

## Prediksi Nilai *Cryptocurrency Bitcoin* menggunakan Algoritme *Extreme Learning Machine (ELM)*

Rahmat Faizal<sup>1</sup>, Budi Darma Setiawan<sup>2</sup>, Imam Cholissodin<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>rhmtfaizal@student.ub.ac.id, <sup>2</sup>s.budidarma@ub.ac.id, <sup>3</sup>imamcs@ub.ac.id

### Abstrak

Bitcoin merupakan salah satu bentuk *cryptocurrency* yang dilirik masyarakat karena pengelolaannya yang terdesentralisasi, kerahasiaan yang terjaga, serta prosesnya yang mudah. Namun, mata uang digital ini mengalami fluktuasi yang ekstrim yang membuat beberapa pemilik aset merasa dirugikan. Banyak cara dilakukan untuk mencegahnya seperti melihat pergerakan nilai secara terus menerus yang, melakukan aksi tanpa mempertimbangkan prospek kedepannya, atau bahkan membiarkannya sampai waktu yang ditentukan oleh pemilik aset. Tentu saja hal itu tidak efisien mengingat tujuan utama penyimpanan aset adalah mendapatkan keuntungan. Maka dari itu, diperlukan sistem yang dapat memprediksi nilai Bitcoin secara tepat dan akurat sehingga dapat membantu mengurangi kerugian serta menjadi bahan pertimbangan dalam proses jual beli *cryptocurrency* Bitcoin. Penelitian ini memiliki tujuan untuk mendapatkan nilai prediksi *cryptocurrency* Bitcoin menggunakan algoritme *Extreme Learning Machine (ELM)*. Berdasarkan hasil implementasi serta analisis yang telah dilakukan menggunakan data Bitcoin dari tanggal 1 Mei 2018 sampai dengan 1 Agustus 2018 diperoleh nilai kesalahan terkecil menggunakan *Mean Absolute Percentage Error (MAPE)* sebesar 2,657% dengan jumlah fitur sebanyak 2, jumlah *hidden neuron* sebanyak 4, persentase jumlah data latih sebesar 80%, serta rentang nilai bobot dengan rentang [-1.8, 1.8].

**Kata kunci:** prediksi, *cryptocurrency bitcoin*, *extreme learning machine*, regresi, *mean absolute percentage error*

### Abstract

*Bitcoin is one of cryptocurrency which is popular among people due to decentralized management, well-maintained confidentiality, and easy process. But, this type of cryptocurrency is extremely volatile which makes the owner feel aggrieved. Lots of actions have been taken to overcome this by seeing the statistic movement over and over, Taking actions without considering the future prospect, or make the the asset being untouched until the considered time. Those are inefficient regarding the goal is to get the profit. Therefore, the need of system which can predict the value of Bitcoin accurately and efficiently so it can help decreasing the risk of losing and could give another consideration on trading cryptocurrency Bitcoin. This research has a purpose to obtain the value of cryptocurrency Bitcoin using Extreme Learning Machine (ELM) algorithm. Based on the implementation and analysis conducted using Bitcoin Data from May 1<sup>th</sup>, 2018 until August 1<sup>th</sup>, 2018, it can be obtained that the smallest error value using Mean Average Percentage Error (MAPE) is 2,657% with the number of features is 2, the number of hudden neuron is 4, and the percentage of training data is 90%, also the range of with range [-1.8, 1.8].*

**Keywords:** prediction, *cryptocurrency bitcoin*, *extreme learning machine*, regression, mean, absolute percentage error

## 1. PENDAHULUAN

Bitcoin merupakan salah satu bentuk *cryptocurrency* yang dibuat oleh Satoshi Nakamoto yang pengelolaannya tidak dipegang oleh satu bank pusat, melainkan dikelola oleh pengguna bitcoin yang identitasnya tidak

dimunculkan. Dengan keadaan seperti itu membuat Bitcoin dilirik masyarakat untuk menyimpan asetnya karena pengelolaannya yang terdesentralisasi, kerahasiaan yang terjaga, serta prosesnya yang mudah (Pratama, 2017). *Cryptocurrency* Bitcoin mengalami fluktuasi yang terbilang sangat cepat dikarenakan tidak adanya nilai dasar seperti halnya pada mata uang

konvensional. Sehingga, pemilik aset tidak ingin dirugikan dengan melakukan berbagai cara yang kurang efisien seperti melihat pergerakan nilai Bitcoin secara terus menerus, menjual tanpa mempertimbangkan prospek kedepan, atau bahkan membiarkannya tertahan sampai waktu yang ditentukan oleh pemilik. Maka dari itu, diperlukan sebuah sistem yang dapat digunakan untuk memprediksi nilai Bitcoin kedepan serta menjadi bahan pertimbangan dalam proses jual beli mata uang digital Bitcoin.

Untuk melakukan prediksi nilai *cryptocurrency* Bitcoin, banyak metode yang bisa digunakan untuk membangun sistem prediksi ini. Salah satunya adalah yang dilakukan oleh Radityo dkk (2017) yang mana melakukan prediksi menggunakan algoritme *Backpropagation*. Namun, kelemahan dari algoritme ini adalah kecepatan dalam mempelajari pola data yang lama serta terkadang tidak bisa mengatasi *local minima*. Oleh karena itu, dalam penelitian ini penulis menggunakan algoritme *Extreme Learning Machine* (ELM) yang dapat mengatasi permasalahan pada algoritme *Backpropagation*. Algoritme ini merupakan salah satu algoritme jaringan syaraf tiruan yang mana memiliki keunggulan dari segi kecepatan serta tingkat akurasi. Algoritme ini juga banyak digunakan oleh peneliti untuk melakukan klasifikasi maupun prediksi. Penelitian yang dilakukan oleh Mosabeth dkk (2018) yang melakukan prediksi harga pasar daging sapi menggunakan algoritme *Extreme Learning Machine* (ELM) menggunakan data *time series* didapatkan hasil terbaik yaitu nilai MAPE sebesar 0,344% dengan perbandingan data latih dengan data uji yaitu 90%:10%, rentang bobot antara -1 dan 1, jumlah *hidden neuron* 7, serta menggunakan fungsi aktivasi sigmoid biner dan jumlah fitur 3.

Penelitian selanjutnya yang dilakukan oleh Permatasi dkk (2018) yang melakukan estimasi hasil produksi benih kenaf yang mana mendapatkan nilai MAPE terbaik sebesar 0,160% dengan jumlah *neuron* pada *hidden layer* sejumlah 8 serta menggunakan fungsi aktivasi *sigmoid biner* dan persentase jumlah data latih dan data uji sebesar 90%:10% (Permatasari dan Rahayudi, 2018).

Selanjutnya, penelitian yang dilakukan oleh Jauhari dkk (2018) yang melakukan prediksi nilai tukar rupiah Indonesia terhadap dollar amerika serikat menggunakan *Recurrent Extreme Learning Machine*. Dari penelitian tersebut didapatkan hasil MAPE sebesar

0.069502%. Berdasarkan uraian di atas, penulis memutuskan untuk melakukan penelitian dengan judul “Prediksi Nilai *Cryptocurrency* Bitcoin menggunakan Algoritme *Extreme Learning Machine* (ELM)”.

## 2. DASAR TEORI

### 2.1. Prediksi

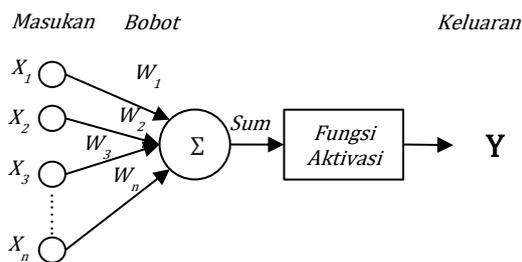
Prediksi menghasilkan suatu nilai yang menggambarkan kondisi suatu entitas berdasarkan data aktual. Prediksi diperlukan untuk memperkirakan bagaimana keadaan yang akan datang sehingga dapat dijadikan suatu acuan dalam pengambilan keputusan serta perencanaan kedepannya. Prediksi sendiri di dalamnya terdapat klasifikasi dan regresi. Klasifikasi yang dimaksud adalah penggolongan suatu entitas ke dalam kelompok-kelompok tertentu sesuai dengan standar tertentu. Selain klasifikasi, terdapat juga regresi yang dapat digunakan untuk melakukan prediksi berdasarkan hubungan antara 2 parameter atau lebih. Regresi dapat melakukan prediksi untuk mendapatkan suatu nilai yang menggambarkan kondisi yang akan datang berdasarkan parameter-parameter yang mempengaruhi. Pada penelitian ini, penulis menggunakan regresi untuk melakukan prediksi. Hal ini dikarenakan penelitian ini bertujuan untuk mendapatkan nilai secara numerik pada masa yang akan datang.

### 2.2. *Cryptocurrency* Bitcoin

Bitcoin merupakan suatu mata uang digital (*cryptocurrency*) yang diciptakan oleh seseorang yang tidak diketahui identitasnya namun mengatasnamakan dirinya Satoshi Nakamoto. Bitcoin mengalami fluktuasi yang cukup ekstrim karena beberapa faktor yaitu volume dan persebaran bitcoin, adanya penipuan yang mengatasnamakan bitcoin, serta spekulasi pemilik aset.

### 2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan model komputasi yang diadaptasi cara kerja otak manusia seperti kemampuannya untuk belajar. JST ini memiliki kesamaan dengan manusia yang memiliki beberapa *neuron* yang berkaitan satu sama lain.. Untuk lebih lanjut mengenai struktur *node* atau *neuron* di jaringan syaraf tiruan ini ditunjukkan pada Gambar 1 dibawah ini.



Gambar 1. Struktur Jaringan Syaraf Tiruan

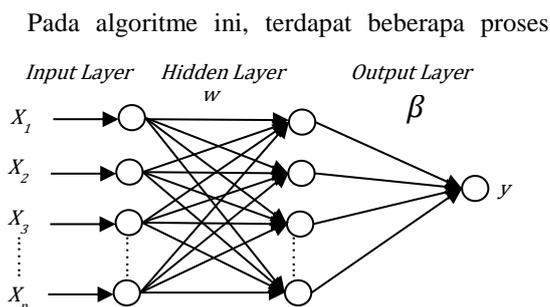
Dalam struktur jaringan syaraf tiruan ini, terdapat lapisan masukan (*input layer*) sejumlah  $n$  yang terdiri dari beberapa *neuron* yaitu  $X_1, X_2, X_3, \dots, X_n$ . Lalu, masing-masing masukan yang memiliki nilai akan dikalikan dengan bobot (*weight*)  $W_1, W_2, W_3, \dots, W_n$  yang sudah diinisialisasi sebelumnya dengan rentang nilai yang sudah ditentukan. Setelah penjumlahan masing-masing masukan dan bobot telah dilakukan, maka proses selanjutnya adalah menjumlahkan seluruh *neuron* dari hasil penjumlahan tersebut dan dimasukkan kedalam sebuah fungsi aktivasi untuk menentukan nilai ambang (*threshold*) *neuron* tersebut. Untuk meneruskan informasi yang didapatkan, nilai masukan hasil aktivasi harus dapat melewati nilai ambang tertentu sehingga dapat diteruskan menjadi keluaran untuk dikirim ke *neuron* lain.

### 2.4. Algoritme Extreme Learning Machine (ELM)

Algoritme *Extreme Learning Machine* (ELM) merupakan algoritme *feedforward* dan tergolong baru dalam jaringan syaraf tiruan yang seringkali disebut sebagai *Single Layer Feedforward Neural Networks* (SLFNs). Kecepatan pembelajaran terhadap data pada algoritme ini ribuan kali lebih cepat daripada algoritme lain dan mendapatkan hasil generalisasi yang lebih baik. Selain itu, algoritme ini cenderung menghasilnya nilai error yang kecil (Huang, Zhu and Siew, 2006).

Pada algoritme ini, terdapat *input layer*, *hidden layer*, serta *output layer* yang dimana masing-masing memiliki *neuron-neuron* yang terhubung satu sama lain. Penggunaan bobot dan bias yang dibangkitkan secara acak diperlukan oleh algoritme ini untuk mempercepat proses pembelajaran (Sun et al., 2008). Selain mempercepat proses pembelajaran, pembangkitan acak ini dilakukan untuk menghindari hasil prediksi yang tidak stabil. Arsitektur ELM ditunjukkan pada Gambar 2 dibawah ini.

Gambar 2. Struktur Algoritme ELM



Pada algoritme ini, terdapat beberapa proses yaitu normalisasi, proses *training*, *testing*, denormalisasi serta evaluasi menggunakan *mean average percentage error* (MAPE).

#### 2.4.1. Normalisasi Data

Normalisasi adalah suatu proses untuk menyamakan satuan pada data yang memiliki rentang nilai yang luas serta variasi data yang beragam. Dengan normalisasi, akan didapatkan data yang memiliki satuan yang sama dengan nilai yang kecil serta terbebas dari *outlier*. Proses normalisasi data dalam penelitian ini menggunakan metode *Min-Max Normalization*. Rumus normalisasi data dapat dilihat pada Persamaan (1).

$$x' = \frac{x - \min}{\max - \min} \tag{1}$$

Keterangan:

- $x'$  = Nilai hasil normalisasi data
- $x$  = Nilai asli data
- $\min$  = Nilai minimum dari dataset
- $\max$  = Nilai maksimum dari dataset

#### 2.4.2. Proses Training

Dalam ELM, terdapat proses pembelajaran (*training*) yang bertujuan untuk melatih sistem sehingga menghasilkan prediksi yang akurat. Berikut ini adalah tahapan-tahapan dalam proses *training* (Liang et al., 2006).

1. Bangkitkan bobot dan bias secara acak dengan rentang yang sudah digunakan.
2. Mendapatkan nilai *output hidden layer* (H) menggunakan fungsi aktivasi yang telah ditentukan yang berbentuk matriks. Untuk mendapatkannya, kita perlu mendapatkan  $H_{init}$ . Rumus persamaan untuk mendapatkan  $H_{init}$  ditunjukkan pada persamaan (2).

$$H_{init} = X * Wt \tag{2}$$

Keterangan:

- $H_{init}$  = Nilai dari hasil perkalian

matriks  $X$  dengan  $Wt$ .

$X$  = Nilai dari data

$Wt$  = Nilai dari bobot yang di *transpose*.

- Setelah mendapatkan  $H_{init}$ , langkah selanjutnya adalah mendapatkan *output hidden layer* ( $H$ ) dengan menggunakan fungsi aktivasi. Untuk fungsi aktivasi, dalam persamaan (3) ini digunakan fungsi aktivasi sigmod biner.

$$H = \frac{1}{1+e^{-(H_{init}+bias)}} \quad (3)$$

Keterangan:

$H$  = Nilai *ouput hidden layer*.

$e$  = Bilangan euler dengan nilai pembulatan 2,718.

$H_{init}$  = Nilai Hinit yang akan dimasukkan kedalam fungsi aktivasi.

$bias$  = Nilai bias yang dibangkitkan secara acak sesuai jumlah fitur.

- Menghitung matriks *moore-penrose pseudo inverse*. Berikut ini merupakan rumus persamaan untuk mendapatkan matriks *moore-penrose pseudo inverse*. Dalam algoritme ini, matriks *moore-penrose pseudo inverse* digunakan karena kemampuannya untuk mendapatkan nilai *inverse* dari matriks yang memiliki ordo berbeda (Barata and Hussein, 2012).

$$H^+ = (H^T * H)^{-1} * H^T \quad (4)$$

Keterangan:

$H^+$  = Matriks *moore-penrose pseudo inverse* dengan fungsi aktivasi.

$(H^T * H)^{-1}$  = Matriks *inverse* perkalian  $H$  *transpose* dengan *output layer* menggunakan fungsi aktivasi.

- Menghitung nilai *output weight*. Berikut ini merupakan persamaan untuk menghitung *output weight* pada persamaan (5).

$$\beta = H^+ * T \quad (5)$$

Keterangan:

$\beta$  = Nilai matriks *output weight*.

$H^+$  = Nilai matriks *moore-penrose pseudo inverse*.

$T$  = Nilai matriks target.

- Menghitung nilai *output layer* pada proses *training*. Berikut ini adalah persamaan untuk menghitungnya.

$$Y = H * \beta \quad (6)$$

Keterangan:

$Y$  = Nilai matriks *output layer*.

$H$  = Nilai *output hidden layer* menggunakan fungsi aktivasi.

$\beta$  = Nilai matriks *output weight*.

### 2.4.3. Proses *Testing*

Tujuan dari proses ini adalah untuk memvalidasi nilai prediksi sekaligus mengevaluasi algoritme ELM dari proses *training* yang sudah dilakukan sebelumnya. Langkah-langkahnya kurang lebih sama dengan proses *training*, yang membedakan adalah tidak adanya perhitungan matriks *moore-penrose pseudo inverse* serta tidak adanya perhitungan *output weight*. Berikut ini adalah langkah-langkah pada proses *testing*.

- Membangkitkan bobot dan bias secara acak.
- Menghitung matriks  $H_{init}$  menggunakan Persamaan (2).
- Menghitung matriks  $H$  menggunakan fungsi aktivasi yang sudah ditentukan sesuai dengan Persamaan (3).

Menghitung nilai *output layer* pada proses *testing* menggunakan Persamaan (6).

### 2.4.4. Denormalisasi Data

Proses denormalisasi data ini bertujuan untuk mendapatkan nilai aktual hasil prediksi. Berikut ini merupakan persamaan untuk mendapatkannya.

$$x = x' * (max - min) + min \quad (7)$$

Keterangan:

$x$  = Nilai hasil prediksi setelah denormalisasi.

$x'$  = Nilai hasil prediksi sebelum denormalisasi.

$min$  = Nilai minimum dari dataset.

$max$  = Nilai maksimum dari dataset.

### 2.4.5. Mean Average Percentage Error (MAPE)

*Mean Absolute Percentage Error* (MAPE)

merupakan suatu metode evaluasi yang sangat umum digunakan untuk mengukur seberapa tepat atau akurat suatu prediksi. MAPE menjadi umum karena kemampuannya untuk mempresentasikan nilai *error* yang mudah dipahami dibandingkan dengan metode yang lain (Kim and Kim, 2016).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| * 100 \tag{8}$$

Keterangan:

$\hat{y}_i$  = Hasil prediksi.

$y_i$  = Nilai aktual.

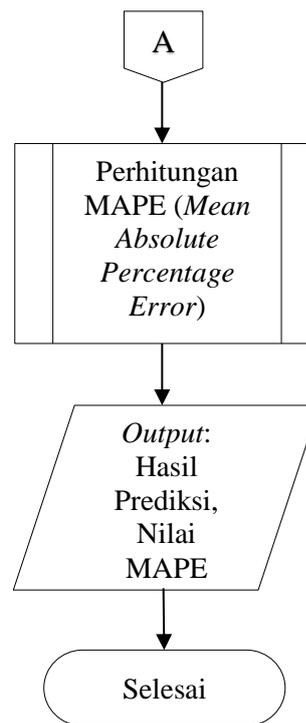
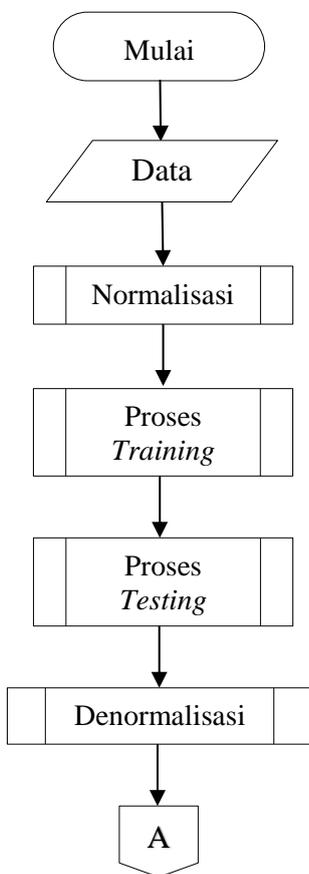
$n$  = banyaknya data yang diuji.

### 3. METODOLOGI

#### 3.1. Pengumpulan Data

Untuk pengumpulan data, data yang diambil berasal dari data pergerakan harian nilai tukar Bitcoin dari laman *investing.com*. Data ini merupakan data *time series* harian dari tanggal 1 Mei 2018 sampai dengan 8 Agustus 2018.

#### 3.2. Diagram Alir Sistem



Gambar 3. Diagram Alir Sistem

Berdasarkan diagram alir pada gambar 1, proses pada sistem dapat dijabarkan sebagai berikut.

1. Data yang akan diolah akan dinormalisasi terlebih dahulu.
2. Setelah itu, data tersebut akan masuk pada tahap *training*, dan *testing*.
3. Setelah melakukan kedua tahap tersebut, data tersebut akan dinormalisasi untuk mendapatkan nilai aktualnya.
4. Setelah mendapatkan nilai aktual, maka dilakukan evaluasi menggunakan MAPE.
5. Hasil akhir dari sistem ini adalah hasil prediksi beserta nilai MAPE.

### 4. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Pengaruh Jumlah Fitur

Pengujian pengaruh jumlah fitur dilakukan untuk mengetahui pengaruh jumlah fitur terhadap nilai kesalahan (*error*) menggunakan MAPE. Untuk pengujiannya sendiri, jumlah fitur yang digunakan mulai dari 2 sampai dengan 10 fitur. Fitur yang digunakan adalah jumlah hari sebelum hari yang akan diprediksi. Pengujian dilakukan menggunakan fungsi aktivasi *sigmoid biner*, lalu jumlah *hidden layer* 2, jumlah data latih 80% dari keseluruhan data dan sisanya untuk data uji, serta rentang bobot [-1,1]. Untuk

lebih jelasnya akan ditampilkan pada Tabel 1.

Tabel 1. Pengujian Pengaruh Jumlah Fitur

Jumlah Fitur (Hari)	Rerata MAPE (%)
2	4.503
3	5.670
4	5.613
5	5.894
6	8.540
7	6.891
8	9.420
9	8.081
10	9.774

Berdasarkan pengujian yang telah dilakukan terhadap masing-masing jumlah fitur yang ditunjukkan pada Tabel 1, menunjukkan nilai kesalahan terkecil pada pengujian jumlah fitur dimiliki oleh pengujian pada jumlah fitur 2. Hal ini disebabkan karena jumlah data yang dimiliki saat jumlah fitur 2 lebih banyak daripada yang lain. Hal ini tentu saja berpengaruh terhadap proses pelatihan karena semakin banyak data latih dapat memperkecil nilai kesalahan.

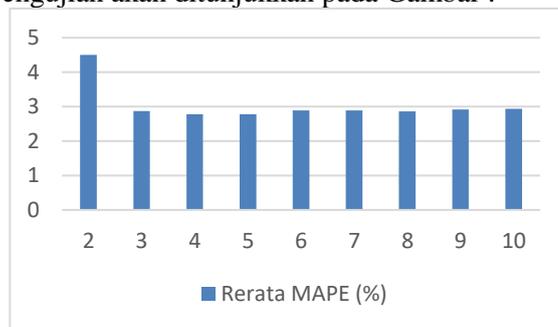
### 3.2. Pengujian Pengaruh Jumlah Hidden Neuron

Pengujian pengaruh jumlah hidden neuron dilakukan untuk mencari berapa jumlah *hidden neuron* terbaik yang dapat memberikan nilai kesalahan terkecil pada prediksi nilai *cryptocurrency* bitcoin menggunakan algoritme ELM ini. Untuk melakukannya, kita harus menentukan berapa saja jumlah *hidden neuron* yang akan diuji. Untuk pengujian ini, jumlah *hidden neuron* yang akan diuji berjumlah mulai dari 2 sampai dengan 10. Serta pengujian ini dilakukan menggunakan fungsi aktivasi *sigmoid biner*, lalu jumlah fitur 2, jumlah data latih 80% dari keseluruhan data dan sisanya untuk data uji, serta rentang bobot [-1,1]. Untuk lebih jelasnya akan ditampilkan pada Tabel 2.

Tabel 2. Pengujian Pengaruh Jumlah Hidden Neuron

Jumlah Hidden Neuron	Rerata MAPE (%)
2	4.503
3	2.871
4	2.774
5	2.778
6	2.886
7	2.891
8	2.863
9	2.915
10	2.936

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 2, Rerata MAPE terkecil untuk pengujian jumlah *hidden neuron* terdapat pada jumlah hidden neuron sebanyak 4 dengan nilai rerata MAPE sebesar 2.744. Hal ini menunjukkan bahwa tidak selalu jumlah *hidden neuron* yang banyak dapat memberikan hasil prediksi yang baik. Untuk visualisasi hasil pengujian akan ditunjukkan pada Gambar .



Gambar 4. Hasil Pengujian Jumlah Hidden Neuron

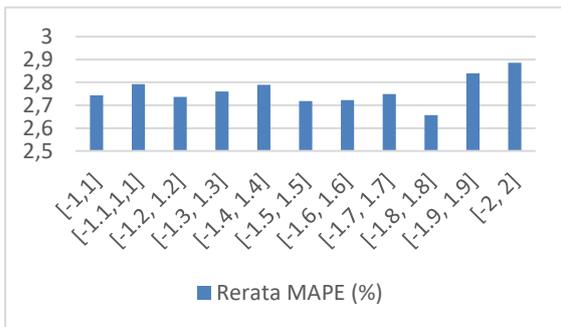
### 3.3. Pengujian Pengaruh Rentang Bobot

Pengujian pengaruh rentang bobot fungsi aktivasi sigmoid biner untuk mencari rentang bobot terbaik pada fungsi aktivasi sigmoid biner yang dapat memberikan nilai kesalahan terkecil prediksi nilai *cryptocurrency* bitcoin menggunakan algoritme ELM ini. Untuk melakukannya, kita harus menentukan jumlah serta rentang bobot yang akan diuji. Pada pengujian ini, jumlah rentang bobot adalah 10 dengan rentang antara -2 sampai dengan 2. Pengujian ini dilakukan menggunakan fungsi aktivasi *sigmoid biner*, dengan jumlah fitur sebanyak 2, jumlah *hidden neuron* sebanyak 4. Setiap rentang bobot akan diuji sebanyak 10 kali yang kemudian akan dicari rerata nilai kesalahannya. Untuk lebih jelasnya akan ditampilkan pada Tabel 3.

Tabel 3. Pengujian Pengaruh Rentang Bobot

Rentang Bobot	Rerata MAPE (%)
[-1,1]	2.744
[-1.1,1,1]	2.792
[-1.2, 1.2]	2.737
[-1.3, 1.3]	2.760
[-1.4, 1.4]	2.790
[-1.5, 1.5]	2.718
[-1.6, 1.6]	2.723
[-1.7, 1.7]	2.749
[-1.8, 1.8]	2.657
[-1.9, 1.9]	2.840
[-2, 2]	2.886

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 3, Rerata MAPE terkecil untuk pengujian ini terdapat pada rentang bobot [-1.8, 1.8] dengan nilai 2.657. Hal ini menunjukkan bahwa rentang bobot berpengaruh terhadap nilai kesalahan yang dihasilkan. Untuk visualisasi hasil pengujian akan ditunjukkan pada Gambar 4.



Gambar 4. Hasil Pengujian Rentang Bobot

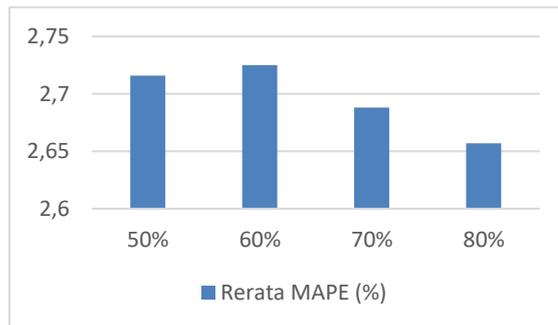
### 3.4. Pengujian Pengaruh Jumlah Data Latih dengan Data Konstan

Pengujian jumlah data latih dengan data konstan dilakukan untuk mencari berapa persentase data latih terbaik yang dapat menghasilkan nilai MAPE terkecil untuk memprediksi nilai *cryptocurrency* bitcoin menggunakan algoritme ELM. Pengujian ini dilakukan dengan melakukan 10 kali percobaan untuk setiap nilai persentase yaitu 50%, 60%, 70%, 80%, dan 90%. Sementara persentase jumlah data uji dibuat konstan yaitu sebesar 20%. Lalu untuk setiap nilai persentase akan didapatkan nilai rerata MAPE. Untuk lebih jelasnya akan ditampilkan pada Tabel 4.

Tabel 4. Pengujian Jumlah Data Latih dengan Data Konstan

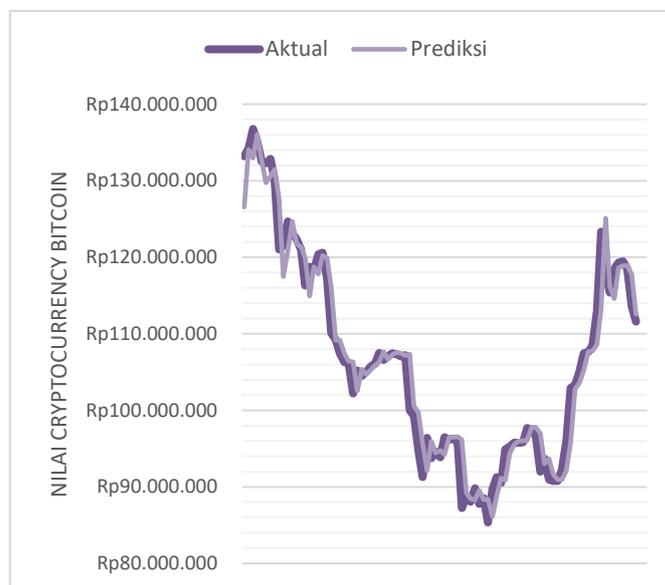
Persentase Jumlah Data Latih	Rerata MAPE (%)
50%	2.716
60%	2.725
70%	2.688
80%	2.657

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 4, persentase jumlah data latih yang menghasilkan rerata nilai MAPE terkecil didapat pada persentase jumlah data latih sebesar 90%. Hal ini menunjukkan bahwa semakin besar jumlah data latih yang digunakan dapat menghasilkan nilai MAPE yang semakin kecil yang menandakan bahwa hasil prediksi mendekati nilai asli. Untuk visualisasi hasil pengujian akan ditunjukkan pada Gambar 5.



Gambar 5. Hasil Pengujian Pengaruh Jumlah Data Latih dengan Data Konstan

Untuk melihat grafik perbandingan hasil prediksi dengan data aktual, ditunjukkan pada Gambar 6.



Gambar 6. Perbandingan hasil prediksi dengan data aktual

## 5. KESIMPULAN

Berdasarkan hasil pengujian dan analisis terhadap sistem prediksi *cryptocurrency* bitcoin menggunakan algoritme *Extreme Learning Machine* (ELM), maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritme *Extreme Learning Machine* (ELM) dapat digunakan untuk memprediksi nilai *cryptocurrency* bitcoin dengan jumlah *hidden neuron* terbaik sebanyak 4.
2. Jumlah fitur terbaik untuk memprediksi nilai *cryptocurrency* bitcoin menggunakan algoritme *Extreme Learning Machine* (ELM) sebanyak 2 fitur.

3. Rentang bobot fungsi aktivasi sigmoid biner terbaik untuk memprediksi nilai *cryptocurrency* bitcoin menggunakan algoritme *Extreme Learning Machine* (ELM) adalah [-1.8, 1.8] dengan persentase data latih 80% dan nilai rerata MAPE terkecil sebesar **2.657%**.

#### DAFTAR PUSTAKA

- Barata, J.C.A. and Hussein, M.S., 2012. The Moore-Penrose Pseudoinverse: A Tutorial Review of the Theory. *Brazilian Journal of Physics*, 42(1–2), pp.146–165.
- Huang, G. Bin, Zhu, Q.Y. and Siew, C.K., 2006. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), pp.489–501.
- Jauhari, D., Cholissodin, I. and Dewi, C., 2017. Prediksi Nilai Tukar Rupiah Indonesia Terhadap Dolar Amerika Serikat Menggunakan Metode Recurrent Extreme Learning Machine Neural Network. 1(11), pp.1188–1197.
- Kim, S. and Kim, H., 2016. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, [online] 32(3), pp.669–679. Available at: <<http://dx.doi.org/10.1016/j.ijforecast.2015.12.003>>.
- Liang, N.-Y., Huang, G.-B., Saratchandran, P. and Sundararajan, N., 2006. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks*, [online] 17(6), pp.1411–1423. Available at: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4012031%5Cnpapers3://publication/doi/10.1109/TNN.2006.880583>>.
- Mosabeth, C., Furqon, M.T. and Wihandika, R.C., 2018. Prediksi Harga Pasar Daging Sapi Di Kota Malang Dengan Menggunakan Metode Extreme Learning Machine ( ELM ). 2(12), pp.6362–6369.
- Permatasari, A.R. and Rahayudi, B., 2018. Estimasi Hasil Produksi Benih Tanaman Kenaf ( *Hibiscus Cannabinus L .* ) Menggunakan Metode Extreme Learning Machine ( ELM ) Pada Balai Penelitian Tanaman Pemanis dan Serat ( Balittas ). 2(11), pp.5475–5483.
- Pratama, A.H., 2017. [Update] *Bitcoin 101 untuk Para Pemula*. [online] Available at: <<https://id.techinasia.com/fakta-penting-tentang-bitcoin>> [Accessed 10 Aug. 2018].
- Radityo, A., Munajat, Q. and Budi, I., 2017. Prediction of Bitcoin Ex change Rate to Americ an Dollar Using Artificial Neural Network Methods. *ICACISIS*.
- Sun, Z.L., Choi, T.M., Au, K.F. and Yu, Y., 2008. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, [online] 46(1), pp.411–419. Available at: <<http://dx.doi.org/10.1016/j.dss.2008.07.009>>.