

**Perbandingan Penerapan Metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average* dalam
Prediksi Harga Saham**

TUGAS AKHIR

**Yoel Agustinus
1118042**



**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
2022**

**Perbandingan Penerapan Metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average* dalam
Prediksi Harga Saham**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh
gelar sarjana dalam bidang Informatika**

**Yoel Agustinus
1118042**



INSTITUT
TEKNOLOGI
HARAPAN
BANGSA

Veritas vos liberabit

**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
2022**

HALAMAN PERNYATAAN ORISINALITAS

**Saya menyatakan bahwa Tugas Akhir yang saya susun ini
adalah hasil karya saya sendiri.**

**Semua sumber yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

**Saya bersedia menerima sanksi pencabutan gelar akademik
apabila di kemudian hari Tugas Akhir ini terbukti plagiat.**

Bandung, 06 Desember 2022

A handwritten signature in black ink, appearing to be 'Yoel Agustinus', written over a horizontal line.

**Yoel Agustinus
1118042**

HALAMAN PENGESAHAN TUGAS AKHIR

Tugas Akhir dengan judul:

**PERBANDINGAN PENERAPAN METODE BIDIRECTIONAL LONG SHORT
TERM MEMORY DAN AUTOREGRESSIVE INTEGRATED MOVING
AVERAGE DALAM PREDIKSI HARGA SAHAM**

yang disusun oleh:

Yoel Agustinus

1118042

telah berhasil dipertahankan di hadapan Dewan Penguji Sidang Tugas Akhir yang
dilaksanakan pada:

Hari / tanggal : Selasa, 06 Desember 2022

Waktu : 12.30 WIB

Menyetujui

Pembimbing Utama:

Ventje J. L. Engel, M.T., CEH.

116019

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Harapan Bangsa, saya yang bertandatangan di bawah ini:

Nama : Yoel Agustinus

NIM : 1118042

Program Studi : Informatika

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Harapan Bangsa **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Rights*) atas karya ilmiah saya yang berjudul:

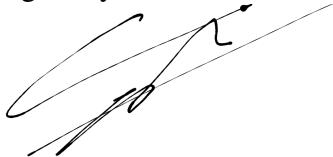
PERBANDINGAN PENERAPAN METODE BIDIRECTIONAL LONG SHORT TERM MEMORY DAN AUTOREGRESSIVE INTEGRATED MOVING AVERAGE DALAM PREDIKSI HARGA SAHAM

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Harapan Bangsa berhak menyimpan, mengalihmediakan, mengelola dalam pangkalan data, dan memublikasikan karya ilmiah saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Bandung, 06 Desember 2022

Yang menyatakan



Yoel Agustinus

ABSTRAK

Nama : Yoel Agustinus
Program Studi : Informatika
Judul : Perbandingan Penerapan Metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average* dalam Prediksi Harga Saham

Perekonomian Indonesia tidak lepas dari pasar modal. Banyak perusahaan yang menawarkan kepemilikan perusahaan dalam bentuk saham kepada publik. Dalam penelitian ini, akan membandingkan model *Long Short Term Memory* (LSTM) dengan *Autoregressive Integrated Moving Average* (ARIMA), untuk menemukan metode terbaik bagi prediksi harga saham. Dengan menggunakan dataset dari sektor *Fast-Moving Consumer Goods*(FMCG) antara lain: PT. Unilever(UNVR), PT. Gudang Garam(GGRM), PT. Prasadha Aneka Niaga(PSDN). ARIMA memiliki performa rata-rata yang lebih baik jika dibandingkan dengan model LSTM. Dapat disimpulkan rata-rata RMSE, MAE, MAPE dari metode ARIMA lebih kecil daripada metode LSTM. Hal ini disebabkan, LSTM menggunakan jumlah fitur yang lebih banyak, tidak hanya *Close* tetapi menggunakan *Open*, *High*, *Low*, *Volume* untuk observasi dan melakukan prediksi, sementara ARIMA hanya menggunakan fitur *Close* untuk observasi dan prediksi. Saham Gorengan dari perusahaan PSDN, mendapatkan nilai error RMSE, MAE yang lebih rendah daripada perusahaan saham *blue chip*. Untuk keseluruhan hasil pengujian nilai *error* untuk *short term forecasting* terlihat lebih kecil dibandingkan dengan ketika melakukan *mid*, *long term forecasting*. Hal ini disebabkan jumlah data pada *short term forecasting* lebih sedikit dibandingkan dengan jumlah data pada *mid*, *long term forecasting*. Nilai rata-rata RMSE, MAE, MAPE ARIMA berturut-turut ialah 363.11, 266.75, 2.06. Lalu, nilai rata-rata *error* LSTM ialah 621.98, 525.51, 3.83.

Kata kunci: Harga Saham, *Long Short Term Memory*, *Autoregressive Integrated Moving Average*, *Neural Network*, Prediksi, *Epoch*, *Unit*, (p,d,q).

ABSTRACT

Name : Yoel Agustinus
Department : Informatics
Title : Comparison of the Application of Bidirectional Long Short Term Memory and Autoregressive Integrated Moving Average Methods in Stock Price Prediction

The Indonesian economy cannot be separated from the capital market. Many companies offer ownership in the form of shares to public companies. In this study, we will compare the Long Short Term Memory (LSTM) model with the Autoregressive Integrated Moving Average (ARIMA), to find the best method for predicting stock prices. Using datasets from the Fast-Moving Consumer Goods (FMCG) sector, among others: PT. Unilever (UNVR), PT. Gudang Garam (GGRM), PT. Prasadha Aneka Niaga (PSDN). ARIMA has a better average performance when compared to the LSTM model. It can be concluded that the average RMSE, MAE, MAPE from the ARIMA method is smaller than the LSTM method. This is because LSTM uses a larger number of features, not only Close but uses Open, High, Low, Volume for observation and prediction, while ARIMA only uses the Close feature for observation and prediction. Fried shares from PSDN companies, get RMSE, MAE error values that are lower than blue chip company shares. For the overall test results, the error value for short-term forecasting looks smaller than when doing medium and long-term forecasting. This is because the amount of data in short-term forecasting is less than the amount of data in medium and long-term forecasting. The average values of RMSE, MAE, MAPE ARIMA are 363.11, 266.75, 2.06 respectively. Then, the average value of the LSTM error is 621.98, 525.51, 3.83.

Keywords: Stocks Price, Long Short Term Memory, Autoregressive Integrated Moving Average, Neural Network, Prediction, Epoch, Unit, (p,d,q).

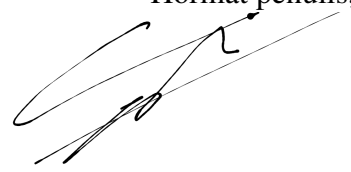
KATA PENGANTAR

Puji serta syukur penulis ucapkan kepada Yesus Kristus yang Maha Esa yang telah memberikan banyak rejeki, dan bimbingan-Nya penulis dapat menyelesaikan karya Tugas Akhir untuk menyelesaikan program Sarjana (S1) dengan judul "Perbandingan Penerapan Metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average* dalam Prediksi Harga Saham". Dengan kesempatan yang diberikan, penulis ucapkan terima kasih sebesar-besarnya dengan ketulusan hati kepada pihak yang telah membantu dan membimbing penulis selama penyusunan Tugas Akhir ini, yakni kepada:

1. Yesus Kristus, karena anugrah dan hikmat-Nya, penulis mendapat ruang pengharapan, hiburan, dan semua bentuk kekuatan untuk menyelesaikan Tugas Akhir ini.
2. Bapak Ventje J. L. Engel, S.T, M.T, selaku pembimbing utama Tugas Akhir yang senantiasa memberikan dukungan, waktu, ilmu, serta saran yang membangun kepada penulis selama pembuatan Tugas Akhir ini.
3. Ibu Ir. Inge Martina, M.T., selaku penguji I Tugas Akhir yang telah memberikan pengujian serta masukan dan pengarahan untuk menyempurnakan Tugas Akhir ini.
4. Ibu Ken Ratri Retno Wardani, S.Kom, M.T., selaku penguji II Tugas Akhir yang telah memberikan pengujian serta masukan dan pengarahan untuk menyempurnakan Tugas Akhir ini.
5. Seluruh dosen dan *staff* Departemen Teknik Informatika ITHB yang turut membantu kelancaran dalam menyelesaikan Tugas Akhir ini.
6. Keluarga, keluarga besar, dan Elshaddai Creative Community Church, Our Daily Bread Indonesia yang selalu menyediakan waktu, doa, semangat, motivasi, dukungan, dan semangat kepada penulis hingga terselesaikannya Tugas Akhir ini.
7. Teman seangkatan, adik, dan kakak kelas Informatika, dan teman-teman dalam prodi ITHB lainnya yang turut membantu dan mendukung penulis dapat menyelesaikan Tugas Akhir ini.

Bandung, 06 Desember 2022

Hormat penulis,

A handwritten signature in black ink, consisting of several fluid, overlapping loops and strokes, positioned below the text 'Hormat penulis,'.

Yoel Agustinus

DAFTAR ISI

ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
DAFTAR ALGORITMA	xiv
BAB 1 PENDAHULUAN	1-1
1.1 Latar Belakang	1-1
1.2 Rumusan Masalah	1-3
1.3 Tujuan Penelitian	1-3
1.4 Batasan Masalah	1-4
1.5 Kontribusi Penelitian	1-4
1.6 Metodologi Penelitian	1-4
1.7 Sistematika Pembahasan	1-5
BAB 2 LANDASAN TEORI	2-1
2.1 Tinjauan Pustaka	2-1
2.1.1 Harga Saham	2-1
2.1.1.1 <i>Candlestick</i>	2-2
2.1.1.2 Teknikal Analisis	2-2
2.1.1.3 Saham <i>Blue Chip</i> dan Gorengan	2-3
2.1.2 <i>Autoregressive Integrated Moving Average</i> (ARIMA)	2-4
2.1.3 <i>Artificial Neural Network</i> (ANN)	2-6
2.1.4 <i>Recurrent Neural Network</i> (RNN)	2-7
2.1.5 Fungsi Aktivasi	2-9
2.1.6 <i>Preprocessing Min-max Scaler</i>	2-11
2.1.7 <i>Long Short Term Memory</i>	2-11

2.1.7.1	<i>Forward Layer LSTM</i>	2-12
2.1.7.2	<i>Bidirectional Long Short Term Memory</i>	2-16
2.1.7.3	<i>Dense / Fully Connected Layer</i>	2-20
2.1.8	Evaluasi <i>Forecasting</i>	2-20
2.1.8.1	<i>Root Mean Square Error (RMSE)</i>	2-21
2.1.8.2	<i>Mean Absolute Error (MAE)</i>	2-21
2.1.8.3	<i>Mean Absolute Percentage Error (MAPE)</i>	2-22
2.1.9	Pustaka <i>python</i>	2-23
2.1.9.1	<i>NumPy</i>	2-23
2.1.9.2	<i>Pandas</i>	2-23
2.1.9.3	<i>Matplotlib</i>	2-24
2.1.9.4	<i>Tensorflow Keras</i>	2-25
2.1.9.5	<i>Statsmodels</i>	2-27
2.2	Tinjauan Studi	2-28
2.3	Tinjauan Objek	2-31
2.3.1	<i>Time series Dataset</i>	2-31
2.3.1.1	<i>Trend</i>	2-32
2.3.1.2	<i>Seasonal</i>	2-33
2.3.1.3	<i>Cyclical</i>	2-33
2.3.2	<i>Forecasting Horizon</i>	2-34
2.3.3	Dataset Saham	2-34

BAB 3 ANALISIS DAN PERANCANGAN SISTEM 3-1

3.1	Analisis Masalah	3-1
3.2	Kerangka Pemikiran	3-1
3.3	Urutan Proses Global	3-3
3.3.1	Proses Training	3-5
3.3.2	Proses Testing	3-6
3.4	Analisis Manual	3-7
3.4.1	Dataset	3-7
3.4.2	Normalisasi <i>Min-max</i>	3-7
3.4.3	<i>Splitting</i> Dataset untuk <i>Training</i> dan <i>Testing</i>	3-8
3.4.4	Perhitungan <i>Long Short Term Memory</i>	3-9
3.4.4.1	<i>Forward Layer LSTM</i>	3-10
3.4.4.2	<i>Backward Layer LSTM</i>	3-23
3.4.4.3	Perhitungan <i>Dense</i> pada <i>Output Layer</i>	3-28
3.4.5	Perhitungan <i>Autoregressive Integrated Moving Average</i>	3-30
3.4.6	Perbandingan Evaluasi <i>Forecasting</i>	3-33

BAB 4	IMPLEMENTASI DAN PENGUJIAN	4-1
4.1	Lingkungan Implementasi	4-1
4.1.1	Spesifikasi Perangkat Keras	4-1
4.1.2	Spesifikasi Perangkat Lunak	4-1
4.2	Implementasi Class	4-1
4.2.1	<i>Class EvaluasiForecasting</i>	4-1
4.2.2	<i>Class Preprocessing</i>	4-2
4.2.3	<i>Class ARIMA_model</i>	4-5
4.2.4	<i>Class LSTM_unit</i>	4-6
4.3	Implementasi Perangkat Lunak	4-8
4.3.1	Implementasi <i>Preprocessing</i>	4-8
4.4	Pengujian	4-9
4.4.1	Skenario Pengujian <i>Autoregressive Integrated Moving Average</i>	4-9
4.4.2	Skenario Pengujian <i>Long Short Term Memory</i>	4-9
4.4.3	Pembahasan Pengujian <i>Long Term Prediction</i>	4-10
4.4.4	Pembahasan Pengujian <i>Mid Term Prediction</i>	4-15
4.4.5	Pembahasan Pengujian <i>Short Term Prediction</i>	4-20
4.4.6	Pembahasan Pengujian	4-25
BAB 5	KESIMPULAN DAN SARAN	5-1
5.1	Kesimpulan	5-1
5.2	Saran	5-2
DAFTAR REFERENSI		i
BAB A	HASIL PENGUJIAN ARIMA	A-1
BAB B	HASIL PENGUJIAN LSTM	B-3

DAFTAR TABEL

2.1	Daftar <i>method</i> yang digunakan dari pustaka NumPy	2-23
2.2	Daftar <i>method</i> yang digunakan dari pustaka Pandas	2-23
2.3	Daftar <i>method</i> yang digunakan dari pustaka Matplotlib	2-24
2.4	Daftar <i>method</i> yang digunakan dari pustaka Tensorflow Keras . . .	2-25
2.5	Daftar <i>method</i> yang digunakan dari Statsmodels	2-27
2.6	Tinjauan Studi	2-28
3.1	Tabel Matriks Data Saham	3-7
3.2	Tabel Matriks Data Saham	3-8
4.1	Daftar <i>method</i> pada class <i>EvaluasiForecasting</i>	4-2
4.2	Daftar <i>method</i> pada class <i>preprocessing</i>	4-3
4.3	Daftar <i>method</i> pada class <i>ARIMA_model</i>	4-5
4.4	Daftar <i>method</i> pada class <i>LSTM_unit</i>	4-6
4.5	Skenario Pengujian kombinasi ARIMA	4-9
4.6	Skenario Pengujian kombinasi LSTM	4-10
4.7	Hasil pengujian <i>Autoregressive Integrated Moving Average</i> untuk <i>Long Term Prediction</i>	4-10
4.8	Hasil pengujian <i>Bidirectional Long Short Term Memory</i> untuk <i>Long</i> <i>Term Prediction</i>	4-11
4.9	Hasil pengujian <i>Autoregressive Integrated Moving Average</i> untuk <i>Mid Term Prediction</i>	4-15
4.10	Hasil pengujian <i>Long Short Term Memory</i> untuk <i>Mid Term Prediction</i>	4-15
4.11	Hasil pengujian <i>Autoregressive Integrated Moving Average</i> untuk <i>Short Term Prediction</i>	4-20
4.12	Hasil pengujian <i>Long Short Term Memory</i> untuk <i>Short Term</i> <i>Prediction</i>	4-20
4.13	Ringkasan keseluruhan hasil pengujian	4-28
A-1	Pengukuran <i>error</i> yang digunakan pada Jurnal	A-1
B-1	Hasil pengujian <i>Autoregressive Integrated Moving Average</i>	B-3
B-2	Hasil pengujian <i>Long Short Term Memory</i>	B-5

DAFTAR GAMBAR

2.1	Candlestick [11]	2-2
2.2	Artificial Neural Network Structure [23]	2-7
2.3	Recurrent Neural Network Loop	2-7
2.4	Recurrent Neural Network [26]	2-8
2.5	Sigmoid Function [24]	2-10
2.6	tanh function [24]	2-11
2.7	Bidirectional LSTM [18]	2-12
2.8	Long Short Term Memory unit [7]	2-13
2.9	Downtrend	2-33
2.10	Seasonal Time series [20]	2-33
2.11	Cyclical Time series [20]	2-34
2.12	Data Sampling [19]	2-35
3.1	Kerangka Pemikiran	3-2
3.2	Urutan Proses Global	3-4
3.3	Flowchart proses <i>training</i>	3-5
3.4	Flowchart proses <i>Testing</i>	3-6
3.5	Bidirectional LSTM with Unit	3-9
4.1	Fungsi-Fungsi untuk melakukan Evaluasi pada model yang dihasilkan	4-2
4.2	Fungsi-Fungsi untuk melakukan <i>Preprocessing</i> 1	4-4
4.3	Fungsi-Fungsi untuk melakukan <i>Preprocessing</i> 2	4-4
4.4	Fungsi-Fungsi untuk melakukan pembuatan Model ARIMA	4-6
4.5	Fungsi-Fungsi untuk melakukan pembuatan Model LSTM	4-7
4.6	Library Long Short Term Memory [29]	4-8
4.7	Perbandingan Chart Data GGRM untuk <i>Long Term</i>	4-12
4.8	Perbandingan Chart Data UNVR untuk <i>Long Term</i>	4-12
4.9	Perbandingan Data PSDN untuk <i>Long Term</i>	4-13
4.10	Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian <i>long term prediction</i> ARIMA dan Bi-LSTM	4-14
4.11	Perbandingan Data GGRM untuk <i>mid Term</i>	4-16
4.12	Perbandingan Data UNVR untuk <i>mid Term</i>	4-17
4.13	Perbandingan Data PSDN untuk <i>mid Term</i>	4-18
4.14	Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian <i>mid term prediction</i> ARIMA dan Bi-LSTM	4-19
4.15	Perbandingan Data GGRM untuk <i>Short Term</i>	4-21

4.16	Perbandingan Data UNVR untuk <i>Short Term</i>	4-22
4.17	Perbandingan Data PSDN untuk <i>Short Term</i>	4-23
4.18	Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian <i>short term prediction</i> ARIMA dan Bi-LSTM	4-24
4.19	Perbandingan nilai rata-rata <i>long term prediction</i> RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM	4-25
4.20	Perbandingan nilai rata-rata <i>mid term prediction</i> RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM	4-25
4.21	Perbandingan nilai rata-rata <i>short term prediction</i> RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM	4-26
4.22	Rata-rata hasil pengujian RMSE, MAE, MAPE Bi-LSTM vs ARIMA	4-26
4.23	Rata-rata hasil pengujian RMSE, MAE, MAPE Blue Chip vs Gorengan	4-27

DAFTAR LAMPIRAN

LAMPIRAN A	A-1
LAMPIRAN B	B-3

DAFTAR ALGORITMA

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perekonomian Indonesia tidak lepas dari pasar modal. Banyak perusahaan yang menawarkan kepemilikan perusahaan dalam bentuk saham kepada publik [1]. Harga saham merupakan faktor yang sangat penting dan harus diperhatikan oleh investor dalam melakukan investasi karena harga saham menunjukkan prestasi emiten sebuah perusahaan. Terdapat empat komponen utama dalam Harga saham dipasar modal, yaitu: harga tertinggi (*high price*), harga terendah (*low price*), harga penutupan (*close price*), harga penutupan yang sudah disesuaikan (*adjusted close*). Harga tertinggi atau terendah merupakan harga yang paling tinggi atau paling rendah yang terjadi pada satu hari bursa, harga penutupan merupakan harga yang terjadi terakhir pada saat akhir jam bursa, harga penutupan yang disesuaikan merupakan harga penutupan yang telah disesuaikan dengan aksi korporasi seperti *right issue*, *stock split* atau *stock reverse*.

Ada penelitian yang menggunakan metode ARIMA untuk memprediksi harga saham Garuda Indonesia sejak 22 April 2019 sampai dengan 20 April 2020 dengan jumlah 255 data [2]. Peneliti melakukan banyak eksperimen dengan mengubah parameter p,d,q dan menemukan nilai parameter terbaik dengan orde 3,1,2 dengan mendapatkan nilai RMSE 38.03.

Pada penelitian [3], dijelaskan bahwa penulis memprediksi harga saham dengan menggunakan dataset *Amman Stock Exchange* dari Januari 2010 sampai Januari 2018 dengan menggunakan metode ARIMA. Lalu, peneliti mencari nilai parameter ARIMA(p,d,q) dengan nilai RMSE paling rendah yaitu 4 dengan menggunakan nilai parameter ARIMA(2,1,1).

Ada penelitian yang menggunakan berbagai macam metode *Machine Learning* dan membandingkan dengan metode *deep learning* untuk prediksi harga saham dengan menggunakan dataset dari *iShares MSCI United Kingdom* sejak Januari 2015 sampai Juni 2018 [4]. Metodenya antara lain: *Artificial Neural Network*, *Random Forest*, *Support Vector Regression*, dan *Long Short Term Memory* dengan mendapatkan nilai RMSE berturut-turut ialah: 0.454131, 0.389482, 0.340657, 0.306543. Namun, yang mempunyai RMSE terendah adalah model yang dibangun dengan metode *Long Short Term Memory*.

Ada penelitian lain yang membandingkan beberapa macam metode untuk

prediksi harga saham antara lain: *Decision Tree*, *Bagging*, *Random Forest*, *AdaBoost*, *Gradient Boosting*, *XGBoost*, ANN, RNN, dan LSTM [5]. Dataset yang digunakan merupakan dataset dari berbagai macam sektor saham yang diambil dalam rentang waktu 10 tahun. Model LSTM merupakan hasil metode perwakilan dari *neural network* yang mendapatkan nilai MAPE, MAE, RMSE, MSE terendah diantara metode yang lainnya dengan nilai 0.77, 10.03, 0.0121, 376.82.

Ada penelitian prediksi saham di masa pandemi COVID-19, dengan menggabungkan metode *Long Short Term Memory* dan *Gated Recurrent Unit* [6]. Penelitian ini menggunakan data saham *blue chip* yang ada di Indonesia antara lain: BBCA, INDF, BBRI, ASII, TLKM, jangka waktu pengambilan data saham mulai sejak 2 Februari 2015 sampai 19 Juni 2020 dengan jumlah 1357 data. Hasil RMSE terkecil dengan nilai 71.658 dan dengan *epoch* 35 dimiliki oleh emiten TLKM dengan hasil *Hidden Layer* GRU-LSTM-GRU-LSTM.

Ada penelitian lain yang menggunakan metode dari *Neural network based* seperti LSTM, GRU, WLSTM metode LSTM dengan menggunakan *Wavelet Transform* dan WLSTM dengan *Attention Model* [7]. Dataset yang digunakan merupakan harga data dari S&P 500, DJIA, dan HSI pada rentang waktu dari tahun 2000 sampai 2019. Hasil evaluasi dengan menggunakan model LSTM untuk masing-masing dataset berturut-turut antara lain: 0.2337, 0.1971, 0.3429.

Dalam penelitian lain yang membandingkan metode *Autoregressive Integrated Moving Average* dengan metode *Long Short Term Memory* dengan menggunakan emiten dari banyak perusahaan [8]. Setelah dilakukan penelitian disimpulkan bahwa rata-rata nilai RMSE dari model LSTM jauh lebih kecil dari rata-rata nilai RMSE dari model ARIMA dengan menunjukkan nilai RMSE dari model ARIMA adalah 511.381 dan nilai RMSE dari model LSTM adalah 64.445 yang bisa diartikan model LSTM lebih baik 87% dari model ARIMA.

Ada penelitian lain yang melakukan prediksi harga saham dengan menggunakan metode LSTM [9]. Dataset yang digunakan merupakan dataset dari indeks NIFTY-50 rentang waktu yang digunakan dari 1 Januari 2011 sampai 31 Desember 2016. Pada penelitian ini menggunakan 2 macam *epoch*: 250 dan 500, menggunakan parameter variabel untuk melakukan *observasi* yang berbeda-beda yaitu: *Open & Close*, *High & Low & Close*, dan *High & Low & Open & Close*. *Hyperparameter unit* yang digunakan ialah 128 dan 64. Setelah melakukan berbagai simulasi dengan jumlah parameter variabel, *epoch* yang berbeda, peneliti menyimpulkan bahwa dengan mengambil 4 set Fitur (*High & Low & Open &*

Close) dengan *epoch* 500 mencapai hasil terbaik dengan mendapatkan RMSE data *training* 0.00983 dan *testing* 0.00859.

Terdapat penelitian lain yang memprediksi pergerakan harga saham dengan menggunakan algoritme *Random Forest* [12]. Dataset yang digunakan adalah PT. Unilever (UNVR), PT. Gudang Garam (GGRM), dengan rentang waktu dari tahun 2010 - 2015. Pada tahap *feature extraction*, peneliti menggunakan 10 indikator teknis dengan parameter selama 10 hari untuk semua indikator pada data agar dapat menangkap trend yang terjadi berdasarkan kejadian pada harga saham. Pada tahap uji, peneliti menggunakan parameter *Random Forest* dengan banyak pohon keputusan yang dibuat: 10, 25, 50, 100, 200, 500, 1000. Maka, diperoleh nilai *f-measure* 88.47239 pada data saham UNVR dengan menggunakan 500 pohon keputusan dan 99.49253 pada data saham GGRM dengan menggunakan 1000 pohon keputusan.

Dalam penelitian ini, akan membandingkan model *Bidirectional Long Short Term Memory* (Bi-LSTM) dengan *Autoregressive Integrated Moving Average* (ARIMA), untuk menemukan metode terbaik bagi prediksi harga saham. Dataset yang digunakan adalah data *time-series* dari sektor bank di Indonesia yang diambil dari *Yahoo Finance* dalam rentang waktu yang telah ditentukan. Saham perusahaan dari sektor *Fast-Moving Consumer Goods* (FMCG) antara lain: PT. Unilever (UNVR), PT. Gudang Garam (GGRM) sebagai perwakilan dari saham sektor FMCG yang berstatus *blue chip*. Sementara, sebagai perwakilan dari saham sektor FMCG yang berstatus gorengan, antara lain: PT. Prasadha Aneka Niaga (PSDN).

1.2 Rumusan Masalah

Dalam pembahasan ini peneliti mencoba untuk merumuskan masalah sebagai berikut:

1. Berapakah perbandingan nilai *MAPE*, *MAE*, *RMSE* antara metode *Bidirectional Long Short Term Memory* (Bi-LSTM) dan *Autoregressive Integrated Moving Average* (ARIMA) untuk memprediksi penutupan harga saham?
2. Berapa hasil prediksi harga saham dengan mengimplementasikan Bi-LSTM dan ARIMA untuk *short-term*, *mid-term*, *long-term forecasting*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, berikut adalah tujuan dari penelitian ini.

1. Mendapatkan model prediksi serta melakukan prediksi penutupan harga saham di Indonesia dengan menggunakan metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average*.
2. Mengevaluasi hasil kinerja model yang dibangun oleh metode *Bidirectional Long Short Term Memory* dan *Autoregressive Integrated Moving Average* untuk memprediksi harga saham.

1.4 Batasan Masalah

Agar penelitian ini terfokus dan terarah, peneliti memberikan beberapa batasan masalah sebagai berikut:

1. Dataset yang digunakan diambil dari *Yahoo Finance* dimana seluruh data saham diambil dari sektor FMCG, antara lain: PT. Unilever (UNVR), PT. Gudang Garam (GGRM), PT. Prsidha Aneka Niaga (PSDN).
2. *Long term forecasting* menggunakan periode dari 1 Januari 2017 sampai 31 Desember 2021 dengan berjumlah 1261 data. *Mid term forecasting* menggunakan periode dari 1 Januari 2017 sampai 31 Desember 2017 dengan berjumlah 253 data. *Short term forecasting* menggunakan periode dari 1 Januari 2017 sampai 31 Maret 2017 dengan berjumlah 62 data.
3. Untuk melihat performa model yang dibuat akan dilihat dari RMSE, MAE, MAPE.

1.5 Kontribusi Penelitian

Kontribusi yang diberikan dari penelitian ini adalah membuat sebuah mesin untuk memprediksi harga saham dengan dari sektor saham yang telah ditentukan. Penelitian ini menggunakan metode Bi-LSTM dan ARIMA lalu menentukan nilai paramater yang tepat dari setiap metode, yang nantinya akan dilakukan mengevaluasi hasil prediksi dengan menggunakan RMSE, MAE, MAPE untuk menentukan hasil manakah yang terbaik.

1.6 Metodologi Penelitian

Penelitian ini dibuat dengan metode penelitian sebagai berikut.

1. Studi Literatur

Penelitian ini diawali dengan melakukan studi kepustakaan yang bersumber dari jurnal penelitian terkait saham, *Long Short Term Memory*, *Autoregressive Integrated Moving Average*, atau metode deep learning yang lain.

2. Analisis Masalah

Tahap ini penulis melakukan analisis terhadap permasalahan yang ada, serta

menentukan tujuan dan batasan yang ada pada penelitian ini.

3. Pencarian Data

Data yang digunakan untuk penelitian ini berupa data-data dari sektor saham yang bersumber dari *Yahoo Finance*, antara lain: PT. Unilever(UNVR), PT. Gudang Garam(GGRM), PT. Prasadha Aneka Niaga(PSDN).

4. *Training* Data

Tahap ini merupakan proses pelatihan kedua model untuk memprediksi penutupan harga saham menggunakan data latih yang telah dimodelkan sebelumnya.

5. *Testing* Data

Tahap ini merupakan tahap pengujian kedua model yang telah dilatih sebelumnya menggunakan data latih untuk memprediksi penutupan harga saham.

6. Evaluasi

Tahap ini merupakan proses untuk mengevaluasi hasil dari prediksi harga penutupan saham dari kedua model dengan menggunakan RMSE, MAE, MAPE.

1.7 Sistematika Pembahasan

Pada penelitian ini, penulis menyusun berdasarkan sistematika pembahasan sebagai berikut

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, kontribusi penelitian, metodologi penelitian, dan sistematika pembahasan.

BAB II LANDASAN TEORI

Bab ini berisi penjelasan dasar mengenai teori yang mendukung implementasi penelitian ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini berisi analisis algoritme Bi-LSTM dan ARIMA untuk membangun model prediksi harga saham.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian yang dilakukan menggunakan metode Bi-LSTM dan ARIMA dengan menggunakan data dari perusahaan GGRM, UNVR, PSDN dan menganalisis hasilnya.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penelitian yang dilakukan dan saran untuk penelitian lebih lanjut di masa mendatang.

BAB 2 LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini menggunakan beberapa teori terkait yang diperlukan dalam pengerjaan yang dilakukan. Penjelasan mengenai teori-teori tersebut akan dijelaskan sebagai berikut.

2.1.1 Harga Saham

Saham adalah surat yang menjadi bukti seseorang memiliki bagian modal suatu perusahaan. Saham (*stock*) merupakan salah satu instrumen pasar keuangan yang paling populer. Menerbitkan saham merupakan salah satu pilihan perusahaan ketika memutuskan untuk pendanaan perusahaan [10]. Pada sisi yang lain, saham merupakan instrument investasi yang banyak dipilih para investor karena saham mampu memberikan tingkat keuntungan yang menarik. Seorang investor umumnya dituntut untuk selalu mengikuti perkembangan informasi pasar dan harga pasar. Terdapat empat komponen utama dalam harga saham dipasar modal, yaitu: harga tertinggi (*high price*), harga terendah (*low price*), harga penutupan (*close price*), harga penutupan yang sudah disesuaikan (*adjusted close*). Harga tertinggi atau terendah merupakan harga yang paling tinggi atau paling rendah yang terjadi pada satu hari bursa, harga penutupan merupakan harga yang terjadi terakhir pada saat akhir jam bursa, harga penutupan yang disesuaikan merupakan harga penutupan yang telah disesuaikan

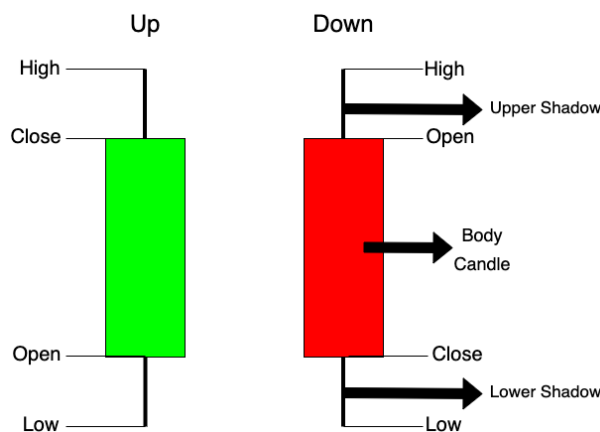
Untuk mendapatkan keuntungan tentunya diperlukan suatu analisis untuk memprediksi arah maupun harga saham. Secara garis besar analisis tersebut dapat digolongkan menjadi dua, yaitu analisis fundamental dan analisis teknikal. Analisis fundamental dalam memperkirakan harga saham di masa yang akan datang dilakukan dengan menggunakan nilai faktor-faktor fundamental, meliputi juga kinerja perusahaan, misalnya *Earning Per Share* (EPS), *Debt to Equity Ratio* (DER), *Return on Equity* (ROE), dan lainnya. Sementara, analisis teknikal adalah suatu metodologi peramalan fluktuasi harga saham yang datanya diambil dari data perdagangan saham yang terjadi di pasar saham (bursa efek) pada saat masa lampau.

Investor atau *trader* menganalisis pergerakan harga saham menggunakan perangkat statistik, seperti grafik dan rumus matematis untuk mendapatkan keuntungan baik dalam jangka pendek ataupun dalam jangka panjang. Analisis

teknikal berperang penting dalam menunjukkan *chart* harga saham, *trend* yang terjadi, *support & resistance*, serta waktu yang tepat untuk menjual maupun membeli dengan bantuan indikator.

2.1.1.1 *Candlestick*

Candlestick adalah salah satu jenis grafik harga saham yang digunakan dalam analisis teknikal yang menunjukkan harga tertinggi, terendah, pembukaan dan penutupan dari suatu saham pada periode waktu tertentu. *Candlestick* yang saat ini digunakan sebagai analisis teknikal di pasar saham memiliki dua komponen utama, yaitu: *Body Candle*, *Shadow Candle*. *Body Candle* adalah bagian dari *candlestick* yang menunjukkan harga pembukaan dan harga penutupan pada titik waktu tertentu yang ditunjukkan dari bentuk persegi empat berwarna merah atau hijau dapat dilihat pada Gambar 2.1. *Shadow Candle* adalah bagian yang menunjukkan harga tertinggi dan harga terendah saham pada titik waktu tertentu yang ditunjukkan dari garis lurus yang membentang di atas dan di bawah *body candlestick* umumnya *shadow candle* itu warnanya sama seperti *body candle*.



Gambar 2.1 Candlestick [11]

Jika *Close* letaknya di atas *Open*, dapat dikatakan *candlestick* lagi mengalami *bullish* atau saham sedang naik, pada warna hijau. Jika *Close* letaknya di bawah *Open*, maka dapat dikatakan *candlestick* lagi mengalami *bearish* atau saham sedang turun.

2.1.1.2 Teknikal Analisis

Trading saham menjadi cara yang populer dilakukan investor untuk memperoleh keuntungan dari investasi saham dengan cepat. Namun, terdapat resiko yang tinggi saat melakukan trading saham, dengan ini trading saham sebaiknya dilakukan dengan segala cara teknikal analisis supaya tetap mendapatkan keuntungan [14]. Terdapat macam metode dalam Teknikal Analisis

yang dapat digunakan:

1. Buat *Trading Plan*

Sebelum memulai *trading*, investor wajib mempunyai *trading plan* yang berfungsi sebagai rencana saat perdagangan saham yang dijadikan pedoman oleh investor dalam melakukan jual beli saham di Bursa Efek [13]. *Trading plan* terdiri dari:

- (a) Daftar saham untuk trading - daftar saham yang berpotensi untuk dilakukan trading dalam jangka pendek.
- (b) Titik *Entry(buy)* - titik *entry* adalah momen dimana untuk membeli saham ketika harganya mencapai titik tertentu.
- (c) Titik *Exit(sell)* - titik *sell* adalah momen dimana untuk menjual saham ketika harganya mencapai titik tertentu, untuk mendapatkan keuntungan.

2. Menggunakan *Moving Average*

Moving Average digunakan juga dalam metode ARIMA. Yang menggunakan pendekatan perhitungan dengan menggabungkan ketergantungan antar nilai observasi dengan rata-rata. Dalam saat melakukan *trading*, investor bisa mengidentifikasi pergerakan tren [14]. Bila investor melihat MA bergerak turun ke bawah semenrara harganya juga bergerak di bawah MA, maka tren yang sedang terjadi adalah sedang *downrend* begitu pula berlaku untuk sebaliknya.

3. Menentukan *Resistance* dan *Support*

Resistance digunakan untuk menandakan wilayah area harga tertentu yang diyakini sebagai titik atau area tertinggi untuk melakukan *sell*. Begitu juga sebaliknya, *Support* digunakan pada area terendah untuk melakukan *buy*.

2.1.1.3 Saham *Blue Chip* dan Gorengan

Dalam berinvestasi pada suatu perusahaan, terdapat istilah-istilah untuk mendeskripsikan suatu perusahaan untuk membedakan mana perusahaan yang mempunyai kinerja yang baik, atau kinerja yang buruk. Maka terdapat 2 macam jenis saham:

1. Saham *Blue Chip*

Istilah *blue chip* dikarenakan koin berwarna biru pada permainan *poker* dan kasino yang memiliki nilai yang paling tinggi [22]. Dan sering diindikasikan mempunyai kinerja perusahaan dan saham yang berkualitas baik. Perusahaan ini biasanya menjual produk-produk yang berkualitas dan digunakan oleh banyak orang. Contoh daftar perusahaan *blue chip*: Bank BCA(BBCA), PT. Telkom(TLKM), PT. Unilever(UNVR), PT. Astra(ASII), PT. Gudang Garam(GGRM).

2. Saham Gorengan

Istilah gorengan didapatkan pada sifat gorengan yang renyah dan enak saat masih hangat namun melempem ketika sudah dingin [22]. Sering diindikasikan saham gorengan adalah saham suatu perusahaan yang mengalami kenaikan tidak wajar akibat adanya rekayasa oleh pelaku pasar yang bertujuan untuk mendapatkan keuntungan jangka pendek. Contoh daftar perusahaan gorengan: Bank IBK Indonesia(AGRS), Bank MNC International(BABP), Trada Alam Minera(TRAM), PT. Surya Permata Andalan(NATO), PT. FKS Multi Agro Tbk(FISH), PT. Prasadha Aneka Niaga(PSDN).

2.1.2 Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average atau yang biasa disingkat menjadi ARIMA, merupakan metode yang menghasilkan ramalan berdasarkan sintesis dari pola data secara historis [2]. Metode ARIMA, juga dikenal sebagai peramalan *Box and Jenkins*. Metode ARIMA sangat baik untuk memprediksi data *time series* karena berdasarkan asumsi bahwa data *time series* tersebut stationer yang berarti rata-rata dan varian suatu data *time series* konstan. Metode ARIMA dibagi dalam 3 unsur, yaitu: *Autoregressive*(AR), *Moving Average*(MA), dan *Integrated*(I). Ketiga unsur ini dimodifikasi secara digabungkan dan menjadi ARIMA(p,d,q), p dinyatakan sebagai ordo AR, d dinyatakan sebagai ordo *Integrated* atau *difference*, dan q dinyatakan sebagai ordo MA. Apabila ada nilai parameter p,d,q berturut-turut ialah ARIMA(1,0,0) akan menjadi model yang menjalankan *Autoregressive*, dan nilai parameter p,d,q berturut-turut ialah ARIMA(0,0,1) akan menjadi model yang menjalankan *Moving Average*. Berikut merupakan persamaan model ARIMA.

$$Y_t = c + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \phi_1 \epsilon_{t-1} + \dots + \phi_q \epsilon_{t-q} + e_t \quad (2.1)$$

Keterangan :

Y_t	: nilai yang diamati saat t
β_p	: <i>coefficient</i> untuk AR
ϕ_q	: <i>coefficient</i> untuk MA
ϵ_t	: nilai rata-rata dari periode tersebut pada t
p	: nilai <i>input</i> dari ordo p
q	: nilai <i>input</i> dari ordo q
e_t	: nilai <i>error white noise</i>

1. Autoregressive (AR)

Model *autoregressive* merupakan model dari *multiple regression* dengan nilai *lagged* dari y_t sebagai prediktor [20]. Bentuk umum persamaan dari model *Autoregressive* adalah sebagai berikut:

$$Y_t = c + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + e_t \quad (2.2)$$

Keterangan :

Y_t : nilai yang diamati saat t
 β_p : *coefficient* untuk AR
 p : nilai *input* dari ordo p
 e_t : nilai *error white noise*

2. Integrated (I)

Integrated merupakan proses pembedaan (*differencing*) agar data yang tidak stasioner menjadi stasioner [2]. Pembedaan pertama (*first difference*) dari suatu deret waktu. Bentuk umum persamaan dari model *first difference* adalah sebagai berikut:

$$Df_t = y_t - y_{t-1} \quad (2.3)$$

Keterangan :

t : urutan nilai
 y_t : nilai yang diamati saat t
 Df_t : hasil nilai *difference*

3. Moving Average (MA)

Sebuah pendekatan yang memperhitungkan dengan menggabungkan ketergantungan antara nilai observasi dan nilai *residual error* dari menggunakan model rata-rata yang diterapkan pada nilai *lagged observations*. Bentuk umum persamaan dari model *Moving Average* adalah sebagai berikut:

$$Y_t = c + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_q \epsilon_{t-q} \quad (2.4)$$

Keterangan :

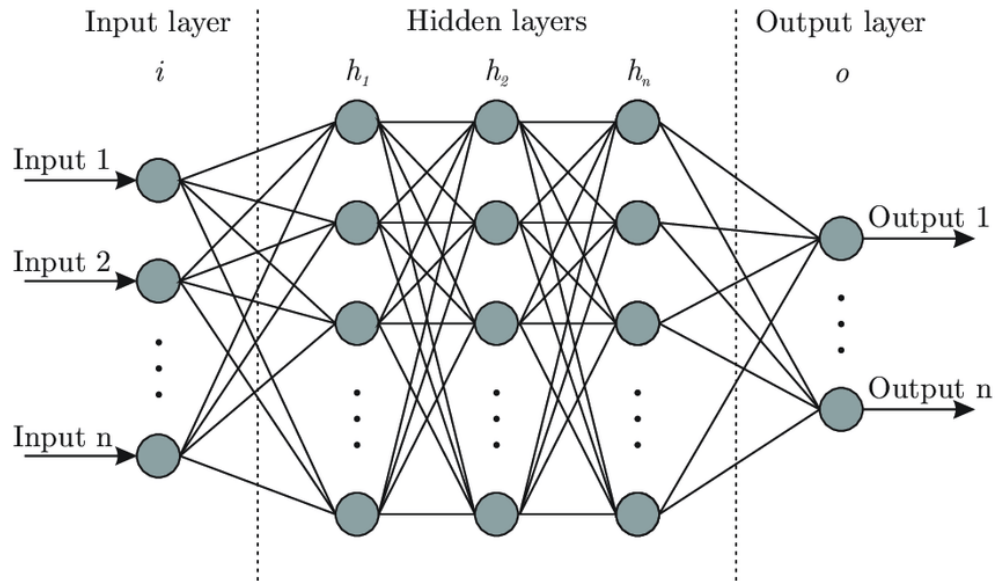
- Y_t : Nilai yang diamati saat t
 ϕ : *coefficient* untuk MA
 ϵ_t : nilai rata-rata dari periode tersebut pada t
 q : Nilai input dari ordo q

2.1.3 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah kecerdasan buatan yang mengadopsi kinerja otak manusia seperti memberi stimulus atau rangsangan, melakukan proses dan memberikan *output*. *Neural Network* mempunyai tiga lapisan layer yang disebut *input layer*, *hidden layer*, dan *output layer*. Jumlah fitur dari kumpulan data menentukan dimensi atau jumlah node di *input layer* [8]. Antar *node* tersebut terhubung melalui garis yang disebut "Synapses" dan terhubung ke *node* yang berada di *hidden layer* yang nantinya untuk diproses. Selanjutnya, akan diteruskan ke *output layer* atau layer paling akhir, yang berfungsi untuk mengeluarkan *output* neuron.

Neural Network terdapat 3 layer dan beserta elemen dasar dalam layer, yaitu [8]:

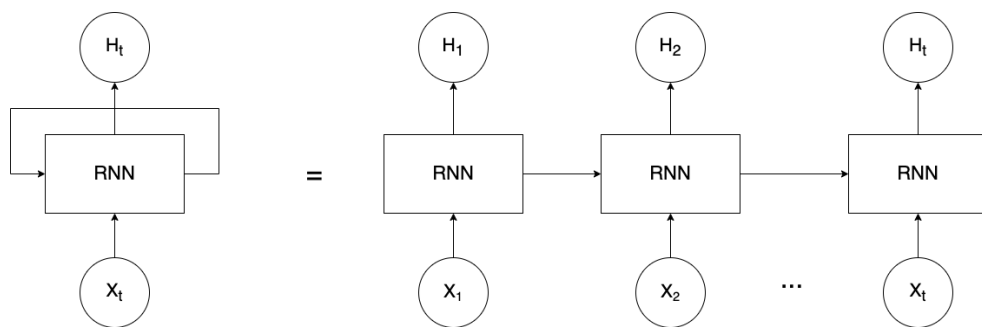
1. Kumpulan *Synapses* masing-masing mempunyai beberapa *weight*. Sementara, *weight* mempunyai peran untuk mengambil keputusan untuk memutuskan sinyal atau input mana yang dapat dilewati dan mana yang tidak.
2. Di *hidden layer*, node menerapkan *activation function* (seperti, *sigmoid* dan *tanh*) pada jumlah *input* yang nanti diproses dan menjadi nilai prediksi.
3. *Output layer* menghasilkan vektor probabilitas untuk berbagai *output* dan mendapatkan nilai *error rate*



Gambar 2.2 Artificial Neural Network Structure [23]

2.1.4 Recurrent Neural Network (RNN)

RNN adalah salah satu kelas dari ANN dimana koneksi antar *node* membentuk sebuah grafik yang bersambung [15]. RNN melakukan fungsi yang sama untuk setiap *input* data. Setelah menghasilkan *output* data, data tersebut kemudian dikirim kembali pada unit berikutnya yang disebut sebagai *recurrent network*. Dalam membuat keputusan, RNN mempertimbangkan *input* titik data saat ini dan *output* dari unit sebelumnya. Hal inilah yang membedakan RNN dari ANN, *input* dan *output* pada ANN tidak bergantung sama lain sedangkan RNN memiliki sebuah memori berisikan hasil perhitungan informasi yang dihasilkan sebelumnya.



Gambar 2.3 Recurrent Neural Network Loop

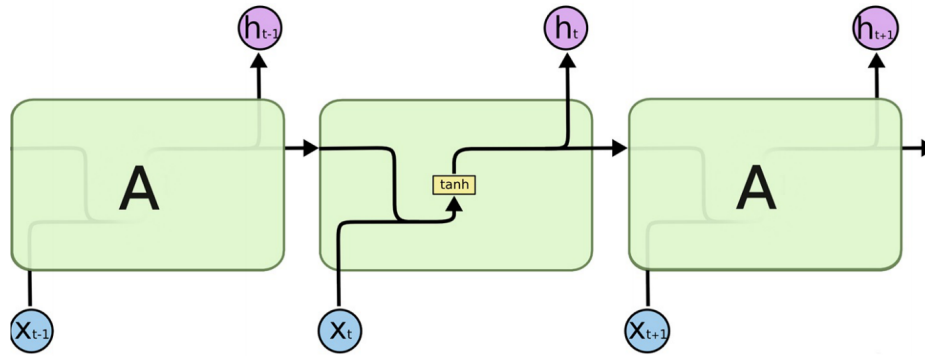
Keterangan :

x_t : nilai *input* pada *timestep* t

h_t : nilai *output* pada *timestep* t

Gambar 2.3 menunjukkan sebuah proses *timestep* RNN menerima *input* X_t dan menghasilkan *output* H_t , dan menunjukkan RNN akan memproses *output* dari unit RNN sebelumnya sebagai *input*. Sifat dasar RNN yang sekuensial tersebut menunjukkan bahwa model RNN memiliki arsitektur yang cocok untuk data berbentuk *sequence* atau *list* [17]. Dalam Gambar 2.3, dijelaskan:

1. Pertama, *timestep* 0 akan memproses *input* X_0 dan menghasilkan *output* H_0 .
2. Setelah perhitungan *timestep* 0 selesai, *output* 0 H_0 dan *input timestep* 1 X_1 keduanya akan menjadi *input* untuk *timestep* 1 dan demikian selanjutnya.
3. Demikian, *timestep* RNN akan tetap mengingat konteks dari hasil pembelajaran sebelumnya.



Gambar 2.4 Recurrent Neural Network [26]

Dalam Gambar 2.4 ditunjukkan bahwa *loop* yang diciptakan oleh *timestep* RNN memungkinkan informasi untuk diteruskan dari satu unit RNN ke *timestep* RNN berikutnya.

Berikut persamaan untuk menghitung *output* memori *layer* pada RNN.

$$H_t = f(H_{t-1}, X_t) \quad (2.5)$$

$$H_t = \tanh(W_{t-1} \cdot H_{t-1} + W_{hx} \cdot X_t) \quad (2.6)$$

Keterangan :

H : Memori Output

H_{t-1} : Memori *Output* sebelumnya

W_{t-1} : Bobot memori sebelumnya

W_{hx} : Bobot *input*

\tanh : Fungsi aktivasi *Hyperbolic Tangent*

Fungsi pada Persamaan 2.5 dijelaskan lebih rinci pada Persamaan 2.6 yang menunjukkan bahwa *input* pada titik data saat ini akan diterapkan fungsi aktivasi *Hyperbolic Tangent* bersamaan dengan menggunakan nilai memori dari *timestep* sebelumnya.

Berikut Persamaan 2.7 untuk menghitung *output* dari sistem RNN dengan mengaktivasi nilai memori terhadap bobot *output*.

$$Y_t = W_{hy}H_t \quad (2.7)$$

Keterangan :

Y_t : *output*

H_t : Memori *Output*

W_{hy} : Bobot *output*

2.1.5 Fungsi Aktivasi

Fungsi aktivasi dalam *neural network* digunakan untuk mengubah data *input* yang telah diberikan bobot menjadi *neuron* tersembunyi. Berikut merupakan beberapa fungsi aktivasi pada *neural network* yang akan digunakan dalam perhitungan LSTM pada penelitian ini:

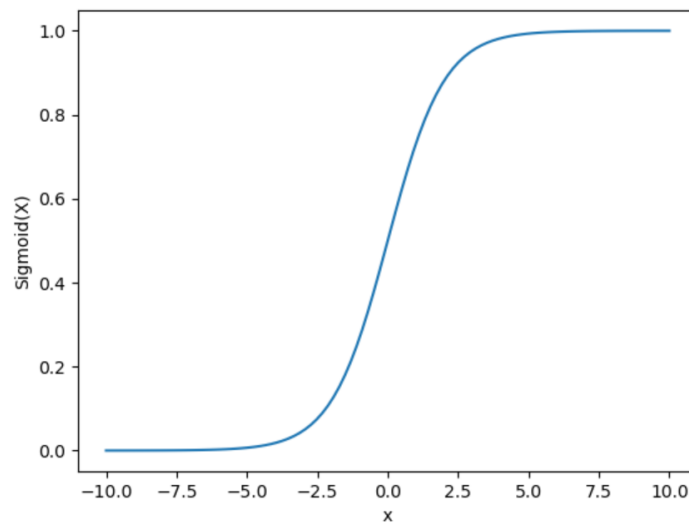
1. Sigmoid Function

Sigmoid function merupakan fungsi aktivasi yang menghasilkan nilai dengan skala 0 sampai dengan 1. Kelebihan dari fungsi aktivasi sigmoid yang menghasilkan nilai antara 0 dan 1 ini cocok digunakan untuk memrepresentasikan kemungkinan terjadinya suatu kondisi. Berikut merupakan Persamaan 2.8 untuk *sigmoid function*.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

Keterangan :

x : *input* bilangan *real*



Gambar 2.5 Sigmoid Function [24]

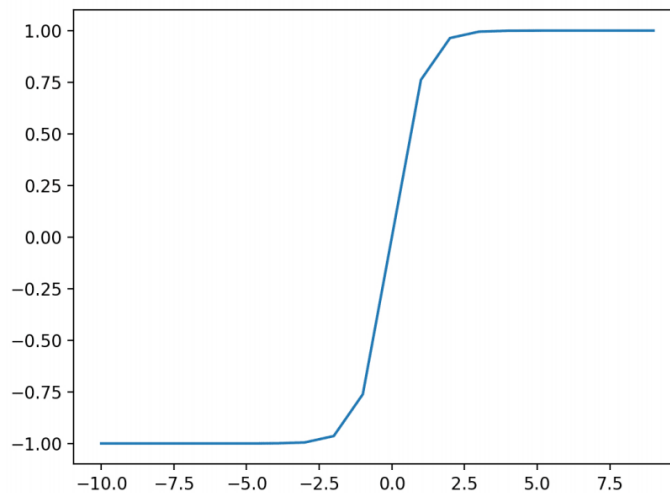
2. Hyperbolic Tangent Function (\tanh)

Hyperbolic Tangent Function merupakan fungsi aktivasi yang menghasilkan nilai dengan rentang -1 sampai dengan 1. Fungsi aktivasi ini bersifat non-linear dan kelebihanya dibandingkan sigmoid function adalah kemampuannya menghasilkan nilai negatif sampai -1. Berikut merupakan Persamaan 2.9 untuk *hyperbolic tangent function*.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.9)$$

Keterangan :

x : *input* bilangan *real*



Gambar 2.6 *tanh function* [24]

2.1.6 Preprocessing Min-max Scaler

Pada bagian ini akan dijelaskan proses *preprocessing* dengan melakukan *method Min-max Scaler* (banyak orang menyebut ini normalisasi) dengan pengertian yang paling sederhana ialah nilai digeser dan diskalakan ulang sehingga berubah dari 0 hingga 1. Dengan melakukan ini dengan mengurangi nilai *min* dan membaginya dengan *max* dikurangi *min*. *Scikit-Learn* menyediakan transformator yang disebut *MinMaxScaler* untuk ini [10].

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.10)$$

Keterangan :

x_{scaled} : Hasil dengan nilai baru

x : nilai yang di *observasi*

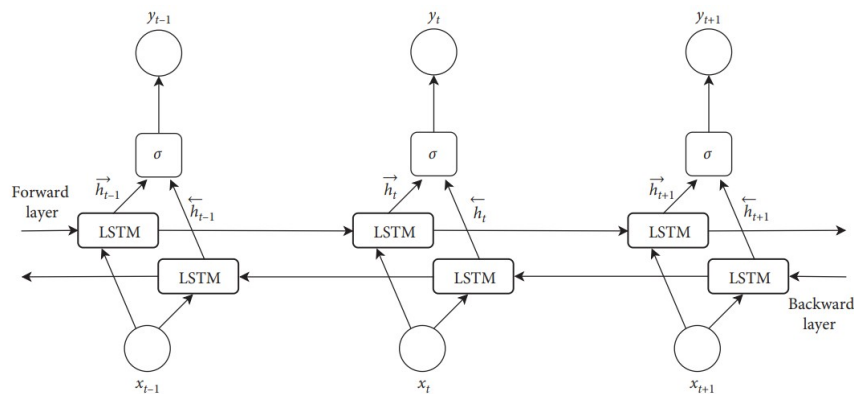
x_{min} : nilai terkecil dari data yang sedang di *observasi*

x_{max} : nilai terbesar dari data yang sedang di *observasi*

2.1.7 Long Short Term Memory

Long Short Term Memory (LSTM) merupakan arsitektur *Recurrent Neural Network* (RNN) yang digunakan dalam bidang *deep learning* untuk melakukan pembelajaran jangka panjang dengan memanfaatkan sebuah *timestep* memori yang disebut *cell* [6]. Arsitektur yang dimiliki oleh LSTM memungkinkan informasi

dari memori sebuah *timestep* diteruskan tidak hanya pada *timestep* berikutnya, melainkan dapat terus memberikan kontribusi pada *timestep* selanjutnya. LSTM didesain untuk mengatasi *vanishing gradient problem* yang terdapat pada RNN dengan memiliki *Constant Error Carousel* yang memungkinkan *error* untuk melakukan *backpropagation* tanpa terjadi *vanishing gradient* atau disebut *backpropagation through time* [11]. *Constant Error Carousel* mempertahankan aktivasi internal yang disebut sebagai (*state*) dengan *reccurent connection* dengan bobot tetap yang bernilai 1.0, yang dapat diatur ulang oleh *forget gate* [25].

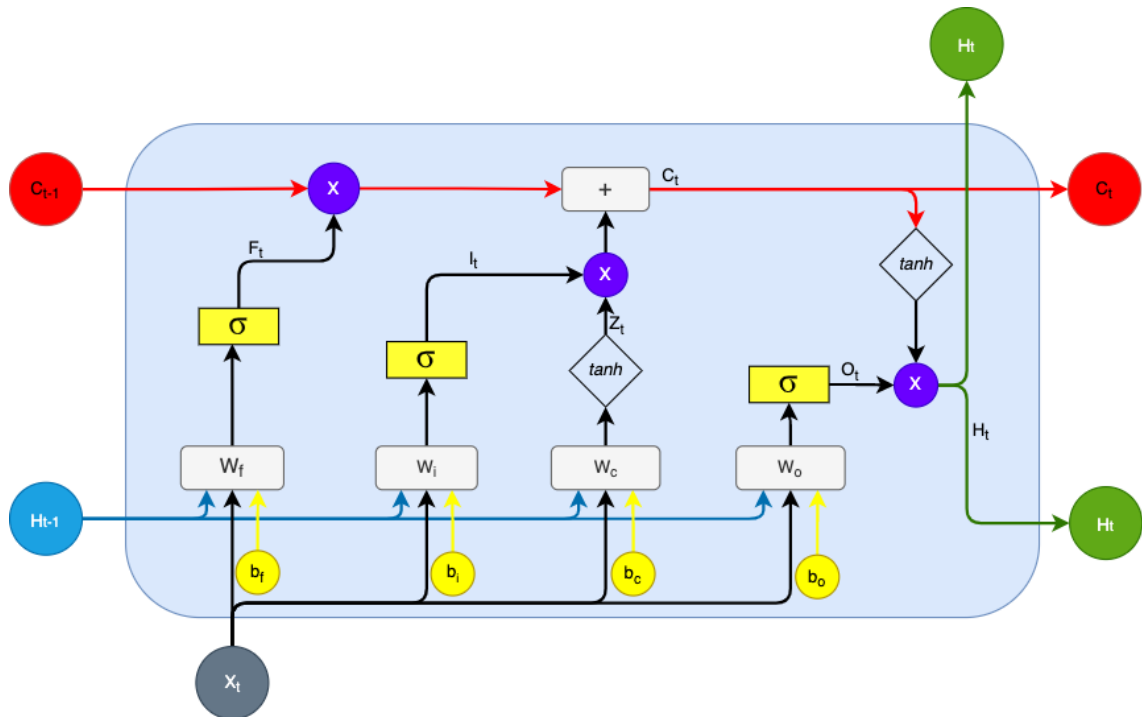


Gambar 2.7 Bidirectional LSTM [18]

Pada Gambar 2.7 merupakan gambar 2 arah dari LSTM dengan menggunakan *Forward Layer* dan *Backward Layer*. Maka, pada subbab selanjutnya akan dijelaskan secara detail mengenai *Forward* dan *Backward Layer* pada *Long Short Term Memory*.

2.1.7.1 Forward Layer LSTM

Forward Layer merupakan salah satu proses belajar dalam perhitungan LSTM yang alurnya maju. Dimana proses ini akan melakukan perhitungan LSTM yang variabel dan rumusnya sudah terdapat dalam masing-masing *gate*.



Gambar 2.8 Long Short Term Memory unit [7]

Keterangan :

- X_t : nilai *input* pada *timestep* t
- C_t : *cell state* pada *timestep* t
- C_{t-1} : *cell state* pada *timestep* $t-1$
- H_t : *output* pada *timestep* t
- H_{t-1} : *output* pada *timestep* $t-1$
- F_t : nilai aktivasi *forget gate*
- I_t : nilai aktivasi *input gate*
- Z_t : nilai aktivasi *memory gate*
- O_t : nilai aktivasi *output gate*
- σ : fungsi aktivasi sigmoid
- \tanh : fungsi aktivasi *hyperbolic tangent*
- W_f : bobot *forget gate*
- W_i : bobot *input gate*
- W_c : bobot *memory gate*
- W_o : bobot *output gate*
- b_f : bias *forget gate*
- b_i : bias *input gate*
- b_c : bias *memory gate*
- b_o : bias *output gate*
- X : fungsi perkalian

Unit LSTM terbagi menjadi 4 *gate* yang berinteraksi untuk menambahkan atau mengeluarkan informasi ke dalam memori setiap unit. Gambar 2.8 menjelaskan setiap *timestep* pada jaringan LSTM menerima 2 *input*, yaitu X_t *timestep input* dan H_{t-1} *output* dari *timestep* sebelumnya. Langkah pertama, LSTM akan memutuskan seberapa penting sebuah informasi untuk disimpan. Langkah ini dilakukan oleh *layer sigmoid "forget gate"* dengan Persamaan 2.11 berikut:

$$F_t = \sigma(W_f * H_{t-1} + U_f * X_t + b_f) \quad (2.11)$$

Keterangan :

- F_t : nilai aktivasi *forget gate*
 H_{t-1} : *output* dari *timestep* t sebelumnya
 X_t : *input* pada *timestep* t
 σ : fungsi aktivasi *sigmoid*
 W_f : bobot *forget gate*
 U_f : bobot *forget gate* untuk *input*
 b_f : bias *forget gate*

Fungsi aktivasi sigmoid akan menggunakan nilai H_{t-1} dan X_t menghasilkan nilai antara 0 dan 1, dimana hasil 1 menunjukkan bahwa nilai memori akan disimpan seluruhnya sedangkan hasil 0 dihapus selamanya [11].

$$I_t = \sigma(W_i * H_{t-1} + U_i * X_t + b_i) \quad (2.12)$$

$$Z_t = \tanh(W_c * H_{t-1} + U_c * X_t + b_c) \quad (2.13)$$

$$C_t = F_t * C_{t-1} + I_t * Z_t \quad (2.14)$$

Keterangan :

I_t	: nilai aktivasi <i>input gate</i>
W_i	: bobot <i>input gate</i>
U_i	: bobot <i>input gate</i> untuk <i>input</i>
H_{t-1}	: <i>output</i> dari <i>timestep</i> t sebelumnya
X_t	: <i>input</i> pada <i>timestep</i> t
Z_t	: nilai aktivasi <i>memory gate</i>
C_t	: <i>cell state</i> pada <i>timestep</i> t
F_t	: nilai aktivasi <i>forget gate</i>
C_{t-1}	: <i>cell state</i> dari <i>timestep</i> t sebelumnya
b_i	: bias <i>input gate</i>
W_c	: bobot <i>memory gate</i>
U_c	: bobot <i>memory gate</i> untuk <i>input</i>
b_c	: bias <i>memory gate</i>
\tanh	: fungsi aktivasi <i>hyperbolic tangent</i>
σ	: fungsi aktivasi <i>sigmoid</i>

Langkah berikutnya adalah melakukan perubahan terhadap nilai dari memori *timestep* yang terbagi menjadi 3 tahap. Pertama, *input gate* akan menentukan berapa besar informasi dengan melakukan fungsi aktivasi *sigmoid* terhadap *input* dari *timestep* yang akan dijadikan memori dengan menggunakan Persamaan 2.12. Tahap kedua, *memory gate* akan melakukan fungsi aktivasi *hyperbolic tangent* terhadap *input timestep* untuk menghasilkan nilai memori yang akan digunakan dengan Persamaan 2.13.

Tahap terakhir adalah untuk menghasilkan nilai memori baru yang diperoleh dari penjumlahan dari hasil perkalian *forget gate* (hasil dari Persamaan 2.11) terhadap nilai memori *timestep* sebelumnya (hasil dari Persamaan 2.12) dengan hasil perkalian *input gate* (hasil dari Persamaan 2.13) dan *cell state* menghasilkan nilai memori *timestep* saat ini yang akan digunakan untuk perhitungan selanjutnya pada Persamaan 2.14.

$$O_t = \sigma(W_o * H_{t-1} + U_o * X_t + b_o) \quad (2.15)$$

$$H_t = O_t.tanh(C_t) \quad (2.16)$$

Keterangan :

O_t	: nilai aktivasi <i>output gate</i>
H_{t-1}	: <i>output</i> dari <i>timestep</i> t sebelumnya
H_t	: <i>output</i> pada <i>timestep</i> t
X_t	: <i>input</i> pada <i>timestep</i> t
C_{t-1}	: <i>cell state</i> dari <i>timestep</i> t sebelumnya
C_t	: <i>cell state</i> pada <i>timestep</i> t
b_f	: bias <i>output gate</i>
W_o	: bobot <i>output gate</i>
U_o	: bobot <i>output gate</i> untuk <i>input</i>
\tanh	: fungsi aktivasi <i>hyperbolic tangent</i>
σ	: fungsi aktivasi <i>sigmoid</i>

Langkah terakhir adalah menghasilkan *output* H_t untuk *timestep* saat ini (t). *Output* yang dihasilkan merupakan nilai dari *memory cell* yang telah diseleksi. Pertama, *output gate* akan menentukan jumlah informasi yang akan dijadikan *output* H_t dengan menggunakan fungsi aktivasi *sigmoid* terhadap *input timestep* saat ini dan *output* dari *timestep* sebelumnya (Persamaan 2.15). Kemudian nilai dari *output gate* akan dikalikan dengan hasil fungsi aktivasi *hyperbolic tangent* terhadap nilai memori sesuai Persamaan 2.16, menghasilkan *output* untuk *timestep* saat ini.

Demikian diperoleh hasil *output* dan memori untuk satu *timestep* LSTM yang akan digunakan oleh *timestep* LSTM berikutnya. Dengan struktur arsitektur ini, maka nilai dari memori *timestep* akan terus tersimpan dan mempengaruhi pengambilan keputusan untuk *timestep* selanjutnya sehingga dapat mengatasi ketidakmampuan RNN dalam mengolah data dengan rentang yang besar.

2.1.7.2 Bidirectional Long Short Term Memory

RNN merupakan model deep learning yang baik dalam mengolah data *time series* karena keadaan internal RNN dapat menampilkan fitur temporal yang dinamis [27]. Pada Gambar 2.7 merupakan arsitektur *Bidirectional LSTM* adalah varian dari jaringan saraf berulang, yang memecahkan ketergantungan jangka panjang dari RNN dan LSTM. Ini menggabungkan LSTM dalam dua arah yang berbeda dan mengekstrak data informasi maju (*forward*) dan mundur (*backward*) secara bersamaan [28].

Backward Layer mempunyai *gate* dan perhitungan yang hampir sama dengan *forward*. *Backward Layer* ada untuk meningkatkan akurasi prediksi dalam

machine learning. Masing-masing dari perhitungan pada *Backward Layer* akan menghasilkan nilai perubahan bobot yang terlibat di setiap perhitungan.

$$\Delta Y_t = \Delta E + W_c \cdot \Delta z_{t+1} + W_i \cdot \Delta i_{t+1} + W_f \cdot \Delta f_{t+1} + W_o \cdot \Delta o_{t+1} \quad (2.17)$$

Keterangan :

- ΔY_t : selisih nilai dari *output* pada *timestep* LSTM yang sedang berjalan
- ΔE : nilai *output timestep* LSTM dan data asli saat H_t
- W_c : nilai bobot *memory gate*
- Δz_{t+1} : nilai *memory* dari *timestep* LSTM sesudahnya
- W_i : nilai bobot *input gate*
- Δi_{t+1} : nilai *input* dari *timestep* sesudahnya
- W_f : nilai bobot *forget gate*
- Δf_{t+1} : nilai *forget* dari *timestep* LSTM sesudahnya
- W_o : nilai bobot *output gate*
- Δo_{t+1} : nilai *output* dari *timestep* LSTM sesudahnya

Pada Persamaan 2.17, merupakan tahap pertama yang berfungsi menghitung selisih nilai dari *output* pada *timestep* LSTM yang sedang berjalan dengan menggunakan nilai *output* (H_t) *timestep* LSTM dan data asli, nilai *memory* dari *timestep* sesudahnya dikalikan bobotnya, nilai *input* dari *timestep* sesudahnya dikalikan bobotnya, nilai *forget* dari *timestep* sesudahnya dikalikan bobotnya, nilai *output* dari *timestep* sesudahnya. nilai *output* pada *timestep* LSTM yang sedang berjalan ini akan digunakan dalam perhitungan selanjutnya.

$$\Delta C_t = \Delta Y_t \cdot O_t \cdot (1 - \tanh^2(c_t)) + \Delta C_{t+1} \cdot f_{t+1} \quad (2.18)$$

$$\Delta O_t = \Delta Y_t \cdot \tanh(c_t) \cdot \sigma(o_t) \quad (2.19)$$

Keterangan	:
ΔC_t	: nilai dari <i>Cell state</i> pada <i>timestep</i> LSTM yang sedang berjalan
ΔY_t	: nilai <i>output</i> pada <i>timestep</i> LSTM yang sedang berjalan
O_t	: <i>output gate</i> pada <i>timestep</i> LSTM yang sedang berjalan
$(1 - \tanh^2(c_t))$: turunan dari nilai <i>tanh</i> terhadap <i>cell state</i>
ΔC_{t+1}	: nilai dari <i>cell state</i> pada <i>timestep</i> LSTM sesudahnya
f_{t+1}	: nilai <i>forget</i> dari <i>timestep</i> LSTM sesudahnya

Pada Persamaan 2.18, merupakan tahap kedua yang berfungsi menghitung nilai dari *cell state* pada *timestep* LSTM yang sedang berjalan yang nantinya hasil dari nilai *cell state* akan digunakan untuk perhitungan selanjutnya. Di dalam persamaan, terdapat nilai turunan dari *tanh* terhadap *cell state* yang digunakan untuk perkalian dengan nilai *output* yang hasil dari *cell state* akan digunakan untuk melakukan perhitungan selanjutnya.

$$\Delta F_t = \Delta C_t \cdot C_{t-1} \cdot (1 - \sigma^2(f_t)) \quad (2.20)$$

Keterangan	:
ΔF_t	: nilai dari <i>Forget gate</i> pada <i>timestep</i> LSTM yang sedang berjalan
ΔC_t	: nilai <i>Cell state</i> pada <i>timestep</i> LSTM yang sedang berjalan
C_{t-1}	: <i>Cell State</i> pada <i>timestep</i> LSTM sesudahnya
$(1 - \sigma^2(f_t))$: turunan dari nilai fungsi aktivasi <i>sigmoid</i> terhadap <i>forget gate</i>

Pada Persamaan 2.20, merupakan tahap ketiga yang berfungsi menghitung nilai dari *forget gate* pada *timestep* LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari *sigmoid* terhadap *forget gate* yang digunakan untuk perkalian dengan nilai *cell state* yang didapatkan pada Persamaan 2.18 dan nilai *cell state* pada *timestep* sesudahnya.

$$\Delta I_t = \Delta C_t \cdot z_t \cdot (1 - \sigma^2(i_t)) \quad (2.21)$$

$$\Delta Z_t = \Delta C_t \cdot i_t \cdot (1 - \tanh^2(z_t)) \quad (2.22)$$

Keterangan :

- ΔI_t : nilai dari *input gate* pada *timestep* LSTM yang sedang berjalan
- ΔZ_t : nilai dari *memory gate* pada *timestep* LSTM yang sedang berjalan
- ΔC_t : nilai *Cell state* pada *timestep* LSTM yang sedang berjalan
- z_t : nilai *memory gate* pada *timestep* LSTM sedang berjalan
- i_t : nilai *input gate* pada *timestep* LSTM sedang berjalan
- $(1 - \sigma^2(i_t))$: turunan dari nilai fungsi aktivasi *sigmoid* terhadap *input gate*
- $(1 - \tanh^2(i_t))$: turunan dari nilai fungsi aktivasi *tanh* terhadap *memory gate*

Pada Persamaan 2.21, merupakan tahap keempat yang berfungsi menghitung nilai dari *input gate* pada *timestep* LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari *sigmoid* terhadap *input gate* yang digunakan untuk perkalian dengan nilai *cell state* dan nilai *memory gate*. Pada Persamaan 2.22, merupakan tahap kelima yang berfungsi menghitung nilai dari *memory gate* pada *timestep* LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari fungsi aktivasi *tanh* terhadap *memory gate* yang digunakan untuk perkalian dengan nilai *cell state* dan nilai *input gate*.

Setelah didapatkan seluruh nilai dari perhitungan LSTM, langkah berikutnya adalah menghitung nilai bobot dan bias. Nilai ini didapatkan dengan menjumlahkan seluruh nilai *gate* yang dikalikan terhadap nilai *input* dan *output* dari *timestep* sesudahnya. Berikut merupakan persamaan untuk mendapatkan perubahan nilai (*update*) bobot dan bias yang digunakan dalam perhitungan LSTM.

$$\Delta W_{xgate} = \sum_{t=1}^T (\Delta gate_t x_t) \quad (2.23)$$

$$\Delta W_{hgate} = \sum_{t=1}^{T-1} (\Delta gate_t h_t) \quad (2.24)$$

$$\Delta b_{gate} = \sum_{t=1}^T (\Delta gate_t) \quad (2.25)$$

Keterangan :

$gate$: substitusi dari simbol $gate(f,i,c,o)$
ΔW_{xgate}	: nilai bobot dari <i>input</i> baru untuk perhitungan pada setiap <i>gate</i>
ΔW_{hgate}	: nilai bobot dari <i>output</i> untuk perhitungan <i>gate</i>
Δb_{gate}	: nilai bias
x_t	: nilai <i>input</i> pada <i>timestep</i> LSTM sedang berjalan
h_t	: nilai output dari <i>timestep</i> LSTM yang sedang diproses

Persamaan 2.23 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk perhitungan sebagai *input* yang dibutuhkan oleh masing-masing *gate timestep* LSTM yang sedang berjalan. Persamaan 2.24 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk perhitungan sebagai nilai *output timestep* LSTM yang sedang berjalan. Persamaan 2.25 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk perhitungan nilai bias yang dibutuhkan oleh masing-masing *gate*.

2.1.7.3 Dense / Fully Connected Layer

Dense atau *Fully Connected Layer* adalah yang mengeksekusi perkalian matriks-vektor. Parameter matriks diambil dengan memperbarui dan pelatihan menggunakan metodologi backpropagation. Hasil akhir dari lapisan padat adalah vektor dari n dimensi. Penggunaan lapisan padat dapat ditemukan secara luas dalam penskalaan, rotasi, penerjemahan, dan manipulasi dimensi vektor.

$$y = f(W_T * x + b) \quad (2.26)$$

Keterangan :

W_T	: bobot sambungan antar layer
f	: fungsi aktivasi
x	: nilai input dalam bentuk vektor
b	: bias

2.1.8 Evaluasi Forecasting

Bagian ini akan menjelaskan berbagai macam pengukuran evaluasi untuk hasil prediksi harga saham.

2.1.8.1 Root Mean Square Error (RMSE)

RMSE adalah aturan penilaian kuadrat yang mengukur besarnya rata-rata kesalahan. Persamaan untuk RMSE diberikan di kedua referensi. Atau penjelasan lainnya yaitu, RMSE adalah perbedaan antara nilai *forecasting* dan nilai yang diamati yang dikuadratkan dan kemudian dirata-ratakan pada sampel. Lalu, akar kuadrat dari rata-rata diambil. Karena kesalahan dikuadratkan sebelum dirata-ratakan, RMSE memberikan bobot yang relatif tinggi untuk nilai kesalahan yang besar. Ini berarti RMSE paling berguna ketika nilai kesalahan yang besar sangat tidak diinginkan. Agar lebih jelas bisa dilihat pada Persamaan 2.27 berikut.

$$RMSE = \sqrt{\frac{1}{N} * \sum_{i=1}^N ((y_p - y_i)^2)} \quad (2.27)$$

Keterangan :

Σ : penjumlahan

y_i : nilai aktual

y_p : nilai prediksi

N : jumlah data

2.1.8.2 Mean Absolute Error (MAE)

MAE mengukur besarnya rata-rata kesalahan dalam saat melakukan *forecasting*, tanpa mempertimbangkan arahnya. MAE mengukur akurasi untuk variabel yang *continue*. Atau penjelasan lainnya yaitu, MAE adalah rata-rata atas sampel verifikasi dari nilai absolut dari perbedaan antara nilai *forecasting* dan nilai observasi. MAE adalah skor linier yang berarti bahwa semua perbedaan individu diberi bobot yang sama dalam rata-rata. Agar lebih jelas bisa dilihat pada Persamaan 2.28 berikut.

$$MAE = \frac{1}{N} * \sum_{i=1}^N |y_p - y_i| \quad (2.28)$$

Keterangan :

Σ : penjumlahan

y_i : nilai aktual

y_p : nilai prediksi

N : jumlah data

2.1.8.3 Mean Absolute Percentage Error (MAPE)

MAPE merupakan teknik pengukuran seberapa akurat sistem *forecasting* yang sudah dibuat [16]. Pengukuran ini mengukur akurasi sebagai persentase, dan dapat dihitung sebagai *Mean Absolute Error* dalam bentuk persentase untuk setiap nilai prediksi periode waktu dikurangi nilai aktual dibagi dengan nilai aktual. MAPE adalah ukuran paling umum yang digunakan untuk memperkirakan kesalahan, dan bekerja paling baik jika tidak ada nilai ekstrem pada data. Agar lebih jelas bisa dilihat pada Persamaan 2.29 berikut.

$$MAPE = 100\% * \frac{1}{N} * \sum_{i=1}^N \frac{|y_p - y_i|}{y_i} \quad (2.29)$$

Keterangan :

Σ : penjumlahan

y_i : nilai aktual

y_p : nilai prediksi

N : jumlah data

RMSE, MAE, MAPE dapat digunakan bersama-sama untuk mendiagnosis variasi kesalahan dalam serangkaian prediksi. RMSE akan selalu lebih besar atau sama dengan MAE; semakin besar perbedaan di antara nilai RMSE dan MAE, semakin besar varians dalam kesalahan individu dalam sampel. Jika RMSE=MAE, maka semua kesalahan memiliki besar yang sama Baik MAE dan RMSE dapat berkisar dari 0 hingga tak hingga. Mereka adalah skor yang berorientasi negatif. Nilai yang lebih rendah atau yang mendekati 0 lebih baik.

Jika ada nilai titik data yang berbeda secara signifikan dari nilai *observasi* yang lain maka disebut *outliers*. RMSE dapat mendeteksi dan sensitif terhadap *outliers*, karena dalam perhitungannya terdapat kuadrat dan akar kuadrat. Sehingga, jika nilai RMSE sangat besar, dalam data tersebut ada *outliers*.

2.1.9 Pustaka *python*

Bagian ini menjelaskan pustaka (*library*) yang digunakan di dalam penelitian.

2.1.9.1 NumPy

NumPy adalah pustaka untuk bahasa pemrograman Python yang memberikan dukungan untuk melakukan fungsi dan operasi matematika tingkat tinggi. NumPy membantu pemrosesan data berbentuk larik. Tabel berikut merupakan rincian fungsi yang akan digunakan dari pustaka ini.

Tabel 2.1 Daftar *method* yang digunakan dari pustaka NumPy

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	<code>array</code>	<code>data</code>	<code>[]</code>	numpy array	Mengubah data larik atau <i>list</i> menjadi sebuah objek NumPy <i>array</i>

2.1.9.2 Pandas

Pandas (*Python for Data Analysis*) adalah pustaka dari Python yang digunakan untuk proses analisis data yang mencakup analisis, persiapan dan pembersihan data. Pandas juga menyediakan struktur data serta fungsi tingkat tinggi untuk membuat pekerjaan menyangkut data terstruktur seperti data tabular lebih cepat dan mudah. Tabel berikut merupakan rincian fungsi yang akan digunakan dari pustaka ini.

Tabel 2.2 Daftar *method* yang digunakan dari pustaka Pandas

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	<code>Dataframe</code>	<code>data</code>	<code>string</code>	-	wadah untuk meletakkan struktur data dalam bentuk tabel.

2	<code>pandas. DataFrame. to_csv</code>	data	string	csv	menyimpan dataframe dalam bentuk csv.
---	--	------	--------	-----	---------------------------------------

2.1.9.3 Matplotlib

Matplotlib adalah pustaka dari Python yang digunakan untuk melakukan visualisasi data seperti membuat grafik. Tabel berikut merupakan rincian fungsi yang akan digunakan dari pustaka ini.

Tabel 2.3 Daftar *method* yang digunakan dari pustaka Matplotlib

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	<code>pyplot.figure</code>	-	-	-	Melakukan inisialisasi grafik yang akan diatur parameternya
2	<code>pyplot.title</code>	-	string	-	Mengatur judul dari grafik yang akan ditampilkan
3	<code>pyplot.xlabel</code>	-	string	-	Mengatur nama sumbu x dari grafik yang akan ditampilkan
4	<code>pyplot.ylabel</code>	-	string	-	Mengatur nama sumbu y dari grafik yang akan ditampilkan
5	<code>pyplot.plot</code>	-	-	-	Membuat plot grafik dari data
6	<code>pyplot. savefig</code>	path file	string	png	Menyimpan grafik yang sudah diplot dalam bentuk png

7	<code>pyplot.show</code>	-	-	grafik	Menampilkan hasil data yang telah diplot
---	--------------------------	---	---	--------	--

2.1.9.4 *Tensorflow Keras*

Keras merupakan pustaka untuk *Neural Network* yang ditulis dengan bahasa Python. Pustaka ini menyediakan fitur seperti membuat model, membuat lapisan dan menambahkannya pada struktur *Neural Network*. Pustaka ini digunakan dengan fokus mempermudah pengembangan lebih dalam mengenai *deep learning*. Tabel berikut merupakan rincian fungsi yang akan digunakan dari pustaka ini.

Tabel 2.4 Daftar *method* yang digunakan dari pustaka Tensorflow Keras

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	<code>models.Sequential</code>	-	-	-	Melakukan inisialisasi model kosong bertipe <i>sequence</i> atau deret
2	<code>layers.lstm</code>	<code>unit,</code> <code>return_sequence,</code> <code>input_shape</code>	<code>int,</code> <code>boolean,</code> <code>int[]</code>	model	Membuat lapisan LSTM yang akan menerima hasil masukan
3	<code>layers.dense</code>	layer	object	-	Membuat lapisan <i>dense</i> yang akan menerima hasil masukan dari hasil proses lapisan sebelumnya yaitu LSTM dan memberikan hasil keluaran

4	<code>models.add</code>	layer	object	-	Menambahkan lapisan yang telah dibuat sebelumnya untuk disusun menjadi sebuah arsitektur LSTM
5	<code>models.compile</code>	loss	string	-	Melakukan kompilasi model yang telah dibuat dan menambahkan pengukuran error dengan menggunakan mse
6	<code>models.fit</code>	x_train y_train epochs batch_size validation_data callbacks	float[] float[] int int float[] object	-	Menjalankan proses NN dengan konfigurasi model yang telah dibuat dan memberikan data latih kepada model untuk proses pelatihan serta jumlah iterasi (<i>epochs</i>) pelatihan model
7	<code>Bidirectional</code>	layer	object	-	Membuat model LSTM dengan melakukan <i>forward pass</i> dan <i>backward pass</i>
8	<code>models.predict</code>	x_test	float[]	nilai pred	Memberikan data uji coba kepada model untuk dilakukan prediksi sesuai dengan hasil pelatihan models

9	<code>callbacks.</code> <code>EarlyStopping</code>	<code>monitor,</code> <code>patience,</code> <code>verbose</code>	<code>string,</code> <code>int, int</code>	<code>object</code>	Memberhentikan pelatihan data pada iterasi <i>epoch</i> setelah kinerja model berhenti meningkat.
---	---	---	---	---------------------	---

2.1.9.5 Statsmodels

Statsmodels adalah modul Python yang menyediakan kelas dan fungsi untuk estimasi banyak model statistik yang berbeda, serta untuk melakukan uji statistik, dan eksplorasi data statistik. Tabel berikut merupakan rincian fungsi yang akan digunakan dari pustaka ini.

Tabel 2.5 Daftar *method* yang digunakan dari Statsmodels

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	<code>statsmodels.</code> <code>tsa. arima.</code> <code>model.ARIMA</code>	<code>x_train</code> <code>order=(p,d,q)</code>	<code>int[]</code> <code>int</code>	<code>model</code>	Melakukan inisialisasi model kosong bertipe deret dengan menggunakan metode ARIMA
2	<code>statsmodels.</code> <code>tsa. arima.</code> <code>forecast</code>	-	-	nilai pred	Memberikan data uji coba kepada model ARIMA untuk dilakukan prediksi sesuai dengan hasil pelatihan model ARIMA

2.2 Tinjauan Studi

Pada Tabel 2.6 diberikan penjelasan perbandingan dari berbagai penelitian terkait dengan prediksi harga saham.

Tabel 2.6 Tinjauan Studi

No	Judul	Rumusan Masalah	Metode	Hasil
1	M. Nikou, G. Mansourfar, J. Bagherzadeh, "Stock Price Prediction using Deep Learning Algorithm and its Comparison with Machine Learning Algorithm," Intelligent Systems in Accounting Finance & Management, December 2019. [4]	Bagaimana hasil dari menggunakan metode LSTM, SVM, <i>Random Forest</i> , ANN?	LSTM, SVM, <i>Random Forest</i> , ANN.	Metode LSTM adalah yang terbaik dengan dibantu menggunakan metode <i>Dropout</i> untuk mengatasi <i>overfitting</i> dengan nilai $p = 0.8$ dan mendapatkan hasil MAE, MSE, RMSE berturut-turut ialah: 0.210350, 0.093969, 0.306543. Menggunakan <i>Dataset of iShares</i> MSCI United Kingdom, fitur yang digunakan hanya <i>Close</i> .

2	S. A. Wadi, M. Almasarweh, A. Alsaraireh, "Predicting Close Price Time Series Data Using ARIMA Model," Modern Applied Science, Vol. 12, No. 11, 2018. [3]	Berapakah nilai parameter ARIMA(p,d,q) untuk mencari RMSE yang terendah pada prediksi harga saham?	<i>Autoregressive Integrated Moving Average</i>	Dengan menggunakan data <i>close price</i> dan memprediksi <i>close price</i> . Mencoba nilai parameter ARIMA(p,d,q) antara 0 sampai 2 untuk mencari RMSE yang terendah. Dan mendapatkan RMSE 4.00, dengan nilai parameter ARIMA(2,1,1). Menggunakan <i>Amman Stock Exchange</i> , fitur yang digunakan hanya <i>Close</i> .
---	---	--	---	--

3	H. Roondiwala, H. Patel, S. Varma, "Predicting Stock Prices Using LSTM," International Journal of Science and Research, vol. 6, April 2017. [9]	Membandingkan hasil RMSE dengan menggunakan epoch, parameter fitur lebih dari 1 untuk memprediksi <i>Close price</i>	<i>Long Short Term Memory</i>	Menggunakan 2 macam <i>epoch</i> : 250 dan 500, menggunakan parameter variabel untuk melakukan <i>observasi</i> yang berbeda-beda pada dataset indeks NIFTY-50 rentang waktu yang digunakan dari 1 Januari 2011 sampai 31 Desember 2016 dan peneliti menyimpulkan bahwa dengan mengambil 4 set fitur (<i>High & Low & Open & Close</i>) dengan <i>epoch</i> 500 mencapai hasil terbaik dengan mendapatkan RMSE data <i>training</i> 0.00983 dan <i>testing</i> 0.00859.
---	---	--	-------------------------------	---

4	F. R. Pratama, "Implementasi Algoritme Random Forest untuk Prediksi Pergerakan Harga Saham," Dept. Informatika ITHB, 2017. [12]	Bagaimana pengaruh setiap indikator terhadap hasil uji dan pengaruh jumlah pohon keputusan yang dibuat terhadap hasil uji?	<i>Random Forest</i>	Memprediksi pergerakan harga saham dengan menggunakan algoritme <i>Random Forest</i> . Pada tahap uji, peneliti menggunakan parameter dengan banyak pohon keputusan yang dibuat: 10, 25, 50, 100, 200, 500, 1000. Maka, diperoleh nilai <i>f-measure</i> 88.47239 pada data saham UNVR dengan menggunakan 500 pohon keputusan dan 99.49253 pada data saham GGRM dengan menggunakan 1000 pohon keputusan. Fitur yang digunakan untuk melakukan penelitian ialah <i>Open, High, Low, Close, Volume</i> .
---	---	--	----------------------	--

2.3 Tinjauan Objek

Bagian ini akan dipaparkan objek yang digunakan terkait dengan prediksi harga saham.

2.3.1 *Time series Dataset*

Dataset Time series adalah kumpulan pengamatan yang diperoleh melalui pengurukan berulang berdasarkan waktu [20]. *Time series* dapat diambil pada variabel apapun yang berubah seiring waktu dengan satuan waktu yaitu jam, hari, minggu, bulan, tahun. Dalam berinvestasi, sangat umum untuk menggunakan data *time series* untuk mendapatkan harga sekuritas dari waktu ke waktu. Dapat dilacak

dalam jangka pendek, seperti harga sekuritas pada jam selama hari kerja, atau jangka panjang, seperti harga sekuritas pada penutupan pada hari terakhir setiap bulan selama lima tahun. *Time series* sangat sering diplot melalui grafik *line chart*. Dengan menggunakan dataset *time series*, bisa melakukan prediksi nilai masa depan dengan memahami, menafsirkan perubahan kronologi dalam nilai suatu variabel di masa lalu. Data *time series* dapat dikategorikan menjadi data yang *stationer* atau *non stationer*. Data *time series* dikatakan stasioner jika rata-rata dan variannya konstan. Stationer bisa dikatakan tidak ada *fluktuasi* yang signifikan terhadap suatu emiten harga saham. Terdapat 4 macam komponen untuk melakukan analisis *time series*:

2.3.1.1 *Trend*

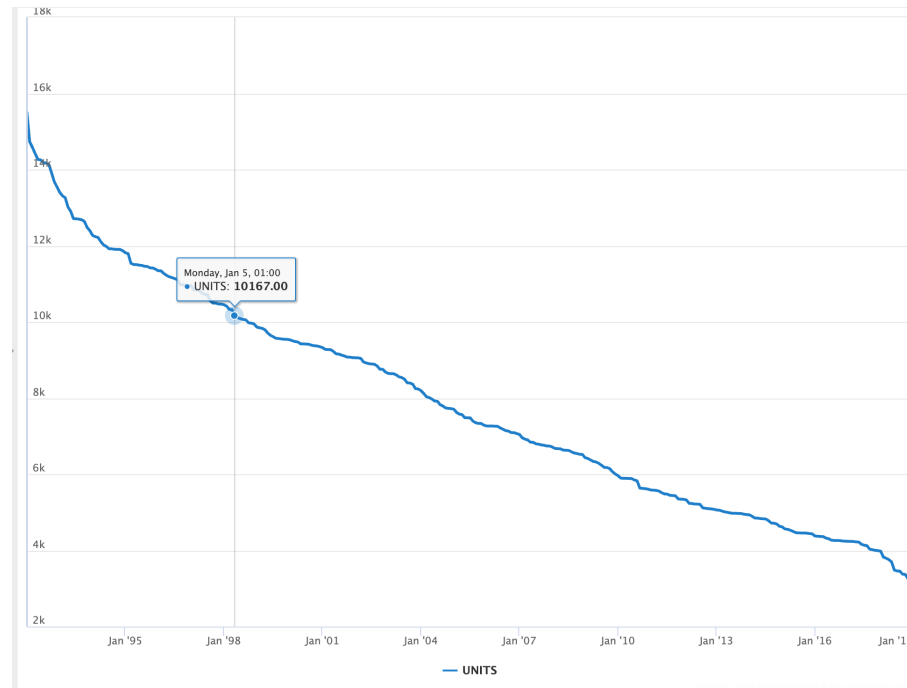
Trend dalam pola dalam data yang menunjukkan pergerakan suatu rangkaian ke nilai yang relatif lebih tinggi atau lebih rendah dalam jangka waktu yang lama. Dengan kata lain, *trend* diamati ketika ada peningkatan atau penurunan dalam *time series* [20]. *Trend* biasanya terjadi untuk beberapa waktu dan kemudian menghilang. Misalnya, beberapa lagu baru datang, menjadi trending untuk sementara waktu, dan kemudian menghilang, dan ada kemungkinan bahwa itu menjadi *trend* lagi. Sebuah *trend* bisa berupa:

1. *Uptrend*

Analisis *time series* menunjukkan pola umum yang naik maka itu adalah *Uptrend*.

2. *Downtrend*

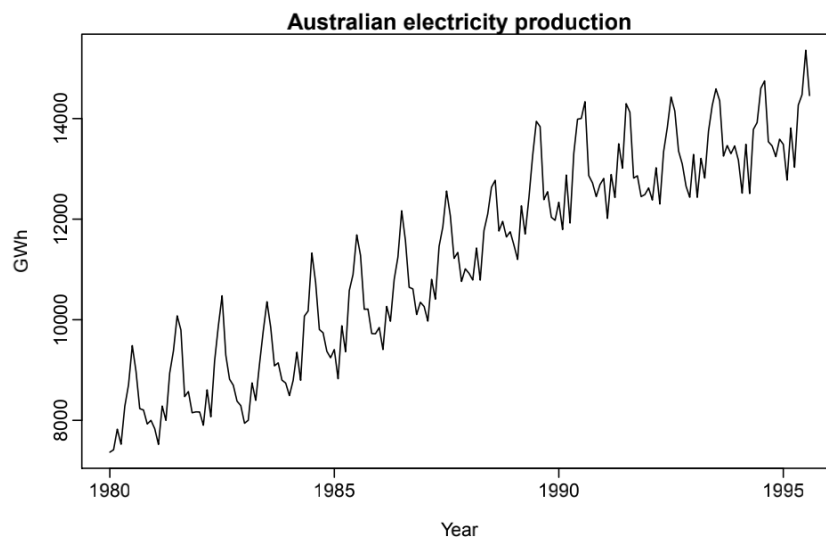
Analisis *time series* menunjukkan pola yang turun maka itu adalah *Downtrend*, seperti yang ada pada Gambar 2.9.



Gambar 2.9 Downtrend

2.3.1.2 Seasonal

Seasonal adalah karakteristik pola dalam data *time series* dimana data mengalami perubahan yang teratur dan dapat diprediksi berulang setiap tahun kalender, seperti yang ada pada Gambar 2.10.

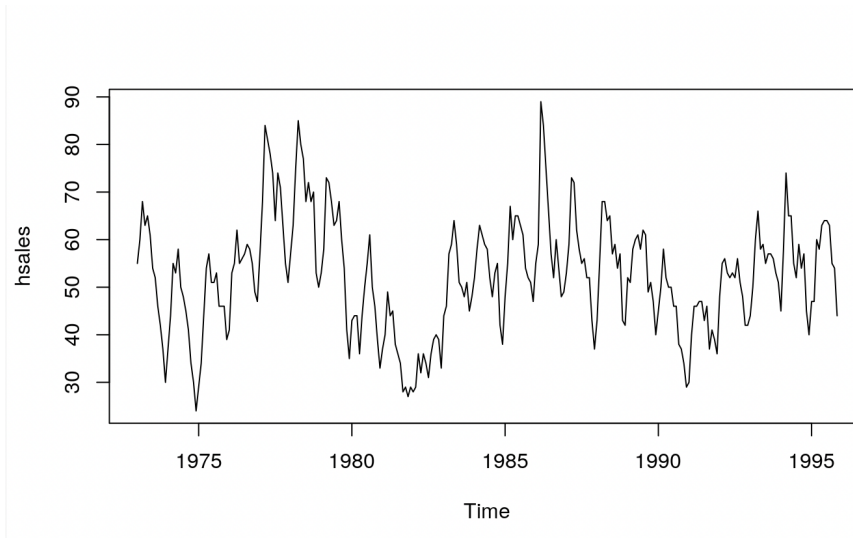


Gambar 2.10 Seasonal Time series [20]

2.3.1.3 Cyclical

Cyclical adalah karakteristik pola dalam data *time series* dimana siklus terjadi ketika data menunjukkan naik dan turun yang tidak tetap. Fluktuasi ini

biasanya disebabkan oleh kondisi ekonomi. Durasi fluktuasi ini biasanya minimal 2 tahun [19]. Seperti pada Gambar 2.11 merupakan data penjualan rumah sebelum tahun 2000. Pada tahun 1975, penjualan rumah mengalami penurunan, dan terjadi lagi saat tahun 1982, ini yang dinamakan pola data *time series cyclical*.



Gambar 2.11 Cyclical Time series [20]

2.3.2 *Forecasting Horizon*

Forecasting arti umumnya adalah untuk memprediksi sesuatu di masa yang akan datang. Banyak bidang yang memakai *forecasting* ini seperti bidang teknologi, ekonomi dan bisnis. Rencana bisnis bisa berjalan efektif di masa depan saat keadaan yang berlaku sesuai dengan perkiraan/peramalan di masa mendatang. Hal ini tidak bisa menjadi acuan pasti, tetapi *forecasting* bisa memberi gambaran agar ekonomi suatu perusahaan dapat dipersiapkan.

Forecasting Horizon merupakan prediksi yang diklasifikasikan menurut periode waktu dan digolongkan menjadi *short-term*, *medium-term*, dan *long-term* [21]. *Short-term forecasting* meramalkan kejadian hanya beberapa periode waktu ke depan (harian, mingguan). *Mid-term forecasting* meramalkan sebulan sampai 3 bulan ke depan, dan *long-term forecasting* bisa meramalkan 3 bulan sampai setahun kedepan.

2.3.3 *Dataset Saham*

Data yang digunakan untuk penelitian ini berupa data-data dari sektor saham yang bersumber dari *Yahoo Finance*. Daftar perusahaan yang digunakan sebagai dataset dalam penelitian ini ialah: PT. Unilever (UNVR), PT. Gudang Garam (GGRM) sebagai perwakilan dari saham sektor FMCG yang berstatus *blue chip*. Sementara, sebagai perwakilan dari saham sektor FMCG yang berstatus

gorengan, antara lain: PT. Prasadha Aneka Niaga(PSDN). Dataset harga saham memiliki 7 variabel yaitu *Date*, *Open*, *High*, *Low*, *Close*, *Adj Close*, *Volume*, contoh data dapat dilihat pada Gambar 2.12.

Date	Open	High	Low	Close*	Adj Close**	Volume
Dec 03, 2021	6,588.01	6,600.17	6,536.91	6,538.51	6,538.51	-
Dec 02, 2021	6,517.34	6,586.90	6,484.58	6,583.82	6,583.82	210,837,500
Dec 01, 2021	6,544.52	6,593.07	6,494.50	6,507.68	6,507.68	195,068,500
Nov 30, 2021	6,605.80	6,647.48	6,533.93	6,533.93	6,533.93	216,798,600
Nov 29, 2021	6,552.80	6,617.29	6,487.75	6,608.29	6,608.29	-
Nov 26, 2021	6,688.57	6,691.71	6,544.90	6,561.55	6,561.55	242,523,500

Gambar 2.12 Data Sampling [19]

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis masalah yang diatasi beserta metode yang digunakan dan hasil yang akan ditampilkan.

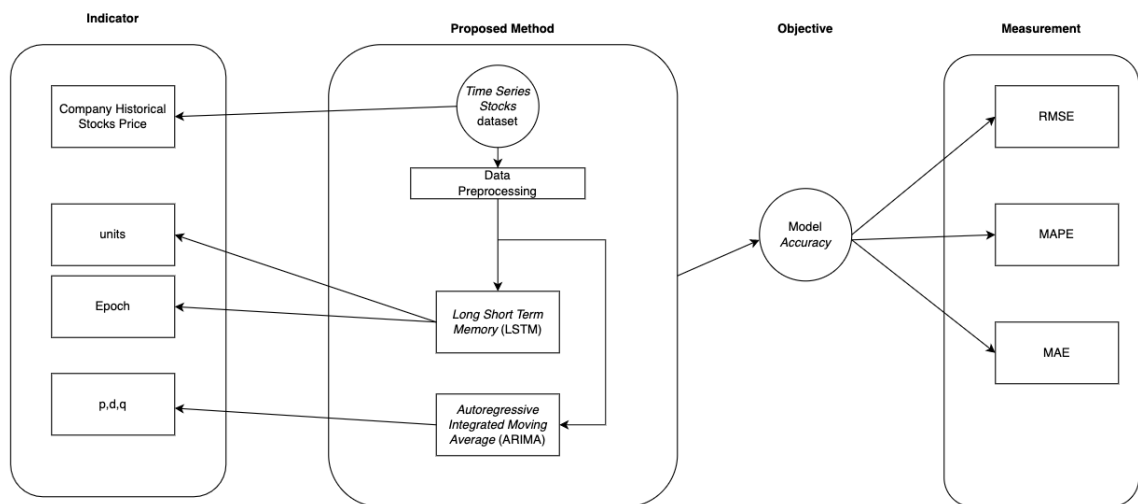
3.1 Analisis Masalah

Seperti yang dijelaskan pada bab 1, beberapa penelitian untuk memprediksi harga saham telah menggunakan beberapa metode *forecasting* oleh *machine learning*, *deep learning* ataupun *traditional forecasting*. Pada penelitian ini akan membangun sebuah sistem prediksi harga saham dengan menggunakan dataset harga saham dari sektor FMCG di Indonesia yang diambil dari *Yahoo Finance* dalam rentang waktu yang telah ditentukan sesuai jangka waktu prediksi.

Penelitian ini akan membangun, menguji, dan membandingkan antara model *Bidirectional Long Short Term Memory* (Bi-LSTM) dan model *Autoregressive Integrated Moving Average* (ARIMA). Alasan dengan menggunakan Bi-LSTM adalah setelah dilakukan *review* jurnal pada bab 1, dapat disimpulkan bahwa Bi-LSTM adalah metode *Machine Learning* atau *Deep Learning* yang dapat memprediksi harga saham dengan hasil yang terbaik. Kemudian, alasan dengan membandingkan dan menggunakan metode ARIMA. Karena, ARIMA merupakan metode *forecasting* yang mempelajari pola berdasarkan nilai masa lalu dan sudah terbukti sering digunakan untuk memprediksi data yang *time series*. *Input* pada sistem ini adalah variabel-variabel dari dataset yang sudah ditentukan, seperti *Date*, *Open*, *High*, *Low*, *Close*, *Volume*. *Output* pada sistem ini berupa harga dari hasil prediksi dari variabel *Close* dan mendapatkan hasil pengukuran berupa RMSE, MAE, MAPE.

3.2 Kerangka Pemikiran

Berikut ini adalah kerangka pemikiran dari metode yang diusulkan untuk membangun sistem prediksi harga saham.



Gambar 3.1 Kerangka Pemikiran

Berikut akan dijelaskan setiap bagian yang ada pada Gambar 3.1:

1. *Indicators* adalah variabel-variabel yang digunakan dan akan memengaruhi hasil akhir. *Indicators* yang digunakan dalam penelitian ini adalah sebagai berikut.

(a) *Company Historical Stocks Price*

Company Historical Stocks Price merupakan berbagai macam dataset historis harga saham perusahaan yang dari sektor FMCG, yang bersumber dari *Yahoo Finance*. Emiten yang digunakan dalam penelitian ini ialah: PT. Unilever (UNVR), PT. Gudang Garam (GGRM) sebagai perwakilan dari saham sektor FMCG yang berstatus *blue chip*. Sementara, sebagai perwakilan dari saham sektor FMCG yang berstatus gorengan, antara lain: PT. Prasadha Aneka Niaga (PSDN).

Rentang waktu yang digunakan dalam prediksi harga saham:

- i. *Short-term forecasting* menggunakan 3 bulan dari keseluruhan data. Rentang waktu yang digunakan dari 1 Januari 2017 sampai 31 Maret 2017, dengan jumlah 62 data.
- ii. *Mid-term forecasting* menggunakan 1 tahun dari keseluruhan data [2]. Rentang waktu yang digunakan dari 1 Januari 2017 sampai 31 Desember 2017, dengan jumlah 253 data.
- iii. *Long-term forecasting* menggunakan 5 tahun dari keseluruhan data [6]. Rentang waktu yang digunakan 1 Januari 2017 sampai 31 Desember 2021, dengan jumlah 1261 data.

(b) *Units*

Units merupakan jumlah atau ukuran dimensi dari *hidden state* atau *output*, atau yang bisa dianggap sebagai *neuron* yang berada di *hidden layer*. Pada penelitian ini akan menggunakan jumlah sebesar 10, 50, 128. Karena pada penelitian [4, 6, 9] menggunakan jumlah *unit* yang sudah ditentukan sebelumnya.

(c) *Epoch*

Epoch merupakan iterasi 1 siklus program selesai dijalankan. Semakin besar *epoch*, semakin bisa meningkatkan akurasi, namun akan semakin lama prosesnya berjalan. *Epoch* yang akan digunakan dalam penelitian ini adalah sebesar 10, 100, 1000, karena pada Penelitian [4], *epoch* yang digunakan sebesar 100, sedangkan pada Penelitian [6] menggunakan beragam *epoch*.

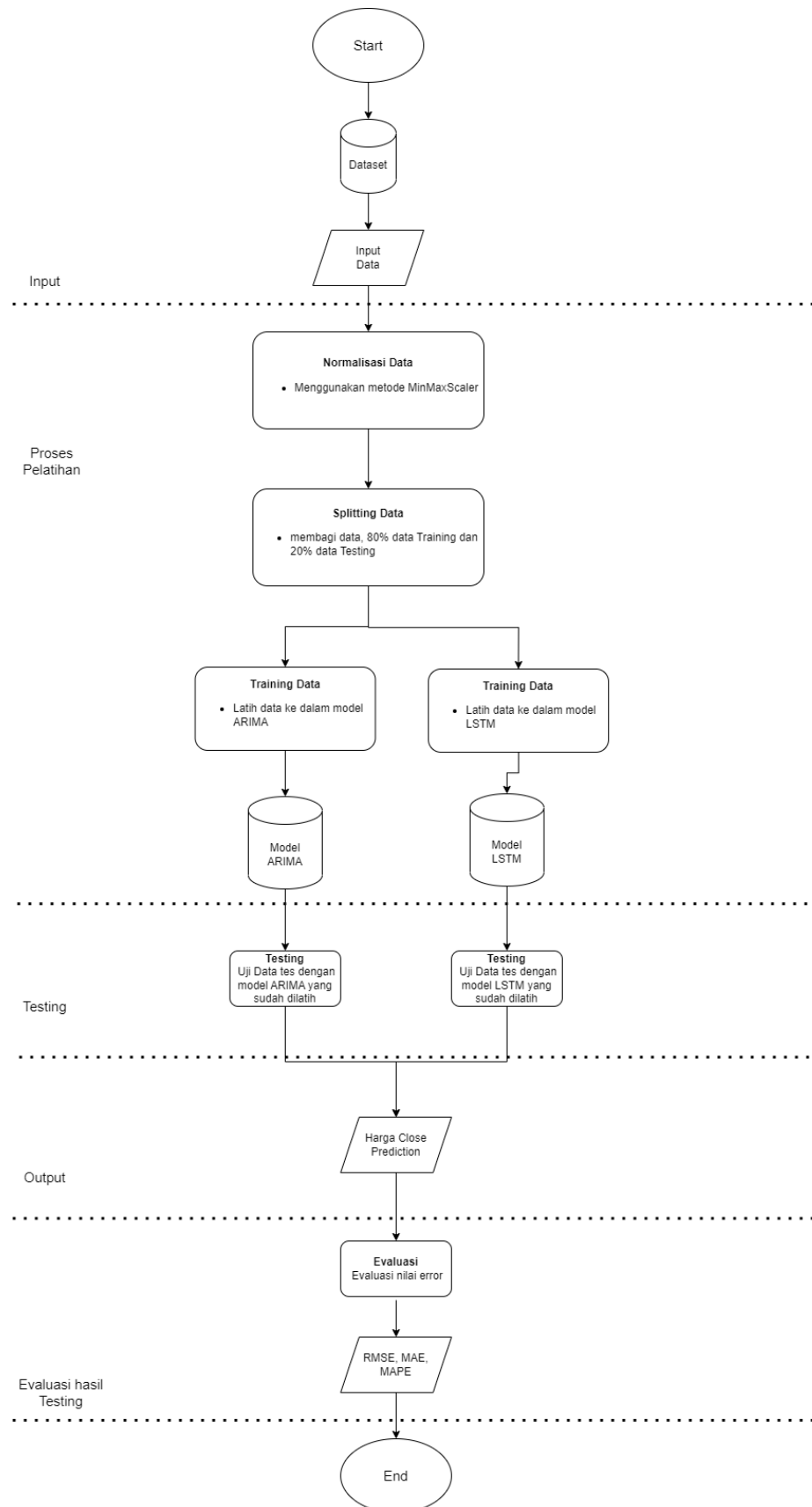
(d) p,d,q

Ordo p merupakan nilai parameter untuk *Autoregressive*(AR), ordo d merupakan nilai parameter untuk *Integrated* (I), dan ordo q merupakan nilai parameter untuk *Moving Average* (MA). Pada penelitian ini menggunakan nilai ordo d adalah 1, karena dengan melakukan 1st difference sudah cukup membuat data menjadi stasioner [2], dan menggunakan nilai ordo p dan q dengan rentang 1 sampai 2.

2. *Proposed Method* adalah bagian yang menjelaskan proses penelitian dari awal hingga akhir. Setelah mendapatkan dataset saham maka dilakukan *preprocessing*, dengan menghapus variabel yang tidak terpakai yang terdapat dalam dataset seperti *volume*. Setelah itu akan dibuat model dengan metode *Autoregressive Integrated Moving Average* dan *Bidirectional Long Short Term Memory*.
3. *Objectives* adalah bagian yang menjelaskan acuan pengukuran. Penelitian ini menggunakan acuan performa dari model yang sudah dibuat.
4. *Measurement* adalah bagian yang menjelaskan ukuran yang dipakai pada bagian *objectives*, Penelitian ini menggunakan RMSE, MAPE, dan MAE.

3.3 Urutan Proses Global

Pada Gambar 3.2 diberikan *flowchart* mengenai urutan proses dalam penelitian ini.



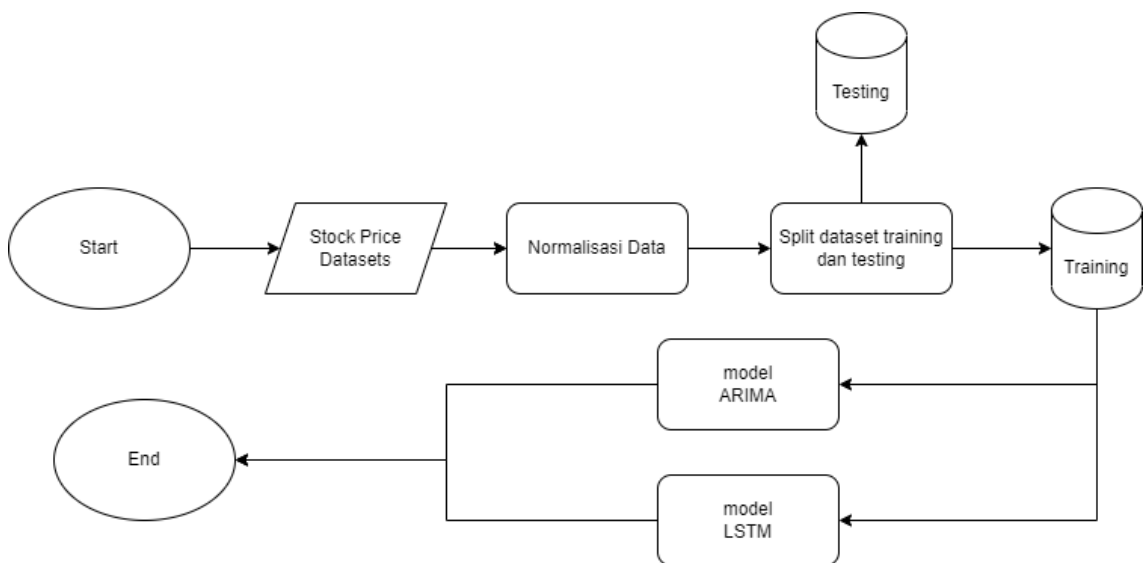
Gambar 3.2 Urutan Proses Global

Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang terlihat pada Gambar 3.2:

Dalam penelitian ini hal pertama adalah *input*-kan *dataset* ke dalam program sesuai dengan *dataset* yang akan diteliti sesuai perusahaan dan jangka waktu prediksi *long*, *mid*, *short*. Setelah berhasil di-*input*, hal selanjutnya adalah melakukan normalisasi data dengan menggunakan metode *MinMaxScaler*. Setelah itu, data akan dibagi menjadi data latih untuk proses pelatihan model dan data tes untuk proses pengujian model. Model prediksi harga saham ini dibangun menggunakan metode ARIMA dan Bi-LSTM dengan bertujuan membandingkan hasil model prediksi dari kedua metode tersebut. Disaat model tersebut dipakai untuk memprediksi data yang ada di data tes. Hasil prediksi tersebut dievaluasi dengan menggunakan RMSE, MAE, MAPE.

3.3.1 Proses Training

Pada penelitian ini, proses *training* model digambarkan pada Gambar 3.3.



Gambar 3.3 Flowchart proses training

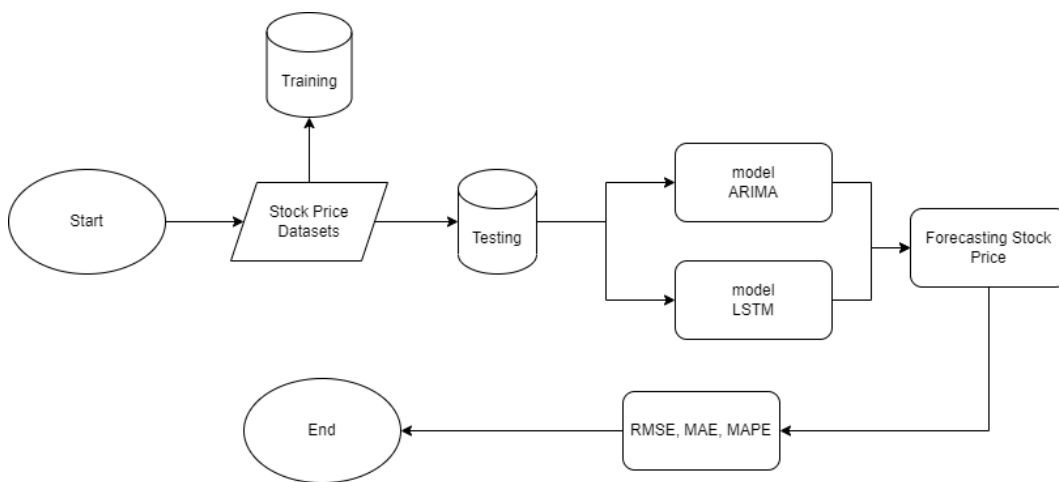
Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang terlihat pada Gambar 3.3:

1. Dengan menggunakan dataset dari perusahaan: GGRM, UNVR, PSDN.
2. Dataset yang digunakan memiliki 5 variabel, yaitu *Date*, *Open*, *High*, *Low*, *Close*, *Volume*.
3. Melakukan Normalisasi data dengan menggunakan *MinMaxScaler*.

4. Membagi dataset sebesar 80% untuk proses training metode ARIMA dan Bi-LSTM. *Long-term* akan menghasilkan 1009 data latih, *mid-term* akan menghasilkan 203 data latih dan *short-term* akan menghasilkan 50 data latih.
5. Melatih Model ARIMA dan Bi-LSTM dengan menggunakan data latih yang sudah di bagi.
6. Terakhir, menghasilkan Model ARIMA dan Bi-LSTM yang sudah dilatih dengan data latih.

3.3.2 Proses Testing

Pada penelitian ini, proses *testing* model diGambarkan pada Gambar 3.4.



Gambar 3.4 Flowchart proses *Testing*

Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang terlihat pada Gambar 3.3:

1. Dengan menggunakan dataset yang setelah di *splitting* dengan sebesar 20% dari jumlah seluruh data. *Long-term* akan menghasilkan 252 data uji, *mid-term* akan menghasilkan 50 data uji dan *short-term* akan menghasilkan 12 data uji.
2. Selanjutnya, akan dilakukan *testing* terhadap data *testing* yang sudah menghasilkan model ARIMA dan model Bi-LSTM saat proses training.
3. Terakhir, akan melakukan prediksi dengan setiap data *testing* pada setiap dataset dan diukur performa dari masing-masing model (ARIMA, Bi-LSTM) dengan menggunakan RMSE, MAE, MAPE.

3.4 Analisis Manual

Pada bagian ini akan dijelaskan analisis tahapan proses yang dilakukan dalam sistem dengan melakukan analisis dan perhitungan secara manual.

3.4.1 Dataset

Dalam penelitian ini, dataset yang digunakan berjumlah 3 macam dataset historis harga saham perusahaan. Macam-macam dataset harga saham yang berasal dari: PT. Unilever(UNVR), PT. Gudang Garam(GGRM) sebagai perwakilan dari saham sektor FMCG yang berstatus *blue chip*. Sementara, sebagai perwakilan dari saham sektor FMCG yang berstatus gorengan, antara lain: PT. Prasadha Aneka Niaga(PSDN). Dataset yang digunakan dalam penelitian ini semua dengan format *Comma Separated Value (CSV)*. Tabel 3.1 contoh matriks data saham.

Tabel 3.1 Tabel Matriks Data Saham

Date	Open	High	Low	Close	Volume
03/01/2022	7325	7400	7300	7325	54287400
04/01/2022	7325	7450	7325	7400	70624000
05/01/2022	7450	7525	7375	7450	76164900
06/01/2022	7500	7525	7425	7475	63657100
07/01/2022	7550	7700	7500	7650	143433300

3.4.2 Normalisasi *Min-max*

Pada tahap ini, data akan dilakukan proses normalisasi dengan menggunakan *Min-Max Scaler*. Normalisasi data diperlukan, karena Bi-LSTM sangat sensitif terhadap data *input* yang relatif besar. MinMaxScaler akan mengubah nilai rentang data antara 0 sampai 1. Berikut adalah contoh perhitungan normalisasi *Min-max*.

Dengan menggunakan data kolom *Close*.

$$x_i = 7325, \min(x) = 7325, \max(x) = 7650 \quad (3.1)$$

$$x_{scaled} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3.2)$$

$$x_{scaled} = \frac{7325 - 7325}{7650 - 7325} \quad (3.3)$$

$$x_{scaled} = 0 \quad (3.4)$$

Pada bagian 3.1 diketahui data kondisi yang digunakan dengan menggunakan kolom *Close* untuk menghitung *Min-max* pada kolom *Close*. Pada bagian tersebut nilai x ke- i adalah 7325, nilai minimum dari kolom *Close* adalah 7325, dan nilai maksimum dari kolom *close* adalah 7650. Setelah semua kolom dilakukan normalisasi dengan *Min-max* maka menghasilkan nilai seperti Tabel 3.2.

Tabel 3.2 Tabel Matriks Data Saham

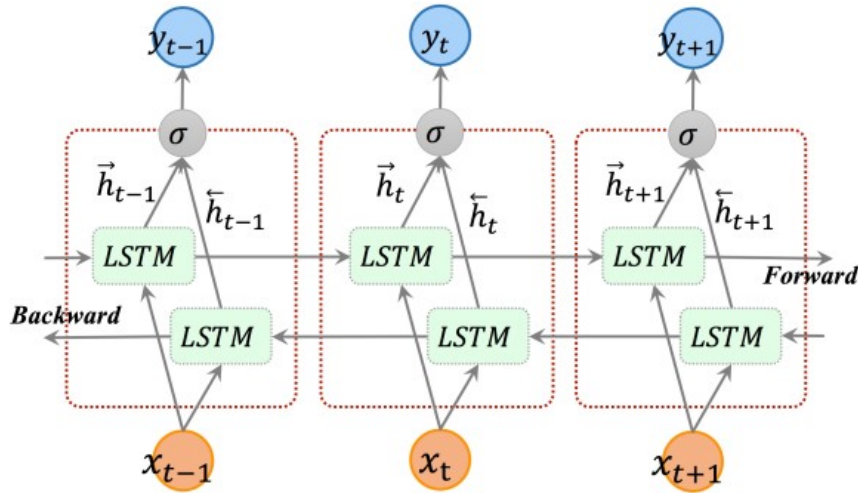
Date	Open	High	Low	Close	Volume
03/01/2022	0	0	0	0	0
04/01/2022	0	0.16	0.12	0.23	0.18
05/01/2022	0.55	0.41	0.37	0.38	0.24
06/01/2022	0.77	0.41	0.62	0.46	0.10
07/01/2022	1	1	1	1	1

3.4.3 *Splitting Dataset untuk Training dan Testing*

Pada tahap ini dilakukan pembagian dataset untuk proses *training* dan *testing* yang akan digunakan untuk membuat model ARIMA dan Bi-LSTM.

Pembagian dataset adalah 80% untuk data *training* dan 20% untuk data *testing*. *Training* dilakukan untuk melatih algoritma dalam mencari model yang sesuai dengan data yang ada, sementara *testing* dilakukan untuk mengetahui performa dengan mengetahui pengukuran *error*/kesalahan model yang sudah dilakukan pada suatu prediksi.

3.4.4 Perhitungan Long Short Term Memory



Gambar 3.5 Bidirectional LSTM with Unit

Pada Gambar 3.5 merupakan arsitektur pada LSTM dengan menggunakan 2 arah (*forward*, *backward*) atau *bidirectional* dengan *timestep* saat $t-1$, *timestep* saat t , *timestep* saat $t+1$. *Forward* adalah input dari *timestep* terkecil ke *timestep* terbesar dengan melakukan proses ke arah depan (alur maju) dengan melakukan perhitungan *gate-gate* dari Persamaan *forward Layer* (Persamaan 2.11-2.16). Sementara, *backward* adalah input dari *timestep* terbesar ke *timestep* terkecil dengan melakukan proses ke arah belakang (alur mundur) dengan melakukan perhitungan Persamaan *backward Layer* (Persamaan 2.17-2.25) untuk mendapatkan nilai baru untuk *update weight* dan *bias*. Setiap *cell LSTM* (Gambar 2.8) terdapat formula untuk perhitungan mencari *Forget*, *Input*, *Memory Gate*, *Cell State* dan *Output*.

Pada perhitungan LSTM, akan dilakukan sekali proses *epoch* dan 2 *unit LSTM*. Dalam satu hari harga saham meliputi fitur *Open*, *High*, *Low*, *Close*, *Volume* akan menjadi *Input* X_t pada Tabel 3.2 yang dibutuhkan untuk menghasilkan *output* H_t . Nilai bobot akan diinisialisasi secara acak dan nilai bias diinisialisasikan nol.

3.4.4.1 Forward Layer LSTM

Pada bagian ini akan dijelaskan perhitungan t kedua adalah perhitungan LSTM pada tanggal 4 Januari 2022, dilakukannya perhitungan t kedua, karena t pertama pada kasus ini akan menghasilkan H_1 bernilai $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$, karena input X_t pada

3 Januari 2022 bernilai $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. Menggunakan nilai bobot dengan rentang -0.4

sampai 0.4 yang diambil secara acak.

Perhitungan t kedua, untuk mendapatkan nilai Forget Gate (F_2) dengan menggunakan Persamaan 2.11 :

$$H_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$W_f = \begin{pmatrix} 0.21000 & 0.06000 \\ 0.02300 & 0.04000 \end{pmatrix}$$

$$U_f = \begin{pmatrix} 0.02000 & -0.01000 & 0.30000 & 0.23000 & 0.21000 \\ 0.12000 & 0.31000 & -0.23000 & 0.25000 & -0.01000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix}$$

$$b_f = 0$$

$$F_t = \sigma(W_f * H_{t-1} + U_f * X_t + b_f)$$

$$F_t = \sigma\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.12510 \\ 0.07770 \end{pmatrix} + 0\right)$$

$$\sigma(0.12510) = \frac{1}{1+e^{-0.12510}}$$

$$\sigma(0.12510) = 0.53120$$

$$\sigma(0.07770) = \frac{1}{1+e^{-0.07770}}$$

$$\sigma(0.07770) = 0.51941$$

$$F_t = \begin{pmatrix} 0.53120 \\ 0.51940 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t kedua pada tanggal 4 Januari 2022. Terdapat bobot W_f yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot U_f sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai input X_t . Dengan mendapatkan nilai *Forget gate* t kedua yaitu $\begin{pmatrix} 0.53120 \\ 0.51940 \end{pmatrix}$

Perhitungan t kedua, untuk mendapatkan nilai *Input Gate* (I_2) dengan menggunakan Persamaan 2.12:

$$H_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$W_i = \begin{pmatrix} 0.12000 & -0.20000 \\ 0.21000 & 0.02300 \end{pmatrix}$$

$$U_i = \begin{pmatrix} 0.13000 & 0.21000 & 0.02000 & -0.21000 & 0.3000 \\ -0.23000 & 0.15000 & -0.01000 & 0.12000 & 0.31000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix}$$

$$b_i = 0$$

$$I_t = \sigma(W_i * H_{t-1} + U_i * X_t + b_i)$$

$$I_t = \sigma \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.13000 & 0.21000 & 0.02000 & -0.21000 & 0.3000 \\ -0.23000 & 0.15000 & -0.01000 & 0.12000 & 0.31000 \end{pmatrix} * \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix} + 0 \right)$$

$$I_t = \sigma \left(0 + \begin{pmatrix} 0.04170 \\ 0.10620 \end{pmatrix} \right)$$

$$\sigma(0.04170) = \frac{1}{1+e^{-0.04170}}$$

$$\sigma(0.04170) = 0.51042$$

$$\sigma(0.10620) = \frac{1}{1+e^{-0.10620}}$$

$$\sigma(0.10620) = 0.52652$$

$$I_t = \begin{pmatrix} 0.51040 \\ 0.52650 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t kedua pada tanggal 4 Januari 2022. Terdapat bobot W_i yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot U_i sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai

input X_t . Dengan mendapatkan nilai *Input gate* t kedua yaitu $\begin{pmatrix} 0.51040 \\ 0.52650 \end{pmatrix}$

Perhitungan t kedua, untuk mendapatkan nilai *Memory Gate* (Z_2) dengan menggunakan Persamaan 2.13:

$$H_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$W_c = \begin{pmatrix} 0.12000 & -0.2000 \\ 0.06000 & 0.21000 \end{pmatrix}$$

$$U_c = \begin{pmatrix} 0.12000 & 0.08000 & 0.012000 & -0.01000 & -0.01000 \\ -0.20000 & 0.02000 & 0.12000 & -0.20000 & 0.08000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix}$$

$$b_c = 0$$

$$Z_t = \tanh(W_c * H_{t-1} + U_c * X_t + b_c)$$

$$z_t = \tanh\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.12000 & 0.08000 & 0.01200 & -0.01000 & -0.01000 \\ -0.20000 & 0.02000 & 0.12000 & -0.20000 & 0.08000 \end{pmatrix} * \right.$$

$$\left. \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix} + 0\right)$$

$$Z_t = \sigma\left(0 + \begin{pmatrix} 0.01010 \\ -0.01400 \end{pmatrix}\right)$$

$$\tanh(0.01010) = \frac{e^{0.01010} - e^{-0.01010}}{e^{0.01010} + e^{-0.01010}}$$

$$\tanh(0.01010) = 0.01010$$

$$\tanh(-0.01400) = \frac{e^{-0.01400} - e^{-(-0.01400)}}{e^{-0.01400} + e^{-(-0.01400)}}$$

$$\tanh(-0.01400) = -0.01400$$

$$Z_t = \begin{pmatrix} 0.01010 \\ -0.01400 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t kedua pada tanggal 4 Januari 2022 untuk mencari *Memory gate* (Z_2). Terdapat bobot W_c yang di inisialisasikan

sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot U_c sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai *input* X_t . Dengan mendapatkan nilai

Memory gate t kedua yaitu $\begin{pmatrix} 0.01010 \\ -0.01400 \end{pmatrix}$

Perhitungan t kedua, untuk mendapatkan nilai *Output Gate* (O_2) dengan menggunakan Persamaan 2.15 :

$$H_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$W_o = \begin{pmatrix} -0.32000 & 0.15000 \\ 0.21000 & 0.06000 \end{pmatrix}$$

$$U_o = \begin{pmatrix} 0.12000 & -0.32000 & 0.22000 & -0.01000 & 0.11000 \\ 0.2000 & 0.12000 & 0.12000 & 0.21000 & 0.17000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix}$$

$$b_o = 0$$

$$O_t = \sigma(W_o * H_{t-1} + U_o * X_t + b_o)$$

$$O_t = \sigma \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.12000 & -0.32000 & 0.22000 & -0.01000 & 0.11000 \\ 0.20000 & 0.12000 & 0.12000 & 0.21000 & 0.17000 \end{pmatrix} * \begin{pmatrix} 0 \\ 0.16000 \\ 0.12000 \\ 0.23000 \\ 0.18000 \end{pmatrix} + 0 \right)$$

$$O_t = \sigma \left(0 + \begin{pmatrix} -0.00730 \\ 0.11250 \end{pmatrix} \right)$$

$$\sigma(-0.00730) = \frac{1}{1+e^{-(-0.00730)}}$$

$$\sigma(-0.00730) = 0.49820$$

$$\sigma(0.11250) = \frac{1}{1+e^{-0.11250}}$$

$$\sigma(0.11250) = 0.52809$$

$$O_t = \begin{pmatrix} 0.49320 \\ 0.52810 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t kedua untuk mencari *Output gate* (O_t). Terdapat bobot W_o yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot U_o sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai *input* X_t . Dengan mendapatkan nilai *Output gate* t kedua yaitu

$$\begin{pmatrix} 0.49320 \\ 0.52810 \end{pmatrix}$$

Setelah mendapatkan nilai *Forget gate*, *Input gate*, *Memory gate*, *Output gate*, maka akan dilakukan perhitungan untuk mendapatkan nilai *Cell state*, dan nilai *Output* H_t pada t kedua.

Perhitungan t kedua, Cell State(C_2) dan Output (H_2) dengan menggunakan Persamaan 2.14 dan 2.16:

$$C_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$F_t = \begin{pmatrix} 0.53120 \\ 0.51940 \end{pmatrix}$$

$$I_t = \begin{pmatrix} 0.51040 \\ 0.52650 \end{pmatrix}$$

$$Z_t = \begin{pmatrix} 0.01010 \\ -0.0140 \end{pmatrix}$$

$$O_t = \begin{pmatrix} 0.49820 \\ 0.52810 \end{pmatrix}$$

$$C_t = F_t \otimes C_{t-1} + I_t \otimes Z_t$$

$$C_t = \begin{pmatrix} 0.53120 \\ 0.51940 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.51040 \\ 0.52650 \end{pmatrix} \otimes \begin{pmatrix} 0.01010 \\ -0.01400 \end{pmatrix}$$

$$C_t = \begin{pmatrix} 0.00520 \\ -0.00740 \end{pmatrix}$$

$$\tanh(C_t) = \begin{pmatrix} 0.00520 \\ -0.00740 \end{pmatrix}$$

$$H_t = O_t \otimes \tanh(C_t)$$

$$H_t = \begin{pmatrix} 0.49820 \\ 0.52810 \end{pmatrix} \otimes \begin{pmatrix} 0.00520 \\ -0.00740 \end{pmatrix}$$

$$H_t = \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix}$$

Pada perhitungan di atas dengan menggunakan pemodelan dari LSTM pada

t kedua maka mendapatkan nilai *Output* $H_t = \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix}$. Hasil dari H_2 dan C_2 bisa digunakan untuk melakukan perhitungan t ketiga. Pada tahap selanjutnya akan dibahas mengenai perhitungan t ketiga.

Perhitungan t ketiga, untuk mendapatkan nilai *Forget Gate* (F_3) dengan menggunakan Persamaan 2.11 :

$$H_2 = \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix}$$

$$W_f = \begin{pmatrix} 0.21000 & 0.06000 \\ 0.02300 & 0.04000 \end{pmatrix}$$

$$U_f = \begin{pmatrix} 0.02000 & -0.01000 & 0.3000 & 0.23000 & 0.21000 \\ 0.12000 & 0.31000 & -0.23000 & 0.25000 & -0.01000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix}$$

$$b_f = 0$$

$$F_t = \sigma(W_f * H_{t-1} + U_f * X_t + b_f)$$

$$F_t = \sigma \left(\begin{pmatrix} 0.21000 & 0.06000 \\ 0.02300 & 0.04000 \end{pmatrix} * \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix} + \begin{pmatrix} 0.25570 \\ 0.14120 \end{pmatrix} + 0 \right)$$

$$F_t = \sigma \left(\begin{pmatrix} 0.00030 \\ -0.0001 \end{pmatrix} + \begin{pmatrix} 0.25570 \\ 0.14120 \end{pmatrix} \right)$$

$$F_t = \sigma \left(\begin{pmatrix} 0.25600 \\ 0.14110 \end{pmatrix} \right)$$

$$\sigma(0.25600) = \frac{1}{1+e^{-0.25600}}$$

$$\sigma(0.25600) = 0.56370$$

$$\sigma(0.1411) = \frac{1}{1+e^{-0.1411}}$$

$$\sigma(0.1411) = 0.5352$$

$$F_3 = \begin{pmatrix} 0.56370 \\ 0.53520 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t ketiga pada tanggal 5 Januari 2022. Terdapat bobot W_f yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} atau H_2 . Dan, terdapat bobot

U_f sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai input X_t . Dengan mendapatkan nilai *Forget gate* t ketiga yaitu

$$\begin{pmatrix} 0.56370 \\ 0.53520 \end{pmatrix}$$

Perhitungan t ketiga, untuk mendapatkan nilai *Input Gate* (I_3) dengan menggunakan Persamaan 2.12 :

$$H_2 = \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix}$$

$$W_i = \begin{pmatrix} 0.12000 & -0.2000 \\ 0.06000 & 0.21000 \end{pmatrix}$$

$$U_i = \begin{pmatrix} 0.13000 & 0.21000 & 0.02000 & -0.21000 & 0.30000 \\ -0.23000 & 0.15000 & -0.01000 & 0.12000 & 0.31000 \end{pmatrix}$$

$$X_3 = \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix}$$

$$b_i = 0$$

$$I_t = \sigma(W_i * H_2 + U_i * X_2 + b_i)$$

$$I_t = \sigma \left(\begin{pmatrix} 0.12000 & -0.2000 \\ 0.06000 & 0.21000 \end{pmatrix} * \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix} + \right.$$

$$\left. \begin{pmatrix} 0.13000 & 0.21000 & 0.02000 & -0.21000 & 0.30000 \\ -0.23000 & 0.15000 & -0.01000 & 0.12000 & 0.31000 \end{pmatrix} * \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix} + 0 \right)$$

$$I_t = \sigma \left(\begin{pmatrix} 0.00110 \\ 0.00050 \end{pmatrix} + \begin{pmatrix} 0.15720 \\ 0.05130 \end{pmatrix} \right)$$

$$I_t = \sigma \left(\begin{pmatrix} 0.15830 \\ 0.05180 \end{pmatrix} \right)$$

$$\sigma(0.15830) = \frac{1}{1+e^{-0.15830}}$$

$$\sigma(0.15830) = 0.53950$$

$$\sigma(0.05180) = \frac{1}{1+e^{-0.05180}}$$

$$\sigma(0.05180) = 0.51290$$

$$I_t = \begin{pmatrix} 0.53950 \\ 0.51290 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t ketiga pada tanggal 5 Januari 2022. Terdapat bobot W_i yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_2 . Dan, terdapat bobot U_i sebagai

matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai input X_3 . Dengan mendapatkan nilai *Input gate* t ketiga yaitu $\begin{pmatrix} 0.53950 \\ 0.51290 \end{pmatrix}$

Perhitungan t ketiga, untuk mendapatkan nilai *Memory Gate* (Z_3) dengan menggunakan Persamaan 2.13 :

$$H_2 = \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix}$$

$$W_c = \begin{pmatrix} 0.12000 & -0.20000 \\ 0.21000 & 0.02300 \end{pmatrix}$$

$$U_c = \begin{pmatrix} 0.12000 & 0.08000 & 0.01200 & -0.01000 & -0.01000 \\ -0.20000 & 0.02000 & 0.12000 & -0.20000 & 0.08000 \end{pmatrix}$$

$$X_t = \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix}$$

$$b_c = 0$$

$$Z_t = \tanh(W_c * H_{t-1} + U_c * X_t + b_c)$$

$$z_t = \tanh\left(\begin{pmatrix} 0.12000 & -0.20000 \\ 0.21000 & 0.02300 \end{pmatrix} * \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix} + \right.$$

$$\left. \begin{pmatrix} 0.12000 & 0.08000 & 0.01200 & -0.01000 & -0.01000 \\ -0.20000 & 0.02000 & 0.12000 & -0.20000 & 0.08000 \end{pmatrix} * \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix} + 0\right)$$

$$Z_t = \sigma\left(\begin{pmatrix} 0.0011 \\ -0.0007 \end{pmatrix} + \begin{pmatrix} 0.09704 \\ -0.1142 \end{pmatrix}\right)$$

$$Z_t = \sigma\left(\begin{pmatrix} 0.09810 \\ -0.11490 \end{pmatrix}\right)$$

$$\tanh(0.09810) = \frac{e^{-0.09810} - e^{-(-0.09810)}}{e^{-0.09810} + e^{-(-0.09810)}}$$

$$\tanh(0.09810) = 0.09780$$

$$\tanh(-0.11490) = \frac{e^{-0.11490} - e^{0.11490}}{e^{-0.11490} + e^{0.11490}}$$

$$\tanh(-0.11490) = -0.11440$$

$$Z_t = \begin{pmatrix} 0.09780 \\ -0.11440 \end{pmatrix}$$

Pada perhitungan di atas adalah perhitungan t ketiga pada tanggal 5 Januari 2022 untuk mencari *Memory gate* (Z_t). Terdapat bobot W_c yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot U_c sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai *input* X_t . Dengan mendapatkan nilai *Memory gate* t ketiga yaitu $\begin{pmatrix} 0.09780 \\ -0.11440 \end{pmatrix}$

Perhitungan t ketiga, untuk mendapatkan nilai *Output Gate* (O_3) dengan menggunakan Persamaan 2.15:

$$\begin{aligned}
 H_2 &= \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix} \\
 W_o &= \begin{pmatrix} -0.32000 & 0.15000 \\ 0.21000 & 0.06000 \end{pmatrix} \\
 U_o &= \begin{pmatrix} 0.12000 & -0.32000 & 0.22000 & -0.01000 & 0.11000 \\ 0.20000 & 0.12000 & 0.12000 & 0.21000 & 0.17000 \end{pmatrix} \\
 X_t &= \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix} \\
 b_o &= 0 \\
 O_t &= \sigma(W_o * H_{t-1} + U_o * X_t + b_o) \\
 O_t &= \sigma \left(\begin{pmatrix} -0.32000 & 0.15000 \\ 0.21000 & 0.06000 \end{pmatrix} * \begin{pmatrix} 0.00260 \\ -0.00390 \end{pmatrix} + \right. \\
 &\quad \left. \begin{pmatrix} 0.12000 & -0.32000 & 0.22000 & -0.01000 & 0.11000 \\ 0.20000 & 0.12000 & 0.12000 & 0.21000 & 0.17000 \end{pmatrix} * \begin{pmatrix} 0.55000 \\ 0.41000 \\ 0.37000 \\ 0.38000 \\ 0.24000 \end{pmatrix} + 0 \right) \\
 O_t &= \sigma \left(\begin{pmatrix} -0.0014 \\ 0.0003 \end{pmatrix} + \begin{pmatrix} 0.03880 \\ 0.22520 \end{pmatrix} \right) \\
 O_t &= \sigma \left(\begin{pmatrix} 0.03740 \\ 0.22550 \end{pmatrix} \right) \\
 \sigma(0.03740) &= \frac{1}{1+e^{-0.03740}} \\
 \sigma(0.03740) &= 0.50930 \\
 \sigma(0.22550) &= \frac{1}{1+e^{-0.22550}} \\
 \sigma(0.22550) &= \\
 O_t &= \begin{pmatrix} 0.50930 \\ 0.55610 \end{pmatrix}
 \end{aligned}$$

Pada perhitungan di atas adalah perhitungan t ketiga untuk mencari *Output gate* (O_t). Terdapat bobot W_o yang di inisialisasikan sebagai matriks dengan ukuran 1x2 dimensi yang digunakan untuk perhitungan H_{t-1} . Dan, terdapat bobot

U_o sebagai matriks dengan ukuran 2x5 dimensi yang digunakan untuk perhitungan dengan nilai *input* X_t . Dengan mendapatkan nilai *Output gate* t ketiga yaitu

$$\begin{pmatrix} 0.50930 \\ 0.55610 \end{pmatrix}$$

Setelah mendapatkan nilai *Forget gate*, *Input gate*, *Memory gate*, *Output gate* untuk t ketiga, maka akan dilakukan perhitungan untuk mendapatkan nilai *Cell state*, dan nilai *Output* H_t pada t ketiga.

Perhitungan t ketiga, Cell State(C_t) dan Output (H_3) dengan menggunakan Persamaan 2.14 dan 2.16:

$$C_{t-1} = \begin{pmatrix} 0.00520 \\ -0.00740 \end{pmatrix}$$

$$F_t = \begin{pmatrix} 0.56370 \\ 0.53520 \end{pmatrix}$$

$$I_t = \begin{pmatrix} 0.53950 \\ 0.51290 \end{pmatrix}$$

$$Z_t = \begin{pmatrix} 0.09780 \\ -0.11440 \end{pmatrix}$$

$$O_t = \begin{pmatrix} 0.50930 \\ 0.55610 \end{pmatrix}$$

$$C_t = F_t \otimes C_{t-1} + I_t \otimes Z_t$$

$$C_t = \begin{pmatrix} 0.56370 \\ 0.53520 \end{pmatrix} \otimes \begin{pmatrix} 0.00520 \\ -0.00740 \end{pmatrix} + \begin{pmatrix} 0.53950 \\ 0.51290 \end{pmatrix} \otimes \begin{pmatrix} 0.09780 \\ -0.11440 \end{pmatrix}$$

$$C_t = \begin{pmatrix} 0.05570 \\ -0.06260 \end{pmatrix}$$

$$\tanh(C_t) = \begin{pmatrix} 0.05560 \\ -0.06250 \end{pmatrix}$$

$$H_t = O_t \otimes \tanh(C_t)$$

$$H_3 = \begin{pmatrix} 0.50930 \\ 0.55610 \end{pmatrix} \otimes \begin{pmatrix} 0.05560 \\ -0.06250 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} 0.02830 \\ -0.03480 \end{pmatrix}$$

Pada perhitungan di atas dengan menggunakan pemodelan dari LSTM pada t ketiga maka mendapatkan nilai *Output* $H_3 = \begin{pmatrix} 0.02830 \\ -0.03480 \end{pmatrix}$. Hasil dari H_3 dan C_3 bisa digunakan untuk melakukan perhitungan t ke-empat dan seterusnya, dengan

melakukan perhitungan yang sama dengan Persamaan yang sama.

3.4.4.2 Backward Layer LSTM

Berikut merupakan proses perhitungan dari proses *backward Layer* yaitu selisih *output* pada *timestep* 5 pada Persamaan 2.17. Nilai tersebut merupakan *error* yang dihasilkan pada tahap *forward Layer*, dan nilai ini yang akan menentukan perhitungan tahap-tahap selanjutnya pada proses *backward Layer*.

$$\Delta Y_5 = \Delta E + W_c \cdot \Delta z_6 + W_i \cdot \Delta i_6 + W_f \cdot \Delta f_6 + W_o \cdot \Delta o_6$$

$$\Delta Y_5 = \begin{pmatrix} 0.10300 \\ -0.11350 \end{pmatrix}$$

Berikut merupakan proses perhitungan nilai dari Persamaan 2.17. Setelah didapatkan hasil dari ΔY_5 dilakukan perhitungan *Output Gate* dan *Cell State* dengan menggunakan Persamaan 2.18 dan Persamaan 2.19.

$$\Delta O_t = \Delta Y_t \cdot \tanh(c_t) \cdot \sigma(o_t)$$

$$\Delta O_t = \begin{pmatrix} 0.1030 \\ -0.1135 \end{pmatrix} \cdot \tanh\left(\begin{pmatrix} 0.19690 \\ -0.17290 \end{pmatrix}\right) \cdot \sigma\left(\begin{pmatrix} 0.62790 \\ 0.65850 \end{pmatrix}\right)$$

$$\Delta O_t = \begin{pmatrix} 0.01270 \\ 0.01290 \end{pmatrix}$$

$$\Delta C_t = \Delta Y_t \cdot O_t \cdot (1 - \tanh^2(c_t)) + \Delta C_{t+1} \cdot f_{t+1}$$

$$\Delta C_5 = \Delta Y_5 \cdot O_5 \cdot (1 - \tanh^2(c_5)) + \Delta C_6 \cdot f_6$$

$$\Delta C_5 = \begin{pmatrix} 0.1030 \\ -0.1135 \end{pmatrix} \cdot \begin{pmatrix} 0.52320 \\ 0.65650 \end{pmatrix} \cdot (1 - \tanh^2\left(\begin{pmatrix} 0.19950 \\ -0.17470 \end{pmatrix}\right)) + 0$$

$$\Delta C_5 = \begin{pmatrix} 0.05180 \\ -0.07230 \end{pmatrix}$$

Setelah didapatkan nilai ΔO_5 dan ΔC_5 maka hasil dari nilai tersebut akan digunakan untuk melakukan perhitungan untuk mencari ΔF_5 , ΔI_5 , dan ΔZ_5 dengan menggunakan Persamaan 2.20, 2.21, 2.22.

$$\begin{aligned}\Delta F_t &= \Delta C_t \cdot C_{t-1} \cdot (1 - \sigma^2(f_t)) \\ \Delta F_5 &= \Delta C_5 \cdot C_4 \cdot (1 - \sigma^2(f_5)) \\ \Delta F_5 &= \begin{pmatrix} 0.05180 \\ -0.07230 \end{pmatrix} \cdot \begin{pmatrix} 0.10550 \\ -0.11150 \end{pmatrix} \cdot (1 - \sigma^2(\begin{pmatrix} 0.68090 \\ 0.58190 \end{pmatrix})) \\ \Delta F_5 &= \begin{pmatrix} 0.00360 \\ 0.00520 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\Delta I_t &= \Delta C_t \cdot z_t \cdot (1 - \sigma^2(i_t)) \\ \Delta I_4 &= \begin{pmatrix} 0.05180 \\ -0.07230 \end{pmatrix} \cdot \begin{pmatrix} 0.20800 \\ -0.18760 \end{pmatrix} \cdot (1 - \sigma^2(\begin{pmatrix} 0.61520 \\ 0.58660 \end{pmatrix})) \\ \Delta I_t &= \begin{pmatrix} 0.00700 \\ 0.00870 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\Delta Z_t &= \Delta C_t \cdot i_t \cdot (1 - \tanh^2(z_t)) \\ \Delta Z_t &= \begin{pmatrix} 0.05180 \\ -0.07230 \end{pmatrix} \cdot \begin{pmatrix} 0.61520 \\ 0.58660 \end{pmatrix} \cdot (1 - \tanh^2(\begin{pmatrix} 0.20800 \\ -0.18760 \end{pmatrix})) \\ \Delta Z_t &= \begin{pmatrix} 0.03050 \\ -0.04100 \end{pmatrix}\end{aligned}$$

Pada perhitungan di atas merupakan bagian dari perhitungan *timestep* ke 5, maka akan dilakukan perhitungan *timestep* selanjutnya sampai ke 1 dengan rumus dan cara yang sama. Setelah dilakukan proses perhitungan, maka gradien atau selisih nilai bias dapat dihitung dengan menggunakan Persamaan 2.23 untuk bobot aktivasi untuk *input*, Persamaan 2.24 untuk bobot aktivasi untuk setiap gate.

$$\Delta W_{xf} = \sum_{t=1}^5 (\Delta f_t x_t)$$

$$\Delta W_{xf} = (\Delta f_1 x_1) + (\Delta f_2 x_2) + (\Delta f_3 x_3) + (\Delta f_4 x_4) + (\Delta f_5 x_5)$$

$$\Delta W_{xf} = \begin{pmatrix} 0.00707 & 0.00548 & 0.00639 & 0.00569 & 0.00410 \\ 0.00920 & 0.00735 & 0.00840 & 0.00759 & 0.00574 \end{pmatrix}$$

$$\Delta W_{xi} = \sum_{t=1}^5 (\Delta i_t x_t)$$

$$\Delta W_{xi} = (\Delta i_1 x_1) + (\Delta i_2 x_2) + (\Delta i_3 x_3) + (\Delta i_4 x_4) + (\Delta i_5 x_5)$$

$$\Delta W_{xi} = \begin{pmatrix} 0.01309 & 0.01067 & 0.01169 & 0.01086 & 0.00839 \\ 0.01633 & 0.01326 & 0.01460 & 0.01352 & 0.01040 \end{pmatrix}$$

$$\Delta W_{xo} = \sum_{t=1}^5 (\Delta o_t x_t)$$

$$\Delta W_{xo} = (\Delta o_1 x_1) + (\Delta o_2 x_2) + (\Delta o_3 x_3) + (\Delta o_4 x_4) + (\Delta o_5 x_5)$$

$$\Delta W_{xo} = \begin{pmatrix} 0.01621 & 0.01472 & 0.01545 & 0.01488 & 0.01338 \\ 0.00468 & 0.00428 & 0.00451 & 0.00433 & 0.00393 \end{pmatrix}$$

$$\Delta W_{xc} = \sum_{t=1}^5 (\Delta c_t x_t)$$

$$\Delta W_{xc} = (\Delta c_1 x_1) + (\Delta c_2 x_2) + (\Delta c_3 x_3) + (\Delta c_4 x_4) + (\Delta c_5 x_5)$$

$$\Delta W_{xc} = \begin{pmatrix} 0.07241 & 0.05895 & 0.06420 & 0.06097 & 0.04391 \\ -0.06354 & -0.05275 & -0.05762 & -0.05412 & -0.04198 \end{pmatrix}$$

$$\Delta W_{hf} = \sum_{t=1}^5 (\Delta f_t h_t)$$

$$\Delta W_{hf} = (\Delta f_5 h_5) + (\Delta f_4 h_4) + (\Delta f_3 h_3) + (\Delta f_2 h_2) + (\Delta f_1 h_1)$$

$$\Delta W_{hf} = \begin{pmatrix} 0.00062 & 0.00062 \\ -0.00091 & -0.00091 \end{pmatrix}$$

$$\Delta W_{hi} = \sum_{t=1}^5 (\Delta i_t h_t)$$

$$\Delta W_{hi} = (\Delta i_5 h_5) + (\Delta i_4 h_4) + (\Delta i_3 h_3) + (\Delta i_2 h_2) + (\Delta i_1 h_1)$$

$$\Delta W_{hi} = \begin{pmatrix} 0.00112 & 0.00112 \\ -0.00126 & -0.00126 \end{pmatrix}$$

$$\Delta W_{ho} = \sum_{t=1}^5 (\Delta o_t h_t)$$

$$\Delta W_{ho} = (\Delta o_5 h_5) + (\Delta o_4 h_4) + (\Delta o_3 h_3) + (\Delta o_2 h_2) + (\Delta o_1 h_1)$$

$$\Delta W_{ho} = \begin{pmatrix} 0.00155 & 0.00155 \\ -0.00172 & -0.00172 \end{pmatrix}$$

$$\Delta W_{hc} = \sum_{t=1}^5 (\Delta c_t h_t)$$

$$\Delta W_{hc} = (\Delta c_5 h_5) + (\Delta c_4 h_4) + (\Delta c_3 h_3) + (\Delta c_2 h_2) + (\Delta c_1 h_1)$$

$$\Delta W_{hc} = \begin{pmatrix} 0.00586 & 0.00586 \\ -0.00665 & -0.00665 \end{pmatrix}$$

Setelah didapatkan nilai untuk setiap bobot pada masing-masing *gate* atau bobot *input* pada *gate*, maka akan dilakukan perhitungan untuk mencari nilai bobot baru yang dengan perhitungan bobot pada setiap *gate* yang dipakai dikurang dengan bobot aktivasi pada setiap *gate*.

$$U_F = U_f - \Delta W_{xf}$$

$$U_F = \begin{pmatrix} 0.02000 & -0.01000 & 0.30000 & 0.23000 & 0.21000 \\ 0.12 & 0.31 & -0.23 & 0.25 & -0.01 \end{pmatrix} - \begin{pmatrix} 0.00707 & 0.00548 & 0.00639 & 0.00569 & 0.00410 \\ 0.00920 & 0.00735 & 0.00840 & 0.00759 & 0.00574 \end{pmatrix}$$

$$U_f = \begin{pmatrix} 0.01293 & -0.01548 & 0.29361 & 0.22431 & 0.20590 \\ 0.00280 & 0.30265 & -0.23840 & 0.24241 & -0.01574 \end{pmatrix}$$

$$U_I = U_i - \Delta W_{xi}$$

$$U_I = \begin{pmatrix} 0.11691 & 0.19933 & 0.00831 & -0.22086 & 0.29161 \\ -0.24633 & 0.13674 & -0.02460 & 0.10648 & 0.29960 \end{pmatrix}$$

$$U_O = U_o - \Delta W_{xo}$$

$$U_O = \begin{pmatrix} 0.10379 & -0.33472 & 0.20455 & -0.02488 & 0.09662 \\ 0.01532 & 0.11572 & 0.11549 & 0.20567 & 0.16607 \end{pmatrix}$$

$$U_C = U_c - \Delta W_{xc}$$

$$U_C = \begin{pmatrix} 0.04759 & 0.02105 & -0.05220 & -0.07097 & -0.05391 \\ -0.13646 & 0.07275 & 0.17762 & -0.14588 & 0.12198 \end{pmatrix}$$

$$W_F = W_f - \Delta W_{hf}$$

$$W_F = \begin{pmatrix} 0.21000 & 0.06000 \\ 0.02300 & 0.04000 \end{pmatrix} - \begin{pmatrix} 0.00062 & 0.00062 \\ -0.00091 & -0.00091 \end{pmatrix}$$

$$W_F = \begin{pmatrix} 0.20938 & 0.05938 \\ 0.02391 & 0.04091 \end{pmatrix}$$

$$W_I = W_i - \Delta W_{hi}$$

$$W_I = \begin{pmatrix} 0.11888 & -0.20112 \\ 0.21126 & 0.02426 \end{pmatrix}$$

$$W_O = W_o - \Delta W_{ho}$$

$$W_O = \begin{pmatrix} -0.32155 & 0.14845 \\ 0.21172 & 0.06172 \end{pmatrix}$$

$$W_C = W_c - \Delta W_{hc}$$

$$W_C = \begin{pmatrix} 0.11414 & -0.20586 \\ 0.06665 & 0.21665 \end{pmatrix}$$

Dengan diperolehnya seluruh nilai bobot yang baru untuk proses perhitungan *epoch* selanjutnya, maka perhitungan LSTM dengan 1 epoch dan 2 unit telah selesai.

3.4.4.3 Perhitungan *Dense* pada *Output Layer*

Setelah melakukan perhitungan LSTM, hasil dari setiap nilai *output* (H_t) merupakan hasil dari 2 *unit* LSTM. Maka, perlu mengubah hasil (H_t) menjadi 1 *output* yang dinamakan perhitungan *Dense* pada *output layer*. Dengan menggunakan Persamaan 2.26 berikut adalah perhitungan *perceptron* pada *hidden layer*.

$$\begin{aligned} x &= \begin{pmatrix} 0.10300 \\ -0.11350 \end{pmatrix} \\ W_T &= \begin{pmatrix} 0.14000 \\ -0.20000 \end{pmatrix} \\ b &= 0 \\ y &= \sigma(W_T * x + b) \\ y &= \sigma\left(\begin{pmatrix} 0.14000 & -0.20000 \end{pmatrix} * \begin{pmatrix} 0.10300 \\ -0.11350 \end{pmatrix}\right) \\ y &= \sigma(0.03713) \\ y &= 0.50928 \end{aligned}$$

Pada perhitungan di atas terdapat variabel y yang merupakan nilai perhitungan *Dense* pada *Output Layer*. Pada contoh di atas merupakan nilai H_t kelima. Variabel w merupakan bobot dengan menggunakan nilai acak dengan skala -0.2 sampai 0.2. Setelah melakukan perhitungan maka didapatkan nilai prediksi pada hari ketiga dengan nilai 0.50928. Hasil tersebut harus dilakukan *Invers Minmax Scaler* untuk mendapatkan harga hasil prediksi yang dapat diketahui.

Tahap perhitungan *Inverse Scaler MinMax* pada hari ke-2.

$$x_{scaled} = 0.50928, \min(x) = 7325, \max(x) = 7650 \quad (3.5)$$

$$x_i = \min(x) + ((x_{scaled} * \max(x)) - (x_{scaled} * \min(x))) \quad (3.6)$$

$$x_i = 7325 + ((0.50928 * 7650) - (0.50928 * 7325)) \quad (3.7)$$

$$x_i = 7490.5 \quad (3.8)$$

Pada perhitungan di atas adalah perhitungan untuk mencari hasil prediksi pada hari tersebut dengan menggunakan *Inverse Scaler MinMax* dengan hasil variabel x_i . Terdapat variabel x_{scaled} yang merupakan hasil dari perhitungan *perceptron*.

3.4.5 Perhitungan Autoregressive Integrated Moving Average

coeffients untuk model *Autoregressive*:

<i>Coefficients</i>	
Intercept	-53.9451
ϕ_1	0.5894
ϕ_2	0.4256

Day	Close Price
<i>Day1</i>	7325
<i>Day2</i>	7400
<i>Day3</i>	7450
<i>Day4</i>	7475
<i>Day5</i>	7650

Date	Close Price
03/01/2022	P_1
04/01/2022	P_2
05/01/2022	P_3
06/01/2022	P_4
07/01/2022	P_5

Berikut adalah inisialisasi data untuk perhitungan ARIMA. Dengan menggunakan nilai parameter p,d,q ialah 2,1,0. Pertama-tama akan dilakukan *difference* dengan ordo 1 dengan menggunakan Persamaan 2.3. Berikut merupakan perhitungannya.

Perhitungan *difference*

$$Df_t = y_t - y_{t-1}$$

$$Df_1 = y_1 - 0$$

$$Df_1 = 0 - 0$$

$$Df_1 = 0$$

$$Df_2 = y_2 - y_1$$

$$Df_2 = 7400 - 7325$$

$$Df_2 = 75$$

$$Df_3 = y_3 - y_2$$

$$Df_3 = 7450 - 7400$$

$$Df_3 = 50$$

$$Df_4 = y_4 - y_3$$

$$Df_4 = 7475 - 7450$$

$$Df_4 = 25$$

$$Df_5 = y_5 - y_4$$

$$Df_5 = 7650 - 7475$$

$$Df_5 = 175$$

Pada perhitungan di atas, adalah hasil dari *Difference* dengan nilai ordo d ialah 1. Karena yang ingin diprediksi saat hari pertama sampai terakhir, maka dilakukan *difference* Df_1 sampai Df_5 dan mendapatkan nilai 0, 75, 50, 25, 175 yang nantinya nilai *difference* tersebut akan digunakan untuk memprediksi harga saham dengan menggunakan perhitungan *Autoregressive* dan *Moving Average* dengan menggunakan Persamaan 2.1 yang berada di bawah.

Perhitungan Autoregressive dan Moving AverageOrdo $p = 2$

<i>Coefficients</i>	
Intercept	-53.9451
ϕ_1	0.5894
ϕ_2	0.4256

Nilai yang sudah di lakukan *Difference*

Y_t	Close
Df_1	0
Df_2	75
Df_3	50
Df_4	25
Df_5	175

Persamaan: $Y_t = c + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \phi_1 \varepsilon_{t-1} + \dots + \phi_q \varepsilon_{t-q}$

$$Y_1 = 7325$$

$$Y_2 = -53.9451 + 0.5894 + 7325 + (0.4256 * 0)$$

$$Y_2 = 7272$$

$$Y_3 = -53.9451 + 0.5894 + 7400 + (0.4256 * 75)$$

$$Y_3 = 7379$$

$$Y_4 = -53.9451 + 0.5894 + 7450 + (0.4256 * 50)$$

$$Y_4 = 7418$$

$$Y_5 = -53.9451 + 0.5894 + 7475 + (0.4256 * 25)$$

$$Y_5 = 7432$$

Pada perhitungan di atas, adalah hasil prediksi yang menggunakan model ARIMA(2,1,0) yang didapatkan setelah melakukan perhitungan manual dengan rumus 2.2. Maka, hasil prediksi P_1 sampai P_5 berturut-turut ialah 7325, 7272, 7379, 7418, 7432.

3.4.6 Perbandingan Evaluasi *Forecasting*

Pada di bagian ini akan dijelaskan perbandingan Evaluasi *Forecasting* dengan RMSE antara hasil prediksi dari Bi-LSTM dan ARIMA yang sudah hitung sebelumnya.

Evaluasi *Forecasting* dengan RMSE, MAE, MAPE

P_t	Close (Aktual)	Close (Bi-LSTM)	Close (ARIMA)
P_1	7325	7487.5	7325
P_2	7400	7487.6	7272
P_3	7450	7488.4	7379
P_4	7475	7489.1	7418
P_5	7650	7490.5	7432

$$RMSE = \sqrt{\frac{1}{N} * \sum_{i=1}^N ((y_p - y_i)^2)}$$

$$MAE = \frac{1}{N} * \sum_{i=1}^N |y_p - y_i|$$

$$MAPE = 100\% * \frac{1}{N} * \sum_{i=1}^N \frac{|y_p - y_i|}{y_i}$$

perhitungan RMSE untuk Bi-LSTM:

$$(y_{i1} - y_{p1})^2 = (7325 - 7487.5)^2 = 26406.25$$

$$(y_{i2} - y_{p2})^2 = (7400 - 7487.6)^2 = 7672.46$$

$$(y_{i3} - y_{p3})^2 = (7450 - 7488.4)^2 = 1473.59$$

$$(y_{i4} - y_{p4})^2 = (7475 - 7489.1)^2 = 199.99$$

$$(y_{i5} - y_{p5})^2 = (7650 - 7490.5)^2 = 25434.97$$

$$= \sqrt{(26406.25 + 7672.46 + 1473.59 + 199.99 + 25434.97)/5}$$

$$RMSE_{LSTM} = 110.62$$

perhitungan MAE untuk Bi-LSTM:

$$\begin{aligned}
 |y_{i1} - y_{p1}| &= |7325 - 7487.5| = 162.50 \\
 |y_{i2} - y_{p2}| &= |7400 - 7487.6| = 87.59 \\
 |y_{i3} - y_{p3}| &= |7450 - 7488.4| = 38.39 \\
 |y_{i4} - y_{p4}| &= |7475 - 7489.1| = 14.14 \\
 |y_{i5} - y_{p5}| &= |7650 - 7490.5| = 159.48 \\
 &= (162.50 + 87.59 + 38.39 + 14.14 + 159.48)/5 \\
 MAE_{Bi-LSTM} &= 92.42
 \end{aligned}$$

perhitungan MAPE untuk Bi-LSTM:

$$\begin{aligned}
 |y_{i1} - y_{p1}|/y_{i1} &= |(7325 - 7487.5)/7325| = 0.02 \\
 |y_{i2} - y_{p2}|/y_{i2} &= |(7400 - 7487.6)/7400| = 0.01 \\
 |y_{i3} - y_{p3}|/y_{i3} &= |(7450 - 7488.4)/7450| = 0.01 \\
 |y_{i4} - y_{p4}|/y_{i4} &= |(7475 - 7489.1)/7475| = 0.00 \\
 |y_{i5} - y_{p5}|/y_{i5} &= |(7650 - 7490.3)/7650| = 0.02 \\
 &= (0.02000 + 0.0100 + 0.0100 + 0.0000 + 0.0200)/5 \\
 MAPE_{Bi-LSTM} &= 1.24\%
 \end{aligned}$$

perhitungan RMSE untuk ARIMA:

$$\begin{aligned}
 (y_{i1} - y_{p1})^2 &= (7325 - 7325)^2 = 0 \\
 (y_{i2} - y_{p2})^2 &= (7400 - 7272)^2 = 16475.19 \\
 (y_{i3} - y_{p3})^2 &= (7450 - 7379)^2 = 5103.06 \\
 (y_{i4} - y_{p4})^2 &= (7475 - 7418)^2 = 3257.63 \\
 (y_{i5} - y_{p5})^2 &= (7650 - 7432)^2 = 47400.12 \\
 &= \sqrt{(0 + 16475.19 + 5103.06 + 3257.63 + 47400.12)/5} \\
 RMSE_{ARIMA} &= 120.20
 \end{aligned}$$

perhitungan MAE untuk ARIMA:

$$|y_{i1} - y_{p1}| = |7325 - 7325| = 0$$

$$|y_{i2} - y_{p2}| = |7400 - 7272| = 128.00$$

$$|y_{i3} - y_{p3}| = |7450 - 7379| = 71.00$$

$$|y_{i4} - y_{p4}| = |7475 - 7418| = 57.00$$

$$|y_{i5} - y_{p5}| = |7650 - 7432| = 218.00$$

$$= (0 + 128 + 71 + 57 + 218)/5$$

$$MAE_{ARIMA} = 94.92$$

perhitungan MAPE untuk ARIMA:

$$|y_{i1} - y_{p1}| = |(7325 - 7325)/7325| = 0$$

$$|y_{i2} - y_{p2}| = |(7400 - 7272)/7400| = 0.02$$

$$|y_{i3} - y_{p3}| = |(7450 - 7379)/7450| = 0.01$$

$$|y_{i4} - y_{p4}| = |(7475 - 7418)/7475| = 0.01$$

$$|y_{i5} - y_{p5}| = |(7650 - 7432)/7650| = 0.03$$

$$= (0 + 0.02 + 0.01 + 0.01 + 0.03)/5$$

$$MAPE_{ARIMA} = 1.26\%$$

Pada perhitungan Evaluasi *Forecasting* di atas dengan menggunakan rumus 2.27, maka dapat diketahui bahwa model Bi-LSTM mempunyai hasil model prediksi lebih baik dengan mendapatkan nilai *error* RMSE, MAE, MAPE berturut-turut: 110.69, 92.44, 1.24% dibandingkan dengan model ARIMA yang mendapatkan *error* 120.20, 94.92, 1.26% dalam memprediksi harga saham BBKA.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan menjelaskan mengenai proses implementasi dan pengujian terhadap sistem yang telah dibangun berdasarkan penjelasan pada bab sebelumnya.

4.1 Lingkungan Implementasi

Pada lingkungan implementasi, akan dijelaskan mengenai perangkat - perangkat yang digunakan selama proses pembangunan sistem baik dari perangkat keras maupun perangkat lunak yang digunakan.

4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan prediksi harga saham adalah sebagai berikut:

1. *Laptop* Acer Nitro 5 AN515-57.
2. *Processor*: intel core i5-11400H 2.70GHz.
3. *Solid State Drive* kapasitas 512GB.
4. RAM dengan kapasistas 16GB

4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi dari perangkat lunak yang digunakan dalam pembangunan prediksi harga saham adalah sebagai berikut:

1. Sistem Operasi Windows 11 Home
2. Python 3.9.7 (Jupyter Notebook)
3. Google Colab

4.2 Implementasi Class

Pada bab ini akan dijelaskan mengenai *class* dan *method* yang akan digunakan dalam pembuatan sistem prediksi harga saham.

4.2.1 *Class EvaluasiForecasting*

Class EvaluasiForecasting digunakan untuk memuat *evaluasi* atau pengukuran nilai *error* dengan menggunakan dari RMSE, MAE, MAPE. Tabel menunjukkan daftar *method* yang digunakan dalam *class EvaluasiForecasting*.

Tabel 4.1 Daftar *method* pada *class EvaluasiForecasting*

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	rmse_metric	<i>actual</i> <i>predicted</i>	float[] float[]	error	Fungsi untuk melakukan perhitungan mendapatkan nilai <i>error</i> RMSE.
2	mae_metric	<i>actual</i> <i>predicted</i>	float[] float[]	error	Fungsi untuk melakukan perhitungan mendapatkan nilai <i>error</i> mae.
3	mape_metric	<i>actual</i> <i>predicted</i>	float[] float[]	error	Fungsi untuk melakukan perhitungan mendapatkan nilai <i>error</i> mape.

Pada Gambar 4.1 merupakan penggalan kode untuk implementasi evaluasi *forecasting* atau pengukuran nilai *error* pada model yang dihasilkan dengan menggunakan RMSE, MAE, MAPE.

```
class EvaluasiForecasting:
    def rmse_metric(actual, predicted):
        mean_error = np.square(np.subtract(actual,predicted)).mean()
        return math.sqrt(mean_error)

    def mae_metric(actual, predicted):
        y_true, predicted = np.array(actual), np.array(predicted)
        return np.mean(np.abs(actual - predicted))

    def mape_metric(actual, predicted):
        actual, predicted = np.array(actual), np.array(predicted)
        return np.mean(np.abs((actual - predicted) / actual)) * 100
```

Gambar 4.1 Fungsi-Fungsi untuk melakukan Evaluasi pada model yang dihasilkan

4.2.2 Class *Preprocessing*

Class Preprocessing digunakan untuk melakukan pengolahan data harga saham. Tabel menunjukkan daftar *method* yang digunakan dalam *class*

preprocessing.

Tabel 4.2 Daftar *method* pada *class preprocessing*

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	feature_selection	data	[]	data	Menyeleksi data yang akan digunakan
2	reshape_data	data	[]	data	mengubah data ke bentuk nilai dalam <i>numpy</i>
3	min_max	data	[]	data	Mengubah data <i>input</i> menjadi scaler dengan fungsi <i>MinMax</i> untuk <i>observasi</i> Bi-LSTM
4	inverse_minmax	data	[]	data	Mengubah data yang sudah di skala dengan min_max menjadi data yang dapat angka harga saham
5	partition_dataset	data	[]	x,y	Mengubah data dalam bentuk x_train untuk melakukan <i>observasi</i> dan y_train merupakan nilai yang akan diprediksi.
6	splitting_dataset	data	[]	data	Membagi data 80% untuk dilakukan pelatihan dan 20% untuk uji

Pada Gambar 4.2 & 4.3 merupakan penggalan kode untuk implementasi *preprocessing* sebelum pembuatan *model* dilakukan. Pada pembuatan *model* Bi-LSTM, data menggunakan 5 variabel untuk melakukan penelitian ini. Diantara lain: *High*, *Low*, *Open*, *Close*, *Volume*. Lalu, dilakukan juga *MinMaxScaler* untuk mengubah nilai saat penelitian supaya Bi-LSTM bisa mengkalkulasi ini.

```
class Preprocessing:
    def feature_selection(df):
        # Indexing Batches
        train_df = df.sort_values(by=['Date']).copy()

        # Daftar Fitur yang digunakan
        FEATURES = ['High', 'Low', 'Open', 'Close', 'Volume']
        data = pd.DataFrame(train_df)
        data_filtered = data[FEATURES]

        # menambahkan kolom prediksi dan menetapkan nilai dummy untuk menyiapkan data untuk penskalaan
        data_filtered_ext = data_filtered.copy()
        data_filtered_ext['Prediction'] = data_filtered_ext['Close']
        return data_filtered_ext, data_filtered, data

    def reshape_data(data_filtered):
        # Dapatkan jumlah baris dalam data
        n_rows = data_filtered.shape[0]

        # Convert the data ke numpy values
        np_data_unscaled = np.array(data_filtered)
        np_data = np.reshape(np_data_unscaled, (n_rows, -1))

        return np_data_unscaled, np_data

    def min_max(np_data_unscaled, data_filtered_ext):
        scaler = MinMaxScaler(feature_range=(0,1))
        np_data_scaled = scaler.fit_transform(np_data_unscaled)

        # Membuat scaler terpisah yang berfungsi pada satu kolom untuk prediksi penskalaan
        scaler_pred = MinMaxScaler(feature_range=(0,1))
        df_Close = pd.DataFrame(data_filtered_ext['Close'])
        np_Close_scaled = scaler_pred.fit_transform(df_Close)

        return np_data_scaled, np_Close_scaled, scaler_pred

    def inverse_minmax(y_pred_scaled, y_test):
        y_pred = scaler_pred.inverse_transform(y_pred_scaled)
        y_test_unscaled = scaler_pred.inverse_transform(y_test.reshape(-1, 1))

        return y_pred, y_test_unscaled
```

Gambar 4.2 Fungsi-Fungsi untuk melakukan *Preprocessing* 1

```
def partition_dataset(sequence_length, data, index_Close):
    x, y = [], []
    data_len = data.shape[0]
    for i in range(sequence_length, data_len):
        x.append(data[i-sequence_length:i,:]) #contains sequence_length values 0-sequence_length * columns
        y.append(data[i, index_Close]) #contains the prediction values for validation, for single-step prediction

    # Convert the x and y to numpy arrays
    x = np.array(x)
    y = np.array(y)
    return x, y

def splitting_dataset(np_data_scaled, data):
    # Set the sequence length - this is the timeframe used to make a single prediction
    sequence_length = 1

    # Prediction Index
    index_Close = data.columns.get_loc("Close")

    # Split the training data into train and train data sets
    # As a first step, we get the number of rows to train the model on 80% of the data
    train_data_len = math.ceil(np_data_scaled.shape[0] * 0.8)

    # Create the training and test data
    train_data = np_data_scaled[0:train_data_len, :]
    test_data = np_data_scaled[train_data_len - sequence_length:, :]

    # Generate training data and test data
    x_train, y_train = Preprocessing.partition_dataset(sequence_length, train_data, index_Close)
    x_test, y_test = Preprocessing.partition_dataset(sequence_length, test_data, index_Close)

    return x_train, y_train, x_test, y_test, train_data_len
```

Gambar 4.3 Fungsi-Fungsi untuk melakukan *Preprocessing* 2

4.2.3 Class *ARIMA_model*

Class ARIMA_model merupakan kelas yang digunakan untuk membuat model prediksi dengan menggunakan metode ARIMA dengan nilai orde parameter p, d, q yang sudah ditentukan.

Tabel 4.3 Daftar *method* pada *class ARIMA_model*

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		
1	forecast_model	train_data test_data p q	float[] float[] int int	nilai pred	Fungsi untuk melakukan inisialisasi model ARIMA.

Pada Gambar 4.4 merupakan penggalan kode untuk implementasi pembuatan Model ARIMA saat melakukan *training* dan dilanjutkan untuk memprediksi dengan menggunakan data. *testing*.

```

class ARIMA_model:
    def forecast_model(train_data, test_data, p,q):
        train_arima = train_data['Close']
        test_arima = test_data['Close']
        history = [x for x in train_arima]
        y = test_arima
        # make first prediction
        predictions = list()
        model = sm.tsa.arima.ARIMA(history, order=(p,1,q))
        model_fit = model.fit()
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        history.append(y[0])
        # rolling forecasts
        for i in range(1, len(y)):
            # predict
            model = sm.tsa.arima.ARIMA(history, order=(p,d,q))
            model_fit = model.fit()
            yhat = model_fit.forecast()[0]
            # invert transformed prediction
            predictions.append(yhat)
            # observation
            obs = y[i]
            history.append(obs)
        return predictions, y

```

Gambar 4.4 Fungsi-Fungsi untuk melakukan pembuatan Model ARIMA

4.2.4 Class LSTM_unit

Class LSTM_unit merupakan kelas yang digunakan untuk membuat model *deep learning* Bi-LSTM untuk melakukan prediksi dengan *hyperparameter epoch* dan *units* yang telah ditentukan.

Tabel 4.4 Daftar *method* pada *class LSTM_unit*

No	Metode	Masukan		Luaran	Keterangan
		Parameter	Variabel		

1	training_model	x_train y_train x_test y_test unit epoch	float[] float[] float[] float int int	x_test model	Fungsi untuk melakukan inisialisasi pelatihan pada model Bi-LSTM.
2	predict_model	x_test model	float[] object	nilai pred	Fungsi untuk melakukan uji pada model Bi-LSTM yang sudah dilatih.

Pada Gambar 4.5 merupakan penggalan kode untuk implementasi pembuatan Model Bi-LSTM saat melakukan *training* dan dilanjutkan untuk memprediksi dengan menggunakan data *testing*.

```
class LSTM_unit:
    def training_model(x_train, y_train, x_test, y_test, unit, epoch):
        # Configure the neural network model
        model = Sequential()
        model.add(Bidirectional(LSTM(unit, return_sequences=False, input_shape=(x_train.shape[1], x_train.shape[2]))))
        model.add(Dense(1))

        # Compile the model
        model.compile(loss='mse')
        # Training the model

        history = model.fit(x_train, y_train,
                            batch_size=16,
                            epochs=epoch,
                            validation_data=(x_test, y_test))
        return x_test, model, history

    def predict_model(x_test, model):
        y_pred_scaled = model.predict(x_test)
        return y_pred_scaled
```

Gambar 4.5 Fungsi-Fungsi untuk melakukan pembuatan Model LSTM

Pada salah satu parameter *library* LSTM pada Gambar 4.6 terdapat parameter `go_backward`. Parameter `go_backward` bertipe *boolean* (*True* atau *False*). Jika, parameter `go_backward` tersebut diaktifkan (*True*), maka lstm akan melakukan perhitungan dari belakang ke depan atau hanya melakukan pembuatan *backward layer*.

```
tf.keras.layers.LSTM(  
    units,  
    activation="tanh",  
    recurrent_activation="sigmoid",  
    use_bias=True,  
    kernel_initializer="glorot_uniform",  
    recurrent_initializer="orthogonal",  
    bias_initializer="zeros",  
    unit_forget_bias=True,  
    kernel_regularizer=None,  
    recurrent_regularizer=None,  
    bias_regularizer=None,  
    activity_regularizer=None,  
    kernel_constraint=None,  
    recurrent_constraint=None,  
    bias_constraint=None,  
    dropout=0.0,  
    recurrent_dropout=0.0,  
    return_sequences=False,  
    return_state=False,  
    go_backwards=False,  
    stateful=False,  
    time_major=False,  
    unroll=False,  
    **kwargs  
)
```

Gambar 4.6 Library Long Short Term Memory [29]

Berbeda dengan Fungsi *Bidirectional* yang terdapat pada Gambar 4.5 sudah melakukan perhitungan *Forward* dan *Backward Layer* secara langsung.

4.3 Implementasi Perangkat Lunak

Pada bab ini akan dijelaskan mengenai cara implementasi sistem prediksi harga saham dengan menggunakan *Long Short Term Memory* dan *Autoregressive Integrated Moving Average*.

4.3.1 Implementasi *Preprocessing*

Seperti yang sudah dijelaskan sebelumnya pada Bab 3, sebelum dilakukan pembuatan model dengan *Long Short Term Memory*, dilakukan *preprocessing* terlebih dahulu. Pada penelitian ini, *preprocessing* data yang dilakukan meliputi normalisasi data dengan menggunakan *method MinMaxScaler* dengan mengubah nilai asli menjadi nilai yang berentang 0 sampai 1. Normalisasi data diperlukan karena, metode *Long Short Term Memory sensitive* terhadap data yang berukuran besar.

4.4 Pengujian

Pada penelitian ini, pengujian yang dilakukan adalah membandingkan algoritme *Autoregressive Integrated Moving Average* (ARIMA) dengan *Long Short Term Memory* (LSTM) untuk melakukan prediksi harga saham.

Pada pengujian dalam penelitian ini akan melibatkan beberapa kombinasi *hyperparameter* dari setiap masing-masing metode untuk mendapatkan kombinasi yang tepat. *Hyperparameter* yang digunakan dalam metode ARIMA antara lain: p, d, q dan dataset untuk setiap jangka waktu prediksi sesuai emiten. Dan, *hyperparameter* yang digunakan dalam metode LSTM antara lain: *epochs*, *units* dan dataset untuk setiap jangka waktu prediksi sesuai emiten saham.

4.4.1 Skenario Pengujian *Autoregressive Integrated Moving Average*

Pengujian pertama akan dilakukan menggunakan algoritme *Autoregressive Integrated Moving Average* dengan menggunakan 4 jenis *hyperparameter* dengan jumlah kombinasi sebesar 36 kombinasi. Kombinasi *hyperparameter* digambarkan pada Table 4.5 di bawah ini.

ARIMA			
Parameter	p	q	Dataset
	1	1	GGRM-long
	2	2	GGRM-mid
			GGRM-short
			UNVR-long
			UNVR-mid
			UNVR-short
			PSDN-long
			PSDN-mid
			PSDN-short
Total	2	2	9
Total Pengujian	2 x 2 x 9		
	36		

Tabel 4.5 Skenario Pengujian kombinasi ARIMA

4.4.2 Skenario Pengujian *Long Short Term Memory*

Pengujian selanjutnya akan dilakukan menggunakan algoritme *Long Short Term Memory* dengan menggunakan 3 jenis *hyperparameter* dengan jumlah

kombinasi sebesar 81 kombinasi. Kombinasi *hyperparameter* digambarkan pada Tabel 4.6 di bawah ini.

LSTM			
Parameter	epochs	units	Dataset
	10	10	GGRM-long
	100	50	GGRM-mid
	1000	128	GGRM-short
			UNVR-long
			UNVR-mid
			UNVR-short
			PSDN-long
			PSDN-mid
			PSDN-short
Total	3	3	9
Total Pengujian	3 x 3 x 9		
	81		

Tabel 4.6 Skenario Pengujian kombinasi LSTM

Maka, total pengujian yang dilakukan saat penelitian adalah **117**.

4.4.3 Pembahasan Pengujian *Long Term Prediction*

Tabel 4.7 Hasil pengujian *Autoregressive Integrated Moving Average* untuk *Long Term Prediction*

No.	Dataset	p	d	q	RMSE	MAE	MAPE
1	GGRM-long	2	1	1	663.47	426.03	1.18
2	UNVR-long	2	1	1	110.93	82.74	1.57
3	PSDN-long	1	1	2	7.54	5.12	3.1

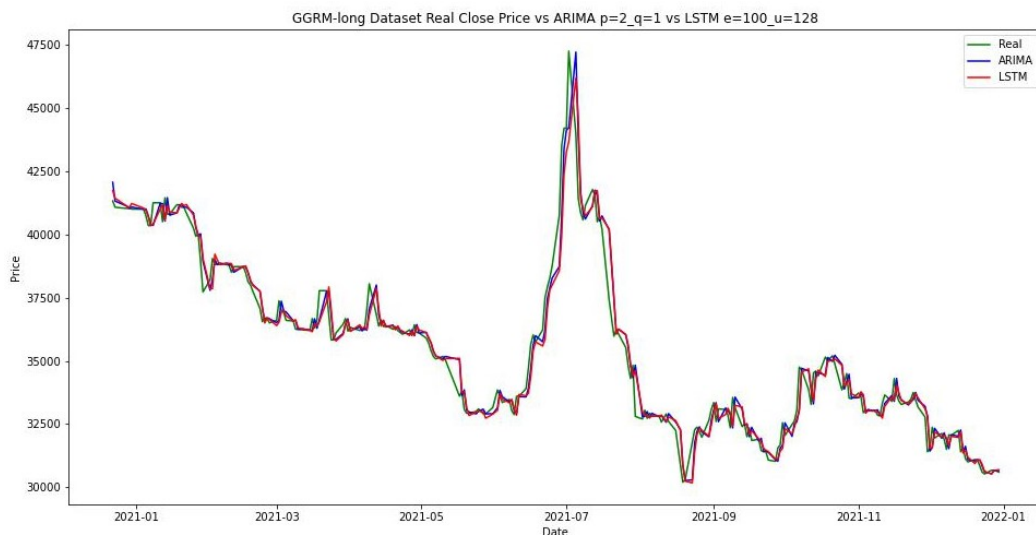
Pada subbab ini akan dibahas pengujian *Long Term Prediction* menggunakan metode *Autoregressive Integrated Moving Average* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu panjang. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 663.473, 426.0, 1.18, dengan

menggunakan nilai ordo parameter p,d,q ialah 2,1,1. Lalu, perusahaan UNVR memiliki RMSE, MAE, MAPE yang paling rendah adalah 110.93, 82.74, 1.57, dengan menggunakan nilai ordo parameter ialah 2,1,1. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 7.54, 5.12, 3.1 dengan menggunakan nilai ordo parameter 1,1,2.

Tabel 4.8 Hasil pengujian *Bidirectional Long Short Term Memory* untuk *Long Term Prediction*

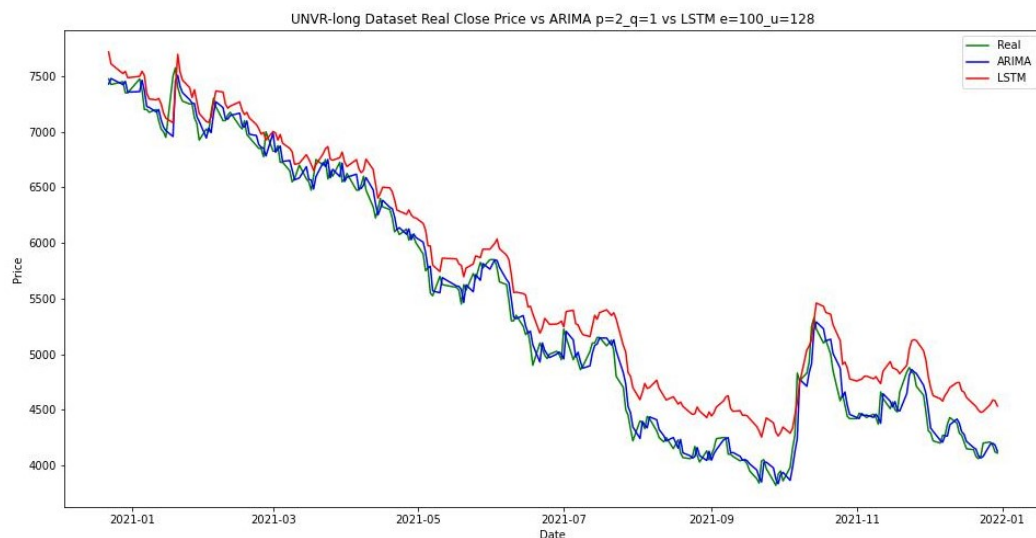
No.	<i>Dataset</i>	<i>Epochs</i>	<i>Units</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>
1	GGRM-long	100	128	733.23	506.21	1.41
2	UNVR-long	100	128	215.92	187.06	3.89
3	PSDN-long	1000	50	7.89	5.22	3.16

Lalu, pada bagian ini akan dibahas pengujian *Long Term Prediction* menggunakan metode *Bidirectional Long Short Term Memory* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu panjang. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 733.23, 506.21, 1.41, dengan menggunakan nilai *hyperparameter epochs* dan *units* ialah 100 dan 128. Lalu, perusahaan UNVR memiliki RMSE, MAE, MAPE yang paling rendah adalah 215.9, 187.06, 3.89, dengan menggunakan nilai *epochs* dan *units* ialah 100 dan 128. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 7.89, 5.22, 3.16 dengan menggunakan nilai *epochs* dan *units* ialah 1000 dan 50. Terlihat dari pengujian terbaik dari masing-masing metode, bahwa ARIMA lebih memiliki nilai *error* yang lebih kecil daripada Bi-LSTM dari setiap perusahaan pada pengujian *long term*.



Gambar 4.7 Perbandingan Chart Data GGRM untuk *Long Term*

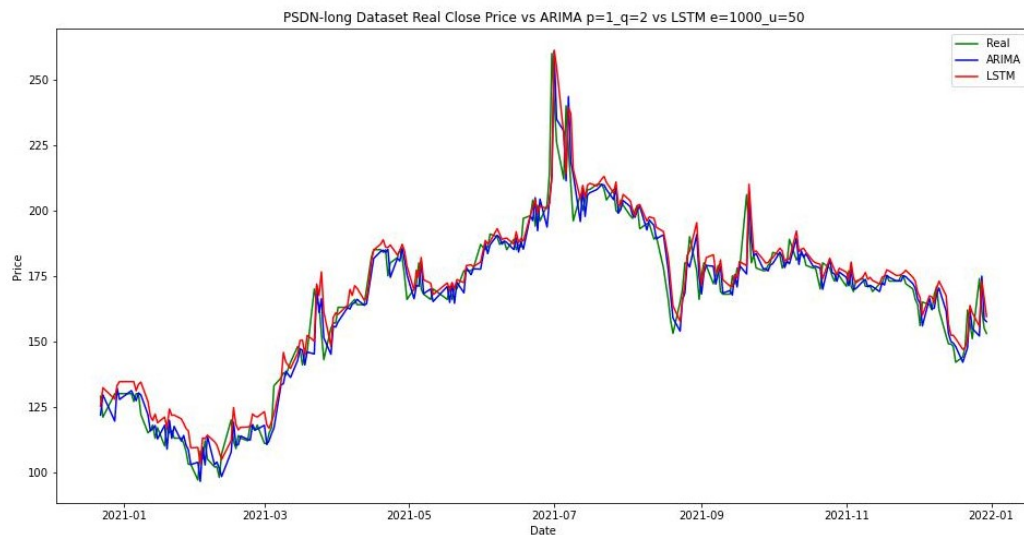
Pada Gambar 4.7, terlihat hasil prediksi *long term* untuk saham GGRM dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 2,1,1 mendapatkan hasil yang mendekati harga asli dengan mendapatkan hasil RMSE, MAE, MAPE, berturut-turut: 663.47, 426.03, 1.18. Sementara, metode BiLSTM dengan menggunakan *hyperparameter epochs* dan *units* ialah 100 dan 128 mendapatkan hasil RMSE, MAE, MAPE: 733.23, 506.21, 1.41. Dan terlihat dari grafik setiap metode memprediksi harga saham cukup mendekati dengan nilai harga asli, dan *trend* pada data yang terlihat naik dan turun di pertengahan *forecasting*.



Gambar 4.8 Perbandingan Chart Data UNVR untuk *Long Term*

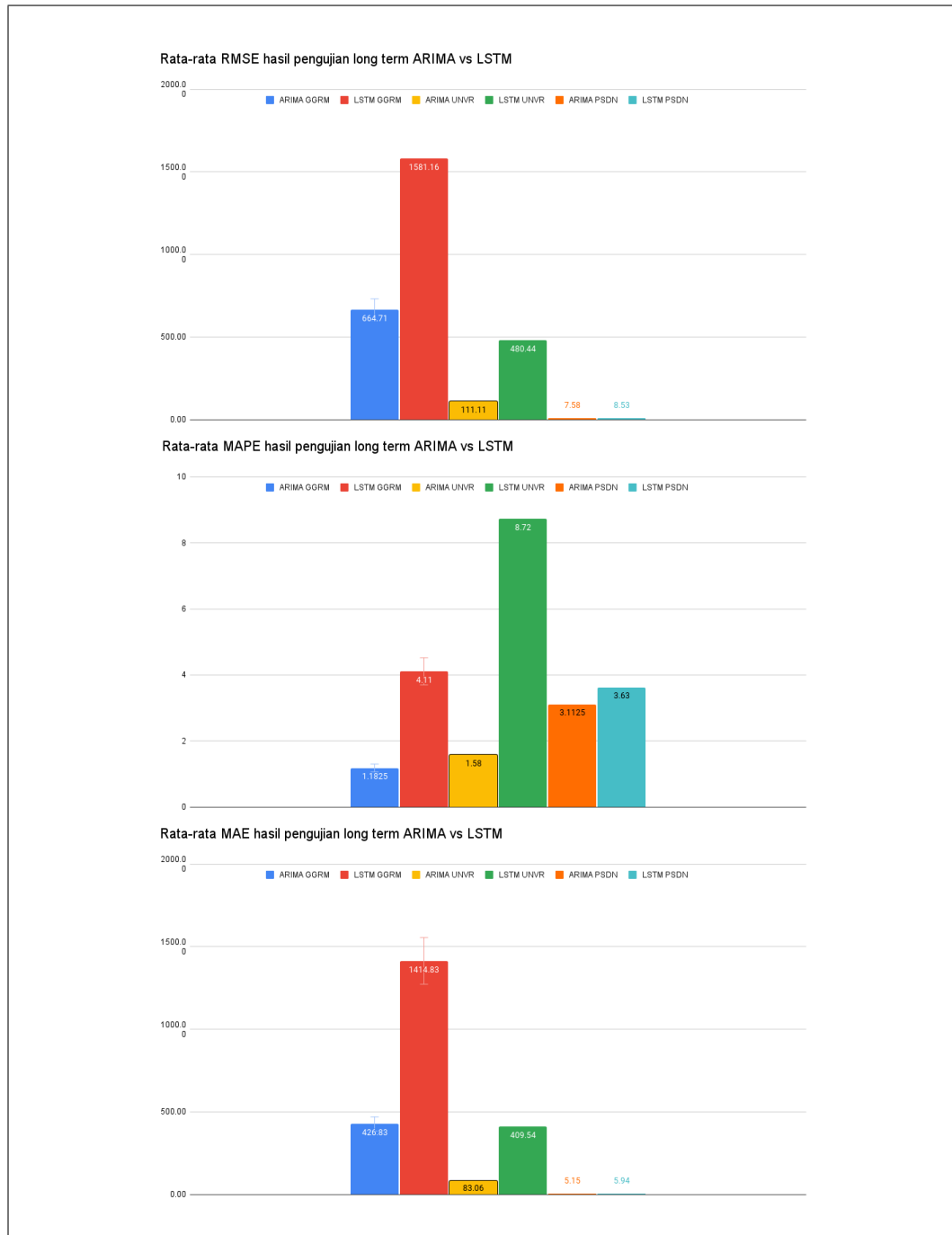
Pada Gambar 4.8, terlihat hasil prediksi *long term* untuk saham UNVR dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 2,1,1 mendapatkan hasil yang mendekati harga asli, karena nilai *error*-nya

berturut-turut 110.93, 82.74, 1.57. Sementara, metode Bi-LSTM dengan menggunakan *epoch* dan *unit* ialah 100 dan 128 mendapatkan hasil *error*-nya ialah 215.92, 187.06, 3.89, *trend* pada data yang terlihat menurun, secara bertahap. Dapat diketahui untuk prediksi *long term* dengan menggunakan data UNVR, ARIMA dapat memprediksi harga dengan lebih baik daripada Bi-LSTM, karena nilai *error* RMSE, MAE, MAPE yang didapatkan lebih kecil.



Gambar 4.9 Perbandingan Data PSDN untuk *Long Term*

Pada Gambar 4.9, terlihat hasil prediksi *long term* untuk saham PSDN dari setiap metode terbaik. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,2 mendapatkan hasil yang sangat mendekati harga asli, karena nilai *error*-nya 7.54, 5.12, 3.1. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epochs* dan *units* ialah 1000 dan 50 mendapatkan hasil RMSE 7.89, 5.22, 3.16, dan terlihat dari grafik kedua metode sangat mendekati nilai harga asli, dan *trend* pada data yang terlihat naik dan turun di pertengahan *forecasting*, dan terlihat lebih bergejolak. Dapat diketahui untuk prediksi *long term* dengan menggunakan data PSDN, ARIMA dapat memprediksi harga dengan lebih baik daripada Bi-LSTM, karena nilai *error* RMSE, MAE, MAPE yang didapatkan lebih kecil. Walaupun selisih RMSEnya hanya sekitar 0.35 dan MAPEnya 0.1.



Gambar 4.10 Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian *long term prediction* ARIMA dan Bi-LSTM

Pada Gambar 4.10, menunjukkan perbandingan nilai rata-rata keseluruhan dari pengukuran nilai *error* RMSE, MAE, MAPE untuk dataset GGRM, UNVR, PSDN prediksi *long term*. Dari *plots* tersebut menunjukkan bahwa, ARIMA memprediksi harga saham lebih baik daripada Bi-LSTM. Karena, nilai *error* yang dihasilkan lebih kecil daripada Bi-LSTM.

4.4.4 Pembahasan Pengujian *Mid Term Prediction*

Tabel 4.9 Hasil pengujian *Autoregressive Integrated Moving Average* untuk *Mid Term Prediction*

No.	<i>Dataset</i>	p	d	q	RMSE	MAE	MAPE
1	GGRM-mid	2	1	1	1604.70	1212.40	1.6
2	UNVR-mid	1	1	2	107.10	76.80	0.75
3	PSDN-mid	1	1	1	29.77	18.77	6.29

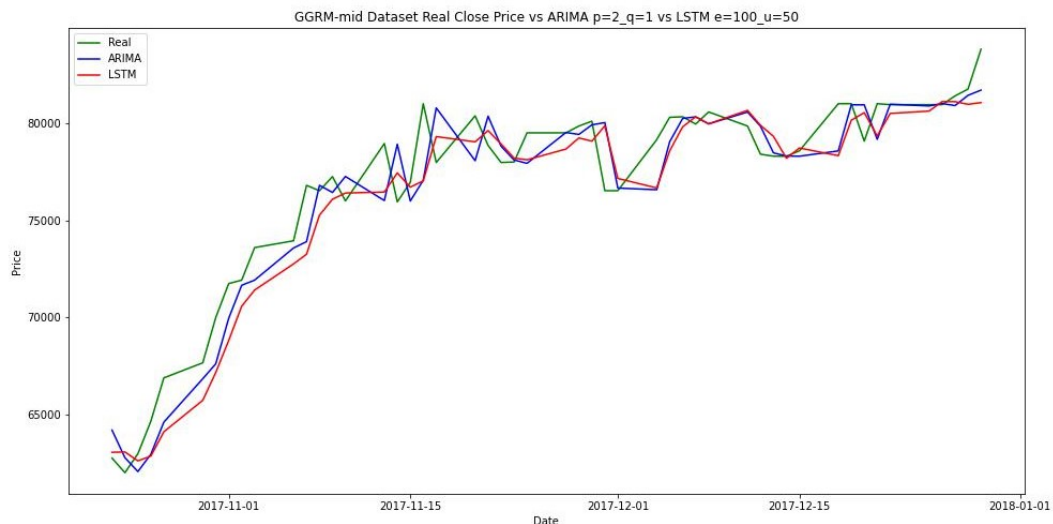
Pada subbab ini akan dibahas pengujian *Mid Term Prediction* menggunakan metode *Autoregressive Integrated Moving Average* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu menengah. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 1604.70, 1212.40, 1.6, dengan menggunakan nilai ordo parameter p, d, q ialah 2,1,1. Lalu, perusahaan UNVR memiliki RMSE, MAE, MAPE yang paling rendah adalah 107.10, 76.80, 0.75, dengan menggunakan nilai ordo parameter ialah 1,1,2. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 29.77, 18.77, 6.29 dengan menggunakan nilai ordo parameter 1,1,1.

Tabel 4.10 Hasil pengujian *Long Short Term Memory* untuk *Mid Term Prediction*

No.	<i>Dataset</i>	Epoch	Units	RMSE	MAE	MAPE
1	GGRM-mid	100	50	1628.43	1296.74	1.7
2	UNVR-mid	1000	128	111.63	83.16	0.81
3	PSDN-mid	1000	10	30.78	19.83	6.62

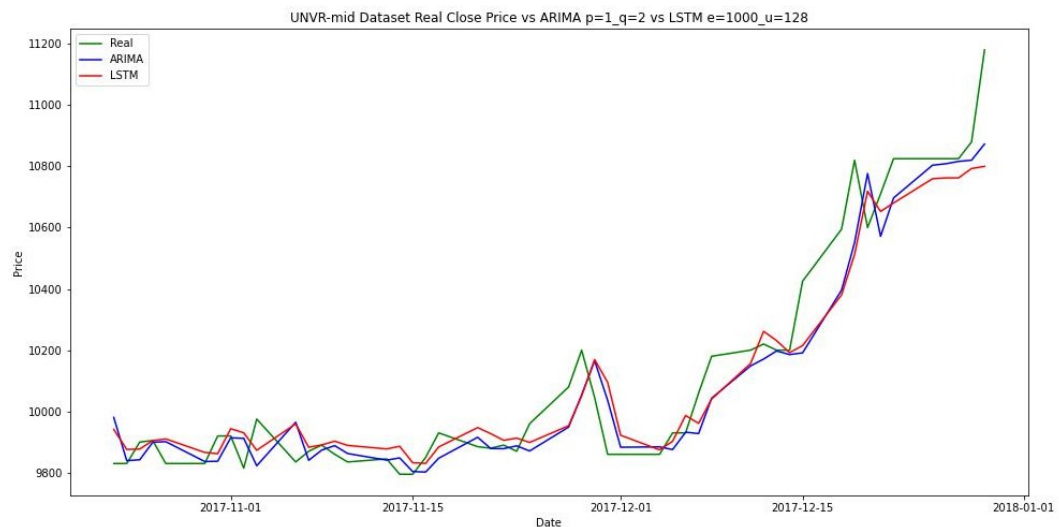
Lalu, pada bagian ini akan dibahas pengujian *Mid Term Prediction* menggunakan metode *Long Short Term Memory* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu menengah. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 1628.43, 1296.74, 1.7, dengan menggunakan nilai *hyperparameter epochs* dan *units* ialah 100 dan 50. Lalu, perusahaan UNVR

memiliki RMSE, MAE, MAPE yang paling rendah adalah 111.63, 83.16, 0.81, dengan menggunakan nilai *epochs* dan *units* ialah 1000 dan 128. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 30.78, 19.83, 6.62 dengan menggunakan nilai *epochs* dan *units* ialah 1000 dan 10.



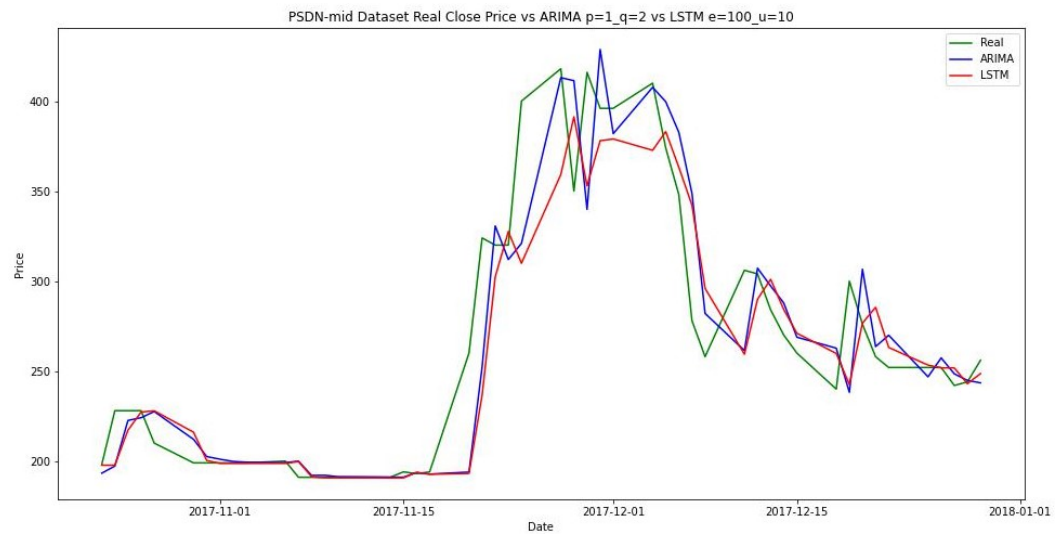
Gambar 4.11 Perbandingan Data GGRM untuk *mid Term*

Pada Gambar 4.11, terlihat hasil prediksi *mid term* untuk saham GGRM dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p,d,q ialah 2,1,1 mendapatkan hasil yang mendekati harga asli dengan mendapatkan hasil RMSE, MAE, MAPE, berturut-turut: 1604.70, 1212.40, 1.6. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epochs* dan *units* ialah 100 dan 50 mendapatkan hasil RMSE, MAE, MAPE: 1628.43, 1296.74, 1.7. Dan terlihat dari grafik setiap metode memprediksi harga saham cukup mendekati dengan nilai harga asli, dan *trend* pada data yang terlihat naik tetapi saat di pertengahan *forecasting* terdapat fluktuasi yang cukup tinggi. Dapat diketahui untuk prediksi *mid term* dengan menggunakan data GGRM, ARIMA dapat memprediksi harga dengan lebih baik daripada Bi-LSTM, karena nilai *error* RMSE, MAE, MAPE yang didapatkan lebih kecil.



Gambar 4.12 Perbandingan Data UNVR untuk *mid Term*

Pada Gambar 4.12, terlihat hasil prediksi *mid term* untuk saham UNVR dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,2, mendapatkan hasil RMSE, MAE, MAPE: 107.10, 76.80, 0.75. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epoch* dan *unit* ialah 1000 dan 128 mendapatkan hasil RMSE, MAE, MAPE: 111.63, 83.16, 0.81, dapat dikatakan bahwa kedua metode tersebut sangat baik untuk memprediksi harga saham dengan jangka waktu menengah dengan menggunakan data UNVR. Namun, ARIMA lebih baik untuk memprediksi, karena nilai *error* RMSE, MAE, MAPE dari ARIMA lebih rendah daripada nilai *error* Bi-LSTM. Terlihat pada Gambar 4.12, *trend* pada data yang sedang terjadi adalah naik. Tetapi, pada hari pertama sampai pertengahan pada data *testing* mungkin terdapat investor yang sedang melakukan *trading* untuk mengambil keuntungan dengan jangka waktu pendek. Karena, gejolak yang dialami pada saat itu selalu naik dan turun di sekitar rentang harga tersebut.



Gambar 4.13 Perbandingan Data PSDN untuk *mid Term*

Pada Gambar 4.13, terlihat hasil prediksi *mid term* untuk saham PSDN dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,1 mendapatkan hasil yang sangat mendekati harga asli, karena nilai *error*-nya hanya 29.77, 18.77, 6.29. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epoch* dan *unit* ialah 1000 dan 10 mendapatkan hasil *error*-nya 30.78, 19.83, 6.62. Dengan memprediksi *mid term* dengan menggunakan data PSDN, ARIMA bisa memprediksi lebih tepat karena nilai *error* yang dihasilkan oleh model lebih rendah daripada nilai *error* oleh model Bi-LSTM. Pada Gambar 4.13, tidak ada *trend* yang sedang terjadi, karena naik dan turun. Tetapi, pada pertengahan pengujian sampai akhir banyak investor melakukan *trading* untuk mengambil keuntungan dengan jangka waktu yang pendek berdasarkan alasan fluktuasi naik-turun harga saham pada waktu itu.



Gambar 4.14 Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian *mid term prediction* ARIMA dan Bi-LSTM

Pada Gambar 4.14, menunjukkan perbandingan nilai rata-rata keseluruhan dari pengukuran nilai *error* RMSE, MAE, MAPE untuk dataset GGRM, UNVR, PSDN prediksi *mid term*. Dari *plots* tersebut menunjukkan bahwa, ARIMA memprediksi harga saham lebih baik daripada Bi-LSTM. Karena, nilai *error* yang dihasilkan lebih kecil daripada Bi-LSTM.

4.4.5 Pembahasan Pengujian *Short Term Prediction*

Tabel 4.11 Hasil pengujian *Autoregressive Integrated Moving Average* untuk *Short Term Prediction*

No.	Dataset	p	d	q	RMSE	MAE	MAPE
1	GGRM-short	1	1	1	606.12	482.77	0.74
2	UNVR-short	1	1	2	111.80	75.07	0.87
3	PSDN-short	1	1	1	5.36	3.32	2.2

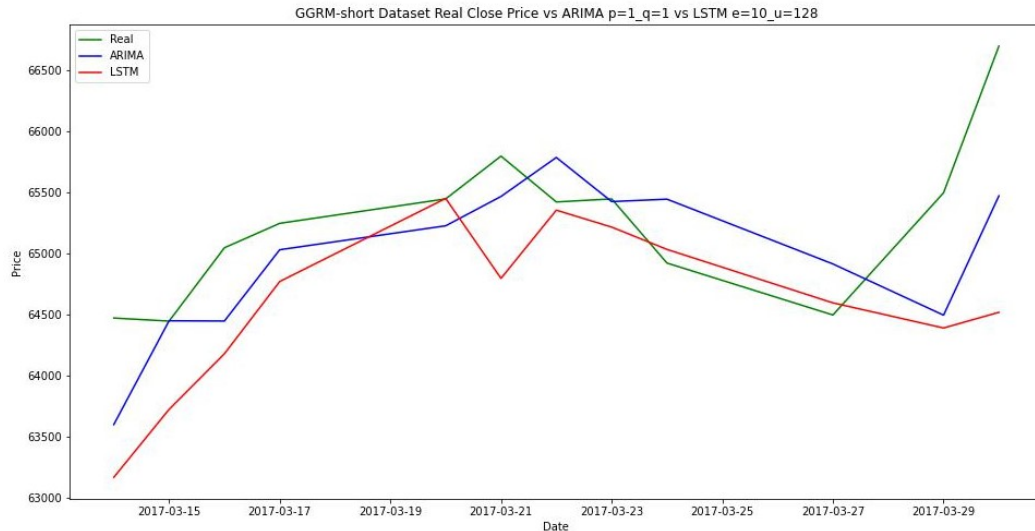
Pada subbab ini akan dibahas pengujian *Short Term Prediction* menggunakan metode *Autoregressive Integrated Moving Average* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu pendek. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 1604.70, 1212.40, 1.6, dengan menggunakan nilai ordo parameter p,d,q ialah 1,1,1. Lalu, perusahaan UNVR memiliki RMSE, MAE, MAPE yang paling rendah adalah 111.80, 75.07, 0.87, dengan menggunakan nilai ordo parameter ialah 1,1,2. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 5.36, 3.32, 2.2 dengan menggunakan nilai ordo parameter 1,1,1.

Tabel 4.12 Hasil pengujian *Long Short Term Memory* untuk *Short Term Prediction*

No.	Dataset	Epoch	Units	RMSE	MAE	MAPE
1	GGRM-short	10	128	759.84	593.17	0.91
2	UNVR-short	1000	128	126.76	105.62	1.21
3	PSDN-short	100	128	6.4	3.75	2.49

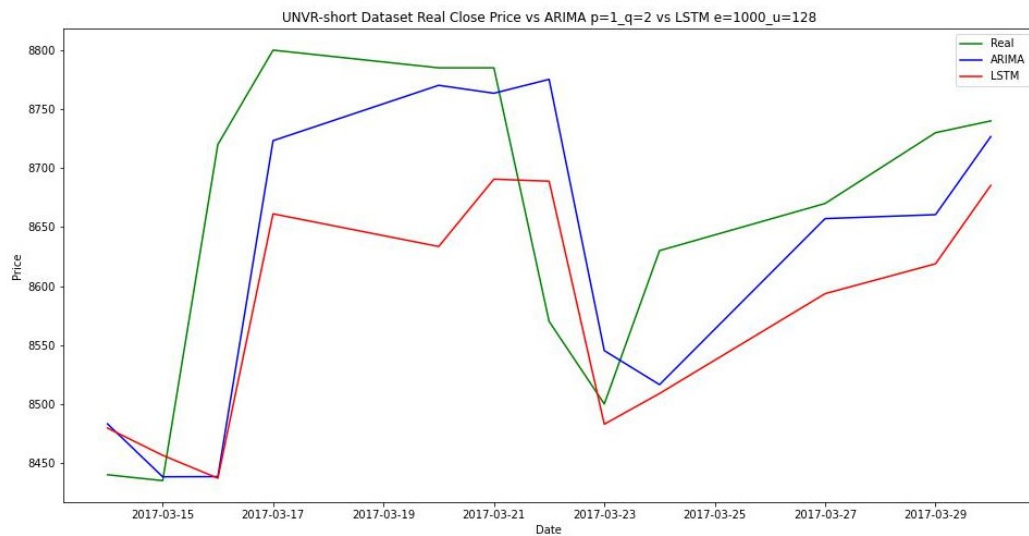
Lalu, pada bagian ini akan dibahas pengujian *Short Term Prediction* menggunakan metode *Long Short Term Memory* untuk memprediksi harga saham dari masing-masing perusahaan dengan jangka waktu pendek. Hasil pengukuran menunjukkan bahwa perusahaan GGRM memiliki RMSE, MAE, MAPE yang paling rendah adalah 759.84, 593.17, 0.91, dengan menggunakan nilai *hyperparameter epochs* dan *units* ialah 10 dan 128. Lalu, perusahaan UNVR memiliki RMSE, MAE, MAPE

yang paling rendah adalah 126.76, 105.62, 1.21, dengan menggunakan nilai *epochs* dan *units* ialah 1000 dan 128. Dan, perusahaan PSDN memiliki pengukuran *error* yang paling rendah adalah 6.4, 3.75, 2.49 dengan menggunakan nilai *epochs* dan *units* ialah 100 dan 128.



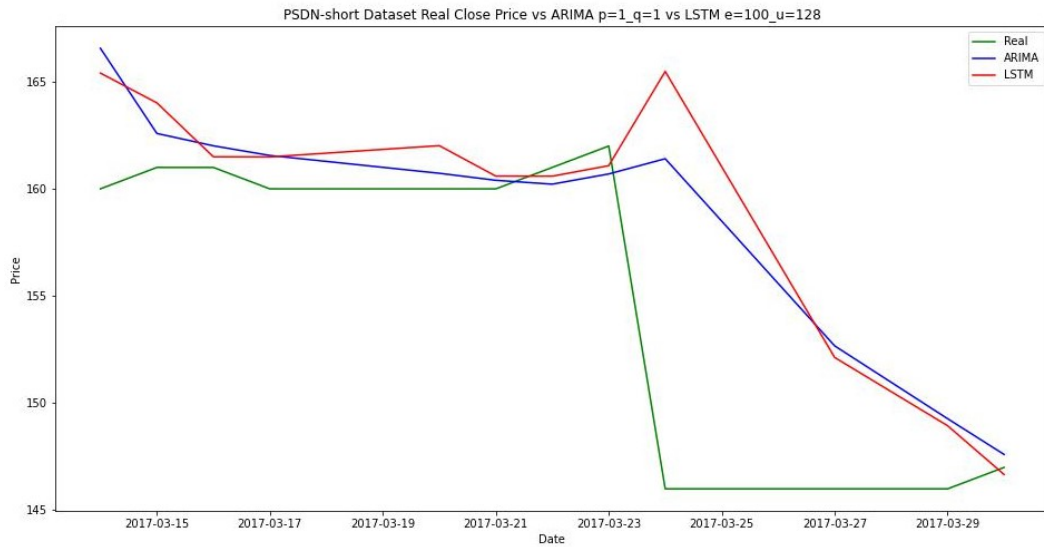
Gambar 4.15 Perbandingan Data GGRM untuk *Short Term*

Pada Gambar 4.15, terlihat hasil prediksi *short term* untuk saham GGRM dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,1, mendapatkan hasil *error*-nya ialah 606.12, 482.77, 0.74. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epoch* dan *unit* ialah 10 dan 128 mendapatkan hasil *error*-nya 759.84, 593.17, 0.91. Dapat disimpulkan, ARIMA lebih baik untuk memprediksi *short term* dengan data GGRM, karena nilai *error* RMSE, MAE, MAPE dari ARIMA lebih rendah daripada nilai *error* Bi-LSTM. Terlihat dari grafik, *trend* yang sedang terjadi ialah menaik. Pada Gambar 4.15, dari hari pertama sampai berakhirnya prediksi, metode Bi-LSTM selalu jauh berada di bawah hasil prediksi dari ARIMA dan nilai asli yang menyebabkan nilai *error* pada data *relative* tinggi.



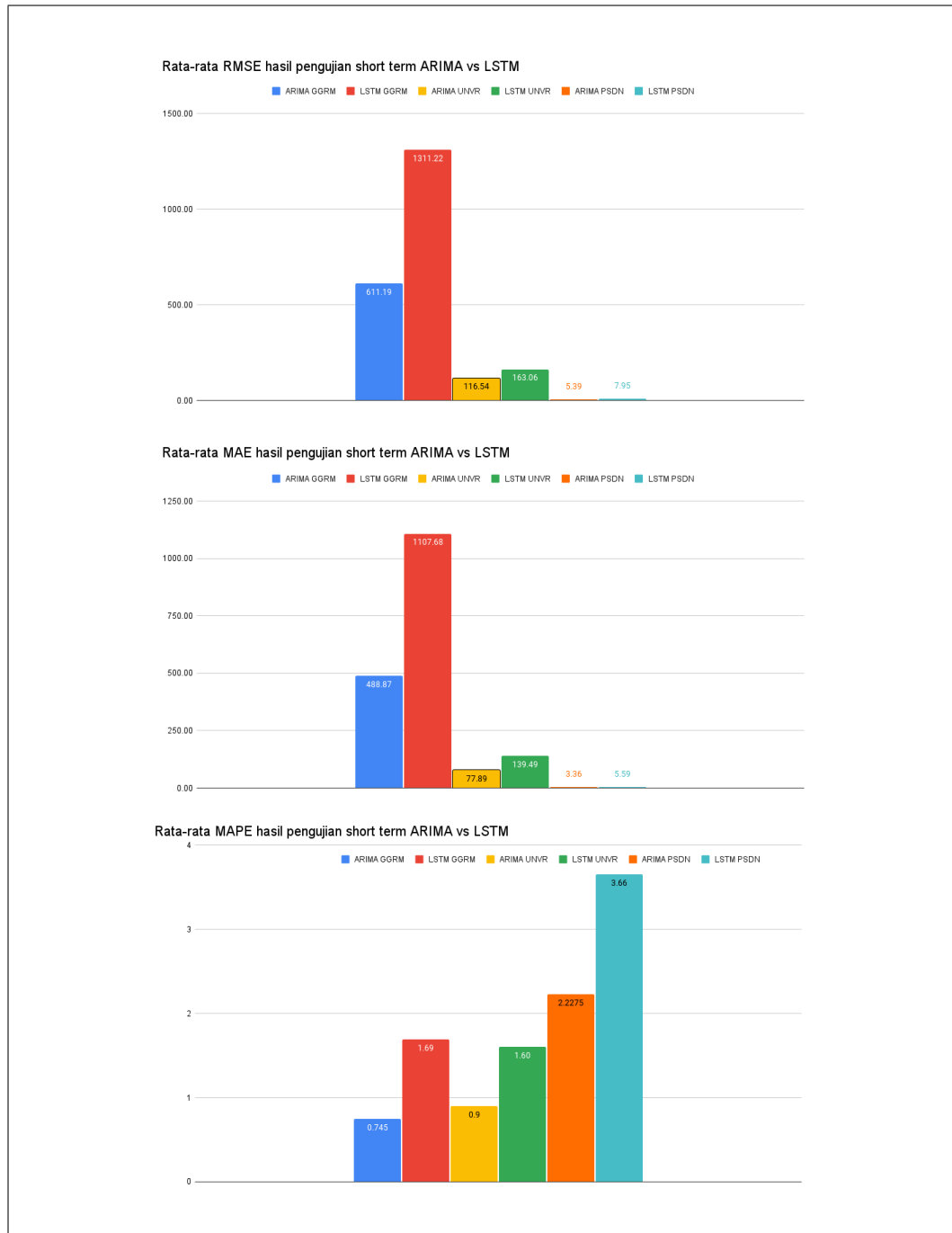
Gambar 4.16 Perbandingan Data UNVR untuk *Short Term*

Pada Gambar 4.16, terlihat hasil prediksi *short term* untuk saham UNVR dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,2, mendapatkan hasil RMSE, MAE, MAPE: 111.80, 75.07, 0.87. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epoch* dan *unit* ialah 1000 dan 128 mendapatkan hasil RMSE, MAE, MAPE: 126.76, 105.62, 1.21, dapat dikatakan bahwa kedua metode tersebut sangat baik untuk memprediksi harga saham dengan jangka waktu pendek dengan menggunakan data UNVR. Namun, ARIMA lebih baik untuk memprediksi, karena nilai *error* RMSE, MAE, MAPE dari ARIMA lebih rendah daripada nilai *error* Bi-LSTM. Terlihat pada Gambar 4.16, *trend* pada data yang sedang terjadi ialah naik. Walaupun, pada hari pertama sampai terakhir prediksi pada data *testing* mungkin terdapat investor yang sedang melakukan *trading* untuk mengambil keuntungan dengan jangka waktu pendek. Karena, fluktuasi/gejolak yang dialami pada saat itu selalu naik dan turun di sekitar rentang harga tersebut.



Gambar 4.17 Perbandingan Data PSDN untuk *Short Term*

Pada Gambar 4.17, terlihat hasil prediksi *short term* untuk saham PSDN dari setiap metode. Metode ARIMA dengan menggunakan nilai parameter orde p, d, q ialah 1,1,1 mendapatkan hasil yang sangat mendekati harga asli, karena nilai *error*-nya hanya 5.36, 3.32, 2.2. Sementara, metode Bi-LSTM dengan menggunakan *hyperparameter epoch* dan *unit* ialah 100 dan 128 mendapatkan hasil RMSE 6.4, 3.75, 2.49. Dengan memprediksi *short term* dengan menggunakan data PSDN, ARIMA bisa memprediksi lebih tepat karena nilai *error* yang dihasilkan oleh model ARIMA lebih rendah daripada nilai *error* oleh model Bi-LSTM. Pada Gambar 4.17, *trend* yang sedang terjadi ialah turun. Karena, sesuai grafik yang ditunjukkan, pada hari pertama sampai terakhir selalu menurun harganya.



Gambar 4.18 Perbandingan rata-rata RMSE,MAE,MAPE hasil pengujian *short term prediction* ARIMA dan Bi-LSTM

Pada Gambar 4.18, menunjukkan perbandingan nilai rata-rata keseluruhan dari pengukuran nilai *error* RMSE, MAE, MAPE untuk dataset GGRM, UNVR, PSDN prediksi *short term*. Dari *plots* tersebut menunjukkan bahwa, ARIMA memprediksi harga saham lebih baik daripada Bi-LSTM. Karena, nilai *error* yang dihasilkan lebih kecil daripada Bi-LSTM.

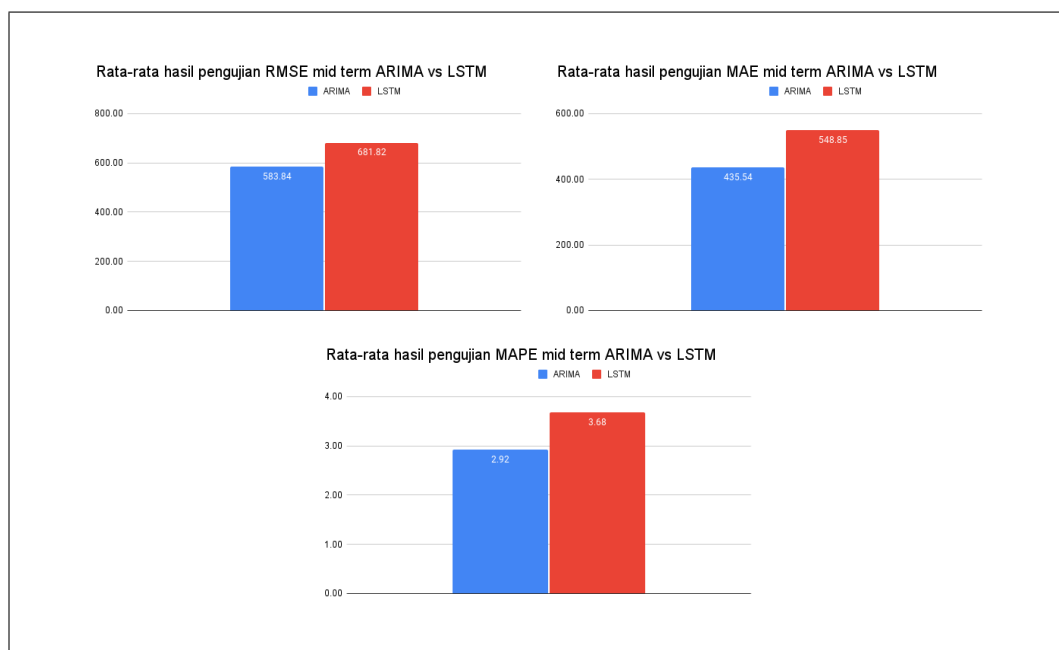
4.4.6 Pembahasan Pengujian

Dari semua hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa ARIMA mendapatkan hasil yang terbaik dibandingkan Bi-LSTM.



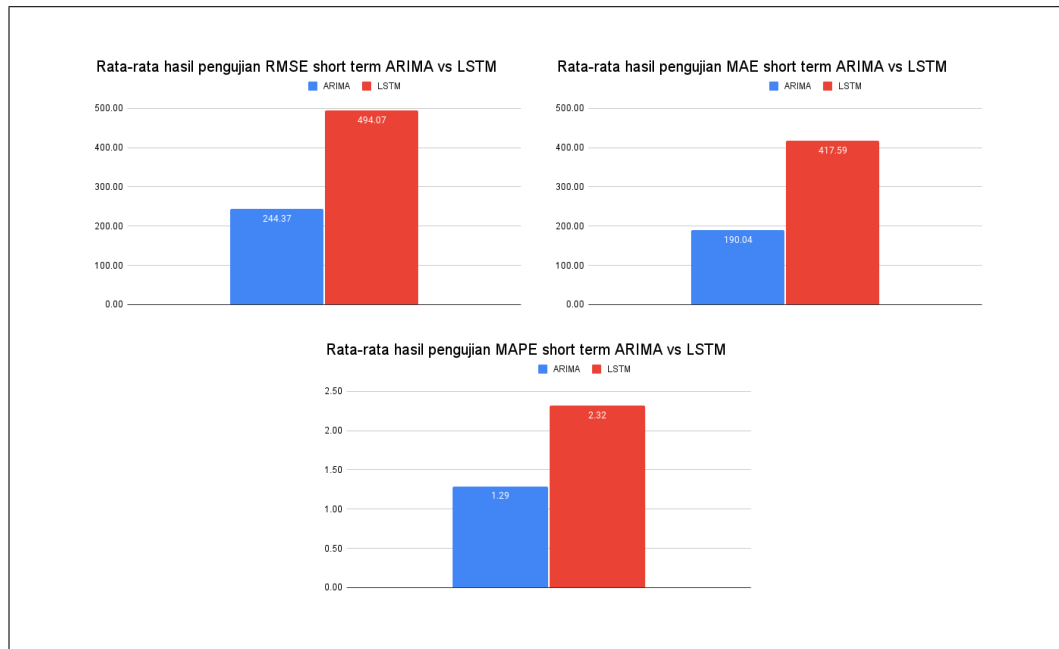
Gambar 4.19 Perbandingan nilai rata-rata *long term prediction* RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM

Pada Gambar 4.19 dapat diketahui hasil pengujian *long term* ARIMA mendapatkan nilai rata-rata *error* RMSE, MAE, MAPE yang lebih kecil dari pada Bi-LSTM.



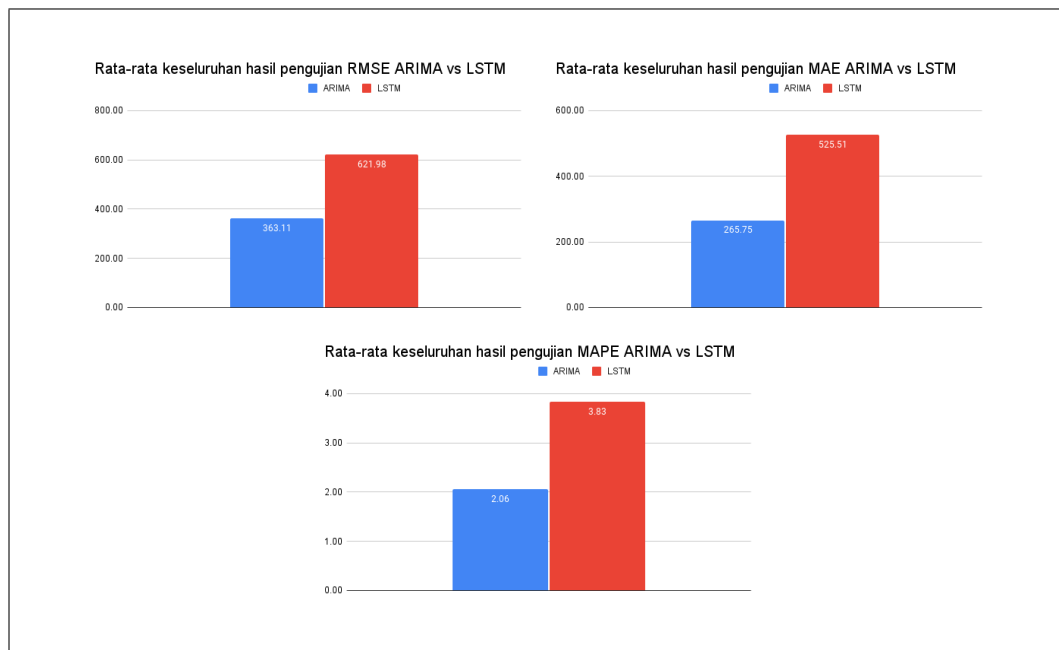
Gambar 4.20 Perbandingan nilai rata-rata *mid term prediction* RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM

Pada Gambar 4.20 dapat diketahui hasil pengujian *mid term* ARIMA mendapatkan nilai rata-rata *error* RMSE, MAE, MAPE yang lebih kecil dari pada Bi-LSTM.



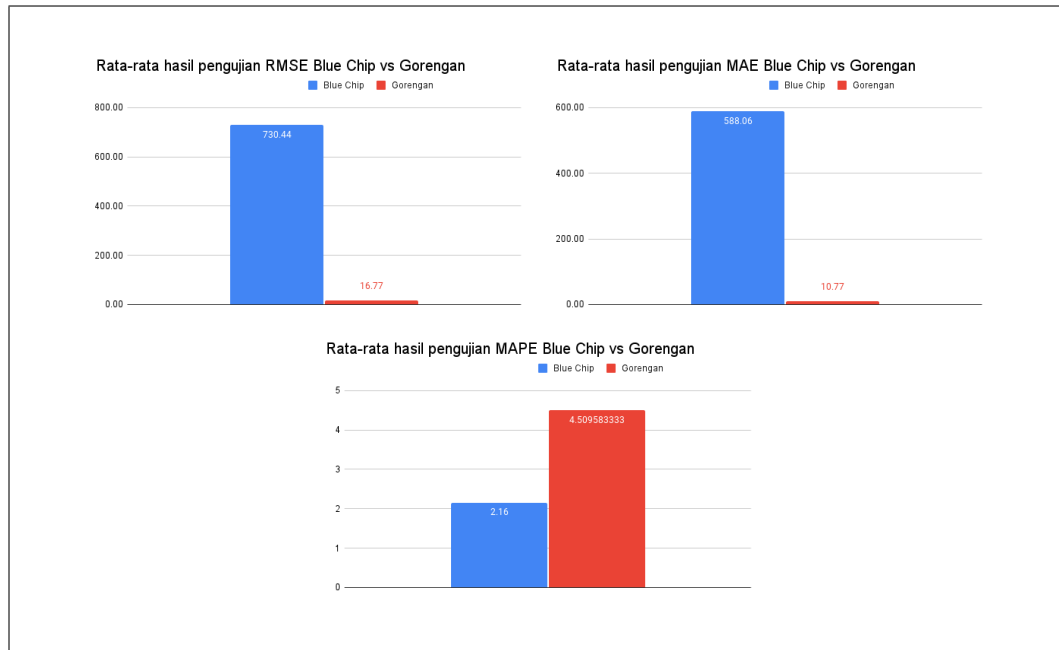
Gambar 4.21 Perbandingan nilai rata-rata *short term prediction* RMSE, MAE, MAPE dari ARIMA dan Bi-LSTM

Pada Gambar 4.21 dapat diketahui hasil pengujian *short term* ARIMA mendapatkan nilai rata-rata *error* RMSE, MAE, MAPE yang lebih kecil dari pada Bi-LSTM.



Gambar 4.22 Rata-rata hasil pengujian RMSE, MAE, MAPE Bi-LSTM vs ARIMA

Pada Gambar 4.22, menunjukkan perbandingan nilai rata-rata keseluruhan dari pengukuran nilai *error* RMSE, MAE, MAPE. Dari *plots* tersebut menunjukkan bahwa, ARIMA memprediksi harga saham lebih baik daripada Bi-LSTM. Karena, nilai *error* yang dihasilkan lebih kecil daripada Bi-LSTM.



Gambar 4.23 Rata-rata hasil pengujian RMSE, MAE, MAPE Blue Chip vs Gorengan

Pada Gambar 4.23 dijelaskan bahwa pengujian yang menggunakan data *Blue chip* pada perusahaan PT. Gudang Garam, PT. Unilever Indonesia dan Gorengan pada perusahaan PT. Prasadha Aneka Niaga di setiap jangka waktu yang diprediksi. Saham dari perusahaan PT. Prasadha Aneka Niaga lebih cocok untuk melakukan prediksi dengan menggunakan ARIMA atau Bi-LSTM, disebabkan RMSE, MAE, MAPE yang didapatkan sangat jauh lebih baik daripada perusahaan *Blue Chip* dan harga PSDN lebih murah (ratusan) dibandingkan harga saham Gudang Garam atau Unilever yang harganya ribuan dan puluhan ribu.

Tabel 4.13 Ringkasan keseluruhan hasil pengujian

No	Method	Dataset	Average		
			RMSE	MAE	MAPE
1	Bi-LSTM	GGRM long term	1581.16	1414.83	4.11%
2		UNVR long term	480.44	409.54	8.72%
3		PSDN long term	8.53	5.94	3.63%
4		GGRM mid term	1851.40	1510.92	1.97%
5		UNVR mid term	153.63	110.29	1.07%
6		PSDN mid term	40.43	25.33	8.02%
7		GGRM short term	1311.22	1107.68	1.69%
8		UNVR short term	163.06	139.49	1.60%
9		PSDN short term	7.95	5.59	3.66%
10	ARIMA	GGRM long term	664.71	426.83	1.18%
11		UNVR long term	111.11	83.06	1.58%
12		PSDN long term	7.58	5.15	3.11%
13		GGRM mid term	1613.47	1210.69	1.60%
14		UNVR mid term	107.32	76.66	0.75%
15		PSDN mid term	30.72	19.27	6.42%
16		GGRM short term	611.19	488.87	0.75%
17		UNVR short term	116.54	77.89	0.90%
18		PSDN short term	5.39	3.36	2.23%

Dari seluruh hasil pengujian dari metode ARIMA dan Bi-LSTM pada bagian 4.4.3 hingga 4.4.5, maka ketekaitan konfigurasi jumlah *epoch*, *unit* pada metode Bi-LSTM dan konfigurasi jumlah *p* dan *q* pada metode ARIMA secara menyeluruh terhadap nilai RMSE, MAE, MAPE untuk *short*, *mid*, *long term forecasting* dapat dijelaskan sebagai berikut.

1. Metode ARIMA memiliki performa rata-rata yang lebih baik jika dibandingkan dengan model Bi-LSTM. Dapat dilihat dari nilai rata-rata RMSE, MAE, MAPE dari metode ARIMA lebih kecil daripada RMSE, MAE, MAPE metode Bi-LSTM. Hal ini disebabkan, Bi-LSTM menggunakan jumlah fitur yang lebih banyak, tidak hanya *Close* tetapi menggunakan *Open*, *High*, *Low*, *Volume* untuk

observarsi dan melakukan prediksi, sementara ARIMA hanya menggunakan fitur *Close* untuk *observarsi* dan prediksi.

2. Saham Gorengan dari perusahaan Prasadha Aneka Niaga, mendapatkan nilai error RMSE, MAE yang lebih rendah daripada perusahaan saham *blue chip*. Jika dilihat pada Gambar 4.9 saham tersebut lebih bergejolak dan tidak ada *trend* yang sedang terjadi.
3. Untuk keseluruhan hasil pengujian nilai *error* untuk *short term forecasting* terlihat lebih kecil dibandingkan dengan ketika melakukan *mid, long term forecasting*. Hal ini disebabkan jumlah data pada *short term forecasting* lebih sedikit dibandingkan dengan jumlah data pada *mid, long term forecasting*.

BAB 5 KESIMPULAN DAN SARAN

Pada bab ini dibahas mengenai kesimpulan penelitian, pelatihan, pengujian yang dilakukan oleh peneliti. Selain itu, terdapat juga saran yang dapat digunakan atau dipertimbangkan pada saat melakukan penelitian di masa mendatang.

5.1 Kesimpulan

Kesimpulan dari metode *Autoregressive Integrated Moving Average* dan *Long Short Term Memory* pada prediksi harga saham adalah:

1. Nilai rata-rata RMSE, MAE, MAPE ARIMA berturut-turut ialah 363.11, 266.75, 2.06. Lalu, nilai rata-rata RMSE, MAE, MAPE Bi-LSTM ialah 621.98, 525.51, 3.83.
2. Pada data saham untuk prediksi jangka waktu panjang, menengah, pendek (*long, mid, short term*), secara rata-rata nilai *error* ARIMA lebih rendah daripada rata-rata nilai *error* Bi-LSTM. Hal itu disebabkan Bi-LSTM menggunakan jumlah fitur yang lebih banyak tidak hanya *Close* tetapi menggunakan *Open, High, Low, Volume* untuk *observasi* dan melakukan prediksi, sementara ARIMA hanya menggunakan fitur *Close* untuk *observasi* dan prediksi.
3. Jumlah *hyperparameter epoch* dan *unit* pada metode Bi-LSTM sangat berpengaruh pada prediksi harga saham. Walaupun pada saham PSDN dengan prediksi *mid term* dengan menggunakan jumlah *unit* 10 mendapatkan nilai *error* yang lebih rendah.
4. Pada metode ARIMA, jumlah ordo *p* dan *q* tidak terlalu berpengaruh saat melakukan prediksi harga saham. Karena, banyak jumlah ordo *p* dan *q* bernilai 1 yang lebih baik daripada jumlah ordo *p* dan *q* yang bernilai 2
5. Pada pengujian yang menggunakan data *Blue chip* pada perusahaan PT. Gudang Garam (GGRM), PT. Unilever Indonesia (UNVR) dan Gorengan pada perusahaan PT. Prasidha Aneka Niaga (PSDN) di setiap jangka waktu yang diprediksi. Saham dari perusahaan PT. Prasidha Aneka Niaga (PSDN) lebih cocok untuk melakukan prediksi dengan menggunakan metode pada penelitian ini. Disebabkan RMSE, MAE, MAPE yang didapatkan sangat jauh lebih baik daripada perusahaan *Blue Chip* dan harga PSDN lebih murah (ratusan) dibandingkan harga saham GGRM atau UNVR yang harganya ribuan dan puluhan ribu.

5.2 Saran

Saran dari peneliti untuk pengembangan model prediksi harga saham di masa mendatang adalah:

1. Menggunakan beberapa dataset saham dari berbagai sektor yang *blue chip* dan *gorengan*, sehingga lebih mengetahui metode apa yang lebih cocok pada setiap sektor.
2. Menambahkan spektrum *hyperparameter tuning* pada Bi-LSTM, supaya hasil RMSE, MAE, MAPE yang didapatkan lebih baik.
3. Menerapkan metode lanjutan dari *Recurrent Neural Network* (RNN) seperti GRU atau metode *Machine Learning* lainnya.

DAFTAR REFERENSI

- [1] M. Taufiq, W. R. Dharmawan, N. Jannah, "Peran dan Kontribusi Pasar Modal terhadap Pertumbuhan Ekonomi Indonesia," TRIANGLE : Journal of Management, Accounting, Economic and Business, vol 2, no. 4, 2021.
- [2] W. Y. Rusyida, V. Y. Pratama, "Prediksi Harga Saham Garuda Indonesia di Tengah Pandemi COVID-19 Menggunakan Metode ARIMA," SQUARE : Journal of Mathematics and Mathematics Education vol 2, no. 1, pp. 73-81, 2020.
- [3] S. A. Wadi, M. Almasarweh, A. A. Alsaraireh, "Predicting Close Price Time Series Data Using ARIMA Model," Modern Applied Science, Published by Canadian Center of Science and Education, vol. 12, no. 11, 2018.
- [4] M. Nikou, G. Mansourfar, J. Bagherzadeh, "Stock Price Prediction using Deep Learning Algorithm and its Comparison with Machine Learning Algorithm," Intelligent Systems in Accounting Finance & Management, December 2019.
- [5] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, Shahab S, "Deep Learning for Stock Market Prediction," Entropy, vol. 22, 2020.
- [6] W. Hastomo, A. S. B. Karno, N. Kalbuana, E. Nisfiani, E. T. P. Lussiana, "Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi COVID-19," Jurnal Edukasi dan Penelitian Informatika, vol. 7, 2021.
- [7] J. Qiu, B. Wang, C. Zhou, "Forecasting Stock Prices with Long-Short Term Memory Neural Network Based on Attention Mechanism," Journal Plos One, 3 Januari 2020.
- [8] S. S. Namini, N. Tavakoli, A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 17th IEEE International Conference on Machine Learning and Applications, 2018.
- [9] H. Roondiwala, H. Patel, S. Varma, "Predicting Stock Prices Using LSTM," International Journal of Science and Research, vol. 6, April 2017.
- [10] A. Geron, *Hands On Machine Learning with ScikitLearn, Keras & Tensorflow*, O'Reilly Media, Inc, 2019
- [11] M. Homma. *The Candle Stick Trading Bible*. Independently Published by Munehisa Homma, April 2020
- [12] A. Zhang, Z. C. Lipton, M. Li, and Alexander, *Dive into Deep learning*, Release 0.17.0, 2021
- [13] F. R. Pratama, "Implementasi Algoritme Random Forest untuk Prediksi Pergerakan Harga Saham," Dept. Informatika ITHB, 2017.
- [14] J. D. Schwager. *Getting Started in Technical Analysis*. John Wiley & Sons,

1999

- [15] Ian G., Yoshua B., Aaron C., *Deep Learning*. MIT Press, 2016
- [16] Towardsdatascience. "Forecast KPIs: RMSE, MAE, MAPE & Bias". [Online]. Available at: <https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d> [Accessed: Feb. 20, 2022]
- [17] A Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Illustrated Edition. Springer, 2012
- [18] D. Stifanic, J. Musulin, A. Miocevic, S. B. Segota, R. Subic. Z. Car, "Impact of COVID-19 on Forecasting Stock Prices: An Integration of Stationary Wavelet Transform and Bidirectional Long Short-Term Memory," Hindawi Complexity, 2020.
- [19] Yahoo Finance, 2022. "Historical Price Data Indonesia Equity". [Online]. Available: <https://finance.yahoo.com/> [Accessed: Jan. 31, 2022].
- [20] R. J. Hyndman, *Forecasting: Principles & Practice*, Illustrated Edition. Springer, 2014
- [21] MBA Knowledge Base, 2021. Time Horizon in Forecasting. [Online]. Available: <https://www.mbaknol.com/managerial-economics/time-horizon-in-forecasting/> [Accessed: March. 28, 2022]
- [22] M. Pradana. Saham *Blue Chip* dan Saham Gorengan . Available: <https://investbro.id/saham-gorengan/> [Accessed: Apr. 12, 2022]
- [23] F Bre, J. M. Gimenez, V. D. Fachinotti. Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. [Online]. Available: researchgate [Accessed: Feb. 22, 2022]
- [24] Geeksforgeeks, 2019. Implement sigmoid function using Numpy. [Online]. Available: geeksforgeeks [Accessed: Feb. 24, 2022]
- [25] X. Wang, Y. Liu, C. Sun, B. Wang and X. Wang, "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1343–1353, Beijing, China, July 26-31, 2015.
- [26] Towardsdatascience, 2019. The Most Intuitive and Easiest Guide for Recurrent Neural Network. [Online]. Available: Towardsdatascience [Accessed: Mar. 24, 2022]
- [27] Q. Chen, W. Zhang, Y. Lou, "Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism Multi-Later Perceptron, and Bidirectional Long-Short Term Memory Neural Network," School of

DAFTAR REFERENSI

- Information Management and Artificial Intelligence, Zhejiang University of Finance and Economics, China, Beijing, China, vol. 8, 2020.
- [28] M. Jia, J. Huang, L. Pang, Q. Zhao, "Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network," *Advances in Computer Science Research*, vol. 90, 2019.
- [29] Tensorflow Keras, 1997. Long Short Term Memory. [Online]. Available: LSTM Keras[Accessed: Des. 09, 2022]

LAMPIRAN A HASIL PENGUJIAN ARIMA

Tabel A-1 Pengukuran *error* yang digunakan pada Jurnal

<i>Jurnal Penelitian</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>
[2] W. Y. Rusyida, V. Y. Pratama, "Prediksi Harga Saham Garuda Indonesia di Tengah Pandemi COVID-19 Menggunakan Metode ARIMA," SQUARE : Journal of Mathematics and Mathematics Education vol 2, no. 1, pp. 73-81, 2020	Ada	Tidak ada	Tidak ada
[3] S. A. Wadi, M. Almasarweh, A. A. Alsaraireh, "Predicting Close Price Time Series Data Using ARIMA Model," Modern Applied Science, Published by Canadian Center of Science and Education, vol. 12, no. 11, 2018	Ada	Tidak ada	Tidak ada
[4] M. Nikou, G. Mansourfar, J. Bagherzadeh, "Stock Price Prediction using Deep Learning Algorithm and its Comparison with Machine Learning Algorithm," Intelligent Systems in Accounting Finance & Management, December 2019.	Ada	Ada	Tidak ada

LAMPIRAN A HASIL PENGUJIAN ARIMA

[5] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, Shahab S, "Deep Learning for Stock Market Prediction," Entropy, vol. 22, 2020	Ada	Ada	Ada
[6] WW. Hastomo, A. S. B. Karno, N. Kalbuana, E. Nisfiani, E. T. P. Lussiana, "Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi COVID19," Jurnal Edukasi dan Penelitian Informatika, vol. 7, 2021.	Ada	Tidak ada	Tidak ada
[7] J. Qiu, B. Wang, C. Zhou, "Forecasting Stock Prices with Long-Short Term Memory Neural Network Based on Attention Mechanism," Journal Plos One, 3 Januari 2020	Ada	Ada	Tidak ada
[8] S. S. Namini, N. Tavakoli, A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 17th IEEE International Conference on Machine Learning and Applications, 2018	Ada	Tidak ada	Tidak ada
[9] H. Roondiwala, H. Patel, S. Varma, "Predicting Stock Prices Using LSTM," International Journal of Science and Research, vol. 6, April 2017.	Ada	Tidak ada	Tidak ada

LAMPIRAN B HASIL PENGUJIAN LSTM

Tabel B-1 Hasil pengujian *Autoregressive Integrated Moving Average*

No.	<i>Dataset</i>	p	d	q	<i>RMSE</i>	<i>MAE</i>	MAPE
1	GGRM-long	1	1	1	668.08	428.97	1.19
2	GGRM-long	2	1	1	663.47	426.03	1.18
3	GGRM-long	1	1	2	663.56	426.09	1.18
4	GGRM-long	2	1	2	663.74	426.22	1.18
5	GGRM-mid	1	1	1	1616.72	1208.81	1.59
6	GGRM-mid	2	1	1	1604.70	1212.4	1.6
7	GGRM-mid	1	1	2	1615.38	1212.68	1.6
8	GGRM-mid	2	1	2	1617.07	1208.88	1.59
9	GGRM-short	1	1	1	606.12	482.77	0.74
10	GGRM-short	2	1	1	610.18	485.81	0.74
11	GGRM-short	1	1	2	610.29	480.84	0.73
12	GGRM-short	2	1	2	618.15	506.05	0.77
13	UNVR-long	1	1	1	111.41	83.66	1.59
14	UNVR-long	2	1	1	110.93	82.74	1.57
15	UNVR-long	1	1	2	111.08	82.90	1.58
16	UNVR-long	2	1	2	111.00	82.95	1.58
17	UNVR-mid	1	1	1	107.40	76.82	0.75

LAMPIRAN B HASIL PENGUJIAN LSTM

18	UNVR-mid	2	1	1	107.11	76.68	0.75
19	UNVR-mid	1	1	2	107.10	76.80	0.75
20	UNVR-mid	2	1	2	107.68	76.34	0.74
21	UNVR-short	1	1	1	115.24	77.24	0.89
22	UNVR-short	2	1	1	112.87	74.86	0.86
23	UNVR-short	1	1	2	111.80	75.07	0.87
24	UNVR-short	2	1	2	126.23	84.40	0.98
25	PSDN-long	1	1	1	7.65	5.20	3.14
26	PSDN-long	2	1	1	7.56	5.14	3.11
27	PSDN-long	1	1	2	7.54	5.12	3.1
28	PSDN-long	2	1	2	7.55	5.13	3.1
29	PSDN-mid	1	1	1	29.83	18.23	6.1
30	PSDN-mid	2	1	1	30.35	19.19	6.41
31	PSDN-mid	1	1	2	29.77	18.77	6.29
32	PSDN-mid	2	1	2	32.91	20.87	6.87
33	PSDN-short	1	1	1	5.36	3.32	2.2
34	PSDN-short	2	1	1	5.38	3.34	2.22
35	PSDN-short	1	1	2	5.45	3.39	2.25
36	PSDN-short	2	1	2	5.38	3.40	2.24

Tabel B-2 Hasil pengujian *Long Short Term Memory*

No.	<i>Dataset</i>	<i>Epoch</i>	<i>Units</i>	<i>RMSE</i>	<i>MAE</i>	MAPE
1	GGRM-long	10	10	2402.95	2271.62	6.61
2	GGRM-long	10	50	2158.8	2003.55	5.83
3	GGRM-long	10	128	1852.62	1711.51	4.97
4	GGRM-long	100	10	1329.92	1176.85	3.41
5	GGRM-long	100	50	870.73	687.11	1.94
6	GGRM-long	100	128	733.23	506.21	1.41
7	GGRM-long	1000	10	1625.46	1464.5	4.28
8	GGRM-long	1000	50	1745.52	1581.73	4.63
9	GGRM-long	1000	128	1511.17	1330.38	3.91
10	GGRM-mid	10	10	2756.34	2384.68	3.06
11	GGRM-mid	10	50	1826.38	1538.94	2.01
12	GGRM-mid	10	128	1634.99	1312.43	1.73
13	GGRM-mid	100	10	1677.53	1359.26	1.79
14	GGRM-mid	100	50	1628.43	1296.74	1.7
15	GGRM-mid	100	128	1689.45	1346.98	1.76
16	GGRM-mid	1000	10	1902.84	1542.3	2
17	GGRM-mid	1000	50	1702.14	1337.41	1.74
18	GGRM-mid	1000	128	1844.5	1479.53	1.92
19	GGRM-short	10	10	3110.18	3062.72	4.69

LAMPIRAN B HASIL PENGUJIAN LSTM

20	GGRM-short	10	50	1162.06	988	1.51
21	GGRM-short	10	128	759.84	593.17	0.91
22	GGRM-short	100	10	1106.76	840.55	1.28
23	GGRM-short	100	50	1166.44	914.79	1.4
24	GGRM-short	100	128	1145.97	894.35	1.37
25	GGRM-short	1000	10	1134.87	919.93	1.41
26	GGRM-short	1000	50	1208.54	1004.2	1.53
27	GGRM-short	1000	128	1006.29	751.44	1.15
28	UNVR-long	10	10	619.3	584.22	11.97
29	UNVR-long	10	50	549.51	503.58	10.5
30	UNVR-long	10	128	416.37	375.77	7.85
31	UNVR-long	100	10	380.13	331.63	7.03
32	UNVR-long	100	50	282.24	253.05	5.23
33	UNVR-long	100	128	216.92	187.06	3.89
34	UNVR-long	1000	10	721.84	562.19	12.47
35	UNVR-long	1000	50	880.98	669.2	14.95
36	UNVR-long	1000	128	256.64	219.13	4.63
37	UNVR-mid	10	10	306.2	244.03	2.36
38	UNVR-mid	10	50	180.57	133.63	1.29
39	UNVR-mid	10	128	143.07	97.44	0.94

LAMPIRAN B HASIL PENGUJIAN LSTM

40	UNVR-mid	100	10	126.94	83.56	0.81
41	UNVR-mid	100	50	127.39	84.89	0.82
42	UNVR-mid	100	128	120.99	80.59	0.78
43	UNVR-mid	1000	10	131.97	92.21	0.89
44	UNVR-mid	1000	50	133.91	93.12	0.9
45	UNVR-mid	1000	128	111.63	83.16	0.81
46	UNVR-short	10	10	347.26	324.76	3.74
47	UNVR-short	10	50	173.4	148.03	1.7
48	UNVR-short	10	128	128.89	101.22	1.17
49	UNVR-short	100	10	135.68	113.32	1.3
50	UNVR-short	100	50	139.77	114.82	1.32
51	UNVR-short	100	128	134.86	111.59	1.28
52	UNVR-short	1000	10	142.51	120.41	1.38
53	UNVR-short	1000	50	138.4	115.65	1.33
54	UNVR-short	1000	128	126.76	105.62	1.21
55	PSDN-long	10	10	8.26	5.44	3.34
56	PSDN-long	10	50	8.18	5.47	3.33
57	PSDN-long	10	128	7.97	5.27	3.26
58	PSDN-long	100	10	8.25	5.44	3.29
59	PSDN-long	100	50	8.24	5.58	3.37

LAMPIRAN B HASIL PENGUJIAN LSTM

60	PSDN-long	100	128	11.63	9.64	6
61	PSDN-long	1000	10	8.42	5.98	3.57
62	PSDN-long	1000	50	7.89	5.22	3.16
63	PSDN-long	1000	128	7.97	5.43	3.33
64	PSDN-mid	10	10	39.59	26.24	8.75
65	PSDN-mid	10	50	34.01	22.29	7.49
66	PSDN-mid	10	128	33.73	21.84	7.32
67	PSDN-mid	100	10	30.78	19.83	6.62
68	PSDN-mid	100	50	31.17	18.91	6.13
69	PSDN-mid	100	128	31	18.76	6.1
70	PSDN-mid	1000	10	57.29	35.18	10.43
71	PSDN-mid	1000	50	54.42	33.47	9.95
72	PSDN-mid	1000	128	51.86	31.44	9.35
73	PSDN-short	10	10	14.63	13.02	8.18
74	PSDN-short	10	50	6.75	4.02	2.7
75	PSDN-short	10	128	8.95	6.66	4.41
76	PSDN-short	100	10	6.89	4.25	2.83
77	PSDN-short	100	50	7.12	4.55	3.01
78	PSDN-short	100	128	6.4	3.75	2.49
79	PSDN-short	1000	10	6.54	4.22	2.8

LAMPIRAN B HASIL PENGUJIAN LSTM

80	PSDN-short	1000	50	6.84	4.57	3.03
81	PSDN-short	1000	128	7.41	5.24	3.46