# Peramalan Kelembapan Relatif di Kabupaten Bogor Menggunakan Model CNN-LSTM

Thariq Abdullah
Bandar lampung, Indonesia
thariqabdullah@student.telkomuniversity.ac.id

Achmad Lukman
Bandung, Indonesia
alukman@telkomuniversity.ac.id

***Abstrak* — Kelembapan udara merupakan salah satu parameter yang berpengaruh terhadap lingkungan dan aktivitas manusia. Prediksi kelembapan udara secara akurat dapat membantu dalam berbagai keperluan, termasuk pengambilan keputusan berbasis cuaca. Namun, model tunggal memiliki keterbatasan dalam menangkap pola non linier dan ketergantungan jangka panjang pada data deret waktu, sehingga sulit memprediksi data dengan baik, terutama data yang bersifat kompleks seperti cuaca. Oleh karena itu, diperlukan model dengan pendekatan hibrida. Pemodelan hibrida merupakan gabungan dari dua atau lebih metode pembelajaran. Penelitian ini mengusulkan pendekatan hibrida dengan menggabungkan lapisan *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) untuk melakukan prediksi kelembapan udara yang lebih akurat dan efektif dibandingkan dengan model tunggal. CNN digunakan untuk mengekstraksi representasi temporal dari data cuaca historis dalam bentuk urutan waktu, sedangkan LSTM berperan dalam mengingat dependensi jangka panjang. Prediksi dilakukan dengan pengumpulan data cuaca, prapemrosesan data, transformasi fitur (termasuk transformasi fitur siklik), perancangan arsitektur CNN-LSTM, pelatihan model, serta evaluasi menggunakan metrik evaluasi seperti *mean absolute error* (MAE), *mean square error* (MSE), *root mean square error* (RMSE), dan *coefficient* of *determination* ($R^2$). Hasil yang diperoleh menunjukkan bahwa model hibrida CNN-LSTM mampu meningkatkan akurasi prediksi kelembapan udara dibandingkan model Tunggal.**

***Kata kunci*— cnn, deret waktu, lstm, prediksi**

## I. PENDAHULUAN

Kelembapan udara merupakan kandungan uap air di dalam udara [1][2]. Kelembapan udara menjadi salah satu parameter penting dalam sistem *monitoring* cuaca yang berpengaruh langsung terhadap berbagai aspek kehidupan manusia, seperti kesehatan hingga kenyamanan. Penelitian telah menunjukkan bahwa kondisi lingkungan yang tidak nyaman, termasuk kelembapan yang tinggi, dapat memengaruhi konsentrasi belajar, kualitas tidur, dan kesehatan fisik serta mental [2]. Dalam dunia kesehatan, kelembapan yang tinggi juga dapat menjadi kondisi yang ideal bagi nyamuk hingga dapat menyebabkan meningkatnya risiko DBD [3]. Informasi tentang kelembapan yang akurat sangat dibutuhkan untuk mendukung pengambilan keputusan yang efektif.

Dalam beberapa tahun terakhir, perkembangan teknologi dan ketersediaan data cuaca historis secara digital memungkinkan diterapkannya pendekatan berbasis data untuk meramalkan kondisi cuaca, termasuk kelembapan udara. Namun, peramalan kelembapan bukanlah hal yang sederhana, karena melibatkan dinamika atmosfer yang bersifat kompleks dan nonlinier. Oleh sebab itu, dibutuhkan metode yang m ampu menangkap hubungan temporal serta pola tertentu dari data untuk menghasilkan peramalan yang akurat. Model hibrida dengan menggabungkan dua lapisan *deep learning* mampu menjadi solusi untuk data yang bersifat kompleks [4].

Penelitian sebelumnya yang berfokus pada peramalan harga saham menunjukkan menunjukkan model CNN-BiLSTM mampu menangani fluktuasi dan kompleksitas data saham [4]. Hasil prediksi dari 3 tren saham Bank, bahwa nilai MAPE yang diperoleh untuk setiap mode yang dihasilkan kurang dari 10% yang berarti bahwa model yang dihasilkan memiliki akurasi yang sangat baik.

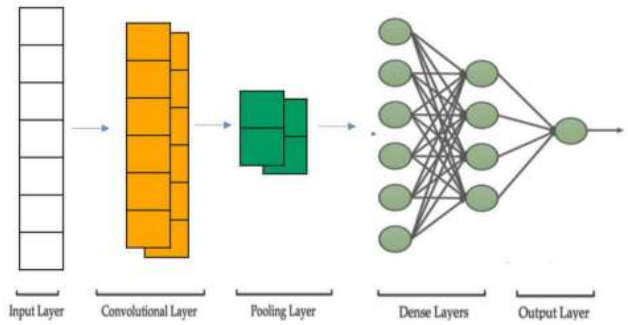
Penelitian terdahulu [5] dapat memprediksi cuaca dengan menggunakan model hibrida CNN-LSTM. Pada penelitian tersebut, model CNN-LSTM mampu memprediksi beberapa variabel ke beberapa langkah ke depan secara bersamaan dengan masukan variabel masa lalu. Namun, dalam beberapa kasus hanya satu variabel tertentu yang menjadi fokus utama, sementara variabel lainnya berperan sebagai faktor pendukung yang mempengaruhi prediksi.

Berbeda dengan penelitian sebelumnya, penelitian ini mengusulkan model hibrida CNN-LSTM yang memprediksi satu varibel dengan 2 variabel lainnya sebagai pendukung. Penggunaan model hibrida CNN-LSTM dapat memberikan kontribusi dalam pengembangan sistem peramalan cuaca berbasis *deep learning* yang cerdas dan adaptif terhadap data historis.

## II. KAJIAN TEORI

### A. CNN 1D

*Convolutional Neural Network (CNN)* telah diadaptasi untuk berbagai jenis data, termasuk data satu dimensi seperti data deret waktu. Penerapan kernel atau filter konvolusi pada CNN 1D secara geser terhadap urutan data, memungkinkan jaringan untuk mengekstrak fitur lokal yang relevan. Arsitektur model CNN 1D dapat dilihat pada Gambar 1.

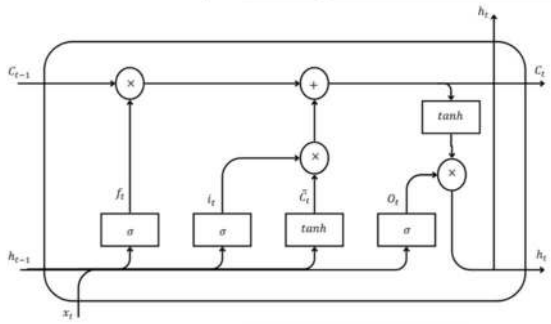



GAMBAR 1

Kernel pada lapisan konvolusi1D akan bergeser satu per satu untuk mempelajari pola pada data [6]. Setiap filter akan mengekstraksi fitur unik yang berbeda-beda. Lapisan *pooling* akan mengambil fitur dengan nilai terbesar [6]. Lapisan Hidden melakukan transformasi *nonlinier* menggunakan agar model dapat mengenali hubungan kompleks antar fitur. Keluaran akhir berupa hasil peramalan beberapa langkah ke depan dihasilkan oleh lapisan keluaran.

B. Long Short-Term Memory

LSTM adalah jenis jaringan saraf rekursif (RNN) yang dirancang untuk menangkap dependensi jangka panjang [7]. LSTM unggul dalam memprediksi pola temporal yang kompleks, seperti ketergantungan waktu [7]. LSTM menggunakan *gates* (gerbang) untuk mengontrol aliran informasi, sehingga dapat "mengingat" atau "melupakan" informasi sesuai kebutuhan, sehingga memungkinkan LSTM untuk menyimpan informasi lebih lama dibandingkan RNN biasa. Arsitektur LSTM dapat dilihat pada Gambar 2.




GAMBAR 2

Setiap unit LSTM terdiri dari *cell state, hidden state,* dan *gates*, yang terdiri atas *forget gates, input gate,* dan *output gate* [9] seperti pada Gambar 2. *Cell state* ($c_t$) mempertahankan informasi yang dibawa dari langkah waktu sebelumnya dan bertindak sebagai "memori jangka panjang" dari LSTM. $ht$ atau *Hidden state* mewakili "memori jangka pendek" dari LSTM dan digunakan untuk menghasilkan *output* pada setiap langkah waktu. *Gates* pada LSTM digunakan untuk mengontrol informasi [8], dengan *Forget Gate* ($f_t$) untuk memutuskan informasi mana yang harus dilupakan, *input gate* ($i_t$) memutuskan informasi baru apa yang harus disimpan, dan *output gate* ($Ot$) yang memutuskan informasi mana yang akan digunakan untuk menghasilkan *output*. *Forget gate* akan menerima informasi yang berupa *hidden state* yang berasal dari *cell* sebelumnya dan informasi baru yang berasal dari *input* saat ini, kemudian informasi tersebut akan di gabung dan diproses dengan menggunakan fungsi *sigmoid* [9] yang akan menghasilkan hasil berupa 0 hingga 1 dengan rumus (1).

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

Hasil yang mendekati 0 akan dibuang, dan hasil yang mendekati akan disimpan. Selanjutnya *input gate* akan menerima informasi berupa *hidden state* yang berasal dari *cell* sebelumnya dan informasi baru yang berasal dari *input* saat ini, kemudian informasi tersebut akan di gabung dan diproses (2) dan membuat kandidat baru (3).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$c_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right) \quad (3)$$

Hasil dari fungsi *sigmoid* akan mengubah nilai menjadi 0 hingga 1 untuk menentukan informasi mana yang akan di*update*. Semakin dekat dengan 1, informasi tersebut semakin penting. *Output gate* akan menerima informasi yang berupa *hidden state* yang berasal dari *cell* sebelumnya dan informasi baru yang berasal dari *input* saat ini kemudian informasi akan digabung dan diproses dengan fungsi *sigmoid* (1).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (4)$$

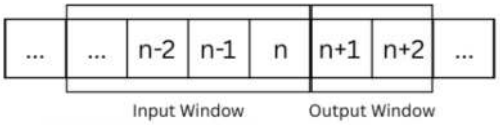
*Cell state* yang baru kemudian akan di proses melalui fungsi *tanh*. Hasil dari fungsi *tanh* akan dikalikan dengan hasil dari fungsi *sigmoid* untuk memeroleh informasi yang akan disimpan pada *hidden state* yang baru (5).

$$h_t = \mathrm{ot} \cdot \tanh(c_t) \quad (5)$$

*Hidden state* dan *Cell state* yang baru kemudian akan diteruskan ke *cell* selanjutnya.

C. *Sliding Window*

Metode *sliding window* merupakan pendekatan yang banyak digunakan dalam pemodelan deret waktu dengan membagi data deret waktu menjadi *window*, yang dapat digunakan sebagai *input* dan *output. Sliding window* terdiri dari *input window* dan *output window*. Konsep *sliding window* ditunjukkan pada Gambar 3.




GAMBAR 3

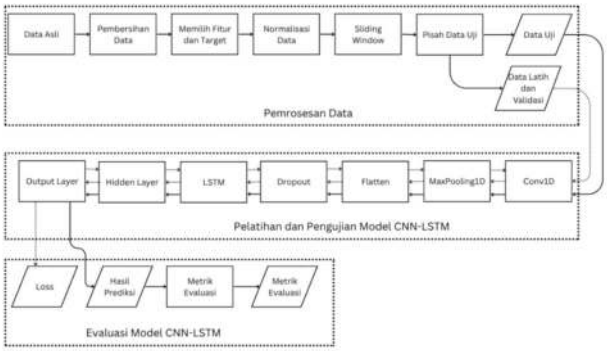
Prinsip dasar dari metode ini adalah dengan mengambil *window* berukuran tetap yang bergeser satu per satu sepanjang rangkaian data untuk membentuk *window* yang berisi *input* dan *output* secara berurutan [10].

*Sliding window* terbagi menjadi dua jenis utama, yaitu *single-output* dan *multi-output*. *Sliding window* pada pendekatan *single-output* memiliki satu *input window* yang digunakan untuk memprediksi satu nilai *output* berikutnya, sedangkan dalam pendekatan *multi-output*, satu *window input* dapat digunakan untuk memprediksi beberapa nilai ke depan dalam satu kali proses prediksi.

## III. METODE

Penelitian ini menggunakan model CNN-LSTM, yaitu model yang merupakan gabungan dari *layer* pada model *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) untuk melakukan peramalan kelembapan udara berdasarkan data catatan cuaca rata-rata per hari. Pemilihan arsitektur CNN-LSTM bertujuan untuk menangkap baik pola lokal (jangka pendek) maupun pola temporal jangka panjang dalam data sekuensial cuaca. Data akan dibersihkan dan diubah menjadi sekuen sebagai *input* model. Model akan dievaluasi dengan grafik *loss* pelatihan,

grafik peramalan, dan metrik evaluasi. Diagram blok ditunjukkan pada Gambar 4.




GAMBAR 4

A. Pengumpulan Data

Data catatan cuaca Kabupaten diperoleh melalui situs web resmi BMKG. Dataset yang digunakan merupakan data deret waktu harian yang dikumpulkan. Data yang digunakan adalah fitur suhu udara dan penyinaran matahari, serta target kelembapan relatif yang masing-masing memiliki 2142 baris. Suhu udara merupakan ukuran panas atau dinginnya udara di suatu tempat, sedangkan durasi penyinaran matahari merupakan durasi matahari bersinar pada suatu tempat.

B. Pra-Pemrosesan Data

Berikut langkah-langkah dalam pra-pemrosesan data:

1. Mengatasi nilai kosong

Mengganti nilai kosong pada data menggunakan teknik interpolasi waktu, yaitu dengan cara memperkirakan nilai data pada waktu tersebut berdasarkan data yang diketahui sebelum dan sesudah waktu tersebut.

2. Mengubah format dan tipe data

Mengubah format kolom tanggal menjadi datetime serta tipe data pada kolom numerik menjadi float agar data konsisten dan seragam, sehingga mudah diproses secara deret waktu.

3. Memilih data fitur dan target

Fitur dan target dipilih berdasarkan korelasi tertinggi dengan target. Data fitur dan target terdapat pada Tabel 1.

TABEL 1

| Data | Keterangan |
|---|---|
| $X_1$ | Suhu Udara |
| $X_2$ | Penyinaran Matahari |
| $Y_1$ | Kelembapan Relatif |

4. Normalisasi Data

Normalisasi fitur dan target dilakukan menggunakan metode Min-Max.

5. Pembentukan Sekuen

Data diubah menjadi bentuk sekuensial

6. Pembagian Data

Data dibagi menjadi 70% data untuk pelatihan, 15% data untuk vallidasi, dan 15% data untuk pengujian model.

C. Perancangan Model

Data diramal dengan menggunakan model CNN-LSTM, yaitu model yang berisi gabungan dari lapisan CNN1D dan LSTM. Model CNN-LSTM terdiri dari lapisan *TimeDistributed* konvolusi1D, *TimeDistributed* max*pooling*1D, *TimeDistributed flatten*, LSTM, *dropout*, *hidden*, dan *output*. Arsitektur model terdapat pada Gambar 5.




GAMBAR 5

Lapisan konvolusi 1D menangani ekstraksi fitur dari urutan data. Kernel akan mempelajari pola lokal dari data, lalu masing-masing filter akan mengekstraksi fitur unik. Lapisan pooling bertujuan untuk mengurangi dimensi fitur *output* lapisan konvolusi1D. Lapisan ini berisi *pool size* atau ukuran jendela pengambilan maksimum yang akan menyimpan nilai terbesar dari beberapa nilai berturut-turut. Layer ini juga dapat mempercepat pelatihan, dan mempertahankan fitur paling kuat. Lapisan *flatten* digunakan untuk mengubah dimensi data sebelum diolah oleh LSTM. Lapisan ini mengurangi dimensi dengan menggabungkan *timestep* dan fitur dari lapisan sebelumnya. Lapisan LSTM berfungsi untuk menangani hubungan temporal dari hasil ekstraksi fitur. Lapisan ini memiliki unit untuk menangkap pola jangka panjang pada data. Lapisan *dropout* digunakan untuk mengurangi *overfitting* dengan cara menonaktifkan sejumlah neuron secara acak pada probabilitas tertentu setiap *batch*. Lapisan *hidden* merupakan lapisan penghubung antara hasil *output* LSTM dengan

lapisan *output*. Lapisan ini melakukan transformasi *nonlinier* menggunakan agar model dapat mengenali hubungan kompleks antar fitur. Lapisan ini memiliki *neuron* untuk menghasilkan keluaran akhir berupa hasil peramalan beberapa langkah ke depan.

D. Evaluasi Model

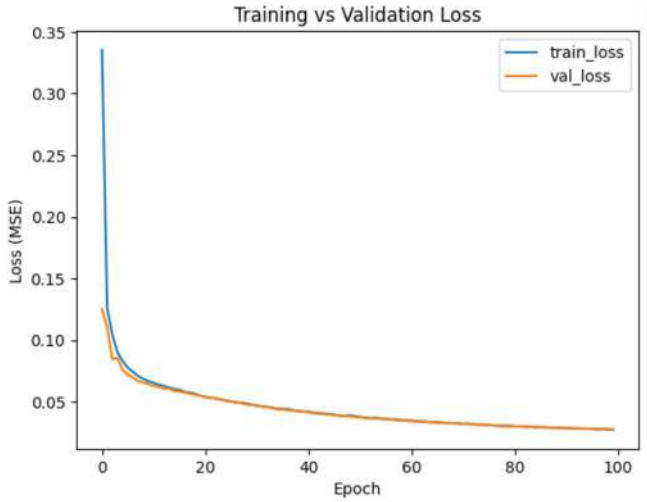
Model diuji menggunakan data uji, dievaluasi dengan metrik evaluasi, dan dianalisis dengan visualisasi hasil untuk mengetahui performa model dalam meramalkan kelembapan relatif meliputi:

1. Grafik Loss
2. Grafik Nilai Aktual dan Prediksi
3. Metrik Evaluasi

## IV. HASIL DAN PEMBAHASAN

A. Hasil

Pengujian dilakukan terhadap model CNN–LSTM yang telah dibangun yang bertujuan untuk mengetahui performa utama model dalam melakukan peramalan kelembapan relatif. Grafik yang menampilkan training loss dari pelatihan model CNN-LSTM ditunjukkan oleh Gambar 6.

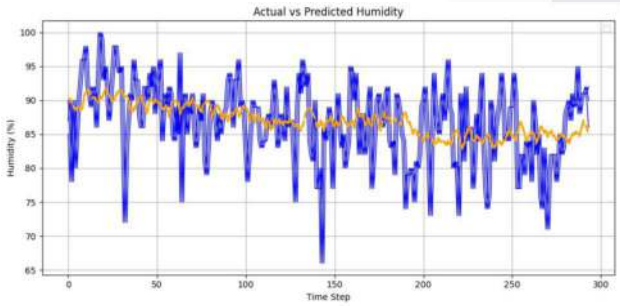



GAMBAR 6

Grafik pada Gambar 6 menunjukkan *training loss* dan validasi *loss* model CNN-LSTM selama 100 kali pelatihan.

Grafik perbandingan nilai aktual dan nilai prediksi model CNN-LSTM pada data uji ditunjukkan oleh Gambar 7.

B.




GAMBAR 7

Gambar 7 menunjukkan grafik nilai prediksi model CNN-LSTM yang direpresentasikan oleh garis warna oranye dan data aktual yang direpresentasikan oleh garis warna biru.

Hasil evaluasi model CNN-LSTM yang diuji dengan data uji ditunjukkan oleh Tabel 2.

TABEL 2

| *Timestep* | MAE | MSE | RMSE | $R^2$ |
|---|---|---|---|---|
| 1 | 4.3166 | 28.9413 | 5.3797 | 0.1052 |
| 2 | 4.3736 | 30.2218 | 5.4974 | 0.0667 |
| 3 | 4.3887 | 30.5749 | 5.5295 | 0.0548 |
| Rata-rata | 4.3596 | 29.9126 | 5.4689 | 0.0756 |

Metrik evaluasi dari berbagai model-model deep learning lainnya ditunjukkan oleh Tabel 3.

TABEL 3

| Model | Metrik Evaluasi | | | |
|---|---|---|---|---|
| | MAE | MSE | RMSE | $R^2$ |
| CNN-LSTM | 4.3596 | 29.9126 | 5.4689 | 0.0756 |
| CNN-BiLSTM | 4.3815 | 30.4224 | 5.5153 | 0.0598 |
| CNN-GRU | 4.4232 | 30.3398 | 5.5073 | 0.0624 |
| CNN | 4.5989 | 32.5019 | 5.7005 | -0.0045 |
| LSTM | 4.4267 | 30.3960 | 5.5131 | 0.0606 |
| BiLSTM | 4.3748 | 29.9074 | 5.4687 | 0.0757 |
| GRU | 4.3168 | 29.3300 | 5.4150 | 0.0936 |

B. Analisis

Pelatihan model CNN-LSTM menghasilkan *loss* dari data latih dan validasi yang menunjukkan generalisasi dari model. Model CNN-LSTM dengan konfigurasi yang berbeda tidak memiliki perbedaan generalisasi yang signifikan. Semua model CNN-LSTM memiliki generalisasi yang cukup baik.

Model CNN-LSTM yang memiliki semua komponen memiliki gene ralisasi yang baik. Grafik menunjukkan penurunan *loss* yang stabil dan seimbang antara *training* dan *validation*.

*Validation loss* dan *training loss* model CNN-LSTM tanpa subsekuen mengalami penurunan dengan baik. Namun, nilai *validation loss* sedikit lebih tinggi dibandingkan *training loss* yang dapat menandakan bahwa model mengalami *overfitting* walaupun hanya sedikit. Hal ini menunjukkan bahwa tanpa pembagian subsekuen, model dapat kehilangan konteks pola lokal pada data deret waktu.

Grafik *loss* model CNN-LSTM yang tidak menggunakan *learning schedule* pada awal pelatihan memiliki hasil *validation loss* yang lebih tinggi dari model lain, tetapi pada pelatihan berikut grafik *validation loss* mengalami penurunan yang stabil dan cenderung sama seperti model CNN- LSTM yang memiliki *learning schedule*. Hal ini menunjukkan bahwa tanpa *learning schedule*, model CNN-LSTM masih memiliki generalisasi yang baik.

Sementara model CNN-LSTM yang tidak menggunakan regularisasi mempunyai generalisasi yang sangat baik. *Validation loss* dan *training loss* mengalami penurunan yang sangat cepat. Model belajar dengan sangat cepat dan mampu mempelajari data dengan sangat baik. Atribut regularisasi tidak berpengaruh baik terhadap regularisasi

model CNN-LSTM yang memiliki subsekuen dan *learning schedule*.

Namun, grafik nilai peramalan yang dihasilkan oleh model CNN-LSTM masih berbeda secara signifikan dibandingkan grafik nilai aktual. Penulis berpendapat, hal ini disebabkan oleh data yang didapatkan berupa data rata-rata harian yang memiliki ketergantungan antar-waktu sangat rendah, sehingga pengaruh data terhadap data sebelumnya cenderung rendah. Oleh sebab itu, model menghasilkan pola lokal yang cenderung linear untuk mendapatkan hasil maksimal dan memiliki bentuk grafik yang sangat jauh dibandingkan grafik nilai aktual data. Sebagai perbandingan, model CNN–LSTM tanpa subsekuen menghasilkan grafik peramalan yang lebih bergelombang. Hal ini terjadi karena model lebih fokus ke pola jangka panjang atau musiman dan kurang memperhatikan pola lokal, sehingga akurasi model mengalami penurunan.

Model-model CNN-LSTM mampu mendapatkan hasil MAE dan RMSE cukup baik. Namun, nilai $R^2$ dari model-model tersebut masih sangat kecil, menandakan pola data sulit dipelajari. Model CNN-LSTM (*baseline*) memiliki performa terbaik, diikuti model CNN-LSTM tanpa *learning schedule*, model CNN-LSTM tanpa regularisasi, dan model CNN-LSTM tanpa subsekuen menjadi model dengan performa yang paling buruk.

Model CNN-LSTM memiliki rata-rata nilai MAE sebesar 4.3596, MSE sebesar 29.9126, RMSE sebesar 5.4689, dan $R^2$ sebesar 0.0756. Hasil ini menunjukkan bahwa model dapat meramalkan data dengan kesalahan peramalan rata-rata absolut melesat 4.3596 dari nilai aktual dengan rentang target 42 satuan.

Model CNN-LSTM lain mengalami penurunan performa dalam meramalkan data. Model CNN-LSTM yang tidak menggunakan subsekuen pada *sliding window* mengalami penurunan performa terbesar, dengan kenaikan MAE sebesar 0.1173 dan penurunan $R^2$ sebesar 0.0233. Hal ini menunjukkan bahwa penggunaan subsekuen pada *sliding window* dapat membantu model memahami pola lokal dan memperbaiki akurasi peramalan

Model CNN-LSTM yang tidak menggunakan *learning rate schedule* dan regularisasi *dropout* juga mengalami penurunan performa. Model CNN-LSTM yang tidak menggunakan *learning rate schedule* mengalami kenaikan nilai MAE sebesar 0.0476 dan penurunan $R^2$ sebesar 0.0092. Sedangkan model CNN-LSTM yang tidak menggunakan regularisasi *dropout* dan l2 mengalami penurunan performa dengan kenaikan nilai MAE sebesar 0.0483 dan penurunan $R^2$ sebesar 0.0042. Hal ini menunjukkan bahwa *learning rate schedule* dan komponen regularisasi turut memengaruhi performa model dalam melakukan peramalan, meskipun efeknya tidak begitu signifikan.

Semua komponen mempunyai pengaruh terhadap performa model, terutama penggunaan subsekuen yang terbukti meningkatkan akurasi dan generalisasi model dengan cukup signifikan dibandingkan komponen lain. Secara keseluruhan, model CNN-LSTM dengan regularisasi, subsekuen, dan *learning schedule* memiliki performa yang seimbang antara akurasi dan generalisasi.

Hasil pengujian menunjukkan bahwa model CNN–LSTM mampu menghasilkan nilai kesalahan dalam peramalan dengan tingkat kesalahan yang relatif rendah dibandingkan beberapa model *deep learning* lainnya. Model CNN–LSTM menunjukkan performa yang paling baik dari CNN-BiLSTM dan CNN-GRU berdasarkan nilai MAE dan RMSE yang lebih rendah. Hal ini mengindikasikan bahwa kombinasi CNN dengan LSTM standar sudah cukup efektif dalam menangkap pola temporal data kelembapan relatif dibandingkan dengan variasi BILSTM maupun GRU.

## V. KESIMPULAN

**S**impulan harus diuraikan dalam bentuk paragraf yang berisi poin utama pembahasan hasil penelitian, berupa uraian dan tidak boleh menggunakan pointer.

Implementasi model yang dirancang dari gabungan lapisan CNN dan lapisan LSTM mampu meramal data kelembapan udara dengan baik. Model mampu menjaga keseimbangan antara kesalahan yang rendah dan kestabilan model. Setiap komponen berpengaruh terhadap performa model dalam meramal data.

Dari hasil dan analisis, model CNN-LSTM yang dikembangkan mampu melakukan peramalan kelembapan udara beberapa langkah ke depan dengan performa yang paling baik dibandingkan model CNN-RNN lainnya. Model CNN-LSTM mampu mendapatkan performa yang seimbang antara akurasi dan regularisasi yang baik dibandingkan model lainnya, menunjukkan bahwa model mampu menangani data deret waktu, sekaligus mengenali pola lokal untuk mendapatkan hasil maksimal dengan baik.

Penelitian ini telah berhasil membangun model prediktif menggunakan pendekatan hibrida CNN-LSTM dengan implementasi lapisan CNN dan LSTM yang mampu melakukan peramalan kelembapan udara untuk 3 hari ke depan dengan baik. Evaluasi dilakukan secara menyeluruh terhadap data pelatihan maupun pengujian dengan menggunakan grafik *loss,* grafik nilai aktual dan prediksi, metrik evaluasi, serta dibandingkan dengan beberapa konfigurasi dan parameter seperti jumlah unit LSTM, filter CNN, *dropout*, regulasi L2, penyesuaian *learning rate*, dan model *deep learning* lainnya. Dari hasil dan analisis, kombinasi arsitektur *convolutional* dan *recurrent network* efektif dalam menangani pola sekuensial yang bersifat musiman dan fluktuatif dengan memiliki performa yang cukup baik dibandingkan model *deep learning* lainnya

## REFERENSI

[1] M. Muhaimin, “Urgensi Kenyamanan Termal dalam Perspektif Pembelajaran,” Geodika: Jurnal Kajian Ilmu dan Pendidikan Geografi, vol. 7, no. 1,, pp. 23-32, 2023, doi: 10.29408/geodika.v7i1.6451

[2] Suwarna, I. P., Pramudita, F., Yulianto, R., Tamala, L. D., Syamila, H., Nurfadillah, S., Ramadhan, P. H. 2025. Analisis Kelembapan udara dan tingkat kenyamanan lingkungan indekos di area sekitar kampus UIN Jakarta. JKPL. 6(1).

[3] Alizkan, Ubed, “Analisis Korelasi Kelembapan Udara terhadap Epidemi Demam Berdarah yang Terjadi di Kabupaten dan Kota Serang,” Gravity, vol. 3, no. 1, 2017, doi:10.30870/gravity.v3i1.2409.

[4] M. Mushiha, “Implementasi CNN-BiLSTM untuk prediksi harga saham ban syariah di Indonesia,” Jambura Joural of Mathematic, vol. 6, no. 2, Agustus 2024, doi: 10.37905/jjom.v6i2.26509.

[5] M. S. Divyashree, V. S. Sparshakala, S. Shridevi, and B. V. Bhavya, “Deep learning: Hybrid CNN–LSTM for

multivariate weather prediction," *International Advanced Research Journal in Science, Engineering and Technology*, vol. 12, no. 10, Oct. 2025, doi: 10.17148/IARJSET.2025.121022.

[6] J. Brownlee, *Deep learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.

[7] Sukatmo, H. A. Nugroho, B. H. Rusanto, and S. Soekirno, "Performance comparison of 1D-CNN and LSTM deep learning models for time series-based electric power prediction," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 13, no. 1, pp. 44–56, Jan. 2025, doi: 10.26760/elkomika.v13i1.44.

[8] Hyndman, R.J., C Athanasopoulos, G. (2018). Forecasting: Principles and Practice. OTexts

[9] Karno,A. S. B., Hastomo, W., Sutarno. 2020. OPTIMALISASI DATA TERBATAS PREDIKSI JANGKA PANJANG COVID-19 DENGAN KOMBINASI LSTM DAN GRU. SeNTIK. 4(1)

[10] Ali Khumaidi, R. Raafiudin, and I. P. Solihin, "Pengujian algoritma long short term memory untuk prediksi kualitas udara dan suhu Kota Bandung," *Jurnal Telematika*, vol. 15, no. 1, Institut Teknologi Harapan Bangsa, Bandung.