

## **Perbandingan Algoritma Random Forest dan Support Vector Machine (SVM) untuk Prediksi Tren Pasar Cryptocurrency Berbasis Website**

**M. Zidni ilman<sup>1</sup>, Santi Rahayu<sup>2</sup>**

<sup>1,2</sup>Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia  
Email: [1imzidni58@gmail.com](mailto:1imzidni58@gmail.com), [dosen02367@unpam.ac.id](mailto:dosen02367@unpam.ac.id)

**Abstrak**—Pasar cryptocurrency memiliki karakteristik yang sangat fluktuatif dan sulit diprediksi secara manual, sehingga dibutuhkan pendekatan berbasis *machine learning* untuk membantu dalam menganalisis tren harga. Penelitian ini bertujuan untuk membandingkan performa algoritma *Random Forest* dan *Support Vector Machine* (SVM) dalam memprediksi tren pasar cryptocurrency, serta mengembangkan sistem prediksi berbasis *website* yang dapat diakses oleh pengguna. Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall* dan data historis *Bitcoin* serta *Ethereum* yang diperoleh dari *CoinGecko*. Proses pengolahan data mencakup pembersihan, pelabelan tren, serta pelatihan model dengan algoritma *Random Forest* dan SVM. Hasil pengujian menunjukkan bahwa kedua algoritma menghasilkan akurasi yang sangat tinggi, yaitu 100% untuk *Bitcoin* dan 98,61% untuk *Ethereum*. Sistem berhasil dikembangkan menggunakan *framework Flask* dan *Chart.js*, dengan fitur pelatihan model, prediksi otomatis, dan visualisasi grafik harga. Kesimpulannya, algoritma yang digunakan sama-sama efektif dalam memprediksi tren pasar kripto, dan sistem yang dibangun dapat dijadikan alat bantu analisis bagi pengguna.

**Kata Kunci:** *cryptocurrency*, prediksi tren, *Random Forest*, *SVM*, *website*

**Abstract**—The cryptocurrency market has highly fluctuating characteristics and is difficult to predict manually; therefore, a machine learning-based approach is needed to assist in analyzing price trends. This study aims to compare the performance of the Random Forest and Support Vector Machine (SVM) algorithms in predicting cryptocurrency market trends, as well as to develop a web-based prediction system accessible to users. The research applies the Waterfall software development method and utilizes historical data of Bitcoin and Ethereum obtained from CoinGecko. The data processing stages include data cleaning, trend labeling, and model training using the Random Forest and SVM algorithms. The test results show that both algorithms produce very high accuracy, namely 100% for Bitcoin and 98.61% for Ethereum. The system was successfully developed using the Flask framework and Chart.js, featuring model training, automatic prediction, and price trend visualization. In conclusion, both algorithms are equally effective in predicting cryptocurrency market trends, and the developed system can serve as an analytical tool for users.

**Keywords:** *cryptocurrency*, *trend prediction*, *Random Forest*, *SVM*, *website*

### **1. PENDAHULUAN**

Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi besar di berbagai bidang, termasuk sektor keuangan. Salah satu dampak signifikan dari transformasi ini adalah munculnya *cryptocurrency* sebagai bentuk mata uang digital yang berbasis *blockchain*. Namun, volatilitas harga *cryptocurrency* yang sangat tinggi menjadikannya sulit untuk diprediksi secara manual. Oleh karena itu, diperlukan pendekatan berbasis teknologi seperti *machine learning* untuk membantu menganalisis dan memprediksi tren pasar.

Dalam penelitian ini, digunakan dua algoritma populer yaitu *Random Forest* dan *Support Vector Machine (SVM)* untuk membandingkan kinerjanya dalam memprediksi tren pasar *cryptocurrency*. Kedua algoritma ini dipilih karena masing-masing memiliki keunggulan dalam mengolah data non-linear serta menghasilkan akurasi tinggi. Penelitian ini juga mengembangkan sistem berbasis *website* agar pengguna dapat mengakses hasil prediksi dengan mudah.

Berdasarkan latar belakang tersebut, terdapat beberapa permasalahan yang dihadapi dalam penelitian ini, yaitu tingginya volatilitas harga *cryptocurrency* yang menyulitkan proses prediksi tren pasar, serta kompleksitas data pasar *cryptocurrency* yang memerlukan algoritma *machine learning* yang andal. Oleh karena itu, rumusan masalah dalam penelitian ini difokuskan pada dua hal utama: pertama, algoritma manakah yang memiliki performa lebih baik antara *Random Forest* dan *Support*

*Vector Machine (SVM)* dalam memprediksi tren pasar *cryptocurrency*, dan kedua, bagaimana cara mengimplementasikan sistem prediksi berbasis *website* untuk tren pasar *cryptocurrency*.

## 2. METODE PENELITIAN

Metode penelitian ini menjelaskan teori pendukung, algoritma yang digunakan, serta tahapan analisis dalam membangun sistem prediksi tren pasar *cryptocurrency* menggunakan *machine learning*. Beberapa teori dasar yang digunakan dalam penelitian ini mencakup konsep *cryptocurrency*, prediksi tren pasar, serta *machine learning* sebagai pendekatan utama dalam pengembangan model.

### 2.1 Cryptocurrency

Menurut (Ramadhani, 2020) *Cryptocurrency* merupakan gabungan antara dua kata yakni “*cryptography*” yang berarti kode rahasia dan “*currency*” yang artinya mata uang. *Cryptocurrency* menggunakan jaringan internet untuk transaksi mata uang digital yang berbentuk virtual.

*Cryptocurrency* adalah suatu konsep/hal meningkatkan akan uang, numnastik, investasi dan teknologi. Akan tetapi para praktisi dan pemangku kepentingan kurang minat terhadap pemahaman komprehensif tentang teori dan fondasi (Disemadi, H. S., & Delvin, D., 2021).

Menurut (Ramadhan, M. C., & Kartika, A., 2023), *Cryptocurrency* adalah sebuah mata uang digital. Ia tidak tersedia dalam bentuk fisik layaknya koin atau uang tunai yang digunakan secara umum di seluruh dunia.

Dari semua definisi *cryptocurrency* diatas, maka dapat disimpulkan bahwa *cryptocurrency* merupakan mata uang digital berbasis kriptografi yang berfungsi sebagai alat transaksi virtual melalui jaringan internet, tanpa bentuk fisik, serta berkembang sebagai instrumen investasi dan inovasi teknologi.

### 2.2 Prediksi Tren Pasar

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa mendatang berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya dapat terminimalisir (Wasik, A., Fatah, Z., & Munazilin, A., 2024).

Tren dapat didefinisikan sebagai cerminan apa yang tampaknya terjadi pada waktu tertentu. Tren bisa terjadi di area mana saja dan tidak hanya mencerminkan mode, budaya pop, dan hiburan. Bisa juga ada tren di pasar saham menjadi bullish atau bearish, tergantung pada indikator ekonomi, atau tren politik yang mencerminkan suasana hati suatu negara saat ini (Nugraha, N., Kusmayanti, T., & Butar-butar, S., 2023).

Dari pengertian diatas dapat diambil kesimpulan bahwa prediksi tren pasar merupakan suatu proses perkiraan yang sistematis dan terukur yang dilakukan pada waktu tertentu.

### 2.3 Machine Learning

*Machine learning* (ML) adalah bidang studi yang berfokus pada perancangan dan analisis algoritma untuk memungkinkan komputer belajar dari data. Menurut Samuel, ML terdiri atas algoritma yang bersifat *generic* (umum), di mana algoritma tersebut mampu menghasilkan informasi atau manfaat dari data tanpa perlu menulis ulang kode secara spesifik (Wahyono, T., 2018).

Mitchel (1997) mengatakan bahwa pembelajaran mesin adalah program komputer yang memperoleh pengalaman (*Experience*, E) dalam menjalankan tugas (*Task*, T) dan mengukur peningkatan kinerja melalui ukuran kinerja (*Performance Measure*, P). Program dikatakan meningkatkan pengalaman E jika kinerja tugas meningkat dengan ukuran kinerja P. Menurut definisi ini, aplikasi pembelajaran mesin harus memiliki tiga komponen utama: tugas (T), pengukuran kinerja (P), dan pengalaman (E), yang masing-masing harus dapat diidentifikasi dalam sistem.

*Machine Learning* adalah Bidang pengajaran komputer meliputi algoritma yang memungkinkan komputer belajar dari pengalaman untuk menyelesaikan tugas tertentu dan meningkatkan kinerjanya tanpa pemrograman ulang secara eksplisit.

#### **2.4 Kelebihan Dan Kekurangan**

Kelebihan:

- a. Efisiensi Backup yang Tinggi dapat Menggunakan incremental backup dan Changed Block Tracking (CBT) untuk mempercepat proses pencadangan tanpa membebani sistem utama.
- b. Pemulihan Data yang Cepat dan Fleksibel, Instant VM Recovery memungkinkan pemulihan mesin virtual secara instan tanpa harus melakukan restore penuh.
- c. Automasi dan Manajemen yang Mudah, Antarmuka yang intuitif memudahkan pengelolaan backup tanpa perlu keahlian teknis tinggi.

Kekurangan:

- a. Memiliki biaya lisensi yang cukup tinggi, terutama untuk bisnis kecil atau startup dengan anggaran terbatas.
- b. Kinerja backup dan restore sangat bergantung pada kecepatan jaringan dan kapasitas penyimpanan
- c. Membutuhkan RAM dan CPU yang cukup tinggi saat menjalankan proses backup dan restore, terutama pada lingkungan dengan beban kerja besar.

#### **2.5 Dataset**

Penelitian yang ditulis oleh (Saadah & Salsabila 2021) berjudul "*Prediksi Harga Bitcoin Menggunakan Metode Random Forest*" menggunakan data historis harga *Bitcoin* yang berubah secara bertahap dan berulang. Data ini dikumpulkan dari 1 Januari 2019 hingga 13 Oktober 2019 dari situs *Investing.com*. *Dataset* ini memiliki atribut *Date*, *Price*, *Open*, *High*, *Low*, *Volume*, dan *Change (%)*. Atribut *Price* menunjukkan harga penutupan *Bitcoin* pada tanggal tertentu, sedangkan *Open* menunjukkan harga pembukaan. *Volume* mencatat jumlah transaksi, dan *Change (%)* menunjukkan perubahan harga harian dalam bentuk persentase. Data ini kemudian diproses melalui tahapan *preprocessing* untuk mengubah format dan menyesuaikan dengan kebutuhan algoritma, serta dilakukan pemilihan fitur (*feature selection*) untuk menentukan atribut yang paling berpengaruh terhadap harga *Bitcoin*.

Penelitian ini menggunakan kumpulan data harga *cryptocurrency*, terutama *Bitcoin*, yang dikumpulkan dari repositori pembelajaran mesin *coingecko.com*. Data ini mengandung informasi tentang berbagai transaksi, termasuk harga pembukaan (*open*), harga penutupan (*close/price*), volume transaksi, harga tertinggi (*high*), harga terendah (*low*), dan persentase perubahan harga (*change %*) selama periode waktu tertentu. *Dataset* ini sangat relevan untuk dianalisis dengan metode klasifikasi dan prediksi berbasis pembelajaran mesin karena mereka menunjukkan tren pasar yang berubah-ubah dalam rentang waktu.

Tabel 1. Variabel dataset

No	Variabel	Keterangan
1.	<i>Open</i>	Harga pembukaan koin kripto pada awal hari
2.	<i>High</i>	Harga tertinggi dalam periode tersebut
3.	<i>Low</i>	Harga terendah dalam periode tersebut
4.	<i>Close</i>	Harga penutupan pada akhir hari
5.	<i>Change%</i>	Persentase perubahan harga dari hari sebelumnya.

Kemampuan untuk menangani data yang kompleks dan tidak linier adalah dasar pemilihan algoritma dalam penelitian ini. Akibatnya, algoritma *Random Forest* dan *Support Vector Machine* (SVM) digunakan untuk membandingkan kinerja mereka dalam memprediksi tren pasar *cryptocurrency*. Agar hasil klasifikasi tidak bias, distribusi kelas dalam *dataset* harus seimbang, misalnya antara tren naik dan tren turun. Jika ada ketidakseimbangan kelas, metrik evaluasi seperti error rate (1 - akurasi) mungkin tidak cukup untuk menunjukkan kinerja model secara menyeluruh. Untuk memberikan gambaran evaluasi yang lebih akurat terhadap hasil prediksi dari masing-masing algoritma dalam hal ini, metrik tambahan seperti ketepatan, *recall*, dan skor F1 juga diperlukan.

## 2.6 Data Preprocessing

Untuk kebutuhan analisis pembelajaran mesin, *preprocessing* adalah tahap awal pengolahan data yang sangat penting, menurut (Nurkhaliza & Wijayanto, 2022b). Data mentah yang berasal dari berbagai sumber biasanya berisi nilai kosong, tipe data yang tidak sesuai, atau data kategorik yang tidak dapat diproses oleh algoritma tertentu. Oleh karena itu, agar data siap digunakan dalam pelatihan model, perlu dilakukan beberapa tahap transformasi. Data yang digunakan masih dalam bentuk mentah sehingga perlu dilakukan *preprocessing* terlebih dahulu, seperti mengubah data kategorik menjadi numerik melalui metode *One-Hot Encoding* dan *Label Encoding*. Ini menunjukkan betapa pentingnya mengubah format data menjadi numerik agar algoritma yang digunakan dapat memprosesnya.

*Preprocessing* juga mencakup pembersihan data teks dan transformasi data numerik, terutama dalam penelitian yang melibatkan analisis sentimen. Dalam konteks teks, data *preprocessing* termasuk beberapa proses, seperti *casefolding* (mengubah huruf kapital menjadi huruf kecil), tokenisasi (memecah kalimat menjadi kata), penghapusan *stopword* (kata umum seperti "dan", "yang", dll.), dan *stemming* (mengubah kata ke bentuk dasar). Sebagaimana dinyatakan oleh (Pusean, 2022) tahap *preprocessing* dilakukan untuk membersihkan data teks sebelum dilakukan klasifikasi, agar model dapat memahami makna dan pola dari teks dengan lebih akurat. Ini sangat penting saat menggunakan algoritma seperti SVM yang sangat sensitif terhadap struktur data input.

(Kurniawan, 2023) menyatakan bahwa kinerja model dapat dipengaruhi langsung oleh strategi *preprocessing* yang digunakan. Dalam penelitiannya tentang analisis sentimen aplikasi Qasir yang menggunakan algoritma SVM dan *Random Forest*, ia menguji empat skenario *preprocessing* dan menemukan bahwa kombinasi *preprocessing* yang digunakan sangat memengaruhi akurasi model. Menurut (Kurniawan, 2023) pemilihan skenario *preprocessing* yang sesuai berpengaruh terhadap performa algoritma, khususnya pada data teks yang mengandung banyak variasi kata dan struktur kalimat. Hal ini menunjukkan bahwa *preprocessing* bukan hanya proses teknis tetapi juga strategi penting dalam *pipeline* analisis data.

Pada penelitian ini, data *preprocessing* dilakukan terhadap data historis harga *cryptocurrency* yang bersifat time-series. Tahapan *preprocessing* yang dilakukan meliputi penghapusan nilai kosong (*missing values*), konversi data waktu menjadi format numerik, normalisasi skala data numerik, dan pelabelan tren harga (naik/turun) sebagai target klasifikasi. Hal ini penting untuk memastikan bahwa algoritma seperti *Random Forest* dan SVM dapat mengenali pola dari data

dengan lebih baik. Berdasarkan temuan dari penelitian-penelitian sebelumnya, *preprocessing* yang tepat dapat meningkatkan akurasi dan efisiensi model dalam melakukan prediksi tren pasar *cryptocurrency* secara akurat.

## **2.7 Tujuan dan Prinsip Umum *Preprocessing***

*Preprocessing* biasanya dilakukan untuk menghilangkan gangguan dan inkonsistensi pada data (seperti nilai kosong, duplikasi, atau format tanggal yang tidak seragam), mengurangi kemungkinan *leakage* data dengan memastikan bahwa transformasi yang dipelajari pada data latih tidak "bocor" ke data uji, menormalkan skala fitur agar algoritma yang sensitif terhadap skala (seperti SVM) lebih stabil, dan menginstruksikan label target (tren naik/turun) agar persoalan prediksi dapat direduksi menjadi masalah klasifikasi biner.

## **2.8 Pembersihan dan Konsistensi Data (*Data Cleaning*)**

Langkah pertama adalah memastikan integritas dan konsistensi *dataset*. Hal ini mencakup (R. Muhammad, 2025):

- a. Penyelarasan tipe dan format data: Kolom tanggal (*Date*) diubah ke tipe *datetime*, dan Kolom numerik (*Open*, *Low*, *Close*, *Volume*, dan *Market Cap*) dipastikan bertipe numerik (*float/int*) agar *pipeline machine learning* dapat memprosesnya.
- b. Pemeriksaan duplikasi: untuk mencegah bias pada model, baris data duplikat dihapus.
- c. Penanganan nilai ekstrem (*outlier*): Nilai ekstrem dan volatilitas kripto tinggi dapat menunjukkan kondisi pasar sebenarnya, tetapi nilai *outlier* yang artefaktual (misalnya karena kesalahan pencatatan) harus diinvestigasi. Studi nasional pada berbagai domain, seperti keuangan, klasifikasi kesehatan, dan rekayasa perangkat lunak, sering melaporkan praktik data pembersihan dan validasi integritas fitur seperti ini.

## **2.9 Penanganan Missing Values**

Ada dua cara yang dapat digunakan jika nilai hilang ditemukan pertama, menghapus baris atau kolom yang memiliki nilai yang hilang dalam jumlah kecil, atau melakukan imputasi. Pada penelitian ini, jumlah nilai yang hilang biasanya minimal karena data harga kripto harian umumnya lengkap. Namun, jika ada kemungkinan, pendekatan imputasi sederhana (seperti *fill forward* untuk deret waktu) atau teknik yang lebih canggih seperti KNN Imputer dapat dipertimbangkan. Penelitian sebelumnya menunjukkan bahwa teknik imputasi tertentu efektif dalam menurunkan galat dan meningkatkan performa model. Sebagai contoh, (Maula, 2022) menunjukkan bahwa performa model berbeda ketika normalisasi dan strategi penanganan nilai hilang diterapkan secara berbeda, penelitian lain menekankan KNN Imputer sebagai metode yang efektif untuk data numerik.

## **2.10 Normalisasi**

Dua metode yang dipertimbangkan:

- a. Min-Max Scaling (0–1) menggunakan *MinMaxScaler* dari *scikit-learn*, yang sering digunakan dalam penelitian nasional. Muhammad (2025) menggunakan untuk stabilisasi skala fitur sebelum pelatihan model *deep learning*.
- b. Ketika distribusi fitur mendekati normal atau ketika diperlukan skala standar lintas fitur yang heterogen, Z-score Standardization (mean=0, std=1) yang sesuai.

Dalam implementasi, penelitian ini memilih Min-Max Scaling karena (1) rentang 0–1 memudahkan interpretasi nilai fitur, (2) menjaga bentuk distribusi asli fitur (tidak mengubah bentuk distribusi seperti beberapa transformasi non-linear).

**2.11 Rekayasa Fitur (*Feature Engineering*) dan Pelabelan Tren****a. Rekayasa fitur**

Penelitian ini juga mempertimbangkan pembuatan fitur turunan sederhana, seperti return harian atau persentase perubahan harga penutupan, yang sering digunakan untuk menunjukkan dinamika harga secara relatif. Fitur inti (*Open, High, Low, Close, Volume*, dan *Market Cap*) sudah memberikan konteks pasar harian. Penggunaan fitur turunan untuk meningkatkan perbandingan *signal-to-noise* sesuai dengan praktik pengembangan fitur. Penelitian nasional menekankan pentingnya transformasi fitur (*encoding, replace value*, hingga korelasi) dalam meningkatkan kinerja model.

**b. Pelabelan tren**

Untuk membandingkan kinerja berbagai model klasifikasi (seperti studi komparatif model klasifikasi biner pada domain citra, kesehatan, atau keuangan), label tren dibuat menjadi biner dengan perbandingan harga penutupan pada hari  $t-1$  terhadap hari  $t$ .

Secara praktis, penelitian ini dapat menggunakan ambang batas perubahan harga untuk mengurangi sensitivitas terhadap perubahan harga yang sangat kecil. Misalnya, ambang batas  $\geq 0,1\%$  dianggap naik dan ambang batas  $\leq -0,1\%$  dianggap turun. Dalam industri keuangan, strategi ini sering digunakan untuk meningkatkan stabilitas label; namun, penerapan ambang harus disesuaikan agar tidak mengorbankan sampel kelas tertentu secara signifikan.

### **3. HASIL DAN PEMBAHASAN**

**3.1 Proses Data Preprocessing**

*Preprocessing* merupakan tahap penting dalam siklus pengembangan sistem berbasis *machine learning*, karena untuk menentukan kualitas data sebelum digunakan untuk pelatihan model. Aplikasi *backend* berbasis *Flask* yang telah dibangun memungkinkan untuk mempermudah proses preprocessing dilakukan. *Dataset* yang digunakan adalah data historis dari berbagai *cryptocurrency*, termasuk *Bitcoin* dan *Ethereum*, yang dikumpulkan dari situs CoinGecko dan disimpan dalam file *crypto.csv*.

Lima langkah utama terdiri dari tahapan preprocessing yaitu: pemeriksaan data awal, pembersihan data, normalisasi, pelabelan tren, dan pembagian data latihan-uji.

**3.2 Pengumpulan dan Pemeriksaan Data**

CoinGecko adalah layanan API yang menawarkan data harga aset kripto yang lengkap dan terstruktur, ini adalah sumber data historis *cryptocurrency* yang digunakan. Data yang diambil dari sistem ini diambil melalui endpoint */market\_chart* dengan interval=harian dan jangka waktu=360.

*Source code* diatas menjelaskan proses pengambilan data dari API coingecko, kemudian diproses dan menghasilkan data yang berisi dua kolom *timestamp* dan *price*. Data ini kemudian diubah menjadi format *datetime*, yang selanjutnya digunakan untuk membentuk fitur lain.

**3.3 Pembentukan Fitur Prediktor**

Sistem menghasilkan sejumlah fitur penting yang akan dimasukkan ke dalam model pembelajaran mesin melalui harga harian yang diperoleh. Pembentukan fitur dilakukan menggunakan metode yang bergantung pada analisis teknis sederhana:



```

1 df["date"] = pd.to_datetime(df["timestamp"], unit="ms")
2 df["open"] = df["price"]
3 df["close"] = df["price"].shift(-1)
4 df["high"] = df[["open", "close"]].max(axis=1)
5 df["low"] = df[["open", "close"]].min(axis=1)
6 df["change_pct"] = ((df["close"] - df["open"]) / df["open"]) * 100
7 df = df.dropna()
8 df["trend"] = (df["close"] > df["open"]).astype(int)

```

**Gambar 1.** Source code pengambilan data



```

1 url = "https://api.coingecko.com/api/v3/coins/{coin_id}/market_chart"
2     params = {
3         "vs_currency": "usd",
4         "days": "360",
5         "interval": "daily"
6     }
7     res = requests.get(url, params=params)
8     res.raise_for_status()

```

**Gambar 2.** Source code pembentukan fitur

Keterangan:

- open* merepresentasikan harga saat ini (hari ke-n).
- close* merupakan harga hari berikutnya (n+1), sehingga sistem belajar mengenali perubahan antar hari.
- high* dan *low* digunakan untuk menunjukkan volatilitas relatif antar dua titik waktu.
- change\_pct* dihitung sebagai perubahan persentase harian.

#### 3.4 Pelabelan Data

Setelah melakukan pembentukan fitur prediktor, selanjutnya yaitu pelabelan data. Fungsinya untuk mempresentasikan arah perubahan harga.



```

1 df["trend"] = (df["close"] > df["open"]).astype(int)

```

**Gambar 3.** Source code pelabelan data

Logika pelabelan ini menghasilkan:

- Label 1 jika harga naik (*bullish signal*)
- Label 0 jika harga turun atau tetap (*bearish/neutral signal*)

Berikut contoh data csv dari pelabelan ini:



1	2024-07-31,66219.85102486102,66219.85102486102,64679.778647832515,64679.778647832515,-2.3245159119781198,0
2	2024-08-01,64679.778647832515,65357.52968767435,64679.778647832515,65357.52968767435,1.047856029842685,1
3	2024-08-02,65357.52968767435,65357.52968767435,61407.29547397454,61407.29547397454,-6.044038318786958,0
4	2024-08-03,61407.29547397454,61407.29547397454,60738.7449259574,-1.0887151823849401,0
5	2024-08-04,60738.7449259574,60738.7449259574,58006.206586686734,58006.206586686734,4.498383989416413,0
6	2024-08-05,58006.206586686734,58006.206586686734,53956.26184159744,53956.26184159744,-6.98191621742889,0
7	2024-08-06,53956.26184159744,55959.84107399357,53956.26184159744,55959.84107399357,3.7133395982743476,1
8	2024-08-07,55959.84107399357,55959.84107399357,55899.95181146735,55899.95181146735,1.5365184857266203,0
9	2024-08-08,55899.95181146735,61859.83159984161,55899.95181146735,61859.83159984161,12.26694319207657,1
10	

**Gambar 4.** Contoh pelabelan data 0 dan 1

Dengan logika pelabelan seperti ini, masalah prediksi tren menjadi masalah klasifikasi biner yakni menentukan apakah harga akan naik atau tidak besok. Sejalan dengan penelitian yang dilakukan oleh (Pratama et al., 2021) menggunakan pendekatan label biner untuk prediksi harga saham berbasis *Machine Learning*.

### 3.5 Pembersihan Data

Seluruh proses pembentukan close (menggunakan `.shift(-1)`) maka akan menghasilkan baris terakhir menjadi NaN. Oleh karena itu, sistem secara otomatis menghapus baris yang memiliki nilai kosong:



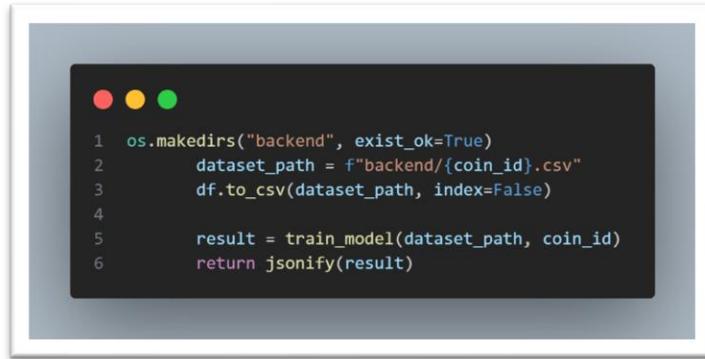
```
1 df = df.dropna()
```

**Gambar 5.** Source code pembersihan data

Tidak ada proses duplikasi eksplisit, dikarenakan data yang diambil dari CoinGecko bersifat unik diupdate per hari.

### 3.6 Penyimpanan Dataset dan Split Data

Setelah dilakukan proses pembersihan data dan siap untuk digunakan, proses selanjutnya yaitu sistem menyimpan ke dalam file csv sesuai dengan id coin (bitcoin.csv dan ethereum.csv), kemudian memanggil fungsi `train_model`.



```
1 os.makedirs("backend", exist_ok=True)
2 dataset_path = f"backend/{coin_id}.csv"
3 df.to_csv(dataset_path, index=False)
4
5 result = train_model(dataset_path, coin_id)
6 return jsonify(result)
```

**Gambar 6.** Source code penyimpanan database

Setelah disimpan ke dalam file csv, kemudian dilakukan proses split data:



```
1 fitur = ['open', 'high', 'low', 'close', 'change_pct']
2 X = df[fitur]
3 y = df['trend']
4
5 # Pisahkan data training dan testing secara acak
6 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

**Gambar 7.** Source code split data

### 3.7 Evaluasi Model

Tujuan dari tahap evaluasi model adalah untuk mengetahui seberapa baik algoritma klasifikasi berfungsi dengan data uji, atau data yang belum pernah dilihat sebelumnya. Untuk memprediksi tren pasar dua *cryptocurrency* utama yaitu *Bitcoin* dan *Ethereum*, evaluasi dilakukan pada dua algoritma *Random Forest* dan *Support Vector Machine* (SVM). Hasil evaluasi didasarkan pada data historis yang dikumpulkan dari 4 Agustus 2024 hingga 29 Juli 2025, sebanyak 72 sampel data uji digunakan yang merupakan 20% dari total data.

Untuk melakukan evaluasi model, maka digunakan empat metrik utama klasifikasi yaitu:

- Accuracy* adalah proporsi prediksi yang benar dibanding total data uji.
- Precision* fokus pada ketepatan model dalam mengklasifikasikan tren naik (1).
- Recall* adalah kemampuan model dalam menemukan semua tren naik (1) yang benar.
- F1-Score* menyeimbangkan antara *precision* dan *recall*.

Semua metrik dihitung dengan menggunakan fungsi *classification\_report* dari pustaka *Scikit-learn*, yang kemudian akan ditampilkan di terminal serta divisualisasi antarmuka web dengan menggunakan *Chart.js*.

### 3.8 Hasil Evaluasi Random Forest

*Random Forest* menunjukkan kinerja yang sangat tinggi, terutama pada data *Bitcoin*. Dengan menggunakan model yang telah dilatih pada 80% data, hasil evaluasi terhadap 20% data uji untuk *Bitcoin*. Berikut hasil dari pelatihan model dengan data *bitcoin*:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	36
1	1.00	1.00	1.00	36
accuracy			1.00	72
macro avg	1.00	1.00	1.00	72
weighted avg	1.00	1.00	1.00	72

**Gambar 8.** Hasil evaluasi *random forest* bitcoin

Hasil menunjukkan akurasi 100% dengan *precision*, *recall*, dan *f1-score* yang sempurna. Dengan kata lain, model dapat dengan akurat memprediksi setiap instance tren "naik" dan "turun". Performa ini menunjukkan bahwa model ini sangat cocok untuk digunakan sebagai model utama dalam sistem prediksi tren Bitcoin.

Model *Random Forest* menunjukkan kinerja yang luar biasa untuk *ethereum*, meskipun sedikit di bawah *bitcoin*:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	37
1	0.97	1.00	0.99	35
accuracy			0.99	72
macro avg	0.99	0.99	0.99	72
weighted avg	0.99	0.99	0.99	72

**Gambar 9.** Hasil evaluasi *random forest* ethereum

Dengan akurasi 98,61%, dengan hanya 1 sampai 2 kasus yang dianggap tidak tepat, Kedua kelas memiliki tingkat *precision* dan *recall* yang tinggi. Dengan nilai *F1-score* sebesar 0.99 menunjukkan bahwa model ini sangat konsisten dan responsif terhadap tren pasar riil.

Sejalan dengan penelitian yang dilakukan oleh (Fadillah et al., 2022) menunjukkan bahwa *Random Forest* berhasil menangani *dataset* keuangan yang fluktuatif dan non-linier. Hal ini disebabkan oleh mekanisme pembelajaran kelompok dan sampling fitur acak yang berhasil dalam mengurangi overfitting sekaligus menemukan pola kompleks yang tersembunyi di balik data kripto sebelumnya.

### 3.9 Hasil Evaluasi Support Vector Machine (SVM)

Model *support vector machine* (SVM) diuji dengan data yang sama seperti model *random forest*. Berdasarkan hasil pelatihan dan pengujian yang telah dilakukan, berikut hasil yang didapatkan:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	36
1	1.00	1.00	1.00	36
accuracy			1.00	72
macro avg	1.00	1.00	1.00	72
weighted avg	1.00	1.00	1.00	72

**Gambar 10.** Hasil SVM bitcoin

	precision	recall	f1-score	support
0	1.00	0.97	0.99	37
1	0.97	1.00	0.99	35
accuracy			0.99	72
macro avg	0.99	0.99	0.99	72
weighted avg	0.99	0.99	0.99	72

Gambar 11. Hasil SVM ethereum

Sama seperti *Random Forest*, SVM mendapatkan akurasi 100% pada Bitcoin dan 98.61% pada Ethereum dengan metrik *precision* dan *recall* yang sama. Ini menunjukkan bahwa model SVM juga mampu menangkap pola tren secara akurat dan dapat menjadi alternatif yang kuat untuk sistem ini.

Sebuah penelitian yang dilakukan oleh (Saputra & Lestari, 2020) menunjukkan bahwa model SVM sangat kompetitif untuk klasifikasi data pasar dalam kondisi stabil dan juga memiliki kemampuan untuk menghindari kesalahan klasifikasi pada data dimensi rendah.

Hal ini cocok dan bekerja dengan baik karena struktur *dataset cryptocurrency* yang digunakan hanya menggunakan fitur numerik seperti harga dan volume.

### 3.10 Perbandingan Kinerja Model

Pada pembahasan ini akan membuat tabel perbandingan kinerja di kedua model yang digunakan, berikut tabel perbandingan nya:

Tabel 2. Perbandingan kinerja model

Bitcoin		
Metrik	Random Forest	SVM
Akurasi	1.00	1.00
Precision	1.00	1.00
Recall	1.00	1.00
F1-Score	1.00	1.00
Ethereum		
Metrik	Random Forest	SVM
Akurasi	0.99	0.99
Precision	0.99	0.99
Recall	0.99	0.99
F1-Score	0.99	0.99

Hasil perbandingan di atas menunjukkan bahwa kedua model memiliki kinerja yang sangat baik dan setara pada Bitcoin dan Ethereum, masing-masing dengan akurasi yang sangat kecil (kurang dari 1%). Namun, penggunaan kedua model terus dilakukan untuk memberikan fleksibilitas dan validasi silang terhadap hasil prediksi. Karena kekuatannya terhadap noise dan outlier, *Random Forest* dapat digunakan sebagai model default dalam praktik implementasi.

Namun, jika pengguna ingin hasil pemrosesan yang cepat dan efisien, SVM masih relevan untuk digunakan.

### **3.11 Pembahasan Hasil**

Hasil evaluasi yang telah dilakukan menunjukkan bahwa baik algoritma *Random Forest* maupun *Support Vector Machine* (SVM) memiliki performa klasifikasi yang sangat baik dalam memprediksi tren pergerakan harga *Bitcoin* dan *Ethereum*. Akurasi kedua model pada data *Bitcoin* adalah 100%, dan akurasi pada data *Ethereum* adalah 98,61%. Kesimpulan ini perlu dianalisis lebih lanjut untuk memahami peran masing-masing model, kondisi data, dan konsekuensi yang terkait dengannya.

Secara umum, pencapaian akurasi tinggi menunjukkan bahwa fitur yang digunakan seperti harga *open*, *close*, *high*, *low* dan *change* memiliki korelasi yang kuat terhadap arah pergerakan harga berikutnya. Dengan kata lain, pola tren pasar dapat dengan akurat digambarkan dari informasi harga sebelumnya.

Beberapa poin penting yang dapat disimpulkan dari hasil pelatihan kedua model, yaitu:

a. Model Random Forest menunjukkan kinerja yang stabil dan konsisten

Dalam klasifikasi kedua jenis koin pada *Bitcoin*, hasilnya sempurna (*f1-score* 1.00). Keunggulan ini berasal dari pendekatan ansambel yang digunakan, yang menggabungkan metode voting dengan berbagai pohon keputusan *Random Forest*. Metode ini terbukti berhasil dalam mengurangi variasi model dan mencegah overfitting, terutama untuk data keuangan yang cenderung fluktuatif dan non-linier. Seperti yang ditunjukkan oleh (Fadillah et al., 2022), kemampuan *Random Forest* untuk melakukan feature bagging dan bootstrap sampling memungkinkannya mempertahankan kinerja yang konsisten terhadap data historis saham dan kripto.

b. Kemampuan SVM dalam Margin Maksimal dan Efisiensi Komputasi

Model SVM yang digunakan dalam sistem ini menggunakan kernel linier dan memiliki kemampuan dalam Margin Maksimal dan Efisiensi Komputasi. Pada data *Bitcoin*, sistem melakukan hal yang sama dengan *Random Forest*, tetapi sedikit berbeda pada *Ethereum*. Keseksian ini menunjukkan bahwa pemisahan kelas tren naik dan turun dalam ruang fitur dapat digambarkan secara linier. Karena hanya mempertimbangkan *support vector* selama proses klasifikasinya, SVM unggul dalam kecepatan pelatihan dan efisiensi memori. Menurut (Saputra & Lestari, 2020), SVM ideal untuk digunakan pada *dataset* berdimensi rendah dengan pola distribusi terstruktur, seperti tren harga kripto harian.

c. Konsistensi Hasil Prediksi Terhadap Data Real-Time

Sistem memungkinkan pengguna untuk melakukan prediksi otomatis berbasis data terbaru dari API CoinGecko, selain melatih model secara offline. Hasil pengujian data aktual per 29 Juli 2025 menunjukkan bahwa prediksi tren yang diberikan model sangat konsisten dengan kondisi pasar yang sebenarnya. Ini menunjukkan bahwa model tidak hanya memiliki kemampuan yang baik untuk mengklasifikasikan data historis tetapi juga dapat generalisasi ke data baru. Oleh karena itu, sistem ini memiliki kemampuan untuk berfungsi sebagai alat bantu pengambilan keputusan (*decision support system*) secara dinamis.

d. Keseimbangan dalam Klasifikasi Dua Kelas (Naik/Turun)

Setelah melakukan evaluasi terhadap metrik *precision* dan *recall* pada kedua kelas (tren naik dan turun), hasilnya menunjukkan bahwa model tidak bias terhadap satu kelas saja. Sementara *Ethereum* hanya memiliki perbedaan kecil, kedua model *Bitcoin* dapat mengkategorikan setiap instance dengan tepat. Dalam dunia keuangan, keseimbangan ini sangat penting karena salah klasifikasi tren, seperti memprediksi turun padahal naik, dapat menyebabkan keputusan investasi yang salah.

**e. Perbandingan Umum dan Implikasi Pemilihan Model**

Meskipun kedua model memiliki kinerja yang hampir identik, pilihan model utama dapat dipertimbangkan dari sudut pandang lain, seperti interpretabilitas, efisiensi komputasi, dan kebutuhan untuk memperluas sistem. *Random Forest* cocok untuk digunakan ketika stabilitas dan akurasi sangat penting, terutama karena ia tahan terhadap data yang bising. Karena sifatnya yang lebih mudah dan cepat dilatih, SVM dapat menjadi pilihan yang lebih baik. Dengan demikian, penggunaan kedua model dalam sistem ini memberikan pengguna fleksibilitas penuh untuk memilih model sesuai kebutuhan.

**f. Potensi Pengembangan dan Peringatan Overfitting**

Meskipun model sangat akurat, kondisi ini harus dipantau secara berkala agar sistem tidak overfitting terhadap pola tertentu. Model yang terlalu spesifik terhadap data pelatihan dapat gagal mengenali pola baru apabila ada perubahan besar di pasar. Oleh karena itu, sistem dapat dikembangkan lebih lanjut dengan menerapkan pendekatan hybrid atau stacking model dan menambahkan fitur indikator teknikal seperti moving average, RSI, atau MACD. Agar hasil prediksi tetap akurat dan relevan, evaluasi harus diperbarui secara berkala.

## **4. KESIMPULAN**

**4.1 Kesimpulan**

Berdasarkan hasil pembahasan pada bab sebelumnya, maka penulis dapat mengambil kesimpulan yaitu dengan menggunakan data historis Bitcoin dan Ethereum, penelitian ini berhasil membandingkan algoritma *Random Forest* dan *Support Vector Machine* (SVM) untuk memprediksi tren pasar *cryptocurrency*. Hasil evaluasi menunjukkan bahwa kedua algoritma memberikan performa yang sangat baik, dengan nilai akurasi 100% pada *Bitcoin* dan 98.61% pada *Ethereum*, serta nilai *precision*, *recall*, dan *f1-skor*nya hampir sempurna. SVM memiliki keunggulan dalam hal efisiensi komputasi, sementara *Random Forest* memiliki keunggulan dalam hal stabilitas dan ketahanan terhadap *overfitting*. Hal ini menunjukkan bahwa, tergantung pada konteks dan kebutuhan penggunaannya, kedua algoritma sama-sama layak digunakan untuk mengklasifikasikan tren pasar kripto jangka pendek.

Penelitian ini mengembangkan sistem prediksi tren *cryptocurrency* berbasis *web* yang menggunakan *Flask* untuk *backend* dan *Chart.js* untuk *frontend*. Pengguna dapat melatih model, melakukan prediksi otomatis berdasarkan data real-time, dan melihat visualisasi akurasi dan grafik harga dengan sistem ini. Dengan tampilan antarmuka yang sederhana namun fungsional, sistem ini dapat diakses secara fleksibel oleh pengguna sebagai alat bantu untuk memahami arah tren pasar, sehingga mendukung proses pengambilan keputusan secara data-driven.

**4.2 Saran**

Adapun saran berdasarkan penemuan ataupun opini dari penulis pada saat melakukan penelitian, berikut ada beberapa saran yang dapat diberikan:

- a. Untuk membuat sistem prediksi ini tetap relevan terhadap perubahan tren pasar, evaluasi berkala terhadap performa model diperlukan agar sistem tetap relevan. Penambahan fitur indikator teknikal dan pengayaan data pasar memungkinkan model untuk mengenali pola yang lebih kompleks.
- a. Sistem dapat disesuaikan untuk mendukung penggunaan yang lebih luas dengan menerapkan pengujian pengguna langsung untuk menilai fitur antarmuka dan kinerja aplikasi secara keseluruhan. Selain itu, sistem dapat dibuat menjadi versi yang ramah ponsel atau terintegrasi dengan fitur notifikasi prediksi

**DAFTAR PUSTAKA**

Akbar Rismayadi, A., Wahyudi Febrianto, R., Rachmat Raharja, A., & Hariyanti, I. (2024). Ifani Hariyanti Perbandingan Kinerja Metode Machine Learning SVM, Random Forest, dan KNN dalam Prediksi Harga Saham Apple.

Fahrezi, K., Mulana, A. R., Melinda, S., Nurhaliza, N., & Mulyati, S. (2021). Penerapan Model Waterfall dalam Pengembangan Sistem Informasi Akademik Berbasis Web sebagai Sistem Pengolahan Nilai Siswa. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 4(2), 98. <https://doi.org/10.32493/jtsi.v4i2.10196>

Filemon, B., Mawardi, V. C., & Perdana, N. J. (2022). PENGGUNAAN METODE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI SENTIMEN E-WALLET. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 10(1). <https://doi.org/10.24912/jiksi.v10i1.17824>

Karim, A. A., Ary Prasetyo, M., & Saputro, M. R. (2023). Perbandingan Metode Random Forest, K-Nearest Neighbor, dan SVM Dalam Prediksi Akurasi Pertandingan Liga Italia (Vol. 2). <http://www.football-data.co.uk>.

Kencana Putri, A., & Ichsanuddin Nur, D. (2023). PENGGUNAAN BAHASA PYTHON UNTUK ANALISIS DAN VISUALISASI DATA PENDUDUK DI DESA SUMBERJO, NGANJUK. In *Jurnal Pengabdian Kepada Masyarakat* (Vol. 3, Issue 3). [https://jurnalfkip.samawa-university.ac.id/karya\\_jpm/index](https://jurnalfkip.samawa-university.ac.id/karya_jpm/index)

Miftahusalam, A., Febby Nuraini, A., Khoirunisa, A. A., & Pratiwi, H. (2024). Perbandingan Algoritma Random Forest, Naïve Bayes, dan Support Vector Machine Pada Analisis Sentimen Twitter Mengenai Opini Masyarakat Terhadap Penghapusan Tenaga Honorer.

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>

Nalatissifa, H., Gata, W., Diantika, S., & Nisa, K. (2021a). Perbandingan Kinerja Algoritma Klasifikasi Naive Bayes, Support Vector Machine (SVM), dan Random Forest untuk Prediksi Ketidakhadiran di Tempat Kerja. *Jurnal Informatika Universitas Pamulang*, 5(4), 578. <https://doi.org/10.32493/informatika.v5i4.7575>

Naufal Hasani, M., Ramadhan, M., Mariyani, K., Setiawan, R., & Sucidha, I. (2022). JURNAL ILMIAH EKONOMI BISNIS ANALISIS CRYPTOCURRENCY SEBAGAI ALAT ALTERNATIF DALAM BERINVESTASI DI INDONESIA PADA MATA UANG DIGITAL BITCOIN. <http://ejournal.stiepancasetia.ac.id/index.php/jiebJilid>

Novianto, E., Suhirman, S., & Prasetyo, D. (2024). PERBANDINGAN METODE KLASIFIKASI RANDOM FOREST DAN SUPPORT VECTOR MACHINE DALAM MEMPREDIKSI CAPAIAN STUDI MAHASISWA. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 9(4), 1821–1833. <https://doi.org/10.29100/jipi.v9i4.5423>

Nurkhaliza, A. A., & Wijayanto, A. W. (2022a). Perbandingan Algoritma Klasifikasi Support Vector Machine dan Random Forest pada Prediksi Status Indeks Mitigasi dan Kesiapsiagaan Bencana (IMKB) Satuan Kerja BPS di Indonesia Tahun 2020. Maret, 7(1), 2622–4615. <https://doi.org/10.32493/informatika.v7i1.16117>

Purbolaksono, M. D., Irvan Tantowi, M., Imam Hidayat, A., & Adiwijaya, A. (2021). Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(2), 393–399. <https://doi.org/10.29207/resti.v5i2.3008>

R. Adrian, M., P. Putra, M., H. Rafialdy, M., & A. Rakhmawati, N. (2021). Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB.

Regina Lo, Alfred Edbert Yunanto, Rasya Nuhaifa Movia, Lambert Aditama Soehardjianto, Ferdinand Wangsa, Natalie Asdyo Lidjaja, & Rahmi Yulia Ningsih. (2023). Penggunaan Bahasa Pemrograman Python dalam Menganalisis Hubungan Kualitas Kopi dengan Lokasi Pertanian Kopi. *Jurnal Publikasi Teknik Informatika*, 2(2), 100–109. <https://doi.org/10.55606/jupi.v2i2.1752>

Saadah, S., & Salsabila, H. (2021). Jurnal Politeknik Caltex Riau Prediksi Harga Bitcoin Menggunakan Metode Random Forest (Studi Kasus: Data Acak Pada Awal Masa Pandemic Covid-19). In *Jurnal Komputer Terapan* (Vol. 7, Issue 1). <https://jurnal.pcr.ac.id/index.php/jkt/>

Samantri, M. (2024). Perbandingan Algoritma Support Vector Machine dan Random Forest untuk Analisis Sentimen Terhadap Kebijakan Pemerintah Indonesia Terkait Kenaikan Harga BBM Tahun 2022. *Jurnal Teknologi Informasi Dan Komunikasi*, 8(1), 2024. <https://doi.org/10.35870/jti>

Septiana, I., & Alita, D. (2024). Perbandingan Random Forest dan SVM dalam Analisis Sentimen Quick Count Pemilu 2024. *Jurnal Informatika: Jurnal Pengembangan IT*, 9(3), 224–233. <https://doi.org/10.30591/jpit.v9i3.6640>

Syauqi, A., Jinan, I., & Id, S. J. (2024). Studi Literatur Penerapan Machine Learning untuk Analisis Data Konsumen pada Minat Beli Konsumen. <https://doi.org/10.38035/jmpd.v2i4>

Andriani, F., & Syamsuddin, I. (2021). Penerapan algoritma Random Forest untuk prediksi harga saham berbasis data historis. *Jurnal Teknologi dan Sistem Komputer*, 9(2), 127–133.

Fadillah, F., Rahmawati, I., & Suryani, A. (2022). Analisis perbandingan algoritma Random Forest dan SVM dalam klasifikasi harga kripto. *Jurnal Ilmu Komputer dan Aplikasinya*, 4(1), 33–40.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Saputra, A. D., & Lestari, R. (2020). Klasifikasi sentimen menggunakan algoritma Support Vector Machine dan Naive Bayes. *Jurnal Informatika Universitas Pamulang*, 5(1), 23–29.

Sutrisno, A., & Pratama, R. Y. (2023). Pengembangan sistem prediksi harga cryptocurrency menggunakan metode Waterfall. *Jurnal Teknologi Informasi dan Terapan*, 10(1), 45–52.