



Implementasi Model Long Short Term Memory (LSTM) dalam Prediksi Harga Saham

Juliandi Kurniansyah, Siska Kurnia Gusti*, Febi Yanto, Muhammad Affandes

Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim, Pekanbaru, Indonesia

Email: ¹juliandikurniansyah@gmail.com, ^{2*}siskakurniagusti@uin-suska.ac.id, ³febiyanto@uin-suska.ac.id, ⁴affandes@uin-suska.ac.id

Abstrak- Investasi di pasar saham semakin diminati meskipun fluktuasi harga saham sering kali sulit diprediksi. Model prediksi saham yang akurat dapat membantu investor dalam pengambilan keputusan. Penelitian ini menerapkan model Long Short-Term Memory (LSTM) untuk melakukan peramalan harga saham BBKA berdasarkan data historis harian dari 1 Januari 2015 hingga 1 Januari 2025. Data diambil dari Yahoo Finance dan hanya menggunakan variabel penutupan (*close*). Proses penelitian diantaranya pra-pemrosesan data, normalisasi Min-Max, pemodelan LSTM dengan variasi *timesteps* (30, 60, 90 hari), serta evaluasi hasil prediksi. Model LSTM dibangun dengan dua lapisan LSTM, lapisan *dropout*, dan lapisan *dense*, serta dilatih menggunakan fungsi *loss mean_squared_error* dan *optimizer* Adam. Hasil evaluasi menunjukkan bahwa konfigurasi model dengan *timesteps* 60 hari mencapai performa optimal dengan nilai RMSE 114.17, MAPE dengan persentase 0.96%, dan R-Squared 0.98, yang mengindikasikan prediksi model sangat akurat dan andal. Penelitian ini menunjukkan bahwa LSTM merupakan model yang efektif untuk prediksi harga saham berdasarkan data deret waktu.

Kata Kunci: BBKA; LSTM; Prediksi; Prediksi Saham; Saham

Abstract- Stock market investment is gaining popularity, although predicting stock price fluctuations remains challenging. Accurate stock prediction models can assist investors in decision-making. In this research, a Long Short-Term Memory (LSTM) model was employed to make predictions regarding the stock prices of BBKA based on daily historical data from January 1 2015 to January 1 2025. The data was gathered from the Yahoo Finance website, utilizing only the closing price (*'close'*) variable. The research process included data pre-processing, Min-Max normalization, LSTM modeling with varying *timesteps* (30, 60, 90 days), and evaluation of prediction results. The LSTM model was built with two LSTM layers, a dropout layer, and a final dense layer, and its training involved the application of the *mean_squared_error* loss function and Adam optimizer. Evaluation results showed that the model configuration with 60 *timesteps* achieved optimal performance with a RMSE of 114.17, MAPE percentage of 0.96%, and an R-Squared of 0.98, indicating highly accurate and reliable predictions. This study demonstrated that LSTM is an effective model for stock price prediction based on time series data.

Keywords: BBKA; LSTM; Prediction; Stock; Stock Prediction

1. PENDAHULUAN

Saham merupakan instrumen finansial yang menandakan porsi kepemilikan dalam sebuah perusahaan. Emiten menggunakan penjualan saham sebagai sarana untuk menambah permodalan dari investor. Para investor sendiri berpotensi mendapatkan profit apabila harga jual saham mereka melebihi harga saat pembelian [1]. Pada era saat ini, investasi di pasar saham sangat sulit untuk mengambil keputusan untuk para investor, sulitnya mengidentifikasi saham yang tepat pada timing yang akurat mendorong kebutuhan akan mesin prediksi yang berkualitas [2]. Fluktuasi indeks harga saham yang terjadi dalam periode singkat seringkali menjadi penyebab keraguan bagi calon investor untuk mulai berinvestasi [3]. Dari awal prediksi atau peramalan harga saham (*stock price forecasting*) adalah tantangan terbesar yang dihadapi oleh saham analis. Oleh karena itu, topik ini telah menarik perhatian bisnis dan komunitas akademis [4].

Dengan meningkatnya jumlah investor pasar modal, *stock price forecasting* sangat diminati, meskipun *efficient market hypothesis* (hipotesis pasar efisien) percaya bahwa fluktuasi harga saham tidak mungkin diprediksi. Namun beberapa peneliti berpendapat bahwa beberapa model dapat diterima selama model tersebut dapat memberikan akurasi yang tinggi [5]. Di antara berbagai teknik prediksi, *Long Short-Term Memory* (LSTM) dikenal sebagai metodologi yang menunjukkan efektivitas tinggi dan kapabilitas yang mumpuni [6]. Diklasifikasikan sebagai salah satu jenis RNN, *Long Short-Term Memory* (LSTM) membedakan dirinya melalui penambahan unit memori (*memory cell*) pada strukturnya. Komponen ini secara spesifik memungkinkan LSTM untuk menyimpan dan mengakses informasi historis dalam durasi yang lebih ekstensif dibandingkan RNN konvensional [7].

Penelitian sebelumnya [8] membandingkan dua model prediksi yaitu ARIMA dan LSTM dalam memprediksi harga saham Apple. Hasilnya memperlihatkan bahwa model LSTM secara signifikan lebih unggul daripada ARIMA dengan nilai evaluasi *Root Mean Square Error* (RMSE) sebesar 0.237, Hal ini menunjukkan akurasi prediksi LSTM yang lebih baik dibandingkan ARIMA.

Penelitian lainnya yang telah dilakukan berjudul “Peramalan harga saham sektor pertambangan di Indonesia”. Menurut [9] penelitian ini menggunakan LSTM dan menghasilkan nilai *Root Mean Square Error* (RMSE) terkecil sebesar 31,76 dengan variasi *epochs* 200 pada emiten TINS. Namun, penelitian ini hanya mengevaluasi *Root Mean Square Error* (RMSE), sehingga kurang memberikan gambaran yang lebih detil mengenai performa model dalam aspek lainnya.

Berdasarkan latar belakang yang dipaparkan diatas, penelitian ini menggunakan data harga penutupan harga saham Bank Central Asia (BBKA). Sebab saham Bank Central Asia (BBKA) memiliki reputasi tinggi, selain memiliki ukuran kapitalisasi pasar yang besar, perusahaan tersebut juga menunjukkan catatan kinerja yang baik dan secara rutin memberikan imbal hasil berupa dividen kepada investornya [10]. Untuk memperoleh nilai kesalahan prediksi yang paling

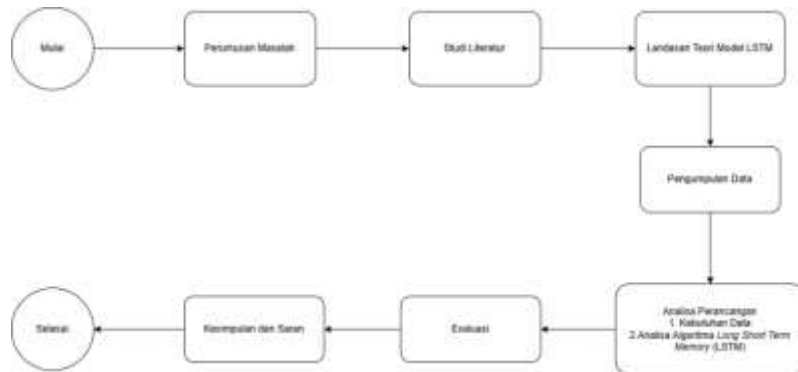


kecil, Kinerja model *Long Short-Term Memory* (LSTM) ini diukur berdasarkan *Root Mean Square Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), dan *R-Squared* (R^2) sebagai dasar evaluasinya.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Fokus dari penelitian ini ada pada data saham yang didapat melalui website Yahoo Finance. Data tersebut didapatkan melalui link berikut: <https://finance.yahoo.com>. Penelitian dimulai dengan merumuskan masalah lalu menelusuri penelitian terdahulu yang berkaitan, pengumpulan data dan model LSTM yang dibangun. Berikut gambaran dari tahapan ini:



Gambar 1. Tahapan Penelitian Implementasi model LSTM

2.2 Model LSTM

Gagasan awal untuk model LSTM, yang merupakan suatu bentuk evolusi dari struktur *Recurrent Neural Network* (RNN), diperkenalkan pertama kali oleh dua orang, yaitu Hochreiter & Schmidhuber pada tahun 1997 [11]. Keunggulan LSTM dibandingkan RNN biasa adalah kapabilitasnya dalam mempelajari serta memutuskan informasi yang perlu dipertahankan atau dihilangkan dari dalam unit memorinya [12]. Penggunaan metode *Long Short-Term Memory* (LSTM) sangat relevan untuk prediksi data bersifat *time series* sebab kemampuannya dalam mengingat serta menyimpan data historis, baik untuk rentang waktu pendek maupun panjang [13].

LSTM memiliki mekanisme gerbang (*gates*) yang berfungsi untuk mengatur penghapusan dan pembaruan informasi di dalam *cell state*. Pembaruan informasi di dalam sel memori ini dilakukan melalui tiga jenis gerbang, yaitu *input gate*, *forget gate*, dan *output gate* [14]. Setiap gerbang menggunakan aktivasi Sigmoid untuk menentukan apakah gerbang tersebut bernilai aktif atau tidak (0/1). *Forget gate* digunakan untuk menentukan apakah data yang datang perlu disimpan atau sebaliknya, *input gate* digunakan untuk menentukan apakah data perlu diperbarui atau tidaknya, dan *output gate* berfungsi untuk menentukan output prediksi dari model [15].

2.3 Pengumpulan Data

Tujuan utama dari tahap pengumpulan data adalah untuk menghimpun semua data relevan yang diperlukan dalam pelaksanaan studi ini. Dalam studi ini, jenis data yang digunakan adalah data sekunder. *Dataset* yang diambil adalah data saham historis harian perusahaan Bank Central Asia (BBCA). Pada penelitian ini data diperoleh di <https://finance.yahoo.com> mengambil waktu periode 1 Januari 2015 hingga 1 Januari 2025 yang berjumlah sebanyak 2475 data.

2.4 Analisa Perancangan

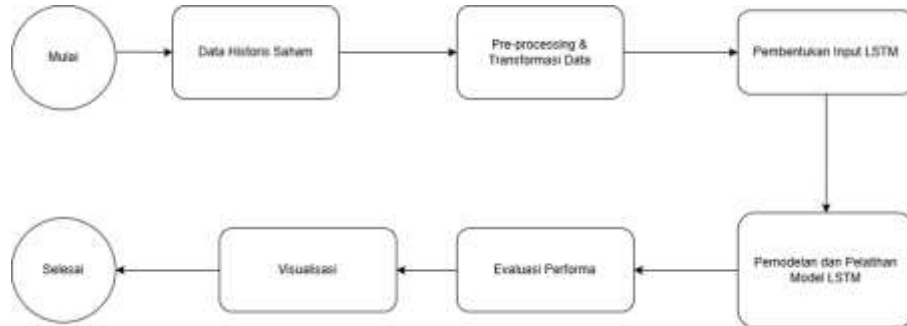
Analisa perancangan yang dilakukan meliputi dua langkah pokok, yaitu analisis kebutuhan data dan analisis model *Long Short Term Memory* (LSTM). Tahapannya dijelaskan sebagai berikut:

1. Analisa Kebutuhan Data

Pada tahapan ini dilakukan analisa terhadap kebutuhan data pada penelitian. Hal pertama adalah menentukan variabel data input berdasarkan data yang diperoleh dari <https://finance.yahoo.com/> dengan variabel *input* sebanyak 1 *inputan*, yaitu berupa data *close*. *Close* merupakan data harga yang menjadi penutup pada hari saham tersebut diperdagangkan. Berikutnya membagi *dataset* menjadi data latih dan uji dengan perbandingan 80:20.

2. Analisa Model LSTM

Proses pertama dalam merancang model LSTM adalah menormalisasi data historis saham dan melakukan transformasi. Pembentukan *inputan* LSTM juga dilakukan pada tahap ini. Setelah data siap maka model dibangun dan dilatih untuk memprediksi dan performanya dievaluasi. Setelah proses evaluasi dilanjutkan dengan pengujian prediksi masa depan. Berikut merupakan gambar yang menjelaskan tahapan pembuatan model LSTM:



Gambar 2. Tahapan Pembuatan Model LSTM

2.5 Evaluasi

Selanjutnya, dilakukan evaluasi terhadap hasil penelitian yang bertujuan untuk dapat menentukan model prediksi terbaik dengan nilai error terkecil. Pengujian ini dilakukan dengan tiga metrik, meliputi:

1. *Root Mean Square Error* (RMSE)
RMSE berfungsi untuk menilai besaran deviasi atau perbedaan antara keluaran prediktif suatu model dan data observasional yang sebenarnya. Semakin kecil nilainya maka model menunjukkan tingkat prediksi yang bagus [16].
2. *Mean Absolute Percentage Error* (MAPE)
MAPE berfungsi untuk menentukan ketepatan prediksi atau estimasi sebuah model, di mana persentase yang lebih rendah menunjukkan bahwa model tersebut lebih akurat [17].
3. *R-Squared* (R^2)
 R^2 adalah indikator yang merefleksikan seberapa signifikan variabel independen (eksogen) mampu mempengaruhi variabel dependen. Rentang nilai untuk metrik ini adalah dari 0 hingga 1, di mana skor yang mendekati 1 mengindikasikan kinerja model yang unggul [18].

3. HASIL DAN PEMBAHASAN

3.1 Data Preparation

Guna memenuhi kebutuhan analisis dalam riset ini, data historis harian terkait harga saham PT Bank Central Asia dikumpulkan melalui metode *web scraping*. Teknik ini diaplikasikan untuk mengambil data dari sumber-sumber publik, di mana secara spesifik data untuk studi ini diperoleh dari situs Yahoo Finance. Proses akuisisi data melibatkan pencarian menggunakan kata kunci 'BBCA.JK' dan selanjutnya dilakukan *filter* data untuk periode antara 1 Januari 2015 dan 1 Januari 2025. Data yang diperoleh memiliki enam variabel dan disimpan berbentuk *file* .csv dan berikut adalah contoh tampilan *dataset* pada Tabel 1.

Tabel 1. Dataset Harga Saham BBKA

Date	Adj. Close	Close	High	Low	Open	Volume
2015/01/04	4.606,88	5.200	5.200	5.200	5.200	0
2015/01/05	4.589,31	5.240	5.245	5.200	5.200	35.956.000
...	...	...	...	...	...	...
2024/12/27	9.142,67	9.400	9.425	9.325	9.325	70.184.600
2024/12/30	9.142,67	9.400	9.450	9.350	9.425	79.903.400

3.2 Pre-processing dan Transformasi Data

Dalam konteks pengolahan data, pra-pemrosesan adalah proses koreksi dan perbaikan data yang fokus pada menghilangkan elemen perusak seperti *noise* dan mengelola data yang tidak lengkap atau hilang (*missing values*). Metode yang digunakan untuk menangani *missing value* ini adalah menggunakan metode *Forward Fill*.

Proses transformasi data melibatkan modifikasi dan penyesuaian data agar selaras dengan representasi format yang diperlukan proses ini dapat juga disebut dengan normalisasi data. Normalisasi data merupakan suatu prosedur untuk mentransformasi nilai asli data ke dalam sebuah skala atau rentang nilai yang spesifik. *Min-Max Normalization* adalah salah satu metode yang umum dimanfaatkan untuk melakukan proses tersebut [11]. Sebagai salah satu metode normalisasi yang sederhana, *Min-Max Normalization* menerapkan transformasi linear pada data asli dengan tujuan agar nilai data sebelum dan sesudah pemrosesan menjadi lebih seimbang [19]. Hal ini juga mencakup pembagian *dataset* menjadi data latih dan data uji sebanyak 80:20, serta mengubah bentuk datanya menjadi *Array*. Proses ini dilakukan dengan kode pada Gambar 3 sebagai berikut:

```
stock_data.ffill(inplace=True)
close_prices = stock_data['Close']
values = close_prices.values
training_data_len = math.ceil(len(values)* 0.8)

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(values.reshape(-1,1))

train_data = scaled_data[0: training_data_len, :]

x_train = []
y_train = []

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
```

Gambar 3. Pre-processing dan Transformasi Data

3.3 Pembentukan Input Model LSTM

Setelah data dinormalisasi dan dibagi menjadi data latih dan data uji, tahap selanjutnya adalah membentuk *input* dalam format yang sesuai dengan kebutuhan arsitektur model LSTM. Model *Long Short Term Memory* (LSTM) yang dirancang khusus memproses data *time series* dan memerlukan struktur data masukan berbentuk tiga dimensi, yaitu: jumlah sampel (*samples*), jumlah langkah waktu (*timesteps*), dan jumlah fitur (*features*).

Untuk mendukung kemampuan model dalam mempelajari pola historis harga saham, digunakan pendekatan *sliding window* dengan ukuran 30, 60, dan 90. Artinya, model akan menggunakan data dari 30, 60, dan 90 hari sebelumnya untuk memprediksi harga pada hari berikutnya. Kode pada tahapan ini dapat dilihat pada Gambar 4 sebagai berikut:

```
test_data = scaled_data[training_data_len-60: , : ]
x_test = []
y_test = values[training_data_len:]

for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

Gambar 4. Pembentukan Input Model LSTM

3.4 Pembangunan dan Pelatihan Model LSTM

Pada tahap ini dilakukan pembuatan model, pelatihan dan prediksi terhadap harga saham menggunakan *dataset* yang sudah disiapkan prosesnya sebagai berikut:

1. Definisi Model

Model *Long Short-Term Memory* (LSTM) dibangun menggunakan struktur sekuensial yang terdiri dari dua *layer* LSTM, setiap *layer* diisi dengan 100 unit neuron. Sebuah *Dropout layer* bernilai laju 0.02 diintegrasikan untuk regularisasi. Kemudian, terdapat dua *dense layer*; *layer* yang pertama memiliki 25 neuron, dan *layer* akhirnya berisi satu neuron yang berfungsi untuk menyajikan satu nilai *output*. Dengan konfigurasi tersebut, arsitektur ini secara

spesifik dikembangkan guna mengidentifikasi serta merepresentasikan pola-pola temporal yang terkandung dalam data deret waktu harga saham. Adapun kodenya terlihat pada Gambar 5.

```
1 model = keras.Sequential()
2 model.add(layers.LSTM(100, return_sequences=True, input_shape=(x_train.shape[1], 1)))
3 model.add(layers.LSTM(100, return_sequences=False))
4 model.add(layers.LSTM(100, return_sequences=False))
5 model.add(layers.LSTM(100, return_sequences=False))
6 model.add(layers.LSTM(100, return_sequences=False))
7 model.add(layers.LSTM(100, return_sequences=False))
8 model.add(layers.LSTM(100, return_sequences=False))
9 model.add(layers.LSTM(100, return_sequences=False))
10 model.add(layers.LSTM(100, return_sequences=False))
11 model.add(layers.LSTM(100, return_sequences=False))
12 model.add(layers.LSTM(100, return_sequences=False))
13 model.add(layers.LSTM(100, return_sequences=False))
14 model.add(layers.LSTM(100, return_sequences=False))
15 model.add(layers.LSTM(100, return_sequences=False))
16 model.add(layers.LSTM(100, return_sequences=False))
17 model.add(layers.LSTM(100, return_sequences=False))
18 model.add(layers.LSTM(100, return_sequences=False))
19 model.add(layers.LSTM(100, return_sequences=False))
20 model.add(layers.LSTM(100, return_sequences=False))
21 model.add(layers.LSTM(100, return_sequences=False))
22 model.add(layers.LSTM(100, return_sequences=False))
23 model.add(layers.LSTM(100, return_sequences=False))
24 model.add(layers.LSTM(100, return_sequences=False))
25 model.add(layers.LSTM(100, return_sequences=False))
26 model.add(layers.LSTM(100, return_sequences=False))
27 model.add(layers.LSTM(100, return_sequences=False))
28 model.add(layers.LSTM(100, return_sequences=False))
29 model.add(layers.LSTM(100, return_sequences=False))
30 model.add(layers.LSTM(100, return_sequences=False))
31 model.add(layers.LSTM(100, return_sequences=False))
32 model.add(layers.LSTM(100, return_sequences=False))
33 model.add(layers.LSTM(100, return_sequences=False))
34 model.add(layers.LSTM(100, return_sequences=False))
35 model.add(layers.LSTM(100, return_sequences=False))
36 model.add(layers.LSTM(100, return_sequences=False))
37 model.add(layers.LSTM(100, return_sequences=False))
38 model.add(layers.LSTM(100, return_sequences=False))
39 model.add(layers.LSTM(100, return_sequences=False))
40 model.add(layers.LSTM(100, return_sequences=False))
41 model.add(layers.LSTM(100, return_sequences=False))
42 model.add(layers.LSTM(100, return_sequences=False))
43 model.add(layers.LSTM(100, return_sequences=False))
44 model.add(layers.LSTM(100, return_sequences=False))
45 model.add(layers.LSTM(100, return_sequences=False))
46 model.add(layers.LSTM(100, return_sequences=False))
47 model.add(layers.LSTM(100, return_sequences=False))
48 model.add(layers.LSTM(100, return_sequences=False))
49 model.add(layers.LSTM(100, return_sequences=False))
50 model.add(layers.LSTM(100, return_sequences=False))
51 model.add(layers.LSTM(100, return_sequences=False))
52 model.add(layers.LSTM(100, return_sequences=False))
53 model.add(layers.LSTM(100, return_sequences=False))
54 model.add(layers.LSTM(100, return_sequences=False))
55 model.add(layers.LSTM(100, return_sequences=False))
56 model.add(layers.LSTM(100, return_sequences=False))
57 model.add(layers.LSTM(100, return_sequences=False))
58 model.add(layers.LSTM(100, return_sequences=False))
59 model.add(layers.LSTM(100, return_sequences=False))
60 model.add(layers.LSTM(100, return_sequences=False))
61 model.add(layers.LSTM(100, return_sequences=False))
62 model.add(layers.LSTM(100, return_sequences=False))
63 model.add(layers.LSTM(100, return_sequences=False))
64 model.add(layers.LSTM(100, return_sequences=False))
65 model.add(layers.LSTM(100, return_sequences=False))
66 model.add(layers.LSTM(100, return_sequences=False))
67 model.add(layers.LSTM(100, return_sequences=False))
68 model.add(layers.LSTM(100, return_sequences=False))
69 model.add(layers.LSTM(100, return_sequences=False))
70 model.add(layers.LSTM(100, return_sequences=False))
71 model.add(layers.LSTM(100, return_sequences=False))
72 model.add(layers.LSTM(100, return_sequences=False))
73 model.add(layers.LSTM(100, return_sequences=False))
74 model.add(layers.LSTM(100, return_sequences=False))
75 model.add(layers.LSTM(100, return_sequences=False))
76 model.add(layers.LSTM(100, return_sequences=False))
77 model.add(layers.LSTM(100, return_sequences=False))
78 model.add(layers.LSTM(100, return_sequences=False))
79 model.add(layers.LSTM(100, return_sequences=False))
80 model.add(layers.LSTM(100, return_sequences=False))
81 model.add(layers.LSTM(100, return_sequences=False))
82 model.add(layers.LSTM(100, return_sequences=False))
83 model.add(layers.LSTM(100, return_sequences=False))
84 model.add(layers.LSTM(100, return_sequences=False))
85 model.add(layers.LSTM(100, return_sequences=False))
86 model.add(layers.LSTM(100, return_sequences=False))
87 model.add(layers.LSTM(100, return_sequences=False))
88 model.add(layers.LSTM(100, return_sequences=False))
89 model.add(layers.LSTM(100, return_sequences=False))
90 model.add(layers.LSTM(100, return_sequences=False))
91 model.add(layers.LSTM(100, return_sequences=False))
92 model.add(layers.LSTM(100, return_sequences=False))
93 model.add(layers.LSTM(100, return_sequences=False))
94 model.add(layers.LSTM(100, return_sequences=False))
95 model.add(layers.LSTM(100, return_sequences=False))
96 model.add(layers.LSTM(100, return_sequences=False))
97 model.add(layers.LSTM(100, return_sequences=False))
98 model.add(layers.LSTM(100, return_sequences=False))
99 model.add(layers.LSTM(100, return_sequences=False))
100 model.add(layers.LSTM(100, return_sequences=False))
```

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 100, 100)	16,000
dropout_3 (Dropout)	(None, 100, 100)	0
lstm_4 (LSTM)	(None, 100)	16,000
dropout_4 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 25)	2,525
dense_3 (Dense)	(None, 1)	25

Total params: 32,025 (481.40 KB)
Trainable params: 32,025 (481.40 KB)
Non-trainable params: 0 (0.00 KB)

Gambar 5. Definisi Model

2. Kompilasi dan Pelatihan Model

Proses pertama meliputi pengaturan (konfigurasi) model LSTM agar siap untuk tahap pelatihan. Pada tahap ini, *optimizer* 'Adam' diaplikasikan untuk penyesuaian bobot model, dengan *learning rate* 0.001 yang mendukung optimalisasi proses pembelajaran model. Selain itu, penggunaan fungsi *loss* 'mean_squared_error' ditujukan untuk mengevaluasi seberapa jauh perbedaan hasil prediksi dibandingkan dengan nilai aktual. Rangkaian proses kompilasi ini secara total bertujuan memastikan kesiapan model untuk tahap pelatihan berikutnya.

Kemudian, dilakukan pelatihan pada model LSTM. Ini adalah sebuah prosedur di mana model secara progresif meningkatkan kemampuannya dalam memprediksi secara akurat dengan menyesuaikan parameter internalnya berdasarkan data latih. Tahapan ini meliputi pemrosesan data latih secara berulang, yang juga melibatkan koreksi terhadap bobot model sesuai dengan perbedaan antara output prediksi dan target sebenarnya. Faktor-faktor seperti jenis *optimizer* yang digunakan, jumlah *epoch*, dan ukuran *batch* memberikan pengaruh signifikan terhadap keseluruhan proses pelatihan ini. Penelitian ini menggunakan 16 *epoch* dan *batch_size* sebesar 30 untuk pelatihan model. Setelah proses pelatihan, model tersebut siap untuk diaplikasikan guna menguji kemampuannya dalam membuat prediksi terhadap data baru yang kemudian diikuti oleh evaluasi guna mengukur performa dan kapabilitas generalisasi model. Sementara itu, kode sumber untuk pelatihan model disajikan pada Gambar 6.

```
1 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='mean_squared_error')
2 model.fit(x_train, y_train, batch_size= 30, epochs=16)
```

```
Epoch 1/16
120/120 --- 10s 57ms/step - loss: 1.6076e-04
Epoch 2/16
120/120 --- 7s 54ms/step - loss: 1.2753e-04
Epoch 3/16
120/120 --- 11s 48ms/step - loss: 1.2600e-04
Epoch 4/16
120/120 --- 10s 60ms/step - loss: 1.3167e-04
Epoch 5/16
120/120 --- 10s 51ms/step - loss: 1.2691e-04
Epoch 6/16
120/120 --- 10s 52ms/step - loss: 1.1799e-04
Epoch 7/16
120/120 --- 10s 52ms/step - loss: 1.2107e-04
Epoch 8/16
120/120 --- 7s 60ms/step - loss: 1.2500e-04
Epoch 9/16
120/120 --- 10s 60ms/step - loss: 1.4307e-04
Epoch 10/16
120/120 --- 9s 52ms/step - loss: 1.3022e-04
Epoch 11/16
120/120 --- 10s 52ms/step - loss: 1.2400e-04
Epoch 12/16
120/120 --- 10s 52ms/step - loss: 1.3264e-04
Epoch 13/16
120/120 --- 7s 60ms/step - loss: 1.4041e-04
Epoch 14/16
120/120 --- 10s 57ms/step - loss: 1.2334e-04
Epoch 15/16
120/120 --- 10s 52ms/step - loss: 1.2229e-04
```

Gambar 6. Kompilasi dan Pelatihan Model

3.5 Evaluasi Performa

Tahap evaluasi model merupakan suatu prosedur sistematis untuk menilai performa sebuah model *machine learning* dengan mengacu pada metrik-metrik kuantitatif yang telah ditentukan. Pada penelitian ini, beberapa metrik pengujian utama digunakan untuk membandingkan kinerja model dengan variasi *timesteps*. Metriks pengujian ini diantaranya adalah Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), dan R-Squared (R^2). Adapun hasil perbandingan evaluasi model berdasarkan variasi *timesteps* dapat dilihat pada Tabel 2.

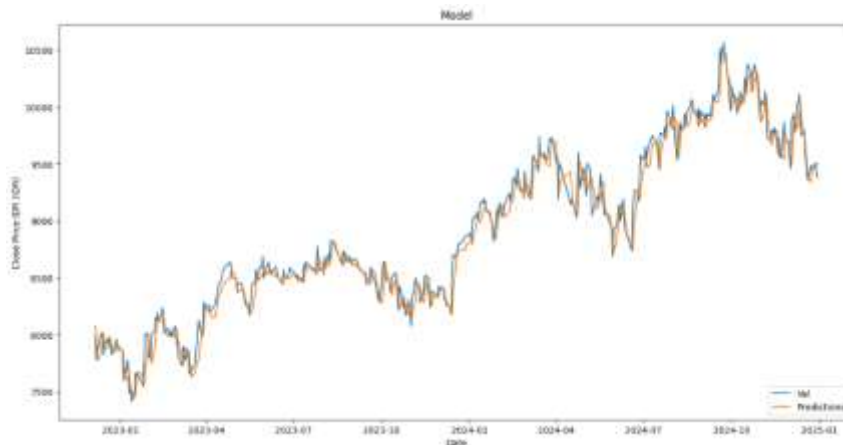
Tabel 2. Hasil Evaluasi Model

<i>Timesteps</i>	RMSE	MAPE	R-Squared
30	119.79	1.03%	0.97
60	114.17	0.96%	0.98
90	124.84	1.11%	0.97

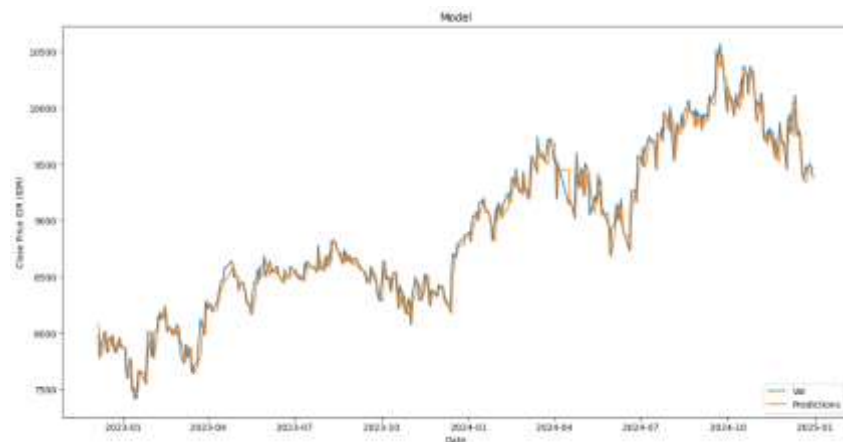
Dapat dilihat dari Tabel 2 bahwa model mendapatkan hasil terbaik ketika menggunakan *timesteps* (atau *sliding window*) sebanyak 60. Hal ini ditunjukkan dengan perolehan nilai RMSE dan MAPE terkecil, yaitu RMSE sebesar 114.17 dan MAPE sebesar 0.96%, serta nilai R-Squared terbesar, yaitu 0.98, dibandingkan dengan konfigurasi *timesteps* lainnya. Hasil ini mengindikasikan bahwa dengan *window* 60 hari, model mampu memprediksi tren dan pergerakan harga saham dengan baik.

3.6 Visualisasi

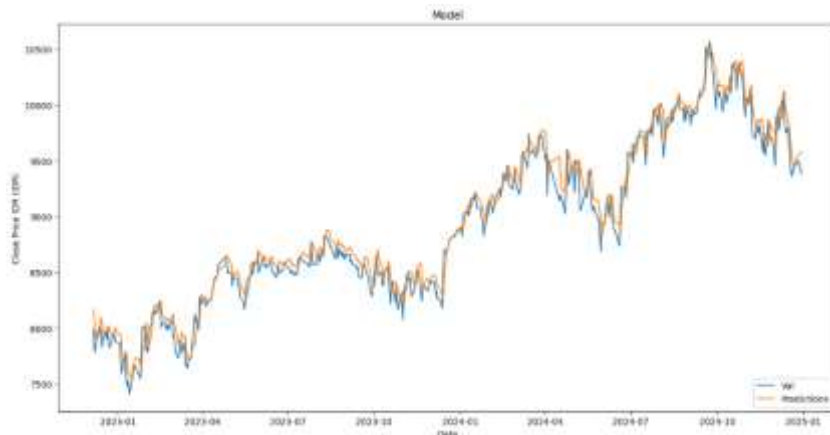
Visualisasi merupakan tahap penting dalam proses analisis data karena dapat memberikan gambaran intuitif mengenai performa model dalam memprediksi data. Dalam konteks ini, visualisasi digunakan untuk membandingkan antara harga penutupan aktual dan hasil prediksi model pada data uji untuk setiap konfigurasi *timesteps*.



Gambar 7. Visualisasi Prediksi dengan *timesteps* 30



Gambar 8. Visualisasi Prediksi dengan *timesteps* 60



Gambar 9. Visualisasi Prediksi dengan *timesteps* 90

Ketiga konfigurasi *timesteps* yang divisualisasikan secara konsisten menunjukkan kemampuan model LSTM untuk mengikuti tren umum harga saham aktual BBKA pada data uji. Meskipun demikian, pada periode sekitar April 2024, seluruh model prediksi tampak menunjukkan prediksi yang lebih datar atau kurang reaktif terhadap lonjakan atau penurunan tajam harga aktual, yang mengindikasikan potensi sensitivitas model terhadap dinamika pasar tertentu pada periode tersebut atau adanya faktor eksternal yang tidak tertangkap oleh data historis semata. Secara khusus, Gambar 8 (*timesteps* 60) secara visual menampilkan kecocokan prediksi yang paling konsisten dan akurat terhadap data aktual di luar periode anomali tersebut, dimana garis prediksi tampak sangat rapat mengikuti fluktuasi harga sebenarnya. Observasi visual ini sejalan dengan hasil metrik evaluasi yang mengidentifikasi *timesteps* 60 sebagai konfigurasi dengan performa terbaik.

4. KESIMPULAN

Dari penelitian implementasi model LSTM untuk prediksi harga saham BBKA yang telah dilakukan, dapat ditarik kesimpulan bahwa model LSTM berhasil diimplementasikan dan efektif untuk melakukan prediksi. Hasil evaluasi performa model menunjukkan bahwa konfigurasi dengan *timesteps* 60 hari memberikan kinerja prediksi paling optimal, mencapai tingkat akurasi prediksi yang sangat baik dengan nilai *Root Mean Square Error* (RMSE) sebesar 114.17, *Mean Absolute Percentage Error* (MAPE) sebesar 0.96%, dan *R-Squared* (R^2) sebesar 0.98. Nilai MAPE yang rendah dan *R-Squared* yang tinggi secara khusus mengindikasikan bahwa model tersebut tidak hanya menunjukkan akurasi dalam prediksi faktual, tetapi juga memiliki keunggulan dalam mempelajari dan menangkap pola variasi data historis harga saham. Penggunaan variabel teknikal lainnya seperti *open*, *volume*, dan indikator teknikal lainnya dapat dijadikan alternatif untuk penelitian selanjutnya. Perbandingan model LSTM dan model prediktif lainnya juga dapat dilakukan untuk menilai keunggulan prediksi dari masing-masing model.

REFERENCES

- [1] E. O. Gultom and M. I. Irawan, "Prediksi Harga Saham Jangka Pendek di Indonesia Menggunakan Metode Gaussian Process Regression," *Jurnal Sains dan Seni ITS*, vol. 11, no. 2, 2022, doi: 10.12962/j23373520.v11i2.76914.
- [2] Pratama Putu Agus Narestha Adi and Saepudin Deni, "Peramalan Return Saham pada IDX30 Menggunakan Economic Constraint Model dan Technical Indicators," 2022. Accessed: Dec. 30, 2024. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/21361>
- [3] A. Arfan and dan Lussiana ETP, "Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory," *Universitas Gunadarma Jl. Margonda Raya No*, vol. 3, no. 1, 2019, [Online]. Available: <https://www.ofx.com>
- [4] S. Maddodi and K. G. N. Kumar, "Stock Market Forecasting: a Review of Literature," *IJCSNS International Journal of Computer Science and Network Security*, vol. Vol 5, No, 2021.
- [5] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between ARIMA, LSTM, and GRU for time series forecasting," in *ACM International Conference Proceeding Series*, 2019. doi: 10.1145/3377713.3377722.
- [6] "Implementasi Long Short-Term Memory Pada Prediksi Harga Saham PT Aneka Tambang Tbk," *Jurnal Ilmiah Komputasi*, vol. 21, no. 1, 2022, doi: 10.32409/jikstik.21.1.2815.



- [7] L. Wiranda and M. Sadikin, "Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 8, no. 3, pp. 184–196, 2019.
- [8] Haoran Wu, Shuqi Chen, and Yicheng Ding, "Comparison of ARIMA and LSTM for Stock Price Prediction," *Financial Engineering and Risk Management*, vol. 6, no. 1, 2023, doi: 10.23977/ferm.2023.060101.
- [9] R. Julian and M. R. Pribadi, "Peramalan Harga Saham Pertambangan Pada Bursa Efek Indonesia (BEI) Menggunakan Long Short Term Memory (LSTM)," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 3, 2021, doi: 10.35957/jatisi.v8i3.1159.
- [10] Moh. Wigi Destriansyah and D. A. N. Sirodj, "Analisis Hubungan Harga Saham Bank Central Asia, Inflasi, Kurs (IDR/USD) dan BI Rate dengan Metode Vector Error Correction Model (VECM)," *Bandung Conference Series: Statistics*, vol. 2, no. 2, 2022, doi: 10.29313/bcss.v2i2.4057.
- [11] N. Selle, N. Yudistira, and C. Dewi, "Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 1, 2022, doi: 10.25126/jtiik.2022915585.
- [12] M. Q. Andiyantama, I. Zahira, and A. Irawan, "Prediksi Energi Listrik Kincir Angin Berdasarkan Data Kecepatan Angin Menggunakan LSTM," *JITCE (Journal of Information Technology and Computer Engineering)*, vol. 5, no. 01, 2021, doi: 10.25077/jitce.5.01.1-7.2021.
- [13] S. Zahara, Sugianto, and M. Bahril Ilmiddafiq, "Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 3, 2019, doi: 10.29207/resti.v3i3.1086.
- [14] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Machine Learning with Applications*, vol. 9, 2022, doi: 10.1016/j.mlwa.2022.100320.
- [15] B. K. Hidayatullah, M. Kallista, C. Setianingsih, P. S1, and T. Komputer, "Prediksi Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory Berbasis Web (Studi Kasus Pada Kota Jakarta)," *e-Proceeding of Engineering*, vol. 9, no. 3, pp. 1247–1255, 2022, [Online]. Available: <https://data.jakarta.go.id/>
- [16] V. L. N. Komanapalli, N. Sivakumaran, and S. Hampannavar, *Advances in Automation, Signal Processing, Instrumentation, and Control: Select Proceedings of i-CASIC 2020*, vol. 700, 2021.
- [17] A. Rosyd, A. Irma Purnamasari, and I. Ali, "PENERAPAN METODE LONG SHORT TERM MEMORY (LSTM) DALAM MEMPREDIKSI HARGA SAHAM PT BANK CENTRAL ASIA," 2024.
- [18] D. Ardiansyah, "PERBANDINGAN MODEL PREDIKSI RADIASI MATAHARI BERBASIS MESIN PEMBELAJARAN PADA STASIUN METEOROLOGI FATMAWATI SOEKARNO BENGKULU," *Megasains*, vol. 14, no. 1, 2023, doi: 10.46824/megasains.v14i1.129.
- [19] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Computer Engineering, Science and System Journal*, vol. 4, no. 1, 2019, doi: 10.24114/cess.v4i1.11458.

