

## PERAMALAN NILAI TUKAR RUPIAH TERHADAP DOLAR AMERIKA DENGAN METODE *LONG SHORT-TERM MEMORY* DAN *GATED RECURRENT UNIT*

Beatrice Marietta<sup>1</sup>, Rukun Santoso<sup>2\*</sup>, Ardiana Alifatus Sa'adah<sup>3</sup>

<sup>1,2,3</sup>Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

\*e-mail: [rukunsantoso25@gmail.com](mailto:rukunsantoso25@gmail.com)

DOI: 10.14710/j.gauss.14.1.13-22

### Article Info:

Received: 2024-06-21

Accepted: 2025-02-10

Available Online: 2025-02-12

### Keywords:

*Economic Recession; Forecasting; Rupiah Exchange Rate; Long-Short Term Memory; Gated Recurrent Unit*

**Abstract:** Economic recession is a significant and sustained decline in economic activity, that can be observed through the weakening of the exchange rate. Information about future exchange rate changes can be obtained through forecasting. This research was conducted to obtain the best model for forecasting the exchange rate of Indonesian rupiah against US dollar using Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) methods. LSTM and GRU are built with memory cells, allowing them to store and access information better. This study uses data from 2 January 2018 – 14 February 2023. The selection of the best model was obtained through hyperparameter tuning. The research concluded that the LSTM model with 1 unit, 32 batch sizes, 50 time steps, and 314 epochs is the best model, as it achieved small values of MSE (0,001465) and MAPE (0,2321061%).

## 1. PENDAHULUAN

Isu resesi ekonomi global pada tahun 2023 sedang banyak diperbincangkan di berbagai media. Indonesia adalah salah satu negara yang diprediksi akan mengalami resesi ekonomi di tahun 2023. Resesi ekonomi adalah penurunan aktivitas ekonomi yang signifikan dan berkelanjutan. Penyebab terjadinya resesi antara lain, adanya guncangan ekonomi, tingginya inflasi, kenaikan suku bunga acuan, dan penurunan permintaan global. Resesi menimbulkan dampak yang cenderung negatif, seperti permintaan ekspor menurun, melambatnya pertumbuhan ekonomi, serta meningkatnya harga barang dan jasa (Surya, 2022). Salah satu ciri suatu negara mengalami resesi ekonomi dapat dilihat dari melemahnya nilai tukar mata uang negaranya terhadap mata uang asing. Peramalan merupakan cara yang dapat dilakukan untuk mengetahui informasi perubahan kurs di masa yang akan datang. Peramalan merupakan aktivitas sistematis untuk memprediksi atau memperkirakan suatu hal yang akan terjadi di masa depan berdasarkan data masa lampau, (Moridu, et al., 2021).

Salah satu metode statistika yang menerapkan ilmu komputer dengan konsep *artificial intelligence* (kecerdasan buatan) pada metode peramalan adalah *neural network* (jaringan saraf tiruan). Jaringan saraf tiruan merupakan algoritma atau sistem yang merepresentasikan cara kerja neuron pada otak manusia dalam penyelesaian suatu masalah. Jaringan saraf tiruan memiliki sifat adaptif, yaitu mampu mempelajari data sebelumnya dan dapat mengenal pola data yang selalu berubah, sebagai dasar dalam pengambilan keputusan terhadap data yang belum dipelajari. *Long-Short Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU) adalah metode jaringan saraf tiruan yang dapat digunakan untuk peramalan. LSTM dan GRU merupakan varian dari *Recurrent Neural Network* (RNN) yang dikembangkan untuk dapat menyimpan dan mengakses informasi lebih baik. LSTM dan GRU dikembangkan dengan tujuan untuk menjaga dan mempertahankan informasi masa lalu yang penting serta menghapus informasi yang tidak relevan selama proses pembentukan model. Hal tersebut dikarenakan LSTM dan GRU dibangun dengan *memory cell*, sehingga LSTM dan GRU

memiliki memori jangka panjang yang lebih kuat serta menghasilkan model dengan akurasi yang lebih baik dibandingkan dengan RNN (Graves, 2013). GRU merupakan modifikasi dari model LSTM dengan arsitektur yang lebih sederhana dari model LSTM, sehingga memiliki komputasi atau perhitungan yang lebih efisien. Model LSTM dan GRU memiliki akurasi prediksi yang sebanding, sehingga penggunaan model LSTM atau GRU untuk prediksi dengan akurasi yang lebih baik bergantung pada data yang digunakan (Cho, et al., 2014).

Penelitian ini membahas mengenai peramalan nilai tukar rupiah terhadap dolar Amerika dengan menggunakan metode LSTM dan GRU. Penelitian dilakukan untuk mencari model peramalan terbaik antara model LSTM atau GRU untuk prediksi nilai tukar rupiah terhadap dolar Amerika. Selain itu, penelitian juga dilakukan agar dapat digunakan sebagai tambahan informasi bagi pembaca.

## 2. TINJAUAN PUSTAKA

Nilai tukar atau kurs (*exchange rate*) adalah nilai mata uang domestik dalam valuta asing atau harga mata uang dalam negeri yang diperlukan untuk membeli satu unit mata uang asing (Simorangkir & Suseno, 2004). Terdapat dua jenis perubahan nilai tukar, yaitu apresiasi dan depresiasi. Apresiasi terjadi ketika nilai tukar mata uang dalam negeri meningkat terhadap mata uang negara lain, sementara depresiasi terjadi ketika nilai tukar mata uang dalam negeri menurun terhadap mata uang negara lain (Moridu, et al., 2021).

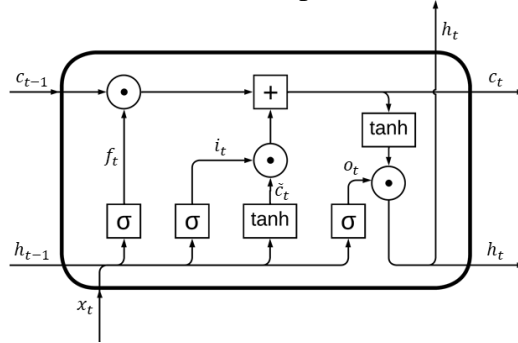
Jaringan saraf tiruan (JST) atau *artificial neural network* (ANN) merupakan sistem pemrosesan informasi yang menunjukkan karakteristik kinerja yang mirip dengan jaringan saraf biologis pada otak manusia. Kelebihan jaringan saraf tiruan, yaitu mampu memproses data non-linier dan kompleks, dapat memproses data yang bervariasi dan tidak terstruktur, serta adaptif atau mampu mempelajari dan menyesuaikan diri berdasarkan input yang berubah (Aggarwal, 2018). Jaringan saraf tiruan terbentuk oleh banyak neuron yang saling berhubungan dan secara umum terdiri atas input, bobot, fungsi jumlahan dan fungsi aktivasi, serta output. Bobot merupakan koefisien adaptif dalam jaringan yang menentukan intensitas input atau kekuatan antar neuron serta menyimpan memori model (Samarasinghe, 2006). Fungsi jumlahan berfungsi untuk menjumlahkan input yang telah diberi bobot. Fungsi aktivasi berfungsi sebagai pembatas nilai output, sehingga output yang dihasilkan berada dalam suatu interval tertentu.

*Recurrent Neural Network* (RNN) atau jaringan saraf berulang adalah salah satu metode dalam JST yang didesain untuk mempelajari pola pada data sekuensial atau data berurutan. Terdapat proses pengulangan pada algoritma RNN, sehingga jaringannya mampu menyimpan informasi terdahulu. Proses pengulangan tersebut adalah siklus dari penggunaan output yang diperoleh pada *hidden layer* sebelumnya sebagai input *hidden layer* saat ini untuk menghasilkan output saat ini. Pengulangan pada RNN menyebabkan adanya keadaan tersembunyi atau *hidden state* yang menyimpan hasil perhitungan yang telah dilakukan.

*Long-Short Term Memory* (LSTM) adalah pengembangan dari RNN dan pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber (1997). RNN memiliki memori jangka pendek yang baik tetapi memiliki memori jangka panjang yang buruk, akibat RNN tidak dapat menyimpan informasi masa lalu dalam jangka waktu yang lama. LSTM dibangun untuk mengatasi hal tersebut, yaitu dengan *memory cell* yang berfungsi untuk menyimpan dan mengingat informasi jangka panjang. *Memory cell* menyebabkan model mampu mengingat informasi penting dan relevan pada data masa lampau dengan jarak waktu yang jauh atau ketergantungan jangka panjang (*long-term dependencies*) (Graves, 2013).

*Memory cell* pada LSTM, yaitu *cell state*, dibangun dengan *fixed self-connection* (koneksi mandiri tetap) serta diatur melalui tiga *gate*, yaitu *forget gate*, *input gate*, dan *output*

*gate*. *Gate* dapat mempelajari informasi penting serta menyeleksi informasi untuk disimpan ataupun dibuang. *Forget gate* memutuskan informasi dalam *cell state* akan disimpan atau diatur ulang, *input gate* mengontrol informasi baru yang akan dikirim ke *cell state*, dan *output gate* mengontrol informasi tersimpan dalam *cell state* yang akan dikeluarkan (Hamoud, et al., 2022). Arsitektur LSTM cell dapat dilihat melalui Gambar 1.



**Gambar 1.** Arsitektur Long-Short Term Memory Cell

Algoritma pada LSTM cell adalah sebagai berikut (Pramesti, et al., 2022).

1. *Forget gate* ( $f_{(t)}$ ) menerima informasi dari input saat ini  $x_{(t)}$  dan *hidden state timestep* sebelumnya  $h_{(t-1)}$ . *Forget gate* akan memilih dan memutuskan banyak informasi dari *cell state* sebelumnya yang akan dibuang, dengan fungsi aktivasi sigmoid ( $\sigma$ ).

$$f_{(t)} = \sigma(\mathbf{b}_f + \mathbf{U}_f \mathbf{x}_{(t)} + \mathbf{W}_f \mathbf{h}_{(t-1)}) \quad (1)$$

2. *Input gate* ( $i_{(t)}$ ) akan memilih dan memutuskan banyak informasi yang akan dimasukkan ke dalam *cell state*, dengan fungsi aktivasi sigmoid ( $\sigma$ ).

$$i_{(t)} = \sigma(\mathbf{b}_i + \mathbf{U}_i \mathbf{x}_{(t)} + \mathbf{W}_i \mathbf{h}_{(t-1)}) \quad (2)$$

3. *Cell state candidate* ( $\tilde{c}_{(t)}$ ) diperoleh dengan fungsi aktivasi tanh.

$$\tilde{c}_{(t)} = \tanh(\mathbf{b}_c + \mathbf{U}_c \mathbf{x}_{(t)} + \mathbf{W}_c \mathbf{h}_{(t-1)}) \quad (3)$$

4. *Cell state* ( $c_{(t)}$ ) akan memperbarui nilai *cell state* lama menjadi nilai *cell state* baru, dengan menggabungkan informasi *forget gate* dengan *cell state* sebelumnya dan *input gate* dengan *cell state candidate*.

$$c_{(t)} = f_{(t)} \odot c_{(t-1)} + i_{(t)} \odot \tilde{c}_{(t)} \quad (4)$$

5. *Output gate* ( $o_{(t)}$ ) akan memilih dan memutuskan banyak informasi yang tersimpan dalam *cell state* untuk dikeluarkan dan digunakan sebagai input pada *hidden state*, dengan fungsi aktivasi sigmoid ( $\sigma$ ).

$$o_{(t)} = \sigma(\mathbf{b}_o + \mathbf{U}_o \mathbf{x}_{(t)} + \mathbf{W}_o \mathbf{h}_{(t-1)}) \quad (5)$$

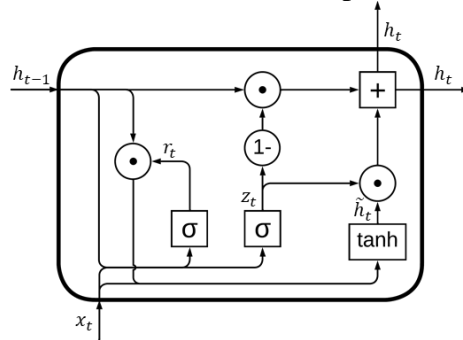
6. *Hidden state* baru ( $h_{(t)}$ ) diperoleh dari kombinasi antara output dengan nilai *cell state* baru, dengan fungsi aktivasi tanh.

$$h_{(t)} = o_{(t)} \odot \tanh(c_{(t)}) \quad (6)$$

dengan,  $\odot$  adalah perkalian *element-wise*,  $\mathbf{b}$  adalah bias, serta  $\mathbf{U}$  dan  $\mathbf{W}$  adalah bobot.

*Gated Recurrent Unit* (GRU) pertama kali diperkenalkan oleh Cho et al. (2014). GRU didesain untuk membuat setiap unit dapat mengatur ulang atau memperbarui informasi pada *memory cell* secara adaptif. GRU dapat dikatakan sebagai bentuk penyederhanaan dari LSTM, karena tidak menggunakan *cell state* secara eksplisit. Sehingga, pembaruan parameter dilakukan secara langsung pada *hidden state*. GRU membutuhkan *gate* yang lebih sedikit untuk melakukan pembaruan pada *hidden state*. *Reset gate* berfungsi untuk memutuskan

banyak informasi yang akan disimpan dari *hidden state* sebelumnya. *Update gate* berfungsi untuk mengontrol dan memutuskan informasi dari *hidden state* sebelumnya yang akan dipertahankan (Aggarwal, 2018). Arsitektur GRU cell dapat dilihat melalui Gambar 2.



**Gambar 2.** Arsitektur *Gated Recurrent Unit Cell*

Algoritma pada GRU cell adalah sebagai berikut (Chung, et al., 2014).

1. *Reset gate* ( $\mathbf{r}_{(t)}$ ) menerima input saat ini  $\mathbf{x}_{(t)}$  dan *hidden state timestep* sebelumnya  $\mathbf{h}_{(t-1)}$ . *Reset gate* akan memilih dan menentukan banyak informasi yang akan disimpan dari *hidden state* sebelumnya, dengan fungsi aktivasi sigmoid ( $\sigma$ ).

$$\mathbf{r}_{(t)} = \sigma(\mathbf{b}_r + \mathbf{U}_r \mathbf{x}_{(t)} + \mathbf{W}_r \mathbf{h}_{(t-1)}) \quad (7)$$

2. *Update gate* ( $\mathbf{z}_{(t)}$ ) akan mengontrol banyak informasi dari *hidden state* sebelumnya yang akan dipertahankan, dengan fungsi aktivasi sigmoid ( $\sigma$ ).

$$\mathbf{z}_{(t)} = \sigma(\mathbf{b}_z + \mathbf{U}_z \mathbf{x}_{(t)} + \mathbf{W}_z \mathbf{h}_{(t-1)}) \quad (8)$$

3. *Hidden state candidate* ( $\tilde{\mathbf{h}}_{(t)}$ ) dibuat, dengan fungsi aktivasi tanh.

$$\tilde{\mathbf{h}}_{(t)} = \tanh(\mathbf{b}_h + \mathbf{U}_h \mathbf{x}_{(t)} + \mathbf{W}_h [\mathbf{r}_{(t)} \odot \mathbf{h}_{(t-1)}]) \quad (9)$$

4. *Hidden state* baru ( $\mathbf{h}_{(t)}$ ) dibuat dari kombinasi antara nilai *reset gate* dan *hidden state candidate* dengan *reset gate* dengan *hidden state* sebelumnya.

$$\mathbf{h}_{(t)} = (1 - \mathbf{z}_{(t)}) \odot \mathbf{h}_{(t-1)} + \mathbf{z}_{(t)} \odot \tilde{\mathbf{h}}_{(t)} \quad (10)$$

dengan,  $\odot$  adalah perkalian *element-wise*,  $\mathbf{b}$  adalah bias, serta  $\mathbf{U}$  dan  $\mathbf{W}$  adalah bobot.

Pada LSTM dan GRU, nilai *hidden state* merupakan output dari LSTM cell atau GRU cell. Nilai *hidden state* kemudian akan digunakan sebagai input pada *output layer* untuk diolah dan menghasilkan output akhir berupa prediksi output.

Fungsi aktivasi dalam jaringan saraf tiruan berfungsi untuk membuat sistem pembelajaran serta memampukan model untuk menjalankan tugas sulit dan kompleks, sehingga jaringan saraf menjadi lebih kuat. Suatu model tetap dapat berjalan secara linier tanpa fungsi aktivasi, namun prosesnya kurang efisien dan hasilnya kurang akurat. Fungsi aktivasi non-linier atau *non-linear activation function* digunakan untuk menghasilkan variasi non-linier dalam jaringan saraf. Fungsi aktivasi non-linier merupakan faktor penting untuk meningkatkan kinerja sistem pengenalan pada model. Fungsi aktivasi dalam arsitektur LSTM dan GRU adalah fungsi aktivasi non-linier, yaitu fungsi aktivasi sigmoid dan fungsi aktivasi tanh (Goodfellow, et al., 2016).

Fungsi aktivasi sigmoid memiliki bentuk grafik “S” dan sering digunakan dalam JST (Haykin, 2009). Fungsi aktivasi sigmoid memiliki range 0 hingga 1, sehingga akan menghasilkan output yang bernilai positif dan berada di antara 0 dan 1. Nilai output dalam rentang 0 hingga 1 bermanfaat dalam klasifikasi biner, sehingga dapat mengendalikan

informasi yang akan disimpan atau dibuang (Aggarwal, 2018). Fungsi aktivasi sigmoid didefinisikan pada persamaan (11).

$$\begin{aligned} \sigma: \mathbb{R} &\rightarrow (0,1) \\ \sigma(x) &= \frac{1}{1 + e^{-x}} \end{aligned} \quad (11)$$

Turunan pertama fungsi aktivasi sigmoid dapat dilihat pada persamaan (12).

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (12)$$

Fungsi aktivasi tanh atau *hyperbolic tangent* merupakan fungsi aktivasi yang memiliki range -1 hingga 1, sehingga akan menghasilkan output yang nilainya berada di antara -1 dan 1. Fungsi aktivasi tanh memiliki rentang yang lebih luas, sehingga mampu menangani informasi positif dan negatif dengan baik (Goodfellow, et al., 2016). Fungsi aktivasi tanh didefinisikan pada persamaan (13).

$$\begin{aligned} \tanh: \mathbb{R} &\rightarrow (-1,1) \\ \tanh(x) &= \frac{e^{2x} - 1}{e^{2x} + 1} \end{aligned} \quad (13)$$

Turunan pertama fungsi aktivasi tanh dapat dilihat pada persamaan (14).

$$\tanh'(x) = 1 - \tanh^2(x) \quad (14)$$

*Hyperparameter* merupakan parameter dari algoritma pembelajaran untuk mengontrol perilaku algoritma pembelajaran. Oleh karena itu, *hyperparameter* tidak terpengaruh oleh algoritma pembelajaran serta harus ditentukan sebelum pelatihan. Penggunaan *hyperparameter* yang tepat merupakan hal penting dalam membangun model yang baik. Penjelasan setiap *hyperparameter* yang digunakan dalam LSTM dan GRU adalah sebagai berikut (Goodfellow, et al., 2016).

1. *Feature* mengacu pada variabel yang digunakan oleh algoritma. Kinerja model dipengaruhi oleh kualitas dan jumlah *feature* yang digunakan. Penggunaan *feature* dengan kualitas yang buruk akan mengakibatkan model kesulitan untuk mempelajari pola-pola pada data serta akan menghasilkan model yang memiliki akurasi rendah.
2. *Unit* atau neuron merupakan *unit* dasar pada jaringan saraf tiruan. Banyak *unit* dapat mempengaruhi kinerja dan akurasi model, karena unit berfungsi untuk menerima dan memproses input untuk menghasilkan output.
3. *Batch size* adalah jumlah sampel data yang digunakan dalam setiap *epoch* selama proses pelatihan. *Batch size* dapat mempengaruhi kecepatan proses pelatihan, konvergensi, dan keakuratan model. *Batch size* yang terlalu kecil dapat memperlambat proses pelatihan dan menghasilkan bobot yang tidak stabil. Sedangkan *batch size* yang terlalu besar akan mengakibatkan model mengalami *overfitting* serta kebutuhan memori yang besar untuk menyimpan aktivasi dan gradien model. *Overfitting* terjadi karena model menjadi terlalu fokus pada data pelatihan, sehingga data baru yang belum pernah dilihat sebelumnya tidak dapat digeneralisasi dengan baik.
4. *Timestep* merupakan jumlah langkah waktu yang diperlukan untuk merepresentasikan data berurutan. Semakin besar nilai *timestep*, maka semakin banyak pula informasi masa lalu yang dapat digunakan model untuk memprediksi nilai selanjutnya. Akan tetapi, nilai *timestep* yang terlalu besar dapat memperlambat proses pembelajaran.
5. *Learning rate* menentukan besar perubahan bobot model pada setiap iterasi selama proses pelatihan dan dapat mempengaruhi konvergensi model, waktu yang dibutuhkan pada proses pelatihan, serta keakuratan prediksi. Nilai *learning rate* yang terlalu kecil akan memperpanjang waktu model dalam mencapai konvergensi, akibat nilai bobot yang

menjadi sangat kecil. Sebaliknya, nilai *learning rate* yang terlalu besar dapat mempersulit model mencapai konvergensi, akibat nilai bobot yang menjadi sangat besar.

6. *Epoch* merupakan satu kali proses *training* pada seluruh data *training*. Pada setiap *epoch*, model akan memperbarui bobot berdasarkan gradien dari kesalahan data *training*. *Epoch* yang terlalu kecil dapat mengakibatkan pelatihan model menjadi kurang baik.
7. *Optimizer* merupakan metode optimasi berbasis gradien stokastik untuk memaksimalkan atau meminimalkan bobot berdasarkan data *training*. Penelitian ini menggunakan *optimizer* Adam (*Adaptive Moment Estimation*) yang menggunakan *adaptive learning rate* dan momentum untuk mempercepat proses pelatihan. Adam dirancang untuk memperoleh keuntungan dari dua *optimizer*, yaitu AdaGrad (dapat bekerja dengan baik pada gradien yang jarang dengan menyesuaikan laju pembelajaran) dan RMSProp (dapat bekerja dengan baik pada pengaturan *online* dan non-stasioner) (Kingma & Ba, 2015).
8. *Loss function* merupakan fungsi yang digunakan untuk mengukur kehilangan atau kesalahan model dalam melakukan prediksi selama proses pelatihan. *Loss function* digunakan sebagai tanda untuk memperbaiki model dengan meminimalkan kesalahan dan memaksimalkan akurasi saat proses pembelajaran. *Loss function* yang digunakan dalam penelitian ini adalah MSE.

*Feature Scaling* merupakan metode *preprocessing data* yang mentransformasikan data numerik dengan rentang yang berbeda ke dalam skala yang sama. *Feature scaling* dilakukan untuk meningkatkan stabilitas data, menghindari kegagalan selama proses pembelajaran (*learning process*), mengurangi waktu dalam proses pembelajaran model peramalan, dan meningkatkan signifikansi kinerja model peramalan. *Feature scaling* dapat berupa standarisasi data atau normalisasi data. *Min-Max Normalization* merupakan salah satu metode normalisasi data dan dilakukan dengan tujuan untuk mengubah skala data ke dalam interval tertentu (Singh & Singh, 2020). Rumus *Min-Max Normalization* dengan interval [a,b] didefinisikan pada persamaan (15).

$$x'_i = a + \frac{(x_i - \min(x))(b - a)}{\max(x) - \min(x)} \quad (15)$$

Denormalisasi data merupakan proses pengembalian data output yang dihasilkan dari peramalan menggunakan data yang dinormalisasi menjadi bentuk asli. Denormalisasi perlu dilakukan untuk memperoleh data hasil prediksi sebenarnya. Rumus *Min-Max Denormalization* dengan interval [a,b] didefinisikan pada persamaan (17).

$$x''_i = \frac{(x'_i - a)(\max(x) - \min(x))}{(b - a)} + \min(x) \quad (16)$$

dengan,  $x_i$  adalah data aktual ke- $i$ ,  $x'_i$  adalah data normalisasi ke- $i$ ,  $x''_i$  adalah data denormalisasi ke- $i$ ,  $\min(x)$  adalah nilai minimum data  $x$ , dan  $\max(x)$  adalah nilai maksimum data  $x$ .

*Data set* adalah kumpulan data yang terorganisir dan terstruktur serta berkaitan dengan suatu topik tertentu yang saling berhubungan satu sama lain. *Lagged data set* merupakan *data set* yang akan ditransformasi dengan menambahkan *lagged* variabel. *Lag* mengacu pada penundaan waktu dalam data deret waktu. Maka *lagged* variabel dapat diartikan sebagai variabel yang berisi data pada waktu sebelumnya. *Lagged data set* berguna untuk mengakses data di masa lampau dan memperhitungkan hubungan temporal antar data (hubungan antara nilai-nilai data pada titik waktu tertentu dengan nilai-nilai data pada titik waktu sebelumnya atau setelahnya) agar model mampu mempelajari ketergantungan temporal yang ada pada data, sehingga meningkatkan akurasi prediksi di masa depan (Shumway & Stoffer, 2010).

Akurasi diperlukan sebagai kriteria utama dalam pemilihan metode peramalan. Akurasi mengacu pada tingkat kebaikan model dalam meramalkan data yang sudah diketahui dan digunakan dalam pemodelan (data *training*). Tingkat akurasi yang baik antara model peramalan dengan data *training* tidak selalu menghasilkan peramalan yang baik. Sehingga, perlu dilakukan pengukuran akurasi dengan melakukan pengujian model menggunakan data *testing*. Pengukuran akurasi dalam penelitian ini menggunakan MSE dalam pelatihan model dan MAPE dalam pengujian model.

*Mean Square Error* atau MSE merupakan salah satu metode statistik yang digunakan dalam pengukuran keakuratan peramalan model. MSE memiliki kelebihan, yaitu lebih mudah ditangani secara matematis. Pemilihan model yang lebih baik dilihat berdasarkan nilai MSE yang lebih kecil. Rumus umum MSE didefinisikan pada persamaan (17) (Makridakis, et al., 1997).

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (17)$$

dengan,  $t$  adalah waktu,  $n$  adalah ukuran sampel,  $y_t$  adalah nilai data aktual waktu ke- $t$ , dan  $\hat{y}_t$  adalah nilai data hasil prediksi waktu ke- $t$ .

*Mean Absolute Percentage Error* atau MAPE adalah salah satu ukuran akurasi prediksi yang populer digunakan dalam industri praktisi, karena MAPE tidak bergantung pada skala dan mudah diinterpretasikan. Rumus MAPE didefinisikan pada persamaan (18) (Byrne, 2012).

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\% \quad (18)$$

dengan,  $t$  adalah waktu,  $n$  adalah ukuran sampel,  $y_t$  adalah nilai data aktual waktu ke- $t$ , dan  $\hat{y}_t$  adalah nilai data hasil prediksi waktu ke- $t$ .

### 3. METODE PENELITIAN

Penelitian ini menggunakan data nilai tukar rupiah terhadap dolar Amerika, yang diperoleh dari website [www.bi.go.id](http://www.bi.go.id) oleh Bank Indonesia. Periode data yang digunakan adalah mulai tanggal 2 Januari 2018 hingga 14 Februari 2023, yaitu sebanyak 1.252 data. Data akan dibagi menjadi dua, yaitu data *training* (data pelatihan) dan data *testing* (data pengujian), dengan proporsi 80% : 20%. Sebelum diolah, data akan diubah menjadi *lagged data set* sesuai dengan jumlah *timestep* yang digunakan. Penelitian ini menggunakan metode *Long-Short Term Memory* dan *Gated Recurrent Unit* dengan bantuan *software* RStudio. Tahapan analisis yang dilakukan adalah sebagai berikut.

1. Memasukkan data yang akan digunakan pada *software* RStudio.
2. Membuat visualisasi data dan melakukan pengujian statistika deskriptif.
3. Melakukan normalisasi data.
4. Membuat *lagged data set*.
5. Membagi data menjadi data *training* dan data *testing*.
6. Membuat rencana pembentukan model LSTM dan GRU, dengan *hyperparameter* yang ditentukan sebelum proses pembelajaran yaitu: *feature* 1; *unit* 1 dan 50; *batch size* 1, 32, dan 64; *timestep* 1 dan 50; *learning rate* 0,001; *optimizer* Adam; *epoch* 1.000; dan *loss function* MSE, dengan *early stopping* yaitu saat nilai *loss* pada *epoch* sebelumnya dan *epoch* saat ini memiliki selisih kurang dari 0,000001.
7. Melakukan pelatihan model LSTM dan GRU dengan data *training*.

8. Menghitung MSE untuk evaluasi model dengan data *training*.
9. Melakukan pengujian model LSTM dan GRU dengan data *testing*.
10. Melakukan denormalisasi data hasil prediksi dengan data *testing*.
11. Menghitung MAPE untuk evaluasi hasil prediksi dengan data *testing*.
12. Pemilihan model LSTM dan model GRU terbaik berdasarkan nilai MSE dan MAPE.
13. Pengambilan kesimpulan.

#### 4. HASIL DAN PEMBAHASAN

Normalisasi data dengan Min-Max Normalization dilakukan dengan tujuan untuk mengubah skala data menjadi berada pada suatu interval tertentu. Normalisasi data pada penelitian ini menggunakan interval  $[-1,1]$ , karena pada pemodelan LSTM dan GRU menggunakan fungsi aktivasi tanh yang memadamkan data pada interval  $[-1,1]$ .

Data yang telah dinormalisasi akan dibuat menjadi *lagged data set* sebelum dilakukan pengolahan data lebih lanjut. Data runtun waktu akan ditransformasi menjadi *data set* dengan dua variabel, yaitu variabel independen (X) dan variabel dependen (Y), dengan variabel X merupakan *lagged* variabel. Kemudian, banyak variabel X ditentukan berdasarkan jumlah *timestep*, yaitu sebanyak 1 dan 50, dengan setiap variabel X juga merupakan *lagged* variabel.

Setiap variabel pada *lagged data set* kemudian akan dibagi dua menjadi data *training* dan data *testing*. Pada *data set* dengan *timestep* 1, data *training* yang terbentuk berjumlah 1.000 data dan data *testing* berjumlah 251 data. Sedangkan pada *data set* dengan *timestep* 50, data *training* yang terbentuk berjumlah 961 data dan data *testing* berjumlah 241 data.

Tahap selanjutnya adalah membuat model *Long-Short Term Memory* (LSTM) dan model *Gated Recurrent Unit* (GRU). Model LSTM atau GRU terbaik ditentukan dengan penyesuaian *hyperparameter*, yaitu dengan membentuk beberapa model menggunakan kombinasi dari *hyperparameter unit*, *batch size*, dan *timestep*. Kemudian model LSTM dan GRU akan dilatih menggunakan data *training* dan *hyperparameter* yang telah ditentukan sebelumnya. Pelatihan model dilakukan dengan *epoch* 1.000, yang artinya model akan dilatih sebanyak 1.000 kali hingga mencapai konvergensi. Proses pelatihan model akan berhenti jika nilai MSE dari *loss function* pada *epoch* sebelumnya dengan *epoch* saat ini memiliki selisih kurang dari sama dengan 0,000001. Setelah pelatihan model, akan dilakukan pengujian model dengan data *testing*. Hasil pelatihan dan pengujian seluruh model LSTM dan model GRU berupa nilai evaluasi model, yaitu MSE dari pelatihan model dan MAPE dari pengujian model dapat dilihat pada Tabel 1.

**Tabel 1.** Hasil Pelatihan dan Pengujian Model LSTM dan Model GRU

Model	Jenis Model	Unit	Batch Size	Timestep	Epoch	MSE Training	MAPE Testing
Model1.1.1	LSTM	1	1	1	85	0,002705733	0,003272611
Model1.1.50	LSTM	1	1	50	37	0,001813173	0,003598405
Model1.32.1	LSTM	1	32	1	522	0,001738949	0,002953895
Model1.32.50	LSTM	1	32	50	314	0,001451447	0,002321061
Model1.64.1	LSTM	1	64	1	771	0,001847914	0,003198559
Model1.64.50	LSTM	1	64	50	493	0,001482736	0,002314096
Model50.1.1	LSTM	50	1	1	294	0,002020418	0,004451178
Model50.1.50	LSTM	50	1	50	Tidak dapat mencapai konvergensi		
Model50.32.1	LSTM	50	32	1	66	0,001459359	0,002325521
Model50.32.50	LSTM	50	32	50	107	0,001344096	0,003223758
Model50.64.1	LSTM	50	64	1	114	0,001466461	0,002331531
Model50.64.50	LSTM	50	64	50	123	0,001445421	0,003462715



Model	Jenis Model	Unit	Batch Size	Timestep	Epoch	MSE Training	MAPE Testing
Model1_1_1	GRU	1	1	1	69	0,002403372	0,003240596
Model1_1_50	GRU	1	1	50	39	0,002016613	0,003289181
Model1_32_1	GRU	1	32	1	469	0,001503593	0,002321311
Model1_32_50	GRU	1	32	50	211	0,001518126	0,002628067
Model1_64_1	GRU	1	64	1	597	0,001671959	0,002720061
Model1_64_50	GRU	1	64	50	317	0,00157408	0,002712309
Model50_1_1	GRU	50	1	1	340	0,002254626	0,004802171
Model50_1_50	GRU	50	1	50	Tidak dapat mencapai konvergensi		
Model50_32_1	GRU	50	32	1	22	0,001475483	0,002412439
Model50_32_50	GRU	50	32	50	145	0,002299691	0,006658798
Model50_64_1	GRU	50	64	1	38	0,001476238	0,002424479
Model50_64_50	GRU	50	64	50	219	0,001239672	0,003696074

Model LSTM dan model GRU dengan *unit* 50, *batch size* 1, dan *timestep* 50 tidak dapat mencapai konvergensi selama proses pelatihan model akibat model yang terlalu kompleks, yaitu *hyperparameter unit* dan *timestep* yang terlalu besar, namun *batch size* terlalu kecil. Hal tersebut, mengakibatkan model kesulitan untuk mengenali pola dan menggeneralisasi data dengan baik selama proses pelatihan. Selain itu, berdasarkan Tabel 1 dapat terlihat bahwa penggunaan *hyperparameter* yang berbeda akan mempengaruhi kecepatan proses pelatihan, konvergensi model, serta akurasi prediksi model. Sehingga, penting untuk memperhatikan penggunaan *hyperparameter* yang tepat dalam membangun model.

Model peramalan terbaik ditentukan dengan mempertimbangkan kedua nilai MSE dan MAPE secara bersama-sama. Berdasarkan Tabel 1 dapat ditentukan bahwa model LSTM dengan *unit* 1, *batch size* 32, *timestep* 50, dan *epoch* 314 merupakan model LSTM terbaik, karena memiliki nilai MSE dan MAPE yang kecil, yaitu 0,001451447 untuk MSE dan 0,2321061% untuk MAPE. Sedangkan model GRU dengan *unit* 50, *batch size* 32, *timestep* 1, dan *epoch* 22 merupakan model GRU terbaik, karena memiliki nilai MSE dan MAPE yang kecil, yaitu 0,001475483 untuk MSE dan 0,2412439% untuk MAPE.

**Tabel 2.** Perbandingan Hasil Pelatihan dan Pengujian Model Terbaik

Model	Epoch	MSE Training	MAPE Testing
Model1_32_50	314	0,001451447	0,002321061
Model50_32_1	22	0,001475483	0,002412439

Model terbaik di antara seluruh model LSTM atau model GRU dapat ditetapkan dengan membandingkan kedua model LSTM dan model GRU terbaik yang telah diperoleh sebelumnya. Nilai MSE dan MAPE antar kedua model pada Tabel 2 tidak berbeda jauh. Hal tersebut dapat terlihat dari selisih nilai MSE ataupun MAPE antar model yang kecil, dengan selisih MSE sebesar 0,000024036 dan selisih MAPE sebesar 0,000091378. Namun, model LSTM dengan *unit* 1, *batch size* 32, *timestep* 50, dan *epoch* 314 memiliki nilai MSE dan MAPE yang lebih rendah dibandingkan model GRU dengan *unit* 50, *batch size* 32, *timestep* 1, dan *epoch* 22. Sehingga, dapat disimpulkan bahwa model LSTM dengan *unit* 1, *batch size* 32, *timestep* 50, dan *epoch* 314 merupakan model peramalan terbaik untuk prediksi nilai tukar rupiah terhadap dolar Amerika serta dapat digunakan sebagai referensi untuk mengamati perubahan nilai tukar rupiah terhadap dolar Amerika di masa yang akan datang.

## 5. KESIMPULAN

Kesimpulan yang diperoleh berdasarkan hasil analisis dan pembahasan mengenai peramalan nilai tukar rupiah terhadap dolar Amerika dengan metode *Long-Short Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU), yaitu model LSTM dengan

hyperparameter unit 1, batch size 32, timestep 50, epoch 314, learning rate 0,001, dan optimizer Adam merupakan model peramalan yang paling baik untuk prediksi nilai tukar rupiah terhadap dolar Amerika. Hal ini dikarenakan model menghasilkan nilai MSE dan MAPE yang kecil, berturut-turut sebesar 0,001451447 dan 0,2321061%.

## DAFTAR PUSTAKA

- Aggarwal, C. C., 2018. *Neural Networks and Deep Learning: A Textbook*. New York: Springer.
- Byrne, R. F., 2012. Beyond Traditional Time-Series: Using Demand Sensing to Improve Forecasts in Volatile Times. *Journal of Business Forecasting*, Volume 31, p. 13.
- Cho, K. et al., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *arXiv*.
- Chung, J., Gulcehre, C., Cho, K. & Bengio, Y., 2014. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, s.l.: arXiv.
- Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. Cambridge(Massachusetts): MIT Press.
- Graves, A., 2013. *Generating Sequences With Recurrent Neural Networks*, s.l.: arXiv.
- Hamoud, A. A., Hoenig, A. & Roy, K., 2022. Sentence subjectivity analysis of a political and ideological debate dataset using LSTM and BiLSTM with attention and GRU models. *Journal of King Saud University – Computer and Information Sciences*, Volume 34, pp. 7974-7987.
- Haykin, S., 2009. *Neural Networks and Learning Machines*. 3rd ed. Upper Saddle River(New Jersey): Pearson Prentice Hall.
- Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp. 1735-1780.
- Kingma, D. P. & Ba, J. L., 2015. *Adam: A Method for Stochastic Optimization*. San Diego, arXiv.
- Makridakis, S. G., Wheelwright, S. C. & Hyndman, R. J., 1997. *Forecasting: Methods and Applications*. 3rd ed. New York: John Wiley & Sons.
- Moridu, I. et al., 2021. *Manajemen Keuangan Internasional*. Bandung: Widina Bhakti Persada.
- Pramesti, D. D., Novitasari, D. C. R., Setiawan, F. & Khaulasari, H., 2022. Long-Short Term Memory (LSTM) for Predicting Velocity and Direction Sea Surface Current on Bali Strait. *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, 16(2), pp. 451-462.
- Samarasinghe, S., 2006. *Neural Networks for Applied Sciences and Engineering*. Boca Raton: Auerbach Publications.
- Shumway, R. H. & Stoffer, D. S., 2010. *Time Series Analysis and Its Applications: With R Examples*. 3rd ed. New York: Springer.
- Simorangkir, I. & Suseno, 2004. *Sistem dan Kebijakan Nilai Tukar*. Jakarta: Pusat Pendidikan dan Studi Kebanksentralan (PPSK) BI.
- Singh, D. & Singh, B., 2020. Investigating The Impact of Data Normalization on Classification Performance. *Applied Soft Computing Journal*, Volume 97b.
- Surya, T. A., 2022. Mewaspada Ancaman Resesi Ekonomi Global. *Info Singkat*, XIV(19).