

Extreme Learning Machine Hyperparameter Optimization Using Particle Swarm Optimization and Technical Indicators for Bitcoin Forecasting

Bagus Putra Sulung^{1*}, Wiwit Agus Triyanto², Pratomo Setiaji³

^{1,2,3}Sistem Informasi, Fakultas Teknik, Universitas Muria Kudus, Kudus, Indonesia
Email: ^{1*}202253111@std.umk.ac.id, ²at.wiwit@umk.ac.id, ³pratomo.setiaji@umk.ac.id
(*: corresponding author)

Abstract - Highly volatile price changes in crypto assets like Bitcoin complicate precise forecasting. The Extreme Learning Machine (ELM) artificial neural network approach offers high computational speed but is prone to performance instability due to random weight initialization and manual hyperparameter determination. To overcome this problem, this study proposes a hybrid PSO-ELM algorithm to automate the search for optimal parameters, namely the number of hidden neurons and the regression regularization penalty. Evaluated using a five-fold Walk-Forward Validation to prevent data leakage, the model comparatively tested five input feature scenarios based on technical indicators, which are mathematical calculations from historical prices to identify market patterns. Results demonstrate the hybrid PSO-ELM outperforms conventional static models, reducing average error (MAPE) by 18.67 percent. The cross-scenario comparison reveals that applying the Simple Moving Average technical indicator yields the best forecasting model, achieving a 26.65 percent error reduction and a final MAPE accuracy of 2.15 percent. The contribution of this research is providing empirical evidence that automatic parameter optimization combined with a random fluctuation filtering feature is proven to be more robust and accurate in responding to extreme volatility compared to the stacking of various complex derivative momentum indicators.

Keywords: Bitcoin, Extreme Learning Machine, Particle Swarm Optimization, Price Forecasting, Technical Indicator

Abstrak – Perubahan harga yang sangat bergejolak pada aset kripto seperti Bitcoin menyulitkan upaya peramalan harga yang presisi. Pendekatan jaringan saraf tiruan *Extreme Learning Machine* (ELM) menawarkan kecepatan komputasi komprehensif, namun rentan terhadap instabilitas kinerja akibat inisialisasi bobot acak serta penentuan parameter hiper secara manual. Untuk mengatasi masalah tersebut, penelitian ini mengusulkan algoritma hibrida PSO-ELM guna mengotomatisasi pencarian parameter terbaik, yaitu batas jumlah neuron tersembunyi dan penalti regularisasi regresi. Model dievaluasi menggunakan skema *Walk-Forward Validation* lima lipatan untuk menghindari kebocoran data, dengan menguji secara komparatif lima skenario fitur input berbasis indikator teknikal, yakni perhitungan matematis dari riwayat harga untuk mengidentifikasi pola pasar. Hasil membuktikan bahwa algoritma hibrida PSO-ELM unggul dengan mereduksi persentase kesalahan rata-rata (MAPE) sebesar 18,67 persen dibandingkan model statis konvensional. Komparasi lintas skenario menyingkap bahwa penerapan fitur indikator teknikal *Simple Moving Average* menjadi model peramalan terbaik dengan penurunan tingkat kesalahan 26,65 persen dan pencapaian akurasi akhir MAPE 2,15 persen. Kontribusi penelitian ini adalah memberikan pembuktian empiris bahwa optimasi parameter otomatis yang dipadukan dengan fitur penyaring fluktuasi acak terbukti lebih tangguh dan akurat dalam merespons volatilitas ekstrem dibandingkan penumpukan berbagai indikator momentum turunan yang kompleks.

Kata Kunci: Bitcoin, *Extreme Learning Machine*, Indikator Teknikal, *Particle Swarm Optimization*, Peramalan Harga

1. PENDAHULUAN

Bitcoin merupakan aset kripto desentralisasi dengan pergerakan harga yang sangat ekstrem dan dapat berfluktuasi drastis dalam waktu singkat [1]. Fluktuasi yang tinggi ini menghadirkan risiko finansial yang besar sekaligus peluang keuntungan bagi investor, sehingga prediksi harga yang akurat sangat penting sebagai sistem peringatan dini dan dasar manajemen risiko. Sifat pergerakan data keuangan yang tidak beraturan dan kompleks (*non-linear*) ini membuat metode peramalan tradisional, seperti ARIMA dan Prophet, sering kali kesulitan dan rentan menghasilkan prediksi yang kurang tepat [2], [3]. Oleh karena itu, pendekatan *machine learning* tingkat lanjut makin diandalkan untuk menghasilkan analisis yang presisi pada data yang kompleks [4]. Permasalahan serupa juga dijumpai pada aset finansial lain yang nilainya tidak stabil. Pergerakan harga saham yang sangat fluktuatif menuntut penerapan model *machine learning* yang mampu menangkap dinamika data secara akurat guna mendukung pengambilan keputusan investasi [5].

Salah satu metode *machine learning* yang unggul untuk memprediksi data yang kompleks adalah *Extreme Learning Machine* (ELM). Metode ini dikenal memiliki proses perhitungan yang cepat dan kemampuan mempelajari pola data yang baik [6], [7], [8]. Namun, ELM memiliki kelemahan utama, yaitu kinerjanya sangat bergantung pada pengaturan awal di dalam sistemnya, di mana penentuan jumlah neuron tersembunyi (n_{hi}) dan parameter penalti regularisasi secara manual memicu kinerja yang tidak stabil [9], [10]. Padahal, pengaturan parameter hiper (*hyperparameters*) secara otomatis pada jaringan saraf sangatlah krusial untuk memaksimalkan performa prediksi serta meminimalkan risiko *overfitting* [11]. Untuk mengatasi masalah ini, algoritma metaheuristik *Particle Swarm Optimization* (PSO) diterapkan secara efisien guna mencari pengaturan ELM yang paling tepat [9]. Penggabungan model PSO-ELM terbukti secara empiris mampu mengatasi kelemahan metode

coba-coba (*trial-and-error*) tradisional dengan menemukan kombinasi *hyperparameter* jaringan yang tepat pada data rumit, sehingga menekan tingkat kesalahan prediksi secara drastis [6], [12], [13].

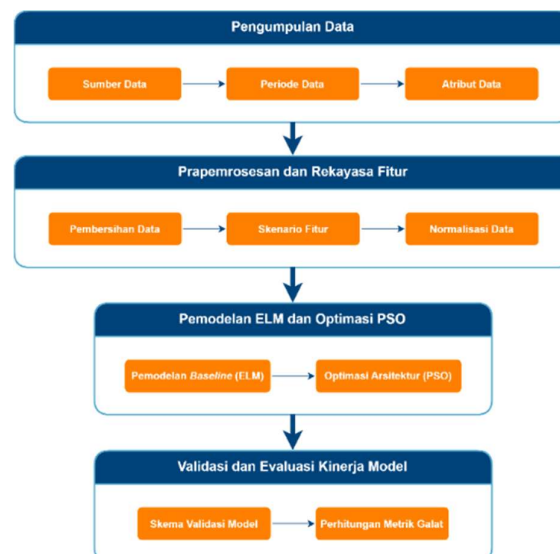
Selain mengoptimalkan algoritma, penambahan ragam fitur input melalui indikator teknikal juga sangat penting untuk memberikan informasi pasar yang lebih lengkap kepada model [14]. Penggunaan data pendukung ini terbukti efektif membantu *machine learning* dalam menangkap pola pergerakan dan hubungan antar variabel pada data keuangan, sehingga model menjadi lebih peka terhadap perubahan tren [15]. Secara umum, kombinasi indikator momentum seperti *Relative Strength Index* (RSI) untuk melihat kondisi beli atau jual berlebih dan *Moving Average Convergence Divergence* (MACD) untuk melihat arah tren, sering digunakan bersama karena dianggap saling melengkapi dalam menangkap momentum pasar yang tidak terlihat hanya dari data harga historis [16], [17], [18]. Meskipun model PSO-ELM dan indikator teknikal telah banyak diteliti, masih terdapat celah penelitian (*research gap*) terkait kurangnya integrasi yang komprehensif antara optimasi arsitektur jaringan, pemilihan indikator yang paling ideal untuk menangani tingginya gejala acak (*noise*) pada harga Bitcoin, dan metode pengujian yang kebal terhadap kebocoran data (*data leakage*). Pergerakan harga Bitcoin yang sangat bergejolak berisiko membuat model kebingungan jika hanya mengandalkan indikator turunan yang rumit. Oleh karena itu, kebaruan (*novelty*) dari penelitian ini terletak pada kombinasi ketiga elemen tersebut secara bersamaan, yakni integrasi antara optimasi *hyperparameter* ELM menggunakan PSO, penerapan pengujian *Walk-Forward Validation*, serta evaluasi komparatif terhadap lima skenario fitur indikator teknikal untuk menemukan pendekatan paling efektif dalam menstabilkan fluktuasi agar pola tren harga menjadi lebih halus dan model dapat belajar dengan lebih stabil [19].

Berdasarkan permasalahan dan gap tersebut, penelitian ini bertujuan membangun model prediksi harga penutupan Bitcoin harian ($t + 1$) menggunakan kombinasi arsitektur PSO-ELM yang mengotomatisasi pencarian parameter hiper [n_{hidden} , C] terbaik dan diuji menggunakan kerangka perbandingan lima skenario fitur input yang berbeda. Untuk memastikan hasil prediksi dapat dipercaya dan terhindar dari risiko kebocoran data (*data leakage*) saat masa pengujian, evaluasi model dilakukan menggunakan skema *Walk-Forward Validation* (WV) dan diukur secara ketat dengan metrik presisi peramalan MAE, RMSE, serta MAPE [20]. Kontribusi penelitian ini adalah merancang kerangka optimasi PSO-ELM yang terhindar dari kebocoran data pengujian, sekaligus menemukan kombinasi indikator terbaik yang mampu menyeimbangkan pembacaan arah tren pasar dengan peredaman fluktuasi harga Bitcoin. Secara praktis, hasil penelitian ini juga dapat menjadi panduan dalam merancang model prediksi aset digital yang akurat, andal, dan adaptif terhadap perubahan pasar.

2. METODE PENELITIAN

2.1. Alur Penelitian

Penelitian ini dilaksanakan melalui empat tahapan utama yang berjalan secara sistematis dan berkesinambungan. Keempat tahapan tersebut meliputi pengumpulan data, prapemrosesan dan rekayasa fitur, pemodelan ELM dan optimasi PSO, serta validasi dan evaluasi kinerja model. Alur logis dari pelaksanaan keseluruhan tahapan penelitian ini digambarkan secara sistematis pada Gambar 1.



Gambar 1. Alur Penelitian

2.2. Pengumpulan Data

Fokus utama eksperimen ini adalah memecahkan masalah volatilitas harga Bitcoin melalui pendekatan *single-step ahead forecasting* (memprediksi harga satu langkah ke depan atau $t + 1$). Data yang digunakan merupakan data sekunder deret waktu (*time series*) harga Bitcoin (BTC-USD) yang diambil menggunakan *Application Programming Interface* (API) Yahoo Finance [21]. Rentang waktu dataset ditetapkan selama lebih dari lima tahun, yakni dari 1 Januari 2021 hingga 31 Maret 2026, menghasilkan total 1.917 baris observasi harian. Pemilihan rentang waktu ini bertujuan agar algoritma dapat mempelajari berbagai kondisi pasar, baik saat tren harga sedang naik (*bullish*) maupun turun tajam (*bearish*), termasuk saat harga mencapai titik tertingginya (*All-Time High*). Atribut mentah yang dikumpulkan mencakup *Open*, *High*, *Low*, *Close*, dan *Volume*, dimana harga penutupan (*Close*) ditetapkan sebagai variabel target utama yang akan diprediksi. Rincian statistik dari atribut data tersebut disajikan pada Tabel 1.

Tabel 1. Statistik Deskriptif Dataset BTC-USD (1 Jan 2021 – 31 Mar 2026)

Atribut	Min	Max	Mean	Std Dev
<i>Open</i>	15,782	124,752	55,445	29,236
<i>High</i>	16,253	126,198	56,521	29,633
<i>Low</i>	15,599	123,196	54,289	28,800
<i>Close</i>	15,787	124,753	55,463	29,231
<i>Volume</i>	5.33B	350.97B	37.49B	22.89B

Statistik pada Tabel 1 mengungkap bahwa harga Bitcoin memiliki rentang pergerakan yang cukup besar serta volume transaksi yang berfluktuasi tinggi. Keragaman pola harga ini dinilai sangat memadai untuk melatih model jaringan saraf tiruan agar lebih tanggap dalam mengenali perubahan pasar yang tidak menentu.

2.3. Prapemrosesan dan Rekayasa Fitur

Tahap prapemrosesan diawali dengan pembersihan data menggunakan teknik interpolasi linear untuk mengisi nilai yang kosong (*missing values*) agar urutan waktu pada data tetap terjaga dan tidak terputus [22]. Selanjutnya, dilakukan ekstraksi indikator teknikal untuk menangkap momentum pasar. Secara definisi, indikator teknikal merupakan perhitungan matematis yang diekstrak dari riwayat harga historis yang mencakup harga pembukaan, penutupan, tertinggi, dan terendah untuk mengidentifikasi pola tren dan memprediksi arah pasar di masa depan. Fitur input (*input features*) disusun secara ketat menggunakan pendekatan lag waktu. Dalam pendekatan ini, riwayat transaksi harga sebelumnya hingga periode terkini (x_t) digunakan secara khusus untuk memprediksi harga satu hari ke depan (y_{t+1}).

Berbeda dengan pendekatan fitur statis, penelitian ini menerapkan rekayasa fitur dinamis (*feature engineering*) untuk menguji lima dimensi informasi pasar, yakni Skenario S1 (Harga Historis/*Baseline*), S2 (OHLC + *Volume*), S3 (Momentum RSI & MACD), S4 (Tren *Simple Moving Average*), dan S5 (Volatilitas *Bollinger Bands*). Untuk menghitung indikator seperti *Simple Moving Average* (SMA) periode 30 hari, model membutuhkan data historis awal sebagai pemanasan (*warm-up period*). Oleh karena itu, 30 baris pertama dari dataset dihapus agar ukuran matriks pada semua skenario menjadi seragam. Selain itu, satu baris paling akhir juga dipisahkan untuk dijadikan target prediksi masa depan. Setelah proses pemotongan ini, dataset akhir yang siap dilatih berjumlah 1.886 baris observasi.

Tabel 2. Skenario Fitur Input

Kode	Skenario	Fitur Input	Dimensi Input	Warm-up (Hari)	Keterangan
S1	Harga Historis	<i>Close</i> , <i>Lag 1</i> , <i>Lag 2</i>	3	2	<i>Baseline</i>
S2	OHLC + <i>Volume</i>	<i>Open</i> , <i>High</i> , <i>Low</i> , <i>Close</i> , <i>Volume</i>	5	0	Informasi pasar
S3	<i>Close</i> + RSI + MACD	<i>Close</i> , <i>Lag 1</i> , <i>Lag 2</i> , RSI (14), MACD (12,26)	5	26	Indikator momentum
S4	<i>Close</i> + SMA	<i>Close</i> , <i>Lag 1</i> , <i>Lag 2</i> , SMA (10), SMA (30)	5	30	Indikator tren
S5	<i>Close</i> + <i>Bollinger Bands</i>	<i>Close</i> , <i>Lag 1</i> , <i>Lag 2</i> , <i>Upper BB</i> , <i>Lower BB</i>	5	20	Indikator volatilitas

Kelima skenario pada Tabel 2 mewakili cara analisis yang berbeda. Skenario S1 berfungsi sebagai standar dasar, sedangkan S2 menggunakan harga mentah tanpa tambahan hitungan lain. Adapun Skenario S3, S4, dan S5 menambahkan perhitungan indikator teknikal khusus untuk membaca momentum, tren, dan volatilitas pergerakan harga. Sebagai contoh perhitungan, pada Skenario S3, susunan kelompok data masukan (x_t) dibentuk dari harga penutupan asli (C_t) beserta data harga beberapa hari sebelumnya (C_{t-1}, C_{t-2}). Data dasar ini kemudian digabungkan dengan nilai *Relative Strength Index* (RSI_t) periode 14 hari, serta nilai *Moving Average Convergence*

Divergence ($MACD_t$) dari selisih *Exponential Moving Average* periode 12 dan 26 [23]. Secara matematis, vektor masukan untuk skenario tersebut diformulasikan sebagai $x_t = [C_t, C_{t-1}, C_{t-2}, RSI_t, MACD_t]$. Logika penyusunan kelompok variabel ini juga diterapkan pada skenario lainnya dengan menyesuaikan jenis indikatornya. Sementara itu, variabel target keluaran akan selalu mengacu pada harga di masa depan (y_{t+1}). Sebelum keseluruhan data dimasukkan ke dalam tahap pelatihan model, seluruh nilai angka pada tabel ditransformasikan menggunakan *Min-Max Scaling* ke dalam rentang skala $[0, 1]$.

Penyesuaian skala ini wajib dilakukan agar tidak ada indikator bernilai masif (seperti harga puluhan ribu dolar) yang mendominasi perhitungan indikator bernilai kecil (seperti RSI). Langkah standardisasi ini sangat penting untuk mencegah kegagalan perhitungan komputer saat algoritma model dijalankan [24]. Guna mencegah terjadinya kebocoran informasi (*data leakage*), proses *Min-Max Scaling* ini tidak dilakukan secara serentak pada keseluruhan dataset di awal. Sebaliknya, perhitungan rentang skala (pencarian nilai minimum dan maksimum absolut) hanya diekstrak dari himpunan data latih (*training set*) pada masing-masing bagian (*fold*) pengujian. Parameter skala yang telah terbentuk dari data latih ini kemudian dikunci, dan secara sepihak diterapkan untuk mentransformasikan data uji (*test set*). Melalui prosedur ini, model dipastikan tidak pernah melihat skala ekstrem dari masa depan saat melakukan peramalan.

2.4. Pemodelan ELM dan Optimasi PSO

a. Pemodelan *Baseline Extreme Learning Machine* (ELM)

Extreme Learning Machine (ELM) merupakan algoritma pembelajaran yang dirancang khusus untuk jaringan saraf tiruan (*Neural Networks*) yang memproses data dalam satu arah (*Feedforward*) dengan menggunakan satu lapisan tersembunyi (*Single Hidden Layer*). Keunggulan utama ELM dibandingkan metode machine learning tradisional adalah proses belajarnya yang jauh lebih cepat. Metode konvensional biasanya memperbarui bobot perhitungan secara berulang-ulang (iteratif) sehingga memakan banyak waktu. Sebaliknya, ELM bekerja dengan cara membangkitkan nilai matriks bobot masukan (W) dan vektor bias (b) secara acak di awal, lalu menguncinya secara permanen [25]. Untuk sebuah jaringan dengan N sampel data masukan dan sejumlah neuron tersembunyi (n_{hidden}), keluaran dari lapisan tersembunyi (H) dihitung secara matematis melalui fungsi aktivasi $g(x)$ pada persamaan berikut.

$$H = g(W \cdot X + b) \quad (1)$$

Setelah matriks keluaran lapisan tersembunyi (H) terbentuk, langkah selanjutnya adalah memetakannya menuju target prediksi (y) dengan persamaan linear berikut.

$$H \beta = y \quad (2)$$

Di mana β merupakan matriks bobot keluaran (*output weights*) yang menghubungkan lapisan tersembunyi dengan dengan hasil akhir. Untuk mencari nilai β ini, ELM tidak melakukan perulangan komputasi, melainkan langsung menghitungnya secara matematis menggunakan rumus invers semu (*pseudo-inverse*) Moore-Penrose [26].

b. Optimasi Arsitektur dengan *Particle Swarm Optimization* (PSO-ELM)

Kelemahan ELM dasar adalah hasil prediksinya sangat bergantung pada nilai acak di awal. Untuk mengatasinya, penelitian ini menerapkan algoritma *Particle Swarm Optimization* (PSO) yang ditugaskan secara khusus untuk mencari pengaturan (*hyperparameter*) terbaik secara otomatis [6]. Diibaratkan seperti sekawanan burung yang sedang mencari titik makanan terbaik, setiap partikel di dalam PSO bertugas mencari kombinasi nilai paling pas antara jumlah neuron tersembunyi (n_{hidden}) dan parameter regularisasi penalti (C). Pencarian partikel dibatasi secara ketat, yakni rentang absolut $[5, 150]$ untuk jumlah neuron tersembunyi, dan rentang skala logaritmik $[10^{-1}, 10^7]$ untuk parameter penalti. Batasan ini ditetapkan agar model tetap akurat namun tidak memberatkan kinerja komputer.

Pencarian solusi ini dikendalikan oleh *hyperparameter* empiris PSO yang meliputi, batas iterasi maksimal ($I = 50$), ukuran populasi partikel ($N = 20$), koefisien percepatan kognitif (c_1) dan sosial (c_2) yang diseimbangkan pada nilai 1,5, serta bobot inersia dinamis (*decreasing inertia weight*) yang menyusut perlahan secara linear dari 0,9 hingga 0,4 untuk menyeimbangkan transisi antara fase eksplorasi (pencarian global) dan eksploitasi (pencarian lokal) [27]. Selama proses pencarian berlangsung, tingkat kecerdasan (*fitness*) setiap partikel dievaluasi berdasarkan persentase galatnya (MAPE). Agar model tidak sekadar menghafal data uji (*data leakage*), evaluasi ini dilakukan secara terpisah menggunakan 25% data dari himpunan data latih sebagai ruang validasi (*validation set*).

Mengingat data yang diproses merupakan data deret waktu (*time series*), pembagian 25% data validasi ini tidak dilakukan secara acak (*random split*). Pemisahan data validasi diterapkan secara ketat dengan mematuhi urutan waktu kronologis, yakni mengambil 25% porsi data pada bagian paling akhir (terbaru) dari himpunan data latih yang tersedia di setiap lipatan (*fold*). Pemisahan secara berurutan ini sangat krusial untuk mencegah model

mengintip informasi dari masa depan, sekaligus memastikan bahwa evaluasi kecerdasan algoritma PSO selalu diuji pada rentang tren pasar yang paling berdekatan dengan data uji sesungguhnya.

Setelah PSO berhasil menemukan pengaturan terbaiknya (nilai n_{hidden} dan C final), bobot keluaran (β) dari keseluruhan jaringan dihitung secara analitis menggunakan metode invers semu (*pseudo-inverse*) Moore-Penrose yang telah diregularisasi, atau yang secara algoritmik dikenal sebagai Regresi Ridge (*Ridge Regression*) melalui persamaan berikut.

$$\beta = \left(H^T H + \frac{I}{C} \right)^{-1} H^T y \quad (3)$$

Penggunaan *Ridge Regression* dalam arsitektur ELM ini berperan penting untuk mengintegrasikan parameter penalti (C) untuk menekan nilai bobot keluaran (H) sehingga menghasilkan model prediksi final yang jauh lebih stabil dan kebal terhadap risiko *overfitting* [28]. Seluruh parameter teknis algoritma PSO dan ELM ditetapkan secara baku sebelum eksekusi, agar setiap perbandingan antar skenario dilakukan pada kondisi yang setara. Rincian nilai parameter tersebut disajikan pada Tabel 3.

Tabel 3. Konfigurasi Parameter PSO dan Arsitektur Pemodelan

Parameter	Nilai
Dimensi Partikel	2
n_{hidden}	5 – 150
$\log_{10} C$	(-1) – 7
Ukuran populasi (N)	20
Iterasi (I_{max})	50
Koefisien kognitif (c_1)	1,5
Koefisien sosial (c_2)	1,5
<i>Inertia weight</i> ($w_{max} - w_{min}$)	0,9 – 0,4
Fungsi aktivasi	<i>Sigmoid</i>
<i>Random Seed</i> ELM	42
Rasio validasi internal	25%
Target optimasi	MAPE (%)
Metode (β)	<i>Ridge regression</i>

Pengaturan pada Tabel 3 tersebut dirancang secara seimbang. Pada arsitektur ELM, fungsi Sigmoid dipilih untuk memetakan pola data harga yang rumit. Selain itu, digunakan nilai seed yang dikunci secara tetap (42) guna memastikan bahwa perhitungan evaluasi algoritma berjalan konsisten dan dapat diulang (*reproducible*) pada tahap pengujian selanjutnya.

2.5. Validasi dan Evaluasi Kinerja Model

Untuk memastikan tidak terjadinya kebocoran informasi (*data leakage*) dari periode waktu selanjutnya ke dalam data historis, kemampuan prediksi model dievaluasi secara ketat menggunakan metode *Walk-Forward Validation* dengan skema *expanding window*. Model dilatih secara bertahap dengan terus memperluas rentang data historis (t) untuk memprediksi harga pada periode selanjutnya ($t + 1$) [6]. Secara keseluruhan, data dibagi secara berurutan menjadi lima bagian (lipatan) pengujian. Empat lipatan (*fold*) pertama memiliki ukuran dasar 314 sampel, sedangkan sisa data dialokasikan pada lipatan (*fold*) kelima sebanyak 316 sampel. Rincian pembagian data ini disajikan pada Tabel 4.

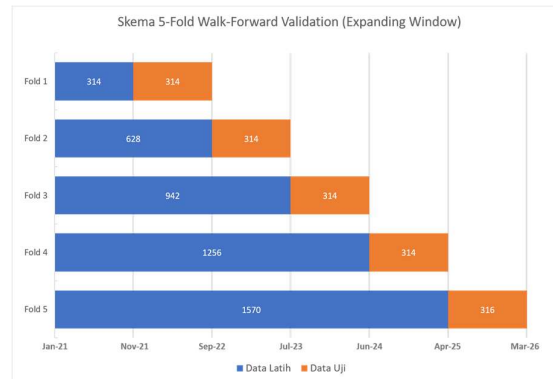
Tabel 4. Skema 5-Fold Walk-Forward Validation (*Expanding Window*)

<i>Fold</i>	Data Latih	Data Uji
1	Baris 1 - 314	Baris 315 - 628
2	Baris 1 - 628	Baris 629 - 942
3	Baris 1 - 942	Baris 943 - 1256
4	Baris 1 - 1256	Baris 1257 - 1570
5	Baris 1 - 1570	Baris 1571 - 1886

Melalui pembagian ini, algoritma dilatih menggunakan ukuran data historis yang terus bertambah untuk memprediksi data uji pada periode berikutnya. Pendekatan yang digambarkan pada Gambar 2 ini sangat menyerupai kondisi peramalan di dunia nyata. Model dibatasi agar murni belajar dari riwayat data sebelumnya, tanpa melibatkan informasi apa pun dari periode yang akan diprediksi.

Gambar 2 secara jelas memperlihatkan bagaimana ukuran data pengujian selalu bergeser maju, seiring dengan bertambahnya jumlah data yang digunakan untuk pelatihan. Secara teknis, pada setiap lipatan (*fold*),

himpunan data latih dipartisi secara kronologis menjadi 75% data pelatihan murni dan 25% data validasi internal. Algoritma PSO kemudian dijalankan ulang dari awal guna mencari *hyperparameter* terbaik pada rentang data tersebut. Setelah parameter dikunci, model ELM dilatih ulang secara penuh, lalu dievaluasi kinerjanya pada data uji (*test set*) yang rentang waktunya berada tepat setelah data latih. Siklus pencarian ulang PSO dan pelatihan model ini diulang secara independen pada kelima lipatan agar model selalu beradaptasi dengan perubahan kondisi pasar terkini.



Gambar 2. Diagram Skema *Walk-Forward Validation 5-Fold*

Luaran prediksi kemudian dikembalikan ke dalam skala harga aktual melalui proses denormalisasi (*inverse transform*). Kinerja model kemudian diukur menggunakan tiga metrik evaluasi untuk melihat selisih antara nilai aktual (y_i) dan nilai tebakan prediksi ($\hat{y}_i$), yaitu *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE) [29]. Metrik pertama adalah *Mean Absolute Error* (MAE) yang mengukur rata-rata selisih absolut antara nilai prediksi dan aktual, dirumuskan sebagai berikut.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

Metrik kedua adalah *Root Mean Squared Error* (RMSE) yang memberikan penalti atau bobot lebih pada galat yang memiliki nilai simpangan ekstrem, dengan formulasi.

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

Serta metrik ketiga berupa *Mean Absolute Percentage Error* (MAPE) yang menyajikan persentase galat relatif agar lebih proporsional terhadap skala harga, melalui persamaan.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (6)$$

Dari ketiga metrik tersebut, nilai persentase kesalahan (MAPE) dijadikan sebagai tolok ukur utama (*fitness function*) yang memandu algoritma PSO dalam mencari parameter terbaiknya. Sebagai tahap akhir penelitian, arsitektur model beserta kombinasi fitur yang menghasilkan stabilitas tertinggi dan persentase kesalahan terendah akan ditetapkan sebagai model prediksi final.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Ekstraksi Fitur dan Prapemrosesan

Tahap awal eksperimen ini telah berhasil mengolah data mentah menjadi sekumpulan indikator teknikal untuk membaca momentum, tren, dan fluktuasi harga pasar. Seluruh data tersebut kemudian disusun ke dalam format yang terstruktur agar dapat diproses secara langsung oleh model jaringan saraf tiruan. Sebagai gambaran mengenai wujud data yang sudah siap dilatih ini, hasil perhitungan pada lima hari pertama setelah masa pemanasan (*warm-up period*) disajikan pada Tabel 5.

Tabel 5. Hasil Kalkulasi Fitur Indikator Teknikal (Representasi Awal Periode)

Tanggal	Close (USD)	RSI	MACD	SMA (10)	SMA (30)	BB Up	BB Low
2021-01-30	34,269.52	44.29	-162.41	32,560.93	34,816.56	39,061.36	29,803.14
2021-01-31	33,114.36	42.35	-183.93	32,789.79	34,941.23	38,942.14	29,677.13
2021-02-01	33,537.18	40.94	-164.96	32,842.94	34,988.23	38,932.83	29,647.86
2021-02-02	35,510.29	48.49	9.18	33,187.20	35,079.17	38,661.42	29,738.66
2021-02-03	37,472.09	54.83	302.01	33,705.47	35,262.51	38,224.08	30,004.48

Berdasarkan Tabel 5, terlihat adanya perbedaan rentang angka yang sangat besar antar fitur indikator. Terdapat perbedaan yang mencolok antara harga penutupan (*Close*) yang bernilai puluhan ribu dolar, nilai RSI yang hanya berkisar pada rentang 0 hingga 100, serta nilai MACD yang bisa bergerak negatif maupun positif tanpa batasan yang pasti. Jika dibiarkan dalam wujud aslinya, atribut dengan angka yang sangat besar (seperti harga Bitcoin) akan mendominasi perhitungan bobot matriks pada *Extreme Learning Machine* (ELM) dan menutupi pengaruh indikator lain yang bernilai kecil. Kondisi ini dapat menyebabkan fungsi aktivasi di dalam model kebingungan dan gagal memproses pola data dengan baik.

Oleh karena itu, untuk memastikan setiap indikator dianggap sama pentingnya dan mencegah gagalanya perhitungan di dalam komputer, seluruh angka pada matriks fitur dan target prediksi diubah skalanya (*scaling*) menjadi rentang antara 0 hingga 1 menggunakan metode *Min-Max Scaler*. Parameter skala ini hanya dihitung berdasarkan himpunan data latih (*training set*) pada setiap tahapan lipatan (*fold*), yang kemudian diterapkan untuk mengubah rentang data uji (*test set*). Eksekusi dinamis ini sangat krusial guna memastikan tidak adanya kebocoran informasi (*data leakage*) dari masa depan. Setelah seluruh data diseragamkan dan siap digunakan, tahap selanjutnya adalah menguji matriks data tersebut ke dalam model ELM dasar untuk mengevaluasi kinerja awalnya.

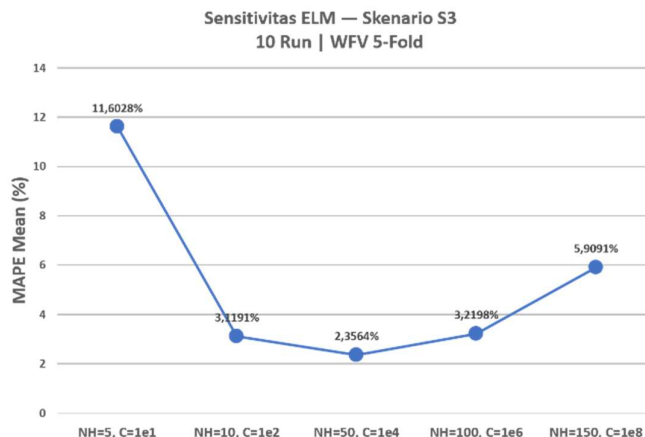
3.2. Analisis Kinerja Arsitektur *Baseline Extreme Learning Machine* (ELM)

Pengujian pertama melakukan uji sensitivitas untuk melihat bagaimana reaksi model *Extreme Learning Machine* (ELM) dasar jika pengaturan parameternya diubah-ubah. Pengujian ini bertujuan untuk membuktikan secara langsung bahwa menentukan parameter model dengan cara coba-coba (*trial and error*) berisiko besar menurunkan tingkat akurasi prediksi. Eksperimen dilakukan menggunakan data Skenario S3 dengan menguji lima kombinasi antara jumlah neuron pada *hidden layer* (n_{hidden}) dan nilai penalti regularisasi (C). Kombinasi ini dimulai dari jumlah neuron yang sangat sedikit hingga sangat banyak. Agar hasilnya objektif, setiap kombinasi diuji sebanyak 10 putaran menggunakan skema *Walk-Forward Validation 5-Fold* untuk mendapatkan rata-rata persentase kesalahannya.

Tabel 6. Hasil Pengujian Sensitivitas *Hyperparameter* ELM

n_{hidden}	C	MAE (USD)	RMSE (USD)	MAPE (%)
5	10^1	5.020,52	5.956,22	11,6028
10	10^2	1.716,18	2.224,95	3,1191
50	10^4	1.279,79	1.731,09	2,3564
100	10^6	1.561,89	2.069,66	3,2198
150	10^8	2.695,83	3.688,33	5,9091

Tabel 6 memperlihatkan bahwa kinerja model ELM sangat peka dan tidak stabil terhadap perubahan pengaturan. Menambah jumlah neuron tidak selalu membuat prediksi model menjadi lebih akurat. Pada pengaturan paling kecil ($n_{hidden} = 5$), model mengalami *underfitting* parah dengan tingkat kesalahan mencolok sebesar 11,6028%. Hal ini terjadi karena jaringan tidak memiliki kapasitas memori yang cukup untuk membaca pola perubahan harga Bitcoin yang rumit. Kinerja model sempat membaik saat arsitektur diperbesar hingga titik menengah ($n_{hidden} = 50$), di mana model secara kebetulan menemukan keseimbangan kapasitas dengan nilai MAPE 2,3564%. Namun, ketika jumlah neuron terus ditambah hingga batas tertinggi ($n_{hidden} = 150$), performanya justru kembali memburuk secara drastis, dan tingkat kesalahannya melonjak menjadi 5,9091%.



Gambar 3. Grafik Uji Sensitivitas ELM

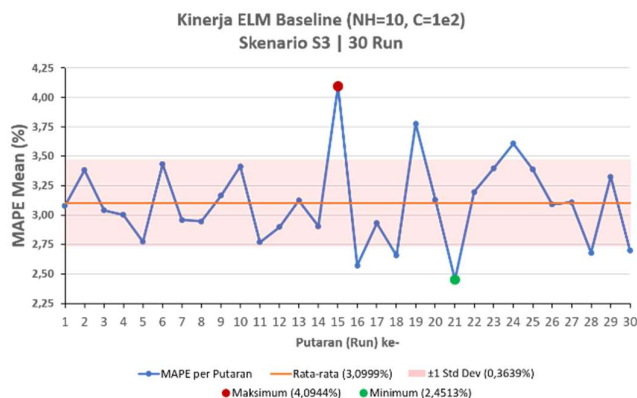
Penurunan performa yang membentuk kurva menyerupai huruf U (*U-Curve*) pada Gambar 3 ini terjadi karena adanya kegagalan perhitungan matematis di dalam komputer. Penggunaan ratusan neuron yang nilainya diacak menyebabkan matriks pembobotan (*weight matrix*), yakni matriks yang digunakan untuk mengalkulasi nilai bobot akhir antara lapisan tersembunyi dan lapisan keluaran, menjadi tidak stabil. Di saat yang sama, pemberian nilai penalti C yang terlalu besar gagal meredam data yang bising (*noise*). Akibatnya, model justru menghafal pola yang salah (*overfitting*). Fluktuasi ini menjadi bukti mengenai tingginya kompleksitas ruang pencarian *hyperparameter* ELM, sehingga diperlukan penerapan algoritma *Particle Swarm Optimization* (PSO) untuk mengotomatisasi pengaturan arsitektur jaringan secara adaptif.

Selain menguji sensitivitas, kinerja ELM dasar juga dievaluasi untuk melihat seberapa stabil model ini jika digunakan secara praktis. Pengaturan arsitektur sengaja dikunci pada nilai standar ($n_{hidden} = 10, C = 10^2$) tanpa bantuan optimasi otomatis. Untuk melihat dampak dari kelemahan bawaan ELM (yakni pengacakan bobot di awal), model dijalankan ulang sebanyak 30 kali putaran. Jumlah putaran ini dipilih untuk memastikan hasil evaluasi statistiknya sah dan dapat diandalkan.

Tabel 7. Statistik Kinerja ELM *Baseline* ($NH = 10, C = 10^2$) — Skenario S3, 30 Run

Statistik	MAE (USD)	RMSE (USD)	MAPE (%)
<i>Mean</i>	1.674,38	2.175,30	3,0999
<i>Std Dev</i>	202,45	236,10	0,3639
<i>Min</i>	1.310,92	1.754,26	2,4513
<i>Max</i>	2.206,22	2.801,01	4,0944

Merujuk pada parameter statistik di Tabel 7, kinerja model ELM dasar (*baseline*) menunjukkan tingkat kesalahan prediksi (MAPE) rata-rata sebesar 3,0999%. Tingkat kesalahan ini tidak stabil, yang dibuktikan secara spesifik oleh selisih yang signifikan antara nilai minimum dan maksimum, serta nilai standar deviasi yang tercatat pada tabel. Pada putaran pengujian yang secara acak mendapatkan inisialisasi bobot awal (*random weight initialization*) yang optimal, model mampu menyentuh tingkat kesalahan terendah sebesar 2,4513%. Tahapan inisialisasi ini merupakan mekanisme bawaan ELM di mana algoritma secara acak menentukan nilai parameter masukan pada lapisan tersembunyi. Sebaliknya, pada putaran yang menghasilkan matriks bobot acak yang kurang ideal, tingkat kesalahan meningkat drastis hingga menyentuh batas tertinggi sebesar 4,0944%.



Gambar 4. Fluktuasi Kinerja 30 Putaran ELM *Baseline*

Ketidakstabilan hasil prediksi ini terekam sangat jelas pada Gambar 4. Nilai simpangan baku sebesar 0,3639% membuktikan bahwa performa ELM konvensional sangat bergantung pada keberuntungan nilai acak di awal. Tingginya ketidakpastian dan ketiadaan jaminan hasil yang konsisten inilah yang menjadi alasan terkuat mengapa model ini wajib diperbaiki menggunakan algoritma optimasi (PSO-ELM).

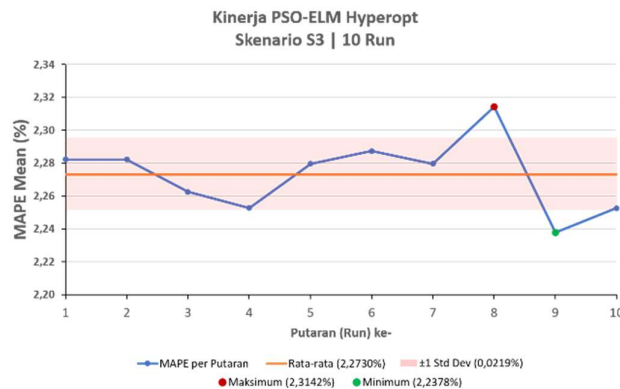
3.3. Optimasi Jaringan Menggunakan PSO-ELM

Setelah mengetahui bahwa ELM dasar memiliki kelemahan pada ketidakstabilan nilai acak di awal, algoritma *Particle Swarm Optimization* (PSO) diterapkan untuk mencari susunan parameter yang paling tepat secara otomatis. Model PSO-ELM dijalankan sebanyak 10 kali menggunakan nilai awal (*seed*) yang berbeda-beda untuk memastikan keandalannya. Pada setiap putaran, PSO mencari pasangan parameter terbaik untuk masing-masing bagian pengujian (*fold*). Berdasarkan Tabel 8, rata-rata persentase kesalahan (MAPE) dari 10 putaran tersebut

sangat presisi, yaitu di angka 2,2730%. Selain itu, jarak antara tingkat kesalahan terendah (2,2378%) dan tertinggi (2,3142%) sangat sempit, sehingga menghasilkan nilai simpangan baku yang nyaris nol (0,0219%).

Tabel 8. Statistik Kinerja PSO-ELM Hyperopt — Skenario S3, 10 Run

Statistik	MAE (USD)	RMSE (USD)	MAPE (%)
<i>Mean</i>	1.284,96	1.739,31	2,2730
<i>Std Dev</i>	16,05	19,60	0,0219
<i>Min</i>	1.272,93	1.720,50	2,2378
<i>Max</i>	1.328,98	1.791,96	2,3142



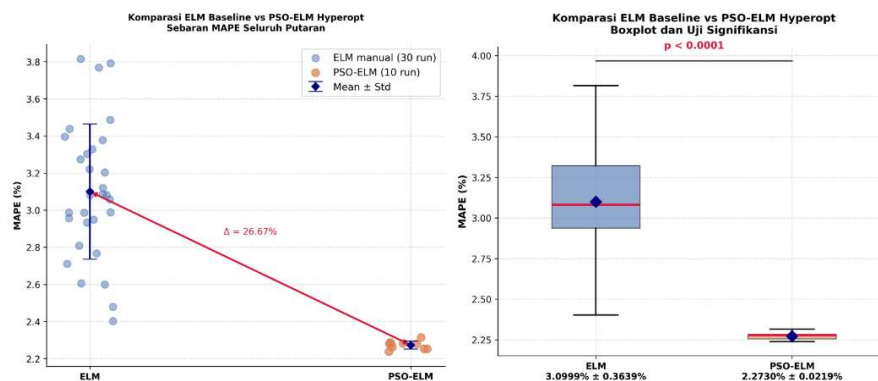
Gambar 5. Stabilitas Kinerja 10 Putaran PSO-ELM

Kestabilan kinerja ini terlihat jelas pada Gambar 5, di mana lintasan grafik nilai MAPE sangat datar. Hasil ini membuktikan bahwa algoritma PSO berhasil mengatasi masalah utama pada ELM dasar. PSO terbukti mampu menemukan kombinasi jumlah neuron (n_{hidden}) dan nilai penalti regularisasi (C) yang membuat kinerja model jaringan menjadi jauh lebih kokoh (*robust*) dan tidak lagi bergantung pada faktor keberuntungan nilai acak di awal. Untuk memastikan bahwa peningkatan akurasi ini benar-benar berarti secara statistik dan bukan sekadar kebetulan, hasil perhitungan model ELM dasar (30 putaran) dibandingkan secara langsung dengan PSO-ELM (10 putaran).

Tabel 9. Komparasi Komprehensif ELM *Baseline* vs PSO-ELM — Skenario S3

Metrik	ELM <i>Baseline</i>	PSO-ELM	Δ MAPE (%)
MAE <i>Mean</i> (USD)	1.674,38	1.284,96	-23,26%
RMSE <i>Mean</i> (USD)	2.175,30	1.739,31	-20,04%
MAPE <i>Mean</i> (%)	3,0999	2,2730	-26,67%
MAPE <i>Std Dev</i> (%)	0,3639	0,0219	-94,00%

Berdasarkan Tabel 9, penggunaan PSO mampu menekan rata-rata tingkat kesalahan prediksi (MAPE) hingga 26,67% (dari 3,0999% turun menjadi 2,2730%), diiringi penurunan galat nominal pada MAE (-23,26%) dan RMSE (-20,04%). Namun, temuan yang paling menonjol adalah penyusutan drastis pada nilai simpangan baku (*Std Dev*) sebesar 94,00%, yang berarti model kini menjadi jauh lebih stabil.



Gambar 6. Sebaran Titik Kesalahan Prediksi (Kiri) dan Uji Signifikansi *Boxplot* (Kanan)

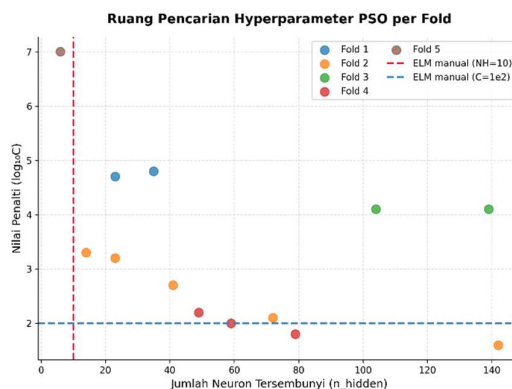
Perbedaan tingkat kestabilan ini digambarkan dengan jelas pada Gambar 6. Pada grafik sebaran (kiri), titik-titik kesalahan prediksi PSO-ELM berkumpul sangat rapat di bagian bawah, meredam fluktuasi liar yang sebelumnya terjadi pada model ELM biasa. Diagram *boxplot* (kanan) juga menegaskan bahwa model yang dioptimasi tidak lagi menghasilkan tebakan yang melenceng terlalu jauh (*outlier*). Untuk memastikan keabsahan perbedaan tersebut, dilakukan pengujian signifikansi secara statistik. Dasar pengujian menggunakan metode non-parametrik *Mann-Whitney U Test*, mengingat distribusi galat peramalan tidak selalu mengikuti asumsi kurva normal. Pengujian ini membandingkan ukuran sampel evaluasi independen antara model ELM dasar ($N = 30$) dan PSO-ELM ($N = 10$). Hasil komputasi mencatatkan nilai probabilitas statistik sebesar $p < 0,0001$. Karena nilai p tersebut berada sangat jauh di bawah ambang batas signifikansi konvensional ($\alpha = 0,05$), maka terdapat perbedaan yang sangat nyata secara keilmuan.

Pencapaian ini, yang didukung pula oleh skor Cohen's d sebesar 3,2074 (*large effect*), secara meyakinkan membuktikan bahwa penambahan algoritma PSO membuat prediksi harga Bitcoin menjadi lebih akurat, stabil, dan dapat diandalkan secara ilmiah. Salah satu kelebihan utama PSO-ELM dibandingkan ELM konvensional adalah kemampuannya untuk mengubah-ubah jumlah neuron dan nilai penalti secara otomatis, menyesuaikan dengan jumlah data latih yang sedang diproses. Rekapitulasi parameter yang dipilih oleh PSO pada setiap lipatan pengujian (*fold*) disajikan pada Tabel 10.

Tabel 10. *Hyperparameter* yang Dipilih PSO per *Fold* — Skenario S3 (10 Run)

<i>Fold</i>	n_{hidden} (Mean $\pm$ Std)	$\log_{10}C$ (Mean $\pm$ Std)	C Efektif (approx.)
F1	33,8 $\pm$ 3,8	4,79 $\pm$ 0,04	$10^{4,79}$
F2	62,9 $\pm$ 45,4	2,46 $\pm$ 0,60	$10^{2,46}$
F3	128,5 $\pm$ 16,9	4,09 $\pm$ 0,00	$10^{4,09}$
F4	53,0 $\pm$ 9,7	2,13 $\pm$ 0,15	$10^{2,13}$
F5	6,0 $\pm$ 0,0	7,00 $\pm$ 0,00	$10^{7,00}$

Berbeda dengan ELM *baseline* yang dikunci pada konfigurasi statis, PSO-ELM memiliki kemampuan beradaptasi yang baik. Algoritma secara mandiri mengekskansi kapasitas memori jaringan, bergeser dari rata-rata 33,8 neuron di tahap awal (F1) menjadi 128,5 neuron (F3) seiring dengan bertambahnya porsi data latih.



Gambar 7. Ruang Pencarian *Hyperparameter* Adaptif PSO per Lipatan (*Fold*)

Pola pencarian parameter ini divisualisasikan pada Gambar 7, yang juga menyoroti sebuah keputusan pencegahan penting oleh sistem pada lipatan pengujian akhir (F5). Saat dihadapkan pada *volume* data latih maksimal (1.570 sampel), kawanan PSO memangkas jumlah neuron secara drastis ke batas minimum (6 neuron) dan serentak memaksimalkan nilai penalti regularisasi. Tindakan penyesuaian otomatis ini membuktikan bahwa algoritma tidak hanya sekadar mencari nilai kesalahan terendah, tetapi juga berusaha menjaga keseimbangan arsitektur agar model tidak hanya sekadar menghafal data baru (*overfitting*).

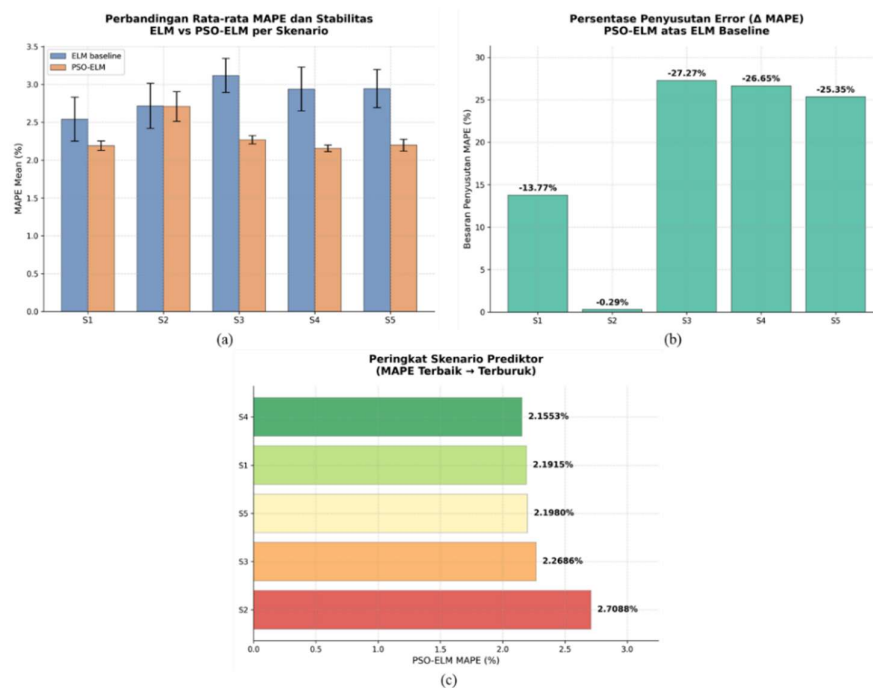
3.4. Evaluasi Komparatif Lintas Indikator Teknikal

Setelah model PSO-ELM terbukti jauh lebih stabil dari ELM dasar, tahap selanjutnya adalah mengujinya pada kelima skenario fitur input, yakni variasi kombinasi antara data harga mentah dan indikator teknikal, sebagaimana yang telah diuraikan pada Tabel 2 di Subbab 2.3 (Prapemrosesan dan Rekayasa Fitur). Tujuannya adalah untuk melihat fitur data masukan mana yang paling efektif dalam memandu arah prediksi. Hasil komputasi dari kelima fitur input tersebut direkapitulasi pada Tabel 11. Seluruh hasil komputasi dari kelima fitur input tersebut diurutkan berdasarkan nilai rata-rata MAPE (%) PSO-ELM dari terbaik ke terburuk.

Tabel 11. Hasil Evaluasi Komprehensif ELM *Baseline* vs PSO-ELM pada 5 Skenario (10 *Run*)

Peringkat	Skenario	Indikator	ELM MAPE (%)	PSO MAPE (%)	Δ MAPE (%)
1	S4	<i>Close</i> + SMA	$2,9384 \pm 0,2896$	$2,1553 \pm 0,0443$	-26,65%
2	S1	Harga Historis	$2,5413 \pm 0,2907$	$2,1915 \pm 0,0632$	-13,77%
3	S5	<i>Close</i> + <i>Bollinger Bands</i>	$2,9446 \pm 0,2527$	$2,1980 \pm 0,0777$	-25,35%
4	S3	<i>Close</i> + RSI + MACD	$3,1191 \pm 0,2242$	$2,2686 \pm 0,0537$	-27,27%
5	S2	OHLC + <i>Volume</i>	$2,7167 \pm 0,2977$	$2,7088 \pm 0,1960$	-0,29%
Rata-rata			2,8520	2,3044	-18,67%

Tabel 11 memaparkan secara jelas keunggulan PSO-ELM dalam mengurangi tingkat kesalahan (MAPE) di berbagai konfigurasi fitur. Secara keseluruhan, integrasi algoritma ini sukses mencatatkan rata-rata penurunan tingkat kesalahan sebesar 18,67% melintasi kelima skenario pengujian tersebut. Skenario S4 (*Close* + SMA) secara meyakinkan tampil sebagai arsitektur terbaik dengan capaian akurasi akhir 2,1553%. Algoritma PSO mencatatkan pengurangan *error* yang sangat besar ketika mengoptimasi fitur indikator teknikal turunan, terbukti dari tingginya persentase penyusutan pada S3 (-27,27%), S4 (-26,65%), dan S5 (-25,35%). Sebaliknya, Skenario S2 (OHLC + *Volume*) terbukti menjadi data yang paling sulit dioptimasi, dengan tingkat penurunan kesalahan yang sangat kecil -0,29%.

**Gambar 8.** Evaluasi Lintas Skenario: (a) Perbandingan MAPE & Simpangan Baku; (b) Reduksi Kesalahan (Δ MAPE); (c) Peringkat Akurasi Final

Ketiga grafik pada Gambar 8 merangkum perbandingan efektivitas model secara keseluruhan. Grafik pada panel (a) membuktikan bahwa algoritma PSO berhasil memperkecil rentang simpangan baku (*error bar*) secara signifikan. Hal ini menunjukkan bahwa optimasi tersebut sukses mengatasi masalah ketidakstabilan tebakan awal pada hampir semua skenario. Selanjutnya, panel (b) memperlihatkan penurunan tingkat kesalahan yang besar (sekitar 25%–27%) pada skenario yang menggunakan indikator turunan (S3, S4, dan S5). Menariknya, Skenario S2 (OHLC dan *Volume*) justru menjadi kombinasi yang paling sulit dioptimasi karena tingkat kesalahannya hanya turun 0,29%. Hal ini menandakan bahwa gabungan data harga mentah mengandung terlalu banyak gejala fluktuasi (*noise*), sehingga algoritma sekuat apa pun tetap kesulitan untuk menemukan pola arah harga yang jelas.

Sementara itu, grafik peringkat pada panel (c) memberikan temuan penting terkait pemilihan indikator. Skenario S4 (Tren SMA) terbukti keluar sebagai model yang paling unggul. Hal ini mengonfirmasi bahwa teknik penghalusan data (*smoothing*) pada SMA mampu menghasilkan pola tren yang lebih bersih dan mudah dipelajari oleh model. Penemuan yang cukup mengejutkan justru terlihat pada Skenario S1 (harga masa lalu), karena ternyata mampu mengalahkan skenario yang lebih rumit seperti S3 (gabungan RSI dan MACD). Kekalahan Skenario S3 ini membuktikan adanya informasi yang tumpang tindih (*redundant*). Indikator RSI dan MACD memiliki pola yang

terlalu mirip dengan data harga masa lalu, sehingga hanya menambah beban hitungan komputer tanpa memberikan informasi baru yang berguna. Hasil ini menegaskan bahwa asal menambahkan banyak indikator belum tentu meningkatkan akurasi, sekaligus memastikan bahwa model PSO-ELM dengan Skenario S4 adalah kombinasi yang paling optimal.

Merujuk pada hasil evaluasi komparatif lintas indikator, kombinasi arsitektur PSO-ELM dengan Skenario S4 (Tren SMA) secara definitif ditetapkan sebagai model terbaik (*champion model*) berkat capaian stabilitas persentase kesalahan terendah (rata-rata MAPE 2,1553% dari 10 putaran). Untuk membedah bagaimana model unggulan ini menyesuaikan diri seiring berjalannya waktu pengujian. Rincian penyesuaian parameter pada masing-masing tahapan pengujian (*Walk-Forward Validation*) untuk satu putaran disajikan pada Tabel 12.

Tabel 12. Performa Model Terbaik (PSO-ELM S4) per Lipatan WFV

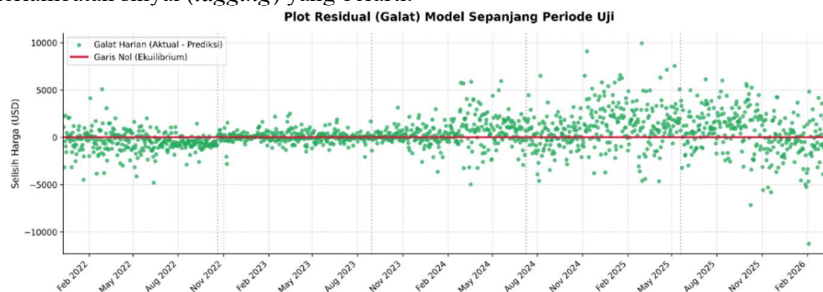
<i>Fold</i>	<i>n_hidden</i>	<i>C</i>	MAE (USD)	RMSE (USD)	MAPE (%)
F1	148	$10^{1,89}$	911,03	1.236,16	3,0725
F2	33	$10^{2,25}$	383,51	592,84	1,6072
F3	65	$10^{1,58}$	1.006,70	1.496,07	1,8939
F4	7	$10^{3,87}$	1.845,59	2.463,01	2,2158
F5	5	$10^{4,79}$	1.798,48	2.344,10	1,8720
Rata-Rata			1.189,06	1.626,43	2,1323

Tabel 12 memperlihatkan betapa reaktifnya algoritma PSO dalam menyesuaikan diri. Pada tahap pengujian awal (F1) saat data latih masih sedikit, tingkat kesalahan berada di angka 3,0725%. Namun, seiring dengan bertambahnya data latih di tahap kedua (F2), tingkat kesalahan langsung turun drastis menjadi 1,6072%. Penyesuaian yang paling penting terlihat pada tahap akhir (F4 dan F5). Saat model dihadapkan pada jumlah data terbesar dengan perubahan fluktuasi harga yang sangat tajam, PSO secara otomatis memangkas jumlah neuronnya menjadi sangat sedikit (7 dan 5 neuron) dan memperketat batas nilai penalti (*C*). Keputusan otomatis ini sukses menghindarkan model dari jebakan *overfitting*, sehingga akurasi tetap solid di kisaran 1,8%–2,2%. Sebagai pembuktian akhir, hasil tebakan prediksi dari model juara ini dibandingkan secara langsung dengan pergerakan harga aktual Bitcoin pada Gambar 9.



Gambar 9. Pergerakan Harga Aktual vs Hasil Prediksi Model Terbaik (PSO-ELM S4)

Grafik pada Gambar 9 memperlihatkan tingkat kecocokan yang sangat tinggi. Model tidak hanya sekadar menebak arah tren secara umum, tetapi benar-benar berhasil mengikuti lekuk pergerakan harga aslinya. Model ini secara tangguh mampu melacak anjlokannya harga yang parah di tahun 2022 (menuju level USD 20.000), serta sangat responsif mengikuti lonjakan rekor harga tertinggi (menembus USD 120.000) di tahun 2024–2025 tanpa mengalami keterlambatan sinyal (*lagging*) yang berarti.



Gambar 10. Plot Residual (Harga Aktual – Prediksi) Model Terbaik Sepanjang Periode Uji

Ketepatan prediksi ini juga diperkuat oleh grafik sebaran selisih kesalahan (residual) pada Gambar 10. Titik-titik selisih harian terlihat tersebar secara seimbang di atas dan di bawah garis nol. Meskipun sebaran titiknya sedikit melebar pada periode 2024–2025 akibat lonjakan *volume* transaksi, tidak ada pola penumpukan tebakan di satu sisi saja. Ketidadaan pola berat sebelah (bias) ini membuktikan bahwa selisih tebakan yang tersisa murni merupakan fluktuasi acak yang memang wajar terjadi di pasar kripto. Hal ini mengonfirmasi bahwa arsitektur PSO-ELM Skenario S4 telah berhasil menyerap dan mempelajari seluruh pola penting dari pergerakan harga Bitcoin secara optimal.

3.5. Diskusi dan Komparasi Hasil

Untuk mengevaluasi performa yang dihasilkan, temuan pada penelitian ini dikomparasikan dengan hasil eksperimen pada studi terdahulu, yakni Hiskiawan (2025) [4], Dhuhita (2026) [17], dan Parlika (2026) [23]. Sebagai persamaan, seluruh penelitian ini sama-sama membuktikan bahwa algoritma prediktif berbasis *machine learning* mampu menangkap dinamika pergerakan harga aset kripto. Namun, terdapat perbedaan hasil performa yang sangat kontras. Hasil pengujian pada penelitian sebelumnya menunjukkan bahwa arsitektur *deep learning* yang kompleks dan penumpukan indikator (seperti MACD dan RSI) justru menghasilkan beban komputasi yang berat [17], lambat merespons perubahan tren mendadak [4], serta rentan terhadap bias tren naik (*bullish*) yang menyebabkan anjloknya akurasi prediksi arah menjadi hanya 51,08% saat pasar sedang turun [23].

Sebaliknya, keunggulan dan kebaruan (*novelty*) dari penelitian ini dibuktikan secara empiris melalui pencapaian tingkat kesalahan peramalan yang sangat rendah, yakni MAPE 2,15%. Hasil ini membuktikan bahwa alih-alih menumpuk indikator multivariabel yang memicu redundansi dimensi, penggunaan *Simple Moving Average* (SMA) sebagai penyaring tunggal terbukti jauh lebih efektif dalam meredam gejolak harga. Selain itu, optimasi PSO-ELM membuat jaringan lebih stabil dalam menangkap tren tanpa terjebak pada bias arah tertentu. Implikasi dari hasil penelitian ini menegaskan bahwa untuk peramalan harga Bitcoin yang sangat bergejolak, penyederhanaan dimensi fitur dipadukan dengan algoritma komputasi yang ringan (PSO-ELM) mampu menghasilkan tingkat akurasi yang lebih unggul dibandingkan model *deep learning* konvensional yang rumit.

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa optimasi *hyperparameter* (n_{hidden} dan C) menggunakan *Particle Swarm Optimization* (PSO) terbukti mampu meningkatkan stabilitas dan akurasi arsitektur *Extreme Learning Machine* (ELM). Melalui pengujian *Walk-Forward Validation*, model PSO-ELM terbukti unggul dengan menekan rata-rata tingkat kesalahan (MAPE) sebesar 18,67% dibandingkan model ELM konvensional. Dari kelima skenario data yang diuji, Skenario S4 (gabungan harga historis dan Tren SMA) ditetapkan sebagai model peramalan terbaik dengan rekor penurunan tingkat kesalahan mencapai 26,65%. Melalui capaian tersebut, kontribusi penelitian ini adalah memberikan landasan empiris bahwa integrasi antara optimasi parameter otomatis dan teknik penyaring fluktuasi tren terbukti tangguh dan efektif dalam merancang arsitektur prediksi yang kebal terhadap gejolak ekstrem aset kripto. Meskipun memberikan hasil peramalan yang menjanjikan, penelitian ini memiliki batasan berupa penggunaan nilai *seed* yang dikunci tetap (*fixed seed*) sehingga membatasi eksplorasi tebakan awal jaringan, ketidadaan model pembandingan *Deep Learning*, dan jangkauan pengujian terbatas pada peramalan tunggal yang hanya memprediksi harga satu hari ke depan ($t + 1$). Oleh karena itu, riset selanjutnya sangat direkomendasikan untuk memperluas jangkauan prediksi menjadi beberapa langkah ke depan ($t + n$) dan mengujinya pada aset kripto lain guna memvalidasi ketangguhan prediksi model. Selain itu, perluasan dimensi pencarian PSO pada optimasi topologi fungsi aktivasi serta perbandingan kinerja secara langsung dengan algoritma *Deep Learning* mutakhir (seperti LSTM atau GRU) diharapkan dapat memberikan landasan empiris dan ide riset lanjutan yang lebih komprehensif.

DAFTAR PUSTAKA

- [1] J. Tarassov and N. Houlie, "Bitcoin: A life in crises," *PLoS One*, vol. 17, no. 9, pp. 1–16, 2022, doi: 10.1371/journal.pone.0274165.
- [2] D. Widyanti, Sudarmo, and T. Widiarihi, "Analisis volatilitas Bitcoin menggunakan model ARCH dan GARCH," *J. GAUSSIAN*, vol. 12, no. 2, pp. 254–265, 2023, doi: 10.14710/J.GAUSS.12.2.254-265.
- [3] K. Koszewski, S. Mazumdar, and A. S. Kumar, "Understanding rate of return dynamics of cryptocurrencies: An experimental campaign," *Artif. Intell. Rev.*, vol. 57, no. 8, pp. 1–29, 2024, doi: 10.1007/s10462-023-10629-7.
- [4] P. Hiskiawan, J. William, L. Feliepe, and T. Jansel, "A hybrid data science framework for forecasting Bitcoin prices using traditional and AI models," *J. Appl. Informatics Comput.*, vol. 9, no. 5, pp. 2089–2101, 2025.
- [5] R. F. T. Wulandari and D. Anubhakti, "Implementasi algoritma support vector machine (SVM) dalam memprediksi harga saham PT. Garuda Indonesia Tbk.," *IDEALIS Indones. J. Inf. Syst.*, vol. 4, no. 2, pp. 250–256, 2021.
- [6] R. Ilmiyah, D. Rachmatin, and R. Marwati, "Peramalan inflasi dengan metode particle swarm optimization – extreme

- learning machine," *INTERVAL J. Ilm. Mat.*, vol. 2, no. 1, pp. 42–51, 2022.
- [7] M. F. S, S. Annas, and N. Ismi, "Penerapan extreme learning machine (ELM) untuk meramalkan laju inflasi di Indonesia," *VARIANSI J. Stat. Its Appl. Teach. Res.*, vol. 6, no. 3, pp. 119–129, 2024, doi: 10.35580/variensium92.
- [8] I. Pratiwi, D. Bachtar, A. Jauhari, M. Yusuf, F. A. Mufarroha, and D. Rosa, "Extreme learning machine untuk memprediksi curah hujan dalam penentuan jadwal tanam padi," *JUSIFOR J. Sist. Inf. dan Inform.*, vol. 4, no. 1, pp. 34–42, 2024.
- [9] A. Kenioua, O. Djebili, and A. Touati, "Hybrid extreme learning machine for real-time rate of penetration prediction," *J. Pet. Explor. Prod. Technol.*, vol. 15, no. 139, pp. 1–21, 2025.
- [10] A. P. Ariyanti, M. I. Mazdadi, A. Farmadi, Muliadi, and R. Herteno, "Application of extreme learning machine method with particle swarm optimization to classify of heart disease," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 3, pp. 281–290, 2023.
- [11] P. Setiaji *et al.*, "Best model and performance of Bayesian regularization method for data prediction," *J. Phys. Conf. Ser.*, vol. 1933, no. 1, p. 012030, 2021, doi: 10.1088/1742-6596/1933/1/012030.
- [12] Q. Lin and M. Wang, "Temperature characteristics modeling for GaN PA based on PSO-ELM," *micromachines*, vol. 15, no. 1008, pp. 1–15, 2024.
- [13] N. Farhana, G. F. Hertono, and B. D. Handari, "Classification of coal mine pillar stability using extreme learning machine and particle swarm optimization adaptive klasifikasi kestabilan pilar tambang batu bara menggunakan extreme learning machine dan particle swarm optimization adaptive weight delay," *J. Tek. Inform.*, vol. 5, no. 6, pp. 1569–1580, 2024.
- [14] Z. Adel, "A comparative study between RSI and MACD to predict opportunities in cryptocurrency market from 2020 to 2022," *J. Econ. Sci. Inst.*, vol. 26, no. 2, pp. 195–214, 2023.
- [15] W. G. Mokodaser, H. P. Koapaha, and S. I. Adam, "Model random forest data historis multivariat untuk prediksi pendapatan asuransi," *IDEALIS Indones. J. Inf. Syst.*, vol. 8, no. 2, pp. 267–276, 2025.
- [16] T. Astutiningsih, D. R. S. Saputro, and Sutanto, "Optimasi algoritme xtreme gradient boosting (XGBoost) pada harga saham PT. United Tractors Tbk.," *SPECTA J. Technol.*, vol. 7, no. 3, pp. 632–641, 2023, doi: 10.35718/specta.v7i3.1031.
- [17] W. M. P. Dhuhita and Y. I. Sumbadri, "Analysis of LSTM-Adamax performance in Bitcoin price prediction using RSI & MACD indicators," *IJCSR Indones. J. Comput. Sci. Res.*, vol. 5, no. 1, pp. 68–78, 2026.
- [18] A. Baradja, Sukoco, and T. I. Tjendrowasono, "Penerapan machine learning untuk meningkatkan prediksi mata uang forex dengan indikator teknikal," *J. Pustaka AI*, vol. 2, no. 1, pp. 38–41, 2023.
- [19] M. M. R. Musyaffa, G. F. Hertono, and B. D. Handari, "Implementation of moving average filter in SARIMA-ANN and SARIMA-SVR methods for forecasting pneumonia incidence in Jakarta," *JAMBURA J. Biomath.*, vol. 6, no. 3, pp. 262–269, 2025.
- [20] T. Liu, "A comparative study of transformer-based and classical models for financial time-series forecasting," *J. Risk Financ. Manag.*, vol. 19, no. 203, pp. 1–16, 2026.
- [21] P. Yanida, L. N. Pratiwi, and W. P. Nugrahani, "Comparative evaluation of portfolio performance: A study of cryptocurrency, stock, and foreign exchange investments," *J. Ilmu Keuang. dan Perbank.*, vol. 13, no. 1, pp. 45–62, 2023.
- [22] E. M. C. Wattimena, P. D. M. Taihuttu, D. V. Waas, C. F. Palembang, and V. E. Pattiradjawane, "Optimasi model LSTM untuk prediksi curah hujan di Kota Ambon: Perbandingan mean imputation dan interpolasi dalam prediksi data time series," *Tensor Pure Appl. Math. J.*, vol. 6, no. 1, pp. 49–56, 2025.
- [23] R. Parluka, I. Asy'ari, and R. A. Ramadhan, "Predicting Bitcoin price trends using an LSTM model based on multi-variable technical indicators," *J. Serumpun Tek. Inform.*, vol. 1, no. 2, pp. 59–66, 2026.
- [24] P. J. Muhammad and Ali, "Investigating the impact of min-max data normalization on the regression performance of k-nearest neighbor with different similarity measurements," *ARO-The Sci. J. Koya Univ.*, vol. 10, no. 1, pp. 85–91, 2022, doi: 10.14500/aro.10955.
- [25] X. Liu, G. Zhou, Y. Zhou, and Q. Luo, "Functional extreme learning machine," *Front. Comput. Neurosci.*, vol. 17, p. 1209372, 2023, doi: 10.3389/fncom.2023.1209372.
- [26] K. C. Pradhisa and R. Yotenka, "Penerapan metode extreme learning machine (ELM) untuk meramalkan produksi air minum dan air yang dijual di Kota Yogyakarta tahun 2023," *Emerg. Stat. Data Sci. J.*, vol. 2, no. 2, pp. 258–263, 2024.
- [27] X. Fang *et al.*, "Equipment remaining useful life prediction algorithm based on hyperparameter optimization and LSTM," *AIP Adv.*, vol. 16, no. 4, p. 045013, 2026, doi: 10.1063/5.0327489.
- [28] T. Nagamani, G. V. Gopal, G. Lakshmi, K. V. S. S. Ramakrishna, N. Srija, and A. Gopi, "Improving model robustness against multicollinearity with a novel statistical regularized extreme learning algorithm," *IAENG Int. J. Comput. Sci.*, vol. 52, no. 11, pp. 4301–4309, 2025.
- [29] S. Aisyah, N. Ulinuha, and A. Hamid, "Penerapan extreme learning machine dalam meramalkan harga minyak sawit mentah," *KUBIK J. Publ. Ilm. Mat.*, vol. 7, no. 2, pp. 97–105, 2022.