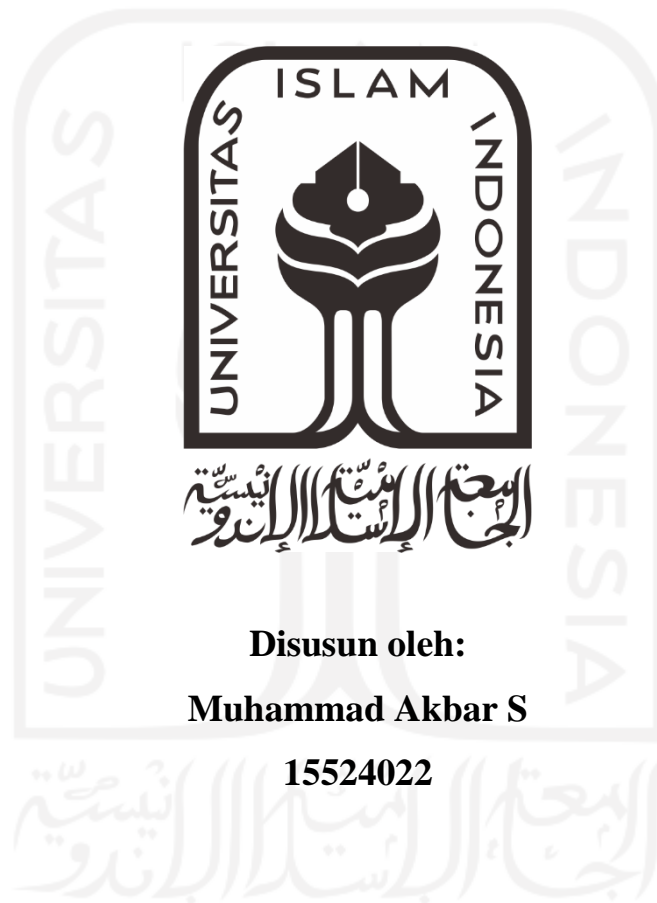


**PERBANDINGAN ALGORITME *NAÏVE BAYES CLASSIFIER* DAN *K-NEAREST NEIGHBORS* PADA PREDIKSI PERGERAKAN MATA UANG DOLLAR AMERIKA (USD) TERHADAP HARGA EMAS**

**SKRIPSI**

untuk memenuhi salah satu persyaratan  
mencapai derajat Sarjana S1



**Disusun oleh:  
Muhammad Akbar S**

**15524022**

**Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta**

**2020**

# LEMBAR PENGESAHAN

## PERBANDINGAN ALGORITME *NAÏVE BAYES CLASSIFIER* DAN *K-NEAREST NEIGHBORS* PADA PREDIKSI PERGERAKAN MATA UANG DOLLAR AMERIKA (USD) TERHADAP HARGA EMAS

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana Teknik  
pada Program Studi Teknik Elektro  
Fakultas Teknologi Industri  
Universitas Islam Indonesia

Disusun oleh:

Muhammad Akbar S  
15524022

Yogyakarta, 31 Desember 2020

Menyetujui,

Pembimbing 1



Dzata Farahiyah, S.T., M.Sc.  
155220509

# LEMBAR PENGESAHAN

## SKRIPSI

### PERBANDINGAN ALGORITME *NAÏVE BAYES CLASSIFIER* DAN *K-NEAREST NEIGHBORS* PADA PREDIKSI PERGERAKAN MATA UANG DOLLAR AMERIKA (USD) TERHADAP HARGA EMAS

Dipersiapkan dan disusun oleh:

**Muhammad Akbar S**

15524022

Telah dipertahankan di depan dewan penguji

Pada tanggal: 15 Februari 2021

Susunan dewan penguji

Ketua Penguji : Dzata Farahiyah, S.T., M.Sc., 

Anggota Penguji 1: Ida Nurcahyani, S.T., M.Eng., 

Anggota Penguji 2: Elvira Sukma Wahyuni, S.Pd., M.Eng., 

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana

Tanggal: 15 Februari 2021

Ketua Program Studi Teknik Elektro



Yusuf Aziz Anwarulloh S.T., M.Eng., Ph.D.

045240101

## PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 31-12-2020



Muhammad Akbar S

## KATA PENGANTAR

*Assalamu 'alaikum Warahmatullahi Wabarakatuh.*

Segala puji dan syukur penulis panjatkan kepada hadirat Allah Swt yang senantiasa melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “*PERBANDINGAN ALGORITME NAÏVE BAYES CLASSIFIER DAN K-NEAREST NEIGHBORS PADA PREDIKSI PERGERAKAN MATA UANG DOLLAR AMERIKA (USD) TERHADAP HARGA EMAS*” ini dengan baik dan lancar. Skripsi ini wajib ditempuh oleh mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia sebagai salah satu syarat untuk menyelesaikan jenjang studi Strata 1.

Kelancaran dalam mempersiapkan dan menyelesaikan Skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis haturkan kepada:

1. Bapak Sulistia Budi, S.T., M.Si, dan Ibu Hazni Chairina, S.E., selaku orang tua penulis serta kakak penulis yakni Muhammad Taufiq, S.T., yang selalu memberikan doa, motivasi, dan kasih sayang yang tulus kepada penulis untuk menyelesaikan skripsi ini.
2. Ibu Dzata Farahiyah, S.T., M.Sc., selaku dosen pembimbing skripsi yang selalu membantu memberi bimbingan dan saran kepada penulis hingga selesainya skripsi ini.
3. Bapak Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D., selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia.
4. Seluruh dosen Jurusan Teknik Elektro, terima kasih sebesar-besarnya atas bimbingan dan ilmu yang telah diberikan selama menempuh kuliah dari semester awal hingga akhir di Jurusan Teknik Elektro.
5. Teman-teman mahasiswa Jurusan Teknik Elektro Universitas Islam Indonesia Angkatan 2015.
6. Teman-teman yang membantu saya dalam mengerjakan skripsi yakni, Fikri Prayoga, Muhammad Ikhsan, Darmastyo Bagas, yang telah memberikan dukungan dan doa untuk saya untuk menyelesaikan skripsi dan saya mengucapkan terima kasih.
7. Semua pihak yang telah memberikan masukan, dorongan, motivasi dan semangat dalam menyelesaikan skripsi ini.

Penulis menyadari kekurangan dari skripsi ini dan jauh dari kata sempurna, baik dari segi materi maupun segi penulisan. Segala saran dan kritik yang disampaikan sangat bermanfaat untuk penulis agar membuat karya tulis berikutnya lebih baik. Semoga skripsi ini bermanfaat bagi yang membacanya.

***Wassalamu 'alaikum Warahmatullahi Wabarakatuh.***



## ARTI LAMBANG DAN SINGKATAN

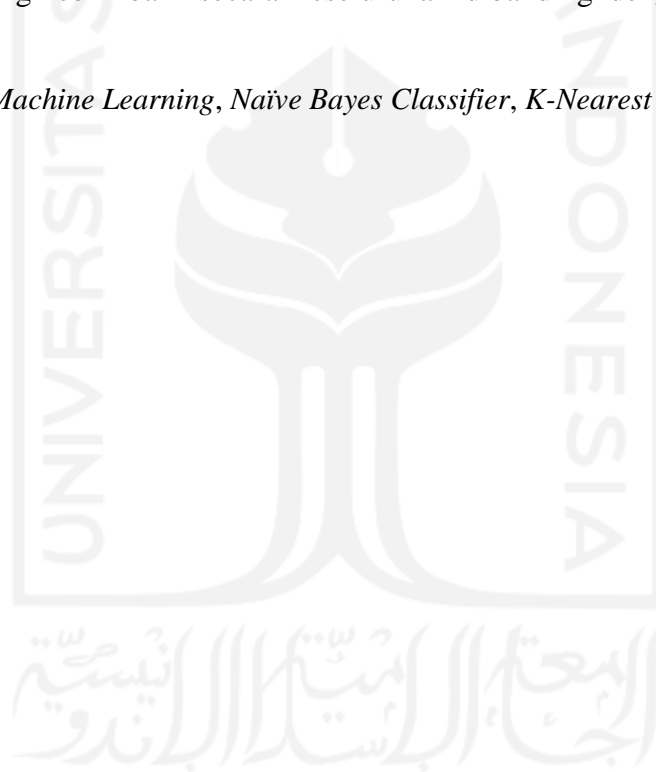
<i>Trader</i>	: Mereka yang melakukan kegiatan jual/beli saham
<i>USD/Troy Ounce</i>	: Satuan yang digunakan untuk menghargai logam berharga
<i>FN</i>	: <i>False Negative</i>
<i>FP</i>	: <i>False Positive</i>
<i>KNN</i>	: <i>K-Nearest Neighbors</i>
<i>N</i>	: <i>Negative</i>
<i>NBC</i>	: <i>Naïve Bayes Classifier</i>
<i>P</i>	: <i>Positive</i>
<i>TN</i>	: <i>True Negative</i>
<i>TP</i>	: <i>True Positive</i>



## ABSTRAK

Emas merupakan salah satu jenis investasi yang terbukti memiliki nilai tukar yang stabil dan menjadi bentuk aset yang penting. Dengan harga emas dunia yang selalu berubah-ubah dapat mengkhawatirkan para investor emas, sehingga dibutuhkan sebuah informasi yang akurat dan cepat terhadap perubahan-perubahan yang terjadi. Skripsi ini bertujuan untuk mengetahui prediksi pergerakan nilai emas dengan menggunakan metode *Machine Learning* dengan algoritme *Naïve Bayes Classifier* dan *K-Nearest Neighbors*. Fitur yang digunakan untuk memprediksi pergerakan nilai saham emas adalah nilai *Open*, nilai *High*, nilai *Low*, dan nilai *Close*. Algoritme *Naïve Bayes Classifier* dan *K-Nearest Neighbors* digunakan untuk melatih model *Machine Learning* dalam 3 skenario pengujian yakni pembagian data latih dan data uji dengan rasio 75%-25%, 80%-20%, dan 90%-10%. Pada hasil pengujian terlihat bahwa pengujian skenario 80%-20% model *Naïve Bayes Classifier* memiliki akurasi tertinggi dengan nilai 62% sedangkan akurasi tertinggi untuk model *K-Nearest Neighbors* ada pada pengujian skenario 90%-10% dengan nilai 65%. Dapat disimpulkan bahwa pengujian yang dilatih dengan menggunakan model *K-Nearest Neighbors* memiliki kinerja yang lebih baik secara keseluruhan dibanding dengan model *Naïve Bayes Classifier*.

**Kata Kunci:** Emas, *Machine Learning*, *Naïve Bayes Classifier*, *K-Nearest Neighbors*.





# DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN .....	vi
ABSTRAK .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
BAB 2 TINJAUAN PUSTAKA .....	4
2.1 Studi Literatur .....	4
2.2 Tinjauan Teori.....	5
2.2.1 <i>Trading Online</i> .....	5
2.2.2 <i>Machine Learning</i> .....	6
2.2.3 <i>Naïve Bayes Classifier (NBC)</i> .....	6
2.2.4 <i>K-Nearest Neighbors (KNN)</i> .....	7
2.2.5 <i>Confusion Matrix</i> .....	9
2.2.6 Akurasi dan Error .....	10
2.2.7 Presisi .....	11

2.2.8 <i>Recall</i> .....	11
2.2.9 <i>F-Score</i> .....	12
<b>BAB 3 METODOLOGI</b> .....	<b>13</b>
3.1 Metodologi Penelitian .....	13
3.2 Proses Penyiapan Data .....	13
3.2.1 Pengumpulan dan Pengolahan Data .....	13
3.2.2 Pembuatan Program .....	16
3.2.3 Pembagian Data Latih dan Data Uji .....	16
3.3 Proses Pelatihan Model .....	17
3.4 Proses Pengujian, Evaluasi, dan Analisis .....	17
<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	<b>19</b>
4.1 Hasil Simulasi <i>Naïve Bayes Classifier</i> .....	19
4.1.1 Akurasi dan Error pada Model <i>Naïve Bayes Classifier</i> .....	19
4.1.2 Presisi pada Model <i>Naïve Bayes Classifier</i> .....	20
4.1.3 <i>Recall</i> pada Model <i>Naïve Bayes Classifier</i> .....	20
4.1.4 <i>F-Score</i> pada Model <i>Naïve Bayes Classifier</i> .....	21
4.2 Hasil Simulasi <i>K-Nearest Neighbors</i> .....	22
4.2.1 Akurasi dan Error pada Model <i>K-Nearest Neighbors</i> .....	22
4.2.2 Presisi pada Model <i>K-Nearest Neighbors</i> .....	23
4.2.3 <i>Recall</i> pada Model <i>K-Nearest Neighbors</i> .....	24
4.2.4 <i>F-Score</i> pada Model <i>K-Nearest Neighbors</i> .....	25
4.3 Perbandingan Kinerja Antar Algoritme .....	26
4.4 Pengujian Dengan <i>K-Fold Cross Validation</i> .....	28
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....	<b>30</b>
5.1 Kesimpulan .....	30
5.2 Saran .....	30
<b>DAFTAR PUSTAKA</b> .....	<b>31</b>
<b>LAMPIRAN</b> .....	<b>1</b>

## DAFTAR GAMBAR

Gambar 3.1 Blok Diagram Program .....	13
Gambar 3.2 Struktur <i>Dataset</i> .....	14
Gambar 4.1 Hasil Grafik Akurasi dan Error pada Model <i>Naïve bayes Classifier</i> .....	19
Gambar 4.2 Hasil Grafik Nilai Presisi Pada Model <i>Naïve Bayes Classifier</i> .....	20
Gambar 4.3 Hasil Grafik Nilai <i>Recall</i> Pada Model <i>Naïve Bayes Classifier</i> .....	21
Gambar 4.4 Hasil Grafik Nilai <i>F-Score</i> Pada Model <i>Naïve Bayes Classifier</i> .....	22
Gambar 4.5 Hasil Grafik Nilai Akurasi dan Error Pada Model <i>K-Nearest Neighbors</i> .....	23
Gambar 4.6 Hasil Grafik Nilai Presisi Pada Model <i>K-Nearest Neighbors</i> .....	24
Gambar 4.7 Hasil Grafik Nilai <i>Recall</i> Pada Model <i>K-Nearest Neighbors</i> .....	25
Gambar 4.8 Hasil Grafik Nilai <i>F-Score</i> Pada Model <i>K-Nearest Neighbors</i> .....	26
Gambar 4.9 Hasil Grafik Nilai Pada Masing-Masing Parameter Pengujian Model .....	27



## DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i> .....	10
Tabel 2.2 <i>Confusion Matrix</i> $2 \times 2$ .....	10
Tabel 3.1 <i>Data Input dan Output</i> .....	14
Tabel 3.2 <i>Label Encoding</i> .....	15
Tabel 3.3 Skenario Pengujian <i>Dataset</i> .....	16
Tabel 3.4 Pembagian Data Latih dan Data Uji.....	17
Tabel 4.1 Akurasi Data Latih .....	28
Tabel 4.2 Akurasi <i>K-Fold Cross Validation</i> .....	29



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Emas merupakan salah satu jenis investasi yang terbukti memiliki nilai tukar yang stabil dan menjadi bentuk aset yang penting. Para investor umumnya membeli emas untuk pegangan atau *safe haven* jika suatu saat terjadi beberapa krisis termasuk ekonomi, politik, sosial atau krisis yang berbasis mata uang.

Menurut Yuliga, Rusli, dan Edy [1], mereka mengatakan bahwa ada banyak kasus di mana banyak calon nasabah atau investor dalam saham emas ini masih belum percaya karena kurangnya pengetahuan mereka tentang apa itu saham yang bergerak dalam bidang emas. Masih takut tertipu dengan pialang-pialang saham emas karena mereka tidak mengetahui kapan harga emas akan naik atau turun sehingga apabila salah langkah dalam transaksi dapat mengakibatkan kerugian bagi para investor. Maka dari itu, dibutuhkan pemanfaatan teknologi komputasi untuk memprediksi pergerakan saham emas, salah satunya adalah menggunakan *Machine Learning*.

*Machine Learning* adalah teknik yang sangat populer karena paling banyak digunakan untuk menggantikan atau menirukan perilaku manusia dalam menyelesaikan suatu masalah, dan akan mencoba menirukan bagaimana proses kerja yang telah dilakukan oleh manusia atau makhluk cerdas dengan belajar, menggeneralisasi, dan mengimplementasikannya. Hal ini sangat membantu manusia dalam penyelesaian masalah tersebut [2].

Penelitian ini penulis bertujuan untuk meningkatkan nilai keakuratan prediksi keuntungan atau *profit* pada investasi emas dengan berupaya untuk melakukan analisis pergerakan harga emas apakah akan naik atau turun dengan menggunakan algoritme *Naïve Bayes Classifier* (NBC) dan *K-Nearest Neighbors* (KNN) yang mana merupakan metode klasifikasi *Machine Learning*. Dipilihnya algoritme KNN dan NBC karena kedua algoritme ini bekerja secara sederhana dan mudah untuk diimplementasikan dalam melakukan klasifikasi, telah terbukti juga dapat menghasilkan nilai akurasi yang terbaik. Yang menjadi perbedaan pada kedua algoritme ini adalah caranya melakukan klasifikasi di mana algoritme KNN melakukan klasifikasi objek berdasarkan data latih terdekat pada ruang fitur sedangkan algoritme NBC menggunakan konsep probabilitas untuk melakukan klasifikasi [3]. Dengan begitu para investor dapat meningkatkan nilai keakuratan dari prediksi kapan akan melakukan jual/beli atau *trading online*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan sebelumnya maka didapati beberapa rumusan masalah, yakni:

1. Bagaimana kinerja dari model prediksi pergerakan mata uang USD terhadap harga emas dunia dengan menggunakan Algoritme *Naïve Bayes Classifier*?
2. Bagaimana kinerja dari model prediksi pergerakan mata uang USD terhadap harga emas dunia dengan menggunakan Algoritme *K-Nearest Neighbors*?
3. Algoritme mana yang terbaik di antara *Naïve Bayes Classifier* dan *K-Nearest Neighbors* dalam prediksi pergerakan mata uang USD terhadap harga emas dunia?

## 1.3 Batasan Masalah

Adapun batasan masalah yang membatasi lingkup penelitian ini, yakni:

1. Fitur yang digunakan untuk melakukan prediksi pergerakan adalah Nilai *Open, High, Low, Close*.
2. Data yang digunakan untuk ke empat fitur tersebut diperoleh dari aplikasi *trading online Metatrader*. (aplikasi bisa diunduh dari website <https://www.metatrader4.com>, dan data dari aplikasi ini diakses secara *real time*)
3. *Dataset* yang digunakan pada penelitian ini mengambil *range* dari tahun 2005–2020 dan satu sampel data bersifat sebulan.
4. Pembagian data latih dan data uji dilakukan secara acak dengan bantuan *scikit-learn library*.
5. Perhitungan komputasi algoritme *Naïve Bayes Classifier* dan *K-Nearest Neighbors* menggunakan bantuan *Scikit-Learn Library*.
6. Rasio skenario pembagian data latih dan data uji yang digunakan, yakni 75:25, 80:20, 90:10.
7. Parameter K untuk model algoritme *K-Nearest Neighbors* ditetapkan bernilai 9 ( $K=9$ ), dan didapat dengan menggunakan bantuan *GridSearchCV* untuk menentukan nilai K terbaik.
8. Parameter yang digunakan untuk mengevaluasi kinerja model adalah akurasi, presisi, *recall*, dan *f-score*.

## 1.4 Tujuan Penelitian

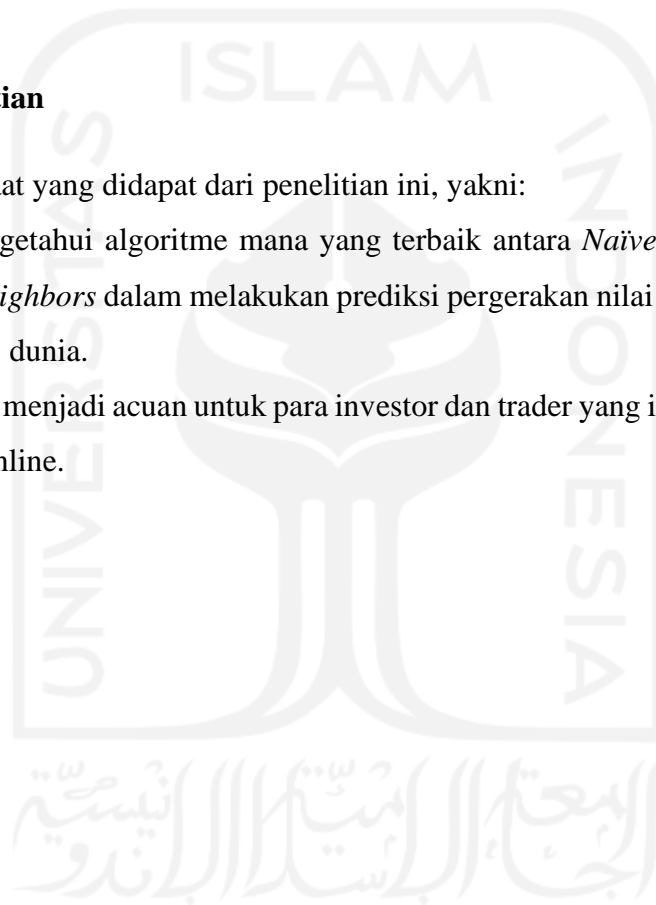
Tujuan dari penelitian ini dibagi menjadi beberapa hal, yakni:

1. Untuk mengetahui kinerja dari model prediksi pergerakan mata uang USD terhadap harga emas dunia dengan menggunakan Algoritme *Naïve Bayes Classifier*.
2. Untuk mengetahui kinerja dari model prediksi pergerakan mata uang USD terhadap harga emas dunia dengan menggunakan Algoritme *K-Nearest Neighbors*.
3. Untuk membandingkan algoritme mana yang terbaik antara *Naïve Bayes Classifier* dan *K-Nearest Neighbors* dalam memprediksi pergerakan mata uang USD terhadap harga emas dunia.

## 1.5 Manfaat Penelitian

Adapun Manfaat yang didapat dari penelitian ini, yakni:

1. Dapat mengetahui algoritme mana yang terbaik antara *Naïve Bayes Classifier* dan *K-Nearest Neighbors* dalam melakukan prediksi pergerakan nilai mata uang USD terhadap harga emas dunia.
2. Agar dapat menjadi acuan untuk para investor dan trader yang ingin melakukan Jual/Beli investasi online.



## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Studi Literatur

Pada Penelitian yang dilakukan Iswanto, Yuliana, dan Reddy [4], peneliti mencoba menerapkan algoritme klasifikasi untuk membuat sebuah sistem rekomendasi dalam mendukung keputusan Jual-Beli salah satu alternatif yang tepat untuk mengatasi masalah ini. Algoritme *K-Nearest Neighbors* (KNN) dipilih karena metode KNN merupakan suatu bentuk algoritme yang dapat dipakai dalam membangun sistem rekomendasi yang dapat mengklasifikasikan data berdasarkan jarak terdekat. Peneliti merancang sistem untuk membantu *trader* dalam mengambil keputusan Jual-Beli, berdasarkan data hasil prediksi. Hasil sistem rekomendasi dari sepuluh percobaan hasil prediksi arima direkomendasikan beli. Lalu saat dibandingkan dengan harga di lapangan *target* keuntungan 7% per minggu dari sepuluh percobaan apabila dirata-rata keuntungan telah melampaui target.

Pada penelitian yang dilakukan Irfan dan Aris [5], peneliti menerapkan algoritme *Support Vector Machine* menggunakan *framework rapidminer* sebagai alat bantu (indikator) untuk memprediksi kurva forex yang berjalan secara *real-time* pada aplikasi *insta Forex*. Pergerakan kurva dijadikan sebagai salah satu indikator dalam pengambilan keputusan untuk beli (*buy*) atau jual (*sell*). Penelitian ini mengusulkan algoritme *Support Vector Machine* (SVM) menggunakan kernel anova untuk memprediksi pergerakan kurva pada *live time trading forex* menggunakan data GBPUSD, 1H. Hasil penelitian menunjukkan bahwa *trend* kurva yang dihasilkan Algoritme *Support Vector Machine* menyerupai pola kurva *online trading forex*, dan hasil prediksi trend (high) sama dengan *real data trading forex online*. Maka dapat diambil kesimpulan bahwa algoritme *Support Vector Machine* dapat digunakan sebagai indikator prediksi pergerakan kurva pada *trading forex*.

Dalam penelitian yang dilakukan Nugroho [6], peneliti memprediksi penutupan harga emas menggunakan metode algoritme *Support Vector Machine* untuk membandingkan variabel A (*open, high, low* dan *close*) dengan variabel B (*open, high, low, close* dan *factory news*) yang ditingkatkan kinerjanya dengan memaksimalkan parameter. Dari hasil pengujian dengan mengukur metode *Support Vector Machine* menggunakan RMSE, diketahui bahwa variabel A menghasilkan nilai RMSE 4,695 dan variabel B nilai RMSE adalah 4,620. Dengan hasil RMSE yang telah didapat, maka variabel B (*open, high, low, close* dan *factory news*) dapat meningkatkan hasil prediksi yang lebih akurat.



Pada Penelitian yang dilakukan oleh Fredy, Tarno, dan Di Asih [7], peneliti menerapkan metode *Naïve Bayes* dan *Bayesian Regularization Neural Network* (BRNN) untuk melakukan klasifikasi atau deteksi sinyal *Stochastic Oscillator*, dan kemudian membandingkan tingkat akurasi hasil klasifikasi antara kedua metode. Hasil penelitian ini menunjukkan bahwa hanya terdapat enam atribut yang dapat digunakan untuk mengidentifikasi apakah suatu sinyal yang keluar merupakan sinyal palsu atau tidak, yaitu kondisi IHSG, kondisi *high price*, kondisi *low price*, kondisi *close price*, posisi %K, dan posisi %D, serta tingkat akurasi dari metode *Naïve Bayes* adalah sebesar 76,92%, sedangkan akurasi dari metode BRNN adalah sebesar 80,77%.

Penelitian [7] tersebut menerapkan *Naïve Bayes Classifier* dan *Bayesian Regularization Neural Network* untuk melakukan klasifikasi dalam mendeteksi sinyal *Stochastic Oscillator*, dan terbukti algoritme klasifikasi ini mendapatkan hasil yang cukup baik yakni 76,92%. Peneliti ingin mencoba menerapkan algoritme klasifikasi *Naïve Bayes Classifier* dan *K-Nearest Neighbors* dengan tiga atribut sebelumnya digunakan oleh peneliti [7] seperti nilai *High*, nilai *Low*, dan nilai *Close*. Lalu menambahkan satu atribut baru yakni nilai *Open* dan menggunakan lebih banyak sampel untuk melakukan prediksi pergerakan mata uang dollar terhadap harga emas agar mendapatkan hasil yang lebih optimal, seperti yang dilakukan peneliti [6] dengan menerapkan algoritme klasifikasi *Support Vector Machine* untuk melakukan prediksi penutupan harga emas dengan empat atribut yang sama, dan mendapatkan hasil yang cukup akurat.

## **2.2 Tinjauan Teori**

### **2.2.1 Trading Online**

*Trading online* adalah suatu bentuk jual-beli (valas, *future*, komoditas) yang dijalankan secara virtual dengan bantuan perangkat komputer atau *mobile* dan jaringan internet. *Trading* atau investasi itu sendiri merupakan kegiatan jual beli saham yang bertujuan untuk mendapatkan keuntungan dari selisih harga beli awal dengan harga jual. Dengan adanya internet di masa sekarang hal ini dapat dengan mudah dilakukan secara *online*, *trader* atau yang melakukan investasi tidak perlu membuka outlet atau toko secara real karena seluruh pergerakan uang mereka akan diatur melalui perantara dunia maya.

Emas merupakan salah satu logam mulia yang sampai saat ini masih digunakan sebagai alat tukar dalam melakukan suatu transaksi. Bahkan, emas menjadi alat tukar resmi sebelum ditemukan uang. Investasi emas dapat dilakukan dengan berbagai bentuk. Misalnya, investasi emas batangan, investasi emas simpanan, reksadana emas, saham pertambangan emas, emas

berjangka, perhiasan, dan koin emas. Emas banyak dipilih oleh para investor karena nilainya yang cenderung stabil [5].

### 2.2.2 Machine Learning

*Machine Learning* atau sebutan lainnya pembelajaran mesin adalah teknik yang sangat populer karena paling banyak digunakan untuk menggantikan atau menirukan perilaku manusia dalam menyelesaikan suatu masalah, dan akan mencoba menirukan bagaimana proses kerja yang telah dilakukan oleh manusia atau makhluk cerdas dengan belajar, menggeneralisasi, dan mengimplementasikannya. Hal ini sangat membantu manusia dalam penyelesaian masalah tersebut. Ada dua pengaplikasian utama dalam *Machine Learning* yakni, klasifikasi dan prediksi. Ciri-ciri dari *Machine Learning* ini adalah adanya proses pembelajaran dan proses pelatihan atau *training*. Karena hal ini juga *Machine Learning* membutuhkan *dataset* yang sangat banyak untuk dipelajari yang disebut juga dengan *data training*. Klasifikasi adalah salah satu metode yang ada pada *Machine Learning* yang digunakan oleh mesin dalam mengklasifikasi atau memilah objek berdasarkan ciri-ciri tertentu yang telah diprogram sebagaimana manusia mencoba untuk membedakan benda yang satu dengan benda yang lainnya [2].

### 2.2.3 Naïve Bayes Classifier (NBC)

*Naïve Bayes Classifier* pada dasarnya adalah sebuah metode dalam pengklasifikasian probabilistik sederhana yang dapat menghitung sekelompok probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari *dataset* yang diberikan. Algoritme menggunakan teorema Bayes dan mengasumsikan semua atribut *independence* atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas [8]. Definisi lainnya mengatakan *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang ditemukan oleh ilmuwan yang berasal dari Inggris bernama Thomas Bayes, yakni dengan memprediksi peluang-peluang yang ada di masa depan berdasarkan pengalaman sebelumnya di masa lalu [9]. Keuntungan penggunaan metode *Naïve Bayes* adalah bahwa dalam proses pengklasifikasian untuk menentukan estimasi parameter yang diperlukan nantinya hanya membutuhkan jumlah data pelatihan atau bisa disebut sebagai *Training Data* yang kecil. *Naïve Bayes* dapat bekerja dengan sangat baik dalam kebanyakan situasi atau masalah dunia nyata yang lebih kompleks lebih dari yang diinginkan [10]. Teorema Bayes ditunjukkan oleh persamaan berikut (2.1)

$$P(A|B) = \frac{(P(B|A) \times P(A))}{P(B)} \quad (2.1)$$

Di mana:

$P(A|B)$  : Probabilitas Label A berdasarkan Fitur B (*posterior probability*)

$P(B|A)$  : Probabilitas Fitur B berdasarkan pada Label A (*likelihood*)

$P(A)$  : Probabilitas Label A (*prior probability*)

$P(B)$  : Probabilitas Fitur B (*evidence*)

Penentuan hasil prediksi label suatu sampel data dilakukan dengan membandingkan *posterior probability* masing-masing label, dan label yang memiliki *posterior probability* paling tinggi akan dijadikan sebagai hasil prediksi. Penelitian yang dilakukan menggunakan dua label yakni “Naik/Beli” dan “Turun/Jual”. Adapun fitur yang digunakan adalah *Open, High, Low, dan Close*.

Berikut adalah penerapan (2.1) dengan menggunakan fitur dan label yang digunakan pada penelitian ini, dan menghasilkan persamaan (2.2) dan (2.3) untuk menghitung masing- masing *posterior probability* label.

$$P(\text{Naik}|X_1, X_2, X_3, X_4) = \frac{P(X_1, X_2, X_3, X_4) \times P(\text{Naik})}{P(X_1, X_2, X_3, X_4)} \quad (2.2)$$

$$P(\text{Turun}|X_1, X_2, X_3, X_4) = \frac{P(X_1, X_2, X_3, X_4) \times P(\text{Turun})}{P(X_1, X_2, X_3, X_4)} \quad (2.3)$$

Di mana:

$X_1$  : Nilai *Open*

$X_2$  : Nilai *High*

$X_3$  : Nilai *Low*

$X_4$  : Nilai *Close*

Penelitian ini menggunakan bantuan *scikit-learn library* untuk melakukan perhitungan *posterior probability*. Pada *scikit-learn library* terdapat algoritme komputasi untuk mempermudah perhitungan yang dilakukan pada jumlah data yang banyak.

#### 2.2.4 K-Nearest Neighbors (KNN)

*K-Nearest Neighbors* merupakan salah satu metode klasifikasi yang mengelompokkan data ke kelompok yang memiliki sifat yang paling mirip dengannya. KNN dilakukan dengan mencari kelompok K objek dalam data *training* yang paling dekat. Algoritme ini hanya mengingat data yang sudah ada dan tidak mempelajari cara mengategorikan data [11].

Pada KNN, parameter K merupakan jumlah dari tetangga terdekat. Algoritme ini menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel data uji yang baru. Pada saat diberikan sampel data uji, maka algoritme akan menemukan sejumlah K objek (titik latih) yang paling dekat dengan titik uji. Pada penelitian ini, nilai K dan metode pengukuran jarak ditentukan dengan bantuan *GridSearchCV*, dan didapatkan nilai K dan metode pengukuran jarak yang terbaik yakni K = 9 dan jarak *Minkowski*. Perhitungan jarak pada penelitian ini dirumuskan oleh persamaan berikut (2.4).

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2.4)$$

Di mana:

- d(x,y) : Jarak
- $x_i$  : Sampel data latih
- $y_i$  : Sampel data uji
- n : Dimensi data
- i : Variabel data

Berdasarkan fitur pada penelitian ini, maka perhitungan jarak antar suatu sampel data latih dan data uji dapat dijabarkan seperti persamaan berikut (2.5).

$$d(x_i, y_i) = |A_{x_i} - A_{y_i}| + |B_{x_i} - B_{y_i}| + |C_{x_i} - C_{y_i}| + |D_{x_i} - D_{y_i}| \quad (2.5)$$

Di mana:

- d(x,y) : Jarak antara data latih dan data uji
- $x_i$  : Sampel data latih
- $y_i$  : Sampel data uji
- A : Nilai *Open*
- B : Nilai *High*
- C : Nilai *Low*
- D : Nilai *Close*
- i : Variabel data

Pada penelitian ini, perhitungan jarak dilakukan dengan menggunakan bantuan *scikit-learn library* yang mana di dalamnya terdapat algoritme komputasi yang mempermudah melakukan perhitungan pada jumlah data yang banyak.

### 2.2.5 Confusion Matrix

*Confusion Matrix* adalah tabel yang digunakan untuk mengukur kinerja pada sebuah model klasifikasi pada satu set data uji yang nilai sebenarnya telah diketahui. *Confusion Matrix* dapat membantu dalam visualisasi kinerja suatu algoritme. Seperti pada Tabel 2.1 yang menampilkan hasil prediksi pada masalah klasifikasi, jumlah prediksi benar dan salah dirangkum dengan nilai-nilai dan dipecah kepada masing-masing label. Hal ini dapat membantu untuk mengetahui kesalahan yang dibuat oleh *Classifier*.



Tabel 2.1 *Confusion Matrix*

<i>Predict Label</i>	<i>Actual Label</i>	
	<i>Positive (P)</i>	<i>Negative (N)</i>
<i>Positive (P)</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
<i>Negative (N)</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Di mana:

*True Positive* : Merupakan data positif dan diprediksi benar

*True Negative* : Merupakan data negatif dan diprediksi benar

*False Positive* : Merupakan data negatif, namun diprediksi sebagai data positif (error tipe 1)

*False Negative*: Merupakan data positif, namun diprediksi sebagai data negatif (error tipe 2)

Dari tabel di atas dapat diterapkan dua label klasifikasi yang akan digunakan untuk penelitian dan ditunjukkan pada Tabel 2.2 berikut.

Tabel 2.2 *Confusion Matrix 2 × 2*

<i>Predict Label</i>	<i>Actual Label</i>	
	Label Naik (U)	Label Turun (D)
Label Naik (U)	UU	UD
Label Turun (D)	DU	DD

## 2.2.6 Akurasi dan Error

**Akurasi** adalah jumlah prediksi yang benar dibagi dengan keseluruhan sampel data. Akurasi didapat dengan menggunakan persamaan berikut (2.6).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

Karena penelitian ini menggunakan dua label sehingga *Confusion Matrix* menjadi  $2 \times 2$ , dan penerapan persamaan berdasarkan pada tabel, menghasilkan persamaan berikut (2.7).

$$\text{Akurasi} = \frac{UU + DD}{UU + UD + DU + DD} \quad (2.7)$$

**Error** merupakan seberapa banyak prediksi yang salah, dinyatakan dengan persamaan berikut (2.8).

$$\text{Error} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.8)$$

lalu, persamaan berubah mengikuti Tabel 2.2, sehingga menjadi persamaan berikut (2.9).

$$\text{Error} = \frac{\text{UD} + \text{DU}}{\text{UU} + \text{UD} + \text{DU} + \text{DD}} \quad (2.9)$$

Bisa juga persamaan (2.9) ditulis menjadi persamaan berikut (2.10).

$$\text{Error} = 1 - \text{Akurasi} \quad (2.10)$$

### 2.2.7 Presisi

Presisi merupakan rasio dari berapa banyak prediksi yang benar-benar positif dari semua kelas positif yang diprediksi dengan benar. Nilai presisi dapat dihitung dengan menggunakan persamaan berikut (2.11).

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.11)$$

Karena setiap label memiliki prediksi positif, maka untuk menghitung presisi dari *Classifier* secara keseluruhan harus dihitung terlebih dahulu presisi per labelnya mengikuti tabel menggunakan persamaan berikut (2.12).

$$\text{Presisi } i = \frac{U_i}{U_i + D_i} \quad (2.12)$$

Setelah itu, melakukan perhitungan presisi total dengan menggunakan persamaan berikut (2.13).

$$\text{Presisi} = \frac{\text{Presisi U} + \text{Presisi D}}{2} \quad (2.13)$$

### 2.2.8 Recall

*Recall* atau *sensitivity* merupakan seberapa banyak prediksi yang benar dari semua kelas positif. Nilai *recall* dapat dihitung dengan menggunakan persamaan berikut (2.14).

$$Recall = \frac{TP}{TP + FN} \quad (2.14)$$

Untuk penelitian ini setiap label memiliki prediksi positif, dan untuk menghitung *recall* dari *Classifier* secara menyeluruh harus menghitung terlebih dahulu masing-masing prediksi labelnya mengikuti tabel dan didapatkan persamaan berikut (2.15).

$$Recall = \frac{U_i}{U_i + D_i} \quad (2.15)$$

Maka *recall* dari *Classifier* didapat menggunakan persamaan berikut (2.16).

$$Recall = \frac{Recall U + Recall D}{2} \quad (2.16)$$

### 2.2.9 F-Score

Presisi dan *recall* merupakan parameter untuk mengukur kinerja model yang sama-sama penting, namun dua parameter ini sedikit bertolak belakang. *Recall* bisa ditingkatkan semaksimal mungkin dengan cara memperbanyak prediksi sampel pada kelas positif yang mengakibatkan FP juga bertambah. Hal ini akan membuat presisi semakin turun karena tujuan utama dari evaluasi model menggunakan presisi adalah mengurangi jumlah FP. Berlaku juga sebaliknya untuk *recall* yang dipengaruhi oleh FN. Sehingga dibutuhkan sebuah cara untuk mendapatkan model yang seimbang, yakni dengan menggunakan *f-score*.

*F-Score* akan mencari nilai tengah dari kedua parameter ini, yang mana merupakan rata-rata yang harmonik dari presisi dan *recall* [12]. *F-Score* dapat dihitung dari persamaan berikut (2.17).

$$F - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.17)$$

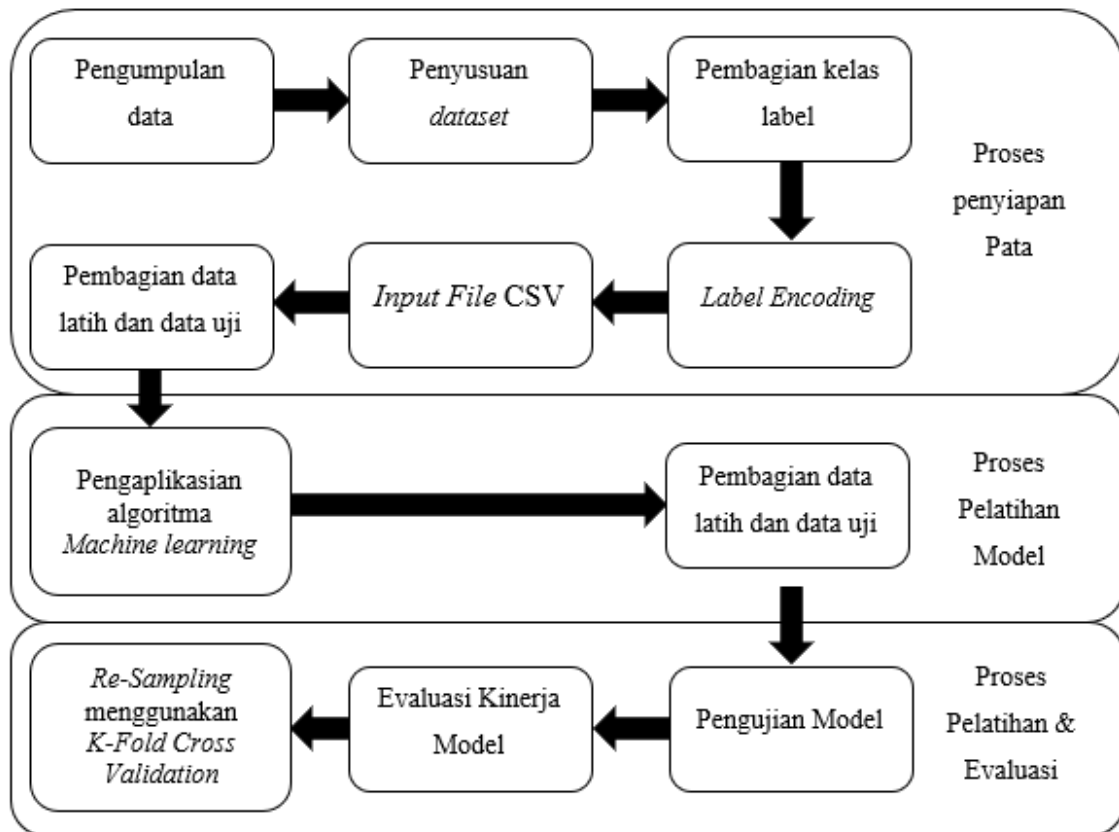


## BAB 3

### METODOLOGI

#### 3.1 Metodologi Penelitian

Agar lebih terstruktur, langkah-langkah yang dilakukan penulis untuk penelitian ini akan diilustrasikan oleh blok diagram pada Gambar 3.1 serta penjelasan mengenai setiap langkahnya.



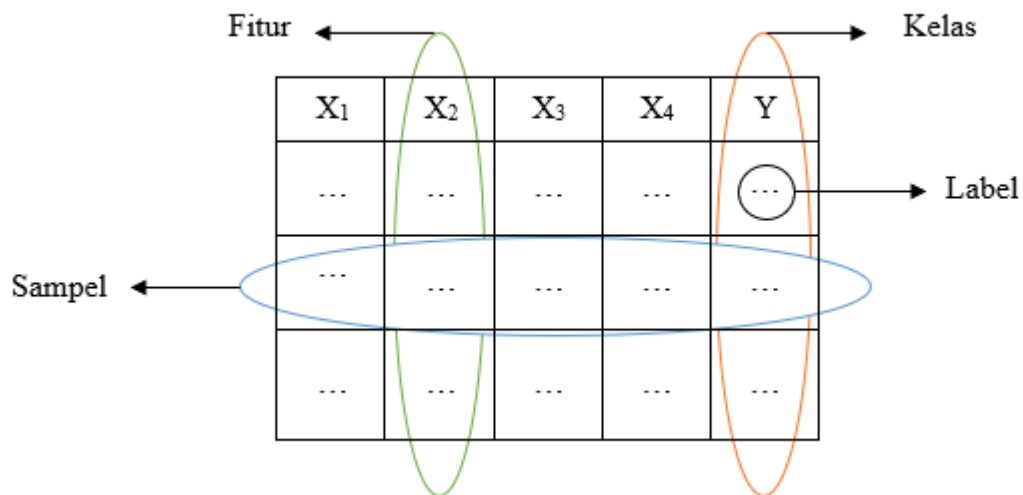
Gambar 3.1 Blok Diagram Program

#### 3.2 Proses Penyiapan Data

##### 3.2.1 Pengumpulan dan Pengolahan Data

Setelah menentukan masalah yang ingin diselesaikan dan membaca literatur terkait, tahap selanjutnya adalah mengumpulkan dan menyiapkan *dataset*. Metode *Machine Learning* melakukan pembelajaran dengan menggunakan data yang diberikan untuk membuat sebuah keputusan atau memberikan prediksi. Maka dari itu, data merupakan suatu hal yang sangat penting. Penelitian ini akan menggunakan jenis pembelajaran klasifikasi (*Classification*) di mana variabel

*output* yang coba diprediksi berupa kategori. *Dataset* pada *Machine Learning* terdiri dari beberapa bagian seperti yang ditunjukkan oleh Gambar 3.2 berikut.



Gambar 3.2 Struktur *Dataset*

Baris pada *dataset* disebut dengan sampel atau data *point* dan kolom pada *dataset* disebut dengan fitur atau atribut, kemudian kolom terakhir pada *dataset* disebut sebagai kelas. Fitur merupakan properti individual yang terukur atau karakteristik dari fenomena objek yang diamati (*input*) sedangkan kelas merupakan suatu hal yang coba diprediksi (*output*). Beberapa fitur yang digunakan untuk memprediksi pergerakan nilai emas adalah Nilai *Open*, Nilai *High*, Nilai *Low*, Nilai *Close*. Fitur-fitur ini dipilih untuk mencoba membuat pola pada model klasifikasi untuk memprediksi pergerakan nilai emas seperti yang diperlihatkan pada Tabel 3.1.

Data yang digunakan pada penelitian ini diambil dalam rentang waktu Januari 2005-Desember 2020. Terdapat 192 sampel data yang diperoleh setelah melakukan pengolahan data.

Tabel 3.1 Data *Input* dan *Output*

Atribut Data	Keterangan
Nilai <i>Open</i>	Fitur ( <i>Input</i> )
Nilai <i>High</i>	Fitur ( <i>Input</i> )
Nilai <i>Low</i>	Fitur ( <i>Input</i> )
Nilai <i>Close</i>	Fitur ( <i>Input</i> )
Pergerakan Nilai Emas	Kelas ( <i>Output</i> )

- Nilai *Open*

Nilai *Open* merupakan harga transaksi pertama harga emas pada bulan itu. Contoh pada Januari 2020 harga ditransaksikan pertama kali adalah 1520.26 USD/*Troy ounce* jadi inilah yang disebut harga *Open*. Harga *Open* sendiri tidak pasti sama dengan harga *Close* atau penutupan bulan lalu. Data didapatkan dari aplikasi *MetaTrader*. *MetaTrader* adalah salah satu jenis *software* atau aplikasi yang biasa digunakan dalam *trading Forex*, CFD (*Contract for Difference*), dan berbagai bentuk investasi virtual lainnya.

- Nilai *High*

Harga tertinggi emas yang dicapai pada bulan itu. Ketika perdagangan masih berlangsung, harga *High* adalah harga tertinggi pada saat itu. Data didapatkan dari aplikasi *MetaTrader*. *MetaTrader* adalah salah satu jenis *software* atau aplikasi yang biasa digunakan dalam *trading Forex*, CFD (*Contract for Difference*), dan berbagai bentuk investasi virtual lainnya.

- Nilai *Low*

Harga terendah emas yang dicapai pada bulan itu. Ketika perdagangan masih berlangsung, harga *Low* adalah harga terendah pada saat itu. Data didapatkan dari aplikasi *MetaTrader*. *MetaTrader* adalah salah satu jenis *software* atau aplikasi yang biasa digunakan dalam *trading Forex*, CFD (*Contract for Difference*), dan berbagai bentuk investasi virtual lainnya.

- Nilai *Close*

Nilai *Close* juga sering disebut sebagai *last* merupakan transaksi terakhir emas pada bulan itu. Ketika perdagangan masih berlangsung, harga *Close* adalah harga terakhir pada saat itu. Data didapatkan dari aplikasi *MetaTrader*. *MetaTrader* adalah salah satu jenis *software* atau aplikasi yang biasa digunakan dalam *trading Forex*, CFD (*Contract for Difference*), dan berbagai bentuk investasi virtual lainnya.

Agar label Pergerakan Naik dan Turun dapat dibaca dan dikomputasi lebih cepat oleh program, dilakukan *Label Encoding*. Yakni mengubah *string* atau huruf menjadi data numerik. Proses *encoding* dilakukan dengan menggunakan bantuan perangkat lunak *Microsoft Excel*. Adapun label yang telah diubah menjadi bentuk numerik ditampilkan oleh Tabel 3.2 berikut.

Tabel 3.2 *Label Encoding*

Label	Hasil <i>Encoding</i>
Naik	1
Turun	0

### 3.2.2 Pembuatan Program

Tahap selanjutnya adalah membuat program dengan bantuan perangkat lunak (*software*) *Spyder* karena *software* ini memiliki antarmuka yang sangat ramah bagi pemula dan dapat diintegrasikan dengan *virtual environment Anaconda*. *Anaconda environment* yang dibuat membutuhkan instalasi *packet library*, beberapa *library* yang digunakan untuk penelitian ini adalah *pandas* dan *scikit-learn*. *Pandas* pada dasarnya berfungsi untuk membaca, memanipulasi dan analisis data. Sedangkan *scikit-learn* digunakan untuk memanggil fungsi *library* yang berisi berbagai macam algoritme pembelajaran untuk *Machine Learning*. Program yang dibuat untuk penelitian ini tercantum pada lampiran.

### 3.2.3 Pembagian Data Latih dan Data Uji

Tahap selanjutnya adalah membagi *dataset* yang telah diolah kemudian dipecah menjadi data latih dan data uji. Data latih adalah data yang akan digunakan untuk membangun model, sedangkan data uji digunakan untuk menguji seberapa baik sistem yang dibangun bekerja. Data uji harus dipisahkan dari data yang akan dilatih karena model dapat mengingat data yang digunakan untuk melatihnya sehingga prediksi akan selalu bernilai benar untuk data yang telah digunakan sebelumnya. Pada pembuatan model *Machine Learning*, pembagian rasio *dataset* untuk data latih dan data uji secara umum sering digunakan berkisar pada (60%-40%), (70%-30%), (75%-25%), (80%-20%) atau (90-10%) [13] dari keseluruhan *dataset*. Pada penelitian ini pembagian data latih dan data uji dilakukan dengan tiga macam skenario pengujian seperti pada berikut.

Tabel 3.3 Skenario Pengujian *Dataset*

Pengujian	Data Latih	Data Uji
Skenario 1	75%	25%
Skenario 2	80%	20%
Skenario 3	90%	10%

Pada pengujian skenario yang ditampilkan pada Tabel 3.3, menghasilkan jumlah dari data latih dan data uji untuk masing-masing skenario seperti yang ditampilkan pada Tabel 3.4. Pembagian data latih dan data uji dilakukan secara acak (*random*) dengan menggunakan bantuan dari *scikit-learn library*.

Tabel 3.4 Pembagian Data Latih dan Data Uji

Pengujian	Data Latih	Data Uji	Jumlah Total
Skenario 1	144	48	192
Skenario 2	153	39	192
Skenario 3	172	20	192

### 3.3 Proses Pelatihan Model

Data latih yang telah dipisah selanjutnya akan digunakan untuk membangun model klasifikasi dengan mengaplikasikan model tersebut ke dalam algoritme *Machine Learning*. Algoritme yang digunakan pada penelitian adalah *Naïve Bayes Classifier* dan *K-Nearest Neighbors*.

### 3.4 Proses Pengujian, Evaluasi, dan Analisis

Pada tahapan ini, model klasifikasi yang telah dibangun menggunakan data latih akan diuji hasil prediksi label kelasnya ketika diberikan fitur (*input*) data uji. Hasil prediksi kemudian dibandingkan dengan nilai label data uji sesungguhnya untuk melihat bagaimana kinerja dari model klasifikasi. Kinerja model dianalisis dengan menggunakan beberapa parameter yakni, akurasi, presisi, *recall*, dan *f-score*.

Proses terakhir pada penelitian ini adalah melakukan *re-sampling* untuk melihat apakah model tergeneralisasi dengan baik atau tidak, apakah terjadi *overfitting* atau *underfitting*. *overfitting* adalah suatu keadaan di mana data yang digunakan untuk pelatihan itu adalah yang "terbaik". Sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai dengan yang diharapkan). Sedangkan *underfitting* merupakan keadaan di mana model pelatihan data yang dibuat tidak mewakili keseluruhan data yang akan digunakan nantinya. Sehingga menghasilkan kinerja yang buruk dalam pelatihan data. Singkatnya apabila nilai akurasi dari data latih lebih besar dari akurasi validasi nya akan terjadi *overfitting*, dan apabila nilai akurasi dari data latih dan nilai dari akurasi validasi sama-sama rendah akan terjadi *underfitting*. Maka dari itu, pada penelitian ini akan digunakan teknik *K-Fold Cross Validation*.

*K-Fold Cross Validation* merupakan metode statistik untuk mengevaluasi kinerja generalisasi yang lebih stabil dan menyeluruh daripada menggunakan pemisahan *dataset* menjadi satu set pelatihan dan pengujian. Dalam *K-Fold Cross Validation*, data akan dibagi beberapa

bagian berulang kali (iterasi) dan kemudian model dilatih. Nilai dari parameter K menentukan seberapa banyak *dataset* yang akan dibagi dan seberapa banyak iterasi *training* dan *testing* yang akan diproses, secara umum nilai yang dipakai adalah K=5 atau K=10. Untuk penelitian ini, Nilai K yang digunakan untuk metode evaluasi *K-Fold Cross Validation* adalah 10 (K=10).

Berikutnya adalah menganalisis kinerja dari simulasi yang dilakukan pada penelitian dan hal ini dibagi menjadi dua bagian. Yang pertama adalah membandingkan kinerja masing-masing skenario pada setiap algoritme. Lalu yang kedua, membandingkan kinerja antar-algoritme secara keseluruhan. Parameter yang digunakan untuk menganalisis kinerja kedua bagian tersebut adalah akurasi, presisi, *recall*, dan *f-score*.



## BAB 4

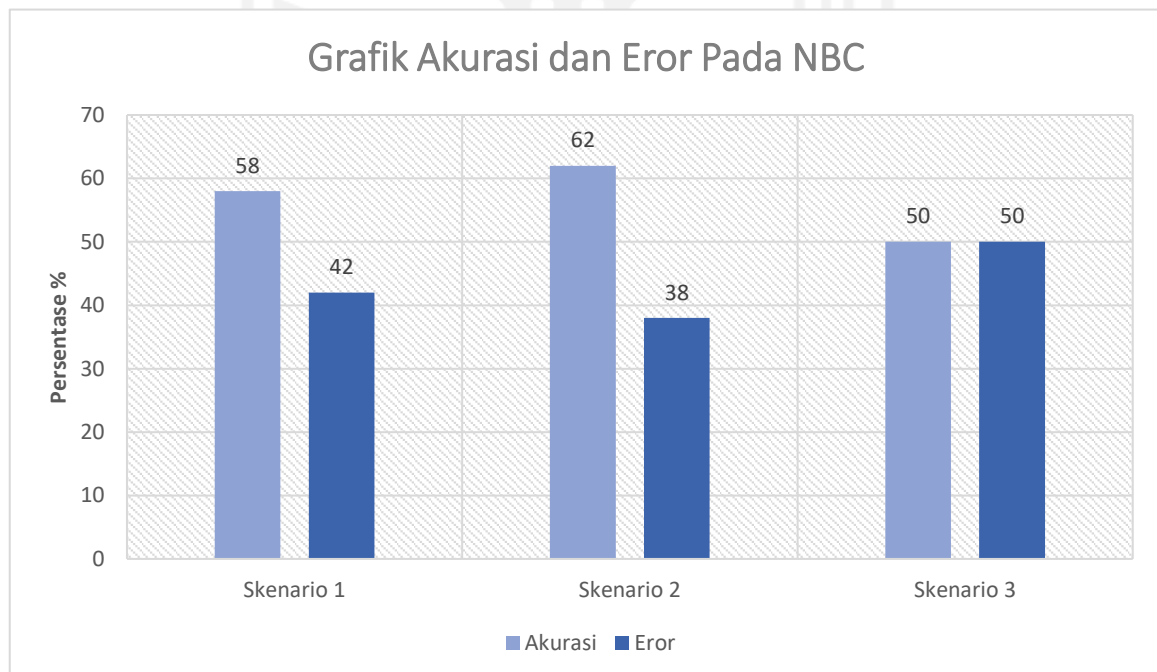
### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Simulasi *Naïve Bayes Classifier*

Pada bab ini, model *Machine Learning* sederhana yang telah dibuat selesai dilatih menggunakan algoritme *Naïve Bayes Classifier*. Prediksi yang dihasilkan kemudian dianalisis menggunakan *Confusion Matrix*. Pada *Confusion Matrix* terdapat beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model. Yakni akurasi, eror, presisi, *recall*, dan *f-score*. Hasil dari setiap skenario dianalisis berdasarkan informasi yang didapat dari *Confusion Matrix*.

##### 4.1.1 Akurasi dan Eror pada Model *Naïve Bayes Classifier*

Pada simulasi akurasi dan eror dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario yakni skenario 1 dengan pembagian data latih dan data uji dengan rasio 75%-25%, skenario 2 dengan rasio 80%-20%, dan skenario 3 dengan rasio 90%-10%. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.1



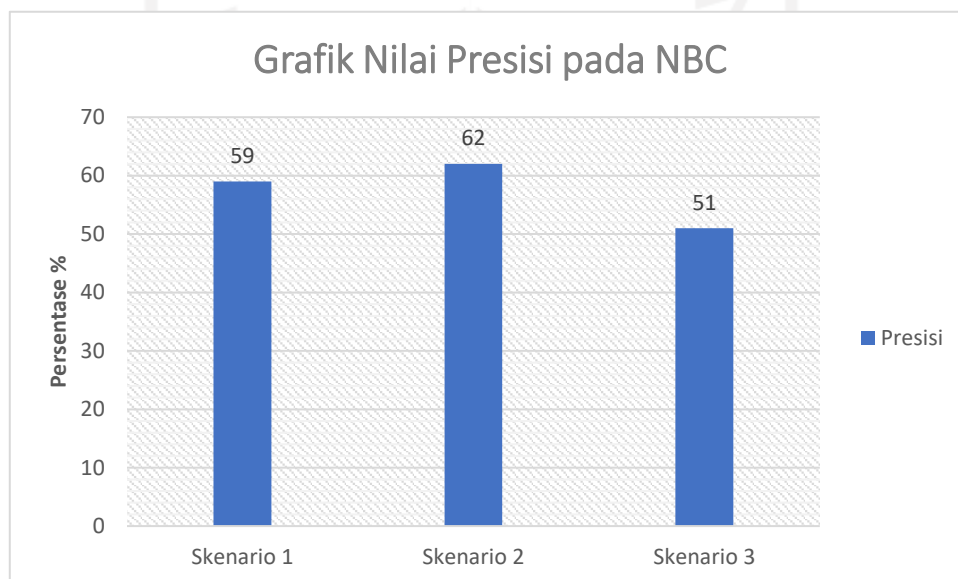
Gambar 4.1 Hasil Grafik Akurasi dan Eror pada Model *Naïve bayes Classifier*

Dapat dilihat pada hasil Gambar 4.1 bahwa akurasi dari masing-masing skenario nilainya tidak berbeda jauh satu sama lain. Pada kasus data ini, model *Naïve Bayes Classifier* memiliki

akurasi data uji disekitar 55%. Pada skenario 1 akurasi bernilai 58%, lalu pada skenario 2 akurasi naik dengan nilai 62% dan turun pada skenario 3 menjadi 50%. Hasil akurasi yang tidak linear pada ketiga skenario menunjukkan bahwa model yang dilatih kurang bisa merepresentasikan data dengan baik. Salah satu cara yang digunakan untuk mengoptimalkan model yang kurang bagus adalah dengan menambahkan data sampel.

#### 4.1.2 Presisi pada Model *Naïve Bayes Classifier*

Pada simulasi presisi dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.5.



Gambar 4.2 Hasil Grafik Nilai Presisi Pada Model *Naïve Bayes Classifier*

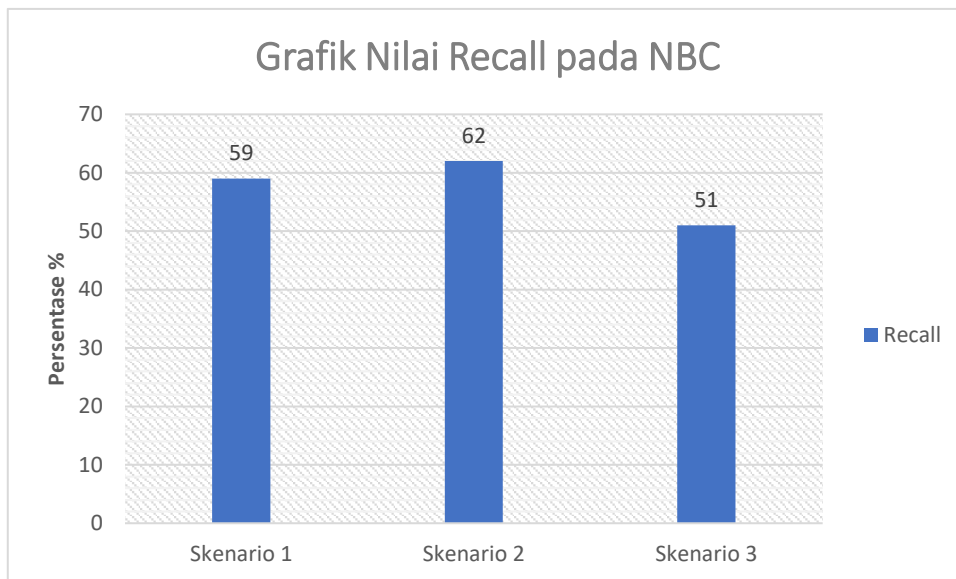
Dapat dilihat dari hasil pada Gambar 4.2 bahwa nilai presisi pada skenario 2 yang tertinggi yakni 62%. Kemudian skenario 1 memiliki nilai 59% dan turun pada skenario 3 dengan nilai terendah yakni 51%. Dari hasil ini menunjukkan bahwa pada skenario 3 dengan rasio pembagian data uji dan latih sebesar 90:10, prediksi banyak menghasilkan *false positive* sehingga membuat nilai presisi menurun [12].

#### 4.1.3 Recall pada Model *Naïve Bayes Classifier*

Pada simulasi *recall* dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion*



*Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *recall* dapat dilihat pada Gambar 4.5.

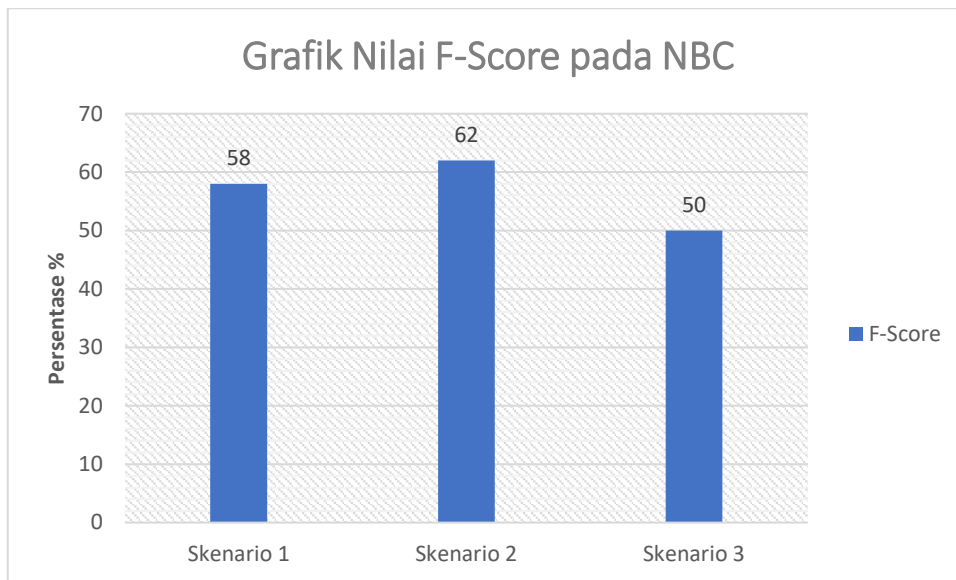


Gambar 4.3 Hasil Grafik Nilai *Recall* Pada Model *Naïve Bayes Classifier*

Dapat dilihat dari hasil pada Gambar 4.3 bahwa nilai *recall* sama dengan nilai presisi pada ketiga skenario, dengan skenario 2 memiliki nilai tertinggi yakni 62%, kemudian skenario 1 memiliki nilai 59% dan turun pada skenario 3 dengan nilai terendah yakni 51%. Pada skenario 3 tampak bahwa model tidak dilatih dengan baik, karena selain banyak menghasilkan prediksi *false positive* seperti yang diperlihatkan oleh hasil presisi, juga menghasilkan *false negative* sehingga nilai *recall* sedikit lebih rendah dari presisi.

#### 4.1.4 *F-Score* pada Model *Naïve Bayes Classifier*

Pada simulasi *f-score* dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.5.



Gambar 4.4 Hasil Grafik Nilai *F-Score* Pada Model *Naïve Bayes Classifier*

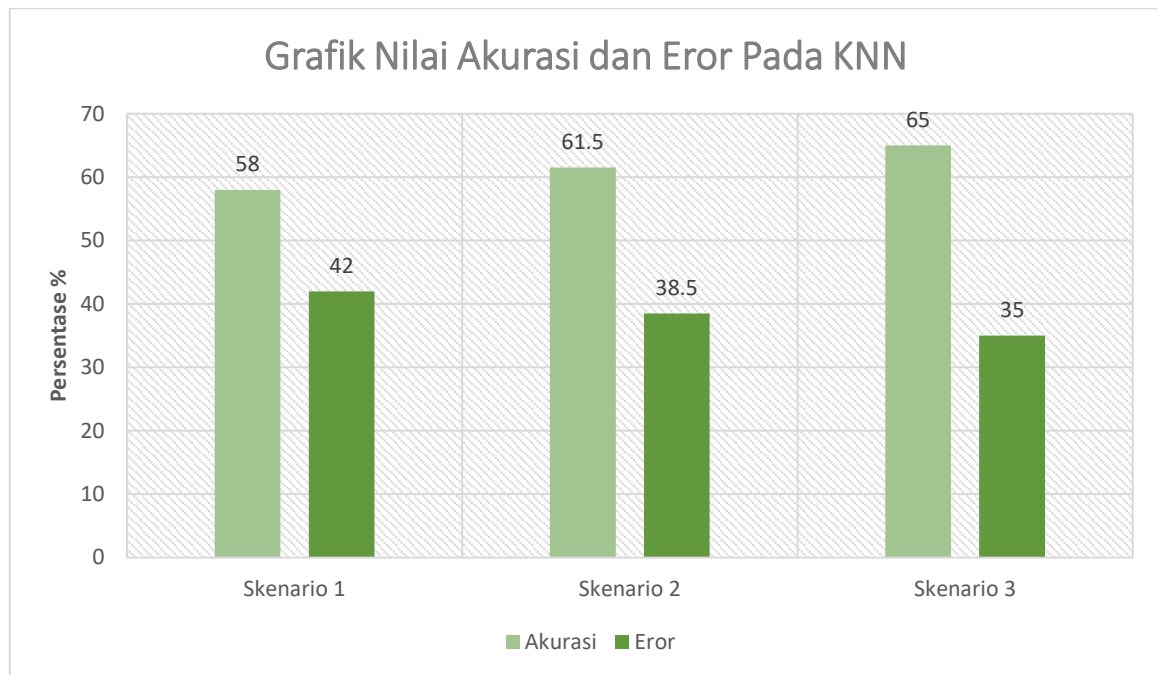
Dapat dilihat dari hasil pada Gambar 4.4 bahwa nilai *f-score* terbaik dimiliki oleh skenario 2 dengan nilai 62%, kemudian skenario 1 memiliki nilai 58%, dan nilai terendah dimiliki oleh skenario 3 yakni 50%. Hasil dari *f-score* tidak berbeda jauh dari presisi dan *recall* karena parameter ini merupakan rata-rata harmonik dari keduanya.

## 4.2 Hasil Simulasi *K-Nearest Neighbors*

Pada bab ini, model *Machine Learning* sederhana yang telah dibuat selesai dilatih menggunakan algoritme *K-Nearest Neighbors*. Prediksi yang dihasilkan kemudian dianalisis menggunakan *Confusion Matrix*. Pada *Confusion Matrix* terdapat beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model. Yakni akurasi, eror, presisi, *recall*, dan *f-score*. Hasil dari setiap skenario dianalisis berdasarkan informasi yang didapat dari *Confusion Matrix*.

### 4.2.1 Akurasi dan Eror pada Model *K-Nearest Neighbors*

Pada simulasi akurasi dan eror dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario yakni skenario 1 dengan pembagian data latih dan data uji dengan rasio 75%-25%, skenario 2 dengan rasio 80%-20%, dan skenario 3 dengan rasio 90%-10%. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.5.

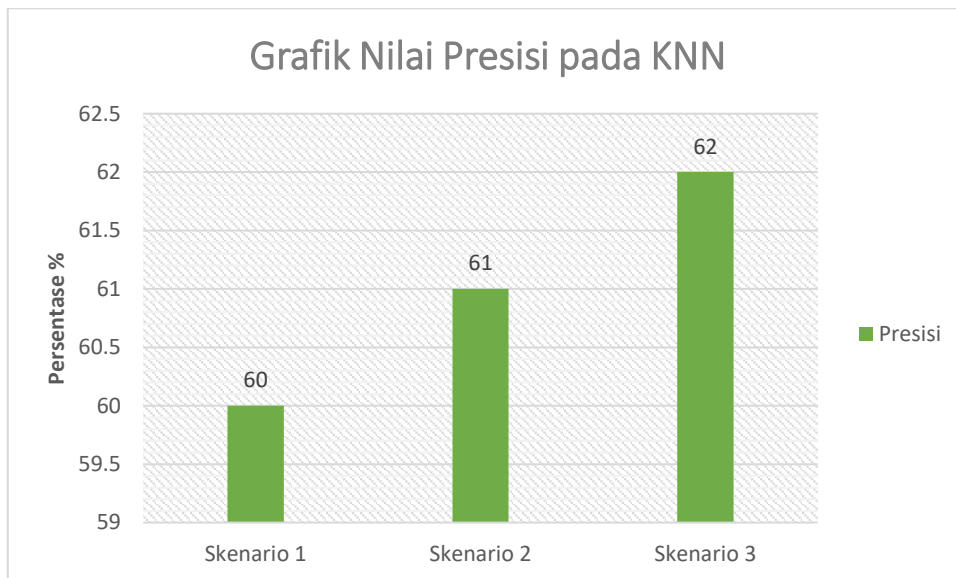


Gambar 4.5 Hasil Grafik Nilai Akurasi dan Error Pada Model *K-Nearest Neighbors*

Dapat dilihat dari hasil pada Gambar 4.5 bahwa skenario pembagian data latih dan uji 3 memiliki akurasi terbaik dan akurasi akan semakin naik ketika data uji yang diberikan berkurang. Pada skenario 1 akurasi bernilai 58% kemudian naik menjadi 61.5% hingga bernilai 65% pada skenario 3. Pada skenario 3 dengan rasio pembagian data uji dan latih sebesar 90:10, dianggap paling optimal dalam membagi *dataset* pada *Machine Learning*. Hasil error pada model berbanding terbalik dengan akurasi untuk kasus data ini. Pada skenario 1 terjadi error sebesar 42% kemudian bertambah pada skenario 2 menjadi 38.5% dan skenario 3 memiliki error paling kecil dengan nilai 35%.

#### 4.2.2 Presisi pada Model *K-Nearest Neighbors*

Pada simulasi presisi dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.5.

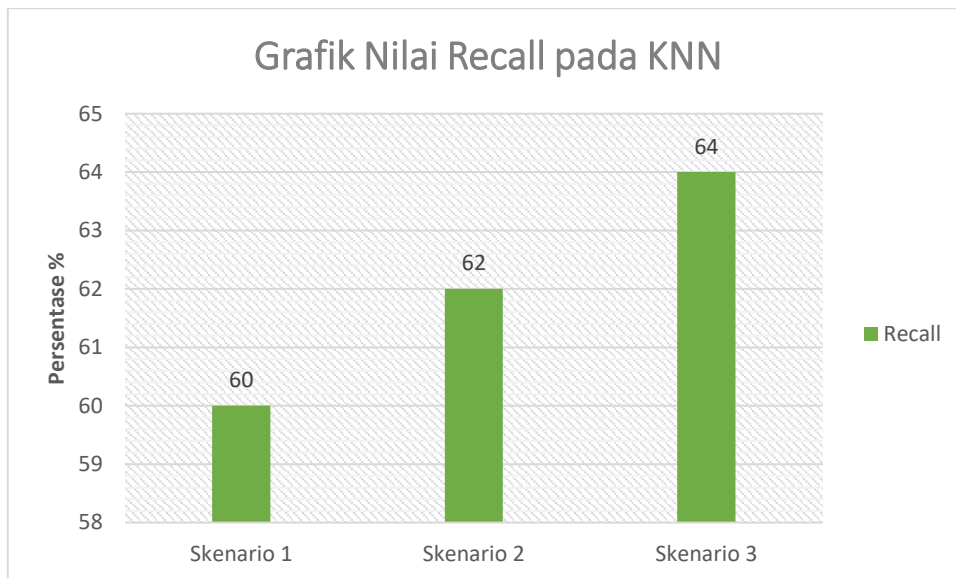


Gambar 4.6 Hasil Grafik Nilai Presisi Pada Model *K-Nearest Neighbors*

Dapat dilihat dari hasil pada Gambar 4.6 bahwa nilai presisi pada skenario 3 yang tertinggi yakni 62%, kemudian skenario 1 memiliki nilai 60% dan naik pada skenario 2 dengan nilai 61%. Dari hasil ini menunjukkan bahwa pada skenario 1 dengan rasio pembagian data uji dan latih sebesar 75:25, prediksi banyak menghasilkan *false positive* sehingga membuat nilai presisi menurun.

#### 4.2.3 Recall pada Model *K-Nearest Neighbors*

Pada simulasi *recall* dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *recall* dapat dilihat pada Gambar 4.5.

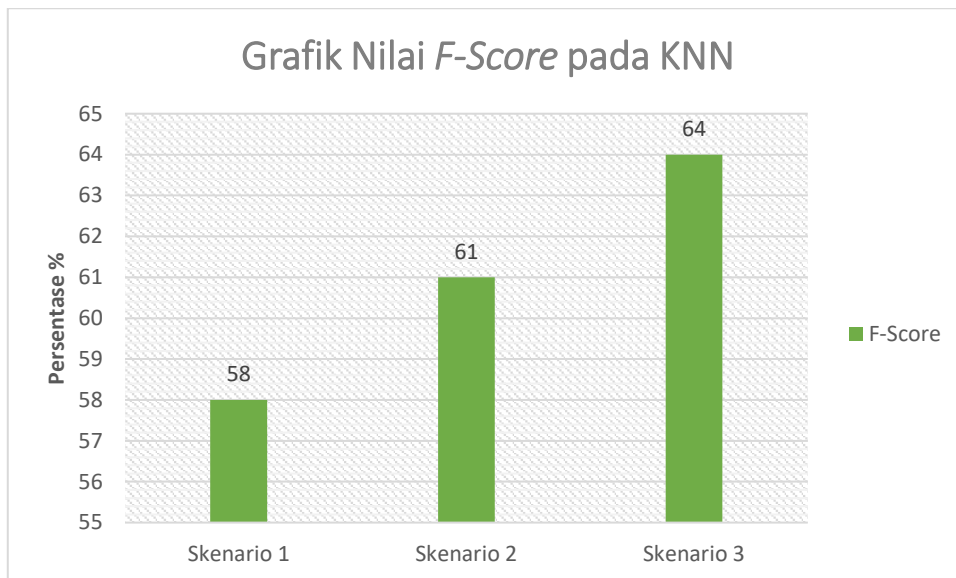


Gambar 4.7 Hasil Grafik Nilai *Recall* Pada Model *K-Nearest Neighbors*

Dapat dilihat dari hasil pada Gambar 4.7 bahwa nilai presisi pada skenario 3 yang tertinggi yakni 64%, kemudian skenario 1 memiliki nilai terendah yakni 60% dan naik pada skenario 2 dengan nilai 62%. Sama halnya dengan presisi, hasil ini menunjukkan model skenario 1 tidak dilatih dengan baik, karena banyak menghasilkan prediksi yang *false positive* dan *false negative* sehingga nilai *recall* sedikit dari presisi.

#### 4.2.4 *F-Score* pada Model *K-Nearest Neighbors*

Pada simulasi *f-score* dilakukan dengan menggunakan tiga skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisis dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.5.

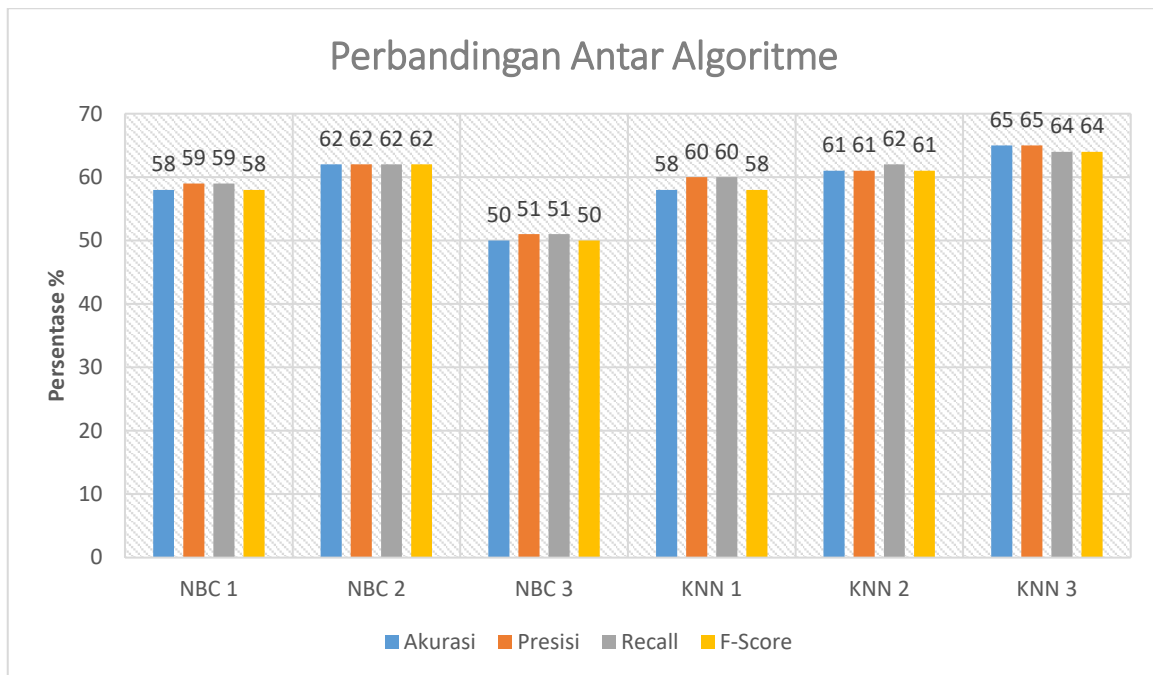


Gambar 4.8 Hasil Grafik Nilai *F-Score* Pada Model *K-Nearest Neighbors*

Dapat dilihat dari hasil pada Gambar 4.8 bahwa nilai *f-score* meningkat seiring ditambahkan data latih, pada skenario 1 memiliki nilai 58%, diikuti dengan skenario 2 bernilai 61%, dan skenario 3 memiliki nilai *f-score* terbaik yakni sebesar 64%. Hasil dari *f-score* tidak berbeda jauh dari presisi dan *recall* karena parameter ini merupakan rata-rata harmonik dari keduanya.

#### 4.3 Perbandingan Kinerja Antar Algoritme

Perbandingan hasil model klasifikasi dari setiap skenario pengujian yang telah dilatih menggunakan algoritme *Naïve Bayes Classifier* dan *K-Nearest Neighbors* dilakukan untuk melihat kinerja dari masing-masing model. Untuk menganalisis masing-masing model, maka digunakan bantuan dari *Confusion Matrix* dengan parameter berupa akurasi, presisi, *recall*, dan *f-score*. Hasil simulasi dari parameter tersebut dapat dilihat pada Gambar 4.9.



Gambar 4.9 Hasil Grafik Nilai Pada Masing-Masing Parameter Pengujian Model

*Dataset* yang digunakan pada penelitian ini memiliki 192 sampel dan banyaknya data sampel yang diberikan akan mempengaruhi hasil dari pengujian. Penelitian ini bertujuan untuk mengetahui hasil dari berbagai skenario pengujian dengan berbagai macam rasio pembagian data latih dan data uji serta pengaruhnya terhadap *dataset* yang sudah dikumpulkan. Didapati bahwa ketika data uji semakin sedikit, maka model akan menghasilkan persentase yang tinggi dari setiap parameter. Hal ini disebabkan karena pengaruh dari prediksi model yang salah terhadap data uji baik itu *false positive* maupun *false negative* akan sangat mempengaruhi dari persentase parameter. Untuk pelatihan model skenario pengujian terbaik algoritme *Naive Bayes Classifier* terjadi pada skenario 2 (rasio 80:20) di mana skenario ini memiliki nilai yang cukup tinggi dan stabil pada setiap parameter. Namun hasil ini memiliki akurasi yang lebih rendah dari penelitian milik Fredy, Tarno, dan Di Asih [7] dengan akurasi: 76.92%, *Naive Bayes Classifier* memiliki kelebihan dalam pengujian menggunakan *dataset* yang besar. Karena itu, dengan memiliki lebih banyak data yang berkualitas akan membantu membangun model klasifikasi (*Classifier*) menjadi lebih baik. Pada pengujian dengan menggunakan algoritme *K-Nearest Neighbors* terlihat bahwa skenario 3 (rasio 90:10) memiliki nilai yang tertinggi dan stabil.

Hasil pengujian menunjukkan bahwa untuk kasus data penelitian ini, secara keseluruhan model yang dilatih dengan menggunakan algoritme *K-Nearest Neighbors* lebih baik dari pada *Naive Bayes Classifier* pada masing-masing skenario pengujian. Perlu dilakukan optimasi tambahan pada model yang dihasilkan oleh *Naive Bayes Classifier* seperti menambahkan fitur baru dan sampel data.

#### 4.4 Pengujian Dengan *K-Fold Cross Validation*

Setelah melakukan *re-sampling* dengan menggunakan *K-Fold Cross Validation*, maka didapati hasil pengujian dari setiap algoritme seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Akurasi Data Latih

Skenario Pengujian	Akurasi Data Latih	Akurasi <i>K-Fold Cross Validation</i>
NBC 1	55.5%	53%
NBC 2	55%	57%
NBC 3	57.5%	57%
KNN 1	59%	58%
KNN 2	58%	55%
KNN 3	59%	58%

Dari hasil ini, bisa dikatakan bahwa terjadi *Underfitting* pada kedua model NBC dan KNN yang telah dilatih sebelumnya ketika dilakukan pembagian data latih dan uji secara acak (*random*) menggunakan bantuan *scikit-learn library*. Model yang dihasilkan terlalu umum atau *general* untuk data tertentu sehingga tidak merepresentasikan data secara keseluruhan dan menghasilkan akurasi yang tidak terlalu tinggi, yang mana jika diberikan *input* data baru maka kinerja model akan turun. Salah satu cara yang umum digunakan untuk mengatasi permasalahan ini adalah dengan melakukan *re-sampling*. Hasil pengujian menunjukkan bahwa akurasi yang dihasilkan setelah melakukan *re-sampling* lebih rendah dibandingkan dengan pengujian yang membagi data latih dan uji secara acak.



Tabel 4.2 Akurasi *K-Fold Cross Validation*

Iterasi	Akurasi <i>K-Fold Cross Validation</i>					
	NBC 1	NBC 2	NBC 3	KNN 1	KNN 2	KNN 3
1	40%	50%	72%	60%	38%	56%
2	60%	44%	44%	60%	31%	39%
3	40%	75%	52%	73%	75%	71%
4	53%	40%	64%	60%	67%	71%
5	50%	53%	53%	43%	53%	65%
6	57%	53%	53%	57%	53%	53%
7	57%	66%	59%	57%	67%	53%
8	57%	66%	64%	50%	67%	65%
9	57%	66%	47%	50%	60%	50%
10	64%	60%	64%	71%	60%	64%
Rata-Rata	53%	57%	57%	58%	55%	58%

Pada hasil pengujian, didapati pada Tabel 4.2 bahwa skenario 1 dan 3 pada model *K-Nearest Neighbors* memiliki akurasi tertinggi di antara skenario pengujian yang dilakukan dengan nilai akurasi sebesar 58% dan skenario 1 pada *Naïve Bayes Classifier* merupakan model dengan akurasi terendah dengan nilai akurasi sebesar 53%. Dari hasil ini juga didapati bahwa akurasi dari model *Naïve Bayes Classifier* akan meningkat ketika diberikan data latih yang lebih banyak.

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Beberapa hal yang bisa disimpulkan oleh penelitian yang telah dilakukan dan dianalisis oleh penulis, yakni:

1. Kenaikan akurasi kinerja model terjadi ketika pembagian data latih dan data uji mengikuti skenario pengujian 2 (rasio 80:20) untuk NBC dengan tingkat akurasi yang didapat 62%.
2. Kenaikan akurasi kinerja model terjadi ketika pembagian data latih dan data uji mengikuti skenario pengujian 3 (rasio 90:10) untuk KNN dengan tingkat akurasi yang didapat 65%.
3. Pada penelitian ini, dapat disimpulkan bahwa model yang dilatih dengan menggunakan algoritme *K-Nearest Neighbors* memiliki kinerja yang lebih bagus dibanding dengan algoritme *Naïve Bayes Classifier*.
4. Algoritme *K-Nearest Neighbors* dan *Naïve Bayes Classifier* dapat digunakan untuk memprediksi pergerakan nilai mata uang USD terhadap harga emas dan pemilihan fitur atau atribut yang sesuai sangat penting dalam melakukan prediksi.

#### 5.2 Saran

Ada beberapa saran yang bisa diberikan oleh penulis untuk penelitian selanjutnya agar bisa mendapatkan hasil yang lebih baik, yakni:

1. Menambahkan beberapa fitur dan sampel data baru yang lebih lengkap untuk hasil prediksi yang lebih baik.
2. Menambahkan model *Deployment* atau membangun *website* agar hasil prediksi dapat diakses secara *online*.

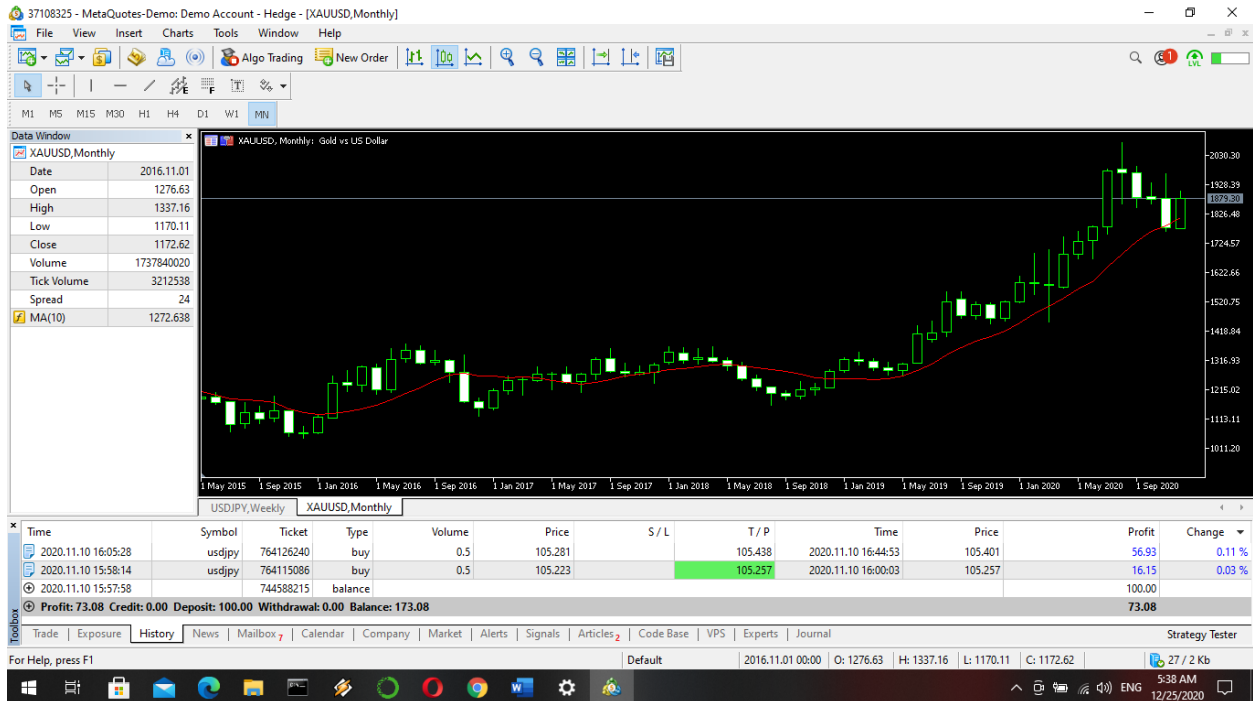
## DAFTAR PUSTAKA

- [1] Y. Mahena, M. Rusli, and E. Winarso, “Prediksi Harga Emas Dunia Sebagai Pendukung Keputusan Investasi Saham Emas Menggunakan Teknik Data Mining,” vol. 2, no. 1, 2015.
- [2] A. Ahmad, “Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning,” *J. Teknol. Indones.*, no. October, p. 3, 2017.
- [3] J. Politeknik and C. Riau, “Prediksi Ketepatan Waktu Lulus Mahasiswa dengan k- Nearest Neighbor dan Naïve Bayes Classifier ( Studi Kasus Prodi D3 Sistem Informasi Universitas Airlangga ),” vol. 3, no. 2, pp. 233–240, 2017.
- [4] Y. M. Pranoto and R. A. Harianto, “Penerapan Algoritma Klasifikasi Untuk Sistem Rekomendasi Buy Sell Di Forex Trading,” vol. 10, no. 2, pp. 152–158, 2020.
- [5] I. Abbas, “Penerapan Algoritma Support Vector Machine ( SVM ) Untuk Memprediksi Dan Membandingkan Hasil Trend Kurva Pada Trading Forex,” vol. 2, no. Sens 2, pp. 25–33, 2016.
- [6] N. D. S, “Penerapan Algoritma Support Vector Machine untuk Prediksi Harga Emas,” pp. 10–19.
- [7] F. Y. Marianto and D. A. I. Maruddani, “Perbandingan Metode Naïve Bayes Dan Bayesian Regularization Neural Network (BRNN) Untuk Klasifikasi Sinyal Palsu Pada Indikator Stochastic Oscillator ( Studi Kasus: Saham PT Bank Rakyat Indonesia (Persero) Tbk Periode Januari 2017 – Agustus 2019 ),” vol. 9, pp. 16–25, 2020.
- [8] T. R. Patil and M. . Sherekar, “Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification,” *Int. J. Comput. Sci. Appl.*, vol. 6, no. 2, pp. 256–261, 2013.
- [9] Bustami, “Penerapan Algoritma Naive Bayes untuk Mengklasifikasi Data Nasabah,” *TECHSI J. Penelit. Tek. Inform.*, vol. 4, pp. 127–146, 2010.
- [10] S. A. Pattekari and A. Parveen, “Prediction system for heart disease using Naive Bayes,” *Int. J. Adv. Comput. Math. Sci.*, vol. 3, no. 3, pp. 2230–9624, 2012.
- [11] U. Hasanuddin, “Perbandingan Algoritma Naïve Bayes Dan K- Nearest Neighbor Untuk Sistem Kelayakan Kredit Pada Nasabah ( Studi Kasus : PT . Armada Finance Cabang Makassar ),” 2017.
- [12] S. Guido, *Introduction to Machine Learning with Python*. .
- [13] N. S. B. Kusrorong, D. R. Sina, N. D. Rumlaklak, J. I. Komputer, and U. N. Cendana, “Kajian Machine Learning Dengan Komparasi Klasifikasi Prediksi Dataset Tenaga Kerja Non-Aktif,” vol. 7, no. 1, pp. 37–49, 2019.

# LAMPIRAN

## Lampiran 1 – Metatrader

Semua data yang digunakan diperoleh dari aplikasi *MetaTrader*:



## Lampiran 2 – Dataset

Open	High	Low	Close	Goal
437.1	437.3	416.5	422.1	0
422.1	437.7	409.8	435	1
435	446.6	423.1	427.8	0
427.7	437.2	421.7	434	1
433.8	434.1	413.3	416.8	0
416.6	443.2	414.2	434.5	1
434.2	436.2	417.8	429.3	0
429.2	449.1	428.6	434.6	0
434.8	475	434.3	468.8	1
467.3	480	459.5	465.2	0
465.6	502.2	454.8	498.5	0
491.5	540.5	489.3	516.5	0
516.8	572.2	516.3	568.7	1
569.2	574.7	534.6	562	0
562.1	589.1	533.7	583.5	0
572.2	655.5	572.2	653.7	1
654.1	729.7	635.6	642.7	1
643.1	645.2	542.5	615.2	0
617	676	602.1	635.7	1
635.2	655.5	607	624.6	1
626	640.2	570.2	597.7	0
598.2	610.6	559.2	606.1	0
606	648.2	605.1	647.2	1
646.1	649.6	611.1	636.1	0
638.2	654.2	601.5	652.7	0
652.5	688.1	642.7	669.1	1
673.2	677	632.5	663.2	0
664.5	693.3	655.9	677.7	1
677.9	690.4	651.5	660.5	1
661.2	673.6	639.8	649.1	0
650.1	687.5	645	663.5	0
663.4	676.4	641.6	672.8	0
672.5	745.4	671	742.9	0
743.7	797	720.3	796.1	1
797.9	845.4	772.5	782.7	0
784.6	841.8	776.8	831.5	0
841	936.7	839.5	924.5	0
923.9	975.8	883.9	973.5	1
976.1	1032.2	904.4	915.8	1
915.4	952.4	862.2	876.7	0
875.3	934.9	844.9	886.3	0
886.8	934.9	857.1	924.6	1
925.4	988	893.6	913.3	1
913.7	916.7	772.6	830.7	0

833.8	923.9	735.9	870.9	1
872.4	931.1	681.4	723.7	0
728.2	830.5	699.7	816	0
815.1	887.8	741.1	865.9	0
874.2	929.2	801.9	926.5	0
923.7	1005.9	888.9	941.6	1
947.5	966.7	883	917.4	1
916.9	932.6	864.54	886.2	0
885.9	980.02	880.03	978.57	1
976.74	989.63	912.88	926.35	0
927.49	958.68	904.78	953.5	1
952.97	971.46	930.08	950.65	0
952.34	1023.87	946.23	1007.3	0
1009.4	1070.58	986.55	1044.96	0
1044.12	1194.95	1042.84	1179.32	1
1179.66	1226.37	1074.48	1098.3	1
1094.61	1161.63	1074.01	1080.67	0
1082.66	1130.75	1044.46	1117.3	1
1117.83	1144.62	1085.05	1112.89	0
1112.98	1181.41	1111.45	1178.72	0
1177.78	1249.12	1157.62	1216	0
1216.65	1265.02	1196.8	1242.13	1
1242.53	1243.73	1157	1180.55	0
1181.13	1249.7	1174.53	1247.2	0
1247.4	1315.59	1236.5	1307.94	0
1307.35	1387.03	1306.72	1359	0
1361.2	1424.3	1325.55	1386.4	0
1383.2	1430.88	1361.19	1418.19	1
1415.71	1423.77	1307.88	1332.4	0
1332.31	1417.94	1324.82	1411.12	0
1410.87	1447.4	1380.88	1431.9	0
1432.48	1569	1412.74	1563.3	1
1547.25	1575.13	1462.35	1535.6	1
1535.05	1557.86	1490.96	1500	0
1500.75	1632.35	1478.05	1626.27	0
1610.2	1911.63	1607.88	1824.56	1
1824.75	1920.61	1531.94	1623.68	0
1623.68	1752.22	1595.65	1713.95	0
1710.93	1802.79	1666.33	1746.03	1
1747.83	1763.03	1522.64	1563.95	0
1567.37	1747.52	1566.07	1738.18	1
1738.07	1790.56	1685.96	1696.35	1
1716.46	1724.1	1628.08	1666.95	1
1671.34	1683.46	1612.39	1664.42	1
1664.55	1671.45	1526.96	1560.9	0
1559.49	1640.63	1544.92	1596.56	0

1597.61	1629.38	1554.94	1613.98	0
1614.74	1692.78	1583.83	1690.8	0
1689.15	1787.39	1685.35	1771.68	1
1769.47	1795.87	1698.5	1720.32	1
1719.89	1754.28	1672.63	1714.13	1
1714.96	1723.27	1635.44	1676.12	1
1673.25	1696.1	1625.8	1663.44	1
1663.93	1684.97	1555.04	1579.99	0
1580.61	1616.84	1561.41	1596.1	1
1600.84	1603.92	1321.54	1476.59	1
1475.67	1488.08	1337.95	1387.64	1
1389.5	1423.89	1180.17	1234.86	0
1235.51	1346.68	1208.14	1325.06	0
1324.21	1433.73	1272.8	1394.81	1
1391.41	1416.35	1291.79	1328.65	1
1327.47	1361.77	1251.67	1322.87	1
1323.33	1327.45	1226.77	1252.92	1
1249.5	1267.76	1182.42	1206.79	0
1225.12	1278.93	1213.57	1243.85	0
1243.65	1345.32	1240.76	1326.21	1
1335.44	1388.91	1282.38	1283.82	0
1284.67	1331.04	1268.41	1291.08	1
1291.3	1315.55	1242.15	1249.23	0
1249.56	1329.34	1240.56	1327.17	1
1326.64	1345.05	1280.48	1282.96	0
1283.12	1322.53	1272.61	1286.66	1
1287.47	1289.71	1204.31	1207.83	1
1208.21	1255.13	1161.42	1171.52	1
1167.31	1207.63	1131.44	1165.88	0
1158.49	1238.25	1141.74	1188.37	0
1186.94	1307.53	1167.33	1283.56	1
1280.42	1285.55	1190.3	1212.78	1
1213.85	1223.18	1142.7	1183.57	0
1183.76	1224.13	1175.14	1183.97	0
1184.55	1232.22	1169.66	1190.42	1
1190.47	1205.68	1162.63	1171.83	1
1174.14	1175	1069.72	1095.48	0
1095.72	1170.01	1080.94	1134.71	1
1134.6	1156.65	1098.71	1114.47	0
1115.56	1191.44	1104.59	1141.74	1
1141.45	1142.94	1052.67	1064.37	1
1064.81	1088.78	1046.23	1060.91	0
1064.88	1127.94	1061.66	1117.99	0
1115.86	1263.34	1115.86	1238.3	1
1239.05	1283.36	1208.15	1232.44	0
1231.56	1296.62	1208.93	1293.68	1

1292.65	1303.65	1199.62	1215.08	0
1216.02	1358.25	1205.91	1321.58	0
1322.58	1375.05	1310.77	1351.11	1
1350.61	1367.21	1304.16	1308.81	0
1309.79	1352.51	1302.41	1315.89	1
1317.19	1319.52	1241.3	1276.87	1
1276.63	1337.16	1170.11	1172.62	1
1173.73	1187.93	1122.62	1152.12	0
1151.67	1219.4	1145.95	1210.36	0
1210.48	1263.79	1198.1	1248	0
1248.5	1260.97	1194.91	1248.96	0
1247.94	1295.45	1243.65	1267.99	0
1268.56	1274.08	1214.23	1268.7	1
1268.67	1296.01	1236.01	1241.43	0
1242.12	1271.04	1204.81	1269.37	0
1269.47	1325.94	1251.44	1321.41	1
1321.85	1357.42	1275.92	1279.37	1
1278.95	1305.91	1260.54	1271	0
1270.32	1299.08	1265.37	1274.7	0
1275.02	1307.41	1236.37	1302.54	0
1309.81	1365.92	1305.7	1344.94	1
1343.39	1361.69	1307.04	1318.08	0
1319.02	1356.84	1302.76	1325.04	1
1325.8	1365.2	1310.18	1314.98	1
1316.11	1325.9	1282	1298.24	1
1298.89	1309.28	1245.84	1252.74	1
1252.73	1265.84	1211.45	1223.94	1
1224.13	1224.76	1160.07	1200.78	1
1200.68	1212.6	1180.66	1191.51	0
1191.89	1243.36	1183.16	1214.48	0
1215.3	1237.33	1196.21	1221.96	0
1222.91	1284.02	1221.25	1280.23	0
1282.21	1326.25	1276.54	1321.05	1
1320.95	1346.66	1302.34	1313.08	1
1314.15	1324.49	1280.77	1292.16	1
1292.15	1310.53	1266.21	1283.29	0
1283.13	1306.92	1266.2	1305.13	0
1306.59	1439.21	1306.49	1409.91	0
1391.67	1452.92	1381.66	1412.23	0
1415.85	1555.21	1400.44	1520.55	1
1529.39	1556.98	1464.43	1472.29	0
1472.93	1519.48	1459.03	1512.13	1
1512.49	1516.09	1445.57	1463.83	0
1462.43	1525.23	1453.89	1520.21	0
1520.26	1611.35	1518.38	1588.83	1
1588.65	1689.25	1547.38	1584.7	1



1581.79	1703.44	1451.13	1576.78	0
1571.41	1747.4	1569.33	1685.98	0
1686.02	1765.08	1670.4	1730.41	0
1735.4	1785.94	1670.55	1780.54	0
1781.61	1983.97	1757.43	1975.02	1
1983.53	2074.87	1862.58	1968.22	1
1968.35	1992.29	1848.75	1885.23	1
1887.32	1933.18	1859.92	1878.37	1
1879.27	1965.46	1764.38	1777.64	0
1776.97	1875.28	1775.74	1837.67	1



### Lampiran 3 – Program Coding Naïve Bayes Classifier

```
# importing libraries
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('C:/Users/ASUS/Downloads/Skripsi/ValueGoldtoUSDWeekly2005-2020.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print (accuracies.mean()*100)
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
print ("Train Score =", classifier.score(X_train, y_train))
from sklearn.metrics import classification_report
cr = classification_report(y_test, y_pred)
print(cr)
acc = accuracy_score(y_test, y_pred)
print (acc)
```

## Lampiran 4 – Program Coding K-Nearest Neighbors

```
# Importing the libraries
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('C:/Users/ASUS/Downloads/Skripsi/ValueGoldtoUSDMonthlydataset.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier

grid = {'n_neighbors':[1,10], 'p':[1,2,3], 'weights':['uniform','distance']}
knn = KNeighborsClassifier(algorithm = "auto")
knn_cv = GridSearchCV(knn,grid,cv=10)
knn_cv.fit(X_train,y_train)
y_pred = knn_cv.predict(X_test)

from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = knn_cv, X = X_train, y = y_train, cv = 10)
print (accuracies.mean()*100)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print (cm)
print ("Hyperparameters:",knn_cv.best_params_)
print ("Train Score:",knn_cv.best_score_)
print ("Test Score:",knn_cv.score(X_test,y_test))

#Classification Report
from sklearn.metrics import classification_report, accuracy_score
cr = classification_report(y_test, y_pred)
print (cr)
acc = accuracy_score(y_test, y_pred)
print (acc)
```

### Lampiran 5 – Confusion Matrix

$$\text{NBC 1} = \begin{bmatrix} 14 & 13 \\ 7 & 14 \end{bmatrix}$$

$$\text{NBC 2} = \begin{bmatrix} 12 & 9 \\ 6 & 12 \end{bmatrix}$$

$$\text{NBC 3} = \begin{bmatrix} 5 & 6 \\ 4 & 5 \end{bmatrix}$$

$$\text{KNN 1} = \begin{bmatrix} 13 & 14 \\ 6 & 15 \end{bmatrix}$$

$$\text{KNN 2} = \begin{bmatrix} 13 & 8 \\ 7 & 11 \end{bmatrix}$$

$$\text{KNN 3} = \begin{bmatrix} 8 & 3 \\ 4 & 5 \end{bmatrix}$$

