

**PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM
BERDASARKAN PARAMETER CUACA
DI KABUPATEN MALANG**

TESIS

**Oleh:
AHMAD ROMADHANI
NIM. 230605220003**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM
BERDASARKAN PARAMETER CUACA
DI KABUPATEN MALANG**

TESIS

**Diajukan Kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:
AHMAD ROMADHANI
NIM. 230605220003**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM
BERDASARKAN PARAMETER CUACA
DI KABUPATEN MALANG**

TESIS

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:
AHMAD ROMADHANI
NIM. 230605220003**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM
BERDASARKAN PARAMETER CUACA DI KABUPATEN MALANG**

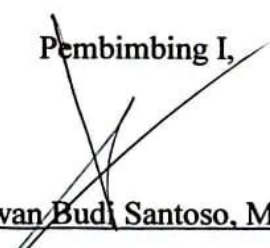
TESIS

Oleh:

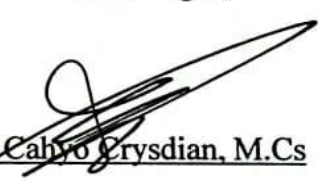
**AHMAD ROMADHANI
NIM. 23230605220003**

Telah diperiksa dan disetujui untuk diuji:
Tanggal: 11 November 2025

Pembimbing I,


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Pembimbing II,


Dr. Cahyo Crysodian, M.Cs
NIP. 19740424 200901 1 008


Mengetahui dan Mengesahkan

Ketua Program Studi Magister Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang




Prof. Dr. Ir. Muhammad Faisal, M.T

NIP. 19740510 200501 1 007

**PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM
BERDASARKAN PARAMETER CUACA DI KABUPATEN MALANG**

TESIS

Oleh:





**AHMAD ROMADHANI
NIM. 23230605220003**

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 11 November 2025

Susunan Dewan Penguji

Penguji I	: <u>Dr. M. Imamudin, Lc., MA</u> NIP. 19740602 201101 1 001
Penguji II	: <u>Dr. Ir. Fresy Nugroho, S.T., M.T., I.P.M., ASEAN Eng</u> NIP. 19710722 201101 1 001
Pembimbing I	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004
Pembimbing II	: <u>Dr. Cahyo Crysdiyan, M.Cs</u> NIP. 19740424 200901 1 008

Tanda Tangan

()
()
()
()

Mengetahui dan Mengesahkan

Ketua Program Studi Magister Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Prof. Dr. Ir. Muhammad Faisal, M.T

NIP. 19740510 200501 1 007

HALAMAN PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Romadhani
NIM : 230605220003
Program Studi : Magister Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-banar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Oktober 2025
Yang membuat pernyataan



Ahmad Romadhani
NIM. 230605220003

MOTTO

“Kesuksesan bukan hasil akhir, melainkan perjalanan panjang untuk terus belajar,
mencoba, dan tidak pernah menyerah.”

PERSEMBAHAN

Dengan mengucapkan syukur *Alhamdulillah rabbil alamin*, Thesis ini saya persembahkan untuk:

1. Seluruh keluarga tercinta (Istri, Kedua Orang Tua, serta Kedua Mertua) yang selalu memberikan dukungan, doa dan semangat yang tiada henti.
2. Seluruh Civitas Akademika Universitas Islam Negeri Maulana Malik Ibrahim Malang untuk dapat menimba ilmu, teknologi serta agama.
3. Seluruh rekan-rekan Asosiasi Mahasiswa Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang semua angkatan atas kerjasamanya selama ini.
4. Teman-teman santri Al-Hamidiyah Dinoyo, Malang yang memberikan semangat kepada penulis untuk menyelesaikan Thesis ini.
5. Bapak, ibu, saudara dan rekan-rekan sekalian yang tidak bisa saya sebutkan satu persatu yang telah mendukung penyelesaian Thesis ini.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur *alhamdulillah* penulis hanturkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Thesis ini dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan *jazakumullah ahsanal jaza'* kepada semua pihak yang telah membantu terselesaikannya Thesis ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Bapak Dr. Irwan Budi Santoso, M.Kom. dan Bapak Dr. Cahyo Crysdiyan, M.Cs. selaku dosen pembimbing Thesis, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
2. Segenap sivitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/ Ibu dosen, terima kasih atas segenap ilmu dan bimbingannya.
3. Ayahanda dan Ibunda tercinta yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu.
4. Kakak pertama dan kakak kedua penulis yang selalu memberikan semangat kepada penulis untuk menyelesaikan Thesis ini.
5. Semua pihak yang ikut membantu dalam menyelesaikan Thesis ini.

Penulis menyadari bahwa dalam penyusunan Thesis ini masih terdapat kekurangan dan penulis berharap semoga Thesis ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

Malang, 27 Oktober 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xvi
ABSTRAK	xvii
ABSTRACT	xviii
مستخلص البحث	xix
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	8
1.3 Tujuan Penelitian	8
1.4 Batasan Masalah	9
1.5 Manfaat Penelitian	9
BAB II	10
STUDI PUSTAKA	10
2.1 Prediksi Curah Hujan	10
2.2 Perkembangan Jaringan Syaraf Tiruan menuju <i>Attention-Based LSTM</i>	19
2.2.1 <i>Artificial Neural Network</i> (ANN)	19
2.2.2 <i>Recurrent Neural Network</i> (RNN)	22
2.2.3 <i>Long Short-Term Memory</i> (LSTM)	23
2.2.4 <i>Attention-Based LSTM</i> (AT-LSTM)	28
2.3 <i>Evaluation Metric</i>	32
2.4 Kerangka Teori	33

BAB III	36
METODOLOGI PENELITIAN	36
3.1 Kerangka Konsep	36
3.2 Desain Prosedur Penelitian	38
3.2.1 Pengumpulan Data	39
3.2.2 Desain Sistem.....	40
3.2.3 Skenario Uji Coba	44
3.2.4 Evaluasi	45
BAB IV	46
PREDIKSI CURAH HUJAN DENGAN LSTM	46
4.1 Deskripsi Data Curah Hujan Harian BMKG	46
4.2 Arsitektur Model LSTM	47
4.3 <i>Pre-processing Data</i>	48
4.3.1 Menangani <i>Outlier</i>	49
4.3.2 <i>Selection</i> Fitur	50
4.3.3 Normalisasi Data.....	55
4.3.4 Pembagian Data	56
4.4 <i>Hyperparameter</i> LSTM	57
4.5 <i>Training</i> Model LSTM.....	59
4.6 Pengujian Model LSTM.....	67
BAB V	68
PREDIKSI CURAH HUJAN DENGAN ATTENTION-BASED LSTM	68
5.1 Arsitektur Model Attention-Based LSTM	68
5.2 <i>Hyperparameter</i> Attention-Based LSTM	69
5.3 <i>Training</i> Model Attention-Based LSTM	71
5.4 Pengujian Model Attention-Based LSTM	79
BAB VI	80
TRANSFORMASI DATA PADA LSTM & ATTENTION-BASED LSTM .	80
6.1 Hasil Prediksi Tanpa Transformasi Data	80
6.2 Hasil Prediksi dengan Transformasi Data.....	88
6.3 Perbandingan Tanpa Transformasi Data vs dengan Transformasi Data	96
6.4 Pembahasan Hasil Model LSTM	97
6.5 Pembahasan Hasil Model Attention-Based LSTM.....	98
6.6 Perbandingan LSTM vs Attention-Based LSTM.....	99

6.7 Prediksi Curah Hujan dalam Pandangan Islam.....	101
BAB VII	110
KESIMPULAN.....	110
7.1 Kesimpulan	110
7.2 Saran	110
DAFTAR PUSTAKA	112
LAMPIRAN.....	115

DAFTAR GAMBAR

Gambar 2.1: Arsitektur ANN.....	20
Gambar 2.2: Arsitektur RNN.....	22
Gambar 2.3: Arsitektur LSTM.....	24
Gambar 2.4: Arsitektur Attention-Based LSTM.....	28
Gambar 2.5: Kerangka Teori.....	34
Gambar 3.1: Kerangka Konsep.....	36
Gambar 3.2: Desain Prosedur Penelitian	38
Gambar 3.3: Desain Sistem.....	40
Gambar 3.4: Algoritma LSTM dan Attention-Based LSTM.....	43
Gambar 4.1: Curah Hujan Harian BMKG	46
Gambar 4.2: Arsitektur Model LSTM	47
Gambar 4.3: RH_avg (%) vs RR (mm).....	53
Gambar 4.4: Tavg (°C) vs RR (mm).....	53
Gambar 4.5: ss (jam) vs RR (mm)	54
Gambar 4.6: ff_avg (m/s) vs RR (mm)	54
Gambar 4.7: Grafik <i>Training</i> Model LSTM dengan variasi <i>Batch 4–Epoch 10</i> , dan <i>Batch 4–Epoch 20</i>	60
Gambar 4.8: Grafik <i>Training</i> Model LSTM dengan variasi <i>Batch 4–Epoch 50</i> , dan <i>Batch 4–Epoch 100</i>	61
Gambar 4.9: Grafik <i>Training</i> Model LSTM dengan variasi <i>Batch 16–Epoch 10</i> , dan <i>Batch 16–Epoch 20</i>	62
Gambar 4.10: Grafik <i>Training</i> Model LSTM dengan variasi <i>Batch 16–Epoch 50</i> , dan <i>Batch 16–Epoch 100</i>	62
Gambar 4.11: Grafik <i>Training</i> Model LSTM dengan variasi <i>Batch 32–Epoch 10</i> , dan <i>Batch 32–Epoch 20</i>	63
Gambar 4.12: Grafik <i>Training</i> Model LSTM dengan variasi	

	<i>Batch 32–Epoch 50, dan Batch 32–Epoch 100</i>	64
Gambar 4.13: Grafik <i>Training Model LSTM</i> dengan variasi		
	<i>Batch 64–Epoch 10, dan Batch 64–Epoch 20</i>	64
Gambar 4.14: Grafik <i>Training Model LSTM</i> dengan variasi		
	<i>Batch 64–Epoch 50, dan Batch 64–Epoch 100</i>	65
Gambar 4.15: Grafik <i>Training Model LSTM</i> dengan variasi		
	<i>Batch 128–Epoch 10, dan Batch 128–Epoch 20</i>	66
Gambar 4.16: Grafik <i>Training Model LSTM</i> dengan variasi		
	<i>Batch 128–Epoch 50, dan Batch 128–Epoch 100</i>	66
Gambar 4.17: Grafik Actual Vs LSTM.....		67
Gambar 5.1: Arsitektur Model Attention-Based LSTM		68
Gambar 5.2: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 4–Epoch 10, dan Batch 4–Epoch 20</i>	68
Gambar 5.3: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 4–Epoch 50, dan Batch 4–Epoch 100</i>	72
Gambar 5.4: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 16–Epoch 10, dan Batch 16–Epoch 20</i>	73
Gambar 5.5: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 16–Epoch 50, dan Batch 16–Epoch 100</i>	74
Gambar 5.6: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 32–Epoch 10, dan Batch 32–Epoch 20</i>	75
Gambar 5.7: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 32–Epoch 50, dan Batch 32–Epoch 100</i>	76
Gambar 5.8: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 64–Epoch 10, dan Batch 64–Epoch 20</i>	76
Gambar 5.9: Grafik <i>Training Model AT-LSTM</i> dengan variasi		
	<i>Batch 64–Epoch 50, dan Batch 64–Epoch 100</i>	77
Gambar 5.10: Grafik <i>Training Model AT-LSTM</i> dengan variasi		

<i>Batch</i> 128– <i>Epoch</i> 10, dan <i>Batch</i> 128– <i>Epoch</i> 20	78
Gambar 5.11: Grafik <i>Training</i> Model AT-LSTM dengan variasi	
<i>Batch</i> 128– <i>Epoch</i> 50, dan <i>Batch</i> 128– <i>Epoch</i> 100	78
Gambar 5.12: Grafik Actual Vs Attention-Based LSTM	79
Gambar 6.1: Kurva loss model LSTM Batch 4 dan Epoch 100.....	80
Gambar 6.2: Kurva loss model LSTM Batch 16 dan Epoch 50.....	81
Gambar 6.3: Kurva loss model LSTM Batch 16 dan Epoch 100.....	81
Gambar 6.4: Kurva loss model LSTM Batch 32 dan Epoch 100.....	82
Gambar 6.5: Kurva loss model LSTM Batch 64 dan Epoch 100.....	82
Gambar 6.6: Kurva loss model AT-LSTM Batch 4 dan Epoch 50.....	83
Gambar 6.7: Kurva loss model AT-LSTM Batch 4 dan Epoch 100.....	84
Gambar 6.8: Kurva loss model AT-LSTM Batch 16 dan Epoch 100.....	84
Gambar 6.9: Kurva loss model AT-LSTM Batch 32 dan Epoch 100.....	85
Gambar 6.10: Kurva loss model AT-LSTM Batch 64 dan Epoch 100...	85
Gambar 6.11: Grafik Actual dan Hasil Prediksi LSTM.....	86
Gambar 6.12: Grafik Actual dan Hasil Prediksi Attention-	
Based LSTM	87
Gambar 6.13: Kurva loss model LSTM Batch 4 dan Epoch 20	
dengan Transformasi Data	88
Gambar 6.14: Kurva loss model LSTM Batch 4 dan Epoch 50	
dengan Transformasi Data	89
Gambar 6.15: Kurva loss model LSTM Batch 4 dan Epoch 100	
dengan Transformasi Data	89
Gambar 6.16: Kurva loss model LSTM Batch 16 dan Epoch 100	
dengan Transformasi Data	90
Gambar 6.17: Kurva loss model LSTM Batch 32 dan Epoch 100	
dengan Transformasi Data	90
Gambar 6.18: Kurva loss model AT-LSTM Batch 4 dan Epoch 50	

dengan Transformasi Data	91
Gambar 6.19: Kurva loss model AT-LSTM Batch 4 dan Epoch 100	
dengan Transformasi Data	92
Gambar 6.20: Kurva loss model AT-LSTM Batch 16 dan Epoch 100	
dengan Transformasi Data	92
Gambar 6.21: Kurva loss model AT-LSTM Batch 32 dan Epoch 50	
dengan Transformasi Data	93
Gambar 6.22: Kurva loss model AT-LSTM Batch 128 dan Epoch 50	
dengan Transformasi Data	93
Gambar 6.23: Grafik Actual dan Hasil Prediksi LSTM	
dengan Transformasi Data	94
Gambar 6.24: Grafik Actual dan Hasil Prediksi Attention-	
Based LSTM dengan Transformasi Data	95
Gambar 6.25: Grafik Perbandingan RMSE	100

DAFTAR TABEL

Tabel 2.1: Daftar Jurnal Penelitian.....	18
Tabel 2.2: Variabel Prediksi Curah Hujan	35
Tabel 3.1: Data Curah Hujan Harian BMKG.....	39
Tabel 4.1: Hasil Data Setelah Penanganan <i>Outlier</i>	50
Tabel 4.2: Korelasi Antar Atribut	51
Tabel 4.3: Hasil Normalisasi Data	56
Tabel 4.4: Hasil Pembagian Data.....	57
Tabel 4.5: <i>Hyperparameter</i> Model LSTM.....	58
Tabel 4.6: Hasil Eksperimen Jumlah <i>Neuron</i> pada LSTM	59
Tabel 5.1: <i>Hyperparameter</i> Model Attention-Based LSTM	70
Tabel 6.1: Perbandingan Model Tanpa vs dengan Transformasi Data ...	96
Tabel 6.2: Hasil metrik terbaik LSTM.....	97
Tabel 6.3: Hasil metrik terbaik Attention-Based LSTM.....	98
Lampiran 1: Tabel Metrik Training LSTM.....	115
Lampiran 2: Tabel Metrik Training Attention-Based LSTM	116

ABSTRAK

Romadhani, Ahmad. 2025. **PREDIKSI CURAH HUJAN MENGGUNAKAN ATTENTION-LSTM BERDASARKAN PARAMETER CUACA DI KABUPATEN MALANG**. Program Studi Magister Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. Cahyo Crysdiandian

Kata Kunci: Prediksi Curah Hujan, *Long Short-Term Memory* (LSTM), Mekanisme Attention, Parameter Cuaca, *Deep Learning*, Kabupaten Malang.

Prediksi curah hujan memegang peranan penting dalam perencanaan pertanian dan mitigasi bencana, terutama di wilayah dengan pola iklim yang kompleks seperti Kabupaten Malang, Jawa Timur. Penelitian ini membandingkan kinerja LSTM dan Attention-Based LSTM dalam memprediksi curah hujan harian menggunakan parameter cuaca historis, termasuk suhu rata-rata, kelembaban relatif, durasi sinar matahari, dan kecepatan angin. Dengan memanfaatkan data harian dari tahun 2000 hingga 2023 yang diperoleh dari BMKG, penelitian ini mengikuti alur terstruktur mulai dari praproses data, pemilihan fitur, pelatihan model, hingga evaluasi performa. Model Attention-Based LSTM dirancang untuk memfokuskan perhatian pada fitur temporal yang relevan sehingga meningkatkan kemampuan model dalam menangkap pola curah hujan yang kompleks. Hasil evaluasi menunjukkan bahwa LSTM menghasilkan nilai RMSE sebesar 0.3090, sedangkan Attention-Based LSTM memberikan hasil yang lebih akurat dan lebih stabil dengan RMSE sebesar 0.3068. Temuan ini membuktikan bahwa integrasi *Attention Mechanism* ke dalam arsitektur *deep learning* secara signifikan meningkatkan akurasi, stabilitas, dan ketahanan model terhadap *overfitting* dalam prediksi curah hujan, serta memberikan kontribusi penting dalam strategi adaptasi perubahan iklim dan pengambilan keputusan di sektor pertanian.

ABSTRACT

Romadhani, Ahmad. 2025. **RAINFALL FORECASTING USING ATTENTION-LSTM BASED ON WEATHER PARAMETERS IN MALANG REGENCY.** Magister of Informatics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang. Advisor: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. Cahyo Crysdian

Keywords: Rainfall Forecasting, Long Short-Term Memory (LSTM), Attention Mechanism, Weather Parameters, Deep Learning, Malang Regency.

Rainfall forecasting is crucial for agricultural planning and disaster mitigation, particularly in regions with complex climatic conditions such as Malang Regency, East Java. The research compares the performance of LSTM and Attention-Based LSTM in predicting daily rainfall using historical weather parameters, including mean temperature, relative humidity, sunshine duration, and wind speed. Utilizing daily records from 2000 to 2023 provided by BMKG, the research employs a structured workflow encompassing data preprocessing, feature selection, model training, and performance evaluation. The Attention-Based LSTM model is designed to emphasize relevant temporal features, thereby enhancing the model's ability to capture complex rainfall patterns. The results indicate that the LSTM yielded an RMSE of 0.3090, while the Attention-Based LSTM provided a more accurate and stable result with an RMSE of 0.3068. These findings demonstrate that integrating the Attention Mechanism into the deep learning architecture significantly improves the model's accuracy, stability, and resistance against overfitting in rainfall forecasting, providing a valuable contribution to climate change adaptation strategies and decision-making in the agricultural sector.

مستخلص البحث

رمضاني، أحمد. 2025. التنبؤ بهطول الأمطار باستخدام نموذج الانتباه - الذاكرة طويلة وقصيرة المدى بناءً على معلمات الطقس في مقاطعة مالانج. رسالة الماجستير، قسم المعلومات، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. إروان بودي سانتوسو، الماجستير؛ المشرف الثاني: د. جهيو كريسديان.

الكلمات الرئيسية: تنبؤ بهطول أمطار، ذاكرة طويلة وقصيرة المدى (LSTM)، آلية انتباه، معلمات طقس، تعلم عميق، مقاطعة مالانج. يعتبر التنبؤ بهطول الأمطار أمرًا بالغ الأهمية في التخطيط الزراعي والتخفيف من الكوارث، خاصة في المناطق التي تتميز بأنماط مناخية معقدة مثل مقاطعة مالانج في جاوى الشرقية. تقارن هذه الرسالة أداء نموذج LSTM ونموذج Attention-Based LSTM المعتمد على الانتباه في التنبؤ بهطول الأمطار اليومي باستخدام معلمات الطقس التاريخية، بما في ذلك متوسط درجة الحرارة، الرطوبة النسبية، مدة أشعة الشمس، وسرعة الرياح. من خلال الاستفادة من البيانات اليومية للفترة من عام 2000 حتى 2023 التي تم الحصول عليها من BMKG، تتبع هذه الدراسة مسارًا منظمًا بدءًا من معالجة البيانات الأولية، واختيار الميزات، وتدريب النماذج، وصولًا إلى تقييم الأداء. تم تصميم نموذج LSTM المعتمد على الانتباه للتركيز على الميزات الزمنية ذات الصلة، مما يعزز قدرة النموذج على التقاط أنماط هطول الأمطار المعقدة. أظهرت نتائج التقييم أن نموذج LSTM حقق قيمة RMSE تبلغ 0.3090، في حين أن Attention-Based LSTM المعتمد على الانتباه قدم نتائج أكثر دقة واستقرارًا بقيمة RMSE تبلغ 0.3068. تثبت هذه النتائج أن دمج آلية الانتباه في بنية التعلم العميق يعزز بشكل كبير الدقة والاستقرار ومثانة النموذج تجاه الإفراط في التكيف أثناء توقع هطول الأمطار، كما يساهم بشكل هام في استراتيجيات التكيف مع تغير المناخ واتخاذ القرارات في قطاع الزراعة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Prediksi curah hujan merupakan kegiatan analisis yang berperan besar dalam mendukung pengelolaan sumber daya air serta sektor pertanian. Proses ini biasanya memanfaatkan berbagai parameter meteorologi seperti suhu, tingkat kelembapan, intensitas penyinaran matahari, dan kecepatan angin, karena faktor-faktor tersebut ikut membentuk pola terjadinya hujan. Kondisi Indonesia yang berada di wilayah tropis dan tepat di garis khatulistiwa membuat proses prediksi semakin menantang, terutama dengan adanya perubahan iklim yang menyebabkan pola cuaca berubah secara signifikan (Fredyan & Kusuma, 2022). Mengingat pentingnya prediksi ini, analisis prediktif khususnya dibutuhkan di Kabupaten Malang, Jawa Timur, dengan wilayah yang menyumbang 3,11% dari produksi beras Provinsi Jawa Timur, atau sekitar 17,89% dari produksi nasional pada tahun 2023.

Kabupaten Malang memiliki karakteristik iklim yang cukup beragam, sehingga berdampak langsung pada dinamika curah hujan di wilayah tersebut. Pemahaman yang baik terhadap pola hujan dan kemampuan untuk memproyeksikannya secara tepat menjadi hal yang krusial, baik untuk meminimalkan potensi kerugian akibat perubahan iklim maupun untuk mendukung aktivitas pertanian dan pengelolaan sumber daya air di daerah tersebut. Perubahan iklim yang terjadi secara global juga memperkuat urgensi untuk memiliki model prediksi yang andal dan akurat.

Penelitian ini berfokus pada Kabupaten Malang, Jawa Timur, yang dikenal dengan kondisi iklimnya yang bervariasi. Wilayah ini merupakan area yang sangat relevan untuk penelitian prediksi curah hujan karena kontribusinya yang besar dan potensi kerugian yang signifikan akibat perubahan iklim. Rentang data curah hujan yang dianalisis pada penelitian ini meliputi rentang waktu dari tahun 2000 sampai 2023. Periode yang dipilih memungkinkan terjadinya evaluasi terhadap perubahan pola curah hujan dalam jangka panjang di Kabupaten Malang. Analisis ini mendukung peningkatan kualitas prediksi dan memastikan hasil penelitian tetap relevan. Temuan dari penelitian ini akan bermanfaat bagi banyak pihak, mulai dari pemerintah daerah hingga petani, sebagai bahan pertimbangan dalam merancang kebijakan dan langkah pengelolaan sumber daya air serta aktivitas pertanian.

Air merupakan sumber daya terpenting dalam pertanian dunia, sehingga prakiraan cuaca diperlukan untuk situasi ini. Prediksi curah hujan juga terkait dengan ketersediaan pangan. Hal ini memengaruhi kegagalan panen akibat banjir atau genangan air dan kondisi cuaca lainnya. Mengingat banyak kegiatan yang bergantung pada cuaca/iklim, maka perlu ditemukan metode yang tepat untuk mengurangi risiko kesalahan dari data curah hujan. Curah hujan merupakan fenomena di dunia. Karena kemunculannya, maka akan menjadi sumber kehidupan sebagaimana dijelaskan dalam Surat Al-Furqan ayat 48-50 (Kementerian Agama, Republik Indonesia, 2019).

وَهُوَ الَّذِي أَرْسَلَ الرِّيحَ بُشْرًا بَيْنَ يَدَيْ رَحْمَتِهِ وَأَنْزَلْنَا مِنَ السَّمَاءِ مَاءً طَهُورًا لِّنُخْلِي بِهِ بَلَدَهُ مَيِّتًا
وَنُسْقِيَهُ مِمَّا خَلَقْنَا أَنْعَامًا وَأَنْآسِيَّ كَثِيرًا وَلَقَدْ صَرَّفْنَاهُ بَيْنَهُمْ لِيَذَكَّرُوا فَأَبَى أَكْثَرُ النَّاسِ إِلَّا كُفُورًا

Artinya: “Dan Dialah yang meniupkan angin (sebagai) pembawa kabar gembira sebelum kedatangan rahmat-Nya (hujan); dan Kami turunkan dari langit air yang sangat bersih. Agar (dengan air itu) Kami menghidupkan negeri yang mati (tandus), dan Kami memberi minum kepada sebagian apa yang telah Kami ciptakan, (berupa) hewan-hewan ternak dan manusia yang banyak. Dan sungguh, Kami telah mempergilirkan (hujan) itu di antara mereka agar mereka mengambil pelajaran; tetapi kebanyakan manusia tidak mau (bersyukur), bahkan mereka mengingkari (nikmat).”

Sebagaimana tertuang dalam Al-Qur'an Surat Az-Zumar ayat 21 (Kementerian Agama RI, 2019), dijelaskan proses turunnya hujan sebagai bagian dari tanda-tanda kekuasaan Allah SWT.

أَلَمْ تَرَ أَنَّ اللَّهَ أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَسَلَكَهُ يَنَابِيعَ فِي الْأَرْضِ ثُمَّ يُخْرِجُ بِهِ زَرْعًا مُّخْتَلِفًا أَلْوَانُهُ ثُمَّ يَهِيجُ فَتَرَاهُ مُصْفَرًّا ثُمَّ يَجْعَلُهُ حُطَامًا إِنَّ فِي ذَلِكَ لَذِكْرًا لِأُولَى الْأَلْبَابِ ﴿٢١﴾

Artinya: “Tidakkah engkau memperhatikan bahwa Allah menurunkan air (hujan) dari langit, lalu Dia mengalirkannya menjadi sumber-sumber air di bumi. Kemudian, dengan air itu Dia tumbuhkan tanam-tanaman yang bermacam-macam warnanya, kemudian ia menjadi kering, engkau melihatnya kekuning-kuningan, kemudian Dia menjadikannya hancur berderai. Sesungguhnya pada yang demikian itu benar-benar terdapat pelajaran bagi ululalbab. Tidakkah engkau memperhatikan bahwa Allah menurunkan air (hujan) dari langit, lalu Dia mengalirkannya menjadi sumber-sumber air di bumi. Kemudian, dengan air itu Dia tumbuhkan tanam-tanaman yang bermacam-macam warnanya, kemudian ia menjadi kering, engkau melihatnya kekuning-kuningan, kemudian Dia menjadikannya hancur berderai. Sesungguhnya pada yang demikian itu benar-benar terdapat pelajaran bagi ululalbab.”

Penjelasan mengenai curah hujan dalam ayat tersebut juga diperkuat dengan tafsir hadis yang diriwayatkan oleh Abu Hurairah, Rasulullah SAW bersabda, "Doa seorang hamba tidak akan ditolak ketika ia berdoa saat hujan." Ini menunjukkan pentingnya curah hujan sebagai rahmat dan kesempatan untuk berdoa (Rohmawati & Alim, 2024).

Selain ayat yang membahas tentang curah hujan, penelitian ini turut menyoroti konsep prediksi yang juga dijelaskan dalam Al-Qur'an melalui Surat Al-Mulk ayat 15 (Kementerian Agama RI, 2019).

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَلْتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِغَدٍ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ خَبِيرٌ بِمَا تَعْمَلُونَ ﴿١٥﴾

Artinya: *"Wahai orang-orang yang beriman, bertakwalah kepada Allah dan hendaklah setiap orang memperhatikan apa yang telah diperbuatnya untuk hari esok (akhirat). Bertakwalah kepada Allah. Sesungguhnya Allah Mahateliti terhadap apa yang kamu kerjakan."*

Adapun hadis terkait ayat tersebut merujuk pada sabda Nabi Muhammad SAW:

إِنَّ اللَّهَ لَا يَغَيِّرُ مَا بِقَوْمٍ حَتَّى يُغَيِّرُوا مَا بِأَنْفُسِهِمْ

Artinya: *"Sesungguhnya Allah SWT tidak akan mengubah keadaan suatu kaum shingga mereka mengubah keadaan yang ada pada diri mereka sendiri."* (HR. Al-Rahawi).

Hadis ini menegaskan prinsip dasar dalam ajaran Islam bahwa dalam konteks prediksi dan pengelolaan fenomena alam, termasuk curah hujan, hadis ini memberikan dorongan moral bagi umat Islam untuk berikhtiar melalui ilmu pengetahuan, penelitian, dan perencanaan. Dengan memahami tanda-tanda alam yang telah Allah bentangkan di dunia ini, manusia dituntut untuk mengantisipasi kemungkinan-kemungkinan di masa depan, termasuk perubahan iklim dan distribusi air. Upaya seperti memperkirakan curah hujan, memanfaatkan teknologi prediksi cuaca, hingga mengelola sumber daya air dengan bijak, merupakan bentuk konkret dari implementasi hadis ini dalam kehidupan modern. Bozorgmehri *et al.*, (2023) menekankan bahwa konsep prediksi dalam ajaran Islam berakar pada tanggung jawab moral manusia untuk menafsirkan tanda-tanda alam, selaras dengan fungsi manusia sebagai khalifah. Oleh sebab itu, makna hadis tersebut tidak

hanya spiritual, tetapi juga menjadi motivasi untuk memajukan ilmu pengetahuan demi kepentingan dan kesejahteraan manusia.

Dalam penelitian yang dilakukan oleh Wayan & Suranata, (2023) dan Carnegie & Chairani, (2023) dalam upaya memahami perubahan iklim yang berpengaruh terhadap karakteristik cuaca menuntut adanya model curah hujan yang presisi. Oleh karena itu, penelitian ini berfokus pada pengembangan model prediksi dengan memanfaatkan teknologi kecerdasan buatan. khususnya metode LSTM dan GRU, untuk memperbaiki perubahan iklim. Metodologi yang digunakan meliputi pengumpulan data dari BMKG, penentuan prioritas untuk mengatasi nilai yang hilang, dan analisis data menggunakan matriks konfusi. Untuk menilai seberapa baik model dalam memprediksi curah hujan, penelitian tersebut melakukan perbandingan antara dua arsitektur jaringan saraf berulang, yaitu LSTM dan GRU. LSTM dipilih sebagai model utama karena kemampuannya yang kuat dalam memproses data berurutan, sedangkan GRU dijadikan acuan pembanding karena sifatnya yang lebih ringkas dan efisien. Melalui perbandingan ini, peneliti berupaya menentukan model mana yang mampu memberikan akurasi prediksi tertinggi sekaligus mempertahankan tingkat kesalahan yang minimal. Studi-studi sebelumnya juga banyak melakukan pendekatan serupa, seperti penelitian Musfiroh *et al.*, (2023) yang membandingkan metode *Principal Component Analysis* (PCA) dengan *Long Short-Term Memory* (LSTM), serta penelitian Made *et al.*, (2022) yang mengevaluasi performa Vanilla RNN dan LSTM dalam memprediksi curah hujan. Pada tahap pra-pengolahan, data harian diubah menjadi data berkala per sepuluh hari, dengan suhu, kelembapan, kecepatan angin, dan visibilitas dihitung

melalui nilai rata-rata, sedangkan curah hujan dihitung dengan menjumlahkan total pada periode yang sama. Penelitian lain yang secara khusus menggunakan metode LSTM juga dilakukan oleh Hendra *et al.*, (2023) dan Badriyah *et al.*, (2022). Keduanya menerapkan algoritma *Long Short-Term Memory* untuk memprediksi curah hujan di wilayah yang berbeda, seperti Pekanbaru dan Surabaya. Dengan memanfaatkan riwayat data curah hujan, metode ini mampu menghasilkan prediksi yang relevan sekaligus memberikan gambaran mengenai pola curah hujan di setiap daerah. Temuan tersebut dinilai bermanfaat bagi masyarakat dan pembuat kebijakan dalam menyusun strategi yang berkaitan dengan kondisi cuaca maupun lingkungan. Meskipun sejumlah penelitian terdahulu menunjukkan bahwa LSTM memiliki performa yang sangat baik dan sering kali melampaui metode prediksi lainnya, model ini tetap memiliki beberapa keterbatasan yang perlu menjadi perhatian dalam penerapannya. Penelitian oleh Wayan & Suranata, (2023) dan Carnegie & Chairani, (2023) sangat bagus dengan hasil akurasi metode LSTM yang lebih baik dari GRU, akan tetapi dalam penelitian mereka terdapat kelemahan, yaitu menggunakan dataset yang terbatas (hanya data 1 tahun dari 2019), dan kurangnya interpretasi model dengan periode waktu hanya satu tahun yang berpotensi menghasilkan prediksi yang kurang akurat. Keterbatasan ini mengakibatkan penurunan kemampuan generalisasi model, dan berdampak negatif pada akurasi hasil prediksi. Maka dari itu, solusi dari kelemahan tersebut yaitu menerapkan teknik regularisasi yang lebih kuat, atau menambah kompleksitas model secara bertahap sambil memantau performa pada set validasi, dan menambahkan analisis sensitivitas atau teknik visualisasi untuk memahami bagaimana model membuat

prediksi. Selain itu, Musfiroh *et al.*, (2023) dalam penelitiannya terdapat kelemahan ketergantungan pada kualitas dan data historis yang dapat memengaruhi hasil prediksi, dan performa yang mungkin menurun untuk prediksi jangka panjang. Maka solusinya adalah meningkatkan proses pengumpulan dan pembersihan data, serta menggunakan teknik imputasi data yang lebih canggih, dan menggabungkan LSTM dengan metode lain untuk prediksi curah hujan jangka panjang. Begitu juga dengan Made *et al.*, (2022) dalam penelitiannya terdapat kelemahan pada performa interpretabilitas model yang terbatas dan potensi *overfitting* pada dataset, terutama ketika jumlah data pelatihan terbatas. Solusi dari kelemahan tersebut yaitu menambahkan analisis sensitivitas atau teknik visualisasi untuk memahami lebih baik bagaimana model membuat prediksi, dan menerapkan teknik regularisasi tambahan, untuk meningkatkan ukuran dataset dalam meningkatkan generalisasi model. Penelitian oleh Hendra *et al.*, (2023), dan Badriyah *et al.*, (2022) terdapat kelemahan pada penggunaan data dari BMKG yang hanya mencakup jangka waktu terbatas, yang mengakibatkan keterbatasan dalam ruang lingkup prediksi. Hal ini dapat memengaruhi kemampuan model untuk generalisasi terhadap data yang lebih luas, dan terdapat kelemahan pada waktu pelatihan yang lama. Maka solusi dari kelemahan tersebut yaitu dengan memperbesar atau memperkaya dataset untuk meningkatkan akurasi prediksi, dan mengoptimalkan arsitektur jaringan atau menggunakan *hardware* yang lebih kuat.

Melihat keterbatasan yang masih muncul pada sejumlah penelitian terdahulu serta peluang untuk mengembangkan pendekatan yang lebih optimal, penelitian ini difokuskan pada perbandingan kinerja antara model LSTM dan Attention-Based

LSTM dalam memprediksi curah hujan. Integrasi mekanisme *Attention* ke dalam arsitektur LSTM diharapkan dapat meningkatkan kemampuan model dalam mengidentifikasi pola waktu yang penting serta membantu memusatkan perhatian pada fitur-fitur meteorologi yang paling berpengaruh. Selain itu, penerapan *Attention* diperkirakan dapat menurunkan kecenderungan model mengalami *overfitting* dengan cara memberikan penekanan lebih besar pada informasi yang relevan dan mengabaikan elemen data yang kurang memiliki kontribusi, sehingga kualitas prediksi menjadi lebih stabil dan menghasilkan akurasi yang lebih baik.

1.2 Pernyataan Masalah

1. Seberapa baik performa prediksi curah hujan yang dihasilkan oleh LSTM dan Attention-Based LSTM berdasarkan parameter cuaca di Kabupaten Malang?
2. Seberapa besar peningkatan performa prediksi Attention-Based LSTM dibandingkan dengan LSTM tanpa mekanisme Attention dalam memprediksi curah hujan?

1.3 Tujuan Penelitian

1. Mengukur performa prediksi curah hujan di Kabupaten Malang menggunakan metode LSTM dan Attention-Based LSTM.
2. Mengukur sejauh mana peningkatan performa Attention-Based LSTM dibandingkan dengan LSTM tanpa mekanisme Attention.

1.4 Batasan Masalah

1. Dalam penelitian ini digunakan dataset curah hujan harian yang mencakup seluruh catatan mulai awal tahun 2000, yakni 1 Januari, sampai akhir tahun 2023, yaitu 31 Desember. Rentang waktu ini dipilih untuk memperoleh informasi yang lebih komprehensif mengenai pola curah hujan di Kabupaten Malang.
2. Penelitian ini hanya menggunakan satu sumber data situs resmi BMKG, Stasiun Klimatologi Kabupaten Malang, Jawa Timur, untuk memastikan keakuratan dan relevansi prediksi. Tidak ada penggabungan dengan data iklim lainnya untuk menghindari adanya ketidaksesuaian data atau kompleksitas tambahan.

1.5 Manfaat Penelitian

1. Temuan dari penelitian ini dapat menjadi opsi bagi BMKG dalam menyusun prediksi curah hujan di masa yang akan datang.
2. Penelitian ini dapat dijadikan alternatif bagi Dinas Pertanian Kabupaten Malang untuk merumuskan kebijakan terkait prediksi curah hujan.

BAB II

STUDI PUSTAKA

2.1 Prediksi Curah Hujan

Di samping penjelasan pada bagian pendahuluan, penelitian ini memperoleh dukungan konseptual dari sejumlah penelitian terdahulu yang relevan dengan tema yang dikaji. Topik penelitian ini tetap berada pada ranah prediksi curah hujan, sehingga tinjauan terhadap penelitian sebelumnya menjadi penting. Berikut beberapa studi yang berhubungan:

Dalam studi oleh Han et al., (2023), peneliti menggunakan data bulanan yang berasal dari dua stasiun di kawasan Sungai Yangtze, yakni Yichang dan Pingshan, dengan cakupan waktu dari Januari 1961 hingga Desember 2009. Variabel yang dievaluasi mencakup T_{avg} , RH_{avg} , RR , lama penyinaran matahari (ss), serta rata-rata kecepatan angin (ff_{avg}). Studi ini mengimplementasikan pendekatan Attention-based LSTM sebagai metode pemodelan utama, yang mengintegrasikan dua mekanisme attention pada lapisan input dan tersembunyi untuk secara otomatis mengenali faktor-faktor utama yang mempengaruhi pola runoff. Fungsi aktivasi, dan optimizer yang tepat, seperti ReLU, dan Adam, berkontribusi pada efektivitas model. Struktur layer yang terdiri dari Dense memungkinkan model untuk menangkap kompleksitas data iklim, sementara penggunaan layer Dropout membantu meningkatkan generalisasi model. Dalam penelitian tersebut, performa model dinilai menggunakan sejumlah metrik seperti Nash-Sutcliffe Efficiency, Bias, Pearson's R, dan MARE. Hasil analisis memperlihatkan bahwa AT-LSTM

bekerja lebih baik dibandingkan PCCs-LSTM. Hal ini ditunjukkan oleh peningkatan nilai NSE mencapai 0,873 di Pingshan dan 0,858 di Yichang, serta penurunan error terutama pada saat terjadi aliran tinggi. Selain meningkatkan akurasi, model ini juga mampu menyoroti faktor-faktor iklim terpenting yang memengaruhi prediksi hidrologi jangka panjang. Studi ini relevan dengan penelitian penulis karena sama-sama mengkaji penggunaan LSTM dan Attention-Based LSTM dalam model prediksi berbasis data temporal.

Li *et al.*, (2022) melakukan penelitian dengan memanfaatkan data curah hujan yang diperoleh dari 3.200 stasiun meteorologi yang tersebar di wilayah China dan sekitarnya untuk periode 2015–2017. Dataset yang digunakan berisi berbagai parameter atmosfer, termasuk curah hujan, tekanan udara, temperatur, kecepatan angin, serta variabel meteorologi lain yang berpengaruh terhadap pola hujan. Studi ini menerapkan model Attention-based Long Short-Term Memory (AT-LSTM) untuk memprediksi perubahan intensitas serta pergerakan sistem hujan dalam skala regional. Model tersebut juga memanfaatkan segmentasi spasial guna mengelompokkan sistem hujan berdasarkan tingkatan intensitas, mulai dari hujan ringan hingga badai. Dalam proses pemodelannya, kombinasi fungsi aktivasi sigmoid, optimizer Adam, dan struktur layer Dense digunakan untuk meningkatkan ketepatan prediksi. Evaluasi performa dilakukan menggunakan metrik Root Mean Squared Error (RMSE) dan Threat Score (TS). Hasil analisis menunjukkan bahwa AT-LSTM memberikan performa yang lebih unggul dibandingkan dua model atmosferik populer dan beberapa algoritma machine learning konvensional dalam memprediksi hujan jangka pendek. Model ini terbukti mampu menghasilkan

estimasi pola hujan yang lebih akurat di berbagai wilayah. Adapun dalam penelitian penulis, penggunaan LSTM dan Attention-Based LSTM difokuskan pada perbandingan kinerja kedua model tersebut dalam mengenali pola curah hujan dengan memanfaatkan parameter meteorologi secara lebih presisi serta adaptif terhadap perubahan data.

Penelitian yang dilakukan oleh Badriyah *et al.*, (2022) memanfaatkan data curah hujan dari Stasiun Meteorologi Juanda, Surabaya, untuk periode 2015–2020. Studi tersebut menerapkan algoritma Long Short-Term Memory (LSTM) dengan pembagian data sebesar 85% untuk pelatihan dan 15% untuk pengujian. Dengan dukungan fungsi aktivasi sigmoid, optimizer Adam, serta lapisan Dense, model yang dikembangkan terbukti menghasilkan performa yang lebih baik dibandingkan metode lain seperti RNN dan GRU, sehingga memberikan kontribusi penting dalam bidang prediksi cuaca. Berdasarkan hasil evaluasi, diperoleh nilai Mean Squared Error (MSE) sebesar 0.489, Mean Absolute Error (MAE) sebesar 0.537, serta koefisien determinasi (R^2) sebesar 0.497. Nilai R^2 yang tidak terlalu tinggi mengindikasikan bahwa model LSTM masih memiliki keterbatasan dalam menangkap keragaman data curah hujan di wilayah Surabaya. Meskipun penggunaan data dari banyak stasiun dapat membantu menangkap variasi spasial yang lebih luas, perbedaan karakteristik cuaca antarlokasi juga meningkatkan kompleksitas pemodelan. Studi ini memiliki keterkaitan dengan penelitian penulis, yang tidak hanya menggunakan LSTM tetapi juga Attention-Based LSTM untuk menekankan pola temporal penting serta menangani variasi geografis data dengan lebih adaptif dan efektif.

Pada penelitian yang dilakukan oleh Musfiroh *et al.*, (2023), digunakan 2.099 data meteorologi harian dari wilayah Sulawesi Selatan. Variabel masukan yang digunakan meliputi tekanan udara, evaporasi, suhu maksimum, kelembapan, serta lama penyinaran matahari, sedangkan curah hujan dijadikan sebagai variabel target. Proses pemodelan diawali dengan penerapan Principal Component Analysis (PCA) untuk seleksi variabel, kemudian dilanjutkan dengan algoritma LSTM sebagai metode prediksi. Pendekatan ini memanfaatkan fungsi aktivasi sigmoid dan optimizer Adam, yang terbukti efektif dalam mengelola kompleksitas data cuaca. Dengan bantuan layer Dense, model mampu mengenali pola pada data historis dan menghasilkan prediksi yang cukup presisi. Tingkat akurasi model tercermin dari nilai Mean Absolute Percentage Error (MAPE) sebesar 0.0018, yang menunjukkan bahwa kesalahan prediksi curah hujan relatif sangat kecil. Temuan ini menegaskan kemampuan LSTM dalam pemodelan cuaca, sementara secara teoritis, penggunaan Attention-Based LSTM berpotensi semakin meningkatkan ketepatan prediksi pada data multivariat.

Dalam studi yang dilakukan oleh Carnegie & Chairani (2023), digunakan data cuaca selama lima tahun yang diperoleh dari BMKG Stasiun Geofisika Lampung Utara untuk memodelkan prediksi curah hujan menggunakan dua algoritma, yaitu LSTM dan GRU. Penelitian tersebut menerapkan fungsi aktivasi sigmoid dan tanh, menggunakan optimizer Adam, serta memanfaatkan layer Dense dalam arsitektur jaringannya. Temuan penelitian menunjukkan bahwa pemilihan kombinasi fungsi aktivasi, optimizer, dan struktur layer yang sesuai memainkan peran penting dalam meningkatkan akurasi prediksi pada model LSTM maupun GRU. Berdasarkan hasil

evaluasi, model LSTM memberikan performa paling unggul dengan nilai Root Mean Squared Error (RMSE) sebesar 16.81, Mean Squared Error (MSE) sebesar 282.55, dan Mean Absolute Deviation (MAD) sebesar 10.43. Nilai tersebut mengindikasikan bahwa LSTM memiliki kemampuan yang baik dalam mengolah data cuaca untuk menghasilkan estimasi curah hujan yang akurat. Temuan ini juga mengisyaratkan bahwa penerapan Attention-Based LSTM berpotensi meningkatkan hasil prediksi lebih lanjut, khususnya dalam menangani pola data yang lebih kompleks.

Penelitian yang dilakukan oleh Made *et al.*, (2022) menggunakan data cuaca harian dari Stasiun Meteorologi Gusti Ngurah Rai, yang meliputi variabel curah hujan, suhu udara, dan kelembapan. Studi tersebut membandingkan performa model Long Short-Term Memory (LSTM) dengan Vanilla Recurrent Neural Network (RNN), disertai proses seleksi fitur menggunakan metode Backward Elimination. Baik pada model LSTM maupun Vanilla RNN, digunakan fungsi aktivasi sigmoid. Optimizer yang diterapkan adalah RMSProp, sementara arsitektur jaringan memanfaatkan kombinasi layer Dense dan Dropout. Berdasarkan hasil evaluasi, model LSTM memperoleh nilai R-Squared (R^2) sebesar 0.6139 serta Root Mean Square Error (RMSE) sebesar 28.4308. Hasil ini menunjukkan bahwa LSTM mampu memberikan tingkat akurasi yang cukup baik dalam memprediksi data deret waktu, khususnya data curah hujan. Secara keseluruhan, penelitian tersebut menegaskan keunggulan LSTM dalam menangani pola data yang bersifat berurutan, dan penerapan Attention-Based LSTM diperkirakan dapat meningkatkan akurasi prediksi, terutama pada pola curah hujan yang lebih kompleks.

Penelitian yang dilakukan oleh Farikhul Firdaus & Paputungan (2022) menggunakan data curah hujan harian dari BMKG Kota Bandung periode 2017–2021. Dalam pengembangannya, model memanfaatkan fungsi aktivasi tanh dan sigmoid, dengan optimizer Adam serta arsitektur jaringan yang terdiri atas layer Dense dan Dropout. Pendekatan tersebut mampu menghasilkan performa prediksi yang baik, sehingga memperkuat efektivitas LSTM dalam menganalisis data deret waktu. Berdasarkan evaluasi menggunakan Root Mean Square Error (RMSE), model LSTM memperoleh nilai RMSE sebesar 12.24 pada data pelatihan dan 8.86 pada data pengujian. Hasil ini mengindikasikan bahwa model cukup andal dalam memprediksi curah hujan harian, sehingga dapat memberikan manfaat bagi perencanaan kegiatan luar ruangan. Secara keseluruhan, penelitian tersebut menegaskan kemampuan LSTM dalam menangani data temporal yang bersifat dinamis, dan penggunaan Attention-Based LSTM pada penelitian penulis diharapkan mampu memberikan peningkatan akurasi yang lebih signifikan.

Penelitian yang dilakukan oleh Hendra *et al.*, (2023) memanfaatkan data curah hujan dari BMKG Pekanbaru untuk periode 2010–2020. Data dibagi menjadi dua bagian yang seimbang, yaitu 50% untuk pelatihan dan 50% untuk pengujian. Model LSTM yang digunakan menerapkan fungsi aktivasi tanh dan sigmoid, serta optimizer RMSprop, dengan arsitektur yang mencakup layer Dense untuk menangkap pola temporal dan layer Dropout guna mengurangi risiko overfitting. Hasil evaluasi menunjukkan bahwa model menghasilkan tingkat kesalahan terendah sebesar 21.328 berdasarkan metrik Root Mean Squared Error (RMSE) pada data pelatihan, sementara Mean Squared Error (MSE) pada data pengujian

mencapai 454.901. Temuan ini mengindikasikan bahwa meskipun performa model cukup baik, terdapat perbedaan tingkat akurasi antara data pelatihan dan pengujian yang perlu diperhatikan. Penelitian tersebut menegaskan potensi LSTM dalam pemodelan data deret waktu, dan penggunaan Attention-Based LSTM dalam penelitian penulis diperkirakan dapat meningkatkan akurasi prediksi melalui pemfokusan pada pola temporal yang lebih relevan.

Penelitian yang dilakukan oleh Wayan & Suranata (2023) menggunakan data cuaca dari BMKG Stasiun Denpasar yang mencakup periode satu tahun pada 2019. Studi ini membandingkan performa model Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) dalam memprediksi curah hujan. Tahap pre-processing dilakukan secara menyeluruh, meliputi penanganan nilai hilang dan pelabelan data untuk memastikan kualitas data sebelum proses pemodelan. Model memanfaatkan fungsi aktivasi ReLU, optimizer Adam, serta struktur layer Dense dengan jumlah neuron bervariasi di setiap layer. Hasil evaluasi menunjukkan bahwa LSTM memberikan performa yang lebih unggul dibandingkan GRU. Model LSTM mencapai nilai Root Mean Squared Error (RMSE) sebesar 8.853, Mean Absolute Error (MAE) 4.090, dan Mean Squared Error (MSE) 78.383. Sementara itu, GRU menghasilkan RMSE sebesar 9.698, MAE 4.291, dan MSE 94.058. Perbedaan ini menandakan bahwa LSTM memiliki tingkat akurasi sekitar 8,7% lebih baik dibandingkan GRU dalam memprediksi curah hujan. Meskipun demikian, penelitian ini memiliki keterbatasan karena hanya menggunakan data satu tahun, sehingga kemampuan generalisasi model terhadap periode data yang lebih panjang belum dapat dievaluasi sepenuhnya. Studi tersebut tetap relevan dengan penelitian

penulis, yang menerapkan LSTM dan Attention-Based LSTM untuk meningkatkan akurasi prediksi curah hujan dalam jangka panjang dengan memanfaatkan informasi temporal yang lebih kaya.

Penelitian yang dilakukan oleh Nayak *et al.*, (2022) memanfaatkan data curah hujan dari tiga subdivisi di Karnataka dengan membandingkan performa tiga metode, yaitu Gated Recurrent Unit (GRU), Multi-Layer Perceptron (MLP), dan ARIMA untuk memprediksi curah hujan di wilayah tersebut. GRU, yang merupakan varian sederhana dari LSTM dengan mekanisme gerbang untuk mengelola ketergantungan jangka panjang, diterapkan bersama beberapa fungsi aktivasi seperti ReLU, tanh, dan sigmoid, serta menggunakan optimizer Adam dalam proses pelatihannya. Hasil penelitian menunjukkan bahwa GRU memberikan performa terbaik dengan nilai Root Mean Absolute Error (RMSE) sebesar 149,45, meskipun model ini masih memiliki kelemahan dalam menangkap pola temporal yang sangat panjang. Temuan ini menegaskan efektivitas GRU dalam pemodelan cuaca, dan menjadi landasan bahwa penerapan LSTM serta Attention-Based LSTM pada penelitian penulis berpotensi memberikan peningkatan akurasi yang lebih signifikan dalam prediksi curah hujan.

Penelitian yang dilakukan oleh Buathongkhue *et al.*, (2024) mengkaji keterkaitan antara Oceanic Niño Index (ONI) dan curah hujan di Thailand dengan memanfaatkan metode Recurrent Neural Network (RNN). RNN, sebagai jaringan saraf berulang yang dirancang untuk memproses data berurutan, memiliki kemampuan dasar untuk mempelajari pola temporal, namun cenderung mengalami kendala vanishing gradient ketika dihadapkan pada urutan data yang panjang,

sehingga performanya berada di bawah LSTM. Dalam penelitian tersebut, fungsi aktivasi ReLU digunakan, sedangkan sigmoid berperan dalam forget gate, input gate, dan output gate pada arsitektur LSTM. Model RNN menghasilkan nilai Mean Absolute Error (MAE) sebesar 3.272, meskipun masih terbatas dalam menangkap hubungan non-linear yang kompleks antara variabel iklim. Temuan ini menunjukkan bahwa meskipun RNN dapat dimanfaatkan untuk analisis cuaca, penggunaan LSTM maupun Attention-Based LSTM memiliki potensi yang lebih besar untuk meningkatkan akurasi prediksi karena kemampuan keduanya dalam memodelkan dinamika temporal secara lebih mendalam.

Tahap selanjutnya adalah menjelaskan dan memilih metode terbaik yang diperoleh dengan mengkaji data penelitian sebelumnya. Berbagai penelitian atau jurnal yang dikutip dalam penelitian ini disajikan pada Tabel 2.1:

Tabel 2.1 Penelitian terkait

No	Peneliti	Topik	Metode	Hasil Akurasi
1	Han <i>et al.</i> , (2023)	<i>Attention-Based LSTM</i> (AT-LSTM)	Data Curah Hujan Stasiun Yichang dan Pingshan, di Sungai Yangtze, China	97%
2	Li <i>et al.</i> , (2022)	<i>Attention-Based LSTM</i> (AT-LSTM)	Data Curah Hujan Stasiun Meteorologi di China	96%
3	Badriyah <i>et al.</i> , (2022)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Meteorologi Juanda Surabaya	95%
4	Musfiroh <i>et al.</i> , (2023)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Meteorologi Sulawesi Selatan	94%
5	Carnegie & Chairani, (2023)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Geofisika Lampung Utara	93%
6	Made <i>et al.</i> , (2022)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Meteorologi Gusti Ngurah Rai	92%
7	Farikhul Firdaus & Papatungan, (2022)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Kota Bandung	92%
8	Hendra <i>et al.</i> , (2023)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Pekanbaru	90%
9	Wayan & Suranata, (2023)	<i>Long Short-Term Memory</i> (LSTM)	Data Curah Hujan Stasiun Geofisika Denpasar	90%

Lanjutan				
10	Nayak <i>et al.</i> , (2023)	<i>Gate Recurrent Unit (GRU)</i>	Data curah hujan 3 subdivisi Karnataka	85%
11	Buathongkhue <i>et al.</i> , (2024)	<i>Recurrent Neural Network (RNN)</i>	Data Indeks Niño Oseania (ONI) dan curah hujan bulanan	80%

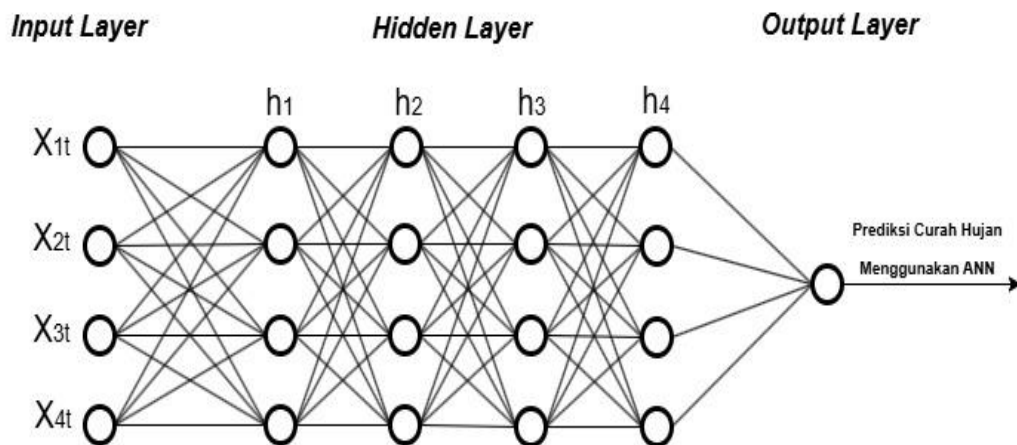
2.2 Perkembangan Jaringan Syaraf Tiruan menuju *Attention-Based LSTM*

Perkembangan kecerdasan buatan (*Artificial Intelligence/AI*) telah menghasilkan berbagai model jaringan saraf tiruan yang dirancang untuk meniru cara kerja otak manusia dalam memproses informasi. Salah satu fondasinya adalah *Artificial Neural Network* (ANN), yang kemudian berkembang menjadi arsitektur yang lebih canggih seperti *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (LSTM), hingga *Attention-Based LSTM* (AT-LSTM). Evolusi model-model ini bertujuan meningkatkan kemampuan dalam menangani data berurutan serta mengenali pola temporal secara lebih efektif. Perkembangan tersebut menunjukkan adanya kemajuan logis menuju model prediksi yang lebih adaptif dan akurat. Dalam konteks penelitian ini, pendekatan LSTM-Attention dipilih untuk memprediksi curah hujan di Kabupaten Malang karena kemampuannya dalam menangkap dinamika data meteorologi secara lebih mendalam dan responsif terhadap perubahan pola cuaca.

2.2.1 *Artificial Neural Network* (ANN)

Jaringan Syaraf Tiruan (*Artificial Neural Network* / ANN) merupakan suatu model komputasi yang dikembangkan berdasarkan konsep kerja otak manusia dalam mengolah informasi. ANN tersusun atas neuron-neuron buatan yang saling terhubung melalui bobot (*weights*), di mana setiap neuron menerima masukan,

memprosesnya menggunakan fungsi aktivasi, kemudian menghasilkan keluaran yang diteruskan ke neuron selanjutnya. Secara struktur, ANN biasanya terdiri dari tiga lapisan utama: lapisan input, lapisan tersembunyi (*hidden layer*), dan lapisan output. Proses pelatihannya umumnya menggunakan algoritma *backpropagation*, yang berfungsi memperbarui bobot berdasarkan perbedaan antara nilai prediksi dan nilai target. Meskipun ANN efektif dalam memodelkan hubungan non-linear antara variabel input dan output, model ini kurang optimal untuk menangani data berurutan (*sequential data*) seperti *time series* karena seluruh *input* diperlakukan sebagai data yang tidak saling berkaitan (Salehin & Kang, 2023).



Gambar 2. 1 Arsitektur ANN

Penjelasan dari arsitektur ANN adalah sebagai berikut:

1) *Input Layer*

Pada bagian ini, jaringan menerima sejumlah fitur pada waktu ke- t :

$x_t = (x_{1t}, x_{2t}, x_{3t}, x_{4t})$. *Input* ini berupa vektor fitur yang menjadi masukan untuk semua *neuron* di *hidden layer*.

2) *Hidden Layer*

- Perkalian bobot dan penjumlahan (*Linear Combination*).

Semua *input* dikalikan bobot masing-masing lalu dijumlahkan:

$$z_i = \sum_{j=1}^n \omega_{ij} x_{jt} + b_i \quad (2.1)$$

Dimana:

ω_{ij} = bobot dari *input* ke-j menuju *neuron* ke-i

b_i = bias *neuron* ke-i

n = jumlah fitur *input*

- Aktivasi Non-Linear.

Hasil *linear combination* dilewatkan ke fungsi aktivasi tanh:

$$h_i = f(z_i) \quad (2.2)$$

- *Forwarding* ke *Neuron* Berikutnya

Output dari *neuron hidden layer* diteruskan ke *neuron* lain di *hidden layer* berikutnya atau ke *output layer*:

$$H = [h_1, h_2, h_3, h_4] \quad (2.3)$$

3) *Output Layer*

- *Output layer* menerima seluruh hasil aktivasi dari *hidden layer*. Berikut rumusnya:

$$y' = \sum_{i=1}^m v_i h_i + b_o \quad (2.4)$$

Dimana:

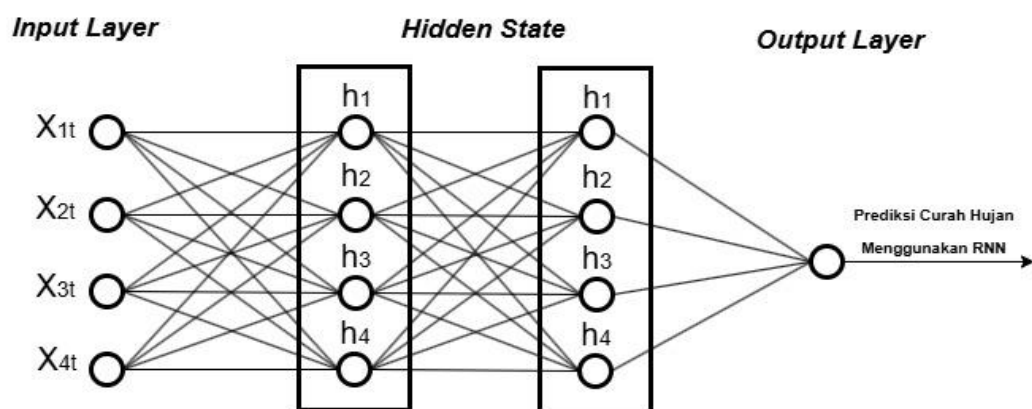
v_i = bobot dari *neuron hidden* ke *neuron output*

m = jumlah *neuron* pada *hidden layer*

b_o = bias pada *output layer*

2.2.2 Recurrent Neural Network (RNN)

Untuk mengatasi keterbatasan ANN dalam menangani data yang memiliki keterurutan, dikembangkanlah *Recurrent Neural Network* (RNN). Arsitektur ini dirancang dengan mekanisme umpan balik (*feedback connection*) yang memungkinkan informasi dari satu langkah waktu disimpan dan digunakan kembali pada langkah waktu selanjutnya. Dengan cara ini, RNN mampu mempertahankan konteks temporal dalam proses pemodelan data. Namun, dalam penerapannya, RNN sering menghadapi kendala berupa *vanishing gradient* dan *exploding gradient* ketika dihadapkan pada urutan data yang panjang. Permasalahan tersebut menyebabkan model sulit mempertahankan informasi dari awal deret waktu, sehingga performanya menurun dalam mempelajari ketergantungan jangka panjang, termasuk pola cuaca maupun curah hujan (Buathongkhue *et al.*, 2024).



Gambar 2. 2 Arsitektur RNN

Penjelasan dari arsitektur RNN adalah sebagai berikut:

1) *Input Layer*

Pada bagian ini, *input* seri waktu berupa fitur pada waktu ke- t :

$x_t = (x_{1t}, x_{2t}, x_{3t}, x_{4t})$. *Input* ini berupa vektor fitur yang menjadi masukan.

2) *Hidden State (Recurrent Loop)*

- Ini bagian terpenting. Tiap *timestep* punya *hidden state*:

$$h_t = f(W_x x_t + W_h h_{t-1} + b) \quad (2.5)$$

Dimana:

W_x = bobot input

W_h = bobot recurrent (dari hidden sebelumnya)

h_{t-1} = memori timestep sebelumnya

f = fungsi aktivasi tanh

3) *Output Layer*

- *Output layer* prediksi curah hujan menggunakan RNN. Berikut rumusnya:

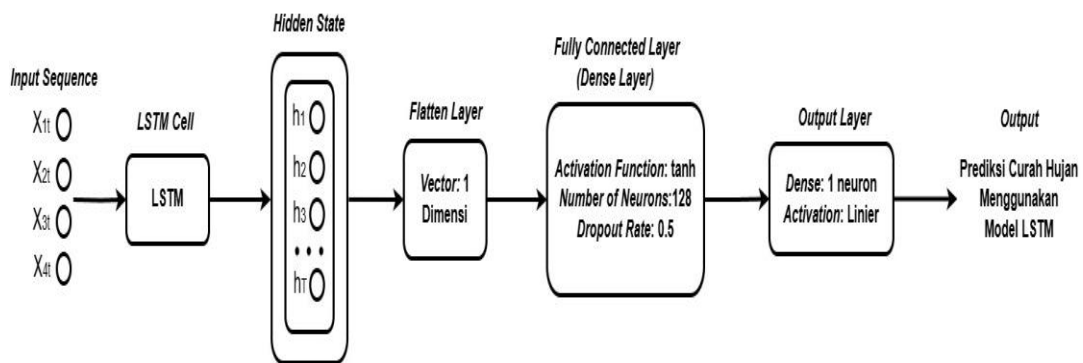
$$\hat{y} = W_y h_T + b_y \quad (2.6)$$

2.2.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan varian dari jaringan saraf berulang (*Recurrent Neural Network*/RNN) yang dikembangkan untuk mengatasi permasalahan *vanishing* dan *exploding gradient* yang umum muncul saat melatih model berbasis urutan atau jaringan saraf dalam. Setiap unit LSTM memiliki komponen berupa sel memori (*cell*), gerbang masukan (*input gate*), gerbang keluaran (*output gate*), serta gerbang lupa (*forget gate*). Ketiga gerbang tersebut berfungsi mengatur aliran informasi sehingga model dapat mempertahankan dan memanfaatkan informasi jangka panjang secara lebih optimal. Secara struktural,

LSTM tersusun atas kumpulan sub-jaringan berulang yang dikenal sebagai *memory blocks*. Konsep utama *memory block* adalah kemampuannya dalam mempertahankan *state* selama beberapa langkah waktu dan mengendalikan informasi yang diteruskan melalui gerbang-gerbang non-linear (Houdt et al., 2022).

Berikut Arsitektur dari LSTM:



Gambar 2.3 Arsitektur LSTM

Penjelasan dari arsitektur LSTM adalah sebagai berikut:

1) *Input Sequence*

- Data *input* diwakili oleh $x = (x_{1t}, x_{2t}, x_{3t}, x_{4t})$, di mana:
 - x adalah sekumpulan data sekuensial (*time series*) yang digunakan sebagai *input* model.
 - T adalah panjang urutan (*timesteps*), yaitu jumlah periode waktu yang digunakan dalam data sekuensial.
 - x_t adalah vektor fitur pada timestep ke- t yang terdiri dari beberapa fitur:
 - x_{1t} = Suhu rata-rata (Tavg)
 - x_{2t} = Kelembaban relatif rata-rata (RH_avg)
 - x_{3t} = Durasi penyinaran matahari (ss)

- x_{4t} = Kecepatan angin rata-rata (ff_avg)
- *Input* ini direpresentasikan sebagai tensor dengan bentuk (*batch_size*, *timesteps*, *features*), di mana:
 - *batch_size* adalah jumlah sampel dalam satu *batch* pelatihan.
 - *timesteps* adalah jumlah periode waktu (*T*) yang digunakan dalam model.
 - *features* adalah jumlah atribut (Tavg, RH_avg, ss, dan ff_avg) pada setiap *timestep*.

2) LSTM Cell

- Lapisan LSTM digunakan untuk mempelajari dependensi temporal dari data sekuensial.
- Pada setiap *timestep* *t*, LSTM memproses *input* x_t dan menghasilkan dua keluaran:
 - Hidden State (h_t): Representasi informasi jangka pendek yang akan diteruskan ke *timestep* berikutnya.
 - Cell State (c_t): Penyimpan informasi jangka panjang yang diperbarui berdasarkan *input* dan *state* sebelumnya.
- Fungsi utama LSTM dapat dirumuskan sebagai berikut:
 - *Input Gate*: Mengatur seberapa banyak informasi baru dari *input* saat ini (x_t) yang akan disimpan dalam *cell state*.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.7)$$
 - *Forget Gate*: Memutuskan informasi mana dari *cell state* sebelumnya (C_{t-1}) yang perlu dilupakan atau dihapus.

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.8)$$

- *Cell State* (c_t): Penyimpan utama informasi jangka panjang yang diperbarui berdasarkan masukan dari *forget gate* dan *input gate*.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.9)$$

- *Output Gate*: Mengontrol bagian dari *cell state* yang akan diubah menjadi *hidden state* dan digunakan untuk *output*.

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.10)$$

- *Hidden State* (h_t): Representasi informasi jangka pendek dalam LSTM, yang digunakan untuk *output* saat ini dan kemudian akan diteruskan ke *timestep* berikutnya.

$$h_t = O_t \odot \tanh(C_t) \quad (2.11)$$

- Di sini, σ adalah fungsi sigmoid, \odot adalah operasi elemen-per-elemen (*Multiplication*), dan W, b adalah bobot dan bias masing-masing.

3) Flatten Layer

- Lapisan Flatten berfungsi untuk mengubah *output* tensor dari LSTM yang biasanya berbentuk tiga dimensi yaitu *batch_size*, *timesteps*, *features* (3D) menjadi vektor satu dimensi (1D) yang merupakan representasi data yang sudah diringkas jadi satu baris angka. Proses ini meratakan seluruh nilai dari setiap langkah waktu menjadi satu rangkaian panjang, sehingga hasilnya dapat diteruskan ke lapisan Dense untuk proses pembelajaran dan prediksi akhir.

- *Hidden state* pada setiap *timestep* ($h_1, h_2, h_3, \dots, h_T$) diratakan menjadi vektor 1 dimensi:

$$H = [h_1, h_2, h_3, \dots, h_T] \in \mathbb{R}^{T \times d} \quad (2.12)$$

4) *Fully Connected Layer (Dense Layer)*

- Lapisan ini memiliki 128 *neuron* yang berfungsi untuk mengubah hasil keluaran dari lapisan *Flatten* menjadi representasi fitur berdimensi tetap sebelum menuju *output layer*. Jumlah 128 dipilih secara empiris untuk memberikan kapasitas yang cukup dalam menangkap hubungan non-linear antar fitur hasil pemrosesan LSTM, dengan fungsi aktivasi tanh, dan *Dropout Rate*: 0.5, yang berarti 50% *neuron* akan dinonaktifkan secara acak selama pelatihan untuk mencegah *overfitting*. f adalah fungsi aktivasi yang memproses input (x) yang merupakan vektor fitur yang berasal dari lapisan sebelumnya. Rumusnya:

$$f(x) = \tanh(W_c \cdot c + b) \quad (2.13)$$

5) *Output Layer*

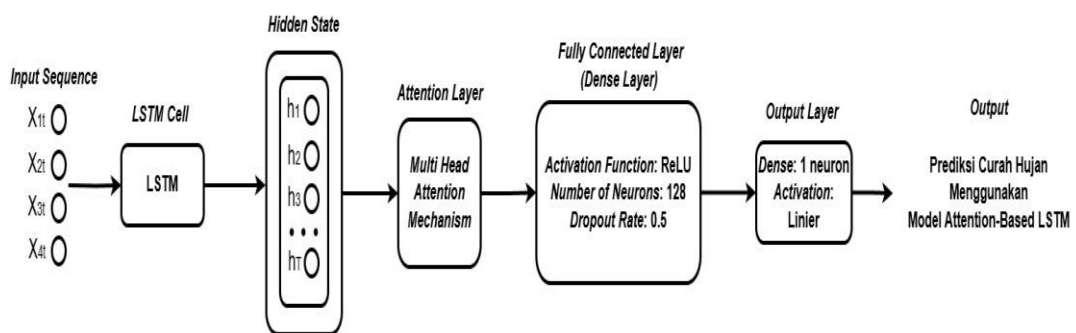
- Lapisan *output* adalah lapisan terakhir dalam arsitektur model dan hanya terdiri dari satu *neuron*. Lapisan ini merupakan kelanjutan dari *output Dense Layer* sebelumnya, yang berisi fitur-fitur utama yang diekstraksi melalui sel LSTM, dan Dense Layer itu sendiri. Pada lapisan ini, fungsi aktivasi linier digunakan, yang secara matematis tidak menerapkan transformasi apa pun pada input tetapi hanya menghitung kombinasi linier dari *output* sebelumnya, \hat{y} merepresentasikan curah hujan (RR) yang diprediksi dalam

milimeter (mm), w_o adalah bobot neuron *output*, f adalah *output* dari *Dense Layer* sebelumnya (yang berisi fitur-fitur yang telah dipelajari), dan b_o adalah bias dari neuron *output*. Berikut rumusnya:

$$\hat{y} = w_o \cdot f + b_o \quad (2.17)$$

2.2.4 Attention-Based LSTM (AT-LSTM)

Attention-Based LSTM (AT-LSTM) merupakan pengembangan dari arsitektur *Long Short-Term Memory* (LSTM) yang dilengkapi dengan mekanisme *Attention* untuk memperkuat kemampuan model dalam mengekstraksi informasi penting dari data berurutan. Melalui mekanisme tersebut, model dapat memberikan bobot lebih besar pada bagian data yang dianggap relevan dan mengurangi pengaruh informasi yang kurang diperlukan. Pendekatan ini tidak hanya meningkatkan akurasi dan efisiensi dalam proses pemodelan, tetapi juga membuat hasil prediksi lebih mudah diinterpretasikan, khususnya dalam konteks peramalan curah hujan (Han *et al.*, 2023). Berikut Arsitektur dari Attention-Based LSTM:



Gambar 2. 4 Arsitektur Attention-Based LSTM

Penjelasan dari arsitektur Attention-Based LSTM adalah sebagai berikut:

1) *Input Sequence*

- Data *input* direpresentasikan sebagai tensor $x = (x_{1t}, x_{2t}, x_{3t}, x_{4t})$, yaitu:

- x merupakan sekumpulan data sekuensial (*time series*) sebagai *input* model.
- T ialah panjang urutan (*timesteps*), yaitu jumlah periode waktu yang diterapkan dalam data sekuensial.
- x_t yaitu vektor fitur pada timestep ke- t yang terdiri dari beberapa fitur yaitu: x_{1t} adalah suhu rata-rata (T_{avg}), x_{2t} ialah kelembaban relatif rata-rata (RH_avg), x_{3t} merupakan durasi penyinaran matahari (ss), dan x_{4t} yakni kecepatan angin rata-rata (ff_avg).
- *Input* berikut ini direpresentasikan sebagai tensor dengan bentuk ($batch_size, timesteps, features$), di mana: $batch_size$ merupakan jumlah sampel dalam satu *batch* pelatihan, $timesteps$ ialah jumlah periode waktu (T) yang digunakan dalam model, dan $features$ yaitu jumlah atribut (T_{avg} , RH_avg , ss , dan ff_avg) pada setiap *timestep*.

2) *LSTM Cell*

- Lapisan LSTM berfungsi untuk mempelajari dependensi temporal (ketergantungan waktu) dan pola jangka panjang dari data sekuensial.
- Pada setiap *timestep* t , LSTM memproses *input* x_t dan menghasilkan dua keluaran utama yaitu: Hidden State (h_t) yang merupakan representasi informasi jangka pendek yang akan diteruskan ke *timestep* berikutnya, yaitu ke dalam mekanisme *Attention*. Kemudian Cell State (c_t) sebagai penyimpanan informasi jangka panjang yang diperbarui dan dikelola oleh gerbang-gerbang LSTM berdasarkan *input* dan *state* sebelumnya.
- Fungsi utama LSTM dapat dirumuskan sebagai berikut:

- *Input Gate*: Menentukan nilai mana dalam *input* saat ini (x_t) yang akan diperbarui, kemudian disimpan dalam *cell state*.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.18)$$

- *Forget Gate*: Menentukan seberapa banyak informasi dari *cell state* sebelumnya (C_{t-1}) yang perlu dihilangkan atau dipertahankan.

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.19)$$

- *Cell State* (c_t): Menggabungkan hasil dari dua fungsi sebelumnya, yang diperbarui berdasarkan masukan dari *forget gate* dan *input gate*.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.20)$$

- *Output Gate*: Menentukan bagian mana dari *cell state* yang akan diubah menjadi *hidden state* dan digunakan untuk *output*.

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.21)$$

- *Hidden State* (h_t): Ringkasan informasi jangka pendek dari LSTM yang digunakan sebagai keluaran saat ini dan diteruskan ke *timestep* berikutnya.

$$h_t = O_t \odot \tanh(C_t) \quad (2.22)$$

3) *Attention Layer*

- *Attention Layer* digunakan untuk memberikan bobot lebih besar pada *timestep* tertentu dalam data *input* yang dianggap lebih relevan untuk tugas prediksi. Pada *Attention-Based LSTM*, mekanisme *Attention* secara langsung menghasilkan *context vector* berdimensi tetap (1D) yang

merupakan hasil pembobotan dari *output* LSTM di setiap langkah waktu. Vektor ini sudah mewakili seluruh informasi temporal yang relevan, sehingga lapisan *Flatten* tidak lagi diperlukan. *Context vector* tersebut kemudian diteruskan langsung ke lapisan Dense untuk proses pembelajaran dan menghasilkan prediksi akhir.

- *Multi-Head Attention Mechanism* menangkap informasi global dari semua *hidden state* ($h_1, h_2, h_3, \dots, h_T$) yang dihasilkan oleh LSTM. Setiap *hidden state* menyimpan informasi dari *timestep* sebelumnya dan saat ini, sehingga mekanisme perhatian dapat memilih *hidden state* mana yang memiliki pengaruh paling besar terhadap *output* akhir.

- Rumus:

- *Attention Score* (e_t)

Skor dihitung dengan *dot product* antara hidden state h_t dan vektor parameter W_a , adalah matriks bobot:

$$e_t = \tanh(W_a \cdot h_t) \quad (2.23)$$

- *Attention Weights* (a_t)

Skor yang telah dihitung diubah menjadi probabilitas menggunakan softmax, untuk menunjukkan seberapa penting *timestep* t :

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^T \exp(e_k)} \quad (2.24)$$

- *Context Vector* (C_t)

Vektor konteks diperoleh sebagai kombinasi linear dari *hidden state* h_t dengan bobot *attention*:

$$c_t = \sum_{t=1}^T a_t \cdot h_t \quad (2.25)$$

4) *Fully Connected Layer (Dense Layer)*

- Lapisan Dense ini menggunakan 128 *neuron* berfungsi untuk memproses *context vector* yang dihasilkan oleh mekanisme *Attention*. Jumlah *neuron* ini memungkinkan model mengoptimalkan representasi dari informasi yang sudah diberi bobot relevansi oleh *Attention*, dengan fungsi aktivasi ReLU. Untuk mencegah *overfitting* dan meningkatkan generalisasi, *Dropout Rate* 0.5 diterapkan, yang secara acak menonaktifkan 50% dari *neuron* selama setiap iterasi pelatihan. Akhirnya, *output* dari lapisan ini dimasukkan ke *Dense 1 neuron* untuk prediksi curah hujan. Berikut rumusnya:

$$f(x) = \text{ReLU}(W_c \cdot c + b) \quad (2.26)$$

5) *Output Layer*

- Lapisan *Output* adalah Lapisan *Dense* terakhir yang terdiri dari satu *neuron* dengan fungsi aktivasi linear, bertugas menghasilkan nilai prediksi curah hujan. Berikut rumusnya:

$$\hat{y} = w_o \cdot f + b_o \quad (2.27)$$

2.3 *Evaluation Metric*

Metrik evaluasi yang digunakan dalam penelitian ini adalah *Root Mean Square Error* (RMSE), yang berfungsi untuk mengukur besarnya kesalahan antara nilai aktual dan hasil prediksi. RMSE dihitung dari selisih kuadrat antara nilai sebenarnya dan nilai yang diprediksi, kemudian diambil akar kuadratnya. Nilai

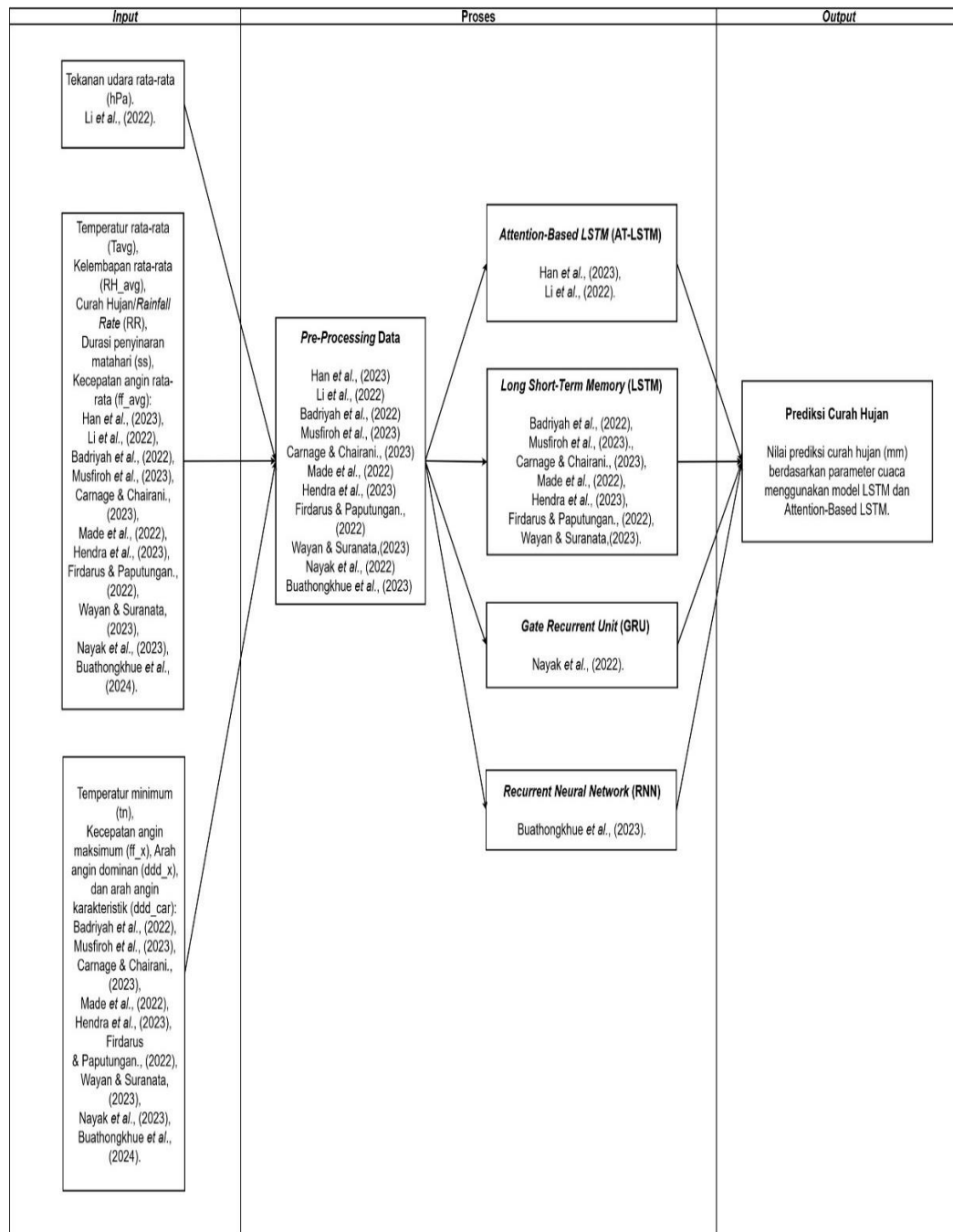
RMSE yang semakin kecil menunjukkan bahwa model memiliki tingkat akurasi yang semakin baik (Muttaqien *et al.*, 2022). Perhitungan RMSE dapat dilihat pada persamaan berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N ||y(i) - \hat{y}(i)||^2}{N}} \quad (2.28)$$

2.4 Kerangka Teori

Kerangka teori berfungsi sebagai pijakan konseptual yang menjelaskan landasan ilmiah serta teori-teori yang mendukung penelitian, termasuk alasan dan mekanisme hubungan antar variabel. Kerangka teori dalam penelitian ini menjelaskan hubungan antara berbagai parameter cuaca, seperti temperatur rata-rata (Tavg), kelembapan relatif (RH_avg), curah hujan sebelumnya (Rainfall Rate/RR), durasi penyinaran matahari (ss), serta kecepatan angin rata-rata (ff_avg) yang telah digunakan secara luas pada penelitian-penelitian sebelumnya (Han *et al.*, 2023; Li *et al.*, 2022; Hendra *et al.*, 2023), sebagai variabel *input* dalam proses pemodelan curah hujan. Sebelum data diolah oleh model, dilakukan tahap pra-pemrosesan yang mencakup pembersihan data (*data cleaning*), normalisasi nilai parameter cuaca ke rentang tertentu (0–1), serta pembentukan urutan data (*data sequence*) agar sesuai dengan kebutuhan arsitektur jaringan LSTM (Han *et al.*, 2023; Li *et al.*, 2022; Badriyah *et al.*, 2022). Proses pemodelan dilakukan menggunakan dua pendekatan utama, yaitu *Long Short-Term Memory* (LSTM) dan *Attention-Based LSTM* (AT-LSTM). LSTM memiliki kemampuan untuk mempelajari ketergantungan jangka panjang (*long-term dependencies*) pada data deret waktu, sementara penerapan mekanisme *Attention* memungkinkan model

memberikan bobot lebih besar pada informasi temporal yang dianggap penting, sehingga fitur yang relevan dapat ditangkap secara lebih efektif (Han *et al.*, 2023; Li *et al.*, 2022). Kerangka teori yang digunakan dalam penelitian ini ditampilkan pada Gambar 2.5:



Gambar 2.5: Kerangka Teori

Kemudian parameter cuaca yang digunakan sebagai variabel dalam prediksi curah hujan ditampilkan pada tabel berikut:

Tabel 2. 2 Variabel Prediksi Curah Hujan

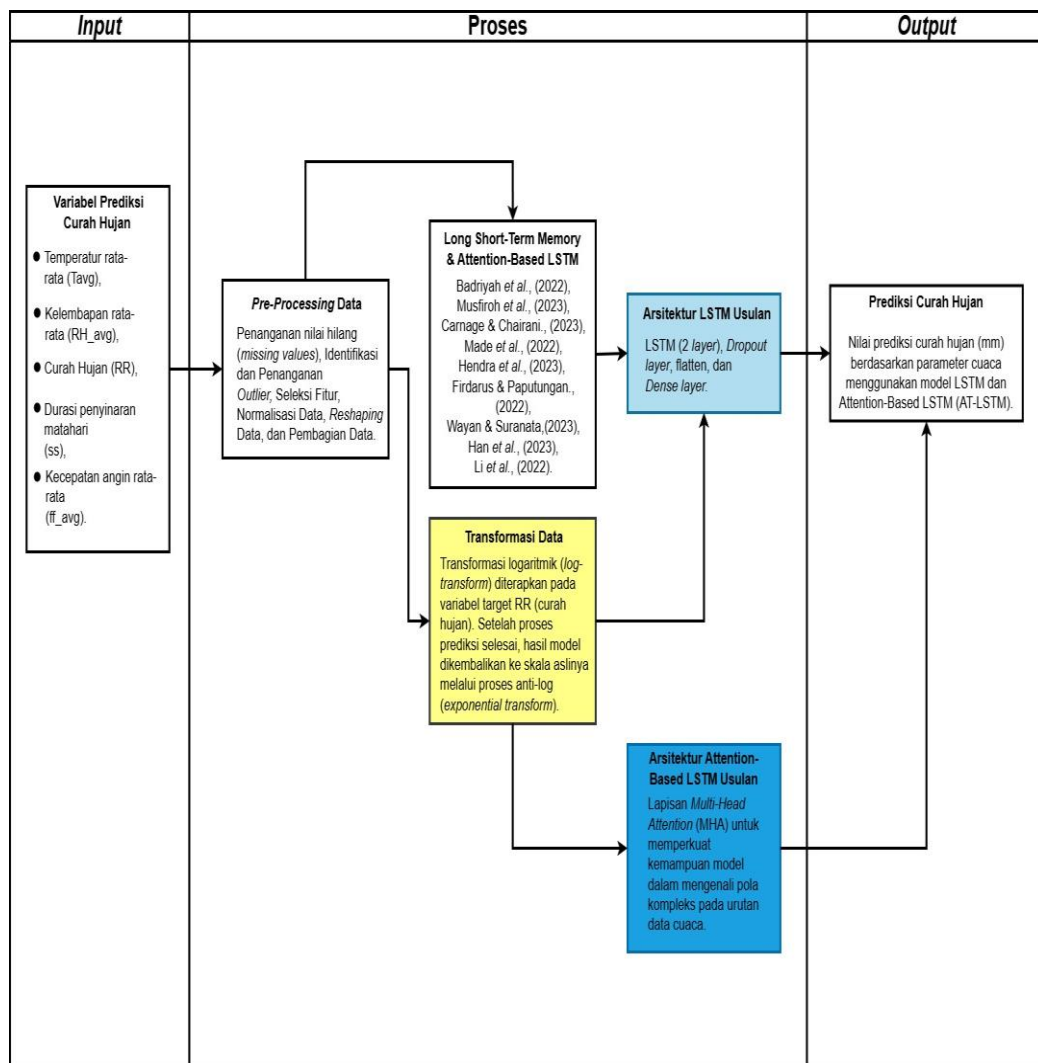
Parameter	Atribut	Satuan
Temperatur rata-rata	Tavg	°C
Kelembapan rata-rata	RH_avg	%
Curah hujan (<i>Rainfall Rate</i>)	RR	mm
Durasi penyinaran matahari	Ss	jam
Kecepatan angin rata-rata	ff_avg	m/s

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Konsep

Kerangka konsep dalam penelitian ini disusun berdasarkan hasil kajian literatur, metode yang digunakan, serta teori-teori relevan yang mendasari penelitian. Kerangka konsep berperan sebagai bentuk operasional dari kerangka teori yang menjadi acuan selama proses penelitian berlangsung. Kerangka konsep tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Kerangka Konsep

Kerangka konsep dalam penelitian ini menggambarkan hubungan antara komponen input, proses, dan output dalam sistem prediksi curah hujan berbasis LSTM dan Attention-Based LSTM (AT-LSTM). Pada bagian input, digunakan lima parameter cuaca utama sebagaimana diadopsi dari Han et al. (2023), yaitu temperatur rata-rata (T_{avg}), kelembapan rata-rata (RH_{avg}), curah hujan atau Rainfall Rate (RR), durasi penyinaran matahari (ss), serta kecepatan angin rata-rata (ff_{avg}) sebagai variabel prediktor.

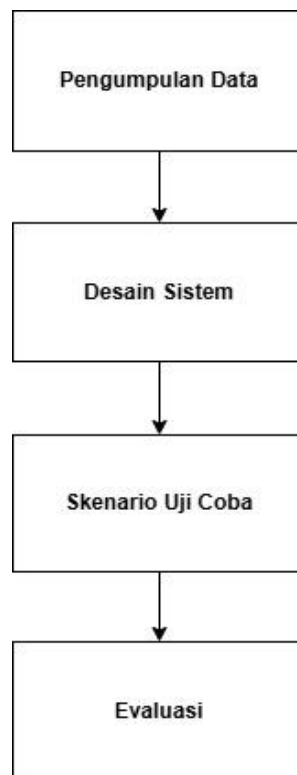
Tahap proses diawali dengan pre-processing data, yang mencakup penanganan missing values, identifikasi dan perbaikan outlier, pemilihan fitur, normalisasi pada fitur (X) dan target (Y) secara terpisah menggunakan MinMaxScaler agar berada dalam rentang 0–1, reshaping data, serta pemisahan data ke dalam subset pelatihan, validasi, dan pengujian. Selain itu, diterapkan transformasi logaritmik (log-transform) pada variabel target RR (curah hujan) untuk menstabilkan varians dan mengurangi pengaruh nilai ekstrem. Setelah proses prediksi selesai, hasil model dikembalikan ke skala aslinya melalui proses anti-log (*exponential transform*) agar interpretasi hasil tetap dalam satuan milimeter (mm). Proses pemodelan kemudian dilakukan dengan dua pendekatan utama, yaitu LSTM dan Attention-Based LSTM (AT-LSTM).

Model LSTM terdiri atas dua *layer* dengan *Dropout layer*, *Flatten layer*, dan *Dense layer* yang berfungsi menghasilkan nilai prediksi curah hujan. Sementara itu, AT-LSTM mengintegrasikan lapisan *Multi-Head Attention* (MHA) di atas arsitektur LSTM untuk memperkuat kemampuan model dalam mengenali pola kompleks dan hubungan temporal yang signifikan pada data cuaca. Tahap *output*

menghasilkan nilai prediksi curah hujan (mm) berdasarkan parameter cuaca yang telah diolah menggunakan model LSTM dan Attention-Based LSTM (AT-LSTM). Hasil prediksi yang dihasilkan dari penelitian ini diharapkan mampu meningkatkan ketepatan estimasi curah hujan serta memberikan dukungan bagi proses pengambilan keputusan di sektor pertanian dan upaya mitigasi risiko iklim, terutama untuk wilayah Kabupaten Malang.

3.2 Desain Prosedur Penelitian

Desain prosedur penelitian disusun untuk memastikan diperolehnya data yang relevan dalam proses penyajian serta interpretasi hasil penelitian. Untuk mempermudah pemahaman alur pelaksanaan penelitian, prosedur tersebut disajikan dalam bentuk diagram pada Gambar 3.2:



Gambar 3. 2 Desain Prosedur Penelitian

3.2.1 Pengumpulan Data

Penelitian ini memfokuskan kajiannya pada prediksi curah hujan di Kabupaten Malang, Jawa Timur, dengan menerapkan metode LSTM dan Attention-Based LSTM. Kabupaten Malang dipilih sebagai lokasi penelitian karena memiliki variabilitas cuaca yang cukup tinggi serta menghadapi dinamika iklim musiman yang berpotensi memengaruhi sektor pertanian dan infrastruktur.

Data penelitian diperoleh dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) melalui Stasiun Klimatologi Kabupaten Malang, yang diakses melalui situs resmi <https://dataonline.bmkg.go.id/data-harian>. Parameter iklim yang digunakan meliputi suhu rata-rata (Tavg), kelembapan relatif rata-rata (RH_avg), curah hujan (RR), durasi penyinaran matahari (ss), dan kecepatan angin rata-rata (ff_avg), sesuai dengan variabel yang direkomendasikan oleh Han et al. (2023).

Secara keseluruhan, dataset berjumlah 8.765 entri harian yang telah melalui proses verifikasi oleh BMKG untuk memastikan validitas dan konsistensinya. Pemilihan data didasarkan pada rentang waktu yang panjang, yakni tahun 2000 hingga 2023, sehingga mampu menangkap pola temporal curah hujan secara komprehensif. Data tersebut disajikan dalam bentuk tabel berikut:

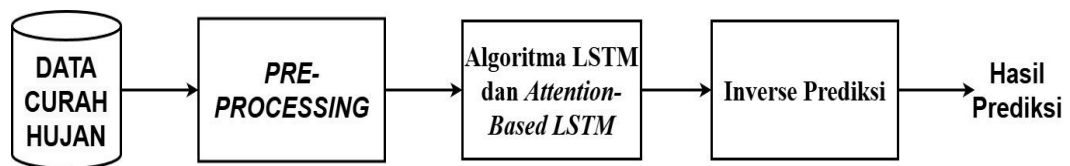
Tabel 3. 1 Data Curah Hujan Harian BMKG

Tanggal	Tavg (°C)	RH_avg (%)	RR (mm)	ss (jam)	ff_avg (m/s)
2000-01-01	24.1	69.0	82.0	5.4	2.0
2000-01-02	22.9	77.0	0.0	1.2	4.0
2000-01-03	23.8	78.0	14.0	4.1	3.0
2000-01-04	24.2	77.0	10.0	1.3	2.0
2000-01-05	23.4	78.0	8888.0	3.6	4.0
...

Lanjutan					
2023-12-27	25.9	78.0	25.3	5.5	1.0
2023-12-28	25.4	86.0	29.3	4.5	1.0
2023-12-29	27.1	79.0	27.5	5.4	2.0
2023-12-30	26.1	82.0	1.4	8.7	2.0

3.2.2 Desain Sistem

Gambar 3.2 berikut menyajikan tahapan desain sistem yang digunakan dalam penelitian ini:



Gambar 3. 3 Desain Sistem

1. Data Curah Hujan

Data yang digunakan dalam penelitian ini merupakan data curah hujan harian dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) Stasiun Klimatologi Kabupaten Malang, Jawa Timur, yang tercatat sejak 1 Januari 2000 hingga 30 Desember 2023. Data tersebut mencakup lima parameter cuaca utama, yaitu suhu rata-rata (Tavg), kelembaban relatif rata-rata (RH_avg), curah hujan (RR), durasi penyinaran matahari (ss), dan kecepatan angin rata-rata (ff_avg). Total data yang diperoleh sebanyak 8.765 entri harian.

2. Pre-Processing

Setelah data yang diperlukan tersedia, langkah berikutnya adalah memproses data tersebut agar siap untuk diterapkan dalam metode yang akan diimplementasikan. Tahapan ini disebut sebagai *pre-processing data*, yaitu proses

pengolahan awal yang dilakukan untuk memastikan kualitas data sebelum digunakan dalam pemodelan. Adapun tahapan *pre-processing* yang dilakukan dalam penelitian ini meliputi beberapa langkah sebagai berikut:

A. Data Masukan

Tahapan pertama melakukan *Input* data kedalam *engine*.

B. Konversi Format Tanggal

Tahapan selanjutnya yaitu melibatkan konversi data pada kolom 'Tanggal' menjadi format *datetime*, yang memudahkan pengelolaan dan analisis data berbasis waktu.

C. Penanganan Data Hilang

Menerapkan normalisasi untuk memastikan bahwa semua parameter berada dalam rentang nilai yang sama, sehingga memudahkan proses pelatihan model.

D. Identifikasi dan Penanganan *Outlier*

Outlier diidentifikasi menggunakan metode *Z-Score* pada kolom semua kolom. *Outlier* yang terdeteksi diganti dengan nilai median untuk meminimalkan dampak distorsi dalam analisis.

E. Seleksi Fitur

Analisis korelasi dilakukan untuk menilai pengaruh setiap fitur terhadap variabel target. Fitur dengan korelasi rendah dihilangkan dari dataset untuk meningkatkan efektivitas model.

F. Normalisasi Data

Data yang telah ditetapkan sebagai fitur dan target kemudian dinormalisasi menggunakan skala Min-Max untuk mengatur ulang rentang nilai dari data,

sehingga semua data memiliki skala yang seragam dan tidak ada data dominan yang dapat mempengaruhi model secara tidak proporsional.

G. *Reshaping Data*

Fitur lag dibuat untuk memanfaatkan informasi historis dalam prediksi, dengan mengintegrasikan data dari periode waktu sebelumnya sebagai bagian dari *input* model dan memetakan data sesuai dengan *input* pada model yang di butuhkan.

H. Pembagian Data

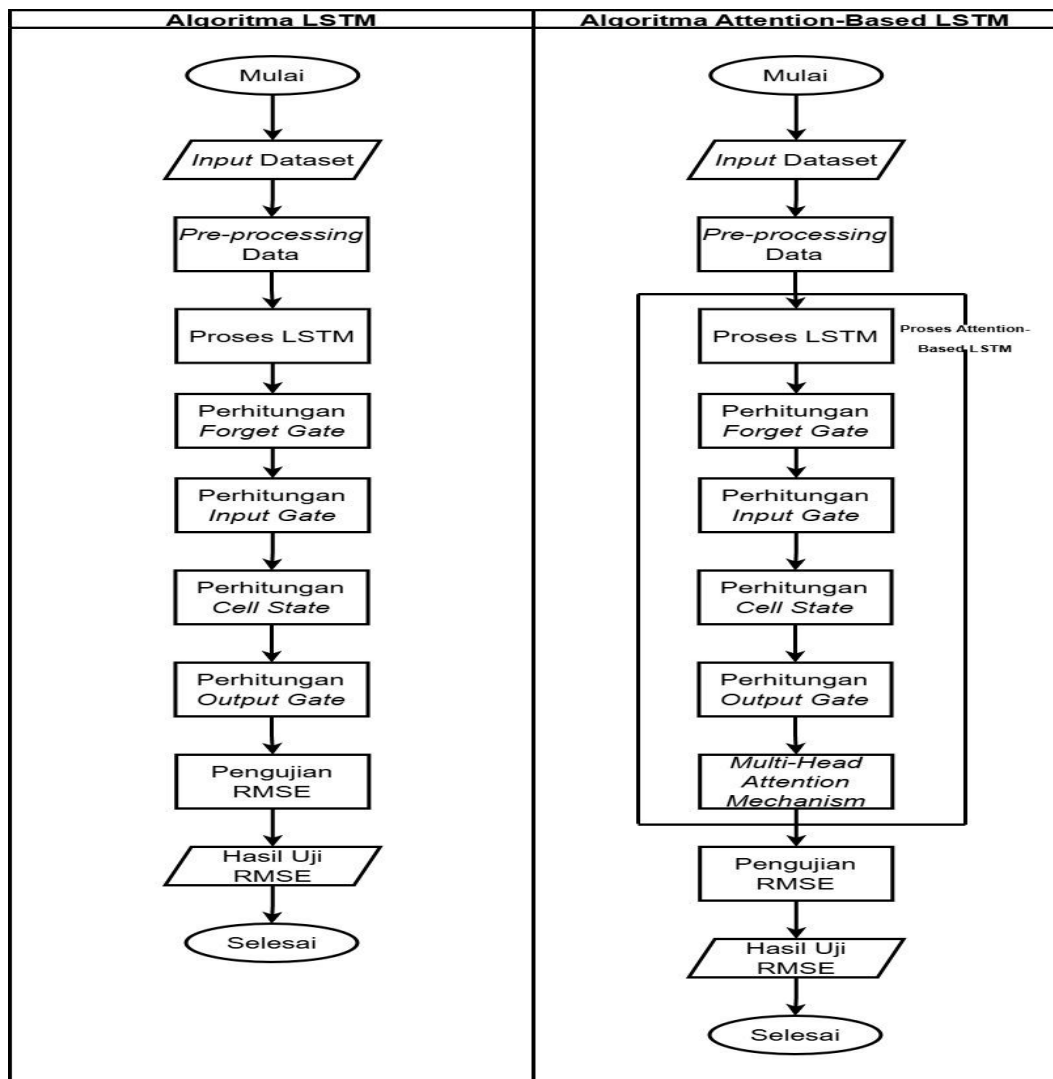
Data dibagi menjadi dua bagian utama, yaitu set pelatihan sebesar 80% dan set pengujian sebesar 20%. Proses pembagian dilakukan secara berurutan (*chronological split*) untuk menjaga struktur waktu pada data, sehingga pola temporal tetap konsisten.

3. Algoritma LSTM dan *Attention-Based LSTM*

Dalam penelitian ini, proses pemodelan prediksi curah hujan dilakukan melalui tiga tahap utama, yaitu input data, pemrosesan model, dan evaluasi. Pada model LSTM, data yang telah melalui tahap pre-processing, meliputi normalisasi dan pembentukan *sequence* diproses secara berurutan melalui mekanisme *forget gate*, *input gate*, *cell state*, dan *output gate*. *Forget gate* menentukan informasi lama yang harus dihilangkan, sedangkan *input gate* memilih informasi baru yang perlu disimpan. Proses ini memperbarui *cell state* sebagai memori utama, dan *output gate* menghasilkan keluaran pada setiap *timestep*, yang kemudian digunakan untuk menghasilkan prediksi curah hujan.

Pada *Attention-Based LSTM*, alur awal bekerja sama seperti LSTM biasa, dimulai dari *input* data hingga proses pembentukan *hidden state* pada setiap

timestep. Perbedaannya terletak pada tahap tambahan berupa *Multi-Head Attention Mechanism*, yang memberikan bobot berbeda pada setiap *hidden state* berdasarkan relevansinya terhadap prediksi akhir. *Timestep* yang lebih penting memperoleh bobot lebih besar sehingga model dapat fokus pada pola cuaca yang signifikan. Hasil mekanisme ini berupa *context vector* yang memuat representasi informasi paling relevan dari seluruh urutan data dan digunakan sebagai dasar prediksi. Evaluasi akhir dilakukan menggunakan RMSE untuk mengukur selisih antara nilai prediksi dan nilai aktual.



Gambar 3. 4 Algoritma LSTM dan *Attention-Based LSTM*

4. Invers Prediksi

Inverse prediksi merupakan proses untuk mengembalikan hasil prediksi model dari skala transformasi (log dan normalisasi) ke skala asli curah hujan (dalam milimeter). Dalam penelitian ini, data curah hujan awal terlebih dahulu dikenai transformasi logaritmik menggunakan fungsi \log_{1p} , untuk menstabilkan varians dan mengurangi pengaruh nilai ekstrem. Selanjutnya, hasil transformasi logaritmik tersebut dinormalisasi menggunakan metode *Min-Max Scaling* sehingga seluruh nilai berada dalam rentang 0–1. Setelah model selesai dilatih dan menghasilkan prediksi dalam skala yang sama (log + normalisasi), dilakukan proses *inverse transform* secara berurutan untuk mengembalikan ke satuan asli. Pertama, dilakukan *inverse normalization* dengan rumus $Y_{log} = Y_{norm} \times (Y_{logmax} - Y_{logmin}) + Y_{logmin}$ menggunakan fungsi `scaler_Y.inverse_transform`. Kedua, dilakukan *anti-log transform* untuk membalik hasil logaritmik dengan fungsi `np.expm1`, yang secara matematis dinyatakan sebagai $Y = e^{Y_{log}} - 1$. Dengan demikian, hasil akhir dari proses inverse prediksi (`y_pred_antilog`) dan data uji (`y_test_antilog`) kembali merepresentasikan curah hujan dalam skala asli (mm), sehingga dapat digunakan untuk evaluasi performa model pada konteks nyata.

3.2.3 Skenario Uji Coba

Setelah model melalui semua proses dalam desain sistem, maka selanjutnya adalah skenario uji coba dengan LSTM dan Attention-Based LSTM, maka model akan dilatih dengan beberapa proses yaitu:

a. Proses Pelatihan Model

Pelatihan model pertama, yaitu data dipisahkan menjadi dua bagian, di mana 80% dari total data digunakan sebagai data pelatihan (*training*) dan 20% sisanya dialokasikan sebagai data pengujian (*testing*), kemudian *validation split 10* pada data pelatihan untuk memastikan kemampuan performa model secara lebih akurat (Wayan & Suranata 2023). Kemudian fungsi aktivasi (*tanh* dan *ReLU*), ukuran *batch* (4, 16, 32, 64, dan 128), serta jumlah *epochs* (10, 20, 50, dan 100) akan ditetapkan sebagai parameter eksperimen. Model yang dihasilkan akan dilatih menggunakan *training data* (X_{train} dan y_{train}) serta divalidasi dengan *validation data* (X_{val} dan y_{val}). Sebuah *callback ModelCheckpoint* akan diterapkan untuk menyimpan model terbaik berdasarkan metrik validasi (*val_loss*).

b. Visualisasi dan Penyimpanan Model Terbaik

Setelah proses pelatihan selesai, hasil dari setiap konfigurasi divisualisasikan dalam bentuk grafik deret waktu yang membandingkan data curah hujan aktual dan hasil prediksi. Grafik ini memudahkan analisis perbandingan performa model pada berbagai tahap pelatihan, untuk evaluasi lebih lanjut dan pemilihan model terbaik.

3.2.4 Evaluasi

Setelah proses pelatihan selesai, model dievaluasi pada data pelatihan dan data validasi menggunakan metrik *Root Mean Squared Error* (RMSE) menjadi lebih objektif dan mengurangi potensi bias dari pembagian data tertentu. Dengan demikian, seluruh proses pelatihan ini bertujuan untuk mengidentifikasi parameter optimal yang meningkatkan kinerja model *neural network* secara keseluruhan.

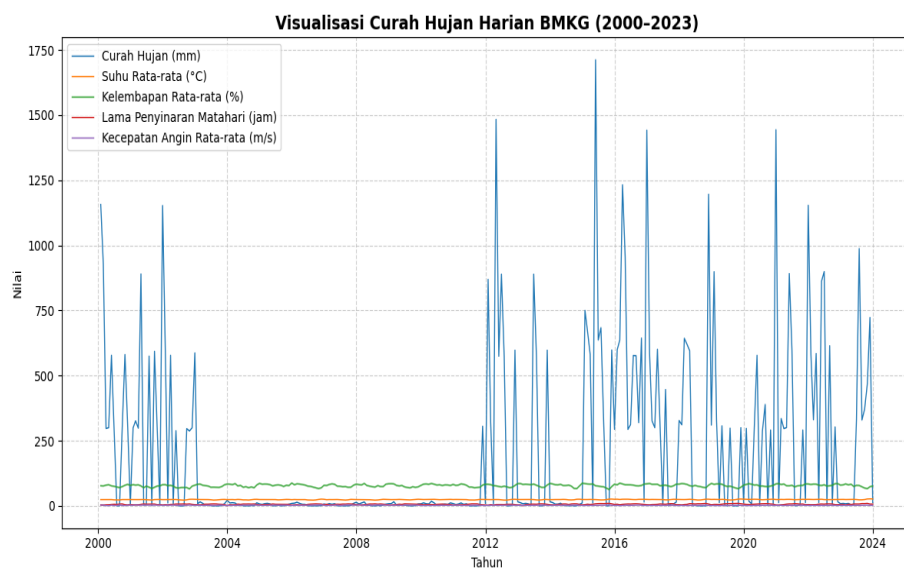
BAB IV

PREDIKSI CURAH HUJAN DENGAN LSTM

4.1 Deskripsi Data Curah Hujan Harian BMKG

Data yang digunakan dalam penelitian ini berupa deret waktu (time series) curah hujan harian yang diperoleh dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) melalui Stasiun Klimatologi Kabupaten Malang, Jawa Timur. Rentang data meliputi periode 1 Januari 2000 hingga 30 Desember 2023, dengan total 8.765 catatan harian.

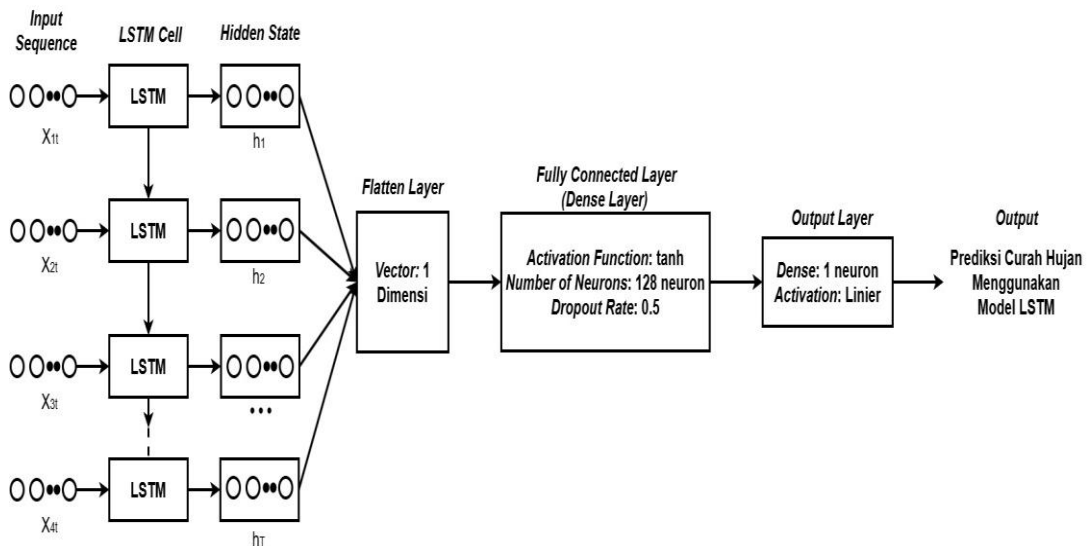
Dataset ini tidak hanya berisi data curah hujan (RR), tetapi juga mencakup berbagai variabel iklim pendukung lainnya, seperti suhu rata-rata (Tavg), kelembapan relatif rata-rata (RH_avg), lama penyinaran matahari (ss), serta kecepatan angin rata-rata (ff_avg). Ringkasan statistik dari data tersebut disajikan pada Gambar 4.2.



Gambar 4. 1 Curah Hujan Harian BMKG

4.2 Arsitektur Model LSTM

Arsitektur Model LSTM yang diterapkan dalam penelitian ini terdiri atas beberapa lapisan utama, seperti yang ditunjukkan pada Gambar 4.2. *input* ke model adalah sekumpulan data sekuensial (*time series*), yang diwakili oleh x . Setiap urutan data memiliki panjang T (*timesteps*), dan pada setiap *timestep* t , inputnya adalah vektor fitur x_t . Vektor fitur ini terdiri dari beberapa variabel, yaitu: x_{1t} suhu rata-rata (Tavg), x_{2t} kelembaban relatif rata-rata (RH_avg), x_{3t} = durasi penyinaran matahari (ss), dan x_{4t} kecepatan angin rata-rata (ff_avg).



Gambar 4. 2 Arsitektur Model LSTM

Input tersebut diproses melalui sel-sel LSTM yang menghasilkan *hidden state* (h_1, h_2, \dots, h_T) pada setiap langkah waktu. Setiap *hidden state* ini adalah representasi ringkas dari informasi yang telah dilihat oleh jaringan hingga langkah waktu tersebut, yang berfungsi sebagai "memori" internal jaringan. *Hidden state* yang dihasilkan kemudian digabungkan dan diratakan melalui flatten layer menjadi sebuah vektor satu dimensi agar lebih mudah diproses pada lapisan selanjutnya. Vektor tersebut kemudian diteruskan ke lapisan *fully connected (dense layer)* yang

terdiri dari 128 neuron dengan fungsi aktivasi tanh, serta dilengkapi *dropout* sebesar 0,5 untuk mengurangi risiko *overfitting*. Pada bagian akhir, terdapat *dense layer* dengan satu *neuron* dan aktivasi linear, yang berfungsi menghasilkan *output* berupa prediksi nilai curah hujan harian. Dengan rancangan ini, model LSTM mampu menangkap pola ketergantungan temporal dalam data historis, kemudian menerjemahkannya menjadi estimasi kuantitatif terhadap intensitas curah hujan pada hari berikutnya.

4.3 Pre-processing Data

Praproses data merupakan langkah krusial dalam analisis untuk memastikan data yang digunakan memiliki kualitas dan konsistensi yang baik sebelum dimasukkan ke dalam model prediksi. Dalam penelitian ini, data yang diolah mencakup variabel tanggal, suhu rata-rata harian (Tavg), kelembapan relatif harian (RH_avg), curah hujan harian (RR), lama penyinaran matahari harian (ss), serta kecepatan angin rata-rata (ff_avg), dengan rentang pengamatan mulai tahun 2000 hingga 2023.

Sebelum tahap praproses utama dilakukan, terdapat beberapa langkah awal yang bersifat teknis, yaitu *formatting* tanggal, yang dilakukan dengan mengubah data tanggal ke dalam format deret waktu (*time series*) agar kompatibel dengan model LSTM, serta *reshaping* data, yaitu menata ulang struktur data menjadi urutan (*sequence*) yang dapat dikenali oleh model. Selain itu, dilakukan pula pengecekan terhadap adanya nilai hilang, mencakup proses identifikasi, pencarian, dan pengisian data yang kosong. Tahapan ini penting agar tidak ada ketidaksesuaian

yang dapat memengaruhi kinerja model, meskipun tidak dikategorikan sebagai bagian utama dari praproses.

4.3.1 Menangani *Outlier*

Pada tahap selanjutnya dari proses *pre-processing* data, penelitian ini melakukan penanganan terhadap nilai *outlier*. *Outlier* merupakan nilai yang berada jauh dari kecenderungan umum data atau menyimpang secara signifikan dari pola utama dalam dataset, sehingga dapat mengganggu distribusi, memengaruhi hasil normalisasi, dan menurunkan akurasi model prediksi. Untuk mendeteksi dan menggantikan *outlier*, penelitian ini menggunakan metode Z-Score. Z-Score dihitung untuk setiap nilai dalam kolom terpilih dengan rumus:

$$Z = \frac{X - \mu}{\sigma} \quad (4.1)$$

Keterangan:

X = nilai data

μ = rata-rata (mean) kolom

σ = standar deviasi kolom

Kriteria $|Z| > 3$ digunakan karena berdasarkan aturan empiris (*empirical rule*), sekitar 99,7% data yang berdistribusi normal berada dalam rentang ± 3 standar deviasi dari rata-rata. Dengan kata lain, hanya sekitar 0,3% dari keseluruhan data yang berada di luar batas tersebut sehingga dapat dikategorikan sebagai *outlier*. Dengan demikian, kriteria $|Z| > 3$ menjadi standar umum dalam deteksi *outlier* berbasis distribusi normal, karena nilai di luar rentang tersebut memiliki probabilitas kemunculan yang sangat kecil (Yaro *et al.*, 2023).

Dalam penelitian ini, nilai-nilai *outlier* yang terdeteksi kemudian digantikan dengan median dari kolom yang bersangkutan. Pemilihan median didasarkan pada sifatnya yang lebih robust dibandingkan mean, karena tidak terpengaruh oleh nilai-nilai ekstrem. Kolom-kolom yang diperiksa untuk outlier mencakup Tavg (°C), RH_avg (%), ss (jam), ff_avg (m/s), dan RR (mm). Berikut data setelah penanganan pada Tabel 4.1:

Tabel 4. 1 Hasil Data Setelah Penanganan *Outlier*

	Tanggal	Tavg (°C)	RH_avg (%)	RR (mm)	ss (jam)	ff_avg (m/s)
0	2000-01-01	24.1	69.0	82.0	5.4	2.0
1	2000-01-02	22.9	77.0	0.0	1.2	4.0
2	2000-01-03	23.8	78.0	14.0	4.1	3.0
3	2000-01-04	24.2	77.0	10.0	1.3	2.0
4	2000-01-05	23.4	78.0	0.0	3.6	4.0
...
8764	2023-12-31	26.6	81.0	34.0	4.0	2.0

4.3.2 *Selection* Fitur

Setelah melakukan penanganan *outlier* pada data dan memastikan data sudah bersih baik dari data hilang dan *outlier*, langkah berikutnya adalah menghitung korelasi antara setiap variabel X, yakni kolom 'Tavg (°C)', 'RH_avg (%)', 'ss (jam)', dan 'ff_avg (m/s)', dengan variabel Y pada kolom 'RR (mm)'. Proses ini bertujuan untuk menilai sejauh mana hubungan linier antara dua variabel tersebut. Metode yang digunakan dalam tahap seleksi ini adalah korelasi Pearson, yang berfungsi untuk mengukur kekuatan dan arah hubungan linier antarvariabel. Rumus koefisien korelasi Pearson ditampilkan pada persamaan (4.2):

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \cdot \sum(y_i - \bar{y})^2}} \quad (4.2)$$

Keterangan:

r_{xy} = koefisien korelasi Pearson antara variabel X dan Y

x_i, y_i = nilai data ke-i pada variabel X dan Y

\bar{x}, \bar{y} = nilai rata-rata dari variabel X dan Y

Koefisien r memiliki rentang nilai dari -1 hingga $+1$. Nilai yang mendekati $+1$ menandakan adanya hubungan linier positif yang kuat, sedangkan nilai yang mendekati -1 menunjukkan hubungan linier negatif yang kuat. Sementara itu, nilai yang berada di sekitar 0 mengindikasikan bahwa tidak terdapat hubungan linier yang berarti antara kedua variabel.

Untuk menguji signifikansi korelasi dengan uji transformasi Fisher (Z-test):

$$Z_{hitung} = \frac{\sqrt{n-3}}{2} \cdot \ln\left(\frac{1+r}{1-r}\right) \quad (4.3)$$

p -value dihitung dengan persamaan:

$$p = 2 \cdot (1 - \Phi(|Z_{hitung}|)) \quad (4.4)$$

Tabel 4.2 menyajikan hasil korelasi antara masing-masing variabel prediktor dengan target:

Tabel 4. 2 Korelasi Antar Atribut

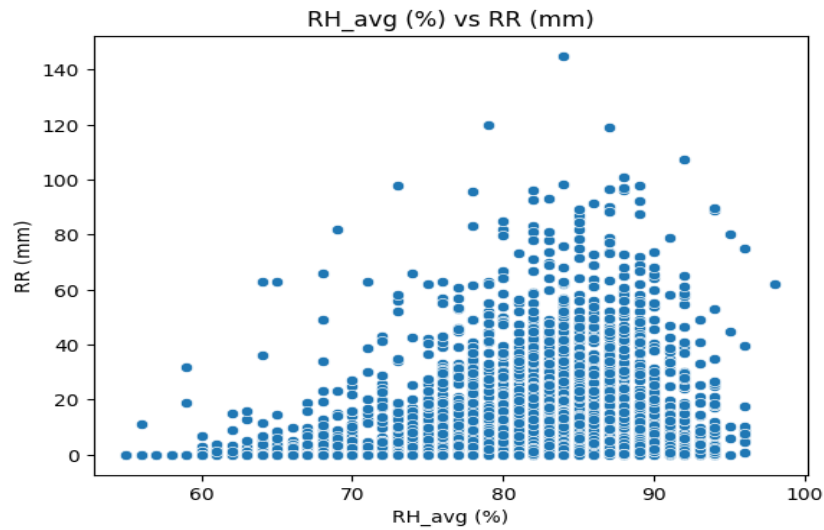
Seleksi Fitur	r (Korelasi)	Z_hitung	Two-tailed ($\alpha=0.05$)	Signifikansi
RH_avg (%) vs RR (mm)	0.326558	31.7295	< 0.0001	Signifikan (+)
Tavg (°C) vs RR (mm)	0.040954	3.8357	0.00013	Signifikan (+)
ss (jam) vs RR (mm)	- 0.257047	- 24.6130	< 0.0001	Signifikan (-)
ff_avg (m/s) vs RR (mm)	- 0.181402	- 17.1703	< 0.0001	Signifikan (-)

Merujuk pada Tabel 4.2, seluruh variabel prediktor menunjukkan hubungan yang signifikan secara statistik dengan curah hujan (RR) pada tingkat signifikansi 5% (*two-tailed*). Hal ini berarti setiap variabel, baik dengan korelasi positif maupun negatif, memberikan pengaruh terhadap variasi curah hujan.

Variabel RH_avg (%) menunjukkan korelasi positif tertinggi ($r = 0.326558$), yang berarti peningkatan kelembaban relatif berasosiasi dengan meningkatnya curah hujan. Variabel Tavg (°C) juga memiliki korelasi positif ($r = 0.040954$), meskipun nilainya kecil, tetap dipertahankan karena suhu rata-rata merupakan parameter klimatologis penting dalam proses pembentukan hujan. Variabel ss (jam) (durasi penyinaran matahari) memiliki korelasi negatif ($r = -0.257047$), yang menunjukkan bahwa semakin lama penyinaran, curah hujan cenderung menurun. Variabel ff_avg (m/s) (kecepatan angin rata-rata) juga memiliki korelasi negatif ($r = -0.181402$), artinya peningkatan kecepatan angin berasosiasi dengan penurunan curah hujan.

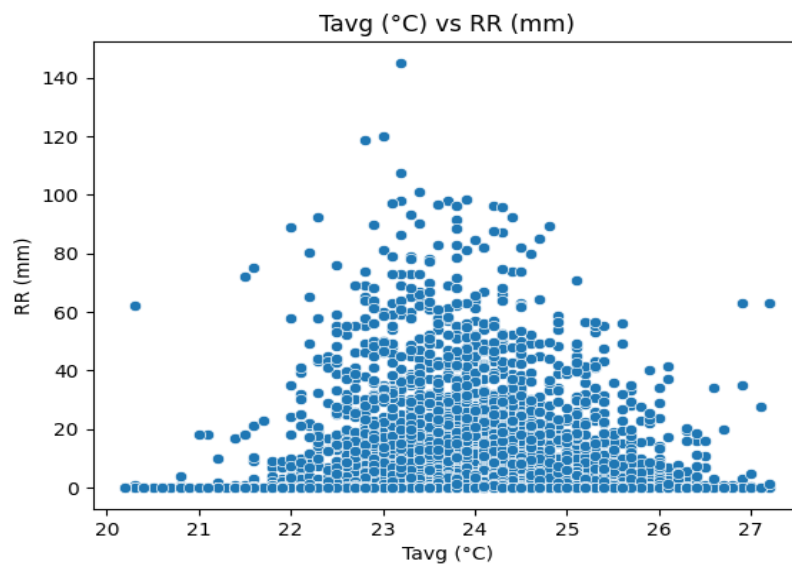
Dengan demikian, keempat variabel tetap dipertahankan dalam pemodelan karena semuanya signifikan. Tanda korelasi (+ atau -) mencerminkan arah pengaruh variabel terhadap curah hujan, sementara besaran nilai r menggambarkan sejauh mana variabel tersebut berperan dalam menggambarkan variasi yang terjadi pada curah hujan.

Variabel RH_avg (kelembapan relatif rata-rata) menunjukkan nilai korelasi positif paling tinggi terhadap curah hujan, yaitu sebesar 0.326558. Hal ini menunjukkan bahwa semakin tinggi kelembaban relatif, curah hujan cenderung semakin meningkat.



Gambar 4. 3 RH_avg (%) vs RR (mm)

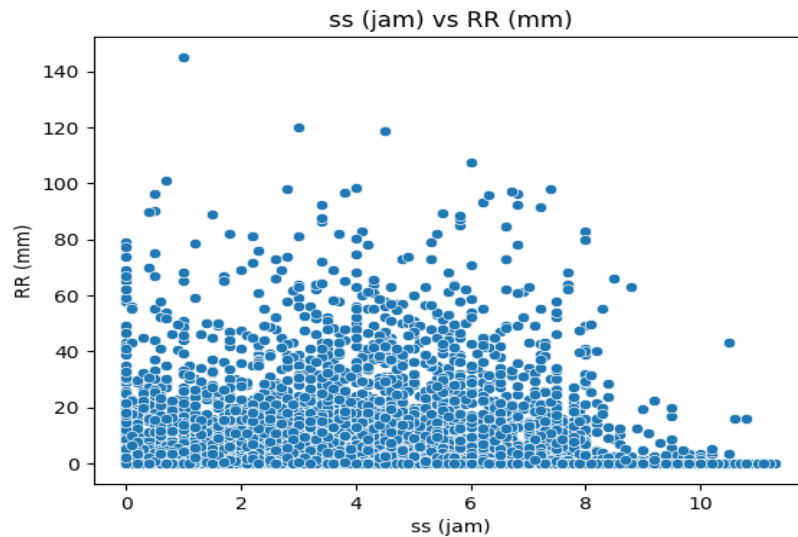
Variabel Tavg (suhu rata-rata) juga memiliki hubungan positif dengan curah hujan dengan nilai korelasi 0.040954. Meskipun nilainya kecil, suhu rata-rata tetap dipertahankan karena merupakan faktor klimatologis yang penting dalam proses pembentukan hujan.



Gambar 4. 4 Tavg (°C) vs RR (mm)

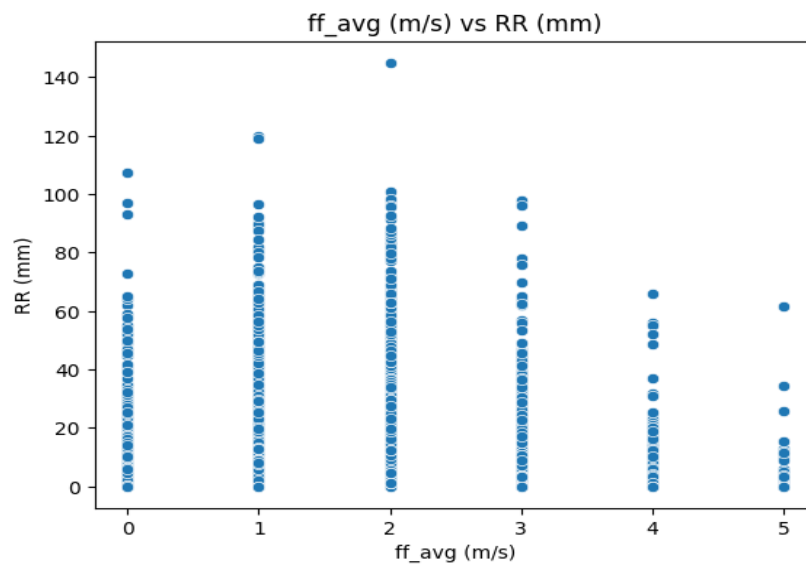
Sementara itu, variabel ss (durasi penyinaran matahari) memiliki koefisien korelasi -0.257047 , yang mengindikasikan hubungan negatif dengan curah hujan.

Artinya, semakin lama durasi penyinaran matahari, curah hujan cenderung menurun.



Gambar 4. 5 ss (jam) vs RR (mm)

Sementara itu, variabel ff_avg (kecepatan angin rata-rata) memiliki korelasi negatif sebesar -0.181402 , yang mengindikasikan bahwa peningkatan kecepatan angin cenderung diikuti oleh penurunan curah hujan.



Gambar 4. 6 $ff_avg(m/s)$ vs RR (mm)

Berdasarkan hasil perhitungan korelasi dan visualisasi scatter plot, keempat variabel terbukti signifikan secara statistik ($p < 0.05$). Oleh karena itu, seluruh variabel RH_avg, Tavg, ss, dan ff_avg tetap digunakan dalam pemodelan. Tanda korelasi (+ atau -) mencerminkan arah pengaruh masing-masing variabel terhadap curah hujan, sedangkan besaran nilai r menggambarkan kontribusinya dalam menjelaskan variasi curah hujan.

4.3.3 Normalisasi Data

Sebelum masuk ke tahap normalisasi, terlebih dahulu dilakukan pemuatan data yang sudah melalui tahap pembersihan. Normalisasi adalah tahap yang krusial untuk menyeragamkan skala setiap variabel, sehingga tidak ada fitur dengan rentang nilai besar yang dapat mendominasi proses pelatihan model. Dalam penelitian ini digunakan metode *Min-Max Normalization* dengan rentang skala [0, 1]. Rumus normalisasi disajikan pada Persamaan 4.2 dan 4.3.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4.5)$$

Dengan normalisasi, seluruh fitur berada pada rentang yang sama sehingga model dapat belajar secara lebih seimbang. Implementasi normalisasi dilakukan menggunakan fungsi *MinMaxScaler* dari pustaka *sklearn.preprocessing*. Kolom yang dinormalisasi meliputi variabel Tavg (°C), RH_avg (%), RR (mm), ss (jam), dan ff_avg (m/s). Selanjutnya, pada variabel target (Y) yaitu curah hujan (RR), dilakukan dua tahap transformasi. Tahap pertama adalah transformasi logaritmik alami untuk meredam variasi ekstrem pada data curah hujan yang sering mengandung nilai lonjakan tinggi (*outlier*). Rumus yang digunakan adalah:

$$Y_{log} = \ln(1 + Y) \quad (4.6)$$

Penerapan transformasi logaritmik berfungsi untuk menstabilkan varians dan menjadikan distribusi data lebih mendekati normal, sehingga pola dalam data dapat lebih mudah dipelajari oleh model. Pada tahap berikutnya, nilai yang telah melalui *log-transform* dinormalisasi menggunakan metode *Min-Max* agar seluruh data berada dalam kisaran $[0,1]$, dengan rumus sebagai berikut:

$$Y_{norm} = \frac{Y_{log} - Y_{logmin}}{Y_{logmax} - Y_{logmin}} \quad (4.7)$$

Normalisasi variabel target (RR) sangat penting agar prediksi model berada dalam rentang yang sama dengan target, dan setelah prediksi selesai, nilai-nilai tersebut akan diubah kembali ke skala aslinya (*denormalization*) untuk evaluasi. Berikut hasil dari tabel normalisasi data:

Tabel 4. 3 Hasil Normalisasi Data

Tavg (°C)	RH_avg (%)	ss (jam)	ff_avg (m/s)	RR (mm)	RR_log	RR_norm
0.557143	0.325581	0.477876	0.4	82.0	4.418841	0.886675
0.385714	0.511628	0.106195	0.8	0.0	0.000000	0.000000
0.514286	0.534884	0.362832	0.6	14.0	2.708050	0.543392
0.571429	0.511628	0.115044	0.4	10.0	2.397895	0.481157
0.457143	0.534884	0.318584	0.8	0.0	0.000000	0.000000
0.542857	0.581395	0.407080	0.6	2.0	1.098612	0.220445
0.385714	0.581395	0.265487	0.6	4.0	1.609438	0.322946

Dengan demikian, data hasil normalisasi sudah siap digunakan pada tahap pemodelan, di mana setiap fitur memiliki skala yang sebanding.

4.3.4 Pembagian Data

Pada tahap ini, data dibagi menjadi dua bagian utama, yaitu data pelatihan dan data pengujian. Sebanyak 80% data dialokasikan sebagai set pelatihan

(*training*), sedangkan 20% sisanya digunakan sebagai set pengujian (*testing*). Pembagian dilakukan menggunakan fungsi `train_test_split` tanpa pengacakan (*shuffle=False*). Proses ini menghasilkan `X_train` dan `y_train` untuk proses pelatihan model, serta `X_val` dan `y_val` yang diperoleh melalui *validation split* sebesar 10% dari data pelatihan. Selain itu, `X_test` dan `y_test` digunakan sebagai data untuk evaluasi akhir model setelah proses pelatihan selesai (Wayan & Suranata, 2023). Adapun hasil pembagian data adalah sebagai berikut:

Tabel 4.4 Hasil Pembagian Data

<i>Training Data</i>	<i>Testing Data</i>
7006	1752

4.4 Hyperparameter LSTM

Pada tahap ini, model LSTM dikonfigurasi menggunakan serangkaian *hyperparameter* yang berfungsi sebagai instruksi yang mengontrol bagaimana model belajar dan beradaptasi. *Batch size* merupakan jumlah sampel data yang diproses model dalam satu kali pembaruan bobot. Penelitian ini menguji lima variasi batch size (4, 16, 32, 64, dan 128) untuk melihat pengaruhnya terhadap stabilitas pelatihan dan kecepatan konvergensi (Nayak *et al.*, 2023). *Epoch* merupakan jumlah berapa kali seluruh dataset dilatih secara penuh. Empat variasi *epoch* (10, 20, 50, dan 100) digunakan untuk mengamati performa model dan mendeteksi *overfitting* (Li *et al.*, 2021).

Optimizer yang digunakan adalah Adam karena mampu menyesuaikan *learning rate* secara adaptif dan menunjukkan kinerja yang baik pada berbagai permasalahan *deep learning*. Fungsi aktivasi yang dipilih adalah tanh, yang membantu

menstabilkan nilai keluaran pada rentang -1 hingga 1 sehingga proses pelatihan lebih cepat dan stabil (Farikhul Firdaus & Paputungan, 2022). *Dropout* sebesar 0.5 digunakan sebagai teknik regularisasi untuk mengurangi risiko *overfitting* pada *hidden layer*, sebagaimana direkomendasikan oleh penelitian terdahulu (Salehin *et al.*, 2023). *Learning rate* ditetapkan pada nilai 0.001 , yang merupakan nilai umum dan terbukti efektif dalam mempercepat proses pelatihan (Hendra *et al.*, 2023). Untuk mengevaluasi tingkat kesalahan prediksi, penelitian ini menggunakan fungsi loss *Mean Squared Error* (MSE), yang menghitung rata-rata kuadrat perbedaan antara nilai prediksi dan nilai sebenarnya (Maknunah *et al.*, 2022). Adapun parameter yang digunakan dalam eksperimen ini meliputi:

Tabel 4. 5 *Hyperparameter* Model LSTM

<i>Hyperparameter yang digunakan</i>	<i>Nilai / Keterangan</i>
<i>Batch size</i>	4, 16, 32, 64, 128
<i>Epochs</i>	10, 20, 50, 100
<i>Optimizer</i>	Adam
Fungsi Aktivasi	tanh
<i>Dropout Rate</i>	0.5
<i>Learning Rate</i>	$1e-3$
<i>Loss Function</i>	<i>Mean Square Error</i>

Pemilihan 128 *neuron* dilakukan berdasarkan dua pendekatan. Pertama, secara teoretis ukuran 128 merupakan unit yang umum digunakan untuk model LSTM karena mampu menyeimbangkan kapasitas representasi dan risiko *overfitting*. Kedua, dalam penelitian ini melakukan eksperimen dengan beberapa konfigurasi (64, 128, dan 256 unit). Hasil pengujian menunjukkan bahwa 128 *neuron* menghasilkan nilai *error* paling rendah (RMSE 0.2076), sehingga dipilih sebagai

konfigurasi terbaik dalam prediksi curah hujan. Berikut tabel 4.6 hasil eksperimen jumlah *neuron* pada LSTM:

Tabel 4. 6 Hasil Eksperimen Jumlah *Neuron* pada LSTM

<i>Neuron</i>	<i>Batch Size</i>	<i>Epoch</i>	RMSE
64	16	100	0.2117
128	16	100	0.2076
256	16	100	0.3302

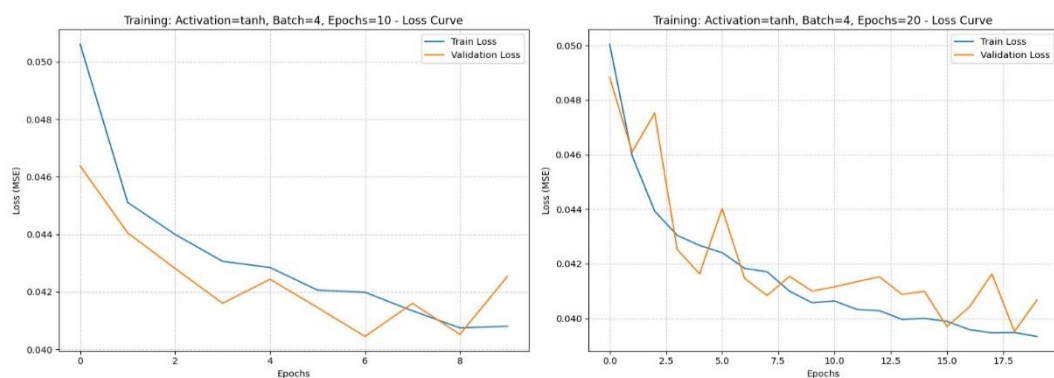
4.5 Training Model LSTM

Pada tahap ini, model mulai dilatih menggunakan parameter yang telah ditentukan pada proses perancangan. Pelatihan dilakukan dengan memanfaatkan algoritma optimasi Adam serta fungsi aktivasi tanh. Beragam kombinasi *hyperparameter* diuji, termasuk variasi *batch size* 4, 16, 32, 64, dan 128 serta jumlah *epochs* 10, 20, 50, dan 100. Data dibagi menjadi dua kelompok, yaitu 80% sebagai *training set* dan 20% sebagai *testing set*. Proses pembagian menggunakan fungsi `train_test_split` tanpa pengacakan (*shuffle=False*) agar urutan kronologis data tetap terjaga, mengingat penelitian ini menggunakan data deret waktu (*time series*). Validasi model tidak menggunakan *validation set* terpisah, melainkan melalui *validation split* sebesar 10% pada data pelatihan untuk meningkatkan kemampuan generalisasi model (Wayan & Suranata, 2023). Selama tahap pelatihan, performa model dipantau melalui nilai *loss function*, yang menjadi tolok ukur untuk mengetahui tingkat perbedaan antara hasil prediksi dan nilai aktual. Tujuan utama dalam pelatihan model adalah untuk meminimalkan nilai *loss*.

Kemampuan ini memungkinkan model untuk secara efisien menemukan bobot optimal yang meminimalkan kesalahan (Elharrouss *et al.*, 2025).

Berikut ini adalah nilai *Loss* (MSE) dari setiap konfigurasi *epochs* yang disajikan dalam bentuk visualisasi. Penyajian grafik ini menampilkan sampel plot dari beberapa *batch* dan *epochs* yang berbeda, sementara tabel metrik *training* LSTM lengkap dapat dilihat pada Lampiran 1.

Pada grafik hasil pelatihan model LSTM, terlihat bahwa baik pada konfigurasi *Batch 4–Epoch 10* maupun *Batch 4–Epoch 20*, nilai *train loss* dan *validation loss* sama-sama mengalami penurunan sehingga menunjukkan proses pembelajaran berjalan efektif. Pada Epoch 10, penurunan *loss* terlihat stabil dengan jarak kecil antara *train* dan *validation loss*, menandakan model belum mengalami *overfitting*. Sementara itu, pada Epoch 20, proses penurunan *loss* berlangsung lebih lama dan menghasilkan MSE yang lebih rendah, meskipun kurva *validation loss* tampak sedikit lebih fluktuatif sehingga mulai menunjukkan indikasi mendekati *overfitting*. Secara keseluruhan, kedua konfigurasi berhasil menurunkan MSE dengan baik, namun peningkatan *epoch* memberikan akurasi lebih tinggi dengan sedikit peningkatan variabilitas pada *validation loss*.



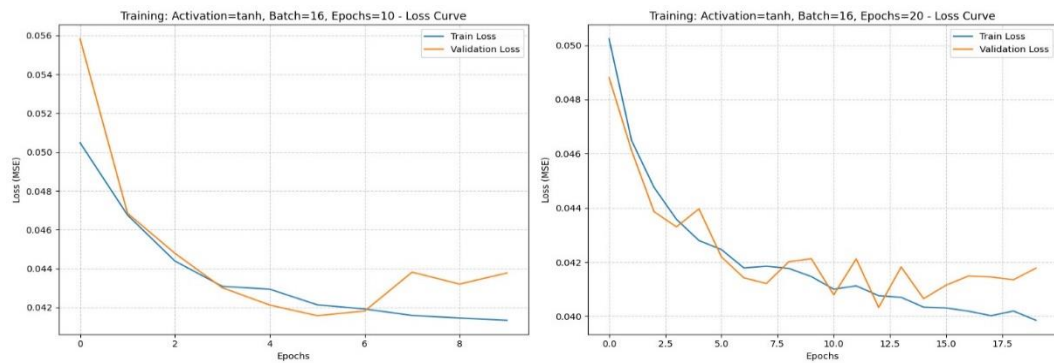
Gambar 4. 7 Grafik *Training* Model LSTM dengan variasi *Batch 4–Epoch 10*, dan *Batch 4–Epoch 20*.

Pada konfigurasi *Batch 4–Epoch 50*, nilai *train loss* dan *validation loss* sama-sama menurun secara stabil hingga berada pada kisaran nilai yang relatif rendah sehingga belum menunjukkan tanda *overfitting* yang kuat. Sebaliknya, pada konfigurasi *Batch 4–Epoch 100*, *train loss* terus menurun secara signifikan hingga sangat kecil, namun nilai *validation loss* mulai mengalami kenaikan setelah memasuki pertengahan proses pelatihan. Pola tersebut menandakan terjadinya *overfitting*, di mana model terlalu mengikuti pola pada data pelatihan sehingga kinerjanya pada data validasi menurun. Oleh karena itu, penambahan jumlah epoch hingga 100 tidak memberikan peningkatan kualitas model, dan konfigurasi *Epoch 50* lebih stabil dan lebih representatif terhadap kemampuan generalisasi model.



Gambar 4. 8 Grafik *Training Model LSTM* dengan variasi *Batch 4–Epoch 50*, dan *Batch 4–Epoch 100*.

Pada konfigurasi *Batch 16–Epoch 10*, nilai *train loss* dan *validation loss* sama-sama menurun secara stabil hingga mulai sedikit berfluktuasi namun tetap berada dekat dengan *train loss*, dan pada konfigurasi *Batch 16–Epoch 20*, pola penurunan *train loss* dan *validation loss* terlihat lebih halus dan konsisten di awal, meskipun *validation loss* menunjukkan sedikit fluktuasi setelah pertengahan *epoch*.



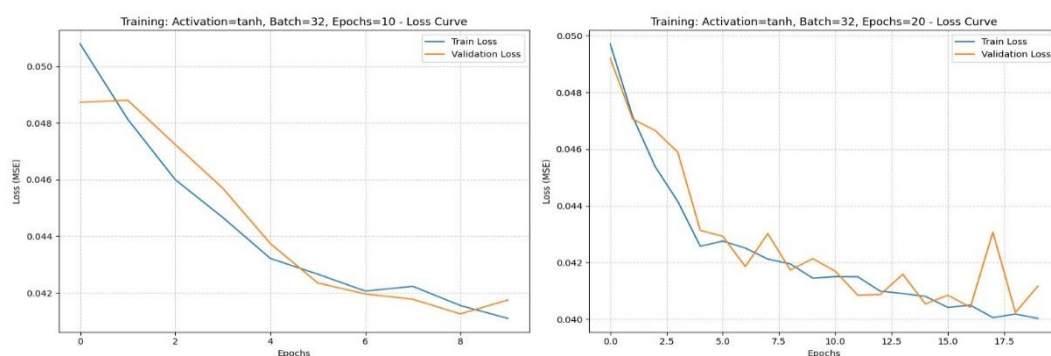
Gambar 4. 9 Grafik *Training Model LSTM* dengan variasi *Batch 16–Epoch 10*, dan *Batch 16–Epoch 20*.

Pada grafik konfigurasi *Batch 16–Epoch 50*, *train loss* terus menurun secara stabil hingga mencapai nilai yang cukup rendah, sementara *validation loss* mengalami fluktuasi tetapi masih bergerak dalam rentang yang mendekati *train loss*, menandakan model masih berada pada kondisi pelatihan yang relatif seimbang tanpa *overfitting* yang signifikan. Namun, pada konfigurasi *Batch 16–Epoch 100*, *train loss* terus turun tajam hingga jauh lebih kecil, sedangkan *validation loss* mulai meningkat dan berfluktuasi lebih besar dibandingkan *epoch 50*. Pola ini menunjukkan tanda *overfitting* yang jelas, sehingga kemampuan generalisasinya terhadap data validasi menurun.



Gambar 4. 10 Grafik *Training Model LSTM* dengan variasi *Batch 16–Epoch 50*, dan *Batch 16–Epoch 100*.

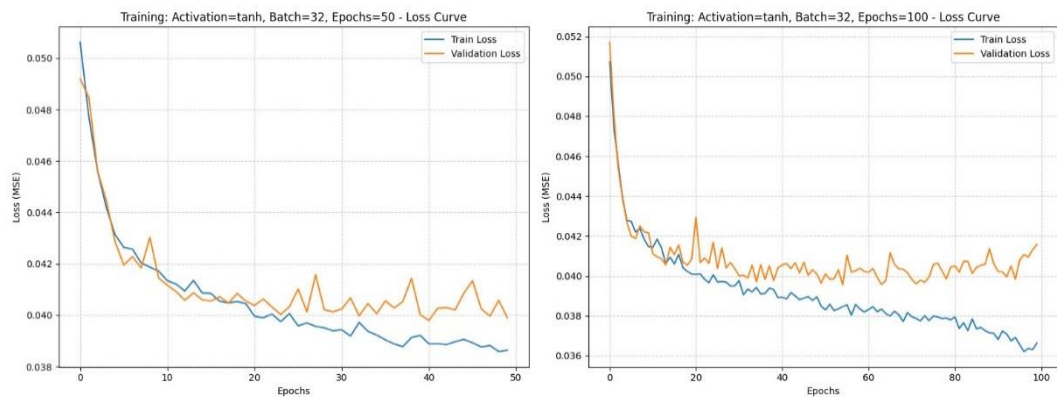
Pada grafik hasil pelatihan model LSTM dengan konfigurasi *Batch Size* 32, terlihat dua skenario pelatihan, yaitu *Epoch* 10 dan *Epoch* 20. Pada *Epoch* 10, baik *train loss* maupun *validation loss* menunjukkan penurunan yang stabil dan konsisten, tanpa adanya fluktuasi signifikan. Pola ini mengindikasikan bahwa model belajar dengan baik dan belum menunjukkan tanda-tanda *overfitting*. Sementara itu, pada *Epoch* 20, *train loss* terus menurun seperti yang diharapkan, namun *validation loss* mulai menunjukkan terdapat sedikit variasi pada beberapa titik *epoch*. Meskipun demikian, nilai *validation loss* tetap berada dalam kisaran yang mirip dengan *train loss* (tidak terlalu melebar), pola ini menandakan bahwa pelatihan yang lebih panjang mulai memberi potensi terjadinya *overfitting* ringan.



Gambar 4. 11 Grafik *Training Model LSTM* dengan variasi *Batch* 32–*Epoch* 10, dan *Batch* 32–*Epoch* 20.

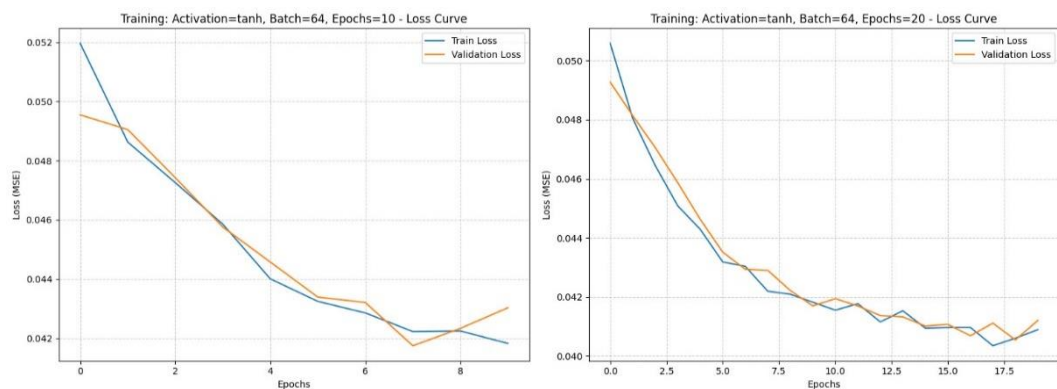
Grafik *training* model LSTM menunjukkan bahwa pada kedua skenario (*Batch* 32–*Epoch* 50 dan *Batch* 32–*Epoch* 100), nilai *loss* MSE untuk *data training* dan *validation* sama-sama menurun di awal proses pelatihan, menandakan model berhasil mempelajari pola dasar data. Pada *Epoch* 50, kurva *training* dan *validation* relatif stabil dan berdekatan, menunjukkan generalisasi yang cukup baik tanpa tanda *overfitting* yang jelas. Sementara itu, pada *Epoch* 100, *loss training* terus

menurun lebih jauh, tetapi *loss validation* mulai berfluktuasi dan mengindikasikan bahwa penambahan *epoch* menyebabkan model mulai sedikit *overfitting*.



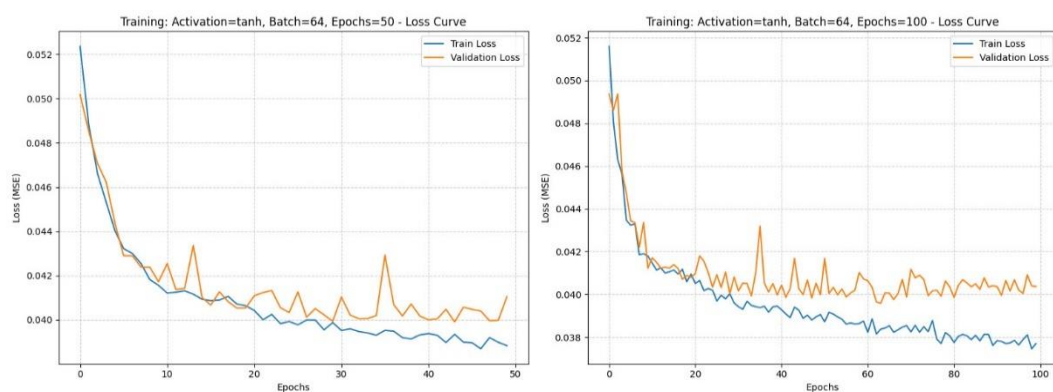
Gambar 4. 12 Grafik *Training Model LSTM* dengan variasi *Batch 32–Epoch 50*, dan *Batch 32–Epoch 100*.

Pada *Batch 64–Epoch 10* dan *Batch 64–Epoch 20*, nilai *loss MSE* untuk *training* dan *validation* sama-sama menurun secara konsisten, menandakan proses pembelajaran berjalan stabil. Pada *Epoch 10*, kurva *train* dan *validation* hampir berimpitan, sehingga model sudah mampu mempelajari pola tanpa indikasi *overfitting*. Ketika jumlah *epoch* ditingkatkan menjadi 20, *loss* terus turun dengan pola yang lebih halus dan hubungan antara *train–validation* tetap dekat, menunjukkan generalisasi yang baik dan kestabilan model.



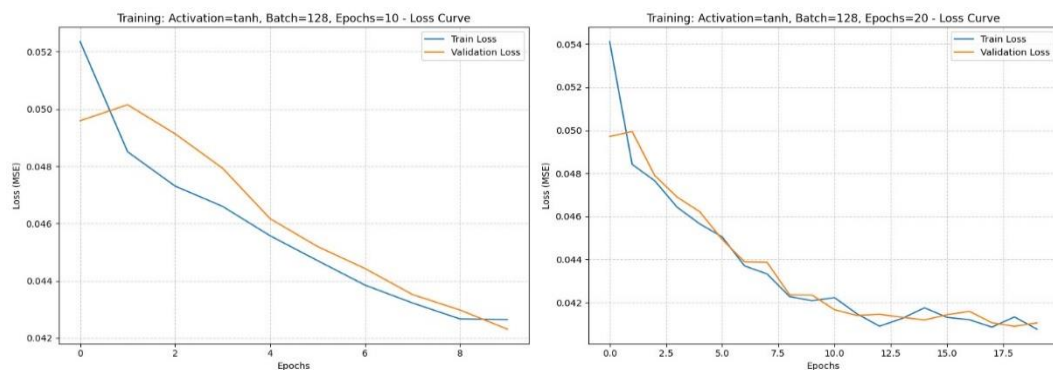
Gambar 4. 13 Grafik *Training Model LSTM* dengan variasi *Batch 64–Epoch 10*, dan *Batch 64–Epoch 20*.

Grafik pelatihan model LSTM dengan kombinasi *Batch 64–Epoch 50* dan *Batch 64–Epoch 100* memperlihatkan bagaimana penambahan jumlah *epoch* memengaruhi proses pembelajaran model dalam menurunkan nilai loss MSE. Pada kedua grafik tersebut, kurva *train loss* dan *validation loss* sama-sama menurun secara konsisten pada tahap awal pelatihan, yang menunjukkan bahwa model mampu mempelajari pola data dengan baik. Perbandingan kedua grafik ini memberikan gambaran mengenai efektivitas proses pembelajaran model serta potensi terjadinya *overfitting* ketika jumlah *epoch* diperbesar.



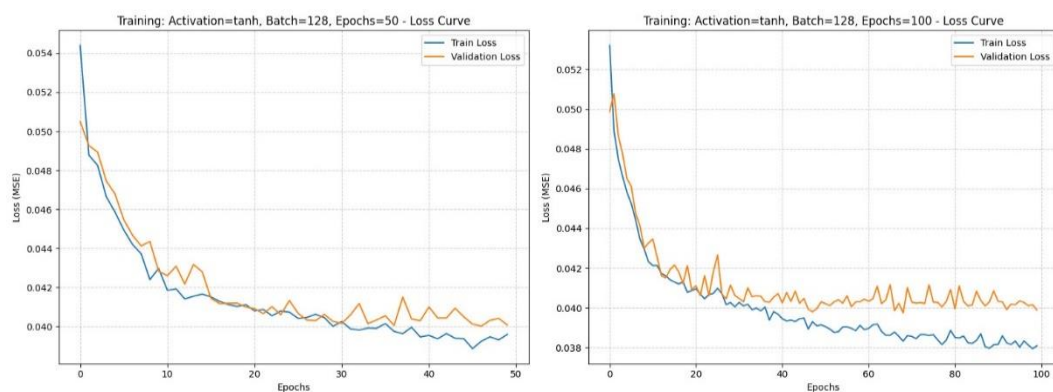
Gambar 4. 14 Grafik *Training Model LSTM* dengan variasi *Batch 64–Epoch 50*, dan *Batch 64–Epoch 100*.

Grafik *training* model LSTM dengan variasi *Batch 128–Epoch 10* dan *Batch 128–Epoch 20* menunjukkan pola penurunan *loss* yang konsisten baik pada *train loss* maupun *validation loss*, menandakan proses pembelajaran berjalan stabil. Penggunaan 20 *epoch* memberikan penurunan *loss* yang lebih halus dibandingkan 10 *epoch*, dengan jarak antara *train* dan *validation loss* yang tetap berdekatan. Hal ini mengindikasikan bahwa model tetap berada dalam kondisi pembelajaran yang optimal dan belum menunjukkan tanda-tanda *overfitting*.



Gambar 4. 15 Grafik *Training* Model LSTM dengan variasi *Batch* 128–*Epoch* 10, dan *Batch* 128–*Epoch* 20.

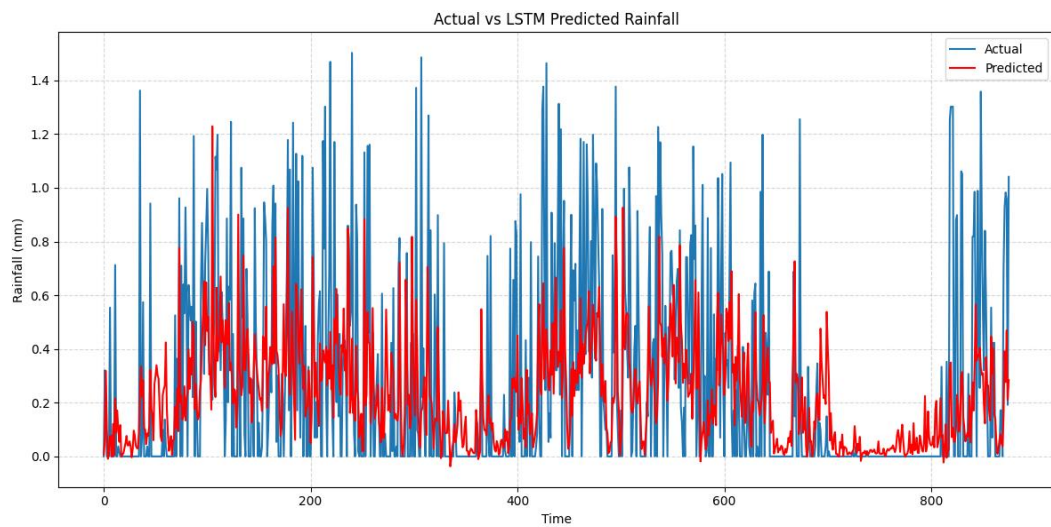
Grafik training model LSTM dengan variasi *Batch* 128–*Epoch* 50 dan *Batch* 128–*Epoch* 100 menunjukkan pola penurunan *loss* yang stabil pada kedua skenario. Pada *Epoch* 50, *train loss* dan *validation loss* turun secara konsisten dengan pergerakan yang relatif seimbang dan tanpa indikasi *overfitting*. Saat jumlah *epoch* diperpanjang menjadi 100, tren penurunan *loss* tetap berlanjut, di mana *train loss* sedikit lebih rendah dibanding *validation loss* namun masih berada dalam jarak yang sangat dekat. Hal ini mengindikasikan bahwa model tetap belajar secara efektif hingga *epoch* tinggi, dan ukuran *batch* 128 mampu menjaga stabilitas proses pelatihan tanpa menghasilkan fluktuasi ekstrem pada *validation loss*.



Gambar 4. 16 Grafik *Training* Model LSTM dengan variasi *Batch* 128–*Epoch* 50, dan *Batch* 128–*Epoch* 100.

4.6 Pengujian Model LSTM

Pada tahap pengujian, model yang telah dilatih digunakan untuk memprediksi nilai curah hujan berdasarkan data *input* X_{test} , kemudian hasil prediksi tersebut dibandingkan dengan nilai aktual pada y_{test} . Dari proses evaluasi diperoleh nilai *Root Mean Squared Error* (RMSE) sebesar 0.3090.



Gambar 4. 17 Grafik Actual Vs LSTM

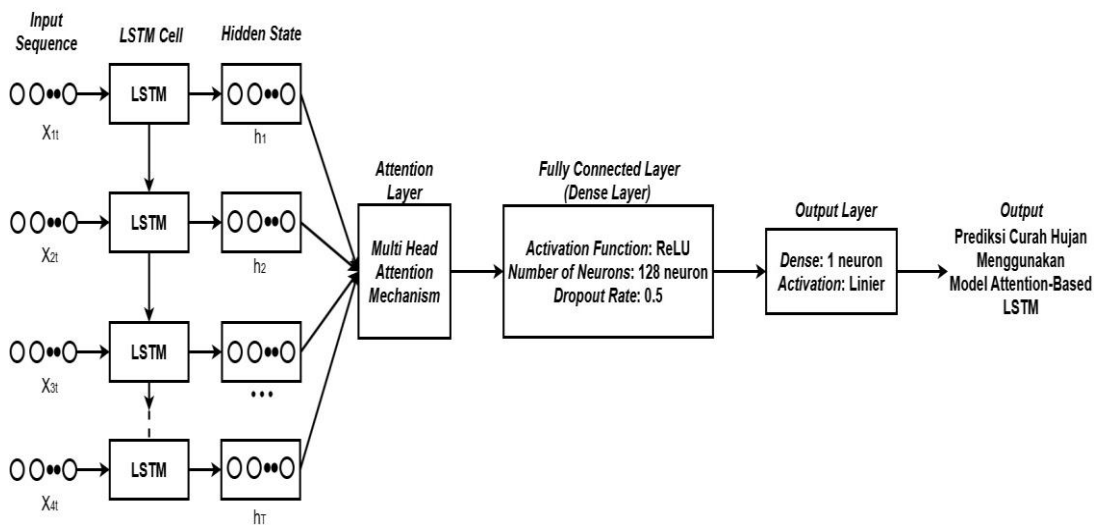
Visualisasi perbandingan antara data aktual (garis biru) dan prediksi model (garis merah) menunjukkan bahwa model LSTM berhasil menangkap tren dasar data curah hujan. Namun, garis prediksi cenderung lebih halus dan stabil, yang mengindikasikan keterbatasan model dalam memprediksi fluktuasi ekstrem atau lonjakan tajam pada data aktual. Meskipun demikian, model ini cukup baik dalam mengenali pola dasar. Performanya dapat ditingkatkan lebih lanjut untuk dapat memprediksi lonjakan curah hujan yang signifikan.

BAB V

PREDIKSI CURAH HUJAN DENGAN ATTENTION-BASED LSTM

5.1 Arsitektur Model Attention-Based LSTM

Arsitektur Model Attention-Based LSTM yang diterapkan dalam penelitian ini tersusun atas beberapa lapisan inti, seperti yang diperlihatkan pada Gambar 5.1. *Input* model berupa data sekuensial (*time series*) dengan panjang T (*timesteps*). Pada setiap *timestep* ke- t , *input* berupa vektor fitur x_t yang terdiri dari lima variabel, yaitu x_{1t} suhu rata-rata (Tavg), x_{2t} kelembaban relatif rata-rata (RH_avg), x_{3t} durasi penyinaran matahari (ss), dan x_{4t} kecepatan angin rata-rata (ff_avg). Setiap *input* sekuensial ini diproses oleh sel LSTM yang menghasilkan *hidden state* (h_1, h_2, \dots, h_T). *Hidden state* h_t merepresentasikan informasi yang dikumpulkan hingga *timestep* ke- t , sehingga h_1 berisi ringkasan informasi dari *input* pertama, h_2 dari dua *input* pertama, dan seterusnya hingga h_T yang menyimpan ringkasan keseluruhan urutan *input*. Dengan kata lain, *hidden state* merupakan memori internal yang menyimpan pola ketergantungan temporal pada data.



Gambar 5. 1 Arsitektur Model Attention-Based LSTM

Berbeda dengan arsitektur LSTM sebelumnya yang tanpa mekanisme Attention, semua *hidden state* ini tidak langsung diratakan (*flatten*), tetapi justru dikirimkan ke *Attention Layer*. Mekanisme *Attention* kemudian menghitung bobot penting dari setiap *hidden state*, sehingga *hidden state* yang lebih relevan terhadap target prediksi diberi kontribusi lebih besar. Proses ini memungkinkan model untuk fokus pada langkah waktu tertentu yang paling berpengaruh terhadap curah hujan. *Output* dari *Attention Layer* kemudian dialirkan ke lapisan *fully connected (dense layer)* yang berisi 128 *neuron* dengan fungsi aktivasi ReLU serta dilengkapi *dropout* sebesar 0,5 untuk mengurangi risiko *overfitting*. Pada bagian akhir, hasil pemrosesan diproyeksikan melalui *dense layer* dengan satu *neuron* dan aktivasi linear, yang menghasilkan *output* berupa prediksi curah hujan harian. Dengan rancangan ini, model tidak hanya menangkap pola ketergantungan temporal melalui *hidden state* hasil LSTM, tetapi juga memperkuat pemanfaatan informasi melalui mekanisme *Attention*, sehingga prediksi curah hujan menjadi lebih akurat.

5.2 Hyperparameter Attention-Based LSTM

Proses pengumpulan dan *pre-processing* data pada metode Attention-Based LSTM dikerjakan dengan prosedur yang sama seperti pada metode LSTM yang telah dijelaskan pada bab sebelumnya. Perbedaan utamanya terletak pada arsitektur model, di mana Attention-Based LSTM menggunakan tambahan *layer* Attention untuk menangkap informasi penting dalam urutan data secara lebih efektif, dan penting untuk menetapkan model yang akan dibangun berdasarkan parameter yang telah dipilih. Tambahan ini berfungsi untuk memperkuat kemampuan model dalam memusatkan perhatian pada *timestep* yang paling penting, sehingga mampu

menghasilkan prediksi yang lebih stabil dan akurat dibandingkan LSTM tanpa mekanisme *Attention*. Beberapa parameter yang digunakan meliputi:

Tabel 5. 1 *Hyperparameter* Model Attention-Based LSTM

<i>Hyperparameter yang diterapkan</i>	<i>Nilai / Keterangan</i>
<i>Batch size</i>	4, 16, 32, 64, 128
<i>Epochs</i>	10, 20, 50, 100
<i>Optimizer</i>	Adam
Fungsi Aktivasi	tanh, ReLU
<i>Dropout Rate</i>	0.5
<i>Learning Rate</i>	1e-3
<i>Loss Function</i>	<i>Mean Square Error</i>

Berdasarkan tabel di atas, Model Attention-Based LSTM dilatih dengan konfigurasi *hyperparameter* yang terdefinisi dengan jelas untuk menguji performanya. Model ini memanfaatkan optimizer Adam dan menggunakan *Mean Squared Error* (MSE) sebagai fungsi loss untuk menilai ketepatan hasil prediksi. Berbagai nilai *batch size* (4, 16, 32, 64, 128) dan *epochs* (10, 20, 50, 100) diterapkan dalam pelatihan. Selain itu, fungsi aktivasi tanh, dan ReLU juga digunakan untuk mengatasi *overfitting*, dan nilai *learning rate* (1e-3, 5e-4, 1e-4) diuji untuk menemukan laju pembelajaran yang paling efisien.

Attention-Based LSTM, yang memiliki kemampuan tambahan dalam memfokuskan perhatian pada informasi relevan, dengan menggunakan *dropout* 0.5 yang berperan ganda, yaitu untuk mengatasi *overfitting*, dan juga membantu menstabilkan distribusi bobot pada lapisan Attention sehingga fokus model tidak terlalu bergantung pada satu fitur tertentu. Selain itu, temuan dari Liu *et al.*, (2023) mengindikasikan bahwa penggunaan *dropout* juga dapat membantu meminimalkan

underfitting pada fase awal pelatihan sehingga pemilihan nilai 0.5 dalam konteks ini masih relevan untuk memperkuat kemampuan eksplorasi model Attention dalam menangkap pola temporal yang kompleks.

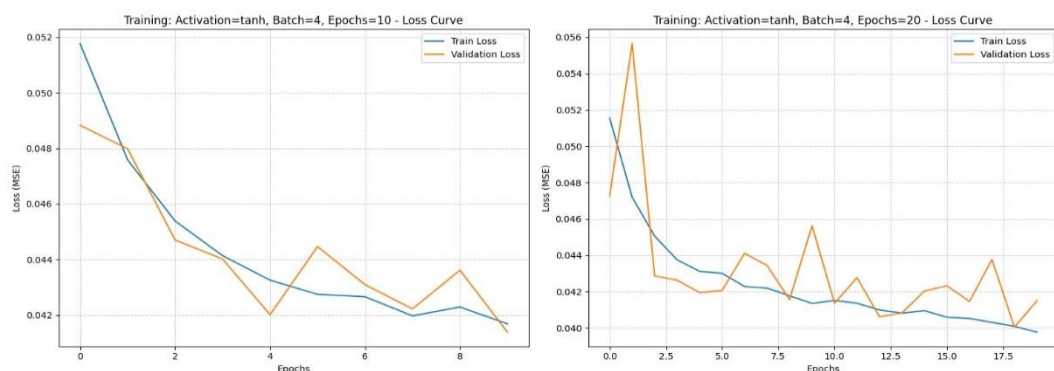
5.3 Training Model Attention-Based LSTM

Pada tahap ini, model dilatih menggunakan algoritma optimasi Adam dengan memanfaatkan fungsi aktivasi tanh dan ReLU. Berbagai kombinasi *hyperparameter* diuji, mencakup ukuran batch (4, 16, 32, 64, dan 128) serta jumlah epochs (10, 20, 50, dan 100). Selama proses pelatihan, model terbaik didapatkan dengan memantau nilai *loss* (*Mean Squared Error* atau MSE) pada data pelatihan dan validasi.

Data kemudian dipisahkan menjadi dua kelompok utama, yaitu 80% untuk pelatihan (*training set*) dan 20% lainnya untuk pengujian (*testing set*). Tahapan validasi model tidak lagi dilakukan melalui pembagian data secara terpisah, melainkan menggunakan metode *validation split 10* pada data pelatihan untuk memastikan keandalan dan keakuratan performa model secara lebih optimal. Selama proses pelatihan, nilai *loss* yang dihasilkan oleh fungsi kerugian digunakan untuk menilai sejauh mana prediksi model menyimpang dari nilai aktual. Tujuan utama dari pelatihan adalah meminimalkan nilai *loss* tersebut. Train loss merupakan nilai *loss* yang dihitung berdasarkan data pelatihan, sedangkan *validation loss* dihitung menggunakan data validasi, yaitu data yang tidak pernah dilihat model selama proses pelatihan. Nilai *validation loss* memberikan gambaran mengenai kemampuan model dalam melakukan generalisasi atau kinerjanya terhadap data baru (Wosu *et al.*, 2025).

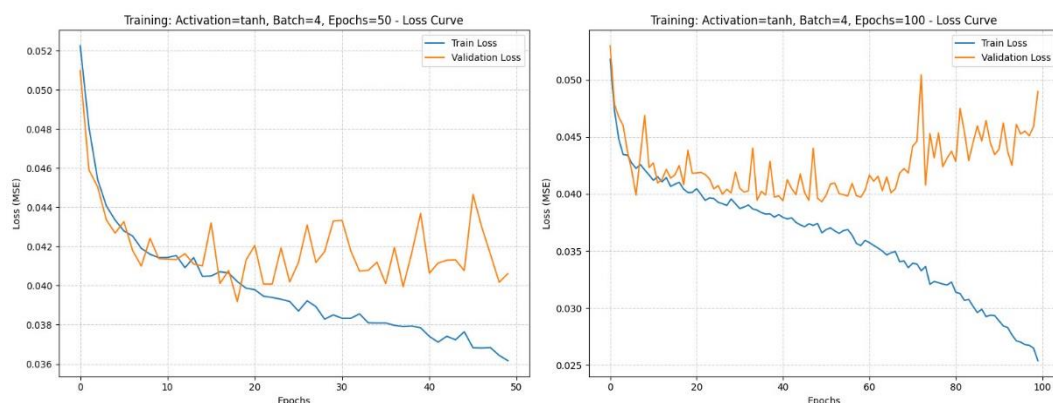
Berikut adalah nilai *loss* MSE dari setiap konfigurasi *epochs* yang disajikan dalam bentuk visualisasi. Penggunaan grafik ini dipilih agar perbandingan performa model pada berbagai jumlah *epochs* terlihat lebih jelas dan mudah dipahami, karena *loss* MSE sering digunakan untuk masalah regresi, seperti prediksi curah hujan. Karena sifatnya yang mudah dipahami dan efisien secara komputasi, sebagai metrik evaluasi standar dalam penelitian regresi (Elharrouss *et al.*, 2025). Visualisasi ini menampilkan sampel plot dari beberapa *batch* dan *epochs* yang berbeda, sementara tabel metrik *training* Attention-Based LSTM penjelasan lebih rinci dapat dilihat pada Lampiran 2.

Pada grafik *training* AT-LSTM menunjukkan penurunan *train loss* dan *validation loss* yang stabil pada epoch 10 maupun 20. Pola ini menandakan bahwa model masih berada dalam fase *learning* yang baik, di mana kurva pelatihan dan validasi bergerak berdekatan tanpa adanya gap yang signifikan. Fluktuasi kecil pada *validation loss* masih termasuk normal dan menunjukkan proses adaptasi model terhadap variasi data. Secara keseluruhan, pada rentang *epoch* rendah (10–20), model belum menunjukkan indikasi *overfitting* dan masih mampu mempertahankan generalisasi yang baik.



Gambar 5. 2 Grafik *Training* Model AT-LSTM dengan variasi *Batch* 4–*Epoch* 10, dan *Batch* 4–*Epoch* 20.

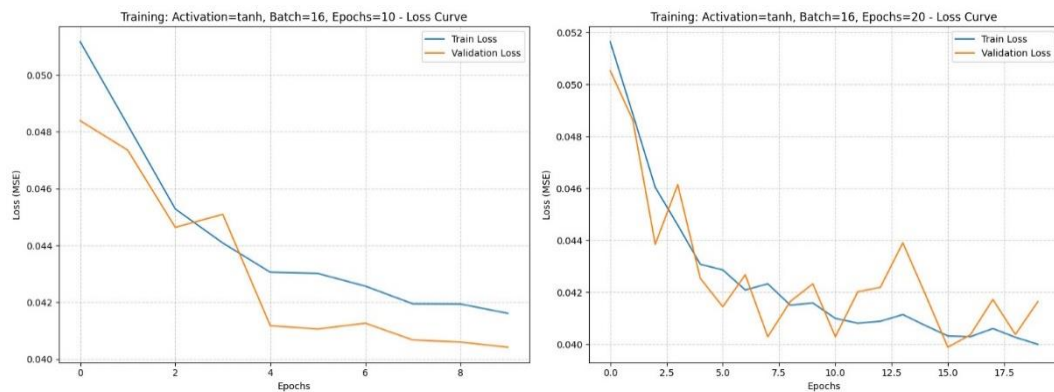
Pada Gambar 5.3, *train loss* terus menurun secara konsisten, namun *validation loss* mulai menunjukkan pola meningkat dan semakin tidak stabil, terutama pada *epoch* 100. Temuan ini menunjukkan bahwa model mulai memasuki fase *overfitting*, di mana model semakin mahir mempelajari detail pada data pelatihan namun kemampuan generalisasinya terhadap data validasi menurun. Perbedaan antara *train loss* dan *validation loss* juga tampak semakin besar seiring bertambahnya jumlah *epoch*. Dengan kata lain, penggunaan *epoch* yang terlalu tinggi (50–100) tidak memberikan peningkatan kinerja model secara menyeluruh, bahkan dapat mengurangi kestabilan hasil prediksi.



Gambar 5. 3 Grafik *Training Model AT-LSTM* dengan variasi *Batch 4–Epoch 50*, dan *Batch 4–Epoch 100*.

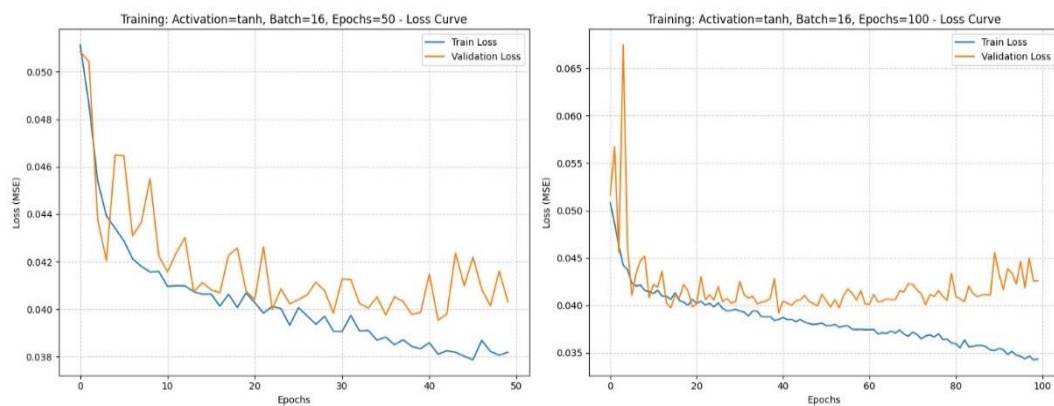
Pada pengujian dengan *Batch Size* 16 dan *Epoch* 10, kurva menunjukkan penurunan *loss* yang konsisten pada data *training* maupun *validation*. Kesesuaian pola penurunan ini mengindikasikan bahwa model mampu mempelajari pola curah hujan tanpa mengalami *overfitting* pada tahap awal. Ketika *epoch* ditingkatkan menjadi 20, *loss training* terus menurun secara bertahap, sementara *loss validation* sedikit lebih fluktuatif, namun masih berada pada kisaran yang cukup dekat dengan

training loss. Hal ini menunjukkan bahwa peningkatan *epoch* hingga 20 masih memberikan generalisasi yang cukup baik bagi model AT-LSTM.



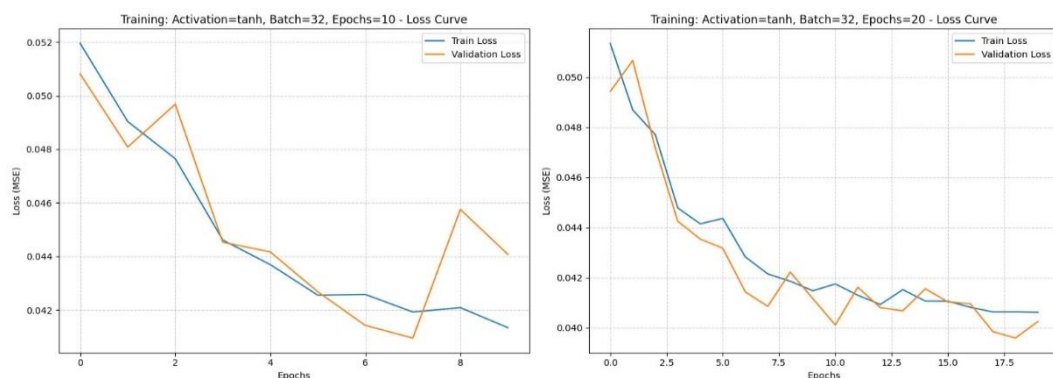
Gambar 5. 4 Grafik *Training Model AT-LSTM* dengan variasi *Batch 16–Epoch 10*, dan *Batch 16–Epoch 20*.

Pada pengujian dengan *Epoch 50*, kurva *training loss* terus mengalami penurunan yang konsisten, sedangkan *validation loss* mulai menunjukkan fluktuasi yang lebih sering. Fluktuasi tersebut mengindikasikan bahwa model mulai mendekati kondisi *overfitting*, karena performa pada data *training* terus membaik sementara performa pada *validation* cenderung tidak stabil. Kondisi ini semakin terlihat jelas pada *Epoch 100*, di mana *training loss* menurun tajam namun *validation loss* justru cenderung meningkat dan semakin tidak konsisten.



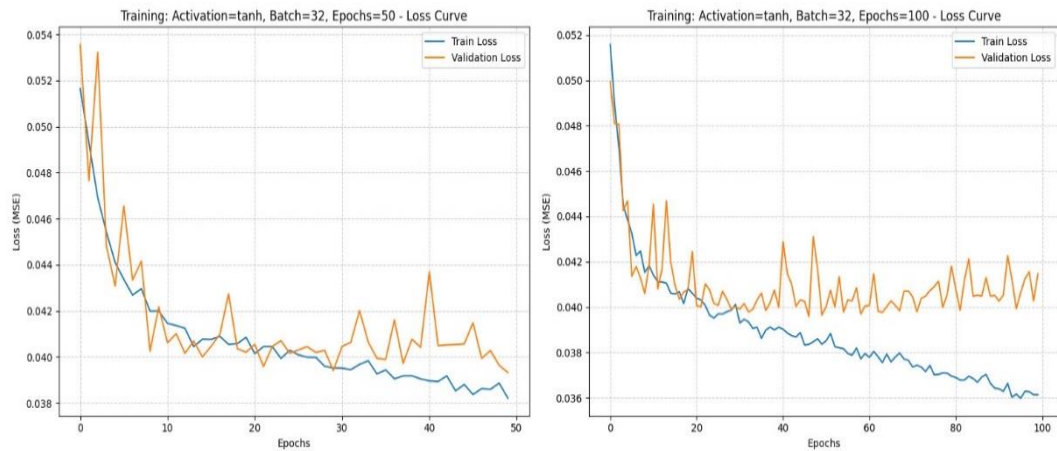
Gambar 5. 5 Grafik *Training Model AT-LSTM* dengan variasi *Batch 16–Epoch 50*, dan *Batch 16–Epoch 100*.

Pada grafik Batch 32 – *Epoch* 10 dan *Epoch* 20 menunjukkan pelatihan yang relatif lebih stabil, karena penggunaan *Batch Size* 32 mengurangi noise pada *gradients*. Pada *Epoch* 10, model tampak *underfitting*, karena *Validation Loss* (oranye) menunjukkan penurunan tajam dan mencapai titik terendah 0.041 tepat sebelum *Epoch* terakhir. Dengan perpanjangan ke 20 *Epoch*, model mencapai *Validation Loss* yang lebih rendah = 0.040, dengan kurva yang lebih halus dan gap antara Pelatihan dan Validasi *Loss* yang terjaga. Ini mengindikasikan bahwa 20 *Epoch* adalah durasi pelatihan yang lebih optimal dan stabil daripada 10 *Epoch* untuk konfigurasi ini.



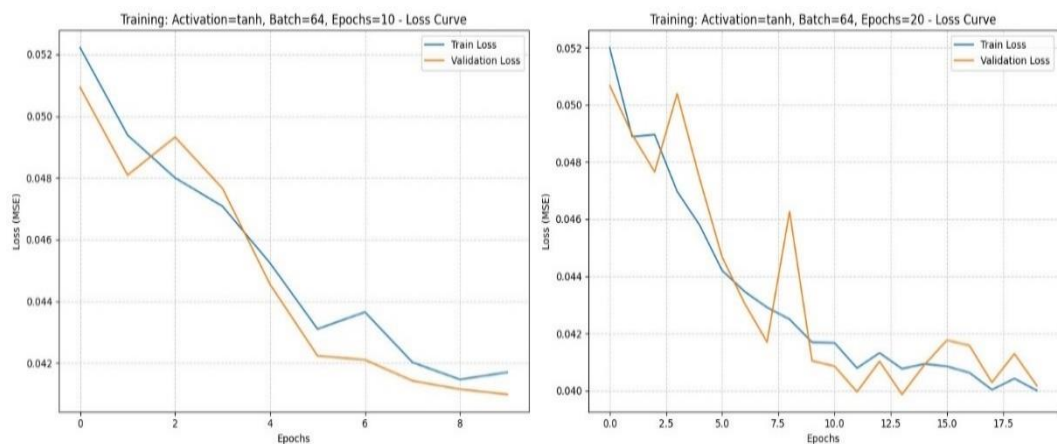
Gambar 5. 6 Grafik *Training Model AT-LSTM* dengan variasi *Batch* 32–*Epoch* 10, dan *Batch* 32–*Epoch* 20.

Pada pelatihan model AT-LSTM dari 20 *Epoch* ke 50 *Epoch* dan 100 *Epoch* menunjukkan tanda-tanda *overfitting* yang semakin nyata. Pada *Epoch* 50, *Training Loss* terus menurun, namun *Validation Loss* mulai stagnan dan berfluktuasi di atas kurva *Training Loss* = 0.040, menandakan model sudah mencapai titik jenuh belajar fitur baru. Ketika diperpanjang hingga 100 *Epoch*, gap antara kedua kurva melebar secara signifikan. *Training Loss* terus turun ke nilai terendah = 0.036, sementara *Validation Loss* melambung dan berfluktuasi tinggi.



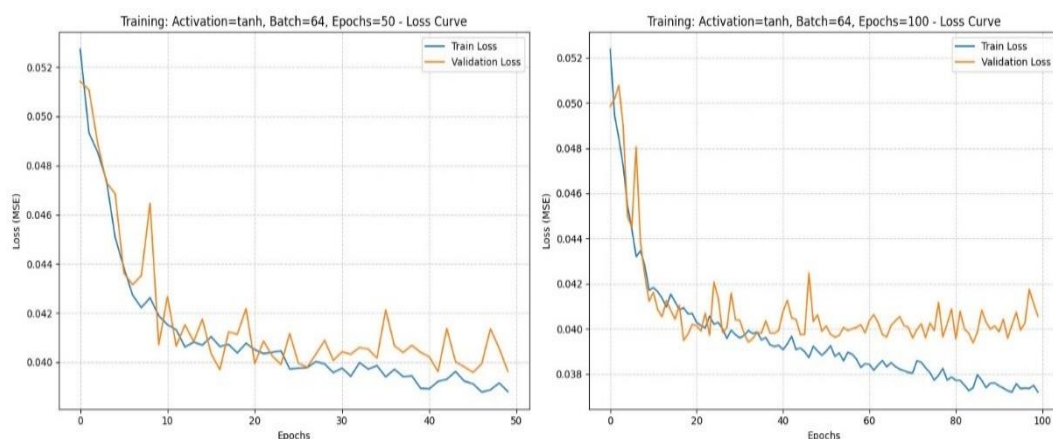
Gambar 5. 7 Grafik *Training Model AT-LSTM* dengan variasi *Batch 32–Epoch 50*, dan *Batch 32–Epoch 100*.

Pada *Batch Size 64* yang efektif mengurangi noise pada *gradients*. Pada *Epoch 10*, *Validation Loss* (oranye) dan *Training Loss* (biru) menurun beriringan, menunjukkan konvergensi yang sehat, meskipun *Validation Loss* masih sedikit di atas *Training Loss* di akhir. Peningkatan ke *20 Epoch* memungkinkan model untuk mencapai *Validation Loss* yang lebih kecil serta jarak antar kurva yang lebih sempit dan stabil, mengonfirmasi bahwa *20 Epoch* adalah durasi yang lebih optimal daripada *10 Epoch* untuk konfigurasi *Batch Size 64*.



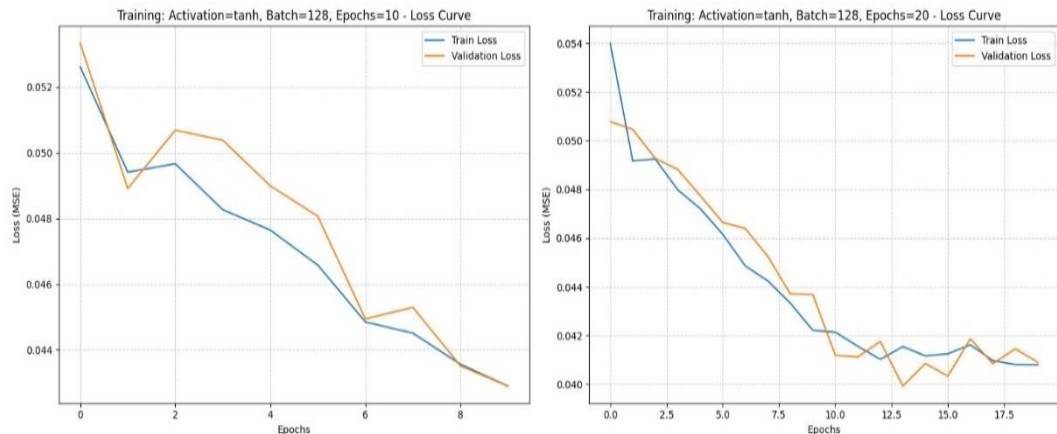
Gambar 5. 8 Grafik *Training Model AT-LSTM* dengan variasi *Batch 64–Epoch 10*, dan *Batch 64–Epoch 20*.

Pada grafik *Batch 64 – Epoch 50* dan *Epoch 100* menunjukkan model AT-LSTM mulai mencapai batas pembelajarannya. Pada *Epoch 50*, *Validation Loss* mulai stagnan dan berfluktuasi di sekitar nilai = 0.040, sementara *Training Loss* terus menurun, menandakan *Early Overfitting*. Perpanjangan ke 100 *Epoch* memperburuk kondisi ini, *Training Loss* terus turun ke nilai terendah = 0.036, tetapi *Validation Loss* menjauh dan tetap fluktuatif di atas = 0.040. Hal ini membuktikan bahwa 50 *Epoch* sudah cukup, dan melanjutkan pelatihan hingga 100 *Epoch* hanya menyebabkan *overfitting* tanpa meningkatkan kinerja generalisasi model secara berarti.



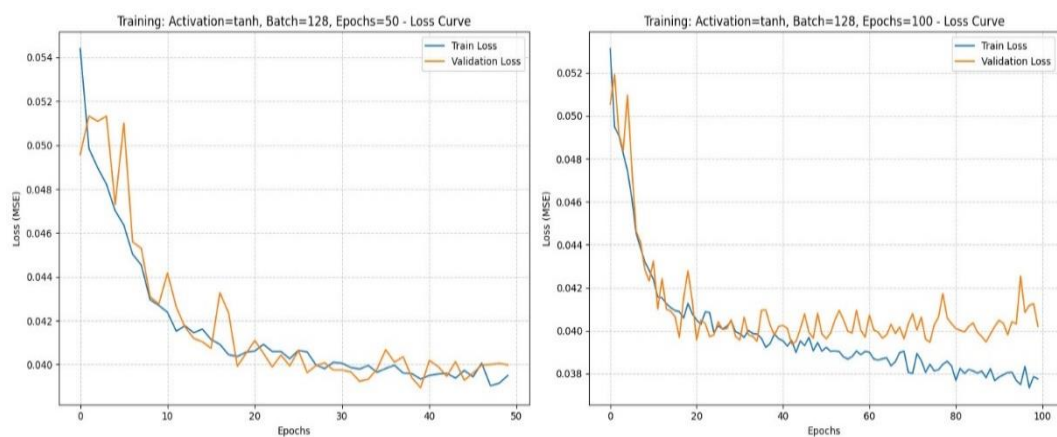
Gambar 5. 9 Grafik *Training Model AT-LSTM* dengan variasi *Batch 64–Epoch 50*, dan *Batch 64–Epoch 100*.

Pada grafik *Training model AT-LSTM* dengan variasi *Batch 128–Epoch 10*, dan *Batch 128–Epoch 20*, model memang menunjukkan penurunan *loss* secara bertahap, namun nilai *loss* akhir pada kedua kurva tetap tinggi, sehingga menandakan bahwa model mengalami *underfitting*. Hal ini menegaskan bahwa 20 *Epoch* adalah durasi yang lebih memadai dibandingkan 10 *Epoch* untuk konfigurasi *Batch Size 128*.



Gambar 5. 10 Grafik *Training Model AT-LSTM* dengan variasi *Batch 128–Epoch 10*, dan *Batch 128–Epoch 20*.

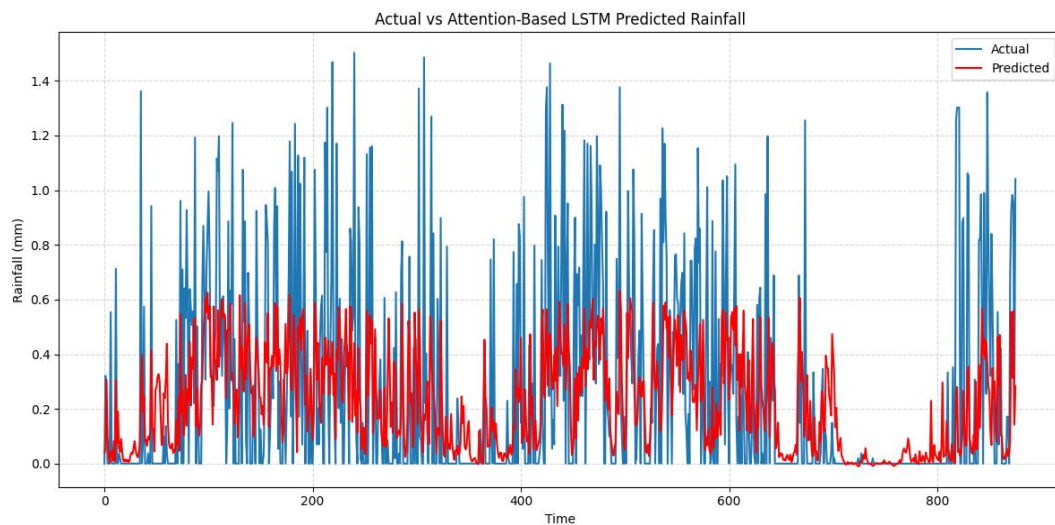
Pada *Batch Size 128* dari *50 Epoch* dan *100 Epoch* menunjukkan bahwa model telah mencapai konvergensi optimal lebih awal. Pada *Epoch 50*, kurva *Validation Loss* (oranye) dan *Training Loss* (biru) sudah berada di level *loss* terendah dan bergerak berdekatan, meskipun fluktuasi *Validation Loss* mulai terlihat. Hal ini menunjukkan bahwa model masih mampu belajar dengan efektif hingga mencapai jumlah *epoch* yang tinggi, dan ukuran *batch 128* mampu menjaga stabilitas proses pelatihan tanpa menghasilkan fluktuasi ekstrem pada *validation loss*.



Gambar 5. 11 Grafik *Training Model AT-LSTM* dengan variasi *Batch 128–Epoch 50*, dan *Batch 128–Epoch 100*.

5.4 Pengujian Model Attention-Based LSTM

Pada tahap pengujian, model terbaik diuji menggunakan data X_{test} dengan membandingkan hasil prediksi curah hujan terhadap nilai aktual pada y_{test} . Dari proses evaluasi diperoleh nilai *Root Mean Squared Error* (RMSE) sebesar 0.3068.



Gambar 5. 12 Grafik Actual Vs Attention-Based LSTM

Grafik di atas menampilkan perbandingan antara data curah hujan aktual dan hasil prediksi yang dihasilkan oleh model Attention-Based LSTM. Garis berwarna biru merepresentasikan nilai curah hujan sebenarnya, sedangkan garis merah menggambarkan nilai prediksi model. Dari grafik tersebut, tampak bahwa Attention-Based LSTM mampu mengikuti pola umum perubahan curah hujan. Namun, pada beberapa titik dengan puncak curah hujan yang tinggi, model belum mampu menirukan lonjakan tersebut secara akurat. Sebaliknya, untuk nilai curah hujan yang rendah, prediksi model terlihat lebih mendekati nilai aktual.

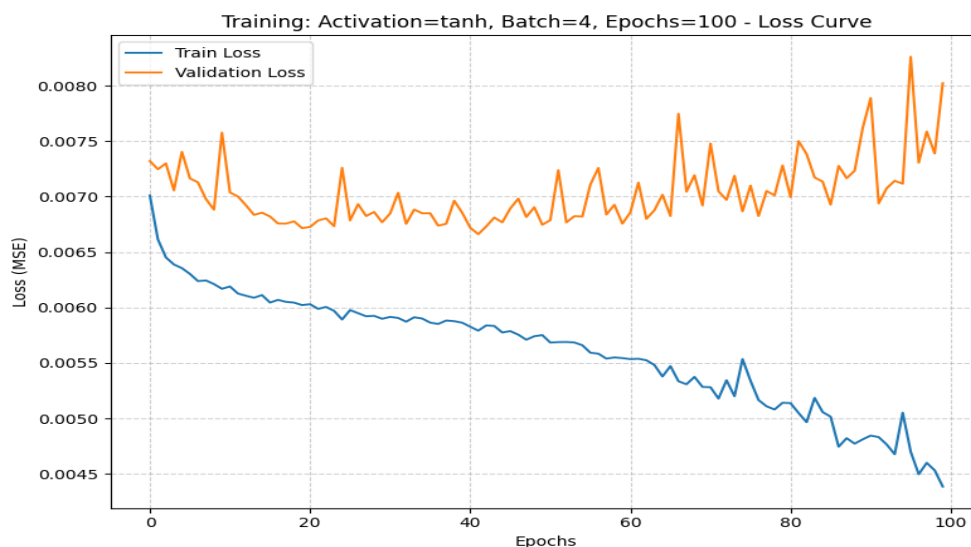
BAB VI

TRANSFORMASI DATA PADA LSTM & ATTENTION-BASED LSTM

6.1 Hasil Prediksi Tanpa Transformasi Data

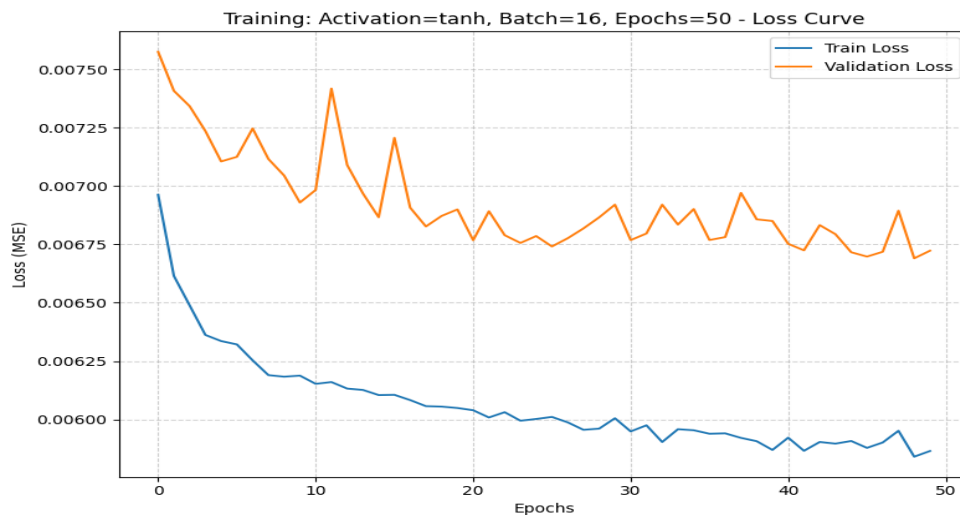
Pada tahap pertama, dilakukan evaluasi model LSTM tanpa menggunakan transformasi data. Hasil evaluasi ditunjukkan melalui grafik kurva *loss* dan MSE untuk lima konfigurasi terbaik. Dari grafik tersebut terlihat bahwa performa model LSTM bervariasi, dengan tren pelatihan yang semakin stabil seiring bertambahnya jumlah *epoch*. Model terbaik dipilih berdasarkan nilai MSE dan RMSE terkecil pada data uji, sehingga model yang menghasilkan galat paling rendah dianggap memiliki performa prediksi paling akurat dan bisa merepresentasikan kemampuan paling optimal dalam memprediksi curah hujan.

Pada konfigurasi *Batch Size* 4 dengan 100 *epoch*, nilai *train loss* menurun secara konsisten, namun *validation loss* cenderung stagnan bahkan sedikit meningkat pada akhir proses pelatihan.



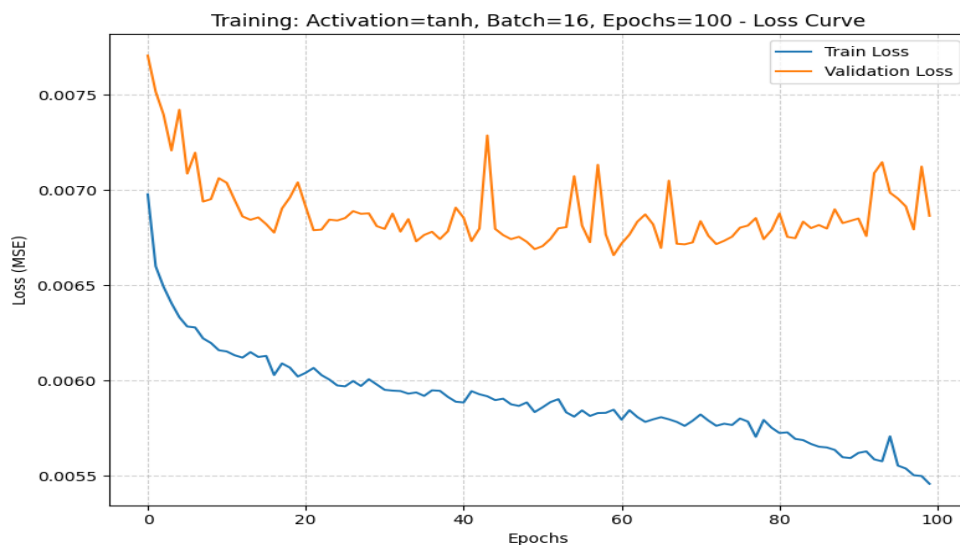
Gambar 6. 1 Kurva loss model LSTM *Batch* 4 dan *Epoch* 100

Berbeda dengan itu, pada *Batch Size* 16 dengan 50 *epoch*, *train loss* juga mengalami penurunan yang baik, sementara *validation loss* lebih stabil dan tidak menunjukkan kenaikan signifikan.



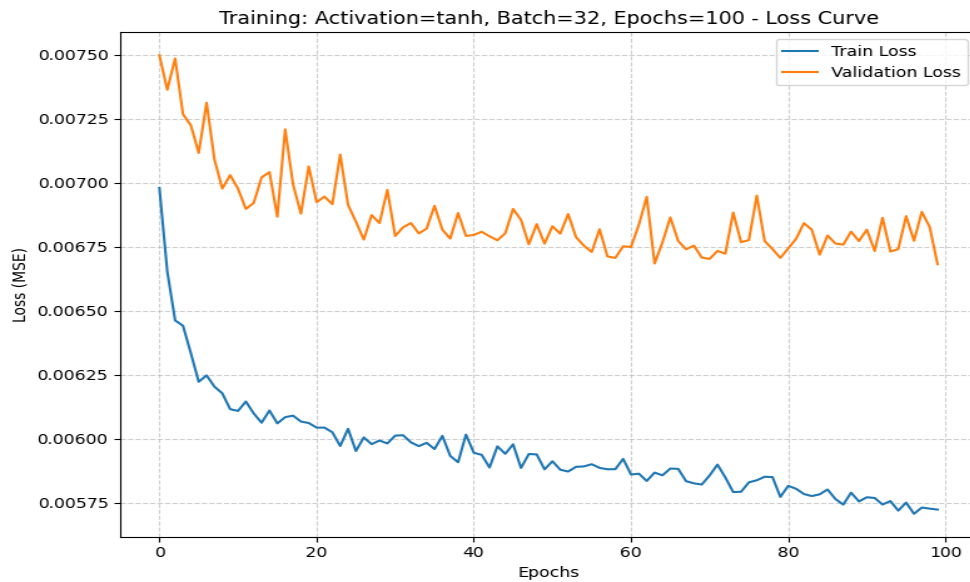
Gambar 6. 2 Kurva loss model LSTM *Batch* 16 dan *Epoch* 50

Kemudian, ketika jumlah *epoch* ditingkatkan menjadi 100 *epoch* pada *batch size* 16, pola yang muncul relatif mirip, hanya saja *validation loss* mulai berfluktuasi pada *epoch-epoch* akhir.



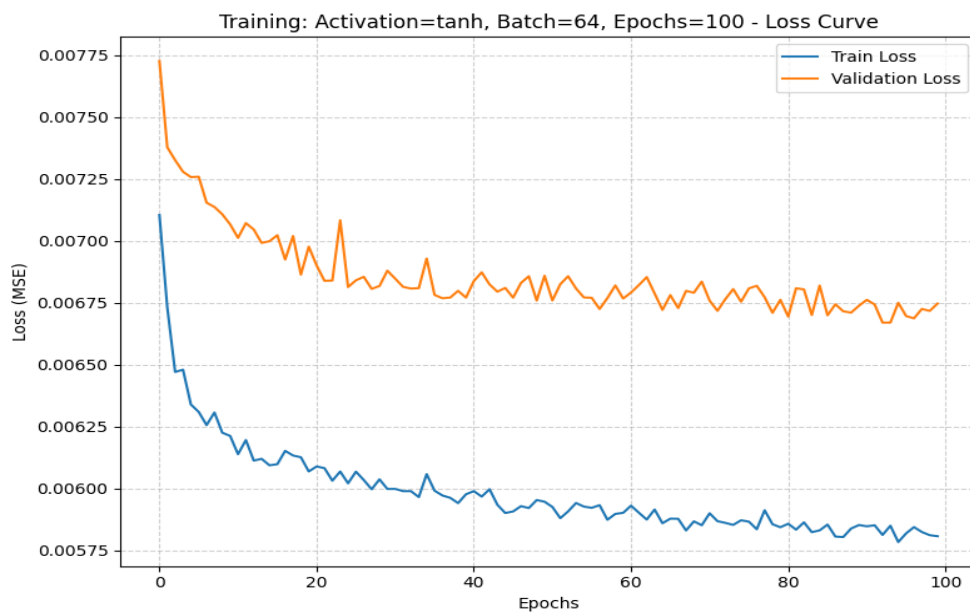
Gambar 6. 3 Kurva loss model LSTM *Batch* 16 dan *Epoch* 100

Selanjutnya, pada konfigurasi *Batch Size* 32 dengan 100 *epoch*, baik *train loss* maupun *validation loss* menunjukkan tren yang stabil.



Gambar 6. 4 Kurva loss model LSTM *Batch* 32 dan *Epoch* 100

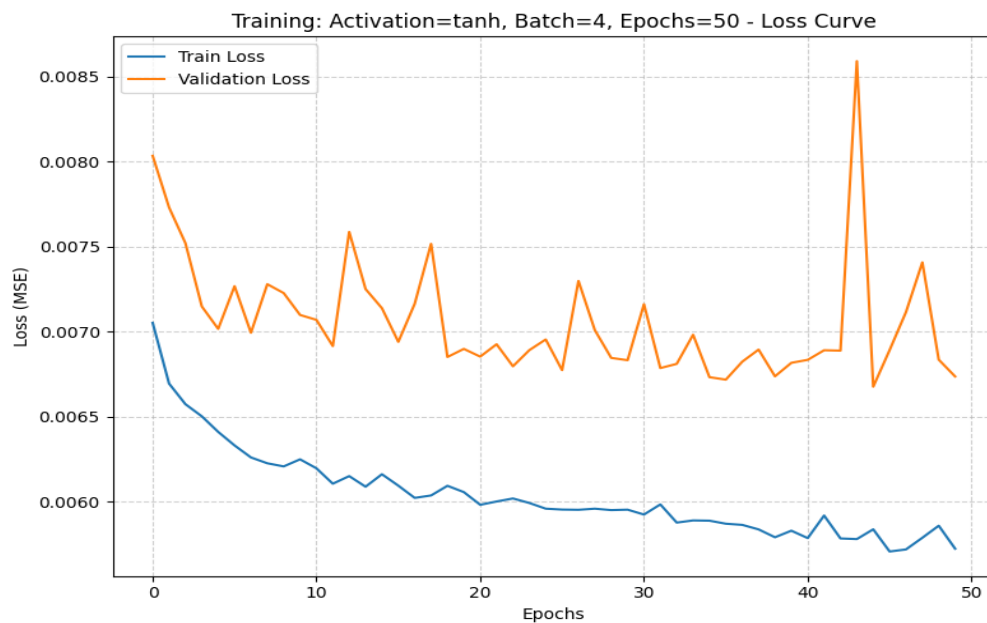
Sementara itu, pada *Batch Size* 64 dengan 100 *epoch*, *train loss* tetap menurun meskipun lebih lambat akibat ukuran *batch* yang lebih besar, tetapi *validation loss* cenderung stabil bahkan sedikit menurun.



Gambar 6. 5 Kurva loss model LSTM *Batch* 64 dan *Epoch* 100

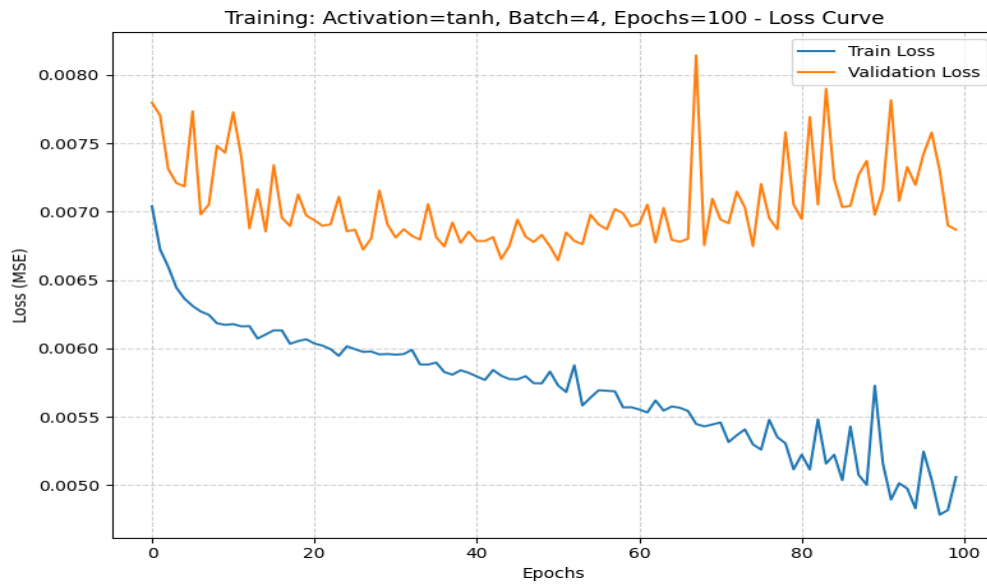
Selanjutnya, dilakukan evaluasi pada model Attention-Based LSTM (AT-LSTM) dengan konfigurasi yang sama. Hasilnya juga divisualisasikan dalam bentuk grafik, yang memperlihatkan performa pelatihan pada lima konfigurasi terbaik. Dibandingkan dengan LSTM tanpa mekanisme Attention, model Attention-Based LSTM secara keseluruhan, model tersebut menunjukkan kinerja yang sedikit unggul berkat perolehan nilai MSE (*Mean Squared Error*) dan RMSE (*Root Mean Squared Error*) yang lebih rendah pada sejumlah konfigurasi parameter. Keunggulan ini disebabkan oleh penggunaan mekanisme *attention*, yang memungkinkan model lebih memusatkan perhatian pada pola-pola kunci dalam data deret waktu, sehingga menghasilkan prediksi dengan tingkat akurasi yang lebih tinggi.

Pada *Batch Size 4, Epoch 50*. *Train loss* menurun stabil, namun *validation loss* cukup fluktuatif.



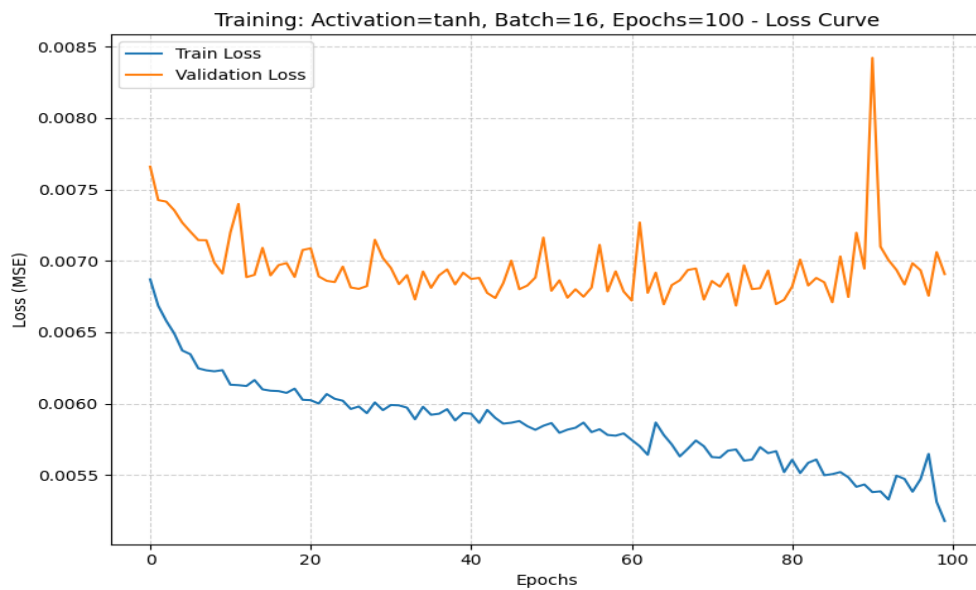
Gambar 6. 6 Kurva loss model AT-LSTM *Batch 4* dan *Epoch 50*

Kemudian pada *Batch Size 4, Epoch 100*. *Train loss* turun konsisten, tetapi *validation loss* semakin tidak stabil.



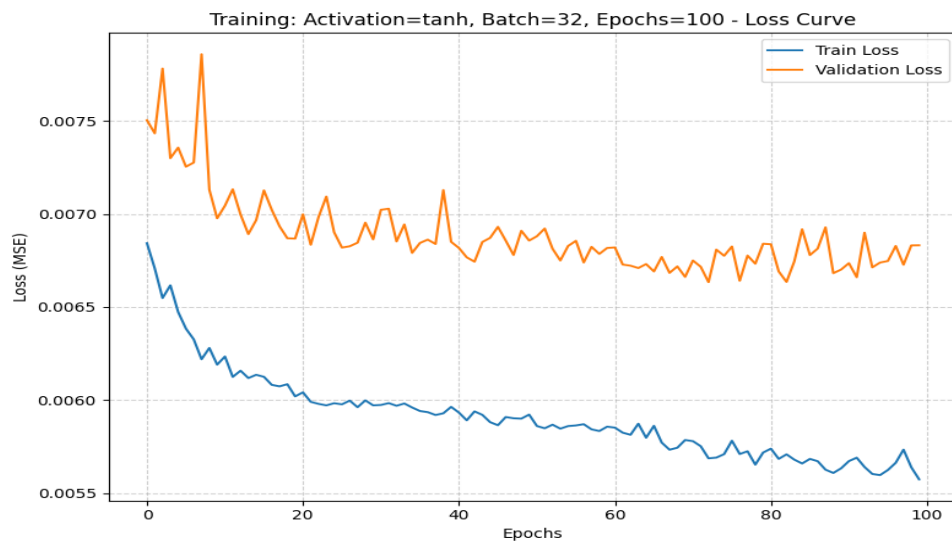
Gambar 6. 7 Kurva loss model AT-LSTM *Batch 4* dan *Epoch 100*

Selanjutnya, *Batch Size 16, Epoch 100*. *Train loss* menurun baik, *validation loss* cenderung lebih stabil dibanding *batch 4*. Masih ada fluktuasi, tetapi generalisasi lebih seimbang.



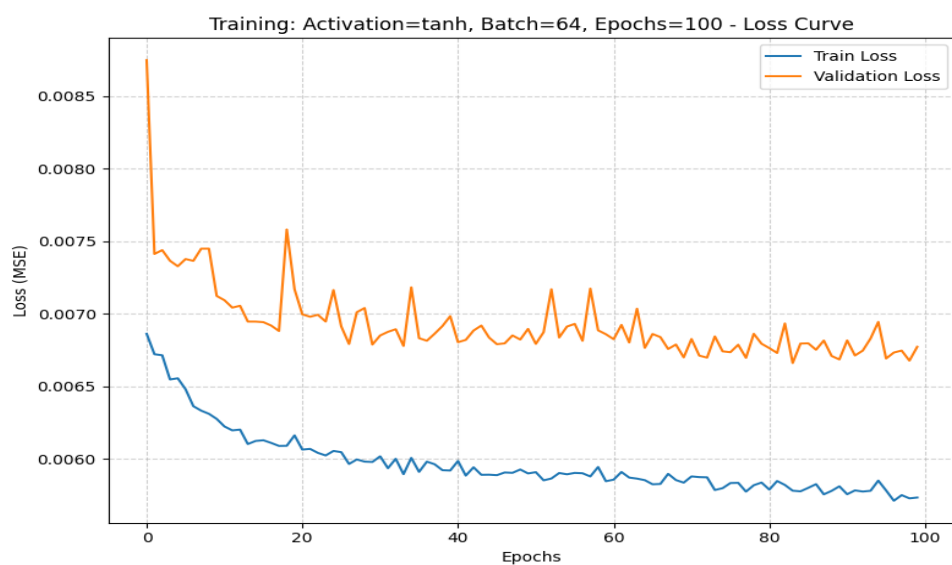
Gambar 6. 8 Kurva loss model AT-LSTM *Batch 16* dan *Epoch 100*

Untuk *Batch Size 32, Epoch 100*. *Train loss* konsisten turun, *validation loss* relatif stabil dan tidak naik signifikan. Konfigurasi ini terlihat cukup seimbang antara training dan validasi.



Gambar 6. 9 Kurva loss model AT-LSTM *Batch 32* dan *Epoch 100*

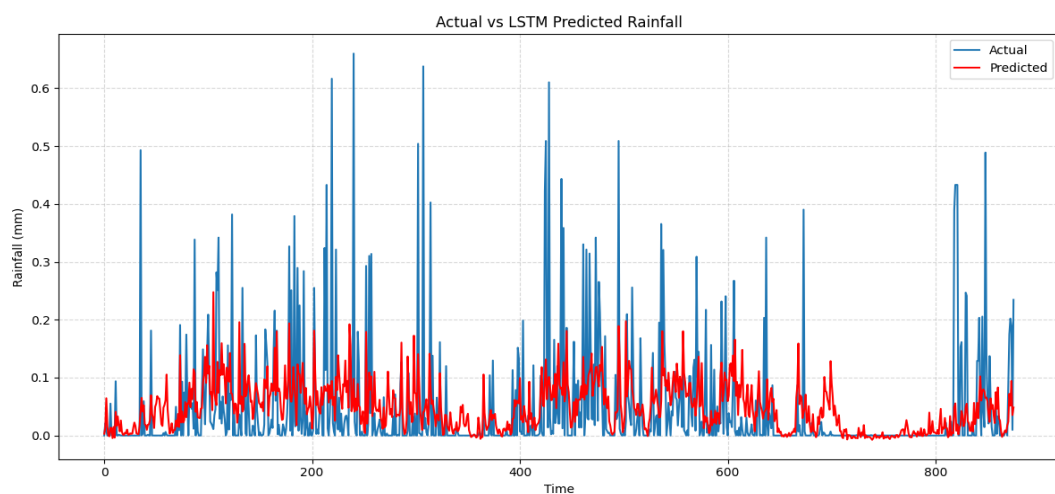
Sementara *Batch Size 64, Epoch 100*. *Train loss* menurun lebih lambat, namun *validation loss* paling stabil di antara semua konfigurasi. Menunjukkan generalisasi cukup baik meskipun kecepatan belajar lebih lambat.



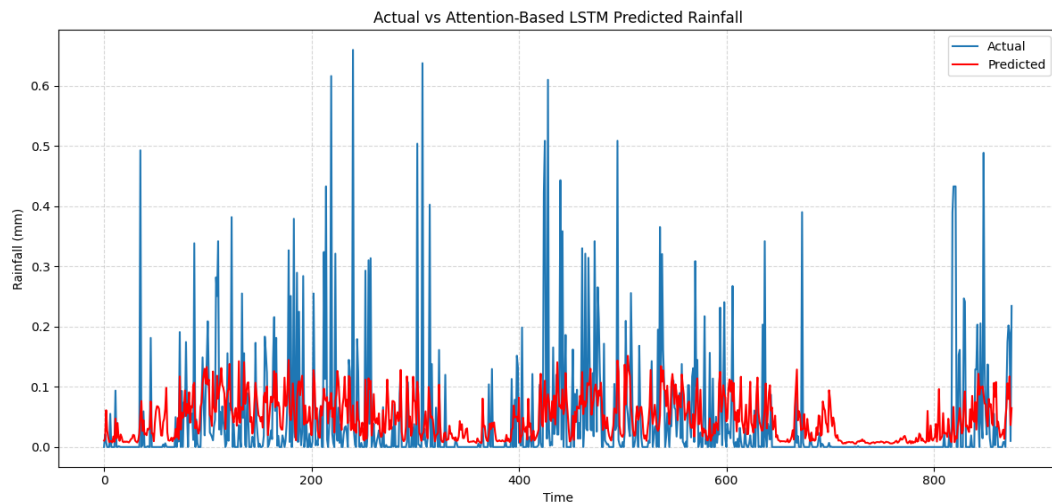
Gambar 6. 10 Kurva loss model AT-LSTM *Batch 64* dan *Epoch 100*

Secara umum, model LSTM dengan *batch* kecil (4) rentan *overfitting*, sedangkan *batch* menengah (16–32) memberikan keseimbangan terbaik antara *training loss* dan *validation loss*. Pada *batch* besar (64), meskipun penurunan *training loss* lebih lambat, generalisasi lebih baik karena *validation loss* lebih stabil. Model Attention-Based LSTM menunjukkan pola serupa, namun lebih konsisten dalam menjaga stabilitas *validation loss*. *Batch* menengah tetap menjadi konfigurasi optimal, sementara pada *batch* besar model ini lebih unggul dibanding LSTM tanpa mekanisme *Attention*.

Untuk memperjelas secara visual perbandingan, grafik pertama menampilkan performa model LSTM terbaik, sementara grafik kedua menyajikan kinerja model Attention-Based LSTM (AT-LSTM) terbaik. Visualisasi ini bertujuan untuk memudahkan pembaca dalam menilai seberapa dekat prediksi dari kedua model tersebut mengikuti pola data aktual, serta bagaimana kecenderungan tiap model dalam menangani fluktuasi atau peningkatan tajam (lonjakan) pada nilai curah hujan.



Gambar 6. 11 Grafik Actual dan Hasil Prediksi LSTM



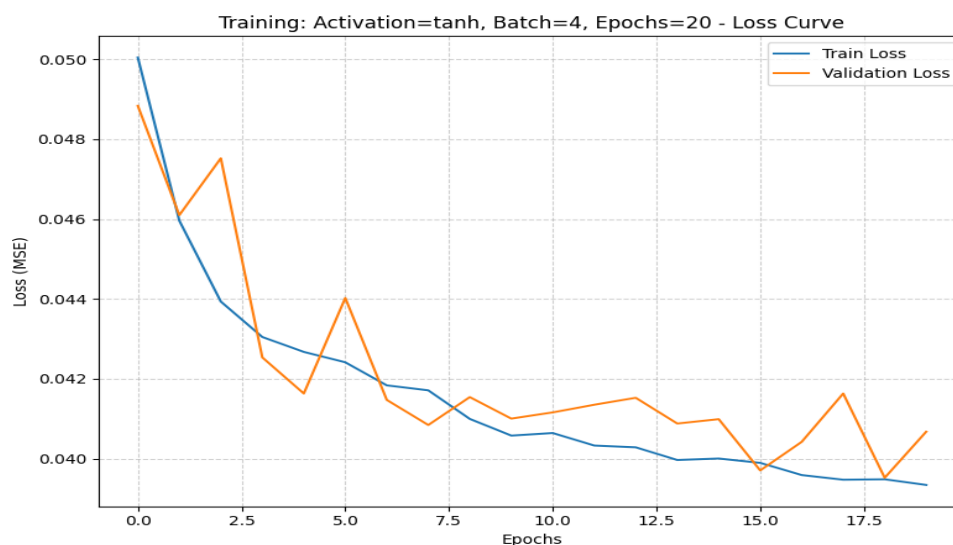
Gambar 6. 12 Grafik Actual dan Hasil Prediksi Attention-Based LSTM

Gambar 6.11 dan Gambar 6.12 menyajikan perbandingan visual antara data curah hujan sebenarnya (aktual) dan nilai yang diramalkan oleh dua model, yakni LSTM dan Attention-Based LSTM. Dari sisi kuantitatif, model Attention-Based LSTM menunjukkan kinerja yang sedikit lebih unggul dalam memprediksi data uji (Test RMSE sebesar 0.0879), dibandingkan dengan model LSTM yang mencatatkan nilai Test RMSE sebesar 0.0881. Model yang dilatih tanpa transformasi data cenderung menghasilkan prediksi yang mendekati nilai rata-rata dari data historis. Hal ini terlihat pada grafik di mana garis prediksi (merah) relatif datar dan kurang responsif terhadap lonjakan atau nilai ekstrem (puncak data) pada garis aktual (biru). Meskipun metrik RMSE mungkin menunjukkan nilai yang cukup rendah (terutama jika variasi data tidak terlalu besar), secara kualitatif model ini gagal menangkap dinamika penuh dari data deret waktu, terutama fluktuasi besar, yang dikenal sebagai masalah *underestimation* pada puncak.

6.2 Hasil Prediksi dengan Transformasi Data

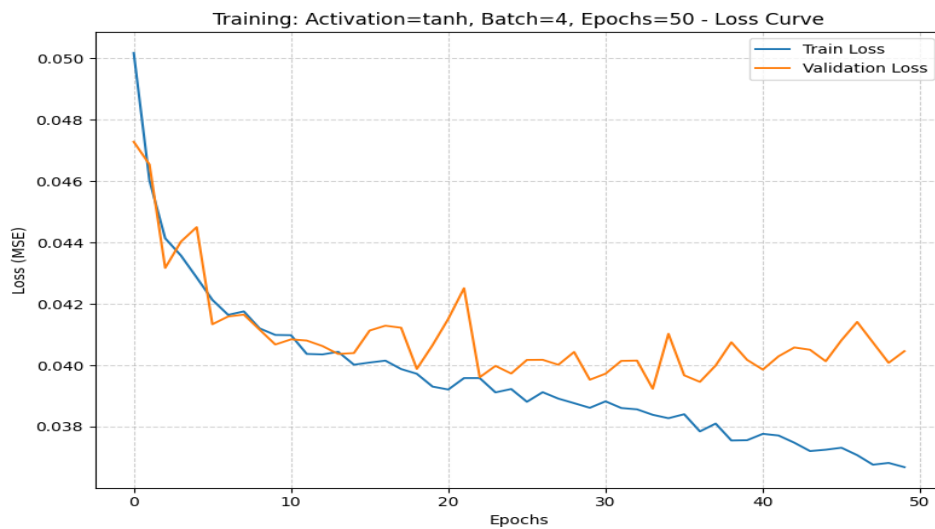
Transformasi data adalah sebuah langkah untuk mengubah skala atau distribusi data asli (mentah) menjadi format yang lebih optimal untuk keperluan analisis. Proses ini dapat dilakukan melalui berbagai metode, seperti normalisasi atau log-transformasi. Pada tahap kedua, dilakukan evaluasi model LSTM dengan menggunakan transformasi data (log-normalisasi). Hasil evaluasi ditunjukkan melalui grafik kurva *loss* dan MSE untuk lima konfigurasi terbaik. Model yang dianggap paling unggul (terbaik) diputuskan berdasarkan metrik MSE (*Mean Squared Error*) dan RMSE (*Root Mean Squared Error*) yang paling kecil yang dicapai setelah transformasi data diterapkan pada data yang digunakan untuk pengujian (data uji). yang menunjukkan kemampuan paling optimal dalam merepresentasikan pola curah hujan.

Pada konfigurasi *Batch Size* 4 dengan 20 *epoch*. Kurva menunjukkan penurunan yang baik dan stabil, namun pelatihan yang sangat singkat membuat kita tidak bisa mengevaluasi kinerja sebenarnya.



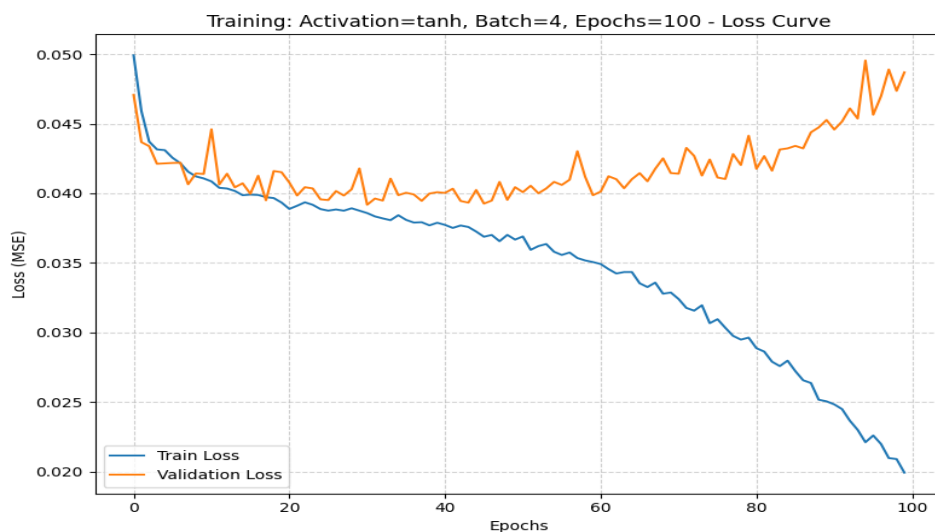
Gambar 6. 13 Kurva loss model LSTM *Batch* 4 dan *Epoch* 20 dengan Transformasi Data

Berbeda dengan itu, pada *Batch Size* 4 dengan 50 *epoch*, model ini menunjukkan tren penurunan yang baik dan kesenjangan yang sempit antara *Train Loss* dan *Validation Loss*. Namun, model ini kemungkinan besar kurang terlatih (*underfitting*) karena pelatihan dihentikan terlalu cepat pada Epoch 50.



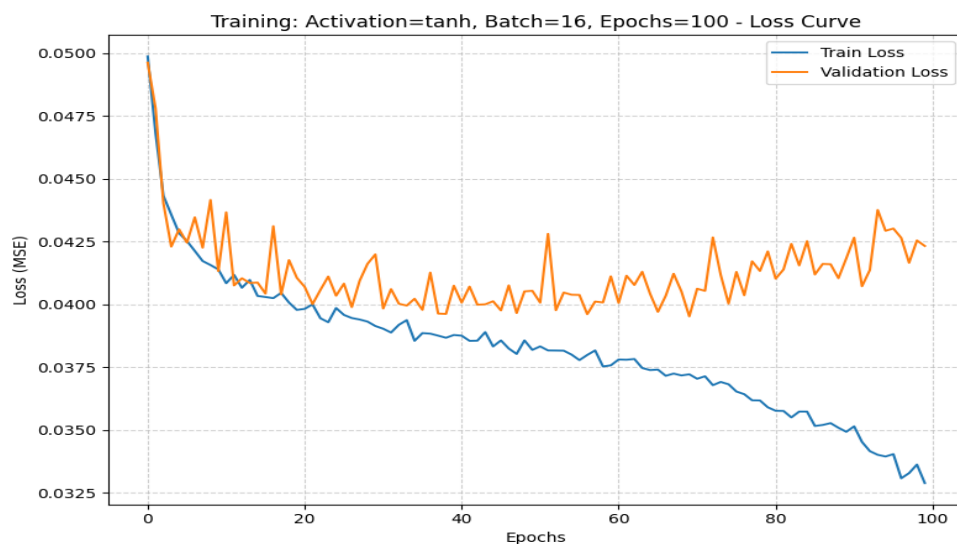
Gambar 6. 14 Kurva loss model LSTM *Batch* 4 dan *Epoch* 50 dengan Transformasi Data

Kemudian, Ketika jumlah iterasi pelatihan (*epoch*) dinaikkan menjadi 100 dengan menggunakan ukuran *batch* sebanyak 4. *Train Loss* turun sangat rendah, menunjukkan model sangat baik dalam menghafal data latih.



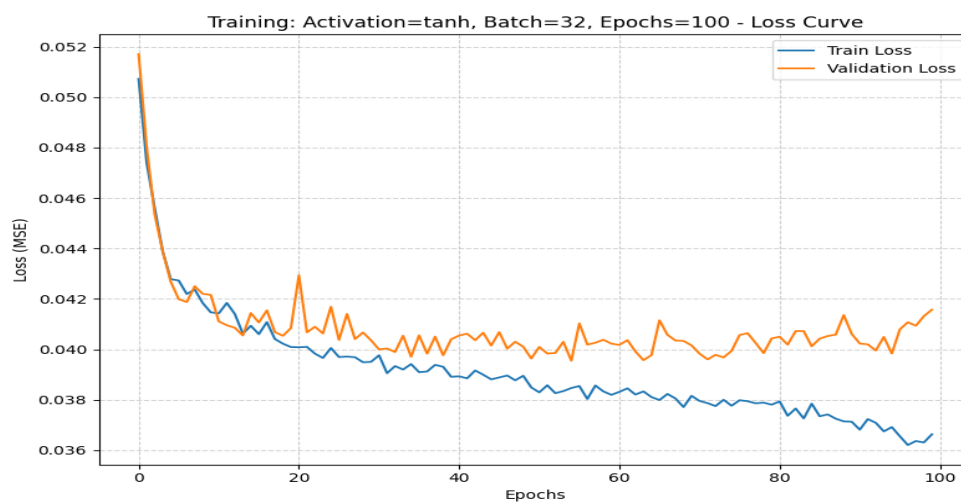
Gambar 6. 15 Kurva loss model LSTM *Batch* 4 dan *Epoch* 100 dengan Transformasi Data

Selanjutnya, pada konfigurasi *Batch Size* 16 dengan 100 *epoch*, di mana *Train Loss* berhasil ditekan hingga di bawah 0.035—nilai terendah di antara semua model.



Gambar 6. 16 Kurva loss model LSTM *Batch* 16 dan *Epoch* 100 dengan Transformasi Data

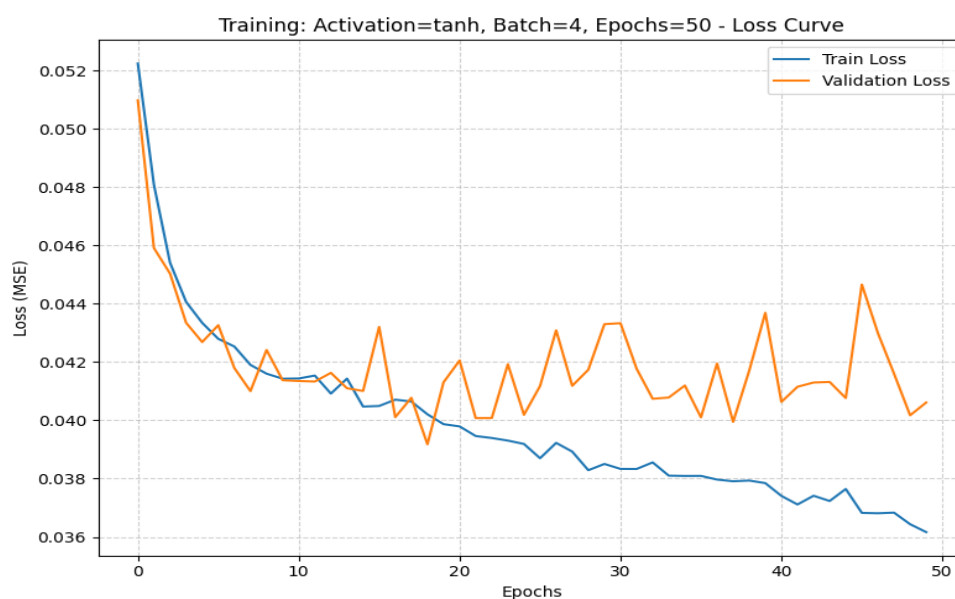
Sementara itu, pada *Batch Size* 32 dengan 100 *epoch*, *Train Loss* dan *Validation Loss*, turun secara konsisten pada tahap awal pelatihan. Setelah mencapai titik stabil (sekitar *Epoch* 20), *Validation Loss* menunjukkan stabilitas luar biasa, berfluktuasi tipis di sekitar 0.040.



Gambar 6. 17 Kurva loss model LSTM *Batch* 32 dan *Epoch* 100 dengan Transformasi Data

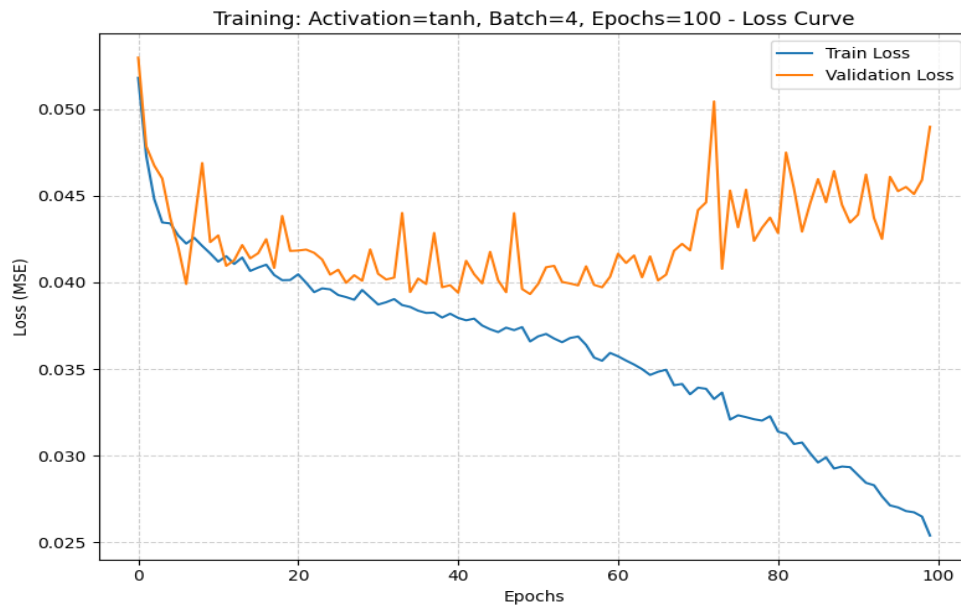
Selanjutnya, dilakukan evaluasi pada model Attention-Based LSTM (AT-LSTM) dengan menggunakan transformasi data dan konfigurasi yang sama. Hasilnya divisualisasikan dalam bentuk grafik, yang memperlihatkan performa pelatihan pada lima konfigurasi terbaik. Dibandingkan dengan LSTM tanpa mekanisme Attention, model Attention-Based LSTM secara umum menunjukkan performa yang sedikit lebih baik setelah transformasi, kinerja model diidentifikasi sebagai lebih baik melalui penurunan nilai MSE (*Mean Squared Error*) dan RMSE (*Root Mean Squared Error*) pada berbagai pengaturan parameter. Keunggulan ini didorong oleh mekanisme Attention, yang memungkinkan model mengalihkan fokus pada pola-pola paling signifikan dalam rangkaian data deret waktu. Sebagai hasilnya, model mampu menghasilkan prediksi yang lebih tepat dan lebih efektif mengikuti kecenderungan (tren) data curah hujan yang ada.

Model dengan *Batch Size* 4 dan *50 Epoch* ini menunjukkan bahwa pelatihan berhasil menekan *Train Loss* secara bertahap.



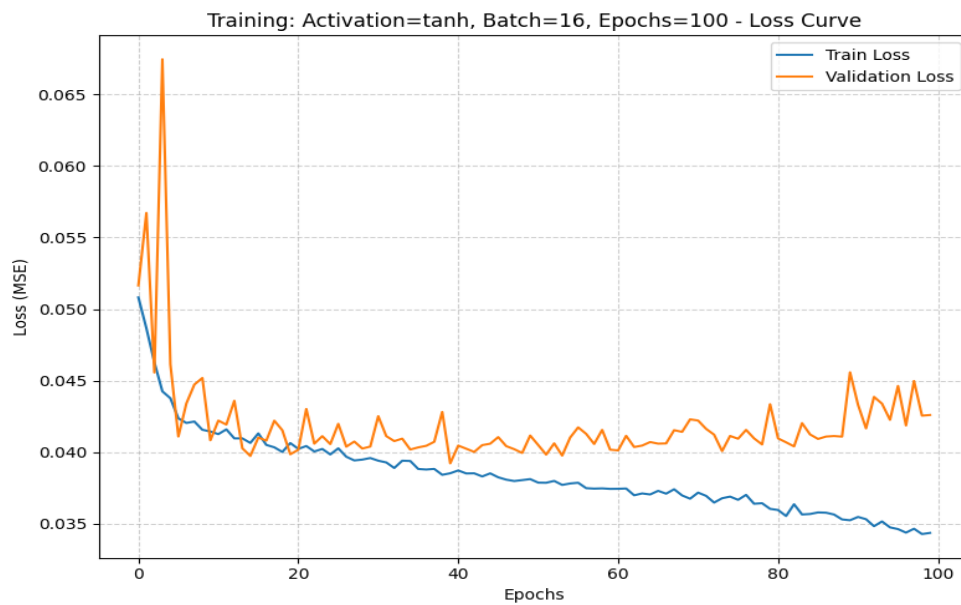
Gambar 6. 18 Kurva loss model AT-LSTM *Batch* 4 dan *Epoch* 50 dengan Transformasi Data

Kemudian pada *Batch Size 4, Epoch 100*. *Train loss* turun konsisten, tetapi *validation loss* memaksa model untuk menghafal setiap sampel data.



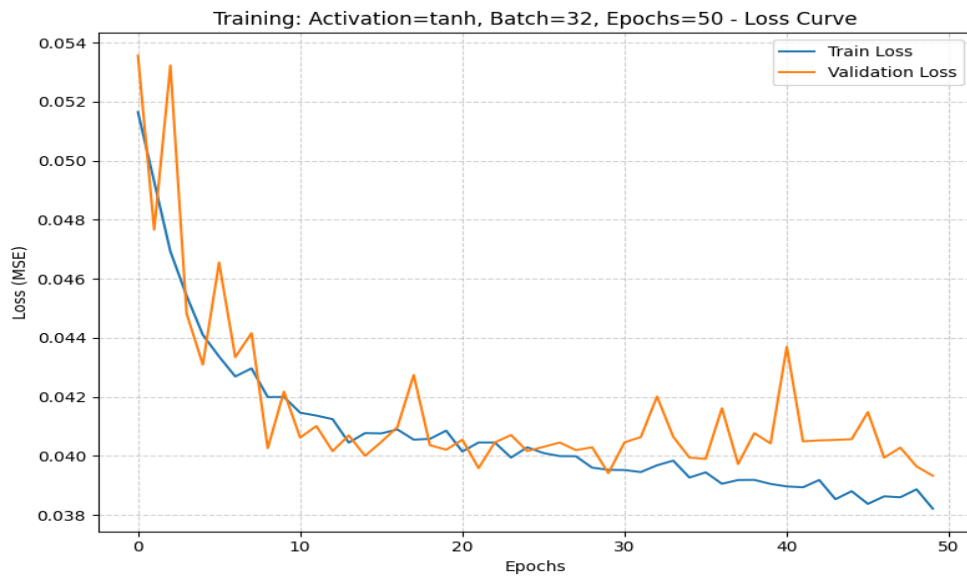
Gambar 6. 19 Kurva loss model AT-LSTM *Batch 4* dan *Epoch 100* dengan Transformasi Data

Selanjutnya, *Batch Size 16, Epoch 100*. *Validation Loss* secara konsisten naik sementara *Train Loss* terus turun.



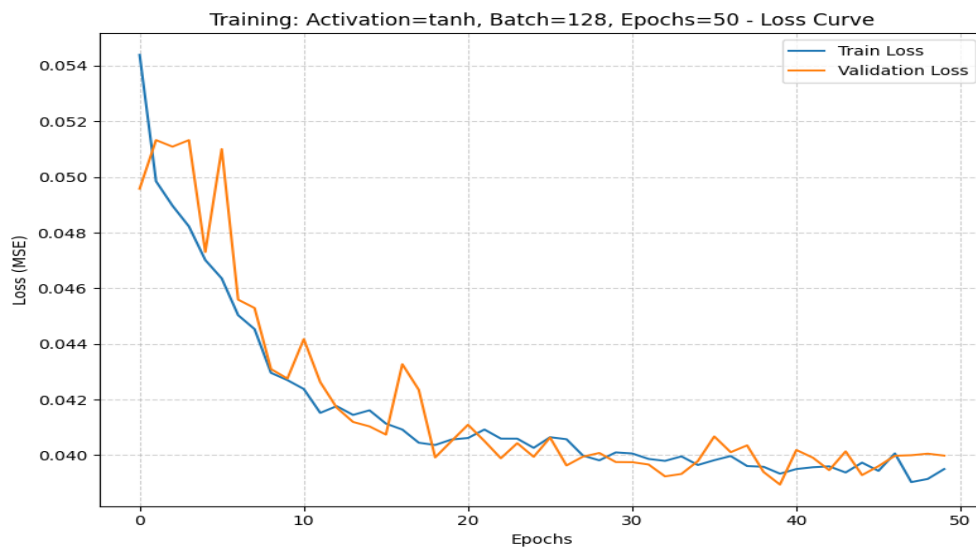
Gambar 6. 20 Kurva loss model AT-LSTM *Batch 16* dan *Epoch 100* dengan Transformasi Data

Untuk *Batch Size 32, Epoch 50*. *Train loss* konsisten turun, *validation loss* relatif stabil dan tidak naik signifikan, dan mengalami penurunan yang baik dan stabil.



Gambar 6. 21 Kurva loss model AT-LSTM *Batch 32* dan *Epoch 50* dengan Transformasi Data

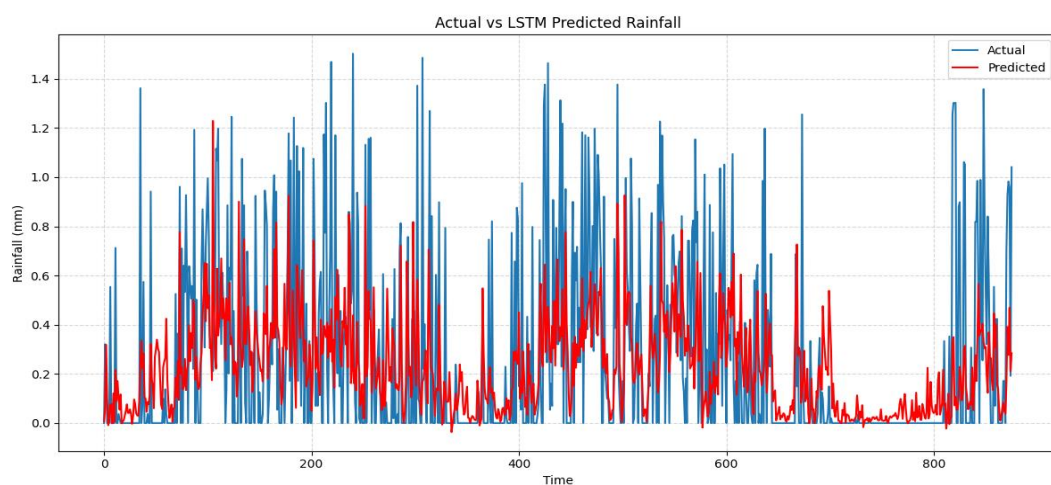
Sementara *Batch Size 64, Epoch 100*. *Train loss* dan *Validation Loss*, turun dengan cepat di awal dan model ini sangat efektif bergeneralisasi dan tidak mengalami *overfitting* sama sekali.



Gambar 6. 22 Kurva loss model AT-LSTM *Batch 128* dan *Epoch 50* dengan Transformasi Data

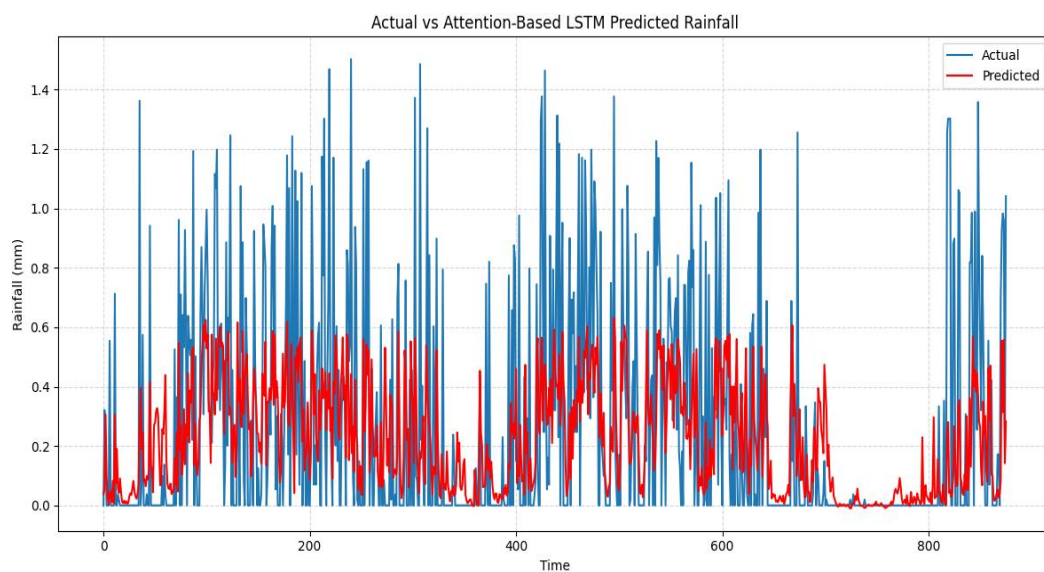
Secara keseluruhan, model AT-LSTM (*Attention-based LSTM*) menunjukkan kestabilan generalisasi yang jauh lebih unggul dibandingkan model yang berjalan tanpa mekanisme *attention*. Model tanpa *attention* terbukti rentan terhadap *overfitting*, ditunjukkan oleh kasus seperti konfigurasi *Batch* 4, *Epochs* 100 di mana *Validation Loss* melonjak drastis, membuktikan kesulitan model dalam menangani *noise* data. Sebaliknya, konfigurasi AT-LSTM terbaik, terutama pada *Batch* 128, *Epochs* 50, berhasil menekan *Train Loss* dan menjaga *Validation Loss* bergerak hampir berimpitan dan sangat stabil di sekitar 0.040 hingga akhir pelatihan, tanpa tanda-tanda *overfitting* yang ekstrem. Mekanisme *Attention* sukses memfokuskan model pada data yang paling relevan, dalam melakukan prediksi pada data curah hujan yang belum pernah dilihat sebelumnya (data baru).

Visualisasi ini bertujuan untuk membandingkan data curah hujan aktual dengan hasil prediksi dari dua model terbaik: LSTM (Grafik 1) dan Attention-Based LSTM (AT-LSTM, Grafik 2). Perbandingan ini penting untuk mengevaluasi secara visual seberapa akurat kedua model tersebut mampu mengikuti pola data sebenarnya, terutama dalam merespons fluktuasi atau lonjakan nilai curah hujan.



Gambar 6. 23 Grafik Actual dan Hasil Prediksi LSTM dengan Transformasi Data

Gambar 6.23 menyajikan perbandingan akhir antara data aktual dan prediksi terbaik model LSTM setelah data dikembalikan ke skala semula (melalui anti-log dari transformasi log). Hasil evaluasi menunjukkan nilai RMSE akhir sebesar 0.3090. Secara umum, prediksi model berhasil mengikuti tren data asli, meskipun masih terdapat perbedaan (deviasi) signifikan pada titik-titik dengan nilai ekstrem.



Gambar 6. 24 Grafik Actual dan Hasil Prediksi Attention-Based LSTM dengan Transformasi Data

Sementara itu, Gambar 6.24 menampilkan hasil prediksi dari model Attention-Based LSTM (AT-LSTM) dengan prosedur transformasi data yang sama. Pola prediksi AT-LSTM terlihat lebih halus dan lebih mendekati data aktual dibandingkan model *LSTM tanpa mekanisme Attention*. Nilai RMSE sebesar 0.3068 menunjukkan performa yang sedikit lebih baik, menandakan bahwa mekanisme *Attention* secara efektif meningkatkan kapabilitas model dalam mengidentifikasi pola-pola krusial dalam data deret waktu, sekaligus mempertahankan konsistensi (stabilitas) dari hasil prediksi.

6.3 Perbandingan Tanpa Transformasi Data vs dengan Transformasi Data

Perbandingan dilakukan untuk mengevaluasi pengaruh penerapan log-transformasi terhadap kinerja model dalam memprediksi curah hujan. Dua skenario diuji, yaitu: (1) model tanpa transformasi data, dan (2) model dengan transformasi log pada data target (RR), dikembalikan ke skala aslinya (*anti-log*). Tujuan utama dari penelitian ini adalah mengevaluasi sejauh mana transformasi data mampu mengoptimalkan kemampuan model dalam mengidentifikasi pola data yang sangat fluktuatif dan ekstrem. Namun, berdasarkan hasil evaluasi, model yang tidak menerapkan transformasi justru mencatatkan nilai *error* RMSE (*Root Mean Square Error*) yang secara numerik lebih rendah, dikarenakan model tersebut beroperasi langsung menggunakan skala data curah hujan aslinya. Namun, model ini cenderung kurang mampu mengikuti pola ekstrem pada data aktual, terutama ketika terjadi lonjakan curah hujan yang tinggi (*spike anomaly*).

Tabel 6. 1 Perbandingan Model Tanpa vs dengan Transformasi Data

Model	Skala Evaluasi	Test RMSE	Karakteristik Prediksi
LSTM	Tanpa Transformasi	0.0881	Pola cukup baik, namun lonjakan curah hujan masih ekstrem.
AT-LSTM	Tanpa Transformasi	0.0879	<i>Error</i> kecil, namun pola prediksi kurang menangkap lonjakan data aktual.
LSTM	Dengan Transformasi Data	0.3090	Pola mengikuti tren aktual dan prediksi sudah baik.
AT-LSTM	Dengan Transformasi Data	0.3068	Hasil lebih stabil dibanding LSTM tanpa <i>Attention</i> , dan pola prediksi paling konsisten dan mengikuti data aktual secara menyeluruh.

Dari tabel di atas terlihat bahwa nilai *error* RMSE model Attention-Based LSTM (0.3068) sedikit lebih rendah dibandingkan model LSTM (0.3090), yang menandakan bahwa penerapan mekanisme *Attention* mampu meningkatkan kemampuan model dalam mengenali hubungan jangka panjang dan menangkap fluktuasi data curah hujan secara lebih akurat. Dengan demikian, dapat disimpulkan bahwa penerapan transformasi log efektif dalam menstabilkan data dan meningkatkan kemampuan model dalam mempelajari pola yang kompleks pada data curah hujan.

6.4 Pembahasan Hasil Model LSTM

Model LSTM menunjukkan nilai RMSE (*Root Mean Square Error*) yang rendah, yang mengindikasikan kemampuan prediksi curah hujan yang cukup baik. Rendahnya RMSE ini menandakan tingkat akurasi estimasi yang memadai, sebab selisih rata-rata akar kuadrat antara nilai prediksi dan nilai aktual tidak terlalu besar. Selain itu, tabel metrik untuk 5 kombinasi *batch size* dan *epoch* terbaik menegaskan kesamaan pola antara hasil prediksi dan data aktual, meskipun pada beberapa titik, prediksi model belum sepenuhnya presisi. Hal ini mungkin disebabkan oleh variabilitas data curah hujan yang kompleks atau keterbatasan model dalam menangkap semua pola dan anomali di dalam data.

Tabel 6. 2 Hasil metrik terbaik LSTM

<i>Batch Size</i>	<i>Epochs</i>	<i>Training RMSE</i>	<i>Validation RMSE</i>	<i>Testing RMSE</i>
4	100	0.1912	0.1979	0.2085
4	50	0.1911	0.1981	0.2069
16	100	0.1893	0.1988	0.2118

Lanjutan				
4	20	0.1941	0.1988	0.2113
32	100	0.1934	0.1989	0.2087

6.5 Pembahasan Hasil Model Attention-Based LSTM

Model Attention-Based LSTM (AT-LSTM) menunjukkan kinerja prediksi yang superior dibandingkan model LSTM biasa, terbukti dari nilai RMSE (Root Mean Square Error) yang lebih rendah. Penurunan RMSE ini mengindikasikan kemampuan yang lebih baik dalam memperkecil selisih antara nilai peramalan dan nilai data aktual, yang pada akhirnya menegaskan bahwa AT-LSTM lebih akurat dalam melakukan estimasi, dengan deviasi prediksi terhadap nilai aktual yang lebih kecil. Tabel hasil metrik 5 *batch size* dan *epochs* terbaik dari LSTM dengan *Attention* juga menunjukkan kesesuaian yang lebih baik dengan nilai aktual, Hasil pengujian mengonfirmasi bahwa model Attention-Based LSTM (AT-LSTM) memiliki kinerja superior dalam meramalkan curah hujan dibandingkan dengan model LSTM tanpa mekanisme Attention.

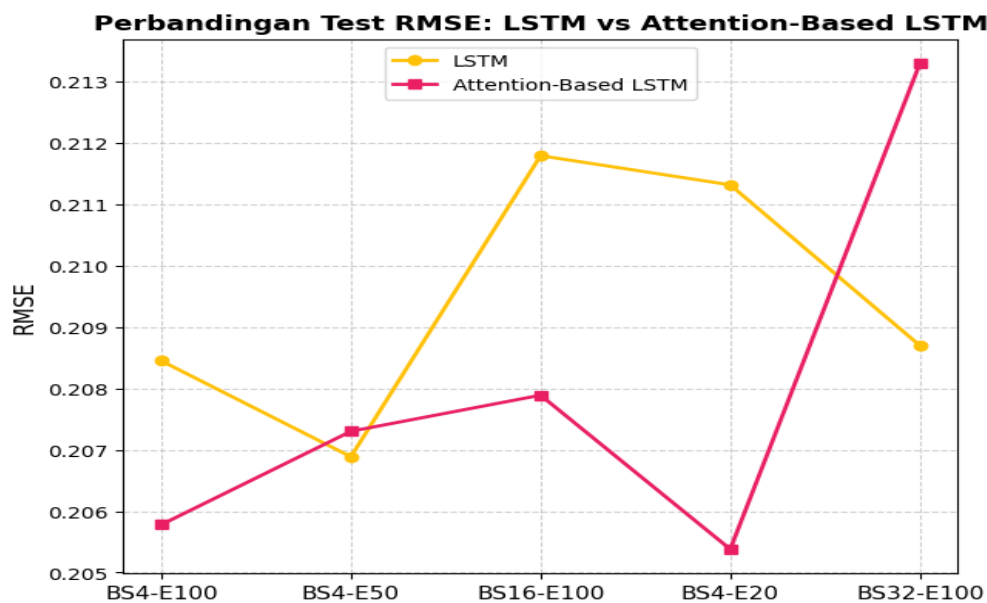
Tabel 6. 3 Hasil metrik terbaik Attention-Based LSTM

<i>Batch Size</i>	<i>Epochs</i>	<i>Training RMSE</i>	<i>Validation RMSE</i>	<i>Testing RMSE</i>
128	50	0.1941	0.1973	0.2058
4	50	0.1937	0.1979	0.2073
16	100	0.1937	0.1981	0.2079
32	50	0.1921	0.1983	0.2054
4	100	0.1863	0.1983	0.2133

6.6 Perbandingan LSTM vs Attention-Based LSTM

Berdasarkan Tabel Perbandingan Nilai RMSE antara model LSTM dan Attention-Based LSTM, tampak bahwa nilai *training* RMSE untuk kedua model relatif rendah dan konsisten. Namun, terdapat sedikit perbedaan dalam nilai validasi, yang dapat menunjukkan kecenderungan terhadap fenomena *overfitting*. Dalam kasus ini, model Attention-Based LSTM memiliki *training* RMSE yang sedikit lebih rendah dibandingkan LSTM, mengindikasikan kemampuan model dalam mempelajari pola data pelatihan secara lebih mendalam. Seperti yang dijelaskan oleh Maknunah *et al.*, (2022), *overfitting* sering terjadi dalam model *supervised learning* ketika jaringan terlalu cepat mengalami konvergensi. Kecenderungan model untuk menyesuaikan parameternya secara berlebihan pada data pelatihan (training data) menyebabkan berkurangnya fleksibilitas saat menghadapi data baru. Situasi ini disebut *overfitting*, yaitu ketika model terlalu menghafal data latih sehingga kehilangan kemampuan untuk mengenali pola umum dalam data. Menurut Carnegie & Chairani (2023), gejala *overfitting* dapat dikenali ketika RMSE (*Root Mean Square Error*) pada data pelatihan lebih rendah daripada RMSE pada data validasi, sebuah kondisi yang teramati pada tabel yang disajikan. Walaupun selisih nilainya kecil, pengontrolan arsitektur dan proses pelatihan tetap krusial untuk menjaga keseimbangan antara akurasi pelatihan dan validasi. Oleh karena itu, memantau perbandingan RMSE antara *set training* dan *validation* sangat penting dilakukan untuk mencegah *overfitting* dan menjamin bahwa model memiliki kemampuan generalisasi yang optimal terhadap data yang belum pernah

diproses. Perbandingan RMSE yang lebih detail dari kedua metode dapat dilihat pada Gambar 6.25.



Gambar 6. 25 Grafik Perbandingan RMSE

Gambar 6.25 menyajikan perbandingan nilai *Root Mean Squared Error* (RMSE) antara model LSTM dan Attention-Based LSTM (AT-LSTM), di mana kedua model menunjukkan pola kesalahan yang relatif serupa. Secara keseluruhan, model AT-LSTM menunjukkan nilai RMSE yang lebih rendah dan karenanya lebih unggul, terutama pada tiga kombinasi parameter spesifik: *Batch Size* 4 dengan 100 *Epoch* (BS4-E100), *Batch Size* 4 dengan 50 *Epoch* (BS4-E50), dan *Batch Size* 4 dengan 20 *Epoch* (BS4-E20). Hasil ini memperlihatkan bahwa integrasi mekanisme *Attention* membantu model dalam memahami hubungan jangka panjang antar waktu (*long-term dependencies*) dengan lebih efektif, sehingga mampu mengurangi tingkat kesalahan prediksi.

Evaluasi model menunjukkan bahwa baik LSTM maupun Attention-Based LSTM (AT-LSTM) memiliki kemampuan prediksi yang baik setelah proses *invers*

transform (anti-log). Secara terperinci, model LSTM mencatatkan nilai *Root Mean Squared Error* (RMSE) sebesar 0.3090 saat dievaluasi menggunakan data pengujian (data uji). Sementara itu, model AT-LSTM menghasilkan nilai *error* yang sedikit lebih rendah, yaitu RMSE 0.3068. Secara keseluruhan, Attention-Based LSTM menunjukkan peningkatan performa yang konsisten dibandingkan LSTM dasar. Meskipun selisihnya relatif kecil, mekanisme Attention terbukti mampu menurunkan nilai RMSE sebesar 0.71%. Penurunan *error* ini ditemukan konsisten di seluruh kombinasi *batch size* dan *epoch* yang diuji. Konsistensi ini menegaskan bahwa mekanisme Attention berhasil membantu LSTM dalam menyoroti dan memprioritaskan informasi temporal yang paling relevan dalam urutan data cuaca. Oleh karena itu, penggabungan mekanisme *Attention* ke dalam arsitektur LSTM telah terbukti efektif dalam meningkatkan kapabilitas model untuk melakukan peramalan curah hujan dengan hasil yang lebih akurat dan lebih stabil.

6.7 Prediksi Curah Hujan dalam Pandangan Islam

Hujan merupakan salah satu fenomena alam yang dipandang sebagai nikmat besar dalam ajaran Islam karena banyak disebut dalam Al-Qur'an. Para ilmuwan mengkaji hujan sebagai bagian dari siklus hidrologi, namun sebelum ilmu *modern* menjelaskannya, Al-Qur'an telah lebih dahulu memaparkan proses tersebut secara jelas. Fenomena hujan menjadi bukti kebesaran Allah (*Mu'amalah Ma'a Allah*) dalam mengatur alam semesta, sekaligus menunjukkan hubungan manusia dengan alam (*Mu'amalah Ma'a al-alam*) yang harus disyukuri dan dikelola dengan bijak. Hal ini tergambar dalam firman Allah dalam Surat Al-A'raf Ayat 57:

وَهُوَ الَّذِي يُرْسِلُ الرِّيَّاحَ بُشْرًا بَيْنَ يَدَيْ رَحْمَتِهِ حَتَّىٰ إِذَا أَقْلَّتْ سَحَابًا ثِقَالًا سُقِّتَاهُ لِبَلَدٍ مَّيِّتٍ فَأَنْزَلْنَا بِهِ
الْمَاءَ فَأَخْرَجْنَا بِهِ مِنْ كُلِّ الثَّمَرَاتِ ۚ كَذَٰلِكَ نُخْرِجُ الْمَوْتَىٰ لَعَلَّكُمْ تَذَكَّرُونَ

Artinya: "Dialah yang mengirimkan angin sebagai pembawa kabar gembira mendahului rahmat-Nya (hujan). Hingga apabila angin itu membawa awan yang berat, Kami halau ke negeri yang tandus, lalu Kami turunkan hujan di daerah itu, dan Kami tumbuhkan dengan hujan itu berbagai macam buah-buahan. Seperti itulah Kami membangkitkan orang-orang mati, mudah-mudahan kamu mengambil pelajaran."

Ayat ini menjelaskan runtutan proses turunnya hujan: Allah menggerakkan angin, mengumpulkan awan tebal, kemudian menurunkan hujan sehingga bumi yang mati menjadi hidup kembali. Menurut Ibn Katsir (*Tafsir al-Qur'an al-Azhim*), ayat ini menunjukkan bahwa angin berfungsi sebagai pembawa kabar gembira akan datangnya rahmat Allah berupa hujan. Angin mengumpulkan awan-awan tebal hingga digiring ke daerah yang tandus, lalu hujan diturunkan sebagai bentuk kasih sayang dan tanda kekuasaan Allah dalam menghidupkan bumi yang sebelumnya gersang. Dalam ayat ini menekankan bahwa setiap unsur dalam proses turunnya hujan, angin, awan, hingga air adalah bagian dari *sunnatullah* dalam pengaturan alam. Manusia diperintahkan untuk memahami, mensyukuri, dan mengelola nikmat tersebut dengan bijaksana sebagai bentuk penghambaan kepada Allah (Ibn Katsir; Shihab, 2001).

Surat An-Nahl Ayat 10, Allah memberikan berbagai nikmat Allah (*Mu'amalah Ma'a Allah*) & *Mu'amalah Ma'a al-alam* yang di anugerahkan kepada manusia. Allah *Subhanahu wa Ta'ala* berfirman:

هُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً ۖ لَكُمْ مِنْهُ شَرَابٌ وَمِنْهُ شَجَرٌ فِيهِ تُسِيمُونَ

Artinya: “Dialah yang menurunkan air dari langit untuk kamu; sebagiannya menjadi minuman dan sebagiannya menyuburkan tumbuh-tumbuhan, yang pada tempat tumbuhnya kamu menggembalakan ternakmu.”

Ayat ini menegaskan bahwa air hujan adalah nikmat langsung maupun tidak langsung bagi manusia. Air menjadi minuman, penyubur tumbuhan, dan penopang ekosistem. Hal ini mencerminkan hubungan manusia dengan nikmat Allah (*Mu'amalah Ma'a Allah*) sekaligus hubungannya dengan alam (*Mu'amalah Ma'a al-alam*). Air hujan memberikan manfaat langsung berupa penyejukan udara yang panas, yang menyegarkan tubuh. Sementara itu, nikmat tidak langsung dari air hujan adalah kemampuannya untuk mengairi lahan pertanian dan menjadi sumber kehidupan bagi segala jenis tumbuh-tumbuhan (Rohmawati & Alim, 2024).

Fenomena hujan dalam Islam bukan sekadar proses alam, tetapi merupakan bukti langsung dari kebesaran Allah (*Mu'amalah Ma'a Allah*), yang dijelaskan dalam Surat An-Nur Ayat 43. Allah *Subhanahu wa Ta'ala* berfirman:

أَلَمْ تَرَ أَنَّ اللَّهَ يَرْزُقُ سَحَابًا ثُمَّ يُؤَلِّفُ بَيْنَهُ ثُمَّ يَجْعَلُهُ رُكَامًا فَتَرَى الْوَدْقَ يَخْرُجُ مِنْ خِلَالِهِ وَيُنَزِّلُ مِنَ السَّمَاءِ مِنْ جِبَالٍ فِيهَا مِنْ بَرَدٍ فَيُصِيبُ بِهِ مَنْ يَشَاءُ وَيَصْرِفُهُ عَنِ مَنْ يَشَاءُ سَنًا بَرْقًا يَذْهَبُ بِالْأَبْصَارِ

Artinya: “Tidakkah kamu melihat bahwa Allah mengarak awan, kemudian mengumpulkan antara (bagian-bagian)nya, kemudian menjadikannya bertindih-tindih, maka kelihatanlah olehmu hujan keluar dari celah-celahnya dan Allah (juga) menurunkan (butiran-butiran) es dari langit, (yaitu) dari (gumpalan-gumpalan awan seperti) gunung-gunung, maka ditimpakan-Nya (butiran-butiran) es itu kepada siapa yang dikehendaki-Nya dan dipalingkan-Nya dari siapa yang dikehendaki-Nya. Kilauan kilat awan itu hampir-hampir menghilangkan penglihatan.”

Ayat ini memberikan gambaran kuat mengenai bagaimana Allah mengatur awan, mengumpulkan, menumpuk, hingga menurunkan hujan. Menurut Tafsir Ibn Katsir, ayat ini menunjukkan bahwa Allah-lah yang mengatur seluruh fase terbentuknya awan hingga turunnya hujan sebagai bentuk kekuasaan-Nya dalam menghidupkan bumi dan memberi manfaat bagi makhluk hidup. Sementara itu, Quraish Shihab dalam Tafsir al-Mishbah menjelaskan bahwa ayat ini mengandung pesan bahwa fenomena alam tidak terjadi secara kebetulan, tetapi mengikuti sunnatullah yang teratur (Ibn Katsir; Shihab, 2001).

Curah hujan merupakan fenomena di dunia. Karena kemunculannya, maka akan menjadi sumber kehidupan sebagaimana dijelaskan dalam Surat Al-Furqan ayat 48-50, yang menunjukkan hubungan antara *Mu'amalah Ma'a Allah*, *Mu'amalah Ma'a al-alam* & *Mu'amalah Ma'a an-nas*. Allah *Subhanahu wa Ta'ala* berfirman:

وَهُوَ الَّذِي أَرْسَلَ الرِّيحَ بُشْرًا بَيْنَ يَدَيْ رَحْمَتِهِ وَأَنْزَلْنَا مِنَ السَّمَاءِ مَاءً طَهُورًا لِّنُحْيِيَ بِهِ بَلْدَةً مَّيْتًا وَنُسْقِيَهُ مِمَّا خَلَقْنَا أَنْعَامًا وَأَنَاسِيَّ كَثِيرًا وَلَقَدْ صَرَّفْنَاهُ بَيْنَهُمْ لِيَذَكَّرُوا فَأَبَى أَكْثَرُ النَّاسِ إِلَّا كُفُورًا

Artinya: “Dan Dialah yang meniupkan angin (sebagai) pembawa kabar gembira sebelum kedatangan rahmat-Nya (hujan); dan Kami turunkan dari langit air yang sangat bersih. Agar (dengan air itu) Kami menghidupkan negeri yang mati (tandus), dan Kami memberi minum kepada sebagian apa yang telah Kami ciptakan, (berupa) hewan-hewan ternak dan manusia yang banyak. Dan sungguh, Kami telah mempergilirkan (hujan) itu di antara mereka agar mereka mengambil pelajaran; tetapi kebanyakan manusia tidak mau (bersyukur), bahkan mereka mengingkari (nikmat).”

Dalam ayat tersebut, Allah *Subhanahu wa Ta'ala* menerangkan bahwa hujan adalah sarana pemulih kehidupan: menyuburkan tanah, memberi minum ternak, dan

memenuhi kebutuhan manusia. Allah memperlirirkan turunnya hujan sebagai bentuk ujian syukur bagi manusia. Nilai *Mu'amalah Ma'a an-nas* tampak karena manfaat hujan berhubungan langsung dengan kehidupan sosial manusia, mulai dari pertanian, peternakan, hingga kebutuhan air sehari-hari. Menurut Tafsir Ibn Katsir, ayat ini menunjukkan bahwa Allah menurunkan hujan sebagai rahmat sekaligus ujian, karena melalui air itulah bumi yang mati menjadi hidup kembali, dan manusia memperoleh berbagai manfaat yang tidak terhitung jumlahnya. Ayat ini menegaskan tiga hubungan utama manusia: hubungan dengan Allah (*Mu'amalah Ma'a Allah*) karena hujan adalah bagian dari rahmat-Nya; hubungan dengan alam (*Mu'amalah Ma'a al-alam*) karena manusia harus mengelola sumber daya air; serta hubungan antar manusia (*Mu'amalah Ma'a an-nas*) karena hujan menentukan keberlangsungan kehidupan sosial dan ekonomi masyarakat (Ibn Katsir; Shihab, 2001).

Dan Surat Al-Hijr Ayat 22, yang membuktikan hubungan antara *Mu'amalah Ma'a Allah & Mu'amalah Ma'a al-alam*. Allah *Subhanahu wa Ta'ala* berfirman:

وَأَرْسَلْنَا الرِّيَّاحَ لَوَاقِحَ فَأَنْزَلْنَا مِنَ السَّمَاءِ مَاءً فَأَسْقَيْنَاكُمُوهُ وَمَا أَنْتُمْ لَهُ بِخَازِنِينَ

Artinya: “Dan Kami telah meniupkan angin untuk mengawinkan (tumbuh-tumbuhan), lalu Kami turunkan hujan dari langit, kemudian Kami beri minum kamu dengan air itu, dan kamu bukanlah orang yang menyimpannya.”

Ayat ini menjelaskan peran angin sebagai “penyerbuk” atau pemicu proses terjadinya awan dan turunnya hujan. Allah memperjelas bahwa sistem atmosfer bekerja berdasarkan *sunnatullah* aturan yang telah ditetapkan-Nya. Melalui ayat ini, manusia diajak memahami bahwa fenomena alam seperti angin, pembentukan

awan, dan turunnya hujan terjadi karena ketetapan Allah dan menjadi bukti keteraturan ciptaan-Nya. Ibn Katsir menyebut bahwa angin yang mengawinkan adalah angin yang membawa uap air dan menyebabkan terjadinya awan hingga hujan turun, sedangkan Quraish Shihab menegaskan bahwa ayat ini menggambarkan harmoni antara hukum alam dan kehendak Allah sebagai bentuk pengaturan-Nya terhadap alam semesta (Ibn Katsir; Shihab, 2001).

Kemudian dalam Ayat Surat Ar-Ra'd Ayat 17, tentang kekuasaan Allah *Mu'amalah Ma'a Allah* (Perumpamaan Hak & Batil). Allah *Subhanahu wa Ta'ala* berfirman:

أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَسَالَتْ أَوْدِيَةٌ بِقَدَرِهَا فَاحْتَمَلَ السَّيْلُ زَبَدًا رَابِيًا وَمِمَّا يُوقِدُونَ عَلَيْهِ فِي النَّارِ
ابْتِغَاءَ حِلْيَةٍ أَوْ مَتَاعٍ زَبَدٌ مِثْلَهُ كَذَلِكَ يَضْرِبُ اللَّهُ الْحَقَّ وَالْبَاطِلَ ۚ فَأَمَّا الزَّبَدُ فَيَذْهَبُ جُفَاءً وَأَمَّا مَا يَنْفَعُ
النَّاسَ فَيَمْكُثُ فِي الْأَرْضِ ۚ كَذَلِكَ يَضْرِبُ اللَّهُ الْأَمْثَالَ

Artinya: "Dia telah menurunkan air dari langit, lalu mengalirlah air itu di lembah-lembah sesuai dengan ukurannya. Arus itu membawa buih yang mengambang. Dari apa (logam) yang mereka lebur dalam api untuk membuat perhiasan atau alat-alat, ada (pula) buih seperti (buih arus) itu. Demikianlah Allah membuat perumpamaan tentang hak dan batil. Buih akan hilang tidak berguna, sedangkan yang bermanfaat bagi manusia akan menetap di dalam bumi. Demikianlah Allah membuat perumpamaan."

Ayat ini menguraikan bagaimana Allah menurunkan air hujan dari langit, dan kemudian air tersebut mengalir melalui lembah-lembah dengan volume atau daya tampung (kapasitas) masing-masing lembah. Fenomena ini menunjukkan ketelitian Allah dalam menetapkan takaran (*qadar*) bagi setiap tempat. Allah kemudian menjadikan aliran air tersebut sebagai perumpamaan antara hak dan batil: buih yang mengambang di permukaan diibaratkan sebagai kebatilan yang akan lenyap,

sedangkan air yang bermanfaat tetap tinggal dan memberi kehidupan bagi manusia. Perumpamaan ini menegaskan kekuasaan Allah (*Mu'amalah Ma'a Allah*) dalam mengatur fenomena alam sekaligus mengajarkan hikmah moral dan spiritual melalui kejadian sehari-hari yang dekat dengan manusia. Penjelasan mengenai makna perumpamaan ini dijelaskan oleh Ibn Katsir dalam *Tafsir al-Qur'an al-'Azhim* dan diperkuat oleh Quraish Shihab dalam *Tafsir al-Mishbah* (Ibn Katsir; Shihab, 2001).

Surat Ar-Rum Ayat 48, menjelaskan hubungan antara *Mu'amalah Ma'a Allah* dan *Mu'amalah Ma'a al-alam*. Allah *Subhanahu wa Ta'ala* berfirman:

اللَّهُ الَّذِي يُرْسِلُ الرِّيَّاحَ فَتُثِيرُ سَحَابًا فَيَبْسُطُهُ فِي السَّمَاءِ كَيْفَ يَشَاءُ وَيَجْعَلُهُ كِسْفًا فَتَرَى الْوَدْقَ يَخْرُجُ
مِنْ خَلَالِهِ فَإِذَا أَصَابَ بِهِ مَنْ يَشَاءُ مِنْ عِبَادِهِ إِذَا هُمْ يَسْتَبْشِرُونَ

Artinya: "Allah-lah yang mengirimkan angin, lalu angin itu menggerakkan awan, kemudian Dia membentangkannya di langit menurut yang Dia kehendaki, dan menjadikannya bergumpal-gumpal, lalu engkau lihat hujan keluar dari celah-celahnya. Maka apabila Dia menurunkannya kepada hamba-hamba-Nya yang Dia kehendaki, tiba-tiba mereka bergembira."

Ayat ini menegaskan bahwa angin, awan, dan hujan bekerja sesuai hukum alam yang Allah tetapkan. Tafsir Ibn Katsir menjelaskan bahwa Allah menggerakkan angin untuk menerbangkan awan, lalu mengumpulkannya, hingga menurunkan hujan sebagai rahmat bagi hamba-Nya. Quraish Shihab menjelaskan bahwa perubahan bentuk awan dan keluarnya hujan merupakan *sunnatullah* yang dapat dipelajari, sehingga manusia terdorong untuk memahami fenomena alam tersebut. (Ibn Katsir; Shihab, 2001).

Selain ayat yang menjelaskan proses turunnya hujan, penelitian ini juga menyoroti konsep prediksi dan perencanaan dalam perspektif Islam yang berkaitan dengan *Mu'amalah Ma'a Allah* dan *Mu'amalah Ma'a an-nas*. Hal ini ditegaskan dalam firman Allah dalam Surat Al-Mulk ayat 15:

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَلْتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِغَدٍ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ خَبِيرٌ بِمَا تَعْمَلُونَ

Artinya: *“Wahai orang-orang yang beriman, bertakwalah kepada Allah dan hendaklah setiap orang memperhatikan apa yang telah diperbuatnya untuk hari esok (akhirat). Bertakwalah kepada Allah. Sesungguhnya Allah Mahateliti terhadap apa yang kamu kerjakan.”*

Adapun hadis terkait ayat tersebut merujuk pada sabda Nabi Muhammad SAW:

إِنَّ اللَّهَ لَا يَغَيِّرُ مَا بِقَوْمٍ حَتَّى يُغَيِّرُوا مَا بِأَنْفُسِهِمْ

Artinya: *"Sesungguhnya Allah SWT tidak akan mengubah keadaan suatu kaum hingga mereka mengubah keadaan yang ada pada diri mereka sendiri."* (HR. Al-Rahawi).

Dalam konteks prediksi curah hujan, hadis ini memberikan dasar moral bahwa penggunaan ilmu pengetahuan, teknologi prediksi curah hujan, pemodelan seperti LSTM dan Attention-Based LSTM, serta perencanaan pengelolaan air, merupakan bagian dari tanggung jawab manusia untuk membaca tanda-tanda alam yang telah Allah tetapkan. Konsep prediksi dalam Islam bukan hanya didasarkan pada dugaan semata, melainkan merupakan bentuk dari tanggung jawab moral untuk membaca tanda-tanda di alam harus dipahami sebagai bagian integral dari tanggung jawab kekhalifahan yang diemban manusia di bumi (Bozorgmehri *et al.*, 2023).

Prediksi curah hujan merupakan bagian dari ikhtiar ilmiah manusia dalam membaca tanda-tanda alam (*āyāt kauniyyah*) yang telah Allah tetapkan sebagai *sunnatullah*. Analisis terhadap angin, kelembapan, suhu, serta formasi awan bukan termasuk ramalan gaib, tetapi proses ilmiah yang Allah sendiri perintahkan untuk

diperhatikan. Dalam penelitian ini, metode LSTM dan Attention-Based LSTM digunakan sebagai bentuk pemanfaatan ilmu pengetahuan untuk memahami pola-pola cuaca yang Allah ciptakan. Upaya ini sejalan dengan nilai keislaman, karena Islam mendorong manusia untuk menggunakan akal, meneliti ciptaan-Nya, dan mengambil pelajaran dari hukum-hukum alam sebagai wujud syukur serta tanggung jawab sebagai khalifah di bumi.

BAB VII

KESIMPULAN

7.1 Kesimpulan

Berdasarkan pengujian model LSTM dan Attention-Based LSTM (AT-LSTM) untuk memprediksi curah hujan di Kabupaten Malang menggunakan parameter cuaca yaitu suhu rata-rata (T_{avg}), kelembapan rata-rata (RH_{avg}), durasi penyinaran matahari (ss), dan kecepatan angin rata-rata (ff_{avg}), diperoleh dua kesimpulan utama:

1. Kemampuan Prediksi Model: Kedua model (LSTM dan AT-LSTM) terbukti mampu mempelajari pola curah hujan berdasarkan parameter cuaca yang digunakan dan mencapai tingkat kesalahan yang rendah. Model LSTM menghasilkan RMSE sebesar 0.3090, sementara AT-LSTM menghasilkan RMSE sedikit lebih rendah, yaitu 0.3068.
2. Keunggulan Attention: Model AT-LSTM menunjukkan kinerja yang superior dibandingkan LSTM tanpa mekanisme Attention. Mekanisme Attention efektif menyoroti informasi temporal yang paling relevan, yang berhasil menurunkan nilai sekitar 0.71%. Hal ini membuktikan bahwa integrasi Attention dapat meningkatkan akurasi dan stabilitas prediksi curah hujan.

7.2 Saran

Beberapa saran yang diajukan untuk penelitian mendatang meliputi:

- a. Eksplorasi Arsitektur: Melakukan eksperimen dengan arsitektur deep learning alternatif, seperti Bi-LSTM, GRU, atau kombinasi CNN-LSTM, untuk menilai potensi peningkatan performa lebih lanjut.
- b. Ekspansi Geografis: Memperluas cakupan studi ke wilayah lain di Indonesia. Hal ini bertujuan untuk menguji keandalan model secara geografis dan kemampuan generalisasinya terhadap pola iklim yang berbeda.
- c. Metrik Evaluasi Komprehensif: Mengajukan penggunaan metrik evaluasi tambahan MAE (Mean Absolute Error), R^2 Score, atau MAPE guna memberikan gambaran kinerja model yang lebih lengkap dan menyeluruh.
- d. Inovasi Aplikasi: Mengembangkan aplikasi prediksi berbasis website atau mobile agar hasil penelitian ini dapat dimanfaatkan secara praktis oleh masyarakat, petani, atau instansi terkait dalam mitigasi bencana dan perencanaan pertanian.

DAFTAR PUSTAKA

- Badriyah, J., Fariza, A., & Harsono, T. (2022). Rainfall Prediction Using Long Short-Term Memory. *Journal of Media Informatics Budidarma*, 6(3), 1297. <https://doi.org/10.30865/mib.v6i3.4008>.
- Carnegie, M. D. A., & Chairani, C. (2023). Comparison of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) for Rainfall Prediction. *Journal of Media Informatics Budidarma*, 7(3), 1022. <https://doi.org/10.30865/mib.v7i3.6213>.
- Farikhul Firdaus, R., & Paputungan, I. V. (2022). Rainfall Prediction in Bandung City Using the Long Short-Term Memory Method. *Journal of Innovative*, 2(3), 453–460. <https://doi.org/10.54082/jupin.99>.
- Fredyan, R., & Kusuma, G. P. (2022). Spatiotemporal Convolutional LSTM with Attention Mechanism for Monthly Rainfall Prediction. *Journal of Communications in Mathematical Biology and Neuroscience*, 2022. <https://doi.org/10.28919/cmbn/7761>.
- Han, D., Liu, P., Xie, K., Li, H., Xia, Q., Cheng, Q., Wang, Y., Yang, Z., Zhang, Y., & Xia, J. (2023). An Attention-Based LSTM Model for Long-Term Runoff Forecasting and Factor Recognition. *Journal of Environmental Research Letters*, 18(2). <https://doi.org/10.1088/1748-9326/acaedd>.
- Hendra, Y., Mukhtar, H., & Hafsari, R. (2023). Rainfall Prediction in Pekanbaru City Using LSTM (Long Short-Term Memory). 3(2), 74–81. <https://ejurnal.umri.ac.id/index.php/SEIS/index>.
- Li, W., Zhang, P., Dong, H., Jia, Y., & Cao, W. (2021). RSDF-AM-LSTM: Regional Scale Division Rainfall Forecasting Using Attention and LSTM. *ACM/IMS Transactions on Data Science*, 2(4), 1–27. <https://doi.org/10.1145/3498333>.
- Liu, Z., Xu, Z., Jin, J., Shen, Z., & Darrell, T. (2023). Dropout Reduces Underfitting. <http://arxiv.org/abs/2303.01500>.
- Made, N., Candra, M., D., Agung, P., B., Kadek, N., & Wirdiani, A. (2022). Rainfall Prediction Using Vanilla RNN and LSTM to Determine the Start of the Rainy and Dry Seasons. *Journal of Education and Informatics Research*.
- Ministry of Religious Affairs of the Republic of Indonesia. (2019). *Al-Qur'an and Its Translation*. Jakarta: Ministry of Religious Affairs of the Republic of Indonesia.
- Musfiroh, M., Novitasari, D. C. R., Intan, P. K., & Wisnawa, G. G. (2023). Application of Principal Component Analysis (PCA) and Long Short-Term Memory (LSTM) in Daily Rainfall Prediction. *Journal of Building*

- Informatics, Technology and Science (BITS), 5(1).
<https://doi.org/10.47065/bits.v5i1.3114>.
- Latifah, A. L. (2022). Comparison of Various Epidemic Models on the COVID-19 Outbreak in Indonesia. *Journal of Technology and Computer System*, 10(1), 53–59. <https://doi.org/10.14710/jtsiskom.2022.14222>.
- Van Houdt, G., Mosquera, C., & Nápoles, G. (2022). A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*, 53(8), 5929–5955. <https://doi.org/10.1007/s10462-020-09838-1>.
- Wayan, I., & Suranata, A. (2023). Development of a Rainfall Prediction Model in Denpasar City Using LSTM and GRU Methods. *Journal of Systems and Informatics*.
- Fredyan, R., & Kusuma, G. P. (2022). Spatiotemporal Convolutional LSTM with Attention Mechanism for Monthly Rainfall Prediction. *Communications in Mathematical Biology and Neuroscience*, 2022. <https://doi.org/10.28919/cmbn/7761>.
- Nayak, G. H. H., Varalakshmi, A., Manjunath, M. G., Veershetty, Avinash, G., & Baishya, M. (2023). Trend Analysis and Prediction of Rainfall Using Deep Learning Models in Three Sub-Divisions of Karnataka. *Journal of Experimental Agriculture International*, 45(4), 36–48. <https://doi.org/10.9734/jeai/2023/v45i42114>.
- Buathongkhue, C., Sureeya, K., & Kaewthong, N. (2024). Analysis and Prediction of Rainfall with Oceanic Nino Index and Climate Variables Using Correlation Coefficient and Deep Learning. *Civil Engineering Journal (Iran)*, 10(5), 1354–1369. <https://doi.org/10.28991/CEJ-2024-010-05-01>.
- Maknunah, J., As, M., Setyowibowo, S., Farida, E., & Dwi Mumpuni, I. (2022). Indonesian consumer price index (CPI) forecasting using an exponential smoothing-state space model *Technology of Information, STMIK PPKIA Pradnya-Paramita. Business and Accounting Research (IJEBAR) Peer Reviewed-International Journal*, 8. <https://jurnal.stie-aas.ac.id/index.php/IJEBAR>.
- Muttaqien, F. H., Syafarina, I., Wahyuni, I. N., & Latifah, A. L. (2022). Time-Series Model for Climatological Forest Fire Prediction over Borneo. *Lontar Komputer: Scientific Journal of Information Technology*, 13(1), 35. <https://doi.org/10.24843/lkjiti.2022.v13.i01.p04>.
- Ibn Katsir, I. (n.d.). *Tafsir al-Qur'an al-'Azhim*. Beirut: Dar al-Kutub al-'Ilmiyyah.
- Shihab, M. Quraish. (2001). *Tafsir al-Mishbah*. Jakarta: Lentera Hati.
- Han, L., Wang, X., Yu, Y., & Wang, D. (2024). Power Load Forecast Based on CS-LSTM Neural Network. *Mathematics*, 12(9). <https://doi.org/10.3390/math12091402>.

- Salehin, I., & Kang, D. K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. In *Electronics (Switzerland)* (Vol. 12, Issue 14). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/electronics12143106>.
- Z. Cipiloglu Yildiz and S. B. Yildiz, "A portfolio construction framework using LSTM-based stock markets forecasting," *International Journal of Finance & Economics*, vol. 27, no. 2, pp. 2356–2366, Apr. 2022, doi: 10.1002/ijfe.2277.
- Y. Gao, R. Wang, and E. Zhou, "Stock Prediction Based on Optimized LSTM and GRU Models," *Sci Program*, vol. 2021, pp. 1–8, Sep. 2021, doi: 10.1155/2021/4055281.
- Rohmawati, F. Y., & Alim, A. S. (2024). The Relationship Between Wind and Rain in the Qur'an: Thematic Interpretation Approach and Meteorological Analysis. *AL-QUDS: Jurnal Studi Alquran Dan Hadis*, 8(1), 1. <https://doi.org/10.29240/alquds.v8i1.6748>.
- F. Liu and C. Liang, "Short-term power load forecasting based on AC-BiLSTM model," *Energy Reports*, vol. 11, pp. 1570–1579, Jun. 2024, doi: 10.1016/j.egyr.2024.01.026.
- D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput Sci*, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.
- F. D. Paiva, R. T. N. Cardoso, G. P. Hanaoka, and W. M. Duarte, "Decision-making for financial trading: A fusion approach of machine learning and portfolio selection," *Expert Syst Appl*, vol. 115, pp. 635–655, Jan. 2019, doi: 10.1016/j.eswa.2018.08.003.
- Yaro, A. S., Maly, F., & Prazak, P. (2023). Outlier Detection in Time-Series Receive Signal Strength Observation Using Z-Score Method with Sn Scale Estimator for Indoor Localization. *Applied Sciences (Switzerland)*, 13(6). <https://doi.org/10.3390/app13063900>.
- D. C. Yıldırım, I. H. Toroslu, and U. Fiore, "Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators," *Financial Innovation*, vol. 7, no. 1, p. 1, Dec. 2021, doi: 10.1186/s40854-020-00220-2.
- M. Agrawal, P. Kumar Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock Prediction Based on Technical Indicators Using Deep Learning Model," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 287–304, 2022, doi: 10.32604/cmc.2022.014637.

LAMPIRAN

Lampiran 1: Tabel Metrik Training LSTM

<i>Batch Size</i>	<i>Epochs</i>	<i>Training RMSE</i>	<i>Validation RMSE</i>	<i>Testing RMSE</i>
4	10	0.1973	0.2011	0.2103
4	20	0.1941	0.1987	0.2113
4	50	0.1911	0.1980	0.2068
4	100	0.1912	0.1979	0.2084
16	10	0.1999	0.2038	0.2131
16	20	0.1970	0.2008	0.2091
16	50	0.1937	0.1989	0.2088
16	100	0.1893	0.1987	0.2117
32	10	0.1994	0.2031	0.2124
32	20	0.1962	0.2005	0.2088
32	50	0.1936	0.1994	0.2079
32	100	0.1934	0.1988	0.2086
64	10	0.2008	0.2043	0.2137
64	20	0.1977	0.2013	0.2103
64	50	0.1942	0.1997	0.2081
64	100	0.1933	0.1989	0.2086
128	10	0.2008	0.2056	0.2142
128	20	0.1988	0.2022	0.2093
128	50	0.1954	0.2000	0.2070
128	100	0.1943	0.1993	0.2076

Lampiran 2: Tabel Metrik Training Attention-Based LSTM

<i>Batch Size</i>	<i>Epochs</i>	<i>Training RMSE</i>	<i>Validation RMSE</i>	<i>Testing RMSE</i>
4	10	0.1994	0.2034	0.2141
4	20	0.1949	0.2001	0.2076
4	50	0.1936	0.1979	0.2073
4	100	0.1863	0.1983	0.2132
16	10	0.1979	0.2010	0.2107
16	20	0.1958	0.1996	0.2069
16	50	0.1918	0.1988	0.2072
16	100	0.1936	0.1980	0.2078
32	10	0.1983	0.2023	0.2133
32	20	0.1948	0.1989	0.2067
32	50	0.1921	0.1983	0.2053
32	100	0.1925	0.1989	0.2075
64	10	0.1991	0.2024	0.2150
64	20	0.1969	0.1996	0.2105
64	50	0.1932	0.1989	0.2061
64	100	0.1902	0.1984	0.2073
128	10	0.2026	0.2071	0.2186
128	20	0.1978	0.1998	0.2096
128	50	0.1941	0.1973	0.2057
128	100	0.1946	0.1984	0.2059

Lampiran 3: Source Code & Segmen Program

Seluruh *source code* dapat diakses melalui akun GitHub pada tautan berikut:

<https://github.com/ahmadromadhani007/RainfallPrediction>

Segmen Program 1 Memuat dataset

```
1. import numpy as np
2. import pandas as pd
3. # Path ke file CSV
4. data_path = 'data1.csv'
5. # Memuat data dari file CSV
6. data = pd.read_csv(data_path, encoding='latin-1')
7. data
```

Segmen Program 2 Mengubah format tanggal, mencari dan mengisi data hilang

```
1. #Convert 'Tanggal' to datetime format
2. data['Tanggal'] = pd.to_datetime(data['Tanggal'],
    format='%d/%m/%Y')
3. #Hitung data hilang
4. data.isna().sum()
5. #Handle missing values
6. data = data.ffill().bfill()
```

Segmen Program 3 Mengatasi outlier with z-score method

```
1. #Fungsi untuk menggantikan outlier menggunakan metode Z-score
2. from scipy.stats import zscore
3.
4. def replace_outliers_zscore(df, columns):
5.     # Iterasi untuk setiap kolom yang akan diperiksa
6.     for column in columns:
7.         # Menghitung Z-score untuk kolom yang dipilih
8.         df['zscore'] = zscore(df[column])
9.         # Menghitung nilai median dari kolom yang dipilih
10.        median = df[column].median()
11.        # Menggantikan nilai yang memiliki Z-score lebih
    dari 3 dengan nilai median
12.        df[column] = np.where(np.abs(df['zscore']) > 3,
    median, df[column])
13.        # Menghapus kolom zscore setelah selesai
14.        df.drop(columns='zscore', inplace=True)
15.    return df
16.
17. # Daftar kolom yang akan diperiksa outlier-nya
18. columns_to_check = ['Tavg (°C)', 'RH_avg (%)', 'ss (jam)',
    'ff_avg (m/s)', 'RR (mm)']
19.
20. # Menggantikan outlier di semua kolom yang dipilih
21. data = replace_outliers_zscore(data, columns_to_check)
22. data
```

Segmen Program 4 Menghitung korelasi

```

1. # Menghitung korelasi Pearson
2. features = data.columns.drop(['Tanggal', 'RR (mm)'])
3. correlation_matrix = data[features].corrwith(data['RR (mm)'])
4. print(correlation_matrix.sort_values(ascending=False))
5.
6. # Hasil korelasi Pearson yang sudah dilakukan
7. correlation_results = {
8.     "RH_avg (%) vs RR (mm)": 0.326558,
9.     "Tavg (°C) vs RR (mm)": 0.040954,
10.    "ss (jam) vs RR (mm)": -0.257047,
11.    "ff_avg (m/s) vs RR (mm)": -0.181402
12. }
13.
14. # Jumlah sampel
15. n = len(data)
16.
17. # Fungsi untuk hitung Z_hitung dan p-value
18. def fisher_test(r, n):
19.     Z = np.sqrt(n - 3)/2 * np.log((1 + r)/(1 - r)) # rumus
    Z_hitung
20.     p = 2 * (1 - norm.cdf(abs(Z))) # two-
    tailed p-value
21.     return Z, p
22.
23. # Simpan hasil ke DataFrame
24. rows = []
25. for fitur, r in correlation_results.items():
26.     Z, p = fisher_test(r, n)
27.     rows.append([fitur, r, Z, p])
28.
29. df_results = pd.DataFrame(rows, columns=["Seleksi Fitur", "r
    (Korelasi)", "Z_hitung", "Two-tailed (p-value)"])
30. df_results["Z_hitung"] = df_results["Z_hitung"].round(4)
31. df_results["Two-tailed (p-value)"] = df_results["Two-tailed
    (p-value)"].apply(lambda x: "< 0.0001" if x < 0.0001 else
    round(x, 5))
32. print(df_results)

```

Segmen Program 5 Menampilkan grafik korelasi

```

1. # Menampilkan scatter plot untuk semua fitur terhadap RR
2. import matplotlib.pyplot as plt
3. import seaborn as sns
4. for feature in features:
5.     sns.scatterplot(data=data, x=feature, y='RR (mm)')
6.     plt.title(f'{feature} vs RR (mm)')
7.     plt.show()
8.
9. # Menghapus fitur yang kurang berkorelasi 'ss' and 'ff_avg'
10. data = data.drop(columns=['Tanggal'])
11.
12. # Menyimpan data final
13. data.to_csv(final_data_path, index=False)

```

Segmen Program 6 Memuat dataset

```
1. # Memuat data dari file CSV
2. data = pd.read_csv(final_data_path)
3. # Menampilkan data
4. Data
```

Segmen Program 7 Normalisasi dan Transformasi Log

```
1. # Normalize data dengan log transform pada target
2. import numpy as np
3. from sklearn.preprocessing import MinMaxScaler
4.
5. feature_columns = ['Tavg (°C)', 'RH_avg (%)', 'ss (jam)',
6. 'ff_avg (m/s)']
7.
8. target_column = ['RR (mm)']
9.
10. scaler_X = MinMaxScaler()
11. scaler_Y = MinMaxScaler()
12.
13. # Normalisasi fitur (X)
14. data[feature_columns] =
15.     scaler_X.fit_transform(data[feature_columns])
16.
17. # Transformasi log pada target (untuk meredam lonjakan
18.     ekstrem)
19. data['RR_log'] = np.log1p(data['RR (mm)'])# log(1 + y)
20.
21. # Normalisasi target setelah log
22. data['RR_norm'] = scaler_Y.fit_transform(data[['RR_log']])
23.
24. # Cek hasil transformasi
25. print(data[['Tavg (°C)', 'RH_avg (%)', 'ss (jam)', 'ff_avg
26.     (m/s)', 'RR (mm)', 'RR_log', 'RR_norm']].head(7))
```

Segmen Program 8 Mengubah dimensi data

```
1. # Definisikan jumlah timestep
2. timesteps = 7
3.
4. # Fungsi untuk membuat dataset baru dengan lagging
5.     berdasarkan timestep
6. def create_lagged_features(features, target, timesteps):
7.     X, y = [], []
8.     for i in range(len(features) - timesteps):
9.         # Input sequence
10.        X.append(features[i:(i + timesteps), :])
11.        # Target sudah pakai log + normalized
12.        y.append(target[i + timesteps])
13.    return np.array(X), np.array(y)
14.
15. # === Pilih kolom fitur (X) ===
16. X_features = data[['Tavg (°C)', 'RH_avg (%)', 'ss (jam)',
17. 'ff_avg (m/s)']].values
```

```

17. # === Target (y) pakai RR_norm ===
18. y_target = data[['RR_norm']].values
19.
20. # Buat dataset lagged
21. X, y = create_lagged_features(X_features, y_target,
    timesteps)
22. # Reshape data menjadi bentuk (sequence, timesteps *
    features)
23. X_test_flattened = X_test.reshape(X_test.shape[0], -1)
24. X_test_df = pd.DataFrame(X_test_flattened,
    columns=[f'Feature_{i}' for i in
    range(X_test_flattened.shape[1])])
25. # Simpan ke CSV
26. X_test_df.to_csv('X_test.csv', index=False)
27.
28. print("X shape:", X.shape)
29. print("y shape:", y.shape)

```

Segmen Program 9 Pembagian data

```

1. # Pembagian Data
2. from sklearn.model_selection import train_test_split
3. import numpy as np
4. # --- Pisahkan data menjadi training (80%) dan testing (20%)
    ---
5. X_train, X_temp, y_train, y_temp = train_test_split(X, y,
    test_size=0.2, shuffle=False)
6. X_val, X_test, y_val, y_test = train_test_split(X_temp,
    y_temp, test_size=0.5, shuffle=False)
7. X_test_print = np.concatenate([X_val, X_test], axis=0)
8. y_test_print = np.concatenate([y_val, y_test], axis=0)
9.
10. print("Shape of X_train:", X_train.shape)
11. print("Shape of y_train:", y_train.shape)
12. print("Shape of X_test:", X_test_print.shape)
13. print("Shape of y_test:", y_test_print.shape)

```