

**Perbandingan Penerapan Metode *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
2021**

**Perbandingan Penerapan Metode *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**



**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
2021**

DAFTAR ISI

| | |
|---|------------|
| ABSTRAK | iv |
| ABSTRACT | v |
| KATA PENGANTAR | vi |
| DAFTAR ISI | vii |
| DAFTAR TABEL | ix |
| DAFTAR GAMBAR | x |
| DAFTAR ALGORITMA | x |
| DAFTAR LAMPIRAN | xi |
| 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 |
| 2.1.2 <i>Autoregressive Integrated Moving Average (ARIMA)</i> | 2-3 |
| 2.1.3 <i>Artificial Neural Network (ANN)</i> | 2-5 |
| 2.1.4 <i>Recurrent Neural Network (RNN)</i> | 2-6 |
| 2.1.5 Fungsi Aktivasi | 2-8 |
| 2.1.6 <i>Long Short Term Memory</i> | 2-10 |
| 2.1.6.1 <i>Forward Propagation LSTM</i> | 2-10 |
| 2.1.6.2 <i>Backward Propagation LSTM</i> | 2-14 |

| | | |
|-------------------------|---|------------|
| 2.1.7 | Evaluasi <i>Forecasting</i> | 2-18 |
| 2.1.7.1 | <i>Root Mean Square Error</i> (RMSE) | 2-18 |
| 2.1.7.2 | <i>Mean Absolute Error</i> (MAE) | 2-18 |
| 2.1.7.3 | <i>Mean Absolute Percentage Error</i> (MAPE) | 2-19 |
| 2.2 | Tinjauan Studi | 2-20 |
| 2.3 | Tinjauan Objek | 2-24 |
| 2.3.1 | <i>Time series Dataset</i> | 2-24 |
| 2.3.1.1 | <i>Trend</i> | 2-24 |
| 2.3.1.2 | <i>Seasonal</i> | 2-25 |
| 2.3.1.3 | <i>Cyclical</i> | 2-25 |
| 2.3.2 | <i>Forecasting Horizon</i> | 2-26 |
| 2.3.3 | Dataset Saham | 2-26 |
| 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 | 3-6 |
| 3.4.1 | Dataset | 3-6 |
| 3.4.2 | Normalisasi Data | 3-7 |
| 3.4.3 | <i>Splitting</i> Dataset untuk <i>Training</i> dan <i>Testing</i> | 3-7 |
| 3.4.4 | Perhitungan Manual | 3-8 |
| 3.4.4.1 | Perhitungan <i>Long Short Term Memory</i> | 3-9 |
| 3.4.4.2 | Perhitungan <i>Autoregressive Integrated Moving Average</i> | 3-17 |
| 3.4.4.3 | Perbandingan Evaluasi <i>Forecasting</i> | 3-21 |
| DAFTAR REFERENSI | | i |

DAFTAR TABEL

| | | |
|-----|--------------------------|-------|
| 2.1 | Tinjauan Studi | .2-20 |
|-----|--------------------------|-------|

DAFTAR GAMBAR

| | | |
|------|---|------|
| 2.1 | Candlestick | 2-2 |
| 2.2 | <i>Artificial Neural Network Structure</i> [22] | 2-6 |
| 2.3 | <i>Recurrent Neural Network Loop</i> | 2-6 |
| 2.4 | <i>Recurrent Neural Network</i> | 2-7 |
| 2.5 | <i>Sigmoid Function</i> [23] | 2-9 |
| 2.6 | <i>tanh function</i> | 2-10 |
| 2.7 | <i>Long Short Term Memory unit</i> | 2-11 |
| 2.8 | Downtrend | 2-25 |
| 2.9 | Seasonal Time series | 2-25 |
| 2.10 | Cyclical Time series [19] | 2-26 |
| 2.11 | Data Sampling [18] | 2-27 |
| 3.1 | Kerangka Pemikiran | 3-1 |
| 3.2 | Urutan Proses Global | 3-4 |
| 3.3 | <i>Flowchart proses training</i> | 3-5 |
| 3.4 | <i>Flowchart proses Testing</i> | 3-6 |
| 3.5 | Normalisasi Data <i>MinMaxScaler</i> | 3-7 |
| 3.6 | Splitting Dataset | 3-8 |
| 3.7 | Bidirectional LSTM | 3-9 |

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.

Penelitian lain oleh Hyeong Kyuu Choi melakukan prediksi harga saham dengan menggunakan gabungan dari metode ARIMA dan LSTM [9]. Dataset yang digunakan merupakan dataset dari indeks S&P 500 pada rentang waktu dari tahun 2008 sampai 2017. *Hybrid* model yang dibuat mendapatkan MSE, RMSE, MAE dengan hasil yang berturut-turut ialah 0.1786, 0.1889, 0.2154.

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 *Long Short Term Memory* (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. FKS Multi Agro Tbk(FISH), 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* dengan mengimplementasikan metode *Long Short Term Memory* dan *Autoregressive Integrated Moving Average* untuk memprediksi penutupan harga saham?
2. Berapa hasil prediksi harga saham dengan mengimplementasikan 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 *Long Short Term Memory* dan *Autoregressive Integrated Moving Average*.
2. Mengevaluasi hasil kinerja model yang dibangun oleh metode *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. FKS Multi Agro Tbk(FISH), PT. Prasadha Aneka Niaga(PSDN).
2. 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 LSTM dan ARIMA lalu menentukan nilai parameter 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

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. FKS Multi Agro Tbk(FISH), 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 pendahuluan yang terdiri dari latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, kontribusi penelitian, dan metodologi penelitian.

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 yang digunakan dalam penelitian ini.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian yang dilakukan menggunakan metode terkait beserta data testing dan 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

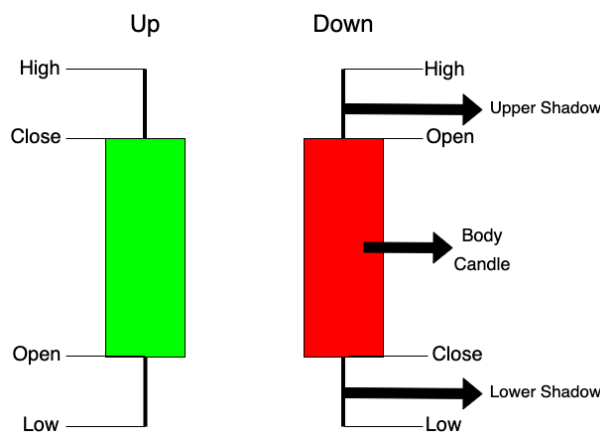
Untuk mendapatkan keuntungan tentunya diperlukan suatu analisis untuk memprediksi arah maupun harga saham. Secara garis besar analisis tersebut dapat digolongkan menjadi dua, yaitu analisis fundamental dan analisis teknikal. Analisis fundamental dalam memperkirakan harga saham di masa yang akan datang 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. *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

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 [13]. 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 [13]. Bila investor melihat MA bergerak turun ke bawah semenrara harganya juga bergerak dibawah 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*.

2.1.2 *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.

$$Y_t = c + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} + e_t \quad (2.1)$$

Keterangan :

- Y_t : nilai yang diamati saat t
 β_p : *coefficient* untuk AR
 ϵ_t : nilai rata-rata dari periode tersebut pada t
 p : nilai *input* dari ordo p
 q : nilai *input* dari ordo q
 e_t : nilai *error white noise*

1. Autoregressive (AR)

Model *autoregressive* merupakan model dari *multiple regression* dengan nilai *lagged* dari y_t sebagai prediktor [19]. 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*
 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 : parameter model komponen MA(q) (dimulai dengan lag terendah)

ϵ_t : nilai rata-rata dari periode tersebut pada t

q : Nilai input dari ordo q

2.1.3 *Artificial Neural Network* (ANN)

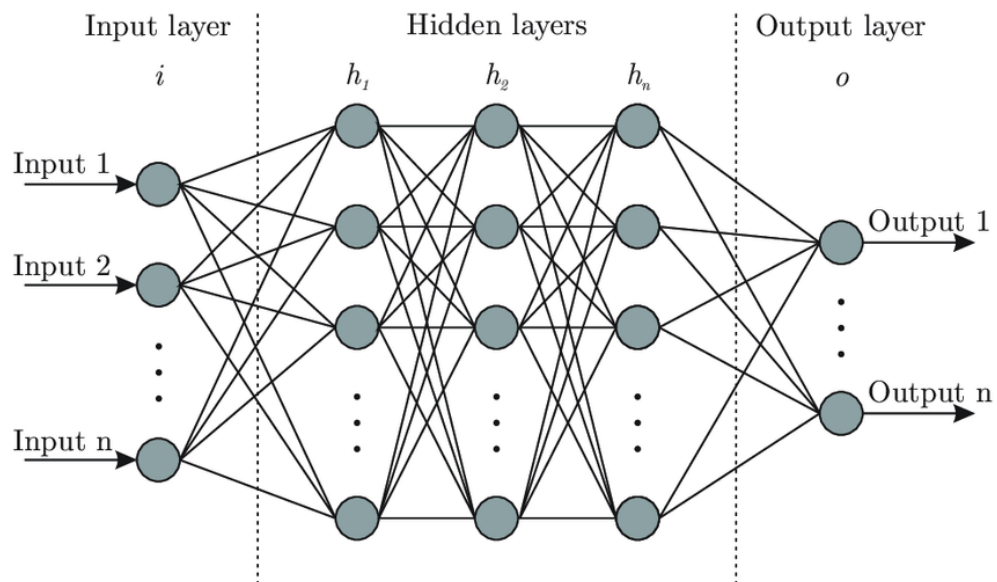
Artificial Neural Network (ANN) adalah kecerdasan buatan yang mengadopsi kinerja otak manusia seperti memberi stimulus atau rangsangan, melakukan proses dan memberikan *output*. *Neural Network* mempunyai tiga lapisan layer yang disebut *input layer*, *hidden layer*, dan *output layer*. Jumlah fitur 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.

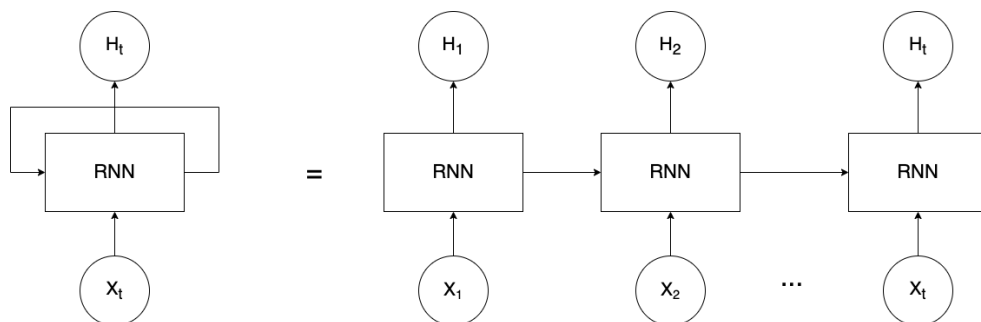
2. Di *hidden layer*, node menerapkan *activation function* (seperti, *sigmoid* dan *tanh*) pada jumlah *input* yang nanti diproses dan menjadi nilai prediksi.
3. *Output layer* menghasilkan vektor probabilitas untuk berbagai *output* dan mendapatkan nilai *error rate*



Gambar 2.2 Artificial Neural Network Structure [22]

2.1.4 Recurrent Neural Network (RNN)

RNN adalah salah satu kelas dari ANN dimana koneksi antar *node* membentuk sebuah grafik yang bersambung [14]. 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*

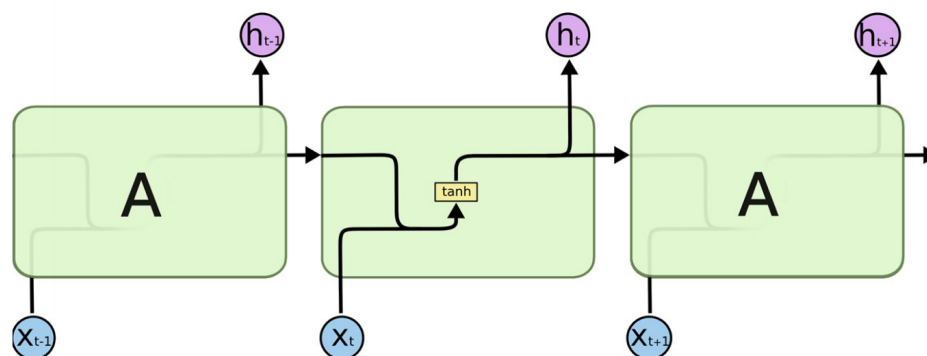
Keterangan :

X_t : nilai *input* pada *timestep* t

H_t : nilai *output* pada *timestep* t

Gambar 2.3 menunjukkan sebuah proses unit 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* [16]. Dalam Gambar 2.3, dijelaskan:

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



Gambar 2.4 *Recurrent Neural Network*

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

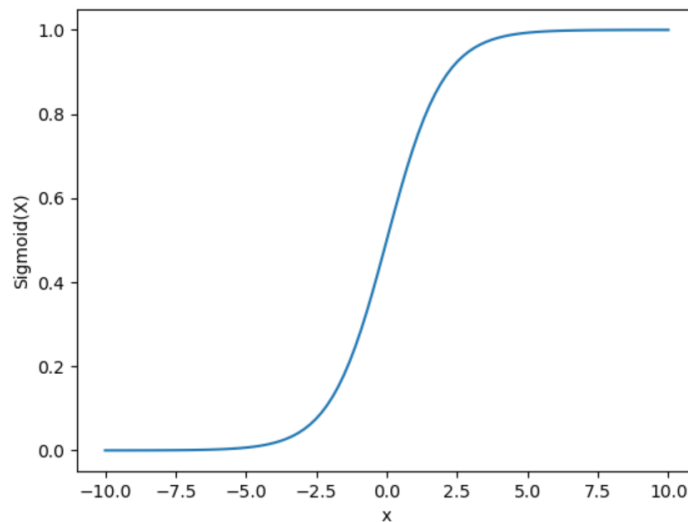
Sigmoid function merupakan fungsi aktivasi yang menghasilkan nilai dengan

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 [23]

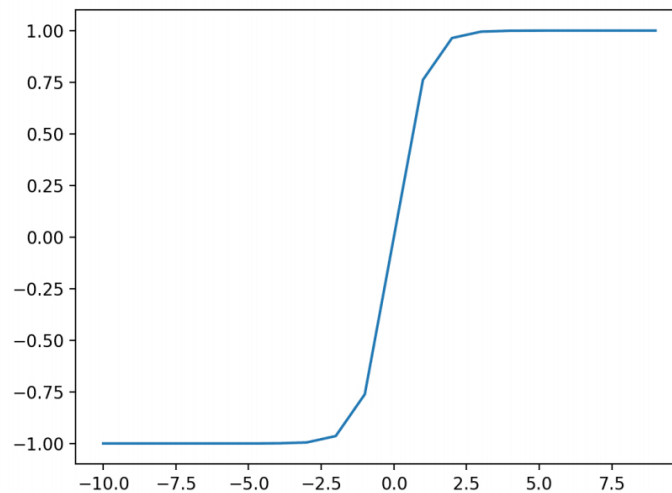
2. Hyperbolic Tangent Function (\tanh)

Hyperbolic Tangent Function merupakan fungsi aktivasi yang menghasilkan nilai dengan rentang -1 sampai dengan 1. Fungsi aktivasi ini bersifat non-linear dan kelebihan 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*



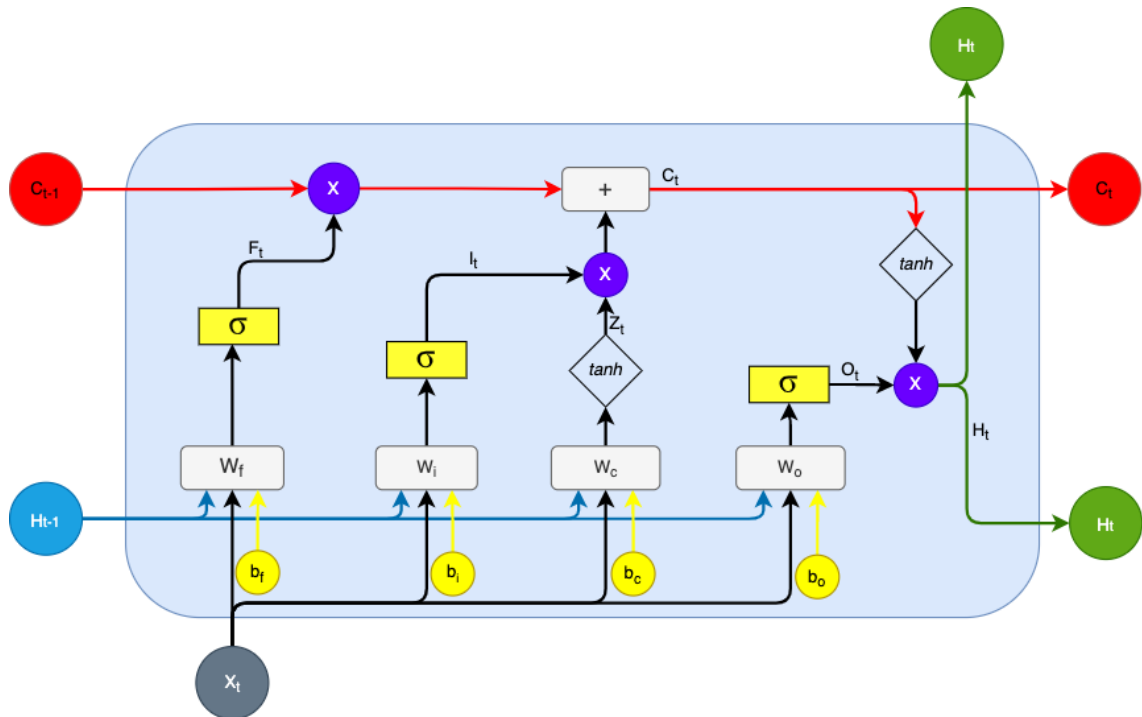
Gambar 2.6 *tanh function*

2.1.6 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 unit memori yang disebut *cell* [6]. Arsitektur yang dimiliki oleh LSTM memungkinkan informasi dari memori sebuah unit diteruskan tidak hanya pada unit berikutnya, melainkan dapat terus memberikan kontribusi pada unit-unit 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 *recurrent connection* dengan bobot tetap yang bernilai 1.0, yang dapat diatur ulang oleh *forget gate* [24].

2.1.6.1 Forward Propagation LSTM

Forward propagation 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 2.7 Long Short Term Memory unit

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 2.7 menjelaskan setiap unit pada jaringan LSTM menerima 2 *input*, yaitu X_t unit *input* dan H_{t-1} *output* dari unit sebelumnya. Langkah pertama, LSTM akan memutuskan seberapa penting sebuah informasi untuk disimpan. Langkah ini dilakukan oleh *layer sigmoid* "forget gate" dengan Persamaan 2.10 berikut:

$$F_t = \sigma(W_f \cdot [H_{t-1}, X_t] + b_f) \quad (2.10)$$

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*
 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].

$$I_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i) \quad (2.11)$$

$$Z_t = \tanh(W_c \cdot [H_{t-1}, X_t] + b_c) \quad (2.12)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot Z_t \quad (2.13)$$

Keterangan :

| | |
|-----------|---|
| I_t | : nilai aktivasi <i>input gate</i> |
| W_i | : bobot <i>input gate</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> |
| 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 unit yang terbagi menjadi 3 tahap. Pertama, *input gate* akan menentukan berapa besar informasi dengan melakukan fungsi aktivasi *sigmoid* terhadap *input* dari unit yang akan dijadikan memori dengan menggunakan Persamaan 2.11. Tahap kedua, *memory gate* akan melakukan fungsi aktivasi *hyperbolic tangent* terhadap *input* unit untuk menghasilkan nilai memori yang akan digunakan dengan Persamaan 2.12.

Tahap terakhir adalah untuk menghasilkan nilai memori baru yang diperoleh dari penjumlahan dari hasil perkalian *forget gate* (hasil dari Persamaan 2.10) terhadap nilai memori unit sebelumnya (hasil dari Persamaan 2.11) dengan hasil perkalian *input gate* (hasil dari Persamaan 2.12) dan *cell state* menghasilkan nilai memori unit saat ini yang akan digunakan untuk perhitungan selanjutnya pada Persamaan 2.13.

$$O_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o) \quad (2.14)$$

$$H_t = O_t \cdot \tanh(C_t) \quad (2.15)$$

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_f | : bobot <i>output gate</i> |
| \tanh | : fungsi aktivasi <i>hyperbolic tangent</i> |
| σ | : fungsi aktivasi <i>sigmoid</i> |

Langkah terakhir adalah menghasilkan *output* H_t untuk unit 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 unit saat ini dan *output* dari unit sebelumnya (Persamaan 2.14). Kemudian nilai dari *output gate* akan dikalikan dengan hasil fungsi aktivasi *hyperbolic tangent* terhadap nilai memori sesuai Persamaan 2.15, menghasilkan *output* untuk unit saat ini.

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

2.1.6.2 Backward Propagation LSTM

Backward propagation merupakan proses belajar dalam perhitungan LSTM yang alurnya mundur. Menyerupai rumus dari *Forward propagation*, *Backward propagation* mempunyai *gate* dan perhitungan yang hampir sama dengan *forward*. *Backward propagation* ada untuk meningkatkan akurasi prediksi dalam *machine learning*. masing-masing dari perhitungan pada *Backward Propagation* akan menghasilkan nilai perubahan bobot yang terlibat di setiap perhitungan.

$$\Delta Y^t = \Delta E + W_{hc} \cdot \Delta z^{t+1} + W_{hi} \cdot \Delta i^{t+1} + W_{hf} \cdot \Delta f^{t+1} + W_{ho} \cdot \Delta o^{t+1} \quad (2.16)$$

Keterangan :

| | |
|------------------|--|
| ΔY^t | : selisih nilai dari <i>output</i> pada unit LSTM yang sedang berjalan |
| ΔE | : selisih nilai <i>output</i> unit LSTM dan data asli |
| W_{hc} | : nilai bobot <i>memory gate</i> |
| Δz^{t+1} | : selisih nilai <i>memory</i> dari unit LSTM sebelumnya |
| W_{hi} | : nilai <i>weight input gate</i> |
| Δi^{t+1} | : selisih nilai <i>input</i> dari unit LSTM sebelumnya |
| W_{hf} | : nilai <i>weight forget gate</i> |
| Δf^{t+1} | : selisih nilai <i>forget</i> dari unit LSTM sebelumnya |
| W_{ho} | : nilai <i>weight output gate</i> |
| Δo^{t+1} | : selisih nilai <i>output</i> dari unit LSTM sebelumnya |

Pada Persamaan 2.16, merupakan tahap pertama yang berfungsi menghitung selisih nilai dari output pada unit LSTM yang sedang berjalan dengan menggunakan selisih nilai *output* unit LSTM dan data asli, selisih nilai *memory* dari unit sebelumnya dikalikan bobotnya, selisih nilai input dari unit sebelumnya dikalikan bobotnya, selisih nilai *forget* dari unit sebelumnya dikalikan bobotnya, selisih nilai *output* dari unit sebelumnya. Selisih nilai *output* pada unit LSTM yang sedang berjalan ini akan digunakan dalam perhitungan selanjutnya.

$$\Delta C^t = \Delta Y^t \cdot O^t \cdot (1 - \tanh^2(c^t)) + \Delta C^{t+1} \cdot f^{t+1} \quad (2.17)$$

Keterangan :

| | |
|----------------------|--|
| ΔC^t | : selisih nilai dari <i>Cell state</i> pada unit LSTM yang sedang berjalan |
| ΔY^t | : selisih nilai <i>output</i> pada unit LSTM yang sedang berjalan |
| O^t | : <i>output gate</i> pada unit LSTM yang sedang berjalan |
| $(1 - \tanh^2(c^t))$ | : turunan dari nilai <i>tanh</i> terhadap <i>cell state</i> |
| ΔC^{t+1} | : selisih nilai dari <i>cell state</i> pada unit LSTM sebelumnya |
| f^{t+1} | : nilai <i>forget</i> dari unit LSTM sebelumnya |

Pada Persamaan 2.17, merupakan tahap kedua yang berfungsi menghitung selisih nilai dari *cell state* pada unit LSTM yang sedang berjalan yang nantinya hasil dari selisih nilai *cell state* akan digunakan untuk perhitungan selanjutnya. Di dalam persamaan, terdapat nilai turunan dari *tanh* terhadap *cell state* yang digunakan untuk perkalian dengan selisih nilai *output* dan nilai *output gate*.

$$\Delta F^t = \Delta C^t \cdot C^{t-1} \cdot (1 - \sigma^2(f^t)) \quad (2.18)$$

Keterangan :

ΔF^t : selisih nilai dari *Forget gate* pada unit LSTM yang sedang berjalan

ΔC^t : selisih nilai *Cell state* pada unit LSTM yang sedang berjalan

C^{t-1} : *Cell State* pada unit LSTM sebelumnya

$(1 - \sigma^2(f^t))$: turunan dari nilai fungsi aktivasi *sigmoid* terhadap *forget gate*

Pada Persamaan 2.18, merupakan tahap ketiga yang berfungsi menghitung selisih nilai dari *forget gate* pada unit LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari *sigmoid* terhadap *forget gate* yang digunakan untuk perkalian dengan selisih nilai *cell state* yang didapatkan pada Persamaan 2.17 dan nilai *cell state* pada unit sebelumnya.

$$\Delta I^t = \Delta C^t \cdot z^t \cdot (1 - \sigma^2(i^t)) \quad (2.19)$$

$$\Delta Z^t = \Delta C^t \cdot i^t \cdot (1 - \tanh^2(z^t)) \quad (2.20)$$

Keterangan :

ΔI^t : selisih nilai dari *input gate* pada unit LSTM yang sedang berjalan

ΔZ^t : selisih nilai dari *memory gate* pada unit LSTM yang sedang berjalan

ΔC^t : selisih nilai *Cell state* pada unit LSTM yang sedang berjalan

z^t : nilai *memory gate* pada unit LSTM sedang berjalan

i^t : nilai *input gate* pada unit LSTM sedang berjalan

$(1 - \sigma^2(i^t))$: turunan dari nilai fungsi aktivasi *sigmoid* terhadap *input gate*

$(1 - \tanh^2(z^t))$: turunan dari nilai fungsi aktivasi *tanh* terhadap *memory gate*

Pada Persamaan 2.19, merupakan tahap keempat yang berfungsi menghitung selisih nilai dari *input gate* pada unit LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari *sigmoid* terhadap *input gate* yang digunakan untuk perkalian dengan selisih nilai *cell state* dan nilai *memory gate*.

Pada Persamaan 2.20, merupakan tahap kelima yang berfungsi menghitung selisih nilai dari *memory gate* pada unit LSTM yang sedang berjalan. Di dalam persamaan, terdapat nilai turunan dari fungsi aktivasi *tanh* terhadap *memory gate* yang digunakan untuk perkalian dengan selisih 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 selisih *gate* yang dikalikan terhadap nilai *input* dan *output* dari unit sebelumnya. Berikut merupakan persamaan untuk mendapatkan perubahan nilai (*update*) bobot dan bias yang digunakan dalam perhitungan LSTM.

$$\Delta W_{xgate} = \sum_{t=0}^T (\Delta gate^{-t} x^t) \quad (2.21)$$

$$\Delta W_{hgate} = \sum_{t=0}^{T-1} (\Delta gate^{-t} h^t) \quad (2.22)$$

$$\Delta b_{gate} = \sum_{t=0}^T (\Delta gate^{-t}) \quad (2.23)$$

Keterangan :

- gate* : substitusi dari simbol $gate(f,i,c,o)$
- ΔW_{xgate} : selisih nilai bobot dari *input* untuk perhitungan pada setiap *gate*
- ΔW_{hgate} : selisih nilai bobot dari *output* untuk perhitungan *gate*
- Δb_{gate} : selisih nilai bias
- $\Delta gate^{-t}$: selisih nilai dari *gate* untuk unit LSTM yang sedang diproses
- x^t : nilai *input* pada unit LSTM sedang berjalan
- h^t : nilai *output* dari unit LSTM yang sedang diproses

Persamaan 2.21 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk perhitungan sebagai *input* yang dibutuhkan oleh masing-masing *gate* unit LSTM yang sedang berjalan. Persamaan 2.22 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk perhitungan sebagai nilai *output* unit LSTM yang sedang berjalan. Persamaan 2.23 merupakan persamaan untuk menghitung selisih nilai bobot yang digunakan untuk

perhitungan nilai bias yang dibutuhkan oleh masing-masing *gate*. Dengan persamaan tersebut, maka diperoleh bobot dan bias yang baru untuk iterasi selanjutnya.

2.1.7 Evaluasi *Forecasting*

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

2.1.7.1 *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 2.24 berikut.

$$RMSE = \sqrt{\sum_{i=1}^N (y_i - y_p)^2 / N} \quad (2.24)$$

Keterangan :

Σ : penjumlahan
 y_i : nilai aktual
 y_p : nilai prediksi
 N : jumlah data

2.1.7.2 *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 2.25 berikut.

$$MAE = \frac{\sum_{i=1}^N |y_i - y_p|}{N} \quad (2.25)$$

Keterangan :

Σ : penjumlahan

y_i : nilai aktual

y_p : nilai prediksi

N : jumlah data

2.1.7.3 Mean Absolute Percentage Error (MAPE)

MAPE merupakan teknik pengukuran seberapa akurat sistem *forecasting* yang sudah dibuat [15]. 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 2.26 berikut.

$$MAPE = 100\% * \frac{1}{N} * \sum_{i=1}^N \frac{|y_i - y_p|}{y_i} \quad (2.26)$$

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 prakiraan. 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.

2.2 Tinjauan Studi

Pada Tabel 2.1 diberikan penjelasan perbandingan dari berbagai penelitian terkait dengan prediksi harga saham.

Tabel 2.1 Tinjauan Studi

| No | Judul | Rumusan Masalah | Metode | Hasil |
|----|-------|--------------------|--------|-------|
|----|-------|--------------------|--------|-------|

| | | | | |
|---|--|--|--|---|
| 1 | <p>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.</p> <p>[4]</p> | <p>Bagaimana hasil dari menggunakan metode LSTM, SVM, <i>Random Forest</i>, ANN?</p> | <p>Membandingkan hasil dari metode LSTM, SVM, <i>Random Forest</i>, ANN.</p> | <p>Membandingkan hasil prediksi dari model yang sudah dibangun dan hasilnya model yang dibangun dengan 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.</p> |
|---|--|--|--|---|

| | | | | |
|---|--|---|--|---|
| 2 | <p>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]</p> | <p>Berapakah nilai parameter ARIMA(p,d,q) untuk mencari RMSE yang terendah pada prediksi harga saham?</p> | <p><i>Autoregressive Integrated Moving Average</i></p> | <p>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).</p> |
|---|--|---|--|---|

| | | | | |
|---|--|---|-----------------------------|--|
| 4 | <p>F. R. Pratama, "Implementasi Algoritme Random Forest untuk Prediksi Pergerakan Harga Saham," Dept. Informatika ITHB, 2017. [12]</p> | <p>Bagaimana pengaruh setiap indikator terhadap hasil uji dan pengaruh jumlah pohon keputusan yang dibuat terhadap hasil uji?</p> | <p><i>Random Forest</i></p> | <p>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.</p> |
|---|--|---|-----------------------------|--|

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 [19]. *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*

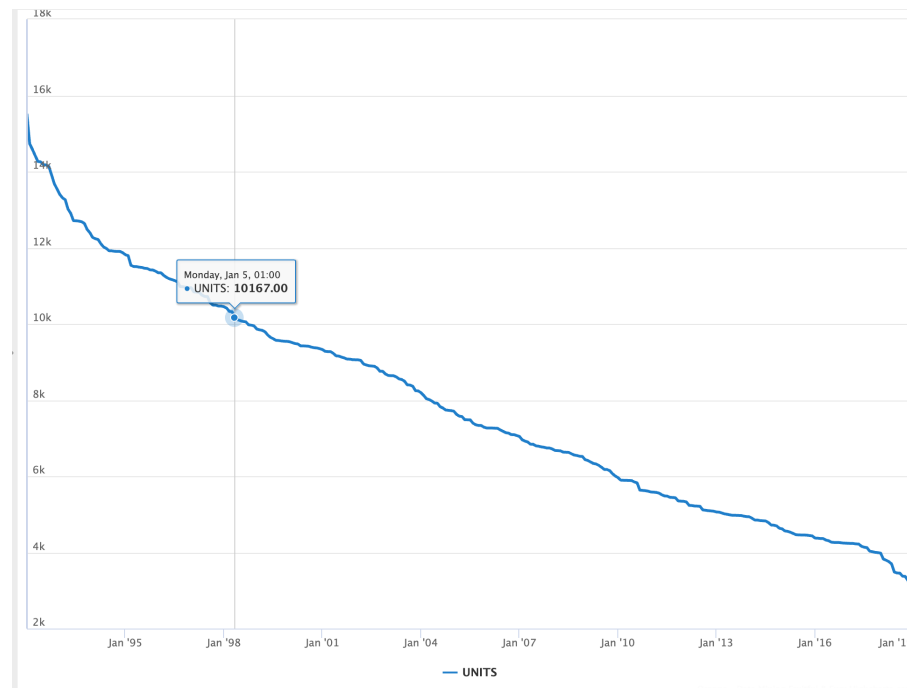
Trend dalam pola dalam data yang menunjukkan pergerakan suatu rangkaian ke nilai yang relatif lebih tinggi atau lebih rendah dalam jangka waktu yang lama. Dengan kata lain, *trend* diamati ketika ada peningkatan atau penurunan dalam *time series* [19]. *Trend* biasanya terjadi untuk beberapa waktu dan 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*

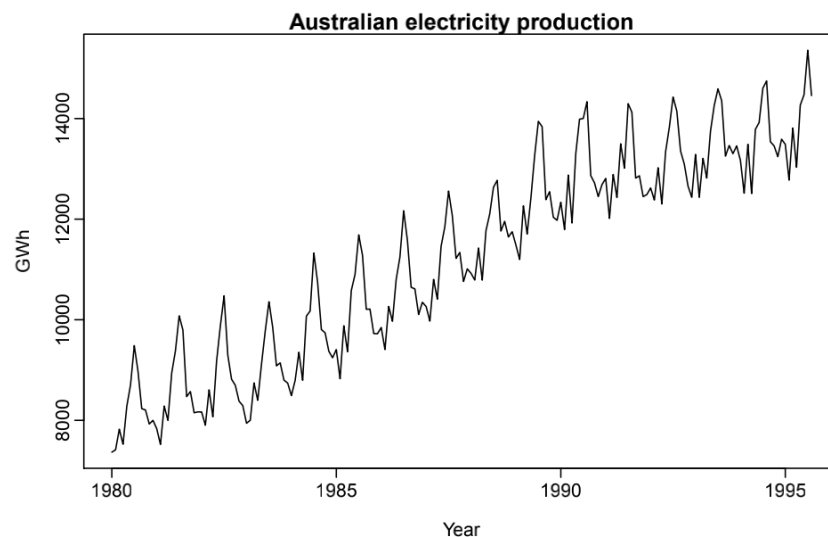
Analisis *time series* menunjukkan pola yang turun maka itu adalah *Downtrend*, seperti yang ada pada Gambar 2.8.



Gambar 2.8 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 2.9 [19].

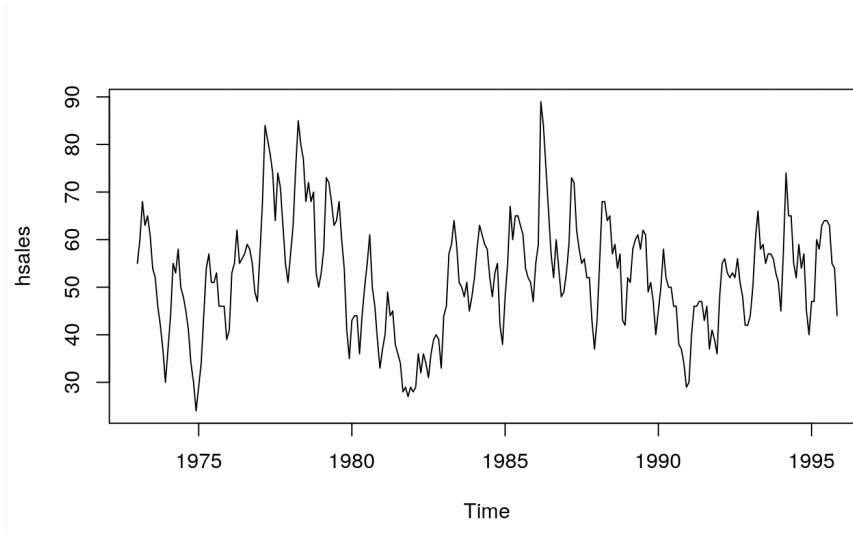


Gambar 2.9 Seasonal Time series

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 2.10 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 2.10 Cyclical Time series [19]

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 secara global merupakan prediksi yang diklasifikasikan menurut periode waktu dan digolongkan menjadi *short-term*, *medium-term*, dan *long-term* [20]. *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. FKS Multi Agro Tbk(FISH), 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 2.11.

| 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 2.11 Data Sampling [18]

Dalam investasi, terdapat 2 macam saham jenis yaitu:

1. Saham *Blue Chip*

Istilah *blue chip* dikarenakan koin berwarna biru pada permainan *poker* dan kasino yang memiliki nilai yang paling tinggi [21]. 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 [21]. 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).

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

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

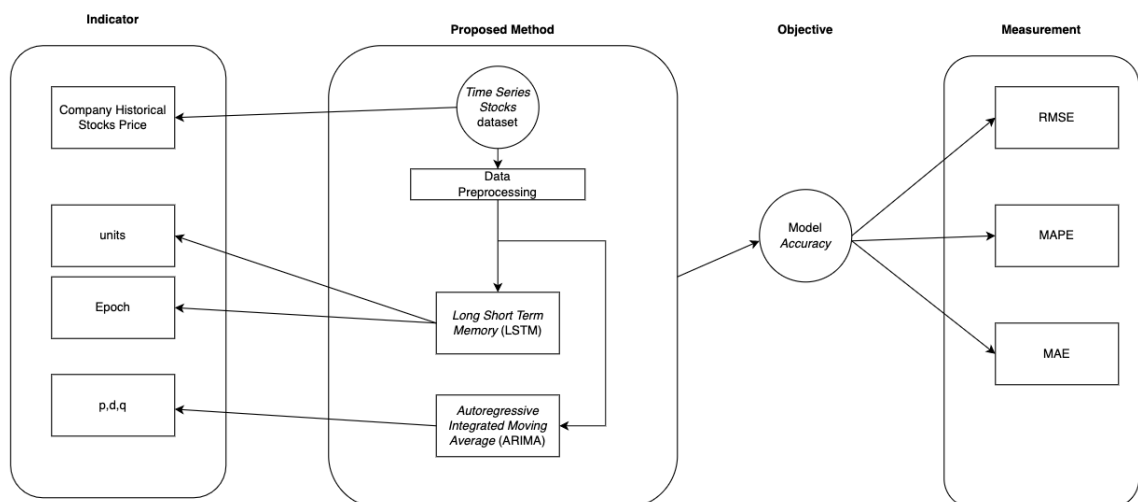
3.1 Analisis Masalah

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 *Long Short Term Memory* (LSTM) dan model *Autoregressive Integrated Moving Average* (ARIMA). *Input* pada sistem ini adalah variabel-variabel dari dataset yang sudah ditentukan, seperti *Date*, *Open*, *High*, *Low*, *Close*, *Adj Close*. *Output* pada sistem ini berupa harga dari hasil prediksi dari setiap variabel 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. FKS Multi Agro Tbk (FISH), PT. Prsidha 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.
- ii. *Mid-term forecasting* menggunakan 1 tahun dari keseluruhan data [2]. Rentang waktu yang digunakan dari 1 Januari 2017 sampai 31 Desember 2017.
- iii. *Long-term forecasting* menggunakan 5 tahun dari keseluruhan data [6]. Rentang waktu yang digunakan 1 Januari 2017 sampai 31 Desember 2021.

(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 1, 2, 10. Karena pada penelitian [4, 5, 6] 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*.

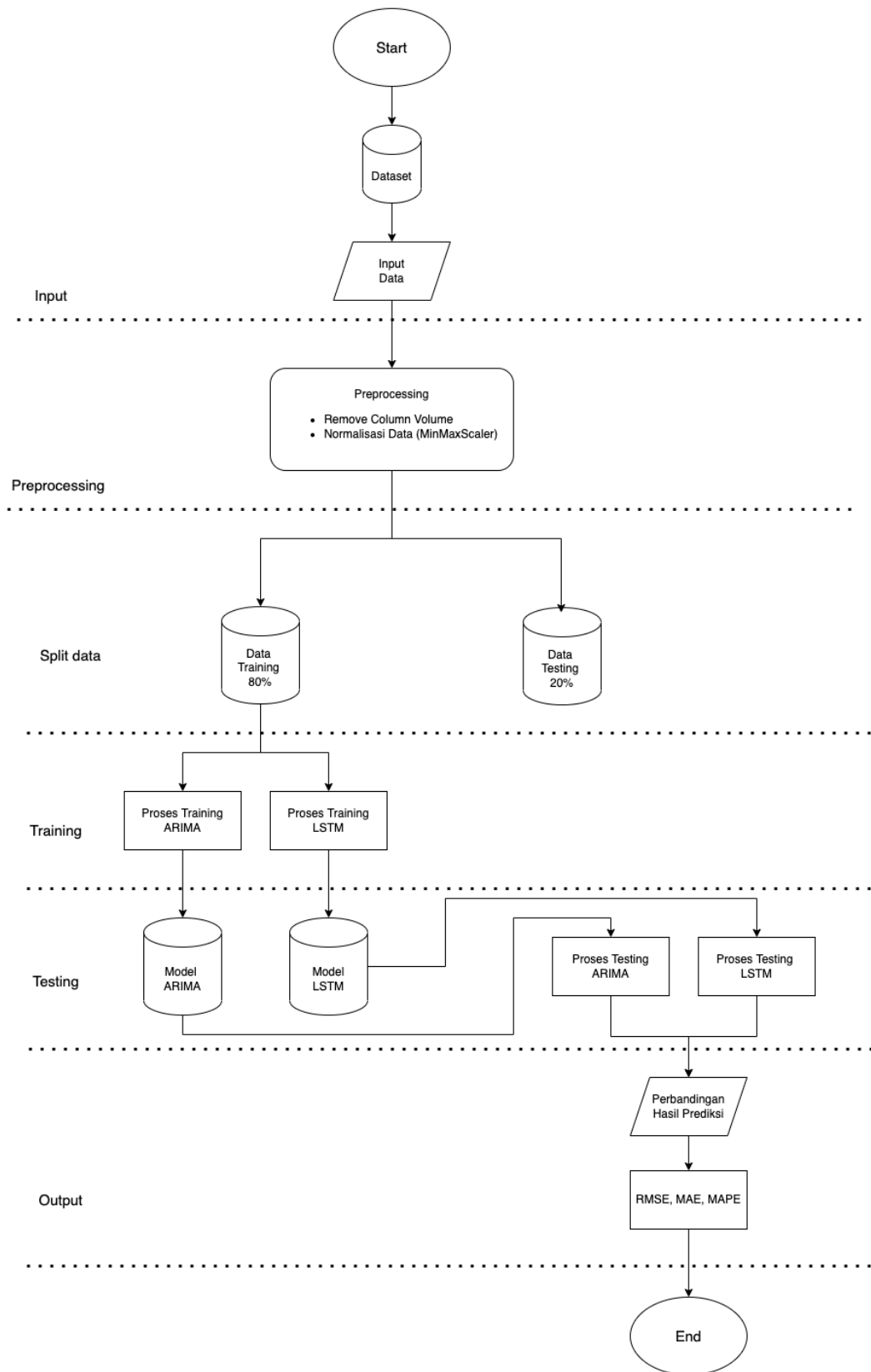
(d) p,d,q

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 3.

2. *Proposed Method* adalah bagian yang menjelaskan proses peneltitan 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 *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

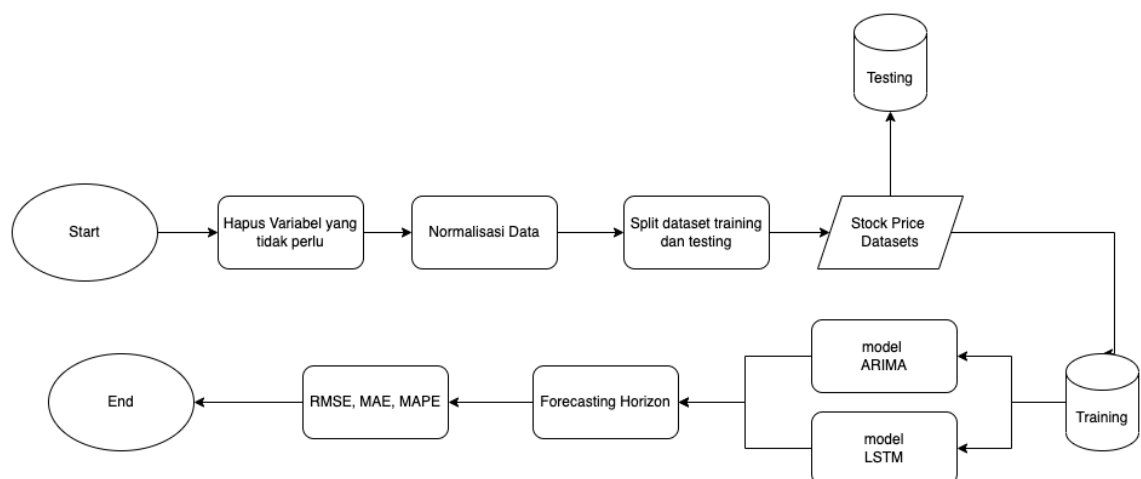
Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang

terlihat pada Gambar 3.2:

Model prediksi harga saham ini dibangun menggunakan metode ARIMA dan LSTM dengan bertujuan membandingkan hasil model prediksi dari kedua metode tersebut. Setelah dilakukan *training*, model diharapkan dapat memprediksi harga saham dalam waktu jangka pendek, jangka menengah atau jangka panjang. Seperti pada Gambar 3.2 dimulai dari *preprocessing* dataset, lalu melakukan *splitting* data untuk melakukan *training* dan *testing* untuk membuat model. Setelah model selesai dibuat, maka akan dilakukan proses *testing* dengan data yang bersumber dari data *testing*, maka sistem akan menghasilkan prediksi, dan performa yang akan di evaluasi 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*

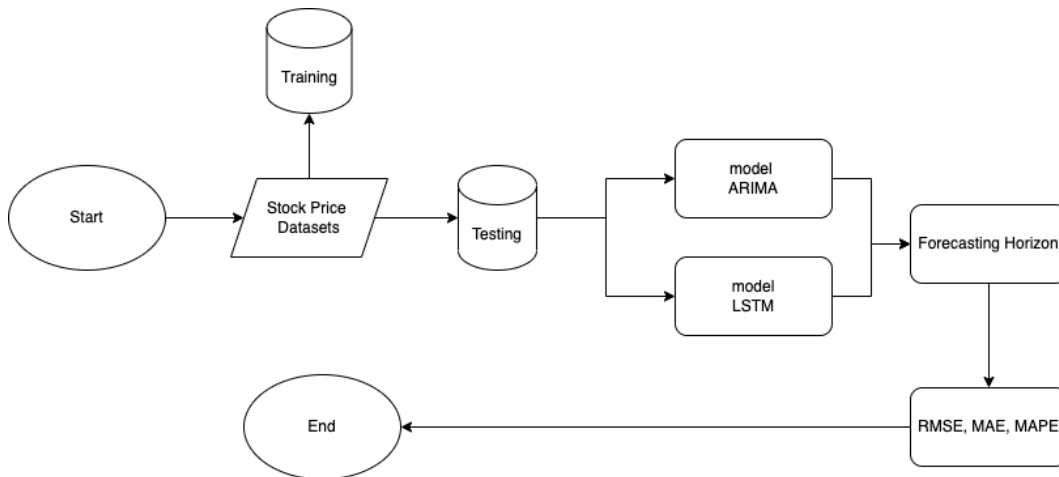
Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang terlihat pada Gambar 3.3:

1. Dataset yang digunakan terdiri dari 246 data yang memiliki 7 variabel, yaitu *Date*, *Open*, *High*, *low close*, *Adj close*, *Volume*.
2. Selanjutnya, akan dilakukan penghapusan terhadap variabel *volume*.
3. Melakukan Normalisasi data dengan menggunakan *MinMaxScaler*.
4. Membagi dataset sebesar 80% untuk proses training metode ARIMA dan LSTM.
5. Dan akan menghasilkan 2 model, yaitu model ARIMA dan model LSTM.

6. Terakhir, akan diukur performa dari masing-masing model dengan menggunakan RMSE, MAE, MAPE.

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.
2. Selanjutnya, akan dilakukan *testing* terhadap data yang sudah menghasilkan model ARIMA dan model LSTM.
3. Terakhir, akan melakukan prediksi *forecasting horizon* dan diukur performa dari masing-masing model dengan menggunakan RMSE, MAE, MAPE.

3.4 Analisis Manual

Pada bagian ini akan dijelaskan analisis tahapan proses yang dilakukan dalam sistem.

3.4.1 Dataset

Dalam penelitian ini, dataset yang digunakan berjumlah 4 macam dataset historis harga saham yang diambil sejak 13 Februari 2021 sampai 13 Februari 2022. 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. FKS Multi Agro

Tbk(FISH), PT. Prasadha Aneka Niaga(PSDN). Dataset yang digunakan dalam penelitian ini semua dengan format *Comma Separated Value* (CSV). Data yang akan dipakai dari dataset ini sejumlah 246 data dan memiliki 7 variabel, tetapi variabel *Volume* tidak digunakan dan akan dihapus.

3.4.2 Normalisasi Data

Pada tahap ini, data akan dilakukan proses normalisasi dengan menggunakan *MinMaxScaler*. Normalisasi data diperlukan, karena LSTM sangat sensitif terhadap data *input* yang relatif besar. *MinMaxScaler* akan mengubah nilai rentang data antara 0 sampai 1. Gambar 3.5 merupakan kode untuk melakukan normalisasi terhadap data menggunakan *MinMaxScaler* pada python.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))
```

```
print(df1)
```

```
[[0.  ]
 [0.12]
 [0.2  ]
 [0.24]
 [0.52]
 [0.44]
 [0.6  ]
 [0.6  ]
 [0.6  ]
 [0.84]
 [0.68]
 [0.56]
 [0.56]
 [0.72]
 [1.  ]]
```

Gambar 3.5 Normalisasi Data *MinMaxScaler*

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 LSTM. Pembagian dataset adalah 80% untuk data *training* dan 20% untuk data *testing*. Gambar 3.6 merupakan kode Python untuk melakukan proses *splitting* dataset.

```
n = int(len(df) * 0.8)
train = df.Close[:n]
test = df.Close[n:]
```

```
print(len(train))
print(len(test))
```

```
12
3
```

```
train
```

```
0    7325.0
1    7400.0
2    7362.5
3    7425.0
4    7462.5
5    7562.5
6    7625.0
7    7650.0
8    7700.0
9    7700.0
10   7775.0
11   7800.0
```

```
Name: Close, dtype: float64
```

```
test
```

```
12    7712.5
13    7675.0
14    7725.0
```

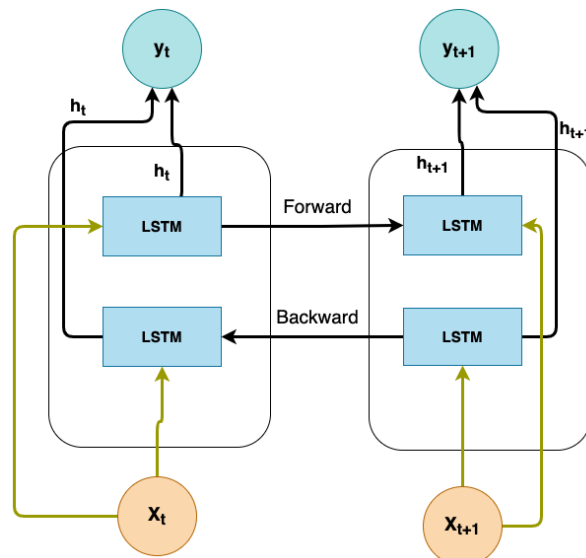
```
Name: Close, dtype: float64
```

Gambar 3.6 Splitting Dataset

3.4.4 Perhitungan Manual

Perhitungan manual akan menggunakan 12 hari data harga saham BBCA dengan memprediksi harga saham BBCA hari ke-13 sampai hari ke-15 dengan menggunakan metode LSTM dan ARIMA, yang nanti hasil prediksi hari ke-13 sampai hari ke-15 akan dilakukan evaluasi *forecasting* untuk mendapatkan nilai RMSE.

3.4.4.1 Perhitungan *Long Short Term Memory*



Gambar 3.7 Bidirectional LSTM

Pada Gambar 3.7 merupakan contoh arsitektur pada LSTM dengan menggunakan 2 arah (*forward*, *backward*) dengan jumlah 2 unit LSTM. Dalam *bidirectional*, terdapat 2 *input* dari 2 arah (*Forward*, *Backward*). *Forward* adalah input dari belakang dan melakukan proses ke arah depan(alur maju). Sementara, *backward* adalah *input* dari depan dan melakukan proses ke arah belakang(alur mundur).

Pada contoh perhitungan LSTM, akan dilakukan sekali proses epoch pada 2 unit LSTM. *Input* X_t merupakan harga yang sudah dinormalisasi dan diinisialisasi pada setiap *timestep* yang dibutuhkan untuk menghasilkan *output* H_t . Nilai bobot dan bias akan diinisialisasi secara acak.

Inisialisasi dan Normalisasi Data:

| Date | Close Price | Hasil Normalisasi |
|------------|-------------|-------------------|
| 03/01/2022 | 7325 | 0 |
| 04/01/2022 | 7400 | 0.12 |
| 05/01/2022 | 7450 | 0.2 |
| 06/01/2022 | 7475 | 0.24 |
| 07/01/2022 | 7650 | 0.52 |
| 10/01/2022 | 7600 | 0.44 |
| 11/01/2022 | 7700 | 0.6 |
| 12/01/2022 | 7700 | 0.6 |
| 13/01/2022 | 7700 | 0.6 |
| 14/01/2022 | 7850 | 0.84 |
| 17/01/2022 | 7750 | 0.68 |
| 18/01/2022 | 7675 | 0.56 |
| 19/01/2022 | P_{13} | 0.56 |
| 20/01/2022 | P_{14} | 0.72 |
| 21/01/2022 | P_{15} | 1 |

untuk memprediksi harga P_{13} ,

maka: $X_t = [0, 0.12, 0.2, 0.24, 0.52, 0.44, 0.6, 0.6, 0.6, 0.84, 0.68, 0.56]$

untuk memprediksi harga P_{14} ,

maka: $X_t = [0.12, 0.2, 0.24, 0.52, 0.44, 0.6, 0.6, 0.6, 0.84, 0.68, 0.56, 0.56]$

untuk memprediksi harga P_{15} ,

maka: $X_t = [0.2, 0.24, 0.52, 0.44, 0.6, 0.6, 0.6, 0.84, 0.68, 0.56, 0.56, 0.72]$

Rentang *weight* = -0.4082 sampai dengan 0.4082

Maka, nilai *weight* yang didapatkan adalah:

$w_f = -0.0862$

$w_i = -0.2373$

$w_c = 0.2134$

$w_o = 0.0356$

$b_f = 0$

$b_i = 0$

$b_c = 0$

$b_o = 0$

Berikut adalah inisialisasi data yang sudah disiapkan untuk melakukan perhitungan LSTM dan ARIMA. Terdapat 12 data penutupan harga saham yang akan di input pada perhitungan LSTM sebagai X_t yang sudah dinormalisasi dengan menggunakan *MinMaxScaler* dan ingin memprediksi 3 data harga saham dengan variabel P_{13} , P_{14} , P_{15} . Nilai bobot dan bias dengan notasi f yang merupakan nilai aktivasi yang akan digunakan oleh *forget gate*, notasi i yang merupakan nilai aktivasi dari *input gate*, notasi c yang merupakan nilai aktivasi dari *memory gate* dan notasi o yang merupakan nilai aktiasi dari *output gate*.

Perhitungan LSTM unit ke-1, prediksi harga P_{13}

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

$$F_1 = \sigma(-0.4739)$$

$$F_1 = 0.3836$$

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

$$I_1 = \sigma(-1.2817)$$

$$I_1 = 0.2172$$

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

$$Z_1 = \tanh(1.1526)$$

$$Z_1 = 0.8186$$

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

$$C_1 = (0.3218 * 0) + (0.2172 * 0.8186)$$

$$C_1 = 0.1778$$

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

$$O_1 = \sigma(0.1923)$$

$$O_1 = 0.5479$$

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

$$H_1 = O_1 * \tanh(C_1)$$

$$H_1 = 0.5479 * \tanh(0.1778)$$

$$H_1 = 0.5479 * 0.1759$$

$$H_1 = 0.0964$$

Pada perhitungan di atas, merupakan perhitungan LSTM unit pertama

dengan menggunakan nilai *input* X_t yang sudah ditentukan sebelumnya dengan bertujuan memprediksi harga saham P_{13} . Sesuai dengan persamaan 2.15, maka nilai H_1 yang didapatkan adalah 0.0964. Yang nantinya nilai H_1 akan digunakan pada perhitungan unit selanjutnya, sebagai H_{t-1} sesuai perhitungan yang berada dibawah.

Perhitungan LSTM unit ke-2, prediksi harga P_{13}

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

$$F_2 = \sigma((-0.0862 * 0.0964) + (-0.4739) + 0)$$

$$F_2 = \sigma(-0.4822)$$

$$F_2 = 0.3817$$

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

$$I_2 = \sigma((-0.2373 * 0.0964) + (-1.2817) + 0)$$

$$I_2 = \sigma(-1.30465114465)$$

$$I_2 = 0.2133$$

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

$$Z_2 = \tanh((0.2134 * 0.0964) + (1.1526) + 0)$$

$$Z_2 = \tanh(1.1732)$$

$$Z_2 = 0.8253$$

$$C_2 = (F_t * C_1) + (I_t * Z_t)$$

$$C_2 = (0.3817 * 0.1778) + (0.2133 * 0.8253)$$

$$C_2 = 0.2439$$

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

$$O_2 = \sigma((0.0356 * 0.0964) + (0.1923) + 0)$$

$$O_2 = \sigma(0.1958)$$

$$O_2 = 0.5488$$

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

$$H_2 = O_2 * \tanh(C_2)$$

$$H_2 = 0.5488 * \tanh(0.2440)$$

$$H_2 = 0.5488 * 0.2392$$

$$H_2 = 0.1322$$

Pada perhitungan di atas, *output* H_2 didapatkan dengan nilai 0.1322. Perlu

diingat, karena hasil H_2 merupakan hasil dari pemodelan LSTM dari nilai *input* yang berupa normalisasi dari *MinMaxScaler*. Maka, nilai *output* perlu di *Inverse Scaler Transform* yang berfungsi untuk mengubah nilai normalisasi *MinMaxScaler* menjadi nilai asli yang dapat dipahami. Maka, hasil H_2 setelah dilakukan *Inverse Scaler Transform* untuk prediksi harga P_{13} adalah 7407.63.

Perhitungan LSTM unit ke-1, prediksi harga P_{14}

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

$$F_1 = \sigma(-0.5139)$$

$$F_1 = 0.3742$$

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

$$I_1 = \sigma(-1.4146)$$

$$I_1 = 0.1954$$

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

$$Z_1 = \tanh(1.2722)$$

$$Z_1 = 0.8544$$

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

$$C_1 = (0.3742 * 0) + (0.8544 * 0.1954)$$

$$C_1 = 0.1670$$

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

$$O_1 = \sigma((0.0356 * 0) + 0.2123)$$

$$O_1 = 0.5528$$

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

$$H_1 = O_1 * \tanh(C_1)$$

$$H_1 = 0.5528 * \tanh(0.1670)$$

$$H_1 = 0.5528 * 0.1654$$

$$H_1 = 0.0915$$

Pada perhitungan di atas, merupakan perhitungan LSTM unit pertama dengan menggunakan nilai *input* X_t yang sudah ditentukan sebelumnya untuk memprediksi harga saham P_{14} . Sesuai dengan persamaan 2.15, maka nilai H_1 yang didapatkan adalah 0.0915. Yang nantinya nilai H_1 akan digunakan pada perhitungan unit selanjutnya, sebagai H_{t-1} sesuai perhitungan yang berada

dibawah.

Perhitungan LSTM unit ke-2, prediksi harga P_{14}

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

$$F_2 = \sigma((-0.0862 * 0.0915) + -0.5139)$$

$$F_2 = \sigma(-0.5218)$$

$$F_2 = 0.3724$$

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

$$I_2 = \sigma((-0.2373 * 0.0915) + -1.4146)$$

$$I_2 = \sigma(-1.4364)$$

$$I_2 = 0.1921$$

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

$$Z_2 = \tanh((0.2134 * 0.0915) + 1.2722)$$

$$Z_2 = \tanh(1.2917)$$

$$Z_2 = 0.8595$$

$$C_2 = (F_t * C_1) + (I_t * Z_t)$$

$$C_2 = (0.3724 * 0.1670) + (0.8595 * 0.1921)$$

$$C_2 = 0.2273$$

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

$$O_2 = \sigma((0.0356 * 0.0915) + 0.2123)$$

$$O_2 = \sigma(0.2155)$$

$$O_2 = 0.5536$$

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

$$H_2 = O_2 * \tanh(C_2)$$

$$H_2 = 0.5536 * \tanh(0.2273)$$

$$H_2 = 0.5536 * 0.2234$$

$$H_2 = 0.1237$$

Pada perhitungan di atas, *output* H_2 didapatkan dengan nilai 0.1237. Perlu diingat, karena hasil H_2 merupakan hasil dari pemodelan LSTM dari nilai *input* yang berupa normalisasi dari *MinMaxScaler*. Maka, nilai *output* perlu di *Inverse Scaler Transform* yang berfungsi untuk mengubah nilai normalisasi *MinMaxScaler* menjadi nilai asli yang dapat dipahami. Maka, hasil H_2 setelah dilakukan *Inverse*

Scaler Transform untuk prediksi harga P_{14} adalah 7402.34.

Perhitungan LSTM unit ke-1, prediksi harga P_{15}

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

$$F_1 = \sigma(-0.5656)$$

$$F_1 = 0.3622$$

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

$$I_1 = \sigma(-1.5571)$$

$$I_1 = 0.1740$$

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

$$Z_1 = \tanh(1.400)$$

$$Z_1 = 0.8854$$

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

$$C_1 = (0.3622 * 0) + (0.8854 * 0.1740)$$

$$C_1 = 0.1541$$

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

$$O_1 = \sigma(0.2337)$$

$$O_1 = 0.5581$$

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

$$H_1 = O_1 * \tanh(C_1)$$

$$H_1 = 0.5581 * \tanh(0.1541)$$

$$H_1 = 0.5581 * 0.1529$$

$$H_1 = 0.0853$$

Pada perhitungan di atas, merupakan perhitungan LSTM unit pertama dengan menggunakan nilai *input* X_t yang sudah ditentukan sebelumnya untuk memprediksi harga saham P_{15} . Sesuai dengan persamaan 2.15, maka nilai H_1 yang didapatkan adalah 0.091500371395874. Yang nantinya nilai H_1 akan digunakan pada perhitungan unit selanjutnya, sebagai H_{t-1} sesuai perhitungan yang berada dibawah.

Perhitungan LSTM unit ke-2, prediksi harga P_{15}

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

$$F_2 = \sigma((-0.0862 * 0.0853) + -0.5656)$$

$$F_2 = \sigma(-0.5730)$$

$$F_2 = 0.3605$$

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

$$I_2 = \sigma((-0.2373 * 0.0853) + -1.5571)$$

$$I_2 = \sigma(-1.5773)$$

$$I_2 = 0.1711$$

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

$$Z_2 = \tanh((0.2134 * 0.0853) + 1.4003)$$

$$Z_2 = \tanh(1.4185)$$

$$Z_2 = 0.8892$$

$$C_2 = (F_t * C_1) + (I_t * Z_t)$$

$$C_2 = (0.3605 * 0.1541) + (0.8892 * 0.1711)$$

$$C_2 = 0.2077$$

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

$$O_2 = \sigma((0.0356 * 0.0853) + 0.2337)$$

$$O_2 = \sigma(0.02367)$$

$$O_2 = 0.5589$$

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

$$H_2 = O_2 * \tanh(C_2)$$

$$H_2 = 0.5589 * \tanh(0.2077)$$

$$H_2 = 0.5589 * 0.2048$$

$$H_2 = 0.1144$$

Pada perhitungan di atas dengan menggunakan pemodelan dari LSTM maka mendapatkan nilai *output* H_2 0.1144. Maka, hasil H_2 setelah dilakukan *Inverse Scaler Transform* untuk prediksi harga P_{15} adalah 7396.56. Maka hasil prediksi P_{13} , P_{14} , P_{15} berturut-turut adalah: 7407.63, 7402.34, 7396.56.

3.4.4.2 Perhitungan *Autoregressive Integrated Moving Average*

Pehitungan Moving Average dengan menggunakan 2 hari kebelakang:
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 |
| <i>Day6</i> | 7600 |
| <i>Day7</i> | 7700 |
| <i>Day8</i> | 7700 |
| <i>Day9</i> | 7700 |
| <i>Day10</i> | 7850 |
| <i>Day11</i> | 7750 |
| <i>Day12</i> | 7675 |
| <i>Day13</i> | 7675 |
| <i>Day14</i> | 7775 |
| <i>Day15</i> | 7950 |

| Date | Close Price |
|------------|-------------|
| 19/01/2022 | P_{13} |
| 20/01/2022 | P_{14} |
| 21/01/2022 | P_{15} |

Berikut adalah inisialisasi data untuk perhitungan ARIMA. Dengan menggunakan nilai parameter p,d,q ialah 2,1,2. Pertama-tama akan dilakukan *Moving Average* dengan bertujuan *smoothing close price*. Lalu, nilai yang setelah dilakukan smooting akan digunakan lagi untuk memprediksi harga P_{13},P_{14},P_{15} dengan menggunakan *Autoregressive* yang sudah dilakukan *difference*.

Perhitungan *Moving Average*

Ordo $q = 2$

$$Y_3 = (Day_2 + Day_1)/2$$

$$Y_3 = (7400 + 7325)/2$$

$$Y_3 = 7362.5$$

$$Y_4 = (Day_3 + Day_2)/2$$

$$Y_4 = (7450 + 7400)/2$$

$$Y_4 = 7425$$

$$Y_5 = (Day_4 + Day_3)/2$$

$$Y_5 = (7475 + 7450)/2$$

$$Y_5 = 7462.5$$

$$Y_6 = (Day_5 + Day_4)/2$$

$$Y_6 = (7650 + 7475)/2$$

$$Y_6 = 7562.5$$

$$Y_7 = (Day_6 + Day_5)/2$$

$$Y_7 = (7600 + 7650)/2$$

$$Y_7 = 7625$$

$$Y_8 = (Day_7 + Day_6)/2$$

$$Y_8 = (7700 + 7600)/2$$

$$Y_8 = 7650$$

$$Y_9 = (Day_8 + Day_7)/2$$

$$Y_9 = (7700 + 7700)/2$$

$$Y_9 = 7700$$

$$Y_{10} = (Day_9 + Day_8)/2$$

$$Y_{10} = (7700 + 7700)/2$$

$$Y_{10} = 7700$$

$$Y_{11} = (Day_{10} + Day_9)/2$$

$$Y_{11} = (7850 + 7700)/2$$

$$Y_{11} = 7775$$

$$Y_{12} = (Day_{11} + Day_{10})/2$$

$$Y_{12} = (7750 + 7850)/2$$

$$Y_{12} = 7800$$

$$Y_{13} = (Day_{12} + Day_{11})/2$$

$$Y_{13} = (7675 + 7750)/2$$

$$Y_{13} = 7712.5$$

$$Y_{14} = (Day_{13} + Day_{12})/2$$

$$Y_{14} = (7675 + 7675)/2$$

$$Y_{14} = 7675$$

$$Y_{15} = (Day_{14} + Day_{13})/2$$

$$Y_{15} = (7775 + 7675)/2$$

$$Y_{15} = 7725$$

Pada perhitungan di atas, adalah hasil dari *Moving Average* dengan nilai parameter ordo q ialah 2. Jadi, setiap perhitungan untuk mencari Y_t akan menggunakan nilai rata-rata dari ϵ_{t-q} atau $\epsilon_{t-1} + \epsilon_{t-2}$.

Perhitungan *difference*

Ordo $q = 1$

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

$$Df_{13} = y_{13} - y_{12}$$

$$Df_{13} = 7712.5 - 7800$$

$$Df_{13} = -87.5$$

$$Df_{14} = y_{14} - y_{13}$$

$$Df_{14} = 7675 - 7712.5$$

$$Df_{14} = -37.5$$

$$Df_{15} = y_{15} - y_{14}$$

$$Df_{15} = 7725 - 7675$$

$$Df_{15} = 50$$

Pada perhitungan di atas, adalah hasil dari *Difference* dengan nilai ordo d ialah 1. Karena yang ingin diprediksi saat hari ke-13 sampai ke-15, maka dilakukan *difference* Df_{13} sampai Df_{15} dan mendapatkan nilai -87.5, -37.5, 50, yang nantinya nilai *difference* tersebut akan digunakan untuk memprediksi harga saham dengan menggunakan perhitungan *Autoregressive* yang berada dibawah.

Perhitungan Autoregressive

Ordo $p = 2$

| Coefficients | |
|--------------|----------|
| Intercept | -53.9451 |
| ϕ_1 | 0.5894 |
| ϕ_2 | 0.4256 |

Nilai yang sudah di *Smoothing Moving Average*

| Y_t | Close |
|----------|--------|
| Y_{11} | 7775 |
| Y_{12} | 7800 |
| Y_{13} | 7712.5 |
| Y_{14} | 7675 |
| Y_{15} | 7725 |

Rumus: $Y_t = c + Df_{13} + \phi_1 * y_{t-1} + \phi_2 * y_{t-2} + \dots + \phi_p y_{t-p}$

$$Y_{13} = c + Df_{13} + \phi_1 * y_{12} + \phi_2 * y_{11}$$

$$Y_{13} = -53.9451 + -87.5 + (7800 * 0.5894) + (7775 * 0.4256)$$

$$Y_{13} = 7764.92$$

$$Y_{14} = c + Df_{14} + \phi_1 * y_{13} + \phi_2 * y_{12}$$

$$Y_{14} = -53.9451 + -37.5 + (7712.5 * 0.5894) + (7800 * 0.4256)$$

$$Y_{14} = 7773.98$$

$$Y_{15} = c + Df_{15} + \phi_1 * y_{14} + \phi_2 * y_{13}$$

$$Y_{15} = -53.9451 + 50 + (7675 * 0.5894) + (7712.5 * 0.4256)$$

$$Y_{15} = 7802.14$$

Pada perhitungan di atas, adalah hasil prediksi yang menggunakan model ARIMA(2,1,2) yang didapatkan setelah melakukan perhitungan manual dengan

rumus 2.2. Maka, hasil prediksi P_{13} sampai P_{15} berturut-turut ialah 7764.92, 7773.98, 7802.14.

3.4.4.3 Perbandingan Evaluasi *Forecasting*

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

Evaluasi *Forecasting* dengan RMSE

| | | |
|---------------------|------------|-------|
| Close Price aktual: | Day_t | Close |
| | Day_{13} | 7675 |
| | Day_{14} | 7775 |
| | Day_{15} | 7950 |

| | | | |
|-----------------------------|----------|--------------|---------------|
| Close Price hasil prediksi: | P_t | Close (LSTM) | Close (ARIMA) |
| | P_{13} | 7407.63 | 7764.92 |
| | P_{14} | 7402.34 | 7773.98 |
| | P_{15} | 7396.56 | 7802.14 |

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

perhitungan RMSE untuk LSTM:

$$(y_{i13} - y_{p13})^2 = (7675 - 7407.63)^2 = 71486.7169$$

$$(y_{i14} - y_{p14})^2 = (7775 - 7402.34)^2 = 138875.4756$$

$$(y_{i15} - y_{p15})^2 = (7950 - 7396.56)^2 = 306295.8336$$

$$\sum_{i=1}^N (y_i - y_p)^2 / N$$

$$= (71486.7169 + 138875.4756 + 306295.8336) / 3$$

$$= \sqrt{172219.342033}$$

$$RMSE_{LSTM} = 414.9930$$

perhitungan RMSE untuk ARIMA:

$$(y_{i13} - y_{p13})^2 = (7675 - 7764.92)^2 = 8085.6064$$

$$(y_{i14} - y_{p14})^2 = (7775 - 7773.98)^2 = 1.0404$$

$$(y_{i15} - y_{p15})^2 = (7950 - 7802.14)^2 = 21862.5796$$

$$\sum_{i=1}^N (y_i - y_p)^2 / N$$

$$= (8085.6064 + 1.0404 + 21862.5796) / 3$$

$$= \sqrt{9983.07546667}$$

$$RMSE_{ARIMA} = 99.9153$$

Pada perhitungan Evaluasi *Forecasting* di atas dengan menggunakan rumus 2.24, maka dapat diketahui bahwa model ARIMA mempunyai hasil model prediksi lebih baik dengan mendapatkan RMSE 99.9153 dibandingkan dengan model LSTM yang mendapatkan RMSE 414.993 dalam memprediksi harga saham BBKA dengan jumlah data *training* 12 rows dan data *testing* 3 rows.

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. K. Choi, "Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model," 1 Oktober 2018.
- [10] A. Geron, *Hands On Machine Learning with ScikitLearn, Keras & Tensorflow*, O'Reilly Media, Inc, 2019

- [11] A. Zhang, Z. C. Lipton, M. Li, and Alexander, *Dive into Deep learning*, Release 0.17.0, 2021
- [12] F. R. Pratama, "Implementasi Algoritme Random Forest untuk Prediksi Pergerakan Harga Saham," Dept. Informatika ITHB, 2017.
- [13] J. D. Schwager. *Getting Started in Technical Analysis*. John Wiley & Sons, 1999
- [14] Ian G., Yoshua B., Aaron C., *Deep Learning*. MIT Press, 2016
- [15] Statistics How to. "RMSE: Root Mean Square Error". [Online]. Available at: <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/> [Accessed: Feb. 20, 2022]
- [16] A Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Illustrated Edition. Springer, 2012
- [17] S. Maniath, A. Ashok, "Deep Learning LSTM Based Ransomware Detection", IEEE, 2017.
- [18] Yahoo Finance, 2022. "Historical Price Data Indonesia Equity". [Online]. Available: <https://finance.yahoo.com/> [Accessed: Jan. 31, 2022].
- [19] R. J. Hyndman, *Forecasting: Principles & Practice*, Illustrated Edition. Springer, 2014
- [20] MBA Knowledge Base, 2021. Time Horizon in Forecasting. [Online]. Available: <https://www.mbaknol.com/managerial-economics/time-horizon-in-forecasting/> [Accessed: March. 28, 2022]
- [21] M. Pradana. Saham *Blue Chip* dan Saham Gorengan . Available: <https://investbro.id/saham-gorengan/> [Accessed: Apr. 12, 2022]
- [22] F Bre, J. M. Gimenez, V. D. Fachinotti. Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. [Online]. Available: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051 [Accessed: Feb. 22, 2022]
- [23] Geeksforgeeks, 2019. Implement sigmoid function using Numpy. [Online]. Available: <https://www.geeksforgeeks.org/>

DAFTAR REFERENSI

- `implement-sigmoid-function-using-numpy/` [Accessed: Feb. 24, 2022]
- [24] 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.
- [25] P. Mladjenovic. *Stock Investing For Dummies*, 6th Edition. John Wiley & Sons, 2020