

**ANALISIS CURAH HUJAN DI ACEH BESAR  
MENGUNAKAN METODE *LONG SHORT TERM  
MEMORY***

**TUGAS AKHIR**

**Diajukan Oleh :**

**ZEBI AMELIA PUTRI**

**NIM. 190705006**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**PRODI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2024 M/1445 H**

## LEMBAR PERSETUJUAN

### ANALISIS CURAH HUJAN DI ACEH BESAR MENGUNAKAN METODE *LONG SHORT TERM MEMORY*

#### TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana  
pada Prodi Teknologi Informasi

Oleh:

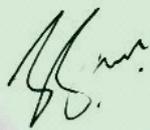
**ZEBI AMELIA PUTRI**

**NIM. 190705006**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing



**Khairan AR, M.Kom**  
**NIP. 198607042014031001**

Mengetahui,  
Ketua Program Studi Teknologi  
Informasi



**Malahayati, M.T**  
**NIP. 198301272015032003**

## LEMBAR PENGESAHAN

### ANALISIS CURAH HUJAN DI ACEH BESAR MENGUNAKAN METODE *LONG SHORT TERM MEMORY*

#### TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)  
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal: Rabu / 12 Juni 2024  
05 Dzulhijjah 1445 H  
di Darussalam, Banda Aceh  
Panitia Ujian Munaqasah Tugas Akhir

Ketua/Sekretaris



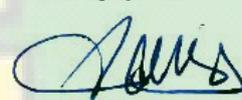
**Khairan AR, M.Kom**  
**NIP.198607042014031001**

Penguji I,



**Mulyan Fadhli, M.T**  
**NIP.198811282020121006**

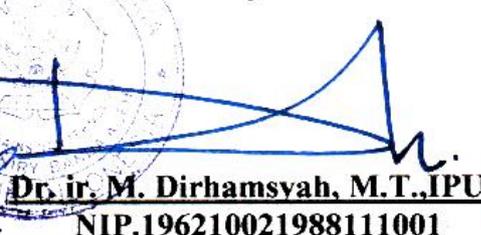
Penguji II,



**Aulia Syarif Aziz, S.kom., MSc**  
**NIP.199305212022031001**

Mengetahui,

Dekan Fakultas Sains dan Teknologi  
UIN Ar-Raniry Banda Aceh



**Dr. ir. M. Dirhamsyah, M.T., IPU**  
**NIP.196210021988111001**

## LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Zebi Amelia Putri  
NIM : 190705006  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul Tugas Akhir : Analisis Curah Hujan di Aceh Besar  
Menggunakan Metode *Long Short Term Memory*

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 12 Juni 2024

Yang Menyatakan  
  
129A0ALX235575522

(Zebi Amelia Putri)

## ABSTRAK

Nama : Zebi Amelia Putri  
NIM : 190705006  
Program Studi : Teknologi Informasi  
Judul : Analysis of Rainfall in Aceh Using the Long Short Term Memory Method  
Tanggal Sidang : 12 Juni 2024  
Jumlah Halaman : 63 Halaman  
Pembimbing : Khairan AR, M.Kom

Curah hujan adalah jumlah air yang turun ke permukaan tanah selama periode tertentu. Data curah hujan ini dicatat oleh BMKG (Badan Meteorologi Klimatologi dan Geofisika) setiap harinya. Faktor-faktor seperti suhu, kelembaban udara, kecepatan angin, dan aktivitas matahari mempengaruhi curah hujan. Ketepatan dalam memprediksi cuaca sangat penting karena cuaca memiliki dampak besar pada aktivitas masyarakat.. Penelitian ini memfokuskan untuk meningkatkan pemahaman kita tentang Algoritma Long short Term Memory dan menjadikan RMSE sebagai evaluasi dari prediksi Curah hujan di Aceh.

Dalam penelitian ini mengumpulkan dataset dari dari situs web resmi BMKG <https://dataonline.bmkg.go.id>. Implementasi metode *Long Short Term Memory* menunjukkan hasil uji performa yang bagus, Nilai epoch 75 memiliki performa tertinggi, dengan nilai RMSE Data Train 14.88, RMSE Data Test 20.18, MAE Data Test 5.32, dan MAE Data Test 7.78. nilai epoch 150 memiliki performa terendah, dengan nilai RMSE Data Train 13.33, RMSE Data Test 22.01, MAE Data Train 6.65, dan MAE Data Test 9.46. Hasil penelitian ini menunjukkan bahwa bahwa pemilihan jumlah Epoch yang tepat sangat penting untuk mencapai keseimbangan antara performa yang baik dan menghindari overfitting pada model.

**Kata Kunci:** Curah Hujan, *Long Short Term Memory*, Analisis Cuaca, RMSE, BMKG

## ABSTRACT

Name : Zebi Amelia Putri  
NIM : 190705006  
Department : Information Technology  
Judul : Rainfall Prediction in Aceh Using the Long Short Term  
Memory Method  
Date : 12 Juni 2024  
Number of Pages : 63 Pages  
Supervisor : Khairan AR, M.Kom

Rainfall is the amount of water that falls to the ground surface during a certain period. This rainfall data is recorded by the BMKG (Meteorology, Climatology and Geophysics Agency) every day. Factors such as temperature, air humidity, wind speed, and solar activity influence rainfall. Accuracy in predicting weather is very important because weather has a big impact on people's activities. This research focuses on increasing our understanding of the Long Short Term Memory Algorithm and making RMSE an evaluation of rainfall predictions in Aceh.

In this research, Collected Datasets from the official BMKG website <https://dataonline.bmkg.go.id>. The implementation of the Long Short Term Memory method shows good performance test results. The epoch value 75 has the highest performance, with an RMSE Data Train value of 14.88, RMSE Data Test 20.18, MAE Data Test 5.32, and MAE Data Test 7.78. The epoch value of 150 has the lowest performance, with an RMSE Data Train value of 13.33, RMSE Data Test 22.01, MAE Data Train 6.65, and MAE Data Test 9.46. The results of this research show that choosing the right number of Epochs is very important to achieve a balance between good performance and avoiding overfitting of the model.

**Keywords:** *Rainfall, Long Short Term Memory, Weather Analys, RMSE, BMKG*

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penayang. Puji syukur penulis panjatkan panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “**Analisis Curah Hujan di Aceh Besar Menggunakan Metode *Long Short Term Memory***”. Shalawat serta salam kepada Rasulullah SAW yang mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang.

Dalam proses penyusunan tugas akhir ini penulis menyadari bahwa tidak dapat menyelesaikan tugas akhir jika tidak adanya dukungan, bimbingan, motivasi dan bantuan dari berbagai pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini. Dengan kerendahan hati penulis ingin berterimakasih kepada :

1. Kedua orangtua yang senantiasa memberikan dukungan dan doa kepada saya dalam menyusun tugas akhir ini.
2. Ketua dan Sekretaris Prodi Teknologi Informasi Ibu Malahayati, MT dan Bapak Khairan AR, M.Kom yang telah memberikan bimbingan dan arahan dalam penyusunan tugas akhir ini.
3. Bapak Khairan AR, M.Kom selaku Pembimbing yang senantiasa memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
4. Ibu Cut Ida Rahmadiana, S.Si selaku *Staff* Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
5. Bapak Dr. Ir. M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry yang telah mendukung dan memberi motivasi untuk kami.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu kepada penulis dalam bidang Teknologi Informasi.

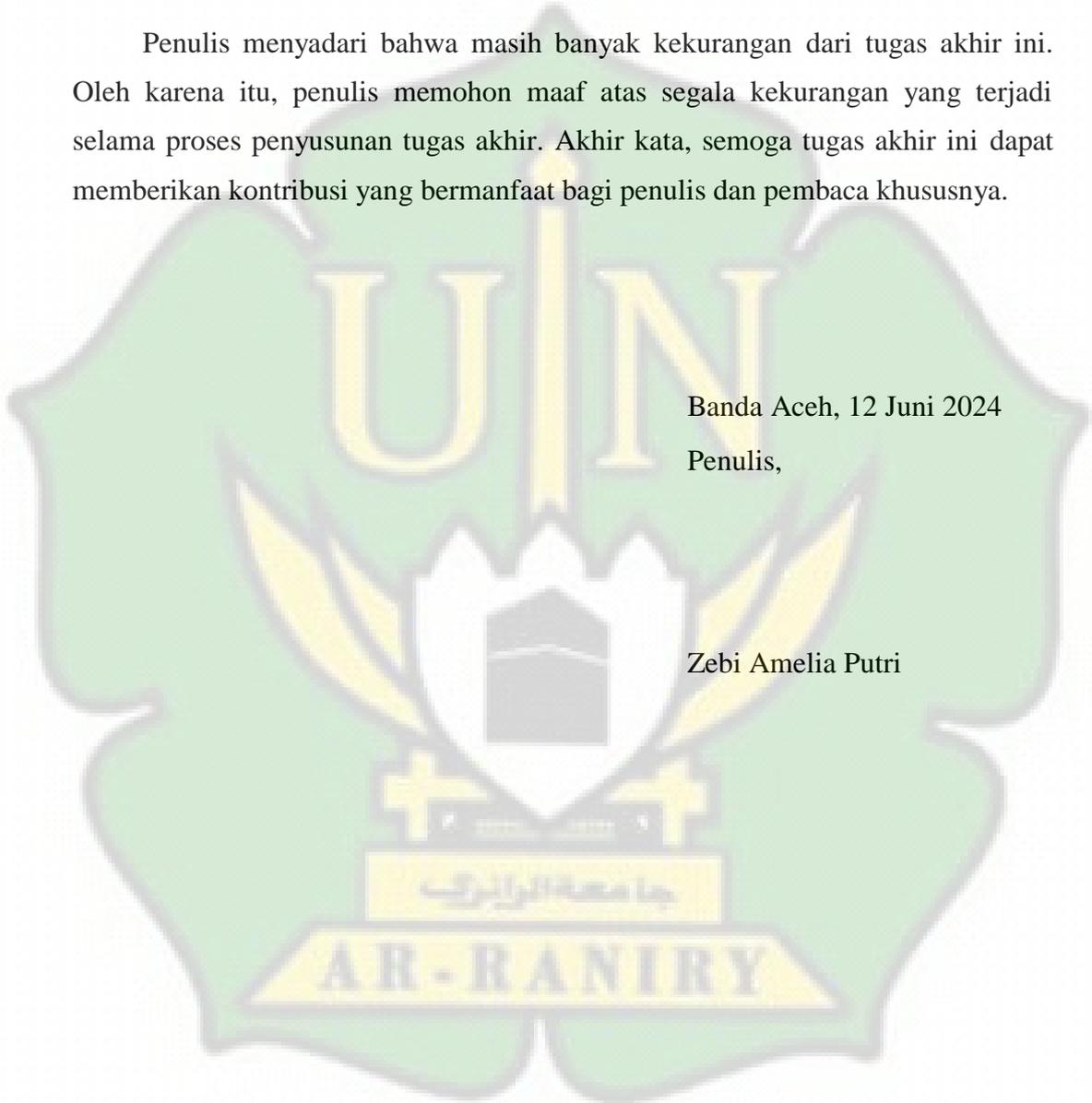
7. Terima kasih untuk pemilik nama Muhammad Alfarizi Yanda, telah menjadi sosok rumah yang selalu ada buat saya, dan memberikan dukungan moral dan mental dalam menyelesaikan penulisan skripsi ini
8. Semua pihak yang telah membantu dalam menyelesaikan proposal tugas akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dari tugas akhir ini. Oleh karena itu, penulis memohon maaf atas segala kekurangan yang terjadi selama proses penyusunan tugas akhir. Akhir kata, semoga tugas akhir ini dapat memberikan kontribusi yang bermanfaat bagi penulis dan pembaca khususnya.

Banda Aceh, 12 Juni 2024

Penulis,

Zebi Amelia Putri



# DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>LEMBAR PERNYATAAN KEASLIAN .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian .....	4
<b>BAB II KAJIAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Penelitian Terdahulu.....	5
2.2 Analisis .....	10
2.3 Curah Hujan.....	10
2.4 <i>Long short Term Memory</i> .....	11
2.4.1. Proses <i>Long Short Term Memory</i> (LSTM) .....	22
2.5 Deret Waktu ( <i>Time Series</i> ) .....	26
2.6 <i>Deep Learning</i> .....	27
2.7 Jaringan Saraf Tiruan (JST).....	27
2.8 <i>Reccurent Neural Network</i> (RNN) .....	28
2.9 Normalisasi.....	28
2.10 RMSE .....	29
2.11 MAE .....	29
2.12 <i>Tools</i> .....	30
2.12.1 <i>Python</i> .....	30
2.12.2 <i>Jupyter Notebook</i> .....	30
<b>BAB III METODE PENELITIAN .....</b>	<b>31</b>
3.1 Tahapan Penelitian .....	31
3.2 Pengumpulan Data.....	31
3.2.1 Dataset.....	32

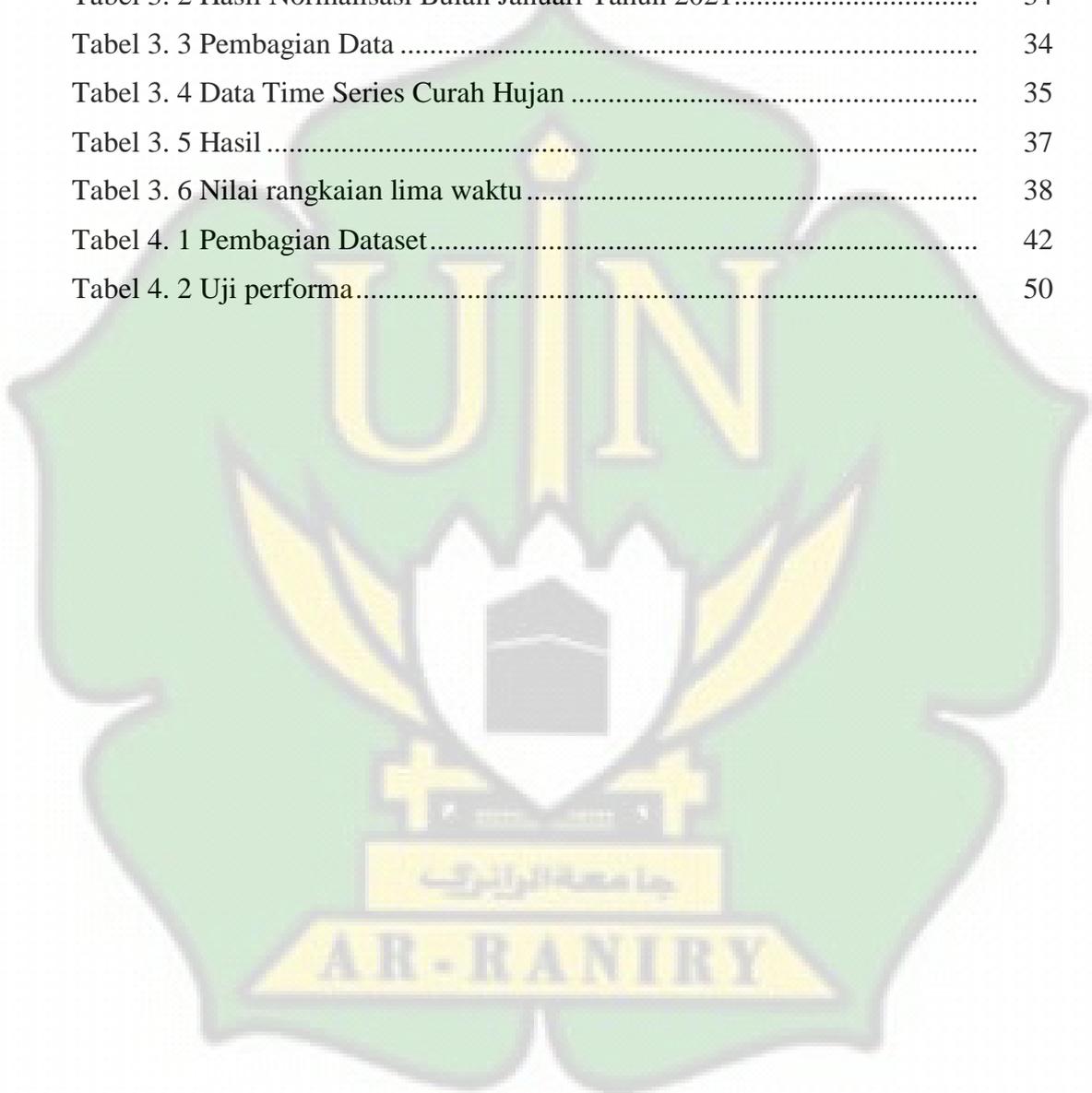
3.3	<i>Preprocessing</i> .....	33
3.3.1	Normalisasi .....	33
3.3.2	Pembagian Data .....	34
3.3.3	Pembentukan Data <i>Time Series</i> .....	35
3.3.4	Inisialisasi parameter.....	36
3.3.5	Pengujian Model .....	36
3.3.6	Denormalisasi.....	37
3.3.7	Evaluasi Hasil.....	38
3.4	Alat Bantu Penelitian.....	38
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>40</b>
4.1	Pengambilan Dataset .....	40
4.2	Membaca Dataset Curah Hujan.....	40
4.3	<i>Preprocessing</i> Data .....	41
4.3.1	Menghilangkan Data Yang Tidak Bernilai (Nan).....	41
4.3.2	Normalisasi Data.....	42
4.3.3	Pembagian Data .....	42
4.3.4	Pembentukan Timesteps.....	43
4.4	Implementasi Model.....	43
4.5	Membuat Prediksi dan Pembalikan Prediksi .....	47
4.6	Evaluasi Hasil .....	47
4.7	Visualisasi Hasil .....	48
4.8	Hasil Uji Performa.....	49
<b>BAB V PENUTUP .....</b>		<b>54</b>
5.1	Kesimpulan.....	54
5.2	Saran .....	55
<b>DAFTAR PUSTAKA .....</b>		<b>56</b>
<b>LAMPIRAN.....</b>		<b>59</b>
<b>RIWAYAT HIDUP .....</b>		<b>65</b>

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Layer LSTM.....	12
Gambar 2. 2 Arsitektur LSTM.....	15
Gambar 2. 3 Memory cell.....	15
Gambar 2. 4 Struktur Sel LSTM.....	17
Gambar 2. 5 Lapisan dari Sigmoid.....	18
Gambar 2. 6 Alur informasi forget Gate.....	18
Gambar 2. 7 Alur Informasi Input Gate.....	19
Gambar 2. 8 Alur cell state.....	20
Gambar 2. 9 Alur Informasi Output gate.....	21
Gambar 3. 1 Diagram Metode Penelitian Long Short Term Memory.....	31
Gambar 3. 2 Tampilan Dashboard Pusat Database BMKG.....	32
Gambar 3. 3 Alur Preprocessing.....	33
Gambar 3. 4 Ilustrasi pola time series.....	35
Gambar 4. 1 Sampel Dataset Curah Hujan.....	40
Gambar 4. 2 Raw Dataset Curah Hujan.....	41
Gambar 4. 3 Hasil Nan.....	41
Gambar 4. 4 Hasil Normalisasi Dataset.....	42
Gambar 4. 5 Hasil Pembentukan timesteps.....	43
Gambar 4. 6 Hasil Uji Epoch.....	44
Gambar 4. 7 Loss Epoch 25.....	44
Gambar 4. 8 Loss Epoch 50.....	45
Gambar 4. 9 Loss Epoch 75.....	45
Gambar 4. 10 Loss Epoch 100.....	46
Gambar 4. 11 Loss Epoch 150.....	46
Gambar 4. 12 Loss Epoch 200.....	47
Gambar 4. 13 Hasil Prediksi Data latih dan Data Uji.....	47
Gambar 4. 14 Hasil perhitungan RMSE.....	47
Gambar 4. 15 Hasil Perhitungan MAE.....	48
Gambar 4. 16 Bloxplot RR.....	48
Gambar 4. 17 Korelasi Matrix.....	49
Gambar 4. 18 Perbandingan RMSE Data Train dan Test.....	51
Gambar 4. 19 Perbandingan MAE Data Train dan Test.....	52

## DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Terdahulu .....	6
Tabel 2. 2 Hasil perhitungan manual .....	26
Tabel 3. 1 Dataset curah hujan bulan januari tahun 2021 .....	32
Tabel 3. 2 Hasil Normalisasi Bulan Januari Tahun 2021.....	34
Tabel 3. 3 Pembagian Data .....	34
Tabel 3. 4 Data Time Series Curah Hujan .....	35
Tabel 3. 5 Hasil .....	37
Tabel 3. 6 Nilai rangkaian lima waktu .....	38
Tabel 4. 1 Pembagian Dataset.....	42
Tabel 4. 2 Uji performa.....	50



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan zaman dan kemajuan teknologi, pemahaman tentang cuaca dan iklim menjadi semakin penting, terutama dalam konteks perubahan iklim global dan dampaknya terhadap lingkungan. Cuaca mengacu pada kondisi atmosfer di suatu lokasi dalam periode waktu tertentu. Hal ini melibatkan faktor-faktor seperti suhu udara, tingkat kelembaban, tekanan udara, arah dan kecepatan angin, serta kondisi awan dan curah hujan. Cuaca adalah kondisi atmosfer yang diukur secara menyeluruh dengan mempertimbangkan perubahan, perkembangan, dan muncul atau menghilangnya fenomena udara (Ardytha Luthfiarta dkk., 2020). Salah satu komponen utama iklim alam adalah curah hujan. Beberapa faktor lain yang memengaruhi cuaca termasuk suhu udara, tekanan udara, kelembaban udara, laju uap air, awan, curah hujan, dan angin. Karena terletak di sepanjang garis khatulistiwa, Indonesia dikenal sebagai negara dengan iklim tropis (Nola Ritha dkk., 2016).

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) memiliki informasi cuaca yang dapat diakses oleh setiap provinsi di Indonesia. BMKG Aceh telah membuat prakiraan cuaca untuk digunakan sendiri, melalui Banyak stasiun salah satunya Stasiun Sultan Iskandar Muda yang terletak di Lintang 5.52244, Bujur 95.41700, dan Elevasi 20 menggunakan model analisis cuaca yang dibuat oleh BMKG Pusat sendiri dan berbagai sumber data, tetapi para ahli BMKG sendiri masih menganggap prediksi tersebut masih kurang akurat. (BMKG., 2008)

Teknologi seperti big data, machine learning, dan Internet of Things (IoT) memungkinkan pengumpulan data cuaca secara *real-time* dan analisis yang lebih mendalam. Dengan pemahaman yang lebih mendalam tentang cuaca dan iklim, diharapkan dapat dikembangkan solusi teknologi yang lebih efisien dan berkelanjutan untuk menghadapi tantangan yang ditimbulkan oleh perubahan iklim global.

Data curah hujan dapat digunakan untuk mengklasifikasikan iklim berdasarkan perbandingan antara jumlah rata-rata bulan kering dengan jumlah rata-rata bulan basah. Bulan dikategorikan sebagai kering jika curah hujannya kurang dari 60 mm/bulan, sementara bulan dikategorikan sebagai basah jika curah hujannya di atas 100 mm/bulan. Di antara bulan-bulan tersebut terdapat bulan yang lembab, yang terjadi ketika curah hujannya berkisar antara 60-100 mm/bulan (Nola Ritha dkk., 2016).

Analisis curah hujan melibatkan proses mengevaluasi data historis tentang curah hujan untuk memahami pola, tren, dan faktor yang mempengaruhi distribusi dan intensitas curah hujan. Curah hujan terjadi ketika air menguap dari laut, tanah, tumbuhan, dan sebagainya, lalu mengalami pendinginan atau kondensasi pada suhu tertentu. Uap air ini kemudian berkumpul dan membentuk awan. Semakin banyak uap air yang bergabung dan membentuk awan, semakin banyak juga titik-titik air yang jatuh ke bumi dalam bentuk hujan. Analisis curah hujan berusaha untuk mengidentifikasi dan memahami faktor-faktor yang mempengaruhi jumlah hujan yang jatuh di suatu lokasi dan waktu tertentu. Tujuannya adalah untuk memberikan informasi yang berguna dalam perencanaan dan pengelolaan risiko, seperti potensi banjir, kekeringan, dan dampak lainnya yang mungkin terjadi akibat perubahan cuaca. Proses analisis curah hujan melibatkan evaluasi data cuaca seperti suhu, kelembaban, tekanan udara, dan pola angin. Dengan menganalisis tren dan pola tersebut, kita dapat memahami lebih baik tentang distribusi curah hujan dan faktor-faktor yang mempengaruhinya (Diera Desmonda dkk., 2018).

Penelitian mengenai Prediksi curah hujan sebelumnya pernah diteliti. Salah satunya adalah penelitian (Muhammad Rizki dkk, 2020) yang berjudul “Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory Untuk Prediksi Curah Hujan Kota Bandung”. Penelitian tersebut mengidentifikasi isu utama dalam prediksi curah hujan di Kota Bandung, yaitu ketidakpastian cuaca dan keterbatasan metode konvensional dalam menangani data yang kompleks.

Analisis mengungkap pola musiman yang jelas, fluktuasi tak beraturan, dan tren jangka panjang yang dipengaruhi oleh perubahan iklim. Metode Long Short Term Memory (LSTM) menunjukkan peningkatan akurasi prediksi dengan RMSE yang lebih rendah, terutama pada parameter optimal epoch 75 dan batch size 1. Selain itu, penggunaan teknologi IoT memungkinkan pengumpulan data curah hujan secara *real-time*, meningkatkan responsivitas terhadap perubahan cuaca. Dengan demikian, penelitian ini menawarkan pendekatan yang lebih canggih dan akurat dalam memantau dan memprediksi curah hujan, mendukung berbagai sektor untuk merencanakan aktivitas mereka lebih baik sesuai kondisi cuaca.

Pada penelitian ini, Penulis memutuskan untuk melakukan penelitian berjudul "Analisis Curah Hujan Menggunakan *Long Short Term Memory* di Aceh Besar", tujuan utama dari penelitian ini adalah untuk meningkatkan pemahaman tentang Algoritma *Long short Term Memory* dan menjadikan RMSE sebagai evaluasi dari prediksi Curah hujan di Aceh Besar.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, rumusan masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana proses Analisis Curah Hujan menggunakan Metode *Long Short Term Memory* (LSTM).
2. Mengukur performa *Long Short Term Memory* (LSTM) dalam menganalisis curah hujan di Aceh.

## 1.3 Tujuan Penelitian

Tujuan penelitian ini, berdasarkan rumusan masalah di atas, adalah :

1. Mengimplementasikan Metode *Long Short Term Memory* (LSTM) untuk Menganalisis Curah Hujan.
2. Mengevaluasi akurasi analisis curah hujan yang dihasilkan oleh *Long Short Term Memory* dengan RMSE.

#### 1.4 Batasan Penelitian

Adapun batasan masalah penelitian ini yaitu :

1. Penelitian ini hanya menggunakan data curah hujan dari data online BMKG (Badan Meteorologi, Klimatologi dan Geofisika).
2. Menggunakan Metode *Long Short Term Memory* dan RMSE untuk mengevaluasi hasil.



## **BAB II**

### **KAJIAN KEPUSTAKAAN**

#### **2.1 Penelitian Terdahulu**

Untuk mengembangkan penelitian yang akan penulis lakukan, maka sangat penting untuk memiliki penelitian-penelitian terdahulu sebagai acuan, khususnya yang berkaitan dengan penelitian yang dilakukan penulis dengan menggunakan algoritma Long Short Term Memory.

Dalam penelitian mengenai prediksi curah hujan, banyak peneliti menggunakan beberapa metode, seperti RNN, Fuzzy Time Series, Levenberg-Marquardt, Backpropagation, dan LSTM. Penelitian tentang prediksi curah hujan telah banyak dilakukan. Salah satunya, "Prediksi Curah Hujan Di Kota Pekanbaru Menggunakan Lstm (Long Short Term Memory)" (Yos Hendra dkk., 2023), menekankan bahwa hasil prediksi yang diperoleh tergolong cukup baik, dengan tingkat error yang paling rendah ditemukan pada komposisi 50% data pelatIH dan 50% data test, masing-masing 21.328 dan 454.901.

Penelitian yang diteliti (Ardytha Luthfiarta dkk., 2020) dengan judul "Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda" menghasilkan analisis regresi linier berganda dengan persamaan  $Y = 0,147X_1 + 0,078X_2 - 0,356X_3 + 0,019X_4$ . Hasilnya menunjukkan bahwa variabel suhu udara memiliki pengaruh yang positif dan signifikan terhadap koefisien regresi sebesar 0,147 (14,7%). Pada penelitian yang berjudul "Prediksi Curah Hujan Menggunakan *Long Short Term Memory*" yang diteliti oleh (Jamilatul Badriyah dkk., 2022) penelitian ini memperoleh hasil nilai MSE 0.489, MAE 0.537 dan R<sup>2</sup> 0.497.

Penelitian yang diteliti (Aulia Herdhyanti dkk., 2022) yang berjudul "Prediksi Curah Hujan dengan Empat Parameter menggunakan Backpropagation (Studi Kasus: Stasiun Meteorologi Ahmad Yani)" .

Menghasilkan nilai MSE sebesar 0,006952 dengan dua neuron tersembunyi, iterasi maksimum sebanyak 1000 iterasi, jumlah data latih sebanyak 70% dari jumlah dataset, dan nilai learning rate sebesar 0,05. Sedangkan akurasi terburuk didapatkan dengan nilai MSE sebesar 0,023566 dengan 2 hidden neuron, jumlah data latih sebanyak 10% dari seluruh dataset, dan iterasi maksimum 1000 dan nilai learning rate 0,05. Penelitian lainnya yang di teliti oleh (Diera Desmonda dkk., 2018) yang berjudul “Prediksi Besaran Curah Hujan Menggunakan Metode Fuzzy Time Series” penelitian ini memperoleh hasil nilai MAPE yang paling baik dihasilkan adalah 0,151% pada periode 2015-2017 dengan menggunakan jumlah interval 401. memiliki nilai akurasi sejumlah 99,849%.

Tabel 2. 1 Perbandingan Penelitian Terdahulu

PENELITI	METODE	JUDUL	DATASET	HASIL
(Yos Hendra dkk., 2023)	<i>Long Short Term Memrory</i>	Prediksi Curah Hujan Di Kota Pekanbaru Menggunakan Lstm ( <i>Long Short Term Memory</i> )	Dataset diambil dari data online BMKG Kota Pekanbaru dari tahun 2010 – 2020	Hasil data Training dan data Test menunjukkan angka 50% dan RMSE 21.328 sedangkan MSE 454.901 dari Percobaan ke 5

(Ardytha Luthfiarta dkk., 2020)	Regresi linear Berganda dengan 4 Parameter	Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda	Dataset cuaca yang digunakan bertipe data kuantitatif dari BMKG sebanyak 1.096 data yang dimulai dari awal Januari 2015 hingga akhir Desember 2017	Dari Persamaan regresi linier berganda, didapatkan hasil bahwa Variabel suhu udara sangat berpengaruh positif dan signifikan terhadap Koefisien regresi Sebesar 0,147 (14,7%)
(Jamilatul Badriyah dkk., 2022)	<i>Long Short Term Memrory</i>	Prediksi Curah Hujan Menggunakan <i>Long Short Term Memory</i>	Data curah Hujan di peroleh dari BMKG Kota Surabaya dari tahun 2015 - 2022	hasil algoritma <i>Long Short Term Memory (LSTM)</i> menunjukkan lebih baik dibandingkan dengan algoritma <i>RNN</i> dan <i>GRU</i> dengan nilai <i>MSE</i> 0.489, <i>MAE</i> 0.537 dan <i>R2</i>

(Jamilatul Badriyah dkk., 2022)	<i>Long Short Term Memrory</i>	Prediksi Curah Hujan Menggunakan <i>Long Short Term Memory</i>	Data curah Hujan di peroleh dari BMKG Kota Surabaya dari tahun 2015 - 2022	Hasil algoritma <i>Long Short Term Memory</i> menunjukkan lebih baik dibandingkan dengan algoritma <i>RNN</i> dan <i>GRU</i> dengan nilai <i>MSE</i> 0.489, <i>MAE</i> 0.537 dan <i>R2</i> 0.497
(Aulia Herdhyanti dkk., 2022)	Backpropagation	Prediksi Curah Hujan dengan Empat Parameter menggunakan Backpropagation (Studi Kasus: Stasiun Meteorologi Ahmad Yani)	Data curah Hujan di ambil dari dari stasiun BMKG Ahmad Yani di Semarang	Dari 4 Pengujian yang di lakukan, Akurasi terbaik untuk memprediksi Curah hujan adalah dengan menggunakan <i>2 Hidden neuron</i> dengan Nilai <i>MSE</i> Sebesar 0,006952

(Diera Desmonda dkk., 2018)	<i>Fuzzy Time Series</i>	Prediksi Besaran Curah Hujan Menggunakan Metode Fuzzy Time Series	Data Curah Hujan Periode 2015 – 2017 Dengan Jumlah Interval 401	Nilai <i>MAPE</i> yang paling baik dihasilkan adalah 0,151% Dari data periode 2015 – 2017 dengan menggunakan jumlah interval 401. Dengan Akurasi sebesar 99,849%.
-----------------------------	--------------------------	-------------------------------------------------------------------	-----------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

Penulis dapat mencapai beberapa kesimpulan dari perbandingan penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan oleh penulis mengenai prediksi curah hujan menggunakan Metode *Long Short Term Memory*. Kesimpulan tersebut adalah sebagai berikut :

- Tinggi rendahnya akurasi dalam memprediksi Curah Hujan sangat dipengaruhi oleh Jumlah dataset yang digunakan untuk *training*. Semakin banyak dataset yang digunakan, semakin akurat prediksi curah hujan.
- Untuk menjaga keseimbangan data pada saat pengujian, data yang digunakan pada proses Prediksi Curah Hujan harus melewati tahapan *pre-processing* dengan baik. Data yang tidak melewati tahapan *pre-processing* dengan baik, akan mengurangi hasil akurasi.

## 2.2 Analisis

analisis adalah aktivitas yang terdiri dari serangkaian kegiatan, seperti mengurai, membedakan, dan memilah sesuatu untuk dikelompokkan kembali menurut kriteria tertentu dan kemudian dicari kaitannya lalu ditafsirkan maknanya. Analisis sangat dibutuhkan untuk mengamati sesuatu yang tentunya bertujuan untuk mendapatkan hasil akhir dari pengamatan yang sudah dilakukan. Kata analisis diadaptasi dari bahasa Inggris, "analysis", yang secara etimologis berasal dari bahasa Yunani kuno yang dibaca "Analisis". Kata Analisis terdiri dari dua suku kata, yaitu "ana", yang artinya kembali, dan "luein", yang artinya melepas atau mengurai. Bila digabungkan maka kata tersebut memiliki arti menguraikan kembali (Irfan dkk., 2018).

Analisis adalah proses pemecahan masalah yang dimulai dengan hipotesis (dugaan, dan sebagainya) sampai terbukti kebenarannya melalui beberapa kepastian (pengamatan dan percobaan). Analisis melibatkan proses sistematis dalam mengevaluasi informasi yang ada tentang peristiwa masa lalu dan saat ini untuk memahami situasi atau masalah yang sedang terjadi. Analisis dilakukan dengan tujuan untuk memberikan wawasan yang seakurat mungkin mengenai kondisi yang ada, daripada memberikan jawaban yang pasti tentang apa yang akan terjadi. Selain itu, analisis melibatkan pemeriksaan nilai-nilai data yang ada untuk memahami tren atau pola yang mendasarinya. Dengan menganalisis situasi tertentu, analisis memberikan informasi penting untuk perencanaan dan pengambilan keputusan. (Mukhlisin dkk., 2020).

## 2.3 Curah Hujan

Curah hujan adalah jumlah air yang turun ke permukaan tanah selama periode tertentu. Data curah hujan ini dicatat oleh BMKG (Badan Meteorologi Klimatologi dan Geofisika) setiap harinya. Faktor-faktor seperti suhu, kelembaban udara, kecepatan angin, dan aktivitas matahari mempengaruhi curah hujan. Ketepatan dalam memprediksi cuaca sangat penting karena cuaca memiliki (Aulia Herdhyanti dkk., 2022).

Ketinggian air hujan yang terkumpul di tempat yang datar dan tidak menguap, tidak meresap, atau mengalir disebut juga curah hujan. Dalam bahasa internasional, curah hujan diukur dalam milimeter atau inchi, tetapi di Indonesia digunakan milimeter (mm). Satu milimeter mewakili curah hujan pada area satu meter persegi dengan volume air satu milimeter atau satu liter. (Putramulyo, S., Alaa, S., 2018).

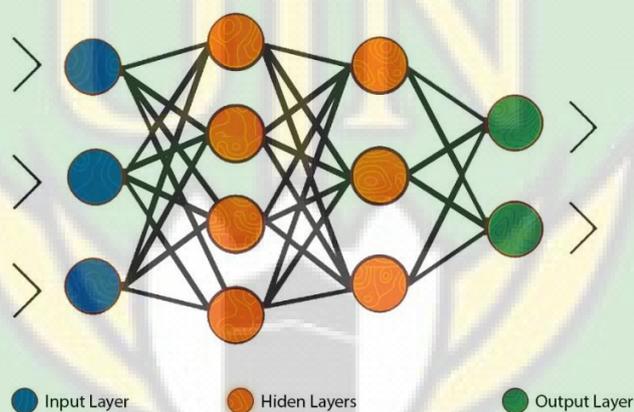
Intensitas curah hujan adalah jumlah curah hujan dalam satuan waktu tertentu, yang biasanya ditunjukkan dalam milimeter per jam, milimeter per hari, milimeter per tahun, dan seterusnya. Istilah seperti hujan jam-jaman, harian, tahunan, dan sebagainya sering digunakan untuk menggambarkan jumlah hujan yang terjadi. Nilai maksimum, minimum, dan rata-rata biasanya digunakan untuk analisis (Anwar, S., 2017).

Curah hujan umumnya diukur dalam satuan milimeter per meter persegi (mm/h). Satuan ini mengindikasikan jumlah air hujan yang jatuh dalam rentang waktu tertentu di suatu wilayah. Misalnya, jika curah hujan adalah 10 mm/h, itu berarti dalam satu jam, 10 milimeter air hujan diperkirakan akan turun di setiap meter persegi wilayah yang diamati. Sejumlah air hujan sebanyak 1 liter pada setiap luasan 1 meter atau air hujan yang jatuh pada ketinggian 1 milimeter pada seluas 1 meter tidak menguap, meresap, atau mengalir disebut curah hujan 1 (satu) milimeter (Bmkg, 2008).

#### **2.4 Long short Term Memory**

Long Short-Term Memory merupakan Salah satu jenis pengembangan dari Reccurent Neural Network (RNN). arsitektur LSTM telah dibuat untuk mengatasi kekurangan yang terjadi pada RNN konvensional, terutama terkait dengan manajemen informasi jangka panjang dalam urutan data. LSTM diciptakan untuk menyelesaikan masalah Vanishing Gradient yang sering terjadi pada RNN konvensional. Vanishing Gradient terjadi karena nilai bobot tidak berubah ketika gradient semakin mengecil hingga lapisan terakhir. Sebaliknya, ketika gradient semakin besar, nilai bobot pada beberapa lapisan juga meningkat, sehingga algoritma optimasi menjadi Divergen atau Konvergen. (Wiranda dan Sadikin, 2019).

LSTM diperkenalkan pertama kali oleh Sepp Hochreiter dan Juergen Schmidhuber pada tahun 1997, dan sejak saat itu sangat populer dalam berbagai aplikasi, terutama dalam prediksi deret waktu, pemodelan urutan, dan pemrosesan bahasa alami (Zahara dkk, 2019). Karena setiap neuron LSTM memiliki beberapa gerbang yang mengatur memorinya sendiri, LSTM dapat mempelajari data mana yang akan disimpan dan mana yang akan dibuang. Sel LSTM dapat menghubungkan informasi sebelumnya dengan informasi selanjutnya, dan efektivitas penyimpanan informasi yang panjang sangat dibutuhkan dalam pengolahan data deret waktu. LSTM adalah jenis arsitektur yang memiliki struktur dengan Input layer, Hidden layer, dan Output layer (Jiayu Qiu dkk, 2020).



Gambar 2. 1 Struktur Layer LSTM

Sumber : Colah, 2015

- **Input Layer:** Ini adalah lapisan pertama dalam jaringan LSTM yang menerima input dari dataset. Setiap node di layer ini mewakili satu fitur atau variabel input. Pada LSTM, input ini bisa terdiri dari urutan data atau deret waktu.
- **Hidden Layer:** Jaringan LSTM memiliki struktur yang lebih kompleks dibandingkan dengan RNN biasa, dengan adanya beberapa unit memori internal yang disebut "sel memori" atau "cell state". Hidden layer dalam LSTM terdiri dari sel-sel memori ini yang berinteraksi satu sama lain secara kompleks melalui gerbang (gate) untuk menyimpan informasi jangka panjang dan pendek.

- **Output Layer:** Ini adalah lapisan terakhir dari jaringan LSTM yang menghasilkan output berdasarkan representasi yang telah diproses oleh sel-sel memori dan informasi yang diperoleh dari data masukan. Output dapat berupa prediksi, klasifikasi, atau tugas lainnya tergantung pada jenis masalah yang sedang diselesaikan.

Dalam setiap langkah waktu, selain mempertimbangkan informasi yang tersimpan dalam memori jangka panjang dan pendek, LSTM menghitung output berdasarkan input saat ini.

Dalam LSTM, gerbang seperti input gate, forget gate, dan output gate, mengontrol aliran informasi yang masuk dan keluar dari sel memori, memungkinkan LSTM untuk mengelola informasi dalam urutan data dengan cara yang lebih cermat.

Perbedaan penting antara LSTM dan RNN tradisional adalah adanya komponen yang disebut "Gate". LSTM menyediakan tiga jenis gate utama, yaitu forget gate, input gate, output gate. Ketiga gerbang ini di rancang untuk dapat membaca, menyimpan, dan memperbaharui informasi terdahulu. Berikut penjelasan masing masing gate :

- **Forget Gate:** Gerbang ini memungkinkan LSTM memutuskan informasi mana yang harus dilupakan dari memori. Hal ini berguna untuk menghapus informasi yang tidak perlu atau berlebihan dari penyimpanan jangka Panjang. Untuk melakukannya, output biner dari dua input  $x(t)$  dan  $h(t-1)$  dikalikan dengan matriks bobot dan ditambahkan dengan nilai bias. Nilai bias kemudian dilewatkan melalui fungsi aktivasi, yang menghasilkan output biner. Output bernilai 0 menunjukkan bahwa informasi tersebut tidak berguna lagi dan dapat dihapus; sebaliknya, output bernilai 1 menunjukkan bahwa informasi tersebut disimpan untuk digunakan di masa mendatang.
- **Input Gate:** Gerbang ini memungkinkan LSTM menentukan informasi baru mana yang harus dimasukkan ke dalam memori. Informasi-informasi baru ini akan ditambahkan ke dalam memori jangka panjang Input gate menambah informasi penting ke cell state.

Pertama, fungsi sigmoid digunakan untuk mengatur informasi dan menyaring nilai yang akan disimpan; proses ini mirip dengan forget gate yang menggunakan input  $h(t-1)$  dan  $x(t)$ .

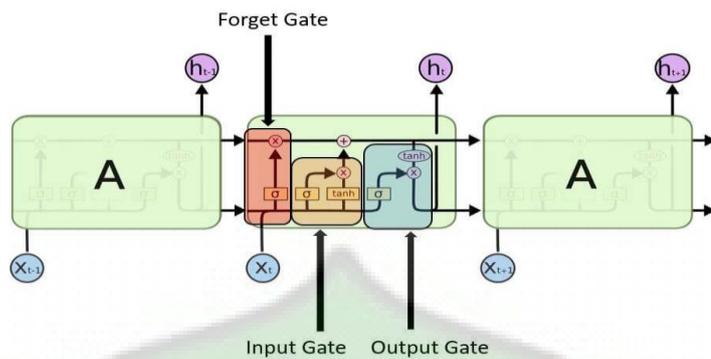
Kemudian, fungsi tanh digunakan untuk membuat vektor dengan output dari -1 hingga +1, yang mengandung semua nilai yang mungkin untuk  $h(t-1)$  dan  $x(t)$ . Terakhir, nilai-nilai vektor dan nilai-nilai yang diatur dikalikan untuk mendapatkan informasi yang bermanfaat.

- Output Gate: Gerbang ini mengatur informasi apa yang akan dikeluarkan dari unit memori LSTM. Pemahaman memori jangka panjang memungkinkan LSTM untuk menentukan informasi mana yang akan digunakan sebagai keluaran.

Tugasnya adalah mengumpulkan informasi penting dari keadaan sel saat ini dan menyajikannya sebagai nilai keluaran yang dihasilkan oleh gate output. Pertama, fungsi tanh digunakan pada sel untuk menghasilkan vektor.

Kemudian, fungsi sigmoid digunakan untuk mengatur informasi dan menyaring nilai-nilai yang akan disimpan dengan menggunakan input  $h_{t-1}$  dan  $x_t$ . Terakhir, nilai vektor dan nilai yang diatur dikalikan untuk dikirim sebagai input dan output ke sel berikutnya.

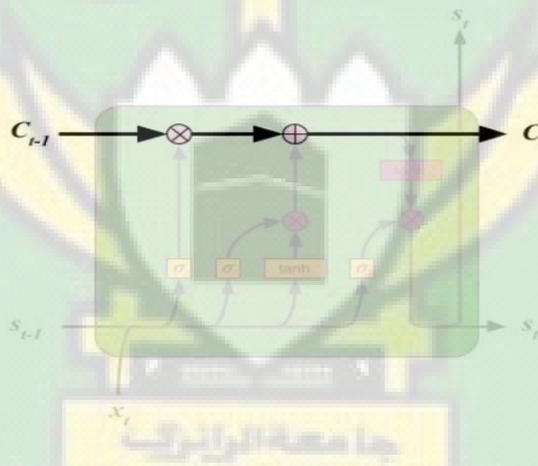
Struktur gerbang model *Long Short Term Memory* (LSTM) menyaring data untuk menjaga dan memperbaharui status memori sel. Komponen-komponen seperti input gate, forget gate, dan output gate adalah bagian dari struktur gerbang ini. Setiap sel memori memiliki tiga lapisan sigmoid dan satu lapisan tanh yang berperan dalam proses ini (Colah, 2015).



Gambar 2. 2 Arsitektur LSTM

Sumber : Colah, 2015

Seperti yang ditunjukkan di bawah ini, jalur yang menghubungkan konteks lama ( $C_{t-1}$ ) ke konteks baru ( $C_t$ ) di bagian atas modul LSTM adalah konsep utama dari LSTM.



Gambar 2. 3 Memory cell

Sumber : Colah, 2015

Konteks  $C_t$  juga dikenal sebagai kondisi sel atau memori sel. Nilai di konteks lama dapat dengan mudah ditransfer ke konteks baru dengan beberapa perubahan yang diperlukan.

Sebagai perancang jaringan LSTM, kami memiliki kemampuan untuk menentukan jumlah elemen dalam vektor konteks. Gagasan di balik ini adalah bahwa setiap elemen akan menangkap karakteristik tertentu dari masukan.

Dalam kasus pemrosesan bahasa alami untuk bahasa Inggris, karakteristik ini dapat mencakup jenis kelamin subjek, apakah itu tunggal atau jamak, dan sebagainya. Melalui proses pelatihan, jaringan LSTM akan secara mandiri mengungkap dan mempelajari fitur-fitur ini.

Fungsi aktivasi sigmoid, yang mengatur jumlah informasi yang dapat melalui, berfungsi untuk mengubah nilai antara -1 dan 1 menjadi nilai antara 1 dan 0. Hal ini digunakan untuk memperbarui atau melupakan data karena setiap angka yang dikalikan 0 hasilnya adalah 0, sehingga nilai akan hilang atau dilupakan, sedangkan setiap angka yang dikalikan 1 hasilnya akan tetap sama, sehingga nilai tersebut dapat disimpan.

Selain fungsi sigmoid, fungsi aktivasi tanh mengontrol nilai yang melalui jaringan selalu berada dekat antara -1 dan 1.

Berikut fungsi sigmoid :

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

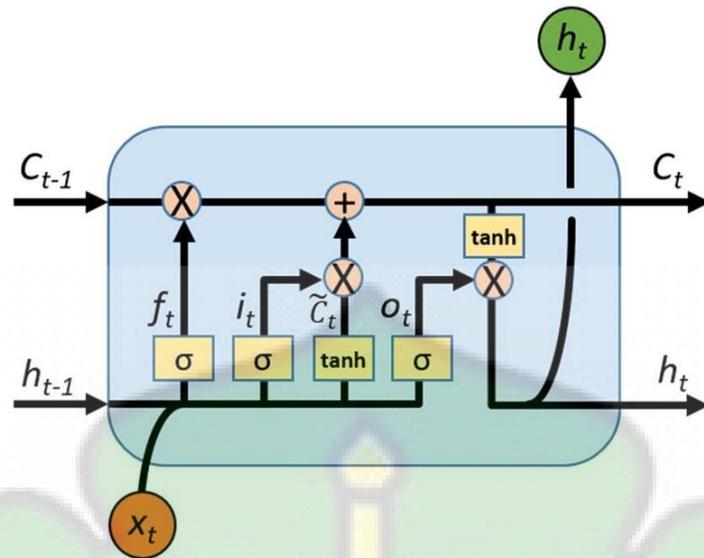
Fungsi tanh ditunjukkan sebagai berikut:

$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.2)$$

Keterangan:

$\sigma$  = Fungsi aktivasi sigmoid

$x$  = Data input

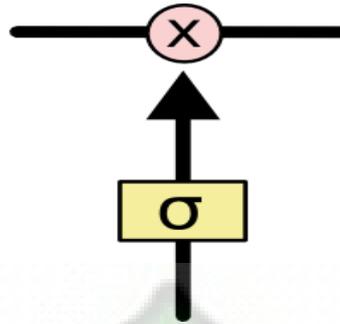


Gambar 2. 4 Struktur Sel LSTM

Sumber : Colah, 2015

Gambar 2.4 menunjukkan struktur dalam satu sel *Long Short Term Memory* (LSTM), terutama jalur yang menghubungkan sel memori lama ( $C_{t-1}$ ) ke sel memori baru ( $C_t$ ). Jalur ini merupakan kunci sel memori *Long Short Term Memory* (LSTM), dan merupakan garis horizontal yang menghubungkan semua level output LSTM mendefinisikan sel memori. Nilai sel memori lama dapat dialihkan ke sel memori baru dengan sedikit usaha (Colah, 2015).

Salah satu hasil dari *Long Short Term Memory* (LSTM), lapisan sigmoid dapat menambah atau menghapus data masa lalu yang masuk ke sel saat ini. Besarnya setiap komponen yang harus dibiarkan masuk ditunjukkan dengan angka antara nol dan satu, di mana nol adalah "tidak lolos" dan satu adalah "lolos". (Colah, 2015).



Gambar 2. 5 Lapisan dari Sigmoid

Sumber : Colah, 2015

langkah kerja *Long Short Term Memory* (LSTM) dijelaskan secara rinci pada gambar 2.2. Pilih data yang ingin dihapus dari status sel. Forget gate, lapisan sigmoid yang dapat dilihat pada  $h_{t-1}$  dan  $x_t$ , menentukan hasilnya, yang berkisar antara 0 dan 1. Hasil dapat dibulatkan menjadi 1 jika keluaran sigmoid lebih besar dari 0,5 dan menjadi 0 jika kurang dari 0,5. (Colah, 2015)



Gambar 2. 6 Alur informasi *forget Gate*

Sumber : Colah, 2015

Persamaan berikut menguraikan persamaan Forget Gate :

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

Keterangan :

$f_t$  = forget gate

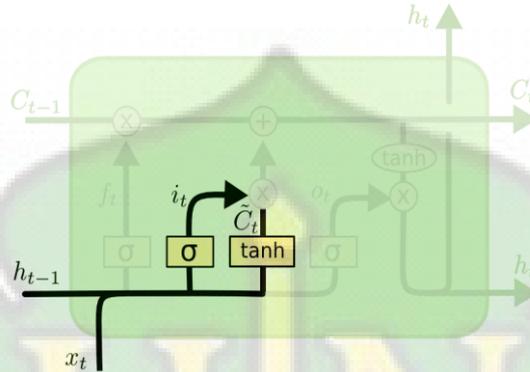
$\sigma$  = fungsi sigmoid

$W_f$  = nilai bobot untuk *forget gate*

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

$x_t$  = nilai input pada orde ke t

$b_f$  = nilai bias *forget gate*



Gambar 2. 7 Alur Informasi *Input Gate*

Sumber : Colah, 2015

Persamaan berikut menguraikan persamaan *Input Gate* :

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

keterangan :

$i_t$  = *input gate*

$\sigma$  = fungsi sigmoid

$W_i$  = nilai bobot untuk *input gate*

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

$x_t$  = nilai input pada orde ke t

$b_i$  = nilai bias *input gate*

Persamaan kandidat baru dituliskan pada Persamaan 2.5

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

Keterangan :

$\tilde{C}_t$  = nilai baru yang dapat ditambahkan ke *cell state*

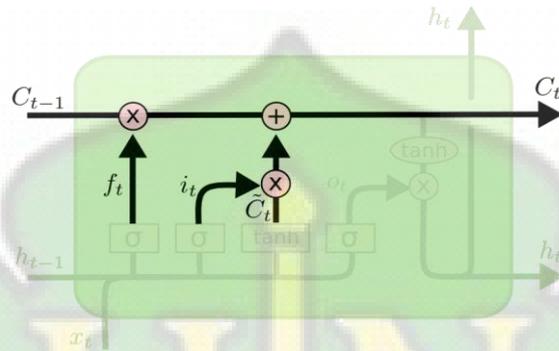
$\tanh$  = fungsi tanh

$W_c$  = nilai bobot untuk *cell state*

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

$x_t$  = nilai input pada orde ke t

$b_c$  = nilai bias *cell state*



Gambar 2. 8 Alur *cell state*

Sumber : Colah, 2015

Seperti yang ditunjukkan pada gambar 2.8, langkah selanjutnya adalah memperbarui *cell state*. Ini dilakukan dengan mengalihkan forget gate ke *cell state* dan kemudian menambah langkah kedua,  $i_t * C_t$ . Persamaan berikut menunjukkan persamaan *cell state*:

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

Keterangan :

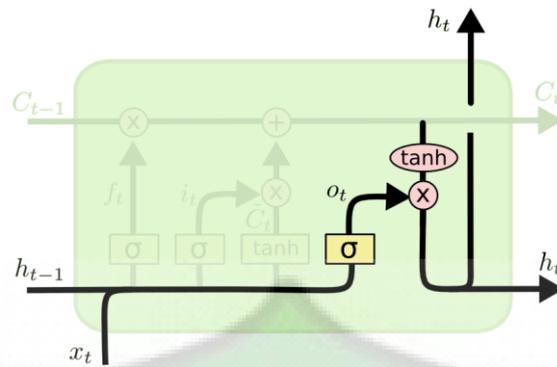
$C_t$  = *cell state*

$f_t$  = *forget gate*

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

$i_t$  = *input gate*

$\tilde{C}_t$  = nilai baru yang dapat ditambahkan ke *cell state*



Gambar 2. 9 Alur Informasi *Output gate*

Sumber : Colah, 2015

Seperti yang ditunjukkan pada gambar 2.9, langkah terakhir dalam metode LSTM ini adalah menentukan hasil output. Lapisan sigmoid menentukan bagian cell state mana yang akan dihasilkan menjadi output. Berikut persamaan dari *output gate*:

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

Keterangan :

$O_t$  = *Output gate*

$\sigma$  = Fungsi sigmoid

$W_o$  = nilai bobot *output gate*

$h_{t-1}$  = nilai *output* sebelum order ke  $t$

$x_t$  = nilai *input* pada order ke  $t$

$b_o$  = nilai bias *output gate*

Setelah mendapatkan nilai dari gate output, langkah selanjutnya adalah melewati *cell state* melalui *tanh* untuk menghasilkan nilai dalam rentang -1 hingga 1. Setelah itu, nilai tersebut ditransfer ke *output gate* dari lapisan *sigmoid*. Persamaan berikut digunakan untuk membuat persamaan nilai *output* order  $t$ :

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

keterangan :

$h_t$  = nilai output order t

$O_t$  = output gate

$\tanh$  = fungsi tanh

$(C_t)$  = cell state

#### 2.4.1. Proses Long Short Term Memory (LSTM)

Setelah mendapatkan data pada tahap Normalisasi, proses inialisasi bobot dilakukan untuk gate forget, gate input, cell state, dan gate output. Menginisialisasikan nilai bobot untuk setiap gate ( $W_f$ ,  $W_i$ ,  $W_{ct}$ ,  $W_o$ ), serta bias ( $b_f$ ,  $b_i$ ,  $b_{ct}$ ,  $b_o$ ). Untuk mencari nilai bobot  $W$ , gunakan persamaan berikut.

$$W = \left( -\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}} \right) \quad (2.9)$$

Keterangan :

$W$  = Bobot

$d$  = Banyaknya Data

$$W = -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}$$

$$W = (-0.447, 0.447)$$

Nilai  $d = 5$  diperoleh dari nilai inputan pada bulan Januari 2021

**Perhitungan hari pertama pada tanggal 1 Januari 2021 :**

Diketahui :

$$h_{t-1} = 0$$

$$\text{bobot} = 0,447$$

$$\text{bias forget gate, input gate, } \bar{C}_t, \text{ dan output gate} = 1,0.5,0,0,1$$

- **Perhitungan *Gate Forget***

Untuk algoritma LSTM, gate forget memungkinkan masuknya informasi dan menentukan informasi apa yang akan dihilangkan dari cell state. Layer sigmoid menghasilkan output yang berkisar antara 0 dan 1. Perhitungan *Forget Gate* menggunakan rumus pada persamaan (2.3) sebagai berikut :

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 &= \sigma(0.447 * [0, 0.001462] + 1) \\
 &= \sigma(0 + 0.447 * 0.001462 + 1) \\
 &= \sigma(1.000653) \\
 &= \frac{1}{(1 + e^{-1.000653})} \\
 &= 0.7311
 \end{aligned}$$

- **Menghitung *Gate Input* Dan Kandidat Baru *Cell State***

$$\begin{aligned}
 I_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 &= \sigma(0.447 * [0, 0.001462] + 0,5) \\
 &= \sigma(0 + 0.447 * 0.001462 + 0,5) \\
 &= \sigma(0.500653) \\
 &= \frac{1}{(1 + e^{-0.500653})} \\
 &= 0.6226
 \end{aligned}$$

$$\begin{aligned}
 \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
 &= \tanh(0.447 * [0, 0.001462] + 1) \\
 &= \tanh(0 + 0.447 * 0.001462 + 1) \\
 &= \tanh(1.0006) \\
 &= 2 * \frac{1}{(1 + e^{-2 * 1.0006})} - 1 \\
 &= 0.7618
 \end{aligned}$$

- **Gate Output**

$$\begin{aligned}
 O_t &= \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \\
 &= \sigma (0.447 * [0, 0.001462] + 0,1) \\
 &= \sigma (0 + 0.447 * 0.001462 + 0,1) \\
 &= \sigma (0.100697) \\
 &= \frac{1}{(1 + e^{-0.100697})} \\
 &= 0.5251
 \end{aligned}$$

- **Cell State**

$$\begin{aligned}
 C_t &= f_t * C_{t-1} + i_t * \bar{C}_t \\
 &= 0.7311 * 0 + 0.6226 * 0.7618 \\
 &= 0.3467
 \end{aligned}$$

- **Hidden state**

$$\begin{aligned}
 h_t &= o_t * \tanh(C_t) \\
 &= 0.5251 * \tanh(0.3467) \\
 &= 0.1750
 \end{aligned}$$

**Perhitungan hari kedua :**

Dengan menggunakan  $C_t$  dan  $h_t$  dari hari pertama, serta nilai bobot dan bias yang sama, diperoleh:

- **Gate Forget**

$$\begin{aligned}
 f_t &= \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \\
 &= \sigma (0.447 * [0.1750, 0.003037] + 1) \\
 &= \sigma (0.1750 + 0.447 * 0.003037 + 1) \\
 &= \sigma (1.176357) \\
 &= \frac{1}{(1 + e^{-1.176357})} \\
 &= 0.7642
 \end{aligned}$$

- Menghitung *Input Gate* dan Kandidat baru untuk *Cell State*

$$\begin{aligned}
 I_t &= \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \\
 &= \sigma (0.447 * [0.1750 , 0.003037] + 0.5 ) \\
 &= \sigma (0.1750 + 0.447 * 0.003037 + 0.5) \\
 &= \sigma (0.676357) \\
 &= \frac{1}{(1 + e^{-0.676357})} \\
 &= 0.6629
 \end{aligned}$$

$$\begin{aligned}
 \bar{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
 &= \tanh (0.447 * [0.1750 , 0.003037] + 1) \\
 &= \tanh (0.1750 + 0.447 * 0.003037 + 1) \\
 &= \tanh ( 1.1763) \\
 &= 2 * \frac{1}{(1 + e^{-2*1.1763})} - 1 \\
 &= 0.8262
 \end{aligned}$$

- *Gate Output*

$$\begin{aligned}
 O_t &= \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \\
 &= \sigma (0.447 * [0.1750 , 0.003037] + 0,1) \\
 &= \sigma (0.1750 + 0.447 * 0.003037 + 0,1) \\
 &= \sigma (0.276357) \\
 &= \frac{1}{(1 + e^{-0.276357})} \\
 &= 0.5686
 \end{aligned}$$

- *Cell State*

$$\begin{aligned}
 C_t &= f_t * C_{t-1} + i_t * \bar{C}_t \langle \rangle \\
 &= 0.7642*0.3467+0.6629*0.8262 \\
 &= 0.8126
 \end{aligned}$$

- **Hidden State**

$$h_t = o_t * \tanh(Ct)$$

$$= 0.5686 * \tanh(0.8126)$$

$$= 0.3815$$

Proses perhitungan yang disebutkan di atas berlanjut hingga hari kelima. Output dari hari kelima ini akan digunakan untuk membandingkan dengan data awal, yang akhirnya menghasilkan tabel berikut:

Tabel 2. 2 Hasil perhitungan manual

Tanggal	Data Curah Hujan	Normalisasi	$h_t$
1/1/2021	13	0,001462	0.1750
2/1/2021	27	0,003037	0.3815
3/1/2021	1	0,000112	0.5281
4/1/2021	7	0.000787	0.6112
5/1/2021	90,5	0,010182	0.6292

## 2.5 Deret Waktu (*Time Series*)

Deret waktu (*time series*) adalah himpunan observasi data yang diurutkan secara berurutan dalam waktu. Metode deret waktu menggunakan analisis pola hubungan antara variabel yang akan diperkirakan dengan variabel waktu untuk melakukan peramalan. Dalam peramalan deret waktu, perlu diperhatikan jenis atau pola data. Secara umum, empat jenis pola data seri waktu adalah horizontal, trend, musiman, dan siklis.

Pola tren merupakan kecenderungan arah data dalam jangka panjang, yang dapat berupa kenaikan atau penurunan; namun, pola horizontal adalah kejadian yang tidak terduga dan acak yang dapat mempengaruhi fluktuasi rangkaian waktu. Pola siklis adalah perubahan data selama lebih dari satu tahun, seperti pola musiman, triwulan, kuartalan, bulanan, mingguan, atau harian. (Hanke & Winchern, 2005).

## 2.6 *Deep Learning*

Jaringan deep learning terdiri dari berbagai lapisan, masing-masing terdiri dari sejumlah node. Sebuah node hanya berfungsi sebagai tempat perhitungan, dan sebuah node input digabungkan dengan bobot. Setelah input dan bobot ini dijumlahkan, jumlahnya melewati tahap yang disebut fungsi aktivasi node. Tahap ini menentukan seberapa jauh sinyal bergerak melalui jaringan dan mempengaruhi hasil akhir. Deep Learning memiliki lebih banyak lapisan tersembunyi daripada neural network. Jika ada lebih dari tiga lapisan (termasuk input dan output) maka dianggap sebagai "deep learning". Oleh karena itu, Deep Learning dapat didefinisikan secara teknis sebagai pembelajaran mesin dengan lebih dari satu lapisan tersembunyi. (Rizki, M., dkk. 2020).

Deep Learning adalah teknik cerdas buatan yang digunakan untuk memodelkan hubungan yang kompleks antara input dan output. Selain itu, memiliki beberapa fitur unik yang membedakannya dari teknik pelatihan model dan konstruksi model lainnya. (Yunanto dkk, 2020).

## 2.7 **Jaringan Saraf Tiruan (JST)**

JST atau jaringan saraf tiruan, juga dikenal sebagai jaringan saraf tiruan (ANN), simulasi jaringan saraf (SNN), atau jaringan saraf tiruan (NN), jaringan yang terdiri dari sejumlah unit pemroses yang sangat kecil yang dimodelkan dari sistem saraf manusia. Singkatnya, JST adalah alat pemodelan data statistik non-linier yang dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk mengidentifikasi pola pada data.

Dengan menggunakan minimal satu lapis tersembunyi dan fungsi aktivasi non-linear, JST dapat memodelkan seluruh fungsi Boreal terukur dari satu dimensi ke dimensi lain, menurut "teorema penaksiran universal" (Nasution dkk, 2015).

## 2.8 *Reccurent Neural Network (RNN)*

Recurrent Neural Network (RNN) adalah jaringan saraf berulang yang terstruktur untuk memproses data sekuensial seperti data deret waktu. RNN dikatakan sebagai saraf berulang jaringan karena keluaran dari lapisan tersembunyi sebelumnya akan digunakan kembali sebagai masukan data untuk diproses lebih lanjut. RNN ini terdiri dari lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Ciri khas RNN dalam melakukan prediksi adalah tidak hanya menggunakan satu input saja suatu waktu tetapi memerlukan masukan dan masukan sebelumnya. Oleh karena itu inputnya saling berhubungan dan dapat memberikan informasi pada lapisan tersembunyi (Sen, dkk., 2020).

## 2.9 **Normalisasi**

Normalisasi data merupakan proses penting dalam machine learning dan deep learning, karena membantu mengunggah model machine learning untuk melakukan prediksi yang lebih akurat dan efisien. Normalisasi memiliki beberapa keuntungan, antara lain:

- Menghilangkan data tidak terstruktur dan redundansi: Normalisasi membuat data lebih standard dan mudah dianalisis oleh model machine learning
- Memastikan penyimpanan data logis: Normalisasi memastikan bahwa data tetap berada dalam rentang yang sama, sehingga model machine learning dapat fokus pada fitur yang penting dari data
- Mengurangi persyaratan untuk algoritma khusus, seperti KNN dan ANN: Normalisasi membuat data lebih sesuai dengan asumsi distribusi yang diperlukan oleh algoritma machine learning, sehingga mereka dapat berfungsi dengan lebih efisien.

Normalisasi data dapat dilakukan menggunakan beberapa metode, seperti normalisasi *min-max* dan standardisasi. Ketika data memiliki skala yang berbeda, normalisasi *min-max* sering digunakan, tetapi standardisasi lebih baik untuk data dengan distribusi yang sama. ( Adhiedashh , 2023).

$$X' \frac{(x - X_{\min})}{(X_{\max} - X_{\min})} \quad (2.10)$$

## 2.10 RMSE

RMSE digunakan untuk menghitung besaran error dalam memprediksi suatu data. RMSE menghitung perbedaan antara nilai sesungguhnya dan nilai yang diharapkan, kemudian membagi hasil total penjumlahan yang diperoleh dengan banyaknya waktu prediksi, dan kemudian menarik akarnya. Persamaan berikut menunjukkan perhitungan Root Mean Square Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{Y}_i)^2}{1}} \quad (2.11)$$

Keterangan :

$\tilde{y}_i$  : Nilai dari hasil peramalan

$y_i$  : Nilai aktual / sebenarnya

N : Jumlah data

## 2.11 MAE

*Mean Absolute Error* (MAE) adalah MAE (Mean Absolute Error) adalah suatu metrik evaluasi kinerja yang umum digunakan dalam statistika dan pembelajaran mesin untuk mengukur seberapa dekat prediksi suatu model dengan nilai sebenarnya (ground truth). MAE diukur sebagai rata-rata dari selisih absolut antara prediksi dan nilai sebenarnya. Persamaannya berikut menunjukkan perhitungan *Mean Absolute Error* (MAE) :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\tilde{y}_i - y_i| \quad (2.12)$$

Keterangan:

$\tilde{y}_i$  : Nilai hasil peramalan

$y_i$  = nilai sebenarnya dari observasi ke- $i$

n = Jumlah peramalan yang terlibat

## 2.12 *Tools*

Proses analisis data membutuhkan *Tools* atau alat bantu yang membantu peneliti dalam menyelesaikan penelitian ini dengan menggunakan bahasa pemrograman Python, metode LSTM dan *Google Colaboratory* sebagai *workflows notebooks*.

### 2.12.1 *Python*

Python merupakan bahasa pemrograman yang populer untuk pemrosesan data. Banyak ilmuwan data yang menggunakan Python karena relatif mudah digunakan (Prasetya dkk., 2020). Selain itu, library yang tersedia untuk mengolah data ini cukup banyak sehingga memudahkan proses analisis bagi peneliti dalam penelitian ini.

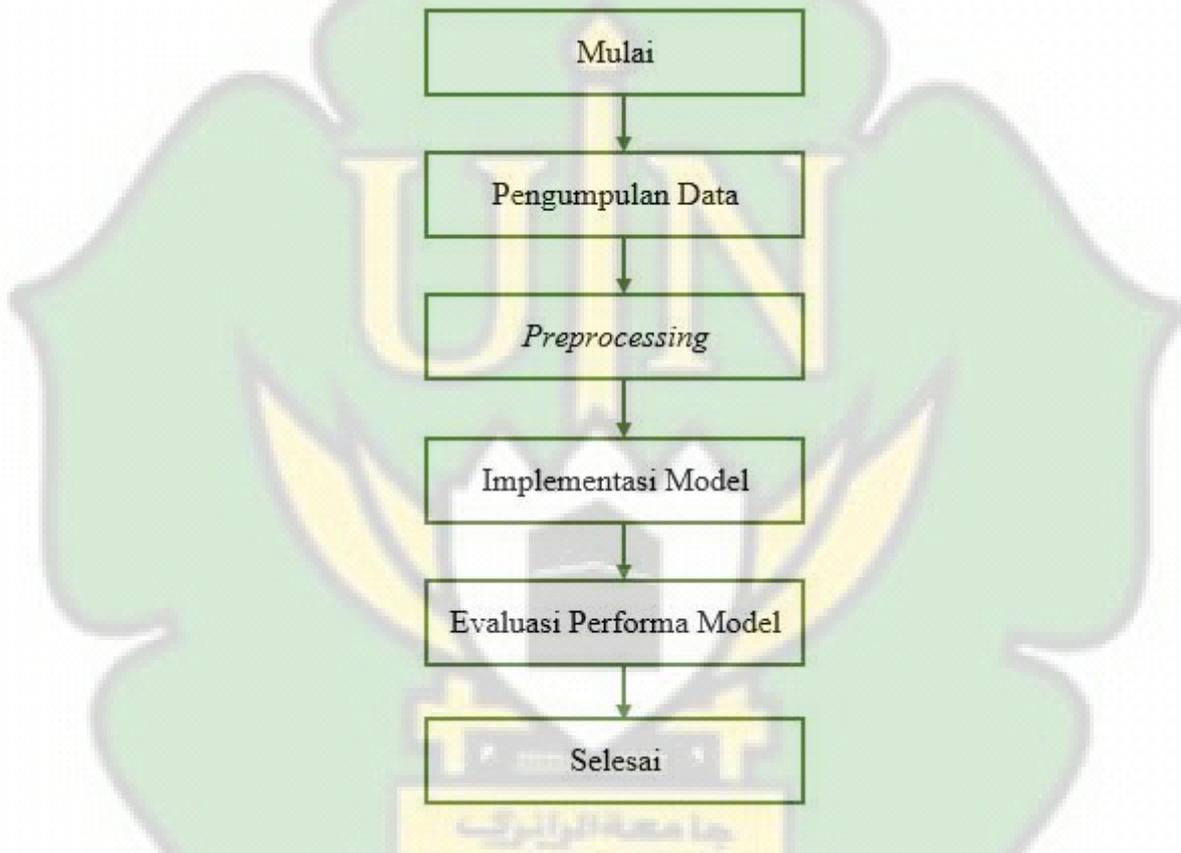
### 2.12.2 *Jupyter Notebook*

*Jupyter notebook* merupakan suatu aplikasi *web open-source* yang memungkinkan pengguna untuk membuat dan berbagi dokumen yang mengandung kode interaktif, persentasi, teks, dan elemen visual lainnya. *Jupyter Notebook* mendukung berbagai bahasa pemrograman, tetapi yang paling sering digunakan untuk bahasa seperti Python, R, dan Julia. Dokumen *Jupyter* disusun dalam sel, yang dapat berisi kode, teks, atau elemen visual.

## **BAB III METODE PENELITIAN**

### **3.1 Tahapan Penelitian**

Langkah-langkah yang di ambil dalam penelitian ini menjelaskan proses penelitian yang di gunakan oleh penulis. Detail dari langkah-langkah tersebut dapat di di lihat pada gambar 3.1.



Gambar 3. 1 Diagram Metode Penelitian *Long Short Term Memory*

### **3.2 Pengumpulan Data**

Gambar di atas menunjukkan bahwa tahap pertama dari penelitian adalah mengumpulkan data. Data yang digunakan penulis untuk penelitian ini adalah dataset curah hujan yang diambil dari situs web resmi BMKG (<https://dataonline.bmkg.go.id/>). Tampilan *dashboard* dari pusat database BMKG bisa di lihat pada gambar 3.2 dibawah ini.

Dashboard

Home » Dashboard

**RIWAYAT PENGUNDUHAN DATA**

10 ▾ baris   Pencarian

Tanggal dan Jam Unduh ▾	Jenis Data ▾	Parameter ▾	Periode Data ▾
18 Nov 2023 15:27:03	Data iklim harian	- Curah hujan	01 Apr 2010 s.d 30 Apr 2010
18 Nov 2023 15:26:32	Data iklim harian	- Curah hujan	01 Mar 2011 s.d 31 Mar 2011
18 Nov 2023 15:26:04	Data iklim harian	- Curah hujan	01 Mar 2010 s.d 31 Mar 2010
18 Nov 2023 15:25:55	Data iklim harian	- Curah hujan	01 Feb 2011 s.d 28 Feb 2011
18 Nov 2023 15:24:59	Data iklim harian	- Curah hujan	01 Feb 2010 s.d 28 Feb 2010

336

hari lagi  
Masa berlaku akun anda habis  
(25 Okt 2024)

Gambar 3. 2 Tampilan *Dashboard* Pusat *Database* BMKG

### 3.2.1 Dataset

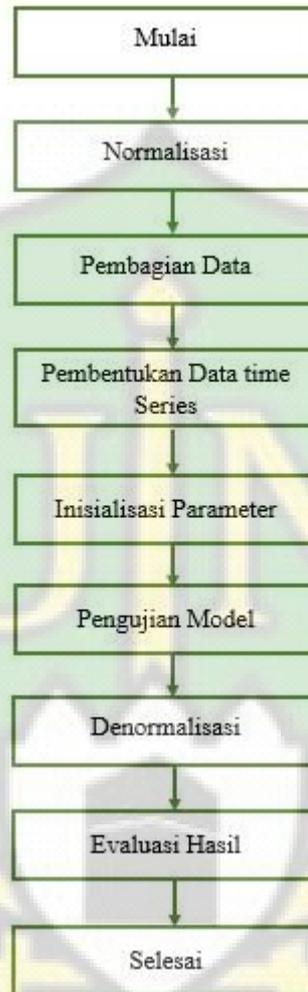
Data yang digunakan untuk manualisasi adalah data curah hujan dari Stasiun Sultan Iskandar Muda Aceh, yang memiliki 4283 dataset dari tahun 2010 hingga 2022. Tabel 3.1 di bawah ini menunjukkan salah satu dataset curah hujan di bulan Januari 2021, yang merupakan data curah hujan yang diperoleh dari BMKG.

Tabel 3. 1 Dataset curah hujan bulan januari tahun 2021

Tanggal	RR (Curah Hujan)
01-01-2021	13
02-01-2021	27
03-01-2021	1
04-01-2021	7
05-01-2021	90,5
06-01-2021	23,7
...	...
28-01-2021	22
29-01-2021	0
30-01-2021	0
31-01-2021	0

### 3.3 Preprocessing

Tahapan-tahapan *preprocessing* adalah sebagai berikut :



Gambar 3. 3 Alur *Preprocessing*

#### 3.3.1 Normalisasi

Penerapan modifikasi linier pada data awal memungkinkan pendekatan *Min-Max* untuk menormalkan data dengan lebih mudah menggunakan kelas preprosesing *MinMaxScaler* dari pustaka *scikit-learn* (*sklearn*), yang menghasilkan data dalam skala  $[0,1]$ . Rumus *MinMaxScaler* yang digunakan adalah sebagai berikut:

$$X' = \frac{(x - X_{min})}{(X_{max} - X_{min})} \quad (3.1)$$

Keterangan:

$X'$  = Hasil normalisasi data

$X_{max}$  = Data maksimum dari data  $X$

$X_{min}$  = Data minimum dari data  $X$

Berikut merupakan contoh perhitungan normalisasi :

Tabel 3. 2 Hasil Normalisasi Bulan Januari Tahun 2021

RR (Curah Hujan)	Normalisasi
13	0,001462
27	0,003037
1	0,000112
7	0.000787
90,5	0,010182
23,7	0.002666
...	...
0	0
22	0,002475
0	0
0	0
0	0

### 3.3.2 Pembagian Data

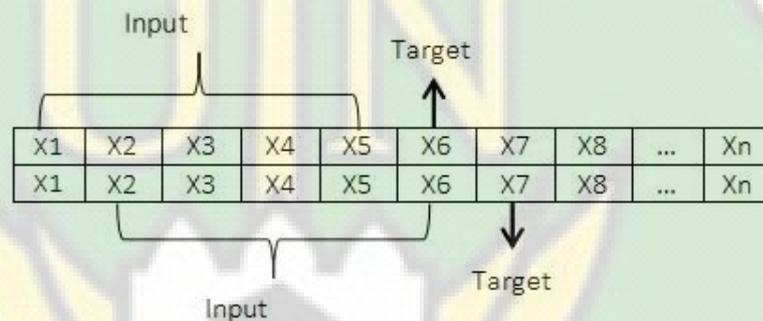
Pembagian data dilakukan menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*), dengan perbandingan 80% : 20%. Data latih digunakan untuk pembentukan dan pengembangan model, sementara data uji digunakan untuk melakukan pengujian beserta evaluasi kinerja model.

Tabel 3. 3 Pembagian Data

Satuan	Data latih	Data uji
Persen	80%	20%
Jumlah data	3444	839

### 3.3.3 Pembentukan Data *Time Series*

Proses pembentukan data time series atau rekonstruksi data melibatkan beberapa percobaan dengan jumlah pola time series mulai dari satu hingga lima. Sejumlah uji coba dilakukan dalam proses pembentukan deret waktu, dengan skenario yang melibatkan pembentukan pola dari data curah hujan harian dalam urutan sebagai berikut:  $\{X_t, X_{t+1}\}$ ,  $\{X_{t-1}, X_t, X_{t+1}\}$ ,  $\{X_{t-2}, X_{t-1}, X_t, X_{t+1}\}$ ,  $\{X_{t-3}, X_{t-2}, X_t, X_{t+1}\}$ ,  $\{X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}, X_t, X_{t+1}\}$ . Di sini,  $X_{t+1}$  berperan sebagai output atau target, sementara yang lainnya adalah variabel input. Berikut adalah gambaran ilustratif dari pola-pola time series tersebut.



Gambar 3. 4 Ilustrasi pola *time series*

Gambar pola seri waktu di atas menunjukkan inputan data dan targetnya, dengan lima pola seri waktu, yang berarti data yang digunakan sebanyak lima inputan dan data yang dicari setelah data kelima. Curah hujan (RR) ditandai dengan X

Tabel 3. 4 Data *Time Series* Curah Hujan

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	Target
0,001462	0,003037	0,000112	0.000787	0,010182	0.002666
0,003037	0,000112	0.000787	0,010182	0.002666	0.001406
0,000112	0.000787	0,010182	0.002666	0.001406	0,001338
0.000787	0,010182	0.002666	0.001406	0,001338	0,001473
0,010182	0.002666	0.001406	0,001338	0,001473	0,003544
...	...	...	...	...	...

### 3.3.4 Inisialisasi parameter

Parameter yang digunakan pada tahap ini termasuk dua parameter dengan skenario pengujian berikut:

1. Epoch : 25, 50, 75, 100, 150, 200
2. Batch size : 32

Epoch adalah satu siklus penuh melalui seluruh dataset pelatihan. Selama satu epoch, algoritma pelatihan memproses setiap sampel dalam dataset satu kali. Dalam proses pelatihan model, dataset biasanya dibagi menjadi batch-batch kecil (*mini-batches*) untuk mempercepat dan mengoptimalkan pelatihan. Selama satu epoch, model memproses semua batch tersebut hingga seluruh dataset telah digunakan untuk memperbarui bobot dan bias model. Tidak ada aturan pasti untuk menentukan parameter; sebaliknya, percobaan dilakukan dengan data dan model yang akan digunakan.

### 3.3.5 Pengujian Model

Tahap-tahap pengujian LSTM sebagai berikut :

- Pertama masukkan data uji. Model optimal dari proses pelatihan digunakan untuk melakukan pengujian yang bertujuan untuk menghitung nilai error jaringan yang dihasilkan oleh proses pelatihan. Nilai error yang kecil menunjukkan bahwa jaringan dapat diterapkan pada proses permalan.
- Menghitung semua fungsi gates unit pada setiap neuron memungkinkan proses pengujian dengan data uji. Ini dilakukan dengan menggunakan nilai  $h_t$  dan  $C_t$  optimal yang diperoleh dari proses pelatihan sebelum menguji proses dengan data latih menggunakan LSTM. Fungsi gate yang akan dihitung secara berurutan adalah forget gates ( $f_t$ ) dengan Persamaan (2.3), gate input ( $i_t$ ) dengan Persamaan (2.4) dan (2.5), gate state ( $C_t$ ) dengan Persamaan (2.6), gate output ( $O_t$ ) dengan Persamaan (2.7), dan akhirnya keluaran terakhir,  $h_t$  dengan Persamaan 2.8.

### 3.3.6 Denormalisasi

Selanjutnya, denormalisasi dilakukan untuk mengembalikan nilai data menjadi hasil nilai prediksi, yang dilakukan dengan menggunakan Persamaan 3.2. Denormalisasi adalah proses pengembalian hasil normalisasi data ke dalam data asli, juga dikenal sebagai data sebenarnya. Ini dilakukan untuk melihat hasil prediksi dengan membandingkan data sebenarnya. Persamaan berikut menyediakan rumus untuk menunjukkan denormalisasi dalam jangkauan tertentu:

$$X_t = x (X_{\max} - X_{\min}) + X_{\min} \quad (3.2)$$

Keterangan :

$x_t$  = nilai data normalisasi

$x$  = hasil output

$X_{\max}$  = nilai maksimal data aktual keseluruhan

$X_{\min}$  = nilai minimal data aktual keseluruhan

Contoh perhitungan manual denormalisasi :

$$\begin{aligned} X &= 0.1750 (8888 - 0) + 0 \\ &= 0.1750 \times 8888 \\ &= 1555.4 \end{aligned}$$

Tabel 3. 5 Hasil Denormalisasi

Tanggal	Data Curah Hujan	Normalisasi	$h_t$	Denormalisasi
1/1/2021	13	0,001462	0.1750	1555.4
2/1/2021	27	0,003037	0.3815	3390.7
3/1/2021	1	0,000112	0.5281	4693.7
4/1/2021	7	0.000787	0.6112	5432.3
5/1/2021	90,5	0,010182	0.6292	5592.3

Dari perhitungan diatas didapatkan tabel seperti berikut:

Tabel 3. 6 Nilai rangkaian lima waktu

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	Target
1555.4	3390.7	4693.7	5432.3	5592.3	23,7

Tabel 3.6 di atas menunjukkan pola rangkaian lima waktu, yang menunjukkan bahwa data terakhir ( $X_5$ ) adalah hasil akhir yang akan digunakan untuk memprediksi dan dibandingkan dengan nilai targetnya. Perhitungan yang ditunjukkan di atas adalah contoh penghitungan dengan melakukan satu iterasi pada data pertama.

### 3.3.7 Evaluasi Hasil

Evaluasi model dilakukan untuk mengetahui apakah model bekerja dengan baik. Jika evaluasi dilakukan dengan data yang sama selama proses pelatihan, hal itu akan menyebabkan overfitting, yang menyebabkan data tidak dapat digeneralisasikan karena model dapat mengingat data yang digunakan selama proses pelatihan.

Pada langkah ini, akurasi peramalan dilakukan dengan menggunakan Root Mean Square Error (RMSE). Evaluasi hasil dilakukan untuk menentukan apakah nilai dari prediksi yang telah dilakukan dengan metode di atas menghasilkan hasil yang optimal. Contoh perhitungan di atas menggunakan data dari hari ke satu hingga hari ke lima. Hasil denormalisasi dari data kelima digunakan untuk pengujian yang akan dilakukan untuk menentukan hari keenam. Untuk mengevaluasi keakuratan hasil prediksi dalam penelitian ini, penulis menggunakan RMSE dan MAE.

## 3.4 Alat Bantu Penelitian

Perangkat lunak dan perangkat keras komputer digunakan untuk membantu penelitian ini. Perangkat keras yang digunakan adalah unit HP laptop 14-cm0xxx, dengan spesifikasi sebagai berikut:

- *Processor* AMD E2-9000e RADEON R2, 4 COMPUTE CORES 2C+2G (2CPUs), ~1.5GHz
- Memory 4096MB RAM
- Operating system Windows 10 Pro 64-bit (10.0, Build 19045)

Penelitian ini menggunakan sistem operasi *Microsoft Windows* 10 dan alat bahasa pemograman *Phyton version 3.8.2* dan *Jupyter Notebook*.



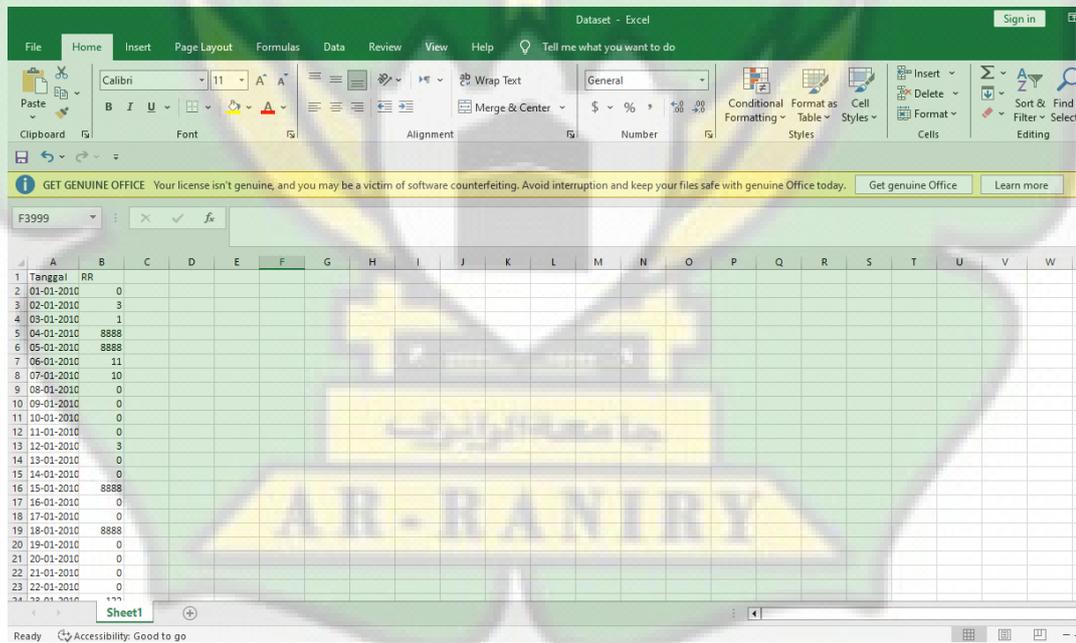
## BAB IV

# HASIL DAN PEMBAHASAN

Pada bab ini, penulis akan menguraikan hasil dari pengumpulan data, langkah-langkah *pre-processing*, serta menampilkan hasil dari percobaan metode yang digunakan. Selain itu, akan dilakukan evaluasi terhadap kinerja model yang dihasilkan.

### 4.1 Pengambilan Dataset

Dataset yang digunakan adalah data Curah Hujan yang di ambil dari website resmi BMKG (<https://dataonline.bmkg.go.id/>). Dengan jumlah sampel dataset sebanyak 4283 data. Seperti yang ditunjukkan pada gambar 4.1 di bawah, dataset tersebut memiliki dua atribut, yaitu tanggal dan curah hujan.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Tanggal	RR																					
2	01-01-2010	0																					
3	02-01-2010	3																					
4	03-01-2010	1																					
5	04-01-2010	8888																					
6	05-01-2010	8888																					
7	06-01-2010	11																					
8	07-01-2010	10																					
9	08-01-2010	0																					
10	09-01-2010	0																					
11	10-01-2010	0																					
12	11-01-2010	0																					
13	12-01-2010	3																					
14	13-01-2010	0																					
15	14-01-2010	0																					
16	15-01-2010	8888																					
17	16-01-2010	0																					
18	17-01-2010	0																					
19	18-01-2010	8888																					
20	19-01-2010	0																					
21	20-01-2010	0																					
22	21-01-2010	0																					
23	22-01-2010	0																					
24	23-01-2010	0																					

Gambar 4. 1 Sampel Dataset Curah Hujan

### 4.2 Membaca Dataset Curah Hujan

Seperti yang ditunjukkan pada Gambar 4.2, langkah ini adalah langkah di mana data excel curah hujan yang digunakan akan dibaca:

```

      Tanggal      RR
0      01-01-2010  0.0
1      02-01-2010  3.0
2      03-01-2010  1.0
3      04-01-2010  8888.0
4      05-01-2010  8888.0
...
4711   27-12-2022  0.0
4712   28-12-2022  0.0
4713   29-12-2022  0.0
4714   30-12-2022  21.5
4715   31-12-2022  0.0

[4716 rows x 2 columns]

```

Gambar 4. 2 Raw Dataset Curah Hujan

Gambar 4.2 diatas merupakan hasil dari pemanggilan dataset menggunakan library *pandas* dan menggunakan fungsi *read\_excel* yang bertujuan untuk membaca dataset yang berbentuk file excel, yang menampilkan dataset curah hujan yang digunakan dalam bentuk Raw data.

### 4.3 *Preprocessing* Data

Tujuan utama dari *Preprocessing* data adalah untuk membersihkan, merapikan, dan menyesuaikan data sehingga data dapat di analisis atau digunakan untuk melatih model *Machine Learning*.

#### 4.3.1 Menghilangkan Data Yang Tidak Bernilai (Nan)

```

Tanggal      0
RR           0
dtype: int64

```

Gambar 4. 3 Hasil Nan

Gambar 4.3 merupakan hasil dari tahap pertama *Preprocessing* yang dilakukan untuk menghilangkan data yang tidak bernilai dan mengganti angka 8888 menjadi Nan, tujuan di ganti agar angka 8888 tersebut tidak mengganggu proses training nantinya yang akan membuat hasil nya tidak bagus, juga ada fungsi untuk mendeklarasikan format tanggal agar tanggal tidak termasuk saat proses training data, dan terakhir ada fungsi interpolasi dan dropna yaitu fungsi agar mengisi nilai yang kosong dan menghilangkan nilai Nan.

Selanjutnya nilai RR (Curah Hujan) akan di bagi menjadi 2 fitur yaitu *input* dan *output*, memisahkan fitur *input* dan *output* menjadi dua kelompok terpisah mempermudah identifikasi dan pemahaman terhadap struktur dataset. fitur *input* dan *output* di butuhkan untuk melatih dan menguji model. Pemisahan fitur ini juga berfungsi saat melakukan validasi dan evaluasi model untuk memastikan bahwa model diterapkan dengan benar pada data uji dan hasil prediksi dapat dibandingkan dengan nilai sebenarnya.

### 4.3.2 Normalisasi Data

```
[[0.      ]
 [0.01588983]
 [0.00529661]
 ...
 [0.      ]
 [0.11387712]
 [0.      ]]
```

Gambar 4. 4 Hasil Normalisasi Dataset

Pada gambar 4.4 diatas menunjukkan hasil dari normalisasi atau *scaling* yang digunakan dalam *preprocessing* data, menggunakan fungsi *MinMaxScaler* untuk mengubah nilai-nilai fitur dalam suatu dataset dalam rentang tertentu, seperti 0 hingga 1.

### 4.3.3 Pembagian Data

Setelah dilakukan normalisasi selanjutnya membagi dataset menjadi dua yaitu data latih dan data uji, ini dilakukan dengan menggunakan fungsi *train\_test\_split* dari *library scikit\_learn*. 20% akan dialokasikan ke data uji dan 80% menjadi data latih.

Tabel 4. 1 Pembagian Dataset

Satuan	Data latih	Data uji
Persen	80%	20%
Jumlah data	3444	839

### 4.3.4 Pembentukan Timesteps

Setelah dilakukan pembagian data selanjutnya membuat *sequence* atau jumlah langkah waktu yang diperhitungkan dalam suatu model deret waktu (*timesteps*). Fungsi dibuat *sequence* dari data *time series* adalah untuk digunakan sebagai inputan dan output pada model LSTM untuk melatih dan membuat prediksi pada data *time series*.

```
[[0.      ]  
 [0.01588983]  
 [0.00529661]  
 ...  
 [0.      ]  
 [0.      ]  
 [0.      ] [[0.      ]  
 [0.01588983]  
 [0.00529661]  
 ...  
 [0.      ]  
 [0.      ]  
 [0.      ]] 30
```

Gambar 4. 5 Hasil Pembentukan *timesteps*

## 4.4 Implementasi Model

Data akan dibagi menjadi dua bagian utama pada tahapan ini: data latih dan data uji. Data pelatihan akan digunakan untuk melatih model, sedangkan data uji akan digunakan untuk menguji model setelah pelatihan selesai.

Setelah membuat model LSTM, model akan dilatih menggunakan epoch sebagai data pelatihan (*x\_train\_seq* dan *y\_train\_seq*) dan dipantau kinerja pada data pengujian (*x\_test\_seq* dan *y\_test\_seq*), yang hasilnya disimpan dalam objek hist. Gambar 4.6 di bawah ini menunjukkan hasil dari pengujian epoch, proses ini menggambarkan proses pengujian epoch dari dataset yang sudah dinormalisasikan sebelumnya.

```

Epoch 1/75
108/108 [=====] - 30s 116ms/step - loss: 0.0067 - accuracy: 0.6106 - val_loss: 0.0115
Epoch 2/75
108/108 [=====] - 11s 102ms/step - loss: 0.0066 - accuracy: 0.6106 - val_loss: 0.0116
Epoch 3/75
108/108 [=====] - 9s 84ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0114 -
Epoch 4/75
108/108 [=====] - 8s 72ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0114 -
Epoch 5/75
108/108 [=====] - 9s 80ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0113 -
Epoch 6/75
108/108 [=====] - 9s 80ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0114 -
Epoch 7/75
108/108 [=====] - 10s 97ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0114 -
Epoch 8/75
108/108 [=====] - 9s 82ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0113 -
Epoch 9/75
108/108 [=====] - 10s 91ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0113 -
Epoch 10/75
108/108 [=====] - 9s 84ms/step - loss: 0.0065 - accuracy: 0.6106 - val_loss: 0.0115 -

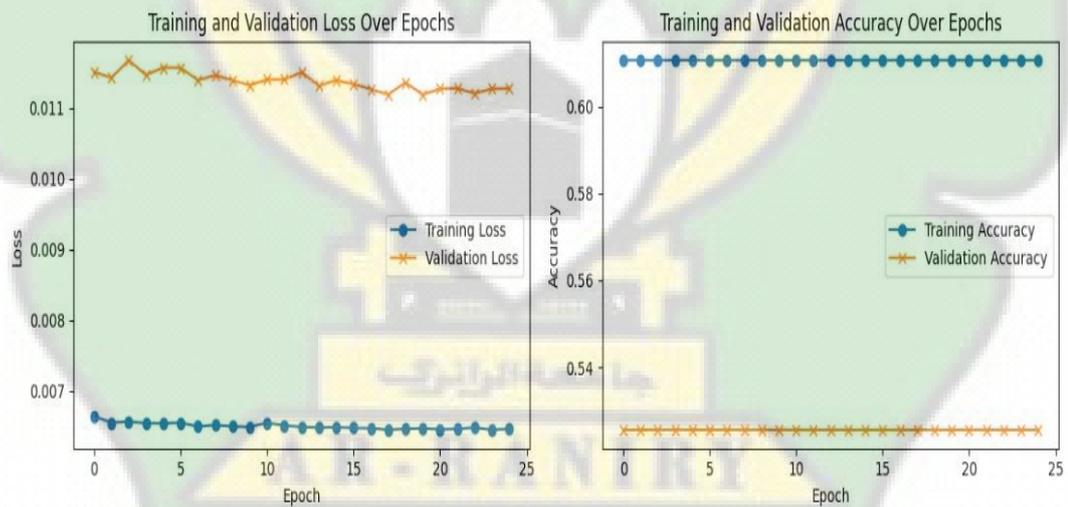
```

Gambar 4. 6 Hasil Uji Epoch

#### 4.4.1 Plot Akurasi Training Setiap Epoch

Dari hasil training diatas didapatkan grafik dari setiap epoch yang digunakan sebagai berikut:

- Epoch 25



Gambar 4. 7 Loss Epoch 25

- Epoch 50



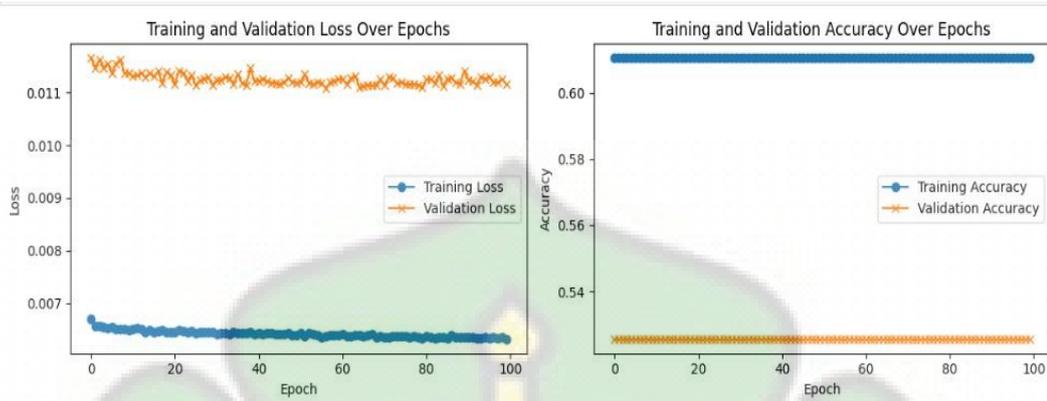
Gambar 4. 8 Loss Epoch 50

- Epoch 75



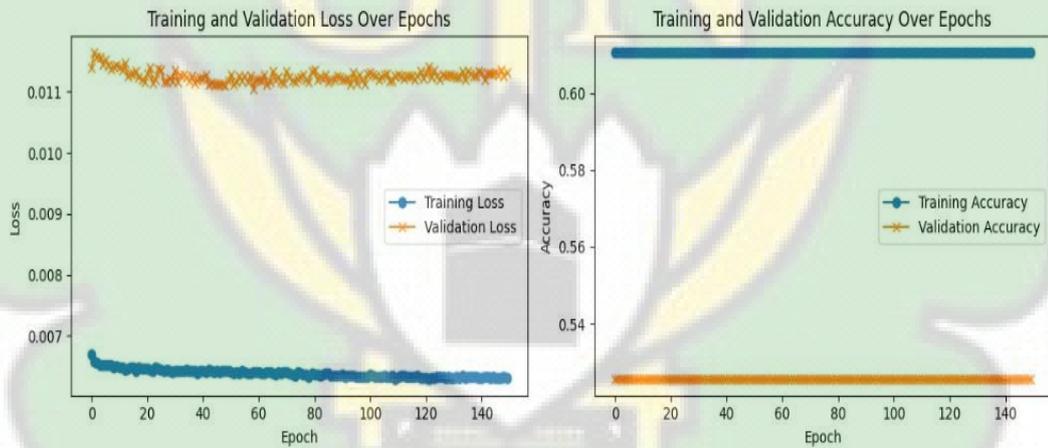
Gambar 4. 9 Loss Epoch 75

- Epoch 100



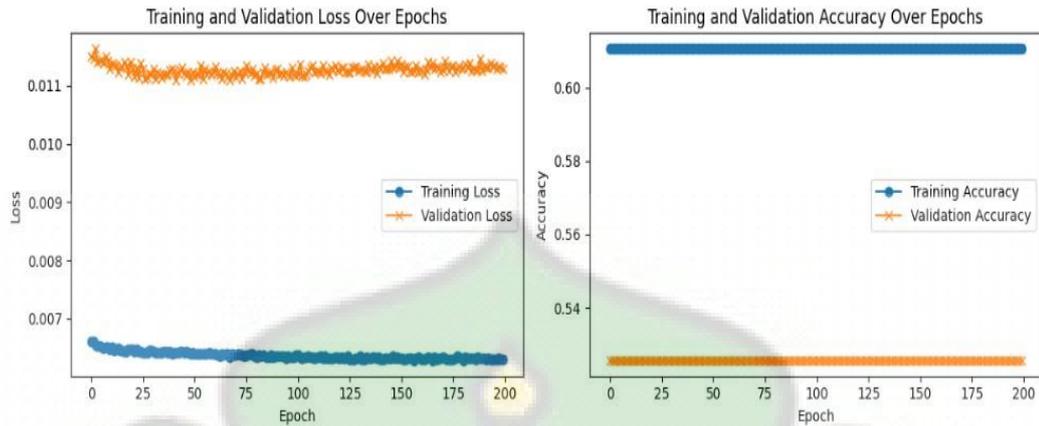
Gambar 4. 10 Loss Epoch 100

- Epoch 150



Gambar 4. 11 Loss Epoch 150

- Epoch 200



Gambar 4. 12 Loss Epoch 200

#### 4.5 Membuat Prediksi dan Pembalikan Prediksi

Langkah selanjutnya membuat prediksi data train dan data test, hasil dari prediksi dapat dilihat pada gambar dibawah ini :

```
108/108 [=====] - 2s 7ms/step
27/27 [=====] - 0s 7ms/step
```

Gambar 4. 13 Hasil Prediksi Data latih dan Data Uji

Untuk memastikan bahwa model yang dibuat sama dengan data asli, langkah selanjutnya adalah melakukan pembalikan prediksi.

#### 4.6 Evaluasi Hasil

Setelah itu, evaluasi dilakukan untuk mengetahui seberapa efektif model yang telah dibuat. Evaluasi RMSE menguji dua model, data pelatihan dan data uji.

```
Train Score: 15.00 RMSE
Test Score: 20.27 RMSE
```

Gambar 4. 14 Hasil perhitungan RMSE

Hasil dari nilai RMSE mengukur seberapa besar deviasi atau kesalahan rata-rata antara nilai prediksi dan nilai aktual, dan hasilnya diukur dalam unit yang sama dengan variabel target. Skor data tes yang lebih rendah menunjukkan bahwa prediksi yang dibuat lebih akurat, dan skor data tes yang lebih tinggi menunjukkan bahwa hasilnya kurang baik.

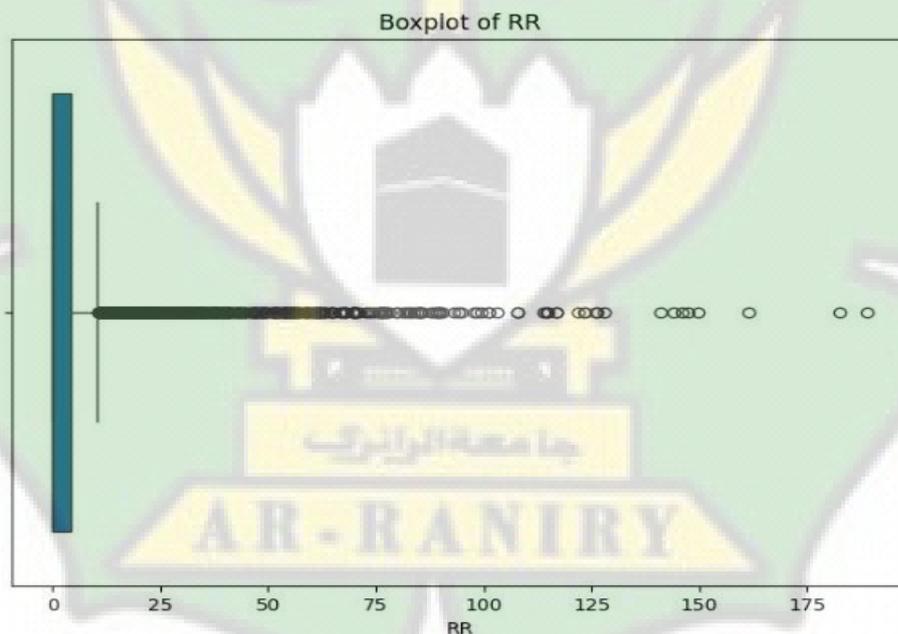
MAE adalah metode evaluasi kedua. Ini juga menguji dua model, data latih dan data uji.

Train Score: 4.72 MAE  
Test Score: 6.38 MAE

Gambar 4. 15 Hasil Perhitungan MAE

## 4.7 Visualisasi Hasil

### 4.7.1 Boxplot Curah Hujan (RR)

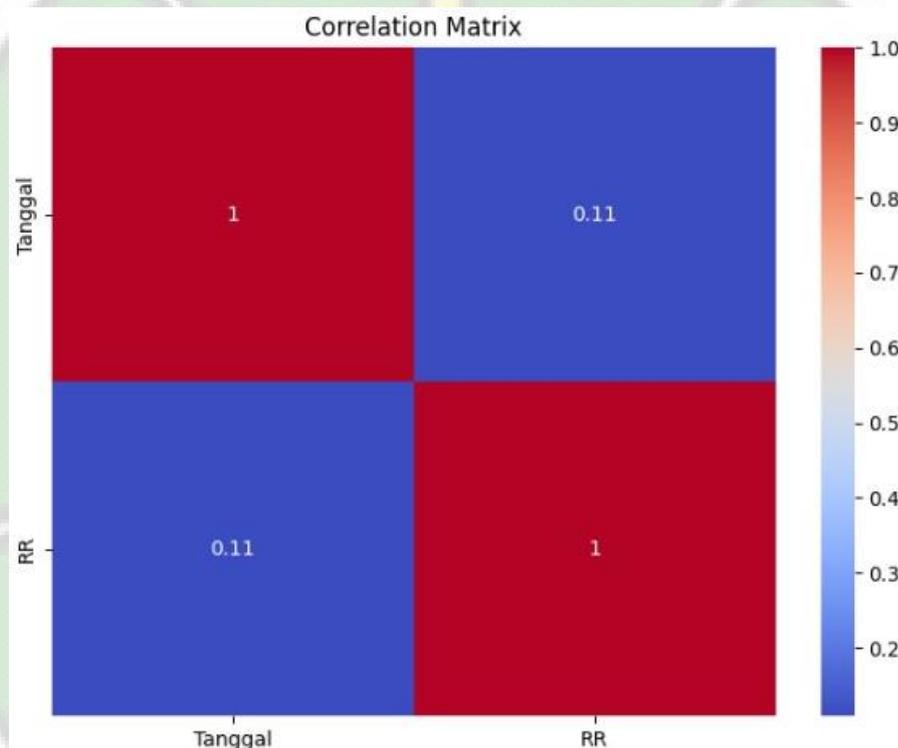


Gambar 4. 16 *Blo*xplot RR

Gambar di atas menunjukkan diagram kotak atau boxplot untuk variabel "RR", yang merupakan hasil dari analisis data statistik. Dalam boxplot ini, lima nilai utama (minimum, kuartil pertama, median, kuartil ketiga, dan maksimum) ditunjukkan untuk menunjukkan distribusi data "RR".

Pada diagram, garis horizontal menunjukkan median, dan kotak menunjukkan rentang interkuartil (IQR), yaitu rentang antara Q1 dan Q3. Garis vertikal, atau "whiskers", yang memanjang di dalam kotak menunjukkan luasnya data yang tidak mencakup outlier. Titik-titik di luar whiskers dianggap sebagai outlier, yang berarti nilainya lebih tinggi atau lebih rendah dari rata-rata data lainnya. Sebagian besar nilai "RR" berada di sekitar nilai yang sangat rendah, tetapi banyak nilai yang sangat tinggi menunjukkan bahwa ada data pencilan.

#### 4.7.2 Korelasi Matrix



Gambar 4. 17 Korelasi Matrix

#### 4.8 Hasil Uji Performa

Tabel hasil uji performa di bawah ini menunjukkan nilai RMSE dan MAE pengujian performa berdasarkan *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE). Epoch menjadi acuan utama untuk menghasilkan nilai RMSE dan MAE terendah untuk data uji performa terbaik.

Tabel 4. 2 Uji performa

Epoch	Data Train	Data Test	RMSE Data Train	RMSE Data Test	MAE Data Train	MAE Data Test
25	3444	839	15.00	20.04	6.29	8.08
75	3444	839	14.88	20.18	5.32	7.78
50	3444	839	15.00	20.27	4.72	6.38
100	3444	839	14.69	20.11	6.33	8.50
150	3444	839	13.33	22.01	6.65	9.46
200	3444	839	12.45	21.84	6.65	8.78

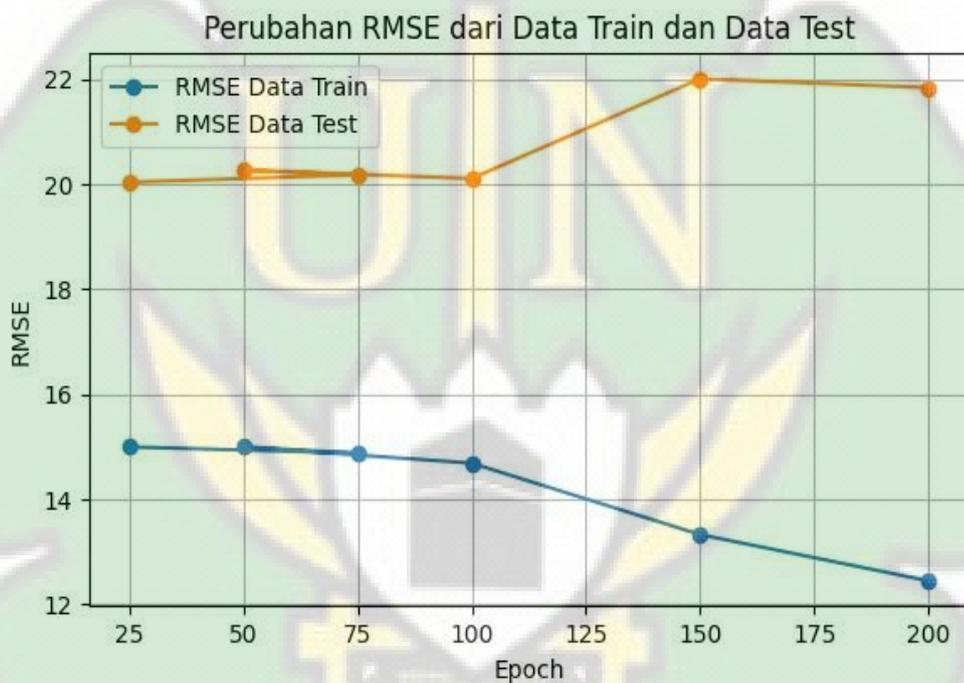
Tabel diatas mengidentifikasi nilai RMSE (*Root Mean Squared Error*) dan MAE (*Mean Absolute Error*) tertinggi dan terendah untuk data pelatihan dan pengujian. RMSE (*Root Mean Squared Error*) tertinggi untuk data pelatihan adalah 15.00 pada Epoch 25 dan 50, sementara yang terendah adalah 12.45 pada Epoch 200. Untuk data pengujian, RMSE tertinggi adalah 22.01 pada Epoch 150, dan yang terendah adalah 20.04 pada Epoch 25. Sedangkan untuk MAE (*Mean Absolute Error*), nilai tertinggi untuk data pelatihan adalah 6.65 pada Epoch 150 dan 200, dan yang terendah adalah 4.72 pada Epoch 50.

Untuk data pengujian, MAE tertinggi adalah 9.46 pada Epoch 150, dan yang terendah adalah 6.38 pada Epoch 50. Nilai-nilai ini menunjukkan performa model pada berbagai epoch, dengan nilai RMSE dan MAE yang lebih rendah menunjukkan performa model yang lebih baik.

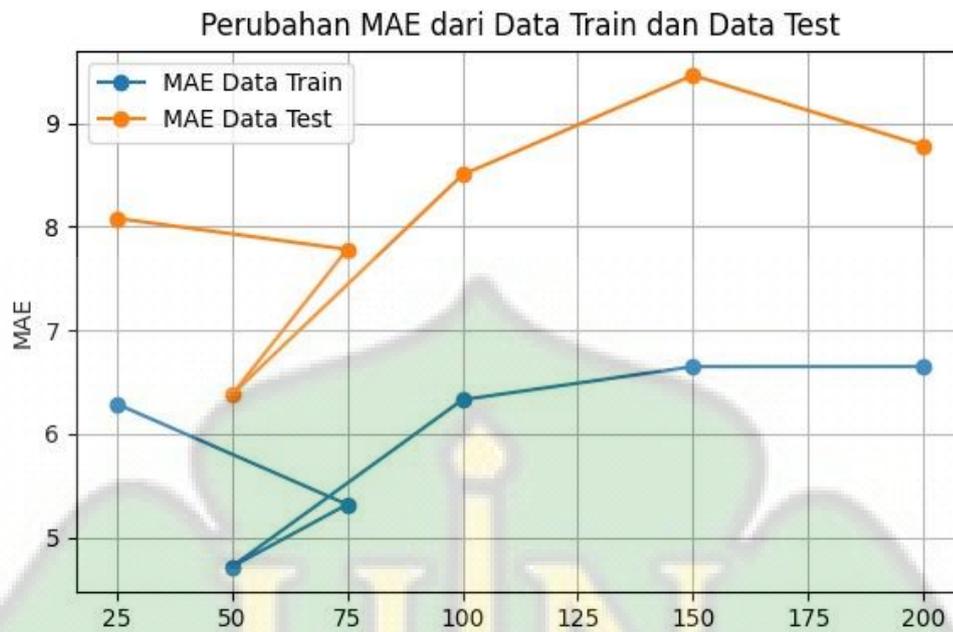
#### 4.8.1 Pengaruh Jumlah Epoch terhadap Performa Model

- Secara keseluruhan, terlihat bahwa peningkatan jumlah Epoch cenderung menghasilkan penurunan nilai RMSE dan MAE pada Data Test hingga suatu titik tertentu.

- Pada awalnya, peningkatan jumlah Epoch dari 25 ke 50 menunjukkan penurunan yang signifikan dalam nilai RMSE dan MAE pada Data Test, menunjukkan peningkatan performa model.
- Namun, setelah mencapai Epoch 100, terlihat variasi yang lebih besar dalam nilai RMSE dan MAE pada Data Test, bahkan ada kenaikan yang kecil pada Epoch 150 dan 200. Hal ini menunjukkan bahwa peningkatan jumlah Epoch terlalu tinggi menyebabkan overfitting pada model.



Gambar 4. 18 Perbandingan RMSE Data Train dan Test



Gambar 4. 19 Perbandingan MAE Data Train dan Test

Menurut hasil uji performa yang ditunjukkan pada tabel di atas, nilai 150 memiliki performa terendah, dengan nilai RMSE Data Train 13.33, RMSE Data Test 22.01, MAE Data Train 6.65, dan MAE Data Test 9.46. Nilai epoch 75 memiliki performa tertinggi, dengan nilai RMSE Data Train 14.88, RMSE Data Test 20.18, MAE Data Train 5.32, dan MAE Data Test 7.78.

#### 4.8.2 Perbandingan dengan penelitian terdahulu

pada penelitian Yos Hendra Tingkat error pada data test sangat tinggi dibandingkan dengan data train, menunjukkan adanya overfitting. Mungkin terjadi kesenjangan besar antara data train dan test yang menyebabkan hasil yang tidak baik pada data test. Sedangkan pada penelitian ini menghasilkan Performa model pada kedua data train dan data test cukup baik pada nilai epoch 75 dengan RMSE dan MAE yang rendah. Ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Pada penelitian Ardytha Luthfiarta, variabel suhu udara memengaruhi koefisien regresi secara signifikan, kinerja model regresi bergantung pada penggunaan variabel independen dan pengaturan hiperparameter seperti jumlah epoch. Sedangkan pada penelitian ini model dengan jumlah epoch 75 memberikan prediksi data yang lebih baik, meskipun koefisien regresi untuk variabel suhu udara adalah 0,147.

Penelitian Jamilatul Badriyah dengan hasil penelitian ini terlihat bahwa hasil algoritma LSTM memiliki performa yang lebih baik dibandingkan dengan hasil dari model regresi linier yang telah disebutkan.

- Nilai MSE dan MAE yang lebih rendah pada algoritma LSTM menunjukkan bahwa model LSTM memiliki tingkat kesalahan yang lebih kecil dalam memprediksi data.
- Nilai R2 yang lebih tinggi pada algoritma LSTM menunjukkan bahwa model LSTM lebih baik dalam menjelaskan variabilitas data dibandingkan dengan model regresi linier.

Penelitian dari Aulia Herdhyanti menggambarkan evaluasi performa model dalam memprediksi curah hujan. Pada penelitiannya, perbandingan dilakukan berdasarkan nilai Mean Squared Error (MSE) antara model terbaik dan terburuk. Hasilnya menunjukkan bahwa model dengan MSE lebih rendah memiliki tingkat kesalahan prediksi yang lebih kecil. Sementara itu, pada penelitian ini, perbandingan dilakukan berdasarkan Root Mean Squared Error (RMSE) dan Mean Absolute Error (MAE) dari model dengan performa terendah dan tertinggi. Model dengan RMSE dan MAE lebih kecil menunjukkan prediksi yang lebih akurat. Dengan demikian, kedua penelitian menunjukkan bahwa prediksi curah hujan yang lebih akurat dibuat oleh model dengan tingkat kesalahan dan akurasi yang lebih rendah.

# BAB V

## PENUTUP

### 5.1 Kesimpulan

Dari hasil penelitian yang telah penulis lakukan, Penulis mencapai beberapa kesimpulan dari penelitian ini, antara lain :

1. Proses analisis curah hujan menggunakan metode LSTM melibatkan beberapa tahap, mulai dari analisis korelasi dan distribusi data hingga pelatihan model LSTM untuk menangkap pola temporal yang kompleks. Analisis awal menunjukkan bahwa hubungan antara tanggal dan curah hujan sangat lemah, yang menegaskan pentingnya penggunaan metode yang lebih canggih seperti LSTM untuk prediksi yang lebih akurat.
2. Pada data pelatihan (*train*), epoch 200 menunjukkan RMSE terendah (12.45) dan MAE tertinggi kedua (6.65), yang menunjukkan performa model yang lebih baik dengan nilai RMSE terendah. Dan pada data pengujian (*test*), epoch 50 menunjukkan MAE terendah (6.38) dan salah satu RMSE tertinggi kedua (20.27), yang menunjukkan performa yang lebih baik dalam hal MAE terendah.

kesimpulan yang dapat ditarik adalah bahwa pada Epoch 200, model menunjukkan performa terbaik untuk data pelatihan dengan RMSE terendah, sementara pada Epoch 50, model menunjukkan performa terbaik untuk data pengujian dengan MAE terendah.

## 5.2 Saran

Peneliti berharap peneliti lain dapat mengembangkan penelitian ini lagi. Saran yang dapat dibuat oleh peneliti adalah:

1. Peneliti dapat meningkatkan hasil penelitian dengan menambahkan parameter dan teknik tambahan, dan membandingkan teknik mana yang lebih baik untuk diterapkan pada penelitian berikutnya.
2. Dapat menambahkan data yang digunakan untuk mendapatkan nilai prediksi yang baik dan nilai akurasi yang akurat.



## DAFTAR PUSTAKA

- Anwar, S. (2017). Peramalan Suhu Udara Jangka Pendek di Kota Banda Aceh dengan Metode Autoregressive Integrated Moving Average (ARIMA). *Malikussaleh Journal of Mechanical Science and Technology*, 5(1), 6–12. [https://www.researchgate.net/publication/326224696\\_Peramalan\\_Suhu\\_Udara\\_Jangka\\_Pendek\\_di\\_Kota\\_Banda\\_Aceh\\_dengan\\_Metode\\_Autoregressive\\_Integrated\\_Moving\\_Average\\_ARIMA?enrichId=rgreq1936f7171069e1ceec615b794def598eXXX&enrichSource=Y292ZXJQYWdlOzMjYjYjNDY5](https://www.researchgate.net/publication/326224696_Peramalan_Suhu_Udara_Jangka_Pendek_di_Kota_Banda_Aceh_dengan_Metode_Autoregressive_Integrated_Moving_Average_ARIMA?enrichId=rgreq1936f7171069e1ceec615b794def598eXXX&enrichSource=Y292ZXJQYWdlOzMjYjYjNDY5)
- Badriyah, J., Fariza, A., & Harsono, T. (2022). Prediksi Curah Hujan Menggunakan Long Short Term Memory. *Jurnal Media Informatika Budidarma*, 6(3), 1297. <https://doi.org/10.30865/mib.v6i3.4008>
- Bambang Triatmodjo, 2008. “*Hidrologi Terapan*”. Yogyakarta : Beta Offset
- Barus, J., & Ramli, R. (2013). Analisis Peramalan Ekspor Indonesia Pasca Krisis Keuangan Eropa Dan Global Tahun 2008 Dengan Metode Dekomposisi. *Jurnal Ekonomi Dan Keuangan*, 1(3), 14880.
- BMKG. (2008). Curah Hujan dan Potensi Bencana Gerakan Tanah. 1–7.
- B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman, “Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study,” *Proc. ACM/IEEE Jt. Conf. Digit. Libr.*, 2017, doi:10.1109/JCDL.2017.7991618.
- Colah. (2015). Retrieved from *Colahsblog*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Desmonda, D., Tursina, T., & Irwansyah, M. A. (2018). Prediksi Besaran Curah Hujan Menggunakan Metode Fuzzy Time Series. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 6(4), 141. <https://doi.org/10.26418/justin.v6i4.27036>
- Hendra, Y., Mukhtar, H., & Hafsari, R. (2023). *Jurnal Software Engineering and Information System ( SEIS ) PREDIKSI CURAH HUJAN DI KOTA PEKANBARU MENGGUNAKAN LSTM*. 3(2).

- Herdhyanti, A., Muflikhah, L., & Cholissodin, I. (2022). Prediksi Curah Hujan dengan Empat Parameter menggunakan Backpropagation (Studi Kasus: Stasiun Meteorologi Ahmad Yani). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(12), 2548–2964. <http://j-ptiik.ub.ac.id>
- Hanke, John E & Wichern, Dean W. 2009, *Business Forecasting* 9th ed. New Jersey. Herjanto, E. (2009). *Sains Manajemen*. Jakarta: Grasindo.
- Luthfiarta, A., Febriyanto, A., Lestiawan, H., & Wicaksono, W. (2020). Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda. *JOINS (Journal of Information System)*, 5(1), 10–17. <https://doi.org/10.33633/joins.v5i1.2760>
- Merdekawati, G. I., & Ismail. (2019). Prediksi Curah Hujan Di Jakarta Berbasis Algoritma Levenberg Marquardt. *Jurnal Ilmiah Informatika Komputer*, 24(2), 116–128. <https://doi.org/10.35760/ik.2019.v24i2.2366>
- Prasetya, B. D., Pamungkas, F. S., & Kharisudin, I. (2020). Pemodelan dan Peramalan Data Saham dengan Analisis Time Series menggunakan Python. *PRISMA, Prosiding Seminar Nasional Matematika*, 3, 714–718. <https://journal.unnes.ac.id/sju/index.php/prisma/article/view/38116>
- Putramulyo, S., & Alaa, S. (2018). Prediksi Curah Hujan Bulanan Di Kota Samarinda Menggunakan Persamaan Regresi Dengan Prediktor Data Suhu dan Kelembapan Udara. *Eigen Mathematics Journal*, 13–16. <https://doi.org/10.29303/emj.v2i2.20>
- Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE*, 15(1), 1–9. <https://doi.org/10.1371/journal.pone.0227222>
- Ritha, N., Bettiza, M., & Dufan, A. (2016). 266810-Prediksi-Curah-Hujan-Dengan-Menggunakan-D6a27956. 5(2).
- 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>

Studi, P., Informatika, T., Informatika, J. T., Komputer, F. I., & Brawijaya, U. (2018). *ADAPTIVE-EXPECTATION BASED MULTI-ATTRIBUTE FUZZY TIME SERIES*.

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.

Yunanto, R., Purfini, A. P., & Prabuwisesa, A. (2021). Survei Literatur: Deteksi Berita Palsu Menggunakan Pendekatan Deep Learning. *Jurnal Manajemen Informatika (JAMIKA)*, 11(2), 118–130. <https://doi.org/10.34010/jamika.v11i2.5362>

Zahara, S., Sugianto, & M. Bahril Ilmiddafiq. (2019). Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 3(3), 357–363. <https://doi.org/10.29207/resti.v3i3.1086>

## LAMPIRAN

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# Membaca dataset
df = pd.read_excel("Dataset.xlsx")
df

# Preprocessing dataset
df['RR'] = df['RR'].replace(8888, pd.NA)
df['Tanggal'] = pd.to_datetime(df['Tanggal'], format='%d-
%m-%Y')
numeric_columns =
df.select_dtypes(include=[np.number]).columns
df[numeric_columns] = df[numeric_columns].interpolate()
df = df.dropna()
df.isnull().sum()

# Menyiapkan data
input_features = ['RR']
output_features = ['RR']

x = df[input_features].values
y = df[output_features].values
```

```

scaler = MinMaxScaler(feature_range=(0, 1))
x_scaled = scaler.fit_transform(x)
y_scaled = scaler.fit_transform(y)

x_train, x_test, y_train, y_test =
train_test_split(x_scaled, y_scaled, test_size=0.2,
shuffle=False)

# Membuat sequences
timesteps = 30
n_features_input = len(input_features)
n_features_output = len(output_features)

def create_sequences(x, y, time_steps):
    x_seq, y_seq = [], []
    for i in range(len(x) - time_steps):
        x_seq.append(x[i: i + time_steps])
        y_seq.append(y[i + time_steps])
    return np.array(x_seq), np.array(y_seq)

x_train_seq, y_train_seq = create_sequences(x_train,
y_train, timesteps)
x_test_seq, y_test_seq = create_sequences(x_test, y_test,
timesteps)

print(x_train, y_train, timesteps)

# Build and train the model
model = Sequential()
model.add(LSTM(60, input_shape=(timesteps,
n_features_input), return_sequences=True))
model.add(Dropout(0.2)) # Tambahkan dropout 20% untuk
mencegah overfitting
model.add(LSTM(60))
model.add(Dropout(0.2)) # Tambahkan dropout 20% untuk
mencegah overfitting
model.add(Dense(n_features_output))
model.compile(loss='mean_squared_error',
optimizer=Adam(learning_rate=0.0001), metrics=['accuracy'])

```

```

hist = model.fit(x_train_seq, y_train_seq, epochs=75,
batch_size=32, validation_data=(x_test_seq, y_test_seq))

# Membuat Scatter Plot pada Hasil Training dan Validation
# Training
train_loss = hist.history['loss']
train_acc = hist.history['accuracy'] if 'accuracy' in
hist.history else None

# Validation
val_loss = hist.history['val_loss']
val_acc = hist.history['val_accuracy'] if 'val_accuracy' in
hist.history else None

# Plot Loss
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(train_loss, label='Training Loss', marker='o')
plt.plot(val_loss, label='Validation Loss', marker='x')
plt.title('Training and Validation Loss Over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

# Plot Accuracy (if available)
if train_acc and val_acc:
    plt.subplot(1, 2, 2)
    plt.plot(train_acc, label='Training Accuracy',
marker='o')
    plt.plot(val_acc, label='Validation Accuracy',
marker='x')

    plt.title('Training and Validation Accuracy Over
Epochs')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend()

```

```

plt.tight_layout()
plt.show()

# Make predictions
trainPredict = model.predict(x_train_seq, batch_size=32)
model.reset_states()
testPredict = model.predict(x_test_seq, batch_size=32)

# Invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
testPredict = scaler.inverse_transform(testPredict)

# Memastikan panjang yang konsisten
y_train_actual = scaler.inverse_transform(y_train_seq.reshape(-1, 1))[:3444]
y_test_actual = scaler.inverse_transform(y_test_seq.reshape(-1, 1))[:839]

# Membuat DataFrame untuk hasil prediksi
df_train_pred = pd.DataFrame({'Data Asli': y_train_actual.flatten(), 'Prediksi': trainPredict.flatten()})
df_test_pred = pd.DataFrame({'Data Asli': y_test_actual.flatten(), 'Prediksi': testPredict.flatten()})

# Menyimpan hasil prediksi ke dalam file Excel
df_train_pred.to_excel('train_predictions.xlsx', index=False)
df_test_pred.to_excel('test_predictions.xlsx', index=False)

# Menampilkan lokasi penyimpanan hasil prediksi
print('Hasil prediksi training telah disimpan di: train_predictions.xlsx')
print('Hasil prediksi testing telah disimpan di: test_predictions.xlsx')

# Invert predictions
trainPredict = scaler.inverse_transform(trainPredict)

```

```

y_train_actual = scaler.inverse_transform(y_train_seq.reshape(-1, 1))
testPredict = scaler.inverse_transform(testPredict)
y_test_actual = scaler.inverse_transform(y_test_seq.reshape(-1, 1))

# Memastikan panjang yang konsisten
y_train = y_train[:3444]
trainPredict = trainPredict[:3444]
y_test = y_test[:839]
testPredict = testPredict[:839]
print('trainPredict shape:', trainPredict.shape)
print('testPredict shape:', testPredict.shape)

# Calculate root mean squared error
trainScore = np.sqrt(mean_squared_error(y_train_actual,
trainPredict.flatten()))
print('Train Score: %.2f RMSE' % trainScore)

testScore = np.sqrt(mean_squared_error(y_test_actual,
testPredict.flatten()))
print('Test Score: %.2f RMSE' % testScore)

# Menghitung MAE
mae_train = np.mean(np.abs(y_train - trainPredict))
mae_test = np.mean(np.abs(y_test - testPredict))

# Menampilkan hasil
print('Train Score: %.2f MAE' % (mae_train))
print('Test Score: %.2f MAE' % (mae_test))

# Plot hasil prediksi untuk data training
plt.figure(figsize=(15, 6))
plt.plot(y_train_actual, label='Actual (Training)',
color='blue')
plt.plot(trainPredict, label='Predicted (Training)',
color='orange')
plt.title('Actual vs Predicted (Training)')

```

```

plt.xlabel('Data Index')
plt.ylabel('Value')
plt.legend()
plt.show()

# Plot hasil prediksi untuk data training
plt.figure(figsize=(15, 6))
plt.plot(y_train_actual, label='Actual (Training)',
color='blue')
plt.plot(trainPredict, label='Predicted (Training)',
color='orange')
plt.title('Actual vs Predicted (Training)')
plt.xlabel('Data Index')
plt.ylabel('Value')
plt.legend()
plt.show()

# Plot hasil prediksi untuk data training dan data testing
plt.figure(figsize=(15, 6))

# Plot data asli (biru)
plt.plot(y_train_actual, label='Actual (Training)',
color='blue')
plt.plot(np.arange(len(y_train_actual), len(y_train_actual)
+ len(y_test_actual)), y_test_actual, label='Actual
(Testing)', color='blue')

# Plot prediksi untuk data pelatihan (orange)
plt.plot(trainPredict, label='Predicted (Training)',
color='orange')

# Plot prediksi untuk data uji (hijau)
plt.plot(np.arange(len(y_train_actual), len(y_train_actual)
+ len(y_test_actual)), testPredict, label='Predicted
(Testing)', color='green')

plt.title('Actual vs Predicted')
plt.xlabel('Data')
plt.ylabel('RR')
plt.legend()
plt.show()

```