

BAB II

LANDASAN TEORI

2.1 Landasan Teori

2.1.2. Cuaca Ekstrem

Cuaca ekstrem merupakan peristiwa alam yang ditandai oleh kondisi seperti suhu udara, curah hujan, dan kecepatan angin, serta jarak pandang, di luar kisaran normal yang dapat menyebabkan kerugian, terutama terhadap keselamatan jiwa dan harta benda (BMKG, 2022). Variabel-variabel tersebut digunakan untuk menentukan pola cuaca yang berpotensi menimbulkan cuaca ekstrem yang berdampak pada lingkungan dan aktivitas manusia. Penjelasan lebih lanjut mengenai masing-masing variabel tersebut disajikan pada bagian berikut.

1. Suhu minimum dan suhu maksimum

Suhu udara merupakan ukuran yang mencerminkan kondisi termal dan radiasi atmosfer yang mengelilingi bumi (Sedrakyan et al., 2023). Paparan suhu udara ekstrem meningkatkan risiko terjadinya fenomena gelombang panas dan dingin, yang berdampak buruk pada kesehatan manusia (Danova, 2023). Suhu ekstrem menjadi indikator utama dalam menentukan potensi cuaca ekstrem seperti gelombang panas atau dingin, yang dapat mempengaruhi kehidupan manusia dan ekosistem. Oleh karena itu, kedua variabel ini dipilih untuk memahami pola perubahan suhu yang dapat mengarah pada kondisi ekstrem.

2. Curah Hujan

Curah hujan merujuk pada jumlah air hujan yang terakumulasi di suatu lokasi tertentu yang diukur dalam satuan milimeter (Fendi Prasetyo & Feri Efendi, 2024). Curah hujan yang ekstrem dalam bentuk hujan lebat dapat menyebabkan bencana alam. Curah hujan yang tinggi dapat memicu banjir dan tanah longsor, yang berdampak negatif pada manusia, dan aspek sosial ekonomi (Sato & Shuin, 2023). Maka dari itu, variabel ini dipilih karena intensitas hujan yang tinggi dapat menjadi indikator penting dalam mendeteksi potensi cuaca ekstrem dan bencana hidrometeorologi.

3. Kecepatan angin

Kecepatan angin adalah laju pergerakan udara yang diukur dalam satuan meter per detik (m/s) atau kilometer per jam (km/jam). Kecepatan angin yang tinggi sering kali terkait dengan fenomena cuaca ekstrem seperti badai tropis, tornado, dan siklon. Angin kencang dapat menyebabkan kerusakan bangunan, menghambat transportasi, dan berkontribusi terhadap eskalasi bencana lainnya seperti kebakaran hutan yang menyebar lebih cepat (Mazumder et al., 2023). Oleh karena itu, variabel ini dipilih untuk memahami hubungan antara pola angin dan kejadian cuaca ekstrem.

2.1.3. Peramalan

Peramalan merupakan metode analisis data historis untuk mengidentifikasi pola dan membuat prediksi terhadap kejadian di masa depan (Garg et al., 2022). Berbagai model peramalan telah dikembangkan untuk menangani beragam jenis data dan kebutuhan prediksi, termasuk model statistik seperti ARIMA, serta model

berbasis *deep learning* seperti LSTM, dan GRU. Pemilihan model yang tepat sangat bergantung pada sifat data yang diprediksi. Dalam peramalan, penting untuk memilih model yang sesuai untuk meningkatkan akurasi hasil prediksi, terutama ketika data yang digunakan memiliki pola tertentu yang harus dipertimbangkan.

Peramalan banyak digunakan dalam berbagai bidang, seperti ekonomi, bisnis, dan cuaca, untuk membantu pengambilan keputusan yang lebih baik (Kolambe & Arora, 2024). Dalam peramalan cuaca, model peramalan digunakan untuk memprediksi variabel meteorologi seperti suhu, curah hujan, dan kecepatan angin dengan tujuan untuk memberikan informasi yang akurat dan tepat waktu bagi masyarakat dan pihak berwenang. Mengingat sifat data cuaca yang kompleks, pemilihan metode yang tepat menjadi kunci untuk meningkatkan akurasi peramalan cuaca.

2.1.4. Data Pre-processing

Data Pre-processing adalah proses mengubah data ke dalam format yang lebih optimal agar dapat diproses secara lebih cepat dan efisien dalam *data mining*, *machine learning*, serta berbagai tugas *data science* lainnya (Varma et al., 2023). Proses ini bertujuan untuk memastikan bahwa data yang digunakan bersih, terstruktur, dan siap untuk diproses lebih lanjut. Adapun beberapa tahapan *data pre-processing* dalam penelitian ini adalah sebagai berikut

1. Date time index

Date time index merupakan proses penataan data *time series* dengan memastikan bahwa setiap titik data memiliki indeks waktu yang tepat dan

konsisten, yang sangat penting dalam analisis data berbasis waktu (Khan et al., 2024).

2. *Handling missing values*

Handling missing values merupakan serangkaian teknik untuk mengatasi ketidaklengkapan data dalam suatu dataset, khususnya pada data *time series*, melalui penghapusan entri, imputasi berbasis metode statistik, atau pendekatan model guna menjaga kualitas serta integritas data (El-Bakry et al., 2021). Dengan menerapkan teknik *handling missing values* yang sesuai, kualitas serta integritas data dapat tetap terjaga, sehingga hasil prediksi atau analisis tidak mengalami distorsi akibat hilangnya informasi penting.

3. Ekstraksi Fitur Waktu

Ekstraksi fitur waktu merupakan salah satu tahapan dalam *preprocessing* data *time series* yang bertujuan untuk mengungkap pola temporal tersembunyi yang mungkin tidak langsung terlihat dari nilai data mentah (Chernikov et al., 2022). Dalam konteks data cuaca, informasi waktu seperti tahun, bulan, hari, dan jam dapat berperan penting karena variabel-variabel meteorologis sering kali menunjukkan pola musiman, tren tahunan, dan fluktuasi periodik.

4. *Data Normalization*

Normalisasi data dalam *deep learning* merupakan tahap *preprocessing* yang bertujuan untuk menyelaraskan rentang nilai variabel *input* guna meningkatkan kinerja model (Tran et al., 2021). Dalam penelitian ini, metode *Min-Max Scaling* digunakan untuk mereskalakan nilai fitur ke dalam rentang $[0,1]$ sehingga distribusi

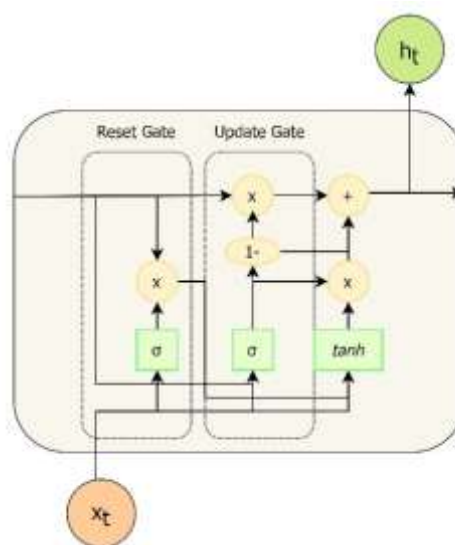
relative data tetap terjaga. Proses normalisasi dalam penelitian ini dilakukan menggunakan Persamaan 2.1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.1)$$

X adalah nilai asli fitur, X_{min} adalah nilai minimum dari fitur, dan X_{max} adalah nilai maksimum dari fitur.

2.1.5. *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit atau GRU adalah salah satu varian dari *Recurrent Neural Network* (RNN) yang dirancang khusus untuk menangani data *time series* (Mahan Zaky, 2022). GRU memiliki arsitektur yang ringan karena tidak menggunakan *cell state* seperti pada LSTM, melainkan langsung memperbarui *hidden state*. Hal ini membuat GRU lebih cepat dalam proses pelatihan dan lebih efisien dalam penggunaan memori, Arsitektur dari model GRU dapat dilihat pada Gambar 2.1



Gambar 2. 1 Model GRU

GRU memiliki dua *gate* utama, yaitu *update gate* dan *reset gate* (Meriani & Rahmatulloh, 2024). Adapun penjelasan mengenai masing-masing *gate* adalah sebagai berikut:

1. *Reset Gate*

Reset gate berfungsi untuk mengontrol informasi sebelumnya yang harus dilupakan (Zahri et al., 2023).. Persamaan pada *reset gate* dinyatakan pada Persamaan 2.2.

$$r_t = \sigma(w_r[h_{t-1}, x_t] + b_r) \quad (2.2)$$

r_t merupakan vektor *reset gate*, σ adalah fungsi aktivasi sigmoid, w_r dan b_r masing-masing adalah bobot serta bias pada *reset gate*, h_{t-1} menunjukkan status tersembunyi pada waktu sebelumnya, sedangkan x_t merepresentasikan *input* pada waktu t .

2. *Update gate*

Update gate berfungsi untuk menentukan seberapa banyak informasi sebelumnya yang harus dipertahankan (Zahri et al., 2023). Persamaan pada *update gate* dinyatakan pada Persamaan 2.3.

$$z_t = \sigma(w_z[h_{t-1}, x_t] + b_z) \quad (2.3)$$

z_t merupakan vektor *update gate*, σ adalah fungsi aktivasi sigmoid, w_z dan b_z masing-masing adalah bobot serta bias pada *update gate*, h_{t-1} menunjukkan status tersembunyi sebelumnya, sedangkan x_t merepresentasikan *input* pada waktu t .

3. *Candidate hidden state*

Candidate hidden state $\tilde{h}_t$ adalah lapisan aktivasi *tanh* yang menghitung *candidate hidden state* baru berdasarkan *input* saat ini dan *hidden state* sebelumnya, yang diatur oleh *reset gate*. Perhitungan *candidate hidden state* dilakukan dengan Persamaan 2.4.

$$\tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, x_t] + b_h) \quad (2.4)$$

$\tilde{h}_t$ merupakan *candidate hidden state*, $\tanh$ adalah fungsi aktivasi *tanh*, W_h dan b_h masing-masing adalah bobot serta bias pada *candidate hidden state*, r_t adalah vektor *reset gate*, h_{t-1} menunjukkan status tersembunyi sebelumnya, dan x_t merupakan *input* pada waktu t .

4. *New hidden state*

Terakhir, dilakukan perhitungan *new hidden state* (h_t) yang merupakan kombinasi dari *hidden state* sebelumnya dan *candidate hidden state*, yang diatur oleh *update gate*. Perhitungan *new hidden state* dilakukan dengan Persamaan 2.5.

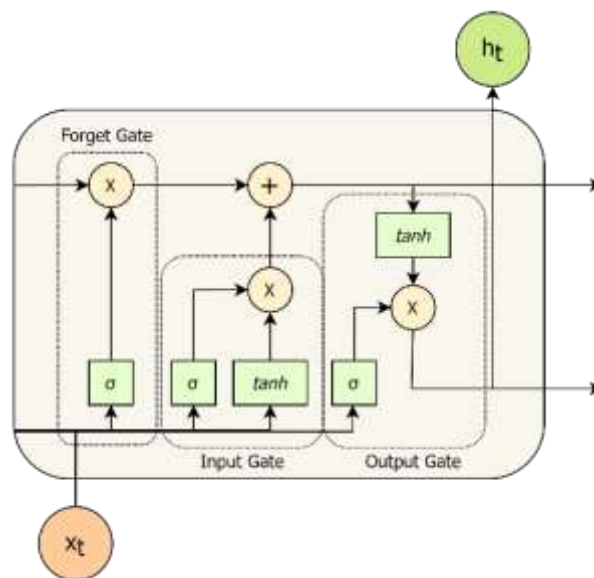
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.5)$$

h_t adalah *hidden state* baru pada waktu t , z_t merupakan vektor *update gate* yang menentukan proporsi antara *hidden state* sebelumnya h_{t-1} , dan *candidate hidden state* $\tilde{h}_t$.

2.1.6. *Long Short Term Memory (LSTM)*

Long Short Term Memory atau LSTM adalah varian *Recurrent Neural Network* (RNN) yang dirancang untuk mengolah data *time series* dan mengatasi masalah *vanishing* serta *exploding gradient* yang sering terjadi pada RNN (Kurnia

Tiara Putri & Danny Wowor, 2024). LSTM dikenal karena kemampuannya yang unggul dalam mempertahankan informasi dalam jangka waktu yang panjang sehingga banyak digunakan dalam berbagai aplikasi berbasis sekuensial, seperti prediksi cuaca, analisis finansial, *natural language processing*, dan deteksi anomali dalam data sensor (Zulfa et al., 2021). Meskipun lebih kompleks dibandingkan dengan RNN biasa, keunggulan LSTM dalam menangani ketergantungan jangka panjang membuatnya menjadi salah satu model yang paling populer dalam pemodelan *time series*. Arsitektur dari LSTM dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Model LSTM

LSTM memiliki struktur yang lebih kompleks dengan tiga gerbang utama yakni *forget gate*, *input gate*, dan *output gate*, yang berfungsi untuk mengatur aliran informasi masuk dan keluar *memory cell* (Jothidurai et al., 2023). Adapun penjelasan mengenai masing-masing *gate* adalah sebagai berikut:

1. *Forget gate*

Forget gate adalah gerbang yang memutuskan informasi mana yang akan disimpan atau dibuang (Nazhiroh et al., 2024). Perhitungan *forget gate* dilakukan dengan Persamaan 2.6 dan Persamaan 2.7.

$$ft = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.6)$$

ft adalah vektor *forget gate*, σ adalah fungsi aktivasi sigmoid, W_f dan b_f masing-masing merupakan bobot dan bias *forget gate*, h_{t-1} adalah *hidden state* sebelumnya, dan x_t adalah *input* saat ini.

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

C_t adalah *cell state*, ft adalah vektor *forget gate*, C_{t-1} adalah *cell state time step* sebelumnya, i_t adalah vektor *input gate*, dan $\tilde{C}_t$ adalah kandidat *cell state*.

2. *Input gate*

Input gate adalah gerbang yang menerima informasi dari *hidden state cell* sebelumnya dan *input* baru, lalu diproses dengan fungsi *sigmoid* dan *tanh* (Nazhiroh et al., 2024). Perhitungan *input gate* dilakukan dengan Persamaan 2.8 dan Persamaan 2.9.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.8)$$

i_t adalah vektor *input gate*, σ adalah fungsi aktivasi sigmoid, W_i dan b_i masing-masing merupakan bobot dan bias *input gate*, h_{t-1} adalah *hidden state* sebelumnya, dan x_t adalah *input* pada waktu t .

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (2.9)$$

$\tilde{C}_t$ adalah kandidat cell state, $\tanh$ adalah fungsi aktivasi tangens hiperbolik, W_C dan b_C masing-masing merupakan bobot dan bias untuk *cell state*, sementara h_{t-1} dan x_t tetap merujuk pada *hidden state* sebelumnya dan *input* saat ini.

3. *Output Gate*

Output gate adalah gerbang yang menentukan *hidden state* yang akan dikirim ke *cell* berikutnya (Nazhiroh et al., 2024). Perhitungan *output gate* dilakukan dengan Persamaan 2.10 dan 2.11.

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

o_t adalah *vektor output gate*, σ merupakan fungsi aktivasi sigmoid, W_o dan b_o masing-masing adalah bobot dan bias *output gate*, h_{t-1} adalah *hidden state* sebelumnya, dan x_t adalah *input* saat ini.

$$h_t = o_t * \tanh(C_t) \quad (2.11)$$

h_t adalah *hidden state* pada waktu t , o_t adalah *output gate*, dan C_t adalah *cell state* pada waktu t , sedangkan $\tanh$ adalah fungsi aktivasi tangens hiperbolik yang digunakan untuk mengatur skala nilai *cell state* sebelum menghasilkan *output hidden state*.

2.1.7. **Model Hybrid dalam Deep Learning**

Model *hybrid* dalam *deep learning* merupakan pendekatan yang menggabungkan dua atau lebih arsitektur jaringan saraf guna mengoptimalkan adaptivitas terhadap data dan kinerja model dalam menyelesaikan permasalahan yang kompleks (Kraft et al., 2020). Dalam konteks data *time series*, model *hybrid* sering digunakan untuk menggabungkan kemampuan beberapa unit RNN

(*Recurrent Neural Network*), seperti GRU (*Gated Recurrent Unit*) dan LSTM (*Long Short-Term Memory*), guna meningkatkan akurasi prediksi dan stabilitas pembelajaran.

Penelitian ini menggabungkan GRU dan LSTM dalam satu model *hybrid* yang bertujuan untuk memanfaatkan GRU yang memiliki akurasi tinggi dan efisiensinya dalam memproses data secara cepat, serta keunggulan LSTM dalam menangani dependensi jangka panjang. Dengan mengombinasikan kedua jenis sel ini, model dapat menangkap pola temporal baik jangka pendek maupun jangka panjang secara lebih efektif, terutama pada data cuaca yang bersifat musiman dan fluktuatif.

2.1.8. Evaluation

Evaluasi model merupakan tahap penting dalam penelitian untuk menilai seberapa baik hasil prediksi suatu model dengan data aktual (Plevris et al., 2022). Dalam penelitian ini, kinerja model diukur menggunakan tiga metrik evaluasi utama, yaitu *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan *Root Mean Squared Error* (RMSE).

1. Mean Absolute Error (MAE)

MAE adalah metrik evaluasi yang mengukur rata-rata kesalahan absolut antara nilai prediksi dan nilai aktual (Hodson, 2022). Metrik ini lebih mudah diinterpretasikan karena memiliki satuan yang sama dengan data aslinya. Nilai MAE dapat dihitung dengan Persamaan 2.12

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

2. *Mean Squared Error (MSE)*

MSE adalah metrik yang menghitung rata-rata kuadrat selisih antara nilai yang diprediksi dan nilai aktual (Wang et al., n.d.). Metrik ini sangat sensitif terhadap kesalahan besar, sehingga memberikan penalti lebih tinggi pada prediksi yang jauh dari nilai sebenarnya. MSE dirumuskan pada Persamaan 2.13.

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2 \quad (2.13)$$

3. *Root Mean Squared Error (RMSE)*

RMSE merupakan akar kuadrat dari MSE dan memberikan ukuran kesalahan dalam satuan yang sama dengan data asli (Hodson, 2022). RMSE dapat dihitung menggunakan Persamaan 2.14.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2} \quad (2.14)$$

2.2 Penelitian Terkait

2.2.1. *State of the art*

Penelitian ini memuat tinjauan pustaka yang disusun berdasarkan rumusan masalah dan tujuan yang telah ditetapkan. Tabel 2.1 menyajikan beberapa penelitian terdahulu yang relevan sebagai referensi utama.

Tabel 2. 1 *State of the Art*

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
1	(Furizal et al., 2024)	GRU LSTM	Suhu	Penelitian ini menganalisis kinerja model LSTM dan GRU dalam peramalan suhu udara. Fokusnya adalah membandingkan efektivitas kedua model menggunakan metrik evaluasi seperti RMSE. Hasil penelitian menunjukkan bahwa GRU memberikan kinerja yang lebih stabil dan akurat dengan rentang RMSE 2.00 dibandingkan LSTM yang memiliki rentang RMSE 2.28.
2	(Gaol et al., 2024)	LSTM MLP	Curah hujan	Penelitian ini menyelidiki prediksi meteorologi di Tangerang selatan menggunakan model LSTM dan MLP, dengan fokus pada prediksi curah hujan. Hasil penelitian ini menunjukkan prediksi curah hujan bulanan menggunakan MLP menghasilkan nilai RMSE sebesar 90,19, dan MAPE

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
				sebesar 40,55, sedangkan model LSTM menghasilkan nilai RMSE sebesar 88,12 dan MAPE sebesar 40,49.
3	(Yan et al., 2020)	GRU + <i>attention mechanism</i>	Kecepatan angin rata-rata Kecepatan angin puncak <i>Solar irradiance</i>	Hasil eksperimen menunjukkan bahwa model <i>hybrid deep learning</i> yang menggabungkan GRU dan <i>attention mechanism</i> memiliki kinerja prediksi yang lebih baik dibandingkan dengan model-model tradisional seperti LSTM dan GRU standar. Hasil evaluasi menunjukkan bahwa model <i>GRU Attention</i> mencapai nilai MAE 26.49, RMSE 36.23, dan MAPE 5.80 untuk musim semi, serta performa yang lebih baik dibandingkan dengan model tradisional seperti LSTM.
4	(J. Lee et al., 2020)	ANN	Curah hujan	Penelitian ini,berfokus pada prediksi curah hujan menggunakan model ANN selama bulan Mei dan Juni untuk cekungan Sungai Han, Korea Selatan. Hasil peramalan curah hujan menunjukkan RMSE sebesar 27,4, 33,6, dan 39,5 mm, dan koefisien korelasi rata-rata sebesar 0,809, 0,725, dan 0,641 untuk set pelatihan, validasi, dan pengujian.
5	(Alam et al., 2024)	LSTM CNN RNN	Suhu Kecepatan angin	Perbandingan model LSTM, CNN, dan RNN dalam peramalan cuaca untuk mengetahui model yang paling. Hasil penelitian menunjukan

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
				bahwa LSTM mengungguli CNN dan RNN dengan nilai AUC tertinggi sebesar 0.999.
6	(Chen et al., 2022)	LSTM	Curah hujan	Hasil penelitian menunjukkan bahwa model LSTM lebih efisien dibandingkan dengan model RF dalam memprediksi curah hujan. Hasil metrik evaluasi menunjukkan bahwa model LSTM menghasilkan peningkatan RMSE antara 12.2% hingga 14.9%
7	(Kieu Tran et al., 2020)	LSTM dan GA	Suhu maksimum	Hasil penelitian menunjukkan bahwa model <i>hybrid</i> yang menggabungkan LSTM dan GA memberikan performa terbaik untuk ramalan jangka panjang, dengan LSTM menunjukkan keunggulan dibandingkan RNN dan ANN, terutama untuk ramalan 15 hari ke depan. Hasil evaluasi menunjukkan bahwa model LSTM yang dioptimalkan dengan GA mencapai RMSE sebesar 2.719°C untuk ramalan suhu maksimum.
8	(Suresh et al., 2020)	CNN-LSTM	<i>Irradiation</i> Kecepatan angin Suhu	Model dalam penelitian ini mengembangkan algoritma peramalan output panel surya menggunakan CNN dengan pendekatan <i>sliding window</i> . Hasil evaluasi menunjukkan bahwa model CNN-LSTM memberikan performa terbaik dengan MAE sebesar 0.0148 kW dan RMSE sebesar 0.0182 kW, lebih baik dibandingkan model CNN biasa yang memiliki MAE 0.0164 kW dan RMSE 0.0203 kW.

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
9	(Bi et al., 2023)	3D <i>deep neural networks</i>	Suhu <i>Wind speed</i> <i>Geopotential height</i>	Hasil evaluasi model mencatat RMSE sebesar 296.7 untuk ramalan <i>geopotential height</i> (Z500) selama 5 haris, lebih rendah dibandingkan dengan RMSE IFS yang mencapai 333.7 dan <i>FourCastNet</i> yang mencapai 462.5. Selain itu, model ini lebih dari 10.000 kali lebih cepat dalam waktu inferensi dibandingkan IFS.
10	(Kumar et al., 2021)	ConvLSTM	Curah hujan	Penelitian menunjukkan bahwa model ConvLSTM berhasil menangkap pola curah hujan dengan baik, terutama di daerah seperti Western Ghats dan Monsoon trough, dengan koefisien korelasi mencapai 0.67 untuk lead day 1 dan 0.42 untuk lead day 2.
11	(S. Lee et al., 2020)	CNN	Suhu harian rata-rata Suhu harian minimum Suhu harian maksimum	Fokus pembahasan dalam penelitian ini adalah membandingkan kinerja model-model <i>neural network</i> dalam meramalkan suhu dengan menggunakan data <i>input</i> yang berbeda. Dalam penelitian ini, model terbaik yang digunakan untuk meramalkan suhu harian adalah CNN. Model ini menunjukkan performa terbaik dengan nilai evaluasi RMSE 2.98 ketika menggunakan data <i>input</i> jam, mengungguli model lainnya.

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
12	(Del Real et al., 2020)	CNN + ANN	Suhu	Model CNN +ANN menunjukkan kinerja yang lebih baik dibandingkan dengan RTE, serta mengungguli pendekatan lain seperti Arima dan model ANN tradisional. Hasil evaluasi menunjukkan bahwa pendekatan ini menghasilkan akurasi ramalan yang tinggi, dengan nilai MAE 808.317, MAPE 1.493%, MBE 21.744, dan MBPE 0.023%.
13	(Narejo et al., 2021)	<i>Temporal Deep Belief Network</i> (DBN)	Curah hujan	Penelitian ini menunjukkan bahwa model Temporal DBN mengungguli model CNN konvensional dalam hal prediksi curah hujan. Model DBN yang dimodifikasi dengan struktur lapisan tersembunyi (300-200-100-10) menunjukkan performa terbaik dengan nilai MSE sebesar 4.59×10^{-5} , RMSE sebesar 0.00680.0068, dan nilai koefisien korelasi RR sebesar 0.940.94 pada sampel pengujian.
14	(Lin et al., 2021)	TCN	Kecepatan angin	Hasil eksperimen menunjukkan bahwa model TCN mencapai MAPE sebesar 5.13% untuk prediksi daya angin selama 72 jam, yang memenuhi persyaratan proyek energi terbarukan mereka. Penelitian ini menekankan bahwa TCN lebih unggul dibandingkan dengan model lain dalam hal volume data <i>input</i> , stabilitas pengurangan kesalahan, dan akurasi prediksi.
15	(Dolatabadi et al., 2020)	BLSTM +DWPT	Kecepatan angin	Fokus utama dari penelitian ini adalah untuk menunjukkan efektivitas model hibrida yang menggabungkan DWPT dan BLSTM dalam

No	Penulis	Algoritma	Fitur	Fokus Pembahasan
				memprediksi kecepatan angin dibandingkan dengan model-model lain yang lebih sederhana. Penelitian ini menyoroti bahwa kombinasi DWPT dan BLSTM dapat lebih baik dalam menangkap pola kompleks dan non-linear dalam data kecepatan angin. Hasil evaluasi menunjukkan bahwa model ini memiliki kinerja yang superior dibandingkan dengan arsitektur dangkal dan model pembelajaran mendalam lainnya.
16	(Mishra et al., 2020)	Deep Convolutional Network (DCN)	Kecepatan Angin Suhu	Hasil penelitian menunjukkan bahwa model <i>Attention</i> dan DCN memberikan performa terbaik ketika menggunakan transformasi data melalui Wavelet atau FFT, sementara beberapa model lainnya lebih baik tanpa manipulasi data.

2.2.2. Matriks Penelitian

Matriks penelitian digunakan untuk membandingkan penelitian yang dilakukan dengan studi terdahulu. Analisis ini bertujuan mengidentifikasi kesenjangan penelitian yang mendukung pengembangan model dalam studi ini. Tabel 2.2 menyajikan perbandingan komprehensif terhadap penelitian sebelumnya.

Tabel 2. 2 Matriks Penelitian

No	Penulis	Algoritma	Fitur			
			Suhu Minimum	Suhu Maksimum	Curah Hujan	Kecepatan Angin
1	(Furizal et al., 2024)	GRU dan LSTM	√	√	-	-
2	(Gaol et al., 2024)	LSTM dan MLP	-	-	√	-
3	(Yan et al., 2020)	GRU+ attention mechanism	-	-	-	√
4	(J. Lee et al., 2020)	ANN	-	-	√	-
4	(Alam et al., 2024)	LSTM, CNN, dan RNN		√		√
6	(Chen et al., 2022a)	LSTM	-	-	√	-
7	(Kieu Tran et al., 2020)	LSTM dan GA	-	√	-	-
8	(Suresh et al., 2020)	CNN-LSTM	√	√	-	√
9	(Bi et al., 2023)	3D deep neural networks	√	√	-	√
10	(Kumar et al., 2021)	ConvLSTM	-	-	√	-
11	(S. Lee et al., 2020)	CNN	√	√	-	-
12	(Del Real et al., 2020)	CNN + ANN	√	√	-	-

No	Penulis	Algoritma	Fitur			
			Suhu Minimum	Suhu Maksimum	Curah Hujan	Kecepatan Angin
13	(Narejo et al., 2021)	<i>Temporal Deep Belief Network</i> (DBN)	-	-	√	-
14	(Lin et al., 2021)	TCN	-	-	-	√
15	(Dolatabadi et al., 2020)	BLSTM+DWPT	-	-	-	√
16	(Mishra et al., 2020)	Deep Convolutional Network (DCN)	√	√	-	√
17	Penelitian ini	GRU+LSTM	√	√	√	√

2.1.9. Kebaruan Penelitian

Tabel 2.2 menyajikan matriks penelitian yang menunjukkan perbedaan antara penelitian yang dilakukan dengan studi sebelumnya. Penelitian terdahulu telah mengeksplorasi berbagai model, seperti LSTM, GRU, CNN dan juga berbagai pendekatan lainnya. Penelitian ini menawarkan dua kebaruan signifikan. Pertama, penelitian ini mengusulkan pengembangan pendekatan *hybrid* GRU-LSTM untuk menggabungkan keunggulan model kedua arsitektur model. Kedua, penelitian ini melakukan pengembangan model *multioutput* yang secara spesifik menggunakan variabel-variabel yang memengaruhi terjadinya potensi cuaca ekstrem. Dengan demikian, hasil dari penelitian ini diharapkan dapat memperluas pemahaman dan penerapan arsitektur *deep learning* dalam konteks peramalan cuaca dan dapat memberikan kontribusi dalam efektivitas peramalan cuaca.