



Sistem Prediksi Kualitas Udara Menggunakan Algoritma Long Short-Term Memory (LSTM)

Muhammad Zaki Wicaksono^{*}, Ledy Elsera Astrianty

Fakultas Sains & Teknologi, Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia

Email: ^{1*}m.zaki.w.mz@gmail.com, ²ledyelsera@gmail.com

Email Penulis Korespondensi: m.zaki.w.mz@gmail.com

Abstrak—Pemantauan kualitas udara di Kota Yogyakarta yang bersifat konvensional dan statis yaitu hanya menyajikan laporan data historis (terdahulu) menghambat upaya mitigasi proaktif terhadap polusi udara. Penelitian ini bertujuan mengembangkan sistem prediksi kualitas udara menggunakan algoritma *Long Short-Term Memory* (LSTM), sebuah metode *deep learning* yang superior untuk analisis data deret waktu. Sistem ini memanfaatkan data historis dari Dinas Lingkungan Hidup (DLH) Kota Yogyakarta dari tahun 2022 hingga 2024, mencakup parameter pencemar seperti PM₁₀, PM_{2.5}, SO₂, CO, O₃, dan NO₂. Fokus utama prediksi adalah nilai ISPU (Indeks Standar Pencemar Udara) yang dihitung berdasarkan konsentrasi parameter pencemar tersebut. Metode penelitian meliputi pra-pemrosesan data seperti penanganan data hilang dengan interpolasi, perancangan arsitektur model LSTM dengan dua lapisan, pelatihan model, dan evaluasi performa menggunakan metrik *Mean Absolute Error* (MAE) dan *Mean Absolute Deviation* (MAD). Hasil penelitian menunjukkan model LSTM yang dikembangkan berhasil memberikan prediksi dengan performa yang baik, di mana rata-rata nilai MAE gabungan (4.85) secara signifikan lebih rendah dari rata-rata MAD data aktual (10.19), yang mengindikasikan bahwa kesalahan prediksi model lebih kecil dari variabilitas alami data. Luaran penelitian ini adalah purwarupa aplikasi dengan antarmuka pengguna grafis (GUI) yang mampu menampilkan prediksi kualitas udara untuk hari berikutnya, mengidentifikasi komponen pencemar kritis, dan menyajikan klasifikasi kondisi udara secara informatif.

Kata Kunci: Prediksi Kualitas Udara; Long Short-Term Memory (LSTM); Deep Learning; ISPU; Kota Yogyakarta

Abstract—Conventional and static air quality monitoring in Yogyakarta City which only presents historical (past) data reports hinders proactive mitigation efforts against air pollution. This research aims to develop an air quality prediction system using the Long Short-Term Memory (LSTM) algorithm, a deep learning method superior for time-series data analysis. The system utilizes historical data from the Yogyakarta City Environmental Agency (DLH) from 2022 to 2024, covering pollutant parameters such as PM₁₀, PM_{2.5}, SO₂, CO, O₃, and NO₂. The primary prediction focus is the AQI (Air Quality Index) value, calculated based on the concentration of these pollutant parameters. The research method includes data preprocessing, such as handling missing data with interpolation, designing a two-layer LSTM model architecture, model training, and performance evaluation using Mean Absolute Error (MAE) and Mean Absolute Deviation (MAD) metrics. The results show that the developed LSTM model successfully provides predictions with good performance, where the combined average MAE value (4.85) is significantly lower than the average MAD of the actual data (10.19), indicating that the model's prediction error is smaller than the natural variability of the data. The output of this research is a prototype application with a graphical user interface (GUI) capable of displaying air quality predictions for the next day, identifying critical pollutant components, and presenting air quality condition classifications informatively.

Keywords: Air Quality Prediction; Long Short-Term Memory (LSTM); Deep Learning; AQI; Yogyakarta City

1. PENDAHULUAN

Kota Yogyakarta sebagai pusat pariwisata dan pendidikan dengan kepadatan penduduk yang tinggi, menghadapi tantangan signifikan terkait kualitas udara. Peningkatan aktivitas industri, transportasi, dan pariwisata berkontribusi pada emisi polutan (Nurwanto dkk., 2025). Diantara jenis emisi polutan yang berbahaya adalah seperti Particulate Matter (PM₁₀ dan PM_{2.5}), Sulfur Dioksida (SO₂), Karbon Monoksida (CO), Ozon (O₃), dan Nitrogen Dioksida (NO₂) (Marlina, 2025). Paparan jangka panjang terhadap polutan ini berdampak negatif pada kesehatan masyarakat terutama kelompok rentan dengan meningkatkan risiko kesehatan seperti infeksi pernapasan, penyakit jantung, stroke, dan kanker paru-paru (Geneva, 2024). Permasalahan utama yang dihadapi saat ini adalah sistem pemantauan kualitas udara yang dilakukan oleh Dinas Lingkungan Hidup (DLH) Kota Yogyakarta masih bersifat konvensional. Data yang disajikan umumnya berupa laporan statis yang tidak memiliki kemampuan prediktif untuk mengantisipasi kondisi di masa depan. Pendekatan reaktif ini menghambat pengambilan keputusan proaktif bagi pemerintah maupun masyarakat dalam melakukan mitigasi dini terhadap potensi lonjakan polusi yang berbahaya.

Kualitas udara itu sendiri ditentukan berdasarkan standar baku mutu yang ditetapkan melalui Indeks Standar Pencemar Udara (ISPU), sebagaimana diatur dalam Peraturan Menteri Lingkungan Hidup dan Kehutanan (Permen LHK) Nomor P.14/MENLHK/SETJEN/KUM.1/7/2020. ISPU mengkonversi konsentrasi berbagai parameter pencemar utama—termasuk Partikulat (PM₁₀ dan PM_{2.5}), Ozon (O₃), Karbon Monoksida (CO), Sulfur Dioksida (SO₂), Nitrogen Dioksida (NO₂), dan Hidrokarbon (HC)—ke dalam skala indeks tunggal. Skala ini kemudian dikategorikan (misalnya Baik, Sedang, Tidak Sehat) untuk menyederhanakan informasi bagi publik. Fokus penelitian ini adalah pada prediksi konsentrasi mentah dari parameter-parameter tersebut, yang merupakan komponen dasar untuk menentukan ISPU.

Di era digital pemanfaatan kecerdasan buatan menawarkan solusi untuk analisis data lingkungan yang kompleks dan *non-linear*. Ini termasuk dalam ranah *Machine Learning*, sebuah cabang ilmu komputer yang bertujuan membuat program yang dapat belajar sendiri dari pengalaman atau data yang diberikan (Ningsih dkk., 2022). Secara khusus, *Deep Learning* sebagai subset dari *Machine Learning*, menggunakan arsitektur jaringan syaraf tiruan (JST) dengan banyak lapisan untuk mengenali pola-pola kompleks dalam data guna menghasilkan wawasan dan prediksi yang akurat



(Rondonuwu dkk., 2025). Untuk data sekuensial seperti data kualitas udara arsitektur *Recurrent Neural Networks* (RNN) dirancang khusus untuk mengenali pola dalam urutan data, di mana RNN tidak hanya mempertimbangkan *input* saat itu, tetapi juga informasi yang diperoleh sebelumnya (Jabat dkk., 2024). Algoritma *Long Short-Term Memory* (LSTM) merupakan varian lanjutan dari RNN yang dirancang khusus untuk menangani dan memodelkan data deret waktu (*time-series*). LSTM mampu mengatasi masalah gradien yang menghilang atau meledak pada RNN konvensional dan dapat mengingat informasi jangka panjang (Akbar dkk., 2024). Keunggulan fundamental LSTM terletak pada arsitektur internalnya yang unik, yang terdiri dari *cell state* dan mekanisme *gates* (*forget*, *input*, *output*). Struktur ini memungkinkannya menangkap dependensi temporal yang kompleks, seperti pola harian, mingguan, atau musiman dalam data kualitas udara.

Sejumlah penelitian sebelumnya telah membuktikan efektivitas LSTM dalam prediksi kualitas udara. Penelitian oleh (Ramadhan dkk., 2024) di DKI Jakarta menunjukkan bahwa model LSTM mampu memberikan prediksi yang konsisten dengan nilai *Root Mean Square Error* (RMSE) yang rendah dalam memprediksi berbagai parameter ISPU. Senada dengan itu, (Lestari & Mahendra, 2023) melakukan studi komparatif antara LSTM dan *Artificial Neural Network* (ANN), dan menyimpulkan bahwa LSTM memiliki performa yang lebih superior dalam memodelkan parameter spesifik PM2.5, yang dikenal sebagai salah satu polutan paling berbahaya. Lebih lanjut, studi literatur sistematis oleh (Alfian dkk., 2021) juga mengonfirmasi bahwa LSTM, baik sebagai model tunggal maupun dalam arsitektur *hybrid*, secara konsisten menunjukkan keunggulan dalam prediksi jangka pendek dan menengah untuk kualitas udara, menjadikannya relevan dan mutakhir untuk sistem pemantauan modern. Konteks implementasi *big data* dan sensor *Internet of Things* (IoT) juga memperkuat relevansi LSTM, dimana LSTM banyak digunakan untuk mengelola data deret waktu dari sensor *real-time*, menjadikannya model yang andal untuk sistem pemantauan cerdas skala kota (Asnawi dkk., 2024).

Meskipun terdapat alternatif seperti *Gated Recurrent Unit* (GRU) yang dikenal lebih efisien secara komputasi, terutama untuk dataset yang lebih kecil (Yudiskara dkk., 2023), LSTM terbukti memiliki keunggulan dalam menangani data deret waktu dengan pola temporal yang sangat kompleks, fluktuatif, dan berskala besar (Asnawi dkk., 2024). Keunggulan ini berkat mekanisme *cell state* dan tiga *gate* (*input*, *output*, *forget*) yang memberikan fleksibilitas lebih tinggi dibandingkan GRU dalam mengelola memori jangka panjang (Alfian dkk., 2021). Oleh karena itu, LSTM dipilih sebagai arsitektur yang paling relevan untuk memodelkan dinamika kualitas udara di Yogyakarta yang bersifat multivariat. Kesenjangan yang ada saat ini adalah kurangnya penerapan sistem prediktif serupa yang divalidasi secara spesifik untuk konteks Kota Yogyakarta. Tanpa adanya sistem prediksi yang akurat, upaya pemantauan akan terus bersifat reaktif, memperlambat respons terhadap lonjakan polusi dan menurunkan efektivitas program kesehatan masyarakat. Dengan adanya sistem ini diharapkan dapat membantu DLH dan masyarakat dalam memantau serta mengambil langkah preventif terhadap pencemaran udara secara lebih efektif.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem prediksi kualitas udara di Kota Yogyakarta menggunakan algoritma *Long Short-Term Memory* (LSTM), dengan mengukur performanya menggunakan metrik *Mean Absolute Error* (MAE) dan *Mean Absolute Deviation* (MAD). Hasil dari penelitian ini berupa purwarupa aplikasi grafis (GUI) yang mampu menampilkan hasil prediksi kualitas udara harian, mengidentifikasi komponen pencemar dominan, serta menyajikan klasifikasi kondisi udara secara informatif untuk mendukung pengambilan keputusan berbasis data di tingkat pemerintah daerah.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan kerangka kerja yang sistematis untuk membangun dan mengevaluasi model prediksi *deep learning*. Pendekatan kuantitatif dipilih karena fokus penelitian ini adalah pada pengukuran, analisis numerik, dan pengujian performa model secara objektif menggunakan data historis kualitas udara yang bersifat deret waktu. Dengan pendekatan ini, setiap hasil analisis dapat diukur dan dibandingkan melalui metrik evaluasi yang jelas, sehingga kesimpulan yang diperoleh bersifat empiris dan dapat dipertanggungjawabkan secara statistik. Selain itu, metode kuantitatif memberikan dasar yang kuat untuk melakukan validasi performa model dan mendukung replikasi penelitian di masa mendatang dengan kondisi data serupa.

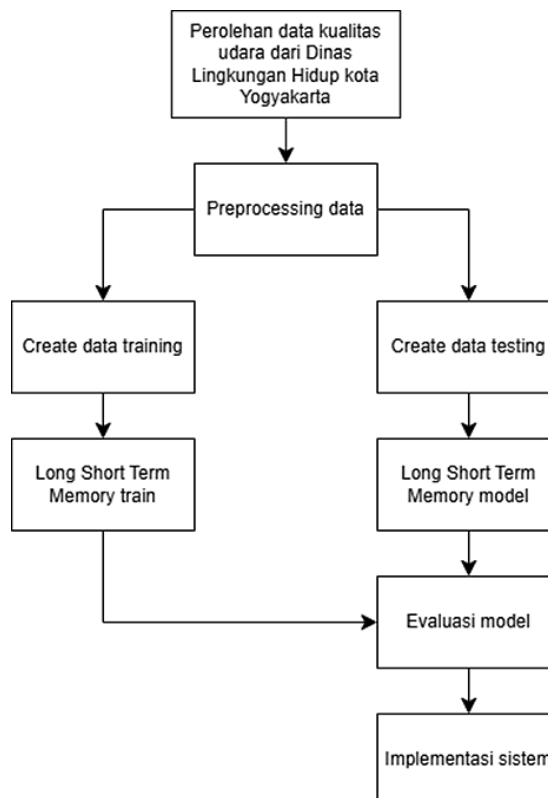
Metodologi penelitian ini dirancang agar setiap tahapan, mulai dari akuisisi data hingga validasi model, dilakukan secara terstruktur, runut, dan terukur. Alur penelitian, sebagaimana diilustrasikan pada Gambar 1, diawali dengan proses perolehan data kualitas udara historis yang bersumber dari Dinas Lingkungan Hidup (DLH) Kota Yogyakarta. Tahap ini memastikan bahwa data yang digunakan bersifat kredibel dan mencerminkan kondisi aktual kualitas udara di wilayah penelitian. Data mentah yang diperoleh kemudian masuk ke tahap pra-pemrosesan data secara intensif, yang mencakup proses pembersihan data dari nilai-nilai anomali, penanganan data hilang menggunakan metode interpolasi, serta normalisasi nilai untuk memastikan seluruh parameter berada dalam skala yang sebanding. Langkah ini sangat krusial karena kualitas *input* data secara langsung menentukan tingkat akurasi dan stabilitas model *deep learning* yang dikembangkan.

Setelah dataset siap digunakan, data dibagi secara kronologis menjadi dua subset utama, yaitu data latih (*training data*) dan data uji (*testing data*). Pembagian ini dilakukan dengan mempertahankan urutan waktu (*non-shuffled*) untuk menjaga kontinuitas pola temporal, di mana data latih digunakan untuk membentuk dan menyesuaikan bobot model, sedangkan data uji disisihkan untuk mengukur kemampuan model dalam melakukan generalisasi terhadap data baru

yang belum pernah dipelajari sebelumnya. *Model Long Short-Term Memory* (LSTM) kemudian dirancang dan dikonfigurasi secara sistematis, mulai dari penentuan jumlah lapisan (*layers*), jumlah neuron, fungsi aktivasi, hingga proses optimisasi. Model ini dilatih menggunakan data latih untuk mempelajari pola-pola temporal kompleks dan hubungan multivariat antar parameter pencemar udara, seperti PM10, PM2.5, SO₂, CO, O₃, NO₂, dan HC.

Setelah tahap pelatihan selesai, evaluasi performa model dilakukan secara kuantitatif menggunakan data uji untuk memastikan model tidak hanya menghafal pola dari data latih (*overfitting*), tetapi benar-benar mampu melakukan prediksi yang akurat terhadap data baru. Evaluasi dilakukan dengan menghitung nilai *Mean Absolute Error* (MAE) dan *Mean Absolute Deviation* (MAD) sebagai metrik utama. Nilai MAE menunjukkan tingkat kesalahan rata-rata dari prediksi terhadap data aktual, sedangkan MAD digunakan sebagai pembanding untuk menilai seberapa jauh variasi alami dalam data. Perbandingan kedua metrik ini memberikan gambaran yang jelas tentang sejauh mana model memiliki nilai tambah prediktif dibandingkan metode sederhana berbasis rata-rata.

Tahap akhir dari penelitian ini adalah implementasi sistem, di mana model LSTM yang telah tervalidasi dan menunjukkan performa terbaik diintegrasikan ke dalam sebuah purwarupa antarmuka pengguna grafis (GUI) berbasis desktop. Purwarupa ini berfungsi sebagai bukti konsep (*proof of concept*) yang mendemonstrasikan bagaimana model prediktif dapat diterapkan secara praktis untuk membantu pengguna dalam menilai kondisi udara harian. Melalui GUI ini, pengguna dapat memasukkan data historis, melakukan prediksi otomatis, dan melihat hasilnya dalam bentuk numerik, klasifikasi ISPU, serta visualisasi grafik tren. Implementasi ini sekaligus menjadi tahapan translasi hasil penelitian dari ranah komputasional menuju aplikasi nyata yang dapat dimanfaatkan oleh instansi lingkungan maupun masyarakat umum. Alur tahapan penelitian ini secara keseluruhan dirangkum dalam arsitektur model yang disajikan pada Gambar 1.



Gambar 1 Arsitektur Model

2.1 Sumber dan Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder berjenis deret waktu (*time-series*) yang bersumber dari situs resmi Dinas Lingkungan Hidup (DLH) Kota Yogyakarta. Data mentah ini diperoleh dengan mengunduh laporan harian yang tersedia secara publik, yang kemudian dikompilasi secara manual ke dalam format CSV tunggal untuk memfasilitasi pengolahan. Data ini mencakup tujuh parameter pencemar udara utama yang menjadi fokus penelitian, yaitu PM10, PM2.5, SO₂, CO, O₃, NO₂, dan HC. Periode data yang diambil adalah tiga tahun penuh, dari awal tahun 2022 hingga akhir tahun 2024. Pemilihan rentang waktu tiga tahun ini dianggap esensial untuk memastikan model memiliki data historis yang cukup untuk mempelajari pola-pola temporal yang kompleks. Contoh tampilan data mentah harian, yang sekaligus menunjukkan perlunya penanganan data hilang ("Tidak Ada Data"), disajikan pada Gambar 2.

**Indek Standar Pencemar Udara (ISPU) Rata-Rata
Kota : Yogyakarta
Tahun Data : 2022**

No.	Waktu	PM ₁₀	PM _{2.5}	SO ₂	CO	O ₃	NO ₂	HC	Critical Component	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	
1	01 Januari 2022	16.2	0.0	7.6	18.2	1.0	11.0	7.0	CO	
2	02 Januari 2022	10.4	0.0	7.5	16.0	1.0	11.0	7.7	CO	
3	03 Januari 2022	10.7	0.0	8.6	16.5	1.0	10.5	6.6	CO	
4	04 Januari 2022	19.3	0.0	8.5	18.9	0.8	10.0	5.9	CO	
5	05 Januari 2022	Tidak Ada Data								
6	06 Januari 2022	0.0	0.0	0.4	14.4	0.0	6.4	2.7	CO	
7	07 Januari 2022	0.0	0.0	9.1	8.0	0.0	6.0	2.4	SO ₂	
8	08 Januari 2022	0.0	0.0	29.4	8.3	0.0	7.5	3.2	SO ₂	

Gambar 2 Contoh Tampilan Data Mentah Harian dari DLH Kota Yogyakarta

2.2 Pra-pemrosesan Data

Tahap pra-pemrosesan data merupakan langkah krusial untuk mengubah data mentah yang tidak terstruktur dan seringkali tidak lengkap menjadi dataset yang bersih, lengkap, dan siap untuk diolah oleh model *deep learning*. Proses ini tidak dapat diabaikan, karena kualitas dataset secara langsung berdampak pada akurasi dan stabilitas model. Proses ini dibagi menjadi empat langkah utama yang sistematis:

- Penggabungan dan Pembersihan Data:** Langkah awal adalah mengkonsolidasikan tiga file data tahunan (2022, 2023, 2024) menjadi satu *dataframe* tunggal menggunakan pustaka Pandas. Selama proses ini, kolom-kolom yang tidak relevan untuk pemodelan prediktif, seperti 'No.' (nomor urut) dan 'Critical Component' (yang merupakan hasil analisis, bukan data mentah), dihapus dari dataset untuk mengurangi *noise* dan kompleksitas yang tidak perlu.
- Penanganan Nilai Hilang (Missing Values):** Dalam dataset mentah, nilai yang hilang atau tidak tercatat sering direpresentasikan dengan simbol non-numerik seperti '-'. Simbol-simbol ini pertama-tama dikonversi menjadi format standar NaN (*Not a Number*). Setelah itu, metode interpolasi linear diterapkan untuk mengisi semua nilai NaN. Metode ini dipilih secara spesifik karena data kualitas udara bersifat kontinu dan memiliki dependensi temporal yang kuat. Interpolasi linear mampu memperkirakan nilai yang hilang berdasarkan tren dari titik data sebelum dan sesudahnya, sehingga menjaga kontinuitas dan integritas pola deret waktu, yang jauh lebih baik daripada mengisi dengan nilai rata-rata (*mean*) atau nol.
- Konversi Tipe Data:** Untuk memastikan data dapat diproses secara matematis, seluruh kolom parameter polutan (PM₁₀, PM_{2.5}, SO₂, CO, O₃, NO₂, dan HC.) dikonversi menjadi tipe data numerik (*float*). Selain itu, kolom 'Waktu' dikonversi menjadi format *datetime*. Langkah ini sangat penting karena memungkinkan dataset untuk diindeks dan diurutkan secara kronologis, yang merupakan syarat fundamental untuk analisis deret waktu.
- Normalisasi Data:** Seluruh fitur data dinormalisasi menggunakan *Min-Max Scaler* dari pustaka *Scikit-learn*. Proses ini mengubah skala semua nilai fitur ke dalam rentang seragam antara 0 dan 1. Normalisasi adalah persyaratan penting untuk model jaringan syaraf seperti LSTM. Ini bertujuan untuk mencegah bias yang disebabkan oleh perbedaan rentang nilai antar parameter (misalnya, nilai PM₁₀ yang berkisar puluhan hingga ratusan dibandingkan CO yang berkisar satuan) dan membantu mempercepat serta menstabilkan proses konvergensi *gradient descent* selama pelatihan.

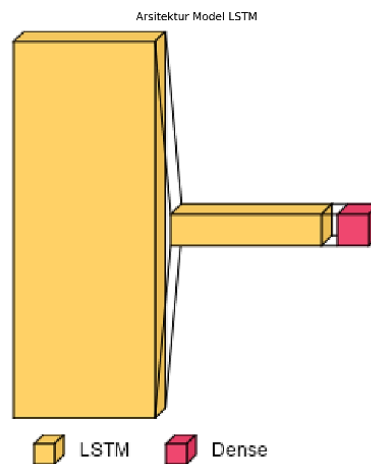
2.3 Perancangan dan Pelatihan Model LSTM

Setelah dataset bersih dan ternormalisasi, langkah inti dari penelitian ini adalah perancangan arsitektur dan pelatihan model LSTM. Pendekatan yang digunakan termasuk dalam kategori *Supervised Learning* dimana model dilatih menggunakan dataset yang telah memiliki label, yaitu setiap data masukan (*input* historis) memiliki pasangan keluaran (*output* aktual) yang diketahui (Ningsih dkk., 2022).

- Pembagian Data Latih dan Uji:** Dataset yang telah diproses dibagi menjadi dua set: 80% sebagai data latih (*training data*) dan 20% sebagai data uji (*testing data*). Pembagian ini dilakukan secara berurutan dengan mengatur parameter *shuffle=False*. Dalam pemodelan *time-series*, mengacak data adalah kesalahan metodologis karena akan menghancurkan pola temporal. Model harus dilatih pada data dari masa lalu (80% pertama) dan diuji kemampuannya untuk memprediksi data di masa depan (20% terakhir) yang belum pernah dilihatnya.
- Pembuatan Sequences (Sequencing):** LSTM tidak memproses data titik per titik, melainkan dalam bentuk *sekuens* (urutan). Data ditransformasikan ke dalam format sekuensial di mana sebuah *timestep* (panjang urutan) sebesar 3 hari ditetapkan. Penetapan ini dipilih berdasarkan eksperimen komparatif yang mempertimbangkan volume data historis yang tersedia (2022–2024), di mana penggunaan *timestep* yang lebih panjang (7 hari) justru menurunkan akurasi prediksi. Hal ini mengindikasikan bahwa dengan rentang data tiga tahun tersebut, model lebih efektif menangkap dependensi jangka pendek (*short-term dependencies*) dibandingkan pola mingguan. Ini berarti model akan menggunakan data dari 3 hari sebelumnya ($t-3$, $t-2$, $t-1$) sebagai satu set *input*, untuk memprediksi data pada hari berikutnya (t). Format *input* ini menghasilkan data 3-dimensi dengan bentuk [jumlah_sampel, *timesteps*, jumlah_fitur], yang sesuai dengan arsitektur *input* LSTM.

- c. **Arsitektur Model:** Model dibangun menggunakan arsitektur *Sequential* dari *Keras*. Arsitektur ini terdiri dari dua lapisan LSTM yang ditumpuk (*stacked*) dan satu lapisan Dense sebagai keluaran:
1. Lapisan LSTM pertama memiliki 60 unit neuron dan diatur dengan `return_sequences=True`. Parameter ini krusial karena menginstruksikan lapisan tersebut untuk mengembalikan *output* pada setiap *timestep*, bukan hanya *timestep* terakhir, sehingga *output* sekuensial ini dapat diteruskan sebagai *input* ke lapisan LSTM berikutnya.
 2. Lapisan LSTM kedua memiliki 64 unit neuron. Penggunaan dua lapisan yang ditumpuk bertujuan agar model dapat mempelajari pola temporal dalam berbagai tingkat abstraksi yang lebih kompleks, yang mungkin tidak tertangkap oleh satu lapisan saja. Arsitektur sel LSTM itu sendiri, dengan mekanisme *forget gate*, *input gate*, dan *output gate*, memungkinkan model untuk mengingat informasi relevan dari jangka panjang dan melupakan yang tidak relevan (Nugroho dkk., 2023)
 3. Lapisan Dense terakhir bertindak sebagai lapisan keluaran dengan jumlah neuron sesuai jumlah fitur yang diprediksi (7 parameter). Fungsi aktivasi ReLU (*Rectified Linear Unit*) dipilih secara spesifik pada lapisan ini untuk menegakkan batasan fisik data pencemaran udara, di mana konsentrasi polutan tidak mungkin bernilai negatif. Penggunaan ReLU menjamin bahwa output model selalu berada dalam rentang non-negatif (≥ 0) tanpa memerlukan pemrosesan tambahan untuk menangani nilai negatif yang mungkin muncul jika menggunakan fungsi aktivasi Linear.

Arsitektur model yang digunakan divisualisasikan secara skematis pada Gambar 3, dengan ringkasan detail lapisan dan jumlah parameter model disajikan pada Tabel 1.



Gambar 3 Visualisasi Arsitektur Model LSTM

Tabel 1 Ringkasan Arsitektur Model Keras

Layer (type)	Output Shape	Param
lstm (LSTM)	(None, 3, 60)	16,320
lstm_1 (LSTM)	(None, 64)	32,000
dense (Dense)	(None, 7)	455

- d. **Kompilasi dan Pelatihan:** Model dikompilasi menggunakan *optimizer* 'adam', yang merupakan pilihan standar yang efisien karena kemampuannya menyesuaikan learning rate secara adaptif. *Loss function* yang digunakan adalah *Mean Absolute Error* (MAE), yang mengukur selisih absolut rata-rata antara prediksi dan nilai sebenarnya. Model kemudian dilatih (*fit*) pada data latih sekuensial selama 200 *epoch*. Satu *epoch* adalah satu siklus penuh pelatihan pada seluruh data latih. Ukuran *batch* (*batch size*) 32 digunakan, yang berarti model memperbarui bobotnya setiap 32 sampel, memberikan keseimbangan antara efisiensi komputasi dan stabilitas pelatihan.

2.4 Evaluasi Model

Performa model dievaluasi secara kuantitatif menggunakan data uji yang belum pernah dilihat sebelumnya. Tahap ini krusial untuk memvalidasi kemampuan generalisasi model terhadap data baru, bukan sekadar menghafal data latih. Proses evaluasi ini berfokus pada metrik statistik yang mengukur akurasi prediksi dan membandingkannya dengan variabilitas alami data.

2.4.1 Mean Absolute Error (MAE)

Metrik utama untuk mengukur akurasi prediksi dalam penelitian ini adalah *Mean Absolute Error* (MAE). MAE menguantifikasi besaran rata-rata dari kesalahan yang terjadi antara nilai aktual dan nilai prediksi. Formula MAE dihitung sebagai rata-rata dari nilai absolut selisih antara setiap prediksi dengan nilai aktualnya, seperti yang ditunjukkan pada Persamaan (1). Pemilihan MAE sebagai *loss function* dan metrik evaluasi utama didasarkan pada dua keunggulan utamanya. Pertama, interpretasinya lugas karena unit kesalahannya sama dengan unit data asli (misalnya,



MAE 4.62 berarti kesalahan rata-rata adalah 4.62 $\mu\text{g}/\text{m}^3$). Kedua, MAE bersifat lebih *robust* terhadap *outlier* atau lonjakan data ekstrem dibandingkan dengan metrik kuadrat seperti *Root Mean Square Error* (RMSE). RMSE akan memberi bobot penalti yang sangat besar pada kesalahan besar, yang mungkin tidak diinginkan untuk data kualitas udara yang secara alami fluktuatif (Rayhan Rizal Mahendra dkk., 2024) Dengan menggunakan MAE, model dievaluasi berdasarkan kesalahan rata-rata yang lebih representatif terhadap performa umum. Pemilihan MAE ini juga sejalan dengan penelitian relevan lainnya di bidang prediksi kualitas udara (Alfian dkk., 2021; Lestari & Mahendra, 2023)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |X_i - Y_i| \quad (1)$$

Dimana: n = jumlah total data, X_i = nilai data aktual ke- i , Y_i = nilai data prediksi ke- i

2.4.2 Mean Absolute Deviation (MAD)

Untuk memberikan konteks terhadap nilai MAE, diperlukan sebuah baseline atau tolok ukur. Sebuah model yang baik harus berkinerja lebih baik daripada tebakan sederhana. Dalam penelitian ini, digunakan *Mean Absolute Deviation* (MAD) sebagai baseline tersebut. MAD adalah ukuran dispersi statistik yang mengukur rata-rata simpangan absolut dari setiap titik data individual terhadap nilai rata-rata aritmatika dari himpunan data tersebut. Secara sederhana MAD berfungsi sebagai tolak ukur untuk memahami seberapa besar variabilitas, fluktuasi, atau *noise* alami dalam data itu sendiri (Syahrul dkk., 2024)). Nilai MAD merepresentasikan tingkat kesulitan dari dataset maka jika MAD tinggi, data sangat fluktuatif, dan jika MAD rendah, data sangat stabil. Ini adalah baseline yang jauh lebih baik daripada sekadar mengharapkan MAE mendekati nol, karena MAD merepresentasikan kesalahan rata-rata yang akan didapat jika kita hanya memprediksi nilai rata-rata data setiap saat. Rumus MAD ditunjukkan pada Persamaan (2).

$$\text{MAD} = \frac{1}{n} \sum_{i=1}^n |X_i - \bar{x}| \quad (2)$$

Dimana: n = jumlah total data, X_i = nilai data aktual ke- i , \bar{x} = nilai rata-rata (*mean*) dari seluruh data aktual

2.4.3 Kriteria Performa Mode

Kriteria utama untuk menilai signifikansi dan nilai tambah (*value added*) dari model dalam penelitian ini adalah dengan membandingkan nilai MAE prediksi (dari Persamaan 1) dengan MAD data aktual (dari Persamaan 2). Sebuah model dianggap memiliki performa yang baik dan nilai prediktif yang nyata jika nilai MAE (rata-rata kesalahan model) secara signifikan lebih rendah daripada nilai MAD (rata-rata variabilitas alami data). Logika di balik kriteria ini sangat fundamental: jika seorang peneliti hanya memprediksi nilai rata-rata data setiap saat (sebuah model baseline yang sangat sederhana), kesalahan rata-rata (MAE) dari model sederhana tersebut akan kira-kira sama dengan simpangan alami data (MAD). Oleh karena itu, untuk membuktikan bahwa model *deep learning* yang kompleks ini berguna, model tersebut harus secara konsisten mengalahkan baseline sederhana ini. Jika $\text{MAE} < \text{MAD}$, ini mengindikasikan bahwa model mampu menangkap pola temporal yang bermakna dalam data dan memberikan prediksi yang lebih akurat daripada sekadar menggunakan nilai rata-rata historis.

2.5 Perangkat Penelitian

Implementasi teknis dari penelitian ini didukung oleh serangkaian perangkat lunak yang spesifik. Bahasa pemrograman utama yang dipilih adalah Python, yang merupakan bahasa populer dengan sintaks yang jelas, logis, dan mudah dipelajari, serta sering digunakan dalam penanganan *big data* dan *machine learning* (Nova dkk., 2024). Seluruh proses pengkodean, pengujian, dan pengembangan model dilakukan menggunakan Visual Studio Code, sebuah *source code editor* ringan namun kuat yang dikembangkan oleh Microsoft dan mendukung penuh ekosistem Python (Riftianto & Amirullah, 2024).

Pengembangan sistem bergantung pada beberapa pustaka inti Python. Pandas digunakan untuk seluruh proses akuisisi, pembersihan, dan manipulasi data tabular (*DataFrame*). Scikit-learn (Sklearn) dimanfaatkan untuk tugas pra-pemrosesan data, terutama *MinMaxScaler* untuk normalisasi fitur dan *train_test_split* untuk pembagian data. Arsitektur model *deep learning* dibangun, dilatih, dan dievaluasi menggunakan pustaka TensorFlow dengan API Keras. Terakhir, purwarupa antarmuka pengguna grafis (GUI) dibangun menggunakan pustaka standar Python, Tkinter, dengan dukungan Matplotlib untuk visualisasi grafik yang terintegrasi.

3. HASIL DAN PEMBAHASAN

Prosedur eksperimental yang menjadi landasan validasi model melibatkan pengolahan data historis kualitas udara harian dari Dinas Lingkungan Hidup (DLH) Kota Yogyakarta periode 2022 hingga 2024. Data mentah yang bersumber dari laporan tahunan tersebut dikonsolidasikan menjadi satu dataset tunggal dan melalui serangkaian tahapan pra-pemrosesan yang komprehensif. Proses ini mencakup eliminasi fitur yang tidak relevan, konversi tipe data, serta penanganan nilai yang hilang (*missing values*) melalui metode interpolasi linear untuk menjaga kontinuitas tren. Seluruh fitur input selanjutnya dinormalisasi ke dalam skala $[0, 1]$ menggunakan *MinMax Scaler* guna optimalisasi komputasi. Guna menjaga integritas pola deret waktu (*time series*), dataset dipartisi secara kronologis (*non-shuffled*) dengan proporsi 80% dialokasikan sebagai data latih (*training set*) dan 20% sebagai data uji (*testing set*). Model

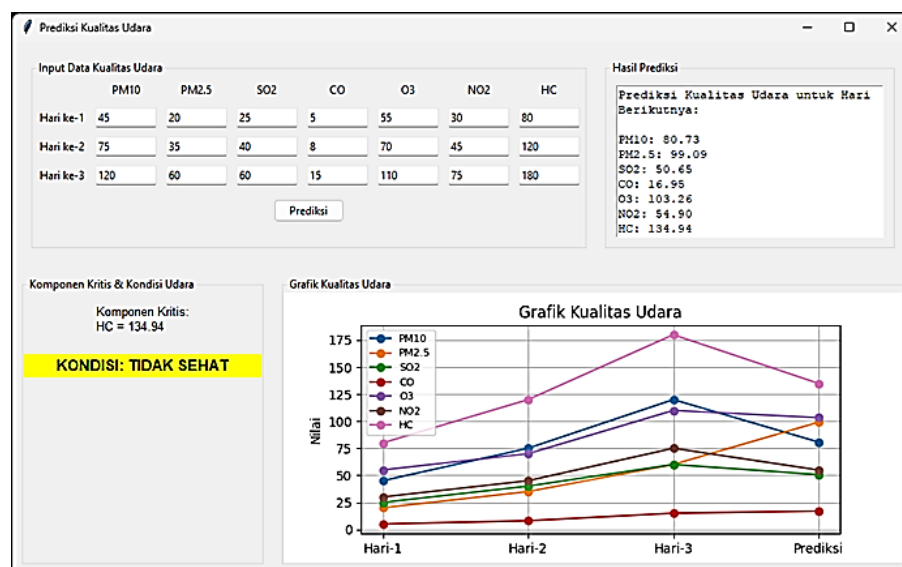
prediksi dibangun menggunakan arsitektur *Long Short-Term Memory* (LSTM) yang terdiri dari dua lapisan tersembunyi (*stacked layers*) dengan masing-masing 60 dan 64 unit neuron, serta lapisan luaran Dense dengan fungsi aktivasi ReLU. Model dilatih menggunakan konfigurasi *window size* (*timestep*) 3 hari selama 200 *epoch* dengan *batch size* 32 dan optimisasi Adam. Hasil pelatihan ini kemudian divalidasi menggunakan data uji sebelum akhirnya diintegrasikan sebagai mesin prediksi utama dalam arsitektur sistem aplikasi.

3.1 Implementasi Sistem

Hasil akhir fungsional dari penelitian ini adalah sebuah purwarupa (prototipe) aplikasi desktop dengan Antarmuka Pengguna Grafis (GUI) yang dibangun menggunakan pustaka Tkinter di Python. Aplikasi ini dirancang untuk menjembatani model *deep learning* yang kompleks dengan pengguna akhir, mengubah data numerik mentah menjadi informasi yang dapat ditindaklanjuti. Alur kerja pengguna pada GUI ini dimulai dengan memasukkan data historis kualitas udara secara manual. Pengguna perlu mengisi nilai untuk ketujuh parameter (PM10, PM2.5, SO₂, CO, O₃, NO₂, dan HC) untuk tiga hari terakhir (t-3, t-2, t-1) ke dalam kolom *input* yang tersedia. Desain *input* tiga hari ini dipilih karena sesuai dengan *timestep* = 3 yang digunakan selama pelatihan model.

Ketika pengguna menginisiasi prediksi melalui tombol "Prediksi", sebuah alur kerja terorkestrasi dieksekusi di latar belakang. Pertama, data mentah yang dimasukkan oleh pengguna dari kolom-kolom antarmuka akan divalidasi dan dikumpulkan. Kedua, data ini menjalani transformasi krusial: data tersebut dinormalisasi menggunakan parameter *scaler* yang identik dengan yang telah disimpan dari fase pelatihan. Langkah ini penting untuk memastikan konsistensi data antara pelatihan dan implementasi. Ketiga, data yang telah dinormalisasi tersebut (dalam bentuk 2D) direstrukturisasi menjadi format sekuensial 3 dimensi yang sesuai dengan arsitektur *input* model LSTM (sampel, *timesteps*, fitur). Keempat, model LSTM yang telah dilatih dan disimpan sebelumnya (dalam format .h5) dimuat untuk melakukan *inference* (prediksi) pada data *sekuens* yang telah disiapkan. Akhirnya, *output* prediksi dari model, yang masih dalam skala ternormalisasi, dikembalikan ke skala aslinya melalui operasi transformasi invers, sehingga menghasilkan nilai prediktif yang dapat diinterpretasikan.

Hasil prediksi ini kemudian disajikan kepada pengguna melalui tiga komponen luaran utama pada antarmuka. Pertama, luaran numerik murni ditampilkan secara jelas dalam area teks, merinci nilai prediksi untuk setiap dari tujuh parameter polutan. Kedua, sistem melakukan analisis kualitatif instan: logika aplikasi secara otomatis mengidentifikasi parameter dengan nilai prediksi tertinggi, yang kemudian ditetapkan sebagai "Komponen Kritis". Nilai kritis ini kemudian dipetakan ke standar ISPU yang telah ditentukan untuk menghasilkan klasifikasi kondisi (misalnya, "BAIK", "SEDANG", "TIDAK SEHAT"). Klasifikasi ISPU ini didasarkan pada standar Permen LHK No. P.14/MENLHK/SETJEN/KUM.1/7/2020. Klasifikasi ini disajikan dengan penanda visual berupa latar belakang berwarna (hijau, biru, kuning, dst.) untuk interpretasi yang cepat dan mudah dipahami. Ketiga, sebuah visualisasi grafik deret waktu dihasilkan menggunakan pustaka Matplotlib yang diintegrasikan ke dalam antarmuka Tkinter. Grafik ini memetakan tren dari tiga hari *input* historis dan satu hari prediksi, memungkinkan pengguna untuk memahami konteks dan arah perubahan kualitas udara secara visual. Tampilan antarmuka purwarupa yang mengilustrasikan ketiga komponen luaran ini—prediksi numerik, klasifikasi kondisi, dan visualisasi grafik—secara terintegrasi pasca-pemrosesan disajikan pada Gambar 4.



Gambar 4 Tampilan GUI

3.2 Analisis Performa Model

Evaluasi model dilakukan dengan membandingkan MAE prediksi terhadap MAD data aktual. Hasil perbandingan untuk setiap parameter disajikan secara rinci pada Tabel 2.



Tabel 2 Perbandingan MAE Prediksi dan MAD Aktual

No.	Parameter	MAE Prediksi	MAD Data Aktual	Keterangan
1.	PM10	4.62	7.22	Performa Baik
2.	PM2.5	8.18	11.49	Performa Baik
3.	SO ₂	3.81	10.12	Performa Baik
4.	CO,	1.22	1.66	Performa Baik
5.	O ₃	5.42	17.56	Performa Baik
6.	NO ₂	7.00	22.94	Performa Baik
7.	HC	3.73	0.35	Perlu Peningkatan
	Rata-rata	4.85	10.19	Performa Baik

Data pada Tabel 1 menunjukkan bahwa model LSTM yang dikembangkan memiliki performa yang sangat baik dan valid secara statistik. Temuan utamanya adalah nilai rata-rata MAE gabungan untuk semua fitur (4.85) secara signifikan lebih rendah dari rata-rata MAD data aktual (10.19). Hal ini memenuhi kriteria performa utama (MAE < MAD) yang telah ditetapkan pada 2.4.3. Hasil ini secara kuantitatif membuktikan bahwa model mampu memberikan prediksi yang jauh lebih akurat daripada sekadar mengandalkan nilai rata-rata data. Ini mengindikasikan bahwa arsitektur *stacked* LSTM dengan *timestep* 3 hari berhasil menangkap pola temporal yang kompleks dan dependensi antar variabel dalam data kualitas udara Yogyakarta.

Analisis lebih rinci pada setiap parameter mengungkapkan bahwa model berhasil memprediksi 6 dari 7 parameter dengan baik. Model menunjukkan performa paling impresif pada parameter O₃ (MAE 5.42 vs MAD 17.56) dan NO₂ (MAE 7.00 vs MAD 22.94). Data untuk kedua parameter ini memiliki variabilitas alami (MAD) yang sangat tinggi, yang berarti nilainya sangat fluktuatif dan sulit diprediksi. Fakta bahwa nilai MAE model jauh lebih rendah daripada MAD menunjukkan bahwa model tidak terkecoh oleh fluktuasi tersebut dan berhasil menemukan pola yang mendasarinya, memberikan nilai prediktif yang sangat tinggi. Performa yang solid juga terlihat pada PM10, PM2.5, SO₂, dan CO, di mana semua MAE secara nyaman berada di bawah MAD. Satu-satunya anomali adalah pada parameter HC, di mana MAE (3.73) lebih tinggi dari MAD (0.35), yang menandakan performa yang perlu peningkatan spesifik untuk parameter tersebut.

3.2 Pembahasan

Temuan bahwa rata-rata MAE (4.85) secara signifikan lebih rendah dari rata-rata MAD (10.19) merupakan inti dari kontribusi penelitian ini. Ini membuktikan bahwa penerapan arsitektur *deep learning* seperti LSTM memberikan nilai tambah prediktif yang nyata. Model ini berhasil melampaui baseline statistik sederhana, yang berarti model tidak hanya menghafal, tetapi benar-benar belajar dan menggeneralisasi pola tersembunyi dalam data deret waktu. Keberhasilan ini secara langsung menjawab permasalahan yang diangkat dalam pendahuluan: sistem ini menyediakan alat proaktif untuk mengantisipasi kualitas udara. Diharapkan menjadi sebuah langkah maju signifikan dari laporan DLH konvensional yang sebelumnya bersifat statis dan reaktif.

Keberhasilan ini juga mengkontekstualisasikan penelitian ini dalam literatur yang ada. Temuan ini sangat sejalan dengan penelitian oleh (Lestari & Mahendra, 2023; Ramadhan dkk., 2024) di DKI Jakarta, yang keduanya menyimpulkan bahwa LSTM merupakan metode yang andal dengan tingkat kesalahan (RMSE atau MAE) yang rendah untuk memprediksi polusi udara. Penelitian ini berhasil mereplikasi keberhasilan tersebut dalam konteks geografis yang berbeda (Kota Yogyakarta), yang memperkuat validitas eksternal LSTM sebagai alat prediksi kualitas udara yang *robust* di berbagai kota besar di Indonesia. Selain itu, ini juga memvalidasi pilihan arsitektur *stacked* LSTM atas model yang lebih sederhana seperti ANN, sebagaimana disimpulkan oleh (Lestari & Mahendra, 2023).

Analisis terhadap anomali parameter HC (MAE 3.73 > MAD 0.35) juga memberikan wawasan penting yang menjadi keterbatasan penelitian ini. Ini tidak berarti model gagal total. Sebaliknya, ini mengindikasikan bahwa data HC dalam dataset uji memiliki variabilitas alami yang sangat rendah (sangat stabil atau hampir datar). Ketika data aktual sangat stabil, nilai rata-ratanya (sebagai baseline MAD) menjadi prediktor yang sangat sulit dikalahkan. Kesalahan prediksi model (MAE 3.73), meskipun secara absolut masih tergolong kecil, menjadi terlihat signifikan jika dibandingkan dengan data yang hampir tidak berfluktuasi. Hal ini menyoroti keterbatasan model: performanya bergantung pada karakteristik data, dan mungkin memerlukan *tuning* hiperparameter spesifik atau bahkan arsitektur berbeda (seperti GRU, yang disebut oleh (Yudiskara dkk., 2023)) untuk parameter dengan variabilitas sangat rendah.

Terakhir, kontribusi praktis dari penelitian ini diwujudkan dalam purwarupa GUI. GUI ini berfungsi sebagai *proof of concept* yang vital, menunjukkan bagaimana model *deep learning* yang kompleks dan abstrak dapat diterjemahkan menjadi alat bantu keputusan yang sederhana dan dapat ditindaklanjuti. Dengan menyajikan data dalam bentuk klasifikasi ("KONDISI: TIDAK SEHAT") dan visualisasi tren, aplikasi ini memberdayakan pengguna non-teknis (baik masyarakat umum maupun staf instansi) untuk membuat keputusan berbasis data. Namun, ketergantungan GUI pada *input* manual juga merupakan keterbatasan yang jelas, yang mengarahkan pada kebutuhan pengembangan selanjutnya.



4. KESIMPULAN

Penelitian ini bertujuan untuk mengatasi keterbatasan pemantauan kualitas udara konvensional di Kota Yogyakarta dengan merancang, membangun, dan mengevaluasi sistem prediksi menggunakan algoritma *deep learning Long Short-Term Memory* (LSTM). Berdasarkan hasil implementasi dan evaluasi yang telah dilakukan, dapat ditarik kesimpulan bahwa sistem telah berhasil dibangun dan divalidasi. Sistem ini terbukti mampu mengolah data historis deret waktu yang kompleks dari DLH, yang mencakup tujuh parameter polutan utama, dan menghasilkan prediksi yang andal untuk hari berikutnya. Validasi kuantitatif menunjukkan kinerja model yang sangat baik, dibuktikan dengan nilai rata-rata MAE gabungan (4.85) yang secara signifikan lebih rendah dari rata-rata MAD data aktual (10.19). Ini memenuhi kriteria validasi utama ($MAE < MAD$), yang mengkonfirmasi bahwa model mampu menghasilkan prediksi dengan tingkat kesalahan yang lebih kecil dari variabilitas alami data itu sendiri, dengan 6 dari 7 parameter polutan berhasil diprediksi secara akurat. Luaran dari penelitian ini adalah sebuah purwarupa aplikasi fungsional dengan antarmuka pengguna grafis (GUI) yang menerjemahkan data numerik kompleks menjadi informasi yang dapat ditindaklanjuti, seperti identifikasi komponen pencemar kritis dan klasifikasi kondisi udara yang mudah dipahami. Meskipun model menunjukkan performa kuat, penelitian ini memiliki keterbatasan, terutama pada prediksi parameter HC yang anomali (kemungkinan karena variabilitas data yang sangat rendah) dan ketergantungan purwarupa GUI pada *input* data manual. Untuk pengembangan di masa mendatang, disarankan untuk melakukan *tuning* hiperparameter lebih lanjut atau eksplorasi arsitektur alternatif (seperti GRU) untuk menangani parameter dengan variabilitas rendah, serta mengintegrasikan sistem ke dalam platform berbasis web dengan *pipeline* data otomatis dari DLH untuk menggantikan *input* manual dan meningkatkan aksesibilitas.

REFERENCES

- Akbar, J., Ali Setyo Yudono, M., & Lucia Kharisma, I. (2024). Peramalan Harga Bitcoin Cash-Usd (Bch-Usd) Pada Time Frame Harian Menggunakan LSTM. *Jurnal Mnemonic*, 7(2), 184–191. <https://doi.org/10.36040/mnemonic.v7i2.10121>
- Alfian, H., Wahyuni, S., Revalino, A., Mirano, M. F., Rahmayana, E., & Mukhtar, H. (2021). Teknik Machine Learning Untuk Analisa Klasifikasi Kualitas Udara: A Review. *Journal of Software Engineering and Information Systems*, 4(2), 108–118. <https://doi.org/10.37859/seis.v4i2.7617>
- Asnawi, M. F., Fitriyanto, N., & Pamoengkas, M. A. (2024). Analisis Big Data Untuk Pemantauan Kualitas Udara: Pendekatan, Implementasi, dan Tantangan dalam Studi Lingkungan. *Journal of Engineering and Informatic*, 3(1). <https://doi.org/https://doi.org/10.56854/jei.v3i1.258>
- Geneva. (2024, Juni 25). *Health consequences of air pollution on populations*. World Health Organization (WHO). <https://www.who.int/news/item/25-06-2024-what-are-health-consequences-of-air-pollution-on-populations>
- Lestari, I. G. A. N., & Mahendra, I. N. D. A. (2023). Prediksi Kualitas Udara dengan Menggunakan Metode Long Short-Term Memory dan Artificial Neural Network. *Jurnal Sistem dan Informatika (JSI)*, 17(2), 121–129. <https://doi.org/10.30864/jsi.v17i2.565>
- Jabat, D. E. B., Sipayung, L. Y., & Dakhi, K. R. syahputra. (2024). Penerapan Algoritma Recurrent Neural Networks (RNN) Untuk Klasifikasi Ulos Batak Toba. *Seminar Nasional Inovasi Sains Teknologi Informasi Komputer*, 1(2), 3025–8715. <https://ejournal.ust.ac.id/index.php/SNISTIK/article/view/3697>
- Marlina. (2025, Oktober 28). *Air quality remains moderate in Dhaka*. alreinamea. <https://alreinamea.com/air-quality-remains-moderate-in-dhaka/>
- Ningsih, P. T. S., Gusvarizon, M., & Hermawan, R. (2022). Analisis Sistem Pendeteksi Penipuan Transaksi Kartu Kredit dengan Algoritma Machine Learning. *Jurnal Teknologi Informatika dan Komputer*, 8(2), 386–401. <https://doi.org/10.37012/jtik.v8i2.1306>
- Nugroho, K. S., Akbar, I., Suksmawati, A. N., & Istiadi. (2023). *Deteksi Depresi dan Kecemasan Pengguna Twitter Menggunakan Bidirectional LSTM*. <https://doi.org/https://doi.org/10.48550/arXiv.2301.04521>
- Nurwanto, I., Tirtana, G. A., & Irkhami, A. L. (2025, Januari 2). *DLH Kota Jogja Sebut Aktivitas Kendaraan Wisatawan saat Libur Nataru Berpotensi Tingkatkan Pencemaran Udara, tetapi Masih Aman*. Radar Jogja. <https://radarjogja.jawapos.com/jogja/655483681/dlh-kota-jogja-sebut-aktivitas-kendaraan-wisatawan-saat-libur-nataru-berpotensi-tingkatkan-pencemaran-udara-tetapi-masih-aman>
- Ramadhan, A. Z. H., Rahayudi, B., & Ratnawati, D. E. (2024). Prediksi Polusi Udara Di Dki Jakarta Dengan Menggunakan Metode Long-short Term Memory (LSTM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK)*, 1(1), 2548–2964. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14122>
- Rayhan Rizal Mahendra, Fetty Tri Anggraeny, & Henni Endah Wahanani. (2024). Implementasi Item-Based Collaborative Filtering Untuk Rekomendasi Film. *Repeater: Publikasi Teknik Informatika dan Jaringan*, 2(3), 213–221. <https://doi.org/10.62951/repeater.v2i3.140>
- Riftianto, A. E., & Amirullah, A. (2024). Implementasi Visual Studio pada Sistem Monitoring Daya dan Proteksi Rele Arus Lebih Menggunakan Automatic Transfer Switch/Automatic Main Failure (ATS/AMF) Disuplai oleh Kombinasi Grid dan Photovoltaic (PV). *Rekayasa*, 17(1), 96–107. <https://doi.org/10.21107/rekayasa.v17i1.22151>



- Rondonuwu, N. T., Setiabudhi, D. O., & Gerungan, C. A. (2025). Pengaturan Penggunaan Kecerdasan Buatan Dalam Tugas Profesional Hakim di Indonesia. *Lex Privatum*, 15(2). <https://ejournal.unsrat.ac.id/index.php/lexprivatum/article/view/60756>
- Nova, S., Khotimah, N., & Wahyuningrum, M. Y. A. (2024). Pemanfaatan Chatbot Menggunakan Natural Language Processing Untuk Pembelajaran Dasar-Dasar Gui Tkinter Pada Bahasa Pemrograman Python. *Jurnal Ilmiah Teknik*, 3(1), 58–65. <https://doi.org/10.56127/juit.v3i1.1162>
- Syahrul, M., Syafwan, H., & Apridonol, Y. (2024). Prediksi Persediaan Oli Sepeda Motor Di Bengkel Amin Dengan Metode Simple Moving Average. *Fusion: Journal of Research in Engineering*, 1(2). <https://ejournal.faastrlibmedia.com/index.php/fusion/article/view/11>
- Yudiskara, I. M. N., Dwidasmara, I. B. G., & Widiartha, I. M. (2023). Prediksi Polusi Udara Kota Jakarta Menggunakan Recurrent Neural Network-gated Recurrent Units. *Jurnal Pengabdian Informatika*, 1(3). <https://ejournal1.unud.ac.id/index.php/jupita/article/view/369>