

## Penerapan Metode Long Short Term Memory untuk Menentukan Prediksi Inflasi di Indonesia

**Diah Santika Kusumah Dimiyati\*, Eti Kurniati, Onoy Rohaeni**

Prodi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Islam Bandung, Indonesia.

\*diah.santika05@gmail.com, etika0101@gmail.com, onoyrohaeni@gmail.com

**Abstract.** Inflation reflects the economic stability of a region, which affects monetary policy, investment, and public welfare. Forecasting the rate of inflation is necessary as an anticipation measure to control future inflation rates. This research aims to predict inflation in Indonesia. The method used is Long Short Term Memory (LSTM), assisted by the Python programming language. The research results show that the LSTM model was built with a single hidden layer architecture with a combination of hyperparameters: 5 neurons, a batch size of 32, and a learning rate of 0.1, trained for 200 epochs with a composition of 90% training data and 10% testing data using Adam optimization, resulting in a MAPE (Mean Absolute Percentage Error) of 9.2533% for training data and 9.91525% for testing data. The inflation rate in Indonesia for July is predicted to be 2.8282988%.

**Keywords:** *Inflation, Prediction, LSTM.*

**Abstrak.** Inflasi mencerminkan stabilitas ekonomi suatu wilayah yang mempengaruhi kebijakan moneter, investasi dan kesejahteraan masyarakat. Prediksi laju inflasi diperlukan sebagai antisipasi untuk mengendalikan laju inflasi di masa mendatang. Penelitian ini bertujuan untuk memprediksi inflasi di Indonesia. Metode yang digunakan adalah *Long Short Term Memory* dibantu dengan bahasa pemrograman python. Hasil penelitian menunjukkan bahwa model LSTM dibangun dengan arsitektur satu hidden layer dengan kombinasi *hyperparameter* 5 neuron, 32 *batch size* dan 0.1 *learning rate* yang dilatih selama 200 epoch dengan komposisi 90% data *training* dan 10% data *testing* menggunakan optimasi adam menghasilkan MAPE data training 9.2533% dan MAPE data testing 9.91525%. Tingkat inflasi Indonesia pada bulan Juli diprediksi sebesar 2.8282988%.

**Kata Kunci:** *Inflasi, Prediksi, LSTM.*

## A. Pendahuluan

Pertumbuhan ekonomi memiliki peran penting dalam kemajuan pembangunan dan keberlanjutan suatu negara. Sebagai inti dari berbagai kehidupan masyarakat, permasalahan ekonomi mempengaruhi kesejahteraan dan stabilitas suatu bangsa. Inflasi merupakan salah satu permasalahan dalam pertumbuhan ekonomi Indonesia. Inflasi merupakan kondisi gambaran yang terjadi ketika harga-harga menghadapi peningkatan baik secara bentuk barang maupun jasa yang umumnya berlangsung secara berkesinambungan dalam rentang jangka waktu yang lama [1]. Inflasi yang tinggi memiliki dampak yang dapat dirasakan langsung oleh masyarakat seperti menurunnya daya beli masyarakat, mengurangi minat masyarakat untuk menabung, merusak struktur ekonomi, memburuknya distribusi pendapatan dan meningkatnya suku bunga.

Tingkat inflasi mencerminkan stabilitas ekonomi suatu wilayah. Inflasi yang tinggi atau tidak stabil memiliki dampak yang luas terhadap berbagai sektor perekonomian, sehingga menimbulkan tantangan yang harus di atasi oleh pemerintah, bank sentral dan pelaku ekonomi. Antisipasi dan tindakan perlu dilakukan untuk mengendalikan tingkat inflasi sehingga tidak melambung terlalu tinggi. Prediksi laju inflasi diperlukan sebagai antisipasi untuk mengendalikan laju inflasi di masa mendatang.

Banyak metode yang dapat digunakan untuk memprediksi data *time series*, namun penting untuk menyesuaikan penggunaannya dengan karakteristik dan variabel dari data tersebut agar diperoleh model yang terbaik. Secara umum, teknik peramalan deret waktu terbagi menjadi dua kategori utama yaitu metode statistika dan metode *computational intelligence* [2] .

*Recurrent Neural Network* merupakan salah satu algoritma yang termasuk dalam *Deep Learning*. *Long Short Term Memory* (LSTM) merupakan jenis RNN yang dikembangkan untuk menangani masalah memori panjang dalam data. LSTM mampu mempelajari pola-pola jangka panjang dan jangka pendek dalam data. Kemampuan ini membuat LSTM menjadi model yang potensial untuk digunakan dalam memprediksi inflasi di Indonesia.

Python merupakan bahasa pemrograman yang populer dan mudah digunakan untuk berbagai aplikasi, termasuk analisis data dan *machine learning*. Python digunakan untuk menyediakan alat dan library untuk membangun, melatih, dan menganalisis model LSTM. Berdasarkan uraian di atas, pada penelitian ini akan menerapkan metode *Long-Short Term Memory* (LSTM) menggunakan python untuk menentukan prediksi inflasi di Indonesia di masa depan.

## B. Metodologi Penelitian

### Inflasi

Inflasi merupakan kondisi kenaikan harga barang dan jasa secara umum dan terus menerus[3]. Kenaikan harga pada satu atau dua barang saja tidak dianggap sebagai inflasi kecuali jika kenaikan harga tersebut meluas atau berdampak pada kenaikan harga barang lainnya. Indeks Harga Konsumen (IHK) sering digunakan sebagai indikator untuk mengukur tingkat inflasi. Perubahan IHK dari waktu ke waktu mencerminkan perubahan harga dari berbagai barang dan jasa yang di konsumsi oleh Masyarakat.

### Prediksi

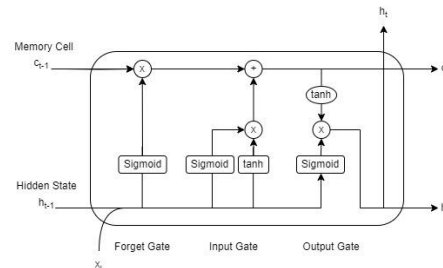
Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil [4]. Prediksi memiliki peranan penting dalam berbagai bidang termasuk ekonomi, layanan kesehatan dan teknologi untuk membantu mengambil keputusan atau kebijakan terkait dengan memperkirakan apa yang mungkin terjadi di masa depan.

### Recurrent Neural Network

*Recurrent Neural Network* (RNN) merupakan proses yang dilakukan secara berulang untuk memproses input (umumnya data sekuensial) sebagai bagian dari deep learning[5]. RNN memiliki koneksi yang melingkar atau kembali ke dirinya sendiri, yang memungkinkan informasi untuk dibawa dari satu langkah waktu ke langkah waktu berikutnya.

### Long Short Term Memory (LSTM)

*Long Short Term Memory* (LSTM) pertama kali diusulkan oleh Hochreiter dan Schmidhuber pada tahun 1997[6]. *Long-Short Term Memory* (LSTM) merupakan bagian dari *Recurrent Neural Networks* (RNN) yang digunakan dalam bidang *Deep Learning* yang mampu mengatasi keterbatasan memori jangka pendek [7]. Struktur RNN memiliki keterbatasan dalam menangani ketergantungan jangka panjang karena kurang efektif dalam menyimpan informasi sebelumnya.



**Gambar 1.** Struktur Sel Long Short Term Memory

Setiap sel LSTM terdiri dari tiga gerbang yaitu *forget gate*, *input gate* dan *output gate*. Gerbang dalam LSTM adalah elemen utama yang mengatur aliran informasi sepanjang urutan waktu. Setiap gerbang memiliki fungsi tertentu yang mempengaruhi cara informasi disimpan, dilupakan, atau diproses. Persamaan yang digunakan adalah sebagai berikut[8]:

1. Forget gate  

$$f_t = \sigma(W_f \cdot [h_{t-1}, x'_t] + b_f)$$
2. Input gate  

$$i_t = \sigma(W_i \cdot [h_{t-1}, x'_t] + b_i)$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x'_t] + b_c)$$
3. Memperbarui Memory Cell  

$$C_t = f_t * C_{t-1} + i_t * C'_t$$
4. Output gate  

$$o_t = \sigma(W_o \cdot [h_{t-1}, x'_t] + b_o)$$
5. Memperbarui hidden state  

$$h_t = o_t * \tanh(C_t)$$

### Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan neuron[9]. Fungsi sigmoid dan tanh merupakan fungsi aktivasi digunakan dalam *long short term memory*. Fungsi sigmoid merupakan fungsi digunakan untuk mengubah nilai input ke rentang antara 0 dan 1. Fungsi tanh atau tangen hiperbolik menghasilkan nilai keluaran yang berkisar antara -1 dan 1. Fungsi sigmoid dan  $\tanh(x)$  terdapat pada persamaan sebagai berikut [8]:

1. Fungsi sigmoid:  

$$\sigma(x) = \frac{1}{1+e^{-x}}$$
2. Fungsi tanh:  

$$\tanh(x) = 2\sigma(2x) - 1$$

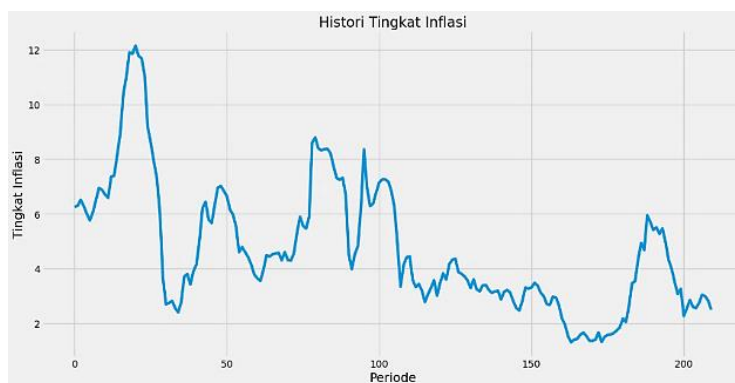
### Backpropagation

*Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak *layer* untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada hidden layer [10]. *Backpropagation* dihitung setelah nilai *loss* diperoleh. Proses ini bertujuan untuk menghitung gradien pada setiap layer. Gradien yang dihitung kemudian digunakan untuk memperbarui bobot di setiap layer LSTM.

### C. Hasil Penelitian dan Pembahasan

#### Sumber Data

Penelitian ini menggunakan data bulanan inflasi Indonesia yang diambil dari *website* Bank Indonesia. Data yang diambil untuk penelitian ini yaitu data inflasi dari bulan Januari 2007 hingga Desember 2023. Pergerakan laju Inflasi dapat dilihat pada Gambar 2.



**Gambar 2.** Laju Inflasi di Indonesia

#### Langkah-Langkah Penerapan Metode LSTM

##### Normalisasi Data

Proses ini dilakukan untuk mempersiapkan data mentah agar siap digunakan dalam model pembelajaran. Proses normalisasi data dilakukan untuk menyamakan skala data dengan skala [0,1]. Normalisasi data dilakukan dengan *MinMaxScaler*. Nilai  $X_{max}$  yaitu 12.14 yang merupakan data inflasi September 2008 dan  $X_{min}$  yaitu 1.32 yang merupakan data inflasi Agustus 2020.

$$x'_t = \frac{(x - X_{min})}{(X_{max} - X_{min})}$$

$$\text{Bulan ke-1} : x'_1 = \frac{6.26 - 1.32}{12.14 - 1.32} = 0.456562$$

$$\text{Bulan ke-2} : x'_2 = \frac{6.3 - 1.32}{12.14 - 1.32} = 0.460259$$

Normalisasi data dilakukan hingga data terakhir, hasil normalisasi data terdapat pada Tabel 1.

**Tabel 1.** Hasil Normalisasi Data

| No  | Data | Hasil Normalisasi |
|-----|------|-------------------|
| 1   | 6.26 | 0.456561922       |
| 2   | 6.30 | 0.46025878        |
| 3   | 6.52 | 0.480591497       |
| ... | ...  | ...               |
| 208 | 3.00 | 0.155268022       |
| 209 | 2.84 | 0.140480591       |
| 210 | 2.51 | 0.109981516       |

##### Proses Prediksi

*Long Short Term Memory* (LSTM) memiliki tiga gerbang utama yang memproses data yaitu *forget gate*, *input gate* dan *output gate*. Pada keadaan awal *hidden state* dan *cell state* diatur

menjadi nol. Bobot dan bias pada setiap gerbang diinisialisasi dengan nilai acak menggunakan Python untuk memulai proses pelatihan model.

Menghitung *forget gate*:

$$\begin{aligned} f_1 &= \sigma(W_f \cdot [h_{t-1}, x'_t] + b_f) \\ &= \sigma(0.2961402 * 0.456561922 + 0.8137982 * 0 + 0.1494483) \\ &= \frac{1}{1+e^{-(0.28465439)}} = 0.570686999 \end{aligned}$$

Menghitung *input gate*:

$$\begin{aligned} i_1 &= \sigma(W_i \cdot [h_{t-1}, x'_t] + b_i) \\ &= \sigma(0.5488135 * 0.456561922 + 0.61209572 * 0 + 0.97676109) \\ &= \frac{1}{1+e^{-(1.227328437)}} = 0.773350646 \end{aligned}$$

Menghitung kandidat *cell state*:

$$\begin{aligned} c'_1 &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ &= \tanh(0.80731896 * 0.456561922 + 0.18619301 * 0 + 0.94737059) \\ &= 2\sigma(2 * 1.315961686) - 1 \\ &= \frac{2}{1+e^{-(2*1.315961686)}} - 1 = 0.865776136 \end{aligned}$$

Menghitung *Cell State*:

$$\begin{aligned} c_1 &= f_t * c_{t-1} + i_t * c'_t \\ &= 0.570686999 * 0 + 0.773350646 * 0.865776136 = 0.669548534 \end{aligned}$$

Menghitung *Output gate*:

$$\begin{aligned} o_1 &= \sigma(W_o[h_{t-1}, x'_t] + b_o) \\ &= \sigma(0.20747008 * 0.456561922 + 0.97749514 * 0 + 0.13248763) \\ &= \frac{1}{1+e^{-(0.227210569)}} = 0.55655953 \end{aligned}$$

Menghitung *Hidden State*:

$$\begin{aligned} h_t &= \tanh(c_t) * o_t \\ &= (2\sigma(0.669548534) - 1) * 0.55655953 \\ &= \left( \frac{2}{1+e^{-(2*0.669548534)}} - 1 \right) * 0.55655953 = 0.325410801 \end{aligned}$$

Proses perhitungan diatas akan berlangsung untuk setiap *time step* dengan menggunakan *output* dari *time step* sebelumnya sebagai input untuk *time step* selanjutnya. Hasil proses perhitungan yang telah dilakukan akan diproses pada lapisan linear dengan menggunakan  $h_t$  terakhir atau  $h_{12}$  yaitu 0.698798619. Nilai bobot dan bias diambil secara acak.

$$\hat{y} = x * a^T + b = 0.698798619 * (-0.312) + 0.3464 = 0.1283748$$

Setelah mendapatkan hasil *forward propagation* maka selanjutnya mencari nilai *loss* menggunakan persamaan *loss function* dengan data ke tiga belas hasil normalisasi sebagai data target yaitu 0.5582255.

$$MAE = \frac{1}{n} \sum_{L=1}^n |\hat{y} - y_L| = \frac{1}{n} \sum_{L=1}^n |0.1283748 - 0.5582255| = 0.4298507$$

Setelah nilai *loss* didapatkan, kemudian dilakukan proses *backpropagation* pada perhitungan nilai *loss*.

$$d_{\hat{y}_L} = 1 \text{ if } \hat{y}_L > y_L \text{ else } -1 = 0.1283748 < 0.5582255 = -1$$

Hasil dari gradien nilai *loss MAE* akan digunakan untuk mencari nilai gradien terhadap bobot dan bias pada lapisan linear. Berikut proses mencari nilai gradien bobot lapisan linear.

$$\frac{d_{loss}}{d_{wLi}} = x_{Li} * d_{\hat{y}_L} = 0.698798619 * -1 = -0.698798619$$

Proses perhitungan gradien bias pada lapisan linear

$$\frac{d_{loss}}{d_{bLi}} = 1 * d_{\hat{y}_L} = 1 * -1 = -1$$

Nilai bobot baru dari lapisan linear dicari dengan menggunakan *learning rate* 0.01

$$\text{Bobot baru} = \text{Bobot lama} - (\alpha * \text{Gradien})$$

$$\text{Bobot baru} = -0.312 - (0.01 * -0.698798619) = -0.3050120$$

Maka didapatkan nilai bobot baru untuk lapisan linear adalah  $-0.3106029$ , kemudian akan dicari nilai bias baru.

$$\text{Bias baru} = \text{Bias lama} - (\alpha * \text{Gradien})$$

$$\text{Bias baru} = 0.3464 - (0.01 * -1) = 0.3564$$

Proses selanjutnya adalah *backpropagation* pada lapisan LSTM yaitu turunan parsial terhadap  $o_t$  dan  $c_t$ .

$$\frac{d_{loss}}{d_{Hot}} = W_{Li} * \tanh(c_t) = -0.3050120 * \tanh(2.25704) = -0.298402$$

$$\frac{d_{loss}}{d_{Hct}} = W_{Li} * o_t * \text{sech}^2(c_t) = -0.3050120 * 0.97833 * \text{sech}^2(2.25704) = -0.0127926$$

*Backpropagation* terhadap proses memperbarui *cell state* dilakukan dengan turunan parsial terhadap  $f_t$ ,  $i_t$  dan  $g_t$ . Hasil dari proses tersebut akan digunakan untuk mencari gradien setiap gerbang. Proses ini menggunakan metode *chain rule* dari  $\frac{d_{loss}}{d_{Hct}}$ .

Selanjutnya akan dicari gradien setiap gerbang dengan menggunakan hasil dari proses *backpropagation* terhadap proses memperbarui *cell state*. Proses perhitungan gradien terhadap  $i_t$  dan  $c'_t$  adalah sebagai berikut:

$$\frac{d_{loss}}{d_{it}} = \frac{d_{loss}}{d_{Cit}} * i_t * (1 - i_t) = -0.01040234 * 0.841785 * (1 - 0.841785) = -0.0013854$$

$$\frac{d_{loss}}{d_{c't}} = \frac{d_{loss}}{d_{c't}} * \text{sech}^2(c'_t) = -0.01076862 * \text{sech}^2(0.813153) = -0.00591543$$

Gradien *forget gate* dan *output gate* dicari dengan cara yang sama dengan perhitungan mencari gradien *input gate* sehingga gradien setiap gerbang didapatkan. Hasil dari *backpropagation* dari setiap *gate* disajikan dalam tabel sebagai berikut:

**Tabel 2.** Hasil Gabungan Gradien

| Gate                       | Nilai       |
|----------------------------|-------------|
| <i>Input gate</i>          | -0.0013854  |
| <i>Forget gate</i>         | -0.005972   |
| Kandidal <i>cell state</i> | -0.00591543 |
| <i>Output gate</i>         | -0.060899   |

Setelah didapatkan seluruh gradien bias setiap *gate* maka akan dicari gradien setiap bobot.

**Tabel 3.** Nilai masukan LSTM

| $x_t$       | $h_t$       |
|-------------|-------------|
| 0.487060998 | 0.487060998 |

Tabel 3 merupakan nilai masukan pada lapisan LSTM. Hasil dari proses mengkalikan matriks tabel 2 dengan 3 merupakan nilai gradien setiap bobot pada LSTM *gate*.

**Tabel 4.** Gradien Bobot

| Gate                       | $W_x$     | $W_h$     |
|----------------------------|-----------|-----------|
| <i>Input gate</i>          | -0.000675 | -0.000675 |
| <i>Forget gate</i>         | -0.002909 | -0.002909 |
| Kandidat <i>Cell state</i> | -0.002881 | -0.002881 |
| <i>Output gate</i>         | -0.029662 | -0.029662 |

Hasil dari Tabel 4 akan digunakan untuk memperbarui bobot setiap gerbang pada LSTM.

**Tabel 5.** Bobot Lama

| Gate  | $I_t$     | $f_t$    | $c'_t$   | $o_t$     |
|-------|-----------|----------|----------|-----------|
| $W_x$ | 0.5488135 | 0.29614  | 0.807319 | 0.2074701 |
| $W_h$ | 0.6120957 | 0.813798 | 0.186193 | 0.9774951 |

Setiap bobot akan diperbarui dengan rumus *Stochastic Gradient Descent* (SGD) dengan gradien bobot pada Tabel 4 menggunakan python.

**Tabel 6.** Bobot Baru

| Gate  | $I_t$       | $f_t$      | $c'_t$     | $o_t$      |
|-------|-------------|------------|------------|------------|
| $W_x$ | 0.544882025 | 0.29616909 | 0.80734781 | 0.20776672 |
| $W_h$ | 0.61210245  | 0.81382709 | 0.18164811 | 0.97779172 |

### Penentuan Parameter Pengujian

Proses sebelum melatih model adalah menentukan parameter-parameter pendukung yang dibutuhkan untuk membuat model LSTM.

Tabel 7. Parameter Pengujian

| Parameter      | Jumlah      | Keterangan          |
|----------------|-------------|---------------------|
| Komposisi Data | Trial Error | 210                 |
| Neuron         | Trial Error | 5,10,15,20,25,30    |
| Batch Size     | Trial Error | 32, 64, 128         |
| Epoch          | Trial Error | 50, 100, 150, 200   |
| Learning Rate  | Trial Error | 0.1, 0.01 dan 0.001 |
| Optimizer      | 1           | ADAM                |

Parameter-parameter tersebut akan diuji, setiap parameter yang menghasilkan *loss* yang cukup baik akan digunakan untuk pengujian tahap selanjutnya.

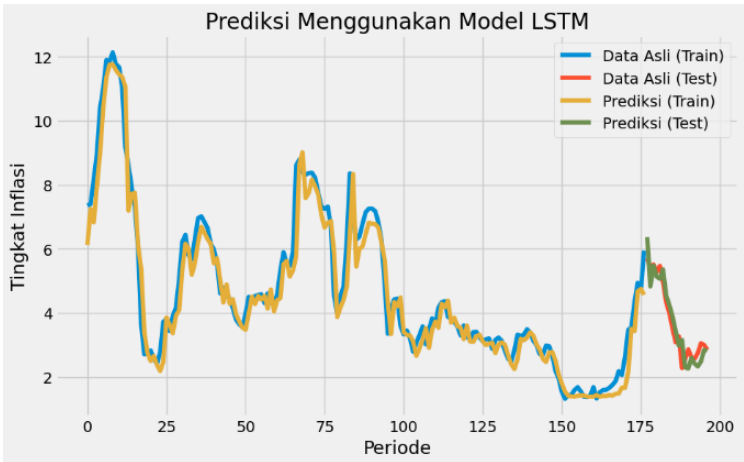
Pengujian Parameter

Parameter yang akan di uji adalah komposisi data, *hidden layer*, jumlah *neuron*, *batch size*, *epoch* dan *learning rate* dengan menggunakan ADAM *optimizer*. Jika nilai kerugian pelatihan (*Loss*) lebih kecil dari validasi (*validation loss*) tergolong *overfitting* jika sebaliknya disebut *underfitting*. Model terbaik diambil dari selisih nilai RMSE yang terkecil. Dari hasil pengujian didapatkan *hyperparameter* optimal untuk model LSTM terdapat pada tabel .

Tabel 8. Parameter Optimal

| Parameter      | Jumlah |
|----------------|--------|
| Komposisi Data | 90:10  |
| Hidden Layer   | 1      |
| Neuron         | 5      |
| Batch Size     | 32     |
| Epoch          | 200    |
| Learning Rate  | 0.1    |
| Optimizer      | ADAM   |

Proses prediksi dilakukan dengan menggunakan *hyperparameter* dari model terbaik yang dipilih setelah pengujian menggunakan metode *Long Short Term Memory* sesuai dengan tabel 8. Gambar 3 merupakan grafik perbandingan data aktual dan hasil prediksi.



Gambar 3. Grafik Perbandingan Data Aktual dan Data Prediksi



Gambar 3 menunjukan secara garis besar pola hasil prediksi mendekati pola dari data aktual yang menunjukan model LSTM telah dilatih dengan baik dan mampu memprediksi dengan tingkat akurasi yang baik.

### Evaluasi Model

Kesalahan peramalan dengan metode *Long Short Term Memory* dihitung menggunakan MAPE. Hasil MAPE untuk data *training* yaitu 9.2533% tergolong sangat baik dan MAPE untuk data *testing* yaitu 9.9152% tergolong sangat baik sehingga prediksi berjalan dengan baik.

### Prediksi 6 Bulan Ke Depan

Prediksi untuk 6 bulan ke depan menggunakan *hyperparameter* dari model terbaik, Hasil dari prediksi disajikan dalam bentuk tabel sebagai berikut:

**Tabel 9.** Hasil Prediksi Inflasi

| Bulan          | Hasil Prediksi |
|----------------|----------------|
| Juli 2024      | 2.828988       |
| Agustus 2024   | 2.664783       |
| September 2024 | 2.4757438      |
| Oktober 2024   | 2.272163       |
| November 2024  | 2.0550873      |
| Desember 2024  | 1.8432692      |

Dari bulan Juli 2024 hingga Desember 2024, ada penurunan inflasi yang konsisten. Inflasi menurun dari 2.828988 di bulan Juli menjadi 1.8432692 di bulan Desember. Ini menunjukkan bahwa model memprediksi penurunan inflasi selama periode ini.

### D. Kesimpulan

Berdasarkan hasil evaluasi, dapat disimpulkan bahwa model *Long Short Term Memory* terbaik yang diperoleh adalah model LSTM dibangun dengan arsitektur satu hidden layer dengan kombinasi *hyperparameter* 5 *neuron*, 32 *batch size* dan 0.1 *learning rate* yang dilatih selama 200 *epoch* dengan komposisi 90% data *training* dan 10% data *testing* menggunakan optimasi ADAM. Hasil MAPE untuk data *training* adalah 9.253% dan data *testing* 9.915%. Hasil prediksi inflasi untuk bulan selanjutnya yaitu bulan juli 2024 adalah 2.8282988%.

### Acknowledge

Penelitian ini didukung oleh Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Bandung. Terima kasih disampaikan kepada semua pihak yang telah berkontribusi dalam penyelesaian penelitian ini.

### Daftar Pustaka

- [1] R. A. Chintia and R. Destiningsih, "Pengaruh Harga Komoditas Pangan Terhadap Inflasi Di Kota Semarang," *Jurnal Ilmiah Ekonomi Bisnis*, vol. 27, no. 2, pp. 244–258, 2022, doi: 10.35760/eb.2022.v27i2.4948.
- [2] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput Ind Eng*, vol. 143, May 2020, doi: 10.1016/j.cie.2020.106435.
- [3] L. Setiawan, D. Susanti, and R. Riaman, "Analisis Perbandingan Hasil Peramalan Harga Saham Menggunakan Model Autoregressive Integrated Moving Average dan Long Short Term Memory," *Jurnal Matematika Integratif*, vol. 19, no. 2, p. 223, Dec. 2023, doi: 10.24198/jmi.v19.n2.42164.223-234.
- [4] K. T. Brilliansyah and U. T. Kartini, "Peralaman Jangka Sangat Pendek Daya Listrik PLTS On Grid Rumah Tinggal Menggunakan Metode Recurrent Neural Network Long Short Term Memory Berdasarkan Data Meteorologi Peramalan Jangka Sangat Pendek

- Daya Listrik PLTS On Grid Rumah Tinggal Menggunakan Metode Recurrent Neural Network Long Short Term Memory Berdasarkan Data Meteorologi.”
- [5] C. A. Maharani, B. Warsito, and R. Santoso, “Analisis Sentimen Vaksin Covid-19 Pada Twitter Menggunakan Recurrent Neural Network (RNN) Dengan Algoritma Long Short-Term Memory (LSTM),” *Jurnal Gaussian*, vol. 12, no. 3, pp. 403–413, Feb. 2024, doi: 10.14710/j.gauss.12.3.403-413.
  - [6] B. Arief, H. Kholifatullah, and A. Prihanto, “Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech,” *Journal of Informatics and Computer Science*, vol. 04, 2023.
  - [7] S. Hochreiter and J. “ Urgen Schmidhuber, “Long Short-Term Memory.”
  - [8] J. Qiu, B. Wang, and C. Zhou, “Forecasting stock prices with long-short term memory neural network based on attention mechanism,” *PLoS One*, vol. 15, no. 1, Jan. 2020, doi: 10.1371/journal.pone.0227222.
  - [9] E. Budhiarti Nababan and M. Zarlis, “Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma Backpropagation Pada Prediksi Kemampuan Siswa,” 2015.
  - [10] E. Rosita et al., “Pemodelan Jaringan Syaraf Tiruan Dengan Cascade Forward Backpropagation Pada Kurs Rupiah Terhadap Dolar Amerika Serikat,” vol. 7, no. 1, pp. 64–72, 2018, [Online]. Available: <https://ejournal3.undip.ac.id/index.php/gaussian/>