

Implementasi Metode *Random Forest* dan *Support Vector Regression* dalam Memprediksi Harga *Cryptocurrency Ethereum*

Azizah Aulia Firdhasari¹, Lilis Sriwahyuni², Sri Nurdianti³, Mohamad Khoirun Najib^{4*}

^{1,2,3,4}*Sekolah Sains Data, Matematika, dan Informatika, IPB University, Bogor 16680, Indonesia*

*Corresponding Email**: mkhoirun@apps.ipb.ac.id

Abstrak

Perkembangan *cryptocurrency* menjadikan Ethereum (ETH) sebagai salah satu aset digital penting, namun pergerakan harganya sangat volatil karena dipengaruhi oleh berbagai faktor fundamental dan eksternal. Kondisi tersebut menyebabkan prediksi harga *close* ETH menjadi permasalahan utama karena akurasi peramalan sangat menentukan analisis dan pengambilan keputusan berbasis data. Penelitian ini bertujuan membangun serta membandingkan model prediksi harga *close* Ethereum menggunakan *Random Forest* dan *Support Vector Regression* (SVR) untuk *forecasting* 30 hari ke depan. Data yang digunakan berupa harga harian Ethereum periode 1 Januari 2020 hingga 30 Desember 2024 dari Yahoo Finance, kemudian dilakukan pra-pemrosesan, standarisasi, dan pembagian data *train-test* 80:20 dengan menjaga urutan waktu. *Feature engineering* dibangun dari harga *close* melalui MA 7, EMA 7, dan lag return 7, serta diterapkan *exponential smoothing* untuk mengurangi *noise*. Model *Random Forest* dan SVR dioptimasi menggunakan *Grid Search CV*, kemudian dievaluasi menggunakan metrik MAPE. Hasil *tuning* menunjukkan konfigurasi terbaik *Random Forest* adalah *max depth* = 10 dan total estimator = 90. Konfigurasi terbaik SVR adalah kernel linear dengan $C = 10$, $\epsilon = 0.5$, dan $\gamma = \text{scale}$. Evaluasi MAPE menunjukkan *Random Forest* lebih unggul dengan MAPE *train* 1,37% dan *test* 2,04%, sedangkan SVR menghasilkan MAPE *train* 5,83% dan *test* 2,22%. Secara keseluruhan, kedua model memberikan akurasi prediksi yang sangat baik, namun *Random Forest* menunjukkan kinerja lebih stabil dan akurat pada data pengujian. Model *Random Forest* kemudian digunakan untuk *forecasting* harga *close* ETH 30 hari ke depan sebagai estimasi jangka pendek yang cenderung stabil dan mengikuti tren data pengujian.

Kata Kunci: Ethereum, forecasting, MAPE, Random Forest, Support Vector Regression

Abstract

The development of *cryptocurrency* has made Ethereum (ETH) a key digital asset, but its price movements are highly volatile due to the influence of various fundamental and external factors. This situation makes predicting the closing price of ETH a major problem because forecasting accuracy is crucial for data-driven analysis and decision-making. This study aims to build and compare Ethereum closing price prediction models using *Random Forest* and *Support Vector Regression* (SVR) for 30-day forecasting. The data used is the daily Ethereum price from January 1, 2020, to December 30, 2024, from Yahoo Finance. Preprocessing, standardization, and an 80:20 train-test split are performed while maintaining the time sequence. *Feature engineering* is built from the closing price through MA 7, EMA 7, and lag return 7, and *exponential smoothing* is applied to reduce noise. The *Random Forest* and SVR models are optimized using *GridSearchCV* and evaluated using the MAPE metric. The tuning results show that the best *Random Forest* configuration is *criterion* = *absolute_error*, *max_depth* = 10, *max_features* = \log_2 , and *n_estimators* = 90. The best SVR configuration is a linear kernel with $C = 10$, $\epsilon = 0.5$, and $\gamma = \text{scale}$. MAPE evaluation shows that *Random Forest* is superior with a MAPE for *train* 1.37% and *test* 2.04%, while SVR produces a MAPE for *train* 5.83% and *test* 2.22%. Overall, both models provide excellent prediction accuracy, but *Random Forest* shows more stable and accurate performance on the test data. The *Random Forest* model is then used to forecast the ETH close price for the next 30 days as a short-term estimate that tends to be stable and follows the trend of the test data.

Keywords: Ethereum, forecasting, MAPE, Random Forest, Support Vector Regression

Received :23-01-2026 Revised :24-04-2026 Accepted :30-04-2026 Published :30-04-2026

1. Pendahuluan

Perkembangan teknologi digital telah mendorong lahirnya *cryptocurrency* sebagai inovasi di sektor keuangan, yaitu aset digital berbasis *blockchain* yang memungkinkan transaksi secara terdesentralisasi

dan transparan [1], [2]. Salah satu *cryptocurrency* yang paling berpengaruh adalah Ethereum (ETH), yang menempati posisi kedua setelah Bitcoin (BTC) [3], [4]. Selain berfungsi sebagai alat tukar, Ethereum juga menjadi platform utama bagi pengembangan *smart contract* dan aplikasi keuangan terdesentralisasi (*Decentralized Finance/DeFi*) [5], [6], [7]. Meskipun demikian, harga Ethereum dikenal sangat fluktuatif dan sulit diprediksi karena dipengaruhi oleh berbagai faktor, seperti sentimen pasar, tingkat adopsi teknologi, regulasi pemerintah, serta pergerakan aset kripto lainnya, khususnya Bitcoin [8], [9].

Sejumlah penelitian sebelumnya telah berupaya memprediksi harga *cryptocurrency* menggunakan pendekatan statistik dan *machine learning*. Menurut [10], model berbasis *deep learning* seperti *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU) mampu menangkap pola temporal, namun kinerjanya tidak selalu konsisten akibat tingginya *noise* dan kompleksitas data kripto. Penelitian tersebut juga menekankan pentingnya integrasi faktor kontekstual, seperti sentimen media sosial dan indikator makroekonomi, untuk meningkatkan akurasi prediksi harga Ethereum. Sementara itu, penelitian [11] membandingkan beberapa metode statistik dan *machine learning* — mencakup *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH), *Random Forest*, serta *Support Vector Regression* (SVR) — dengan temuan bahwa pendekatan *machine learning* lebih unggul dalam menangkap pola taklinear maupun volatilitas pasar kripto dibandingkan model klasik..

Meskipun LSTM dan GRU menunjukkan kemampuan dalam menangkap dependensi temporal jangka panjang, kedua model tersebut memiliki sejumlah keterbatasan yang signifikan dalam konteks prediksi harga kripto. Arsitektur *deep learning* tersebut membutuhkan volume data yang sangat besar, waktu komputasi yang lebih tinggi, serta proses *tuning hyperparameter* yang jauh lebih kompleks dibandingkan pendekatan *machine learning* konvensional. Selain itu, penelitian [10] sendiri mengakui bahwa kinerja LSTM tidak selalu konsisten akibat tingginya *noise* dan kompleksitas data kripto, sehingga akurasi prediksinya cenderung tidak stabil pada kondisi pasar yang bergejolak.

Sebaliknya, *Random Forest* dan SVR menawarkan beberapa keunggulan komparatif yang relevan. *Random Forest* bersifat robust terhadap *overfitting* berkat mekanisme ensemble pohon keputusan, tidak sensitif terhadap *outlier*, serta mampu menangani data berdimensi tinggi tanpa memerlukan asumsi distribusi tertentu. Sementara itu, SVR unggul dalam memodelkan hubungan taklinear melalui fungsi kernel, efektif pada dataset berukuran sedang, sekaligus lebih efisien secara komputasi dibandingkan *deep learning*. Keterbatasan penelitian-penelitian sebelumnya juga terletak pada cakupan fitur yang terbatas. Banyak studi hanya mengandalkan data harga historis mentah tanpa melibatkan rekayasa fitur teknikal seperti *Moving Average*, *Exponential Moving Average*, maupun *lag return*. Dengan demikian, kemampuan model dalam menangkap dinamika tren jangka pendek menjadi kurang optimal.

Penelitian [12], secara khusus mengembangkan model prediksi harga harian Ethereum menggunakan SVR dan menunjukkan bahwa algoritma ini efektif dalam menangani data yang bersifat non-linear dan sangat volatil. Dengan memanfaatkan data historis ETH periode 2019–2024 serta fitur teknikal seperti *Moving Average* dan *Exponential Moving Average*, SVR yang dioptimasi menggunakan *GridSearchCV* mampu menghasilkan performa yang sangat baik. Di sisi lain, penelitian [13] membandingkan *Random Forest*, SVM, dan LSTM, hasil penelitian menunjukkan bahwa *Random Forest* unggul dalam kestabilan dan efisiensi komputasi, sedangkan SVR lebih sensitif terhadap perubahan tren harga. Temuan ini menunjukkan potensi kombinasi kedua algoritma untuk saling melengkapi dalam meningkatkan akurasi peramalan harga kripto.

Berdasarkan kajian tersebut, penelitian ini bertujuan untuk mengevaluasi dan menentukan metode terbaik dalam memprediksi harga harian *Ethereum* (ETH) dengan membandingkan kinerja *Support Vector Regression* (SVR) dan *Random Forest* (RF) yang dioptimasi menggunakan *Grid Search*, serta (2) melakukan prediksi harga *Ethereum* harian untuk periode 30 hari ke depan menggunakan metode terbaik yang diperoleh. Evaluasi kinerja model dilakukan menggunakan *Mean Absolute Percentage Error*

(MAPE) sebagai metrik utama. Pendekatan ini diharapkan dapat meningkatkan akurasi prediksi harga *Ethereum* serta memberikan kontribusi terhadap pengembangan strategi investasi berbasis data dan kajian akademik di bidang keuangan digital.

2. Landasan Teori

Indikator teknikal merupakan alat analisis yang digunakan untuk mengidentifikasi pola pergerakan harga berdasarkan data historis suatu aset. Dalam pasar aset digital yang bersifat volatil dan dinamis, indikator teknikal berperan penting dalam membantu memahami tren harga, momentum, serta potensi perubahan arah pergerakan harga. Menurut penelitian [14], analisis teknikal pada *cryptocurrency* membutuhkan indikator yang mampu menangkap pola non-linear, ketidakstabilan tren, dan perubahan pasar yang terjadi secara cepat. Beberapa indikator teknikal yang umum digunakan dalam analisis dan peramalan harga *cryptocurrency* antara lain *Moving Average* (MA), *Exponential Moving Average* (EMA), dan *lag return*. Dengan mengintegrasikan indikator teknikal tersebut, model peramalan diharapkan dapat merepresentasikan dinamika pergerakan harga aset digital secara lebih akurat.

2.1. Moving Average (MA)

Moving Average (MA) adalah indikator teknikal yang digunakan untuk menghaluskan fluktuasi jangka pendek dalam deret waktu dengan menghitung nilai rata-rata dari harga pada periode tertentu. MA membantu menonjolkan pola tren sehingga memudahkan analisis pergerakan harga. Menurut [12], MA merupakan salah satu indikator teknikal yang digunakan dalam prediksi harga *Ethereum* karena mampu memberikan gambaran tren yang lebih stabil pada data yang volatil. MA dapat diformulasikan sebagai berikut:

$$\hat{Y}_t = \frac{1}{n} \sum_{i=1}^n Y_{t-i} \quad (1)$$

dengan Y_t merupakan harga pada waktu ke- t , \hat{Y}_{t-i} adalah harga pada periode sebelumnya, dan n adalah panjang periode rata-rata yang digunakan. MA banyak digunakan sebagai fitur input dalam model *machine learning* karena membantu model menangkap kecenderungan tren jangka pendek dan menengah.

2.2 Exponential Moving Average (EMA)

Exponential Moving Average (EMA) adalah indikator teknikal yang memberikan bobot lebih besar pada harga terbaru sehingga lebih responsif terhadap perubahan harga jangka pendek. EMA sering digunakan pada aset *cryptocurrency* karena mampu menangkap sinyal perubahan harga secara lebih cepat dibandingkan MA. Penelitian [12] menjelaskan bahwa EMA digunakan sebagai salah satu fitur dalam pemodelan prediksi harga *Ethereum* berbasis *machine learning* karena sifatnya yang adaptif terhadap dinamika pasar. EMA diformulasikan sebagai berikut:

$$\hat{Y}_t = \alpha Y_t + (1 - \alpha) \hat{Y}_{t-1} \quad (2)$$

dengan $\alpha = \frac{2}{n+1}$, Y_t adalah harga pada waktu ke- t , \hat{Y}_t adalah nilai EMA pada waktu ke- t , dan n adalah panjang periode yang digunakan. Parameter α berfungsi sebagai *smoothing factor* yang menentukan besarnya bobot pada harga terbaru.

2.3 Lag Return

Lag Return merupakan pendekatan transformasi data dalam analisis deret waktu yang memanfaatkan nilai *return* pada periode sebelumnya sebagai variabel prediktor untuk memperkirakan *return* atau harga pada periode mendatang. Pada aset kripto seperti *Ethereum*, yang memiliki volatilitas tinggi dan dipengaruhi dinamika pasar *real-time*, penggunaan *Lag Return* menjadi penting karena pergerakan harga

sering menunjukkan pola momentum jangka pendek maupun *reversal*, sehingga informasi *return* historis dapat memuat sinyal terhadap arah perubahan harga berikutnya. *Return* harga *Ethereum* umumnya dihitung menggunakan *log return* karena lebih stabil terhadap perubahan skala dan mampu merefleksikan pertumbuhan relatif secara konsisten. Formulasinya dituliskan sebagai berikut.

$$R_t = \ln \left(\frac{P_t}{P_{t-1}} \right). \quad (3)$$

Lag Return dengan selang k periode kemudian didefinisikan sebagai:

$$\text{lag}_k(R_t) = R_{t-k} \quad (4)$$

dengan P_t adalah harga penutupan *Ethereum* pada waktu ke- t dan R_t adalah *log return* harga *Ethereum* pada waktu ke- t .

3. Metode

3.1 Sumber dan Jenis Data

Penelitian ini menggunakan data harga harian *Ethereum* sebagai dasar dalam proses pemodelan dan prediksi. *Ethereum* dipilih karena memiliki kapitalisasi pasar terbesar dan volatilitas harga yang tinggi, sehingga relevan untuk dianalisis menggunakan metode *machine learning* seperti *Random Forest* dan *Support Vector Regression* (SVR). Penggunaan data harian memungkinkan model menangkap perubahan harga dengan resolusi yang lebih tinggi. Jenis data yang digunakan merupakan data sekunder yang diperoleh dari platform *Yahoo Finance* pada halaman riwayat harga ETH-USD. Sumber ini dipilih karena menyediakan data historis yang lengkap, mudah diakses, dan umum digunakan dalam penelitian keuangan. Rentang data mencakup periode Januari 2020 hingga Desember 2024, yang merepresentasikan berbagai kondisi pasar kripto. Data yang digunakan meliputi komponen *Open*, *High*, *Low*, *Close*, dan *Volume* (OHLCV), yang selanjutnya disesuaikan dengan kebutuhan analisis dan pemodelan.

3.2 Random Forest

Random forest mulai banyak digunakan sejak pertama kali diperkenalkan oleh [10] dan terbukti memiliki beberapa keunggulan, yaitu dapat mengatasi masalah *overfitting*, tidak sensitif terhadap pencilan dan dapat menghasilkan akurasi yang baik. Penggunaan *random forest* juga dapat diterapkan pada data tidak seimbang dalam jumlah besar dengan memberikan hasil performa yang baik dan waktu eksekusi yang cepat [15]. *Random forest* merupakan salah satu metode *ensemble*, yaitu metode untuk meningkatkan akurasi analisis klasifikasi [16], [17]. Metode ini membangun beberapa pengklasifikasi dasar atau tunggal dari data latih dan melakukan agregasi dengan mengambil *vote* terbanyak untuk menentukan hasil prediksi akhir [18]. *Random forest* dapat digunakan untuk regresi, yaitu ketika *output* yang diprediksi berupa nilai kontinu. Pada kasus regresi, pemilihan pemisah terbaik pada setiap node menggunakan fungsi kerugian (*loss function*) seperti *Mean Squared Error* dan *Mean Absolute Error* [19].

3.2.1 Mean Squared Error (MSE)

Mean Squared Error mengukur rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi untuk setiap data. Semakin kecil nilai MSE, maka semakin baik prediksi pada data tersebut.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (5)$$

Jika diterapkan pada satu node berisi himpunan data D yang berjumlah $|D|$, maka:

$$MSE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} (y_i - \hat{y})^2. \quad (6)$$

Jika node dipisahkan menjadi D_1 dan D_2 , maka proses pemisahan dilakukan dengan meminimisasi total *weighted mean squared error* setelah split, yaitu:

$$MSE_{split} = \frac{|D_1|}{|D|} MSE(D_1) + \frac{|D_2|}{|D|} SE(D_2). \quad (7)$$

Penimbang $\frac{|D_1|}{|D|}$ dan $\frac{|D_2|}{|D|}$ digunakan karena jumlah data pada masing-masing node dapat berbeda, sehingga kontribusi error tiap subset harus proporsional dengan ukuran datanya.

3.2.2 Mean Absolute Error (MAE)

MAE mengukur rata-rata nilai absolut selisih antara nilai aktual dan prediksi. MAE lebih *robust* terhadap pencilan dibanding MSE.

$$MAE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |y_i - \gamma| \quad (8)$$

dengan γ adalah median dari nilai target pada node tersebut. Pemilihan pemisah terbaik dilakukan dengan meminimalkan nilai MAE gabungan:

$$MAE_{split} = \frac{|D_1|}{|D|} MAE(D_1) + \frac{|D_2|}{|D|} MAE(D_2). \quad (9)$$

Pada *random forest* regresi, hasil akhir prediksi bukan menggunakan *majority vote*, melainkan menggunakan rata-rata dari prediksi seluruh pohon.

3.3. Support Vector Regression

Tujuan dari SVR adalah mendapatkan fungsi $f(x)$ yang mempunyai nilai galat paling banyak sebesar ϵ dari target yang didapatkan y_i terhadap setiap data latih [20]. Metode ini mempunyai tujuan untuk mencari batas atas dan batas bawah terhadap *hyperplane* dengan jarak ϵ sesempit mungkin. Nilai ϵ yang sempit menunjukkan toleransi yang rendah terhadap galat dan mengakibatkan semakin banyak *support vectors* yang terbentuk. Berdasarkan [20] pertama-tama perlu dideskripsikan terlebih dulu kasus fungsi linear f yaitu

$$f(x) = \langle w, x \rangle + b \quad (10)$$

dengan $w \in X, b \in \mathbb{R}, \langle \cdot, \cdot \rangle$ adalah dot product antara vektor bobot w dan data pada peubah penjelas x , dan b adalah intersep model.

Kesempitan jarak antara batas atas dan batas bawah terhadap *hyperplane* (ϵ) pada persamaan tersebut berarti bahwa diperlukan pencarian nilai w yang kecil. Meminimalkan $\|w\|^2$ merupakan permasalahan optimasi yang diformulasikan:

$$\text{minimalkan } \frac{1}{2} \|w\|^2 \quad (11)$$

dengan kendala $\begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases}$

Adakalanya diinginkan presisi yang dihasilkan tidak sebesar ϵ seperti pada persamaan tersebut yang *feasible*. Namun, membolehkan eror melebihi batas dengan cara menggunakan *soft margin loss function*. Cara ini menuntut penambahan peubah *slack* untuk mengatasi *infeasible constraints*. Penambahan peubah *slack* diformulasikan melalui persamaan:

$$\text{minimalkan } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (12)$$

$$\text{dengan kendala } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

dengan C adalah parameter regularisasi yang berfungsi untuk mengontrol *trade-off* antara kompleksitas model dan toleransi kesalahan prediksi.

Tiga jenis kernel yang paling umum digunakan dalam *Support Vector Regression* (SVR) adalah kernel linear, kernel *polynomial*, dan kernel *Radial Basis Function* (RBF) [21], [22], [23]. Setiap kernel memiliki formulasi matematis dan karakteristik yang berbeda dalam merepresentasikan hubungan antara variabel *input* dan *output*. Kernel linear sesuai untuk data yang memiliki hubungan linear. Di lain sisi, kernel *polynomial* mampu menangkap pola taklinear melalui derajat polinomial (d) yang berfungsi sebagai indikator kompleksitas model, dengan formulasi $K(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + r)^d$. Nilai d yang kecil (misalnya $d = 2$ atau $d = 3$) menghasilkan batas keputusan yang relatif sederhana, sedangkan nilai d yang besar meningkatkan fleksibilitas model dalam menangkap pola yang lebih kompleks, meskipun berpotensi menyebabkan *overfitting*. Sementara itu, kernel RBF dikenal memiliki fleksibilitas tinggi dalam memodelkan hubungan taklinear yang kompleks karena kemampuannya memetakan data ke ruang fitur berdimensi sangat tinggi. Oleh karena itu, pemilihan kernel yang tepat menjadi faktor krusial dalam menentukan kinerja SVR, karena berpengaruh langsung terhadap akurasi prediksi dan kemampuan generalisasi model.

3.4 Grid Search

Grid Search CV merupakan metode eksplorasi sistematis untuk menemukan kombinasi parameter terbaik melalui evaluasi *k-fold cross-validation*. Metode ini banyak diterapkan untuk melakukan *tuning* parameter pada model *machine learning* seperti SVR dan *Random Forest* karena menghasilkan pemilihan parameter yang lebih stabil dan memiliki generalisasi lebih baik. *Grid Search* merupakan teknik pencarian parameter yang efektif terutama ketika ruang pencarian relative kecil namun sensitif terhadap performa model [24]. Misalkan ruang pencarian parameter SVR didefinisikan sebagai:

$$C \in \{C_1, C_2, \dots, C_m\} \text{ dan } \gamma \in \{\gamma_1, \gamma_2, \dots, \gamma_n\}$$

Proses *Grid Search CV* akan mencari pasangan parameter optimal:

$$(C_i, \gamma_j) \arg \min CV_{error}(C_i, \gamma_j)$$

Parameter yang umum dioptimasi dalam proses ini meliputi parameter SVR seperti C , ε , dan γ , serta parameter *Random Forest* seperti $n_estimators$, max_depth , $max_features$, dan $min_samples_split$.

3.5 Mean Absolut Percentage Error (MAPE)

MAPE merupakan metrik evaluasi yang digunakan untuk mengukur akurasi model prediksi dengan mengekspresikan besar kesalahan dalam bentuk persentase. Secara matematis, MAPE didefinisikan sebagai:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (13)$$

dengan y_t adalah nilai aktual, γ_t adalah nilai prediksi, dan n adalah jumlah observasi. Pada setiap titik pengamatan, kesalahan dihitung dalam bentuk *percentage error*.

$$APE_t = \left| \frac{y_t - \gamma_t}{y_t} \right| \times 100\% \quad (14)$$

Menurut [25], kinerja prediksi model dari MAPE terbaik digunakan dapat dilihat melalui kriteria pada Tabel 1.

Tabel 1. Kriteria pada nilai MAPE

Nilai MAPE	Kriteria
< 10%	Kemampuan model dalam melakukan prediksi sangat baik
10% – 20%	Kemampuan model dalam melakukan prediksi baik
21% – 50%	Kemampuan model dalam melakukan prediksi layak
> 50%	Kemampuan model dalam melakukan prediksi buruk

Tabel 1 menyatakan kriteria nilai MAPE yang digunakan untuk melihat kemampuan model dalam melakukan prediksi. Model prediksi yang diharapkan harus memiliki nilai MAPE di bawah 20% untuk memiliki kemampuan prediksi yang baik sehingga hasil prediksi menjadi akurat.

4. Hasil dan Pembahasan

3.1 Prapemrosesan dan Eksplorasi Data

Penelitian ini menggunakan data historis harga Ethereum yang diperoleh dari *Yahoo Finance* pada periode 1 Januari 2020 hingga 30 Desember 2024. Dataset mencakup total 1826 observasi harga harian, yang ringkasannya disajikan pada pada Tabel 2 berikut ini.

Tabel 2. Data Historis Harga Ethereum

	Tanggal	Close	High	Low	Open	Volume
0	2020-01-01	130,802002	132,835358	129,198288	129,630661	7935230330
1	2020-01-02	127,410179	130,820038	126,95491	130,820038	8032709256
2	2020-01-03	134,171707	134,554016	126,490021	127,411263	10476845358
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1823	2024-12-28	3397,902344	3419,920166	3318,033936	3328,774658	14305648523
1824	2024-12-29	3349,513428	3406,648438	3321,664795	3397,862549	13440907792
1825	2024-12-30	3356,392578	3428,527344	3298,804443	3349,585938	26981583962

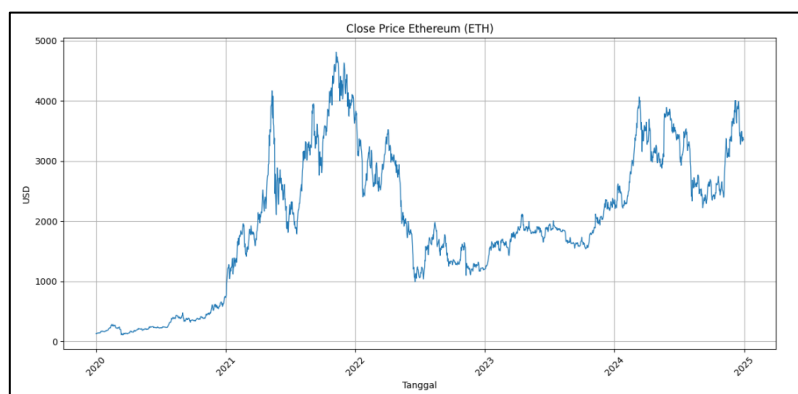
Setelah melalui serangkaian pra-pemrosesan data untuk memastikan kualitas dan konsistensi dataset, Tabel 3 menyajikan statistik deskriptif dari harga *Ethereum* yang mencakup ukuran pemusatan dan penyebaran data, termasuk jumlah data, nilai rata-rata, standar deviasi, *skewness*, kurtosis, serta nilai minimum dan maksimum, untuk memberikan gambaran komprehensif mengenai karakteristik dataset tersebut.

Tabel 3. Statistik Deskriptif

	Kolom	n	Rataan	Min	Max	Simp Baku	Skewness	Kurtosis
0	Open	1826	1980	110	4810	1148,00	0,09	-0,80
1	High	1826	2033	116	4892	1178,98	0,09	-0,80
2	Low	1826	1923	95	4718	1112,60	0,08	-0,80
3	Close	1826	1982	111	4812	1147,28	0,09	-0,80
4	Volume	1826	16,69 M	2,08 M	84,48 M	10,54 M	1,77	4,63

Berdasarkan statistik deskriptif, harga penutupan (*close*) ETH menunjukkan rentang yang sangat lebar dengan nilai minimum 111 dan maksimum 4.812. Selisih yang signifikan ini mengindikasikan fluktuasi harga yang ekstrem, mencerminkan volatilitas tinggi yang menjadi karakteristik khas aset kripto. Nilai simpangan baku yang lebih kecil dari rata-rata menunjukkan bahwa meskipun fluktuasi cukup besar, variasi harga masih proporsional terhadap nilai rata-ratanya. Nilai *skewness* sebesar 0.09 yang positif mengindikasikan distribusi data sedikit menceng ke kanan, menunjukkan bahwa sebagian besar harga ETH terkonsentrasi di sekitar nilai rata-rata dengan beberapa *outlier* pada sisi kanan. Sementara itu, kurtosis sebesar -0.80 yang lebih kecil dari 3 mengkonfirmasi bahwa frekuensi harga ekstrem (sangat tinggi atau sangat rendah) relatif lebih sedikit dibandingkan distribusi normal, menandakan distribusi yang lebih datar (*platykurtic*).

Analisis lebih lanjut terhadap data OHLV mengungkapkan pola serupa dimana selisih nilai minimum-maksimum pada semua variabel menunjukkan rentang yang sangat lebar, semakin memperkuat karakteristik volatilitas tinggi aset kripto. Nilai *skewness* yang positif pada seluruh variabel, dengan *skewness* volume mencapai 1.77, mengindikasikan adanya beberapa transaksi volume sangat besar yang menarik rata-rata lebih tinggi dari median, sehingga distribusi data lebih terkonsentrasi di sisi kiri. Nilai kurtosis yang kurang dari 3 pada sebagian besar variabel harga mengkonfirmasi distribusi yang relatif stabil dengan *occasional extreme movements*. Namun, kurtosis volume yang melebihi 3 (*leptokurtic*) menandakan adanya ekor yang lebih berat, dimana fluktuasi volume ekstrem terjadi lebih sering dibandingkan distribusi normal, mengindikasikan periode-periode lonjakan *trading activity* yang signifikan. Gambar 1 menyajikan pergerakan harga *close* ETH dari tahun 2020 sampai 2025.

**Gambar 1.** Harga Penutupan (*close*) ETH

Plot *time series* pada Gambar 1 mengkonfirmasi volatilitas tinggi harga *close* ETH melalui pola fluktuasi ekstrem yang selaras dengan temuan statistik deskriptif sebelumnya.

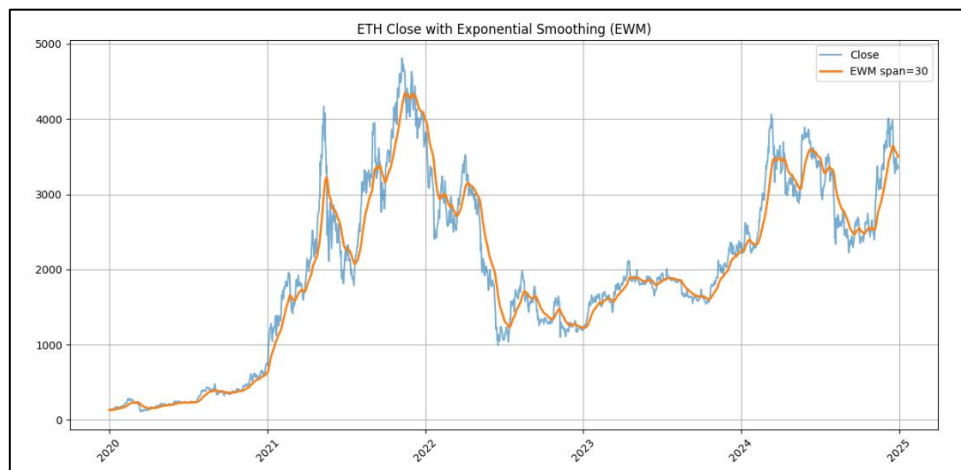
4.2 Standarisasi dan Feature Engineering

Proses persiapan data dilakukan secara sistematis dan berurutan, diawali dengan tahap standarisasi menggunakan *Standard Scaler*. Metode ini dipilih karena mampu mentransformasikan seluruh variabel ke dalam skala yang seragam dengan nilai rata-rata nol dan simpangan baku satu melalui formulasi $z = \frac{x-\mu}{\sigma}$, di mana x adalah nilai asli, μ adalah rata-rata, dan σ adalah simpangan baku fitur tersebut. *Standard Scaler* lebih diprioritaskan dibandingkan metode normalisasi lain seperti *Min-Max Scaler* karena data harga Ethereum memiliki distribusi yang tidak seragam dan mengandung nilai ekstrem (*outlier*) akibat volatilitas pasar kripto yang tinggi. Kondisi ini mengakibatkan *Min-Max Scaler* rentan terdistorsi secara signifikan oleh nilai minimum dan maksimum. Selain itu, algoritma SVR yang berbasis jarak (*distance-based*) sangat sensitif terhadap perbedaan skala antar variabel, sehingga standarisasi menjadi langkah krusial untuk menjaga stabilitas komputasi serta meningkatkan kinerja model secara keseluruhan. Proses standarisasi menjadi langkah yang krusial karena model *Support Vector Regression* (SVR) bersifat sensitif terhadap perbedaan skala antar variabel, sehingga standarisasi diperlukan untuk menjaga stabilitas komputasi serta meningkatkan kinerja model secara keseluruhan. Setelah tahap standarisasi, dilakukan *feature engineering* yang ditampilkan pada Tabel 4

Tabel 4. *Feature Engineering* Harga Penutupan (*close*) ETH

	Close	MA_7	EMA_7	lag_return_7
2020-01-08	141,258133	137,433473	138,667865	0,079939
2020-01-09	138,979202	139,086190	138,745699	0,090801
2020-01-10	143,963776	140,485057	140,050218	0,072982
2020-01-11	142,927109	141,607592	140,769441	0,058176
2020-01-12	145,873932	142,978614	142,045564	0,070424
2020-01-13	144,226593	142,967534	142,590821	-0,000537
2020-01-14	165,955353	146,169157	148,431954	0,156129
2020-01-15	166,230682	149,736664	152,881636	0,176787
2020-01-16	164,391007	153,366922	155,758979	0,182846
2020-01-17	170,779953	157,197804	159,514222	0,186270
2020-01-18	175,365677	161,831885	163,477086	0,226959
2020-01-19	166,969849	164,845588	164,350277	0,144617
2020-01-20	167,120514	168,116148	165,042836	0,158736
2020-01-21	169,697159	168,650691	166,206417	0,022547

Tabel 4 menunjukkan hasil proses *feature engineering* yaitu membangun tiga fitur turunan dari harga *close* Ethereum. Fitur MA 7 dihitung sebagai rata-rata harga penutupan selama tujuh hari untuk menangkap tren jangka pendek. Fitur EMA 7 dikembangkan dengan memberikan bobot yang lebih besar pada data terbaru, sehingga lebih responsif terhadap perubahan harga terkini. Sementara itu, fitur *lag return 7* dihitung berdasarkan perbandingan antara harga penutupan saat ini dan harga penutupan tujuh hari sebelumnya, yang merepresentasikan *return* mingguan sebagai indikator momentum harga. Proses *feature engineering* ini menghasilkan dataset sebanyak 1.819 observasi, yang selanjutnya diproses menggunakan teknik *exponential smoothing*. Hasil dari proses *smoothing* ditampilkan pada Gambar 2.



Gambar 2. Harga Aktual dan EWM Ethereum

Gambar 2 menampilkan perbandingan antara harga *close* Ethereum aktual dan hasil *penerapan exponential smoothing*. Selanjutnya, dataset dibagi menjadi data *training* dan *testing* dengan proporsi 80:20, sehingga diperoleh 1.455 observasi pada data *training* dan 364 observasi pada data *testing*. Pada data *training*, diterapkan metode *exponentially weighted moving average* (EWM) dengan *span* 30 hari pada variabel target Ethereum *close*. Penerapan teknik ini bertujuan untuk mengurangi *noise* dan fluktuasi harian yang bersifat ekstrem, sekaligus mempertahankan tren utama pergerakan harga. Proses pembagian data dilakukan dengan tetap mempertahankan urutan waktu (*time-ordered split*) guna menjaga integritas temporal data deret waktu dan menghindari data *leakage*.

4.3 Tuning Hyperparameter Support Vector Regression (SVR)

Support Vector Regression (SVR) juga memerlukan proses *tuning hyperparameter* yang bertujuan untuk memperoleh kombinasi parameter yang menghasilkan fungsi regresi paling optimal dengan tetap menjaga kemampuan generalisasi model dan menghindari *overfitting*. *Hyperparameter* yang dituning dalam penelitian ini meliputi *C*, epsilon, gamma, dan kernel. Kombinasi *hyperparameter* yang diuji untuk menentukan *best score* pada proses *tuning* SVR selanjutnya disajikan pada Tabel 5.

Tabel 5. Tuning Hyperparameter Support Vector Regression (SVR)

Indeks	C	Epsilon	Gamma	Kernel	Best Score
6	0,1	0,01	0,01	linear	-1,498682
9	0,1	0,01	0,1	linear	-1,498682
3	0,1	0,01	auto	linear	-1,498682
⋮	⋮	⋮	⋮	⋮	⋮
96	10	0,5	scale	linear	-0,094788
⋮	⋮	⋮	⋮	⋮	⋮
107	10,0	0,50	0,1	rbf	-1,361848
101	10,0	0,50	auto	rbf	-1,895909
98	10,0	0,50	scale	rbf	-1,895909

Berdasarkan hasil proses *tuning hyperparameter* yang disajikan pada Tabel 5, warna biru menunjukkan *best score*. Nilai parameter *C* yang relatif besar menunjukkan bahwa model memberikan penalti yang lebih tinggi terhadap kesalahan prediksi, sehingga mampu menyesuaikan fungsi regresi secara lebih akurat terhadap data pelatihan. Sementara itu, nilai ϵ yang lebih besar mengindikasikan adanya toleransi

kesalahan yang lebih lebar, yang berkontribusi dalam mengurangi sensitivitas model terhadap fluktuasi kecil dan *noise* pada data. Penggunaan Pemilihan kernel linear sebagai kernel terbaik mengindikasikan bahwa hubungan antara fitur hasil *feature engineering* dan harga penutupan Ethereum cenderung bersifat linear pada ruang fitur yang digunakan. Sebaliknya, kernel RBF menunjukkan performa yang lebih rendah berdasarkan nilai *best score* yang diperoleh dari proses *Grid Search CV*, di mana kombinasi parameter terbaik kernel RBF yakni $C = 10$, $\varepsilon = 0,5$, dan $\gamma = \text{scale}$ hanya menghasilkan skor sebesar $-1,895909$, jauh lebih rendah dibandingkan kernel linear yang mencapai $-0,094788$ pada kombinasi parameter yang sama. Hal ini mengindikasikan bahwa meskipun kernel RBF memiliki fleksibilitas tinggi dalam memetakan data ke ruang fitur berdimensi sangat tinggi, kompleksitas tersebut justru tidak memberikan manfaat signifikan pada data fitur hasil *feature engineering* yang digunakan dalam penelitian ini, sehingga peningkatan kompleksitas model tidak selalu berbanding lurus dengan peningkatan kinerja prediksi. Konfigurasi *hyperparameter* ini memberikan keseimbangan yang optimal antara akurasi dan kemampuan generalisasi, sehingga dipilih sebagai model SVR terbaik untuk tahap pemodelan dan prediksi harga Ethereum selanjutnya.

4.4 Tuning Hyperparameter Random Forest

Sebelum membangun model *Random Forest*, dilakukan proses *tuning hyperparameter* yang bertujuan untuk menyeimbangkan kompleksitas model dengan kemampuan generalisasinya melalui pengaturan berbagai *hyperparameter*. Proses ini diharapkan dapat meningkatkan stabilitas model, mengurangi risiko *overfitting*, serta menghasilkan performa prediktif yang optimal. Dalam proses *tuning*, model dengan nilai skor evaluasi tertinggi dipilih sebagai model terbaik. Tabel 6 menyajikan kombinasi *hyperparameter* yang diuji untuk menentukan *best score* pada proses *tuning Random Forest*.

Tabel 6. Tuning Hyperparameter Random Forest

Indeks	Criterion	Max depth	Max features	Estimators	Best Score
30	<i>absolute_error</i>	5	log2	80	-1,054053
0	<i>squared_error</i>	5	log2	80	-1,058726
31	<i>absolute_error</i>	5	log2	90	-1,050080
⋮	⋮	⋮	⋮	⋮	⋮
41	<i>absolute_error</i>	10	log2	90	-0,929924
⋮	⋮	⋮	⋮	⋮	⋮
28	<i>squared_error</i>	15	sqrt	110	-0,934562
59	<i>absolute_error</i>	15	sqrt	200	-0,935958
29	<i>squared_error</i>	15	sqrt	200	-0,933152

Berdasarkan hasil proses *tuning hyperparameter* yang disajikan pada Tabel 6, warna biru menunjukkan *best score*. Nilai skor evaluasi yang lebih mendekati nol mengindikasikan kesalahan prediksi yang lebih kecil, sehingga model memiliki kemampuan prediktif yang lebih baik. Penggunaan *criterion absolute error* menunjukkan bahwa model lebih *robust* terhadap nilai ekstrem (*outlier*), sementara pemilihan kedalaman pohon sebesar 10 mampu menyeimbangkan kompleksitas model dan kemampuan generalisasi. Selain itu, pemilihan *max features = log2* membantu meningkatkan diversitas antar pohon keputusan, sedangkan jumlah 90 estimator dinilai cukup untuk menghasilkan prediksi yang stabil tanpa meningkatkan kompleksitas komputasi secara berlebihan. Konfigurasi *hyperparameter* ini memberikan keseimbangan yang optimal antara akurasi dan generalisasi, sehingga dipilih sebagai model *Random Forest* terbaik untuk tahap pemodelan dan prediksi harga Ethereum selanjutnya.

4.5 Evaluasi Support Vector Regression dan Random Forest

Support Vector Regression (SVR) dan *Random Forest* selanjutnya dibangun menggunakan kombinasi *hyperparameter* terbaik yang diperoleh dari proses *tuning*. Model yang telah dikonstruksi kemudian dilatih

menggunakan data *training* dan diuji pada data *testing* untuk mengevaluasi kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya. Kinerja prediktif kedua model dievaluasi dengan membandingkan nilai prediksi terhadap data aktual menggunakan *Mean Absolute Percentage Error* (MAPE) sebagai metrik evaluasi utama. Nilai MAPE yang dihasilkan mencerminkan tingkat kesalahan relatif antara hasil prediksi dan data aktual, sehingga memberikan gambaran kuantitatif mengenai akurasi masing-masing model. Hasil evaluasi kinerja SVR dan *Random Forest* selanjutnya disajikan secara ringkas pada Tabel 7

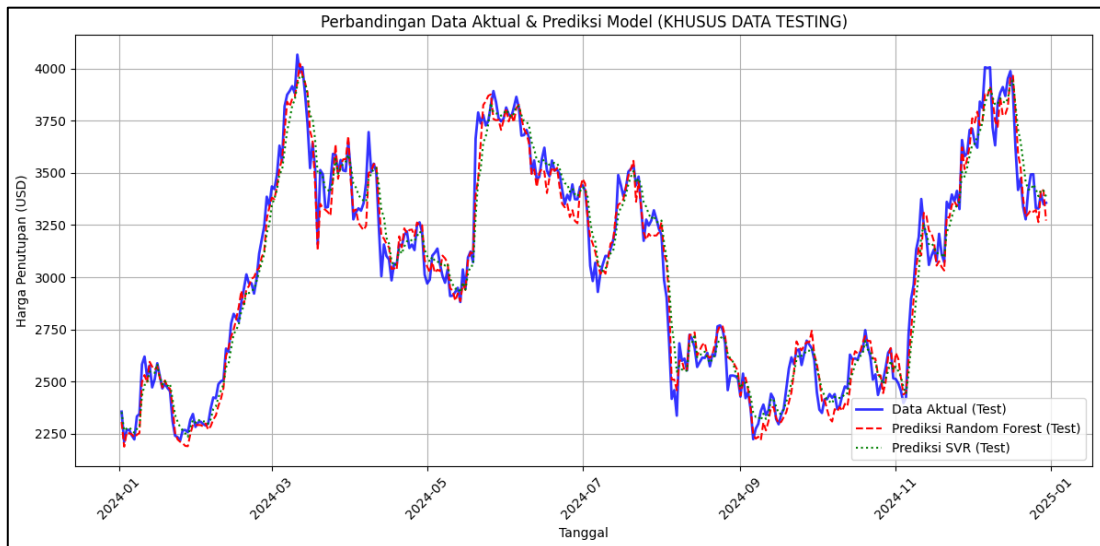
Tabel 7. Perbandingan MAPE Model *Random Forest* dan SVR

Model	MAPE <i>train</i> (%)	MAPE <i>test</i> (%)
SVR	5,835306	2,223804
<i>Random Forest</i>	1,372168	2,045167

Tabel 7 menyajikan perbandingan nilai MAPE pada data *training* dan *testing* untuk model SVR dan *Random Forest*. Hasil evaluasi menunjukkan bahwa model *Random Forest* menghasilkan nilai MAPE yang lebih rendah dibandingkan SVR, baik pada data *training* maupun data *testing*. Pada data *training*, *Random Forest* memperoleh nilai MAPE sebesar 1,37%, sedangkan SVR menghasilkan nilai MAPE sebesar 5,84%. Pada data *testing*, *Random Forest* tetap menunjukkan kinerja yang lebih baik dengan nilai MAPE sebesar 2,05%, sedikit lebih rendah dibandingkan nilai MAPE SVR sebesar 2,22%. Nilai MAPE yang relatif rendah dan konsisten antara data *training* dan *testing* pada model *Random Forest* mengindikasikan kemampuan generalisasi yang baik serta stabilitas model dalam memprediksi harga Ethereum. Meskipun selisih MAPE *test* antara *Random Forest* (2,04%) dan SVR (2,22%) tampak kecil secara absolut, perbedaan sebesar 0,18% ini memiliki implikasi yang cukup berarti dalam konteks strategi investasi. Sebagai ilustrasi, pada harga ETH sebesar \$3.000, selisih kesalahan prediksi antara kedua model setara dengan sekitar \$5,4 per unit aset. Dalam skenario perdagangan dengan volume tinggi atau posisi investasi yang besar, akumulasi selisih ini dapat berdampak signifikan terhadap keputusan *entry* dan *exit* posisi. Lebih jauh, *Random Forest* juga menunjukkan kesenjangan yang jauh lebih kecil antara MAPE *training* dan *testing* (0,67%) dibandingkan SVR (3,61%), yang mengindikasikan bahwa *Random Forest* lebih stabil dan tidak rentan terhadap *overfitting*. Ini adalah karakteristik yang sangat penting dalam penerapan model prediksi pada kondisi pasar kripto yang dinamis dan sulit diprediksi.

4.6 Prediksi Harga Close Ethereum

Setelah dilakukan evaluasi kinerja model *Support Vector Regression* (SVR) dan *Random Forest* melalui perhitungan nilai *error*, tahap selanjutnya adalah melakukan prediksi harga *close* Ethereum menggunakan kedua model tersebut. Prediksi dilakukan untuk periode historis tahun 2020 hingga 2024 guna membandingkan hasil prediksi model dengan data aktual dan mengevaluasi kemampuan masing-masing model dalam menangkap pola pergerakan harga dalam jangka panjang. Hasil prediksi harga *close* Ethereum menggunakan model SVR dan *Random Forest* selama periode tersebut selanjutnya disajikan secara visual pada Gambar 3.

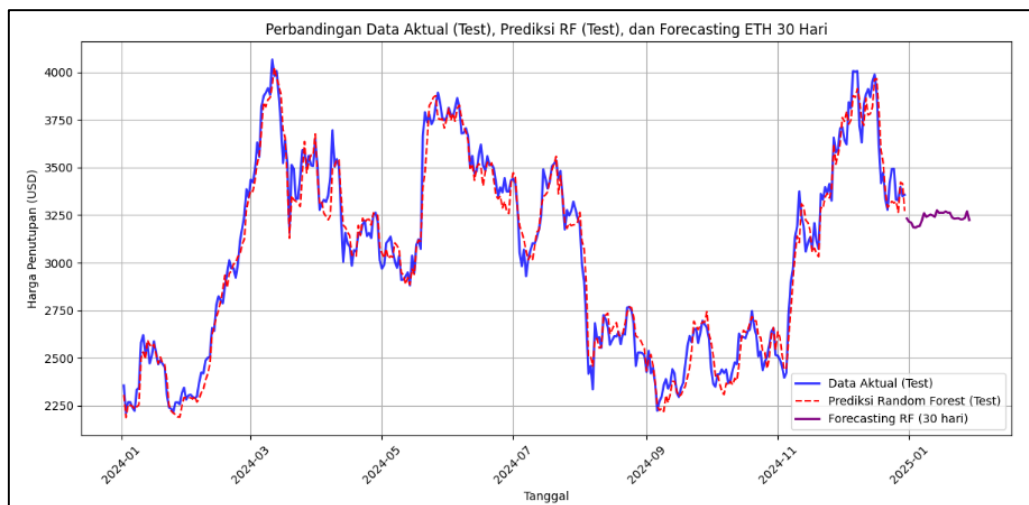


Gambar 3. Perbandingan data aktual dan prediksi data *close* ethereum

Berdasarkan hasil visualisasi dan evaluasi performa model dapat dilihat bahwa prediksi RF menunjukkan tingkat kesesuaian yang lebih baik dibandingkan SVR. Oleh karena itu, RF lebih akurat pada data pengujian sehingga untuk *forecasting* ETH 30 hari ke depan akan menggunakan model *Random Forest*.

4.7 Forecasting Harga Close Ethereum

Forecasting harga *close* Ethereum merupakan proses memprediksi nilai harga ETH pada periode mendatang berdasarkan data harga sebelumnya. Proses *forecasting* bertujuan untuk menilai kemampuan model dalam mengikuti perubahan harga aktual dan menghasilkan prediksi yang cukup tepat untuk mendukung analisis pada tahap pengujian. Pada Gambar 4 ditampilkan grafik *forecasting* harga ETH 30 hari ke depan dengan metode *Random Forest*.



Gambar 4. Grafik Perbandingan Data Aktual, Prediksi RF dan *Forecasting* ETH 30 Hari

Pada grafik tersebut terlihat bahwa pola prediksi cukup mendekati pola data aktual sehingga model mampu mengikuti tren perubahan harga dengan data *testing*. Sementara itu, pada *forecasting*

menunjukkan prediksi stabil tanpa perubahan ekstrem. Hal ini menunjukkan bahwa model mampu memberikan prediksi yang cenderung stabil saat memprediksi data di luar data testing. Secara umum, grafik ini menunjukkan bahwa *Random Forest* dapat mengikuti pola harga sebelumnya dengan baik dan dapat memberikan prediksi awal untuk pola harga periode selanjutnya.

5. Simpulan

Penelitian ini berhasil membangun dan mengimplementasikan dua model prediksi harga *close* Ethereum (ETH), yaitu *Random Forest* dan *Support Vector Regression* (SVR), dengan memanfaatkan data historis serta fitur teknikal MA, EMA, dan *lag return* sebagai variabel input. Tingkat akurasi kedua model yang diukur menggunakan MAPE menunjukkan bahwa *Random Forest* lebih unggul dan lebih stabil pada data pengujian karena memiliki MAPE *test* lebih rendah dibandingkan SVR, sehingga dipilih sebagai model terbaik untuk *forecasting*. Selanjutnya, model *Random Forest* digunakan untuk melakukan *forecasting* harga *close* ETH selama 30 hari ke depan, dan hasilnya menunjukkan pola prediksi yang relatif stabil serta konsisten mengikuti tren pada periode pengujian tanpa perubahan ekstrem yang tidak wajar.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan. Pertama, fitur yang digunakan hanya bersumber dari data teknikal internal harga ETH itu sendiri (MA, EMA, dan *lag return*), tanpa mempertimbangkan faktor eksternal seperti sentimen media sosial, indikator makroekonomi, maupun pergerakan aset kripto lain seperti Bitcoin yang secara empiris terbukti berpengaruh terhadap harga Ethereum. Kedua, horizon *forecasting* yang dihasilkan hanya mencakup 30 hari ke depan dengan pola yang cenderung stabil, sehingga model belum tentu mampu mengantisipasi perubahan harga yang bersifat mendadak dan ekstrem akibat peristiwa tak terduga (*black swan event*). Ketiga, penelitian ini tidak mempertimbangkan aspek *concept drift*, yakni perubahan distribusi data seiring waktu yang umum terjadi pada pasar kripto, sehingga performa model berpotensi menurun apabila diterapkan pada periode pasar yang berbeda secara struktural dari data pelatihan.

Berdasarkan keterbatasan tersebut, beberapa arah pengembangan yang dapat dilakukan pada penelitian selanjutnya antara lain: (1) mengintegrasikan fitur eksternal seperti indeks sentimen pasar (*Fear & Greed Index*), volume transaksi *on-chain*, serta data makroekonomi untuk meningkatkan representasi model terhadap dinamika pasar; (2) mengeksplorasi pendekatan *hybrid* yang menggabungkan *Random Forest* dengan model berbasis sekuens seperti LSTM guna memanfaatkan keunggulan masing-masing dalam menangkap pola tak-linear sekaligus dependensi temporal; serta (3) menerapkan mekanisme *retraining* berkala atau pendekatan *online learning* agar model dapat beradaptasi terhadap perubahan perilaku pasar secara dinamis.

Referensi

- [1] U. Kayani and F. Hasan, "Unveiling Cryptocurrency Impact on Financial Markets and Traditional Banking Systems: Lessons for Sustainable Blockchain and Interdisciplinary Collaborations," *J. Risk Financ. Manag.*, vol. 17, no. 2, 2024, doi: 10.3390/jrfm17020058.
- [2] A. Kumari and N. C. Devi, "The Impact of FinTech and Blockchain Technologies on Banking and Financial Services," *Technol. Innov. Manag. Rev.*, vol. 12, no. 1–2, 2022, doi: 10.22215/timreview/1481.
- [3] B. Łęć, K. Sobański, W. Świder, and K. Włosik, "Is the cryptocurrency market efficient? Evidence from an analysis of fundamental factors for Bitcoin and Ethereum," *Int. J. Manag. Econ.*, vol. 58, no. 4, pp. 351–370, 2022, doi: 10.2478/ijme-2022-0030.
- [4] Yaya Sovbetov, "Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litecoin, and Monero," *J. Econ. Financ. Anal.*, vol. 2, no. 2, pp. 1–27, 2018, [Online]. Available: https://mpr.aub.uni-muenchen.de/id/eprint/85036%0Ahttps://mpr.aub.uni-muenchen.de/85036/1/MPRA_paper_85036.pdf
- [5] A. Alamsyah, G. N. W. Kusuma, and D. P. Ramadhani, "A Review on Decentralized Finance Ecosystems," *Futur. Internet*, vol. 16, no. 3, 2024, doi: 10.3390/fi16030076.
- [6] R. Garg, "Ethereum based Smart Contracts for Trade & Finance," *World Acad. Sci. Eng. Technol.*

- Int. J. Econ. Manag. Eng.*, vol. 16, no. 11, pp. 619–629, 2022, [Online]. Available: <https://www.researchgate.net/publication/357510533>
- [7] B. Sriman and S. G. Kumar, “Decentralized finance (DeFi): The Future of Finance and Defi Application for Ethereum blockchain based Finance Market,” *Proc. - IEEE Int. Conf. Adv. Comput. Commun. Appl. Informatics, ACCAI 2022*, 2022, doi: 10.1109/ACCAI53970.2022.9752657.
- [8] H. M. Kim, G. W. Bock, and G. Lee, “Predicting Ethereum prices with machine learning based on Blockchain information,” *Expert Syst. Appl.*, vol. 184, 2021, doi: 10.1016/j.eswa.2021.115480.
- [9] Z. Asif and S. Unar, “Cryptocurrency Market Dynamics: Trends, Volatility, and Regulatory Challenges,” *Bull. Bus. Econ.*, vol. 13, no. 1, 2024, doi: 10.61506/01.00249.
- [10] A. Viéitez, M. Santos, and R. Naranjo, “Machine learning Ethereum cryptocurrency prediction and knowledge-based investment strategies,” *Knowledge-Based Syst.*, vol. 299, 2024, doi: 10.1016/j.knsys.2024.112088.
- [11] G. Dudek, P. Fiszeder, P. Kobus, and W. Orzeszko, “Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study,” *Appl. Soft Comput.*, vol. 151, 2024, doi: 10.1016/j.asoc.2023.111132.
- [12] Muhamad Abdul Fatah, Martanto, A. R. Dikananda, and A. Rifai, “Support Vector Regression to Improve Ethereum Price Prediction for Trading Strategies,” *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 2, pp. 738–744, 2025, doi: 10.59934/jaiea.v4i2.740.
- [13] S. A. Khoiri and A. Wahid, “Analisis Kinerja Algoritma Machine Learning dalam Prediksi Harga Cryptocurrency,” *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 9, no. 2, pp. 133–141, 2024.
- [14] K. Murray, A. Rossi, D. Carraro, and A. Visentin, “On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles,” *Forecasting*, vol. 5, no. 1, pp. 196–209, 2023, doi: 10.3390/forecast5010010.
- [15] A. S. More and D. P. Rana, “Review of random forest classification techniques to resolve data imbalance,” in *Proceedings - 1st International Conference on Intelligent Systems and Information Management, ICISIM 2017*, 2017, pp. 72–78. doi: 10.1109/ICISIM.2017.8122151.
- [16] G. Chairunisa *et al.*, “Life Expectancy Prediction Using Decision Tree, Random Forest, Gradient Boosting, and XGBoost Regressions,” *J. Sintak*, vol. 2, no. 2, pp. 71–82, 2024, doi: 10.62375/jsintak.v2i2.249.
- [17] E. H. Nugrahani, S. Nurdiati, F. Bukhari, M. K. Najib, D. M. Sebastian, and P. A. N. Fallahi, “Sensitivity and feature importance of climate factors for predicting fire hotspots using machine learning methods,” *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 2210–2223, 2024, doi: 10.11591/ijai.v13.i2.pp2212-2225.
- [18] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA, USA: Pearson Addison-Wesley, 2006.
- [19] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. New York, NY, USA: Springer, 2013.
- [20] J. S. Alex and B. Schoelkopf, “A tutorial on support vector regression,” *Stat. Comput.*, vol. 14, no. 2, pp. 199–222, 2004.
- [21] A. Razaque, M. Ben Haj Frej, M. Almi’ani, M. Alotaibi, and B. Alotaibi, “Improved support vector machine enabled radial basis function and linear variants for remote sensing image classification,” *Sensors*, vol. 21, no. 13, 2021, doi: 10.3390/s21134431.
- [22] G. L. Prajapati and A. Patle, “On performing classification using SVM with radial basis and polynomial kernel functions,” *Proc. - 3rd Int. Conf. Emerg. Trends Eng. Technol. ICETET 2010*, pp. 512–515, 2010, doi: 10.1109/ICETET.2010.134.
- [23] M. Alida and M. Mustikasari, “Rupiah Exchange Prediction of US Dollar Using Linear, Polynomial, and Radial Basis Function Kernel in Support Vector Regression,” *J. Online Inform.*, vol. 5, no. 1, pp. 53–60, 2020, doi: 10.15575/join.v5i1.537.
- [24] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A Practical Guide to Support Vector Classification,” National Taiwan University, 2003.
- [25] P. C. Chang, Y. W. Wang, and C. H. Liu, “The development of a weighted evolving fuzzy neural

network for PCB sales forecasting,” *Expert Syst. Appl.*, vol. 32, no. 1, pp. 86–96, 2007, doi: 10.1016/j.eswa.2005.11.021.



© **The Author(s) 2026.** This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License. Editorial of Journal of Mathematics: Theory and Applications, Department of Mathematics, Universitas Sulawesi Barat, Jalan Prof. Dr. Baharuddin Lopa, S.H., Talumung, Majene 91412, Sulawesi Barat.