menuju stabilitas baru, yang memungkinkan analisis bagaimana metode forecasting merespons pemulihan ekonomi serta

tren baru yang muncul pasca-pandemi (Nansi, Natalia dan Rohman, 2024). Dengan pembagian ini, penelitian ini tidak hanya mengevaluasi keakuratan prediksi dalam kondisi normal, tetapi juga dalam situasi ekstrem dan fase pemulihan.

Peneliti ingin melakukan uji dengan Teknik *forecasting* pada 3 periode data pada indeks harga saham gabungan di Indonesia, 3 periode tersebut diambil dari pertama, masa sebelum covid, kedua masa covid dimana pertama kali covid masuk ke Indonesia sampai kebijakan pembatasan dicabut oleh pemerintah, dan ketiga pada masa setelah covid yaitu masa setelah pembatasan oleh pemerintah dihentikan. Penelitian ini bertujuan untuk melihat akurasi dari 3 set data tersebut sehingga berbagai teknik forecasting yang telah diuji akan diketahui teknik mana yang cenderung cocok untuk keadaan normal, krisis atau recovery. Diharapkan bahwa penelitian ini menemukan Teknik forecasting tersebut cocok untuk keadaan sebelum Covid-19, masa Covid-19 dan setelah Covid-19.

1.2 Rumusan Masalah

Dalam penelitian ini, dilakukan analisis terhadap performa model *forecasting* ARIMA-GARCH dan LSTM dalam memprediksi pergerakan IHSG pada tiga periode berbeda, yaitu sebelum, selama, dan setelah pandemi COVID-19. Mengingat pandemi telah memberikan dampak signifikan terhadap volatilitas pasar modal, penting untuk mengevaluasi efektivitas metode peramalan dalam berbagai kondisi pasar. Oleh karena itu, penelitian ini merumuskan beberapa pertanyaan yang menjadi fokus utama analisis serta tujuan yang ingin dicapai, sebagaimana diuraikan berikut:

1. Bagaimana performa dan tingkat akurasi model *forecasting* ARIMA-GARCH, dan LSTM dalam memprediksi pergerakan IHSG pada periode sebelum COVID-19, ketika kondisi pasar relatif stabil dan tidak mengalami guncangan besar akibat faktor eksternal?
2. Bagaimana performa dan tingkat akurasi model ARIMA-GARCH, dan LSTM dalam meramalkan pergerakan IHSG selama pandemi COVID-19, ketika terjadi peningkatan volatilitas pasar akibat ketidakpastian ekonomi dan kebijakan pemerintah?

3. Bagaimana performa dan tingkat akurasi model forecasting ARIMA-GARCH, dan LSTM dalam memprediksi IHSG setelah pandemi COVID-19, ketika pasar mengalami pemulihan dan memasuki fase transisi menuju kondisi ekonomi pasca-pandemi?
4. Apakah terdapat perubahan karakteristik pola data IHSG sebelum, selama, dan setelah COVID-19, dan bagaimana perubahan ini memengaruhi akurasi masing-masing model *forecasting*?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengevaluasi performa model forecasting ARIMA-GARCH dan LSTM dalam memprediksi pergerakan IHSG pada berbagai kondisi pasar, yakni sebelum, selama, dan setelah pandemi COVID-19. Dengan adanya perbedaan karakteristik pasar pada setiap periode, penting untuk memahami sejauh mana model-model tersebut mampu mengakomodasi perubahan pola data dan tingkat volatilitas yang bervariasi. Adapun tujuan penelitian ini dirinci sebagai berikut:

1. Menganalisis dan mengevaluasi performa dan tingkat akurasi model *forecasting* ARIMA-GARCH dan LSTM dalam memprediksi pergerakan IHSG pada periode sebelum COVID-19, ketika kondisi pasar relatif stabil dan tidak mengalami guncangan besar akibat faktor eksternal, serta mengidentifikasi model yang paling sesuai untuk kondisi pasar yang lebih terkendali.
2. Menganalisis dan mengevaluasi performa dan tingkat akurasi model *forecasting* ARIMA-GARCH dan LSTM dalam meramalkan pergerakan IHSG selama pandemi COVID-19, ketika terjadi peningkatan volatilitas pasar akibat ketidakpastian ekonomi dan kebijakan pemerintah, serta mengkaji sejauh mana model-model tersebut mampu menangkap fluktuasi pasar yang ekstrem.
3. Menganalisis dan mengevaluasi performa dan tingkat akurasi model *forecasting* ARIMA-GARCH dan LSTM dalam memprediksi IHSG setelah pandemi COVID-19, ketika pasar mengalami pemulihan dan memasuki fase transisi menuju kondisi ekonomi pasca-pandemi, serta menentukan apakah model yang efektif sebelum dan selama pandemi tetap relevan dalam periode pemulihan.

4. Menganalisis perubahan karakteristik pola data IHSG sebelum, selama, dan setelah COVID-19 serta mengkaji bagaimana perubahan tersebut memengaruhi akurasi masing-masing model *forecasting*, guna memahami adaptabilitas model dalam berbagai kondisi pasar dan meningkatkan ketepatan prediksi untuk masa depan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam aspek akademis, praktis, serta pengembangan teknologi dalam bidang keuangan dan pasar modal. Adapun manfaat yang dapat diperoleh dari penelitian ini dijelaskan sebagai berikut:

1. Manfaat Akademis

- Memberikan kontribusi terhadap pengembangan literatur terkait penerapan model ARIMA-GARCH dan LSTM dalam peramalan indeks harga saham, khususnya dalam konteks pasar modal Indonesia.
- Menyediakan analisis empiris mengenai efektivitas metode *forecasting* dalam kondisi pasar yang berbeda, yaitu sebelum, selama, dan setelah pandemi COVID-19.
- Menghasilkan wawasan ilmiah mengenai perubahan pola pergerakan IHSG akibat pandemi COVID-19, yang dapat dijadikan referensi dalam penelitian lanjutan di bidang ekonomi, keuangan, dan data science.

2. Manfaat Praktis

- Memberikan informasi bagi investor, analis keuangan, dan manajer investasi mengenai keandalan model ARIMA-GARCH dan LSTM dalam kondisi pasar yang berbeda, sehingga dapat digunakan sebagai dasar dalam pengambilan keputusan investasi yang lebih berbasis data.
- Membantu regulator dan otoritas pasar modal dalam memahami dinamika pergerakan pasar saham selama periode krisis dan pemulihan, sehingga dapat digunakan dalam perumusan kebijakan yang lebih adaptif dan responsif terhadap volatilitas pasar.

- Memfasilitasi pemahaman yang lebih baik terkait risiko pasar dan ketidakpastian ekonomi, sehingga pelaku pasar dapat mengantisipasi dampak dari faktor eksternal terhadap pasar saham secara lebih efektif.

3. Manfaat dalam Pengembangan Teknologi dan Inovasi

- Berkontribusi terhadap pengembangan metode *forecasting* berbasis statistik dan data science, khususnya dalam penerapan *machine learning* dan *hybrid modeling* untuk peramalan pasar saham.
- Menyediakan perbandingan empiris antara model ARIMA-GARCH yang berbasis statistik dan model LSTM yang berbasis *deep learning*, sehingga dapat menjadi dasar bagi pengembangan sistem prediksi pasar saham yang lebih adaptif.
- Mendorong inovasi dalam pengembangan algoritma prediktif yang lebih akurat, terutama dalam menangkap pola volatilitas pasar akibat faktor eksternal, seperti krisis ekonomi atau pandemi global.

1.5 Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan yang ditetapkan guna menjaga fokus dan cakupan analisis agar tetap terarah serta sesuai dengan tujuan yang telah dirumuskan. Batasan masalah yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Objek Penelitian

Penelitian ini berfokus pada Indeks Harga Saham Gabungan (IHSG) sebagai representasi pasar modal Indonesia. Indeks saham lainnya, seperti indeks sektoral atau indeks saham global, tidak termasuk dalam ruang lingkup penelitian ini.

2. Periode Pengamatan

Penelitian ini hanya menggunakan data historis IHSG yang dibagi ke dalam tiga periode utama:

- Periode sebelum COVID-19: Mewakili kondisi pasar yang relatif stabil sebelum adanya guncangan akibat pandemi yang dimulai dari 1 Januari 2018 sampai 1 Maret 2022.
- Periode selama COVID-19: Mewakili kondisi pasar saat pandemi berlangsung, yang ditandai dengan peningkatan volatilitas dan ketidakpastian ekonomi yang dimulai dari 2 Maret 2020 sampai 31 Desember 2022.
- Periode setelah COVID-19: Mewakili fase pemulihan pasar setelah pandemi mereda dan aktivitas ekonomi kembali normal yang dimulai dari 1 Januari 2023 sampai 5 September 2024.

Data lain yang tidak tercantum dalam periode pengamatan tidak diteliti dalam penelitian ini.

3. Metode Forecasting

Model prediksi yang digunakan dalam penelitian ini terbatas pada tiga metode utama, yaitu:

- *Autoregressive Integrated Moving Average-Generalized Autoregressive Conditional Heteroskedasticity* (ARIMA-GARCH): Model berbasis statistik yang banyak digunakan dalam peramalan data keuangan dengan volatilitas tinggi.
- *Long Short-Term Memory* (LSTM): Model berbasis deep learning yang mampu menangkap pola non-linear dalam data deret waktu.
- Perbandingan performa model: Analisis dilakukan untuk membandingkan efektivitas ketiga metode dalam meramalkan IHSG pada ketiga periode yang telah ditetapkan.

4. Sumber Data

Data yang digunakan dalam penelitian ini berasal dari sumber sekunder Yahoo Finance yang menyediakan data historis pergerakan IHSG. Data makroekonomi atau faktor eksternal lainnya, seperti sentiment pasar, kebijakan moneter atau geopolitik global, tidak menjadi fokus utama dalam penelitian ini.

5. Indikator Evaluasi Model

Penelitian ini mengevaluasi performa model forecasting berdasarkan indikator akurasi, seperti *Mean Absolute Percentage Error* (MAPE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE). Faktor lain, seperti dampak psikologis investor atau perubahan regulasi pasar modal, tidak dianalisis secara mendalam dalam penelitian ini.

Dengan adanya batasan masalah ini, penelitian diharapkan dapat lebih fokus dalam mengevaluasi efektivitas model forecasting terhadap pergerakan IHSG dalam berbagai kondisi pasar, tanpa menyertakan faktor-faktor eksternal yang dapat memperluas cakupan penelitian di luar ruang lingkup yang telah ditentukan.

1.6 Kebaharuan Penelitian

Novelty penelitian ini terletak pada analisis perbedaan performa model ARIMA-GARCH dan LSTM dalam meramalkan IHSG di tiga periode berbeda: sebelum, selama, dan setelah pandemi Covid-19. Pendekatan ini belum banyak dieksplorasi dalam penelitian sebelumnya, yang umumnya hanya berfokus pada satu metode pada satu periode waktu tertentu.

Selain itu, penelitian ini memberikan perspektif unik dengan mengungkap bagaimana pola volatilitas dan hubungan temporal IHSG berubah akibat pandemi serta bagaimana masing-masing model menyesuaikan diri terhadap perubahan tersebut. Dengan demikian, penelitian ini tidak hanya menguji keakuratan suatu model, tetapi juga menbandingkan model tradisional (ARIMA-GARCH) dengan model deep learning (LSTM).

1.7 Sistematika Penulisan

Penelitian ini disusun dalam lima bab utama yang membahas secara sistematis berbagai aspek terkait forecasting IHSG menggunakan metode ARIMA-GARCH dan LSTM dalam tiga periode berbeda: sebelum, selama, dan setelah pandemi COVID-19. Struktur penelitian ini dirancang untuk memberikan alur pembahasan yang terstruktur dan komprehensif.

BAB I Pendahuluan menjelaskan latar belakang penelitian terkait pentingnya forecasting IHSG dalam tiga periode utama: sebelum, selama, dan setelah COVID-19. Bab ini menyoroti bagaimana

pandemi menyebabkan volatilitas tinggi di pasar modal dan urgensi untuk mengevaluasi metode forecasting yang dapat menangkap dinamika tersebut. Selain itu, bab ini menguraikan perumusan masalah, tujuan penelitian, manfaat yang diharapkan, batasan masalah untuk memperjelas ruang lingkup penelitian, serta sistematika penulisan guna memberikan gambaran alur penelitian secara keseluruhan.

BAB II Tinjauan Pustaka membahas konsep dasar yang menjadi landasan penelitian, meliputi pasar modal, saham, indeks saham, dan peran IHSG sebagai indikator utama dalam pasar keuangan Indonesia. Selain itu, dijelaskan pula dampak pandemi COVID-19 terhadap pasar modal dan perekonomian secara umum. Bab ini juga menguraikan metode forecasting yang digunakan, yaitu ARIMA-GARCH dan LSTM, serta membahas penelitian terdahulu yang relevan. Dengan adanya tinjauan ini, penelitian dapat memperkuat dasar teori dan memberikan perbandingan terhadap studi sebelumnya.

BAB III Metodologi Penelitian menguraikan pendekatan penelitian yang digunakan, termasuk jenis penelitian, sumber data, serta metode analisis yang diterapkan. Bab ini menjelaskan teknik pengumpulan data IHSG dalam tiga periode yang telah ditentukan, serta prosedur penerapan model forecasting ARIMA-GARCH dan LSTM. Selain itu, dibahas pula metode evaluasi performa model berdasarkan indikator statistik seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), dan Root Mean Squared Error (RMSE) untuk menilai akurasi prediksi yang dihasilkan oleh masing-masing model.

BAB IV Hasil Penelitian dan Analisis menyajikan temuan utama dari penelitian ini, mencakup hasil forecasting IHSG dalam tiga periode serta perbandingan performa model ARIMA-GARCH dan LSTM. Analisis dilakukan untuk melihat keandalan masing-masing metode dalam menangkap pola pergerakan IHSG di berbagai kondisi pasar. Selain itu, bab ini juga membahas perubahan karakteristik data serta bagaimana kondisi sebelum, selama, dan setelah COVID-19 memengaruhi efektivitas model forecasting yang digunakan.

BAB V Kesimpulan dan Saran merangkum hasil penelitian dengan menyoroti temuan utama terkait akurasi model forecasting dalam berbagai kondisi pasar. Bab ini juga memberikan implikasi akademis dan praktis terkait penerapan model forecasting dalam pengambilan keputusan investasi. Selain itu, diberikan rekomendasi bagi penelitian selanjutnya untuk mengembangkan metode yang

lebih adaptif terhadap perubahan pasar serta mempertimbangkan faktor-faktor eksternal lain yang dapat memengaruhi prediksi pergerakan IHSG di masa depan.

1.8 Kaitan permasalahan yang diteliti dengan Al-Qur'an

إِنَّ اللَّهَ عِنْدَهُ عِلْمُ السَّاعَةِ وَيُنَزِّلُ الْغَيْثَ وَيَعْلَمُ مَا فِي الْأَرْحَامِ وَمَا تَدْرِي نَفْسٌ مَّاذَا تَكْسِبُ غَدًا وَمَا تَدْرِي
نَفْسٌ بِأَيِّ أَرْضٍ تَمُوتُ ۚ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

QS. Luqman: 34

“Dan tiada seorang pun yang dapat mengetahui (dengan pasti) apa yang akan diusahakannya besok.”

Relevansi QS. Luqman: 34 dalam peramalan saham terletak pada keterbatasan manusia dalam mengetahui masa depan secara pasti, termasuk dalam memprediksi pergerakan harga saham. Meskipun metode seperti LSTM dan ARIMA-GARCH dapat mengidentifikasi pola historis dan memberikan estimasi tren, pasar saham tetap dipengaruhi oleh faktor eksternal yang tidak selalu dapat dimasukkan ke dalam model, seperti sentimen pasar, kebijakan pemerintah, dan kejadian tak terduga. Ayat ini mengajarkan bahwa meskipun analisis dan teknologi dapat membantu dalam pengambilan keputusan investasi, hasil akhirnya tetap berada dalam ketentuan Allah, sehingga investor harus tetap berhati-hati, menerapkan manajemen risiko, dan tidak sepenuhnya bergantung pada prediksi model.

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Hasil Penelitian

Dalam suatu penelitian, diperlukan dukungan dari hasil-hasil penelitian sebelumnya yang relevan untuk memahami kontribusi penelitian ini terhadap perkembangan ilmu pengetahuan. Oleh karena itu, bab ini menyajikan tinjauan terhadap beberapa penelitian terdahulu yang berkaitan dengan peramalan Indeks Harga Saham Gabungan (IHSG) menggunakan metode *Autoregressive Integrated Moving Average - Generalized Autoregressive Conditional Heteroskedasticity* (ARIMA-GARCH) dan *Long Short-Term Memory* (LSTM).

2.1.1 Penelitian Terdahulu

Berikut adalah Tabel 2.1 yang menyajikan tinjauan pustaka dari berbagai penelitian terdahulu yang membahas prediksi IHSG dengan pendekatan ARIMA-GARCH dan LSTM, khususnya dalam konteks sebelum, selama, dan setelah pandemi Covid-19.

Tabel 2.1. Penelitian-penelitian Terdahulu

No	Penulis	Judul	Range Data	Keakuratan	Penjelasan	Perbedaan
1	Chung, H., & Shin, K. S. (2018).	<i>Genetic algorithm-optimized long short-term memory network for stock market prediction</i>	2000 - 2016	MAPE: 0.91%	Mengusulkan kombinasi <i>Long Short-Term Memory</i> (LSTM) dan <i>Genetic Algorithm</i> (GA) untuk mengoptimalkan prediksi pasar saham. GA digunakan untuk menentukan parameter optimal dalam arsitektur LSTM, termasuk ukuran jendela waktu dan jumlah unit LSTM dalam lapisan tersembunyi. Eksperimen menunjukkan bahwa pendekatan ini lebih baik dalam menangkap pola temporal pasar dibandingkan model konvensional	Paper ini mengoptimalkan LSTM dengan <i>Genetic Algorithm</i> di KOSPI, sementara penelitian ini mengombinasikan ARIMA-GARCH dan LSTM serta menganalisis dampak Covid-19 pada IHSG.

No	Penulis	Judul	Range Data	Keakuratan	Penjelasan	Perbedaan
2	Fischer, T., & Krauss, C. (2018).	<i>Deep learning with long short-term memory networks for financial market predictions.</i>	1992–2015	Tidak disebutkan	Membandingkan kinerja LSTM dalam memprediksi pergerakan indeks saham pada S&P 500 dengan berbagai metode lain. LSTM terbukti unggul dalam menangkap pola kompleks dalam data pasar keuangan dibandingkan model <i>free memory</i> seperti <i>Random Forest</i> dan <i>Logistic Regression</i> . Namun, keunggulan model ini berkurang setelah tahun 2010 karena arbitrase pasar	Paper ini hanya menggunakan LSTM untuk memprediksi S&P 500, sedangkan penelitian ini mengombinasikan ARIMA-GARCH dan LSTM serta menganalisis IHSG dan dampak Covid-19.
3	Budiharto, W. (2021).	<i>Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM)</i>	2018-2020	Best Accuracy: 94.59%	Mengembangkan model LSTM berbasis data besar menggunakan R dan Python untuk memprediksi harga saham di Indonesia selama pandemi Covid-19. Hasil eksperimen menunjukkan bahwa prediksi jangka pendek (1 tahun) lebih akurat dibanding jangka panjang	Paper ini hanya menggunakan LSTM untuk memprediksi saham BCA dan Bank Mandiri selama Covid-19, sedangkan penelitian ini mengombinasikan ARIMA-GARCH dan LSTM serta menganalisis IHSG sebelum, selama, dan setelah Covid-19.

No	Penulis	Judul	Range Data	Keakuratan	Penjelasan	Perbedaan
4	Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022)	<i>Predicting stock market index using LSTM</i>	2006–2020	RMSE: Kisaran 40.45 - 197.24 tergantung jumlah neuron; MAPE: Kisaran 0.79 - 5.40 tergantung jumlah neuron;	Membandingkan model LSTM satu lapisan dan multilayer dalam memprediksi indeks S&P 500. Model LSTM satu lapisan memberikan akurasi prediksi yang lebih tinggi dibandingkan multilayer, dengan pemilihan fitur dari data fundamental, makroekonomi, dan indikator teknikal	Paper ini hanya menggunakan LSTM untuk memprediksi S&P 500, sementara penelitian ini mengombinasikan ARIMA-GARCH dan LSTM serta menganalisis IHSG dan dampak Covid-19.
5	Moghar, A., & Hamiche, M. (2020).	<i>Stock market prediction using LSTM recurrent neural network.</i>	2010-2019	Tidak disebutkan	Penerapan LSTM untuk memprediksi harga saham, dengan fokus pada pengaruh jumlah epoch terhadap akurasi prediksi. Paper ini membandingkan performa LSTM dengan berbagai epoch dalam analisis kuantitatif keuangan	Paper ini hanya menggunakan LSTM untuk memprediksi saham GOOGL dan NKE, sementara penelitian ini mengombinasikan ARIMA-GARCH dan LSTM serta berfokus pada IHSG dan dampak Covid-19.
6	Abidha, H. L., & Ahdika, A. (2024)	<i>Unlocking market insight: Forecasting PT Bank Central Asia Tbk stock prices with ARIMA-GARCH analysis</i>	2020-2023	MAPE: 0.9610548	Memprediksi harga saham BBKA menggunakan ARIMA-GARCH, yang mampu menangkap tren harga serta volatilitas saham. Model terbaik yang ditemukan adalah ARIMA(0,1,1)-GARCH(6,0), yang memiliki MAPE rendah, menunjukkan bahwa model ini dapat memberikan prediksi harga saham BBKA dengan tingkat akurasi yang tinggi.	Paper ini hanya meramalkan saham BBKA dengan ARIMA-GARCH, sedangkan penelitian ini menganalisis IHSG dengan ARIMA-GARCH dan LSTM dalam tiga periode (sebelum, saat, dan setelah Covid-19).

No	Penulis	Judul	Range Data	Keakuratan	Penjelasan	Perbedaan
7	Fadhilah, D. N., Parmikanti, K., & Ruchjana, B. N. (2024)	Peramalan Return Saham Subsektor Perbankan Menggunakan Model ARIMA-GARCH.	2019-2024	RMSE: 0.01628	Penggunaan model ARIMA-GARCH untuk memodelkan volatilitas return saham subsektor perbankan, khususnya saham BBRI. Hasil penelitian menunjukkan bahwa model ARIMA(2,0,2)-GARCH(1,1) adalah yang paling optimal dalam menangkap fluktuasi harga saham dan memberikan prediksi yang lebih akurat. Selain itu, penelitian ini menyoroti tren peningkatan volatilitas return saham dari waktu ke waktu, yang dapat menjadi informasi penting bagi investor dalam mengelola risiko investasi di sektor perbankan.	Paper ini hanya meramalkan return saham BBRI dengan ARIMA-GARCH, sementara penelitian ini menganalisis IHSG dengan ARIMA-GARCH dan LSTM dalam tiga periode (sebelum, saat, dan setelah Covid-19).
8	Putri, F. T. A., Zukhronah, E., & Pratiwi, H. (2021).	Model ARIMA-GARCH Pada Peramalan Harga Saham PT. Jasa Marga (Persero)	2020-2021	MAPE: 1.5647	pengaplikasian ARIMA-GARCH untuk menangani volatilitas dalam peramalan harga saham PT Jasa Marga. Model terbaik yang ditemukan adalah ARIMA(1,1,1)-GARCH(2,2), yang menunjukkan peningkatan akurasi dalam menangkap pola harga saham dibandingkan model ARIMA konvensional.	Paper ini hanya meramalkan harga saham PT Jasa Marga dengan ARIMA-GARCH, sedangkan penelitian ini menganalisis IHSG dengan ARIMA-GARCH dan LSTM dalam tiga periode (sebelum, saat, dan setelah Covid-19).

No	Penulis	Judul	Range Data	Keakuratan	Penjelasan	Perbedaan
9	Mustapa, F. H., & Ismail, M. T. (2019)	<i>Modelling and forecasting S&P 500 stock prices using hybrid ARIMA-GARCHH Model.</i>	2001-2017	AIC = 10.6374	Penggunaan ARIMA-GARCH untuk memodelkan dan meramalkan harga indeks S&P 500 sebagai salah satu indikator utama perekonomian AS. Dengan menggunakan metode hybrid ARIMA(2,1,2)-GARCH(1,1), penelitian ini menunjukkan bahwa model ini lebih baik dalam menangkap pola fluktuasi indeks dibandingkan model ARIMA biasa. Hasilnya menunjukkan bahwa dynamic forecasting lebih akurat dibandingkan static forecasting, yang dapat membantu investor dalam mengambil keputusan investasi yang lebih baik di pasar modal AS.	Paper ini meramalkan indeks S&P 500 dengan ARIMA-GARCH, sementara penelitian ini menganalisis IHSG dengan ARIMA-GARCH dan LSTM, serta membandingkan pola sebelum, saat, dan setelah Covid-19.
10	Pahlavani, M., & Roshan, R. (2015).	<i>The comparison among ARIMA and hybrid ARIMA-GARCH models in forecasting the exchange rate of Iran.</i>	2014-2015	MAPE: 0.000661 - 0.000833 (Tergantung model)	Membandingkan beberapa model peramalan, termasuk ARIMA dan berbagai varian ARIMA-GARCH seperti IGARCH, GJR-GARCH, dan EGARCH. Tujuan utama penelitian ini adalah menangkap volatilitas nilai tukar Iran dan menentukan model terbaik untuk prediksi jangka pendek dan menengah. Hasil penelitian menunjukkan bahwa ARIMA((7,2),(12))-EGARCH(2,1) adalah model yang paling baik dalam menangkap volatilitas dan leverage effect, sehingga lebih unggul dibandingkan model lainnya dalam memprediksi pergerakan nilai tukar Rial Iran.	Paper ini meramalkan nilai tukar Rial Iran (IRR/USD) dengan berbagai model ARIMA-GARCH, sedangkan penelitian ini menganalisis IHSG dengan ARIMA-GARCH dan LSTM, serta membandingkan pola sebelum, saat, dan setelah Covid-19.

Sumber: Referensi Penelitian Terdahulu

Meskipun Tabel 2.1 telah menyajikan beberapa penelitian dengan tema serupa, penelitian ini memiliki perbedaan utama dalam fokus dan metodologi yang digunakan. Penelitian ini secara khusus menganalisis peramalan Indeks Harga Saham Gabungan (IHSG) dengan membandingkan metode ARIMA-GARCH dan LSTM dalam tiga periode berbeda: sebelum, selama, dan setelah pandemi Covid-19. Pendekatan ini memberikan perspektif baru dalam memahami bagaimana kedua model menangkap pola volatilitas dan tren IHSG dalam kondisi pasar yang berubah akibat pandemi.

2.2 Konsep Dasar

2.2.1 Pasar Modal

Pasar modal merupakan salah satu pilar utama dalam sistem keuangan yang berfungsi sebagai sarana untuk menghimpun dana dari masyarakat dan menyalurkannya ke berbagai sektor ekonomi yang membutuhkan pembiayaan. Menurut Juliati (2015), pasar modal adalah tempat terorganisir yang mempertemukan pihak yang memiliki kelebihan dana dengan pihak yang membutuhkan dana melalui berbagai instrumen keuangan seperti saham, obligasi, dan derivatif. Pasar modal memiliki peran strategis dalam mendukung pertumbuhan ekonomi dengan menyediakan sumber pendanaan jangka panjang bagi perusahaan, pemerintah, serta institusi lainnya. Dengan demikian, pasar modal berfungsi sebagai mekanisme utama dalam alokasi sumber daya keuangan yang efisien dan optimal.

Pasar modal terdiri dari dua jenis utama, yaitu pasar perdana dan pasar sekunder. Pasar perdana adalah tempat pertama kali diterbitkannya efek seperti saham dan obligasi oleh perusahaan untuk memperoleh dana langsung dari investor. Setelah melewati tahap ini, efek yang telah diterbitkan kemudian diperjualbelikan di pasar sekunder, di mana investor dapat membeli dan menjual instrumen investasi tersebut sesuai dengan harga pasar yang terbentuk berdasarkan mekanisme penawaran dan permintaan (Mubarok, 2017).

Di Indonesia, pasar modal mengalami perkembangan pesat sejak deregulasi pada tahun 1988 yang memberikan kesempatan bagi perusahaan untuk memperoleh dana melalui penerbitan saham dan obligasi di Bursa Efek Indonesia (BEI) (Muklis, 2016). Sejak saat itu, IHSG (Indeks Harga Saham Gabungan) menjadi salah satu indikator utama yang mencerminkan kinerja pasar saham Indonesia.

Menurut Oktarina, (2015) pergerakan IHSG dipengaruhi oleh berbagai faktor seperti pertumbuhan ekonomi, inflasi, suku bunga, nilai tukar, serta stabilitas politik dan ekonomi global. Selain itu, kebijakan makroekonomi seperti intervensi pemerintah dalam bentuk stimulus fiskal dan kebijakan moneter juga dapat mempengaruhi pergerakan indeks saham. Sejarah mencatat bahwa selama krisis ekonomi, baik yang terjadi secara domestik maupun global, IHSG mengalami fluktuasi tajam yang mencerminkan ketidakpastian pasar dan sentimen investor.

Selain berfungsi sebagai sumber pendanaan bagi perusahaan, pasar modal juga menjadi wadah bagi investor untuk mengalokasikan modal mereka guna memperoleh keuntungan. Menurut Suhardini (2015), pasar modal memungkinkan diversifikasi risiko melalui berbagai instrumen investasi yang tersedia, sehingga investor dapat menyesuaikan strategi investasi mereka berdasarkan tingkat risiko yang dapat diterima. Diversifikasi ini memungkinkan investor untuk mengoptimalkan portofolio investasi mereka dengan mengalokasikan dana ke berbagai jenis efek yang memiliki tingkat risiko dan potensi keuntungan yang berbeda.

Namun, pasar modal juga rentan terhadap fluktuasi harga yang dipengaruhi oleh faktor internal dan eksternal, seperti kebijakan moneter, perubahan regulasi, serta kejadian global seperti krisis keuangan atau pandemi COVID-19 (Sambuari, Saerang dan Maramis (2020). Misalnya, selama pandemi COVID-19, pasar modal mengalami volatilitas yang tinggi akibat ketidakpastian ekonomi yang disebabkan oleh kebijakan pembatasan aktivitas dan penurunan daya beli masyarakat. Meskipun demikian, pasar modal juga memiliki mekanisme pemulihan yang memungkinkan kembalinya kepercayaan investor melalui berbagai kebijakan stimulus dan strategi mitigasi risiko.

Di era digitalisasi, inovasi dalam pasar modal semakin berkembang dengan hadirnya teknologi finansial (*fintech*) dan perdagangan berbasis elektronik. Menurut Mahfuzh dan Yuliantari (2022), adopsi teknologi dalam pasar modal meningkatkan efisiensi, transparansi, dan aksesibilitas bagi investor individu maupun institusional. Salah satu inovasi yang signifikan adalah perdagangan berbasis kecerdasan buatan (*artificial intelligence*) yang memungkinkan analisis data pasar secara real-time untuk membantu pengambilan keputusan investasi. Selain itu, hadirnya platform perdagangan online dan aplikasi investasi telah meningkatkan partisipasi investor ritel dalam pasar modal, sehingga semakin memperluas basis investor dan meningkatkan likuiditas pasar. Dengan

perkembangan ini, peran pasar modal semakin signifikan dalam menciptakan ekosistem keuangan yang lebih inklusif dan stabil bagi perekonomian nasional maupun global.

Secara keseluruhan, pasar modal merupakan elemen penting dalam ekosistem keuangan yang tidak hanya mendukung pertumbuhan ekonomi melalui pendanaan jangka panjang, tetapi juga memberikan peluang investasi bagi berbagai pihak. Dengan adanya regulasi yang ketat serta perkembangan teknologi, pasar modal terus berkembang menjadi instrumen keuangan yang semakin efisien, transparan, dan mudah diakses oleh masyarakat luas. Ke depan, tantangan seperti ketidakpastian ekonomi global, perkembangan regulasi, serta inovasi teknologi akan menjadi faktor penentu dalam arah perkembangan pasar modal, baik di Indonesia maupun di tingkat internasional (Salisa, 2021).

2.2.1 Saham

Saham merupakan instrumen investasi yang mewakili kepemilikan seseorang dalam suatu perusahaan. Menurut Fadila dan Nuswandari (2022), kepemilikan saham memberikan hak kepada pemegangnya atas bagian keuntungan perusahaan serta hak suara dalam keputusan strategis melalui Rapat Umum Pemegang Saham (RUPS). Saham terbagi menjadi saham biasa dan saham preferen, di mana saham biasa memberikan hak suara tetapi tidak menjamin dividen tetap, sementara saham preferen memberikan prioritas dalam pembagian dividen namun sering kali tidak memiliki hak suara dalam pengambilan keputusan perusahaan. Pemegang saham juga berpotensi mendapatkan keuntungan dari capital gain, yaitu selisih harga beli dan harga jual saham yang lebih tinggi. Oleh karena itu, saham menjadi salah satu instrumen investasi yang menarik bagi berbagai kalangan investor, baik individu maupun institusi (Huda dan Suseno, 2023).

Dalam teori keuangan, pasar saham beroperasi berdasarkan hipotesis pasar efisien atau *Efficient Market Hypothesis* (EMH). Tjatan, (2015) menyatakan bahwa harga saham mencerminkan semua informasi yang tersedia di pasar, yang dikategorikan dalam tiga bentuk: *weak form*, *semi-strong form*, dan *strong form*. Dalam bentuk *weak form*, harga saham hanya mencerminkan informasi historis, sementara dalam *semi-strong form*, harga saham mencerminkan semua informasi publik, seperti laporan keuangan dan kebijakan pemerintah. Sedangkan dalam *strong form*, harga saham bahkan mencerminkan informasi privat yang hanya diketahui oleh pihak internal perusahaan.

Hipotesis ini menjadi dasar bagi banyak penelitian tentang efisiensi pasar dan mempengaruhi strategi investasi yang dilakukan oleh investor.

Pergerakan harga saham dipengaruhi oleh berbagai faktor makroekonomi, seperti inflasi, suku bunga, nilai tukar, dan pertumbuhan ekonomi. Časta (2023) menjelaskan bahwa inflasi yang tinggi dapat mengurangi daya beli masyarakat, yang pada akhirnya menurunkan profitabilitas perusahaan dan berdampak negatif terhadap harga saham. Sebaliknya, suku bunga yang lebih rendah dapat meningkatkan daya tarik investasi di pasar saham karena biaya pinjaman menjadi lebih murah bagi perusahaan, yang memungkinkan ekspansi bisnis dan peningkatan laba. Selain itu, nilai tukar yang fluktuatif juga mempengaruhi kinerja saham, terutama bagi perusahaan yang memiliki eksposur terhadap perdagangan internasional, seperti perusahaan ekspor dan impor.

Psikologi investor memainkan peran penting dalam pergerakan pasar saham. Menurut Hon, Moslehpour dan Woo (2021), teori *behavioral finance* menunjukkan bahwa investor tidak selalu rasional dalam pengambilan keputusan. Faktor psikologis seperti *overconfidence*, *loss aversion*, dan *herding behavior* sering kali menyebabkan penyimpangan harga saham dari nilai fundamentalnya. Misalnya, ketika banyak investor mengikuti tren pasar tanpa analisis yang mendalam, harga saham dapat mengalami lonjakan atau penurunan yang tidak sesuai dengan kondisi fundamental perusahaan. Hal ini menjelaskan mengapa pasar saham sering kali mengalami volatilitas yang tinggi, terutama dalam kondisi ketidakpastian ekonomi dan geopolitik.

Sentimen pasar juga memiliki dampak besar terhadap pergerakan saham. Bai et al. (2023) menjelaskan bahwa sentimen investor dapat menyebabkan harga saham bergerak secara ekstrem meskipun tidak ada perubahan signifikan dalam fundamental perusahaan. Sentimen positif biasanya mendorong investor untuk membeli saham secara agresif, sementara sentimen negatif dapat menyebabkan aksi jual besar-besaran yang mengakibatkan penurunan harga saham. Sentimen pasar sering kali dipengaruhi oleh berita ekonomi, kebijakan pemerintah, serta kondisi global, seperti perang dagang atau pandemi.

Media sosial dan berita memiliki peran yang semakin besar dalam membentuk ekspektasi pasar. Khan (2022) menunjukkan bahwa berita negatif yang tersebar luas di media massa sering kali menyebabkan aksi jual yang berlebihan, bahkan sebelum dampak fundamentalnya benar-benar terjadi. Dengan perkembangan teknologi, analisis big data dan kecerdasan buatan digunakan untuk menganalisis pola berita dan media sosial dalam memprediksi pergerakan saham. Hal ini

menambah kompleksitas dalam pasar saham, di mana informasi dapat tersebar dengan cepat dan mempengaruhi keputusan investasi dalam hitungan detik.

Teknologi finansial (*fintech*) telah mengubah cara investor berpartisipasi dalam pasar saham. Menurut Fang et al. (2021), perkembangan teknologi keuangan telah meningkatkan aksesibilitas pasar saham bagi investor ritel melalui aplikasi perdagangan daring dan platform investasi digital. Dengan adanya inovasi ini, investor dapat melakukan transaksi saham dengan mudah tanpa harus melalui perantara tradisional. Teknologi juga memungkinkan investor untuk menggunakan algoritma perdagangan berbasis kecerdasan buatan guna mengoptimalkan strategi investasi mereka. Meskipun demikian, adopsi teknologi ini juga menimbulkan tantangan baru, seperti risiko manipulasi pasar dan perlunya regulasi yang ketat.

Strategi investasi dalam saham dapat dibedakan menjadi beberapa pendekatan. Roca (2021) memperkenalkan konsep *value investing*, di mana investor mencari saham yang diperdagangkan di bawah nilai intrinsiknya berdasarkan analisis fundamental. Strategi ini berlawanan dengan *growth investing*, yang berfokus pada saham dengan potensi pertumbuhan tinggi meskipun valuasinya relatif mahal. Selain itu, strategi investasi berbasis dividen (*dividend investing*) menjadi pilihan bagi investor yang mencari pendapatan pasif dari perusahaan yang secara konsisten membayar dividen kepada pemegang saham. Setiap strategi memiliki keunggulan dan risikonya masing-masing, tergantung pada profil risiko dan tujuan investasi individu.

Perkembangan global juga mempengaruhi dinamika pasar saham. Lehkonen (2015) mencatat bahwa faktor-faktor seperti perang dagang, ketegangan geopolitik, serta perubahan kebijakan moneter di negara-negara besar seperti Amerika Serikat dan China dapat memberikan dampak signifikan terhadap volatilitas pasar saham di seluruh dunia. Investor global cenderung menyesuaikan portofolio mereka berdasarkan perkembangan ekonomi makro dan kebijakan yang diambil oleh bank sentral utama, seperti Federal Reserve dan European Central Bank. Hal ini menunjukkan bahwa pasar saham tidak hanya dipengaruhi oleh faktor domestik tetapi juga oleh dinamika global yang kompleks.

Saham merupakan instrumen investasi yang menawarkan potensi keuntungan tinggi tetapi juga memiliki risiko yang signifikan. Faktor ekonomi makro, sentimen investor, serta perkembangan teknologi dan regulasi semuanya berkontribusi terhadap dinamika pasar saham. Oleh karena itu, pemahaman yang mendalam tentang mekanisme pasar serta strategi investasi yang tepat menjadi

kunci bagi investor dalam mengambil keputusan investasi yang optimal. Dengan terus berkembangnya teknologi dan meningkatnya partisipasi investor, pasar saham akan terus mengalami transformasi dan menghadapi tantangan serta peluang baru di masa depan (Muliadi dan Fahmi, 2016).

2.2.1 Indeks Saham

Indeks saham merupakan indikator yang digunakan untuk mengukur kinerja sekelompok saham yang dipilih berdasarkan kriteria tertentu. Menurut Mie (2014), indeks saham berfungsi sebagai tolok ukur bagi investor dalam menilai pergerakan pasar secara keseluruhan serta sebagai dasar dalam pengelolaan portofolio investasi. Indeks saham dapat mencerminkan kondisi ekonomi suatu negara, karena perubahan nilainya sering kali dipengaruhi oleh faktor makroekonomi seperti inflasi, suku bunga, dan pertumbuhan ekonomi. Oleh karena itu, indeks saham menjadi instrumen penting dalam analisis pasar keuangan dan pengambilan keputusan investasi.

Metode perhitungan indeks saham dapat bervariasi tergantung pada bobot yang diberikan kepada masing-masing saham dalam indeks. Menurut Robiyanto, (2018), terdapat dua metode utama dalam menghitung indeks saham, yaitu metode berbasis kapitalisasi pasar (*market capitalization-weighted index*) dan metode berbasis harga (*price-weighted index*). Indeks berbasis kapitalisasi pasar, seperti S&P 500 dan IHSG, memberikan bobot lebih besar pada saham dengan kapitalisasi pasar yang lebih tinggi, sementara indeks berbasis harga, seperti *Dow Jones Industrial Average* (DJIA), menghitung rata-rata harga saham tanpa memperhitungkan kapitalisasi pasar. Pemilihan metode ini berpengaruh pada sensitivitas indeks terhadap pergerakan harga saham individu dalam indeks tersebut.

Indeks saham memiliki peran penting dalam strategi investasi pasif. Menurut Kusumawardana (2016), banyak investor memilih untuk berinvestasi dalam reksa dana indeks atau ETF yang mengikuti pergerakan indeks saham tertentu untuk mendapatkan diversifikasi dan mengurangi risiko investasi individu. Strategi ini dikenal sebagai *index investing* dan telah menjadi pendekatan yang semakin populer di kalangan investor ritel maupun institusional. Dengan berinvestasi dalam indeks, investor dapat memperoleh eksposur pasar yang luas tanpa perlu melakukan analisis individual terhadap setiap saham dalam portofolio mereka.

Pergerakan indeks saham dipengaruhi oleh berbagai faktor eksternal, termasuk kebijakan moneter dan fiskal. Menurut Annisa dan Riofita (2024), kebijakan suku bunga yang ditetapkan oleh bank sentral memiliki dampak langsung terhadap pergerakan indeks saham. Suku bunga yang rendah dapat meningkatkan daya tarik investasi di pasar saham karena biaya pinjaman yang lebih murah bagi perusahaan, sehingga berpotensi meningkatkan laba perusahaan dan nilai saham. Sebaliknya, kenaikan suku bunga dapat menekan harga saham karena meningkatkan biaya modal dan mengurangi daya beli investor. Oleh karena itu, pengumuman kebijakan moneter sering kali menjadi perhatian utama para pelaku pasar saham.

Selain faktor ekonomi, sentimen pasar dan volatilitas juga berperan dalam menentukan pergerakan indeks saham. Menurut Fassas dan Siriopoulos (2021), indeks saham dapat mengalami fluktuasi tajam akibat perubahan sentimen investor, terutama dalam situasi ketidakpastian ekonomi atau krisis keuangan. Sentimen pasar yang positif dapat mendorong kenaikan indeks saham melalui peningkatan aktivitas pembelian, sedangkan sentimen negatif dapat menyebabkan aksi jual besar-besaran yang menekan harga saham secara keseluruhan. Faktor psikologis ini menunjukkan bahwa pergerakan indeks saham tidak hanya dipengaruhi oleh analisis fundamental, tetapi juga oleh faktor perilaku investor.

Secara global, indeks saham utama seperti S&P 500, Nikkei 225, dan FTSE 100 menjadi acuan bagi pasar keuangan di berbagai negara. Pagnottoni et al. (2022) menjelaskan bahwa indeks saham global sering kali bergerak secara bersamaan sebagai respons terhadap kejadian ekonomi dan geopolitik yang signifikan. Misalnya, krisis keuangan global atau perubahan kebijakan perdagangan dapat menyebabkan penurunan indeks saham secara serentak di berbagai negara. Hal ini menunjukkan bahwa pasar saham semakin terintegrasi dalam ekonomi global, sehingga pergerakan indeks saham di satu negara dapat mempengaruhi indeks saham di negara lain.

Di Indonesia, Indeks Harga Saham Gabungan (IHSG) merupakan indeks utama yang mencerminkan pergerakan seluruh saham yang tercatat di Bursa Efek Indonesia. Menurut Desfiandi, Desfiandi dan Ali (2017), IHSG mencerminkan kondisi perekonomian Indonesia serta menjadi acuan bagi investor dalam menilai kinerja pasar saham domestik. Selain IHSG, terdapat pula indeks sektoral yang mengukur kinerja saham dalam industri tertentu, seperti indeks LQ45 yang terdiri dari 45 saham dengan likuiditas dan kapitalisasi pasar terbesar. Perkembangan indeks

saham di Indonesia menunjukkan pertumbuhan yang signifikan dalam beberapa dekade terakhir, didorong oleh peningkatan investasi domestik maupun asing serta reformasi pasar keuangan.

2.2.1 Indeks Harga Saham Gabungan

Indeks Harga Saham Gabungan (IHSG) pertama kali diperkenalkan pada tanggal 1 April 1983 oleh Bursa Efek Indonesia (BEI), yang saat itu masih dikenal sebagai Bursa Efek Jakarta (BEJ). Menurut Wahyudi dan Ramani (2022) IHSG dibuat sebagai indikator utama untuk mengukur kinerja pasar saham Indonesia secara keseluruhan. Pada saat pertama kali diperkenalkan, IHSG memiliki nilai dasar sebesar 100, dan sejak itu telah mengalami pertumbuhan yang signifikan seiring dengan perkembangan ekonomi dan pasar modal di Indonesia. Selama beberapa dekade terakhir, IHSG mengalami berbagai fluktuasi akibat perubahan kondisi ekonomi domestik maupun global, termasuk krisis keuangan Asia pada 1997-1998, krisis keuangan global 2008, serta pandemi COVID-19 pada 2020. Meskipun mengalami berbagai tantangan, IHSG tetap menjadi acuan utama bagi investor dalam menilai kondisi pasar saham di Indonesia serta menjadi alat penting dalam analisis investasi.

Menurut Wulan et al. (2023), IHSG merupakan indikator utama yang mencerminkan kinerja keseluruhan pasar saham di Indonesia, mencakup seluruh saham yang tercatat di Bursa Efek Indonesia (BEI). Sebagai barometer pasar, IHSG digunakan untuk mengukur perubahan harga rata-rata saham dalam periode tertentu, memberikan wawasan penting bagi investor, analis, serta pembuat kebijakan dalam menilai tren pasar dan potensi investasi. Dengan memahami pergerakan IHSG, investor dapat mengevaluasi kondisi pasar dan menentukan strategi investasi yang sesuai. Perubahan IHSG juga mencerminkan respons pasar terhadap berbagai peristiwa ekonomi dan politik, baik di dalam negeri maupun global. Oleh karena itu, indeks ini menjadi salah satu ukuran utama dalam analisis pasar modal, membantu investor dalam mengukur risiko dan peluang investasi dalam berbagai kondisi ekonomi. Selain itu, pergerakan IHSG sering digunakan oleh pemerintah dan otoritas keuangan dalam merancang kebijakan ekonomi guna menjaga stabilitas pasar dan mendorong pertumbuhan ekonomi yang berkelanjutan.

Ardian et al. (2024) menambahkan bahwa secara historis, pergerakan IHSG sangat dipengaruhi oleh berbagai faktor, baik domestik maupun global. Faktor-faktor seperti pertumbuhan ekonomi, tingkat inflasi, suku bunga, nilai tukar, stabilitas politik, hingga dinamika pasar internasional

memiliki peran signifikan dalam menentukan arah IHSG. Selain itu, kebijakan moneter bank sentral, baik Bank Indonesia maupun The Fed (*Federal Reserve*) di Amerika Serikat, dapat memberikan dampak besar terhadap pergerakan indeks. Misalnya, kenaikan suku bunga acuan oleh The Fed cenderung mendorong arus modal keluar dari pasar saham domestik menuju aset yang lebih aman, seperti obligasi pemerintah AS. Begitu pula, fluktuasi harga komoditas global, seperti minyak mentah dan batu bara, dapat memengaruhi kinerja sektor-sektor tertentu di pasar saham Indonesia, terutama sektor energi dan pertambangan. Faktor geopolitik, seperti konflik antarnegara atau ketegangan perdagangan global, juga dapat meningkatkan volatilitas IHSG, karena investor cenderung menghindari risiko di tengah ketidakpastian ekonomi yang tinggi. Dengan demikian, memahami faktor-faktor yang memengaruhi IHSG sangat penting bagi pelaku pasar untuk mengembangkan strategi investasi yang lebih adaptif dan efektif.

Sementara itu, studi yang dilakukan oleh Devi dan Wibowo (2021) menunjukkan bahwa IHSG memiliki sensitivitas tinggi terhadap perubahan sentimen pasar, terutama selama periode ketidakpastian ekonomi atau gejolak global, seperti krisis keuangan dan pandemi COVID-19. Saat pandemi melanda, IHSG mengalami tekanan jual yang signifikan akibat kepanikan investor terhadap dampak ekonomi yang ditimbulkan oleh pembatasan aktivitas dan perlambatan pertumbuhan ekonomi. Namun, seiring dengan diterapkannya stimulus fiskal dan moneter oleh pemerintah serta pemulihan aktivitas ekonomi, IHSG mulai berangsur pulih dan kembali mencatatkan tren positif. Studi ini juga menekankan bahwa volatilitas IHSG selama krisis ekonomi dapat menjadi peluang bagi investor jangka panjang yang mampu memanfaatkan momentum pasar untuk membeli saham di harga rendah dan mendapatkan keuntungan ketika pasar mulai stabil. Selain itu, IHSG juga memiliki peran strategis dalam pengelolaan portofolio investasi, baik bagi investor institusional maupun ritel. Banyak manajer investasi yang menggunakan IHSG sebagai acuan dalam strategi alokasi aset, terutama dalam pengelolaan reksa dana berbasis indeks, dana pensiun, hingga instrumen derivatif seperti indeks futures dan *Exchange Traded Fund* (ETF). Oleh karena itu, pemahaman terhadap dinamika IHSG tidak hanya penting dalam analisis pasar saham, tetapi juga dalam perencanaan investasi jangka panjang yang lebih optimal.

2.2.1 Covid-19

Sebelum pandemi COVID-19 melanda dunia, perekonomian global berada dalam kondisi pertumbuhan yang stabil dengan tren ekspansi di berbagai sektor industri. Menurut Altig et al. (2020), pertumbuhan ekonomi dunia diperkirakan mencapai 2,9% pada tahun tersebut, didorong oleh meningkatnya perdagangan internasional, investasi langsung asing, serta perkembangan teknologi dan digitalisasi. Stabilitas pasar keuangan juga tercermin dari tren positif dalam pasar saham, rendahnya tingkat pengangguran di banyak negara maju, serta meningkatnya daya beli masyarakat global.

Di bidang kesehatan, sistem kesehatan di berbagai negara masih berfokus pada penanganan penyakit menular yang sudah dikenal serta penyakit tidak menular seperti diabetes dan penyakit jantung. WHO (2018) mencatat bahwa meskipun terdapat beberapa ancaman pandemi dari virus-virus seperti Ebola dan flu burung, upaya pencegahan dan kesiapsiagaan global terhadap kemungkinan wabah pandemi masih relatif terbatas. Sistem kesehatan banyak negara tidak memiliki kesiapan yang cukup dalam menangani lonjakan kasus secara bersamaan, termasuk keterbatasan infrastruktur medis dan tenaga kesehatan.

Menurut Kumar et al. (2021) COVID-19 pertama kali terdeteksi di Kota Wuhan, Provinsi Hubei, Tiongkok pada akhir Desember 2019. Virus ini disebabkan oleh Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), yang termasuk dalam keluarga virus corona yang sebelumnya telah menyebabkan wabah SARS pada tahun 2002-2003 dan MERS pada tahun 2012. Menurut penelitian yang dilakukan oleh Andersen et al. (2020), virus ini diduga berasal dari kelelawar, dengan kemungkinan perantara hewan lain sebelum akhirnya menular ke manusia. Penularan pertama dikaitkan dengan pasar makanan laut dan hewan liar di Wuhan, yang diduga menjadi tempat awal penyebaran virus melalui interaksi manusia dengan hewan yang terinfeksi.

Penyebaran COVID-19 terjadi dengan sangat cepat akibat mobilitas manusia yang tinggi, terutama melalui perjalanan internasional. WHO secara resmi menetapkan COVID-19 sebagai pandemi global pada 11 Maret 2020 setelah lebih dari 100 negara melaporkan kasus infeksi yang terus meningkat secara eksponensial. Berdasarkan studi yang dilakukan oleh Alimohamadi, Taghdir dan Sepandi (2020), tingkat reproduksi dasar (R_0) SARS-CoV-2 diperkirakan lebih tinggi dibandingkan virus corona sebelumnya, sehingga memudahkan penyebarannya melalui droplet pernapasan, kontak langsung, serta kemungkinan melalui aerosol dalam ruang tertutup.

COVID-19 berdampak besar pada sistem kesehatan global, dengan rumah sakit dan fasilitas kesehatan mengalami tekanan berat akibat lonjakan pasien yang memerlukan perawatan intensif. Menurut laporan dari Sanyaolu et al. (2020), pasien dengan komorbiditas seperti diabetes, hipertensi, dan penyakit paru-paru kronis memiliki risiko lebih tinggi untuk mengalami kondisi parah akibat infeksi SARS-CoV-2. Gejala yang ditimbulkan berkisar dari ringan hingga berat, dengan beberapa pasien mengalami pneumonia akut dan sindrom gangguan pernapasan akut (ARDS), yang memerlukan ventilasi mekanis.

Selain dampak fisik, pandemi ini juga menyebabkan lonjakan gangguan kesehatan mental. Menurut penelitian Cullen, Gulati dan Kelly (2020), isolasi sosial yang berkepanjangan, ketakutan terhadap infeksi, serta ketidakpastian ekonomi meningkatkan prevalensi gangguan kecemasan, depresi, serta stres pasca-trauma (PTSD). Kelompok yang paling terdampak termasuk tenaga medis yang bekerja di garis depan, yang mengalami kelelahan ekstrem akibat beban kerja tinggi dan risiko paparan virus secara terus-menerus.

Pandemi COVID-19 menyebabkan perlambatan ekonomi global yang signifikan akibat kebijakan lockdown, pembatasan perjalanan, dan gangguan pada rantai pasokan global. Brodeur et al. (2021) menjelaskan bahwa berbagai sektor ekonomi mengalami kontraksi yang tajam, terutama sektor pariwisata, transportasi, manufaktur, dan perhotelan yang sangat bergantung pada mobilitas manusia. Banyak perusahaan mengalami kebangkrutan dan terjadi pemutusan hubungan kerja secara massal, meningkatkan angka pengangguran di berbagai negara.

Sektor pendidikan juga mengalami perubahan drastis selama pandemi. Menurut Daniel (2020), lebih dari 1,6 miliar siswa di seluruh dunia terdampak oleh penutupan sekolah, yang menyebabkan peralihan besar-besaran ke pembelajaran daring. Meskipun metode ini memungkinkan keberlanjutan pendidikan, banyak negara berkembang menghadapi kesenjangan digital yang menyebabkan ketimpangan akses terhadap pembelajaran.

Meskipun pandemi menyebabkan dampak ekonomi negatif, beberapa sektor justru mengalami pertumbuhan. Laporan dari Ting et al. (2020) menunjukkan bahwa pandemi mempercepat digitalisasi ekonomi, dengan meningkatnya penggunaan layanan digital seperti e-commerce, pembayaran digital, dan konferensi virtual. Perusahaan yang mampu beradaptasi dengan teknologi digital mengalami pertumbuhan pesat, sementara bisnis yang masih bergantung pada sistem konvensional menghadapi tantangan besar dalam mempertahankan kelangsungan usahanya.

Setelah COVID-19 mulai mereda dan vaksinasi massal diterapkan, berbagai negara mulai mengalami pemulihan ekonomi secara bertahap. Hargreaves dan Logie (2020) melaporkan bahwa banyak negara mulai menghapus kebijakan lockdown dan membuka kembali perbatasan internasional untuk menghidupkan kembali sektor pariwisata dan perdagangan global. Namun, pemulihan ekonomi tidak merata di semua sektor, karena beberapa industri masih menghadapi tantangan dalam pemulihan, terutama yang terkait dengan rantai pasokan global dan tenaga kerja.

Di sisi kesehatan, meskipun vaksin telah membantu mengurangi tingkat keparahan infeksi COVID-19, dampak jangka panjang bagi pasien penyintas COVID-19 masih menjadi tantangan bagi dunia medis. Laporan Yong (2021) menunjukkan bahwa beberapa individu mengalami gejala berkepanjangan yang dikenal sebagai Long COVID, yang meliputi kelelahan kronis, gangguan pernapasan, serta masalah neurologis.

Dalam bidang sosial dan pendidikan, pandemi COVID-19 telah mempercepat adopsi pembelajaran daring dan digitalisasi dalam berbagai aspek kehidupan. Namun, kesenjangan digital menjadi tantangan bagi negara berkembang yang masih memiliki akses internet terbatas. Menurut Maity, Sahu dan Sen (2021), banyak siswa mengalami ketertinggalan akademik akibat kesulitan mengakses pembelajaran daring selama pandemi.

Pandemi COVID-19 merupakan peristiwa yang telah mengubah berbagai aspek kehidupan manusia, baik dalam bidang kesehatan, ekonomi, maupun sosial. Menurut Noerkaisar (2021) Dampak dari pandemi Covid-19 menunjukkan betapa rentannya sistem global terhadap krisis kesehatan yang muncul secara tiba-tiba. Meskipun dunia telah mengambil langkah-langkah besar dalam menangani pandemi ini, masih ada berbagai tantangan yang harus dihadapi dalam proses pemulihan. Investasi dalam penelitian, kebijakan kesehatan yang lebih kuat, serta kerja sama internasional menjadi kunci utama untuk membangun ketahanan terhadap pandemi di masa depan.

2.2.1 Deret Waktu (*Time Series*)

Deret waktu (*time series*) merupakan sekumpulan observasi yang dikumpulkan secara berurutan berdasarkan waktu dan memiliki keterkaitan antar-pengamatan. Menurut Hamilton (2020), deret waktu adalah serangkaian data yang diamati pada interval waktu tetap dengan tujuan memahami

pola historis dan meramalkan nilai di masa depan. Pola dalam deret waktu dapat menunjukkan tren, siklus, atau fluktuasi yang disebabkan oleh faktor musiman atau faktor eksternal lainnya.

Secara matematis, deret waktu dapat dinyatakan sebagai Y_t , yaitu suatu variabel yang diamati pada waktu t . Gujarati (2009) menekankan bahwa deret waktu berbeda dengan data cross-section karena adanya dependensi antar-observasi di dalamnya, di mana nilai masa lalu dapat memengaruhi nilai di masa depan. Oleh karena itu, analisis deret waktu memerlukan pendekatan yang mempertimbangkan aspek temporal agar dapat menghasilkan peramalan yang akurat.

Menurut Van Greunen et al. (2014), analisis deret waktu memiliki peran penting dalam berbagai bidang seperti ekonomi, keuangan, kesehatan, dan teknik. Dengan memahami pola historis dalam deret waktu, analis dapat mengidentifikasi perubahan struktural dalam data dan mengembangkan model peramalan yang lebih baik. Oleh karena itu, penting untuk menerapkan metode yang tepat agar hasil analisis dapat digunakan dalam pengambilan keputusan yang lebih efektif.

Deret waktu memiliki berbagai komponen utama yang membantu dalam analisis dan peramalan data. Juanda dan Junaidi (2021) mengidentifikasi empat komponen utama dalam deret waktu, yaitu tren, musiman, siklus, dan komponen acak. Tren mencerminkan arah jangka panjang dari data, baik meningkat maupun menurun, yang dapat disebabkan oleh faktor ekonomi atau sosial.

Musiman adalah pola berulang dalam periode tertentu, seperti siklus tahunan dalam penjualan ritel. Ruhiat dan Suwanda (2019) menyatakan bahwa pola musiman terjadi karena adanya pengaruh faktor eksternal seperti cuaca, perayaan hari besar, atau kebiasaan konsumen. Oleh karena itu, metode dekomposisi sering digunakan untuk memisahkan komponen musiman dari data utama agar analisis lebih akurat.

Komponen siklikal dalam deret waktu berbeda dengan pola musiman karena berlangsung dalam periode yang lebih panjang dan sering dikaitkan dengan siklus ekonomi. Menurut Fadilah et al. (2024), siklus ini dapat mencerminkan pola naik dan turunnya aktivitas ekonomi yang tidak terikat dengan periode waktu tertentu, melainkan dipengaruhi oleh dinamika makroekonomi. Dengan memahami komponen ini, pengambil kebijakan dapat mengantisipasi perubahan dalam siklus ekonomi.

Selain itu, komponen acak atau residual mencerminkan variasi dalam data yang tidak dapat dijelaskan oleh tren, musiman, atau siklus. Menurut Putri, Hendayanti dan Nurhidayati (2017),

komponen ini dapat muncul karena faktor eksternal yang tidak terprediksi, seperti bencana alam atau perubahan kebijakan yang tiba-tiba. Oleh karena itu, model deret waktu sering mencakup elemen stokastik untuk menangkap variabilitas acak dalam data.

Dengan memahami berbagai jenis dan karakteristik deret waktu, analisis data dapat dilakukan dengan lebih efektif dan akurat. Pemilihan model yang tepat sangat bergantung pada identifikasi komponen utama dalam deret waktu serta tujuan dari analisis yang dilakukan (Wauchope et al., 2021).

2.2.1 *Forecasting*

Forecasting merupakan proses analisis yang digunakan untuk memperkirakan nilai atau tren suatu variabel di masa depan berdasarkan data historis. Dalam pasar keuangan, *forecasting* banyak digunakan untuk memprediksi pergerakan harga saham, indeks pasar, serta indikator ekonomi lainnya dengan tujuan membantu pengambilan keputusan investasi. Metode *forecasting* dapat dikategorikan menjadi pendekatan statistik, seperti *Autoregressive Integrated Moving Average* (ARIMA), serta pendekatan berbasis kecerdasan buatan (AI), seperti *Long Short-Term Memory* (LSTM). Keakuratan hasil forecasting sangat bergantung pada kualitas data yang digunakan serta ketepatan model yang diterapkan dalam analisis (Petroopoulos et al., 2022).

Perc et al. (2020) menyatakan bahwa teknik *forecasting* merupakan metode yang digunakan untuk memprediksi harga saham di masa depan dengan mengandalkan data historis guna menganalisis tren dan pola yang ada. Akurasi hasil *forecasting* sangat bergantung pada kualitas serta relevansi data yang digunakan dalam analisis. Dalam kondisi pasar yang normal, metode forecasting dapat menghasilkan prediksi yang relatif akurat, dengan kecenderungan harga saham meningkat seiring pertumbuhan ekonomi yang stabil. Namun, pada saat terjadi krisis ekonomi atau kondisi pasar yang tidak terduga, seperti resesi atau krisis finansial, akurasi forecasting sering kali menurun akibat volatilitas yang tinggi dan ketidakpastian pasar. Oleh karena itu, untuk meningkatkan efektivitas prediksi, diperlukan data yang relevan dan terbaru, serta pertimbangan terhadap faktor eksternal seperti kebijakan pemerintah, dinamika ekonomi global, dan peristiwa tak terduga lainnya.

Foroni et al. (2022) menyoroti tantangan dalam melakukan *forecasting* selama periode krisis, terutama dalam situasi yang tidak terduga seperti pandemi COVID-19. Mereka berpendapat bahwa tingkat ketidakpastian yang tinggi membuat prediksi harga saham dan kondisi pasar menjadi jauh lebih sulit. Salah satu kendala utama dalam forecasting pada masa krisis adalah ketidakmampuan untuk secara akurat menentukan kapan proses pemulihan ekonomi akan terjadi. Dalam kasus pandemi COVID-19, misalnya, meskipun berbagai model telah dikembangkan untuk memperkirakan dampaknya terhadap perekonomian, faktor-faktor seperti durasi pandemi, efektivitas vaksin, dan respons kebijakan global tetap menjadi elemen yang sulit diprediksi. Selain itu, volatilitas pasar yang ekstrem selama periode krisis semakin memperumit prediksi harga saham dan aset keuangan lainnya. Oleh karena itu, Foroni et al. (2022) menekankan pentingnya mengadopsi pendekatan *forecasting* yang lebih fleksibel dan responsif terhadap perubahan kondisi, serta mempertimbangkan variabel-variabel yang tidak dapat diprediksi yang dapat memengaruhi pemulihan ekonomi secara keseluruhan.

2.2.1 Autocorrelation Function (ACF)

Korelasi merupakan ukuran hubungan linear antara dua variabel yang berbeda, sedangkan autokorelasi mengacu pada kondisi di mana terdapat keterkaitan antara nilai-nilai dalam suatu deret waktu yang sama pada titik waktu yang berbeda. Dalam analisis deret waktu, *Autocorrelation Function* (ACF) digunakan untuk mengukur hubungan antara suatu variabel dengan versi tertundanya (lag) pada periode tertentu (Yakubu dan Saputra, 2022).

Menurut Alsharif, Younes dan Kim (2019) koefisien autokorelasi pada lag- k , yang dinotasikan sebagai ρ_k , merepresentasikan tingkat keterkaitan antara dua nilai dalam deret waktu, yaitu Z_t dan Z_{t+k} . Koefisien ini mengukur seberapa besar pengaruh nilai masa lalu terhadap nilai saat ini, dengan asumsi bahwa rata-rata kedua nilai tersebut tetap konstan sepanjang waktu. Secara matematis, hubungan ini dapat dinyatakan dengan persamaan sebagai berikut:

$$\rho_k = \frac{E [(Z_t - \mu)(Z_{t+k} - \mu)]}{\sigma^2} \quad (2.1)$$

dengan Z_t sebagai nilai deret waktu pada waktu t , μ sebagai rata-rata deret waktu, dan σ^2 sebagai variansnya. Jika nilai ρ_k mendekati 1, maka terdapat korelasi positif yang kuat antara observasi

pada waktu t dan $t + k$, sedangkan jika ρ_k mendekati -1, hubungan antara kedua nilai tersebut bersifat negatif. Jika ρ_k mendekati 0, maka tidak ada hubungan linear yang signifikan antara nilai pada lag yang berbeda.

Menurut Hyndman (2021), ACF merupakan alat penting dalam analisis deret waktu karena dapat membantu dalam mengidentifikasi pola tren, musiman, serta menentukan sifat stasioneritas suatu data. Dengan memahami pola autokorelasi dalam suatu data, analisis dapat memilih model yang paling sesuai untuk peramalan, seperti model autoregresif (AR) atau *moving average* (MA).

2.2.1 Partial Autocorrelation Function (PACF)

Partial Autocorrelation Function (PACF) adalah suatu fungsi statistik yang digunakan untuk mengukur hubungan antara suatu variabel dalam deret waktu dengan versi tertundanya (lag), setelah mengeliminasi pengaruh lag-lag yang berada di antara keduanya. Dengan kata lain, PACF mengisolasi korelasi langsung antara Y_t dan Y_{t-k} dengan mengontrol dampak dari lag-lag sebelumnya (Nasirudin, 2022).

Menurut Demir, Zontul dan Yelmen (2020) secara matematis, PACF dapat diinterpretasikan sebagai koefisien regresi parsial dalam model autoregresif, di mana nilai koefisien pada lag tertentu merepresentasikan hubungan murni antara dua titik waktu setelah efek dari pengamatan antara dikeluarkan. Jika ϕ_k menyatakan koefisien autokorelasi parsial pada lag-k, maka PACF pada lag-k dapat diperoleh dari estimasi regresi berikut:

$$Y_t = \alpha + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_k Y_{t-k} + \epsilon_t \quad (2.2)$$

di mana ϵ_t merupakan komponen kesalahan yang tidak berkorelasi. Nilai PACF pada lag-k sama dengan koefisien ϕ_k , yang menunjukkan seberapa besar pengaruh Y_{t-k} terhadap Y_t setelah memperhitungkan semua pengaruh dari lag-lag sebelumnya.

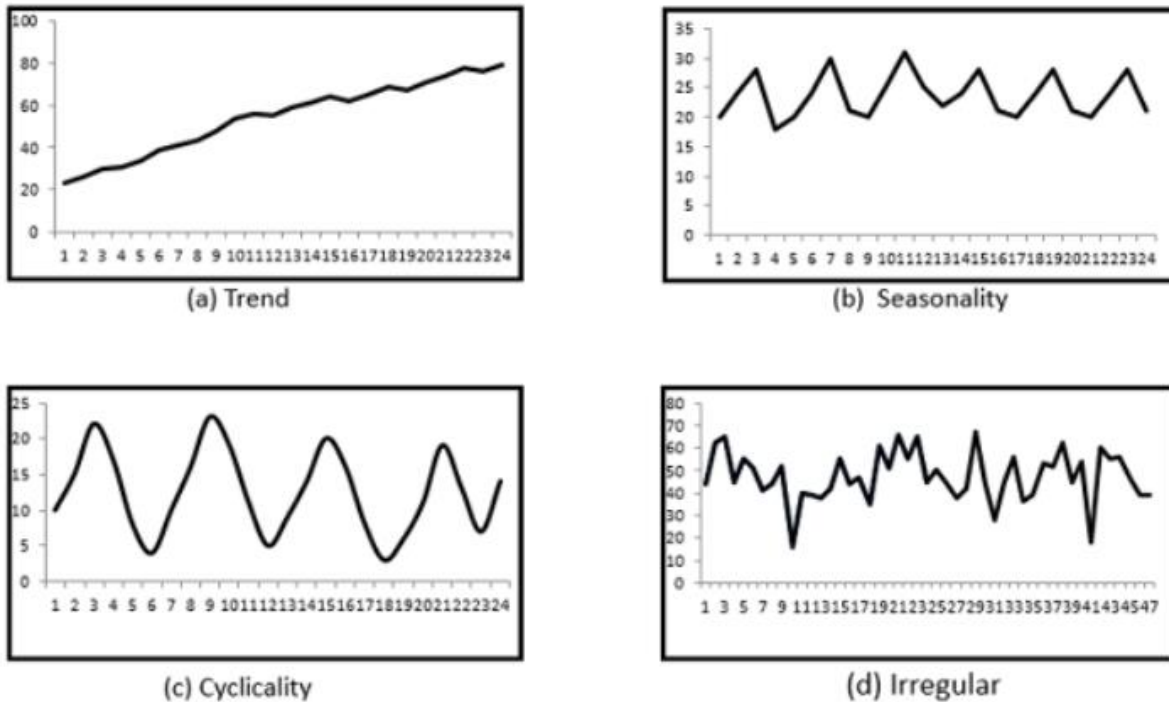
PACF sering digunakan dalam pemilihan orde model autoregresif (AR) dalam metode Box-Jenkins atau ARIMA. Menurut Wei et al. (2023), grafik PACF yang menunjukkan cutoff setelah sejumlah lag tertentu mengindikasikan adanya proses autoregresif (AR) dengan orde yang sesuai. Oleh karena itu, PACF menjadi alat penting dalam identifikasi struktur ketergantungan dalam data deret waktu serta dalam menentukan parameter optimal untuk model peramalan yang lebih akurat.

2.2.1 Stasioneritas

Dalam analisis deret waktu, suatu data dapat dikategorikan sebagai stasioner atau nonstasioner. Deret waktu dikatakan stasioner apabila memiliki keseimbangan statistik di sekitar rata-rata yang konstan dan tidak mengalami perubahan struktur dalam jangka waktu tertentu (liu et al., 2022).

Menurut Huang et al. (2024) data times series memiliki 4 pola umum sebagai berikut:

- **Trend (a)**
Pola ini menunjukkan pergerakan jangka panjang yang mengindikasikan arah peningkatan atau penurunan nilai variabel dari waktu ke waktu. Trend bisa bersifat linier atau non-linier, mencerminkan dinamika pertumbuhan atau penurunan dalam data.
- **Seasonality (b)**
Pola ini menggambarkan fluktuasi data yang terjadi secara berulang dalam periode waktu tertentu, seperti bulanan atau tahunan. Karakteristik utama dari seasonality adalah kestabilan pola yang dapat diprediksi berdasarkan siklus waktu tertentu.
- **Cyclicality (c)**
Pola ini mencerminkan pergerakan data yang berulang dalam siklus yang lebih panjang dan tidak memiliki interval tetap, sering kali dipengaruhi oleh faktor ekonomi atau bisnis. Berbeda dengan seasonality, siklus ini tidak selalu terjadi dalam jangka waktu yang konsisten.
- **Irregular (d)**
Pola ini merepresentasikan fluktuasi acak yang tidak memiliki pola tetap, sering kali disebabkan oleh faktor eksternal atau kejadian tidak terduga. Komponen ini sulit diprediksi dan biasanya dianggap sebagai noise dalam analisis time series.



Gambar 2.1. Jenis-jenis Pola Data *Times Series*

Sumber: analyticsvidhya.com/blog/2023/02/various-techniques-to-detect-and-isolate-time-series-components-using-python/

Menurut Maulana (2018), stasioneritas dapat dibagi 2 yaitu:

1. Stasioner Kuat

Suatu deret waktu dikatakan stasioner kuat apabila memiliki rata-rata dan variansi yang konstan, serta struktur autokovariansi yang stabil seiring waktu. Artinya, tidak ada perubahan pola yang signifikan dalam distribusi data di berbagai periode waktu, sehingga sifat statistiknya tidak tergantung pada waktu pengamatan.

2. Stasioner Lemah

Sebuah deret waktu dikatakan stasioner lemah jika hanya memenuhi syarat rata-rata dan variansi yang konstan, tetapi struktur autokovariansinya dapat berubah seiring waktu. Dengan kata lain, meskipun nilai rata-rata dan penyebaran data tetap, pola hubungan antar-data dalam waktu tertentu masih mungkin mengalami perubahan.

Menurut Aktivani (2020), deret waktu yang tidak stasioner dapat diklasifikasikan ke dalam tiga kategori utama:

- Tidak stasioner pada rata-rata hitung: Ditandai dengan adanya tren naik atau turun yang menunjukkan bahwa nilai rata-rata berubah seiring waktu.
- Tidak stasioner pada variansi: Variansi data tidak konstan dari waktu ke waktu, misalnya adanya perubahan volatilitas yang meningkat atau menurun secara drastis.
- Tidak stasioner pada rata-rata dan variansi: Kombinasi dari kedua kondisi sebelumnya, di mana rata-rata berubah dan variasi data juga tidak stabil sepanjang periode pengamatan.

Untuk melihat data stasioner atau tidak dapat dilihat dari

1. Plot Data Deret Waktu

Salah satu cara sederhana untuk menilai kestasioneran adalah dengan membuat plot data terhadap waktu. Jika pola data menunjukkan nilai rata-rata yang tetap sepanjang waktu dan fluktuasi data relatif stabil, maka deret waktu dapat dianggap stasioner. Sebaliknya, jika terdapat tren naik atau turun, atau volatilitas data berubah, maka deret waktu kemungkinan tidak stasioner (Hidayat dan Mustawinar, 2022).

2. Uji *Augmented Dickey-Fuller* (ADF)

Menurut Saputri, Febritama dan RA (2025) penilaian berdasarkan plot data bersifat subjektif, maka diperlukan uji statistik yang lebih formal, salah satunya adalah Uji *Augmented Dickey-Fuller* (ADF). Uji ini bertujuan untuk menguji apakah suatu deret waktu memiliki sifat stasioner atau tidak dengan menggunakan hipotesis berikut:

- Hipotesis nol (H_0): $\delta = 0 \rightarrow$ Data tidak stasioner (memiliki akar unit).
- Hipotesis alternatif (H_1): $\delta < 0 \rightarrow$ Data stasioner

Statistik uji ADF:

$$ADF = \frac{\hat{\delta}}{SE(\hat{\delta})} \quad (2.3)$$

Dimana:

- $\hat{\delta}$ adalah estimasi koefisien dari regresi ADF
- $SE(\hat{\delta})$ adalah standar error dari estimasi δ

Kriteria Uji:

- Tolak H_0 jika nilai statistik uji $T_{hitung} < T_{kritis}$ (*Dickey-Fuller*).
- Jika H_0 ditolak, maka data dianggap stasioner. Jika tidak, maka data dianggap memiliki akar unit dan tidak stasioner.

Uji ADF banyak digunakan dalam analisis deret waktu untuk memastikan data sudah memenuhi asumsi kestasioneran sebelum dilakukan pemodelan lebih lanjut, seperti dalam model ARIMA atau VAR.

Dalam analisis runtun waktu, asumsi kestasioneran data sangat penting untuk memastikan validitas model yang digunakan. Jika data tidak stasioner, maka perlu dilakukan langkah-langkah tertentu untuk mengubahnya menjadi stasioner sebelum analisis lebih lanjut dilakukan (Ruhiat dan Suwanda, 2019).

Jika ketidakstasioneran hanya terjadi pada rata-rata tetapi variansi tetap konstan, maka metode pembedaan (*differencing*) dapat diterapkan untuk menghilangkan tren dalam data (Fitriani et al., (2021). Pembedaan dilakukan dengan menggunakan operator differencing yang dinyatakan dalam rumus sebagai berikut:

$$\nabla^d = (1 - B)^d \quad (2.4)$$

dimana:

- B adalah operator shift mundur (backward shift operator), yang berfungsi untuk menggeser nilai data ke periode sebelumnya,
- d adalah orde *differencing* yang menunjukkan seberapa banyak operasi *differencing* harus dilakukan.

Sebagai contoh, untuk pembedaan orde pertama, rumusnya menjadi:

$$(1 - B)Z_t = Z_t - BZ_t = Z_t - Z_{t-1} \quad (2.5)$$

Proses ini membantu menghilangkan tren dalam data sehingga menjadi lebih stabil dan memenuhi asumsi kestasioneran.

Jika ketidakstasioneran terjadi pada variansi tetapi rata-rata tetap stabil, maka transformasi Box-Cox dapat diterapkan untuk menstabilkan fluktuasi variansi dalam data (Pamungkas dan Wibowo, 2019). Transformasi ini diberikan oleh rumus berikut:

$$X_t \rightarrow Y_t = \frac{x_t^\lambda - 1}{\lambda}, \text{ untuk } \lambda \neq 0 \quad (2.6)$$

Osbourne (2010) berpendapat nilai parameter λ menentukan jenis transformasi yang digunakan:

- $\lambda = -1$, Maka transformasi kebalikan: $Y_t = 1/X_t$
- $\lambda = -0,5$, Maka Transformasi akar kebalikan: $Y_t = 1/\sqrt{X_t}$
- $\lambda = 0$, Maka transformasi logaritmik: $Y_t = \ln X_t$
- $\lambda = 0,5$, maka transformasi akar kuadrat: $Y_t = \sqrt{X_t}$
- $\lambda = 1$, maka tidak ada transformasi X_t

Jika data mengalami ketidakstasioneran baik pada rata-rata maupun variansi, maka pendekatan yang digunakan adalah kombinasi transformasi dan *differencing*. Langkah pertama adalah menerapkan transformasi untuk menstabilkan variansi, kemudian dilanjutkan dengan *differencing* untuk menghilangkan tren dalam data (Rosyidah, Rahmawati dan Prahutama, 2018). Dengan demikian, data dapat memenuhi asumsi kestasioneran yang diperlukan dalam analisis runtun waktu.

2.3 Landasan Teori ARIMA-GARCH

2.3.1 Model ARIMA

Model *Box-Jenkins* (ARIMA) diklasifikasikan ke dalam tiga kategori utama, yaitu model *autoregressive* (AR), model *moving average* (MA), dan model campuran *autoregressive moving average* (ARIMA). Model *autoregressive* (AR) menggunakan hubungan linier antara nilai saat ini dengan nilai sebelumnya dalam deret waktu, sedangkan model *moving average* (MA)

mengandalkan hubungan linier antara nilai saat ini dengan kesalahan (*error*) dari periode sebelumnya. Sementara itu, model ARIMA merupakan kombinasi dari kedua model tersebut, di mana model ini menggabungkan komponen *autoregressive* dan *moving average* untuk menghasilkan prediksi yang lebih akurat dalam analisis deret waktu (Shumway et al., 2017).

2.3.1.1 Model Autoregressive (AR)

Menurut Dalal, Li dan Taori (2019) model *Autoregressive* (AR) adalah model dalam analisis deret waktu yang menunjukkan bahwa variabel dependen (terikat) dipengaruhi oleh nilai-nilai sebelumnya dari variabel tersebut pada periode sebelumnya. Dengan kata lain, model ini menyatakan bahwa nilai suatu variabel dalam suatu waktu tertentu bergantung pada nilai historisnya. Dalam konteks ini, variabel dependen merupakan variabel yang nilainya ditentukan oleh variabel bebas (independen), yang dalam model autoregressive adalah nilai masa lalu dari variabel itu sendiri.

Secara umum, model *autoregressive* dengan ordo p (AR(p)) merepresentasikan jumlah observasi sebelumnya yang digunakan untuk memprediksi nilai pada periode mendatang. Ordo p menunjukkan banyaknya lag yang digunakan dalam model. Bentuk matematis dari model AR(p) dapat dinyatakan dengan persamaan sebagai berikut:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t \quad (2.7)$$

atau dapat ditulis dalam bentuk operator shift mundur (*backward shift operator*) sebagai:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) X_t = e_t \quad (2.8)$$

Jika nilai harapan (ekspektasi) dari deret waktu bersifat konstan, yaitu:

$$E(X_t) = E(X_{t-1}) = \dots = E(X_{t-p}) = \mu \neq 0 \quad (2.9)$$

maka model autoregressive dapat ditulis ulang dalam bentuk berikut:

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \dots + \phi_p (X_{t-p} - \mu) + e_t \quad (2.10)$$

dimana:

- X_t : Nilai variabel pada waktu t , yang bergantung pada nilai-nilai sebelumnya.
- c : Konstanta dalam model yang dapat mencerminkan rata-rata dari proses deret waktu jika ada.
- p : Parameter *autoregressive* untuk lag ke- p , yang menunjukkan seberapa besar pengaruh nilai sebelumnya terhadap nilai saat ini.
- B : *Operator backward shift*, yang menggeser nilai deret waktu ke periode sebelumnya, misalnya $BX_t = X_{t-1}$
- μ : Nilai rata-rata dari deret waktu dalam kondisi stasioner.
- e_t : Komponen kesalahan atau gangguan acak pada waktu t , yang diasumsikan sebagai white noise dengan rata-rata nol dan variansi konstan.

2.3.1.2 Moving Average (MA)

Menurut Lu et al. (2020) model *Moving Average* (MA) berbeda dengan model *Autoregressive* (AR) dalam hal jenis variabel independennya. Pada model AR(p), variabel independen berupa nilai-nilai sebelumnya dari variabel dependen, sedangkan dalam model MA(q), variabel independennya terdiri dari nilai residual atau kesalahan (error term) pada periode sebelumnya.

Ordo dalam model MA dinyatakan dengan q , yang menunjukkan jumlah periode residual yang dimasukkan dalam model. Model ini digunakan untuk menangkap pola fluktuasi dalam data yang disebabkan oleh kejadian acak atau ketidakteraturan dalam sistem.

Bentuk umum model Moving Average dengan ordo q dapat dinyatakan sebagai berikut:

$$X_t = c + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.11)$$

atau dapat ditulis dalam bentuk operator Backward Shift (B) sebagai:

$$X_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) e^t \quad (2.12)$$

dimana:

- X_t : Nilai variabel pada waktu t , yang dipengaruhi oleh kesalahan (error term) pada periode sebelumnya.
- c : Konstanta dalam model yang mewakili rata-rata proses jika ada.

- θ_q : Parameter moving average untuk lag ke-q, yang menunjukkan sejauh mana pengaruh error sebelumnya terhadap nilai saat ini.
- e^t : Komponen kesalahan atau gangguan acak pada waktu t, yang diasumsikan sebagai white noise dengan nilai rata-rata nol dan variansi konstan.
- B : Operator shift mundur (backward shift), yang menggeser nilai residual ke periode sebelumnya, misalnya $Be_t = e_{t-1}$

Model *Moving Average* sering digunakan dalam analisis deret waktu untuk menghaluskan fluktuasi jangka pendek dalam data dan menangkap efek dari shock atau kejutan yang memengaruhi system.

2.3.1.3 Model Autoregressive Moving Average

Menurut Zhang (2018) Model *Autoregressive Moving Average* (ARMA) merupakan kombinasi antara model *Autoregressive* (AR) dan *Moving Average* (MA), yang digunakan untuk menganalisis dan meramalkan data deret waktu. Model ini dapat diperluas menjadi *Autoregressive Integrated Moving Average* (ARIMA) jika terdapat proses diferensiasi (*differencing*) untuk menangani data nonstasioner.

Secara umum, model ARMA(p,q) atau model ARIMA(p,0,q) dapat dinyatakan dengan rumus sebagai berikut:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.13)$$

atau dalam bentuk operator *backward shift* (B):

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) X_t = \mu + (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) e_t \quad (2.13)$$

Jika rata-rata $E(X_t) = E(X_{t-1}) = \dots = E(X_{t-p}) = \mu \neq 0$, maka model ARMA(p,q) dapat ditulis sebagai berikut ini:

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \dots + \phi_p (X_{t-p} - \mu) + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (2.14)$$

Apabila data masih nonstasioner dan perlu dilakukan proses *differencing* (pembedaan), maka model dapat diperluas menjadi ARIMA(p,d,q) dengan rumus:

$$\begin{aligned} (1 - B)^d((1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)X_t(1 - B) \\ = c + (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)e_t \end{aligned} \quad (2.15)$$

Dimana:

- c : Konstanta
- ϕ_p : Parameter *autoregressive* pada lag ke-p
- θ_p : Parameter *moving average* pada lag ke-p
- X_t : Variabel dependen (observasi saat ini)
- $X_{t-1}, X_{t-2}, \dots, X_{t-q}$: Observasi sebelumnya
- e_t : Nilai kesalahan (error term) pada waktu t
- $e_{t-1}, e_{t-2}, \dots, e_{t-k}$: Nilai kesalahan pada waktu sebelumnya
- B : Operator shift mundur (*backward shift*), di mana $BX_t = X_{t-1}$
- d : Orde differencing (jumlah kali data perlu dibuat stasioner)

Model ARMA dan ARIMA sangat berguna dalam pemodelan data deret waktu yang menunjukkan pola ketergantungan dengan periode sebelumnya, terutama dalam bidang ekonomi dan keuangan.

2.3.1 Model ARCH-GARCH

Heteroskedastisitas terjadi ketika variansi dari error (kesalahan) dalam suatu model regresi tidak konstan untuk setiap observasi, sehingga melanggar asumsi dasar bahwa $VAR(E_t) = \sigma^2$. Dengan kata lain, dalam kondisi heteroskedastisitas, sebaran kesalahan tidak memiliki distribusi yang seragam sepanjang rentang data yang diamati. Ketika suatu model mengalami masalah heteroskedastisitas, dampaknya adalah meskipun estimator yang diperoleh tetap konsisten, namun tidak lagi efisien. Hal ini disebabkan oleh keberadaan estimator lain yang memiliki variansi lebih kecil dibandingkan dengan estimator yang berasal dari model yang mengandung heteroskedastisitas. Akibatnya, interpretasi dan peramalan yang dihasilkan oleh model tersebut menjadi kurang optimal (Dana, 2016).

Dalam analisis data runtun waktu, terutama dalam bidang keuangan, masalah variansi error yang tidak konstan dapat diatasi dengan menggunakan model *Autoregressive Conditional Heteroskedasticity* (ARCH). Model ini kemudian dikembangkan lebih lanjut menjadi *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH) untuk meningkatkan fleksibilitas dalam menangani volatilitas yang bersifat dinamis. Model ARCH dan GARCH memungkinkan variansi error pada suatu periode bergantung pada variansi error di periode sebelumnya. Pendekatan ini sangat berguna dalam analisis volatilitas di pasar keuangan, di mana fluktuasi harga aset sering kali menunjukkan pola volatilitas yang berubah-ubah seiring waktu (Francq Zakoian, 2019).

2.3.2.1 Model ARCH (*Autoregressive Conditional Heteroscedasticity*)

Menurut PETRICĂ dan Stancu (2017) Model *Autoregressive Conditional Heteroskedasticity* (ARCH) pertama kali diperkenalkan oleh Robert F. Engle pada tahun 1982 sebagai metode untuk menangani volatilitas yang berubah-ubah dalam data runtun waktu, terutama dalam analisis finansial. Dalam model ARCH, variansi dari error (σ_t^2) tidak tetap, melainkan bergantung pada kuadrat error dari periode sebelumnya (σ_{t-1}^2). Hal ini berarti bahwa fluktuasi variansi dalam suatu periode memiliki ketergantungan terhadap volatilitas yang terjadi di masa lalu.

Inti utama dari model ARCH adalah bahwa error dari return aset tidak berkorelasi secara parsial, tetapi tetap memiliki ketergantungan dinamis, yang dapat dijelaskan melalui fungsi kuadratik sederhana. Model ARCH diklasifikasikan sebagai model yang menangkap pola variansi bersyarat dan sering kali digunakan bersama dengan model mean terbaik, yang diestimasi secara simultan untuk memperoleh parameter yang optimal. Model mean yang digunakan dalam kombinasi dengan ARCH dapat berupa model-model ARIMA untuk menangkap pola nilai tengah (*mean process*) dari data yang diamati.

Pada model ARCH dengan ordo p (ARCH(p)), asumsi yang digunakan adalah sebagai berikut:

$$E_t = \sigma_t x_t \quad (2.16)$$

$$\sigma_t^2 = a_0 + a_1 \sigma_{t-1}^2 + a_2 \sigma_{t-2}^2 + \cdots + ap E_{t-p}^2 \quad (2.17)$$

dengan ketentuan:

- $X_t \sim i.i.d. N(0,1)$ yang berarti bahwa X_t adalah variabel independen dan identik terdistribusi secara normal standar.
- $\alpha_0 > 0$ untuk menjamin variansi positif.
- $\alpha_i \geq 0$ untuk $i > 0$, guna memastikan bahwa variansi tidak bernilai negatif.

Dalam praktiknya, nilai X_t sering kali diasumsikan mengikuti distribusi normal normal ($N(0,1)$), sehingga model ARCH dapat ditulis dalam bentuk berikut:

$$E_t = \sigma_t X_t \quad (2.18)$$

Dengan σ_t^2 digunakan untuk menotasikan variansi bersyarat dari error pada waktu t . Model variansi yang memenuhi persamaan ARCH(p) adalah model yang menggambarkan hubungan antara variansi error pada waktu t dengan kuadrat error dari periode sebelumnya. Model ini sangat berguna dalam analisis volatilitas di pasar keuangan, di mana perubahan variabilitas harga aset cenderung memiliki efek berantai dari satu periode ke periode lainnya.

2.3.2.2 Model GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*)

Model *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH) dikembangkan oleh Tim Bollerslev pada tahun 1986 sebagai penyempurnaan dari model *Autoregressive Conditional Heteroskedasticity* (ARCH) yang diperkenalkan oleh Engle pada tahun 1982. Model GARCH dirancang untuk mengatasi keterbatasan model ARCH yang sering kali membutuhkan orde yang sangat tinggi agar dapat menangkap pola volatilitas yang kompleks. Dengan menerapkan prinsip parsimony, yaitu memilih model yang lebih sederhana namun tetap efektif, model GARCH memungkinkan analisis volatilitas dengan jumlah parameter yang lebih sedikit dibandingkan ARCH, sehingga meningkatkan efisiensi dalam estimasi dan memastikan bahwa variansi tetap positif.

Menurut Bollerslev (1987), suatu variabel residual ϵ_t yang diperoleh dari model regresi:

$$\epsilon_t = X_t - \mu_t \quad (2.19)$$

mengikuti model GARCH(P,Q) jika variansi bersyaratnya σ^2 memenuhi persamaan berikut:

$$\sigma_t^2 = a_0 + a_1\epsilon_{t-1}^2 + a_2\epsilon_{t-2}^2 + \dots + a_p\epsilon_{t-p}^2 + \beta_1\sigma_{t-1}^2 + \beta_2\sigma_{t-2}^2 + \dots + \beta_q\sigma_{t-q}^2 \quad (2.20)$$

Atau dalam bentuk notasi sigma:

$$\sigma_t^2 = a_0 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 \quad (2.21)$$

Dengan

$$\epsilon_t = \sigma_t X_t \quad (2.22)$$

Dimana:

- σ_t^2 = variansi bersyarat dari residual pada waktu t.
- a_0 = komponen konstanta yang harus lebih besar dari nol ($\alpha_0 > 0$).
- a_i = parameter ARCH yang menunjukkan pengaruh kuadrat residual sebelumnya terhadap variansi bersyarat.
- ϵ_{t-i}^2 = kuadrat dari residual pada waktu t-i.
- β_j = parameter GARCH yang menunjukkan pengaruh variansi bersyarat sebelumnya terhadap variansi saat ini.
- σ_{t-j}^2 = variansi bersyarat dari residual pada waktu t-j.

Dalam model ini, diasumsikan bahwa:

$$X_t \sim i. i. d. N(0,1) \quad (2.23)$$

dengan syarat:

- $\alpha_0 > 0$ untuk memastikan bahwa variansi tetap positif.

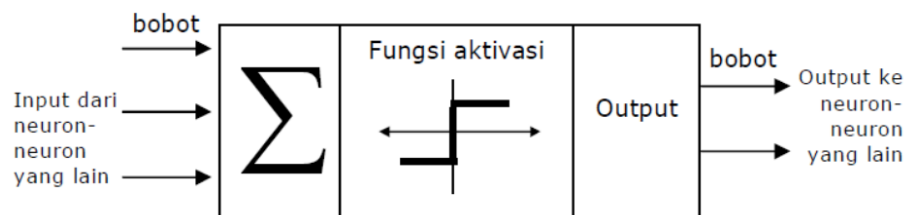
- $\alpha_i \geq 0$ untuk $i = 1, 2, \dots, p$ agar variansi tidak bernilai negatif.
- $\beta_j \geq 0$ untuk $j = 1, 2, \dots, q$
- $0 < \alpha_i + \beta_j < 1$ untuk menjamin stasioneritas variansi.

Persamaan GARCH (P,Q) ini menghubungkan variansi residual pada waktu ke-t dengan kombinasi kuadrat residual dari periode sebelumnya dan variansi bersyarat dari periode sebelumnya. Dengan menggunakan model ini, volatilitas data finansial dapat dimodelkan dengan lebih baik tanpa perlu menggunakan model ARCH dengan orde yang terlalu tinggi.

2.4 Landasan Teori LSTM

2.4.1 Artificial Neural Network (ANN)

Abiodun et al. (2018) berpendapat Syaraf Tiruan (*Artificial Neural Network* - ANN) merupakan suatu paradigma dalam pemrosesan informasi yang terinspirasi dari mekanisme kerja sistem saraf biologis, khususnya otak manusia, dalam mengolah dan menganalisis data. Paradigma ini mengusung pendekatan baru dalam sistem pemrosesan informasi yang terdiri dari sejumlah besar elemen pemrosesan (neuron) yang saling terhubung dan berinteraksi untuk menyelesaikan berbagai permasalahan tertentu.



Gambar 2.2. Struktur *neuron* Jaringan Syaraf Tiruan, Widagda (2016)

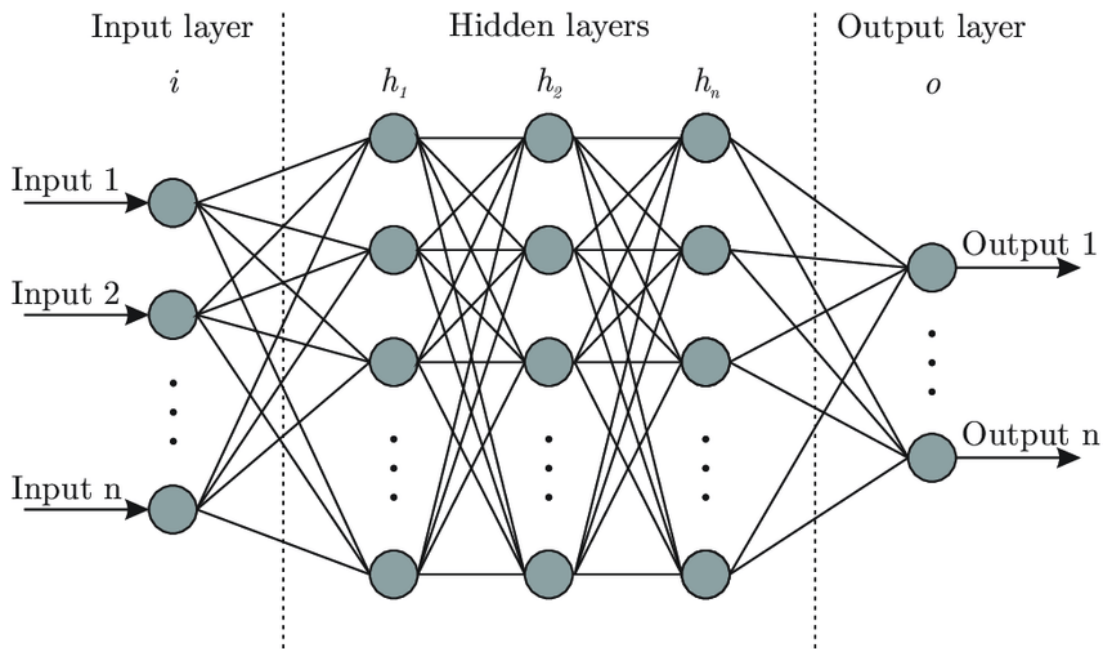
Berdasarkan Gambar 2.2, komponen utama dalam ANN dapat dijelaskan sebagai berikut:

1. Input: Berperan serupa dengan dendrit pada sistem saraf manusia, yang bertugas menerima informasi dari neuron lain sebagai masukan awal dalam proses pengolahan data.
2. Neuron: Merupakan unit pemrosesan utama dalam ANN yang bertanggung jawab dalam mengolah informasi yang diterima.

3. Bobot (*Weight*): Memiliki fungsi yang mirip dengan sinapsis dalam otak manusia, yaitu menentukan seberapa kuat hubungan antar-neuron dalam jaringan.
4. Fungsi Aktivasi: Berperan dalam memetakan hasil penjumlahan input yang diterima oleh suatu neuron ke dalam suatu nilai tertentu yang digunakan untuk menentukan apakah sinyal akan diteruskan ke tahap selanjutnya.
5. Output: Memiliki peran yang menyerupai akson pada otak manusia, yakni menghasilkan keluaran dari proses pemrosesan informasi dalam jaringan, baik sebagai hasil akhir maupun sebagai masukan bagi neuron lain dalam sistem.

Pengembangan suatu model ANN sangat bergantung pada proses pelatihan atau training, yang bertujuan untuk menentukan bobot optimal dalam jaringan. Proses ini umumnya dilakukan melalui dua mekanisme utama, yaitu *supervised training* dan *unsupervised training* (Dike et al. (2018). Pada *supervised training*, proses pembelajaran memerlukan bimbingan eksternal dengan menggunakan pasangan data input-output yang telah diketahui kebenarannya. Output yang dihasilkan oleh jaringan dibandingkan dengan output yang diharapkan, dan selisihnya digunakan untuk memperbarui bobot sehingga jaringan dapat menghasilkan output yang lebih akurat. Salah satu algoritma yang paling umum digunakan dalam *supervised training* adalah *backpropagation*, di mana bobot dalam jaringan diperbarui secara bertahap berdasarkan perbedaan antara output yang dihasilkan dan output yang seharusnya (Fatimah et al., 2018).

Sebaliknya, dalam *unsupervised training*, jaringan belajar secara mandiri tanpa adanya supervisi eksternal. ANN akan mencari pola atau struktur tersembunyi dalam data tanpa referensi output yang benar. Metode ini banyak digunakan dalam analisis kluster dan reduksi dimensi, di mana sistem harus mengelompokkan data berdasarkan karakteristik yang ditemukan.



Gambar 2.3. Arsitektur Artificial neural network (ANN), Bre, Gimenez dan Fachinotti (2018)

Struktur ANN terdiri dari tiga lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* merupakan bagian awal dalam jaringan yang bertugas menerima masukan dari lingkungan luar. Masukan ini dapat berupa representasi dari suatu masalah atau data mentah yang akan dianalisis lebih lanjut oleh jaringan. Selanjutnya, informasi yang diterima akan diteruskan ke *hidden layer*, yang berisi neuron-neuron yang bertugas untuk mengolah dan mentransformasikan informasi dengan cara mengalikan setiap input dengan bobot tertentu dan menjumlahkan hasilnya. Semakin kompleks masalah yang ingin diselesaikan, semakin banyak *hidden layer* yang diperlukan. Setelah proses pada *hidden layer* selesai, hasilnya akan dikirim ke *output layer*, yang akan menghasilkan keluaran akhir berdasarkan pemrosesan sebelumnya. Lapisan ini terdiri dari neuron-neuron yang bertanggung jawab untuk menyampaikan hasil pemrosesan kepada pengguna atau sistem lain yang membutuhkan (Bre, Gimenez dan Fachinotti, 2018).

Dalam proses pengolahan data, ANN menggunakan mekanisme perhitungan berbasis bobot dan nilai ambang (*threshold*). Setiap input yang diterima oleh *input layer* akan dikalikan dengan bobot tertentu, lalu dijumlahkan dalam hidden layer. Jika hasil perhitungan melebihi nilai ambang yang telah ditentukan, maka informasi akan diteruskan ke lapisan berikutnya hingga mencapai *output layer*. Sebaliknya, jika hasil perhitungan tidak memenuhi ambang batas, informasi tidak akan diteruskan lebih lanjut.

Meskipun ANN sangat efektif dalam pemrosesan data dan analisis pola, salah satu keterbatasannya adalah ketidakmampuannya dalam mengingat informasi yang telah diproses sebelumnya. Hal ini disebabkan karena setiap input baru diproses secara independen, tanpa mempertimbangkan data yang telah diterima sebelumnya. Untuk mengatasi keterbatasan ini, dikembangkanlah *Recurrent Neural Networks* (RNN), yang memiliki arsitektur jaringan dengan mekanisme loop di dalamnya. Loop ini memungkinkan informasi dari satu waktu ke waktu berikutnya tetap tersimpan dalam jaringan, sehingga RNN mampu mengenali pola data sekuensial dan mempertahankan konteks dari informasi sebelumnya. Oleh karena itu, RNN sering digunakan dalam berbagai aplikasi yang melibatkan data berbasis waktu, seperti pemrosesan bahasa alami (*natural language processing*), analisis deret waktu (*time series analysis*), dan pengenalan suara (Kaur dan Mohta, 2019).

2.4.1 Recurrent Neural Network (RNN)

Menurut Sherstinsky (2020) *Recurrent Neural Network* (RNN) merupakan salah satu jenis *Artificial Neural Network* yang dirancang khusus untuk menangani data sekuensial, di mana proses pembelajarannya bergantung pada informasi dari masa lalu. Kemampuan ini memungkinkan RNN untuk memproses dan menganalisis data deret waktu (*time series*) dengan lebih efektif. Struktur dasar RNN terdiri dari tiga komponen utama, yaitu *input layer*, *hidden layer*, dan *output layer*.

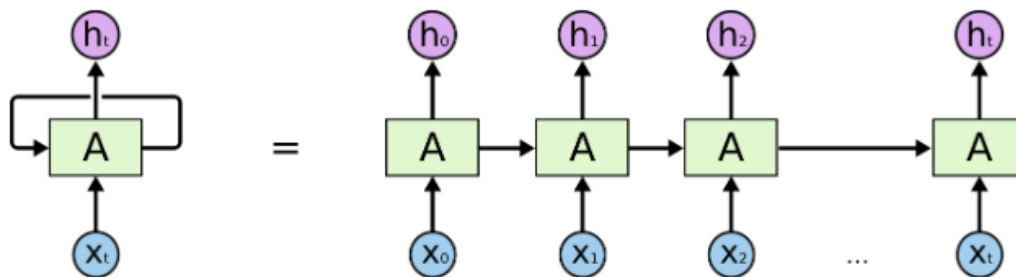
Dalam aliran informasi RNN, data mengalir secara searah dari input layer ke hidden layer, dan dari hidden layer sebelumnya menuju hidden layer berikutnya. Dengan mekanisme ini, output dari hidden layer sebelumnya akan digunakan sebagai input tambahan bagi hidden layer selanjutnya, sehingga memungkinkan model untuk mempertahankan dan memproses informasi historis dalam suatu urutan (Pascanu, 2013).

Ketika melakukan prediksi, RNN tidak hanya mempertimbangkan data masukan (*input*) pada waktu saat ini tetapi juga memperhitungkan informasi dari *input* yang telah diproses sebelumnya. Hubungan antara input-input ini membantu dalam menyebarkan informasi ke seluruh *hidden layer*, menciptakan semacam memori internal yang merekam hasil perhitungan dari langkah-langkah sebelumnya.

Secara konseptual dan berdasarkan arsitekturnya, RNN seharusnya mampu menangani ketergantungan jangka panjang dalam data sekuensial. Namun, dalam praktiknya, RNN memiliki

keterbatasan dalam menyimpan informasi dalam jangka waktu yang panjang. Kelemahan ini disebabkan oleh masalah *vanishing gradient*, di mana bobot dalam jaringan mengalami penyusutan secara eksponensial saat melakukan pembaruan dalam proses *backpropagation*. Akibatnya, model kesulitan mengingat informasi dari langkah-langkah sebelumnya yang jauh.

Untuk mengatasi permasalahan ini, Hochreiter & Schmidhuber pada tahun 1997 mengembangkan jenis RNN yang lebih canggih, yaitu *Long Short-Term Memory* (LSTM). LSTM dirancang dengan mekanisme khusus yang memungkinkan model mempertahankan informasi lebih lama dan mengelola ketergantungan jangka panjang secara lebih efektif dibandingkan dengan standar RNN (Sherstinsky, 2020).

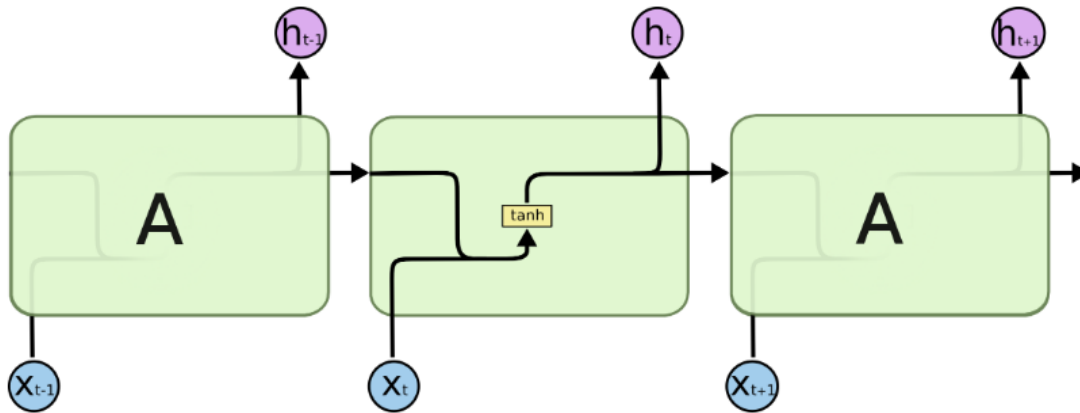


Gambar 2.4. Recurrent Neural Network (RNN), Olah (2016)

2.4.1 Long Short-Term Memory (LSTM)

Menurut Staudemeyer dan Morris (2019) *Long Short-Term Memory* (LSTM) merupakan pengembangan dari arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi kelemahan utama RNN dalam menangani ketergantungan jangka panjang pada data sekuensial. Model ini pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 sebagai solusi terhadap masalah menghilangnya gradien yang sering terjadi dalam algoritma *backpropagation* pada RNN konvensional.

Secara struktural, LSTM memiliki kesamaan dengan RNN karena keduanya terdiri atas tiga lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*. Namun, perbedaan signifikan terletak pada arsitektur di dalam lapisan tersembunyi. Pada RNN, *hidden layer* hanya terdiri dari satu lapisan sederhana dengan fungsi aktivasi tangens hiperbolik (*tanh*) (Graves dan Graves (2012)). Fungsi *tanh* ini didefinisikan sebagai:



Gambar 2.5. Perulangan pada RNN dengan satu layer, Olah (2016)

Gambar 2.5 di atas menunjukkan proses perulangan dalam RNN yang hanya memiliki satu lapisan, yaitu lapisan *tanh*, yang akan dijelaskan lebih lanjut dalam persamaan berikut ini:

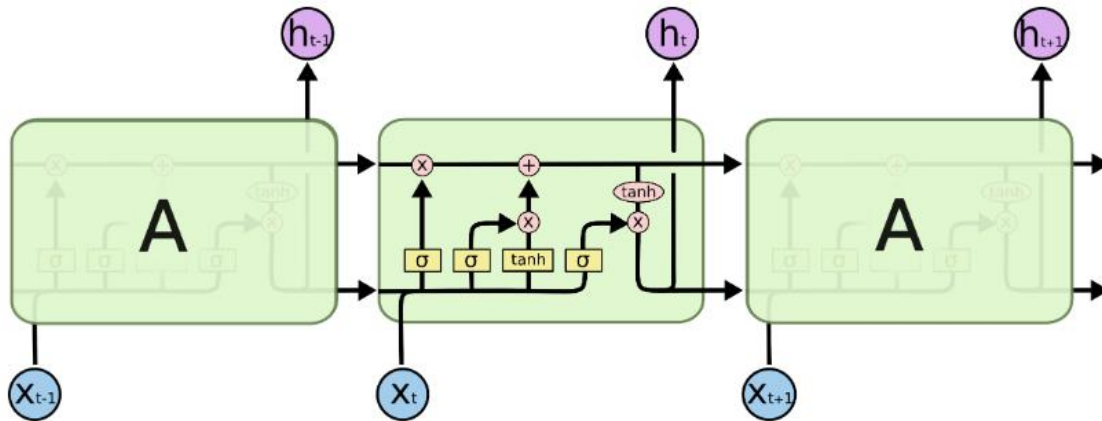
$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.24)$$

dimana:

σ = fungsi sigmoid

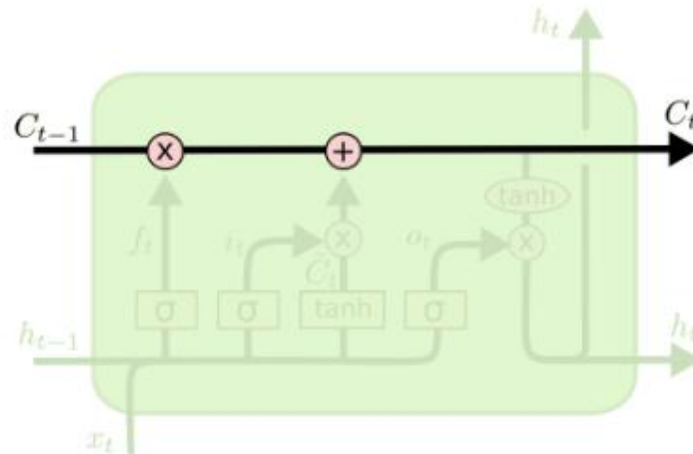
x = data input

Sebaliknya, LSTM mengadopsi arsitektur yang lebih kompleks dengan memasukkan elemen tambahan berupa *memory cell* dan tiga jenis gerbang (*gates*), yaitu *forget gate*, *input gate*, dan *output gate* (Hochreiter,1997). Kombinasi dari elemen-elemen ini memungkinkan LSTM untuk secara selektif mempertahankan atau membuang informasi dalam jangka panjang, sehingga mengatasi permasalahan pada RNN konvensional pada Gambar 2.6.



Gambar 2.6. Perulangan pada LSTM, Olah (2016)

Olah (2015) telah membuat Langkah-langkah dalam LSTM, seperti yang ditunjukkan pada Gambar 2.6, akan dijelaskan secara rinci dalam bagian ini. Komponen utama dalam metode LSTM adalah *cell state*, yang berperan sebagai jalur informasi utama. *Cell state* direpresentasikan sebagai garis horizontal di bagian atas diagram yang menghubungkan seluruh lapisan output dalam LSTM, sebagaimana ditampilkan pada Gambar 2.7.



Gambar 2.7. Cell State pada LSTM, Olah (2016)

LSTM memiliki beberapa *gates* yang terdiri dari lapisan sigmoid dan operasi *pointwise*. *Gates* ini berfungsi untuk menambahkan atau menghapus informasi, menentukan apakah suatu informasi akan diteruskan atau dihentikan. Lapisan sigmoid berperan dalam mengonversi output ke dalam rentang $[0,1]$, di mana nilai 0 menunjukkan bahwa informasi akan dihentikan, sedangkan nilai 1 menandakan bahwa informasi akan diteruskan. Persamaan sigmoid dijelaskan sebagai berikut:

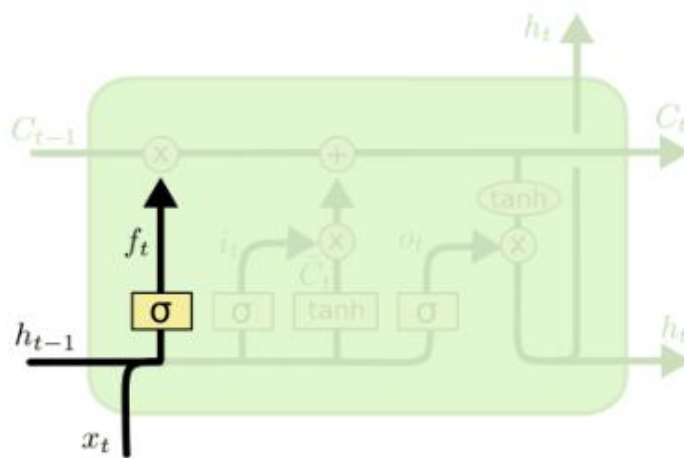
$$\sigma(x) = \frac{1}{(1 + e^{-x})} \quad (2.25)$$

dimana:

- $x = \text{input data}$
- $e = \text{konstanta matematika dengan nilai } 2,71828 \ 18284 \ 59045 \ 23536 \ 02874 \ 71352 \dots$

Proses kerja metode LSTM dapat dijelaskan sebagai berikut:

Langkah pertama dimulai pada *forget gate*, yang berfungsi menentukan informasi mana yang perlu dihapus dari *cell state*. Keputusan ini dibuat melalui lapisan sigmoid, yang menghasilkan nilai antara 0 dan 1. Perhitungan pada *forget gate* dilakukan dengan mempertimbangkan output sebelumnya, h_{t-1} , serta data masukan saat ini, x_t , sebagaimana ditunjukkan pada Gambar 2.8.



Gambar 2.8. Alur *Forget Gate* pada LSTM, Olah (2016)

Persamaan yang digunakan dalam *forget gate* dijelaskan sebagai berikut:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.26)$$

dimana:

$f_t = \text{forget gate}$

$\sigma = \text{fungsi sigmoid}$

W_f = bobot (*weight*) untuk *forget gate*

h_{t-1} = output pada langkah sebelumnya (t-1)

x_t = input pada langkah ke-t

b_f = nilai bias pada *forget gate*

Bobot (*weight*) dalam *forget gate* dirumuskan dengan persamaan berikut:

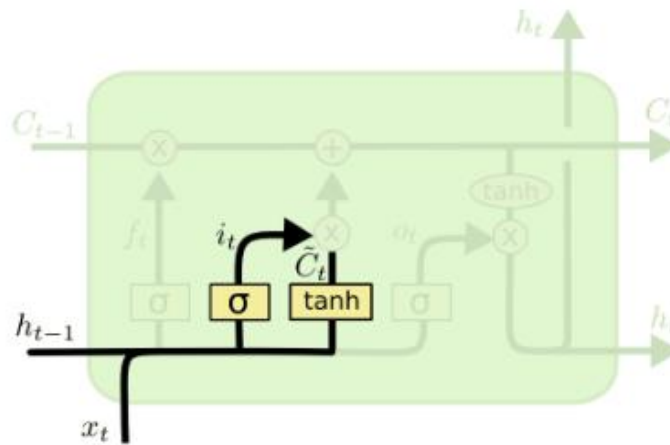
$$W = \left(-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}} \right) \quad (2.27)$$

Dimana:

W = Bobot (*weight*)

d = jumlah variable

Tahap ini diilustrasikan pada Gambar 2.6.



Gambar 2.9. Alur input gate pada LSTM, Olah (2016)

Persamaan yang digunakan dalam *input gate* dirumuskan sebagai berikut:

$$i_t = \sigma (W_i [h_{t-1}, x_t] + b_i) \quad (2.28)$$

dimana:

i_t = *input gate*

σ = fungsi sigmoid

W_i = bobot (*weight*) untuk *input gate*

h_{t-1} = output pada langkah sebelumnya (t-1)

x_t = input pada langkah ke-t

b_i = nilai bias pada *input gate*

Persamaan untuk kandidat nilai baru dirumuskan sebagai berikut:

$$\tilde{C}_t = \tanh(W_c [h_{t-1}, X_t] + b_c) \quad (2.29)$$

dimana:

$\tilde{C}_t$ = nilai baru yang akan ditambahkan ke *cell state*

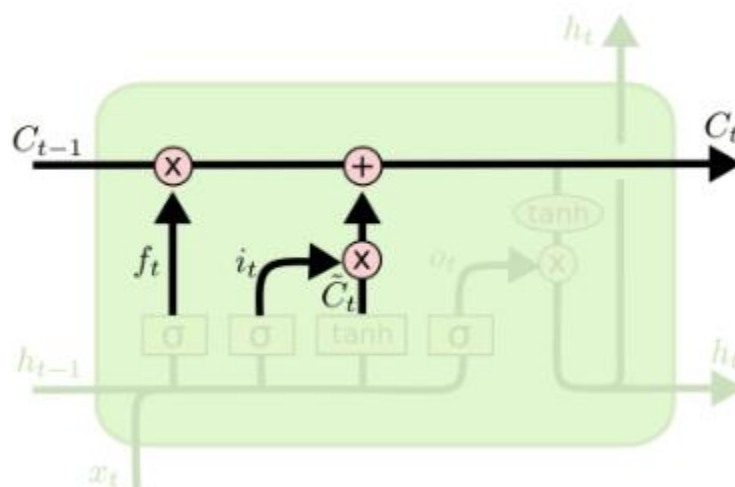
$\tanh$ = fungsi aktivasi *tanh*

W_c = bobot (*weight*) untuk *cell state*

h_{t-1} = output pada langkah sebelumnya (t-1)

X_t = input pada langkah ke-t

b_c = nilai bias pada *cell state*



Gambar 2.10. Alur Pembaruan *Cell State* pada LSTM, Olah (2016)

Langkah ketiga dalam LSTM adalah memperbarui *cell state*. Proses pembaruan ini digambarkan dalam Gambar 2.7. Berdasarkan ilustrasi tersebut, *cell state* diperbarui dengan mengalikan *forget gate* dengan *cell state* sebelumnya, lalu menjumlahkannya dengan hasil perkalian antara *input gate* dan kandidat nilai baru $i_t * \tilde{C}_t$. Persamaan untuk pembaruan *cell state* dirumuskan sebagai berikut:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.30)$$

dimana:

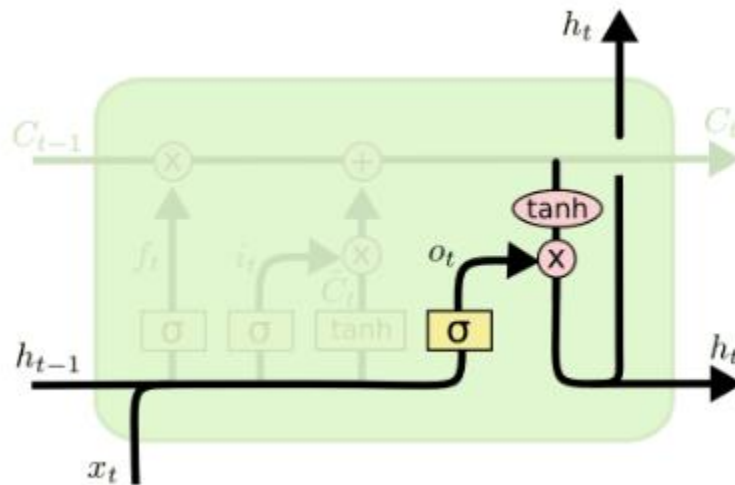
C_t = cell state saat ini

f_t = forget gate

C_{t-1} = *cell state* pada langkah sebelumnya (t-1)

i_t = input gate

$\tilde{C}_t$ = nilai baru yang akan ditambahkan ke cell state



Gambar 2.11. Alur output pada LSTM, Olah (2016)

Langkah terakhir dalam metode LSTM adalah menentukan output, sebagaimana diilustrasikan dalam Gambar 2.11. Pada tahap ini, lapisan sigmoid berperan dalam memilih bagian dari *cell state* yang akan digunakan sebagai output.

Persamaan untuk *output gate* dirumuskan sebagai berikut:

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \quad (2.31)$$

dimana:

o_t = *output gate*

σ = fungsi sigmoid

W_o = bobot (weight) untuk *output gate*

h_{t-1} = *output* pada langkah sebelumnya (t-1)

x_t = input pada langkah ke-t

b_o = nilai bias pada *output gate*

Setelah nilai dari *output gate* diperoleh, *cell state* kemudian diproses melalui fungsi aktivasi *tanh* untuk mengubahnya ke dalam rentang nilai antara -1 hingga 1. Selanjutnya, hasil tersebut dikalikan dengan *output gate* yang diperoleh dari lapisan sigmoid.

Persamaan untuk nilai output pada langkah ke-t dirumuskan sebagai berikut:

$$h_t = o_t * \tanh (C_t) \quad (2.32)$$

dimana:

h_t = nilai *output* pada langkah ke-t

o_t = *output gate*

$\tanh$ = fungsi aktivasi *tanh*

C_t = *cell state*

2.4.1 Optimizer Adaptive Moment Estimation Optimization (Adam)

Optimasi dalam pembelajaran mesin bertujuan untuk menemukan bobot yang optimal dengan cara meminimalkan kesalahan (*error*) serta memaksimalkan akurasi model. Selama proses pelatihan, bobot model diperbarui secara bertahap untuk mengurangi nilai fungsi error sehingga model dapat

melakukan prediksi dengan tingkat akurasi yang tinggi. Namun, menentukan strategi yang tepat terkait bagaimana, kapan, dan sejauh mana perubahan bobot tersebut harus dilakukan masih menjadi tantangan tersendiri. Oleh karena itu, algoritma optimasi digunakan untuk mengintegrasikan fungsi error dengan parameter model serta memperbarui bobot berdasarkan umpan balik dari fungsi *loss* (kerugian). Dengan kata lain, algoritma optimasi berfungsi untuk menyempurnakan model dengan menyesuaikan bobot sehingga model dapat mencapai performa terbaiknya (Yi, Ahn dan Ji, 2020).

Salah satu algoritma optimasi yang digunakan dalam penelitian ini adalah *Adaptive Moment Estimation* (Adam), yang merupakan metode populer dalam bidang *deep learning* karena kemampuannya dalam mencapai hasil yang optimal secara cepat. Adam merupakan pengembangan dari dua algoritma optimasi sebelumnya, yaitu *Adaptive Gradient Algorithm* (AdaGrad) dan *Root Mean Square Propagation* (RMSProp). Adam mengadaptasi *learning rate* secara dinamis berdasarkan rata-rata momentum pertama (mean) seperti pada RMSProp, sekaligus memanfaatkan rata-rata momentum kedua (varians tidak terpusat) dari gradien, sehingga mampu memberikan optimasi yang lebih stabil (Varshney dan Sharma, 2024).

Algoritma Adam bekerja dengan menghitung rata-rata pergerakan eksponensial dari gradien dan gradien kuadratnya, di mana parameter β_1 dan β_2 digunakan untuk mengontrol tingkat peluruhan rata-rata pergerakan tersebut. Parameter ini berperan dalam menentukan seberapa besar pengaruh gradien historis terhadap pembaruan bobot saat ini. Menurut penelitian yang dilakukan oleh Aldi, Jondri, dan Aditsania (2018), Adam menawarkan berbagai keunggulan dibandingkan metode optimasi lainnya, di antaranya:

- Pembaruan parameter tidak secara langsung bergantung pada skala gradien, melainkan dikendalikan oleh *hyperparameter step size*, sehingga memberikan kestabilan dalam proses optimasi.
- Adam dapat bekerja dengan baik pada data yang tidak memiliki distribusi stasioner, sehingga lebih fleksibel dalam berbagai jenis dataset.
- Metode ini tidak memerlukan penyesuaian *hyperparameter* secara ekstensif karena nilai default yang direkomendasikan telah terbukti cukup efektif.

Dalam implementasinya, nilai parameter yang umum digunakan dalam Adam adalah $\beta_1 = 0.9, \beta_2 = 0.999$, dan $\epsilon = 10^{-8}$, di mana β_1 dan β_2 merepresentasikan tingkat penurunan

eksponensial dari rata-rata pergerakan gradien, sedangkan ϵ berfungsi sebagai faktor stabilisasi untuk mencegah pembagian oleh nol dalam proses pembaruan parameter. Dengan mekanisme ini, Adam mampu memberikan hasil optimasi yang lebih cepat dan stabil dibandingkan algoritma lain, sehingga menjadi pilihan utama dalam berbagai aplikasi deep learning.

2.4.1 Normalisasi

Untuk mengurangi kesalahan dalam analisis data, diperlukan proses normalisasi. Menurut Sinsomboonthong (2022) Normalisasi berperan dalam mencegah anomali data serta menjaga konsistensi pola dalam dataset. Selain itu, normalisasi bertujuan untuk menyesuaikan skala data agar lebih kecil tanpa mengubah nilai atau informasi yang terkandung di dalamnya. Salah satu metode normalisasi yang umum digunakan adalah *Min-Max Scaling* (Mondal et al, 2024). Metode ini membantu mengatasi perbedaan skala yang signifikan antara nilai-nilai dalam dataset, sehingga hasil analisis menjadi lebih akurat dan tidak terpengaruh oleh skala variabel yang berbeda.

Menurut Patro (2015) Teknik *Min-Max Scaling* bekerja dengan mengubah nilai data asli ke dalam rentang skala antara 0 dan 1 tanpa kehilangan informasi penting. Transformasi data ini dilakukan menggunakan persamaan berikut:

$$X' = \frac{(x - X_{min})}{(X_{max} - X_{min})} \quad (2.33)$$

dimana:

- X' = Data hasil normalisasi
- x = Data asli
- X_{min} = nilai minimum dari data x
- X_{max} = nilai maximum dari data x

Dengan penerapan normalisasi Min-Max Scaling, distribusi data menjadi lebih seragam, sehingga dapat meningkatkan akurasi dalam berbagai metode pemodelan dan analisis data.

2.4.1 Denormalisasi

Menurut Wibawa et al. (2021) Denormalisasi merupakan suatu proses untuk mengembalikan data yang sebelumnya telah dinormalisasi ke dalam bentuk aslinya atau nilai sebenarnya. Proses ini bertujuan agar hasil prediksi dapat dievaluasi dengan lebih akurat melalui perbandingan langsung dengan data aktual. Denormalisasi sangat penting dalam berbagai analisis data, terutama dalam model prediktif, guna memastikan bahwa nilai yang diperoleh tetap sesuai dengan skala data awal.

Pada kasus normalisasi dalam rentang $[0,1]$, denormalisasi dapat dilakukan menggunakan rumus berikut:

$$x = X_t (X_{max} - X_{min}) + X_{min} \quad (2.34)$$

dimana:

- X_t adalah nilai data yang telah dinormalisasi
- x merupakan hasil output setelah dikembalikan ke skala awal
- X_{min} menunjukkan nilai minimum dari keseluruhan data aktual
- X_{max} menunjukkan nilai maksimum dari keseluruhan data aktual

Dengan menerapkan proses denormalisasi ini, nilai prediksi dapat dikonversi kembali ke satuan aslinya sehingga dapat dibandingkan langsung dengan data nyata untuk mengevaluasi kinerja model.

2.5 Kriteria Pemilihan Model Terbaik

Dalam suatu analisis deret waktu, sering kali terdapat beberapa model yang dapat digunakan untuk merepresentasikan pola data yang diamati. Oleh karena itu, diperlukan suatu prosedur pemilihan model terbaik yang dapat memberikan hasil yang paling sesuai dengan karakteristik data. Pemilihan model yang optimal dilakukan dengan mempertimbangkan berbagai kriteria yang berhubungan dengan akurasi estimasi dan kesalahan prediksi.

Model terbaik dipilih berdasarkan evaluasi terhadap residual model, yaitu selisih antara nilai aktual dan nilai yang diprediksi oleh model. Residual yang memenuhi asumsi statistik yang baik menunjukkan bahwa model memiliki kemampuan yang lebih akurat dalam menangkap pola data yang sebenarnya. Selain itu, pemilihan model juga dapat didasarkan pada seberapa kecil tingkat kesalahan dalam peramalan (*forecasting error*), karena model yang lebih baik akan menghasilkan prediksi yang lebih akurat di masa depan.

Berbagai kriteria statistik digunakan dalam pemilihan model terbaik, antara lain:

2.5.1 Mean Squared Error (MSE)

Menurut Chicco, Warrens dan Jurman (2021) *Mean Squared Error* (MSE) merupakan metrik yang digunakan untuk mengukur rata-rata kesalahan kuadrat antara nilai aktual dan nilai prediksi. MSE dihitung dengan menentukan selisih antara data aktual dan data prediksi, kemudian dikuadratkan selisih tersebut, menjumlahkan keseluruhan nilai yang telah dikuadratkan, dan membaginya dengan jumlah periode peramalan yang terlibat. Persamaan MSE dapat dirumuskan sebagai berikut:

$$MSE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n} \quad (2.35)$$

dimana:

- Y_i = nilai aktual
- $\hat{Y}_t$ = nilai prediksi
- n = jumlah observasi
- $\sum_{t=1}^n (Y_t - \hat{Y}_t)^2$ = jumlah kuadrat kesalahan (*Sum of Squared Errors*, SSE)

MSE memiliki keunggulan dalam memberikan bobot lebih besar pada kesalahan yang lebih besar karena kesalahan dikuadratkan sebelum dihitung rata-rata. Oleh karena itu, metrik ini sering digunakan dalam evaluasi model prediktif untuk menilai sejauh mana hasil prediksi menyimpang dari data aktual. Keunggulan lainnya adalah bahwa MSE mempertimbangkan semua kesalahan tanpa memperhitungkan arah kesalahan tersebut, yang berarti kesalahan positif

maupun negatif akan memiliki dampak yang sama dalam perhitungan. Hal ini sangat berguna dalam model prediksi yang memerlukan akurasi tinggi.

Namun, MSE juga memiliki kelemahan. Karena kesalahan dikuadratkan, metrik ini lebih sensitif terhadap outlier dibandingkan dengan metrik lain seperti Mean Absolute Error (MAE). Jika terdapat nilai kesalahan yang sangat besar, maka kontribusinya terhadap nilai akhir MSE menjadi lebih dominan, yang dapat menyebabkan overestimation terhadap tingkat kesalahan model.

Secara umum, semakin kecil nilai MSE, semakin baik performa model prediksi yang digunakan, karena menunjukkan bahwa kesalahan antara nilai aktual dan prediksi semakin kecil. Model dengan nilai MSE yang lebih rendah dianggap lebih akurat dibandingkan model dengan MSE yang lebih tinggi. Namun, dalam beberapa kasus, penurunan MSE yang terlalu besar juga perlu dianalisis lebih lanjut, karena bisa saja menunjukkan overfitting, di mana model terlalu menyesuaikan diri dengan data latih tetapi kurang mampu melakukan generalisasi terhadap data baru.

2.5.1 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) adalah metrik evaluasi dalam analisis peramalan (*forecasting*) yang mengukur persentase rata-rata kesalahan absolut antara nilai aktual dan nilai prediksi. MAPE sering digunakan untuk menilai keakuratan model dalam meramalkan data. Berikut ini merupakan Rumus dari MAPE:

$$MAPE = \frac{1}{T} \sum_{t=1}^T \left[\frac{Y_t - \hat{Y}_t}{Y_t} \right] \times 100\% \quad (2.36)$$

dimana

- Y_t = nilai aktual pada observasi ke-t
- $\hat{Y}_t$ = nilai prediksi pada observasi ke-t
- T = jumlah observasi

Interpretasi MAPE:

- $MAPE < 10\% \rightarrow$ Model memiliki akurasi yang sangat baik.
- $10\% \leq MAPE < 20\% \rightarrow$ Model memiliki akurasi yang baik.
- $20\% \leq MAPE < 50\% \rightarrow$ Model memiliki akurasi yang cukup.
- $MAPE \geq 50\% \rightarrow$ Model memiliki akurasi yang buruk.

MAPE adalah metrik yang dapat digunakan dalam analisis peramalan untuk mengukur seberapa besar kesalahan model dalam bentuk persentase. Semakin kecil nilai MAPE, semakin baik akurasi model peramalan.

2.5.1 Akaike Information Criterion (AIC)

Menurut Vivas, Allende-Cid dan Salas (2020) *Akaike Information Criterion* (AIC) adalah metode yang digunakan untuk memilih model statistik terbaik dengan mempertimbangkan keseimbangan antara akurasi dan kompleksitas model. AIC pertama kali diperkenalkan oleh Hirotugu Akaike pada tahun 1973 dan banyak digunakan dalam analisis regresi, deret waktu, dan model statistik lainnya. Berikut ini formula dari AIC:

$$AIC = 2k - 2\ln(L) \quad (2.37)$$

dimana:

k = jumlah parameter dalam model

L = likelihood maksimum dari model

Interpretasi AIC:

- Model dengan nilai AIC yang lebih kecil dianggap sebagai model yang lebih baik dibandingkan model dengan AIC yang lebih besar.
- AIC tidak memberikan informasi mutlak mengenai kualitas model, tetapi hanya digunakan untuk membandingkan beberapa model pada dataset yang sama.

AIC adalah alat yang berguna dalam pemilihan model statistik dengan mempertimbangkan keseimbangan antara kompleksitas dan akurasi, di mana model dengan AIC lebih kecil lebih disukai.

2.5.1 Root Mean Square Error (RMSE)

Menurut Hodson (2022) *Root Mean Square Error* (RMSE) merupakan suatu metrik yang digunakan untuk mengukur tingkat kesalahan antara nilai aktual dan nilai hasil prediksi. RMSE dihitung dengan cara menentukan selisih antara data aktual dan data prediksi, kemudian mengkuadratkan selisih tersebut, menjumlahkan keseluruhan nilai yang telah dikuadratkan, membaginya dengan jumlah periode peramalan yang terlibat, dan akhirnya mengambil akar kuadrat dari hasil pembagian tersebut.

RMSE memiliki keunggulan dalam menghindari penggunaan nilai absolut yang dapat menyebabkan distorsi dalam berbagai perhitungan matematis. Oleh karena itu, metrik ini sering digunakan dalam evaluasi model prediktif untuk menilai sejauh mana hasil prediksi menyimpang dari data aktual. Persamaan RMSE dapat dirumuskan sebagai berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (2.38)$$

dimana:

- Y_t merupakan nilai permintaan aktual pada periode ke-i
- $\hat{Y}_t$ adalah nilai hasil peramalan pada periode ke-i
- n menunjukkan jumlah periode peramalan yang digunakan dalam evaluasi

Dengan menggunakan RMSE, kesalahan prediksi dapat diinterpretasikan dalam satuan yang sama dengan data aslinya, sehingga mempermudah analisis terhadap kinerja model yang digunakan.

BAB 3

METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Dalam penelitian ini, peneliti menerapkan pendekatan kuantitatif deskriptif yang dikombinasikan dengan kajian literatur sebagai metode analisis. Pendekatan kuantitatif deskriptif digunakan untuk menggambarkan fenomena yang diamati berdasarkan data numerik yang dapat diukur secara objektif. Data yang dianalisis dalam penelitian ini adalah data harga penutupan Indeks Harga Saham Gabungan (IHSG), yang menjadi indikator utama dalam mengevaluasi pergerakan pasar modal.

Selain itu, kajian literatur digunakan sebagai metode penelitian yang mengandalkan berbagai sumber pustaka guna memperoleh landasan teori dan memperkuat argumen dalam pemecahan masalah yang dikaji. Melalui kajian literatur, penelitian ini dapat membandingkan, mengonfirmasi, serta memperluas wawasan terhadap hasil temuan yang diperoleh dari analisis data numerik. Dengan demikian, penelitian ini menggabungkan analisis empiris dengan tinjauan teoritis untuk memperoleh kesimpulan yang lebih komprehensif.

3.2 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari *Yahoo Finance*. Data yang digunakan merupakan data historis Indeks Harga Saham Gabungan (IHSG), yang berfungsi sebagai indikator utama dalam mengukur kinerja pasar modal di Indonesia.

Agar analisis yang dilakukan lebih komprehensif, data dibagi menjadi tiga periode berdasarkan kondisi ekonomi yang berbeda. Periode pertama mencakup data sebelum pandemi COVID-19, yaitu dari 1 Januari 2018 hingga 1 Maret 2020. Periode kedua mencerminkan masa pandemi COVID-19, yang berlangsung dari 2 Maret 2020 hingga 30 Desember 2022. Sedangkan periode ketiga mencakup data setelah pandemi, yaitu dari 1 Januari 2023 hingga 5 September 2024.

Data ini diambil untuk memahami bagaimana pergerakan IHSG pada berbagai fase ekonomi, termasuk sebelum, selama, dan setelah pandemi. Penggunaan data dari *Yahoo Finance* memastikan

aksesibilitas dan kelengkapan informasi, mengingat platform ini menyediakan data historis yang dapat digunakan untuk berbagai analisis keuangan dan investasi.

3.3 Variabel Penelitian

Variabel yang digunakan dalam penelitian kali ini adalah historis harga harian penutupan indeks harga saham gabungan (IHSG) sebagai variabel dependen.

3.4 Tahapan Pengelolaan Data untuk Metode ARIMA-GARCH

Dalam penelitian ini, peneliti mengimplementasikan model ARIMA-GARCH dengan menggunakan metode *Maximum Likelihood Estimation* (MLE) untuk menganalisis data penutupan Indeks Harga Saham Gabungan (IHSG) dalam tiga periode berbeda, yaitu sebelum pandemi (1 Januari 2018 – 1 Maret 2020), selama pandemi COVID-19 (2 Maret 2020 – 30 Desember 2022), dan setelah pandemi (1 Januari 2023 – 5 September 2024).

Langkah-langkah dalam penerapan model ini dilakukan secara sistematis guna memastikan bahwa analisis yang dilakukan dapat menggambarkan pola volatilitas dan tren pergerakan indeks saham secara akurat. Model ARIMA digunakan untuk menangkap pola autoregresif dan pergerakan tren data time series, sedangkan model GARCH digunakan untuk mengakomodasi volatilitas yang bersifat heteroskedastis atau berubah-ubah seiring waktu.

Proses implementasi dilakukan dengan mengestimasi parameter model menggunakan metode *Maximum Likelihood Estimation* (MLE), yang bertujuan untuk memperoleh nilai parameter yang paling mungkin sesuai dengan distribusi data yang diamati. Dengan pendekatan ini, penelitian diharapkan dapat memberikan pemahaman yang lebih mendalam mengenai karakteristik fluktuasi IHSG dalam tiga periode berbeda serta dampaknya terhadap pasar modal di Indonesia.

Berikut adalah langkah-langkah sistematis yang dilakukan dalam penelitian ini untuk mengimplementasikan model ARIMA-GARCH dengan pendekatan *Maximum Likelihood Estimation* (MLE) menggunakan iterasi Newton-Raphson:

1. Uji Stasioneritas dan Uji Normalitas

Dilakukan pengujian terhadap sifat stasioneritas data menggunakan uji *Augmented Dickey-Fuller* (ADF) atau uji *Phillips-Perron* (PP), guna memastikan bahwa data tidak memiliki tren atau perubahan pola musiman yang dapat mengganggu hasil estimasi. Selain itu, dilakukan uji normalitas menggunakan metode statistik seperti uji *Jarque-Bera* untuk mengevaluasi distribusi data.

2. Identifikasi Model ARIMA Berdasarkan Grafik ACF dan PACF

Grafik *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF) dianalisis untuk menentukan ordo terbaik dari model ARIMA, dengan memperhatikan pola keterkaitan lag dalam data time series.

3. Estimasi Model ARIMA dengan Metode *Maximum Likelihood*

Parameter model ARIMA diestimasi menggunakan metode *Maximum Likelihood Estimation* (MLE) dengan pendekatan iterasi Newton-Raphson, yang bertujuan untuk mendapatkan nilai parameter optimal sesuai dengan karakteristik data.

4. Pengujian Signifikansi Parameter Model ARIMA

Setelah estimasi dilakukan, parameter dalam model ARIMA diuji signifikansinya dengan melihat nilai p-value dan tingkat kepercayaan statistik, guna memastikan bahwa variabel dalam model memiliki kontribusi yang signifikan terhadap pergerakan IHSG.

5. Uji Heteroskedastisitas dan Uji Korelasi pada Model ARIMA

Pengujian heteroskedastisitas dilakukan untuk mendeteksi apakah terjadi perubahan variabilitas residual dari model ARIMA, menggunakan uji ARCH-LM. Selain itu, dilakukan uji korelasi melalui uji Ljung-Box guna memastikan bahwa residual model tidak memiliki pola keterkaitan yang berulang atau bersifat white noise.

6. Identifikasi Model GARCH sebagai Pendamping Model ARIMA

Model GARCH kemudian diidentifikasi berdasarkan karakteristik volatilitas yang masih tersisa dalam residual model ARIMA, dengan memilih model GARCH yang paling sesuai berdasarkan kriteria informasi seperti AIC (*Akaike Information Criterion*) dan BIC (*Bayesian Information Criterion*).

7. Penentuan Model ARIMA-GARCH dengan Metode *Maximum Likelihood*

Setelah model GARCH diidentifikasi, dilakukan penggabungan model ARIMA dan GARCH untuk membentuk model ARIMA-GARCH. Estimasi parameter dilakukan

kembali menggunakan metode *Maximum Likelihood* dengan iterasi Newton-Raphson guna memastikan akurasi dalam peramalan volatilitas IHSG.

8. Pengujian Signifikansi Parameter Model ARIMA-GARCH

Parameter dalam model ARIMA-GARCH diuji signifikansinya untuk memastikan bahwa kombinasi model ARIMA dan GARCH mampu secara efektif menangkap pola data time series serta volatilitas pasar saham.

9. Uji Normalitas, Uji Heteroskedastisitas, dan Uji Korelasi pada Model ARIMA-GARCH

Setelah estimasi model ARIMA-GARCH diperoleh, dilakukan serangkaian uji diagnostik, termasuk uji normalitas residual untuk memastikan distribusi data residual sesuai asumsi normalitas, uji heteroskedastisitas untuk mendeteksi fluktuasi volatilitas yang tersisa, serta uji korelasi untuk memastikan residual tidak memiliki pola keterkaitan yang dapat mengindikasikan model masih perlu disempurnakan.

10. Peramalan IHSG menggunakan Model ARIMA-GARCH

Hasil estimasi yang diperoleh dari model ARIMA-GARCH berupa model akan digunakan untuk meramalkan pergerakan IHSG dalam berbagai periode mendatang guna memberikan gambaran lebih lanjut mengenai tren pasar saham Indonesia.

11. Evaluasi Performa Model ARIMA-GARCH

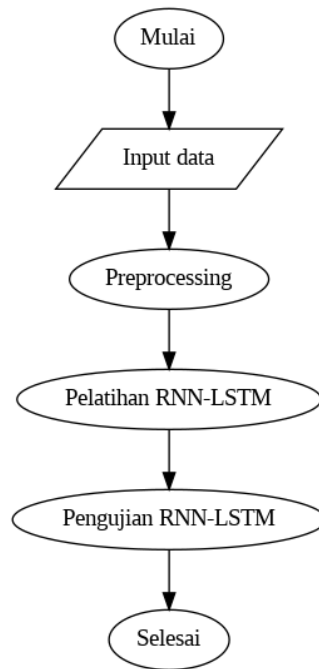
Evaluasi performa model ARIMA-GARCH dilakukan dengan menggunakan beberapa metrik untuk mengukur tingkat akurasi dan efisiensi model. Root Mean Square Error (RMSE) mengukur rata-rata kesalahan dalam satuan asli data, sementara Mean Absolute Percentage Error (MAPE) menghitung persentase kesalahan rata-rata antara prediksi dan nilai aktual. Selain itu, Sum of Squared Errors (SSE) digunakan untuk mengukur total kesalahan kuadrat antara nilai aktual dan prediksi, serta Akaike Information Criterion (AIC) membantu menilai keseimbangan antara akurasi dan kompleksitas model, di mana nilai AIC yang lebih rendah menunjukkan model yang lebih optimal.

Dengan mengikuti tahapan-tahapan tersebut secara sistematis, penelitian ini bertujuan untuk mendapatkan model yang mampu menangkap pola pergerakan IHSG dengan tingkat akurasi yang tinggi serta memberikan pemahaman lebih mendalam mengenai dinamika volatilitas pasar modal di Indonesia.



Gambar 3.1. *Flowchart Analisis Data*

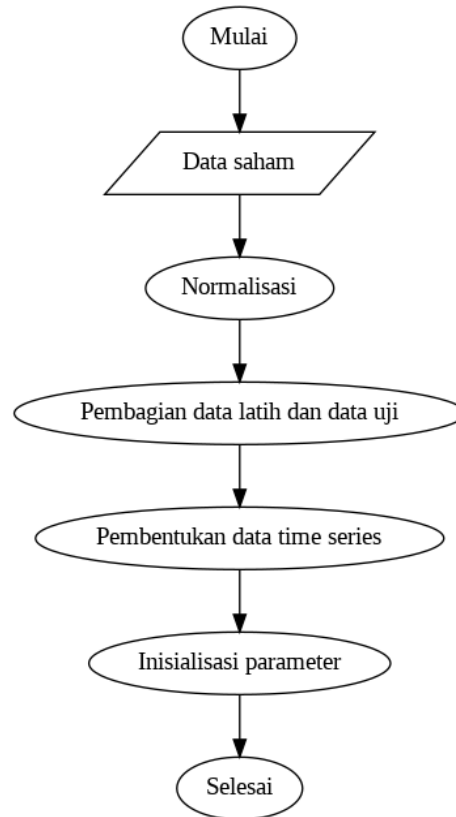
3.5 Tahapan Pengolahan Data untuk Metode LSTM



Gambar 3.2. Diagram Alur metode *forecasting* LSTM, Sumber: Wardani (2021)

Tahapan analisis penelitian sesuai dengan diagram alur pada gambar 3.2 diatas adalah sebagai berikut:

1. Penginputan Data yang dibutuhkan untuk penelitian ini dimasukkan ke dalam sistem sebelum dilakukan tahap pemrosesan lebih lanjut. Setelah data di inputkan kemudian dilakukan *preprocessing* data dengan tahapan seperti berikut:

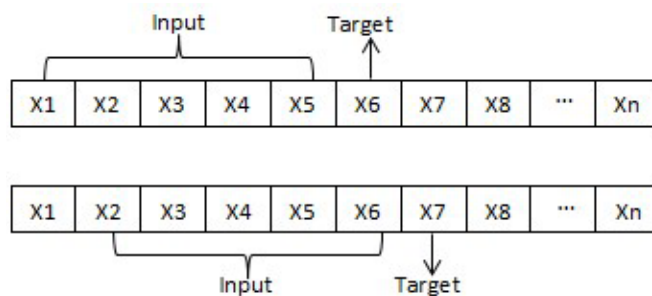


Gambar 3.3. Alur preprocessing, Sumber: Wardani (2021)

2. Tahapan-tahapan melakukan Analisa preprocessing adalah sebagai berikut:
 - a) Normalisasi data dengan rentang nilai $[0,1]$. Normalisasi ini bertujuan untuk mengurangi skala perbedaan antar variabel sehingga dapat meningkatkan stabilitas dan akurasi model.
 - b) Pembagian data dibagi menjadi dua kelompok, yaitu data latih (*training data*) dan data uji (*testing data*). Proporsi pembagian yang digunakan adalah 70% untuk data latih dan 30% untuk data uji. Proporsi ini bertujuan untuk memberikan keseimbangan antara pelatihan model dan evaluasi performanya pada data yang belum pernah dilihat sebelumnya. (Bichri, Chergui dan Hain, 2024)
 - c) Proses pembentukan data time series atau rekonstruksi yang berarti data dikonversi menjadi format deret waktu dengan beberapa pola percobaan yang bervariasi. Pola ini dibuat dengan mempertimbangkan referensi dari penelitian sebelumnya. Beberapa skenario percobaan dilakukan dengan membentuk pola input berdasarkan data harga saham harian. Berikut adalah variasi pola yang digunakan:

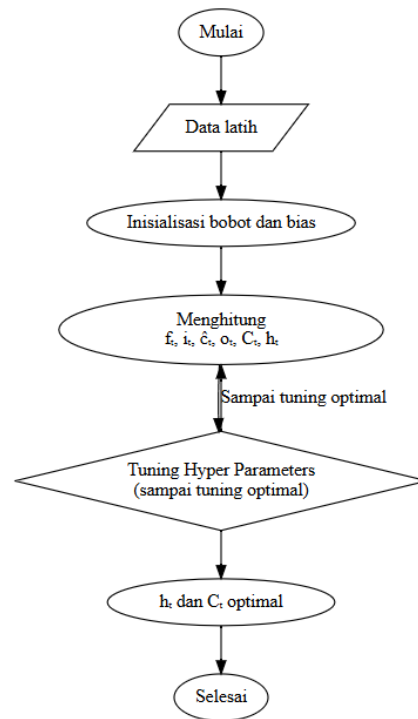
- a. $\{X_t, X_{t+1}\}$
- b. $\{X_{t-1}, X_t, X_{t+1}\}$
- c. $\{X_{t-2}, X_{t-1}, X_t, X_{t+1}\}$
- d. $\{X_{t-3}, X_{t-2}, X_{t-1}, X_t, X_{t+1}\}$
- e. $\{X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}, X_t, X_{t+1}\}$

Dalam setiap pola, nilai X_{t+1} berperan sebagai target (*output*), sedangkan nilai sebelumnya digunakan sebagai variabel input.



Gambar 3.4 Ilustrasi pola time series, Sumber: Wardani (2021)

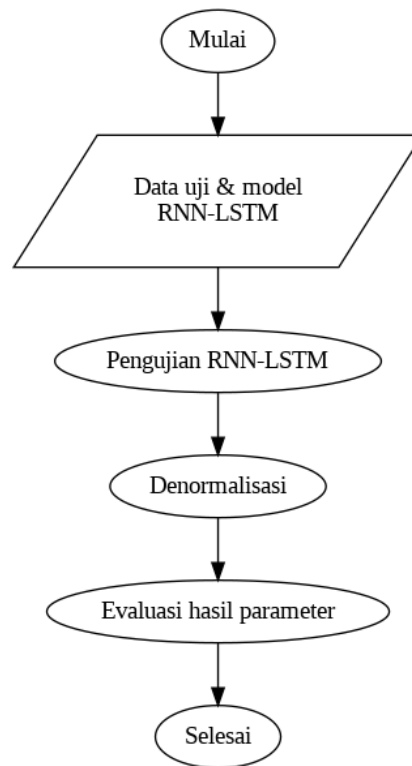
- d) Ilustrasi pola *time series* di atas menggambarkan bagaimana data historis digunakan sebagai masukan untuk memprediksi harga saham di masa mendatang. Dengan pola ini, model dapat mempelajari hubungan antara nilai sebelumnya dan nilai prediksi yang diharapkan.
- e) Inisialisasi parameter Untuk memastikan model bekerja dengan optimal, diperlukan penentuan parameter utama yang mencakup jumlah neuron dalam layer tersembunyi (*hidden layer*), ukuran batch (*batch size*), serta jumlah *epoch* maksimum yang digunakan dalam proses pelatihan. Pemilihan parameter ini dilakukan dengan mempertimbangkan keseimbangan antara akurasi dan efisiensi komputasi, sehingga model tidak mengalami *overfitting* atau *underfitting* dalam proses pembelajaran.



Gambar 3.4. Alur pelatihan prediksi harga saham menggunakan LSTM, Sumber: Wardani (2021)

4. Tahapan-tahapan pelatihan (*training*) data LSTM adalah sebagai berikut:
 - a) Pemrosesan Data Latih yang telah melalui tahap preprocessing digunakan untuk membangun model LSTM. Data ini akan menjadi dasar dalam proses pelatihan untuk memahami pola dalam deret waktu.
 - b) Inisialisasi Bobot dan Bias dengan melakukan Sebelum model *machine learning* dilakukan inisialisasi terhadap bobot dan bias yang akan digunakan dalam perhitungan setiap neuron. Parameter bobot yang diinisialisasi mencakup W_f, W_i, W_{ct}, W_o , sedangkan parameter bias terdiri dari b_f, b_i, b_{ct}, b_o . Bobot dan bias ini akan terus diperbarui selama proses pelatihan untuk meningkatkan akurasi model.
 - c) Perhitungan Fungsi Aktivasi pada Setiap Neuron pada data latih yang kemudian diproses melalui jaringan LSTM dengan menerapkan serangkaian perhitungan pada unit-unit gates di dalamnya. Tahapan perhitungan ini meliputi:
 - a. *Forget Gate* (f_t): Bertugas menentukan informasi mana yang harus dilupakan dari memori sebelumnya.

- b. *Input Gate* (i_t) ($\tilde{C}_t$): Mengontrol informasi baru yang akan ditambahkan ke dalam cell state.
 - c. *Cell State* (C_t): Merekam informasi yang telah diperbarui dengan mempertimbangkan forget dan input gate.
 - d. *Output Gate* (o_t): Mengatur bagian dari *cell state* yang akan diteruskan sebagai output jaringan.
 - e. *Hidden State* (h_t): Merupakan keluaran akhir (*Output*) dari model yang berisi representasi informasi dari data sebelumnya dan digunakan dalam tahap selanjutnya.
- d) Evaluasi Model untuk Mendapatkan Nilai Optimal dengan melakukan Proses pembelajaran terus berulang hingga diperoleh parameter h_t dan C_t yang optimal, yang menandakan bahwa model telah berhasil menangkap pola dalam data secara efektif.



Gambar 3.5. Alur pengujian prediksi harga saham menggunakan LSTM, Sumber: Wardani (2021)

5. Tahapan-tahapan pengujian (*testing*) data LSTM adalah sebagai berikut:

- a) Memasukan data uji yang telah dipersiapkan digunakan sebagai input untuk menguji performa model LSTM yang telah dilatih sebelumnya. Model optimal yang diperoleh dari tahap pelatihan akan diaplikasikan untuk mengukur sejauh mana jaringan dapat mengenali pola dalam data uji. Pengujian ini bertujuan untuk mengukur tingkat kesalahan (*error*) yang dihasilkan oleh model. Jika nilai error yang diperoleh kecil, maka model dianggap memiliki generalisasi yang baik dan dapat digunakan untuk prediksi atau peramalan dengan tingkat kepercayaan yang tinggi.
- b) Pengelolaan Data uji dengan memproses data menggunakan model yang telah dilatih dengan parameter h_t dan C_t yang optimal. Proses ini melibatkan perhitungan kembali semua fungsi gates dalam jaringan LSTM untuk mendapatkan output prediksi, dengan tahapan sebagai berikut:
 - a. *Forget Gate* (f_t): Menentukan informasi dari state sebelumnya yang harus dipertahankan atau dilupakan.
 - b. *Input Gate* (i_t) ($\tilde{C}_t$): Mengontrol informasi baru yang akan dimasukkan ke dalam cell state.
 - c. *Cell State* (C_t): Mengupdate memori berdasarkan input baru dan informasi dari forget gate.
 - d. *Output Gate* (o_t): Memilih bagian dari *cell state* yang akan digunakan sebagai hasil keluaran model.
 - e. *Hidden State* (h_t): Menghasilkan representasi akhir dari model yang akan dibandingkan dengan nilai aktual.
- c) Denormalisasi hasil prediksi dilakukan setelah model menghasilkan output prediksi, dilakukan proses denormalisasi untuk mengubah nilai yang telah dinormalisasi sebelumnya ke dalam skala aslinya. Proses ini penting agar hasil prediksi memiliki interpretasi yang sesuai dengan data aktual, seperti harga saham atau indikator finansial lainnya.
- d) Terakhir, Evaluasi performa model adalah tahapan terakhir yang dilakukan dengan mengevaluasi hasil prediksi dengan menggunakan metrik tertentu guna mengukur tingkat akurasi model. Beberapa metrik evaluasi yang umum digunakan dalam peramalan berbasis LSTM meliputi:

- a. *Root Mean Square Error* (RMSE): Digunakan untuk mengukur seberapa besar rata-rata kesalahan model dalam satuan asli data. Nilai RMSE yang lebih kecil menunjukkan bahwa model memiliki performa yang baik.
- b. *Mean Absolute Percentage Error* (MAPE): Menghitung persentase kesalahan rata-rata antara prediksi dan nilai aktual, sehingga dapat digunakan untuk memahami seberapa besar tingkat kesalahan relatif dalam bentuk persentase.
- c. *Mean Squared Error* (MSE): Mengukur rata-rata dari selisih kuadrat antara nilai aktual dan nilai prediksi. Semakin kecil nilai MSE, semakin baik model dalam menangkap pola data.
- d. *Akaike Information Criterion* (AIC): Digunakan untuk mengevaluasi model berdasarkan keseimbangan antara keakuratan dan kompleksitasnya. Model dengan nilai AIC yang lebih rendah dianggap lebih optimal karena memberikan keseimbangan terbaik antara ketepatan dan jumlah parameter yang digunakan.

3.6 Perbandingan Model ARIMA-GARCH dan LSTM

Dalam penelitian ini, model ARIMA-GARCH dan LSTM digunakan sebagai pendekatan utama dalam memprediksi pergerakan IHSG. Untuk menilai kinerja masing-masing model, tiga metrik evaluasi utama digunakan, yaitu Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), dan Root Mean Squared Error (RMSE). Pemilihan metrik ini didasarkan pada pertimbangan bahwa masing-masing memberikan perspektif yang berbeda dalam mengukur tingkat kesalahan prediksi, sehingga dapat memberikan evaluasi yang lebih komprehensif terhadap performa model.

- MSE (Mean Squared Error) digunakan karena memberikan bobot lebih besar pada kesalahan yang lebih besar, sehingga sangat berguna dalam mengidentifikasi model yang menghasilkan deviasi prediksi yang signifikan. Semakin kecil nilai MSE, semakin baik model dalam meminimalkan kesalahan kuadrat rata-rata antara prediksi dan nilai aktual.
- MAPE (Mean Absolute Percentage Error) dipilih karena memberikan ukuran kesalahan dalam bentuk persentase, memungkinkan perbandingan performa model secara relatif terhadap skala data. MAPE sangat berguna dalam menilai efektivitas model dalam berbagai periode, terutama ketika skala harga mengalami perubahan signifikan.

- RMSE (Root Mean Squared Error) digunakan karena mempertahankan satuan yang sama dengan data aktual, sehingga lebih mudah diinterpretasikan dalam konteks pasar keuangan. RMSE memberikan gambaran seberapa besar rata-rata kesalahan prediksi dalam unit indeks harga saham gabungan.

Pada penelitian ini, model ARIMA-GARCH dan LSTM diterapkan untuk menganalisis pergerakan IHSG dalam tiga periode utama, yaitu sebelum pandemi (1 Januari 2018 – 1 Maret 2020), selama pandemi (2 Maret 2020 – 31 Desember 2022), dan setelah pandemi (1 Januari 2023 – 5 September 2024). Evaluasi dilakukan dengan membandingkan nilai MSE, MAPE, dan RMSE yang dihasilkan oleh kedua model pada setiap periode.

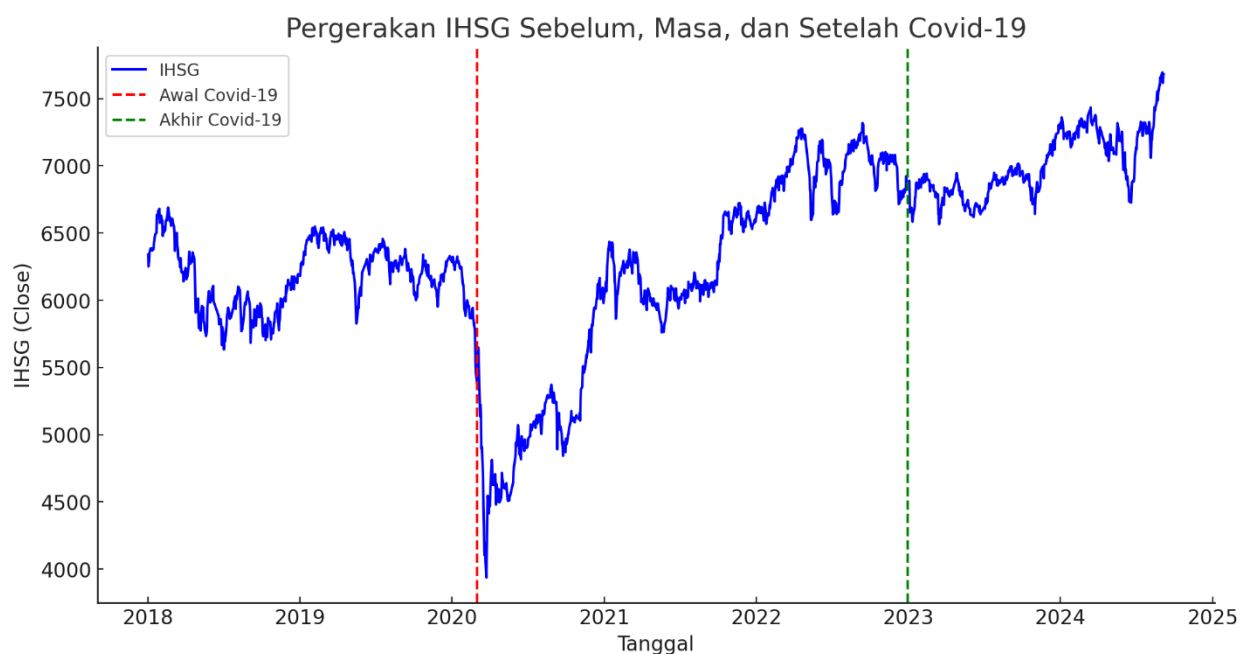
Setelah model ARIMA-GARCH dan LSTM diterapkan, hasil evaluasi ketiga metrik akan dianalisis untuk menentukan model yang lebih optimal dalam menangani pergerakan IHSG pada setiap periode. Model dengan nilai MSE, MAPE, dan RMSE yang lebih rendah akan dianggap lebih baik dalam memprediksi IHSG, dengan mempertimbangkan karakteristik volatilitas yang terjadi pada setiap keadaan pasar.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Analisis Deskriptif Data IHSG

Sebelum dilakukan analisis menggunakan metode ARIMA-GARCH dan LSTM, penelitian ini akan terlebih dahulu menyajikan serta menganalisis data pergerakan Indeks Harga Saham Gabungan (IHSG) berdasarkan tiga periode utama, yaitu periode sebelum pandemi COVID-19, periode selama pandemi COVID-19, dan periode setelah pandemi COVID-19. Data yang digunakan dalam analisis ini merupakan data harian harga penutupan IHSG, yang akan dianalisis secara deskriptif untuk memberikan gambaran awal mengenai karakteristik dan pola pergerakan indeks pada masing-masing periode sebelum dilakukan pemodelan lebih lanjut.



Gambar 4.1. Pergerakan IHSG dari 1 Januari 2018 sampai 5 September 2024. Sumber: Data diolah

Gambar 4.1 memperlihatkan pergerakan Indeks Harga Saham Gabungan (IHSG) dari 1 Januari 2018 hingga 5 September 2024, yang mencerminkan respons pasar modal Indonesia terhadap

dinamika ekonomi domestik dan global. Pada periode 2018–2019, IHSG relatif stabil dalam kisaran 6.000–6.500, mencerminkan kondisi makroekonomi yang cukup kondusif. Pertumbuhan ekonomi Indonesia selama dua tahun tersebut berkisar antara 5,1–5,2% per tahun, dengan inflasi terjaga dalam rentang 3–3,2%, dan nilai tukar rupiah terhadap dolar AS juga relatif stabil di kisaran Rp14.000–Rp14.500. Stabilitas ini mencerminkan persepsi positif investor terhadap prospek ekonomi nasional serta pengelolaan fiskal dan moneter yang hati-hati oleh pemerintah dan Bank Indonesia.

Namun, memasuki awal 2020, IHSG mengalami penurunan tajam hingga menyentuh level di bawah 4.000 sebagai respons terhadap guncangan global akibat pandemi COVID-19. Pada saat yang sama, ekonomi Indonesia mengalami kontraksi sebesar -2,07% pada tahun 2020, pertama kalinya dalam lebih dari dua dekade, dengan tekanan berat pada sektor konsumsi dan investasi. Inflasi menurun menjadi sekitar 1,68%, mencerminkan lemahnya permintaan domestik, sementara nilai tukar rupiah sempat melemah tajam hingga menyentuh Rp16.000 per dolar AS pada Maret 2020 akibat capital outflow besar-besaran dari pasar keuangan domestik. Kondisi ini turut memperdalam tekanan terhadap IHSG sebagai refleksi dari sentimen negatif pelaku pasar.

Seiring dengan berbagai kebijakan stimulus fiskal dan moneter seperti Program Pemulihan Ekonomi Nasional (PEN), pemangkasan suku bunga acuan, serta peluncuran vaksinasi COVID-19, IHSG mulai menunjukkan pemulihan dari pertengahan 2020 hingga 2021. Pertumbuhan ekonomi nasional kembali positif sebesar 3,69% pada 2021 dan meningkat menjadi 5,3% pada 2022, didorong oleh pulihnya aktivitas konsumsi dan ekspor komoditas utama seperti batubara dan kelapa sawit. Nilai tukar rupiah juga menguat kembali ke kisaran Rp14.200–Rp14.500, sementara inflasi tetap terjaga di bawah 4%, yang memberikan kepercayaan lebih kepada investor. Hal ini tercermin dari tren penguatan IHSG yang stabil menuju kisaran 6.800–7.200 sepanjang tahun 2022–2023, meskipun diselingi fluktuasi akibat isu global seperti ketegangan geopolitik dan kebijakan suku bunga The Fed.

Pada tahun 2024, IHSG menunjukkan tren penguatan yang lebih konsisten dengan level mendekati 8.000, menandakan sentimen pasar yang semakin positif. Tren ini selaras dengan proyeksi pertumbuhan ekonomi Indonesia yang tetap solid di atas 5%, ditopang oleh stabilitas harga, peningkatan konsumsi domestik, serta iklim investasi yang membaik. Selain itu, penguatan nilai tukar rupiah yang tetap berada di bawah Rp15.000 dan inflasi yang terkendali dalam rentang 2,5–

3% turut mendukung kestabilan makroekonomi. Kombinasi antara indikator ekonomi makro yang positif dan reformasi struktural yang berkelanjutan menjadi pendorong utama penguatan IHSG pada periode pasca-pandemi, mencerminkan optimisme pelaku pasar terhadap keberlanjutan pemulihan dan prospek pertumbuhan jangka menengah ekonomi Indonesia.

4.1.1 Data IHSG Sebelum Covid-19

Pada periode sebelum COVID-19, yaitu antara 1 Januari 2018 hingga 1 Maret 2020, pergerakan Indeks Harga Saham Gabungan (IHSG) dipengaruhi oleh kombinasi faktor global dan domestik, termasuk dinamika makroekonomi yang cukup kompleks. Dari sisi global, perang dagang antara Amerika Serikat dan Tiongkok sejak awal 2018 menimbulkan ketidakpastian tinggi di pasar keuangan global. Peningkatan tarif impor antar kedua negara tidak hanya memperlambat laju perdagangan internasional, tetapi juga menurunkan prospek pertumbuhan ekonomi global, sehingga memicu kehati-hatian investor asing untuk menempatkan dananya di pasar negara berkembang, termasuk Indonesia. Ketidakpastian ini turut menyebabkan peningkatan volatilitas di pasar saham, tercermin dari fluktuasi IHSG yang berada di kisaran 5.600–6.500 pada periode tersebut.

Kebijakan moneter global juga memberikan tekanan tambahan. The Federal Reserve (The Fed) secara bertahap menaikkan suku bunga acuan (Fed Funds Rate) sepanjang tahun 2018 hingga mencapai 2,5% pada akhir tahun, yang mendorong terjadinya capital outflow dari emerging markets. Di Indonesia, hal ini menyebabkan tekanan pada nilai tukar rupiah, yang sempat melemah hingga Rp15.200 per USD pada Oktober 2018. Untuk merespons kondisi tersebut, Bank Indonesia menaikkan suku bunga acuan 7-Day Reverse Repo Rate sebanyak enam kali sepanjang 2018, dari 4,25% menjadi 6,00%. Langkah ini dilakukan untuk menjaga stabilitas rupiah dan menahan laju inflasi, yang meskipun tetap terkendali dalam kisaran 3,13% pada 2018 dan 2,72% pada 2019, tetap menghadapi risiko dari imported inflation. Pengetatan suku bunga ini turut menekan sektor-sektor sensitif terhadap pembiayaan, seperti properti dan otomotif, yang tercermin dalam stagnasi kinerja saham sektor tersebut.

Di sisi domestik, perkembangan politik juga memengaruhi dinamika pasar modal. Pelaksanaan Pilkada serentak 2018 dan Pemilu Presiden 2019 menimbulkan ketidakpastian jangka pendek akibat kecenderungan investor untuk wait-and-see selama periode transisi politik. Namun,

kemenangan Presiden Joko Widodo pada pemilu 2019 dinilai memberikan kepastian politik dan keberlanjutan kebijakan pembangunan, yang secara umum diterima positif oleh pelaku pasar. Stabilitas politik ini diperkuat oleh pertumbuhan ekonomi Indonesia yang cukup solid, yakni sebesar 5,17% pada 2018 dan 5,02% pada 2019, yang ditopang oleh konsumsi rumah tangga dan belanja infrastruktur pemerintah. Di sisi lain, pelaksanaan program amnesti pajak tahap lanjutan dan reformasi struktural turut mendorong perbaikan iklim investasi dan likuiditas pasar.

Meskipun IHSG menghadapi tekanan dari faktor eksternal seperti perang dagang dan normalisasi moneter global, serta dinamika politik domestik, pergerakan indeks tetap relatif stabil dalam jangka menengah. Ditopang oleh fundamental ekonomi yang kuat dan manajemen kebijakan moneter yang adaptif, pasar modal Indonesia mampu bertahan hingga awal 2020. Namun demikian, stabilitas tersebut akhirnya terguncang secara signifikan ketika pandemi COVID-19 mulai memengaruhi pasar global dan domestik pada kuartal pertama tahun 2020, yang menandai awal dari periode ketidakpastian baru bagi IHSG dan perekonomian nasional secara keseluruhan.

Tabel 4.1. Analisis Data Deskriptif IHSG Sebelum Covid-19

<i>Analisis Deskriptif Sebelum Covid-19</i>	
Mean	6181.092
Median	6210.698
Standard Deviation	244.290
Range	1236.583
Minimum	5452.704
Maximum	6689.287
Sum	3257435.540
Count	527.000

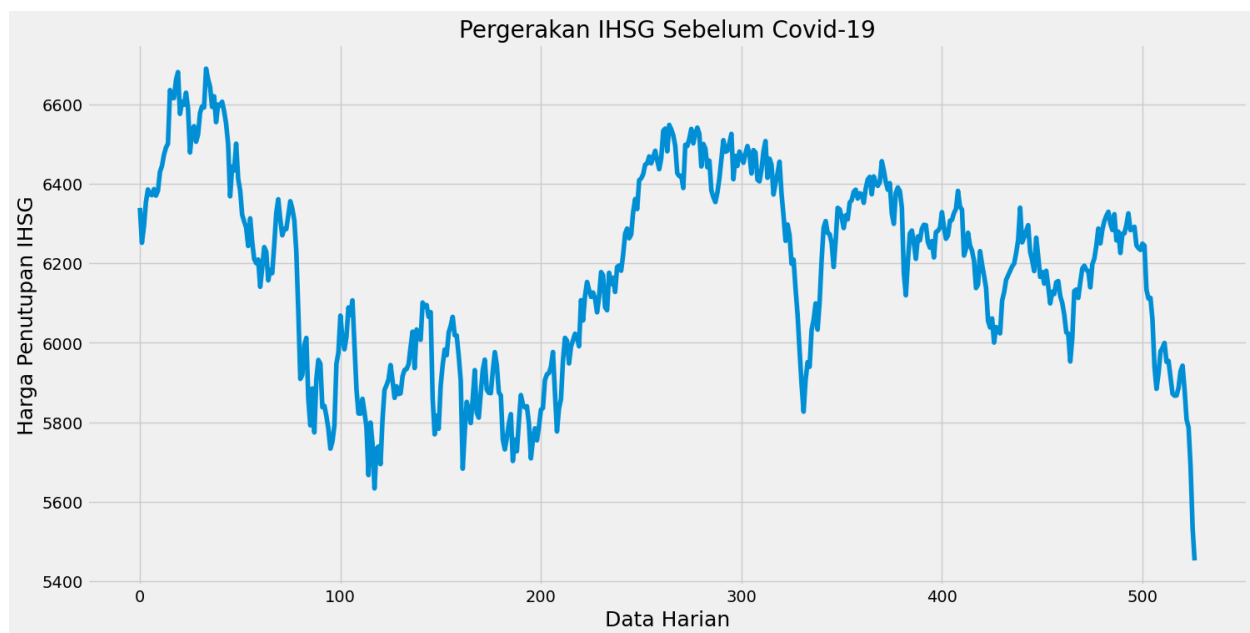
Sumber: Data diolah

Berdasarkan table 4.1 Hasil analisis statistik deskriptif menunjukkan bahwa selama periode tersebut, IHSG memiliki nilai rata-rata (mean) sebesar 6.181,092, yang menggambarkan tingkat harga indeks secara umum selama periode yang diamati. Sementara itu, nilai median sebesar 6.210,698 menunjukkan bahwa separuh dari total observasi berada di atas nilai ini, yang menandakan distribusi data yang relatif simetris.

Dalam hal penyebaran data, standar deviasi sebesar 244,290 mencerminkan tingkat volatilitas pasar yang cukup tinggi. Nilai ini menunjukkan bahwa harga IHSG dalam periode tersebut mengalami fluktuasi rata-rata sekitar 244 poin dari nilai rata-ratanya yang menggambarkan besarnya variasi data IHSG terhadap nilai rata-rata.

Dalam periode pengamatan ini, nilai maksimum IHSG tercatat sebesar 6.689,287, yang mencerminkan titik tertinggi indeks selama kurun waktu tersebut. Sementara itu, nilai minimum sebesar 5.452,704 menunjukkan level terendah yang dicapai IHSG dalam periode yang sama. Dengan rentang (range) sebesar 1.236,583, dapat disimpulkan bahwa perbedaan antara harga tertinggi dan terendah dalam rentang waktu tersebut cukup besar, mencerminkan volatilitas pasar yang signifikan.

Selain itu, jumlah total nilai indeks selama periode tersebut mencapai 3.257.435,540, dengan jumlah observasi sebanyak 527 hari perdagangan. Tidak adanya nilai mode dalam distribusi data menunjukkan bahwa tidak ada satu harga tertentu yang paling sering muncul selama periode pengamatan.



Gambar 4.2. Pergerakan IHSG Sebelum Covid. Sumber: Data diolah

Berdasarkan Gambar 4.2, Pada awal periode, IHSG mengalami tren bullish, di mana indeks mengalami kenaikan yang cukup signifikan dan mencapai puncaknya di atas 6.600. Namun,

setelah mencapai titik tertinggi, terjadi koreksi tajam yang menyebabkan IHSG turun hingga menyentuh level 5.800, mencerminkan adanya tekanan jual yang cukup besar di pasar.

Setelah koreksi tersebut, IHSG memasuki fase konsolidasi, dengan pola pergerakan naik-turun yang cukup tajam. Beberapa kali indeks mencoba kembali ke level 6.400 – 6.500, tetapi setiap kenaikan selalu diikuti oleh koreksi yang cukup dalam, menunjukkan bahwa pasar berada dalam kondisi yang belum stabil.

Menjelang akhir periode, khususnya pada awal tahun 2020, IHSG mulai mengalami tren penurunan tajam. Penurunan ini semakin curam seiring dengan meningkatnya kekhawatiran global terhadap pandemi COVID-19. Dalam grafik terlihat bahwa indeks jatuh dengan cepat hingga mendekati 5.500, mencerminkan aksi jual besar-besaran akibat ketidakpastian yang meningkat di pasar keuangan.

Berdasarkan hasil analisis deskriptif dan visualisasi data, pergerakan IHSG dalam periode 1 Januari 2018 hingga 1 Maret 2020 menunjukkan dinamika yang cukup kompleks. Meskipun rata-rata IHSG berada di kisaran 6.181,092, pergerakan indeks mengalami volatilitas yang cukup tinggi, dengan perbedaan yang cukup besar antara nilai tertinggi dan terendah. Nilai skewness yang sedikit negatif menunjukkan bahwa IHSG cenderung mengalami lebih banyak periode penurunan dibandingkan kenaikan. Sementara itu, peningkatan volatilitas yang signifikan menjelang awal tahun 2020 menunjukkan respons pasar terhadap ketidakpastian ekonomi global yang meningkat akibat ancaman pandemi COVID-19.

4.1.1 Data IHSG pada Masa Covid-19

Pada periode pandemi COVID-19, yaitu antara 2 Maret 2020 hingga 31 Desember 2022, pasar modal Indonesia mengalami gejolak yang sangat signifikan, ditandai dengan kepanikan pasar, anjloknya Indeks Harga Saham Gabungan (IHSG), dan meningkatnya volatilitas akibat ketidakpastian global yang disebabkan oleh penyebaran virus secara masif. Pada Maret 2020, IHSG sempat jatuh di bawah level 4.000, level terendah dalam lebih dari satu dekade. Sentimen negatif ini dipicu oleh kebijakan lockdown dan pembatasan mobilitas di berbagai negara yang menyebabkan kontraksi tajam dalam aktivitas ekonomi, termasuk konsumsi, investasi, dan ekspor.

Hal ini tercermin dalam pertumbuhan ekonomi Indonesia yang mengalami kontraksi sebesar -2,07% pada tahun 2020 pertama kalinya sejak krisis moneter 1998.

Penurunan ekonomi ini juga berdampak pada nilai tukar rupiah, yang sempat menyentuh level Rp16.575 per USD pada akhir Maret 2020, sebelum akhirnya mulai stabil berkat intervensi aktif dari Bank Indonesia (BI). Untuk menjaga stabilitas makroekonomi, BI menurunkan suku bunga acuan (BI 7-Day Reverse Repo Rate) secara bertahap dari 5,00% pada awal 2020 menjadi 3,50% pada awal 2021—terendah dalam sejarah. Selain itu, pemerintah Indonesia menggelontorkan stimulus fiskal dalam bentuk Program Pemulihan Ekonomi Nasional (PEN) dengan total anggaran mencapai lebih dari Rp600 triliun pada 2020, yang kemudian dilanjutkan pada 2021 dan 2022. Program ini mencakup bantuan sosial, subsidi gaji, insentif pajak, serta dukungan untuk UMKM dan sektor kesehatan.

Dari sisi inflasi, tekanan harga sempat mereda selama awal pandemi akibat rendahnya permintaan domestik. Inflasi Indonesia tercatat hanya sebesar 1,68% pada 2020, jauh di bawah target Bank Indonesia. Namun, mulai 2021, tekanan inflasi secara bertahap meningkat seiring dengan pulihnya permintaan dan gangguan rantai pasok global. Pada 2022, inflasi kembali melonjak dan mencapai 5,51% (year-on-year) pada akhir tahun, terutama disebabkan oleh kenaikan harga energi dan pangan, serta penyesuaian harga BBM subsidi dalam negeri.

Pada tahun 2021 dan 2022, IHSG menunjukkan pemulihan bertahap, seiring meningkatnya cakupan vaksinasi COVID-19 nasional yang mencapai lebih dari 70% populasi pada akhir 2021 dan pelonggaran bertahap kebijakan pembatasan sosial. Pertumbuhan ekonomi Indonesia kembali positif sebesar 3,69% pada 2021 dan meningkat menjadi 5,31% pada 2022, didorong oleh pemulihan konsumsi rumah tangga dan ekspor komoditas unggulan seperti batu bara, nikel, dan CPO yang melonjak akibat disrupsi pasokan global.

Namun, pemulihan ini tidak sepenuhnya linier. Ketidakpastian masih membayangi akibat kemunculan varian Delta (2021) dan Omicron (2022) yang menyebabkan lonjakan kasus dan menimbulkan kekhawatiran akan perlambatan ekonomi. Di samping itu, kebijakan tapering off oleh The Federal Reserve yang dimulai pada akhir 2021, serta rencana kenaikan suku bunga acuan AS pada 2022, turut memicu tekanan di pasar keuangan global. Rupiah kembali melemah secara bertahap terhadap dolar AS, berada di kisaran Rp15.500 per USD pada akhir 2022.

Selain faktor kesehatan dan moneter, krisis energi di Eropa serta konflik Rusia–Ukraina yang pecah pada Februari 2022 memberikan tekanan tambahan. Konflik tersebut memicu lonjakan harga energi dan pangan global, yang turut berdampak pada pasar domestik, terutama sektor energi, pertambangan, dan transportasi. Meskipun IHSG mengalami pemulihan dari posisi terendah di 2020, volatilitas tetap tinggi hingga akhir 2022, mencerminkan ketidakpastian dalam transisi menuju fase pasca-pandemi serta respons pasar terhadap tekanan inflasi global dan gejolak geopolitik internasional.

Tabel 4.2. Analisis Data Deskriptif IHSG pada masa Covid-19

<i>Analisis Deskriptif pada Masa Covid-19</i>	
Mean	6156.890
Median	6227.728
Standard Deviation	809.003
Range	3380.384
Minimum	3937.632
Maximum	7318.016
Sum	4266724.759
Count	693.000

Sumber: Data diolah

Berdasarkan Tabel 4.2 hasil analisis deskriptif, nilai rata-rata (mean) IHSG selama periode ini adalah 6.156,890, yang menunjukkan tingkat harga indeks secara umum selama pandemi. Sementara itu, nilai median sebesar 6.227,728 menunjukkan bahwa lebih dari setengah periode perdagangan memiliki harga indeks di atas angka tersebut.

Dalam hal fluktuasi pasar, IHSG mengalami volatilitas yang jauh lebih tinggi dibandingkan periode sebelum pandemi. Standar deviasi sebesar 809,003 mencerminkan tingginya ketidakstabilan harga indeks selama periode ini. Nilai ini jauh lebih besar dibandingkan standar deviasi sebelum pandemi (244,290), yang menunjukkan bahwa gejolak pasar selama pandemi jauh lebih ekstrem yang menggambarkan tingkat dispersi data yang sangat besar.

Dalam periode pengamatan ini, IHSG mencapai titik terendah (minimum) di 3.937,632, yang terjadi di awal pandemi saat terjadi kepanikan besar di pasar. Sementara itu, nilai tertinggi

(maksimum) yang dicapai IHSG dalam periode ini adalah 7.318,016, yang mencerminkan pemulihan dan optimisme pasar pasca-pandemi. Rentang (range) sebesar 3.380,384 menunjukkan adanya perbedaan harga yang sangat besar antara titik tertinggi dan terendah dalam kurun waktu tersebut.

Secara umum, jumlah total nilai indeks selama periode ini mencapai 4.266.724,759, dengan jumlah observasi sebanyak 693 hari perdagangan. Tidak seperti periode sebelum pandemi, di mana tidak ditemukan mode (nilai yang paling sering muncul), pada periode ini terdapat nilai mode sebesar 6.078,568, yang menunjukkan bahwa harga ini adalah yang paling sering muncul selama periode tersebut.



Gambar 4.3. Pergerakan IHSG pada masa Covid. Sumber: Data diolah

Pada awal periode, Gambar 4.3 memperlihatkan IHSG mengalami penurunan yang sangat tajam, yang mencerminkan kepanikan pasar akibat ketidakpastian ekonomi yang ditimbulkan oleh pandemi COVID-19. IHSG anjlok hingga menyentuh level 3.937, yang merupakan titik terendah dalam periode ini.

Setelah menyentuh level terendah, IHSG mulai pulih secara bertahap, didorong oleh berbagai kebijakan stimulus ekonomi dari pemerintah dan bank sentral. Secara bertahap, indeks mulai

menunjukkan tren bullish, dengan kenaikan yang cukup signifikan menuju level 5.500 - 6.000 dalam beberapa bulan setelah krisis awal.

Tren pemulihan semakin menguat pada tahun 2021 dan 2022, dengan IHSG akhirnya mencapai 7.318, level tertinggi dalam periode ini. Selama periode pemulihan ini, terdapat beberapa fase koreksi akibat faktor ekonomi global dan domestik, tetapi secara umum tren kenaikan tetap dominan hingga akhir tahun 2022.

Periode 2 Maret 2020 hingga 31 Desember 2022 menunjukkan dinamika pasar saham yang sangat berbeda dibandingkan periode sebelum pandemi. Awalnya, IHSG mengalami penurunan drastis hingga 3.937,632 akibat ketidakpastian global yang sangat tinggi. Namun, seiring dengan pemulihan ekonomi dan kebijakan stimulus yang diterapkan, IHSG menunjukkan pemulihan yang kuat dan mencapai level tertinggi di 7.318,016.

Dari segi analisis deskriptif, volatilitas IHSG selama pandemi jauh lebih tinggi dibandingkan periode sebelum Covid-19, dengan standar deviasi yang meningkat lebih dari tiga kali lipat. Meskipun secara umum distribusi data menunjukkan kecenderungan negatif, pemulihan pasar yang kuat menjelang akhir periode menunjukkan adanya optimisme investor terhadap prospek ekonomi pasca-pandemi.

4.1.1 Data IHSG Setelah Covid-19

Pada periode 1 Januari 2022 hingga 5 September 2024, pasar modal Indonesia berada dalam fase transisi dari krisis pandemi menuju kondisi ekonomi pasca-pandemi yang lebih stabil. Meskipun aktivitas ekonomi mulai pulih, kondisi pasar keuangan global masih diliputi ketidakpastian tinggi akibat normalisasi kebijakan moneter di negara-negara maju, khususnya Amerika Serikat. The Federal Reserve secara agresif menaikkan suku bunga acuan dari 0,25% pada awal 2022 menjadi lebih dari 5,25% pada pertengahan 2023 sebagai respons terhadap lonjakan inflasi AS yang sempat menembus 9% (yoy) pada pertengahan 2022. Kebijakan ini memicu capital outflow dari negara berkembang, termasuk Indonesia, karena investor global beralih ke aset-aset safe haven seperti US Treasury. Akibatnya, nilai tukar rupiah mengalami tekanan dan sempat melemah ke kisaran Rp15.600–Rp15.900 per USD selama tahun 2022–2023.

Namun demikian, perekonomian domestik menunjukkan ketahanan yang cukup baik. Pertumbuhan ekonomi Indonesia tetap solid, tercatat sebesar 5,31% pada tahun 2022 dan sedikit melambat ke kisaran 5,0–5,1% pada tahun 2023, ditopang oleh konsumsi rumah tangga, pemulihan pariwisata, serta ekspor komoditas unggulan seperti batu bara, nikel, dan CPO. Di sisi lain, inflasi domestik juga sempat meningkat akibat kenaikan harga energi global dan penyesuaian harga bahan bakar subsidi, dengan inflasi mencapai 5,51% pada akhir 2022. Pemerintah merespons dengan menjaga pasokan pangan, memberikan subsidi energi, serta memperluas jangkauan bantuan sosial. Seiring masuknya tahun 2023 dan 2024, inflasi mulai mereda ke kisaran target Bank Indonesia sebesar 2,5–3,5%, membantu memperkuat daya beli masyarakat dan menjaga stabilitas sektor riil.

Dari sisi kebijakan moneter, Bank Indonesia secara bertahap menaikkan suku bunga acuan dari level terendah 3,5% menjadi 6,00% pada 2023 sebagai langkah antisipatif terhadap tekanan inflasi dan menjaga daya tarik aset dalam negeri di tengah tren suku bunga tinggi secara global. Kebijakan ini diimbangi dengan koordinasi yang erat dengan kebijakan fiskal untuk memastikan keberlanjutan pemulihan ekonomi. Sektor manufaktur dan industri pengolahan mengalami ekspansi, tercermin dalam indeks PMI manufaktur Indonesia yang tetap berada di atas 50 sepanjang sebagian besar periode 2023–2024. Sementara itu, neraca perdagangan tetap mencatat surplus, didorong oleh tingginya ekspor komoditas strategis meskipun harga beberapa komoditas mulai mengalami koreksi pada pertengahan 2023.

Geopolitik global juga turut memengaruhi dinamika pasar. Ketegangan antara Rusia dan Ukraina sejak awal 2022 menyebabkan lonjakan harga minyak dan gas, yang berdampak pada sektor energi dan biaya logistik global. Sementara itu, ketegangan antara Tiongkok dan Taiwan menimbulkan kekhawatiran terhadap rantai pasok semikonduktor dan barang elektronik global. Namun, Indonesia justru mendapat keuntungan relatif dari krisis energi global melalui peningkatan ekspor batu bara dan gas alam, yang memperkuat posisi neraca perdagangan dan menopang penerimaan negara. Stabilitas fiskal juga tercermin dalam defisit anggaran yang terkendali dan penurunan rasio utang terhadap PDB secara bertahap.

Meskipun pasar modal Indonesia tetap menghadapi volatilitas akibat tekanan eksternal, tren pergerakan IHSG selama 2022 hingga kuartal ketiga 2024 menunjukkan pemulihan yang berkelanjutan. IHSG bergerak bullish dari kisaran 6.500 menuju hampir 8.000, mencerminkan optimisme investor terhadap fundamental ekonomi yang semakin solid. Kepercayaan pasar juga

didukung oleh arah kebijakan pemerintah yang konsisten dalam menjaga stabilitas makroekonomi, mendorong hilirisasi industri, serta meningkatkan daya saing investasi. Fase ini menandai transisi penting dari fase krisis menuju fondasi pertumbuhan ekonomi jangka menengah yang lebih kuat dan inklusif.

Tabel 4.3. Analisis Data Deskriptif IHSG setelah Covid-19

<i>Analisis Deskriptif Setelah Covid-19</i>	
Mean	7009.388
Standard Error	12.234
Median	6940.008
Standard Deviation	244.063
Range	1128.802
Minimum	6565.728
Maximum	7694.530
Sum	2789736.413
Count	398.000

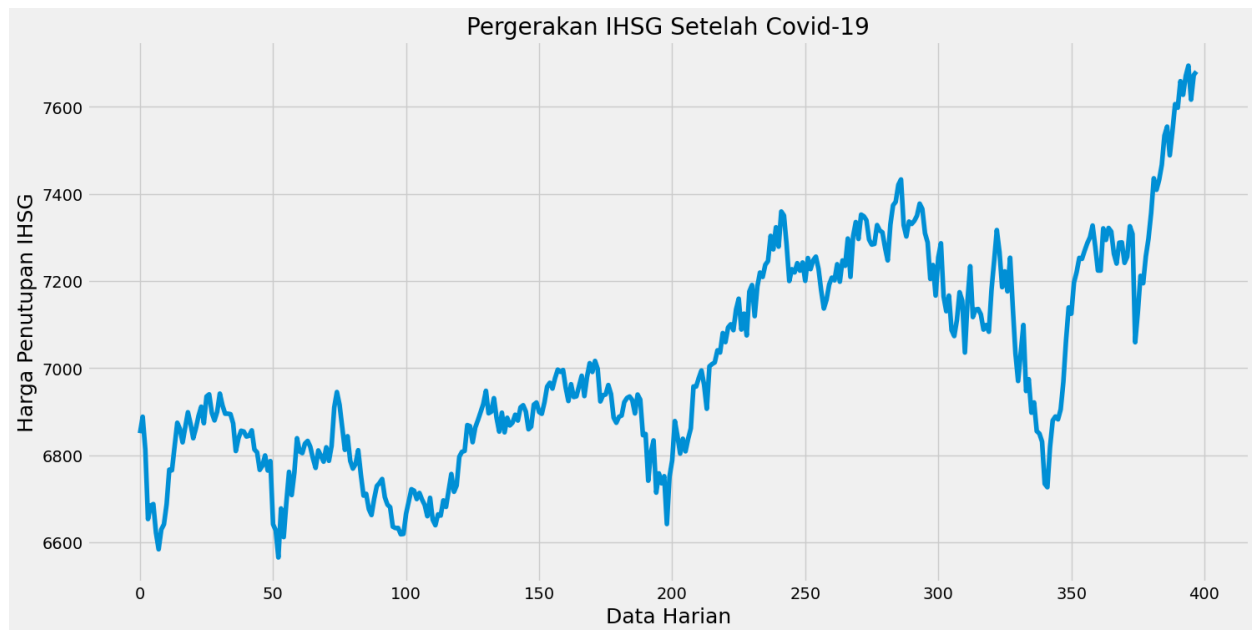
Sumber: Data diolah

Berdasarkan Tabel 4.3, Hasil analisis deskriptif menunjukkan bahwa nilai rata-rata (mean) IHSG selama periode ini adalah 7.009,388, dengan median sebesar 6.940,008. Nilai median yang lebih rendah dari mean menunjukkan bahwa terdapat beberapa hari perdagangan di mana IHSG mengalami kenaikan signifikan yang menarik rata-rata ke atas.

Dalam hal volatilitas, standar deviasi sebesar 244,063 menunjukkan bahwa meskipun IHSG mengalami fluktuasi, volatilitasnya relatif lebih rendah dibandingkan periode pandemi (809,003) tetapi mirip dengan kondisi sebelum pandemi (244,290). Hal ini mengindikasikan bahwa pasar telah kembali lebih stabil dan menunjukkan bahwa penyebaran data tidak terlalu jauh dari nilai rata-rata.

Selama periode ini, IHSG mencapai titik terendahnya di 6.565,728, sedangkan titik tertinggi yang dicapai adalah 7.694,530, dengan rentang harga sebesar 1.128,802. Total nilai indeks selama periode ini mencapai 2.789.736,413, dengan jumlah observasi sebanyak 398 hari perdagangan.

Tidak ditemukan nilai mode, yang berarti tidak ada nilai yang muncul dengan frekuensi dominan dalam data ini.



Gambar 4.4. Pergerakan IHSG setelah Covid. Sumber: Data diolah

Visualisasi dari Gambar 4.4 memberikan gambaran pergerakan IHSG yang menunjukkan pada awal tahun 2023 dengan tren yang relatif fluktuatif, di mana indeks sempat mengalami penurunan di bawah 6.700 sebelum akhirnya kembali mengalami penguatan. Tren naik lebih jelas terlihat setelah pertengahan 2023, di mana IHSG terus mengalami kenaikan bertahap menuju level 7.000 - 7.200.

Pada pertengahan 2024, terjadi peningkatan signifikan yang mendorong IHSG ke 7.694, mencerminkan sentimen positif investor terhadap prospek ekonomi nasional dan global. Kenaikan ini didorong oleh berbagai faktor, termasuk stabilitas makroekonomi, kebijakan suku bunga yang lebih longgar, dan meningkatnya kepercayaan investor terhadap pemulihan ekonomi Indonesia.

Meskipun terdapat beberapa fase koreksi, IHSG secara keseluruhan menunjukkan tren bullish, dengan kenaikan yang lebih stabil dibandingkan periode pandemi, mencerminkan fase pemulihan ekonomi yang lebih matang.

4.2 Hasil Pengujian dan Pembahasan ARIMA-GARCH

4.2.1 Pembagian Data Training dan Testing

Dalam penelitian ini, data harga penutupan IHSG dibagi menjadi dua bagian utama, yaitu data training dan data testing, dengan proporsi 70% untuk training dan 30% untuk testing. Pembagian ini dilakukan untuk memastikan bahwa model memiliki cukup data untuk belajar sebelum diuji pada data yang belum pernah digunakan dalam proses pelatihan. Pembagian data dilakukan untuk tiga periode utama: sebelum Covid-19, pada masa Covid-19, dan setelah Covid-19.

4.2.1.1 Pembagian Data Training dan Testing Sebelum Covid-19

Data sebelum Covid-19 terdiri dari 527 data harian, yang mencakup periode 1 Januari 2018 hingga 1 Maret 2020. Dengan pembagian 70% training dan 30% testing, maka:

- Data training: $0.7 \times 527 = 369$ data
- Data testing: $0.3 \times 527 = 158$ data

Dengan demikian, 369 data pertama digunakan untuk melatih model, sedangkan 158 data sisanya digunakan untuk pengujian.

4.2.1.2 Pembagian Data Training dan Testing pada Masa Covid-19

Data pada masa Covid-19 terdiri dari 693 data harian, yang mencakup periode 2 Maret 2020 hingga 31 Desember 2022. Dengan pembagian 70% training dan 30% testing, maka:

Data training: $0.7 \times 693 = 485$ data

Data testing: $0.3 \times 693 = 208$ data

Dalam hal ini, 485 data pertama digunakan untuk melatih model, sedangkan 208 data terakhir digunakan untuk pengujian.

4.2.1.3 Pembagian Data Training dan Testing Setelah Covid-19

Data setelah Covid-19 terdiri dari 398 data harian, yang mencakup periode 1 Januari 2023 hingga 5 September 2024. Dengan pembagian 70% training dan 30% testing, maka:

Data training: $0.7 \times 398 = 278$ data

Data testing: $0.3 \times 398 = 120$ data

Dengan demikian, 279 data pertama digunakan untuk melatih model, sedangkan 119 data terakhir digunakan untuk pengujian.

Pembagian data ini dilakukan untuk memastikan bahwa model dapat belajar dari pola historis pada setiap periode sebelum diuji dengan data yang belum pernah digunakan dalam proses pelatihan. Hal ini bertujuan untuk mengukur sejauh mana model dapat menggeneralisasi pola pergerakan IHSG dalam setiap periode ekonomi yang berbeda yang tercermin pada periode sebelum Covid-19, masa Covid-19 dan setelah Covid-19.

4.2.1 Uji Stasioneritas Data

Dalam model ARIMA-GARCH, asumsi utama yang harus dipenuhi adalah bahwa data yang digunakan bersifat stasioner. Stasioneritas dalam deret waktu berarti bahwa statistik seperti mean, varians, dan autokorelasi tetap konstan sepanjang waktu. Oleh karena itu, sebelum melakukan pemodelan, langkah awal yang perlu dilakukan adalah menguji stasioneritas data untuk memastikan bahwa data telah memenuhi syarat untuk dianalisis lebih lanjut.

Untuk menguji apakah data yang digunakan dalam penelitian ini bersifat stasioner, dilakukan pengujian Augmented Dickey-Fuller (ADF). Pengujian ini bertujuan untuk mengidentifikasi keberadaan akar unit dalam data, yang mengindikasikan adanya tren atau perubahan pola dalam mean dari waktu ke waktu. Hipotesis yang digunakan dalam uji ADF adalah sebagai berikut:

- H_0 (Hipotesis nol): Data memiliki akar unit (tidak stasioner).
- H_1 (Hipotesis alternatif): Data tidak memiliki akar unit (stasioner).

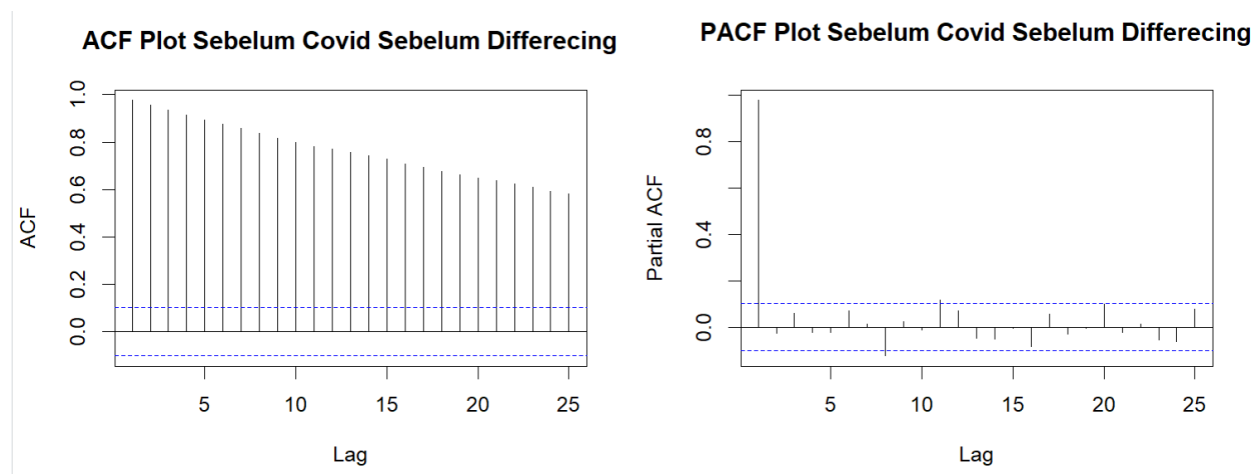
Jika nilai p-value dari uji ADF lebih kecil dari tingkat signifikansi (misalnya 5% atau 1%), maka hipotesis nol ditolak, yang berarti data bersifat stasioner. Sebaliknya, jika p-value lebih besar dari tingkat signifikansi, maka data tidak stasioner dan perlu dilakukan transformasi lebih lanjut, seperti differencing.

Selain pengujian ADF, dilakukan juga analisis grafis dengan menggunakan plot Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF).

- Plot ACF digunakan untuk melihat bagaimana keterkaitan antara suatu nilai dalam deret waktu dengan nilai-nilai sebelumnya pada berbagai lag. Jika grafik ACF menunjukkan autokorelasi yang perlahan menurun, maka data kemungkinan tidak stasioner.
- Plot PACF membantu dalam mengidentifikasi order dari model ARIMA dengan menunjukkan korelasi parsial antara suatu nilai dengan nilai sebelumnya setelah mengontrol efek dari lag di antaranya.

4.2.2.1 Hasil Uji Stasioneritas Sebelum Covid-19

Pengujian stasioneritas untuk periode sebelum COVID-19 (1 Januari 2018 – 1 Maret 2020) dilakukan menggunakan tiga metode, yaitu visualisasi Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF), serta uji Augmented Dickey-Fuller (ADF).



Gambar 4.5. ACF dan PACF pada Data Sebelum Covid-19 sebelum Differencing.

Pada Gambar 4.5, plot ACF menunjukkan bahwa autokorelasi mengalami penurunan secara perlahan seiring bertambahnya lag, tanpa ada cutoff yang jelas. Pola ini mengindikasikan bahwa data kemungkinan mengandung akar unit dan tidak bersifat stasioner. Sementara itu, plot PACF menunjukkan cutoff yang signifikan pada lag pertama, yang dapat mengindikasikan adanya

struktur autoregresif dalam data. Namun, karena ACF masih menunjukkan pola decay yang lambat, diperlukan konfirmasi lebih lanjut dengan uji formal seperti ADF.

Tabel 4.4. Hasil Pengujian ADF test pada Data Sebelum Covid-19

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	-0.0266	0.636
[2,]	1	0.0534	0.659
[3,]	2	0.0209	0.65
[4,]	3	-0.0303	0.635
[5,]	4	-0.0633	0.626
[6,]	5	-0.0447	0.631
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	-1.94	0.347
[2,]	1	-1.95	0.347
[3,]	2	-1.83	0.395
[4,]	3	-1.86	0.384
[5,]	4	-1.9	0.366
[6,]	5	-1.75	-1.75
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-2	0.578

[2,]	1	-1.99	0.578
[3,]	2	-1.88	0.626
[4,]	3	-1.92	0.611
[5,]	4	-1.97	0.59
[6,]	5	-1.82	0.654

Uji ADF dilakukan dengan tiga skenario berbeda:

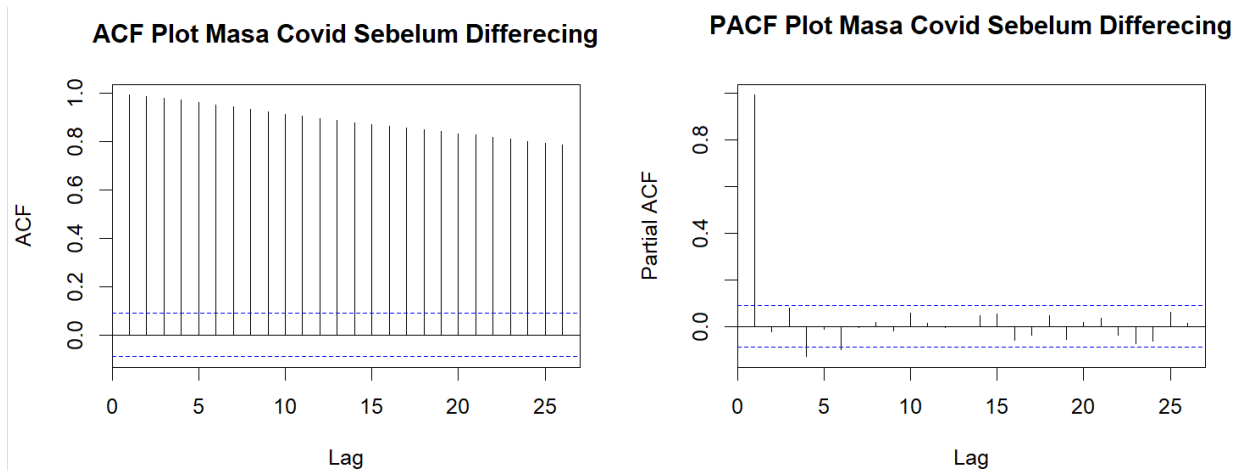
- Type 1 (tanpa drift dan tanpa tren): Nilai ADF berkisar antara -0.0266 hingga -0.0633 dengan p-value lebih besar dari 0.05, menunjukkan bahwa hipotesis nol (H_0 : data memiliki akar unit/tidak stasioner) gagal ditolak.
- Type 2 (dengan drift, tanpa tren): Nilai ADF berkisar antara -1.75 hingga -1.95 dengan p-value masih di atas 0.05, sehingga hipotesis nol tetap gagal ditolak.
- Type 3 (dengan drift dan tren): Nilai ADF berkisar antara -1.82 hingga -2.00, dan p-value masih jauh di atas 0.05, yang berarti data tetap tidak stasioner meskipun mempertimbangkan adanya tren.

Dari ketiga pengujian ADF di atas, tidak ada satu pun yang menunjukkan nilai p-value di bawah 0.05, sehingga dapat disimpulkan bahwa data sebelum COVID-19 tidak stasioner dalam level.

Berdasarkan hasil visualisasi ACF, PACF, serta uji ADF, dapat disimpulkan bahwa data sebelum COVID-19 tidak bersifat stasioner. Oleh karena itu, diperlukan transformasi lebih lanjut, seperti differencing, untuk menghilangkan akar unit dan mencapai stasioneritas. Pada bagian berikutnya, akan dilakukan differencing pertama ($d=1$) dan pengujian ulang menggunakan ACF, PACF, serta ADF untuk memastikan apakah data telah mencapai stasioneritas setelah transformasi tersebut.

4.2.2.2 Hasil Uji Stasioneritas pada Masa Covid-19

Pengujian stasioneritas untuk periode masa COVID-19 dilakukan menggunakan visualisasi Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF), serta uji Augmented Dickey-Fuller (ADF).



Gambar 4.6. ACF dan PACF pada Data pada Masa Covid-19 sebelum Differencing

Pada Gambar 4.6, plot ACF menunjukkan bahwa autokorelasi mengalami penurunan secara perlahan seiring bertambahnya lag, tanpa adanya cutoff yang jelas. Hal ini menunjukkan bahwa data kemungkinan memiliki akar unit dan tidak bersifat stasioner. Plot PACF menunjukkan cutoff yang signifikan pada lag pertama, yang dapat mengindikasikan adanya struktur autoregresif dalam data. Namun, karena ACF masih menunjukkan pola decay yang lambat, diperlukan konfirmasi lebih lanjut dengan uji formal seperti ADF.

Tabel 4.5. Hasil Pengujian ADF test pada Data pada Masa Covid-19

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	0.872	0.895
[2,]	1	0.765	0.864
[3,]	2	0.789	0.871

[4,]	3	0.871	0.812
[5,]	4	0.672	0.837
[6,]	5	0.817	0.879
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	-0.68	0.815
[2,]	1	-0.639	0.829
[3,]	2	-0.472	0.888
[4,]	3	-0.747	0.791
[5,]	4	-0.8	0.772
[6,]	5	-1.165	0.643
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-3.2	0.0876
[2,]	1	-3.53	0.0393
[3,]	2	-3.66	0.026
[4,]	3	-4.11	-4.11
[5,]	4	-3.84	0.0167
[6,]	5	-3.54	0.0377

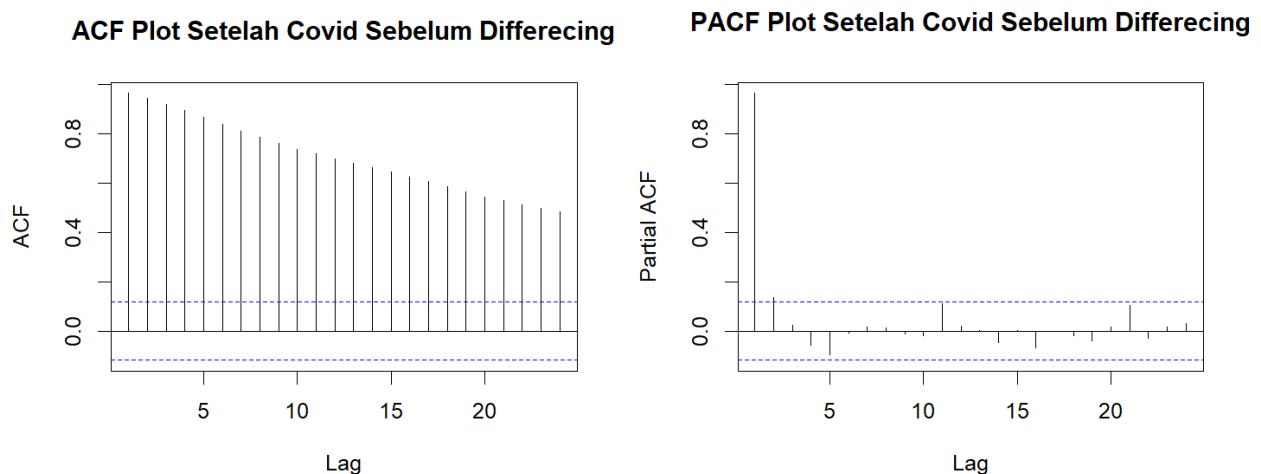
Pengujian formal dilakukan menggunakan uji ADF dengan tiga jenis spesifikasi:

- Type 1 (Tanpa Drift, Tanpa Tren): Hasil uji ADF menunjukkan nilai 0.872 dengan p-value 0.895, yang lebih besar dari tingkat signifikansi 0.05. Ini berarti hipotesis nol tidak dapat ditolak, sehingga data tidak stasioner.
- Type 2 (Dengan Drift, Tanpa Tren): Hasil uji ADF menunjukkan nilai -0.680 dengan p-value 0.815, yang juga lebih besar dari tingkat signifikansi 0.05. Dengan demikian, data tetap tidak stasioner meskipun mempertimbangkan adanya drift.
- Type 3 (Dengan Drift dan Tren): Hasil uji ADF pada beberapa lag menunjukkan hasil yang lebih baik. Pada lag 3, nilai ADF adalah -4.11 dengan p-value 0.0100, yang lebih kecil dari 0.05, sehingga hipotesis nol ditolak. Ini menunjukkan bahwa dalam model dengan drift dan tren, data bisa dianggap stasioner.

Hasil pengujian menunjukkan bahwa data tidak stasioner dalam kondisi tanpa tren, tetapi dapat dianggap stasioner jika tren diperhitungkan. Namun, dalam pemodelan ARIMA-GARCH, stasioneritas dalam bentuk tanpa tren lebih diutamakan. Oleh karena itu, untuk memastikan data memenuhi syarat stasioneritas, perlu dilakukan transformasi differencing dan uji ulang menggunakan ACF, PACF, serta ADF.

4.2.2.3 Hasil Uji Stasioneritas Setelah Covid-19

Pengujian stasioneritas untuk periode setelah COVID-19 dilakukan menggunakan visualisasi Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF), serta uji Augmented Dickey-Fuller (ADF).



Gambar 4.7. ACF dan PACF pada Data Setelah Covid-19 sebelum Differencing

Pada Gambar 4.7, plot ACF menunjukkan bahwa autokorelasi mengalami penurunan secara perlahan seiring bertambahnya lag, tanpa adanya cutoff yang jelas. Hal ini menunjukkan bahwa data kemungkinan memiliki akar unit dan tidak bersifat stasioner. Plot PACF menunjukkan cutoff yang signifikan pada lag pertama, yang dapat mengindikasikan adanya struktur autoregresif dalam data. Namun, karena ACF masih menunjukkan pola decay yang lambat, diperlukan konfirmasi lebih lanjut dengan uji formal seperti ADF.

Tabel 4.6. Hasil Pengujian ADF test pada Data Setelah Covid-19

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	0.649	0.831
[2,]	1	0.719	0.851
[3,]	2	0.863	0.892
[4,]	3	1.134	0.931
[5,]	4	1.08	0.924
[6,]	5	1.096	0.926
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	-1.202	0.626
[2,]	1	-0.81	0.764
[3,]	2	-0.721	0.795
[4,]	3	-0.897	0.667
[5,]	4	-1.085	0.667

[6,]	5	-1.015	0.692
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-2.86	0.213
[2,]	1	-2.56	0.342
[3,]	2	-2.29	0.454
[4,]	3	-2.06	0.551
[5,]	4	-2.24	0.475
[6,]	5	-2.18	0.5

Pengujian formal dilakukan menggunakan uji ADF dengan tiga jenis spesifikasi:

- Type 1 (Tanpa Drift, Tanpa Tren): Hasil uji ADF menunjukkan nilai 0.649 dengan p-value 0.831, yang lebih besar dari tingkat signifikansi 0.05. Ini berarti hipotesis nol tidak dapat ditolak, sehingga data tidak stasioner.
- Type 2 (Dengan Drift, Tanpa Tren): Hasil uji ADF menunjukkan nilai -1.202 dengan p-value 0.626, yang juga lebih besar dari tingkat signifikansi 0.05. Dengan demikian, data tetap tidak stasioner meskipun mempertimbangkan adanya drift.
- Type 3 (Dengan Drift dan Tren): Hasil uji ADF menunjukkan bahwa pada semua lag, p-value tetap lebih besar dari 0.05. Nilai ADF terbaik diperoleh pada lag 0 dengan ADF -2.86 dan p-value 0.213, yang masih lebih besar dari 0.05. Ini menunjukkan bahwa bahkan dengan mempertimbangkan tren, data tetap tidak stasioner.

Hasil pengujian menunjukkan bahwa data tidak stasioner dalam semua spesifikasi, baik tanpa tren maupun dengan tren. Oleh karena itu, untuk memastikan data memenuhi syarat stasioneritas dalam pemodelan ARIMA-GARCH, perlu dilakukan transformasi differencing dan uji ulang menggunakan ACF, PACF, serta ADF.

Berdasarkan hasil pengujian stasioneritas menggunakan ACF, PACF, dan uji Augmented Dickey-Fuller (ADF) pada tiga periode data sebelum COVID-19, masa COVID-19, dan setelah COVID-19 dapat disimpulkan bahwa data pada ketiga periode tersebut bersifat non-stasioner. Hal ini ditunjukkan oleh pola ACF yang mengalami decay perlahan tanpa cutoff yang jelas, PACF dengan cutoff signifikan pada lag pertama, serta hasil uji ADF yang menunjukkan p-value lebih besar dari 0.05 dalam semua spesifikasi, sehingga hipotesis nol tidak dapat ditolak. Dengan demikian, data masih memiliki akar unit dan perlu dilakukan transformasi agar menjadi stasioner sebelum digunakan dalam pemodelan ARIMA-GARCH. Oleh karena itu, differencing akan diterapkan untuk menghilangkan pola non-stasioneritas dalam data, dan selanjutnya dilakukan pengujian ulang menggunakan ACF, PACF, serta uji ADF setelah differencing.

4.2.1 Transformasi Data dengan Differencing

Berdasarkan hasil pengujian sebelumnya, dapat disimpulkan bahwa data pada ketiga periode memiliki sifat non-stasioner, yang ditunjukkan oleh:

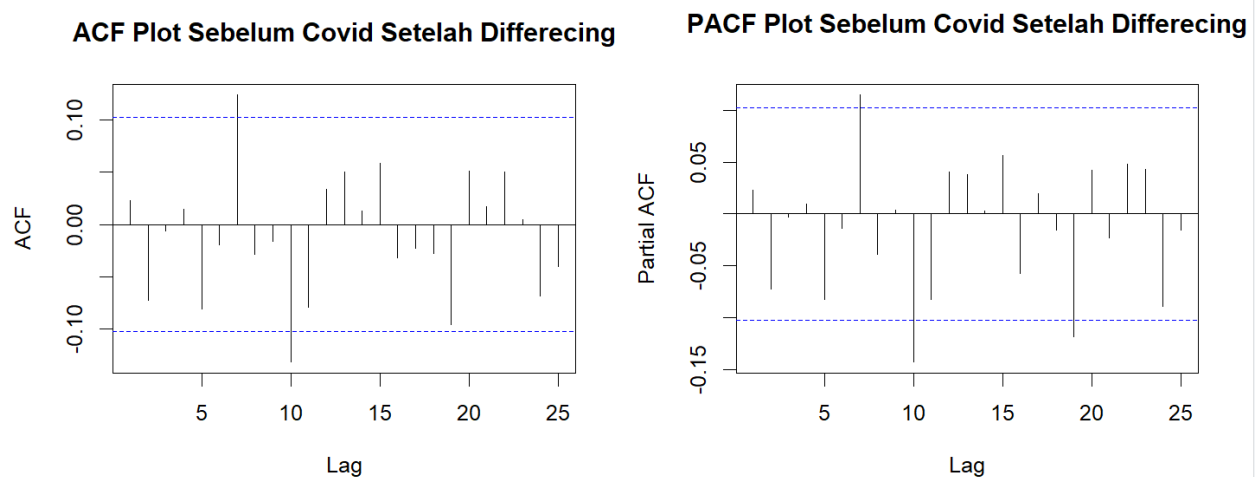
- Plot ACF yang mengalami decay perlahan tanpa cutoff yang jelas,
- Plot PACF yang memiliki cutoff pada lag pertama,
- Uji ADF yang memberikan p-value lebih besar dari 0.05, sehingga hipotesis nol (data memiliki akar unit) tidak dapat ditolak.

Dalam model ARIMA-GARCH, stasioneritas adalah syarat utama agar model dapat diestimasi dengan baik. Oleh karena itu, untuk memastikan data memenuhi asumsi stasioneritas, differencing akan diterapkan. Differencing dilakukan dengan mengubah data menjadi selisih dari nilai sebelumnya, yang dapat membantu menghilangkan tren atau pola non-stasioneritas dalam data.

Selanjutnya, uji ulang terhadap data yang telah mengalami differencing akan dilakukan dengan menggunakan visualisasi ACF, PACF, serta uji ADF pada masing-masing periode. Berikut adalah hasil setelah differencing untuk periode sebelum COVID-19:

4.2.3.1 Analisis Stasioneritas Data Sebelum COVID-19 Setelah Differencing

Untuk memastikan bahwa data yang digunakan dalam pemodelan ARIMA-GARCH bersifat stasioner, dilakukan uji stasioneritas dengan metode Augmented Dickey-Fuller (ADF) setelah proses differencing. Berikut adalah analisis berdasarkan plot ACF dan PACF serta hasil uji ADF setelah differencing untuk periode sebelum COVID-19.



Gambar 4.8. ACF dan PACF Sebelum Covid-19 Setelah Differencing

Gambar 4.8, memperlihatkan setelah differencing autokorelasi tidak lagi memiliki pola decay perlahan seperti sebelum differencing, melainkan berfluktuasi di sekitar nol tanpa pola yang jelas. Hal ini mengindikasikan bahwa tren yang sebelumnya ada telah dieliminasi. Sementara itu, plot PACF menunjukkan cutoff pada beberapa lag awal, yang merupakan karakteristik data yang telah berhasil ditransformasi menjadi stasioner.

Tabel 4.7. Hasil Pengujian ADF test pada Data Sebelum Covid-19 Setelah Differencing

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	-18.73	0.01
[2,]	1	-14.31	0.01
[3,]	2	-11.43	0.01

[4,]	3	-9.68	0.01
[5,]	4	-9.34	0.01
[6,]	5	-8.46	0.01
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	18.71	0.01
[2,]	1	-14.3	0.01
[3,]	2	-11.42	0.01
[4,]	3	-9.67	0.01
[5,]	4	-9.33	0.01
[6,]	5	-8.45	0.01
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-18.69	0.01
[2,]	1	-14.29	0.01
[3,]	2	-11.43	0.01
[4,]	3	-9.68	0.01
[5,]	4	-9.35	0.01
[6,]	5	-8.48	0.01

Uji ADF dilakukan dengan tiga spesifikasi berikut:

1. Tanpa drift dan tanpa tren (Type 1)

Hasil uji ADF menunjukkan nilai ADF yang sangat negatif, yaitu -18.73 pada lag 0 dan tetap signifikan pada seluruh lag yang diuji. P-value sebesar 0.01 menunjukkan bahwa hipotesis nol tentang adanya akar unit dapat ditolak.

2. Dengan drift tetapi tanpa tren (Type 2)

Pada spesifikasi ini, nilai ADF tetap sangat negatif dengan nilai -18.71 pada lag 0 dan serupa pada lag lainnya. P-value tetap 0.01, yang berarti data telah menjadi stasioner setelah differencing.

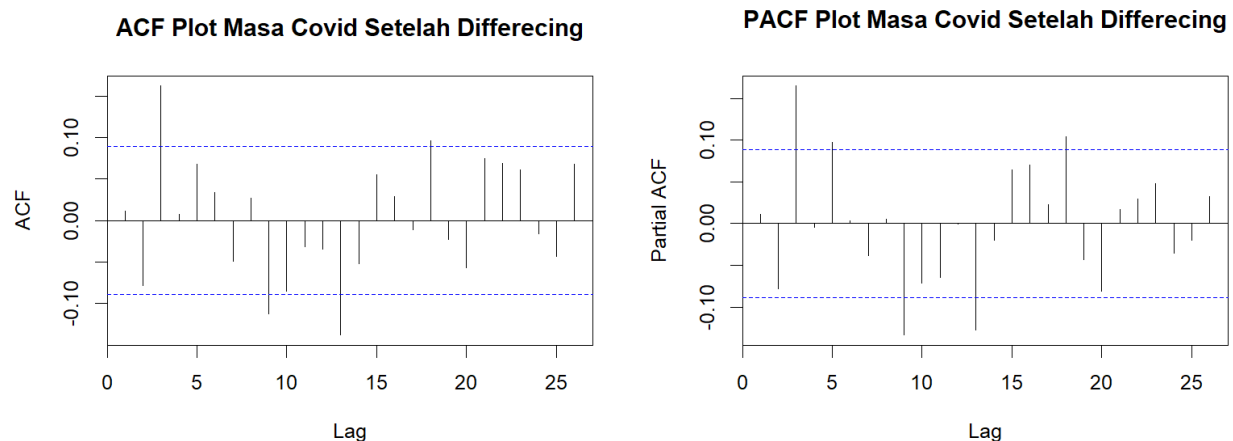
3. Dengan drift dan tren (Type 3)

Hasil ADF menunjukkan nilai yang serupa, dengan ADF sebesar -18.69 pada lag 0 dan tetap signifikan di seluruh lag yang diuji. P-value 0.01 memastikan bahwa data setelah differencing telah menjadi stasioner.

Berdasarkan analisis ACF, PACF, dan uji ADF, dapat disimpulkan bahwa data sebelum COVID-19 telah menjadi stasioner setelah differencing, ditunjukkan oleh pola acak dalam ACF dan PACF serta hasil ADF yang menunjukkan nilai sangat negatif dengan p-value jauh di bawah 0.05. Dengan demikian, data ini siap untuk digunakan dalam pemodelan ARIMA-GARCH.

4.2.3.2 Analisis Stasioneritas Data Masa COVID-19 Setelah Differencing

Untuk memastikan bahwa data pada periode masa COVID-19 telah menjadi stasioner, dilakukan analisis melalui plot ACF dan PACF serta uji Augmented Dickey-Fuller (ADF) setelah differencing. Berikut adalah hasil analisisnya.



Gambar 4.9. ACF dan PACF pada Masa Covid-19 Setelah Differencing

Pada gambar 4.9, ACF setelah differencing menunjukkan, autokorelasi menunjukkan pola yang tidak lagi mengalami decay secara perlahan, melainkan berfluktuasi di sekitar nol tanpa pola tertentu. Hal ini mengindikasikan bahwa data telah kehilangan tren dan kemungkinan besar telah menjadi stasioner. Sementara itu, plot PACF menunjukkan adanya cutoff pada beberapa lag awal, yang merupakan indikasi bahwa komponen autoregresif masih berperan dalam data tetapi dalam kondisi yang lebih stasioner dibandingkan sebelum differencing.

Tabel 4.8. Hasil Pengujian ADF test pada Data pada Masa Covid-19 Setelah Differencing

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	-21.74	0.01
[2,]	1	-16.85	0.01
[3,]	2	-11.26	0.01
[4,]	3	-9.85	0.01
[5,]	4	-8.14	0.01
[6,]	5	-7.53	0.01
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	-21.75	0.01
[2,]	1	-16.86	0.01
[3,]	2	-11.28	0.01
[4,]	3	-9.87	0.01
[5,]	4	-8.19	0.01

[6,]	5	-7.58	0.01
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-21.76	0.01
[2,]	1	-16.9	0.01
[3,]	2	-11.31	0.01
[4,]	3	-9.89	0.01
[5,]	4	-8.16	0.01
[6,]	5	-7.56	0.01

Uji ADF dilakukan dengan tiga spesifikasi sebagai berikut:

1. Tanpa drift dan tanpa tren (Type 1)

Nilai ADF sebesar -21.74 pada lag 0 menunjukkan hasil yang lebih negatif dibandingkan periode sebelum COVID-19 setelah differencing, dengan p-value 0.01, yang berarti hipotesis nol tentang adanya akar unit dapat ditolak.

2. Dengan drift tetapi tanpa tren (Type 2)

Pada spesifikasi ini, nilai ADF sebesar -21.75 pada lag 0 dan tetap negatif secara signifikan di seluruh lag yang diuji. P-value tetap 0.01, menunjukkan bahwa data setelah differencing sudah stasioner.

3. Dengan drift dan tren (Type 3)

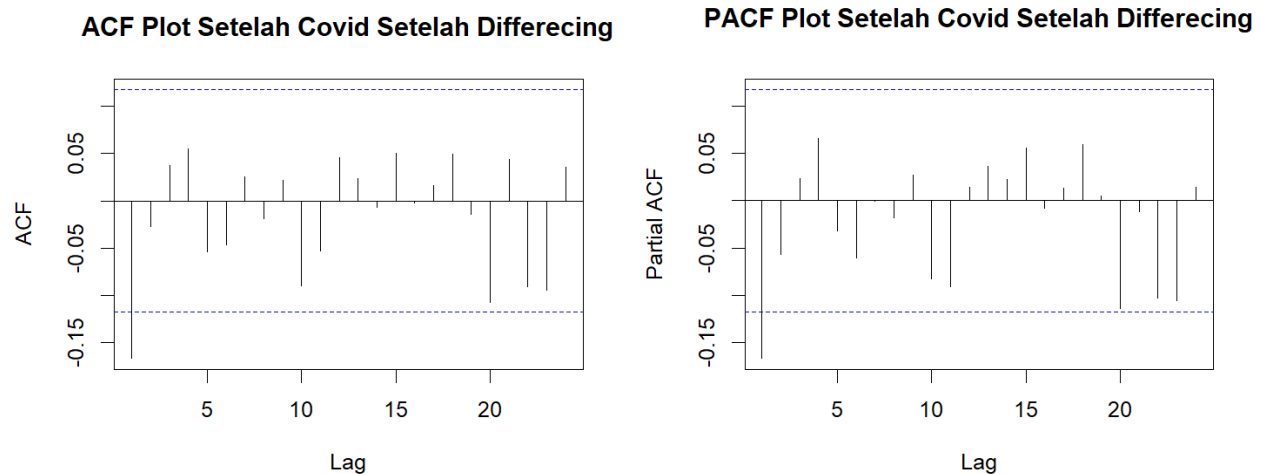
Nilai ADF sebesar -21.76 pada lag 0 dan tetap signifikan di seluruh lag menunjukkan bahwa data setelah differencing telah menjadi stasioner, dengan p-value 0.01.

Berdasarkan analisis ACF, PACF, dan uji ADF, data pada periode masa COVID-19 telah menjadi stasioner setelah differencing, sebagaimana ditunjukkan oleh pola acak dalam ACF dan PACF

serta nilai ADF yang sangat negatif dengan p-value jauh di bawah 0.05. Dengan demikian, data ini siap digunakan dalam pemodelan ARIMA-GARCH.

4.2.3.3 Analisis Stasioneritas Data Setelah COVID-19 Setelah *Differencing*

Untuk memastikan bahwa data pada periode setelah COVID-19 telah menjadi stasioner, dilakukan analisis melalui plot ACF dan PACF serta uji Augmented Dickey-Fuller (ADF) setelah differencing. Berikut adalah hasil analisisnya.



Gambar 4.10. ACF dan PACF Setelah Covid-19 Setelah Differencing

Pada gambar 4.10, ACF setelah differencing, autokorelasi tidak menunjukkan pola yang jelas, dengan sebagian besar nilai jatuh dalam batas signifikansi, yang mengindikasikan bahwa data telah menjadi lebih stasioner. Sementara itu, pada plot PACF, terdapat cutoff pada beberapa lag awal, yang menunjukkan kemungkinan keberadaan komponen autoregresif yang signifikan. Namun, pola acak dalam ACF dan PACF mengonfirmasi bahwa tren dan musiman dalam data telah teratasi setelah proses differencing.

Tabel 4.9. Hasil Pengujian ADF test pada Data Setelah Covid-19 Setelah Differencing

Type 1: no drift no trend			
	lag	ADF	p.value
[1,]	0	-19.59	0.01

[2,]	1	-13.31	0.01
[3,]	2	-10.5	0.01
[4,]	3	-8.23	0.01
[5,]	4	-7.51	0.01
[6,]	5	-7.4	0.01
Type 2: with drift no trend			
	lag	ADF	p.value
[1,]	0	-19.59	0.01
[2,]	1	-13.33	0.01
[3,]	2	-10.56	0.01
[4,]	3	-8.31	0.01
[5,]	4	-7.59	0.01
[6,]	5	-7.51	0.01
Type 3: with drift and trend			
	lag	ADF	p.value
[1,]	0	-19.63	0.01
[2,]	1	-13.37	0.01
[3,]	2	-10.56	0.01
[4,]	3	-8.3	0.01
[5,]	4	-7.59	0.01
[6,]	5	-7.51	0.01

Uji ADF dilakukan dengan tiga spesifikasi sebagai berikut:

1. Tanpa drift dan tanpa tren (Type 1)

Nilai ADF sebesar -19.59 pada lag 0 menunjukkan bahwa hipotesis nol tentang adanya akar unit dapat ditolak, dengan p-value 0.01, yang mengindikasikan bahwa data telah stasioner.

2. Dengan drift tetapi tanpa tren (Type 2)

Hasil uji dengan drift tetap menunjukkan nilai ADF yang signifikan (-19.59 pada lag 0), dengan p-value 0.01, sehingga dapat disimpulkan bahwa data setelah differencing tetap stasioner di seluruh lag.

3. Dengan drift dan tren (Type 3)

Nilai ADF sebesar -19.63 pada lag 0 dan tetap signifikan di semua lag menunjukkan bahwa data sudah dalam kondisi stasioner dengan p-value 0.01.

Berdasarkan hasil ACF, PACF, dan uji ADF, data pada periode setelah COVID-19 telah menjadi stasioner setelah differencing, dengan tidak adanya pola yang jelas dalam ACF dan PACF serta nilai ADF yang sangat negatif dan signifikan. Dengan data yang sudah stasioner pada ketiga periode pengujian, langkah selanjutnya adalah membangun model ARIMA-GARCH, di mana pemilihan parameter ARIMA (p,d,q) akan didasarkan pada pola cutoff dalam PACF dan decay dalam ACF, sementara model GARCH akan ditentukan berdasarkan volatilitas dari residual ARIMA.

4.2.1 Penentuan Nilai p dan q Berdasarkan ACF, PACF, dan Uji ARIMA

Setelah memastikan bahwa data telah mencapai stasioneritas melalui differencing, langkah berikutnya adalah menentukan parameter p (order AR) dan q (order MA) untuk membangun model ARIMA yang optimal. Nilai ini ditentukan berdasarkan pola Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF), serta dikonfirmasi melalui uji beberapa kombinasi model ARIMA.

Autocorrelation Function (ACF) mengukur korelasi total antara observasi pada berbagai lag tanpa mengendalikan pengaruh lag-lag di antaranya, sehingga memberikan gambaran menyeluruh mengenai pola autokorelasi dalam data. Jika plot ACF menunjukkan cut-off secara tiba-tiba setelah lag ke- q , hal ini menandakan bahwa pengaruh autokorelasi berkurang drastis di luar lag tersebut, yang mengindikasikan bahwa model Moving Average (MA(q)) dapat diterapkan. Di sisi lain, Partial Autocorrelation Function (PACF) mengukur korelasi antara observasi saat ini dan observasi pada lag tertentu setelah menghilangkan pengaruh lag-lag perantara, sehingga memberikan indikasi yang lebih jelas mengenai hubungan langsung. Oleh karena itu, apabila plot PACF menunjukkan cut-off setelah lag ke- p , hal ini biasanya mengindikasikan bahwa model Autoregressive (AR(p)) adalah pilihan yang tepat (Halwa, 2023).

Deviana et al. (2021) berpendapat analisis simultan dari kedua plot ini sangat penting untuk menentukan model yang paling sesuai. Jika PACF mengalami cutoff secara tajam setelah lag tertentu sementara ACF menurun secara bertahap, maka model AR lebih cocok, karena pola tersebut mencerminkan adanya hubungan langsung yang signifikan pada lag tertentu. Sebaliknya, jika ACF menunjukkan cut-off pada lag tertentu dan PACF menurun secara bertahap, maka model MA lebih sesuai. Apabila kedua fungsi tersebut menunjukkan peluruhan secara bertahap, hal ini mengindikasikan adanya kombinasi efek AR dan MA, sehingga model ARMA harus dipertimbangkan. Berikut hasil pengujian untuk masing-masing periode waktu.

4.2.4.1 Periode Sebelum Covid-19

Dari hasil plot ACF dan PACF setelah differencing, dilakukan pengujian beberapa model ARIMA untuk menentukan kombinasi (p,d,q) yang paling sesuai. Model yang diuji antara lain:

- ARIMA (0,1,0)
- ARIMA (1,1,1)
- ARIMA (1,1,0)
- ARIMA (0,1,1)
- ARIMA (2,1,2)
- ARIMA (1,1,2)
- ARIMA (2,1,1)

- ARIMA (0,1,2)
- ARIMA (2,1,0)

Berdasarkan Gambar 4.8, plot ACF dan PACF sebelum Covid setelah differencing, pemilihan model ARIMA dilakukan dengan mempertimbangkan pola cutoff dan decay pada kedua grafik. Model ARIMA (0,1,0) digunakan jika data setelah differencing menyerupai white noise tanpa pola autokorelasi yang signifikan. ARIMA (1,1,0) dipilih jika PACF menunjukkan cutoff di lag 1 sementara ACF mengalami decay bertahap, sedangkan ARIMA (0,1,1) sesuai jika ACF memiliki cutoff di lag 1 sementara PACF lebih menyebar. Jika baik ACF maupun PACF menunjukkan cutoff pada lag 2, model ARIMA (2,1,2) lebih cocok, sementara ARIMA (1,1,2) dipilih jika PACF cutoff di lag 1 dan ACF di lag 2. ARIMA (2,1,1) lebih relevan jika PACF menunjukkan cutoff di lag 2 sementara ACF cutoff di lag 1, sedangkan ARIMA (0,1,2) cocok untuk ACF dengan cutoff di lag 2 tanpa pola yang jelas pada PACF. Terakhir, ARIMA (2,1,0) digunakan jika PACF cutoff di lag 2 sementara ACF mengalami decay tanpa cutoff yang tegas. Model terbaik akan ditentukan berdasarkan kriteria AIC/BIC serta validasi residual untuk memastikan white noise.

4.2.4.2 Periode Masa Covid-19

Pada periode ini, analisis ACF dan PACF menunjukkan pola yang berbeda dibandingkan sebelum COVID, dengan indikasi bahwa model dengan nilai p yang lebih besar mungkin lebih sesuai. Oleh karena itu, model yang diuji meliputi:

- ARIMA (5,1,0)
- ARIMA (4,1,0)
- ARIMA (3,1,0)
- ARIMA (2,1,0)
- ARIMA (1,1,0)
- ARIMA (0,1,0)
- ARIMA (5,1,1)
- ARIMA (4,1,1)

- ARIMA (3,1,1)
- ARIMA (2,1,1)
- ARIMA (1,1,1)
- ARIMA (0,1,1)
- ARIMA (5,1,2)
- ARIMA (4,1,2)
- ARIMA (3,1,2)
- ARIMA (2,1,2)
- ARIMA (1,1,2)
- ARIMA (0,1,2)

Berdasarkan Gambar 4.9, plot ACF dan PACF setelah differencing pada masa Covid, pemilihan model ARIMA didasarkan pada pola cutoff dan decay yang muncul. Model ARIMA (0,1,0) dipilih jika tidak ada autokorelasi signifikan, menunjukkan data mendekati white noise. Model ARIMA (p,1,0) dengan $p = 1$ hingga 5 digunakan jika PACF menunjukkan cutoff pada lag tertentu sementara ACF mengalami decay bertahap, di mana semakin besar p menyesuaikan jumlah lag signifikan. ARIMA (0,1,1) dipilih jika ACF cutoff di lag 1 sementara PACF lebih menyebar, sedangkan ARIMA (p,1,1) digunakan ketika PACF cutoff di lag p dan ACF memiliki pola decay. Model ARIMA (p,1,2) lebih cocok jika PACF cutoff di lag p dan ACF di lag 2, menyesuaikan kompleksitas pola autokorelasi yang lebih panjang. Pemilihan model ini bertujuan untuk menangkap pola perubahan data selama masa Covid yang kemungkinan memiliki fluktuasi berbeda dibandingkan periode sebelumnya. Model optimal akan dipilih dengan mempertimbangkan AIC/BIC serta analisis residual untuk memastikan tidak adanya pola sistematis dalam error.

4.2.4.3 Periode Setelah Covid-19

Untuk periode setelah COVID, pola ACF dan PACF kembali mengalami perubahan. Oleh karena itu, model yang diuji mencakup:

- ARIMA (3,1,0)
- ARIMA (2,1,0)
- ARIMA (1,1,0)
- ARIMA (0,1,0)
- ARIMA (3,1,1)
- ARIMA (2,1,1)
- ARIMA (1,1,1)
- ARIMA (0,1,1)
- ARIMA (3,1,2)
- ARIMA (2,1,2)
- ARIMA (1,1,2)
- ARIMA (0,1,2)

Berdasarkan Gambar 4.10, plot ACF dan PACF setelah differencing pada periode setelah Covid, pemilihan model ARIMA didasarkan pada pola autokorelasi yang tersisa. Model ARIMA (0,1,0) digunakan jika ACF dan PACF tidak menunjukkan autokorelasi signifikan, menandakan bahwa data bersifat white noise setelah differencing. Model ARIMA (p,1,0) dengan p antara 1 hingga 3 dipilih karena PACF menunjukkan cutoff pada lag tertentu sementara ACF mengalami peluruhan bertahap, yang mengindikasikan bahwa komponen autoregresif (AR) berperan dalam model. Model ARIMA (0,1,1) dipertimbangkan jika ACF mengalami cutoff pada lag pertama, menunjukkan bahwa efek moving average (MA) dominan dalam menangkap pola data. Kombinasi model ARIMA (p,1,1) dan ARIMA (p,1,2) dengan p antara 1 hingga 3 digunakan jika PACF mengalami cutoff di lag p sementara ACF mengalami pola peluruhan atau cutoff di lag 2, menunjukkan bahwa kombinasi komponen AR dan MA diperlukan untuk menangkap pola fluktuatif dalam data setelah pandemi. Model dengan performa terbaik akan dipilih berdasarkan AIC/BIC, dan dilakukan pengecekan residual untuk memastikan keacakan serta validitas model.

Berdasarkan pengujian di atas, model ARIMA yang paling optimal untuk masing-masing periode akan ditentukan dengan melihat nilai AIC/BIC, uji diagnostik residual, serta kriteria keakuratan prediksi. Jika ditemukan pola heteroskedastisitas dalam residual, maka akan dipertimbangkan penggunaan model ARIMA-GARCH untuk menangani volatilitas yang tinggi.

4.2.1 Pemilihan Model ARIMA Terbaik

Setelah melakukan differencing untuk mencapai stasioneritas, langkah berikutnya adalah menentukan model Autoregressive Integrated Moving Average (ARIMA) yang paling sesuai untuk setiap periode: sebelum COVID (1 Januari 2018 – 1 Maret 2020), masa COVID (2 Maret 2020 – 31 Desember 2022), dan setelah COVID (1 Januari 2023 – 5 September 2024). Pemilihan model ARIMA dilakukan dengan menguji berbagai kombinasi nilai p (autoregressive/AR) dan q (moving average/MA), serta mempertimbangkan nilai Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC). Model terbaik dipilih berdasarkan nilai AIC terendah, dengan syarat bahwa minimal salah satu komponen AR atau MA tidak bernilai nol. Berikut ini adalah hasil pengujian model ARIMA untuk masing-masing periode, yang kemudian dianalisis lebih lanjut untuk memastikan kesesuaian model sebelum digunakan dalam peramalan.

4.2.5.1 Pemodelan ARIMA untuk Periode Sebelum COVID-19

Setelah memastikan bahwa data telah mencapai stasioneritas dengan differencing, langkah selanjutnya adalah menentukan model Autoregressive Integrated Moving Average (ARIMA) yang paling sesuai. Model yang optimal dipilih berdasarkan nilai Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC) yang paling rendah, serta memastikan bahwa tidak ada parameter AR atau MA yang bernilai nol.

Untuk menentukan nilai p (AR) dan q (MA), beberapa model diuji dengan berbagai kombinasi parameter, kemudian dibandingkan untuk menemukan model yang memberikan keseimbangan antara kompleksitas dan keakuratan. Berikut adalah hasil pengujian berbagai model ARIMA untuk periode sebelum COVID (1 Januari 2018 – 1 Maret 2020):

Tabel 4.10. Ringkasan Hasil Pengujian ARIMA sebelum Covid-19

ARIMA		AR		MA		AIC	BIC
		1	2	1	2		
ARIMA(0,1,0)	Koefisien					4232.61	4236.51
	SE						
ARIMA(1,1,1)	Koefisien	0.0257		-1		3998.32	4010.02
	SE	0.0524		0.0098			
ARIMA(1,1,0)	Koefisien	- 0.4504				4152.18	4159.99
	SE	0.0468					
ARIMA(0,1,1)	Koefisien			-1		3996.55	4004.36
	SE			0.0101			
ARIMA(2,1,2)	Koefisien	0.045	- 0.0712	- 1.0176	0.0178	4000.5	4020.01
	SE	0.6708	0.0545	0.6725	0.672		
ARIMA(1,1,2)	Koefisien	- 0.9063		- 0.0635	- 0.9365	3998.69	4014.3
	SE	0.0891		0.0734	0.0734		
ARIMA(2,1,1)	Koefisien	0.0274	- 0.0706	-1		3998.5	4014.11
	SE	0.0523	0.0523	0.0107			
ARIMA(0,1,2)	Koefisien			- 0.9702	- 0.0298	3998.28	4009.98
	SE			0.0574	0.0565		

ARIMA		AR		MA		AIC	BIC
		1	2	1	2		
ARIMA(2,1,0)	Koefisien	- 0.6127	- 0.3593			4104.08	4115.79
	SE	0.049	0.049				

Sumber: Data diolah

Dari tabel 4.10 di atas, model dengan AIC paling rendah adalah ARIMA(0,1,1) dengan AIC = 3996.55, tetapi model ini tidak dapat dipilih karena memiliki AR = 0. Sebagai alternatif, model dengan AIC terendah berikutnya yang memenuhi syarat adalah ARIMA(1,1,1) dengan AIC = 3998.32 dan BIC = 4010.02. Model ini memiliki keseimbangan antara AR dan MA serta menghasilkan nilai AIC yang lebih kecil dibandingkan model lainnya yang memenuhi kriteria. Dengan demikian, ARIMA(1,1,1) dipilih sebagai model terbaik untuk periode sebelum COVID.

4.2.5.2 Pemodelan ARIMA untuk Periode pada Masa COVID-19

Setelah memastikan bahwa data telah mencapai stasioneritas dengan differencing, langkah selanjutnya adalah menentukan model Autoregressive Integrated Moving Average (ARIMA) yang paling sesuai. Model yang optimal dipilih berdasarkan nilai Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC) yang paling rendah, serta memastikan bahwa tidak ada parameter AR atau MA yang bernilai nol.

Untuk menentukan nilai p (AR) dan q (MA), beberapa model diuji dengan berbagai kombinasi parameter, kemudian dibandingkan untuk menemukan model yang memberikan keseimbangan antara kompleksitas dan keakuratan. Berikut adalah hasil pengujian berbagai model ARIMA untuk periode masa COVID (2 Maret 2020 – 31 Desember 2022):

Tabel 4.11. Ringkasan Hasil Pengujian ARIMA pada Masa Covid-19

ARIMA		AR					MA		AIC	BIC
		1	2	3	4	5	1	2		
ARIMA (5,1,0)	Koefisien	- 0.859	- 0.837	- 0.520	- 0.377	- 0.128			5505.83	5530.91
	SE	0.046	0.060	0.068	0.061	0.047				
ARIMA (4,1,0)	Koefisien	- 0.829	- 0.786	- 0.420	- 0.269				5511.07	5531.97
	SE	0.045	0.057	0.058	0.046					
ARIMA (3,1,0)	Koefisien	- 0.767	- 0.610	- 0.202					5541.82	5558.54
	SE	0.045	0.050	0.045						
ARIMA (2,1,0)	Koefisien	- 0.670	- 0.473						5559.41	5571.95
	SE	0.040	0.040							
ARIMA (1,1,0)	Koefisien	- 0.456							5678.33	5686.69
	SE	0.041								
ARIMA (0,1,0)	Koefisien								5788.21	5792.39
	SE									
ARIMA (5,1,1)	Koefisien	0.025	- 0.101	0.179	- 0.001	0.111	- 1.000		5458.33	5487.59
	SE	0.046	0.046	0.045	0.046	0.048	0.006			
ARIMA (4,1,1)	Koefisien	0.028	- 0.079	0.171	- 0.002		- 1.000		5461.74	5486.82

ARIMA		AR					MA		AIC	BIC
		1	2	3	4	5	1	2		
	SE	0.046	0.045	0.045	0.046		0.007			
ARIMA (3,1,1)	Koefisien	0.028	- 0.079	0.171			- 1.000		5459.75	5480.65
	SE	0.045	0.045	0.045			0.007			
ARIMA (2,1,1)	Koefisien	0.014	- 0.078				- 1.000		5471.73	5488.45
	SE	0.046	0.046				0.009			
ARIMA (1,1,1)	Koefisien	0.013					- 1.000		5472.6	5485.14
	SE	0.046					0.008			
ARIMA (0,1,1)	Koefisien						- 1.000		5470.68	5479.04
	SE						0.008			
ARIMA (5,1,2)	Koefisien	0.046	- 0.101	0.181	- 0.004	0.111	- 1.022	- 1.022	5460.32	5493.76
	SE	0.321	0.047	0.052	0.071	0.048	0.321	0.321		
ARIMA (4,1,2)	Koefisien	- 0.387	- 0.070	0.138	0.039		- 0.579	- 0.421	5463.14	5492.4
	SE	0.411	0.050	0.059	0.079		0.410	0.410		
ARIMA (3,1,2)	Koefisien	- 0.235	- 0.075	0.145			- 0.730	- 0.271	5461.59	5486.67
	SE	0.362	0.047	0.064			0.368	0.368		

ARIMA		AR					MA		AIC	BIC
		1	2	3	4	5	1	2		
ARIMA (2,1,2)	Koefisien	- 0.593	- 0.115				- 0.378	- 0.622	5464.96	5485.86
	SE	0.118	0.048				0.112	0.112		
ARIMA (1,1,2)	Koefisien	- 0.643					- 0.269	- 0.731	5468.26	5484.98
	SE	0.116					0.101	0.101		
ARIMA (0,1,2)	Koefisien						- 0.984	- 0.016	5472.58	5485.12
	SE						0.051	0.050		

Sumber: Data diolah

Dari tabel 4.11 di atas, model dengan AIC paling rendah adalah ARIMA(5,1,1) dengan AIC = 5458.33. Model ini memiliki keseimbangan antara kompleksitas dan akurasi serta tidak memiliki parameter AR atau MA yang bernilai nol, sehingga layak dipilih sebagai model terbaik untuk periode masa COVID.

4.2.5.3 Pemodelan ARIMA untuk Periode Setelah COVID-19

Setelah memastikan bahwa data telah mencapai stasioneritas dengan differencing, langkah selanjutnya adalah menentukan model Autoregressive Integrated Moving Average (ARIMA) yang paling sesuai. Model yang optimal dipilih berdasarkan nilai Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC) yang paling rendah, serta memastikan bahwa tidak ada parameter AR atau MA yang bernilai nol.

Untuk menentukan nilai p (AR) dan q (MA), beberapa model diuji dengan berbagai kombinasi parameter, kemudian dibandingkan untuk menemukan model yang memberikan keseimbangan antara kompleksitas dan keakuratan. Berikut adalah hasil pengujian berbagai model ARIMA untuk periode setelah COVID (1 Januari 2023 – 5 September 2024):

Tabel 4.12. Ringkasan Hasil Pengujian ARIMA Setelah Covid-19

ARIMA		AR			MA		AIC	BIC
		1	2	3	1	2		
ARIMA (3,1,0)	Koefisien	- 0.9405	- 0.6917	- 0.3325			2900.82	2915.30
	SE	0.0575	0.0699	0.0582				
ARIMA (2,1,0)	Koefisien	- 0.8064	- 0.4273				2929.58	2940.44
	SE	0.0555	0.0555					
ARIMA (1,1,0)	Koefisien	- 0.5611					2981.10	2988.34
	SE	0.0501						
ARIMA (0,1,0)	Koefisien						3082.38	3086.00
	SE							
ARIMA (3,1,1)	Koefisien	- 0.1713	- 0.0471	0.0310	- 1.0000		2854.38	2872.48
	SE	0.0603	0.0618	0.0625	0.0217			
ARIMA (2,1,1)	Koefisien	- 0.1729	- 0.0533		- 1.0000		2852.62	2867.10
	SE	0.0602	0.0605		0.0242			
ARIMA (1,1,1)	Koefisien	- 0.1641			- 1.0000		2851.40	2862.26
	SE	0.0595			0.0193			

ARIMA		AR			MA		AIC	BIC
		1	2	3	1	2		
ARIMA (0,1,1)	Koefisien				- 1.0000		2856.89	2864.13
	SE				0.0135			
ARIMA (3,1,2)	Koefisien	0.2296	0.0250	0.0665	- 1.4010	0.4010	2856.01	2877.73
	SE	0.5098	0.1057	0.0642	0.5082	0.5080		
ARIMA (2,1,2)	Koefisien	- 0.3155	- 0.0780		- 0.8573	-0.1427	2854.56	2872.66
	SE	0.5667	0.1060		0.5662	0.5657		
ARIMA (1,1,2)	Koefisien	0.0248			- 1.1969	0.1969	2852.90	2867.38
	SE	0.2724			0.2656	0.2647		
ARIMA (0,1,2)	Koefisien				- 1.1730	0.1730	2850.91	2861.77
	SE				0.0632	0.0594		

Sumber: Data diolah

Dari tabel 4.12 di atas, ARIMA(0,1,2) memang memiliki AIC paling rendah (2850.91), namun model tersebut memiliki AR = 0, sehingga tidak memenuhi kriteria bahwa tidak boleh ada parameter AR atau MA yang bernilai nol. Model terbaik berikutnya yang memenuhi syarat adalah ARIMA(1,1,1) dengan AIC = 2851.40, di mana komponen AR dan MA sama-sama tidak bernilai nol. Dengan demikian, ARIMA(1,1,1) dipilih sebagai model terbaik untuk periode setelah COVID.

Selanjutnya, model dalam 3 periode tersebut akan dianalisis lebih lanjut untuk memastikan bahwa residual memenuhi asumsi white noise, yaitu tidak memiliki autokorelasi yang signifikan. Jika asumsi ini terpenuhi, model dapat digunakan untuk peramalan pada periode berikutnya.

4.2.1 Uji Normalitas Residual

Uji normalitas residual dilakukan untuk mengevaluasi apakah sisaan dari model terdistribusi secara normal. Hipotesis nol (H_0) menyatakan bahwa residual terdistribusi normal, sedangkan hipotesis alternatif (H_1) menyatakan bahwa residual tidak terdistribusi normal. Uji Jarque-Bera digunakan untuk menguji normalitas ini, dengan kriteria pengambilan keputusan bahwa jika $p\text{-value} < 0.05$, maka H_0 ditolak.

- **Periode Sebelum Covid-19**

Hasil uji Jarque-Bera menunjukkan nilai X-squared = 28.54, $df = 2$, dan $p\text{-value} = 6.349e-07$. Karena $p\text{-value}$ sangat kecil (< 0.05), H_0 ditolak, yang mengindikasikan bahwa residual tidak terdistribusi normal.

- **Periode Masa Covid-19**

Pada periode ini, hasil uji menghasilkan X-squared = 739.63, $df = 2$, dan $p\text{-value} < 2.2e-16$. $P\text{-value}$ yang ekstrem rendah mengonfirmasi bahwa H_0 ditolak, sehingga residual juga tidak terdistribusi normal.

- **Periode Setelah Covid-19**

Hasil uji menunjukkan X-squared = 5.3772, $df = 2$, dan $p\text{-value} = 0.06798$. Karena $p\text{-value}$ lebih besar dari 0.05, gagal untuk menolak H_0 , yang berarti residual terdistribusi normal pada periode ini.

Dengan demikian, uji normalitas residual mengungkapkan bahwa model menghasilkan sisaan yang tidak normal pada periode sebelum dan selama COVID, sedangkan pada periode setelah COVID, residual menunjukkan distribusi normal.

4.2.1 Uji Autokorelasi Residual

Uji autokorelasi residual dilakukan untuk memastikan bahwa sisaan dari model tidak memiliki pola tertentu atau korelasi satu sama lain. Hipotesis yang diuji dalam uji Box-Ljung adalah sebagai berikut:

- H_0 : Tidak ada autokorelasi dalam residual (residual bersifat white noise).
- H_1 : Terdapat autokorelasi dalam residual.

Keputusan diambil berdasarkan nilai p-value, di mana jika $p\text{-value} < 0.05$, maka H_0 ditolak, yang berarti residual memiliki autokorelasi. Sebaliknya, jika $p\text{-value} \geq 0.05$, maka tidak ada bukti cukup untuk menolak H_0 , sehingga residual dianggap bebas dari autokorelasi.

- **Periode Sebelum Covid-19**

Hasil uji Box-Ljung menunjukkan nilai X-squared = 0.002787, $df = 1$, dan $p\text{-value} = 0.9579$. Karena p-value jauh lebih besar dari 0.05, maka tidak ada cukup bukti untuk menolak H_0 . Ini mengindikasikan bahwa residual tidak memiliki autokorelasi dan dapat dianggap sebagai white noise pada periode ini.

- **Periode Masa Covid-19**

Pada periode ini, hasil uji menghasilkan X-squared = 0.050079, $df = 1$, dan $p\text{-value} = 0.8229$. Nilai p-value yang juga jauh di atas 0.05 menunjukkan bahwa H_0 tidak dapat ditolak, sehingga residual pada periode ini juga tidak memiliki autokorelasi yang signifikan.

- **Periode Setelah Covid-19**

Hasil uji pada periode setelah COVID menunjukkan nilai X-squared = 0.011539, $df = 1$, dan $p\text{-value} = 0.9145$. Dengan p-value yang tinggi, H_0 kembali tidak ditolak, yang berarti residual tetap bebas dari autokorelasi pada periode ini.

Dengan demikian, hasil uji autokorelasi menunjukkan bahwa residual dari model pada ketiga periode tidak memiliki autokorelasi yang signifikan, sehingga dapat dikatakan bahwa residual bersifat white noise dan model yang digunakan tidak meninggalkan pola yang belum terjelaskan.

4.2.1 Uji Rerata Residual Sama dengan Nol

Uji rerata residual bertujuan untuk memastikan bahwa nilai rata-rata dari residual mendekati nol, yang merupakan salah satu asumsi penting dalam model ARIMA-GARCH. Jika rerata residual tidak berbeda secara signifikan dari nol, maka model yang digunakan dapat dikatakan tidak memiliki bias sistematis.

Hipotesis yang diuji dalam uji One Sample t-test ini adalah sebagai berikut:

- H_0 : Mean residual = 0 (Rerata residual tidak berbeda secara signifikan dari nol).
- H_1 : Mean residual $\neq$ 0 (Rerata residual berbeda secara signifikan dari nol).

Keputusan diambil berdasarkan nilai p-value. Jika p-value < 0.05 , maka H_0 ditolak, yang berarti residual memiliki rerata yang berbeda secara signifikan dari nol. Jika p-value ≥ 0.05 , maka tidak ada cukup bukti untuk menolak H_0 , sehingga residual dapat dianggap memiliki rerata nol.

- **Periode Sebelum Covid-19**

Hasil uji menunjukkan $t = 0.061642$, $df = 367$, dan p-value = 0.9509. Dengan p-value yang jauh lebih besar dari 0.05, H_0 tidak dapat ditolak, yang berarti tidak ada cukup bukti bahwa rerata residual berbeda dari nol. Interval kepercayaan 95% untuk rerata residual berada pada rentang (-5.549644, 5.908834), yang mencakup nol, sehingga residual pada periode ini dapat dianggap memiliki rerata nol.

- **Periode Masa Covid-19**

Pada periode ini, hasil uji menghasilkan $t = 1.024$, $df = 483$, dan p-value = 0.3064. Karena p-value masih lebih besar dari 0.05, H_0 tetap tidak ditolak, sehingga tidak ada bukti yang cukup untuk menyatakan bahwa rerata residual berbeda dari nol. Interval kepercayaan 95% berada dalam rentang (-2.880328, 9.149534), yang mencakup nol, sehingga residual pada periode ini juga dapat dianggap memiliki rerata nol.

- **Periode Setelah Covid-19**

Pada periode setelah COVID, uji menghasilkan $t = 1.2615$, $df = 276$, dan p-value = 0.2082. Dengan p-value yang masih lebih besar dari 0.05, H_0 tidak dapat ditolak, yang berarti tidak ada cukup bukti bahwa rerata residual berbeda dari nol. Interval kepercayaan 95% untuk rerata

residual berada dalam rentang $(-1.756852, 8.025389)$, yang kembali mencakup nol, sehingga residual pada periode ini tetap memenuhi asumsi rerata nol.

Dari hasil uji pada ketiga periode, nilai p-value selalu lebih besar dari 0.05, yang berarti tidak ada cukup bukti untuk menolak H_0 . Dengan demikian, residual dari model dapat dianggap memiliki rerata nol pada seluruh periode analisis. Hal ini menunjukkan bahwa model yang digunakan tidak memiliki bias sistematis dan cukup memenuhi asumsi fundamental dalam analisis time series.

4.2.1 Uji Homoskedastisitas Residual dengan White Test

Uji homoskedastisitas bertujuan untuk menguji apakah varians residual bersifat konstan (homoskedastisitas) atau tidak (heteroskedastisitas). Jika residual memiliki varians yang tidak konstan, maka terjadi heteroskedastisitas, yang dapat menyebabkan model tidak efisien dalam estimasi parameter.

Dalam uji White Test yang menggunakan studentized Breusch-Pagan test, hipotesis yang diuji adalah:

- H_0 : Residual memiliki varians yang konstan (homoskedastisitas).
- H_1 : Residual memiliki varians yang tidak konstan (heteroskedastisitas).

Keputusan diambil berdasarkan nilai p-value. Jika $p\text{-value} < 0.05$, maka H_0 ditolak, yang berarti residual memiliki heteroskedastisitas. Sebaliknya, jika $p\text{-value} \geq 0.05$, maka tidak ada cukup bukti untuk menolak H_0 , sehingga residual dapat dianggap memiliki homoskedastisitas.

- **Periode Sebelum COVID**

Hasil pengujian dengan White Test menunjukkan $BP = 10.212$, $df = 2$, dan $p\text{-value} = 0.00606$. Karena $p\text{-value} < 0.05$, maka H_0 ditolak, yang berarti terdapat heteroskedastisitas pada residual. Dengan kata lain, varians residual tidak konstan pada periode sebelum COVID, yang mengindikasikan bahwa model mungkin mengalami masalah dalam menangani volatilitas data selama periode ini.

- **Periode Masa Covid-19**

Pada periode ini, hasil pengujian menghasilkan $BP = 24.87$, $df = 2$, dan $p\text{-value} = 3.977e-06$. Karena $p\text{-value} < 0.05$, maka H_0 kembali ditolak, yang berarti residual memiliki heteroskedastisitas. Hal ini menunjukkan bahwa selama periode COVID, volatilitas data sangat bervariasi, sehingga model ARIMA yang digunakan tidak mampu menangani perubahan varians yang tinggi.

- **Periode Setelah Covid-19**

Pada periode setelah COVID, hasil uji menunjukkan $BP = 7.5882$, $df = 2$, dan $p\text{-value} = 0.0225$. Karena $p\text{-value} < 0.05$, maka H_0 ditolak, yang berarti residual masih mengalami heteroskedastisitas. Dengan kata lain, setelah COVID, varians residual tetap tidak konstan, sehingga perlu dilakukan pendekatan tambahan untuk menangani efek volatilitas dalam model.

Dari hasil uji White Test pada ketiga periode, semua $p\text{-value}$ berada di bawah 0.05, sehingga dapat disimpulkan bahwa residual mengalami heteroskedastisitas di seluruh periode analisis. Karena model ARIMA tidak mampu menangani heteroskedastisitas, maka langkah selanjutnya adalah menggunakan model yang dapat menangkap pola volatilitas yang tidak konstan, seperti Generalized Autoregressive Conditional Heteroskedasticity (GARCH). Model GARCH akan digunakan untuk memperbaiki ketidakstabilan varians residual sehingga dapat menghasilkan peramalan yang lebih akurat.

4.2.1 Uji Efek ARCH pada Residual dan Pemilihan Model GARCH

Dalam model ARIMA-GARCH, uji efek ARCH digunakan untuk menentukan apakah terdapat autokorelasi dalam varians residual. Jika efek ARCH signifikan, maka model ARIMA saja tidak cukup, dan diperlukan model GARCH untuk menangkap volatilitas dalam data.

Uji dilakukan menggunakan ARCH-LM Test, dengan hipotesis sebagai berikut:

- H_0 : Tidak terdapat efek ARCH dalam residual (volatilitas residual tidak berkorelasi).
- H_1 : Terdapat efek ARCH dalam residual (volatilitas residual berkorelasi).

Jika $p\text{-value} < 0.05$, maka H_0 ditolak, menunjukkan adanya efek ARCH yang signifikan, sehingga model GARCH diperlukan.

- **Periode Sebelum COVID (1 Januari 2018 – 1 Maret 2020)**

Hasil uji ARCH menunjukkan bahwa p-value untuk lag ke-1 adalah 0.016, sehingga H_0 ditolak, yang berarti terdapat efek ARCH yang signifikan. Namun, pada lag ke-2 (0.051), p-value sedikit di atas 0.05, dan pada lag-lag berikutnya p-value semakin besar. Hal ini menunjukkan bahwa efek ARCH dominan pada lag kecil, tetapi tidak terlalu kuat pada lag yang lebih besar. Berdasarkan hasil ini, model GARCH(0,1), GARCH(1,1), GARCH(1,2), dan GARCH(2,2) diuji untuk melihat bagaimana volatilitas dapat dimodelkan dengan baik.

- **Periode Masa COVID (2 Maret 2020 – 31 Desember 2022)**

Pada periode ini, hampir semua p-value sangat kecil (mendekati nol), menunjukkan adanya efek ARCH yang sangat signifikan di semua lag. Efek ARCH sangat kuat pada periode ini, yang menandakan adanya volatilitas tinggi selama masa pandemi. Model yang lebih fleksibel seperti GARCH(1,1) atau GARCH(1,2) dapat digunakan, tetapi model dengan orde lebih tinggi seperti GARCH(2,1) atau GARCH(2,2) juga layak dipertimbangkan untuk menangkap efek volatilitas yang lebih kompleks.

- **Periode Setelah COVID (1 Januari 2023 – 5 September 2024)**

Hasil uji pada periode ini identik dengan periode masa COVID, di mana semua p-value sangat kecil hingga nol pada beberapa lag. Efek ARCH tetap kuat setelah pandemi, menunjukkan bahwa volatilitas tidak kembali ke kondisi stabil seperti sebelum COVID. Sama seperti pada masa COVID, model GARCH(1,1) atau GARCH(1,2) adalah pilihan awal yang baik. Jika pola volatilitas lebih kompleks, model GARCH(2,1) atau GARCH(2,2) dapat dipertimbangkan.

4.2.1 Pemilihan Model GARCH Terbaik

Setelah dilakukan uji efek ARCH pada residual model ARIMA di ketiga periode data, hasil menunjukkan adanya heteroskedastisitas yang signifikan, terutama pada masa pandemi COVID-19 dan periode setelahnya. Hasil ini menegaskan pentingnya penggunaan model volatilitas seperti GARCH untuk menangkap dinamika fluktuasi varians yang tidak konstan (time-varying volatility) dalam data pasar modal. Oleh karena itu, pendekatan GARCH dipilih sebagai solusi untuk memodelkan karakteristik volatilitas tersebut.

Dalam menentukan model GARCH terbaik yang digunakan pada penelitian ini, peneliti menggunakan kriteria Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC). Kedua indikator ini merupakan metode yang umum dan kuat dalam pemilihan model statistik, karena mempertimbangkan trade-off antara kompleksitas model dan goodness-of-fit. Model dengan nilai AIC dan BIC yang lebih rendah dianggap memberikan estimasi yang optimal karena mampu meminimalkan kesalahan prediksi dengan jumlah parameter yang efisien. Berdasarkan hasil estimasi, model yang terpilih dalam masing-masing periode menunjukkan bahwa Standard GARCH (sGARCH) merupakan model yang paling sesuai untuk mewakili karakteristik volatilitas pada data IHSG.

Namun demikian, perlu disampaikan bahwa terdapat indikasi adanya sifat asimetris dalam volatilitas, terutama di periode pandemi dan pasca-pandemi. Hal ini terlihat dari pola respons pasar yang cenderung lebih sensitif terhadap informasi negatif dibandingkan informasi positif—sebuah fenomena yang dikenal dalam literatur sebagai "leverage effect". Model-model lanjutan seperti EGARCH (Exponential GARCH) atau TARCH (Threshold ARCH) sejatinya dapat digunakan untuk menangkap aspek asimetris ini secara lebih tepat. Model EGARCH, misalnya, memungkinkan pengaruh kejutan negatif dan positif terhadap volatilitas untuk bersifat tidak simetris, tanpa mengharuskan varians selalu bernilai positif. Demikian pula, TARCH dapat membedakan efek guncangan berdasarkan arahnya.

Meski demikian, sesuai dengan batasan metodologis yang telah dijelaskan dalam Bab 1, penelitian ini secara eksplisit hanya menggunakan pendekatan GARCH standar sebagai fokus utama, sehingga penggunaan EGARCH atau TARCH tidak dilakukan dalam tahap estimasi lanjutan. Keputusan ini didasarkan pada kesesuaian dengan tujuan awal penelitian dan ruang lingkup analisis yang telah ditetapkan. Oleh karena itu, meskipun secara statistik terdapat peluang untuk mengeksplorasi model-model GARCH asimetris, langkah tersebut tidak dilakukan secara spesifik dalam studi ini agar tetap berada dalam koridor metodologi yang telah ditentukan.

Berikut ini adalah hasil pemilihan model GARCH berdasarkan kombinasi ARIMA-GARCH di tiga periode: sebelum pandemi COVID-19, selama pandemi COVID-19, dan setelah pandemi COVID-19.

4.2.11.1 Pemilihan Model GARCH untuk Periode Sebelum COVID-19

Pada periode sebelum pandemi COVID-19, pasar keuangan relatif stabil dengan tingkat volatilitas yang lebih terkendali dibandingkan periode berikutnya. Meskipun demikian, hasil uji efek ARCH pada residual model ARIMA menunjukkan adanya heteroskedastisitas, sehingga diperlukan pendekatan Generalized Autoregressive Conditional Heteroskedasticity (GARCH) untuk menangkap pola volatilitas yang dinamis. Oleh karena itu, dilakukan pemodelan ARIMA-GARCH guna memperoleh model terbaik yang dapat menggambarkan karakteristik data sebelum pandemi.

Tabel 4.13. Ringkasan Hasil Pengujian GARCH Sebelum Covid-19

ARIMA	GARCH	AIC	BIC
(1,1,1)	(0,1)	10.9	10.943
(1,1,1)	(1,1)	10.878	10.931
(1,1,1)	(1,2)	10.875	10.939
(1,1,1)	(2,2)	10.88	10.955

Dalam proses pemodelan, model ARIMA(1,1,1) dipilih sebagai model dasar karena memberikan hasil yang optimal dalam menangkap pola data time series. Selanjutnya, model ini dikombinasikan dengan beberapa spesifikasi GARCH, dan hasil evaluasi menunjukkan bahwa model ARIMA(1,1,1)-GARCH(1,2) memiliki performa terbaik berdasarkan kriteria Akaike Information Criterion (AIC) dan Bayesian Information Criterion (BIC). Model ini memberikan nilai AIC sebesar 10.875 dan BIC sebesar 10.939, yang lebih rendah dibandingkan dengan model GARCH lainnya.

Pemilihan model GARCH(1,2) didasarkan pada kemampuannya menangkap efek volatilitas yang lebih kompleks dibandingkan model GARCH(1,1). Dengan dua lag pada variabel varians bersyarat, model ini memungkinkan volatilitas saat ini dipengaruhi oleh dua periode sebelumnya, sehingga memberikan fleksibilitas lebih baik dalam menggambarkan dinamika fluktuasi pasar sebelum pandemi. Hal ini menjadi penting karena meskipun volatilitas pada periode ini lebih rendah dibandingkan masa pandemi, tetap terdapat pergerakan harga yang perlu dimodelkan dengan pendekatan yang lebih akurat.

Berdasarkan hasil evaluasi, model ARIMA(1,1,1)-GARCH(1,2) dipilih sebagai model terbaik untuk periode sebelum COVID-19. Model ini tidak hanya mampu menangkap pola volatilitas secara lebih akurat, tetapi juga memberikan dasar yang kuat untuk membandingkan karakteristik volatilitas di masa pandemi dan setelah pandemi. Keberhasilan model ini dalam menggambarkan dinamika pasar sebelum pandemi juga menunjukkan bahwa model yang lebih kompleks dalam struktur varians bersyarat dapat meningkatkan kemampuan prediksi dan pemahaman terhadap risiko pasar pada periode yang lebih stabil.

4.2.11.2 Pemilihan Model GARCH untuk Periode Masa COVID-19

Periode pandemi COVID-19 membawa ketidakpastian tinggi di pasar keuangan, yang tercermin dari volatilitas yang lebih besar dibandingkan periode sebelumnya. Hasil uji ARCH menunjukkan adanya efek heteroskedastisitas yang signifikan dalam residual model ARIMA, sehingga diperlukan pemodelan GARCH untuk menangkap pola volatilitas yang lebih dinamis. Oleh karena itu, dilakukan eksplorasi terhadap berbagai kombinasi model ARIMA-GARCH untuk menentukan model terbaik yang dapat menggambarkan volatilitas di masa pandemi.

Tabel 4.14. Ringkasan Hasil Pengujian GARCH pada Masa Covid-19

ARIMA	GARCH	AIC	BIC
(5,1,1)	(0,1)	11.238	11.272
(5,1,1)	(1,1)	11.064	11.108
(5,1,1)	(1,2)	11.076	11.128
(5,1,1)	(2,2)	11.073	11.133

Pada periode ini, model ARIMA(5,1,1) dipilih sebagai model dasar karena memberikan hasil yang optimal dalam menangkap pola data. Model ini kemudian dikombinasikan dengan berbagai spesifikasi GARCH, dan hasil evaluasi menunjukkan bahwa model ARIMA(5,1,1)-GARCH(1,1) memiliki performa terbaik berdasarkan kriteria AIC sebesar 11.064 dan BIC sebesar 11.108. Model ini memiliki nilai AIC dan BIC yang lebih rendah dibandingkan dengan model lainnya, yang mengindikasikan bahwa model ini lebih sesuai untuk menangkap volatilitas pada periode pandemi.

Model GARCH(1,1) dipilih karena kemampuannya dalam menangkap volatilitas pasar yang tinggi selama pandemi. Model ini memungkinkan volatilitas saat ini bergantung pada volatilitas periode sebelumnya dan efek kejutan dari periode sebelumnya, yang sesuai dengan pola volatilitas tinggi selama pandemi. Dengan model ini, dapat dilihat bagaimana dampak pandemi terhadap pasar keuangan menyebabkan volatilitas yang lebih tinggi dibandingkan periode sebelum pandemi.

Berdasarkan hasil evaluasi, model ARIMA(5,1,1)-GARCH(1,1) dipilih sebagai model terbaik untuk periode pandemi COVID-19. Model ini tidak hanya memberikan estimasi volatilitas yang lebih akurat, tetapi juga menunjukkan bahwa selama periode krisis, volatilitas pasar cenderung lebih dipengaruhi oleh volatilitas sebelumnya dibandingkan dengan struktur volatilitas yang lebih kompleks. Hal ini memberikan wawasan penting bagi analisis risiko dan strategi investasi dalam menghadapi ketidakpastian ekstrem seperti pandemi COVID-19.

4.2.11.3 Pemilihan Model GARCH untuk Periode Setelah COVID-19

Setelah pandemi COVID-19, pasar keuangan mulai menunjukkan pemulihan dengan volatilitas yang lebih rendah dibandingkan masa pandemi. Namun, dampak dari pandemi masih dapat dirasakan dalam bentuk fluktuasi pasar yang tidak sepenuhnya kembali ke tingkat stabil sebelum pandemi. Oleh karena itu, dalam analisis periode ini, dilakukan pemodelan ARIMA-GARCH untuk menangkap pola volatilitas yang masih ada pasca pandemi.

Tabel 4.15. Ringkasan Hasil Pengujian GARCH Setelah Covid-19

ARIMA	GARCH	AIC	BIC
(1,1,1)	(0,1)	10.311	10.364
(1,1,1)	(1,1)	10.267	10.333
(1,1,1)	(1,2)	10.276	10.354
(1,1,1)	(2,2)	10.279	10.371

Dalam proses pemodelan, model ARIMA(1,1,1) dipilih sebagai model dasar karena memberikan hasil terbaik dalam menangkap pola data setelah pandemi. Model ini kemudian dikombinasikan dengan berbagai spesifikasi GARCH untuk menentukan model terbaik. Hasil evaluasi

menunjukkan bahwa model ARIMA(1,1,1)-GARCH(1,1) memiliki performa terbaik berdasarkan nilai AIC sebesar 10.267 dan BIC sebesar 10.333, yang lebih rendah dibandingkan model lainnya.

Pemilihan model GARCH(1,1) untuk periode ini menunjukkan bahwa meskipun volatilitas pasar mulai menurun, pola pergerakan harga masih dipengaruhi oleh volatilitas periode sebelumnya. Model ini lebih sederhana dibandingkan model GARCH yang lebih kompleks seperti GARCH(1,2) atau GARCH(2,2), yang mengindikasikan bahwa pasar pasca pandemi cenderung memiliki struktur volatilitas yang lebih mirip dengan periode sebelum pandemi, meskipun masih terdapat efek volatilitas yang tersisa dari masa krisis.

Dengan mempertimbangkan hasil evaluasi, model ARIMA(1,1,1)-GARCH(1,1) dipilih sebagai model terbaik untuk periode setelah COVID-19. Model ini menunjukkan bahwa meskipun dampak pandemi terhadap volatilitas pasar mulai mereda, masih terdapat ketergantungan pada volatilitas masa lalu. Oleh karena itu, pemahaman terhadap pola volatilitas pasca pandemi sangat penting untuk mendukung pengambilan keputusan dalam investasi dan manajemen risiko di periode pemulihan ekonomi ini.

Berdasarkan hasil pemodelan ARIMA-GARCH untuk ketiga periode, ditemukan bahwa model terbaik bervariasi tergantung pada kondisi volatilitas di masing-masing periode. Pada periode sebelum pandemi COVID-19, model ARIMA(1,1,1)-GARCH(1,2) dipilih karena dapat menangkap pola volatilitas yang relatif stabil namun tetap dinamis. Selama periode pandemi, model ARIMA(5,1,1)-GARCH(1,1) menjadi model terbaik karena dapat mengakomodasi volatilitas tinggi akibat ketidakpastian pasar yang ekstrem. Sementara itu, pada periode setelah pandemi, model ARIMA(1,1,1)-GARCH(1,1) dipilih karena volatilitas pasar mulai menurun dan lebih mirip dengan kondisi sebelum pandemi.

Hasil ini menunjukkan bahwa volatilitas pasar bersifat dinamis dan dipengaruhi oleh faktor eksternal seperti krisis global. Model GARCH dengan struktur yang lebih kompleks diperlukan saat volatilitas pasar meningkat secara signifikan, seperti yang terjadi selama pandemi. Sebaliknya, ketika volatilitas lebih stabil, model GARCH yang lebih sederhana seperti GARCH(1,1) sudah cukup untuk menggambarkan dinamika pasar.

4.2.1 Peramalan Model ARIMA-GARCH

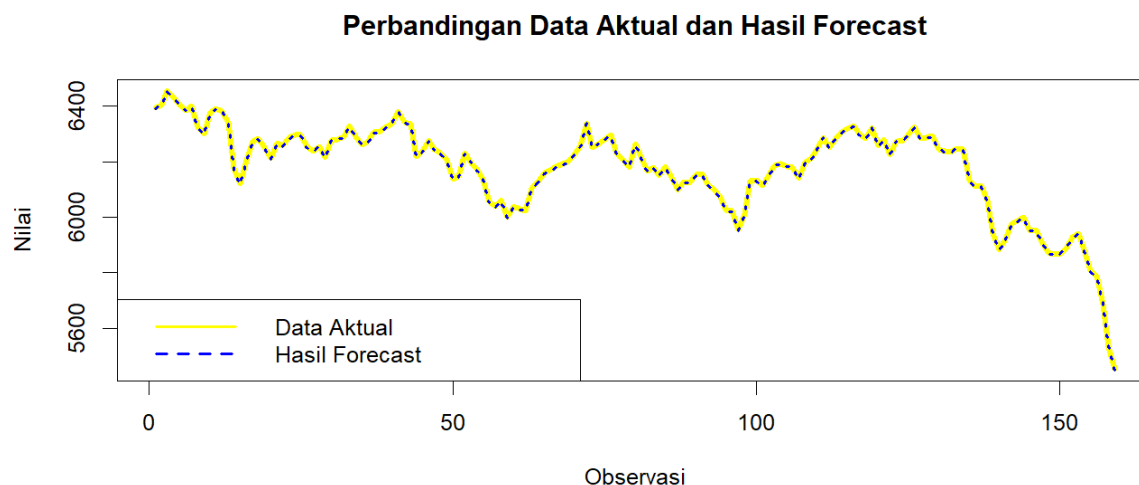
Setelah menentukan model ARIMA-GARCH terbaik untuk masing-masing periode, langkah selanjutnya adalah melakukan peramalan menggunakan model yang telah dipilih. Peramalan dengan ARIMA-GARCH bertujuan untuk mengestimasi nilai masa depan berdasarkan pola historis yang telah teridentifikasi dalam data. Model ini tidak hanya mempertimbangkan pola tren dan musiman melalui komponen ARIMA tetapi juga menangkap karakteristik volatilitas yang berubah-ubah melalui komponen GARCH.

Pada penelitian ini, peramalan akan dilakukan secara terpisah untuk tiga periode berbeda, yaitu sebelum COVID-19, masa COVID-19, dan setelah COVID-19. Setiap periode memiliki karakteristik volatilitas yang unik, sehingga model yang digunakan untuk peramalan juga berbeda sesuai dengan hasil pemilihan model sebelumnya. Untuk periode sebelum COVID-19, digunakan model ARIMA(1,1,1)-GARCH(1,2), sementara pada masa pandemi digunakan ARIMA(5,1,1)-GARCH(1,1), dan setelah pandemi digunakan ARIMA(1,1,1)-GARCH(1,1).

Dalam proses peramalan ini, hasil prediksi model akan dibandingkan dengan nilai aktual untuk menilai sejauh mana model ARIMA-GARCH dapat menangkap pola pergerakan data secara akurat. Selain itu, beberapa metrik kesalahan seperti Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), dan Root Mean Squared Error (RMSE) akan digunakan untuk mengukur tingkat akurasi model. Metrik-metrik ini membantu dalam mengevaluasi apakah model yang digunakan mampu memberikan estimasi yang cukup baik atau masih memiliki kelemahan yang perlu diperbaiki.

4.2.12.1 Peramalan Model ARIMA-GARCH pada Periode Sebelum COVID-19

Peramalan menggunakan model ARIMA-GARCH bertujuan untuk menghasilkan prediksi yang lebih akurat dengan menangkap pola serial dari data serta volatilitas yang bersifat heteroskedastik. Pada periode sebelum COVID-19 (1 Januari 2018 – 1 Maret 2020), model yang digunakan dalam peramalan ini adalah kombinasi ARIMA(1,1,1) dengan GARCH(1,2), yang dipilih berdasarkan kriteria informasi Akaike (AIC) dan Bayesian (BIC) pada tahap seleksi model sebelumnya. Model ini mempertimbangkan komponen autoregresi (AR), perbedaan (I), dan moving average (MA) dari data deret waktu, serta memperhitungkan volatilitas dalam variabel residual dengan pendekatan GARCH.



Gambar 4.11. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19

Gambar 4.11 menampilkan hasil peramalan dibandingkan dengan data aktual untuk mengevaluasi akurasi model. Grafik tersebut membandingkan antara data aktual dan hasil forecast menunjukkan bahwa model ARIMA-GARCH mampu mengikuti pola pergerakan data historis dengan cukup baik. Garis kuning yang mewakili data aktual dan garis biru putus-putus yang menunjukkan hasil peramalan memiliki pola yang sangat mirip, menandakan bahwa model mampu menangkap tren utama serta fluktuasi yang terjadi selama periode ini.

Tabel 4.16. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19

No	Aktual	Ramalan
1	6394.61	6390.48
2	6403.29	6402.88
3	6456.54	6452.78
4	6433.55	6432.8
5	6403.81	6400.35
⋮	⋮	⋮
155	5807.05	5804.87

No	Aktual	Ramalan
156	5787.14	5784.96
157	5688.92	5686.75
158	5535.69	5533.52
159	5452.7	5450.53

Selain visualisasi, evaluasi kuantitatif dilakukan dengan membandingkan beberapa titik data aktual dan hasil peramalan. Sebagai contoh, pada observasi pertama, nilai aktual sebesar 6394.61, sedangkan hasil peramalan adalah 6390.48. Perbedaan serupa terlihat pada observasi lainnya, baik di awal maupun di akhir periode, seperti pada observasi ke-159 di mana nilai aktual sebesar 5452.70 dan hasil peramalan adalah 5450.53. Perbedaan yang kecil ini menunjukkan bahwa model memiliki kemampuan prediksi yang cukup baik dalam merepresentasikan pola data historis.

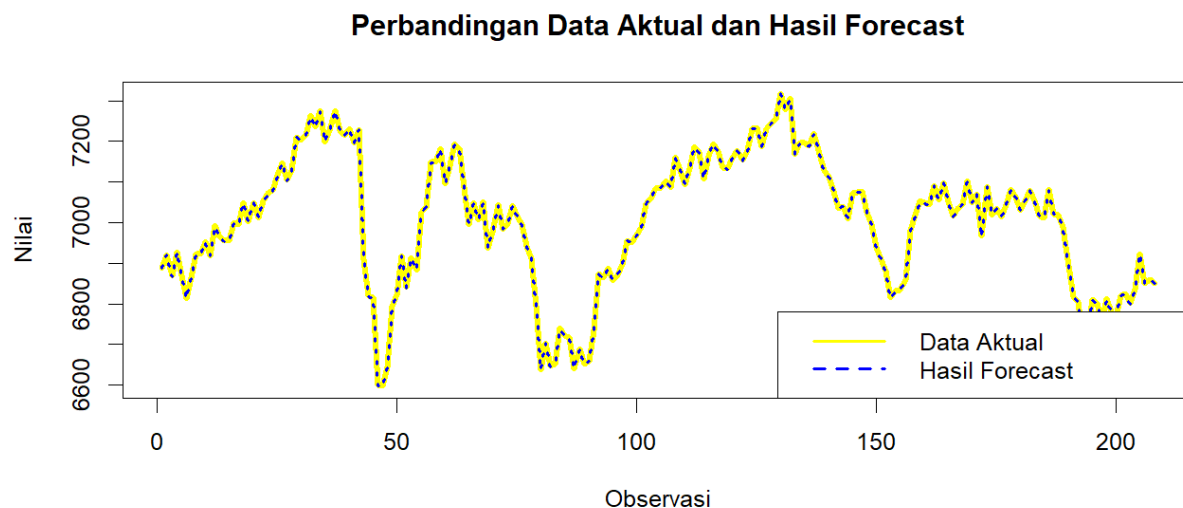
Untuk mengukur tingkat kesalahan peramalan, tiga metrik utama digunakan: Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), dan Root Mean Squared Error (RMSE). Hasil perhitungan menunjukkan bahwa nilai MSE adalah 4.887, yang mencerminkan rata-rata kuadrat selisih antara nilai aktual dan hasil peramalan. Sementara itu, nilai MAPE yang sangat kecil, yaitu 0.035%, mengindikasikan bahwa kesalahan rata-rata dalam satuan persentase relatif terhadap data aktual sangat rendah. RMSE yang diperoleh adalah 2.211, menunjukkan bahwa kesalahan peramalan memiliki besaran yang kecil dibandingkan skala data.

Berdasarkan hasil peramalan ini, dapat disimpulkan bahwa model ARIMA(1,1,1)-GARCH(1,2) memiliki performa yang baik dalam menangkap pola data dan memprediksi nilai pada periode sebelum COVID-19. Model ini tidak hanya mampu merepresentasikan tren data dengan baik tetapi juga menghasilkan tingkat kesalahan yang rendah, yang menunjukkan bahwa kombinasi ARIMA dan GARCH efektif dalam mengatasi sifat heteroskedastik pada volatilitas pada data finansial.

4.2.12.2 Peramalan Model ARIMA-GARCH pada Periode Masa COVID-19

Pada periode masa COVID-19 (2 Maret 2020 – 31 Desember 2022), volatilitas pasar mengalami lonjakan akibat ketidakpastian ekonomi yang tinggi. Faktor seperti kebijakan lockdown, perubahan suku bunga, serta gangguan pada rantai pasok global menyebabkan fluktuasi harga yang jauh lebih besar dibandingkan periode sebelumnya. Untuk menangkap pola yang lebih dinamis selama periode ini, model ARIMA(5,1,1)-GARCH(1,1) dipilih berdasarkan kriteria informasi Akaike (AIC) dan Bayesian (BIC). Model ini memiliki komponen autoregresi dan moving average yang lebih kompleks dibandingkan periode sebelum COVID-19 guna mengakomodasi perubahan tren yang lebih tajam dan volatilitas yang meningkat.

Gambar 4.12 Perbandingan Data Aktual dan Forecast pada Periode Masa Covid-19



Gambar 4.12. Perbandingan Data Aktual dan *Forecast* pada Periode Masa Covid-19

Gambar 4.12 menunjukkan hasil peramalan dibandingkan dengan data aktual, yang mengindikasikan bahwa model ARIMA-GARCH mampu menangkap pola pergerakan data dengan sangat baik. Grafik perbandingan memperlihatkan bahwa garis kuning yang mewakili data aktual dan garis biru putus-putus dari hasil peramalan bergerak hampir berimpitan, menandakan bahwa model ini memiliki tingkat akurasi yang sangat tinggi. Penyimpangan antara hasil peramalan dan data aktual terlihat minimal, bahkan pada titik-titik dengan volatilitas tinggi, yang menunjukkan bahwa model mampu beradaptasi dengan fluktuasi yang terjadi selama periode pandemi.

Tabel 4.17. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19

No	Aktual	Ramalan
1	6888.171	6887.984
2	6921.441	6921.107
3	6868.402	6867.952
4	6928.328	6927.788
5	6869.066	6868.455
⋮	⋮	⋮
204	6835.808	6834.939
205	6923.027	6922.157
206	6850.52	6849.65
207	6860.077	6859.208
208	6850.619	6849.75

Evaluasi numerik dilakukan dengan membandingkan beberapa titik antara nilai aktual dan hasil peramalan. Sebagai contoh, pada lima observasi pertama, nilai aktual tercatat sebesar 6888.171, 6921.441, 6868.402, 6928.328, dan 6869.066, sementara hasil peramalan menunjukkan angka 6887.984, 6921.107, 6867.952, 6927.788, dan 6868.455. Perbedaan antara nilai aktual dan hasil peramalan sangat kecil, yang menunjukkan bahwa model ARIMA-GARCH mampu menangkap tren utama dengan baik dan memberikan prediksi yang sangat akurat. Selain itu, perbandingan di beberapa titik lain juga menunjukkan hasil yang konsisten, seperti pada observasi ke-204 di mana nilai aktual tercatat sebesar 6835.808 dan hasil peramalan adalah 6834.939, serta pada observasi ke-208 dengan nilai aktual 6850.619 dan hasil peramalan 6849.75. Perbedaan yang kecil ini menunjukkan bahwa model tidak hanya mampu menangkap tren keseluruhan tetapi juga dapat

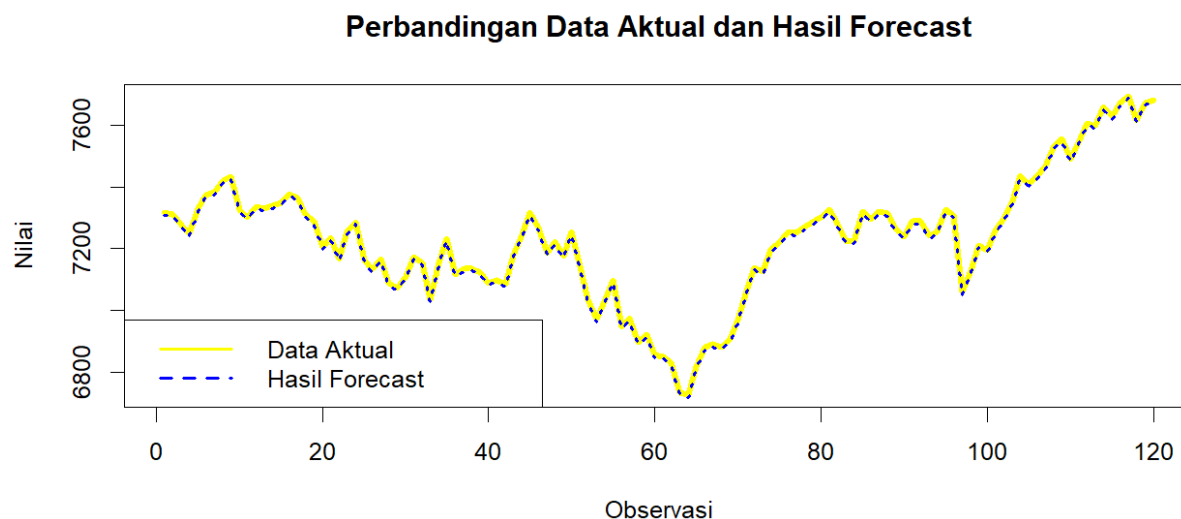
mereplikasi fluktuasi kecil dalam data, sehingga sangat efektif dalam memprediksi pergerakan harga selama periode masa COVID-19 yang ditandai dengan volatilitas tinggi.

Untuk mengukur akurasi peramalan, tiga metrik utama digunakan: MSE, MAPE, dan RMSE. Hasil perhitungan menunjukkan bahwa nilai MSE adalah 0.735, yang jauh lebih kecil dibandingkan periode sebelum dan setelah pandemi. Sementara itu, nilai MAPE sebesar 0.012% menunjukkan bahwa kesalahan rata-rata relatif terhadap nilai aktual tetap rendah. Nilai RMSE sebesar 0.858 menunjukkan bahwa kesalahan prediksi dalam satuan aslinya juga lebih kecil dibandingkan kedua periode lainnya.

Berdasarkan hasil ini, dapat disimpulkan bahwa model ARIMA(5,1,1)-GARCH(1,1) menunjukkan performa terbaik dibandingkan dengan periode lainnya, dengan tingkat kesalahan yang paling rendah. Keakuratan tinggi ini tercermin dari nilai MSE, RMSE, dan MAPE yang jauh lebih kecil dibandingkan dengan periode sebelum dan setelah COVID-19. Model ini berhasil menangkap pola utama dalam data serta merespons volatilitas dengan sangat baik, sehingga mampu menghasilkan peramalan yang sangat mendekati nilai aktual. Hasil ini mengindikasikan bahwa kombinasi ARIMA dan GARCH sangat efektif dalam menghadapi ketidakstabilan pasar selama pandemi, di mana pergerakan harga mengalami fluktuasi yang signifikan namun tetap dalam pola tertentu. Keunggulan model ini menegaskan bahwa pendekatan berbasis ARIMA-GARCH dapat digunakan sebagai metode andalan dalam kondisi pasar yang penuh ketidakpastian.

4.2.12.3 Peramalan Model ARIMA-GARCH pada Periode Setelah COVID-19

Setelah pandemi COVID-19, pasar mulai menunjukkan tanda-tanda pemulihan, meskipun masih terdapat fluktuasi yang cukup signifikan akibat ketidakpastian ekonomi global dan perubahan kebijakan moneter. Pada periode setelah COVID-19 (1 Januari 2023 – 5 September 2024), model terbaik yang dipilih adalah ARIMA(1,1,1)-GARCH(1,1), yang memiliki kombinasi parameter yang memungkinkan model menangkap tren pemulihan serta fluktuasi yang masih terjadi dalam data.



Gambar 4.13. Perbandingan Data Aktual dan *Forecast* pada Periode Setelah Covid-19

Gambar 4.13 memperlihatkan hasil peramalan yang dibandingkan dengan data actual. Gambar tersebut menunjukkan bahwa model masih mampu mengikuti pola historis, tetapi dengan deviasi yang sedikit lebih besar jika dibandingkan dengan periode sebelum dan selama COVID-19. Grafik perbandingan memperlihatkan bahwa garis kuning dari data aktual dan garis biru putus-putus dari hasil peramalan masih bergerak dalam pola yang sama, meskipun hanya terlihat sedikit perbedaan yang sangat tipis terutama pada periode dengan volatilitas yang meningkat. Ini menunjukkan bahwa model ARIMA-GARCH tetap dapat menangkap pola utama dan masih cukup mampu mengakomodasi pergerakan harga yang ekstrem.

Tabel 4.18. Perbandingan Data Aktual dan Forecast pada Periode Sebelum Covid-19

No	Aktual	Ramalan
1	7316.111	7308.542
2	7311.907	7304.154
3	7276.749	7268.991
4	7247.46	7239.702
5	7329.801	7322.043

No	Aktual	Ramalan
:	:	:
116	7670.733	7662.975
117	7694.53	7686.772
118	7616.521	7608.763
119	7672.896	7665.138
120	7681.043	7673.285

Evaluasi kuantitatif dilakukan dengan membandingkan beberapa titik data aktual dan hasil peramalan. Sebagai contoh, dalam lima observasi pertama, nilai aktual tercatat sebesar 7316.11, 7311.91, 7276.75, 7247.46, dan 7329.80, sedangkan hasil peramalan menunjukkan angka 7308.54, 7304.15, 7268.99, 7239.70, dan 7322.04. Perbedaan antara nilai aktual dan hasil prediksi tetap berada dalam kisaran yang wajar, dengan model yang mampu menangkap tren utama secara konsisten. Pada observasi selanjutnya, model tetap menunjukkan kinerja yang baik dengan selisih yang tidak terlalu besar, seperti pada observasi ke-116 hingga ke-120, di mana nilai aktual berkisar antara 7670.73 hingga 7681.04, sementara hasil peramalan berada di rentang 7662.98 hingga 7673.29. Hasil ini menegaskan bahwa model ARIMA-GARCH tetap memiliki kemampuan yang baik dalam memprediksi pergerakan harga, meskipun terdapat beberapa titik di mana deviasi menjadi lebih besar, terutama pada kondisi volatilitas yang lebih tinggi.

Untuk menilai akurasi peramalan, tiga metrik utama digunakan. Hasil evaluasi menunjukkan bahwa nilai MSE sebesar 60.159, yang jauh lebih tinggi dibandingkan periode sebelumnya, menunjukkan bahwa kesalahan rata-rata kuadrat dalam peramalan lebih besar pada periode setelah COVID-19. Nilai MAPE sebesar 0.107% menunjukkan bahwa kesalahan dalam satuan persentase masih relatif kecil tetapi meningkat dibandingkan periode sebelum dan selama COVID-19. Nilai RMSE sebesar 7.756 menunjukkan bahwa dalam satuan aslinya, kesalahan prediksi lebih besar dibandingkan dengan dua periode lainnya.

Berdasarkan hasil ini, dapat disimpulkan bahwa model ARIMA(1,1,2)-GARCH(1,1) tetap dapat digunakan untuk peramalan setelah COVID-19, tetapi dengan tingkat kesalahan yang lebih besar dibandingkan periode sebelumnya. Hal ini menunjukkan bahwa meskipun model ini masih menangkap pola utama dalam data, terdapat tantangan dalam mengakomodasi volatilitas pasar yang meningkat setelah pandemi.

Berdasarkan hasil penelitian ini, model ARIMA-GARCH menunjukkan performa yang sangat baik dalam memprediksi pergerakan IHSG pada berbagai kondisi pasar. Model ini memiliki tingkat akurasi yang lebih tinggi dibandingkan LSTM dalam kondisi pasar yang stabil maupun volatil, sebagaimana ditunjukkan oleh nilai MSE, MAPE, dan RMSE yang lebih rendah pada semua periode yang diuji. Untuk memperkaya analisis ini, hasil penelitian saya dibandingkan dengan beberapa studi terdahulu yang juga menerapkan model ARIMA-GARCH dalam berbagai konteks, termasuk pasar keuangan dan komoditas.

Dalam penelitian Xiang (2022) model ARIMA(1,1,0)-GARCH(1,1) diterapkan untuk menganalisis fluktuasi harga minyak internasional. Hasil penelitian tersebut menunjukkan bahwa model ini mampu meningkatkan akurasi prediksi dengan signifikan, di mana nilai MAPE turun dari 1.549% menjadi 0.045%, dan RMSE menurun dari 1.032 menjadi 0.071. Jika dibandingkan dengan hasil penelitian saya, meskipun objek yang dianalisis berbeda, efektivitas model ARIMA-GARCH dalam menangkap pola volatilitas pasar tetap konsisten. Dalam kasus saya, nilai MAPE sebesar 0.035% sebelum pandemi dan 0.012% selama pandemi menunjukkan bahwa model ini juga sangat efektif dalam memprediksi IHSG dengan tingkat kesalahan yang sangat rendah.

Selain itu, penelitian yang dilakukan oleh Abreu et al. (2019) menguji model ARIMA-GARCH dalam memprediksi nilai tukar EUR/USD dan membandingkannya dengan teknik Singular Spectrum Analysis (SSA). Hasil penelitian mereka menunjukkan bahwa meskipun SSA lebih unggul dalam beberapa kondisi, model ARIMA-GARCH masih memberikan prediksi yang lebih baik pada kondisi pasar tanpa tren yang jelas. Dalam penelitian saya, hasil serupa juga ditemukan, di mana ARIMA-GARCH tetap memberikan performa yang lebih baik dibandingkan LSTM dalam menangani volatilitas yang tinggi, terutama selama pandemi.

Studi yang dilakukan oleh Merabet et al. (2021) menerapkan model ARIMA-GARCH untuk memodelkan volatilitas harga minyak dan menemukan bahwa kombinasi ARIMA(3,1,1) dengan EGARCH(1,2) memberikan hasil terbaik dalam memprediksi pergerakan harga minyak dunia.

Model ini dipilih berdasarkan minimasi kriteria AIC dan BIC serta analisis residual yang menunjukkan bahwa varians bersyarat lebih baik dijelaskan dengan pendekatan GARCH asimetris. Dalam penelitian ini, model EGARCH menunjukkan keunggulan dalam menangkap efek leverage dan klusterisasi volatilitas, yang juga relevan dalam konteks pasar saham. Jika dibandingkan dengan penelitian saya, model ARIMA-GARCH yang digunakan juga berhasil menangkap pola volatilitas IHSG dengan cukup baik, terutama dalam periode pandemi di mana nilai RMSE tercatat sebesar 0.858, menunjukkan kemampuan model dalam mengakomodasi perubahan volatilitas yang tinggi.

Berdasarkan perbandingan ini, dapat disimpulkan bahwa model ARIMA-GARCH secara konsisten menunjukkan performa yang unggul dalam berbagai keadaan ekonomi. Meskipun ada beberapa pendekatan lain seperti SSA yang dapat digunakan dalam kondisi tertentu, ARIMA-GARCH tetap menjadi metode yang lebih andal dalam menangani volatilitas pasar dan memberikan prediksi yang lebih akurat. Hasil penelitian ini semakin menegaskan bahwa ARIMA-GARCH adalah alat yang sangat berguna dalam analisis pasar modal, terutama ketika dihadapkan pada perubahan pola volatilitas yang signifikan.

4.3 Hasil Pengujian dan Pembahasan LSTM

4.3.1 Normalisasi Data

Pada tahap awal analisis menggunakan metode Long Short-Term Memory (LSTM), data akan terlebih dahulu dinormalisasi menggunakan metode Min-Max Scaler. Normalisasi ini bertujuan untuk mengonversi nilai data ke dalam rentang tertentu, biasanya antara 0 dan 1, guna meningkatkan stabilitas dan efektivitas proses pelatihan model. Dengan menerapkan Min-Max Scaler, skala data yang berbeda dapat diseragamkan, sehingga memungkinkan model LSTM untuk lebih mudah mengenali pola dalam data historis serta mengurangi risiko eksplosinya gradien selama proses pembelajaran.

4.3.1.1 Normalisasi Data IHSG sebelum Covid-19

Pada tahap ini, data Indeks Harga Saham Gabungan (IHSG) sebelum pandemi COVID-19 (periode 1 Januari 2018 – 1 Maret 2020) dinormalisasi menggunakan metode Min-Max Scaler. Proses normalisasi ini bertujuan untuk mengonversi nilai data ke dalam rentang 0 hingga 1, sehingga memudahkan model dalam mengidentifikasi pola pergerakan data serta meningkatkan efisiensi

dalam proses pelatihan model LSTM. Tabel 4.19 menyajikan ringkasan statistik deskriptif dari data sebelum COVID-19 setelah dilakukan normalisasi.

Tabel 4.19. Normalisasi Data IHSG sebelum Covid-19

Date	Close	Normalisasi
1/2/2018	6339.238	0.71692
1/3/2018	6251.479	0.64595
1/4/2018	6292.321	0.67898
1/5/2018	6353.738	0.72865
1/8/2018	6385.404	0.75426
⋮	⋮	⋮
2/24/2020	5807.050	0.37991
2/25/2020	5787.138	0.35857
2/26/2020	5688.921	0.25326
2/27/2020	5535.694	0.08898
2/28/2020	5452.704	0

Normalisasi data menggunakan metode Min-Max Scaler dilakukan dengan menggunakan persamaan 2.33 Berikut ini variabel perhitungan normalisasi data IHSG sebelum Covid-19

- X = Nilai asli IHSG
- X_{min} = nilai minimum IHSG sebelum Covid-19 (5452.704)
- X_{max} = nilai maksimum IHSG sebelum Covid-19 (6689.287)

Berikut ini contoh perhitungan normalisasi IHSG

- Normalisasi IHSG pada 2 Januari 2018

$$X_{scaled} = \frac{6339.238 - 5452.704}{6689.287 - 5452.704}$$

$$X_{scaled} = \frac{886.534}{1236.583} = 0.71692$$

- Normalisasi IHSG pada 3 Januari 2018

$$X_{scaled} = \frac{6251.479 - 5452.704}{6689.287 - 5452.704}$$

$$X_{scaled} = \frac{798.775}{1236.583} = 0.64595$$

Dengan demikian, hasil normalisasi IHSG pada 2 Januari 2018 dan 3 Januari 2018 adalah 0.71692 dan 0.64595, sesuai dengan tabel 4.19.

4.3.1.2 Normalisasi Data IHSG pada masa Covid-19

Selanjutnya, data IHSG selama pandemi COVID-19 (periode 2 Maret 2020 – 31 Desember 2022) juga dinormalisasi dengan metode yang sama, yaitu Min-Max Scaler. Tujuan dari normalisasi ini adalah untuk memastikan bahwa skala data tetap seragam sehingga model LSTM dapat menganalisis tren pergerakan harga IHSG secara lebih efektif. Dalam periode ini, IHSG mengalami volatilitas yang lebih tinggi akibat ketidakpastian pasar yang dipicu oleh pandemi global. Ringkasan statistik deskriptif setelah normalisasi ditampilkan dalam tabel 4.20 berikut ini.

Tabel 4.20. Normalisasi Data IHSG pada Covid-19

Date	Close	Normalisasi
3/2/2020	5361.246	0.42114
3/3/2020	5518.628	0.46770
3/4/2020	5650.136	0.50660
3/5/2020	5638.130	0.50305
3/6/2020	5498.540	0.46175
⋮	⋮	⋮
12/26/2022	6835.808	0.85735
12/27/2022	6923.027	0.88315
12/28/2022	6850.520	0.86170
12/29/2022	6860.077	0.86453
12/30/2022	6850.619	0.86173

Normalisasi data menggunakan metode Min-Max Scaler dilakukan dengan menggunakan persamaan 2.33 Berikut ini variabel perhitunagn normalisasi data IHSG pada masa Covid-19

- X = Nilai asli IHSG
- X_{min} = nilai minimum IHSG pada masa Covid-19 (3937.632)
- X_{max} = nilai maksimum IHSG pada masa Covid-19 (7318.016)

Berikut ini contoh perhitungan normalisasi IHSG

- Normalisasi IHSG pada 2 Maret 2020

$$X_{scaled} = \frac{5361.246 - 3937.632}{7318.016 - 3937.632}$$

$$X_{scaled} = \frac{1423.614}{3380.384} = 0.42114$$

- Normalisasi IHSG pada 3 Maret 2020

$$X_{scaled} = \frac{5518.628 - 3937.632}{7318.016 - 3937.632}$$

$$X_{scaled} = \frac{1580.996}{3380.384} = 0.4677$$

Dengan demikian, hasil normalisasi IHSG pada 2 Maret 2020 dan 3 Maret 2020 adalah 0.42114 dan 0.4677, sesuai dengan tabel 4.20.

4.3.1.3 Normalisasi Data IHSG sebelum Covid-19

Pada tahap terakhir, data IHSG setelah pandemi COVID-19 (periode 1 Januari 2023 – 5 September 2024) dinormalisasi menggunakan Min-Max Scaler. Normalisasi ini diperlukan agar model LSTM dapat mempelajari pola pergerakan harga IHSG dengan lebih akurat. Dalam periode ini, IHSG menunjukkan tren pemulihan dan pertumbuhan setelah mengalami guncangan selama pandemi. Statistik deskriptif hasil normalisasi dapat dilihat pada tabel 4.21 berikut ini.

Tabel 4.21. Normalisasi Data IHSG setelah Covid-19

Date	Close	Normalisasi
1/2/2023	6850.984	0.25271
1/3/2023	6888.758	0.28617
1/4/2023	6813.239	0.21927
1/5/2023	6653.841	0.07806
1/6/2023	6684.558	0.10527
⋮	⋮	⋮
8/30/2024	7670.733	0.97892
9/2/2024	7694.53	1
9/3/2024	7616.521	0.93089
9/4/2024	7672.896	0.98083
9/5/2024	7681.043	0.98805

Normalisasi data menggunakan metode Min-Max Scaler dilakukan dengan menggunakan persamaan 2.33. Berikut ini variabel perhitungan normalisasi data IHSG setelah Covid-19

- X = Nilai asli IHSG

- X_{min} = nilai minimum IHSG setelah Covid-19 (6565.728)
- X_{max} = nilai maksimum IHSG setelah Covid-19 (7694.530)

Berikut ini contoh perhitungan normalisasi IHSG

- Normalisasi IHSG pada 2 Januari 2018

$$X_{scaled} = \frac{6850.984 - 6565.728}{7694.530 - 6565.728}$$

$$X_{scaled} = \frac{285.256}{1128.802} = 0.25271$$

- Normalisasi IHSG pada 3 Januari 2018

$$X_{scaled} = \frac{6888.758 - 6565.728}{7694.530 - 6565.728}$$

$$X_{scaled} = \frac{323.030}{1128.802} = 0.28617$$

5

Dengan demikian, hasil normalisasi IHSG pada 2 Januari 2022 dan 3 Januari 2022 adalah 0.25271 dan 0.28617, sesuai dengan tabel 4.21.

4.3.1 Pembagian Data Training dan Testing

Dalam penelitian ini, data harga penutupan IHSG dibagi menjadi dua bagian utama, yaitu data training dan data testing, dengan proporsi 70% untuk training dan 30% untuk testing. Pembagian ini dilakukan untuk memastikan bahwa model memiliki cukup data untuk belajar sebelum diuji pada data yang belum pernah digunakan dalam proses pelatihan. Pembagian data dilakukan untuk tiga periode utama: sebelum Covid-19, pada masa Covid-19, dan setelah Covid-19.

4.3.2.1 Pembagian Data Training dan Testing Sebelum Covid-19

Data sebelum Covid-19 terdiri dari 527 data harian, yang mencakup periode 1 Januari 2018 hingga 1 Maret 2020. Dengan pembagian 70% training dan 30% testing, maka:

- Data training: $0.7 \times 527 = 369$ data
- Data testing: $0.3 \times 527 = 158$ data

Dengan demikian, 369 data pertama digunakan untuk melatih model, sedangkan 158 data sisanya digunakan untuk pengujian.

4.3.2.2 Pembagian Data Training dan Testing pada Masa Covid-19

Data pada masa Covid-19 terdiri dari 693 data harian, yang mencakup periode 2 Maret 2020 hingga 31 Desember 2022. Dengan pembagian 70% training dan 30% testing, maka:

- Data training: $0.7 \times 693 = 486$ data
- Data testing: $0.3 \times 693 = 207$ data

Dalam hal ini, 485 data pertama digunakan untuk melatih model, sedangkan 208 data terakhir digunakan untuk pengujian.

4.3.2.3 Pembagian Data Training dan Testing Setelah Covid-19

Data setelah Covid-19 terdiri dari 398 data harian, yang mencakup periode 1 Januari 2023 hingga 5 September 2024. Dengan pembagian 70% training dan 30% testing, maka:

- Data training: $0.7 \times 398 = 279$ data
- Data testing: $0.3 \times 398 = 119$ data

Dengan demikian, 279 data pertama digunakan untuk melatih model, sedangkan 119 data terakhir digunakan untuk pengujian.

Pembagian data ini dilakukan untuk memastikan bahwa model dapat belajar dari pola historis pada setiap periode sebelum diuji dengan data yang belum pernah digunakan dalam proses pelatihan. Hal ini bertujuan untuk mengukur sejauh mana model dapat menggeneralisasi pola pergerakan IHSG dalam setiap periode ekonomi yang berbeda yang tercermin pada periode sebelum Covid-19, masa Covid-19 dan setelah Covid-19.

4.3.1 Penentuan Jumlah *Neuron* dan *Epoch* LSTM

Dalam penelitian ini, dilakukan eksperimen untuk menentukan konfigurasi terbaik dari model *Long Short-Term Memory* (LSTM) dalam melakukan prediksi terhadap pergerakan Indeks Harga

Saham Gabungan (IHSG). Salah satu faktor penting dalam membangun model LSTM yang optimal adalah menentukan jumlah *neuron* di dalam *hidden layer* dan jumlah *epoch* atau jumlah iterasi pelatihan model terhadap data.

Untuk memperoleh kombinasi terbaik, dilakukan serangkaian eksperimen dengan variasi jumlah neuron (10, 20, 30, 40, dan 50) serta jumlah *epoch* (100, 500, dan 1000). Percobaan ini dilakukan pada tiga periode waktu yang berbeda, yaitu sebelum Covid-19, masa Covid-19, dan setelah Covid-19. Pemisahan periode ini bertujuan untuk menganalisis bagaimana performa model dalam kondisi pasar yang berbeda, terutama dalam menghadapi situasi volatil seperti pandemi Covid-19 yang berdampak besar terhadap pergerakan IHSG.

4.3.3.1 Penentuan Jumlah Neuron dan Epoch Periode Sebelum Covid-19

Pada periode sebelum Covid-19, pasar cenderung bergerak stabil dengan volatilitas yang relatif lebih rendah dibandingkan saat pandemi terjadi. Hal ini memungkinkan model LSTM untuk belajar pola pergerakan IHSG dengan lebih baik. Dari hasil eksperimen, yang dapat dilihat pada tabel 4.22 nilai Mean Squared Error (MSE) terendah diperoleh ketika menggunakan 20 neuron dengan 100 epoch, yang menghasilkan MSE sebesar 14.75.

Tabel 4.22. Hasil Percobaan *Neuron* dan *Epoch* sebelum Covid-19

No	Jumlah Neuron	Epoch	MSE
1	10	100	150.67
2		500	346.41
3		1000	2812.14
4	20	100	14.75
5		500	523.01
6		1000	711.20
7	30	100	171.57
8		500	1035.26
9		1000	2547.25
10	40	100	108.48
11		500	336.24
12		1000	208.74
13	50	100	815.52
14		500	1102.69
15		1000	1551.18

Nilai MSE yang sangat rendah ini menunjukkan bahwa model mampu menangkap pola pergerakan harga IHSG dengan baik dan memprediksi harga secara akurat. Sementara itu, ketika jumlah neuron ditingkatkan menjadi 50 dan epoch mencapai 1000, MSE meningkat menjadi 1551.18, yang menunjukkan bahwa peningkatan jumlah neuron dan epoch secara berlebihan dapat menyebabkan overfitting, di mana model terlalu menyesuaikan diri dengan data pelatihan tetapi kehilangan kemampuan untuk melakukan generalisasi pada data baru.

Secara umum, hasil pada periode sebelum Covid-19 menunjukkan bahwa dengan jumlah neuron dan epoch yang tidak terlalu besar, model sudah cukup mampu menangkap pola pergerakan harga saham dengan baik.

4.3.3.2 Penentuan Jumlah *Neuron* dan *Epoch* Periode Masa Covid-19

Ketika pandemi Covid-19 terjadi, pasar saham mengalami volatilitas ekstrem yang belum pernah terjadi sebelumnya. IHSG mengalami penurunan drastis dalam waktu singkat akibat ketidakpastian ekonomi global dan kebijakan pembatasan sosial yang menghambat aktivitas ekonomi. Kondisi ini berdampak langsung pada performa model LSTM dalam melakukan prediksi.

Tabel 4.23. Hasil Percobaan *Neuron* dan *Epoch* pada masa Covid-19

No	Jumlah Neuron	Epoch	MSE
1	10	100	24384.24
2		500	34963.94
3		1000	63421.47
4	20	100	19778.53
5		500	23266.36
6		1000	115352.38
7	30	100	5105.11
8		500	26456.36
9		1000	67584.37
10	40	100	21471.05
11		500	16764.43
12		1000	129864.61
13	50	100	11986.70

14		500	11858.06
15		1000	113630.41

Dari hasil percobaan, pada tabel 4.23 terlihat bahwa nilai MSE mengalami peningkatan yang sangat signifikan dibandingkan periode sebelum Covid-19. Kombinasi terbaik diperoleh ketika menggunakan 30 neuron dengan 100 epoch, yang menghasilkan MSE sebesar 5105.11. Meskipun kombinasi ini menghasilkan nilai error yang paling rendah di antara kombinasi lainnya, nilai MSE tetap jauh lebih tinggi dibandingkan periode sebelum pandemi.

Peningkatan MSE ini menunjukkan bahwa ketidakstabilan pasar membuat model sulit untuk menangkap pola pergerakan harga saham. Dalam kondisi pasar yang sangat volatile, data historis menjadi kurang efektif untuk dijadikan dasar prediksi, karena pola pergerakan harga menjadi lebih tidak teratur dan dipengaruhi oleh faktor eksternal yang sulit diprediksi, seperti kebijakan pemerintah, sentimen investor, dan perkembangan kasus Covid-19.

Beberapa kombinasi lain dengan jumlah neuron dan epoch yang lebih besar justru menghasilkan MSE yang lebih tinggi, seperti pada model dengan 40 neuron dan 1000 epoch, yang memiliki MSE sebesar 129864.61. Hal ini menandakan bahwa pada kondisi volatilitas tinggi, model dengan parameter yang terlalu kompleks justru mengalami kesulitan dalam melakukan prediksi, kemungkinan karena overfitting terhadap noise pada data.

4.3.3.3 Penentuan Jumlah *Neuron* dan *Epoch* Periode Setelah Covid-19

Setelah pandemi mulai mereda dan perekonomian kembali berangsur pulih, volatilitas pasar saham mulai berkurang, meskipun tidak kembali sepenuhnya stabil seperti sebelum pandemi. Dalam periode ini, performa model LSTM mulai membaik dibandingkan saat pandemi, tetapi tetap menunjukkan MSE yang lebih tinggi dibandingkan sebelum pandemi terjadi.

Tabel 4.24. Hasil Percobaan *Neuron* dan *Epoch* setelah Covid-19

No	Jumlah Neuron	Epoch	MSE
1	10	100	3084.60
2		500	1817.49
3		1000	12810.91
4	20	100	1375.02

No	Jumlah Neuron	Epoch	MSE
5		500	107196.66
6		1000	94298.51
7		100	758.10
8	30	500	25667.25
9		1000	93333.12
10		100	126.96
11	40	500	23743.87
12		1000	57055.10
13		100	352.88
14	50	500	42410.32
15		1000	169680.20

Dari hasil percobaan, kombinasi terbaik diperoleh dengan 40 neuron dan 100 epoch, yang menghasilkan MSE sebesar 126.96. Nilai MSE ini jauh lebih rendah dibandingkan periode masa Covid-19, tetapi masih lebih tinggi dibandingkan periode sebelum pandemi. Hal ini menunjukkan bahwa meskipun pasar sudah lebih stabil, dampak dari pandemi masih mempengaruhi pola pergerakan IHSG, sehingga model belum dapat mencapai tingkat akurasi yang sama seperti sebelum pandemi.

Hasil lain menunjukkan bahwa penggunaan neuron yang lebih besar atau epoch yang lebih tinggi tidak selalu menghasilkan performa yang lebih baik. Sebagai contoh, model dengan 50 neuron dan 1000 epoch memiliki MSE sebesar 169680.20, yang menunjukkan adanya overfitting dan kesulitan model dalam menangkap pola pergerakan IHSG pasca-pandemi.

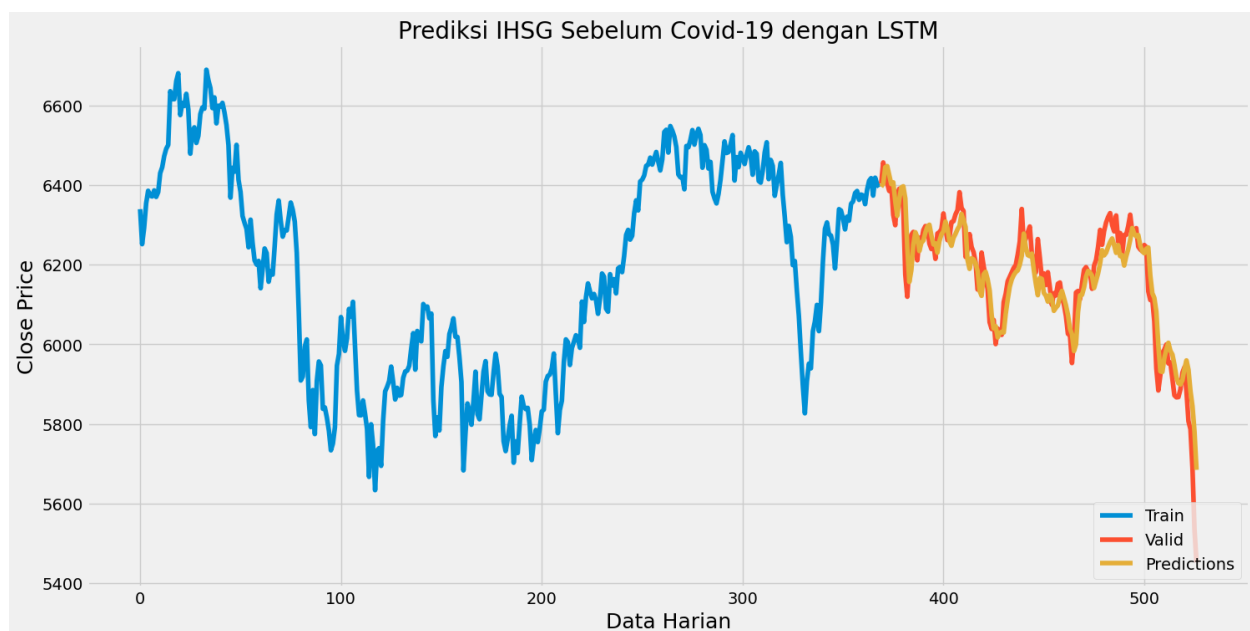
4.3.1 Hasil Prediksi IHSG Menggunakan Metode LSTM

Pada bagian ini, dilakukan prediksi terhadap Indeks Harga Saham Gabungan (IHSG) menggunakan metode Long Short-Term Memory (LSTM). Data yang digunakan dalam proses ini adalah harga penutupan (close price) IHSG, yang kemudian dibagi menjadi data training sebesar 70% dan data testing sebesar 30%. Model LSTM yang digunakan telah dikonfigurasi dengan jumlah neuron dan epoch yang optimal, sehingga menghasilkan prediksi yang lebih akurat. Setelah model dilatih dan diuji, diperoleh grafik hasil prediksi yang memperlihatkan pola pergerakan indeks, dengan garis merah sebagai data testing dan garis oranye sebagai hasil prediksi. Pola yang

terbentuk menunjukkan bahwa model LSTM dapat menangkap tren pergerakan IHSG dengan baik. Untuk mendapatkan nilai prediksi dalam skala sebenarnya, dilakukan denormalisasi pada hasil prediksi, karena sebelumnya data telah dinormalisasi menggunakan MinMax Scaler dengan rentang nilai $[0,1]$. Analisis dilakukan dengan membandingkan tiga periode utama, yaitu sebelum Covid-19, di mana pasar bergerak stabil; masa Covid-19, saat terjadi volatilitas tinggi akibat ketidakpastian ekonomi; dan setelah Covid-19, ketika pasar mulai menunjukkan pemulihan. Selanjutnya, pada bagian berikut akan dibahas hasil prediksi secara lebih rinci, termasuk perhitungan denormalisasi dan analisis tingkat akurasi model yang digunakan.

4.3.4.1 Hasil Prediksi IHSG Sebelum Covid-19

Penelitian ini memprediksi pergerakan Indeks Harga Saham Gabungan (IHSG) sebelum terjadinya pandemi COVID-19, yaitu pada periode 1 Januari 2018 hingga 1 Maret 2020. Hasil prediksi yang dihasilkan oleh model LSTM kemudian dibandingkan dengan nilai aktual IHSG untuk mengevaluasi kinerja model. Analisis ini disajikan dalam dua gambar yang menggambarkan data historis dan hasil prediksi. Model LSTM ini menggunakan jumlah neuron 20 dan epoch 100 dan didapat gambar 4.14

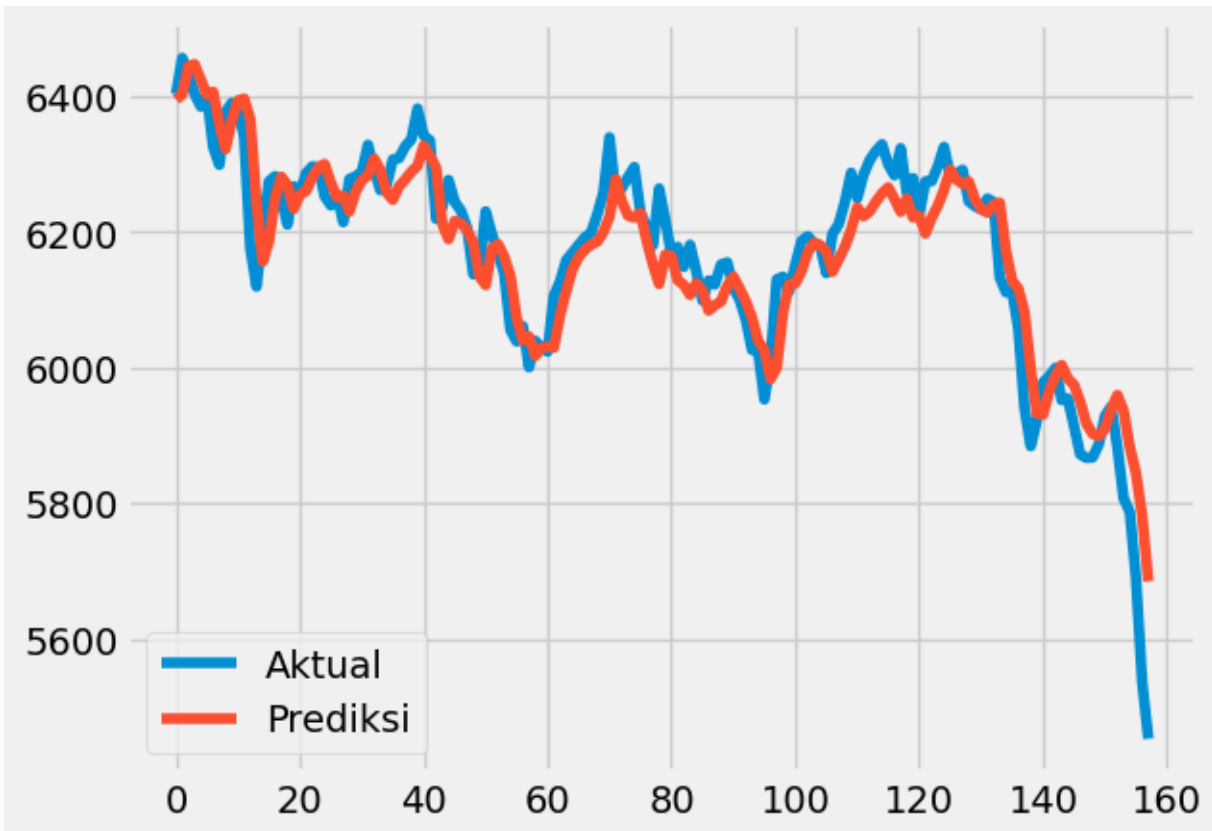


Gambar 4.14. Prediksi IHSG Menggunakan LSTM pada Periode Sebelum Pandemi COVID-19

Hasil tersebut menggambarkan perbandingan antara nilai aktual (ditandai dengan garis merah) dan hasil prediksi model LSTM (garis oranye) terhadap pergerakan IHSG dalam periode 1 Januari 2018 hingga 1 Maret 2020, yaitu sebelum pandemi COVID-19. Sumbu vertikal (y-axis) menunjukkan nilai indeks IHSG yang berkisar antara 5.000 hingga 6.400, sedangkan sumbu horizontal (x-axis) merepresentasikan data harian dalam rentang waktu tersebut.

Secara umum, tren pergerakan IHSG pada periode ini menunjukkan kestabilan dengan fluktuasi yang relatif terkendali. Model LSTM berhasil menangkap pola utama dalam pergerakan harga, terutama pada tren jangka panjang, dengan tingkat akurasi yang cukup tinggi. Hal ini dapat dilihat dari kecocokan antara garis prediksi dan nilai aktual, yang menunjukkan bahwa model mampu memahami dinamika pasar dalam kondisi yang stabil.

Namun, meskipun performa model cukup baik, terdapat deviasi kecil di beberapa titik tertentu, terutama ketika terjadi perubahan harga yang mendadak. Faktor-faktor eksternal seperti rilis data ekonomi, kebijakan moneter, atau sentimen pasar dapat menjadi penyebab utama dari ketidaksesuaian prediksi dengan nilai aktual. Model berbasis LSTM cenderung lebih optimal dalam menangkap pola tren utama, tetapi kurang responsif terhadap perubahan pasar yang bersifat jangka pendek dan volatilitas yang tiba-tiba.



Gambar 4.15. Analisis Perbandingan Prediksi dan Aktual IHSR Sebelum Pandemi COVID-19

Gambar 4.15 memberikan visualisasi yang lebih rinci mengenai perbandingan antara nilai aktual dan hasil prediksi pergerakan IHSR sebelum pandemi COVID-19, melengkapi analisis yang ditampilkan pada Gambar 4.14. Dalam grafik ini, rentang sumbu vertikal dibatasi antara 5.600 hingga 6.400, memberikan fokus lebih tajam terhadap pergerakan harga yang terjadi dalam periode tersebut.

Secara umum, garis biru (nilai aktual) dan garis merah (hasil prediksi) menunjukkan keselarasan yang cukup baik, terutama dalam tren jangka panjang seperti kenaikan bertahap maupun penurunan yang terkendali. Hal ini menegaskan bahwa model LSTM memiliki kemampuan yang kuat dalam mengidentifikasi pola makro dari pergerakan harga saham. Namun, ketika terjadi perubahan yang lebih cepat dan tajam, seperti lonjakan atau penurunan harga secara tiba-tiba, terlihat bahwa model cenderung mengalami keterlambatan dalam menangkap pergerakan tersebut.

Ketertinggalan model dalam merespons perubahan harga yang bersifat mendadak mengindikasikan bahwa LSTM lebih optimal dalam memprediksi tren jangka menengah hingga panjang, tetapi masih memiliki keterbatasan dalam menangkap volatilitas jangka pendek. Faktor eksternal seperti perubahan sentimen pasar, kebijakan ekonomi, atau peristiwa global yang tidak terduga dapat menjadi penyebab utama ketidaktepatan prediksi dalam periode dengan volatilitas tinggi.

Berdasarkan hasil yang ditampilkan dalam Gambar 4.14 dan Gambar 4.15, dapat disimpulkan bahwa model LSTM memberikan performa prediksi yang cukup baik dalam kondisi pasar yang relatif stabil sebelum pandemi COVID-19. Akurasi model cukup tinggi dalam mengikuti pola pergerakan harga historis, menjadikannya alat yang efektif untuk analisis time series dalam pasar saham. Namun, model ini masih perlu penyempurnaan, terutama dalam menangani perubahan harga yang lebih dinamis dan ekstrem, seperti yang terjadi menjelang awal pandemi pada tahun 2020. Hal ini menunjukkan bahwa meskipun LSTM memiliki keunggulan dalam menangkap pola jangka panjang, pendekatan tambahan atau kombinasi dengan teknik lain dapat meningkatkan akurasi dalam menghadapi kondisi pasar yang lebih kompleks dan volatil.

Hasil prediksi yang diperoleh sebelumnya telah melalui proses normalisasi menggunakan metode Min-Max Scaler dengan rentang nilai minimum dan maksimum sebesar 5452.704 dan 6689.287. Untuk mendapatkan kembali nilai prediksi dalam skala aslinya, dilakukan proses denormalisasi menggunakan persamaan 2.34

Sebagai contoh, untuk data pada tanggal 18 Juli 2019 dengan nilai prediksi Min-Max Scaler sebesar 0.76233, maka hasil denormalisasi adalah:

$$x_{18 \text{ Juli}} = 0.76233 (6689.287 - 5452.704) + 5452.704 = 6395.389$$

Selanjutnya, untuk data pada tanggal 24 Juli 2019 dengan nilai prediksi Min-Max Scaler sebesar 0.78602, hasil denormalisasi adalah:

$$x_{24 \text{ Juli}} = 0.78602 (6689.287 - 5452.704) + 5452.704 = 6424.683$$

Berdasarkan perhitungan sebelumnya dan analisis lebih lanjut terhadap data yang tersedia, diperoleh hasil prediksi yang dirangkum dalam Tabel 4.25 di bawah ini. Tabel ini menyajikan data sebelum periode pandemi COVID-19, termasuk nilai yang telah dinormalisasi menggunakan Min-

Max Scaler serta hasil prediksi menggunakan model LSTM. Dengan tingkat akurasi yang tinggi, tabel ini memberikan gambaran mengenai pola perubahan data yang terjadi akibat dampak pandemi serta tren pemulihannya di periode setelahnya.

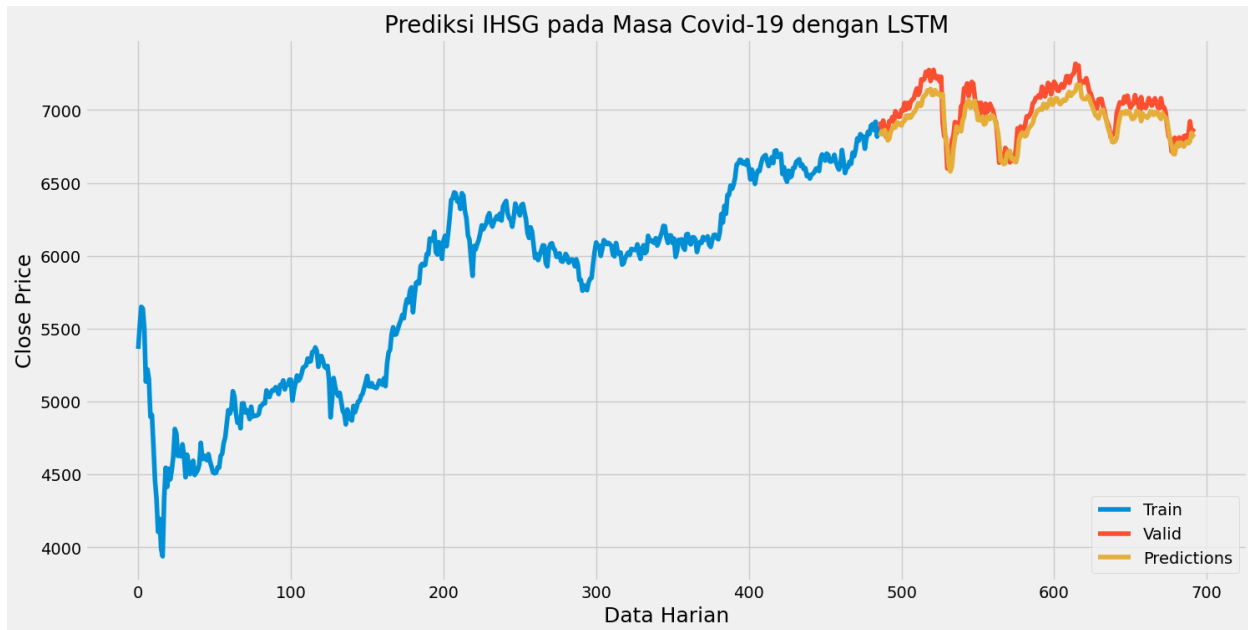
Tabel 4.25. Perhitungan Prediksi IHSG Sebelum Pandemi Covid-19

No	Tanggal	Close	Min-Max Scaler	Predictions
1	7/18/2019	6403.294	0.76233	6395.389
2	7/19/2019	6456.539	0.76835	6402.831
3	7/22/2019	6433.547	0.80202	6444.468
4	7/23/2019	6403.810	0.80437	6447.370
5	7/24/2019	6384.987	0.78602	6424.683
⋮	⋮	⋮	⋮	⋮
154	2/24/2020	5807.050	0.39118	5936.432
155	2/25/2020	5787.138	0.34761	5882.556
156	2/26/2020	5688.921	0.31677	5844.421
157	2/27/2020	5535.694	0.26552	5781.041
158	2/28/2020	5452.704	0.18776	5684.886

Setelah dilakukan denormalisasi pada tabel prediksi, diperoleh hasil evaluasi model menggunakan Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), dan Sum of Squared Errors (SSE). Pada periode sebelum pandemi, model menunjukkan kinerja prediksi yang sangat baik dengan MSE sebesar 14.75, MAPE 0.74%, dan RMSE 3.840. Nilai kesalahan yang rendah menunjukkan bahwa pola harga aset masih bergerak stabil dan dapat diprediksi dengan baik oleh model. Hal ini menunjukkan bahwa model dapat menangkap tren harga secara akurat dalam kondisi pasar yang relatif normal dan terkendali.

4.3.4.2 Prediksi IHSG pada masa Covid-19

Penelitian ini juga melakukan prediksi terhadap pergerakan IHSG selama masa pandemi COVID-19 dari 2 Maret 2020 sampai 31 Desember 2022. Periode ini memiliki karakteristik yang berbeda karena adanya ketidakpastian dan volatilitas pasar yang tinggi akibat dampak pandemi. Model LSTM kembali digunakan untuk memprediksi pergerakan IHSG dan hasilnya dibandingkan dengan nilai aktual. Model LSTM ini menggunakan jumlah neuron 30 dan epoch 100 dan didapat gambar 4.16 dibawah ini



Gambar 4.16. Prediksi IHSG Menggunakan LSTM pada Periode pada Masa Pandemi COVID-

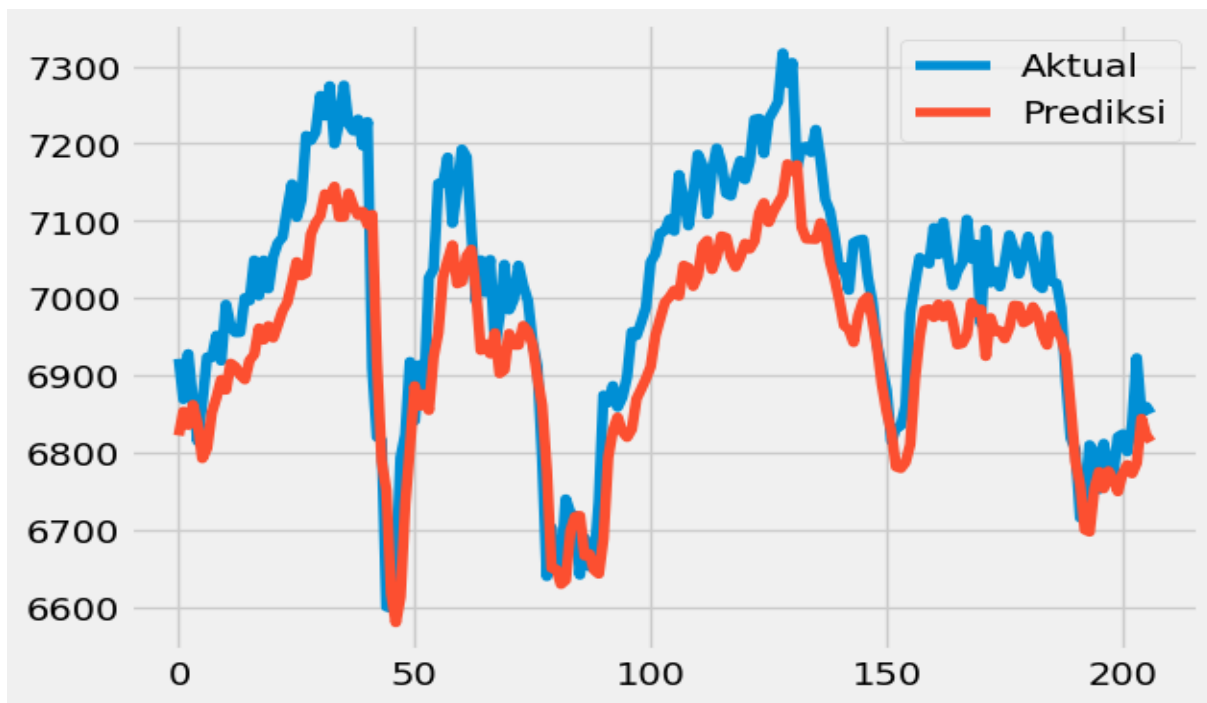
19

Hasil ini menampilkan prediksi pergerakan IHSG selama masa pandemi COVID-19, di mana volatilitas pasar mengalami peningkatan signifikan akibat ketidakpastian ekonomi dan krisis kesehatan global. Pada sumbu vertikal, rentang nilai IHSG berkisar antara 4.000 hingga 7.300, menggambarkan dinamika pasar yang sangat fluktuatif selama periode tersebut.

Garis merah mewakili nilai aktual IHSG, yang memperlihatkan penurunan tajam di awal pandemi, diikuti dengan fase pemulihan bertahap seiring dengan intervensi kebijakan ekonomi dan adaptasi pasar terhadap kondisi baru. Sementara itu, garis oranye yang mewakili prediksi model LSTM menunjukkan kemampuan dalam menangkap tren pemulihan secara umum. Namun, terlihat bahwa deviasi antara nilai aktual dan prediksi lebih besar dibandingkan periode sebelum pandemi, terutama dalam menghadapi perubahan harga yang mendadak.

Hal ini menegaskan bahwa meskipun model LSTM cukup baik dalam mengidentifikasi pola jangka panjang, kinerjanya menurun ketika dihadapkan pada kejadian tak terduga seperti pengumuman kebijakan lockdown, stimulus ekonomi, atau lonjakan kasus COVID-19. Model tampaknya kesulitan dalam menangkap respons pasar terhadap faktor eksternal yang bersifat mendadak dan tidak terduga.

Hasil ini menunjukkan bahwa LSTM tetap mampu memberikan prediksi yang relatif baik dalam kondisi pasar yang bergejolak, meskipun akurasinya berkurang ketika volatilitas meningkat. Untuk meningkatkan performa model dalam situasi seperti ini, pendekatan yang lebih adaptif misalnya dengan memasukkan variabel makroekonomi atau menggunakan teknik ensemble learning dapat menjadi strategi yang lebih efektif dalam menangkap perubahan pasar yang cepat dan ekstrem.



Gambar 4.17. Analisis Perbandingan Prediksi dan Aktual IHSX pada Masa Pandemi COVID-19

Gambar 4.17 menyajikan analisis lebih mendalam mengenai perbandingan antara nilai aktual dan prediksi IHSX selama masa pandemi COVID-19. Dibandingkan dengan grafik sebelumnya, visualisasi ini menawarkan cakupan yang lebih spesifik, dengan rentang sumbu vertikal yang terbatas pada 6.600 hingga 7.300, sehingga memungkinkan pengamatan lebih rinci terhadap dinamika pergerakan pasar. Sumbu horizontal tetap merepresentasikan waktu dengan interval yang lebih detail, memberikan gambaran yang lebih akurat tentang bagaimana model LSTM menangkap pola pergerakan IHSX.

Dalam grafik ini, garis biru masih merepresentasikan nilai aktual, sedangkan garis merah menggambarkan prediksi yang dihasilkan oleh model LSTM. Secara umum, terlihat bahwa model mampu mengikuti tren pasar dengan cukup baik, terutama ketika pergerakan harga berlangsung

dalam pola yang relatif stabil. Namun, pada periode dengan volatilitas tinggi—seperti saat terjadi lonjakan atau penurunan tajam—deviasi antara prediksi dan nilai aktual menjadi lebih signifikan.

Meskipun demikian, model LSTM tetap menunjukkan ketangguhan dalam memprediksi tren pasar meskipun dihadapkan pada kondisi ekonomi yang sangat tidak stabil akibat pandemi. Kemampuannya dalam mengenali pola historis memberikan gambaran tentang arah pergerakan harga, meskipun belum sepenuhnya akurat dalam menangkap perubahan mendadak yang sering kali dipicu oleh faktor eksternal, seperti kebijakan fiskal dan moneter, pengumuman terkait pandemi, serta dinamika sentimen pasar global.

Perbedaan yang lebih besar antara nilai aktual dan prediksi selama periode ini mengindikasikan bahwa model menghadapi tantangan lebih kompleks dalam menghadapi perubahan drastis yang tidak dapat sepenuhnya diprediksi dengan data historis semata. Hal ini menunjukkan perlunya pendekatan prediksi yang lebih canggih, misalnya dengan mengintegrasikan variabel makroekonomi, indikator sentimen pasar, atau bahkan teknik machine learning berbasis hybrid untuk meningkatkan akurasi model dalam menghadapi situasi pasar yang penuh ketidakpastian.

Hasil prediksi yang diperoleh sebelumnya telah melalui proses normalisasi menggunakan metode Min-Max Scaler dengan rentang nilai minimum dan maksimum sebesar 3937.632 dan 7318.016. Untuk mendapatkan kembali nilai prediksi dalam skala aslinya, dilakukan proses denormalisasi menggunakan persamaan 3.34

Sebagai contoh, untuk data pada tanggal 1 Maret 2022 dengan nilai prediksi Min-Max Scaler sebesar 0.85329, maka hasil denormalisasi adalah:

$$x_{1 \text{ Maret}} = 0.85329 (7318.016 - 3937.632) + 3937.632 = 6822.078$$

Selanjutnya, untuk data pada tanggal 8 Maret 2022 dengan nilai prediksi Min-Max Scaler sebesar 0.85796, hasil denormalisasi adalah:

$$x_{8 \text{ Maret}} = 0.85796 (7318.016 - 3937.632) + 3937.632 = 6837.882$$

Berdasarkan perhitungan sebelumnya dan analisis lebih lanjut terhadap data yang tersedia, diperoleh hasil prediksi yang dirangkum dalam Tabel 4.26 di bawah ini. Tabel ini menyajikan data selama periode pandemi COVID-19, termasuk nilai yang telah dinormalisasi menggunakan Min-Max Scaler serta hasil prediksi menggunakan model LSTM. Dengan tingkat akurasi yang tinggi,

tabel ini memberikan gambaran mengenai pola perubahan data yang terjadi akibat dampak pandemi serta tren pemulihannya di periode setelahnya.

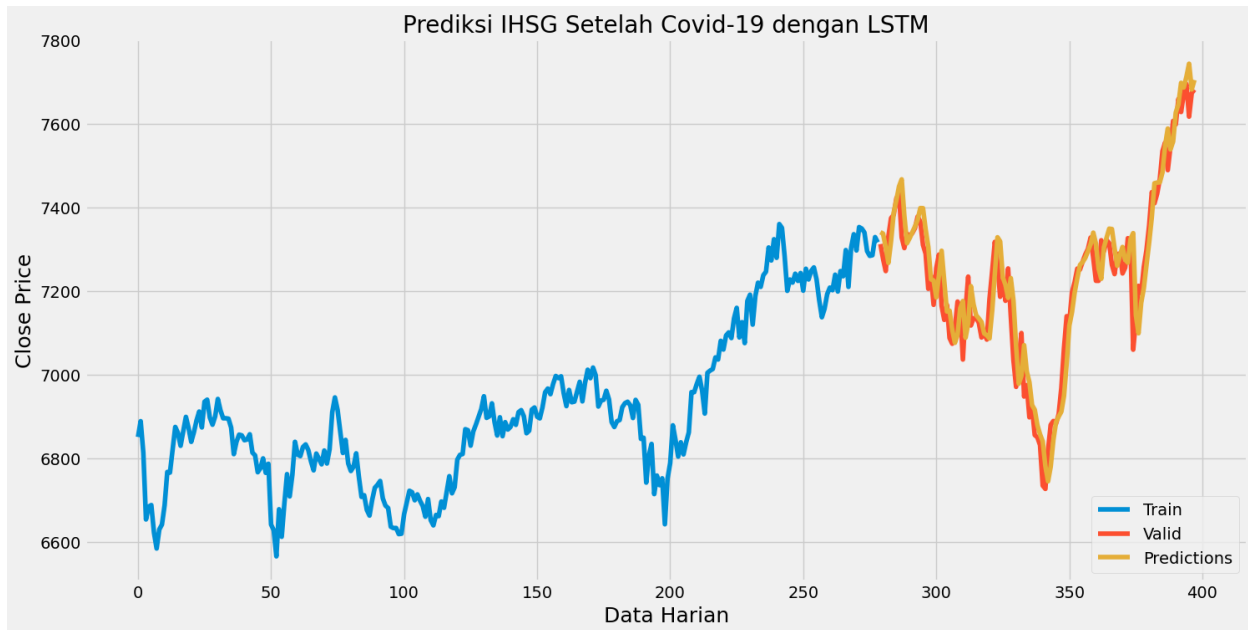
Tabel 4.26. Perhitungan Prediksi IHSG Pada Masa Pandemi Covid-19

No	Tanggal	Close	Min-Max Scaler	Predictions
1	3/1/2022	6921.441	0.85329	6822.078
2	3/2/2022	6868.402	0.86267	6853.778
3	3/4/2022	6928.328	0.85709	6834.909
4	3/7/2022	6869.066	0.86528	6862.595
5	3/8/2022	6814.183	0.85796	6837.882
⋮	⋮	⋮	⋮	⋮
203	12/26/2022	6835.808	0.83856	6772.299
204	12/27/2022	6923.027	0.84270	6786.295
205	12/28/2022	6850.520	0.85995	6844.599
206	12/29/2022	6860.077	0.85348	6822.719
207	12/30/2022	6850.619	0.85095	6814.166

Setelah dilakukan denormalisasi pada tabel prediksi, diperoleh hasil evaluasi model menggunakan Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), dan Sum of Squared Errors (SSE). Pada masa pandemi, terjadi peningkatan volatilitas yang signifikan dalam pergerakan harga aset, sehingga model mengalami kesulitan dalam memprediksi tren dengan akurasi tinggi. Hal ini tercermin dari MSE yang meningkat drastis menjadi 5105.11, MAPE naik menjadi 1.16%, dan RMSE sebesar 71.450. Ketidakpastian global akibat pandemi, kebijakan pemerintah, serta perubahan perilaku investor menyebabkan harga aset berfluktuasi tajam, yang membuat prediksi menjadi lebih sulit dan kurang akurat dibandingkan dengan kondisi pasar yang stabil.

4.3.4.3 Prediksi IHSG Setelah Covid-19

Penelitian ini juga melakukan prediksi terhadap pergerakan IHSG pada periode setelah COVID-19, yaitu dari 1 Januari 2023 hingga 5 September 2024. Periode ini dipilih untuk melihat bagaimana model LSTM dapat memprediksi pergerakan IHSG dalam kondisi pasar yang relatif stabil setelah masa krisis. Model LSTM ini menggunakan jumlah neuron 40 dan epoch 100 dan didapat gambar 4.18 dibawah ini



Gambar 4.18. Prediksi IHSG Menggunakan LSTM pada Periode Setelah Pandemi COVID-19

Gambar 4.18 menggambarkan pergerakan IHSG pada periode pasca-pandemi, ketika kondisi pasar mulai menunjukkan tanda-tanda stabilisasi setelah mengalami turbulensi akibat krisis global. Dalam fase ini, meskipun masih terjadi fluktuasi, volatilitasnya lebih terkendali dibandingkan dengan masa pandemi. Grafik ini menampilkan perbandingan antara nilai aktual IHSG (garis merah) dan hasil prediksi menggunakan model LSTM (garis oranye), dengan rentang harga yang mencerminkan tren pemulihan pasar.

Terlihat bahwa model LSTM mampu menangkap pola pergerakan harga dengan cukup baik, terutama dalam tren kenaikan dan penurunan bertahap. Keselarasan antara prediksi dan nilai aktual lebih baik dibandingkan periode saat pandemi, yang menunjukkan bahwa model ini bekerja lebih optimal dalam kondisi pasar yang lebih stabil dan dapat diprediksi. Dalam lingkungan yang lebih teratur, model dapat lebih mudah mengenali pola historis dan menerapkannya dalam prediksi jangka pendek hingga menengah.

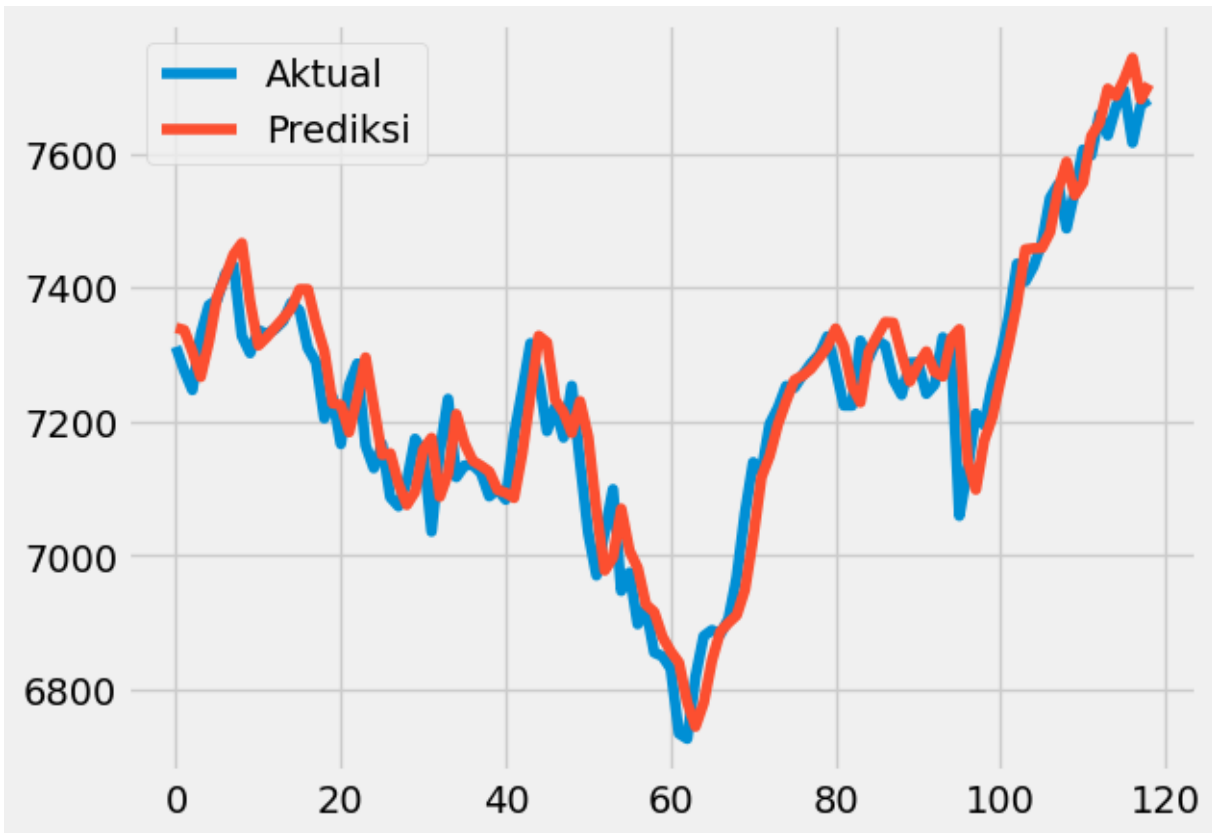
Namun, meskipun performanya meningkat, masih terdapat beberapa titik deviasi, terutama ketika pasar bereaksi terhadap kejadian eksternal yang tiba-tiba. Faktor-faktor seperti perubahan kebijakan moneter, ketidakpastian geopolitik, sentimen investor global, dan pergerakan ekonomi makro tetap menjadi tantangan bagi model dalam memprediksi lonjakan atau penurunan harga

yang tidak terduga. Hal ini menyoroti keterbatasan LSTM dalam menangkap anomali pasar yang tidak dapat sepenuhnya diwakili oleh data historis.

Hasil ini menegaskan bahwa model berbasis deep learning seperti LSTM sangat berguna dalam kondisi pasar yang relatif stabil, tetapi tetap memiliki keterbatasan dalam menghadapi kejadian yang bersifat outlier atau eksogen. Untuk meningkatkan akurasi prediksi, pendekatan yang lebih komprehensif diperlukan, seperti menggabungkan analisis teknikal dan fundamental, serta mempertimbangkan faktor-faktor non-linear seperti sentimen pasar dan berita ekonomi global.

Gambar 4.19 menyajikan analisis yang lebih mendalam mengenai perbandingan antara nilai aktual dan hasil prediksi IHSG pada periode setelah pandemi COVID-19. Dalam grafik ini, sumbu vertikal difokuskan pada rentang nilai IHSG antara 6.800 hingga 7.600, sementara sumbu horizontal merepresentasikan waktu dengan interval yang lebih detail. Garis biru menunjukkan nilai aktual IHSG, sementara garis merah merepresentasikan prediksi model LSTM.

Hasil visualisasi ini mengindikasikan bahwa model LSTM mampu menangkap pola pergerakan IHSG dengan tingkat akurasi yang cukup baik, terutama pada periode ketika pasar berada dalam tren yang stabil. Model dapat mengikuti pergerakan harga secara berkelanjutan, menunjukkan bahwa ia dapat mengenali pola historis dan menerapkannya untuk memprediksi tren masa depan. Keselarasan antara prediksi dan nilai aktual semakin jelas ketika pasar bergerak secara teratur, mencerminkan efektivitas LSTM dalam kondisi pasar yang lebih dapat diprediksi.



Gambar 4.19. Analisis Perbandingan Prediksi dan Aktual IHSG Setelah Pandemi COVID-19

Namun, terdapat beberapa periode di mana prediksi model mengalami deviasi yang cukup signifikan dibandingkan nilai aktual, terutama saat terjadi fluktuasi mendadak. Hal ini menyoroti keterbatasan LSTM dalam menangkap pergerakan harga yang dipengaruhi oleh faktor eksternal yang tidak tercermin dalam data historis. Faktor-faktor seperti perubahan kebijakan moneter, sentimen investor global, volatilitas pasar keuangan internasional, atau kejadian geopolitik dapat menyebabkan pergerakan harga yang tidak terduga, yang sulit diantisipasi oleh model berbasis time series seperti LSTM.

Hasil ini memperkuat temuan bahwa model LSTM memiliki kinerja yang solid dalam kondisi pasar yang lebih stabil, tetapi tetap memiliki keterbatasan dalam menghadapi dinamika pasar yang bersifat non-linear. Oleh karena itu, untuk meningkatkan akurasi prediksi dalam kondisi yang lebih kompleks, pendekatan hybrid dengan mengintegrasikan variabel makroekonomi, analisis sentimen

pasar, atau metode machine learning lainnya dapat menjadi solusi yang lebih adaptif terhadap perubahan pasar yang tidak terduga.

Hasil prediksi yang diperoleh sebelumnya telah melalui proses normalisasi menggunakan metode Min-Max Scaler dengan rentang nilai minimum dan maksimum sebesar 6565.728 dan 7694.530. Untuk mendapatkan kembali nilai prediksi dalam skala aslinya, dilakukan proses denormalisasi menggunakan persamaan 3.34

Sebagai contoh, untuk data pada tanggal 1 Maret 2024 dengan nilai prediksi Min-Max Scaler sebesar 0.686256, maka hasil denormalisasi adalah:

$$x_{1 \text{ Maret}} = 0.686256 (7694.530 - 6565.728) + 6565.728 = 7340.375$$

Selanjutnya, untuk data pada tanggal 7 Maret 2024 dengan nilai prediksi Min-Max Scaler sebesar 0.667712, hasil denormalisasi adalah:

$$x_{7 \text{ Maret}} = 0.667712 (7694.530 - 6565.728) + 6565.728 = 7319.442$$

Berdasarkan perhitungan sebelumnya dan analisis lebih lanjut terhadap data yang tersedia, diperoleh hasil prediksi yang dirangkum dalam Tabel 4.27 di bawah ini. Tabel ini menyajikan data setelah periode pandemi COVID-19, termasuk nilai yang telah dinormalisasi menggunakan Min-Max Scaler serta hasil prediksi menggunakan model LSTM. Dengan tingkat akurasi yang tinggi, tabel ini memberikan gambaran mengenai pola perubahan data yang terjadi akibat dampak pandemi serta tren pemulihannya di periode setelahnya.

Tabel 4.27. Perhitungan Prediksi IHSG Setelah Pandemi Covid-19

No	Tanggal	Close	Min-Max Scaler	Predictions
1	3/1/2024	7311.907	0.686256	7340.375
2	3/4/2024	7276.749	0.683177	7336.899
3	3/5/2024	7247.460	0.654937	7305.022
4	3/6/2024	7329.801	0.621205	7266.945
5	3/7/2024	7373.964	0.667712	7319.442
:	:	:	:	:
114	8/29/2024	7627.604	1.002378	7697.213
115	8/30/2024	7670.733	0.992623	7686.202
116	9/2/2024	7694.530	1.014956	7711.413
117	9/3/2024	7616.521	1.042907	7742.963

No	Tanggal	Close	Min-Max Scaler	Predictions
118	9/4/2024	7672.896	0.988381	7681.415
119	9/5/2024	7681.043	1.008466	7704.086

Setelah dilakukan denormalisasi pada tabel prediksi, diperoleh hasil evaluasi model menggunakan Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), dan Sum of Squared Errors (SSE). Setelah pandemi, kondisi pasar mulai kembali stabil dengan volatilitas yang lebih rendah dibandingkan saat pandemi. Hal ini tercermin dari perbaikan nilai error model, dengan MSE turun menjadi 126.96, MAPE kembali menurun menjadi 0.71%, dan RMSE sebesar 11.267. Hal ini menandakan bahwa model kembali mampu menangkap pola harga dengan lebih baik setelah pasar menunjukkan tanda-tanda pemulihan.

Berdasarkan hasil evaluasi model pada tiga periode tersebut, terlihat bahwa model prediksi berbasis LSTM bekerja sangat baik dalam kondisi pasar yang stabil (sebelum dan setelah pandemi), tetapi mengalami kesulitan dalam menangkap pola pergerakan harga selama periode volatilitas tinggi seperti saat pandemi COVID-19. Peningkatan nilai error selama pandemi menunjukkan bahwa faktor eksternal yang tidak terduga dapat mempengaruhi kinerja model secara signifikan. Namun, setelah kondisi pasar kembali stabil, model mampu memprediksi harga dengan lebih baik, menunjukkan bahwa pendekatan ini tetap efektif dalam analisis harga aset jangka panjang.

Hasil ini sejalan dengan temuan Li et al. (2023) yang menunjukkan bahwa model LSTM sering kali lebih efektif dalam kondisi pasar yang lebih stabil, tetapi mengalami tantangan dalam menangkap perubahan mendadak akibat volatilitas ekstrem. Mereka menemukan bahwa kombinasi metode seperti hybrid ARIMA-GARCH dan LSTM dapat meningkatkan akurasi prediksi karena mampu menangkap pola linear dan non-linear secara bersamaan. Studi ini juga menemukan bahwa penggunaan optimasi seperti Sine-Cosine Algorithm (SCA) dapat meningkatkan kinerja LSTM dalam menghadapi volatilitas tinggi, suatu pendekatan yang dapat menjadi alternatif untuk meningkatkan performa model di masa depan.

Selain itu, penelitian yang dilakukan oleh Gao et al. (2021) menunjukkan bahwa model LSTM memiliki keterbatasan dalam menangani volatilitas tinggi, yang dapat menyebabkan peningkatan error prediksi selama periode ketidakpastian pasar. Dalam studi mereka, model LSTM yang

dioptimalkan dengan Particle Swarm Optimization (PSO) menunjukkan perbaikan dalam beberapa aspek, tetapi masih kalah dalam menangkap volatilitas dibandingkan dengan pendekatan berbasis ARIMA-GARCH. Hal ini mendukung temuan dalam penelitian ini, di mana ARIMA-GARCH cukup kesulitan dalam menangani volatilitas, terutama selama pandemi.

Lebih lanjut, penelitian oleh Lakshminarayanan et al. (2019) membandingkan berbagai pendekatan prediksi harga saham dan menemukan bahwa hybrid ARIMA-GARCH dengan LSTM dapat memberikan hasil yang lebih akurat dibandingkan model LSTM murni, terutama ketika pasar mengalami guncangan besar. Studi mereka menyoroti pentingnya kombinasi metode statistik dan machine learning untuk menangani dinamika pasar yang kompleks. Hal ini semakin menegaskan bahwa dalam kondisi pasar yang sangat volatil, model hybrid yang menggabungkan pendekatan tradisional dan deep learning dapat menjadi solusi yang lebih optimal dibandingkan dengan pendekatan tunggal.

4.4 Perbandingan Model ARIMA-GARCH dengan Model LSTM

Pemodelan time series dalam dunia keuangan menghadapi tantangan yang besar, terutama dalam menangkap pola volatilitas yang terjadi akibat dinamika pasar. Untuk mengevaluasi efektivitas model dalam menangani pergerakan harga, dilakukan perbandingan antara model ARIMA-GARCH, yang menggabungkan pendekatan klasik dalam analisis data deret waktu, dan model Long Short-Term Memory (LSTM), yang merupakan salah satu bentuk deep learning yang banyak digunakan dalam peramalan time series. Evaluasi ini dilakukan berdasarkan tiga metrik utama, yaitu Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), dan Root Mean Squared Error (RMSE), dengan pengujian yang dibagi ke dalam tiga periode: sebelum COVID-19, selama COVID-19, dan setelah COVID-19.

Tabel 4.28. Perbandingan Akurasi Model ARIMA-GARCH dan LSTM

	Model ARIMA-GARCH			Model LSTM		
	Sebelum	Masa	Setelah	Sebelum	Masa	Setelah
MSE	4.887	0.735	60.159	14.748	5105.110	126.956

MAPE (%)	0.035	0.012	0.107	0.745	1.157	0.711
RMSE	2.211	0.858	7.756	3.840	71.450	11.267

4.4.1 Perbandingan Pada Periode Sebelum COVID-19

Sebelum pandemi COVID-19, kondisi pasar cenderung lebih stabil dengan tingkat volatilitas yang relatif lebih rendah dibandingkan dua periode lainnya. Dalam kondisi ini, model ARIMA-GARCH mampu menangkap pola pergerakan harga dengan sangat baik, sebagaimana ditunjukkan oleh nilai MSE sebesar 4.887, MAPE sebesar 0.035%, dan RMSE sebesar 2.211. Nilai-nilai ini mengindikasikan bahwa kesalahan peramalan yang dihasilkan oleh model ini sangat kecil, yang berarti model dapat merepresentasikan tren historis dengan akurasi yang tinggi.

Sebaliknya, model LSTM mengalami kesulitan dalam menangkap pola pergerakan harga pada periode ini, sebagaimana ditunjukkan oleh nilai MSE sebesar 14.748, MAPE sebesar 0.745%, dan RMSE sebesar 3.840. Kesalahan yang jauh lebih besar ini menunjukkan bahwa meskipun LSTM memiliki kemampuan untuk mengenali pola jangka panjang, model ini kurang optimal dalam menangani data dengan fluktuasi rendah dan perubahan yang lebih halus seperti yang terjadi sebelum pandemi. Kemungkinan besar, hal ini terjadi karena LSTM memerlukan lebih banyak data untuk dapat memahami pola dengan baik, sementara ARIMA-GARCH lebih unggul dalam menangkap pola linear dari data historis yang relatif stabil.

Dengan demikian, dalam kondisi pasar yang cenderung stabil, ARIMA-GARCH lebih superior dibandingkan LSTM, karena model ini dapat dengan lebih baik memprediksi pergerakan harga dengan tingkat kesalahan yang jauh lebih rendah.

4.4.1 Perbandingan Pada Periode Selama COVID-19

Periode pandemi COVID-19 membawa perubahan yang sangat signifikan dalam pola pergerakan harga, dengan tingkat volatilitas yang jauh lebih tinggi akibat ketidakpastian ekonomi global. Pada kondisi ini, ARIMA-GARCH tetap mampu beradaptasi dengan cukup baik, dengan MSE sebesar 0.735, MAPE sebesar 0.012%, dan RMSE sebesar 0.858. Nilai-nilai ini menunjukkan bahwa

model masih dapat menangkap pola utama dalam data, meskipun menghadapi tantangan dalam menangani lonjakan volatilitas yang terjadi secara tiba-tiba.

Sebaliknya, model LSTM mengalami kesulitan yang sangat besar dalam menghadapi periode ini, sebagaimana ditunjukkan oleh lonjakan nilai error yang sangat signifikan: MSE sebesar 5105.110, MAPE sebesar 1.157%, dan RMSE sebesar 71.450. Hasil ini menunjukkan bahwa LSTM gagal mengakomodasi perubahan pasar yang sangat dinamis selama pandemi. Salah satu kemungkinan penyebabnya adalah bahwa LSTM, sebagai model berbasis deep learning, sangat bergantung pada pola yang telah dipelajari sebelumnya dan kurang fleksibel dalam menangani perubahan struktural yang tiba-tiba dalam data.

Perbedaan ini semakin memperjelas bahwa dalam kondisi pasar yang sangat fluktuatif, ARIMA-GARCH tetap menjadi pilihan yang lebih andal dibandingkan LSTM, karena model ini lebih mampu menyesuaikan diri dengan perubahan pola volatilitas tanpa kehilangan akurasi prediksi secara signifikan.

4.4.1 Periode Setelah COVID-19

Setelah pandemi, pasar mulai menunjukkan tanda-tanda pemulihan, tetapi masih mengalami tingkat volatilitas yang cukup tinggi. Pada periode ini, performa ARIMA-GARCH tetap lebih baik dibandingkan LSTM, dengan MSE sebesar 60.159, MAPE sebesar 0.107%, dan RMSE sebesar 7.756. Meskipun nilai error mengalami peningkatan dibandingkan periode sebelumnya, model ini masih mampu menangkap pola utama dalam data dan memberikan hasil peramalan yang relatif akurat.

Sementara itu, model LSTM masih menunjukkan performa yang kurang optimal dengan MSE sebesar 126.956, MAPE sebesar 0.711%, dan RMSE sebesar 11.267. Kesalahan yang lebih besar ini menunjukkan bahwa meskipun kondisi pasar sudah lebih stabil dibandingkan saat pandemi, model LSTM masih mengalami kesulitan dalam menyesuaikan diri dengan pola pergerakan harga yang baru. Salah satu faktor yang mungkin mempengaruhi adalah overfitting pada data pelatihan, yang menyebabkan model LSTM kurang fleksibel dalam menghadapi perubahan tren yang berbeda dari pola historisnya.

Hasil ini semakin menegaskan bahwa dalam periode pemulihan setelah pandemi, ARIMA-GARCH masih memberikan hasil yang lebih akurat dibandingkan LSTM, karena model ini lebih mampu menangkap pola volatilitas yang masih tinggi pasca-krisis.

Berdasarkan evaluasi dalam tiga periode tersebut, ARIMA-GARCH secara konsisten menunjukkan performa yang lebih baik dibandingkan dengan LSTM. Model ini memiliki tingkat kesalahan yang lebih rendah dalam semua metrik evaluasi (MSE, MAPE, dan RMSE), baik sebelum, selama, maupun setelah pandemi.

Kelebihan utama dari ARIMA-GARCH adalah kemampuannya dalam menangkap pola linear dan volatilitas dalam data finansial, yang membuatnya lebih stabil dalam berbagai kondisi pasar. Sebaliknya, LSTM mengalami kesulitan dalam menyesuaikan diri dengan perubahan drastis dalam data, terutama dalam kondisi volatilitas tinggi selama pandemi dan periode pemulihan setelahnya. Hal ini kemungkinan besar disebabkan oleh ketergantungan LSTM pada jumlah data yang besar dan tantangan dalam tuning parameter yang optimal.

Berdasarkan hasil penelitian ini, model ARIMA-GARCH terbukti lebih unggul dibandingkan model LSTM dalam memprediksi pergerakan IHSG sebelum, selama, dan setelah pandemi COVID-19. Namun, untuk memahami lebih dalam implikasi hasil ini, perlu dilakukan perbandingan dengan beberapa penelitian sebelumnya. Artikel oleh Mehtarizadeh et al. (2025) menunjukkan bahwa model hybrid yang menggabungkan ARIMA-GARCH dan LSTM-SCA mampu meningkatkan akurasi prediksi dengan menangkap pola linear dan non-linear secara simultan. Model ini mengoptimalkan parameter LSTM menggunakan Sine-Cosine Algorithm (SCA) sehingga dapat mengatasi tantangan dalam memodelkan volatilitas tinggi pada pasar keuangan.

Selanjutnya, studi oleh Tran et al. (2024) membandingkan berbagai metode pembelajaran mesin dalam prediksi harga saham dan menemukan bahwa model berbasis LSTM sering kali mengalami overfitting jika tidak dioptimalkan dengan baik. Hasil ini sejalan dengan temuan penelitian ini, di mana LSTM memiliki tingkat kesalahan yang lebih tinggi dibandingkan ARIMA-GARCH, terutama pada periode pandemi COVID-19 yang memiliki volatilitas tinggi. Dalam konteks ini, ARIMA-GARCH tetap lebih stabil karena dapat menangkap pola volatilitas historis dengan lebih baik.

Selain itu, penelitian oleh Vo & Ślepaczuk (2022) menunjukkan bahwa penggunaan hybrid ARIMA-SGARCH dalam strategi investasi pada indeks S&P 500 mampu meningkatkan akurasi prediksi dibandingkan model individual. Hasil penelitian ini memperkuat argumen bahwa integrasi model statistik dan deep learning dapat menjadi strategi yang lebih unggul dalam meramalkan pergerakan pasar. Meskipun dalam penelitian ini LSTM menunjukkan performa yang kurang optimal, pendekatan hybrid tetap memiliki potensi untuk meningkatkan prediksi di masa depan.

Dengan demikian, berdasarkan hasil penelitian ini dan perbandingan dengan studi sebelumnya, disarankan untuk mengembangkan model hybrid yang mengintegrasikan keunggulan ARIMA-GARCH dalam menangkap volatilitas dengan fleksibilitas LSTM dalam mengenali pola non-linear. Kombinasi ini diharapkan dapat menghasilkan model prediksi yang lebih akurat dan adaptif terhadap berbagai kondisi pasar keuangan.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dalam penelitian ini, dilakukan analisis perbandingan antara model ARIMA-GARCH dan LSTM dalam memprediksi pergerakan IHSG pada tiga periode berbeda, yaitu sebelum, selama, dan setelah pandemi COVID-19. Perubahan kondisi pasar akibat pandemi memberikan tantangan tersendiri bagi model peramalan, terutama dalam menangkap pola volatilitas yang bervariasi di setiap periode. Oleh karena itu, evaluasi performa kedua model dilakukan dengan menggunakan tiga metrik utama, yaitu MSE, RMSE, dan MAPE, guna menentukan model yang paling akurat dalam berbagai kondisi pasar. Hasil penelitian menunjukkan bahwa ARIMA-GARCH secara

umum memiliki performa yang lebih baik dibandingkan LSTM dalam hampir semua periode, meskipun terdapat perbedaan efektivitas masing-masing model dalam menangkap pola data yang berubah akibat pandemi.

Pertama, pada periode sebelum COVID-19, ketika kondisi pasar relatif stabil dan tidak mengalami guncangan besar akibat faktor eksternal, model ARIMA-GARCH terbukti lebih unggul dibandingkan LSTM dalam meramalkan pergerakan IHSG. Hal ini ditunjukkan oleh nilai MSE (4.887), RMSE (2.211), dan MAPE (0.035%), yang jauh lebih rendah dibandingkan model LSTM yang memiliki MSE (14.748), RMSE (61.149), dan MAPE (0.745%). Performa ARIMA-GARCH yang lebih baik dalam kondisi pasar stabil disebabkan oleh kemampuannya dalam menangkap pola time series yang linier dan tren historis yang tidak mengalami fluktuasi ekstrem. Sebaliknya, LSTM, yang berbasis deep learning, kurang mampu memberikan prediksi yang akurat karena pola data yang relatif sederhana sebelum pandemi tidak memerlukan kompleksitas pemodelan non-linear yang menjadi keunggulan utama dari LSTM.

Kedua, selama pandemi COVID-19, pasar mengalami lonjakan volatilitas yang signifikan akibat ketidakpastian ekonomi dan kebijakan pemerintah. Dalam kondisi ini, ARIMA-GARCH kembali menunjukkan performa yang lebih baik dibandingkan LSTM, dengan nilai MSE (0.735), RMSE (0.858), dan MAPE (0.012%), sedangkan model LSTM memiliki nilai error yang jauh lebih tinggi, yaitu MSE (5105.110), RMSE (94.167), dan MAPE (1.157%). Peningkatan volatilitas selama pandemi menyebabkan pola pergerakan IHSG menjadi lebih dinamis dan tidak menentu, namun ARIMA-GARCH masih mampu menangkap perubahan volatilitas ini dengan cukup baik melalui komponen GARCH yang mengakomodasi heteroskedastisitas dalam data keuangan. Sebaliknya, LSTM mengalami kesulitan dalam mengidentifikasi pola pasar yang sangat tidak stabil, yang mengakibatkan prediksi yang kurang akurat.

Ketiga, setelah pandemi COVID-19, pasar memasuki fase transisi menuju pemulihan ekonomi, dengan volatilitas yang masih tinggi tetapi menunjukkan tren stabilisasi. Dalam kondisi ini, performa ARIMA-GARCH mengalami penurunan, dengan nilai MSE (60.159), RMSE (7.756), dan MAPE (0.107%), yang mengindikasikan bahwa model ini mulai kesulitan dalam menangkap pola pergerakan IHSG yang lebih kompleks. Sebaliknya, LSTM menunjukkan peningkatan performa dibandingkan periode sebelumnya, meskipun masih memiliki tingkat kesalahan yang cukup tinggi, dengan MSE (126.956), RMSE (67.303), dan MAPE (0.711%). Hasil

ini menunjukkan bahwa ketika pasar mulai memasuki kondisi transisi, model berbasis deep learning seperti LSTM mulai dapat menangkap pola non-linear dengan lebih baik dibandingkan ARIMA-GARCH, yang lebih efektif untuk kondisi pasar yang lebih stabil atau memiliki pola volatilitas yang dapat dimodelkan dengan pendekatan statistik.

Keempat, perbandingan performa kedua model pada tiga periode ini mengindikasikan adanya perubahan karakteristik pola data IHSG sebelum, selama, dan setelah COVID-19, yang berdampak pada akurasi masing-masing model forecasting. Sebelum pandemi, pola pergerakan IHSG cenderung lebih stabil, sehingga model ARIMA-GARCH lebih efektif dalam menangkap tren historis yang linier. Selama pandemi, peningkatan volatilitas menyebabkan kesulitan bagi model LSTM dalam mempelajari pola data yang tidak stabil, sementara ARIMA-GARCH masih dapat mengakomodasi volatilitas melalui komponen GARCH. Setelah pandemi, pasar mengalami pemulihan dengan karakteristik yang lebih kompleks dan volatilitas yang tidak sepenuhnya dapat dijelaskan oleh model statistik, sehingga performa LSTM mulai membaik. Berdasarkan temuan ini, dapat disimpulkan bahwa ARIMA-GARCH lebih unggul dalam kondisi pasar yang stabil atau sangat volatil tetapi masih dapat dimodelkan secara statistik, sementara LSTM memiliki potensi lebih baik dalam menangkap pola non-linear yang lebih kompleks ketika pasar memasuki fase transisi. Oleh karena itu, pendekatan hybrid yang menggabungkan keunggulan kedua model seperti LSTM-GARCH dapat menjadi strategi yang lebih efektif dalam meningkatkan akurasi prediksi IHSG dalam berbagai kondisi pasar.

5.2 Saran

Sebagai tindak lanjut dari temuan dalam penelitian ini, terdapat beberapa saran yang dapat dipertimbangkan untuk meningkatkan akurasi model peramalan IHSG serta memperkaya kajian di masa mendatang. Saran-saran ini mencakup aspek metodologi, pengembangan model, serta aplikasi dalam pengambilan keputusan investasi dan kebijakan keuangan.

1. Penggunaan Model Hybrid

Hasil penelitian menunjukkan bahwa ARIMA-GARCH lebih unggul dalam kondisi pasar stabil dan sangat volatil, sedangkan LSTM mulai menunjukkan keunggulannya dalam fase pemulihan pasca-pandemi. Oleh karena itu, penelitian di masa depan disarankan untuk

mengembangkan model hybrid yang menggabungkan LSTM dengan GARCH guna memanfaatkan keunggulan masing-masing model dalam menangkap pola data yang berbeda. LSTM yang lebih unggul dalam menangkap pola non linier dan GARCH yang lebih unggul dalam menangkap pola volatilitas sehingga model hybrid ini diharapkan dapat melampaui model ARIMA-GARCH yang paling unggul dalam penelitian ini.

2. Penggunaan Data Eksternal

Model yang digunakan dalam penelitian ini hanya mempertimbangkan data historis IHSG. Untuk meningkatkan akurasi prediksi, disarankan agar model di masa depan memasukkan variabel makroekonomi seperti suku bunga, inflasi, nilai tukar, harga komoditas, dan indeks pasar global guna menangkap dampak faktor eksternal terhadap IHSG.

3. Peningkatan Teknik Preprocessing Data

Peningkatan akurasi model dapat dilakukan dengan menggunakan teknik feature engineering yang lebih canggih, seperti normalisasi data, rolling window analysis, atau pemilihan variabel yang lebih optimal untuk LSTM, agar model dapat menangkap pola yang lebih kompleks dalam data IHSG.

4. Eksplorasi Model Machine Learning Lainnya

Selain LSTM, terdapat metode machine learning lain yang berpotensi memberikan hasil lebih baik dalam menangkap pola non-linear, seperti GRU (Gated Recurrent Unit), Transformer-based models, atau XGBoost. Eksplorasi model-model ini dapat menjadi alternatif untuk meningkatkan akurasi prediksi IHSG.

5. Pengujian pada Periode yang Lebih Lama dan Beragam

Untuk memperoleh generalisasi hasil yang lebih baik, penelitian selanjutnya dapat memperluas cakupan data historis IHSG dengan periode waktu yang lebih panjang serta mempertimbangkan berbagai periode krisis ekonomi lainnya guna menguji ketahanan model terhadap berbagai kondisi pasar.

Dengan menerapkan saran-saran tersebut, diharapkan penelitian mendatang dapat mengembangkan model forecasting yang lebih akurat dan aplikatif dalam berbagai kondisi pasar,

sehingga dapat memberikan manfaat yang lebih luas bagi akademisi, praktisi keuangan, serta pembuat kebijakan.

DAFTAR PUSTAKA

- Abidha, H. L., & Ahdika, A. (2024). Unlocking market insight: Forecasting PT Bank Central Asia Tbk stock prices with ARIMA-GARCH analysis. *Desimal: Jurnal Matematika*, 7(2), 217-228.
- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11).
- Abreu, R. J., Souza, R. M., & Oliveira, J. G. (2019). Applying singular spectrum analysis and ARIMA-GARCH for forecasting EUR/USD exchange rate. *RAM. Revista de Administração Mackenzie*, 20, eRAMF190146.
- Aktivani, S. (2020). Uji Stasioneritas Data Inflasi Kota Padang Periode 2014-2019. *Statistika*, 20(2), 83-90.
- Aldi, M. W. P., Jondri, J., & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *eProceedings of Engineering*, 5(2).
- Alimohamadi, Y., Taghdir, M., & Sepandi, M. (2020). Estimate of the basic reproduction number for COVID-19: a systematic review and meta-analysis. *Journal of Preventive Medicine and Public Health*, 53(3), 151.
- Alsharif, M. H., Younes, M. K., & Kim, J. (2019). Time series ARIMA model for prediction of daily and monthly average global solar radiation: The case study of Seoul, South Korea. *Symmetry*, 11(2), 240.
- Altig, D., Baker, S., Barrero, J. M., Bloom, N., Bunn, P., Chen, S., ... & Thwaites, G. (2020). Economic uncertainty before and during the COVID-19 pandemic. *Journal of public economics*, 191, 104274.
- Anggraini, D. (2021). Dampak COVID-19 terhadap perubahan harga saham. *Jurnal Bisnis, Ekonomi, Manajemen, dan Kewirausahaan*, 1-13.

- Annisa, R. D., & Riofita, H. (2024). Tantangan Implementasi Kebijakan Moneter dan Fiskal di Negara-Negara Berpenghasilan Rendah: Tinjauan Literatur. *Journal of Accounting Law Communication and Technology*, 1(2), 475-480.
- Ardian, R., Hendayana, Y., & Sulistyowati, A. (2024). Pengaruh inflasi, suku bunga dan nilai tukar terhadap indeks harga saham gabungan. *Kinerja: Jurnal Ekonomi dan Bisnis*, 6(01), 180-193.
- Arianto, B. (2020). Dampak pandemi COVID-19 terhadap perekonomian dunia. *Jurnal Ekonomi Perjuangan*, 2(2), 106-126.
- Badan Pemeriksa Keuangan Republik Indonesia (BPK RI). (2020). *Keputusan Presiden Nomor 11 Tahun 2020* Retrieved from <https://peraturan.bpk.go.id/Details/135058/keppres-no-11-tahun-2020>
- Badan Pemeriksa Keuangan Republik Indonesia (BPK RI). (2023). *Keputusan Presiden Nomor 17 Tahun 2023*. Retrieved from <https://peraturan.bpk.go.id/Details/253985/keppres-no-17-tahun-2023>
- Bai, C., Duan, Y., Fan, X., & Tang, S. (2023). Financial market sentiment and stock return during the COVID-19 pandemic. *Finance Research Letters*, 54, 103709.
- Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, 9, 100320.
- Bichri, H., Chergui, A., & Hain, M. (2024). Investigating the Impact of Train/Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets. *International Journal of Advanced Computer Science & Applications*, 15(2).
- Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics*, 542-547.
- Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 1429-1441.

- Brodeur, A., Gray, D., Islam, A., & Bhuiyan, S. (2021). A literature review of the economics of COVID-19. *Journal of economic surveys*, 35(4), 1007-1044.
- Budiharto, W. (2021). Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM). *Journal of big data*, 8, 1-9.
- Caiado, J., & Lúcio, F. (2023). Stock market forecasting accuracy of asymmetric GARCH models during the COVID-19 pandemic. *The North American Journal of Economics and Finance*, 68, 101971.
- Časta, M. (2023). Inflation, interest rates and the predictability of stock returns. *Finance Research Letters*, 58, 104380.
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, 7, e623.
- Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10), 3765.
- Cullen, W., Gulati, G., & Kelly, B. D. (2020). Mental health in the COVID-19 pandemic. *QJM: An International Journal of Medicine*, 113(5), 311-312.
- Dalal, M., Li, A. C., & Taori, R. (2019). Autoregressive models: What are they good for?. arXiv preprint arXiv:1910.07737.
- Dana, A. N. (2016). Modelling and estimation of volatility using ARCH/GARCH models in Jordan's stock market. *Asian Journal of Finance & Accounting*, 8(1), 152-167.
- Daniel, S. J. (2020). Education and the COVID-19 pandemic. *Prospects*, 49(1), 91-96.
- Demir, V., Zontul, M., & Yelmen, I. (2020). Drug sales prediction with ACF and PACF supported ARIMA method. In *2020 5th International Conference on Computer Science and Engineering (UBMK)* (pp. 243-247). IEEE.

- Desfiandi, A., Desfiandi, A., & Ali, H. (2017). Composite Stock Price Index (IHSG) Macro Factor in Investment in Stock (Equity Funds). *International Journal of Economics and Financial Issues*, 7(3), 534-536.
- Devi, S. S., & Wibowo, D. (2021). Pengaruh inflasi dan nilai tukar/kurs Terhadap Indeks Harga Saham Gabungan (IHSG) yang terdaftar di Bursa Efek Indonesia (BEI) pada masa pandemi covid-19 bulan Januari-Desember tahun 2020. *Jurnal Inovatif Mahasiswa Manajemen*, 1(2), 139-149.
- Deviana, S., Nusyirwan, N., Azis, D., & Ferdias, P. (2021). Analisis model autoregressive integrated moving average data deret waktu dengan metode momen sebagai estimasi parameter. *Jurnal Siger Matematika*, 57-67.
- Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018, October). Unsupervised learning based on artificial neural network: A review. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (pp. 322-327). IEEE.
- Eylasov, N., & Çiçek, M. (2024). Kripto para fiyatlarının tahmini: ARIMA-GARCH ve LSTM yöntemlerinin karşılaştırılması. *Finans Ekonomi ve Sosyal Araştırmalar Dergisi*, 9(1), 48-62.
- Fadhilah, D. N., Parmikanti, K., & Ruchjana, B. N. (2024). Peramalan Return Saham Subsektor Perbankan Menggunakan Model ARIMA-GARCH. *Jurnal Fourier*, 13(1), 1-19.
- Fadila, A. N., & Nuswandari, C. (2022). Apa saja faktor-faktor yang mempengaruhi harga saham. *E-Bisnis: Jurnal Ilmiah Ekonomi Dan Bisnis*, 15(2), 283-293.
- Fadilah, M. H., Novendra, H., Kurnianto, B. A., & Kasih, T. P. (2024). ANALISA DERET WAKTU DENGAN MENGGUNAKAN METODE ARIMA DAN VAR PADA APLIKASI ORANGE. *Jurnal Review Pendidikan dan Pengajaran (JRPP)*, 7(1), 330-333.
- Fang, H., Chung, C. P., Lu, Y. C., Lee, Y. H., & Wang, W. H. (2021). The impacts of investors' sentiments on stock returns using fintech approaches. *International Review of Financial Analysis*, 77, 101858.
- Fassas, A. P., & Siriopoulos, C. (2021). Implied volatility indices—A review. *The Quarterly Review of Economics and Finance*, 79, 303-329.

Fatimah, F., Rosadi, D., Hakim, R. F., & Alcantud, J. C. R. (2018). N-soft sets and their decision making algorithms. *Soft Computing*, 22, 3829-3842.

Fatmawati, A. K., Badriah, L. S., & Gunawan, D. S. (2024). Perkembangan Indeks Harga Saham Gabungan (IHSG) Sebelum dan Sesudah Krisis Ekonomi Global Tahun 2008. *Journal of Economic, Management, Accounting and Technology*, 7(1), 31-39.

Fatmawati, A., & Hendrayani, S. Y. (2020). The Risk Perception of Covid-19 in Indonesia. *Jurnal Vokasi Keperawatan (JVK)*, 3(2), 103-108.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2), 654-669.

fitriani, ine, Prizka Rismawati Arum, & Amri, S. (2024). Pemodelan Seasonal Autoregressive Integrated Moving Average (SARIMA) untuk Meramalkan Volume Angkutan Barang Kereta Api di Pulau Jawa Tahun 2021: Memberikan prediksi volume angkutan barang kereta api di pulau jawa untuk masa mendatang. *Journal Of Data Insights*, 2(1), 26–35.

Foroni, C., Marcellino, M., & Stevanovic, D. (2022). Forecasting the Covid-19 recession and recovery: Lessons from the financial crisis. *International Journal of Forecasting*, 38(2), 596-612.

Foroni, Claudia, Massimiliano Marcellino, and Dalibor Stevanovic. "Forecasting the Covid-19 recession and recovery: Lessons from the financial crisis." *International Journal of Forecasting* 38.2 (2022): 596-612.

Francq, C., & Zakoian, J. M. (2019). GARCH models: structure, statistical inference and financial applications. John Wiley & Sons.

Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*, 2021(1), 4055281.

Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37-45.

Gujarati, D. N., & Porter, D. C. (2009). *Basic Econometrics* (5th ed.). McGraw-Hill.

- Halwa, A. K. (2023). *Analisis Deret Waktu Univariat*. UNIPMA Press
- Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.
- Hargreaves, J. R., & Logie, C. H. (2020). Lifting lockdown policies: A critical moment for COVID-19 stigma. *Global public health*, 15(12), 1917-1923.
- Hidayat, R., & Mustawinar, B. H. (2022). Peramalan Jumlah Wisatawan Asing Dengan Model ARIMA. *Infinity: Jurnal Matematika dan Aplikasinya*, 2(2), 104-115.
- Hochreiter, S. (1997). Long Short-term Memory. *Neural Computation MIT-Press*.
- Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, 2022, 1-10.
- Hon, T. Y., Moslehpour, M., & Woo, K. Y. (2021). Review on behavioral finance with empirical evidence. *Advances in Decision Sciences*, 25(4), 1-30.
- Huang, S., Liu, Y., Zhang, F., Li, Y., Li, J., & Zhang, C. (2024). Crosswavenet: A dual-channel network with deep cross-decomposition for long-term time series forecasting. *Expert Systems with Applications*, 238, 121642.
- Huda, E. N., & Suseno, P. (2023). Comparative Analysis of Sharia Stock Screening Methods in Indonesia, Malaysia, and the United States. *Maliki Islamic Economics Journal*, 3(1), 1-18.
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.
- Ivashka, N. N., Arsana, I. M. M., & Dewi, N. N. S. R. T. (2024). Kinerja Keuangan Pada Perusahaan Sektor Kesehatan Yang Terdaftar Di Bursa Efek Indonesia Sebelum Dan Semasa Pandemi Covid-19. *Journal Research of Accounting*, 5(2), 310-321.
- Juanda, B., & Junaidi, J. (2021). *Ekonometrika Deret Waktu: Teori dan Aplikasi*. PT Penerbit IPB Press.

Julian, R., & Pribadi, M. R. (2021). Peramalan harga saham pertambangan pada bursa efek indonesia (BEI) menggunakan long short-term memory (LSTM). *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(3), 1570-1580.

Juliati, Y. S. (2015). Peranan pasar modal dalam perekonomian negara. *HUMAN FALAH: Jurnal Ekonomi dan Bisnis Islam*, 2(1), 95-112.

Junaedi, D., & Salistia, F. (2020). Dampak pandemi Covid-19 terhadap pasar modal di Indonesia: studi kasus indeks saham komposit (Ihsg). *Al-Kharaj: Jurnal Ekonomi, Keuangan & Bisnis Syariah*, 2(2), 109-131.

Jurado, K., Ludvigson, S. C., & Ng, S. (2015). Measuring uncertainty. *American Economic Review*, 105(3), 1177-1216.

Kaur, M., & Mohta, A. (2019). A review of deep learning with recurrent neural network. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 460-465). IEEE.

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, 1-24.

Kompas. (2020). *Pengumuman Kasus Pertama COVID-19 di Indonesia*. Retrieved from <https://nasional.kompas.com/read/2020/03/02/12002701/ini-pengumuman-lengkap-jokowi-soal-2-wni-positif-corona?page=all>

Kumar, A., Singh, R., Kaur, J., Pandey, S., Sharma, V., Thakur, L., Sati, S., Mani, S., Asthana, S., Sharma, T. K., Chaudhuri, S., Bhattacharyya, S., & Kumar, N. (2021). *Wuhan to World: The COVID-19 Pandemic*. *Frontiers in Cellular and Infection Microbiology*, 11, 596201.

Kusumawardana, V. (2016). Komparasi Strategi Investasi Aktif Dan Pasif Untuk Optimalkan Return Saham Yang Terdaftar Di Bursa Efek Indonesia. *Journal of Innovation in Business and Economics*, 7(1), 41-54.

- Lakshminarayanan, S. K., & McCrae, J. P. (2019). A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction. In *AICS* (pp. 446-457).
- Lehkonen, H. (2015). Stock market integration and the global financial crisis. *Review of Finance*, 19(5), 2039-2094.
- Li, Z., Yu, H., Xu, J., Liu, J., & Mo, Y. (2023). Stock market analysis and prediction using LSTM: A case study on technology stocks. *Innovations in Applied Engineering and Technology*, 1-6.
- Liu, Y., Wu, H., Wang, J., & Long, M. (2022). Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35, 9881-9893.
- Lu, J., Peng, J., Chen, J., & Sugeng, K. A. (2020). Prediction method of autoregressive moving average models for uncertain time series. *International Journal of General Systems*, 49(5), 546-572.
- Mahfuzh, M. F., & Yuliantari, R. V. (2022). Analisis penerapan artificial neural network algoritma propagasi balik untuk meramalkan harga saham pada bursa efek Indonesia. *Journal of Applied Electrical Engineering*, 6(1), 1-3.
- Maity, S., Sahu, T. N., & Sen, N. (2021). Panoramic view of digital education in COVID-19: A new explored avenue. *Review of Education*, 9(2), 405-423.
- Manurung, A. H., Budiharto, W., & Prabowo, H. (2018). Algorithm and modeling of stock prices forecasting based on long short-term memory (LSTM). *ICIC Express Letters*, 12(12), 1277-1283.
- Maulana, H. A. (2018). Pemodelan deret waktu dan peramalan curah hujan pada dua belas stasiun di Bogor. *Jurnal Matematika, Statistika dan Komputasi*, 15(1), 50-63.
- Mehtarizadeh, H., Mansouri, N., Mohammad Hasani Zade, B., & Hosseini, M. M. (2025). Stock price prediction with SCA-LSTM network and Statistical model ARIMA-GARCH. *The Journal of Supercomputing*, 81(2), 366.

- Merabet, F., Zeghdoudi, H., Yahia, R. H., & Saba, I. (2021). Modelling of oil price volatility using ARIMA-GARCH models. *Adv. Mathematics*, 10, 2361-2380.
- Mie, M. (2014). Analisis Pengaruh Indeks Harga Saham Gabungan Asing Terhadap Indeks Harga Saham Gabungan Indonesia.
- Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia computer science*, 170, 1168-1173.
- Mondal, K. C., Ghosh, M., Fajriyah, R., & Roy, A. (2024). Introducing suffix forest for mining tri-clusters from time-series data. *Innovations in Systems and Software Engineering*, 20(4), 765-787.
- Mubarok, A. H. (2017). Mekanisme Pasar Sekunder Dan Fungsi Lembaga Nya Dalam Pasar Modal Indonesia. *Business Law Review*, 48.
- Muklis, F. (2016). Perkembangan dan tantangan pasar modal Indonesia. *Al-Masraf: Jurnal Lembaga Keuangan dan Perbankan*, 1(1), 65-76.
- Muliadi, H., & Fahmi, I. (2016). Pengaruh Dividen Per Share, Return on Equity Dan Net Profit Margin Terhadap Harga Saham Perusahaan Manufaktur Di Bursa Efek Indonesia. *Jurnal Ilmiah Mahasiswa Ekonomi Manajemen*, 1(2), 71-80.
- Mustapa, F. H., & Ismail, M. T. (2019). Modelling and forecasting S&P 500 stock prices using hybrid Arima-Garch Model. In *Journal of Physics: Conference Series* (Vol. 1366, No. 1, p. 012130). IOP Publishing.
- Nabi, K. N. (2020). Forecasting COVID-19 pandemic: A data-driven analysis. *Chaos, Solitons & Fractals*, 139, 110046.
- Nansi, M. R., Natalia, M. E., & Rohman, N. R. (2024). Pemulihan ekonomi biru Indonesia pasca pandemi. *Innovative: Journal of Social Science Research*, 4(2), 6992-7001.

Nasirudin, F., Pindianti, M., Said, D. I. S., & Widodo, E. (2022). Peramalan jumlah produksi kopi di Jawa Timur pada tahun 2020-2021 menggunakan metode seasonal autoregressive integrated moving average (sarima). *AGRIUM: Jurnal Ilmu Pertanian*, 25(1), 34-43.

Noerkaisar, N. (2021). Efektivitas penyaluran bantuan sosial pemerintah untuk mengatasi dampak Covid-19 di Indonesia. *Jurnal Manajemen Perbendaharaan*, 2(1), 83-104.

Oktarina, D. (2015). Pengaruh beberapa indeks saham global dan indikator makroekonomi terhadap pergerakan IHSG. *Journal of Business & Banking*, 5(2), 163-182.

Olah, C. Understanding LSTM Networks—Colah's Blog. 2016. Available online: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (last accessed on 12 February 2025).

Osborne, J. (2010). Improving your data transformations: Applying the Box-Cox transformation. *Practical Assessment, Research, and Evaluation*, 15(1).

Pagnottoni, P., Spelta, A., Flori, A., & Pammolli, F. (2022). Climate change and financial stability: Natural disaster impacts on global stock markets. *Physica A: Statistical Mechanics and Its Applications*, 599, 127514.

Pahlavani, M., & Roshan, R. (2015). The comparison among ARIMA and hybrid ARIMA-GARCH models in forecasting the exchange rate of Iran. *International Journal of Business and Development Studies*, 7(1), 31-50.

Pambudi, A. S. (2023). Respons kebijakan pemerintah daerah dalam pemanfaatan anggaran DID pada periode awal pandemi COVID-19. *Jurnal Kebijakan Pembangunan Daerah*, 7(1), 48-67.

Pamungkas, M. B., & Wibowo, A. (2019). Aplikasi metode arima box-jenkins untuk meramalkan kasus DBD di Provinsi Jawa Timur. *The Indonesian Journal of Public Health*, 13(2), 183.

Pascanu, R. (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Patro, S. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Perc, M., Gorišek Miksić, N., Slavinec, M., & Stožer, A. (2020). Forecasting covid-19. *Frontiers in physics*, 8, 127.

PETRICĂ, A. C., & Stancu, S. (2017). Empirical Results of Modeling EUR/RON Exchange Rate using ARCH, GARCH, EGARCH, TARCH and PARCH models. *Romanian Statistical Review*, (1).

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., Ellison, J., Ziel, F. (2022). *Forecasting: Theory and Practice. International Journal of Forecasting*, 38(3), 705-871.

Presiden Republik Indonesia (Presiden RI). (2022). *Pemerintah Resmi Cabut Kebijakan PPKM*. Retrieved from <https://www.presidentri.go.id/siaran-pers/pemerintah-resmi-cabut-kebijakan-ppkm-mulai-hari-ini/>

Putri, F. T. A., Zukhronah, E., & Pratiwi, H. (2021). Model ARIMA-GARCH Pada Peramalan Harga Saham PT. Jasa Marga (Persero). *Business Innovation and Entrepreneurship Journal*, 3(3), 164-170.

Putri, G. A. M. A., Hendayanti, N. P. N., & Nurhidayati, M. (2017). Pemodelan Data Deret Waktu Dengan Autoregressive Integrated Moving Average Dan Logistic Smoothing Transition Autoregressive. *Jurnal Varian*, 1(1), 54-63.

Robiyanto, R. (2018). Performance evaluation of stock price indexes in the Indonesia Stock Exchange. *International Research Journal of Business Studies*, 10(3), 173-182.

Roca, F. (2021). What is New in Value Investing? A Systematic Literature Review. *Journal of New Finance*, 2(2), 2.

Rorizki, F., Sakinah, S., Dalimunthe, A., & Silalahi, P. R. (2022). Perkembangan dan tantangan pasar modal Indonesia. *Economic Reviews Journal*, 1(2), 147-157.

- Rosyidah, H., Rahmawati, R., & Prahutama, A. (2018). Pemodelan Vector Autoregressive X (VARX) untuk Meramalkan Jumlah Uang Beredar di Indonesia. *Jurnal Gaussian*, 6(3), 333-343.
- Ruhat, D., & Suwanda, C. (2019). Peramalan data deret waktu berpola musiman menggunakan metode regresi spektral (studi kasus: debit sungai citarum-nanjung). *Teorema: Teori dan Riset Matematika*, 4(1), 1-12.
- Salisa, N. R. (2021). Faktor yang Mempengaruhi Minat Investasi di Pasar Modal: Pendekatan Theory of Planned Behaviour (TPB). *Jurnal Akuntansi Indonesia*, 9(2), 182.
- Sambuari, I. B., Saerang, I. S., & Maramis, J. B. (2020). Reaksi pasar modal terhadap peristiwa virus corona (COVID-19) pada perusahaan makanan dan minuman yang terdaftar di bursa efek Indonesia. *JMBI UNSRAT (Jurnal Ilmiah Manajemen Bisnis dan Inovasi Universitas Sam Ratulangi)*, 7(2).
- San Marino, W., & Rohanah, A. S. (2021). Pengaruh COVID-19 terhadap pasar modal di Indonesia. *BanKu: Jurnal Perbankan dan Keuangan*, 2(2), 98-104.
- Sanyaolu, A., Okorie, C., Marinkovic, A., Patidar, R., Younis, K., Desai, P., Hosein, Z., Padda, I., Mangat, J., & Altaf, M. (2020). Comorbidity and its impact on patients with COVID-19. *SN Comprehensive Clinical Medicine*, 2, 1069-1076.
- Saputri, T. A., Febritama, D., & RA, A. M. (2025). ANALISIS ARIMA UNTUK MEMPROYEKSI PERTUMBUHAN PENDUDUK DI KOTA METRO. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, 8(1), 176-183.
- Sharma, P., Leung, T. Y., Kingshott, R. P., Davcik, N. S., & Cardinali, S. (2020). Managing uncertainty during a global pandemic: An international business perspective. *Journal of business research*, 116, 188-192.
- Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306.
- Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306.

- Shumway, R. H., Stoffer, D. S., Shumway, R. H., & Stoffer, D. S. (2017). ARIMA models. *Time series analysis and its applications: with R examples*, 75-163.
- Sinsomboonthong, S. (2022). Performance Comparison of New Adjusted Min-Max with Decimal Scaling and Statistical Column Normalization Methods for Artificial Neural Network Classification. *International Journal of Mathematics and Mathematical Sciences*, 2022(1), 3584406.
- Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.
- Suhardini, E. D. (2015). Peran dan Fungsi Pasar Modal dalam Pembangunan Perekonomian Indonesia. *Wacana Paramarta: Jurnal Ilmu Hukum*, 14(1).
- Ting, D. S. W., Carin, L., Dzau, V., & Wong, T. Y. (2020). Digital technology and COVID-19. *Nature medicine*, 26(4), 459-461.
- Țițan, A. G. (2015). The efficient market hypothesis: Review of specialized literature and empirical research. *Procedia Economics and Finance*, 32, 442-449.
- Van Greunen, J., Heymans, A., Van Heerden, C., & Van Vuuren, G. (2014). The prominence of stationarity in time series forecasting. *Studies in Economics and Econometrics*, 38(1), 1-16.
- Varshney, R. P., & Sharma, D. K. (2024). Optimizing Time-Series forecasting using stacked deep learning framework with enhanced adaptive moment estimation and error correction. *Expert Systems with Applications*, 249, 123487.
- Vivas, E., Allende-Cid, H., & Salas, R. (2020). A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. *Entropy*, 22(12), 1412.
- Wahyudi, H., & Ramani, F. (2022). Pengaruh Jangka Pendek dan Jangka Panjang Saham Global terhadap Indeks Harga Saham Gabungan (IHSG) Periode 2015: M01-2020: M12. *Reviu Akuntansi, Manajemen, dan Bisnis*, 2(1), 15-25.
- Wardani, W. W. K. (2021). *Prediksi harga saham syariah menggunakan metode recurrent neural network-long short term memory* (Skripsi, Universitas Islam Negeri Sunan Ampel Surabaya).

- Wauchope, H. S., Amano, T., Geldmann, J., Johnston, A., Simmons, B. I., Sutherland, W. J., & Jones, J. P. (2021). Evaluating impact using time-series data. *Trends in Ecology & Evolution*, 36(3), 196-205.
- Wei, C. H., Aleksandrov, B., Faymonville, M., & Jentsch, C. (2023). Partial autocorrelation diagnostics for count time series. *Entropy*, 25(1), 105.
- Wibawa, A. P., Izdiyar, Z. N., Utama, A. B. P., & Hernandez, L. (2021). Min-Max Backpropagation Neural Network to Forecast e-Journal Visitors. In *2021 International conference on artificial intelligence in information and communication (ICAIIC)* (pp. 052-058). IEEE.
- Widagda, I. A. (2016). Pengenalan Pola Dua Dimensi Sederhana dengan Jaringan Syaraf Tiruan Hebb. Karya Tulis Ilmiah FMIPA Universitas Udayana, 4.
- World Health Organization. (2018). Global status report on alcohol and health 2018. World Health Organization.
- Wulan, R., Nurpadilah, N., & Pebrian, R. (2023). Pengaruh Inflasi, Harga Minyak Dunia, Dan Suku Bunga (Bi Rate) Terhadap Indeks Harga Saham Gabungan (Ihsg)(Data Per Bulan Periode 2011-2020). *Jurnal Pijar*, 1(2), 130-143.
- Xiang, Y. (2022). Using ARIMA-GARCH Model to Analyze Fluctuation Law of International Oil Price. *Mathematical Problems in Engineering*, 2022(1), 3936414.
- Xing, Y., Yan, C., & Xie, C. C. (2024). Predicting nvidia's next-day stock price: A comparative analysis of lstm, mlp, arima, and arima-garch models. *arXiv preprint arXiv:2405.08284*.
- Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, 167, 2091-2100.
- Yakubu, U. A., & Saputra, M. P. A. (2022). Time series model analysis using autocorrelation function (ACF) and partial autocorrelation function (PACF) for E-wallet transactions during a pandemic. *International Journal of Global Operations Research*, 3(3), 80-85.

Yaseen, A., Suseno, P., & Shah, S. G. H. (2024). IMPACT OF MACROECONOMIC VARIABLES AND MORTALITY RATES UNDER 5 ON CO. *Journal of Developing Economies* Vol, 9(2), 243-258.

Yi, D., Ahn, J., & Ji, S. (2020). An effective optimization method for machine learning based on ADAM. *Applied Sciences*, 10(3), 1073.

Yong, S. J. (2021). Long COVID or post-COVID-19 syndrome: putative pathophysiology, risk factors, and treatments. *Infectious diseases*, 53(10), 737-754.

Zhang, M. (2018). Time series: Autoregressive models ar, ma, arma, arima. University of Pittsburgh.

Lampiran 1. Data Penelitian (Sebelum, Masa dan Seteleh)

<https://drive.google.com/drive/folders/1hNtkv-jZFwdArPJnGUW-eisVR0iyiSRT>

Lampiran 2. Sintaks R Model ARIMA GARCH Sebelum Covid-19

```
> library(ggplot2)
> library(tsibble)
> library(tseries)
> library(MASS)
> library(dplyr)
> library(TTR)
> library(forecast)
> library(lmtest)
> library(HoRM)
> library(dLagM)
> library(dynlm)
> library(MLmetrics)
> library(car)
> library(TSA)
> library(aTSA)
> library(rugarch)
> library(imputeTS)
> library(tidyr)
> # Membaca data
> setwd("C:/Users/asus/Dropbox/Penelitian UII/Analisis Data")
> datasaham <- read.csv("Sebelum Covid.csv")
> # Menghapus kolom yang tidak diperlukan (sesuaikan dengan struktur data Anda)
> datasaham <- datasaham[, -c(2,3,4,6,7)]
> # Konversi kolom Date ke format waktu
> datasaham$Date <- as.Date(datasaham$Date, format="%m/%d/%Y")
> datasaham$Date <- as.POSIXct(datasaham$Date)
> # Konversi data menjadi time series
> ts.datasaham <- ts(datasaham$Close)
> # Menampilkan subset data dengan tabel yang lebih rapi
> kableExtra::kable(head(datasaham), caption = 'Subset Data Harian IHSG')
```

Table: Subset Data Harian IHSG

Date	Close
2018-01-02	6339.238
2018-01-03	6251.479
2018-01-04	6292.321
2018-01-05	6353.738
2018-01-08	6385.404
2018-01-09	6373.144

```
> # Melihat struktur dan ringkasan data
> str(datasaham)
'data.frame': 527 obs. of 2 variables:
 $ Date : POSIXct, format: "2018-01-02" ...
 $ Close: num 6339 6251 6292 6354 6385 ...
> summary(datasaham)
      Date      Close
Min.   :2018-01-02 00:00:00.00 Min.   :5453
1st Qu.:2018-07-23 12:00:00.00 1st Qu.:5980
Median :2019-02-04 00:00:00.00 Median :6211
Mean    :2019-02-02 15:34:29.82 Mean    :6181
3rd Qu.:2019-08-22 12:00:00.00 3rd Qu.:6373
```

```

Max.      :2020-02-28 00:00:00.00    Max.      :6689
> # Konversi kolom Date ke format Date
> datasaham$Date <- as.Date(datasaham$Date)
> # Konversi ke time series (menggunakan hargautupan saham)
> ts.datasaham <- ts(datasaham$Close)
> # Cek kembali struktur setelah perubahan
> str(datasaham)
'data.frame': 527 obs. of 2 variables:
 $ Date : Date, format: "2018-01-02" ...
 $ Close: num 6339 6251 6292 6354 6385 ...
> # Membuat Boxplot Harga Tutupan Saham
> y <- datasaham$Date
> x <- datasaham$Close
> boxplot(x, xlab="Harga Tutupan IHSG", col="lightblue")
> # Plot harga harian IHSG datasaham Covid-19
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line(color = "blue") +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020",
+         x = "Tanggal",
+         y = "Harga IHSG") +
+   theme(plot.title = element_text(face = "bold", hjust = 0.5),
+         plot.subtitle = element_text(hjust = 0.5),
+         panel.background = element_rect(fill = NA))
> # Menentukan proporsi data pelatihan (70%)
> p = 0.7
> # Menghitung jumlah data untuk pelatihan
> jumlah.train <- as.integer(p * nrow(datasaham))
> # Membuat plot
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line() +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020") +
+   geom_vline(aes(xintercept = Date[jumlah.train],
+                 col="Pemisah_data_train"), lty=2, lwd=.7) +
+   theme(plot.title = element_text(face = "bold", hjust = .5),
+         plot.subtitle = element_text(hjust = .5),
+         legend.position = "bottom",
+         panel.background = element_rect(fill=NA)) +
+   scale_color_manual(name = "", values = c(Pemisah_data_train = "blue"))
> # Menentukan data pelatihan
> datasaham[jumlah.train,1]
[1] "2019-07-16"
> train = 1:jumlah.train
> # Membuat time series untuk data pelatihan dan pengujian
> train.ts <- ts(datasaham$Close[train])
> testing.ts <- ts(datasaham$Close[-train], start = jumlah.train + 1)
> # Menentukan jumlah data pelatihan dan pengujian
> jumlah.train <- length(train.ts)
> jumlah.testing <- length(testing.ts)
> # Menampilkan hasil
> cat("Jumlah data pelatihan: ", jumlah.train, "\n")
Jumlah data pelatihan: 368
> cat("Jumlah data pengujian: ", jumlah.testing, "\n")
Jumlah data pengujian: 159
> #Identifikasi Stasioneritas = differencing
> library(tseries)
> par(mfrow=c(1,2))
> acf(train.ts, main = "ACF Plot for MA(q)")
> pacf(train.ts, main = "PACF Plot for AR(p)")
> adf.test(train.ts)
Augmented Dickey-Fuller Test
alternative: stationary

```


Type 1: no drift no trend

	lag	ADF	p.value
[1,]	0	-0.0266	0.636
[2,]	1	0.0534	0.659
[3,]	2	0.0209	0.650
[4,]	3	-0.0303	0.635
[5,]	4	-0.0633	0.626
[6,]	5	-0.0447	0.631

Type 2: with drift no trend

	lag	ADF	p.value
[1,]	0	-1.94	0.352
[2,]	1	-1.95	0.347
[3,]	2	-1.83	0.395
[4,]	3	-1.86	0.384
[5,]	4	-1.90	0.366
[6,]	5	-1.75	0.428

Type 3: with drift and trend

	lag	ADF	p.value
[1,]	0	-2.00	0.578
[2,]	1	-1.99	0.578
[3,]	2	-1.88	0.626
[4,]	3	-1.92	0.611
[5,]	4	-1.97	0.590
[6,]	5	-1.82	0.654

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> library(forecast)
> train.diff<-diff(train.ts, differences = 1)
> par(mfrow=c(1,2))
> acf(train.diff, main = "ACF Plot for MA(q)")
> pacf(train.diff, main = "PACF Plot for AR(p)")
> adf.test(train.diff)
```

Augmented Dickey-Fuller Test

alternative: stationary

Type 1: no drift no trend

	lag	ADF	p.value
[1,]	0	-18.73	0.01
[2,]	1	-14.31	0.01
[3,]	2	-11.43	0.01
[4,]	3	-9.68	0.01
[5,]	4	-9.34	0.01
[6,]	5	-8.46	0.01

Type 2: with drift no trend

	lag	ADF	p.value
[1,]	0	-18.71	0.01
[2,]	1	-14.30	0.01
[3,]	2	-11.42	0.01
[4,]	3	-9.67	0.01
[5,]	4	-9.33	0.01
[6,]	5	-8.45	0.01

Type 3: with drift and trend

	lag	ADF	p.value
[1,]	0	-18.69	0.01
[2,]	1	-14.29	0.01
[3,]	2	-11.43	0.01
[4,]	3	-9.68	0.01
[5,]	4	-9.35	0.01
[6,]	5	-8.48	0.01

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> #Identifikasi Model ARIMA dengan ACF, PACF, dan EACF
> acf(train.diff, main="ACF IHSG datasaham Covid-19")
> pacf(train.diff, main="PACF IHSG datasaham Covid-19")
```

```

> library(TSA)
> eacf(train.diff)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 0 x 0 0 x 0 0 0 0
1 x 0 0 0 0 0 x 0 0 0 x 0 0 0
2 0 x 0 0 0 0 0 0 0 0 0 0 0 0
3 x 0 x 0 0 0 0 0 0 0 0 0 0 0
4 x 0 x 0 0 0 0 0 0 0 0 0 0 0
5 x x x 0 x 0 0 0 0 x 0 0 0
6 x x x 0 0 x 0 0 0 x 0 0 0
7 x 0 x 0 x x 0 0 0 x 0 0 0
> #penentuan model ARIMA
> library(forecast)
> model_arima <- auto.arima(train.diff, trace = T)

Fitting models using approximations to speed things up...

ARIMA(2,0,2) with non-zero mean : Inf
ARIMA(0,0,0) with non-zero mean : 4000.573
ARIMA(1,0,0) with non-zero mean : 4000.939
ARIMA(0,0,1) with non-zero mean : 4002.384
ARIMA(0,0,0) with zero mean : 3998.555
ARIMA(1,0,1) with non-zero mean : 4002.264

Now re-fitting the best model(s) without approximations...

ARIMA(0,0,0) with zero mean : 3998.555

Best model: ARIMA(0,0,0) with zero mean

> summary(model_arima)
Series: train.diff
ARIMA(0,0,0) with zero mean

sigma^2 = 3140: log likelihood = -1998.27
AIC=3998.54 AICc=3998.55 BIC=4002.45

Training set error measures:
              ME      RMSE      MAE  MPE  MAPE      MASE
Training set 0.1706869 56.03319 43.27015 100  100 0.7219408
              ACF1
Training set 0.02276686
> model1 <- Arima(train.diff,order = c(0,1,0),method ="ML")
> summary(model1)
Series: train.diff
ARIMA(0,1,0)

sigma^2 = 6131: log likelihood = -2115.3
AIC=4232.61 AICc=4232.62 BIC=4236.51

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.1943243 78.19594 59.7728 28.24499 350.2904 0.9972792
              ACF1
Training set -0.4475857
> model2 <- Arima(train.diff,order = c(1,1,1),method ="ML")
> summary(model2)
Series: train.diff
ARIMA(1,1,1)

Coefficients:
      ar1      ma1
0.0257 -1.0000

```

s.e. 0.0524 0.0098

sigma^2 = 3164: log likelihood = -1996.16
AIC=3998.32 AICc=3998.38 BIC=4010.02

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.8664748	56.01853	43.22519	124.2181	134.1285	0.7211906

ACF1

Training set 0.005235341

```
> model3 <- Arima(train.diff,order = c(1,1,0),method ="ML")
```

```
> summary(model3)
```

Series: train.diff

ARIMA(1,1,0)

Coefficients:

ar1
-0.4504

s.e. 0.0468

sigma^2 = 4905: log likelihood = -2074.09
AIC=4152.18 AICc=4152.22 BIC=4159.99

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.3195567	69.84689	54.43677	132.7616	350.307	0.9082503

ACF1

Training set -0.1577695

```
> model4 <- Arima(train.diff,order = c(0,1,1),method ="ML")
```

```
> summary(model4)
```

Series: train.diff

ARIMA(0,1,1)

Coefficients:

ma1
-1.0000

s.e. 0.0101

sigma^2 = 3157: log likelihood = -1996.28
AIC=3996.55 AICc=3996.59 BIC=4004.36

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.8516645	56.03294	43.21613	127.1971	134.2284	0.7210395

ACF1

Training set 0.02896417

```
> model5 <- Arima(train.diff,order = c(2,1,2),method ="ML")
```

```
> summary(model5)
```

Series: train.diff

ARIMA(2,1,2)

Coefficients:

	ar1	ar2	ma1	ma2
	0.0450	-0.0712	-1.0176	0.0178
s.e.	0.6708	0.0545	0.6725	0.6720

sigma^2 = 3165: log likelihood = -1995.25
AIC=4000.5 AICc=4000.67 BIC=4020.01

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.8473477	55.87603	42.99562	115.388	124.126	0.7173604

ACF1

Training set 0.003261644

```

> model6 <- Arima(train.diff,order = c(1,1,2),method ="ML")
> summary(model6)
Series: train.diff
ARIMA(1,1,2)

Coefficients:
      ar1      ma1      ma2
-0.9063 -0.0635 -0.9365
s.e.    0.0891   0.0734   0.0734

sigma^2 = 3158: log likelihood = -1995.34
AIC=3998.69 AICc=3998.8 BIC=4014.3

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.8446919 55.88958 43.07947 115.5465 139.9961 0.7187593
      ACF1
Training set 0.003787936
> model7 <- Arima(train.diff,order = c(2,1,1),method ="ML")
> summary(model7)
Series: train.diff
ARIMA(2,1,1)

Coefficients:
      ar1      ar2      ma1
 0.0274 -0.0706 -1.0000
s.e.    0.0523   0.0523   0.0107

sigma^2 = 3156: log likelihood = -1995.25
AIC=3998.5 AICc=3998.61 BIC=4014.11

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.847598 55.86867 42.98933 115.3043 124.0011 0.7172555
      ACF1
Training set 0.003102997
> model8 <- Arima(train.diff,order = c(0,1,2),method ="ML")
> summary(model8)
Series: train.diff
ARIMA(0,1,2)

Coefficients:
      ma1      ma2
-0.9702 -0.0298
s.e.    0.0574   0.0565

sigma^2 = 3164: log likelihood = -1996.14
AIC=3998.28 AICc=3998.34 BIC=4009.98

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.868609 56.01603 43.2219 123.6222 133.8922 0.7211357
      ACF1
Training set 0.001366036
> model9 <- Arima(train.diff,order = c(2,1,0),method ="ML")
> summary(model9)
Series: train.diff
ARIMA(2,1,0)

Coefficients:
      ar1      ar2
-0.6127 -0.3593
s.e.    0.0490   0.0490

```

```
sigma^2 = 4286: log likelihood = -2049.04
AIC=4104.08 AICc=4104.15 BIC=4115.79
```

Training set error measures:

```
ME RMSE MAE MPE MAPE MASE
Training set 0.3293502 65.20124 49.86791 103.4904 278.9366 0.8320212
```

```
ACF1
Training set -0.09295882
```

```
> #uji residual
> sisaan <- model2$residuals
> checkresiduals(sisaan)
```

Ljung-Box test

```
data: Residuals
Q* = 16.764, df = 10, p-value = 0.07976
```

Model df: 0. Total lags used: 10

```
> tsdiag(model2)
> par(mfrow = c(2,2))
> qqnorm(sisaan)
> qqline(sisaan, col = "blue", lwd = 2)
> plot(sisaan, type="o",
+       ylab = "Sisaan", xlab = "Order", main = "Sisaan Model 4 vs Order")
> abline(h = 0, col='red')
> acf(sisaan)
> pacf(sisaan)
> adf.test(sisaan)
```

Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend

	lag	ADF	p.value
[1,]	0	-19.00	0.01
[2,]	1	-14.43	0.01
[3,]	2	-11.59	0.01
[4,]	3	-9.80	0.01
[5,]	4	-9.39	0.01
[6,]	5	-8.53	0.01

Type 2: with drift no trend

	lag	ADF	p.value
[1,]	0	-18.98	0.01
[2,]	1	-14.41	0.01
[3,]	2	-11.57	0.01
[4,]	3	-9.78	0.01
[5,]	4	-9.38	0.01
[6,]	5	-8.52	0.01

Type 3: with drift and trend

	lag	ADF	p.value
[1,]	0	-18.97	0.01
[2,]	1	-14.43	0.01
[3,]	2	-11.61	0.01
[4,]	3	-9.84	0.01
[5,]	4	-9.44	0.01
[6,]	5	-8.59	0.01

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> jarque.bera.test(sisaan)
```

Jarque Bera Test

```
data: sisaan
X-squared = 21.17, df = 2, p-value = 2.529e-05
```

```
> Box.test(sisaan, type = "Ljung")
```

Box-Ljung test

```
data: sisaan
X-squared = 0.010141, df = 1, p-value = 0.9198
```

```
> t.test(sisaan, mu = 0, conf.level = 0.95)
```

One sample t-test

```
data: sisaan
t = 0.29595, df = 366, p-value = 0.7674
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -4.890917  6.623866
sample estimates:
mean of x
0.8664748
```

```
> #white Test
```

```
> white_test <- bptest(sisaan ~ fitted(model2) + I(fitted(model2)^2))
```

```
> print(white_test)
```

studentized Breusch-Pagan test

```
data: sisaan ~ fitted(model2) + I(fitted(model2)^2)
BP = 11.763, df = 2, p-value = 0.00279
```

```
> #uji box cox penanganan heterogenitas variansi, dilakukan transformasi kuadratik
```

```
> par(mfrow=c(1,1))
```

```
> train.ts.new <- (train.ts)^2
```

```
> index <- seq(1:368)
```

```
> bc = boxcox(train.ts.new~index, lambda = seq(-2, 4, by=.01))
```

```
> #Nilai Rounded Lambda
```

```
> lambda <- bc$x[which.max(bc$y)]
```

```
> lambda
```

```
[1] 0.99
```

```
> #SK
```

```
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
```

```
[1] -0.35 -0.34 -0.33 -0.32 -0.31 -0.30 -0.29 -0.28 -0.27 -0.26
[11] -0.25 -0.24 -0.23 -0.22 -0.21 -0.20 -0.19 -0.18 -0.17 -0.16
[21] -0.15 -0.14 -0.13 -0.12 -0.11 -0.10 -0.09 -0.08 -0.07 -0.06
[31] -0.05 -0.04 -0.03 -0.02 -0.01  0.00  0.01  0.02  0.03  0.04
[41]  0.05  0.06  0.07  0.08  0.09  0.10  0.11  0.12  0.13  0.14
[51]  0.15  0.16  0.17  0.18  0.19  0.20  0.21  0.22  0.23  0.24
[61]  0.25  0.26  0.27  0.28  0.29  0.30  0.31  0.32  0.33  0.34
[71]  0.35  0.36  0.37  0.38  0.39  0.40  0.41  0.42  0.43  0.44
[81]  0.45  0.46  0.47  0.48  0.49  0.50  0.51  0.52  0.53  0.54
[91]  0.55  0.56  0.57  0.58  0.59  0.60  0.61  0.62  0.63  0.64
[101] 0.65  0.66  0.67  0.68  0.69  0.70  0.71  0.72  0.73  0.74
[111] 0.75  0.76  0.77  0.78  0.79  0.80  0.81  0.82  0.83  0.84
[121] 0.85  0.86  0.87  0.88  0.89  0.90  0.91  0.92  0.93  0.94
[131] 0.95  0.96  0.97  0.98  0.99  1.00  1.01  1.02  1.03  1.04
[141] 1.05  1.06  1.07  1.08  1.09  1.10  1.11  1.12  1.13  1.14
[151] 1.15  1.16  1.17  1.18  1.19  1.20  1.21  1.22  1.23  1.24
[161] 1.25  1.26  1.27  1.28  1.29  1.30  1.31  1.32  1.33  1.34
[171] 1.35  1.36  1.37  1.38  1.39  1.40  1.41  1.42  1.43  1.44
[181] 1.45  1.46  1.47  1.48  1.49  1.50  1.51  1.52  1.53  1.54
[191] 1.55  1.56  1.57  1.58  1.59  1.60  1.61  1.62  1.63  1.64
[201] 1.65  1.66  1.67  1.68  1.69  1.70  1.71  1.72  1.73  1.74
[211] 1.75  1.76  1.77  1.78  1.79  1.80  1.81  1.82  1.83  1.84
```

```
[221] 1.85 1.86 1.87 1.88 1.89 1.90 1.91 1.92 1.93 1.94
[231] 1.95 1.96 1.97 1.98 1.99 2.00 2.01 2.02 2.03 2.04
[241] 2.05 2.06 2.07 2.08 2.09 2.10 2.11 2.12 2.13 2.14
[251] 2.15 2.16 2.17 2.18 2.19 2.20 2.21 2.22 2.23 2.24
[261] 2.25 2.26 2.27 2.28 2.29 2.30 2.31 2.32 2.33 2.34
[271] 2.35
```

```
> plot.ts(train.ts.new, lty=1, xlab="waktu", ylab="Data Transformasi", main="Plot Transformasi Data")
```

```
> #eksekusi peramalan data train (70%)
> ramalan.da <- forecast::forecast(model2, h = 159)
> data.ramalan.da <- ramalan.da$mean
> plot(ramalan.da)
> summary(ramalan.da)
```

Forecast method: ARIMA(1,1,1)

Model Information:

Series: train.diff

ARIMA(1,1,1)

Coefficients:

```
      ar1      ma1
      0.0257 -1.0000
s.e.  0.0524  0.0098
```

sigma^2 = 3164: log likelihood = -1996.16

AIC=3998.32 AICc=3998.38 BIC=4010.02

Error measures:

```
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.8664748 56.01853 43.22519 124.2181 134.1285 0.7211906
              ACF1
Training set 0.005235341
```

Forecasts:

```
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
369      -0.2608199 -72.44479 71.92315 -110.6567 110.1350
370       0.1523045 -72.06048 72.36509 -110.2876 110.5922
371       0.1629098 -72.05002 72.37584 -110.2772 110.6031
372       0.1631820 -72.04975 72.37612 -110.2770 110.6033
373       0.1631890 -72.04975 72.37612 -110.2770 110.6033
374       0.1631892 -72.04975 72.37612 -110.2770 110.6033
375       0.1631892 -72.04975 72.37612 -110.2770 110.6033
376       0.1631892 -72.04975 72.37612 -110.2770 110.6033
377       0.1631892 -72.04975 72.37612 -110.2770 110.6033
378       0.1631892 -72.04975 72.37612 -110.2770 110.6033
379       0.1631892 -72.04975 72.37612 -110.2770 110.6033
380       0.1631892 -72.04975 72.37612 -110.2770 110.6033
381       0.1631892 -72.04975 72.37612 -110.2770 110.6033
382       0.1631892 -72.04975 72.37612 -110.2770 110.6033
383       0.1631892 -72.04975 72.37612 -110.2770 110.6033
384       0.1631892 -72.04975 72.37612 -110.2770 110.6033
385       0.1631892 -72.04975 72.37612 -110.2770 110.6033
386       0.1631892 -72.04975 72.37612 -110.2770 110.6033
387       0.1631892 -72.04975 72.37612 -110.2770 110.6033
388       0.1631892 -72.04975 72.37612 -110.2770 110.6033
389       0.1631892 -72.04975 72.37612 -110.2770 110.6033
390       0.1631892 -72.04975 72.37612 -110.2770 110.6033
391       0.1631892 -72.04975 72.37612 -110.2770 110.6033
392       0.1631892 -72.04975 72.37612 -110.2770 110.6033
393       0.1631892 -72.04975 72.37612 -110.2770 110.6033
394       0.1631892 -72.04975 72.37612 -110.2770 110.6033
395       0.1631892 -72.04975 72.37612 -110.2770 110.6033
```

[illegible]

[illegible]

```

524      0.1631892 -72.04975 72.37612 -110.2770 110.6033
525      0.1631892 -72.04975 72.37612 -110.2770 110.6033
526      0.1631892 -72.04975 72.37612 -110.2770 110.6033
527      0.1631892 -72.04975 72.37612 -110.2770 110.6033
> #pengembalian ke nilai awal
> pt_1 <- train.ts.new[368] #nilai akhir data latih
> head(pt_1)
[1] 40984066
> train.ts[368]
[1] 6401.88
> hasil.forc.Diff <- data.ramalan.da
> hasil <- diffinv(hasil.forc.Diff, differences = 1) + pt_1
> hasil
Time Series:
Start = 368
End = 527
Frequency = 1
 [1] 40984066 40984066 40984066 40984066 40984066 40984066 40984067
 [8] 40984067 40984067 40984067 40984067 40984067 40984068 40984068
[15] 40984068 40984068 40984068 40984068 40984069 40984069 40984069
[22] 40984069 40984069 40984069 40984070 40984070 40984070 40984070
[29] 40984070 40984070 40984070 40984071 40984071 40984071 40984071
[36] 40984071 40984071 40984072 40984072 40984072 40984072 40984072
[43] 40984072 40984073 40984073 40984073 40984073 40984073 40984073
[50] 40984074 40984074 40984074 40984074 40984074 40984074 40984075
[57] 40984075 40984075 40984075 40984075 40984075 40984076 40984076
[64] 40984076 40984076 40984076 40984076 40984077 40984077 40984077
[71] 40984077 40984077 40984077 40984078 40984078 40984078 40984078
[78] 40984078 40984078 40984078 40984079 40984079 40984079 40984079
[85] 40984079 40984079 40984080 40984080 40984080 40984080 40984080
[92] 40984080 40984081 40984081 40984081 40984081 40984081 40984081
[99] 40984082 40984082 40984082 40984082 40984082 40984082 40984083
[106] 40984083 40984083 40984083 40984083 40984083 40984084 40984084
[113] 40984084 40984084 40984084 40984084 40984085 40984085 40984085
[120] 40984085 40984085 40984085 40984086 40984086 40984086 40984086
[127] 40984086 40984086 40984086 40984087 40984087 40984087 40984087
[134] 40984087 40984087 40984088 40984088 40984088 40984088 40984088
[141] 40984088 40984089 40984089 40984089 40984089 40984089 40984089
[148] 40984090 40984090 40984090 40984090 40984090 40984090 40984091
[155] 40984091 40984091 40984091 40984091 40984091 40984091 40984092
> hasil1 <- abs(hasil)^(1/2) #hasil peramalan arima nilai data telah dikembali
kan ke asli
> pt_1 <- c(train.ts.new[368],testing.ts) #nilai akhir data latih
> pt_1
 [1] 40984066.036      6394.609      6403.294      6456.539      6433.547
 [6]      6403.810      6384.987      6401.365      6325.237      6299.035
[11]      6376.996      6390.505      6381.542      6340.180      6175.703
[16]      6119.471      6204.195      6274.671      6282.132      6250.595
[21]      6210.962      6267.335      6257.586      6286.657      6296.715
[26]      6295.738      6252.967      6239.245      6255.597      6214.510
[31]      6278.171      6281.646      6289.119      6328.470      6290.546
[36]      6261.590      6269.664      6306.803      6308.950      6326.213
[41]      6336.673      6381.954      6342.174      6334.843      6219.435
[46]      6236.690      6276.633      6244.470      6231.473      6206.199
[51]      6137.608      6146.404      6230.334      6196.889      6169.102
[56]      6138.250      6055.425      6038.529      6061.252      6000.582
[61]      6039.601      6029.160      6023.641      6105.800      6126.877
[66]      6158.166      6169.592      6181.014      6191.947      6198.987
[71]      6225.497      6257.806      6339.647      6252.345      6265.384
[76]      6281.138      6295.747      6228.317      6207.191      6180.344
[81]      6264.152      6217.545      6165.624      6177.986      6148.740
[86]      6180.992      6142.501      6098.950      6128.345      6122.625
[91]      6152.090      6155.109      6117.364      6100.242      6070.762
[96]      6026.188      6023.039      5953.060      6011.830      6130.055

```

```

[101]    6133.896    6112.879    6152.117    6186.868    6193.791
[106]    6183.505    6180.099    6139.397    6197.318    6211.592
[111]    6244.352    6287.250    6249.930    6284.372    6305.910
[116]    6319.443    6329.314    6299.539    6283.581    6323.466
[121]    6257.403    6279.346    6225.686    6274.493    6274.941
[126]    6296.567    6325.406    6283.365    6286.048    6291.657
[131]    6245.043    6238.153    6233.453    6249.210    6244.109
[136]    6133.208    6111.184    6113.045    6057.596    5940.048
[141]    5884.170    5922.339    5978.510    5987.145    5999.607
[146]    5952.083    5954.397    5913.081    5871.954    5866.945
[151]    5867.523    5886.962    5928.791    5942.487    5882.255
[156]    5807.050    5787.138    5688.921    5535.694    5452.704
> coef(model2)
      ar1      ma1
0.02567077 -0.99999943
> library(TSA)
> model2 <- Arima(train.ts,order = c(1,1,1))
> for (i in 1:15) {
+   ArchTest <- ArchTest(model2$residuals, lags=i, demean=TRUE)
+   cat("P Value LM Test lag ke", i,"adalah" , ArchTest$p.value, "\n") }
P Value LM Test lag ke 1 adalah 0.01607126
P Value LM Test lag ke 2 adalah 0.05167288
P Value LM Test lag ke 3 adalah 0.1123621
P Value LM Test lag ke 4 adalah 0.07808124
P Value LM Test lag ke 5 adalah 0.1205032
P Value LM Test lag ke 6 adalah 0.1739571
P Value LM Test lag ke 7 adalah 0.1131096
P Value LM Test lag ke 8 adalah 0.1590189
P Value LM Test lag ke 9 adalah 0.1720806
P Value LM Test lag ke 10 adalah 0.242177
P Value LM Test lag ke 11 adalah 0.2800074
P Value LM Test lag ke 12 adalah 0.3544076
P Value LM Test lag ke 13 adalah 0.4358759
P Value LM Test lag ke 14 adalah 0.5267488
P Value LM Test lag ke 15 adalah 0.2272605
> #pengecekan efek GARCH
> train.diff.new <- (train.diff)^2
> sisaan.new <- (sisaan)^2
> garch01 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garcho
rder=c(0,1)))
> garch01

*-----*
*          GARCH Model Spec          *
*-----*

Conditional Variance Dynamics
-----
GARCH Model      : sGARCH(0,1)
Variance Targeting : FALSE

Conditional Mean Dynamics
-----
Mean Model      : ARFIMA(1,0,1)
Include Mean    : FALSE
GARCH-in-Mean   : FALSE

Conditional Distribution
-----
Distribution    : norm
Includes Skew   : FALSE
Includes Shape  : FALSE

```

Includes Lambda: FALSE

```
> garch01fit <- rugarch::ugarchfit(data =train.diff, spec = garch01)
> garch01fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(0,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate  Std. Error  t value Pr(>|t|)
ar1    -0.90425    0.129999   -6.9559 0.00000
ma1     0.93608    0.106914    8.7555 0.00000
omega   1.16417    0.962135    1.2100 0.22628
beta1   0.99900    0.000275 3626.7535 0.00000
```

Robust Standard Errors:

```
      Estimate  Std. Error  t value Pr(>|t|)
ar1    -0.90425    0.166124   -5.4432 0.000000
ma1     0.93608    0.137524    6.8067 0.000000
omega   1.16417    0.336714    3.4574 0.000545
beta1   0.99900    0.000229 4354.5515 0.000000
```

LogLikelihood : -1996.192

Information Criteria

```
-----
Akaike          10.900
Bayes            10.943
Shibata          10.900
Hannan-Quinn    10.917
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
                        statistic p-value
Lag[1]                  0.003382 0.9536
Lag[2*(p+q)+(p+q)-1][5] 1.616723 0.9951
Lag[4*(p+q)+(p+q)-1][9] 4.460533 0.5808
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                        statistic p-value
Lag[1]                  5.452 0.01955
Lag[2*(p+q)+(p+q)-1][2] 5.461 0.03062
Lag[4*(p+q)+(p+q)-1][5] 6.371 0.07346
d.o.f=1
```

Weighted ARCH LM Tests

```
-----
      Statistic Shape Scale P-Value
ARCH Lag[2]    0.01881 0.500 2.000 0.8909
ARCH Lag[4]    0.94892 1.397 1.611 0.7232
ARCH Lag[6]    1.67577 2.222 1.500 0.7531
```

Nyblom stability test

Joint Statistic: 3.3177

Individual Statistics:

ar1 0.08520

ma1 0.06393

omega 0.80349

beta1 0.86110

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic: 1.07 1.24 1.6

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.8317	0.406104	
Negative Sign Bias	3.2679	0.001187	***
Positive Sign Bias	0.8254	0.409686	
Joint Effect	16.1591	0.001052	***

Adjusted Pearson Goodness-of-Fit Test:

group	statistic	p-value(g-1)
1 20	16.71	0.6098
2 30	35.26	0.1961
3 40	34.85	0.6595
4 50	55.21	0.2517

Elapsed time : 0.1269071

```
> garch11 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garchO
rder=c(1,1)))
> garch11fit <- rugarch::ugarchfit(spec = garch11, data = train.diff)
> garch11fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

GARCH Model	: sGARCH(1,1)
Mean Model	: ARFIMA(1,0,1)
Distribution	: norm

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
ar1	-0.906238	0.081339	-11.1415	0.000000
ma1	0.940605	0.063141	14.8968	0.000000
omega	42.104464	42.379946	0.9935	0.320466
alpha1	0.032319	0.014163	2.2818	0.022498
beta1	0.953683	0.020706	46.0583	0.000000

Robust Standard Errors:

	Estimate	Std. Error	t value	Pr(> t)
ar1	-0.906238	0.067736	-13.3790	0.000000
ma1	0.940605	0.051278	18.3433	0.000000
omega	42.104464	41.814339	1.0069	0.313964

```
alpha1 0.032319    0.013423    2.4077 0.016053
beta1   0.953683    0.021675   43.9986 0.000000
```

LogLikelihood : -1991.135

Information Criteria

```
-----
Akaike          10.878
Bayes           10.931
Shibata         10.878
Hannan-Quinn    10.899
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
                        statistic p-value
Lag[1]                  0.009157 0.9238
Lag[2*(p+q)+(p+q)-1][5] 1.070595 1.0000
Lag[4*(p+q)+(p+q)-1][9] 3.450679 0.8112
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                        statistic p-value
Lag[1]                  2.307 0.1288
Lag[2*(p+q)+(p+q)-1][5] 3.765 0.2849
Lag[4*(p+q)+(p+q)-1][9] 5.252 0.3933
d.o.f=2
```

Weighted ARCH LM Tests

```
-----
Statistic Shape Scale P-Value
ARCH Lag[3]      1.156 0.500 2.000 0.2823
ARCH Lag[5]      1.769 1.440 1.667 0.5245
ARCH Lag[7]      2.790 2.315 1.543 0.5540
```

Nyblom stability test

```
-----
Joint Statistic: 0.6339
Individual Statistics:
ar1    0.08301
ma1    0.06094
omega  0.33766
alpha1 0.27984
beta1  0.30060
```

Asymptotic Critical values (10% 5% 1%)

```
Joint Statistic:      1.28 1.47 1.88
Individual Statistic: 0.35 0.47 0.75
```

Sign Bias Test

```
-----
t-value    prob sig
Sign Bias   0.7844 0.433332
Negative Sign Bias 2.4280 0.015670 **
Positive Sign Bias 1.5020 0.133959
Joint Effect 12.7877 0.005119 ***
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1      20      20.74      0.3514
```

2	30	30.19	0.4044
3	40	42.26	0.3318
4	50	49.49	0.4538

Elapsed time : 0.117476

```
> garch12 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "SGARCH", garchO
rder=c(1,2)))
> garch12fit <- rugarch::ugarchfit(spec = garch12, data = train.diff)
> garch12fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(1,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate Std. Error t value Pr(>|t|)
ar1      -0.900185   0.091773  -9.80880 0.000000
ma1       0.933512   0.073944  12.62451 0.000000
omega  109.746760  121.210338   0.90542 0.365241
alpha1    0.079482   0.037534   2.11760 0.034209
beta1     0.027685   0.065799   0.42075 0.673938
beta2     0.857675   0.070291  12.20182 0.000000
```

Robust Standard Errors:

```
      Estimate Std. Error t value Pr(>|t|)
ar1      -0.900185   0.074003 -12.16416 0.000000
ma1       0.933512   0.059756  15.62203 0.000000
omega  109.746760  138.084767   0.79478 0.426743
alpha1    0.079482   0.045617   1.74238 0.081443
beta1     0.027685   0.065091   0.42533 0.670600
beta2     0.857675   0.078698  10.89834 0.000000
```

LogLikelihood : -1989.502

Information Criteria

```
-----
Akaike          10.875
Bayes           10.939
Shibata         10.874
Hannan-Quinn    10.900
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
              statistic p-value
Lag[1]              0.01313 0.9088
Lag[2*(p+q)+(p+q)-1][5] 1.01750 1.0000
Lag[4*(p+q)+(p+q)-1][9] 3.16645 0.8638
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

	statistic	p-value
Lag[1]	0.3824	0.5363
Lag[2*(p+q)+(p+q)-1][8]	3.7870	0.5431
Lag[4*(p+q)+(p+q)-1][14]	5.2808	0.7340

d.o.f=3

Weighted ARCH LM Tests

	Statistic	Shape	Scale	P-value
ARCH Lag[4]	0.9115	0.500	2.000	0.3397
ARCH Lag[6]	2.2989	1.461	1.711	0.4284
ARCH Lag[8]	2.8815	2.368	1.583	0.5659

Nyblom stability test

Joint Statistic: 0.6727

Individual Statistics:

ar1 0.08098

ma1 0.06536

omega 0.29182

alpha1 0.28077

beta1 0.27410

beta2 0.26899

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic: 1.49 1.68 2.12

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.6557	0.51242	
Negative Sign Bias	1.7710	0.07740	*
Positive Sign Bias	1.8146	0.07042	*
Joint Effect	10.7503	0.01316	**

Adjusted Pearson Goodness-of-Fit Test:

group	statistic	p-value(g-1)
1 20	19.87	0.4027
2 30	25.45	0.6546
3 40	37.25	0.5498
4 50	54.39	0.2768

Elapsed time : 0.1763299

```
> garch22 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "SGARCH",garchO
rder=c(2,2)))
> garch22fit <- rugarch::ugarchfit(spec = garch22, data = train.diff)
> garch22fit
```

```
*-----*
*                GARCH Model Fit                *
*-----*
```

Conditional Variance Dynamics

```
GARCH Model      : SGARCH(2,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```


Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
ar1	-0.900074	0.091899	-9.79421	0.00000
ma1	0.933419	0.074126	12.59226	0.00000
omega	109.669952	125.449369	0.87422	0.38200
alpha1	0.079506	0.041575	1.91236	0.05583
alpha2	0.000000	0.033074	0.00000	1.00000
beta1	0.027673	0.090259	0.30659	0.75915
beta2	0.857662	0.079070	10.84691	0.00000

Robust Standard Errors:

	Estimate	Std. Error	t value	Pr(> t)
ar1	-0.900074	0.074157	-12.13743	0.000000
ma1	0.933419	0.059816	15.60491	0.000000
omega	109.669952	149.771841	0.73225	0.464018
alpha1	0.079506	0.043015	1.84834	0.064553
alpha2	0.000000	0.033072	0.00000	1.000000
beta1	0.027673	0.081879	0.33797	0.735385
beta2	0.857662	0.069444	12.35042	0.000000

LogLikelihood : -1989.502

Information Criteria

Akaike	10.880
Bayes	10.955
Shibata	10.879
Hannan-Quinn	10.910

Weighted Ljung-Box Test on Standardized Residuals

	statistic	p-value
Lag[1]	0.01321	0.9085
Lag[2*(p+q)+(p+q)-1][5]	1.01728	1.0000
Lag[4*(p+q)+(p+q)-1][9]	3.16606	0.8639
d.o.f=2		
H0 : No serial correlation		

Weighted Ljung-Box Test on Standardized Squared Residuals

	statistic	p-value
Lag[1]	0.3816	0.5367
Lag[2*(p+q)+(p+q)-1][11]	4.6638	0.6320
Lag[4*(p+q)+(p+q)-1][19]	7.1813	0.7647
d.o.f=4		

Weighted ARCH LM Tests

	Statistic	Shape	Scale	P-Value
ARCH Lag[5]	0.5414	0.500	2.000	0.4618
ARCH Lag[7]	1.9470	1.473	1.746	0.5158
ARCH Lag[9]	2.5589	2.402	1.619	0.6492

Nyblom stability test

Joint Statistic: 0.8905

Individual Statistics:

ar1	0.08099
ma1	0.06529
omega	0.29287
alpha1	0.28175

```
alpha2 0.21031
beta1  0.27517
beta2  0.27002
```

```
Asymptotic Critical values (10% 5% 1%)
Joint Statistic:      1.69 1.9 2.35
Individual Statistic: 0.35 0.47 0.75
```

Sign Bias Test

```
-----
Sign Bias          t-value   prob   sig
Negative Sign Bias 1.7703 0.07752  *
Positive Sign Bias 1.8149 0.07037  *
Joint Effect       10.7486 0.01317 **
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1    20      19.87      0.4027
2    30      25.45      0.6546
3    40      37.25      0.5498
4    50      54.39      0.2768
```

Elapsed time : 0.205591

```
> #eksekusi peramalan dengan efek garch
> par(mfrow=c(1,1))
> forc12<- ugarchforecast(fitORspec = garch12fit, n.ahead = 159, n.roll = 0)
> plot(forc12, which= 1)
> forc12
```

```
*-----*
*          GARCH Model Forecast          *
*-----*
```

```
Model: sGARCH
Horizon: 159
Roll Steps: 0
Out of Sample: 0
```

0-roll forecast [T0=368-01-01]:

```
Series Sigma
T+1  -4.133e+00 44.07
T+2   3.721e+00 46.39
T+3  -3.349e+00 44.79
T+4   3.015e+00 46.59
T+5  -2.714e+00 45.42
T+6   2.443e+00 46.82
T+7  -2.199e+00 45.98
T+8   1.980e+00 47.08
T+9  -1.782e+00 46.48
T+10  1.604e+00 47.36
T+11 -1.444e+00 46.94
T+12  1.300e+00 47.64
T+13 -1.170e+00 47.36
T+14  1.053e+00 47.92
T+15 -9.483e-01 47.74
T+16  8.536e-01 48.20
T+17 -7.684e-01 48.10
T+18  6.917e-01 48.48
T+19 -6.227e-01 48.44
T+20  5.605e-01 48.76
T+21 -5.046e-01 48.75
```

T+22	4.542e-01	49.02
T+23	-4.089e-01	49.05
T+24	3.681e-01	49.28
T+25	-3.313e-01	49.33
T+26	2.982e-01	49.53
T+27	-2.685e-01	49.59
T+28	2.417e-01	49.78
T+29	-2.176e-01	49.85
T+30	1.958e-01	50.01
T+31	-1.763e-01	50.09
T+32	1.587e-01	50.24
T+33	-1.429e-01	50.32
T+34	1.286e-01	50.45
T+35	-1.158e-01	50.54
T+36	1.042e-01	50.66
T+37	-9.381e-02	50.75
T+38	8.444e-02	50.87
T+39	-7.601e-02	50.95
T+40	6.843e-02	51.06
T+41	-6.160e-02	51.14
T+42	5.545e-02	51.24
T+43	-4.991e-02	51.33
T+44	4.493e-02	51.42
T+45	-4.045e-02	51.50
T+46	3.641e-02	51.59
T+47	-3.278e-02	51.67
T+48	2.950e-02	51.76
T+49	-2.656e-02	51.84
T+50	2.391e-02	51.92
T+51	-2.152e-02	51.99
T+52	1.937e-02	52.07
T+53	-1.744e-02	52.14
T+54	1.570e-02	52.22
T+55	-1.413e-02	52.29
T+56	1.272e-02	52.36
T+57	-1.145e-02	52.42
T+58	1.031e-02	52.49
T+59	-9.279e-03	52.56
T+60	8.353e-03	52.62
T+61	-7.519e-03	52.68
T+62	6.769e-03	52.74
T+63	-6.093e-03	52.80
T+64	5.485e-03	52.86
T+65	-4.938e-03	52.92
T+66	4.445e-03	52.98
T+67	-4.001e-03	53.03
T+68	3.602e-03	53.09
T+69	-3.242e-03	53.14
T+70	2.919e-03	53.19
T+71	-2.627e-03	53.25
T+72	2.365e-03	53.30
T+73	-2.129e-03	53.35
T+74	1.916e-03	53.39
T+75	-1.725e-03	53.44
T+76	1.553e-03	53.49
T+77	-1.398e-03	53.53
T+78	1.258e-03	53.58
T+79	-1.133e-03	53.62
T+80	1.020e-03	53.67
T+81	-9.179e-04	53.71
T+82	8.263e-04	53.75
T+83	-7.438e-04	53.79
T+84	6.696e-04	53.83
T+85	-6.028e-04	53.87

T+86	5.426e-04	53.91
T+87	-4.884e-04	53.94
T+88	4.397e-04	53.98
T+89	-3.958e-04	54.02
T+90	3.563e-04	54.05
T+91	-3.207e-04	54.09
T+92	2.887e-04	54.12
T+93	-2.599e-04	54.15
T+94	2.340e-04	54.19
T+95	-2.106e-04	54.22
T+96	1.896e-04	54.25
T+97	-1.707e-04	54.28
T+98	1.536e-04	54.31
T+99	-1.383e-04	54.34
T+100	1.245e-04	54.37
T+101	-1.121e-04	54.40
T+102	1.009e-04	54.43
T+103	-9.081e-05	54.45
T+104	8.174e-05	54.48
T+105	-7.358e-05	54.51
T+106	6.624e-05	54.53
T+107	-5.963e-05	54.56
T+108	5.368e-05	54.58
T+109	-4.832e-05	54.61
T+110	4.349e-05	54.63
T+111	-3.915e-05	54.66
T+112	3.525e-05	54.68
T+113	-3.173e-05	54.70
T+114	2.856e-05	54.72
T+115	-2.571e-05	54.75
T+116	2.314e-05	54.77
T+117	-2.083e-05	54.79
T+118	1.875e-05	54.81
T+119	-1.688e-05	54.83
T+120	1.520e-05	54.85
T+121	-1.368e-05	54.87
T+122	1.231e-05	54.89
T+123	-1.109e-05	54.90
T+124	9.979e-06	54.92
T+125	-8.983e-06	54.94
T+126	8.086e-06	54.96
T+127	-7.279e-06	54.98
T+128	6.552e-06	54.99
T+129	-5.898e-06	55.01
T+130	5.310e-06	55.03
T+131	-4.780e-06	55.04
T+132	4.303e-06	55.06
T+133	-3.873e-06	55.07
T+134	3.487e-06	55.09
T+135	-3.139e-06	55.10
T+136	2.825e-06	55.12
T+137	-2.543e-06	55.13
T+138	2.289e-06	55.15
T+139	-2.061e-06	55.16
T+140	1.855e-06	55.17
T+141	-1.670e-06	55.19
T+142	1.503e-06	55.20
T+143	-1.353e-06	55.21
T+144	1.218e-06	55.22
T+145	-1.097e-06	55.24
T+146	9.871e-07	55.25
T+147	-8.886e-07	55.26
T+148	7.999e-07	55.27
T+149	-7.201e-07	55.28

```

T+150 6.482e-07 55.29
T+151 -5.835e-07 55.30
T+152 5.252e-07 55.31
T+153 -4.728e-07 55.33
T+154 4.256e-07 55.34
T+155 -3.831e-07 55.35
T+156 3.449e-07 55.36
T+157 -3.105e-07 55.37
T+158 2.795e-07 55.37
T+159 -2.516e-07 55.38

```

```

> hasil.forc.Diff12 <- forc12@forecast$seriesFor[,1]
> hasil.forc.Diff12

```

T+1	T+2	T+3	T+4	T+5
-4.133208e+00	3.720651e+00	-3.349273e+00	3.014965e+00	-2.714025e+00
2.443124e+00	-2.199263e+00	1.979743e+00	-1.782134e+00	1.604250e+00
-1.444121e+00	1.299976e+00	-1.170219e+00	1.053413e+00	-9.482662e-01
8.536148e-01	-7.684110e-01	6.917118e-01	-6.226684e-01	5.605166e-01
-5.045685e-01	4.542049e-01	-4.088683e-01	3.680570e-01	-3.313193e-01
2.982486e-01	-2.684788e-01	2.416805e-01	-2.175571e-01	1.958416e-01
-1.762936e-01	1.586968e-01	-1.428564e-01	1.285972e-01	-1.157612e-01
1.042065e-01	-9.380509e-02	8.444191e-02	-7.601332e-02	6.842603e-02
-6.159607e-02	5.544784e-02	-4.991330e-02	4.493119e-02	-4.044637e-02
3.640920e-02	-3.277501e-02	2.950356e-02	-2.655866e-02	2.390770e-02
-2.152134e-02	1.937319e-02	-1.743945e-02	1.569872e-02	-1.413175e-02
1.272119e-02	-1.145142e-02	1.030839e-02	-9.279457e-03	8.353225e-03
-7.519446e-03	6.768890e-03	-6.093252e-03	5.485052e-03	-4.937560e-03
4.444716e-03	-4.001066e-03	3.601698e-03	-3.242194e-03	2.918573e-03
-2.627255e-03	2.365015e-03	-2.128950e-03	1.916449e-03	-1.725158e-03
1.552961e-03	-1.397952e-03	1.258415e-03	-1.132806e-03	1.019734e-03
-9.179493e-04	8.263239e-04	-7.438442e-04	6.695972e-04	-6.027612e-04
5.425964e-04	-4.884370e-04	4.396835e-04	-3.957964e-04	3.562899e-04
-3.207267e-04	2.887133e-04	-2.598953e-04	2.339538e-04	-2.106016e-04
1.895804e-04	-1.706573e-04	1.536231e-04	-1.382892e-04	1.244858e-04
-1.120602e-04	1.008749e-04	-9.080606e-05	8.174223e-05	-7.358310e-05
6.623839e-05	-5.962678e-05	5.367512e-05	-4.831752e-05	4.349470e-05
-3.915326e-05	3.524517e-05	-3.172716e-05	2.856031e-05	-2.570955e-05
2.314335e-05	-2.083329e-05	1.875381e-05	-1.688189e-05	1.519682e-05
-1.367995e-05	1.231448e-05	-1.108531e-05	9.978823e-06	-8.982784e-06

```

8.086165e-06 -7.279042e-06 6.552483e-06 -5.898445e-06 5.309690e-06
T+131 T+132 T+133 T+134 T+135
-4.779702e-06 4.302615e-06 -3.873148e-06 3.486549e-06 -3.138538e-06
T+136 T+137 T+138 T+139 T+140
2.825264e-06 -2.543259e-06 2.289403e-06 -2.060886e-06 1.855178e-06
T+141 T+142 T+143 T+144 T+145
-1.670003e-06 1.503311e-06 -1.353258e-06 1.218182e-06 -1.096589e-06
T+146 T+147 T+148 T+149 T+150
9.871325e-07 -8.886016e-07 7.999056e-07 -7.200628e-07 6.481895e-07
T+151 T+152 T+153 T+154 T+155
-5.834903e-07 5.252491e-07 -4.728212e-07 4.256264e-07 -3.831424e-07
T+156 T+157 T+158 T+159
3.448990e-07 -3.104728e-07 2.794828e-07 -2.515862e-07
> hasilgarch12 <- diffinv(hasil.forc.Diff12, differences = 1)+ pt_1
> hasilgarch12
[1] 40984066.036 6390.476 6402.881 6452.777 6432.800
[6] 6400.349 6383.969 6398.148 6324.000 6296.016
[11] 6375.581 6387.646 6379.983 6337.451 6174.027
[16] 6116.847 6202.424 6272.132 6280.284 6248.125
[21] 6209.052 6264.921 6255.626 6284.288 6294.714
[26] 6293.406 6250.933 6236.943 6253.536 6212.232
[31] 6276.089 6279.387 6287.019 6326.227 6288.432
[36] 6259.360 6267.538 6304.584 6306.815 6324.002
[41] 6334.530 6379.750 6340.025 6332.644 6217.281
[46] 6234.496 6274.475 6242.280 6229.312 6204.011
[51] 6135.444 6144.218 6228.168 6194.706 6166.934
[56] 6136.068 6053.256 6036.348 6059.082 5998.402
[61] 6037.430 6026.981 6021.469 6103.622 6124.704
[66] 6155.989 6167.419 6178.837 6189.773 6196.810
[71] 6223.323 6255.630 6337.473 6250.169 6263.210
[76] 6278.962 6293.573 6226.141 6205.016 6178.169
[81] 6261.977 6215.369 6163.449 6175.810 6146.565
[86] 6178.817 6140.326 6096.775 6126.170 6120.450
[91] 6149.915 6152.934 6115.189 6098.067 6068.587
[96] 6024.013 6020.864 5950.885 6009.655 6127.880
[101] 6131.721 6110.704 6149.942 6184.693 6191.616
[106] 6181.330 6177.924 6137.222 6195.143 6209.417
[111] 6242.177 6285.075 6247.755 6282.197 6303.735
[116] 6317.268 6327.139 6297.364 6281.406 6321.291
[121] 6255.228 6277.171 6223.511 6272.318 6272.766
[126] 6294.392 6323.231 6281.190 6283.873 6289.482
[131] 6242.868 6235.978 6231.278 6247.035 6241.934
[136] 6131.033 6109.009 6110.870 6055.421 5937.873
[141] 5881.995 5920.164 5976.335 5984.970 5997.432
[146] 5949.908 5952.222 5910.906 5869.779 5864.770
[151] 5865.348 5884.787 5926.616 5940.312 5880.080
[156] 5804.875 5784.963 5686.746 5533.519 5450.529
> #pembuatan tabel perbandingan train dan testing
> perbandingan12 <- data.frame("Aktual"= testing.ts,
+ "Ramalan" = hasilgarch12[-1])
> head(perbandingan12,10)
Aktual Ramalan
1 6394.609 6390.476
2 6403.294 6402.881
3 6456.539 6452.777
4 6433.547 6432.800
5 6403.810 6400.349
6 6384.987 6383.969
7 6401.365 6398.148
8 6325.237 6324.000
9 6299.035 6296.016
10 6376.996 6375.581
> str(perbandingan12)
'data.frame': 159 obs. of 2 variables:

```

```

$ Aktual : Time-Series from 369 to 527: 6395 6403 6457 6434 6404 ...
$ Ramalan: num 6390 6403 6453 6433 6400 ...
> # Menulis data perbandingan ke file CSV
> write.csv(perbandingan12, file = "C:/Users/asus/Dropbox/Penelitian UII/Analisis Data/perbandingan Sebelum.csv", row.names = FALSE)
> #MAPE PERBANDINGAN 12
> Date<-c(datasaham$Date[369:527])
> dataframe <- data.frame(Date, perbandingan12)
> T <- nrow(dataframe)
> MAPE <- 1/T*sum(abs((dataframe$Aktual-dataframe$Ramalan)/dataframe$Aktual)*100)
> MAPE
[1] 0.03534329
> coef(model2)
          ar1          ma1
-0.9067115  0.9366584
> # Menghitung Error (Selisih Aktual dan Ramalan)
> error <- perbandingan12$Aktual - perbandingan12$Ramalan
> # Menghitung Mean Squared Error (MSE)
> MSE <- mean(error^2)
> # Menghitung Root Mean Squared Error (RMSE)
> RMSE <- sqrt(MSE)
> # Menampilkan hasil
> cat("Mean Squared Error (MSE):", MSE, "\n")
Mean Squared Error (MSE): 4.886652
> cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
Root Mean Squared Error (RMSE): 2.210577
> # Mengonversi kedua kolom ke dalam format time series
> # Mengonversi perbandingan12[,1] (data aktual) ke time series
> ts_aktual <- ts(perbandingan12[, 1], frequency = 1) # Gantilah frequency sesuai data (1 untuk data harian)
> # Mengonversi perbandingan12[,2] (hasil forecast) ke time series
> ts_forecast <- ts(perbandingan12[, 2], frequency = 1) # Gantilah frequency sesuai data
> # Membuat plot perbandingan antara data aktual dan hasil forecast
> plot(ts_aktual, type = "l", col = "yellow", lty = 1, # Garis solid untuk data aktual
+       xlab = "Observasi", ylab = "Nilai",
+       main = "Perbandingan Data Aktual dan Hasil Forecast",
+       lwd = 4) # Rentang Y agar kedua data terlihat
> # Menambahkan garis untuk hasil forecast
> lines(ts_forecast, col = "blue", lty = 3, lwd = 2) # Garis putus-putus untuk hasil forecast
> # Menambahkan legend untuk membedakan data
> legend("bottomleft", legend = c("Data Aktual", "Hasil Forecast"),
+       col = c("yellow", "blue"), lty = c(1, 2), lwd = 2)

```

Lampiran 3. Sintaks R Model ARIMA GARCH Masa Covid-19

```
> library(ggplot2)
> library(tsibble)
> library(tseries)
> library(MASS)
> library(dplyr)
> library(TTR)
> library(forecast)
> library(lmtest)
> library(HoRM)
> library(dLagM)
> library(dynlm)
> library(MLmetrics)
> library(car)
> library(TSA)
> library(aTSA)
> library(rugarch)
> library(imputeTS)
> library(tidyr)
> # Membaca data
> setwd("C:/Users/asus/Dropbox/Penelitian UII/Analisis Data")
> datasaham <- read.csv("Masa Covid.csv")
> # Menghapus kolom yang tidak diperlukan (sesuaikan dengan struktur data Anda)
> datasaham <- datasaham[,-c(2,3,4,6,7)]
> # Konversi kolom Date ke format waktu
> datasaham$Date <- as.Date(datasaham$Date, format="%m/%d/%Y")
> datasaham$Date <- as.POSIXct(datasaham$Date)
> # Konversi data menjadi time series
> ts.datasaham <- ts(datasaham$Close)
> # Menampilkan subset data dengan tabel yang lebih rapi
> kableExtra::kable(head(datasaham), caption = 'Subset Data Harian IHSG')
```

Table: Subset Data Harian IHSG

Date	Close
2020-03-02	5361.246
2020-03-03	5518.628
2020-03-04	5650.136
2020-03-05	5638.130
2020-03-06	5498.540
2020-03-09	5136.809

```
> # Melihat struktur dan ringkasan data
> str(datasaham)
'data.frame': 693 obs. of 2 variables:
 $ Date : POSIXct, format: "2020-03-02" ...
 $ Close: num 5361 5519 5650 5638 5499 ...
> summary(datasaham)
      Date      Close
Min.   :2020-03-02 00:00:00.00 Min.   :3938
1st Qu.:2020-11-19 00:00:00.00 1st Qu.:5612
Median :2021-08-09 00:00:00.00 Median :6228
Mean    :2021-08-05 19:46:29.60 Mean    :6157
3rd Qu.:2022-04-19 00:00:00.00 3rd Qu.:6850
Max.    :2022-12-30 00:00:00.00 Max.    :7318
> # Konversi kolom Date ke format Date
```



```

> datasaham$Date <- as.Date(datasaham$Date)
> # Konversi ke time series (menggunakan hargautupan saham)
> ts.datasaham <- ts(datasaham$Close)
> # Cek kembali struktur setelah perubahan
> str(datasaham)
'data.frame': 693 obs. of 2 variables:
 $ Date : Date, format: "2020-03-02" ...
 $ Close: num 5361 5519 5650 5638 5499 ...
> # Membuat Boxplot Harga Tutupan Saham
> y <- datasaham$Date
> x <- datasaham$Close
> boxplot(x, xlab="Harga Tutupan IHSG", col="lightblue")
> # Plot harga harian IHSG datasaham Covid-19
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line(color = "blue") +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020",
+         x = "Tanggal",
+         y = "Harga IHSG") +
+   theme(plot.title = element_text(face = "bold", hjust = 0.5),
+         plot.subtitle = element_text(hjust = 0.5),
+         panel.background = element_rect(fill = NA))
> # Menentukan proporsi data pelatihan (70%)
> p = 0.7
> # Menghitung jumlah data untuk pelatihan
> jumlah.train <- as.integer(p * nrow(datasaham))
> # Membuat plot
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line() +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020") +
+   geom_vline(aes(xintercept = Date[jumlah.train],
+                 col="Pemisah_data_train"), lty=2, lwd=.7) +
+   theme(plot.title = element_text(face = "bold", hjust = .5),
+         plot.subtitle = element_text(hjust = .5),
+         legend.position = "bottom",
+         panel.background = element_rect(fill=NA)) +
+   scale_color_manual(name = "", values = c(Pemisah_data_train = "blue"))
> # Menentukan data pelatihan
> datasaham[jumlah.train,1]
[1] "2022-02-24"
> train = 1:jumlah.train
> # Membuat time series untuk data pelatihan dan pengujian
> train.ts <- ts(datasaham$Close[train])
> testing.ts <- ts(datasaham$Close[-train], start = jumlah.train + 1)
> # Menentukan jumlah data pelatihan dan pengujian
> jumlah.train <- length(train.ts)
> jumlah.testing <- length(testing.ts)
> # Menampilkan hasil
> cat("Jumlah data pelatihan: ", jumlah.train, "\n")
Jumlah data pelatihan: 485
> cat("Jumlah data pengujian: ", jumlah.testing, "\n")
Jumlah data pengujian: 208
> #Identifikasi Stasioneritas = differencing
> library(tseries)
> par(mfrow=c(1,2))
> acf(train.ts, main = "ACF Plot for MA(q)")
> pacf(train.ts, main = "PACF Plot for AR(p)")
> adf.test(train.ts)
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
lag ADF p.value

```

```
[1,] 0 0.872 0.895
[2,] 1 0.765 0.864
[3,] 2 0.789 0.871
[4,] 3 0.586 0.812
[5,] 4 0.672 0.837
[6,] 5 0.817 0.879
```

Type 2: with drift no trend

```
lag ADF p.value
[1,] 0 -0.680 0.815
[2,] 1 -0.639 0.829
[3,] 2 -0.472 0.888
[4,] 3 -0.747 0.791
[5,] 4 -0.800 0.772
[6,] 5 -1.165 0.643
```

Type 3: with drift and trend

```
lag ADF p.value
[1,] 0 -3.20 0.0876
[2,] 1 -3.53 0.0393
[3,] 2 -3.66 0.0260
[4,] 3 -4.11 0.0100
[5,] 4 -3.84 0.0167
[6,] 5 -3.54 0.0377
```

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> library(forecast)
> train.diff<-diff(train.ts, differences = 1)
> par(mfrow=c(1,2))
> acf(train.diff, main = "ACF Plot for MA(q)")
> pacf(train.diff, main = "PACF Plot for AR(p)")
> adf.test(train.diff)
```

Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend

```
lag ADF p.value
[1,] 0 -21.74 0.01
[2,] 1 -16.85 0.01
[3,] 2 -11.26 0.01
[4,] 3 -9.85 0.01
[5,] 4 -8.14 0.01
[6,] 5 -7.53 0.01
```

Type 2: with drift no trend

```
lag ADF p.value
[1,] 0 -21.75 0.01
[2,] 1 -16.86 0.01
[3,] 2 -11.28 0.01
[4,] 3 -9.87 0.01
[5,] 4 -8.19 0.01
[6,] 5 -7.58 0.01
```

Type 3: with drift and trend

```
lag ADF p.value
[1,] 0 -21.76 0.01
[2,] 1 -16.90 0.01
[3,] 2 -11.31 0.01
[4,] 3 -9.89 0.01
[5,] 4 -8.16 0.01
[6,] 5 -7.56 0.01
```

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> #Identifikasi Model ARIMA dengan ACF, PACF, dan EACF
> acf(train.diff, main="ACF IHSG datasaham Covid-19")
> pacf(train.diff, main="PACF IHSG datasaham Covid-19")
> library(TSA)
> eacf(train.diff)
```

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o o x o o o o o x o o o x o
1 x o x o o o o o o o o o x o
2 x x x o o o o o o o o o x o
3 o x x o x o o o x o o o x o
4 o x x x o o o o o o o o o
5 o x x x o o o o o o o o o
6 o x o x x x o o o o o o o
7 x x o x x x x o o o o o o
```

```
> #penentuan model ARIMA
> library(forecast)
> model_arima <- auto.arima(train.diff, trace = T)
```

Fitting models using approximations to speed things up...

```
ARIMA(2,0,2) with non-zero mean : 5464.454
ARIMA(0,0,0) with non-zero mean : 5474.831
ARIMA(1,0,0) with non-zero mean : 5472.741
ARIMA(0,0,1) with non-zero mean : 5476.786
ARIMA(0,0,0) with zero mean      : 5473.737
ARIMA(1,0,2) with non-zero mean : 5472.271
ARIMA(2,0,1) with non-zero mean : 5462.415
ARIMA(1,0,1) with non-zero mean : 5473.311
ARIMA(2,0,0) with non-zero mean : 5469.243
ARIMA(3,0,1) with non-zero mean : 5460.385
ARIMA(3,0,0) with non-zero mean : 5458.564
ARIMA(4,0,0) with non-zero mean : 5456.002
ARIMA(5,0,0) with non-zero mean : 5418.134
ARIMA(5,0,1) with non-zero mean : 5419.571
ARIMA(4,0,1) with non-zero mean : 5422.775
ARIMA(5,0,0) with zero mean      : 5417.021
ARIMA(4,0,0) with zero mean      : 5454.559
ARIMA(5,0,1) with zero mean      : 5418.406
ARIMA(4,0,1) with zero mean      : 5426.191
```

Now re-fitting the best model(s) without approximations...

```
ARIMA(5,0,0) with zero mean      : 5461.631
```

Best model: ARIMA(5,0,0) with zero mean

```
> summary(model_arima)
Series: train.diff
ARIMA(5,0,0) with zero mean
```

```
Coefficients:
      ar1      ar2      ar3      ar4      ar5
    0.0239 -0.1017  0.1779 -0.0016  0.1106
s.e.  0.0456  0.0460  0.0451  0.0459  0.0475
```

```
sigma^2 = 4589: log likelihood = -2724.73
AIC=5461.46  AICC=5461.63  BIC=5486.55
```

Training set error measures:

```
Training set 2.344167 67.39004 48.1729 892.433 1002.819 0.6780353
```

```
ACF1
Training set 0.00244405
```

```
> model1 <- Arima(train.diff,order = c(5,1,0),method ="ML")
> summary(model1)
Series: train.diff
ARIMA(5,1,0)
```

Coefficients:

	ar1	ar2	ar3	ar4	ar5
	-0.8589	-0.8367	-0.5204	-0.3770	-0.1281
s.e.	0.0458	0.0596	0.0681	0.0608	0.0474

sigma^2 = 5137: log likelihood = -2746.92
 AIC=5505.83 AICc=5506.01 BIC=5530.91

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.9250162	71.22999	50.44341	1093.923	1454.161	0.7099928

ACF1
 Training set 0.006370586

> model2 <- Arima(train.diff,order = c(4,1,0),method ="ML")

> summary(model2)

Series: train.diff

ARIMA(4,1,0)

Coefficients:

	ar1	ar2	ar3	ar4
	-0.8287	-0.7858	-0.4200	-0.2688
s.e.	0.0447	0.0570	0.0575	0.0461

sigma^2 = 5205: log likelihood = -2750.53
 AIC=5511.07 AICc=5511.19 BIC=5531.97

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.9225229	71.77147	50.84445	1317.311	1747.324	0.7156374

ACF1
 Training set -0.01497064

> model3 <- Arima(train.diff,order = c(3,1,0),method ="ML")

> summary(model3)

Series: train.diff

ARIMA(3,1,0)

Coefficients:

	ar1	ar2	ar3
	-0.767	-0.6097	-0.2022
s.e.	0.045	0.0500	0.0452

sigma^2 = 5562: log likelihood = -2766.91
 AIC=5541.82 AICc=5541.9 BIC=5558.54

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.8858616	74.27105	52.73637	958.8237	1093.078	0.7422663

ACF1
 Training set -0.04379001

> model4 <- Arima(train.diff,order = c(2,1,0),method ="ML")

> summary(model4)

Series: train.diff

ARIMA(2,1,0)

Coefficients:

	ar1	ar2
	-0.6704	-0.4725
s.e.	0.0402	0.0403

sigma^2 = 5782: log likelihood = -2776.7
 AIC=5559.41 AICc=5559.46 BIC=5571.95

Training set error measures:

ME	RMSE	MAE	MPE	MAPE	MASE

```

Training set -0.806208 75.8026 54.81336 2273.273 2426.822 0.7715
              ACF1
Training set -0.09092571
> model5 <- Arima(train.diff,order = c(1,1,0),method ="ML")
> summary(model5)
Series: train.diff
ARIMA(1,1,0)

Coefficients:
      ar1
      -0.4555
s.e.    0.0406

sigma^2 = 7419: log likelihood = -2837.17
AIC=5678.33  AICc=5678.36  BIC=5686.69

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set -0.6236782 85.95619 61.62867 1175.185 1842.317 0.8674257
              ACF1
Training set -0.2142254
> model6 <- Arima(train.diff,order = c(0,1,0),method ="ML")
> summary(model6)
Series: train.diff
ARIMA(0,1,0)

sigma^2 = 9338: log likelihood = -2893.11
AIC=5788.21  AICc=5788.22  BIC=5792.39

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set -0.536076 96.53424 70.90131 2167.309 3668.851 0.9979385
              ACF1
Training set -0.4538318
> model7 <- Arima(train.diff,order = c(5,1,1),method ="ML")
> summary(model7)
Series: train.diff
ARIMA(5,1,1)

Coefficients:
      ar1      ar2      ar3      ar4      ar5      ma1
s.e.    0.0248  -0.1005  0.1792  -0.0006  0.1112  -1.0000
s.e.    0.0457  0.0461  0.0453  0.0460  0.0476  0.0064

sigma^2 = 4603: log likelihood = -2722.16
AIC=5458.33  AICc=5458.56  BIC=5487.59

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.134603 67.34996 47.97893 776.8336 926.878 0.6753051
              ACF1
Training set 0.01014054
> model8 <- Arima(train.diff,order = c(4,1,1),method ="ML")
> summary(model8)
Series: train.diff
ARIMA(4,1,1)

Coefficients:
      ar1      ar2      ar3      ar4      ma1
s.e.    0.0279  -0.0789  0.1708  -0.0016  -1.0000
s.e.    0.0460  0.0454  0.0453  0.0462  0.0067

sigma^2 = 4643: log likelihood = -2724.87
AIC=5461.74  AICc=5461.92  BIC=5486.82

```

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.51941	67.71719	48.26059	603.947	649.4292	0.6792695

ACF1

Training set 0.00618245

```
> model9 <- Arima(train.diff,order = c(3,1,1),method ="ML")
```

```
> summary(model9)
```

Series: train.diff

ARIMA(3,1,1)

Coefficients:

	ar1	ar2	ar3	ma1
	0.0277	-0.0787	0.1708	-1.0000
s.e.	0.0452	0.0452	0.0453	0.0067

sigma^2 = 4634: log likelihood = -2724.87

AIC=5459.75 AICc=5459.87 BIC=5480.65

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.506675	67.71783	48.25466	596.6786	642.0995	0.679186

ACF1

Training set 0.006236635

```
> model10 <- Arima(train.diff,order = c(2,1,1),method ="ML")
```

```
> summary(model10)
```

Series: train.diff

ARIMA(2,1,1)

Coefficients:

	ar1	ar2	ma1
	0.0135	-0.0778	-1.0000
s.e.	0.0457	0.0459	0.0088

sigma^2 = 4757: log likelihood = -2731.86

AIC=5471.73 AICc=5471.81 BIC=5488.45

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	5.099219	68.68424	48.5987	-51.03191	297.0707	0.6840283

ACF1

Training set 0.01587825

```
> model11 <- Arima(train.diff,order = c(1,1,1),method ="ML")
```

```
> summary(model11)
```

Series: train.diff

ARIMA(1,1,1)

Coefficients:

	ar1	ma1
	0.0133	-1.0000
s.e.	0.0458	0.0076

sigma^2 = 4777: log likelihood = -2733.3

AIC=5472.6 AICc=5472.65 BIC=5485.14

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	4.438732	68.90037	48.76709	-29.82009	265.3732	0.6863985

ACF1

Training set 0.004551961

```
> model12 <- Arima(train.diff,order = c(0,1,1),method ="ML")
```

```
> summary(model12)
```

Series: train.diff

ARIMA(0,1,1)

Coefficients:

```
      ma1
      -1.0000
s.e.    0.0077
```

sigma^2 = 4768: log likelihood = -2733.34

AIC=5470.68 AICc=5470.71 BIC=5479.04

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	4.528334	68.90447	48.71049	-61.07997	295.6657	0.6856018

ACF1

Training set 0.01671657

```
> model13 <- Arima(train.diff,order = c(5,1,2),method = "ML")
```

```
> summary(model13)
```

Series: train.diff

ARIMA(5,1,2)

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ma1	ma2
	0.0461	-0.1011	0.1809	-0.0042	0.1112	-1.0215	0.0215
s.e.	0.3210	0.0469	0.0517	0.0713	0.0476	0.3212	0.3211

sigma^2 = 4612: log likelihood = -2722.16

AIC=5460.32 AICc=5460.63 BIC=5493.76

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.125867	67.34996	47.97823	774.1621	923.5292	0.6752953

ACF1

Training set 0.01045013

```
> model14 <- Arima(train.diff,order = c(4,1,2),method = "ML")
```

```
> summary(model14)
```

Series: train.diff

ARIMA(4,1,2)

Coefficients:

	ar1	ar2	ar3	ar4	ma1	ma2
	-0.3873	-0.0696	0.1380	0.0394	-0.5787	-0.4212
s.e.	0.4113	0.0502	0.0586	0.0787	0.4103	0.4102

sigma^2 = 4647: log likelihood = -2724.57

AIC=5463.14 AICc=5463.37 BIC=5492.4

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.702718	67.67196	48.3463	796.3845	840.3155	0.6804759

ACF1

Training set 0.002354229

```
> model15 <- Arima(train.diff,order = c(3,1,2),method = "ML")
```

```
> summary(model15)
```

Series: train.diff

ARIMA(3,1,2)

Coefficients:

	ar1	ar2	ar3	ma1	ma2
	-0.2354	-0.0751	0.145	-0.7295	-0.2705
s.e.	0.3620	0.0470	0.064	0.3679	0.3679

sigma^2 = 4641: log likelihood = -2724.79

AIC=5461.59 AICc=5461.76 BIC=5486.67

Training set error measures:

```

      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.841016 67.70056 48.40331 816.197 859.7991 0.6812783
      ACF1
Training set 0.002270077
> model16 <- Arima(train.diff,order = c(2,1,2),method ="ML")
> summary(model16)
Series: train.diff
ARIMA(2,1,2)

Coefficients:
      ar1      ar2      ma1      ma2
    -0.5933  -0.1154  -0.3781  -0.6219
s.e.   0.1183   0.0480   0.1120   0.1117

sigma^2 = 4681: log likelihood = -2727.48
AIC=5464.96  AICc=5465.08  BIC=5485.86

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 5.050396 68.06113 48.39911 601.5752 693.9557 0.6812191
      ACF1
Training set 0.01195534
> model17 <- Arima(train.diff,order = c(1,1,2),method ="ML")
> summary(model17)
Series: train.diff
ARIMA(1,1,2)

Coefficients:
      ar1      ma1      ma2
    -0.6431  -0.2688  -0.7312
s.e.   0.1157   0.1007   0.1006

sigma^2 = 4725: log likelihood = -2730.13
AIC=5468.26  AICc=5468.35  BIC=5484.98

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 4.267052 68.45426 48.99265 548.3943 714.2369 0.6895733
      ACF1
Training set -0.05433955
> model18 <- Arima(train.diff,order = c(0,1,2),method ="ML")
> summary(model18)
Series: train.diff
ARIMA(0,1,2)

Coefficients:
      ma1      ma2
    -0.9842  -0.0158
s.e.   0.0509   0.0503

sigma^2 = 4777: log likelihood = -2733.29
AIC=5472.58  AICc=5472.63  BIC=5485.12

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 4.423617 68.89958 48.77797 -23.87075 259.7866 0.6865517
      ACF1
Training set 0.002252418
> #uji residual
> sisaan <- model7$residuals
> checkresiduals(sisaan)

```

Ljung-Box test


```
data: Residuals
Q* = 9.4046, df = 10, p-value = 0.4942
```

```
Model df: 0.    Total lags used: 10
```

```
> tsdiag(model2)
> par(mfrow = c(2,2))
> qqnorm(sisaan)
> qqline(sisaan, col = "blue", lwd = 2)
> plot(sisaan, type="o",
+       ylab = "Sisaan", xlab = "Order", main = "Sisaan Model 7 vs Order")
> abline(h = 0, col='red')
> acf(sisaan)
> pacf(sisaan)
> adf.test(sisaan)
```

```
Augmented Dickey-Fuller Test
alternative: stationary
```

```
Type 1: no drift no trend
```

	lag	ADF	p.value
[1,]	0	-21.62	0.01
[2,]	1	-14.86	0.01
[3,]	2	-12.00	0.01
[4,]	3	-10.31	0.01
[5,]	4	-10.22	0.01
[6,]	5	-8.55	0.01

```
Type 2: with drift no trend
```

	lag	ADF	p.value
[1,]	0	-21.65	0.01
[2,]	1	-14.89	0.01
[3,]	2	-12.04	0.01
[4,]	3	-10.38	0.01
[5,]	4	-10.34	0.01
[6,]	5	-8.67	0.01

```
Type 3: with drift and trend
```

	lag	ADF	p.value
[1,]	0	-21.65	0.01
[2,]	1	-14.91	0.01
[3,]	2	-12.07	0.01
[4,]	3	-10.43	0.01
[5,]	4	-10.46	0.01
[6,]	5	-8.79	0.01

```
-----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
> jarque.bera.test(sisaan)
```

```
Jarque Bera Test
```

```
data: sisaan
X-squared = 739.63, df = 2, p-value < 2.2e-16
```

```
> Box.test(sisaan, type = "Ljung")
```

```
Box-Ljung test
```

```
data: sisaan
X-squared = 0.050079, df = 1, p-value = 0.8229
```

```
> t.test(sisaan, mu = 0, conf.level = 0.95)
```

```
One Sample t-test
```

```
data: sisaan
t = 1.024, df = 483, p-value = 0.3064
```

```

alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -2.880328  9.149534
sample estimates:
mean of x
 3.134603

```

```

> #White Test
> white_test <- bptest(sisaan ~ fitted(model7) + I(fitted(model7)^2))
> print(white_test)

```

studentized Breusch-Pagan test

```

data:  sisaan ~ fitted(model7) + I(fitted(model7)^2)
BP = 24.87, df = 2, p-value = 3.977e-06

```

```

> #uji box cox penanganan heterogenitas variansi, dilakukan transformasi kuadr
atik
> par(mfrow=c(1,1))
> train.ts.new <- (train.ts)^2
> index <- seq(1:485)
> bc = boxcox(train.ts.new~index, lambda = seq(-2, 4, by=.01))
> #Nilai Rounded Lambda
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1.68
> #SK
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
[1] 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49 1.50 1.51 1.52 1.53 1.54
[14] 1.55 1.56 1.57 1.58 1.59 1.60 1.61 1.62 1.63 1.64 1.65 1.66 1.67
[27] 1.68 1.69 1.70 1.71 1.72 1.73 1.74 1.75 1.76 1.77 1.78 1.79 1.80
[40] 1.81 1.82 1.83 1.84 1.85 1.86 1.87 1.88 1.89 1.90 1.91 1.92 1.93
[53] 1.94
> plot.ts(train.ts.new, lty=1, xlab="waktu", ylab="Data Transformasi", main="P
lot Transformasi Data")
> #eksekusi peramalan data train (70%)
> ramalan.da <- forecast::forecast(model2, h = 208)
> data.ramalan.da <- ramalan.da$mean
> plot(ramalan.da)
> summary(ramalan.da)

```

Forecast method: ARIMA(4,1,0)

```

Model Information:
Series: train.diff
ARIMA(4,1,0)

```

```

Coefficients:
      ar1      ar2      ar3      ar4
-0.8287 -0.7858 -0.4200 -0.2688
s.e.    0.0447  0.0570  0.0575  0.0461

```

```

sigma^2 = 5205: log likelihood = -2750.53
AIC=5511.07 AICc=5511.19 BIC=5531.97

```

Error measures:

```

              ME      RMSE      MAE      MPE      MAPE      MASE
Training set -0.9225229 71.77147 50.84445 1317.311 1747.324 0.7156374
              ACF1
Training set -0.01497064

```

Forecasts:

```

Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
486      -12.96102 -105.41867  79.49664 -154.3628 128.4408

```

487	11.16420	-82.63963	104.96802	-132.2964	154.6248
488	-38.27561	-132.31696	55.76574	-182.0995	105.5482
489	-10.67055	-111.24069	89.89959	-164.4793	143.1382
490	-28.82678	-132.74487	75.09130	-187.7558	130.1022
491	-21.19204	-131.07169	88.68760	-189.2385	146.8544
492	-11.55698	-124.36643	101.25248	-184.0842	160.9702
493	-25.33568	-140.39923	89.72786	-201.3102	150.6388
494	-19.81459	-139.13189	99.50270	-202.2946	162.6655
495	-19.66157	-142.25589	102.93275	-207.1534	167.8303
496	-20.92959	-146.66446	104.80529	-213.2245	171.3653
497	-18.61426	-147.39322	110.16470	-215.5647	178.3362
498	-21.08494	-152.69100	110.52112	-222.3590	180.1892
499	-20.36538	-155.08622	114.35547	-226.4031	185.6724
500	-19.65183	-157.26596	117.96231	-230.1145	190.8108
501	-20.39325	-160.76440	119.97790	-235.0724	194.2859
502	-19.97765	-163.12718	123.17188	-238.9060	198.9507
503	-20.23256	-166.07781	125.61269	-243.2836	202.8185
504	-20.22829	-168.75557	128.29899	-247.3812	206.9246
505	-20.00678	-171.15184	131.13828	-251.1632	211.1496
506	-20.19835	-173.89660	133.49989	-255.2595	214.8628
507	-20.14693	-176.37901	136.08515	-259.0833	218.7894
508	-20.13319	-178.85344	138.58706	-262.8749	222.6085
509	-20.16406	-181.33350	141.00537	-266.6514	226.3233
510	-20.11938	-183.70186	143.46310	-270.2972	230.0584
511	-20.15174	-186.10863	145.80514	-273.9609	233.6574
512	-20.15076	-188.45262	148.15110	-277.5462	237.2447
513	-20.13661	-190.75055	150.47732	-281.0681	240.7949
514	-20.14753	-193.04136	152.74631	-284.5658	244.2708
515	-20.14131	-195.28618	155.00356	-288.0023	247.7197
516	-20.14409	-197.51097	157.22278	-291.4033	251.1151
517	-20.14589	-199.70771	159.41593	-294.7620	254.4702
518	-20.14189	-201.87215	161.58836	-298.0743	257.7905
519	-20.14430	-204.01706	163.72847	-301.3534	261.0648
520	-20.14394	-206.13478	165.84689	-304.5924	264.3045
521	-20.14354	-208.22856	167.94147	-307.7948	267.5077
522	-20.14422	-210.30035	170.01191	-310.9629	270.6745
523	-20.14348	-212.34843	172.06148	-314.0956	273.8086
524	-20.14382	-214.37593	174.08828	-317.1962	276.9085
525	-20.14394	-216.38230	176.09441	-320.2646	279.9767
526	-20.14370	-218.36801	178.08061	-323.3016	283.0142
527	-20.14386	-220.33441	180.04668	-326.3089	286.0212
528	-20.14377	-222.28144	181.99389	-329.2867	288.9991
529	-20.14379	-224.20999	183.92241	-332.2361	291.9485
530	-20.14384	-226.12053	185.83284	-335.1580	294.8703
531	-20.14378	-228.01339	187.72583	-338.0529	297.7654
532	-20.14381	-229.88926	189.60165	-340.9218	300.6342
533	-20.14381	-231.74848	191.46086	-343.7652	303.4776
534	-20.14380	-233.59149	193.30389	-346.5839	306.2963
535	-20.14381	-235.41874	195.13112	-349.3784	309.0908
536	-20.14380	-237.23059	196.94299	-352.1494	311.8618
537	-20.14380	-239.02746	198.73986	-354.8975	314.6099
538	-20.14381	-240.80970	200.52209	-357.6232	317.3356
539	-20.14380	-242.57765	202.29004	-360.3270	320.0394
540	-20.14380	-244.33166	204.04405	-363.0095	322.7219
541	-20.14380	-246.07206	205.78445	-365.6713	325.3836
542	-20.14380	-247.79915	207.51154	-368.3126	328.0250
543	-20.14380	-249.51324	209.22563	-370.9341	330.6465
544	-20.14380	-251.21461	210.92700	-373.5361	333.2485
545	-20.14380	-252.90354	212.61594	-376.1191	335.8315
546	-20.14380	-254.58031	214.29271	-378.6835	338.3959
547	-20.14380	-256.24517	215.95757	-381.2297	340.9421
548	-20.14380	-257.89838	217.61077	-383.7580	343.4704
549	-20.14380	-259.54016	219.25256	-386.2689	345.9813
550	-20.14380	-261.17077	220.88316	-388.7627	348.4751

551	-20.14380	-262.79041	222.50281	-391.2398	350.9522
552	-20.14380	-264.39932	224.11171	-393.7004	353.4128
553	-20.14380	-265.99770	225.71009	-396.1449	355.8573
554	-20.14380	-267.58575	227.29814	-398.5736	358.2860
555	-20.14380	-269.16368	228.87607	-400.9868	360.6992
556	-20.14380	-270.73167	230.44406	-403.3849	363.0973
557	-20.14380	-272.28991	232.00230	-405.7680	365.4804
558	-20.14380	-273.83858	233.55097	-408.1365	367.8489
559	-20.14380	-275.37785	235.09024	-410.4906	370.2030
560	-20.14380	-276.90789	236.62029	-412.8306	372.5430
561	-20.14380	-278.42888	238.14127	-415.1567	374.8691
562	-20.14380	-279.94095	239.65335	-417.4693	377.1816
563	-20.14380	-281.44428	241.15667	-419.7684	379.4808
564	-20.14380	-282.93901	242.65140	-422.0544	381.7668
565	-20.14380	-284.42528	244.13767	-424.3274	384.0398
566	-20.14380	-285.90324	245.61563	-426.5878	386.3002
567	-20.14380	-287.37303	247.08542	-428.8356	388.5480
568	-20.14380	-288.83478	248.54717	-431.0712	390.7836
569	-20.14380	-290.28862	250.00101	-433.2946	393.0070
570	-20.14380	-291.73467	251.44706	-435.5062	395.2186
571	-20.14380	-293.17307	252.88546	-437.7060	397.4184
572	-20.14380	-294.60393	254.31632	-439.8943	399.6067
573	-20.14380	-296.02737	255.73976	-442.0713	401.7837
574	-20.14380	-297.44350	257.15589	-444.2371	403.9495
575	-20.14380	-298.85243	258.56483	-446.3919	406.1043
576	-20.14380	-300.25428	259.96668	-448.5358	408.2482
577	-20.14380	-301.64915	261.36154	-450.6691	410.3815
578	-20.14380	-303.03714	262.74953	-452.7918	412.5042
579	-20.14380	-304.41835	264.13075	-454.9042	414.6166
580	-20.14380	-305.79289	265.50528	-457.0064	416.7188
581	-20.14380	-307.16084	266.87324	-459.0985	418.8109
582	-20.14380	-308.52231	268.23470	-461.1807	420.8931
583	-20.14380	-309.87737	269.58977	-463.2531	422.9655
584	-20.14380	-311.22613	270.93852	-465.3158	425.0282
585	-20.14380	-312.56867	272.28106	-467.3690	427.0814
586	-20.14380	-313.90507	273.61746	-469.4129	429.1253
587	-20.14380	-315.23542	274.94781	-471.4475	431.1599
588	-20.14380	-316.55980	276.27219	-473.4729	433.1853
589	-20.14380	-317.87829	277.59068	-475.4894	435.2018
590	-20.14380	-319.19096	278.90335	-477.4970	437.2094
591	-20.14380	-320.49790	280.21029	-479.4958	439.2081
592	-20.14380	-321.79917	281.51157	-481.4859	441.1983
593	-20.14380	-323.09486	282.80725	-483.4675	443.1799
594	-20.14380	-324.38503	284.09742	-485.4406	445.1530
595	-20.14380	-325.66975	285.38214	-487.4054	447.1178
596	-20.14380	-326.94909	286.66148	-489.3620	449.0744
597	-20.14380	-328.22312	287.93551	-491.3105	451.0228
598	-20.14380	-329.49190	289.20429	-493.2509	452.9633
599	-20.14380	-330.75549	290.46789	-495.1834	454.8958
600	-20.14380	-332.01397	291.72637	-497.1081	456.8205
601	-20.14380	-333.26739	292.97979	-499.0250	458.7374
602	-20.14380	-334.51582	294.22821	-500.9343	460.6467
603	-20.14380	-335.75930	295.47170	-502.8361	462.5484
604	-20.14380	-336.99791	296.71030	-504.7303	464.4427
605	-20.14380	-338.23169	297.94408	-506.6172	466.3296
606	-20.14380	-339.46070	299.17310	-508.4969	468.2093
607	-20.14380	-340.68501	300.39740	-510.3693	470.0817
608	-20.14380	-341.90465	301.61705	-512.2346	471.9469
609	-20.14380	-343.11969	302.83208	-514.0928	473.8052
610	-20.14380	-344.33018	304.04257	-515.9441	475.6565
611	-20.14380	-345.53616	305.24855	-517.7885	477.5009
612	-20.14380	-346.73769	306.45008	-519.6260	479.3384
613	-20.14380	-347.93481	307.64720	-521.4569	481.1693
614	-20.14380	-349.12758	308.83997	-523.2811	482.9935

615	-20.14380	-350.31604	310.02843	-525.0987	484.8111
616	-20.14380	-351.50023	311.21263	-526.9097	486.6221
617	-20.14380	-352.68021	312.39261	-528.7144	488.4268
618	-20.14380	-353.85602	313.56842	-530.5126	490.2250
619	-20.14380	-355.02770	314.74009	-532.3045	492.0169
620	-20.14380	-356.19530	315.90769	-534.0902	493.8026
621	-20.14380	-357.35885	317.07124	-535.8697	495.5821
622	-20.14380	-358.51840	318.23079	-537.6431	497.3555
623	-20.14380	-359.67399	319.38638	-539.4104	499.1228
624	-20.14380	-360.82566	320.53805	-541.1717	500.8841
625	-20.14380	-361.97345	321.68584	-542.9271	502.6395
626	-20.14380	-363.11740	322.82979	-544.6767	504.3890
627	-20.14380	-364.25754	323.96994	-546.4204	506.1327
628	-20.14380	-365.39392	325.10632	-548.1583	507.8707
629	-20.14380	-366.52658	326.23897	-549.8905	509.6029
630	-20.14380	-367.65554	327.36793	-551.6171	511.3295
631	-20.14380	-368.78084	328.49324	-553.3381	513.0505
632	-20.14380	-369.90253	329.61492	-555.0536	514.7660
633	-20.14380	-371.02063	330.73302	-556.7636	516.4760
634	-20.14380	-372.13518	331.84757	-558.4682	518.1805
635	-20.14380	-373.24621	332.95860	-560.1673	519.8797
636	-20.14380	-374.35375	334.06614	-561.8612	521.5736
637	-20.14380	-375.45784	335.17024	-563.5497	523.2621
638	-20.14380	-376.55852	336.27091	-565.2331	524.9455
639	-20.14380	-377.65580	337.36819	-566.9112	526.6236
640	-20.14380	-378.74973	338.46212	-568.5842	528.2966
641	-20.14380	-379.84032	339.55272	-570.2522	529.9646
642	-20.14380	-380.92763	340.64002	-571.9150	531.6274
643	-20.14380	-382.01166	341.72406	-573.5729	533.2853
644	-20.14380	-383.09246	342.80485	-575.2259	534.9383
645	-20.14380	-384.17005	343.88244	-576.8739	536.5863
646	-20.14380	-385.24446	344.95685	-578.5171	538.2295
647	-20.14380	-386.31571	346.02811	-580.1554	539.8678
648	-20.14380	-387.38385	347.09624	-581.7890	541.5014
649	-20.14380	-388.44888	348.16127	-583.4178	543.1302
650	-20.14380	-389.51084	349.22323	-585.0419	544.7543
651	-20.14380	-390.56976	350.28215	-586.6614	546.3738
652	-20.14380	-391.62566	351.33805	-588.2763	547.9887
653	-20.14380	-392.67857	352.39096	-589.8866	549.5989
654	-20.14380	-393.72850	353.44090	-591.4923	551.2047
655	-20.14380	-394.77550	354.48789	-593.0935	552.8059
656	-20.14380	-395.81958	355.53197	-594.6903	554.4027
657	-20.14380	-396.86077	356.57316	-596.2827	555.9951
658	-20.14380	-397.89908	357.61147	-597.8706	557.5830
659	-20.14380	-398.93455	358.64694	-599.4543	559.1667
660	-20.14380	-399.96720	359.67959	-601.0336	560.7459
661	-20.14380	-400.99704	360.70944	-602.6086	562.3210
662	-20.14380	-402.02411	361.73650	-604.1793	563.8917
663	-20.14380	-403.04843	362.76082	-605.7459	565.4583
664	-20.14380	-404.07001	363.78240	-607.3083	567.0207
665	-20.14380	-405.08888	364.80127	-608.8665	568.5789
666	-20.14380	-406.10506	365.81745	-610.4206	570.1330
667	-20.14380	-407.11857	366.83096	-611.9706	571.6830
668	-20.14380	-408.12944	367.84183	-613.5166	573.2290
669	-20.14380	-409.13767	368.85007	-615.0586	574.7710
670	-20.14380	-410.14330	369.85570	-616.5966	576.3090
671	-20.14380	-411.14635	370.85874	-618.1306	577.8430
672	-20.14380	-412.14683	371.85922	-619.6607	579.3731
673	-20.14380	-413.14476	372.85715	-621.1869	580.8993
674	-20.14380	-414.14016	373.85256	-622.7092	582.4216
675	-20.14380	-415.13306	374.84545	-624.2277	583.9401
676	-20.14380	-416.12346	375.83586	-625.7424	585.4548
677	-20.14380	-417.11140	376.82379	-627.2533	586.9657
678	-20.14380	-418.09688	377.80927	-628.7605	588.4729

```

679      -20.14380 -419.07993 378.79232 -630.2640 589.9763
680      -20.14380 -420.06056 379.77295 -631.7637 591.4761
681      -20.14380 -421.03879 380.75118 -633.2598 592.9722
682      -20.14380 -422.01464 381.72704 -634.7522 594.4646
683      -20.14380 -422.98813 382.70052 -636.2410 595.9534
684      -20.14380 -423.95927 383.67166 -637.7263 597.4387
685      -20.14380 -424.92808 384.64048 -639.2079 598.9203
[ reached 'max' / getOption("max.print") -- omitted 8 rows ]
> #pengembalian ke nilai awal
> pt_1 <- train.ts.new[485] #nilai akhir data latih
> head(pt_1)
[1] 46482667
> train.ts[485]
[1] 6817.82
> hasil.forc.Diff <- data.ramalan.da
> hasil <- diffinv(hasil.forc.Diff, differences = 1) + pt_1
> hasil
Time Series:
Start = 485
End = 693
Frequency = 1
[1] 46482667 46482654 46482665 46482627 46482616 46482588 46482566
[8] 46482555 46482529 46482510 46482490 46482469 46482450 46482429
[15] 46482409 46482389 46482369 46482349 46482329 46482309 46482289
[22] 46482268 46482248 46482228 46482208 46482188 46482168 46482147
[29] 46482127 46482107 46482087 46482067 46482047 46482027 46482006
[36] 46481986 46481966 46481946 46481926 46481906 46481886 46481865
[43] 46481845 46481825 46481805 46481785 46481765 46481745 46481724
[50] 46481704 46481684 46481664 46481644 46481624 46481604 46481583
[57] 46481563 46481543 46481523 46481503 46481483 46481463 46481442
[64] 46481422 46481402 46481382 46481362 46481342 46481322 46481301
[71] 46481281 46481261 46481241 46481221 46481201 46481181 46481160
[78] 46481140 46481120 46481100 46481080 46481060 46481040 46481019
[85] 46480999 46480979 46480959 46480939 46480919 46480899 46480878
[92] 46480858 46480838 46480818 46480798 46480778 46480758 46480737
[99] 46480717 46480697 46480677 46480657 46480637 46480617 46480596
[106] 46480576 46480556 46480536 46480516 46480496 46480476 46480455
[113] 46480435 46480415 46480395 46480375 46480355 46480335 46480314
[120] 46480294 46480274 46480254 46480234 46480214 46480194 46480173
[127] 46480153 46480133 46480113 46480093 46480073 46480053 46480032
[134] 46480012 46479992 46479972 46479952 46479932 46479912 46479891
[141] 46479871 46479851 46479831 46479811 46479791 46479771 46479750
[148] 46479730 46479710 46479690 46479670 46479650 46479630 46479609
[155] 46479589 46479569 46479549 46479529 46479509 46479488 46479468
[162] 46479448 46479428 46479408 46479388 46479368 46479347 46479327
[169] 46479307 46479287 46479267 46479247 46479227 46479206 46479186
[176] 46479166 46479146 46479126 46479106 46479086 46479065 46479045
[183] 46479025 46479005 46478985 46478965 46478945 46478924 46478904
[190] 46478884 46478864 46478844 46478824 46478804 46478783 46478763
[197] 46478743 46478723 46478703 46478683 46478663 46478642 46478622
[204] 46478602 46478582 46478562 46478542 46478522 46478501
> hasil1 <- abs(hasil)^(1/2) #hasil peramalan arima nilai data telah dikembali
kan ke asli
> pt_1 <- c(train.ts.new[485],testing.ts) #nilai akhir data latih
> pt_1
[1] 46482667.153      6888.171      6921.441      6868.402      6928.328
[6] 6869.066      6814.183      6864.441      6924.008      6922.602
[11] 6952.204      6918.185      6992.395      6964.385      6954.965
[16] 6955.181      7000.822      6996.115      7049.685      7002.532
[21] 7049.603      7011.689      7053.190      7071.442      7078.760
[26] 7116.218      7148.299      7104.216      7127.367      7210.835
[31] 7203.794      7214.781      7262.777      7235.532      7275.289
[36] 7199.232      7227.362      7276.193      7225.606      7215.979
[41] 7232.153      7196.764      7228.914      6909.751      6819.794

```

```

[46] 6816.203 6599.840 6597.993 6644.467 6793.414
[51] 6823.335 6918.144 6840.775 6914.141 6883.504
[56] 7026.256 7037.565 7148.970 7148.723 7182.961
[61] 7096.582 7141.045 7193.314 7182.830 7086.648
[66] 6995.442 7049.882 7007.050 7050.326 6936.967
[71] 6976.377 7044.071 6984.311 6998.267 7042.937
[76] 7016.055 6996.456 6942.351 6911.582 6794.328
[81] 6639.172 6703.266 6646.410 6652.587 6740.219
[86] 6722.145 6718.294 6640.990 6690.087 6651.905
[91] 6659.253 6736.092 6874.743 6864.131 6886.962
[96] 6858.407 6871.537 6898.216 6956.817 6951.123
[101] 6968.783 6988.157 7046.635 7057.348 7084.655
[106] 7086.849 7102.880 7086.236 7160.385 7129.277
[111] 7093.276 7133.453 7186.558 7172.434 7107.983
[116] 7163.265 7194.706 7174.208 7135.248 7132.045
[121] 7159.473 7178.590 7153.104 7177.179 7231.883
[126] 7233.155 7186.757 7232.017 7242.656 7254.464
[131] 7318.016 7278.078 7305.601 7168.870 7195.488
[136] 7196.951 7188.314 7218.906 7178.583 7127.503
[141] 7112.449 7077.031 7036.198 7040.798 7009.718
[146] 7072.256 7075.385 7076.623 7026.783 6994.395
[151] 6939.145 6909.209 6880.625 6814.530 6831.115
[156] 6834.494 6860.416 6980.652 7017.771 7053.044
[161] 7048.380 7043.939 7091.761 7056.040 7098.890
[166] 7052.303 7015.690 7034.573 7045.527 7102.396
[171] 7050.126 7070.084 6966.837 7089.206 7019.392
[176] 7035.500 7014.384 7044.986 7082.181 7063.247
[181] 7030.588 7054.123 7080.515 7053.150 7017.358
[186] 7012.069 7081.313 7020.803 7019.639 6987.328
[191] 6892.570 6818.752 6804.227 6715.118 6734.452
[196] 6810.316 6801.749 6751.860 6812.193 6779.698
[201] 6768.316 6820.663 6824.432 6800.673 6835.808
[206] 6923.027 6850.520 6860.077 6850.619

> coef(model2)
      ar1      ar2      ar3      ar4
-0.8287351 -0.7858204 -0.4200140 -0.2687958
> library(TSA)
> model2 <- Arima(train.ts,order = c(1,1,1))
> for (i in 1:15) {
+   ArchTest <- ArchTest(model2$residuals, lags=i, demean=TRUE)
+   cat("P Value LM Test lag ke", i,"adalah" , ArchTest$p.value, "\n") }
P Value LM Test lag ke 1 adalah 7.337454e-06
P Value LM Test lag ke 2 adalah 8.096901e-11
P Value LM Test lag ke 3 adalah 8.670842e-14
P Value LM Test lag ke 4 adalah 4.918288e-14
P Value LM Test lag ke 5 adalah 8.582024e-14
P Value LM Test lag ke 6 adalah 0
P Value LM Test lag ke 7 adalah 0
P Value LM Test lag ke 8 adalah 0
P Value LM Test lag ke 9 adalah 0
P Value LM Test lag ke 10 adalah 0
P Value LM Test lag ke 11 adalah 0
P Value LM Test lag ke 12 adalah 0
P Value LM Test lag ke 13 adalah 0
P Value LM Test lag ke 14 adalah 0
P Value LM Test lag ke 15 adalah 0
> #pengecekan efek GARCH
> train.diff.new <- (train.diff)^2
> sisaan.new <- (sisaan)^2
> garch01 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garchO
rder=c(0,1)))
> garch01

```

```
*-----*
*          GARCH Model Spec          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(0,1)
Variance Targeting : FALSE
```

Conditional Mean Dynamics

```
-----
Mean Model       : ARFIMA(1,0,1)
Include Mean     : FALSE
GARCH-in-Mean    : FALSE
```

Conditional Distribution

```
-----
Distribution      : norm
Includes Skew     : FALSE
Includes Shape    : FALSE
Includes Lambda   : FALSE
```

```
> garch01fit <- rugarch::ugarchfit(data =train.diff, spec = garch01)
> garch01fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(0,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate Std. Error   t value Pr(>|t|)
ar1      0.784516   0.526945   1.4888e+00 0.13654
ma1     -0.780162   0.533821  -1.4615e+00 0.14389
omega    0.000013   0.021275   6.0600e-04 0.99952
beta1    0.998371   0.000015   6.7984e+04 0.00000
```

Robust Standard Errors:

```
      Estimate Std. Error   t value Pr(>|t|)
ar1      0.784516   0.188992   4.1511e+00 0.000033
ma1     -0.780162   0.220838  -3.5327e+00 0.000411
omega    0.000013   0.056747   2.2700e-04 0.999819
beta1    0.998371   0.000022   4.4495e+04 0.000000
```

LogLikelihood : -2715.577

Information Criteria

```
-----
Akaike          11.238
Bayes           11.272
Shibata         11.238
Hannan-Quinn    11.252
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
              statistic    p-value
```



```

Lag[1]                0.1686 6.813e-01
Lag[2*(p+q)+(p+q)-1][5] 9.7588 1.175e-12
Lag[4*(p+q)+(p+q)-1][9] 13.7723 1.602e-04
d.o.f=2
H0 : No serial correlation

```

Weighted Ljung-Box Test on Standardized Squared Residuals

```

-----
                statistic  p-value
Lag[1]                13.27 2.697e-04
Lag[2*(p+q)+(p+q)-1][2] 26.27 1.007e-07
Lag[4*(p+q)+(p+q)-1][5] 52.71 1.055e-14
d.o.f=1

```

Weighted ARCH LM Tests

```

-----
Statistic Shape Scale  P-value
ARCH Lag[2]    25.79 0.500 2.000 3.806e-07
ARCH Lag[4]    45.80 1.397 1.611 1.946e-12
ARCH Lag[6]    62.74 2.222 1.500 1.110e-16

```

Nyblom stability test

```

-----
Joint Statistic: 1.4222
Individual Statistics:
ar1  0.1232
ma1  0.1265
omega 0.1302
beta1 0.1714

```

Asymptotic Critical Values (10% 5% 1%)

```

Joint Statistic: 1.07 1.24 1.6
Individual Statistic: 0.35 0.47 0.75

```

Sign Bias Test

```

-----
                t-value    prob  sig
Sign Bias        1.209 2.273e-01
Negative Sign Bias 4.079 5.301e-05 ***
Positive Sign Bias 2.205 2.790e-02 **
Joint Effect      30.179 1.266e-06 ***

```

Adjusted Pearson Goodness-of-Fit Test:

```

-----
group statistic p-value(g-1)
1    20      32.20    0.029684
2    30      40.75    0.072371
3    40      65.92    0.004509
4    50      62.69    0.090438

```

Elapsed time : 0.1610231

```

> garch11 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sgARCH", garchO
rder=c(1,1)))
> garch11fit <- rugarch::ugarchfit(spec = garch11, data = train.diff)
> garch11fit

```

```

*-----*
*                GARCH Model Fit                *
*-----*

```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm

```

Optimal Parameters

```
-----
      Estimate Std. Error t value Pr(>|t|)
ar1      0.32961   0.278606   1.1831 0.236783
ma1     -0.41418   0.264619  -1.5652 0.117533
omega  429.21296  146.278531   2.9342 0.003344
alpha1   0.17905   0.052439   3.4143 0.000639
beta1    0.71624   0.070445  10.1674 0.000000

```

Robust Standard Errors:

```
      Estimate Std. Error t value Pr(>|t|)
ar1      0.32961   0.166766   1.9765 0.048100
ma1     -0.41418   0.171925  -2.4091 0.015992
omega  429.21296  160.272661   2.6780 0.007406
alpha1   0.17905   0.067117   2.6677 0.007637
beta1    0.71624   0.077539   9.2372 0.000000

```

LogLikelihood : -2672.561

Information Criteria

```
-----
Akaike          11.064
Bayes           11.108
Shibata         11.064
Hannan-Quinn    11.081

```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
              statistic p-value
Lag[1]              0.4226 5.156e-01
Lag[2*(p+q)+(p+q)-1][5] 6.4198 1.359e-05
Lag[4*(p+q)+(p+q)-1][9] 9.3076 1.795e-02
d.o.f=2
H0 : No serial correlation

```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
              statistic p-value
Lag[1]              0.07768 0.7805
Lag[2*(p+q)+(p+q)-1][5] 0.62588 0.9366
Lag[4*(p+q)+(p+q)-1][9] 1.02267 0.9853
d.o.f=2

```

Weighted ARCH LM Tests

```
-----
      Statistic Shape Scale P-value
ARCH Lag[3]    0.6634 0.500 2.000 0.4154
ARCH Lag[5]    0.9048 1.440 1.667 0.7614
ARCH Lag[7]    1.0163 2.315 1.543 0.9109

```

Nyblom stability test

```
-----
Joint Statistic: 0.9794
Individual Statistics:
ar1      0.09156
ma1      0.08188

```

```
omega 0.25785
alpha1 0.60449
beta1 0.45502
```

```
Asymptotic Critical values (10% 5% 1%)
Joint Statistic:      1.28 1.47 1.88
Individual Statistic: 0.35 0.47 0.75
```

Sign Bias Test

```
-----
Sign Bias          t-value   prob sig
Negative Sign Bias 1.3290 0.1845
Positive Sign Bias 1.4839 0.1385
Joint Effect       6.9533 0.0734 *
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1    20      26.66    0.11280
2    30      35.67    0.18339
3    40      54.35    0.05214
4    50      50.92    0.39805
```

Elapsed time : 0.1512761

```
> garch12 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "SGARCH",garchO
rder=c(1,2)))
> garch12fit <- rugarch::ugarchfit(spec = garch12, data = train.diff)
> garch12fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(1,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
Estimate Std. Error t value Pr(>|t|)
ar1      0.33539    0.289746   1.1575 0.247054
ma1      -0.41722    0.275456  -1.5147 0.129857
omega    450.83595  166.059614   2.7149 0.006629
alpha1    0.18543    0.062442   2.9697 0.002981
beta1     0.70531    0.267848   2.6333 0.008457
beta2     0.00000    0.214682   0.0000 1.000000
```

Robust Standard Errors:

```
Estimate Std. Error t value Pr(>|t|)
ar1      0.33539    0.17651   1.9001 0.057417
ma1      -0.41722    0.18111  -2.3037 0.021239
omega    450.83595  209.71607   2.1497 0.031575
alpha1    0.18543    0.09063   2.0461 0.040751
beta1     0.70531    0.28224   2.4990 0.012456
beta2     0.00000    0.19113   0.0000 1.000000
```

LogLikelihood : -2674.46

Information Criteria

```
-----
Akaike      11.076
Bayes       11.128
Shibata     11.076
Hannan-Quinn 11.097
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
                        statistic  p-value
Lag[1]                  0.5015  4.788e-01
Lag[2*(p+q)+(p+q)-1][5] 6.2424  2.873e-05
Lag[4*(p+q)+(p+q)-1][9] 8.9969  2.395e-02
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                        statistic  p-value
Lag[1]                  0.005152  0.9428
Lag[2*(p+q)+(p+q)-1][8] 0.390550  0.9982
Lag[4*(p+q)+(p+q)-1][14] 1.771914  0.9948
d.o.f=3
```

Weighted ARCH LM Tests

```
-----
Statistic Shape Scale P-Value
ARCH Lag[4]    0.3492 0.500 2.000 0.5546
ARCH Lag[6]    0.4047 1.461 1.711 0.9181
ARCH Lag[8]    0.6971 2.368 1.583 0.9634
```

Nyblom stability test

```
-----
Joint Statistic: 1.1273
Individual Statistics:
ar1    0.09914
ma1    0.08872
omega  0.27422
alpha1 0.61330
beta1  0.47050
beta2  0.45740
```

Asymptotic Critical Values (10% 5% 1%)
 Joint Statistic: 1.49 1.68 2.12
 Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

```
-----
                        t-value    prob sig
Sign Bias              0.176 0.86039
Negative Sign Bias     1.409 0.15956
Positive Sign Bias     1.353 0.17657
Joint Effect           6.684 0.08267 *
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1    20      31.21      0.03831
2    30      35.30      0.19494
3    40      59.64      0.01827
```

4 50 62.90 0.08759

Elapsed time : 0.2007141

```
> garch22 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "SGARCH", garchO
rder=c(2,2)))
> garch22fit <- rugarch::ugarchfit(spec = garch22, data = train.diff)
> garch22fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(2,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate Std. Error  t value Pr(>|t|)
ar1      0.331838   0.311721  1.064536 0.287086
ma1     -0.402081   0.300564 -1.337756 0.180976
omega  787.032309 264.244709  2.978422 0.002897
alpha1   0.140352   0.056533  2.482655 0.013041
alpha2   0.186661   0.069023  2.704320 0.006844
beta1    0.000001   0.301645  0.000002 0.999998
beta2    0.481249   0.270388  1.779846 0.075101
```

Robust Standard Errors:

```
      Estimate Std. Error  t value Pr(>|t|)
ar1      0.331838   0.169557  1.957092 0.050337
ma1     -0.402081   0.175023 -2.297303 0.021602
omega  787.032309 301.485314  2.610516 0.009041
alpha1   0.140352   0.084820  1.654713 0.097983
alpha2   0.186661   0.075086  2.485972 0.012920
beta1    0.000001   0.391611  0.000002 0.999999
beta2    0.481249   0.313106  1.537013 0.124290
```

LogLikelihood : -2672.601

Information Criteria

```
-----
Akaike      11.073
Bayes      11.133
Shibata     11.072
Hannan-Quinn 11.096
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
              statistic  p-value
Lag[1]              0.1869 6.655e-01
Lag[2*(p+q)+(p+q)-1][5] 6.0744 5.753e-05
Lag[4*(p+q)+(p+q)-1][9] 8.8897 2.642e-02
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
```

	statistic	p-value
Lag[1]	0.1116	0.7384
Lag[2*(p+q)+(p+q)-1][11]	0.7250	0.9990
Lag[4*(p+q)+(p+q)-1][19]	2.7880	0.9967

d.o.f=4

Weighted ARCH LM Tests

	Statistic	Shape	Scale	P-Value
ARCH Lag[5]	0.01988	0.500	2.000	0.8879
ARCH Lag[7]	0.16539	1.473	1.746	0.9775
ARCH Lag[9]	0.50284	2.402	1.619	0.9838

Nyblom stability test

Joint Statistic: 1.2878
 Individual Statistics:

ar1	0.1612
ma1	0.1500
omega	0.2546
alpha1	0.6009
alpha2	0.4762
beta1	0.4480
beta2	0.4993

Asymptotic Critical values (10% 5% 1%)
 Joint Statistic: 1.69 1.9 2.35
 Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.06908	0.9450	
Negative Sign Bias	1.57913	0.1150	
Positive Sign Bias	1.09288	0.2750	
Joint Effect	6.86891	0.0762	*

Adjusted Pearson Goodness-of-Fit Test:

group	statistic	p-value(g-1)
1 20	25.92	0.1325
2 30	34.93	0.2070
3 40	43.27	0.2939
4 50	48.85	0.4791

Elapsed time : 0.2334521

```
> #eksekusi peramalan dengan efek garch
> par(mfrow=c(1,1))
> forc11<- ugarchforecast(fit0Rspec = garch01fit, n.ahead = 208, n.roll = 0)
> plot(forc01, which= 1)
> forc11
```

```
*-----*
*          GARCH Model Forecast          *
*-----*
```

Model: sGARCH
 Horizon: 208
 Roll Steps: 0
 Out of Sample: 0

0-roll forecast [T0=485-01-01]:

	Series	Sigma
T+1	-1.874e-01	46.48
T+2	-1.470e-01	46.44
T+3	-1.153e-01	46.40
T+4	-9.047e-02	46.37
T+5	-7.098e-02	46.33
T+6	-5.568e-02	46.29
T+7	-4.368e-02	46.25
T+8	-3.427e-02	46.21
T+9	-2.689e-02	46.18
T+10	-2.109e-02	46.14
T+11	-1.655e-02	46.10
T+12	-1.298e-02	46.06
T+13	-1.018e-02	46.03
T+14	-7.990e-03	45.99
T+15	-6.268e-03	45.95
T+16	-4.917e-03	45.91
T+17	-3.858e-03	45.88
T+18	-3.026e-03	45.84
T+19	-2.374e-03	45.80
T+20	-1.863e-03	45.76
T+21	-1.461e-03	45.73
T+22	-1.146e-03	45.69
T+23	-8.994e-04	45.65
T+24	-7.056e-04	45.62
T+25	-5.535e-04	45.58
T+26	-4.343e-04	45.54
T+27	-3.407e-04	45.50
T+28	-2.673e-04	45.47
T+29	-2.097e-04	45.43
T+30	-1.645e-04	45.39
T+31	-1.290e-04	45.36
T+32	-1.012e-04	45.32
T+33	-7.942e-05	45.28
T+34	-6.231e-05	45.25
T+35	-4.888e-05	45.21
T+36	-3.835e-05	45.17
T+37	-3.009e-05	45.14
T+38	-2.360e-05	45.10
T+39	-1.852e-05	45.06
T+40	-1.453e-05	45.02
T+41	-1.140e-05	44.99
T+42	-8.941e-06	44.95
T+43	-7.014e-06	44.91
T+44	-5.503e-06	44.88
T+45	-4.317e-06	44.84
T+46	-3.387e-06	44.81
T+47	-2.657e-06	44.77
T+48	-2.084e-06	44.73
T+49	-1.635e-06	44.70
T+50	-1.283e-06	44.66
T+51	-1.006e-06	44.62
T+52	-7.896e-07	44.59
T+53	-6.194e-07	44.55
T+54	-4.859e-07	44.51
T+55	-3.812e-07	44.48
T+56	-2.991e-07	44.44
T+57	-2.346e-07	44.41
T+58	-1.841e-07	44.37
T+59	-1.444e-07	44.33
T+60	-1.133e-07	44.30
T+61	-8.888e-08	44.26
T+62	-6.973e-08	44.22
T+63	-5.470e-08	44.19

T+64	-4.291e-08	44.15
T+65	-3.367e-08	44.12
T+66	-2.641e-08	44.08
T+67	-2.072e-08	44.04
T+68	-1.626e-08	44.01
T+69	-1.275e-08	43.97
T+70	-1.000e-08	43.94
T+71	-7.849e-09	43.90
T+72	-6.158e-09	43.87
T+73	-4.831e-09	43.83
T+74	-3.790e-09	43.79
T+75	-2.973e-09	43.76
T+76	-2.333e-09	43.72
T+77	-1.830e-09	43.69
T+78	-1.436e-09	43.65
T+79	-1.126e-09	43.62
T+80	-8.835e-10	43.58
T+81	-6.932e-10	43.55
T+82	-5.438e-10	43.51
T+83	-4.266e-10	43.47
T+84	-3.347e-10	43.44
T+85	-2.626e-10	43.40
T+86	-2.060e-10	43.37
T+87	-1.616e-10	43.33
T+88	-1.268e-10	43.30
T+89	-9.946e-11	43.26
T+90	-7.803e-11	43.23
T+91	-6.121e-11	43.19
T+92	-4.802e-11	43.16
T+93	-3.767e-11	43.12
T+94	-2.956e-11	43.09
T+95	-2.319e-11	43.05
T+96	-1.819e-11	43.02
T+97	-1.427e-11	42.98
T+98	-1.120e-11	42.95
T+99	-8.783e-12	42.91
T+100	-6.891e-12	42.88
T+101	-5.406e-12	42.84
T+102	-4.241e-12	42.81
T+103	-3.327e-12	42.77
T+104	-2.610e-12	42.74
T+105	-2.048e-12	42.70
T+106	-1.606e-12	42.67
T+107	-1.260e-12	42.63
T+108	-9.887e-13	42.60
T+109	-7.757e-13	42.56
T+110	-6.085e-13	42.53
T+111	-4.774e-13	42.49
T+112	-3.745e-13	42.46
T+113	-2.938e-13	42.42
T+114	-2.305e-13	42.39
T+115	-1.808e-13	42.36
T+116	-1.419e-13	42.32
T+117	-1.113e-13	42.29
T+118	-8.732e-14	42.25
T+119	-6.850e-14	42.22
T+120	-5.374e-14	42.18
T+121	-4.216e-14	42.15
T+122	-3.308e-14	42.11
T+123	-2.595e-14	42.08
T+124	-2.036e-14	42.05
T+125	-1.597e-14	42.01
T+126	-1.253e-14	41.98
T+127	-9.829e-15	41.94

T+128	-7.711e-15	41.91
T+129	-6.049e-15	41.87
T+130	-4.746e-15	41.84
T+131	-3.723e-15	41.81
T+132	-2.921e-15	41.77
T+133	-2.291e-15	41.74
T+134	-1.798e-15	41.70
T+135	-1.410e-15	41.67
T+136	-1.106e-15	41.64
T+137	-8.680e-16	41.60
T+138	-6.810e-16	41.57
T+139	-5.342e-16	41.53
T+140	-4.191e-16	41.50
T+141	-3.288e-16	41.47
T+142	-2.579e-16	41.43
T+143	-2.024e-16	41.40
T+144	-1.588e-16	41.37
T+145	-1.245e-16	41.33
T+146	-9.771e-17	41.30
T+147	-7.666e-17	41.26
T+148	-6.014e-17	41.23
T+149	-4.718e-17	41.20
T+150	-3.701e-17	41.16
T+151	-2.904e-17	41.13
T+152	-2.278e-17	41.10
T+153	-1.787e-17	41.06
T+154	-1.402e-17	41.03
T+155	-1.100e-17	41.00
T+156	-8.629e-18	40.96
T+157	-6.770e-18	40.93
T+158	-5.311e-18	40.90
T+159	-4.166e-18	40.86
T+160	-3.269e-18	40.83
T+161	-2.564e-18	40.80
T+162	-2.012e-18	40.76
T+163	-1.578e-18	40.73
T+164	-1.238e-18	40.70
T+165	-9.713e-19	40.66
T+166	-7.620e-19	40.63
T+167	-5.978e-19	40.60
T+168	-4.690e-19	40.56
T+169	-3.679e-19	40.53
T+170	-2.887e-19	40.50
T+171	-2.265e-19	40.47
T+172	-1.777e-19	40.43
T+173	-1.394e-19	40.40
T+174	-1.093e-19	40.37
T+175	-8.578e-20	40.33
T+176	-6.730e-20	40.30
T+177	-5.279e-20	40.27
T+178	-4.142e-20	40.24
T+179	-3.249e-20	40.20
T+180	-2.549e-20	40.17
T+181	-2.000e-20	40.14
T+182	-1.569e-20	40.10
T+183	-1.231e-20	40.07
T+184	-9.656e-21	40.04
T+185	-7.575e-21	40.01
T+186	-5.943e-21	39.97
T+187	-4.662e-21	39.94
T+188	-3.658e-21	39.91
T+189	-2.870e-21	39.88
T+190	-2.251e-21	39.84
T+191	-1.766e-21	39.81

```

T+192 -1.386e-21 39.78
T+193 -1.087e-21 39.75
T+194 -8.527e-22 39.71
T+195 -6.690e-22 39.68
T+196 -5.248e-22 39.65
T+197 -4.117e-22 39.62
T+198 -3.230e-22 39.58
T+199 -2.534e-22 39.55
T+200 -1.988e-22 39.52
T+201 -1.560e-22 39.49
T+202 -1.224e-22 39.46
T+203 -9.599e-23 39.42
T+204 -7.531e-23 39.39
T+205 -5.908e-23 39.36
T+206 -4.635e-23 39.33
T+207 -3.636e-23 39.30
T+208 -2.853e-23 39.26

```

```

> hasil.forc.Diff11 <- forc11@forecast$seriesFor[,1]
> hasil.forc.Diff11

```

```

      T+1      T+2      T+3      T+4      T+5
-1.873704e-01 -1.469950e-01 -1.153199e-01 -9.047028e-02 -7.097536e-02
      T+6      T+7      T+8      T+9      T+10
-5.568128e-02 -4.368284e-02 -3.426988e-02 -2.688526e-02 -2.109191e-02
      T+11      T+12      T+13      T+14      T+15
-1.654693e-02 -1.298133e-02 -1.018406e-02 -7.989551e-03 -6.267929e-03
      T+16      T+17      T+18      T+19      T+20
-4.917288e-03 -3.857690e-03 -3.026418e-03 -2.374273e-03 -1.862654e-03
      T+21      T+22      T+23      T+24      T+25
-1.461282e-03 -1.146398e-03 -8.993675e-04 -7.055679e-04 -5.535291e-04
      T+26      T+27      T+28      T+29      T+30
-4.342523e-04 -3.406778e-04 -2.672670e-04 -2.096752e-04 -1.644935e-04
      T+31      T+32      T+33      T+34      T+35
-1.290477e-04 -1.012400e-04 -7.942434e-05 -6.230965e-05 -4.888290e-05
      T+36      T+37      T+38      T+39      T+40
-3.834940e-05 -3.008571e-05 -2.360271e-05 -1.851670e-05 -1.452664e-05
      T+41      T+42      T+43      T+44      T+45
-1.139638e-05 -8.940637e-06 -7.014070e-06 -5.502648e-06 -4.316914e-06
      T+46      T+47      T+48      T+49      T+50
-3.386687e-06 -2.656909e-06 -2.084387e-06 -1.635234e-06 -1.282867e-06
      T+51      T+52      T+53      T+54      T+55
-1.006429e-06 -7.895595e-07 -6.194219e-07 -4.859462e-07 -3.812324e-07
      T+56      T+57      T+58      T+59      T+60
-2.990828e-07 -2.346352e-07 -1.840750e-07 -1.444097e-07 -1.132917e-07
      T+61      T+62      T+63      T+64      T+65
-8.887911e-08 -6.972706e-08 -5.470197e-08 -4.291456e-08 -3.366714e-08
      T+66      T+67      T+68      T+69      T+70
-2.641240e-08 -2.072095e-08 -1.625591e-08 -1.275301e-08 -1.000494e-08
      T+71      T+72      T+73      T+74      T+75
-7.849033e-09 -6.157690e-09 -4.830804e-09 -3.789842e-09 -2.973190e-09
      T+76      T+77      T+78      T+79      T+80
-2.332515e-09 -1.829894e-09 -1.435581e-09 -1.126236e-09 -8.835496e-10
      T+81      T+82      T+83      T+84      T+85
-6.931586e-10 -5.437938e-10 -4.266148e-10 -3.346860e-10 -2.625664e-10
      T+86      T+87      T+88      T+89      T+90
-2.059875e-10 -1.616004e-10 -1.267781e-10 -9.945938e-11 -7.802745e-11
      T+91      T+92      T+93      T+94      T+95
-6.121376e-11 -4.802315e-11 -3.767492e-11 -2.955657e-11 -2.318759e-11
      T+96      T+97      T+98      T+99      T+100
-1.819103e-11 -1.427115e-11 -1.119594e-11 -8.783391e-12 -6.890708e-12
      T+101      T+102      T+103      T+104      T+105
-5.405869e-12 -4.240989e-12 -3.327122e-12 -2.610180e-12 -2.047727e-12
      T+106      T+107      T+108      T+109      T+110

```

```

-1.606474e-12 -1.260304e-12 -9.887284e-13 -7.756729e-13 -6.085276e-13
      T+111      T+112      T+113      T+114      T+115
-4.773995e-13 -3.745274e-13 -2.938226e-13 -2.305085e-13 -1.808375e-13
      T+116      T+117      T+118      T+119      T+120
-1.418699e-13 -1.112991e-13 -8.731592e-14 -6.850071e-14 -5.373989e-14
      T+121      T+122      T+123      T+124      T+125
-4.215978e-14 -3.307501e-14 -2.594787e-14 -2.035651e-14 -1.597000e-14
      T+126      T+127      T+128      T+129      T+130
-1.252872e-14 -9.828975e-15 -7.710986e-15 -6.049389e-15 -4.745841e-15
      T+131      T+132      T+133      T+134      T+135
-3.723187e-15 -2.920899e-15 -2.291491e-15 -1.797711e-15 -1.410332e-15
      T+136      T+137      T+138      T+139      T+140
-1.106428e-15 -8.680099e-16 -6.809674e-16 -5.342296e-16 -4.191116e-16
      T+141      T+142      T+143      T+144      T+145
-3.287996e-16 -2.579484e-16 -2.023646e-16 -1.587582e-16 -1.245483e-16
      T+146      T+147      T+148      T+149      T+150
-9.771011e-17 -7.665512e-17 -6.013714e-17 -4.717853e-17 -3.701230e-17
      T+151      T+152      T+153      T+154      T+155
-2.903673e-17 -2.277977e-17 -1.787109e-17 -1.402015e-17 -1.099903e-17
      T+156      T+157      T+158      T+159      T+160
-8.628910e-18 -6.769516e-18 -5.310791e-18 -4.166399e-18 -3.268606e-18
      T+161      T+162      T+163      T+164      T+165
-2.564272e-18 -2.011712e-18 -1.578220e-18 -1.238138e-18 -9.713388e-19
      T+166      T+167      T+168      T+169      T+170
-7.620306e-19 -5.978250e-19 -4.690031e-19 -3.679403e-19 -2.886549e-19
      T+171      T+172      T+173      T+174      T+175
-2.264543e-19 -1.776570e-19 -1.393747e-19 -1.093416e-19 -8.578023e-20
      T+176      T+177      T+178      T+179      T+180
-6.729594e-20 -5.279472e-20 -4.141829e-20 -3.249330e-20 -2.549150e-20
      T+181      T+182      T+183      T+184      T+185
-1.999848e-20 -1.568912e-20 -1.230836e-20 -9.656105e-21 -7.575366e-21
      T+186      T+187      T+188      T+189      T+190
-5.942994e-21 -4.662372e-21 -3.657704e-21 -2.869526e-21 -2.251189e-21
      T+191      T+192      T+193      T+194      T+195
-1.766093e-21 -1.385528e-21 -1.086968e-21 -8.527436e-22 -6.689907e-22
      T+196      T+197      T+198      T+199      T+200
-5.248337e-22 -4.117403e-22 -3.230167e-22 -2.534117e-22 -1.988055e-22
      T+201      T+202      T+203      T+204      T+205
-1.559660e-22 -1.223578e-22 -9.599160e-23 -7.530692e-23 -5.907946e-23
      T+206      T+207      T+208
-4.634877e-23 -3.636134e-23 -2.852604e-23
> hasilgarch11 <- diffinv(hasil.forc.Diff11, differences = 1)+ pt_1
> hasilgarch11
[1] 46482667.153      6887.984      6921.107      6867.952      6927.788
[6]      6868.455      6813.516      6863.730      6923.263      6921.830
[11]      6951.411      6917.376      6991.573      6963.552      6954.124
[16]      6954.334      6999.970      6995.260      7048.827      7001.671
[21]      7048.740      7010.825      7052.325      7070.576      7077.893
[26]      7115.350      7147.431      7103.348      7126.499      7209.966
[31]      7202.925      7213.912      7261.908      7234.663      7274.420
[36]      7198.363      7226.492      7275.323      7224.737      7215.110
[41]      7231.283      7195.895      7228.045      6908.881      6818.924
[46]      6815.334      6598.970      6597.124      6643.597      6792.545
[51]      6822.465      6917.275      6839.905      6913.272      6882.634
[56]      7025.386      7036.695      7148.101      7147.854      7182.091
[61]      7095.712      7140.175      7192.444      7181.961      7085.778
[66]      6994.572      7049.012      7006.180      7049.457      6936.097
[71]      6975.507      7043.201      6983.442      6997.398      7042.067
[76]      7015.186      6995.587      6941.482      6910.712      6793.459
[81]      6638.302      6702.397      6645.541      6651.717      6739.350
[86]      6721.275      6717.424      6640.121      6689.217      6651.035
[91]      6658.383      6735.222      6873.874      6863.261      6886.092
[96]      6857.538      6870.668      6897.346      6955.947      6950.254
[101]      6967.914      6987.288      7045.765      7056.479      7083.785

```

```

[106] 7085.980 7102.010 7085.366 7159.515 7128.407
[111] 7092.406 7132.584 7185.689 7171.565 7107.113
[116] 7162.396 7193.837 7173.338 7134.379 7131.175
[121] 7158.604 7177.720 7152.234 7176.310 7231.013
[126] 7232.285 7185.887 7231.148 7241.786 7253.594
[131] 7317.147 7277.209 7304.732 7168.001 7194.618
[136] 7196.082 7187.444 7218.036 7177.713 7126.633
[141] 7111.580 7076.161 7035.329 7039.928 7008.848
[146] 7071.386 7074.515 7075.754 7025.914 6993.525
[151] 6938.275 6908.339 6879.755 6813.660 6830.246
[156] 6833.625 6859.546 6979.782 7016.901 7052.174
[161] 7047.510 7043.069 7090.892 7055.171 7098.021
[166] 7051.434 7014.820 7033.704 7044.657 7101.526
[171] 7049.256 7069.214 6965.967 7088.337 7018.523
[176] 7034.630 7013.514 7044.116 7081.312 7062.378
[181] 7029.718 7053.254 7079.646 7052.280 7016.488
[186] 7011.199 7080.443 7019.934 7018.770 6986.459
[191] 6891.700 6817.882 6803.358 6714.249 6733.583
[196] 6809.446 6800.879 6750.990 6811.323 6778.829
[201] 6767.446 6819.794 6823.563 6799.803 6834.939
[206] 6922.157 6849.650 6859.208 6849.750

> #pembuatan tabel perbandingan train dan testing
> perbandingan11 <- data.frame("Aktual"= testing.ts,
+                               "Ramalan" = hasilgarch11[-1])
> head(perbandingan11,10)
   Aktual  Ramalan
1  6888.171 6887.984
2  6921.441 6921.107
3  6868.402 6867.952
4  6928.328 6927.788
5  6869.066 6868.455
6  6814.183 6813.516
7  6864.441 6863.730
8  6924.008 6923.263
9  6922.602 6921.830
10 6952.204 6951.411
> str(perbandingan11)
'data.frame': 208 obs. of 2 variables:
 $ Aktual : Time-Series from 486 to 693: 6888 6921 6868 6928 6869 ...
 $ Ramalan: num 6888 6921 6868 6928 6868 ...
> dim(perbandingan11)
[1] 208 2
> #perbandingan
> # Menulis data perbandingan ke file csv
> write.csv(perbandingan11, file = "C:/Users/asus/Dropbox/Penelitian UII/Anali
sis Data/perbandingan Masa.csv", row.names = FALSE)
> #MAPE PERBANDINGAN 12
> Date<-c(datasaham$Date[486:693])
> dataframe <- data.frame(Date, perbandingan11)
> T <- nrow(dataframe)
> MAPE <- 1/T*sum(abs((dataframe$Aktual-dataframe$Ramalan)/dataframe$Aktual)*1
00)
> MAPE
[1] 0.01220303
> length(Date)
[1] 208
> coef(model2)
      ar1      ma1
0.2901343 -0.2854199
> # Menghitung Error (Selisih Aktual dan Ramalan)
> error <- perbandingan11$Aktual - perbandingan11$Ramalan
> # Menghitung Mean Squared Error (MSE)
> MSE <- mean(error^2)
> # Menghitung Root Mean Squared Error (RMSE)

```

```

> RMSE <- sqrt(MSE)
> # Menampilkan hasil
> cat("Mean Squared Error (MSE):", MSE, "\n")
Mean Squared Error (MSE): 0.7354352
> cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
Root Mean Squared Error (RMSE): 0.8575752
> # Mengonversi kedua kolom ke dalam format time series
> # Mengonversi perbandingan12[,1] (data aktual) ke time series
> ts_aktual <- ts(perbandingan11[, 1], frequency = 1) # Gantilah frequency se
suai data (1 untuk data harian)
> # Mengonversi perbandingan12[,2] (hasil forecast) ke time series
> ts_forecast <- ts(perbandingan11[, 2], frequency = 1) # Gantilah frequency
sesuai data
> # Membuat plot perbandingan antara data aktual dan hasil forecast
> plot(ts_aktual, type = "l", col = "yellow", lty = 1, # Garis solid untuk dat
a aktual
+       xlab = "Observasi", ylab = "Nilai",
+       main = "Perbandingan Data Aktual dan Hasil Forecast",
+       lwd = 4) # Rentang Y agar kedua data terlihat
> # Menambahkan garis untuk hasil forecast
> lines(ts_forecast, col = "blue", lty = 3, lwd = 2) # Garis putus-putus untu
k hasil forecast
> # Menambahkan legend untuk membedakan data
> legend("bottomright", legend = c("Data Aktual", "Hasil Forecast"),
+       col = c("yellow", "blue"), lty = c(1, 2), lwd = 2)

```

Lampiran 4. Sintaks R Model ARIMA GARCH SetelahCovid-19

```
> library(ggplot2)
> library(tsibble)
> library(tseries)
> library(MASS)
> library(dplyr)
> library(TTR)
> library(forecast)
> library(lmtest)
> library(HoRM)
> library(dLagM)
> library(dynlm)
> library(MLmetrics)
> library(car)
> library(TSA)
> library(aTSA)
> library(rugarch)
> library(imputeTS)
> library(tidyr)
> # Membaca data
> setwd("C:/Users/asus/Dropbox/Penelitian UII/Analisis Data")
> datasaham <- read.csv("Setelah Covid.csv")
> # Menghapus kolom yang tidak diperlukan (sesuaikan dengan struktur data Anda)
> datasaham <- datasaham[,-c(2,3,4,6,7)]
> # Konversi kolom Date ke format waktu
> datasaham$Date <- as.Date(datasaham$Date, format="%m/%d/%Y")
> datasaham$Date <- as.POSIXct(datasaham$Date)
> # Konversi data menjadi time series
> ts.datasaham <- ts(datasaham$Close)
> # Menampilkan subset data dengan tabel yang lebih rapi
> kableExtra::kable(head(datasaham), caption = 'Subset Data Harian IHSG')
```

Table: Subset Data Harian IHSG

Date	Close
2023-01-02	6850.984
2023-01-03	6888.758
2023-01-04	6813.239
2023-01-05	6653.841
2023-01-06	6684.558
2023-01-09	6688.265

```
> # Melihat struktur dan ringkasan data
> str(datasaham)
'data.frame': 398 obs. of 2 variables:
 $ Date : POSIXct, format: "2023-01-02" ...
 $ Close: num 6851 6889 6813 6654 6685 ...
> summary(datasaham)
      Date      Close
Min.   :2023-01-02 00:00:00.0   Min.   :6566
1st Qu.:2023-06-07 06:00:00.0   1st Qu.:6828
Median :2023-11-01 12:00:00.0   Median :6940
Mean    :2023-11-03 17:47:20.2   Mean    :7009
3rd Qu.:2024-04-01 18:00:00.0   3rd Qu.:7224
Max.    :2024-09-05 00:00:00.0   Max.    :7695
> # Konversi kolom Date ke format Date
> datasaham$Date <- as.Date(datasaham$Date)
> # konversi ke time series (menggunakan hargautupan saham)
> ts.datasaham <- ts(datasaham$Close)
```

```

> # Cek kembali struktur setelah perubahan
> str(datasaham)
'data.frame': 398 obs. of 2 variables:
 $ Date : Date, format: "2023-01-02" ...
 $ Close: num 6851 6889 6813 6654 6685 ...
> # Membuat Boxplot Harga Tutupan Saham
> y <- datasaham$Date
> x <- datasaham$Close
> boxplot(x, xlab="Harga Tutupan IHSG", col="lightblue")
> # Plot harga harian IHSG datasaham Covid-19
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line(color = "blue") +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020",
+         x = "Tanggal",
+         y = "Harga IHSG") +
+   theme(plot.title = element_text(face = "bold", hjust = 0.5),
+         plot.subtitle = element_text(hjust = 0.5),
+         panel.background = element_rect(fill = NA))
> # Menentukan proporsi data pelatihan (70%)
> p = 0.7
> # Menghitung jumlah data untuk pelatihan
> jumlah.train <- as.integer(p * nrow(datasaham))
> # Membuat plot
> ggplot(datasaham, aes(x = Date, y = Close)) +
+   geom_line() +
+   labs(title = "Plot Data Harian Harga IHSG datasaham Covid-19",
+         subtitle = "1 Januari 2018 - 1 Maret 2020") +
+   geom_vline(aes(xintercept = Date[jumlah.train],
+                   col="Pemisah_data_train"), lty=2, lwd=.7) +
+   theme(plot.title = element_text(face = "bold", hjust = .5),
+         plot.subtitle = element_text(hjust = .5),
+         legend.position = "bottom",
+         panel.background = element_rect(fill=NA)) +
+   scale_color_manual(name = "", values = c(Pemisah_data_train = "blue"))
> # Menentukan data pelatihan
> datasaham[jumlah.train,1]
[1] "2024-02-28"
> train = 1:jumlah.train
> # Membuat time series untuk data pelatihan dan pengujian
> train.ts <- ts(datasaham$Close[train])
> testing.ts <- ts(datasaham$Close[-train], start = jumlah.train + 1)
> # Menentukan jumlah data pelatihan dan pengujian
> jumlah.train <- length(train.ts)
> jumlah.testing <- length(testing.ts)
> # Menampilkan hasil
> cat("Jumlah data pelatihan:", jumlah.train, "\n")
Jumlah data pelatihan: 278
> cat("Jumlah data pengujian: ", jumlah.testing, "\n")
Jumlah data pengujian: 120
> #Identifikasi Stasioneritas = differencing
> library(tseries)
> par(mfrow=c(1,2))
> acf(train.ts, main = "ACF Plot for MA(q)")
> pacf(train.ts, main = "PACF Plot for AR(p)")
> adf.test(train.ts)
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
      lag   ADF p.value
[1,]    0 0.649  0.831
[2,]    1 0.719  0.851
[3,]    2 0.863  0.892

```

```
[4,] 3 1.134 0.931
[5,] 4 1.080 0.924
[6,] 5 1.096 0.926
```

Type 2: with drift no trend

```
lag ADF p.value
[1,] 0 -1.202 0.626
[2,] 1 -0.810 0.764
[3,] 2 -0.721 0.795
[4,] 3 -0.897 0.733
[5,] 4 -1.085 0.667
[6,] 5 -1.015 0.692
```

Type 3: with drift and trend

```
lag ADF p.value
[1,] 0 -2.86 0.213
[2,] 1 -2.56 0.342
[3,] 2 -2.29 0.454
[4,] 3 -2.06 0.551
[5,] 4 -2.24 0.475
[6,] 5 -2.18 0.500
```

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> library(forecast)
> train.diff<-diff(train.ts, differences = 1)
> par(mfrow=c(1,2))
> acf(train.diff, main = "ACF Plot for MA(q)")
> pacf(train.diff, main = "PACF Plot for AR(p)")
> adf.test(train.diff)
```

Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend

```
lag ADF p.value
[1,] 0 -19.59 0.01
[2,] 1 -13.31 0.01
[3,] 2 -10.50 0.01
[4,] 3 -8.23 0.01
[5,] 4 -7.51 0.01
[6,] 5 -7.40 0.01
```

Type 2: with drift no trend

```
lag ADF p.value
[1,] 0 -19.59 0.01
[2,] 1 -13.33 0.01
[3,] 2 -10.56 0.01
[4,] 3 -8.31 0.01
[5,] 4 -7.59 0.01
[6,] 5 -7.51 0.01
```

Type 3: with drift and trend

```
lag ADF p.value
[1,] 0 -19.63 0.01
[2,] 1 -13.37 0.01
[3,] 2 -10.56 0.01
[4,] 3 -8.30 0.01
[5,] 4 -7.59 0.01
[6,] 5 -7.51 0.01
```

Note: in fact, p.value = 0.01 means p.value <= 0.01

```
> #Identifikasi Model ARIMA dengan ACF, PACF, dan EACF
> acf(train.diff, main="ACF IHSG datasaham Covid-19")
> pacf(train.diff, main="PACF IHSG datasaham Covid-19")
> library(TSA)
> eacf(train.diff)
```

AR/MA

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o o o o o o o o o o o o
```



```

1 x o o o o o o o o o o o o o o
2 x x o o o o o o o o o o o o
3 x x o o o o o o o o o o o o
4 x x x o o o o o o o o o o o
5 x x x o x o o o o o o o o o
6 o x x x x o o o o o o o o o
7 o o o o x o o o o o o o o o

```

```

> #penentuan model ARIMA
> library(forecast)
> model_arima <- auto.arima(train.diff, trace = T)

```

Fitting models using approximations to speed things up...

```

ARIMA(2,0,2) with non-zero mean : 2845.551
ARIMA(0,0,0) with non-zero mean : 2860.622
ARIMA(1,0,0) with non-zero mean : 2855.074
ARIMA(0,0,1) with non-zero mean : 2854.246
ARIMA(0,0,0) with zero mean      : 2859.059
ARIMA(1,0,2) with non-zero mean : 2858.728
ARIMA(2,0,1) with non-zero mean : 2848.346
ARIMA(3,0,2) with non-zero mean : 2842.511
ARIMA(3,0,1) with non-zero mean : 2840.576
ARIMA(3,0,0) with non-zero mean : 2838.599
ARIMA(2,0,0) with non-zero mean : 2854.259
ARIMA(4,0,0) with non-zero mean : 2840.234
ARIMA(4,0,1) with non-zero mean : 2842.232
ARIMA(3,0,0) with zero mean      : 2837.899
ARIMA(2,0,0) with zero mean      : 2852.993
ARIMA(4,0,0) with zero mean      : 2839.415
ARIMA(3,0,1) with zero mean      : 2839.907
ARIMA(2,0,1) with zero mean      : 2847.922
ARIMA(4,0,1) with zero mean      : 2841.415

```

Now re-fitting the best model(s) without approximations...

```
ARIMA(3,0,0) with zero mean      : 2856.474
```

Best model: ARIMA(3,0,0) with zero mean

```

> summary(model_arima)
Series: train.diff
ARIMA(3,0,0) with zero mean

```

```

Coefficients:
      ar1      ar2      ar3
-0.1729 -0.0485  0.0304
s.e.    0.0601  0.0615  0.0622

```

```

sigma^2 = 1730: log likelihood = -1424.16
AIC=2856.33  AICc=2856.47  BIC=2870.82

```

Training set error measures:

```

              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 2.020422 41.36275 32.20808 9.94447 186.581 0.6437399

```

```

              ACF1
Training set -0.004436864

```

```
> model1 <- Arima(train.diff, order = c(3,1,0), method = "ML")
```

```

> summary(model1)
Series: train.diff
ARIMA(3,1,0)

```

```

Coefficients:
      ar1      ar2      ar3
-0.9405 -0.6917 -0.3325

```

```
s.e.    0.0575    0.0699    0.0582
```

```
sigma^2 = 2101: log likelihood = -1446.41
AIC=2900.82  AICc=2900.96  BIC=2915.3
```

```
Training set error measures:
```

```
Training set      ME      RMSE      MAE      MPE      MAPE      MASE
                  ACF1
Training set -0.03928843
```

```
> model2 <- Arima(train.diff,order = c(2,1,0),method ="ML")
> summary(model2)
```

```
Series: train.diff
ARIMA(2,1,0)
```

```
Coefficients:
```

```
      ar1      ar2
-0.8064 -0.4273
s.e.    0.0555    0.0555
```

```
sigma^2 = 2343: log likelihood = -1461.79
AIC=2929.58  AICc=2929.67  BIC=2940.44
```

```
Training set error measures:
```

```
Training set      ME      RMSE      MAE      MPE      MAPE      MASE
                  ACF1
Training set -0.1175815
```

```
> model3 <- Arima(train.diff,order = c(1,1,0),method ="ML")
> summary(model3)
```

```
Series: train.diff
ARIMA(1,1,0)
```

```
Coefficients:
```

```
      ar1
-0.5611
s.e.    0.0501
```

```
sigma^2 = 2838: log likelihood = -1488.55
AIC=2981.1  AICc=2981.15  BIC=2988.34
```

```
Training set error measures:
```

```
Training set      ME      RMSE      MAE      MPE      MAPE      MASE
                  ACF1
Training set -0.2239703
```

```
> model4 <- Arima(train.diff,order = c(0,1,0),method ="ML")
> summary(model4)
```

```
Series: train.diff
ARIMA(0,1,0)
```

```
sigma^2 = 4117: log likelihood = -1540.19
AIC=3082.38  AICc=3082.39  BIC=3086
```

```
Training set error measures:
```

```
Training set      ME      RMSE      MAE      MPE      MAPE      MASE
                  ACF1
Training set -0.5559526
```

```
> model5 <- Arima(train.diff,order = c(3,1,1),method ="ML")
> summary(model5)
```

```
Series: train.diff
ARIMA(3,1,1)
```

```

Coefficients:
      ar1      ar2      ar3      ma1
-0.1713 -0.0471  0.0310 -1.0000
s.e.    0.0603  0.0618  0.0625  0.0217

sigma^2 = 1738: log likelihood = -1422.19
AIC=2854.38  AICc=2854.6  BIC=2872.48

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.280217 41.31553 32.3895 11.47587 199.6115 0.647366
      ACF1
Training set 0.0002981737
> model6 <- Arima(train.diff,order = c(2,1,1),method ="ML")
> summary(model6)
Series: train.diff
ARIMA(2,1,1)

Coefficients:
      ar1      ar2      ma1
-0.1729 -0.0533 -1.0000
s.e.    0.0602  0.0605  0.0242

sigma^2 = 1733: log likelihood = -1422.31
AIC=2852.62  AICc=2852.77  BIC=2867.1

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.300962 41.32946 32.4162 13.94556 194.7471 0.6478995
      ACF1
Training set 0.003147984
> model7 <- Arima(train.diff,order = c(1,1,1),method ="ML")
> summary(model7)
Series: train.diff
ARIMA(1,1,1)

Coefficients:
      ar1      ma1
-0.1641 -1.0000
s.e.    0.0595  0.0193

sigma^2 = 1732: log likelihood = -1422.7
AIC=2851.4  AICc=2851.49  BIC=2862.26

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.134269 41.39565 32.61023 38.63453 181.2499 0.6517776
      ACF1
Training set -0.006419377
> model8 <- Arima(train.diff,order = c(0,1,1),method ="ML")
> summary(model8)
Series: train.diff
ARIMA(0,1,1)

Coefficients:
      ma1
-1.0000
s.e.    0.0135

sigma^2 = 1776: log likelihood = -1426.44
AIC=2856.89  AICc=2856.93  BIC=2864.13

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE

```

Training set 2.552162 41.98615 32.89618 98.43869 120.3918 0.6574929

ACF1
Training set -0.1610647

```
> model9 <- Arima(train.diff,order = c(3,1,2),method ="ML")
```

```
> summary(model9)
```

Series: train.diff

ARIMA(3,1,2)

Coefficients:

	ar1	ar2	ar3	ma1	ma2
	0.2296	0.0250	0.0665	-1.4010	0.401
s.e.	0.5098	0.1057	0.0642	0.5082	0.508

sigma^2 = 1743: log likelihood = -1422

AIC=2856.01 AICc=2856.32 BIC=2877.73

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.252878	41.29432	32.37094	11.78915	204.4748	0.646995

ACF1
Training set 0.0008297825

```
> model10 <- Arima(train.diff,order = c(2,1,2),method ="ML")
```

```
> summary(model10)
```

Series: train.diff

ARIMA(2,1,2)

Coefficients:

	ar1	ar2	ma1	ma2
	-0.3155	-0.078	-0.8573	-0.1427
s.e.	0.5667	0.106	0.5662	0.5657

sigma^2 = 1739: log likelihood = -1422.28

AIC=2854.56 AICc=2854.78 BIC=2872.66

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.301382	41.3256	32.40618	12.72558	196.1226	0.6476994

ACF1
Training set 0.002679543

```
> model11 <- Arima(train.diff,order = c(1,1,2),method ="ML")
```

```
> summary(model11)
```

Series: train.diff

ARIMA(1,1,2)

Coefficients:

	ar1	ma1	ma2
	0.0248	-1.1969	0.1969
s.e.	0.2724	0.2656	0.2647

sigma^2 = 1735: log likelihood = -1422.45

AIC=2852.9 AICc=2853.05 BIC=2867.38

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	3.250056	41.35241	32.49964	23.18138	189.3116	0.6495673

ACF1
Training set 0.002249664

```
> model12 <- Arima(train.diff,order = c(0,1,2),method ="ML")
```

```
> summary(model12)
```

Series: train.diff

ARIMA(0,1,2)

Coefficients:

	ma1	ma2

```

      -1.1730  0.1730
s.e.    0.0632  0.0594

```

```

sigma^2 = 1729: log likelihood = -1422.45
AIC=2850.91 AICc=2851 BIC=2861.77

```

Training set error measures:

```

      ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.242671 41.35295 32.51084 24.22337 189.3233 0.6497911
      ACF1
Training set 0.002945227

```

```

data: sisaan
X-squared = 0.011539, df = 1, p-value = 0.9145

```

```
> t.test(sisaan, mu = 0, conf.level = 0.95)
```

One sample t-test

```

data: sisaan
t = 1.2615, df = 276, p-value = 0.2082
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1.756852  8.025389
sample estimates:
mean of x
 3.134269

```

```

> #white Test
> white_test <- bptest(sisaan ~ fitted(model7) + I(fitted(model7)^2))
> print(white_test)

```

studentized Breusch-Pagan test

```

data: sisaan ~ fitted(model7) + I(fitted(model7)^2)
BP = 7.5882, df = 2, p-value = 0.0225

```

```
> #uji box cox penanganan heterogenitas variansi, dilakukan transformasi kuadratik
```

```

> par(mfrow=c(1,1))
> train.ts.new <- (train.ts)^2
> index <- seq(1:278)
> bc = boxcox(train.ts.new~index, lambda = seq(-2, 4, by=.01))
> #Nilai Rounded Lambda
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] -2
> #SK
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] -2.00 -1.99 -1.98 -1.97 -1.96 -1.95 -1.94 -1.93 -1.92 -1.91
[11] -1.90 -1.89 -1.88 -1.87 -1.86 -1.85 -1.84 -1.83 -1.82 -1.81
[21] -1.80 -1.79 -1.78 -1.77 -1.76 -1.75 -1.74 -1.73 -1.72 -1.71
[31] -1.70 -1.69 -1.68 -1.67 -1.66 -1.65 -1.64 -1.63 -1.62 -1.61
[41] -1.60 -1.59 -1.58 -1.57 -1.56 -1.55 -1.54 -1.53 -1.52 -1.51
[51] -1.50 -1.49 -1.48 -1.47 -1.46 -1.45 -1.44 -1.43 -1.42 -1.41
[61] -1.40 -1.39 -1.38 -1.37 -1.36 -1.35 -1.34 -1.33 -1.32 -1.31
[71] -1.30 -1.29 -1.28 -1.27 -1.26 -1.25 -1.24 -1.23 -1.22 -1.21
[81] -1.20 -1.19 -1.18 -1.17 -1.16 -1.15 -1.14 -1.13 -1.12 -1.11
[91] -1.10 -1.09 -1.08 -1.07 -1.06 -1.05 -1.04 -1.03 -1.02 -1.01
[101] -1.00 -0.99 -0.98 -0.97 -0.96 -0.95 -0.94 -0.93 -0.92 -0.91
[111] -0.90 -0.89 -0.88 -0.87 -0.86 -0.85 -0.84 -0.83 -0.82 -0.81
[121] -0.80 -0.79 -0.78 -0.77 -0.76 -0.75 -0.74 -0.73 -0.72 -0.71
[131] -0.70 -0.69 -0.68 -0.67 -0.66 -0.65 -0.64 -0.63 -0.62 -0.61
[141] -0.60 -0.59 -0.58 -0.57 -0.56 -0.55 -0.54 -0.53 -0.52 -0.51
[151] -0.50 -0.49 -0.48 -0.47 -0.46 -0.45

```

```
> plot.ts(train.ts.new, lty=1, xlab="waktu", ylab="Data Transformasi", main="Plot Transformasi Data")
> #eksekusi peramalan data train (70%)
> ramalan.da <- forecast::forecast(model2, h = 120)
> data.ramalan.da <- ramalan.da$mean
> plot(ramalan.da)
> summary(ramalan.da)
```

Forecast method: ARIMA(1,1,1)

Model Information:
Series: train.ts
ARIMA(1,1,1)

Coefficients:

	ar1	ma1
	0.0254	-0.1987
s.e.	0.2696	0.2614

sigma^2 = 1727: log likelihood = -1424.43
AIC=2854.85 AICc=2854.94 BIC=2865.72

Error measures:

	ME	RMSE	MAE	MPE	MAPE
Training set	2.078381	41.32987	32.21772	0.02722575	0.4669163

	MASE	ACF1
Training set	0.9815286	-0.001842747

Forecasts:

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
279	7321.217	7267.962	7374.471	7239.771	7402.663
280	7321.029	7251.933	7390.124	7215.356	7426.702
281	7321.024	7239.220	7402.828	7195.916	7446.132
282	7321.024	7228.239	7413.808	7179.122	7462.925
283	7321.024	7218.427	7423.620	7164.116	7477.932
284	7321.024	7209.475	7432.573	7150.424	7491.623
285	7321.024	7201.189	7440.858	7137.753	7504.295
286	7321.024	7193.441	7448.607	7125.903	7516.145
287	7321.024	7186.137	7455.911	7114.732	7527.315
288	7321.024	7179.209	7462.839	7104.136	7537.912
289	7321.024	7172.603	7469.444	7094.034	7548.013
290	7321.024	7166.280	7475.768	7084.363	7557.685
291	7321.024	7160.204	7481.843	7075.072	7566.976
292	7321.024	7154.351	7487.697	7066.119	7575.928
293	7321.024	7148.695	7493.352	7057.470	7584.577
294	7321.024	7143.220	7498.828	7049.097	7592.951
295	7321.024	7137.908	7504.139	7040.973	7601.075
296	7321.024	7132.746	7509.301	7033.078	7608.969
297	7321.024	7127.722	7514.325	7025.395	7616.653
298	7321.024	7122.826	7519.222	7017.906	7624.142
299	7321.024	7118.047	7524.001	7010.597	7631.450
300	7321.024	7113.378	7528.670	7003.457	7638.591
301	7321.024	7108.812	7533.236	6996.474	7645.574
302	7321.024	7104.342	7537.705	6989.638	7652.410
303	7321.024	7099.963	7542.085	6982.940	7659.108
304	7321.024	7095.668	7546.379	6976.372	7665.675
305	7321.024	7091.454	7550.594	6969.927	7672.120
306	7321.024	7087.316	7554.732	6963.599	7678.449
307	7321.024	7083.250	7558.798	6957.380	7684.668
308	7321.024	7079.252	7562.795	6951.266	7690.782
309	7321.024	7075.319	7566.728	6945.251	7696.796
310	7321.024	7071.449	7570.599	6939.332	7702.716
311	7321.024	7067.637	7574.411	6933.502	7708.545
312	7321.024	7063.882	7578.166	6927.759	7714.288

313	7321.024	7060.181	7581.867	6922.099	7719.949
314	7321.024	7056.532	7585.516	6916.518	7725.530
315	7321.024	7052.932	7589.116	6911.013	7731.035
316	7321.024	7049.380	7592.668	6905.581	7736.467
317	7321.024	7045.874	7596.174	6900.218	7741.829
318	7321.024	7042.412	7599.636	6894.924	7747.124
319	7321.024	7038.993	7603.055	6889.694	7752.353
320	7321.024	7035.614	7606.434	6884.527	7757.520
321	7321.024	7032.275	7609.773	6879.421	7762.627
322	7321.024	7028.974	7613.073	6874.373	7767.675
323	7321.024	7025.710	7616.337	6869.381	7772.667
324	7321.024	7022.482	7619.565	6864.444	7777.604
325	7321.024	7019.288	7622.759	6859.559	7782.488
326	7321.024	7016.128	7625.919	6854.726	7787.321
327	7321.024	7013.000	7629.047	6849.943	7792.105
328	7321.024	7009.904	7632.144	6845.207	7796.840
329	7321.024	7006.838	7635.209	6840.518	7801.529
330	7321.024	7003.802	7638.246	6835.875	7806.173
331	7321.024	7000.794	7641.253	6831.275	7810.772
332	7321.024	6997.815	7644.233	6826.719	7815.329
333	7321.024	6994.863	7647.185	6822.204	7819.844
334	7321.024	6991.937	7650.111	6817.729	7824.319
335	7321.024	6989.037	7653.011	6813.294	7828.754
336	7321.024	6986.162	7655.885	6808.897	7833.150
337	7321.024	6983.312	7658.736	6804.538	7837.510
338	7321.024	6980.485	7661.562	6800.215	7841.833
339	7321.024	6977.682	7664.366	6795.928	7846.120
340	7321.024	6974.901	7667.146	6791.675	7850.372
341	7321.024	6972.143	7669.905	6787.457	7854.591
342	7321.024	6969.406	7672.641	6783.271	7858.776
343	7321.024	6966.691	7675.357	6779.118	7862.930
344	7321.024	6963.996	7678.052	6774.996	7867.051
345	7321.024	6961.321	7680.727	6770.906	7871.142
346	7321.024	6958.666	7683.382	6766.845	7875.203
347	7321.024	6956.030	7686.018	6762.814	7879.234
348	7321.024	6953.413	7688.634	6758.812	7883.236
349	7321.024	6950.815	7691.233	6754.838	7887.209
350	7321.024	6948.235	7693.813	6750.892	7891.156
351	7321.024	6945.672	7696.375	6746.973	7895.074
352	7321.024	6943.127	7698.920	6743.081	7898.967
353	7321.024	6940.599	7701.449	6739.214	7902.833
354	7321.024	6938.088	7703.960	6735.374	7906.674
355	7321.024	6935.593	7706.455	6731.558	7910.490
356	7321.024	6933.114	7708.934	6727.766	7914.281
357	7321.024	6930.650	7711.397	6723.999	7918.048
358	7321.024	6928.203	7713.845	6720.256	7921.792
359	7321.024	6925.770	7716.278	6716.535	7925.512
360	7321.024	6923.352	7718.695	6712.838	7929.210
361	7321.024	6920.949	7721.099	6709.162	7932.885
362	7321.024	6918.560	7723.487	6705.509	7936.539
363	7321.024	6916.186	7725.862	6701.877	7940.170
364	7321.024	6913.825	7728.223	6698.267	7943.781
365	7321.024	6911.477	7730.570	6694.677	7947.371
366	7321.024	6909.144	7732.904	6691.107	7950.940
367	7321.024	6906.823	7735.225	6687.558	7954.489
368	7321.024	6904.515	7737.532	6684.029	7958.019
369	7321.024	6902.220	7739.828	6680.519	7961.529
370	7321.024	6899.938	7742.110	6677.028	7965.020
371	7321.024	6897.667	7744.380	6673.556	7968.492
372	7321.024	6895.409	7746.639	6670.102	7971.945
373	7321.024	6893.163	7748.885	6666.667	7975.381
374	7321.024	6890.928	7751.119	6663.250	7978.798
375	7321.024	6888.705	7753.342	6659.850	7982.198
376	7321.024	6886.494	7755.554	6656.468	7985.580

```

377      7321.024 6884.293 7757.754 6653.102 7988.945
378      7321.024 6882.104 7759.943 6649.754 7992.294
379      7321.024 6879.926 7762.122 6646.422 7995.625
380      7321.024 6877.758 7764.290 6643.107 7998.941
381      7321.024 6875.601 7766.447 6639.808 8002.240
382      7321.024 6873.454 7768.594 6636.525 8005.523
383      7321.024 6871.317 7770.730 6633.257 8008.791
384      7321.024 6869.191 7772.857 6630.005 8012.043
385      7321.024 6867.074 7774.973 6626.768 8015.280
386      7321.024 6864.968 7777.080 6623.546 8018.502
387      7321.024 6862.871 7779.177 6620.339 8021.709
388      7321.024 6860.783 7781.264 6617.146 8024.901
389      7321.024 6858.705 7783.342 6613.968 8028.079
390      7321.024 6856.636 7785.411 6610.805 8031.243
391      7321.024 6854.577 7787.471 6607.655 8034.393
392      7321.024 6852.526 7789.521 6604.519 8037.529
393      7321.024 6850.485 7791.563 6601.397 8040.651
394      7321.024 6848.452 7793.596 6598.288 8043.760
395      7321.024 6846.428 7795.620 6595.192 8046.855
396      7321.024 6844.413 7797.635 6592.110 8049.938
397      7321.024 6842.406 7799.642 6589.040 8053.007
398      7321.024 6840.407 7801.641 6585.984 8056.064
> #pengembalian ke nilai awal
> pt_1 <- train.ts.new[278] #nilai akhir data latih
> head(pt_1)
[1] 53708909
> train.ts[278]
[1] 7328.636
> hasil.forc.Diff <- data.ramalan.da
> hasil <- diffinv(hasil.forc.Diff, differences = 1) + pt_1
> hasil
Time Series:
Start = 278
End = 398
Frequency = 1
[1] 53708909 53716230 53723551 53730872 53738193 53745514 53752835
[8] 53760156 53767477 53774798 53782119 53789440 53796761 53804082
[15] 53811404 53818725 53826046 53833367 53840688 53848009 53855330
[22] 53862651 53869972 53877293 53884614 53891935 53899256 53906577
[29] 53913898 53921219 53928540 53935861 53943182 53950503 53957824
[36] 53965145 53972466 53979787 53987108 53994429 54001750 54009071
[43] 54016392 54023713 54031034 54038355 54045676 54052997 54060318
[50] 54067639 54074960 54082281 54089602 54096923 54104244 54111565
[57] 54118887 54126208 54133529 54140850 54148171 54155492 54162813
[64] 54170134 54177455 54184776 54192097 54199418 54206739 54214060
[71] 54221381 54228702 54236023 54243344 54250665 54257986 54265307
[78] 54272628 54279949 54287270 54294591 54301912 54309233 54316554
[85] 54323875 54331196 54338517 54345838 54353159 54360480 54367801
[92] 54375122 54382443 54389764 54397085 54404406 54411727 54419048
[99] 54426370 54433691 54441012 54448333 54455654 54462975 54470296
[106] 54477617 54484938 54492259 54499580 54506901 54514222 54521543
[113] 54528864 54536185 54543506 54550827 54558148 54565469 54572790
[120] 54580111 54587432
> hasil1 <- abs(hasil)^(1/2) #hasil peramalan arima nilai data telah dikembali
kan ke asli
> pt_1 <- c(train.ts.new[278],testing.ts) #nilai akhir data latih
> pt_1
[1] 53708908.992      7316.111      7311.907      7276.749      7247.460
[6]      7329.801      7373.964      7381.907      7421.207      7433.315
[11]      7328.054      7302.449      7336.746      7331.128      7338.353
[16]      7350.152      7377.760      7365.664      7310.092      7288.813
[21]      7205.061      7236.984      7166.844      7254.399      7286.882
[26]      7164.807      7130.841      7166.814      7087.317      7073.820
[31]      7110.813      7174.533      7155.294      7036.075      7155.784

```



```

[36] 7234.197 7117.425 7134.724 7135.890 7123.612
[41] 7088.795 7099.261 7083.763 7179.831 7246.696
[46] 7317.238 7266.691 7186.037 7222.382 7176.420
[51] 7253.626 7140.229 7034.142 6970.736 7036.191
[56] 7099.312 6947.670 6974.897 6897.950 6921.548
[61] 6855.691 6850.097 6831.564 6734.832 6726.919
[66] 6819.321 6879.978 6889.165 6882.704 6905.642
[71] 6967.951 7063.577 7139.626 7125.142 7196.755
[76] 7220.889 7253.372 7250.977 7269.801 7287.042
[81] 7300.407 7327.580 7278.863 7224.293 7224.218
[86] 7321.071 7294.495 7321.976 7313.857 7262.759
[91] 7240.277 7288.167 7288.897 7241.864 7255.762
[96] 7325.985 7308.123 7059.653 7129.215 7212.131
[101] 7195.121 7256.996 7297.625 7356.638 7436.039
[106] 7409.501 7432.090 7466.831 7533.985 7554.593
[111] 7488.676 7544.298 7606.195 7597.881 7658.875
[116] 7627.604 7670.733 7694.530 7616.521 7672.896
[121] 7681.043
> coef(model2)
      ar1      ma1
0.02535456 -0.19866759
> library(TSA)
> model2 <- Arima(train.ts,order = c(1,1,1))
> for (i in 1:15) {
+   ArchTest <- ArchTest(model2$residuals, lags=i, demean=TRUE)
+   cat("P Value LM Test lag ke", i,"adalah" , ArchTest$p.value, "\n") }
P Value LM Test lag ke 1 adalah 0.2313214
P Value LM Test lag ke 2 adalah 0.3595773
P Value LM Test lag ke 3 adalah 0.01024156
P Value LM Test lag ke 4 adalah 0.001019021
P Value LM Test lag ke 5 adalah 0.001072382
P Value LM Test lag ke 6 adalah 0.0009142692
P Value LM Test lag ke 7 adalah 0.0004820666
P Value LM Test lag ke 8 adalah 0.0008633758
P Value LM Test lag ke 9 adalah 0.001440456
P Value LM Test lag ke 10 adalah 0.00257893
P Value LM Test lag ke 11 adalah 0.002962398
P Value LM Test lag ke 12 adalah 0.0008796368
P Value LM Test lag ke 13 adalah 0.001265924
P Value LM Test lag ke 14 adalah 0.002070809
P Value LM Test lag ke 15 adalah 0.003660397
> #pengecekan efek GARCH
> train.diff.new <- (train.diff)^2
> sisaan.new <- (sisaan)^2
> garch01 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garcho
rder=c(0,1)))
> garch01

*-----*
*          GARCH Model Spec          *
*-----*

Conditional Variance Dynamics
-----
GARCH Model          : sGARCH(0,1)
Variance Targeting   : FALSE

Conditional Mean Dynamics
-----
Mean Model           : ARFIMA(1,0,1)
Include Mean         : FALSE
GARCH-in-Mean        : FALSE

```

Conditional Distribution

```
-----
Distribution      : norm
Includes Skew     : FALSE
Includes Shape    : FALSE
Includes Lambda   : FALSE
```

```
> garch01fit <- rugarch::ugarchfit(data =train.diff, spec = garch01)
> garch01fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(0,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate  Std. Error  t value Pr(>|t|)
ar1      0.02441    0.265617  0.09190 0.92678
ma1     -0.20130    0.257401 -0.78204 0.43419
omega   44.20093   28.343927  1.55945 0.11889
beta1    0.97227    0.017686 54.97470 0.00000
```

Robust Standard Errors:

```
      Estimate  Std. Error  t value Pr(>|t|)
ar1      0.02441    0.192021  0.12712 0.89884
ma1     -0.20130    0.165868 -1.21361 0.22490
omega   44.20093    6.400015  6.90638 0.00000
beta1    0.97227    0.002643 367.87088 0.00000
```

```
LogLikelihood : -1424.126
```

Information Criteria

```
-----
Akaike          10.311
Bayes           10.364
Shibata         10.311
Hannan-Quinn    10.332
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
                        statistic p-value
Lag[1]                  0.000736 0.9784
Lag[2*(p+q)+(p+q)-1][5] 0.816790 1.0000
Lag[4*(p+q)+(p+q)-1][9] 1.768608 0.9922
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                        statistic p-value
Lag[1]                  1.391 0.238172
Lag[2*(p+q)+(p+q)-1][2] 1.928 0.275037
Lag[4*(p+q)+(p+q)-1][5] 10.655 0.006243
d.o.f=1
```

Weighted ARCH LM Tests

```

-----
Statistic Shape Scale P-value
ARCH Lag[2]      1.058 0.500 2.000 0.303778
ARCH Lag[4]     10.960 1.397 1.611 0.002822
ARCH Lag[6]     12.970 2.222 1.500 0.002518

```

Nyblom stability test

```

-----
Joint Statistic: 1.0254
Individual Statistics:
ar1    0.5196
ma1    0.5007
omega  0.3399
beta1  0.3352

```

```

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:      1.07 1.24 1.6
Individual Statistic: 0.35 0.47 0.75

```

Sign Bias Test

```

-----
Sign Bias          t-value    prob sig
Negative Sign Bias 2.2636 0.02439 **
Positive Sign Bias 0.2905 0.77164
Joint Effect       5.6831 0.12809

```

Adjusted Pearson Goodness-of-Fit Test:

```

-----
group statistic p-value(g-1)
1    20      16.50      0.6236
2    30      33.29      0.2662
3    40      48.05      0.1517
4    50      49.53      0.4518

```

Elapsed time : 0.104223

```

> garch11 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garcho
rder=c(1,1)))
> garch11fit <- rugarch::ugarchfit(spec = garch11, data = train.diff)
> garch11fit

```

```

*-----*
*          GARCH Model Fit          *
*-----*

```

Conditional Variance Dynamics

```

-----
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm

```

Optimal Parameters

```

-----
Estimate Std. Error t value Pr(>|t|)
ar1      0.18611    0.322216   0.5776 0.563534
ma1     -0.30863    0.306416  -1.0072 0.313832
omega   275.54919  150.806722   1.8272 0.067675
alpha1   0.14653    0.073914   1.9824 0.047434
beta1    0.68924    0.132204   5.2134 0.000000

```

Robust Standard Errors:

	Estimate	Std. Error	t value	Pr(> t)
ar1	0.18611	0.225820	0.82416	0.409848
ma1	-0.30863	0.181183	-1.70339	0.088494
omega	275.54919	150.791352	1.82735	0.067647
alpha1	0.14653	0.068787	2.13016	0.033159
beta1	0.68924	0.122231	5.63880	0.000000

LogLikelihood : -1417.007

Information Criteria

Akaike	10.267
Bayes	10.333
Shibata	10.267
Hannan-Quinn	10.293

Weighted Ljung-Box Test on Standardized Residuals

	statistic	p-value
Lag[1]	0.4409	0.5067
Lag[2*(p+q)+(p+q)-1][5]	0.8673	1.0000
Lag[4*(p+q)+(p+q)-1][9]	1.4804	0.9973

d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals

	statistic	p-value
Lag[1]	0.4141	0.5199
Lag[2*(p+q)+(p+q)-1][5]	2.0465	0.6075
Lag[4*(p+q)+(p+q)-1][9]	3.1822	0.7285

d.o.f=2

Weighted ARCH LM Tests

	Statistic	Shape	Scale	P-Value
ARCH Lag[3]	2.016	0.500	2.000	0.1556
ARCH Lag[5]	2.916	1.440	1.667	0.3022
ARCH Lag[7]	3.399	2.315	1.543	0.4415

Nyblom stability test

Joint Statistic: 1.0726

Individual Statistics:

ar1	0.63724
ma1	0.58880
omega	0.10476
alpha1	0.08816
beta1	0.10473

Asymptotic Critical values (10% 5% 1%)

Joint Statistic: 1.28 1.47 1.88

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.3156	0.7526	
Negative Sign Bias	0.5378	0.5912	
Positive Sign Bias	0.9077	0.3648	
Joint Effect	1.7350	0.6292	

Adjusted Pearson Goodness-of-Fit Test:

	group	statistic	p-value(g-1)
1	20	17.80	0.535736
2	30	34.81	0.211013
3	40	65.96	0.004464
4	50	68.31	0.035459

Elapsed time : 0.1241832

```
> garch12 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "SGARCH",garcho
rder=c(1,2)))
> garch12fit <- rugarch::ugarchfit(spec = garch12, data = train.diff)
> garch12fit
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
GARCH Model      : SGARCH(1,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t)
ar1	0.086668	0.277123	0.31274	0.754477
ma1	-0.228200	0.265768	-0.85864	0.390537
omega	370.210090	204.602530	1.80941	0.070387
alpha1	0.149498	0.069573	2.14878	0.031652
beta1	0.029746	0.177871	0.16724	0.867184
beta2	0.593095	0.219570	2.70117	0.006910

Robust Standard Errors:

	Estimate	Std. Error	t value	Pr(> t)
ar1	0.086668	0.199961	0.43342	0.664707
ma1	-0.228200	0.161141	-1.41615	0.156731
omega	370.210090	192.483262	1.92334	0.054438
alpha1	0.149498	0.057746	2.58891	0.009628
beta1	0.029746	0.149399	0.19911	0.842179
beta2	0.593095	0.175914	3.37150	0.000748

LogLikelihood : -1417.171

Information Criteria

```
Akaike          10.276
Bayes           10.354
Shibata         10.275
Hannan-Quinn    10.307
```

Weighted Ljung-Box Test on Standardized Residuals

	statistic	p-value
Lag[1]	0.2612	0.6093
Lag[2*(p+q)+(p+q)-1][5]	0.5758	1.0000

```
Lag[4*(p+q)+(p+q)-1][9]    1.2177  0.9992
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                    statistic p-value
Lag[1]                0.2696  0.6036
Lag[2*(p+q)+(p+q)-1][8]  3.9790  0.5099
Lag[4*(p+q)+(p+q)-1][14]  5.2342  0.7400
d.o.f=3
```

Weighted ARCH LM Tests

```
-----
Statistic Shape Scale P-value
ARCH Lag[4]    0.1615 0.500 2.000 0.6878
ARCH Lag[6]    0.5083 1.461 1.711 0.8897
ARCH Lag[8]    1.4327 2.368 1.583 0.8521
```

Nyblom stability test

```
-----
Joint Statistic: 1.3251
Individual Statistics:
ar1    0.7269
ma1    0.6791
omega  0.1353
alpha1 0.0881
beta1  0.1459
beta2  0.1450
```

```
Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:    1.49 1.68 2.12
Individual Statistic: 0.35 0.47 0.75
```

Sign Bias Test

```
-----
t-value  prob sig
Sign Bias    0.1732 0.8626
Negative Sign Bias 0.4074 0.6840
Positive Sign Bias 0.8086 0.4195
Joint Effect  1.4815 0.6865
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1    20      27.48    0.09403
2    30      37.62    0.13100
3    40      44.01    0.26778
4    50      53.87    0.29354
```

Elapsed time : 0.1353309

```
> garch22 <- rugarch::ugarchspec(mean.model = list(armaOrder = c(1,1), include
.mean = FALSE),
+                               variance.model = list(model = "sGARCH",garchO
rder=c(2,2)))
> garch22fit <- rugarch::ugarchfit(spec = garch22, data = train.diff)
> garch22fit
```

```
*-----*
*                GARCH Model Fit                *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model      : SGARCH(2,2)
Mean Model       : ARFIMA(1,0,1)
Distribution      : norm
```

Optimal Parameters

```
-----
      Estimate  Std. Error  t value  Pr(>|t|)
ar1      0.226784    0.34564   0.656134  0.51174
ma1     -0.346963    0.32949  -1.053032  0.29233
omega  311.401382   207.85544   1.498163  0.13409
alpha1   0.093856    0.10018   0.936889  0.34882
alpha2   0.072562    0.11527   0.629493  0.52903
beta1    0.642887    0.63397   1.014064  0.31055
beta2    0.000011    0.46645   0.000024  0.99998
```

Robust Standard Errors:

```
      Estimate  Std. Error  t value  Pr(>|t|)
ar1      0.226784    0.22949   0.988189  0.323060
ma1     -0.346963    0.19397  -1.788777  0.073651
omega  311.401382   164.93916   1.887977  0.059029
alpha1   0.093856    0.13678   0.686158  0.492613
alpha2   0.072562    0.11427   0.634985  0.525438
beta1    0.642887    0.28798   2.232400  0.025589
beta2    0.000011    0.25551   0.000045  0.999964
```

LogLikelihood : -1416.631

Information Criteria

```
-----
Akaike          10.279
Bayes           10.371
Shibata         10.278
Hannan-Quinn    10.316
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
              statistic p-value
Lag[1]              0.4674  0.4942
Lag[2*(p+q)+(p+q)-1][5] 0.9122  1.0000
Lag[4*(p+q)+(p+q)-1][9] 1.5267  0.9967
d.o.f=2
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
              statistic p-value
Lag[1]              0.1632  0.6862
Lag[2*(p+q)+(p+q)-1][11] 2.4373  0.9271
Lag[4*(p+q)+(p+q)-1][19] 3.6037  0.9871
d.o.f=4
```

Weighted ARCH LM Tests

```
-----
      Statistic Shape Scale P-Value
ARCH Lag[5]    0.3948 0.500 2.000 0.5298
ARCH Lag[7]    0.7916 1.473 1.746 0.8161
ARCH Lag[9]    0.9252 2.402 1.619 0.9412
```

Nyblom stability test

```
-----
```

Joint Statistic: 1.7775

Individual Statistics:

ar1 0.62269
 ma1 0.59763
 omega 0.10903
 alpha1 0.09249
 alpha2 0.11352
 beta1 0.11046
 beta2 0.11535

Asymptotic Critical values (10% 5% 1%)

Joint Statistic: 1.69 1.9 2.35

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

	t-value	prob	sig
Sign Bias	0.3404	0.7338	
Negative Sign Bias	0.8794	0.3800	
Positive Sign Bias	0.7422	0.4586	
Joint Effect	2.0944	0.5530	

Adjusted Pearson Goodness-of-Fit Test:

	group	statistic	p-value(g-1)
1	20	17.08	0.58449
2	30	36.54	0.15842
3	40	48.05	0.15166
4	50	67.22	0.04293

Elapsed time : 0.161623

```
> #eksekusi peramalan dengan efek garch
> par(mfrow=c(1,1))
> forc11<- ugarchforecast(fitORspec = garch01fit, n.ahead = 120, n.roll = 0)
> plot(forc01, which= 1)
> forc11
```

```
*-----*
*          GARCH Model Forecast          *
*-----*
```

Model: sGARCH
 Horizon: 120
 Roll Steps: 0
 Out of Sample: 0

0-roll forecast [T0=278-01-01]:

	Series	Sigma
T+1	-7.568e+00	39.93
T+2	-1.847e-01	39.93
T+3	-4.510e-03	39.93
T+4	-1.101e-04	39.93
T+5	-2.687e-06	39.93
T+6	-6.559e-08	39.93
T+7	-1.601e-09	39.93
T+8	-3.909e-11	39.93
T+9	-9.541e-13	39.93
T+10	-2.329e-14	39.93
T+11	-5.685e-16	39.93
T+12	-1.388e-17	39.93
T+13	-3.387e-19	39.93
T+14	-8.269e-21	39.93

T+15	-2.018e-22	39.93
T+16	-4.927e-24	39.93
T+17	-1.203e-25	39.93
T+18	-2.936e-27	39.93
T+19	-7.167e-29	39.93
T+20	-1.749e-30	39.93
T+21	-4.270e-32	39.93
T+22	-1.042e-33	39.93
T+23	-2.544e-35	39.93
T+24	-6.211e-37	39.93
T+25	-1.516e-38	39.93
T+26	-3.701e-40	39.93
T+27	-9.034e-42	39.93
T+28	-2.205e-43	39.93
T+29	-5.383e-45	39.93
T+30	-1.314e-46	39.93
T+31	-3.208e-48	39.92
T+32	-7.830e-50	39.92
T+33	-1.911e-51	39.92
T+34	-4.665e-53	39.92
T+35	-1.139e-54	39.92
T+36	-2.780e-56	39.92
T+37	-6.786e-58	39.92
T+38	-1.656e-59	39.92
T+39	-4.044e-61	39.92
T+40	-9.870e-63	39.92
T+41	-2.409e-64	39.92
T+42	-5.881e-66	39.92
T+43	-1.436e-67	39.92
T+44	-3.504e-69	39.92
T+45	-8.554e-71	39.92
T+46	-2.088e-72	39.92
T+47	-5.097e-74	39.92
T+48	-1.244e-75	39.92
T+49	-3.037e-77	39.92
T+50	-7.414e-79	39.92
T+51	-1.810e-80	39.92
T+52	-4.418e-82	39.92
T+53	-1.078e-83	39.92
T+54	-2.632e-85	39.92
T+55	-6.426e-87	39.92
T+56	-1.569e-88	39.92
T+57	-3.829e-90	39.92
T+58	-9.346e-92	39.92
T+59	-2.281e-93	39.92
T+60	-5.569e-95	39.92
T+61	-1.359e-96	39.92
T+62	-3.318e-98	39.92
T+63	-8.100e-100	39.92
T+64	-1.977e-101	39.92
T+65	-4.827e-103	39.92
T+66	-1.178e-104	39.92
T+67	-2.876e-106	39.92
T+68	-7.020e-108	39.92
T+69	-1.714e-109	39.92
T+70	-4.183e-111	39.92
T+71	-1.021e-112	39.92
T+72	-2.493e-114	39.92
T+73	-6.084e-116	39.92
T+74	-1.485e-117	39.92
T+75	-3.625e-119	39.92
T+76	-8.850e-121	39.92
T+77	-2.160e-122	39.92
T+78	-5.273e-124	39.92

```

T+79 -1.287e-125 39.92
T+80 -3.142e-127 39.92
T+81 -7.670e-129 39.92
T+82 -1.872e-130 39.92
T+83 -4.570e-132 39.92
T+84 -1.116e-133 39.92
T+85 -2.723e-135 39.92
T+86 -6.648e-137 39.92
T+87 -1.623e-138 39.92
T+88 -3.961e-140 39.92
T+89 -9.669e-142 39.92
T+90 -2.360e-143 39.92
T+91 -5.761e-145 39.92
T+92 -1.406e-146 39.92
T+93 -3.433e-148 39.92
T+94 -8.380e-150 39.92
T+95 -2.046e-151 39.92
T+96 -4.993e-153 39.92
T+97 -1.219e-154 39.92
T+98 -2.975e-156 39.92
T+99 -7.263e-158 39.92
T+100 -1.773e-159 39.92
T+101 -4.328e-161 39.92
T+102 -1.056e-162 39.92
T+103 -2.579e-164 39.92
T+104 -6.295e-166 39.92
T+105 -1.537e-167 39.92
T+106 -3.751e-169 39.92
T+107 -9.155e-171 39.92
T+108 -2.235e-172 39.92
T+109 -5.455e-174 39.92
T+110 -1.332e-175 39.92
T+111 -3.251e-177 39.92
T+112 -7.935e-179 39.92
T+113 -1.937e-180 39.92
T+114 -4.728e-182 39.92
T+115 -1.154e-183 39.92
T+116 -2.817e-185 39.92
T+117 -6.877e-187 39.92
T+118 -1.679e-188 39.92
T+119 -4.098e-190 39.92
T+120 -1.000e-191 39.92

```

```

> hasil.forc.Diff11 <- forc11@forecast$seriesFor[,1]
> hasil.forc.Diff11

```

T+1	T+2	T+3	T+4
-7.568446e+00	-1.847477e-01	-4.509740e-03	-1.100839e-04
T+5	T+6	T+7	T+8
-2.687177e-06	-6.559469e-08	-1.601183e-09	-3.908530e-11
T+9	T+10	T+11	T+12
-9.540823e-13	-2.328940e-14	-5.685002e-16	-1.387724e-17
T+13	T+14	T+15	T+16
-3.387469e-19	-8.268901e-21	-2.018460e-22	-4.927113e-24
T+17	T+18	T+19	T+20
-1.202721e-25	-2.935873e-27	-7.166543e-29	-1.749372e-30
T+21	T+22	T+23	T+24
-4.270262e-32	-1.042382e-33	-2.544481e-35	-6.211145e-37
T+25	T+26	T+27	T+28
-1.516157e-38	-3.700977e-40	-9.034181e-42	-2.205267e-43
T+29	T+30	T+31	T+32
-5.383114e-45	-1.314032e-46	-3.207586e-48	-7.829801e-50
T+33	T+34	T+35	T+36
-1.911275e-51	-4.665471e-53	-1.138854e-54	-2.779971e-56
T+37	T+38	T+39	T+40

```

-6.785982e-58 -1.656476e-59 -4.043500e-61 -9.870288e-63
      T+41      T+42      T+43      T+44
-2.409363e-64 -5.881317e-66 -1.435645e-67 -3.504446e-69
      T+45      T+46      T+47      T+48
-8.554444e-71 -2.088162e-72 -5.097257e-74 -1.244254e-75
      T+49      T+50      T+51      T+52
-3.037256e-77 -7.414019e-79 -1.809781e-80 -4.417723e-82
      T+53      T+54      T+55      T+56
-1.078378e-83 -2.632348e-85 -6.425629e-87 -1.568513e-88
      T+57      T+58      T+59      T+60
-3.828780e-90 -9.346151e-92 -2.281420e-93 -5.569005e-95
      T+61      T+62      T+63      T+64
-1.359408e-96 -3.318351e-98 -8.100181e-100 -1.977275e-101
      T+65      T+66      T+67      T+68
-4.826580e-103 -1.178181e-104 -2.875970e-106 -7.020316e-108
      T+69      T+70      T+71      T+72
-1.713677e-109 -4.183131e-111 -1.021113e-112 -2.492564e-114
      T+73      T+74      T+75      T+76
-6.084412e-116 -1.485221e-117 -3.625462e-119 -8.849847e-121
      T+77      T+78      T+79      T+80
-2.160271e-122 -5.273277e-124 -1.287220e-125 -3.142139e-127
      T+81      T+82      T+83      T+84
-7.670042e-129 -1.872277e-130 -4.570277e-132 -1.115616e-133
      T+85      T+86      T+87      T+88
-2.723248e-135 -6.647520e-137 -1.622677e-138 -3.960996e-140
      T+89      T+90      T+91      T+92
-9.668894e-142 -2.360202e-143 -5.761315e-145 -1.406352e-146
      T+93      T+94      T+95      T+96
-3.432941e-148 -8.379898e-150 -2.045555e-151 -4.993252e-153
      T+97      T+98      T+99      T+100
-1.218866e-154 -2.975283e-156 -7.262743e-158 -1.772854e-159
      T+101     T+102     T+103     T+104
-4.327584e-161 -1.056374e-162 -2.578637e-164 -6.294520e-166
      T+105     T+106     T+107     T+108
-1.536509e-167 -3.750657e-169 -9.155452e-171 -2.234870e-172
      T+109     T+110     T+111     T+112
-5.455374e-174 -1.331671e-175 -3.250643e-177 -7.934905e-179
      T+113     T+114     T+115     T+116
-1.936931e-180 -4.728098e-182 -1.154141e-183 -2.817288e-185
      T+117     T+118     T+119     T+120
-6.877073e-187 -1.678711e-188 -4.097778e-190 -1.000278e-191
> hasilgarch11 <- diffinv(hasil.forc.Diff11, differences = 1)+ pt_1
> hasilgarch11
[1] 53708908.992      7308.542      7304.154      7268.991      7239.702
[6]      7322.043      7366.206      7374.149      7413.449      7425.557
[11]      7320.296      7294.691      7328.988      7323.370      7330.595
[16]      7342.394      7370.002      7357.906      7302.334      7281.055
[21]      7197.303      7229.226      7159.086      7246.641      7279.124
[26]      7157.049      7123.083      7159.056      7079.559      7066.062
[31]      7103.055      7166.775      7147.536      7028.317      7148.026
[36]      7226.439      7109.667      7126.966      7128.132      7115.854
[41]      7081.037      7091.503      7076.005      7172.073      7238.938
[46]      7309.480      7258.933      7178.279      7214.624      7168.662
[51]      7245.868      7132.471      7026.384      6962.978      7028.433
[56]      7091.554      6939.912      6967.139      6890.192      6913.790
[61]      6847.933      6842.339      6823.806      6727.074      6719.161
[66]      6811.563      6872.220      6881.407      6874.946      6897.884
[71]      6960.193      7055.819      7131.868      7117.384      7188.997
[76]      7213.131      7245.614      7243.219      7262.043      7279.284
[81]      7292.649      7319.822      7271.105      7216.535      7216.460
[86]      7313.313      7286.737      7314.218      7306.099      7255.001
[91]      7232.519      7280.409      7281.139      7234.106      7248.004
[96]      7318.227      7300.365      7051.895      7121.457      7204.373
[101]      7187.363      7249.238      7289.867      7348.880      7428.281

```

```

[106]      7401.743      7424.332      7459.073      7526.227      7546.835
[111]      7480.918      7536.540      7598.437      7590.123      7651.117
[116]      7619.846      7662.975      7686.772      7608.763      7665.138
[121]      7673.285
> #pembuatan tabel perbandingan train dan testing
> perbandingan11 <- data.frame("Aktual"= testing.ts,
+                               "Ramalan" = hasilgarch11[-1])
> head(perbandingan11,10)
      Aktual  Ramalan
1  7316.111  7308.542
2  7311.907  7304.154
3  7276.749  7268.991
4  7247.460  7239.702
5  7329.801  7322.043
6  7373.964  7366.206
7  7381.907  7374.149
8  7421.207  7413.449
9  7433.315  7425.557
10 7328.054  7320.296
> str(perbandingan11)
'data.frame': 120 obs. of  2 variables:
 $ Aktual: Time-Series from 279 to 398: 7316 7312 7277 7247 7330 ...
 $ Ramalan: num 7309 7304 7269 7240 7322 ...
> dim(perbandingan11)
[1] 120  2
> #perbandingan
> # Menulis data perbandingan ke file CSV
> write.csv(perbandingan11, file = "C:/Users/asus/Dropbox/Penelitian UII/Anali
sis Data/perbandingan Setelah.csv", row.names = FALSE)
> #MAPE PERBANDINGAN 12
> Date<-c(datasaham$Date[279:398])
> dataframe <- data.frame(Date, perbandingan11)
> T <- nrow(dataframe)
> MAPE <- 1/T*sum(abs((dataframe$Aktual-dataframe$Ramalan)/dataframe$Aktual)*1
00)
> MAPE
[1] 0.1073281
> length(Date)
[1] 120
> coef(model2)
      ar1      ma1
0.02535456 -0.19866759
> # Menghitung Error (Selisih Aktual dan Ramalan)
> error <- perbandingan11$Aktual - perbandingan11$Ramalan
> # Menghitung Mean Squared Error (MSE)
> MSE <- mean(error^2)
> # Menghitung Root Mean Squared Error (RMSE)
> RMSE <- sqrt(MSE)
> # Menampilkan hasil
> cat("Mean Squared Error (MSE):", MSE, "\n")
Mean Squared Error (MSE): 60.15892
> cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
Root Mean Squared Error (RMSE): 7.756218
> # Mengonversi kedua kolom ke dalam format time series
> # Mengonversi perbandingan12[,1] (data aktual) ke time series
> ts_aktual <- ts(perbandingan11[, 1], frequency = 1) # Gantilah frequency se
suai data (1 untuk data harian)
> # Mengonversi perbandingan12[,2] (hasil forecast) ke time series
> ts_forecast <- ts(perbandingan11[, 2], frequency = 1) # Gantilah frequency
sesuai data
> # Membuat plot perbandingan antara data aktual dan hasil forecast
> plot(ts_aktual, type = "l", col = "yellow", lty = 1, # Garis solid untuk dat
a aktual
+       xlab = "Observasi", ylab = "Nilai",

```

```

+     main = "Perbandingan Data Aktual dan Hasil Forecast",
+     lwd = 4) # Rentang Y agar kedua data terlihat
> # Menambahkan garis untuk hasil forecast
> lines(ts_forecast, col = "blue", lty = 3, lwd = 2) # Garis putus-putus untuk
hasil forecast
> # Menambahkan legend untuk membedakan data
> legend("bottomleft", legend = c("Data Aktual", "Hasil Forecast"),
+       col = c("yellow", "blue"), lty = c(1, 2), lwd = 2)

```

```

▶ import pandas as pd
import numpy as np

from datetime import datetime
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot as plt
from pandas import read_excel, DataFrame, concat
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.metrics import mean_squared_error

from keras.models import Sequential, load_model
from keras.layers import Dense, LSTM

import math
import pandas_datareader as web

plt.style.use("fivethirtyeight")

```

```

[ ] file_path = '/content/Sebelum Covid.csv' # Replace with your file path
df = pd.read_csv(file_path) # Use pd.read_csv to read CSV files

```

```

[ ] df.info()

```

```

▶ df.info()

```

```

↔ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 527 entries, 0 to 526
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        527 non-null   object
1   Open        527 non-null   float64
2   High        527 non-null   float64
3   Low         527 non-null   float64
4   Close       527 non-null   float64
5   Adj Close   527 non-null   float64
6   Volume      527 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 28.9+ KB

```

```
df.describe()
```



	Open	High	Low	Close	Adj Close	Volume
count	527.000000	527.000000	527.000000	527.000000	527.000000	5.270000e+02
mean	6185.172470	6211.327700	6151.015391	6181.092105	6181.092105	8.092720e+07
std	244.388702	238.691816	248.842003	244.290436	244.290436	2.859127e+07
min	5436.171875	5456.275879	5288.370117	5452.704102	5452.704102	0.000000e+00
25%	5977.517578	6013.781495	5943.386475	5979.697266	5979.697266	5.899035e+07
50%	6213.932129	6240.570801	6180.836914	6210.698242	6210.698242	7.760430e+07
75%	6374.117432	6392.671631	6349.342041	6373.411133	6373.411133	1.030736e+08
max	6687.230957	6693.465820	6634.888184	6689.287109	6689.287109	1.792013e+08

```
df_close=df['Close']
df_close
```



	Close
0	6339.237793
1	6251.479004
2	6292.320801
3	6353.737793
4	6385.403809
...	...
522	5807.049805
523	5787.138184
524	5688.920898
525	5535.693848
526	5452.704102

527 rows × 1 columns

dtype: float64

```
df_close.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 527 entries, 0 to 526
Series name: Close
Non-Null Count  Dtype
-----
527 non-null    float64
dtypes: float64(1)
memory usage: 4.2 KB
```

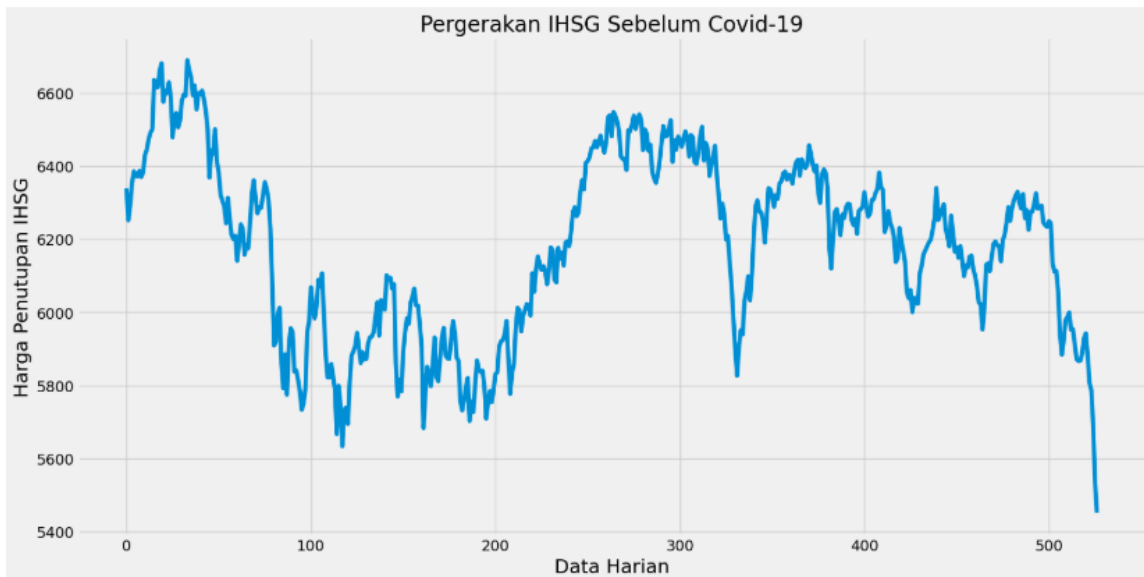
[+ Code](#)
[+ Text](#)

```
[ ] df_close = df['close'].replace('null', np.nan)
df_close = pd.DataFrame(df_close) # Convert Series to DataFrame
print(df_close.shape)
```

```
(527, 1)
```

[+ Code](#)
[+ Text](#)

```
plt.figure(figsize=(16, 8))
plt.title('Pergerakan IHSG Sebelum Covid-19')
plt.plot(df_close)
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Harga Penutupan IHSG', fontsize=18)
plt.show()
```




```
data=df_close.filter(['Close'])
dataset=data.values
training_data_len=math.ceil(len(dataset)*.7)
training_data_len
```

↗ 369

+ Code

+ Text

```
[ ] testing_data_len = math.ceil(len(dataset) * 0.3) # Use math.ceil to round up
testing_data_len
```

↗ 159

```
test=df_close[training_data_len:].reset_index(drop=True)
test
```

↗

Close

0 6403.293945

1 6456.539063

2 6433.546875

3 6403.810059

4 6384.986816

...

153 5807.049805

154 5787.138184

155 5688.920898

156 5535.693848

157 5452.704102

158 rows × 1 columns


```
[ ] train_data=scaled_data[0:training_data_len, :]
    x_train=[]
    y_train=[]
    for i in range(60, len(train_data)):
        x_train.append(train_data[i-60:i, 0])
        y_train.append(train_data[i, 0])
```

```
[ ] x_train, y_train=np.array(x_train), np.array(y_train)
    x_train=np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
```

[+ Code](#)
[+ Text](#)

```
[ ] x_train.shape[0]
```

 309

```
[ ] x_train
```

 array([[0.7169221],
 [0.64595332],
 [0.67898127],
 ...,
 [0.61297473],
 [0.60446227],
 [0.61188452]],
 [[0.64595332],
 [0.67898127],
 [0.72864796],

```

▶ model=Sequential()
  model.add(LSTM(units=20,return_sequences=True,input_shape=(x_train.shape[1],1)))
  model.add(LSTM(units=20,return_sequences=False))
  model.add(Dense(units=1))

```

```

↔ /usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning:
  super().__init__(**kwargs)

```

```

[ ] model.compile(optimizer='adam', loss='mean_squared_error')

```

```

[ ] model.fit(x_train, y_train, batch_size=16, epochs=100, validation_split=0.3)

```

```

↔ Epoch 1/100
  14/14 ————— 5s 77ms/step - loss: 0.1911 - val_loss: 0.0180
Epoch 2/100
  14/14 ————— 1s 51ms/step - loss: 0.0183 - val_loss: 0.0131
Epoch 3/100
  14/14 ————— 1s 50ms/step - loss: 0.0109 - val_loss: 0.0101
Epoch 4/100
  14/14 ————— 1s 52ms/step - loss: 0.0106 - val_loss: 0.0110
Epoch 5/100
  14/14 ————— 1s 45ms/step - loss: 0.0103 - val_loss: 0.0092
Epoch 6/100
  14/14 ————— 1s 47ms/step - loss: 0.0079 - val_loss: 0.0098
Epoch 7/100

```

```

▶ test_data=scaled_data[training_data_len-60:, :]
  x_test=[]
  y_test=dataset[training_data_len:, :]
  for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

```

[+ Code](#)
[+ Text](#)

```

[ ] x_test=np.array(x_test)
    x_test=np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

```

```

[ ] predictions_normalised=model.predict(x_test)
    predictions_normalised

```

```

↔ WARNING:tensorflow:5 out of the last 11 calls to <function TensorFlowTrainer.make_pr
1/5 ————— 1s 341ms/stepWARNING:tensorflow:5 out of the last 11 calls
5/5 ————— 1s 104ms/step

```

```

array([[0.7623301 ],
       [0.7683483 ],
       [0.8020202 ],
       [0.80436677],
       [0.78602   ],
       [0.7678166 ],
       [0.77138305],
       [0.7332439 ],
       [0.70325553],
       [0.7368399 ],
       [0.7610656 ]])

```

```

▶ predictions= scaler.inverse_transform(predictions_normalised)
predictions

```

```

↩ array([[6395.3887],
         [6402.8306],
         [6444.4683],
         [6447.37   ],
         [6424.6826],
         [6402.173  ],
         [6406.5835],
         [6359.421  ],
         [6322.338  ],
         [6363.8677],
         [6393.8247],
         [6396.648  ],
         [6366.195  ],
         [6241.8506],
         [6156.478  ],
         [6185.833  ],
         [6250.205  ],
         [6281.3623],
         [6269.0083],
         [6234.114  ],
         [6254.9473],
         [6260.4463],
         [6280.425  ],
         [6295.5483],
         [6300.107  ],

```

```

[ ] mse=np.mean(predictions-y_test)**2
mse

```

```

↩ 14.747889508105136

```

```

▶ mape=np.mean(np.abs((predictions-y_test)/y_test))*100
mape

```

```

↩ 0.7448151788318933

```

```

[ ] rmse=np.sqrt(np.mean((predictions-y_test)**2))
rmse

```

```

↩ 61.149311135528855

```

[+ Code](#)
[+ Text](#)

```

train=data[:training_data_len]
valid=data[training_data_len:]
valid['Predictions']=predictions
plt.figure(figsize=(16,8))
plt.title('Prediksi IHSG Sebelum Covid-19 dengan LSTM')
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Close Price', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Valid', 'Predictions'], loc='lower right')
plt.show()

```

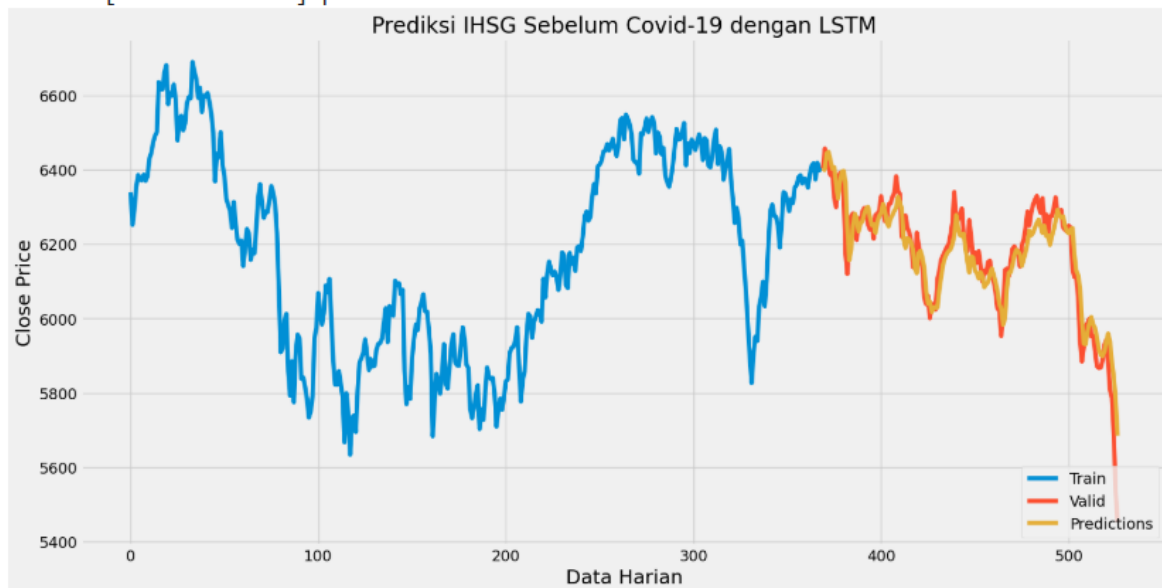
 <ipython-input-54-ed4e97449456>:3: SettingWithCopyWarning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/using_indexers.html
 valid['Predictions']=predictions

Prediksi IHSG Sebelum Covid-19 dengan LSTM

 <ipython-input-54-ed4e97449456>:3: SettingWithCopyWarning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/using_indexers.html
 valid['Predictions']=predictions



```

▶ # Periksa prediksi dan nilai aktual
for i in range(10): # Cetak 10 data pertama
    print(f"Prediksi: {predictions[i]}, Aktual: {y_test[i]}")

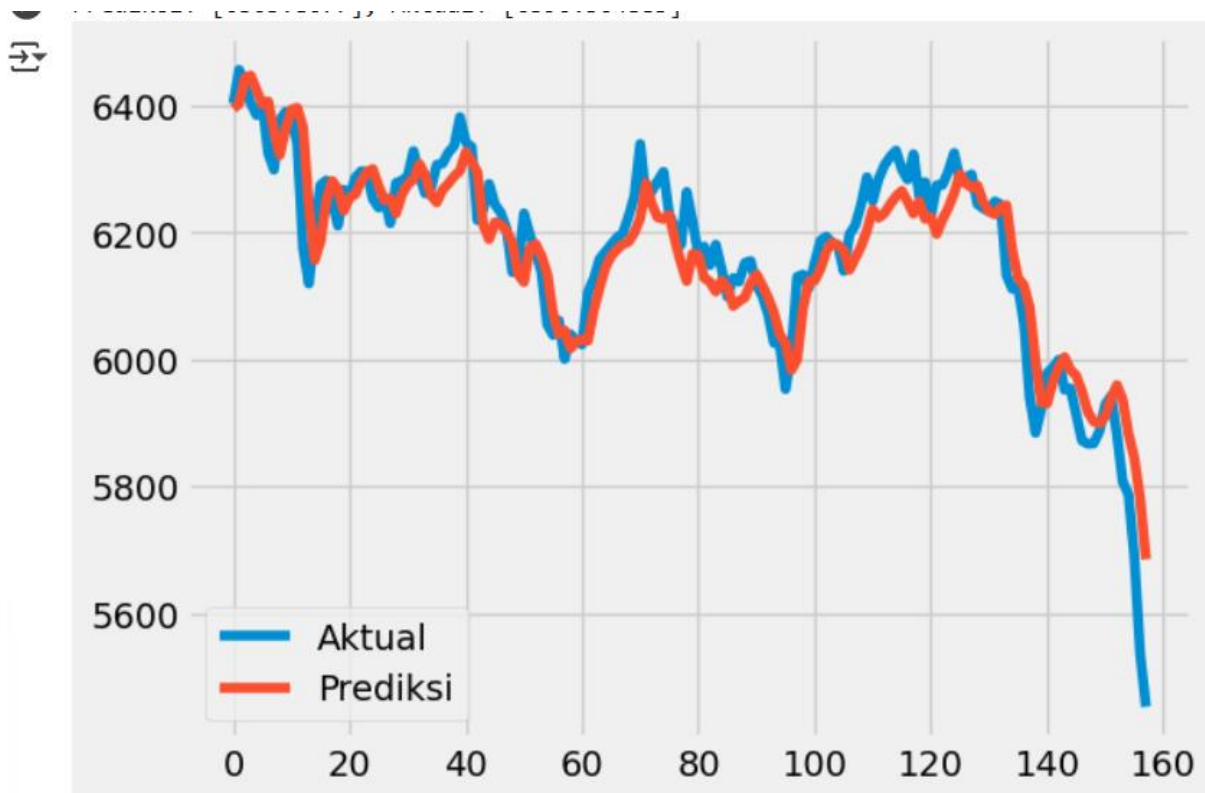
# Visualisasi data untuk cek outlier
plt.plot(y_test, label="Aktual")
plt.plot(predictions, label="Prediksi")
plt.legend()
plt.show()

```

```

⇨ Prediksi: [6395.3887], Aktual: [6403.293945]
Prediksi: [6402.8306], Aktual: [6456.539063]
Prediksi: [6444.4683], Aktual: [6433.546875]
Prediksi: [6447.37], Aktual: [6403.810059]
Prediksi: [6424.6826], Aktual: [6384.986816]
Prediksi: [6402.173], Aktual: [6401.365234]
Prediksi: [6406.5835], Aktual: [6325.236816]
Prediksi: [6359.421], Aktual: [6299.035156]
Prediksi: [6322.338], Aktual: [6376.996094]
Prediksi: [6363.8677], Aktual: [6390.504883]

```





valid



	Close	Predictions
369	6403.293945	6395.388672
370	6456.539063	6402.830566
371	6433.546875	6444.468262
372	6403.810059	6447.370117
373	6384.986816	6424.682617
...	...	...
522	5807.049805	5936.431641
523	5787.138184	5882.555664
524	5688.920898	5844.420898
525	5535.693848	5781.041016
526	5452.704102	5684.885742

158 rows × 2 columns

Lampiran 6. Sintaks Python Model LSTM Masa Covid-19

```

▶ import pandas as pd
import numpy as np

from datetime import datetime
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot as plt
from pandas import read_excel, DataFrame, concat
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.metrics import mean_squared_error

from keras.models import Sequential, load_model
from keras.layers import Dense, LSTM

import math
import pandas_datareader as web

plt.style.use("fivethirtyeight")

```

```

[ ] file_path = '/content/Masa Covid.csv' # Replace with your file path
df = pd.read_csv(file_path) # Use pd.read_csv to read CSV files

```

```

▶ df.info()

```

```

↔ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 693 entries, 0 to 692
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        693 non-null   object
1   Open        693 non-null   float64
2   High        693 non-null   float64
3   Low         693 non-null   float64
4   Close       693 non-null   float64
5   Adj Close   693 non-null   float64
6   Volume      693 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 38.0+ KB

```

 `df.describe()`



	Open	High	Low	Close	Adj Close	Volume
count	693.000000	693.000000	693.000000	693.000000	693.000000	6.930000e+02
mean	6158.972781	6197.101692	6114.846417	6156.889983	6156.889983	1.661147e+08
std	809.834567	805.482852	812.876086	809.002572	809.002572	6.876410e+07
min	3937.632080	4123.562012	3911.716064	3937.632080	3937.632080	0.000000e+00
25%	5583.330078	5651.854980	5563.654785	5612.415039	5612.415039	1.208111e+08
50%	6232.102051	6275.742188	6184.517090	6227.728027	6227.728027	1.730795e+08
75%	6839.016113	6893.375977	6809.715820	6850.198242	6850.198242	2.108375e+08
max	7317.895996	7377.495117	7278.198242	7318.016113	7318.016113	3.961367e+08



`df_close=df['Close']`
`df_close`



	Close
0	5361.246094
1	5518.627930
2	5650.136230
3	5638.129883
4	5498.540039
...	...
688	6835.808105
689	6923.026855
690	6850.520020
691	6860.077148
692	6850.619141

693 rows × 1 columns

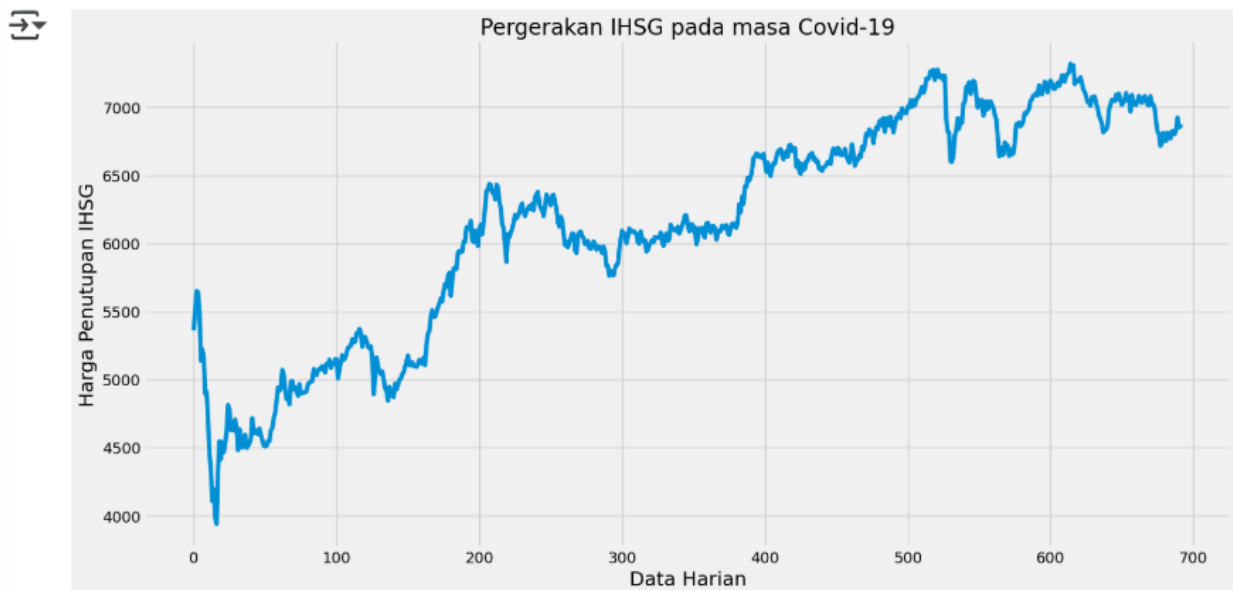
```
df_close.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 693 entries, 0 to 692
Series name: Close
Non-Null Count  Dtype
-----
693 non-null    float64
dtypes: float64(1)
memory usage: 5.5 KB
```

```
[ ] df_close = df['Close'].replace('null', np.nan)
df_close = pd.DataFrame(df_close) # Convert Series to DataFrame
print(df_close.shape)
```

```
(693, 1)
```

```
plt.figure(figsize=(16, 8))
plt.title('Pergerakan IHSG pada masa Covid-19')
plt.plot(df_close)
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Harga Penutupan IHSG', fontsize=18)
plt.show()
```



```
[ ] data=df_close.filter(['Close'])
    dataset=data.values
    training_data_len=math.ceil(len(dataset)*.7)
    training_data_len
```

↔ 486

```
▶ testing_data_len = math.ceil(len(dataset) * 0.3) # Use math.ceil to round up
testing_data_len
```

↔ 208

[+ Code](#)

[+ Text](#)

```
[ ] test=df_close[training_data_len:].reset_index(drop=True)
test
```

↔

	close
0	6921.440918
1	6868.401855
2	6928.328125
3	6869.065918
4	6814.183105
...	...
202	6835.808105
203	6923.026855
204	6850.520020
205	6860.077148
206	6850.619141

207 rows × 1 columns

```
▶ scaler=MinMaxScaler(feature_range=(0,1))  
scaled_data=scaler.fit_transform(dataset)  
scaled_data
```

```
↵ array([[0.42113973],  
         [0.46769711],  
         [0.50660047],  
         [0.5030487 ],  
         [0.46175462],  
         [0.35474579],  
         [0.37960009],  
         [0.35986234],  
         [0.28343406],  
         [0.28693152],  
         [0.22276319],  
         [0.15356745],  
         [0.11627133],  
         [0.04963631],  
         [0.07611909],  
         [0.01534885],  
         [0.          ],  
         [0.11870596],  
         [0.17984309],  
         [0.14106916],  
         [0.17787863],  
         [0.15631509],  
         [0.17573535],  
         [0.20287551],  
         [0.25860821],
```

```

▶ train_data=scaled_data[0:training_data_len, :]
  x_train=[]
  y_train=[]
  for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])

```

[+ Code](#)
[+ Text](#)

```

[ ] x_train, y_train=np.array(x_train), np.array(y_train)
    x_train=np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

```

[+ Code](#)
[+ Text](#)

```

[ ] x_train.shape[0]

```

```

↔ 426

```

```

[ ] x_train

```

```

↔ array([[0.42113973],
         [0.46769711],
         [0.50660047],
         ...,
         [0.2413867 ],
         [0.26916313],
         [0.29682242]],

        [[0.46769711],
         [0.50660047],
         [0.5030487 ]],

```

```
[ ] model=Sequential()
    model.add(LSTM(units=30,return_sequences=True,input_shape=(x_train.shape[1],1)))
    model.add(LSTM(units=30,return_sequences=False))
    model.add(Dense(units=1))
```

 /usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: super().__init__(**kwargs)

```
[ ] model.compile(optimizer='adam', loss='mean_squared_error')
```

 `model.fit(x_train, y_train, batch_size=16, epochs=100, validation_split=0.3)`

 Epoch 1/100
19/19  1s 49ms/step - loss: 5.8243e-04 - val_loss: 0.0012
 Epoch 2/100
19/19  1s 47ms/step - loss: 6.6814e-04 - val_loss: 4.0519e-04
 Epoch 3/100
19/19  1s 51ms/step - loss: 6.0417e-04 - val_loss: 5.6796e-04
 Epoch 4/100
19/19  1s 71ms/step - loss: 4.8061e-04 - val_loss: 2.7812e-04
 Epoch 5/100
19/19  2s 83ms/step - loss: 4.1187e-04 - val_loss: 7.9682e-04
 Epoch 6/100
19/19  1s 47ms/step - loss: 4.9808e-04 - val_loss: 0.0011
 Epoch 7/100
19/19  1s 47ms/step - loss: 5.2798e-04 - val_loss: 0.0012
 Epoch 8/100
19/19  1s 47ms/step - loss: 5.4399e-04 - val_loss: 7.1381e-04


```

▶ test_data=scaled_data[training_data_len-60:, :]
  x_test=[]
  y_test=dataset[training_data_len:, :]
  for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

```

[+ Code](#)
[+ Text](#)

```

[ ] x_test=np.array(x_test)
    x_test=np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

```

[+ Code](#)
[+ Text](#)

```

[ ] predictions_normalised=model.predict(x_test)
    predictions_normalised

```

```

↔ 7/7 ————— 0s 22ms/step
array([[0.8532893 ],
       [0.8626672 ],
       [0.857085  ],
       [0.86527514],
       [0.8579646 ],
       [0.8443734 ],
       [0.84834635],
       [0.86201036],
       [0.8678412 ],
       [0.8740642 ]])

```

```

▶ predictions= scaler.inverse_transform(predictions_normalised)
  predictions

```

```

↔ array([[6822.078 ],
       [6853.7783],
       [6834.9087],
       [6862.5947],
       [6837.882 ],
       [6791.939 ],
       [6805.368 ],
       [6851.558 ],
       [6871.269 ],
       [6895.009 ],
       [6880.5674],
       [6915.957 ],
       [6911.479 ],
       [6900.2437],
       [6894.062 ],
       [6919.2896],
       [6927.064 ],
       [6961.6445],

```

```

▶ mse=np.mean(predictions-y_test)**2
mse

```

```

↔ 5105.110188644202

```

[+ Code](#)
[+ Text](#)

```

[ ] mape=np.mean(np.abs((predictions-y_test)/y_test))*100
mape

```

```

↔ 1.1573946025459934

```

[+ Code](#)
[+ Text](#)

```

[ ] rmse=np.sqrt(np.mean((predictions-y_test)**2))
rmse

```

```

↔ 94.16673544251704

```

```

▶ train=data[:training_data_len]
valid=data[training_data_len:]
valid['Predictions']=predictions
plt.figure(figsize=(16,8))
plt.title('Prediksi IHSG pada Masa Covid-19 dengan LSTM')
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Close Price', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Valid', 'Predictions'], loc='lower right')
plt.show()

```

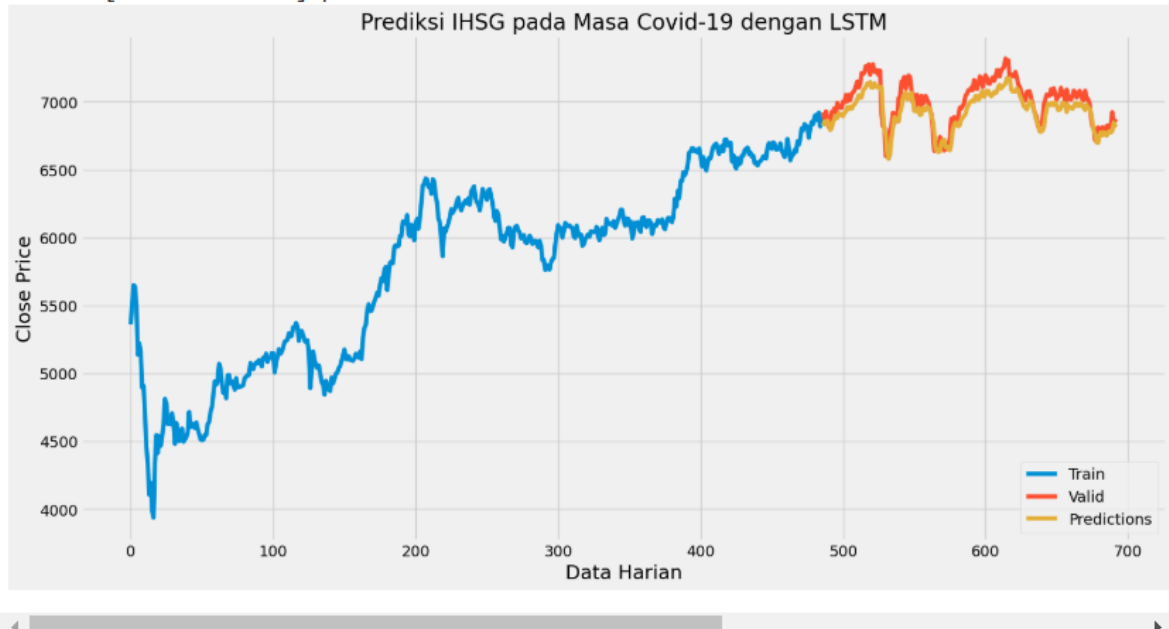
```

↔ <ipython-input-44-db1c24bf1e1a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min#integer-location-indexing-guide>

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html#using-valid-predictions>
`valid['Predictions']=predictions`

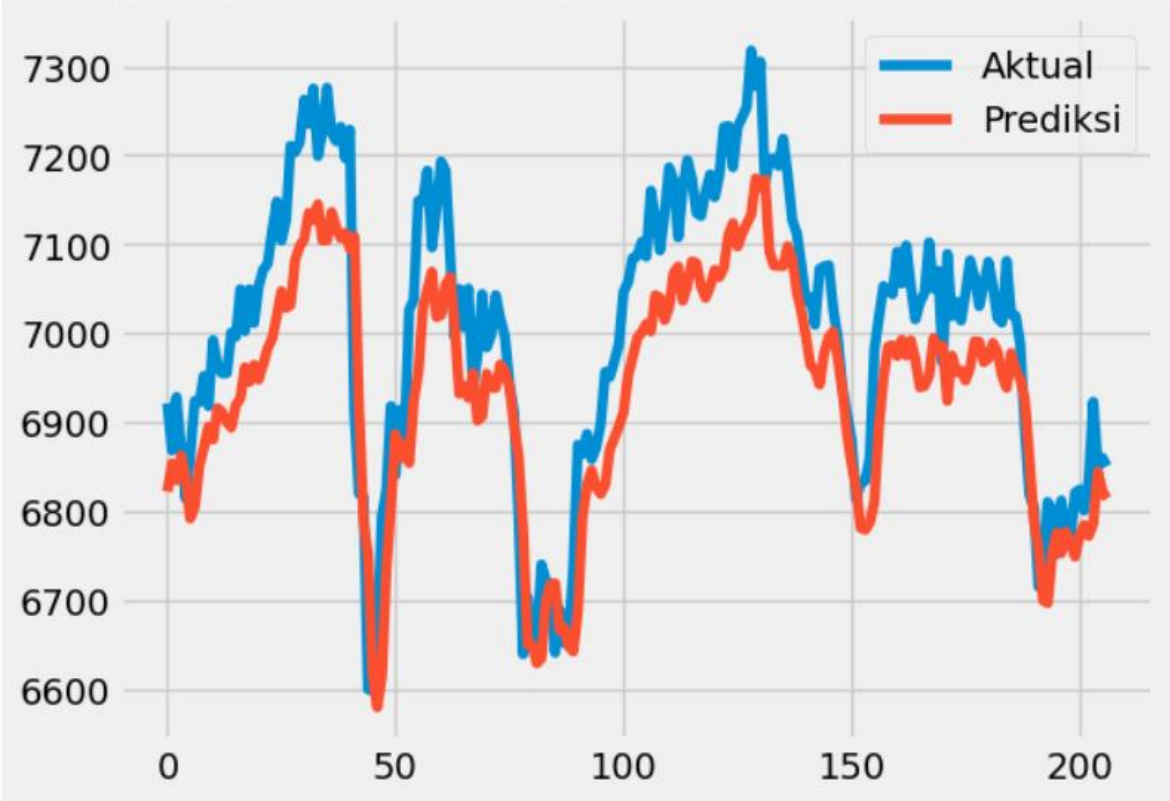


```
# Periksa prediksi dan nilai aktual
for i in range(10): # Cetak 10 data pertama
    print(f"Prediksi: {predictions[i]}, Aktual: {y_test[i]}")

# Visualisasi data untuk cek outlier
plt.plot(y_test, label="Aktual")
plt.plot(predictions, label="Prediksi")
plt.legend()
plt.show()
```

```
Prediksi: [6822.078], Aktual: [6921.440918]
Prediksi: [6853.7783], Aktual: [6868.401855]
Prediksi: [6834.9087], Aktual: [6928.328125]
Prediksi: [6862.5947], Aktual: [6869.065918]
Prediksi: [6837.882], Aktual: [6814.183105]
Prediksi: [6791.939], Aktual: [6864.440918]
Prediksi: [6805.368], Aktual: [6924.007813]
Prediksi: [6851.558], Aktual: [6922.602051]
Prediksi: [6871.269], Aktual: [6952.204102]
Prediksi: [6895.009], Aktual: [6918.185059]
```

▶ Prediksi: [6895.009], Aktual: [6918.185059]



▶ valid



	Close	Predictions
486	6921.440918	6822.078125
487	6868.401855	6853.778320
488	6928.328125	6834.908691
489	6869.065918	6862.594727
490	6814.183105	6837.881836
...	...	...
688	6835.808105	6772.299316
689	6923.026855	6786.294922
690	6850.520020	6844.598633
691	6860.077148	6822.718750
692	6850.619141	6814.166016

207 rows × 2 columns

Lampiran 7. Sintaks Python Model LSTM Setelah Covid-19

Add text cell

```

import pandas as pd
import numpy as np

from datetime import datetime
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot as plt
from pandas import read_excel, DataFrame, concat
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.metrics import mean_squared_error

from keras.models import Sequential, load_model
from keras.layers import Dense, LSTM

import math
import pandas_datareader as web

plt.style.use("fivethirtyeight")

[ ] file_path = '/content/Setelah Covid.csv' # Replace with your file path
df = pd.read_csv(file_path) # Use pd.read_csv to read CSV files

```

df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    Date        398 non-null    object
1    Open         398 non-null    float64
2    High         398 non-null    float64
3    Low          398 non-null    float64
4    Close        398 non-null    float64
5    Adj Close    398 non-null    float64
6    Volume       398 non-null    int64
dtypes: float64(5), int64(1), object(1)
memory usage: 21.9+ KB

```

 `df.describe()`



	Open	High	Low	Close	Adj Close	Volume
count	398.000000	398.000000	398.000000	398.000000	398.000000	3.980000e+02
mean	7010.481055	7044.532780	6975.370851	7009.387971	7009.387971	1.610887e+08
std	244.211972	244.097506	242.999127	244.063330	244.063330	4.875133e+07
min	6565.728027	6622.794922	6542.791992	6565.728027	6565.728027	0.000000e+00
25%	6829.975952	6861.038940	6801.372192	6827.863648	6827.863648	1.359838e+08
50%	6939.753662	6970.722168	6913.584473	6940.008057	6940.008057	1.524306e+08
75%	7228.282837	7262.028687	7185.972900	7223.596192	7223.596192	1.735118e+08
max	7706.650879	7726.662109	7669.921875	7694.529785	7694.529785	6.150719e+08

 `df_close=df['Close']`
`df_close`



	Close
0	6850.983887
1	6888.757813
2	6813.238770
3	6653.840820
4	6684.558105
...	...
393	7670.732910
394	7694.529785
395	7616.520996
396	7672.895996
397	7681.042969

398 rows × 1 columns

dtype: float64

df_close.info()

```
<class 'pandas.core.series.Series'>
RangeIndex: 398 entries, 0 to 397
Series name: Close
Non-Null Count  Dtype
-----
398 non-null    float64
dtypes: float64(1)
memory usage: 3.2 KB
```

+ Code

+ Text

```
[ ] df_close = df['close'].replace('null', np.nan)
df_close = pd.DataFrame(df_close) # Convert Series to DataFrame
print(df_close.shape)
```

(398, 1)

```
plt.figure(figsize=(16, 8))
plt.title('Pergerakan IHSG Setelah Covid-19')
plt.plot(df_close)
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Harga Penutupan IHSG', fontsize=18)
plt.show()
```



```
data=df_close.filter(['Close'])
dataset=data.values
training_data_len=math.ceil(len(dataset)*.7)
training_data_len
```

279

+ Code

+ Text

```
[ ] testing_data_len = math.ceil(len(dataset) * 0.3) # Use math.ceil to round up
testing_data_len
```

120

+ Code

+ Text

```
[ ] test=df_close[training_data_len:].reset_index(drop=True)
test
```



Close

0	7311.907227
1	7276.749023
2	7247.459961
3	7329.800781
4	7373.963867
...	...
114	7670.732910
115	7694.529785
116	7616.520996
117	7672.895996
118	7681.042969

119 rows × 1 columns


```
▶ scaler=MinMaxScaler(feature_range=(0,1))  
scaled_data=scaler.fit_transform(dataset)  
scaled_data
```

```
↔ array([[0.25270678],  
        [0.28617052],  
        [0.21926857],  
        [0.0780587 ],  
        [0.10527099],  
        [0.10855503],  
        [0.05029315],  
        [0.01658847],  
        [0.05687983],  
        [0.06741844],  
        [0.10837076],  
        [0.17860693],  
        [0.17723051],  
        [0.22517612],  
        [0.27392155],  
        [0.26145067],  
        [0.2340554 ],  
        [0.26496224],  
        [0.29522715],  
        [0.27175093],  
        [0.24239311],  
        [0.2626943 ]],  
       dtype=float64)
```

```

train_data=scaled_data[0:training_data_len, :]
x_train=[]
y_train=[]
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])

```

+ Code

+ Text

```

[ ] x_train, y_train=np.array(x_train), np.array(y_train)
x_train=np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

```

+ Code

+ Text

```

[ ] x_train.shape[0]

```

```

⇒ 219

```

```

[ ] x_train

```

```

⇒ array([[0.25270678],
         [0.28617052],
         [0.21926857],
         ...,
         [0.12686468],
         [0.17239528],
         [0.2424766 ]],

```

```

550 00000000

```

```

[ ] model=Sequential()
model.add(LSTM(units=40,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=40,return_sequences=False))
model.add(Dense(units=1))

```

```

⇒ /usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning:
super().__init__(**kwargs)

```

+ Code

+ Text

```

[ ] model.compile(optimizer='adam', loss='mean_squared_error')

```

```

[ ] model.fit(x_train, y_train, batch_size=16, epochs=100, validation_split=0.3)

```

```

⇒ Epoch 1/100
10/10 ━━━━━━━━━━━ 4s 101ms/step - loss: 0.0566 - val_loss: 0.0069
Epoch 2/100
10/10 ━━━━━━━━━━━ 1s 51ms/step - loss: 0.0082 - val_loss: 0.0452
Epoch 3/100
10/10 ━━━━━━━━━━━ 1s 53ms/step - loss: 0.0059 - val_loss: 0.0306
Epoch 4/100
10/10 ━━━━━━━━━━━ 1s 53ms/step - loss: 0.0040 - val_loss: 0.0172
Epoch 5/100
10/10 ━━━━━━━━━━━ 1s 51ms/step - loss: 0.0047 - val_loss: 0.0252
Epoch 6/100
10/10 ━━━━━━━━━━━ 1s 50ms/step - loss: 0.0036 - val_loss: 0.0177

```

```
[ ] test_data=scaled_data[training_data_len-60:, :]
    x_test=[]
    y_test=dataset[training_data_len:, :]
    for i in range(60, len(test_data)):
        x_test.append(test_data[i-60:i, 0])
```

```
[ ] x_test=np.array(x_test)
    x_test=np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

[+ Code](#)
[+ Text](#)

```
[ ] predictions_normalised=model.predict(x_test)
    predictions_normalised
```

 4/4  1s 116ms/step

```
array([[0.6862559 ],
       [0.68317693],
       [0.65493673],
       [0.6212052 ],
       [0.6677117 ],
       [0.7273595 ],
       [0.7537634 ],
       [0.78312   ],
       [0.7981052 ],
       [0.72045755],
       [0.662888  ],
       [0.67367536],
       [0.68618107],
```

 predictions= scaler.inverse_transform(predictions_normalised)
predictions

 array([[7340.3745],
 [7336.8994],
 [7305.0215],
 [7266.9453],
 [7319.442],
 [7386.773],
 [7416.5776],
 [7449.7153],
 [7466.6304],
 [7378.9814],
 [7313.997],
 [7326.174],
 [7340.29],
 [7354.12],
 [7369.809],
 [7397.8823],
 [7397.7373],
 [7346.7188],
 [7305.239],
 [7227.6104],

```

mse=np.mean(predictions-y_test)**2
mse

```

```

126.95563480806743

```

[+ Code](#)
[+ Text](#)

```

[ ] mape=np.mean(np.abs((predictions-y_test)/y_test))*100
mape

```

```

0.7114608914286356

```

[+ Code](#)
[+ Text](#)

```

[ ] rmse=np.sqrt(np.mean(((predictions-y_test)**2)))
rmse

```

```

67.30312019653441

```

```

[ ] train=data[:training_data_len]

```

```

train=data[:training_data_len]
valid=data[training_data_len:]
valid['Predictions']=predictions
plt.figure(figsize=(16,8))
plt.title('Prediksi IHSG Setelah Covid-19 dengan LSTM')
plt.xlabel('Data Harian', fontsize=18)
plt.ylabel('Close Price', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Valid', 'Predictions'], loc='lower right')
plt.show()

```

```

<ipython-input-28-fdaa00dfa3f6>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

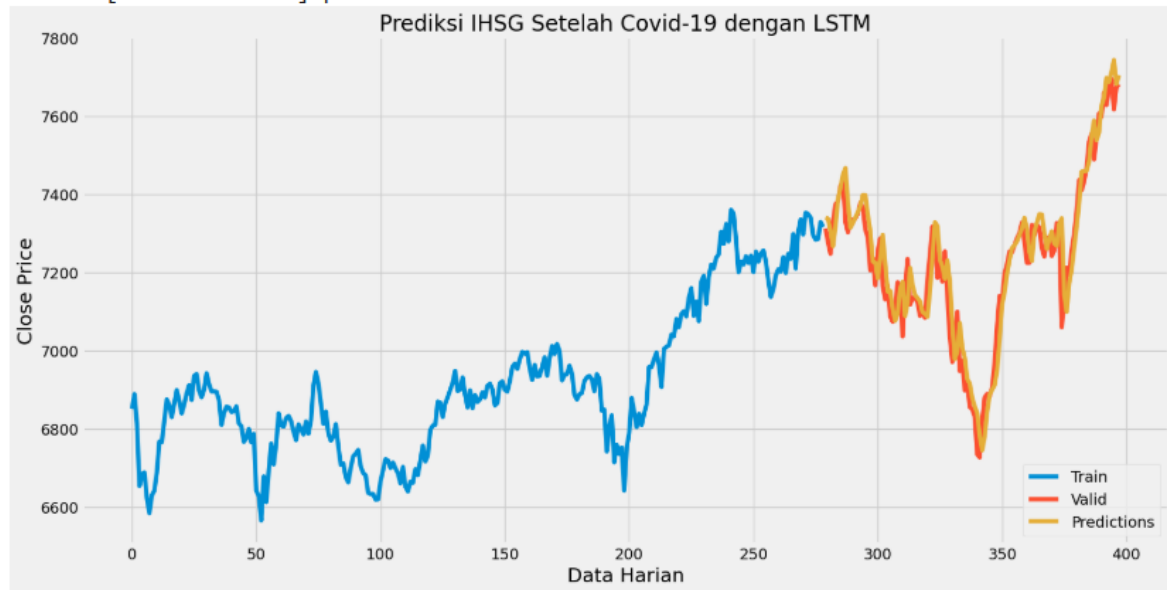
See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/using.html#copy-on-write>

7800

Prediksi IHSG Setelah Covid-19 dengan LSTM



```
valid['Predictions']=predictions
```



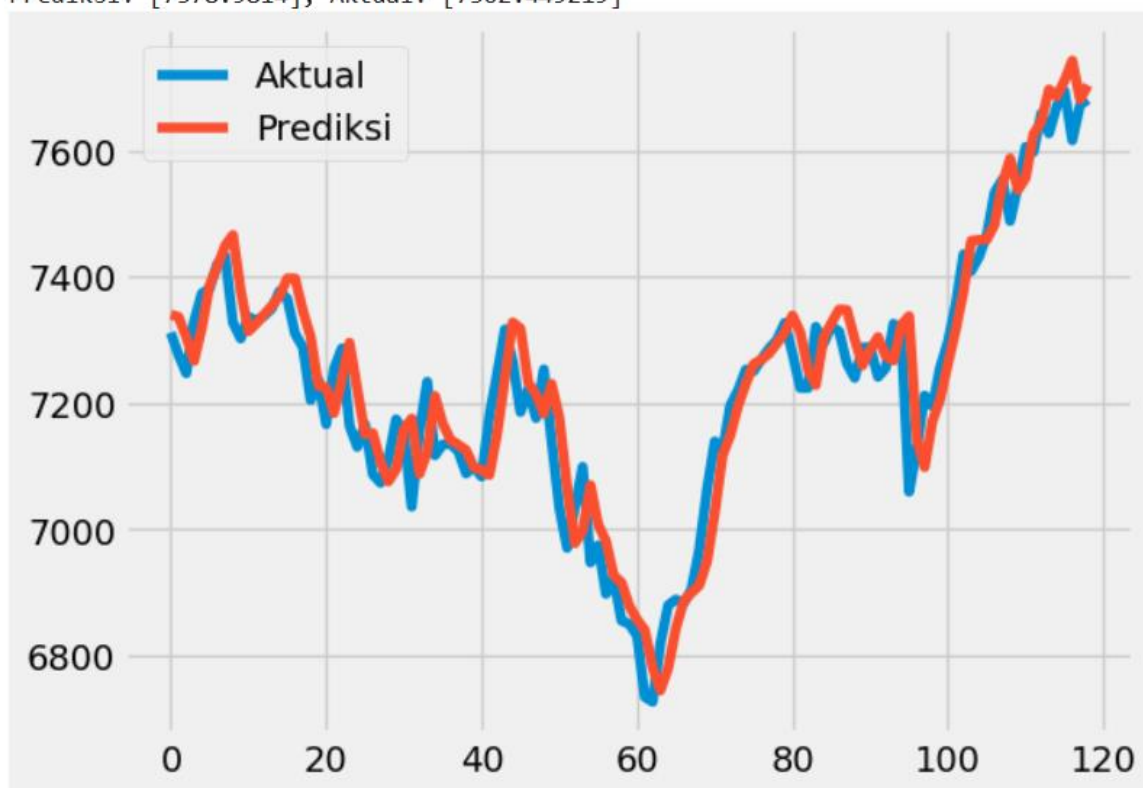
```
# Periksa prediksi dan nilai aktual
for i in range(10): # Cetak 10 data pertama
    print(f"Prediksi: {predictions[i]}, Aktual: {y_test[i]}")

# Visualisasi data untuk cek outlier
plt.plot(y_test, label="Aktual")
plt.plot(predictions, label="Prediksi")
plt.legend()
plt.show()
```



```
Prediksi: [7340.3745], Aktual: [7311.907227]
Prediksi: [7336.8994], Aktual: [7276.749023]
Prediksi: [7305.0215], Aktual: [7247.459961]
Prediksi: [7266.9453], Aktual: [7329.800781]
Prediksi: [7319.442], Aktual: [7373.963867]
Prediksi: [7386.773], Aktual: [7381.907227]
Prediksi: [7416.5776], Aktual: [7421.207031]
Prediksi: [7449.7153], Aktual: [7433.314941]
Prediksi: [7466.6304], Aktual: [7328.054199]
Prediksi: [7378.9814], Aktual: [7302.449219]
```

↳ Prediksi: [7378.9814], Aktual: [7302.449219]



valid



Close Predictions

279	7311.907227	7340.374512
280	7276.749023	7336.899414
281	7247.459961	7305.021484
282	7329.800781	7266.945312
283	7373.963867	7319.441895
...	...	...
393	7670.732910	7686.202148
394	7694.529785	7711.412598
395	7616.520996	7742.962891
396	7672.895996	7681.414551
397	7681.042969	7704.085938

