

Penambahan Parameter *PM2.5* dalam Prediksi Kualitas Udara : *Long Short Term Memory*

LISA KRISTIANA, DIAN MIYANTO

Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional Bandung
Email : lisa@itenas.ac.id

Received 05 Juli 2023 | *Revised* 28 Juli 2023 | *Accepted* 11 September 2023

ABSTRAK

Kualitas udara yang buruk menjadi masalah serius yang mempengaruhi kesehatan manusia dan lingkungan. Penyebabnya bervariasi, termasuk polusi udara, emisi industri, dan aktivitas manusia. Masalah yang dibahas adalah bagaimana menggunakan metode Long Short-Term Memory untuk memprediksi kualitas udara. Data kualitas udara yang mencakup parameter yang digunakan yaitu Partikel Kasar (PM10), Sulfur dioksida (SO), Karbon monoksida (CO), Ozon (O3) Nitrogen dioksida (NO2) serta penambahan Partikel Halus (PM2.5) yang belum terdapat pada penelitian sebelumnya. Penggunaan metode LSTM jenis khusus dari model Recurrent Neural Network (RNN), yang dirancang untuk mengatasi masalah "vanishing gradient" yang sering muncul dalam pelatihan RNN tradisional. Salah satu keunggulan LSTM adalah kemampuannya untuk menangkap dependensi jarak jauh dalam data berurutan, yang berarti dapat "mengingat" informasi dalam jangka waktu yang lama. Dengan melakukan evaluasi data pengujian menggunakan metrik evaluasi Root Mean Squared Error dengan hasil 4.0279.

Kata kunci: LSTM, RNN, Kualitas Udara, Prediksi, PM2.5

ABSTRACT

Poor air quality is a serious problem affecting human health and the environment. The causes are varied, including air pollution, industrial emissions, and human activities. The problem discussed is how to use the Long Short-Term Memory method to predict air quality. Air quality data which includes the parameters used are Coarse Particles (PM10), Sulfur dioxide (SO), Carbon monoxide (CO), Ozone (O3) Nitrogen dioxide (NO2) and the addition of Fine Particles (PM2.5) which have not been found in research previously. The use of the LSTM method is a special type of Recurrent Neural Network (RNN) model, which is designed to overcome the "vanishing gradient" problem that often arises in traditional RNN training. One advantage of LSTM is its ability to capture remote dependencies in sequential data, meaning it can "remember" information over long periods of time. By evaluating the test data using the evaluation metric Root Mean Squared Error with a result of 4.0279.

Keyword: LSTM, RNN, Air Quality, Prediction, PM2.5

1. PENDAHULUAN

Kualitas udara yang buruk memiliki dampak serius yang meluas terhadap kesehatan manusia, lingkungan, dan ekonomi **(Aini dkk., 2019)**. Polusi udara yang disebabkan oleh emisi dari industri, transportasi, dan kegiatan manusia lainnya dapat mengakibatkan peningkatan risiko penyakit pernapasan, gangguan kesehatan jangka panjang, dan kerusakan lingkungan. Konsentrasi partikel polutan pada kota Bandung yang ditetapkan oleh BMKG dengan nilai sebesar 112,3 ($\mu\text{g}/\text{m}^3$) melebihi ambang batas **(Khumaidi dkk., 2020)**.

Kualitas udara yang baik memiliki peran yang sangat penting bagi kesehatan manusia dan lingkungan. Peningkatan polusi udara di berbagai daerah perkotaan dan industri telah menjadi perhatian serius **(Akbar dkk., 2023)**. Prediksi kualitas udara dengan akurasi tinggi memiliki manfaat besar dalam mengambil tindakan pencegahan dan mengurangi dampak buruk terhadap kesehatan manusia dan lingkungan. Di kota besar yang menjadi salah satu masalah utamanya yaitu kualitas udara **(Oktaviani & Hustinawati, 2021)**.

Recurrent Neural Network (RNN) adalah jenis arsitektur pendahulu dari *LSTM* sehingga Penggunaan algoritma *LSTM* dalam prediksi kualitas udara memiliki beberapa keunggulan **(Aprianto dkk., 2018)**. Pertama, *LSTM* dapat menangkap dependensi jangka panjang dalam data, yang sering kali penting dalam analisis deret waktu seperti data kualitas udara. Kedua, *LSTM* mampu mengatasi masalah *vanishing gradient* yang sering terjadi dalam jaringan saraf tiruan tradisional, sehingga dapat menghasilkan prediksi yang lebih akurat **(Oktaviani & Hustinawati, 2021)**.

Algoritma *Long Short Term Memory* adalah salah satu jenis algoritma dalam bidang kecerdasan buatan yang digunakan untuk memodelkan dan memprediksi data sekuensial, termasuk data kualitas udara **(Badriyah dkk., 2022)**. *LSTM* dikembangkan khusus untuk mengatasi masalah memori jangka panjang dan ketergantungan jangka panjang dalam data sekuensial **(Faishol dkk., 2020)**. Untuk mengatasi masalah ini, arsitektur *LSTM* dikembangkan. *LSTM* menggunakan unit memori khusus yang disebut "*memory cell*" untuk menyimpan informasi dalam jangka panjang. Unit memori ini memungkinkan *LSTM* untuk mengontrol aliran informasi dan menghindari masalah *gradient* yang hilang **(Arkadia dkk., 2022)**.

Karbon merupakan komponen penting dari partikel di daerah perkotaan. Sumber utama polutan ini adalah pembakaran bahan bakar fosil dan biomassa **(Karyadi, 2022)**. *PM2.5* adalah rumah karbon yang dihasilkan dalam proses pembakaran. *PM2.5* adalah partikel yang telah lama berada di atmosfer dan dapat menembus jauh ke dalam sistem pernapasan manusia **(Qori dkk., 2022)**. Maka dari itu, analisa polutan *PM2.5* sangat berguna dalam penilaian kualitas udara. Salah satu bahan penyebab pencemaran udara yang sangat berbahaya bagi kesehatan adalah *partikulat matter 2.5 (PM2.5)* **(Putri dkk., 2020)**.

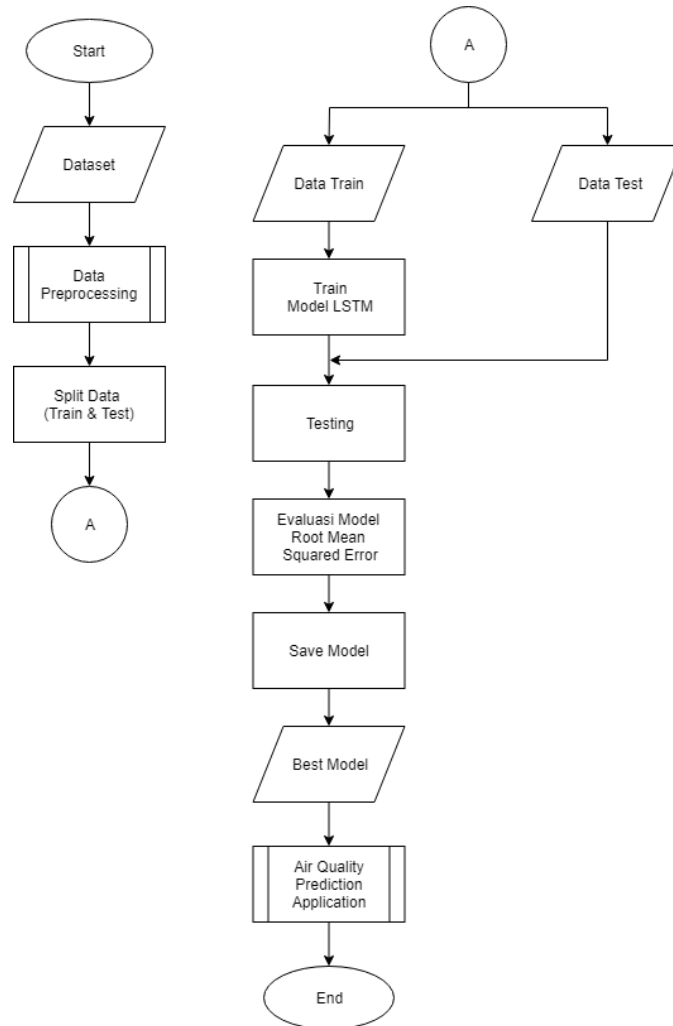
Dalam penelitian **(Hidayatullah dkk., 2022)** tidak menggunakan parameter Partikel Halus (*PM2.5*). Berdasarkan data *AQMS* konsentrasi *PM2.5* di Kota Bandung mencapai angka 127 *AQI* dengan status *unhealthy for sensitive group* (101-150) Sebenarnya parameter tersebut memiliki nilai yang cukup dominan dan menjadi nilai kritis pada *dataset* kualitas udara.

Berdasarkan uraian diatas, maka akan dilakukan penelitian menggunakan metode *Long Short Term Memory* dengan tujuan dapat memprediksi Kualitas Udara.

2. METODE

2.1 Diagram Alir Utama

Pada *flowchart* utama terlihat bahwa dataset kualitas udara akan digunakan. Alur tersebut dapat diamati pada Gambar 1 sebagai berikut.



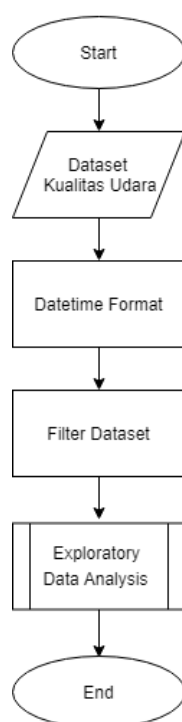
Gambar 1. Diagram Alir Utama

Melakukan pemanggilan dataset kualitas udara berbentuk *file.csv* dari *Google Drive*. Dilanjutkan tahap pra-pemrosesan data, mula- mula Dataset akan di analisis dengan menggunakan penggambaran grafik dan pemeriksaan data menggunakan beberapa teknik visualisasi data dan juga *Exploratory Data Analysis (EDA)*. Data – data yang telah melalui tahapan pra-pemrosesan data tersebut kemudian akan dibagi menjadi 2 buah bagian, dimana data bagian pertama akan digunakan sebagai data latih untuk pelatihan model (*Model Training*), dan data pada bagian kedua akan digunakan sebagai data uji yang akan digunakan untuk menguji model yang telah dilatih sebelumnya (*Model Testing*). Setelah dilakukan pembagian data, langkah selanjutnya yaitu dengan menggunakan data latih, dilakukan pelatihan model dengan menggunakan model *Long Short Term Memory*. Model tersebut akan

di uji menggunakan data uji sebelum kemudian nantinya akan dilakukan evaluasi. Hasil evaluasi akan dibandingkan satu sama lain untuk mengetahui hasil evaluasi mana yang menghasilkan model dengan performansi paling optimal berdasarkan dari hasil eksperimen dan analisisnya. Model yang menghasilkan nilai *Error Metrics* terendah dari hasil evaluasi akan disimpan modelnya. Kemudian nantinya digunakan dalam pembuatan aplikasi berbasis *website* pada kasus prediksi kualitas udara dikota Bandung.

2.2 Pra-pemrosesan Data

Pada tahap *pre-processing*, dataset terlebih dahulu dianalisis dengan penggambaran grafik dan pengecekan data menggunakan beberapa teknik visualisasi data dan juga *Exploratory Data Analysis* (EDA). Diagram alir Data *Pre-processing* dapat dilihat pada Gambar 2. Setelah melakukan semua tahapan pra-pemrosesan data, akhirnya data yang siap digunakan untuk pemodelan *Machine Learning* menggunakan model *Long Short Term Memory*.



Gambar 2. Diagram Alir Pra-pemrosesan Data

Pada Gambar 2 diatas maka, dapat dijelaskan bahwa Dataset Kualitas Udara akan digunakan pada sistem penelitian Implementasi Algoritma *Long Short Term Memory* untuk Prediksi Terhadap Kualitas Udara. *Dataset* akan diatur terlebih dahulu format tanggal waktunya setelah itu lakukan *filter* dataset untuk menentukan data lokasi yang dibutuhkan. Kemudian dianalisis dengan dilakukannya penggambaran grafik dan pemeriksaan data menggunakan beberapa teknik visualisasi data dan juga *Exploratory Data Analysis* (EDA).

2.2.1 Proses *Exploratory Data Analysis* (EDA)

Pada tahap *Exploratory Data Analysis* dirancang diagram alir atau *flowchart* dari sistem Implementasi Algoritma *Long Short Term Memory* untuk Prediksi Terhadap Kualitas Udara. Gambar 3. dibawah menunjukkan diagram alirnya.

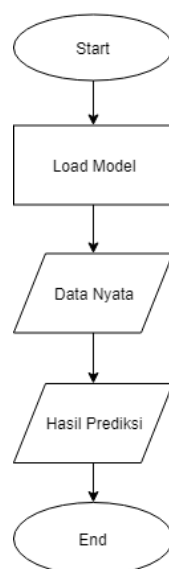


Gambar 3. Diagram Alir EDA

Berdasarkan Gambar 3. tersebut, maka dapat dijelaskan bahwa melakukan pemanggilan dataset Kualitas Udara akan digunakan untuk Prediksi Terhadap Kualitas Udara. Menampilkan Informasi Dasar Data Untuk melihat ringkasan singkat dari *dataset*, dengan menggunakan fungsi *info()* dan *describe()*. Menampilkan Visualisasi Data dengan menggunakan matplotlib dan seaborn untuk melakukan visualisasi data. Contohnya, membuat *histogram*, *boxplot*, dan *scatter plot* untuk melihat distribusi data dan hubungan antar variabel. Melakukan pengecekan Korelasi antar kolom dalam *Heatmap* dengan menggunakan fungsi *corr()* dan ini membantu dalam melihat hubungan atau ketergantungan antar variabel untuk melihat korelasi antar variabel numerik. Cek *Missing Values* dengan menggunakan fungsi *isnull()* dan *sum()* untuk mengetahui jumlah *missing values* dalam data setelah itu dilakukan Pembersihan Data dengan melakukan proses penghapusan *missing values*. Melakukan Standarisasi data dengan mengubah semua variabel menjadi skala yang sama, biasanya kisaran *mean* (rata-rata) 0 dan *standard deviation* (standar deviasi) 1.

2.3 Aplikasi Prediksi Kualitas Udara

Setelah menyelesaikan kinerja model *Long Short Term Memory* berdasarkan eksperimen dan analisis, model optimal yang dihasilkan akan disimpan untuk kemudian digunakan dalam pembuatan aplikasi berbasis *web* untuk memprediksi kualitas udara di Bandung. Alur aplikasi akan terlihat seperti diagram alir yang ditunjukkan pada Gambar 4.

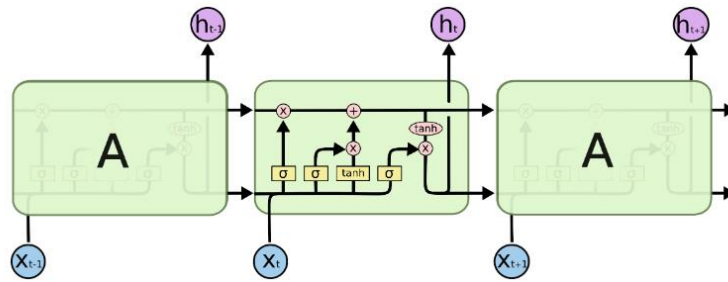


Gambar 4. Diagram Alir Aplikasi

Berdasarkan Gambar 4. diatas, maka dapat dijelaskan mengenai konsep secara umum dari alur aplikasi prediksi kualitas udara, mula – mula aplikasi akan melakukan pemanggilan model *LSTM* paling optimal yang telah di dapatkan sebelumnya, pengguna aplikasi atau biasa disebut *User* kemudian dapat memasukkan data – data yang diperlukan untuk memprediksi kualitas udara yang ingin diketahui harganya, kemudian berdasarkan data – data yang dimasukkan oleh *User*, keluaran (*Output*) aplikasi nantinya akan berupa kualitas udara dari hasil prediksinya.

2.4 Long Short Term Memory

LSTM adalah salah satu jenis arsitektur jaringan *RNN* yang dirancang khusus untuk mengatasi kendala dalam pemahaman dan pemeliharaan informasi dalam jangka panjang (**Supriyadi, 2021**). *Node RNN* dalam *hidden layer* secara tradisional digunakan untuk memodelkan hubungan temporal dalam data. Namun, *node RNN* memiliki keterbatasan dalam mengingat informasi jangka panjang dan mengatasi masalah *vanishing gradient* (**Wildan Putra Aldi & Aditsania, 2018**). Hal ini dapat menyebabkan performa yang buruk dalam memprediksi data deret waktu yang kompleks. *LSTM cells* dirancang untuk mengatasi keterbatasan tersebut. *LSTM* menggunakan struktur yang lebih kompleks dengan adanya "*gates*" (gerbang) yang memungkinkan aliran informasi yang selektif dalam sel *LSTM* (**Writers, 2018**). *LSTM* terdiri dari tiga gates utama: *Input Gate*, *Forget Gate*, dan *Output Gate* seperti pada Gambar 5. *Gates-gates* ini bertanggung jawab untuk mengontrol aliran informasi dalam sel *LSTM* dan menentukan apakah informasi harus disimpan atau dilupakan. Berikut adalah Persamaan (1) sampai dengan Persamaan (6) yang digunakan dalam *LSTM*:



Gambar 5. Long Short-Term Memory (Gallan, 2022)

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

Pada Persamaan (1), *Forget gate* menggunakan fungsi aktivasi *sigmoid* untuk menghasilkan vektor (*forget vector*) yang bernilai antara 0 dan 1. Nilai 0 menunjukkan bahwa informasi sebelumnya dalam sel memori harus dilupakan sepenuhnya, sementara nilai 1 menunjukkan bahwa informasi tersebut harus dipertahankan tanpa perubahan.

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

Pada Persamaan (2), *Input gate* menerima masukan saat ini dan masukan sebelumnya, kemudian menghasilkan vektor (*update vector*) yang bernilai antara 0 dan 1. Nilai 0 menunjukkan bahwa elemen data masukan tidak akan diperbarui atau diabaikan, sedangkan nilai 1 menunjukkan bahwa elemen data masukan akan sepenuhnya diperbarui dan disimpan dalam sel memori.

$$\tilde{c}_t = \tanh(u_c x_t + W_c h_{t-1} + b_c) \quad (3)$$

Pada Persamaan (3), Setelah melewati *input gate*, *LSTM* menggunakan fungsi aktivasi *tanh* untuk menghasilkan vektor nilai baru yang akan disimpan dalam sel memori (*memory cell*). Fungsi aktivasi *tanh* (*tangent hyperbolic*) memetakan nilai *input* ke dalam rentang -1 hingga 1. Fungsi ini membantu dalam mengontrol aliran informasi dalam sel memori dengan menghilangkan skala dari nilai yang dipertimbangkan.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Pada Persamaan (4), setelah nilai baru dihasilkan melalui kombinasi dari *forget gate* dan *input gate*, proses selanjutnya adalah memperbarui nilai pada *memory cell* sebelumnya dengan nilai baru tersebut. Untuk melakukan hal ini, *LSTM* menggunakan operasi perkalian *elemen-wise* (*element-wise multiplication*) antara *output* dari *forget gate* dan *output* dari *input gate*. Operasi ini menggabungkan informasi dari kedua *gates* tersebut untuk memperbarui nilai pada *memory cell*.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

Pada Persamaan (5), *Output gate* dalam *LSTM* menerima *input* saat ini dan *state* sebelumnya, kemudian menggunakan fungsi aktivasi *sigmoid* untuk menghasilkan vektor (*output vector*) yang bernilai antara 0 dan 1. Nilai 0 menunjukkan bahwa informasi dalam *memory cell* tidak akan dikeluarkan atau diabaikan, sedangkan nilai 1 menunjukkan bahwa informasi tersebut akan sepenuhnya dikeluarkan.

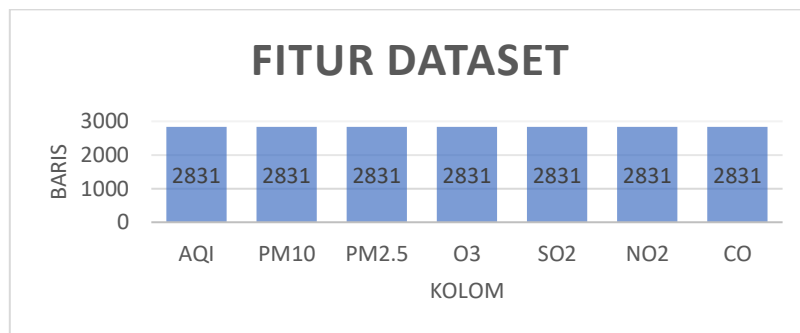
$$h_t = o_t * \tanh(C_t) \tag{6}$$

Pada Persamaan (6), Dengan mengalikan *Output gate* dengan nilai memory *Cell state* yang baru, kita mendapatkan hasil akhir yang akan dikeluarkan dari *LSTM*.

3. HASIL DAN PEMBAHASAN

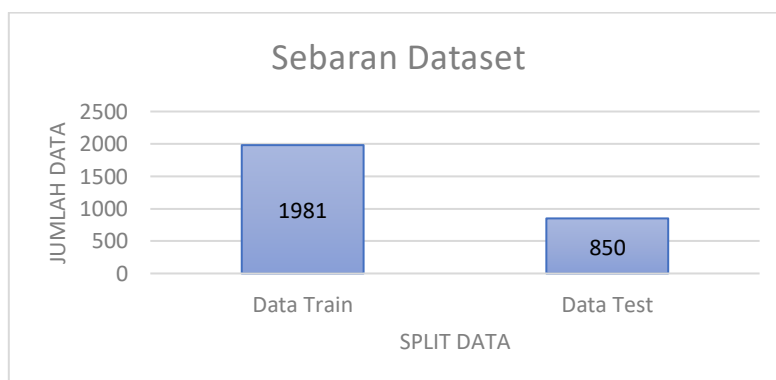
3.1 Kumpulan data

Dataset yang digunakan dalam penelitian ini berisi tentang kualitas udara di Bandung. Semua data kualitas udara dalam dataset dikumpulkan dan diperoleh dari 16 Januari 2022 pukul 17.00 hingga tanggal 16 Mei 2022 pukul 17.00, Dataset yang akan digunakan dalam penelitian ini dapat diambil dari situs resmi Github.



Gambar 6. Jumlah Data pada Setiap Fitur

Berdasarkan Gambar 6. diatas, Jumlah data yang ada yaitu sebesar 2831 data dan memiliki 7 fitur di dalamnya seperti *Air Quality Index (AQI)*, Partikel Kasar (PM10), Partikel Halus (PM2.5), Sulfur dioksida (SO), Karbon Monoksida (CO), Ozon (O3) dan Nitrogen Dioksida (NO2).

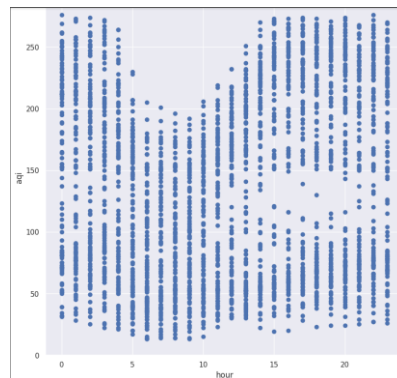


Gambar 7. Sebaran Dataset

Berdasarkan Gambar 7. diatas, Sesuai dengan rencana penelitian, dataset dibagi menjadi 2 bagian yaitu data latih sebanyak 70% (1981 data) dan data uji sebanyak 30% (850 data). Distribusi data ini sangat penting dilakukan untuk mencegah terjadinya *overfitting* pada model *Machine Learning*.

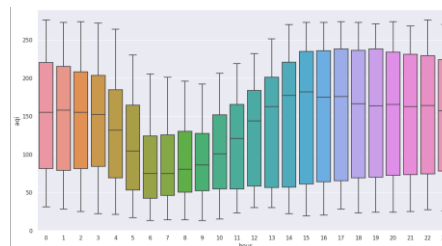
3.2 Visualisasi Data dan *Analisis Data Eksplorasi (EDA)*

Pada tahap visualisasi dan *Exploratory data analysis*, untuk memahami dan menganalisis data dengan menggunakan teknik visualisasi grafis dan metode statistik. Tujuannya adalah untuk mendapatkan pemahaman yang lebih dalam tentang data dan mendapatkan pengetahuan untuk analisis lanjutan. Visualisasi menggunakan *scatter*, *boxplot*, *heatmap* pada fitur dataset dapat lakukan pengecekan visualisasinya dan untuk selanjutnya *Exploratory data analysis*.



Gambar 8. Grafik *Scatter Plot*

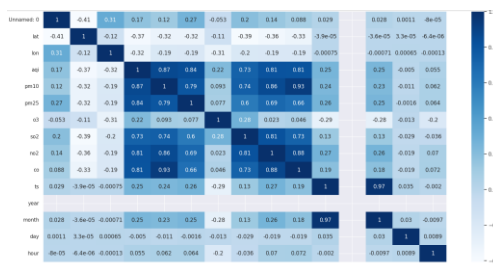
Pada Gambar 8. Diatas grafik *scatter plot* bahwa setiap titik merupakan nilai dari variabel indeks kualitas udara dan variabel waktu dalam jam. Titik polusi tertinggi terjadi pada nilai indeks kualitas udara diangka 250 lebih dan pada jam 15 lebih sore hari. Pada jam tersebut biasanya terjadi lonjakan kendaraan pada jalanan.



Gambar 9. Grafik *Box Plot*

Pada Gambar 9. grafik *box plot* bahwa sumbu y merupakan nilai indeks kualitas udara dan sumbu x merupakan waktu pada hitungan jam dalam sehari. Terlihat bahwa polusi terjadi pada saat jam pulang kerja jam 15 sampai jam 19. Aktivitas rumah tangga dan pabrik merupakan penyebab polusi udara.

Penambahan Parameter PM2.5 dalam Prediksi Kualitas Udara : Long Short Term Memory



Gambar 10. Grafik Heat Map

Pada Gambar 10. grafik *heatmap* representasi grafis dari data di mana nilai individual direpresentasikan sebagai warna, menunjukkan seberapa kuat hubungan antar variabel, Garis biru yang membentang secara diagonal menunjukkan nilai yang berkorelasi antar variabel.

3.3 Pembersihan Data

Berdasarkan hasil visualisasi data dan juga *Exploratory data analysis* (EDA), langkah selanjutnya adalah melakukan *data cleansing* terlebih dahulu sebelum data siap digunakan dalam pemodelan *LSTM*. Langkah pertama untuk membersihkan data dilakukan dengan menghilangkan beberapa fitur yang tidak akan digunakan untuk proses pemodelan, fitur tersebut adalah Kota, Garis Lintang, dan Garis Bujur. Kemudian, melakukan pengecekan data terlebih dahulu apakah ada data yang bersifat *NULL* dan ternyata terdapat 53 data bersifat *NULL*. Maka dari itu dilakukan perintah *DROP* untuk menghapus data yang kosong. Tahap ini dilakukan untuk memastikan data yang bersih, valid, dan siap untuk digunakan dalam analisis, pemodelan, dan visualisasi yang melibatkan pengolahan data untuk proses selanjutnya.

3.4 Skenario Eksperimen

Sebelum dilakukan pemodelan *lstm* pada penelitian ini, terdapat beberapa percobaan yang dilakukan pada masing-masing parameter, skenario eksperimen model adalah sebagai berikut.

3.4.1 Eksperimen Model dengan Epoch 10

Pada eksperimen model yang dilakukan untuk menentukan *Epoch* yang paling optimal dalam memperoleh nilai matrik evaluasi *RMSE* yang paling optimal pada model. *Epoch* yang digunakan dalam eksperimen model ini dengan *Epoch* 10 menggunakan parameter pendukung Lapisan *LSTM*, Lapisan *Dense*, *Learning Rate* dan *Batch Size* (2, 4, 8, 16, 32 dan 64). Perhatikan pada Tabel 1. berikut dibawah ini.

Tabel 1. Eksperimen Epoch 10

Lapisan LSTM	Lapisan Dense	Learning Rate	Batch Size	Epoch	RMSE
64	25	0.001	2	10	14,34655638
			4		15,35948685
			8		15,44080227
			16		16,1432101
			32		16,87678313
			64		21,0622662

Berdasarkan pada Tabel 1. diatas terlihat bahwa percobaan eksperimen model dengan *Epoch* yang menghasilkan nilai *error* paling rendah pada *Batch Size* 2 dengan memperoleh nilai matrik Evaluasi *RMSE* sebesar 14,34655638.

3.4.2 Eksperimen Model dengan *Epoch* 50

Pada eksperimen model yang dilakukan untuk menentukan *Epoch* yang paling optimal dalam memperoleh nilai matrik evaluasi *RMSE* yang paling optimal pada model. *Epoch* yang digunakan dalam eksperimen model ini dengan *Epoch* 50 menggunakan parameter pendukung Lapisan *LSTM*, Lapisan *Dense*, *Learning Rate* dan *Batch Size* (2, 4, 8, 16, 32 dan 64). Perhatikan pada Tabel 2. berikut dibawah ini.

Tabel 2. Eksperimen *Epoch* 50

Lapisan <i>LSTM</i>	Lapisan <i>Dense</i>	<i>Learning Rate</i>	<i>Batch Size</i>	<i>Epoch</i>	<i>RMSE</i>
64	25	0.001	2	50	6,598163021
			4		8,336598259
			8		10,81224971
			16		10,81724516
			32		14,00884157
			64		13,94307566

Berdasarkan pada Tabel 2. diatas terlihat bahwa percobaan eksperimen model dengan *Epoch* yang menghasilkan nilai *error* paling rendah pada *Batch Size* 2 dengan memperoleh nilai matrik Evaluasi *RMSE* sebesar 6,598163021.

3.4.3 Eksperimen Model dengan *Epoch* 100

Pada eksperimen model yang dilakukan untuk menentukan *Epoch* yang paling optimal dalam memperoleh nilai matrik evaluasi *RMSE* yang paling optimal pada model. *Epoch* yang digunakan dalam eksperimen model ini dengan *Epoch* 100 menggunakan parameter pendukung Lapisan *LSTM*, Lapisan *Dense*, *Learning Rate* dan *Batch Size* (2, 4, 8, 16, 32 dan 64). Perhatikan pada Tabel 3. berikut dibawah ini.

Tabel 3. Eksperimen *Epoch* 100

Lapisan <i>LSTM</i>	Lapisan <i>Dense</i>	<i>Learning Rate</i>	<i>Batch Size</i>	<i>Epoch</i>	<i>RMSE</i>
64	25	0.001	2	100	9,828107965
			4		6,34880394
			8		6,862513213
			16		7,765258498
			32		10,51175983
			64		11,73561536

Berdasarkan pada Tabel 3. diatas terlihat bahwa percobaan eksperimen model dengan *Epoch* yang menghasilkan nilai *error* paling rendah pada *Batch Size* 4 dengan memperoleh nilai matrik Evaluasi *RMSE* sebesar 6,34880394.

3.4.4 Eksperimen Model dengan *Epoch* 500

Pada eksperimen model yang dilakukan untuk menentukan *Epoch* yang paling optimal dalam memperoleh nilai matrik evaluasi *RMSE* yang paling optimal pada model. *Epoch* yang digunakan dalam eksperimen model ini dengan *Epoch* 500 menggunakan parameter pendukung Lapisan *LSTM*, Lapisan *Dense*, *Learning Rate* dan *Batch Size* (2, 4, 8, 16, 32 dan 64). Perhatikan pada Tabel 4. berikut dibawah ini.

Tabel 4. Eksperimen *Epoch* 500

Lapisan <i>LSTM</i>	Lapisan <i>Dense</i>	<i>Learning Rate</i>	<i>Batch Size</i>	<i>Epoch</i>	<i>RMSE</i>
64	25	0.001	2	500	4,442917236
			4		5,076252089
			8		5,475965195
			16		4,027943957
			32		4,325119302
			64		5,68862202

Berdasarkan pada Tabel 4. diatas terlihat bahwa percobaan eksperimen model dengan *Epoch* yang menghasikan nilai *error* paling rendah pada *Batch Size* 16 dengan memperoleh nilai matrik Evaluasi *RMSE* sebesar 4.0279439569756885.

3.4.5 Eksperimen Model dengan *Epoch* 1000

Pada eksperimen model yang dilakukan untuk menentukan *Epoch* yang paling optimal dalam memperoleh nilai matrik evaluasi *RMSE* yang paling optimal pada model. *Epoch* yang digunakan dalam eksperimen model ini dengan *Epoch* 1000 menggunakan parameter pendukung Lapisan *LSTM*, Lapisan *Dense*, *Learning Rate* dan *Batch Size* (2, 4, 8, 16, 32 dan 64). Perhatikan pada Tabel 5. berikut dibawah ini.

Tabel 5. Eksperimen *Epoch* 1000

Lapisan <i>LSTM</i>	Lapisan <i>Dense</i>	<i>Learning Rate</i>	<i>Batch Size</i>	<i>Epoch</i>	<i>RMSE</i>
64	25	0.001	2	1000	5,038720335
			4		4,671155029
			8		5,283535609
			16		6,500291664
			32		5,515091128
			64		5,700240847

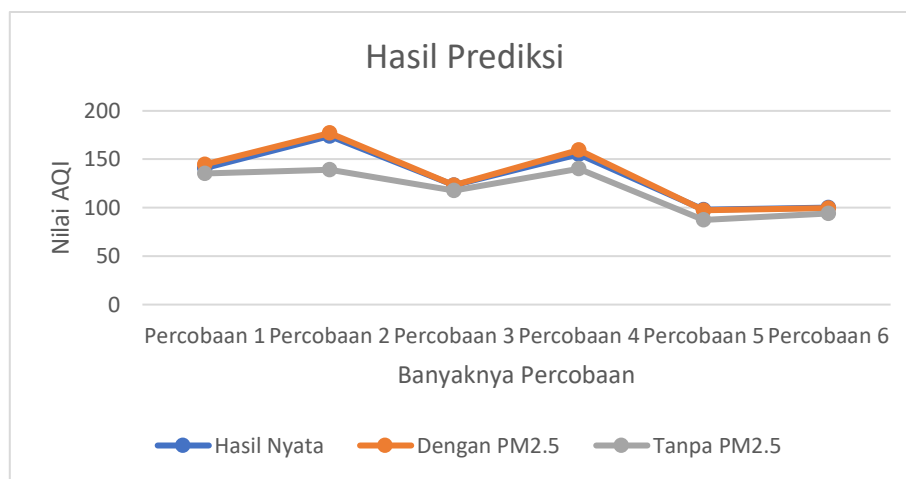
Berdasarkan pada Tabel 5. diatas terlihat bahwa percobaan eksperimen model dengan *Epoch* yang menghasikan nilai *error* paling rendah pada *Batch Size* 4 dengan memperoleh nilai matrik Evaluasi *RMSE* sebesar 4,671155029.

Kedua *hyperparameter* ini saling terkait dan dapat mempengaruhi kinerja dan keberhasilan model *Long Short Term Memory*. Adapun analisa yang didapat dari eksperimen model yaitu *Batch size* yang lebih kecil dapat memberikan gradien yang lebih akurat, tetapi membutuhkan

waktu pelatihan yang lebih lama. *Batch size* yang lebih besar dapat meningkatkan kecepatan pelatihan, tetapi mungkin menyebabkan model kurang adaptif terhadap perubahan dalam data. Jumlah *epoch* yang terlalu sedikit mungkin tidak memberikan model cukup kesempatan untuk mempelajari pola yang rumit dalam data. Jumlah *epoch* yang terlalu banyak dapat menyebabkan *overfitting* pada data pelatihan.

3.7 Model dan Implementasi

Dari hasil implementasi eksperimen, maka model *Long Short Term Memory* dengan *Batch Size* 16, dan *Epoch* 500. yang menghasilkan performansi paling optimal menggunakan beberapa data dari *dataset* kualitas udara. Serta juga digunakan data – data yang diambil dari dunia nyata yang di dapat dari beberapa website situs resmi prediksi kualitas udara. Validasi tersebut dilakukan pada aplikasi berbasis website yang telah dirancang dan dibuat sebelumnya.



Gambar 11. Hasil Prediksi Kualitas Udara

Parameter *PM2.5* (*Particulate Matter 2.5*) memiliki pengaruh signifikan dalam kualitas udara. *PM2.5* merujuk pada partikel-partikel kecil dengan diameter kurang dari 2,5 mikrometer yang terdapat dalam udara. Partikel-partikel ini dapat berasal dari berbagai sumber, termasuk aktivitas pembakaran (seperti kendaraan bermotor, industri, dan pembakaran biomassa), debu, polusi industri, dan reaksi kimia di *atmosfer*. *PM2.5* tersedia secara luas dan dianggap sebagai polutan udara paling berbahaya yang berdampak pada kesehatan manusia.

4. KESIMPULAN

Prediksi kualitas udara menggunakan *dataset* kualitas udara di Bandung. Dilakukan dengan menggunakan metode *Long Short Term Memory*. Dengan melakukan eksperimen model dengan *Batch Size* 16 & *Epoch* 500 memiliki performa yang paling optimal dibandingkan dengan eksperimen lainnya. Dengan melakukan evaluasi data pengujian menggunakan matrik evaluasi *Root Mean Squared Error* dengan hasil 4.0279.

Pengujian model menggunakan metode *Long Short Term Memory* dengan *Batch Size* 16 & *Epoch* 500 membuktikan bahwa model yang dihasilkannya ternyata tidak memiliki perilaku *Overfitting*. Hal tersebut dapat dilihat pada saat pelatihan model menggunakan data latih dan

pengujian model dengan data uji, model yang dihasilkannya memiliki nilai matriks *error* terkecil dibandingkan dengan beberapa eksperimen lainnya pada penelitian ini.

Aplikasi prediksi kualitas udara berbasis website yang telah dibuat, dapat digunakan untuk melakukan prediksi kualitas udara. Pada Validasi Hasil Akhir Implementasi Model pada aplikasi prediksi kualitas udara berbasis website jika menggunakan parameter PM2.5 akan menghasilkan nilai prediksi yang mendekati nilai AQI pada dataset kualitas udara dan jika tanpa parameter tersebut menghasilkan nilai prediksi yang jauh dari nilai AQI pada dataset kualitas udara.

DAFTAR RUJUKAN

- Aini, N., Ruktiari, R., Pratama, M. R., & Buana, A. F. (2019). Sistem Prediksi Tingkat Pencemaran Polusi Udara dengan Algoritma Naïve Bayes di Kota Makassar. In *Prosiding Seminar Nasional Komunikasi dan Informatika* (Vol. 3).
- Akbar, R., Santoso, R., & Warsito, B. (2023). Prediksi Tingkat Temperatur Kota Semarang Menggunakan Metode Long Short-Term Memory (Lstm). *Jurnal Gaussian*, *11*(4), 572–579. <https://doi.org/10.14710/j.gauss.11.4.572-579>
- Aprianto, Y., Nurhasanah, N., & Sanubary, I. (2018). Prediksi Kadar Particulate Matter (PM10) untuk Pemantauan Kualitas Udara Menggunakan Jaringan Syaraf Tiruan Studi Kasus Kota Pontianak. *Positron*, *8*(1), 15. <https://doi.org/10.26418/positron.v8i1.25470>
- Arkadia, A., Hananto, B., Prasvita, D. S., & Lstm, K. K. (2022). *Optimasi Long Short Term Memory Dengan Adam Menggunakan Data Udara Kota DKI Jakarta*.
- Badriyah, J., Fariza, A., & Harsono, T. (2022). Prediksi Curah Hujan Menggunakan Long Short Term Memory. *Jurnal Media Informatika Budidarma*, *6*(3), 1297. <https://doi.org/10.30865/mib.v6i3.4008>
- Faishol, M. A., Endroyono, E., & Irfansyah, A. N. (2020). Predict Urban Air Pollution in Surabaya Using Recurrent Neural Network – Long Short Term Memory. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *18*(2), 102. <https://doi.org/10.12962/j24068535.v18i2.a988>
- Gallan, V. (2022). *LSTM (Long Short Term Memory)*. Bina Nusantara IT Division. <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>
- Hermawan, A., & Sela, E. I. (2019). SPKU: Sistem Prediksi Kualitas Udara (Studi Kasus: Dki Jakarta). In *Program Studi Teknik Informatika, Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta*. <http://eprints.uty.ac.id/3552/>
- Hidayatullah, B. K., Kallista, M., & Setianingsih, C. (2022). Prediksi Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory Berbasis Web (Studi Kasus Pada Kota Jakarta). In *eProceedings of Engineering* (Vol. 9, Issue 3). <https://data.jakarta.go.id/>
- Karyadi, Y. (2022). Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(1), 671–684. <https://doi.org/10.35957/jatisi.v9i1.1588>

- Khumaidi, A., Raafi'udin, R., & Solihin, I. P. (2020). Pengujian Algoritma Long Short-Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung. *Jurnal Telematika*, 15(1), 13–18.
- Oktaviani, A., & Hustinawati. (2021). Prediksi Rata-Rata Zat Berbahaya Di Dki Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory. *Jurnal Ilmiah Informatika Komputer*, 26(1), 41–55. <https://doi.org/10.35760/ik.2021.v26i1.3702>
- Putri, S. E., Kusuma, D. A., & Ruchjana, B. N. (2020). Penerapan Metode Universal Kriging (Uk) Untuk Prediksi Konsentrasi Particulate Matter 2.5 (Pm2.5) Di Kota Bandung. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 14(2), 285–296. <https://doi.org/10.30598/barekengvol14iss2pp285-296>
- Qori, P. A., Oktafani, D. S., & Kharisudin, I. (2022). Analisis Peramalan dengan Long Short Term Memory pada Data Kasus Covid-19 di Provinsi Jawa Tengah. *PRISMA, Prosiding Seminar Nasional Matematika*, 5, 752–758.
- Supriyadi, E. (2021). Prediksi Parameter Cuaca Menggunakan Deep Learning Long-Short Term Memory (Lstm). *Jurnal Meteorologi Dan Geofisika*, 21(2), 55. <https://doi.org/10.31172/jmg.v21i2.619>
- Wildan Putra Aldi, M., & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *E-Proceeding of Engineering*, Vol. 5, 3548–3555.
- Writers, Iqa. S. (2018). *Polutan udara diukur dalam AQI*. Iqair.Com. <https://www.iqair.com/id/newsroom/what-is-aqi>