



Analisis Prediksi Curah Hujan di Kota Tangerang Menggunakan Metode LSTM dan GRU

Dahlan Supriatna¹, Sajarwo Anggai², Tukiyat³

^{1,2,3}Teknik Informatika, Pascasarjana, Universitas Pamulang

¹dahlansupriatna165@gmail.com, ²sajarwo@gmail.com, ³dosen02711@unpam.ac.id

Abstract

Unpredictable rainfall can significantly impact various sectors, including agriculture, energy and infrastructure. Therefore, accurate rainfall prediction is crucial for mitigating the risks of both floods and droughts. This research aims to compare the accuracy of rainfall prediction using two prominent deep learning algorithms LSTM and GRU. The goal is to contribute to more effective water resource management. The models were applied to historical rainfall data and related meteorological variables. The secondary data for this study was taken from the BMKG of Tangerang City started from January 2014 to January 2025 with the total is 4,062 data points. The performance of model was evaluated using standard metrics such as MAE, MSE, RMSE, and R2. The findings indicate that the LSTM model with its optimal hyperparameter configuration consistently delivered superior performance. This optimal setup for LSTM included: 36-month timesteps, 64 memory units, 100 training epochs, a batch size of 16, a dropout rate of 0.3, and a learning rate of 0.0001. Under these conditions, the LSTM model achieved the following best evaluation metrics, MAE 0.08473, MSE 0.00973, RMSE 0.09863 R2 0.65601. The relatively high R2 score suggests that the LSTM model was able to explain approximately 65.6% of the variability in the actual rainfall data, demonstrating its robust predictive capability. In comparison, the GRU model's best performance (achieved with a batch size of 32) showed slightly lower evaluation metrics, MAE 0.08883, MSE 0.01078, RMSE 0.10383, R2 0.61878., LSTM generally proved to be more effective in its predictive capabilities.

Keywords: Prediction, Rainfall, Deep Learning, LSTM, GRU

Abstrak

Curah hujan yang tidak menentu dapat memengaruhi berbagai sektor, seperti pertanian, energi, dan infrastruktur. Akurasi prediksi curah hujan sangat penting untuk mitigasi risiko bencana banjir maupun kekeringan. Penelitian ini bertujuan untuk membandingkan akurasi prediksi curah hujan menggunakan dua algoritma deep learning, yaitu LSTM dan GRU serta dapat memberikan kontribusi pada pengelolaan sumber daya air yang lebih efektif. Model ini diterapkan pada data historis curah hujan dan variabel meteorologi terkait, data penelitian adalah data sekunder yang bersumber dari data BMKG Kota Tangerang periode Januari 2014 – Januari 2025 sebanyak 4.062 data. Evaluasi kinerja model dilakukan menggunakan metrik seperti MAE, MSE, RMSE, dan R². Hasil menunjukkan Model LSTM dengan konfigurasi hyperparameter optimal—terdiri dari timesteps 36 bulan, 64 unit memori, 100 epoch pelatihan, batch size 16, dropout 0.3, dan learning rate 0.0001—menghasilkan metrik evaluasi terbaik MAE sebesar 0.08473, MSE sebesar 0.00973, RMSE sebesar 0.09863, dan R2 sebesar 0.65601. Nilai R2 yang relatif tinggi ini mengindikasikan bahwa model LSTM mampu menjelaskan sekitar 65.6% dari variabilitas dalam data curah hujan aktual. Sebagai perbandingan, model GRU dengan kinerja terbaiknya (menggunakan batch size 32) menunjukkan metrik evaluasi yang sedikit di bawah LSTM, yaitu MAE 0.08883, MSE 0.01078, RMSE 0.10383, dan R2 Score 0.61878, secara keseluruhan, LSTM terbukti lebih unggul dalam kapabilitas prediksinya.

Kata Kunci: Prediksi, Curah Hujan, Deep Learning, LSTM, GRU.

1. Pendahuluan

Hujan merupakan elemen krusial dalam siklus hidrologi dan memegang peranan penting bagi kehidupan manusia. Informasi curah hujan yang akurat sangat esensial untuk perencanaan dan pengelolaan sumber daya air, termasuk operasional waduk dan pencegahan banjir. Selain itu, curah hujan memengaruhi berbagai aspek kehidupan perkotaan seperti lalu lintas dan sistem pembuangan limbah[1]. Prediksi curah hujan memiliki dampak luas, mulai dari sektor pertanian hingga pariwisata, dan dapat berfungsi sebagai sistem peringatan dini banjir serta alat efektif untuk pengelolaan sumber daya air[2]. Prediksi ini dapat memberikan bantuan kepada para pembambil kebijakan dalam menentukan waktu yang tepat untuk melakukan intervensi atau mitigasi[3]. Prediksi adalah salah satu cara untuk mencari solusi dari masalah yang dihadapi dengan menggunakan beberapa variabel[4].

Meskipun demikian, prediksi curah hujan tetap menjadi tantangan besar dalam hidrologi operasional. Kompleksitas proses *atmosfer* dan variasi *spasial-temporal* yang ekstrem menjadikan pemahaman serta pemodelannya sangat sulit. Curah hujan dipengaruhi oleh parameter-parameter *dependen* seperti kelembaban, kecepatan angin, dan suhu, yang bervariasi secara geografis[5]. Hal ini berarti model prediksi yang efektif untuk satu lokasi mungkin tidak cocok untuk wilayah lain. Sifat curah hujan yang *nonlinier* juga menyebabkan akurasi prediksi dengan teknik-teknik tradisional masih kurang memuaskan, meskipun telah banyak kemajuan dalam prakiraan cuaca selama beberapa dekade terakhir[6].

Dalam menghadapi tantangan ini, jaringan saraf tiruan (JST) muncul sebagai pendekatan *induktif* yang menjanjikan dalam prediksi curah hujan. Kemampuannya dalam memodelkan sistem *nonlinier*, fleksibilitas, dan pembelajaran berbasis data tanpa memerlukan pengetahuan awal tentang perilaku daerah tangkapan air menjadikannya pilihan menarik. Untuk kumpulan data yang besar, teknik pembelajaran mendalam (*deep learning*), yang merupakan turunan dari jaringan saraf, digunakan untuk menemukan hasil yang bermakna. JST telah terbukti berhasil dalam berbagai bidang sains dan teknik karena kemampuannya memodelkan sistem *linier* dan *nonlinier* tanpa asumsi seperti pendekatan statistik tradisional, bahkan mengungguli model *regresi linier* sederhana[7].

Berdasarkan hal tersebut, penelitian ini bertujuan untuk membandingkan kinerja dua metode *deep learning*, yaitu LSTM dan GRU, guna mendapatkan prediksi curah hujan yang lebih baik. Kami akan mengabaikan hukum fisika yang mengatur proses curah hujan dan berfokus pada prediksi pola curah hujan berdasarkan fitur-fitur dan data historis yang ada[8]. Pendekatan ini menggunakan pengenalan pola untuk memprediksi curah hujan. Model prediktif yang dikembangkan dalam penelitian ini didasarkan pada teknik pembelajaran mendalam. Kami mengusulkan pendekatan berbasis LSTM dan GRU untuk prediksi curah hujan menggunakan data BMKG Kota Tangerang dari Januari 2014 hingga Januari 2025, dan akan membandingkan hasilnya dengan model-model *deep learning* mutakhir. Dengan demikian, metode ini diharapkan dapat berkontribusi dalam memprediksi curah hujan dan penyelesaian masalah terkait curah hujan.

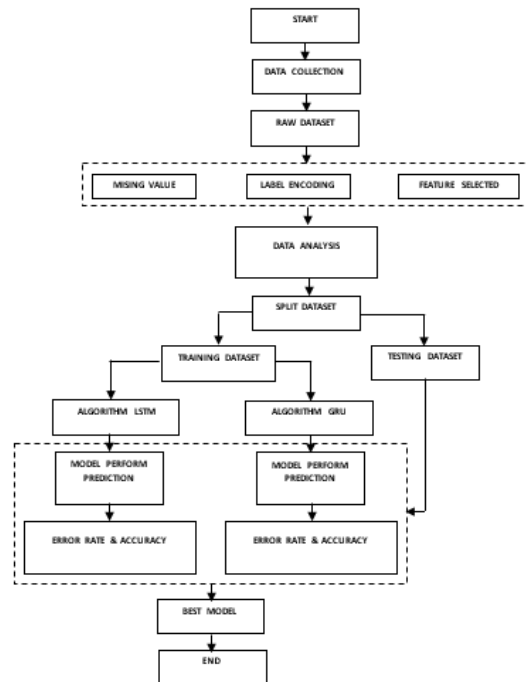
2. Metode Penelitian

Penelitian ini bertujuan untuk memprediksi curah hujan di Kota Tangerang menggunakan algoritma *deep learning* berbasis *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU). Metodologi yang digunakan mencakup tahapan analisis kebutuhan data, perancangan penelitian, pengumpulan dan pengolahan data, pembangunan model, hingga evaluasi performa prediksi. Dataset yang digunakan bersumber dari data BMKG Kota Tangerang periode Januari 2014 hingga Januari 2025. Terdapat tujuh atribut utama yang digunakan, seperti yang ditunjukkan pada tabel 1 dibawah ini.

Tabel 1. Atribut Yang Digunakan

Atribut	Keterangan
Tanggal	Tanggal Pengukuran Data
TN	Suhu Minimum dalam °C
TX	Suhu Maximum dalam °C
TAVG	Suhu Rata-rata dalam °C
RH_AVG	Kelembaban Rata-rata dalam mm
RR	Jumlah Curah Hujan dalam mm
SS	Durasi Sinar Matahari dalam Jam

Selanjutnya perancangan penelitian, tahap dimana dibuatkan suatu rancangan untuk memproses data mulai dari input data sampai mendapatkan nilai akurasi yang terbaik dari algoritma *Long Short Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU), seperti yang ditunjukkan oleh gambar 1.



Gambar 1 Perancangan Penelitian

Dalam perancangan penelitian tahapan pertama setelah inisiasi adalah pengumpulan data. Dalam konteks penelitian ini, data yang akan dikumpulkan adalah data curah hujan dari BMKG Kota Tangerang, mencakup periode Januari 2014 hingga Januari 2025.

Setelah data terkumpul, data tersebut akan berupa *raw dataset* atau dataset mentah yang mungkin masih mengandung beberapa anomali atau kekurangan, sebelum data dapat dianalisis dan digunakan untuk pemodelan, *raw dataset* akan melalui serangkaian proses *preprocessing*.

Setelah melalui tahapan *preprocessing*, data akan siap untuk dianalisis. Tahapan ini melibatkan pemahaman lebih lanjut tentang karakteristik data dan pola-pola yang mungkin ada, dataset yang telah dianalisis kemudian akan dibagi menjadi dua bagian utama, *training dataset* dan *testing dataset*. Pembagian ini penting untuk melatih model dan kemudian mengevaluasi kinerjanya pada data yang belum pernah dilihat sebelumnya.

Setelah model dilatih dengan masing-masing algoritma LSTM dan GRU menggunakan *training dataset*, tahapan selanjutnya adalah evaluasi kinerja model. Setelah kedua model dievaluasi berdasarkan *error rate* dan *accuracy* pada *testing dataset*, akan dilakukan perbandingan untuk menentukan model mana yang memiliki kinerja terbaik. Model dengan *error rate* terendah dan atau *accuracy* tertinggi akan diidentifikasi sebagai model terbaik.

Evaluasi model dilakukan dengan mengukur tingkat kesalahan prediksi menggunakan *metric Mean Absolute Error* (MAE) untuk mengukur rata-rata kesalahan absolut antara nilai prediksi dan aktual, *Mean Squared Error* (MSE) untuk mengukur rata-rata kesalahan kuadrat, memperbesar efek kesalahan besar, *Root Mean Squared Error* (RMSE) yang merupakan akar dari MSE, digunakan untuk menginterpretasikan besaran kesalahan dalam satuan yang sama dengan nilai target dan *Koefisien Determinasi* (R^2) untuk menunjukkan proporsi variabilitas yang dapat dijelaskan oleh model. Nilai R^2 yang mendekati 1 menunjukkan performa model yang baik.

Teknik analisis yang digunakan bersifat kuantitatif dan regresif, karena variabel target (curah hujan) bersifat kontinu. Hasil dari evaluasi metrik tersebut menjadi dasar dalam menentukan algoritma yang paling optimal.

3. Hasil dan Pembahasan

Penelitian ini dilakukan dengan pendekatan kuantitatif menggunakan dua arsitektur *deep learning*, yaitu *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU), untuk memodelkan dan memprediksi curah hujan berdasarkan data deret waktu. Data yang digunakan adalah data sekunder dari BMKG Kota Tangerang yang mencakup 4.062 data harian dari Januari 2014 hingga Februari 2025.

Tujuan utama penelitian adalah mengevaluasi dan membandingkan performa kedua model dalam memprediksi curah hujan bulanan berdasarkan fitur meteorologis.

3.1. Dataset dan Fitur

Dataset diperoleh dari situs resmi BMKG (<https://dataonline.bmkg.go.id/>) dan terdiri atas data harian yang kemudian *diresample* menjadi data bulanan. Fitur-fitur yang digunakan dalam model meliputi, Suhu minimum (TN), Suhu maksimum (TX), Suhu rata-rata (TAVG), Kelembaban rata-rata (RH_AVG), Durasi penyinaran matahari (SS), Curah hujan (RR). Variabel RR digunakan sebagai target prediksi (*output*), sementara lima fitur lainnya digunakan sebagai input (*prediktor*).

3.2. Pra-pemrosesan Data

Proses pra-pemrosesan data adalah serangkaian langkah krusial untuk memastikan data siap digunakan dalam pelatihan model, menghasilkan performa yang optimal, dan meningkatkan stabilitas model[9]. Tahapan ini dimulai dengan pembersihan data, di mana nilai-nilai yang tidak valid seperti *NULL*, 8888, dan 9999 diidentifikasi dan dihapus dari semua fitur. Langkah ini penting untuk menghilangkan *noise* dan memastikan kualitas data.

Selanjutnya, dilakukan *resampling* data. Data harian diubah menjadi data bulanan. Untuk fitur suhu dan kelembaban, metode *agregasi* yang digunakan adalah rata-rata (*mean*), sementara untuk curah hujan (RR), metode yang digunakan adalah penjumlahan (*sum*). *Resampling* ini membantu dalam menyederhanakan data dan menangkap pola bulanan yang lebih relevan.

Tahap ketiga adalah normalisasi. Seluruh fitur dinormalisasi menggunakan *MinMaxScaler* ke dalam rentang [0, 1][10]. Normalisasi ini bertujuan untuk mempercepat proses pelatihan model dan meningkatkan stabilitasnya dengan memastikan bahwa semua fitur memiliki skala yang seragam, sehingga tidak ada fitur yang mendominasi karena rentang nilainya yang lebih besar.

Setelah normalisasi, dilakukan pembentukan urutan data (*Time Series Sequence*). Pada tahap ini, urutan data dibuat menggunakan parameter *timesteps* tertentu, yaitu 24 dan 36. *Timesteps* ini menentukan jumlah bulan historis yang akan digunakan sebagai *input* untuk memprediksi bulan berikutnya, yang sangat penting untuk analisis deret waktu[11].

Terakhir, pembagian dataset dilakukan. Dataset dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%[12]. Pembagian ini dilakukan menggunakan fungsi *train_test_split* tanpa melakukan *shuffle*. Hal ini penting untuk menjaga urutan *temporal* data, yang merupakan karakteristik fundamental dalam analisis deret waktu. Pembagian ini memastikan bahwa model dilatih pada data historis dan dievaluasi pada data yang belum pernah dilihat sebelumnya secara kronologis.

3.3. Arsitektur Model

Untuk membangun model prediksi, kami menggunakan pustaka *Keras* dengan *TensorFlow* sebagai *backend*. Dua arsitektur model utama telah dikembangkan, yaitu Model LSTM dan Model GRU, keduanya dirancang untuk tugas *regresi*.

Model LSTM terdiri dari beberapa *layer*. Dimulai dengan satu *layer* LSTM yang memiliki 64 unit, bertanggung jawab untuk mempelajari *dependensi* jangka panjang dalam data. Untuk mencegah *overfitting*, *layer* ini diikuti oleh *dropout* dengan *rate* antara 0.2 hingga 0.3. Selanjutnya, ada *dense layer* dengan 32 *neuron* yang menggunakan fungsi aktivasi *ReLU*, berfungsi untuk menambahkan kompleksitas *non-linear* pada model. Terakhir, terdapat *dense output layer* dengan 1 *neuron* dan fungsi aktivasi *linear* untuk menghasilkan prediksi *regresi*.

Sementara itu, Model GRU memiliki arsitektur yang sangat mirip dengan Model LSTM, namun perbedaannya terletak pada penggunaan GRU *layer* alih-alih LSTM. Meskipun keduanya adalah jenis *recurrent neural network* (RNN) yang efektif untuk data sekuensial, GRU umumnya memiliki struktur yang lebih sederhana dan lebih sedikit parameter dibandingkan LSTM, yang terkadang dapat menghasilkan waktu pelatihan yang lebih cepat tanpa mengorbankan performa secara signifikan.

Kedua model, baik LSTM maupun GRU, dikompilasi menggunakan *optimizer Adam* dengan *learning rate* sebesar 0.0001. *Learning rate* yang kecil ini membantu model untuk belajar secara bertahap dan menghindari *overshooting* solusi optimal. Sebagai *loss function*, kami memilih *Mean Squared Error* (MSE), yang merupakan metrik umum untuk tugas *regresi* dan mengukur rata-rata kuadrat perbedaan antara nilai prediksi dan nilai sebenarnya.

3.4. Hyperparameter Pelatihan

Untuk menganalisis dan membandingkan performa model LSTM dan GRU. Kedua model diuji dengan berbagai kombinasi *hyperparameter* dalam enam skenario pengujian. Evaluasi dilakukan berdasarkan empat metrik utama: MAE, MSE, RMSE, dan R² untuk menilai tingkat akurasi dan generalisasi masing-masing model

Model diuji dengan berbagai konfigurasi *hyperparameter* untuk mengevaluasi performa optimal[13]. Tabel 2 berikut merangkum kombinasi yang digunakan.

Tabel 2. Hyperparameter Yang Digunakan

Hyperparameter	Nilai yang di ujikan
Timesteps	24 dan 36
Unit Memori	32 dan 64
Epoch	50 dan 100
Batch Size	16 dan 32
Dropout	0.2 dan 0.3
Learning Rate	0.001 dan 0.0001

Model juga menggunakan teknik *Early Stopping* dengan *patience*=20 untuk menghentikan pelatihan lebih awal jika *validasi loss* tidak membaik

3.5. Skenario Pengujian

Untuk memahami bagaimana berbagai konfigurasi *hyperparameter* memengaruhi kinerja model, serangkaian skenario pengujian komprehensif telah dirancang. Skenario ini mencakup aspek-aspek krusial dalam pembelajaran deret waktu, seperti panjang historis data (*timesteps*), kapasitas memori model (unit), jumlah *epoch* pelatihan, ukuran *batch*, tingkat *dropout*, dan kecepatan pembelajaran (*learning rate*).

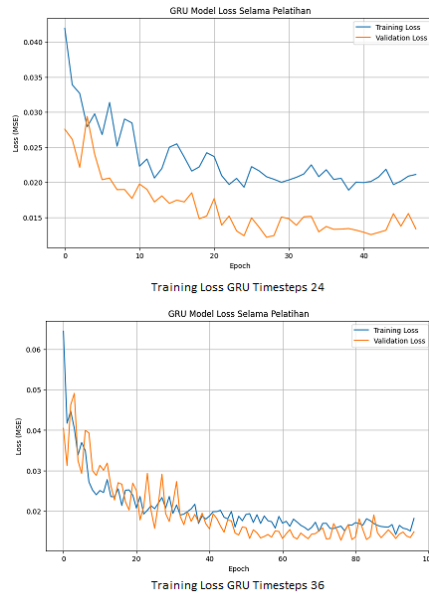
3.5.1. Variasi *Timesteps* (24 vs 36 bulan)

Tabel 3. Perbandingan Model Dengan *Timesteps* 24 vs 36

Model	Time steps	MAE	MSE	RSME	R ²
LSTM	24	0,097	0,012	0,113	0,526
LSTM	36	0,084	0,010	0,100	0,650
GRU	24	0,086	0,012	0,110	0,553
GRU	36	0,089	0,013	0,113	0,546

Pengujian dilakukan dengan dua panjang input historis: 24 bulan (2 tahun) dan 36 bulan (3 tahun). Hasilnya menunjukkan bahwa *timesteps* 36 bulan umumnya menghasilkan kinerja yang lebih baik. Hal ini karena data historis yang lebih panjang memungkinkan model untuk menangkap pola musiman tahunan secara lebih lengkap dan akurat. Secara spesifik, model LSTM dengan *timesteps* 36 bulan mencapai nilai R² sebesar 0.650, sementara GRU dengan *timesteps* yang sama menghasilkan R2 sebesar 0.546.

Gambar 2 Grafik Training Loss LSTM *Timesteps* 24 vs 36



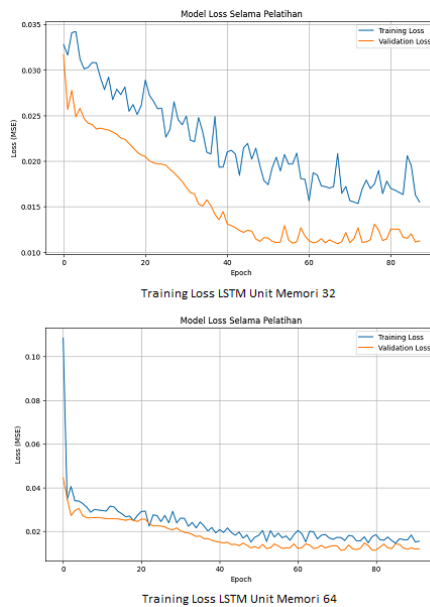
Gambar 3 Grafik Training Loss GRU *Timesteps* 24 vs 36

3.5.2. Variasi *Unit Memori* (32 vs 64)

Tabel 4. Perbandingan Model Dengan *Unit Memori* 32 vs 64

Model	Unit Memori	MAE	MSE	RSME	R ²
LSTM	32	0,089	0,010	0,104	0,613
LSTM	64	0,084	0,010	0,100	0,650
GRU	32	0,091	0,012	0,110	0,570
GRU	64	0,086	0,013	0,113	0,546

Eksperimen ini bertujuan untuk mengevaluasi pengaruh jumlah neuron tersembunyi, atau unit memori, dalam menyimpan dan mengolah informasi sekuensial. Ditemukan bahwa model dengan 64 unit cenderung menunjukkan kinerja yang lebih tinggi dalam menangkap kompleksitas pola data. Namun, perlu dicatat bahwa model dengan 32 unit terbukti lebih efisien secara komputasi. Sebagai contoh, LSTM dengan 64 unit mencapai MAE sebesar 0.084 dan R² sebesar 0.650, sedangkan GRU dengan 32 unit menunjukkan MAE 0.086 dan R² 0.546.



Gambar 4 Grafik Training Loss LSTM Memori 32 vs 64



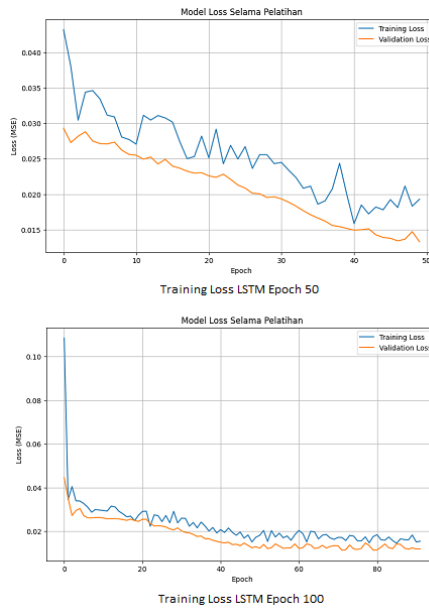
Gambar 5 Grafik Training Loss GRU Memori 32 vs 64

3.5.3. Variasi Epoch (50 vs 100)

Tabel 5. Perbandingan Model Dengan Epoch 50 vs 100

Model	Epoch	MAE	MSE	RSME	R ²
LSTM	50	0,099	0,013	0,115	0,530
LSTM	100	0,084	0,010	0,100	0,650
GRU	50	0,093	0,013	0,117	0,514
GRU	100	0,086	0,013	0,113	0,546

Pengujian ini mengevaluasi dampak jumlah iterasi pelatihan (*epoch*) terhadap akurasi model. Hasilnya menunjukkan bahwa model LSTM mengalami peningkatan performa yang signifikan ketika dilatih selama 100 *epoch*, menunjukkan bahwa model ini membutuhkan lebih banyak iterasi untuk konvergen secara optimal. Di sisi lain, model GRU menunjukkan kinerja yang relatif stabil antara 50 dan 100 *epoch*. Secara spesifik, LSTM 100 *epoch* mencapai MSE 0.0099, sementara GRU 50 *epoch* memiliki MSE 0.0137.



Gambar 6 Grafik Training Loss LSTM Epoch 50 vs 100



Gambar 7 Grafik Training Loss GRU Epoch 50 vs 100

3.5.4. Variasi Batch Size (16 vs 32)

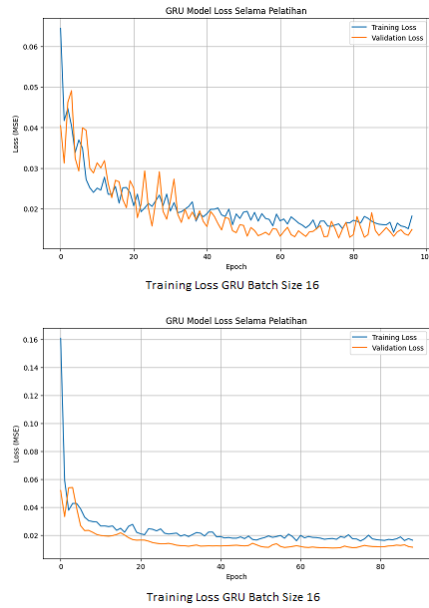
Tabel 6. Perbandingan Model Dengan Batch Size 16 vs 32

Model	Batch	MAE	MSE	RSME	R ²
LSTM	16	0,084	0,010	0,100	0,650
LSTM	32	0,090	0,011	0,106	0,601
GRU	16	0,086	0,013	0,113	0,546
GRU	32	0,088	0,010	0,103	0,618

Eksperimen dengan ukuran *batch* menunjukkan bahwa penggunaan *batch size* yang lebih besar (32) dapat mempercepat proses pelatihan. Namun, kecepatan ini datang dengan potensi risiko konvergensi yang kurang halus, di mana model mungkin kesulitan menemukan titik optimal yang presisi. Sebagai ilustrasi, GRU dengan *batch size* 32 mencapai R² sebesar 0.61878, sementara LSTM dengan batch size 32 menghasilkan R² sebesar 0.60162.



Gambar 8 Grafik Training Loss LSTM Batch Size 16 vs 32



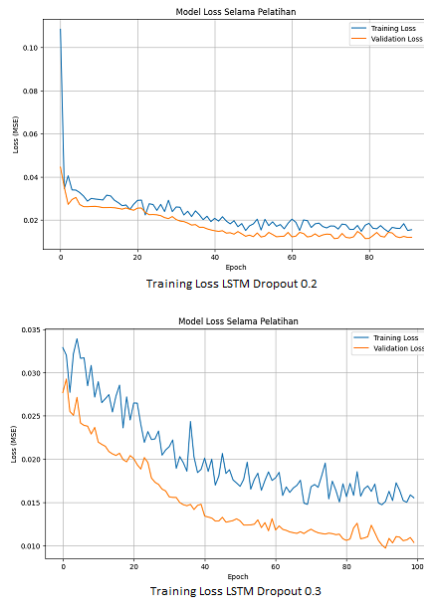
Gambar 9 Grafik Training Loss GRU *Batch Size* 16 vs 32

3.5.5. Variasi *Dropout* (0.2 vs 0.3)

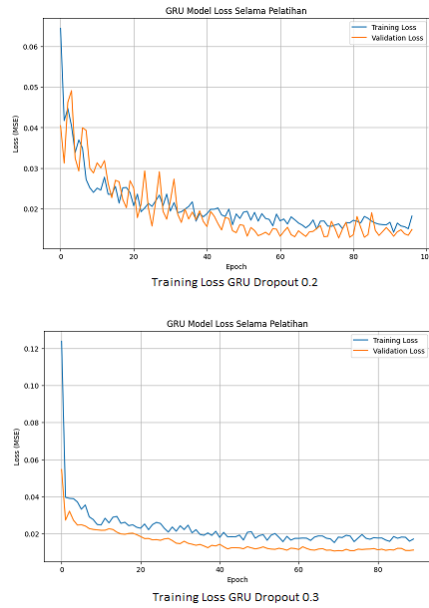
Tabel 7. Perbandingan Model Dengan *Dropout* 0.2 vs 0.3

Model	Dropout	MAE	MSE	RSME	R ²
LSTM	0.2	0,084	0,010	0,100	0,650
LSTM	0.3	0,084	0,009	0,098	0,656
GRU	0.2	0,086	0,013	0,113	0,546
GRU	0.3	0,083	0,010	0,103	0,619

Pengujian *dropout* menunjukkan bahwa tingkat *dropout* yang lebih tinggi membantu secara efektif mencegah *overfitting*, memastikan model memiliki kemampuan generalisasi yang lebih baik terhadap data yang belum terlihat. Hasil terbaik diperoleh pada *dropout* 0.3 di model LSTM, dengan nilai MAE 0.08473 dan R² 0.65601.



Gambar 10 Grafik Training Loss LSTM *Dropout* 0.2 vs 0.3



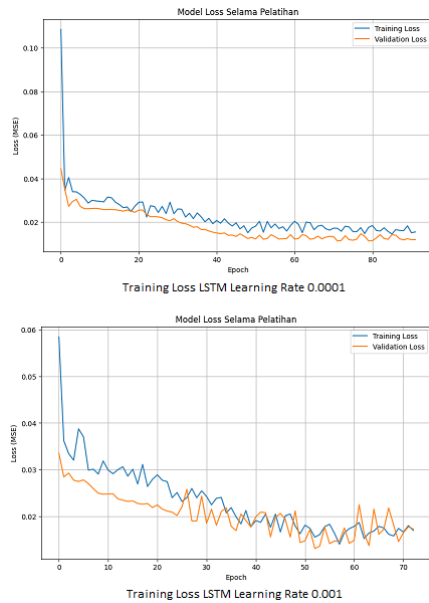
Gambar 11 Grafik Training Loss GRU Dropout 0.2 vs 0.3

3.5.6. Variasi Learning Rate (0.0001 vs 0.001)

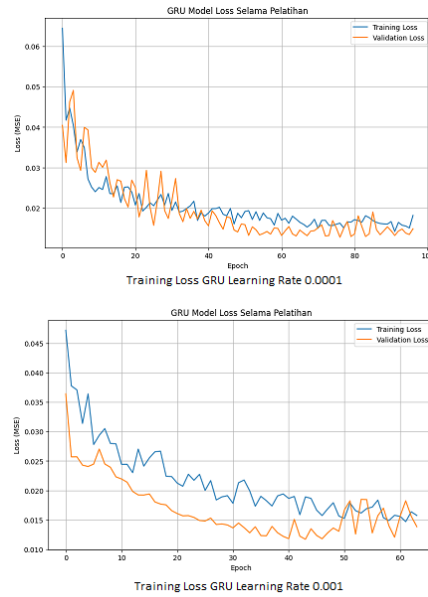
Tabel 8. Perbandingan Model Dengan Learning Rate 0.0001 vs 0.001

Model	Learning Rate	MAE	MSE	RSME	R ²
LSTM	0.0001	0,084	0,010	0,100	0,650
LSTM	0.001	0,101	0,013	0,114	0,534
GRU	0.0001	0,086	0,013	0,113	0,546
GRU	0.001	0,089	0,011	0,108	0,585

Eksperimen dengan learning rate menegaskan bahwa *learning rate* yang lebih rendah, yaitu 0.0001, memberikan proses pelatihan yang lebih stabil dan membantu menghindari *overshooting* (melampaui titik optimal). Semua model yang diuji menunjukkan konvergensi yang baik dengan *learning rate* ini, mengindikasikan bahwa nilai ini optimal untuk memastikan pelatihan yang terkontrol dan efektif.



Gambar 12 Training Loss LSTM Learning Rate 0.0001 vs 0.001



Gambar 13 Training Loss GRU Learning Rate 0.0001 vs 0.001

Melalui skenario pengujian yang sistematis ini, kami berhasil mengidentifikasi kombinasi *hyperparameter* yang paling efektif untuk model prediksi curah hujan bulanan di Kota Tangerang. Informasi ini menjadi landasan kuat untuk pengembangan lebih lanjut.

3.6. Analisis Kinerja Model

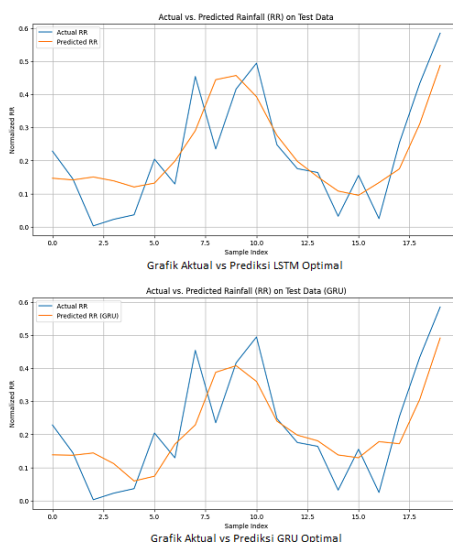
Setelah serangkaian skenario pengujian yang komprehensif, kami melakukan analisis mendalam terhadap kinerja kedua arsitektur model, yaitu LSTM dan GRU, untuk mengidentifikasi model prediksi curah hujan bulanan terbaik. Berdasarkan semua metrik evaluasi yang digunakan, model LSTM secara konsisten menunjukkan performa yang *superior*.

Tabel 9. Perbandingan Model Optimal

Model	LSTM	GRU
Timesteps	36	36
Unit Memori	64	64
Epoch	100	100
Batch Size	16	16
Dropout	0.3	0.3
Learning Rate	0.0001	0,0001
MAE	0,084	0,083
MSE	0,009	0,010
RSME	0,098	0,103
R ²	0,656	0,619

Model LSTM dengan konfigurasi *hyperparameter* optimal—terdiri dari *timesteps* 36 bulan, 64 *unit memori*, 100 *epoch* pelatihan, *batch size* 16, *dropout* 0.3, dan *learning rate* 0.0001—menghasilkan metrik evaluasi terbaik: MAE (*Mean Absolute Error*) sebesar 0.08473, MSE (*Mean Squared Error*) sebesar 0.00973, RMSE (*Root Mean Squared Error*) sebesar 0.09863, dan R² *Score* sebesar 0.65601. Nilai R² yang relatif tinggi ini mengindikasikan bahwa model LSTM mampu menjelaskan sekitar 65.6% dari variabilitas dalam data curah hujan aktual.

Sebagai perbandingan, model GRU dengan kinerja terbaiknya (menggunakan *hyperparameter* yang sama) menunjukkan metrik evaluasi yang sedikit di bawah LSTM, yaitu MAE 0.08386, MSE 0.01076, RMSE 0.10372, dan R² *Score* 0.61955. Meskipun GRU memberikan hasil yang kompetitif pada beberapa konfigurasi spesifik, secara keseluruhan, LSTM terbukti lebih unggul dalam kapabilitas prediksinya.



Gambar 14 Grafik Aktual vs Prediski Model Optimal

Dapat disimpulkan bahwa LSTM memiliki keunggulan dalam menangkap pola musiman yang kompleks secara lebih efektif. Kemampuan ini sangat terlihat ketika model diberikan konteks historis yang cukup panjang (36 bulan) dan dikombinasikan dengan teknik regularisasi seperti *dropout*. Penggunaan *dropout* sebesar 0.3 pada LSTM membantu mencegah *overfitting*, memastikan model mampu menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Selain itu, model LSTM juga menunjukkan konvergensi pelatihan yang lebih stabil. Hal ini tercermin dari kurva *training loss* dan *validation loss* yang sejajar dan menurun secara konsisten selama pelatihan, menandakan bahwa model belajar dengan baik tanpa mengalami *overfitting* yang signifikan.

Dengan demikian, model LSTM menjadi pilihan yang lebih *robust* dan akurat untuk prediksi curah hujan bulanan di Kota Tangerang, memberikan landasan yang kuat untuk aplikasi praktis.

4. Kesimpulan

Dalam berfokus pada pengembangan model prediksi curah hujan bulanan di Kota Tangerang, memanfaatkan arsitektur jaringan saraf berulang (RNN) seperti *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU). Dengan menggunakan data historis curah hujan dan fitur cuaca dari BMKG, serangkaian eksperimen ekstensif dilakukan untuk menemukan kombinasi *hyperparameter* optimal yang menghasilkan performa terbaik.

Berdasarkan hasil eksperimen dan analisis yang mendalam, beberapa kesimpulan penting dapat ditarik:

Pertama, Model LSTM menunjukkan performa yang *superior* dibandingkan GRU dalam menangkap pola data curah hujan. Ini dibuktikan oleh nilai evaluasi terbaik yang dicapai oleh konfigurasi LSTM, dengan nilai MAE 0.084, MSE 0.009, RMSE 0.098, dan R2 sebesar 0.656.

Kedua, pemilihan *hyperparameter* memiliki pengaruh yang sangat signifikan terhadap performa model. Kombinasi terbaik yang ditemukan adalah pada *Timesteps* 36, *Unit* 64, *Epoch* 100, *Batch size* 16, *Dropout* 0.3, dan *Learning rate* 0.0001.

Ketiga, *Dropout* dan *learning rate* terbukti sangat efektif dalam mencegah *overfitting*. *Dropout* sebesar 0.3 memberikan generalisasi yang lebih baik, sementara *learning rate* 0.0001 memastikan proses pelatihan yang stabil dan konvergen.

Keempat, model menunjukkan kemampuan adaptasi yang baik meskipun dataset terbatas secara temporal. Hal ini mengindikasikan bahwa arsitektur LSTM sangat cocok untuk prediksi deret waktu dengan data historis yang mungkin terbatas, namun memiliki pola musiman atau korelasi jangka panjang yang kuat.

Terakhir, model prediksi yang dikembangkan ini memiliki potensi besar untuk dijadikan dasar sistem peringatan dini atau alat pengambilan keputusan berbasis cuaca. Aplikasinya dapat mencakup berbagai sektor, seperti pertanian, pengelolaan air, dan mitigasi bencana.

Berdasarkan pencapaian dalam penelitian ini, ada beberapa saran yang dapat menjadi referensi untuk penelitian

selanjutnya:

Pertama, penambahan data eksternal seperti indeks iklim global (contohnya ENSO, IOD) atau data spasial dari wilayah sekitarnya berpotensi besar untuk meningkatkan akurasi prediksi dan memperkaya konteks informasi model.

Kedua, eksplorasi arsitektur lain sangat direkomendasikan. Model seperti *Bidirectional LSTM*, *attention-based models*, atau kombinasi CNN-LSTM dapat diuji untuk menangkap pola *spasial-temporal* yang lebih kompleks.

Ketiga, untuk meningkatkan nilai praktis, integrasi model ke dalam aplikasi berbasis *web* atau *mobile* sangat disarankan. Hal ini akan memungkinkan hasil prediksi dapat diakses dan dimanfaatkan secara langsung oleh instansi terkait atau masyarakat umum.

Keempat, penggunaan teknik *ensemble* atau *model hybrid* yang menggabungkan pendekatan *deep learning* dengan metode statistik tradisional (misalnya ARIMA-LSTM) dapat menjadi solusi untuk meningkatkan stabilitas prediksi dan mengurangi kesalahan ekstrem.

Terakhir, validasi silang (*cross-validation*) dan eksperimen multi-lokasi dapat dilakukan untuk memastikan generalisasi model terhadap wilayah dengan pola curah hujan yang berbeda.

Dengan menerapkan saran-saran ini, diharapkan penelitian di bidang prediksi curah hujan berbasis deep learning dapat terus berkembang dan memberikan kontribusi nyata dalam mitigasi risiko iklim serta pengambilan keputusan berbasis data di masa depan.

Daftar Rujukan

- [1] A. Luthfiarta, A. Febriyanto, H. Lestiawan, and W. Wicaksono, "Analisa Prakiraan Cuaca dengan Parameter Suhu, Kelembaban, Tekanan Udara, dan Kecepatan Angin Menggunakan Regresi Linear Berganda," *JOINS (Journal Inf. Syst.*, vol. 5, no. 1, pp. 10–17, 2020, doi: 10.33633/joins.v5i1.2760.
- [2] I. W. A. Suranata, "Pengembangan Model Prediksi Curah Hujan di Kota Denpasar Menggunakan Metode LSTM dan GRU," *J. Sist. dan Inform.*, vol. 18, no. 1, pp. 64–73, 2023, doi: 10.30864/jsi.v18i1.603.
- [3] D. Sangaji and T. Sutabri, "Analisis XGBoost dan Random Forest untuk Prediksi Curah Hujan dalam Mendukung Mitigasi Karhutla," vol. 5, no. 1, pp. 13–18, 2025.
- [4] D. Eka Putra and R. Ikhbal Salam, "Prediksi Penjualan Gas Menggunakan Metode Monte Carlo," *J. Pustaka AI (Pusat Akses Kaji. Teknol. Artif. Intell.*, vol. 4, no. 1, pp. 26–30, 2024, doi: 10.55382/jurnalpustakaai.v4i1.754.
- [5] M. Musfiroh, D. C. R. Novitasari, P. K. Intan, and G. G. Wisnawa, "Penerapan Metode Principal Component Analysis (PCA) dan Long Short-Term Memory (LSTM) dalam Memprediksi Prediksi Curah Hujan Harian," *Build. Informatics, Technol. Sci.*, vol. 5, no. 1, pp. 1–11, 2023, doi: 10.47065/bits.v5i1.3114.
- [6] M. Rizki, S. Basuki, and Y. Azhar, "Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang," *J. Repos.*, vol. 2, no. 3, pp. 331–338, 2020, doi: 10.22219/repositor.v2i3.470.
- [7] D. R. Rochmawati, "Prediksi Cuaca Dengan Jaringan Syaraf Tiruan Menggunakan Python," *J. Teknol. Komput. dan Inform.*, vol. 2, no. 2, pp. 162–171, 2024, doi: 10.59820/tekomin.v2i2.228.
- [8] I. Farisi, J. Shadiq, W. Priyadi, D. Maulana, and F. Sonia, "Penerapan Model Recurrent Neural Network (RNN) untuk Prediksi Curah Hujan Berbasis Data Historis," vol. 9, no. 2, pp. 217–226, 2024.
- [9] M. Hasanudin, S. Dwiasnati, and W. Gunawan, "Pelatihan Datascience pada Pra-Pemrosesan Data untuk Siswa SMK Media Informatika - Jakarta Datascience Training on Data Pre-Processing for Media Informatics Vocational School Students - Jakarta," vol. 9, no. 4, pp. 882–888, 2024.
- [10] H. Rusanto and S. Soekirno, "Performance Comparison of 1D-CNN and LSTM Deep Learning Models for Time Series-Based Electric Power Prediction," vol. 13, no. 1, pp. 44–56, 2025.
- [11] A. A. Rizal and S. Soraya, "MULTI TIME STEPS PREDICTION DENGAN RECURRENT NEURAL," vol. 18, no. 1, pp. 115–124, 2018.
- [12] W. Musu *et al.*, "Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4 . 5," vol. X, no. 1, pp. 186–195.
- [13] M. M. Öztürk, "Hyperparameter Optimization of a Parallelized LSTM for Time Series Prediction," *Vietnam J. Comput. Sci.*, vol. 10, no. 3, pp. 303–328, 2023, doi: 10.1142/S2196888823500033.
