

**PERBANDINGAN METODE *RECCURENT NEURAL NETWORK* (RNN) DAN
LONG SHORT TERM MEMORY (LSTM) UNTUK MEMPREDIKSI HARGA
SAHAM BANK TERBESAR DI INDONESIA**

LAPORAN TUGAS AKHIR

**Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Strata 1 (S1) Pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang**



Disusun Oleh :

Nailatul Ftriani A.J

32602100005

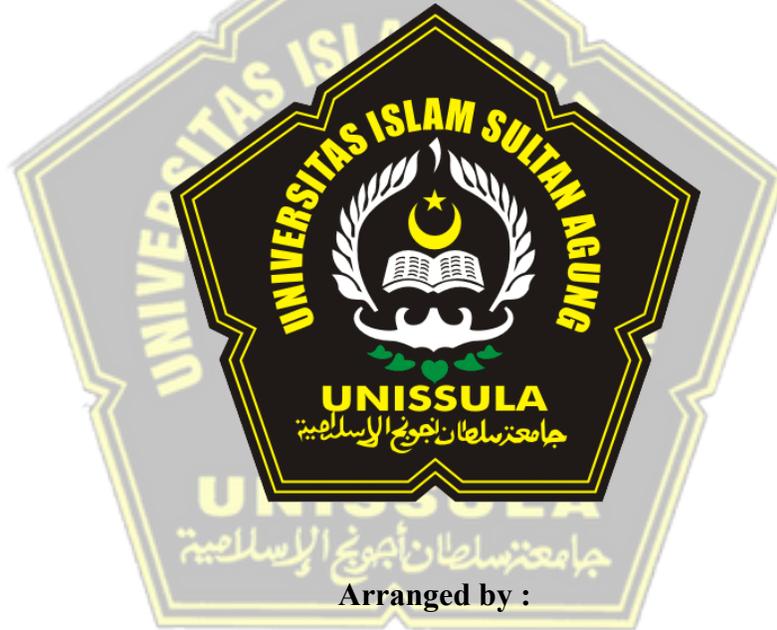
**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM AGUNG SEMARANG**

2025

FINAL PROJECT

**COMPARISON OF RECURRENT NEURAL NETWORK (RNN) AND LONG
SHORT TERM MEMORY (LSTM) METHODS TO PREDICTE STOCK
PRICES OF THE LARGEST BANKS IN INDONESIA**

*Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatic Engineering Department of Industrial Technology Faculty
Sultan Agung Islamic University*



Arranged by :

NAILATUL FITRIANI A.J

32602100005

**MAJORING OF INFORMATIC ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SULTAN AGUNG ISLAMIC UNIVERSITY**

2025

LEMBAR PENGESAHAN
TUGAS AKHIR

PERBANDINGAN METODE RECURRENT NEURAL NETWORK (RNN)
DAN LONG SHORT TERM MEMORY (LSTM) UNTUK MEMPREDIKSI
HARGA SAHAM BANK TERBESAR DI INDONESIA

NAILATUL FITRIANI A.J
NIM. 32602100005

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 29 Februari 2025

TIM PENGUJI UJIAN SARJANA :

Andi Riansyah, S.T., M.Kom

NIK. 210616053

(Ketua Penguji)

 04-03-2025

Bagus Satrio Waluyo Poetro, S.Kom., M.Cs

NIK. 210616051

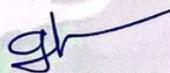
(Anggota Penguji)

 27-02-2025

Ghufron, S.T., M.Kom

NIK. 210622056

(Pembimbing)

 04-03-2025

Semarang, 04 Maret 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung


Moch Taufik
NIK. 216040034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Nailatul Fitriani A.J

NIM : 32602100005

Judul Tugas Akhir : Perbandingan Metode *Reccurent Neural Network* (RNN) dan *Long Short Term Memory* (LSTM) untuk Memprediksi Harga Saham Bank Terbesar di Indonesia

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 04 maret 2025

Yang Menyatakan,



Nailatul Fitriani A.J

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan tugas akhir dengan judul “Perbandingan Metode *Reccurent Neural Network* (RNN) dan *Long Short Term Memory* (LSTM) untuk Memprediksi Harga Saham Bank Terbesar Di Indonesia”

Tugas akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis. Oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, S.T., M.T., IPU., ASEAN. Eng
3. Dosen pembimbing penulis Bapak Ghufron, S.T., M.Kom yang telah meluangkan waktu dan memberi banyak nasehat dan saran
4. Orang tua penulis yang memberikan dukungan dan doa agar tugas akhir ini berjalan dengan lancar
5. Dan untuk semua pihak yang tidak bisa saya sebutkan satu persatu

Dengan segala kerendahan hati, penulis menyadari masih banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmy pengetahuan dalam menyusun laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 20 februari 2025

Nailatul Fitriani A.J

DAFTAR ISI

COVER	i
LEMBAR PENGESAHAN TUGAS AKHIR	iii
LEMBAR KEASLIAN TUGAS AKHIR	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
ABSTRAK	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.3 <i>Reccurent Neural Network</i>	8
2.4 <i>Long Short Term Memory</i>	9
BAB III METODOLOGI PENELITIAN	13
3.1 Metode Penelitian.....	13
3.2 Studi Literatur	14
3.3 Pengumpulan Data	14
3.4 <i>Preprocessing Data</i>	14
3.4.1 <i>Data Cleaning</i>	14

3.4.2	Normalisasi Data.....	14
3.4.3	Pembagian data	14
3.5	Perancangan Model.....	15
3.6	Denormalisasi.....	16
3.7	Evaluasi Model	16
3.8	<i>Deployment Model</i>	16
3.9	Analisis Kebutuhan	17
3.10	Perancangan sistem	19
BAB IV	HASIL DAN ANALISIS PENELITIAN	20
4.1	Hasil	20
4.1.1	Dashboard Streamlit.....	20
4.1.2	Hasil prediksi	21
4.2	Analisis.....	29
4.2.1	Persiapan Data.....	29
4.2.2	Data preprocessing	32
4.2.3	Perancangan model RNN dan LSTM	36
4.2.4	Denormalisasi.....	41
4.2.5	Evaluasi model.....	41
4.2.6	Hasil Perbandingan model	43
BAB V	KESIMPULAN DAN SARAN	47
5.1	Kesimpulan	47
5.2	Saran.....	47
	DAFTAR PUSTAKA.....	48
	LAMPIRAN.....	51



DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi <i>deep learning</i>	7
Gambar 2. 2 arsitektur RNN	8
Gambar 2. 3 arsitektur LSTM.....	9
Gambar 3. 1 Flowchart Metode Penelitian	13
Gambar 3. 2 <i>Flowchart</i> Perancangan sistem.....	19
Gambar 4. 1 Tampilan Utama Sistem	20
Gambar 4. 2 Tampilan Utama Sistem untuk Data Prediksi Bank BRI	21
Gambar 4. 3 Tampilan Utama Sistem untuk Data Prediksi Bank BRI	21
Gambar 4. 4 Tampilan Utama Sistem untuk Data Prediksi Bank BRI	22
Gambar 4. 5 Tampilan Utama Sistem untuk Data Prediksi Bank BRI	22
Gambar 4. 6 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri	23
Gambar 4. 7 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri	23
Gambar 4. 8 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri	24
Gambar 4. 9 Tampilan Utama Sistem Untuk Data Prediksi Bank Mandiri	24
Gambar 4. 10 Tampilan Utama Sistem untuk Data Prediksi Bank BNI	25
Gambar 4. 11 Tampilan Utama Sistem untuk Data Prediksi Bank BNI	25
Gambar 4. 12 Tampilan Utama Sistem untuk Data Prediksi Bank BNI	26
Gambar 4. 13 Tampilan Utama Sistem untuk Data Prediksi Bank BNI	26
Gambar 4. 14 Tampilan Utama Sistem untuk Data Prediksi Bank BCA.....	27
Gambar 4. 15 Tampilan Utama Sistem untuk Data Prediksi Bank BCA.....	27
Gambar 4. 16 Tampilan Utama Sistem untuk Data Prediksi Bank BCA.....	28
Gambar 4. 17 Tampilan Utama Sistem untuk Data Prediksi Bank BCA.....	28
Gambar 4. 18 Grafik dari Masing-Masing Bank	31
Gambar 4. 19 Cek <i>missing value</i>	32
Gambar 4. 20 heatmap hasil korelasi fitur	33
Gambar 4. 21 Grafik Perbandingan Model RNN dan LSTM Bank Mandiri.....	43
Gambar 4. 22 Grafik Perbandingan Model RNN dan LSTM Bank BRI	44
Gambar 4. 23 Grafik Perbandingan Model RNN dan LSTM Bank BRI	45

Gambar 4. 24 Grafik Perbandingan Model RNN dan LSTM Bank BRI..... 45



DAFTAR TABEL

Tabel 3. 1 Pengujian Parameter.....	16
Tabel 3. 2 Skala nilai akurasi MAPE	16



ABSTRAK

Prediksi harga saham merupakan salah satu tantangan utama dalam bidang keuangan, mengingat volatilitas pasar yang tinggi dan kompleksitas pola pergerakan harga. Penelitian ini membandingkan performa dua metode berbasis deep learning, yaitu *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM), dalam memprediksi harga saham empat bank terbesar di Indonesia, yaitu Bank Mandiri, BRI, BNI, dan BCA. Evaluasi dilakukan dengan mengukur akurasi prediksi menggunakan *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE). Hasil penelitian menunjukkan bahwa model LSTM memiliki performa yang lebih baik dibandingkan RNN, dengan nilai RMSE dan MAPE yang lebih rendah. Visualisasi prediksi juga menunjukkan bahwa LSTM lebih mampu menangkap pola tren harga saham dan lebih stabil dalam menghadapi fluktuasi pasar dibandingkan RNN. Dengan demikian, LSTM lebih direkomendasikan sebagai metode prediksi harga saham karena kemampuannya dalam menangani dependensi jangka panjang dan pola data yang kompleks. Kata kunci: Prediksi harga saham, RNN, LSTM

ABSTRACT

Stock price prediction is one of the main challenges in finance, given the high market volatility and complexity of price movement patterns. This study compares the performance of two deep learning-based methods, namely Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), in predicting the stock prices of four largest banks in Indonesia, namely Bank Mandiri, BRI, BNI, and BCA. Evaluation is done by measuring prediction accuracy using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The results showed that the LSTM model performed better than RNN, with lower RMSE and MAPE values. The prediction visualization also shows that LSTM is better able to capture stock price trend patterns and is more stable in the face of market fluctuations than RNN. Thus, LSTM is more recommended as a stock price prediction method due to its ability to handle long-term dependencies and complex data patterns.

Keyword : Stock Price Prediction, RNN, LSTM

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan digitalisasi telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk di dunia finansial. Salah satu sektor yang paling terpengaruh oleh kemajuan teknologi ini adalah pasar saham. Saham merupakan bentuk kepemilikan modal dalam sebuah perusahaan atau badan usaha yang memperoleh dana melalui pasar modal. Pasar saham selalu menjadi daya tarik bagi investor, baik individu maupun institusi keuangan besar, karena potensinya dalam menghasilkan keuntungan dan mencerminkan dinamika ekonomi global. Menurut data Bursa Efek Indonesia (BEI) per September 2024, jumlah single investor identification (SID) tercatat mencapai sekitar 14 juta, mengalami peningkatan dari 12,2 juta pada akhir tahun 2023. Artinya, dalam waktu kurang dari 10 bulan, terdapat penambahan sekitar 1,8 juta investor baru. Hal ini menunjukkan minat masyarakat terhadap pasar modal terus meningkat, didorong oleh literasi keuangan yang semakin baik serta kemudahan akses investasi melalui platform digital.

Ada beberapa sektor industri saham yang tersedia di pasar modal, namun sektor perbankan tetap menjadi salah satu yang paling menarik bagi para investor. Hal ini dikarenakan stabilitasnya dan dampak yang signifikan terhadap kinerja pasar secara keseluruhan, terutama pada bank-bank besar di Indonesia seperti BCA, BRI, BNI, dan Mandiri. Akan tetapi, pasar saham dikenal karena sifatnya yang fluktuatif dan dipengaruhi oleh berbagai faktor, termasuk kondisi ekonomi makro, kebijakan moneter, perubahan regulasi, dan faktor geopolitik.

Salah satu tantangan utama yang dihadapi investor adalah memprediksi pergerakan harga saham di masa mendatang. Meskipun pasar saham menawarkan peluang keuntungan yang besar, tetapi resiko kerugian dalam waktu singkat juga tidak dapat dihindari (Rosyd dkk., 2024). Oleh karena itu, untuk meminimalkan resiko

kerugian, dibutuhkan pemantauan yang teliti terhadap fluktuasi harga saham. Fluktuasi harga saham yang sangat cepat dan tidak terduga menjadi tantangan utama bagi para investor. Oleh karena itu kemampuan untuk memprediksi harga saham dengan akurat menjadi aspek yang sangat penting.

Dalam beberapa dekade terakhir, telah berkembang berbagai pendekatan untuk memprediksi harga saham, mulai dari analisis teknikal dan fundamental hingga penerapan metode statistik dan kecerdasan buatan untuk meningkatkan akurasi prediksi (Murphy, 1999). Salah satu cabang dari kecerdasan buatan yaitu *machine learning* merupakan metode yang efektif untuk memprediksi harga saham. *Machine learning* merupakan bagian dari kecerdasan buatan yang bertujuan meningkatkan pemahaman dan performa (provost & Fawcett, 2013). *Reccurent Neural Network* (RNN) dan *Long Sort Term Memory* (LSTM) merupakan salah satu algoritma *deep learning* yang digunakan dalam melakukan prediksi berbasis data berurutan, seperti data saham dari waktu ke waktu. RNN merupakan salah satu algoritma *deep learning* yang memiliki tingkat keakuratan yang tinggi dalam memprediksi data berbentuk deret waktu (Cao dkk, 2013). Karena dapat memanfaatkan informasi yang telah disimpan sebelumnya meskipun urutannya bervariasi (Sari & Septiano, 2023). Selain itu RNN juga mengembangkan sebuah algoritma yaitu *Long Short Term Memory* (LSTM). LSTM merupakan salah satu pendekatan yang banyak digunakan dalam beberapa tahun terakhir. LSTM dikenal karena kemampuannya dalam menangani data urutan waktu (*time series*) dan mengatasi masalah *vanishing gradient* yang sering ditemui oleh jaringan saraf tradisional (Hochreiter & Schmidhuber, 1997).

Dalam konteks prediksi harga saham bank terbesar di Indonesia, perbandingan antara metode RNN dan LSTM dapat memberikan potensi untuk menghasilkan prediksi yang lebih akurat berdasarkan data historis dan pemahaman yang lebih dalam mengenai kelebihan dan kekurangan pada masing-masing metode. Melalui penelitian ini, diharapkan dapat diperoleh solusi yang lebih efektif dalam meramalkan pergerakan harga saham, yang dapat membantu dalam pengambilan keputusan investasi yang lebih baik dan lebih akurat.

1.2 Perumusan Masalah

Rumusan masalah dari penelitian ini yaitu sebagai berikut :

1. Bagaimana implementasi algoritma RNN dan LSTM untuk prediksi harga saham bank terbesar di Indonesia?
2. Bagaimana akurasi prediksi yang dihasilkan oleh RNN dan LSTM dalam prediksi harga saham bank terbesar di Indonesia?

1.3 Pembatasan Masalah

Adapun batasan masalah dari penelitian ini yaitu sebagai berikut :

1. Data yang digunakan dalam penelitian ini adalah data historis harga penutupan saham harian dari beberapa bank besar di Indonesia, seperti Bank Mandiri, Bank Central Asia (BCA), dan Bank Rakyat Indonesia (BRI), Bank Negara Indonesia (BNI).
2. Penelitian ini hanya membandingkan dua jenis algoritma, yaitu RNN dan LSTM

1.4 Tujuan

Tujuan dari Tugas akhir ini yaitu mengimplementasikan algoritma RNN dan LSTM dan membandingkan performa dan tingkat akurasi dari metode RNN dan LSTM dalam memprediksi harga saham bank terbesar di Indonesia

1.5 Manfaat

Manfaat dari tugas kakhir ini adalah sebagai beirkut :

1. Penelitian ini dapat menambah referensi tentang implementasi metode RNN dan LSTM dalam prediksi harga saham
2. Penelitian ini dapat membantu membandingkan efektifitas metode RNN dan LSTM dalam memprediksi data deret waktu
3. Hasil penelitian ini dapat membantu investor membuat keputusan investasi berdasarlan harga saham yang lebih akurat

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, batasan masalah tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang berkaitan untuk membantu memahami konsep algoritma RNN dan LSTM untuk melengkapi penelitian ini.

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evaluasi hasil prediksi.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari hasil prediksi dari kedua algoritma kemudian dari hasil tersebut dibandingkan mana yang lebih baik dalam memprediksi.

BAB V : SARAN DAN KESIMPULAN

Pada BAB V merangkum keseluruhan proses penelitian dari awal sampai akhir

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Saham menjadi salah satu opsi investasi yang menarik karena menawarkan potensi keuntungan lebih tinggi dibandingkan jenis usaha lainnya. Namun, untuk meminimalkan risiko kerugian, diperlukan ketelitian dalam memantau pergerakan saham serta perkembangan pasar modal sebagai indikator penting. Berkat kemajuan teknologi dalam bidang prediksi dan pembelajaran mesin, proses identifikasi serta peramalan harga saham kini dapat dilakukan secara otomatis (Yendriani dkk., 2015).

Prediksi saham merupakan upaya memperkirakan pergerakan harga saham di masa mendatang dengan menggunakan data historis. Dalam dunia keuangan, tujuan prediksi saham adalah memberikan wawasan kepada investor mengenai tren pasar sehingga mereka dapat membuat keputusan investasi yang lebih optimal. Prediksi ini memiliki peran penting karena fluktuasi harga saham mencerminkan berbagai aspek, seperti ekonomi, sentimen pasar, dan faktor eksternal lainnya (Sitanggang & Hidayat, 2013).

Prediksi harga saham bank dilakukan dengan mengimplementasikan kecerdasan buatan dengan membandingkan dua metode, yaitu RNN dan LSTM yang mana metode tersebut dapat melakukan prediksi berbasis data berurutan. Untuk mendukung penelitian ini, terdapat beberapa penelitian yang berhubungan langsung maupun tidak langsung yaitu diantaranya :

Penelitian yang dilakukan oleh (Kiramy dkk., 2024) mengenai Analisis performa algoritma RNN dan LSTM dalam memprediksi jumlah jamaah umrah di PT. Hajar Aswad menunjukkan bahwa LSTM menghasilkan prediksi yang lebih akurat dibandingkan RNN.

Penelitian yang dilakukan oleh (Homepage dkk., 2024) mengenai Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan

Keuangan Indonesia menunjukkan bahwa model LSTM cenderung menghasilkan kinerja yang lebih stabil dibanding dengan RNN untuk skenario yang sudah dilakukan. LSTM mampu mengatasi *vanishing gradient* yang sering dihadapi oleh RNN, sehingga memberikan hasil yang lebih baik dalam prediksi data deret waktu.

Penelitian lain yang dilakukan oleh (Selle dkk., 2022) mengenai perbandingan Prediksi Penggunaan Listrik Dengan Menggunakan Metode *Long Short Term Memory* (LSTM) Dan *Recurrent Neural Network* (RNN) menunjukkan bahwa LSTM memiliki performa lebih unggul dan mampu mengurangi nilai error, terutama ketika prediksi pada deret waktu yang lebih panjang.

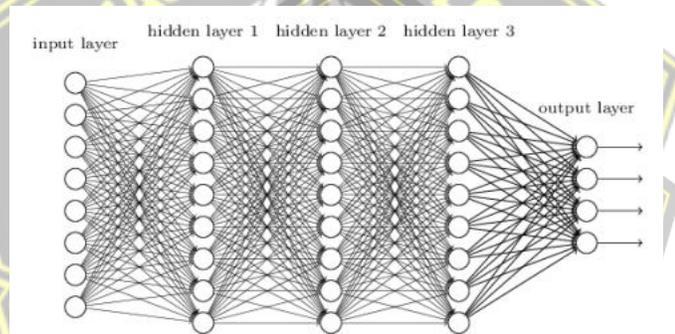
Penelitian lain yang dilakukan oleh (Roni Merdiansah dkk., 2024) mengenai Analisis kinerja model RNN, LSTM, dan BLSTM dalam memprediksi jumlah gempa bulanan di Indonesia menunjukkan bahwa penggunaan unit LSTM dengan konfigurasi Bidirectional (BLSTM), yang masih termasuk dalam keluarga RNN, dapat menghasilkan prediksi yang akurat terhadap rata-rata kejadian gempa bumi setiap bulan di Indonesia.

Penelitian lainnya yang dilakukan oleh (Budiprasetyo Gunawan dkk., 2022) mengenai Peramalan harga saham blue chip dengan algoritma LSTM menghasilkan prediksi harga penutupan saham yang telah disesuaikan menggunakan kecerdasan buatan berbasis algoritma RNN. Metode LSTM menunjukkan tingkat akurasi yang tinggi, dengan nilai rata-rata MAPE sebesar 2,6%, yang mendekati harga sebenarnya. Hasil ini membuktikan bahwa LSTM dapat diandalkan dalam pengambilan keputusan investasi berdasarkan prediksi yang akurat dan mendekati nilai asli.

Penelitian lainnya yang dilakukan oleh (Nilsen, 2022) mengenai perbandingan Perbandingan Model RNN, Model LSTM, dan Model GRU dalam Memprediksi Harga Saham-Saham LQ45 menunjukkan bahwa model LSTM memiliki hasil yang lebih baik dalam memprediksi harga saham diikuti dengan model GRU

2.2 Dasar Teori

Deep Learning merupakan sebuah cabang dari *machine learning* yang berfungsi untuk mempelajari data yang tersedia menggunakan algoritma tertentu (Schmidhuber, 2015). Perbedaan utama antara *Deep Learning* dan *neural network* terletak pada jumlah *hidden layer*. *Deep Learning* memiliki lebih banyak *hidden layer* dibandingkan *neural network* biasa. Sebuah jaringan dikategorikan sebagai "*deep learning*" jika memiliki lebih dari tiga lapisan (termasuk lapisan *input* dan *output*). Secara teknis, *Deep Learning* dapat didefinisikan sebagai bentuk *machine learning* dengan lebih dari satu *hidden layer*. Sebagai contoh, pada Gambar 2.1, terlihat ilustrasi *Deep Learning* yang memiliki 4 lapisan, dengan jumlah node yang bervariasi di setiap lapisan (Rizki dkk., 2020)



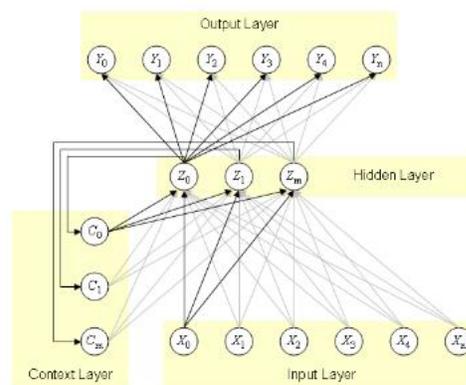
Gambar 2. 1 Ilustrasi *deep learning*

Deep learning banyak diterapkan dalam berbagai proyek besar maupun kecil. Jenis algoritma yang terdapat pada *deep learning* yaitu antara *Lain Convolutional Neural Network (CNN)*, *Reccurent Neura Network (RNN)*, *Long Short Term Memory (LSTM)*, dan lain sebagainya. Beberapa penerapannya digunakan dalam produk sehari-hari yaitu deteksi gambar, pengenalan suara, deteksi anomali, mesin rekomendasi, prediksi, hingga pemrosesan bahasa alami (NLP) (Erwin dkk., 2023). Maka dari itu, dalam penelitian ini menggunakan RNN dan LSTM untuk memprediksi saham bank terbesar di Indonesia

2.3 Recurrent Neural Network

Recurrent Neural Network (RNN) merupakan salah satu jenis jaringan syaraf tiruan yang mampu untuk memproses data berurutan (*sequential*) dengan konteks temporal sehingga dapat digunakan pada data *time series* (Luchia dkk., 2024). Peran yang signifikan dari RNN adalah dalam memori *input* dan menghasilkan *output* yang tepat sesuai dengan konteksnya. Kemampuan RNN untuk menyimpan informasi dari *input* dan mengeluarkan *output* yang sesuai dengan konteksnya adalah aspek yang krusial. Memori internal menjadi fokus penting dalam RNN karena memungkinkannya untuk memprediksi kejadian selanjutnya. Karena alasan tersebut, RNN sangat sesuai untuk aplikasi pada deret waktu, audio, video, mesin pencarian, teks dan juga mesin kasir (Rezk dkk., 2020).

Secara umum, pada RNN keluaran dari unit-unit di dalam *hidden layer* akan dikirimkan kembali ke dalam jaringan (*feedback*). Proses ini memungkinkan RNN untuk menggunakan informasi pada proses sebelumnya sebagai bagian dalam pemrosesan data saat ini. Mekanisme pada model RNN menjadikannya cocok digunakan untuk pemodelan yang melibatkan data sekuensial seperti peramalan data deret waktu. Namun demikian, pada model RNN terdapat masalah *vanishing gradient* yang menyebabkan RNN hanya memiliki memori jangka pendek sehingga kemampuannya dalam memprediksi jangka panjang menjadi terbatas (Bianchi dkk., 2017).



Gambar 2. 2 arsitektur RNN

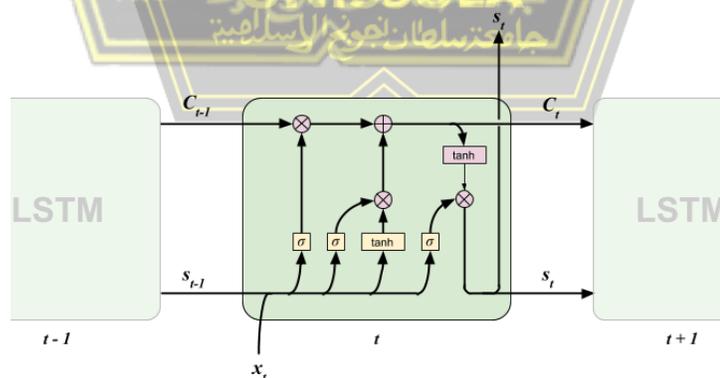
Gambar 2.2 menjelaskan arsitektur RNN yang terdiri dari tiga lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*. Perbedaan utama terletak pada cara proses komputasi berlangsung di *hidden layer*, yang sering disebut sebagai isi dari *memory cell*. Langkah pertama pada model RNN disajikan pada Persamaan (1) yaitu menghitung *hidden state* baru h_t melalui fungsi tanh dari input saat ini X_t dan *hidden state* sebelumnya H_{t-1} . Selanjutnya, untuk *output* numerik maka nilai keluaran dari model RNN diperoleh langsung melalui Persamaan (2) yaitu total nilai *hidden state* baru berdasarkan bobot masing-masing unitnya.

$$h_t = \tau(W_x X_t + W_h h_{t-1}) + b_h \quad (1)$$

$$O_t = W_o h_t + b_o \quad (2)$$

2.4 Long Short Term Memory

Long Short Term Memory (LSTM) adalah salah satu variasi dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi masalah pada *hidden layer*. Algoritma LSTM berfokus pada penggabungan kontrol non-linear dan hubungan antar data dalam sel RNN (Gao dkk, 2017), sekaligus memastikan gradien fungsi tujuan tetap terjaga agar sinyal tidak hilang. LSTM dirancang untuk mengatasi permasalahan *vanishing gradient*, yaitu kondisi di mana gradien bernilai nol atau mendekati nol, dengan menggunakan mekanisme gate (Pulver & Lyu, 2017).



Gambar 2.3 arsitektur LSTM

Berdasarkan gambar 2.3 menunjukkan struktur dan aliran dari memory cell pada LSTM, yang berfungsi mirip dengan *neuron* dalam jaringan syaraf tiruan lainnya. LSTM terdiri dari *memory cell* dan *gate unit* yang memiliki peran penting dalam pengaturan memori setiap *neuron*. *Memory cell* ini menjalankan empat proses utama dengan fungsi aktivasi sigmoid yang digunakan atau tanh untuk setiap input yang diterima oleh *neuron* yang dikenal sebagai *gate unit*. *Gate unit* ini memiliki tiga komponen *input gate*, *forget gate*, dan *output gate*. Selain itu, terdapat *cell gate* yang bertanggung jawab untuk memperbarui nilai dalam cell dengan nilai baru yang relevan (Aji Riyantoko dkk., 2020). Berikut ini merupakan struktur dari arsitektur LSTM:

1. *Forget Gate*

Lapisan pertama dalam LSTM disebut sebagai *forget gate* yang bertujuan untuk memutuskan informasi mana yang kurang relevan dan tidak dibutuhkan oleh sebuah sistem, akan dilupakan atau dihapus menggunakan fungsi sigmoid. Data x_t merupakan input data (vektor input x dalam timestep t) dan h_{t-1} merupakan vektor *hidden state* dalam *time step* sebelumnya $t - 1$. Perhitungan nilai *forget gate* adalah sebagai berikut:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

dimana:

f_t : *forget gate*

σ : fungsi sigmoid

W_f : nilai *weight* untuk *forget gate*

h_{t-1} : nilai *output* dalam *timestep* $t - 1$

x_t : nilai *output* dalam *timestep* t

b_f : nilai bias *forget gate*

2. *Input Gate*

Lapisan kedua disebut *input gate* yang bertujuan untuk menseleksi dan menentukan informasi berguna ke bagian cell state menggunakan fungsi sigmoid. *Input gate* mempunyai 2 fungsi, fungsi pertama adalah menambahkan informasi

sebelumnya yang telah diseleksi melewati forget gate. Berikut merupakan perhitungan nilai input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

dimana:

i_t : *input gate*

σ : fungsi sigmoid

W_i : nilai *weight* untuk *input gate*

h_{t-1} : nilai *output* dalam *timestep* $t - 1$

x_t : nilai *input* dalam *timestep* t

b_i : nilai bias *input gate*

Input gate akan membentuk kandidat vektor baru \tilde{C}_t menggunakan fungsi aktivasi *tanh* agar dapat mengontrol berapa banyak informasi baru ditambahkan.

Perhitungan yang dilakukan adalah sebagai berikut:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

dimana:

\tilde{C}_t : nilai baru yang ditambahkan pada *cell state*

tanh : fungsi tanh

W_c : nilai *weight* untuk *cell state*

h_{t-1} : nilai *output* dalam *timestep* $t - 1$

x_t : nilai *input* dalam *timestep* t

b_c : nilai bias *cell state*

Selanjutnya yaitu memperbarui nilai *cell state* lama C_{t-1} menjadi *cell state* baru C_t menggunakan fungsi berikut:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

dimana:

C_t : nilai *cell state* baru

C_{t-1} : *cell state* sebelumnya $t - 1$

\tilde{C}_t : nilai baru yang ditambahkan pada *cell state*

f_t : *forget gate*

i_t : *input gate*

3. Output Gate

Lapisan terakhir LSTM adalah output gate yang bertujuan untuk menjalankan aktivasi sigmoid untuk menghasilkan nilai output pada hidden state dan menempatkan cell state pada tanh. Hasil aktivasi nilai output sigmoid dan output tanh selanjutnya akan dilakukan perkalian sebelum ke tahap berikutnya. Berikut merupakan perhitungannya:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

dimana:

o_t : *output gate*

σ : fungsi sigmoid

W_o : nilai *weight* untuk *output gate*

h_{t-1} : nilai *output* dalam *timestep* $t - 1$

x_t : nilai *input* dalam *timestep* t

b_o : nilai bias *output gate*

Nilai output terakhir dapat didefinisikan sebagai berikut:

$$h_t = o_t * \tanh(C_t) \quad (8)$$

dimana:

h_t : nilai *output* ke- t

o_t : *output gate*

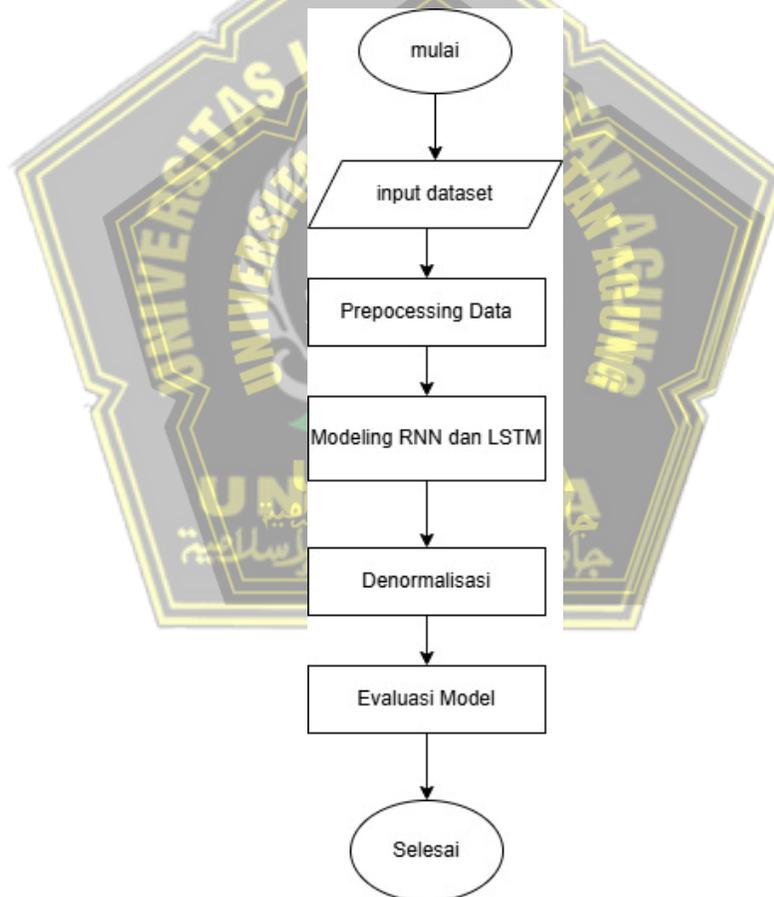
\tanh : fungsi tanh

C_t : *cell state*

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode RNN dan LSTM untuk memprediksi harga saham bank terbesar di Indonesia, serta membandingkan metode mana yang memberikan hasil terbaik dalam memprediksi harga saham. Berikut merupakan tahapan yang akan dilakukan pada penelitian ini yang dapat ditemukan pada Gambar 3.1



Gambar 3. 1 Flowchart Metode Penelitian

3.2 Studi Literatur

Dalam penelitian ini akan dilakukan tinjauan terhadap beberapa makalah, jurnal, dan skripsi terdahulu yang akan diulas, tujuan dari tinjauan ini adalah untuk mempelajari teori dibalik konsep predidiksi harga saham, model RNN dan LSTM.

3.3 Pengumpulan Data

Sumber data yang digunakan dalam penelitian ini berupa data harga penutupan saham harian dari beberapa bank terbesar di Indonesia (seperti Bank Mandiri, BCA, BNI, dan BRI). Data diambil dari sumber resmi seperti *Yahoo Finance* dengan periode data menggunakan data historis saham dalam rentang waktu 10 tahun terakhir yaitu dari tahun 2010 hingga 2024 untuk memberikan ruang analisis tren jangka panjang.

3.4 Preprocessing Data

3.4.1 Data Cleaning

Data cleaning merupakan proses dalam pengolahan data yang bertujuan untuk memastikan kualitas dan konsistensi data dengan memperbaiki atau menghapus data yang tidak lengkap atau duplikat, mengganti tipe data yang tidak sesuai, serta mengubah *index*.

3.4.2 Normalisasi Data

Perbedaan rentang harga yang signifikan dapat menyebabkan tingginya tingkat error pada model yang dibangun sehingga perlu dilakukan normalisasi untuk mengubah nilai harga penutupan saham yang sebenarnya menjadi nilai dengan rentang 1 hingga 0 menggunakan *min-max normalization* (Wiranda & Sadikin, 2019).

3.4.3 Pembagian data

Pembagian data dilakukan dengan membagi dataset menjadi data latih dan data uji dengan rasio 80:20. Karena data yang digunakan merupakan data time series, pemisahan dilakukan secara berurutan tanpa pengacakan dan hanya dilakukan sekali.

Dengan demikian, 80% data pertama digunakan sebagai data latih, sementara 20% data terakhir digunakan sebagai data uji. Data latih berfungsi untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi performa atau kinerja model.

3.5 Perancangan Model

Dalam penelitian ini model RNN dan LSTM menggunakan dua lapisan Simple RNN dan LSTM masing-masing 128 dan 64 neuron. Lapisan pertama menggunakan `return_sequences=True` agar dapat meneruskan informasi ke lapisan berikutnya, sedangkan lapisan kedua menggunakan `return_sequences=False` karena *output* dari lapisan ini akan masuk ke lapisan padat (*fully connected layers*). Dua lapisan *Dense* kemudian ditambahkan untuk menghasilkan prediksi akhir, dengan satu *neuron* pada lapisan *output* yang bertugas memprediksi harga saham. Proses kompilasi model dilakukan dengan menggunakan *optimizer Adam*, yang dikenal sebagai salah satu algoritma optimasi terbaik dalam pembelajaran mendalam karena mampu menyesuaikan *learning rate* secara adaptif. Sementara itu, fungsi kehilangan (*loss function*) yang digunakan adalah *Mean Squared Error* (MSE), yang umum digunakan dalam permasalahan regresi karena menghitung rata-rata kesalahan kuadrat antara prediksi dan nilai aktual. Kemudian model dilatih menggunakan *epoch* dan *batch size*. Jumlah *epoch* dapat disesuaikan secara bebas, tanpa terikat pada jumlah *input*. Bergantung pada arsitektur model, pemilihan jumlah unit dan *epoch* yang terlalu rendah bisa menyebabkan *underfitting* (model tidak dapat menangkap hubungan yang berarti antara *input* dan *output*), sementara jumlah unit dan *epoch* yang terlalu tinggi dapat menyebabkan *overfitting* (model terlalu menyesuaikan diri dengan data pelatihan, sehingga performa model menurun pada data uji). (I Nyoman Cerdas Janastu & Dhoriva Urwatul Wutsqa, 2024). Maka dari itu akan dilakukan beberapa uji coba menggunakan variasi *epoch* dan *batch size* untuk mendapatkan hasil akurasi yang baik.

Tabel 3. 1 Pengujian Parameter

Epoch	Batch size
32	30
32	50
32	100
64	30
64	50
64	100

3.6 Denormalisasi

Denormalisasi merupakan Proses ini dilakukan untuk mengubah data data prediksi yang awalnya berada dalam rentang interval 0-1 kembali ke nilai aktual harga saham (Bode, 2017).

3.7 Evaluasi Model

Evaluasi model bertujuan untuk mengukur akurasi dari model yang telah dikembangkan. Dalam penelitian ini evaluasi model menggunakan *Mean Absolute Percentage Error* (MAPE). MAPE adalah ukuran persentase kesalahan yang mengukur perbedaan antara nilai aktual dan nilai prediksi (Maricar dkk., 2017). Berikut skala nilai akurasi prediksi berdasarkan nilai MAPE yang dapat dilihat pada tabel 2

Tabel 3. 2 Skala nilai akurasi MAPE

Skala MAPE	Interpretesis
<10%	Hasil prediksi sangat tepat
10-20%	Hasil prediksi tepat
20-25%	Hasil prediksi cukup
>50%	Hasil prediksi tidak akurat

3.8 Deployment Model

Software yang digunakan untuk melakukan *deployment* pada penelitian ini adalah streamlit. Streamlit adalah sebuah *framework open-source* yang memungkinkan untuk membuat antarmuka pengguna (UD) interaktif dalam *Python*. Dalam penelitian

ini menggunakan *streamlit* agar dapat mengubah skrip *Python* menjadi aplikasi web interaktif dengan cepat, tanpa memerlukan pengetahuan mendalam tentang pengembangan web atau UI sehingga bisa menghemat waktu.

3.9 Analisis Kebutuhan

Dalam penelitian ini, dibutuhkan berbagai perangkat lunak yang diperlukan untuk merancang sistem agar berjalan sesuai yang diharapkan. Berikut daftar perangkat lunak yang digunakan :

A. *Python* 3.10.12

Python merupakan bahasa pemrograman tingkat tinggi yang dirancang untuk kemudahan dan fleksibilitas. Pada penelitian ini menggunakan *python* versi 3.10.12 karena versi ini mendukung banyak pustaka dan fitur terbaru dari *Python*, sehingga sangat cocok untuk pengembangan aplikasi *machine learning*, *data science*, dan *deep learning*.

B. *Tensorflow/Keras*

Tensorflow/Keras adalah pustaka *open source* yang digunakan untuk membangun dan melatih model *deep learning*. Pustaka ini dirancang agar mudah digunakan dan mendukung pengembangan model dengan cepat. Salah satu kegunaan utama *keras* yaitu untuk memprediksi deret waktu, seperti harga saham. melalui implementasi model-model jaringan saraf seperti RNN dan LSTM yang dapat menganalisis pola temporal dalam data.

C. *Pandas*

Pandas merupakan pustaka *python* yang sangat populer untuk manipulasi dan analisis data. Pustaka ini menyediakan struktur data seperti *dataframe* dan *series* yang memungkinkan pengguna melakukan operasi yang kompleks pada data seperti pembersihan, transformasi, dan analisis statistik. Pada pustaka ini juga bisa mengimpor data dari berbagai format seperti CSV, Excel, SQL, dan JSON.

D. *Numpy*

NumPy adalah pustaka inti untuk komputasi numerik dalam *Python*. Pustaka ini menawarkan objek *array* multidimensi yang memungkinkan operasi vektor dan matriks secara efisien. Selain itu, *NumPy* menyediakan berbagai fungsi matematika dan statistik penting, serta mudah terintegrasi dengan pustaka lain seperti *Pandas* dan *Matplotlib* untuk analisis dan visualisasi data.

E. *Matplotlib/Seaborn*

Matplotlib merupakan pustaka visualisasi data yang digunakan untuk membuat grafik dan plot dari data numerik, seperti membuat berbagai jenis plot, mulai dari grafik garis, diagram batang, hingga *scatter plot*. Sementara *seaborn* merupakan pustaka visualisasi yang dibangun di atas *Matplotlib* dan dirancang untuk membuat visualisasi statistik yang lebih menarik dan informatif. *Seaborn* mempermudah pembuatan plot statistik, seperti *heatmap* dan lain sebagainya.

F. *Scikit-learn*

Scikit-learn adalah pustaka *Python* yang sangat populer untuk *machine learning*. Pustaka ini menyediakan berbagai alat untuk *preprocessing data*, seperti normalisasi dan pembagian data, serta berbagai algoritma untuk klasifikasi, regresi, pengelompokan, dan evaluasi model. *Scikit-learn* juga mendukung teknik validasi silang dan tuning hyperparameter untuk mengoptimalkan model.

G. *Jupyter Notebook*

Jupyter Notebook adalah aplikasi berbasis *web* yang digunakan untuk membuat dan menjalankan kode pemrograman secara interaktif. *Jupyter* banyak digunakan dalam *data science*, *machine learning*, dan penelitian akademik karena memungkinkan pengguna untuk menulis kode, melihat *output*, serta menyertakan teks, gambar, dan grafik dalam satu dokumen.

3.10 Perancangan sistem



Gambar 3. 2 *Flowchart* Perancangan sistem

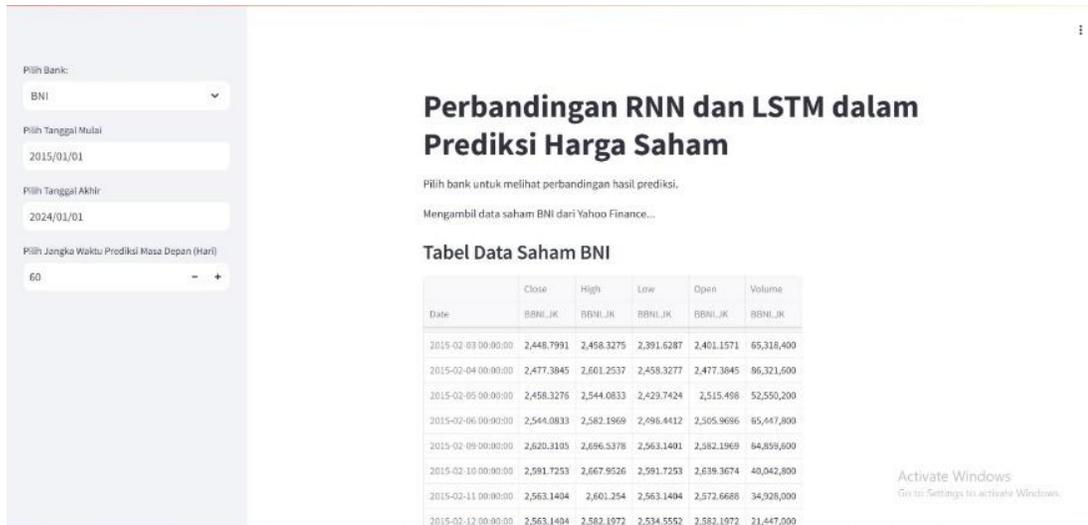
Flowchart tersebut menggambarkan alur proses prediksi harga saham Bank. Proses dimulai dengan memilih data Bank yang akan digunakan (Bank Mandiri, Bank BCA, Bank BNI, Bank BRI). Setelah itu, pengguna memasukkan rentang waktu data historis dengan menentukan data awal dan data akhir. Selanjutnya, pengguna menentukan jangka waktu prediksi yang diinginkan, misalnya untuk satu minggu atau satu bulan ke depan. Setelah semua parameter dimasukkan, sistem akan memproses data dan melakukan perhitungan prediksi menggunakan metode RNN dan LSTM.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil

4.1.1 Dashboard Streamlit



Gambar 4. 1 Tampilan Utama Sistem

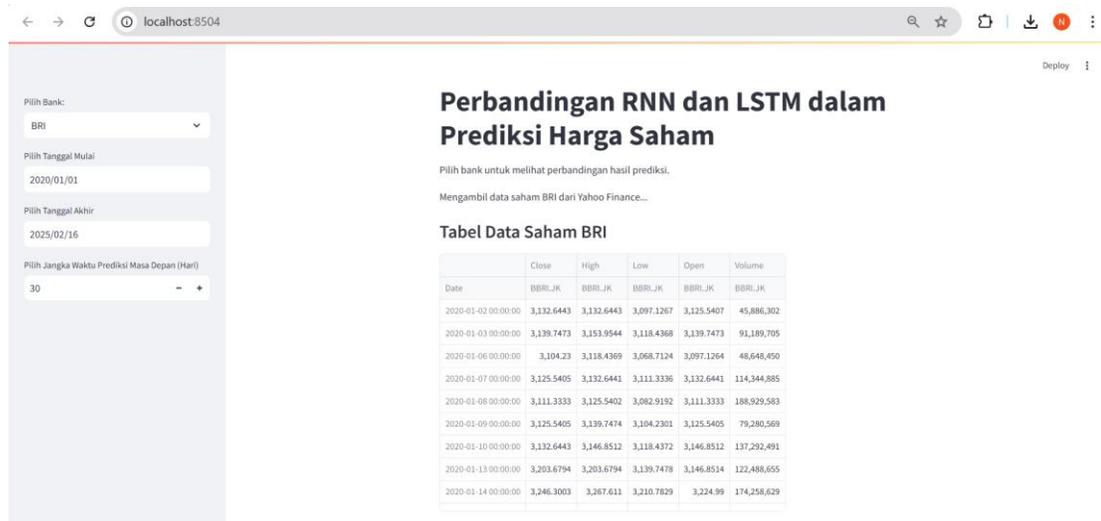
Gambar 4.1 merupakan tampilan *dashboard* yang memungkinkan pengguna untuk melakukan perbandingan antara hasil prediksi harga saham menggunakan dua model RNN dan LSTM. Pengguna dapat memilih bank yang diinginkan (dalam contoh ini, Bank BNI), serta menentukan periode waktu analisis dan jangka waktu prediksi yang diinginkan.

Fitur utama dari dashboard meliputi: Pemilihan bank yang ingin dianalisis. Penentuan tanggal mulai dan tanggal akhir untuk memperoleh data saham dari Yahoo Finance. pemilihan jangka waktu prediksi dalam hari (misalnya 60 hari ke depan) untuk memproyeksikan harga saham masa depan. Selain itu, dashboard juga menampilkan tabel data saham yang mencakup informasi penting seperti harga penutupan, harga tertinggi, harga terendah, harga pembukaan, dan volume perdagangan untuk setiap tanggal yang dipilih

4.1.2 Hasil prediksi

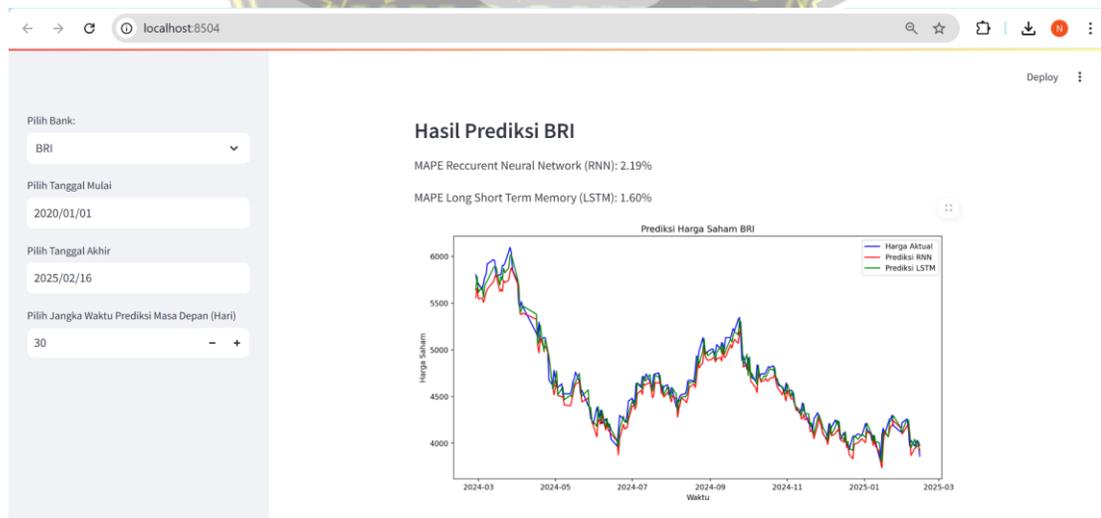
Berikut merupakan hasil uji coba sistem untuk hasil prediksi dari masing-masing bank dari rentang waktu 01 Januari 2020 hingga 16 Februari 2024.

a. Bank BRI



Gambar 4. 2 Tampilan Utama Sistem untuk Data Prediksi Bank BRI

Gambar 4.2 merupakan tampilan data dari bank BRI dari dari tanggal 01 Januari 2020 hingga 16 Februari 2024. Tabel tersebut memuat data *Date*, *Close*, *High*, *Low*, *Open*, dan *Volume*



Gambar 4. 3 Tampilan Utama Sistem untuk Data Prediksi Bank BRI

Gambar tersebut menampilkan grafik hasil prediksi dari model RNN dan LSTM untuk memprediksi harga saham Bank BRI. Terlihat bahwa hasil model LSTM lebih baik dari model RNN dengan nilai MAPE 1,60%



Tabel Perbandingan Hasil Prediksi

Date	data_aktual	prediksi_rnn	prediksi_lstm
2024-02-28 00:00:00	5,803.6836	5,556.835	5,641.79
2024-02-29 00:00:00	5,710.4517	5,657.1045	5,785.252
2024-03-01 00:00:00	5,710.4517	5,547.1738	5,616.3813
2024-03-04 00:00:00	5,640.5278	5,556.0161	5,677.356
2024-03-05 00:00:00	5,710.4517	5,508.6606	5,563.4067
2024-03-06 00:00:00	5,780.3755	5,555.5674	5,683.4961
2024-03-07 00:00:00	5,803.6836	5,611.0972	5,719.6567
2024-03-08 00:00:00	5,920.2231	5,650.186	5,739.9404
2024-03-13 00:00:00	5,966.8394	5,730.5835	5,878.4546
2024-03-14 00:00:00	5,952.3213	5,782.5801	5,895.1196

Gambar 4. 4 Tampilan Utama Sistem untuk Data Prediksi Bank BRI

Gambar tersebut merupakan tampilan dari hasil data prediksi dari Model RNN dan LSTM yang mana data hasil prediksi dari model LSTM cenderung lebih mendekati nilai aktual dibandingkan dengan model RNN



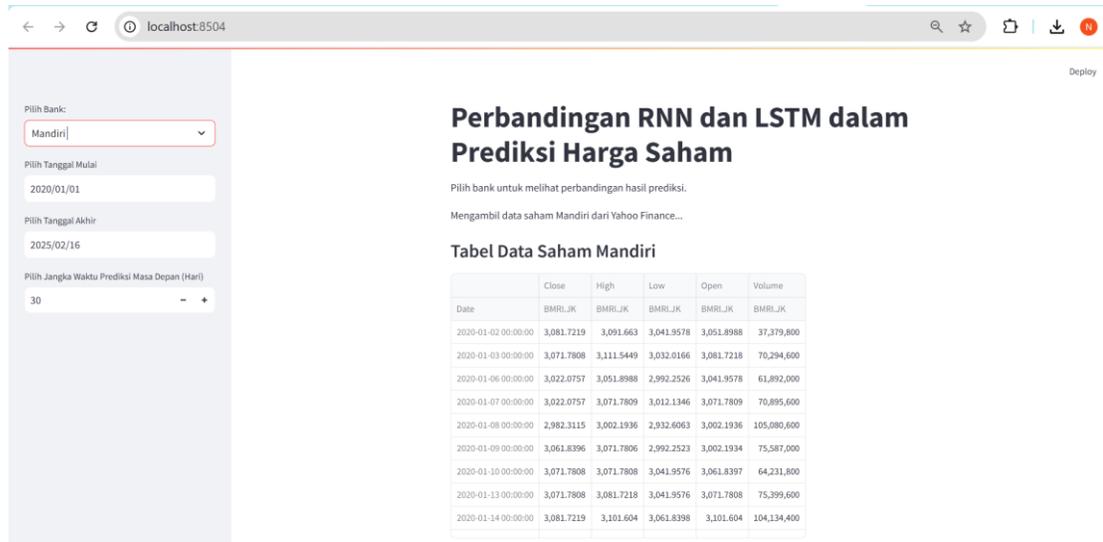
Prediksi Harga Saham BRI 30 Hari Ke Depan

	Waktu di Masa Depan	Prediksi RNN	Prediksi LSTM
0	2025-02-15 00:00:00	3,776.6396	3,839.6243
1	2025-02-16 00:00:00	3,717.6123	3,839.5271
2	2025-02-17 00:00:00	3,666.9951	3,834.0217
3	2025-02-18 00:00:00	3,631.426	3,827.531
4	2025-02-19 00:00:00	3,590.9773	3,820.9512
5	2025-02-20 00:00:00	3,548.5747	3,814.6687
6	2025-02-21 00:00:00	3,536.7927	3,808.8489
7	2025-02-22 00:00:00	3,462.9666	3,803.5479
8	2025-02-23 00:00:00	3,353.1511	3,798.752
9	2025-02-24 00:00:00	3,305.197	3,794.4011

Gambar 4. 5 Tampilan Utama Sistem untuk Data Prediksi Bank BRI

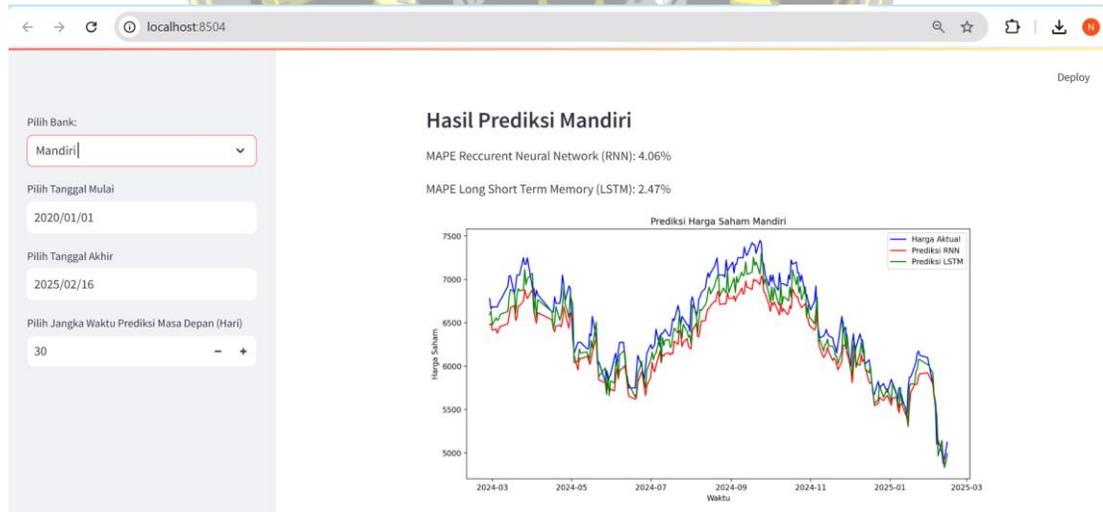
Gambar tersebut merupakan tampilan perbandingan dari hasil prediksi model RNN dan LSTM untuk 30 hari kedepan

b. Bank Mandiri



Gambar 4. 6 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri

Gambar tersebut merupakan tampilan data dari bank Mandiri dari dari tanggal 01 Januari 2020 hingga 16 Februari. Tabel tersebut memuat *data Date, Close, High, Low, Open, dan Volume*



Gambar 4. 7 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri

Gambar tersebut menampilkan grafik hasil prediksi dari model RNN dan LSTM untuk memprediksi harga saham Bank BRI. Terlihat bahwa hasil model LSTM lebih baik dari model RNN dengan nilai MAPE 2,47%.



Pilih Bank:
Mandiri

Pilih Tanggal Mulai
2020/01/01

Pilih Tanggal Akhir
2025/02/16

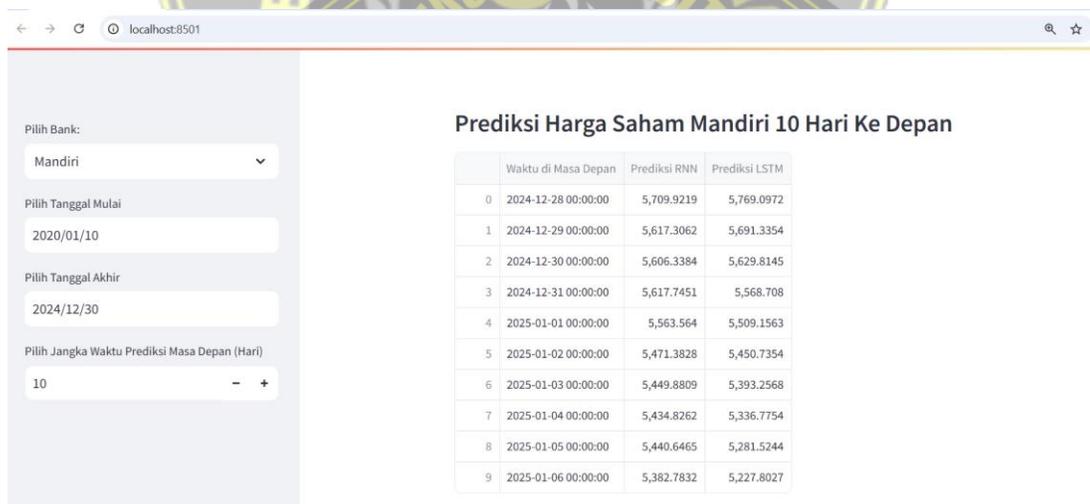
Pilih Jangka Waktu Prediksi Masa Depan (Hari)
30

Tabel Perbandingan Hasil Prediksi

Date	data_aktual	prediksi_rnn	prediksi_lstm
2024-02-28 00:00:00	6,778.3403	6,481.6748	6,592.0464
2024-02-29 00:00:00	6,659.4224	6,487.0127	6,637.0166
2024-03-01 00:00:00	6,683.2061	6,414.2246	6,475.019
2024-03-04 00:00:00	6,683.2061	6,429.9683	6,559.3535
2024-03-05 00:00:00	6,683.2061	6,380.9961	6,528.3062
2024-03-06 00:00:00	6,730.7734	6,423.6235	6,539.9224
2024-03-07 00:00:00	6,754.5566	6,461.5869	6,593.0908
2024-03-08 00:00:00	6,778.3403	6,459.8252	6,604.249
2024-03-13 00:00:00	6,921.0425	6,489.3506	6,630.3623
2024-03-14 00:00:00	7,039.9609	6,580.5996	6,795.2241

Gambar 4. 8 Tampilan Utama Sistem untuk Data Prediksi Bank Mandiri

Gambar tersebut merupakan tampilan dari hasil data prediksi dari Model RNN dan LSTM yang mana data hasil prediksi dari model LSTM cenderung lebih mendekati nilai aktual dibandingkan dengan model RNN



Pilih Bank:
Mandiri

Pilih Tanggal Mulai
2020/01/10

Pilih Tanggal Akhir
2024/12/30

Pilih Jangka Waktu Prediksi Masa Depan (Hari)
10

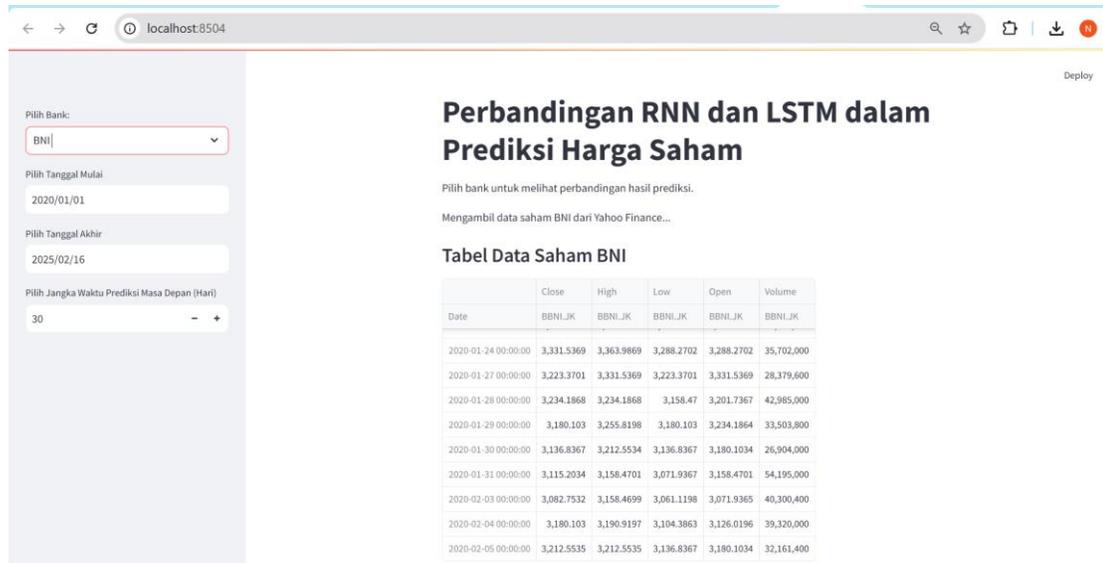
Prediksi Harga Saham Mandiri 10 Hari Ke Depan

	Waktu di Masa Depan	Prediksi RNN	Prediksi LSTM
0	2024-12-28 00:00:00	5,709.9219	5,769.0972
1	2024-12-29 00:00:00	5,617.3062	5,691.3354
2	2024-12-30 00:00:00	5,606.3384	5,629.8145
3	2024-12-31 00:00:00	5,617.7451	5,568.708
4	2025-01-01 00:00:00	5,563.564	5,509.1563
5	2025-01-02 00:00:00	5,471.3828	5,450.7354
6	2025-01-03 00:00:00	5,449.8809	5,393.2568
7	2025-01-04 00:00:00	5,434.8262	5,336.7754
8	2025-01-05 00:00:00	5,440.6465	5,281.5244
9	2025-01-06 00:00:00	5,382.7832	5,227.8027

Gambar 4. 9 Tampilan Utama Sistem Untuk Data Prediksi Bank Mandiri

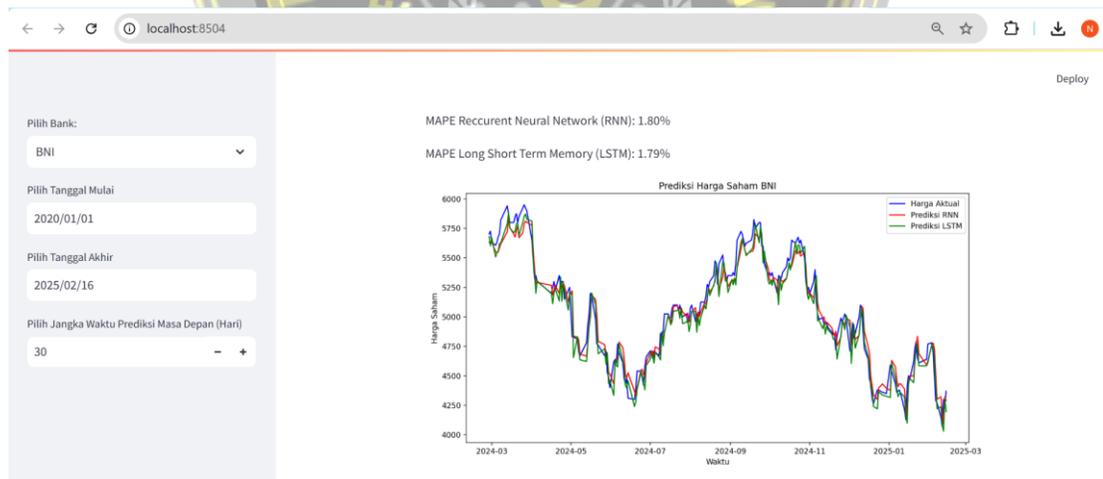
Gambar tersebut merupakan tampilan perbandingan dari hasil prediksi model RNN dan LSTM untuk 10 hari kedepan

c. Bank BNI



Gambar 4. 10 Tampilan Utama Sistem untuk Data Prediksi Bank BNI

Gambar tersebut merupakan tampilan data dari bank BRI dari dari tanggal 01 Januari 2020 hingga 14 Februari 2025. Tabel tersebut memuat data *Date*, *Close*, *High*, *Low*, *Open*, dan *Volume*



Gambar 4. 11 Tampilan Utama Sistem untuk Data Prediksi Bank BNI

Gambar tersebut menampilkan grafik hasil prediksi dari model RNN dan LSTM untuk memprediksi harga saham Bank BRI. Terlihat bahwa hasil model LSTM lebih baik dari model RNN dengan nilai MAPE 1,79%.

Pilih Bank:
BNI

Pilih Tanggal Mulai
2020/01/01

Pilih Tanggal Akhir
2025/02/16

Pilih Jangka Waktu Prediksi Masa Depan (Hari)
30

Tabel Perbandingan Hasil Prediksi

Date	data_aktual	prediksi_rnn	prediksi_lstm
2024-02-28 00:00:00	5,702.4863	5,630.9243	5,680.2632
2024-02-29 00:00:00	5,726.3462	5,623.4126	5,602.6099
2024-03-01 00:00:00	5,630.9067	5,634.9883	5,670.3931
2024-03-04 00:00:00	5,607.0474	5,552.4375	5,507.644
2024-03-05 00:00:00	5,630.9067	5,545.5142	5,544.8452
2024-03-06 00:00:00	5,678.6265	5,548.9126	5,556
2024-03-07 00:00:00	5,702.4863	5,586.6113	5,620.2397
2024-03-08 00:00:00	5,821.7852	5,617.7173	5,627.7378
2024-03-13 00:00:00	5,941.084	5,720.7002	5,789.7144
2024-03-14 00:00:00	5,869.5049	5,833.4131	5,886.2441

Gambar 4. 12 Tampilan Utama Sistem untuk Data Prediksi Bank BNI

Gambar tersebut merupakan tampilan dari hasil data prediksi dari Model RNN dan LSTM yang mana data hasil prediksi dari model LSTM cenderung lebih mendekati nilai aktual dibandingkan dengan model RNN.

Pilih Bank:
BNI

Pilih Tanggal Mulai
2020/01/01

Pilih Tanggal Akhir
2025/02/16

Pilih Jangka Waktu Prediksi Masa Depan (Hari)
30

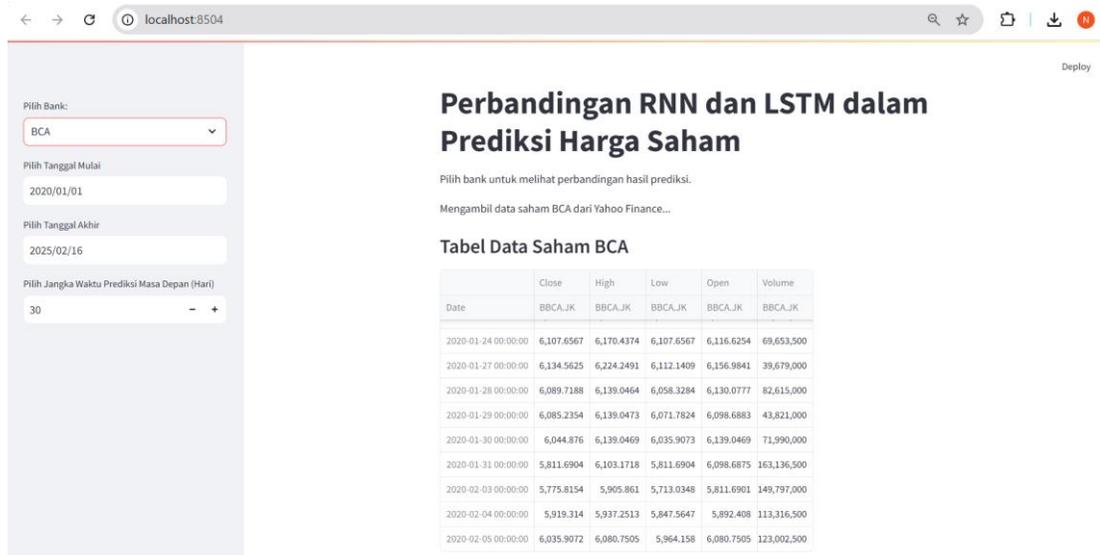
Prediksi Harga Saham BNI 30 Hari Ke Depan

	Waktu di Masa Depan	Prediksi RNN	Prediksi LSTM
0	2025-02-15 00:00:00	4,399.5684	4,372.3213
1	2025-02-16 00:00:00	4,437.6831	4,333.2466
2	2025-02-17 00:00:00	4,493.9355	4,302.4893
3	2025-02-18 00:00:00	4,558.8286	4,271.4478
4	2025-02-19 00:00:00	4,624.1138	4,242.0674
5	2025-02-20 00:00:00	4,654.3765	4,214.5908
6	2025-02-21 00:00:00	4,667.8843	4,188.5986
7	2025-02-22 00:00:00	4,671.6523	4,163.521
8	2025-02-23 00:00:00	4,648.4219	4,138.896
9	2025-02-24 00:00:00	4,664.7124	4,114.4517

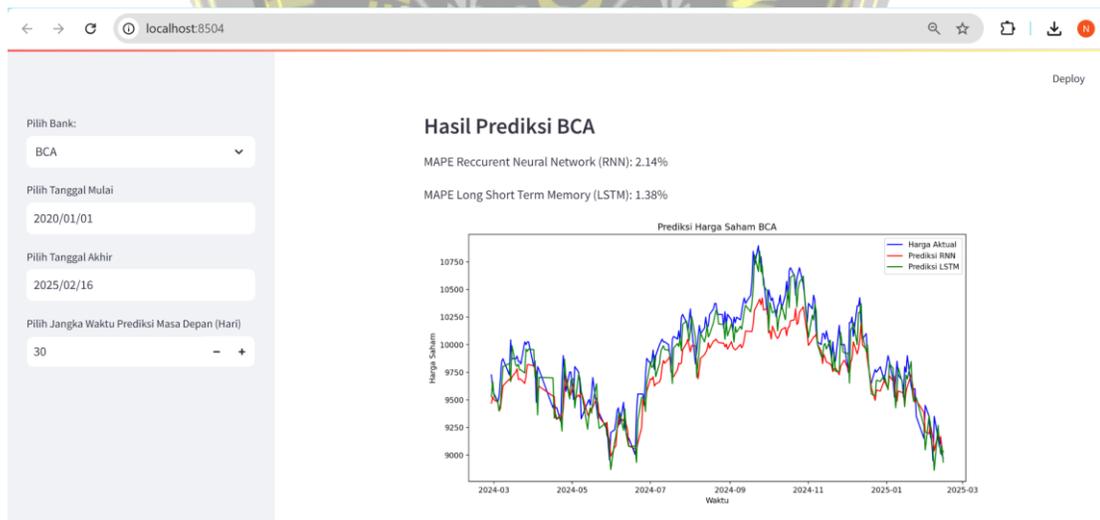
Gambar 4. 13 Tampilan Utama Sistem untuk Data Prediksi Bank BNI

Gambar tersebut merupakan tampilan perbandingan dari hasil prediksi model RNN dan LSTM untuk 10 hari kedepan

d. Bank BCA



Gambar 4. 14 Tampilan Utama Sistem untuk Data Prediksi Bank BCA gambar tersebut merupakan tampilan data dari bank BCA dari dari tanggal 10 Januari 2020 hingga 30 Desember 2024. Tabel tersebut memuat data *Date*, *Close*, *High*, *Low*, *Open*, dan *Volume*



Gambar 4. 15 Tampilan Utama Sistem untuk Data Prediksi Bank BCA Gambar tersebut menampilkan grafik hasil prediksi dari model RNN dan LSTM untuk memprediksi harga saham Bank BRI. Terlihat bahwa hasil model LSTM lebih baik dari model RNN dengan nilai MAPE 1.38%.



Pilih Bank: BCA

Pilih Tanggal Mulai: 2020/01/01

Pilih Tanggal Akhir: 2025/02/16

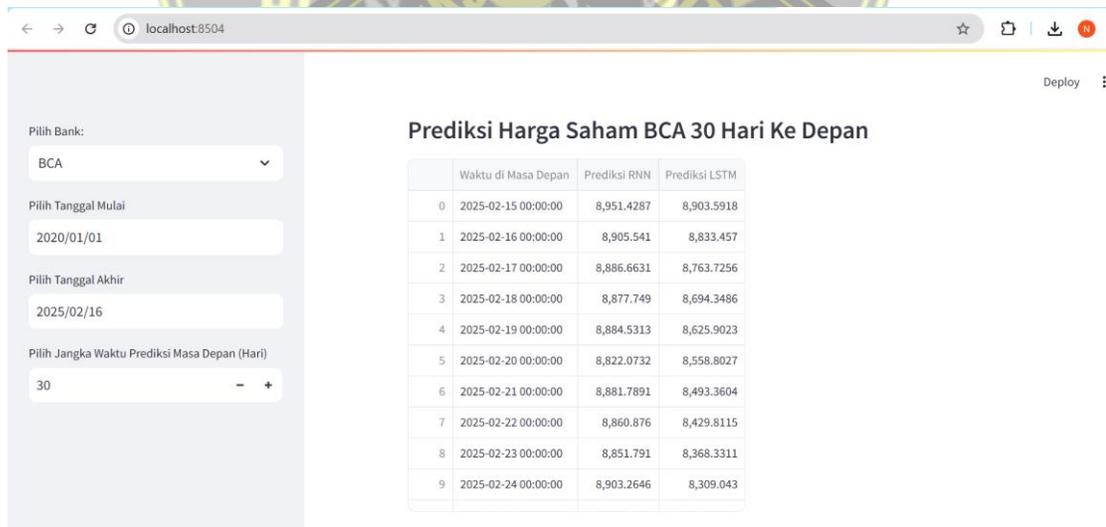
Pilih Jangka Waktu Prediksi Masa Depan (Hari): 30

Tabel Perbandingan Hasil Prediksi

Date	data_aktual	prediksi_rnn	prediksi_lstm
2024-02-28 00:00:00	9,726.2422	9,469.8877	9,529.6465
2024-02-29 00:00:00	9,604.6641	9,525.877	9,664.7061
2024-03-01 00:00:00	9,556.0332	9,501.2041	9,540.2412
2024-03-04 00:00:00	9,483.0859	9,478.1523	9,479.4092
2024-03-05 00:00:00	9,531.7168	9,400.5332	9,398.3896
2024-03-06 00:00:00	9,677.6113	9,419.5547	9,449.623
2024-03-07 00:00:00	9,847.8203	9,517.6885	9,610.7002
2024-03-08 00:00:00	9,872.1357	9,626.5283	9,800.7822
2024-03-13 00:00:00	9,726.2422	9,687.46	9,829.6465
2024-03-14 00:00:00	10,042.3447	9,671.2617	9,664.417

Gambar 4. 16 Tampilan Utama Sistem untuk Data Prediksi Bank BCA

Gambar tersebut merupakan tampilan dari hasil data prediksi dari Model RNN dan LSTM yang mana data hasil prediksi dari model LSTM cenderung lebih mendekati nilai aktual dibandingkan dengan model RNN.



Pilih Bank: BCA

Pilih Tanggal Mulai: 2020/01/01

Pilih Tanggal Akhir: 2025/02/16

Pilih Jangka Waktu Prediksi Masa Depan (Hari): 30

Prediksi Harga Saham BCA 30 Hari Ke Depan

	Waktu di Masa Depan	Prediksi RNN	Prediksi LSTM
0	2025-02-15 00:00:00	8,951.4287	8,903.5918
1	2025-02-16 00:00:00	8,905.541	8,833.457
2	2025-02-17 00:00:00	8,886.6631	8,763.7256
3	2025-02-18 00:00:00	8,877.749	8,694.3486
4	2025-02-19 00:00:00	8,884.5313	8,625.9023
5	2025-02-20 00:00:00	8,822.0732	8,558.8027
6	2025-02-21 00:00:00	8,881.7891	8,493.3604
7	2025-02-22 00:00:00	8,860.876	8,429.8115
8	2025-02-23 00:00:00	8,851.791	8,368.3311
9	2025-02-24 00:00:00	8,903.2646	8,309.043

Gambar 4. 17 Tampilan Utama Sistem untuk Data Prediksi Bank BCA

Gambar tersebut merupakan tampilan perbandingan dari hasil prediksi model RNN dan LSTM untuk 10 hari kedepan

4.2 Analisis

4.2.1 Persiapan Data

Pada tahap persiapan data, penelitian ini menggunakan data transaksi yang diperoleh dari empat bank besar di Indonesia, yaitu Bank Mandiri, BRI, BNI, dan BCA. Data transaksi ini disimpan dalam bentuk file CSV terpisah untuk setiap bank. Berikut data historis dari masing-masing bank yang terdapat pada Tabel 4.1 hingga Tabel 4.4

Tabel 4. 1 Data Bank mandiri

Date	Close	High	Low	Open	Volume
2010-01-04	733.054810	733.054810	703.286093	733.054810	212406442
2010-01-05	725.612671	736.775986	714.449430	725.612671	111068453
2010-01-06	714.449341	729.333661	707.007181	714.449341	69937714
2010-01-07	703.286011	710.728244	692.122771	703.286011	118305283
2010-01-08	710.728333	714.449413	703.286098	710.728333	52838212
...
2024-12-20	5675.000000	5750.000000	5625.000000	5675.000000	148568600
2024-12-23	5825.000000	5875.000000	5725.000000	5800.000000	82910100
2024-12-24	5750.000000	5875.000000	5750.000000	5825.000000	41240900
2024-12-27	5800.000000	5850.000000	5750.000000	5750.000000	46126800
2024-12-30	5700.000000	5750.000000	5650.000000	5675.000000	96782100

Tabel 4. 2 Data Bank BRI

Date	Close	High	Low	Open	Volume
2010-01-04	432.139069	434.909174	423.828677	432.139069	125094160
2010-01-05	434.909210	437.679353	432.139104	434.909210	163644326
2010-01-06	429.369049	440.449550	426.598942	429.369049	104842810
2010-01-07	423.828735	429.368985	418.288523	423.828735	101933260
2010-01-08	426.598846	426.598846	418.288491	426.598846	87237006
...
2024-12-20	3929.500000	3987.571429	3919.821429	3939.178571	252689600
2024-12-23	4074.678467	4074.678467	3977.892755	3997.249897	167689800
2024-12-24	4065.000000	4113.392857	4035.964286	4084.357143	199536100
2024-12-27	4100.000000	4120.000000	4080.000000	4100.000000	143104400
2024-12-30	4080.000000	4120.000000	4070.000000	4080.000000	153934700

Tabel 4. 3 Data Bank BNI

Date	Close	High	Low	Open	Volume
2010-01-04	647.517456	658.905947	644.263571	647.517456	17285826
2010-01-05	647.517456	658.905947	644.263571	647.517456	31016809
2010-01-06	644.263611	650.771338	641.009768	644.263611	32037904
2010-01-07	641.009705	650.771274	634.502020	641.009705	21151852
2010-01-08	641.009705	647.517432	637.755862	641.009705	12106513

...
2024-12-20	4260.000000	4360.000000	4260.000000	4330.000000	83872500
2024-12-23	4380.000000	4390.000000	4320.000000	4320.000000	43668100
2024-12-24	4380.000000	4400.000000	4340.000000	4380.000000	21458900
2024-12-27	4360.000000	4430.000000	4360.000000	4380.000000	22838700
2024-12-30	4350.000000	4380.000000	4270.000000	4340.000000	43899600

Tabel 4. 4 Data Bank BCA

Date	Close	High	Low	Open	Volume
2010-01-04	796.387695	796.387695	760.905075	796.387695	77812500
2010-01-05	804.272644	804.272644	780.617566	804.272644	44442500
2010-01-06	780.617615	804.272694	780.617615	780.617615	27782500
2010-01-07	784.560120	784.560120	760.905041	784.560120	38632500
2010-01-08	764.847717	784.560287	764.847717	764.847717	27427500
...
2024-12-20	9650.000000	9750.000000	9625.000000	9650.000000	91576900
2024-12-23	9775.000000	9850.000000	9700.000000	9700.000000	43292100
2024-12-24	9750.000000	9900.000000	9750.000000	9850.000000	32415700
2024-12-27	9800.000000	9825.000000	9725.000000	9800.000000	24016700
2024-12-30	9675.000000	9825.000000	9675.000000	9800.000000	56350100

Pada Tabel 4.1 hingga Tabel 4.4 memuat dataset historis yang berisi mengenai pergerakan harga saham dari tanggal 4 Januari 2010 hingga 30 Desember 2024. Terdapat kolom *Date* (Tanggal data saham dicatat), *Close* (Harga penutupan saham pada hari tersebut), *High* (Harga tertinggi saham yang dicapai pada hari tersebut), *Low* (Harga terendah saham yang dicapai pada hari tersebut), *Open* (Harga pembukaan saham pada hari tersebut), *Volume* (Jumlah saham yang diperdagangkan pada hari

tersebut). Untuk memberikan gambaran yang lebih jelas, pergerakan harga saham masing-masing bank dapat dilihat pada Gambar 4.1.



Gambar 4. 18 Grafik dari Masing-Masing Bank

Gambar 4.1 merupakan grafik harga saham penutupan (*Close*) dari empat bank: Mandiri, BNI, BRI, BCA dalam rentang waktu dari tahun 2010 hingga 2024. Masing-masing bank digambarkan dengan warna yang berbeda, yaitu BCA menunjukkan tren kenaikan yang sangat stabil dan konsisten sepanjang periode yang diamati, dengan lonjakan harga yang lebih tajam dari tahun 2014. BRI dan Mandiri juga menunjukkan kenaikan harga saham, tetapi dengan fluktuasi yang lebih terlihat di beberapa periode. BNI mengalami lebih banyak fluktuasi dengan beberapa periode penurunan yang lebih besar dibandingkan dengan yang lainnya. Perbedaan tren ini bisa dipengaruhi oleh berbagai faktor, termasuk performa keuangan masing-masing bank, kondisi ekonomi makro, dan kebijakan yang diterapkan oleh pemerintah atau masing-masing bank tersebut selama periode waktu yang berbeda.

4.2.2 Data preprocessing

A. Data Cleaning

Tahap pembersihan data dilakukan untuk memastikan bahwa dataset yang digunakan dalam penelitian bebas dari nilai yang hilang (*missing values*). Proses ini penting untuk menjaga kualitas analisis dan mencegah bias dalam hasil penelitian. Pengecekan *missing values* dilakukan dengan menggunakan perintah `isnull().sum()` pada setiap dataset bank.

```
#Data Cleaning: Cek dan Bersihkan Missing Values
for bank, df in data.items():
    print(f"\nMissing Values in {bank}:")
    print(df.isnull().sum())
```

Dari kode tersebut menghasilkan :

```
Missing Values in BCA:
Close    0
High     0
Low      0
Open     0
Volume   0
dtype: int64

Missing Values in BRI:
Close    0
High     0
Low      0
Open     0
Volume   0
dtype: int64

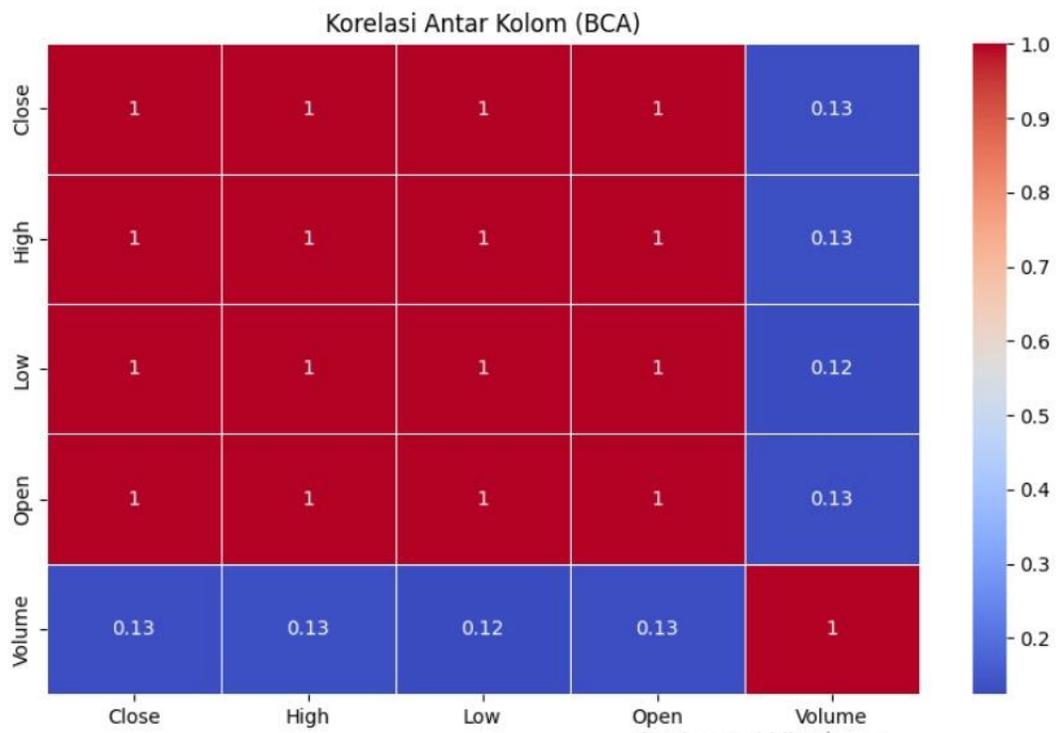
Missing Values in Mandiri:
Close    0
High     0
Low      0
Open     0
Volume   0
dtype: int64

Missing Values in BNI:
Close    0
High     0
Low      0
Open     0
Volume   0
dtype: int64
```

Gambar 4. 19 Cek *missing value*

Gambar 4.19 menunjukkan hasil bahwa tidak terdapat *missing values* pada kolom-kolom data seperti *Close*, *High*, *Low*, *Open*, dan *Volume* untuk semua data bank. Dengan demikian, dataset yang digunakan sudah bersih dan siap untuk dianalisis lebih lanjut. Selanjutnya dilakukan tahap Analisis Data Eksploratori (EDA). Tahap ini

dilakukan untuk pemeriksaan hubungan antar variabel dalam data saham menggunakan analisis korelasi. Korelasi antar kolom ini bertujuan untuk memahami sejauh mana hubungan antara variabel-variabel seperti harga penutupan (*Close*), harga tertinggi (*High*), harga terendah (*Low*), harga pembukaan (*Open*), dan volume transaksi (*Volume*).



Gambar 4. 20 heatmap hasil korelasi fitur

Hasil analisis korelasi menggunakan heatmap menunjukkan pada Gambar 4. 20 yaitu adanya korelasi positif sempurna antara variabel harga saham (*Close*, *High*, *Low*, dan *Open*). Hal ini mengindikasikan bahwa pergerakan salah satu variabel harga cenderung diikuti oleh variabel harga lainnya dalam arah yang sama. Sementara itu, korelasi antara Volume dan variabel harga menunjukkan hubungan yang sangat lemah, dengan nilai korelasi berkisar antara 0.12 hingga 0.13. Temuan ini mengindikasikan bahwa perubahan volume perdagangan tidak memiliki pengaruh signifikan terhadap fluktuasi harga saham BCA dalam periode yang dianalisis.

Oleh karena itu, data yang akan digunakan untuk prediksi adalah data harga penutupan (Close), karena harga penutupan mencerminkan nilai akhir transaksi dalam satu periode perdagangan dan sering digunakan sebagai acuan utama dalam analisis tren serta strategi investasi. Selain itu, harga penutupan juga lebih stabil dibandingkan harga intraday (Open, High, Low), sehingga lebih ideal untuk model prediktif berbasis time series seperti RNN dan LSTM (Ramdhani dkk, 2019)

B. Normalisasi data

Untuk memastikan bahwa data memiliki skala yang seragam sebelum digunakan dalam proses pemodelan, dilakukan normalisasi data menggunakan teknik *Min-Max Normalization*. Normalisasi ini penting untuk menghindari dominasi fitur dengan skala besar terhadap fitur lainnya dalam model prediktif. Proses normalisasi dilakukan dengan menggunakan pustaka `scikit-learn` melalui fungsi `MinMaxScaler`, yang mengubah skala data ke dalam rentang 0 hingga 1.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(mandiri)
scaled_data
```

Hasil dari proses normalisasi sebagai berikut :

```
array([[0.0131212 ],
       [0.01202778],
       [0.01038762],
       ...,
       [0.75022962],
       [0.75757581],
       [0.74288343]])
```

Hasil normalisasi menunjukkan bahwa seluruh nilai dalam dataset telah dikonversi ke dalam rentang antara 0 hingga 1. Sebelum melatih model prediksi, data yang telah dinormalisasi perlu dipersiapkan dalam format yang sesuai untuk digunakan sebagai input dan target. Untuk keperluan ini, data saham yang telah dinormalisasi digunakan dengan teknik *sliding window* untuk membentuk dataset yang terdiri dari 100 data sebelumnya sebagai fitur dan data ke-101 sebagai target yang ingin diprediksi.

```

x_data = []
y_data = []

for i in range(100, len(scaled_data)):
    x_data.append(scaled_data[i-100:i])
    y_data.append(scaled_data[i])

import numpy as np
x_data, y_data = np.array(x_data), np.array(y_data)

```

Proses ini dilakukan dengan cara mengambil 100 data sebelumnya sebagai *input* (variabel `x_data`) dan nilai pada indeks ke-101 sebagai target (variabel `y_data`). Setelah itu, `x_data` dan `y_data` yang awalnya berbentuk *list* dikonversi menjadi *array NumPy* untuk mempermudah pengolahan lebih lanjut dalam model prediksi.

C. Pembagian data

Setelah data dipersiapkan dalam bentuk yang sesuai untuk model, langkah selanjutnya adalah membagi data menjadi dua set, *training set* dan *test set*. Pembagian ini dilakukan untuk melatih model dengan sebagian data (80%) dan menguji model dengan sisa data (20%), yang memungkinkan evaluasi performa model yang lebih objektif.

```

splitting_len = int(len(x_data)*0.8)
x_train = x_data[:splitting_len]
y_train = y_data[:splitting_len]

x_test = x_data[splitting_len:]
y_test = y_data[splitting_len:]

```

Training set : 80% dari data digunakan untuk melatih model. Data ini terdiri dari variabel *input* (`x_train`) dan target (`y_train`). *Test set*: 20% dari data digunakan untuk menguji model. Data ini terdiri dari variabel *input* (`x_test`) dan target (`y_test`). Pembagian ini memastikan bahwa model dapat dilatih pada sebagian data dan diuji pada data yang belum pernah dilihat sebelumnya, memberikan gambaran yang lebih baik tentang bagaimana model akan bekerja pada data nyata di luar sampel pelatihan. Setelah membagi data menjadi *training set* dan *test set*, langkah selanjutnya adalah memeriksa dimensi data untuk memastikan bahwa jumlah sampel dan fitur sesuai

dengan ekspektasi. Fungsi *shape* digunakan untuk memverifikasi dimensi dari *array input* dan target pada kedua set tersebut.

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

Hasil dari kode tersebut :

```
(2880, 100, 1)
(2880, 1)
(721, 100, 1)
(721, 1)
```

Hasil yang diperoleh adalah sebagai berikut: *Training Set: Input* (x_{train}): (2880, 100, 1) menunjukkan bahwa terdapat 2880 sampel, masing-masing dengan 100 data sebelumnya sebagai fitur dan 1 fitur per waktu. Target (y_{train}): (2880, 1) menunjukkan bahwa untuk setiap sampel terdapat satu nilai target yang ingin diprediksi. *Test Set Input* (x_{test}): (721, 100, 1) menunjukkan bahwa terdapat 721 sampel dalam data uji, masing-masing dengan 100 data sebelumnya sebagai fitur. Target (y_{test}): (721, 1) menunjukkan bahwa untuk setiap sampel terdapat satu nilai target dalam data uji. Dimensi ini menunjukkan bahwa data telah berhasil dibagi dengan baik, dan setiap set memiliki jumlah sampel yang sesuai untuk pelatihan dan pengujian model.

4.2.3 Perancangan model RNN dan LSTM

Uji coba parameter model pertama memiliki dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam *optimizer* dengan MSE *loss function*. Model ini dilatih menggunakan batch size 32 dan 30 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 30,
                                validation_data=(x_test, y_test),
verbose=1)

```

Uji coba parameter model kedua menggunakan dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 32 dan 50 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 50,
                                validation_data=(x_test,y_test),
verbose=1)

```

Uji coba parameter model ketiga memiliki dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 32 dan 100 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 100,
                                validation_data=(x_test,y_test),
verbose=1)

```

Uji coba parameter model ke empat memiliki dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model

ini dilatih menggunakan batch size 64 dan 30 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 30,
                                validation_data=(x_test,y_test),
                                verbose=1)

```

Uji coba parameter model kelima memiliki dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 64 dan 50 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 50,
                                validation_data=(x_test,y_test),
                                verbose=1)

```

Uji coba parameter model keenam memiliki dua lapisan SimpleRNN, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 64 dan 100 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_rnn_mandiri = Sequential()
model_rnn_mandiri.add(SimpleRNN(128,
return_sequences=True,input_shape=(x_train.shape[1],1)))
model_rnn_mandiri.add(SimpleRNN(64,return_sequences=False))
model_rnn_mandiri.add(Dense(25))
model_rnn_mandiri.add(Dense(1))
model_rnn_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_rnn_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 100,
                                validation_data=(x_test,y_test),
                                verbose=1)

```

Uji coba parameter model ketujuh memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 32 dan 30 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128, return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 30,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

Uji coba parameter model kedelapan memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 32 dan 50 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128,return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 50,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

Uji coba parameter model ke sembilan memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih

menggunakan batch size 32 dan 100 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128,return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=32, epochs = 100,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

Uji coba parameter model ke sepuluh memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 64 dan 30 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128,return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 30,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

Uji coba parameter model ke sebelas memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 64 dan 50 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128,return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 50,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

Uji coba parameter model ke dua belas memiliki dua lapisan LSTM, dua lapisan Dense, dan menggunakan Adam optimizer dengan MSE loss function. Model ini dilatih menggunakan batch size 64 dan 100 epoch, serta diuji dengan data validasi untuk mengevaluasi performanya.

```

model_lstm_mandiri = Sequential()
model_lstm_mandiri.add(LSTM(128,return_sequences=True,
input_shape=(x_train.shape[1],1)))
model_lstm_mandiri.add(LSTM(64,return_sequences=False))
model_lstm_mandiri.add(Dense(25))
model_lstm_mandiri.add(Dense(1))
model_lstm_mandiri.compile(optimizer='adam', loss='mean_squared_error')

history = model_lstm_mandiri.fit(x_train, y_train,
                                batch_size=64, epochs = 100,
                                validation_data=(x_test, y_test),
                                verbose=1)

```

4.2.4 Denormalisasi

Setelah model melakukan prediksi pada data yang telah dinormalisasi, langkah selanjutnya adalah mengembalikan hasil prediksi dan data target ke skala aslinya. Hal ini dilakukan dengan menggunakan metode `inverse_transform` dari `MinMaxScaler`, yang mengubah nilai kembali ke rentang aslinya, yaitu rentang sebelum proses normalisasi.

```

inv_predict_lstm_mandiri= scaler.inverse_transform(predict_lstm_mandiri)
inv_y_test = scaler.inverse_transform(y_test)

```

`inv_predict_lstm_mandiri` : Ini adalah hasil prediksi yang telah dikembalikan ke skala asli setelah proses normalisasi. `inv_y_test`: Ini adalah data target yang juga telah dikembalikan ke skala asli, yang memungkinkan perbandingan langsung antara nilai yang diprediksi dan nilai aktual. Proses ini sangat penting untuk memastikan bahwa perbandingan antara hasil prediksi dan data aktual dilakukan dalam skala yang konsisten dan dapat dimengerti.

4.2.5 Evaluasi model

Setelah dilakukan beberapa pengujian diatas, didapatkan hasil evaluasi model sebagai berikut :

Tabel 4. 5 Hasil evaluasi model

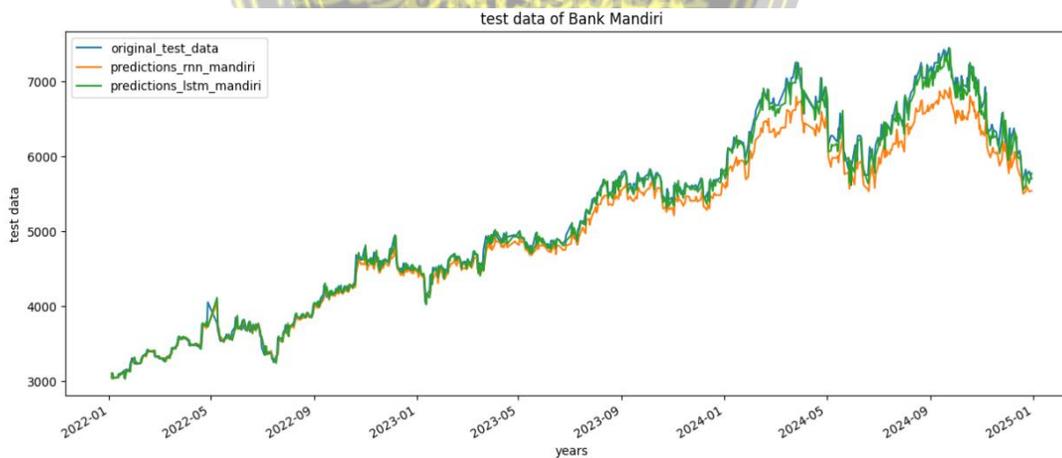
Bank	Model	Batch size	Epoch	MAPE
Mandiri	RNN	32	30	2,61%
	RNN	32	50	4,36%
	RNN	32	100	2,46%
	RNN	64	30	2,80%
	RNN	64	50	4,45%
	RNN	64	100	3,15%
	LSTM	32	30	1,55%
	LSTM	32	50	2,92%
	LSTM	32	100	1,48%
	LSTM	64	30	3,24%
	LSTM	64	50	1,94%
BRI	LSTM	64	100	1,46%
	RNN	32	30	3,35%
	RNN	32	50	2,09%
	RNN	32	100	1,30%
	RNN	64	30	1,87%
	RNN	64	50	4,17%
	RNN	64	100	1,87%
	LSTM	32	30	3,52%
	LSTM	32	50	1,32%
	LSTM	32	100	1,41%
	LSTM	64	30	2,17%
BNI	LSTM	64	50	1,43%
	LSTM	64	100	2,27%
	RNN	32	30	1,59%
	RNN	32	50	1,67%
	RNN	32	100	1,35%
	RNN	64	30	2,18%
	RNN	64	50	2,01%
	RNN	64	100	2,63%
	LSTM	32	30	1,24%
	LSTM	32	50	1,14%
	BCA	LSTM	32	100
LSTM		64	30	1,73%
LSTM		64	50	1,27%
LSTM		64	100	1,47%
BCA	RNN	32	30	2,33%
	RNN	32	50	2,64%
	RNN	32	100	2,18%

	RNN	64	30	4,11%
	RNN	64	50	2,80%
	RNN	64	100	1,49%
	LSTM	32	30	1,26%
	LSTM	32	50	1,17%
	LSTM	32	100	1,97%
	LSTM	64	30	2,25%
	LSTM	64	50	1,80%
	LSTM	64	100	1.18%

Berdasarkan Tabel 4.5, model LSTM cenderung memberikan nilai MAPE yang lebih rendah dibandingkan model RNN. Hal ini membuktikan bahwa model LSTM memiliki akurasi lebih tinggi dalam memprediksi harga saham Bank Mandiri, BRI, BNI, dan BCA, sesuai dengan skala keakuratan nilai MAPE yang dijelaskan pada Tabel 3.2. Selain itu, penggunaan batch size 64 dan epoch 50 menunjukkan hasil terbaik, karena mampu menurunkan nilai error secara signifikan dibandingkan parameter lainnya. Oleh karena itu, model LSTM dengan batch size 64 dan epoch 50 dapat dianggap sebagai pilihan paling optimal untuk memprediksi harga saham dengan tingkat akurasi yang lebih baik.

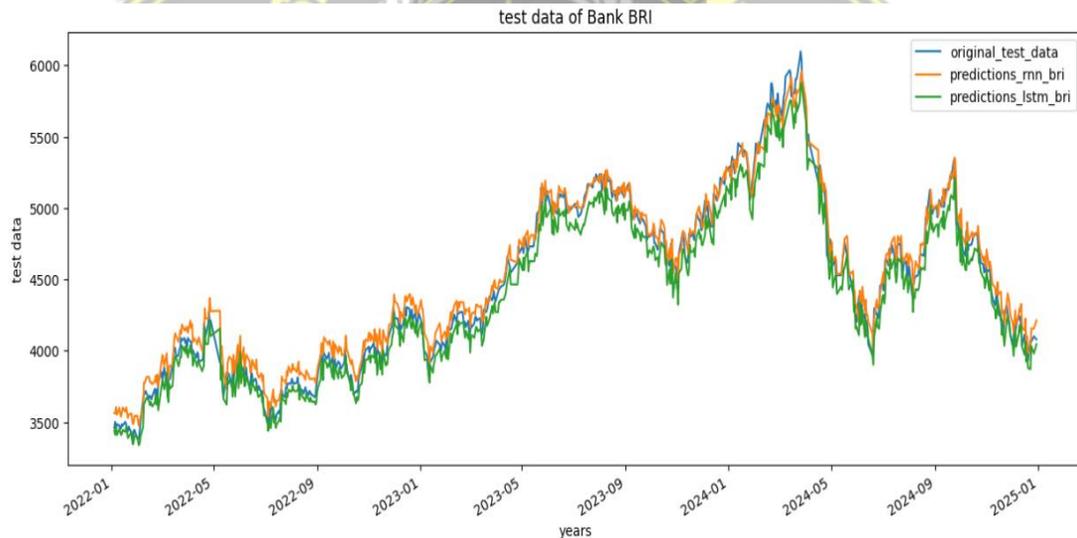
4.2.6 Hasil Perbandingan model

Berikut hasil grafik perbandingan data prediksi dari model RNN dan LSTM dengan parameter terbaik.



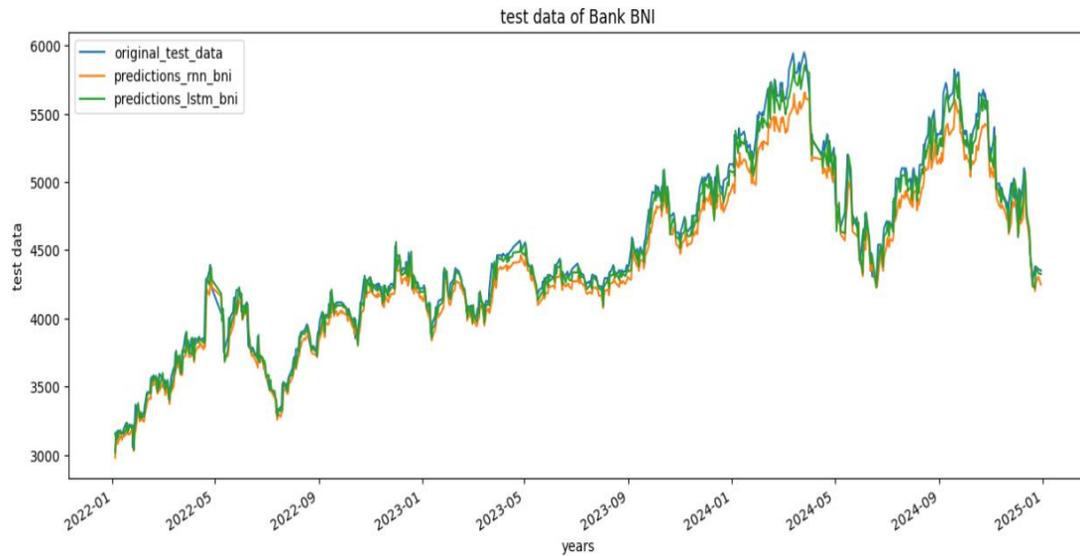
Gambar 4. 21 Grafik Perbandingan Model RNN dan LSTM Bank Mandiri

Grafik tersebut menampilkan perbandingan antara data aktual harga saham Bank Mandiri (*original_test_data*) dengan hasil prediksi dari dua model, yaitu RNN (*predictions_rnn_mandiri*) dan LSTM (*predictions_lstm_mandiri*), dalam rentang waktu Januari 2022 hingga Januari 2025. Dari grafik ini, terlihat bahwa garis hijau (LSTM) lebih mendekati garis biru (data aktual) dibandingkan dengan garis oranye (RNN). Hal ini menunjukkan bahwa model LSTM memberikan prediksi yang lebih akurat dibandingkan dengan RNN, karena lebih mampu menangkap pola pergerakan harga saham dengan lebih baik. Sementara itu, garis oranye (RNN) masih mengikuti tren harga saham, tetapi dengan penyimpangan yang lebih besar dibandingkan LSTM. Secara keseluruhan, grafik ini mengonfirmasi bahwa model LSTM lebih unggul dalam memprediksi harga saham Bank Mandiri, yang sejalan dengan hasil evaluasi menggunakan MAPE sebelumnya.



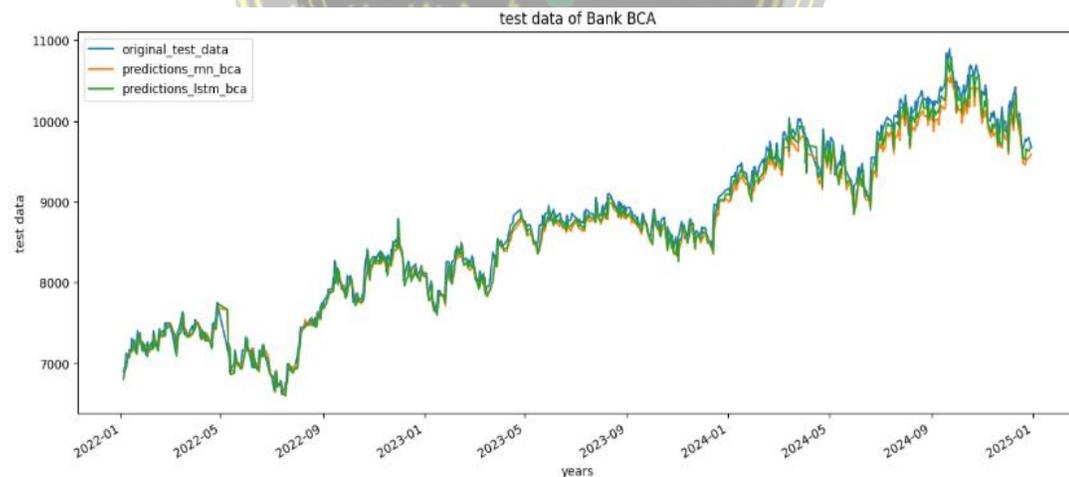
Gambar 4. 22 Grafik Perbandingan Model RNN dan LSTM Bank BRI

Pada grafik tersebut terlihat bahwa model LSTM lebih akurat dalam mengikuti tren harga saham dibandingkan RNN pada data bank BRI, karena prediksinya lebih mendekati data aktual. Model RNN cenderung memiliki penyimpangan yang lebih besar, terutama saat terjadi lonjakan harga, sedangkan LSTM lebih stabil dalam menangkap pola naik turun harga saham.



Gambar 4. 23 Grafik Perbandingan Model RNN dan LSTM Bank BRI

Pada grafik gambar 4.6 tersebut terlihat bahwa kedua model dapat mengikuti tren pergerakan harga saham dengan baik, tetapi model LSTM terlihat lebih akurat dibandingkan RNN, karena prediksinya lebih mendekati data aktual. Model RNN cenderung memiliki sedikit deviasi dari harga sebenarnya, terutama pada puncak dan penurunan harga yang tajam. Hal ini menunjukkan bahwa model LSTM lebih mampu menangkap pola pergerakan saham dengan lebih baik, sehingga lebih direkomendasikan untuk prediksi harga saham Bank BNI dibandingkan RNN.



Gambar 4. 24 Grafik Perbandingan Model RNN dan LSTM Bank BRI

Pada grafik tersebut kedua model dapat mengikuti pola pergerakan harga saham dengan baik. Namun, model LSTM tampak lebih akurat dibandingkan RNN, karena hasil prediksinya lebih mendekati data asli. Model RNN cenderung memiliki sedikit penyimpangan dari harga sebenarnya, terutama pada pergerakan harga yang lebih ekstrem. Dengan demikian, model LSTM lebih disarankan untuk prediksi harga saham Bank BCA karena memiliki performa yang lebih baik dalam menangkap pola pergerakan harga dibandingkan RNN.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perbandingan metode RNN dan LSTM dalam memprediksi harga saham bank terbesar di Indonesia, dapat disimpulkan bahwa model LSTM memiliki performa yang lebih baik dibandingkan RNN. Hal ini terlihat dari nilai *Root Mean Square Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE) yang lebih rendah pada model LSTM dibandingkan RNN, menunjukkan bahwa prediksi LSTM lebih akurat dan lebih dekat dengan data asli. Selain itu, dari visualisasi data saham Bank Mandiri, BRI, BNI, dan BCA, model LSTM lebih mampu mengikuti pola tren harga saham dengan lebih stabil, sementara model RNN cenderung mengalami penyimpangan yang lebih besar, terutama pada fluktuasi harga yang ekstrem. Oleh karena itu, dalam konteks prediksi harga saham, metode LSTM lebih direkomendasikan karena kemampuannya dalam menangani dependensi jangka panjang dan menangkap pola kompleks dalam pergerakan harga saham.

5.2 Saran

Berdasarkan hasil penelitian, disarankan untuk menggunakan model LSTM dalam aplikasi prediksi harga saham di Indonesia, terutama untuk memprediksi harga saham yang memiliki ketergantungan jangka panjang dan fluktuasi yang kompleks. LSTM dapat memberikan hasil yang lebih stabil dan akurat dibandingkan dengan RNN. Untuk meningkatkan akurasi lebih lanjut, disarankan untuk melatih model dengan dataset yang lebih besar dan lebih bervariasi, mencakup berbagai faktor ekonomi dan pasar yang dapat memengaruhi harga saham. Dengan data yang lebih beragam, model dapat menangkap pola yang lebih kompleks dan memberikan prediksi yang lebih baik.

DAFTAR PUSTAKA

- Aji Riyantoko, P., Maulana Fahrudin, T., Maulida Hindrayani, K., & Maya Safitri, E. (2020). Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (LSTM). *Seminar Nasional Informatika (SEMNASIF)*, 1(1), 427–435.
- Alkahfi, C., Kurnia, A., & Saefuddin, A. (2024). *MALCOM: Indonesian Journal of Machine Learning and Computer Science Performance Comparison of RNN-Based Models in Forecasting Indonesian Economic and Financial Data Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan Keuangan Indonesia*. 4(October), 1235–1243.
- Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jensen, R. (2017). *An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting*. 1–41. <https://doi.org/10.1007/978-3-319-70338-1>
- Bode, A. (2017). K-Nearest Neighbor Dengan Feature Selection Menggunakan Backward Elimination Untuk Prediksi Harga Komoditi Kopi Arabika. *ILKOM Jurnal Ilmiah*, 9(2), 188–195. <https://doi.org/10.33096/ilkom.v9i2.139.188-195>
- Budiprasetyo Gunawan, Hani'ah Mumluatul, & Aflah Zahira Darin. (2022). Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 08(135).
- Bursa Efek Indonesia, 2024. 14 Juta Investor Pasar Modal: Sinergi, Inovasi Digital, dan Akses Informasi yang Kian Inklusif. Bursa Efek Indonesia. Tersedia pada: <https://www.idx.co.id>
- Erwin, E., Tinggi, S., Ekonomi, I., Makassar, C., Chatra, A., Stie, P., & Kerinci, S. A. (2023). *Transformasi Digital*. <https://www.researchgate.net/publication/379374858>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

- I Nyoman Cerdas Janastu, & Dhoriva Urwatul Wutsqa. (2024). Prediksi Harga Saham Pada Sektor Perbankan Menggunakan Algoritma Long Short-Term Memory. *Jurnal Statistika Dan Sains Data*, 1(2), 1–14.
- Kiramy, R. Al, Permana, I., & Marsal, A. (2024). *Comparison of RNN and LSTM Algorithm Performance in Predicting the Number of Umrah Pilgrims at PT . Hajar Aswad Perbandingan Performa Algoritma RNN dan LSTM dalam Prediksi Jumlah Jamaah Umrah pada PT . Hajar Aswad*. 4(October), 1224–1234.
- Luchia, N. T., Tasia, E., Ramadhani, I., Rahmadeyan, A., & Zahra, R. (2024). Performance Comparison Between Artificial Neural Network, Recurrent Neural Network and Long Short-Term Memory for Prediction of Extreme Climate Change. *Public Research Journal of Engineering, Data Technology and Computer Science*, 1(2), 62–70. <https://doi.org/10.57152/predatecs.v1i2.864>
- Maricar, M. A., Widiadnyana, P., & Arta Wijaya, I. W. (2017). Analysis of Data Mining for Forecasting Total Goods Delivery with Moving Average Method. *International Journal of Engineering and Emerging Technology*, 2(1), 7. <https://doi.org/10.24843/ijeet.2017.v02.i01.p02>
- Murphy, J. J. (1999). Technical analysis of the financial markets. In *Pennsylvania Dental Journal* (Vol. 77, Issue 2).
- Nilsen, A. (2022). Perbandingan Model RNN, Model LSTM, dan Model GRU dalam Memprediksi Harga Saham-Saham LQ45. *Jurnal Statistika Dan Aplikasinya*, 6(1), 137–147. <https://doi.org/10.21009/jsa.06113>
- Rezk, N. M., Purnaprajna, M., Nordstrom, T., & Ul-Abdin, Z. (2020). Recurrent Neural Networks: An Embedded Computing Perspective. *IEEE Access*, 8, 57967–57996. <https://doi.org/10.1109/ACCESS.2020.2982416>
- Rizki, M., Basuki, S., & Azhar, Y. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang. *Jurnal Repositor*, 2(3), 331–338. <https://doi.org/10.22219/repositor.v2i3.470>

- Roni Merdiansah, Khofifah Wulandari, Mentari Hasibuan, & Yuyun Umidah. (2024). Perbandingan Kinerja Model RNN, LSTM, dan BLSTM dalam Memprediksi Jumlah Gempa Bulanan di Indonesia. *Jurnal Penelitian Rumpun Ilmu Teknik*, 3(1), 262–277. <https://doi.org/10.55606/juprit.v3i1.3466>
- Rosyd, A., Irma Purnamasari, A., & Ali, I. (2024). Penerapan Metode Long Short Term Memory (Lstm) Dalam Memprediksi Harga Saham Pt Bank Central Asia. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 501–506. <https://doi.org/10.36040/jati.v8i1.8440>
- Sari, L., & Septiano, R. (2023). Modal Meningkatkan Harga Saham Perbankan di Indonesia. *Jurnal Revenue : Jurnal Ilmiah Akuntansi*, 4(1), 33–39.
- Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Selle, N., Yudistira, N., & Dewi, C. (2022). Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), 155–162. <https://doi.org/10.25126/jtiik.2022915585>
- Sitanggang, M., & Hidayat, P. (2013). Analisis Kausalitas Antara Volatilitas Saham Dengan Variabel Makroekonomi Indonesia. *Jurnal Ekonomi Dan Keuangan*, 1(5), 14744.
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), 184–196.
- Yendriani, D., Jondri, J., & ... (2015). Prediksi Harga Saham Menggunakan Hidden Markov Model (HMM) dan Fuzzy Model. *EProceedings ...*, 2(2), 6592–6599.