

# PERAMALAN INDEKS HARGA SAHAM GABUNGAN MENGGUNAKAN METODE EXTREME LEARNING MACHINE DENGAN OPTIMASI BAT ALGORITHM

Oleh: Imelya Susianti  
Universitas Muhammadiyah Semarang

Article history	Abstract
Submission :	The Composite Stock Price Index (JCI) is an economic index that shows stock price movements. The value of the JCI is one aspect that can be used to describe economic conditions in Indonesia. Therefore, information that can describe how the JCI will be in the future is very much needed. The forecasting method used in this research is the Extreme Learning Machine method with Bat Algorithm Optimization. Extreme learning machine is used because it has advantages in terms of relatively short learning speed and can produce small error values. While the Bat algorithm is used to optimize input weights and bias in the Extreme Learning Machine method. The purpose of this study is to find out the general description of JCI data, JCI forecasting and forecasting accuracy using the Extreme Learning Machine method with Bat Algorithm Optimization. Based on the results of the analysis, the best architecture is the architecture with 5 input neurons, 10 hidden neurons, and 5 bat populations because it produces the smallest MAPE value which is 1,0474%.
Revised :	
Accepted :	
<b>Keyword:</b> <i>Bat Algorithm, Composite Stock Price Index, Extreme Learning Machine, Forecasting</i>	

## Pendahuluan

Salah satu bentuk investasi paling populer saat ini adalah investasi saham. Saham merupakan bukti posisi kepemilikan dalam perusahaan. Hal yang menjadi daya tarik dari investasi saham salah satunya yaitu investasi pada bursa saham rata-rata dapat menghasilkan tingkat pengembalian sebesar 10-30% tiap tahunnya (Bakti, 2014). Namun dalam investasi saham diperlukan kecermatan investor dalam memperkirakan harga saham karena harga saham relatif berubah-ubah setiap waktu. Pergerakan harga saham dapat dilihat pada indeks harga saham. Indeks Harga Saham Gabungan (IHSG) merupakan rangkaian informasi historis dari pergerakan harga saham.

Berdasarkan data yang diperoleh dari [www.investing.com](http://www.investing.com) menunjukkan bahwa nilai IHSG selama tahun 2020 masih belum stabil. Sedangkan kestabilan nilai IHSG sangat penting karena salah satu aspek yang dapat menggambarkan kondisi perekonomian di Indonesia adalah besarnya nilai IHSG. Nilai IHSG yang mengalami penurunan biasanya disebabkan oleh kondisi perekonomian yang sedang mengalami permasalahan dan IHSG yang mengalami peningkatan mengindikasikan

adanya perbaikan kinerja perekonomian (Jordan et al., 2011). Oleh karena itu informasi yang dapat menggambarkan bagaimana keadaan IHSG kedepan sangat diperlukan. Salah satu hal yang dapat dilakukan yaitu memprediksi angka IHSG untuk beberapa periode kedepan atau peramalan.

Terdapat banyak metode yang dapat digunakan dalam melakukan peramalan, salah satunya yaitu metode Jaringan Syaraf Tiruan (JST) yang merupakan jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan sistem syaraf manusia. JST banyak digunakan untuk peramalan karena memiliki kelebihan dalam hal prediksi dan pengenalan pola serta dapat menghasilkan nilai peramalan yang mampu mendekati nilai sebenarnya. Data IHSG merupakan data yang relatif berubah-ubah setiap waktu, sehingga dalam meramalkan IHSG diperlukan suatu metode peramalan dengan waktu komputasi yang cepat. Salah satu metode dalam JST yang dapat digunakan yaitu metode *Extreme Learning Machine* (ELM) karena ELM merupakan jaringan syaraf tiruan *feedforward* dengan *single hidden layer* yang dibuat untuk mengatasi kelemahan-kelemahan dari jaringan syaraf tiruan *feedforward* terutama dalam hal

*learning speed* (Huang et al, 2004). Selain itu peramalan menggunakan ELM dapat memberikan nilai error yang kecil.

Pemilihan *input weight* dan bias pada metode ELM dipilih secara acak atau *random* sedangkan *input weight* dan bias merupakan pembobot yang diberikan pada masing-masing *layer* pada arsitektur metode ELM yang akan mentransformasi nilai *input* menjadi nilai *output* sehingga pemilihan *input weight* dan bias yang acak dapat menyebabkan hasil yang didapat dalam perhitungan menjadi kurang maksimal (Handika, 2016). Oleh karena itu untuk mendapatkan hasil yang lebih optimal metode ELM dapat digabungkan dengan algoritma optimasi. Algoritma Optimasi digunakan untuk mengoptimasi *input weight* dan bias pada metode ELM. Salah satu algoritma optimasi yang dapat digunakan yaitu *Bat Algorithm* hal ini karena *Bat Algorithm* merupakan salah satu algoritma optimasi metaheuristik yang dapat memberikan performa yang lebih baik dibandingkan beberapa algoritma optimasi lain seperti Algoritma Genetika, *Particle Swarm Optimization* dan *Geometric Particle Swarm Optimization* (Taha et al., 2013).

Beberapa penelitian terkait ELM, IHSG dan *Bat Algorithm* telah banyak dilakukan, diantaranya yang dilakukan oleh Fifi Oktaviani Putri pada tahun 2019 yang berjudul “Prediksi Harga Emas di Indonesia Menggunakan *Extreme Learning Machine* (ELM)”, dalam penelitian ini menghasilkan MAPE *training* sebesar 0.5974% dan MAPE *testing* sebesar 0.6208%. Penelitian lain oleh Seng Hansum pada tahun 2012 yang berjudul “Peramalan Data IHSG Menggunakan *Fuzzy Time Series*”, dalam penelitian ini menghasilkan tingkat kesalahan menggunakan MSE sebesar 5.404564 dan menggunakan MAPE sebesar 0.04777038%. Selanjutnya penelitian oleh I Putu Susila Handika, Ida Ayu Giriantari, Agus Dharma pada tahun 2016 dengan judul "Perbandingan Metode *Extreme Learning Machine* dan *Particle Swarm Optimization Extreme Learning Machine* untuk Peramalan Jumlah Penjualan Barang", dalam penelitian ini nilai MSE dan range error yang dihasilkan oleh metode PSO ELM lebih kecil dibanding metode ELM. Serta penelitian yang dilakukan oleh Bayu Prabawa, Jondri Nasri dan Mahmud Dwi Sulistyio pada tahun 2015 dengan

judul “Prediksi Harga Saham dengan menggunakan Metode *Autoregressive* dan Algoritma Kelelawar”, dalam penelitian ini menghasilkan tingkat error dibawah 4%.

Berdasarkan uraian di atas, penelitian yang dilakukan dengan menggabungkan algoritma optimasi mendapatkan hasil yang lebih optimal. Sehingga pada penelitian ini akan digunakan metode ELM dengan optimasi *Bat Algorithm* untuk meramalkan nilai IHSG. Harapannya agar mendapatkan hasil peramalan yang lebih optimal dengan nilai kesalahan yang kecil.

## Landasan Teori

### Indeks Harga Saham Gabungan

Indeks Harga Saham Gabungan (IHSG) merupakan angka indeks harga saham yang sudah disusun dan dihitung dengan menghasilkan *trend*, dimana angka indeks adalah angka yang diolah sedemikian rupa sehingga dapat digunakan untuk membandingkan kejadian yang berupa perubahan harga saham dari dari waktu ke waktu (Jogiyanto, 2013 :147).

### Peramalan

Peramalan merupakan proses untuk memperkirakan suatu nilai pada masa yang akan datang dengan memperhatikan data atau informasi masa lalu maupun saat ini. Salah satu manfaat dari peramalan yaitu untuk melihat gambaran-gambaran tentang masa yang akan datang sehingga dapat mengantisipasi apa yang akan terjadi.

### Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan adalah teknik untuk memodelkan pemrosesan informasi berdasarkan kemampuan sistem syaraf pada otak manusia. JST menggunakan cara kerja dari syaraf pada otak manusia untuk pengolahan informasi dengan menyelesaikan sejumlah proses perhitungan pada komputer (Kusumadewi, 2003).

### *Extreme Learning Machine*

*Extreme Learning Machine* (ELM) merupakan pengembangan dari metode jaringan syaraf tiruan yang pertama kali ditemukan oleh Huang, Zhu, dan Siew pada tahun 2004. Kelebihan dari metode ini yaitu hanya

menggunakan satu *hidden layer* dan penggunaan *input weight* dan bias mampu memberikan hasil prediksi yang lebih akurat dengan *learning speed* yang lebih cepat dibandingkan dengan jaringan *feedforward* biasa (Huang, et. al., 2004). Secara umum fungsi matematis metode ELM dengan jumlah *hidden nodes* sebanyak  $N$  dan *activation function*  $g(x)$  dapat dilihat pada persamaan (1)

$$\sum_{i=1}^N \beta_i g_i(x_j) = \sum_{i=1}^N \beta_i g_i(W_i X_j + b_i) \quad (1)$$

Keterangan :

$j = 1, 2, \dots, N$

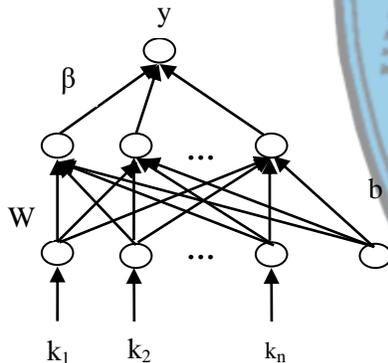
$W_i = (W_{i1}, W_{i2}, \dots, W_{in})^T$ , vektor dari *weight* yang menghubungkan *hidden nodes* ke- $i$  dan *input nodes*

$\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})^T$ , vektor dari bobot yang menghubungkan *hidden nodes* ke- $i$  dan *output nodes*

$b_i = \text{threshold}$  dari *hidden nodes* ke- $i$

$W_i X_i = \text{inner product}$  dari  $W_i$  dan  $X_i$

Struktur jaringan pada metode ELM dapat dilihat pada Gambar 1.



**Gambar 1 Struktur Jaringan Metode ELM**

### Normalisasi Data

Normalisasi data merupakan proses perubahan bentuk data menjadi nilai dalam batas 0-1. Proses ini ditujukan untuk menyesuaikan data *input* terhadap data *output*. Menurut Siang (2005) perhitungan normalisasi data berdasarkan pada persamaan (2).

$$Y'_t = \frac{0,8(Y_t - \min(x))}{(\max(x) - \min(x))} + 0,1 \quad (2)$$

Keterangan :

$Y'_t = \text{Data ke-t dari hasil normalisasi}$

$Y_t = \text{Data ke-t sebelum di normalisasi}$

$\min(x) = \text{Nilai minimum data sebelum di normalisasi}$

$\max(x) = \text{Nilai maksimum data sebelum di normalisasi}$

### Proses Training

Tujuan dari proses *training* yaitu untuk mendapatkan *output weight* dengan tingkat kesalahan yang rendah. Menurut Chandra, et. al. (2018) langkah-langkah proses *training* metode ELM yaitu :

1. Inisialisasi *input weight* dan bias dengan bilangan acak yang kecil secara *random*
2. Menghitung *output* di *hidden layer* menggunakan fungsi aktivasi. Perhitungan *output hidden layer* berdasarkan pada persamaan (3).

$$H_{net_j} = \left( \sum_{k=1}^n x_{ik} w_{jk} \right) + b_j \quad (3)$$

Keterangan :

$H_{net_j} = \text{nilai output hidden layer}$

$i = (1, 2, \dots, N)$ ,  $N$  adalah jumlah data

$j = (1, 2, \dots, \bar{N})$ ,  $\bar{N}$  adalah jumlah *hidden neuron*

$k = \text{Jumlah input neuron}$

$w = \text{Bobot input dengan ukuran ordo matriks, hidden neuron} \times \text{input neuron}$

$x = \text{data input}$

$b = \text{bias}$

3. Menghitung fungsi aktivasi *sigmoid biner* berdasarkan pada persamaan (4).

$$H_{ij} = \frac{1}{1 + e^{-(H_{net_j})}} \quad (4)$$

Keterangan :

$H_{ij} = \text{Matriks dari fungsi aktivasi output pada hidden layer}$

$e = \text{Eksponensial}$

$H_{net_j} = \text{nilai output hidden layer}$

4. Menghitung Matriks *Moore-Penrose Generalized Inverse/ Moore-Penrose Pseudo Inverse* ( $H^+$ ) berdasarkan pada persamaan (5).

$$H^+ = (H^T H)^{-1} H^T \quad (5)$$

Keterangan :

$H^+ = \text{Matriks Moore-Penrose Generalized Inverse.}$

$H^T = \text{Matriks H yang telah ditranspose}$

$H = \text{Matriks H hasil keluaran hidden layer yang telah diaktivasi}$

$(H^T H)^{-1} = \text{Inverse dari perkalian matriks } H^T \text{ dengan } H$

5. Menghitung *output weight* dari *hidden layer* ke *output layer* berdasarkan pada persamaan (6).

$$\beta = H^+T \quad (6)$$

Keterangan :

$\beta$  = Matriks *output weight* dari hidden layer ke *output layer*

$H^+$  = Matriks *Moore-Penrose Generalized Inverse*

$T$  = Matriks Target.

### Proses *Testing*

Proses *testing* bertujuan untuk mengevaluasi kemampuan metode ELM dalam memprediksi, yang dilakukan berdasarkan *input weight*, bias, dan *output weight* dari perhitungan *training*. Menurut Chandra, et. al. (2018) langkah-langkah proses *testing* yaitu :

1. Inisialisasi *input weight* dan bias yang telah didapatkan dari proses *training*
2. Menghitung semua *output* di *hidden layer* menggunakan fungsi aktivasi menggunakan persamaan (3) dan (4).
3. Menghitung *output* pada *output layer* menggunakan hasil *output weight* dari *hidden layer* ke *output layer* dari proses *training* yang akan menghasilkan nilai prediksi, yang dapat dihitung berdasarkan pada Persamaan (7).

$$y = H\beta \quad (7)$$

Keterangan :

$y$  = *Output* hasil prediksi

$H$  = Matriks  $H$  hasil keluaran *hidden layer* yang telah diaktivasi

$\beta$  = Matriks *output weight* dari *hidden layer* ke *output layer*

### Denormalisasi Data

Denormalisasi data merupakan proses yang bertujuan untuk mengembalikan nilai yang telah dinormalisasi menjadi nilai asli. Menurut Siang (2005) perhitungan denormalisasi data sebagai berikut :

$$\hat{Y}_t = \frac{(\hat{Y}'_t - 0.1)(\max(x) - \min(x))}{0.8} + \min(x) \quad (8)$$

Keterangan :

$\hat{Y}'_t$  = Nilai hasil normalisasi

$\hat{Y}_t$  = Hasil denormalisasi

$\min(x)$  = Nilai minimum data sebelum di normalisasi

$\max(x)$  = Nilai maksimum data sebelum di normalisasi

### Bat Algorithm

*Bat Algorithm* (BA) adalah salah satu algoritma *metaheuristik* yang diciptakan oleh Yang dan Xin-She pada tahun 2010 yang diadaptasi dari perilaku *echolocation* kelelawar dalam mencari makanan. Kemampuan *echolocation* ini membuat kelelawar dapat menemukan mangsa dan dapat membedakan rintangan dengan sumber makanan.

Menurut Yang dan Xin-She (2010) implementasi *Bat Algorithm* untuk optimasi sebagai berikut :

#### 1. Posisi

Posisi *bat* merupakan representasi dari tiap solusi. Penyesuaian posisi *bat* dapat dihitung dengan persamaan (9).

$$x_i^t = x_i^{t-1} + v_i^t \quad (9)$$

Keterangan :

$x_i^t$  = posisi *bat* ke -  $i$  pada iterasi ke-  $t$

$x_i^{t-1}$  = posisi *bat* ke -  $i$  pada iterasi  $t-1$

$v_i^t$  = kecepatan *bat* ke -  $i$  pada iterasi ke -  $t$

#### 2. Frekuensi

*Frekuensi* merupakan *elemen* bilangan *real* yang akan mempengaruhi nilai kecepatan. Dihitung dengan persamaan (10).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (10)$$

Keterangan :

$f_i$  = *frekuensi bat* ke-  $i$

$f_{\min}$  = nilai minimum *frekuensi*

$f_{\max}$  = nilai maksimum *frekuensi*

$\beta$  = variabel *konstan* diantara 0 dan 1

#### 3. Loudness

*Loudness* ( $A$ ) merupakan perubahan dalam suatu iterasi yang terjadi selama pencarian lokal di sekitar posisi *bat* terbaik global dan pencarian lokal di setiap *bat*.

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \quad (11)$$

Keterangan :

$x_{\text{new}}$  = posisi *bat* baru

$x_{\text{old}}$  = posisi *bat* sebelumnya

$A^t$  = *loudness* rata-rata semua *bat* dalam satu iterasi

$\varepsilon$  = bilangan acak diantara -1 dan 1

*Loudness* akan berkurang saat *bat* mulai mendekati posisi terbaiknya. Nilai *loudness* dapat dihitung dengan persamaan (12).

$$A_i^{t+1} = a + A_i^t \quad (12)$$

Keterangan :

$A_i^{t+1}$  = *loudness bat* ke -  $i$  untuk iterasi ke  $t + 1$

$A_i^t$  = *loudness bat* ke -  $i$  pada iterasi ke -  $t$

$\alpha$  = variabel konstan diantara 0 dan 1

#### 4. Kecepatan

Kecepatan masing-masing *bat* direpresentasikan dengan bilangan integer positif. Kecepatan pada *bat* dapat dihitung dengan persamaan (13).

$$v_i^t = v_i^{t-1} + (x^* - x_i^t) f_i \quad (13)$$

Keterangan :

$v_i^t$  = kecepatan *bat* ke- i pada iterasi ke- t  
 $v_i^{t-1}$  = kecepatan *bat* ke- i pada iterasi t-1  
 $(x^* - x_i^t)$  = perbedaan posisi *bat* global (\*) dengan *bat* ke- i pada iterasi ke- t.

#### 5. Pulse Rate

*Pulse rate* (r) berperan dalam penentuan waktu pencarian lokal *bat* terbaik global yang dilewati. Ketika *bat* mendekati posisi terbaik, nilai *pulse rate* akan perlahan-lahan berkurang. Nilai *pulse rate* dapat dihitung dengan persamaan (14).

$$r^{t+1} = r^0 + [1 - \exp(-\gamma t)] \quad (14)$$

Keterangan :

$r^{t+1}$  = *pulse rate bat* pada iterasi t+1  
 $r^0$  = nilai *pulse rate* awal pada *bat*  
 $\gamma$  = variabel konstan diantara 0 dan 1

#### Ukuran Kesalahan Peramalan

Ukuran kesalahan yang digunakan dalam penelitian ini adalah nilai rata-rata kesalahan presentase absolut (*Mean Absolute Percentage Error*). MAPE merupakan pengukuran besarnya kesalahan dengan menghitung ukuran presentase penyimpangan antara data aktual dengan data peramalan. Rumus perhitungannya adalah sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\% \quad (15)$$

Keterangan :

n = Jumlah data  
 $\hat{y}_t$  = Nilai prediksi pada periode ke-t  
 $y_t$  = Nilai data aktual pada periode ke-t  
 Menurut Chang, Wang dan Liu (2007) kriteria nilai MAPE adalah sebagai berikut:

**Tabel 1 Kriteria MAPE**

MAPE	Pengertian
<10%	Kemampuan Peramalan Sangat Baik
10%-20%	Kemampuan Peramalan Baik
20%-50%	Kemampuan Peramalan Cukup
>50%	Kemampuan Peramalan Buruk

#### Metode Penelitian

##### Sumber Data

Data yang digunakan pada penelitian ini yaitu data sekunder Indeks Harga Saham Gabungan (IHSG) pada hari kerja dalam bentuk harian dari tanggal 2 Januari 2015 sampai 30 Desember 2020 dengan jumlah data sebanyak 1455 data. Data yang diperoleh akan dibagi menjadi data *training* dan data *testing*. Menurut Agustina (2010) data *training* dan data *testing* dibagi dengan komposisi 80% data *training* dan 20% data *testing*. Sumber data diperoleh dari [www.investing.com](http://www.investing.com).

##### Variabel Penelitian dan Struktur Data

Variabel dalam penelitian ini yaitu Indeks Harga Saham Gabungan (IHSG). Struktur data pada penelitian ini dapat dilihat pada Tabel 2.

**Tabel 2 Struktur Data**

No	Tanggal	Y
1	2 Januari 2015	Y1
2	5 Januari 2015	Y2
3	6 Januari 2015	Y3
⋮	⋮	⋮
⋮	⋮	⋮
1455	30 Desember 2020	Y1455

dimana :

Y = Indeks Harga Saham Gabungan (IHSG)

##### Langkah-langkah Penelitian

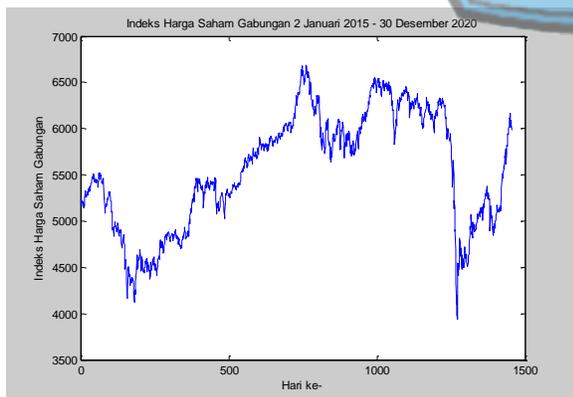
1. Menentukan jumlah *bat* (n) dan menentukan *maximum* iterasi (t).
2. Mendeskripsikan data.
3. Melakukan pembagian data *training* dan data *testing*.
4. Melakukan normalisasi data, untuk standarisasi data pada jarak tertentu menggunakan persamaan (2.5).
5. Menentukan posisi (xi), kecepatan (vi), frekuensi (fi), loudness (Ai), pulse rate (ri) masing-masing *bat* secara acak.
6. Melakukan proses *training*, untuk mendapatkan nilai MAPE sebagai nilai *fitness Bat Algorithm*
7. Jika  $t < \text{maximum iterasi}$  maka update posisi (xi), frekuensi (fi), kecepatan (vi) pada masing-masing *bat* dengan persamaan (2.12), (2.13), dan (2.16).

8. Membangkitkan nilai *random* untuk masing-masing *bat*.
9. Jika nilai *random*  $> r_i$  maka membangkitkan posisi lokal disekitar posisi ( $x_i$ ) yang telah di *update* dengan persamaan (2.14),  
Jika nilai *random*  $< r_i$  dan  $< A_i$  serta  $f(x_i^t) < (f(x^*))$  maka menaikkan nilai  $r_i$  dengan persamaan (2.17) dan mengurangi nilai  $A_i$  dengan persamaan (2.15), kemudian pilih posisi terbaik dari keseluruhan.
10. Melakukan proses *testing*, untuk mengevaluasi proses *training* dalam memprediksi data.
11. Melakukan denormalisasi data, untuk mengembalikan nilai yang telah dinormalisasi menjadi nilai asli.
12. Menghitung nilai hasil prediksi
13. Menghitung nilai kesalahan peramalan menggunakan MAPE (*Mean Absolute Percentage Error*).

## Hasil Penelitian dan Pembahasan

### Analisis Deskriptif

Data yang digunakan pada penelitian ini merupakan data sekunder Indeks Harga Saham Gabungan (IHSG) pada hari kerja dalam bentuk harian dari tanggal 2 Januari 2015 sampai 30 Desember 2020 dengan jumlah data sebanyak 1455 data. Data yang diperoleh bersumber dari [www.investing.com](http://www.investing.com). Plot data IHSG dapat dilihat pada Gambar 2.



**Gambar 2 Plot Data IHSG**

Berdasarkan Gambar 1 dapat dikatakan bahwa data IHSG di Indonesia mengalami fluktuasi pada setiap periode dan terjadi penurunan drastis pada hari ke-1272. Statistika

deskriptif pada data IHSG dapat dilihat pada Tabel 3.

**Tabel 3 Statistik Deskriptif**

Mean	Maximum	Minimum	Standar Deviasi
5549.44	6689.29	3937.63	629.96076

Berdasarkan Tabel 3 dapat diketahui bahwa rata-rata IHSG pada periode 2 Januari 2015 sampai dengan 30 Desember 2020 sebesar 5549.44. Dengan nilai rata-rata dari IHSG sebesar 5549.44, nilai maximum IHSG sebesar 6689.29, nilai minimum IHSG sebesar 3937.63 dan standar deviasi sebesar 629.96076.

### Normalisasi Data

Normalisasi data dilakukan agar data yang diinputkan memiliki nilai dengan range tertentu. Hal ini diperlukan karena fungsi aktivasi yang digunakan pada penelitian ini adalah fungsi sigmoid biner dimana *range* keluaran fungsi aktivasi sigmoid adalah [0,1]. Fungsi sigmoid merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1, sehingga akan lebih baik jika interval yang digunakan lebih kecil yaitu interval [0.1, 0.9]. Perhitungan dilakukan menggunakan persamaan (2).

**Tabel 4 Normalisasi Data**

Tanggal	Normalisasi Data
2 Januari 2015	0.4794
5 Januari 2015	0.4728
6 Januari 2015	0.4580
.	.
.	.
.	.
.	.
30 Desember 2020	0.6935

### Arsitektur Jaringan

Arsitektur jaringan terdiri dari 3 lapisan, yaitu *input layer*, *hidden layer*, dan *output layer*. Penentuan *neuron input* dan *neuron hidden* dalam jaringan syaraf tiruan tidak ada ketentuan tertentu, pada penelitian ini *neuron input* yang digunakan sebanyak 5 neuron dengan asumsi merupakan perwakilan dari 5 hari kerja dalam seminggu. Sedangkan *neuron hidden* pada penelitian ini dibatasi sejumlah 1 sampai 10 berdasarkan *trial dan error*. Selanjutnya neuron dalam *hidden layer* akan dikombinasi dengan

neuron input untuk membangun beberapa arsitektur.

**Tabel 5 Pemilihan Hidden Neuron**

Jumlah Hidden Neuron	MAPE
1	7.5513
2	1.8552
3	0.7794
4	0.9762
5	0.8885
6	0.6838
7	0.6681
8	0.6484
9	0.6407
10	0.6391

Berdasarkan Tabel 5 dapat diketahui bahwa *hidden neuron* dengan jumlah 10 memiliki nilai MAPE yang paling kecil yaitu sebesar 0.6391 sehingga neuron optimal yang digunakan adalah 5 *neuron input* dan 10 *neuron hidden*.

**Inisialisasi Parameter Bat Algorithm**

Tahapan awal yang dilakukan dalam proses optimasi menggunakan *Bat Algorithm* (BA) adalah menentukan parameter BA kemudian menentukan posisi awal untuk sejumlah *bat* sebagai nilai awal *input weight* dan bias. Penentuan posisi awal dilakukan secara random. Parameter BA dalam penelitian ini dapat dilihat pada Tabel 6.

**Tabel 6 Parameter Bat Algorithm**

Parameter	Nilai
Jumlah bat	5
Iterasi Maximum	5
A	0.5
r	0.5

**Proses Training**

Proses *training* pada ELM dilakukan untuk mengevaluasi nilai *fitness* pada *Bat Algorithm*, karena ukuran kesalahan yang digunakan berdasarkan nilai MAPE maka nilai *fitness* dari setiap posisi *bat* akan dihitung berdasarkan nilai MAPE pula dan posisi *bat* dengan nilai MAPE terkecil akan dianggap sebagai posisi yang optimal. Posisi pada setiap *bat* akan di *update* sesuai jumlah iterasi *maximum* dan akan dipilih posisi terbaik yang memiliki nilai MAPE terkecil secara keseluruhan. Berikut hasil nilai *fitness* posisi disetiap *bat* :

**Tabel 7 Posisi Bat Terbaik**

Posisi Bat	MAPE
1	0.6386
2	0.6465
3	0.6413
4	0.6462
5	0.6474

Berdasarkan Tabel 7 diketahui bahwa posisi *bat* yang terbaik berada pada posisi *bat* ke-1 dengan nilai MAPE 0.6386 sehingga nilai *input weight* dan bias yang optimal akan menggunakan nilai yang sesuai pada posisi *bat* ke-1. Berikut merupakan nilai *input weight* dan bias yang optimal :

$$W_{ij} = \begin{bmatrix} 0.2607 & 0.5944 & 0.0225 & 0.4252 & 0.3127 \\ 0.1615 & 0.1788 & 0.4229 & 0.0942 & 0.5985 \\ 0.4709 & 0.6959 & 0.6999 & 0.6385 & 0.0336 \\ 0.0688 & 0.3196 & 0.5309 & 0.6544 & 0.4076 \\ 0.8200 & 0.7184 & 0.9686 & 0.5313 & 0.3251 \\ 0.1056 & 0.6110 & 0.7788 & 0.4234 & 0.0908 \\ 0.2664 & 0.1536 & 0.2810 & 0.4401 & 0.5271 \\ 0.4574 & 0.8754 & 0.5180 & 0.9436 & 0.6377 \\ 0.9577 & 0.2407 & 0.6761 & 0.2890 & 0.6718 \\ 0.6951 & 0.0680 & 0.2548 & 0.2240 & 0.6678 \end{bmatrix}_{10 \times 5}$$

$$b_j = [0.8444 \quad 0.3445 \quad 0.7805 \quad 0.6753 \quad 0.0607 \quad 0.6022 \quad 0.3868 \quad 0.9160 \quad 0.0012 \quad 0.4624]_{1 \times 10}$$

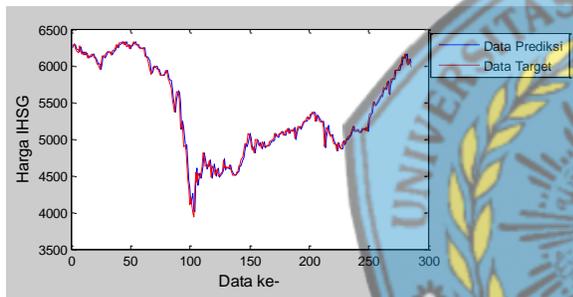
Nilai *input weight* dijelaskan pada matriks  $W_{ij}$  yang merupakan bobot pada *input neuron* ke-  $i$  dan *hidden neuron* ke-  $j$  dimana banyaknya *input weight* sesuai dengan ukuran (jumlah *input neuron* x jumlah *hidden neuron*). Sedangkan nilai bias dijelaskan pada matriks  $b_j$  yang merupakan bias pada *hidden neuron* ke-  $j$  dimana banyaknya bias sesuai dengan ukuran (1 x jumlah *hidden neuron*).

Nilai *input weight* dan bias selanjutnya akan digunakan untuk menghitung *output weight* dari *hidden layer* ke *output layer* ( $\beta$ ) dengan menggunakan perhitungan matriks *Pseudo Inverse* ( $H^+$ ). Nilai  $\beta$  yang didapatkan sebagai berikut :

$$\beta = \begin{bmatrix} 3.1981 \\ 13.2689 \\ -1.2514 \\ -1.3652 \\ 9.2523 \\ -12.6997 \\ 5.1113 \\ -2.6157 \\ -6.7260 \\ -3.8932 \end{bmatrix}_{10 \times 1}$$

## Proses Testing

Proses *testing* dilakukan untuk mengevaluasi kemampuan metode ELM dalam memprediksi. Proses ini menggunakan *input weight* dan bias yang optimal dari *bat algorithm* serta menggunakan *output weight* ( $\beta$ ) yang telah didapatkan dari proses *training*. Perhitungan dilakukan menggunakan persamaan (3), (4), dan (7) sehingga akan menghasilkan suatu nilai prediksi. Nilai prediksi tersebut kemudian akan dilakukan denormalisasi data menggunakan persamaan (8) untuk dikembalikan menjadi nilai asli karena nilai prediksi yang telah didapatkan masih menggunakan data normalisasi. Selanjutnya hasil dari data prediksi akan dibandingkan dengan data target untuk mengetahui kebaikan dari arsitektur yang diperoleh. Perbandingan data prediksi dengan data target dapat dilihat pada Gambar 3.



**Gambar 3 Perbandingan Data Prediksi dan Data Target**

Berdasarkan Gambar 3 diketahui bahwa arsitektur yang diperoleh merupakan arsitektur yang baik karena data prediksi yang dihasilkan mendekati data sebenarnya dimana perbedaan dari data prediksi dan data target tidak berbeda jauh dan menghasilkan nilai MAPE sebesar 1.0474% yang mana nilai tersebut  $<10\%$  yang menunjukkan bahwa arsitektur yang digunakan memberikan keakuratan peramalan yang sangat baik dengan akurasi peramalan sebesar 98.9526% sehingga arsitektur tersebut dapat digunakan untuk proses peramalan IHSN pada periode kedepan.

## Peramalan Nilai IHSN

Arsitektur yang diperoleh berdasarkan proses *training* dan *testing* yaitu arsitektur dengan 5 neuron pada *input layer*, 10 neuron pada *hidden neuron*, dan 5 populasi *bat*. Arsitektur tersebut merupakan arsitektur yang baik karena perbedaan dari data aktual dan data

prediksi tidak berbeda jauh. Peramalan nilai IHSN untuk 3 periode kedepan yaitu tanggal 4 Januari 2021, 5 Januari 2021, dan 6 Januari 2021 yang diperoleh menunjukkan hasil bahwa nilai IHSN selalu naik disetiap periodenya. Hasil peramalan dapat dilihat pada Tabel 8.

**Tabel 8 Hasil Prediksi IHSN 3 Periode ke Depan**

Tanggal	Peramalan
04/01/2021	5981.96
05/01/2021	5988.78
06/01/2021	5991.40

## Simpulan dan Saran

### Simpulan

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat diambil beberapa kesimpulan sebagai berikut :

1. Data IHSN di Indonesia pada periode 2 Januari 2015 sampai dengan 30 Desember 2020 mengalami fluktuasi pada setiap periodenya dan terjadi penurunan drastis pada periode ke-1320. Nilai rata-rata IHSN sebesar 5549.44, nilai maximum sebesar 6689.29, nilai minimum sebesar 3937.63 dan standar deviasi sebesar 629.96076.
2. Hasil peramalan IHSN pada 3 periode kedepan yaitu tanggal 4 Januari 2021, 5 Januari 2021, dan 6 Januari 2021 yang diperoleh menunjukkan bahwa peramalan IHSN selalu naik disetiap periodenya.
3. Nilai MAPE yang dihasilkan sebesar 1.0474% nilai tersebut  $<10\%$  yang menunjukkan bahwa metode yang digunakan cocok untuk meramalkan IHSN karena hasil peramalan yang didapatkan memberikan keakuratan peramalan yang sangat baik dengan akurasi peramalan sebesar 98.9526%.

### Saran

Berdasarkan hasil penelitian ini, maka penulis memberikan saran dan bahan pertimbangan untuk penelitian yang akan dilakukan selanjutnya yaitu :

1. Penelitian ini menggunakan optimasi *Bat Algorithm*, untuk penelitian selanjutnya dapat melakukan pengembangan lain dengan membandingkan beberapa optimasi

lain agar mendapatkan hasil peramalan yang lebih baik.

2. Penelitian ini mengoptimasi bobot dan bias pada *Extreme Learning Machine* untuk penelitian selanjutnya dapat melakukan optimasi pada *hidden layer* agar meningkatkan hasil peramalan yang lebih baik.

#### Daftar Pustaka

- Aditya, P., R. Suyanto. 2012. Analisis dan Implementasi *Bat Algorithm* untuk *Continuous Optimization Task*. Universitas Telkom.
- Agustina, I. D. 2010. Penerapan Metode *Extreme Learning Machine* untuk Peramalan Permintaan. *Skripsi*. Institut Teknologi uluh ember. Surabaya.
- Ainurrohmah, I. 2020. *Hybrid Jaringan Syaraf Tiruan Metode Extreme Learning Machine dengan Firefly Algorithm dan Simulated Annealing* untuk Meramalkan Harga Saham. *Skripsi*. Universitas Airlangga. Surabaya.
- Alfiyatin A.N., W.F. Mahmudy, C.F. Ananda, dan Y.P. Anggodo. 2019. Penerapan *Extreme Learning Machine* (ELM) untuk Peramalan Laju Inflasi Di Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer* 6 (2), 179-186.
- Anoraga, P., P. Pakarti. 2008. Pengantar Pasar Modal. 3rd ed. PT Rineka Cipta. Jakarta.
- Arifia.A, Mulyono, M. Irzal. 2017. Peramalan Indeks Harga Saham Gabungan (IHSG) dengan Metode *Extreme Learning Machine* (ELM). *JMT: Jurnal Matematika dan Terapan* 1 (1), 22-32.
- Ashar, N. M., I. Cholissodin, C. Dewi. 2018. Penerapan Metode *Extreme Learning Machine* (ELM) untuk Memprediksi Jumlah Produksi Pipa yang Layak (Studi Kasus Pada PT. KHI *Pipe Industries*). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN 2548, 964X.
- Bakti, L.S. 2014. Prediksi Pergerakan Harga Saham Menggunakan Algoritma Memetika (*Prediction of Stock ket Price Movement Using Memetic Algorithm*). Universitas Telkom.
- Chandra, K. A., I.M. Widiartha., A. Muliantara. 2016. Analisis dan Implementasi Algoritma Kelelawar Sebagai *Fitur Selektor* dalam Klasifikasi *Dermatology*. *Jurnal Ilmu Komputer* 9 (2).
- Chandra, R.A., E. Santoso., S. Adinugroho. 2018. Optimasi Metode *Extreme Learning Machine* dalam Penentuan Kualitas Air Sungai Menggunakan Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN 2548, 964X.
- Chang, P. C., Wang, Y. W., Liu, C. H. 2007. *The Development of A Weighted Evolving Fuzzy Neural Network for PCB Sales Forecasting*. *Expert Systems with Applications*, Vol 32, No.1 : Hal 86-89.
- Daranti, E. Y., B.D. Setiawan, F.A. Bachtiar. 2018. *Particle Swarm Optimization* untuk Optimasi Bobot *Extreme Learning Machine* dalam Memprediksi Produksi Gula Kristal Putih Pabrik Gula Candi Baru-Sidoarjo. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN 2548, 964X.
- Desvina, A. P., N. Rahmah. 2017. Penerapan Metode ARCH/GARCH dalam Peramalan Indeks Harga Saham Sektorial. *Jurnal Sains Matematika dan Statistika: Jurnal Hasil Penelitian Matematika, Statistika, dan Aplikasinya* 2 (1), 1-10.
- Dwipa, S., N. Mursetya. 2016. Identifikasi Model I-GARCH (*Integrated Generalized Autoregressive Conditionally Heterocedastic*) untuk Peramalan *Value at Risk*. *Jurnal Derivat* 3 (1), 25-38.
- Eliyawati, W.Y., R.R. Hidayat, D.F. Azizah. 2014. Penerapan Model GARCH (*Generalized Autoregressive Conditional Heteroscedasticity*) untuk Menguji Pasar Modal Efisien di Indonesia (Studi pada Harga Penutupan (Closing Price) Indeks Saham LQ 45 Periode 2009-2011). *Jurnal Administrasi Bisnis* 7 (2).9.
- Fath, N., E.K. Laksana., R. Sirait. 2018. Implementasi *Bat Algorithm* dalam Optimasi Penempatan *Femtocell*. *Jurnal Rekayasa Elektrika* 14 (3), 272971.
- Jordan, G., dkk. 2011. Analisis Perbandingan Metode Peramalan Indeks Harga Saham

- Gabungan (IHSG) dengan Metode OLS-ARCH/GARCH dan ARIMA. Salatiga.
- Handika, I.P.S., I.A. Giriantari, A. Dharma. 2016. Perbandingan Metode *Extreme Learning Machine* dan *Particle Swarm Optimization Extreme Learning Machine* untuk Peramalan Jumlah Penjualan Barang. vol 15, 84-90.
- Hansun, S. 2012. Peramalan Data IHSG Menggunakan *Fuzzy Time Series*. IJJS (*Indonesian Journal of Computing and Cybernetics Systems*) 6 (2).
- Huang, G.B., Q.Y. Zhu., C.K. Siew. 2004. *A fast constructive learning algorithm for single-hidden-layer neural networks. ICARCV 2004 8th Control, Automation, Robotics and Vision Conference*. 3, 1907-1911.
- Huang, G.B., Q.Y. Zhu., C.K. Siew. 2004. *Extreme learning machine: a new learning scheme of feedforward neural networks. IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)* 2, 985-990.
- Jakarta Stock Exchange Composite Historical Rates (JKSE). <http://www.investing.com>. 5 Januari 2020 (18.10).
- Jannah, R. 2020. Implementasi Metode *Extreme Learning Machine* (ELM) dalam Klasifikasi Tipe Gangguan *Skizofrenia*. *Skripsi*. Universitas Islam Negeri Sunan Ampel. Surabaya.
- Jogiyanto. 2013. Teori Portofolio dan Analisis Investasi. Edisi Ketujuh. BPF. Yogyakarta.
- Kotteeswaran, R., Sivaku, L. 2013. *A el Bat algorithm based re-tuning of PI controller of coal gasifier for optimum response. Mining Intelligence and Knowledge Exploration*, 506-517.
- Kusumadewi, S. 2003. *Artificial Intelligence* (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta.
- Maulana, R. 2015. Penerapan *Bat Algorithm* (BA) untuk menyelesaikan Penjadwalan *Flowshop*". *Skripsi*. Departemen Matematika. Fakultas Sains dan Teknologi. Universitas Airlangga. Surabaya.
- Novyan, T. 2015. Pencarian Solusi pada Permasalahan Sistem Persamaan Nonlinier Menggunakan Metode *Bat Algorithm*. *Skripsi*. Program Studi Matematika. Fakultas Sains Dan Teknologi. Universitas Islam Negeri Syarif Hidayatullah. Jakarta.
- Pratama, M. I., P.P. Adikara., S. Adinugroho. 2018. Peramalan Harga Saham Menggunakan Metode *Extreme Learning Machine* (ELM) Studi Kasus Saham Bank Mandiri. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN 2548, 964X.
- Puspitaningrum, D. 2006. Pengantar Jaringan Saraf Tiruan. Andi. Yogyakarta.
- Putri, O. F. 2019. Prediksi Harga Emas di Indonesia Menggunakan *Extreme Learning Machine* (ELM). *Skripsi*. Program Studi Statistika. Universitas Muhammadiyah Seang, Seang.
- Ramadhan, R., Mahmudy, W.F. 2019. Optimasi Bobot Awal *Extreme Learning Machine* menggunakan Algoritme Genetika untuk Klasifikasi Penanganan *Human Papilloma Virus*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN 2548, 964X.
- Sari, V. 2017. Aplikasi *Extreme Learning Machine* untuk Peramalan Data *Tim Series* (Studi Kasus: Saham Bank Bri). *Akademi Statistika Muhammadiyah Seang*.
- She, X. Yang. 2010. *A new metaheuristic bat-inspired algorithm n Nature inspired cooperative strategies for optimization* (NISCO 2010). Springer, pp. 65-74.
- Sholekhah, M. 2017. *Hybrid Jaringan Syaraf Tiruan Metode Extreme Learning Machine* (ELM) dengan *Genetic Algorithm* (GA). *Skripsi*. Universitas Airlangga. Surabaya.
- Siang, J. 2009. Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab. ANDI. Yogyakarta.
- Sun, Z.L., dkk. 2018. *Sales Forecasting using Extreme Learning Machine with Application in Fashion Retailing. Elsevier ision Support System*, 46 411-419.
- Taha, A.M., dkk. 2013. *Bat Algorithm for Rough SetAttribute Reduction. Journal*

*of Theoretical and Applied Information Technology*, Vol 51, No 1.

- Ubay, M.S. 2012. Peramalan Harga Saham dengan Menggunakan Jaringan Syaraf Tiruan Metode *Extreme Learning Machine*. *Skripsi*. Fakultas Sains dan Teknologi. Universitas Airlangga. Surabaya.
- Wang, Y., dkk. 2019. *A el Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization*. *Mathematics* 7 (2), 135.
- Zulinda. 2020. Penerapan Jaringan Syaraf Tiruan *Extreme Learning Machine* dan *Backpropagation* untuk Memprediksi Harga Saham Pt Bank Mandiri (Persero) Tbk (Studi Kasus: Harga Penutupan Saham Harian PT. Bank Mandiri (Persero) Tbk Dari Januari 2019 sampai dengan Januari 2020). *Skripsi*. Program Studi Statistika. Universitas Islam Indonesia. Yogyakarta.

