

***DEPLOYMENT MODEL PREDIKSI HARGA SAHAM  
APPLE INC PADA BEBERAPA BURSA EFEK  
MENGUNAKAN METODE MULTIVARIATE GATED  
RECURRENT UNIT***

**SKRIPSI**



**CECILIA TANIA EMANUELLA**

**H071181002**

**Pembimbing Utama : Dr. Eng. Armin Lawi, S.Si., M.Eng.**  
**Pembimbing Pendamping : Dr. Hendra, S.Si., M.Kom.**  
**Penguji : 1. A. Muh. Amil Siddik, S.Si., M.Si.**  
**2. Andi Muhammad Anwar, S.Si., M.Si.**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2022**

***DEPLOYMENT MODEL PREDIKSI HARGA SAHAM  
APPLE INC PADA BEBERAPA BURSA EFEK  
MENGUNAKAN METODE MULTIVARIATE GATED  
RECURRENT UNIT***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains  
pada Program Studi Sistem Informasi Departmen Matematika Fakultas  
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

**CECILIA TANIA EMANUELLA  
H071181002**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2022**

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Cecilia Tania Emanuella  
NIM : H071181002  
Program Studi : Sistem Informasi  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

***DEPLOYMENT MODEL PREDIKSI HARGA SAHAM APPLE  
INC PADA BEBERAPA BURSA EFEK MENGGUNAKAN  
METODE MULTIVARIATE GATED RECURRENT UNIT***

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa Sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 26 Agustus 2022  
Yang menyatakan,



**Cecilia Tania Emanuella**  
NIM: H071181002

***DEPLOYMENT MODEL PREDIKSI HARGA SAHAM APPLE  
INC PADA BEBERAPA BURSA EFEK MENGGUNAKAN  
METODE MULTIVARIATE GATED RECURRENT UNIT***

Disusun dan diajukan oleh

**CECILIA TANIA EMANUELLA**

**H071181002**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal Agustus 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama



**Dr. Eng. Armin Irawi, S.Si., M.Eng.**  
NIP: 19720423 199512 1 001

Pembimbing Pertama



**Dr. Hendra, S.Si., M.Kom.**  
NIP: 19760102 200212 1 001



## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Cecilia Tania Emanuella  
NIM : H071181002  
Program Studi : Sistem Informasi  
Judul Skripsi : *Deployment Model Prediksi Harga Saham Apple Inc Pada Beberapa Bursa Efek Menggunakan Metode Multivariate Gated Recurrent Unit*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

### DEWAN PENGUJI

Tanda Tangan

Ketua	: Dr. Eng. Ar min Lawi, S.Si., M.Eng.	(.....)
Sekretaris	: Dr. Hendra, S.Si., M.Kom.	(.....)
Anggota	: A. Muh. Amil Siddik, S.Si., M.Si.	(.....)
Anggota	: Andi Muhammad Anwar, S.Si., M.Si.	(.....)

Ditetapkan di: Makassar

Tanggal: 26 Agustus 2022



## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa oleh karena anugerah-Nya yang melimpah, kemurahan dan kasih setia yang besar sehingga penulis dapat menyelesaikan skripsi yang berjudul “*Deployment Model Prediksi Harga Saham Apple Inc Pada Beberapa Bursa Efek Menggunakan Metode Multivariate Gated Recurrent Unit*”. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Sains pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua tercinta, Ayahanda Albert Parrang, dan Ibunda Windhy Krisnasari, yang dengan penuh kesabaran dalam mengasuh dan mendidik penulis, yang senantiasa mencurahkan kasih sayang yang tak pernah putus serta memberikan dukungan doa yang tulus sehingga penulis dapat menyelesaikan skripsi ini. Ucapan terima kasih juga kepada saudara-saudari tercinta, Kanesya Regina Ursula dan Jessica Anastacia Aprilia serta seluruh keluarga yang senantiasa memberikan dukungan dan doa bagi penulis dalam menyelesaikan skripsi ini.

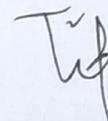
Penulisan skripsi ini dapat terselesaikan berkat bantuan dan motivasi dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan yang tulus kepada:

1. Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc., selaku Rektor Universitas Hasanuddin.
2. Bapak Dr. Eng. Amiruddin, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
3. Bapak Dr. Eng. Armin Lawi, S.Si., M.Si., selaku pembimbing utama dan Bapak Dr. Hendra, S.Si., M.Kom., selaku pembimbing pertama untuk segala ilmu, nasihat, dan kesabaran dalam membimbing dan mengarahkan penulis, serta bersedia meluangkan waktunya untuk mendampingi penulis sehingga skripsi ini dapat terselesaikan.

4. Bapak A. Muh. Amil Siddik, S.Si., M.Si. dan Bapak Andi Muhammad Anwar, S.Si., M.Si., selaku tim penguji yang telah bersedia meluangkan waktunya untuk memberikan saran dan arahan kepada penulis dalam penulisan skripsi ini.
5. Bapak/Ibu Dosen Pengajar Departemen Matematika terkhusus kepada Bapak/Ibu Dosen Pengajar Program Studi Sistem Informasi yang telah membekali ilmu kepada penulis selama menjadi mahasiswa di Program Studi Sistem Informasi serta seluruh staf atas bantuannya dalam pengurusan akademik selama ini.
6. Teman-teman Sistem Informasi 2018 seperjuangan atas dukungan, kebersamaan dan kerjasamanya selama kurang lebih 4 tahun di Universitas Hasanuddin dan yang telah mendukung dan berjuang selama ini.
7. Keluarga besar GMKI Kom. FMIPA UH dan teman – teman MIPA KRISTEN 2018 yang senantiasa mendukung dalam doa.
8. *Google, Stackoverflow, GeeksforGeeks*, dan *website – website* lainnya yang telah membantu penulis dalam membantu menyelesaikan skripsi ini.
9. Serta segala pihak yang tidak dapat disebutkan satu persatu, yang telah membantu penulis dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga skripsi ini dapat bermanfaat bagi yang membacanya, terutama untuk pengembangan ilmu pengetahuan.

Makassar, 16 Agustus 2022



Penulis

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIS**

---

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Cecilia Tania Emanuella  
NIM : H071181002  
Program Studi : Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

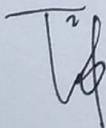
**Deployment Model Prediksi Harga Saham Apple Inc Pada Beberapa Bursa  
Efek Menggunakan Metode Multivariate Gated Recurrent Unit**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 2 Agustus 2022

Yang menyatakan



(Cecilia Tania Emanuella)

## ABSTRAK

Saham menjadi salah satu instrumen pasar keuangan dan investasi yang banyak diminati oleh investor. Pergerakan harga saham yang nonlinear dan nonstasioner yang dipengaruhi oleh banyak faktor, sehingga sangat sulit untuk meramalkan harga saham. Salah satu metode yang cocok untuk digunakan dalam memprediksi harga saham yang bersifat nonlinear adalah *Recurrent Neural network* (RNN). Penelitian ini menggunakan salah satu arsitektur RNN yaitu *Gated Recurrent Unit* (GRU) untuk memprediksi harga saham dengan menggunakan model *multivariate* GRU. Evaluasi kinerja yang digunakan untuk mengukur tingkat kesalahan prediksi adalah MAE, RMSE, MAPE dan RMSPE. Berdasarkan hasil MAE dan RMSE dari AAPL(*Nasdaq*), APC.F(*Frankfurt*) dan AAPL.MX(*Mexico*) memberikan hasil tingkat kesalahan yang bernilai kecil. Adapun berdasarkan hasil evaluasi MAPE dan RMSPE memberikan hasil yang sangat baik dengan masing – masing persentase kesalahan yang dihasilkan  $< 10\%$ . Kemudian didapatkan hasil yang baik pada hasil evaluasi RMSPE prediksi data *train* pada AAPL.MX(*Mexico*) dengan masing – masing persentase kesalahan yang dihasilkan berada pada  $10\% \leq \text{RMSPE} < 20\%$ . Kemudian hasil model di *deploy* ke dalam *website* menggunakan *framework* *Streamlit* yang menampilkan proses prediksi dengan menampilkan data, visualisasi data prediksi, hasil evaluasi dari model dan terakhir memberikan hasil prediksi pada satu hari bursa berikutnya untuk harga *low*, *open*, *close* dan *high*.

Kata kunci: Harga Saham, Prediksi, *Multivariate* GRU, *Deploy*, *Website*, *Streamlit*.

## ***ABSTRACT***

Stocks are one of the financial and investment market instruments that are in great demand by investors. Stock price movements are nonlinear and nonstationary which are influenced by many factors, making it very difficult to forecast stock prices. One method that is suitable to be used in predicting stock prices that are nonlinear is Recurrent Neural network (RNN). This study uses one of the RNN architecture that is Gated Recurrent Unit (GRU) to predict stock prices using multivariate GRU model. Performance evaluation used to measure the level of prediction error is MAE, RMSE, MAPE and RMSPE. Based on MAE and RMSE results from AAPL(Nasdaq), APC.F(Frankfurt) dan AAPL.MX(Mexico) provides error rate results that are of little value. Based on the results of MAPE and RMSPE evaluation, it gives very good results with each percentage of errors produced  $< 10\%$ . Then obtained good results on the results of the RMSPE prediction data train evaluation on AAPL.MX (Mexico) with each generated error percentage being at  $10\% \leq \text{RMSPE} < 20\%$ . Then the results of the model are deployed to the website using the Streamlit framework that displays the prediction process by displaying data, visualizing prediction data, evaluating the results of the model and finally providing prediction results on the next exchange day for low, open, close and high prices.

*Keywords: Stock Price, Prediction, Multivariate GRU, Deploy, Website, Streamlit.*

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERNYATAAN KEASLIAN .....	ii
HALAMAN PERSETUJUAN PEMBIMBING .....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR .....	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR .....	vii
ABSTRAK.....	viii
<i>ABSTRACT</i> .....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	5
1.3    Batasan Masalah.....	5
1.4    Tujuan Penelitian.....	6
1.5    Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1    Prediksi <i>Stock</i> / Saham .....	7
2.2 <i>Recurrent Neural Network</i> .....	8
2.3 <i>Gated Recurrent Unit</i> .....	9
2.3.1 <i>Reset Gate</i> .....	11
2.3.2 <i>Update Gate</i> .....	12
2.3.3 <i>Candidate Hidden State</i> .....	13
2.3.4 <i>Hidden State</i> .....	14
2.4    Metode Evaluasi Kinerja Model.....	14
2.4.1 <i>Loss Function</i> .....	15
2.4.2 <i>Mean Absolute Error (MAE)</i> .....	15
2.4.3 <i>Root Mean Square Error (RMSE)</i> .....	16
2.4.4 <i>Mean Absolute Percentage Error (MAPE)</i> .....	16
2.4.5 <i>Root Mean Square Percentage Error (RMSPE)</i> .....	16
2.5    Inisialisasi <i>Hyperparameter</i> .....	17
2.5.1 <i>Optimizer dan Learning rate</i> .....	17

2.5.2	<i>Batch size</i> .....	18
2.5.3	<i>Epoch</i> .....	18
2.5.4	<i>Hidden Layer dan Neuron</i> .....	18
2.6	<i>Dropout</i> .....	19
2.7	Normalisasi Data .....	20
2.8	<i>Deployment Model</i> .....	20
BAB III	METODE PENELITIAN.....	22
3.1	Waktu dan Lokasi Penelitian.....	22
3.2	<i>Dataset</i> .....	22
3.3	Instrumen Penelitian.....	23
3.4	Tahapan Penelitian .....	23
3.5	<i>Preprocessing</i> .....	24
3.6	<i>Split Data</i> .....	24
3.7	<i>Training Multivariate GRU</i> .....	24
3.8	Evaluasi .....	27
3.9	<i>Deployment Model</i> .....	27
BAB IV	HASIL DAN PEMBAHASAN .....	29
4.1	Deskripsi Data .....	29
4.2	<i>Preprocessing</i> .....	31
4.2.1	Konversi Harga Saham .....	31
4.2.2	Normalisasi Data.....	32
4.2.3	Segmentasi Data.....	34
4.3	<i>Split Data</i> .....	36
4.4	<i>Modeling Multivariate GRU</i> .....	36
4.5	<i>Training Data</i> .....	37
4.6	Evaluasi .....	43
4.7	<i>Deployment Model</i> .....	46
BAB V	KESIMPULAN DAN SARAN.....	50
5.1	Kesimpulan.....	50
5.2	Saran .....	51
DAFTAR PUSTAKA	.....	52
LAMPIRAN	.....	55

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur <i>Recurrent Neural Network</i> .....	8
Gambar 2.2	Arsitektur <i>Gated Recurrent Unit</i> .....	9
Gambar 2.3	Grafik Fungsi Aktivasi <i>Sigmoid</i> dan <i>Tanh</i> .....	10
Gambar 2.4	<i>Reset Gate</i> .....	11
Gambar 2.5	<i>Update Gate</i> .....	12
Gambar 2.6	<i>Candidate Hidden State</i> .....	13
Gambar 2.7	<i>Output Hidden State</i> .....	14
Gambar 2.8	Arsitektur <i>Neural Network</i> .....	19
Gambar 2.9	Perbedaan Jaringan Syaraf Biasa dan Setelah Teknik <i>Dropout</i> .....	20
Gambar 3.1	Data Harga Saham AAPL pada <i>Website Yahoo Finance</i> .....	22
Gambar 3.2	Diagram Alur Penelitian .....	23
Gambar 3.3	Arsitektur Model .....	25
Gambar 3.4	Rancangan Sistem Web .....	28
Gambar 4.1	Grafik <i>Time Series</i> Saham AAPL( <i>Nasdaq</i> ) .....	30
Gambar 4.2	Visualisasi Data Saham AAPL( <i>Nasdaq</i> ) Sebelum Normalisasi .....	33
Gambar 4.3	Visualisasi Data Saham AAPL( <i>Nasdaq</i> ) Setelah dilakukan Normalisasi Data .....	34
Gambar 4.4	Gambaran Segmentasi 40 Data pada Saham AAPL( <i>Nasdaq</i> ), APC.F( <i>Frankfurt</i> ) dan AAPL.MX( <i>Mexico</i> ) .....	35
Gambar 4.5	Arsitektur <i>Multivariate GRU</i> .....	37
Gambar 4.6	Visualisasi Hasil Prediksi Data <i>Training</i> pada Harga Saham AAPL( <i>Nasdaq</i> ) .....	38
Gambar 4.7	Visualisasi Hasil Prediksi Data <i>Training</i> pada Harga Saham APC.F( <i>Frankfurt</i> ) .....	39
Gambar 4.8	Visualisasi Hasil Prediksi Data <i>Training</i> pada Harga Saham AAPL.MX( <i>Mexico</i> ) .....	39
Gambar 4.9	Visualisasi <i>Training Loss</i> dan <i>Validation Loss</i> pada Harga Saham AAPL( <i>Nasdaq</i> ) .....	40
Gambar 4.10	Visualisasi <i>Training Loss</i> dan <i>Validation Loss</i> pada Harga Saham APC.F( <i>Frankfurt</i> ) .....	41
Gambar 4.11	Visualisasi <i>Training Loss</i> dan <i>Validation Loss</i> pada Harga Saham AAPL.MX( <i>Mexico</i> ) .....	42
Gambar 4.12	Visualisasi Hasil Prediksi Data <i>Testing</i> pada Harga Saham AAPL( <i>Nasdaq</i> ) .....	43
Gambar 4.13	Visualisasi Hasil Prediksi Data <i>Testing</i> pada Harga Saham APC.F( <i>Frankfurt</i> ) .....	44
Gambar 4.14	Visualisasi Hasil Prediksi Data <i>Testing</i> pada Harga Saham AAPL.MX( <i>Mexico</i> ) .....	44
Gambar 4.15	Implementasi Perancangan Tampilan <i>Sidebar</i> .....	47
Gambar 4.16	Implementasi Proses Prediksi pada Perancangan <i>Website</i> .....	47
Gambar 4.17	Proses <i>Deployment Website</i> di <i>Streamlit</i> .....	48
Gambar 4.18	Hasil Rancangan <i>Website</i> .....	48
Gambar 4.19	Pengujian <i>Website</i> dalam Melakukan Prediksi .....	49

## DAFTAR TABEL

Tabel 4.1 Data Harga Saham AAPL( <i>Nasdaq</i> ) .....	29
Tabel 4.2 Data Saham APC.F( <i>Frankfurt</i> ) Sebelum dan Sesudah Konversi Harga Saham ke USD.....	31
Tabel 4.3 Data Saham AAPL( <i>Nasdaq</i> ) Sebelum dan Sesudah dilakukan Normalisasi Data .....	32
Tabel 4.4 Segmentasi Data Saham AAPL( <i>Nasdaq</i> ) .....	35
Tabel 4.5 Komposisi Data Saham .....	36
Tabel 4.6 Evaluasi MAE dan RMSE .....	45
Tabel 4.7 Evaluasi MAPE dan RMSPE.....	46

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam membentuk suatu perekonomian pada suatu negara, pasar modal merupakan salah satu komponen yang cukup penting. Di masa hingga saat ini kesadaran akan pentingnya perencanaan keuangan selalu meningkat dari tahun ke tahun. Salah satu bentuk perencanaan keuangan yang cukup populer adalah investasi. Investasi merupakan aktivitas menanam modal dengan harapan akan memberikan keuntungan pada masa depan. Saham menjadi salah satu instrumen pasar keuangan dan investasi yang banyak diminati oleh investor. Saham dikenal mampu memberikan tingkat keuntungan yang besar atau *capital gain* yang didapatkan dari selisih harga jual dan harga beli saham, tetapi risiko fluktuasi harga saham yang sangat cepat dapat mengakibatkan kerugian yang besar dalam waktu yang singkat. Adanya fluktuasi pergerakan saham menimbulkan ketidakpastian dalam pasar modal yang dapat menimbulkan risiko kerugian dalam pembelian saham (Fauzi, 2019). Transaksi jual beli sangat krusial dan penting sehingga memerlukan analisis secara fundamental terhadap pergerakan harga saham. Proses analisis seperti ini biasanya membutuhkan waktu cukup lama serta membutuhkan analisa teknikal. Oleh karena itu prediksi terkait pergerakan saham diperlukan untuk menjadi alat bantu bagi perencanaan yang lebih efektif dan efisien untuk mengetahui pergerakan selanjutnya sehingga dapat mengurangi risiko kerugian dan potensi keuntungan dapat dimaksimalkan.

Sampai saat ini berbagai penelitian yang bertujuan untuk memprediksi suatu kejadian di masa mendatang dengan tingkat akurasi tinggi masih terus dilakukan. Bagaimana meningkatkan akurasi prediksi harga indeks saham dianggap sebagai salah satu topik paling menarik di pasar saham (Gao P. dkk., 2020). Tidak ada yang pernah mengetahui dengan tepat apa yang akan terjadi di masa yang akan datang, namun manusia dengan kemampuan berpikirnya berusaha untuk dapat memperkirakan kejadian di masa depan dengan berbagai metode dan analisa untuk memperoleh tingkat kesalahan yang sekecil mungkin (Karno, 2020). Prediksi deret

waktu bertujuan untuk membangun model untuk mensimulasikan nilai masa depan yang diberikan dari nilai masa lalu.

Peramalan harga aset keuangan merupakan tantangan yang kompleks. Secara umum, harga saham ditentukan oleh jumlah variabel yang tidak terbatas. Dalam model prediksi pasar saham dapat mempertimbangkan hubungan antara harga penutupan dan harga pembukaan, rata-rata pergerakan, harga tertinggi harian, harga saham lain, siklus ekonomi, perkembangan politik, kejadian tak terduga, faktor psikologis, sentimen pasar, dan bahkan cuaca, semua variabel ini dapat mempengaruhi harga (Müller, 2020).

*Deep learning* adalah salah satu bentuk *machine learning* yang perkembangannya cukup pesat, *deep learning* juga merupakan bagian dari *machine learning* yang merupakan salah satu dasar dari *artificial intelligence*. Salah satu metode yang cocok untuk digunakan dalam memprediksi harga saham yang bersifat nonlinear adalah *Recurrent Neural network* (RNN) yang melakukan perulangan pada prosesnya. Beberapa penelitian algoritma *deep learning* telah banyak diimplementasikan dalam bidang *financial*. Penelitian yang dilakukan oleh Obaidur dkk., pada 2019 melakukan prediksi harga saham *LOW*, *KO* dan *AAPL* menggunakan *GRU* dengan empat masukan dan satu *output* untuk evaluasi menggunakan *RMSE*. Penelitian yang dilakukan oleh Dwiyanto dkk., pada 2019 melakukan prediksi harga saham menggunakan metode *LSTM* dengan menggunakan analisis historis harga saham dalam suatu perusahaan. Dalam penelitian tersebut mengidentifikasi fitur harga terendah, harga tertinggi, harga buka, harga tutup, *volume*, rata-rata harga, dan pergerakan. Prediksi tujuh fitur variabel dengan RNN menghasilkan akurasi sebesar 94% untuk data latih dan 55% untuk data uji. Akurasi diperoleh setelah pelatihan dengan menggunakan 1218 data. Penelitian yang dilakukan oleh Zayini dan Habibi, pada 2020 melakukan analisis prediksi harga *adjusted close* pada saham *AAPL* menggunakan metode *RNN/LSTM* dengan evaluasi menggunakan *RMSE*. Penelitian yang dilakukan oleh Tanwar dkk., pada 2021 melakukan prediksi *cryptocurrencies* menggunakan *GRU* dan *LSTM*, evaluasi menggunakan *RMSE*. Pada penelitian ini algoritma *deep learning* telah menunjukkan hasil yang akurat dalam prediksi berbagai pasar keuangan dan kedepannya akan membuat prediksi dengan menggunakan lebih dari

satu *inter-dependency* (Tanwar dkk., 2021). Penelitian yang dilakukan oleh Islam dan Hossain, pada 2021 memprediksi pasar valuta asing menggunakan algoritma LSTM dan GRU, untuk evaluasi hasil menggunakan *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE) dan skor *R-squared* (R<sup>2</sup>). Penelitian yang dilakukan oleh Siringoringo, pada 2021 memprediksi tingkat inflasi nasional Indonesia menggunakan GRU untuk evaluasi menggunakan MAPE. Penelitian yang dilakukan oleh Gao Y. dkk., pada 2021 memprediksi harga saham *Shanghai Composite Index* dengan indikator sentimen pasar dan data keuangan menggunakan LSTM dan GRU dengan metode reduksi dimensi LASSO dan PCA untuk evaluasi menggunakan MAE, MSE dan RMSE. Penelitian yang dilakukan oleh Karno, pada 2020 memprediksi harga saham BRI menggunakan LSTM dengan evaluasi RMSE. Penelitian yang dilakukan oleh Fauzi, pada 2019 memprediksi harga saham syariah menggunakan LSTM dengan evaluasi MSE dan RMSE. Penelitian yang dilakukan oleh Khalis dkk., pada 2021 membandingkan algoritma *linear regresi*, LSTM dan GRU dalam memprediksi harga saham KEJU dengan evaluasi MAE, MSE dan RMSE. Penelitian yang dilakukan oleh Satyo dkk., pada 2020 memprediksi harga saham BCA, BRI dan BMRI menggunakan LSTM dan GRU dengan evaluasi MSE dan RMSE. Penelitian yang dilakukan oleh Ghudafa dkk., pada 2022 memprediksi harga saham JKSE dengan indikator *Combine News DJIA* menggunakan LSTM dan GRU dengan *transformer* untuk evaluasi menggunakan *report accuracy*. Penelitian yang dilakukan oleh Arfan dan ETP, pada 2019 memprediksi harga saham UNVR menggunakan LSTM untuk evaluasi menggunakan MSE. Penelitian yang dilakukan oleh Gao P. dkk., pada 2020 memprediksi harga saham SP500, Nikkei, dan CSI300 dengan indikator MACD, ATR, *Exchange Rate* dan *Interest Rate* menggunakan MLP, LSTM, CNN dan UA untuk evaluasi menggunakan RMSE, *R-squared* dan MAPE. Penelitian yang dilakukan oleh Hastomo dkk., pada 2021 memprediksi harga saham BBCA, BBRI, INDF, ASII, JSMR, PTBA, TLKM, UNTR dan UNVR dengan indikator korelasi menggunakan LSTM-GRU untuk evaluasi menggunakan MSE dan RMSE.

Penelitian yang dilakukan oleh Rahmi dkk., pada 2015 menggunakan hasil optimasi algoritma genetika dan menggunakan model regresi dalam memprediksi harga saham di masa mendatang dengan data historis harga saham yang digunakan

adalah harga rata-rata saham harian (rata-rata harga *open* dan harga *close*). Prediksi terbaik dengan nilai MSE terkecil 47.5023, dengan tingkat akurasi sebesar 73.78% hanya kurang realistis dalam memprediksi harga saham sehingga akurasi yang didapatkan tidak sesuai dengan nilai harga saham sebenarnya. Kemudian penelitian lain yang dilakukan oleh Huang dkk., pada 2018 menggunakan lima fitur yang disertai penambahan fitur *volume* menggunakan metode algoritma *Koordinate Sub Mode* mendapatkan tingkat akurasi sebesar 62.29% dan dibutuhkan komputasi yang besar serta waktu yang lama. Penelitian yang dilakukan oleh Rochman dan Djunaidy pada 2014 menggunakan dua pendekatan dalam memprediksi harga saham yaitu analisa teknikal yang berdasarkan data masa lampau dan faktor eksternal yang berdasarkan kondisi makroekonomi (tingkat inflasi, tingkat suku bunga, harga minyak), kondisi nonekonomi (pergerakan indeks saham luar negeri) dan kondisi perusahaan. Hasil penelitian prediksi harga penutupan dan jumlah saham yang diperdagangkan dengan menggunakan jaringan syaraf tiruan didapatkan perbandingan nilai RMSE antara saham yang dikombinasi dengan faktor eksternal adalah 0.00271 lebih kecil dibandingkan dengan yang hanya menggunakan data historis saham saja yaitu sebesar 0.00293. Penelitian yang dilakukan Santi dan Widodo pada 2021 menggunakan data historis harga *open*, *high*, *low*, *close* dan jumlah *volume* transaksi dengan target harga *close*. Melakukan prediksi dengan algoritma *neural network backpropagation* berdasarkan teknik analisa teknikal yang mengacu pada data historis, didapatkan hasil evaluasi RMSE untuk saham BBCA sebesar 123.84, EXCL sebesar 37.36 dan PGLI sebesar 6.16.

Dalam penelitian ini, peneliti menggunakan salah satu arsitektur RNN yaitu *Gated Recurrent Unit* (GRU) untuk memprediksi harga saham yang diharapkan dapat memprediksi harga saham dengan menggunakan model *multivariate* GRU. Karena kemampuannya untuk mengelola dependensi jangka panjang dalam data *sequential* GRU berkinerja baik untuk masalah peramalan deret waktu (Wojtkiewicz dkk., 2019). GRU memiliki 2 *sigmoid* dan 1 *tanh*, sedangkan LSTM memiliki 3 *sigmoid* dan 2 *tanh*. Karena kompleksitas tersebut, maka secara teori kemampuan latih GRU lebih cepat (Hastomo dkk., 2021). Model *multivariate* mempertimbangkan beberapa faktor, model *multivariate* tidak dapat sepenuhnya menutupi kompleksitas pasar. Namun, hal ini menawarkan realitas yang lebih rinci

daripada model *univariate*. Dengan demikian model *multivariate* cenderung memberikan prediksi yang lebih akurat dan berkinerja lebih baik daripada model *univariate* (Müller, 2020).

Pokok permasalahan penelitian ini adalah melakukan prediksi harga saham *Apple Inc* pada beberapa bursa efek menggunakan model *multivariate* GRU dengan menggunakan pendekatan faktor analisa teknikal yang berdasarkan data historis harga saham. Sebagai masukannya, menggunakan harga *low*, *open*, *close*, *high* dan *volume* selama kurang lebih 11 tahun. Model diimplementasikan menggunakan bahasa pemrograman *Python*. Selanjutnya model dites menggunakan data untuk evaluasi, *error* model dihitung menggunakan metode MAE, RMSE, MAPE dan RMSPE serta menghasilkan prediksi harga *low*, *open*, *close* dan *high* untuk satu hari bursa selanjutnya.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana mengimplementasikan model *multivariate* GRU untuk memprediksi pergerakan harga saham *Apple Inc* pada beberapa bursa efek?
2. Bagaimana kinerja model *multivariate* GRU dalam memprediksi harga saham *Apple Inc* pada beberapa bursa efek?
3. Bagaimana melakukan *deployment* aplikasi *website* terhadap hasil model *multivariate* GRU yang telah dibuat dalam melakukan prediksi harga saham *Apple Inc* pada beberapa bursa efek?

## 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Penelitian ini menggunakan data primer diambil dari *website Yahoo Finance* dengan kode AAPL(*Nasdaq*), APC.F(*Frankfurt*) dan AAPL.MX(*Mexico*).
2. Menggunakan nilai tukar *realtime* dari *website Yahoo Finance* dengan kode EURUSD=X dan MXNUSD=X.
3. Data yang digunakan berupa data numerik dan nilai harga saham dalam USD dengan menggunakan 5 variabel, yaitu harga *low*, *open*, *close*, *high*, dan *volume*.

4. Algoritma yang digunakan adalah *multivariate* GRU.
5. Menggunakan bahasa pemrograman *Python*.
6. *Output* yang dihasilkan program yaitu nilai MAE, RMSE, MAPE, RMSPE dan prediksi harga *low*, *open*, *close* dan *high* satu hari bursa selanjutnya.

#### **1.4 Tujuan Penelitian**

Berdasarkan rumusan masalah, berikut tujuan dari penelitian:

1. Untuk mengimplementasikan model *multivariate* GRU dalam memprediksi harga saham *Apple Inc* pada beberapa bursa efek.
2. Untuk mengetahui kinerja model *multivariate* GRU dalam memprediksi harga saham *Apple Inc* pada beberapa bursa efek.
3. Untuk melakukan *deployment* aplikasi *website* terhadap hasil model *multivariate* GRU yang telah dibuat dalam melakukan prediksi harga saham *Apple Inc* pada beberapa bursa efek.

#### **1.5 Manfaat Penelitian**

Penelitian ini menghasilkan model yang dapat digunakan untuk memprediksi harga saham *Apple Inc* pada beberapa bursa efek, memberikan informasi tentang implementasi metode *multivariate* GRU dengan *multioutput* untuk memprediksi harga suatu saham, serta implementasi dari metode prediksi diharapkan dapat dijadikan sebagai pembanding dengan penelitian lain dan selanjutnya dalam hal memprediksi harga suatu saham.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Prediksi *Stock* / Saham**

Prediksi adalah suatu proses memperkirakan tentang sesuatu yang mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki. Prediksi dilakukan agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban yang sedekat mungkin dengan apa yang akan terjadi (Herdianto, 2013).

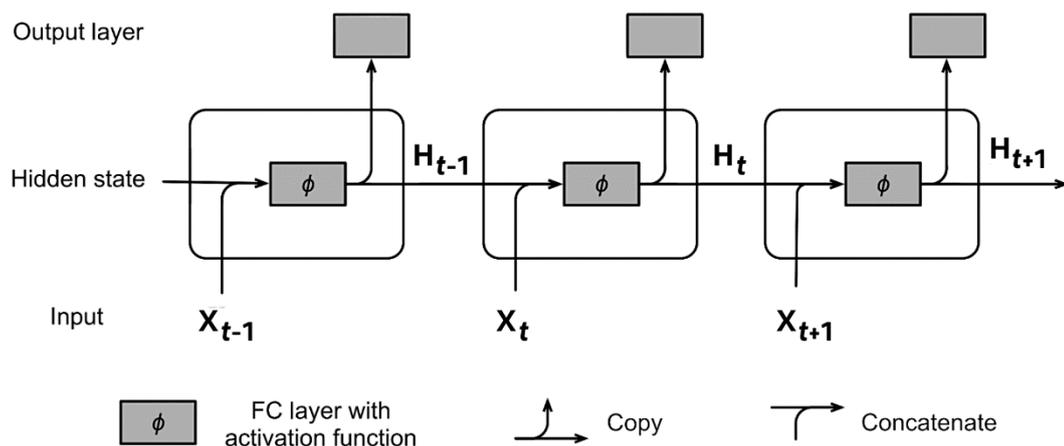
Saham merupakan salah satu instrumen pasar keuangan dan investasi yang banyak diminati oleh investor (Khalis dkk., 2021). Saham merupakan surat berharga atau instrumen pasar keuangan yang menjadi bukti kepemilikan seseorang atas suatu perusahaan maupun badan usaha yang menanamkan modal pada suatu perusahaan penerbit (Fauzi, 2019). Semakin besar saham yang dimiliki, maka semakin besar kekuasaannya di perusahaan tersebut (Ghudafa dkk., 2022). Saham dikenal mampu memberikan tingkat keuntungan yang besar tetapi risiko fluktuasi harga saham yang sangat cepat dapat mengakibatkan kerugian yang besar dalam waktu yang singkat (Arfan & ETP, 2019).

Pergerakan harga saham cenderung nonlinear dan nonstasioner yang dipengaruhi oleh banyak faktor, sehingga sangat sulit untuk meramalkan harga saham (Rochman & Djunaidy, 2014). Peramalan harga saham merupakan tantangan yang kompleks dan sulit untuk ditebak akan pergerakannya. Dalam memprediksi dapat dilakukan dengan tiga faktor yang mempengaruhi, yaitu faktor teknikal, faktor fundamental dan faktor sentimen. Faktor teknikal merupakan pergerakan dengan cara mengamati harga pada masa lalu (*high price*, *low price*, *open price*, *close price*, *volume* dan perubahan setiap harinya), faktor fundamental merupakan teknik analisis pendekatan secara bisnis yang terjadi, dan faktor sentimen merupakan pergerakan harga saham yang dipengaruhi oleh faktor-faktor bisnis, berita dan pelaku operasional bisnis (Dwiyanto dkk., 2019).

Harga *low* dan *high*, masing-masing adalah pencapaian harga terendah dan tertinggi pada hari tersebut. Harga *open* dan *close* masing-masing adalah harga pembukaan dan penutupan pada hari tersebut. *Adjusted close* adalah harga penutupan setelah penyesuaian untuk semua pembagian dan pembagian dividen yang berlaku. *Volume* adalah banyaknya pertukaran yang diperdagangkan selama beberapa periode waktu, seringkali selama satu hari.

## 2.2 Recurrent Neural Network

*Recurrent Neural Network* (RNN) merupakan salah satu model *neural network* yang mengakomodasi *output* jaringan untuk menjadi *input* jaringan kembali yang merupakan pengembangan dari *feedforward neural network* yang mampu memproses data *sequential*. RNN adalah jaringan saraf berulang karena nilai neuron pada *hidden layer* sebelumnya digunakan kembali sebagai data *input*. Penggunaan neuron pada *hidden layer* akan disimpan ke dalam *context layer* (Wardana, 2020). Data pada *context layer* ini akan digunakan sebagai *input* pada *time step* berikutnya. *Context layer* menjadi sebuah memori berisi informasi dari setiap pemrosesan *time step* sebelumnya dan selanjutnya urutan data lain dapat dihasilkan setelah urutan data terdahulu dipelajari (Siringoringo, 2021). Namun RNN sulit dilatih untuk ketergantungan jangka panjang (*long-term temporal dependency*). Arsitektur RNN dapat dilihat pada Gambar 2.1.

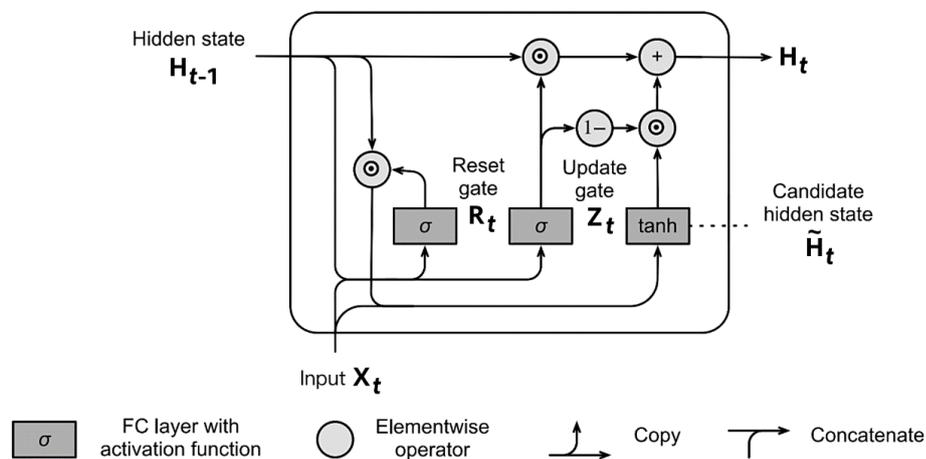


Gambar 2.1 Arsitektur *Recurrent Neural Network*

(Sumber: *Dive into Deep Learning* (h.326), oleh Zhang, A., dkk., 2022)

### 2.3 Gated Recurrent Unit

*Gated Recurrent Unit* (GRU) merupakan turunan dari arsitektur *Recurrent Neural network* (RNN). GRU adalah arsitektur yang diciptakan oleh Kyunghun Cho pada 2014 (Cho dkk., 2014). Tujuan utama dari pembuatan GRU adalah untuk membuat setiap *recurrent unit* untuk dapat menangkap *dependencies* dalam skala waktu yang berbeda-beda secara adaptif (Wardana, 2020). Serupa dengan LSTM, GRU juga menggunakan sistem gerbang, arsitektur GRU lebih sederhana daripada LSTM. GRU tidak menggunakan *cell state*, tetapi memanfaatkan *hidden state* untuk menyimpan informasi (Ghudafa dkk., 2022; Zaman dkk., 2019). Di dalam GRU, komponen pengatur alur informasi tersebut disebut sebagai *gate* dan GRU mempunyai 2 *gate*, yaitu *reset gate* dan *update gate*. Banyaknya informasi dari *time step* terdahulu yang dapat dilupakan ditentukan pada *reset gate*. Sementara *update gate* akan menentukan seberapa banyak informasi dari *time step* terdahulu yang dapat disimpan untuk digunakan sebagai *input* untuk *time step* berikutnya. Kemampuan GRU dirancang untuk menjadi lebih baik dari LSTM terutama untuk *dataset* yang jumlahnya sedikit (Hastomo dkk., 2021). Arsitektur GRU dapat dilihat pada Gambar 2.2.

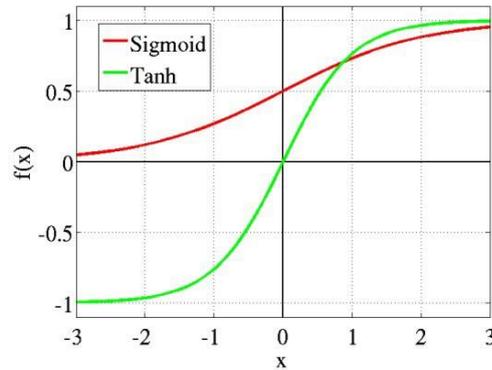


Gambar 2.2 Arsitektur *Gated Recurrent Unit*

(Sumber: *Dive into Deep Learning* (h.351), oleh Zhang, A., dkk., 2022)

GRU merupakan *cell* yang mempunyai 2 *gate* dan 3 fungsi aktivasi, yaitu dua *sigmoid* dan sebuah *tanh*. Dengan *gate* dan fungsi aktivasi yang sedikit ini tentunya akan mempercepat proses pengolahan data yang umumnya berjumlah sangat besar.

Fungsi aktivasi memutuskan apakah neuron harus diaktifkan atau tidak agar dapat berfungsi dengan benar dan memastikan bahwa *neural network* belajar untuk menggunakan informasi yang berguna. Dalam model GRU menggunakan dua jenis fungsi aktivasi yaitu *sigmoid* dan *tanh*.



Gambar 2.3 Grafik Fungsi Aktivasi *Sigmoid* dan *Tanh*

(Sumber: [medium.com/@baguspurnama](https://medium.com/@baguspurnama))

Gambar 2.3 merupakan visualisasi dari fungsi aktivasi *sigmoid* dan *tanh*. Terlihat fungsi aktivasi *sigmoid* mentransformasi *range* nilai dari *input* menjadi antara 0 dan 1, fungsi *sigmoid* diuraikan pada persamaan (2.1).

$$f(x) = \sigma(x) = \frac{1}{(1 + e^{-x})} \quad (2.1)$$

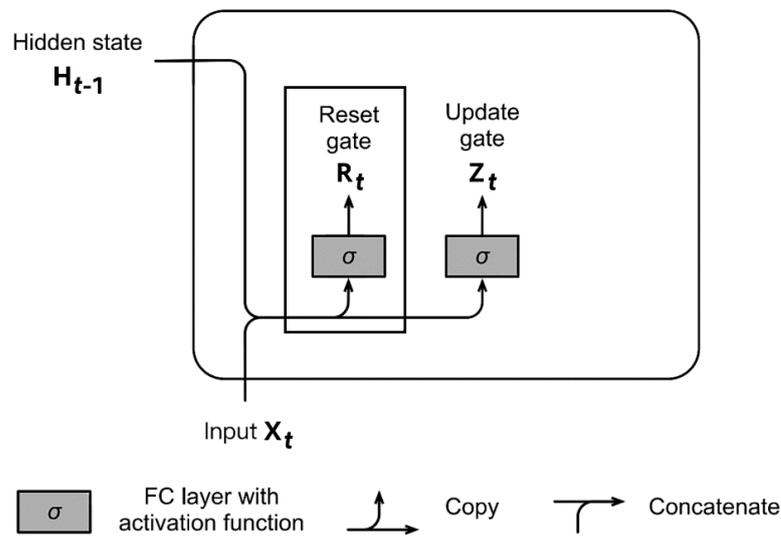
Sedangkan fungsi aktivasi *tanh* mentransformasi *range* nilai antara -1 sampai 1, fungsi *tanh* dapat diuraikan pada persamaan (2.2).

$$f(x) = \tanh(x) = \frac{2}{(1 + e^{-2x})} - 1 \quad (2.2)$$

Di mana  $x = \text{data input}$  dan  $e \approx 2.718281828459045\dots$ . Karakteristik terpusat di nol pada fungsi aktivasi *tanh* dapat lebih mudah memusatkan data sehingga proses pengoptimalan menjadi lebih mudah dibanding pada fungsi aktivasi *sigmoid* dengan *threshold* berada di 0.5 yang hanya dapat memutuskan bahwa *input* yang diberikan hanya memiliki dua tipe kelas. Karena *output* fungsi *sigmoid* yang tidak berpusat pada membuat proses optimal lebih sulit, sehingga fungsi *tanh* selalu lebih baik jika dibandingkan dengan fungsi *sigmoid*.

### 2.3.1 Reset Gate

Langkah pertama pada arsitektur GRU adalah dengan menentukan bagaimana menggabungkan informasi dari *time step* sebelumnya dan masukan baru di *reset gate*. Pada proses ini *reset gate* akan menentukan berapa banyak informasi dari *time step* terdahulu yang dapat dilupakan menggunakan fungsi aktivasi *sigmoid*. *Output* dari *reset gate* bernilai 0 dan 1. Jika *output* semakin mendekati 0 berarti informasi dari *time step* sebelumnya tidak terlalu berpengaruh dan akan dihapus sedangkan jika mendekati 1 berarti informasi dari *time step* terdahulu berpengaruh dan akan disimpan. Proses pada *reset gate* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Reset Gate

(Sumber: *Dive into Deep Learning* (h.349), oleh Zhang, A., dkk., 2022)

Perhitungan pada *reset gate* diuraikan pada persamaan (2.3).

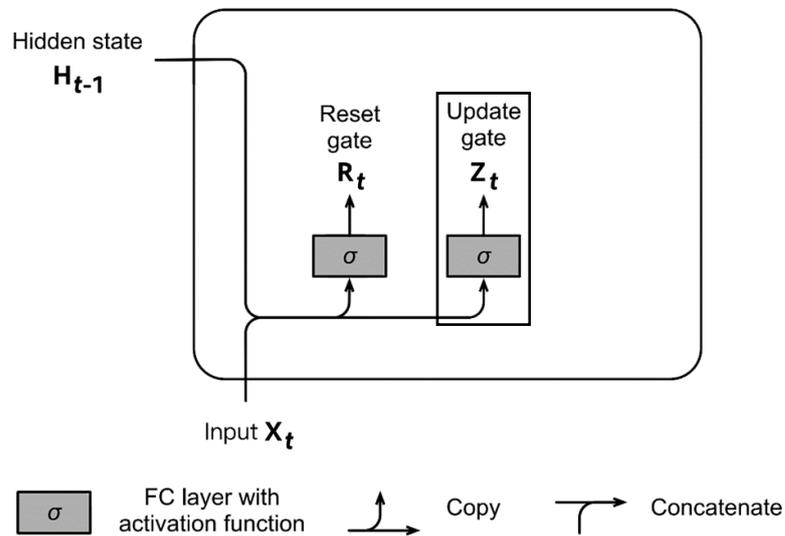
$$R_t = \sigma(X_t W_r + b_{xr} + H_{t-1} U_r + b_{hr}) \quad (2.3)$$

Keterangan:

- $R_t$  : Reset gate
- $\sigma$  : Fungsi sigmoid
- $X_t$  : Nilai data input
- $W_r, U_r$  : Nilai weight pada reset gate
- $b_{xr}, b_{hr}$  : Nilai bias pada reset gate
- $H_{t-1}$  : Hidden state dari time step sebelumnya

### 2.3.2 Update Gate

Langkah selanjutnya adalah menentukan seberapa banyak informasi dari *time step* terdahulu yang dapat disimpan untuk perhitungan *hidden state* yang digunakan sebagai *input* pada *time step* berikutnya dan menentukan pengaruh informasi dari *time step* terdahulu pada *output* di *time step* saat ini. Proses ini berlangsung di *update gate* menggunakan fungsi aktivasi *sigmoid*. Ketika *output* sama dengan 1, informasi dari *time step* sebelumnya memiliki pengaruh pada *output* saat ini, dan jika *output* sama dengan 0 maka informasi dari *time step* sebelumnya tidak berpengaruh pada *output* saat ini. Proses pada *update gate* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Update Gate

(Sumber: *Dive into Deep Learning* (h.349), oleh Zhang, A., dkk., 2022)

Perhitungan pada *update gate* diuraikan pada persamaan (2.4).

$$Z_t = \sigma(X_t W_z + b_{xz} + H_{t-1} U_z + b_{hz}) \quad (2.4)$$

Keterangan:

$Z_t$  : Update gate

$\sigma$  : Fungsi sigmoid

$X_t$  : Nilai data input

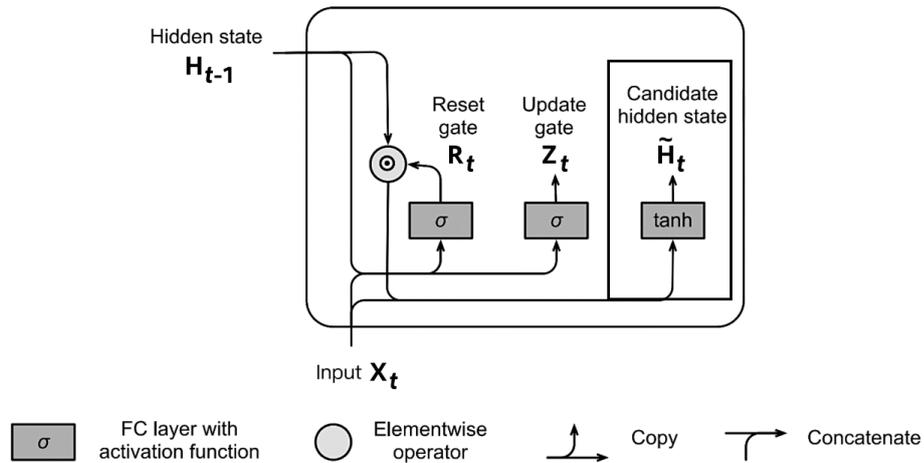
$W_z, U_z$  : Nilai *weight* pada *update gate*

$b_{xz}, b_{hz}$  : Nilai *bias* pada *update gate*

$H_{t-1}$  : *Hidden state* dari *time step* sebelumnya

### 2.3.3 Candidate Hidden State

Langkah selanjutnya adalah menentukan *candidate hidden state* atau *implicit output* pada *time step* saat ini (t) dari informasi yang relevan pada *time step* masa lalu (t-1) dengan menggunakan fungsi aktivasi *tanh*. *candidate hidden state* ini bukanlah *output final* dari unit melainkan *content memory* pada *time step* saat ini. Nilai *candidate hidden state* dipengaruhi oleh *output* dari *reset gate*. Proses penentuan nilai *candidate hidden state* dapat dilihat pada Gambar 2.6.



Gambar 2.6 *Candidate Hidden State*

(Sumber: *Dive into Deep Learning* (h.350), oleh Zhang, A., dkk., 2022)

Proses penentuan *candidate hidden state* diuraikan pada persamaan (2.5).

$$\tilde{H}_t = \tanh(X_t W_h + b_{xh} + R_t \odot (H_{t-1} U_h + b_{hh})) \quad (2.5)$$

Keterangan:

$\tilde{H}_t$  : *Candidate hidden state*

$R_t$  : *Output* pada *reset gate*

$X_t$  : Nilai data *input*

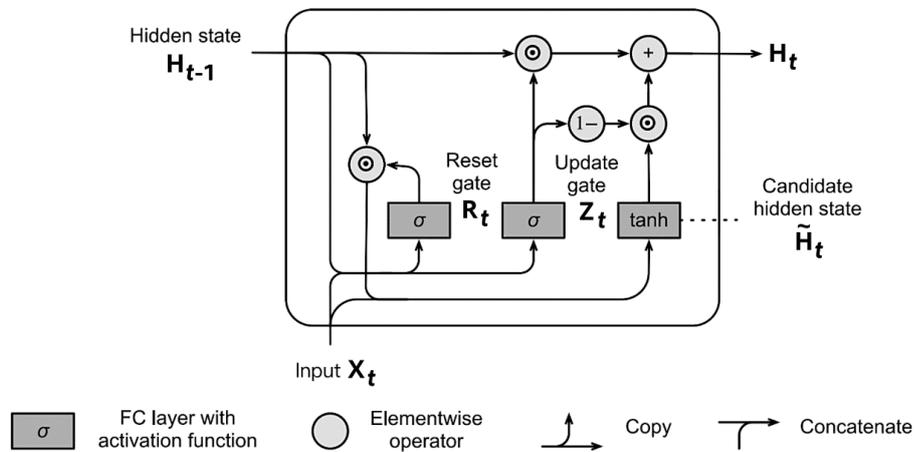
$W_h, U_h$  : Nilai *weight* pada *candidate hidden state*

$b_{xh}, b_{hh}$  : Nilai *bias* pada *candidate hidden state*

$H_{t-1}$  : *Hidden state* dari *time step* sebelumnya

### 2.3.4 Hidden State

Proses terakhir yaitu jaringan akan menghitung *output* akhir dari unit saat ini dan meneruskannya ke *time step* berikutnya sebagai *hidden state*. *Hidden state* yang akan dikirimkan ke *time step* berikutnya akan digunakan untuk menghitung kembali *output* pada unit *time step* tersebut. Prosesnya akan berjalan sama dan berulang seperti pada proses di *time step* saat ini hanya dengan nilai *input* yang berbeda. Perhitungan *output* terakhir ini dipengaruhi oleh nilai *candidate hidden state*, nilai *hidden state* pada *time step* sebelumnya dan *output* dari *update gate*. Proses penghitungan nilai *hidden state* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Output Hidden State

(Sumber: *Dive into Deep Learning* (h.351), oleh Zhang, A., dkk., 2022)

Proses penghitungan *output* terakhir diuraikan pada persamaan (2.6).

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \quad (2.6)$$

Keterangan:

$H_t$  : Output

$\tilde{H}_t$  : Candidate hidden state

$Z_t$  : Output pada update gate

$H_{t-1}$  : Hidden state dari *time step* sebelumnya

## 2.4 Metode Evaluasi Kinerja Model

Dalam metode pembelajaran *machine learning* dan *deep learning* evaluasi kinerja penting untuk dilakukan. Hal ini dilakukan karena ini bertujuan untuk mengukur tingkat akurasi kinerja suatu metode yang digunakan. Dalam penelitian

ini dalam mengukur kinerja metode yang digunakan dengan melihat *Loss Function* dan menghitung *Mean Absolute Error*, *Root Mean Square Error*, *Mean Absolute Percentage Error* dan *Root Mean Square Percentage Error*.

#### 2.4.1 Loss Function

*Loss function* digunakan untuk mengoptimalkan algoritma *machine learning*. *Loss function* dihitung berdasarkan *training data* dan *validasi data* serta interpretasinya didasarkan pada seberapa baik kinerja model dalam dua set ini. *Loss function* adalah jumlah kesalahan yang dibuat untuk setiap contoh dalam set *training* atau *validasi*. Nilai kerugian (*loss*) menyiratkan seberapa buruk atau baiknya suatu model diterapkan pada setiap iterasi optimasi. *Error* yang didapat dari *loss function* akan dipropagasi ke belakang (*backpropagation*) agar model dapat melakukan *update* parameter bobot.

Dalam *deep learning*, *loss* adalah nilai yang coba diminimalkan oleh *neural network*. Itulah bagaimana *neural network* belajar dengan menyesuaikan bobot dan *bias* dengan cara mengurangi *loss*. *Training loss* dan *validation loss* berbeda karena *training loss* diterapkan pada data *train* sedangkan, *validation loss* diterapkan pada data *validation* atau *testing*. Dengan demikian, *validation loss* adalah indikasi yang baik tentang bagaimana kinerja model pada data yang baru dikenali.

#### 2.4.2 Mean Absolute Error (MAE)

*Mean Absolute Error* (MAE) adalah nilai mutlak (absolut) kesalahan rata-rata dari nilai sebenarnya dengan nilai prediksi. Rumus MAE diuraikan pada persamaan (2.7).

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (2.7)$$

Keterangan:

$Y_i$  : Nilai aktual ke  $i$

$\hat{Y}_i$  : Nilai hasil prediksi ke  $i$

$n$  : Jumlah data

### 2.4.3 Root Mean Square Error (RMSE)

*Root Mean Square Error* (RMSE) adalah jumlah dari kesalahan kuadrat atau selisih antara nilai sebenarnya dengan nilai prediksi kemudian membagi jumlah tersebut dengan banyaknya data dan kemudian menarik akarnya. Semakin kecil nilai RMSE mengindikasikan semakin baik performa dari model dalam memprediksi. Rumus RMSE diuraikan pada persamaan (2.8).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (2.8)$$

Keterangan:

$Y_i$  : Nilai aktual ke  $i$

$\hat{Y}_i$  : Nilai hasil prediksi ke  $i$

$n$  : Jumlah data

### 2.4.4 Mean Absolute Percentage Error (MAPE)

*Mean Absolute Percentage Error* (MAPE) adalah ukuran akurasi dari suatu prediksi dengan menunjukkan nilai mutlak (absolut) dari persentase *error* data terhadap rata-rata. Rumus MAPE diuraikan pada persamaan (2.9).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \times 100\% \quad (2.9)$$

Keterangan:

$Y_i$  : Nilai aktual ke  $i$

$\hat{Y}_i$  : Nilai hasil prediksi ke  $i$

$n$  : Jumlah data

Nilai evaluasi MAPE memiliki kriteria sebagai berikut (Siringoringo, 2021):

MAPE < 10% : Kemampuan prediksi sangat baik.

10% ≤ MAPE < 20% : Kemampuan prediksi baik.

20% ≤ MAPE < 50% : Kemampuan prediksi cukup.

MAPE ≥ 50% : Kemampuan prediksi buruk.

### 2.4.5 Root Mean Square Percentage Error (RMSPE)

*Root Mean Square Percentage Error* (RMSPE) adalah ukuran akurasi dari suatu prediksi dengan menunjukkan nilai kesalahan kuadrat dari persentase *error*

terhadap rata-rata dan kemudian menarik akarnya. Rumus RMSPE diuraikan pada persamaan (2.10).

$$RMSPE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{Y_i - \hat{Y}_i}{Y_i} \right)^2} \times 100\% \quad (2.10)$$

Keterangan:

- $Y_i$  : Nilai aktual ke  $i$
- $\hat{Y}_i$  : Nilai hasil prediksi ke  $i$
- $n$  : Jumlah data

Nilai evaluasi RMSPE memiliki kriteria sebagai berikut:

- RMSPE < 10% : Kemampuan prediksi sangat baik.
- 10% ≤ RMSPE < 20% : Kemampuan prediksi baik.
- 20% ≤ RMSPE < 50% : Kemampuan prediksi cukup.
- RMSPE ≥ 50% : Kemampuan prediksi buruk.

## 2.5 Inisialisasi *Hyperparameter*

*Hyperparameter* adalah variabel yang menentukan bagaimana sebuah model dilatih. *Hyperparameter* salah satu faktor yang mempengaruhi kinerja dari model yang dibuat untuk mendapatkan performa yang baik. *Hyperparameter* tidak bisa didapatkan secara langsung pada proses *training*, inisialisasi *hyperparameter* dalam penelitian ini meliputi *optimizer* dan *learning rate*, *batch size*, jumlah *epoch*, jumlah *hidden layer* dan jumlah *neuron* pada setiap *layer*.

### 2.5.1 *Optimizer* dan *Learning rate*

*Optimizer* digunakan untuk menyesuaikan parameter jaringan syaraf dalam meminimalkan *cost function*, dalam penelitian ini menggunakan *Adaptive Moment Estimation* (Adam). Adam adalah algoritma pengoptimalan yang dapat digunakan untuk memperbarui *weight network* secara iteratif berdasarkan data *training*, optimasi ini efisien secara komputasi, kebutuhan memori yang kecil dan sesuai untuk masalah yang memiliki banyak data maupun parameter. Adam adalah algoritma yang populer digunakan karena mencapai hasil yang baik dan cepat dibanding dengan *optimizer* lainnya (Wardana, 2020). *Learning rate* untuk mengontrol kecepatan pelatihan dan mempengaruhi ketelitian jaringan untuk

menghitung nilai koreksi pada proses pembelajaran. Semakin besar *learning rate* mengakibatkan pembelajaran set bobot terlalu cepat atau proses pelatihan yang tidak stabil, maka ketelitian akan semakin berkurang, jika semakin kecil maka ketelitiannya semakin bertambah tetapi dengan proses *training* yang semakin lama (Wardana, 2020).

### **2.5.2 Batch size**

*Batch size* adalah jumlah sampel dari *dataset* yang dimasukkan ke dalam model setiap *epoch*. Untuk mempermudah dan mempercepat proses *training dataset* dibagi per *batch*. *Batch size* yang besar memerlukan memori yang lebih besar dan waktu pelatihan menjadi lebih cepat. Dalam memilih ukuran *batch* yang optimal, *batch* dimulai dari ukuran *batch* yang kecil terlebih dahulu biasanya dimulai dari 32 karena pada umumnya memberikan hasil yang terbaik, kemudian jika memungkinkan untuk meningkatkan nilai ukuran *batch* sampai diperoleh hasil yang memuaskan. Ukuran *batch* yang terlalu besar biasanya menyebabkan akurasi yang lebih rendah karena telah diamati bahwa ketika menggunakan *batch* yang lebih besar ada penurunan yang signifikan dalam kualitas model, yang diukur dengan kemampuannya untuk menggeneralisasi (Thakur, 2021).

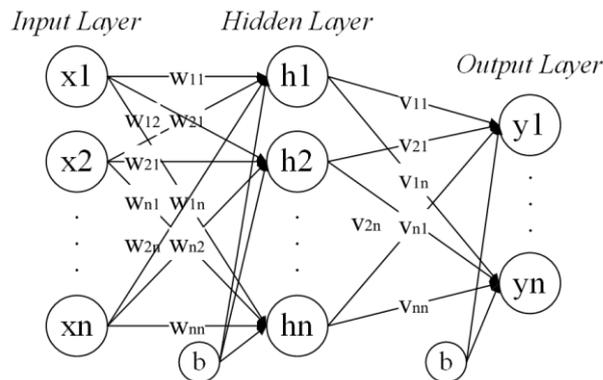
### **2.5.3 Epoch**

*Epoch* adalah jumlah iterasi *training* yang digunakan disaat semua *minibatch* masuk kedalam *optimizer* dan keluar dari *optimizer* atau di mana ketika seluruh *dataset* sudah melalui proses *training* sampai dikembalikan ke awal dalam satu kali putaran data *training*. Parameter *epoch* juga diisi dengan nilai yang eksperimental. Nilai *epoch* bisa dimulai dari angka yang kecil seperti 10 lalu mulai bertambah sampai *loss function* sudah tidak mengalami penurunan yang signifikan. Semakin banyak *epoch* maka waktu *training* juga akan bertambah.

### **2.5.4 Hidden Layer dan Neuron**

*Hidden layer* adalah lapisan antara *input layer* dan *output layer*, *layer* ini tidak pernah muncul akan tetapi semua proses pelatihan dan pengenalan dijalankan di *layer* ini, di mana neuron yang memiliki sekumpulan *input weight* dan *bias* serta prosedur untuk menghasilkan *output* neuron melalui *activation function* dan mentransfer hasilnya ke lapisan *output*. Jumlah *layer* ini tergantung dari arsitektur

yang akan dirancang, tetapi pada umumnya terdiri dari satu lapisan *hidden layer*. Neuron merupakan suatu unit yang berfungsi sebagai menyimpan dan mengirimkan informasi. Jumlah neuron pada *layer* harus sama dengan ukuran *minibatch* dari data *input* (Müller, 2020). *Hidden layer* dan neuron dapat dilihat pada arsitektur *neural network* pada Gambar 2.8.



Keterangan:

$x_n$  : neuron *input*

$h_n$  : neuron *hidden layer*

$y_n$  : neuron *output*

$w_n$  : bobot dari *input* ke *hidden layer*

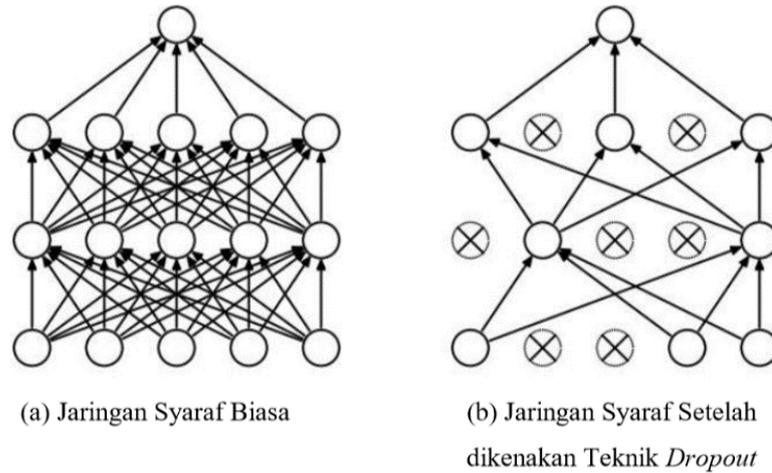
$v_n$  : bobot dari *hidden layer* ke *output*

$b$  : *bias*

Gambar 2.8 Arsitektur Neural Network

## 2.6 Dropout

*Dropout* merupakan teknik regularisasi model jaringan syaraf tiruan untuk mengurangi *overfitting* dan juga mempercepat proses *learning* pada *dataset*. *Deep neural network* umumnya memiliki lapisan yang banyak yang memungkinkan model mempelajari relasi kompleks antara *input* dan *output*. Metode ini secara acak melakukan *drop* neuron atau tidak digunakan pada jaringan selama proses *training*. Dengan melakukan *drop* pada neuron berarti, neuron yang dibuang tidak akan mengambil bagian dalam propagasi maju dan akan diberhentikan sementara dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan propagasi mundur selama proses *training*. Contoh implementasi *dropout* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Perbedaan Jaringan Syaraf Biasa dan Setelah Teknik *Dropout*

(Sumber: [medium.com/@16611110](https://medium.com/@16611110))

## 2.7 Normalisasi Data

Normalisasi data adalah proses membuat skala nilai pada variabel sehingga data berada pada rentang nilai yang sama. Langkah ini sangat penting karena dapat meminimalkan *error* dan juga mempercepat proses *train*. Tujuan dari normalisasi data adalah menghindari fitur yang memiliki nilai yang lebih besar mendominasi fitur yang memiliki nilai lebih kecil (Wardana, 2020). Metode normalisasi data yang digunakan dalam penelitian ini adalah *Min-Max Scaling Normalization*. Persamaan umum untuk scaling dalam rentang [0, 1] diuraikan pada persamaan (2.11).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.11)$$

Keterangan:

$x$  : Nilai aktual

$x'$  : Nilai hasil normalisasi

$x_{max}$  dan  $x_{min}$  : Nilai tertinggi dan terendah dari data

## 2.8 Deployment Model

*Deployment* model adalah suatu proses untuk membuat model (model *machine learning*) tersebut tersedia pada lingkungan produksi, di mana model tersebut dapat memberikan prediksi ke sistem perangkat lunak yang lain. *Deployment* model berarti menerapkan model untuk dapat diakses oleh orang lain. *Streamlit* adalah

sebuah *framework* berbasis *python* dan bersifat *open-source* yang dibuat untuk memudahkan dalam membangun aplikasi *web* di bidang *data science* dan *machine learning* yang interaktif (Singh, 2021). Dengan menggunakan *framework* ini hanya perlu *install library Streamlit* dan membuat *file Python* dalam merancang *website* sehingga tidak perlu lagi membuat *file HTML, CSS, JavaScript, Flask, Django* atau *tools* lainnya.