Analisis Perbandingan Klasifikasi dalam *Data Mining* pada Prediksi Hujan dengan menggunakan Algoritma *LSTM* dan *GRU*

D. Diffran Nur Cahyo[1)], Andi Sunyoto[2)]

[1)] Informatika, Fakultas Sains dan Komunikasi Visual, Universitas Cendekia Mitra Indonesia
Yogyakarta, Indonesia
[1)] diffrannurcahyo@gmail.com

[2)] Magister Informatika, Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
[2)] andi@amikom.ac.id

Abstrak

Hujan memiliki dampak signifikan terhadap berbagai sektor, seperti pertanian, transportasi, dan manajemen sumber daya. Oleh karena itu, akurasi prediksi hujan sangat penting untuk mendukung perencanaan dan pengambilan keputusan yang efektif. Seiring kemajuan teknologi, pemanfaatan metode deep learning seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) semakin berkembang dalam bidang prediksi cuaca. Penelitian ini bertujuan untuk membandingkan performa kedua algoritma tersebut dalam mengklasifikasikan data meteorologi untuk prediksi hujan. Dataset yang digunakan berasal dari Australian Bureau of Meteorology melalui Kaggle, terdiri dari 145.460 data dengan 23 atribut. Setelah melalui tahapan pra-pemrosesan dan resampling menggunakan metode SMOTE, kedua model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil menunjukkan bahwa model LSTM memperoleh akurasi sebesar 81,46%, sedangkan GRU sebesar 81,39%. Nilai F1-score GRU lebih tinggi dibandingkan LSTM, masing-masing sebesar 60,96% dan 58,95%. Hasil ini mengindikasikan bahwa kedua model memiliki performa yang kompetitif dan efektif untuk diterapkan dalam sistem prediksi hujan berbasis deep learning.

Kata kunci: Klasifikasi, *Data Mining*, Prediksi Hujan, *LSTM*, *GRU*

Abstract

Rain has a significant impact on various sectors such as agriculture, transportation, and resource management. Therefore, accurate rainfall prediction is crucial for effective planning and decision-making. With technological advancements, deep learning methods such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have become increasingly popular in weather prediction tasks. This study aims to compare the performance of these two algorithms in classifying meteorological data for rainfall prediction. The dataset used was obtained from the Australian Bureau of Meteorology via Kaggle, consisting of 145,460 records with 23 attributes. After undergoing preprocessing and resampling using the SMOTE method, both models were evaluated using accuracy, precision, recall, and F1-score metrics. The results showed that the LSTM model achieved an accuracy of 81.46%, while the GRU model achieved 81.39%. However, GRU obtained a higher F1-score (60.96%) compared to LSTM (58.95%), indicating better detection capability for rainfall occurrences. These findings suggest that both models are competitive and effective for implementing deep learning-based rainfall prediction systems.

Keywords: *Classification, Data Mining, Rain Prediction, LSTM, GRU*

1. PENDAHULUAN

Hujan adalah fenomena meteorologis penting yang memiliki dampak substansial terhadap berbagai aspek kehidupan manusia, termasuk pertanian, manajemen sumber daya air, dan mitigasi bencana. Akurasi dalam prediksi hujan esensial untuk perencanaan dan pengambilan keputusan yang efektif [1]. Dalam dekade

DOI: 10.34128/jsi.v11i1.1212 *Received: November 21st, 2024*
Accepted: July 15th, 2025

terakhir, peningkatan data dan kemajuan dalam teknologi komputasi telah mengarah pada pengembangan model prediksi cuaca yang lebih canggih, termasuk penggunaan teknik *machine learning* [2].

Klasifikasi adalah proses mengelompokkan objek ke dalam kelas berdasarkan kemiripan fitur [3]. Dalam konteks ilmiah, klasifikasi digunakan untuk mengidentifikasi, memisahkan, dan menganalisis data secara sistematis [4]. Metode ini penting dalam berbagai bidang penelitian karena membantu dalam pengambilan keputusan dan prediksi berdasarkan informasi yang telah dikategorikan [5]. Misalnya, dalam penelitian tentang ras kucing [6], klasifikasi membantu mengidentifikasi dan mengelompokkan kucing berdasarkan karakteristik tertentu. Disisi lain, klasifikasi pada prediksi hujan menggunakan metode konvensional seringkali terbatas oleh kompleksitas spasial dan temporal dari data cuaca [7]. Dalam dekade terakhir, metode deep learning seperti *Convolutional Neural Network (CNN)* [8] dan khususnya *Recurrent Neural Networks (RNN)* [9], telah menunjukkan potensi yang signifikan dalam memprediksi fenomena berbasis waktu seperti cuaca. Dua variasi dari RNN yang paling menonjol dalam penelitian ini adalah *Long Short-Term Memory (LSTM)* [10] dan *Gated Recurrent Unit (GRU)* [11]. Keduanya dirancang untuk mengatasi tantangan yang berkaitan dengan dependensi jangka panjang dalam data berurutan yang sering ditemukan dalam data cuaca.

LSTM telah menjadi dasar bagi banyak inovasi dalam pemrosesan data berurutan. Performa utama *LSTM* terletak pada kemampuannya untuk mempertahankan informasi untuk jangka waktu yang panjang, membuatnya ideal untuk analisis data deret waktu. Disisi lain, *GRU* merupakan pendekatan yang lebih baru, diperkenalkan oleh [12]. *GRU* menawarkan arsitektur yang lebih sederhana dibandingkan dengan *LSTM*, yang memungkinkan pelatihan yang lebih cepat dengan mempertahankan efektivitas yang serupa. Dalam penelitian terkini menunjukkan bahwa kedua *LSTM* dan *GRU* efektif dalam memodelkan data deret waktu [13]–[15], termasuk dalam aplikasi meteorologi. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan efektivitas kedua model tersebut dalam konteks prediksi hujan. Menggunakan dataset yang komprehensif dari Meteorologi Australia, yang mencakup 145.460 baris data dengan 23 kolom, penelitian ini berusaha untuk memperoleh pemahaman yang lebih dalam tentang potensi *LSTM* dan *GRU* dalam peningkatan performa klasifikasi untuk prediksi hujan. Hasil dari penelitian ini diharapkan memberikan kontribusi signifikan pada aplikasi praktis *machine learning* dalam meteorologi, khususnya dalam memperbaiki akurasi dan reliabilitas prediksi hujan.

2. TINJAUAN PUSTAKA

Beberapa penelitian telah mengeksplorasi prediksi hujan dengan metodologi dan hasil yang beragam. Penelitian [16] menggunakan data BMKG Indonesia dan empat metode *data mining*, menemukan bahwa *Multilayer Perceptron* dan *J48* memberikan akurasi tertinggi sebesar 78.4%. Penelitian ini menekankan pentingnya dataset pelatihan yang besar dan memilih teknik data mining yang tepat.

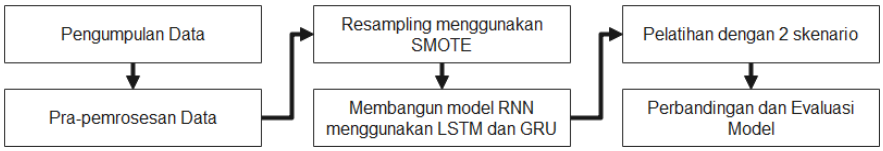
Penelitian [17] memanfaatkan data cuaca sepuluh tahun dari BMKG dengan metode *1D CNN*, menghasilkan akurasi 81.46%, menunjukkan pentingnya normalisasi data yang tepat. Penelitian [18] menunjukkan efektivitas *Support Vector Machine (SVM)* dalam memprediksi hujan di Tanjungpinang dengan presisi 82%. Penelitian [19] mengaplikasikan teknik *machine learning* pada data curah hujan di Australia, dengan *neural networks* sebagai model terbaik yang mencapai akurasi 85%. Penelitian terakhir [20] menggunakan algoritma *deep learning* pada data dari 49 stasiun cuaca di Australia, dengan *Random Fores*t menunjukkan performa terbaik dengan akurasi 85.6416% dan *F1-score* 61.8736%.

Berdasarkan tinjauan, penelitian ini bertujuan untuk mengklasifikasikan prediksi hujan menggunakan *LSTM* dan *GRU* sebagai peningkatan pada metode sebelumnya.

3. METODE PENELITIAN

Metodologi yang digambarkan pada Gambar 1 tersebut adalah proses yang digunakan dalam penelitian untuk mengembangkan model yang mampu melakukan prediksi atau klasifikasi. Metode tersebut dimulai dengan pengumpulan data, data tersebut diolah melalui tahap pra-pemrosesan, *resampling* menggunakan *SMOTE*, membangun model menggunakan *LSTM* dan GRU, kemudian dilakukan pelatihan dengan dua skenario yang berbeda, dan terakhir kedua model tersebut dikomparasikan serta dievaluasi menggunakan *confusion matrix*.

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 1 Metode Penelitian

3.1 Pengumpulan Data

Data cuaca historis harian diperoleh dari situs *Kaggle (https://www.kaggle.com/datasets/trisha2094/weatheraus)*, Kumpulan data ini berisi pengamatan cuaca harian selama sekitar 10 tahun dari banyak lokasi di seluruh Australia. Data asli yang diambil dari *Bureau of Meteorology (BoM)* terdiri dari 23 atribut yang ditunjukkan pada Tabel 1.

Tabel 1. Atribut, tipe data, dan deskripsi dari pengumpulan data

Atribut	*Tipe Data*	*Deskripsi*
Date	*Object*	Waktu (tanggal, bulan, tahun).
Location	*Object*	Nama kota dari Australia.
MinTemp	*Float*	Suhu minimum pada hari tertentu (derajat *Celcius*).
MaxTemp	*Float*	Suhu maksimum pada hari tertentu (derajat *Celcius*).
Rainfall	*Float*	Curah hujan pada hari tertentu (milimeter).
Evaporation	*Float*	selama hari tertentu (milimeter).
Sunshine	*Float*	Sinar matahari cerah pada hari tertentu (jam).
WindGustDir	*Object*	Arah hembusan terkuat pada hari tertentu (16 titik kompas).
WindGustSpeed	*Float*	Kecepatan hembusan terkuat pada hari tertentu (kilometer per jam).
WindDir9am	*Object*	Arah angin selama 10 menit sebelum jam 9 pagi (titik kompas).
WindDir3pm	*Object*	Arah angin selama 10 menit sebelum jam 3 sore (titik kompas).
WindSpeed9am	*Float*	Kecepatan angin selama 10 menit sebelum jam 9 pagi (kilometer per jam).
WindSpeed3pm	*Float*	Kecepatan angin selama 10 menit sebelum jam 3 sore (kilometer per jam).
Humidity9am	*Float*	Kelembaban angin pada jam 9 pagi (persen).
Humidity3pm	*Float*	Kelembaban angin pada jam 3 sore (persen).
Pressure9am	*Float*	Tekanan atmosfer pada jam 9 pagi (*hektopaskal).*
Pressure3pm	*Float*	Tekanan atmosfer pada pukul 3 sore (*hektopaskal*).
Cloud9am	*Float*	Bagian langit yang tertutup awan pada pukul 9 pagi.
Cloud3pm	*Float*	Bagian langit yang tertutup awan pada pukul 3 sore.
Temp9am	*Float*	Suhu pada jam 9 pagi (derajat *Celcius*).
Temp3pm	*Float*	Suhu pada jam 3 sore (derajat *Celcius*).
RainToday	*Object*	Jika hari ini hujan maka 'Ya'. Jika hari ini tidak hujan maka 'Tidak'.
RainTomorrow	*Object*	Jika besok hujan maka 1 (Ya). Jika besok tidak hujan maka 0 (Tidak).

3.2 Pra-pemrosesan Data

Tahap pra-pemrosesan dalam data mining adalah proses penting yang melibatkan serangkaian langkah untuk mengubah data mentah menjadi format yang lebih sesuai dan efisien untuk analisis [21]. Proses ini dapat meningkatkan kualitas dan efektivitas analisis data mining dengan mengeliminasi gangguan dan anomali yang dapat mengurangi akurasi model. Tahapan pra-pemrosesan pada penelitian ini meliputi:

a. Transformasi data: Langkah awal melibatkan konversi struktural kolom *Date* ke format *datetime*, sebuah proses standardisasi yang esensial untuk memfasilitasi manipulasi temporal yang konsisten dalam analisis data. Subsekuennya, prosedur ini melanjutkan dengan dekomposisi entitas temporal tersebut ke dalam entitas yang lebih granular, yaitu hari, bulan, dan tahun, merepresentasikan setiap entitas dalam kolom terpisah. Inisiatif ini mengkatalisasi kemampuan analitis dengan menyediakan akses ke komponen waktu secara diskrit.
b. Mengubah data kategorikal menjadi format numerik: Pertama menggunakan *LabelEncoder* untuk mengonversi nilai-nilai dalam kolom *Location, WindGustDir, WindDir9am*, dan *WindDir3pm* menjadi bentuk numerik, memudahkan pengolahan data dalam pra-pemrosesan. Selanjutnya, untuk kolom *RainToday* dan *RainTomorrow* mengganti teks *Yes* dan *No* dengan nilai biner 1 dan 0, sebuah langkah sederhana namun efektif untuk mengolah data *Boolean*.
c. Mengatasi masalah nilai yang hilang *(NaN)*: Pertama, menghitung nilai median dari setiap kolom dengan fungsi median. Nilai median dipilih karena merupakan ukuran tendensi sentral yang tahan terhadap *outlier* dan dapat merepresentasikan distribusi data dengan lebih baik dalam beberapa kasus, terutama jika data

tidak simetris atau memiliki *outlier*. Selanjutnya menggunakan fungsi *fillna* untuk mengganti semua nilai *NaN* dengan nilai median yang sesuai dari setiap kolom. Penggunaan *inplace=True* mengindikasikan bahwa perubahan ini dilakukan langsung pada *DataFrame* asli tanpa perlu membuat salinan baru.

d. Reorganisasi struktur kolom dalam *DataFrame*: Menyusun kembali data dalam urutan yang lebih logis, intuitif, atau efisien untuk analisis atau pemodelan data. Mengatur ulang kolom dapat membantu dalam meningkatkan keterbacaan dan aksesibilitas data, serta memastikan bahwa data diatur sesuai dengan prioritas analisis yang telah ditentukan.

Tabel 2 menunjukkan atribut-atribut yang akan digunakan setelah melalui tahap pra-pemrosesan.

Tabel 2. Atribut yang digunakan setelah melalui tahap pra-pemrosesan data

Atribut dan Tipe Data		*Atribut dan Tipe Data*	
Day	*(Int)*	*WindSpeed9am*	*(Float)*
Month	*(Int)*	*WindSpeed3pm*	*(Float)*
Year	*(Int)*	*Humidity9am*	*(Float)*
Location	*(Int)*	*Humidity3pm*	*(Float)*
MinTemp	*(Float)*	*Pressure9am*	*(Float)*
MaxTemp	*(Float)*	*Pressure3pm*	*(Float)*
Rainfall	*(Float)*	*Cloud9am*	*(Float)*
Evaporation	*(Float)*	*Cloud3pm*	*(Float)*
Sunshine	*(Float)*	*Temp9am*	*(Float)*
WindGustDir	*(Object)*	*Temp3am*	*(Float)*
WindGustSpeed	*(Float)*	*RainToday*	*(Object)*
WindDir9am	*(Object)*	*RainTomorrow*	*(Object)*
WindDir3pm	*(Object)*		

3.3 Pembagian Data

Penelitian ini membagi dataset menjadi 80% *data training* dan 20% *data testing*.

3.4 Resampling

Pada tahap ini dilakukan resampling menggunakan *SMOTE (Synthetic Minority Over-sampling Technique)*. *SMOTE* adalah metode *resampling* untuk memperbanyak data pada kelas minoritas dengan cara menduplikasikan data atau menggunakan data sintetik yang ada dikelas minoritas dengan tujuan untuk mengurangi ketidakkeseimbangan data [22]. Dengan menggunakan *SMOTE* akan dapat menghindari masalah *overfitting* yang tinggi. Alasan digunakannya *SMOTE* karena data seleteh pra-pemrosesan tergolong *imbalance* atau tidak seimbang. Data sebelum menggunakan resampling *SMOTE* pada label tidak hujan sebanyak 90.911 dan pada label hujan sebanyak 25.457 data. Sedangkan sesudah menggunakan *resampling SMOTE* pada label tidak hujan dan hujan sama-sama memiliki 90.911 data.

3.5 Model

Pada penelitian ini membandingkan model *LSTM* dan *GRU*, seperti yang ditunjukkan pada Gambar 2. *LSTM* merupakan evolusi dari *RNN* yang dikembangkan untuk mengatasi isu *vanishing gradient* dan *exploding gradient* [23]. Isu tersebut muncul signifikan selama *RNN* beroperasi pada *data sequence* yang panjang, mengakibatkan kesulitan bagi *RNN* untuk efektif memahami *long-time dependencies* yang melibatkan langkah-langkah temporal yang ekstensif [24]. Implikasi dari masalah ini termanifestasi sebagai penurunan akurasi pada *RNN*.

Sebagai respons terhadap tantangan tersebut, *LSTM* diintroduksi dengan mekanisme *memory cell* dalam menyimpan informasi jangka panjang [25]. Persamaan (1) merupakan rumus hasil akhir dari *LSTM* yang dimana h_t adalah *final output*, o_t adalah nilai *output gate*, C_t adalah nilai *memory cell state*, dan *tanh* atau tangen hiperbolik adalah sebuah fungsi aktivasi yang sering digunakan dalam *Artificial Neural Network (ANN)* [26], termasuk dalam arsitektur seperti *LSTM*.

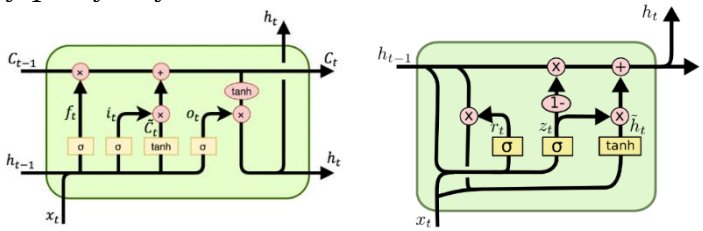
$$h_t = o_t \times tanh\,(C_t) \tag{1}$$

GRU merupakan variasi yang lebih ringkas dari *LSTM*, dimana integrasi fungsi dari *input gate* dan *forget gate* pada *LSTM* terkonsolidasi ke dalam satu mekanisme yang disebut *update gate* [27]. Integrasi ini

p-ISSN: 2460-173X
e-ISSN: 2598-5841

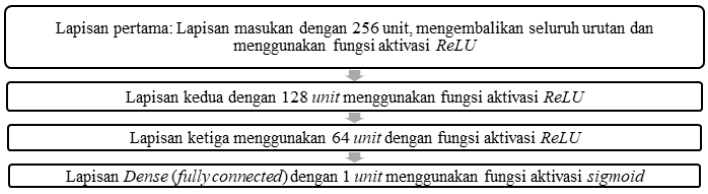
memungkinkan *GRU* untuk beroperasi dengan efisiensi waktu yang lebih tinggi dibandingkan *LSTM*, dengan desain yang dioptimalkan untuk performa superior pada dataset yang lebih terbatas dalam skala [12]. Dalam rumus *GRU* pada persamaan (2), h_t adalah *final output*, z_t mengindikasikan *update gate* pada interval waktu *t*, h_{t-1} merujuk pada *hidden state* dari langkah waktu sebelumnya, dan $\hbar_t$ adalah *hidden state candidate*.

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \hbar_t \quad (2)$$



Gambar 2. Model dari *LSTM* (kiri) dan *GRU* (kanan)

Setelah menentukan model yang digunakan, implementasi model dilakukan. Meskipun algoritma berbeda, parameter yang digunakan akan tetap sama. Pada Gambar 3 merupakan paramater yang digunakan. Penelitian ini juga memanfaatkan *callback* yang berguna untuk mencegah terjadinya *overfitting* pada model.



Gambar 3. Paramater yang digunakan untuk membangun model

Dalam model, fungsi aktivasi *Rectified Linear Unit (ReLU)* diaplikasikan pada *hidden layer*, sesuai dengan pemanfaatannya yang luas dalam *deep learning*. Adapun untuk *layer dense* terakhir, digunakan fungsi aktivasi *sigmoid*, yang efektivitasnya telah terdokumentasi dalam menghasilkan estimasi probabilitas pada *output layer*, khususnya dalam klasifikasi biner [28]. Mengenai *loss function*, penelitian ini mengadopsi *binary_crossentropy*, dipilih sesuai dengan struktur dataset yang hanya terdiri dari dua label. Selain itu, fungsi optimasi yang digunakan adalah *Adam* [29] yang berdasarkan bukti empiris, menunjukkan superioritas dalam beberapa kasus, sering kali mengungguli metode adaptif lainnya [23]. Untuk evaluasi, penelitian ini mengandalkan *metric accuracy* untuk mengukur performa model berdasarkan *data training* dan *validation*.

4. PEMBAHASAN

Penelitian ini menggunakan *LSTM* dan G*RU*, yang dirancang untuk mengolah data sekuensial dalam prediksi kejadian hujan. Kedua model dibangun dengan parameter pelatihan yang seragam, yaitu *batch size* sebesar 32 dan jumlah *epoch* sebanyak 100. Penggunaan parameter yang sama bertujuan untuk memastikan perbandingan performa dilakukan secara adil dan setara dari sisi pengaturan pelatihan. *LSTM* memiliki struktur yang terdiri atas tiga gerbang utama, yaitu *input gate, forget gate,* dan *output gate*, yang membuatnya lebih kompleks dan mampu menangani hubungan jangka panjang dalam data. Sebaliknya, GRU menyederhanakan arsitektur dengan hanya menggunakan dua *gate*, yaitu *update gate,* dan *reset gate*, sehingga cenderung lebih ringan dan efisien secara komputasi.

Tabel 3. Struktur Arsitektur Model LSTM

Layer (Type)	*Output Shape*	*Jumlah Parameter*
LSTM	*(None,* 1, 256*)*	287.744
LSTM_1	*(None,* 1, 128*)*	197.120
LSTM_2	*(None,* 64*)*	49.408
Dense	*(None,* 1*)*	65
Total		534.337

p-ISSN: 2460-173X
e-ISSN: 2598-5841

Tabel 4. Struktur Arsitektur Model GRU

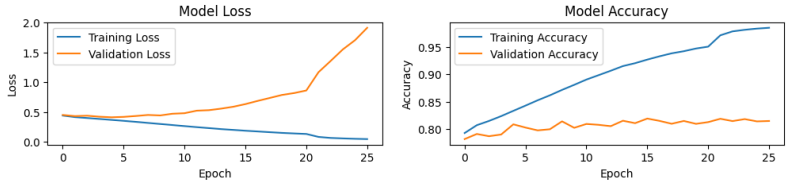
Layer (Type)	*Output Shape*	*Jumlah Parameter*
GRU	*(None,* 1, 256*)*	216.576
GRU_1	*(None,* 1, 128*)*	148.224
GRU_2	*(None,* 64*)*	37.248
Dense	*(None,* 1*)*	65
Total		402.113

Perbedaan jumlah parameter menunjukkan bahwa *LSTM* memiliki kompleksitas arsitektur yang lebih tinggi dibandingkan *GRU*. Hal ini memengaruhi beban komputasi saat pelatihan, meskipun belum tentu berbanding lurus dengan peningkatan akurasi prediksi. Setelah menentukan konfigurasi arsitektur dan jumlah parameter pada masing-masing model, selanjutnya adalah model dilatih menggunakan dataset yang telah di-*resample* menggunakan metode *SMOTE* untuk mengatasi ketidakseimbangan kelas. Evaluasi dilakukan berdasarkan metrik akurasi, presisi, *recall*, dan *F1-score* yang ditunjukkan Tabel 5 berikut.

Tabel 5. Hasil setelah dilakukan model pelatihan

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
LSTM	81,46%	57,85%	60,08%	58,93%
GRU	81,39%	57,07%	64,29%	60,96%

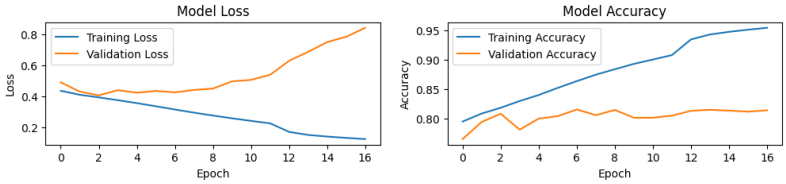
Model *LSTM* menunjukkan akurasi yang sedikit lebih tinggi, sedangkan *GRU* unggul pada nilai recall dan F1-score, yang menandakan kemampuannya dalam mengklasifikasikan data hujan secara lebih konsisten. Setelah melakukan penelitian *LSTM* mendapatkan akurasi yang terbaik yaitu 81,46% dibandingkan dengan *GRU* yang mendapatkan akurasi 81,39%. Hasil penelitian tersebut masih tergolong cukup rendah karena pada penelitian sebelumnya mendapatkan akurasi yang bisa melebihi dari akurasi yang telah diusulkan [17]–[20]. Namun, pada penelitian ini tidak mengalami *overfitting* karena pada grafik *loss* dan akurasi tidak mengalami naik turun yang signifikan. Gambar 4 menunjukkan hasil grafik pada model *LSTM*.



Gambar 4. Grafik *loss* (kiri) dan akurasi (kanan) pada model *LSTM*

Grafik training dan validasi pada Gambar 4 (kiri) adalah *loss* dan (kanan) adalah akurasi. Garis biru merupakan *data training* dan garis oranye merupakan *data validation*. Grafik menunjukkan model ini adalah termasuk baik karena peningkatan pada akurasi dan penurunan pada *loss* disetiap *epoch* mengalami stabil.

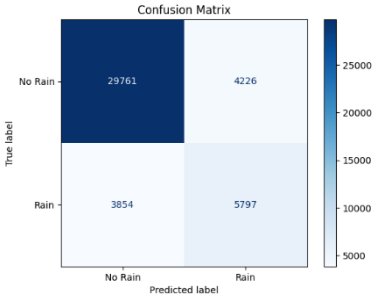
Lalu, Grafik training dan validasi pada model GRU dapat dilihat pada Gambar 5 (kiri) adalah *loss* dan (kanan) adalah akurasi. Garis biru menggambarkan data training, dan garis oranye menggambarkan data validation. Secara umum, grafik ini menunjukkan bahwa model GRU mengalami proses training yang stabil, dengan penurunan loss dan peningkatan akurasi yang konsisten, yang mengindikasikan performa model yang baik dan tidak mengalami overfitting.



Gambar 5. Grafik *loss* (kiri) dan akurasi (kanan) pada model GRU

Setelah memperoleh hasil pelatihan model, langkah selanjutnya adalah melakukan evaluasi menggunakan confusion matrix yang ditampilkan pada Gambar 6 dan Gambar 7, di mana Gambar 6 merupakan confusion matrix untuk model LSTM, dan Gambar 7 menunjukkan confusion matrix untuk model GRU.

p-ISSN: 2460-173X
e-ISSN: 2598-5841



Gambar 6. *Confusion Matrix* pada *LSTM*

Berdasarkan Gambar 6, *confusion matrix* menggambarkan hasil klasifikasi model LSTM terhadap data pengujian. Diperoleh True Positive (TP) sebanyak 5.797 (jumlah data yang diprediksi hujan dan benar-benar hujan). Selanjutnya True Negative (TN) sebanyak 29.761 (data yang diprediksi tidak hujan dan memang tidak hujan). Kemudian False Positive (FP) sebanyak 4.226 (data yang diprediksi hujan tetapi sebenarnya tidak hujan). Terakhir, False Negative (FN) sebanyak 3.854 (data yang diprediksi tidak hujan namun kenyataannya hujan).

Berdasarkan nilai-nilai tersebut, dilakukan perhitungan terhadap beberapa metrik evaluasi model. Pertama, akurasi (*accuracy*) sebagai berikut.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5.797 + 29.761}{5.797 + 29.761 + 4.226 + 3.854} \approx 81{,}46\%$$

Selanjutnya, presisi (*precision*) merupakan rasio antara jumlah prediksi hujan yang benar terhadap seluruh prediksi hujan, yaitu:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{5.797}{5.797 + 4.226} \approx 57{,}85\%$$

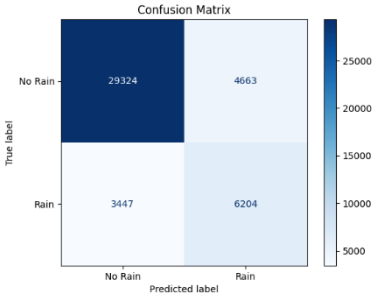
Kemudian, *recall* atau sensitivitas mengukur sejauh mana model mampu mendeteksi seluruh kejadian hujan yang sebenarnya, yaitu:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{5.797}{5.797 + 3.854} \approx 60{,}08\%$$

Terakhir, F1-score digunakan untuk memberikan gambaran seimbang antara presisi dan *recall*, terutama ketika data bersifat tidak seimbang.

$$\text{F1_Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0{,}5785 \times 0{,}6008}{0{,}5785 + 0{,}6008} \approx 58{,}95\%$$

Hasil evaluasi ini menunjukkan bahwa model *LSTM* memiliki performa klasifikasi yang cukup baik, dengan tingkat akurasi dan F1-score yang mendekati hasil model *GRU*. Kedua model sama-sama menunjukkan efektivitas yang kompetitif dalam memprediksi kejadian hujan berdasarkan data historis cuaca.



Gambar 7. Confusion Matrix pada GRU

Berdasarkan Gambar 7, *confusion matrix* menggambarkan hasil klasifikasi model *GRU* terhadap data pengujian. Diperoleh TP sebanyak 6.204, TN sebanyak 29.324, TF sebanyak 4.663 dan FN sebanyak 3.447. Berdasarkan nilai-nilai tersebut, dilakukan perhitungan terhadap beberapa metrik evaluasi model, yaitu *accuracy*, *precision*, *recall* dan *F1-Score*.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{6.204 + 29.324}{6.204 + 29.324 + 4.663 + 3.447} = \frac{35.528}{43.638} \approx 81{,}39\%$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{6.204}{6.204 + 4.663} \approx 57{,}07\%$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{6.204}{6.204 + 3.447} \approx 64{,}29\%$$

$$\text{F1_Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0{,}5707 \times 0{,}6429}{0{,}5707 + 0{,}6429} \approx 60{,}96\%$$

Hasil evaluasi ini menunjukkan bahwa model GRU memiliki performa klasifikasi yang kompetitif terhadap model LSTM, dengan nilai akurasi yang sebanding dan F1-score yang sedikit lebih tinggi. Hal ini menunjukkan kemampuan GRU dalam menangani pola data waktu pada prediksi hujan secara efektif. Sebagai bentuk pembandingan kontekstual, Tabel 6 menyajikan hasil dari beberapa penelitian terdahulu yang menggunakan berbagai pendekatan machine learning untuk melakukan prediksi hujan.

Tabel 6. Hasil perbandingan penelitian dengan penelitian sebelumnya

Author	*Dataset*	*Model*	*Accuracy*
[16]	BMKG Kota Semarang	*C4.5/J48*	78,4%
[17]	BMKG Kota Semarang	*1D CNN*	81,46%
[18]	BMKG Riau	*SVM*	82%
[19]	*Kaggle* (Badan Meteorologi Australia)	*ANN*	85%
[20]	*Kaggle* (Badan Meteorologi Australia)	*KNN*	85,64%
Penelitian ini	*Kaggle* (Badan Meteorologi Australia)	*LSTM*	81.46%
Penelitian ini	*Kaggle* (Badan Meteorologi Australia)	*GRU*	81,39%

Berdasarkan Tabel 6, terlihat beberapa algoritma konvensional seperti *Artificial Neural Network (ANN)* dan *K-Nearest Neighbor (KNN)* mampu menghasilkan akurasi yang lebih tinggi dibandingkan *LSTM* (81,46%) maupun *GRU* (81,39%) pada penelitian ini. Fenomena ini tidak serta-merta menunjukkan kelemahan *deep learning*, melainkan lebih mengindikasikan efektivitas suatu algoritma sangat bergantung pada karakteristik data, konfigurasi model, dan proses pelatihan yang digunakan.

Model *LSTM* dan *GRU* memiliki kompleksitas arsitektur yang memungkinkan untuk menangkap hubungan jangka panjang dalam data sekuensial. Namun demikian, model tersebut sangat bergantung pada proses pemilihan *hyperparameter*, struktur jaringan (jumlah layer dan neuron), serta strategi regularisasi dan optimasi. Tanpa desain yang optimal, model *deep learning* berpotensi mengalami *underfitting* atau bahkan tidak menunjukkan performa superior yang signifikan. Sebaliknya, algoritma klasik seperti *KNN* atau *SVM* cenderung bekerja lebih efisien pada dataset yang bersih, seimbang, dan memiliki fitur-fitur prediktif yang kuat, sehingga mampu menghasilkan akurasi yang lebih tinggi dalam konteks tertentu.

Dengan demikian, meskipun pendekatan berbasis *deep learning* memiliki keunggulan konseptual, namun dalam praktiknya masih memerlukan optimasi dan adaptasi yang cermat terhadap karakteristik data. Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi arsitektur model yang lebih kompleks, melakukan *tuning hyperparameter* secara lebih menyeluruh, serta mempertimbangkan penggunaan pendekatan *ensemble* atau *hybrid* guna meningkatkan akurasi dan stabilitas model prediksi hujan.

5. KESIMPULAN

Penelitian ini membandingkan performa model *LSTM* dan *GRU* dalam melakukan klasifikasi prediksi hujan menggunakan data meteorologi dari *Australian Bureau of Meteorology*. Setelah melalui

tahapan pra-pemrosesan, *resampling* menggunakan *SMOTE*, dan proses pelatihan model, diperoleh hasil bahwa *LSTM* menghasilkan akurasi sebesar 81,46% dan *GRU* sebesar 81,39%. Meskipun nilai akurasi *LSTM* sedikit lebih tinggi, model *GRU* menunjukkan nilai *F1-score* yang lebih besar, yaitu 60,96% dibandingkan *LSTM* sebesar 58,95%. Hal ini menunjukkan bahwa *GRU* memiliki kemampuan deteksi terhadap kejadian hujan yang lebih baik. Secara keseluruhan, kedua model memberikan performa yang kompetitif dan layak diterapkan dalam sistem prediksi cuaca berbasis *deep learning*.

Untuk pengembangan lebih lanjut, disarankan agar penelitian berikutnya mencoba arsitektur model yang lebih kompleks serta menggunakan kombinasi data dari berbagai sumber seperti BMKG. Penelitian juga dapat diperluas dengan mengevaluasi model lain seperti *CNN, BiLSTM,* atau *hybrid* model untuk mendapatkan performa yang lebih optimal dan generalisasi yang lebih baik terhadap variasi pola cuaca. Dari penelitian sebelumnya [19] dan [20] menggunakan dari dataset yang sama karena pada dataset tersebut datanya bisa dibilang seimbang. Selain dataset, kami menyarankan untuk menggunakan metode dan model yang lainnya agar hasilnya bisa beragam.

DAFTAR PUSTAKA

[1] A. Parmar, K. Mistree, and M. Sompura, "Machine Learning Techniques For Rainfall Prediction: A Review Machine Learning View project Rainfall prediction using ANN View project Machine Learning Techniques For Rainfall Prediction: A Review," *Int. Conf. Innov. Inf. Embed. Commun. Syst.*, no. 3, p. 6, 2017.

[2] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 802–810, 2015.

[3] F. A. D. Aji Prasetya Wibawa, Muhammad Guntur Aji Purnama, Muhammad Fathony Akbar, "Metode-metode Klasifikasi," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 1, p. 134, 2018.

[4] A. A. Soofi and A. Awan, "Classification Techniques in Machine Learning: Applications and Issues," *J. Basic Appl. Sci.*, vol. 13, pp. 459–465, 2017.

[5] B. Mahesh, "Machine learning Algorithms-A Review," *Int. J. Sci. Res. (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/ART20203995.

[6] D. D. N. Cahyo, M. A. Fauzi, and J. T. Nugroho, "Analisis Perbandingan Optimizer pada Arsitektur NASNetMobile Convolutional Neural Network untuk Klasifikasi Ras Kucing," vol. 15, pp. 171–177, 2022.

[7] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From action to activity: Sensor-based activity recognition," *Neurocomputing*, vol. 181, pp. 108–115, 2016, doi: 10.1016/j.neucom.2015.08.096.

[8] D. Diffran Nur Cahyo, A. Sunyoto, and D. Ariatmanto, "Transfer Learning and Fine-tuning Effect Analysis on Classification of Cat Breeds using a Convolutional Neural Network," *2023 6th Int. Conf. Inf. Commun. Technol. ICOIACT 2023*, pp. 488–493, 2023, doi: 10.1109/ICOIACT59844.2023.10455771.

[9] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent Advances in Recurrent Neural Networks," pp. 1–21, 2017, [Online]. Available: http://arxiv.org/abs/1801.01078.

[10] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," pp. 1–42, 2019, [Online]. Available: http://arxiv.org/abs/1909.09586.

[11] R. Dey and F. M. Salem, "Gate-Variants of Gated Recurrent Unit (GRU)," *Midwest Symp. Circuits Syst. Inst. Electr. Electron. Eng. Inc.*, vol. 784, no. 2017, pp. 1597–1600, 2017.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.

[13] B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis, and A. M. Cardoso, "Comparing LSTM and GRU models to predict the condition of a pulp paper press," *Energies*, vol. 14, no. 21, pp. 1–21, 2021, doi: 10.3390/en14216958.

[14] W. Surta, K. T. Basuki, N. E. Surya, and ..., "Rainfall Prediction in Palembang City Using the GRU and LSTM Methods," *J. Data ...*, vol. 2023, 2023, [Online]. Available: http://eprints.intimal.edu.my/1730/.
[15] N. Gruber and A. Jockisch, "Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?," *Front. Artif. Intell.*, vol. 3, no. June, pp. 1–6, 2020, doi: 10.3389/frai.2020.00040.
[16] M. T. Anwar, W. Hadikurniawati, E. Winarno, and W. Widiyatmoko, "Performance Comparison of Data Mining Techniques for Rain Prediction Models in Indonesia," *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2020*, pp. 83–88, 2020, doi: 10.1109/ISRITI51436.2020.9315460.
[17] Y. R. Sari, E. C. Djamal, and F. Nugraha, "Daily Rainfall Prediction Using One Dimensional Convolutional Neural Networks," *2020 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020*, pp. 90–95, 2020, doi: 10.1109/IC2IE50715.2020.9274572.
[18] N. Hayaty, H. Kurniawan, M. R. Rathomi, F. Chahyadi, and M. Bettiza, "Rainfall Prediction with Support Vector Machines: A Case Study in Tanjungpinang City, Indonesia," *BIO Web Conf.*, vol. 70, p. 01003, 2023, doi: 10.1051/bioconf/20237001003.
[19] D. Mahajan and S. Sharma, "Prediction of Rainfall Using Machine Learning," *4th Int. Conf. Emerg. Res. Electron. Comput. Sci. Technol. ICERECT 2022*, 2022, doi: 10.1109/ICERECT56837.2022.10059679.
[20] H. Duan, Y. Li, Q. Li, Y. Wang, Y. Xie, and H. Peng, "Application and Analysis of Machine Learning Based Rainfall Prediction," *2023 8th Int. Conf. Intell. Comput. Signal Process. ICSP 2023*, no. September, pp. 1941–1949, 2023, doi: 10.1109/ICSP58490.2023.10248891.
[21] Suad A. Alasadi and Wesam S. Bhaya, "Review_of_Data_Preprocessing_Techniques," *Jurnal of Engineering and Applied Sciences*, vol. 12, no. 16. pp. 4102–4107, 2017.
[22] K. Li, W. Zhang, Q. Lu, and X. Fang, "An improved SMOTE imbalanced data classification method based on support degree," *Proc. - 2014 Int. Conf. Identification, Inf. Knowl. Internet Things, IIKI 2014*, pp. 34–38, 2014, doi: 10.1109/IIKI.2014.14.
[23] N. K. Manaswi, *Deep Learning with Applications Using Python*. 2018.
[24] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustain.*, vol. 10, no. 10, 2018, doi: 10.3390/su10103765.
[25] A. Hanifa and S. Akbar, "Detection of Unstable Approaches in Flight Track with Recurrent Neural Network," *2018 Int. Conf. Inf. Commnunications Technol.*, pp. 735–740, 2018, [Online]. Available: http://www.nber.org/papers/w16019.
[26] T. Guillod, P. Papamanolis, and J. W. Kolar, "Artificial neural network (ann) based fast and accurate inductor modeling and design," *IEEE Open J. Power Electron.*, vol. 1, no. June, pp. 284–299, 2020, doi: 10.1109/OJPEL.2020.3012777.
[27] A. Dutta, S. Kumar, and M. Basu, "A Gated Recurrent Unit Approach to Bitcoin Price Prediction," *J. Risk Financ. Manag.*, vol. 13, no. 2, 2020, doi: 10.3390/jrfm13020023.
[28] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," pp. 1–20, 2018, [Online]. Available: http://arxiv.org/abs/1811.03378.
[29] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th Int. Symp. Qual. Serv. IWQoS 2018*, pp. 1–2, 2019, doi: 10.1109/IWQoS.2018.8624183.

Biodata Penulis

D. Diffran Nur Cahyo, M.Kom, Memperoleh gelar Magister Komputer di Universitas Amikom Yogyakarta pada tahun 2024. Menjadi Dosen Informatika di Universitas Cendekia Mitra Indonesia pada tahun 2024.

Dr. Andi Sunyoto, M.Kom, Memperoleh gelar Doktoral di Universitas Gadjah Mada pada tahun 2019. Menjadi Dosen Magister Informatika di Universitas Amikom Yogyakarta.