

**PREDIKSI HARGA BITCOIN BERDASARKAN INFORMASI  
BLOCKCHAIN MENGGUNAKAN METODE *LONG-SHORT*  
*TERM MEMORY***

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana  
Program Studi Statistika



Disusun Oleh

Kinanti Dhea Larasati

16611065

**PROGRAM STUDI STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2020**

## HALAMAN PERSETUJUAN PEMBIMBING

### TUGAS AKHIR

Judul : Prediksi Harga Bitcoin berdasarkan Informasi  
Blockchain Menggunakan Metode *Long-Short Term  
Memory*

Nama Mahasiswa : Kinanti Dhea Larasati

NIM : 16611065

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK  
DIUJIKAN**

Yogyakarta, 10 Maret 2020

**Pembimbing**



(Arum Handini Primandari, S.Pd.Si., M.Si.)

# HALAMAN PENGESAHAN

## TUGAS AKHIR

PREDIKSI HARGA BITCOIN BERDASARKAN INFORMASI  
BLOCKCHAIN MENGGUNAKAN METODE *LONG-SHORT TERM*

*MEMORY*

Nama Mahasiswa : Kinanti Dhea Larasati

NIM : 16611065

TUGAS AKHIR INI TELAH DIUJIKAN  
PADA TANGGAL: 02 APRIL 2020

Nama Penguji:

Tanda Tangan

1. Muhammad Muhajir, S.Si., M.Sc.

2. Dr. RB Fajriya Hakim, M.Si.

3. Arum Handini Primandari, S.Pd, Si., M.Sc.

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb*

Alhamdulillahirabbil'aalamin, puji syukur kehadirat Allah SWT yang mana beliau telah melimpahkan rahmat, hidayah dan inayah nya sehingga penulis dapat menyelesaikan Tugas Akhir ini, Sholawat serta salam semoga terlimpahkan kepada junjungan nabi besar Muhammad SAW yang telah membawa umatnya dari zaman jahiliah menuju zaman terang benderang dan penuh ilmu pengetahuan.

Tugas Akhir dengan judul "**Prediksi Harga Bitcoin berdasarkan Informasi Blockchain Menggunakan Metode *Long-Short Term Memory***" disusun sebagai syarat untuk memperoleh gelar sarjana Statistika pada program studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.

Selama proses penyusunan Tugas Akhir ini, tentunya tak lepas dari bantuan dan dukungan dari berbagai pihak, baik secara langsung maupun tidak langsung. Maka, penulis bermaksud menyampaikan terima kasih kepada :

1. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.
2. Bapak Dr.Edy Widodo,S.Si., M.Si.selaku ketua Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia.
3. Ibu Arum Handini Primandari, S.Pd.Si., M.Si.selaku dosen pembimbing yang telah membantu dan memberikan masukan serta motivasi dari awal sampai selesai Tugas Akhir ini.
4. Seluruh staff pengajar Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia yang telah memberikan ilmu ilmu yang bermanfaat bagi penulis.
5. Kedua Orang tua penulis, Bapak Kurniadi dan Ibu Irma Muchayati, Dua saudara perempuan penulis Yuninda Safira Mentari dan Aqila Shafa

Ramadani serta seluruh keluarga besar yang selalu memberikan kasih sayang, dukungan, motivasi, dan doa untuk kelancaran segala urusan penulis.

6. Sahabat seperjuangan: Ria Amelia dan Barlinda Titania atas rasa kekeluargaan serta semua dukungan, bantuan dan memberikan semangat bagi penulis.
7. Sahabat penulis: Nilda Aulia sebagai partner Tugas Akhir yang selalu menemani, membantu dan berperan besar dalam penyelesaian Tugas Akhir.
8. Teman satu bimbingan Tugas Akhir yang saling menguatkan dan memberikan hiburan selama proses pengerjaan Tugas Akhir.
9. Teman Statistika Angkatan 2016 yang telah menemani proses perjalanan penulis selama mengembangkan diri dan memberikan pelajaran pelajaran berharga bagi penulis.
10. Semua pihak yang telah membantu yang tidak bisa penulis sebutkan satu per satu.

Semoga ALLAH SWT senantiasa selalu memberikan rahmat, Karunia dan Anugerahnya kepada mereka semua.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penulisan Tugas Akhir ini dikarenakan keterbatasan pengetahuan dan kemampuan yang dimiliki oleh Penulis. Oleh karena itu, penulis mengharapkan kritik dan saran untuk menyempurnakan penulisan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi banyak pihak.

*Wassalamualaikum Wr.Wb*

Yogyakarta, 10 Maret 2020



(Kinanti Dhea Larasati)

## DAFTAR ISI

HALAMAN SAMPUL .....	i
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	5
BAB 2 TINJAUAN PUSTAKA.....	6
BAB 3 LANDASAN TEORI.....	11
3.1 <i>Cryptocurrency</i> .....	11
3.2 Bitcoin .....	12
3.2.1 Cara Kerja Bitcoin Secara Teknis .....	13
• Proof-of-Work.....	15
• Network.....	16
• Incentive.....	17
• Reclaiming Disk Space .....	18
• Verifikasi Pembayaran Sederhana.....	19
• Menggabungkan dan Membagi Nilai .....	19
• Privacy .....	20
3.2.2 Cara Kerja Bitcoin untuk Pengguna .....	21
3.3 Blockchain.....	23
3.4 Analisis Deskriptif .....	25
3.4.1 Penyajian dalam Bentuk Visualisasi.....	26
3.4.2 Ukuran Pemusatan Data.....	28
3.5 <i>Time Series</i> .....	29
3.5.1 Trend.....	29
3.5.2 Variasi Siklis .....	30
3.5.3 Musiman .....	30
3.5.4 Fluktuasi Tak Beraturan.....	31

3.6	<i>Artificial Neural Networks</i> .....	31
3.7	<i>Recurrent Neural Network</i> .....	33
3.8	<i>Long Short Term Memory</i> .....	35
3.9	Parameter Evaluasi.....	40
	3.9.1 Mean Square Error.....	40
	3.9.2 Root Mean Square Error .....	41
	3.9.3 Mean Absolute Percentage Error.....	41
3.10	Sistem Pengerjaan LSTM.....	41
	3.10.1 Preprocessing Data .....	41
	3.10.2 Inisialisasi Parameter .....	42
	3.10.3 Training LSTM Network .....	42
	3.10.4 Testing .....	42
3.11	<i>Adaptive Momen Estimation (Adam)</i> .....	43
BAB 4	METODOLOGI PENELITIAN .....	44
4.1	Populasi Penelitian.....	44
4.2	Sumber Data .....	44
4.3	Variabel Penelitian dan Definisi Operasional .....	44
4.4	Metode Analisis Data.....	45
BAB 5	HASIL DAN PEMBAHASAN .....	47
5.1	Analisis Deskriptif .....	47
5.2	Analisis <i>Multivariate Long-Short Term Memory</i> .....	47
	5.2.1 Normalisasi .....	48
	5.2.2 Pembentukan Data Training dan Data Testing .....	49
	5.2.3 Penentuan Jumlah Neuron dan Epoch .....	49
5.3	Prediksi Data .....	52
BAB 6	PENUTUP.....	55
6.1	Kesimpulan.....	55
6.2	Saran.....	55

## DAFTAR TABEL

<b>Tabel 2.1.</b> Penelitian Terdahulu Mengenai LSTM .....	6
<b>Tabel 4.1.</b> Definisi Operasional Variabel Penelitian.....	44
<b>Tabel 5.1.</b> Hasil Normalisasi $x_1$ - $x_5$ .....	48
<b>Tabel 5.2.</b> Hasil Normalisasi $x_6$ - $x_{10}$ .....	49
<b>Tabel 5.3.</b> Hasil Pengujian dengan Jumlah <i>Neuron</i> dan <i>Epoch</i> yang Berbeda.....	50
<b>Tabel 5.4.</b> Hasil Pengujian.....	53



## DAFTAR GAMBAR

<b>Gambar 3.1</b> Proses Transaksi Keuangan Terdesentralisasi .....	12
<b>Gambar 3.2</b> Transaksi Bitcoin .....	14
<b>Gambar 3.3</b> Server <i>Timestamp</i> .....	15
<b>Gambar 3.4</b> Sistem <i>Proof-of-Work</i> .....	16
<b>Gambar 3.5</b> Transaksi <i>Hash</i> .....	18
<b>Gambar 3.6</b> Rantai <i>Proof-of-Work</i> .....	19
<b>Gambar 3.7</b> Rantai <i>Proof-of-Work</i> .....	20
<b>Gambar 3.8</b> Model Privasi Bitcoin.....	20
<b>Gambar 3.9</b> Contoh Transaksi Bitcoin .....	21
<b>Gambar 3.10</b> BlockchainSederhana .....	23
<b>Gambar 3.11</b> Histogram .....	26
<b>Gambar 3.12</b> <i>Pie Chart</i> .....	26
<b>Gambar 3.13</b> Poligon.....	27
<b>Gambar 3.14</b> Ogive .....	27
<b>Gambar 3.15</b> Diagram Batang Daun .....	28
<b>Gambar 3.16</b> Pola Data <i>Trend</i> .....	29
<b>Gambar 3.17</b> Pola Data Siklis .....	30
<b>Gambar 3.18</b> Pola Data Musiman .....	31
<b>Gambar 3.19</b> Pola Data Fluktuasi Tak Beraturan .....	31
<b>Gambar 3.20</b> BlockchainSederhana .....	33
<b>Gambar 3.21</b> Struktur RNN .....	34
<b>Gambar 3.22</b> Struktur LSTM.....	35
<b>Gambar 3.23</b> Struktur LSTM untuk <i>Forget Gate</i> .....	36
<b>Gambar 3.24</b> Struktur LSTM untuk <i>Input Gate</i> .....	37
<b>Gambar 3.25</b> Struktur LSTM untuk <i>Cell State</i> .....	37
<b>Gambar 3.26</b> Struktur LSTM untuk <i>Output Gate</i> .....	38
<b>Gambar 4.1</b> . <i>Flowchart</i> Analisis Data.....	46
<b>Gambar 5.1</b> . <i>Plot</i> Harga Bitcoin.....	47

<b>Gambar 5.2.</b> Hasil Bobot dari Setiap <i>Neuron</i> untuk Variabel <i>Price</i> .....	51
<b>Gambar 5.3.</b> Bias untuk Setiap <i>Neuron</i> .....	51
<b>Gambar 5.4.</b> <i>Plot</i> Perbandingan Data Prediksi dan Data Aktual .....	52
<b>Gambar 5.5.</b> Nilai MAPE untuk Model Terbaik.....	54

## DAFTAR LAMPIRAN

<b>Lampiran 1</b> Data Harga dan Informasi Blockchain .....	62
<b>Lampiran 2</b> Data Hasil Normalitas .....	67
<b>Lampiran 3</b> Nilai MSE .....	72
<b>Lampiran 4</b> <i>Syntax</i> Prediksi Harga Bitcoin Menggunakan <i>Multivariate LSTM</i> ..	75

## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 10 Maret 2020



## **ABSTRAK**

### **PREDIKSI HARGA BITCOIN BERDASARKAN INFORMASI BLOCKCHAIN MENGGUNAKAN METODE *LONG-SHORT TERM MEMORY***

Kinanti Dhea Larasati

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Sejak didirikan pada 2008, BTC telah muncul sebagai mata uang digital utama dalam hal kapitalisasi pasar dan terus menarik perhatian investor dan pembuat kebijakan. BTC telah ditandai oleh volatilitas harga yang tinggi dalam beberapa tahun terakhir; peningkatan substansial pada 2016 diikuti oleh penurunan yang signifikan pada 2018. Bitcoin dapat diperdagangkan setiap saat karena tidak memiliki periode tutup yang membedakannya dengan pasar lainnya. Namun hal ini juga mengandung resiko, untuk memperkecil resiko tersebut dapat dilakukan prediksi harga Bitcoin dengan melihat faktor lain yaitu informasi blockchain. Metode yang digunakan pada penelitian ini adalah *Long-Short Term Memory* (LSTM) yang merupakan bentuk pengembangan dari *Recurrent Neural Network* (RNN). Pengembangan ini dilakukan untuk menghindari masalah ketergantungan jangka panjang dan sesuai untuk memproses serta memprediksi deret waktu. Data yang digunakan adalah data *time series* pada tanggal 04 Agustus 2018 s.d 21 Januari 2020. Berdasarkan penelitian yang dilakukan dengan melihat nilai MSE didapatkan bahwa model dengan jumlah neuron 20 dan *epoch* 500 memiliki nilai MSE terkecil. Kemudian dilakukan prediksi dan diperoleh bahwa nilai prediksi tidak jauh berbeda dari data aktual, dengan tingkat akurasi sebesar 91,07% berdasarkan nilai MAPE.

**Kata Kunci:** Bitcoin, Informasi Blockchain, Prediksi, *Long-Short Term Memory*.

## **ABSTRACT**

### **PREDICTION OF BITCOIN PRICES BASED ON BLOCKCHAIN INFORMATION USING LONG-SHORT TERM MEMORY METHOD**

Kinanti Dhea larasati

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Since its founding in 2008, BTC has emerged as a major digital currency in terms of market capitalization and continues to attract the attention of investors and policy makers. BTC has been characterized by high price volatility in recent years; Substantial increases in 2016 are followed by significant decreases in 2018. Bitcoin can be traded at any time because it does not have a closing period that distinguishes it from other markets. But this also contains risks, to minimize the risk can be predicted Bitcoin prices by looking at other factors, namely the blockchain information. The method to be used in this research is Long-Short Term Memory (LSTM) which is a form of development of the Recurrent Neural Network (RNN). This development is carried out to avoid long-term dependency problems and is suitable for processing and predicting time series. The data used are time series data on August 4, 2018 until January 21, 2020. Based on research conducted by looking at the MSE values, it is found that the model with the number of neurons 20 and epoch 500 has the smallest MSE value. Then a prediction is made and the prediction value is not much different from the actual data, with an accuracy level of 91.07% based on the MAPE value.

**Keywords:**Bitcoin, Blockchain Information, Predictions, Long-Short Term Memory.

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang Masalah

Teknologi disajikan sebagai sesuatu yang baru karena mendorong perubahan pada tingkat yang semakin meningkat. Dalam kenyataannya, teknologi informasi (TI) memproses data, mengumpulkan informasi, menyimpan materi yang dikumpulkan, mengakumulasi pengetahuan, dan mempercepat komunikasi, memainkan peran penting dalam banyak aspek operasi sehari-hari dunia bisnis saat ini. Menanggapi perubahan teknologi baru, pelaku tindak kriminal dan masyarakat semakin menemukan cara baru untuk berkembang. Oleh karena itu, teknologi percepatan diperkenalkan ke banyak mekanisme keuangan seperti *cryptocurrency* (Bakar & Rosbi, 2017).

Elemen umum dari sistem *cryptocurrency* yang berbeda adalah buku besar publik ('Blockchain') yang dibagikan antara peserta jaringan dan penggunaan token asli sebagai cara untuk mendorong peserta untuk menjalankan jaringan tanpa adanya otoritas pusat. Namun, ada perbedaan yang signifikan antara beberapa *cryptocurrency* terkait dengan tingkat inovasi yang ditampilkan (Hileman & Rauchs, 2017).

Sebagian besar *cryptocurrency* adalah kloning Bitcoin atau *cryptocurrency* lainnya dan hanya menampilkan nilai parameter yang berbeda (contohnya Waktu blok yang berbeda, pasokan mata uang, dan skema penerbitan). *Cryptocurrency* jenis ini menunjukkan sedikit atau tidak ada inovasi dan sering disebut sebagai 'altcoin'. Contohnya Dogecoin dan Ethereum Classic.

Sebaliknya, sejumlah *cryptocurrency* yang telah muncul meminjam beberapa konsep dari Bitcoin, menyediakan fitur-fitur baru dan inovatif yang menawarkan perbedaan substansial. Ini dapat mencakup pengenalan mekanisme konsensus baru (contohnya bukti kepemilikan) serta *platform* komputasi terdesentralisasi dengan kemampuan "Smart Contract" yang menyediakan fungsionalitas yang sangat berbeda dan memungkinkan kasus penggunaan non-moneter. Inovasi *cryptocurrency* dan Blockchain' ini dapat dikelompokkan ke dalam dua kategori: sistem Blockchain baru (publik) yang menampilkan

Blockchain mereka sendiri (contohnya Ethereum, Peercoin, Zcash), dan Aplikasi / Lainnya yang ada pada lapisan tambahan yang dibangun di atas Blockchain yang ada sistem (contohnya Counterparty, Augur).

Sejak didirikan pada 2008, BTC telah muncul sebagai mata uang digital utama dalam hal kapitalisasi pasar dan terus menarik perhatian investor dan pembuat kebijakan. BTC telah ditandai oleh volatilitas harga yang tinggi dalam beberapa tahun terakhir; peningkatan substansial pada 2016 diikuti oleh penurunan yang signifikan pada 2018 (Al-Yahyaee & dkk, 2019). Transaksi Bitcoin pertama terjadi pada Januari 2009. Lebih dari dua tahun kemudian, berbagai laporan memperkirakan sirkulasi Bitcoin menjadi lebih dari 6,5 juta dengan sekitar 10.000 pengguna. Sementara transaksi awal dalam Bitcoin tampaknya berfungsi sesuai dengan niat awal, segera muncul laporan bahwa Bitcoin digunakan untuk membeli obat-obatan terlarang. Pembuat kebijakan di seluruh dunia menjadi khawatir dengan anonimitas yang diberikan oleh Bitcoin (Blau, 2018).

Glaser et al. (2014) mengajukan pertanyaan yang berfokus pada kemampuan bitcoin sebagai media pertukaran. Bitcoin cukup likuid, karena seseorang dapat menukar mata uang apa pun dengan Bitcoin tanpa ada batasan waktu. Namun karena kelangkaannya, Bitcoin memiliki keterbatasan likuiditas. Serta menemukan bahwa mayoritas pengguna memperlakukan investasi bitcoin mereka sebagai aset spekulatif daripada sebagai alat pembayaran. Oleh karena itu bitcoin mungkin paling bermanfaat sebagai aset daripada mata uang. Selanjutnya, Bohme et al. (2015) menemukan bahwa transaksi dapat ditunda hingga satu jam yang sangat mengurangi kemungkinan likuiditas. Namun, protokol bitcoin tidak membatasi transfer karena daftar pantauan atau negara yang diembargo karena identitas penggunanya tidak diketahui. Ini memberikan fleksibilitas dan kecepatan transfer internasional bitcoin yang sangat besar dibandingkan dengan mata uang lain yang dikelola oleh bank (Dyhrberg, 2016).

Hal ini menyebabkan semakin banyak masyarakat yang tertarik untuk melakukan investasi menggunakan Bitcoin. Penyediaan informasi sangat



dibutuhkan untuk merencanakan maupun mengambil keputusan agar mendapatkan keuntungan dan mengurangi resiko.

Keragaman penelitian menunjukkan pentingnya pasar mata uang digital dan khususnya BTC untuk investor dan akademisi internasional. BTC dicirikan oleh tingkat volatilitas yang tinggi dan harganya menunjukkan sisi atas dan bawah yang ekstrem selama periode waktu yang singkat. Indeks ketidakpastian dapat menjadi faktor penentu yang membantu pelaku pasar untuk memprediksi pengembalian harga BTC dan hubungan antara BTC dan indeks ketidakpastian (Al-Yahyaee & dkk, 2019).

Bitcoin serta *cryptocurrency* lainnya dapat diperdagangkan setiap saat karena tidak memiliki periode tutup yang membedakannya dengan pasar lainnya. Bitcoin dapat diperdagangkan setiap saat karena tidak memiliki periode tutup yang membedakannya dengan pasar lainnya. Namun hal ini juga mengandung resiko, untuk memperkecil resiko tersebut dapat dilakukan prediksi harga Bitcoin dengan melihat faktor lain yaitu informasi blockchain. Salah satu cara yang digunakan untuk melakukan hal tersebut adalah melakukan prediksi harga Bitcoin secara akurat.

Saat melakukan prediksi diperlukan metode yang tepat. Salah satu metode yang dapat digunakan adalah *Long-Short Term Memory* (LSTM) yang merupakan bentuk pengembangan dari *Recurrent Neural Network* (RNN). Tujuan dari pengembangan ini dimaksudkan untuk menghindari masalah ketergantungan jangka panjang dan sesuai untuk memproses serta memprediksi deret waktu. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, model LSTM terdiri dari serangkaian sel memori unik yang menggantikan neuron lapisan tersembunyi dari RNN. RNN sendiri merupakan bentuk pengembangan dari *Artificial Neural Network* (ANN). Pengembangan tersebut dilakukan karena ANN tidak mampu melakukan proses *training* secara cepat pada data yang banyak. RNN sendiri dirancang untuk memproses data bersambung atau berurutan, serta biasa digunakan untuk menyelesaikan masalah dengan data *time series*. Tapi RNN tidak mampu mengingat data historis yang terlalu lama atau dapat dikatakan

bahwa RNN kurang baik digunakan untuk mempelajari pola data berurutan yang terlalu panjang.

Perbedaan antara LSTM dan RNN terletak pada *forget gate* yang dimiliki oleh LSTM. *Forget gate* merupakan kelebihan yang dimiliki oleh LSTM dimana pada bagian ini akan menentukan informasi tersebut dibuang atau disimpan pada model. Kelemahan dari LSTM pada penelitian ini adalah ketidakmampuan untuk memprediksi harga Bitcoin yang akan datang. Hal ini disebabkan karena pada penelitian ini menggunakan lebih dari satu variabel, sehingga untuk mengetahui prediksi yang akan datang harus diketahui nilai-nilai dari variabel lainnya.

Sejumlah penelitian telah dilakukan baru-baru ini pada pemodelan rangkaian waktu harga Bitcoin sebagai variabel pasar baru dengan aturan teknis tertentu. (Qiu, Wang, & Zhou, 2020) melakukan penelitian berupa membandingkan hasil dengan tiga model lainnya, termasuk model LSTM, model LSTM dengan *wavelet denoising* dan model jaringan saraf *Gated Recurrent Unit* (GRU) pada S&P 500, DJIA, dataset HIS dan menunjukkan bahwa koefisien penentuan model LSTM lebih tinggi dari 0,94, dan kesalahan kuadrat rata-rata dari kedua model lebih rendah dari 0,05. (Jang & Lee, 2018) melakukan penelitian untuk mengungkapkan efek dari *Bayesian Neural Networks* (BNN) dengan menganalisis deret waktu proses Bitcoin menggunakan informasi Blockchain yang sangat terlibat dalam penawaran dan permintaan Bitcoin serta menggunakannya untuk melatih model untuk meningkatkan kinerja prediksi proses penetapan harga Bitcoin terbaru.

Berdasarkan beberapa penelitian terdahulu, peneliti tertarik untuk melakukan analisis praktis pada pemodelan dan prediksi harga Bitcoin menggunakan *Multivariate LSTM* dengan menggunakan informasi Blockchain.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah di atas, maka rumusan masalah yang akan di bahas pada penelitian ini adalah :

1. Bagaimana deskripsi data harian Bitcoin?

2. Bagaimana memprediksi harga Bitcoin berdasarkan informasi Blockchain dengan menggunakan LSTM?

### **1.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Data yang digunakan adalah *Price Bitcoin, Miners Revenue, Difficulty, Hash Rate, Median Confirmation Time, Average Block Size, Total Number of a Unique Bitcoin, Transaction per Block, Confirmed Transaction*, dan *Cosh % per Transaction*.
2. Data tersebut diambil dari periode 04 Agustus 2018 s.d 21 Januari 2020.
3. Software yang di gunakan adalah *Python*

### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini yaitu :

1. Mengetahui deskripsi dari data harian Bitcoin.
2. Mengetahui hasil prediksi harga Bitcoin menggunakan *Multivariate LSTM* berdasarkan informasi Blockchain.

### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut :

1. Memberikan pengetahuan terkait pemodelan dan prediksi harga Bitcoin berdasarkan informasi Blockchain.
2. Penelitian ini dapat digunakan sebagai sarana belajar dan mendalami ilmu terkait perdagangan atau investasi sebagai pertimbangan dalam mengambil keputusan.
3. Dapat menjadi referensi bagi peneliti selanjutnya tentang metode *Multivariate LSTM* maupun tentang prediksi harga Bitcoin.

## BAB 2 TINJAUAN PUSTAKA

Terkait dengan penelitian yang dilakukan penulis, maka penelitian terdahulu menjadi sangat penting agar dapat diketahui hubungan antara penelitian yang dilakukan sebelumnya dengan penelitian yang dilakukan saat ini sehingga dapat diketahui kontribusi penelitian ini terhadap perkembangan ilmu pengetahuan. Penelitian mengenai bitcoin sudah banyak dilakukan oleh banyak peneliti sebelumnya, namun prediksi harga bitcoin menggunakan metode LSTM Multivariat belum banyak dilakukan.

**Tabel 2.1.** Penelitian Terdahulu Mengenai LSTM

No	Nama Peneliti dan Tahun	Judul Penelitian	Metode Penelitian	Isi
1.	(Qiu, Wang, & Zhou, 2020)	<i>Forecasting stock prices with long-short term memory neural network based on attention mechanism</i>	<i>Long-Short Term Memory Neural Network</i>	Pada penelitian ini peneliti membandingkan hasil dengan tiga model lainnya, termasuk model LSTM, model LSTM dengan <i>wavelet denoising</i> dan model jaringan saraf <i>Gated Recurrent Unit (GRU)</i> pada S&P 500, DJIA, dataset HSI. Hasil dari percobaan pada dataset S&P 500 dan DJIA menunjukkan bahwa koefisien penentuan model LSTM lebih tinggi dari 0,94, dan kesalahan kuadrat rata-rata dari kedua model lebih rendah dari 0,05.
2.	(Aldi,	Analisis dan	<i>Long Short</i>	Sistem yang dibangun

No	Nama Peneliti dan Tahun	Judul Penelitian	Metode Penelitian	Isi
	Jondri, & Aditsania, (2018)	Implementasi <i>Long Short Term Memory Neural Network</i> untuk Prediksi Harga Bitcoin	<i>Term Memory Neural Networ</i>	pada penelitian ini adalah menggunakan metode jaringan syaraf tiruan yaitu dengan menggunakan arsitektur <i>Long Short Term Memory Neural Networks</i> . Dalam penelitian ini akan dianalisis beberapa parameter seperti jumlah pola time series, jumlah neuron hidden, <i>max epoch</i> , dan komposisi data latih dan uji terhadap akurasi prediksi yang didapatkan. Hasil analisis menunjukkan bahwa sistem yang dibangun mampu memprediksi harga Bitcoin dengan baik, dengan rata-rata tingkat akurasi sebesar 93.5% terhadap data testing.
3.	(Jang & Lee, 2018)	<i>An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information</i>	<i>Bayesian Neural Networks</i>	Makalah ini mengungkapkan efek dari <i>Bayesian Neural Networks</i> (BNN) dengan menganalisis deret waktu proses Bitcoin. Menggunakan informasi Blockchain yang sangat terlibat dalam penawaran dan permintaan Bitcoin sertamenggunakannya untuk melatih model

No	Nama Peneliti dan Tahun	Judul Penelitian	Metode Penelitian	Isi
				<p>untuk meningkatkan kinerja prediksi proses penetapan harga Bitcoin terbaru. Penelitian ini membandingkan BNN dengan model tolok ukur linear dan non-linear lainnya dalam pemodelan dan memprediksi proses Bitcoin. Hasil dari penelitian ini menunjukkan bahwa BNN berkinerja baik dalam memprediksi harga Bitcoin.</p>
4.	(Che & dkk, 2018)	<i>Recurrent Neural Networks for Multivariate Time Series with Missing Values</i>	<i>Recurrent Neural Networks for Multivariate Time Series</i>	<p>Data deret waktu multivariat dalam aplikasi praktis, seperti perawatan kesehatan, geosains, dan biologi, dicirikan oleh berbagai nilai yang hilang. GRU-D didasarkan pada <i>Gated Recurrent Unit</i> (GRU), sebuah jaringan saraf berulang yang canggih. Eksperimen tugas klasifikasi deret waktu pada dataset klinis dunia nyata (MIMIC-III, PhysioNet) dan dataset sintesis menunjukkan bahwa model tersebut mencapai kinerja yang baik dan</p>

No	Nama Peneliti dan Tahun	Judul Penelitian	Metode Penelitian	Isi
				memberikan wawasan yang bermanfaat untuk pemahaman yang lebih baik dan pemanfaatan nilai-nilai yang hilang dalam rangkaian analisis runtun waktu.
5.	(Zheng, Xu, Zhang, & Li, 2017)	<i>Electric Load Forecasting in Smart Grid Using Long-Short-Term-Memory based Recurrent Neural Network</i>	<i>Long-Short Term Memory based Recurrent Neural Network</i>	Peramalan beban listrik jangka pendek memprediksi beban beberapa jam hingga beberapa minggu ke depan. Makalah ini mengeksplorasi <i>Long-Short-Term-Memory</i> (LSTM) yang berbasis <i>Recurrent Neural Network</i> (RNN) untuk menghadapi tantangan ini. Hasil percobaan menunjukkan bahwa metode peramalan berbasis LSTM dapat mengungguli sebagian besar metode peramalan tradisional dalam masalah peramalan beban listrik jangka pendek.
6.	(Lipton & dkk, 2016)	<i>Learning to Diagnose with LSTM Recurrent Neural Networks</i>	<i>Long-Short Term Memory Neural Network</i>	Penelitian pertama yang secara empiris mengevaluasi kemampuan LSTM untuk mengenali pola dalam serangkaian waktu pengukuran klinis multivariat.

No	Nama Peneliti dan Tahun	Judul Penelitian	Metode Penelitian	Isi
				<p>Penelitian ini dilakukan dengan cara menetapkan efektivitas jaringan LSTM sederhana untuk memodelkan data klinis. Selanjutnya menunjukkan strategi pelatihan yang langsung dan efektif di mana penelitian ini mereplikasi target pada setiap langkah urutan. Dilatih hanya pada seri waktu mentah, model tersebut mengungguli beberapa garis dasar yang kuat, termasuk perceptron multilayer yang dilatih pada fitur <i>hand-engineered</i>.</p>

Pada **Tabel 2.1** telah disebutkan adanya penelitian dengan tema yang serupa, namun penelitian yang akan dilakukan oleh peneliti memiliki perbedaan yaitu dengan menambahkan *Price* (harga Bitcoin) pada variabel independen serta metode yang digunakan adalah Multivariat LSTM dengan menggunakan *software Python*.



## BAB 3 LANDASAN TEORI

### 3.1 *Cryptocurrency*

*Cryptocurrency* adalah metode untuk membentuk “koin” virtual dan menyediakan kepemilikan dan transaksi yang aman menggunakan masalah kriptografi. Masalah ini dirancang agar mudah diverifikasi tetapi secara komputasi sulit untuk mencapai solusi. Berbagai *cryptocurrency* menggunakan fungsi yang berbeda untuk tujuan ini, yang paling umum menjadi target *hash*, dimana *hash* dihitung sehingga lebih rendah dari nilai tertentu. Target *hash* (contoh : kesulitan masalah) disesuaikan setiap waktu berdasarkan pada daya komputasi total pada kurang konstan. Bukti kerja intensif komputasi adalah metode dimana transaksi diverifikasi sebagai kode unik dan dapat dipercaya. Untuk mendorong partisipasi, transaktor dapat menyertakan biaya transaksi yang masuk ke pengguna pertama yang berhasil melakukan verifikasi. (Harwick, 2016).

*Cryptocurrency* adalah mata uang digital yang dibangun menggunakan teknologi *Blockchain*. Dimana teknologi ini tidak memerlukan pihak ketiga sebagai perantaranya sehingga setiap transaksi menjadi lebih transparan. Dengan teknologi *Blockchain* setiap data yang ada akan saling terhubung, dimana data tersebut dimiliki oleh setiap orang yang berada dalam lingkungan pengguna sistem *cryptocurrency* tersebut (Bhiantara, 2018).

*Cryptocurrency* pertama adalah Bitcoin yang ditemukan oleh Satoshi Nakamoto pada 2008 dan telah beredar sejak 2009. Sejak itu, Bitcoin telah berubah menjadi *cryptocurrency* paling terkenal dan istilah representatif untuk *cryptocurrency* / mata uang digital. Secara teknis, "penambangan" *cryptocurrency* berarti bahwa dengan memperkuat kekuatan komputasi untuk berpartisipasi dalam jaringan *cryptocurrency* dan menyetujui transaksi, sejumlah kecil biaya akan dibayarkan dalam *cryptocurrency*. Fitur paling penting dari *cryptocurrency* adalah menggunakan kontrol desentralisasi sehingga pembeli dan penjual melakukan transaksi antara satu sama lain secara langsung (*peer-to-peer*), para pedagang akan tetap anonim sehingga privasi dilindungi hingga batas maksimum, catatan tidak dapat dipulihkan, ini berlaku di seluruh dunia, efisien, dan bebas

masalah untuk pengeluaran ganda. Banyak peneliti telah mencari faktor penentu Bitcoin dan telah menyelidiki beberapa faktor lintas disiplin ilmu, misalnya, ketidakpastian kebijakan, kekuatan pasar, ketidakpastian global, biaya karakteristik pengguna produksi, peraturan keuangan, dan tekanan finansial(Hassani, Huang, & Silva, 2018).



**Gambar 3.1**Proses Transaksi Keuangan Terdesentralisasi

Sumber: (Eka, 2018)

Sifat terdesentralisasi artinya tidak ada pihak khusus yang menjadi perantara. Transaksi dilakukan secara *peer-to-peer* dari pengirim ke penerima. Setiap transaksi yang dilakukan dicatat dalam komputer yang berada di jaringan tersebut, di seluruh dunia, atau disebut dengan *miner* (penambang yang ikut membantu mengamankan serta mencatat transaksi di jaringan). *Miner* akan mendapatkan uang virtual yang digunakan, tidak semua orang dapat menjadi *miner* karena dibutuhkan keahlian khusus dengan pemrosesan komputasi yang rumit untuk memecahkan kriptografi yang digunakan. Inilah salah satu alasan mengapa para *miner cryptocurrency* umumnya menggunakan komputer berspesifikasi tinggi dan khusus.

### 3.2 Bitcoin

Bitcoin atau disingkat dengan “BTC” adalah mata uang digital dengan sistem pembayaran secara elektronik yang didasarkan pada bukti kriptografi yang memungkinkan dua pihak bertansaksi secara langsung tanpa perantara pihak ketiga. Bitcoin memanfaatkan jaringan *peer-to-peer network* dengan *proof-of-work* untuk mencatat sejarah transaksi publik secara praktis namun tidak praktis secara komputasi bagi peretas. Pertama diusulkan oleh Satoshi Nakamoto pada tahun 2008 (Nakamoto, 2008). Transaksi Bitcoin pertama terjadi pada tahun 2009.

Setelah 2 tahun, berbagai laporan memperkirakan sirkulasi Bitcoin lebih dari US\$ 6,5 juta dengan sekitar 10.000 pengguna (Blau, 2018).

Aktivitas jual beli di internet sepenuhnya bergantung pada lembaga keuangan terpercaya sebagai pihak ketiga untuk memproses pembayaran elektronik. Sistem ini bekerja cukup baik untuk sebagian besar transaksi, namun masih memiliki kelemahan dari model berbasis kepercayaan. Sebenarnya tidak ada transaksi yang tidak dapat dibatalkan karena lembaga keuangan tidak dapat menghindari mediasi perselisihan.

Transaksi yang sepenuhnya tidak dapat dibatalkan sebenarnya tidak mungkin, karena lembaga keuangan tidak dapat menghindari penyelesaian konflik perselisihan. Biaya penyelesaian konflik meningkatkan biaya transaksi, membatasi ukuran transaksi praktis minimum dan memotong kemungkinan untuk transaksi kasual kecil, dan ada biaya yang lebih luas dalam hilangnya kemampuan untuk melakukan pembayaran non-reversibel untuk layanan yang tidak dapat dikembalikan. Yang dibutuhkan adalah sistem pembayaran elektronik yang didasarkan pada bukti kriptografi dibandingkan kepercayaan, yang memungkinkan dua pihak yang mau bertransaksi secara langsung satu sama lain tanpa perlu pihak ketiga yang terpercaya. Transaksi yang secara komputasi tidak praktis untuk dibalik akan melindungi penjual dari penipuan, dan mekanisme *escrow* rutin dapat dengan mudah diterapkan untuk melindungi pembeli. Hal inilah yang menyebabkan lahirnya Bitcoin, yaitu mengusulkan solusi untuk masalah pengeluaran ganda menggunakan server *timestamp* terdistribusi *peer-to-peer* untuk menghasilkan bukti komputasi dari urutan kronologis transaksi. Sistem ini aman selama node jujur secara kolektif mengontrol lebih banyak daya CPU daripada kelompok node penyerang yang bekerja sama (Nakamoto, 2008).

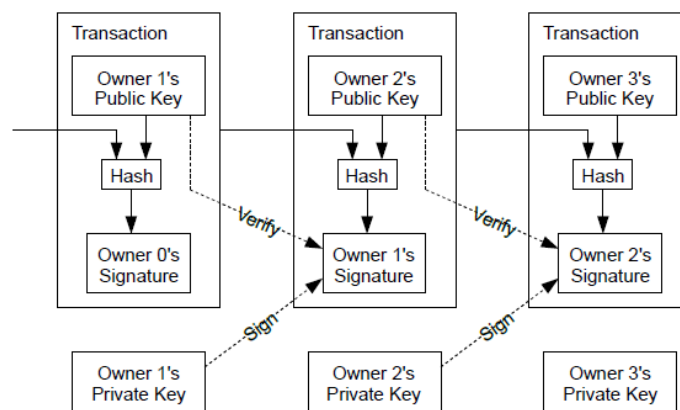
### **3.2.1 Cara Kerja Bitcoin Secara Teknis**

Bitcoin cukup mudah diakses karena seseorang dapat menukar mata uang apapun dengan Bitcoin, namun karena kelangkaannya Bitcoin memiliki keterbatasan likuiditas seperti akomodasi lainnya (Dyhrberg, 2016). Tidak seperti mata uang dan aset tradisional, Bitcoin mudah dibawa, dapat dibagi, dan tidak

dapat diubah. Bitcoin meningkatkan efisiensi sistem dan memungkinkan penyediaan layanan keuangan dengan biaya yang jauh lebih rendah, memberi pengguna lebih banyak kekuatan dan kebebasan (Blockchain, 2020).

- Transaksi

Koin elektronik didefinisikan sebagai rantai tanda tangan digital. Setiap pemilik mentransfer koin ke pemilik berikutnya dengan menandatangani *hash* secara digital dari transaksi sebelumnya dan kunci publik dari pemilik berikutnya serta menambahkannya ke bagian akhir koin. Seorang penerima pembayaran dapat memverifikasi tanda tangan untuk memverifikasi rantai kepemilikan.



**Gambar 3.2** Transaksi Bitcoin  
Sumber: (Nakamoto, 2008)

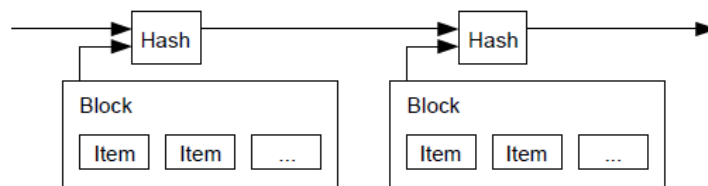
Masalahnya, tentu saja, penerima pembayaran tidak dapat memverifikasi bahwa salah satu pemilik tidak menghabiskan koin dua kali lipat. Solusinya adalah memperkenalkan otoritas pusat tepercaya, atau *mint*, yang memeriksa setiap transaksi untuk pengeluaran ganda. Setelah setiap transaksi, koin harus dikembalikan ke *mint* untuk mengeluarkan koin baru, dan hanya koin yang diterbitkan langsung dari *mint* yang dipercaya tidak akan dihabiskan dua kali lipat. Masalah dengan solusi ini adalah seluruh sistem uang tergantung pada perusahaan yang menjalankan *mint*, dengan setiap transaksi harus melalui mereka, seperti bank.

Dibutuhkan cara bagi penerima pembayaran untuk mengetahui bahwa pemilik sebelumnya tidak menandatangani transaksi sebelumnya. Untuk tujuan tersebut, transaksi paling awal adalah transaksi yang diperhitungkan, jadi tidak pembelanjaan ganda di kemudian hari tidak diperhatikan. Satu-satunya cara untuk

mengkonfirmasi tidak adanya transaksi adalah dengan mengetahui semua transaksi. Dalam model berbasis *mint*, *mint* mengetahui semua transaksi dan memutuskan mana yang lebih dulu. Untuk mencapai hal ini tanpa pihak yang tepercaya, transaksi harus diumumkan kepada publik, dan dibutuhkan sistem bagi peserta untuk menyepakati satu sejarah tunggal dari urutan dimana mereka diterima. Penerima perlu bukti bahwa pada saat setiap transaksi, mayoritas node setuju itu adalah yang pertama diterima.

- *Timestamp*

Solusi ini dimulai dengan server *timestamp*. Server *timestamp* bekerja dengan mengambil *hash* blok item yang akan dicatat berdasarkan waktu dan mempublikasikan *hash* secara luas. Cap waktu membuktikan bahwa data harus ada pada saat itu dan jelas, untuk masuk ke *hash*. Setiap stempel waktu menyertakan stempel waktu sebelumnya dalam *hash*-nya, membentuk rantai, dengan setiap stempel waktu tambahan memperkuat yang sebelumnya.



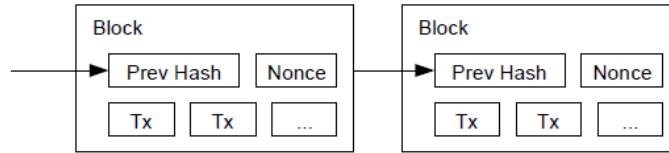
**Gambar 3.3** Server *Timestamp*  
 Sumber: (Nakamoto, 2008)

- *Proof-of-Work*

Untuk mengimplementasikan server *timestamp* terdistribusi berdasarkan *peer-to-peer*, kita perlu menggunakan sistem *proof-of-work* mirip dengan *Hashcash Adam Back*. Pembuktian kerja melibatkan pemindaian untuk nilai *hash* ketika *hash* dimulai dengan sejumlah bit nol. Pekerjaan rata-rata yang diperlukan adalah eksponensial dalam jumlah bit nol yang diperlukan dan dapat diverifikasi dengan mengeksekusi *hash* tunggal.

Untuk jaringan *timestamp* diterapkan *proof-of-work* dengan menambah *nonce* di blok hingga ditemukan nilai yang memberikan *hash* blok tersebut bit nol yang diperlukan. Setelah CPU dikeluarkan agar memenuhi *proof-of-work*, blok tidak dapat diubah tanpa mengulangi pekerjaan. Sebagai blok kemudian

dirantai setelahnya, pekerjaan untuk mengubah blok akan mencakup mengulang semua blok setelahnya.



**Gambar 3.4** Sistem *Proof-of-Work*

Sumber: (Nakamoto, 2008)

*Proof-of-work* memecahkan masalah menentukan representasi dalam pengambilan keputusan mayoritas. Jika mayoritas didasarkan pada *one-IP-address-one-vote*, bisa ditumbangkan oleh siapa saja yang dapat mengalokasikan banyak IP. Bukti kerja pada dasarnya adalah *one-CPU-one-vote*. Keputusan mayoritas diwakili oleh rantai terpanjang, yang memiliki upaya kerja bukti terbesar yang diinvestasikan di dalamnya. Jika mayoritas daya CPU dikendalikan oleh node yang jujur, rantai yang jujur akan tumbuh tercepat dan melampaui rantai yang bersaing. Untuk memodifikasi blok yang lalu, penyerang harus mengulangi pembuktian dari blok tersebut dan semua blok setelahnya dan kemudian mengejar ketinggalan dan melampaui pekerjaan dari node yang jujur. Nanti akan ditunjukkan bahwa kemungkinan seorang penyerang yang lebih lambat mengejar ketinggalan berkurang secara eksponensial ketika blok-blok berikutnya ditambahkan.

Untuk mengimbangi peningkatan kecepatan perangkat keras dan beragam minat dalam menjalankan node dari waktu ke waktu, kesulitan pembuktian kerja ditentukan oleh rata-rata bergerak yang menargetkan jumlah rata-rata blok per jam. Jika mereka dihasilkan terlalu cepat, kesulitannya meningkat.

- *Network*

Langkah-langkah untuk menjalankan jaringan adalah sebagai berikut:

1. Transaksi baru disiarkan ke semua node.
2. Setiap node mengumpulkan transaksi baru ke dalam blok.
3. Setiap node bekerja untuk menemukan bukti kerja yang sulit untuk bloknnya.
4. Ketika sebuah node menemukan *proof-of-work*, maka node tersebut menyiarkan blok ke semua node.

5. Node menerima blok hanya jika semua transaksi di dalamnya valid dan belum dibelanjakan.
6. Node menyatakan penerimaan mereka atas blok dengan bekerja membuat blok berikutnya dalam rantai, menggunakan *hash* dari blok yang diterima sebagai *hash* sebelumnya.

Node selalu menganggap rantai terpanjang untuk menjadi yang benar dan akan terus berupaya memperpanjangnya. Jika dua node menyiarkan versi berbeda dari blok berikutnya secara bersamaan, beberapa node mungkin menerima satu atau yang lain lebih dulu. Dalam hal ini, mereka bekerja pada yang pertama yang mereka terima, tetapi menyimpan cabang lain jika itu menjadi lebih lama. Ikatan akan rusak ketika bukti kerja berikutnya ditemukan dan satu cabang menjadi lebih panjang; node yang bekerja di cabang lain kemudian akan beralih ke yang lebih panjang.

Traksaksi baru tidak perlu menjangkau semua node. Selama transaksi mencapai banyak node maka transaksi tersebut akan masuk ke blok sebelumnya. Pemblokiran transaksi juga toleran terhadap pesan syang diberikan. Jika sebah node tidak menerima blok, maka akan diminta ketika menerima blok selanjutnya dan menyadi bahwa belum diterima sebelumnya.

- *Incentive*

Berdasarkan kesepakatan, taransaksi pertama dalam satu blok adalah transaksi khusus yang dimulai dengan koin baru yang dimiliki oleh pemilik blok tersebut. Ini menambah insentif bagi node untuk mendukung jaringan dan menyediakan cara untuk mendistribusikan koin ke sirkulasi karena tidak ada otoritas pusat untuk mengeluarkannya. Penambahan terus-menerus dari jumlah konstan koin baru adalah analog yang sama dengan penambang emas yaitu menghabiskan sumber daya untuk menambahkan emas ke sirkulasi. Dalam kasus ini, CPU dan listrik yang dikeluarkan.

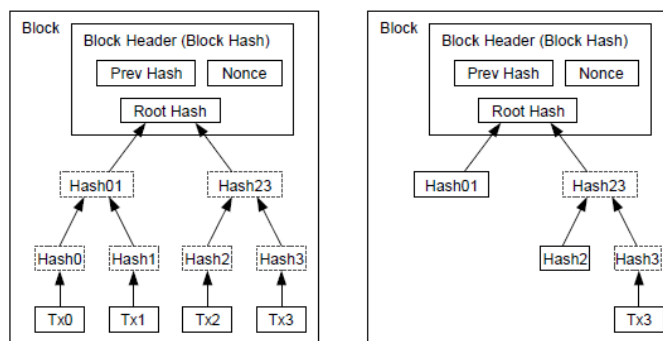
Insentif juga dapat didanai dengan biaya transaksi, jika nilai output transaksi kurang dari nilai inputnya. Perbedaannya adalah biaya transaksi yang ditambahkan ke nilai insentif blok yang berisi transaksi. Setelah sejumlah koin

yang telah ditentukan memasuki sirkulasi, insentif dapat beralih sepenuhnya ke biaya transaksi dan sepenuhnya bebas inflasi.

Insentif dapat membantu mendorong simpul untuk tetap sesuai. Jika penyerang mampu mengumpulkan lebih banyak kekuatan CPU daripada semua node yang sesuai, maka harus memilih antara menggunakannya untuk menipu orang dengan mencuri pembayaran atau menggunakannya untuk menghasilkan koin baru.

- *Reclaiming Disk Space*

Setelah transaksi terakhir dalam koin terkubur di bawah blok, transaksi yang dihabiskan sebelumnya dapat dibuang untuk menghemat ruang disk. Untuk memfasilitasi ini tanpa melanggar *hash* blok, transaksi *hash* dalam Pohon Merkle, dengan hanya root yang termasuk dalam *hash* blok. Blok-blok tua dapat dipadatkan dengan memotong cabang-cabang pohon. *Hash* interior tidak perlu disimpan.



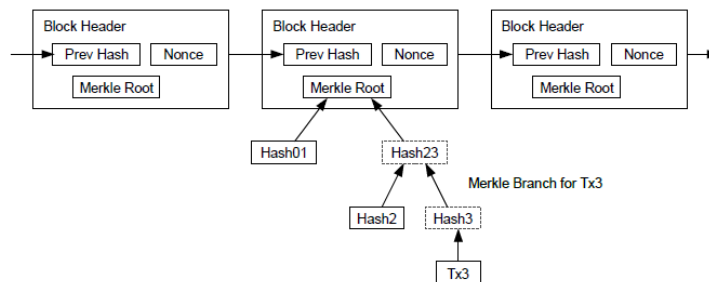
**Gambar 3.5** Transaksi *Hash* dalam Pohon Merkle dan Memangkas Tx2 dari blok  
 Sumber: (Nakamoto, 2008)

*Header* blok tanpa transaksi adalah sekitar 80 *byte*. Misalkan blok dihasilkan setiap 10 menit, maka memerlukan memori sekitar 4.2MB per tahun. Dengan sistem komputer yang biasanya menjual dengan 2GB RAM pada 2008, dan Hukum Moore memprediksi pertumbuhan saat ini sebesar 1,2GB per tahun, penyimpanan seharusnya tidak menjadi masalah bahkan jika *header* blok harus disimpan dalam memori.



- Verifikasi Pembayaran Sederhana

Terdapat kemungkinan saat memverifikasi pembayaran tidak menjalankan node jaringan penuh. Seorang pengguna hanya perlu menyimpan salinan *header* blok rantai bukti-kerja terlama, yang bisa didapatkan dengan menanyakan node jaringan sampai yakin memiliki rantai terpanjang, dan mendapatkan cabang Merkle yang menautkan transaksi ke blok tersebut. Saat itu *timestamp* masuk. *Timestamp* tidak dapat memeriksa transaksi untuk dirinya sendiri, tetapi dengan menghubungkannya ke suatu tempat dalam rantai maka dapat dilihat bahwa node jaringan telah menerimanya, dan blok ditambahkan setelah kemudian mengkonfirmasi jaringan telah menerimanya.

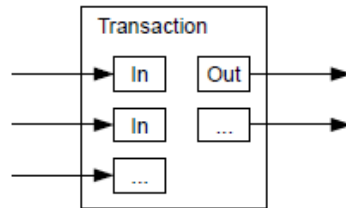


**Gambar 3.6**Rantai *Proof-of-Work*  
 Sumber: (Nakamoto, 2008)

Untuk mengkompensasi peningkatan kecepatan perangkat keras dan minat menjalankan titik komputasi yang berubah-ubah seiring perkembangan waktu, tingkat kesulitan dari *proof-of-work* ditentukan oleh target rata-rata jumlah blok per jam yang selalu berubah-ubah juga. Jika mereka dibuat terlalu cepat, maka tingkat kesulitan akan meningkat

- Menggabungkan dan Membagi Nilai

Meskipun mungkin untuk menangani koin secara individu, akan sulit untuk melakukan transaksi terpisah dalam transfer. Untuk memungkinkan nilai dipecah dan digabungkan, transaksi akan mengandung banyak input dan output. Biasanya akan ada input tunggal dari transaksi sebelumnya yang lebih besar atau beberapa input yang menggabungkan jumlah yang lebih kecil. Setidaknya terdapat paling banyak dua *output*, yang pertama untuk pembayaran dan kedua untuk pengembalian perubahan jika ada.

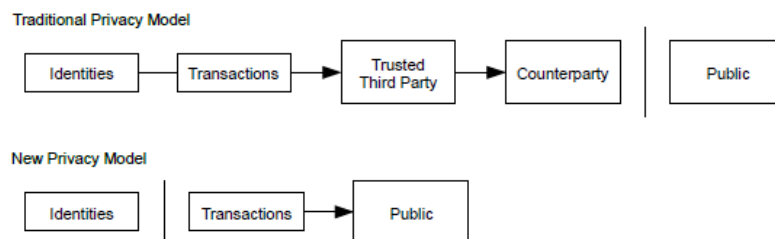


**Gambar 3.7**Rantai *Proof-of-Work*  
 Sumber: (Nakamoto, 2008)

Perlu dicatat bahwa fan-out, saat transaksi bergantung pada beberapa transaksi, dan transaksi-transaksi itu bergantung pada lebih banyak lagi, bukanlah masalah di sini. Tidak pernah ada kebutuhan untuk mengekstrak salinan lengkap lengkap dari riwayat transaksi.

- *Privacy*

Model perbankan tradisional mencapai tingkat privasi dengan membatasi akses informasi kepada pihak-pihak yang terlibat dan pihak ketiga yang tepercaya. Perlunya mengumumkan semua transaksi di depan umum menghalangi metode ini, tetapi privasi masih dapat dipertahankan dengan memutus aliran informasi di tempat lain: dengan menjaga kunci publik tetap anonim. Publik dapat melihat bahwa seseorang mengirim sejumlah uang kepada orang lain, tetapi tanpa informasi yang menghubungkan transaksi dengan siapa pun. Ini mirip dengan tingkat informasi yang dikeluarkan oleh bursa saham, di mana waktu dan ukuran perdagangan individu, "rekaman", diumumkan kepada publik, tetapi tanpa memberi tahu siapa pihak-pihak itu.



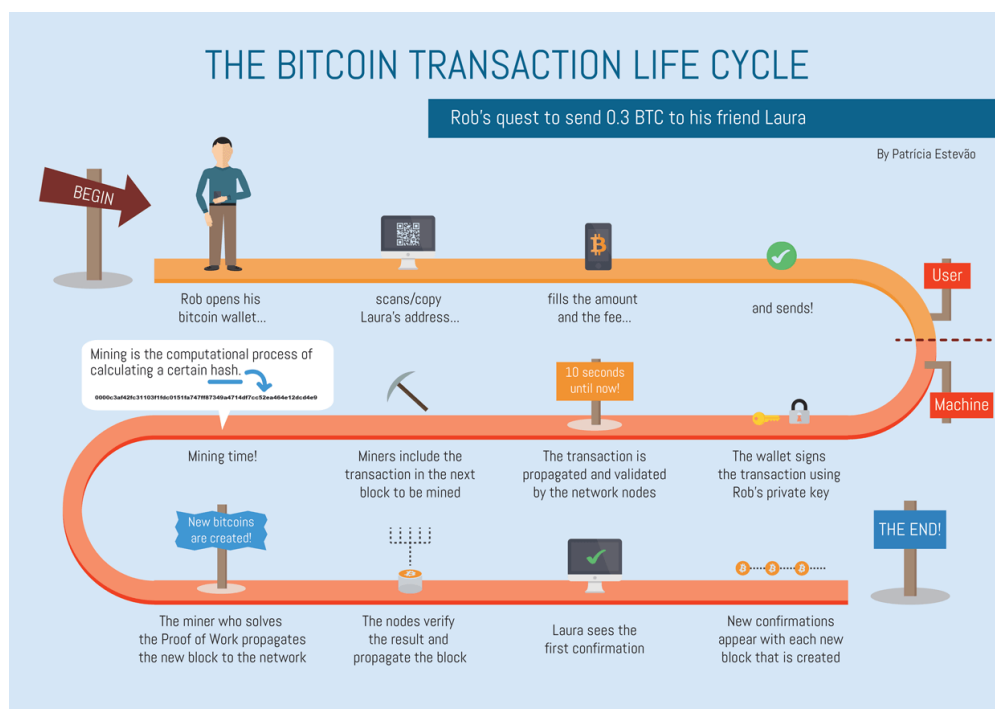
**Gambar 3.8**Model Privasi Bitcoin  
 Sumber: (Nakamoto, 2008)

Sebagai *firewall* tambahan, pasangan kunci baru harus digunakan untuk setiap transaksi agar mereka tidak tertaut ke pemilik bersama. Beberapa penautan masih tidak dapat dihindari dengan transaksi multi-input, yang tentu saja

mengungkapkan bahwa input mereka dimiliki oleh pemilik yang sama. Risikonya adalah jika pemilik kunci terungkap, menghubungkan dapat mengungkapkan transaksi lain milik pemilik yang sama.

### 3.2.2 Cara Kerja Bitcoin untuk Pengguna

Sebagai pengguna baru, pengguna tidak harus memahami detail teknisnya. Setelah menginstal *wallet* Bitcoin di komputer atau ponsel, secara otomatis akan membuat alamat Bitcoin pertama dan bisa membuat lebih banyak alamat bitcoin lagi kapanpun dibutuhkan. Pengguna bisa memberitahukan alamat Bitcoin itu kepada orang lain sehingga bisa melakukan transaksi. Sangat mirip dengan cara kerja *email*(bitcoin.org, 2020). Contoh dari transaksi dapat dilihat pada **Gambar 3.9**:



**Gambar 3.9** Contoh Transaksi Bitcoin

Sumber: (Livebitnews, 2017)

Rob ingin mengirim Bitcoin ke Laura. Transaksi Bitcoin itu sendiri terdiri dari tiga bagian, yaitu *input*, jumlah, dan *output*. *Input* merupakan catatan dari mana Rob menerima Bitcoin yang ingin dikirimkan ke Laura, jumlah merupakan banyaknya jumlah spesifik dari Bitcoin yang ingin dikirim ke Laura, sedangkan *output* merupakan alamat Bitcoin (*Bitcoin address*) dari Laura. Untuk

mengirimkan Bitcoin diperlukan akses kunci publik dan pribadi yang terkait dengan jumlah Bitcoin.

Kunci publik atau yang biasa disebut alamat bitcoin merupakan urutan acak huruf dan angka yang memiliki fungsi seperti alamat *email* atau nama pengguna media sosial. Karena bersifat publik maka dapat dibagikan dengan orang lain. Alamat inilah yang diberikan kepada orang lain jika ingin melakukan transaksi. Sedangkan kunci pribadi sama seperti kunci publik, namun harus dirahasiakan seperti kata sandi untuk *email* atau akun lainnya.

Alamat Bitcoin pada dasarnya merupakan kotak transparan. Orang lain dapat melihat apa yang ada di dalamnya, namun tidak dapat mengakses dana yang terdapat di dalamnya kecuali orang tersebut memiliki kunci pribadi. Dalam contoh di atas Rob ingin mengirimkan Bitcoin kepada Laura. Untuk melakukan hal ini Rob menggunakan kunci pribadinya untuk menandatangani pesan dengan detail spesifik transaksi, kemudian pesan tersebut dikirim ke blockchain berupa *input*, jumlah, dan *output*. Transaksi tersebut disebarkan ke jaringan Bitcoin dimana penambang memverifikasi bahwa kunci Rob dapat mengakses *input* yang digunakan sebagai kontrol. Proses konfirmasi ini dikenal sebagai penambangan (*miner*) karena membutuhkan sumber daya komputasi intensif dan penambang akan mendapatkan imbalan jika blok telah diselesaikan. Proses ini juga proses Bitcoin baru dibuat.

Semua transaksi Bitcoin harus diverifikasi oleh penambang di blockchain. Perlu diingat bahwa penambang tidak melakukan transaksi tapi menambang blok yang merupakan kumpulan transaksi. Ketika melakukan transaksi terdapat kemungkinan bahwa transaksi tersebut akan dikeluarkan dari blok saat ini dan ditunda hingga blok berikutnya disusun. Protokol Bitcoin memuat persyaratan agar setiap blok membutuhkan waktu sekitar 10 menit untuk menambang dan diatasi hingga 1MB. Pembatasan jumlah transaksi yang dapat memasuki blok secara efektif memperlambat waktu konfirmasi dan perluasan seluruh jaringan Bitcoin.

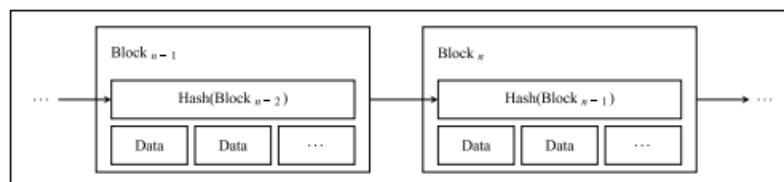
Biaya transaksi Bitcoin dihitung menggunakan berbagai faktor. Banyak aplikasi yang memungkinkan pengguna secara manual mengatur biaya transaksi.

Setiap transaksi yang tidak terhitung kepada penerima atau dikembalikan sebagai perubahan dimasukkan sebagai biaya. Biaya kepada penambang dan dapat digunakan untuk meningkatkan kecepatan konfirmasi dengan memberi biaya tambahan kepada penambang agar mengutamakan transaksi yang diinginkan.

### 3.3 Blockchain

Blockchain merupakan *database* terdistribusi yang menyimpan data catatan yang terus bertambah, dikendalikan oleh beberapa *entity*. Blockchain (*distributed ledger*) adalah sistem layanan yang dapat dipercaya ke sekelompok *node* atau pihak yang tidak saling percaya satu sama lain, umumnya Blockchain bertindak sebagai pihak ketiga yang terpercaya dan dapat diandalkan untuk mempertahankan keadaan bersama, menengahi pertukaran, dan menyediakan mesin komputasi yang aman (Cachin & Vukolić, 2017).

Kemudian Narayanan et al. (2016) dalam jurnal yang ditulis Jang dan Lee (2017) menjelaskan secara rinci bagaimana Blockchain dapat mencapai tujuan di atas dalam lingkungan Bitcoin. Seorang peserta dalam jaringan Bitcoin bertindak sebagai bagian dari sistem jaringan dengan menyediakan sumber daya perangkat keras dari komputer mereka sendiri, yang disebut “sistem terdistribusi”. Semua penerbitan dan transaksi uang dilakukan melalui jaringan *peer-to-peer*. Semua riwayat perdagangan dicatat dalam Blockchain dan dibagikan oleh jaringan, dan semua riwayat transaksi sebelumnya diverifikasi oleh semua peserta jaringan. Unit ini disebut “blok”, yang mencakup transaksi terbaru dan nilai hash dari blok sebelumnya, membuat data yang tidak dapat diubah dengan fungsi hash, dan ditunjukkan dari blok berikutnya. Urutan dari blok-blok tersebut dibuat dengan menyimpan *hash* kriptografi dari rekaman sebelumnya dalam setiap blok, seperti pada **Gambar 3.10**(Patel, 2019):



**Gambar 3.10**Blockchain Sederhana

Sumber: (Patel, 2019)

Penggunaan fungsi *hash* yang ireversibel sebagai mekanisme rantai juga berfungsi untuk memverifikasi integritas blok sebelumnya dan karakteristik inilah

yang memunculkan properti kunci pada *Blockchain* sebagai penyimpanan data permanen. Setiap percobaan untuk merusak data pada blok akan mudah dideteksi karena perubahan dari salah satu *hash* dalam blok akan menyebabkan perubahan pada semua *hash* berikutnya pada rantai (Patel, 2019). Algoritma ini disebut *Proof of Work* (PoW), dan kesulitannya secara otomatis diatur untuk memastikan bahwa masalah dapat diselesaikan dalam waktu sekitar 10 menit. PoW juga memberikan insentif untuk memotivasi peserta untuk mempertahankan nilai Bitcoin dengan membayar Bitcoin untuk peserta yang membuat blok (Jang & Lee, 2018).

Algoritma kesepakatan PoW dilengkapi dengan beberapa risiko yang melekat. Pertama, validitas blok dapat diintervensi ketika sebagian besar peserta diduduki oleh kelompok dengan tujuan spesifik yang disebut masalah 51%. Kedua, ketika *Blockchain* bercabang, banyak waktu yang diperlukan untuk membentuk *Blockchain* yang disepakati sampai rantai terpanjang dipilih setelah beberapa blok dibuat. Kondisi ini menyebabkan penundaan transaksi karena transaksi tidak dapat diselesaikan selama waktu itu. Terakhir, mungkin ada batas kapasitas *Blockchain* atau batas kinerja setiap node. Keamanan *Blockchain* saat ini dapat dipantau dengan mengamati variabel terukur dalam *Blockchain* dari <https://Blockchain.info/>.

Ada banyak variabel yang ditulis di buku besar publik diantaranya (Blockchain, 2017):

1. Average Block Size (Rata-rata Ukuran Blok): ukuran rata-rata blok dalam MB.
2. Transactions per Block (Transaksi per Blok): jumlah rata-rata transaksi per block.
3. Median Transaction Confirmation Time (with fee) (Median Konfirmasi Transaksi Waktu (dengan biaya)): waktu rata-rata untuk transaksi yang akan diterima ke blok tambang.
4. Hash Rate (Tingkat Hash): perkiraan jumlah hash tera per detik jaringan Bitcoin kinerja.
5. Difficulty (Kesulitan): ukuran relatif dari betapa sulitnya untuk menemukan blok baru.

6. Cost% of Transaction Volume (Biaya% Transaksi Volume): sebuah grafik yang menunjukkan penambang pendapatan sebagai persentase dari volume transaksi.
7. Mining Revenue (Penghasilan Pertambangan): jumlah total hadiah blok coinbase dan ongkos-ongkos transaksi yang dibayarkan kepada para penambang.
8. Unique Addresses (Alamat Unik): jumlah total alamat unik yang digunakan pada Blockchain Bitcoin.
9. Total Number of Transactions per Day (Total Jumlah Transaksi per Hari): jumlah transaksi Bitcoin harian dikonfirmasi.

### **3.4 Analisis Deskriptif**

Menurut Purwoto, analisis data deskriptif adalah cara mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa bermaksud membuat kesimpulan yang berlaku untuk umum/generalisasi. Ciri-ciri analisis data deskriptif yaitu penyajian data lebih ditekankan dalam bentuk tabel, grafik, dan ukuran-ukuran statistik seperti persentase, rata-rata, variansi, korelasi, dan angka indeks. Selain itu, analisis ini tidak menggunakan uji signifikansi dan taraf kesalahan karena tidak ada kesalahan generalisasi (A. Purwoto 2007).

Menurut Hasan dalam jurnal yang ditulis oleh Nasution (2017) analisis deskriptif adalah merupakan bentuk analisis data penelitian untuk menguji generalisasi hasil penelitian berdasarkan satu sample. Analisa deskriptif ini dilakukan dengan pengujian hipotesis deskriptif. Hasil analisisnya adalah apakah hipotesis penelitian dapat digeneralisasikan atau tidak. Jika hipotesis nol ( $H_0$ ) diterima, berarti hasil penelitian dapat digeneralisasikan. Analisis deskriptif ini menggunakan satu variabel atau lebih tapi bersifat mandiri, oleh karena itu analisis ini tidak berbentuk perbandingan atau hubungan. Selanjutnya menjelaskan : Statistik deskriptif atau statistik deduktif adalah bagian dari statistik mempelajari cara pengumpulan data dan penyajian data sehingga mudah dipahami. Statistik deskriptif hanya berhubungan dengan hal menguraikan atau memberikan keterangan-keterangan mengenai suatu data atau keadaan atau fenomena. Dengan

kata statistik deskriptif berfungsi menerangkan keadaan, gejala, atau persoalan (Nasution, 2017).

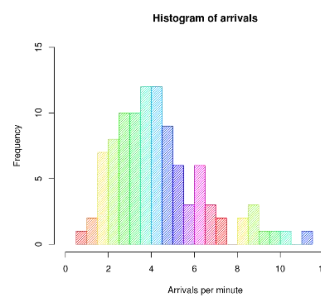
Analisis deskriptif dapat disajikan sebagai berikut:

### 3.4.1 Penyajian dalam Bentuk Visualisasi

Penyajian data dalam bentuk grafis terdiri dari beberapa macam, yaitu (Ardiyanto, 2019):

- **Histogram**

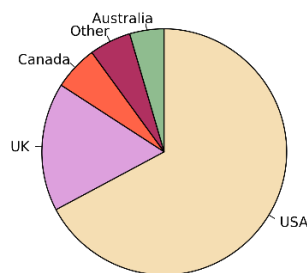
Histogram merupakan grafik yang berasal dari distribusi frekuensi suatu variabel dengan bentuk berupa balok. Penyajian data ini terdiri dari dua sumbu utama dengan sudut  $90^0$  sengan sumbu X sebagai absis dan sumbu Y sebagai ordinat. Lear balok merupakan jarak dari batas kelas interval, sedangkan tinggi balok menunjukkan besarnya frekuensi.



**Gambar 3.11** Histogram  
Sumber: (Wikipedia, 2020)

- **Pie Chart**

*Pie chart* atau diagram kue merupakan sebuah lingkaran yang dibagi menjadi beberapa sektor, dimana tiap sektor menunjukan bagian dari masing-masing data atau menyatakan besarnya persentase dari data tersebut.

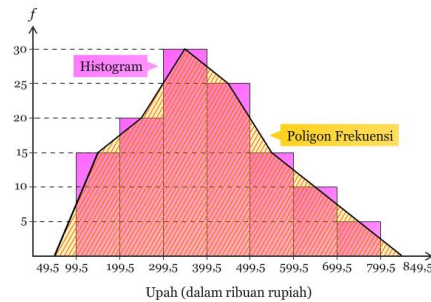


**Gambar 3.12** Pie Chart  
Sumber: (Wikipedia, 2020)



- Poligon

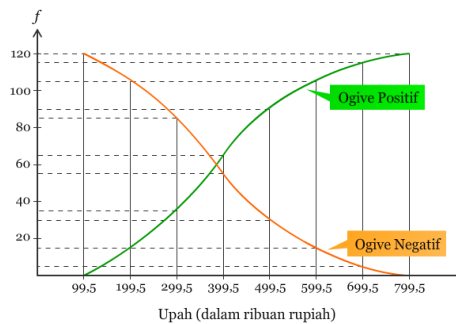
Polygon merupakan grafik yang berasal dari distribusi frekuensi ditampilkan berupa garis-garis patah yang diperoleh dengan menghubungkan puncak dari masing-masing nilai tengah kelas. Polygon digunakan untuk membandingkan bentuk dari dua distribusi.



**Gambar 3.13** Poligon  
Sumber: (Kristanto, 2013)

- Ogive

Ogive merupakan gambar dari distribusi frekuensi kumulatif suatu variabel. Ogive dapat berupa ogive positif dan ogive negatif.



**Gambar 3.14** Ogive  
Sumber: (Kristanto, 2013)

- Diagram Batang Daun (*Stem and Leaf*)

Diagram batang daun sama seperti histogram, namun informasi yang diperoleh lebih baik karena diagram batang daun memperlihatkan nilai-nilai dari pengamatan. Pada diagram ini ditampilkan bilangan-bilangan yang juga sebagai batang dan disebelah kanannya ditulis bilangan sisanya.

Batang	Daun
4	4 7
5	6 9
6	1 3 4 5 6 8
7	0 1 1 1 1 2 3 3 3 4 5 5 6 6 6 9
8	3 4 4 5 5 9
9	1 4 5 7

**Gambar 3.15** Diagram Batang Daun  
 Sumber: (Kristanto, 2013)

### 3.4.2 Ukuran Pemusatan Data

Statistika deskriptif yang biasa digunakan dalam menggambarkan kondisi suatu data adalah ukuran pemusatan data. Terdapat tiga jenis ukuran pemusatan data (tendensi *central*) yaitu (Tiyas, 2019):

- Mean

Mean atau Rata-rata hitung adalah suatu metode yang paling banyak dipakai dalam menunjukkan ukuran tendensi sentral. *Mean* ini dihitung dengan menjumlahkan seluruh nilai data pengamatan lalu dibagi dengan banyaknya data.

- Median

Median adalah nilai yang membagi himpunan pengamatan menjadi dua bagian yang sama besar atau 50% dari pengamatan yang berada di bawah median serta 50% lagi berada di atas median. Median dari  $n$  pengukuran/ pengamatan  $x_1, x_2, \dots, x_n$  merupakan suatu nilai pengamatan yang berada di tengah gugus data sesudah data tersebut diurutkan. Jika banyaknya pengamatan ( $n$ ) ganjil, median berada tepat ditengah gugus data, sementara jika  $n$  genap, median didapatkan dengan cara interpolasi. Yakni cara dimana rata-rata dari dua data yang berada di tengah gugus data.

- Modus

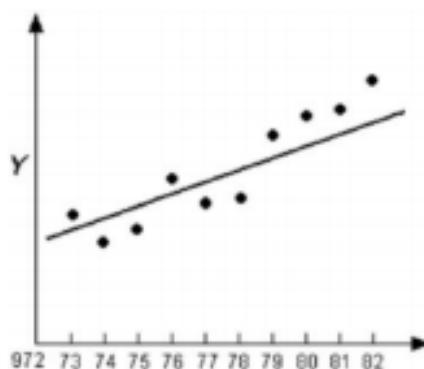
Modus adalah suatu data yang paling sering muncul atau terjadi. Untuk menentukan adanya modus, pertama kali dengan menyusun data dalam urutan meningkat atau sebaliknya. Lalu diikuti dengan menghitung frekuensinya. Nilai yang frekuensinya paling besar atau sering muncul itulah yang dinamakan sebagai modus. Modus dipakai baik untuk tipe data numerik maupun data kategoris.

### 3.5 Time Series

Menurut Lincoln Arsyad (2009) dalam jurnal yang ditulis oleh Nawangwulan dan Angesti (2016), setiap variabel yang terdiri dari data yang dikumpulkan, dicatat atau diobservasi sepanjang waktu yang berurutan disebut data runtut waktu (time series). Analisis runtut waktu dilakukan untuk menemukan pola pertumbuhan atau perubahan masa lalu, yang dapat digunakan untuk memperkirakan pola pada masa yang akan datang. Analisis ini cukup penting dalam proses peramalan dan membantu mengurangi kesalahan dalam peramalan tersebut. Dalam analisis runtut waktu terdapat 4 komponen yaitu (Nawangwulan & Angesti, 2016):

#### 3.5.1 Trend

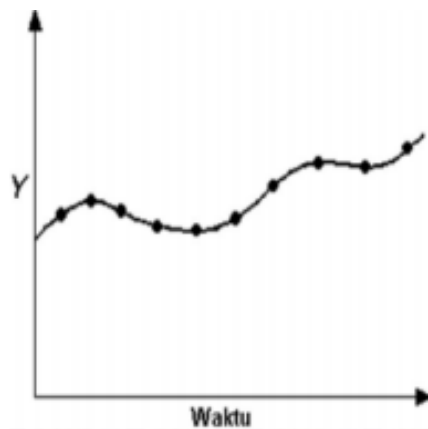
*Trend* ialah perkembangan jangka panjang dalam suatu runtut waktu yang dapat digambarkan dengan sebuah garis lurus atau sebuah kurva yang menghasilkan atau mempengaruhi *trend* dari suatu seri dapat berupa perubahan populasi, perubahan harga, perubahan teknologi, dan peningkatan produktivitas. Dalam analisis trend variabel bebasnya adalah waktu. Jika grafik berbentuk garis lurus dalam skala aritmatika, maka peneliti akan menggunakan persamaan linier dalam analisis datanya. Jika data dinyatakan dalam bentuk semilogaritma dan terbentuk grafik dengan garis lurus, maka peneliti akan memilih model eksponensial dalam datanya. Metode untuk menjelaskan trend linier ialah metode kuadrat terkecil.



**Gambar 3.16** Pola Data *Trend*  
Sumber: (Andini & Auristandi, 2016)

### 3.5.2 Variasi Siklis

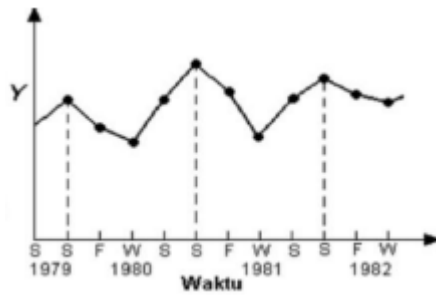
Komponen siklis ialah suatu seri fluktuasi seperti gelombang atau siklus yang mempengaruhi keadaan ekonomi selama lebih dari satu tahun. Hal tersebut dapat dilihat dari perbedaan antara nilai yang diharapkan (trend) dengan nilai yang sebenarnya yaitu variasi residual yang berfluktuasi sekitar trend. Komponen siklis dan tak beraturan dari data runtut waktu dapat diidentifikasi dengan cara menghilangkan pengaruh trend, metode ini disebut metode residual (residual method). Tahap metode residual tergantung pada dimulainya menganalisis dengan data tahunan, bulanan, atau kuartalan. Jika data yang digunakan ialah data bulanan atau kuartalan, maka pengaruh trend dan komponen-komponen musiman harus dihilangkan. Jika datanya ialah data tahunan, maka pengaruh trend yang dihilangkan.



**Gambar 3.17** Pola Data Siklis  
Sumber: (Andini & Auristandi, 2016)

### 3.5.3 Musiman

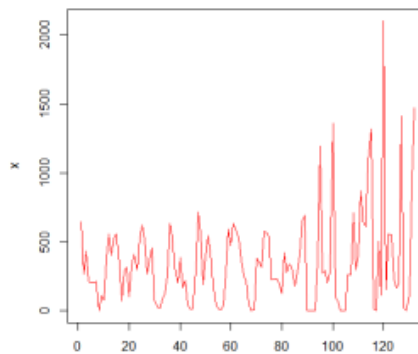
Fluktuasi musiman biasanya dijumpai pada data yang dikelompokkan secara kuartalan, bulanan, atau mingguan. Variasi musiman ini menggambarkan pola perubahan yang berulang secara teratur dari waktu ke waktu. Komponen musiman runtut waktu diukur dalam bentuk angka indeks. Interpretasi angka indeks ini, yang mencerminkan besarnya pengaruh musiman untuk suatu segmen tahun tertentu, berkaitan dengan perbandingan nilai terhitung atau nilai yang diharapkan dari segmen tersebut (bulan, kuartal, dan sebagainya).



**Gambar 3.18** Pola Data Musiman  
 Sumber: (Andini & Auristandi, 2016)

### 3.5.4 Fluktuasi Tak Beraturan

Komponen tidak beraturan terbentuk dari fluktuasi-fluktuasi yang disebabkan oleh peristiwa-peristiwa yang tidak terduga seperti perubahan cuaca, pemogokan, perang, pemilihan umum, rumor perang, dan lain-lain.



**Gambar 3.19** Pola Data Fluktuasi Tak Beraturan  
 Sumber: (Darmawan & dkk, 2016)

## 3.6 Artificial Neural Networks

*Artificial Neural Networks* (ANN) adalah teknik pembelajaran mesin yang terinspirasi oleh jaringan saraf biologis di otak manusia. Penentuan jumlah lapisan dan jumlah *neuron* di lapisan tersembunyi dan koneksi di antara neuron tersebut adalah hal yang penting (Saritas & Yasar, 2019). *Neuron* mempunyai karakteristik yang sama dalam ANN, terdiri dalam kelompok-kelompok yang disebut *layer*. Neuron-neuron dalam satu *layer* terhubung dalam *layer-layer*lainnya yang berdekatan. Kekuatan hubungan antar *neuron* yang berdekatan direpresentasikan dalam kekuatan hubungan atau bobot (Dharma, Putera, & Ardana, 2011).

Langkah penting dalam pengembangan sebuah model ANN adalah penentuan bobot matrik melalui pelatihan (*training*). Ada dua tipe mekanisme

training yaitu *supervised training* dan *unsupervised training*. *Supervised training* memerlukan supervisi dari luar untuk memandu proses *training*. Algoritma ini menggunakan sejumlah pasangan data *input-output* yang dipergunakan sebagai contoh, dimana data yang dipergunakan sebagai contoh sebaiknya menggunakan data yang sudah diketahui kebenarannya. *Output* dari jaringan lalu dibandingkan dengan data *output* yang diharapkan untuk mendapatkan selisih antara *output* perkiraan dengan *output* sebenarnya. Selisih inilah yang dipergunakan untuk mengubah bobot jaringan sehingga diperoleh *output* yang sama atau mendekati target. Mekanisme sebuah *supervised training* yang disebut algoritma *back-propagation training* umumnya banyak digunakan dalam aplikasi-aplikasi *engineering*. Karena ANN tidak mempertimbangkan fisik dari permasalahan, ANN merupakan sebuah model *blackbox* namun dapat mendeteksi proses fisik dalam model ANN yang sudah *training* (Dharma, Putera, & Ardana, 2011).

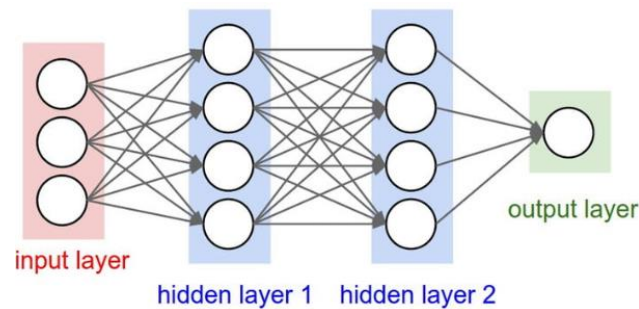
Jaringan saraf terdiri dari lapisan *input* dan *output*, serta (dalam kebanyakan kasus) *hidden layer* yang terdiri dari unit yang mengubah *input* menjadi sesuatu yang dapat digunakan oleh lapisan *output*. *Input*, *output*, dan *hidden layer* adalah alat yang sangat baik untuk menemukan pola yang terlalu rumit atau banyak bagi programmer untuk mengekstraksi dan mengajar mesin untuk mengenali.

Sementara jaringan saraf (juga disebut "*perceptrons*") telah ada sejak 1940-an, hanya dalam beberapa dekade terakhir di mana mereka telah menjadi bagian utama dari kecerdasan buatan. Hal ini disebabkan oleh temuan teknik yang disebut "*backpropagation*" yang memungkinkan jaringan untuk menyesuaikan lapisan *neuron* tersembunyi mereka dalam situasi di mana hasilnya tidak sesuai dengan harapan peneliti misalnya seperti jaringan yang dirancang untuk mengenali kucing, tapi mengidentifikasi kelinci.

Kemajuan penting lainnya adalah munculnya *deep learning neural networks*, dimana lapisan yang berbeda dari jaringan *multilayer* mengekstraksi fitur yang berbeda hingga dapat mengenali apa yang dicari.

Untuk ide dasar tentang bagaimana *deep learning neural networks*, bayangkan garis pabrik. Setelah bahan baku (kumpulan data) dimasukkan, mereka kemudian diturunkan melalui ban berjalan, dengan setiap pemberhentian

berikutnya atau lapisan mengekstraksi serangkaian fitur tingkat tinggi yang berbeda. Jika jaringan dimaksudkan untuk mengenali objek, lapisan pertama mungkin menganalisis kecerahan pikselnya.



**Gambar 3.20**BlockchainSederhana  
Sumber: (Patel, 2019)

Lapisan selanjutnya dapat mengidentifikasi setiap tepi dalam gambar, berdasarkan garis piksel yang sama. Setelah ini, layer lain mungkin mengenali tekstur dan bentuk, dan seterusnya. Pada saat lapisan keempat atau kelima tercapai, jaring pembelajaran yang dalam akan menciptakan detektor fitur yang kompleks. Dapat diketahui bahwa elemen gambar tertentu (seperti sepasang mata, hidung, dan mulut) umumnya ditemukan bersama. Setelah ini dilakukan, para peneliti yang telah melatih jaringan dapat memberikan label ke output, dan kemudian menggunakan backpropagation untuk memperbaiki kesalahan yang telah dibuat. Setelah beberapa saat, jaringan dapat melakukan tugas klasifikasi sendiri tanpa memerlukan manusia untuk membantu setiap saat.

Namun pada *Artificial Neural Network* tidak dapat mengingat informasi yang diberikan sebelumnya. Hal ini diatasi oleh *Recurrent Neural Networks* (RNN). RNN adalah jaringan dengan loop di dalamnya, memungkinkan informasi untuk bertahan.

### **3.7 Recurrent Neural Network**

RNN adalah sejenis jaringan saraf tiruan yang memiliki kemampuan yang untuk melihat korelasi tersembunyi yang terjadi pada data dalam aplikasi untuk pengenalan suara, pemrosesan bahasa alami, dan prediksi deret waktu. RNN sangat baik untuk masalah pemodelan urutan dengan beroperasi pada informasi

input serta jejak informasi yang diperoleh sebelumnya karena koneksi berulang (Tian & dkk, 2018).

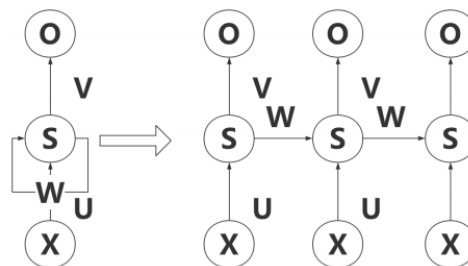
$$S_t = f((U * X_t) + (W * S_{t-1})) \quad (3.1)$$

$$O_t = g(V * S_t) \quad (3.2)$$

dengan  $S_t$  adalah memori jaringan pada waktu  $t$ ;  $U$ ,  $W$ , dan  $V$  adalah matriks bobot berbagi di setiap lapisan;  $X_t$  dan  $O_t$  mewakili input dan output pada waktu  $t$ ; dan  $f(\cdot)$  dan  $g(\cdot)$  mewakili fungsi nonlinear.

*Recurrent Neural Network* (RNN) terdiri dari unit *input*, unit *output*, dan unit tersembunyi. Model RNN pada dasarnya memiliki aliran informasi satu arah dari unit *input* ke unit tersembunyi, dan sintesis aliran informasi satu arah dari unit tersembunyi sementara sebelum ke unit tersembunyi waktu saat ini. Unit tersembunyi dapat dilihat sebagai penyimpanan seluruh jaringan, yang mengingat informasi dari ujung ke ujung (Yin, 2017). Seperti yang ditunjukkan pada

**Gambar 3.3:**



**Gambar 3.2** Struktur RNN  
Sumber: (Tian & dkk, 2018)

Berbeda dengan koneksi bobot yang dibangun antara lapisan dalam jaringan saraf dasar, RNN dapat menggunakan status internal (memori) untuk memproses urutan *input*. Status tersembunyi menangkap informasi pada titik waktu sebelumnya, dan *output* berasal dari waktu saat ini dan memori sebelumnya. RNN berkinerja baik ketika *output* dekat dengan *input* terkait informasi dari node sebelumnya diteruskan ke node berikutnya. Secara teori, RNN juga mampu menangani ketergantungan jangka panjang. Namun, dalam aplikasi praktis, RNN tidak dapat menyimpan informasi sebelumnya dengan baik ketika interval waktu lama karena masalah gradien menghilang. Untuk mengatasi kelemahan ini dan

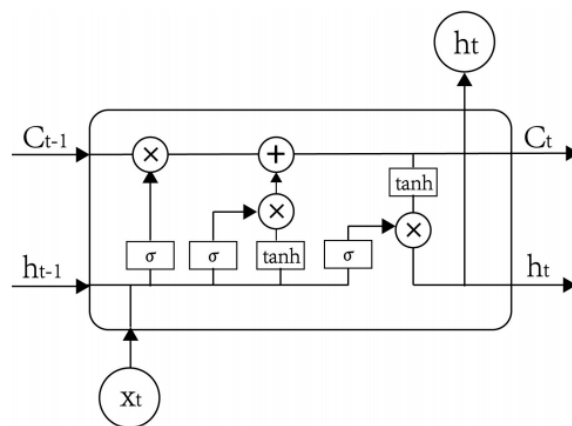


meningkatkan kinerja RNN, sebuah tipe khusus arsitektur RNN yang disebut LSTM diusulkan oleh Hochreiter & Schmidhuber (1997) (Tian & dkk, 2018).

### 3.8 Long Short Term Memory

Long Short Term Memory (LSTM) menggunakan salah satu bentuk RNN yang paling umum yang dimaksudkan untuk menghindari masalah ketergantungan jangka panjang dan sesuai untuk memproses serta memprediksi deret waktu. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, model LSTM terdiri dari serangkaian sel memori unik yang menggantikan neuron lapisan tersembunyi dari RNN, dan kuncinya adalah keadaan sel-sel memori. Model LSTM menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur pintunya mencakup gerbang *input*, *forget gate*, dan gerbang *output*. Setiap sel memori memiliki tiga lapisan sigmoid dan satu lapisan *tanh*. Gambar 3.4 menampilkan struktur sel memori LSTM (Qiu, Wang, & Zhou, 2020).

Lapisan sigmoid menampilkan angka antara nol dan satu, menggambarkan berapa banyak masing-masing komponen yang harus dilewati. Nilai nol berarti "jangan biarkan apa pun lewat" sementara nilai satu berarti "biarkan semuanya lewat". Sigmoid akan bernilai nol jika terletak pada rentang  $0 \leq \sigma \leq 0,5$  dan bernilai satu jika terletak pada rentang  $0,5 < \sigma \leq 1$  (R, 2017).



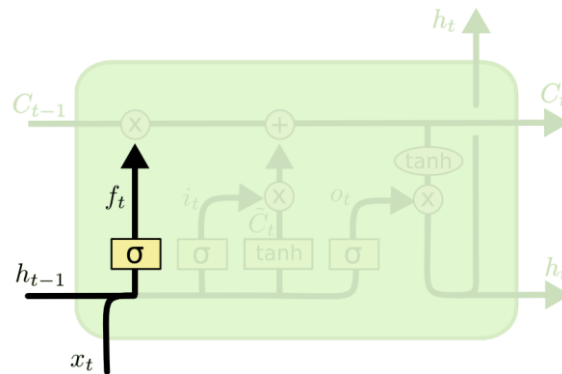
**Gambar 3.22** Struktur LSTM  
Sumber: (Qiu, Wang, & Zhou, 2020)

Fungsi sigmoid dan tanh ditunjukkan pada persamaan (3.3) dan (3.4) (Ma, 2015):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3.3)$$

$$\tanh(x) = 2\sigma(2x) - 1 \quad (3.4)$$

Dimana  $x$  adalah data input dan  $e$  adalah konstanta matematika.



**Gambar 3.23** Struktur LSTM untuk *Forget Gate*  
Sumber: (Olah, 2015)

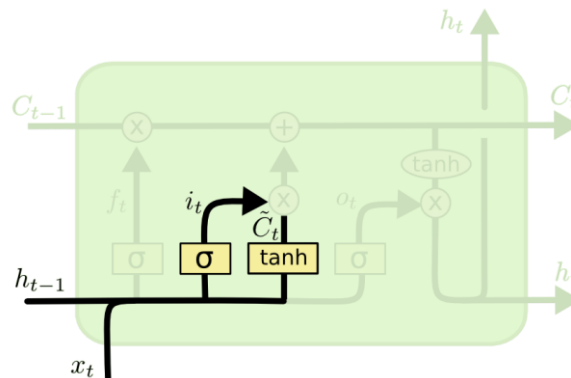
*Forget gate* dalam unit LSTM menentukan informasi status sel mana yang dibuang dari model. Seperti yang ditunjukkan pada **Gambar 3.24**, sel memori menerima output  $h_{t-1}$  dari momen sebelumnya dan informasi eksternal  $x_t$  dari momen saat ini sebagai *input* dan menggabungkannya dalam vektor panjang  $[h_{t-1}, x_t]$ . Notasi  $[h_{t-1}, x_t]$  merupakan operasi konkatenasi, artinya menambahkan baris dari  $x_t$  dengan baris dari  $h_{t-1}$  melalui transformasi  $\sigma$  menjadi:

$$f_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_f) \quad (3.5)$$

Keterangan:

- $f_t$  : *Forget gate*
- $\sigma$  : Fungsi sigmoid
- $W_t$  : Nilai *weight* untuk *forget gate*
- $h_{t-1}$  : Nilai *output* sebelum orde ke-t
- $x_t$  : Nilai *input* pada orde ke-t
- $b_f$  : Nilai bias pada *forget gate*

Fungsi utama dari *forget gate* adalah untuk merekam seberapa banyak status sel  $C_{t-1}$  dari waktu sebelumnya dicadangkan ke status sel  $C_t$  dari waktu saat ini. Gerbang akan menampilkan nilai antara 0 dan 1 berdasarkan pada  $h_{t-1}$  dan  $x_t$ , dimana 1 menunjukkan reservasi lengkap dan 0 menunjukkan pembuangan lengkap.



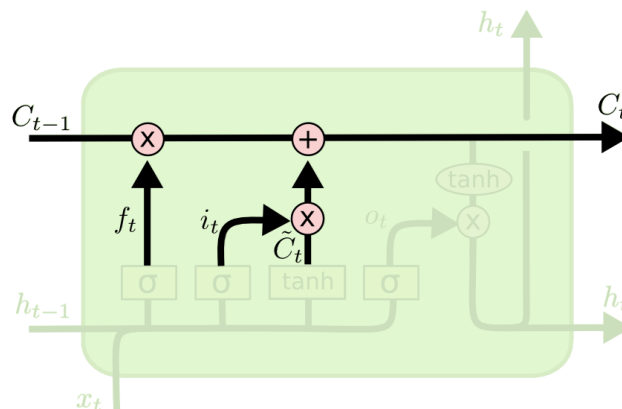
**Gambar 3.24** Struktur LSTM untuk *Input Gate*  
 Sumber: (Olah, 2015)

Gerbang input menentukan berapa banyak input jaringan waktu saat ini  $x_t$  dicadangkan ke dalam status sel  $C_t$ , yang mencegah konten tidak signifikan dari memasuki sel memori. Gerbang input memiliki dua fungsi, pertama adalah menemukan keadaan sel yang harus diperbarui; nilai yang akan diperbarui dipilih oleh lapisan sigmoid, seperti dalam Persamaan (3.6):

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.6)$$

Keterangan:

- $i_t$  : *Input gate*
- $\sigma$  : Fungsi sigmoid
- $W_i$  : Nilai *weight* untuk *input gate*
- $h_{t-1}$  : Nilai *output* sebelum orde ke-t
- $x_t$  : Nilai *input* pada orde ke-t
- $b_i$  : Nilai bias pada *input gate*



**Gambar 3.25** Struktur LSTM untuk *Cell State*  
 Sumber: (Olah, 2015)

Sedangkan fungsi lainnya adalah untuk memperbarui informasi ke b diperbarui ke keadaan sel. Vektor kandidat baru  $\hat{C}_t$  dibuat melalui lapisan  $\tanh$  untuk mengontrol berapa banyak informasi baru ditambahkan, seperti dalam Persamaan (3.7). Dan Persamaan (3.8) digunakan untuk memperbarui keadaan sel dari sel memori

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3.7)$$

Keterangan:

$\hat{C}_t$  : Nilai baru yang ditambahkan ke *cell state*

$\tanh$  : Fungsi tanh

$W_c$  : Nilai *weight* untuk *cell state*

$h_{t-1}$  : Nilai *output* sebelum orde ke-t

$x_t$  : Nilai *input* pada orde ke-t

$b_c$  : Nilai bias pada *cell state*

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (3.8)$$

Keterangan:

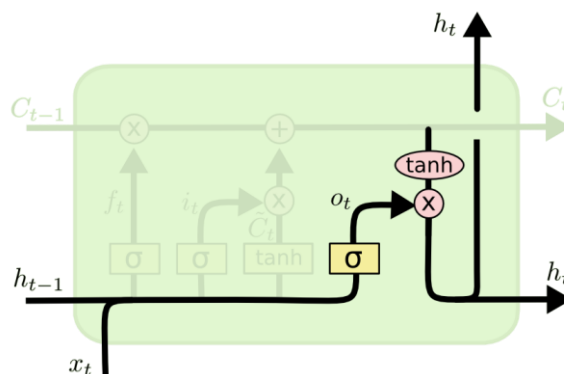
$C_t$  : Nilai baru yang ditambahkan ke *cell state*

$f_t$  : *Forget gate*

$C_{t-1}$  : *Cell state* sebelum orde ke-t

$i_t$  : *Input gate*

$\hat{C}_t$  : Nilai baru yang ditambahkan ke *cell state*



**Gambar 3.26** Struktur LSTM untuk *Output Gate*  
Sumber: (Olah, 2015)

Gerbang *output* mengontrol seberapa banyak keadaan sel saat ini dibuang. Informasi keluaran pertama-tama ditentukan oleh lapisan sigmoid, dan kemudian

keadaan sel diproses oleh *tanh* dan dikalikan dengan keluaran lapisan sigmoid untuk mendapatkan bagian keluaran akhir:

$$O_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.9)$$

Keterangan:

- $O_t$  : *Output gate*
- $\sigma$  : Fungsi sigmoid
- $W_o$  : Nilai *weight* untuk *output gate*
- $h_{t-1}$  : Nilai *output* sebelum orde ke-t
- $x_t$  : Nilai *input* pada orde ke-t
- $b_o$  : Nilai bias pada *output gate*

Nilai output akhir sel didefinisikan sebagai:

$$h_t = O_t * \tanh(C_t) \quad (3.10)$$

Keterangan:

- $h_t$  : Nilai *output* orde ke t
- $O_t$  : *Output gate*
- $\tanh$  : Fungsi tanh
- $C_t$  : *Cell state*

Algoritma yang digunakan dalam LSTM adalah sebagai berikut (Brownlee, 2017):

1. Menghitung nilai sigmoid dan tanh.
2. Mengubah data yang akan digunakan menjadi *supervised learning problem*. *Supervised learning problem* adalah algoritma yang digunakan untuk mempelajari fungsi pemetaan dari input ke output. Karena data yang digunakan adalah data *time series* maka inputnya adalah data hari kemarin (t-1) dan outputnya adalah data hari ini (t)
3. Melakukan normalisasi menggunakan *min-max scaling*.
4. Menghapus variabel yang tidak dibutuhkan.
5. Membagi data menjadi data *training* dan *testing*.

6. Membuat model LSTM yang akan digunakan dengan menentukan banyaknya *hidden layer, neuron* dan *epoch* yang akan digunakan. Dalam model LSTM akan dilakukan proses berupa:
  - Menghitung nilai *forget gate*.
  - Menghitung nilai *input gate*.
  - Memperbarui memori yang berada pada *cell*.
  - Menghitung *output gate* dan nilai *output* akhir.
7. Melakukan evaluasi terhadap model LSTM yang telah dibuat.

### 3.9 Parameter Evaluasi

Menurut Carlo Vercilis (Vercellis, 2009) terdapat dua alasan utama untuk melihat tingkat akurasi pada prediksi model *time series*. Pertama, pada tahap pengembangan dan identifikasi model, ukuran akurasi digunakan untuk membandingkan model-model alternatif satu sama lain dan untuk menentukan nilai parameter yang muncul dalam ekspresi untuk fungsi prediksi. Untuk mengidentifikasi model prediksi yang paling akurat, masing-masing model dianggap diterapkan pada data masa lalu, dan model dengan total nilai error minimum yang dipilih.

Kedua, setelah model prediksi telah dikembangkan dan digunakan untuk menghasilkan prediksi untuk masa mendatang, perlu untuk secara berkala menilai keakuratan, untuk mendeteksi kelainan dan kekurangan dalam model yang mungkin timbul di lain waktu. Evaluasi keakuratan prediksi pada tahap ini membuat mungkin untuk menentukan apakah model masih akurat atau memerlukan suatu revisi. Untuk mengevaluasi akurasi dan peramalan kinerja model berbeda, penelitian ini mengadopsi dua indeks evaluasi: *Mean Square Error* (MSE) atau *Root Mean Square Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE). Formula untuk menghitung indeks ini diberikan di bawah ini (Budiman, 2016):

#### 3.9.1 Mean Square Error

*Mean Square Error* (MSE) adalah penjumlahan kuadrat *error* atau selisih antara nilai sebenarnya dan nilai prediksi, kemudian membagi hasil penjumlahan

tersebut dengan banyaknya waktu peramalan. Persamaannya ditulis seperti di bawah ini:

$$MSE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n} \quad (3.11)$$

### 3.9.2 Root Mean Square Error

*Root Mean Square Error* adalah penjumlahan kuadrat *error* atau selisih antara nilai sebenarnya dan nilai prediksi, serta membagi hasil penjumlahan tersebut dengan banyaknya waktu peramalan dan menarik akarnya. Persamaannya ditulis seperti di bawah ini:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (3.12)$$

### 3.9.3 Mean Absolute Percentage Error

*Mean Absolute Percentage Error* (MAPE) adalah nilai *absolute* dari persentase *error* data teradap *mean*. Persamaannya ditulis seperti di bawah ini:

$$MAPE = \frac{\sum_{i=1}^n \frac{|\hat{Y}_i - Y_i|}{Y_i}}{n} * 100\% \quad (3.13)$$

## 3.10 Sistem Pengerjaan LSTM

Langkah-langkah pengerjaan LSTM terdiri dari *preprocessing data*, inisialisasi parameter, *training LSTM network*, dan melakukan uji terhadap data *testing*. Langkah-langkah tersebut dijelaskan sebagai berikut (Aldi, Jondri, & Aditsania, 2018):

### 3.10.1 Preprocessing Data

Untuk meminimalkan *error*, dilakukan normalisasi pada dataset dengan mengubah data aktual menjadi nilai dengan range interval [0,1]. Teknik normalisasi yang digunakan menggunakan *min-max scaling*. Adapun untuk rumus normalisasi *min-max scaling* adalah :

$$x' = \frac{x - \min_x}{\max_x - \min_x} \quad (3.14)$$

dimana  $x'$  adalah hasil normalisasi,  $x$  adalah data yang akan dinormalisasikan,  $\min_x$  adalah nilai minimum dari keseluruhan data, dan  $\max_x$  adalah nilai maksimum dari keseluruhan data.

### 3.10.2 Inisialisasi Parameter

Setelah dilakukan *preprocessing* data selanjutnya menentukan inisialisasi parameter-parameter dasar yang dibutuhkan, antara lain:

- Jumlah *hidden layer*
- Jumlah *Neuron* pada *hidden layer*
- Target nilai *error* yang berupa *Mean Square Error* (MSE)
- *Epoch* maksimum

### 3.10.3 Training LSTM Network

Berikut adalah penjelasan dari proses *training* pada LSTM:

- Hitung semua fungsi *gates* unit pada setiap neurons. Dengan berurut fungsi *gates* yang akan dihitung adalah *forget gates* dengan persamaan (3.4), fungsi *input gates* dengan persamaan (3.5) dan (3.6), fungsi *cell gates* dengan persamaan (3.7), dan yang terakhir fungsi *output gates* dengan persamaan (3.8) dan (3.9).
- Jika telah melakukan perulangan sebanyak epoch yang telah ditentukan, maka berhenti. Jika belum, akan dilakukan optimasi dengan optimasi Adam dan memperbarui bobot dan bias pada sistem, kemudian kembali ke langkah dua.

### 3.10.4 Testing

Model yang telah didapatkan pada proses training akan diuji dengan menggunakan data testing yang telah didapat dari preprocessing data, dengan metode akurasi yang digunakan menggunakan MSE, RMSE, dan MAPE dengan persamaan (3.10), (3.11), dan (3.12).

Sebelum menghitung akurasi hasil prediksi, terlebih dahulu dilakukan proses denormalisasi hasil output dari RNN LSTM. Denormalisasi dilakukan untuk mendapatkan nilai real dari hasil prediksi yang diberikan. Sementara itu akurasi dihitung dengan melihat persen akurasi dari hasil prediksi. Proses denormalisasi dilakukan dengan persamaan (Rizal & Soraya, 2018):

$$dn_i = ((n_i)(x_{max} - x_{min})) + (x_{min}) \quad (3.15)$$

Keterangan:

$dn_i$  : Data hasil denormalisasi



$n_i$  : Data ke- $i$   
 $x_{max}$  : Data dengan nilai maksimum  
 $x_{min}$  : Data dengan nilai minimum

### 3.11 Adaptive Momen Estimation (Adam)

Optimasi berbasis gradien stokastik adalah kepentingan praktis inti dalam banyak bidang sains dan teknik. Banyak masalah dalam bidang ini dapat dilemparkan sebagai optimalisasi beberapa fungsi tujuan berstandarisasi skalar yang membutuhkan maksimalisasi atau minimalisasi berkenaan dengan parameternya. Adam merupakan algoritma optimasi yang dikembangkan dengan memanfaatkan kelebihan dari algoritma Adaptive Gradient (AdaGrad) yang bekerja dengan baik dengan gradien jarang dan Root Mean Square Propagation (RMSProp) yang bekerja dengan baik secara on-line dan non Pengaturan-Stasioner. Adam tidak hanya mengadaptasi tingkat pembelajaran parameter berdasarkan rata-rata pertama (mean) seperti dalam RMSProp, namun Adam juga menggunakan rata-rata kedua dari gradien (*varians uncentered*). Algoritma menghitung rata-rata pergerakan eksponensial dari gradien dan gradien kuadratnya, dan parameter  $\beta_1$  dan  $\beta_2$  mengontrol tingkat peluruhan rata-rata pergerakan (Aldi, Jondri, & Aditsania, 2018).

Beberapa keuntungan Adam adalah bahwa besarnya pembaruan parameter tidak sama dengan penskalaan gradien, ukurannya kira-kira dibatasi oleh *hyperparameter stepsize*, tidak memerlukan alat tulis stasioner. *Adaptive Moment Estimation* (Adam) adalah metode yang menghitung *adaptive learning rate* untuk setiap parameter. Nilai parameter yang direkomendasikan adalah  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , dan  $\epsilon = 10^{-8}$  dengan  $\beta_1 = \beta_2 =$  tingkat penurunan eksponensial dan  $\epsilon =$  nilai epsilon untuk update parameter (Kingma & Ba, 2015)

## BAB 4 METODOLOGI PENELITIAN

### 4.1 Populasi Penelitian

Populasi dalam penelitian ini adalah seluruh data harga Bitcoin dan informasi Blockchain. Sedangkan yang menjadi sampel penelitian adalah data *Price Bitcoin, Miners Revenue, Difficulty, Hash Rate, Median Confirmation Time, Average Block Size, Total Number of a Unique Bitcoin, Transaction per Block, Confirmed Transaction*, dan *Cosh % per Transaction* masing-masing variabel tersebut dari periode 04 Agustus 2018 s.d 21 Januari 2020.

### 4.2 Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data tersebut diperoleh dari beberapa website yaitu dari [www.Blockchain.com](http://www.Blockchain.com) untuk memperoleh data informasi Blockchain bitcoin, kemudian dari website [www.investing.com](http://www.investing.com) untuk memperoleh data bitcoin. Penelitian ini memiliki jumlah data 537, untuk masing-masing variabel, sehingga total data dari penelitian ini ada sebanyak 5.370.

### 4.3 Variabel Penelitian dan Definisi Operasional

Variabel yang digunakan untuk penelitian terdiri dari 26 variabel. Definisi operasional variabel penelitian merupakan penjelasan dari masing-masing variabel yang digunakan dalam penelitian terhadap indikator-indikator yang membentuknya. Definisi operasional penelitian ini dapat dilihat pada tabel 4.1:

**Tabel 4.1.** Definisi Operasional Variabel Penelitian

No.	Variabel	Kategori Data	Definisi Operasional
1.	Price (Y) dan ( $X_1$ )	Harga Bitcoin	Harga Bitcoin yang dikonversi dalam <i>United States Dollar</i> (USD)
2.	<i>Miners Revenue</i> ( $X_2$ )	Informasi Blockchain	Jumlah total hadiah blok coinbase dan ongkos-ongkos transaksi yang dibayarkan kepada para penambang.

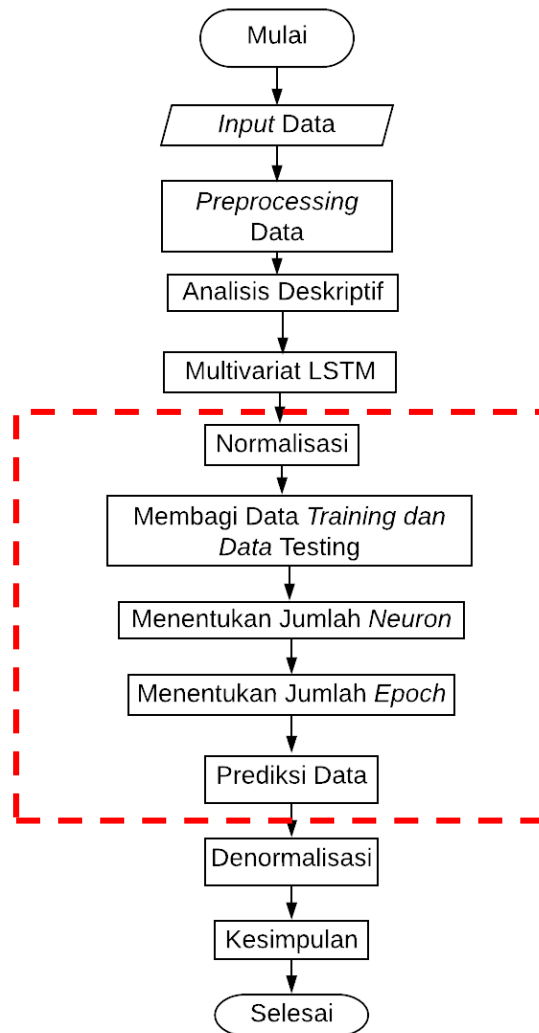
No.	Variabel	Kategori Data	Definisi Operasional
3.	<i>Difficulty</i> ( $X_3$ )	Informasi Blockchain	Ukuran relatif dari betapa sulitnya untuk menemukan blok baru.
4.	<i>Hash Rate</i> ( $X_4$ )	Informasi Blockchain	Perkiraan jumlah Tera (trilyun) hash per detik semua penambang (pelaku pasar untuk menyelesaikan masalah hash untuk membuat blok) berkinerja
5.	<i>Median Confirmation Time</i> ( $X_5$ )	Informasi Blockchain	Waktu rata-rata setiap transaksi untuk diterima ke dalam blok yang ditambang dan dicatat ke buku besar
6.	<i>Average Block Size</i> ( $X_6$ )	Informasi Blockchain	Ukuran blok rata-rata 24 jam dalam MB ( <i>Megabytes</i> )
7.	<i>Total Number of a Unique Bitcoin</i> ( $X_7$ )	Informasi Blockchain	Jumlah total alamat unik yang digunakan pada Blockchain Bitcoin
8.	<i>Transaction per Block</i> ( $X_8$ )	Informasi Blockchain	Jumlah rata-rata transaksi per blok
9.	<i>Confirmed Transaction</i> ( $X_9$ )	Informasi Blockchain	Jumlah konfirmasi validitas transaksi per hari.
10.	<i>Cosh % per Transaction</i> ( $X_{10}$ )	Informasi Blockchain	Penambang pendapatan sebagai persentase volume transaksi.

#### 4.4 Metode Analisis Data

Langkah-langkah yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. *Preprocessing* data dengan melihat ada atau tidaknya *missing value*
2. Melakukan analisis deskriptif pada data harian Bitcoin
3. Melakukan analisis Multivariat LSTM dengan langkah-langkah berikut ini:

- Melakukan normalisasi pada data penelitian dengan mengubah data aktual menjadi nilai dengan range interval [0,1] menggunakan *min-max scaling*.
  - Membagi data menjadi data *training* dan data *testing*
  - Menentukan jumlah *neuron* pada *hidden layer*
  - Menentukan jumlah *epoch*
  - Prediksi data
4. Membandingkan nilai MSE, RMSE, dan MAPE dari setiap percobaan.
- Di bawah ini merupakan *flowchat* dari analisis di atas:

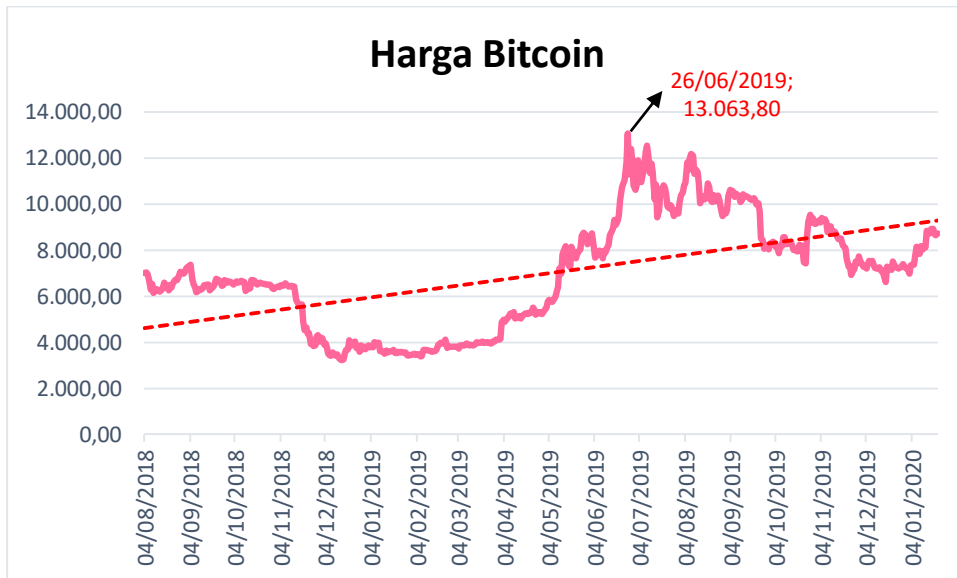


**Gambar 4.1.**Flowchart Analisis Data

## BAB 5 HASIL DAN PEMBAHASAN

### 5.1 Analisis Deskriptif

Analisis deskriptif pada data harian Bitcoin dilakukan untuk mengetahui gambaran umum Bitcoin pada tanggal 04 April 2018 s.d. 04 Januari 2020.



Gambar 5.1. Plot Harga Bitcoin

Berdasarkan **Gambar 5.1** dapat dilihat bahwa pergerakan harga Bitcoin cenderung memiliki *trend* naik atau mengalami kenaikan. Dapat dilihat pada tanggal 26 Juni 2019 Bitcoin mencapai harga tertinggi yaitu US\$ 13.063,80. Harga tersebut masih rendah jika dibandingkan pada tahun 2017 karena berhasil mencapai \$19.533 per coin. Harga Bitcoin terendah berkisar US\$ 3.300 yang terjadi pada awal desember 2018 terjadi penurunan sekitar 83% dari harga tertinggi (Farras, 2018).

### 5.2 Analisis *Multivariate Long-Short Term Memory*

Setelah melakukan analisis deskriptif, peneliti melakukan analisis terhadap data harian Bitcoin menggunakan *Multivariate Long-Short Term Memory*. Penelitian ini menggunakan 10 variabel dan membandingkan hasil dari masing-masing percobaan.

### 5.2.1 Normalisasi

Pada tahap ini data aktual akan diubah menjadi nilai dengan range interval [0,1] menggunakan *min-max scaling*. Rumus untuk menghitung *min-max scaling* terdapat pada persamaan (3.13). Di bawah ini merupakan contoh perhitungan *min-max scaler* menggunakan data pertama pada variabel *pricedan* variabel *Average Block Size*:

$$\begin{aligned}
 x'_{11} &= \frac{x - \min_x}{\max_x - \min_x} = \frac{7014,30 - 3228,70}{13063,80 - 3228,70} \\
 &= \frac{3785,60}{9835,10} \\
 &= 0,384907 \\
 x'_{61} &= \frac{x - \min_x}{\max_x - \min_x} = \frac{0,5740821 - 0,569716}{1,314410 - 0,569716} \\
 &= \frac{0,004361}{0,744694} \\
 &= 0,005864
 \end{aligned}$$

**Tabel 5.1.** Hasil Normalisasi  $x_1 - x_5$

No.	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
1.	0,384907	0,477111	0,087179	0,226014	0,065007
2.	0,386209	0,400315	0,087179	0,172682	0,098548
3.	0,377719	0,288066	0,087179	0,076019	0,484094
4.	0,355482	0,368886	0,087179	0,142683	0,221992
5.	0,310612	0,345915	0,087179	0,162683	0,271784
...	...	...	...	...	...
533.	0,577971	0,54799	1	0,955659	0,360996
534.	0,578296	0,424792	1	0,732138	0,349585
535.	0,556934	0,513674	1	0,881152	0,2787
536.	0,550396	0,474952	1	0,897709	0,254495
537.	0,559618	0,503086	1	0,905988	0,392808

**Tabel 5.2.** Hasil Normalisasi  $x_6$ -  $x_{10}$ 

No.	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
1.	0,005864	0,231191	0,05126	0,092534	0,565891
2.	0,011266	0,140486	0,043987	0,024816	0,649639
3.	0,646267	0,310222	0,379942	0,18624	0,238073
4.	0,599901	0,365906	0,308847	0,236441	0,321494
5.	0,326634	0,410834	0,283593	0,24432	0,250225
...	...	...	...	...	...
533.	0,535981	0,382175	0,589085	0,552667	0,159702
534.	0,51745	0,243444	0,667516	0,424868	0,342022
535.	0,409624	0,205668	0,507102	0,413474	0,464204
536.	0,566495	0,333175	0,603753	0,516745	0,18062
537.	0,577513	0,358851	0,601846	0,522131	0,290268

### 5.2.2 Pembentukan Data *Training* dan Data *Testing*

Data penelitian dibagi menjadi data *training* dan data *testing* dengan perbandingan 80% untuk data *training* atau sebanyak 420 dan 20% untuk data *testing* atau sebanyak 107. Penggunaan data *training* yang lebih banyak dilakukan agar mesin pembelajaran atau algoritma pembelajaran lebih terlatih dengan pola data dari data *training*. Data *training* tersebut dilatih menggunakan metode *Multivariate LSTM*. Dari pelatihan tersebut akan menghasilkan suatu model, dimana model tersebut akan diberikan kepada data *testing* untuk diuji performasinya terhadap data *testing*. Proses ini terus dilakukan hingga mendapatkan model dengan nilai akurasi yang paling bagus. Jika sudah didapatkan model yang terbaik, maka model tersebut yang akan digunakan untuk proses prediksi.

### 5.2.3 Penentuan Jumlah *Neuron* dan *Epoch*

Penentuan jumlah optimal *neuron* pada *hidden layer* dan *epoch* adalah tugas penting dalam *Multivariate LSTM*. Jaringan dengan beberapa *neuron* pada *hidden layer* mungkin tidak mampu menangkap hubungan kompleks antara variabel target dan *input*. Namun, jika terlalu banyak *neuron* ditugaskan di bagian

tersembunyi dari jaringan maka akan membuat kemampuan prediksi buruk dari data yang tidak terlihat karena *overparameterization*. Oleh karena itu, sejumlah neuron dalam lapisan tersembunyi harus dicoba, dan kinerja arsitektur harus dinilai setelah masing-masing dijalankan dengan sejumlah neuron dalam lapisan tersembunyi.

Pada penelitian ini jaringan yang dibentuk dengan 10 variabel *input* dan 1 *output layer* dengan jumlah *neuron* pada *hidden layer* yang akan digunakan untuk percobaan yaitu 10, 20, 30, 40, dan 50. Sedangkan untuk *epoch* akan digunakan 100, 500, dan 1000. Untuk mengetahui jumlah *neuron* dan *epoch* yang tepat dapat dilihat nilai *loss* yang terkecil, dimana nilai *loss* merupakan nilai MSE. Serta menggunakan optimasi Adam. Optimasi Adam digunakan karena menggunakan teknik *bias correction*. Tidak ada aturan dalam menentukan jumlah *neuron* dan *epoch* sehingga jumlah *neuron* dan *epoch* didapatkan melalui percobaan sampai mendapatkan yang paling optimal dalam memprediksi *time series*. Berikut merupakan tabel hasil pengujian dengan beberapa jumlah *neuron* dan *epoch* tersebut.

**Tabel 5.3.** Hasil Pengujian dengan Jumlah *Neuron* dan *Epoch* yang Berbeda

<b>Jumlah Neuron</b>	<b>Jumlah Epoch</b>	<b>MSE</b>
10	100	699199,312
	500	69395,375
	1000	87834,422
20	100	1992050,750
	500	68231,836
	1000	79453,750
30	100	646334,250
	500	70125,391
	1000	74488,625
40	100	701207,875
	500	69515,039
	1000	81914,812
50	100	999175,125
	500	70999,648
	1000	75650,141

Berdasarkan **Tabel 5.1** dapat diketahui bahwa jumlah *neuron* 20 dan jumlah *epoch* (iterasi) 500 menghasilkan nilai MSE yang terkecil jika dibandingkan dengan jumlah *neuron* dan *epoch* lainnya. Jumlah *epoch* yang terlalu sedikit



mengakibatkan jaringan yang terbentuk bersifat terlalu general (umum), artinya kemampuan jaringan dalam mengenali pola terlalu sedikit atau tidak ada sama sekali. Sedangkan jumlah *epoch* yang terlalu banyak dapat mengakibatkan jaringan mengalami kondisi *Overfit* atau jaringan terlalu bersifat terlalu spesifik terhadap data pelatihan.

Nilai bobot dan bias yang didapatkan dari 20 *neuron* dan 500 *epochs* dapat dilihat pada **Gambar 5.2** dan **Gambar 5.3**.

```

W_i[0]
array([ 0.04891053,  0.38845372, -0.02309303,  0.00698618,  0.20547616,
        -0.01673069,  0.41041806,  0.16973795, -0.00445261, -0.21795782,
         0.12717782,  0.10005609,  0.10773643,  0.23486105,  0.22297241,
        -0.00076477,  0.10844541, -0.28669253, -0.25354853,  0.07614724],
      dtype=float32)

W_f[0]
array([ 0.07831699, -0.16591972, -0.09731318, -0.2518856 , -0.10960013,
         0.23857257,  0.21642837,  0.07451889, -0.2394152 , -0.11519846,
        -0.0911006 , -0.18471584, -0.13887428,  0.04047704,  0.25461718,
         0.05831087,  0.01036477, -0.177281 , -0.22412126,  0.05594727],
      dtype=float32)

W_c[0]
array([-0.21956667,  0.26303318,  0.11383312, -0.06578061, -0.01992246,
        -0.03267276, -0.40426335,  0.86311466,  0.36938375, -0.01360906,
        -0.19584109, -0.41018346, -0.22954132, -0.17026925,  0.0010251 ,
        -0.03582549,  0.30731595,  0.10397169, -0.07712977, -0.12385806],
      dtype=float32)

W_o[0]
array([ 0.22949691, -0.07387494, -0.113127 ,  0.2152284 , -0.11221238,
        -0.1075 ,  0.22412285,  0.3701 ,  0.20737891,  0.16639605,
         0.46185178,  0.42671558,  0.04682501,  0.15358819,  0.26158622,
         0.10510428,  0.1360235 , -0.13238764, -0.28717265, -0.14416732],
      dtype=float32)

```

**Gambar 5.2.**Hasil Bobot dari Setiap *Neuron* untuk Variabel *Price*

```

b_i
array([ 0.05883237,  0.11260492,  0.08995706,  0.02576099,  0.04113714,
         0.01548249,  0.14412013,  0.7453386 ,  0.0661538 , -0.06955299,
         0.20050119,  0.10129908,  0.10646046,  0.18793572, -0.04723757,
        -0.04005893,  0.23936278,  0.0100099 , -0.04904279, -0.10293747],
      dtype=float32)

b_f
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
        1., 1., 1.], dtype=float32)

b_c
array([ 0.00705093,  0.01328159,  0.00671102,  0.02953802,  0.01408733,
         0.02128516, -0.02506395, -0.09149887, -0.02319848,  0.01200798,
        -0.01830644, -0.00736396, -0.01000169, -0.01533034,  0.03443636,
        -0.01539222,  0.00265756, -0.00829947, -0.01175768,  0.02442924],
      dtype=float32)

b_o
array([ 0.05654599,  0.12846348,  0.10173108,  0.01775204,  0.0502679 ,
         0.02539903,  0.13105091,  0.8466481 ,  0.05990457, -0.07735933,
         0.20251267,  0.08666528,  0.09384051,  0.17972258, -0.03188042,
        -0.03637034,  0.2360181 ,  0.01893345, -0.05558386, -0.0934607 ],
      dtype=float32)

```

**Gambar 5.3.**Bias untuk Setiap *Neuron*

Berdasarkan **Gambar 5.2**  $W_i[0]$ ,  $W_f[0]$ ,  $W_c[0]$ , dan  $W_o[0]$  berturut-turut merupakan bobot pada variabel pertama (*Price*) untuk *input gate*, *forget*

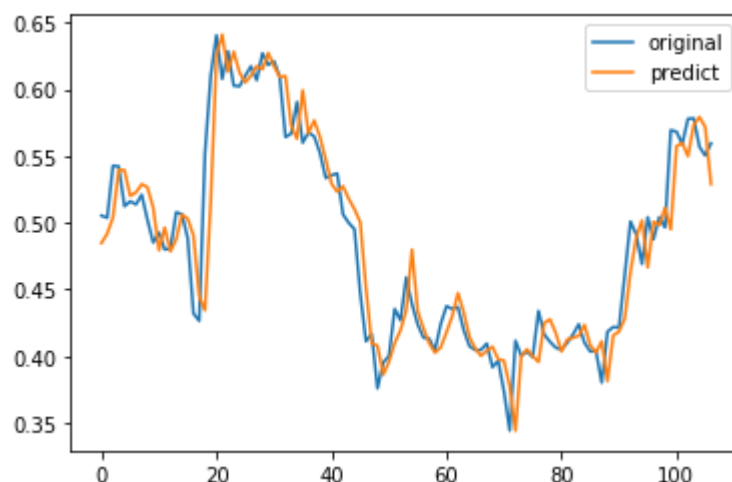
*gate*, *cell state*, dan *output gate*. *Inputgate* merupakan gerbang yang menentukan berapa banyak input jaringan waktu saat ini yang dicadangkan ke dalam status sel, *forget gate* merupakan gerbang yang menentukan informasi status sel mana yang dibuang dari model, sedangkan *output gate* merupakan gerbang yang mengontrol seberapa banyak keadaan sel saat ini dibuang. Banyaknya bobot sesuai dengan banyaknya *neuron* yang digunakan yaitu 20 dan masing-masing variabel memiliki bobotnya masing-masing. Sehingga ketika dijumlahkan terdapat 200 bobot untuk masing-masing *input gate*, *forget gate*, *cell state*, dan *output gate*. Sedangkan untuk **Gambar 5.2**  $b_i$ ,  $b_f$ ,  $b_c$ , dan  $b_o$  berturut-turut merupakan bias untuk *input gate*, *forget gate*, *cell state*, dan *output gate*. Banyaknya bias sesuai dengan banyaknya *neuron* yang digunakan yaitu 20 sehingga ketika dijumlahkan terdapat 80 bias secara keseluruhan.

### 5.3 Prediksi Data

Setelah mendapatkan model pada proses *training*, maka model tersebut akan diuji menggunakan data *testing* yang telah didapat pada pembentukan data *training* dan data *testing*.

Untuk melihat baik atau tidaknya hasil dari model yang dibentuk dapat dilihat dari plot data aktual dan data prediksi. Plot tersebut dapat dilihat pada

**Gambar 5.4:**



**Gambar 5.4.** Plot Perbandingan Data Prediksi dan Data Aktual

Pada **Gambar 5.4** dapat dilihat bahwa model yang dibentuk dapat menghasilkan output yang sesuai. Hal tersebut dapat dilihat dari pola data prediksi

yang mengikuti pola yang dibentuk oleh data aktual. Data aktual diwakili oleh warna biru dan data prediksi diwakili oleh warna merah. Pada **Tabel 5.2** akan ditampilkan prediksi harga. Hasil prediksi sebelumnya ditampilkan dalam nilai dengan range interval [0;1], sehingga nilai tersebut harus dikembalikan kedalam nilai sebenarnya dengan melakukan denormalisasi. Persamaan denormalisasi terdapat pada persamaan (3.14). Di bawah ini merupakan contoh perhitungan denormalisasi menggunakan data pertama dan ke-103:

$$\begin{aligned}
 dn_1 &= ((n_1)(x_{max} - x_{min})) + (x_{min}) \\
 &= ((0,48474732)(13063,80 - 3228,70)) + (3228,70) \\
 &= ((0,48474732)(9835,10)) + (3228,70) \\
 &= 7996,238
 \end{aligned}$$

$$\begin{aligned}
 dn_{103} &= ((n_{103})(x_{max} - x_{min})) + (x_{min}) \\
 &= ((0,5499253)(13063,80 - 3228,70)) + (3228,70) \\
 &= ((0,5499253)(9835,10)) + (3228,70) \\
 &= 8637,270
 \end{aligned}$$

**Tabel 5.4.** Hasil Pengujian

No.	Tanggal	Data Aktual	Hasil Prediksi dalam <i>Min-Max Scaler</i>	Hasil Prediksi
1.	05 Oktober 2019	8127,30	0,48474732	7996,238
2.	06 Oktober 2019	7881,90	0,49185348	8066,128
3.	07 Oktober 2019	8198,60	0,50396669	8185,265
4.	08 Oktober 2019	8182,90	0,53967196	8536,427
5.	09 Oktober 2019	8566,70	0,53941983	8533,497
...	...	...	...	...
103.	17 Januari 2020	8913,10	0,5499253	8637,270
104.	18 Januari 2020	8916,30	0,5734591	8868,727
105.	19 Januari 2020	8706,20	0,57933766	8926,543
106.	20 Januari 2020	8641,90	0,5717758	8852,172
107.	21 Januari 2020	8732,60	0,52899164	8431,385

Hasil dari setiap model yang dilatih diukur oleh *Mean Absolute Percentage Error* (MAPE). Nilai MAPE yang dihasilkan dari prediksi dapat dilihat pada **Gambar 5.5**.

```
mape=np.mean(np.abs((inv_y-inv_yhat)/inv_y))*100
mape
8.92966577970615e-166
```

**Gambar 5.5.**Nilai MAPE untuk Model Terbaik

Dari nilai MAPE tersebut dapat dilihat bahwa hasil prediksi dengan menggunakan 20 *neuron* dan 500 *epoch* memiliki nilai akurasi sebesar 91,07 %.

## BAB 6 PENUTUP

### 6.1 Kesimpulan

Berdasarkan rumusan masalah pada bab sebelumnya, diperoleh beberapa kesimpulan sebagaimana berikut ini:

1. Pergerakan harga Bitcoin cenderung memiliki *trend* naik atau mengalami kenaikan. Harga tertinggi terdapat pada tanggal 26 Juni 2019 yaitu mencapai US\$ 13.063,80. Harga tersebut masih rendah jika dibandingkan pada tahun 2017 karena berhasil mencapai \$19.533 per coin. Harga Bitcoin terendah berkisar US\$ 3.300 yang terjadi pada awal desember 2018 terjadi penurunan sekitar 83% dari harga tertinggi.
2. Prediksi menggunakan *Multivariate LSTM* langkah pertama yang harus dilakukan adalah normalisasi menggunakan *min-max scaling*. Langkah kedua yaitu membagi data menjadi data *training* dan data *testing*, dalam penelitian ini menggunakan perbandingan 80% data *training* dan 20% data *testing*. Langkah ketiga yaitu menentukan jumlah *neuron* dan *epoch*, pada penelitian ini menggunakan jumlah *neuron* 10, 20, 30, 40, dan 50, sedangkan jumlah *epoch* yang digunakan adalah 100, 500, dan 1000, berdasarkan nilai MSE didapatkan bahwa model dengan jumlah neuron 20 dan *epoch* 500 memiliki nilai MSE terkecil. Kemudian dilakukan prediksi dan diperoleh bahwa nilai prediksi tidak jauh berbeda dari data aktual, dengan tingkat akurasi sebesar 91,07% berdasarkan nilai MAPE.

### 6.2 Saran

Berdasarkan hasil penelitian saran yang dapat diberikan penulis:

1. Penelitian selanjutnya dapat menggunakan optimasi lain (misalnya AdaMax, RMSprop, dll.) sehingga diperoleh prediksi yang lebih akurat.
2. Dapat menambahkan variabel makro ekonomi untuk melihat hubungan antara *cryptocurrency* dengan kondisi ekonomi.

## DAFTAR PUSTAKA

- Aldi, M. W., Jondri, & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *e-Proceeding of Engineering*, 3551-3552. [https://repository.telkomuniversity.ac.id/pustaka/files/144393/jurnal\\_eproc/analisis-dan-implementasi-long-short-term-memory-neural-network-untuk-prediksi-harga-bitcoin.pdf](https://repository.telkomuniversity.ac.id/pustaka/files/144393/jurnal_eproc/analisis-dan-implementasi-long-short-term-memory-neural-network-untuk-prediksi-harga-bitcoin.pdf).
- Al-Yahyaee, K. H., & dkk. (2019). Can Uncertainty Indices Predict Bitcoin Prices? A Revisited Analysis Using Partial and Multivariate Wavelet Approaches. *North American Journal of Economics and Finance*, 48. <https://www.sciencedirect.com/science/article/pii/S1062940818306703>.
- Andini, T. D., & Auristandi, P. (2016). Peramalan Jumlah Stok Alat Tulis Kantor Di UD ACHMAD. *JITIKA*, 2-3.
- Ardiyanto, R. (2019, Desember 24). *Statistik Deskriptif – Pengertian, Materi, Jenis – Jenis, Contoh Soal*. Retrieved from <https://rumus.co.id/statistik-deskriptif/>
- Bakar, N. A., & Rosbi, S. (2017). Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction. *International Journal of Advanced Engineering Research and Science*, 130.
- Bhiantara, I. B. (2018). Teknologi Blockchain Cryptocurrency di Era Revolusi Digital. *SENAPATI*, 175.
- bitcoin.org. (2020, Maret 2). *Bagaimana cara kerja Bitcoin?* Retrieved from <https://bitcoin.org/id/cara-kerja>
- Blau, B. M. (2018). Price Dynamics and Speculative Trading in Bitcoin. *Research in International Business and Finance*, 16. <https://ideas.repec.org/a/eee/riibaf/v43y2018icp15-21.html>.
- Blockchain. (2020, Februari 10). Retrieved from Bitcoin untuk Pemula: <https://www.blockchain.com/id/learning-portal/bitcoin-faq>

- Brownlee, J. (2017, Agustus 14). *Multivariate Time Series Forecasting with LSTMs in Keras*. Retrieved from <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>
- Budiman, H. (2016). Analisis Dan Perbandingan Akurasi Model Prediksi Rentet Waktu Support Vector Machines Dengan Support Vector Machines Particle Swarm Optimization Untuk Arus Lalu Lintas Jangka Pendek. *SYSTEMIC*, 21-22. <http://jurnalsaintek.uinsby.ac.id/index.php/SYSTEMIC/article/view/103>.
- Cachin, C., & Vukolić, M. (2017). Blockchain Consensus Protocols in The Wild. <http://drops.dagstuhl.de/opus/volltexte/2017/8016/pdf/LIPIcs-DISC-2017-1.pdf>.
- Che, Z., & dkk. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 1. <https://www.nature.com/articles/s41598-018-24271-9.pdf>.
- Darmawan, G., & dkk. (2016). PERAMALAN PADA DATA IRREGULAR SINUSOIDAL DENGAN MENGGUNAKAN MODEL HOLT-WINTERS . *Prosiding Seminar Nasional Matematika dan Terapannya*, 183.
- Dharma, I. S., Putera, I. A., & Ardana, P. D. (2011). Artificial Neural Networks untuk Pemodelan Curah Hujan Limpasan pada Daerah Aliran Sungai (DAS) di Pulau Bali. *Jurnal Bumi Lestari*, 10. <https://ojs.unud.ac.id/index.php/blje/article/view/80/62>.
- Dyhrberg, A. H. (2016). Bitcoin, Gold and The dollar –A GARCH Volatility Analysis. *Finance Research Letters*, 86. <https://www.sciencedirect.com/science/article/pii/S1544612315001038>.
- Eka, R. (2018, Maret 15 ). *Mengenal Cryptocurrency dan Mekanisme Transaksinya*. Retrieved from Daily Social: <https://dailysocial.id/post/mengenal-cryptocurrency-dan-mekanisme-transaksinya>

- Farras, B. (2018). *Harga Bitcoin Anjlok, Pengangguran Bertambah*. Jakarta: CNBC Indonesia.
- Habibi, M. Y., & Riksakomara, E. (2013). Peramalan Harga Garam Konsumsi Menggunakan Artificial Neural Network Feedforward-Backpropagation (Studi Kasus : PT. Garam Mas, Rembang, Jawa Tengah). *Jurnal Teknik ITS*, 307. <http://ejurnal.its.ac.id/index.php/teknik/article/download/23200/4387>.
- Harwick, C. (2016). Cryptocurrency and the. *The Independent Review*, 570. [http://www.independent.org/pdf/tir/tir\\_20\\_04\\_05\\_harwick.pdf](http://www.independent.org/pdf/tir/tir_20_04_05_harwick.pdf).
- Hassani, H., Huang, X., & Silva, E. (2018). Big-Crypto: Big Data, Blockchain and Cryptocurrency. *Big Data and Cognitive Computing*, 2-3. <https://www.mdpi.com/2504-2289/2/4/34>.
- Hileman, G., & Rauchs, M. (2017). *Global Cryptocurrency Benchmarking Study*. Cambridge: University of Cambridge Judge Business School.
- Jang, H., & Lee, J. (2018). An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. *IEEE*, vol. 6, ISSN: 2169-3536, 5427-5437. <https://ieeexplore.ieee.org/document/8125674>.
- Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *ICLR*, 2.
- Kristanto, Y. D. (2013, Agustus 23). *Diagram Batang Daun*. Retrieved from <https://yos3prens.wordpress.com/2013/08/23/diagram-batang-daun/>
- Kristanto, Y. D. (2013, Agustus 18). *Histogram, Poligon, dan Ogive*. Retrieved from <https://yos3prens.wordpress.com/2013/08/18/histogram-poligon-dan-ogive/>
- Lipton, Z. C., & dkk. (2016). Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv:1511.03677v7*, 1-18. <https://arxiv.org/pdf/1511.03677.pdf>.
- Livebitnews. (2017, September 1). *How Bitcoin Transactions Work*. Retrieved from <https://livebitnews.com/featured/how-bitcoin-transactions-work/>



- Ma, X. (2015). Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data. *Transportation Research Part C*, 191. <https://www.sciencedirect.com/science/article/pii/S0968090X15000935>.
- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. 8. <https://bitcoin.org/bitcoin.pdf>.
- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. ., <http://bitcoin.org/bitcoin.pdf>.
- Nasution, L. M. (2017). Statistik Deskriptif. *Jurnal Hikmah*, 1. <http://jurnalhikmah.staisumatera-medan.ac.id/index.php/hikmah/article/view/8>.
- Nawangwulan, S., & Angesti, D. (2016). Analisis Time Series pada Metode Winter Jumlah Penderita Gastroenteritis Rawat Inap berdasarkan Data Rekam Medis di RSUD Soetomo Surabaya. *Jurnal Manajemen Kesehatan STIKES Yayasan RS. Dr. Soetomo*, 20-21. <https://media.neliti.com/media/publications/258464-analisis-time-series-metode-winter-jumla-a024036e.pdf>.
- Olah, C. (2015, Agustus 27). *Understanding LSTM Networks*. Retrieved from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Patel, V. (2019). A Framework for Secure and Decentralized Sharing of Medical Imaging Data Via Blockchain Consensus. *Health Informatics Journal*, 1400.
- Purwoto, A. (2007). *Panduan Laboratorium Statitik Inferensial*. Jakarta: Gramedia Widiasana Indonesia.
- Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting Stock Prices with Long-Short Term Memory Neural Network Based on Attention Mechanism. *PLOS ONE* 15(1): e0227222, 2-3. <https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0227222>.
- R, V. K. (2017, Agustus 24). *What is the sigmoid function, and what is its use in machine learning's neural networks? How about the sigmoid derivative*

*function?* Retrieved from <https://www.quora.com/What-is-the-sigmoid-function-and-what-is-its-use-in-machine-learning-neural-networks-How-about-the-sigmoid-derivative-function>

- Rizal, A. A., & Soraya, S. (2018). Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory. *MATRIK*, 119-120.
- Saputra, A. H. (2013, Maret 28). *Pengertian Analisis Runtun Waktu dan Peramalan*. Retrieved from <https://statistikawanku.wordpress.com/>
- Saritas, M. M., & Yasar, A. (2019). Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification. *International Journal of Intelligent Systems and Applications in Engineering*, 89. <https://ijisae.org/IJISAE/article/view/934/585>.
- Tian, C., & dkk. (2018). A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. *Energies*, 3-4. <https://www.mdpi.com/1996-1073/11/12/3493/pdf>.
- Tiyas. (2019, 08 24). *Statistika Deskriptif*. Retrieved from [yuksinau: https://www.yuksinau.id/statistika-deskriptif/](https://www.yuksinau.id/statistika-deskriptif/)
- Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. United Kingdom: John Wiley & Sons.
- Wikipedia. (2020, Februari 12). *Histogram*. Retrieved from <https://en.wikipedia.org/wiki/Histogram>
- Wikipedia. (2020, Februari 24). *Pie Chart*. Retrieved from [https://en.wikipedia.org/wiki/Pie\\_chart](https://en.wikipedia.org/wiki/Pie_chart)
- Yin, C. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE*, 21955-21956. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8066291>.
- Zheng, J., & dkk. (2017). Electric Load Forecasting in Smart Grid Using Long-Short-Term-Memory based. 1-6.
- Zheng, J., Xu, C., Zhang, Z., & Li, X. (2017). Electric Load Forecasting in Smart Grid Using Long-Short-Term-Memory based Recurrent Neural Network. *IEEE*, 1-6.

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7926112&isnumber=7926061>.

## LAMPIRAN

### Lampiran 1 Data Harga dan Informasi Blockchain

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
04/08/2018	7,014.30	16086605	5.95E+12	52051672	4.15	0.574082	416161	1088.96	191657	3.360264155
05/08/2018	7,027.10	14248433	5.95E+12	47319702	4.554167	0.578105	374557	1076.131	172181	3.815317256
06/08/2018	6,943.60	11561668	5.95E+12	38743006	9.2	1.050987	452410	1668.756	218607	1.579022046
07/08/2018	6,724.90	13496173	5.95E+12	44657969	6.041667	1.016458	477951	1543.344	233045	2.032299925
08/08/2018	6,283.60	12946324	5.95E+12	46432457	6.641667	0.812958	498558	1498.796	235311	1.645052472
09/08/2018	6,538.80	12039937	5.95E+12	44066472	6.416667	0.804356	467197	1492.121	222326	1.281178171
10/08/2018	6,152.30	14364533	5.95E+12	52051672	5.291667	0.792378	510030	1284.932	226148	1.889428067
11/08/2018	6,231.60	13368973	6.2E+12	52376370	4.541667	0.611511	389256	1131.988	192438	2.646248172
12/08/2018	6,313.00	12290113	6.39E+12	49230273	4.575	0.584547	361544	1064.8	165044	3.682653893
13/08/2018	6,255.30	12178415	6.39E+12	47642200	7.833333	0.906497	459526	1481.527	222229	1.442942022
14/08/2018	6,190.20	10463533	6.39E+12	43195594	7.633333	1.011793	514445	1820.75	247622	0.988311789
15/08/2018	6,256.90	13104014	6.39E+12	51135961	5.666667	0.791067	539383	1528.36	246066	1.469288958
16/08/2018	6,306.70	12659504	6.39E+12	50183117	6.729167	0.761947	479347	1470.475	232335	1.842721284
17/08/2018	6,581.70	12438186	6.39E+12	48277429	7.266667	0.833029	512305	1538.118	233794	1.588500343
18/08/2018	6,379.10	13176949	6.39E+12	50818346	4.779167	0.59006	399289	1193.344	190935	3.508085192
19/08/2018	6,481.50	11859088	6.39E+12	46054126	6.475	0.619528	402087	1154.483	167400	4.351836573
20/08/2018	6,268.10	12603282	6.39E+12	47642200	7.208333	0.818182	459856	1433.22	214983	1.947851278
21/08/2018	6,482.20	11982909	6.39E+12	47006970	7.070833	0.779787	449916	1471.466	217777	1.372616577
22/08/2018	6,359.60	13754538	6.39E+12	50818346	6.816667	0.774898	484463	1439.581	230333	1.344606505

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
23/08/2018	6,521.20	12091835	6.39E+12	47324585	6.075	0.768391	444346	1458.738	217352	1.367162663
24/08/2018	6,680.40	12238689	6.51E+12	47211617	6.879167	0.820711	487991	1505.808	219848	1.852838346
25/08/2018	6,734.80	11462792	6.73E+12	45145646	6.9	0.763552	415348	1454.963	196420	2.23443539
26/08/2018	6,712.70	14782109	6.73E+12	57853309	4.366667	0.569716	368938	998.5376	172747	3.830628312
27/08/2018	6,897.60	16066332	6.73E+12	61866256	4.9	0.70155	464084	1208.951	223656	2.038224364
28/08/2018	7,074.60	13671393	6.73E+12	51165065	8.516667	0.879881	488605	1555.81	238039	1.398134111
29/08/2018	7,029.10	12427794	6.73E+12	46483295	9.316667	0.907886	509197	1708.691	237508	1.42657418
30/08/2018	6,993.70	13524160	6.73E+12	51499478	6.9625	0.817751	463657	1502.539	231391	1.69373398
31/08/2018	7,033.80	13715514	6.73E+12	51833890	6.716667	0.809181	511307	1529.523	237076	1.581323438
01/09/2018	7,189.60	13547551	6.73E+12	50496241	6.15	0.769296	437404	1412.788	213331	2.552713877
02/09/2018	7,293.90	13150418	6.73E+12	48155356	5.875	0.648997	387889	1299.618	187145	2.945127436
03/09/2018	7,264.20	13221830	6.73E+12	48155356	9.108333	0.846752	455957	1547.896	222897	1.824192662
04/09/2018	7,377.50	11701743	6.73E+12	42135936	9.383333	0.957671	474899	1843.373	232265	1.381821978
05/09/2018	6,706.80	11985554	6.73E+12	44142409	11.90833	1.008444	497298	1867.25	246477	1.26817328
06/09/2018	6,514.00	12250782	6.73E+12	50161829	8.0375	0.888037	501001	1681.74	252261	1.1313016
07/09/2018	6,390.90	12582725	6.97E+12	52673943	7.408333	0.889911	539383	1518.355	230790	1.687668882
08/09/2018	6,184.30	12048687	7.02E+12	52338943	5.933333	0.71054	414919	1362.853	204428	3.305085876
09/09/2018	6,238.50	13080457	7.02E+12	56874985	5.05	0.616029	387389	1158.969	188912	2.793516392
10/09/2018	6,321.00	10889257	7.02E+12	47105049	8.133333	0.899062	451965	1650.859	222866	1.536186547
11/09/2018	6,293.90	11243963	7.02E+12	49198606	8.883333	0.941821	474061	1684.901	237571	1.516868646
12/09/2018	6,338.00	13463775	7.02E+12	58619616	4.9625	0.709051	461586	1388.542	233275	2.014361565
13/09/2018	6,482.90	12034014	7.02E+12	51641090	7.333333	0.819059	466217	1582.493	234209	0.963608234
14/09/2018	6,483.70	11020991	7.02E+12	46058270	8.7	1.057485	471373	1691.288	223250	1.490270084

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
15/09/2018	6,519.00	11500219	7.02E+12	48849680	8.004167	0.789859	427958	1507.65	211071	3.070390911
16/09/2018	6,500.20	11508924	7.02E+12	48500754	5.008333	0.627328	370212	1291.245	179483	4.050375827
17/09/2018	6,251.90	10885341	7.02E+12	47105049	7.916667	0.907648	481968	1778.904	240152	1.900205163
18/09/2018	6,342.00	11443546	7.02E+12	49547533	8.091667	0.936314	479309	1720.134	244259	1.876311088
19/09/2018	6,390.40	13390617	7.02E+12	54432501	5.3	0.804437	480228	1609.212	251037	1.893534463
20/09/2018	6,489.20	12451119	7.03E+12	53827511	6.325	0.815061	493141	1647.234	253674	1.824567664
21/09/2018	6,766.70	12165755	7.15E+12	51200544	8.725	1.012302	563131	1896.188	273051	1.235184545
22/09/2018	6,729.60	13446563	7.15E+12	55111697	5.625	0.732039	445042	1455.077	225537	2.898030271
...	...	...	...	...	...	...	...	...	...	...
01/12/2019	7,396.40	12925726	1.3E+13	89641525	6.066667	0.870623	468193	1975.691	274621	3.151231135
02/12/2019	7,298.20	13446042	1.3E+13	93510943	7.6	1.019306	552276	2170.552	314730	1.487299692
03/12/2019	7,291.80	14637722	1.3E+13	1.02E+08	6.1	0.961583	565368	2129.823	336512	1.329286501
04/12/2019	7,204.40	15591131	1.3E+13	1.08E+08	7.675	0.934352	569753	2019.665	337284	1.063153865
05/12/2019	7,399.20	13451289	1.3E+13	93386357	7.716667	1.018665	535061	2148.055	311468	1.62104085
06/12/2019	7,530.80	13603962	1.29E+13	92816137	8.266667	1.059104	534431	2175.076	315386	1.101500749
07/12/2019	7,510.90	15286888	1.29E+13	99217250	6.616667	0.86581	465570	1840.542	285284	1.81406519
08/12/2019	7,524.40	13969510	1.29E+13	94736471	5.058333	0.835228	434939	1884.142	278853	4.537030423
09/12/2019	7,355.80	13970882	1.29E+13	94736471	7.041667	1.089034	538820	2119.115	313629	1.419024244
10/12/2019	7,235.70	13646564	1.29E+13	94736471	8.85	1.106343	548706	2162.595	320064	1.13403933
11/12/2019	7,208.00	12062019	1.29E+13	84494691	8.833333	1.195491	521113	2206.424	291248	1.25614982
12/12/2019	7,208.00	12076200	1.29E+13	85134802	9.25	1.095938	531404	2344.323	311795	0.958747839
13/12/2019	7,255.20	14060284	1.29E+13	97296916	7.533333	0.990434	547195	2124.263	322888	1.543834671
14/12/2019	7,080.80	13824045	1.29E+13	96656805	6.216667	0.902877	413605	1943.682	293496	3.034936661

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
15/12/2019	7,128.50	11126048	1.29E+13	79373800	8.608333	1.12013	375238	2124.524	263441	3.356662636
16/12/2019	6,903.50	14132644	1.29E+13	1E+08	7.416667	1.038497	481698	2083.344	327085	1.605200667
17/12/2019	6,613.30	12121834	1.29E+13	90255692	9.4	1.31441	485661	2264.213	319254	0.713163943
18/12/2019	7,276.00	11695183	1.29E+13	85774913	9.816667	1.152717	490049	2345.142	314249	0.707152823
19/12/2019	7,165.50	14591652	1.29E+13	1.01E+08	7.533333	1.01337	471708	2035.382	319555	1.230739194
20/12/2019	7,196.40	14124384	1.29E+13	1E+08	7.4625	0.96181	476911	2132.417	332657	1.184714145
21/12/2019	7,156.20	15844785	1.29E+13	1.13E+08	4.558333	0.756154	414726	1767.716	311118	3.273568687
22/12/2019	7,495.80	15337833	1.29E+13	1.09E+08	4.75	0.74065	372537	1663.136	281070	3.749221778
23/12/2019	7,322.80	13758031	1.29E+13	92045964	8.25	1.037492	454864	2131.783	304845	1.201709417
24/12/2019	7,268.30	11525832	1.29E+13	80459759	9.816667	1.111658	430465	2323.488	290436	1.495865709
25/12/2019	7,224.80	11634910	1.29E+13	82390793	6.991667	0.889325	351294	1911.453	244666	1.347367792
26/12/2019	7,210.90	15752638	1.29E+13	1.07E+08	5.233333	0.837858	416540	1772.832	296063	2.00997963
27/12/2019	7,261.70	13514433	1.29E+13	93333320	7.766667	1.01724	470109	2222.097	322204	1.595361314
28/12/2019	7,321.50	13698701	1.29E+13	95908032	6.4	0.889533	424968	2001.926	298287	2.848944144
29/12/2019	7,397.50	14317659	1.29E+13	99770101	5.466667	0.787269	389596	1805.968	279925	3.75529837
30/12/2019	7,261.80	15446347	1.29E+13	1.07E+08	6.104167	0.944623	478277	1954.801	324497	1.615317474
31/12/2019	7,196.40	13152944	1.29E+13	93333320	7.475	0.982281	426495	2023.772	293447	1.539393374
01/01/2020	7,199.80	15907076	1.3E+13	1.13E+08	3.65	0.643639	334841	1446.753	251735	5.719048837
02/01/2020	6,967.00	12576131	1.38E+13	96717718	7.133333	1.012365	410885	2096.22	295567	1.817195049
03/01/2020	7,343.10	15705586	1.38E+13	1.16E+08	7.166667	0.91063	449208	1872.882	316517	0.912494011
04/01/2020	7,376.80	15650931	1.38E+13	1.15E+08	4.966667	0.794684	395429	1709.613	287215	2.511977716
05/01/2020	7,372.50	15404397	1.38E+13	1.12E+08	5.741667	0.790838	394533	1765.515	287779	4.335555415
06/01/2020	7,759.10	13051403	1.38E+13	92602071	10.6	1.127611	456930	2215.896	299146	1.356608999

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
07/01/2020	8,155.70	17646229	1.38E+13	1.21E+08	6.958333	0.948406	500250	1954.994	344079	1.571835216
08/01/2020	8,059.60	14890477	1.38E+13	97403660	8.891667	1.063491	483591	2239.57	318019	1.255746658
09/01/2020	7,842.40	15730240	1.38E+13	1.08E+08	7.35	1.04487	478535	2051.318	322057	1.010256771
10/01/2020	8,187.10	15541381	1.38E+13	1.02E+08	8.1	1.120476	507550	2219.691	330734	1.213589346
11/01/2020	8,024.10	16893865	1.38E+13	1.12E+08	5.866667	0.843841	435879	1896.135	309070	3.298074874
12/01/2020	8,187.60	15625351	1.38E+13	1.04E+08	6.116667	0.825021	383784	1875.073	283136	3.458136782
13/01/2020	8,111.40	14633330	1.38E+13	98089601	7.333333	0.981563	436615	2039.14	291597	1.659585759
14/01/2020	8,829.20	15748081	1.38E+13	99607271	10.35833	1.219665	535911	2392.779	346953	0.542858294
15/01/2020	8,818.30	16792710	1.48E+13	1.11E+08	8.083333	1.065172	503782	2203.033	332658	0.682998931
16/01/2020	8,726.90	17478169	1.48E+13	1.17E+08	6.716667	0.983502	479286	2049.981	325947	1.318362054
17/01/2020	8,913.10	17783148	1.48E+13	1.17E+08	7.716667	0.968858	485413	2037.686	323992	1.153184721
18/01/2020	8,916.30	14834325	1.48E+13	96958912	7.579167	0.955058	421781	2176.038	287237	2.14384001
19/01/2020	8,706.20	16961771	1.48E+13	1.1E+08	6.725	0.87476	404454	1893.067	283960	2.807736352
20/01/2020	8,641.90	16034934	1.48E+13	1.12E+08	6.433333	0.991581	462938	2063.559	313661	1.266845824
21/01/2020	8,732.60	16708344	1.48E+13	1.12E+08	8.1	0.999786	474715	2060.196	315210	1.862630627



**Lampiran 2 Data Hasil Normalitas**

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
04/08/2018	0.384907	0.477111	0.087179	0.226014	0.065007	0.005864	0.231191	0.05126	0.092534	0.565891
05/08/2018	0.386209	0.400315	0.087179	0.172682	0.098548	0.011266	0.140486	0.043987	0.024816	0.649639
06/08/2018	0.377719	0.288066	0.087179	0.076019	0.484094	0.646267	0.310222	0.379942	0.18624	0.238073
07/08/2018	0.355482	0.368886	0.087179	0.142683	0.221992	0.599901	0.365906	0.308847	0.236441	0.321494
08/08/2018	0.310612	0.345915	0.087179	0.162683	0.271784	0.326634	0.410834	0.283593	0.24432	0.250225
09/08/2018	0.33656	0.308047	0.087179	0.136017	0.253112	0.315083	0.34246	0.279809	0.199171	0.183258
10/08/2018	0.297262	0.405165	0.087179	0.226014	0.159751	0.298999	0.435845	0.162355	0.21246	0.2952
11/08/2018	0.305325	0.363572	0.112867	0.229673	0.09751	0.056124	0.172533	0.075652	0.09525	0.434484
12/08/2018	0.313601	0.318499	0.132668	0.194215	0.100277	0.019916	0.112115	0.037564	0	0.625224
13/08/2018	0.307735	0.313832	0.132668	0.176317	0.370678	0.452241	0.325736	0.273803	0.198834	0.213029
14/08/2018	0.301115	0.242187	0.132668	0.126202	0.35408	0.593636	0.445471	0.466106	0.287126	0.129359
15/08/2018	0.307897	0.352503	0.132668	0.215693	0.190871	0.297238	0.499841	0.300352	0.281716	0.217878
16/08/2018	0.312961	0.333932	0.132668	0.204954	0.279046	0.258135	0.36895	0.267538	0.233973	0.286604
17/08/2018	0.340922	0.324685	0.132668	0.183476	0.323651	0.353587	0.440805	0.305884	0.239046	0.239817
18/08/2018	0.320322	0.35555	0.132668	0.212114	0.11722	0.02732	0.194407	0.110434	0.090024	0.593096
19/08/2018	0.330734	0.300491	0.132668	0.158419	0.257953	0.06689	0.200507	0.088404	0.008192	0.748379
20/08/2018	0.309036	0.331583	0.132668	0.176317	0.318811	0.333648	0.326456	0.246418	0.173639	0.305952
21/08/2018	0.330805	0.305664	0.132668	0.169158	0.3074	0.28209	0.304784	0.2681	0.183354	0.200086

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
22/08/2018	0.318339	0.379681	0.132668	0.212114	0.286307	0.275526	0.380104	0.250024	0.227012	0.194931
23/08/2018	0.33477	0.310215	0.132668	0.172737	0.224758	0.266788	0.29264	0.260884	0.181876	0.199082
24/08/2018	0.350957	0.316351	0.144635	0.171464	0.291494	0.337044	0.387796	0.287568	0.190555	0.288466
25/08/2018	0.356488	0.283935	0.167612	0.14818	0.293223	0.26029	0.229419	0.258744	0.109095	0.358695
26/08/2018	0.354241	0.422611	0.167612	0.291401	0.082988	0	0.128235	0	0.026784	0.652457
27/08/2018	0.373041	0.476264	0.167612	0.336628	0.127248	0.177031	0.335673	0.119282	0.203796	0.322584
28/08/2018	0.391038	0.376207	0.167612	0.216021	0.427386	0.4165	0.389134	0.315914	0.253806	0.204782
29/08/2018	0.386412	0.324251	0.167612	0.163256	0.493776	0.454107	0.434029	0.40258	0.251959	0.210017
30/08/2018	0.382813	0.370056	0.167612	0.21979	0.298409	0.33307	0.334742	0.285715	0.23069	0.259184
31/08/2018	0.38689	0.37805	0.167612	0.223559	0.278008	0.321563	0.438629	0.301011	0.250457	0.238496
01/09/2018	0.402731	0.371033	0.167612	0.208483	0.230982	0.268003	0.277505	0.234835	0.167895	0.41727
02/09/2018	0.413336	0.354441	0.167612	0.1821	0.20816	0.106461	0.169553	0.17068	0.076846	0.48949
03/09/2018	0.410316	0.357425	0.167612	0.1821	0.476487	0.372014	0.317955	0.311427	0.201156	0.283194
04/09/2018	0.421836	0.293918	0.167612	0.114259	0.499308	0.520959	0.359252	0.478931	0.233729	0.20178
05/09/2018	0.353642	0.305775	0.167612	0.136873	0.708852	0.589139	0.408087	0.492466	0.283145	0.180865
06/09/2018	0.334038	0.316856	0.167612	0.204714	0.387621	0.427453	0.41616	0.387302	0.303256	0.155675
07/09/2018	0.321522	0.330724	0.19284	0.233027	0.335408	0.429969	0.499841	0.294681	0.228601	0.258068
08/09/2018	0.300516	0.308413	0.197806	0.229251	0.213001	0.189103	0.228484	0.206528	0.136939	0.555736
09/09/2018	0.306026	0.351518	0.197806	0.280375	0.139696	0.062191	0.168462	0.090948	0.08299	0.461587
10/09/2018	0.314415	0.259973	0.197806	0.170263	0.395574	0.442257	0.309251	0.369796	0.201049	0.230189

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
11/09/2018	0.311659	0.274792	0.197806	0.193858	0.457815	0.499676	0.357425	0.389094	0.252178	0.226634
12/09/2018	0.316143	0.367533	0.197806	0.300037	0.132434	0.187104	0.330227	0.22109	0.237241	0.318192
13/09/2018	0.330876	0.3078	0.197806	0.221386	0.329184	0.334827	0.340324	0.33104	0.240489	0.124813
14/09/2018	0.330957	0.265477	0.197806	0.158465	0.4426	0.654993	0.351565	0.392715	0.202384	0.221739
15/09/2018	0.334547	0.285498	0.197806	0.189926	0.384855	0.295616	0.256911	0.288612	0.160037	0.512543
16/09/2018	0.332635	0.285862	0.197806	0.185993	0.136238	0.077364	0.131013	0.165933	0.050205	0.692899
17/09/2018	0.307389	0.25981	0.197806	0.170263	0.377593	0.453787	0.374664	0.442384	0.261153	0.297183
18/09/2018	0.31655	0.283131	0.197806	0.197791	0.392116	0.49228	0.368867	0.409067	0.275433	0.292786
19/09/2018	0.321471	0.364476	0.197806	0.252847	0.160443	0.315192	0.370871	0.346186	0.299	0.295955
20/09/2018	0.331517	0.325226	0.199061	0.246028	0.245505	0.329458	0.399024	0.367741	0.308169	0.283263
21/09/2018	0.359732	0.313304	0.211605	0.216421	0.444675	0.594319	0.551616	0.508871	0.375543	0.174793
22/09/2018	0.35596	0.366814	0.211605	0.260502	0.187414	0.217973	0.294158	0.258809	0.210336	0.480822
...	...	...	...	...	...	...	...	...	...	...
01/12/2019	0.423758	0.345054	0.813532	0.649668	0.224066	0.404068	0.344632	0.553941	0.381002	0.527421
02/12/2019	0.413773	0.366792	0.813532	0.693278	0.351314	0.603725	0.52795	0.664406	0.520462	0.221192
03/12/2019	0.413122	0.416579	0.813532	0.787767	0.226833	0.526213	0.556494	0.641317	0.596199	0.192112
04/12/2019	0.404236	0.456411	0.813532	0.853182	0.357538	0.489645	0.566054	0.578869	0.598883	0.143133
05/12/2019	0.424042	0.367011	0.811745	0.691874	0.360996	0.602864	0.490418	0.651653	0.50912	0.245806
06/12/2019	0.437423	0.37339	0.803564	0.685447	0.406639	0.657167	0.489044	0.666971	0.522743	0.15019
07/12/2019	0.4354	0.4437	0.803564	0.757591	0.26971	0.397605	0.338913	0.477326	0.418078	0.28133
08/12/2019	0.436772	0.388662	0.803564	0.70709	0.140387	0.356539	0.272131	0.502042	0.395717	0.782462

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
09/12/2019	0.41963	0.388719	0.803564	0.70709	0.304979	0.697357	0.498613	0.635247	0.516634	0.208627
10/12/2019	0.407418	0.37517	0.803564	0.70709	0.455048	0.7206	0.520167	0.659895	0.539009	0.156179
11/12/2019	0.404602	0.30897	0.803564	0.591661	0.453665	0.840312	0.460008	0.684742	0.438815	0.178652
12/12/2019	0.404602	0.309562	0.803564	0.598875	0.488243	0.706629	0.482445	0.762916	0.510257	0.123918
13/12/2019	0.409401	0.392454	0.803564	0.735948	0.345781	0.564955	0.516873	0.638165	0.548828	0.231597
14/12/2019	0.391669	0.382584	0.803564	0.728733	0.236515	0.447381	0.225619	0.535795	0.446631	0.506018
15/12/2019	0.396519	0.269866	0.803564	0.533946	0.434993	0.739115	0.141971	0.638313	0.342129	0.565228
16/12/2019	0.373641	0.395477	0.803564	0.772019	0.3361	0.629495	0.374076	0.614968	0.563421	0.242891
17/12/2019	0.344135	0.311469	0.803564	0.65659	0.500692	1	0.382716	0.717502	0.536192	0.078721
18/12/2019	0.411516	0.293644	0.803564	0.60609	0.53527	0.782874	0.392283	0.76338	0.51879	0.077615
19/12/2019	0.400281	0.414654	0.805218	0.773426	0.345781	0.595754	0.352295	0.587779	0.537239	0.173975
20/12/2019	0.403422	0.395132	0.810984	0.771076	0.339903	0.526517	0.363639	0.642787	0.582795	0.165505
21/12/2019	0.399335	0.467008	0.810984	0.916167	0.098893	0.250355	0.228063	0.436041	0.507903	0.549936
22/12/2019	0.433864	0.445828	0.810984	0.865385	0.114799	0.229537	0.136082	0.376756	0.403426	0.637475
23/12/2019	0.416274	0.379826	0.810984	0.676767	0.405256	0.628145	0.315572	0.642428	0.486092	0.168633
24/12/2019	0.410733	0.286568	0.810984	0.546185	0.53527	0.727738	0.262377	0.751104	0.435991	0.222769
25/12/2019	0.40631	0.291126	0.810984	0.567949	0.30083	0.429182	0.089768	0.517525	0.276848	0.195439
26/12/2019	0.404897	0.463158	0.810984	0.850876	0.15491	0.36007	0.232018	0.438942	0.455557	0.317386
27/12/2019	0.410062	0.369649	0.810984	0.691276	0.365145	0.600951	0.348809	0.693626	0.54645	0.24108
28/12/2019	0.416142	0.377348	0.810984	0.720294	0.251729	0.429462	0.250392	0.568813	0.46329	0.471788
29/12/2019	0.42387	0.403207	0.810984	0.763822	0.174274	0.292138	0.173274	0.457726	0.399444	0.638593
30/12/2019	0.410072	0.450362	0.810984	0.843621	0.227178	0.503438	0.366617	0.542099	0.554422	0.244753
31/12/2019	0.403422	0.354547	0.810984	0.691276	0.340941	0.554006	0.253722	0.581198	0.446461	0.23078

Date	Price	Miners Revenue	Difficulty	Hash Rate	Median Confirmation time	Average Block Size	Total Number of a unique Bitcoin	Transaction per Block	Confirmed transaction	Cosh % per Transaction
01/01/2020	0.403768	0.46961	0.819574	0.909755	0.023513	0.099267	0.053897	0.25409	0.301427	1
02/01/2020	0.380098	0.330448	0.898905	0.72942	0.312586	0.594404	0.219689	0.622268	0.453832	0.281906
03/01/2020	0.418338	0.461192	0.898905	0.945884	0.315353	0.457791	0.303241	0.495659	0.526676	0.115406
04/01/2020	0.421765	0.458909	0.898905	0.938153	0.13278	0.302094	0.185991	0.403103	0.424792	0.409773
05/01/2020	0.421328	0.448609	0.898905	0.899499	0.197095	0.296931	0.184038	0.434794	0.426753	0.745383
06/01/2020	0.460636	0.350305	0.898905	0.683035	0.600277	0.74916	0.320076	0.690111	0.466276	0.19714
07/01/2020	0.500961	0.54227	0.898905	1	0.298064	0.508518	0.414523	0.542208	0.62251	0.23675
08/01/2020	0.49119	0.427138	0.898905	0.737151	0.458506	0.663058	0.378203	0.703532	0.531898	0.178578
09/01/2020	0.469106	0.462222	0.898905	0.853114	0.330567	0.638053	0.36718	0.596813	0.545938	0.133398
10/01/2020	0.504153	0.454332	0.898905	0.791267	0.392808	0.739579	0.430438	0.692263	0.576109	0.170819
11/01/2020	0.48758	0.510837	0.898905	0.899499	0.207469	0.368104	0.274181	0.508841	0.500782	0.554446
12/01/2020	0.504204	0.45784	0.898905	0.806728	0.228216	0.342832	0.160603	0.496901	0.410609	0.583904
13/01/2020	0.496457	0.416395	0.898905	0.744882	0.329184	0.553042	0.275785	0.589909	0.440028	0.2529
14/01/2020	0.56944	0.462968	0.900997	0.761986	0.580221	0.872774	0.492271	0.790385	0.632503	0.047378
15/01/2020	0.568332	0.506611	1	0.889431	0.391425	0.665315	0.422223	0.682819	0.582798	0.07317
16/01/2020	0.559039	0.535248	1	0.955659	0.278008	0.555646	0.368817	0.596055	0.559464	0.190101
17/01/2020	0.577971	0.54799	1	0.955659	0.360996	0.535981	0.382175	0.589085	0.552667	0.159702
18/01/2020	0.578296	0.424792	1	0.732138	0.349585	0.51745	0.243444	0.667516	0.424868	0.342022
19/01/2020	0.556934	0.513674	1	0.881152	0.2787	0.409624	0.205668	0.507102	0.413474	0.464204
20/01/2020	0.550396	0.474952	1	0.897709	0.254495	0.566495	0.333175	0.603753	0.516745	0.18062
21/01/2020	0.559618	0.503086	1	0.905988	0.392808	0.577513	0.358851	0.601846	0.522131	0.290268

### Lampiran 3 Nilai MSE

Jumlah Neuron	Jumlah Epoch	MSE
10	100	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 699199.312</p>
	500	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 69395.375</p>
	1000	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 87834.422</p>
20	100	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 1992050.750</p>
	500	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 68231.836</p>

<b>Jumlah Neuron</b>	<b>Jumlah Epoch</b>	<b>MSE</b>
	1000	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 79453.750</p>
30	100	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 646334.250</p>
	500	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 70125.391</p>
	1000	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 74488.625</p>
40	100	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 701207.875</p>

<b>Jumlah Neuron</b>	<b>Jumlah Epoch</b>	<b>MSE</b>
	500	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 69515.148</p>
	1000	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 81914.812</p>
50	100	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 999715.125</p>
	500	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 70999.648</p>
	1000	<pre># calculate MSE mse = mean_squared_error(inv_y, inv_yhat) print('Test MSE: %.3f' % mse)</pre> <p>Test MSE: 75650.141</p>



## Lampiran 4 Syntax Prediksi Harga Bitcoin Menggunakan *Multivariate* LSTM

```

jupyter Epoch 500 Neuron 20 Last Checkpoint: 11 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 C
+ - < > Run C Code

In [1]: import pandas as pd
import numpy as np
from datetime import datetime
from math import sqrt
from numpy import concatenate
from matplotlib import pyplot
from pandas import read_excel
from pandas import DataFrame
from pandas import concat
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from keras.models import Sequential, load_model
from keras.layers import Dense
from keras.layers import LSTM

Using TensorFlow backend.

In [2]: # Load dataset
dataset = read_excel('DATA.xlsx', header=0, index_col=0)
dataset

Out[2]:
      Price  Miners Revenue  Difficulty  Hash Rate  Median Confirmation time  Average Block Size  Total Number of a unique Bitcoin  Transaction per Block  Confirmed transaction  Cosh % per Transaction
Date
2018-08-04  7014.3  1.608861e+07  5.949437e+12  5.205167e+07  4.150000  0.574082  416161  1088.960227  191657  3.360264
2018-08-05  7027.1  1.424843e+07  5.949437e+12  4.731970e+07  4.554167  0.578105  374557  1076.131250  172181  3.815317
2018-08-06  6943.6  1.156167e+07  5.949437e+12  3.874301e+07  9.200000  1.050987  452410  1668.755725  218607  1.579022
2018-08-07  6724.9  1.349617e+07  5.949437e+12  4.465797e+07  6.041667  1.016458  477951  1543.344371  233045  2.032300
2018-08-08  6283.6  1.294632e+07  5.949437e+12  4.643246e+07  6.641667  0.812958  498558  1498.798178  235311  1.645052
...
...

In [3]: values = dataset.values
values

Out[3]: array([[7.01430000e+03, 1.60886054e+07, 5.94943737e+12, ...,
1.08896023e+03, 1.91657000e+05, 3.36026416e+00],
[7.02710000e+03, 1.42484333e+07, 5.94943737e+12, ...,
1.07613125e+03, 1.72181000e+05, 3.81531726e+00],
[6.94360000e+03, 1.15616683e+07, 5.94943737e+12, ...,
1.66875573e+03, 2.18607000e+05, 1.57902205e+00],
...,
[8.70620000e+03, 1.69617711e+07, 1.47763675e+13, ...,
1.89306667e+03, 2.83960000e+05, 2.80773635e+00],
[8.64190000e+03, 1.60349343e+07, 1.47763675e+13, ...,
2.06355921e+03, 3.13661000e+05, 1.26684582e+00],
[8.73260000e+03, 1.67083442e+07, 1.47763675e+13, ...,
2.06019608e+03, 3.15210000e+05, 1.86263063e+00]])

In [4]: # convert series to supervised Learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg

In [5]: # ensure all data is float
values = values.astype('float32')

In [6]: # normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)

```

```
In [7]: # frame as supervised Learning
reframed = series_to_supervised(scaled, 1, 1)
reframed
```

Out[7]:

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	var7(t-1)	var8(t-1)	var9(t-1)	var10(t-1)	var1(t)	var2(t)	var3(t)	var4(t)	var5
1	0.384907	0.477111	0.087179	0.226014	0.065007	0.005864	0.231191	0.051260	0.092534	0.565891	0.386209	0.400315	0.087179	0.172682	0.0985
2	0.386209	0.400315	0.087179	0.172682	0.098548	0.011266	0.140486	0.043987	0.024816	0.649639	0.377719	0.288066	0.087179	0.076019	0.4840
3	0.377719	0.288066	0.087179	0.076019	0.484094	0.646267	0.310222	0.379942	0.186240	0.238073	0.355482	0.368886	0.087179	0.142683	0.2219
4	0.355482	0.368886	0.087179	0.142683	0.221992	0.599901	0.365906	0.308847	0.236441	0.321494	0.310612	0.345915	0.087179	0.162683	0.2717
5	0.310612	0.345915	0.087179	0.162683	0.271784	0.326634	0.410834	0.283593	0.244320	0.250225	0.336560	0.308047	0.087179	0.136017	0.2531
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
531	0.559039	0.535248	1.000000	0.955659	0.278008	0.555646	0.368817	0.596055	0.559464	0.190101	0.577971	0.547990	1.000000	0.955659	0.3609
532	0.577971	0.547990	1.000000	0.955659	0.360996	0.535981	0.382175	0.589085	0.552667	0.159702	0.578296	0.424792	1.000000	0.732138	0.3495
533	0.578296	0.424792	1.000000	0.732138	0.349585	0.517450	0.243444	0.667516	0.424868	0.342022	0.556934	0.513674	1.000000	0.881152	0.2787
534	0.556934	0.513674	1.000000	0.881152	0.278700	0.409624	0.205668	0.507102	0.413474	0.464204	0.550396	0.474952	1.000000	0.897709	0.2544
535	0.550396	0.474952	1.000000	0.897709	0.254495	0.566495	0.333175	0.603753	0.516745	0.180620	0.559618	0.503086	1.000000	0.905988	0.3928

535 rows x 20 columns

```
In [8]: # drop columns we don't want to predict
reframed.drop(reframed.columns[[11,12,13,14,15,16,17,18,19]], axis=1, inplace=True)
print(reframed.head())
```

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)
1	0.384907	0.477111	0.087179	0.226014	0.065007	0.005864
2	0.386209	0.400315	0.087179	0.172682	0.098548	0.011266
3	0.377719	0.288066	0.087179	0.076019	0.484094	0.646267
4	0.355482	0.368886	0.087179	0.142683	0.221992	0.599901
5	0.310612	0.345915	0.087179	0.162683	0.271784	0.326634

	var7(t-1)	var8(t-1)	var9(t-1)	var10(t-1)	var1(t)
1	0.231191	0.051260	0.092534	0.565891	0.386209
2	0.140486	0.043987	0.024816	0.649639	0.377719
3	0.310222	0.379942	0.186240	0.238073	0.355482
4	0.365906	0.308847	0.236441	0.321494	0.310612
5	0.410834	0.283593	0.244320	0.250225	0.336560

```
In [9]: # print(reframed.tail())
```

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)
531	0.559039	0.535248	1.0	0.955659	0.278008	0.555646
532	0.577971	0.547990	1.0	0.955659	0.360996	0.535981
533	0.578296	0.424792	1.0	0.732138	0.349585	0.517450
534	0.556934	0.513674	1.0	0.881152	0.278700	0.409624
535	0.550396	0.474952	1.0	0.897709	0.254495	0.566495

	var7(t-1)	var8(t-1)	var9(t-1)	var10(t-1)	var1(t)
531	0.368817	0.596055	0.559464	0.190101	0.577971
532	0.382175	0.589085	0.552667	0.159702	0.578296
533	0.243444	0.667516	0.424868	0.342022	0.556934
534	0.205668	0.507102	0.413474	0.464204	0.550396
535	0.333175	0.603753	0.516745	0.180620	0.559618

```
In [10]: # split into train and test sets
values = reframed.values
n_train = 428
train = values[:n_train, :]
test = values[n_train:, :]
```

```
In [11]: # split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]
```

```
In [12]: # reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)

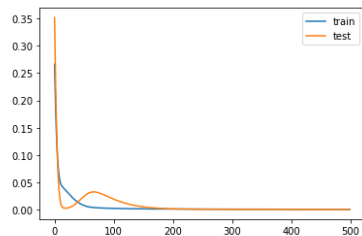
(428, 1, 10) (428,) (107, 1, 10) (107,)
```

```
In [13]: # design network
model = Sequential()
model.add(LSTM(20, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mse', optimizer='adam')
```

```
In [14]: # fit network
history = model.fit(train_X, train_y, epochs=500, batch_size=72, validation_data=(test_X, test_y), verbose=2, shuffle=False)
model.save('./savedModel')
```

```
Train on 428 samples, validate on 107 samples
Epoch 1/500
- 1s - loss: 0.2659 - val_loss: 0.3513
Epoch 2/500
- 0s - loss: 0.2173 - val_loss: 0.2793
Epoch 3/500
- 0s - loss: 0.1771 - val_loss: 0.2176
Epoch 4/500
- 0s - loss: 0.1439 - val_loss: 0.1661
Epoch 5/500
- 0s - loss: 0.1171 - val_loss: 0.1241
Epoch 6/500
- 0s - loss: 0.0960 - val_loss: 0.0905
Epoch 7/500
- 0s - loss: 0.0798 - val_loss: 0.0644
Epoch 8/500
- 0s - loss: 0.0677 - val_loss: 0.0447
Epoch 9/500
- 0s - loss: 0.0590 - val_loss: 0.0303
```

```
In [15]: # plot history
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend()
plt.show()
```



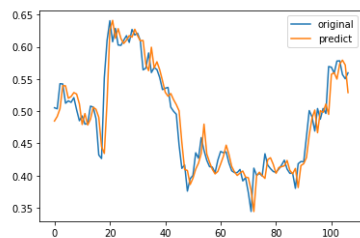
```
In [16]: # make a prediction
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], test_X.shape[2]))
```

```
In [17]: # yhat
```

```
Out[17]: array([[0.48474732],
 [0.49185348],
 [0.50396669 ],
 [0.53967196],
 [0.53941983],
 [0.52015824],
 [0.52274483],
 [0.5289072 ],
 [0.52666914],
 [0.5112585 ],
 [0.4793559 ],
 [0.49635276],
 [0.4785281 ],
 [0.48759127],
 [0.50615233],
 [0.502941 ],
 [0.49054852],
 [0.4447853 ],
 [0.43431386],
 [0.4785281 ]])
```

```
In [18]: # plot original and predicted values
plt.plot(test_y, label="original")
plt.plot(yhat, label="predict")
plt.legend(loc="best")
```

```
Out[18]: <matplotlib.legend.Legend at 0x15dd26f5ec8>
```



```
In [19]: # invert scaling for predict
inv_yhat = concatenate((yhat, test_X[:, 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
```

```

In [20]: print("Hasil Prediksi")
print(inv_yhat)

Hasil Prediksi
[7996.238 8066.128 8185.265 8536.427 8533.947 8344.429 8369.947
8430.555 8408.544 8256.899 7943.213 8110.3784 7934.992 8024.2085
8206.758 8175.175 8053.2935 7603.207 7500.2197 8299.683 9378.247
9534.587 9260.618 9408.25 9256.39 9180.813 9231.441 9301.566
9278.084 9398.772 9302.559 9221.707 9227.786 8861.618 8766.185
9123.8545 8806.033 8901.234 8779.857 8613.123 8435.745 8374.802
8416.584 8323.56 8246.313 8152.0815 7665.0136 7256.068 7240.0366
7025.2383 7120.641 7259.780 7351.622 7499.761 7948.352 7492.166
7366.395 7258.3457 7187.9277 7226.2954 7338.779 7457.0884 7627.147
7486.6284 7302.6055 7218.9106 7164.407 7201.6304 7231.822 7138.6016
7129.039 6938.194 6612.3687 7159.485 7213.5234 7162.7217 7119.207
7406.117 7434.499 7333.6074 7196.711 7284.8125 7298.1724 7311.3574
7392.821 7242.1465 7188.497 7271.426 6978.989 7315.2153 7343.2173
7438.7275 7783.7295 8037.9307 8163.182 7817.067 8153.616 8128.742
8255.285 8097.153 8710.462 8735.09 8637.2705 8868.727 8926.543
8852.172 8431.385 ]

In [21]: # invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]

In [22]: print("Data Aktual")
print(inv_y)

Data Aktual
[8198.601 8182.9 8566.7 8562.3 8267.8 8304.4 8281.501
8353.3 8167.2 8000.4 8073.3 7948.5 7957.3 8223.4
8208.2 8031.4 7477. 7422.7 8658.301 9230.6 9529.601
9207.2 9411.301 9157.9 9152.601 9230.1 9300.601 9198.301
9396.401 9310.801 9338.9 9223.501 8778.2 8804.501 9036.8
8736.9 8812.6 8783.101 8661.201 8476.3 8497.301 8509.8
8208.4 8145.6997 8099.3 7627.9 7272.5 7324.1 6924.7
7115.6 7162.8 7510.6 7426.7 7742.7 7546.6 7396.4
7298.2007 7291.7993 7204.4 7399.2007 7530.8 7510.9 7524.4004
7355.8 7235.7 7207.9995 7207.9995 7255.2 7080.8 7128.5
6903.5 6613.3 7276. 7165.5 7196.4 7156.2 7495.8
7322.8 7268.3 7224.8 7210.9 7261.7 7321.5005 7397.5
7261.8 7196.4 7199.8 6967. 7343.1 7376.8 7372.5
7759.1 8155.7007 8059.5996 7842.4 8187.0996 8024.1 8187.6006
8111.4 8829.2 8818.3 8726.9 8913.1 8916.301 8706.2
8641.9 8732.6 ]

In [23]: # calculate MSE
mse = mean_squared_error(inv_y, inv_yhat)
print('Test MSE: %.3f' % mse)

Test MSE: 68231.836

In [24]: # calculate RMSE
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
print('Test RMSE: %.3f' % rmse)

Test RMSE: 261.212

In [25]: mape=np.mean(np.abs((inv_y-inv_yhat)/inv_y))**100
mape

Out[25]: 8.92966577970615e-166

In [26]: units = int(int(model.layers[0].trainable_weights[0].shape[1])/4)

In [27]: W = model.layers[0].get_weights()[0]
b = model.layers[0].get_weights()[2]

In [28]: W_i = W[:, :units]
W_f = W[:, units: units * 2]
W_c = W[:, units * 2: units * 3]
W_o = W[:, units * 3:]

In [29]: W_i[0]

Out[29]: array([ 0.04891053, 0.38845372, -0.02309303, 0.00698618, 0.20547616,
-0.01673069, 0.41041806, 0.16973795, -0.00445261, -0.21795782,
0.12717782, 0.10005609, 0.10773643, 0.23486105, 0.22297241,
-0.00076477, 0.10844541, -0.28669253, -0.25354853, 0.07614724],
dtype=float32)

In [30]: W_f[0]

Out[30]: array([ 0.07831699, -0.16591972, -0.09731318, -0.2518856 , -0.10960013,
0.23857257, 0.21642837, 0.07451889, -0.2394152 , -0.11519846,
-0.0911006 , -0.18471584, -0.13887428, 0.04047704, 0.25461718,
0.05831087, 0.01036477, -0.177281 , -0.22412126, 0.05594727],
dtype=float32)

```

