



## METODE *GENETIC ALGORITHM - LONG SHORT-TERM MEMORY* PADA PERAMALAN HARGA SAHAM

Lathifatul Azizah<sup>✉</sup>, YL Sukestiyarno

Jurusan Matematika, FMIPA, Universitas Negeri Semarang  
Gedung D7 Lt. 1, Kampus Sekaran GunungPati, Semarang, 50229

### Info Artikel

Sejarah Artikel:  
Diterima Oktober 2022  
Disetujui November 2022  
Dipublikasikan November 2022

#### Keywords:

*Time Series, Deep Learning,  
Genetic Algoritkm – Long Short-  
Term Memory*

### Abstrak

Pelatihan dan keberhasilan algoritma *deep learning* sangat dipengaruhi terhadap pemilihan *hyperparameter* yang tepat. Dalam penelitian ini akan dilakukan *hybrid* antara *Genetic Algoriithm* dan *Long Short-Term Memory* untuk mencari model yang cocok dalam memprediksi harga saham Bank Rakyat Indonesia Tbk. *Hybrid LSTM* yang diintegrasikan dengan GA untuk menemukan *window size*, *epoch*, dan jumlah unit LSTM. Pemilihan algoritma untuk pengoptimalan menggunakan *optimizers* untuk mendapatkan model terbaik sehingga dapat ditemukan *hyperparameter* terbaik untuk peramalan data *time series*. Dari metode *Genetic Algorithm – Long Short-Term Memory* yang telah diterapkan menghasilkan, bahwa metode tersebut memiliki tingkat akurasi yang baik dengan nilai MAPE di bawah 10% di setiap *optimizer* yang digunakan. Tingkat kesalahan yang dihasilkan cukup rendah dengan nilai RMSE 93,03 sampai dengan 94,40 saat *training* dan *testing*. Kemudian *hyperparameter* terpilih yang dapat digunakan yaitu *epoch* sebesar 24, dengan *neurons* [4, 5, 2] dan *window size* 2, serta *optimizer* Adam.

### Abstract

*The training and success of deep learning algorithms is strongly influenced by the selection of the right hyperparameters. In this research, a hybrid between Genetic Algorithm and Long Short-Term Memory will be carried out to find a suitable model in predicting the stock price of Bank Rakyat Indonesia Tbk. Hybrid LSTM integrated with GA to find window size, epoch, and number of LSTM units. Algorithm selection for optimization uses optimizers to get the best model so that the best hyperparameters can be found for forecasting time series data. From the Genetic Algorithm – Long Short-Term Memory method that has been applied, it shows that this method has a good level of accuracy with MAPE values below 10% in each optimizer used. The resulting error rate is quite low with an RMSE value of 93.03 to 94.40 during training and testing. Then the selected hyperparameter that can be used is epoch of 24, neurons [4, 5, 2], window size 2, and optimizer Adam.*

### How to cite:

Azizah, L. & Sukestiyarno, YL. 2022. Metode *Genetic Algorithm – Long Short-Term Memory Pada Peramalan Harga Saham*. *UNNES Journal of Mathematics*. 11(2):153-160.

## PENDAHULUAN

Kemajuan pesat teknologi komputer dapat membantu dalam mempermudah serta mempercepat proses peramalan data. Dalam era *Artificial Intelligence*, data *time series* hanyalah satu bidang di mana *machine learning* telah maju secara signifikan. *Deep Learning* merupakan metode pembelajaran mesin yang menyerupai bagaimana cara kerja sistem dasar otak manusia yang disebut *neural networks*, sehingga dapat dikatakan bahwa *deep learning* menggunakan *Artificial Neural Networks*. Yotenka et al., (2020) menjelaskan, “*Deep learning* merupakan salah satu aplikasi *machine learning* yang memanfaatkan ANN dengan banyak lapisan.”. Salah satu metode *deep learning* yang dapat bekerja untuk data deret waktu atau *time series* adalah *Recurrent Neural Network* (RNN) yang mana metode ini dirancang untuk bekerja dengan data sekuensial.

Kemajuan RNN berkembang cukup pesat dalam berbagai bidang, namun RNN memiliki kelemahan dalam mengolah data *time series* karena nilai ketergantungan pendek. RNN yang masih mempunyai kekurangan dalam ketidakmampuannya untuk belajar menghubungkan informasi sebelumnya dan informasi sesudahnya, dan karena LSTM menggunakan *memory cells* dan *gates units* untuk mengelola memori di setiap input, LSTM mampu mengatasi keterbatasan ini. (Arfan & Lussiana ETP, 2019). Penelitian yang dilakukan oleh Arfan & Lussiana ETP (2019) tentang penggunaan data sekunder berjumlah dua yakni data indeks saham dan nilai tukar USD terhadap IDR untuk membuat prakiraan harga saham di Indonesia dengan menggunakan data dari tanggal 09 Juni 2019 sampai 06 Juni 2019 dengan metode LSTM yang menghasilkan pengujian bawah LSTM dapat memprediksikan harga saham dari tahun 2017 hingga tahun 2019 dengan baik, ditunjukkan melalui hasil nilai *error* yang kecil, sehingga dapat ditarik kesimpulan dengan hasil yang akurat, LSTM dapat memperkirakan harga saham dan dapat mengatasi ketergantungan jangka panjang. Adapun penelitian yang telah dilakukan oleh Yotenka et al., (2020) tentang implementasi LSTM pada harga saham tiga perusahaan perkebunan di Indonesia menghasilkan LSTM terbaik untuk saham SSMS adalah 70, yang menghasilkan RMSE 21.328 menggunakan *hidden neuron* dan opsi pengoptimal RMSProp. Kemudian, model LSTM terbaik adalah saham LSIP, yang

menghasilkan skor RMSE 33,097 dengan Adam dan *hidden neuron* diatur ke maksimum 80 di pengoptimal. Model terbaik adalah saham SIMP, yang bila digunakan dengan pengaturan pengoptimal Adamax dan 100 *hidden neuron*, menghasilkan skor RMSE sebesar 8,337.

Penelitian yang dilakukan oleh Sugiyarto & Faddillah (2017) tentang optimasi *artificial neural network* dengan *algorithm genetic* yang digunakan untuk prediksi *approval credit card* dengan menerapkan algoritma *neural network* memperoleh peningkatan hasil yang awalnya 85,42% meningkat menjadi 87,82%. Kemudian terdapat penelitian yang dilakukan oleh Chung & Shin (2018) dalam memprediksi harga saham KOSPI dengan menerapkan penggunaan *Genetic Algorithm - Long Short Term Memory* menghasilkan model LSTM terbaik adalah dengan pengaturan *window size* 10 dan menyusun 2 lapisan tersembunyi dengan *node* masing-masing 15 dan 7, didapatkan MSE dan MAE masing-masing sebesar 181,99% dan 10,21%.

Penggerak utama ekonomi nasional adalah konsumsi, ekspor impor, serta investasi, dan tidak dapat dipungkiri bahwa perbankan memiliki peran besar dalam sektor tersebut dan juga termasuk penggerak utama Produk Domestik (PDB) Indonesia. Salah satu saham perbankan yang terdapat di pasar modal Indonesia adalah saham PT. Bank Rakyat Indonesia (Persero) Tbk. PT ini telah tercatat di dalam pasar saham modal sejak 10 November 2003 dengan bidang usaha utama Perbankan dan berada pada sektor Keuangan dan subsektor Bank. PT ini mempunyai 30 anak perusahaan jenis *remittance*, perbankan, *multi finance* dan asuransi dengan laba yang mencapai Rp 32, 4 Triliun pada tahun 2018 (PT Bursa Efek Indonesia, n.d.).

Pelatihan dan keberhasilan algoritma *deep learning* sangat dipengaruhi terhadap pemilihan *hyperparameter* yang tepat. *Hyperparameter* adalah variabel-variabel yang nilainya diberikan secara manual ke model untuk membantu dalam proses pembelajaran (Peter & Matskevichus, 2019). Contoh *hyperparameter* dari ANN, misalnya jumlah neuron dan *hidden layer*, *learning rate*, *activation function*. Penelitian yang dilakukan oleh Chung & Shin (2018) berhasil menemukan *hyperparameter* melalui implementasi *Genetic Algorithm - Long Short Term Memory* dalam menemukan model terbaik. Dalam penelitian

ini akan dilakukan *hybrid* antara *Genetic Algorithm* dan *Long Short-Term Memory* untuk mencari model yang cocok dalam memprediksi harga saham Bank Rakyat Indonesia Tbk. *Genetic Algorithm* digunakan pada optimalisasi arsitektur seperti jumlah *epoch*, *window size*, dan jumlah unit LSTM di *hidden layer*. Pemilihan algoritma optimasi yang tepat akan menghasilkan nilai terbaik dari setiap parameter model, oleh karena itu akan dilakukan pula pengoptimalan penyetalan menggunakan beberapa optimizer untuk mendapatkan model terbaik, sehingga dapat ditemukan parameter terbaik untuk peramalan data *time series*.

Tujuan penelitian ini adalah (1) untuk mengidentifikasi bagaimana penerapan metode *Genetic Algorithm - Long Short-Term Memory* dalam memprediksi harga saham Bank Rakyat Indonesia Tbk dan (2) untuk mengidentifikasi bagaimana kinerja metode *Genetic Algorithm - Long Short-Term Memory* dalam peramalan harga saham Bank Rakyat Indonesia Tbk.

## METODE

Dalam penelitian ini menggunakan metode *Genetic Algorithm - Long Short-Term Memory* (GA-LSTM) yang digunakan untuk *forecasting* harga saham PT Bank Rakyat Indonesia (Persero) Tbk. Data yang dikumpulkan dan digunakan dalam penelitian ini yaitu dataset harga saham Bank BRI Tbk yang diambil pada 3 Februari 2022 dalam bentuk format CSV (*Comma Separated Values*) yang diperoleh dari website [www.finance.yahoo.com](http://www.finance.yahoo.com). Adapun data yang digunakan yaitu data *close price* sebanyak 2.487 data harga yang diubah dalam format yang dapat digunakan *python* dengan menggunakan modul *Pandas*.

Dilakukan pengecekan apakah ada data yang hilang atau tidak lengkap, agar data yang hilang tersebut tidak mempengaruhi pengolahan data secara keseluruhan. Kemudian perlu dilakukan transformasi data dan normalisasi data sebelum digunakan, dilakukan transformasi data untuk membuat data bersifat stasioner dimana data yang akan digunakan tidak memiliki kecenderungan terhadap suatu trend tertentu. Untuk normalisasi dalam penelitian ini, menggunakan implementasi dari objek transformasi dari perpustakaan *scikit-learn* kelas *MinMaxScaler*. Selanjutnya dilakukan pembagian data menjadi

data *training* 80% dan 20% data sisanya untuk data *testing*. Pada penelitian ini menggunakan rancangan LSTM dengan 3 lapisan *neural* beserta modul *Dropout* setiap lapisannya, menggunakan *loss function mean squared error* serta menggunakan beberapa optimizer dengan berbantuan *library tensorflow*.

LSTM memiliki empat komponen yaitu *input gate*, *forget gate*, *cell state*, dan *output gate* (Goodfellow et al., 2016). Pada *input gate* terdapat dua fungsi untuk menerima informasi baru:  $r_t$  dan  $d_t$ .  $r_t$  merangkai sebelumnya vektor tersembunyi  $h_{t-1}$  dengan informasi yang baru  $x_t$ . Artinya,  $[h_{t-1}, x_t]$ , lalu dikalikan dengan matriks bobot  $W_r$ , setelahnya ditambah vektor kebisingan  $b_r$ .  $d_t$  melakukan hal yang sama. Selanjutnya mengalikan  $r_t$  dan  $d_t$  dengan elemen-bijaksana sehingga didapatkan *cell state*  $c_t$ .

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (1)$$

$$d_t = \tanh(W_d \cdot [h_{t-1}, x_t] + b_d) \quad (2)$$

*Forget gate*  $f_t$  terlihat begitu mirip dengan  $r_t$  pada *input gate*. Gerbang ini mengontrol batas hingga nilainya disimpan dalam memori.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

*Cell state* digunakan untuk perhitungan perkalian didasarkan pada elemen antara *cell state* sebelumnya  $C_{t-1}$  dan *forget gate*  $f_t$ , lalu ditambahkan hasil *input gate*  $r_t$  dan  $d_t$ .

$$C_t = f_t \cdot C_{t-1} + r_t \cdot d_t \quad (4)$$

Dari sini  $o_t$  merupakan *output gate* pada langkah waktu  $t$ , dan  $W_o$  dan  $b_o$  merupakan bobot baris dan bias untuk *output gate*.  $H_t$  merupakan lapisan tersembunyi akan menuju pada langkah berikutnya, atau naik ke *output* sebagai  $y_t$  yang diperoleh dari menerapkan *tanh* ke  $h_t$ . Perhatikan bahwa *output*  $o_t$  bukan *output* dari  $y_t$ , melainkan *gate* yang digunakan untuk mengontrol *output*.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \tanh C_t \quad (6)$$

Dalam bukunya, El-Amir & Hamdy (2020) berpendapat bahwa *optimizer* yang paling banyak digunakan dalam *deep learning* adalah tipe *mini-batch* yang terdiri dari Adagrad, Adadelta, RMSprop, Nadam, dan Adam.

Pada penerapan Adagrad, *learning rate* dinormalisasi untuk setiap dimensi di mana fungsi biaya bergantung. *Learning rate* pada setiap iterasi merupakan *learning rate* global dibagi dengan norm l2 dari *gradient* sebelumnya sampai iterasi saat ini untuk setiap dimensi. Formula yang digunakan pada Adagrad sebagai berikut:

$$S_t = S_{t-1} + \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (7)$$

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (8)$$

Keterangan :

- $t$  = untuk langkah waktu
- $w$  = bobot/ parameter yang akan diperbarui
- $\alpha$  = *learning rate* (0.001)
- $\frac{\partial L}{\partial w}$  = *gradient* L (*Loss function*)
- $S$  = jumlah kumulatif kuadrat dari *gradient* saat ini dan sebelumnya

Adadelta merupakan ekstensi Adagrad sebagai alternatif untuk mengurangi keagresifan Adagrad, mengurangi *learning rate* secara monoton, juga lebih fokus pada komponen *learning rate*. Formula yang digunakan adalah sebagai berikut:

$$D_t = \beta D_{t-1} + (1 - \beta) [\Delta w_t]^2 \quad (9)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (10)$$

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} + \epsilon}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (11)$$

Keterangan :

- $D$  = perbedaan antara bobot saat ini dan bobot yang diperbarui
- $\beta$  = 0,9 ;  $\epsilon$  = 1e-7

RMSprop merupakan *learning rate* adaptif lain yang merupakan peningkatan Adagrad. Formula yang digunakan sebagai berikut:

$$S_t = \beta S_{t-1} + (1 - \beta) \left[ \frac{\partial L}{\partial w_t} \right]^2 \quad (12)$$

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (13)$$

RMSprop dan AdaGrad dikombinasikan dengan momentum sehingga terbentuklah Adam. Adam mengukur tingkat pembelajaran melalui gradien kuadrat, mirip dengan RMSprop, dan menggunakan rata-rata gradien dinamis untuk memanfaatkan momentum. (Goodfellow et al., 2016). Formula yang digunakan sebagai berikut:

$$\hat{V}_t = \frac{V_t}{1 - \beta_1^t} \quad (14)$$

$$\hat{S}_t = \frac{S_t}{1 - \beta_2^t} \quad (15)$$

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{S}_t + \epsilon}} \cdot \hat{V}_t \quad (16)$$

Keterangan :

- $\hat{V}$  = rata-rata penurunan *gradient* dengan momentum menggantikan *gradient* saat ini
- $\hat{S}$  = rata-rata jumlah kumulatif kuadrat dari *gradient* saat ini dan sebelumnya
- $\beta_1$  = 0,9
- $\beta_2$  = 0,999
- $\epsilon$  = 1e-7

Nadam digunakan untuk *noisy gradient* atau gradien dengan kelungkan tinggi. Proses pembelajaran dipercepat dengan menjumlahkan *decay* eksponensial dari rata-rata bergerak untuk gradien sebelumnya dan saat ini, Nadam mengambil *gradient* satu langkah lebih jauh dengan menggunakan Nesterov untuk menggantikan  $\hat{V}$  sebelumnya pada persamaan sebelumnya dengan  $\hat{V}$  saat ini. Formula yang digunakan sebagai berikut:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{S}_t + \epsilon}} \left( \beta_1 \hat{V}_t + \frac{1 - \beta_1}{1 - \beta_1^t} \cdot \frac{\partial L}{\partial w_t} \right) \tag{17}$$

dimana,

$$\hat{V}_t = \frac{V_t}{1 - \beta_1^t} ; \hat{S}_t = \frac{S_t}{1 - \beta_2^t} \tag{18}$$

dimana,

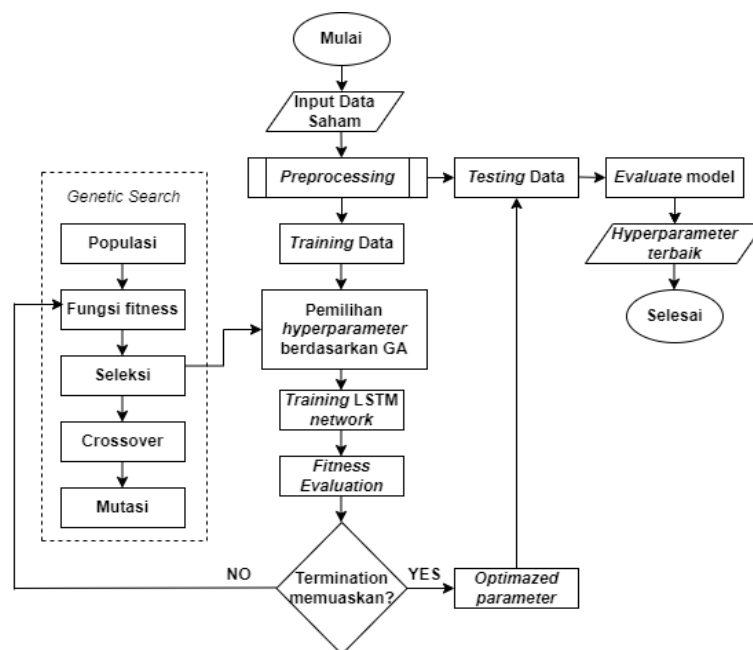
$$V_t = \beta_1 V_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t} ; S_t = \beta_2 S_{t-1} + (1 - \beta_2) \left[ \frac{\partial L}{\partial w_t} \right]^2 \tag{19}$$

Keterangan :

- $\alpha$  = learning rate (0,002)
- $\beta_1$  = 0,9
- $\beta_2$  = 0,999
- $\epsilon$  = 1e-7

Sedangkan untuk rancangan inialisasi parameter *Genetic-Algorithm* menggunakan *library DEAP* dari *python*, dilakukan tahap awal dengan menentukan populasi awal yang mana merupakan kumpulan kromosom-kromosom yang berisikan solusi untuk jumlah *window size*, *epoch*, dan jumlah unit. Pembentukan kromosom dilakukan secara biner menggunakan bilangan

biner. Struktur dasar algoritma genetika terdiri dari beberapa langkah (Haupt & Haupt, 2004) yaitu: 1) Inialisasi populasi; 2) Evaluasi populasi; 3) Seleksi populasi yang akan dikenai operator genetik; 4) Proses *crossover* pasangan kromosom tertentu; 5) Proses mutasi kromosom tertentu; 6) Evaluasi populasi baru; 7) Mengulangi dari Langkah 3 jika syarat berhenti belum terpenuhi. Dalam penelitian ini menggunakan: populasi = 5, maksimum generasi = 10, *crossover rate* = 0,4 dan *mutation rate* = 0,1. Setiap rancangan dilakukan untuk mengevaluasi kesesuaian GA. Proses GA diulangi lebih dari sekali dengan menetapkan nilai yang berbeda untuk jumlah *window size*, *epoch*, dan jumlah unit.



Gambar 1. Flowchart model GA-LSTM

**HASIL DAN PEMBAHASAN**

Penelitian menggunakan metode GA-LSTM sebagai proses perhitungan dengan menerapkan beberapa optimasi yang berbeda untuk menemukan *hyperparameter* jumlah *epoch*, *window size*, dan jumlah unit LSTM di *hidden layer*. Pada penelitian ini terbatas menggunakan harga *close* untuk jenis harga prediksi. Hasil dari penelitian ini yaitu model dengan *hyperparameter* terbaik yang didapatkan dari *training* dan *testing* data dengan nilai MAE dan MAPE terendah, lalu model yang didapatkan dapat digunakan untuk memprediksi harga saham Bank Rakyat Indonesia TBK.

*Root Mean Square Error* (RMSE) digunakan untuk mengukur perbedaan antara target yang diperkirakan dan target actual dengan cara menghitung nilai akar kuadrat dari MSE. Semakin tinggi nilai yang dihasilkan RMSE, maka tingkat keakuratan semakin rendah, begitupun sebaliknya, jika nilai RMSE yang dihasilkan semakin rendah, maka tingkat keakuratan semakin tinggi (Sabar Sautomo & Hilman Ferdinandus Pardede, 2021). Adapun rumus RMSE ditunjukkan pada persamaan berikut.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (20)$$

Keterangan :

- $y_i$  = nilai observasi ke -  $i$
- $\hat{y}_i$  = nilai hasil *forecast*
- $n$  = banyaknya data

*Mean Absolute Percentage Error* (MAPE) untuk mengukur kesalahan dengan menghitung cara rata-rata kesalahan absolut yang dibagi dengan nilai sebenarnya, yang mana hasilnya menunjukkan nilai persentase kesalahan absolut dari hasil model prediksi. Model prediksi semakin baik jika nilai MAPE semakin rendah (Hadi & Abidin, 2019) Rumus RMSE ditunjukkan pada persamaan berikut.

$$MAPE = \frac{100\%}{n} \sum \left| \frac{y - \hat{y}}{y} \right| \quad (21)$$

Keterangan :

- $\hat{y}$  = nilai hasil *forecast*
- $y$  = nilai observasi ke -  $i$
- $n$  = banyaknya data

Tabel 1 Range MAPE value

| Range MAPE | Arti                        |
|------------|-----------------------------|
| <10%       | Tingkat akurasi sangat baik |
| 10 – 20%   | Tingkat akurasi baik        |
| 20 – 50%   | Tingkat akurasi layak       |
| >50%       | Tingkat akurasi buruk       |

Setelah melakukan seluruh tahapan penelitian, didapatkan output berupa nilai RMSE dan MAPE, serta grafik perbandingan harga asli dengan data hasil prediksi, hasil ditunjukkan sebagai berikut.

Pada model yang menggunakan optimasi Adagrad dengan *hyperparameter* jumlah *epoch* sebanyak 33, *window size* sebanyak 1, dan jumlah unit LSTM di *hidden layer* yaitu [4, 4, 3], nilai RMSE dan MAPE pada saat *training* yaitu 100,58 dan 1,49%, sedangkan pada *testing* yaitu 94,43 dan 1,86%. Kemudian model dengan *hyperparameter* jumlah *epoch* sebanyak 38, *window size* sebanyak 1, dan jumlah unit LSTM di *hidden layer* yaitu [4, 4, 3], nilai RMSE dan MAPE pada saat *training* yaitu 108,30 dan 1,53%, sedangkan pada *testing* yaitu 95,39 dan 1,88%.

Tabel 2 Hasil peramalan dengan optimasi Adagrad

| Epoc h | Neuron s  | Windo w Size | Data             | RMS E   | MAPE(%) |
|--------|-----------|--------------|------------------|---------|---------|
| 33     | [4, 4, 3] | 1            | <i>Trainin g</i> | 100, 58 | 1, 49   |
|        |           |              | <i>Testin g</i>  | 94, 43  | 1, 86   |
|        |           |              | <i>Trainin g</i> | 108, 31 | 1, 53   |
| 38     | [4, 4, 3] | 1            | <i>Testin g</i>  | 95, 39  | 1, 88   |

Pada model yang menggunakan optimasi Adadelta dengan *hyperparameter* jumlah *epoch* sebanyak 35, *window size* sebanyak 1, dan jumlah unit LSTM di *hidden layer* yaitu [1, 3, 2], nilai RMSE dan MAPE pada saat *training* yaitu 106,43 dan 1,51%, sedangkan pada *testing* yaitu 94,47 dan 1,86%. Kemudian model dengan *hyperparameter* jumlah *epoch* sebanyak 18, *window size* sebanyak 2, dan jumlah unit LSTM di *hidden layer* yaitu [2, 1, 4], nilai RMSE dan MAPE pada saat *training* yaitu 99, 69 dan 1,58%, sedangkan pada *testing* yaitu 98,09 dan 1,91%.

Tabel 3 Hasil peramalan dengan optimasi Adadelata

| Epoc h | Neuron s  | Windo w Size | Data           | RMS E  | MAPE(%) |
|--------|-----------|--------------|----------------|--------|---------|
| 35     | [1, 3, 2] | 1            | <i>Trainin</i> | 106,   | 1, 51   |
|        |           |              | <i>g</i>       | 43     |         |
|        |           |              | <i>Testing</i> | 94, 47 | 1, 86   |
| 18     | [2, 1, 4] | 2            | <i>Trainin</i> | 99, 69 | 1, 58   |
|        |           |              | <i>g</i>       |        |         |
|        |           |              | <i>Testing</i> | 98, 09 | 1, 91   |

Pada model yang menggunakan optimasi RMSprop dengan *hyperparameter* jumlah *epoch* sebanyak 41, *window size* sebanyak 1, dan jumlah unit LSTM di *hidden layer* yaitu [4, 5, 5], nilai RMSE dan MAPE pada saat *training* yaitu 102,91 dan 1,52%, sedangkan pada *testing* yaitu 94,40 dan 1,88%. Kemudian model dengan *hyperparameter* jumlah *epoch* sebanyak 43, *window size* sebanyak 5, dan jumlah unit LSTM di *hidden layer* yaitu [3, 1, 4], nilai RMSE dan MAPE pada saat *training* yaitu 102,83 dan 1,53%, sedangkan pada *testing* yaitu 95,49 dan 1,91%.

Tabel 4 Hasil peramalan dengan optimasi RMSProp

| Epoc h | Neuron s  | Windo w Size | Data           | RMS E  | MAPE(%) |
|--------|-----------|--------------|----------------|--------|---------|
| 41     | [4, 5, 5] | 1            | <i>Trainin</i> | 102,   | 1, 52   |
|        |           |              | <i>g</i>       | 91     |         |
|        |           |              | <i>Testing</i> | 94, 40 | 1, 88   |
| 43     | [3, 1, 4] | 5            | <i>Trainin</i> | 102,   | 1, 53   |
|        |           |              | <i>g</i>       | 83     |         |
|        |           |              | <i>Testing</i> | 95, 49 | 1, 91   |

Pada model yang menggunakan optimasi RMSprop dengan *hyperparameter* jumlah *epoch* sebanyak 41, *window size* sebanyak 1, dan jumlah unit LSTM di *hidden layer* yaitu [4, 5, 5], nilai RMSE dan MAPE pada saat *training* yaitu 102,91 dan 1,52%, sedangkan pada *testing* yaitu 94,40 dan 1,88%. Kemudian model dengan *hyperparameter* jumlah *epoch* sebanyak 43, *window size* sebanyak 5, dan jumlah unit LSTM di *hidden layer* yaitu [3, 1, 4], nilai RMSE dan MAPE pada saat *training* yaitu 102,83 dan 1,53%, sedangkan pada *testing* yaitu 95,49 dan 1,91%.

Pada model yang menggunakan optimasi Adam dengan *hyperparameter* jumlah *epoch* sebanyak 38, *window size* sebanyak 4, dan jumlah unit LSTM di *hidden layer* yaitu [4, 5, 2], nilai RMSE dan MAPE pada saat *training* yaitu 94,28 dan 1,51%, sedangkan pada *testing*

yaitu 95,49 dan 1,87%. Kemudian model dengan *hyperparameter* jumlah *epoch* sebanyak 24, *window size* sebanyak 4, dan jumlah unit LSTM di *hidden layer* yaitu [4, 5, 2], nilai RMSE dan MAPE pada saat *training* yaitu 95,93 dan 1,51%, sedangkan pada *testing* yaitu 95,06 dan 1,87%.

Tabel 5 Hasil peramalan dengan optimasi Adam

| Epoc h | Neuron s  | Windo w Size | Data           | RMS E | MAPE(%) |
|--------|-----------|--------------|----------------|-------|---------|
| 38     | [4, 5, 2] | 4            | <i>Trainin</i> | 94,28 | 1, 51   |
|        |           |              | <i>g</i>       |       |         |
|        |           |              | <i>Testing</i> | 95,49 | 1, 87   |
| 24     | [4, 5, 2] | 2            | <i>Trainin</i> | 95,93 | 1, 51   |
|        |           |              | <i>g</i>       |       |         |
|        |           |              | <i>Testing</i> | 95,06 | 1, 87   |

Tabel 6 Hasil peramalan dengan optimasi Nadam

| Epoc h | Neuron s  | Windo w Size | Data           | RMS E  | MAPE(%) |
|--------|-----------|--------------|----------------|--------|---------|
| 15     | [3, 3, 2] | 6            | <i>Trainin</i> | 93, 03 | 1, 51   |
|        |           |              | <i>g</i>       |        |         |
|        |           |              | <i>Testing</i> | 95, 62 | 1, 87   |
| 28     | [4, 5, 4] | 2            | <i>Trainin</i> | 95, 99 | 1, 51   |
|        |           |              | <i>g</i>       |        |         |
|        |           |              | <i>Testing</i> | 95, 17 | 1, 87   |

Dari hasil *training* dan *testing* masing-masing optimasi, dapat dilihat bahwa nilai RMSE yang dihasilkan menunjukkan nilai yang kecil yang berarti bawah model yang dihasilkan dari prediksi ini memiliki tingkat kesalahan yang kecil. Dan hasil dari nilai MAPE memiliki nilai di bawah 10% yang menunjukkan model prediksi memiliki tingkat akurasi yang sangat baik. Dengan hasil menggunakan beberapa optimasi dapat menghasilkan nilai RMSE dan MAPE yang kecil, maka pada penelitian ini dapat dikatakan bahwa GA-LSTM dapat meningkatkan kinerja dan menghemat waktu karenanya dalam sekali proses kita dapat menemukan *hyperparameter* yang baik untuk digunakan. Meskipun begitu, jika dilihat pada tabel 2 hingga tabel 6, dapat diketahui bahwa nilai MAPE yang dihasilkan oleh *optimizers* Adam dan Nadam memiliki nilai yang sama dan stabil. Sedangkan RMSE *testing* pada model kedua Adam memiliki nilai terendah yaitu 95,06 dibandingkan dengan model Adam pertama maupun kedua model yang dihasilkan optimizer Nadam. Sehingga terlihat bahwa *hyperparameter* terpilih yang

dapat digunakan yaitu *epoch* sebesar 24, dengan *neurons* [4, 5, 2] dan *window size* 2, serta *optimizer* Adam.

## PENUTUP

Penerapan metode *Genetic Algorithm – Long Short-Term Memory* menunjukkan hasil yang sangat baik, dibuktikan nilai RMSE dan MAPE yang dihasilkan pada proses *training* dan *testing* data yang menunjukkan tingkat kesalahan yang cukup rendah dan juga tingkat akurasi yang cukup baik. Dengan nilai MAPE di bawah 10% di setiap *optimizer* yang digunakan. Tingkat kesalahan yang dihasilkan cukup rendah dengan nilai RMSE 93,03 sampai dengan 94,40 saat *training* dan *testing*. Kemudian *hyperparameter* terpilih yang dapat digunakan yaitu *epoch* sebesar 24, dengan *neurons* [4, 5, 2] dan *window size* 2, serta *optimizer* Adam. Penelitian ini dapat digunakan pada karakteristik data yang sama, dengan metode *Genetic Algorithm – Long Short-Term Memory* untuk mencari *hyperparameter*. Penelitian selanjutnya dapat menerapkan *hybrid Genetic Algorithm* dengan metode *deep learning* lainnya seperti *Gated Recurrent Unit* (GRU).

## UCAPAN TERIMAKASIH

Terima kasih kepada Universitas Negeri Semarang, FMIPA, serta Jurusan Matematika yang telah memberikan kesempatan bagi kami untuk melakukan penelitian.

## DAFTAR PUSTAKA

Arfan, A., & Lussiana ETP. 2019. *Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory*. Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK). Jakarta 22 Agustus 2019.

Chung, H., & Shin, K. S. 2018. Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10): 1-18.

El-Amir, H., & Hamdy, M. 2020. *Deep Learning Pipeline: Building a Deep Learning Model with TensorFlow*. New York : Apress Media LLC.

Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. Cambridge: MIT

Press.

Hadi, M. I., & Abidin, Z. 2019. Prediksi Harga Cryptocurrency menggunakan Metode Long Short Term Memory dengan Optimasi Adaptive Moment Estimation. *Scientific Journal of Informatics*, 6(1): 1–11.

Haupt, R. L., & Haupt, S. E. 2004. *Practical Genetic Algorithms*. Hoboken: John Wiley & Sons, Inc.

Peter, G., & Matskevichus, M. 2019. *Hyperparameters Tuning for Machine Learning Models for Time Series Forecasting*. 2019 6th International Conference on Social Networks Analysis, Management and Security (SNAMS). Granada 22-25 Oktober 2019.

PT Bursa Efek Indonesia. (n.d.). Diakses di <https://www.idx.co.id/perusahaan-tercatat/profil-perusahaan-tercatat/detail-profile-perusahaan-tercatat/?kodeEmiten=BBRI> tanggal 3 Februari 2022.

Sabar Sautomo, & Hilman Ferdinandus Pardede. 2021. Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(1): 99–106.

Sugiyarto, I., & Faddillah, U. 2017. Optimasi Artificial Neural Network dengan Algorithm Genetic pada Prediksi Approval Credit Card. *Jurnal Teknik Informatika STMIK Antar Bangsa*, 3(2): 151–156.

Yotenka, R., Fikri, F., & Huda, E. 2020. Implementasi Long Short-Term Memory Pada Harga Saham Perusahaan Perkebunan Di Indonesia. *Jurnal UJMC*, 6(1): 9–18.