

# Prediksi dan *Forecasting* Kualitas Udara Menggunakan Algoritma *Kernel Extreme Learning Machine* (Studi Kasus: ISPU DKI Jakarta)

1<sup>st</sup> Sultan Chisson Obie  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

[sultanchissonobie@student.telkomuniversity.ac.id](mailto:sultanchissonobie@student.telkomuniversity.ac.id)

2<sup>nd</sup> Meta Kallista  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

[metakallista@telkomuniversity.ac.id](mailto:metakallista@telkomuniversity.ac.id)

3<sup>rd</sup> Ig. Prasetya Dwi Wibawa  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

[prasdwibawa@telkomuniversity.ac.id](mailto:prasdwibawa@telkomuniversity.ac.id)

**Abstrak** — Polusi udara adalah salah satu komponen yang paling berdampak pada kesehatan manusia. Ibu kota Indonesia, Jakarta, menduduki peringkat ke-9 untuk kualitas udara dan polusi kota. Informasi tentang kualitas udara pada suatu daerah tentu dibutuhkan manusia, seperti daerah DKI Jakarta. Masyarakat perlu mengetahui informasi tentang kualitas udara agar lebih peduli terhadap pengaruh polusi udara terhadap kesehatannya. Penelitian ini bertujuan untuk menyediakan informasi tentang kualitas udara di DKI Jakarta. Pada penelitian ini akan dilakukan prediksi dan peramalan pada data kualitas udara di DKI Jakarta. Metode yang digunakan untuk proses tersebut adalah *Kernel Extreme Learning Machine* (K-ELM). Metode ini memungkinkan pemanfaatan fungsi kernel yang dapat memberikan performa dan stabilitas yang tinggi. Berdasarkan hasil pengujian yang telah dilakukan, diperoleh nilai evaluasi metrik untuk proses prediksi dengan nilai *Root Mean Square Error* (RMSE) paling optimal sebesar 0.041, dan *R-Squared* paling optimal sebesar 0.898. Nilai evaluasi metrik tersebut didapat menggunakan fungsi *kernel Laplacian* dengan nilai *gamma* sebesar 0.01 dan nilai koefisien (C) sebesar 30. Untuk proses *forecasting*, didapat hasil terbaik dengan nilai evaluasi metrik *Root Mean Square Error* (RMSE) paling optimal sebesar. Hasil tersebut didapat dengan menggunakan *kernel Laplacian* dengan parameter *gamma* sebesar 0.01 dan nilai koefisien (C) sebesar 300.

**Kata kunci**— *Evaluasi Metrik, Kernel Extreme Learning Machine, Kualitas Udara, Peramalan, Prediksi*

## I. PENDAHULUAN

Polusi udara adalah salah satu komponen yang paling berdampak pada kesehatan manusia, seperti konsentrasi  $PM_{10}$  dan  $PM_{2.5}$  yang dapat membahayakan manusia. Oleh karena itu, informasi tentang kualitas udara sangat penting untuk melindungi kesehatan manusia dan mengendalikan polusi udara[1]. Menurut data dari IQAir, pada bulan Oktober 2021, Ibu kota Indonesia, Jakarta, menduduki peringkat ke-9 untuk kualitas udara dan polusi kota. Indonesia menempati peringkat ke-9 dari 106 negara untuk negara paling berpolusi di dunia pada tahun 2020 dari konsentrasi  $PM_{2.5}$ [2].

DKI Jakarta merupakan kota yang memiliki kepadatan kendaraan bermotor yang tinggi. 70% polutan dihasilkan oleh kendaraan bermotor. Ini karena kendaraan bermotor

mengeluarkan banyak zat yang sangat berbahaya dan merusak kesehatan manusia, termasuk merusak lingkungan[3].

Salah satu bagian dari *Artificial Intelligence* (AI) adalah *Machine Learning* (ML). Tujuan *machine learning* adalah untuk mempelajari data—data yang sudah ada sehingga mesin dapat membuat keputusan secara mandiri tanpa bantuan manusia lagi. *Machine Learning* dapat melakukan analisis regresi dimana proses ini dapat digunakan untuk mengevaluasi bagaimana variabel penelitian berkorelasi satu sama lain. Analisis regresi membantu dalam membuat prediksi, estimasi, dan pendugaan serta mengidentifikasi variabel yang mempengaruhi variabel dependen. Dalam makalah ini, kami mengusulkan salah satu metode *Machine Learning* untuk melakukan prediksi dan peramalan yaitu, *Kernel Extreme Learning Machine* (K-ELM).

*Kernel Extreme Learning Machine* (K-ELM) merupakan salah satu perkembangan metode *Extreme Learning Machine* (ELM). Dengan performa yang tinggi, ELM tetap memiliki kekurangan pada hasil prediksi yang tidak stabil[4]. Jumlah *hidden neuron* juga memengaruhi hasil akurasi. Hasil penelitian[5] didapatkan bahwa K-ELM mampu memberikan hasil prediksi lebih akurat dibandingkan ELM. K-ELM menunjukkan kemampuan untuk menangani masalah variasi yang disebabkan oleh bobot dan bias acak yang ditetapkan pada ELM, namun tetap mempertahankan kecepatan proses pembelajaran[6].

Berdasarkan tinjauan pustaka yang telah dikaji, memberikan kesempatan untuk mempelajari metode KELM dalam proses prediksi dan *forecasting*, sehingga makalah ini berisi penelitian untuk mendapatkan nilai prediksi dan peramalan pada data kualitas udara di DKI Jakarta. Penelitian ini melakukan perhitungan akurasi dengan MSE, RMSE dan *Mean Absolute Error* (MAE).

## II. KAJIAN TEORI

### A. Kualitas Udara

Menurut data dari IQAir, pada bulan Oktober 2021, Ibu kota Indonesia, Jakarta, menduduki peringkat ke-9 untuk kualitas udara dan polusi kota. Indonesia menempati

peringkat ke-9 dari 106 negara untuk negara paling berpolusi di dunia pada tahun 2020 dari konsentrasi  $PM_{2.5}$ [2]. DKI Jakarta merupakan kota yang memiliki kepadatan kendaraan bermotor yang tinggi. 70% polutan dihasilkan oleh kendaraan bermotor. Ini karena kendaraan bermotor mengeluarkan banyak zat yang sangat berbahaya dan merusak kesehatan manusia, termasuk merusak lingkungan [3].

### B. Regresi

Proses regresi dalam *machine learning* adalah suatu metode yang digunakan untuk mensimulasikan bagaimana variabel dependen, atau output, berhubungan dengan salah satu atau lebih variabel independen, atau input. Tujuan utamanya adalah untuk mengembangkan model yang memiliki kemampuan untuk memprediksi nilai output berdasarkan nilai-nilai input yang diberikan. Proses regresi berfokus pada data yang bersifat kontinu, seperti prediksi harga rumah berdasarkan fitur-fitur tertentu atau prediksi penjualan berdasarkan faktor-faktor seperti iklan, cuaca, dan harga produk. Selain itu regresi juga digunakan untuk melakukan proses *forecasting* atau ramalan. Pada makalah ini akan dilakukan proses regresi untuk mendapatkan hasil prediksi dan *forecasting* menggunakan algoritma *machine learning* dengan metode *Kernel Extreme Learning Machine* (K-ELM).

### C. Kernel Extreme Learning Machine

*Extreme Learning Machine* [7]. Dirancang sebagai jaringan saraf tiruan feed-forward dengan satu lapisan tersembunyi, juga disebut sebagai jaringan saraf tiruan *feed-forward* dengan satu lapisan tersembunyi (SLFN). Hal ini menunjukkan bahwa kecepatan pembelajarannya jauh lebih cepat dibandingkan dengan algoritma lainnya seperti SVM. Ringkasan proses ELM diilustrasikan sebagai berikut. Diberikan  $N$  sampel pelatihan  $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$  dimana  $\mathbf{x}_i$  adalah vektor input dan  $\mathbf{t}_i$  adalah vektor target yang sesuai, output dari jaringan SLFN dengan  $L$  node tersembunyi dapat diekspresikan sebagai berikut:

$$f_L(\mathbf{x}_j) = \sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i), \quad j = 1, \dots, N. \quad (1)$$

Hal ini dapat dituliskan sebagai  $f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}$ . Dimana,  $\mathbf{w}_i$  dan  $b_i$  masing – masing menyatakan bobot input dan bias ke lapisan tersembunyi,  $\beta_i$  adalah bobot output yang menghubungkan node tersembunyi ke- $i$  lapisan output dan  $g(\cdot)$  adalah fungsi aktivasi node tersembunyi[8].

Dalam kernel ELM yang baru dikembangkan, sebuah koefisien regularisasi positif dimasukkan ke dalam sistem pembelajaran agar lebih stabil. Asumsikan  $H^T H$  adalah nonsingular, koefisien  $1/C$  ditambahkan ke diagonal  $H^T H$  dalam perhitungan bobot output  $\boldsymbol{\beta}$ . Solusi yang dihasilkan lebih stabil dan dengan kinerja generalisasi yang lebih baik, kita dapat memiliki  $\boldsymbol{\beta} = H^T \left( \frac{1}{C} + HH^T \right)^{-1} \mathbf{T}$ . Fungsi yang sesuai dengan ELM yang diregularisasi adalah:

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} = \mathbf{h}(\mathbf{x})H^T \left( \frac{1}{C} + HH^T \right)^{-1} \mathbf{T} \quad (2)$$

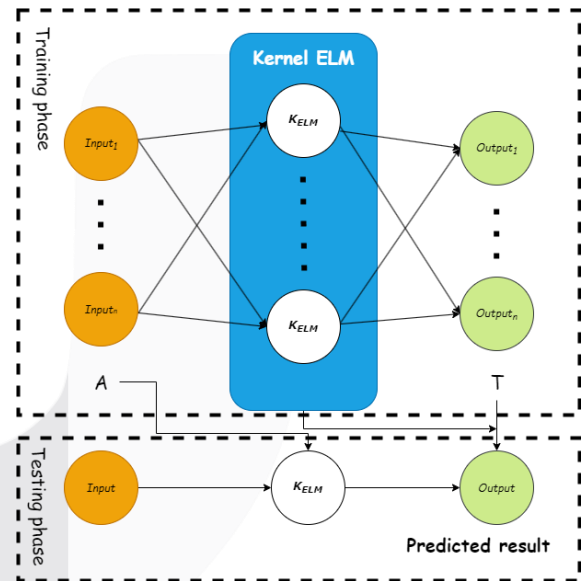
Dapat dilihat pada ELM bahwa matriks kernel untuk ELM dapat didefinisikan sebagai berikut. Misalkan  $\boldsymbol{\Omega}_{ELM} = HH^T$ ,  $\boldsymbol{\Omega}_{ELM_{i,j}} = h(\mathbf{x}_i)h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$ . Fungsi *output* ELM dapat ditulis secara ringkas sebagai:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{h}(\mathbf{x})H^T \left( \frac{1}{C} + HH^T \right)^{-1} \mathbf{T} \\ &= \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T \left( \frac{1}{C} + \boldsymbol{\Omega}_{ELM} \right)^{-1} \mathbf{T} \end{aligned} \quad (3)$$

Dalam implementasi ini, pemetaan fitur lapisan tersembunyi  $\mathbf{h}(\mathbf{x})$  tidak perlu diketahui oleh pengguna, sebagai gantinya kernel  $K(\mathbf{x}, \mathbf{x}_1)$  yang sesuai dapat dihitung[8].

## III. METODE

*Kernel Extreme Learning Machine* (K-ELM) merupakan salah satu perkembangan metode *Extreme Learning Machine* (ELM). Diferensiasi yang signifikan antara metode K-ELM dengan ELM terkait dengan penggunaan fungsi *kernel* di *hidden layer*-nya. Dibandingkan dengan ELM, pemetaan fitur lapisan tersembunyi tidak diperlukan dan jumlah neuron tersembunyi tidak dipilih[9]. Berikut adalah arsitektur *Kernel Extreme Learning Machine* (K-ELM).



GAMBAR 1  
Arsitektur K-ELM

K-ELM meningkatkan stabilitas ELM. Tahapan – tahapan pada metode K-ELM dijelaskan pada sub bab dibawah.

### A. Preprocessing Data

Sebelum masuk kedalam algoritma *Kernel Extreme Learning Machine* (K-ELM) dilakukan proses *preprocessing* data dan normalisasi data. Preprocessing yang dilakukan adalah *impute missing value* menggunakan metode KNN *Imputer* dari library *scikit-learn*. Imputer ini digunakan selain imputer lain dikarenakan imputer ini adalah yang paling efektif dan akurat dibandingkan dengan metode yang lain. Setelah dilakukan proses impute, data perlu disamakan skalanya agar dapat diproses menggunakan algoritma *machine learning* dengan baik. Proses ini disebut dengan

normalisasi. Metode normalisasi yang dilakukan pada makalah ini adalah *Min-Max Scaling*.

#### B. Pembagian dataset (*train* dan *test*)

Metode pembagian dataset pada makalah ini adalah *K-Fold Cross Validation*. Pembagian dataset dengan metode ini dengan membagi dataset asli secara acak menjadi *K* subset atau "*fold*" dengan ukuran sama. Nilai *K* yang dipilih pada makalah ini adalah 5. Hal ini berarti dataset dilatih dan dievaluasi sebanyak 5 kali. Presentase data *train* dibanding data *tes* adalah 80:20.



GAMBAR 2  
Diagram K-Fold Cross Validation

#### C. Training

Berikut langkah – langkah proses *training* metode K-ELM.

1. Inisialisasi kernel, nilai koefisien (*C*), serta nilai *gamma* ( $\lambda$ ) yang mengontrol bentuk fungsi kernel.
2. Perhitungan fungsi *kernel* untuk membentuk matriks omega data *training* ( $\Omega_{KELM}$ ). Bentuk matriks dapat dilihat pada persamaan (4).

$$\Omega_{KELM} = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_N) \end{bmatrix}_{N \times N} \quad (4)$$

Keterangan:

$N$  = jumlah data training

$K(x_i, x_j)$  = fungsi *kernel* data ke-*i* dan data ke-*j*

Selanjutnya dilanjutkan dengan perhitungan fungsi *kernel*. Makalah ini menerapkan fungsi *kernel* RBF, *Laplacian*, *Linear*, *Sigmoid*, dan *Cosine* yang diformulasikan pada persamaan (12), persamaan (13), persamaan (14), persamaan (15), dan persamaan (16).

3. Menghitung nilai *output weight* ( $\beta$ ) dengan persamaan (5).

$$\beta = \left( \frac{I}{C} + \Omega_{KELM} \right)^{-1} t \quad (5)$$

Keterangan:

$\Omega_{KELM}$  = matriks omega data *training*

$\left( \frac{1}{\lambda} + \Omega_{KELM} \right)^{-1}$  = operasi *inverse* dari matriks

$I$  = matriks identitas

$C$  = regularisasi parameter / koefisien

$t$  = target data *training*

#### D. Testing

Tujuan dari proses ini adalah untuk mengevaluasi penerapan teknik KELM berdasarkan proses pelatihan pada tahap sebelumnya. Berikut langkah – langkah proses *testing* metode K-ELM.

1. Mengambil nilai *output weight* dari proses pelatihan.
2. Perhitungan fungsi *kernel* adalah membentuk matriks omega data *testing* ( $\Omega_{KELM}$ ). Bentuk matriks dapat dilihat pada persamaan (6).

$$\Omega_{KELM} = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}_{N \times N}^T$$

$$\Omega_{KELM} = [K(x, x_1) \quad \cdots \quad K(x, x_N)]_{1 \times N} \quad (6)$$

Keterangan:

$K(x, x_1)$  = fungsi *kernel* pasangan data *testing* dan data *training* ke-*j*.  $j = 1, 2, \dots, N$ .

$T$  = *transpose* dari matriks

3. Menghitung nilai target *output* dengan persamaan (7).

$$\hat{y} = \Omega_{KELM} \cdot \beta \quad (7)$$

Keterangan:

$\hat{y}$  = target *output*

$\Omega_{KELM}$  = matriks omega data *testing*

$\beta$  = matriks *output weight*

#### E. Evaluasi

Hasil prediksi dapat dinilai melalui evaluasi. Pada makalah ini, evaluasi dilakukan dengan menggunakan model *metric evaluation* MSE, RMSE, MAE, dan R-Squared. Nilai evaluasi optimal yang diharapkan adalah mendekati 0 untuk (*Mean Square Error*) MSE, (*Root Mean Square Error*) RMSE, dan (*Mean Absolute Error*) MAE. Sedangkan nilai evaluasi optimal untuk R-Squared adalah nilai yang mendekati 1. Persamaan (8) sampai (11) digunakan dalam perhitungan *metrics evaluation*:

1. (*Mean Square Error*) MSE

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_{pred} - y_{true})^2 \quad (8)$$

Keterangan:

MSE = nilai *Mean Square Error* yang ingin dihitung.

$n$  adalah jumlah total sampel data.

$\Sigma$  = simbol sigma yang menunjukkan operasi penjumlahan.

$y_{pred}$  = nilai prediksi.

$y_{true}$  = nilai observasi atau sebenarnya.

2. (*Root Mean Square Error*) RMSE

$$RMSE = \sqrt{\frac{1}{n} \sum (y_{pred} - y_{true}) \times 2} \quad (9)$$

Keterangan:

RMSE = nilai *Root Mean Square Error* yang ingin dihitung.

$n$  = jumlah total sampel data.

$\Sigma$  = simbol sigma yang menunjukkan operasi penjumlahan.

$y_{pred}$  = nilai prediksi.

$y_{true}$  = nilai observasi atau sebenarnya.

### 3. (Mean Absolute Error) MAE

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{pred} - y_{true}| \quad (10)$$

Keterangan:

MAE = nilai *Mean Absolute Error* yang ingin dihitung.

$n$  = jumlah total sampel data.

$\Sigma$  = simbol sigma yang menunjukkan operasi penjumlahan.

$y_{pred}$  = nilai prediksi.

$y_{true}$  = nilai observasi atau sebenarnya.

### 4. R-Squared

$$R^2 = \frac{SSR}{SST} = \frac{\sum (y_{pred} - y_{true})^2}{\sum (y_{pred} - y_{true})^2} \quad (11)$$

Keterangan:

$R^2$  = nilai *R-Squared* yang ingin dihitung.

SSR = singkatan dari *Sum of Squares of Residuals* atau Jumlah Kuadrat Residu.

SST = singkatan dari *Total Sum of Squares* atau Jumlah Kuadrat Total.

$n$  = jumlah total sampel data.

$\Sigma$  = simbol sigma yang menunjukkan operasi penjumlahan.

$y_{pred}$  = nilai prediksi.

$y_{true}$  = nilai observasi atau sebenarnya.

### F. Fungsi Kernel

Fungsi *kernel* dapat menghitung *inner product* secara langsung pada ruang dimensi lebih tinggi (ruang fitur) tanpa perlu memahami fungsi pemetaan dari ruang input ke ruang fitur. Dengan kata lain, fungsi kernel adalah fungsi produk dalam pada ruang fitur. Pemilihan *kernel* yang tepat akan meningkatkan performa *machine learning*[6]. Kernel RBF, *Laplacian*, *Linear*, *Sigmoid*, dan *Cosine* adalah beberapa fungsi kernel yang populer. Pada penelitian ini, digunakan 5 fungsi kernel tersebut. Fungsi kernel yang dipilih adalah sebagai berikut.

#### 1. (Radial Basic Function) RBF Kernel

$$K(x_i, x_j)_{rbf} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (12)$$

Keterangan:

$\sigma$  adalah parameter yang mengontrol seberapa cepat penurunan nilai kernel dengan meningkatnya jarak antara vector  $x_i$  dan vector  $x_j$ .

#### 2. Laplacian Kernel

$$K(x_i, x_j)_{lin} = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right) \quad (13)$$

Keterangan:

$\sigma$  adalah parameter yang mengontrol seberapa cepat penurunan nilai kernel dengan meningkatnya jarak antara vector  $x_i$  dan vector  $x_j$ .

#### 3. Linear Kernel

$$K(x_i, x_j)_{lin} = (x_i \cdot x_j) \quad (14)$$

#### 4. Sigmoid Kernel

$$K(x_i, x_j)_{sig} = \tanh(\alpha(x_i \cdot x_j) + c) \quad (15)$$

#### 5. Cosine Kernel

$$K(x_i, x_j)_{cos} = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} \quad (16)$$

### G. Parameter

Gamma adalah parameter yang mengontrol bentuk fungsi kernel yang digunakan dalam pemetaan data ke ruang fitur yang lebih tinggi. Nilai gamma yang digunakan pada proses prediksi menggunakan *Kernel Extreme Learning Machine* adalah 0.0001, 0.001 dan 0.01. Sedangkan untuk proses *forecasting* digunakan nilai gamma 0.01, 0.02, dan 0.03.

Koefisien (C) merupakan parameter yang dapat mengontrol kompleksitas model dan mengatasi overfitting. Nilai koefisien (C) yang digunakan pada proses prediksi menggunakan *Kernel Extreme Learning Machine* adalah 10, 20, dan 30. Sedangkan untuk proses *forecasting* digunakan nilai koefisien (C) 100, 200, dan 300.

## IV. HASIL DAN PEMBAHASAN

Pada penelitian ini memiliki hasil akhir berupa nilai evaluasi matriks proses prediksi dan *forecasting*. Pada algoritma prediksi menggunakan metode K-ELM digunakan beberapa parameter, yaitu kernel untuk memetakan data input kedalam ruang fitur yang lebih tinggi, *regularization* parameter atau C berfungsi untuk mengendalikan penalti terhadap kesalahan model pada fase pelatihan, *gamma* parameter yang mengontrol bentuk fungsi kernel yang digunakan dalam pemetaan data ke ruang fitur yang lebih tinggi.

Dari hasil pengujian yang dilakukan, ketiga parameter tersebut mempengaruhi hasil dari model *machine learning* prediksi menggunakan K-ELM. Hal tersebut dapat ditinjau dari nilai metrik evaluasi pada hasil prediksi K-ELM. Dari hasil pengujian prediksi K-ELM, kernel paling baik yang bisa digunakan untuk dataset ISPU adalah kernel *Laplacian*. Berikut adalah hasil terbaik dari setiap partikel.

### A. Prediksi

#### 1. PM<sub>10</sub> dari DKI4 dengan kernel Laplacian

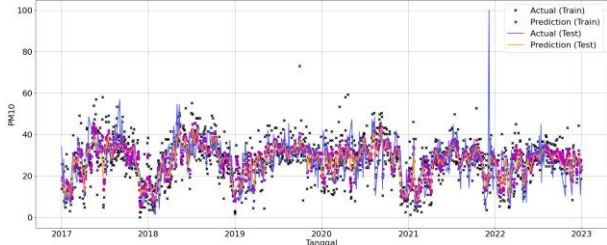
TABEL 1  
Data Train PM10 DKI4

| Data Train Evaluation |       |       |                |
|-----------------------|-------|-------|----------------|
| RMSE                  | MSE   | MAE   | R <sup>2</sup> |
| 0.045                 | 0.002 | 0.034 | 0.789          |

TABEL 2  
Data Test PM10 DKI4

| Data Test Evaluation |       |       |                |
|----------------------|-------|-------|----------------|
| RMSE                 | MSE   | MAE   | R <sup>2</sup> |
| 0.071                | 0.005 | 0.053 | 0.488          |

Berikut adalah plot data dari PM10 DKI4 dengan kernel Laplacian.



Gambar 3  
PM10 DKI4

2. SO<sub>2</sub> dari DKI3 dengan kernel Laplacian

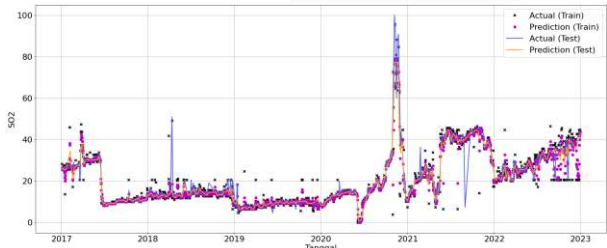
Tabel 3  
Data Train SO2 DKI3

| Data Train Evaluation |       |       |                |
|-----------------------|-------|-------|----------------|
| RMSE                  | MSE   | MAE   | R <sup>2</sup> |
| 0.022                 | 0.001 | 0.011 | 0.969          |

Tabel 4  
Data Test SO2 DKI3

| Data Test Evaluation |       |       |                |
|----------------------|-------|-------|----------------|
| RMSE                 | MSE   | MAE   | R <sup>2</sup> |
| 0.041                | 0.002 | 0.019 | 0.898          |

Berikut adalah plot data dari SO<sub>2</sub> DKI3 dengan kernel Laplacian.



GAMBAR 4  
SO2 DKI3

3. CO dari DKI3 dengan kernel Laplacian

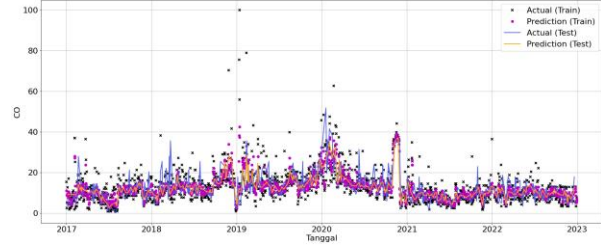
TABEL 5  
Data Train CO DKI3

| Data Train Evaluation |       |       |                |
|-----------------------|-------|-------|----------------|
| RMSE                  | MSE   | MAE   | R <sup>2</sup> |
| 0.040                 | 0.002 | 0.024 | 0.750          |

TABEL 6  
Data Test CO DKI3

| Data Test Evaluation |       |       |                |
|----------------------|-------|-------|----------------|
| RMSE                 | MSE   | MAE   | R <sup>2</sup> |
| 0.058                | 0.003 | 0.037 | 0.465          |

Berikut adalah plot data dari CO DKI3 dengan kernel Laplacian.



GAMBAR 5  
CO DKI3

4. O<sub>3</sub> dari DKI1 dengan kernel Laplacian

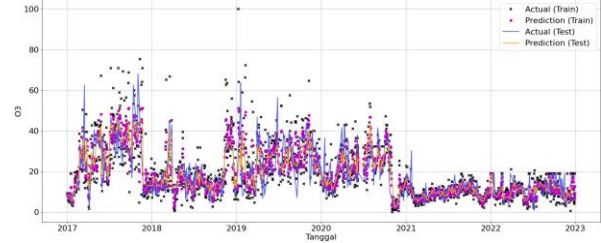
Tabel 7  
Data Train O3 DKI1

| Data Train Evaluation |       |       |                |
|-----------------------|-------|-------|----------------|
| RMSE                  | MSE   | MAE   | R <sup>2</sup> |
| 0.047                 | 0.002 | 0.031 | 0.861          |

Tabel 8  
Data Test O3 DKI1

| Data Test Evaluation |       |       |                |
|----------------------|-------|-------|----------------|
| RMSE                 | MSE   | MAE   | R <sup>2</sup> |
| 0.079                | 0.006 | 0.051 | 0.602          |

Berikut adalah plot data dari O<sub>3</sub> DKI1 dengan kernel Laplacian.



Gambar 6  
O3 DKI1

5. NO<sub>2</sub> dari DKI2 dengan kernel Laplacian

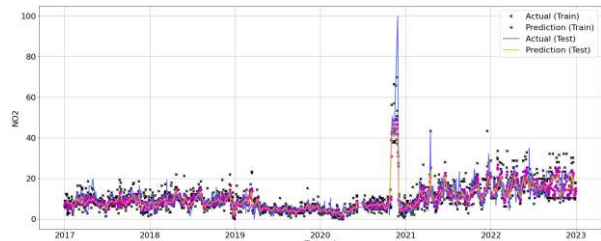
TABEL 9  
Data Train NO2 DKI2

| Data Train Evaluation |       |       |                |
|-----------------------|-------|-------|----------------|
| RMSE                  | MSE   | MAE   | R <sup>2</sup> |
| 0.027                 | 0.001 | 0.018 | 0.869          |

TABEL 10  
Data Test NO2 DKI2

| Data Test Evaluation |       |       |                |
|----------------------|-------|-------|----------------|
| RMSE                 | MSE   | MAE   | R <sup>2</sup> |
| 0.041                | 0.002 | 0.027 | 0.700          |

Berikut adalah plot data dari NO<sub>2</sub> DKI2 dengan kernel Laplacian.



GAMBAR 7  
NO2 DKI2

Pada tabel – tabel tersebut terlihat bahwa zat yang memiliki matriks evaluasi paling kecil untuk proses prediksi adalah SO<sub>2</sub> dari daerah DKI3 (Jagakarsa) dengan RMSE sebesar 0.041, MSE sebesar 0.002, MAE sebesar 0.019, dan R-squared sebesar 0.898.

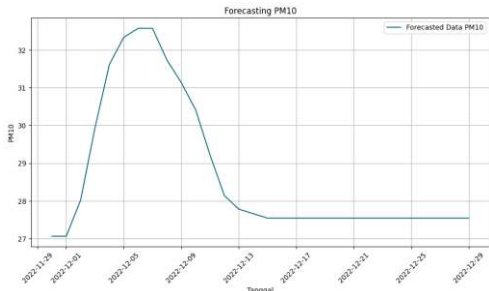
B. Forecasting

1. PM<sub>10</sub> dari DKI4 dengan kernel RBF

TABEL 11  
Forecast PM10 DKI4

| RMSE  | MSE   | MAE   |
|-------|-------|-------|
| 0.063 | 0.004 | 0.042 |

Berikut adalah plot *forecasting* dari PM<sub>10</sub> DKI4 dengan kernel RBF.



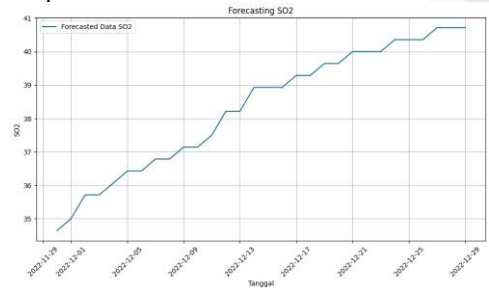
GAMBAR 8  
PM10 DKI4

2. SO<sub>2</sub> dari DKI5 dengan kernel Laplacian

TABEL 12  
Forecast SO2 DKI5

| RMSE  | MSE   | MAE   |
|-------|-------|-------|
| 0.060 | 0.004 | 0.052 |

Berikut adalah plot *forecasting* dari SO<sub>2</sub> DKI5 dengan kernel Laplacian.



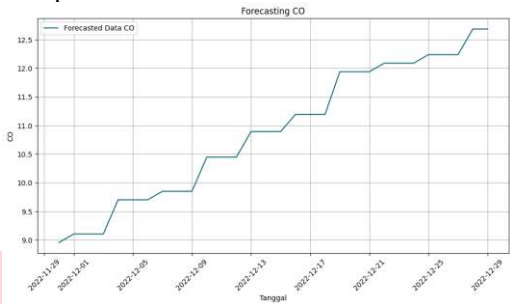
GAMBAR 9  
SO2 DKI5

3. CO dari DKI5 dengan kernel Laplacian

TABEL 13  
Forecast CO DKI5

| RMSE  | MSE   | MAE   |
|-------|-------|-------|
| 0.030 | 0.001 | 0.028 |

Berikut adalah plot *forecasting* dari CO DKI5 dengan kernel Laplacian.



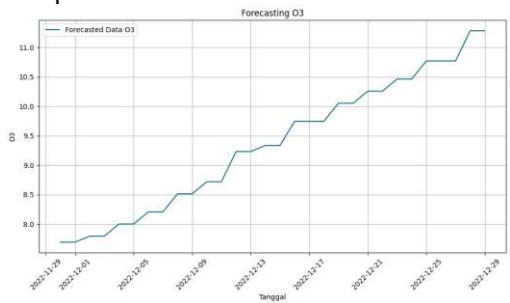
GAMBAR 10  
CO DKI5

4. O<sub>3</sub> dari DKI1 dengan kernel Laplacian

TABEL 14  
Forecast O3 DKI1

| RMSE  | MSE   | MAE   |
|-------|-------|-------|
| 0.060 | 0.004 | 0.049 |

Berikut adalah plot *forecasting* dari O<sub>3</sub> DKI1 dengan kernel Laplacian.



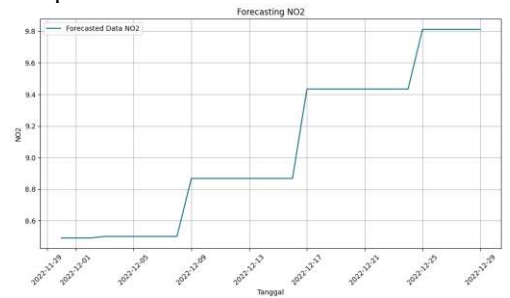
GAMBAR 11  
O3 DKI1

5. NO<sub>2</sub> dari DKI3 dengan kernel Laplacian

TABEL 15  
Forecast O3 DKI1

| RMSE  | MSE   | MAE   |
|-------|-------|-------|
| 0.034 | 0.001 | 0.023 |

Berikut adalah plot *forecasting* dari NO<sub>2</sub> DKI3 dengan kernel Laplacian.



GAMBAR 12  
NO2 DKI3

Pada tabel – tabel tersebut terlihat bahwa zat yang memiliki matriks evaluasi paling kecil untuk proses *forecasting* adalah NO<sub>2</sub> dari daerah DKI3 (Jagakarsa) dengan RMSE sebesar 0.034, MSE sebesar 0.001, dan MAE sebesar 0.023.

## V. KESIMPULAN

Berdasarkan pengujian dan analisis yang dilakukan untuk prediksi dan *forecasting* dataset ISPU DKI Jakarta menggunakan *Kernel Extreme Learning Machine* didapatkan hasil terbaik untuk proses prediksi pada partikel SO<sub>2</sub> di daerah DKI3. Hasil tersebut didapat berdasarkan nilai evaluasi matriks terkecil, yaitu RMSE sebesar 0.022, MSE sebesar 0.001, MAE 0.011, dan *R-squared* 0.969 untuk data *train*. Sedangkan untuk data *test* menghasilkan nilai evaluasi matriks terkecil untuk RMSE sebesar 0.041, MSE sebesar 0.002, MAE 0.019, dan *R-squared* 0.898. Kernel yang dipakai untuk hasil tersebut adalah *Laplacian*. Selain kernel, beberapa parameter juga mempengaruhi hasil evaluasi matriks tersebut. Parameter yang digunakan adalah *gamma* sebesar 0.01 dan nilai koefisien (C) sebesar 30. Untuk proses *forecasting*, didapat hasil terbaik pada partikel NO<sub>2</sub> di daerah DKI3 dengan nilai evaluasi matriks RMSE sebesar 0.034, MSE sebesar 0.001, dan MAE sebesar 0.023. Hasil tersebut didapat dengan menggunakan kernel *Laplacian* dengan parameter *gamma* sebesar 0.01 dan nilai koefisien (C) sebesar 300.

## REFERENSI

- [1] T. Xayasouk and H. Lee, "Air pollution prediction system using deep learning," *WIT Transactions on Ecology and the Environment*, vol. 230, pp. 71–79, 2018, doi: 10.2495/AIR180071.
- [2] IQAir, "Kualitas Udara di Jakarta," *IQAir*, Oct. 24, 2022. <https://www.iqair.com/id/indonesia/jakarta> (accessed Jul. 31, 2023).
- [3] PT. Gramedia Asri Media, "Dampak Negatif dari Pencemaran Udara & Solusinya," *Gramedia*, Oct. 24, 2022. <https://www.gramedia.com/literasi/dampak-negatif-dari-pencemaran-udara> (accessed Jul. 31, 2023).
- [4] A. Refanda Permatasari and B. Rahayudi, "Estimasi Hasil Produksi Benih Tanaman Kenaf (*Hibiscus Cannabinus* L.) Menggunakan Metode Extreme Learning Machine (ELM) Pada Balai Penelitian Tanaman Pemanis dan Serat (Balittas)," 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] X. Li *et al.*, "Empirical analysis: stock market prediction via extreme learning machine," *Neural Comput Appl*, vol. 27, no. 1, pp. 67–78, Jan. 2016, doi: 10.1007/s00521-014-1550-z.
- [6] M. Adiza Putri Nasution and I. Cholissodin, "Prediksi Price Earning Ratio Saham Menggunakan Algoritme Kernel Extreme Learning Machine (Studi Kasus: PT TELKOM)," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] G. Bin Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, Apr. 2012, doi: 10.1109/TSMCB.2011.2168604.
- [8] IEEE Engineering in Medicine and Biology Society, Annual International Conference of the IEEE Engineering in Medicine and Biology Society 35 2013.07.03-07 Osaka, EMBC 35 2013.07.03-07 Osaka, and EMBS Annual Conference 35 2013.07.03-07 Osaka, 2013 35th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 3-7 July 2013, Osaka, Japan.
- [9] B. Li, X. Rong, and Y. Li, "An improved kernel based extreme learning machine for robot execution failures," *The Scientific World Journal*, vol. 2014, 2014, doi: 10.1155/2014/906546.