

**LAPORAN *RESEARH BASED LEARNING* (RBL)
FI3203 SIMULASI DAN PEMODELAN SISTEM FISIS**

**ANALISIS PREDIKSI HARGA SAHAM PT BANK CENTAL ASIA
(BCA) MENGGUNAKAN *LONG SHORT TERM MEMORY* (LSTM)**

oleh

Kelompok 4

**Mita Refalina, Arif Laksana, Tysha Alya Ramadhany, Leonardo Diaugusto
10221027, 10221030, 10221075, 10221077**



**PROGRAM STUDI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI BANDUNG
2024-2025**

ABSTRAK

Penelitian ini bertujuan untuk menerapkan metode Long Short-Term Memory (LSTM) dalam memprediksi harga saham PT Bank Central Asia (BCA), menghasilkan model prediksi dengan tingkat akurasi yang baik, serta menilai keunggulan metode LSTM dalam memberikan hasil prediksi yang akurat. Data historis harga saham yang digunakan mencakup periode Desember 2023 hingga Desember 2024, diperoleh melalui Yahoo Finance. Proses penelitian melibatkan pengumpulan data, preprocessing, pembangunan model LSTM, pelatihan, evaluasi, dan analisis hasil prediksi. Hasil penelitian menunjukkan bahwa model LSTM berhasil memproyeksikan pergerakan harga saham dengan menangkap pola umum tren kenaikan dan penurunan yang stabil. Evaluasi model menunjukkan nilai Mean Absolute Error (MAE) sebesar 150,11 IDR, MAE relatif sebesar 1,53%, dan Mean Absolute Percentage Error (MAPE) sebesar 1,44%, yang mencerminkan tingkat kesalahan prediksi yang sangat kecil. Dengan hasil ini, model LSTM terbukti memiliki kemampuan prediksi yang akurat dan dapat membantu investor maupun perusahaan dalam mengambil keputusan investasi yang lebih terinformasi. Namun, model ini memiliki keterbatasan karena tidak mempertimbangkan faktor eksternal yang bersifat dinamis, seperti perubahan kondisi ekonomi, kebijakan pemerintah, atau sentimen pasar, yang dapat memengaruhi hasil prediksi. Oleh karena itu, kombinasi antara hasil prediksi model dengan analisis fundamental dan indikator tambahan direkomendasikan untuk mendukung pengambilan keputusan yang lebih holistik dan andal.

Kata kunci: akurasi, data deret waktu, LSTM, prediksi harga saham, PT Bank Central Asia

DAFTAR ISI

ABSTRAK	1
DAFTAR ISI	2
Bab I Pendahuluan.....	4
I.1 Latar Belakang	4
I.2 Rumusan Masalah.....	6
I.3 Tujuan	6
Bab II Metodologi.....	7
II.1 Pengumpulan Data	7
II.2 Preprocessing Data	8
II.3 Membangun Model LSTM	9
II.4 Model Training	11
II.5 Evaluasi Model	11
II.6 Analisis	12
Bab III Hasil Simulasi.....	13
III.1 Pengumpulan Data	13
III.2 Preprocessing Data	14
III.3 Membangun Model LSTM	18
III.4 Model Training	19
III.5 Evaluasi Model	20
III.6 Prediksi Pada Data Testing	21
III.7 Menentukan Akurasi Model	22
III.8 Menyimpan Model.....	23
III.9 Prediksi Harga Saham Beberapa Hari Akan Datang	23
Bab IV Analisis.....	25
IV.1 Perbandingan Hasil Prediksi dan Data Aktual.....	25
IV.2 Evaluasi Nilai MAE dan MAPE.....	26
IV.3 Prediksi Harga Saham 5 Hari ke Depan	27
IV.4 Manfaat untuk Investor dan Manajemen Risiko.....	27
IV.5 Keandalan Model untuk Pengambilan Keputusan.....	28
Bab V Kesimpulan.....	29

DAFTAR PUSTAKA.....	30
LAMPIRAN	31

Bab I Pendahuluan

I.1 Latar Belakang

Investasi di pasar modal telah menjadi salah satu pilihan populer untuk mengembangkan aset keuangan. Di antara berbagai instrumen investasi, saham menjadi salah satu yang paling diminati karena menawarkan potensi keuntungan yang tinggi. Saham (stock) merupakan instrumen pasar keuangan yang sangat digemari dibandingkan dengan instrumen lainnya karena memiliki potensi imbal hasil (yield) yang menarik. Popularitas saham ini dapat terlihat dari data yang dirilis oleh PT Bursa Efek Indonesia (BEI) pada konferensi pers akhir tahun 2022. Hingga 28 Desember 2022, jumlah total investor pasar modal di Indonesia mencapai 10,3 juta, menunjukkan peningkatan sebesar 37,5 persen dibandingkan akhir tahun 2021 yang mencatatkan 7,48 juta investor. Bahkan, jika dibandingkan dengan lima tahun sebelumnya, jumlah investor ini telah meningkat hingga lima kali lipat (Sukma, 2022).

Namun, permasalahan utama yang dihadapi oleh investor adalah bagaimana memprediksi pergerakan harga saham di masa depan agar dapat mengambil keputusan investasi yang optimal. Hal ini menjadi tantangan besar, terutama bagi investor pemula yang seringkali belum memiliki pengetahuan dan pengalaman yang memadai dalam menganalisis pasar saham. Pasar saham menawarkan peluang keuntungan yang besar, tetapi juga disertai dengan risiko kerugian yang signifikan dalam waktu singkat. Oleh karena itu, diperlukan pemahaman yang mendalam dan strategi yang tepat untuk meminimalkan potensi kerugian. Dalam upaya membantu investor, khususnya pemula, diperlukan sebuah sistem yang dapat memprediksi pergerakan harga saham berdasarkan faktor teknis. Sistem ini harus mampu memproses data historis, menganalisis tren, dan memberikan rekomendasi yang jelas dan mudah dipahami.

Bank adalah salah satu sektor yang memiliki peranan penting dalam mendukung stabilitas dan pertumbuhan ekonomi. Harga saham bank mencerminkan kesehatan sektor keuangan dan sering kali digunakan sebagai indikator kepercayaan investor terhadap perekonomian suatu negara. Oleh karena itu, analisis dan prediksi harga

saham bank memiliki nilai strategis, baik bagi investor, regulator, maupun perusahaan itu sendiri. Penelitian tentang prediksi harga saham telah banyak dilakukan menggunakan berbagai metode. Salah satu contohnya adalah penelitian oleh W. R. U. Fadilah, D. Agfiannisa, dan Y. Azhar (Fadilah et al., 2020) yang membahas prediksi harga saham PT. Telekomunikasi Indonesia menggunakan metode Support Vector Machine (SVM). Hasil penelitian menunjukkan bahwa SVM memberikan nilai akurasi yang sangat baik, yaitu sebesar 0,9641 dengan nilai RMSE 0,0932. Selain itu, terdapat penelitian oleh A. Site, D. Birant, dan Z. Isik (Özkurt et al., 2019) membandingkan algoritma Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), dan Gated Recurrent Unit (GRU). Hasil penelitian menunjukkan bahwa algoritma LSTM memiliki konsistensi dan tingkat akurasi prediksi yang lebih baik dibandingkan algoritma RNN dan GRU.

Salah satu pendekatan yang berkembang pesat dalam analisis prediksi harga saham adalah penggunaan kecerdasan buatan (Artificial Intelligence) dan pembelajaran mesin (Machine Learning). Di antara berbagai teknik yang ada, Long Short Term Memory (LSTM), sebuah varian dari jaringan saraf tiruan (Recurrent Neural Network), telah terbukti efektif dalam menganalisis data deret waktu (time series). LSTM dirancang untuk menangkap pola dan hubungan jangka panjang dalam data historis, sehingga mampu memprediksi pergerakan harga saham dengan tingkat akurasi yang lebih baik dibandingkan metode tradisional (Abdul et al., 2019).

Bank sebagai salah satu pilar utama perekonomian memiliki peran yang sangat signifikan di pasar saham. Stabilitas dan kinerja saham bank mencerminkan kondisi ekonomi secara umum, menjadikannya sebagai perhatian utama bagi investor. Oleh karena itu, analisis prediktif terhadap harga saham bank menggunakan LSTM menjadi relevan untuk memberikan wawasan mendalam mengenai tren harga saham, membantu investor memanfaatkan peluang pasar, dan mengurangi risiko keputusan investasi yang tidak terinformasi.

I.2 Rumusan Masalah

Rumusan masalah dari proyek Research Based Learning (RBL) ini, yaitu:

1. Bagaimana metode Long Short Term Memory (LSTM) dapat diterapkan untuk memprediksi harga saham PT Bank Central Asia (BCA) secara efektif?
2. Bagaimana cara menghasilkan model prediksi harga saham PT Bank Central Asia (BCA) yang memiliki tingkat akurasi yang baik?
3. Apakah metode LSTM memiliki keunggulan dalam memberikan hasil prediksi harga saham yang lebih akurat?

I.3 Tujuan

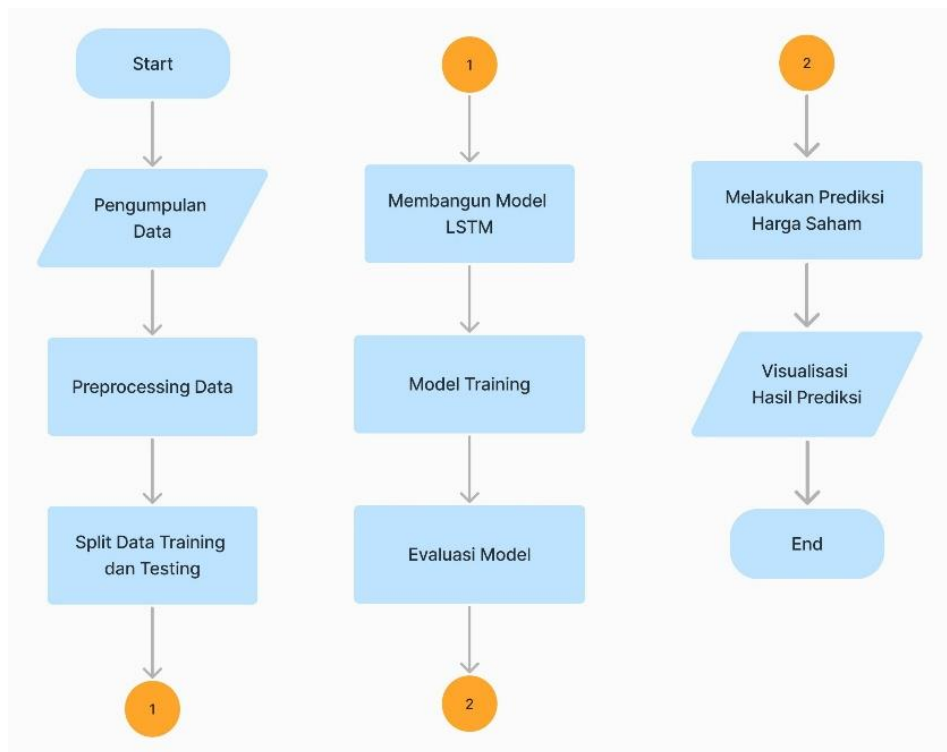
Tujuan dari proyek Research Based Learning (RBL) ini, yaitu:

1. Menerapkan metode Long Short Term Memory (LSTM) dalam memprediksi harga saham PT Bank Central Asia (BCA).
2. Menghasilkan model prediksi harga saham PT Bank Central Asia (BCA) dengan tingkat akurasi yang baik.
3. Menilai keunggulan metode LSTM dalam memberikan hasil prediksi yang akurat.

Bab II Metodologi

Dalam melakukan analisis harga saham, metode yang digunakan adalah algoritma *Long Short Term Memory* (LSTM). Algoritma LSTM merupakan salah satu jenis *Reccurent Neural Network* (RNN) yang dirancang untuk menangani data sekuensial. Secara umum langkah-langkah yang diterapkan dalam penelitian ini terdiri dari tahap pengumpulan data, tahap *preprocessing* data, pembentukan model machine learning LSTM, *model training*, evaluasi model, dan analisis data.

Berikut merupakan flowchart analisis prediksi saham menggunakan metode LSTM.



Gambar 1. Flowchart LSTM

II.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan menggunakan bahasa pemrograman Python dengan memanfaatkan *platform* Google Colab. Data yang digunakan merupakan data historis harga saham perbankan PT Bank Cental Asia yang diunduh melalui yahoo finance. Data yang digunakan merupakan data

historis harga penutupan saham yang dimulai pada tanggal 1 Desember 2023 sampai dengan tanggal 1 Desember 2024.

II.2 Preprocessing Data

Tahap yang dapat dilakukan setelah pengumpulan data adalah tahap *preprocessing*. Tahap *preprocessing* merupakan tahap untuk melakukan proses perbaikan data yaitu dengan melakukan penghilangan noise atau missing value [1]. Data yang telah melalui tahap *preprocessing* akan menjadi data yang bersih dan terstruktur. Tahap *preprocessing* sangat penting, terutama dalam analisis saham, karena data saham bersifat *time series*. Data jenis ini sering kali mengandung elemen yang tidak relevan atau tidak konsisten, seperti nilai kosong, lonjakan data (outliers), dan skala data yang berbeda-beda. Oleh karena itu, *preprocessing* membantu menjaga integritas data dan memungkinkan model pembelajaran mesin, seperti LSTM, untuk mengenali pola yang lebih akurat. Terdapat beberapa tahap yang dapat dilakukan yaitu pengumpulan data historis, pembersihan data, normalisasi data, pembagian dataset, dan perubahan bentuk data.

1. Pengumpulan Data Historis

Data historis saham yang dikumpulkan, yaitu harga pembukaan (open), penutupan (close), harga tertinggi (high), harga terendah (low), dan volume. Dari data historis tersebut, akan digunakan data penutupan (close) untuk dianalisis menggunakan metode LSTM.

2. Pembersihan Data

Pada proses ini dilakukan penghilangan nilai kosong, duplikasi data, dan elemen yang tidak relevan yang tidak memberikan makna apapun.

3. Normalisasi Data

Proses normalisasi dilakukan dengan mengubah nilai aktual harga penutupan menjadi nilai dengan rentang 0-1. Salah satu teknik yang dapat digunakan untuk normalisasi data untuk min-max normalization. Hal ini dilakukan untuk mempermudah proses pelatihan model.

4. Split Data

Proses ini merupakan proses pemisahan data menjadi data training dan data testing. Rasio pembagian data disesuaikan dengan jumlah data

yang tersedia pada data historis. Misalnya dapat digunakan rasio 70% untuk data training dan 30% untuk data testing.

5. Perubahan Bentuk Data

Proses ini merupakan proses perubahan data sesuai dengan format yang dibutuhkan. Pada model LSTM dilakukan perubahan data ke dalam bentuk sekuensial. Data juga dapat diubah menjadi bentuk tiga dimensi dengan format (samples, timesteps, Features) agar mempermudah proses oleh model LSTM. Proses ini memastikan bahwa model dapat mengenali pola temporal dalam data.

II.3 Membangun Model LSTM

Pada tahap ini, akan dibangun salah satu model *machine learning*, yaitu model *Long Short Term Memory* (LSTM) untuk prediksi harga saham. Arsitektur dari model LSTM menyerupai arsitektur *Artificial Neural Network* (ANN). Arsitektur LSTM terdiri dari 3 bagian, antara lain forget gate, input gate, dan output gate. Cell pertama menentukan apakah suatu informasi dari timestamp sebelumnya relevan atau dapat dilupakan. Cell kedua akan mencoba mempelajari informasi baru dari input. Kemudian, cell ketiga mengirimkan informasi yang telah diperbarui dari timestamp sekarang ke timestamp selanjutnya. LSTM juga terdiri dari hidden state berfungsi untuk melakukan karakterisasi atau selective memory dari data sebelumnya dan cell state yang berfungsi untuk menampung semua informasi sebelumnya.

1. Forget Gate

Gate pertama berfungsi untuk menentukan apakah informasi dari timestamp sebelumnya perlu dipertahankan atau tidak.

$$f_t = \sigma(x_t * U_f + H_{t-1} * W_f) \quad (II.1)$$

Dengan x_t , adalah input ke dalam *timestamp* saat ini, U_f merupakan bobot atau *weight input*, H_{t-1} adalah *hidden state* untuk *timestamp* sebelumnya, dan W adalah bobot matriks untuk *hidden state*.

Kemudian, hasil data persamaan diatas akan diaplikasikan fungsi *sigmoid* untuk mengubah menjadi bentuk angka diantara 0 dan 1, lalu dikalikan dengan *cell state timestamp* sebelumnya (C_{t-1}). Angka 0 menandakan informasi yang dilupakan dan angka 1 menandakan informasi yang dipertahankan.

$$C_{t-1} * f_t = 0 \dots \text{jika } f_t = 0 \quad (II.2)$$

$$C_{t-1} * f_t = C_{t-1} \dots \text{jika } f_t = 1 \quad (II.3)$$

2. Input Gate

Input Gate digunakan untuk mengukur pentingnya informasi baru yang dibawa oleh input, dengan persamaan berikut.

$$i_t = \sigma(x_t * U_i + H_{t-1} * W_i) \quad (II.4)$$

Dengan x_t adalah input ke dalam *timestamp* saat ini, U_i merupakan bobot atau *weight matriks input*, H_{t-1} adalah *hidden state* untuk *timestamp* sebelumnya, dan W , adalah bobot matriks untuk *hidden state*.

Hasil i_t diatas akan diterapkan fungsi *sigmoid* seperti gate sebelumnya dan nilai 0 serta 1 akan menentukan status informasi tersebut.

3. Informasi Baru

Informasi baru perlu melewati *cell state*, merupakan fungsi *hidden state* (H_{t-1}) dan input x_t dengan aktivasi *tanh* sehingga nilai N_t antara -1 dan 1.

$$N_t = \tanh(x_t * U_c + H_{t-1} * W_c) \quad (II.5)$$

Jika nilai N_t negatif, maka informasi akan dikurangi dari *cell state*. Namun, jika nilai N_t positif, informasi akan ditambahkan ke dalam *cell state* (x_t). Kemudian, nilai N_t akan melewati persamaan berikut untuk ditambahkan.

$$C_t = f_t * C_{t-1} + i_t * N_t \quad (II.6)$$

4. Output Gate

Berikut merupakan persamaan untuk output gate dengan rentang nilai 0 sampai 1 oleh karena fungsi *sigmoid*.

$$\sigma_t = \sigma(\chi_t * U_o + H_{t-1} * W_o) \quad (II.7)$$

Untuk menentukan *hidden state* saat ini, akan digunakan persamaan sebagai berikut.

$$H_t = o_t * \tanh(c_t) \quad (II.8)$$

Output dari fungsi diatas merupakan token dengan nilai maksimum dari *output timestamp* saat ini.

II.4 Model Training

Model training merupakan proses inti dari pembelajaran mesin di mana model belajar untuk memetakan data input ke output yang diinginkan. Pada proses ini, model belajar dari data training untuk menemukan pola dan hubungan antara input dan output yang dapat digunakan untuk membuat prediksi pada data testing. Pada tahap pelatihan model ini, model akan mengoptimalkan bobot dan parameter internalnya untuk meminimalkan error atau kesalahan prediksi. Model dilatih dengan mengikuti proses iterasi selama epoch tertentu. Pada setiap epoch, model akan mengupdate bobotnya berdasarkan backpropagation untuk meminimalkan loss function.

II.5 Evaluasi Model

Evaluasi model merupakan tahap yang dilakukan untuk menilai performa model dalam memprediksi harga saham berdasarkan data testing. Dalam melakukan evaluasi algoritma Long Short Term Memory dalam melakukan analisis saham, digunakan Mean Square Error (MSE) [2]. MSE merupakan metrik yang digunakan dengan mengukur rata-rata kesalahan kuadrat antara nilai prediksi dan nilai sebenarnya. Semakin kecil nilai MSE yang didapatkan, maka semakin baik

performa model yang digunakan. Evaluasi model dilakukan berdasarkan hasil loss dan val_loss pada tahap sebelumnya

II.6 Analisis

Setelah proses evaluasi selesai dilakukan, tahap selanjutnya adalah proses prediksi untuk memproyeksikan pergerakan kurva saham dan membandingkannya dengan harga saham sesungguhnya untuk melihat seberapa akurat model yang digunakan. Proses prediksi dilakukan dengan memanfaatkan model LSTM yang telah dilatih sebelumnya, menggunakan sintaks “`model.predict()`” untuk menghasilkan prediksi berdasarkan data testing. Hasil dari proses prediksi ini kemudian divisualisasikan dalam bentuk grafik, yang menunjukkan proyeksi pergerakan harga saham di masa mendatang. Grafik ini memungkinkan investor atau analis untuk memahami tren yang diperkirakan terjadi, seperti kenaikan, penurunan, atau stabilisasi harga saham. Selain itu, informasi ini dapat digunakan sebagai dasar pengambilan keputusan investasi atau strategi perdagangan. Dengan pendekatan berbasis model LSTM, prediksi ini diharapkan mampu memberikan wawasan yang lebih akurat mengenai pola pergerakan harga saham berdasarkan data historis yang tersedia.

Bab III Hasil Simulasi

III.1 Pengumpulan Data

Berikut merupakan python code untuk pengumpulan data.

```
import yfinance as yf

# Mengambil data saham
ticker = ['BBCA.JK']

data_harian = yf.download(ticker, start='2023-12-01',
end='2024-12-01', interval='1d')['Adj Close']
print()
print(data_harian)
```

Diperoleh data saham BBKA sebagai berikut.

Ticker	BBKA.JK
Date	
2023-12-01	8663.650391
2023-12-04	8680.670898
2023-12-05	8656.355469
2023-12-06	8559.092773
2023-12-07	8583.408203
...	...
2024-11-22	9850.000000
2024-11-25	10175.000000
2024-11-26	10000.000000
2024-11-28	10000.000000
2024-11-29	10000.000000
[237 rows x 1 columns]	

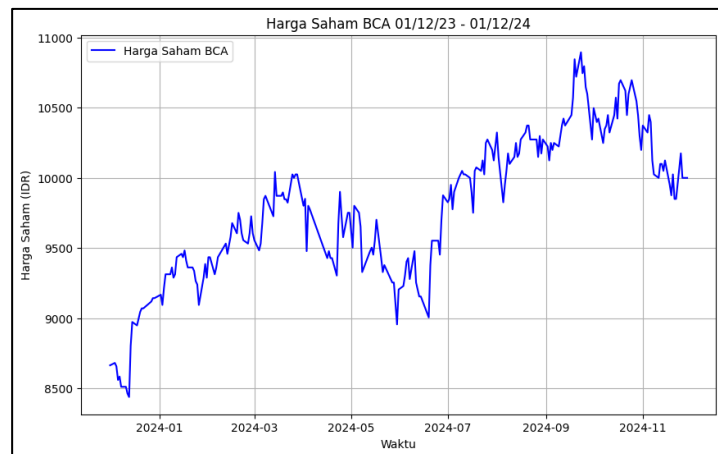
Gambar III.1 Data Saham BBKA

Jika dilakukan plot grafik, diperoleh grafik saham BBKA sebagai berikut.

```
import matplotlib.pyplot as plt

# Plot grafik saham
plt.figure(figsize=(10, 6))
plt.plot(data_harian, label='Harga Saham BCA', color='blue')
plt.title('Harga Saham BCA 01/12/23 - 01/12/24')
plt.xlabel('Waktu')
plt.ylabel('Harga Saham (IDR)')
plt.legend()
plt.grid()
```

```
plt.show()
```



Gambar III.2 Grafik Saham BBKA

III.2 Preprocessing Data

Berikut merupakan python code untuk preprocessing data.

Assessing Data

```
# Memeriksa missing value  
data_harian.isna().sum()
```

```
0  
Ticker  
BBKA.JK 0  
dtype: int64
```

Gambar III.3 Missing Value

```
# Melihat parameter statistik data  
data_harian.describe()
```

Ticker	BBCA.JK
count	237.000000
mean	9791.703372
std	520.157528
min	8437.514648
25%	9427.977539
50%	9847.820312
75%	10199.130859
max	10895.657227

Gambar III.4 Parameter Statistik BBKA

Normalisasi Data

```
from sklearn.preprocessing import MinMaxScaler

# Melakukan normalisasi data ke rentang [0,1]
scaler = MinMaxScaler(feature_range=(0, 1))
normalized_data =
scaler.fit_transform(data_harian.values.reshape(-1, 1))

# Menampilkan data normalisasi
print(normalized_data)
```

```
[0.21762096]
[0.20772917]
[0.22751275]
[0.24729673]
[0.25718852]
[0.25718852]
[0.2769721 ]
[0.28686429]
[0.28686429]
[0.29675608]
[0.26708031]
[0.31653966]
[0.35610721]
[0.35610721]
[0.37589119]
[0.34621542]
[0.35610721]
[0.40556616]
```

Gambar III.5 Output Normalisasi Data (hanya sebagian)

Membuat Window untuk Data Time Series

```
# Window Data Time Series untuk memprediksi harga saham
berikutnya

import numpy as np

# X: data input model selama window_size hari terakhir
# y: harga saham pada hari ke-i

X, y = [], []
window_size = 60
for i in range(window_size, len(normalized_data)):
    X.append(normalized_data[i-window_size:i, 0])
    y.append(normalized_data[i, 0])

X = np.array(X)
y = np.array(y)
```

Split Data Training dan Testing

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, shuffle=False)
```

```
print(X_train)

[[0.09199456 0.09891869 0.0890269 ... 0.52426883 0.47480949 0.45502551]
 [0.09891869 0.0890269 0.04945935 ... 0.47480949 0.45502551 0.42535014]
 [0.0890269 0.04945935 0.05935114 ... 0.45502551 0.42535014 0.44513372]
 ...
 [0.29161333 0.23089477 0.38269177 ... 0.71664525 0.73688491 0.72676508]
 [0.23089477 0.38269177 0.45353056 ... 0.73688491 0.72676508 0.75712456]
 [0.38269177 0.45353056 0.45353056 ... 0.72676508 0.75712456 0.78748404]]

print(X_test)

[[0.45353056 0.45353056 0.45353056 ... 0.75712456 0.78748404 0.80772369]
 [0.45353056 0.45353056 0.41305125 ... 0.78748404 0.80772369 0.78748404]
 [0.45353056 0.41305125 0.51424912 ... 0.80772369 0.78748404 0.81784352]
 ...
 [0.72676508 0.68628617 0.73688491 ... 0.5746149 0.5746149 0.70682855]
 [0.68628617 0.73688491 0.71664525 ... 0.5746149 0.70682855 0.63563658]
 [0.73688491 0.71664525 0.73688491 ... 0.70682855 0.63563658 0.63563658]]
```

Gambar III.6 Hasil Split Data X

```

print(y_train)

[0.42535014 0.44513372 0.50448525 0.57372818 0.58361997 0.52426883
0.65286289 0.58361997 0.58361997 0.59351216 0.57372818 0.57372818
0.56383639 0.64580686 0.63568704 0.64580686 0.64580686 0.55472843
0.57496808 0.42317108 0.55472843 0.5446086 0.40293142 0.42317108
0.40293142 0.40293142 0.35233229 0.50412929 0.59520773 0.52436895
0.46365038 0.53448877 0.53448877 0.4332909 0.55472843 0.53448877
0.49400947 0.36245212 0.42317108 0.4332909 0.41305125 0.45353056
0.51424912 0.40293142 0.36245212 0.38269177 0.33209264 0.33209264
0.27137408 0.21065512 0.31185299 0.32197281 0.35233229 0.3928116
0.40293142 0.34221247 0.42317108 0.33209264 0.31185299 0.29161333
0.29161333 0.23089477 0.38269177 0.45353056 0.45353056 0.45353056
0.41305125 0.51424912 0.58508791 0.56484825 0.57496808 0.61544739
0.5446086 0.59520773 0.63568704 0.64580686 0.65592669 0.64580686
0.64580686 0.63568704 0.59520773 0.53448877 0.65592669 0.66604652
0.65592669 0.68628617 0.64580686 0.73688491 0.74700473 0.71664525
0.68628617 0.72676508 0.76724439 0.696406 0.56484825 0.61544739
0.65592669 0.70652543 0.67616634 0.696406 0.73688491 0.696406
0.70652543 0.74700473 0.76724439 0.78748404 0.78748404 0.74700473
0.74700473 0.74700473 0.696406 0.75712456 0.70652543 0.74700473
0.72676508 0.68628617 0.73688491 0.71664525 0.73688491 0.72676508
0.75712456 0.78748404 0.80772369]

print(y_test)

[0.78748404 0.81784352 0.86844265 0.97976035 0.92916121 1.
0.93928104 0.95952069 0.89880213 0.87856248 0.74700473 0.83808317
0.81784352 0.79760387 0.80772369 0.73688491 0.77736421 0.78748404
0.81784352 0.76724439 0.81784352 0.86844265 0.80772369 0.90892196
0.91904139 0.88868231 0.81784352 0.87856248 0.89880213 0.91904139
0.85832283 0.81784352 0.75712456 0.71664525 0.78748404 0.76724439
0.81784352 0.79760387 0.68628617 0.64580686 0.63568704 0.67616634
0.67616634 0.65592669 0.68628617 0.61544739 0.58508791 0.64580686
0.5746149 0.5746149 0.70682855 0.63563658 0.63563658 0.63563658]

```

Gambar III.7 Hasil Split Data y

Mengubah Bentuk Data

```

# mengubah bentuk data untuk model LSTM [samples, time_steps,
features]
X_train = X_train.reshape(-1, window_size, 1)
X_test = X_test.reshape(-1, window_size, 1)

```

```

print(X_train)

[[[0.09199456]
 [0.09891869]
 [0.0890269 ]
 ...
 [0.52426883]
 [0.47480949]
 [0.45502551]]

 [[0.09891869]
 [0.0890269 ]
 [0.04945935]
 ...
 [0.47480949]
 [0.45502551]
 [0.42535014]]

 [[0.0890269 ]
 [0.04945935]
 [0.05935114]
 ...
 [0.45502551]
 [0.42535014]
 [0.44513372]]

 ...

```

Gambar III.8 Bentuk Data X train untuk Model LSTM

```

print(X_test)

[[[0.45353056]
  [0.45353056]
  [0.45353056]
  ...
  [0.75712456]
  [0.78748404]
  [0.80772369]]

  [[0.45353056]
  [0.45353056]
  [0.41305125]
  ...
  [0.78748404]
  [0.80772369]
  [0.78748404]]

  [[0.45353056]
  [0.41305125]
  [0.51424912]
  ...
  [0.80772369]
  [0.78748404]
  [0.81784352]]

  ...

```

Gambar III.9 Bentuk Data X test untuk Model LSTM

III.3 Membangun Model LSTM

```

# Model LSTM

import tensorflow as tf

model = tf.keras.models.Sequential([
    # Input layer
    tf.keras.Input(shape=(window_size, 1)),

    # LSTM layer pertama
    tf.keras.layers.LSTM(120, return_sequences=True),
    tf.keras.layers.Dropout(0.2),

    # LSTM layer kedua
    tf.keras.layers.LSTM(120, return_sequences=False),
    tf.keras.layers.Dropout(0.2),

    # Output layer
    tf.keras.layers.Dense(1)
])

# Compile model
model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mae'])

# Menampilkan model
model.summary()

```

Model: "sequential"		
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 120)	58,560
dropout (Dropout)	(None, 60, 120)	0
lstm_1 (LSTM)	(None, 120)	115,680
dropout_1 (Dropout)	(None, 120)	0
dense (Dense)	(None, 1)	121
Total params: 174,361 (681.10 KB)		
Trainable params: 174,361 (681.10 KB)		
Non-trainable params: 0 (0.00 B)		

Gambar III.10 Struktur Model LSTM

III.4 Model Training

```
# Melatih model
```

```
history = model.fit(X_train, y_train, epochs=50,
batch_size=32, validation_split=0.2)
```

Epoch 1/50	
4/4	4s 260ms/step - loss: 0.1916 - mae: 0.3984 - val_loss: 0.0266 - val_mae: 0.1585
Epoch 2/50	
4/4	1s 128ms/step - loss: 0.0217 - mae: 0.1177 - val_loss: 0.0294 - val_mae: 0.1678
Epoch 3/50	
4/4	1s 127ms/step - loss: 0.0222 - mae: 0.1248 - val_loss: 0.0446 - val_mae: 0.2083
Epoch 4/50	
4/4	1s 133ms/step - loss: 0.0213 - mae: 0.1236 - val_loss: 0.0019 - val_mae: 0.0344
Epoch 5/50	
4/4	1s 180ms/step - loss: 0.0122 - mae: 0.0893 - val_loss: 0.0017 - val_mae: 0.0360
Epoch 6/50	
4/4	2s 245ms/step - loss: 0.0132 - mae: 0.0956 - val_loss: 0.0124 - val_mae: 0.1059
Epoch 7/50	
4/4	1s 266ms/step - loss: 0.0108 - mae: 0.0895 - val_loss: 0.0126 - val_mae: 0.1067
Epoch 8/50	
4/4	1s 133ms/step - loss: 0.0101 - mae: 0.0830 - val_loss: 0.0013 - val_mae: 0.0282
Epoch 9/50	
4/4	1s 127ms/step - loss: 0.0123 - mae: 0.0888 - val_loss: 0.0013 - val_mae: 0.0282
Epoch 10/50	
4/4	1s 136ms/step - loss: 0.0109 - mae: 0.0851 - val_loss: 0.0151 - val_mae: 0.1177
Epoch 11/50	
4/4	1s 125ms/step - loss: 0.0101 - mae: 0.0875 - val_loss: 0.0169 - val_mae: 0.1253
Epoch 12/50	
4/4	1s 136ms/step - loss: 0.0094 - mae: 0.0805 - val_loss: 0.0075 - val_mae: 0.0789
Epoch 13/50	
4/4	1s 129ms/step - loss: 0.0098 - mae: 0.0821 - val_loss: 0.0035 - val_mae: 0.0499
Epoch 14/50	
4/4	1s 128ms/step - loss: 0.0084 - mae: 0.0727 - val_loss: 0.0029 - val_mae: 0.0450
Epoch 15/50	
4/4	1s 133ms/step - loss: 0.0087 - mae: 0.0766 - val_loss: 0.0026 - val_mae: 0.0416
Epoch 16/50	
4/4	1s 126ms/step - loss: 0.0093 - mae: 0.0791 - val_loss: 0.0033 - val_mae: 0.0485
Epoch 17/50	
4/4	1s 134ms/step - loss: 0.0078 - mae: 0.0735 - val_loss: 0.0047 - val_mae: 0.0593
Epoch 18/50	
4/4	1s 129ms/step - loss: 0.0086 - mae: 0.0787 - val_loss: 0.0094 - val_mae: 0.0904
Epoch 19/50	
4/4	1s 135ms/step - loss: 0.0089 - mae: 0.0811 - val_loss: 0.0069 - val_mae: 0.0747
Epoch 20/50	
4/4	1s 127ms/step - loss: 0.0095 - mae: 0.0793 - val_loss: 0.0019 - val_mae: 0.0341
Epoch 21/50	
4/4	1s 129ms/step - loss: 0.0105 - mae: 0.0822 - val_loss: 0.0106 - val_mae: 0.0967
Epoch 22/50	
4/4	1s 140ms/step - loss: 0.0118 - mae: 0.0933 - val_loss: 0.0268 - val_mae: 0.1599
Epoch 23/50	
4/4	1s 128ms/step - loss: 0.0131 - mae: 0.0974 - val_loss: 0.0020 - val_mae: 0.0361
Epoch 24/50	
4/4	1s 168ms/step - loss: 0.0079 - mae: 0.0701 - val_loss: 0.0021 - val_mae: 0.0365
Epoch 25/50	
4/4	2s 255ms/step - loss: 0.0091 - mae: 0.0770 - val_loss: 0.0127 - val_mae: 0.1072
Epoch 26/50	
4/4	1s 257ms/step - loss: 0.0099 - mae: 0.0846 - val_loss: 0.0087 - val_mae: 0.0862
Epoch 27/50	
4/4	1s 150ms/step - loss: 0.0092 - mae: 0.0821 - val_loss: 0.0026 - val_mae: 0.0428

Gambar III.11 (a) Model Training

Epoch 27/50	4/4	1s 150ms/step	- loss: 0.0092 - mae: 0.0821 - val_loss: 0.0026 - val_mae: 0.0428
Epoch 28/50	4/4	1s 140ms/step	- loss: 0.0082 - mae: 0.0698 - val_loss: 0.0030 - val_mae: 0.0461
Epoch 29/50	4/4	1s 131ms/step	- loss: 0.0075 - mae: 0.0710 - val_loss: 0.0098 - val_mae: 0.0923
Epoch 30/50	4/4	1s 134ms/step	- loss: 0.0076 - mae: 0.0736 - val_loss: 0.0014 - val_mae: 0.0281
Epoch 31/50	4/4	1s 130ms/step	- loss: 0.0099 - mae: 0.0764 - val_loss: 0.0022 - val_mae: 0.0373
Epoch 32/50	4/4	1s 126ms/step	- loss: 0.0086 - mae: 0.0771 - val_loss: 0.0088 - val_mae: 0.0867
Epoch 33/50	4/4	1s 131ms/step	- loss: 0.0086 - mae: 0.0793 - val_loss: 0.0069 - val_mae: 0.0750
Epoch 34/50	4/4	1s 133ms/step	- loss: 0.0073 - mae: 0.0730 - val_loss: 0.0065 - val_mae: 0.0718
Epoch 35/50	4/4	1s 132ms/step	- loss: 0.0085 - mae: 0.0777 - val_loss: 0.0109 - val_mae: 0.0978
Epoch 36/50	4/4	1s 138ms/step	- loss: 0.0069 - mae: 0.0685 - val_loss: 0.0057 - val_mae: 0.0666
Epoch 37/50	4/4	1s 132ms/step	- loss: 0.0077 - mae: 0.0733 - val_loss: 0.0045 - val_mae: 0.0585
Epoch 38/50	4/4	1s 126ms/step	- loss: 0.0083 - mae: 0.0705 - val_loss: 0.0023 - val_mae: 0.0388
Epoch 39/50	4/4	1s 132ms/step	- loss: 0.0080 - mae: 0.0740 - val_loss: 0.0117 - val_mae: 0.1018
Epoch 40/50	4/4	1s 136ms/step	- loss: 0.0083 - mae: 0.0774 - val_loss: 0.0020 - val_mae: 0.0346
Epoch 41/50	4/4	1s 128ms/step	- loss: 0.0109 - mae: 0.0833 - val_loss: 0.0028 - val_mae: 0.0448
Epoch 42/50	4/4	1s 136ms/step	- loss: 0.0077 - mae: 0.0707 - val_loss: 0.0216 - val_mae: 0.1426
Epoch 43/50	4/4	1s 130ms/step	- loss: 0.0091 - mae: 0.0800 - val_loss: 0.0036 - val_mae: 0.0520
Epoch 44/50	4/4	1s 170ms/step	- loss: 0.0078 - mae: 0.0723 - val_loss: 0.0013 - val_mae: 0.0280
Epoch 45/50	4/4	1s 237ms/step	- loss: 0.0089 - mae: 0.0735 - val_loss: 0.0066 - val_mae: 0.0731
Epoch 46/50	4/4	1s 259ms/step	- loss: 0.0082 - mae: 0.0760 - val_loss: 0.0068 - val_mae: 0.0743
Epoch 47/50	4/4	1s 124ms/step	- loss: 0.0073 - mae: 0.0732 - val_loss: 0.0040 - val_mae: 0.0550
Epoch 48/50	4/4	1s 128ms/step	- loss: 0.0078 - mae: 0.0735 - val_loss: 0.0059 - val_mae: 0.0687
Epoch 49/50	4/4	1s 125ms/step	- loss: 0.0070 - mae: 0.0697 - val_loss: 0.0026 - val_mae: 0.0426
Epoch 50/50	4/4	1s 135ms/step	- loss: 0.0082 - mae: 0.0714 - val_loss: 0.0013 - val_mae: 0.0273

Gambar III.11 (b) Model Training

III.5 Evaluasi Model

```
# Evaluasi nilai loss dan mae model
test_loss, test_mae = model.evaluate(X_test, y_test)

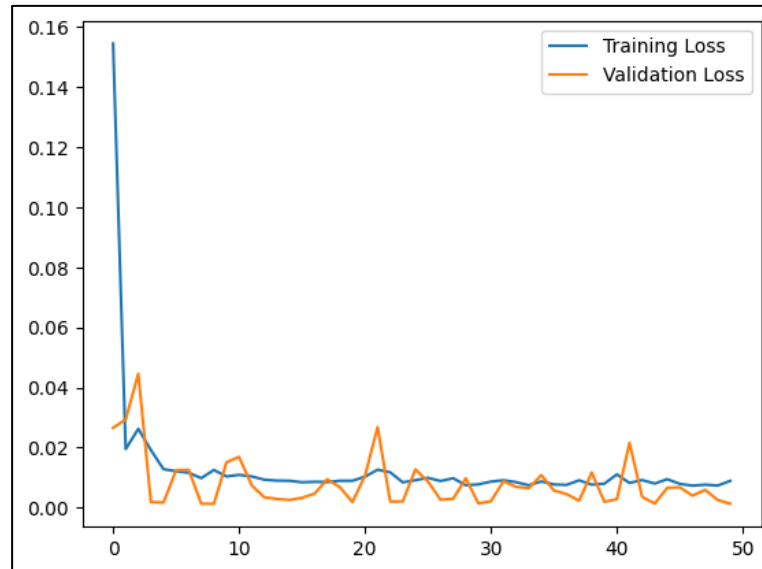
# Menampilkan nilai loss dan mae
print(f"Test Loss (MSE): {test_loss}")
print(f"Test MAE: {test_mae}")
```

```
2/2 ————— 0s 32ms/step - loss: 0.0060 - mae: 0.0622
Test Loss (MSE): 0.0057324920780956745
Test MAE: 0.0610673762857914
```

Gambar III.12 Hasil Evaluasi Model

```
# Visualisasi performa model

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.show()
```



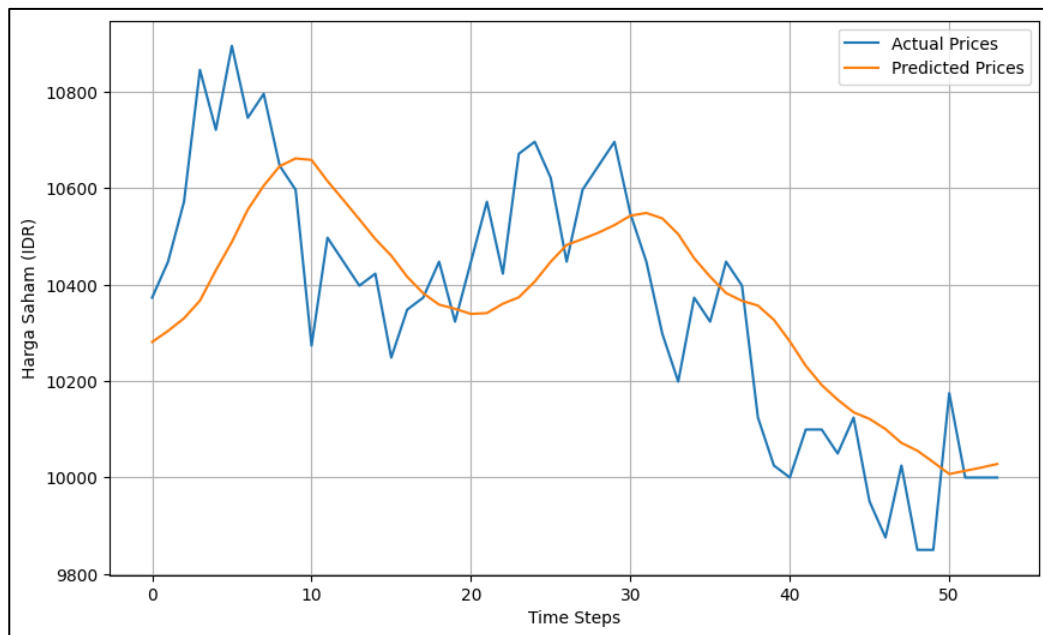
Gambar III.13 Visualisasi Performa Model

III.6 Prediksi Pada Data Testing

```
# Melakukan prediksi
prediksi = model.predict(X_test)

# Mengembalikan ke skala awal
hasil_prediksi = scaler.inverse_transform(prediksi)
hasil_aktual = scaler.inverse_transform(y_test.reshape(-1,
1))

# Visualisasi prediksi harga saham dan harga saham sebenarnya
plt.figure(figsize=(10, 6))
plt.plot(hasil_aktual, label='Actual Prices')
plt.plot(hasil_prediksi, label='Predicted Prices')
plt.xlabel('Time Steps')
plt.ylabel('Harga Saham (IDR)')
plt.legend()
plt.grid()
plt.show()
```



Gambar III.14 Perbandingan Hasil Prediksi dan Data Aktual

III.7 Menentukan Akurasi Model

```
# Menentukan Mean Absolute Error
errors = np.abs(hasil_prediksi - hasil_aktual)
MAE = np.mean(errors)
print(f"Mean Absolute Error: {MAE:.2f}")

# Menentukan Mean Absolute Error Relatif
MAE_relatif = (MAE / data_harian.mean().item()) * 100
print(f"Mean Absoulter Error Relatif: {MAE_relatif:.2f}%")
```

```
Mean Absolute Error: 150.11
Mean Absoulter Error Relatif: 1.53%
```

Gambar III.15 Output Nilai MAE

```
# Menentukan Mean Absolute Percentage Error (MAPE)

# Memastikan data berbentuk array 1D
actual = hasil_aktual.flatten()
predicted = hasil_prediksi.flatten()

# Menghitung MAPE
mape = np.mean(np.abs((actual - predicted) / actual)) * 100
```

```
print(f"MAPE: {mape:.2f}%")
```



Gambar III.16 Output Nilai MAPE

III.8 Menyimpan Model

```
# Menyimpan model
model.save('LSTM_model.h5')
```

III.9 Prediksi Harga Saham Beberapa Hari Akan Datang

```
# Memuat model yang telah dilatih
from tensorflow.keras.models import load_model

model_path = "/content/LSTM_model.h5"
trained_model = load_model(model_path)

# Mengambil data saham terbaru
import yfinance as yf
ticker = 'BBCA.JK'
data_terbaru = yf.download(ticker, start='2024-10-4',
end='2025-1-4', interval='1d')['Adj Close']

# Normalisasi data terbaru
from sklearn.preprocessing import MinMaxScaler

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

# Mengambil data terakhir berdasarkan window size (60 hari
terakhir)
window_size = 60
X_input = normalisasi_data_terbaru[-window_size:].reshape(1,
window_size, 1)

# Prediksi beberapa hari ke depan
import numpy as np

prediksi_saham = []
for i in range(5):      # 5 hari
    prediksi_harga = trained_model.predict(X_input)
```



```

# Menambahkan hasil prediksi ke dalam list
prediksi_saham.append(prediksi_harga[0][0])

# Memperbarui input untuk prediksi berikutnya dengan
menambahkan hasil prediksi terakhir
X_input = np.append(X_input[:, 1:, :],
prediksi_harga.reshape(1, 1, 1), axis=1)

# Mengembalikan hasil prediksi ke skala asli
prediksi_stock =
scaler.inverse_transform(np.array(prediksi_saham).reshape(-1,
1))

# Menampilkan hasil prediksi untuk beberapa hari ke depan
for i, pred in enumerate(prediksi_stock):
    print(f"Prediksi harga saham untuk hari ke-{i+1}:
{pred[0]:.2f} IDR")

```

```

1/1 ————— 0s 46ms/step
1/1 ————— 0s 44ms/step
1/1 ————— 0s 39ms/step
1/1 ————— 0s 37ms/step
1/1 ————— 0s 38ms/step
Prediksi harga saham untuk hari ke-1: 9966.44 IDR
Prediksi harga saham untuk hari ke-2: 9988.87 IDR
Prediksi harga saham untuk hari ke-3: 10011.34 IDR
Prediksi harga saham untuk hari ke-4: 10033.59 IDR
Prediksi harga saham untuk hari ke-5: 10055.45 IDR

```

Gambar III.17 Prediksi Harga Saham 5 Hari Ke Depan

Bab IV Analisis

IV.1 Perbandingan Hasil Prediksi dan Data Aktual

Hasil analisis saham yang didapatkan berupa grafik yang membandingkan antara data aktual harga saham dan harga prediksi. Berdasarkan grafik yang dihasilkan, terlihat grafik menunjukkan bahwa model LSTM berhasil menangkap pola pergerakan harga saham secara umum. Dapat dilihat pada Gambar III.14 bahwa grafik berwarna orange yang merupakan grafik prediksi mampu memprediksi kenaikan harga saham di awal dan penurunan secara bertahap hingga akhir periode. Namun, terdapat beberapa deviasi antara grafik prediksi dan grafik data aktual terutama pada titik-titik puncak dan lembah pergerakan harga saham. Hal ini menunjukkan bahwa meskipun model LSTM mampu menangkap pola pergerakan utama, namun model cenderung mengalami keterlambatan dalam melakukan prediksi perubahan harga. Perbedaan antara hasil prediksi dan data aktual dapat disebabkan oleh sensitivitas model terhadap volatilitas tinggi atau fluktuasi mendadak yang dipengaruhi oleh banyak faktor yang tidak dapat dijelaskan oleh pola historis.

Selain itu, pada gambar terlihat bahwa grafik prediksi yang dihasilkan lebih halus dibandingkan dengan data aktual. Hal ini dapat menjadi kelebihan karena ini menunjukkan bahwa model mampu mereduksi noise dalam data sehingga gambaran tren yang diberikan lebih jelas. Namun, hal ini juga dapat menjadi kelemahan karena fluktuasi harga saham harian yang signifikan tidak dapat terprediksi dengan akurat oleh model.

Pengaruh parameter model terhadap fenomena prediksi harga saham menjadi faktor penting dalam analisis. Parameter tersebut diantaranya jumlah neurons pada setiap lapisan LSTM, *learning rate*. Jumlah epoch, dan ukuran *window time series* yang memberikan peran dalam menentukan kualitas prediksi. Contohnya pengaruh jumlah *epoch*, ketika jumlah epoch terlalu sedikit maka model akan mengalami *underfitting*, sedangkan jumlah epoch yang terlalu banyak berpotensi menyebabkan *overfitting*.

IV.2 Evaluasi Nilai MAE dan MAPE

MAE (*Mean Absolute Error*) adalah salah satu metrik evaluasi dalam machine learning yang digunakan untuk mengukur seberapa dekat prediksi model dengan nilai aktual. Metrik ini menghitung rata-rata dari nilai absolut selisih antara prediksi dan nilai sebenarnya, tanpa mempertimbangkan arah kesalahan (positif atau negatif). MAE memberikan ukuran rata-rata dari kesalahan prediksi dalam skala yang sama dengan variabel yang diamati, dan semakin rendah nilai MAE, semakin baik modelnya dalam melakukan prediksi.

Sedangkan MAPE (*Mean Absolute Percentage Error*) adalah metrik yang digunakan dalam analisis statistik untuk menilai akurasi model prediksi atau peramalan. Metrik ini memberikan gambaran tentang seberapa akurat prediksi model jika dibandingkan dengan nilai aktual yang sebenarnya. Nilai MAPE yang berada di bawah 10% menunjukkan bahwa kemampuan model peramalan sangat baik.

Berdasarkan pengolahan, hasil evaluasi performa model ditunjukkan pada Gambar III.15 dan Gambar III.16, yaitu nilai MAE sebesar 150,11, MAE Relatif sebesar 1,53%, dan MAPE sebesar 1,44%. Nilai MAE menunjukkan rata-rata kesalahan absolut model dalam memprediksi harga saham BCA sebesar 150.11 IDR. Nilai ini cukup kecil karena harga saham BCA berada pada kisaran 10.000 IDR. Kemudian nilai MAE relatif sebesar 1,53% menunjukkan bahwa rata-rata kesalahan absolut ini hanya sekitar 1,53% dari harga saham BCA sebenarnya, nilai yang kecil ini menunjukkan bahwa prediksi model sangat dekat dengan nilai sebenarnya secara relatif dan menjadi indikator akurasi tinggi untuk analisis ini.

Sedangkan nilai MAPE sebesar 1.44% menunjukkan bahwa rata-rata kesalahan relatif dari prediksi model adalah sekitar 1,44%. Hal tersebut berarti rata-rata perbedaan antara harga saham BCA yang diprediksi dengan harga saham BCA aktual adalah sekitar 1,44% dari nilai penjualan aktual. Angka ini berada dibawah 10% yang menunjukkan bahwa kemampuan model peramalan sangat baik dan menjadi indikator akurasi tinggi untuk analisis ini. Dengan nilai MAE ini, para investor dapat memperkirakan bahwa rata-rata kesalahan prediksi hanya sekitar

150 IDR dari harga saham BCA aktual, sehingga memberikan keyakinan lebih dalam memanfaatkan model ini untuk analisis dan pengambilan keputusan investasi.

IV.3 Prediksi Harga Saham 5 Hari ke Depan

Berdasarkan pengolahan, hasil prediksi harga saham untuk 5 hari ke depan ditunjukkan pada Gambar III.17, dengan nilai prediksi masing-masing: hari ke-1 sebesar 9966.44 IDR, hari ke-2 sebesar 9988.87 IDR, hari ke-3 sebesar 10011.34 IDR, hari ke-4 sebesar 10033.59 IDR, dan hari ke-5 sebesar 10055.45 IDR. Prediksi ini memperlihatkan tren yang stabil, dengan fluktuasi kecil antar hari, yang mencerminkan pola harga saham yang terjaga dan terkendali.

Pergerakan harga saham yang diprediksi konsisten dengan pola tren historis, yaitu stabil dengan sedikit kenaikan pada hari-hari terakhir. Hal ini menunjukkan bahwa model Long Short-Term Memory (LSTM) mampu menangkap pola umum dari data historis dan menghasilkan prediksi yang realistis. Dengan demikian, model ini dapat diandalkan untuk memberikan gambaran tentang tren jangka pendek, meskipun validasi lebih lanjut dengan data aktual tetap diperlukan untuk memastikan tingkat akurasi prediksi yang optimal.

IV.4 Manfaat untuk Investor dan Manajemen Risiko

Prediksi yang dihasilkan oleh model LSTM memiliki potensi besar untuk membantu investor dalam mengidentifikasi tren harga saham jangka pendek. Informasi ini memungkinkan investor untuk membuat keputusan yang lebih terinformasi, seperti menentukan waktu yang tepat untuk membeli atau menjual saham guna memaksimalkan keuntungan atau meminimalkan risiko kerugian. Dengan kemampuan model ini untuk mengantisipasi pergerakan harga saham secara akurat, investor dapat merancang strategi investasi yang lebih proaktif dan adaptif terhadap kondisi pasar.

Selain itu, model ini dapat menjadi alat penting bagi tim manajemen risiko perusahaan dalam memantau pergerakan harga saham dan merancang strategi mitigasi risiko yang lebih efektif. Sebagai contoh, prediksi yang dihasilkan oleh

model dapat membantu perusahaan mengidentifikasi potensi volatilitas pasar sebelum terjadi, sehingga memungkinkan mereka untuk mengambil tindakan antisipatif seperti menyesuaikan portofolio aset, melindungi nilai investasi, atau mengalokasikan sumber daya dengan lebih efisien.

Perusahaan juga dapat memanfaatkan prediksi ini untuk mengurangi kerugian akibat fluktuasi harga yang tidak terduga. Dengan prediksi yang lebih akurat, efisiensi dalam pengelolaan portofolio dapat ditingkatkan, baik melalui alokasi aset yang lebih strategis maupun melalui optimasi keputusan keuangan lainnya. Selain itu, prediksi ini memungkinkan tim manajemen risiko untuk membuat proyeksi yang lebih tepat terkait dampak perubahan pasar terhadap arus kas, profitabilitas, dan kinerja keuangan perusahaan secara keseluruhan. Dengan demikian, model ini tidak hanya berfungsi sebagai alat analisis, tetapi juga sebagai fondasi untuk mendukung pengambilan keputusan yang lebih cerdas dan terencana.

IV.5 Keandalan Model untuk Pengambilan Keputusan

Hasil evaluasi menunjukkan bahwa model LSTM memiliki nilai MAE dan MAPE yang rendah, yang menunjukkan tingkat akurasi yang baik. Namun, penting untuk memahami bahwa model ini hanya menangkap pola historis dari data dan tidak mempertimbangkan faktor eksternal yang bersifat dinamis, seperti perubahan kondisi ekonomi, kebijakan pemerintah, atau sentimen pasar. Oleh karena itu, meskipun model ini cukup andal dalam memberikan gambaran tren jangka pendek, keputusan investasi tetap harus didukung oleh analisis tambahan, seperti pemantauan berita pasar, indikator fundamental, dan data real-time. Model LSTM ini dapat digunakan sebagai alat pendukung, tetapi bukan satu-satunya dasar pengambilan keputusan. Kombinasi antara hasil prediksi model dengan data real-time dan analisis fundamental dapat memberikan hasil yang lebih holistik dan kontekstual, sehingga memungkinkan investor atau perusahaan untuk memitigasi risiko dan memaksimalkan potensi keuntungan.

Bab V Kesimpulan

Berdasarkan hasil simulasi, diperoleh kesimpulan sebagai berikut:

1. Penerapan algoritma *Long Short-Term Memory* (LSTM) dalam memprediksi harga saham PT Bank Central Asia (BCA) telah menunjukkan hasil yang signifikan. Dengan menggunakan data historis harga saham, model LSTM berhasil memproyeksikan pergerakan harga saham untuk periode tertentu. Hasil prediksi menunjukkan bahwa model mampu menangkap pola pergerakan harga saham secara umum, termasuk tren kenaikan dan penurunan yang stabil.
2. Evaluasi model menunjukkan bahwa nilai MAE sebesar 150,11 IDR mengindikasikan rata-rata kesalahan absolut yang relatif kecil dibandingkan harga saham aktual, sementara MAE relatif sebesar 1,53% menunjukkan bahwa kesalahan prediksi berada dalam kisaran yang sangat kecil. Selain itu, nilai MAPE sebesar 1,44% menunjukkan bahwa model memiliki kemampuan peramalan yang sangat baik, dengan rata-rata kesalahan relatif hanya 1,44% dari harga saham aktual. Nilai-nilai ini menunjukkan bahwa model LSTM mampu memberikan hasil prediksi yang akurat dalam konteks pergerakan harga saham.
3. Meskipun nilai MAE dan MAPE rendah, model tetap memiliki keterbatasan, yaitu tidak mempertimbangkan faktor eksternal yang bersifat dinamis, seperti perubahan kondisi ekonomi, kebijakan pemerintah, atau sentimen pasar, sehingga dapat memengaruhi hasil prediksi pada periode tertentu. Secara keseluruhan, model LSTM dapat digunakan sebagai alat bantu yang baik untuk memberikan panduan tren jangka pendek bagi investor dan perusahaan. Kombinasi dengan analisis fundamental dan indikator tambahan akan memberikan hasil yang lebih holistik dan mendukung pengambilan keputusan yang lebih baik.

DAFTAR PUSTAKA

- Fadilah, W. R. U., Agfiannisa, D., & Azhar, Y. (2020). Analisis Prediksi Harga Saham PT. Telekomunikasi Indonesia Menggunakan Metode Support Vector Machine. *Fountain of Informatics Journal*, 5(2), 45. <https://doi.org/10.21111/fij.v5i2.4449>.
- M. Abdul Dwiyanto Suyudi, E. C. Djamal, A. Maspupah. (2019). “Prediksi Harga Saham menggunakan Metode Recurrent Neural Network,” Jurusan Informatika, and F. Sains dan Informatika Universitas Jenderal Achmad Yani Cimahi, Semin. Nas. Apl. Teknol. Inf., pp. 1907–5022.
- Özkurt, N., Yıldırım, T., Yaşar Üniversitesi, Institute of Electrical and Electronics Engineers. Turkey Section., & Institute of Electrical and Electronics Engineers. (2019). 2019 Innovations in Intelligent Systems and Applications Conference (ASYU): proceedings: 31 October-2 November 2019, Izmir, Turkey.
- Sukma, T. (2022). Didominasi Generasi Muda, Investor Pasar Modal Indonesia Tembus 10,3 Juta. <https://www.idxchannel.com/market-news/didominasi-generasi-muda-investor-pasar-modal-indonesia-tembus-103-juta>.
- 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, doi: 10.23887/JANAPATI.V8I3.19139.

LAMPIRAN

Berikut merupakan link python code dari pemrograman yang dilakukan.

https://colab.research.google.com/drive/1EHLQCDwL2qBxuJcGYaIpY_vbT1aJrX?usp=sharing

Berikut merupakan hasil training model LSTM.

https://drive.google.com/file/d/1bw5uG6n2u_EZR3t1mwMwgqdEPHs7-BFf/view?usp=drive_link