



# Perbandingan Algoritma KNN dan SVM Dalam Memprediksi Metaverse Crypto Token

Kurnia Ramadhan Putra<sup>a</sup>, Bahy Tsany Rizqullah<sup>b</sup>

<sup>a,b</sup> Program Studi Sistem Informasi, Institut Teknologi Nasional, Bandung, Indonesia

email: <sup>a\*</sup>[kurniaramadhan@itenas.ac.id](mailto:kurniaramadhan@itenas.ac.id)

\*Korespondensi

Dikirim 01 November 2024; Direvisi 15 April 2025; Diterima 25 April 2025; Diterbitkan 17 Juni 2025

## Abstrak

Perubahan harga cryptocurrency cenderung fluktuatif, sehingga diperlukan teknik khusus untuk memprediksi harga cryptocurrency tersebut. Teknik prediksi data mining dengan metode regresi digunakan untuk membangun model prediksi harga cryptocurrency tersebut. Algoritma yang digunakan untuk pemodelan yaitu K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM), kemudian hasil dari kedua algoritma tersebut dibandingkan untuk mendapatkan pemodelan yang lebih baik. Beberapa variabel yang digunakan pada data yaitu variabel independen seperti open, high, dan low serta variabel dependen seperti close. Evaluasi terhadap pengujian model dilakukan menggunakan berbagai teknik seperti menggunakan data murni, remove outlier, pembagian data mulai dari 70:30, 80:20, dan 90:10, serta standarisasi. Dari hasil evaluasi didapatkan bahwa algoritma SVM dengan teknik standarisasi z-score, dan pembagian data 70:30 memberikan nilai error paling kecil yaitu 5% serta nilai R2 Square yaitu 0,98.

**Kata Kunci:** Cryptocurrency, Data Mining, KNN, SVM, R2 Square.

## Comparison of KNN and SVM Algorithms in Predicting Metaverse Crypto Tokens

## Abstract

Cryptocurrency price changes tend to fluctuate, so special techniques are needed to predict the cryptocurrency's price. The data mining prediction technique with the regression method is used to build a cryptocurrency price prediction model. The algorithms used for modeling are K-Nearest Neighbors (KNN) and Support Vector Machine (SVM), and then the results of the two algorithms are compared to obtain better modeling. Several variables used in the data are independent variables such as open, high, and low and dependent variables such as close. Model testing is evaluated using various techniques such as using pure data, removing outliers, data division starting from 70:30, 80:20, and 90:10, and standardization. From the evaluation results, it was obtained that the SVM algorithm with the z-score standardization technique, and data division 70:30 gave the smallest error value of 5% and the R2 Square value of 0.98.

**Keywords:** : Cryptocurrency, Data Mining, KNN, SVM, R2 Square

Untuk mengutip artikel ini dengan APA Style:

Putra, K. R., & Rizqullah, B. T. (2025). Perbandingan Algoritma KNN dan SVM Dalam Memprediksi Metaverse Crypto Token. TEKNOLOGI: Jurnal Ilmiah Sistem Informasi, 15(1), 44-56: <https://doi.org/10.26594/teknologi.v15i1.5066>



© 2022 Penulis. Diterbitkan oleh Program Studi Sistem Informasi, Universitas Pesantren Tinggi Darul Ulum. Ini adalah artikel open access di bawah lisensi CC BY-NC-NA (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

## 1. Pendahuluan

Revolusi Industri 4.0 membawa perkembangan terhadap teknologi dengan menghadirkan berbagai inovasi hampir di seluruh aspek kehidupan manusia termasuk dalam kegiatan ekonomi (Moch Farryz Rizkillah & Sri Widyanesti, 2022). Salah satu aspek inovasi dari perkembangan teknologi tersebut adalah munculnya mata uang elektronik yang dapat digunakan sebagai alat pembayaran yang sah, seperti Gopay, Shopee Pay, Dana, dan Ovo yang sudah digunakan secara umum oleh Masyarakat (Widiantoro et al., 2021; Widiantoro & Harnadi, 2022). Salah satu teknologi informasi yang banyak diminati pada saat ini adalah metaverse (Alfiras et al., 2023).

Metaverse adalah teknologi digital yang mampu menciptakan dunia virtual 3D berbasis *Augmented Reality* dan *Virtual Reality* (Endarto & Martadi, 2022). Metaverse merupakan salah satu contoh dari penerapan teknologi informasi yang memudahkan penggunaanya dalam mengakses informasi. Pada konsep baru *metaverse*, hubungan antara dunia virtual dan dunia nyata sudah bukan menjadi hambatan, dimana metaverse memungkinkan penggunanya untuk bekerja, berinteraksi, dan bersosialisasi melalui pengalaman

bersama yang dimulaskan (Dwivedi et al., 2022). Metaverse merupakan bagian dari internet yang di dalamnya dibuat sedemikian rupa menyerupai dunia nyata (Sopiandi & Susanti, 2022).

Pada metaverse terdapat mata uang elektronik yang disebut *Metaverse Crypto Tokens* yang memiliki banyak jenis mata uang dengan setiap harga yang berbeda. Dimana pada *cryptocurrency*, setiap data transaksi akan dikodekan menggunakan algoritma kriptografi tertentu (Kim et al., 2022; Parekh et al., 2022; Robiyanto et al., 2019). *Cryptocurrency* tidak berwujud dan tidak bisa diterbitkan oleh suatu negara atau bank sentral karena tidak berada dibawah kendali pemerintah. Namun beberapa negara maju seperti Jepang, Singapura, Amerika Serikat, dan Inggris telah melegalkan *Cryptocurrency* sebagai bentuk inovasi pembayaran di era modern (Farida & Khasanah, 2021). Penggunaan Metaverse Crypto Tokens tentunya diliirk oleh para pelaku bisnis ataupun investor yang siap menginvestasikan dana untuk meraih keuntungan. Selain dianggap mudah untuk dibawa, *cryptocurrency* sulit dipalsukan karena memiliki fitur keamanan (Vaulia et al., 2023). *Cryptocurrency* memiliki harga yang fluktuatif setiap detiknya, sehingga para pelaku bisnis atau investor mengalami kesulitan dalam menentukan jenis mata uang *cryptocurrency* yang akan diinvestasikan (Raharja, 2021). Perubahan harga pada mata uang *cryptocurrency* ini tentunya dapat diketahui dengan menggunakan metode forecasting atau prediksi. Untuk membantu para investor, data mining dapat digunakan untuk memprediksi nilai mata uang di masa depan (Pradito & Purnia, 2022).

Prediksi merupakan proses untuk mengolah data yang dimiliki guna menghasilkan informasi yang berguna di masa yang akan datang. Hasil dari prediksi tidak perlu memberikan jawaban pasti, melainkan berusaha untuk mencari jawaban sedekat mungkin (Kafil, 2019). Salah satu metode data mining yang dapat digunakan untuk menangani kasus prediksi adalah K-Nearest Neighbors (KNN) (Cholil et al., 2021). KNN merupakan metode supervised yang bertujuan untuk mendapatkan pola data baru dengan cara menghubungkan pola data sebelumnya dengan pola data baru guna mengklasifikasikan data ke dalam tipe kelas yang berbeda berdasarkan metode atribut yang ada (Ariyani et al., 2022). Metode KNN dipilih karena memiliki kelebihan yaitu efektif diterapkan pada data dengan jumlah besar terhadap suatu data latih yang noise dan memiliki range nilai paling jauh dibanding data lainnya tetapi dapat mengganggu struktur data yang ada (Sumarlinda & Lestari, 2022). Selain kelebihan yang dimiliki, pada KNN juga terdapat kelemahan yaitu kurang optimal dalam menentukan nilai  $k$  yang merupakan jumlah tetangga terdekat dan harus menentukan atribut yang akan dipilih atau seleksi fitur guna mendapat hasil terbaik (Fajri et al., 2020). KNN pada dasarnya digunakan untuk melakukan klasifikasi, tetapi dapat juga untuk melakukan regresi yang biasa disebut dengan KNN Regression, dimana hasil regresi didapatkan dari rata – rata nilai tetangga terdekat (Seruni et al., 2020). Regresi linier berganda adalah teknik untuk menentukan hubungan antara variabel prediktor atau antara variabel lain dengan variabel prediktor. Prediksi regresi linier berganda digunakan sebagai teknik untuk menguji bagaimana variabel berhubungan dengan proses prediksi data. Jadi metode regresi linier berganda adalah cara yang tepat untuk menyelesaikan masalah prediksi. Metode yang dikembangkan untuk menangani kasus regresi diantaranya: Linear/Nonlinear Regression, ANN, KNN, SVM, GA, dan Regression Tree (Maulud & Abdulazeez, 2020).

Salah satu penelitian yang membahas mengenai KNN dalam memprediksi harga sebelumnya telah dilakukan oleh Haerul Fatah dan Agus Surbakti dengan judul “Prediksi Harga Cryptocurrency Dengan Metode K-Nearest Neighbors” yang bertujuan untuk menganalisis metode machine learning yang paling akurat untuk memprediksi harga *cryptocurrency* dengan melihat harga dari 3 jenis *Cryptocurrency* yang paling diminati yaitu Bitcoin, Ethereum, dan Ripple dikarenakan harganya yang dapat berubah secara fluktuatif, sehingga diperlukan suatu prediksi untuk mengetahui harga *cryptocurrency* tersebut di masa mendatang (Altayef et al., 2022; Tarakci & Ozkan, 2021; Venkatesh et al., 2023). Metode K-Nearest Neighbors (KNN) mampu memprediksi harga *cryptocurrency* dengan baik, yang mana pemberian nilai parameter  $k=3$  yang menghasilkan nilai Mean Absolute Error (MAE) sebesar 0.0018 dan Root Mean Squared Error (RMSE) sebesar 0.0089 (Hodson, 2022).

Penelitian ini bertujuan untuk melakukan komparasi beberapa metode dan teknik yang digunakan dalam prediksi untuk mendapatkan metode terbaik dalam memprediksi harga Metaverse Crypto Tokens. Metode prediksi yang dibandingkan yaitu K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM) dengan menggunakan beberapa teknik pengujian seperti penggunaan data murni, menghapus data outlier, penggunaan teknik normalisasi seperti min-max dan z-score. Selanjutnya akan dilakukan evaluasi pada kedua metode tersebut untuk membandingkan kinerja yang paling baik untuk memprediksi harga Metaverse Crypto Tokens berdasarkan nilai RMSE, MSE, MAE, MAPE, dan R2 Score dari setiap metode yang digunakan.

## 2. State of the Art

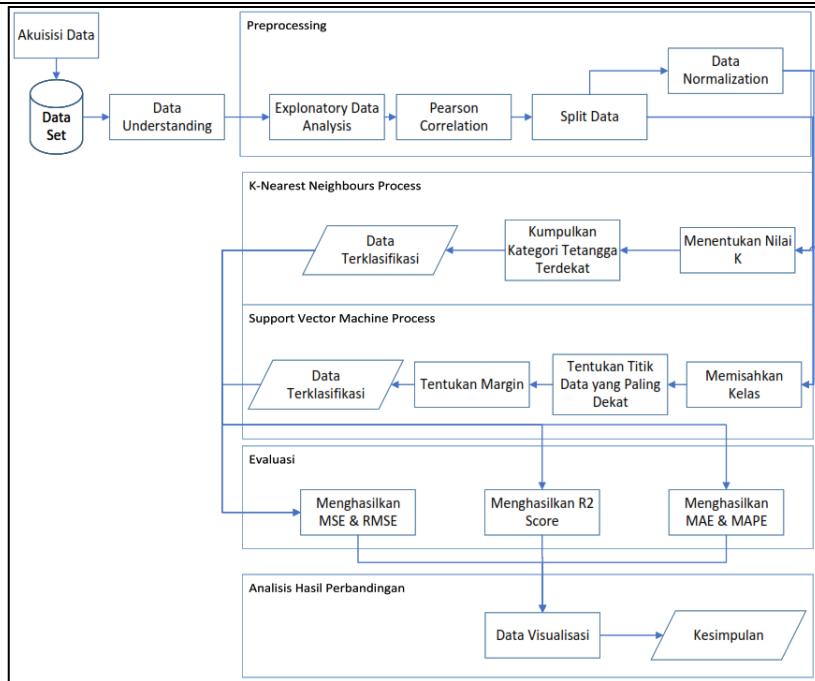
Penelitian sebelumnya yang memfokuskan pada prediksi harga *cryptocurrency* dengan menggunakan model hybrid yaitu *Gated Recurrent Unit* (GRU) dan *Long Short-Term Memory* (LSTM) untuk meningkatkan akurasi. Data dari empat *cryptocurrency* utama yaitu Bitcoin, Ethereum, Ripple, dan Binance digunakan untuk memprediksi harga. Model ini menunjukkan hasil yang sangat baik dengan *Root Mean Squared Error* (RMSE) mendekati nol setelah normalisasi dan nilai *Mean Absolute Percentage Error* (MAPE) di bawah 10% (Ferdiansyah et al., 2023). Penelitian lainnya yaitu pendekatan multivariat dengan tiga model Recurrent Neural Networks (RNN) yaitu LSTM, Bidirectional LSTM (Bi-LSTM), dan GRU untuk prediksi harga lima *cryptocurrency* utama (Bitcoin, Ethereum, Cardano, Tether, dan Binance Coin). Hasil eksperimen menunjukkan bahwa Bi-LSTM dan GRU memiliki akurasi serupa, sementara GRU menunjukkan hasil lebih baik dalam hal waktu eksekusi dan konsistensi hasil (Hansun et al., 2022). Selanjutnya penelitian lain yang berfokus pada prediksi harga Ethereum (ETH) menggunakan teknik Recurrent Neural Networks (RNN). Dengan mengandalkan analisis teknikal dan tren harga historis, model ini mencapai akurasi prediksi sebesar 95%. Studi ini menyimpulkan bahwa model RNN dapat memberikan hasil prediksi yang sangat akurat untuk harga *cryptocurrency* dan bermanfaat bagi peneliti di masa depan (Gunarto et al., 2023). Selanjutnya penelitian untuk memprediksi harga *cryptocurrency* menggunakan model hybrid GRU-LSTM dengan fokus pada optimisasi jumlah epoch untuk meningkatkan akurasi. Hasilnya berupa aplikasi web yang dapat memprediksi harga *cryptocurrency* untuk beberapa hari ke depan, berdasarkan data historis dari 7, 14, 30, 60, hingga 90 hari sebelumnya (Ardiyansyah et al., 2022). Terakhir, penelitian dengan mengembangkan model prediksi harga *cryptocurrency* dengan menggunakan LSTM yang dipadukan dengan *Change Point Detection* (CPD) menggunakan algoritma PELT. Teknik ini memungkinkan model untuk mendeteksi perubahan signifikan dalam harga dan menyesuaikan prediksi. Evaluasi terutama dilakukan pada Bitcoin, dan hasilnya menunjukkan bahwa model gabungan ini lebih akurat dibandingkan dengan LSTM standar, berdasarkan metrik MAE, MSE, dan RMSE (Akila et al., 2023). Sedangkan penelitian yang diusulkan menggunakan teknik regresi untuk memprediksi harga *cryptocurrency* dengan membandingkan dua algoritma *K-Nearest Neighbors* dan *Support Vector Machine*. Variabel independen yang digunakan termasuk harga open, high, dan low, sedangkan variabel dependen adalah harga close. Hasil evaluasi menunjukkan bahwa algoritma SVM dengan teknik standardisasi z-score dan pembagian data 70:30 memberikan nilai kesalahan terkecil sebesar 5% dan nilai R-squared ( $R^2$ ) sebesar 0.98. Penelitian yang diusulkan menggunakan metode klasik yaitu KNN dan SVM, namun menghasilkan akurasi tinggi dengan  $R^2$  sebesar 0.98, dibandingkan model *deep learning* yang cenderung lebih kompleks seperti penelitian sebelumnya. Penelitian yang diusulkan juga memperkenalkan beberapa teknik evaluasi, seperti pembersihan outlier, pembagian data yang bervariasi (70:30, 80:20, 90:10), dan standardisasi z-score. Hal ini memberikan pendekatan yang lebih mendetail dalam proses evaluasi model dibandingkan penelitian sebelumnya yang lebih fokus pada metrik akurasi dasar seperti RMSE dan MAPE dan tidak memfokuskan pada manipulasi dan pembersihan data.

## 3. Metode Penelitian

Pada penelitian ini, metode yang digunakan untuk memprediksi harga *Metaverse Crypto Tokens* yaitu *K-Nearest Neighbors* dan *Support Vector Machine*. Tujuan penggunaan kedua metode tersebut yaitu untuk mengetahui metode yang terbaik dalam melakukan prediksi harga, sehingga dapat membantu para pengusaha atau investor dalam memilih mata uang kripto yang akan diinvestasikan.

### 3.1. Akuisisi Data

Akuisisi Data merupakan pengambilan data dengan menggunakan metode sekunder dari repositori online yaitu Kaggle. Data yang digunakan pada penelitian ini adalah data histori mata uang *Metaverse Crypto Tokens* dapat diakses pada tautan <https://www.kaggle.com/datasets/kaushiksuresh147/metaverse-cryptos-historical-data>. Tahapan akuisisi data ini berguna untuk mengetahui jenis serta informasi dari data yang akan digunakan. Alur metode penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian

### 3.2. Data Understanding

Pada tahap data understanding dilakukan dengan Exploratory Data Analysis (EDA) yang bertujuan untuk memahami isi data yang akan digunakan. Proses EDA menggunakan bantuan perangkat lunak RapidMiner Studio, yang mana tahapannya secara umum dapat dilihat pada Gambar 2.



Gambar 2. Proses Data Understanding

Proses *data understanding* menggunakan *Exploratory Data Analysis* dijelaskan lebih detail sebagai berikut:

#### a. Import Dataset

Workspace RapidMiner yang terdiri dari beberapa panel, termasuk *Design View* dan *Results View*. *Design View* adalah area utama untuk merancang alur proses analisis data di RapidMiner. Beberapa operator yang dapat digunakan seperti Read CSV, Filter, Replace Missing Values, dan lain-lain, sehingga memungkinkan untuk membangun alur kerja analisis tanpa menulis kode. *Results View* adalah area untuk melihat hasil dari proses yang telah dijalankan di *Design View*, sehingga dapat dilihat hasil evaluasi data atau model yang telah diproses seperti output data yang telah diproses, hasil analisis statistik, atau kinerja model *machine learning*.

#### b. Memilih Kolom dan Baris Data

Ada tujuh kolom yang dipilih yaitu Date, Open, High, Low, Close, Volume, dan Currency dengan total 1.705 baris data.

#### c. Menentukan Tipe Data

Setelah memilih kolom dan baris data yang akan digunakan, selanjutnya memeriksa dan mengatur parameter terutama jika menggunakan file seperti CSV atau Excel, dengan mempertimbangkan beberapa hal sebagai berikut:

- *Delimiter* yaitu pemisah yang digunakan dalam file seperti koma, tab, atau pemisah lainnya.
- *Header* yang harus pastikan bahwa apakah file memiliki baris header atau tidak.
- *Data Types* yaitu tipe data dari masing-masing kolom seperti numerik, teks, atau nominal, namun dapat disesuaikan secara manual jika diperlukan.
- *Missing Values* yaitu nilai yang hilang dari baris data.

Untuk kolom Date dan Currency dengan tipe data polynominal, kolom Open, High, Low, dan Close dengan tipe data real atau decimal, dan kolom Volume dengan tipe data integer atau bilangan bulat.

d. Mengeksekusi Proses EDA

Setelah semua konfigurasi selesai selanjutnya menjalankan proses EDA. Hasilnya akan ditampilkan di panel Results View, di mana dapat dilakukan pemeriksaan terlebih dahulu apakah data telah dimuat dengan benar dan siap digunakan dalam analisis lebih lanjut. Hasil Proses EDA dapat dilihat pada Gambar 3.

Name	Type	Missing	Statistics	Filter (7 / 7 attributes):	Search for Attributes
Date	Polynomial	0	Least: 2022-07-10 (1), Most: 2017-11-09 (1)	Values: 2017-11-09 (1), 2017-11-10 (1), ... [1703 more]	
Open	Real	0	Min: 0.009	Max: 5.634	Average: 0.627
High	Real	0	Min: 0.015	Max: 1548.864	Average: 1.565
Low	Real	0	Min: 0.005	Max: 5.401	Average: 0.601
Close	Real	0	Min: 0.009	Max: 5.634	Average: 0.628
Volume	Integer	0	Min: 0	Max: 10115679	Average: 485016.516
Currency	Polynomial	0	Least: USD (1705)	Most: USD (1705)	Values: USD (1705)

Gambar 3. Proses Data Understanding

Dataset memiliki tujuh variabel yang terbagi ke dalam variabel independen dan variabel dependen. Variabel independen yaitu *Date*, *Open*, *High*, *Low*, *Volume*, dan *Currency*, sedangkan variabel dependen yaitu *Close* yang dijabarkan pada Tabel 1.

Tabel 1. Variabel pada Dataset

Atribut	Deskripsi	Jangkauan Nilai
<i>Date</i>	Tanggal Transaksi	2017-11-09 – 2022-07-10
<i>Open</i>	Harga Pembuka	0,164 – 75,779
<i>High</i>	Harga Tertinggi	0,265 – 87,163
<i>Low</i>	Harga Terendah	0,119 – 66,232
<i>Close</i>	Harga Akhir / Penutup	0,191 – 76,07
<i>Volume</i>	Jumlah Transaksi	0 - 104632000
<i>Currency</i>	Mata Uang	USD

### 3.3. Pra-pemrosesan Data

Pra-pemrosesan data adalah tahap awal dalam analisis data yang bertujuan untuk mempersiapkan dan membersihkan data sebelum dilakukan analisis lebih lanjut atau pemodelan. Dataset yang digunakan secara tidak langsung mungkin saja terdapat *noise*, *outliers*, atau *missing values* yang dapat memengaruhi hasil analisis. Oleh karena itu, pra-pemrosesan membantu meningkatkan kualitas data agar lebih sesuai dan akurat untuk diolah. Pada Gambar 4, variabel *Date* dikenali oleh sistem sebagai tipe data *object*, sehingga tidak dapat dilakukan proses *data mining*. Oleh karena itu, perlu dilakukan transformasi tipe data terlebih dahulu, untuk mengubah tipe data *object* menjadi *datetime*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1705 entries, 0 to 1704
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
 --- 
 0   Date      1705 non-null   datetime64[ns] 
 1   Open       1705 non-null   float64  
 2   High       1705 non-null   float64  
 3   Low        1705 non-null   float64  
 4   Close      1705 non-null   float64  
 5   Volume     1705 non-null   int64   
 6   Currency   1705 non-null   object  
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 93.4+ KB
None
```

Gambar 4. Transformasi Tipe Data

Beberapa teknik yang digunakan pada tahap pra-pemrosesan data yaitu menangani data hilang, menghapus *outlier*, uji korelasi, normalisasi dan standarisasi data, dan *data splitting*.

a. Menangani Data Hilang

Pada tahapan ini dilakukan pengecekan data pada dataset yang digunakan terdapat data yang hilang atau tidak. Pada Gambar 5 merupakan kode program untuk memeriksa data yang hilang menggunakan fungsi `df.isnull().sum()`. Dapat dilihat bahwa tidak ditemukan satupun adanya data yang hilang pada setiap variabel dataset.

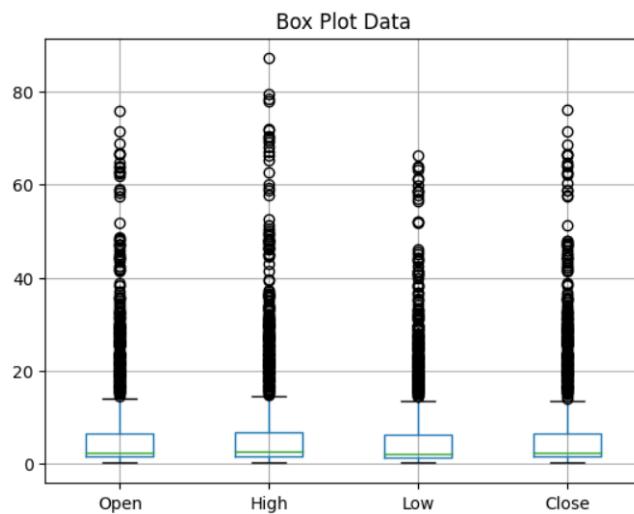
```
[ ] #null values
df.isnull().sum()

Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
Currency  0
dtype: int64
```

Gambar 5. Pengecekan Data Hilang pada Dataset

b. Menghapus *Outlier*

Pada tahapan selanjutnya dilakukan pengecekan *outlier* dari variabel yang ada pada dataset. *Outlier* adalah data yang berbeda dan memiliki rentang jauh dibandingkan sekumpulan mayoritas data lainnya, yang mana *outlier* yang ditemukan pada variabel divisualkan menggunakan *box plot* seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Pengecekan Outlier pada Dataset

Dapat dilihat dari keempat variabel tersebut memiliki *outlier* dengan rentang tertinggi berada pada variabel *High*. Sedangkan mayoritas harga dari masing – masing variabel tersebut memiliki rentang yang sama yaitu mulai dari 0 – 17 dollar. Pada Gambar 7 merupakan kode program untuk menghapus *outlier*, karena *outlier* yang tinggi dari variabel *Open*, *High*, *Low*, dan *Close* tentunya berpengaruh pada keakuratan hasil prediksi.

```

import pandas as pd

# Contoh dataframe
df = pd.read_csv('Fistiana.csv')

# Identifikasi outlier menggunakan IQR
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
outlier_mask = ((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)

# Hapus baris yang mengandung outlier
df_cleaned = df[~outlier_mask]

# Cetak dataframe hasil
print(df_cleaned)

```

Gambar 7. Kode Program untuk Penghapusan Outlier

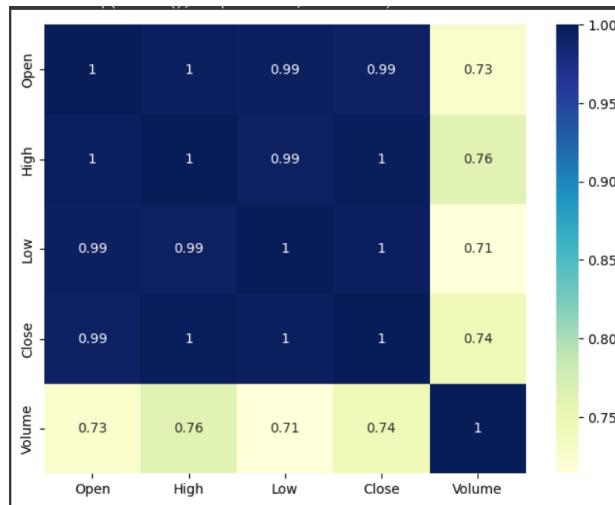
c. Uji Korelasi

Uji korelasi digunakan untuk melihat sejauh mana hubungan linear antar variabel menggunakan *Pearson Correlation* yang dapat dihitung menggunakan Persamaan 1.

$$r_{x,y} = \frac{n \sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{((n \sum x_i^2 - (\sum x_i)^2)((n \sum y_i^2 - (\sum y_i)^2))}} \quad (1)$$

Dimana,  $r_{x,y}$  adalah korelasi antar variabel  $x$  dengan variabel  $y$ ,  $n$  adalah banyaknya sampel data,  $x_i$  adalah nilai variabel  $x$  ke- $i$ , dan  $y_i$  adalah nilai variabel  $y$  ke- $i$ .

Nilai dari *Pearson Correlation* berada pada rentang -1 hingga 1, dimana nilai -1 menunjukkan korelasi yang negatif, nilai 0 menunjukkan tidak adanya korelasi, dan nilai 1 menunjukkan korelasi positif sempurna. Gambar 8 merupakan hasil visualisasi korelasi antara variabel yang ada pada dataset.



Gambar 8. Korelasi Antar Variabel Menggunakan Perason Correlation

Dapat dilihat bahwa variabel *Open*, *High*, *Low*, dan *Close* memiliki korelasi yang tinggi dengan nilai yang hampir sempurna yaitu mendekati 1, sedangkan untuk variabel *Volume* memiliki nilai yang cenderung lebih kecil terhadap variabel lainnya.

d. Normalisasi Data

Normalisasi *min-max* digunakan untuk mengubah skala data agar nilai-nilainya berada dalam rentang tertentu, biasanya antara 0 dan 1, yang dapat dihitung menggunakan Persamaan 2.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

Dimana,  $x'$  adalah nilai dari variabel  $x$  yang baru setelah dinormalisasi,  $x$  adalah nilai dari variabel  $x$  yang lama,  $x_{min}$  adalah nilai absolut minimal dari variabel  $x$ , dan  $x_{max}$  adalah absolut maksimal dari variabel  $x$ .

Proses normalisasi dilakukan pada variabel Date, Open, High, Low, Close, dan Volume, sedangkan variabel Currency tetap dengan nilai "USD" karena tidak bertipe data numerik.

e. Standarisasi Data

Standarisasi data menggunakan z-score digunakan untuk mengubah data sehingga memiliki distribusi dengan rata-rata 0 dan standar deviasi 1, yang dapat dihitung menggunakan Persamaan 3.

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

Dimana,  $z$  adalah nilai yang telah distandarisasi,  $x$  adalah nilai asli dari variabel  $x$ ,  $\mu$  adalah nilai rata-rata, dan  $\sigma$  adalah standar deviasi.

f. *Data Splitting*

Data splitting digunakan untuk membagi dataset menjadi beberapa subset, yang biasanya mencakup:

- *Training set* yaitu data yang digunakan untuk melatih model.
- *Test set* yaitu data yang digunakan untuk mengevaluasi performa akhir model, yang mana data tersebut tidak digunakan oleh model selama proses pelatihan, sehingga dapat memberikan gambaran yang lebih objektif tentang seberapa baik model dapat bekerja pada data yang belum pernah dilihat sebelumnya.

Total seluruh data pada dataset sebanyak 1.705 baris yang mana pembagian data set nya dapat dilihat pada Tabel 2.

Tabel 2. Data Splitting pada Dataset

Teknik Data Splitting		Jumlah Data
90:10	<i>Training Set</i>	1.534
	<i>Test Set</i>	171
80:20	<i>Training Set</i>	1.364
	<i>Test Set</i>	341
70:30	<i>Training Set</i>	1.193
	<i>Test Set</i>	512

### 3.4. Pemodelan Menggunakan KNN

KNN bekerja dengan cara mencari sejumlah k tetangga terdekat dari data baru dan menentukan hasil berdasarkan mayoritas (untuk klasifikasi) atau rata-rata (untuk regresi) dari tetangga-tetangga tersebut. Berikut adalah langkah-langkah dasar untuk KNN:

- Menghitung jarak antara data yang ingin diklasifikasikan dengan setiap titik data dalam dataset training.
- Mengurutkan tetangga berdasarkan jarak terdekat.
- Memilih k tetangga terdekat.
- mengambil keputusan dengan mengambil kelas yang paling sering muncul di antara tetangga.

Untuk menghitung jarak pada KNN menggunakan *Euclidean Distance*, yang ditunjukkan pada Persamaan 3.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (\mathbf{p}_i - \mathbf{q}_i)^2} \quad (3)$$

Dimana,  $d(\mathbf{p}, \mathbf{q})$  adalah jarak antara kedua titik  $\mathbf{p}$  dan  $\mathbf{q}$ ,  $n$  adalah jumlah variabel,  $\mathbf{p}_i$  dan  $\mathbf{q}_i$  adalah nilai atrbut ke-i dari titik p dan q.

### 3.5. Pemodelan Menggunakan SVM

SVM bekerja dengan cara menemukan hyperplane yang paling baik memisahkan dua kelas dalam data. SVM mencoba untuk memaksimalkan margin antara hyperplane dan titik data terdekat dari kedua kelas, yang disebut support vectors. Untuk memecahkan masalah optimasi pada SVM menggunakan Rumus 5, dengan syarat memenuhi Rumus 6.

$$\text{Minimize} \frac{1}{2} \|\mathbf{w}\|^2 \quad (5)$$

$$\mathbf{y}_i = (\mathbf{w} * \mathbf{x}_i + \mathbf{b}) \geq 1 \quad (6)$$

Dimana,  $\mathbf{w}$  adalah vektor bobot yang menentukan arah *hyperline*,  $\mathbf{x}_i$  adalah data pada titik ke-i,  $\mathbf{y}_i$  adalah label kelas dari data ke-i (biasanya +1 atau -1).  $\mathbf{b}$  adalah bias atau *intercept*,  $\|\mathbf{w}\|$  adalah norma atau panjang dari vektor bobot w, dan  $\mathbf{w} * \mathbf{x}_i$  adalah dot product antara  $\mathbf{w}$  dan  $\mathbf{x}_i$ .

#### 4. Hasil dan Pembahasan

##### 4.1. Hasil Pemodelan Menggunakan SVM

Pada Gambar 8, merupakan hasil visualisasi plot grafik dari pemodelan KNN dengan menggunakan data murni yang telah dihapus outliernya pada dataset, dengan pembagian dataset 70% untuk data latih dan 30% untuk data uji. Evaluasi menggunakan Mean Absolute Percentage Error (MAPE) untuk mengukur seberapa akurat prediksi model dalam perbandingan dengan nilai sebenarnya. Untuk perhitungan MAPE menggunakan Persamaan 7.

$$MAPE = \frac{1}{n} \sum_{i=0}^n |\mathbf{y}_i - \hat{\mathbf{y}}_i| * 100 \quad (7)$$

Dimana,  $\mathbf{y}_i$  adalah nilai actual,  $\hat{\mathbf{y}}_i$  adalah nilai prediksi,  $n$  adalah jumlah data, dan  $|\mathbf{y}_i - \hat{\mathbf{y}}_i|$  adalah kesalahan absolut antara nilai aktual dan nilai prediksi.



Gambar 8. Hasil Pemodelan Menggunakan KNN

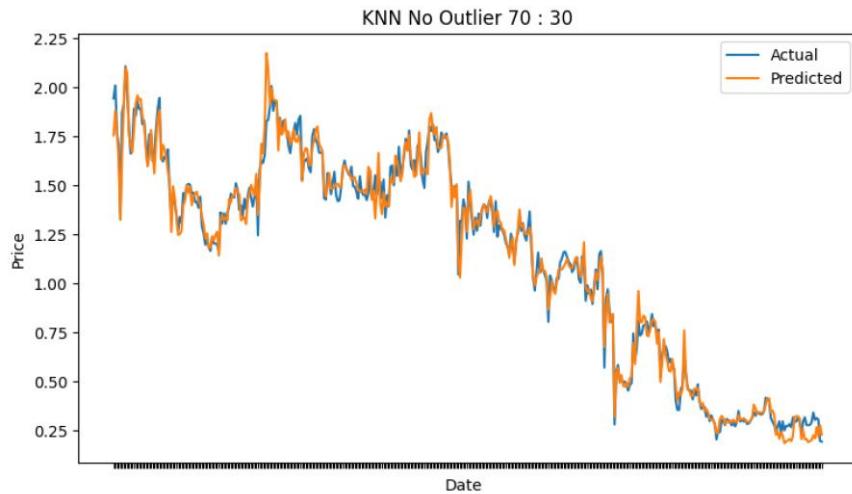
Hasil evaluasi dari pemodelan KNN tersebut diuraikan pada Tabel 3.

Tabel 3. Hasil Evaluasi Model KNN

Metrik Evaluasi	Hasil Evaluasi
RMSE	0.059982626727457314
MSE	0.0035979155091254763
MSE (Train)	0.019876872662838822
MSE (Test)	0.0035979155091254763
MAE	0.04330727358555059
MAPE	5.156978945002663%
R2 Score (Train)	0.9973630228077194
R2 Score (Test)	0.9872624056637286

##### 4.2. Hasil Pemodelan Menggunakan SVM

Pada Gambar 9, merupakan hasil visualisasi plot grafik dari pemodelan SVM dengan menggunakan data murni yang telah dihapus outliernya pada dataset, dengan pembagian dataset 70% untuk data latih dan 30% untuk data uji.



Gambar 9. Hasil Pemodelan Menggunakan SVM

Hasil evaluasi dari pemodelan KNN tersebut diuraikan pada Tabel 4.

Tabel 4. Hasil Evaluasi Model KNN

Metrik Evaluasi	Hasil Evaluasi
RMSE	0.07426574309139418
MSE	0.005515400596916962
MSE (Train)	0.4860238385827644
MSE (Test)	0.005515400596916962
MAE	0.052246975902956946
MAPE	5.034819778545642%
R2 Score (Train)	0.9963754057123995
R2 Score (Test)	0.9877629104953137

#### 4.3. Hasil Perangkingan Metode

Hasil evaluasi dan perankingan terhadap seluruh konfigurasi model dan prosesing, yang berfokus pada metrik MAPE karena sifatnya yang kokoh terhadap outlier, dapat dilihat Tabel 5.

Tabel 5. Hasil Perangkingan Model

Ranking	Metode	Teknik EDA	Split Data	Nilai MAPE
1	SVM	Standarisasi Z Score	70 : 30	5,03481%
2	KNN	Tanpa Outlier	70 : 30	5,15697%
3	SVM	Tanpa Outlier	70 : 30	5,16587%
4	KNN	Data Murni	70 : 30	5,24687%
5	SVM	Data Murni	70 : 30	5,44704%
6	SVM	Standarisasi Z Score Dengan Volume	80 : 20	5,68236%
7	SVM	Standarisasi Z Score	80 : 20	5,78272%
8	KNN	Data Murni	80 : 20	6,18770%
9	SVM	Tanpa Outlier	80 : 20	6,19249%

Hasil pengujian menampilkan 9 tingkat pengujian dengan hasil terbaik, yang mana pada urutan pertama pengujian menggunakan metode SVM teknik EDA Standarisasi Z Score dengan teknik split data 70 : 30. Pada urutan ke-2, pengujian dengan menggunakan metode KNN teknik EDA tanpa outlier dengan teknik split data 70 : 30. Pada urutan ke-3, pengujian dengan menggunakan metode SVM teknik EDA tanpa outlier dengan teknik split data 70 : 30. Pada urutan ke-4, pengujian dengan menggunakan metode KNN teknik EDA data murni dengan teknik split data 70 : 30. Pada urutan ke-5, pengujian dengan menggunakan metode SVM teknik EDA data murni dengan teknik split data 70 : 30. Pada urutan ke-6, pengujian dengan menggunakan metode SVM teknik EDA Standarisasi Z Score dengan menambahkan atribut volume dan teknik split data 80 : 20. Pada urutan ke-7, pengujian dengan menggunakan metode SVM teknik EDA Standarisasi Z Score dengan teknik split data 80 : 20. Pada urutan ke-8, pengujian dengan menggunakan

metode KNN teknik EDA data murni dengan teknik split data 80 : 20. Pada urutan terakhir, pengujian dengan menggunakan metode SVM teknik EDA tanpa outlier dengan teknik split data 80 : 20.

### Diskusi Hasil

Hasil analisis menunjukkan bahwa pemodelan SVM dengan preprocessing Standarisasi Z-Score dan split data 70:30 memberikan hasil terbaik, ditunjukkan oleh nilai MAPE terkecil yaitu 5.03%. Hal ini dapat dijelaskan melalui karakteristik algoritma SVM dan pentingnya transformasi skala dalam algoritma tersebut.

- Mengapa SVM dengan Standarisasi Z-Score unggul?

SVM bekerja dengan membangun hyperplane optimal di ruang berdimensi tinggi. Algoritma ini sangat sensitif terhadap skala fitur, karena kernel (terutama RBF) dan perhitungan margin bergantung pada jarak Euclidean antar titik. Fitur yang memiliki skala lebih besar akan mendominasi perhitungan jarak, yang bisa menyebabkan bias model terhadap dimensi tertentu.

Dengan menerapkan standarisasi Z-Score, seluruh fitur data distandardkan sehingga memiliki mean = 0 dan standar deviasi = 1. Ini membuat kontribusi tiap fitur dalam perhitungan jarak menjadi setara. Akibatnya, model SVM dapat menemukan margin yang lebih representatif dan akurat terhadap distribusi data yang sebenarnya.

- Mengapa standarisasi lebih berdampak pada SVM dibanding KNN?

Walaupun KNN juga menggunakan perhitungan jarak, dampak standarisasi terhadap KNN tidak selalu signifikan karena KNN bekerja secara lokal pada tetangga terdekat. Jika distribusi data tidak terlalu bervariasi antar fitur, KNN bisa tetap stabil meski tanpa standarisasi. Namun, pada SVM, ketidakseimbangan skala dapat merusak bentuk optimal dari margin, bahkan jika hanya satu fitur yang berskala jauh lebih besar.

### 5. Kesimpulan

Penelitian ini menunjukkan bahwa tahapan analisis yang sistematis dimulai dari pemilihan dataset, pemahaman atribut, pengecekan tipe data, penanganan missing value, hingga transformasi data berkontribusi signifikan terhadap peningkatan akurasi model prediktif. Standarisasi Z-Score terbukti memberikan dampak positif paling signifikan, terutama pada metode Support Vector Machine (SVM), dengan menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) terkecil sebesar 5,03% dan  $R^2$  Score mendekati sempurna, yaitu 0,987762.

Evaluasi dilakukan dengan menggunakan berbagai metrik seperti RMSE, MSE, MAE, MAPE, dan  $R^2$  Score terhadap model K-Nearest Neighbors (KNN) dan SVM. Perangkingan model berdasarkan nilai MAPE menunjukkan bahwa SVM dengan Z-Score dan split data 70:30 adalah konfigurasi terbaik, diikuti oleh KNN tanpa outlier dan SVM tanpa outlier. Namun demikian, meskipun model KNN tanpa outlier berada pada peringkat kedua dalam perangkingan, penghapusan outlier tidak direkomendasikan secara umum untuk prediksi data aktual (real-world data). Hal ini dikarenakan:

- Penghapusan outlier mengubah distribusi alami data sehingga hasil model menjadi tidak sepenuhnya merepresentasikan kondisi nyata.
- Dalam konteks penerapan nyata (*deployment*), data baru yang masuk tetap mengandung kemungkinan *outlier*, sehingga model yang *dilatih* tanpa outlier bisa menjadi kurang kokoh terhadap data dunia nyata.
- Tujuan penghapusan *outlier* dalam penelitian ini adalah untuk eksplorasi dan komparasi teknik preprocessing, bukan sebagai pendekatan utama dalam pemodelan produksi.

Dengan demikian, penghapusan outlier hanya diposisikan sebagai teknik pembanding untuk menguji sensitivitas model terhadap data ekstrem, dan bukan sebagai rekomendasi utama untuk praktik pemodelan prediktif. Pendekatan yang lebih tepat adalah melakukan normalisasi (misalnya Z-Score) tanpa menghapus outlier, agar model tetap adaptif terhadap variasi data yang wajar di dunia nyata.

### Kontribusi Penelitian

Kontribusi penelitian ini tidak hanya memberikan pemahaman mendalam mengenai dampak preprocessing terhadap performa model prediktif, tetapi juga dapat dimanfaatkan dalam pengembangan sistem prediksi pada berbagai sektor teknologi di Indonesia, seperti sistem prediksi permintaan logistik, prediksi konsumsi energi, serta sistem pengambilan keputusan berbasis data pada industri dan pemerintahan. Secara lebih luas, penelitian ini berkontribusi dalam kemajuan sains dan teknologi di bidang machine learning terapan, serta mendukung penyelesaian masalah nasional melalui peningkatan akurasi sistem berbasis kecerdasan buatan yang andal dan adaptif terhadap karakteristik data Indonesia yang sering kali memiliki distribusi tak merata dan *outlier* yang signifikan.

## 6. Kontribusi Penulis

**Kurnia Ramadhan Putra:** *Formal Analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, dan Writing – review & editing.* **Bahy Tsany Rizqullah:** *Data Collection, Evaluation Techniques dan Writing – original draft.*

## 7. Declaration of Competing Interest

Penulis menyatakan bahwa data yang dipublikasikan pada naskah tersebut tidak ada konflik kepentingan terhadap pihak-pihak manapun. Jika di kemudian hari ditemukan adanya hal tersebut, tanggung jawab sepenuhnya mengenai hal tersebut berada di pihak penulis.

## 8. Referensi

- Akila, V., Nitin, M. V. S., Prasanth, I., Reddy M, S., & Akash Kumar, G. (2023). A Cryptocurrency Price Prediction Model using Deep Learning. *E3S Web of Conferences*, 391. <https://doi.org/10.1051/e3sconf/202339101112>
- Alfiras, M., Bojiah, J., Mohammed, M. N., Ibrahim, F. M., Ahmed, H. M., & Abdullah, O. I. (2023). Powered education based on Metaverse: Pre- and post-COVID comprehensive review. *Open Engineering*, 13(1). <https://doi.org/10.1515/eng-2022-0476>
- Altayef, E., Anayi, F., Packianather, M., Benmahamed, Y., & Kherif, O. (2022). Detection and Classification of Lamination Faults in a 15 kVA Three-Phase Transformer Core Using SVM, KNN and DT Algorithms. *IEEE Access*, 10(January), 50925–50932. <https://doi.org/10.1109/ACCESS.2022.3174359>
- Ardiyansyah, G., Ferdiansyah, F., & Ependi, U. (2022). Deep Learning Model Analysis and Web-Based Implementation of Cryptocurrency Prediction. *Journal of Information Systems and Informatics*, 4(4), 958–974. <https://doi.org/10.51519/journalisi.v4i4.365>
- Ariyani, V., Putri, P., Prasetijo, A. B., & Eridani, D. (2022). Perbandingan Kinerja Algoritme Naïve Bayes Dan K-Nearest Neighbor (Knn) Untuk Prediksi Harga Rumah. *Jurnal Ilmiah Teknik Elektro*, 24(2), 162–171. <https://ejournal.undip.ac.id/index.php/transmisi>
- Cholil, S. R., Handayani, T., Prathivi, R., & Ardianita, T. (2021). Implementasi Algoritma Klasifikasi K- Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 6(2), 118–127. <https://doi.org/10.31294/ijcit.v6i2.10438>
- Dwivedi, Y. K., Hughes, L., Baabdullah, A. M., Ribeiro-Navarrete, S., Giannakis, M., Al-Debei, M. M., Dennehy, D., Metri, B., Buhalis, D., Cheung, C. M. K., Conboy, K., Doyle, R., Dubey, R., Dutot, V., Felix, R., Goyal, D. P., Gustafsson, A., Hinsch, C., Jebabli, I., ... Wamba, S. F. (2022). Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, 66(July), 102542. <https://doi.org/10.1016/j.ijinfomgt.2022.102542>
- Endarto, I. A., & Martadi. (2022). Analisis Potensi Implementasi Metaverse Pada Media Edukasi Interaktif. *Jurnal Barik*, 4(1), 37–51. <https://ejournal.unesa.ac.id/index.php/JDKV/>
- Fajri, M. S., Septian, N., & Sanjaya, E. (2020). Evaluasi Implementasi Algoritma Machine Learning K- Nearest Neighbors (kNN) pada Data Spektroskopi Gamma Resolusi Rendah. *Al-Fiziya: Journal of Materials Science, Geophysics, Instrumentation and Theoretical Physics*, 3(1), 9–14. <https://doi.org/10.15408/fiziya.v3i1.16180>
- Farida, Y., & Khasanah, Z. S. U. (2021). Analisis Performa Mata Uang Virtual (Cryptocurrency) Menggunakan Preference Ranking Organization Method For Enrichment Evaluation (Promethee). *Rekayasa*, 14(1), 1–9. <https://doi.org/10.21107/rekayasa.v14i1.8793>
- Ferdiansyah, Othman, S. H., Radzi, R. Z. M., Stiawan, D., & Sutikno, T. (2023). Hybrid gated recurrent unit bidirectional-long short-term memory model to improve cryptocurrency prediction accuracy. *IAES International Journal of Artificial Intelligence*, 12(1), 251–261. <https://doi.org/10.11591/ijai.v12.i1.pp251-261>
- Gunarto, D. M., Sa'adah, S., & Utama, D. Q. (2023). Predicting Cryptocurrency Price Using RNN and LSTM Method. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 12(1), 1–8. <https://doi.org/10.32736/sisfokom.v12i1.1554>
- Hansun, S., Wicaksana, A., & Khaliq, A. Q. M. (2022). Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches. *Journal of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00601-7>
- Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, 15(14), 5481–5487. <https://doi.org/10.5194/gmd-15-5481-2022>
- Kafil, M. (2019). Penerapan Metode K-Nearest Neighbors Untuk Prediksi Penjualan Berbasis Web Pada

- Boutiq Dealove Bondowoso. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 3(2), 59–66.  
<https://doi.org/10.36040/jati.v3i2.860>
- Kim, G., Shin, D. H., Choi, J. G., & Lim, S. (2022). A Deep Learning-Based Cryptocurrency Price Prediction Model That Uses On-Chain Data. *IEEE Access*, 10(May), 56232–56248.  
<https://doi.org/10.1109/ACCESS.2022.3177888>
- Maulud, D., & Abdulazeez, A. M. (2020). A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends*, 1(2), 140–147.  
<https://doi.org/10.38094/jastt1457>
- Moch Farryz Rizkilloh, & Sri Widyanesti. (2022). Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(1), 25–31. <https://doi.org/10.29207/resti.v6i1.3630>
- Parekh, R., Patel, N. P., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., Davidson, I. E., & Sharma, R. (2022). DL-GuesS: Deep Learning and Sentiment Analysis-Based Cryptocurrency Price Prediction. *IEEE Access*, 10(April), 35398–35409. <https://doi.org/10.1109/ACCESS.2022.3163305>
- Pradito, B., & Purnia, D. S. (2022). Komparasi Algoritma Linear Regression dan Neural Network Untuk Memprediksi Nilai Kurs Mata Uang. *EVOLUSI : Jurnal Sains Dan Manajemen*, 10(2), 64–71.  
<https://doi.org/10.31294/evolusi.v10i2.13284>
- Raharja, P. A. (2021). Prediksi Harga Ethereum Menggunakan Metode Vector Autoregressive. *Journal of Informatics, Information System, Software Engineering and Applications*, 3(2), 71–79.  
<https://journal.ittelkom-pwt.ac.id/index.php/inista/article/view/285>
- Robiyanto, R., Susanto, Y. A., & Ernayani, R. (2019). Examining the day-of-the-week-effect and the-month-of-the-year-effect in cryptocurrency market. *Jurnal Keuangan Dan Perbankan*, 23(3), 361–375. <https://doi.org/10.26905/jkdp.v23i3.3005>
- Seruni, D. S., Furqon, M. T., & Wihandika, R. C. (2020). Sistem Prediksi Pertumbuhan Jumlah Penduduk Kota Malang menggunakan Metode K-Nearest Neighbor Regression. *Sistem Prediksi Pertumbuhan Jumlah Penduduk Kota Malang Menggunakan Metode K-Nearest Neighbor Regression*, 4(4), 1075–1082.
- Sopiandi, I., & Susanti, D. (2022). Menganalisis Informasi Metaverse Pada Game Online Roblox Secara Garis Besar. *JURNAL PETISI (Pendidikan Teknologi Informasi)*, 3(1), 1–4.  
<https://doi.org/10.36232/jurnalpetisi.v3i1.2021>
- Sumarlinda, S., & Lestari, W. (2022). Aplikasi K-Nearest Neighbor (KNN) untuk Klasifikasi Penyakit Kardiovaskuler. *Sumarlinda, Sri Lestari, Wiji*, 55, 259–262.  
<http://ojs.udb.ac.id/index.php/Senatib/article/download/1897/1487>
- Tarakci, F., & Ozkan, A. (2021). Comparison of classification performance of kNN and WKNN algorithms. *Selcuk University Journal of Engineering Sciences*, 20(02), 32–37. <http://sujes.selcuk.edu.tr/sujes>
- Vaulia, N. P., Yuliari, K., Kadiri, U., & Author, C. (2023). Investments: Performance Cryptocurrencies And Stocks Investasi : Kinerja Cryptocurrencies Dan Saham. *Management Studies and Entrepreneurship Journal*, 4(1), 825–833. [www.ojk.go.id](http://www.ojk.go.id)
- Venkatesh, S., Kumar, M. V. V., & Virupakshappa, A. D. (2023). An effective imputation scheme for handling missing values in the heterogeneous dataset. *Indonesian Journal of Electrical Engineering and Computer Science*, 32(1), 423–431. <https://doi.org/10.11591/ijeecs.v32.i1.pp423-431>
- Widiantoro, A. D., & Harnadi, B. (2022). Review of User Comments for the OVO Fintech application using LDA. *6th International Conference on Information Technology, InCIT 2022*, 326–330.  
<https://doi.org/10.1109/InCIT56086.2022.10067746>
- Widiantoro, A. D., Wibowo, A., & Harnadi, B. (2021). User Sentiment Analysis in the Fintech OVO Review Based on the Lexicon Method. *2021 6th International Conference on Informatics and Computing, ICIC 2021*. <https://doi.org/10.1109/ICIC54025.2021.9632909>