SD211123 – ANALISIS DATA EKSPLORATIF

ANALISIS FAKTOR-FAKTOR YANG MEMENGARUHI KUALITAS UDARA PERKOTAAN SERTA EVALUASI MODEL PREDIKSI BERBASIS KORELASI, REGRESI DATA PANEL, DAN DERET WAKTU

KELOMPOK JOSJIS :
Desi Nofitasari – 24083010058
Diva Anggraeni – 24083010065
Siti Rania Azaria – 24083010072
Achmad Dany Gunawan – 24083010075

DOSEN PENGAMPU
Amri Muhaimin, S.Stat. M. Stat, M.S
Shindi Shella May Wara, M.Stat.

KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI SAINS DATA
SURABAYA

2025

ABSTRAK

Nama Mahasiswa / NPM	:	Desi Nofitasari / 24083010058 Diva Anggraeni / 24083010065 Siti Rania Azaria / 24083010072 Achmad Dany Gunawan / 24083010075
Judul Laporan	:	Analisis Faktor-Faktor yang Memengaruhi Kualitas Udara Perkotaan Serta Evaluasi Model Prediksi Berbasis Korelasi, Regresi Data Panel, dan Deret Waktu
Dosen Pengampu	:	1. Amri Muhaimin, S.Stat. M.Stat, M.S 2. Shindi Shella May Wara, M.Stat.

Penelitian ini bertujuan untuk menganalisis faktor-faktor yang memengaruhi Air Quality Index (AQI) serta membandingkan performa model prediksi berbasis pendekatan korelasional, deret waktu, dan machine learning menggunakan data UrbanAirNet tahun 2020–2023 dengan jumlah 175205 data. Hasil analisis korelasi Spearman menunjukkan bahwa PM2.5 memiliki keterkaitan paling kuat dengan AQI (0,93), sedangkan polutan gas serta variabel meteorologi menunjukkan korelasi sangat lemah. Model ARIMA(0,0,0) menunjukkan galat tinggi (RMSE 0,914) dan gagal menangkap fluktuasi data AR(1)-GARCH(1,1) hanya memberikan peningkatan marginal (RMSE 0,912) dan masih belum merepresentasikan volatilitas AQI dengan baik Tahap prediksi diperkuat menggunakan metode machine learning, dimana GRU menjadi model terbaik (RMSE 0,13353) diikuti LSTM dan CNN, sementara SVR menunjukkan galat tertinggi. Selanjutnya, regresi data panel menghasilkan model fixed effect dengan nilai R-squared (0,9209), menunjukkan bahwa 92% variasi AQI dapat dijelaskan oleh variabel Humidity, PM2.5, O_3, Pressure, dan CO. Uji t menegaskan bahwa PM2.5 dan O_3 berpengaruh signifikan terhadap AQI, sedangkan variabel meteorologi dan CO tidak signifikan. Secara keseluruhan, ketiga pendekatan ini menegaskan bahwa PM2.5 merupakan faktor dominan yang memengaruhi AQI, model machine learning khususnya GRU memberikan prediksi paling akurat, sementara regresi panel memberikan kejelasan kontribusi setiap variabel terhadap perubahan AQI.

Kata kunci : AQI(*Air Quality Index*), ARIMA, *GRU(Gated Recurrent Unit)*, Machine Learning, Regresi Data Panel.

ABSTRACT

Student Name / NPM	:	Desi Nofitasari / 24083010058 Diva Anggraeni / 24083010065 Siti Rania Azaria / 24083010072 Achmad Dany Gunawan / 24083010075
Project Title	:	Analysis of Factors Affecting Urban Air Quality and Evaluation of Prediction Models Based on Correlation, Panel Data Regression, and Time Series
Advisor	:	1. Amri Muhaimin, S.Stat. M.Stat, M.S 2. Shindi Shella May Wara, M.Stat.

This study aims to analyze the factors influencing the Air Quality Index (AQI) and to compare the performance of predictive models based on correlational, time-series, and machine-learning approaches using the UrbanAirNet dataset for 2020–2023, consisting of 175,205 observations. The Spearman correlation results indicate that PM2.5 has the strongest association with AQI (0.93), while gaseous pollutants and meteorological variables exhibit very weak correlations. The ARIMA(0,0,0) model produces a high error (RMSE 0.914) and fails to capture data fluctuations, whereas the AR(1)-GARCH(1,1) model provides only marginal improvement (RMSE 0.912) and still does not represent AQI volatility adequately. The prediction stage is further enhanced using machine-learning methods, with GRU achieving the best performance (RMSE 0.13353), followed by LSTM and CNN, while SVR shows the highest error. Panel data regression identifies the fixed-effect model as the best specification with an R-squared value of 0.9209, indicating that 92% of AQI variation is explained by Humidity, PM2.5, O_3, Pressure, and CO. The t-tests confirm that PM2.5 and O_3 have significant effects on AQI, whereas meteorological variables and CO do not. Overall, the three analytical approaches consistently highlight PM2.5 as the dominant factor affecting AQI, with machine-learning models—particularly GRU—providing the most accurate predictions, while panel regression offers clarity regarding the contribution of each variable to AQI variation.

Keywords: AQI(*Air Quality Index*), ARIMA, *GRU(Gated Recurrent Unit)*, Machine Learning, Panel Data Regression.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kualitas udara merupakan salah satu faktor penting yang berpengaruh langsung terhadap kesehatan masyarakat dan keberlanjutan lingkungan. Indeks Kualitas Udara atau *Air Quality Index* (AQI) digunakan untuk menyederhanakan informasi polusi udara menjadi indikator yang lebih mudah dipahami. WHO menyatakan polusi udara luar ruangan di kota-kota dan daerah pedesaan diperkirakan menyebabkan 4,2 juta kematian dini di seluruh dunia per tahun pada tahun 2019, kematian ini disebabkan oleh paparan partikel halus, yang menyebabkan penyakit kardiovaskular dan pernapasan, serta kanker [1]. Tingginya AQI sering dikaitkan dengan meningkatnya polusi dari sektor transportasi, industri, pembakaran terbuka, serta variabilitas kondisi meteorologis seperti suhu, kelembapan, dan kecepatan angin. Oleh karena itu, pemahaman mengenai faktor-faktor yang memengaruhi AQI menjadi penting untuk mendukung perencanaan kebijakan lingkungan yang efektif.

Seiring meningkatnya perhatian terhadap kualitas udara global, pendekatan berbasis data menjadi semakin relevan untuk memahami dinamika polusi udara. Analisis korelasi sering digunakan sebagai langkah awal untuk melihat hubungan antara AQI dan variabel-variabel lain, seperti polutan utama atau faktor lingkungan. Pendekatan ini membantu mengidentifikasi variabel yang memiliki keterkaitan paling kuat terhadap perubahan nilai AQI, sehingga dapat menjadi dasar dalam pemilihan variabel untuk analisis lanjutan.

Selain itu, AQI merupakan data yang tercatat secara berurutan dari waktu ke waktu, sehingga pendekatan analisis deret waktu sangat diperlukan untuk mempelajari pola historisnya. Analisis deret waktu dapat mengungkap tren jangka panjang, fluktuasi jangka pendek, dan pola musiman yang mungkin terjadi. Misalnya, model seperti ARIMA dan AR-GARCH telah terbukti efektif dalam memprediksi kualitas udara. Studi forecasting pada kualitas udara di Sofia (Eropa Timur) menunjukkan bahwa metode ARIMA dapat digunakan untuk memproyeksikan konsentrasi polutan (CO, NO_2, O_3, PM2.5) dengan hasil prediksi lebih dekat ke nilai aktual [4]. Lebih lanjut, penelitian terbaru menunjukkan bahwa kombinasi model statistik dan teknik pembelajaran mesin (machine learning), seperti GRU (*Gated Recurrent Unit*), LSTM (*Long Short-Term Memory*), RNN (*Recurrent*

Neural Network), CNN (*Convolutional Neural Network*), dan SVR (*Support Vector Regression*), mampu meningkatkan akurasi prediksi AQI dibandingkan model tradisional [5].

Namun, perubahan kualitas udara tidak hanya dipengaruhi oleh waktu, tetapi juga oleh perbedaan karakteristik antar lokasi pengamatan. Oleh karena itu, metode regresi data panel menjadi sangat relevan digunakan dalam penelitian modern. Konsep regresi data panel memungkinkan penggabungan variasi deret waktu dan variasi antar unit *cross-section*, sehingga analisis dapat mengontrol perbedaan karakteristik tetap yang tidak dapat diamati secara langsung [6]. Studi terbaru menunjukkan bahwa model panel, seperti *Pooled OLS*, *Fixed Effect*, dan *Random Effect*, mampu memberikan estimasi yang lebih stabil dalam menganalisis hubungan antara variabel respon dan prediktor serta variabel kontrol [6].

Penelitian ini memanfaatkan seluruh variabel dalam dataset, baik variabel polutan maupun variabel meteorologis, untuk mengidentifikasi faktor-faktor yang memengaruhi perubahan kualitas udara. Analisis korelasi digunakan untuk menggambarkan hubungan antarvariabel dan menentukan variabel yang memiliki keterkaitan awal dengan perubahan indeks kualitas udara. Selanjutnya, analisis regresi data panel diterapkan untuk memperoleh pemahaman yang lebih mendalam mengenai variabel yang paling berpengaruh terhadap fluktuasi indeks kualitas udara pada berbagai lokasi pengamatan. Analisis deret waktu kemudian digunakan untuk mempelajari pola historis, tren, dan dinamika perubahan indeks kualitas udara, yang selanjutnya diuji dengan berbagai metode pemodelan, mulai dari model statistik konvensional hingga metode modern berbasis machine learning yang bertujuan mengevaluasi kemampuan prediksi yang paling akurat. Melalui rangkaian analisis tersebut, penelitian ini bertujuan menghasilkan pemahaman komprehensif mengenai perilaku kualitas udara serta menyediakan dasar ilmiah yang kuat dalam mendukung pengelolaan kualitas udara yang lebih efektif dan berkelanjutan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka hal yang menjadi permasalahan pada penelitian ini adalah sebagai berikut:

1. Bagaimana hubungan antara seluruh variabel dalam dataset dengan Air Quality Index (AQI) berdasarkan analisis korelasi?

2. Faktor-faktor mana saja diantara variabel polutan (PM2.5, O_3, CO) dan variabel meteorologi (Humidity, Pressure) yang berpengaruh signifikan terhadap perubahan AQI berdasarkan analisis regresi data panel?
3. Bagaimana performa metode pemodelan deret waktu dengan pendekatan statistik (ARIMA/AR-GARCH) maupun pendekatan machine learning (RNN, LSTM, GRU, CNN, SVR) dalam memprediksi nilai AQI?

1.3 Batasan Masalah

Penelitian ini dibatasi pada penggunaan data AQI beserta variabel lingkungan dan aktivitas lain yang tersedia dalam dataset, tanpa penambahan variabel eksternal. Metode yang digunakan terbatas pada analisis korelasi, analisis deret waktu, dan regresi data panel. Model prediksi yang digunakan berbasis pada data historis tanpa menyertakan simulasi faktor eksternal. Selain itu, penelitian tidak membahas aspek geografis, kebijakan, maupun perbandingan antarwilayah karena dataset tidak memuat informasi lokasi atau negara.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang dihadapi, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis hubungan antara seluruh variabel dalam dataset dengan Air Quality Index (AQI) berdasarkan analisis korelasi.
2. Menganalisis faktor-faktor mana saja diantara variabel polutan (PM2.5, O_3, CO) dan variabel meteorologi (Humidity, Pressure) yang berpengaruh signifikan terhadap perubahan AQI berdasarkan analisis regresi data panel
3. Mengidentifikasi performa metode pemodelan deret waktu dengan pendekatan statistik (ARIMA/AR-GARCH) maupun pendekatan machine learning (RNN, LSTM, GRU, CNN, SVR) dalam memprediksi nilai AQI.

1.5 Manfaat Penelitian

Penelitian ini memberikan kontribusi teoritis dalam pengembangan ilmu pengetahuan, khususnya pada kajian kualitas udara dan metode statistik terapan, dengan memperkaya literatur melalui pendekatan analisis yang memadukan korelasi, deret waktu, dan regresi data panel sebagai kerangka metodologis yang dapat dijadikan acuan bagi penelitian selanjutnya.

Secara praktis, penelitian ini bermanfaat bagi berbagai pihak yang membutuhkan pemahaman mengenai faktor penyebab perubahan AQI untuk mendukung pengambilan keputusan, evaluasi lingkungan, dan pemantauan kualitas udara. Hasil yang diperoleh dapat membantu mengidentifikasi variabel yang paling berpengaruh serta menjadi dasar dalam perencanaan, pengawasan, maupun pengembangan sistem prediksi kualitas udara di masa mendatang.

BAB II
TINJAUAN PUSTAKA

2.1 Mean

Mean adalah salah satu ukuran gejala pusat. Mean dapat dikatakan sebagai wakil kumpulan data. Menentukan mean dapat dilakukan dengan cara menjumlahkan seluruh nilai data, kemudian membaginya dengan banyaknya data [16]. Rumus untuk menghitung mean adalah sebagai berikut:

rumus data tunggal:

$$\bar{x} = \frac{\Sigma n}{n} \tag{2.1}$$

rumus data kelompok:

$$\bar{x} = \frac{\Sigma F_n . X_n}{\Sigma F_n} \tag{2.2}$$

keterangan :

$\bar{x}$: rata-rata sampel

n : jumlah sampel

ΣF_n : jumlah hasil kali antara frekuensi dengan nilai data kelas ke-n

X_n : nilai data kelas ke-n

2.2 Median

Median adalah salah satu teknik penjelasan kelompok yang didasarkan atas nilai tengah dari kelompok data yang telah disusun urutannya dari yang terkecil sampai yang terbesar, atau sebaliknya dari yang terbesar sampai yang terkecil [16]. Rumus median adalah sebagai berikut:

rumus n ganjil:

$$Me = x_{(\frac{n+1}{2})} \quad (2.3)$$

rumus n genap:

$$Me = (\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}) \quad (2.4)$$

keterangan:

Me: median

n: jumlah data

$x_{\frac{n}{2}}$: data ke- $\frac{n}{2}$ setelah data diurutkan

$x_{(\frac{n+1}{2})}$: data ke- $\frac{n+1}{2}$ setelah data diurutkan

2.3 Standar Deviasi

Standar deviasi atau simpangan baku adalah persebaran data pada suatu sampel untuk melihat seberapa jauh atau seberapa dekat nilai data dengan rata-ratanya [15]. Nilai standar deviasi yang semakin kecil menandakan semakin dekat dengan dengan rata-ratanya sedangkan semakin besar nilai standar deviasi maka semakin lebar variasi datanya. Berikut rumus standar deviasi:

rumus data tunggal:

$$s = \frac{\sqrt{\sum_{i}^{n}=1(x_i - \bar{x})}}{n-1} \quad (2.5)$$

rumus data kelompok:

$$s = \frac{\sqrt{\Sigma f_1 (x_i - \bar{x})^2}}{\Sigma f_1} \tag{2.6}$$

keterangan:

S : standar deviasi

x_i : nilai tengah

$\bar{x}$: rata-rata sampel

f_1 : frekuensi

2.4 Korelasi

Korelasi merupakan bagian dari teknik analisis yang termasuk dalam kelompok *measures of association*, yakni seperangkat metode statistik bivariat yang digunakan untuk menilai derajat keterkaitan antara dua variabel. Berbagai bentuk pengukuran asosiasi telah dikembangkan, namun hingga kini terdapat dua metode korelasi yang paling sering diterapkan, yaitu Korelasi Pearson Product Moment dan Korelasi Rank Spearman [7]. Sedangkan korelasi Spearman, yaitu metode nonparametrik yang mengukur hubungan berdasarkan peringkat nilai data. Korelasi menggunakan metode Spearman digunakan untuk mengetahui korelasi antar variabel yang berbentuk ordinal/berjenjang dengan variabel yang digunakan tidak mesti sama, langkah awal yang dilakukan pada analisis korelasi Spearman adalah dengan diberi rangking berdasarkan skor masing-masing dari yang terbesar hingga ke yang terkecil, dan kemudian dapat dihitung menggunakan rumus [8]:

keterangan:

$$\rho = 1 - \frac{n\Sigma d^2}{n(n^2-1)} \tag{2.7}$$

$\rho\ (rho)$: koefisien korelasi Rank Spearman

d : beda peringkat yang berpasangan

n : jumlah data

berikut tabel batas kategori korelasi:

Tabel 2.1 Kategori Korelasi

Interval koefisien	Keeratan korelasi
< 0,2	Sangat lemah
0,2 - 0,399	Lemah
0,4 - 0,599	Sedang
0,6 - 0,799	Kuat
0,8 - 0,999	Sangat kuat
1	Korelasi sempurna

2.5 Regresi Data Panel

Metode Regresi Data Panel dilakukan dengan cara melakukan tiga pendekatan model, yakni: model *Common Effect Model* (CEM), *Fixed Effect Model* (FEM), dan *Random Effect Model* (REM). Pendekatan ketiga model ini bertujuan untuk menentukan model terbaik diantara tiga pendekatan tersebut. Adapun model yang digunakan adalah sebagai berikut:

$$Y_{it} = \alpha + \sum_{k=1}^{K} \beta_k X_{k_{it}} + \varepsilon_{it} \tag{2.8}$$

keterangan:

Y_{it} : variabel dependen pada unit i dan waktu t

α : konstanta model

β_k : koefisien regresi variabel independen ke-k

X_{kit} : variabel independen ke-k pada unit i dan waktu t

ε_{it} : error term pada unit i dan waktu t

2.6 Estimasi Model Regresi

Estimasi model regresi dilakukan untuk memilih salah satu model terbaik di antara model *common effect*, *fixed effect* atau *random effect*. Terdapat beberapa uji yang dapat dilakukan, diantaranya, yaitu:

2.6.1 Uji Chow

Uji ini dilakukan untuk membandingkan model *common effect* dengan model *fixed effect*. Hipotesis yang dibentuk dalam uji chow adalah sebagai berikut:

H_0 = Model *Common Effect*

H_1 = Model *Fixed Effect*

2.6.2 Uji Hausman

Uji ini dilakukan untuk membandingkan model *random effect* dengan model *fixed effect*. Hipotesis yang dibentuk dalam uji hausman adalah sebagai berikut:

H_0 = Model *Random Effect*

H_1 = Model *Fixed Effect*

2.7 Uji Asumsi Klasik

Untuk memenuhi asumsi klasik, data panel harus bersifat non-multikolinieritas dan non-heteroskedastisitas. Berikut adalah beberapa uji yang dapat dilakukan:

2.7.1 Uji Multikolinieritas

Uji multikolinearitas bertujuan untuk menguji apakah dalam model regresi ditemukan adanya korelasi yang tinggi atau sempurna antar variabel independen [26]. Uji multikolinearitas dapat dilakukan dengan uji regresi, dengan patokan nilai Tolerance dan nilai VIF (*Variance Inflation Factor*). Untuk mendeteksi multikolinearitas pedomannya adalah sebagai berikut:

Nilai *Tolerance* > 0,10 dan nilai *VIF* < 10 = tidak terjadi multikolinearitas.

Nilai *Tolerance* < 0,10 dan nilai *VIF* > 10 = terjadi multikolinearitas.

2.7.2 Uji Heteroskedastisitas

Uji ini dilakukan guna melihat apakah pada sebuah model regresi terjadi ketidaksamaan varian dari residual dalam satu pengamatan ke pengamatan lainnya. Hipotesis yang dibentuk dalam uji heteroskedastisitas adalah sebagai berikut:

H_0 = terjadi heteroskedastisitas

H_1 = bebas heteroskedastisitas

2.7.3 Uji Autokorelasi

Uji autokorelasi bertujuan untuk menguji apakah dalam suatu model regresi linier terdapat korelasi antar kesalahan pengganggu (*residual*) pada periode t-1 (sebelumnya). Jika terjadi korelasi, maka dinamakan terdapat permasalahan autokorelasi. Metode pengujian yang sering digunakan yaitu dengan uji *Durbin-Watson* (uji DW) dengan ketentuan sebagai berikut [3]:

Jika d lebih kecil dari dL atau lebih besar dari (4-dL) maka hipotesis nol ditolak, yang berarti terdapat autokorelasi.

Jika d terletak antara dU dan (4-dU), maka hipotesis nol diterima, yang berarti tidak ada autokorelasi.

Jika d terletak antara antara dL dan dU atau diantara (4-dU) dan (4-dL), maka tidak menghasilkan kesimpulan yang pasti.

2.7.4 Uji Normalitas

Uji normalitas digunakan untuk menguji apakah dalam model regresi, residual berdistribusi normal atau tidak. Metode pengujian yang dilakukan pada penelitian ini adalah menggunakan uji Shapiro Wilk dengan ketentuan sebagai berikut:

Jika p-value > 0,05 = gagal tolak H_0

Jika p-value < 0,05 =tolak H_0

2.8 Uji Hipotesis

Uji hipotesis adalah uji statistik inferensial yang dilakukan dengan menaksir parameter populasi berdasarkan data sampel, yaitu untuk menguji kebenaran suatu pernyataan secara statistik serta menarik kesimpulan menerima atau menolak pernyataan tersebut [26]. Dalam uji statistik terdapat tiga indikator penting yaitu uji F (uji simultan), uji t (uji parsial) dan Koefisien Determinasi. Ketiga indikator ini digunakan untuk mengukur signifikansi model, menguji pengaruh variabel secara individu maupun simultan, serta menilai seberapa besar kemampuan model dalam menjelaskan variabel dependen.

2.8.1 Uji F-statistik

Uji F bertujuan untuk mengetahui apakah variabel independen secara simultan berpengaruh terhadap variabel dependen. Keputusan hipotesis didasarkan pada nilai signifikansi: jika p > 0,05 atau t hitung < t tabel, H0 gagal tolak (tidak ada pengaruh); jika p < 0,05 atau t hitung > t tabel, H0 ditolak (variabel independen berpengaruh signifikan terhadap variabel dependen). Di dalam pengujian hipotesis digunakan statistik uji F sebagai berikut :

$$F_{hitung} = \frac{SSR/k}{SSE/(n-k-1)} \qquad (2.10)$$

keterangan:

SSE : Regression Sum of Square (jumlah kuadrat regresi

SSR : Error Sum of Square (jumlah kuadrat residual)

n : jumlah data

k : jumlah parameter (variabel independen)

2.8.2 Uji t-statistik

Uji t-statistik dapat dilakukan untuk melihat apakah variabel bebas secara parsial berpengaruh secara signifikan terhadap variabel terikat. Variabel bebas secara parsial berpengaruh signifikan terhadap

variabel terikat apabila nilai probabilitasnya < α = 0,05. Untuk rumus dalam uji ini dapat ditulis menjadi:

$$t = \frac{\bar{\beta}_k}{SE(\bar{\beta}_k)} \quad (2.9)$$

keterangan:

$\bar{\beta}$: hasil estimasi

β_0 : nilai koefisien dibawah hipotesis

$SE(\bar{\beta})$: standar error dari koefisien

2.8.3 Uji Koefisien Determinasi

Koefisien determinasi merupakan koefisien yang menjelaskan hubungan antara variabel dependen (Y) dengan variabel independen (Y) dalam suatu model [26]. Koefisien determinasi (R^2) menunjukkan seberapa besar proporsi variasi pada variabel dependen (Y) yang dapat dijelaskan oleh variabel independen (X). Nilai R^2 yang mendekati 1 menunjukkan bahwa model memiliki kemampuan penjelasan yang kuat, sedangkan nilai yang mendekati 0 menunjukkan kemampuan penjelasan yang lemah.

Secara statistik dirumuskan berikut :

$$R^2 = 1 - \frac{SSE}{SST} \quad (2.11)$$

keterangan:

SSE : regression Sum of Square (jumlah kuadrat regresi)

SST : Total Sum of Square (jumlah kuadrat total)

2.9 Analisis Deret Waktu

Selain menggunakan metode analisis korelasi, penelitian ini menggunakan pendekatan analisis deret waktu (*time series*) untuk mempelajari pola perubahaan AQI dari waktu ke waktu. Analisis deret waktu merupakan analisis statistika yang digunakan untuk mengolah data observasi atau data amatan yang berbentuk urutan waktu (*sequential*).

2.9.1 Model ARIMA

Metode Arima adalah metode peramalan jangka pendek akurat dengan menggunakan variabel data *time series* dengan mengabaikan independennya. Model ini juga disebut sebagai model Box-Jenkins dengan bentuk umumnya adalah ARIMA (p, d, q), yang mana p menyatakan ordo *autoregressive* (AR), d merupakan ordo *integrated* (I), dan q merupakan ordo *moving average* (MA) [10]. Hal ini membuat model cukup berguna untuk digunakan dalam memperkirakan kondisi kualitas udara, terutama pada wilayah perkotaan yang memiliki data historis yang konsisten dan panjang. Dalam penyusunannya, model Arima digambarkan menggunakan operator *backshift* untuk mempermudah penulisan komponen *autoregressive* maupun *moving average*. Secara khusus, bagian *moving average* dapat ditulis dalam bentuk polinomial, sehingga hubungan antara nilai sekarang dan error dapat digambarkan dengan lebih ringkas. Bentuk umum polinomial *moving average* dapat digambarkan sebagai berikut:

$$\Theta(B) = 1 - \Theta_1 B - \Theta_2 B^2 - \ldots\ldots - \Theta_q B^q \qquad (2.11)$$

keterangan:

$\Theta(B)$: polinomial MA (*moving average*)

B : operator *backward* (lag operator)

Θ_i : koefisien MA ke-i

q : orde MA

2.9.2 Stasioneritas

Dalam teori *time series*, penting untuk memastikan data bersifat stasioner melalui pengujian seperti ADF Test. Stasioner merupakan suatu kondisi yang dimana tidak ada kenaikan maupun penurunan data yang ekstrem, artinya data berada pada persekitaran nilai yang sama [9]. Adapun jika terdapat suatu data yang tidak stasioner, maka dapat distasionerkan menggunakan rumus:

$$Y_t = \varphi + \beta t + \Theta D_t + yD_t t + \rho Y_{t-1} + \sum_{j=1}^{k} c_t \Delta Y_{t-1} \quad (2.10)$$

keterangan:

Y_t : nilai variabel deret waktu pada waktu ke-t

t : komponen tren (kemiringan tren waktu)

φ : konstanta

β : koefisien tren

D_t : variabel dummy

Θ : koefisien dummy level

y : koefisien dummy pada tren

ρY_{t-1} : nilai variabel dengan lag 1

ΔY_{t-1} : perubahan (*difference*) variabel dengan lag 1

c_t : koefisien lag dari ΔY

k : jumlah lag

ϵ_t : error term

2.9.3 Uji *White Noise*

Dalam metode analisis deret waktu, diperlukan pemeriksaan apakah residual yang dihasilkan model bersifat acak atau tidak. Uji Ljung-Box digunakan untuk menguji apakah residual memiliki autokorelasi yang signifikan. Residual dikatakan *white noise* apabila tidak menunjukkan pola tertentu, sebab serangkaian error tidak saling berkorelasi dan memiliki varians konstan dari waktu ke waktu. Secara matematis, proses *white noise* dapat dinyatakan sebagai berikut:

$$\varepsilon_t \approx \text{ıı}d(0, \sigma^2) \quad (2.12)$$

keterangan:

ϵ_t : *white noise* pada waktu ke-t

ud : *uniform distribution* (distribusi uniform)

$(0, \sigma^2)$: parameter distribusi (mean 0 dan varians σ^2)

2.9.4 Model AR-GARCH

Selain autokorelasi, heteroskedastisitas pada residual juga perlu di periksa. Uji ARCH (*Autoregressive Conditional Heteroskedasticity*) digunakan untuk mengetahui apakah varians residual berubah dari waktu ke waktu. Heteroskedastisitas merupakan sebuah kondisi dimana semua faktor tidak memiliki varian yang sama, kondisi ini juga disebut sebagai varian nir-konstan atau varian nir-homogen [11]. Untuk Mendeteksi heteroskedastisitas pada suatu deret waktu, menggunakan rumus Model ARCH(1) serta Model ARCH(q). Adapun bentuk dari rumus tersebut, sebagai berikut:

Pada Model ARCH(1), model ini hanya memasukkan satu lag kuadrat residual, sehingga varians pada waktu ke-t hanya dipengaruhi oleh besarnya shock atau error pada periode sebelumnya.

$$\sigma_t^2 = \sigma_0 + \sigma_1 \epsilon_{t-1}^2 \tag{2.13}$$

Pada Model ARCH(q), varians kondisional ditemukan oleh q lag kuadrat residual, sehingga shock dari beberapa periode sebelumnya ikut memengaruhi varians saat ini.

$$\sigma_t^2 = \sigma_0 + \sigma_1 \epsilon_{t-1}^2 + \sigma_2 \epsilon_{t-2}^2 + \ldots + \sigma_q \epsilon_{t-q}^2 \tag{2.14}$$

keterangan:

σ_t^2	: varians kondisional pada waktu ke-t
$\sigma 0$	: konstanta (harus bernilai positif)
$\sigma_1, \sigma_2, \ldots, \sigma_q$	: koefisien ARCH pada lag 1 hingga lag 1
$\epsilon_{t-1}^2, \epsilon_{t-2}^2, \ldots, \epsilon_{t-q}^2$	: kuadrat residual dari lag sebelumnya
q	: jumlah lag residual yang dipakai dalam model

2.10 *Machine Learning Forecasting*

Sebagai upaya untuk mengatasi keterbatasan model deret waktu konvensional, penerapan pendekatan *machine learning* yang memiliki kemampuan lebih baik dalam menangkap pola non-linier serta dependensi jangka panjang dalam

data sangat dibutuhkan. Berbeda dari model statistik klasik, metode *machine learning* mampu mempelajari representasi kompleks secara otomatis melalui proses pelatihan berbasis data. Beberapa model yang digunakan dalam penelitian ini adalah: RNN, LSTM, GRU, CNN, dan SVR.

2.10.1 RNN *(Recurrent Neural Network)*

RNN merupakan salah satu jenis ANN yang dapat memproses data secara berurutan atau sekuensial. RNN memiliki memori internal yang memungkinkannya untuk mengingat informasi dari masa lalu dan menggunakan informasi tersebut untuk memprediksi masa depan. Langkah pertama yang dilakukan pada model RNN yaitu menghitung *hidden state* baru h_t melalui fungsi dari input saat ini x_t dan *hidden state* sebelumnya h_{t-1}. untuk output numerik dari model RNN diperoleh dari total nilai *hidden state* berdasarkan bobot masing-masing unitnya [12].

$$h_t = \tau(W_x x_t + W_h h_{t-1} + b_h) \tag{2.15}$$

$$o_t = W_0 h_t + b_0 \tag{2.16}$$

keterangan:

h_t	: *hidden state* pada waktu t
τ	: fungsi aktivasi
W_x, W_h, W_o	: matriks bobot
x_t	: input pada waktu t
h_{t-1}	: *hidden state* sebelumnya
b_h, b_o	: bias
o_t	: *output* pada waktu t

2.10.2 LSTM *(Long Short-term Memory)*

LSTM merupakan jenis dari RNN yang terdapat penambahan *memory cell* dan dapat menyimpan informasi untuk jangka waktu

yang lebih lama. LSTM diusulkan sebagai solusi untuk menangani masalah *vanishing gradient* pada model RNN standar saat memproses data deret waktu yang panjang. Penanganan masalah *vanishing gradient* pada RNN dilakukan melalui skema memori yang terdiri dari tiga bagian yaitu *forget gate, input gate,* dan *output gate* [12].

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.17}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.18}$$

$$\bar{c}_t = \tau(W_c[h_{t-1}, x_t] + b_c) \tag{2.19}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \bar{c}_t \tag{2.20}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.21}$$

$$h_t = o_t \odot \tau(c_t) \tag{2.22}$$

keterangan:

f_t	: *forget gate*
σ	: sigmoid
W_f, W_i, W_c, W_o	: matriks bobot
h_t, h_{t-1}	: *hidden state* saat ini dan sebelumnya
x_t	: input waktu t
b_f, b_i, b_c, b_o	: bias
i_t	: *input gate*
c_t, c_{t-1}	: *cell state* saat ini dan sebelumnya
τ	: fungsi aktivasi
f_t	: *forget gate*
$\odot$	: operasi perkalian elemen (elemen-wise)
$\bar{c}_t$	: *candidate cell state*

2.10.3 GRU *(Gated Recurrent Unit)*

GRU bisa dibilang sebagai variasi lain dari LSTM, karena kedua algoritma ini dapat memberikan hasil yang sangat baik untuk banyak kasus empiris. GRU memiliki jumlah *gate* lebih sedikit dibandingkan LSTM yaitu *update gate* dan *reset gate,* GRU juga dianggap mampu menangani masalah *vanishing gradient* yang terjadi dalam jaringan RNN standar. Adapun tahapan pembelajaran pada model GRU, dapat dilihat pada rumus sebagai berikut [12]:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (2.23)$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (2.24)$$

$$\overline{h}_t = \tau(W_h[r_t \Theta h_{t-1}, x_t] + b_h) \quad (2.25)$$

$$h_t = (1 - z_t) \Theta h_{t-1} + z_t \Theta \overline{h}_t \quad (2.26)$$

$$o_t = \tau(W_0 h_t + b_0) \quad (2.27)$$

keterangan:

z_t	: *update gate*
r_t	: *reset gate*
$\overline{h}_t$	: kandidat *hidden state*
h_t	: *final hidden state*
o_t	: output layer
W_z, W_r, W_h,W_o	: matriks bobot
b_z, b_r, b_h, b_o	: bias
σ	: sigmoid
h_{t-1}	: *hidden state* pada waktu sebelumnya
x_t	: input pada waktu ke-t
τ	: fungsi aktivasi
⊙	: operasi perkalian elemen (elemen-wise)

2.10.4 SVR (*Support Vector Regression*)

Algoritma SVR adalah turunan dari metode *Support Vector Machine* (SVM) yang dimana sering dipakai untuk proses klasifikasi. Penggunaan dari metode SVR dipakai dalam permasalahan regresi. Perbedaan yang paling mendasar antara SVM dan SVR terletak pada nilai yang akan diregresikan. Tujuan dari SVR adalah untuk

menemukan fungsi *hyperline* (garis pemisah) berupa fungsi regresi yang sesuai dengan semua input data dengan sebuah error serta membuatnya sekecil mungkin. Adapun untuk fungsi regresi yang digunakan dari metode SVR adalah sebagai berikut:

$$f(x) = w\varphi(x) + b \tag{2.28}$$

keterangan:

$f(x)$: hasil prediksi

w : vektor bobot

$\varphi(x)$: fungsi kernel / *mapping* ke ruang fitur

b : bias

2.10.5 CNN (*Convolutional Neural Network*)

Convolutional Neural Network merupakan salah satu jenis *Neural Network* (NN) yang biasa digunakan dalam *computer vision*. *Computer vision* sendiri merupakan salah satu bidang kecerdasan buatan yang memungkinkan komputer memahami dan menafsirkan gambar. *Convolutional Neural Network* memiliki fungsi untuk menemukan pola dalam gambar untuk mengenali objek[19]. CNN merupakan metode yang efektif untuk mengklasifikasikan data non-gambar seperti audio, deret waktu, dan sinyal. Model CNN memiliki struktur yang terdiri dari *convolution layer, polling layer,* dan *fully connected layer. Convolution layer* bertujuan untuk mengekstrak fitur lokal seperti dimensi dari gambar yang akan diinput, untuk menghitung dimensinya dapat dirumuskan sebagai berikut:

$$V = \frac{W - F + 2P}{S} + 1 \tag{2.29}$$

keterangan:

V : volume dimensi

W : panjang (tinggi input)

F : panjang (tinggi filter)

P : *zero padding*

S : *stride*

Selanjutnya, pada *polling layer* berfungsi untuk mengurangi beban komputasi untuk memproses data dengan mengurangi beban dimensi komputasi. Pada *fully connected layer*, lapisan ini menjadi lapisan akhir sebelum lapisan *output* dan membentuk beberapa lapisan terakhir dari arsitektur CNN. Sehingga, CNN akan kurang akurat ketika diberikan dataset yang kecil. Sebaliknya, CNN akan dapat menunjukkan akurasi yang signifikan ketika diberi kumpulan dataset gambar yang besar[19].

2.11 AQI (*Air Quality Index*)

AQI atau bahasa indonesianya ISPU adalah singkatan dari Indeks Standar Pencemaran udara. Indeks ini digunakan untuk mengukur tingkat polusi udara dalam satu wilayah dan memberikan informasi kepada masyarakat mengenai risiko kesehatan akibat polusi udara tersebut. ISPU mengukur konsentrasi beberapa polutan udara seperti Particulate Matter (PM10 dan PM2.5), Ozon (O_3), Nitrogen Dioksida (NO_2), dan Sulfur Dioksida (SO_2). ISPU memiliki rentang skala dari 0 hingga 500, dan semakin tinggi nilai dari ISPU maka dapat dikatakan semakin buruk kualitas di wilayah tersebut dan semakin berbahaya bagi kesehatan manusia [23].

2.12 Polutan Udara

Polutan udara, seperti partikel debu, gas beracun (misalnya SO_2, NO_2, CO, dan O_3), dapat berasal dari berbagai sumber seperti industri, transportasi, dan aktivitas pembakaran. Adapun berbagai macam polutan dan sumber-sumber utamanya adalah sebagai berikut [13]:

2.12.1 Partikel Debu (PM10 dan PM2.5)

Partikulat merupakan partikel padat atau cair berukuran sangat kecil yang tersuspensi di udara. PM10 adalah partikel dengan diameter $\leq$ 10 μm, sedangkan PM2.5 berukuran $\leq$ 2,5 μm. PM2.5 lebih berbahaya karena mampu menembus saluran pernapasan hingga ke alveoli paru-paru.

Konsentrasi partikulat yang tinggi berhubungan dengan peningkatan risiko penyakit pernapasan, kardiovaskular, serta penurunan kualitas hidup masyarakat [13]. Oleh karena itu beberapa penelitian menjadikan PM2.5 sebagai variabel respon ,seperti penelitian oleh Willia yang meneliti kualitas udara di DKI Jakarta dengan PM2.5 sebagai variabel respon [18].

2.12.2 Gas Beracun (SO_2, NO_2, CO, O_3)

Gas beracun meliputi sulfur dioksida (SO_2), nitrogen dioksida (NO_2), karbon monoksida (CO), dan ozon (O_3). SO_2 dan NO_2 terutama dihasilkan dari pembakaran bahan bakar fosil pada aktivitas industri, pembangkit listrik, dan kendaraan bermotor, sedangkan CO terbentuk akibat pembakaran tidak sempurna. O_3 di lapisan troposfer bukan merupakan emisi langsung, melainkan terbentuk melalui reaksi fotokimia antara nitrogen oksida (NOx) dan senyawa organik volatil (VOCs) di bawah pengaruh sinar matahari. Keberadaan gas-gas tersebut berkontribusi terhadap penurunan kualitas udara dan berdampak negatif terhadap kesehatan manusia serta lingkungan [13]. Namun, pengaruh masing-masing polutan gas terhadap kualitas udara tidak selalu bersifat langsung. Penelitian Winda yang mengacu pada teori Lu et al. menyatakan bahwa pengaruh polutan seperti NO_2, O_3, dan CO dimediasi oleh faktor meteorologi lain, seperti suhu dan kecepatan angin [20]. Oleh karena itu, hubungan antara polutan gas dan kualitas udara bersifat kontekstual, dipengaruhi oleh sumber emisi, kondisi atmosfer, serta interaksi antar polutan [22], sehingga pengendalian polutan gas perlu dilakukan secara terpadu untuk menekan risiko pencemaran udara di suatu wilayah.

2.13 Faktor Cuaca

Beberapa parameter cuaca seperti suhu udara, kecepatan, angin, lama penyinaran matahari, curah hujan, dan indeks labilitas) juga mempengaruhi kelabilan atmosfer. Kondisi ini dapat mengangkat partikel-partikel yang terlepas di udara hingga ke level yang tinggi. Sehingga dapat membuat kondisi udara di permukaan menjadi bersih. Selain dari pengaruh cuaca, terdapat faktor lain yang mempengaruhi pengurangan konsentrasi PM2.5 dan CO, yaitu pembatasan aktivitas manusia, seperti

yang terjadi pada saat pemberlakuan WFH pada tahun 2020 [24]. Sementara itu penelitian lain juga menyatakan bahwa konsentrasi banyak polutan termasuk partikulat halus umumnya berkorelasi positif dengan tekanan atmosfer ketika variabel seperti kecepatan angin dan curah hujan dikendalikan [21].

Berbagai penelitian menunjukkan bahwa faktor-faktor meteorologi memiliki pengaruh penting terhadap perubahan kualitas udara. Parameter cuaca seperti suhu, kelembapan, kecepatan angin, dan tekanan atmosfer terbukti mempengaruhi konsentrasi polutan seperti PM2.5, PM10, NO_2, CO, dan O_3. Kecepatan angin umumnya membantu menurunkan konsentrasi polutan melalui proses dispersi, sementara suhu tinggi dapat meningkatkan pembentukan ozon melalui reaksi fotokimia.Pemahaman mengenai pengaruh cuaca ini menjadi dasar penting dalam analisis kualitas udara serta perumusan strategi pengendalian polusi di wilayah perkotaan [25].

BAB III
METODOLOGI PENELITIAN

3.1 Variabel Penelitian dan Sumber Data

Sumber data dalam penelitian ini menggunakan data sekunder yang diperoleh dari Kaggle, yaitu *UrbanAirNet: Urban Air Quality and Weather Dataset*. Dataset ini memuat data kualitas udara dan cuaca yang didapatkan dari beberapa stasiun pemantauan perkotaan (*urban monitoring stations*) pada tahun 2020 hingga 2023.

Dataset ini diperoleh dari Kaggle pada tanggal 4 November 2025 melalui tautan: https://www.kaggle.com/datasets/ziya07/urbanairnet-urban-air-quality-and-weather-dataset. Perlu diketahui bahwa dataset ini tidak terdapat informasi spesifik mengenai negara atau kota asal data, sehingga penelitian dilakukan dengan menggunakan variabel yang ada tanpa fokus pada lokasi geografis tertentu.

Penelitian ini menggunakan beberapa variabel yang disesuaikan dengan kebutuhan tiap analisis. Variabel yang digunakan adalah sebagai berikut:

Tabel 3.1 Variabel Penelitian

Analisis	**Kode**	**Nama variabel**	**Skala data**
Korelasi		PM2.5	Rasio
		PM10	Rasio
		NO_2	Rasio
		SO_2	Rasio
		CO	Rasio
		O_3	Rasio
		Temp_C	Interval
		Humidity_%	Rasio
		Wind_Speed_mps	Rasio
		Pressure_hPa	Rasio
		AQI_Target	Rasio
		Rain_Binary	Nominal

Analisis	Kode	Nama variabel	Skala data
Deret waktu		AQI Target	Rasio
Regresi data panel	Y	AQI Target	Rasio
	X_1	Humidity_%	Rasio
	X_2	NO_2	Rasio
	X_3	O_3	Rasio
	X_4	Pressure_hPa	Rasio
	X_5	PM2.5	Rasio
	Z_1	Station ID	Nominal
	Z_2	DateTime	Interval

3.2 Langkah Analisis

Langkah analisis berikut menjelaskan prosedur yang dilakukan mulai dari pemrosesan data hingga evaluasi model. Setiap langkah yang disusun sesuai dengan alur pada diagram alir penelitian.

1. Import Data (file CSV)

   Pada tahap awal ini dilakukan dengan membaca dataset mentah dalam format CSV dengan menggunakan Python. Proses ini mencakup pemanggilan file, pengecekan struktur data (tampilan dataset, tipe data, dan jumlah baris), serta memastikan seluruh variabel terbaca dengan benar sebelum pengolahan data awal.

2. Pengolahan Data Awal

   Pada tahap ini dilakukan seleksi untuk menghapus kolom yang tidak diperlukan, mengonversi kolom tanggal ke format *DateTime*, memeriksa dan menangani nilai kosong maupun nilai nol. Variabel *Rain Category* diubah menjadi variabel kategorik biner (1 = rain, 0 = no rain). Pada tahap ini juga mencakup perhitungan statistik deskriptif dasar untuk memahami karakteristik data awal.

3. Analisis Statistik Deskriptif dan Uji Normalitas

   Pada tahap ini dilakukan analisis statistik deskriptif untuk melihat karakteristik awal data, meliputi ukuran pemusatan dan penyebaran (mean,

minimum, maksimum, standar deviasi). Untuk memahami distribusi lebih lanjut, dilakukan visualisasi histogram dan uji normalitas menggunakan Shapiro Wilk. Hasil dari tahap ini digunakan sebagai dasar pemilihan metode korelasi dan model statistik yang sesuai.

4. Analisis Korelasi

   Pada tahapan ini untuk mengidentifikasi hubungan antarvariabel dalam dataset secara menyeluruh. Analisis korelasi menggunakan metode Spearman sebab mampu menangani data dengan distribusi tidak normal. Hasil korelasi divisualisasikan dalam bentuk *heatmap* sehingga pola hubungan antarvariabel dapat terlihat secara komprehensif dan menjadi dasar pemilihan variabel penting pada tahap pemodelan.

5. Analisis Deret Waktu

   Tahap analisis deret waktu diawali dengan pemeriksaan asumsi menggunakan uji stasioneritas ADF untuk memastikan kestabilan rata-rata, uji *white noise* Ljung Box untuk mendeteksi autokorelasi dalam residual, serta uji heteroskedastisitas ARCH untuk melihat adanya dinamika varians. Setelah asumsi dicek, pemodelan dilakukan dengan menggunakan ARIMA yang melalui proses identifikasi menghasilkan model ARIMA (0,0,0) sebagai model awal, kemudian dilanjutkan dengan pendekatan AR-GARCH yang mengombinasikan pola ketergantungan sekaligus perubahan variabilitas data. Seluruh model dievaluasi menggunakan RMSE, MAE, dan MAPE untuk menilai akurasi prediksi dan membandingkan performansi metode deret waktu.

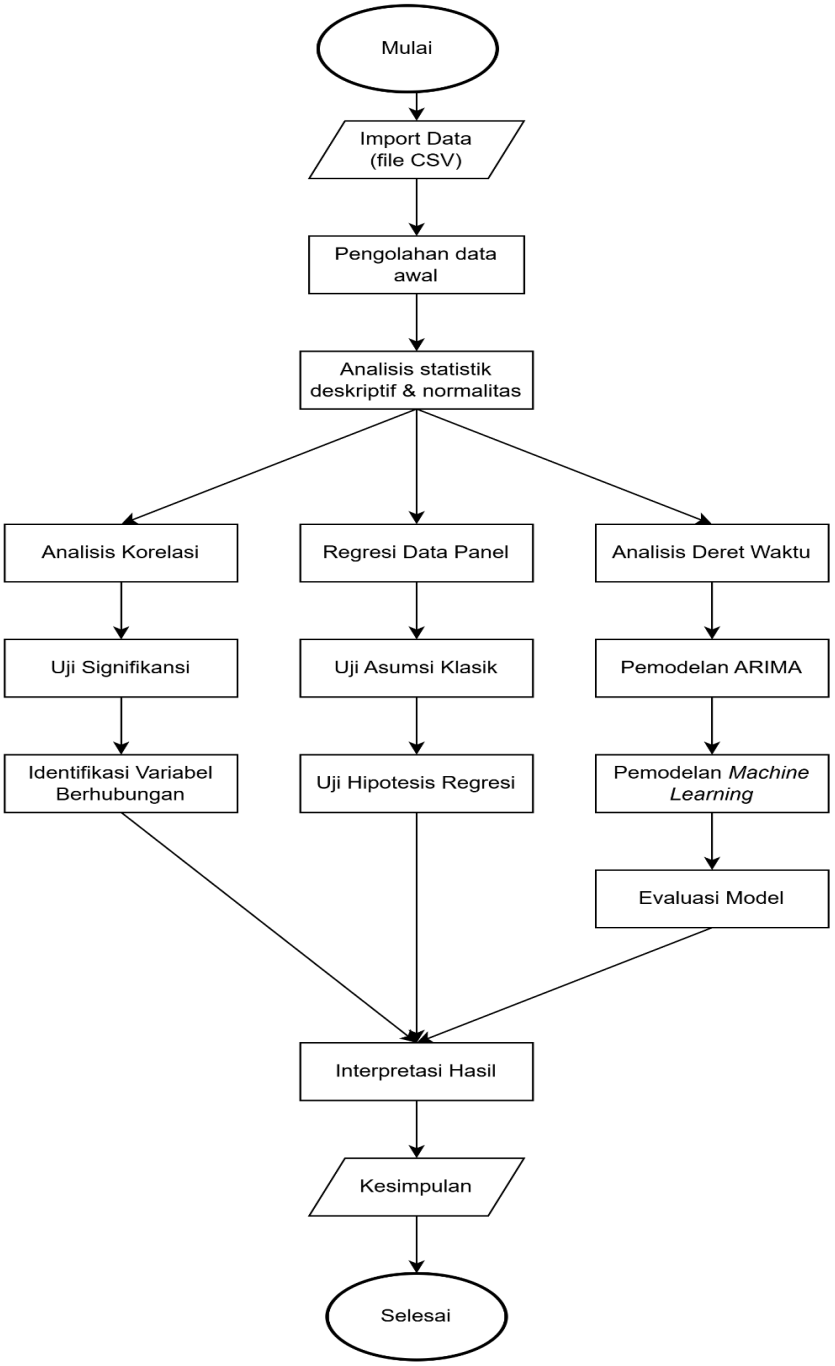
   Setelah pendekatan statistik deret waktu, tahap selanjutnya adalah pemodelan prediktif berbasis *machine learning* untuk meningkatkan akurasi prediksi AQI. Metode yang digunakan meliputi CNN, LSTM, GRU, RNN, dan SVR. Seluruh model dilatih menggunakan variabel pencemar dan variabel meteorologi sebagai fitur, dengan AQI sebagai target. Proses ini mencakup pembagian data menjadi *train-test*, normalisasi fitur, pelatihan model pada data historis, serta evaluasi performa menggunakan RMSE, MAE dan MAPE. Tahap ini menjadi pembanding terhadap model deret waktu, dengan fokus pada keunggulan *machine learning* dalam menangkap pola non-linier dan hubungan kompleks antarvariabel.

6. Analisis Regresi Data Panel

Tahap ini dilakukan untuk menganalisis pengaruh variabel pencemar dan variabel meteorologi terhadap AQI dalam konteks data lintas waktu dan lintas lokasi. Model regresi panel yang digunakan mencakup *Pooled OLS, Fixed Effect,* dan *Random Effect*, dengan pemilihan model terbaik ditentukan melalui uji Hausman. Analisis ini bertujuan memberikan pemahaman statistik mengenai variabel mana yang secara signifikan memengaruhi AQI, sehingga mendukung interpretasi hasil prediksi model-model sebelumnya.

3.3 Diagram Alir

Diagram alir berfungsi menggambarkan alur analisis yang dilakukan, mulai dari pengolahan data hingga evaluasi model. Diagram alir yang digunakan adalah sebagai berikut.



Gambar 3.1 Diagram Alir

BAB IV
HASIL DAN PEMBAHASAN

4.1 Statistika Deskriptif

Analisis deskriptif dilakukan untuk memberikan gambaran umum mengenai karakteristik masing-masing variabel penelitian. Berikut adalah hasil analisis menggunakan perintah python :

Tabel 4.1 Statistika Deskriptif

index	count	mean	min	max	std
PM2.5	175205	60,028	-30,550	145,072	19,993
PM10		89,970	-33,232	224,750	30,053
NO_2		24,949	-23,087	68,019	9,964
SO_2		10,002	-6,575	28,328	3,995
CO		0,599	-0,357	1,499	0,199
O_3		30,001	-11,472	75,843	10,002
Temp_C		27,005	3,083	48,162	4,992
Humidity_%		64,957	-3,501	130,720	15,001
Wind_Speed_mps		2,503	-1,857	7,156	1,003
Pressure_hPa		1009,995	987,584	1033,106	4,996
Rain_mm		0,094	0,0	1,0	0,238
AQI_Target		43,499	-8,410	89,239	10,627

Dari hasil statistika deskriptif, terlihat bahwa konsentrasi PM2.5 memiliki rata-rata 60,028 µg/m³ dengan simpangan baku 19,993, menunjukkan adanya variasi yang cukup besar. Nilai minimumnya bahkan mencapai −30,550, yang mengindikasikan kemungkinan adanya data anomali. Pola yang hampir serupa terlihat pada PM10, yang memiliki rata-rata 89,970 µg/m³ dengan rentang nilai yang sangat lebar dari −33,232 hingga 224,750. Simpangan baku yang tinggi (30,053) menegaskan bahwa variabel ini mengalami fluktuasi yang signifikan.

Pada polutan gas, NO_2 menunjukkan rata-rata 24,949 µg/m³ dengan variasi sedang, sementara SO_2 tampak lebih stabil dengan rata-rata hanya 10,002 µg/m³ dan

simpangan baku 3,995. Begitu pula CO, yang memiliki rata-rata 0,599 mg/m³ dan variasi yang relatif rendah, sehingga konsentrasinya cenderung lebih konsisten. Berbeda dengan itu, O_3 menunjukkan rentang nilai yang cukup lebar, dari 0,799 hingga 75,843 µg/m³, dengan rata-rata 40,987 µg/m³ dan simpangan baku 10,002, mengindikasikan fluktuasi moderat.

Variabel meteorologis juga menunjukkan karakteristik yang menarik. Suhu udara memiliki rata-rata 27,005°C dengan variasi moderat, sedangkan kelembapan memiliki simpangan baku yang cukup besar dan nilai minimum negatif, yang mengarah pada kemungkinan adanya noise dalam data. Untuk kecepatan angin, nilai rata-rata 2,503 m/s dengan simpangan baku 1,003 menunjukkan variasi yang relatif rendah. Sementara itu, tekanan udara tampak stabil dengan rata-rata 1009,995 hPa dan simpangan baku kecil sebesar 4,996.

Pada aspek curah hujan, Rain_mm memiliki rata-rata yang sangat rendah yaitu 0,094 mm, menunjukkan bahwa hujan jarang terjadi dan intensitasnya rendah. Terakhir, nilai AQI_Target menunjukkan rata-rata 43,499 yang termasuk dalam kategori baik, dengan fluktuasi moderat berdasarkan simpangan baku 10,627 dan rentang nilai yang cukup luas, sehingga mencerminkan variasi kualitas udara dari sangat baik hingga sedang selama periode pengamatan.

4.2 Uji Korelasi

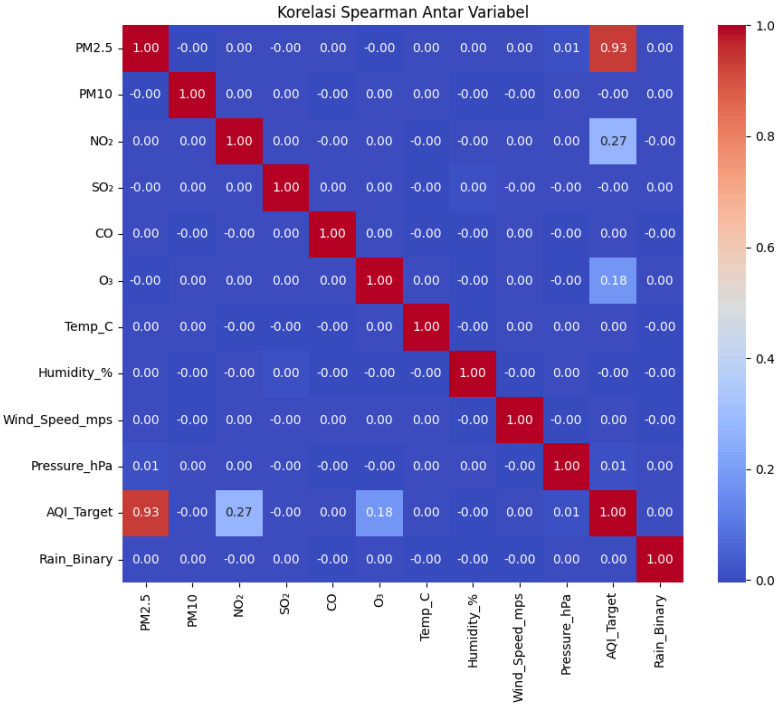
Uji korelasi dalam penelitian ini dilakukan enggunakan perintah Python. Sebelum melakukan uji korelasi, diperlukan uji normalitas untuk memastikan apakah data berdistribusi normal atau tidak. Hasil uji ini penting karena menentukan jenis metode korelasi yang akan digunakan. Berdasarkan hasil uji normalitas diperoleh temuan sebagai berikut:

Tabel 4.2 Uji Normalitas

Variabel	shapiro_pvalue	summary
PM2.5	0,67081	Normal
PM10	0,65060	Normal
NO_2	0,79390	Normal
SO_2	0,98487	Normal
CO	0,96245	Normal

Variabel	shapiro_pvalue	summary
O_3	0,82459	Normal
Temp_C	0,91387	Normal
Humidity_%	0,92174	Normal
Wind_Speed_mps	0,98587	Normal
Pressure_hPa	0,84599	Normal
AQI_Target	0,83260	Normal
Rain_Binary	0,0000	Not Normal

Dari temuan uji normalitas seluruh variabel sudah berdistribusi normal kecuali Rain_Binary (curah hujan), sehingga metode korelasi yang tepat adalah metode Spearman. Metode ini dipilih karena tidak membutuhkan asumsi normalitas dan tahan terhadap outlier. Metode Spearman menghitung tingkat keeratan hubungan berdasarkan peringkat masing-masing variabel sehingga lebih robust terhadap data jenis ini. Berikut adalah hasil dari uji korelasi menggunakan metode korelasi spearman:



Gambar 4.1 *Heatmap* Korelasi Antar Variabel

Hasil korelasi divisualisasikan menggunakan *heatmap* korelasi agar mempermudah untuk memahami pola hubungan. Semakin gelap warna mendekati warna merah maka korelasi antarvariabel tersebut sangat kuat. Hasil matriks korelasi

menunjukkan bahwa variabel yang terhubung ke *AQI_Target* memiliki beberapa hubungan yang signifikan. Hubungan paling kuat terdapat pada variabel PM2.5 dan *AQI_Target*, dengan nilai korelasi sebesar 0,93 dan arah hubungan positif. Nilai ini mengindikasikan bahwa peningkatan konsentrasi PM2.5 secara langsung diikuti oleh kenaikan AQI, sehingga kualitas udara memburuk seiring meningkatnya kadar partikulat halus tersebut. Temuan ini sejalan dengan pedoman WHO yang menyatakan bahwa PM2.5 merupakan indikator paling sensitif dalam menilai dampak kesehatan karena mampu menembus sawar paru-paru dan masuk ke sistem peredaran darah, sehingga meningkatkan risiko penyakit kardiovaskular, gangguan pernapasan, dan kanker [2].

Partikulat berukuran 2.5 mikrometer sangat memengaruhi AQI_Target, tetapi partikulat berukuran 10 mikrometer (PM10) memiliki korelasi yang sangat lemah dengan koefisien korelasi 0,00059, sehingga dalam data ini kontribusinya terhadap AQI hampir tidak terlihat. Perbedaan ini mengindikasikan bahwa partikulat berukuran lebih besar seperti PM10 memiliki pengaruh yang jauh lebih kecil terhadap fluktuasi AQI dibandingkan PM2.5 yang ukurannya jauh lebih halus dan lebih berbahaya bagi kesehatan.

Sementara itu, polutan gas seperti NO_2, SO_2, CO, dan O_3 memiliki korelasi yang lemah terhadap AQI_Target, dengan nilai korelasi yang umumnya berada di bawah 0,30. Variabel NO_2 tercatat memiliki korelasi tertinggi di antara polutan gas tersebut meskipun tetap berada pada kategori lemah, sedangkan variabel lainnya menunjukkan hubungan yang sangat kecil. Selain itu, korelasi negatif antara AQI_Target dan SO_2 mengindikasikan bahwa peningkatan AQI cenderung diikuti oleh penurunan konsentrasi SO_2. Temuan ini konsisten dengan literatur yang menyebutkan bahwa kondisi atmosfer di wilayah *urban* cenderung lebih tidak stabil akibat turbulensi dan pencampuran udara yang tinggi, sehingga konsentrasi SO_2 menjadi lebih mudah terdispersi [3]. Hal ini menyebabkan korelasi SO_2 terhadap AQI menjadi tidak signifikan.

Variabel meteorologis seperti suhu (Temp_C), kelembapan (Humidity_%), tekanan udara (Pressure_hPa), dan kecepatan angin (Wind_Direction_deg), dan Rain_binary (curah hujan) pada analisis ini menunjukkan korelasi yang sangat lemah atau tidak terbaca terhadap AQI_Target. Kondisi ini dapat dipahami karena parameter meteorologi umumnya berperan dalam proses dispersi, akumulasi, dan transformasi polutan di atmosfer, sehingga pengaruhnya terhadap nilai AQI tidak

selalu tercermin melalui hubungan linier sederhana. Literatur juga menunjukkan bahwa faktor meteorologi memiliki efek yang bersifat dinamis dan kompleks, sehingga korelasi terhadap konsentrasi polutan dapat menjadi rendah atau tidak signifikan [3]. Oleh karena itu, nilai koefisien korelasi yang mendekati nol mencerminkan tidak adanya hubungan linier yang berarti antara variabel meteorologis dan AQI_Target dalam konteks dataset ini.

Secara keseluruhan, polutan gas maupun variabel meteorologis tidak menunjukkan korelasi yang berarti dalam analisis ini. Korelasi antara polutan gas seperti NO_2, SO_2, CO, dan O_3 umumnya sangat rendah, demikian pula hubungan antara faktor meteorologis seperti suhu, kelembapan, kecepatan angin, dan tekanan udara. Kedua kelompok variabel tersebut juga tidak saling berkorelasi satu sama lain. Temuan ini menunjukkan bahwa hubungan antarvariabel dalam dataset bersifat lemah dan tidak linier, sehingga tidak dapat dijelaskan hanya melalui koefisien korelasi sederhana. Secara keseluruhan, satu-satunya hubungan yang terlihat jelas adalah korelasi antara variabel-variabel tertentu dengan AQI_Target, sesuai dengan fokus utama analisis yang ditujukan untuk mengamati kualitas udara.

4.3 Regresi Data Panel

Pada tahap ini, analisis dilanjutkan dengan menerapkan regresi data panel untuk mengetahui pengaruh variabel independen terhadap variabel dependen secara simultan. Pendekatan regresi panel dipilih karena mampu memadukan dimensi waktu dan individu sehingga memberikan estimasi yang lebih informatif dibandingkan model regresi biasa. Regresi data panel merupakan pengembangan dari analisis regresi yang merupakan gabungan dari data *time series* dan data *cross section* [27]. Analisis ini menjadi dasar untuk menentukan model yang paling sesuai sebelum dilakukan estimasi pada subbab berikutnya.

4.3.1 Estimasi Model Regresi

Estimasi regresi data panel dilakukan untuk menentukan model terbaik di antara Common Effect Model (CEM), Fixed Effect Model (FEM), dan Random Effect Model (REM). Pemilihan model dilakukan melalui Uji Chow dan Uji Hausman. Hasil pengujian disajikan pada Tabel 4.3.

Tabel 4.3 Hasil Uji Chow & Uji Hausman

	Cross-section F	**P-value**
Uji Chow	1,8189	0,1220
Uji Hausman	18,29756	0,00108

Pemilihan model regresi data panel dilakukan menggunakan Uji Chow dan Uji Hausman. Berdasarkan hasil Uji Chow, diperoleh nilai p-value sebesar 0,1220 (> 0,05) sehingga H_0 gagal ditolak, yang menunjukkan bahwa Common Effect Model lebih sesuai dibandingkan Fixed Effect Model. Selanjutnya, Uji Hausman digunakan untuk membandingkan Fixed Effect Model dan Random Effect Model, dengan hasil p-value sebesar 0,00108 (< 0,05) sehingga H_0 ditolak. Dengan demikian, dapat disimpulkan bahwa Fixed Effect Model (FEM) merupakan model yang paling sesuai. Oleh karena itu, FEM digunakan sebagai model regresi data panel dalam penelitian ini karena mampu mengakomodasi perbedaan karakteristik individu yang bersifat tetap sepanjang waktu. Berikut adalah hasil regresi dari model Fixed Effect :

Tabel 4.4 Hasil Regresi Model Fixed Effect

Variabel	**Kode**	**Coefficient**
Humidity_%	X_1	-0,00007228
PM2.5	X_2	0,5001
O_3	X_3	0,2006
Pressure_hPa	X_4	0,0008
CO	X_5	-0,0135

Model *fixed effect* dapat digambarkan dalam persamaan regresi berikut:

$\widehat{Y} = \alpha_i$ - 0,00007228 (X_1) + 0,5001 (X_2) + 0,2006 (X_3) + 0,0008 (X_4) - 0,0135(X_5)

4.3.2 Uji Asumsi Klasik

Sebelum model dinyatakan layak untuk digunakan ,maka harus memenuhi seluruh uji asumsi. Uji asumsi tersebut meliputi uji multikolinieritas,

uji heteroskedastisitas, uji normalitas, dan uji autokorelasi. Berikut adalah hasil dari rangkaian uji asumsi:

Tabel 4.5 Hasil Uji Asumsi Klasik

Jenis Uji	Metode/Statistik Uji	Nilai Statistik	p-value	Keputusan
Normalitas	Shapiro-Wilk	0,99998	0,7950	Gagal tolak H_0
Heteroskedastisitas	Breusch–Pagan	2,4895	0,7781	Gagal tolak H_0
Autokorelasi	Durbin–Watson	2,0034	–	Tidak terjadi autokorelasi
Multikolinearitas	Variance Inflation Factor (VIF)	VIF maks 1,000038	–	Tidak terjadi multikolinearitas

Berdasarkan hasil uji normalitas menggunakan Shapiro-Wilk, diperoleh nilai p-value sebesar 0,7950 > 0,05 yang menghasilkan keputusan gagal tolak H_0. Dengan begitu, residual data telah berdistribusi normal. Pada uji heteroskedastisitas menggunakan metode Breusch–Pagan, diperoleh nilai p-value sebesar 0,7781 > 0,05 yang menghasilkan keputusan gagal tolak H_0. Artinya, tidak terdapat gejala heteroskedastisitas dalam model regresi, sehingga varians error bersifat konstan atau homoskedastis dan model memenuhi asumsi klasik.

Setelah model dinyatakan memenuhi asumsi normalitas dan homoskedastisitas, selanjutnya dilakukan uji autokorelasi dan multikolinearitas. Berdasarkan uji autokorelasi menggunakan Durbin–Watson diperoleh nilai sebesar 2,0034, sehingga tidak terjadi autokorelasi karena nilai telah mendekati angka 2. Pada uji multikolinearitas, diperoleh nilai VIF maksimal sebesar 1,000038 yang artinya tidak terdapat multikolinearitas karena nilai VIF > 10.

4.3.3 Uji Hipotesis Regresi

Setelah model dinyatakan memenuhi seluruh asumsi, tahap berikutnya adalah pengujian hipotesis untuk melihat apakah setiap variabel independen berpengaruh signifikan terhadap variabel dependen. Untuk mengetahui apakah seluruh variabel berpengaruh secara simultan, diperlukan uji-F. Berikut adalah hasil dari uji F:

Tabel 4.6 Hasil Uji F-statistik

	F-test	**p-value**
F-statistic	407,800	0,0000

Berdasarkan hasil uji F, diperoleh nilai p-value sebesar 0,0000 < 0,05 dan F-test 407,800. Dengan demikian, keputusan yang diambil adalah tolak H_0, yang berarti bahwa variabel Humidity_%, PM2.5, O_3, Pressure_hPa, dan CO secara simultan terbukti berpengaruh signifikan terhadap AQI_Target.

Setelah dilakukan uji-F untuk mengetahui apakah variabel berpengaruh signifikan secara bersama-sama, selanjutnya dilakukan uji-t. Uji t-statistik dilakukan untuk melihat apakah variabel bebas secara parsial berpengaruh signifikan terhadap variabel terikat. Berikut adalah nilai probabilitas masing-masing variabel bebas:

Tabel 4.7 Hasil Uji t-statistik

Variabel	**Kode**	**p-value**
Humidity_%	X_1	0,8793
PM2.5	X_2	0,0000
O_3	X_3	0,0000
Pressure_hPa	X_4	0,5656
CO	X_5	0,7066

Dari hasil p-value pada uji t-statistik, diperoleh nilai probabilitas pada variabel Humidity (X_1) sebesar 0,8793 > 0,05 Dengan begitu, dapat diambil keputusan berupa gagal tolak H0 yang berarti Humidity tidak berpengaruh signifikan terhadap AQI.

Sementara itu pada variabel PM2.5 (X_2) diperoleh nilai probabilitas sebesar 0,0000 < 0,05. Dengan begitu dapat diambil keputusan berupa tolak H_0 yang berarti PM2.5 berpengaruh signifikan terhadap AQI. Kemudian pada variabel O_3 (X_3) diperoleh nilai p-value sebesar 0,0000 < 0,05. Dengan begitu, dapat diambil keputusan berupa tolak H_0 yang berarti O_3 berpengaruh signifikan terhadap AQI.

Pada variabel Pressure (X_4) diperoleh sebesar 0,5656 > 0,05. Dengan begitu, dapat diambil keputusan berupa gagal tolak H_0 yang berarti Pressure

tidak berpengaruh signifikan terhadap AQI. Sedangkan, hasil p-value pada uji t-statistik diperoleh nilai probabilitas pada variabel CO sebesar 0,7066 > 0,05. Dengan begitu, dapat diambil keputusan berupa gagal tolak H_0 yang berarti CO tidak berpengaruh signifikan terhadap AQI.

Setelah melakukan uji-F dan uji-t, perlu dilakukan uji koefisien determinasi untuk mengukur sejauh mana kemampuan model dalam menggambarkan variasi variabel dependen secara keseluruhan. Nilai koefisien determinasi yang digunakan dalam penelitian ini adalah R-squared (Within) sebesar 0,9209, yang menunjukkan bahwa sekitar 92,09% variasi AQI_Target dalam masing-masing entitas sepanjang waktu dapat dijelaskan oleh variabel Humidity_%, PM2.5, O_3, Pressure_hPa, dan CO. Sisanya sebesar 7,91% dipengaruhi oleh faktor lain di luar model.

4.3.4 Pembahasan Hasil Model Regresi

Dalam penelitian ini, variabel kelembaban (Humidity_%) menunjukkan pengaruh negatif dan tidak signifikan terhadap Indeks Kualitas Udara (AQI_Target) dengan koefisien sebesar −0,00007228. Hasil ini mengindikasikan bahwa perubahan tingkat kelembaban hanya memberikan pengaruh yang lemah terhadap kualitas udara. Kelembaban yang tinggi dapat mengurangi resuspensi polutan dari permukaan tanah sehingga partikel-partikel di udara cenderung lebih stabil dan tidak mudah terangkat. Temuan ini sejalan dengan penelitian Winda yang mengacu pada teori Lu et al., yang menyatakan bahwa pengaruh kelembapan bersifat tidak langsung dan dimediasi oleh faktor meteorologi lain seperti suhu, kecepatan angin, serta interaksi antar polutan [20]. Penelitian lain juga menyebutkan bahwa pada kondisi kelembaban tinggi, partikel polutan lebih sulit terdispersi sehingga efeknya terhadap kualitas udara menjadi negatif dan tidak signifikan [21]. Dengan demikian, kontribusi kelembaban terhadap variasi AQI relatif kecil dan tidak menjadi faktor dominan dalam menentukan kualitas udara.

Berbeda dengan kelembaban, variabel PM2.5 menunjukkan pengaruh positif dan signifikan terhadap AQI_Target dengan koefisien sebesar 0,5001. Hal ini menunjukkan bahwa peningkatan konsentrasi partikulat berukuran 2,5 mikrometer secara langsung berkaitan dengan peningkatan nilai AQI yang mencerminkan penurunan kualitas udara. PM2.5 juga memiliki koefisien

pengaruh tertinggi dibandingkan variabel lain yang digunakan dalam model, sehingga menjadi faktor paling dominan dalam memengaruhi kualitas udara. Temuan ini didukung oleh berbagai penelitian sebelumnya yang menyebutkan bahwa PM2.5 merupakan proksi umum polusi udara dan termasuk polutan paling berbahaya karena berkontribusi terhadap penyakit kardiovaskular, gangguan pernapasan, kanker, serta tingginya angka kematian global [1]. Penelitian oleh Willia juga menunjukkan bahwa PM2.5 menjadi variabel utama dalam menilai kualitas udara di DKI Jakarta [18]. Selain itu, studi lain menyebutkan bahwa kontribusi partikulat berukuran 2,5 mikrometer terhadap peningkatan polutan di udara lebih besar dibandingkan dengan partikel berukuran lain [18].

Sementara itu, variabel ozon (O_3) dalam penelitian ini menunjukkan pengaruh positif namun tidak signifikan terhadap AQI_Target dengan koefisien sebesar 0,2006. Hal ini mengindikasikan bahwa peningkatan kadar ozon cenderung diikuti oleh peningkatan nilai AQI, namun pengaruh tersebut relatif lemah. Temuan ini sejalan dengan penelitian Winda yang mengacu pada teori Lu et al., yang menyatakan bahwa pengaruh polutan gas seperti NO_2, O_3, dan CO bersifat tidak langsung dan sangat dipengaruhi oleh faktor meteorologi seperti suhu dan kecepatan angin [20]. Kondisi atmosfer di wilayah urban yang cenderung tidak stabil akibat turbulensi dan pencampuran udara yang tinggi menyebabkan kontribusi masing-masing polutan terhadap kualitas udara menjadi beragam dan tidak selalu signifikan secara individual [3].

Hasil penelitian juga menunjukkan bahwa tekanan udara (Pressure_hPa) memiliki pengaruh positif dan tidak signifikan terhadap AQI_Target dengan koefisien sebesar 0,0008. Hal ini menunjukkan bahwa peningkatan tekanan udara hanya memberikan dampak lemah terhadap peningkatan indeks kualitas udara. Temuan ini sejalan dengan studi berbasis data panel di Cina yang menyatakan bahwa peningkatan tekanan atmosfer berkorelasi positif dengan konsentrasi polutan udara, terutama ketika variabel meteorologi lain seperti kecepatan angin dan curah hujan dikendalikan [21]. Kondisi tekanan tinggi dapat menghambat dispersi polutan sehingga memungkinkan terjadinya akumulasi polutan di dekat permukaan. Namun demikian, pengaruh tekanan udara terhadap AQI tetap bersifat tidak

signifikan, yang menegaskan bahwa kualitas udara dipengaruhi oleh kombinasi berbagai faktor dan karakteristik wilayah penelitian.

Adapun variabel karbon monoksida (CO) menunjukkan pengaruh negatif dan tidak signifikan terhadap AQI_Target dengan koefisien sebesar −0,0135. Hasil ini mengindikasikan bahwa perubahan konsentrasi CO hanya memberikan pengaruh yang lemah terhadap kualitas udara. Temuan ini sejalan dengan literatur yang menyatakan bahwa variasi konsentrasi CO sangat dipengaruhi oleh faktor meteorologi seperti kelembaban, tekanan udara, dan suhu, yang berperan dalam proses pengenceran dan dispersi polutan di atmosfer. Selain itu, hubungan antara CO dan kualitas udara bersifat kontekstual, tergantung pada sumber emisi lokal, kondisi atmosfer, serta interaksi dengan polutan lain [22]. Dengan demikian, meskipun pengaruh CO terhadap AQI dalam penelitian ini tidak signifikan, perannya tetap relevan sebagai bagian dari sistem polusi udara yang kompleks dan saling berkaitan.

4.4 Analisis Deret Waktu

Uji *time series* dalam penelitian ini dilakukan untuk memahami pola perubahan nilai pada suatu variabel dari waktu ke waktu untuk menghasilkan peramalan pada periode berikutnya. Peramalan (*forecasting)* adalah suatu teknik analisis perhitungan untuk memperkirakan kejadian di masa yang akan datang dengan menggunakan pengalaman di masa lampau. Penelitian ini menggunakan data AQI harian dari Januari hingga Desember 2023 untuk mengamati dinamika kualitas udara. Pada penelitian ini, uji deret waktu dilakukan dengan menggunakan dua pendekatan metode peramalan, yaitu metode statistik tradisional (ARIMA) dan metode modern berbasis *machine learning* (GRU, LSTM, RNN, CNN, dan SVR). Berikut ini adalah hasil dari uji deret waktu dengan menggunakan ARIMA dan *machine learning*:

4.4.1 ARIMA

Sebelum menentukan model ARIMA diperlukan uji stasioneritas untuk menentukan apakah data sudah stasioner dalam *mean* dan varians. Hal ini digunakan untuk mengidentifikasi data perlu dilakukan *differencing* atau tidak. Selanjutnya, plot ACF dan

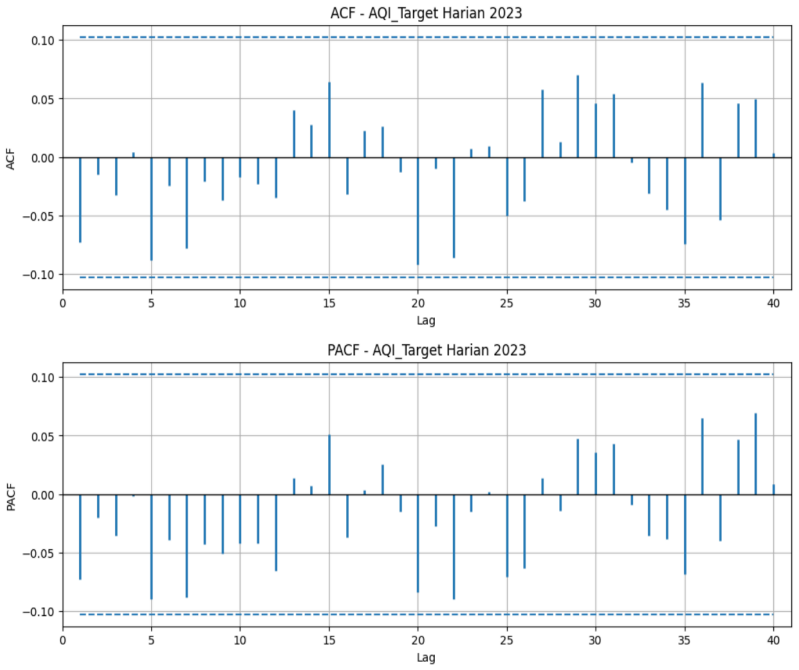
PACF pada uji *white noise* digunakan untuk menentukan AR (*Autoregressive*) dan MA (*Moving Average*) pada model ARIMA. Berikut adalah hasil uji stasioneritas dan *white noise*:

Tabel 4.8 Hasil Uji Stasioneritas

ADF	-20.517095071253582
p-value	0,0

Berdasarkan hasil uji stasioneritas, diperoleh p-value bernilai 0,0. Hal ini membuktikan bahwa data telah stasioner karena p-value bernilai <0,05. Selain itu, pada nilai ADF sebesar −20.517 yang jauh lebih kecil daripada nilai kritis pada tingkat signifikansi umum, sehingga mengindikasikan tidak adanya akar unit dalam data. Maka H_0 ditolak, yang berarti data telah stasioner dan layak untuk dianalisis menggunakan model *time series.* Sehingga, dapat diketahui bahwa data telah stasioner dan tidak perlu melakukan *differencing* (d=0).

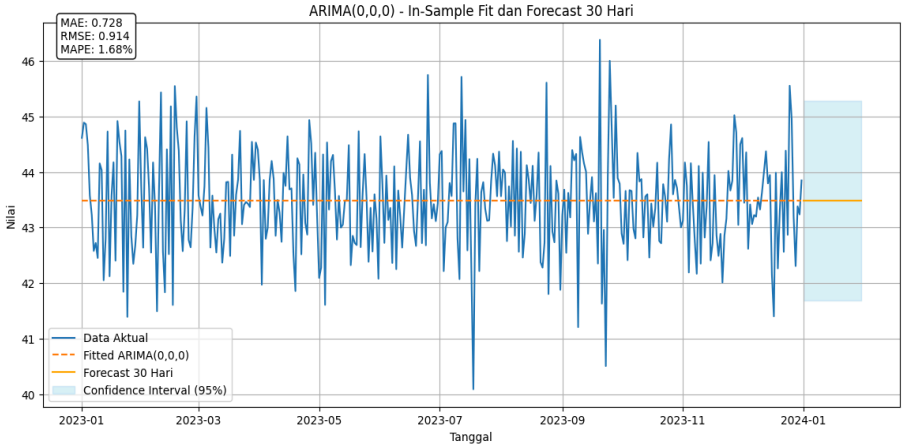
Setelah data dinyatakan stasioner berdasarkan uji ADF dan tidak memerlukan proses *differencing*, analisis dilanjutkan dengan pemeriksaan plot ACF dan PACF sebagai dasar dalam penentuan orde model ARIMA. Berikut adalah hasil plot ACF dan PACF.



Gambar 4.2 Plot ACF & PACF

Berdasarkan plot ACF dan PACF, terlihat bahwa seluruh *spike* berada dalam batas kepercayaan (*confidence interval*) yang berarti data tidak terdapat autokorelasi dan residual sudah bersifat *white noise.* Hal ini dibuktikan dari AR atau nilai p yang tidak menunjukkan adanya lag yang memotong (*cut-off)* di awal dan cenderung menyebar acak (*tailing*). Hal yang sama juga terjadi pada komponen MA atau nilai q yang juga tidak memiliki pola *cut-off* yang jelas pada *lag* awal. Sehingga, nilai p dan q sama-sama bernilai 0 yang artinya residual tidak memiliki pola tertentu dan bersifat acak (*white noise*).

Tidak ditemukannya pola *cut-off* yang jelas pada komponen AR dan MA menunjukkan bahwa nilai p dan q bernilai 0. Dikombinasikan dengan hasil uji stasioneritas yang menunjukkan d = 0, maka model ARIMA yang digunakan sebagai model awal adalah ARIMA(0,0,0). Berikut adalah tampilan grafik model ARIMA(0,0,0).

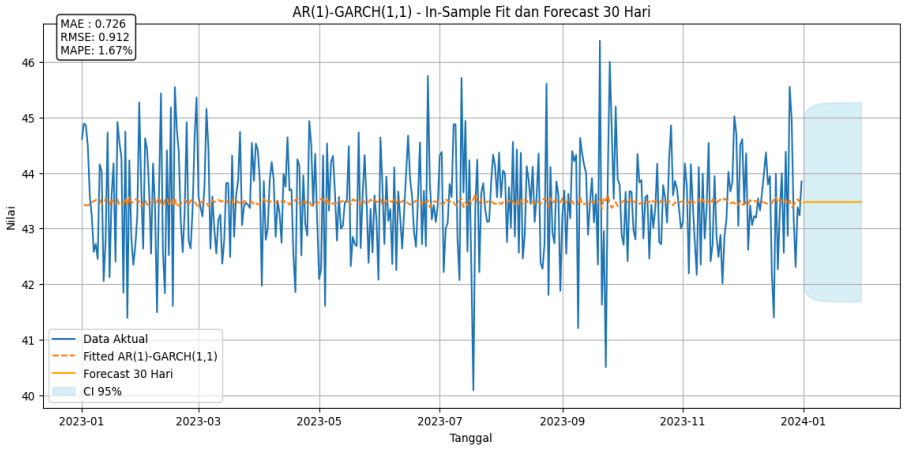


Gambar 4.3 Grafik Model ARIMA(0,0,0)

Gambar di atas merupakan visualisasi model ARIMA (0,0,0) yang terlihat bahwa garis *fitted* (oranye) maupun *forecast* 30 hari yang tetap mendatar, sementara *Confidence Interval* melebar tanpa memberikan arah prediksi yang jelas. Meskipun metrik error menunjukkan nilai MAE=0.728, RMSE=0.914, dan MAPE=1.68%, nilai tersebut relatif tinggi jika dibandingkan dengan rentang data yang berada di kisaran 40-46. Dengan RMSE hampir 1 poin, model memiliki penyimpangan yang cukup besar terhadap variabilitas data yang dinamis. Hal tersebut menunjukkan bahwa model dinilai tidak

cocok untuk data pada penelitian ini, sehingga analisis dilanjutkan menggunakan pendekatan AR-GARCH, yang lebih sesuai untuk data dengan fluktuasi dan volatilitas tidak stabil.

Dari model ARIMA (0,0,0) menghasilkan *fitted* dan *forecast* yang tampak datar karena hanya merepresentasikan nilai rata-rata historis, selanjutnya analisis dilakukan menggunakan model AR(1)-GARCH(1,1) yang mampu menangkap pola autokorelasi ringan serta dinamika volatilitas.



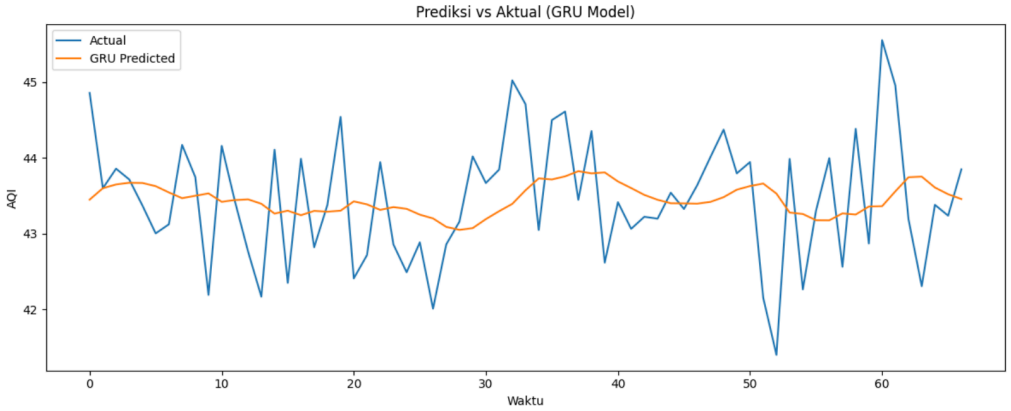
Gambar 4.4 Grafik Model AR(1)-GARCH(1,1)

Grafik AR(1)-GARCH(1,1) di atas terlihat bahwa garis *fitted* mulai mengikuti pola data historis meskipun sangat halus, sebab model AR(1) hanya menangkap hubungan linier sederhana antar waktu dan komponen GARCH mengestimasi volatilitas. Secara kuantitatif, model menghasilkan MAE 0.726, RMSE 0.912, dan MAPE 1.67% dimana nilai RMSE yang mendekati 1 pada skala data 40-46 tergolong cukup tinggi sehingga menunjukkan bahwa model belum mampu menangkap dinamika fluktuasi harian dengan baik. Pada *forecast* 30 hari ke depan, garis prediksi cenderung mendatar di sekitar nilai tengah data sebelumnya, dan rentang *Confidence Interval* yang lebar mengindikasikan ketidakpastian serta keterbatasan model dalam mengikuti perubahan pola yang lebih rumit. Berdasarkan keterbatasan tersebut, analisis kemudian dilanjutkan dengan pendekatan *machine learning* yang diharapkan dapat menangkap pola non-linier dan hubungan kompleks antarvariabel.

4.4.2 *Machine Learning Forecasting*

Setelah dilakukan analisis deret waktu menggunakan metode ARIMA dan AR-GARCH, hasil menunjukkan bahwa model kurang mampu merepresentasikan pola data. Hal ini disebabkan karena pola data tidak menunjukkan tren apapun, melainkan fluktuasi yang tidak konsisten. Oleh karena itu, analisis dilanjutkan dengan menggunakan model *machine learning*. Metode yang digunakan diantaranya adalah GRU, LSTM, RNN, CNN, dan SVR. Analisis pertama yang dilakukan adalah analisis berdasarkan visual plot dari kelima model *machine learning* tersebut. Analisis secara visual digunakan untuk melihat seberapa jauh model mampu mengikuti pola data asli, terutama pada bagian fluktuasi.

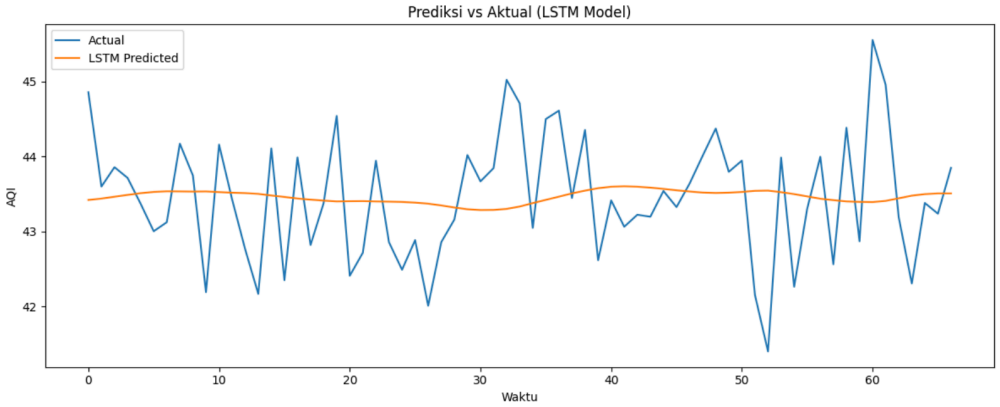
Berikut ini adalah plot data aktual dan prediksi menggunakan metode *machine learning* dengan model GRU (*Gated Recurrent Unit*)



Gambar 4.5 Grafik Model GRU

Pada plot GRU, terlihat bahwa garis prediksi tampak cukup halus dan kurang mengikuti lonjakan ekstrem yang ada pada data aktual. Meskipun begitu, garis prediksinya terlihat berdekatan dengan pola data aktual, yaitu dengan jarak antara prediksi dan nilai aktual yang relatif kecil. Sehingga, garis prediksi kurang lebih telah mampu menangkap tren umum data. Hal ini menunjukkan bahwa GRU mampu menjaga kestabilan prediksi dan tidak terlalu berpengaruh oleh *noise* dalam fluktuasi data.

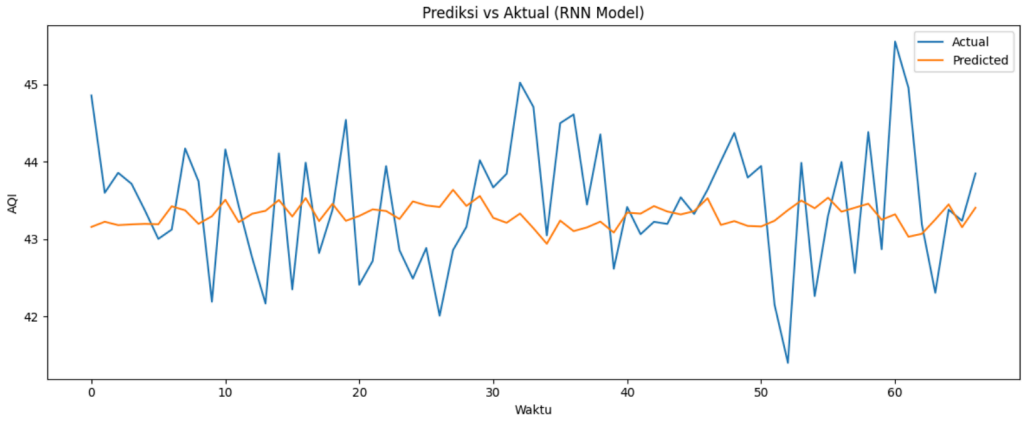
Berikut ini adalah plot data aktual dan prediksi menggunakan metode *machine learning* dengan model LSTM (*Long Short-Term Memory*)



Gambar 4.6 Grafik Model LSTM

Berdasarkan plot LSTM di atas, menunjukkan bahwa garis prediksi cenderung bergerak halus mengikuti tren atau pola utama data. Namun, terlihat bahwa kurva yang dihasilkan lebih datar dan kurang mampu menangkap perubahan nilai (fluktuasi data) jika dibandingkan dengan GRU. Pergerakan garis prediksi terlihat mempertahankan konsistensi pola umumnya sehingga terlihat *flat* atau datar. Dari plot di atas juga dapat diketahui bahwa model LSTM telah melakukan proses *smoothing* untuk mempertahankan konsistensi prediksi tepat di nilai rata-rata.

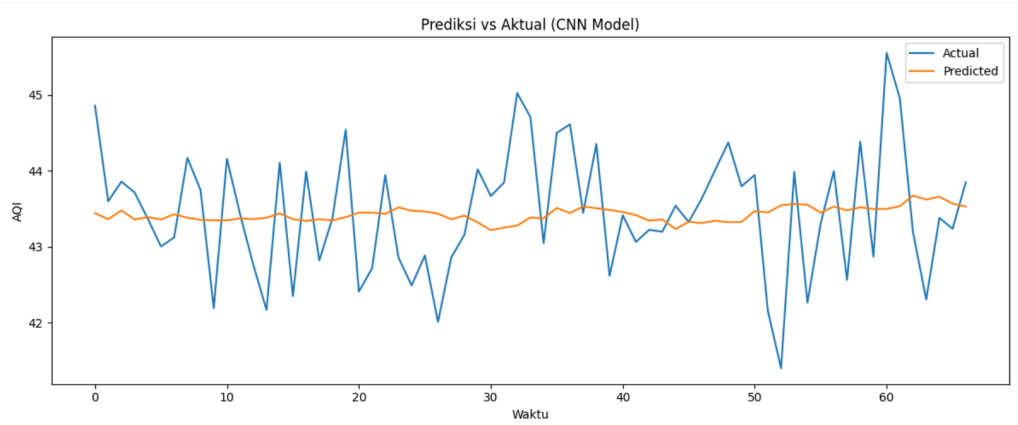
Berikut ini adalah plot data aktual dan prediksi menggunakan metode *machine learning* dengan model RNN (*Recurrent Neural Network*).



Gambar 4.7 Grafik Model RNN

Dari plot RNN di atas, garis prediksi terlihat mengikuti pola naik turunnya data aktual dengan sedikit lebih fleksibel dibandingkan dengan LSTM. Namun, pada beberapa bagian terlihat deviasi yang jelas dimana prediksi yang dihasilkan tidak setepat GRU dan LSTM. Kondisi ini terlihat pada pertengahan grafik (di titik waktu 25), yaitu ketika terjadi penurunan mendadak, RNN justru tidak menangkap pola prediksi tersebut.

Selain menggunakan model GRU, LSTM, dan RNN dilakukan juga pemodelan menggunakan CNN. Berikut ini adalah plot data aktual dan prediksi menggunakan metode *machine learning* dengan model CNN (*Convolutional Neural Network)*

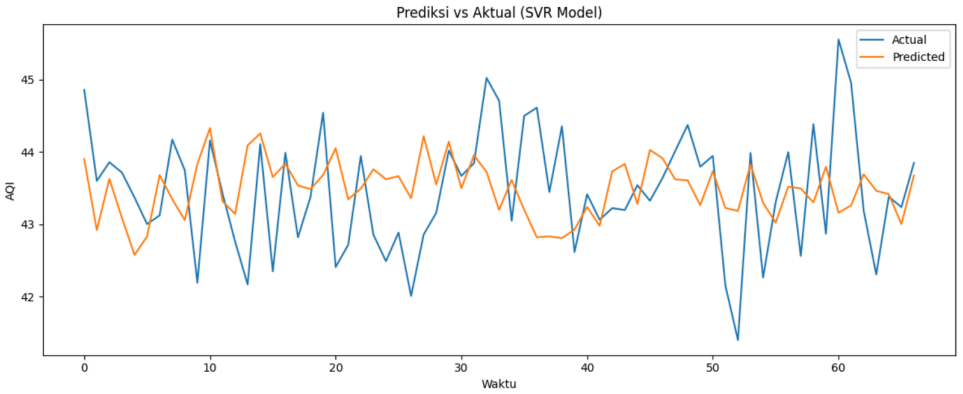


Gambar 4.8 Grafik Model CNN

Berdasarkan plot CNN di atas, terlihat garis atau kurva prediksi yang halus dan cenderung datar. Akibatnya, model yang dihasilkan kurang menangkap pola naik turun (fluktuasi) terhadap data aktualnya. Terdapat banyak titik perubahan data aktual yang tidak diikuti oleh garis prediksi CNN. Sehingga secara umum, model ini mampu menangkap pola dasar, namun tidak cukup tanggap terhadap volatilitas atau perubahan nilai ekstrem pada data *time series*. Hal ini terbukti pada titik waktu 52 yang saat itu mengalami penurunan ekstrem, model CNN justru tidak menangkap adanya prediksi yang menunjukkan penurunan juga.

Pada pemodelan terakhir, prediksi dilakukan menggunakan metode SVR. Berikut ini adalah plot data aktual dan prediksi

menggunakan metode *machine learning* dengan model SVR (*Support Vector Regression*)



Gambar 4.9 Grafik Model SVR

Pada model SVR, terlihat bahwa garis prediksi memiliki jarak paling jauh dari data aktual dibanding semua model lainnya. Fluktuasi data tidak tertangkap dengan baik dan prediksinya tampak lebih menyimpang pada mayoritas titik waktu. Meskipun secara kurva paling terlihat pola fluktuasinya, namun pola yang dihasilkan terdapat deviasi yang menyimpang. Misalnya pada titik kenaikan ekstrem di titik waktu 60, kurva prediksi justru tidak menangkap nilai yang sesuai dengan data aktualnya.

4.4.3 Estimasi Model Deret Waktu Terbaik

Dari seluruh analisis secara visual maka diperlukan diperlukan juga analisis berdasarkan evaluasi metrik pengukuran galat agar kemampuan prediksi tiap model dapat dibandingkan secara kuantitatif. Pada penelitian ini digunakan tiga metrik evaluasi, yaitu RMSE (*Root Mean Squared Error),* MAE *(Mean Absolute Error),* MAPE *(Mean Absolute Percentage Error).* Ketiga metrik tersebut berfungsi untuk menilai seberapa jauh nilai prediksi model bisa menyimpang dari data atau nilai aktual.

RMSE menghitung akar rata-rata kuadrat selisih prediksi dan aktual, MAE mengukur rata-rata selisih absolut, sedangkan MAPE menyatakan kesalahan dalam bentuk persentase. Semakin kecil nilai ketiga metrik tersebut, semakin tinggi tingkat akurasi model. Oleh karena itu, pemilihan model terbaik ditentukan berdasarkan konsistensi nilai galat yang paling rendah. Berikut hasil evaluasi dari seluruh model yang telah dianalisis:

Tabel 4.9 Hasil Evaluasi Metode Deret Waktu

	RMSE	**MAE**	**MAPE**
ARIMA	0,914	0,728	1,68%
AR-GARCH	0,912	0,726	1,67%
GRU	0,13353	0,10900	23,11%
LSTM	0,13418	0,10849	23,11%
RNN	0,13814	0,11106	22,64%
CNN	0,13393	0,10973	23,24%
SVR	0,14620	0,11730	24,95%

Berdasarkan hasil plot visualisasi data aktual dan prediksi serta penilaian metrik, dapat disimpulkan bahwa model GRU merupakan model dengan performa terbaik. Hal ini juga ditandai dengan nilai RMSE-nya yang tergolong kecil, yaitu 0,13353. Kondisi ini juga didukung dari model yang mampu menghasilkan prediksi yang paling dekat dengan nilai aktual dan cukup menunjukkan pola prediksi yang stabil serta mengikuti tren data meski tidak menangkap seluruh fluktuasi ekstrem. Salah satu penyebab utama dari prediksi yang kurang mampu menangkap fluktuasi data aktual adalah karena cakupan dataset yang digunakan cukup besar. Oleh karena itu, model GRU dirancang sebagai model yang lebih sederhana dari LSTM dengan fungsi aktivasi minimal, sehingga mampu mempercepat pengolahan data besar [14].

Hal ini didukung dengan penelitian yang telah dilakukan sebelumnya oleh Jasmine Kezia Halim et al. (2021), dimana metode GRU digunakan untuk memprediksi kualitas udara di Jakarta. Dalam penelitian tersebut, metode GRU dipilih karena mampu menangani data *time series* dengan baik serta memiliki komputasi yang lebih efisien dibandingkan metode lain seperti LSTM [17]. Evaluasi dilakukan menggunakan nilai MAPE dan RMSE. Hasilnya diperoleh bahwa metode GRU memberikan prediksi yang cukup akurat untuk sebagian besar parameter, kecuali NO_2 dengan MAPE lebih dari 50% [17].

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah diuraikan sebelumnya, dapat disimpulkan bahwa:

1. Berdasarkan hasil analisis korelasi, hubungan antar variabel yang paling kuat ada pada variabel PM2.5 dan *AQI_Target*, dengan nilai korelasi sebesar 0,93 dan arah hubungan positif. Hal ini menunjukkan bahwa partikulat berukuran 2.5 mikrometer menjadi kontributor utama dalam peningkatan AQI. Sementara itu, pada variabel polutan gas seperti NO_2, SO_2, CO, dan O_3 memiliki korelasi yang lemah terhadap AQI_Target, dengan nilai korelasi yang umumnya berada di bawah 0,30. Hal yang sama juga terdapat pada hubungan antara AQI dengan variabel meteorologis, seperti suhu (Temp_C), kelembapan (Humidity_%), tekanan udara (Pressure_hPa), kecepatan angin (Wind_Direction_deg), dan Rain_binary (curah hujan). Variabel meteorologis menunjukkan korelasi yang sangat lemah, sehingga pengaruhnya tidak langsung tercermin pada perubahan nilai AQI.
2. Berdasarkan hasil analisis regresi data panel, PM 2.5 menjadi faktor yang paling signifikan dalam memengaruhi AQI. Artinya, semakin buruk kualitas udara maka akan diikuti oleh kenaikan konsentrasi partikulat berukuran 2,5 mikrometer di udara. Sementara itu, pada faktor meteorologi dan polutan gas tidak memiliki pengaruh yang signifikan terhadap perubahan AQI.
3. Berdasarkan hasil analisis deret waktu dengan pendekatan statistik, diperoleh hasil bahwa performa dengan model ARIMA/AR-GARCH belum cukup baik untuk menangkap pola prediksi berdasarkan data aktual. Hal ini disebabkan karena pola data tidak menunjukkan tren apapun, melainkan fluktuasi yang tidak konsisten Sehingga, proses analisis deret waktu dilanjutkan menggunakan pendekatan *machine learning* yang diperoleh hasil bahwa performa dengan model GRU mampu menangkap pola prediksi berdasarkan data aktual dengan pola paling baik yang ditandai dengan nilai metrik evaluasi terkecil dari kelima metode *machine learning* lainnya.

5.2 Saran Pengembangan

Berdasarkan kesimpulan yang telah dipaparkan sebelumnya, saran dan pengembangan yang dapat diberikan adalah:

1. Pengendalian kualitas udara sebaiknya difokuskan pada pengurangan konsentrasi PM 2.5 yang telah terbukti menjadi faktor paling dominan yang memengaruhi peningkatan AQI dan semakin memperburuk kualitas udara.
2. Pengembangan sistem peramalan (*forecasting*) berbasis GRU (*Gated Recurrent Unit*) dapat membantu dalam proses pengambilan keputusan, kebijakan, dan mitigasi bagi masyarakat ketika kualitas udara sedang menurun. Mengingat bahwa model dengan metode GRU memiliki performa terbaik pada penelitian ini untuk menangkap pola prediksi.
3. Penelitian selanjutnya, disarankan untuk menambahkan variabel lingkungan lain yang mungkin memiliki kontribusi terhadap peningkatan AQI. Seperti jarak terhadap kawasan industri, radiasi matahari, dan faktor lainnya yang dapat memperdalam pemahaman faktor penyebab polusi.
4. Untuk memperkuat akurasi model, disarankan untuk melakukan eksperimen lanjutan pada metode *machine learning* lainnya, seperti *Hybird* ARIMA-GRU, *Hybird* LSTM-GRU, dan model *machine learning* lainnya. Selain itu, peningkatan kualitas data dengan penggunaan resolusi data waktu yang lebih tinggi juga dapat memungkinkan untuk meningkatkan performa model prediksi.

DAFTAR PUSTAKA

[1] World Health Organization, “Ambient (outdoor) air pollution,” Fact-sheet, 24 October 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health. Diakses: Nov. 29, 2025.

[2] C40 Knowledge Hub, “WHO air quality guidelines,” *C40 Knowledge Hub*. [Online].https://c40knowledgehub.org/s/article/WHO-Air-Quality-Guidelines?language=en_US. Diakses: Des. 4, 2025.

[3] A. T. Flores, “Atmospheric pollution in urban environments,” Ph.D. dissertation, Univ. of Aveiro, Aveiro, Portugal, 2020. [Online].https://pmc.ncbi.nlm.nih.gov/articles/PMC7697832/

[4] E. Marinov and D. Petrova-Arsus (D. P.-A.), “Time Series Forecasting of Air Quality: A Case Study of Sofia City,” *Atmosphere*, vol. 13, no. 5, 2022.https://www.mdpi.com/2073-4433/13/5/788

[5] I. Talamanova and S. Pllana, “Data-driven Real-time Short-term Prediction of Air Quality: Comparison of ES, ARIMA, and LSTM,” *arXiv preprint*, 2022.https://arxiv.org/abs/2211.09814

[6] A. S. Ma’arif, *Analisis Determinan Degradasi Lingkungan di Pulau Jawa Periode 2018-2023*, Skripsi, Fak. Ekonomi & Bisnis, UIN Syarif Hidayatullah Jakarta, 2025.https://repository.uinjkt.ac.id/dspace/handle/123456789/82606

[7] N. M. Pratasya, L. Q. Nada, R. T. Nanda, N. S. Harnida, U. Pratiwi, A. Mauluddin, S. Taskia dan V. Alpenita, “PENELITIAN KORELASI,” Jurnal Pendidikan Sosial dan Humaniora, vol. 4 , pp. 2644-2655, 2025. https://publisherqu.com/index.php/pediaqu/article/view/1951/1759

[8] D. Mustofani, “Penerapan Uji Korelasi Rank Spearman Untuk Mengetahui Hubungan Tingkat Pengetahuan Ibu Terhadap Tindakan Swamedikasi Dalam

Penanganan Demam Pada Anak," UJMC, vol. 9, no. 1, pp. Hal. 9-13, 2023. https://e-jurnal.unisda.ac.id/index.php/ujmc/article/view/4272

[9] H. A. Khoiri, Analisis Deret Waktu Univariat, Madiun: UNIPMA PRESS, 2023

[10] S. P. Fauzani dan D. Rahmi, "Penerapan Metode ARIMA Dalam Peramalan Harga Produksi," Jurnal Teknologi dan
Manajemen Industri Terapan, vol. Vol. 2, no. 4, pp. 269 - 277, 2023 : https://jurnal-tmit.com/index.php/home/article/view/283/75

[11] C. A., M. J. Prang and M. L. Mananohas, "Analisis Heteroskedastisitas Pada Data Cross Section dengan White Heteroscedasticity Test dan Weighted Least Squares," Jurnal Matematika, Vol. %1 dari %2 Vol. 4,, no. 2, pp. 173 - 179, 2015. https://ejournal.unsrat.ac.id/v3/index.php/decartesian/article/view/9056/8628

[12] C. Alkahfi, A. Kurnia dan A. Saefuddin, "Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan Keuangan Indonesia," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. Vol. 4, no. 4, pp. 1235-1243, 2024: https://www.journal.irpi.or.id/index.php/malcom/article/view/1415

[13] F. Sarie, B. Suhartawan, S. E. Priana, J. L. Marlina, M. W. Sari, F. Moniaga, A. M. Tri Haksami, M. Taufik dan B. Utomo, Pengantar Teknik Lingkungan, Padang: CV. Gita Lentera, 2024: https://books.google.co.id/books?hl=en&lr=&id=PMoSEQAAQBAJ&oi=fnd&pg=PA15&dq=Apa+itu+air+quality&ots=OKd5vx_GAx&sig=O4U5J1mML0h5IxlDpvArFlO4VMQ&redir_esc=y#v=onepage&q=Apa%20itu%20air%20quality&f=false

[14] F. Cela, "Studi Perbandingan Performa Metode GRU dan ARIMA-GRU untuk Prediksi Kualitas Udara di Kota Palembang," Skripsi, Fak. Teknik, Universitas Lampung, 2025. https://digilib.unila.ac.id/93089/

[15] detik.com, "Rumus standar deviasi: Pengertian, fungsi, jenis, dan contoh," *Detik Bali*, Nov. 15, 2022. [Online].

https://www.detik.com/bali/berita/d-6407981/rumus-standar-deviasi-pengertian-fungsi-jenis-dan-contoh. Diakses: 09 Des 2025.

[16] detik.com, "Pengertian mean, median, modus, dan cara menghitungnya," *DetikEdu*, Nov. 16, 2021. [Online]. https://www.detik.com/edu/detikpedia/d-5813307/pengertian-mean-median-modus-dan-cara-menghitungnya. Diakses: Des. 9, 2025.

[17] J.K. Halim, D.E. Herwindiati, and J.Hendryli, "Penerapan Gated Recurrent Unit untuk Prediksi Zat Pencemar Udara," Jurnal Ilmu Komputer dan Sistem Informasi, vol. 10, no. 2, 2022. doi: https://doi.org/10.24912/jiksi.v10i2.22540

[18] W. D. Puspitasari, "Peramalan Kualitas Udara di Jakarta Berdasarkan Jumlah Kendaraan Bermotor Menggunakan Metode Artificial Neural Network Backpropagation," Skripsi, Fak. Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim. http://etheses.uin-malang.ac.id/66029/2/200605110004.pdf

[19] K.Azmi, S. Defit, and Sumijan, "Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Batik Tanah Liat Sumatera Barat," Jurnal Unitek, vol.16, no. 1, pp. 28-40, 2023. https://doi.org/10.52072/unitek.v16i1.504

[21] I. F. N. Ilahi, E. Ferdiansyah, and F. Arifianto, "Pendugaan PM2.5 Menggunakan Metode Geographically Temporally Weighted Regression di DKI Jakarta," Jurnal Ilmu Lingkungan, vol. 22, no. 6, pp. 1435–1440, Nov. 2024. doi:10.14710/jil.22.6.1435-1440.

[22] Y. Liu, Y. Zhou, and J. Lu, "Exploring the relationship between air pollution and meteorological conditions in China under environmental governance," Scientific Reports, vol. 10, no. 1, p. 14518, 2020. doi:10.1038/s41598-020-71338-7

[23] A. E. Putra dan T. Rismawan, "Klasifikasi Kualitas Udara Berdasarkan Indeks Standar Pencemaran Udara (ISPU) Menggunakan Metode Fuzzy Tsukamoto," Jurnal Komputer dan Aplikasi, vol. 11, no. 02, pp. 190-196, 2023 : https://doi.org/10.26418/coding.v11i2.58704

[24] R. M. Sibarani, H. A. Belgaman, I. Athoillah dan S. Wirahma, “Analisis Hubungan Parameter Cuaca Terhadap Konsentrasi Polutan (PM2.5 dan CO) di Wilayah Jakarta Selama Periode Work From Home (WFH) Maret 2020,” Jurnal Sains & Teknologi Modifikasi Cuaca,, vol. Vol.22, no. 2, p. 85 – 94, 2021 : https://ejournal.brin.go.id/JSTMC/article/view/1187

[25] T. F. Prasetyo, A. Gani dan Z. Mufid, “Pengaruh Suhu, Kelembapan dan Angin terhadap Polusi Udara: Studi Kasus Dataset Air Quality,” Jurnal Inovasi dan Sains Teknik Elektro, vol. 6, no. 1, pp. 43-51, 2025 : https://jurnal.bsi.ac.id/index.php/insantek/article/view/8826/2180

[26] S. K. P. Dewi. “BAB III Metode Penelitian,” Repository STEI. http://repository.stei.ac.id/6090/3/BAB%20III.pdf. Diakses: Des. 10, 2025.

[27] Model Regresi Data Panel Untuk Mengetahui Faktor Yang Mempengaruhi Tingkat Kemiskinan Di Pulau Madura,” Jurnal Gaussian,, vol. 9, no. 3, pp. 355-363, 2020: https://ejournal3.undip.ac.id/index.php/gaussian/article/view/28925/24521

LAMPIRAN

Lampiran 1. Data Penelitian

	DateTime	PM2.5	PM10	NO_2	SO_2	CO	O_3	Temp_C	Humidity_%	Wind_Speed_mps	Pressure_hPa	Rain_mm	AQI_Target
0	01/01/2020 00:00	86.397.213	111.814.972	26.599.649	3.875.088	0.570793	42.903.768	31.491.409	45.704.988	3.114.026	1.012.641.964	1.0	59.759.255
1	01/01/2020 01:00	73.311.679	110.919.391	18.585.412	11.820.056	0.562121	15.127.661	24.191.965	41.544.655	3.480.094	1.011.779.447	0.0	45.256.996
2	01/01/2020 02:00	61.359.818	47.063.317	20.605.215	16.531.417	0.466105	32.752.213	30.719.383	73.849.227	4.949.460	1.004.118.484	0.0	43.411.916
3	01/01/2020 03:00	54.081.632	122.981.322	14.682.654	14.453.442	0.607025	18.910.033	18.243.150	42.086.443	1.834.147	1.009.154.244	0.0	35.227.619
4	01/01/2020 04:00	43.221.175	102.259.959	41.515.463	17.495.670	0.537119	39.582.884	33.949.777	42.393.921	1.987.593	1.021.100.094	0.0	41.981.803
5	01/01/2020 05:00	57.580.764	123.738.765	25.065.104	12.861.456	0.787644	26.450.625	24.403.035	67.301.615	2.677.779	1.010.270.384	0.0	41.600.038
6	01/01/2020 06:00	75.005.738	107.976.958	11.639.916	8.443.351	0.618616	24.656.870	30.186.860	63.996.762	2.669.073	1.007.996.453	0.0	45.926.218
7	01/01/2020 07:00	20.135.624	28.489.550	30.896.095	11.067.648	0.687732	31.344.728	28.072.837	79.535.351	2.944.619	1.015.204.547	0.0	25.605.586
8	01/01/2020 08:00	63.884.057	64.914.664	21.201.531	10.710.285	0.632864	28.935.811	20.059.076	74.546.143	3.334.228	1.018.218.452	1.0	44.089.650
9	01/01/2020 09:00	39.739.410	134.760.536	27.478.330	10.773.493	0.549189	26.820.304	30.441.334	49.678.391	4.125.432	1.017.037.632	0.0	33.477.265
…	…	…	…	…	…	…	…	…	…	…	…	…	…
1751 95	30/12/2023 15:00	42.040.144	117.529.046	22.997.444	13.947.681	0.564202	33.218.312	34.309.546	66.645.138	1.985.020	1.006.955.549	0.0	34.562.967
1751 96	30/12/2023 16:00	57.372.952	105.848.900	13.087.789	6.883.558	0.357771	26.871.932	27.896.880	41.838.197	1.781.676	1.007.344.358	0.0	37.987.199
1751 97	30/12/2023 17:00	46.013.729	109.997.919	1.622.097	14.100.733	0.345111	41.126.941	33.672.424	81.908.527	3.669.099	1.009.969.368	0.0	31.718.882
1751 98	30/12/2023 18:00	88.996.786	130.026.465	23.085.873	2.854.691	0.866649	37.740.997	34.177.297	56.785.010	1.871.766	1.014.315.659	0.0	58.972.355
1751	30/12/2023 19:00	54.989.674	92.933.3	22.931.932	14.364.418	0.418933	11.363.515	17.267.548	76.164.149	2.490.599	1.012.704.	0.2	36.647.119

99			50								773		
17 52 00	30/12/202 3 20:00	60.01 0.320	95.4 08.7 79	38.98 5.652	9.635 .815	0.455 887	37.39 0.524	31.18 8.920	65.47 2.012	2.839 .655	1.019 .407. 051	0.2	49.17 8.960
17 52 01	30/12/202 3 21:00	85.77 6.727	111. 944. 941	20.84 3.777	9.536 .969	0.764 993	38.50 6.328	21.09 4.983	59.38 4.117	3.346 .297	1.006 .211. 655	0.0	56.84 2.762
17 52 02	30/12/202 3 22:00	57.83 6.385	113. 044. 005	20.54 1.992	8.475 .458	0.401 075	52.26 3.306	29.75 5.230	67.45 0.861	2.802 .787	1.009 .150. 300	0.0	45.53 3.451
17 52 03	30/12/202 3 23:00	58.69 1.021	36.1 32.6 43	42.93 2.013	19.27 9.567	0.506 091	26.69 9.367	30.12 6.836	79.30 5.358	2.758 .214	1.016 .894. 754	0.0	47.56 4.988
17 52 04	31/12/202 3 00:00	111.4 87.30 2	93.4 35.7 27	40.46 0.490	18.33 7.540	1.000 .370	18.39 1.326	31.37 5.802	47.72 4.773	2.303 .228	1.009 .530. 461	0.0	71.56 0.063

Lampiran 2. Source Code

Lampiran 2.1 Source Code Korelasi

```
#korelasi spearman kolom numerik
import pandas as pd

num_cols = df.select_dtypes(include=['int64', 'float64']).columns
corr_spearman = df[num_cols].corr(method='spearman')

corr_spearman

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))
sns.heatmap(corr_spearman,      annot=True,      cmap='coolwarm',
fmt=".2f")
plt.title("Korelasi Spearman Antar Variabel")
plt.show()
```

Lampiran 2.2 Source Code ARIMA (0,0,0)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error,
mean_squared_error

# df_daily sudah didefinisikan sebelumnya

# ======================================
# 1. FIT MODEL ARIMA(0,0,0)
# ======================================
model_000 = ARIMA(df_daily, order=(0,0,0))
result_000 = model_000.fit()

print(result_000.summary())

# ======================================
# 2. PREDIKSI IN-SAMPLE (fit model)
# ======================================
fitted_values = result_000.fittedvalues

# ======================================
# 3. HITUNG MAPE, MAE, RMSE
# ======================================
actual = df_daily.values
predicted = fitted_values.values

mae = mean_absolute_error(actual, predicted)
rmse = np.sqrt(mean_squared_error(actual, predicted))
mape = np.mean(np.abs((actual - predicted) / actual)) * 100

print("====================================")
print(f"MAE  : {mae:.4f}")
print(f"RMSE : {rmse:.4f}")
print(f"MAPE : {mape:.2f}%")
print("====================================")

# ======================================
# 4. FORECAST 30 HARI KE DEPAN
# ======================================
n_forecast = 30
forecast_30 = result_000.get_forecast(steps=n_forecast)
forecast_mean = forecast_30.predicted_mean
ci = forecast_30.conf_int()
```

```
future_index = pd.date_range(
    start=df_daily.index[-1] + pd.Timedelta(days=1),
    periods=n_forecast,
    freq='D'
)

# ====================================
# 5. GRAFIK GABUNG: FITTED + FORECAST
# ====================================
plt.figure(figsize=(14,6))

# Data aktual
plt.plot(df_daily.index, df_daily.values, label='Data Aktual',
linewidth=1.5)

# Fitted (in-sample)
plt.plot(df_daily.index,       fitted_values,       label='Fitted
ARIMA(0,0,0)', linestyle='--')

# Forecast
plt.plot(future_index, forecast_mean, label='Forecast 30 Hari',
color='orange')

# Confidence interval
plt.fill_between(
    future_index,
    ci.iloc[:, 0],
    ci.iloc[:, 1],
    color='skyblue',
    alpha=0.3,
    label='Confidence Interval (95%)'
)

plt.title("ARIMA(0,0,0) - In-Sample Fit dan Forecast 30 Hari")
plt.xlabel("Tanggal")
plt.ylabel("Nilai")
plt.grid(True)
plt.legend()

# Tambahkan textbox MSE
plt.text(
    0.02, 0.92,
    f"MAE: {mae:.3f}\nRMSE: {rmse:.3f}\nMAPE: {mape:.2f}%",
```

```
    transform=plt.gca().transAxes,
    fontsize=10,
    bbox=dict(boxstyle="round", fc="white")
)

plt.show()
```

Lampiran 2.3 Source Code AR-GARCH

```
!pip install arch > /dev/null 2>&1

import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from       sklearn.metrics       import       mean_absolute_error,
mean_squared_error

# ====================================================
# 1. DATA
# ====================================================
series = df_daily.copy()

# ====================================================
# 2. FUNGSI AUTO-GARCH
# ====================================================
def auto_garch(data, p_range=3, q_range=3, ar_lags=[0,1]):
    best_model = None
    best_aic = np.inf
    best_params = None

    for l in ar_lags:
        for p in range(1, p_range+1):
            for q in range(1, q_range+1):
                try:
                    model = arch_model(
                        data,
                        vol="GARCH",
                        p=p, q=q,
                        mean="AR" if l > 0 else "Constant",
```

```
                        lags=l,
                        dist="normal"
                    )
                    result = model.fit(disp="off")
                    if result.aic < best_aic:
                        best_aic = result.aic
                        best_model = result
                        best_params = (l, p, q)
                except:
                    pass
    return best_model, best_params, best_aic

# ====================================================
# 3. FITTING MODEL TERBAIK
# ====================================================
best_model, best_params, best_aic = auto_garch(series)

print("Model Terbaik (AR lags, p, q):", best_params)
print("AIC :", best_aic)
print(best_model.summary())  # <-- tetap tampil!

# ====================================================
# 4. HITUNG METRIK FITTING
# ====================================================
actual = series.values
fitted = series - best_model.resid

aligned_actual = actual[~np.isnan(fitted)]
aligned_fitted = fitted.dropna().values

mae = mean_absolute_error(aligned_actual, aligned_fitted)
rmse         =         np.sqrt(mean_squared_error(aligned_actual,
aligned_fitted))
mape   =   np.mean(np.abs((aligned_actual   -   aligned_fitted)   /
aligned_actual)) * 100

print("\n==============================")
print(f"MAE  : {mae:.4f}")
print(f"RMSE : {rmse:.4f}")
print(f"MAPE : {mape:.2f}%")
print("==============================")

# ====================================================
```

```
# 5. FORECAST 30 HARI
# ======================================================
fc_horizon = 30
fc = best_model.forecast(horizon=fc_horizon)

mean_fc = fc.mean.squeeze().values
var_fc = fc.variance.squeeze().values
std_fc = np.sqrt(var_fc)

future_index = pd.date_range(
    start=series.index[-1] + pd.Timedelta(days=1),
    periods=fc_horizon,
    freq='D'
)

# ======================================================
# 6. GABUNGKAN FITTED + FORECAST DALAM 1 GRAFIK
# ======================================================
plt.figure(figsize=(13,5))

# Actual → biru
plt.plot(series.index, actual, label="Actual", color="tab:blue")

# Fitted → oranye dashed
plt.plot(series.index,   fitted,   label="Fitted   (AUTO-GARCH)",
linestyle="--", color="tab:orange")

# Forecast → oranye solid
plt.plot(future_index,   mean_fc,   label="Forecast   30   Hari",
color="tab:orange")

# Confidence Interval → abu-abu
plt.fill_between(
    future_index,
    mean_fc - 1.96*std_fc,
    mean_fc + 1.96*std_fc,
    color='gray',
    alpha=0.3,
    label='95% CI'
)

plt.title("AUTO-GARCH — In-Sample Fit dan Forecast 30 Hari")
plt.xlabel("Tanggal")
```

```
plt.ylabel("AQI")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.4 Source Code GRU

```
import numpy as np
import pandas as pd # Import pandas if df_daily is a pandas Series/DataFrame
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GRU
from tensorflow.keras.callbacks import EarlyStopping

# ==== PRE-PROCESSING (Copied from previous relevant cells) ====
# Ensure df_daily is available. If not, it needs to be loaded/created.
# Assuming df_daily is a pandas Series/DataFrame from previous executions (e.g., cell 9g_hSnvD1v8t)

# Assuming df_daily exists and is the output of `df_daily = df_2023['AQI_Target'].resample('D').mean()`
data = df_daily.values.reshape(-1, 1)

# Normalisasi
scaler = MinMaxScaler()
scaled = scaler.fit_transform(data)

# Windowing sequence
def make_sequence(dataset, window=30):
    X, y = [], []
    for i in range(window, len(dataset)):
        X.append(dataset[i-window:i, 0])
        y.append(dataset[i, 0])
    return np.array(X), np.array(y)
```

```
window = 30
X, y = make_sequence(scaled, window)

# GRU input (samples, timesteps, features)
X_gru = X.reshape((X.shape[0], X.shape[1], 1))

# Train-Test split 80:20
split = int(len(X_gru) * 0.8)
X_train, X_test = X_gru[:split], X_gru[split:]
y_train, y_test = y[:split], y[split:]

# ======================================================
# 1. MODEL GRU
# ======================================================

es      =      EarlyStopping(monitor="val_loss",      patience=5,
restore_best_weights=True)

model_gru = Sequential([
    GRU(64, return_sequences=True, input_shape=(window, 1)),
    GRU(32),
    Dense(1)
])

model_gru.compile(optimizer="adam", loss="mse")

history_gru = model_gru.fit(
    X_train, y_train,
    epochs=30,
    batch_size=16,
    validation_data=(X_test, y_test),
    callbacks=[es],
    verbose=1
)

# EVALUASI
pred_gru = model_gru.predict(X_test)

rmse_gru = np.sqrt(mean_squared_error(y_test, pred_gru))
mae_gru = mean_absolute_error(y_test, pred_gru)
mape_gru  =  np.mean(np.abs((y_test  -  pred_gru.reshape(-1))  /
y_test + 1e-8)) * 100
```

```
print("\n=== METRIK EVALUASI GRU ===")
print(f"RMSE : {rmse_gru:.5f}")
print(f"MAE  : {mae_gru:.5f}")
print(f"MAPE : {mape_gru:.2f}%")

# ==== VISUALISASI ====
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
pred_gru_inv = scaler.inverse_transform(pred_gru)

plt.figure(figsize=(12, 5))
plt.plot(y_test_inv, label='Actual')
plt.plot(pred_gru_inv, label='GRU Predicted')
plt.title("Prediksi vs Aktual (GRU Model)")
plt.xlabel("Waktu")
plt.ylabel("AQI")
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.5 Source Code LSTM

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.callbacks import EarlyStopping

# ==== MODEL LSTM ====

es      =      EarlyStopping(monitor="val_loss",      patience=5,
restore_best_weights=True)

model_lstm = Sequential([
    LSTM(64, return_sequences=True, input_shape=(window, 1)),
    LSTM(32),
    Dense(1)
])

model_lstm.compile(optimizer="adam", loss="mse")

history_lstm = model_lstm.fit(
    X_train, y_train,
    epochs=30,
    batch_size=16,
    validation_data=(X_test, y_test),
```

```
    callbacks=[es],
    verbose=1
)

# ==== EVALUASI ====
pred_lstm = model_lstm.predict(X_test)

rmse_lstm = np.sqrt(mean_squared_error(y_test, pred_lstm))
mae_lstm = mean_absolute_error(y_test, pred_lstm)
mape_lstm = np.mean(np.abs((y_test - pred_lstm.reshape(-1)) /
y_test + 1e-8)) * 100

print("\n=== METRIK EVALUASI LSTM ===")
print(f"RMSE : {rmse_lstm:.5f}")
print(f"MAE  : {mae_lstm:.5f}")
print(f"MAPE : {mape_lstm:.2f}%")

# ==== VISUALISASI ====
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
pred_lstm_inv = scaler.inverse_transform(pred_lstm)

plt.figure(figsize=(12, 5))
plt.plot(y_test_inv, label='Actual')
plt.plot(pred_lstm_inv, label='LSTM Predicted')
plt.title("Prediksi vs Aktual (LSTM Model)")
plt.xlabel("Waktu")
plt.ylabel("AQI")
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.6 Source Code RNN

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error,
mean_absolute_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, SimpleRNN
from tensorflow.keras.callbacks import EarlyStopping
```

```
# ==== PRE-PROCESSING ====
data = df_daily.values.reshape(-1, 1)

# Normalisasi\scaler = MinMaxScaler()
scaled = scaler.fit_transform(data)

# Windowing sequence
def make_sequence(dataset, window=30):
    X, y = [], []
    for i in range(window, len(dataset)):
        X.append(dataset[i - window:i, 0])
        y.append(dataset[i, 0])
    return np.array(X), np.array(y)

window = 30
X, y = make_sequence(scaled, window)
X_rnn = X.reshape((X.shape[0], X.shape[1], 1))

# Split data train-test
split = int(len(X_rnn) * 0.8)
X_train, X_test = X_rnn[:split], X_rnn[split:]
y_train, y_test = y[:split], y[split:]

# ==== MODEL RNN ====
es      =      EarlyStopping(monitor="val_loss",      patience=5,
restore_best_weights=True)

model_rnn = Sequential([
       SimpleRNN(64,  return_sequences=True,  input_shape=(window,
1)),
    SimpleRNN(32),
    Dense(1)
])

model_rnn.compile(optimizer="adam", loss="mse")

history = model_rnn.fit(
    X_train, y_train,
    epochs=30,
    batch_size=16,
    validation_data=(X_test, y_test),
    callbacks=[es],
```

```
    verbose=1
)

# ==== EVALUASI ====
pred_rnn = model_rnn.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, pred_rnn))
mae = mean_absolute_error(y_test, pred_rnn)
mape = np.mean(np.abs((y_test - pred_rnn.reshape(-1)) / y_test +
1e-8)) * 100

print("\n=== METRIK EVALUASI RNN ===")
print(f"RMSE : {rmse:.5f}")
print(f"MAE  : {mae:.5f}")
print(f"MAPE : {mape:.2f}%")

# ==== VISUALISASI ====
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
pred_inv = scaler.inverse_transform(pred_rnn)

plt.figure(figsize=(12, 5))
plt.plot(y_test_inv, label='Actual')
plt.plot(pred_inv, label='Predicted')
plt.title("Prediksi vs Aktual (RNN Model)")
plt.xlabel("Waktu")
plt.ylabel("AQI")
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.7 Source Code RNN

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error,
mean_absolute_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, MaxPooling1D,
Flatten
from tensorflow.keras.callbacks import EarlyStopping
```

```
# ==== PRE-PROCESSING ====
data = df_daily.values.reshape(-1, 1)

# Normalisasi
scaler = MinMaxScaler()
scaled = scaler.fit_transform(data)

# Windowing sequence
def make_sequence(dataset, window=30):
    X, y = [], []
    for i in range(window, len(dataset)):
        X.append(dataset[i - window:i, 0])
        y.append(dataset[i, 0])
    return np.array(X), np.array(y)

window = 30
X, y = make_sequence(scaled, window)
X_cnn = X.reshape((X.shape[0], X.shape[1], 1))   # tetap 3D

# Split data train-test
split = int(len(X_cnn) * 0.8)
X_train, X_test = X_cnn[:split], X_cnn[split:]
y_train, y_test = y[:split], y[split:]

# ==== MODEL CNN (pengganti RNN) ====
es      =      EarlyStopping(monitor="val_loss",      patience=5,
restore_best_weights=True)

model_cnn = Sequential([
          Conv1D(filters=64,  kernel_size=3,  activation='relu',
input_shape=(window, 1)),
    MaxPooling1D(pool_size=2),
    Flatten(),
    Dense(32, activation='relu'),
    Dense(1)
])

model_cnn.compile(optimizer="adam", loss="mse")

history = model_cnn.fit(
    X_train, y_train,
    epochs=30,
    batch_size=16,
```

```
    validation_data=(X_test, y_test),
    callbacks=[es],
    verbose=1
)

# ==== EVALUASI ====
pred_cnn = model_cnn.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, pred_cnn))
mae = mean_absolute_error(y_test, pred_cnn)

# tambahkan epsilon agar tidak divide by zero
mape = np.mean(np.abs((y_test - pred_cnn.reshape(-1)) / (y_test +
1e-8))) * 100

print("\n=== METRIK EVALUASI CNN ===")
print(f"RMSE : {rmse:.5f}")
print(f"MAE  : {mae:.5f}")
print(f"MAPE : {mape:.2f}%")

# ==== VISUALISASI ====
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
pred_inv = scaler.inverse_transform(pred_cnn)

plt.figure(figsize=(12, 5))
plt.plot(y_test_inv, label='Actual')
plt.plot(pred_inv, label='Predicted')
plt.title("Prediksi vs Aktual (CNN Model)")
plt.xlabel("Waktu")
plt.ylabel("AQI")
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.8 Source Code SVR

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR
```

```
from sklearn.metrics import mean_squared_error,
mean_absolute_error

# ==== PRE-PROCESSING ====
data = df_daily.values.reshape(-1, 1)

# Normalisasi
scaler = MinMaxScaler()
scaled = scaler.fit_transform(data)

# Windowing sequence
def make_sequence(dataset, window=30):
    X, y = [], []
    for i in range(window, len(dataset)):
        X.append(dataset[i - window:i, 0])
        y.append(dataset[i, 0])
    return np.array(X), np.array(y)

window = 30
X, y = make_sequence(scaled, window)          # output shape
(n,window)
# SVR menerima input 2D, jadi tidak perlu reshape seperti RNN/CNN
# X tetap (samples, window_features)

# Split data train-test
split = int(len(X) * 0.8)
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

# ===== MODEL SVR =====
svr_model = SVR(kernel='rbf', C=100, gamma='scale', epsilon=0.01)
# (C, epsilon boleh di-tuning untuk hasil lebih baik)

svr_model.fit(X_train, y_train)

# ==== PREDIKSI & EVALUASI ====
pred_svr = svr_model.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, pred_svr))
mae = mean_absolute_error(y_test, pred_svr)
mape = np.mean(np.abs((y_test - pred_svr) / (y_test + 1e-8))) *
100
```

```
print("\n=== METRIK EVALUASI SVR ===")
print(f"RMSE : {rmse:.5f}")
print(f"MAE  : {mae:.5f}")
print(f"MAPE : {mape:.2f}%")

# ==== VISUALISASI ====
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
pred_inv = scaler.inverse_transform(pred_svr.reshape(-1, 1))

plt.figure(figsize=(12, 5))
plt.plot(y_test_inv, label='Actual')
plt.plot(pred_inv, label='Predicted')
plt.title("Prediksi vs Aktual (SVR Model)")
plt.xlabel("Waktu")
plt.ylabel("AQI")
plt.legend()
plt.tight_layout()
plt.show()
```

Lampiran 2.9 Source Code Regresi Data Panel

import pandas as pd

```
import numpy as np
import statsmodels.api as sm
from statsmodels.stats.diagnostic import het_breuschpagan
from          statsmodels.stats.outliers_influence          import
variance_inflation_factor
from scipy import stats

!pip install linearmodels
from linearmodels.panel import PanelOLS, RandomEffects

# ================================
# 1. LOAD & PERSIAPAN DATA
# ================================
panel_data = pd.read_csv('/content/UrbanAirPollutionDataset.csv')

# Rename kolom yang mengandung karakter khusus
panel_data = panel_data.rename(columns={
    'PM2.5': 'PM25',
    'NO₂': 'NO2',
    'O₃': 'O3'
})
```

```
# Convert datetime
panel_data['DateTime'] = pd.to_datetime(panel_data['DateTime'])

# Set panel index
panel_data = panel_data.set_index(['Station_ID', 'DateTime'])

# ================================
# 2. MODEL REGRESI PANEL
# ================================

# ---- POOLED OLS ----
pooled_model = sm.OLS.from_formula(
     'AQI_Target ~ Q("Humidity_%") + PM25 + O3 + Pressure_hPa +
CO',
    data=panel_data
)
pool_results = pooled_model.fit()
print("\n=== POOLED OLS ===")
print(pool_results.summary())

# ---- FIXED EFFECT ----
fixed_model = PanelOLS.from_formula(
    'AQI_Target ~ Q("Humidity_%") + PM25 + O3 + Pressure_hPa + CO
+ EntityEffects',
    data=panel_data
)
fixed_results = fixed_model.fit()
print("\n=== FIXED EFFECTS ===")
print(fixed_results.summary)

# ---- RANDOM EFFECT ----
#  Removed 'CO' to resolve ZeroDivisionError (neffects == nvar
issue)
random_model = RandomEffects.from_formula(
    'AQI_Target ~ Q("Humidity_%") + PM25 + O3 + Pressure_hPa',
    data=panel_data
)
random_results = random_model.fit()
print("\n=== RANDOM EFFECTS ===")
print(random_results.summary)

# ================================
```

```
# 3. UJI ASUMSI KLASIK
# ================================

# ---- Breusch-Pagan ----
bp_test = het_breuschpagan(pool_results.resid, pooled_model.exog)
print("\n=== BREUSCH-PAGAN ===")
print(bp_test)

# ---- Chow Test ----
print("\n=== CHOW TEST ===")
print(f"F-statistic: {fixed_results.f_pooled.stat:.4f}")
print(f"P-value: {fixed_results.f_pooled.pval:.4f}")

# ---- Hausman Test ----
fe_params = fixed_results.params
re_params = random_results.params

# Define common_coef based on the variables included in both
models
# Note: fixed_model includes 'CO', random_model does not.
# So, common_coef will only include variables present in both.
common_coef                                                    =
list(set(re_params.index).intersection(fe_params.index))

b_FE = fe_params[common_coef]
b_RE = re_params[common_coef]

cov_FE = fixed_results.cov.loc[common_coef, common_coef]
cov_RE = random_results.cov.loc[common_coef, common_coef]

diff = b_FE - b_RE
cov_diff = cov_FE - cov_RE

H = np.dot(np.dot(diff.T, np.linalg.inv(cov_diff)), diff)
df = len(diff)
pval = 1 - stats.chi2.cdf(H, df)

print("\n=== HAUSMAN TEST ===")
print("Statistik Hausman :", H)
print("df                :", df)
print("p-value           :", pval)

# ---- Shapiro Normality ----
```

```
resid_clean = pool_results.resid.dropna()
normal_test = stats.shapiro(resid_clean)
print("\n=== SHAPIRO NORMALITY ===")
print(normal_test)

# ---- Durbin-Watson ----
dw_test = sm.stats.durbin_watson(pool_results.resid)
print("\n=== DURBIN-WATSON ===")
print(dw_test)

# ================================
# 4. UJI MULTIKOLINEARITAS (VIF)
# ================================
X  =  panel_data[['Humidity_%',  'PM25',  'O3',  'Pressure_hPa',
'CO']].dropna()
X = sm.add_constant(X, prepend=False)

vif_data = pd.DataFrame()
vif_data["Variable"] = X.columns
vif_data["VIF"] = [
    variance_inflation_factor(X.values, i)
    for i in range(X.shape[1])
]

print("\n=== VIF ===")
print(vif_data)
```