

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

**PROPOSAL SKRIPSPI**

**OLEH:**

**EVINDO AMANDA RIZA**

**218160018**



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

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

**PROPOSAL 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  
2024**

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
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
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
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 .....	16
2.7. Parameter Evaluasi .....	16
2.7.1. <i>Mean Square Error</i> (MSE) .....	16
2.7.2. <i>Root Mean Square Error</i> (RMSE) .....	17
2.7.3. <i>Mean Absolute Error</i> (MAE) .....	17
2.8. <i>Flowchart Diagram</i> .....	18
2.9. <i>Unified Modeling Language</i> (UML) .....	19
2.9.1. <i>Use Case Diagram</i> .....	19
2.9.2. <i>Activity Diagram</i> .....	20
2.10. Penelitian Terdahulu.....	21
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>24</b>
3.1. Perangkat Penelitian .....	24
3.2. Tahapan Penelitian .....	24
3.2.1.     Pengumpulan Data .....	25
3.2.2. <i>Preprocessing</i> Data .....	27
3.2.3.     Implementasi Model RNN & LSTM .....	29
3.2.4.     Pelatihan dan Pengujian Model.....	31
3.2.5.     Evaluasi Model.....	37
3.3. Tahapan Perancangan.....	40
3.3.1. <i>Unified Modeling Language</i> (UML).....	40
3.3.2. <i>User Interface</i> (UI).....	42
<b>BAB IV HASIL &amp; PEMBAHASAN.....</b>	<b>43</b>
4.1. Hasil.....	43

4.1.1. Tahap Pengumpulan Data .....	<b>Error! Bookmark not defined.</b>
4.1.2. Tahap <i>Preprocessing</i> .....	<b>Error! Bookmark not defined.</b>
4.1.3. Tahap Implementasi Model RNN & LSTM .....	45
4.1.4. Tahap Pelatihan & Pengujian Model .....	45
4.1.5. Tahap Evaluasi Model .....	<b>Error! Bookmark not defined.</b>
4.1.6. Menampilkan Proses Prediksi Pada Aplikasi Web.....	46
4.2. Pembahasan .....	47
4.2.1. .....	47
4.2.2. .....	47
4.2.3. Hyperparameter Tuning RandomizedSearchCV .....	47
4.2.4. .....	47
<b>DAFTAR PUSTAKA</b> .....	<b>50</b>

## DAFTAR TABEL

	Halaman
<b>Tabel 2.1.</b> Komponen <i>Flowchart Diagram</i> .....	18
<b>Tabel 2.2.</b> Komponen <i>Use Case Diagram</i> .....	19
<b>Tabel 2.3.</b> Komponen <i>Activity Diagram</i> .....	20
<b>Tabel 2.4.</b> Penelitian Terdahulu.....	21
<b>Tabel 3.1.</b> <i>Dataset Saham BBRI</i> .....	26
<b>Tabel 3.2.</b> Hasil Normalisasi <i>Close Price Saham BBRI</i> .....	28
<b>Tabel 3.3.</b> Pembentukan <i>Sliding Window Close Price Saham BBRI</i> .....	29
<b>Tabel 3.4.</b> <i>Hyperparameter</i> Yang Digunakan .....	30
<b>Tabel 3.5.</b> Hasil Perhitungan RNN.....	33
<b>Tabel 3.6.</b> Hasil Perhitungan LSTM .....	35
<b>Tabel 3.7.</b> Nilai Yang Digunakan Pada Prediksi RNN dan LSTM .....	37
<b>Tabel 3.8.</b> Hasil Denormalisasi RNN dan LSTM.....	38
<b>Tabel 3.9.</b> Hasil Evaluasi Metrik RNN dan LSTM .....	40

## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1.</b> Arsitektur <i>Looping</i> pada RNN.....	7
<b>Gambar 2.2.</b> Arsitektur <i>Looping</i> pada LSTM .....	10
<b>Gambar 2.3.</b> Alur <i>Cell State</i> pada LSTM .....	10
<b>Gambar 2.4.</b> Alur <i>Forget Gate</i> pada LSTM .....	11
<b>Gambar 2.5.</b> Alur <i>Input Gate</i> pada LSTM .....	12
<b>Gambar 2.6.</b> Alur memperbarui <i>Cell State</i> pada LSTM.....	13
<b>Gambar 2.7.</b> Alur <i>Output Gate</i> pada LSTM.....	14
<b>Gambar 3.1.</b> <i>Flowchart</i> Tahapan Penelitian.....	25
<b>Gambar 3.2.</b> <i>Use Case Diagram</i> .....	40
<b>Gambar 3.3.</b> <i>Activity Diagram</i> Pelatihan Model Terbaik .....	41
<b>Gambar 3.4.</b> <i>Activity Diagram</i> Menampilkan Hasil Prediksi.....	41
<b>Gambar 3.5.</b> Desain UI Halaman Aplikasi Web .....	42

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Saham merupakan instrumen keuangan 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, kurs mata uang, ekonomi makro, tren industri, dan sentimen pasar. Prediksi harga saham menjadi sangat sulit karena volatilitas pasar yang tinggi dan pola data yang kompleks (Janastu & Wutsqa, 2024).

Prediksi harga saham yang akurat dengan strategi yang tepat dapat memberikan keuntungan yang signifikan bagi investor, trader dan analis keuangan, serta menghindari risiko kerugian, baik dalam pengambilan keputusan membeli, menahan, atau menjual saham. Investor harus melakukan analisis karena data harga saham berfluktuasi sepanjang waktu, sehingga tidak ada saham yang terus naik atau turun. Beberapa pola dihasilkan oleh kenaikan dan penurunan yang bergantian seiring berjalannya waktu. Oleh karena itu, untuk mengoptimalkan keuntungan sebagai tujuan utama investor di pasar modal, diperlukan prediksi pergerakan harga saham yang akurat. Metode yang tepat untuk melakukan analisis adalah dengan menggunakan teknik peramalan atau prediksi (Faridah & Sugiantoro, 2023).

Perkembangan kecerdasan buatan (*Artificial Intelligence*), telah menjadi pilihan utama sebagai metode yang sering digunakan untuk meningkatkan prediksi saham. Penelitian tentang prediksi saham saat ini lebih banyak menggunakan pendekatan dengan teknik *deep learning* (Pipin dkk., 2023). *Recurrent Neural Network* (RNN) merupakan algoritma dalam *deep learning* yang memiliki tingkat keakuratan tinggi dalam melakukan prediksi data berbentuk *time series*. Sementara itu dalam RNN juga terdapat algoritma lain yang telah dikembangkan seperti algoritma *Long Short-Term Memory* (LSTM) yang merupakan bagian dari keluarga RNN. LSTM dikembangkan sebagai penyempurnaan arsitektur RNN dengan menambahkan *memory cell* agar dapat menyimpan informasi dalam waktu yang lebih lama serta mengatasi terjadinya *vanishing gradient* pada RNN saat memproses data sekuensial yang panjang (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 *deep learning* khususnya RNN dan LSTM yang masing-masing menghasilkan akurasi yang baik, tetapi belum banyak penelitian yang secara langsung membandingkan performa dan kinerja antara keduanya. Sebagian besar penelitian hanya menggunakan satu jenis model tanpa mempertimbangkan perbandingan performa antara RNN dan LSTM pada dataset yang sama dalam konteks prediksi harga saham. Oleh karena itu, penelitian ini dilakukan untuk menentukan model yang lebih unggul berdasarkan analisis terhadap metrik evaluasi seperti RMSE, MSE dan MAE dalam mengevaluasi performa model.

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 di atas, dapat dirumuskan masalah penelitian sebagai berikut: 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 saham diperoleh dari website Yahoo Finance.
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 dalam pengembangan ilmu pengetahuan, khususnya penerapan RNN dan LSTM dalam prediksi harga saham, serta menjadi referensi bagi penelitian lanjutan di bidang kecerdasan buatan.
2. Mendukung pengembangan kecerdasan buatan (AI) 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.

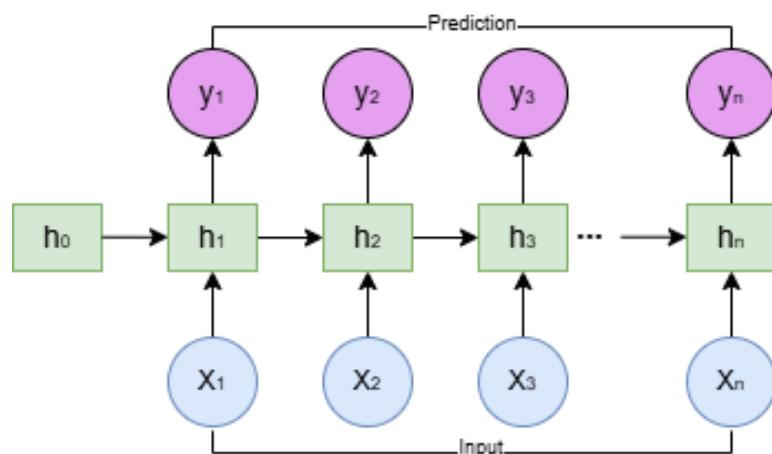
## **2.2. Prediksi Harga Saham**

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 sentimen (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 (Rosyid dkk., 2024).

### 2.3. Recurrent Neural Network (RNN)

RNN adalah salah satu jenis *Artificial Neural Network* (ANN) yang dirancang untuk memproses data sekuensial (berurutan), sehingga cocok digunakan untuk menangani data *time series* (runtun waktu). RNN memiliki struktur yang terdiri dari *input layer*, *hidden layer*, dan *output layer*, dengan aliran informasi yang mempertahankan hubungan dengan *hidden layer* sebelumnya. Output dari *hidden layer sebelumnya* digunakan sebagai input tambahan untuk *hidden layer* saat ini, (Wardani, 2021).



**Gambar 2.1.** Arsitektur Looping pada RNN

(Sumber: Mienye dkk., 2024)

Dalam proses pelatihan, RNN menggunakan algoritma *Backpropagation Through Time* (BPTT) yang menjumlahkan kesalahan di setiap *time step* 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.

$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh} \mathbf{x}_t + \mathbf{W}_{hh} \mathbf{h}_{t-1} + \mathbf{b}_h) \quad (2.1)$$

$$\mathbf{y}_t = \tanh(\mathbf{W}_{hy} \mathbf{h}_t + \mathbf{b}_y) \quad (2.2)$$

Dimana:

$\mathbf{h}_t$  : *hidden state*

$\mathbf{W}_{xh}, \mathbf{W}_{hh}, \mathbf{W}_{hy}$  : matriks bobot di setiap layer

$\mathbf{b}_h, \mathbf{b}_y$  : vector bias

$\mathbf{x}_t$  dan  $\mathbf{y}_t$  : input dan output pada waktu ke -t

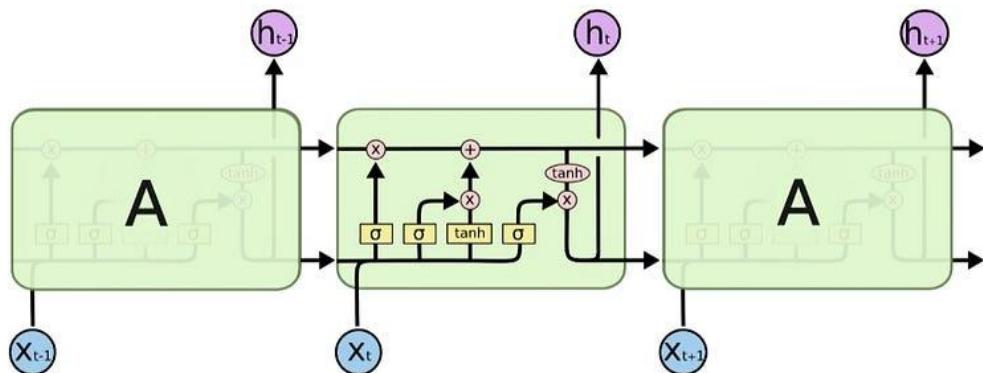
$\tanh$  : fungsi aktivasi layer

#### **2.4. Long Short-Term Memory (LSTM)**

LSTM adalah 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).

Berbeda dengan RNN, LSTM tidak memiliki kelemahan ini karena mampu mengatur memori pada setiap input dengan menggunakan *memory cells* dan *gate units* (Arfan & Lussiana ETP, 2019). Terdapat 3 komponen gate utama dalam LSTM yaitu *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. Dengan alasan ini, LSTM merupakan algoritma RNN terbaik untuk aplikasi data sekuenzial dan banyak digunakan untuk pemrosesan teks, video, dan data *time series* lainnya. (Kusuma, 2024).

Pada dasarnya LSTM memiliki struktur yang sama dengan RNN. Perbedaannya terletak pada susunan jaringan pada *hidden layer*. RNN hanya terdiri dari satu layer sederhana seperti fungsi aktivasi *tanh*. Pada LSTM, terdapat dua fungsi aktivasi, yaitu fungsi *sigmoid* dan *tanh* serta dilengkapi dengan *memory cell* dan *gates* (Wardani, 2021).

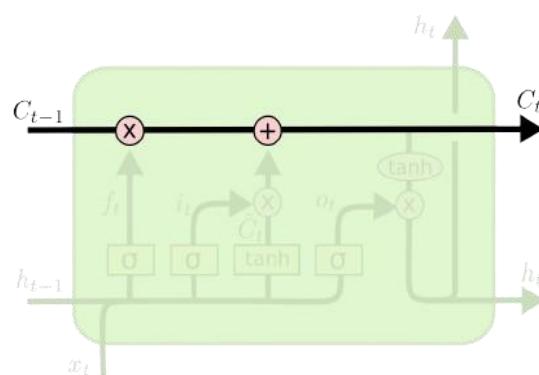


**Gambar 2.2.** Arsitektur *Looping* pada LSTM

(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:

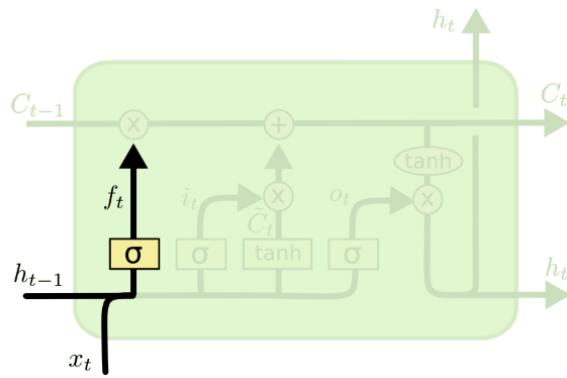


**Gambar 2.3.** Alur *Cell State* pada LSTM

(Sumber: Olah, Christopher)

Langkah pertama, memutuskan informasi harus disimpan atau dibuang dari *cell state* dengan menggunakan *gates* yang disebut *forget gate*. *Forget gate* akan memproses informasi berupa *hidden state* sebelumnya ( $\mathbf{h}_{t-1}$ ) dan informasi baru dari input saat ini ( $\mathbf{x}_t$ ), kemudian informasi tersebut akan digabung dan diproses

dengan menggunakan fungsi *sigmoid* dan menghasilkan output pada *cell state* berupa angka 1 menunjukkan bahwa informasi harus disimpan, sedangkan angka 0 menunjukkan informasi sudah tidak dibutuhkan lagi sehingga boleh dibuang (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:

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

Dimana:

$f_t$  : *forget gate*

$\sigma$  : fungsi *sigmoid*

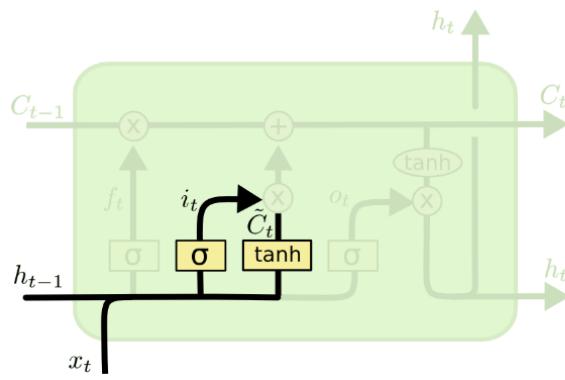
$W_f$  : nilai *weight* untuk *forget gate*

$h_{t-1}$  : nilai output sebelum orde ke-t

$x_t$  : nilai input pada orde ke-t

$b_f$  : nilai bias pada *forget gate*

Langkah kedua, menentukan informasi apa yang disimpan ke dalam *cell state*. Pada langkah ini terdiri dari dua bagian, yaitu *gates* yang bernama *input gate* dan lapisan *tanh*. Pertama, pada lapisan *input gate* memutuskan nilai yang akan diperbarui. Kedua, lapisan *tanh* akan membuat satu kandidat dengan nilai baru  $\tilde{C}_t$  yang dapat di tambahkan ke *cell state*. 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 2.5 berikut:

$$i_t = \sigma (\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (2.5)$$

Dimana:

$i_t$  : *input gate*

$\sigma$  : fungsi *sigmoid*

$\mathbf{W}_i$  : nilai *weight* untuk *input gate*

$\mathbf{h}_{t-1}$  : nilai output sebelum orde ke-t

$\mathbf{x}_t$  : nilai input pada orde ke-t

$\mathbf{b}_i$  : nilai bias pada *input gate*

Nilai kandidat baru dirumuskan pada persamaan 2.6 berikut:

$$\bar{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.6)$$

Dimana:

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

**tanh** : fungsi *tanh*

$W_c$  : nilai *weight* untuk *cell state*

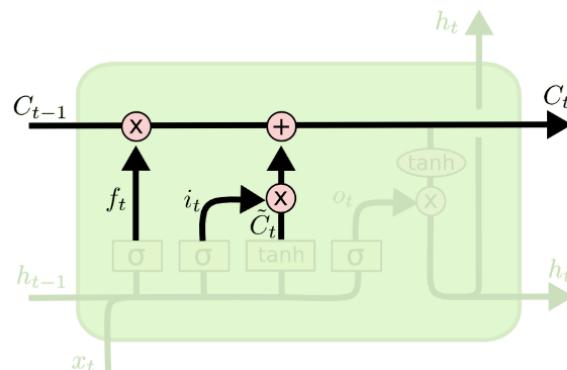
$h_{t-1}$  : nilai output sebelum orde ke-t

$x_t$  : nilai input pada orde ke-t

$b_c$  : nilai bias pada *cell state*

Langkah ketiga, memperbarui *cell state* lama,  $C_{t-1}$  ke dalam *cell state* baru

$C_t$ . Langkahnya yaitu dengan mengalikan state lama  $C_{t-1}$  dengan  $f_t$ , untuk membuang informasi yang sudah ditentukan sebelumnya pada *forget gate*, setelah itu ditambahkan dengan  $i_t * \bar{C}_t$ , yang merupakan nilai baru dan digunakan untuk memperbarui state (Syahrul, 2023). Alur memperbaharui *cell state* ditunjukkan pada Gambar 2.6 berikut:



**Gambar 2.6.** Alur memperbarui *Cell State* pada LSTM

(Sumber: Olah, Christopher)

Nilai *Cell State* diurumuskan pada persamaan 2.7 berikut:

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (2.7)$$

Dimana:

$C_t$  : cell state

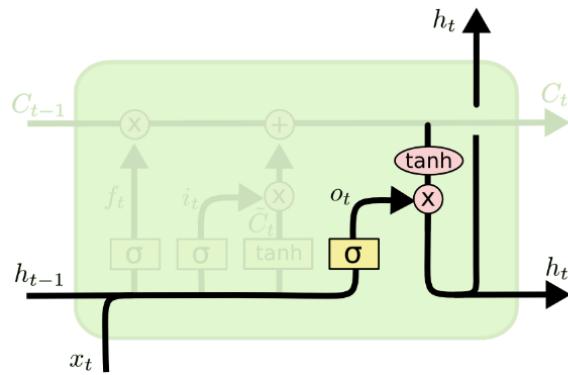
$f_t$  : forget gate

$C_{t-1}$  : cell state sebelum orde ke-t

$i_t$  : input gate

$\bar{C}_t$  : nilai baru yang dapat 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 2.8 berikut:

$$O_t = \sigma(W_o \cdot [h_{t-1} x_t] + b_o) \quad (2.8)$$

Dimana:

$O_t$  : output gate

$\sigma$  : fungsi *sigmoid*

$W_o$  : nilai *weight* untuk *output gate*

$h_{t-1}$  : nilai output sebelum orde ke-t

$x_t$  : nilai input pada orde ke-t

$b_o$  : nilai bias pada *output gate*

Output *cell state* dimasukkan kedalam lapisan *tanh* yang mengubah nilai menjadi range -1 dan 1 (Syahrul, 2023). Nilai output orde-t diurumuskan menggunakan persamaan 2.9 berikut:

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{c}_t) \quad (2.9)$$

Dimana:

$\mathbf{h}_t$  : nilai output orde-t

$\mathbf{o}_t$  : *output gate*

**tanh** : fungsi *tanh*

$\mathbf{c}_t$  : *cell state*

## 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. eknik 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 2.10 berikut:

$$\mathbf{x}_t = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (2.10)$$

Dimana:

$\mathbf{x}_t$  : 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 2.11 berikut:

$$x_t = x(x_{max} - x_{min}) + x_{min} \quad (2.11)$$

Dimana:

$x_t$  : 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 seperti berikut ini.

### 2.7.1. *Mean Square 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. Nilai yang lebih rendah mengindikasikan bahwa prediksi model lebih mendekati harga aktual, mencerminkan kinerja yang lebih baik (Pipin dkk., 2023) dirumuskan menggunakan persamaan 2.12 berikut:

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

### 2.7.2. Root Mean Square 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 2.13 berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (2.13)$$

### 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 2.14 berikut:

$$MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n} \quad (2.14)$$

Dimana:

$Y_i$  : 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/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.untuk diproses untuk memberikan awal harga.
	<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 berbagai 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		Swimlane	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

Penelitian terdahulu adalah penelitian yang telah dilakukan oleh peneliti sebelumnya terkait dengan penelitian yang akan dilakukan oleh penulis saat ini. Hasil penelitian sebelumnya tentang prediksi harga saham menggunakan RNN dan LSTM ditunjukkan sebagai berikut.

**Tabel 2.4.** Penelitian Terdahulu

No	Peneliti (Tahun)	Judul	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 time series dengan akurasi prediksi hingga 95% dan loss 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 timestep dan epoch, untuk mengoptimalkan akurasi. Model univariate menghasilkan MAPE 1.546%, sedangkan multivariate (dengan kurs rupiah) mencapai

			1.558%. Hasil menunjukkan model univariate 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 Central Asia Tbk, Dengan Menggunakan Algoritma <i>Long Short-Term Memory</i> (LSTM)	Penelitian ini memprediksi tren harga saham PT Bank Central Asia Tbk (BBCA) menggunakan algoritma LSTM dengan data dari 2011 hingga 2022. Model diuji dengan variasi epoch, batch size 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 error prediksi masih dalam batas akurasi yang memadai.
5	(Pan, 2024)	Stock Price Prediction System LSTM Based on Deep Learning	Penelitian ini menggunakan model LSTM untuk memprediksi harga saham dari empat perusahaan teknologi besar: Apple (AAPL), Tesla (TSLA), Microsoft (MSFT),

			<p>dan Nvidia (NVDA). Hasilnya menunjukkan bahwa LSTM unggul dibandingkan metode tradisional seperti Linear Regression, Support Vector Regression, Decision Tree Regression, dan Gradient Boosting Decision Trees dalam hal akurasi prediksi berdasarkan nilai Mean Squared Error (MSE). LSTM secara konsisten menghasilkan nilai MSE terendah untuk semua saham, menunjukkan kemampuan prediksi yang lebih baik dibandingkan metode lainnya.</p>
6	(Alim, 2023)	Pemodelan Time Series Data Saham LQ45 dengan Algoritma LSTM, RNN, dan ARIMA	<p>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.</p>

## **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*):

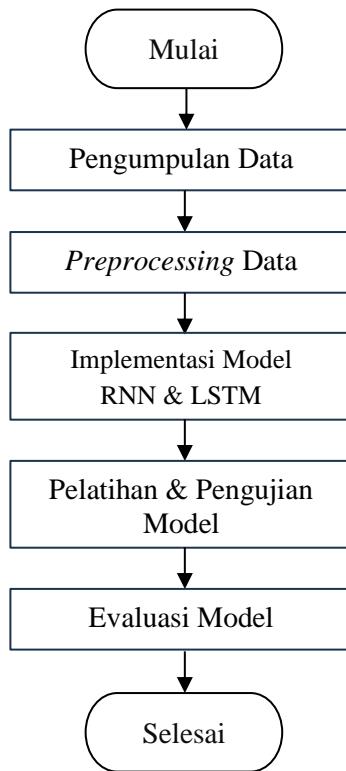
- a. Device : Lenovo LOQ 15IRX9
- b. Processor : Intel® Core™ i5-13450HX CPU @ 2.4GHz
- c. GPU : NVIDIA GeForce RTX 3050 6GB
- d. RAM : 20 GB
- e. SSD : 512 GB

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 merupakan dataset sekunder dari saham perbankan PT Bank Rakyat Indonesia Tbk, dengan kode saham BBRI yang diperoleh menggunakan API (*Application Programming Interface*) dari Yahoo Finance dengan laman: <https://finance.yahoo.com/quote/BBRI.JK/history/>.

Berdasarkan Forbes 2000 The World Biggest Companies yang memuat lebih dari 2000 perusahaan publik terbesar di dunia salah satunya terdapat bank di Indonesia dibawah BUMN yang masuk didalamnya yaitu, Bank Rakyat Indonesia (BRI). Bank Rakyat Indonesia merupakan salah satu perbankan yang diminati investor yang mempunyai kapitalisasi pasar sangat besar dan likuiditas yang tinggi sehingga tidak mudah dimanipulasi dan terjadi lonjakan harga yang signifikan.

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

### **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 dan melakukan penghapusan atau menggantinya dengan nilai *mean* atau nilai rata-rata.

#### **b. Pembagian Data**

Penulis membagi data menjadi tiga bagian, yaitu *training*, *validation* dan *testing*. Data *training* digunakan untuk membangun model dengan mempelajari pola dari data historis, *validation* digunakan untuk menyesuaikan parameter model dan mencegah *overfitting* dan *underfitting* selama proses pelatihan dan data *testing* digunakan untuk menguji performa model pada data baru yang belum pernah digunakan dalam pelatihan. Dalam penelitian ini, data dibagi dengan persentase 70% *training*, 15% *validation* dan 15% *testing*.

#### **c. Normalisasi**

Pada tahap ini, setelah data dibagi, dilakukan normalisasi pada data *training*, normalisasi hanya menggunakan *min* dan *max* dari *training set*, lalu menerapkan skala yang sama pada *validation* dan *testing*. Penelitian ini berfokus pada prediksi harga saham BBRI dengan menggunakan harga penutupan (*Close*). Berikut merupakan contoh perhitungan normalisasi pada data *training* menggunakan persamaan 2.10.

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 *Close Price* dapat dilihat pada tabel 3.2 berikut.

**Tabel 3.2.** Hasil Normalisasi *Close Price* Saham 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
6.	11/01/2010	403.20	0.0112
...	...	...	...

#### d. Pembentukan *Sliding Window*

Pada tahap ini, data *time series* dibentuk dengan pendekatan *sliding window* agar dapat diterima dan diproses oleh model RNN dan LSTM, di mana nilai harga saham sebelumnya digunakan sebagai input untuk memprediksi harga saham pada hari berikutnya. Dengan demikian, model akan belajar mengenali pola historis pergerakan harga saham dan membuat prediksi berdasarkan pola tersebut.

*Window size* atau *timestep* ditentukan sebanyak 5 pada contoh ini, berarti model menggunakan data 5 hari terakhir sebagai variabel input yaitu ( $X_1$  -  $X_5$ ) dan satu target output (Y), yaitu harga saham pada hari ke-6 atau harga pada hari berikutnya. Pembentukan teknik *sliding window* ini mengambil nilai normalisasi data pada Tabel 3.2 yang dapat dilihat pada Tabel 3.3. berikut.

**Tabel 3.3.** Pembentukan *Sliding Window Close Price Saham 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. Implementasi Model RNN & LSTM

Setelah pembentukan *sliding window*, dilakukan perancangan model RNN dan LSTM dengan framework Tensorflow, termasuk penentuan *hyperparameter* yang akan memengaruhi kinerja model selama pelatihan. *Hyperparameter* merupakan parameter yang ditetapkan oleh pengguna sebelum pelatihan dan tidak diperbarui selama proses pembelajaran, seperti bobot dan bias yang diperbarui selama pelatihan. Untuk memperoleh kombinasi terbaik, penelitian ini menerapkan teknik *hyperparameter tuning* menggunakan metode *RandomSearchCV*. Berikut merupakan daftar *hyperparameter* yang digunakan dalam penelitian ini beserta penjelasannya yang dapat dilihat pada Tabel 3.4.

**Tabel 3.4.** Hyperparameter Yang Digunakan

Hyperparameter	Keterangan
<b>Timesteps</b>	Jumlah langkah waktu dalam satu sekuens input yang diproses dalam satu waktu. Peneliti menentukan nilai <i>timesteps</i> 30 hari perdagangan sebelumnya untuk menentukan prediksi harga saham berikutnya.
<b>{Model}_Layers</b>	Menentukan berapa banyak layer RNN dan LSTM. Peneliti menentukan 1 - 2 layer.
<b>{Model}_Units</b>	Menentukan berapa banyak <i>neuron</i> atau unit RNN dan LSTM. Peneliti menentukan jumlah neuron sebanyak 64, 128 dan 256 pada tiap layer.
<b>{Model}_Dropout</b>	Teknik regulasi yang menonaktifkan <i>neuron</i> selama proses pelatihan untuk menghindari <i>overfitting</i> . Peneliti menentukan <i>dropout</i> sebesar 0.2 dan 0.3 sudah cukup.
<b>Epochs</b>	Menentukan berapa kali seluruh dataset latih diproses atau model melihat seluruh dataset latih. Peneliti menentukan maksimal <i>epoch</i> sebanyak 100.
<b>Batch_Size</b>	Menentukan berapa banyak sampel yang diproses dalam satu waktu, peneliti menentukan nilai sebesar 64 pada <i>batch size</i> .
<b>Optimizer &amp; Learning Rate</b>	<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.
<b>Loss</b>	Menghitung jarak 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. Pelatihan dan Pengujian Model

Setelah membuat model, tahap berikutnya adalah melatih model menggunakan data pelatihan. Model yang telah terlatih dengan performa yang baik akan dievaluasi menggunakan data uji untuk mengukur akurasi prediksi dengan metrik yang sudah ditentukan.

Proses pelatihan RNN dan LSTM melibatkan dua tahap yaitu *Forward Propagation Through Time* (FPTT) dan *Backpropagation Through Time* (BPTT).

#### 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. diatas. Data pertama yang digunakan sebagai input adalah data yang telah dinormalisasi dibawah, ditunjukkan dalam bentuk representasi pada Tabel 3.4. untuk memudahkan pembaca.

<b>X<sub>1</sub></b>	<b>X<sub>2</sub></b>	<b>X<sub>3</sub></b>	<b>X<sub>4</sub></b>	<b>X<sub>5</sub></b>	<b>Target (Y)</b>
0.0130	0.0140	0.0121	0.0103	0.0112	0.0112

Bobot (*weight*) dan bias diinisialisasikan sebagai berikut:

Nilai bobot (*weight*) dirumuskan dengan :

$$W = \left( -\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}} \right) = \left( -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}} \right) = (-0,447, 0,447)$$

dengan:

$$W = \text{bobot } (\textit{weight})$$

$$d = \text{jumlah variabel}$$

Sehingga nilai bobot dan bias RNN yang digunakan adalah sebagai berikut:

$$\text{Bobot Input } (W_{xh}) : 0.447$$

$$\text{Bobot Hidden } (W_{hh}) : 0.447$$

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:

$$\begin{aligned} h_t &= \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \\ y_t &= \tanh(W_{hy}h_t + b_y) \end{aligned}$$

**Perhitungan untuk timestep pertama ( $t_1$ ):**

$$\begin{aligned} h_1 &= \tanh(0.447 \times 0.0130 + 0.447 \times 0 + 0) \\ &= \tanh(0.005811) \\ &= 0.0058 \end{aligned}$$

$$\begin{aligned} y_1 &= \tanh(0.447 \times 0.0058 + 0) \\ &= \tanh(0.002593) \\ &= 0.0026 \end{aligned}$$

**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 3.5. berikut.

**Tabel 3.5.** Hasil Perhitungan RNN

RNN	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$x_t$	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. Backpropagation Through Time (BPTT) Pada RNN

- a. Perhitungan *Loss*:

$$\text{Target output}(Y) = 0.0112$$

$$\text{Prediksi output } (y_5) = 0.0040$$

Fungsi *loss* yang digunakan adalah Mean Squared Error (MSE):

$$\begin{aligned} L &= \frac{1}{2} (y_{pred} - y_{target})^2 \\ &= \frac{1}{2} (0.0040 - 0.0112)^2 \\ &= 0.00002592 \end{aligned}$$

- b. 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 optimasi seperti Adam, SGD atau RMSProp. Proses ini dilakukan untuk seluruh *timestep* dengan propagasi *error* kebelakang hingga *timestep* pertama.

## 2. 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 dengan pada RNN diatas, ditunjukkan dalam bentuk representasi tabel pada Tabel 3.4 untuk memudahkan pembaca.

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	Target (Y)
0.0130	0.0140	0.0121	0.0103	0.0112	0.0112

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 ( $C_t$ ):**

$$\begin{aligned}
 C_t &= f_t * C_{t-1} + i_t * \bar{C}_t \\
 &= (0.7322 \times 0) + (0.5015 \times 0.0058) \\
 &= 0 + 0.0029 = 0.0029
 \end{aligned}$$

**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. sehingga nilai dari tiap proses perhitungan LSTM di setiap *gates* didapatkan seperti pada Tabel 3.6. berikut.

**Tabel 3.6.** Hasil Perhitungan LSTM

LSTM	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$x_t$	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

## 2. Backpropagation Through Time (BPTT) Pada LSTM

a. Perhitungan *Loss*:

$$\begin{aligned}
 \text{Target output}(Y) &= 0.0112 \\
 \text{Prediksi output } (h_5) &= 0.0050
 \end{aligned}$$

Fungsi *loss* yang digunakan adalah Mean Squared Error (MSE):

$$\begin{aligned} L &= \frac{1}{2}(y_{pred} - y_{target})^2 \\ &= \frac{1}{2}(0.0050 - 0.0112)^2 \\ &= 0.00001922 \end{aligned}$$

- b. *Backward propagation* 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 disebut Backpropagation Through Time (BPTT) karena melibatkan propagasi *error* ke *timestep* sebelumnya untuk memperbarui parameter model secara bertahap menggunakan algoritma optimasi, seperti Adam, SGD atau RMSProp.

Tujuan pelatihan model yaitu menemukan bobot optimal yang dapat meminimalkan *error*. Pelatihan model mencakup langkah-langkah seperti *forward propagation*, perhitungan *loss*, dan *backward propagation* pada setiap *timesteps* dalam RNN dan LSTM, yang diulang secara iteratif hingga model mencapai konvergensi dan ditandai dengan perubahan nilai *loss* yang kecil dan prediksi yang stabil. Proses optimasi dilakukan menggunakan optimizer untuk mempercepat konvergensi. Proses ini berlangsung selama *n-epoch*, dengan evaluasi *error* pada setiap *epoch* untuk mencegah *overfitting* atau *underfitting*. Setelah pelatihan selesai, model diuji menggunakan *testing* set, yaitu data yang tidak digunakan selama pelatihan, guna mengevaluasi kinerjanya.

### 3.2.5. Evaluasi Model

Setelah data input diproses oleh RNN dan LSTM, kemudian data tersebut perlu dilakukan denormalisasi agar dapat dibandingkan dengan data aktual, karena sebelumnya data melalui proses normalisasi. Nilai yang digunakan dalam proses denormalisasi untuk mendapatkan nilai prediksi pada model adalah nilai ( $y_t$ ) pada RNN dan nilai ( $h_t$ ) pada LSTM yang ditunjukkan pada Tabel 3.7.

**Tabel 3.7.** Nilai Yang Digunakan Pada 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

Selanjutnya dilakukan denormalisasi data. Berikut merupakan contoh perhitungan denormalisasi data menggunakan persamaan 2.11.

$$x_{denormalisasi} = x(x_{max} - x_{min}) + x_{min}$$

$$\begin{aligned} x_{1RNN} &= 0.0026(3182.40 - 371.78) + 371.78 \\ &= 0.0026 \times 2810.62 + 371.78 \\ &= 7.3076 + 371.78 \\ &= 379.09 \end{aligned}$$

$$\begin{aligned} x_{1LSTM} &= 0.0015(3182.40 - 371.78) + 371.78 \\ &= 0.0015 \times 2810.62 + 371.78 \\ &= 4.2159 + 371.78 \\ &= 375.99 \end{aligned}$$

Sehingga didapatkan nilai baru (prediksi) yang akan diukur tingkat keakuratannya berdasarkan nilai aktual antara *Close Price* dan nilai prediksi pada model RNN dan LSTM. Berikut merupakan hasil denormalisasi data yang ditunjukkan pada Tabel 3.8.

**Tabel 3.8.** Hasil 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

Evaluasi model dilakukan untuk mengukur sejauh mana model RNN dan LSTM mampu menangkap pola dalam data dan menghasilkan prediksi yang mendekati nilai aktual. Proses evaluasi ini dilakukan dengan menggunakan tiga metrik, yaitu MSE, RMSE, dan MAE.

### 1. Mean Squared Error (MSE)

MSE dihitung dengan menggunakan rumus dari persamaan 2.12.

#### MSE untuk RNN:

$$\begin{aligned}
 MSE_{RNN} &= \\
 &= \frac{(408.44 - 379.09)^2 + (411.06 - 382.74)^2 + (405.82 - 383.58)^2}{5} \\
 &= \frac{861.12 + 801.99 + 494.62 + 318.48 + 407.24}{5} = \frac{2883.45}{5} = 576.69
 \end{aligned}$$

**MSE untuk LSTM:**

$$MSE_{LSTM} =$$

$$= \frac{(408.44 - 375.99)^2 + (411.06 - 379.65)^2 + (405.82 - 382.46)^2 + (400.59 - 384.15)^2 + (403.20 - 385.83)^2}{5}$$
$$= \frac{1053.02 + 986.46 + 545.59 + 270.29 + 301.69}{5} = \frac{3156.95}{5} = 631.69$$

**2. Root Mean Squared Error (RMSE)**

RMSE dihitung dengan menggunakan rumus dari persamaan 2.13.

**RMSE untuk RNN:**

$$RMSE_{RNN} \sqrt{576.69} = 24.02$$

**RMSE untuk LSTM:**

$$RMSE_{LSTM} \sqrt{631.69} = 25.12$$

**3. Mean Absolute Error (MAE)**

MAE dihitung dengan menggunakan rumus dari persamaan 2.14.

**MAE untuk RNN:**

$$MAE_{RNN} = \frac{(408.44 - 379.09) + (411.06 - 382.74) + (405.82 - 383.58) + (400.59 - 382.74) + (403.20 - 383.02)}{5}$$
$$= \frac{29.35 + 28.32 + 22.24 + 17.85 + 20.18}{5} = \frac{117.94}{5} = 23.59$$

**MAE untuk LSTM:**

$$MAE_{LSTM}$$

$$= \frac{(408.44 - 375.99) + (411.06 - 379.65) + (405.82 - 382.46) + (400.59 - 384.15) + (403.20 - 385.83)}{5}$$
$$= \frac{32.45 + 31.41 + 23.36 + 16.44 + 17.37}{5} = \frac{121.03}{5} = 24.21$$

Sehingga didapatkan hasil evaluasi model RNN dan LSTM berdasarkan metrik MSE, RMSE, dan MAE dirangkum dalam Tabel 3.9.

**Tabel 3.9.** Hasil Evaluasi Metrik RNN dan LSTM

Model	MSE	RMSE	MAE
RNN	<b>576.69</b>	<b>24.02</b>	<b>23.59</b>
LSTM	<b>631.69</b>	<b>25.12</b>	<b>24.21</b>

### 3.3. Tahapan Perancangan

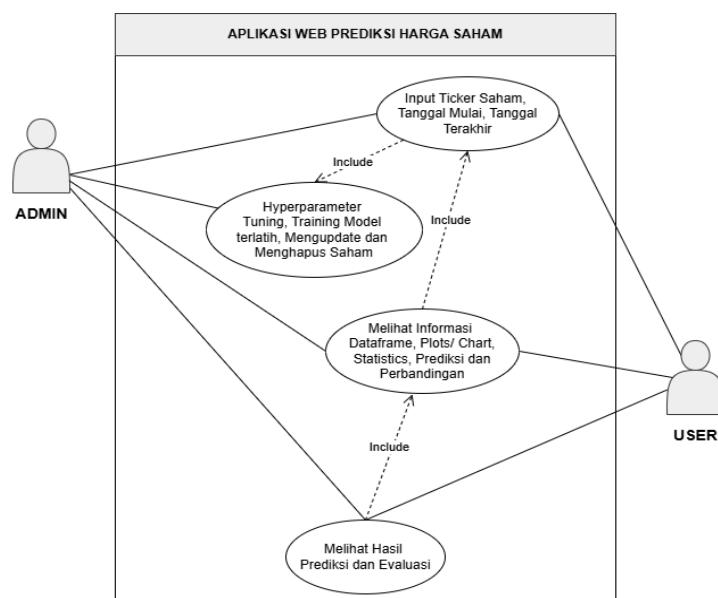
Perancangan yang akan dibangun didalamnya memuat UML dan perancangan *User Interface* (UI).

#### 3.3.1. *Unified Modeling Language* (UML)

*Unified Modeling Language* (UML) yang digunakan dalam memvisualisasikan penelitian ini meliputi:

##### 1. *Use Case Diagram*

Proses prediksi harga saham menggunakan Aplikasi Web ini menggambarkan bagaimana pengguna berinteraksi dengan sistem.

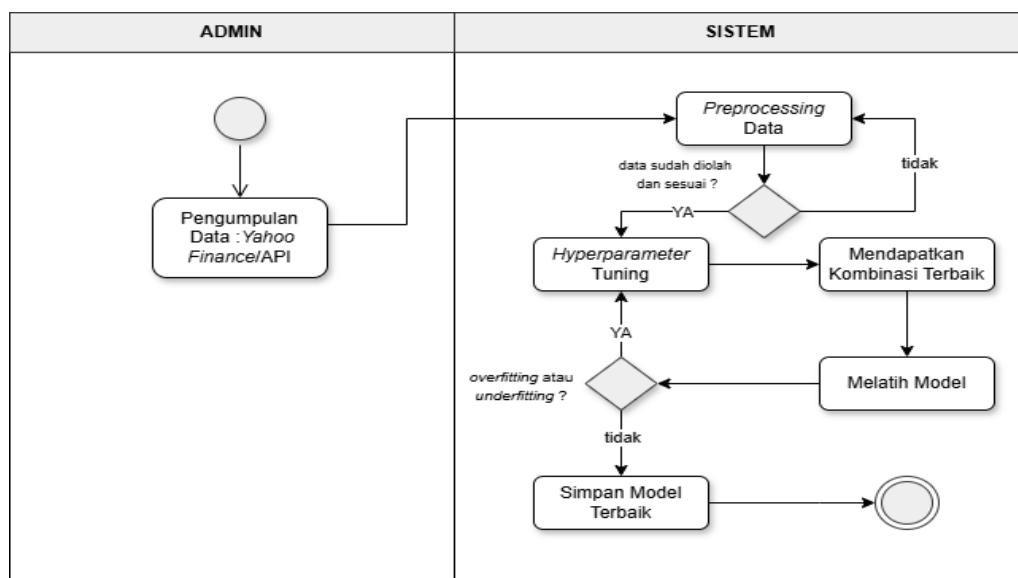


**Gambar 3.2.** *Use Case Diagram*

## 2. Activity Diagram

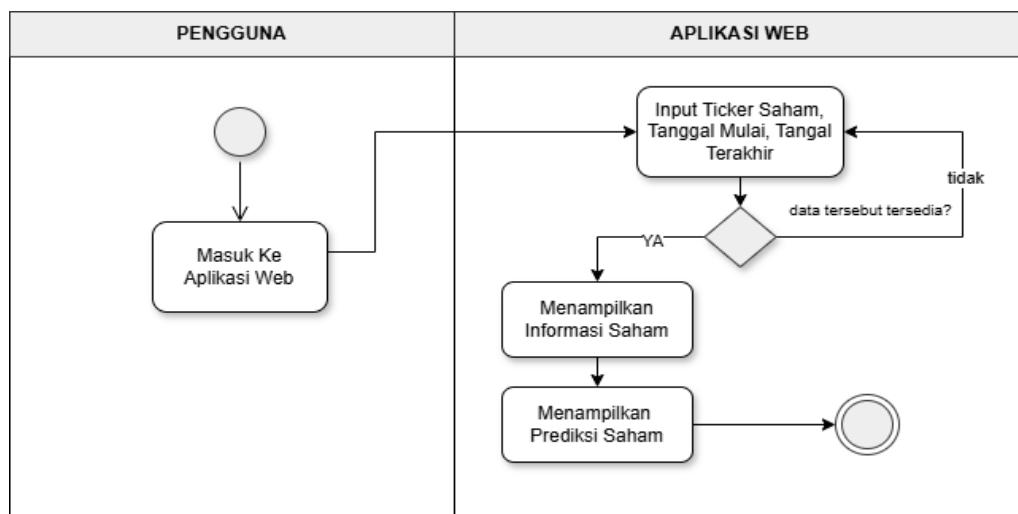
Pada bagian ini peneliti menjelaskan berbagai proses yang terjadi dalam program atau aplikasi prediksi harga saham, terdapat dua proses utama yaitu pelatihan model terbaik dan menampilkan hasil prediksi. Melatih model terbaik adalah kegiatan yang dilakukan oleh admin, sementara pengguna hanya dapat melihat informasi saham dan hasil prediksi dalam bentuk aplikasi web.

### 1) Proses Pelatihan Model Terbaik



Gambar 3.3. Activity Diagram Pelatihan Model Terbaik

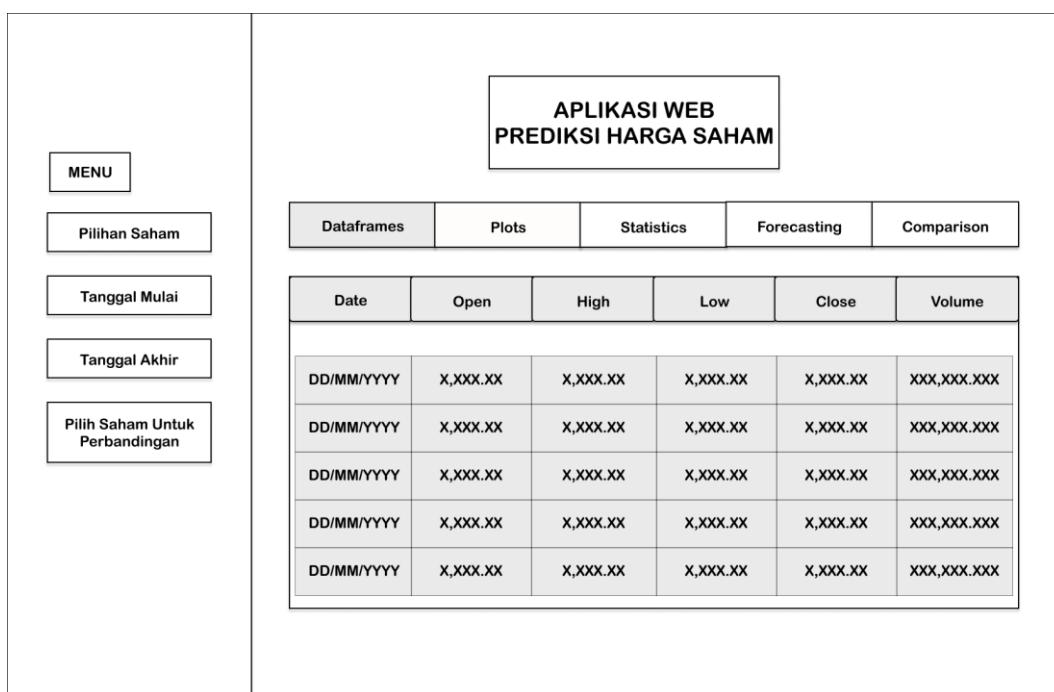
### 2) Proses Menampilkan Hasil Prediksi



Gambar 3.4. Activity Diagram Menampilkan Hasil Prediksi

### 3.3.2. *User Interface* (UI)

Perancangan UI merupakan proses mendesain tampilan antarmuka aplikasi 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 ini merupakan desain UI halaman aplikasi web ditunjukkan pada Gambar 3.5.



**Gambar 3.5.** Desain UI Halaman Aplikasi Web

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. Hasil**

##### **4.1.1. Deskripsi Dataset**

Dataset saham BBRI diperoleh melalui API Yahoo Finance yang memungkinkan pengambilan data secara otomatis, cepat dan efisien. Dengan hanya menginstal pustaka Yahoo Finance di lingkungan Python, dataset saham dapat diakses dan digunakan untuk keperluan analisis. Pada tahap awal, peneliti mengunduh dataset tersebut dalam format Excel (.xlsx) agar dataset tersusun secara rapi dan tersimpan dalam folder *datasets*, sebelum kemudian diimpor dan diproses untuk tujuan berikutnya.

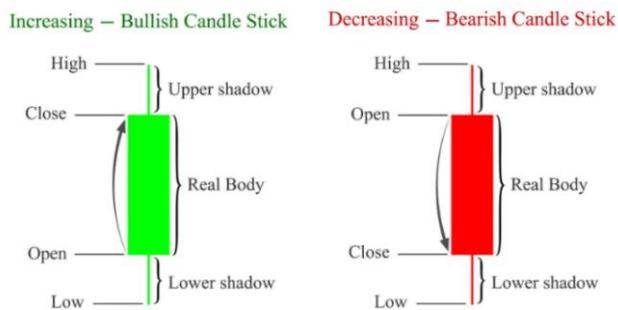
Pergerakan harga saham BBRI selama 25 tahun yang diambil pada tanggal 1 Januari 2010 – 1 Januari 2025, direpresentasikan dalam bentuk grafik *candlestick* yang dapat dilihat pada Gambar 4.1.



**Gambar 4.1.** Grafik Candlestick Saham BBRI 2010-2025

Grafik tersebut terlihat seperti grafik garis dikarenakan data yang sangat banyak. Grafik tersebut menunjukkan tren kenaikan harga secara keseluruhan meskipun beberapa periode berfluktuasi tajam terutama penurunan signifikan sekitar tahun 2020 dikarenakan terjadi pandemi COVID-19 dan awal tahun 2024.

Grafik *candlestick* tersebut terdiri dari empat komponen harga, yaitu harga pembukaan (*open*), penutupan (*close*), tertinggi (*high*), dan terendah (*low*). Struktur candlestick sendiri terbagi menjadi dua bagian utama, yaitu badan (*body*) dan ekor (*shadow*).



**Gambar 4.2.** Grafik Candlestick Saham BBRI 30 Hari Terakhir

Gambar 4.2. merupakan grafik *candlestick* yang terlihat lebih jelas mencakup data *open*, *high*, *low*, *close*, dan *volume* dibagian bawah. *Candle* diatur berwarna hijau dan merah. Apabila nilai *open* lebih tinggi daripada nilai *close* maka *candle* berwarna merah. Sebaliknya, jika nilai *open* lebih rendah daripada nilai *close* maka candle berwarna hijau. Garis dibagian atas dan bawah candle merepresentasikan nilai *high* dan *low* pada hari tersebut.

#### **4.1.2. Tahap Implementasi Model RNN & LSTM**

#### **4.1.3. Tahap Pelatihan & Pengujian Model**

#### **4.1.4. Tampilan Aplikasi Web Prediksi Harga Saham**

Aplikasi Web prediksi harga saham ini dibangun untuk membantu pengguna dalam menganalisis prediksi harga saham menggunakan *machine learning*. Aplikasi Web yang dibangun memberikan prediksi berbasis model seperti RNN dan LSTM.

#### **4.1.5. Menampilkan Proses Prediksi Pada Aplikasi Web**

Setelah proses membangun model RNN & LSTM untuk melakukan *training* dan *testing* pada prediksi harga saham, kemudian menampilkannya dalam bentuk yang lebih informatif melalui aplikasi web menggunakan *streamlit*. Streamlit merupakan framework open-source berbasis Python yang digunakan untuk membuat aplikasi web interaktif, khususnya untuk bidang *data science* dan *machine learning*. Dalam penelitian ini akan menampilkan informasi terkait saham dan hasil prediksinya. Sebelum menjalankan aplikasi ini, diperlukan pemasangan modul *streamlit* pada Python Environment. Proses instalasinya cukup mudah, yaitu dengan menjalankan perintah **pip install streamlit** pada command line. Setelah instalasi selesai, aplikasi dapat dijalankan menggunakan perintah **streamlit run nama\_file.py**. Dengan demikian, informasi mengenai saham dan prediksi dapat ditampilkan dengan lebih mudah dipahami dan diakses oleh pengguna.

## 4.2. Pembahasan

### 4.2.1.

### 4.2.2.

### 4.2.3. Hyperparameter Tuning RandomizedSearchCV

Hyperparameter	Nilai
Model	<b>RNN</b>
Unit/Neuron	128
Layer	2
Dropout	0.2
Learning Rate	0.001
Epoch	100
Batch Size	64
Loss	MSE
Optimizer	Adam

Hyperparameter	Nilai
Model	<b>LSTM</b>
Unit/Neuron	256
Layer	1
Dropout	0.2
Learning Rate	0.001
Epoch	100
Batch Size	64
Loss	MSE
Optimizer	Adam

### 4.2.4.

Pembacaan dataset dengan menggunakan library **pandas** yang diinisialisasikan menjadi variabel **pd**. Pandas (*Python For Data Analysis*) merupakan sebuah *library* python yang berfokus pada proses analisis data seperti

manipulasi, persiapan, dan pembersihan data. Kemudian mengubah dataset menjadi sebuah *dataframe* agar dapat dibaca dan diolah yang ditampung pada variabel `df`. Berikut tampilan 5 baris data pertama dan 5 baris data terakhir.

	Date	Open	Low	High	Close	Adj Close	Volume
0	2010-01-04	709.08	695.44	713.62	709.08	432.14	125094160
1	2010-01-05	713.62	709.08	718.17	713.62	434.91	163644326
2	2010-01-06	704.53	699.99	722.71	704.53	429.37	104842810
3	2010-01-07	695.44	686.35	704.53	695.44	423.83	101933260
4	2010-01-08	699.99	686.35	699.99	699.99	426.60	87237006
...	...	...	...	...	...	...	...
3696	2024-12-20	4070.00	4050.00	4120.00	4060.00	3929.50	252689600
3697	2024-12-23	4130.00	4110.00	4210.00	4210.00	4074.68	167689800
3698	2024-12-24	4220.00	4170.00	4250.00	4200.00	4065.00	199536100
3699	2024-12-27	4100.00	4080.00	4120.00	4100.00	4100.00	143104400
3700	2024-12-30	4080.00	4070.00	4120.00	4080.00	4080.00	153934700

3701 rows × 7 columns

Berdasarkan output, harga saham harian PT. Bank Rakyat Indonesia berisi 3701 baris data dengan 7 kolom.

### ***Pre-Processing Data***

Data mentah akan diolah agar dapat dipelajari oleh model dengan tahapan seperti berikut ini.

### **Pembersihan Data**

Memvisualisasikan data dalam bentuk grafik agar dapat dibandingkan dengan grafik setelah prediksi. Sumbu X diberi label "Date", sementara sumbu Y diberi label "Price". Grafik berwarna biru menunjukkan "Close Price" atau harga penutupan, yang merepresentasikan nilai pada sumbu Y, yaitu Price yang ditunjukkan pada gambar berikut.



Gambar tersebut menampilkan grafik riwayat harga penutupan atau *Close Price* saham BBRI dari tahun 2010 hingga akhir tahun 2024.

## 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/Rjcse.74>
- Faridah, N., & Sugiantoro, B. (2023). Analisis Optimasi Pada Algoritma Long Shortterm 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](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 Long Short-Term Memory. *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*. Sekolah Tinggi Manajemen Informatika Dan Komputer Bani Saleh.
- 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>
- Kusuma, N. P. N. (2024). Prediksi Harga Saham Blue Chip Pada Indeks IDX 30 Menggunakan Algoritma Recurrent Neural Network (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.). <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 Long Short-Term Memory*. <Https://Repository.Unja.Ac.Id/62231/>

- Riyantoko, P. A., Fahruddin, 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>
- Sumirat, L. P., dkk. (2023). Dasar-Dasar Rekayasa Perangkat Lunak. <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. Dalam *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. <https://journal.uii.ac.id/Snati/article/view/13398>
- Syahrul. (2023). Penggunaan Metode Long Short Term Memory Pada Peramalan Harga Saham Pt Unilever Indonesia Tbk. *SKRIPSI*. <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*. <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>