

**ANALISIS PERBANDINGAN *RECURRENT NEURAL
NETWORK* DAN *LONG SHORT-TERM MEMORY*
DALAM MEMPREDIKSI HARGA SAHAM**

SKRIPSI

OLEH:

**EVINDO AMANDA RIZA
218160018**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2025**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 12/12/25

Access From (repository.uma.ac.id)12/12/25

**ANALISIS PERBANDINGAN *RECURRENT NEURAL
NETWORK* DAN *LONG SHORT-TERM MEMORY*
DALAM MEMPREDIKSI HARGA SAHAM**

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana di Fakultas Teknik
Universitas Medan Area

OLEH:

EVINDO AMANDA RIZA

218160018

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2025**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 12/12/25

Access From (repository.uma.ac.id)12/12/25

HALAMAN PENGESAHAN

Judul Skripsi : Analisis Perbandingan *Recurrent Neural Network* dan *Long Short-Term Memory* Dalam Memprediksi Harga Saham

Nama : Evindo Amanda Riza
Npm : 218160018
Fakultas : Teknik



Dekan Fakultas Teknik

Ka. Prodi Teknik Informatika



Tanggal Lulus: 17 September 2025

UNIVERSITAS MEDAN AREA

©-Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 17 September 2025



Evindo Amanda Riza
218160018

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Evindo Amanda Riza
Npm : 218160018
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis Karya : Skripsi

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

Analisis Perbandingan *Recurrent Neural Network* dan *Long Short-Term Memory* Dalam Memprediksi Harga Saham

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di : Medan
Pada Tanggal : 17 September 2025
Yang Menyatakan



Evindo Amanda Riza
218160018

RIWAYAT HIDUP

Evindo Amanda Riza, lahir di Kota Berastagi, Kabupaten Karo, Provinsi Sumatera Utara, pada tanggal 25 Mei 2003. Penulis merupakan anak pertama dari empat bersaudara, putra dari ayah Johan dan ibu Suryati. Penulis menempuh pendidikan di SD Swasta Letjen Jamin Ginting Berastagi pada tahun 2009-2015, kemudian melanjutkan pendidikan di SMP Negeri 1 Berastagi pada tahun 2015-2018, dan melanjutkan pendidikan di SMA Negeri 1 Berastagi pada tahun 2018-2021. Saat ini, penulis menempuh pendidikan tinggi di Universitas Medan Area (UMA), Fakultas Teknik, Program Studi Teknik Informatika.

Pada masa perkuliahan, penulis telah melaksanakan kerja praktek (KP) di Dinas Perpustakaan dan Arsip Provinsi Sumatera Utara, tepatnya di Ruang Pembinaan Kelembagaan dan Sumber Daya Manusia Perpustakaan.

KATA PENGANTAR

Allhamdulillah, segala puji dan syukur penulis ucapkan atas kehadiran Allah Subhaanahu wa Ta'aala atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “**Analisis Perbandingan *Recurrent Neural Network* dan *Long Short-Term Memory* Dalam Memprediksi Harga Saham**”. Adapun tujuan dalam penulisan skripsi ini sebagai salah satu syarat untuk memperoleh gelar sarjana (S1) di Fakultas Teknik, Program Studi Teknik Informatika di Universitas Medan Area. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Dengan penuh rasa syukur kepada Allah SWT. atas segala petunjuk, pertolongan, dan perlindungan-Nya selama penyusunan karya ini berlangsung.
2. Yang sangat tercinta dan teristimewa kedua orang tua penulis, ayah Johan dan ibu Suryati yang senantiasa memberikan doa, nasihat, serta dukungan kepada penulis, sehingga dapat menyelesaikan pendidikan hingga perguruan tinggi.
3. Bapak Prof. Dr. Dadan Ramdan, M.Eng., M.Sc., selaku Rektor Universitas Medan Area.
4. Bapak Dr. Eng. Supriatno, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Medan Area.
5. Bapak Rizki Muliono, S.Kom., M.Kom., selaku Kepala Program Studi Teknik Informatika sekaligus sebagai dosen pembimbing yang telah memberikan bimbingan dan arahan yang membantu penulis sehingga skripsi ini dapat terselesaikan dengan baik.

6. Seluruh Dosen Teknik Informatika Universitas Medan Area yang telah mendidik dan membekali penulis ilmu yang sangat bermanfaat.
7. Bang Robby Kurniawan Sari Damanik S.T., selaku *IT Support* Teknik Informatika yang meringankan segala urusan terkait pengurusan administrasi.
8. Kepada saudara-saudara penulis, Herry, Ewin, Echika dan Elvano, atas segala doa, dorongan semangat, serta dukungan yang senantiasa diberikan selama penyusunan skripsi ini.
9. Kepada seluruh teman seperjuangan penulis yang telah memberikan semangat dan bersama-sama berjuang di Teknik Informatika Universitas Medan Area.

Penulis menyadari bahwa penulisan skripsi ini masih memiliki banyak kekurangan. Penulis berharap semoga hasil penelitian ini dapat bermanfaat, baik bagi penulis maupun bagi siapa saja yang membacanya. Aamiin.

Medan, 17 September 2025



Evindo Amanda Riza
218160018

ABSTRAK

Saham merupakan salah satu instrumen investasi yang bersifat fluktuatif, sehingga diperlukan metode prediksi yang dapat mendukung pengambilan keputusan dalam aktivitas jual dan beli saham. Seiring dengan perkembangan teknologi kecerdasan buatan (AI), khususnya *deep learning* kini banyak dimanfaatkan untuk membantu investor dalam memahami pergerakan pasar. Penelitian ini bertujuan untuk membandingkan performa model *deep learning*, seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) dalam memprediksi harga saham PT Bank Rakyat Indonesia Tbk (BBRI). Kedua model dilatih menggunakan kombinasi hyperparameter terbaik yang diperoleh melalui metode *RandomizedSearchCV*. Hasil penelitian ini menunjukkan bahwa model RNN menghasilkan nilai MSE sebesar 9.340,06, RMSE sebesar 96,64, dan MAE sebesar 76,57. Sementara itu, model LSTM menghasilkan nilai MSE sebesar 5.535,33, RMSE sebesar 74,39, dan MAE sebesar 57,25. Selain itu, hasil prediksi harga penutupan saham BBRI selama 7 hari ke depan menunjukkan bahwa RNN memiliki rata-rata persentase *error* sebesar 1,71%, sedangkan LSTM hanya sebesar 1,31%. Berdasarkan hasil tersebut, model LSTM terbukti memiliki performa yang lebih baik dibandingkan RNN dalam memprediksi harga penutupan saham BBRI, yang ditunjukkan oleh nilai *error* yang lebih rendah pada seluruh metrik evaluasi. Model prediksi diimplementasikan ke dalam sebuah aplikasi web menggunakan *streamlit*.

Kata Kunci: Saham, Prediksi Harga Saham, BBRI, RNN, LSTM, *Streamlit*

ABSTRACT

Stocks are one of the investment instruments that are inherently volatile, thus requiring predictive methods to support decision-making in stock buying and selling activities. With the advancement of artificial intelligence (AI) technology, particularly deep learning, these methods are increasingly being utilized to help investors understand market movements. This study aims to compare the performance of deep learning models such as Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) in predicting the stock price of PT Bank Rakyat Indonesia Tbk (BBRI). Both models were trained using the best combination of hyperparameters obtained through the RandomizedSearchCV method. The results of this study show that the RNN model achieved a Mean MSE of 9.340,06, RMSE of 96,64, and MAE of 76,57. Meanwhile, the LSTM model achieved an MSE of 5.535,33, RMSE of 74,39, and MAE of 57,25. Additionally, the 7-day ahead stock closing price predictions indicate that the RNN model has an average percentage error of 1,71%, while the LSTM model has only 1,31%. Based on these results, the LSTM model proves to have better performance compared to the RNN model in predicting the closing price of BBRI stock, as indicated by the lower error values across all evaluation metrics. The predictive models were implemented into a web application using Streamlit.

Keywords: Stock, Stock Price Prediction, BBRI, RNN, LSTM, Streamlit

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	Error! Bookmark not defined.
HALAMAN PERNYATAAN	Error! Bookmark not defined.
RIWAYAT HIDUP	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
 BAB I PENDAHULUAN	 1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
 BAB II TINJAUAN PUSTAKA	 5
2.1. Saham	5
2.2. Prediksi Harga Saham	6
2.3. <i>Recurrent Neural Network</i> (RNN)	7
2.4. <i>Long Short-Term Memory</i> (LSTM).....	9
2.5. Normalisasi.....	15
2.6. Denormalisasi	15
2.7. Parameter Evaluasi	16
2.7.1. <i>Mean Squared Error</i> (MSE).....	16
2.7.2. <i>Root Mean Squared Error</i> (RMSE).....	16
2.7.3. <i>Mean Absolute Error</i> (MAE).....	17
2.8. <i>Flowchart Diagram</i>	17
2.9. <i>Unified Modeling Language</i> (UML)	18
2.9.1. <i>Use Case Diagram</i>	18
2.9.2. <i>Activity Diagram</i>	19
2.10. Penelitian Terdahulu.....	20
 BAB III METODOLOGI PENELITIAN	 23
3.1. Perangkat Penelitian	23
3.2. Tahapan Penelitian	23
3.2.1. Pengumpulan Data.....	24

3.2.2. <i>Preprocessing Data</i>	27
3.2.3. Pembangunan Model RNN dan LSTM	29
3.2.4. <i>Tuning Hyperparameter</i> RandomizedSearchCV	31
3.2.5. Pelatihan dan Pengujian Model	31
3.2.6. Evaluasi Model	37
3.3. Tahapan Perancangan.....	40
3.3.1. <i>Unified Modeling Language (UML)</i>	40
3.3.2. <i>User Interface (UI)</i>	43
BAB IV HASIL DAN PEMBAHASAN	44
4.1. Hasil.....	44
4.1.1. Deskripsi Dataset	44
4.1.2. Aplikasi Web Prediksi Harga Saham.....	45
4.1.3. Hasil Evaluasi Model.....	49
4.2. Pembahasan	50
4.2.1. RandomizedSearchCV	51
4.2.2. Pelatihan dan Pengujian Model	52
4.2.3. Prediksi Model RNN dan LSTM.....	54
4.2.4. Perbandingan dengan Hasil Penelitian Terdahulu	56
BAB V KESIMPULAN DAN SARAN	57
5.1. Kesimpulan.....	57
5.2. Saran	58
DAFTAR PUSTAKA.....	59
LAMPIRAN.....	62

DAFTAR TABEL

	Halaman
Tabel 2.1. Komponen <i>Flowchart Diagram</i>	17
Tabel 2.2. Komponen <i>Use Case Diagram</i>	19
Tabel 2.3. Komponen <i>Activity Diagram</i>	19
Tabel 2.4. Penelitian Terdahulu.....	20
Tabel 3.1. Dataset Saham BBRI.....	25
Tabel 3.2. Perhitungan Normalisasi Harga Penutupan BBRI	28
Tabel 3.3. Proses Pembentukan <i>Sliding Window</i> Harga Penutupan BBRI	29
Tabel 3.4. Hyperparameter yang Digunakan.....	30
Tabel 3.5. Sampel Pertama Pelatihan Model	31
Tabel 3.6. Hasil Perhitungan RNN.....	33
Tabel 3.7. Hasil Perhitungan LSTM	36
Tabel 3.8. Nilai Output Prediksi RNN dan LSTM.....	37
Tabel 3.9. Hasil Perhitungan Denormalisasi RNN dan LSTM	38
Tabel 3.10. Hasil Perhitungan Sampel Pertama RNN dan LSTM	38
Tabel 4.1. Hasil Prediksi Model RNN dan LSTM Pada Data <i>Testing</i>	49
Tabel 4.2. Hasil Evaluasi Model RNN dan LSTM	50
Tabel 4.3. Peringkat Pencarian RandomizedSearchCV Pada RNN.....	51
Tabel 4.4. Peringkat Pencarian RandomizedSearchCV Pada LSTM.....	52
Tabel 4.5. Hasil Prediksi Saham BBRI 7 Hari Kedepan Pada RNN.....	55
Tabel 4.6. Hasil Prediksi Saham BBRI 7 Hari Kedepan Pada LSTM	55

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Arsitektur RNN.....	7
Gambar 2.2. Arsitektur LSTM	9
Gambar 2.3. Alur <i>Cell State</i> pada LSTM	10
Gambar 2.4. Alur <i>Forget Gate</i> pada LSTM	11
Gambar 2.5. Alur <i>Input Gate</i> pada LSTM.....	12
Gambar 2.6. Alur <i>Cell State</i> pada LSTM	13
Gambar 2.7. Alur <i>Output Gate</i> pada LSTM.....	14
Gambar 3.1. <i>Flowchart</i> Tahapan Penelitian	24
Gambar 3.2. Komponen <i>Candlestick</i>	26
Gambar 3.3. <i>Use Case Diagram</i> Aplikasi Web	40
Gambar 3.4. <i>Activity Diagram</i> Pelatihan Model Terbaik Oleh Admin	41
Gambar 3.5. <i>Activity Diagram</i> Menampilkan Hasil Prediksi Oleh User.....	42
Gambar 3.6. Desain UI Halaman Aplikasi Web	43
Gambar 4.1. Grafik <i>Candlestick</i> Saham BBRI (2010-2025).....	44
Gambar 4.2. Grafik <i>Candlestick</i> Saham BBRI (30 Hari Terakhir)	45
Gambar 4.3. Halaman Awal Aplikasi Web	46
Gambar 4.4. Halaman Tab <i>Dataframes</i>	46
Gambar 4.5. Halaman Tab <i>Charts</i>	47
Gambar 4.6. Halaman Tab <i>Statistics</i>	47
Gambar 4.7. Halaman Tab <i>Prediction</i>	48
Gambar 4.8. Halaman Tab <i>Comparison</i>	49
Gambar 4.9. Visualisasi Prediksi RNN dan LSTM Pada Data <i>Testing</i>	50
Gambar 4.10. Grafik <i>Training vs Validation Loss</i> Pada RNN	53
Gambar 4.11. Grafik <i>Training vs Validation Loss</i> Pada LSTM.....	53
Gambar 4.12. Grafik Prediksi Harga Saham BBRI 7 Hari Kedepan	54

BAB I

PENDAHULUAN

1.1. Latar Belakang

Saham merupakan surat berharga yang mencerminkan kepemilikan seseorang atau entitas dalam sebuah perusahaan, yang artinya pemegang saham memiliki porsi kepemilikan perusahaan tersebut (Rosyd, dkk., 2024). Pergerakan harga saham dipengaruhi oleh berbagai faktor, baik internal seperti kinerja perusahaan, manajemen perusahaan, maupun eksternal seperti kebijakan pemerintah, inflasi, suku bunga, ekonomi makro, tren industri, dan sentimen pasar. Prediksi harga saham menjadi sulit karena volatilitas pasar yang tinggi dan pola data yang kompleks (Janastu & Wutsqa, 2024).

Berdasarkan data Kustodian Sentral Efek Indonesia (KSEI), jumlah investor pasar modal Indonesia menunjukkan pertumbuhan signifikan sejak pandemi COVID-19. Fenomena ini mencerminkan minat masyarakat yang tinggi terhadap investasi saham, namun juga menuntut adanya kemampuan analisis yang cermat dalam mengambil keputusan investasi, mengingat harga saham berfluktuasi sepanjang waktu dan tidak ada saham yang selalu naik ataupun turun. Oleh karena itu, strategi yang tepat sangat diperlukan guna mengoptimalkan potensi keuntungan dan meminimalkan risiko kerugian, baik bagi seorang investor, trader, maupun analis keuangan dalam menentukan waktu yang tepat untuk membeli, menahan, atau menjual saham. Salah satu pendekatan yang dapat digunakan adalah teknik peramalan atau prediksi harga saham (Faridah & Sugiantoro, 2023).

Seiring dengan perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*), metode ini semakin banyak digunakan sebagai pilihan dalam meningkatkan kemampuan prediksi harga saham dibandingkan metode konvensional. Saat ini, penelitian mengenai prediksi saham cenderung menggunakan pendekatan *deep learning* (Pipin dkk., 2023). Salah satu algoritma *deep learning* yang digunakan adalah *Recurrent Neural Network* (RNN), yang memiliki kemampuan cukup baik dalam memprediksi data deret waktu (*time series*). Namun, RNN memiliki keterbatasan dalam menyimpan informasi jangka panjang. Oleh karena itu, dikembangkanlah varian arsitektur baru, yaitu *Long Short-Term Memory* (LSTM), yang masih termasuk dalam keluarga RNN. LSTM dirancang dengan menambahkan *memory cell* sehingga mampu menyimpan informasi dalam rentang waktu lebih lama serta mengatasi permasalahan *vanishing gradient* yang kerap muncul pada RNN (Hanafiah dkk., 2023).

Beberapa penelitian tentang prediksi harga saham telah dilakukan, salah satunya oleh Zhu Yongqiong (2020), yang memprediksi pasar saham perusahaan Apple menggunakan RNN selama sepuluh tahun terakhir disimpulkan bahwa model ini memiliki keunggulan yang mampu mengolah data *time series* dengan akurasi prediksi hingga 95% dan *loss* mendekati 0,1%. Dari hasil penelitian, prediksi lebih akurat pada *timesteps*=5 dibandingkan *timesteps*=10. Ketika *timesteps* meningkat terjadi peningkatan kesalahan prediksi yang terlihat dari MSE, MAE, RMSE yang lebih tinggi, nilai MSE 68,49 (*timesteps*=5), 109,22 (*timesteps*=10), MAE 5,90 (*timesteps*=5), 7,84 (*timesteps*=10) dan RMSE juga meningkat 8,28 (*timesteps*=5), 10,45 (*timesteps*=10), menunjukkan kesalahan prediksi yang lebih besar pada interval waktu yang lebih panjang.

Penelitian lainnya oleh Wathani, M. N., Kusrini, K., & Kusnawi, K. (2023) memprediksi tren harga saham PT Bank Central Asia Tbk (BBCA) menggunakan algoritma LSTM dengan data dari tahun 2011 hingga 2022. Model diuji menggunakan beberapa variasi *epoch*, dengan *batch size* 1 dan optimasi Adam untuk mendapatkan hasil prediksi terbaik. Dengan nilai MAE sebesar 96,92 menunjukkan akurasi yang cukup baik, MSE sebesar 16.185,22 menunjukkan model memiliki tingkat kesalahan kuadrat rata-rata yang dapat diterima, RMSE sebesar 127,22 mengindikasikan error prediksi yang masih berada dalam batas akurasi yang memadai.

Meskipun banyak penelitian sebelumnya yang membahas prediksi harga saham menggunakan RNN dan LSTM yang masing-masing menghasilkan akurasi yang baik dan tingkat *error* yang rendah, tetapi belum banyak penelitian yang secara langsung membandingkan performa antara keduanya. Sebagian besar penelitian hanya menggunakan satu jenis model tanpa melakukan analisis komparatif antara RNN dan LSTM dalam konteks prediksi harga saham. Oleh karena itu, penelitian ini dilakukan untuk membandingkan performa kedua model berdasarkan metrik evaluasi seperti RMSE, MSE dan MAE.

Berdasarkan latar belakang yang telah diuraikan, maka penelitian ini mengambil judul yaitu “Analisis Perbandingan *Recurrent Neural Network* dan *Long Short-Term Memory* Dalam Memprediksi Harga Saham.”

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, masalah penelitian yang dapat dirumuskan adalah bagaimana performa antara RNN dan LSTM dalam memprediksi harga saham berdasarkan metrik evaluasinya.

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini ditetapkan agar penelitian ini berfokus pada ruang lingkup yang telah ditentukan, diantaranya sebagai berikut:

1. Model yang dilakukan perbandingan adalah RNN dan LSTM.
2. Dataset diperoleh dari website Yahoo Finance dengan ticker saham BBRI.JK.
3. Dataset dibatasi pada rentang waktu 1 Januari 2010 hingga 1 Januari 2025.
4. Model dievaluasi menggunakan metrik seperti MSE, RMSE, dan MAE.
5. Aplikasi dan program dibangun menggunakan bahasa pemrograman Python.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah menganalisis dan membandingkan performa RNN dan LSTM dalam memprediksi harga saham berdasarkan metrik evaluasi yang digunakan.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi penelitian dalam penerapan kecerdasan buatan, khususnya *deep learning* seperti RNN dan LSTM dalam prediksi harga saham, serta menjadi referensi bagi penelitian lanjutan.
2. Mendukung penerapan *deep learning* untuk meningkatkan strategi investasi dan pengambilan keputusan di pasar modal.

BAB II

TINJAUAN PUSTAKA

2.1. Saham

Pasar modal memiliki pengaruh yang signifikan terhadap perekonomian suatu negara. Pasar modal merupakan instrumen keuangan jangka panjang yang memfasilitasi transaksi jual beli saham, mendukung pendanaan bisnis, serta aktivitas investasi lain. Investasi saham adalah salah satu cara yang dapat dilakukan di pasar modal. Bursa Efek Indonesia (BEI) merupakan salah satu pasar modal atau bursa saham yang terdapat di Indonesia (Riyantoko dkk., 2020). Produk yang diperdagangkan di pasar modal umumnya terbagi menjadi dua jenis, yaitu surat berharga berbentuk kepemilikan dan surat berharga yang berbentuk hutang. Beberapa produk yang tersedia di pasar modal meliputi reksadana, obligasi, dan saham.

Saham yang diperdagangkan harus terdaftar di Bursa Efek Indonesia (BEI), yang mengatur dan mengelola aktivitas jual-beli saham. Dalam periode tertentu, investor yang memiliki saham di suatu perusahaan berkesempatan untuk mengambil bagian dari keuntungan yang dihasilkan oleh perusahaan yang disebut *dividen*. Selain itu, pemegang saham juga bisa memperoleh keuntungan lain yang disebut *capital gain*, yang diperoleh saat menjual saham dengan harga lebih tinggi di pasar modal (Kustina dkk., 2019)

Menurut Iman & Wulandari, 2023, saham adalah dokumen berharga yang mewakili sebagian kepemilikan dalam suatu perusahaan. Artinya, ketika seseorang membeli saham, mereka memperoleh sebagian kepemilikan perusahaan tersebut. Dengan memiliki saham, mereka juga berhak atas aset dan pendapatan yang dihasilkan perusahaan, sesuai dengan proporsi kepemilikan yang dimiliki.

Salah satu saham terbesar di Indonesia adalah milik Bank Rakyat Indonesia (BRI). BRI didirikan oleh Raden Bei Aria Wirjaatmadja pada 16 Desember 1895 di Purwokerto, Jawa Tengah. Pada 10 November 2003, BRI menjadi perusahaan terbuka dan mencatatkan sahamnya di Bursa Efek Jakarta, yang sekarang dikenal sebagai Bursa Efek Indonesia, dengan kode BBRI. Sebagian besar saham BRI, yaitu 56,75%, dimiliki oleh Negara Republik Indonesia, sementara sisanya dimiliki oleh pemegang saham publik.

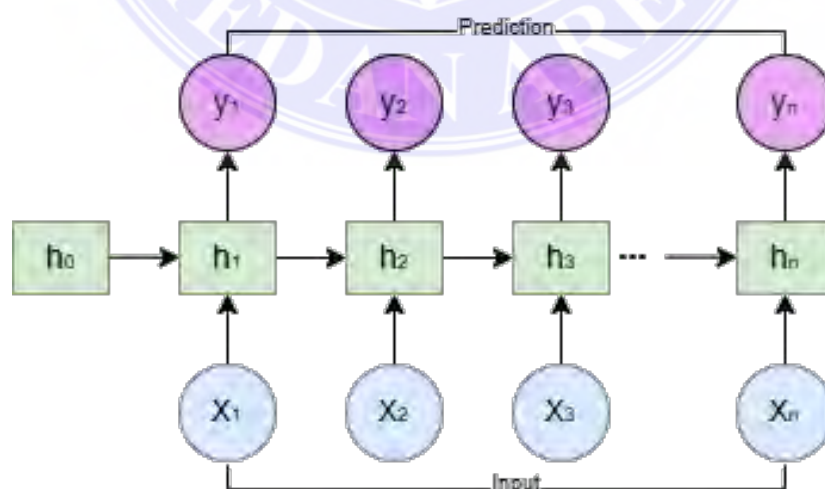
2.2. Prediksi Harga Saham

Manusia, dengan kemampuan berpikir mereka, berusaha memperkirakan kejadian di masa depan melalui berbagai teknik dan analisis untuk mengurangi kesalahan sekecil mungkin. Namun, tidak ada yang bisa mengetahui secara pasti apa yang akan terjadi (Karno, 2020). Oleh karena itu, manusia melakukan prediksi. Prediksi adalah proses memperkirakan secara sistematis kejadian yang paling mungkin terjadi di masa depan berdasarkan informasi dari masa lalu dan sekarang, agar kesalahan (selisih antara kejadian nyata dan perkiraan) dapat diminimalkan. Prediksi tidak selalu memberikan jawaban pasti, tetapi berupaya untuk mendekati hasil yang mungkin terjadi (Wathani dkk., 2023).

Masalah utama investor adalah bagaimana memprediksi pergerakan harga saham di masa depan agar bisa membuat keputusan investasi yang tepat. Menentukan arah pergerakan harga saham tidaklah mudah, karena beberapa faktor yang mempengaruhi prediksi ini yakni faktor teknikal, fundamental, dan sentiment (Suyudi dkk., 2019). Faktor teknikal didasarkan pada analisis pergerakan harga sebelumnya, faktor fundamental melibatkan analisis terhadap aspek bisnis perusahaan dan faktor sentimen mencakup pengaruh dari berita, kondisi bisnis, dan aktivitas pelaku bisnis terhadap pergerakan harga saham (Rosyd dkk., 2024).

2.3. Recurrent Neural Network (RNN)

RNN merupakan salah satu jenis *Artificial Neural Network* (ANN) yang dirancang khusus untuk memproses data bersifat sekuensial (berurutan), sehingga sangat sesuai dalam pengolahan data *time series*. RNN memiliki arsitektur yang terdiri dari *input layer*, *hidden layer*, dan *output layer*, di mana aliran informasi bersifat rekursif yang memungkinkan jaringan mempertahankan hubungan dengan *hidden state* dari langkah waktu sebelumnya (Wardani, 2021).



Gambar 2.1. Arsitektur RNN

(Sumber: Mienye dkk., 2024)

Dalam proses pelatihan, RNN menggunakan algoritma *Backpropagation Through Time* (BPTT) yang menjumlahkan kesalahan di setiap *timestep* untuk menyesuaikan parameter model. Namun, RNN menghadapi dua masalah utama yaitu jika gradien terlalu kecil, pembaruan bobot menjadi sulit atau bahkan berhenti (*vanishing gradient*). Sebaliknya, jika gradien terlalu besar, bobot model bisa membesar secara tidak terkendali hingga menghasilkan NaN (Not a Number), menyebabkan model tidak stabil (*exploding gradient*). Salah satu solusi untuk mengatasi masalah ini adalah dengan mengurangi jumlah *hidden layer* dan menyederhanakan struktur jaringan agar lebih stabil dalam proses pembelajaran.

RNN biasanya digunakan untuk menangani masalah yang bersifat berurutan atau temporal, seperti penerjemahan bahasa, pemrosesan bahasa alami atau *Natural Language Processing* (NLP), pengenalan suara, dan pembuatan deskripsi gambar. Teknologi ini juga diterapkan dalam berbagai aplikasi populer, seperti Cortana, Google Assistant, Siri, dan Google Translate (Jaya, 2022). Adapun rumus RNN ditunjukkan pada persamaan berikut.

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (2.1)$$

$$y_t = \tanh(W_{hy}h_t + b_y) \quad (2.2)$$

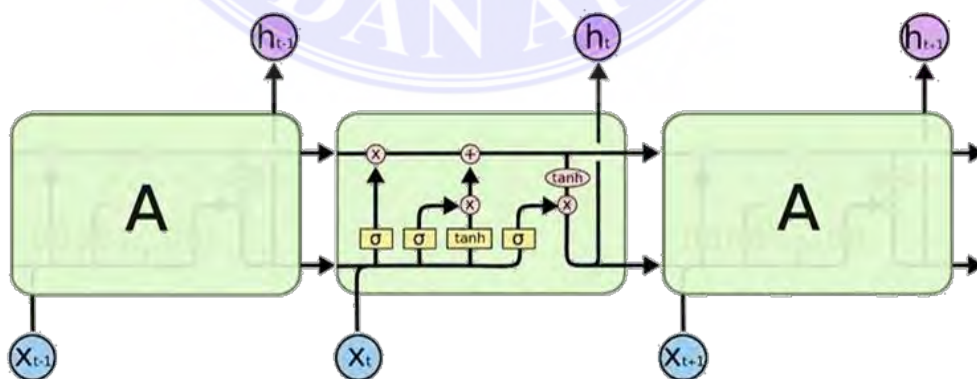
Dengan:

- h_t : *hidden state* pada waktu ke-t
- W_{xh}, W_{hh}, W_{hy} : matriks bobot di setiap layer
- b_h, b_y : vector bias
- x_t dan y_t : input dan output pada waktu ke -t
- $\tanh$: fungsi aktivasi layer

2.4. Long Short-Term Memory (LSTM)

LSTM merupakan varian serta pengembangan arsitektur dari RNN yang diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 untuk mengatasi beberapa kelemahan seperti masalah ketergantungan jangka panjang (*long term dependency*) dan *vanishing gradient* atau gradien yang menghilang pada RNN (Ramadhan, 2024).

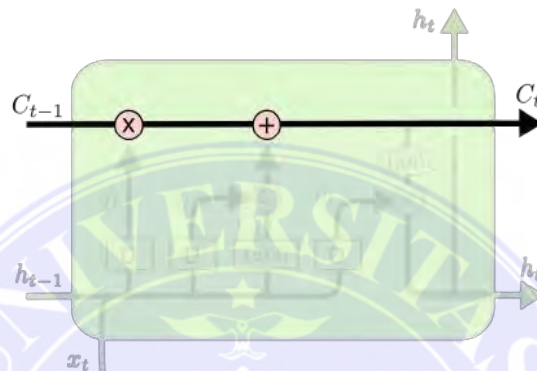
Tidak seperti RNN, LSTM tidak memiliki kelemahan ini karena mampu mengatur memori pada setiap input dengan menggunakan *memory cells* dan *gate units* (Arfan & Lussiana ETP, 2019). Komponen *gate units* pada LSTM terdiri dari *forget gate*, *input gate*, dan *output gate* (Gumelar dkk., 2022). Ketiga *gates* dan *memory cell* ini dirancang untuk dapat membaca, menyimpan, dan memperbarui informasi terdahulu (Faridah & Sugiantoro, 2023). Karena struktur unik sistem ini, LSTM dapat mengingat data dengan durasi waktu acak dan tidak akan mengalami masalah yang hilang. Oleh karena itu, LSTM merupakan varian RNN yang efektif untuk aplikasi data sekuensial dan banyak digunakan untuk pemrosesan teks, video, dan data *time series* lainnya (Kusuma, 2024).



Gambar 2.2. Arsitektur LSTM

(Sumber: Olah, Christopher)

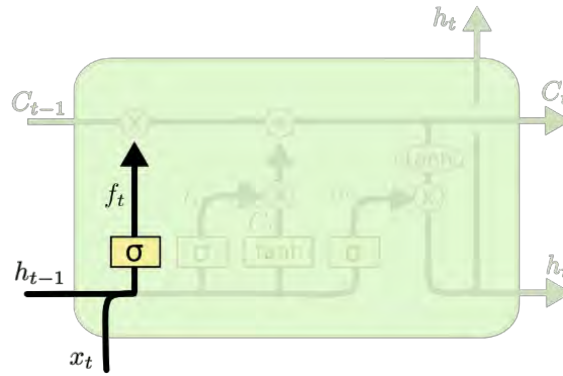
Ilustrasi dari arsitektur LSTM dapat dilihat pada Gambar 2.2 diatas. Tahapan untuk memproses data akan dijabarkan satu persatu pada bagian ini. Kunci utama LSTM adalah *cell state* yaitu garis horizontal di bagian atas diagram yang menghubungkan seluruh *output layer* (Wardani, 2021). Alur *cell state* ditunjukkan pada Gambar 2.3 berikut:



Gambar 2.3. Alur *Cell State* pada LSTM

(Sumber: Olah, Christopher)

Langkah pertama, menentukan informasi mana yang akan disimpan atau dibuang dari *cell state* melalui *forget gate*. *Forget gate* memproses *hidden state* sebelumnya (h_{t-1}) dan input saat ini (x_t), kemudian informasi tersebut akan digabung dan diproses dengan menggunakan fungsi *sigmoid* dan menghasilkan output pada *cell state* berupa nilai mendekati 1 menunjukkan bahwa informasi perlu disimpan, sedangkan nilai mendekati 0 berarti informasi tersebut dapat dihapus dari *cell state* (Syahrul, 2023). Alur *forget gate* ditunjukkan pada Gambar 2.4 berikut:



Gambar 2.4. Alur *Forget Gate* pada LSTM

(Sumber: Olah, Christopher)

Nilai *forget gate* dirumuskan pada persamaan 2.3 berikut:

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.3)$$

Dengan:

f_t : *forget gate* pada waktu ke- t

σ : fungsi aktivasi *sigmoid*

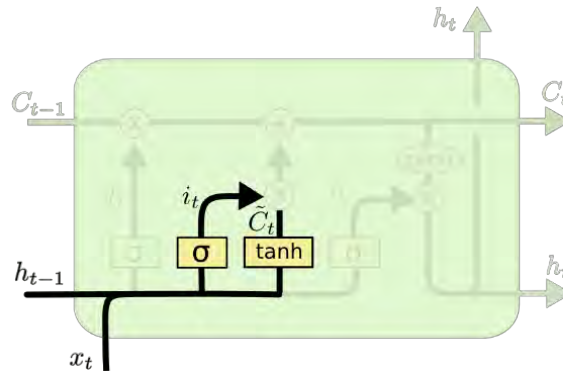
W_f : bobot (*weight*) pada *forget gate*

h_{t-1} : output dari langkah waktu sebelumnya ($t - 1$)

x_t : input pada waktu ke- t

b_f : *bias* pada *forget gate*

Langkah kedua, menentukan informasi yang akan disimpan ke dalam *cell state*, yang melibatkan dua komponen utama, yaitu *input gate* dan aktivasi *tanh*. Pertama, *input gate* menggunakan fungsi aktivasi *sigmoid* untuk menentukan bagian informasi mana yang akan diperbarui. Kedua, lapisan *tanh* menghasilkan kandidat nilai baru ($\bar{C}_t$). Selanjutnya adalah output dari lapisan *sigmoid* yaitu *input gate* dan lapisan *tanh* akan digabungkan untuk memperbarui *cell state* (Syahrul, 2023). Alur *input gate* ditunjukkan pada Gambar 2.5 berikut:



Gambar 2.5. Alur *Input Gate* pada LSTM

(Sumber: Olah, Christopher)

Nilai *input gate* dirumuskan pada persamaan 2.4 berikut:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.4)$$

Dengan:

i_t : *input gate* pada waktu ke-t

σ : fungsi aktivasi *sigmoid*

W_i : bobot (*weight*) pada *input gate*

h_{t-1} : output dari langkah waktu sebelumnya ($t - 1$)

x_t : input pada waktu ke-t

b_i : *bias* pada *input gate*

Nilai kandidat baru dirumuskan pada persamaan 2.5 berikut:

$$\bar{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.5)$$

Dengan:

$\bar{C}_t$: kandidat nilai baru yang akan ditambahkan ke *cell state*

$\tanh$: fungsi aktivasi *tanh*

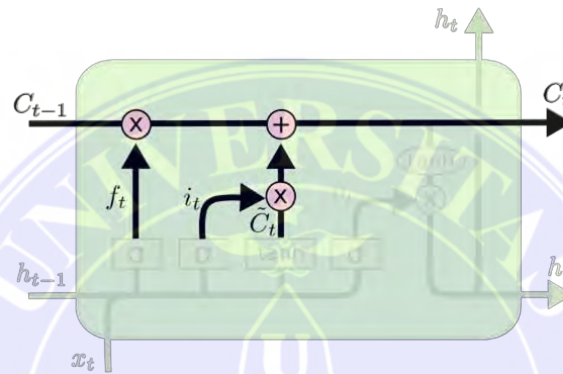
W_c : bobot (*weight*) kandidat *cell state*

h_{t-1} : output dari waktu sebelumnya ($t - 1$)

x_t : input pada waktu ke-t

b_c : *bias* kandidat *cell state*

Langkah ketiga, memperbarui *cell state* dari nilai sebelumnya (C_{t-1}) menjadi nilai *cell state* baru (C_t). Proses ini dilakukan dengan mengalikan *cell state* sebelumnya dengan *forget gate* untuk membuang informasi yang tidak dibutuhkan. Selanjutnya, hasil tersebut ditambahkan dengan $i_t * \bar{C}_t$, yaitu informasi baru yang telah dipilih oleh *input gate* untuk dimasukkan ke dalam memori (Syahrul, 2023). Alur pembaruan *cell state* ini ditunjukkan pada Gambar 2.6 berikut:



Gambar 2.6. Alur *Cell State* pada LSTM

(Sumber: Olah, Christopher)

Nilai *Cell State* dirumuskan pada persamaan 2.6 berikut:

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (2.6)$$

Dengan:

C_t : *cell state* pada waktu ke-t

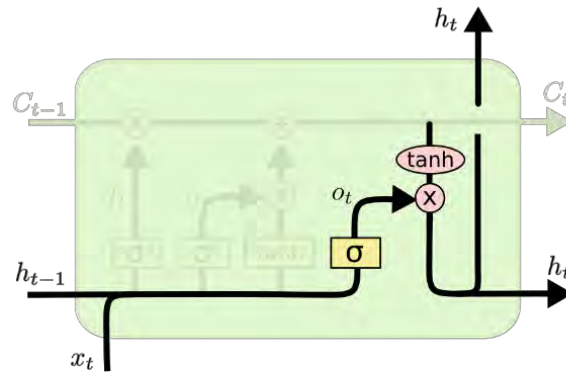
f_t : *forget gate* pada waktu ke-t

C_{t-1} : *cell state* dari langkah waktu sebelumnya ($t - 1$)

i_t : *input gate* pada waktu ke-t

$\bar{C}_t$: kandidat nilai baru yang akan ditambahkan ke *cell state*

Langkah terakhir, menentukan hasil output yang harus sesuai dengan *cell state*, yaitu lapisan *sigmoid* menentukan bagian dari *cell state* yang menjadi output (Syahrul, 2023). Alur *output gate* ditunjukkan pada Gambar 2.7 berikut:



Gambar 2.7. Alur *Output Gate* pada LSTM

(Sumber: Olah, Christopher)

Nilai *Output Gate* dirumuskan pada persamaan 2.7 berikut:

$$O_t = \sigma(W_o \cdot [h_{t-1}x_t] + b_o) \quad (2.7)$$

Dengan:

O_t : *output gate* pada waktu ke-t

σ : fungsi aktivasi *sigmoid*

W_o : bobot (*weight*) pada *output gate*

h_{t-1} : output dari langkah waktu sebelumnya ($t - 1$)

x_t : input pada waktu ke-t

b_o : *bias* pada *output gate*

Output *cell state* dimasukkan kedalam lapisan *tanh* yang mengubah nilai menjadi range -1 dan 1 (Syahrul, 2023). Nilai output pada waktu ke-t dirumuskan menggunakan persamaan 2.8 berikut:

$$h_t = O_t * \tanh(C_t) \quad (2.8)$$

Dengan:

h_t : output pada waktu ke-t

O_t : *output gate* pada waktu ke-t

$\tanh$: fungsi aktivasi *tanh*

C_t : *cell state* pada waktu ke-t

2.5. Normalisasi

Untuk meminimalkan terjadinya *error* perlu dilakukan normalisasi agar data lebih konsisten. Normalisasi berfungsi untuk mengubah skala data menjadi lebih kecil tanpa mengubah nilai asli. Teknik normalisasi yang digunakan dalam penelitian ini adalah *Min-Max Scaling*, di mana nilai pada data aktual diubah menjadi nilai dalam rentang interval [0-1] (Budiprasetyo dkk., 2023). Rumus normalisasi dapat dihitung menggunakan persamaan 2.9 berikut:

$$x_t = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (2.9)$$

Dengan:

- x_t : data hasil normalisasi
- x : data aktual
- x_{min} : nilai minimum data aktual keseluruhan
- x_{max} : nilai maksimum data aktual keseluruhan

2.6. Denormalisasi

Denormalisasi merupakan proses untuk mengembalikan data yang telah di normalisasi ke bentuk aslinya atau data yang sebenarnya. Proses ini dilakukan untuk membandingkan hasil prediksi dengan data asli (Wardani, 2021). Rumus denormalisasi dihitung menggunakan persamaan 2.10 berikut:

$$x_t = x(x_{max} - x_{min}) + x_{min} \quad (2.10)$$

Dengan:

- x_t : data hasil denormalisasi
- x : data prediksi dalam bentuk normalisasi
- x_{min} : nilai minimum data aktual keseluruhan
- x_{max} : nilai maksimum data aktual keseluruhan

2.7. Parameter Evaluasi

Parameter Evaluasi adalah proses mengukur kinerja suatu model dengan menggunakan metrik tertentu. Beberapa parameter metrik evaluasi performa model adalah seperti berikut ini.

2.7.1. Mean Squared Error (MSE)

MSE adalah selisih antara nilai asli dikurangi nilai prediksi yang dikuadratkan, kemudian dijumlahkan keseluruhan lalu membaginya dengan banyak waktu data prediksi. MSE memberikan gambaran tentang sejauh mana model mampu menangkap variasi dan besarnya pergerakan harga. Semakin rendah nilai yang diperoleh, semakin mendekati hasil prediksi model terhadap harga aktual, yang menunjukkan kinerja model yang lebih baik (Pipin dkk., 2023) dirumuskan menggunakan persamaan 2.11 berikut:

$$MSE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n} \quad (2.11)$$

2.7.2. Root Mean Squared Error (RMSE)

RMSE adalah selisih antara nilai asli dikurangi nilai prediksi yang dikuadratkan, kemudian dijumlahkan keseluruhan lalu membaginya dengan banyak waktu data prediksi dan mengakarkannya. Nilai yang lebih kecil menunjukkan kualitas model yang lebih baik (Wathani dkk., 2023), dirumuskan menggunakan persamaan 2.12 berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (2.12)$$

2.7.3. Mean Absolute Error (MAE)

MAE mengukur nilai rata-rata kesalahan mutlak atau absolut selisih antara nilai sebenarnya dengan nilai prediksi. MAE kurang sensitif terhadap outlier dibandingkan RMSE dan MAE dan memberikan ukuran akurasi prediksi yang lebih sederhana. Semakin kecil nilai MAE, semakin baik kinerja model dalam melakukan prediksi (Deshpande, 2023) dirumuskan menggunakan persamaan 2.13 berikut:

$$MAE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)}{n} \quad (2.13)$$

Dengan:

Y_i : nilai data aktual

$\hat{Y}_i$: nilai data prediksi

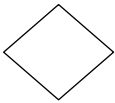
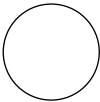
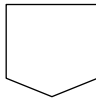
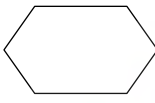
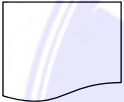
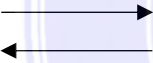
n : banyaknya data

2.8. Flowchart Diagram

Flowchart merupakan representasi grafis dari tahapan penyelesaian masalah dalam bentuk simbol-simbol tertentu, menunjukkan alur logika program secara sistematis. Flowchart berfungsi untuk menggambarkan proses atau algoritma agar mudah dipahami, komunikatif, dan menjadi pedoman dalam pengembangan program. (Khesya, 2021).

Tabel 2.1. Komponen *Flowchart Diagram*

Simbol	Keterangan
	<i>Terminal symbol</i> : Awal atau akhir suatu program (Prosedur).
	<i>Input and Output</i> : menyatakan proses <i>input</i> dan <i>output</i> .

	<i>Processing symbol</i> : pengolahan yang diproses komputer.
	<i>Decision symbol</i> : proses pemilihan berdasarkan pada kondisi yang berlangsung pada dua kemungkinan, ya atau tidak.
	<i>Connector symbol</i> : penyambungan keluar masuk dalam halaman yang sama.
	<i>Offline Connector symbol</i> : penyambungan keluar masuk dalam halaman yang berbeda.
	<i>Preparation symbol</i> : mempersiapkan penyimpanan.
	<i>Predefine Process</i> : pelaksanaan sebuah sub program.
	<i>Document symbol</i> : <i>input</i> berasal dari dokumen cetak. <i>Output</i> akan dicetak pada kertas.
	<i>Flow direction</i> : penghubung antar symbol menyatakan jalannya arus suatu proses.

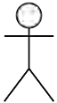
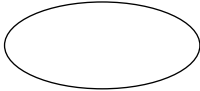
2.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah standar visual untuk memodelkan sistem perangkat lunak maupun non-perangkat lunak. Dengan diagram dan notasi, UML membantu analisis, perancangan, dan dokumentasi sistem, serta mempermudah komunikasi tim pengembang (Huda, 2025)

2.9.1. Use Case Diagram

Use Case Diagram merupakan alat pemodelan untuk merepresentasikan perilaku sistem yang akan dibangun. Diagram ini berperan dalam mendeskripsikan setiap fitur dalam sistem serta menentukan hak akses pengguna atau aktor sesuai dengan fungsionalitas yang telah dirancang (Sumirat, 2023).

Tabel 2.2. Komponen *Use Case Diagram*

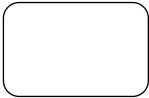
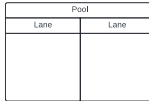
Simbol	Nama	Keterangan
	<i>Actor</i>	Pengguna yang menjalin interaksi dengan <i>use case</i> .
	<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berperilaku dan struktur data dari objek induk (<i>ancestor</i>).
	<i>Include</i>	Mengidentifikasi hubungan antara dua <i>use case</i> .
	<i>Extend</i>	Mengidentifikasi <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber.
	<i>Association</i>	Relasi yang terjalin antara <i>actor</i> dan <i>use case</i> .

2.9.2. Activity Diagram

Activity diagram merupakan diagram yang mengilustrasikan alur kerja dalam suatu sistem, menunjukkan fungsionalitas setiap komponen serta bagaimana pengguna berinteraksi dengan sistem. Diagram ini merepresentasikan urutan aktivitas, mulai dari awal, pengambilan keputusan, hingga akhir, termasuk kemungkinan adanya proses yang berjalan secara parallel (Hasanah, 2020).

Tabel 2.3. Komponen *Activity Diagram*

No	Simbol	Nama	Keterangan
1		Status Awal	Menandakan tindakan awal pada diagram aktivitas.

2		Aktivitas	Aktivitas yang sedang dilakukan dan sedang terjadi.
3		Arah	Menunjukkan perpindahan dari suatu tindakan ke tindakan lainnya.
4		Penggabungan (<i>join</i>)	Menggabungkan beberapa aktivitas yang parallel.
5		<i>Swimlane</i>	Memisahkan tanggung jawab tiap objek terhadap aktivitas yang dilakukan.
6		Percabangan (<i>Decision</i>)	Menunjukkan pemilihan terhadap sebuah kondisi.
7		Status Akhir	Menandakan tindakan akhir pada diagram aktivitas.

2.10. Penelitian Terdahulu

Berikut beberapa penelitian terdahulu terkait dengan penelitian yang akan dilakukan oleh penulis tentang prediksi harga saham menggunakan RNN dan LSTM ditunjukkan sebagai berikut.

Tabel 2.4. Penelitian Terdahulu

No	Peneliti (Tahun)	Judul Penelitian	Hasil Penelitian
1	(Zhu, 2020)	Stock price prediction using the RNN model	Penelitian ini menggunakan RNN untuk memprediksi pasar saham Apple selama sepuluh tahun terakhir. Hasilnya menunjukkan model ini mampu memproses data <i>time series</i> dengan akurasi prediksi hingga 95% dan <i>loss</i> mendekati 0,1%. Prediksi lebih akurat pada <i>timesteps</i> :5 dibandingkan <i>timesteps</i> :10, di mana peningkatan <i>timesteps</i> meningkatkan kesalahan prediksi. Nilai MSE, MAE, dan RMSE meningkat pada <i>timesteps</i> =10 (MSE: 109.22; MAE:

			7.84; RMSE: 10.45) dibandingkan <i>timesteps</i> =5 (MSE: 68.49; MAE: 5.90; RMSE: 8.28), menunjukkan akurasi lebih rendah pada interval waktu lebih panjang.
2	(Pahlawan, 2020)	Peramalan Harga Saham Berdasarkan Faktor Ekonomi Makro Menggunakan Metode <i>Recurrent Neural Network</i> (Studi Kasus: Pt Astra International Tbk)	Penelitian ini memprediksi harga saham PT Astra International Tbk. menggunakan metode RNN dengan arsitektur Elman-RNN, berdasarkan data harga saham dan kurs rupiah terhadap dollar dari Oktober 2014 hingga September 2019. Model diuji dengan berbagai parameter, seperti <i>timestep</i> dan <i>epoch</i> , untuk mengoptimalkan akurasi. Model <i>univariate</i> menghasilkan MAPE 1.546%, sedangkan <i>multivariate</i> (dengan kurs rupiah) mencapai 1.558%. Hasil menunjukkan model <i>univariate</i> lebih optimal, menunjukkan bahwa tambahan variabel kurs rupiah terhadap dollar tidak signifikan meningkatkan performa model.
3	(Wu, 2021)	A Comparative Study of Stock Forecasts by LSTM and RNN Neural Networks	Penelitian ini membandingkan model LSTM dan RNN untuk memprediksi harga saham menggunakan data pasar saham Shanghai dan saham AS (CATL dan NVDA). Untuk saham CATL, LSTM memiliki error 10.4%, sedangkan RNN mencapai 15.3%. Untuk saham NVDA, LSTM mencatat error sebesar 12.6%, sementara RNN mencapai 19.3%. Hasilnya menunjukkan bahwa LSTM lebih unggul dalam memprediksi harga saham dengan akurasi yang lebih tinggi dan lebih baik dan error yang lebih kecil dibandingkan RNN yang cenderung menghasilkan kesalahan lebih besar 4% - 6% lebih tinggi dibandingkan LSTM.
4	(Wathani dkk., 2023)	Prediksi Tren Pergerakan Harga Saham PT Bank	Penelitian ini memprediksi tren harga saham PT Bank Central Asia Tbk (BBCA) menggunakan algoritma

		Central Asia Tbk, Dengan Menggunakan Algoritma <i>Long Short-Term Memory</i> (LSTM)	LSTM dengan data dari 2011 hingga 2022. Model diuji dengan variasi <i>epoch</i> , <i>batch size</i> 1, dan optimasi Adam untuk hasil terbaik. Nilai MAE sebesar 96.92 menunjukkan akurasi yang cukup baik, MSE 16,185.22 menunjukkan kesalahan kuadrat rata-rata yang dapat diterima, dan RMSE 127.22 menunjukkan <i>error</i> prediksi masih dalam batas akurasi yang memadai.
5	(Pan, 2024)	Stock Price Prediction System LSTM Based on <i>Deep Learning</i>	Penelitian ini menggunakan model LSTM untuk memprediksi harga saham dari empat perusahaan teknologi besar: Apple (AAPL), Tesla (TSLA), Microsoft (MSFT), dan Nvidia (NVDA). Hasilnya menunjukkan bahwa LSTM unggul dibandingkan metode tradisional seperti <i>Linear Regression</i> , <i>Support Vector Regression</i> , <i>Decision Tree Regression</i> , dan <i>Gradient Boosting Decision Trees</i> dalam hal akurasi prediksi berdasarkan nilai <i>Mean Squared Error</i> (MSE). LSTM secara konsisten menghasilkan nilai MSE terendah untuk semua saham, menunjukkan kemampuan prediksi yang lebih baik dibandingkan metode lainnya.
6	(Alim, 2023)	Pemodelan <i>Time Series</i> Data Saham LQ45 dengan Algoritma LSTM, RNN, dan ARIMA	Penelitian ini membandingkan metode prediksi saham LQ45 (ANTM dan ICBP) menggunakan ARIMA, LSTM, dan RNN. Hasilnya menunjukkan bahwa LSTM dan RNN lebih unggul daripada ARIMA. LSTM memberikan akurasi terbaik untuk ANTM dengan MAE 0.0079, MAPE 0.0408, dan RMSE 0.013. RNN optimal untuk ICBP dengan MAE 0.0071, MAPE 0.0295, dan RMSE 0.0112, serta waktu pelatihan tercepat. ARIMA memiliki akurasi terendah, membuatnya kurang cocok dibanding metode modern ini.

BAB III

METODOLOGI PENELITIAN

3.1. Perangkat Penelitian

Perangkat yang digunakan dalam penelitian ini adalah komputer personal dengan spesifikasi sebagai berikut:

1. Perangkat Keras (*Hardware*):

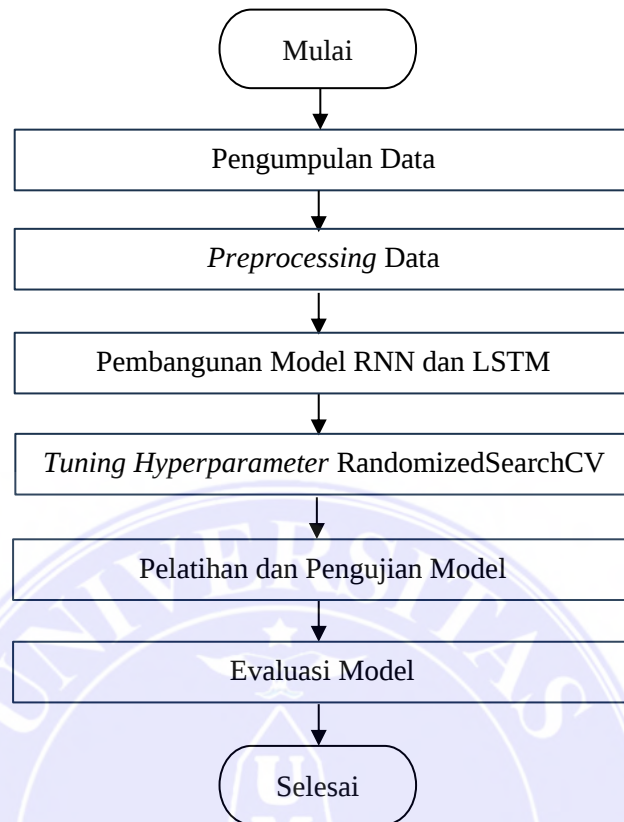
- a. Device : Lenovo LOQ 15IRX9
- b. Processor : Intel® Core™ i5-13450HX CPU @ 2.4GHz
- c. GPU : NVIDIA GeForce RTX 3050 6GB
- d. RAM : 12 GB
- e. SSD : 512 GB

2. Perangkat Lunak (*Software*):

- a. Sistem Operasi : Microsoft Windows 11 64-bit operating system
- b. Bahasa Pemrograman : Python
- c. Code Editor : Visual Studio Code
- d. Office : Microsoft Word 2021

3.2. Tahapan Penelitian

Dalam menyelesaikan permasalahan yang ada dalam penelitian ini, penyesuaian dilakukan agar setiap langkah penelitian terarah dan sistematis. Tahapan penelitian akan menggambarkan struktur yang perlu dilakukan untuk mencapai tujuan penelitian. Berikut tahapan penelitian yang dilakukan:



Gambar 3.1. Flowchart Tahapan Penelitian

3.2.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa dataset harga saham dari PT. Bank Rakyat Indonesia Tbk, dengan kode saham BBRI.JK. Dataset tersebut diperoleh melalui API (*Application Programming Interface*) dari Yahoo Finance dengan tautan: <https://finance.yahoo.com/quote/BBRI.JK/history/>.

Berdasarkan daftar *Forbes 2000 The World's Biggest Companies*, BRI termasuk salah satu perusahaan publik terbesar di dunia. Sebagai bank milik negara (BUMN), BRI memiliki kapitalisasi pasar yang besar, likuiditas tinggi, dan merupakan salah satu saham yang aktif diperdagangkan di Bursa Efek Indonesia (BEI). Menurut data dari IDX, BBRI secara konsisten masuk dalam jajaran Top 10 saham dengan volume dan nilai transaksi tertinggi.

Dengan alasan tersebut, penulis memilih dataset saham BBRI ini sebagai bahan penelitian. Berikut adalah dataset yang digunakan dalam periode 1 Januari 2010 hingga 1 Januari 2025, selama 15 tahun dengan total data 3701 data, tersusun secara harian setiap hari Senin-Jumat, tidak termasuk hari Sabtu-Minggu dan hari libur tertentu karena pasar saham tidak buka.

Tabel 3.1. Dataset Saham BBRI

No	Date	Open	High	Low	Close	Volume
1.	2010/01/04	408.44	411.06	400.59	408.44	125,094,160
2.	2010/01/05	411.06	413.68	408.44	411.06	163,644,326
3.	2010/01/06	405.82	416.29	403.20	405.82	104,842,810
4.	2010/01/07	400.59	405.82	395.35	400.59	101,933,260
5.	2010/01/08	403.20	403.20	395.35	403.20	87,237,006
...	...	...	...	...	...	...
3697.	2024/12/20	3723.15	3768.88	3704.85	3714.00	252,689,600
3698.	2024/12/23	3778.03	3851.21	3759.74	3851.21	167,689,800
3699.	2024/12/24	3860.36	3887.81	3814.62	3842.07	199,536,100
3700.	2024/12/27	3875.15	3894.05	3856.24	3875.15	143,104,400
3701.	2024/12/30	3856.24	3894.05	3846.79	3856.24	153,934,700

Sumber: *finance.yahoo.com*

Dimana:

Date : tanggal

Open : harga pembukaan

High : harga tertinggi

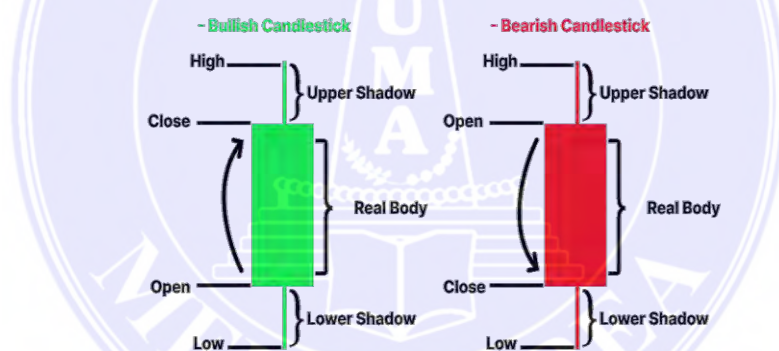
Low : harga terendah

Close : harga penutupan

Volume : jumlah transaksi saham

Salah satu bentuk visualisasi yang paling efektif dalam analisis pergerakan harga saham adalah grafik *candlestick*, karena memberikan gambaran komprehensif pergerakan harga dalam periode waktu tertentu. Dengan menggunakan grafik *candlestick*, kita dapat memperoleh informasi yang lebih mendalam dan bermanfaat, mengidentifikasi titik masuk dan keluar yang tepat, serta mengamati tren yang berkembang pada saham tersebut.

Setiap *candlestick* merepresentasikan empat komponen utama harga, yaitu harga pembukaan (*open*), penutupan (*close*), tertinggi (*high*), dan terendah (*low*). Struktur *candlestick* terbagi menjadi dua bagian utama, yaitu badan (*body*) dan ekor (*shadow*) seperti ditunjukkan pada Gambar 3.2.



Gambar 3.2. Komponen *Candlestick*

Apabila nilai *open* lebih rendah daripada nilai *close*, candlestick akan berwarna hijau dan disebut sebagai *bullish candlestick*, yang mengindikasikan kecenderungan kenaikan harga. Sebaliknya, jika nilai *open* lebih tinggi daripada nilai *close*, candlestick akan berwarna merah dan disebut sebagai *bearish candlestick*, yang menandakan kecenderungan penurunan harga. Garis dibagian atas dan bawah *candle* merepresentasikan nilai *high* dan *low* pada hari tersebut.

3.2.2. *Preprocessing* Data

Tahapan ini dilakukan untuk mempersiapkan data sebelum diolah model prediksi RNN & LSTM dan memastikan data tersebut siap digunakan dalam pelatihan serta evaluasi model. Tahapan *preprocessing* data sebagai berikut:

a. Pembersihan data

Pada proses ini, dilakukan pembersihan data dengan melakukan pengecekan apakah terdapat nilai *missing value* atau *null* pada dataset, kemudian melakukan penghapusan atau menggantinya dengan nilai rata-rata (*mean*).

b. Pembagian Data

Penulis membagi dataset menjadi tiga bagian, yaitu data pelatihan (*training*), data validasi (*validation*), dan data pengujian (*testing*). Data pelatihan digunakan untuk melatih model dengan mempelajari pola dari data historis. Data validasi digunakan untuk mengoptimalkan hyperparameter model dan memantau *overfitting*. Data pengujian digunakan untuk menguji performa akhir model pada data baru yang belum pernah digunakan sebelumnya. Dalam penelitian ini, dataset dibagi dengan persentase 70% *training*, 15% *validation* dan 15% *testing*.

c. Normalisasi

Pada tahap ini, setelah data dibagi, dilakukan normalisasi pada data. Normalisasi hanya menggunakan *min* dan *max* dari data *training*, lalu menerapkan skala yang sama pada *validation* dan *testing*. Penelitian ini berfokus pada prediksi harga saham BBRI dengan menggunakan harga penutupan (*close price*). Berikut merupakan contoh perhitungan normalisasi data menggunakan persamaan 2.9.

Misalkan, kita memiliki data *training* dengan rentang nilai tertentu. Diketahui bahwa nilai maksimum dalam rentang tersebut adalah 3182,40, dan nilai minimumnya adalah 371,78. Nilai input data pertama adalah 408,44. Perhitungan normalisasi dari data *training* adalah sebagai berikut.

$$x_1 = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{408,44 - 371,78}{3182,40 - 371,78} = \frac{36,66}{2810,62} = 0,0130$$

$$x_2 = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{411,06 - 371,78}{3182,40 - 371,78} = \frac{39,28}{2810,62} = 0,0140$$

Perhitungan dilanjutkan hingga input ke-n data. Berikut merupakan hasil normalisasi data harga saham harian PT Bank Rakyat Indonesia Tbk berdasarkan harga penutupan dapat dilihat pada Tabel 3.2.

Tabel 3.2. Perhitungan Normalisasi Harga Penutupan BBRI

No	Date	Close Price	Normalisasi
1.	04/01/2010	408,44	0,0130
2.	05/01/2010	411,06	0,0140
3.	06/01/2010	405,82	0,0121
4.	07/01/2010	400,59	0,0103
5.	08/01/2010	403,20	0,0112
...	...	...	...

d. Pembentukan *Sliding Window*

Data harga penutupan saham yang telah dinormalisasi dibentuk menggunakan pendekatan *sliding window* agar dapat diproses oleh model *deep learning*. Teknik ini memungkinkan model mempelajari pola historis dengan menggunakan data harga saham sebelumnya sebagai input untuk memprediksi harga pada hari berikutnya.

Pada contoh ini, digunakan *timestep* sebanyak 5. Artinya, model menggunakan data harga saham dari 5 hari terakhir (X_1 – X_5) sebagai input, dan harga saham pada hari ke-6 (Y) sebagai output atau target prediksi. Proses ini dilakukan secara berulang hingga seluruh data historis terbentuk menjadi pasangan input-output seperti ditunjukkan pada Tabel 3.3.

Tabel 3.3. Proses Pembentukan *Sliding Window* Harga Penutupan BBRI

X_1	X_2	X_3	X_4	X_5	Target (Y)
0,0130	0,0140	0,0121	0,0103	0,0112	0,0112
0,0140	0,0121	0,0103	0,0112	0,0112	0,0140
0,0121	0,0103	0,0112	0,0112	0,0140	0,0140
0,0103	0,0112	0,0112	0,0140	0,0140	0,0140
0,0112	0,0112	0,0140	0,0140	0,0140	0,0130
...	...	...	...	...	...

3.2.3. Pembangunan Model RNN dan LSTM

Tahap selanjutnya, membangun model RNN dan LSTM menggunakan framework TensorFlow/Keras. Model dikembangkan dalam bentuk kode program, termasuk penentuan hyperparameter yang berpengaruh terhadap kinerja model selama pelatihan. Hyperparameter merupakan parameter yang ditetapkan oleh pengguna sebelum pelatihan. Untuk memperoleh kombinasi hyperparameter terbaik, penelitian ini menerapkan *tuning hyperparameter* menggunakan metode RandomSearchCV. Daftar hyperparameter yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.4.

Tabel 3.4. Hyperparameter yang Digunakan

Hyperparameter	Keterangan
Timesteps	Jumlah langkah waktu dalam satu sekuens input yang diproses dalam satu waktu. Peneliti menentukan nilai <i>timestep</i> 30 hari perdagangan sebelumnya untuk menentukan prediksi harga saham berikutnya.
{Model}_Layers	Menentukan berapa banyak layer RNN dan LSTM. Peneliti menentukan 1 - 2 layer.
{Model}_Units	Menentukan berapa banyak <i>neuron</i> atau unit RNN dan LSTM. Peneliti menentukan jumlah neuron atau unit sebanyak 64, 128 dan 256 pada tiap layer.
{Model}_Dropout	Teknik regulasi yang menonaktifkan <i>neuron</i> selama proses pelatihan untuk mengurangi <i>overfitting</i> . Peneliti menentukan <i>dropout</i> sebesar 0,2 dan 0,3 sudah cukup.
Epochs	Menentukan berapa kali seluruh dataset latih diproses atau model melihat seluruh dataset latih. Peneliti menentukan maksimal <i>epoch</i> sebanyak 100.
Batch_Size	Menentukan berapa banyak sampel yang diproses dalam satu waktu, peneliti menentukan nilai sebesar 64 pada <i>batch size</i> .
Optimizer & Learning Rate	<i>Optimizer</i> merupakan algoritma untuk menyesuaikan bobot model dan mengontrol cara model belajar dari <i>error</i> dan <i>learning rate</i> adalah besaran langkah pembaruan bobot selama pelatihan. Peneliti menentukan optimizer <i>Adam</i> dengan <i>learning rate</i> 0,001 dan 0,0001.
Loss	Menghitung perbedaan atau selisih antara nilai prediksi dan nilai target. Fungsi ini digunakan untuk mengevaluasi kinerja model selama pelatihan. Peneliti menentukan fungsi <i>loss</i> yaitu MSE.

3.2.4. *Tuning Hyperparameter RandomizedSearchCV*

RandomizedSearchCV secara acak mengambil sejumlah kombinasi dari ruang pencarian hyperparameter, sesuai jumlah iterasi yang ditentukan. Penentuan variasi nilai hyperparameter tidak memiliki aturan baku, sehingga dapat disesuaikan dengan kebutuhan peneliti.

Metode RandomizedSearchCV dipilih karena lebih efisien dalam mengurangi waktu pencarian dan sumber daya komputasi dibandingkan metode seperti GridSearchCV, yang menguji semua kemungkinan kombinasi hyperparameter secara menyeluruh. Selain itu proses pencarian kombinasi hyperparameter secara manual satu per satu tidak hanya kurang praktis, tetapi juga melelahkan, dan memakan waktu.

3.2.5. *Pelatihan dan Pengujian Model*

Setelah kombinasi hyperparameter terbaik diperoleh, model dilatih secara penuh menggunakan seluruh data pelatihan. Model yang telah terlatih kemudian akan dievaluasi menggunakan data uji untuk mengukur performa dengan metrik yang sudah ditentukan.

Proses pelatihan RNN dan LSTM melibatkan tahap *Forward Propagation Through Time* (FPTT), perhitungan *loss* dan *Backpropagation Through Time* (BPTT). Data yang digunakan pada sampel pertama pelatihan ditunjukkan pada Tabel 3.5, yang berasal dari data pada Tabel 3.3.

Tabel 3.5. Sampel Pertama Pelatihan Model

X_1	X_2	X_3	X_4	X_5	Target (Y)
0,0130	0,0140	0,0121	0,0103	0,0112	0,0112

1. Forward Propagation Through Time (FPTT) Pada RNN

FPTT pada RNN dilakukan dengan menghitung nilai setiap *neuron* dari *input layer* ke *hidden layer* hingga *output layer*. *Hidden state* dan output dihitung menggunakan persamaan 2.1 dan 2.2. Data yang digunakan sebagai input adalah data yang telah dinormalisasi, ditunjukkan pada Tabel 3.5.

Dalam implementasi secara komputasi, bobot (*weight*) dan bias diinisialisasi secara acak. Namun, untuk mempermudah pemahaman dalam contoh perhitungan matematis ini, bobot diinisialisasi dan disederhanakan dengan nilai 1 dibagi jumlah variabel input. Misalnya, untuk 5 input, bobot dihitung sebagai 1/5. Oleh karena itu, nilai bobot dan bias yang digunakan dalam perhitungan matematis ini adalah sebagai berikut:

Bobot Input (W_{xh})	: 0,447
Bobot Hidden (W_{hh})	: 0,447
Bobot Output (W_{hy})	: 0,447
Bias Hidden (b_h)	: 0
Bias Output (b_y)	: 0
hidden state awal (h_t)	: 0
Fungsi aktivasi	: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Fungsi aktivasi untuk *hidden state* adalah *tanh*, sehingga:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = \tanh(W_{hy}h_t + b_y)$$

Perhitungan untuk *timestep* pertama (t_1):

$$h_1 = \tanh(0,447 \times 0,0130 + 0,447 \times 0 + 0)$$

$$= \tanh(0,005811)$$

$$= 0,0058$$

$$\begin{aligned} y_1 &= \tanh(0,447 \times 0,0058 + 0) \\ &= \tanh(0,002593) \\ &= 0,0026 \end{aligned}$$

Perhitungan untuk *timestep* kedua (t_2):

$$\begin{aligned} h_2 &= \tanh(0,447 \times 0,0140 + 0,447 \times 0,0058 + 0) \\ &= \tanh(0,006258 + 0,002593) \\ &= \tanh(0,008851) \\ &= 0,0088 \end{aligned}$$

$$\begin{aligned} y_2 &= \tanh(0,447 \times 0,0088 + 0) \\ &= \tanh(0,003934) \\ &= 0,0039 \end{aligned}$$

Proses dilanjutkan hingga *timestep* kelima (t_5), sehingga nilai *hidden state* dan nilai output didapatkan seperti pada Tabel 3.6.

Tabel 3.6. Hasil Perhitungan RNN

RNN	t_1	t_2	t_3	t_4	t_5
x_t (input)	0,0130	0,0140	0,0121	0,0103	0,0112
h_t	0,0058	0,0088	0,0093	0,0088	0,0089
y_t	0,0026	0,0039	0,0042	0,0039	0,0040

2. Perhitungan *loss* pada RNN

Perhitungan *loss* pada RNN dilakukan dengan membandingkan nilai output (y_5) dengan target aktual (Y) menggunakan fungsi MSE adalah sebagai berikut:

$$\text{Target output (Y)} = 0,0112$$

$$\text{Prediksi RNN (} y_5 \text{)} = 0,0040$$

Karena ini hanya satu sampel, $n = 1$. maka perhitungan *loss*-nya:

$$\begin{aligned}
 L &= \frac{1}{n}(y_{target} - y_{pred})^2 \\
 &= (Y - y_5)^2 \\
 &= (0,0112 - 0,0040)^2 \\
 &= 0,00005184
 \end{aligned}$$

3. *Backpropagation Through Time (BPTT)* Pada RNN

Langkah selanjutnya adalah menghitung gradien dari *loss* terhadap bobot output dan bias output, gradien terhadap bobot hidden, bobot input, bias hidden, dan memperbarui bobot menggunakan algoritma optimasi. Proses ini dilakukan untuk seluruh *timestep* dengan propagasi *error* kebelakang hingga *timestep* pertama.

4. *Forward Propagation Through Time (FPTT)* pada LSTM

FPTT pada LSTM menggunakan *gated mechanisms* dengan menghitung seluruh fungsi *gates* unit pada setiap *neurons* secara berurut dengan persamaan yang telah diketahui sebelumnya. Data pertama yang digunakan sebagai input sama seperti pada RNN sebelumnya,

Bobot dan bias LSTM yang digunakan adalah sebagai berikut:

Bobot : $W_f, W_i, W_c, W_o = 0,447$

Bias : $b_f = 1, b_i, b_c, b_o = 0$

Hidden state awal (h_t) : 0

Cell state awal (C_t) : 0

Fungsi aktivasi : $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, dan sigmoid = $\sigma(x) = \frac{1}{1 + e^{-x}}$

Perhitungan untuk *timestep* pertama (t_1):

1) Forget Gate (f_t):

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ &= \sigma(0,447 \cdot [0 + 0,0130] + 1) \\ &= \sigma([0,005811 + 1]) = \sigma(1,0058) \\ &= \frac{1}{1 + e^{-1,0058}} = 0,7322 \end{aligned}$$

2) Input Gate (i_t):

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ &= \sigma(0,447 \cdot [0 + 0,0130] + 0) \\ &= \sigma(0,005811) \\ &= \frac{1}{1 + e^{-0,005811}} = 0,5015 \end{aligned}$$

3) Candidate Cell State ($\bar{C}_t$):

$$\begin{aligned} \bar{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ &= \tanh(0,447 \cdot [0 + 0,0130] + 0) \\ &= \tanh(0,005811) = 0,0058 \end{aligned}$$

4) Cell State Update (C_t):

$$\begin{aligned} C_t &= f_t * C_{t-1} + i_t * \bar{C}_t \\ &= (0,7322 \times 0) + (0,5015 \times 0,0058) \\ &= 0 + 0,0029 = 0,002 \end{aligned}$$

5) Output Gate (O_t) :

$$\begin{aligned} O_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ &= \sigma(0,447 \cdot [0 + 0,0130] + 0) \\ &= \sigma(0,005811) \\ &= \frac{1}{1 + e^{-0,005811}} = 0,5015 \end{aligned}$$

6) Hidden State (h_t):

$$\begin{aligned} h_t &= O_t * \tanh(C_t) \\ &= 0,5015 * \tanh(0,0029) \\ &= 0,5015 \times 0,0029 = 0,0015 \end{aligned}$$

Proses dilanjutkan hingga *timestep* kelima (t_5), menggunakan nilai C_t dan h_t dari tiap *timestep* yang dihitung sebelumnya dibawa ke *timestep* berikutnya. Sehingga nilai tiap proses perhitungan LSTM di setiap *gates* didapatkan seperti pada Tabel 3.7.

Tabel 3.7. Hasil Perhitungan LSTM

LSTM	t_1	t_2	t_3	t_4	t_5
x_t (input)	0,0130	0,0140	0,0121	0,0103	0,0112
f_t	0,7322	0,7325	0,7325	0,7324	0,7326
i_t	0,5015	0,5017	0,5017	0,5016	0,5017
$\bar{c}_t$	0,0058	0,0069	0,0067	0,0063	0,0070
C_t	0,0029	0,0056	0,0075	0,0087	0,0099
O_t	0,5015	0,5017	0,5017	0,5016	0,5017
h_t	0,0015	0,0028	0,0038	0,0044	0,0050

5. Perhitungan Loss Pada LSTM

Perhitungan *loss* pada LSTM dilakukan dengan membandingkan nilai output (h_5) dengan target aktual (Y) menggunakan fungsi *loss* MSE adalah sebagai berikut:

$$\text{Target output (Y)} = 0,0112$$

$$\text{Prediksi LSTM (} h_5 \text{)} = 0,0050$$

Karena ini hanya satu sampel, $n = 1$, maka perhitungan *loss*-nya:

$$\begin{aligned}
 L &= \frac{1}{n}(y_{\text{target}} - y_{\text{pred}})^2 \\
 &= (Y - h_5)^2 \\
 &= (0,0112 - 0,0050)^2 \\
 &= 0,00003844
 \end{aligned}$$

6. Backpropagation Through Time (BPTT) Pada LSTM

BPTT pada LSTM menghitung gradien dari *loss* terhadap bobot (W_f, W_i, W_c, W_o) dan bias (b_f, b_i, b_c, b_o), serta gradien terhadap *hidden state* (h_t) dan *cell state* (C_t). Proses ini melibatkan propagasi *error* ke *timestep* sebelumnya untuk memperbarui parameter model secara bertahap menggunakan algoritma optimasi.

3.2.6. Evaluasi Model

Sebelum evaluasi model, perlu dilakukan denormalisasi agar hasil prediksi dapat dibandingkan dengan data aktual, karena sebelumnya data melalui proses normalisasi. Nilai yang digunakan dalam proses denormalisasi untuk memperoleh hasil prediksi adalah nilai (y_t) pada RNN dan nilai (h_t) pada LSTM yang ditunjukkan pada Tabel 3.8.

Tabel 3.8. Nilai Output Prediksi RNN dan LSTM

Date	Close Price	Normalisasi	RNN (y_t)	LSTM (h_t)
04/01/2010	408,44	0,0130	0,0026	0,0015
05/01/2010	411,06	0,0140	0,0039	0,0028
06/01/2010	405,82	0,0121	0,0042	0,0038
07/01/2010	400,59	0,0103	0,0039	0,0044
08/01/2010	403,20	0,0112	0,0040	0,0050

Berikut contoh perhitungan denormalisasi data menggunakan persamaan 2.10.

$$\begin{aligned}
 x_{denormalisasi} &= x(x_{max} - x_{min}) + x_{min} \\
 x_{1RNN} &= 0,0026(3182,40 - 371,78) + 371,78 \\
 &= 0,0026(2810,62) + 371,78 \\
 &= 7,30 + 371,78 \\
 &= 379,09
 \end{aligned}$$

$$\begin{aligned}
x_{1LSTM} &= 0,0015(3182,40 - 371,78) + 371,78 \\
&= 0,0015(2810,62) + 371,78 \\
&= 4,21 + 371,78 \\
&= 375,99
\end{aligned}$$

Sehingga didapatkan nilai baru pada model RNN dan LSTM. Berikut merupakan hasil denormalisasi data yang ditunjukkan pada Tabel 3.9.

Tabel 3.9. Hasil Perhitungan Denormalisasi RNN dan LSTM

Date	Close Price	RNN	LSTM
04/01/2010	408,44	379,09	375,99
05/01/2010	411,06	382,74	379,65
06/01/2010	405,82	383,58	382,46
07/01/2010	400,59	382,74	384,15
08/01/2010	403,20	383,02	385,83

Dari hasil perhitungan denormalisasi di atas didapatkan tabel seperti pada Tabel 3.10. berikut.

Tabel 3.10. Hasil Perhitungan Sampel Pertama RNN dan LSTM

X_t	X_1	X_2	X_3	X_4	X_5	Target (Y)
RNN	379,09	382,74	383,58	382,74	383,02	403,20
LSTM	375,99	379,65	382,46	384,15	385,83	403,20

Evaluasi model dilakukan untuk mengukur sejauh mana model RNN dan LSTM mampu menangkap pola dalam data dan menghasilkan prediksi yang mendekati nilai target aktual (Y). Perhitungan sampel diatas menggunakan data dari hari ke satu hingga hari ke lima. Hasil denormalisasi pada data kelima (X_5) yang akan digunakan dalam memprediksi dan kemudian dibandingkan dengan nilai target aktual (Y). Berikut adalah contoh perhitungan matematis menggunakan metrik evaluasi seperti MSE, RMSE, dan MAE.

1. Mean Squared Error (MSE)

MSE dihitung dengan menggunakan rumus dari persamaan 2.11.

MSE untuk RNN:

$$\begin{aligned} MSE_{RNN} &= \frac{(403,20 - 383,02)^2}{1} = \frac{(20,18)^2}{1} \\ &= 407,23 \end{aligned}$$

MSE untuk LSTM:

$$\begin{aligned} MSE_{LSTM} &= \frac{(403,20 - 385,83)^2}{1} = \frac{(17,37)^2}{1} \\ &= 301,72 \end{aligned}$$

2. Root Mean Squared Error (RMSE)

RMSE dihitung dengan menggunakan rumus dari persamaan 2.12.

RMSE untuk RNN:

$$RMSE_{RNN} = \sqrt{407,23} = 20,18$$

RMSE untuk LSTM:

$$RMSE_{LSTM} = \sqrt{301,72} = 17,37$$

3. Mean Absolute Error (MAE)

MAE dihitung dengan menggunakan rumus dari persamaan 2.13.

MAE untuk RNN:

$$\begin{aligned} MAE_{RNN} &= \frac{403,20 - 383,02}{1} \\ &= 20,18 \end{aligned}$$

MAE untuk LSTM:

$$\begin{aligned} MAE_{LSTM} &= \frac{403,20 - 383,02}{1} \\ &= 17,37 \end{aligned}$$

Nilai ketiga metrik yang dihitung berasal dari satu iterasi menggunakan sampel pertama pelatihan, yang terdiri dari 5 *timestep* yaitu data harga pada hari ke-1 hingga ke-5 dengan target pada hari ke-6. Proses ini diulang untuk setiap sampel berikutnya dengan menggeser satu *window* data (misalnya, untuk sampel kedua, inputnya adalah data dari hari ke-2 hingga ke-6, dengan target pada hari ke-7, dan seterusnya). Setelah seluruh sampel diproses, kita dapat menghitung rata-rata MSE, RMSE, dan MAE untuk menilai kinerja model secara keseluruhan. Untuk mempermudah prosesnya, semua *timestep* atau pasangan input-output diproses secara komputasi.

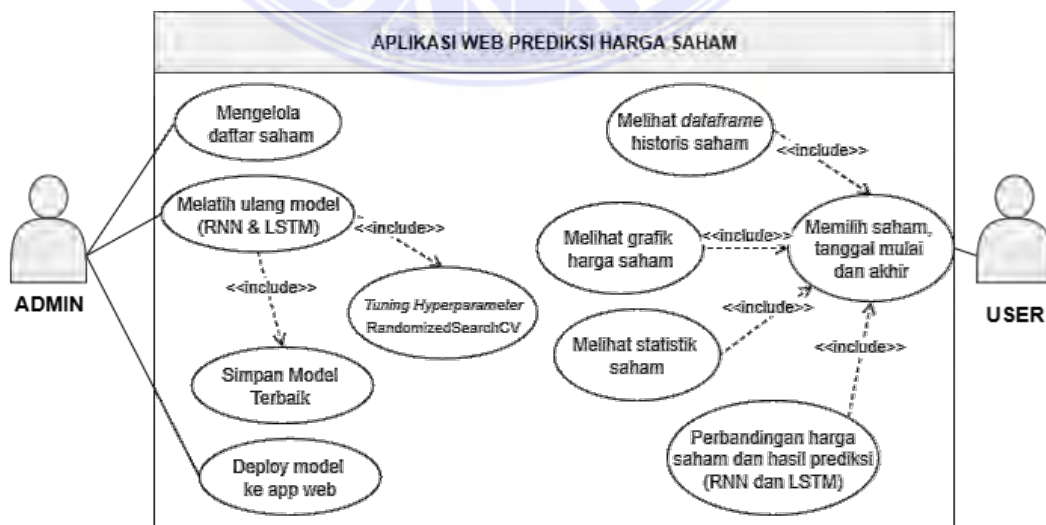
3.3. Tahapan Perancangan

Perancangan yang akan dibangun didalamnya memuat UML dan perancangan *User Interface* (UI).

3.3.1. Unified Modeling Language (UML)

1. Use Case Diagram

Proses prediksi harga saham menggunakan aplikasi web ini menggambarkan bagaimana pengguna berinteraksi dengan sistem.

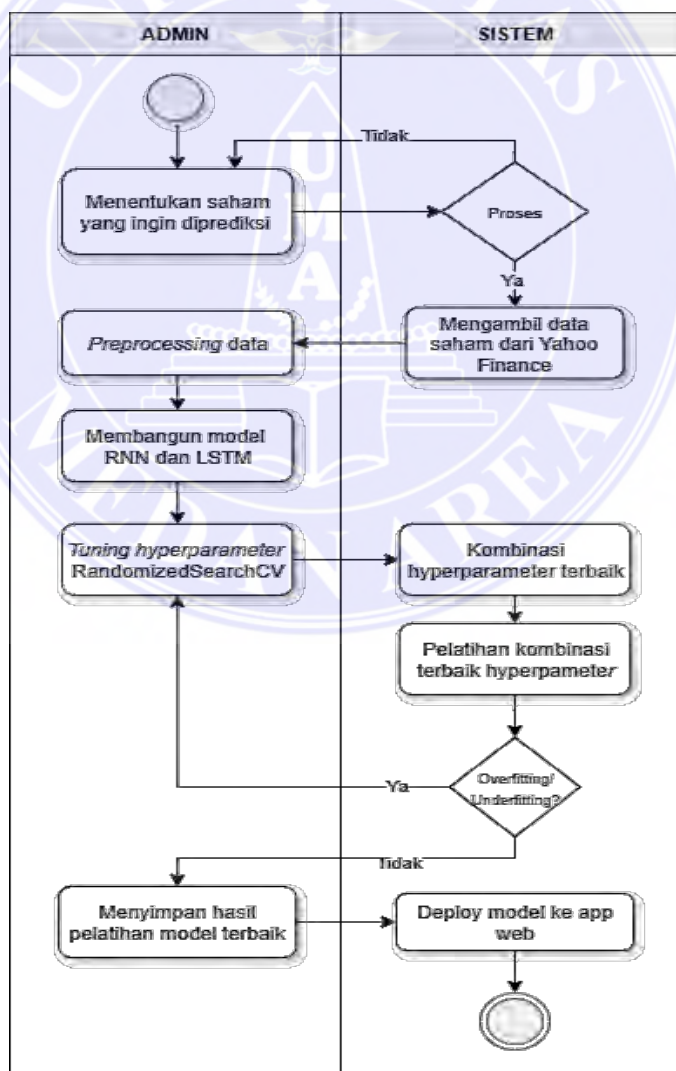


Gambar 3.3. Use Case Diagram Aplikasi Web

2. Activity Diagram

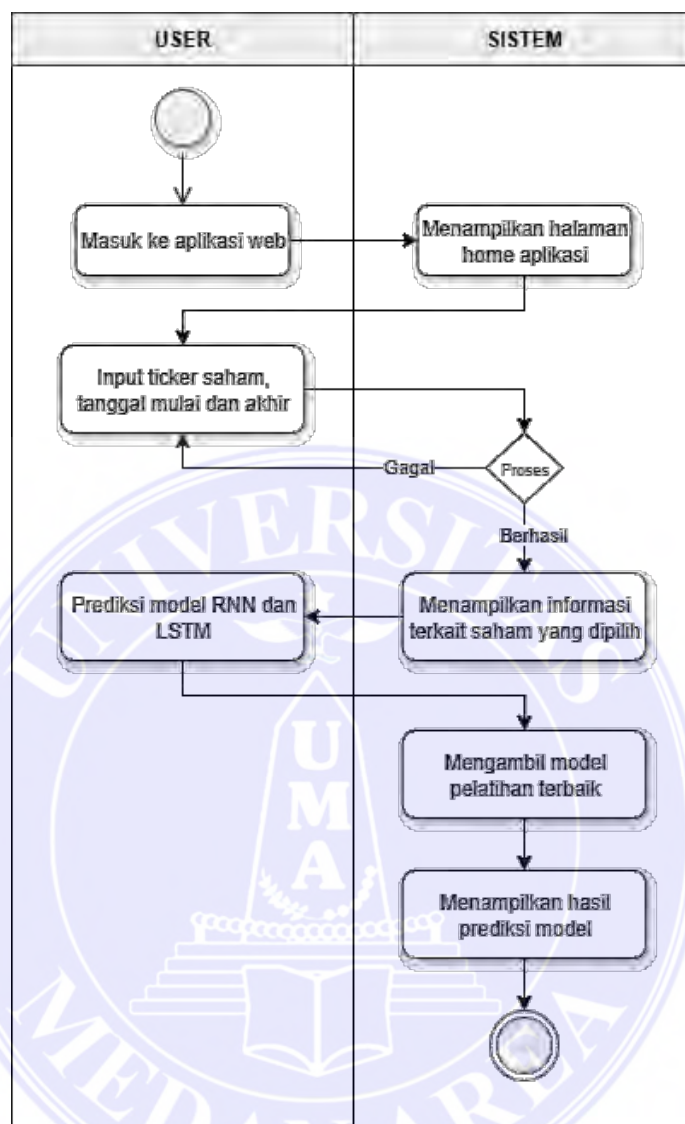
Pada bagian ini, peneliti menjelaskan berbagai proses yang terjadi dalam program atau aplikasi web prediksi harga saham. Terdapat dua proses, yaitu pelatihan model terbaik yang hanya dilakukan oleh admin, dan penggunaan model terlatih untuk mengakses serta menampilkan hasil prediksi dalam aplikasi web. Pengguna hanya dapat melihat informasi saham dan hasil prediksi yang disajikan dalam aplikasi web berdasarkan hasil pelatihan terbaik yang dilakukan oleh admin.

a. Proses Pelatihan Model Terbaik



Gambar 3.4. Activity Diagram Pelatihan Model Terbaik Oleh Admin

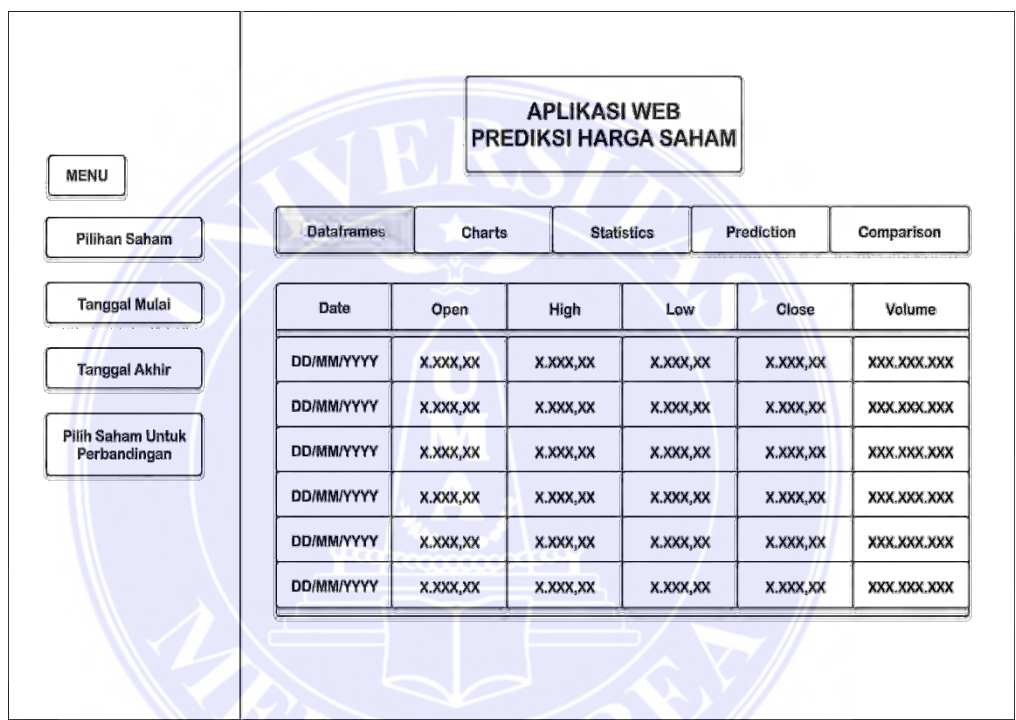
b. Proses Menampilkan Hasil Prediksi



Gambar 3.5. Activity Diagram Menampilkan Hasil Prediksi Oleh User

3.3.2. User Interface (UI)

Perancangan UI merupakan proses mendesain tampilan antarmuka aplikasi web agar pengguna dapat berinteraksi dengan sistem secara efektif. *User Interface* dirancang untuk menyajikan informasi dengan jelas, memudahkan navigasi, serta meningkatkan pengalaman pengguna saat menggunakan aplikasi. Berikut merupakan desain UI halaman aplikasi web pada Gambar 3.6 berikut.



Gambar 3.6. Desain UI Halaman Aplikasi Web

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diperoleh kesimpulan sebagai berikut:

1. Model RNN dan LSTM yang digunakan untuk memprediksi harga saham BBRI menggunakan skema data 70 % *training*, 15 % *validation* dan 15 % *testing*, dengan konfigurasi *timestep* 30 hari perdagangan dan metode *hyperparameter tuning* RandomizedSearchCV. Berdasarkan hasil tuning, konfigurasi terbaik untuk masing-masing model adalah sebagai berikut:

- a. Model RNN: menggunakan dua layer dengan masing-masing 128 neuron, dropout sebesar 0,2, optimizer Adam dengan *learning rate* sebesar 0,001.
- b. Model LSTM: menggunakan satu layer dengan 256 neuron, dropout sebesar 0,2, optimizer Adam dengan *learning rate* sebesar 0,001.

Kedua model dilatih menggunakan jumlah *epoch* 100 dan *batch size* 64.

2. Berdasarkan hasil penelitian yang dilakukan untuk memprediksi harga saham BBRI pada data uji, RNN mendapatkan nilai MSE sebesar 9.340,06, RMSE sebesar 96,64, dan MAE sebesar 76,57. Sedangkan pada LSTM mendapatkan nilai MSE sebesar 5.535,33, RMSE sebesar 74,39, dan MAE sebesar 57,25. Berdasarkan ketiga metrik, model LSTM menunjukkan performa yang lebih baik dibandingkan RNN dalam memprediksi harga saham BBRI. Hal ini ditunjukkan oleh nilai *error* yang lebih rendah.

3. Hasil prediksi saham BBRI untuk 7 hari ke depan cenderung mengalami penurunan. Perhitungan nilai rata-rata *error* model prediksi RNN menghasilkan nilai *error* sebesar 1,71%, sementara LSTM menghasilkan nilai *error* sebesar 1,31%. Hal ini menunjukkan bahwa tingkat kesalahan prediksi pada LSTM lebih kecil, mengindikasikan bahwa LSTM lebih baik dibandingkan dengan RNN.

5.2. Saran

Terdapat berbagai keterbatasan sehingga memerlukan pengembangan untuk penelitian selanjutnya. Beberapa saran yang dapat dipertimbangkan antara lain:

1. Disarankan mempertimbangkan penggunaan model GRU (*Gated Recurrent Unit*), yang juga merupakan varian RNN.
2. Penelitian ini hanya menggunakan satu variabel (*univariate*) yaitu harga penutupan (*close price*), disarankan menambahkan variabel lain (*multivariate*) seperti data makroekonomi (CPI), indikator teknikal, data sentimen dan lainnya.
3. Untuk meningkatkan performa model *deep learning*, disarankan menggunakan *hardware* dengan spesifikasi tinggi agar mempercepat proses pelatihan model dan eksplorasi terhadap lebih banyak kombinasi hyperparameter.
4. Pengembangan selanjutnya disarankan dalam bentuk aplikasi web. Saat ini, pelatihan data dan kombinasi hyperparameter terbaik berasal dari saham BBRI yang hanya valid pada saham tersebut. Oleh karena itu, untuk saham lain perlu ditambahkan hasil pelatihan data untuk masing-masing jenis saham.

DAFTAR PUSTAKA

- Alim, M. N. (2023). Pemodelan Time Series Data Saham LQ45 Dengan Algoritma LSTM, RNN, Dan ARIMA. *Prosiding Seminar Nasional Matematika*, 6, 694–701. <https://journal.unnes.ac.id/sju/index.php/prisma/>
- Arfan, A., & Lussiana ETP, Dan. (2019). Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory. *Universitas Gunadarma Jl. Margonda Raya No*, 3(1). <https://www.ofx.com>
- Budiprasetyo, G., Hani'ah, M., & Aflah, D. Z. (2023). Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 8(3), 164–172. <https://doi.org/10.25077/teknosi.v8i3.2022.164-172>
- Deshpande, V. (2023). Implementation of Long Short-Term Memory (LSTM) Networks for Stock Price Prediction. *Research Journal of Computer Systems and Engineering*, 4(2), 60–72. <https://doi.org/10.52710/rjcs.74>
- Faridah, N., & Sugiantoro, B. (2023). Analisis Optimasi Pada Algoritma Long Short-Term Memory Untuk Memprediksi Harga Saham. *Jurnal Media Informatika Budidarma*, 7(1), 575. <https://doi.org/10.30865/mib.v7i1.5421>
- Gumelar, F., Adha, F. Z., Rafi, F. A., & Pontoh, R. S. (2022). Peramalan Harga Saham Bank BUMN Indonesia Menggunakan Long Short-Term Memory (LSTM). *Seminar Nasional Statistika Aktuaria I* (2022). <http://prosiding.statistics.unpad.ac.id>
- Hanafiah, A., Arta, Y., Nasution, H. O., & Lestari, Y. D. (2023). Penerapan Metode Recurrent Neural Network Dengan Pendekatan Long Short-Term Memory (LSTM) Untuk Prediksi Harga Saham. *Bulletin Of Computer Science Research*, 4(1), 27–33. <https://doi.org/10.47065/bulletincsr.v4i1.321>
- Hasanah, F.N., Untari, R. S., (2020). Rekayasa Perangkat Lunak. <https://press.umsida.ac.id/index.php/umsidapress/article/view/978-623-6833-89-6>
- Huda, M. (2025). Pemodelan Sistem Informasi Dengan UML. https://www.academia.edu/127867864/PEMODELAN_SISTEM_INFORMASI_DENGAN_UML?source=swp_share
- Iman, F. N., & Wulandari, D. (2023). Prediksi Harga Saham Menggunakan Metode Long Short-Term Memory. *Jurnal Ilmu Komputer Dan Pendidikan*. <https://journal.mediapublikasi.id/index.php/logic>

- Janastu, I. N. C., & Wutsqa, D. U. (2024). Prediksi Harga Saham Pada Sektor Perbankan Menggunakan Algoritma LSTM. *Jurnal Statistika Dan Sains Data*, 1(2), 1–14. <https://Journal.Student.Uny.Ac.Id/Index.Php/Jssd>
- Jaya, G. P. (2022). *Rancang Bangun Aplikasi Prediksi Harga Saham Menggunakan Deep Learning Dengan Algoritma Long Short-Term Memory*. (Skripsi, Sekolah Tinggi Manajemen Informatika dan Komputer Bani Saleh. <https://github.com/GesangPJ/skripsi-prediksi-harga-saham-deeplearning-lstm/blob/main/README.md>
- Karno, A. S. B. (2020). Prediksi Data Time Series Saham Bank BRI Dengan Mesin Belajar LSTM (Long Short-Term Memory). *Journal Of Informatic and Information Security*, 1(1), 1–8. <https://Doi.Org/10.31599/Jiforty.V1i1.133>
- Khesya, N. (2021). Mengenal Flowchart Dan Pseudocode Dalam Algoritma Dan Pemrograman. <https://Doi.Org/10.31219/Osf.Io/Dq45e>
- Kustina, L., Safitri, O., & Anwar, S. (2019). Kebijakan Deviden Dan Capital Gain: Pengaruhnya Terhadap Harga Saham. *Jurnal Investasi*, 5(1), 24–37. <https://Doi.Org/10.31943/Investasi.V5i1.16>
- Kustodian Sentral Efek Indonesia. (2024). *Statistik Publik Pasar Modal Indonesia 2024*. https://www.ksei.co.id/files/Statistik_Publik_Januari_2024_v3.pdf
- Kusuma, N. P. N. (2024). Prediksi Harga Saham Blue Chip Pada Indeks IDX 30 Menggunakan Algoritma RNN, *Jurnal Ekonomi Bisnis Vol 23 No 1: 90 - 97*. <https://pdfs.semanticscholar.org/7e66/86cb16ef69123b27414a213fcb8dfbe7914a.pdf>
- Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, And Applications. *Information*, 15(9), 517. <https://Doi.Org/10.3390/Info15090517>
- Pahlawan, M. R. (2020). *Peramalan Harga Saham Berdasarkan Faktor Ekonomi Makro Menggunakan Metode Recurrent Neural Network (Studi Kasus: PT Astra International Tbk.)* (Tugas Akhir, Institut Teknologi Sepuluh Nopember). <http://Repository.Its.Ac.Id/Id/Eprint/78228>
- Pan, X. (2024). Stock Price Prediction System LSTM Based on Deep Learning. *Journal of Economics and Public Finance*, 10(4), p235. <https://doi.org/10.22158/jepf.v10n4p235>
- Pipin, S. J., Purba, R., & Kurniawan, H. (2023). Prediksi Saham Menggunakan Recurrent Neural Network (RNN-LSTM) Dengan Optimasi Adaptive Moment Estimation. *Journal Of Computer System and Informatics (Josyc)*, 4(4), 806–815. <https://Doi.Org/10.47065/Josyc.V4i4.4014>

- Ramadhan, M. F. (2024). *Prediksi Harga Bitcoin Menggunakan Metode LSTM*. (Skripsi, Universitas Jambi). <https://Repository.Unja.Ac.Id/62231/>
- Riyantoko, P. A., Fahrudin, T. M., Hindrayani, K. M., & Safitri, E. M. (2020). Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (LSTM). *Seminar Nasional Informatika, 2020*. <http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/4135>
- Rosyd, A., Purnamasari, A. I., & Ali, I. (2024). Penerapan Metode Long Short-Term Memory (LSTM) Dalam Memprediksi Harga Saham PT Bank Central Asia. *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Nomor 1). <https://ejournal.itn.ac.id/index.php/jati/article/view/8440>
- Setyawan, R. (2024). *Prediksi Harga Saham Berbasis Web Menggunakan Metode Long Short-Term Memory (LSTM) Dan Gated Recurrent Unit (GRU)* (Bachelor's Thesis, UIN Syarif Hidayatullah Jakarta). <https://repository.uinjkt.ac.id/dspace/handle/123456789/83911>
- Sumirat, L. P., dkk. (2023). *Dasar-Dasar Rekayasa Perangkat Lunak* [eBook]. <http://repository.unitomo.ac.id/3449/1/Ebook%20DasarDasar%20Rekayasa%20Perangkat%20Lunak.pdf>
- Suyudi, M. A. D., Djamal, E. C., & Maspupah, A. (2019). Prediksi Harga Saham Menggunakan Metode Recurrent Neural Network, *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. <https://journal.uui.ac.id/Snati/article/view/13398>
- Syahrul. (2023). *Penggunaan Metode Long Short-Term Memory Pada Peramalan Harga Saham PT Unilever Indonesia Tbk* (Skripsi, Universitas Lampung, Bandar Lampung). <http://digilib.unila.ac.id/72508/>
- Wardani, W. W. K. (2021). *Prediksi Harga Saham Syariah Menggunakan Metode Recurrent Neural Network-Long Short-Term Memory*. (Skripsi, Universitas Islam Negeri Sunan Ampel Surabaya). <https://digilib.uinsa.ac.id/49542/>
- Wathani, M. N., Kusrini, K., & Kusnawi, K. (2023). Prediksi Tren Pergerakan Harga Saham PT Bank Central Asia Tbk, Dengan Menggunakan Algoritma Long Shot Term Memory (LSTM). *Infotek: Jurnal Informatika Dan Teknologi*, 6(2), 502–512. <https://doi.org/10.29408/Jit.V6i2.19824>
- Wu, S. (2021). A Comparative Study of Stock Forecasts by LSTM and RNN Neural Networks. *Modern Economics & Management Forum*, 2(5). <https://doi.org/10.32629/memf.v2i5.497>
- Zhu, Y. (2020). Stock Price Prediction Using the RNN Model. *Journal Of Physics: Conference Series*, 1650(3). <https://doi.org/10.1088/1742-6596/1650/3/032103>

LAMPIRAN

1. Source Code

Install Library:

```
%pip install numpy pandas scikeras tensorflow matplotlib
mplfinance scikit-learn==1.5.0
```

Import Library:

```
# Import library berkaitan dengan seeding
import os # Interaksi sistem operasi
import numpy as np # Komputasi numerik
import random # Mengatur seed agar hasil konsisten
import tensorflow as tf # Deep learning

# Tetapkan nilai SEED
SEED = 42
# Atur Seed untuk Semua Library
os.environ['PYTHONHASHSEED'] = str(SEED) # Menjaga deterministik
pada Python
np.random.seed(SEED) # Untuk NumPy
random.seed(SEED) # Untuk Python `random`
tf.random.set_seed(SEED) # Untuk TensorFlow
import pandas as pd
import math
import sys
import sklearn
import logging
import matplotlib
import time
import logging
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler # Penskalaan data
from tensorflow.keras.models import Sequential # Model sequential
from tensorflow.keras.layers import SimpleRNN, LSTM, Dense,
Dropout # Layer model
from tensorflow.keras.optimizers import Adam # Optimizer
from tensorflow.keras.callbacks import EarlyStopping,
ModelCheckpoint # Callbacks
from sklearn.model_selection import RandomizedSearchCV
from scikeras.wrappers import KerasRegressor
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error,
root_mean_squared_error, mean_absolute_error

# Menampilkan dataset saham BBRI
df = pd.read_excel('datasets/BBRI_2010-2025.xlsx')

Pembagian Data : Training, Validation dan Testing (70%-15%-15%)
# Rasio pembagian data :
length_data = len(data) # Panjang data
```

```

# Tentukan jumlah data untuk masing-masing set
train_len = math.ceil(length_data * 0.7) # 70%
data_training
val_len = int(length_data * 0.15) # 15% data validation
test_len = length_data - (train_len + val_len) # 15% data
testing

Normalisasi Data:
# Normalisasi data dengan metode MinMaxScaler
sc = MinMaxScaler(feature_range=(0, 1))

# Pembentukan data training
train_set = data[0:train_len]
train_set

# Normalisasi data training
train_scaled_data = sc.fit_transform(train_set)
train_scaled_data[:10]

# Nilai time_step yang ditentukan yaitu 30 hari perdagangan
time_step = 30
# Membuat fungsi untuk pembentukan teknik sliding window
x_train = []
y_train = []
for i in range(time_step, len(train_scaled_data)):
    x_train.append(train_scaled_data[i-time_step:i, 0])
    y_train.append(train_scaled_data[i, 0])
    # Cek hanya untuk 2 iterasi pertama agar tidak terlalu
panjang
    if i < time_step + 2:
        print(f"x_train[{i - time_step}]:
{train_scaled_data[i-time_step:i, 0]}")
        print(f"y_train[{i - time_step}]:
{train_scaled_data[i, 0]}")
        print()

# Mengubah tipe data list menjadi numpy array 2D
x_train, y_train = np.array(x_train), np.array(y_train)
print("x_train before reshape :",x_train.shape)
print("y_train before reshape :",y_train.shape)

# Ubah bentuk data menjadi bentuk yang diterima oleh model ke
dalam bentuk 3D (Sample, Time_Step, Feature)
x_train = np.reshape(x_train, (x_train.shape[0],
x_train.shape[1], 1))
y_train = np.reshape(y_train, (y_train.shape[0],1))
print("x_train after reshape :",x_train.shape)
print("y_train after reshape :",y_train.shape)

# Pembentukan data validasi dan normalisasi data
val_set = data[train_len - time_step :(train_len + val_len),
:]
val_scaled_data = sc.transform(val_set)

```

```

# Buat fungsi pembentukan teknik sliding window seperti
sebelumnya
x_val = []
y_val = []
for i in range(time_step, len(val_scaled_data)):
    x_val.append(val_scaled_data[i - time_step:i, 0])
    y_val.append(val_scaled_data[i, 0])
x_val, y_val = np.array(x_val), np.array(y_val)
x_val = np.reshape(x_val, (x_val.shape[0], x_val.shape[1],
1))
y_val = np.reshape(y_val, (y_val.shape[0],1))

# Pembentukan data testing dan normalisasi data
test_set = data[(-test_len - time_step):, :]
test_scaled_data = sc.transform(test_set)
# Pembentukan data untuk melakukan prediksi pada data testing
x_test = []
y_test = df[['Close']].iloc[-test_len :, :].values
for i in range(time_step, len(test_scaled_data)):
    x_test.append(test_scaled_data [i-time_step:i, 0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0],
x_test.shape[1], 1))

Fungsi Membuat Hyperparameter RandomizedSearchCV Pada RNN:
# First RNN layer (harus return_sequences=True jika lebih
dari 1 layer)
def build_model_RNN(n_units=None, n_layers=None,
dropout_rate=None, learning_rate=None):
    tf.random.set_seed(SEED)
    model = Sequential()
    model.add(tf.keras.Input(shape=(x_train.shape[1], 1))) #
Definisikan input layer
    # Layer pertama harus mempertimbangkan jumlah layer yang
ada
    return_seq = True if n_layers > 1 else False
    model.add(SimpleRNN(units=n_units, activation="tanh",
return_sequences=return_seq))
    model.add(Dropout(dropout_rate)) # Dropout setelah layer
pertama

    # Menambahkan additional RNN layers sesuai jumlah layer
yang ditentukan
    for i in range(1, n_layers):
        return_seq = i < (n_layers - 1) # Hanya sebelum
layer terakhir pakai return_sequences=True
        model.add(SimpleRNN(units=n_units, activation="tanh",
return_sequences=return_seq))
        model.add(Dropout(dropout_rate)) # Dropout setelah
setiap RNN layer

    # Menambahkan output layer
    model.add(Dense(units=1))

```

```

        # Konfigurasi optimizer dan loss function serta compile
model
        optimizer = Adam(learning_rate=learning_rate)
        model.compile(optimizer=optimizer, loss='mse')
        return model
# Membungkus model dalam KerasRegressor dengan parameter yang
akan dituning
model_RNN = KerasRegressor(model=build_model_RNN, verbose=0)
# Menentukan ruang pencarian hyperparameter
param_distributions = {
    'model__n_units': [64, 128, 256],
    'model__n_layers': [1, 2],
    'model__dropout_rate': [0.2, 0.3],
    'model__learning_rate': [0.001, 0.0001],
    'epochs': [100],
    'batch_size': [64]
}

# Konfigurasi logging untuk menampilkan progress pencarian
logging.basicConfig(level=logging.INFO, format="%(asctime)s -
%(message)s")
# Inisialisasi RandomizedSearchCV
random_search = RandomizedSearchCV(
    estimator=model_RNN,
    param_distributions=param_distributions,
    n_iter=20,
    n_jobs=-1,
    scoring='neg_mean_squared_error',
    verbose=1,
    cv=5,
    random_state=SEED
)
# Fungsi early stopping
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=10,
    min_delta=0.00001,
    verbose=1,
    restore_best_weights=True
)
fit_params = {
    "callbacks": [early_stopping]
}
# Catat waktu mulai
start_time = time.time()
# Jalankan pencarian hyperparameter
logging.info("Memulai RandomizedSearchCV...")
random_search.fit(x_train, y_train, **fit_params)
logging.info("Pencarian hyperparameter selesai.")
# Hitung durasi
end_time = time.time()

```

```

elapsed_time = end_time - start_time
# Konversi ke jam, menit, dan detik
hours, rem = divmod(elapsed_time, 3600)
minutes, seconds = divmod(rem, 60)
logging.info(f"Durasi pencarian: {int(hours)} jam
{int(minutes)} menit {int(seconds)} detik")
# Menampilkan cv_results
cv_results = random_search.cv_results_
logging.info("\nCV Results:")
for mean_score, params, rank in
zip(cv_results['mean_test_score'], cv_results['params'],
cv_results['rank_test_score']):
    logging.info(f"Rank: {rank} - Params: {params} - Mean
Test Score: {mean_score:.6f}")
# Menampilkan hyperparameter terbaik
logging.info(f"\nBest Hyperparameters:
{random_search.best_params_}")
logging.info(f"Best Score: {random_search.best_score_}")

# Menggunakan model terbaik
best_params = best_params
best_model_RNN = build_model_RNN(
    n_units=best_params['model__n_units'],
    n_layers=best_params['model__n_layers'],
    dropout_rate=best_params['model__dropout_rate'],
    learning_rate=best_params['model__learning_rate']
)
# Menampilkan summary model
logging.info("Menampilkan Model Arsitektur RNN Summary:")
best_model_RNN.summary()
Training Model RNN:

# Melatih model dengan hyperparameter terbaik
history = best_model_RNN.fit(
    x_train, y_train,
    epochs=best_params['epochs'],
    batch_size=best_params['batch_size'],
    validation_data=(x_val, y_val),
    callbacks=[checkpoint_model],
)
# Plotting Training Loss dan Validation Loss
plt.figure(figsize=(6, 4))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation
Loss')
plt.title('Loss vs Epoch RNN', fontsize=14)
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

Prediksi RNN:
# Proses prediksi
y_pred_RNN = best_model_RNN.predict(x_test)

```

```

y_pred_RNN = sc.inverse_transform(y_pred_RNN)
# Prediksi data tesing
df_pred_RNN = df[-test_len:].copy()
df_pred_RNN['Prediction'] = np.round(y_pred_RNN, 2)
# Evaluasi dengan metrik
mse = mean_squared_error(y_test, y_pred_RNN)
rmse = root_mean_squared_error(y_test, y_pred_RNN)
mae = mean_absolute_error(y_test, y_pred_RNN)

# Evaluasi model
print(f'Testing Data Performance Metrics :')
print(f'=====')
print(f'Mean Squared Error (MSE): {mse:.2f}')
print(f'Root Mean Squared Error (RMSE): {rmse:.2f}')
print(f'Mean Absolute Error (MAE): {mae:.2f}\n')
training_loss = best_model_RNN.evaluate(x_train, y_train,
verbose=2)
validation_loss = best_model_RNN.evaluate(x_val, y_val,
verbose=2)
testing_loss = mean_squared_error(sc.transform(y_test),
sc.transform(y_pred_RNN))
print(f"\nTraining Loss: {training_loss:.6f}")
print(f"Validation Loss: {validation_loss:.6f}")
print(f"Testing Loss (In Normalized): {testing_loss:.6f}")

# Membuat data untuk visualisasi
train = df[:train_len]
valid = df[train_len:(train_len + val_len)]

# Visualisasi prediksi
plt.figure(figsize=(16, 8))
plt.title('Prediksi Harga Penutupan Saham BBRI (RNN)')
plt.plot(train.index, train['Close'], label='Training')
plt.plot(valid.index, valid['Close'], label='Validation')
plt.plot(df_pred_RNN.index, df_pred_RNN['Close'],
label='Testing')
plt.plot(df_pred_RNN.index, df_pred_RNN['Prediction'],
label='Prediksi', linestyle='--')
plt.xlabel('Date', fontsize=15)
plt.ylabel('Close Price', fontsize=15)
plt.legend()
plt.grid(True, linewidth=0.18, axis='y')
plt.show()

```


iThenticate®

Similarity Report ID: oid:29477:106663700

● **8% Overall Similarity**

Top sources found in the following databases:

- 8% Internet database
- Crossref database



- 2% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	digilib.unila.ac.id Internet	<1%
2	123dok.com Internet	<1%
3	digilib.uinsby.ac.id Internet	<1%
4	repositori.uma.ac.id Internet	<1%
5	journal.mediapublikasi.id Internet	<1%
6	hostjournals.com Internet	<1%
7	repository.unja.ac.id Internet	<1%
8	repository.uigm.ac.id Internet	<1%
9	docplayer.info Internet	<1%

[Sources overview](#)



UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate / Jalan Gedung PBSI, Medan 20223
Kampus II : Jalan Sei Serayu Nomor 70 A / Jalan Setia Budi Nomor 79 B, Medan 20112 Telepon : (061) 8225602, 8201994
Fax : (061) 8226331 HP : 0811 607 259 website: www.uma.ac.id Email : univ_medanarea@uma.ac.id

Nomor : 3154/FT/01.10/XI/2024
Lampiran : -
Hal : Pembimbing Tugas Akhir

14 November 2024

Yth. Pembimbing Tugas Akhir
RIZKI MULIONO S.Kom, M.Kom (Sebagai Pembimbing)
di Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : EVINDO AMANDA RIZA
NIM : 218160018
Jurusan : TEKNIK INFORMATIKA

Maka dengan hormat kami mengharapkan kesediaan saudara :

RIZKI MULIONO S.Kom, M.Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

**ANALISIS PERBANDINGAN RECURRENT NEURAL NETWORK DAN LONG SHORT
TERM MEMORY DALAM MEMPREDIKSI HARGA SAHAM**

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,



Dr Eng. Supriatno.ST, MT.





UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 75 /FT.6/01.10/III/2025

21 Maret 2025

Lamp

: -

Hal

: Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Wakil Rektor Bidang Mutu Sumber Daya Manusia dan Perekonomian

Jln. Kolam No.1

Di

Medan

Dengan hormat, kami mohon kesediaan bapak kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PRODI
1	Evindo Amanda Riza	218160018	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area**.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan Ilmiah dan Skripsi, yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

Analisis Perbandingan *Recurrent Neural Network* dan *Long Short Term Memory* dalam Memprediksi Harga Saham.

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya Surat ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.


Dr. Eng. Supriatno, ST, MT

Tembusan :

1. Ka. BPMPP
2. Mahasiswa
3. File



UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan 20122
Website: www.uma.ac.id E-Mail: univ_medanarea@uma.ac.id

Nomor : 546/UMA/B/01.7/III/2025
Lamp. : -
Hal : Izin Penelitian dan Pengambilan Data Tugas Akhir

26 Maret 2025

Kepada Yth.

Dekan Fakultas Teknik

Universitas Medan Area

di - Medan

Dengan hormat, sesuai dengan surat dari Dekan Fakultas Teknik Universitas Medan Area nomor : 75/FT.6/01.10/III/2025 tertanggal 21 Maret 2025, perihal izin penelitian dan pengambilan data tugas akhir di Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area oleh mahasiswa sebagai berikut :

Nama : Evindo Amanda Riza
NPM : 218160018
Program Studi : Teknik Informatika

Dengan Judul Penelitian "*Analisis Perbandingan Recurrent Neural Network dan Long Short Term Memory dalam Memprediksi Harga Saham*".

Pada Prinsipnya disetujui yang bersangkutan untuk melaksanakan izin penelitian dan pengambilan data tugas akhir di Fakultas Teknik Universitas Medan Area guna penyusunan Skripsi untuk kepentingan Ilmiah dan Akademik.

Demikian surat ini diterbitkan untuk dipergunakan sebagaimana mestinya.

Wakil Rektor Bidang Mutu SDM dan
Perekonomian,



Dr. Dedi Sahputra, S.Sos, MA

Tembusan :

1. Mahasiswa Ybs
2. File



UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan 20122
Website: www.uma.ac.id E-Mail: univ_medanarea@uma.ac.id

SURAT KETERANGAN Nomor : 775/UMA/B/01.7/IV/2025

Rector Universitas Medan Area dengan ini menerangkan bahwa :

Nama : Evindo Amanda Riza
No.Pokok Mahasiswa : 218160018
Program Studi : Teknik Informatika
Fakultas : Teknik Informatika

Benar telah selesai Pengambilan Data terhitung pada tanggal di Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area dengan Judul Skripsi "*Analisis Perbandingan Recurrent Neural Network dan Long Short Term Memory dalam Memprediksi Harga Saham*".

Dan kami harapkan Data tersebut kiranya dapat membantu yang bersangkutan dalam penyusunan skripsi dan dapat bermanfaat bagi mahasiswa khususnya Fakultas Teknik Universitas Medan Area.

Demikian surat ini diterbitkan untuk dapat digunakan seperlunya

Medan, 06 Mei 2025.

an Rector

Wakil Rector Bidang Mutu SD
Perekonomian,



Devi Sahputra, S.Sos, MA

CC :
- Arsip