Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

Pemodelan Prediksi Harga Ethereum (Atribut: Open, High dan Low) dengan Algoritma Extreme Learning Machine

Kasliono[1*], Niken Candraningrum[2], Kartika Sari[1]

[1] Fakultas MIPA, Program Studi Sistem Komputer, Universitas Tanjungpura, Pontianak, Indonesia
[2] Fakultas Teknik, Program Studi Teknik Informatika, Universitas Tanjungpura, Pontianak, Indonesia
Email: [1,*]kasliono@siskom.untan.ac.id, [2]nikenc@informatika.untan.ac.id, [3]kartika.sari@siskom.untan.ac.id
Email Penulis Korespondensi: kasliono@siskom.untan.ac.id
Submitted: **03/06/2023**; Accepted: **27/06/2023**; Published**: 29/06/2023**

Abstrak−Harga cryptocurrency seperti Ethereum sering mengalami fluktuasi yang tinggi dan sulit untuk diprediksi. Penelitian ini bertujuan untuk memprediksi harga Ethereum menggunakan algoritma Extreme Learning Machine (ELM) yang merupakan metode pembelajaran mesin yang cepat dan efisien. Data harga Ethereum dikumpulkan dari CoinMarketCap dengan scraping data menggunakan CoinmarketCap Scraper dari library cryptocmd menggunakan python. Sebuah model ELM dibangun dengan mengganti jumlah dari hidden node untuk menentukan model prediksi yang optimum dari harga Ethereum berdasarkan rata-rata MAPE terkecil. Performa model dievaluasi dengan menggunakan mean absolute percentage error (MAPE) pada data test set. Hasil penelitian menunjukkan bahwa model ELM yang dibangun dapat memprediksi harga Ethereum dengan akurasi 96.96%. MAPE yang diperoleh adalah sebesar 3.035334%, dengan 9 hidden node pada model arsitektur jaringan ELM yang dibangun. Hal ini menunjukkan bahwa model dapat menjelaskan sekitar 96.96% variasi data harga Ethereum. Oleh karena itu, model ELM dapat digunakan sebagai alat bantu dalam pengambilan keputusan investasi.

Kata Kunci: Blockchain; Cryptocurrency; ELM; Ethereum; ETH; Extreme Learning Machine; Prediksi

Abstract−The price of cryptocurrencies such as Ethereum often experiences high fluctuations and is difficult to predict. This study aims to predict Ethereum prices using the Extreme Learning Machine (ELM) algorithm which is a fast and efficient machine learning method. Ethereum price data is collected from CoinMarketCap by scraping the data using CoinmarketCap Scraper from the cryptocmd library using Python. An ELM model is built by changing the number of hidden nodes to determine the optimal prediction model of Ethereum prices based on the smallest average MAPE. Model performance was evaluated using the mean absolute percentage error (MAPE) on the test data set. The results show that the ELM model built can predict Ethereum prices with an accuracy of 96.96%. The MAPE obtained is 3.035334%, with 9 hidden nodes in the ELM network architecture model that was built. This shows that the model can explain about 96.96% of the variation in Ethereum price data. Therefore, the ELM model can be used as an aid in making investment decisions.

Keywords: Blockchain; Cryptocurrency; ELM; Ethereum; ETH; Extreme Learning Machine; Prediction

1. PENDAHULUAN

Saat ini, kehidupan kita dipengaruhi oleh berbagai aplikasi karena kemajuan yang semakin pesat dalam teknologi komunikasi dan informasi. Kepentingan teknologi informasi dan komunikasi tidak hanya dirasakan dibidang pendidikan; kebutuhan akan alat bantu yang representatif, efisien, dan efektif untuk menyelesaikan masalah yang terkendala oleh teknologi ini juga dirasakan diberbagai bidang kehidupan lainnya. Dengan demikian, teknologi komunikasi dan informasi saat ini menjadi lebih penting daripada yang sebelumnya [1]. Tidak dapat disangkal bahwa perkembangan pesat teknologi informasi adalah salah satu faktor utama yang menyebabkan era globalisasi berkembang lebih cepat. Internet, perdagangan elektronik, pertukaran data, kantor virtual, dan banyak lagi telah melampaui batas fisik antara negara. *Cryptocurrecy* sebagai pengganti mata uang konvensional adalah salah satu perkembangan ekonomi terbaru [2].

Mata uang kripto (*cryptocurrency*), juga dikenal sebagai mata uang virtual atau digital, adalah produk dari kemajuan teknologi keuangan. Mata uang digital semakin populer sebagai metode pembayaran online. Mata uang ini dibuat untuk menawarkan kemudahan dan keamanan pembayaran. Mata uang digital bernama *cryptocurrency* menggunakan sistem kriptografi untuk setiap transaksi data tanpa memerlukan pihak ketiga [3]. Meski banyak peminatnya, berinvestasi pada aset kripto juga memiliki resiko yang cukup besar, karena pergerakan harga yang tidak menentu dan sulitnya untuk memprediksi. Pada dasarnya investor atau trader hanya melakukan trasaksi secara intuitif, yang membuat banyak orang terjebak dan karena kesalahan dalam mengambil keputusan jual beli aset kripto [4].

Semakin banyak orang yang berinvestasi dalam *cryptocurrency* (aset kripto) saat ini membuat blockchain menjadi perhatian besar. Pada dasarnya, blockchain adalah buku besar digital yang datanya tidak dapat diganti atau diubah. Dengan demikian, blockchain memiliki kemampuan untuk memastikan dan menyimpan validasi transaksi yang transparan, cepat dan aman. Di masa mendatang, blockchain mungkin memiliki peran yang signifikan dalam pendidikan, pemerintahan, ekonomi dan kesehatan, dan akan menciptakan banyak potensi. Menurut riset PwC, teknologi blockchain memiliki potensi untuk meningkatkan ekonomi dunia sebesar US$ 1,76 triliun pada tahun 2030. Sektor pendidikan, kesehatan dan administrasi publik akan mendapatkan keuntungan yang paling besar [5].

Tidak seperti mata uang digital lainnya yang dapat diterbitkan secara terpusat, diedarkan dalam komunitas atau lokasi geografis, atau terkait dengan mata uang fiat atau organisasi yang menerbitkannya, *cryptocurrency* memiliki karakteristik yang sangat berbeda. Teknologi *blockchain* yang digunakan oleh *cryptocurrency*, seperti Bitcoin, adalah buku besar terdistribusi terbuka (*open distributed ledger*) yang mencatat tindakan transaksi. Ini memecahkan masalah

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

pengeluaran ganda dan tidak memerlukan pihak ketiga yang tepercaya. Desentralisasi memungkinkan teknologi *Blockchain* untuk meningkatkan kapasitas, keamanan yang lebih baik, dan pembayaran yang lebih cepat. Beberapa dari fitur ini berada di daftar teratas kekurangan sistem keuangan tradisional. Akibatnya, *blockchain* dan *cryptocurrency* telah menjadi dua topik yang paling mendesak [6].

Salah satu aset kripto yang memiliki banyak peminat adalah Ethereum (ETH), namun para investor kesulitan dalam menentukan waktu untuk membeli atau menjual ETH yang mereka ingin beli atau miliki. Karena harga yang begitu sangat fluktuatif dimana peminat dari ETH berasal dari berbagai negara sehingga tingkat resiko yang dihadapi sangat besar, karena hal tersebut diperlukan sebuah model yang dapat memprediksi nilai atau harga Ethereum berdasarkan data-data dan variable tertentu menggunakan Algoritma *Extreme Learning Machine* (ELM), sehingga para investor dapat menggunakan sebagai referensi sebelum membeli atau menjual ETH yang dimiliki. Dengan menggunakan hasil prediksi sebagai referensi, maka diharapkan dapat menguangi resiko kerugian yang mungkin dialami akibat membeli atau menjual aset kripto yang dimiliki.

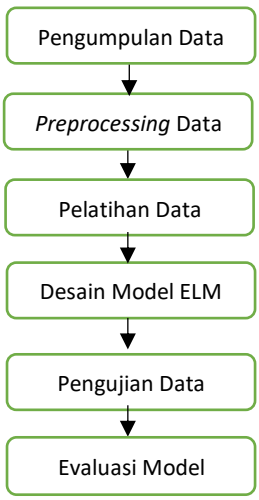
Jaringan syaraf tiruan (JST) adalah salah satu bagian dari cabang ilmu artificial intelligence (AI) yang meniru pola kerja jaringan syaraf (*Neural Network*) pada makhluk hidup dalam proses penyelesaian masalah [7],[8]. Beragam metode telah dikembangkan dalam JST dan salah satu metode yang dikembangkan tersebut adalah *Extreme Learning Machine* (ELM) [9]. ELM adalah metode pembelajaran maju (*Feedforward*) yang didasarkan pada teori matriks matematika. ELM menawarkan solusi pembelajaran dengan waktu pembelajaran relative cepat pada jaringan syaraf tiruan (JST). ELM memiliki algoritma yang sederhana yang memiliki kemampuan belajar yang sangat cepat dengan tingkat kesalahan (*error*) yang sangat kecil [10].

Model prediksi yang akan dibuat menggunakan metode *Extreme Learning Machine* (ELM). ELM menggunakan konsep single hidden layer feed forward networks (SLFNs). SLFNs dimaksudkan untuk mengatasi kelemahan metode jaringan syaraf tiruan feed forward, khususnya proses learning speed. ELM dapat digunakan untuk mengatasi masalah peramalan dalam waktu yang sangat cepat, juga mudah untuk diaplikasikan pada masalah-masalah kompleks dan diterapkan pada kehidupan nyata [11]. Karena hal tersebutlah maka dipilih metode ini sebagai metode dalam memprediksi harga ethereum.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Dalam Penelitian, terdapat beberapa tahapan-tahapan yang akan dilakukan seperti pengumpulan data, *pre-processing* data, alokasi data, desain model ELM, pelatihan data, pengujian data, dan evaluasi model. Berikut tahapan-tahapan dalam sistem prediksi harga Ethereum menggunakan penerapan metode algoritma Extreme Learning Machine (ELM) yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.2 Pengumpulan Data dan *Pre-processing* Data

Seluruh data yang akan digunakan berasal dari data historis harian Ethereum dari laman resmi coinmarketcap.com yang berjumlah 424 data dari tanggal 1 Januari 2022 sampai 29 Mei 2023, setiap data memiliki 3 atribut seperti *Open*, *High*, dan *Low*. Data harga Ethereum dikumpulkan dari CoinMarketCap dengan scraping data menggunakan CoinmarketCap Scraper dari library cryptocmd menggunakan python. Data yang akan menjadi Output dari rancangan sistem adalah nilai terakhir harian dari Ethereum yaitu harga Ethereum pada *Open*, *High*, dan *Low*.

Copyright © 2023 **Kasliono**, Page 96
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

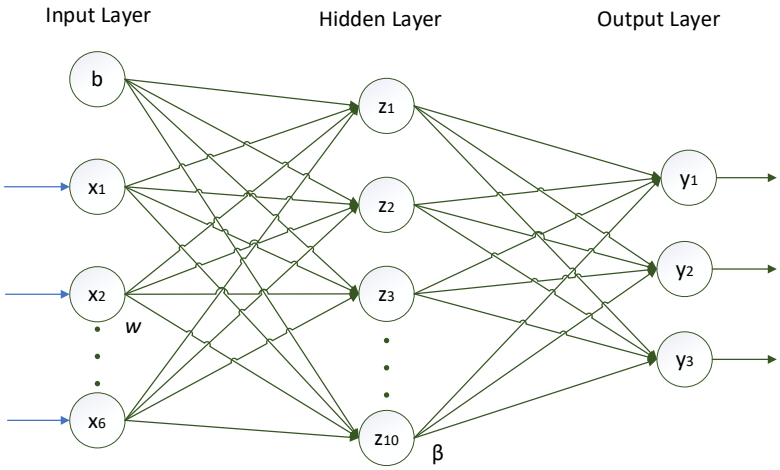
Kemudian, data yang diperoleh akan masuk ke tahap selanjutnya yaitu *pre-processing* data yaitu tahap pemrosesan pengubahan data mentah menjadi ke dalam bentuk yang dibutuhkan dan lebih mudah dipahami. Data yang diperoleh dibagi menjadi 2, yaitu *data training* sebanyak 318 data dan *data testing* sebanyak 106 data.

2.3 Extreme Learning Machine (ELM)

Algoritma Extreme Learning Machine (ELM) merupakan salah satu kategori algoritma jaringan syaraf tiruan feedforward network [9]. Metode ELM dibuat untuk mengatasi kelemahan-kelemahan dari jaringan syaraf tiruan feedforward terutama dalam learning speed karena menggunakan algoritma *gradient descent* untuk proses pembelajarannya. Algoritma pembelajaran *gradient descent* memerlukan iterasi untuk menemukan nilai-nilai bobot pada jaringan yang tepat sehingga membutuhkan waktu pelatihan yang lama. Beberapa parameter yang dibutuhkan untuk pelatihan, seperti laju pembelajaran dan kondisi penghentian, harus ditentukan dengan tepat melalui beberapa percobaan untuk mendapatkan hasil pelatihan yang baik. Pemilihan nilai laju pembelajaran yang terlalu kecil akan membutuhkan waktu yang lama untuk mencapai konvergen, namun jika nilainya terlalu besar menyebabkan algoritma menjadi tidak stabil, kondisi penghentian yang tidak tepat juga dapat menyebabkan kondisi *over fitting* yang menyebabkan kemampuan generalisasi *network* menurun [12].

ELM mampu melakukan pelatihan dalam waktu yang jauh lebih cepat (*extreme learning*) dibanding algoritma *gradient decent* karena tidak memerlukan iterasi. Parameter pada *hidden node* (bobot dan bias) dapat ditentukan di awal secara random dan nilainya tidak perlu diubah-ubah selama pelatihan melalui proses iterasi. Selanjutnya bobot dari *hidden layer* ke *output layer* ditentukan secara analitis menggunakan *invers matriks moore penrose*. ELM juga mampu menghasilkan kemampuan generalisasi yang baik karena ELM tidak hanya berusaha untuk meminimalkan *error*, namun juga *norm of weight* $\|\beta\|$. [9], [13].

Arsitektur yang digunakan merupakan jaringan syaraf *feedforward network* dengan *single hidden layer*, terdiri dari 3 *layer* yaitu lapisan *input*, lapisan tersembunyi (*hidden layer*) dan lapisan *output*. Berikut merupakan arsitektur ELM yang terlihat pada Gambar 2 dimana *b* adalah bias pada *hidden node*.



Gambar 2. Arsitektur Jaringan ELM

Pada penelitian ini, arsitektur yang digunakan terdiri dari jumlah *hidden node* antara 1 hingga 10 *hidden node* yang ditentukan setelah dilakukan percobaan dan perhitungan rata-rata MAPE, kemudian dipilih jumlah hidden node dengan nilai MAPE terkecil. Terdapat 3 *output node*, dimana y_1 adalah hasil prediksi harga *Open* Ethereun, y_2 adalah hasil prediksi harga *High* Ethereum dan y_3 adalah hasil prediksi harga *Low* Ethereum. Untuk input layer (x_i) adalah harga Ethereum pada *Open*, *High* dan *Low* yang jumlah input node-nya tergantung pada banyaknya data yang digunakan untuk melakukan prediksi, pada penelitian ini data yang digunakan saat training adalah 3 hari seperti yang terlihat pada (Gambar 2) yang merupakan data harga ETH pada 2 hari pertama sebagai *input*, kemudian data pada hari ke-3 digunakan sebagai target saat proses *training*, selanjutnya bobot dari hasil training tersebut akan digunakan untuk prediksi hari ke-4 dengan menggunakan data 3 hari sebelum yang akan diprediksi.

Data yang digunakan untuk pelatihan SLFN terdiri atas pasangan data input dan data output (x_i, y_i) untuk jumlah N sampel yang berbeda, dimana $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T \epsilon R^d$, dan $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \epsilon R^m$. Standar SLFN dengan L jumlah hidden node dan fungsi aktivasi $G(x)$ direpresentasikan secara matematis sebagai berikut.

$$f(x) = \sum_{i=1}^{L} \beta_i G(\boldsymbol{w}_i . \boldsymbol{x}_j + b_i) = \boldsymbol{y}_j, \qquad j = 1, \dots, N \qquad (1)$$

Dimana $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \dots, w_{iL}]^T$ merupakan vektor bobot yang menghubungkan hidden node ke-i dengan input node. $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ adalah vektor bobot yang menghubungkan hidden node ke-i dengan output node, dan b_i adalah bias dari hidden node ke-i. $\boldsymbol{w}_i . \boldsymbol{x}_j$ dinotasikan sebagai inner product dari $\boldsymbol{w}_i$ dan $\boldsymbol{x}_j$.

Copyright © 2023 **Kasliono**, Page 97
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

Pada *input layer*, *hidden node*, dan *output layer* pada ELM disarankan dalam bentuk "generalized" *single-hidden layer feedfordward networks* (SLFN). *Hidden layer* tidak menjadi neuron yang sama pada jaringan ini [14],[15],[16]. Output dari fungsi ELM untuk *generalized* SLFN direpresentasikan oleh persamaan (2). [17],[18],[19].

$$f(x) = \sum_{i=1}^{L} \beta_i g(\boldsymbol{w_i}, \boldsymbol{x_j}, b_i) = \boldsymbol{h(x)} \cdot \beta_i, \quad (2)$$

Sehingga persamaan (2) dapat ditulis secara lengkap dalam bentuk perkalian matriks pada persamaan (3).

$$H\beta = Y \quad (3)$$

Dimana nilai vektor $h(x)$ dihitung untuk semua data yang ada sehingga akan didapatkan matriks H yang merupakan matriks keluaran dari hidden layer, seperti pada persamaan (4). Untuk matriks bobot β yang menghubungkan hidden layer dengan output layer seperti pada persamaan (5). Sedangkan Matriks Y merupakan matriks target dari data latih seperti pada persamaan (6).

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(w_1.x_1 + b_1) & \cdots & G(w_L.x_1 + b_L) \\ \vdots & \vdots & \vdots \\ G(w_1.x_N + b_1) & \cdots & G(w_L.x_N + b_L) \end{bmatrix}_{N \times L} \quad (4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad (5)$$

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m} \quad (6)$$

Pada ELM bobot input dan bias ditentukan secara random, maka untuk mencari nilai bobot output yang menghubungkan hidden layer dengan output layer ditentukan dengan persamaan (7).

$$\beta = H^{\dagger}Y \quad (7)$$

Dimana matriks invers $H^{\dagger}$ merupakan matriks output H yang nilai inversnya diperoleh dengan persamaan invers matriks moore penrose generalized inverse.

$$H^{\dagger} = (H^T H)^{-1} H^T \quad (8)$$

Secara lebih ringkas, algoritma ELM dengan SLFN dapat dimodelkan dengan langkah sebagai berikut [20],[12],[21],[22].

Input : Data *training* (x_i, y_i) dengan $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T \in R^d$ dan $y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$. L adalah banyak *hidden node/neuron* dan fungsi aktivasi $G(x)$.

a. Tentutkan vektor bobot w_i dan bias b_i secara random, untuk $i = 1,2, \dots, L$.
b. Hitung matriks H yang merupakan nilai pada *node* di *hidden layer*, dengan menggunakan persamaan (4).
c. Hitung matriks Invers *Moore Penrose Generalized* $H^{\dagger}$ menggunakan persamaan (8).
d. Hitung matriks bobot *output* β yang merupakan bobot menuju *output node* menggunakan persamaan (7), dimana Y diperoleh dengan persamaan (6).

Output : Sebuah *Single hidden layer Feedforward neural Network* (SLFN) dengan vektor bobot *output* β dan vektor bobot w_i dan bias b_i yang ditentukan secara *random*.

Untuk setiap *input* $x_i \in R^d$ nilai output dari y dapat hitung menggunakan persamaan (1). Dimana w_i, b_i dan fungsi aktivasi $G(x)$ adalah *input*, serta vektor bobot β adalah *output* dari algoritma ELM.

2.4 Pelatihan Data (*Data Training*)

Pelatihan data pada metode *Extreme Learning Machine* (ELM) berguna untuk membentuk model dan pelatihan yang akurat dalam melakukan prediksi. Berikut adalah beberapa tahapan pelatihan data. Pada penelitian ini untuk data *trainig* berjumlah 318 data.

a. Menghasilkan atau menentukan nilai bobot dan bias pada hidden node $(w_i.b_i)_{i=1}^{L}$. Nilai-nilai ini diperoleh secara acak (*random*) dalam rentang 0 hingga 1. Nilai bobot mewakili kekuatan koneksi antar neuron, sedangkan nilai bias mewakili ambang di mana neuron diaktifkan.
b. Memperoleh nilai output dari lapisan tersembunyi (H) menggunakan fungsi aktivasi hidden node $G(w_i.x_j + b_i)$ dalam bentuk matriks.
c. Memperoleh nilai output dari lapisan tersembunyi (H) menggunakan fungsi aktivasi. Fungsi aktivasi yang digunakan dalam hal ini adalah fungsi aktivasi linier ($y = x$). Nilai keluaran dari lapisan tersembunyi kemudian diteruskan sebagai masukan ke lapisan (*layer*) berikutnya.

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

d. Tentukan matriks Moore Penrose PseudoInverse menggunakan persamaan (8).
e. Hitung matriks output weight menggunakan persamaan (7).

2.5 Pengujian Data (*Data Testing*)

Testing Data adalah proses dimana hasil prediksi divalidasi dan algoritma ELM dievaluasi berdasarkan pelatihan data yang dilakukan sebelumnya. Pada penelitian ini digunakan 106 data sebagai data *testing*. Langkah-langkah dalam proses ini mirip dengan proses pelatihan, tetapi tidak ada perhitungan *pseudo-invers Moore-Penrose* dan bobot keluaran. Berikut langkah-langkah dalam proses pengujian:

1. Menghitung *output hidden layer* vektor $h(x)$ menggunakan fungsi aktivasi hidden node $G(w_i.x_j + b_i)$ yang telah ditentukan dan nilai bobot $(w_i.b_i)_{i=1}^{L}$ yang diperoleh dari proses pelatihan.
2. Menghitung nilai vektor output menggunakan $f(x) = h(x)\beta$ dengan β adalah matriks bobot dari hidden node ke output node yang diperoleh dari proses pelatihan.
3. Hitung selisih nilai hasil peramalan yang diperoleh dengan persamaan (2) dan data aktual menggunakan Mean Absolute Percentage Error (MAPE) dengan persamaan (9).

2.6 Evaluasi Model

Salah satu metode untuk menemukan nilai absolut dari selisih antara nilai prediksi dan nilai observasi dibagi dengan data observasi, atau data aktual, adalah *Mean Absolute Percentage Error* (MAPE). Kemudian, nilai ini ditampilkan dengan persentase. MAPE menunjukkan seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya (data aktual) dari data runtun waktu. Semakin kecil nilai presentasi kesalahan (*percentage error*) pada MAPE, maka semakin akurat hasil peramalan tersebut. Rumus MAPE dilihat pada persamaan (9). [23],[24],[25],[26],[27],[22].

$$MAPE = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{p_j - y_j}{y_j}\right| \times 100\% \qquad (9)$$

dimana
N: banyak periode ramalan (1 periode),
p_j: data *output* ke-j (prediksi harga ethereum),
y_j: data harga Ethereum aktual ke-j (data *testing*).

Besar kecilnya kemampuan peramalan dalam penelitian ini ditentukan oleh kecilnya persentase kesalahan yang diperoleh dari proses pengujian, semakin rendah kesalahan yang dicapai maka semakin tinggi nilai akurasinya dan semakin baik kemampuan peramalannya.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini akan dilakukan eksperimen untuk menguji dengan mengubah ukuran *hidden node* untuk memprediksi setiap hari pada data pengujian. Kemudian akan terlihat eksperimen mana yang memiliki performa baik berdasarkan MAPE yang diperoleh dari masing-masing eksperimen tersebut.

Hasil eksperimen diperoleh dengan mengubah ukuran *hidden node* ke jaringan. Selanjutnya mengukur keakuratan hasil prediksi terhadap data aktual untuk semua data pengujian dengan menghitung MAPE semua data pada hari yang diprediksi. Terdapat 10 eksperimen dimana satu eksperimen memiliki ukuran *hidden node* yang berbeda dengan 1000 kali pengulangan untuk setiap percobaan kemudian didapatkan rata-rata MAPE. Pengulangan diperlukan karena parameter pada *hidden node* (bobot dan bias) ditentukan di awal secara acak. Oleh karena itu, dengan melakukan pengulangan akan diperoleh hasil pengukuran *error* yang optimal (*Average* MAPE).

3.1 Prediksi Harga Ethereum dengan ELM

Langkah awal dalam prediksi menggunakan ELM adalah menentukan *input* pada *input layer*, dalam skenario ini digunakan data pada Tabel 1, dimana yang menjadi *input* adalah harga ETH pada tanggal 1 april 2022 dengan atribut *Open, High* dan *Low* secara berurutan adalah x_1, x_2 dan x_3. Kemudian disusul dengan data pada tanggal 2 april 2022 sebagai x_4, x_5 dan x_6. Data pada tanggal 3 april 2022 adalah Target (y), yaitu y_1, y_2 dan y_3, dimana y_1 untuk atribut harga *Open*, y_2 untuk atribut harga *High* dan y_3 untuk atribut harga *Low*.

Tabel 1. Nilai *Input* (x) dan Target (y) saat *Training*

Date	*Open*	*High*	*Low*
01/04/2022	3282.576	3467.556	3223.891304
02/04/2022	3449.789	3521.285	3442.00016
03/04/2022	3444.811	3573.96	3421.259779

Selanjutnya, menentukan matriks nilai bobot input dan bias secara random dengan *range* nilai random [0, 1]. Pada skenario ini jumlah *hidden node*.adalah 3 *node*. Sehingga diperoleh nilai bobot dan bias seperti yang terlihat pada Tabel 2

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

Tabel 2. Nilai bobot dan bias antara *input layer* dan *hidden layer*

w_{ij}	w_{i1}	w_{i2}	w_{i3}
w_{1j}	0.313383	0.733131	0.5463
w_{2j}	0.109798	0.816029	0.226013
w_{3j}	0.534772	0.302178	0.112081
w_{4j}	0.910303	0.961085	0.457725
w_{5j}	0.881861	0.89695	0.478305
w_{6j}	0.398973	0.534844	0.130365
bj	0.986665	0.695955	0.554647

Selanjutnya dihitung matriks H dengan menggunakan persamaan (4). Sehingga diperoleh matriks H yang merupakan matriks keluaran dari *hidden layer*.

$$H = [10753.3694 \quad 14525.9547 \quad 6650.8919]$$

Langkah ketiga adalah mencari nilai bobot *output* antara *hidden layer* dan *output layer*. Persamaan invers matriks yang digunakan adalah invers matriks *moore penrose* dari matriks H yang merupakan matriks output dari *hidden layer* yang dinotasikan dengan $H^{\dagger}$. Dengan mengunakan persamaan (8), maka diperoleh invers dari matriks H, selanjutnya mengalikan matriks invers $H^{\dagger}$ dengan nilai target untuk memperoleh nilai bobot β dengan menggunakan persamaan (7). Diperoleh Nilai bobot *output* β seperti yang terlihat pada Tabel 3.

Tabel 3. Nilai bobot *output* β

y_1	y_2	y_3
$\beta_{11} = 0.0998$	$\beta_{12} = 0.1036$	$\beta_{13} = 0.0991$
$\beta_{21} = 0.1349$	$\beta_{22} = 0.1399$	$\beta_{23} = 0.1340$
$\beta_{31} = 0.0617$	$\beta_{32} = 0.0640$	$\beta_{33} = 0.0613$

Kemudian, untuk prediksi bobot output β pada Tabel 3 akan digunakan dalam proses *testing*. Langkahnya hampir sama dengan proses pelatihan, perbedaannya adalah setelah menentukan matriks H, maka dapat ditentukan hasil prediksi dengan persamaan (3) menggunakan nilai bobot output β yang diperoleh saat pelatihan. Sehingga dapat diprediksi harga Ethereum pada tanggal 4 April 2022, dimana data hasil prediksi dan data aktual dapat dilihat pada Tabel 4.

Tabel 4. Hasil Prediksi dan Aktual Harga ($) Ethereum

Date	*Open*		*High*		*Low*	
	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual
04/04/2022	3510.377	3522.365	3641.985	3535.148	3486.378	3422.001

Cara yang sama dilakukan untuk seluruh proses eksperimen dalam penentuan jumlah *hidden node* yang optimal dalam prediksi harga Ethereum. Kemudian data hasil prediksi dibandingkan dengan data aktual dengan menghitung MAPE menggunakan persamaan (9). Selanjutnya, dihitung nilai MAPE untuk setiap atribut harga yang diprediksi (*Open*, *High*, dan *Low*) kemudian dihitung rata-rata MAPE untuk setiap atribut pada masing-masing eksperimen.

3.2 Rata-Rata MAPE pada Semua Eksperimen

Pada tahap ini, dilakukan pengujian banyaknya *hidden node* pada *hidden layer* untuk mengetahui pengaruh banyaknya *node* terhadap hasil prediksi. Dilakukan pengujian dengan mengganti banyak *hidden node* antara 1 *hidden node* hingga 10 *hidden node*, selanjutnya menghitung rata-rata dari MAPE pada tiap eksperimen, dimana untuk tiap eksperimen dilakukan 1000 kali perulangan untuk mendapatkan rata-rata MAPE yang lebih *smooth*. Hasilnya dapat terlihat pada Tabel 5.

Tabel 5. Rata-rata MAPE pada setiap eksperimen

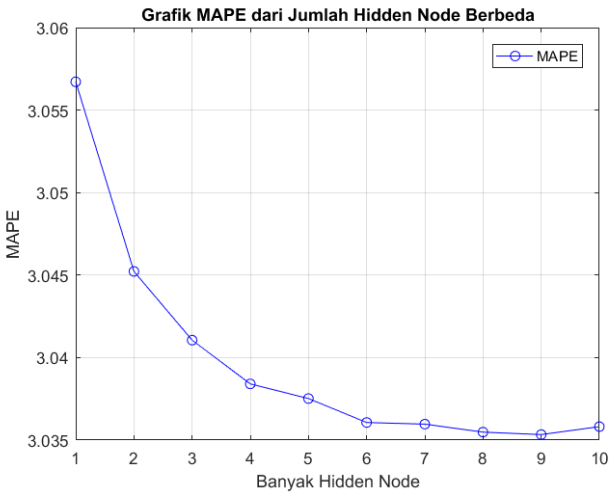
Eksperimen ke-	Ukuran *Hidden Node*	Rata-Rata MAPE (%)
1	1	3.056715
2	2	3.045222
3	3	3.041058
4	4	3.038406
5	5	3.037519
6	6	3.036067
7	7	3.035968
8	8	3.035482
9	9	3.035334

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

10	10	3.035816

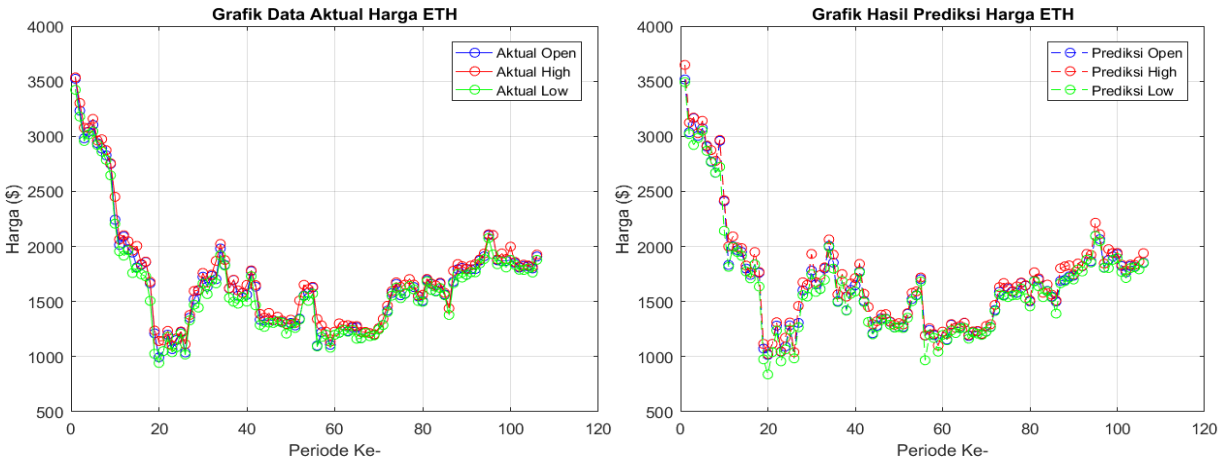
Kemudian plot grafik dari rata-rata MAPE pada seluruh eksperimen setelah dilakukan perulangan sebanyak 1000 kali yang dapat terlihat pada Gambar 3. Pada gambar tersebut terlihat bahwa nilai rata-rata MAPE mengalami penurunan seiring dengan penambahan jumlah *hidden node*. Namun, ada kemungkinan bahwa jumlah *hidden node* terbesar tidak selalu memiliki rata-rata MAPE terkecil, dari hasil eksperimen pada penelitian ini terlihat bahwa ukuran *hidden node* yang memiliki rata-rata MAPE terkecil adalah 9 *hidden node* yaitu sebesar 3.035334%. Dari eksperimen yang dilakukan pada penelitian ini juga terlihat meski rata-rata MAPE terkecil adalah 9 *hidden node*, tetapi tidak terjadi perbedaan nilai rata-rata MAPE yang signifikan untuk seluruh ukuran *hidden node*.



Gambar 3. Grafik Rata-Rata MAPE untuk setiap ukuran *Hidden Node*

3.3 Model Prediksi

Dari hasil pengujian yang telah dilakukan pada tiap banyaknya hidden node yang berbeda, model yang dihasilkan oleh ELM dengan 9 *hidden node* memiliki nilai *error* (MAPE) yang kecil dalam melakukan prediksi. Model jaringan syaraf tiruan ELM dengan 9 *hidden node* mampu menghasilkan nilai prediksi dengan nilai rata-rata MAPE sebesar 3.035334%. Hasil terbaik dari ELM didapatkan dengan 9 *hidden node* pada arsitektur jaringan ELM yang digunakan untuk prediksi, artinya untuk memprediksi harga Ethereum pada *Open*, *High* dan Low hari berikutnya menggunakan 9 *hidden node* pada arsitektur jaringan ELM yang digunakan untuk prediksi.



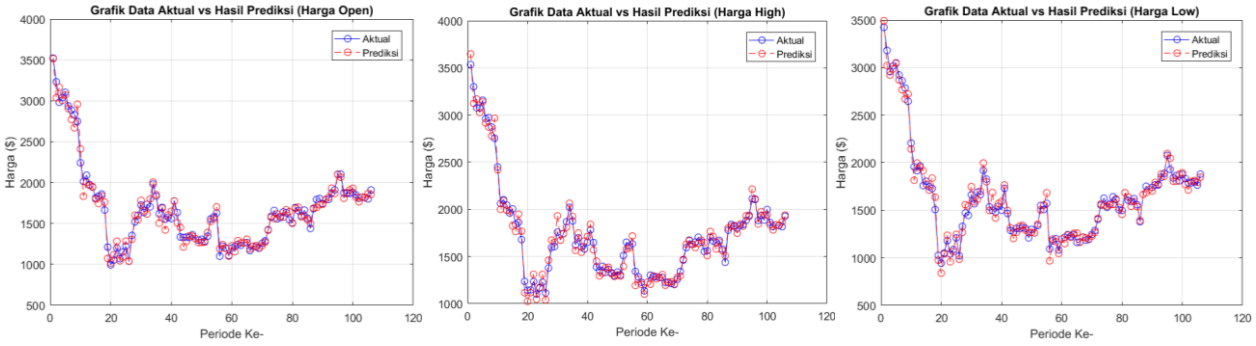
(a) (b)

Gambar 4. Grafik Data Aktual dan Grafik Hasil Prediksi Harga ETH

Dengan menggunakan model yang telah dihasilkan, prediksi dapat dilakukan untuk semua data *testing*. Sehingga diperoleh hasil prediksi seperti pada Gambar 4. Berdasarkan *hidden node* optimum yang telah digunakan untuk memprediksi, dapat dikatakan bahwa untuk memprediksi harga Ethereum ukuran *hidden node* akan memberikan pengaruh terhadap hasil prediksi, sehingga diperlukan pemodelan arsitektur ELM yang akan digunakan untuk melakukan prediksi.

Copyright © 2023 **Kasliono**, Page 101
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567



(a) (b) (c)

Gambar 5. Grafik Harga *Open*, *High* dan *Low* ETH

Pada Gambar 5 untuk Harga *Open*, *High* dan *Low* terlihat memiliki pola yang serupa. Sehingga hasil prediksi untuk tiap atribut tersebut juga memiliki pola yang serupa, jika disandingkan antara hasil prediksi dan data aktual dari ketiga atribut tersebut, terlihat pola yang serupa juga. Hal ini dipengaruhi oleh ukuran data *testing* atau penentuan nilai *input* pada proses pelatihan (*window size*). Sehingga dapat dikatakan bahwa model ELM yang dibuat bisa digunakan untuk memprediksi harga Ethereum.

4. KESIMPULAN

Dari penilitian yang dilakukan dapat disimpulkan bahwa, banyak *hidden node* yang memiliki *error* (MAPE) terkecil adalah arsitektur jaringan ELM dengan 9 *hidden node* yang memiliki nilai rata-rata MAPE sebesar 3.035334%, sehingga metode *Extreme Learning Machine* dapat digunakan untuk memprediksi harga Ethereum dengan menggunakan arsitektur SLFN dimana memiliki 9 *hidden node*. Terlihat pola data yang yang mirip antara data aktual dan hasil prediksi dimana hal tersebut disebabkan oleh ukuran data uji atau penentuan nilai *input* (*window size*) pada proses *training*, sehingga nilai akurasi pada penelitian ini bisa dioptimumkan dengan melakukan perubahan pada ukuran data uji dan penentuan nilai *input* saat proses *training*. Kemudian, penelitian ini tidak melakukan eksperimen terhadap perbedaan *window size* yaitu sebagai jumlah dari *input node* serta pengukuran *running time* untuk melihat ukuran *hidden node* yang memiliki *running time* optimum, sehingga dapat dilakukan penelitian lebih lanjut, dengan mengganti ukuran *input* pada *input layer*, kemudian melihat apakah ada perbedaan pada nilai akurasi prediksi serta mengukur *running time* pada masing-masing eksperimen tersebut.

REFERENCES

[1] R. Muntaqo, “Teknologi Informasi Dan Komunikasi Dalam Perkembangan Budaya Masyarakat,” *Pemberlakuan Pembatasan Kegiat. Masy.*, pp. 12–20, 2017.
[2] N. Huda and R. Hambali, “Risiko dan Tingkat Keuntungan Investasi Cryptocurrency,” *J. Manaj. dan Bisnis Performa*, vol. 17, no. 1, pp. 72–84, 2020.
[3] A. R. Ashariansyah, N. Iriawan, and A. Mukarromah, “Pemodelan Harga Cryptocurrency Menggunakan Markov Switching Autoregressive,” *Inferensi*, vol. 3, no. 2, p. 81, 2020, doi: 10.12962/j27213862.v3i2.7726.
[4] I. G. M. H. PRATAMA, I. W. SUMARJAYA, and N. L. P. SUCIPTAWATI, “Peramalan Harga Bitcoin Dengan Metode Smooth Transition Autoregressive (Star),” *E-Jurnal Mat.*, vol. 11, no. 2, p. 100, 2022, doi: 10.24843/mtk.2022.v11.i02.p367.
[5] A. Rahma, “Daftar Kelebihan dan Kekurangan Blockchain yang Wajib Kamu Tahu!,” *TokoNews*, 2022. https://news.tokocrypto.com/2022/02/17/daftar-kelebihan-dan-kekurangan-blockchain-yang-wajib-kamu-tahu/ (accessed Jun. 06, 2023).
[6] D. K. C. Lee, L. Guo, and Y. Wang, “Cryptocurrency: A new investment opportunity?,” *J. Altern. Investments*, vol. 20, no. 3, pp. 16–40, 2018, doi: 10.3905/jai.2018.20.3.016.
[7] Suyanto, *Artificial Intelligence : Searching, Reasoning, Planning dan Learning Revisi Kedua*, Revisi Ked. Bandung: Informatika Bandung, 2014.
[8] W. Budiharto, *Machine Learning dan Computational Intelligence*. Yogyakarta: CV. Andi Offset (Penerbit ANDI), 2016.
[9] G. Huang, Q. Zhu, and C. Siew, “Extreme Learning Machine : A New Learning Scheme of Feedforward Neural Networks,” *IEEE Int. Jt. Conf. Neural Networks*, vol. 2, pp. 985–990, 2004, doi: 10.1109/IJCNN.2004.1380068.
[10] U. Mahdiyah, M. I. Irawan, and E. M. Imah, “Study Comparison Backpropogation, Support Vector Machine, and Extreme Learning Machine for Bioinformatics Data,” *J. Ilmu Komput. dan Inf. (Journal Comput. Sci. Information)*, vol. 1, pp. 53–59, 2015, [Online]. Available: http://dx.doi.org/10.21609/jiki.v8i1.284.
[11] A. N. Alfiyatin, W. F. Mahmudy, C. F. Ananda, and Y. P. Anggodo, “Penerapan Extreme Learning Machine (Elm) Untuk Peramalan Laju Inflasi Di Indonesia Implementation Extreme Learning Machine for Inflation Forecasting in Indonesia,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 2, pp. 179–186, 2018, doi: 10.25126/jtiik.20186900.
[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 17, no. 1, pp. 489–501, 2006, doi: 10.1007/s10462-013-9405-z.
[13] G. Bin Huang, D. H. Wang, and Y. Lan, “Extreme learning machines: A survey,” *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, 2011, doi: 10.1007/s13042-011-0019-y.

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 5, No 1, Juni 2023 Page: 95−103
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v5i1.3567

[14] G. Bin Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007, doi: 10.1016/j.neucom.2007.02.009.
[15] G. Bin Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008, doi: 10.1016/j.neucom.2007.10.008.
[16] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 42, no. 2, pp. 513–529, 2012, doi: 10.1109/TSMCB.2011.2168604.
[17] G. Bin Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010, doi: 10.1016/j.neucom.2010.02.019.
[18] M. A. A. Albadr and S. Tiun, "Extreme learning machine: A review," *Int. J. Appl. Eng. Res.*, vol. 12, no. 14, pp. 4610–4623, 2017.
[19] A. Kumar Patel and S. Kumar Jain, "Arterial Parameters and Elasticity Estimation in Common Carotid Artery Using Deep Learning Approach," *Int. J. Image, Graph. Signal Process.*, vol. 11, no. 11, pp. 18–28, 2019, doi: 10.5815/ijigsp.2019.11.03.
[20] R. Singh and S. Balasundaram, "Application of Extreme Learning Machine Method for Time Series Analysis," *Proc. World Acad. Sci.*, vol. 26, no. Part 1, pp. 361–367, 2007, doi: 10.1148/radiology.138.2.7455105.
[21] A. A. Anbarasa Pandian and R. Balasubramanian, "Analysis on Shape Image Retrieval Using DNN and ELM Classifiers for MRI Brain Tumor Images," *Int. J. Inf. Eng. Electron. Bus.*, vol. 8, no. 4, pp. 63–72, 2016, doi: 10.5815/ijieeb.2016.04.08.
[22] Kasliono, Suprapto, and F. Makhrus, "Point Based Forecasting Model of Vehicle Queue with Extreme Learning Machine Method and Correlation Analysis," *Int. J. Intell. Syst. Appl.*, vol. 13, no. 3, pp. 11–22, 2021, doi: 10.5815/ijisa.2021.03.02.
[23] I. Nabillah and I. Ranggadara, "Mean Absolute Percentage Error untuk Evaluasi Hasil Prediksi Komoditas Laut," *JOINS (Journal Inf. Syst.*, vol. 5, no. 2, pp. 250–255, 2020, doi: 10.33633/joins.v5i2.3900.
[24] H. Sarjono and B. S. Abbas, *Forecasting: Aplikasi Penelitian Bisnis QM for Windows vs MINITAB vs MANUAL*. Bogor: Mitra Wacana Media, 2017.
[25] P. M. Swamidass, Ed., "MAPE (mean absolute percentage error)MEAN ABSOLUTE PERCENTAGE ERROR (MAPE)," in *Encyclopedia of Production and Manufacturing Management*, Boston, MA: Springer US, 2000, p. 462.
[26] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, "Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error," *J. Phys. Conf. Ser.*, vol. 930, no. 1, 2017, doi: 10.1088/1742-6596/930/1/012002.
[27] Fahirah and L. Wulandari, "The Implementation of Least Square Method on the Palm Shells Sales Forecasting Application," *Int. J. Inf. Eng. Electron. Bus.*, vol. 12, no. 5, pp. 1–13, 2020, doi: 10.5815/ijieeb.2020.05.01.

This Journal is licensed under a Creative Commons Attribution 4.0 International License