

**PENERAPAN GATED RECURRENT UNIT UNTUK
PREDIKSI PERGERAKAN HARGA SAHAM PADA BURSA
EFEK INDONESIA**

TUGAS AKHIR

**Joseph Axel Ripto
1118032**



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

**PENERAPAN GATED RECURRENT UNIT UNTUK
PREDIKSI PERGERAKAN HARGA SAHAM PADA BURSA
EFEK INDONESIA**

TUGAS AKHIR

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

**Joseph Axel Ripto
1118032**



**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
TAHUN 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, 2 Juli 2022

A handwritten signature in black ink, appearing to be 'J. Ripto', written in a cursive style.

**Joseph Axel Ripto
1118032**

HALAMAN PENGESAHAN TUGAS AKHIR

Tugas Akhir dengan judul:

**PENERAPAN GATED RECURRENT UNIT UNTUK PREDIKSI
PERGERAKAN HARGA SAHAM PADA BURSA EFEK INDONESIA**

yang disusun oleh:

Joseph Axel Ripto

1118032

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

Hari / tanggal : Sabtu, 2 juli 2022

Waktu : 11.00 WIB

Menyetujui

Pembimbing Utama:



Dr. Hery Heryanto, M.Kom.

116007

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama : Joseph Axel Ripto

NIM : 1118032

Program Studi : Informatika

demi 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:

PENERAPAN GATED RECURRENT UNIT UNTUK PREDIKSI
PERGERAKAN HARGA SAHAM PADA BURSA EFEK INDONESIA

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, 2 Juli 2022

Yang menyatakan



Joseph Axel Ripto

ABSTRAK

Nama : Joseph Axel Ripto
Program Studi : Informatika
Judul : Penerapan Gated Recurrent Unit untuk Prediksi Pergerakan Harga Saham pada Bursa Efek Indonesia

Prediksi harga saham yang merupakan *time series data* adalah proses yang tidak mudah. Hal ini dikarenakan karakteristik data saham yang mempunyai *noise* yang tinggi, kompleksitas yang tinggi, dan struktur nonlinier. Faktor anomali dalam pasar saham juga memengaruhi hasil prediksi harga saham. Penelitian ini menguji model GRU untuk memprediksi harga saham di Bursa Efek Indonesia.

Penelitian ini menggunakan empat indikator, yaitu *training period*, *unit*, *batch size*, dan *epoch*. Penelitian ini memperlihatkan pengaruh dari setiap indikator terhadap masing-masing *dataset*. Alat ukur yang digunakan untuk menghitung akurasi dalam penelitian ini adalah *Root Mean Square Error* (RMSE). Penelitian ini dilakukan terhadap enam *dataset*, yaitu PT. Aneka Tambang, PT. Vale Indonesia, PT. Indofood, PT. Japfa Comfeed Indonesia, PT. XL Axiata, dan PT. Telkom.

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, nilai akurasi terbaik sebesar 0.009977 didapatkan melalui percobaan terhadap *dataset* PT. Japfa Comfeed Indonesia. PT. Aneka Tambang menghasilkan akurasi terbaik sebesar 0.012138. Percobaan yang dilakukan pada *dataset* PT. Vale Indonesia menghasilkan akurasi terbaik sebesar 0.019060. Percobaan pada PT. XL Axiata menghasilkan akurasi terbaik sebesar 0.024266. Percobaan yang dilakukan pada PT. Indofood menghasilkan akurasi terbaik sebesar 0.029635. Terakhir, percobaan pada PT. Telkom memiliki akurasi terbaik sebesar 0.031479. Hasil pengujian membuktikan bahwa semua indikator yang digunakan dalam pengujian memiliki pengaruh terhadap nilai akurasi.

Kata kunci: *Gated Recurrent Unit* (GRU), *Recurrent Neural Network*, *Deep learning*, *Stock prediction*, *Time series forecasting*.

ABSTRACT

Name : Joseph Axel Ripto
Department : Informatics
Title : Application of Gated Recurrent Units for Predicting Stock Price Movements on the Bursa Efek Indonesia

Predicting stock prices which are time series data was not an easy process. This is due to the characteristics of the data that have high noise, high complexity, and nonlinear structure. Anomaly factors in the stock market also affect the results of stock price predictions. This study tested the GRU model to predict stock prices on the Bursa Efek Indonesia.

This study uses four indicators, namely training period, unit, batch size, and epoch. This study shows the effect of each indicator on each dataset. The measuring instrument used to calculate accuracy in this study is the Root Mean Square Error (RMSE). This study was conducted on three main datasets, namely PT. Aneka Tambang, PT. Indofood, and PT. Telkom.

Based on the results of the tests carried out in this study, the best accuracy value of 0.009977 was obtained through experiments on the dataset PT. Japfa Comfeed Indonesia. PT. Aneka Tambang produces the best accuracy of 0.012138. Experiments conducted on the PT. Vale Indonesia produced the best accuracy of 0.019060. An experiment at PT. XL Axiata produces the best accuracy of 0.024266. Experiments conducted at PT. Indofood produced the best accuracy of 0.029635. Finally, the experiment on PT. Telkom has the best accuracy of 0.031479. The test results prove that all the indicators used in the test have an influence on the accuracy value.

Keyword: Gated Recurrent Unit (GRU), Recurrent Neural Network, Deep learning, Stock prediction, Time series forecasting.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan yang Maha Esa karena atas rahmat dan bimbingan-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul "Penerapan *Gated Recurrent Unit* untuk Prediksi Pergerakan Harga Saham Pada Bursa Efek Indonesia". Laporan ini disusun sebagai salah satu syarat kelulusan di Institut Teknologi Harapan Bangsa (ITHB). Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang Maha Esa, karena oleh-Nya penulis selalu mendapat pengharapan dan penguatan untuk menyelesaikan tugas akhir ini.
2. Bapak Dr. Hery Heryanto, M.Kom., selaku pembimbing utama Tugas Akhir yang senantiasa memberi saran, ilmu dan dukungan kepada penulis selama proses pembuatan Tugas Akhir.
3. Bapak Ventje Jeremias Lewi Engel, S.T., M.T. selaku penguji satu yang telah memberikan ilmu, masukan dan dukungan dalam menyelesaikan laporan Tugas Akhir.
4. Ibu Ir. Inge Martina, M.T. selaku penguji dua yang telah memberikan ilmu, masukan dan dukungan dalam menyelesaikan laporan Tugas Akhir.
5. Seluruh dosen dan staf Program Studi Informatika ITHB yang telah membantu dalam menyelesaikan Laporan Tugas Akhir ini.
6. Seluruh staf dan karyawan ITHB yang turut membantu kelancaran dalam menyelesaikan Laporan Tugas Akhir ini.
7. Orang tua yang selalu menyediakan waktu untuk memberikan doa, semangat dan dukungan kepada penulis untuk menyelesaikan Laporan Tugas Akhir.
8. Hanjaya Suryalim, Daniel Christianto dan Daniel Alexander yang sudah mendukung dan memberikan semangat serta berjuang bersama untuk menyelesaikan Tugas Akhir.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna karena keterbatasan waktu dan pengetahuan yang dimiliki oleh penulis. Oleh karena itu, kritik dan saran untuk membangun kesempurnaan tugas akhir ini sangat diharapkan. Semoga dengan adanya tugas akhir ini dapat membantu pihak yang membutuhkan.

Bandung, 2 Juli 2022

Hormat penulis,

A handwritten signature in black ink, appearing to be 'J. Ripto', written in a cursive style.

Joseph Axel Ripto

DAFTAR ISI

ABSTRAK	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1-1
1.1 Latar Belakang	1-1
1.2 Rumusan Masalah	1-2
1.3 Tujuan Penelitian	1-2
1.4 Batasan Masalah	1-3
1.5 Kontribusi Penelitian	1-3
1.6 Metodologi Penelitian	1-3
1.7 Sistematika Pembahasan	1-5
BAB 2 LANDASAN TEORI	2-1
2.1 Tinjauan Pustaka	2-1
2.1.1 Pembelajaran Mesin	2-1
2.1.2 <i>Artificial Neural Network</i> (ANN)	2-2
2.1.2.1 <i>Perceptron</i>	2-2
2.1.2.2 <i>Multi Layer Perceptron</i> (MLP)	2-3
2.1.3 Fungsi Aktivasi	2-4
2.1.3.1 Fungsi <i>Sigmoid</i>	2-4
2.1.3.2 Fungsi <i>Hyperbolic Tangent</i> (tanh)	2-4
2.1.4 <i>Recurrent Neural Network</i> (RNN)	2-5
2.1.5 <i>Root Mean Square Error</i> (RMSE)	2-7
2.1.6 Normalisasi <i>Min-max</i>	2-7
2.1.7 <i>Gated Recurrent Unit</i> (GRU)	2-8
2.1.8 <i>Library</i> yang digunakan	2-10
2.1.8.1 <i>Library</i> Pandas	2-10

2.1.8.2	<i>Library NumPy</i>	2-11
2.1.8.3	<i>Library Matplotlib</i>	2-12
2.1.8.4	<i>Library Keras</i>	2-12
2.1.8.5	<i>Gated Recurrent Unit (GRU)</i> dalam <i>Library Keras</i>	2-14
2.2	Tinjauan Studi	2-17
2.3	Tinjauan Objek	2-20
2.3.1	Jenis-jenis Saham	2-20
2.3.1.1	Berdasarkan Hak Kepemilikan Saham	2-20
2.3.1.2	Berdasarkan Cara Peralihan Saham	2-21
2.3.1.3	Berdasarkan Kinerja Perdagangan	2-21
2.3.2	Teori Prediksi Saham	2-21
2.3.3	Penjelasan Tentang Dataset	2-23
BAB 3	ANALISIS DAN PERANCANGAN SISTEM	3-1
3.1	Analisis Masalah	3-1
3.2	Kerangka Pemikiran	3-2
3.3	Urutan Proses Global	3-3
3.4	Analisis Manual	3-4
3.4.1	<i>Dataset</i>	3-4
3.4.2	Normalisasi <i>Min-max</i>	3-5
3.4.3	<i>Split Data</i>	3-6
3.4.4	Perhitungan <i>Gated Recurrent Unit (GRU)</i>	3-7
3.4.4.1	Analisis Parameter pada Model GRU	3-8
3.4.4.2	Contoh Perhitungan Inisialisasi <i>U</i>	3-9
3.4.4.3	Contoh Perhitungan Inisialisasi <i>W</i>	3-10
3.4.4.4	Perhitungan <i>t</i> Pertama	3-11
3.4.4.5	Perhitungan <i>t</i> Kedua	3-14
3.4.5	Perhitungan <i>Perceptron</i> pada Lapisan Keluaran	3-18
3.4.6	Perhitungan <i>Root Mean Square Error (RMSE)</i>	3-19
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 Perangkat Lunak	4-1
4.2.1	Implementasi <i>Class</i>	4-2
4.2.1.1	<i>Class Dataset</i>	4-2
4.2.1.2	<i>Class Preprocessing</i>	4-2

4.2.1.3	<i>Class DataSplit</i>	4-3
4.2.1.4	<i>Class Training</i>	4-4
4.2.1.5	<i>Class Visualization</i>	4-4
4.2.2	Penggunaan Jupyter Notebook	4-5
4.2.3	Penggunaan <i>Dataset</i>	4-6
4.3	Implementasi Aplikasi	4-7
4.4	Pengujian	4-8
4.4.1	Pengujian <i>Training Period</i> pada <i>Dataset</i> Utama	4-9
4.4.2	Pengujian <i>Unit</i> pada <i>Dataset</i> Utama	4-9
4.4.3	Pengujian <i>Batch Size</i> pada <i>Dataset</i> Utama	4-9
4.4.4	Pengujian <i>Epoch</i> pada <i>Dataset</i> Utama	4-9
4.4.5	Pengujian Kombinasi Parameter Terbaik pada <i>Dataset</i> Utama	4-10
4.4.6	Pengujian <i>Dataset</i> Pembanding Menggunakan Parameter Terbaik pada <i>Dataset</i> Utama	4-10
4.5	Hasil Pengujian	4-10
4.5.1	Hasil Pengujian <i>Training Period</i> pada <i>Dataset</i> Utama	4-10
4.5.2	Hasil Pengujian <i>Unit</i> pada <i>Dataset</i> Utama	4-11
4.5.3	Hasil Pengujian <i>Batch Size</i> pada <i>Dataset</i> Utama	4-12
4.5.4	Hasil Pengujian <i>Epoch</i> pada <i>Dataset</i> Utama	4-13
4.5.5	Hasil Pengujian Kombinasi Parameter Terbaik pada <i>Dataset</i> Utama	4-14
4.5.6	Hasil Pengujian <i>Dataset</i> Pembanding Menggunakan Parameter Terbaik pada <i>Dataset</i> Utama	4-16
4.6	Analisis Kesalahan	4-18
4.6.1	Validasi untuk Mendeteksi Masalah <i>Overfitting</i> atau <i>Underfitting</i>	4-18
4.6.2	<i>Dataset</i> yang Digunakan	4-18
4.6.3	<i>Seleksi Fitur</i>	4-18
BAB 5	KESIMPULAN DAN SARAN	5-1
5.1	Kesimpulan	5-1
5.2	Saran	5-1

DAFTAR TABEL

2.1	Tabel Fungsi <i>Library Pandas</i>	2-10
2.2	Tabel Fungsi <i>Library Numpy</i>	2-11
2.3	Tabel Fungsi <i>Library Matplotlib.Pyplot</i>	2-12
2.4	Tabel Fungsi <i>Library Keras</i>	2-13
2.5	Tinjauan Studi	2-17
2.6	Tabel Karakteristik <i>Dataset</i> Utama	2-24
2.7	Tabel Karakteristik <i>Dataset</i> Pembanding	2-26
3.1	Tabel Contoh Matriks Data Saham	3-5
3.2	Tabel Contoh Matriks Data Saham Sebelum Normalisasi	3-5
3.3	Tabel Contoh Matriks Data Saham Sesudah Normalisasi	3-6
3.4	Tabel Contoh Matriks Data Latihan	3-7
3.5	Tabel Penjelasan Parameter Model GRU	3-8
4.1	Atribut pada <i>Class Dataset</i>	4-2
4.2	Daftar <i>Method Class Dataset</i>	4-2
4.3	Daftar <i>Method Class Preprocessing</i>	4-3
4.4	Atribut pada <i>Class DataSplit</i>	4-3
4.5	Daftar <i>Method Class DataSplit</i>	4-4
4.6	Atribut pada <i>Class Training</i>	4-4
4.7	Daftar <i>Method Class Training</i>	4-4
4.8	Daftar <i>Method Class Visualization</i>	4-5
4.9	Tabel Pembagian <i>Dataset</i>	4-7
4.10	Tabel Nilai RMSE Berdasarkan <i>Training Period</i>	4-10
4.11	Tabel Nilai RMSE Berdasarkan <i>Unit</i>	4-11
4.12	Tabel Nilai RMSE Berdasarkan <i>Batch Size</i>	4-12
4.13	Tabel Nilai RMSE Berdasarkan <i>Epoch</i>	4-14
4.14	Tabel Model dengan RMSE Terbaik	4-15
4.15	Tabel Hasil Terbaik pada <i>Dataset</i> Utama	4-16
4.16	Tabel Pengujian <i>Dataset</i> Pembanding Menggunakan Parameter Terbaik pada <i>Dataset</i> Utama	4-17
4.17	Tabel Model dengan RMSE Terbaik pada <i>Dataset</i> Pembanding	4-17

DAFTAR GAMBAR

2.1	<i>Threshold logic unit</i> : Sebuah neuron buatan yang menghitung jumlah bobot dari setiap masukan lalu diterapkan ke dalam <i>step function</i> [9]	2-2
2.2	Contoh arsitektur <i>Multi Layer Perceptron</i> (MLP)	2-3
2.3	Fungsi <i>sigmoid</i> dengan keluaran dalam skala 0 sampai 1 [8]	2-4
2.4	Fungsi <i>Hyperbolic Tangent</i> (tanh) dengan keluaran dalam skala -1 sampai 1 [8]	2-5
2.5	(a) sistematis dari sebuah sel RNN; (b) sel RNN (<i>unrolled</i>)	2-6
2.6	Struktur Gated Recurrent Unit	2-9
3.1	Diagram Kerangka Pemikiran	3-2
3.2	Flowchart Urutan Proses Global	3-4
3.3	Arsitektur perhitungan GRU	3-8
4.1	Halaman Utama	4-8
4.2	Tampilan Hasil Prediksi	4-8
4.3	Model Terbaik ANTM	4-15
4.4	Model Terbaik INDF	4-16
4.5	Model Terbaik TLKM	4-16
4.6	Korelasi Fitur pada <i>Dataset</i> ANTM	4-19
4.7	Korelasi Fitur pada <i>Dataset</i> INDF	4-19
4.8	Korelasi Fitur pada <i>Dataset</i> TLKM	4-19

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Terdapat dua jenis sekuritas yang diperdagangkan di bursa efek, yaitu saham dan obligasi. Saham dan obligasi adalah instrumen keuangan yang disebut sekuriti. Perbedaan saham dan obligasi terletak pada perlakuan sekuritas terhadap pemilik sekuriti. Pemegang saham dianggap sebagai pemilik perusahaan penerbit saham tersebut, sedangkan pemegang obligasi dianggap sebagai pemberi pinjaman kepada penerbit obligasi [1].

Menurut artikel yang ditulis oleh Setiani Widiarti, seorang investor harus memutuskan kapan dan ke mana dirinya menginvestasikan pendapatannya. Tujuan berinvestasi adalah memperoleh keuntungan. Keuntungan yang didapat investor bergantung pada nilai harga saham [1]. Proses jual beli saham di Indonesia dilakukan di Bursa Efek Indonesia untuk perusahaan-perusahaan yang telah terdaftar [2].

Prediksi harga saham yang merupakan data *time series*, data yang bergantung pada waktu, adalah proses yang tidak mudah. Hal ini dikarenakan karakteristik data saham yang mempunyai *noise* yang tinggi, kompleksitas yang tinggi, dan struktur nonlinier [3]. Faktor anomali dalam pasar saham juga memengaruhi hasil prediksi harga saham. Pada beberapa tahun terakhir, studi yang sudah ada telah menunjukkan bahwa teknologi *Machine Learning* (ML) mampu secara efektif menangkap struktur nonlinier dalam data pasar saham yang kompleks [4].

Hingga saat ini, banyak penelitian yang mengangkat topik tentang memprediksi pergerakan harga saham. Tidak sedikit juga metode yang digunakan untuk membuat model prediksi pergerakan harga saham. Salah satu model yang digunakan untuk memprediksi pergerakan saham adalah *Support Vector Machine* (SVM). Model SVM yang diuji dengan menggunakan data SSE 50 Index menghasilkan akurasi sebesar 71,33%. Nilai akurasi tersebut meningkat menjadi 89,93% setelah menambahkan *sentiment variables* kedalam data yang diuji [5].

Selain SVM, metode yang digunakan untuk model prediksi pergerakan harga saham adalah *Long Short-Term Memory* (LSTM). Hasil penelitian yang dilakukan oleh S. Chen menunjukkan bahwa LSTM mempunyai hasil *Mean Squared Error* (MSE) sebesar 0.0072. MSE yang dihasilkan oleh model LSTM

lebih kecil dibandingkan dengan *Principal Component Analysis - Support Vector Machine* (PCA-SVM) dan Random Forest. MSE yang lebih kecil tersebut menandakan bahwa LSTM mempunyai kesalahan yang lebih kecil dalam prediksi harga saham. Selain menggunakan LSTM, S. Chen mencoba untuk melakukan seleksi fitur dengan metode *Genetic Algorithm* (GA). Percobaan tersebut menghasilkan MSE yang lebih kecil dibandingkan dengan LSTM [6].

Penelitian tahun 2019 menunjukkan bahwa *Gated Recurrent Unit* (GRU) dan LSTM memiliki performa yang lebih baik dibandingkan *Multilayer Perceptron* (MLP) dan SVM. Model GRU memiliki *return ratio* (keuntungan) paling besar, sebesar 5.722242 yang menunjukkan bahwa model tersebut memprediksi pergerakan harga saham dengan baik. GRU memiliki nilai RMSE sebesar 0.000511. RMSE pada GRU memiliki perbedaan 0.0001 dengan hasil RMSE pada LSTM [7].

Penelitian ini menguji metode GRU untuk memprediksi harga saham. GRU adalah model LSTM yang telah dimodifikasi. GRU menggunakan dua *gate* dalam prosesnya, yaitu *Reset gate* dan *update gate* yang digunakan untuk *hidden state*. *Reset gate* menentukan memperhitungkan kombinasi *input* saat ini dengan *historic memory* (data masa lalu yang disimpan). *Update gate* bertanggung jawab untuk memutuskan tingkatan *historic memory* mana yang harus dipertahankan dalam *node* [7].

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat diidentifikasi rumusan masalah sebagai berikut:

1. Berapakah nilai akurasi dalam penerapan algoritme *Gated Recurrent Unit* dalam prediksi harga saham pada *dataset* perusahaan di Bursa Efek Indonesia?
2. Parameter apa saja yang berpengaruh dalam penerapan algoritme *Gated Recurrent Unit* untuk prediksi harga saham pada *dataset* perusahaan di Bursa Efek Indonesia?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, penelitian memiliki tujuan sebagai berikut:

1. Menguji nilai akurasi dalam penerapan algoritme *Gated Recurrent Unit* dalam prediksi harga saham pada *dataset* Bursa Efek Indonesia
2. Menguji parameter yang sesuai pada *Gated Recurrent Unit* (GRU), sehingga

prediksi harga saham lebih akurat

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan yaitu *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset" dalam rentang waktu satu hari.
2. *Dataset* yang digunakan terdapat pada rentang tanggal 6 Juli 2015 hingga 14 Oktober 2021.
3. Sektor saham yang diamati adalah pertambangan, telekomunikasi, dan makanan.
4. Data perusahaan yang diamati adalah PT. Telkom, PT. Aneka Tambang, PT. XL Axiata, PT. Vale Indonesia, PT. Japfa Comfeed Indonesia, dan PT. Indofood.

1.5 Kontribusi Penelitian

Kontribusi yang diberikan dari penelitian ini adalah menguji apakah metode *Gated Recurrent Unit* menghasilkan tingkat akurasi yang tinggi dalam menentukan strategi dalam jual beli saham di Bursa Efek Indonesia.

1.6 Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur

Penelitian ini diawali dengan tinjauan studi yaitu mengumpulkan bahan-bahan referensi dari buku, artikel, penelitian, jurnal, makalah, dan *internet* yang berhubungan dengan topik penelitian ini, yaitu memprediksi pergerakan harga saham pada Bursa Efek Indonesia menggunakan Metode *Gated Recurrent Unit*.

2. Data Sampling

Data sampling diambil dari *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset" dalam rentang waktu satu hari. Data perusahaan yang diamati adalah PT. Telkom, PT. Aneka Tambang, dan PT. Indofood.

3. Analisis Masalah

Analisis masalah adalah tahap menganalisa permasalahan yang terjadi saat memprediksi harga saham menggunakan metode GRU, batasan masalah, dan kebutuhan yang diperlukan untuk memprediksi harga saham menggunakan metode GRU.

4. Perancangan dan Implementasi Algoritme

Pada tahap ini dilakukan perancangan model GRU yang dipakai untuk menyelesaikan masalah prediksi harga saham.

5. Pengujian

Pada tahap ini dilakukan pengujian terhadap model GRU. Pengujian ini menggunakan *Root Mean Square Error* (RMSE) sebagai alat ukur untuk mendapatkan performa dari hasil prediksi harga saham menggunakan metode GRU.

6. Dokumentasi

Pada tahap ini dilakukan dokumentasi terhadap hasil analisis dan implementasi secara tertulis dalam bentuk laporan tugas akhir.

1.7 Sistematika Pembahasan

BAB 1 PENDAHULUAN

Bab 1 merupakan pendahuluan yang berisikan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, kontribusi penelitian, dan metode penelitian. Pada latar belakang menjelaskan perkembangan saham di Indonesia dan pengenalan tentang teknologi yang sudah ada untuk prediksi harga saham. Rumusan masalah berisi tentang masalah parameter yang berpengaruh dan nilai akurasi yang dihadapi selama penelitian dijalani. Penelitian ini bertujuan untuk menguji nilai akurasi dan menguji parameter pada metode GRU. Batasan masalah penelitian ini berhubungan dengan *dataset* yang digunakan. Kontribusi penelitian ini untuk menguji dan membuktikan akurasi yang dihasilkan oleh metode GRU. Metodologi penelitian menjelaskan metodologi penelitian yang diterapkan pada penelitian ini.

BAB 2 LANDASAN TEORI

Bab 2 merupakan landasan teori yang menjelaskan dasar teori dari metode dan algoritme yang digunakan untuk mendukung penelitian ini. Teori-teori yang dipakai dalam penelitian ini, yaitu pembelajaran mesin, Artificial Neural Network (ANN), *Perceptron*, *Multi Layer Perceptron* (MLP), fungsi aktivasi, fungsi *Sigmoid*, fungsi *Hyperbolic Tangent* (tanh), *Recurrent Neural Network* (RNN), *Root Mean Square Error* (RMSE), *Gated Recurrent Unit* (GRU), normalisasi *Min-Max*, dan library yang digunakan.

BAB 3 ANALISIS DAN PERANCANGAN

Bab 3 merupakan analisis dan perancangan yang berisi analisis masalah, kerangka pemikiran, urutan proses global, dan analisis manual.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab 4 merupakan bagian yang menjelaskan bagaimana cara implementasi dan pengujian dilakukan dalam penelitian ini. Bab 4 dipecah ke dalam beberapa bagian, yaitu lingkungan implementasi, implementasi perangkat lunak, implementasi aplikasi, hasil pengujian, dan analisis kesalahan.

BAB 5 KESIMPULAN DAN SARAN

Bab 5 merupakan bagian penutup yang berisikan kesimpulan dari penelitian dan saran untuk perkembangan penelitian ini.

BAB 2 LANDASAN TEORI

Pada bab ini dibahas teori-teori dasar yang menjadi landasan dasar penelitian ini. Bab ini terdiri dari tinjauan pustaka, tinjauan studi, dan tinjauan objek.

2.1 Tinjauan Pustaka

Pada subbab ini dibahas teori-teori dasar yang dibutuhkan pada penelitian ini. Teori tersebut mencakup pembelajaran mesin, *Artificial Neural Network* (ANN), *Perceptron*, *Multi Layer Perceptron* (MLP), fungsi aktivasi, fungsi *Sigmoid*, fungsi *Hyperbolic Tangent* (tanh), *Recurrent Neural network* (RNN), *Root Mean Square Error* (RMSE), *Gated Recurrent Unit* (GRU), *Gated Recurrent Unit* (GRU) dalam *Library Keras*, normalisasi *Min-max*, dan *library* yang digunakan.

2.1.1 Pembelajaran Mesin

Pembelajaran mesin atau *machine learning* merupakan percabangan dari kecerdasan buatan. Pembelajaran mesin juga mempunyai bagian lebih kompleks yang disebut *Deep Learning* [8]. Pembelajaran mesin adalah ilmu dan seni dari pemrograman komputer sehingga komputer mampu belajar dari data. Komputer seolah diberi kemampuan untuk belajar tanpa harus diprogram secara eksplisit. Jadi, Pembelajaran mesin memberikan komputer kemampuan belajar untuk menyelesaikan masalah berdasarkan pola data.

Pembelajaran mesin menyederhanakan kode dalam program dan memiliki kinerja lebih baik daripada pendekatan tradisional. Pembelajaran mesin menyelesaikan masalah rumit yang tidak bisa diselesaikan dengan pendekatan tradisional. Selain itu, Pembelajaran mesin mampu menyelesaikan masalah yang melibatkan data dalam jumlah besar. Pembelajaran mesin juga beradaptasi dengan data baru [9].

Pembelajaran mesin dibagi menjadi empat tipe, yaitu *supervised learning*, *unsupervised learning*, *semisupervised learning*, dan *reinforcement learning*. Pada *Supervised Learning*, data latihan yang diberikan pada model disertakan dengan solusi yang diinginkan, biasa disebut label. *Supervised learning* biasa digunakan untuk menyelesaikan masalah klasifikasi atau regresi. *Unsupervised learning* tidak membutuhkan label pada data latihannya. *Unsupervised learning* biasanya

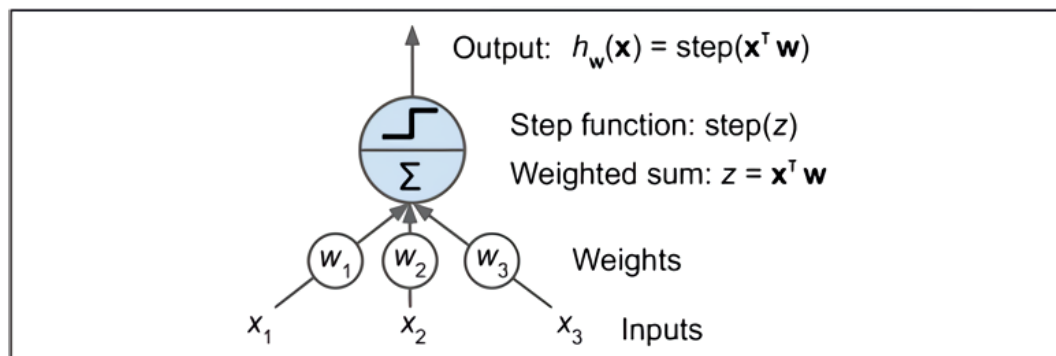
digunakan untuk menyelesaikan masalah pengelompokan, deteksi anomali, *association rule learning*, visualisasi, dan pengurangan dimensi. *Semisupervised learning* adalah gabungan dari *supervised learning* dan *unsupervised learning*. *Reinforcement learning* sangat berbeda dari tipe pembelajaran mesin yang lain. *Reinforcement learning* menggunakan sistem pembelajaran yang disebut agen yang melakukan aksi dan mengamati lingkungan. Setelah berhasil melakukan aksi, agen akan mendapatkan *reward* atau *penalty*. *Reinforcement learning* bertujuan untuk mencari strategi terbaik agar agennya mendapatkan poin *reward* sebanyak mungkin [9].

2.1.2 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah inti dari *Deep Learning*. ANN terinspirasi dari studi tentang sistem saraf pusat atau neuron pada otak manusia. ANN terdiri dari kumpulan neuron yang saling berhubungan dan terdiri dari beberapa lapisan. Cara kerja dasar ANN adalah neuron pada suatu lapisan memberikan pesan kepada neuron pada lapisan selanjutnya. Dalam ANN terdapat beberapa arsitektur, yaitu *Perceptron* dan *Multi Layer Perceptron* (MLP) [8] [9].

2.1.2.1 Perceptron

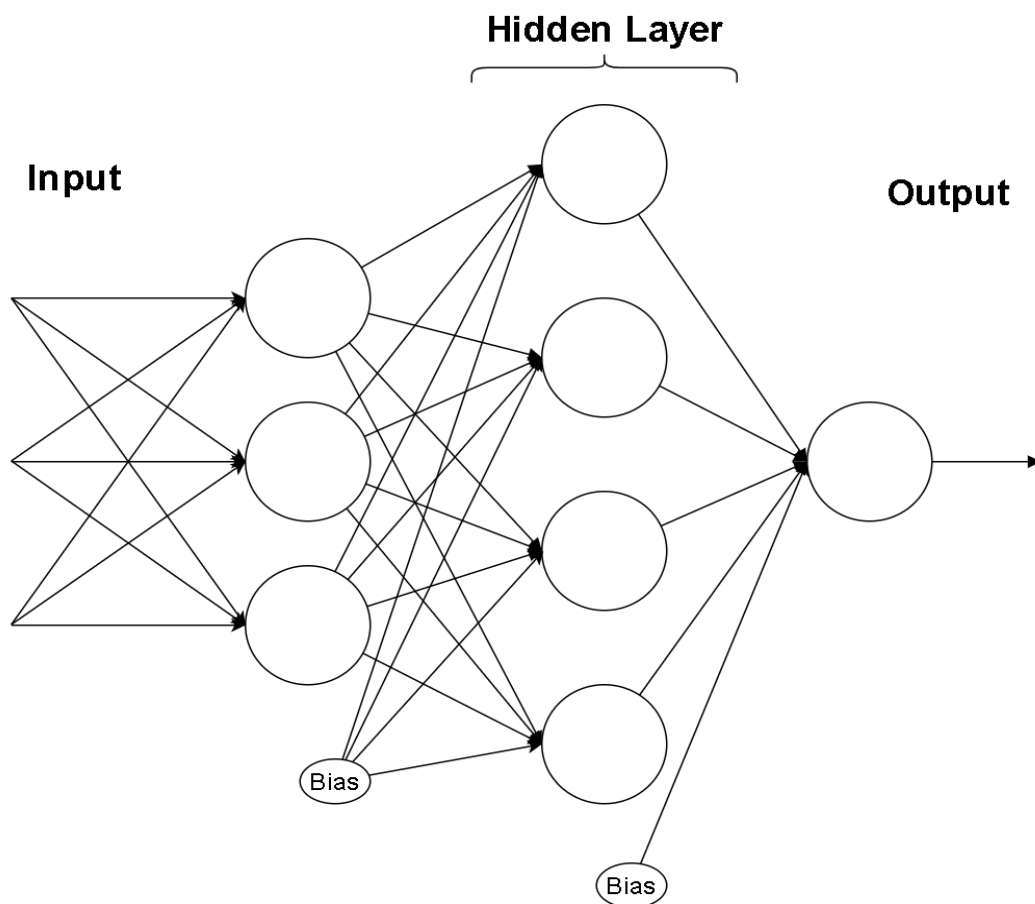
Perceptron adalah salah satu arsitektur ANN yang paling simpel. Gambar 2.1 merupakan gambar dari sebuah proses komputasi neuron buatan yang disebut *Threshold Logic Unit* (TLU), atau terkadang disebut *Linear Threshold Unit* (LTU). Masukan dan keluaran yang dihasilkan dari TLU berupa angka dan setiap masukan diasosiasikan dengan sebuah bobot. TLU menghitung jumlah bobot dari setiap masukan dengan rumus ($z = w_1x_1 + w_2x_2 + \dots + w_nx_n = \mathbf{x}^T \mathbf{w}$), lalu jumlah bobot tersebut dimasukkan ke dalam *step function* sehingga menghasilkan keluaran seperti pada gambar 2.1 [9].



Gambar 2.1 *Threshold logic unit*: Sebuah neuron buatan yang menghitung jumlah bobot dari setiap masukan lalu diterapkan ke dalam *step function* [9]

2.1.2.2 *Multi Layer Perceptron (MLP)*

Multi Layer Perceptron (MLP) adalah salah satu contoh arsitektur ANN lainnya yang memiliki banyak lapis. MLP memiliki lapisan yang tersembunyi, terletak diantara lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*), disebut *hidden layer*. Proses yang terjadi pada lapisan masukan dan lapisan keluaran terlihat dari luar arsitektur, di sisi lain proses yang terjadi pada lapisan tersembunyi tidak bisa dilihat dari luar. Arsitektur MLP dapat dilihat pada gambar 2.2 [8].



Gambar 2.2 Contoh arsitektur *Multi Layer Perceptron (MLP)*

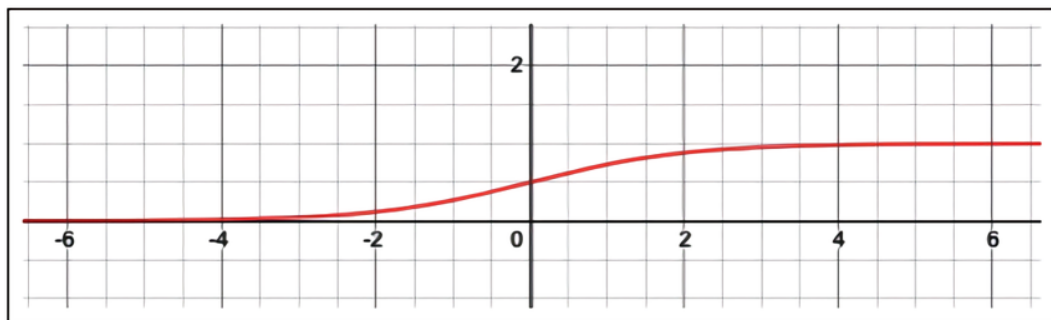
Lingkaran yang terdapat pada gambar di atas disebut node atau neuron buatan. Konsep neuron buatan pada MLP sama dengan konsep pada *perceptron*. Neuron tersebut menghitung jumlah bobot dari setiap masukan lalu diterapkan ke dalam *step function*. Setelah ada penelitian tentang algoritme *backpropagation*, *step function* diganti menjadi fungsi *sigmoid*. *step function* dan fungsi *sigmoid* disebut dengan fungsi aktivasi [8].

2.1.3 Fungsi Aktivasi

Fungsi aktivasi adalah rangkaian rumus matematika yang mengubah suatu nilai. Fungsi aktivasi dalam ANN dipakai ketika jumlah bobot yang sudah ditambahkan bias diteruskan ke lapisan selanjutnya. Fungsi aktivasi memastikan adanya nonlinearitas di antara lapisan *neural network*. Jika pada setiap lapisan *neural network* tidak mempunyai nonlinieritas, maka sebanyak apa pun lapisan tersembunyi yang digunakan setara dengan hanya memakai satu lapisan tersembunyi. Fungsi aktivasi pada *neural network* yang digunakan dalam arsitektur GRU adalah fungsi *sigmoid* dan fungsi *hyperbolic tangent* (tanh) [9].

2.1.3.1 Fungsi Sigmoid

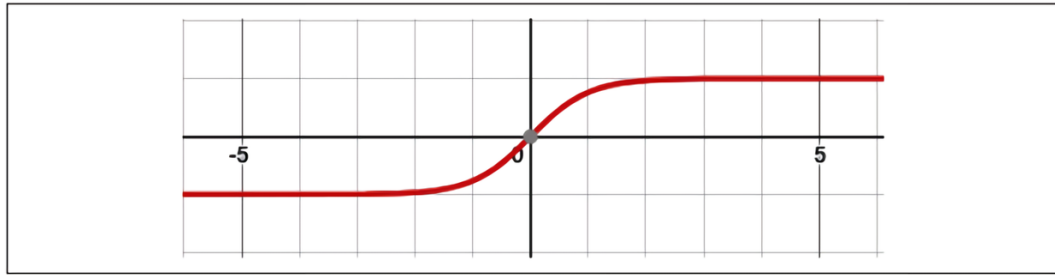
Fungsi *Sigmoid* adalah fungsi aktivasi yang menghasilkan nilai dengan skala 0 sampai 1. Fungsi *Sigmoid* menerima masukan dari skala negatif tak hingga sampai tak hingga. Fungsi *sigmoid* didefinisikan dengan persamaan $\sigma(x) = \frac{1}{1+e^{-x}}$. Fungsi *sigmoid* merupakan fungsi aktivasi nonlinier. Tipikal fungsi *Sigmoid* direpresentasikan dalam gambar 2.3 [8].



Gambar 2.3 Fungsi *sigmoid* dengan keluaran dalam skala 0 sampai 1 [8]

2.1.3.2 Fungsi Hyperbolic Tangent (tanh)

Fungsi *Hyperbolic Tangent* (tanh) adalah fungsi aktivasi nonlinier yang menghasilkan nilai dengan skala -1 sampai 1. Skala -1 sampai 1 membuat keluaran setiap lapisan kurang lebih berpusat di sekitar 0 di awal pelatihan, yang sering membantu mempercepat konvergensi. Fungsi tanh menerima masukan dari skala negatif tak hingga sampai tak hingga. Fungsi tanh memiliki persamaan $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. Fungsi tanh direpresentasikan dalam gambar 2.4 [8][9].



Gambar 2.4 Fungsi *Hyperbolic Tangent* (tanh) dengan keluaran dalam skala -1 sampai 1 [8]

2.1.4 *Recurrent Neural Network* (RNN)

Recurrent Neural Network (RNN) merupakan salah satu kelas dari *neural network*. RNN lebih populer digunakan untuk masukan berupa teks. RNN memanfaatkan masukan yang memiliki karakteristik berurutan. Masukan yang memiliki karakteristik berurutan memiliki beberapa bentuk, yaitu teks, ucapan, dan deret waktu. Masukan yang memiliki karakteristik berurutan adalah data berurutan yang bergantung atau memiliki hubungan dengan data sebelumnya [8].

Tradisional MLP membuat asumsi bahwa semua masukan tidak memiliki hubungan dengan masukan lainnya. Asumsi ini tidak berlaku pada banyak tipe data berurutan. Contohnya, kata dalam kalimat, notasi musik dalam komposisi, harga saham dari waktu ke waktu, atau bahkan molekul dalam suatu senyawa. Oleh karena itu, RNN membuat *hidden state* atau memori yang menyimpan esensial dari rentetan masukan yang terjadi sampai suatu waktu. Perhitungan dasar yang terjadi pada *hidden state* atau memori diperlihatkan pada persamaan berikut [8] [9].

$$h_t = f(h_{t-1}, X_t) \quad (2.1)$$

Keterangan :

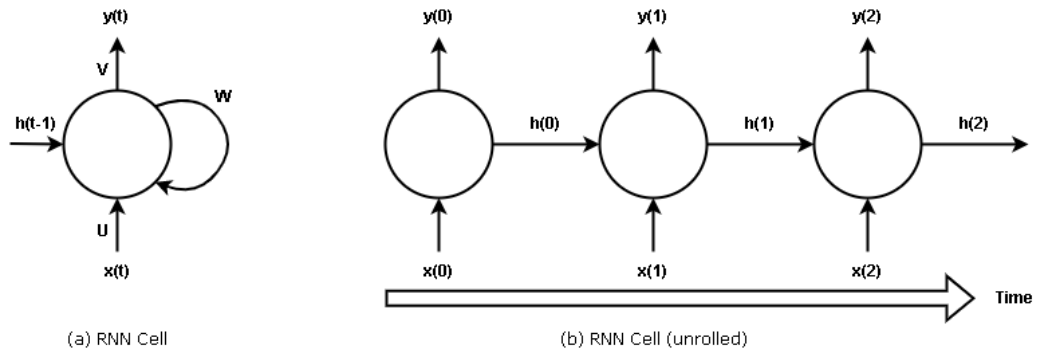
h_t : Nilai *hidden state* pada waktu t

h_{t-1} : Nilai *hidden state* pada waktu t-1

X_t : Nilai masukan pada waktu t

f : Fungsi untuk menghitung Nilai *hidden state* pada waktu t

RNN menggunakan tiga matriks bobot U, V, dan W sebagai parameter. Bobot U adalah bobot dari masukan, bobot V adalah bobot dari keluaran, dan bobot W adalah bobot dari *hidden state* atau memori.



Gambar 2.5 (a) sistematik dari sebuah sel RNN; (b) sel RNN (*unrolled*)

Gambar 2.5(b) adalah bentuk lengkap dari gambar 2.5(a) jika bagian pengulangannya dijabarkan. Gambar 2.5(b) mempunyai tiga langkah waktu. Bobot U , V , dan W dibagikan pada setiap langkah waktu tersebut. $y(t)$ merupakan keluaran pada waktu t yang diwakili oleh nilai $h(t)$ atau *hidden state* pada waktu t . Perhitungan $y(t)$ dapat dilihat pada persamaan 2.3. *Hidden state* pada waktu t digunakan pada perhitungan *hidden state* pada waktu $t+1$. Perhitungan tersebut dapat dilihat pada persamaan 2.2 [8] [9].

$$h_t = \tanh(W h_{t-1} + U x_t) \quad (2.2)$$

$$y_t = \text{softmax}(V h_t) \quad (2.3)$$

Keterangan :

- h_t : Nilai *hidden state* pada waktu t
- h_{t-1} : Nilai *hidden state* pada waktu $t-1$
- W : Nilai bobot dari *hidden state* pada waktu $t-1$
- U : Nilai bobot dari masukan pada waktu t
- x_t : Nilai masukan pada waktu t
- \tanh : Fungsi aktivasi *Hyperbolic Tangen*
- y_t : Nilai Keluaran pada waktu t
- V : Nilai bobot dari *hidden state* pada waktu t
- softmax : Fungsi aktivasi *softmax*

2.1.5 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) adalah salah satu alat ukur kinerja sebuah kasus regresi. RMSE memberi gambaran tentang bagaimana banyak kesalahan yang dibuat sistem dalam hasil prediksinya, dengan memberi bobot yang lebih tinggi untuk kesalahan besar. Persamaan 2.4 memperlihatkan rumus matematika untuk menghitung nilai RMSE [9].

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (2.4)$$

Keterangan :

m : Jumlah *instance* atau baris data dalam kumpulan data.

$x^{(i)}$: Vektor dari semua nilai fitur (tidak termasuk label) dari *instance* atau baris data ke-i dalam kumpulan data

$y^{(i)}$: Vektor berisi label (nilai keluaran yang diinginkan untuk baris ke-i)

X : Matriks yang berisi semua nilai fitur (tidak termasuk label) dari semua baris dalam kumpulan data.

h : Fungsi prediksi sistem atau hipotesis.

$RMSE(X, h)$: Fungsi biaya yang diukur pada kumpulan data menggunakan hipotesis h

2.1.6 Normalisasi *Min-max*

Normalisasi *Min-max* adalah salah satu teknik untuk mengubah nilai dari fitur yang dimiliki. Pada Normalisasi *Min-max*, nilai digeser dan dihitung ulang sehingga menghasilkan skala nol hingga satu. Normalisasi *Min-max* mengurangi nilai x dengan nilai minimum dan membaginya dengan nilai maksimum dikurangi dengan nilai minimum seperti pada persamaan 2.5. *Library Scikit-Learn* menyediakan fungsi untuk menghitung normalisasi *Min-max* yang disebut *MinMaxScaler*. Namun, penelitian ini membuat normalisasi *Min-max* dari awal [9].

$$x_{scaled} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2.5)$$

Keterangan :

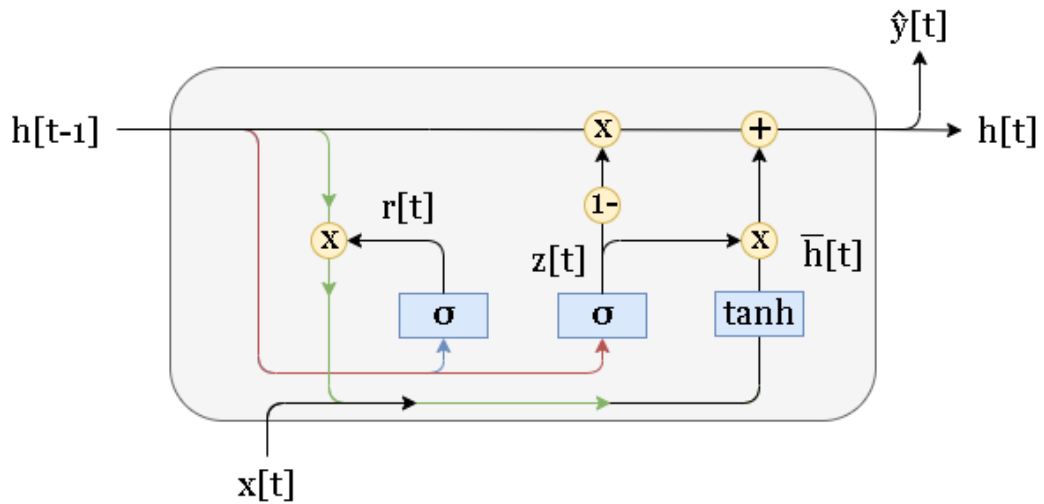
- x_{scaled} : Hasil normalisasi sebuah nilai x
- x_i : Nilai ke- i dari kumpulan data yang ingin dinormalisasi.
- $\min(x)$: nilai minimum dari kumpulan data
- $\max(x)$: nilai maksimum dari kumpulan data

2.1.7 *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit (GRU) adalah varian *gated RNN* selain dari *Long Short-Term Memory (LSTM)*. GRU pertama kali diperkenalkan oleh Kyunghyun Cho pada tahun 2014. Perbedaan utama GRU dengan LSTM adalah bahwa satu unit gerbang secara bersamaan mengontrol faktor lupa dan keputusan untuk memperbarui unit dalam *hidden state*. Dengan kata lain GRU menggabungkan proses untuk menentukan faktor unit yang ingin dibuang dan unit mana yang ingin diperbarui ke dalam satu *gate* yang disebut *update gate* [8][9].

Buku yang ditulis oleh Antonio Gulli menyatakan bahwa GRU dan LSTM menyelesaikan masalah *Vanishing gradient* tapi GRU lebih cepat saat dilatih karena komputasinya lebih sedikit [8]. Menurut buku yang ditulis Aurelien Geron, GRU mempunyai performa yang sama bagusnya dengan LSTM [9]. Tujuan utama GRU adalah untuk menangkap dependensi atau faktor yang berpengaruh terhadap hasil prediksi di waktu yang berbeda dengan perubahan yang adaptif. Sebagai contoh, manusia tidak membutuhkan semua informasi masa lalu untuk membuat keputusan. Jika seseorang merencanakan untuk pergi liburan, maka informasi masa lalu seperti nilai ujian kemarin tidak banyak memengaruhi keputusan untuk tempat tujuan pergi liburan.

GRU menggunakan tipe *hidden unit* baru yang terinspirasi dari LSTM. Di dalam struktur GRU, terdapat komponen yang disebut *gate* berfungsi untuk mengatur alur informasi model GRU. GRU hanya mempunyai dua *gate*, yaitu *reset gate* dan *update gate*. *Reset gate* pada GRU menggabungkan *input* baru dengan informasi masa lalu. *Update gate* menentukan berapa banyak informasi masa lalu yang harus tetap disimpan [8].



Gambar 2.6 Struktur Gated Recurrent Unit

Keterangan :	
$r[t]$: Reset gate
$z[t]$: Update gate
$\bar{h}[t]$: New remember atau hidden state
$h[t]$: Hidden state akhir
$y[t]$: output pada waktu t
$x[t]$: Input baru
$h[t - 1]$: Informasi masa lalu

Dalam GRU hal pertama yang terjadi adalah *reset gate* dikomputasi untuk menggabungkan informasi masa lalu dan input baru dengan rumus sebagai berikut:

$$r_t = \sigma(W_r h_{t-1} + U_r x_t + b_r) \quad (2.6)$$

Di mana σ adalah fungsi aktivasi *sigmoid*, b_r adalah bias dan W_r, U_r adalah weight dari persamaan tersebut. Setelah *reset gate*, komputasi dilanjutkan ke *new remember* atau *hidden state* dengan rumus sebagai berikut:

$$\bar{h}[t] = \tanh(W(r_t * h_{t-1}) + U x_t + b) \quad (2.7)$$

Setelah itu masuk ke dalam *update gate* untuk menentukan berapa banyak informasi masa lalu yang harus tetap disimpan dengan rumus sebagai berikut:

$$z_t = \sigma(W_z h_{t-1} + U_z x_t + b_z) \quad (2.8)$$

Terakhir, *hidden state* final dihitung dengan rumus sebagai berikut:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \bar{h}[t] \quad (2.9)$$

Dari *hidden state* diakhiri dengan penentuan hasil prediksi [8].

2.1.8 *Library* yang digunakan

Subbab ini menjelaskan tentang *library* apa saja yang digunakan dalam penelitian ini. Subbab ini menjelaskan latar belakang *library* yang digunakan. Subbab ini juga menjelaskan tentang fungsi yang digunakan pada setiap *library*.

2.1.8.1 *Library* Pandas

Library Pandas adalah *library* yang dipakai untuk menganalisis sebuah data. Pada penelitian ini *Library* Pandas digunakan untuk memuat data dari format tertentu menjadi *DataFrame*. Selain itu, *Library* Pandas membuat pengolahan data menjadi lebih mudah. Fungsi-fungsi yang digunakan pada *Library* Pandas dapat dilihat pada tabel 2.1 [10].

Tabel 2.1 Tabel Fungsi *Library* Pandas

No	Fungsi	Masukan	Keluaran	Keterangan
1	read_csv	path: string	DataFrame	Memuat data yang disimpan dalam format <i>Comma-separated values</i> (csv) menjadi pandas DataFrame.

2	drop	label: string, axis: int, inplace: boolean	DataFrame atau none jika inplace = true	menghapus baris atau kolom berdasarkan parameter dan mengembalikan hasil dalam bentuk DataFrame.
3	describe	-	DataFrame atau series	Memberi informasi rangkuman statistik pada sebuah DataFrame atau series.

2.1.8.2 Library NumPy

NumPy adalah singkatan dari *Numerical Python*. *Library NumPy* adalah *library Python open source* yang digunakan pada bidang sains dan teknik. *Library NumPy* merupakan standar universal yang dipakai untuk mengolah data numerik dengan Python. Pada penelitian ini *Library NumPy* digunakan untuk mengubah bentuk sebuah himpunan lama menjadi himpunan baru tanpa mengubah data. Fungsi-fungsi yang digunakan pada *Library NumPy* dapat dilihat pada tabel 2.2 [11].

Tabel 2.2 Tabel Fungsi *Library Numpy*

No	Fungsi	Masukan	Keluaran	Keterangan
1	reshape	a: array, newshape: int atau tuple of ints	ndarray	Mengubah bentuk sebuah himpunan lama menjadi himpunan baru tanpa mengubah data.

2.1.8.3 *Library Matplotlib*

Library Matplotlib digunakan untuk melakukan visualisasi terhadap nilai yang dihasilkan oleh model pada penelitian ini. Antarmuka yang kita pakai untuk penelitian ini adalah antarmuka *Pyplot* yang disediakan oleh *Library Matplotlib*. Fungsi-fungsi yang digunakan pada *Library Matplotlib* dapat dilihat pada tabel 2.3 [12].

Tabel 2.3 Tabel Fungsi *Library Matplotlib.Pyplot*

No	Fungsi	Masukan	Keluaran	Keterangan
1	figure	figsize: (float, float)	figure	Membuat figur baru.
2	title	label: string, loc: string ('center', 'left', 'right')	text	Memberi judul pada plot.
3	plot	x,y: array angka	list of lines	Membuat plot dari kumpulan data kedalam sumbu x dan sumbu y.
4	legend	handles: list of lines, labels: list of string, loc: string	legend	Memberi legenda pada hasil plot data.
5	show	-	figure	Menampilkan figur yang telah dibuat.

2.1.8.4 *Library Keras*

Library utama yang digunakan pada penelitian ini adalah *Library Keras*. *Library Keras* digunakan untuk membangun model GRU, melakukan proses pelatihan, melakukan proses pengujian dan mengevaluasi hasil nilai model. Fungsi-fungsi yang digunakan pada *Library Keras* dapat dilihat pada tabel 2.4 [13].

Tabel 2.4 Tabel Fungsi *Library Keras*

No	Fungsi	Masukan	Keluaran	Keterangan
1	dense	units: positive int, activation: string	N-D tensor	Sebuah lapisan dense yang dapat ditambahkan pada sebuah model.
2	sequential	list of keras.layer	-	Membuat model dengan lapisan yang tersusun secara berurutan.
3	input	shape: tuple of int	tensor	Sebuah objek untuk mengatur inisialisasi masukan dalam sebuah model.
4	add	keras.layer	-	Menambahkan sebuah lapisan kedalam model.
5	compile	loss: string	-	Konfigurasi model untuk pelatihan.
6	fit	x: input data, y: target data, batch_size: int, epochs: int	history object	Melakukan pelatihan model terhadap dataset sesuai epoch yang ditentukan.
7	save	path: string	-	Menyimpan model ke dalam sebuah direktori.

8	evaluate	x: input data, y: target data	scalar atau list of scalar	Mengembalikan nilai loss berdasarkan hasil percobaan.
9	predict	x: input data	NumPy array hasil prediksi	Mengembalikan hasil prediksi berdasarkan sample data masukan.

2.1.8.5 Gated Recurrent Unit (GRU) dalam Library Keras

Salah satu cara untuk mengimplementasi GRU ke dalam sebuah model adalah dengan menggunakan *library* Keras. GRU dalam *Library* Keras memiliki banyak parameter. Berikut penjelasan tiap parameter yang tersedia:

1. **units:** Integer positif, dimensi dari keluaran. Units digunakan untuk menentukan berapa banyak GRU cell dalam satu lapisan. Nilai unit tidak bisa diisi kurang dari 0 [13].
2. **activation:** Fungsi aktivasi yang digunakan. Secara *default*, *activation* berisi fungsi aktivasi tanh. *Activation* ini digunakan pada $\bar{h}[t]$. Keras menyediakan pilihan fungsi aktivasi lain seperti relu, sigmoid, softmax, softplus, softsign, selu, elu, dan exponential [13].
3. **recurrent_activation:** Fungsi aktivasi yang digunakan dalam proses recurrent step. Secara *default* *activation* berisi fungsi aktivasi sigmoid. *Recurrent_activation* digunakan pada *reset gate* dan *update gate*. Keras menyediakan pilihan fungsi aktivasi lain seperti relu, tanh, softmax, softplus, softsign, selu, elu, dan exponential [13].
4. **use_bias:** Boolean. *use_bias* memiliki nilai *default True*. *use_bias* menentukan apakah layer GRU menggunakan bias atau tidak [13].
5. **kernel_initializer:** Inisialisasi untuk kernel matriks bobot pada input. Nilai *default* dari parameter ini adalah *glorot_uniform*. Glorot uniform disebut juga Xavier uniform. bobot terdistribusi diantara *-limit* sampai *limit*. *Limit* memiliki persamaan $limit = \sqrt{6/(fan_in + fan_out)}$. *fan_in* adalah jumlah input unit (fitur) dan *fan_out* jumlah output unit. Keras menyediakan pilihan *weight initializer* lain seperti RandomNormal, RandomUniform, TruncatedNormal, Zeros, Ones, GlorotNormal, HeNormal, HeUniform, Identity, Orthogonal, Constant, dan VarianceScaling [13].

6. `recurrent_initializer`: Inisialisasi untuk kernel bobot matrix pada *recurrent state* atau bobot yang dipakai dalam *update gate*, *reset gate*, dan *hidden state* sebelumnya. Nilai *default* `recurrent_initializer` adalah *orthogonal*. *Orthogonal initializer* mengembalikan matriks *orthogonal*. Jika matriks yang *recurrent state* memiliki sedikit baris daripada kolom maka output akan memiliki baris *orthogonal*, begitu juga sebaliknya. Keras menyediakan pilihan *weight initializer* lain seperti *RandomNormal*, *RandomUniform*, *TruncatedNormal*, *Zeros*, *Ones*, *GlorotNormal*, *GlorotUniform*, *HeNormal*, *HeUniform*, *Identity*, *Constant*, dan *VarianceScaling* [13].
7. `bias_initializer`: Inisialisasi untuk bias. Nilai *default* parameter ini adalah *zeros*. Seperti namanya *zeros* berarti nilai awalnya adalah nol. Keras menyediakan pilihan *weight initializer* lain seperti *RandomNormal*, *RandomUniform*, *TruncatedNormal*, *Ones*, *GlorotNormal*, *GlorotUniform*, *HeNormal*, *HeUniform*, *Identity*, *Orthogonal*, *Constant*, dan *VarianceScaling* [13].
8. `kernel_regularizer`: Fungsi regularisasi yang diterapkan pada kernel bobot untuk input. Nilai *default* dari parameter ini adalah *none*. Nilai parameter ini diisi dengan *weight regularizers* yang disediakan Keras. Keras menyediakan pilihan *weight regularizers* seperti *L1*, *L2*, *L1L2*, dan *OrthogonalRegularizer* [13].
9. `recurrent_regularizer`: Fungsi regularisasi yang diterapkan pada kernel bobot untuk *recurrent state*. Nilai *default* dari parameter ini adalah *none*. Nilai parameter ini diisi dengan *weight regularizers* yang disediakan Keras. Keras menyediakan pilihan *weight regularizers* seperti *L1*, *L2*, *L1L2*, dan *OrthogonalRegularizer* [13].
10. `bias_regularizer`: Fungsi regularisasi yang diterapkan pada bias. Nilai *default* dari parameter ini adalah *none*. Nilai parameter ini diisi dengan *weight regularizers* yang disediakan Keras. Keras menyediakan pilihan *weight regularizers* seperti *L1*, *L2*, *L1L2*, dan *OrthogonalRegularizer* [13].
11. `activity_regularizer`: Fungsi regularisasi yang diterapkan pada hasil keluaran dari layer GRU. Nilai *default* dari parameter ini adalah *none*. Nilai parameter ini diisi dengan *weight regularizers* yang disediakan Keras. Keras menyediakan pilihan *weight regularizers* seperti *L1*, *L2*, *L1L2*, dan *OrthogonalRegularizer* [13].
12. `kernel_constraint`: *Constraint function* yang diterapkan pada kernel untuk bobot input. Nilai *default* dari parameter ini adalah *none*. Parameter ini berguna untuk membatasi nilai bobot input sesuai aturan fungsi batasannya. Keras menyediakan pilihan *Constraint function* seperti *MaxNorm*, *MinMaxNorm*, *NonNeg*, *UnitNorm*, dan *RadialConstraint* [13].
13. `recurrent_constraint`: *Constraint function* yang diterapkan pada kernel untuk

- bobot *recurrent state*. Nilai *default* dari parameter ini adalah *none*. Parameter ini berguna untuk membatasi nilai bobot *recurrent state* sesuai aturan fungsi batasannya. Keras menyediakan pilihan *Constraint function* seperti *MaxNorm*, *MinMaxNorm*, *NonNeg*, *UnitNorm*, dan *RadialConstraint* [13].
14. *bias_constraint*: *Constraint function* yang diterapkan pada kernel untuk bias. Nilai *default* dari parameter ini adalah *none*. Parameter ini berguna untuk membatasi nilai bias sesuai aturan fungsi batasannya. Keras menyediakan pilihan *Constraint function* seperti *MaxNorm*, *MinMaxNorm*, *NonNeg*, *UnitNorm*, dan *RadialConstraint* [13].
 15. *dropout*: *Dropout* adalah parameter untuk mengeliminasi input dalam GRU cell. Nilai parameter ini diisi dengan angka nol sampai satu. Nilai *default* parameter *dropout* adalah nol [13].
 16. *recurrent_dropout*: *Recurrent_dropout* adalah parameter untuk mengeliminasi *recurrent state* dalam GRU cell. Nilai parameter ini diisi dengan angka nol sampai satu. Nilai *default* parameter *dropout* adalah nol [13].
 17. *return_sequences*: Boolean. Parameter ini berfungsi untuk menentukan hasil yang dikembalikan GRU dalam bentuk urutan keseluruhan saja atau hasil akhirnya saja. *Default* dari parameter ini adalah *False* [13].
 18. *return_state*: Boolean. Parameter ini berfungsi untuk menentukan GRU mengembalikan kondisi akhir dari hasil keluaran atau tidak. *Default* dari parameter ini adalah *False* [13].
 19. *go_backwards*: Boolean dengan nilai *default False*. Nilai *True* pada parameter ini menandakan proses urutan input terbalik [13].
 20. *stateful*: Boolean dengan nilai *default False*. Jika parameter ini bernilai *True* maka state terakhir pada batch akan dijadikan nilai inisialisasi pada batch selanjutnya [13].
 21. *unroll*: Boolean dengan nilai *default False*. Parameter ini menentukan apakah GRU menerapkan *unrolling RNN* atau tidak. *Unrolling RNN* mempercepat proses namun dibutuhkan memory yang besar. *Unrolling RNN* cocok untuk data jangka pendek [13].
 22. *time_major*: Boolean dengan nilai *default False*. Parameter ini berfungsi untuk menentukan bentuk input. Jika *False* maka bentuk input harus dalam bentuk [timesteps, batch, feature], jika *True* maka bentuk input harus dalam bentuk [batch, timesteps, feature] [13].
 23. *reset_after*: Boolean dengan nilai *default True*. Parameter ini berfungsi untuk menentukan GRU menggunakan variasi model baru atau lama. Nilai *True* berarti GRU memakai variasi lama, variasi lama digunakan sebagai default

karena cocok untuk pemrosesan menggunakan GPU [13].

2.2 Tinjauan Studi

Tabel 2.5 menunjukkan studi terkait dalam penelitian ini. Terdapat banyak model yang sudah diteliti sampai saat ini. Pada penelitian yang dilakukan oleh Sethia A., Raut P. menghasilkan kesimpulan bahwa GRU dan LSTM memiliki performa yang lebih bagus dibandingkan algoritme MLP dan SVM. GRU memiliki *return ratio* yang lebih baik dibandingkan dengan LSTM. *Return ratio* menunjukkan bahwa model tersebut memprediksi pergerakan saham dengan lebih baik. Oleh karena itu, penelitian ini menggunakan algoritme GRU.

Tabel 2.5 Tinjauan Studi

No	Peneliti	Data Preprocessing	Algoritme	Hasil
1	Y. Ji, A. W. - C. Liew and L. Yang. (2021) [14]	-	<i>Support Vector Regression (SVR), LSTM, Particle Swarm Optimization - Long Short-Term Memory (PSO-LSTM), Improved Particle Swarm Optimization - Long Short-Term Memory (IPSO-LSTM)</i>	Dengan <i>look-back</i> sebesar 20 hari, IPSO-LSTM mendapatkan <i>Mean Absolut Percentage Error (MAPE)</i> sebesar 0.72348, <i>Root Mean Square Error (RMSE)</i> sebesar 67.322435, <i>Mean Absolute Error (MAE)</i> sebesar 43.133272, dan R^2 sebesar 0.975952.

2	S. Chen and C. Zhou. (2021) [6]	<i>Genetic Algorithm</i> (GA)	LSTM	GA-LSTM(K=10) paling optimal (K adalah jumlah fitur yang diurutkan berdasarkan nilai tertinggi oleh algoritme GA). GA-LSTM(K=10) mendapat nilai <i>Mean Square Error</i> (MSE) sebesar 0.0047 pada <i>dataset</i> China Construction Bank.
3	R. Ren, D. D. Wu and T. Liu. (2019) [5]	-	<i>Sentiment Analysis</i> dan SVM	Percobaan pertama yang dilakukan dengan menggunakan data SSE 50 Index menghasilkan akurasi sebesar 71,33%. Percobaan kedua yang dilakukan dengan menggunakan hasil <i>Sentiment Analysis</i> dan data SSE 50 Index menghasilkan akurasi sebesar 89,93%.

4	Sethia A., Raut P. (2019) [7]	<i>Z-score</i> , Normalisasi <i>Min-max</i> dan <i>Independent Components Analysis</i> (ICA)	LSTM, GRU, SVM, dan Multi Layer Perceptron	LSTM mendapatkan RMSE sebesar 0.000428 dan <i>return ratio</i> sebesar 4.308454. GRU mendapatkan RMSE sebesar 0.000511 dan <i>return ratio</i> sebesar 5.722242. SVM mendapatkan RMSE sebesar 0.000543 dan <i>return ratio</i> sebesar -1.858130. MLP mendapatkan RMSE sebesar 0.001052 dan <i>return ratio</i> sebesar 2.478719.
5	Rahman, M.O., Hossain, M.S., Junaid, T.S., Forhad, M.S.A. and Hossen, M.K. (2019) [15]	Normalisasi <i>Min-max</i>	<i>Gated Recurrent Unit</i> (GRU)	RMSE yang didapat dengan menggunakan data perusahaan Lowe adalah 0.0127464. RMSE yang didapat dengan menggunakan data perusahaan Coca-Cola adalah 0.0144508. RMSE yang didapat dengan menggunakan data perusahaan apple adalah 0.013996.

Pada penelitian yang dilakukan oleh Sethia A. dan Rahman, normalisasi *Min-max* digunakan untuk mengubah skala kumpulan data pada tahap *data preprocessing*. Normalisasi *Min-max* menghasilkan skala nol sampai satu yang cocok untuk data saham karena saham tidak mungkin lebih kecil dari nol dan

mempermudah proses perhitungan dalam model GRU. Penelitian ini juga menggunakan normalisasi *Min-max* digunakan untuk mengubah skala kumpulan data pada tahap *data preprocessing*. Penelitian yang dilakukan Rahman menunjukkan *batch size* dan *epoch* berpengaruh terhadap hasil prediksi. Karena itu, penelitian ini juga menguji pengaruh *batch size* dan *epoch* terhadap hasil prediksi model. Semua penelitian pada tabel 2.5 menggunakan jumlah *hidden layer* (unit) yang berbeda. Oleh karena itu, penelitian ini menguji pengaruh unit terhadap hasil prediksi.

2.3 Tinjauan Objek

Pada penelitian ini, objek yang ingin diteliti adalah perubahan data harga saham pada Bursa Efek Indonesia. Menurut Kamus Besar Bahasa Indonesia (KBBI) saham adalah surat bukti pemilikan bagian modal perseroan terbatas yang memberi hak atas dividen dan lain-lain menurut besar kecilnya modal yang disetor. Saham juga berarti hak yang dimiliki orang (pemegang saham) terhadap perusahaan berkat penyerahan bagian modal sehingga dianggap berbagi dalam pemilikan dan pengawasan. Harga saham menentukan seberapa besar pemegang saham mendapatkan bagian dari nilai suatu perusahaan. Oleh karena itu, harga saham pada Bursa Efek Indonesia menjadi objek yang menarik untuk diteliti.

2.3.1 Jenis-jenis Saham

Saham memiliki banyak jenis. Menurut artikel ilmiah yang ditulis oleh Setiani Widiarti, Saham dibagi ke dalam 3 aspek seperti berikut.

2.3.1.1 Berdasarkan Hak Kepemilikan Saham

Saham dibagi menjadi dua jenis jika dilihat berdasarkan hak kepemilikan saham, yaitu saham biasa (*common stock*) dan saham preferen (*preferred stock*). Saham biasa mewakili klaim kepemilikan terhadap penghasilan dan aktiva (aset) yang dimiliki perusahaan. Pemegang saham biasa ini memiliki kewajiban yang terbatas. Jika perusahaan yang dipegang investor saham biasa bangkrut, kerugian maksimum yang ditanggung investor sama besarnya dengan nilai yang sudah diinvestasikan pada saham perusahaan tersebut [1].

Saham preferen dikatakan memiliki karakteristik obligasi dan saham biasa. Saham preferen secara keseluruhan mirip seperti saham biasa tetapi sekuritas ini memberikan tingkat pendapatan tetap seperti halnya obligasi. Pemegang saham preferen Pemegang saham preferen mungkin tidak bisa menerima pembayaran dividen dalam waktu yang sudah ditetapkan sebelumnya jika emiten mengalami

kerugian [1].

2.3.1.2 Berdasarkan Cara Peralihan Saham

Saham ditinjau dari cara peralihannya dibagi menjadi dua jenis yaitu saham atas unjuk (*bearer stocks*) dan saham atas nama (*registered stocks*). Saham atas unjuk adalah saham yang nama pemiliknya tidak dicantumkan dalam saham tersebut sehingga saham ini mudah untuk dialihkan ke orang lain. Saham atas unjuk bertujuan untuk mempermudah proses jual beli. Saham atas nama adalah kebalikan dari saham atas unjuk. Saham atas nama memiliki nama kepemilikan yang tertulis di surat berharga. Saham atas nama harus melalui prosedur hukum untuk melakukan balik nama saham saat ingin mengganti kepemilikan saham tersebut [1].

2.3.1.3 Berdasarkan Kinerja Perdagangan

Saham dibagi menjadi lima jenis jika ditinjau dari kinerja perdagangan, yaitu *blue-chip stocks*, *income stocks*, *growth stocks*, *speculative stocks*, *counter cyclical stocks*. *Blue-chip stocks* merupakan saham biasa dari emiten yang memiliki reputasi tinggi. emiten yang mengeluarkan *blue-chip stocks* memiliki pendapatan yang stabil dan konsisten dalam membayar dividen [1].

Income stocks adalah saham yang membayar dividen lebih besar dari rata-rata dividen yang diberikan periode sebelumnya. Perusahaan yang mengeluarkan *income stocks* mampu meningkatkan keuntungan tiap periode. *Growth stocks* dibagi lagi menjadi dua jenis yaitu *well-known* dan *Lesser-known*. *Well-known* adalah saham dari perusahaan yang memiliki pertumbuhan pendapatan yang tinggi. *Lesser-known* adalah saham yang dikeluarkan perusahaan yang pertumbuhan pendapatannya tidak tinggi tapi mempunyai pertumbuhan dalam pendapatannya [1].

Speculative stocks adalah saham dari suatu perusahaan yang memiliki kemungkinan menghasilkan keuntungan yang tinggi namun belum pasti. emiten yang mengeluarkan *Speculative stocks* tidak bisa memperoleh keuntungan konsisten tiap tahunnya. *Counter cyclical stocks* adalah saham yang tidak terpengaruh oleh kondisi ekonomi makro maupun situasi bisnis secara umum. *Counter cyclical stocks* adalah saham yang memiliki kondisi paling stabil [1].

2.3.2 Teori Prediksi Saham

Terdapat tiga tipe cara manusia memprediksi harga saham, yaitu *technical analysis*, *fundamental analysis*, dan *gambler* (asal tebak). Mental atau emosi dari

manusia itu sendiri juga termasuk salah satu faktor yang menentukan keputusan dalam menjual atau membeli saham. Jika seorang investor mempunyai ekspektasi bahwa suatu saham akan naik, maka investor itu akan membeli saham tersebut. Sebaliknya jika seorang investor mempunyai ekspektasi bahwa suatu saham akan turun, maka investor itu akan menjual saham tersebut [16].

Technical analysis adalah pengetahuan tentang harga dengan grafik sebagai alat utamanya. *Technical analysis* mengandalkan pola data atau pola grafik dalam memprediksi harga saham. *Technical analysis* berbicara tentang sifat tren harga, konfirmasi dan divergensi, volume yang mencerminkan perubahan harga, dan istilah *support/resistance* [16].

Buku yang ditulis oleh Steven B. Achelis menggambarkan bahwa harga saham adalah hasil dari pertarungan satu lawan satu antara pembeli dan penjual. Pembeli menekan harga menjadi lebih tinggi dan penjual menekan harga lebih rendah (mengikuti prinsip *supply and demand*). Arah harga naik atau turun adalah hasil dari siapa yang menang dalam pertarungan tersebut. Arah naik atau turunnya pergerakan saham dalam periode tertentu biasa disebut dengan tren [16].

Support adalah nilai batas atau area di mana harga saham berada di bawah. Area *support* tercipta karena adanya dukungan dari pembeli sehingga harga saham tidak jatuh terlalu jauh. Selain karena dukungan dari pembeli, area *support* juga tercipta karena penjual tidak mau menjual sahamnya di bawah harga tersebut. Pada area *support*, jumlah pembeli lebih banyak dari jumlah penjual sehingga nilai saham mengalami peningkatan. Area *support* yang kuat menyebabkan nilai saham sulit untuk melewati batas atau turun lebih jauh [16].

Resistance adalah nilai batas atau area di mana harga saham berada di atas. Area *resistance* tercipta karena adanya pengaruh dari penjual terhadap saham tersebut. Penjual mencegah harga saham menjadi terlalu mahal yang menyebabkan pembeli tidak mau membeli saham tersebut. Pada area *resistance*, jumlah penjual lebih banyak dari jumlah pembeli sehingga nilai saham mengalami penurunan. Area *resistance* yang kuat menyebabkan nilai saham sulit untuk melewati batas atau naik lebih jauh [16].

Area *support* dan *resistance* menahan harga saham agar tidak melewati batas area tersebut. Akan tetapi, tidak menutup kemungkinan harga saham untuk menembus batas area *support* atau *resistance*. Peristiwa di mana harga saham menembus batas area *support* atau *resistance* disebut *breakout*. Ketika terjadi *breakout*, area *support* yang lama menjadi area *resistance* yang baru, berlaku

sebaliknya [16].

Fundamental analysis adalah cara memprediksi saham berdasarkan laporan keuangan triwulan perusahaan terkait atau berita yang berhubungan dengan perusahaan terkait. Jika ada berita yang negatif tentang perusahaan terkait, maka investor tidak membeli saham perusahaan tersebut. Jika terjadi kenaikan keuntungan pada laporan keuangan perusahaan terkait, maka investor membeli saham perusahaan tersebut [16].

Gambler adalah salah satu cara manusia untuk memprediksi harga saham akan naik atau turun. Tidak sedikit investor yang menggunakan cara ini. Ketika investor membeli sebuah saham atau sekuriti, investor tersebut tidak tahu harga saham tersebut akan naik atau turun. Tetapi jika investor tersebut membeli saham ketika tren harga sedang naik, maka kemungkinan investor tersebut mendapat keuntungan akan meningkat [16].

Berdasarkan teori Dow market dibagi menjadi tiga tren. Tren tersebut, yaitu *Primary trend*, *Secondary trends*, dan *Minor trends*. *Primary trend* bertahan satu tahun sampai beberapa tahun. Artinya, harga saham konsisten naik atau turun dalam satu tahun sampai beberapa tahun. *Secondary trends* bertahan satu bulan sampai empat bulan. Artinya, harga saham konsisten naik atau turun dalam satu bulan sampai empat bulan. *Secondary trends* merupakan bentuk koreksi dari *Primary trend*. Artinya, *Secondary trends* terjadi setelah *Primary trend*. Jika *Primary trend* merupakan tren naik maka *Secondary trends* merupakan tren turun. Terakhir *Minor trends* adalah tren yang bertahan hanya satu hari sampai tiga minggu [16].

2.3.3 Penjelasan Tentang Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset". *Dataset* tersebut merupakan data runut waktu. *Dataset* tersebut berisi data pergerakan harga saham tiap harinya di Bursa Efek Indonesia. *Dataset* "Indonesia Stocks" memiliki 24 berkas data pergerakan saham perusahaan per hari. Data perusahaan yang digunakan dari *dataset* "Indonesia Stocks" adalah PT. Telkom, PT. Aneka Tambang, dan PT. Indofood.

Data dalam *dataset* "Dataset Saham Indonesia / Indonesia Stock Dataset" diambil melalui Yahoo Finance's public data dan website IDX. Pada *dataset* tersebut terdapat 782 berkas data pergerakan saham perusahaan per hari. Data

perusahaan yang digunakan dari *dataset* "Indonesia Stocks" adalah PT. XL Axiata, PT. Vale Indonesia, dan PT. Japfa Comfeed Indonesia.

Rentang waktu yang digunakan untuk penelitian ini adalah enam Juli 2015 sampai empat belas Oktober 2021. Format yang digunakan dalam *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset" adalah *Comma Separated Values* (CSV). Enam Kolom atau fitur yang dimiliki pada *dataset*, yakni tanggal, harga buka (open), harga tertinggi (high), harga terendah (low), harga tutup (close), volume pasar. Penelitian ini mengamati pergerakan saham PT. Telkom, PT. Aneka Tambang, PT. XL Axiata, PT. Vale Indonesia, PT. Japfa Comfeed Indonesia, dan PT. Indofood.

Perusahaan yang dipilih dibagi menjadi tiga sektor, yaitu pertambangan, makanan, dan telekomunikasi. Pada sektor pertambangan terdapat PT. Aneka Tambang sebagai objek utama dan PT. Vale Indonesia sebagai objek pembanding. PT. Aneka Tambang dan PT. Vale Indonesia memiliki sektor yang sama dan kedua perusahaan tersebut memperdagangkan emas. Pada sektor makanan terdapat PT. Indofood dan PT. Japfa Comfeed Indonesia. Pada sektor telekomunikasi terdapat PT. Telkom dan PT. XL Axiata. Karakteristik dari data perusahaan tersebut dapat dilihat pada tabel berikut.

Tabel 2.6 Tabel Karakteristik *Dataset* Utama

Perusahaan	Fitur	Min	Max	Mean
PT. Aneka Tambang	Open	290	3300	968
PT. Aneka Tambang	Close	287	3190	966
PT. Aneka Tambang	High	294	3440	989
PT. Aneka Tambang	Low	285	3060	948
PT. Aneka Tambang	Volume	1.37 juta	2.19 miliar	129 juta
PT. Indofood	Open	4680	9200	7080
PT. Indofood	Close	4680	9150	7080
PT. Indofood	High	4800	9200	7170
PT. Indofood	Low	4560	9050	6990
PT. Indofood	Volume	968 ribu	81.1 juta	9.13 juta
PT. Telkom	Open	2550	4800	3680
PT. Telkom	Close	2560	4800	3680

PT. Telkom	High	2590	4840	3730
PT. Telkom	Low	2450	4780	3640
PT. Telkom	Volume	20.7 juta	624 juta	100 juta

Tabel 2.7 Tabel Karakteristik *Dataset* Pemandangan

Perusahaan	Fitur	Min	Max	Mean
PT. Vale Indonesia	Open	1285	6875	3193
PT. Vale Indonesia	Close	1250	6725	3184
PT. Vale Indonesia	High	1345	7100	3253
PT. Vale Indonesia	Low	1190	6600	3127
PT. Vale Indonesia	Volume	1.37 juta	166 juta	15 juta
PT. Japfa Comfeed Indonesia	Open	299	3040	1447
PT. Japfa Comfeed Indonesia	Close	297	3050	1445
PT. Japfa Comfeed Indonesia	High	305	3100	1477
PT. Japfa Comfeed Indonesia	Low	296	2950	1416
PT. Japfa Comfeed Indonesia	Volume	3	315 juta	16 juta
PT. XL Axiata	Open	1410	4118	2866
PT. XL Axiata	Close	1410	4118	2862
PT. XL Axiata	High	1500	4216	2921
PT. XL Axiata	Low	1315	4073	2807
PT. XL Axiata	Volume	393 ribu	358 juta	16 juta

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

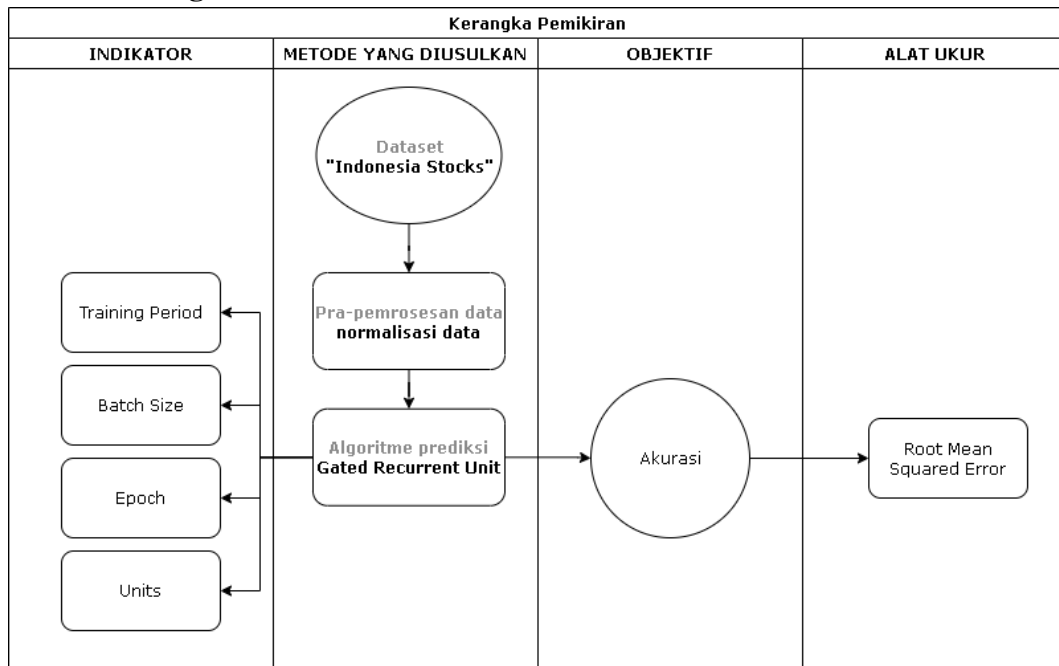
Bab ini menjelaskan tentang analisis masalah dan garis besar perancangan model *deep learning* yang dikembangkan. Garis besar perancangan model *deep learning* meliputi alur kerja pembuatan model *deep learning* dan hasil akhir yang ingin dicapai.

3.1 Analisis Masalah

Masalah pembuatan model prediksi dengan data runut waktu semakin meningkat jumlahnya dan semakin dibutuhkan. Model prediksi tersebut biasanya digunakan untuk memprediksi cuaca, angka kenaikan Covid-19, dan harga saham. Penelitian ini membangun model prediksi yang digunakan untuk memprediksi harga saham. Masalah yang dihadapi dalam membangun model prediksi harga saham adalah mencari parameter dan mencari arsitektur yang cocok untuk prediksi harga saham. Penelitian ini menguji akurasi yang dihasilkan model prediksi dengan arsitektur GRU. Penelitian ini juga menguji pengaruh parameter *training periode*, *unit*, *batch size*, dan *epoch* terhadap akurasi.

Masukan yang digunakan adalah data harga saham di Indonesia. Fitur *close* (harga tutup) dijadikan label. Data masukan melalui tahap data *Preprocessing* kemudian dilatih dengan model GRU. Keluaran model GRU adalah prediksi harga tutup. Keluaran dievaluasi menggunakan RMSE.

3.2 Kerangka Pemikiran



Gambar 3.1 Diagram Kerangka Pemikiran

Pertimbangan parameter pada bagian indikator:

1. *Training Period* adalah perlakuan berbeda pada saat pelatihan model khususnya pada periode yang digunakan. Data latihan dibagi menjadi dua yaitu, data dengan periode satu tahun dan data dengan periode lima tahun [15].
2. *Batch size* adalah hyperparameter yang digunakan untuk menentukan jumlah batch pada saat melatih model GRU. Menurut penelitian tahun 2019, *batch size* optimal berada pada angka 32 [15]. Penelitian ini menguji *batch size* sebesar 32, 64, dan 128.
3. *Epoch* adalah hyperparameter yang digunakan untuk menentukan jumlah iterasi pada saat melatih model GRU. Menurut penelitian tahun 2019, *epoch* memiliki hasil terbaik pada angka 100 [15]. Penelitian ini menguji *epoch* sebesar 10, 100, dan 1000.
4. *Units* adalah parameter yang digunakan saat membentuk model GRU. *Units* digunakan untuk menentukan jumlah *perceptron* pada satu layer. Penelitian ini menguji *units* sebesar 16, 32, 64, dan 128.

Bagian metode yang diusulkan menjelaskan metode-metode yang diterapkan untuk melakukan prediksi harga saham dalam penelitian ini. *Dataset* yang digunakan adalah *dataset "Indonesia Stocks"* dan "*Dataset Saham Indonesia / Indonesia Stock Dataset*" yang didapat dari situs Kaggle. Data dinormalisasi

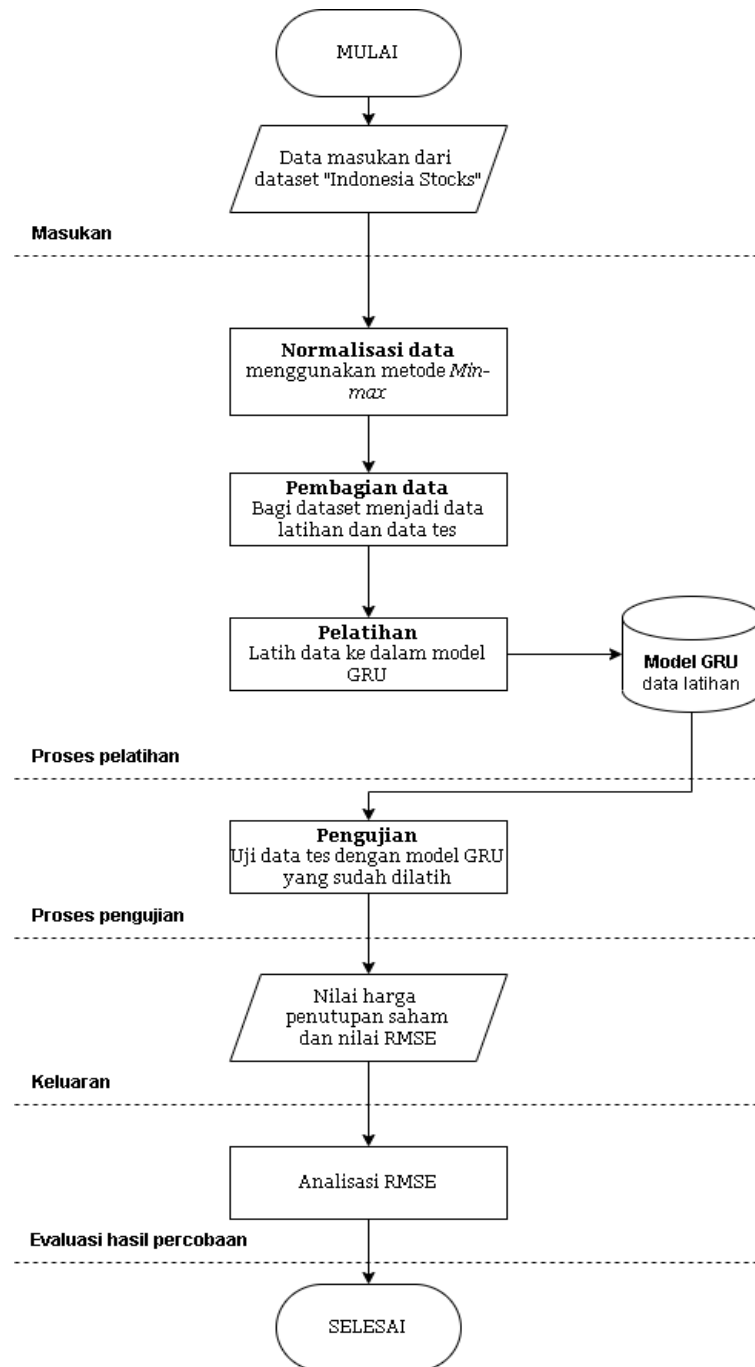
sehingga skala data tidak terlalu jauh. Proses normalisasi data dalam penelitian ini menggunakan normalisasi *Min-Max*. Setelah itu, data dibagi menjadi dua kelompok, yaitu data latihan dan data tes. Data latihan yang digunakan dibagi menjadi dua periode, yaitu periode satu tahun dan periode dua tahun. Selanjutnya, data latihan dilatih menggunakan model GRU. Setelah dilatih, model GRU siap untuk melakukan prediksi terhadap data tes.

Objektif adalah acuan pengukuran evaluasi model. Penelitian ini berfokus untuk menguji nilai akurasi prediksi harga saham. Bagian pengukuran adalah alat yang dipakai untuk mengukur suatu Objektif. Penelitian ini menggunakan *Root Mean Squared Error* (RMSE) untuk mengukur akurasi yang dihasilkan oleh model prediksi harga saham.

3.3 Urutan Proses Global

Bab ini menjelaskan urutan proses yang dilalui dalam penelitian secara global. Urutan proses global dapat dilihat pada gambar 3.2. Dalam penelitian ini hal pertama yang dilakukan adalah memasukkan *dataset* ke dalam program. Selanjutnya, data yang sudah dimuat ke dalam sistem dinormalisasi menggunakan normalisasi *Min-max*. Setelah data sudah diolah, data tersebut dibagi menjadi data latihan untuk proses pelatihan model dan data tes untuk proses pengujian model.

Penelitian ini menyiapkan dua periode data latihan. data latihan pertama menggunakan data dengan periode satu tahun dan data latihan kedua menggunakan data dengan periode lima tahun. Selanjutnya, data latihan dilatih menggunakan model GRU. Lalu, Model tersebut dipakai untuk memprediksi data yang digolongkan ke dalam data tes. Hasil prediksi tersebut dievaluasi dengan menganalisis nilai RMSE. Nilai RMSE yang semakin kecil menandakan nilai prediksi yang dihasilkan model mendekati nilai aslinya.



Gambar 3.2 Flowchart Urutan Proses Global

3.4 Analisis Manual

Bab ini menjelaskan analisis tahapan proses yang dilakukan oleh sistem dengan melakukan analisis dan perhitungan secara manual.

3.4.1 Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset". Penelitian ini

menggunakan data dari perusahaan PT. Telkom, PT. Aneka Tambang, dan PT. Indofood sebagai *dataset* utama. *Dataset* perusahaan PT. XL Axiata, PT. Vale Indonesia, dan PT. Japfa Comfeed Indonesia dijadikan *dataset* pembanding. *Dataset* memiliki fitur tanggal, *Open*, *High*, *Low*, *Close*, dan *Volume*. Tabel 3.1 adalah contoh matriks data saham.

Tabel 3.1 Tabel Contoh Matriks Data Saham

Date	Open	High	Low	Close	Volume
7/6/2015	566	566	558	558	2313516
7/7/2015	554	558	550	550	3800461
7/8/2015	545	550	520	524	10658865
7/9/2015	520	524	508	520	3959394
7/10/2015	516	533	516	520	2524117

3.4.2 Normalisasi *Min-max*

Data dinormalisasi agar skalanya tidak terlalu lebar. Normalisasi yang digunakan dalam penelitian ini adalah Normalisasi *Min-max*. Normalisasi *Min-max* mengurangkan nilai x dengan nilai minimum dan membaginya dengan nilai maksimum dikurangi dengan nilai minimum seperti pada persamaan 2.5. Persamaan diterapkan pada semua baris dan kolom pada *dataset* kecuali kolom *date*. Berikut adalah contoh perhitungan normalisasi *Min-max*.

Tabel 3.2 Tabel Contoh Matriks Data Saham Sebelum Normalisasi

Date	Open	High	Low	Close	Volume
7/6/2015	566	566	558	558	2313516
7/7/2015	554	558	550	550	3800461
7/8/2015	545	550	520	524	10658865
7/9/2015	520	524	508	520	3959394
7/10/2015	516	533	516	520	2524117

$$x_i = 566, \min(x) = 516, \max(x) = 566 \quad (3.1)$$

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

$$x_{scaled} = \frac{566 - 516}{566 - 516} \quad (3.3)$$

$$x_{scaled} = 1 \quad (3.4)$$

Pada bagian 3.1 menjelaskan kondisi yang diperlukan untuk menghitung normalisasi *Min-max*. Pada bagian tersebut dapat dilihat bahwa nilai x ke- i dari fitur *open* adalah 566, nilai minimum x adalah nilai minimum dari fitur *open* sebesar 516, dan nilai maksimum x adalah nilai maksimum dari fitur *open* sebesar 566. Setelah normalisasi diterapkan di setiap fitur maka akan menghasilkan nilai seperti tabel 3.3.

Tabel 3.3 Tabel Contoh Matriks Data Saham Sesudah Normalisasi

Date	Open	High	Low	Close	Volume
7/6/2015	1	1	1	1	0
7/7/2015	0.76	0.81	0.84	0.79	0.18
7/8/2015	0.58	0.62	0.24	0.11	1
7/9/2015	0.08	0	0	0	0.2
7/10/2015	0	0.2	0.16	0	0.03

3.4.3 Split Data

Split data dilakukan secara manual dengan cara mencari panjang data dengan periode satu tahun dan periode lima tahun. Selanjutnya, data latihan akan dipisah dari *dataset* dengan indeks ke-0 sampai panjang data dengan periode satu tahun atau lima tahun. Periode satu tahun dan lima tahun untuk data latih diambil dari penelitian yang ditulis oleh Rahman [15]. 100 hari setelah indeks terakhir data

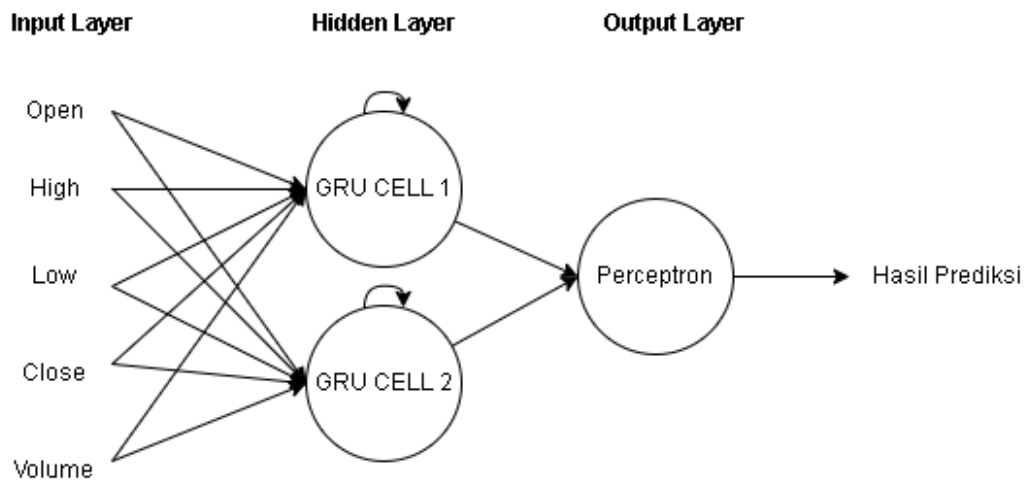
latihan dijadikan data tes. data tes 100 hari digunakan karena berdasarkan teori Dow tren periode satu tahun atau lebih dapat digunakan untuk memprediksi tren satu bulan sampai empat bulan. 100 Hari digunakan karena berada pada rentang waktu satu bulan sampai empat bulan.

3.4.4 Perhitungan *Gated Recurrent Unit* (GRU)

Tabel 3.4 Tabel Contoh Matriks Data Latihan

Date	Open	High	Low	Close	Volume
7/6/2015	1	1	1	1	0
7/7/2015	0.76	0.81	0.84	0.79	0.18
7/8/2015	0.58	0.62	0.24	0.11	1
7/9/2015	0.08	0	0	0	0.2
7/10/2015	0	0.2	0.16	0	0.03

Tabel di atas menunjukkan contoh hasil data pada fitur *open* yang sudah dinormalisasi menggunakan persamaan 3.2. Persamaan tersebut berlaku untuk menghitung fitur *high*, *low*, *close*, dan *volume*. Label merupakan nilai *close* dari data $t + 1$. Contoh perhitungan GRU hanya dilakukan dengan data pada tanggal 6 Juli 2015 sampai 7 Juli 2015. Jangka 2 hari sudah cukup untuk mewakili keseluruhan proses perhitungan Perhitungan GRU karena yang berbeda pada t selanjutnya hanya nilai h_{t-1} dan bobot saja. Perhitungan GRU menggunakan 2 unit untuk jumlah neuron di *hidden layer*. 2 unit digunakan untuk memberikan contoh yang terjadi jika unit yang digunakan lebih dari 1 karena pada penelitian ini unit yang digunakan lebih dari 1. Berikut adalah gambar arsitektur GRU untuk kasus perhitungan ini.



Gambar 3.3 Arsitektur perhitungan GRU

Nilai bobot U diinisialisasi dengan Glorot Uniform karena pada penelitian ini parameter untuk inisialisasi bobot U menggunakan Glorot Uniform. Nilai bobot W diinisialisasi dengan Orthogonal karena pada penelitian ini parameter untuk inisialisasi bobot W menggunakan Orthogonal. Nilai bias diinisialisasi dengan Zeros jadi nilai bias selalu diinisialisasi dengan angka 0. Inisialisasi Zeros digunakan karena penelitian ini memakai nilai *default* yaitu inisialisasi Zeros untuk inisialisasi bias.

3.4.4.1 Analisis Parameter pada Model GRU

Bagian ini menjelaskan tentang parameter yang tersedia pada model GRU dan alasan penelitian ini memakai nilai setiap parameter tersebut. Hubungan antara parameter tersebut dengan rumus yang digunakan dalam teori juga dijelaskan dalam bagian ini. Parameter dalam GRU yang tidak menggunakan nilai *default* hanya unit. 2 unit digunakan untuk keperluan analisis perhitungan saja sedangkan pada tahap implementasi unit yang digunakan adalah 16, 32, 64 dan 128. Nilai *default* pada parameter GRU digunakan karena pada penelitian sebelumnya tidak ada yang mengubah parameter GRU secara spesifik. Berikut adalah tabel penjelasan parameter model GRU.

Tabel 3.5 Tabel Penjelasan Parameter Model GRU

No	Parameter	Simbol dalam rumus	Nilai yang dipakai
1	units	-	2
2	activation	σ (persamaan 2.6, 2.8)	sigmoid
3	recurrent_activation	\tanh (persamaan 2.7)	\tanh
4	use_bias	-	true

5	kernel_initializer	U (persamaan 2.6, 2.7, 2.8)	glorot_uniform
6	recurrent_initializer	W	orthogonal
7	bias_initializer	b (persamaan 2.6, 2.7, 2.8)	zeros
8	kernel_regularizer	-	none
9	recurrent_regularizer	-	none
10	bias_regularizer	-	none
11	activity_regularizer	-	none
12	kernel_constraint	-	none
13	recurrent_constraint	-	none
14	bias_constraint	-	none
15	dropout	-	0
16	recurrent_dropout	-	0
17	return_sequences	-	false
18	return_state	-	false
19	go_backwards	-	false
20	stateful	-	false
21	unroll	-	false
22	time_major	-	false
23	reset_after	-	true

3.4.4.2 Contoh Perhitungan Inisialisasi U

Bobot U dengan inisialisasi menggunakan Glorot Uniform terdistribusi diantara $-limit$ sampai $limit$. Limit memiliki persamaan $limit = \sqrt{6/(fan_in + fan_out)}$. fan_in adalah jumlah input unit (fitur) dan fan_out jumlah output unit.

Perhitungan Inisialisasi U

$$fan_in = 5$$

$$fan_out = 2$$

$$limit = \sqrt{\frac{6}{(fan_in + fan_out)}}$$

$$limit = \sqrt{\frac{6}{(5+2)}}$$

$$limit = \sqrt{\frac{6}{(7)}}$$

$$limit = 0.926$$

Fan_in yang digunakan dalam perhitungan ini adalah lima karena jumlah fitur yang digunakan adalah lima. Fan_out yang digunakan dalam perhitungan ini adalah dua karena jumlah output yang dihasilkan GRU sesuai parameter unit yang digunakan, yaitu dua. Limit yang dihasilkan adalah 0.926 artinya bobot U bernilai acak dengan skala -0.926 sampai 0.926 hasil distribusi uniform.

3.4.4.3 Contoh Perhitungan Inisialisasi W

Bobot W dengan inisialisasi menggunakan Orthogonal *initializer* yang menghasilkan matriks orthogonal. Matriks orthogonal adalah matriks yang nilai inversinya sama dengan nilai transposenya. Jika matriks yang recurrent state memiliki sedikit baris daripada kolom maka output akan memiliki baris Orthogonal, begitu juga sebaliknya. Skala yang dihasilkan inisialisasi ini adalah -1 sampai 1 dan merupakan hasil distribusi normal. Berikut adalah contoh perhitungan inisialisasi W .

Perhitungan Inisialisasi W

A merupakan matriks orthogonal

$$A = \begin{pmatrix} -0.939 & -0.342 \\ 0.342 & -0.939 \end{pmatrix}$$

$$A^T = \begin{pmatrix} -0.939 & 0.342 \\ -0.342 & -0.939 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} -0.939 & 0.342 \\ -0.342 & -0.939 \end{pmatrix}$$

jika matriks h_{t-1} seperti demikian dimana kolom lebih sedikit dari baris maka nilai W adalah kolom dari matriks orthogonal

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

$$W = \begin{pmatrix} -0.939 & -0.342 \end{pmatrix}$$

3.4.4.4 Perhitungan t Pertama

Perhitungan t pertama adalah perhitungan GRU pada tanggal 6 Juli 2015.

Perhitungan pada *reset gate* t pertama

$$W_r = \begin{pmatrix} 0.032 & 0.023 \end{pmatrix}$$

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

$$U_r = \begin{pmatrix} 0.040 & 0.030 & 0.010 & 0.015 & 0.035 \\ 0.044 & -0.030 & 0.010 & -0.015 & 0.020 \end{pmatrix}$$

$$x_t = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$b_r = 0$$

$$r_t = \sigma(W_r h_{t-1} + U_r x_t + b_r)$$

$$r_t = \sigma\left(0 + \begin{pmatrix} 0.095 \\ 0.009 \end{pmatrix} + 0\right)$$

$$r_t = \begin{pmatrix} 0.524 \\ 0.502 \end{pmatrix}$$

Perhitungan pada *reset gate* t pertama dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W_r yang merupakan bobot dari nilai h_{t-1} . h_{t-1} adalah nilai *hidden state* sebelumnya. h_{t-1} bernilai nol karena perhitungan t pertama merupakan awal perhitungan dan tidak memiliki nilai *hidden state* sebelumnya. U_r adalah bobot dari nilai x_t . U_r mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur pada tanggal 6 Juli 2015. b_r merupakan nilai bias yang digunakan dalam perhitungan *reset gate* t pertama. Perhitungan ini menggunakan fungsi aktivasi *sigmoid* yang dilambangkan dengan σ . Perhitungan *reset gate* t pertama menghasilkan nilai r_t sebesar $\begin{pmatrix} 0.524 \\ 0.502 \end{pmatrix}$.

Perhitungan pada <i>new remember</i> t pertama
$W = \begin{pmatrix} 0.010 & -0.030 \end{pmatrix}$ $h_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $U = \begin{pmatrix} -0.010 & -0.010 & 0.015 & 0.040 & -0.035 \\ 0.024 & -0.020 & -0.040 & -0.023 & 0.010 \end{pmatrix}$ $x_t = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ $b = 0$ $\bar{h}[t] = \tanh(W(r_t * h_{t-1}) + Ux_t + b)$ $\bar{h}[t] = \tanh \left(\begin{pmatrix} 0.01 & -0.03 \end{pmatrix} \bullet \left(\begin{pmatrix} 0.524 \\ 0.502 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right) + \begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix} + 0 \right)$ $\bar{h}[t] = \tanh \left(0 + \begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix} + 0 \right)$ $\bar{h}[t] = \begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix}$

Perhitungan pada *new remember* t pertama dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W yang merupakan bobot dari nilai $r_t \otimes h_{t-1}$. U adalah bobot dari nilai x_t . U mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur

pada tanggal 6 Juli 2015. b merupakan nilai bias yang digunakan dalam perhitungan *new remember* t pertama. Perhitungan ini menggunakan fungsi aktivasi tanh. Perhitungan *new remember* t pertama menghasilkan nilai $\bar{h}[t]$ sebesar $\begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix}$.

Perhitungan pada <i>update gate</i> t pertama
$W_z = \begin{pmatrix} 0.030 & -0.045 \end{pmatrix}$ $h_{t-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $U_z = \begin{pmatrix} -0.033 & 0.014 & 0.020 & 0.005 & 0.042 \\ -0.041 & 0.025 & -0.030 & 0.010 & -0.011 \end{pmatrix}$ $x_t = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ $b_z = 0$ $z_t = \sigma(W_z h_{t-1} + U_z x_t + b_z)$ $z_t = \sigma\left(0 + \begin{pmatrix} 0.006 \\ -0.036 \end{pmatrix} + 0\right)$ $z_t = \begin{pmatrix} 0.501 \\ 0.491 \end{pmatrix}$

Perhitungan pada *update gate* t pertama dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W_z yang merupakan bobot dari nilai h_{t-1} . U_z adalah bobot dari nilai x_t . U_z mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur pada tanggal 6 Juli 2015. b_z merupakan nilai bias yang digunakan dalam perhitungan *update gate* t pertama. Perhitungan ini menggunakan fungsi aktivasi *sigmoid* yang dilambangkan dengan σ . Perhitungan *update gate* t pertama menghasilkan nilai z_t sebesar $\begin{pmatrix} 0.501 \\ 0.491 \end{pmatrix}$.

Perhitungan pada *hidden state* t pertama

$$z_t = \begin{pmatrix} 0.502 \\ 0.491 \end{pmatrix}$$

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

$$\bar{h}[t] = \begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \bar{h}[t]$$

$$h_t = \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.502 \\ 0.491 \end{pmatrix} \right) \otimes \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.502 \\ 0.491 \end{pmatrix} \otimes \begin{pmatrix} 0.035 \\ -0.059 \end{pmatrix}$$

$$h_t = 0 + \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$$

$$h_t = \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$$

Perhitungan pada *hidden state* t pertama dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol z_t yang merupakan *update gate* t pertama. h_{t-1} adalah nilai *hidden state* sebelumnya. $\bar{h}[t]$ merupakan nilai *new remember* t pertama. Perhitungan *hidden state* t pertama menghasilkan nilai h_t sebesar $\begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$.

3.4.4.5 Perhitungan t Kedua

Perhitungan t kedua adalah perhitungan GRU pada tanggal 7 Juli 2015.

Perhitungan pada <i>reset gate</i> t kedua
$W_r = \begin{pmatrix} 0.010 & 0.020 \end{pmatrix}$ $h_{t-1} = \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$ $U_r = \begin{pmatrix} 0.010 & 0.030 & 0.020 & 0.015 & -0.040 \\ -0.014 & 0.010 & 0.035 & -0.015 & 0.030 \end{pmatrix}$ $x_t = \begin{pmatrix} 0.760 \\ 0.810 \\ 0.840 \\ 0.790 \\ 0.180 \end{pmatrix}$ $b_r = 0$ $r_t = \sigma(W_r h_{t-1} + U_r x_t + b_r)$ $r_t = \sigma(-0.0004 + \begin{pmatrix} 0.053 \\ 0.020 \end{pmatrix} + 0)$ $r_t = \sigma \begin{pmatrix} 0.053 \\ 0.020 \end{pmatrix}$ $r_t = \begin{pmatrix} 0.513 \\ 0.505 \end{pmatrix}$

Perhitungan pada *reset gate* t kedua dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W_r yang merupakan bobot dari nilai h_{t-1} . h_{t-1} adalah nilai *hidden state* sebelumnya. U_r adalah bobot dari nilai x_t . U_r mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur pada tanggal 7 Juli 2015. b_r merupakan nilai bias yang digunakan dalam perhitungan *reset gate* t kedua. Perhitungan ini menggunakan fungsi aktivasi *sigmoid* yang dilambangkan dengan σ . Perhitungan *reset gate* t kedua menghasilkan nilai r_t sebesar $\begin{pmatrix} 0.513 \\ 0.505 \end{pmatrix}$.

Perhitungan pada *new remember* t kedua

$$W = \begin{pmatrix} -0.020 & -0.040 \end{pmatrix}$$

$$h_{t-1} = \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$$

$$U = \begin{pmatrix} 0.011 & 0.005 & 0.030 & -0.020 & -0.016 \\ -0.012 & -0.04 & 0.020 & 0.036 & -0.020 \end{pmatrix}$$

$$x_t = \begin{pmatrix} 0.760 \\ 0.810 \\ 0.840 \\ 0.790 \\ 0.180 \end{pmatrix}$$

$$b = 0$$

$$\bar{h}[t] = \tanh(W(r_t * h_{t-1}) + Ux_t + b)$$

$$\bar{h}[t] = \tanh \left(\begin{pmatrix} -0.02 & -0.04 \end{pmatrix} \bullet \left(\begin{pmatrix} 0.513 \\ 0.505 \end{pmatrix} \otimes \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix} \right) + \begin{pmatrix} 0.019 \\ 0.00012 \end{pmatrix} + 0 \right)$$

$$\bar{h}[t] = \tanh \left(0.0004 + \begin{pmatrix} 0.019 \\ 0.00012 \end{pmatrix} + 0 \right)$$

$$\bar{h}[t] = \tanh \begin{pmatrix} 0.019 \\ 0.0005 \end{pmatrix}$$

$$\bar{h}[t] = \begin{pmatrix} 0.019 \\ 0.0005 \end{pmatrix}$$

Perhitungan pada *new remember* t kedua dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W yang merupakan bobot dari nilai $r_t \otimes h_{t-1}$. U adalah bobot dari nilai x_t . U mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur pada tanggal 7 Juli 2015. b merupakan nilai bias yang digunakan dalam perhitungan *new remember* t kedua. Perhitungan ini menggunakan fungsi aktivasi tanh. Perhitungan *new remember* t kedua menghasilkan nilai $\bar{h}[t]$ sebesar $\begin{pmatrix} 0.019 \\ 0.0005 \end{pmatrix}$.

Perhitungan pada *update gate* t kedua

$$W_z = \begin{pmatrix} 0.012 & -0.034 \end{pmatrix}$$

$$h_{t-1} = \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix}$$

$$U_z = \begin{pmatrix} 0.010 & 0.030 & -0.005 & -0.010 & -0.030 \\ 0.020 & 0.040 & 0.005 & -0.020 & -0.040 \end{pmatrix}$$

$$x_t = \begin{pmatrix} 0.760 \\ 0.810 \\ 0.840 \\ 0.790 \\ 0.180 \end{pmatrix}$$

$$b_z = 0$$

$$z_t = \sigma(W_z h_{t-1} + U_z x_t + b_z)$$

$$z_t = \sigma(0.001 + \begin{pmatrix} 0.0144 \\ 0.0288 \end{pmatrix} + 0)$$

$$z_t = \sigma \begin{pmatrix} 0.016 \\ 0.030 \end{pmatrix}$$

$$z_t = \begin{pmatrix} 0.504 \\ 0.508 \end{pmatrix}$$

Perhitungan pada *update gate* t kedua dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol W_z yang merupakan bobot dari nilai h_{t-1} . U_z adalah bobot dari nilai x_t . U_z mempunyai dimensi dua kali lima karena perhitungan menggunakan dua unit GRU dan lima fitur. x_t merupakan nilai fitur pada tanggal 7 Juli 2015. b_z merupakan nilai bias yang digunakan dalam perhitungan *update gate* t kedua. Perhitungan ini menggunakan fungsi aktivasi *sigmoid* yang dilambangkan dengan σ . Perhitungan *update gate* t kedua menghasilkan nilai z_t sebesar $\begin{pmatrix} 0.504 \\ 0.508 \end{pmatrix}$.

Perhitungan pada *hidden state* t kedua

$$\begin{aligned}
 z_t &= \begin{pmatrix} 0.504 \\ 0.508 \end{pmatrix} \\
 h_{t-1} &= \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix} \\
 \bar{h}[t] &= \begin{pmatrix} 0.019 \\ 0.0005 \end{pmatrix} \\
 h_t &= (1 - z_t) * h_{t-1} + z_t * \bar{h}[t] \\
 h_t &= \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.504 \\ 0.508 \end{pmatrix} \right) \otimes \begin{pmatrix} 0.018 \\ -0.029 \end{pmatrix} + \begin{pmatrix} 0.504 \\ 0.508 \end{pmatrix} \otimes \begin{pmatrix} 0.019 \\ 0.0005 \end{pmatrix} \\
 h_t &= \begin{pmatrix} 0.009 \\ -0.014 \end{pmatrix} + \begin{pmatrix} 0.001 \\ 0.0003 \end{pmatrix} \\
 h_t &= \begin{pmatrix} 0.019 \\ -0.014 \end{pmatrix}
 \end{aligned}$$

Perhitungan pada *hidden state* t kedua dapat dilihat di atas. Pada perhitungan tersebut terdapat simbol z_t yang merupakan *update gate* t kedua. h_{t-1} adalah nilai *hidden state* sebelumnya. $\bar{h}[t]$ merupakan nilai *new remember* t kedua. Perhitungan *hidden state* t kedua menghasilkan nilai h_t sebesar $\begin{pmatrix} 0.019 \\ -0.014 \end{pmatrix}$ yang merupakan hasil pembulatan.

3.4.5 Perhitungan *Perceptron* pada Lapisan Keluaran

Setelah melewati perhitungan GRU, hasil keluaran GRU masuk ke dalam *perceptron* pada lapisan keluaran. *Perceptron* yang digunakan hanya memiliki satu unit karena penelitian ini hanya membutuhkan satu hasil, yaitu nilai *close*. *Perceptron* menggunakan fungsi aktivasi *sigmoid*. Fungsi aktivasi *sigmoid* dipilih karena fungsi aktivasi *sigmoid* memiliki skala nol sampai satu, cocok untuk harga saham yang tidak pernah bernilai negatif. Seperti yang dapat dilihat pada gambar 2.1, *perceptron* memiliki persamaan $h(x) = \sigma(x^T w)$. Berikut adalah contoh perhitungan *perceptron* pada lapisan keluaran.

Perhitungan *perceptron* pada lapisan keluaran

$$\begin{aligned}x &= \begin{pmatrix} 0.019 \\ -0.014 \end{pmatrix} \\w &= \begin{pmatrix} 0.040 \\ 0.030 \end{pmatrix} \\h(x) &= \sigma(x^T w) \\h(x) &= \sigma \left(\begin{pmatrix} 0.019 & -0.014 \end{pmatrix} \cdot \begin{pmatrix} 0.040 \\ 0.030 \end{pmatrix} \right) \\h(x) &= \sigma(0.00034) \\h(x) &= 0.500\end{aligned}$$

Pada perhitungan di atas terdapat simbol x yang merupakan nilai masukan pada *perceptron*. Nilai x pada perhitungan tersebut diambil dari nilai h_t pada perhitungan *hidden state* t kedua. Simbol w adalah bobot dari x . Bobot menggunakan nilai acak dengan skala -0.05 sampai 0.05.

3.4.6 Perhitungan *Root Mean Square Error* (RMSE)

RMSE memiliki persamaan 2.4. Pada persamaan tersebut terdapat $h(x^{(i)})$ yang diartikan sebagai hasil prediksi (\hat{y}). Oleh karena itu, persamaan tersebut disederhanakan seperti persamaan 3.5.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2} \quad (3.5)$$

Perhitungan RMSE menggunakan persamaan 3.5. Nilai \hat{y} dan y diasumsikan sudah tersedia dan nilai tersebut didapat dari hasil percobaan. Perhitungan menggunakan lima data sehingga nilai m adalah lima. Berikut adalah perhitungan RMSE.

Perhitungan RMSENilai \hat{y} dan y :

i	\hat{y}	y
1	565	558
2	530	550
3	530	524
4	522	520
5	520	520

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2}$$

$$RMSE = \sqrt{\frac{1}{5} ((565 - 558)^2 + (530 - 550)^2 + (530 - 524)^2 + (522 - 520)^2 + (520 - 520)^2)}$$

$$RMSE = \sqrt{\frac{1}{5} (49 + 400 + 36 + 4 + 0)}$$

$$RMSE = \sqrt{97.8}$$

$$RMSE = 9.889$$

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang proses implementasi dan pengujian terhadap sistem yang dibangun dalam penelitian ini. Bab ini dibuat untuk memberi gambaran besar dari hasil penelitian ini.

4.1 Lingkungan Implementasi

Bagian ini menjelaskan tentang perangkat yang digunakan dalam proses implementasi dan pengujian sistem. Proses implementasi dan pengujian ini melibatkan penggunaan perangkat keras dan perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

Spesifikasi dari perangkat keras yang digunakan dalam proses implementasi dan pengujian sistem adalah sebagai berikut:

1. *Laptop* ASUS TUF GAMING FX504GD_FX80GD.
2. *Processor* Intel Core i7-8750H CPU @ 2.2GHz (12 CPUs).
3. *Solid State Drive* kapasitas 512 GB.
4. RAM dengan kapasitas 16 GB.

4.1.2 Spesifikasi Perangkat Lunak

Lingkungan perangkat lunak yang digunakan dalam proses implementasi dan pengujian sistem adalah sebagai berikut:

1. Sistem Operasi Windows 10 PRO.
2. IDE: Jupyter Notebook 6.1.4.
3. *Development Tools* Anaconda 3, 2020.07 (Python 3.8.3 64-bit).

4.2 Implementasi Perangkat Lunak

Bagian ini membahas proses implementasi perangkat lunak untuk membangun sistem. Hal yang dibahas antara lain daftar *class* dan *method* yang diimplementasi, penggunaan Jupyter Notebook, dan penggunaan *dataset*.

4.2.1 Implementasi Class

Bagian ini menjelaskan tentang *class* dan *method* yang diimplementasi pada penelitian ini. *Class* dan *method* yang dibahas pada bagian ini dibuat dari awal. *Method* dikelompokkan ke dalam sebuah *class* sesuai dengan fungsinya.

4.2.1.1 Class Dataset

Class Dataset diciptakan untuk memuat *dataset* ke dalam sistem, membuat label pada *dataset*, dan mencari nilai minimum dan maksimum tiap fitur. *Class* ini mempunyai atribut dan *method* yang dijelaskan pada tabel berikut.

Tabel 4.1 Atribut pada *Class Dataset*

atribut:					
DataFrame	raw	Array	minX	Array	maxY
Array numPy	x	Array	maxX		
Array numPy	y	Array	minY		

Tabel 4.2 Daftar *Method Class Dataset*

No.	Method	Masukan	Luaran	Keterangan
1	__init__	dataset_name: string	-	Memuat dataset ke dalam sistem dan menyimpannya ke dalam atribut raw.
2	create_fitur _and_label	-	-	Membuat fitur dan label yang disimpan kedalam atribut x dan y.
3	get_min_max _value	-	-	Mencari nilai minimum dan maksimum dari atribut x dan y.

4.2.1.2 Class Preprocessing

Class Preprocessing diciptakan untuk melakukan *data preprocessing*. *Class Preprocessing* tidak mempunyai atribut. *Class* ini mempunyai *method* yang dijelaskan pada tabel berikut.

Tabel 4.3 Daftar *Method Class Preprocessing*

No	Method	Masukan	Luaran	Keterangan
1	min_max	data: float, min: float, max: float	float	Melakukan perhitungan nilai berdasarkan rumus normalisasi Min-max.
2	inverse_min_max	data: float, min: float, max: float	float	Melakukan perhitungan nilai berdasarkan hasil inversi rumus normalisasi Min-max.
3	transform_min_max	datas: array, mins: array, maxs: array	array numpy	Melakukan perhitungan terhadap kumpulan data menggunakan fungsi min_max.
4	inverse_transform_min_max	datas: array, mins: array, maxs: array	array numpy	Melakukan perhitungan terhadap kumpulan data menggunakan fungsi inverse_min_max.

4.2.1.3 Class DataSplit

Class DataSplit adalah sebuah *class* yang bertugas untuk melakukan pembagian data menjadi data latih dan data tes. *Class* ini mempunyai atribut dan *method* yang dijelaskan pada tabel berikut.

Tabel 4.4 Atribut pada *Class DataSplit*

atribut:			
Array numPy	x_train	Array numPy	y_test
Array numpy	y_train	Array numpy	x_test

Tabel 4.5 Daftar *Method Class DataSplit*

No	Method	Masukan	Luaran	Keterangan
1	<code>__init__</code>	x,y: Array numPy, length: integer	-	Membagi data menjadi data latih dan data tes kemudian menyimpannya ke dalam atribut <code>x_train</code> , <code>y_train</code> , <code>x_test</code> , dan <code>y_test</code> .

4.2.1.4 Class Training

Class Training adalah sebuah *class* yang bertugas untuk membuat model GRU dan melatih data ke dalam model tersebut. *Class* ini mempunyai atribut dan *method* yang dijelaskan pada tabel berikut.

Tabel 4.6 Atribut pada *Class Training*

atribut:			
Array	BATCH_SIZE	Array	UNITS
Array	EPOCHS	string	FOLDER_PATH

Tabel 4.7 Daftar *Method Class Training*

No	Method	Masukan	Luaran	Keterangan
1	<code>__init__</code>	-	-	inisialisasi nilai pada setiap atribut.
2	<code>training_model</code>	x_train,y_train: Array Numpy, dataset_name: string, train_period: string	-	Melakukan proses latihan terhadap data dengan menggunakan kombinasi hyperparameter yang berbeda setiap perulangan.

4.2.1.5 Class Visualization

Class Visualization diciptakan untuk melakukan visualisasi yang dihasilkan pada tahap pelatihan dan pengujian. *Class Visualization* tidak mempunyai atribut.

Class ini mempunyai *method* yang dijelaskan pada tabel berikut.

Tabel 4.8 Daftar *Method Class Visualization*

No	Method	Masukan	Luaran	Keterangan
1	ploting_training_loss	df: DataFrame, rmse: array, model_name: string	-	Menampilkan dan Menyimpan figur hasil plot data.
2	ploting_prediction	yhat: array, y_test: array, minY: float, maxY: float, model_name: string	-	Menampilkan dan Menyimpan figur hasil plot data.
3	training_visualization	dataset_name: string, train_period: string	-	Melakukan visualisasi terhadap hasil pelatihan dari semua percobaan dengan memanggil <i>method</i> ploting_training_loss dalam suatu perulangan.
4	testing_visualization	dataset_name: string, train_period: string, X_test: array, y_test: array, minY: float, maxY: float	-	Melakukan visualisasi terhadap hasil pengujian dari semua percobaan dengan memanggil <i>method</i> ploting_prediction dalam suatu perulangan.

4.2.2 Penggunaan Jupyter Notebook

Penelitian ini menggunakan Jupyter Notebook sebagai *Integrated Development Environment* (IDE). Jupyter Notebook mudah digunakan untuk melakukan pemrograman berbasis Python. Jupyter Notebook menyediakan *cell* yang dijalankan secara terpisah. Jupyter Notebook yang digunakan dalam penelitian ini masih menggunakan sumber daya *Central Processing Unit* (CPU) namun cukup untuk melakukan penelitian ini.

4.2.3 Penggunaan *Dataset*

Penelitian ini menggunakan tiga *dataset* perusahaan, yaitu PT. Aneka Tambang, PT. Indofood, dan PT. Telkom sebagai *dataset* utama. *Dataset* perusahaan PT. XL Axiata, PT. Vale Indonesia, dan PT. Japfa Comfeed Indonesia dijadikan pembanding untuk *dataset* utama. Semua data yang ada pada *dataset* tersebut digunakan secara keseluruhan. Tidak ada fitur yang dibuang kecuali tanggal.

Dataset PT. Aneka Tambang memiliki data lebih sedikit dengan selisih dua hari. Dalam dua hari tersebut, PT. Aneka Tambang tidak mempunyai volume. Dengan kata lain, dalam dua hari tersebut tidak terjadi transaksi pada PT. Aneka Tambang. Hal ini dikonfirmasi pada *dataset* "Dataset Saham Indonesia / Indonesia Stock Dataset". Hal serupa juga terjadi pada PT. Japfa Comfeed Indonesia dan PT. XL Axiata. Namun, selisih tersebut tidak berpengaruh terhadap proses pelatihan karena selisih tersebut dimasukkan ke dalam data pengujian. Pembagian *dataset* dapat dilihat pada tabel berikut.

Tabel 4.9 Tabel Pembagian *Dataset*

No	Dataset	Training period	Total Data	Pembagian Data	
				Pelatihan	Pengujian
1	ANTM	1 tahun	1501	235	100
2		5 tahun	1501	1191	100
3	INDF	1 tahun	1503	235	100
4		5 tahun	1503	1191	100
5	TLKM	1 tahun	1503	235	100
6		5 tahun	1503	1191	100
7	INCO	1 tahun	1503	235	100
8		5 tahun	1503	1191	100
9	JPFA	1 tahun	1502	235	100
10		5 tahun	1502	1191	100
11	EXCL	1 tahun	1504	235	100
12		5 tahun	1504	1191	100

Pembagian data pelatihan dan data pengujian didasari oleh teori Dow yang sudah dijelaskan pada bagian 2.3.2. Penelitian ini menjadikan *Primary trend* sebagai data pelatihan untuk memprediksi *Secondary Trends* sebagai data pengujian. Nilai *training period* 1 tahun dan 5 tahun diambil dari penelitian yang dilakukan oleh Rahman [15]. *Training period* 1 tahun menghasilkan data pelatihan sebanyak 235 hari. *Training period* 5 tahun akan menghasilkan data pelatihan sebanyak 1191 hari. Data pelatihan satu tahun memiliki rentang waktu 6 Juli 2015 sampai 6 Juli 2016 dan 100 hari setelah 6 Juli 2015 dijadikan data pengujian. Data pelatihan satu tahun memiliki rentang waktu 6 Juli 2015 sampai 6 Juli 2020 dan 100 hari setelah 6 Juli 2020 dijadikan data pengujian.

4.3 Implementasi Aplikasi

Bagian ini membahas tentang *Graphical User Interface* (GUI) yang digunakan untuk mempresentasikan sistem prediksi harga saham. Tampilan GUI dibuat untuk menguji model prediksi yang telah dihasilkan dalam percobaan. Pengguna harus memasukkan parameter seperti *dataset* yang digunakan, *training period*, *unit*, *batch size*, dan *epoch* yang tersedia. GUI dibangun menggunakan *framework* Flask dan HTML.

Form prediksi harga saham menggunakan model GRU

Dataset*
PT.Aneka tambang

Training period*
1 tahun

Unit*
16

Batch size*
32

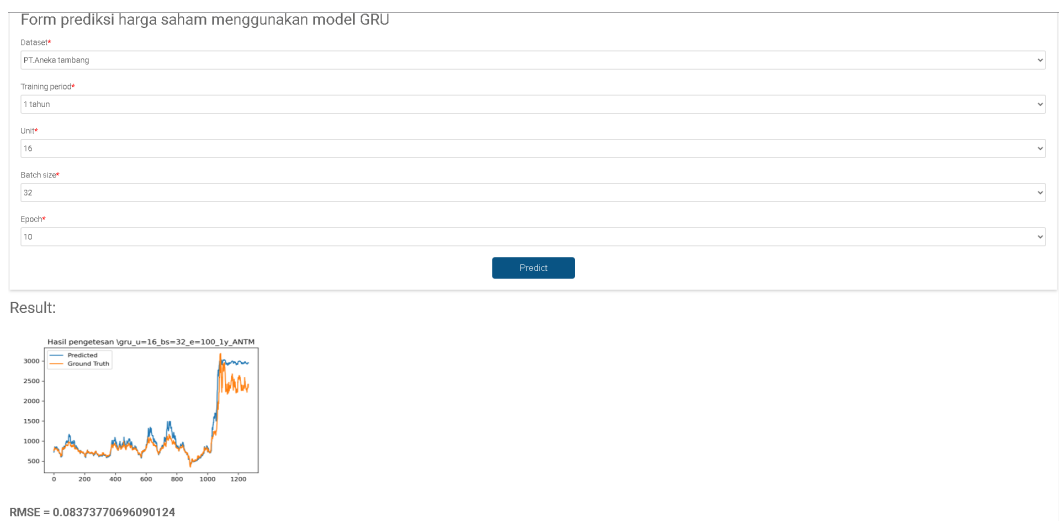
EPOCH*
10

Predict

Result:

Gambar 4.1 Halaman Utama

Gambar 4.1 memperlihatkan tampilan utama GUI yang dibuat. Terdapat formulir yang berisi beberapa masukan berupa *dropdown*. Terdapat tombol bertuliskan "predict" yang berfungsi untuk menjalankan proses prediksi dan menampilkan hasilnya. Setelah tombol tersebut ditekan, masukan yang dipilih diproses dan sistem memuat model prediksi yang ada. Setelah model dimuat maka proses prediksi dilakukan dan hasil prediksi ditampilkan pada halaman yang sama. gambar 4.2.



Gambar 4.2 Tampilan Hasil Prediksi

4.4 Pengujian

Bagian ini menjelaskan pengujian yang dilakukan terhadap aplikasi prediksi harga saham yang telah dibuat. Pengujian yang dilakukan adalah menguji

pengaruh *hyperparameter* terhadap akurasi model. Pengujian ini diterapkan pada semua *dataset* yang digunakan. Pengujian yang dilakukan dirincikan sebagai berikut.

4.4.1 Pengujian *Training Period* pada *Dataset Utama*

Berdasarkan penelitian [15], *training period* memengaruhi akurasi dari model prediksi yang dibuat. Penggunaan *training period* lima tahun memiliki performa yang lebih baik dibandingkan *training period* satu tahun. Oleh karena itu, pengujian ini dilakukan untuk membuktikan hal tersebut. Pengujian ini akan menggunakan nilai *training period* satu tahun dan nilai *training period* lima tahun.

4.4.2 Pengujian *Unit* pada *Dataset Utama*

Jumlah *unit* dalam satu lapisan memengaruhi akurasi. Semakin besar *unit* yang digunakan maka semakin besar kemungkinan menemukan pola baru. Jumlah *unit* pada model GRU menentukan berapa banyak *hidden state* yang dihasilkan. Pengujian ini dilakukan untuk membuktikan nilai *unit* berpengaruh terhadap akurasi. Nilai-nilai yang digunakan dalam pengujian *unit* adalah 16, 32, 64, dan 128. Pengujian ini mencari nilai *unit* yang menghasilkan akurasi terbaik.

4.4.3 Pengujian *Batch Size* pada *Dataset Utama*

Batch size merupakan salah satu *hyperparameter* yang sering digunakan untuk meningkatkan akurasi. Berdasarkan penelitian [15], angka 32 merupakan *batch size* optimal dalam penelitian tersebut. pengujian terhadap *batch size* dilakukan untuk membuktikan apakah *batch size* berpengaruh terhadap akurasi dan membuktikan apakah nilai 32 juga optimal dalam pengujian ini. Nilai-nilai yang digunakan dalam pengujian *batch size* adalah 32, 64, dan 128. Pengujian ini mencari nilai *batch size* yang menghasilkan akurasi terbaik.

4.4.4 Pengujian *Epoch* pada *Dataset Utama*

Epoch merupakan salah satu *hyperparameter* yang sering digunakan untuk meningkatkan akurasi. Berdasarkan penelitian [15], angka 100 merupakan *epoch* optimal dalam penelitian tersebut. pengujian terhadap *epoch* dilakukan untuk membuktikan apakah *epoch* berpengaruh terhadap akurasi dan membuktikan apakah nilai 100 juga optimal dalam pengujian ini. Nilai-nilai yang digunakan dalam pengujian *epoch* adalah 10, 100, dan 1000. Pengujian ini mencari nilai *epoch* yang menghasilkan akurasi terbaik.

4.4.5 Pengujian Kombinasi Parameter Terbaik pada *Dataset* Utama

Pengujian ini dilakukan untuk mencari kombinasi terbaik untuk masing-masing *dataset*. Pengujian ini dilakukan dengan cara mencoba semua kombinasi yang terbentuk dari nilai-nilai yang diuji pada setiap parameter. Nilai RMSE paling kecil menjadi indikasi bahwa kombinasi parameter tersebut bagus untuk memprediksi harga saham *dataset* tersebut.

4.4.6 Pengujian *Dataset* Pembanding Menggunakan Parameter Terbaik pada *Dataset* Utama

Pengujian ini dilakukan untuk menguji hasil *dataset* pembanding menggunakan parameter *dataset* utama. *Dataset* pembanding merupakan perusahaan yang bergerak dalam satu sektor dengan perusahaan *dataset* utama. Nilai RMSE paling kecil menjadi indikasi bahwa kombinasi parameter tersebut bagus untuk memprediksi harga saham *dataset* tersebut.

4.5 Hasil Pengujian

Bagian ini menjelaskan tentang hasil pengujian yang telah dipaparkan pada bagian 4.4. Hasil pengujian yang dilakukan berupa pengujian *training period*, nilai *unit*, *batch size*, *epoch*, dan kombinasi parameter terbaik.

4.5.1 Hasil Pengujian *Training Period* pada *Dataset* Utama

Pengujian ini membandingkan hasil dari model yang menggunakan data latihan satu tahun dengan lima tahun. Pengujian dilakukan dengan menggunakan parameter terbaik pada *training period* satu tahun untuk *training period* lima tahun pada masing masing *dataset* dan sebaliknya. Hasil tersebut dapat dilihat pada tabel 4.10.

Tabel 4.10 Tabel Nilai RMSE Berdasarkan *Training Period*

No	Dataset	Training period	Parameter			RMSE	Rataan
			Unit	Batch Size	Epoch		
1	ANTM	1 tahun (terbaik)	16	128	1000	0.013073	0.021051
2		1 tahun	128	32	1000	0.029030	
3		5 tahun	16	128	1000	0.014519	0.013329
4		5 tahun (terbaik)	128	32	1000	0.012138	
5	INDF	1 tahun (terbaik)	32	32	100	0.065499	0.100419
6		1 tahun	64	32	1000	0.135338	

7		5 tahun	32	32	100	0.034478	0.0320565
8		5 tahun (terbaik)	64	32	1000	0.029635	
9	TLKM	1 tahun (terbaik)	64	128	100	0.031611	0.035633
10		1 tahun	64	32	1000	0.039654	
11		5 tahun	64	128	100	0.033459	0.032469
12		5 tahun (terbaik)	64	32	1000	0.031479	

Tabel di atas menunjukkan hasil rata-rata nilai RMSE pada masing-masing *dataset*. Hasil tersebut menunjukkan bahwa *training period* lima tahun lebih bagus dibandingkan dengan *training period* satu tahun untuk memprediksi 100 hari ke depan. Pernyataan tersebut didukung oleh hasil penelitian yang dilakukan oleh Rahman [15]. *Training period* terbukti memengaruhi hasil dari nilai RMSE.

4.5.2 Hasil Pengujian Unit pada Dataset Utama

Pengujian ini dilakukan dengan menggunakan asumsi sementara, yaitu nilai *epoch* terbaik adalah 100 dan nilai *batch size* terbaik adalah 32 [15]. Pengujian ini dilakukan terhadap tiga *dataset* perusahaan dengan menggunakan *training period* yang menghasilkan nilai rata-rata RMSE terbaik berdasarkan tabel 4.10. Hasil pengujian dapat dilihat pada tabel 4.11.

Tabel 4.11 Tabel Nilai RMSE Berdasarkan Unit

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	5 tahun	16	32	100	0.017608
2			32			0.023137
3			64			0.017017
4			128			0.019932
5	INDF	5 tahun	16			0.034469
6			32			0.034478
7			64			0.036393
8			128			0.042334
9	TLKM	5 tahun	16			0.047460
10			32			0.046731
11			64			0.047051

12			128			0.046574
----	--	--	-----	--	--	----------

Tabel di atas menunjukkan bahwa parameter unit memiliki pengaruh terhadap nilai RMSE. Akan tetapi, nilai unit yang menghasilkan RMSE terbaik bergantung pada karakteristik *dataset*.

4.5.3 Hasil Pengujian *Batch Size* pada *Dataset* Utama

Pengujian ini dilakukan dengan menggunakan *training period* dan *unit* dengan hasil terbaik yang diperoleh tiap *dataset* berdasarkan pengujian *training period* dan *unit*. Pengujian ini menggunakan asumsi nilai *epoch* terbaik adalah 100 [15]. Hasil pengujian dapat dilihat pada tabel 4.12.

Tabel 4.12 Tabel Nilai RMSE Berdasarkan *Batch Size*

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	5 tahun	64	32	100	0.017017
2				64		0.014184
3				128		0.013360
4	INDF	5 tahun	16	32		0.034469
5				64		0.030491
6				128		0.030132
7	TLKM	5 tahun	128	32		0.046574
8				64		0.034128
9				128		0.034324

Tabel di atas menunjukkan bahwa pada pengujian yang dilakukan terhadap *dataset* PT. Aneka Tambang mendapatkan nilai RMSE terbaik dengan menggunakan nilai *batch size* sebesar 128. Pada hasil pengujian terhadap *dataset* PT. Indofood menunjukkan bahwa nilai *batch size* sebesar 128 menghasilkan nilai RMSE terbaik. Hasil pengujian terhadap *dataset* PT. Telkom menunjukkan hasil yang berbeda, pengujian ini menghasilkan nilai *batch size* 64 menghasilkan nilai RMSE terbaik. Hal ini membuktikan bahwa nilai *batch size* memiliki pengaruh terhadap nilai RMSE. *Batch size* dengan nilai 32 tidak selalu menghasilkan nilai

RMSE yang terbaik.

4.5.4 Hasil Pengujian *Epoch* pada *Dataset* Utama

Pengujian ini dilakukan dengan menggunakan kombinasi *training period*, *unit*, dan *batch size* dengan hasil terbaik yang diperoleh tiap *dataset* berdasarkan pengujian sebelumnya. Hasil pengujian dapat dilihat pada tabel 4.13.

Tabel 4.13 Tabel Nilai RMSE Berdasarkan *Epoch*

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	5 tahun	64	128	10	0.013117
2					100	0.013360
3					1000	0.012673
4	INDF	5 tahun	16	128	10	0.031099
5					100	0.030132
6					1000	0.030281
7	TLKM	5 tahun	128	64	10	0.033184
8					100	0.034128
9					1000	0.033513

Tabel di atas menunjukkan bahwa pengujian terhadap *dataset* PT. Aneka Tambang mendapatkan hasil terbaik dengan menggunakan nilai *epoch* sebesar 1000. Pengujian terhadap *dataset* PT. Indofood mendapatkan hasil terbaik dengan nilai *epoch* sebesar 100. Pengujian terhadap *dataset* PT. Telkom mendapatkan hasil terbaik dengan nilai *epoch* yang berbeda dari pengujian yang lain, yaitu sebesar 10. Hal ini membuktikan bahwa nilai *epoch* berpengaruh terhadap nilai RMSE. Nilai *epoch* 100 tidak selalu menghasilkan nilai RMSE terbaik. Kombinasi parameter pada tabel di atas belum tentu merupakan kombinasi terbaik. Oleh karena itu, pengujian selanjutnya dilakukan untuk membuktikan hal tersebut.

4.5.5 Hasil Pengujian Kombinasi Parameter Terbaik pada *Dataset* Utama

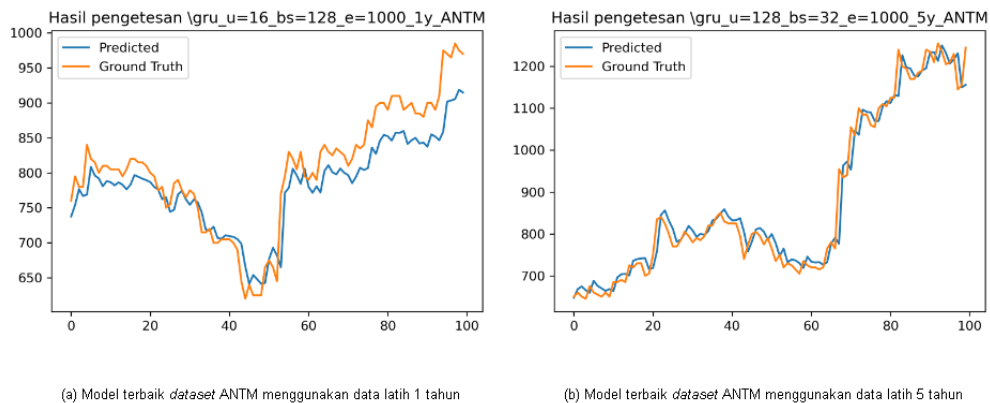
Pengujian ini dilakukan sebanyak kombinasi *training period*, *unit*, *batch size*, dan *epoch* yang dijelaskan pada bagian 4.4. Tabel 4.14 memperlihatkan kombinasi parameter masing-masing *dataset* yang menghasilkan nilai RMSE paling kecil.

Tabel 4.14 Tabel Model dengan RMSE Terbaik

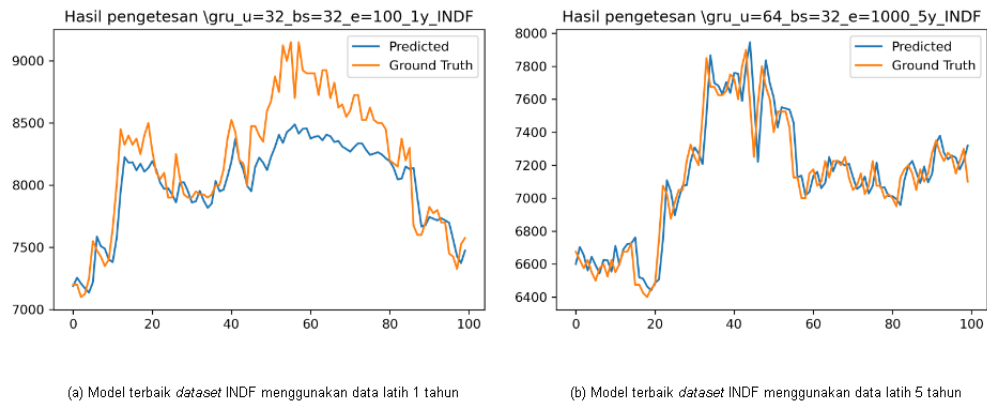
No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	1 tahun	16	128	1000	0.013073
2		5 tahun	128	32	1000	0.012138
3	INDF	1 tahun	32	32	100	0.065499
4		5 tahun	64	32	1000	0.029635
5	TLKM	1 tahun	64	128	100	0.031611
6		5 tahun	64	32	1000	0.031479

Tabel di atas membuktikan bahwa kombinasi pada tabel 4.13 bukan merupakan hasil terbaik. Percobaan pada *dataset* PT. Aneka Tambang menunjukkan model terbaik menggunakan *training period* lima tahun seperti yang dijelaskan pada bagian 4.5.1. Model tersebut menggunakan parameter *unit* sebesar 128, *batch size* sebesar 32 dan *epoch* sebesar 1000. Percobaan pada *dataset* PT. INDF menunjukkan model terbaik menggunakan *training period* lima tahun seperti yang dijelaskan pada bagian 4.5.1. Model tersebut menggunakan parameter *unit* sebesar 64, *batch size* sebesar 32 dan *epoch* sebesar 1000.

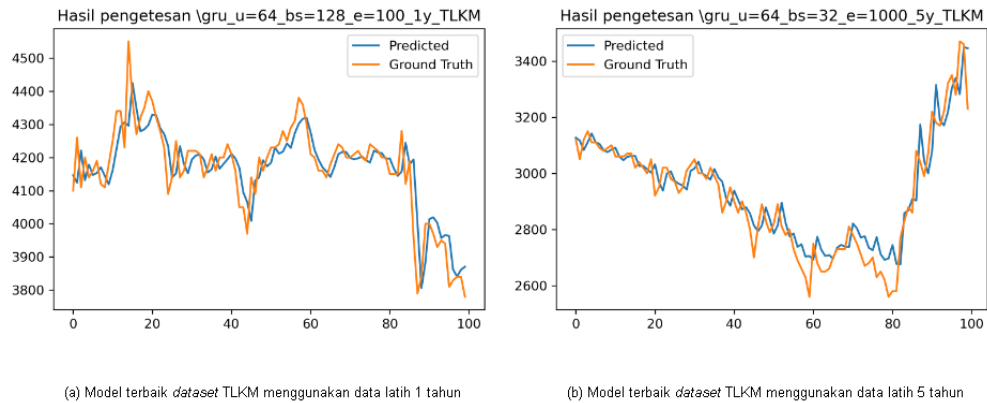
Percobaan pada *dataset* PT. Telkom menunjukkan model terbaik menggunakan *training period* lima tahun seperti yang dijelaskan pada bagian 4.5.1. Model tersebut menggunakan parameter *unit* sebesar 64, *batch size* sebesar 32 dan *epoch* sebesar 1000. Perbandingan hasil prediksi harga saham yang dihasilkan oleh model pada tabel di atas dapat dilihat pada gambar berikut.



Gambar 4.3 Model Terbaik ANTM



Gambar 4.4 Model Terbaik INDF



Gambar 4.5 Model Terbaik TLKM

4.5.6 Hasil Pengujian *Dataset* Pembanding Menggunakan Parameter Terbaik pada *Dataset* Utama

Pengujian ini dilakukan sebanyak kombinasi *training period*, *unit*, *batch size*, dan *epoch* yang dijelaskan pada bagian 4.4. Pengujian ini bertujuan untuk menguji apakah parameter terbaik pada *dataset* utama dapat menghasilkan nilai akurasi terbaik pada *dataset* pembanding dengan catatan perusahaan memiliki sektor yang sama. Berikut adalah hasil percobaan menggunakan parameter terbaik pada *dataset* utama.

Tabel 4.15 Tabel Hasil Terbaik pada *Dataset* Utama

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	5 tahun	128	32	1000	0.012138
2	INDF	5 tahun	64	32	1000	0.029635

3	TLKM	5 tahun	64	32	1000	0.031479
---	------	---------	----	----	------	----------

Tabel 4.16 Tabel Pengujian *Dataset* Pembanding Menggunakan Parameter Terbaik pada *Dataset* Utama

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	INCO	5 tahun	128	32	1000	0.020385
2	JPFA	5 tahun	64	32	1000	0.010212
3	EXCL	5 tahun	64	32	1000	0.024334

Dari tabel di atas menunjukkan bahwa hasil pengujian *dataset* pembanding menggunakan parameter terbaik pada *dataset* utama menghasilkan nilai RMSE yang cukup baik. Setelah menguji setiap kombinasi parameter pada *dataset* pembanding, kombinasi parameter yang menghasilkan nilai RMSE terbaik ditunjukkan pada tabel berikut.

Tabel 4.17 Tabel Model dengan RMSE Terbaik pada *Dataset* Pembanding

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	INCO	1 tahun	16	128	1000	0.035606
2		5 tahun	128	64	10	0.019060
3	JPFA	1 tahun	128	64	1000	0.025999
4		5 tahun	32	32	1000	0.009977
5	EXCL	1 tahun	128	32	1000	0.041279
6		5 tahun	128	128	1000	0.024266

Tabel di atas menjelaskan bahwa parameter terbaik pada *dataset* utama masih bisa diterapkan pada *dataset* pembanding dengan sektor yang sama. Akan tetapi, parameter tersebut tidak menghasilkan nilai RMSE terbaik. *Dataset* pembanding harus dilakukan *hyperparameter tuning*, penyetelan terhadap nilai

parameter, guna menghasilkan nilai RMSE rendah pada arsitektur tersebut.

4.6 Analisis Kesalahan

Pada bagian ini dibahas mengenai analisis kesalahan yang terjadi selama proses pengujian model prediksi harga saham. Kesalahan ini terjadi karena adanya faktor yang memengaruhi model selama proses pengujian. Berikut adalah kesalahan yang dibahas.

4.6.1 Validasi untuk Mendeteksi Masalah *Overfitting* atau *Underfitting*

Salah satu cara untuk mendeteksi masalah *overfitting* atau *underfitting* adalah membandingkan performa hasil prediksi dengan performa saat proses latihan. Seperti yang ditulis pada tabel 4.9, penelitian ini tidak menggunakan validasi saat pembagian data. Hal tersebut menyebabkan masalah *overfitting* atau *underfitting* tidak terdeteksi. Oleh karena itu, validasi diperlukan untuk mendeteksi apakah model yang dibangun mempunyai masalah *overfitting* atau *underfitting* pada saat proses pelatihan model.

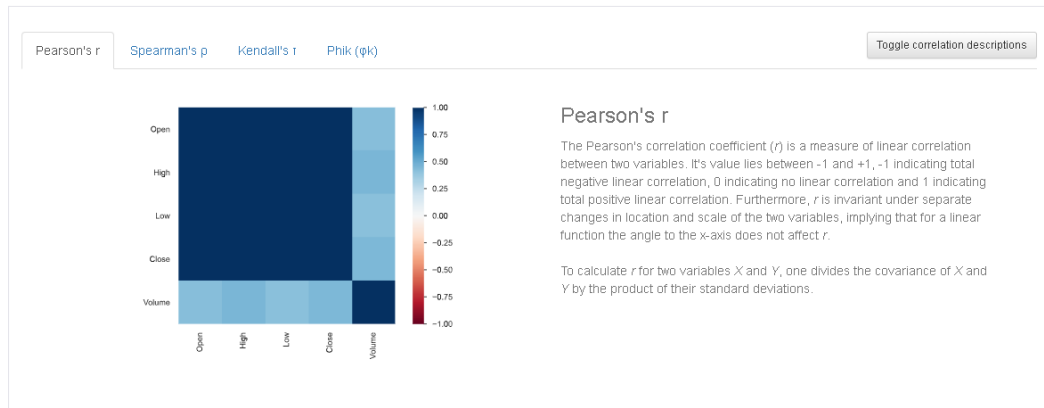
4.6.2 Dataset yang Digunakan

Dataset "Indonesia Stocks" hanya memiliki data dengan periode enam tahun tiga bulan. Dataset ini juga hanya menyediakan 24 data perusahaan. Tabel 4.9 memperlihatkan bahwa data yang digunakan untuk pengujian *training period* lima tahun lebih sedikit dibandingkan data pelatihannya. Semakin lama periode dan semakin banyak data perusahaan yang ada dalam *dataset* maka semakin banyak percobaan yang dilakukan.

4.6.3 Seleksi Fitur

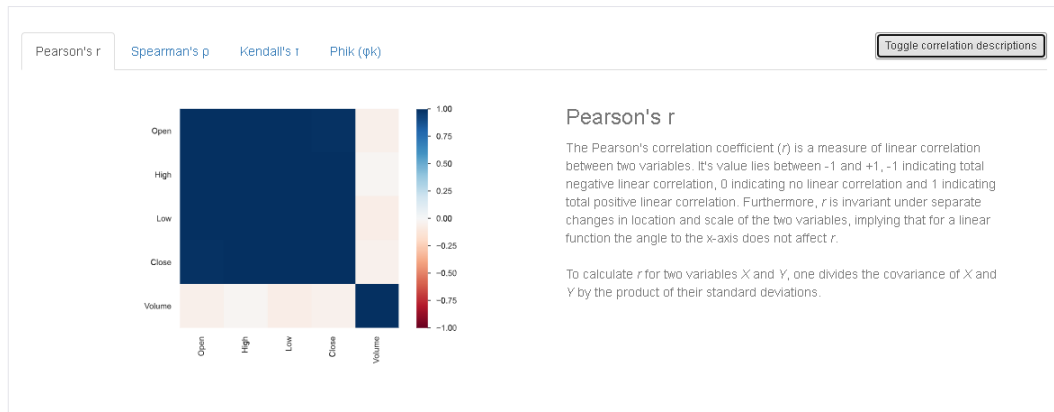
Penelitian ini menggunakan semua fitur yang disediakan dalam *dataset* "Dataset Saham Indonesia / Indonesia Stock Dataset". Fitur yang digunakan, yaitu *open*, *close*, *high*, *low*, dan *volume*. Fitur tersebut digunakan untuk memprediksi harga *close* untuk hari selanjutnya. Terdapat kemungkinan bahwa fitur yang digunakan tidak memiliki pengaruh besar terhadap nilai *close* hari selanjutnya. Oleh karena itu, penelitian ini mencoba untuk melihat hubungan antar fitur dengan menggunakan korelasi Pearson.

Correlations



Gambar 4.6 Korelasi Fitur pada *Dataset ANTM*

Correlations



Gambar 4.7 Korelasi Fitur pada *Dataset INDF*

Correlations



Gambar 4.8 Korelasi Fitur pada *Dataset TLKM*

Tiga gambar di atas menunjukkan bahwa nilai *volume* tidak memiliki hubungan terhadap nilai *close*. Salah satu cara agar hal ini tidak terjadi adalah

dengan menggunakan seleksi fitur. Menurut Penelitian yang dilakukan oleh S. Chen, penerapan seleksi fitur pada *dataset* memengaruhi hasil akurasi model tersebut [6].

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, menghasilkan kesimpulan yang menjawab pertanyaan pada rumusan masalah. Kesimpulan dibagi menjadi tiga poin, yaitu:

1. Percobaan menghasilkan akurasi terbaik sebesar 0.009977. Akurasi tersebut didapatkan melalui percobaan yang dilakukan pada *dataset* pembandingan PT. Japfa Comfeed Indonesia. Nilai tersebut dihasilkan dengan menggunakan *training period* lima tahun, *unit* sebesar 32, *batch size* sebesar 32 dan *epoch* sebesar 1000. Percobaan yang dilakukan pada *dataset* PT. Aneka Tambang memiliki akurasi terbaik sebesar 0.012138. Percobaan yang dilakukan pada *dataset* PT. Vale Indonesia menghasilkan akurasi terbaik sebesar 0.019060. Pengujian terhadap *dataset* PT. XL Axiata menghasilkan akurasi terbaik sebesar 0.024266. Percobaan pada *dataset* PT. Indofood menghasilkan akurasi terbaik sebesar 0.029635. Terakhir, PT. Telkom menghasilkan akurasi terbaik sebesar 0.031479.
2. Indikator *training period*, *unit*, *batch size*, dan *epoch* berpengaruh terhadap akurasi model prediksi harga saham. Setiap *dataset* memiliki kombinasi nilai indikator yang berbeda untuk menghasilkan performa yang baik. Menggunakan *dataset* untuk proses pelatihan dalam jumlah besar belum tentu menghasilkan performa yang lebih baik dibandingkan menggunakan *dataset* dalam jumlah kecil. Nilai parameter *unit* tidak berhubungan dengan panjang data latih yang digunakan. Nilai *batch size* dan *epoch* optimal bergantung pada *dataset* yang digunakan.
3. Parameter pada *dataset* utama dapat digunakan untuk *dataset* pembandingan yang memiliki sektor yang sama. Namun, parameter tersebut tidak menghasilkan nilai akurasi terbaik.

5.2 Saran

Saran untuk pengembangan yang dilakukan untuk sistem prediksi harga saham dengan menggunakan metode GRU adalah sebagai berikut:

1. Pada proses pelatihan, validasi ditambahkan untuk mendeteksi model

overfitting dan *underfitting* lebih awal.

2. *Dataset* yang lebih banyak datanya berguna untuk memberikan fleksibilitas terhadap percobaan yang dilakukan terhadap data tersebut.
3. Seleksi fitur dilakukan pada penelitian selanjutnya untuk memastikan agar model hanya menggunakan fitur yang berpengaruh terhadap fitur yang ingin diprediksi.
4. Penelitian selanjutnya dapat melihat pengaruh parameter *dataset* utama terhadap *dataset* pembanding yang memiliki sektor yang berbeda.

DAFTAR REFERENSI

- [1] Setiani Widiarti, ANALISA PENILAIAN DAN PERHITUNGAN HARGA SAHAM. April 2020. Available https://www.researchgate.net/publication/340595378_ANALISA_PENILAIAN_DAN_PERHITUNGAN_HARGA_SAHAM [Accessed: 11 May 2022]
- [2] Yuneita Anisma, Maret 2012. Faktor-faktor yang mempengaruhi harga saham perusahaan perbankan yang listing di Bursa Efek Indonesia, Vol 2, No. 5. [Accessed: 11 October 2021]
- [3] Fama, E. F., Efficient capital markets: A review of theory and empirical work. The Journal of Finance 25 (2), 383-417. 1970. [Accessed: 11 October 2021]
- [4] Guizhu Shen, Qingping Tan, Haoyu Zhang, Ping Zeng, Jianjun Xu, Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions, Procedia Computer Science, Volume 131, Pages 895-903, May 2018, <https://doi.org/10.1016/j.procs.2018.04.298>. [Accessed: 27 September 2021]
- [5] R. Ren, D. D. Wu and T. Liu, "Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine," in IEEE Systems Journal, vol. 13, no. 1, pp. 760-770, March 2019, doi: 10.1109/JSYST.2018.2794462. [Accessed: 4 October 2021]
- [6] S. Chen and C. Zhou, "Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network," in IEEE Access, vol. 9, pp. 9066-9072, 2021, doi: 10.1109/ACCESS.2020.3047109. [Accessed: 4 October 2021]
- [7] Sethia A., Raut P. Application of LSTM, GRU and ICA for Stock Price Prediction. In: Satapathy S., Joshi A. (eds) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 107. Springer, Singapore. 2019. https://doi.org/10.1007/978-981-13-1747-7_46 [Accessed: 4 October 2021]
- [8] A. Gulli, A. Kapoor, and S. Pal, Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API, 2nd Edition, 2nd ed. Birmingham, England: Packt Publishing, 2019. [Accessed: 12 March 2022]

- [9] A. Geron, Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems, 2nd ed. Sebastopol, CA: O'Reilly Media, 2019. [Accessed: 12 March 2022]
- [10] Pandas Development Team, "Getting started - pandas 1.4.1 documentation," Pandas documentation, Feb 12, 2022. [Online]. Available: https://pandas.pydata.org/docs/getting_started/index.html. [Accessed: 26 March 2022]
- [11] NumPy community, "NumPy: the absolute basics for beginners," NumPy documentation, Jan 14, 2022. [Online]. Available: https://numpy.org/doc/1.22/user/absolute_beginners.html. [Accessed: 26 March 2022]
- [12] John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team, "Pyplot function overview," matplotlib.pyplot. [Online]. Available: https://matplotlib.org/stable/api/pyplot_summary.html. [Accessed: 26 March 2022]
- [13] Keras team on Google, "Keras API Reference," Keras documentation. [Online]. Available: <https://keras.io/api/>. [Accessed: 26 March 2022]
- [14] Y. Ji, A. W. -C. Liew and L. Yang, "A Novel Improved Particle Swarm Optimization With Long-Short Term Memory Hybrid Model for Stock Indices Forecast," in IEEE Access, vol. 9, pp. 23660-23671, 2021, doi: 10.1109/ACCESS.2021.3056713. [Accessed: 4 October 2021]
- [15] Rahman, M.O., Hossain, M.S., Junaid, T.S., Forhad, M.S.A. and Hossen, M.K., "Predicting prices of stock market using gated recurrent units (GRUs) neural networks," Int. J. Comput. Sci. Netw, Secur, 19(1), pp.213-222, 2019 [Accessed: 12 March 2022]
- [16] Achelis, S.B., Technical analysis from A to Z: covers every trading tool from the absolute breath index to the zig zag (No. E70 70). McGraw-Hill. 2001.[Accessed: 12 May 2022]