

Perbandingan Model Gated Recurrent Unit dan Long Short-Term Memory Dalam Prediksi Harga Saham

Sofiana Ardi Lestari^{1*}, Vihi Atina², Herliyani Hasanah³

¹Sistem Informasi/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

^{1*}210101038@mhs.udb.ac.id

²Teknologi Rekayasa Perangkat
Lunak/Fakultas Ilmu Komputer

Universitas Duta Bangsa

²vihi_atina@udb.ac.id

³Sistem Informasi/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

³herliyani_hasanah@udb.ac.id

Abstrak— Mengikuti perkembangan ilmu perekonomian yang semakin meluas, pentingnya investasi di pasar modal kini semakin disadari oleh masyarakat. Ketidakstabilan dan risiko yang dialami bagi para investor menjadikan pertimbangan menyeluruh akan strategi dalam menghadapi pergerakan harga saham yang sulit ditebak. Penelitian ini menganalisa harga saham dengan membandingkan model terbaik pada PT. Telkom Indonesia Tbk. menggunakan algoritma Gated Recurrent Unit (GRU) dan Long Short-Term Memory (LSTM). Penelitian ini menggunakan JupyterLab sebagai tools mengeksekusi proses pengembangan model. Kedua model dievaluasi berdasarkan nilai metric Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), dan R^2 . Hasil pengujian menunjukkan bahwa model LSTM memiliki tingkat kesalahan minimal, dengan RMSE sebesar 57,41, MAE sebesar 42.76, R^2 sebesar 0.95, dan MAPE sebesar 0.02, yang menandakan model LSTM memberikan ketepatan prediksi yang lebih baik dibandingkan dengan model GRU.

Kata kunci— Prediksi Harga Saham, Gated Recurrent Unit, Long Short-Term Memory, Data Mining.

Abstract— Following developments in economic science, the importance of investing in the capital market is now increasingly recognized by the public. The instability and risks experienced by investors require comprehensive consideration of strategies for dealing with unpredictable stock price movements. This study analyses stock prices by comparing the best models at PT. Telkom Indonesia Tbk. using the *Gated Recurrent Unit* (GRU) and *Long Short-Term Memory* (LSTM) algorithms. This study uses *JupyterLab* as the tool to execute the model development process. Both models are evaluated based on the metric *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), and R^2 metrics. The test results show that the LSTM model has the lowest error rate, with an RMSE of 57,41, MAE of 42.76, R^2 of 0.95, and MAPE of 0.02, indicating that the LSTM model provides more accurate predictions compared to the GRU model.

Keywords— Stock Price Prediction, Gated Recurrent Unit, Long Short-Term Memory, Data Mining.

I. PENDAHULUAN

Sejalan dengan berkembangannya perekonomian suatu negara, banyak masyarakat yang paham akan pentingnya investasi. Salah satunya berinvestasi di pasar modal. Pasar modal merupakan suatu instrument keuangan jangka panjang yang didalamnya terdapat kegiatan jual beli saham serta tempat pendanaan bagi perusahaan dan kegiatan investasi[1].

Berbicara mengenai saham, saham merupakan bukti kepemilikan dari suatu perusahaan[2]. Maka dari itu, suatu perusahaan harus memiliki kinerja keuangan dengan kemampuan pengelolaan risiko pasar dan operasional stabil yang tidak hanya bergantung pada pendapatan dan profitabilitas, tetapi mampu mengukur berbagai risiko yang timbul baik dari fluktuasi pasar, perubahan regulasi, maupun sentimen pasar global[3]. Namun, karena minimnya

pengalaman investor dalam memilih saham yang berkualitas membuat mereka lebih rentan terhadap dampak volatilitas, sehingga beresiko mengalami kerugian pada pengambilan keputusan investasi[4].

Karena ketidakpastian pasar saham dan dominasi investor pemula yang kurang berpengalaman mengakibatkan risiko kerugian akibat pengambilan keputusan, dengan adanya penerapan teknologi *AI* diharapkan memberikan petunjuk terkait pertimbangan tujuan investasi dan risiko tentang arah pergerakan harga saham[3]. Melihat kondisi tersebut, salah satu strategi penting dalam menghadapi pergerakan harga saham yang sulit ditebak yaitu melakukan analisis prediksi berdasarkan data historis dengan memanfaatkan teknologi *deep learning* yang telah memungkinkan pengembangan model prediksi harga saham yang lebih akurat[5].

Pada penelitian sebelumnya, mengenai perbandingan prediksi harga saham yang pernah dilakukan oleh [6] Daffa menggunakan metode Long Short-Term Memory (LSTM) dan ARIMA (Studi Kasus: Alfamart dan Alfamidi) menghasilkan model dengan nilai RMSE terendah dan akurasi tertinggi sebesar 99.5% pada harga saham Alfamidi yang menjadikan metode LSTM sebagai pilihan yang lebih baik dibandingkan menggunakan metode ARIMA. Penelitian berikutnya menggunakan metode berbeda yaitu Linear Regression, KNN, dan SVM pernah dilakukan oleh [7] Siti dan Didih diperoleh hasil penelitian berupa nilai MSE, RMSE dan MAE dengan performa nilai MSE mendekati nol ($9.82e-27$), RMSE $9.91E-14$, dan MAE $8.71E-14$ dengan menggunakan model Linear Regression, dengan hasil ketepatan prediksi yang sangat baik dibandingkan KNN dan SVM. Penelitian lain yang membandingkan metode Long Short-Term Memory (LSTM) dan GRU dalam memprediksi harga saham Bank Rakyat Indonesia pernah dilakukan oleh [8] Yogi dkk yang menunjukkan model GRU memiliki kinerja terbaik dibandingkan model LSTM, dengan nilai R^2 sebesar 10.7% (dari 0.7615 menjadi 0.8429) dan penurunan MAPE sebesar 18.5% (dari 1.6886% menjadi 1.3764%). Berdasarkan penelitian yang telah dilakukan, terbukti model GRU lebih efektif menangkap pola dan memprediksi pergerakan harga saham dikarenakan arsitektur GRU lebih sederhana dan memiliki karakteristik noise tinggi dan volatilitas yang bervariasi[5]. Dan LSTM juga telah menunjukkan kemampuan dalam menangani masalah pemrosesan data *sequential* (berurutan) yang dapat digunakan di masa depan untuk menangkap pola-pola dan dependensi jangka panjang dalam data historis[9][1].

Penelitian ini hanya fokus pada perbandingan hasil evaluasi antara model GRU dan LSTM pada prediksi harga saham PT. Telkom Indonesia Tbk (TLKM). PT. Telkom Indonesia Tbk dipilih sebagai objek penelitian karena likuiditas tinggi terhadap indeks pasar serta perannya sebagai parameter pertumbuhan sektor telekomunikasi dan digital Indonesia. Pada penelitian ini, menggunakan data harga open, low, high, dan close saham TLKM selama periode Oktober 2020 hingga April 2025. Penelitian ini bertujuan untuk melakukan perbandingan model GRU dan LSTM yang memperoleh hasil minim nilai

error pada nilai metric yaitu *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), dan R^2 .

II. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode CRISP-DM (*Cross Industry Standard for Data Mining*) untuk melakukan pengembangan model prediksi harga saham. Adapun tahapan pelaksanaan penelitian yaitu:

A. Business Understanding

Tahap pertama adalah mempresentasikan dan menggambarkan data sebagai objek bisnis yang dapat dijadikan deskripsi bisnis dan menentukan tujuan penelitian[10].

B. Data Understanding

Tahap berikutnya adalah mengumpulkan data dan mengidentifikasi isi. Pada tahap ini menjadi fondasi untuk sebuah bisnis dapat diidentifikasi potensi masalah yang akan ditimbulkan kedepannya. Pada tahap ini, sumber data dikumpulkan selama kurang lebih 5 tahun terakhir berupa dataset harga saham PT. Telkom Indonesia Tbk (TLKM).

C. Data Preparation

Pada tahap selanjutnya yaitu memperbaiki masalah yang ada didalam data dan memastikan data tersebut cukup matang untuk pembangunan model[11][12]. Pada proses ini data yang berisi harga penjualan saham yang tidak terstruktur diubah menjadi format terstruktur agar dapat diolah dan diproses oleh komputer. Persiapan tahap ini meliputi langkah-langkah penting, seperti melakukan *cleanning* data dan membagi variabel pada dataset dari keseluruhan data yang akan digunakan.

Tabel 1. Pembagian Dataset

Data	Persentase
Train	64%
Test	20%
Validation	16%

D. Data Modelling

Tahap berikutnya dilakukan pembuatan model prediktif atau deskriptif dengan menggunakan algoritma *Gated Recurrent Unit* dan *Long Short-Term Memory*. Pada pengembangan kedua model ini, dilakukan menggunakan bahasa pemrograman Python dan dieksekusi menggunakan platform JupyterLab.

E. Evaluation

Pada tahapan ini yaitu tahap dimana hasil output dinilai serta dilaporkan sesuai kebutuhan dan tujuan penelitian. Tahapan ini memanfaatkan data uji sebanyak 20% yang digunakan untuk mengukur nilai metric sebagai hasil akhir pengujian performa model. Dalam tahap ini dihasilkan nilai metric berupa RMSE, MAPE, MAE, dan R². Langkah ini dipakai guna mengevaluasi kesesuaian hasil prediksi dengan harga actual pada saham TLKM pada hari berikutnya.

F. Deployment

Tahap terakhir adalah perencanaan penggunaan model dari awal proses *business understanding* sampai *evaluation* selesai di analisis. Pada proses ini digambarkan dengan hasil pembuatan laporan atau jurnal publikasi.

III. HASIL DAN PEMBAHASAN

A. Bussiness Understanding

Pada tahapan ini, permasalahan yang melatarbelakangi adalah sulitnya memahami dan mengurangi resiko para investor pemula dalam mempertimbangkan pergerakan harga saham PT. Telkom Indonesia Tbk (TLKM) di masa depan. Selain itu, investor perlu mempresentasikan dan bersiap untuk pengambilan keputusan bisnis, apakah harus melakukan transaksi jual-beli pada saham TLKM karena fluktuasi harga saham yang sangat berisiko.

B. Data Understanding

Pada tahapan ini dilakukan proses pengumpulan data keseluruhan yang menjadi fondasi dalam menentukan jenis data yang dikumpulkan, sumber data, dan metode pengumpulan yang digunakan. Data yang digunakan mencakup Date, OpenPrice, High, Low, Close dengan rentang waktu 1 Oktober 2020 hingga 30 April 2025. Deskripsi dari fitur data tersebut disajikan dalam table 2.

Tabel 2. Deskripsi Fitur Data

Fitur	Indikator	Tipe Data
Date	Tanggal transaksi saham	Datetime
OpenPrice	Harga awal saham di tanggal tersebut	Integer
High	Harga tertinggi saham di tanggal tersebut	Integer
Low	Harga terendah saham di tanggal tersebut	Integer

Close	Harga penutupan saham di tanggal tersebut	Integer
Date	Tanggal transaksi saham	Datetime

C. Data Preparation

Menyiapkan keseluruhan kumpulan data yang akan digunakan menjadi dataset yang siap diolah ketahap berikutnya. Pada penelitian ini, memanfaatkan 1103 baris data dengan dataset harian harga saham TLKM. Dataset ini diambil langsung dari API IDX dalam format JSON yang dieksekusi oleh JupyterLab menggunakan *scraper*.

	OpenPrice	High	Low	Close
Date				
2020-10-01	2600.0	2760.0	2600.0	2750.0
2020-10-02	2750.0	2750.0	2650.0	2680.0
2020-10-05	2690.0	2710.0	2640.0	2650.0
2020-10-06	2680.0	2700.0	2620.0	2650.0
2020-10-07	2650.0	2670.0	2620.0	2660.0
...	...	...	...	...
2025-04-24	2630.0	2650.0	2600.0	2630.0
2025-04-25	2630.0	2650.0	2570.0	2600.0
2025-04-28	2600.0	2610.0	2560.0	2560.0
2025-04-29	2570.0	2600.0	2560.0	2590.0
2025-04-30	2600.0	2640.0	2600.0	2640.0

1103 rows × 4 columns

Gambar 1. Tampilan Dataset Saham TLKM

Setelah dilakukan pengambilan data, berikutnya dilakukan proses preprocessing data. Berikut merupakan code preprocessing data:

```

from sklearn.preprocessing import MinMaxScaler
import numpy as np

# Constants
DATA_LOOKBACK = 90
VALIDATION_DATA_SIZE = 0.2
COLLECTED_DATA = ['OpenPrice', 'High', 'Low', 'Close']

# Create dataset function
def create_multivariate_dataset(data, look_back=1):
    X, y = [], []
    for i in range(len(data) - look_back):
        X.append(data[i:i+look_back])
        y.append(data[i+look_back])
    return np.array(X), np.array(y)

# Cleaning data
used_data = data_frame[COLLECTED_DATA].astype('float64').dropna()

# Normalize data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(used_data.values)

# Create dataset for multivariate prediction
X, y = create_multivariate_dataset(scaled_data, DATA_LOOKBACK)

# Split data
test_size = int(VALIDATION_DATA_SIZE * len(X))
val_size = int(VALIDATION_DATA_SIZE * (len(X) - test_size))

X_train = X[:test_size - val_size]
y_train = y[:test_size - val_size]

X_val = X[-test_size - val_size::test_size]
y_val = y[-test_size - val_size::test_size]

X_test = X[-test_size:]
y_test = y[-test_size:]

print(f"X_train shape: {X_train.shape}, y_train shape: {y_train.shape}")
print(f"X_test shape: {X_test.shape}, y_test shape: {y_test.shape}")
print(f"X_val shape: {X_val.shape}, y_val shape: {y_val.shape}")

X_train shape: (649, 90, 4), y_train shape: (649, 4)
X_test shape: (202, 90, 4), y_test shape: (202, 4)
X_val shape: (162, 90, 4), y_val shape: (162, 4)

```

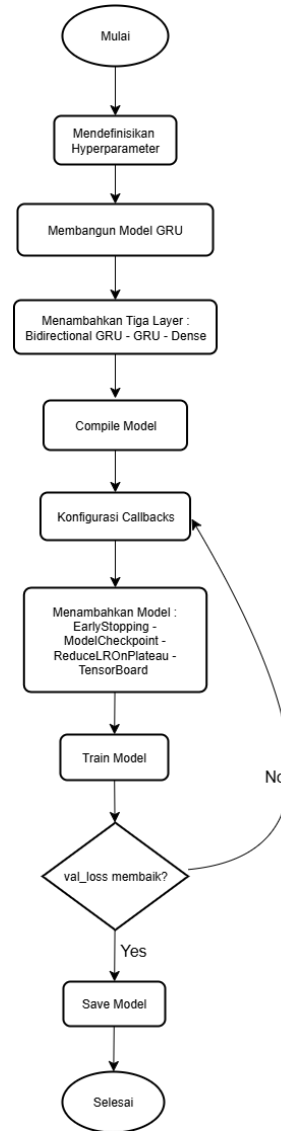
Gambar 2. Jumlah Data Preprocessing

Proses selanjutnya adalah tahap *preprocessing data*. Pada proses ini dilakukan cleaning data dengan fungsi *dropna*. Berikutnya, data dibagi menjadi dua variabel yaitu *x* dan *y*. Setiap variabel akan dibagi lagi menjadi tiga yaitu *train*, *test*, dan *validation*. Data uji diambil sebanyak 20% dari total data. Hasil pembagian variabel data *x* dan *y* diluar dari *DATA_LOOKBACK* 90 hari kedepan yang digunakan untuk menentukan hasil prediksi. Pada tahapan ini merupakan tahapan awal sebelum mengimplementasi model *Deep Learning* pada saham TLKM.

D. Data Modelling

Tahapan selanjutnya dilakukan modelling menggunakan dua jenis model jaringan saraf tiruan, yaitu *Gated Recurrent Unit* (GRU) dan *Long Short-Term Memory* (LSTM). Pada penelitian ini, menggunakan model *sequential* karena

mekanismenya dapat menggunakan banyak layers dan gates.



Gambar 3. Flowchart GRU

Berikut code yang digunakan untuk modelling data menggunakan metode GRU:

```

# Constants
EPOCH_MODEL_SIZE = 256
BATCH_SIZE_MODEL = 16
MODEL_CHECKPOINT_NAME = 'best_model2'
MODEL_CHECKPOINT_TYPE = 'keras.weights.h5'
TRAIN_DATA_SIZE = 0.8
VALIDATION_DATA_SIZE = 0.2
DEFAULT_UNIT_SIZE = [64, 32, 16, 8]
SHOW_VERBOSE = 0 # 0 to hide logs, 1 to show logs
INITIAL_LEARNING_RATE = 0.001
PATIENCE_TIME = 20

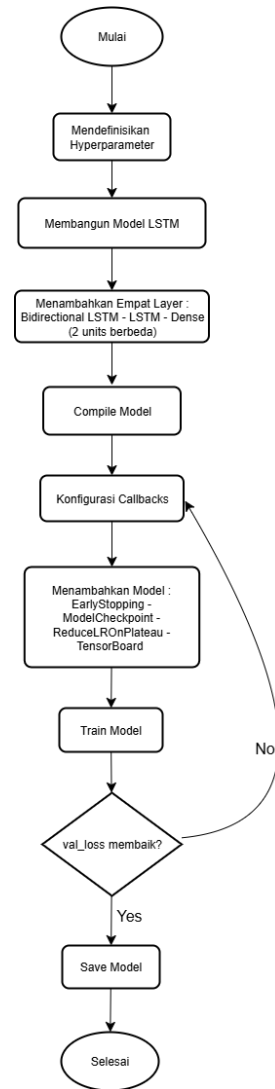
# Model Function
def create_gru_model(input_shape, units=DEFAULT_UNIT_SIZE,
dropout_rate=DROPOUT_RATE):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(Bidirectional(GRU(units=units[0], return_sequences=True)))
    model.add(GRU(units=units[0], return_sequences=False))
    model.add(Dense(len(COLLECTED_DATA)))
    optimiser = SGD(
        learning_rate=INITIAL_LEARNING_RATE,
        decay=1e-7, momentum=0.9, nesterov=False)
    model.compile(
        optimiser=optimiser, loss="mean_squared_error",
        metrics=['mae', 'mse'])
    return model

# Callbacks
def model_callbacks():
    return [
        EarlyStopping(
            monitor='val_loss', patience=PATIENCE_TIME,
            restore_best_weights=True, mode='min', verbose=SHOW_VERBOSE),
        ModelCheckpoint(
            f"{MODEL_CHECKPOINT_NAME}.{MODEL_CHECKPOINT_TYPE}",
            monitor='val_loss', save_best_only=True, mode='min',
            verbose=SHOW_VERBOSE, save_weights_only=True),
        ReduceLROnPlateau(
            monitor='val_loss', patience=PATIENCE_TIME,
            verbose=SHOW_VERBOSE),
        TensorBoard(log_dir='./logs')]

# Implement model function
gru_model = create_gru_model((DATA_LOOKBACK, len(COLLECTED_DATA)))
print("Fitting data using GRU Model")
gru_history = gru_model.fit(
    X_train, y_train, epochs=EPOCH_MODEL_SIZE,
    batch_size=BATCH_SIZE_MODEL, verbose=SHOW_VERBOSE,
    validation_data=(X_val, y_val), callbacks=model_callbacks()
)

```

Pada model ini menggunakan tiga lapisan layer untuk pelatihan model GRU secara berurutan, dengan model Bidirectional, GRU dan berakhir dengan output layer yaitu Dense.



Gambar 4. Flowchart LSTM

Berikut code yang digunakan untuk modelling data menggunakan metode LSTM:

```
# Constants
EPOCH_MODEL_SIZE = 256
BATCH_SIZE_MODEL = 16
MODEL_CHECKPOINT_NAME = 'best_model2'
MODEL_CHECKPOINT_TYPE = 'keras.weights.h5'
TRAIN_DATA_SIZE = 0.8
VALIDATION_DATA_SIZE = 0.2
DEFAULT_UNIT_SIZE = [64, 32, 16, 8]
SHOW_VERBOSE = 0 # 0 to hide logs, 1 to show logs
INITIAL_LEARNING_RATE = 0.001
PATIENCE_TIME = 20

# Model Function
def create_lstm_model(input_shape, units=DEFAULT_UNIT_SIZE, dropout_rate=DROPOUT_RATE):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(Bidirectional(LSTM(units=units[0], return_sequences=True)))
    model.add(LSTM(units=units[1], return_sequences=False))
    model.add(Dense(units[2]))
    model.add(Dense(len(COLLECTED_DATA)))
    optimiser = Adam(
        learning_rate=INITIAL_LEARNING_RATE,
        beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=True, clipnorm=1.0,
    )
    model.compile(
        optimiser=optimiser, loss="mean_squared_error",
        metrics=['mae', 'mse'])
    return model

# Callbacks
def model_callbacks():
    return [
        EarlyStopping(
            monitor='val_loss', patience=PATIENCE_TIME,
            restore_best_weights=True, mode='min', verbose=SHOW_VERBOSE),
        ModelCheckpoint(
            f"{MODEL_CHECKPOINT_NAME}.{MODEL_CHECKPOINT_TYPE}",
            monitor='val_loss', save_best_only=True, mode='min',
            verbose=SHOW_VERBOSE, save_weights_only=True),
        ReduceLROnPlateau(
            monitor='val_loss', patience=PATIENCE_TIME,
            verbose=SHOW_VERBOSE),
        TensorBoard(log_dir='./logs')]

# Implement model function
lstm_model = create_lstm_model((DATA_LOOKBACK, len(COLLECTED_DATA)))
print("Fitting data using LSTM Model")
lstm_history = lstm_model.fit(X_train, y_train, epochs=EPOCH_MODEL_SIZE,
    batch_size=BATCH_SIZE_MODEL, verbose=SHOW_VERBOSE,
    validation_data=(X_val, y_val), callbacks=model_callbacks())
```

Selanjutnya model ini menggunakan empat lapisan layer untuk pelatihan model LSTM secara berurutan, dengan model Bidirectional, LSTM dan berakhir dengan dua output layer Dense dengan units yang berbeda.

Berikutnya setiap model diinisialisasi dan dilatih dengan metode fit menggunakan data latih x_train dan y_train, lalu tiap model ditambahkan hyperparameter epochs=256, batch_size=16 dan learning_rate=0.001 untuk menyeimbangkan kecepatan pelatihan dan akurasi, sementara callbacks seperti EarlyStopping, ModelCheckpoint, ReduceLROnPlateau, dan TensorBoard digunakan untuk mengoptimalkan proses pelatihan model.

E. Evaluation

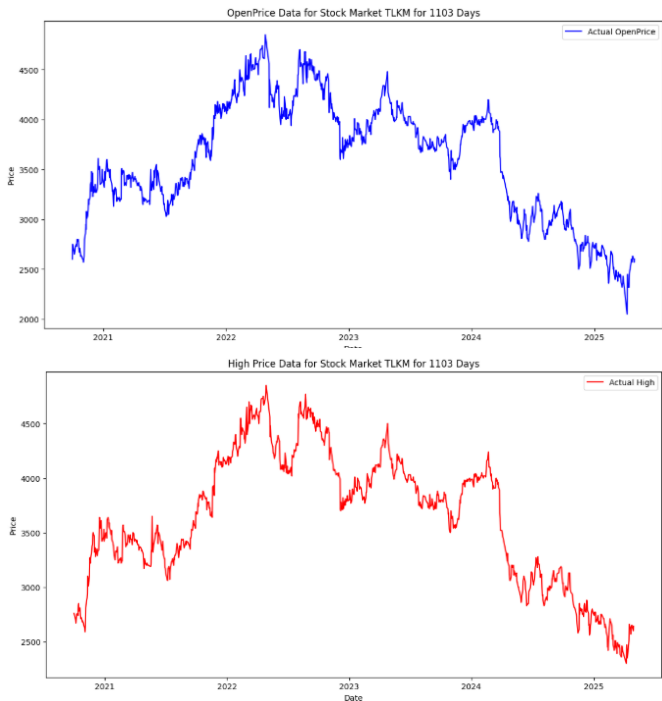
Tahapan terakhir pada penelitian ini yaitu melakukan evaluasi menggunakan perbandingan dua model algoritma *Gated Recurrent Unit* (GRU) dan *Long Short-Term Memory* (LSTM). Hasil pengujian

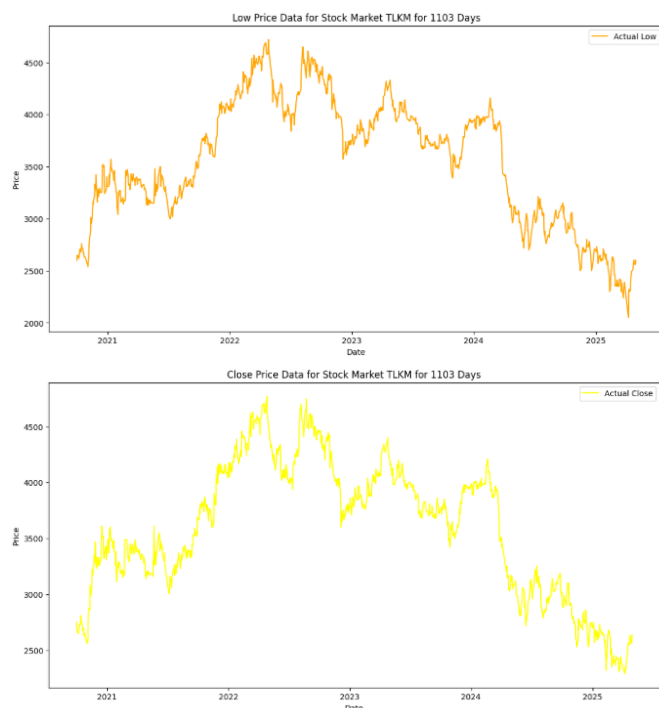
performa model melibatkan empat nilai metric yaitu *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE), dan R^2 .

Tabel 3. Nilai Perbandingan

Algoritma	Nilai Metric	Open Price	High	Low	CLOSE
Gated Recurrent Unit (GRU)	RMSE	69.08	81.97	76.71	90.11
	MAE	54.2	64.02	63.81	71.97
	R ²	0.92	0.88	0.91	0.87
	MAPE	0.02	0.02	0.02	0.02
Long Short-Term Memory (LSTM)	RMSE	57.41	61.44	65.53	75.41
	MAE	42.76	48.52	51.35	58.99
	R ²	0.95	0.94	0.93	0.91
	MAPE	0.02	0.02	0.02	0.02

Nilai perbandingan performa model pada table 3 diatas, menghasilkan nilai metric antara dua model GRU dan LSTM yang menghasilkan tingkat error terendah pada model LSTM sebesar 57.41 untuk harga OpenPrice, 61.44 untuk harga High, 65.53 untuk harga Low, dan 75.41 untuk harga Close pada emiten saham TLKM. Nilai metric tersebut cukup baik karena tingkat error yang cukup rendah dibandingkan dengan model GRU.





Gambar 5. Grafik harga saham TLKM Actual

GRU Predictions:				
	OpenPrice	High	Low	Close
2025-05-01	2562.9	2585.6	2519.4	2546.4

LSTM Predictions:				
	OpenPrice	High	Low	Close
2025-05-01	2605.1	2654.5	2556.8	2640.8

Gambar 6. Prediksi harga saham TKLM di hari berikutnya

Pada gambar 5 menampilkan grafik informasi terkait harga actual dan gambar 6 menampilkan prediksi harga satu hari berikutnya. Berdasarkan nilai metric pada table 3 ditemukan model LSTM memiliki tingkat error terendah dibandingkan GRU, yang ditunjukkan pada penelitian sebelumnya oleh Yogi dkk.[8] melakukan perbandingan model Gated Recurrent Unit (GRU) dan LSTM dengan hasil evaluasi bahwa model GRU memiliki kinerja terbaik dibandingkan model LSTM, dengan nilai R^2 sebesar 10.7% (dari 0.7615 menjadi 0.8429) dan penurunan MAPE sebesar 18.5% (dari 1.6886% menjadi 1.3764%), sedangkan pada penelitian ini juga membandingkan model LSTM dan GRU, LSTM memiliki kinerja yang lebih baik dibanding GRU dengan nilai R^2 tertinggi 0.95 dan untuk MAPE 0.02. Penelitian lainnya dilakukan oleh I Gusti dkk.[12] menggunakan algoritma Random Forest dengan nilai RMSE terendah 243.63, MAE terendah 208.57,

R^2 tertinggi 0.96 dan MAPE terendah 5.89%, hasil menunjukkan bahwa pada penelitian ini LSTM memiliki tingkat error lebih rendah (RMSE) sebesar 57.41 untuk harga OpenPrice saham TLKM.

IV. KESIMPULAN

Berdasarkan pembahasan dan hasil penelitian yang didapatkan, dapat disimpulkan antara dua perbandingan nilai metric yaitu model *Gated Recurrent Unit* (GRU) dan *Long Short-Term Memory* (LSTM) pada saham emiten TLKM. Pada kedua model tersebut memiliki nilai RMSE terendah sebesar 57.41 untuk harga OpenPrice dan nilai tertinggi sebesar 90.11 untuk harga Close. Adapun nilai metric lainnya yaitu nilai MAE terendah sebesar 42.76 untuk harga OpenPrice dan tertinggi sebesar 71.97 untuk harga Close, nilai koefisien determinasi (R^2) tertinggi sebesar 0.95 untuk harga OpenPrice dan terendah sebesar 0.87 untuk harga Close, serta untuk nilai MAPE memiliki nilai yang sama yaitu 0.02. Dari indicator nilai metric membuktikan kinerja model Long Short-Term Memory (LSTM) sangat baik dengan tingkat akurasi model 95%-91% dengan risiko kesalahan minimal dibandingkan dengan model Gated Recurrent Unit (GRU).

REFERENSI

- [1] I. K. A. Enriko, F. N. Gustiyana, and R. H. Putra, "Komparasi Hasil Optimasi Pada Prediksi Harga Saham PT. Telkom Indonesia Menggunakan Algoritma Long Short Term Memory," *J. Media Inform. Budidarma*, vol. 7, no. 2, p. 659, 2023, doi: 10.30865/mib.v7i2.5822.
- [2] R. N. Silalahi and M. Muljono, "Perbandingan Kinerja Metode Linear Regression, LSTM dan GRU Untuk Prediksi Harga Penutupan Saham Coca-Cola," *Komputika J. Sist. Komput.*, vol. 13, no. 2, pp. 201–211, 2024, doi: 10.34010/komputika.v13i2.12265.
- [3] Nabilah Selayanti, Dwi Amalia Putri, Trimono Trimono, and Mohammad Idhom, "Prediksi Harga Penutupan Saham Bbri Dengan Model Hybrid Lstm-Xgboost," *Inform. J. Tek. Inform. dan Multimed.*, vol. 5, no. 1, pp. 52–64, 2025, doi: 10.51903/informatika.v5i1.1011.
- [4] F. A. Trimono, "Model ARMA-GARCH Prediksi Value-at-Risk pada Saham," vol. 2, pp. 116–121, 2021.
- [5] N. W. Saputra, F. Insani, S. Agustian, and S. Sanjaya, "Penerapan Deep Learning Menggunakan Gated Recurrent Unit Untuk Memprediksi Harga Minyak Mentah Dunia," *Build. Informatics, Technol. Sci.*, vol. 5, no. 1, 2023, doi: 10.47065/bits.v5i1.3552.
- [6] D. K. H. Putra, "Program studi sistem informasi depok agustus 2024," 2024.
- [7] S. Mulia and D. Gustian, "Analisis Kinerja Algoritma Prediksi Saham pada PT GoTo Gojek Tokopedia Tbk (GOTO)," vol. 11, no. 1, pp. 53–67, 2025.
- [8] Y. Perdana *et al.*, "Analisis Perbandingan Model Gru Dan Lstm Untuk," pp. 54–60, 2021.

- [9] R. A. Hamzar and F. A. Setyaningsih, "Dampak Ukuran Dataset Pelatihan Terhadap Performa LSTM Network dalam Konteks Harga Saham," *Coding J. Komput. dan Apl.*, vol. 11, no. 2, p. 238, 2023, doi: 10.26418/coding.v11i2.62142.
- [10] J. Brzozowska, J. Pizoń, G. Baytikenova, A. Gola, A. Zakimova, and K. Piotrowska, "Data Engineering in Crisp-Dm Process Production Data – Case Study," *Appl. Comput. Sci.*, vol. 19, no. 3, pp. 83–95, 2023, doi: 10.35784/acs-2023-26.
- [11] V. Plotnikova, M. Dumas, A. Nolte, and F. Milani, "Designing a data mining process for the financial services domain," *J. Bus. Anal.*, vol. 6, no. 2, pp. 140–166, 2023, doi: 10.1080/2573234X.2022.2088412.
- [12] G. Arya *et al.*, "Analisis Pengaruh Rasio Keuangan Perbankan Terhadap Harga Saham Menggunakan Algoritma Random Forest," *Dangin Puri Klod, Kec. Denpasar Tim.*, vol. 12, no. 2, p. 244445, 2025, [Online]. Available: <http://jurnal.mdp.ac.id>