TESIS

KOMPARASI METODE OPTIMASI PADA ALGORITMA GAUSSIAN PROCESS REGRESSION DALAM PREDIKSI KUALITAS UDARA

disusun oleh

Nama : **David Budi Hartanto**
NIM : **23.55.2503**
Konsentrasi : **Digital Transformation Intelligence**

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

TESIS

KOMPARASI METODE OPTIMASI PADA ALGORITMA GAUSSIAN PROCESS REGRESSION DALAM PREDIKSI KUALITAS UDARA

COMPARISON OF OPTIMIZATION METHODS IN THE GAUSSIAN PROCESS REGRESSION ALGORITHM FOR AIR QUALITY PREDICTION

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana
Program Studi S2 PJJ INFORMATIKA

disusun oleh

Nama : **David Budi Hartanto**
NIM : **23.55.2503**
Konsentrasi : **Digital Transformation Intelligence**

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2026

HALAMAN PERSETUJUAN

KOMPARASI METODE OPTIMASI PADA ALGORITMA GAUSSIAN PROCESS REGRESSION DALAM PREDIKSI KUALITAS UDARA

COMPARISON OF OPTIMIZATION METHODS IN THE GAUSSIAN PROCESS REGRESSION ALGORITHM FOR AIR QUALITY PREDICTION

yang disusun dan diajukan oleh

David Budi Hartanto

23.55.2503

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 7 Januari 2026

Dosen Pembimbing

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PENGESAHAN

KOMPARASI METODE OPTIMASI PADA ALGORITMA GAUSSIAN PROCESS REGRESSION DALAM PREDIKSI KUALITAS UDARA

COMPARISON OF OPTIMIZATION METHODS IN THE GAUSSIAN PROCESS REGRESSION ALGORITHM FOR AIR QUALITY PREDICTION

yang disusun dan diajukan oleh

David Budi Hartanto

23.55.2503

Telah dipertahankan di depan Dewan Penguji
pada tanggal 7 Januari 2026

Susunan Dewan Penguji

Nama Penguji	Tanda Tangan
Dr. Andi Sunyoto, M.Kom. **NIK. 190302052**	
I Made Artha A., S.T., M.Eng., PhD **NIK. 190302352**	
Prof. Dr. Kusrini, M.Kom. **NIK. 190302106**	

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 7 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : David Budi Hartanto
NIM : 23.55.2503

Menyatakan bahwa Tesis dengan judul berikut:

Komparasi Metode Optimasi Pada Algoritma Gaussian Process Regression Dalam Prediksi Kualitas Udara

Dosen Pembimbing : Prof. Dr. Kusrini, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 3 Februari 2026

Yang Menyatakan,

David Budi Hartanto

HALAMAN PERSEMBAHAN

Tesis ini penulis persembahkan kepada Keluarga terutama istri dan anak tercinta atas doa, dukungan, kesabaran, serta pengertian yang diberikan selama proses penyusunan tesis ini..

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT. atas limpahan rahmat dan hidayah-Nya sehingga tesis yang berjudul "Komparasi Metode Optimasi pada Algoritma Gaussian Process Regression dalam Prediksi Kualitas Udara" dapat diselesaikan dengan baik. Tesis ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Magister Komputer pada Program Studi Magister Teknik Informatika Universitas Amikom Yogyakarta. Penyusunan tesis ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan rasa syukur dan terima kasih kepada:

1. Keluarga, khususnya terhadap istri, anak dan orang tua yang telah memberikan do'a, dukungan serta pengertiannya kepada penulis selama menuntut ilmu sampai bisa menyelesaikan tanggung jawab.
2. Ibu Prof. Kusrini, M.Kom., selaku pembimbing utama yang telah sabar dalam membimbing penulis sampai bisa menyelesaikan tesis ini dengan sangat baik.
3. Seluruh jajaran, staf, dosen, dan rekan-rekan seperjuangan, serta khususnya admin PJJ yang telah meluangkan waktu dan dengan sabar membantu serta menanggapi berbagai pertanyaan dari penulis selama proses perkuliahan dan penyusunan tesis.

Yogyakarta, 3 Februari 2026

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMBANG DAN SINGKATAN

- GPR : Gaussian Process Regression
- GA : Genetic Algorithm
- RMSE : Root Mean Squared Error
- MAPE : Mean Absolute Percentage Error
- MAE : Mean Absolute Error

INTISARI

Kualitas udara merupakan salah satu indikator penting dalam menjaga kesehatan masyarakat dan lingkungan, sehingga diperlukan metode prediksi yang akurat untuk mendukung pengambilan keputusan. Penelitian ini bertujuan untuk mengevaluasi performa algoritma Gaussian Process Regression (GPR) dalam memprediksi kualitas udara dengan menerapkan optimasi hiperparameter menggunakan Genetic Algorithm (GA) dan Grid Search. Dataset yang digunakan terdiri dari beberapa parameter polutan udara, yang kemudian diproses melalui tahapan praproses data dan pembagian data menjadi data training, validation, dan test. Proses optimasi hiperparameter dilakukan dengan menyesuaikan parameter kernel GPR, yaitu constant kernel, length-scale pada Radial Basis Function (RBF), dan noise level pada WhiteKernel. Evaluasi performa model dilakukan menggunakan metrik koefisien korelasi (R), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), dan Mean Absolute Percentage Error (MAPE). Hasil penelitian menunjukkan bahwa baik Genetic Algorithm mampu menghasilkan performa prediksi yang konsisten dan stabil, dengan nilai R sekitar 0,918 serta RMSE, MAE, dan MAPE yang lebih baik di bandigkan algoritma yang lainnya. Hasil proses optimasi mampu menghasilkan konfigurasi kernel yang lebih optimal dan stabil secara sistematis. Dari sisi waktu komputasi, Grid Search menunjukkan waktu eksekusi yang lebih lama dibandingkan Genetic Algorithm, sementara Genetic Algorithm menawarkan fleksibilitas pencarian yang lebih adaptif terhadap ruang solusi yang luas. Dengan demikian, penelitian ini menunjukkan bahwa optimasi hiperparameter memberikan kontribusi penting dalam meningkatkan kualitas dan keandalan model GPR untuk prediksi kualitas udara.

Kata kunci: Gaussian Process Regression, Optimasi Hiperparameter.

ABSTRACT

Air quality is one of the important indicators in maintaining public health and environmental sustainability; therefore, accurate prediction methods are required to support decision-making processes. This study aims to evaluate the performance of the Gaussian Process Regression (GPR) algorithm in predicting air quality by applying hyperparameter optimization using Genetic Algorithm (GA) and Grid Search. The dataset used consists of several air pollutant parameters, which are processed through data preprocessing stages and divided into training, validation, and test datasets. The hyperparameter optimization process focuses on tuning the GPR kernel parameters, including the constant kernel, the length-scale of the Radial Basis Function (RBF), and the noise level of the WhiteKernel. Model performance is evaluated using the correlation coefficient (R), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The results show that the Genetic Algorithm is capable of producing consistent and stable prediction performance, achieving an R value of approximately 0.918 and lower RMSE, MAE, and MAPE compared to other optimization methods. The optimization process successfully identifies more optimal and systematically stable kernel configurations. In terms of computational time, Grid Search requires longer execution time than the Genetic Algorithm, while the Genetic Algorithm offers more flexible and adaptive search capabilities in exploring a wider solution space. Therefore, this study demonstrates that hyperparameter optimization plays a significant role in improving the quality and reliability of GPR models for air quality prediction.

Keywords: *Gaussian Process Regression, Hyperparameter Optimization.*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Konsentrasi polusi yang tinggi di udara dapat menurunkan kualitas udara dan berdampak buruk pada kesehatan masyarakat [1]. Polusi udara merupakan masalah serius yang mempengaruhi kualitas hidup penduduk dan lingkungan secara negatif. Polusi udara ini tidak hanya mengancam kesehatan tetapi juga mempengaruhi ekosistem dan kualitas udara di wilayah sekitarnya. Laporan Organisasi Kesehatan Dunia (WHO) menyatakan bahwa sekitar 90% warga dunia tinggal di wilayah di mana polusi udara melebihi ambang batas yang ditetapkan [2]. Polusi udara di dunia merupakan masalah serius yang mengubah struktur atmosfir bumi sehingga membuka celah masuknya bahaya radiasi sinar matahari (ultra violet) dan pada waktu yang bersamaan, keadaan udara yang tercemar merupakan fungsi insulator yang mencegah aliran panas kembali ke ruang angkasa, dengan demikian mengakibatkan peningkatan suhu bumi [3]. Langkah-langkah nyata seperti mengurangi emisi kendaraan, mengembangkan transportasi umum yang lebih baik, meningkatkan pengawasan industri, dan mendorong penggunaan energi bersih merupakan upaya dalam mengurangi masalah polusi udara. Namun masih banyak faktor lain yang dapat mempengaruhi kualitas udara.

Machine learning dapat berkontribusi secara signifikan dalam upaya ini dengan menyediakan alat prediksi yang akurat untuk kualitas udara, memungkinkan tindakan pencegahan yang lebih efektif dan pengambilan keputusan yang berbasis data. Dengan meningkatkan akurasi dalam prediksi

kualitas udara, kita dapat mempercepat tercapainya tujuan udara bersih, oleh karena itu, sangat penting untuk sepenuhnya memanfaatkan peran model pembelajaran mesin dalam tugas-tugas prediksi kualitas udara [4].

Formula Air Quality Index (AQI) telah ditetapkan secara resmi dan mampu mengonversi konsentrasi polutan menjadi nilai indeks, rumus tersebut hanya berfungsi untuk menghitung kondisi kualitas udara saat ini dan tidak memiliki kemampuan prediksi. US EPA [5] menjelaskan bahwa formula AQI dirancang sebagai indikator real-time, bukan untuk forecasting. Kualitas udara dipengaruhi oleh hubungan non-linear antara berbagai polutan serta faktor meteorologis yang kompleks, sehingga pendekatan deterministik berbasis rumus tidak cukup merepresentasikan pola dinamis tersebut. Pada penelitian [6] menunjukkan bahwa interaksi antarpolutan dan cuaca tidak dapat ditangkap secara akurat oleh formula AQI. Machine Learning menjadi relevan karena mampu memodelkan hubungan non-linear dan memanfaatkan seluruh fitur pencemar untuk menghasilkan prediksi yang lebih akurat. Pada penelitian Survey [7] ini menunjukkan bahwa teknik ML (regresi dan deep learning) sangat umum digunakan untuk forecasting kualitas udara, dan ML lebih efektif dibanding model tradisional. Selain itu, pada penelitian [8] menggabungkan Random Forest (ML) dengan ARIMA agar bisa menangkap pola non-linear dan temporal sekaligus, sehingga menghasilkan performa prediksi yang lebih baik dan lebih efektif untuk sistem peringatan dini..

Dalam beberapa tahun terakhir, metode pembelajaran mesin dan kecerdasan buatan telah digunakan secara luas untuk memprediksi kualitas udara. Seperti Metode Airformer [9] untuk meprediksi udara secara nasional di tiongkok, pada

penelitian lain model ARIMA-WOA-LSTM [10] menggunakan metode ARIMA untuk mengekstrak bagian data linier dan menghasilkan data non-liniernya, sementara model WOA-LSTM digunakan untuk memprediksi bagian non-linier tersebut, dengan penelitian berlokasi di Baoding, China.

Model lain seperti SAC-QBSO-BiLSTM [11] digunakan untuk meprediksi kualitas udara dengan autokorelasi Spasial Selama Pandemi COVID-19 di shanghai, pada penelitian prediksi kualiatas udara yang lain menggunakan model Temporal Difference-Based Graph Transformer Networks (TDGTN) [12] untuk memprediksi kualitas udara PM2.5 di Tiongkok. Metode lain EMD–IPSO–LSTM [13] menggunakan optimasi swarm partikel yang diperbaiki (IPSO) untuk mengidentifikasi parameter LSTM yang optimal, sehingga mendapatkan nilai prediksi yang lebih akurat di bandingkan metode LSTM. Pada penelitian Nature inspired [14] dalam penelitian tersebut mengusulkan pembelajaran deep learning, mengintegrasikan neural network (NN), inferensi sistem fuzzy, untuk memprediksi polutan udara yang paling dominan yang mempengaruhi Delhi, India.

Untuk metode Graph Neural Network (GNN) [15] teknik yang diperkenalkan dalam penelitian ini adalah kombinasi dari Gated Recurrent Unit (GRU) dan Graph Convolutional Network (GCN) dimana hasil model yang di usulkan lebih unggul dari Temporal Graph Convolutional Network (TGCN) dan LSTM-GRU. Pada penelitian lain [16] membandingkan algoritma Artificial neural network (ANN) dengan algoritma Gaussian process regression (GPR) pada kualitas udara di india, dimana hasil yang didapatkan algortima GPR dapat mengunguli algoritma ANN dalam beberapa indicator seperti RMSE, MAPE, dan MAE.

Beberapa kelebihan dari model machine learning dan deep learning yang ditemukan adalah memiliki nilai akurasi yang tinggi, dan kemampuannya dalam menangani data yang besar dapat mengurangi biaya komputasi dan meningkatkan fleksibilitas jaringan saraf dinamis, namun ada juga kekurangan dari model-model tersebut seperti keterbatasan data, yang mempengaruhi tingkat akurasi suatu model. Beberapa model machine learning dan deep learning memiliki kompleksitas komputasi yang tinggi, yang dapat menjadi hambatan dalam aplikasi real-time yang membutuhkan prediksi cepat.

Gaussian Process Regression (GPR) merupakan salah satu metode machine learning yang menjanjikan untuk prediksi kualitas udara karena sifatnya yang fleksibel dan non-parametrik. Meskipun Gaussian Process Regression (GPR) adalah metode yang fleksibel dan non-parametrik, ada beberapa kekurangan yang dapat diidentifikasi. Salah satunya adalah bahwa GPR dapat menjadi komputasi yang mahal, terutama ketika berhadapan dengan dataset yang besar, karena kompleksitas waktu dan ruangnya meningkat secara kuadratik dengan jumlah data [17]. Selain itu, GPR juga dapat mengalami kesulitan dalam menangani data yang sangat bising atau tidak teratur, yang dapat mempengaruhi akurasi prediksi. Performa GPR sangat bergantung pada pemilihan hyperparameter yang tepat, khususnya parameter kernel function, length scale, dan noise variance.

Untuk mengatasi tantangan optimasi hyperparameter pada Gaussian Process Regression (GPR), berbagai metode optimasi telah dikembangkan dan digunakan dalam penelitian terdahulu. Genetic Algorithm (GA) merupakan metode optimasi berbasis evolusi yang meniru mekanisme seleksi alam melalui proses

seleksi, crossover, dan mutasi untuk mencari solusi terbaik dalam ruang parameter yang kompleks . GA mampu melakukan eksplorasi ruang solusi secara luas dan efektif sehingga cocok untuk menangani permasalahan optimasi non-linear dan multidimensional. Namun, metode ini memerlukan waktu komputasi yang lebih besar karena bergantung pada evaluasi populasi secara iteratif.

Sementara itu, Grid Search merupakan metode optimasi berbasis pencarian menyeluruh (exhaustive search) yang mengevaluasi seluruh kombinasi hyperparameter dalam ruang parameter yang telah ditentukan [3], [4]. Pendekatan ini sederhana dan mudah diimplementasikan, tetapi memiliki kelemahan berupa kebutuhan komputasi yang tinggi terutama ketika jumlah parameter atau rentang pencarian semakin besar, sehingga kurang efisien untuk ruang hyperparameter berskala besar [5]. Masing-masing metode optimasi ini memiliki karakteristik, kelebihan, dan keterbatasan yang berbeda, sehingga pemilihannya perlu disesuaikan dengan kompleksitas model dan kebutuhan eksperimen..

Beberapa penelitian telah mengeksplorasi penggunaan metode optimasi untuk meningkatkan performa algoritma machine learning. Penelitian oleh (Nama, dkk, tahun) menggunakan GA untuk optimasi Support Vector Machine dan menunjukkan peningkatan akurasi sebesar X%. Sementara itu, penelitian (Nama, dkk, tahun) membandingkan PSO dan GA dalam optimasi Neural Network dan menemukan bahwa PSO lebih cepat konvergen namun GA menghasilkan solusi yang lebih robust. Namun, penelitian komparasi yang komprehensif mengenai berbagai metode optimasi pada algoritma GPR untuk prediksi kualitas udara masih terbatas.

Mengingat pentingnya optimasi hyperparameter dalam meningkatkan performa GPR dan beragamnya metode optimasi yang tersedia, penelitian ini bertujuan untuk melakukan komparasi sistematis terhadap beberapa model optimasi pada algoritma Gaussian Process Regression dalam memprediksi kualitas udara. Model optimasi yang akan dibandingkan meliputi Genetic Algorithm (GA), dan Grid Search. Komparasi akan dilakukan berdasarkan beberapa aspek meliputi akurasi prediksi yang diukur dengan metrik RMSE, MAE, dan MAPE dan efisiensi komputasi yang diukur dari waktu eksekusi dan jumlah iterasi.

Melalui komparasi yang komprehensif, penelitian ini diharapkan dapat memberikan rekomendasi evidence-based mengenai metode optimasi yang paling efektif untuk GPR dalam konteks prediksi kualitas udara. Dengan prediksi yang akurat tentang kualitas udara, masyarakat dapat diberitahu lebih awal tentang potensi polusi tinggi. Hasil komparasi ini juga diharapkan dapat menjadi referensi bagi peneliti dan praktisi dalam memilih metode optimasi yang sesuai dengan karakteristik data dan kebutuhan aplikasi mereka, serta membuka peluang pengembangan hybrid optimization methods yang mengkombinasikan kelebihan dari berbagai metode untuk mencapai performa optimal.

1.2 Rumusan Masalah

Berdasarkan penjabaran pada latar belakang masalah, maka rumusan masalah pada penelitian ini adalah sebagai berikut :

a. Bagaimana performa masing-masing model optimasi (Genetic Algorithm, dan Grid Search) dalam mengoptimalkan hyperparameter algoritma Gaussian Process Regression untuk prediksi kualitas udara?

b. Mengukur berapa nilai akurasi prediksi pada algoritma Gaussian Process Regression dengan optimasi berdasarkan metrik evaluasi R, RMSE, MAE, dan MAPE?

1.3 Batasan Masalah

Agar penelitian ini tetap fokus, relevan, dan tidak terlalu luas, maka perlu ditetapkan batasan-batasan masalah sebagai berikut:

a. Penelitian ini menggunakan algoritma Gaussian Process Regression
b. Dataset yang digunakan berasal dari jurnal rujukan sebelumnya (Suri, dkk, 2023) https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india.
c. Model optimasi yang dibandingkan terbatas pada dua metode yaitu Genetic Algorithm dan Grid Search.
d. Indikator yang digunakan adalah R, RMSE, MAPE, dan MAE.
e. Menggunakan korelasi untuk pemilihan fitur
f. Menggunakan Perbandingan rasio data 70 : 30
g. Penelitian ini menggunakan Bahasa pemrograman python
h. Penelitian ini berfokus untuk meningkatkan performa Algoritma GPR

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, berikut beberapa tujuan dari penelitian :

a. Mengetahui kombinasi parameter optimal yang digunakan pada model GPR yang dioptimasi dengan Genetic Algorithm dan Grid Search.
b. Membandingkan hasil akurasi prediksi kualitas udara dengan algoritma GPR yang sudah dioptimasi.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini dijabarkan sebagai berikut :

a. Penelitian ini dapat memberikan pemahaman yang lebih mendalam tentang bagaimana algoritma GPR bekerja dan bagaimana algoritma ini dapat dioptimalkan.
b. Di harapkan Algoritma yang di kembangkan dapat di terapkan pada dataset yang ada di negara lain, baik dalam hal bidang keilmuan yang sama atau hal bidang keilmuan yang lain.

BAB 2

TINJAUAN PUSTAKA

2.1 Tijauan Pustaka

Penelitian mengenai kualitas udara telah banyak dilakukan di berbagai negara (Zhang, dkk, 2024). Berbagai model deep learning telah diterapkan untuk menemukan hasil yang optimal di wilayah-wilayah tersebut. Salah satu penelitian menggunakan model Graph Neural Network (GNN), yang menunjukkan potensi besar dalam meningkatkan akurasi prediksi kualitas udara. Selain itu, model GNN juga telah diterapkan pada penelitian di bidang lain, menunjukkan kemampuannya untuk memberikan hasil yang optimal dalam berbagai konteks analisis data.

Penelitian kualitas udara yang di lakukan di madrid (Iskandaryan, dkk, 2023), mengusulkan metode A3T-GCN, yang menggabungkan Attention, Gated Recurrent Unit (GRU), dan Graph Convolutional Network (GCN). Metode ini dirancang untuk menangkap informasi spatiotemporal dari data kualitas udara, meteorologi, dan lalu lintas. Model diuji menggunakan data dari stasiun pemantauan kualitas udara di Madrid untuk periode Januari-Juni 2019 (set pelatihan) dan Januari-Juni 2022 (set pengujian). Model dievaluasi menggunakan metrik seperti Root Mean Square Error (RMSE), Mean Absolute Error (MAE), dan Koefisien Korelasi Pearson. A3T-GCN menunjukkan kinerja yang lebih baik dibandingkan dengan model referensi lainnya, seperti Temporal Graph Convolutional Network (TGCN), Long Short-Term Memory (LSTM), dan GRU. Secara spesifik, A3T-GCN mengungguli TGCN dengan 10.89%, LSTM dengan 14.29%, dan GRU dengan 14.13% dalam hal RMSE. Penelitian ini menunjukkan

bahwa A3T-GCN adalah metode yang efektif untuk prediksi kualitas udara, dengan kemampuan untuk mengintegrasikan berbagai jenis data dan menangkap hubungan spatiotemporal yang kompleks. Peneliti mencatat bahwa meskipun metode yang diusulkan menunjukkan hasil yang baik, tantangan tetap ada dalam integrasi data yang besar dan kompleks. Penelitian lebih lanjut diperlukan untuk mengatasi masalah ini dan untuk mengeksplorasi aplikasi metode ini di daerah lain.

Pada penelitian selanjutnya, membahas pengembangan model baru yang disebut Temporal Difference-Based Graph Transformer Networks (TDGTN) (Zhang, dkk, 2022) untuk prediksi kualitas udara, khususnya konsentrasi PM2.5. Model ini menggabungkan keunggulan Graph Neural Networks (GNNs) dan Transformer untuk memproses data yang terstruktur dalam bentuk graf dan menangkap ketergantungan temporal yang panjang. Penelitian ini berfokus pada prediksi PM2.5 menggunakan data deret waktu dari stasiun pemantauan udara. Penulis mengusulkan TDGTN sebagai pendekatan baru yang memanfaatkan informasi perbedaan temporal antara momen waktu yang berdekatan. Model TDGTN dibangun untuk mengolah data deret waktu PM2.5 dan membangun data terstruktur graf tanpa memerlukan struktur graf eksplisit. Hasil eksperimen menunjukkan bahwa TDGTN mengungguli model-model lain seperti ARMA, SVR, CNN, LSTM, dan Transformer asli dalam tugas prediksi kualitas udara baik untuk jangka pendek (1 jam) maupun jangka panjang (6, 12, 24, 48 jam). Model ini menunjukkan kemampuan yang lebih baik dalam menangkap pola temporal dan hubungan kompleks dalam data PM2.5. Peneliti menyarankan agar penelitian lebih lanjut dilakukan untuk mengeksplorasi penerapan model ini pada dataset yang lebih

besar dan beragam, serta mempertimbangkan faktor-faktor eksternal yang mungkin mempengaruhi kualitas udara.

Pada peelitian ini membahas pengembangan model baru untuk peramalan kualitas udara, khususnya konsentrasi PM2.5, menggunakan jaringan Transformer (sparse attention-based Transformer networks, STN) (Zhang & Zhang, 2023). Penelitian ini memperkenalkan model STN yang dirancang untuk mempelajari ketergantungan jangka panjang dan hubungan kompleks dari data deret waktu PM2.5. Model ini menggunakan mekanisme perhatian jarang multi-kepala dalam encoder dan decoder untuk menangkap informasi dinamis temporal jangka panjang sambil mengurangi kompleksitas waktu. Model ini dapat memproses seluruh data deret waktu PM2.5 secara bersamaan berkat mekanisme perhatian diri yang digunakan, berbeda dengan metode yang sebelumnya yang memerlukan data deret waktu untuk diproses dalam urutan tertentu. Penelitian ini juga mencatat bahwa metode pembelajaran mendalam sebelumnya, seperti RNN dan LSTM, memiliki keterbatasan dalam memodelkan hubungan jangka panjang dan kompleks karena masalah "gradient vanishing and exploding". Model Transformer menawarkan solusi untuk masalah ini. Pada dataset PM2.5 Beijing, model ini mencapai nilai R^2 sebesar 0.9113, sedangkan pada dataset PM2.5 Taizhou, model ini mencapai R^2 sebesar 0.924, RMSE sebesar 5.79, dan MAE sebesar 3.76. Peramalan Jangka Panjang untuk PM2.5, model STN juga menunjukkan hasil yang lebih baik dibandingkan dengan metode yang lain. Peneliti mencatat masih ada kebutuhan untuk penelitian lebih lanjut untuk mengeksplorasi penerapan model ini dalam konteks yang lebih luas dan untuk dataset yang lebih beragam.

Penelitian berikutnya membahas masalah kualitas udara yang serius di China (Huang, dkk, 2022), yang dipengaruhi oleh perubahan iklim, polusi industri, dan urbanisasi. Penelitian ini mengusulkan model kombinasi baru yang menggabungkan Empirical Mode Decomposition (EMD), Particle Swarm Optimization (PSO), dan Long Short-Term Memory (LSTM) untuk memprediksi Indeks Kualitas Udara (AQI). Model EMD–IPSO–LSTM menunjukkan peningkatan akurasi prediksi AQI dibandingkan dengan model lain seperti BP, LR, dan EMD–LSTM. Decomposisi EMD memungkinkan pemisahan data menjadi komponen dengan frekuensi yang berbeda, yang membantu LSTM dalam memprediksi dengan lebih baik. Algoritma PSO digunakan untuk menentukan jumlah neuron optimal dalam lapisan tersembunyi LSTM, yang berkontribusi pada peningkatan akurasi model. Hasil eksperimen menunjukkan bahwa model yang diusulkan memiliki kinerja prediksi yang lebih baik dan stabil di berbagai lokasi pengukuran (Dongsi, Guanyuan, dan Tiantan). Penelitian ini menyarankan agar faktor meteorologi dan emisi kendaraan dipertimbangkan dalam model di masa depan untuk meningkatkan akurasi prediksi lebih lanjut.

Penelitian selanjutnya dilakukan di India (Suri, dkk, 2023) prediksi kualitas udara menggunakan pendekatan pembelajaran mesin, khususnya Artificial Neural Network (ANN) dan Gaussian Process Regression (GPR). Dengan meningkatnya masalah polusi udara di kota-kota besar, penelitian ini bertujuan untuk mengembangkan model yang akurat untuk memprediksi Indeks Kualitas Udara (AQI) di enam kota di India. Data yang digunakan dalam penelitian ini diambil dari situs Kaggle, mencakup berbagai parameter polusi. Hasil analisis menunjukkan

bahwa model GPR secara signifikan lebih efektif dibandingkan dengan model ANN, dengan koefisien korelasi (R) sebesar 0.9843 untuk GPR, dibandingkan dengan 0.9611 untuk ANN. Temuan ini menegaskan potensi GPR sebagai alat yang lebih andal dalam memprediksi kualitas udara. Kesimpulan dari penelitian ini merekomendasikan penggunaan GPR untuk pengembangan strategi pengendalian polusi yang lebih baik, meskipun juga mencatat beberapa kelemahan terkait kompleksitas komputasi dan pemilihan fungsi kovarians.

Model optialisasi GA dan GPR juga di terapkan pada bidang penelitian lain, seperti penelitian asupan nutrisi dan performa ayam broiler (Ahmadi, dkk, 2023), analisis mendalam tentang penerapan kombinasi Gaussian Process Regression (GPR) dan Genetic Algorithm (GA). Tujuan utama dari penelitian ini adalah untuk memahami trade-off antara dua respons penting, yaitu average daily gain (ADG) dan gain feed ratio, yang dapat meningkatkan akurasi dalam formulasi pakan. Peneliti menghipotesiskan bahwa dengan mengkuantifikasi trade-off ini, formulasi pakan dapat dilakukan dengan lebih tepat, terutama ketika hubungan antara variabel tidak bersifat linier. Dalam studi ini, data yang digunakan berasal dari eksperimen sebelumnya yang mengevaluasi pengaruh asupan tiga nutrisi : digestible glycine equivalents, digestible threonine, dan total choline terhadap performa ayam broiler. Metode GPR diterapkan untuk memodelkan hubungan nonlinier antara variabel input dan respons, sementara GA digunakan untuk mengoptimalkan kombinasi nutrisi yang memberikan hasil terbaik. Hasil penelitian menunjukkan bahwa pendekatan GPR-GA dapat memberikan prediksi yang lebih baik dan membantu

dalam pengambilan keputusan yang lebih informasional dalam formulasi pakan ayam broiler.

Pada penelitian (Hai, dkk, 2024) beberapa model optimasi digunakan pada algoritma Gaussian Process Regression (GPR) untuk memprediksi viskositas suspense. Penelitian ini menggunakan tiga algoritma metaheuristic yaitu : Genetic Algorithm (GA), Particle Swarm Optimization (PSO), dan Marine Predators Algorithm (MPA). Melalui analisis hyperparameter yang komprehensif, penulis mengelompokkan hyperparameter ke dalam tiga kategori berdasarkan efektivitasnya dan mengoptimalkan masing-masing kelompok secara terpisah. Hasil penelitian menunjukkan bahwa model GPR yang dioptimalkan mencapai nilai R yang sangat tinggi (0.999224). Secara keseluruhan, penelitian ini memberikan kontribusi signifikan terhadap pengembangan metodologi optimasi hyperparameter dalam konteks aplikasi material dan energi.

2.2 Keaslian Penelitian

Tabel 2. 1 Matriks literatur review dan posisi penelitian

Komparasi Metode Optimasi Pada Algoritma Gaussian Process Regression Dalam Prediksi Kualitas Udara

No	JudulL Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	Air Quality Prediction - A Study Using Neural Network Based Approach	Raunaq Singh Suri, Ajay Kumar Jain, Nishant Raj Kapoor, Aman Kumar, Harish Chandra Arora, Krishna Kumar, Hashem Jahangir (2023) Q2	Artificial Neural Network (ANN) dan Gaussian Process Regression (GPR)	Evaluasi model GPR R^2 : 0.9843, RMSE : 21.41 MAPE : 7.89% MAE : 13.59 Evaluasi model ANN : R : 0.9611 RMSE: 33.28 MAPE: 13.24% MAE: 22.94	**Keunggulan:** Membandingkan 2 algortima, dengan hasil penelitian menunjukkan bahwa model GPR memiliki kinerja yang lebih baik dibandingkan dengan model ANN dalam hal Evaluasi R^2, RMSE, MAPE, dan MAE **Kelemahan:** Belum menerapkan hyperparameter Tuning	Penelitian ini menambahkan Hyperparameter Tuning Genetic Algoritm dan Grid Search

No	Judul. Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
2	A deep learning approach for prediction of air quality index in smart city	D. Sivabalaselvamani, P. Kanimozhi, M. Gopikrishnan, S. Aravind, A. M. Senthilkumar, dan K. Sathesh Kumar (2024) Q2	Menangani nilai hilang dengan Deep GAN (Generative Adversarial Network) dan memperkuat akurasi prediksi dengan model GRU (Gated Recurrent Unit)	Hasil evaluasi model : MAE = 0.1013 MSE = 0.0134 RMSE = 0.1156 R = 0.9479 MAPE = 0.1152	**Keunggulan:** mengatasi masalah data hilang dan ketidaklengkapan data menggunakan Deep GAN **Kelemahan:** memperluas cakupan data, menggunakan model berbasis AI yang lebih kompleks seperti deep learning hybrid	Tidak ada kesamaan pada algoritma yang digunakan, dan adanya penerapan yang sama dalam proses imputasi data.
3	A real-time environmental air pollution predictor model using dense deep learning approach in IoT infrastructure	P. Preethi, T. Saravanan, R. Mohanraj, dan P. G. Gayathri (2024) Q3	Mengembangkan model prediksi polusi udara real-time berbasis Dense Residual Convolutional Network dengan Bi-LSTM yang diintegrasikan ke dalam infrastruktur	Polutan dengan akurasi tertinggi CO (R^2= 0.97, RMSE = 0.15)	**Keunggulan:** Melakukan proses imputasi dan seleksi fitur spasial-temporal menggunakan CNN **Kelemahan:**	Pada penelitian ini nilai prediksi terdapat pada AQI bukan pada polutan udara dan tidak ada kesamaan model yang digunakan.

No	Judul. Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
			Internet of Things (IoT)		Penelitian pada setiap polutan dibeberapa wilayah	
4	Impact on Air Quality Index of India Due to Lockdown	Aditya Dubey dan Akhtar Rasool. (2023) Q1	Support Vector Regression (SVR) Kernel yang digunakan: Linear kernel Extreme Gradient Boosting (XGBoost Regressor) Menggunakan model gradient boosted decision trees.	Hasil Evaluasi SVR R^2 : 0.804 MAPE : 0.183 Hasil Evaluasi XGBoost : R^2 : 0.703 MAPE : 0.177	**Keunggulan:** Teknik Imputasi nilai diisi dengan mean atau median dari fitur tersebut. Alat yang digunakan, yaitu SimpleImputer **Kelemahan:** Tidak Adanya Optimasi Model untuk Edge Computing	Pada penelitian ini tidak terdapat model optimasi untuk hypertuning model
5	Optimized machine learning model for air quality index prediction in major cities in India	Suresh Kumar Natarajan, Prakash Shanmurthy, Daniel Arockiam, Balamurugan Balusamy, dan Shitharth Selvarajan	Algoritma Grey Wolf Optimization (GWO) untuk mengoptimalkan fitur-fitur relevan dalam dataset, serta decision tree regression untuk melakukan	Nilai akurasi tertinggi pada model decision tree regression dengan nilai evaluasi R : 0.894 MSE: 0.1288 RMSE: 0.271 MAE: 0.0575	**Keunggulan:** model machine learning yang dioptimalkan menggunakan algoritma Grey Wolf Optimization dan decision tree regression mampu memberikan prediksi indeks kualitas udara (AQI) yang lebih akurat	Pada penelitian ini juga mencoba menerapkan GWO utuk melakukan fitur selection, serta menggunakan korelasi sebagai pembanding.

No	Judul. Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
		(2024) Q1	prediksi indeks kualitas udara (AQI). Serta algoritma lain Support Vector Regression (SVR), Random Forest Regression (RFR), dan K-Nearest Neighbor (KNN)		dibandingkan dengan metode konvensional **Kelemahan:** Tidak adanya hypertuning pada algoritma yang digunakan	
6	Prediction of Air Quality Index Using Machine Learning Techniques: A Comparative Analysis	N. Srinivasa Gupta, Yashvi Mohta, Khyati Heda, Raahil Armaan, B. Valarmathi, dan G. Arulkumaran (2023) Q2	Menggunakan model Linear Regression (LR), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN)	algoritma Random Forest memberikan hasil prediksi AQI paling akurat dibandingkan model lain dengan nilai evaluasi. MAE : 0.027 MSE : 0.011 RMSE : 0.105 R2 : 0.987	**Keunggulan:** Membandingkan banyak algoritma. **Kelemahan:** Tidak adanya hypertuning pada model	Pada penelitian ini melakukan hypertuning pada algoritma yang digunakan. Dan tidak ada persamaan pada algoritma yang digunakan

2.3 Landasan Teori

2.3.1 Pencemaran Udara

Pencemaran udara terjadi ketika unsur-unsur berbahaya masuk atau bercampur dengan atmosfer, yang dapat menyebabkan kerusakan lingkungan, mengganggu kesehatan manusia, dan secara keseluruhan menurunkan kualitas lingkungan [3]. Ketika Standar Kualitas Udara Ambien Nasional (National Ambient Air Quality Standards) ditetapkan pada tahun 1970, pencemaran udara terutama dianggap sebagai ancaman bagi kesehatan pernapasan. Seiring dengan kemajuan penelitian pencemaran udara selama beberapa dekade berikutnya, kekhawatiran kesehatan masyarakat meluas mencakup penyakit kardiovaskular, diabetes, obesitas, gangguan reproduksi, neurologis, sistem imun, dan kanker [18]. Pada tahun 2013, Badan Internasional untuk Penelitian Kanker (International Agency for Research on Cancer) dari Organisasi Kesehatan Dunia (WHO) mengklasifikasikan pencemaran udara sebagai karsinogen bagi manusia.

2.3.2 Zat-zat Polutan Udara

Terdapat banyak zat-zat polutan pencemar udara yang dapat diidentifikasi, namun beberapa di antaranya yang utama adalah sebagaimana disajikan dalam Tabel 2.2 berikut :

Tabel 2. 2 Zat-zat polutan udara

Polutan udara	Sumber	Dampak terhadap Kesehatan
$PM_{2.5}$ (partikel dengan diameter kurang dari 2.5 mikrometer) & PM_{10} (partikel dengan	Debu di tepi jalan, polutan, kegiatan konstruksi besar, pembakaran untuk pengurangan bahaya, garam	Risiko meningkat terhadap penyakit kardiopulmoner dan kanker paru-paru, asma

Polutan udara	Sumber	Dampak terhadap Kesehatan
diameter kurang dari 10 mikrometer)	laut, pembangkit listrik, mesin motor, pemanas kayu, kebakaran semak, dan proses pembakaran.	pada anak-anak, aritmia jantung, serangan jantung, serangan asma, dan bronkitis.
NO_2 (Dioksida Nitrogen)	Bau Menyengat yang Tajam. Kebakaran, bahan bakar fosil, mesin pembakaran dalam, dan uji coba nuklir.	Iritasi paru-paru & peningkatan kemungkinan infeksi saluran pernapasan, penyakit pengisi silo.
NH_3 (Amonia)	Materi hewan dan tanaman yang mengandung nitrogen, air hujan, dan aktivitas vulkanik.	Ketidaknyamanan di tenggorokan, hidung, mata, dan saluran pernapasan, kerusakan paru-paru, kebutaan, dan kematian.
CO (karbon Monoksida)	Pembakaran thermal, degradasi fotokimia dari bahan tanaman, reaksi kimia dengan senyawa organik yang diemisikan oleh aktivitas manusia, gunung berapi, kebakaran hutan dan semak, pembakaran bahan bakar yang tidak sempurna, dan asap tembakau.	Kelelahan, mual, sakit kepala, gangguan penglihatan, nyeri dada, kebingungan, penurunan fungsi otak, pusing, dan dapat berakibat fatal pada konsentrasi yang sangat tinggi.
SO_2 (Sulfur Dioksida)	Pembakaran bahan bakar yang mengandung sulfur, peleburan bijih mineral yang mengandung sulfur, gunung berapi.	Iritasi kulit, batuk, iritasi tenggorokan, kesulitan bernapas, sekresi lendir, asma, dan bronkitis kronis.
O_3 (Ozon)	Perangkat pembersih udara bertegangan tinggi, dan pelepasan listrik serta aksi UV pada dioxygen.	Alergi, sakit tenggorokan, asma, mata gatal dan berair, pembengkakan dan penyumbatan di sistem pernapasan.

Dari jenis-jenis zat tersebut dapat diketahui sebab-sebab pencemaran udara berasal dari industri, Emisi kendaraan bermotor, dan kebakaran hutan. Dampak pencemaran udara lebih mempengaruhi anak-anak dari pada orang dewa. Terutama

kepada anak-anak miskin. karena kondisi lingkungannya, mereka terekspos pada lebih banyak jenis polutan dan tingkat pencemaran yang lebih tinggi. Beberapa studi membuktikan bahwa anak-anak yang tinggal di kota dengan tingkat pencemaran udara lebih tinggi mempunyai paru-paru lebih keci, lebih sering tidak bersekolah karena sakit, dan lebih sering dirawat di rumah sakit.

2.3.3 Algoritma Gaussian Process Regression (GPR)

Algoritma Gaussian Process Regression (GPR) termasuk dalam metode supervised learning [19]. Salah satu keunggulan dari Gaussian Process Regression dalam memperkirakan kualitas udara adalah kemampuannya untuk menghasilkan rata-rata prediksi serta varians prediktif [16]. Regresi Proses Gaussian (GPR) adalah model probabilistik berbasis kernel yang non-parametrik [20]. Berdasarkan definisi dalam beberapa referensi, model GPR menjelaskan respons dengan menggunakan fungsi dasar eksplisit dan variabel laten dari proses Gaussian (GP). GP sendiri adalah proses stokastik yang didefinisikan oleh variabel acak dengan distribusi Gaussian bersama. Fungsi kovarians dari variabel laten ini menangkap kelancaran respons, dan fungsi dasar memproyeksikan input ke ruang fitur berdimensi p. Oleh karena itu, model GPR bersifat probabilistik dan menggunakan pendekatan Bayesian. Tidak seperti algoritma pembelajaran mesin yang diawasi lainnya, pendekatan Bayesian tidak memerlukan nilai pasti untuk setiap parameter fungsi tujuan, melainkan memperkirakan distribusi probabilitas atas semua nilai yang mungkin.

Asumsi umumnya adalah bahwa target (y) didefinisikan sebagai f(x)+ ξ. Dimana ξ ~ N(0,σ^2), yang berarti bahwa kesalahan observasi didistribusikan secara independen dengan rata-rata nol dan varians σ^2, sementara f(x) diambil dari proses Gaussian. Model Gaussian Process Regression memiliki persamaan (2.1):

$$f(x) \sim GP(m(x), k(x, x^1)) \quad (2.1)$$

Dimana m(x) adalah fungsi rata-rata, yang dalam banyak penelitian sebelumnya dianggap m(x)=0 dan k(x,x^1) adalah fungsi kovarians yang sangat dipengaruhi oleh pemilihan kernel dan berpengaruh besar terhadap kualitas prediksi. Distribusi bersama antara output pelatihan f=(f_1,f_2,…,f_n) dan output pengujian f_*=(f_(n+1),f_(n+2),…,f_(n+n)).

Optimasi pada Gaussian Process Regression (GPR) dilakukan untuk menemukan kombinasi hyperparameter kernel yang paling sesuai dengan pola data melalui proses memaksimalkan marginal log-likelihood (MLL), sehingga model dapat menghasilkan prediksi yang paling mungkin terhadap data observasi yang tersedia [21].

Secara lebih mendalam, efektivitas proses optimasi sangat bergantung pada bagaimana masing-masing hyperparameter mempengaruhi fleksibilitas dan kompleksitas fungsi yang dibangun oleh GPR. Pemilihan Length-scale (ℓ) adalah hyperparameter yang mengatur seberapa cepat fungsi GPR boleh berubah. Jika nilai ℓ kecil, model menganggap hubungan antar titik data berubah sangat cepat, sehingga prediksi menjadi lebih berfluktuasi atau terlihat "berisik". Sebaliknya, jika ℓ besar, model menganggap perubahan data berlangsung perlahan, sehingga menghasilkan fungsi yang sangat halus dan cenderung rata. Secara praktik, titik

awal pencarian ℓ biasanya disesuaikan dengan skala fitur input. Jika fitur memiliki rentang sangat besar atau tidak distandarisasi, ℓ akan sulit ditafsirkan. Karena itu, normalisasi/standarisasi hampir selalu dilakukan agar ℓ bekerja dengan skala yang konsisten.

Parameter Signal variance (σ^2_f) mengontrol seberapa besar variasi vertikal fungsi yang dimodelkan GPR. Jika σ^2_f besar, fungsi prior GPR memungkinkan nilai output berubah dengan rentang yang lebar. Jika σ^2_f kecil, fungsi menjadi lebih mendatar di sekitar nilai rata-rata. Parameter ini hampir selalu dioptimasi bersamaan dengan ℓ karena keduanya bekerja bersama dalam menentukan bentuk fungsi yang dipelajari GPR.

Sementara itu, noise variance (σ^2_n) menggambarkan seberapa besar "kebisingan" pada data observasi. Jika σ^2_n terlalu kecil, model akan mencoba mengikuti semua titik data termasuk noise, sehingga cenderung overfitting. Jika σ^2_n terlalu besar, model menganggap sebagian besar variasi data hanyalah noise, sehingga fungsi menjadi terlalu halus dan kehilangan pola penting (underfitting).

Ketiga hyperparameter ini saling terkait dan menciptakan trade-off. Misalnya, model bisa menjelaskan variasi sebagai sinyal (σ^2_f besar, ℓ kecil) atau sebagai noise (σ^2_n besar, ℓ besar). Akibatnya, permukaan marginal log-likelihood (MLL) yang harus dioptimasi sering memiliki banyak "puncak lokal" (multimodal). Inilah mengapa pemilihan titik awal dan metode optimasi seperti GA menjadi penting agar model tidak terperangkap pada solusi yang kurang optimal..

Pemilihan hyperparameter seperti length-scale, signal variance, dan noise variance sangat menentukan bentuk fungsi yang dapat dipelajari GPR, dan

pemilihan nilai yang tidak tepat dapat menyebabkan model mengalami *overfitting* atau *underfitting* [21]. Oleh karena itu, proses optimasi diperlukan untuk menyesuaikan hyperparameter agar kernel mampu merepresentasikan struktur non-linear yang terdapat pada data secara optimal [21]. Dalam implementasi praktis, optimasi hyperparameter membantu mengurangi kesalahan prediksi dan meningkatkan generalisasi model, karena tujuan utamanya adalah mencari kombinasi parameter yang memberikan performa prediktif terbaik pada data baru [22]. Metode optimasi seperti algoritma evolusioner seperti Genetic Algorithm (GA) digunakan untuk menjelajahi ruang hyperparameter secara efisien tanpa harus mencoba seluruh kombinasi seperti pada Grid Search yang berbiaya komputasi sangat tinggi [23].

2.3.4 Genetic Algoritm (GA)

Algoritma genetika (GA) adalah algoritma optimasi yang terinspirasi oleh seleksi alam [24]. Ini adalah algoritma pencarian berbasis populasi yang memanfaatkan konsep kelangsungan hidup yang paling cocok. Populasi baru dihasilkan secara iteratif melalui penggunaan operator genetik pada individu yang ada dalam populasi. Elemen-elemen kunci dari GA termasuk representasi kromosom, seleksi, crossover, mutasi, dan perhitungan fungsi fitness. Prosedur GA adalah sebagai berikut: sebuah populasi yang terdiri dari sejumlah kromosom diinisialisasi secara acak. Fitness dari setiap kromosom dihitung, lalu dua kromosom dipilih berdasarkan nilai fitness-nya. Operator crossover titik tunggal dengan probabilitas crossover diterapkan untuk menghasilkan keturunan baru.

Selanjutnya, operator mutasi seragam diterapkan pada keturunan dengan probabilitas mutasi untuk menghasilkan keturunan yang diubah. Proses seleksi, crossover, dan mutasi diulangi sampai populasi baru terbentuk.

GA secara dinamis mengubah proses pencarian melalui probabilitas crossover dan mutasi untuk mencapai solusi optimal. GA dapat memodifikasi gen yang terkode. GA juga dapat mengevaluasi banyak individu dan menghasilkan beberapa solusi optimal. Oleh karena itu, GA memiliki kemampuan pencarian global yang lebih baik. Keturunan yang dihasilkan dari crossover kromosom induk berpotensi menghilangkan pola genetik yang baik dari kromosom induk, dan rumus crossover didefinisikan sebagaimana persamaan (2.2):

$$R = (G + 2\sqrt{g})/3G \tag{2.2}$$

di mana g adalah jumlah generasi, dan G adalah total jumlah generasi evolusi yang ditetapkan oleh populasi. Dari Persamaan (2.2) dapat dilihat bahwa R berubah secara dinamis dan meningkat seiring dengan meningkatnya jumlah generasi evolusi. Pada tahap awal GA, kesamaan antara individu sangat rendah. Nilai R harus rendah untuk memastikan bahwa populasi baru tidak akan menghancurkan skema genetik yang baik dari individu. Pada akhir evolusi, kesamaan antara individu sangat tinggi, sehingga nilai R juga harus tinggi.

Beberapa penelitian terkini telah mengeksplorasi penggunaan Algoritma Genetika (GA) untuk optimasi hyperparameter pada model Gaussian Process Regression (GPR), menunjukkan bahwa GA dapat meningkatkan performa GPR terutama ketika ruang hyperparameter sangat kompleks. Sebagai dalam studi [25] GA, bersama dengan Particle Swarm Optimization (PSO) dan Marine Predators

Algorithm (MPA), digunakan untuk menyesuaikan 12 hyperparameter GPR. Hasilnya, model GPR yang dioptimasi menggunakan GA mampu mencapai nilai R lebih tinggi dibandingkan tanpa optimasi., dalam konteks desain otomotif cerdas, penelitian [26] "Adaptive GPR-Based Multidisciplinary Design Optimization of Structural and Control Parameters of Intelligent Bus" menggunakan GA yang dimodifikasi (menggabungkan mekanisme percepatan gravitasi) untuk mengoptimalkan kombinasi fungsi kernel GPR. Metode ini meningkatkan kemampuan pencarian global dan konvergensi cepat untuk menemukan hyperparameter kernel GPR yang paling optimal. Dari berbagai studi tersebut menjelaskan GA dapat digunakan untuk optimasi GPR karena kapasitasnya menjelajahi ruang hyperparameter yang besar dan kompleks, serta kemampuan mencapai solusi optimal secara global, yang mungkin sulit dicapai dengan metode optimasi lain.

2.3.5 Grid Search

Grid Search adalah metode optimasi hyperparameter yang melakukan pencarian menyeluruh (exhaustive search) terhadap seluruh kombinasi nilai hyperparameter yang telah ditentukan sebelumnya dalam ruang pencarian (search space) [27]. Metode ini bekerja dengan mendefinisikan himpunan nilai kandidat untuk setiap hyperparameter, kemudian mengevaluasi model pada setiap kombinasi nilai tersebut untuk menemukan konfigurasi hyperparameter dengan performa terbaik berdasarkan metrik evaluasi tertentu, seperti MAE, RMSE, atau akurasi [28]. Karena mengevaluasi seluruh kombinasi, Grid Search merupakan pendekatan

deterministik yang menjamin ditemukannya kombinasi terbaik dalam ruang pencarian yang ditetapkan, meskipun tidak mampu menjamin optimalitas global jika ruang pencarian tidak mencakup seluruh kemungkinan nilai [29].

Jika terdapat k hyperparameter, dan setiap hyperparameter ke-i memiliki n_i kandidat nilai, maka jumlah total kombinasi Grid Search dapat dihitung dengan persamaan (2.3) berikut

$$N_{\text{grid}} = \prod_{i=1}^{k} \quad n_i \tag{2.3}$$

Rumus ini menggambarkan kompleksitas komputasi Grid Search, yang meningkat secara eksponensial seiring dengan bertambahnya jumlah hyperparameter ataupun jumlah kandidat nilai yang diuji [27].

Proses optimasi menggunakan Grid Search dilakukan melalui beberapa tahapan yang sistematis. Pertama, peneliti menentukan hyperparameter yang ingin dioptimasi, seperti *length-scale*, *signal variance*, atau *noise variance* pada Gaussian Process Regression (GPR). Selanjutnya, ditetapkan daftar nilai kandidat untuk setiap hyperparameter, yang umumnya berupa rentang nilai diskrit yang ditentukan secara manual atau berdasarkan pengetahuan awal terhadap karakteristik data [28]. Setelah itu, ruang pencarian (*grid*) dibangun dengan menyusun seluruh kombinasi dari nilai-nilai kandidat tersebut sesuai dengan rumus $N_{\text{grid}} = \prod n_i$, yaitu jumlah total kombinasi yang harus dievaluasi. Setiap kombinasi kemudian digunakan untuk melatih model, dan performanya dihitung menggunakan metrik evaluasi yang relevan, seperti MAE atau RMSE. Dalam praktiknya, Grid Search sering digabungkan dengan *k-fold cross-validation* untuk memperoleh estimasi performa

yang lebih stabil dan memastikan bahwa hyperparameter yang dipilih memiliki kemampuan generalisasi yang baik terhadap data baru [30]. Setelah seluruh kombinasi diuji, hyperparameter dengan performa terbaik dipilih sebagai konfigurasi final model.

Sejumlah penelitian telah memanfaatkan Grid Search untuk mengoptimasi hyperparameter Gaussian Process Regression (GPR). Pada penelitian [31] membahas penggunaan Grid Search dalam konteks pemilihan hyperparameter GPR, sekaligus menyoroti kelemahannya terutama ketika data bersifat spars sehingga metode ini berpotensi menyebabkan overfitting jika ruang pencarian terlalu luas. Selain penelitian akademik, pada publikasi Wiley (2019) [32], mencatat bahwa Grid Search merupakan metode optimasi dasar yang sering digunakan sebagai baseline sebelum menerapkan teknik yang lebih canggih seperti Bayesian Optimization atau Genetic Algorithm dalam proses tuning hyperparameter GPR.

2.3.6 Metrik Evaluasi

Performa regresi mengacu pada seberapa baik model regresi mampu memprediksi nilai variabel dependen atau target berdasarkan variabel independen atau fitur yang ada. Performa regresi diukur dengan berbagai metrik evaluasi yang mencerminkan seberapa baik model regresi dapat menyesuaikan data dan memprediksi nilai yang akurat. Beberapa metrik evaluasi kinerja regresi yang umum digunakan meliputi:

a. R (Koefisien Korelasi Pearson)

Koefisien korelasi Pearson (R) merupakan ukuran statistik yang digunakan untuk menilai kekuatan dan arah hubungan linier antara dua variabel kuantitatif. Nilai R berada dalam rentang −1 hingga +1, di mana nilai mendekati +1 menunjukkan hubungan linier positif yang kuat, nilai mendekati −1 menunjukkan hubungan linier negatif yang kuat, sedangkan nilai mendekati 0 menunjukkan tidak adanya hubungan linier yang signifikan antara dua variabel [33].

Secara matematis, koefisien korelasi Pearson didefinisikan sebagai rasio antara kovarians dua variabel terhadap hasil kali standar deviasi masing-masing variabel. Rumusnya dituliskan sebagai persamaan (2.4) berikut:

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (2.4)$$

Dimana :

x_i: nilai variabel x pada data ke-i

y_i: nilai variabel y pada data ke-i

$\bar{x}$: rata-rata seluruh nilai variabel x

$\bar{y}$: rata-rata seluruh nilai variabel y

n : jumlah total data/observasi dalam dataset

b. RMSE (Root Mean Squared Error)

RMSE adalah akar kuadrat dari variansi residu dan menunjukkan kecocokan absolut model terhadap data (perbedaan antara data yang diamati dengan nilai prediksi model) [34]. Ini dapat diinterpretasikan sebagai deviasi standar dari variansi yang tidak terjelaskan, dan memiliki satuan yang sama dengan variabel

respons. Nilai yang lebih rendah menunjukkan kecocokan model yang lebih baik. RMSE didefinisikan sebagaimana persamaan (2.5) berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \quad (2.5)$$

Keterangan :

x_i : data aktual pada periode t

y_i : nilai peramalan

n : jumlah data

c. MAPE (Mean Percentage Absolute Error)

MAPE adalah salah satu alat ukur yang paling banyak digunakan untuk akurasi peramalan, karena memiliki keuntungan dalam hal independensi skala dan interpretabilitas [35]. MAPE dihitung dengan menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian, merata-ratakan kesalahan persentase absolut tersebut. MAPE merupakan pengukuran kesalahan yang menghitung ukuran presentase penyimpangan antara data actual dengan data peramalan. Nilai MAPE dapat dihitung dengan persamaan (2.6):

$$MAPE = \frac{1}{n}\sum_{t-1}^{n}\left|\frac{X_t - F_t}{X_t}\right| \quad (2.6)$$

Dimana :

Xt : data aktual pada periode t

Ft : nilai peramalan

n : jumlah data

d. MAE (Mean Absolute Error)

MAE didefinisikan sebagai rata-rata dari nilai absolut dari selisih antara nilai observasi dan prediksi model, MAE dianggap lebih robust dibandingkan RMSE dalam situasi di mana kesalahan tidak terdistribusi normal (Hodson, 2022). Misalnya nilai MAE 0 menunjukkan tidak ada perbedaan antara nilai prediksi dan nilai observasi. Nilai MAE dapat dihitung dengan persamaan (2.7):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (2.7)$$

Dimana :

n : adalah jumlah total sampel yang dievaluasi

y_i : adalah keluaran (aktual) yang diamati ke-i,

$\hat{y}_i$: adalah keluaran (prediksi) ke-i

Berdasarkan uraian tersebut, pemilihan metrik evaluasi yang tepat sangat penting untuk menilai kinerja model regresi secara komprehensif. Setiap metrik memiliki kelebihan dan keterbatasannya masing-masing, sehingga kombinasi beberapa metrik sering digunakan untuk mendapatkan gambaran yang lebih akurat mengenai kemampuan model dalam memprediksi. Dengan demikian, analisis R, RMSE, MAPE, dan MAE secara bersamaan dapat memberikan penilaian yang lebih menyeluruh terhadap performa model regresi yang dikembangkan.

2.3.7 Feature Selection

Feature selection adalah proses memilih subset fitur yang paling relevan untuk membangun model prediksi yang lebih akurat, efisien, dan mudah diinterpretasikan. Tujuan utama seleksi fitur adalah mengurangi noise, mencegah overfitting, serta menurunkan kompleksitas komputasi. Secara umum, metode feature selection terbagi menjadi tiga kategori: filter (berbasis statistik fitur), wrapper (menggunakan model untuk menilai subset), dan embedded (seleksi dilakukan sebagai bagian dari pelatihan model) [36].

Salah satu metode *feature selection* yang paling sederhana namun efektif adalah seleksi fitur berbasis korelasi (*correlation-based feature selection*) yang menilai kekuatan hubungan linear antara fitur dengan variabel target [37]. Metode ini termasuk kategori *filter*, karena proses seleksi dilakukan menggunakan statistik korelasi tanpa melibatkan algoritma pembelajaran mesin sehingga bersifat cepat dan independen terhadap model. Dalam pemodelan kualitas udara, analisis korelasi digunakan untuk mengidentifikasi variabel yang memiliki pengaruh signifikan terhadap AQI atau polutan tertentu, sehingga fitur yang tidak relevan atau memiliki korelasi sangat rendah dapat dieliminasi sejak tahap awal pemrosesan data [38].

Selain itu, korelasi juga digunakan untuk mendeteksi multikolinearitas, yaitu kondisi ketika dua atau lebih fitur memiliki hubungan sangat kuat, sehingga fitur yang bersifat redundan dapat dihapus untuk meningkatkan stabilitas model dan mencegah overfitting [39]. Metode ini penting karena mampu menyederhanakan ruang fitur, mengurangi kompleksitas komputasi, dan meningkatkan interpretabilitas model tanpa mengorbankan akurasi prediksi [39].

BAB 3

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Untuk jenis penelitian yang akan diterapkan merupakan penelitian Eksperimen dimana peneliti melakukan suatu tindakan pada suatu algoritma untuk melihat suatu peningkatan performa suatu algoritma, dengan melakukan simulasi dan pengujian yang dilakukan pada kombinasi tersebut.

Sifat penelitian yang akan dilakukan adalah eksploratif dimana penelitian akan mencoba mengkombinasikan GPR dan optimasi yaitu GA dan GridSearch digunakan untuk mensimulasikan prediksi kualitas udara. Tujuan utamanya adalah untuk mengetahui kombinasi parameter GPR yang dapat meningkatkan akurasi prediksi.

Untuk pendekatan penelitian termasuk ke jenis pendekatan kuantitatif karena menggunakan data numerik, serta melakukan pengukuran dengan metrik kuantitatif seperti R, RMSE, MAPE, dan MAE.

3.2 Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini bersumber dari dataset publik yang tersedia pada platform Kaggle, dengan tautan akses sebagai berikut: https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india. Dataset ini telah digunakan pada beberapa penelitian sebelumnya sebagai bahan analisis kualitas udara di wilayah perkotaan di India, sehingga pemanfaatannya dalam penelitian ini diharapkan dapat memberikan kesinambungan serta memungkinkan

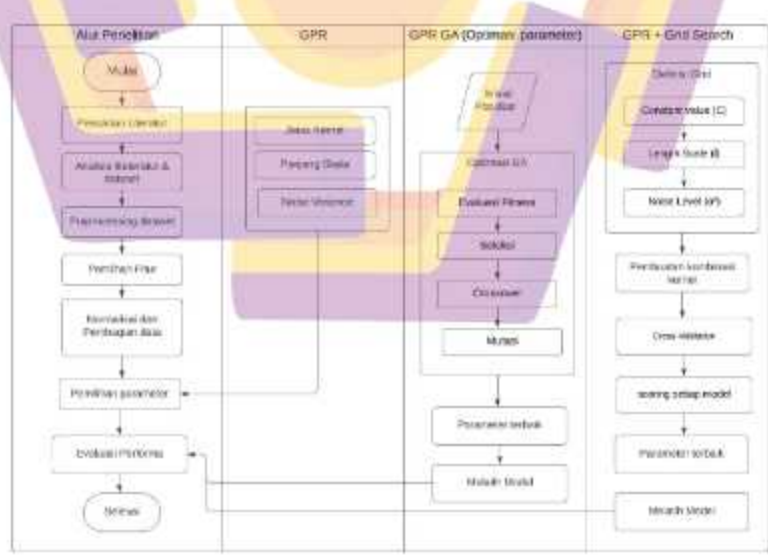
perbandingan hasil yang lebih relevan. Dataset yang dipilih adalah city_day.csv, yang memuat 16 kolom (attributes) dan 29.531 baris (records).

3.3 Metode Analsis Data

Penelitian ini memanfaatkan platform *Google Colab* sebagai alat komputasi berbasis *cloud* yang terintegrasi dengan bahasa pemrograman Python untuk proses analisis data. Visualisasi dilakukan menggunakan statistik deskriptif dataset dan *scatter plot*, sedangkan pemodelan memanfaatkan algoritma *Gaussian Process Regression* (GPR) serta GPR yang dioptimasi dengan *Genetic Algorithm* (GPR-GA) dan GridSearch sebagai pembanding.

3.4 Alur Penelitian

Alur penelitian yang digunakan dalam studi ini ditunjukkan pada Gambar 3.1 berikut.



Gambar 3. 1 Alur Penelitian

Berdasarkan Gambar 3.1 tahapan penelitian dapat di jeaskan sebagai berikut :

a. Pencarian literatur

Pada tahap ini, peneliti melakukan kajian literatur secara komprehensif terhadap penelitian-penelitian terdahulu yang relevan, antara lain karya [16], yang dijadikan sebagai acuan sekaligus pembanding bagi penelitian yang akan dilaksanakan. Peneliti juga mengidentifikasi dan memperoleh sumber dataset yang digunakan, serta mempelajari secara mendalam langkah-langkah metodologis yang diterapkan pada penelitian acuan tersebut. Selain itu, peneliti menelaah referensi tambahan, guna mengadaptasi dan mengimplementasikan pendekatan yang sesuai dengan tujuan dan konteks penelitian ini.

b. Analisa literatur dan dataset

Pada tahap ini, peneliti berupaya mengidentifikasi kebaruan (novelty) penelitian dengan mempelajari secara mendalam penelitian-penelitian terdahulu serta menelusuri literatur-literatur yang relevan. Tujuannya adalah untuk menemukan pendekatan atau metode yang dapat digunakan untuk mengoptimalkan hasil penelitian sebelumnya. Setelah memperoleh dataset yang akan digunakan, peneliti melakukan analisis awal terhadap dataset tersebut, meliputi identifikasi jenis data, jumlah kolom, kelengkapan data, serta kondisi keseluruhan dataset, sebagai langkah awal sebelum dilakukan pengolahan dan pemodelan lebih lanjut.

c. Preprocessing data

Pada tahap ini dataset yang digunakan telah disesuaikan dengan rujukan yang menjadi acuan penelitian. Selanjutnya, dilakukan proses data cleaning yang

mencakup penghapusan missing values (nilai hilang) dan penghapusan outlier dan teknik imputasi.

d. Feature Selection

Tahap berikutnya adalah melakukan feature selection dengan tujuan mengidentifikasi variabel-variabel yang memiliki pengaruh signifikan maupun pengaruh rendah terhadap nilai Air Quality Index (AQI). Serta membandingkan jumlah fitur yang paling relevan dengan tingkat korelasi.

e. Normalisasi dan Pembagian data

Pada tahap awal pemodelan, data penelitian dibagi menjadi dua komponen utama, yaitu fitur (features) dan target. Setelah normalisasi, data dibagi menjadi tiga subset dengan metode train-test split bertingkat (nested splitting). proporsi pembagian data adalah 70% pelatihan, 15% validasi, dan 15% pengujian.

f. Model Gaussian Process Regression (GPR)

Pada tahap ini, pemodelan menggunakan algoritma Gaussian Process Regression (GPR) dengan pendekatan kernel gabungan yang mengombinasikan tiga komponen utama, yaitu ConstantKernel, Radial Basis Function (RBF), dan WhiteKernel.

g. Model GPR-GA

Proses optimasi Genetic Algorithm (GA) pada Gaussian Process Regression (GPR) dimulai dengan mendefinisikan hyperparameter yang akan dioptimalkan, seperti length-scale, variance, atau noise level pada kernel GPR, serta menetapkan rentang nilai kandidat untuk masing-masing hyperparameter. Selanjutnya, GA

membuat populasi awal yang terdiri dari sejumlah individu, di mana setiap individu mewakili satu kombinasi hyperparameter yang berbeda. Setiap individu dievaluasi performanya dengan melatih model GPR menggunakan data latih dan menghitung metrik evaluasi, misalnya RMSE atau R, sebagai fitness function. Berdasarkan nilai fitness, individu terbaik dipilih melalui proses seleksi, kemudian dilakukan crossover untuk menghasilkan keturunan baru dengan kombinasi hyperparameter yang memadukan karakteristik dari orang tua, dan mutasi secara acak diterapkan pada beberapa individu untuk menjaga keragaman populasi. Proses ini diulang secara iteratif selama sejumlah generasi yang telah ditentukan, sehingga populasi secara bertahap bergerak menuju kombinasi hyperparameter yang optimal. Setelah generasi terakhir, individu dengan nilai fitness terbaik dipilih sebagai konfigurasi hyperparameter optimal, dan model GPR dibangun menggunakan konfigurasi tersebut, kemudian dilakukan evaluasi akhir performa model.

h. Model GPR optimasi Grid Search

Proses optimasi Grid Search pada Gaussian Process Regression (GPR) dimulai dengan menentukan hyperparameter utama yang akan dioptimalkan, seperti length-scale pada kernel RBF, variance, atau noise level. Setiap hyperparameter diberikan daftar nilai kandidat yang akan dicoba, sehingga membentuk sebuah grid kombinasi. Selanjutnya, untuk setiap kombinasi hyperparameter, model GPR dibangun dan dilatih menggunakan data latih, kemudian dievaluasi performanya dengan menggunakan data validasi berdasarkan metrik tertentu, seperti RMSE atau R. Proses ini dilakukan secara sistematis untuk seluruh kombinasi dalam grid, sehingga setiap kemungkinan dicoba. Setelah semua kombinasi dievaluasi,

hyperparameter yang menghasilkan performa terbaik dipilih sebagai konfigurasi optimal model, kemudian model tersebut digunakan untuk prediksi dan dievaluasi performanya menggunakan metrik yang telah ditentukan

i. Evaluasi performa

Model dievaluasi meliputi koefisien korelasi Pearson (R), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), dan Mean Absolute Error (MAE).

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Analisis Dataset

Dataset penelitian diperoleh dari dataset publik "*Air Quality Data in India*" yang tersedia di *Kaggle* dengan tahap awal analisis yaitu dilakukan pemeriksaan awal terhadap dataset guna memperoleh gambaran umum mengenai struktur dan karakteristik data. Berdasarkan hasil pemeriksaan tersebut, diketahui bahwa dataset terdiri atas 29.531 baris (*records*) dan 16 kolom (*attributes*), yang mencakup tipe data numerik dan objek (*string*). Hasil eksplorasi lebih lanjut menunjukkan bahwa dataset mengandung sejumlah nilai hilang (*missing values*) yang bervariasi pada beberapa kolom. Keberadaan nilai hilang ini berpotensi memengaruhi hasil analisis apabila tidak dilakukan penanganan yang tepat. Namun demikian, hasil pengecekan terhadap kemungkinan adanya data duplikat menunjukkan bahwa dataset bebas dari entri ganda. Rincian hasil pemeriksaan awal tersebut disajikan secara terperinci pada Tabel 4.1 berikut.

Tabel 4. 1. Struktur dan Karakteristik Data

No.	Kolom	Tipe Data	Data Kosong
1	City	Object	0
2	Date	Object	0
3	PM2.5	Float64	4598
4	PM10	Float64	11140
5	NO	Float64	3582
6	NO2	Float64	3585
7	NOx	Float64	4185
8	NH3	Float64	10328
9	CO	Float64	2059
10	SO2	Float64	3854
11	O3	Float64	4022

No.	Kolom	Tipe Data	Data Kosong
12	Benzene	Float64	5623
13	Toluene	Float64	8041
14	Xylene	Float64	18109
15	AQI	Float64	4681
16	AQI Bucket	Object	4681

Berdasarkan Tabel 4.1, dapat disimpulkan bahwa meskipun dataset memiliki cakupan data yang luas dengan 29.531 entri, terdapat variasi jumlah missing values pada beberapa atribut, dengan jumlah tertinggi ditemukan pada kolom *Xylene* sebanyak 18.109 data hilang. Kondisi ini menunjukkan perlunya penerapan metode penanganan nilai hilang agar hasil analisis tidak bias.

4.2 *Preprocessing* dataset

Tahapan preprocessing data merupakan langkah penting sebelum melakukan pemodelan kualitas udara menggunakan algoritma machine learning, karena kualitas data sangat mempengaruhi performa model. Pada dataset dilakukan pengecekan terhadap data duplikat serta identifikasi data kosong (NaN). Data kosong pada kolom AQI dihapus, sedangkan data kosong pada kolom lainnya diimputasi. Setelah proses imputasi, dilakukan pengecekan dan pembersihan outlier serta noise agar jumlah data yang hilang tidak terlalu banyak. Tahap berikutnya adalah analisis korelasi untuk mengetahui hubungan antar atribut terhadap nilai AQI.

4.2.1 Penanganan Missing Values

Langkah awal setelah melakukan pengecekan dataset adalah menangani nilai hilang (missing values) pada dataset. Proses ini penting karena keberadaan data kosong dapat menurunkan kualitas model dan menyebabkan bias pada proses

pelatihan. Pada tahap awal dilakukan pengecekan jumlah missing values pada setiap variabel untuk mengetahui sebaran dan tingkat kelengkapan data. Selanjutnya diterapkan metode imputasi menggunakan pendekatan statistik, yaitu menggantikan nilai yang hilang dengan nilai median dari masing-masing variabel. Pemilihan median bertujuan untuk menjaga kestabilan distribusi data, terutama pada variabel yang memiliki outlier. Setelah imputasi dilakukan, seluruh variabel akan terisi, dan bebas dari missing values seperti pada gambar 4.1.

```
=== TABEL MISSING VALUES SEBELUM & SESUDAH IMPUTASI ===
         Missing Sebelum  Missing Sesudah
PM2.5               4598                0
PM10               11140                0
NO                  3582                0
NO2                 3585                0
NOx                 4185                0
NH3                10328                0
CO                  2059                0
SO2                 3854                0
O3                  4022                0
Benzene             5623                0
Toluene             8041                0
Xylene             18109                0
```

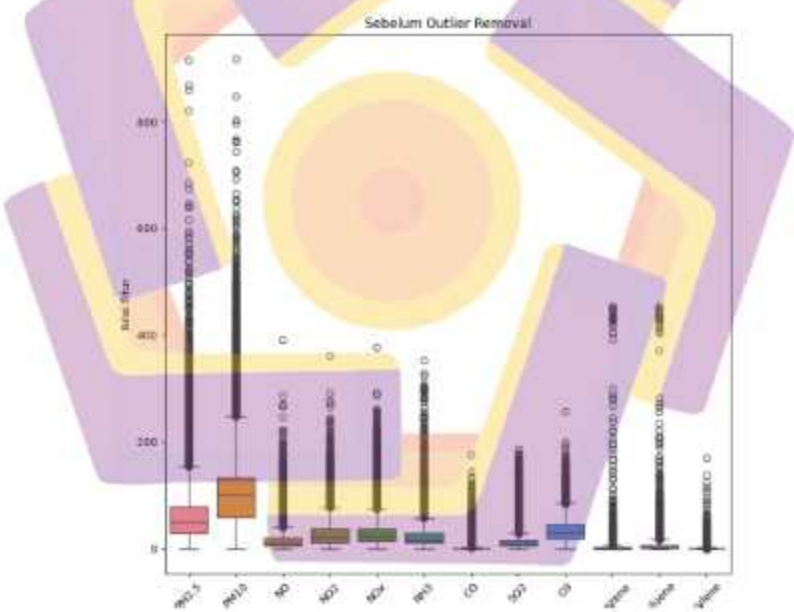
Gambar 4. 1. Perbadingan hasil Imputasi

Pada gambar 4.1 menunjukkan perbandingan jumlah data kosong sebelum dan sesudah imputasi menunjukkan bahwa proses ini berhasil memperbaiki kualitas dataset secara menyeluruh sehingga dataset siap digunakan pada tahapan analisis berikutnya, seperti deteksi outlier dan pemodelan GPR.

4.2.2 Pembersihan Data (Data Cleaning)

Pada implementasi proses pembersihan data (data cleaning) yang berfokus pada pengecekan outlier. Pada tahap pertama dilakukan deteksi dan penghapusan outlier menggunakan metode Interquartile Range (IQR), yaitu dengan menghitung

kuartil pertama (Q1), kuartil ketiga (Q3), serta rentang antar kuartil (IQR = Q3 – Q1). Data yang berada di luar batas bawah (Q1 – 1.5 × IQR) dan batas atas (Q3 + 1.5 × IQR) dianggap sebagai outlier dan dieliminasi dari dataset. Dengan demikian, proses ini bertujuan untuk memastikan bahwa dataset yang digunakan pada tahap analisis dan pemodelan memiliki kualitas yang baik, bebas dari gangguan nilai hilang dan outlier yang berpotensi menurunkan akurasi model prediksi. Pada proses penghapusan outlier menggunakan metode Interquartile Range (IQR). Kondisi awal dataset di tampilkan dalam bentuk gambar 4.2 berikut.



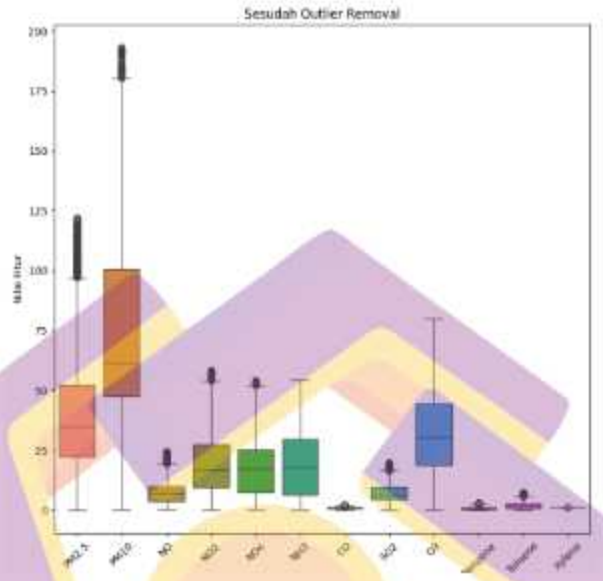
Gambar 4. 2. Outlier pada Dataset

Dari gamar 4.2 menunjukkan hasil visualisasi boxplot bahwa sebagian besar variabel polutan dalam dataset memiliki sebaran data yang relatif beragam dengan

keberadaan sejumlah outlier. Polutan PM2.5 dan PM10 tampak memiliki distribusi yang paling lebar dan banyak titik ekstrem di atas whisker, yang mengindikasikan adanya kejadian polusi dengan konsentrasi sangat tinggi. Kondisi ini wajar karena kedua polutan tersebut merupakan penyumbang utama indeks kualitas udara dan sangat dipengaruhi oleh aktivitas transportasi maupun industri.

Variabel lain seperti NO, NO_2, NOx, dan NH_3 juga memperlihatkan adanya outlier, meskipun dalam jumlah lebih sedikit, sehingga mayoritas datanya tetap berada dalam rentang yang stabil. Sementara itu, polutan seperti CO, SO_2, Benzene, Toluene, dan Xylene menunjukkan sebaran yang lebih sempit dengan sedikit outlier, menandakan konsentrasi yang cenderung konsisten. Ozon (O_3) memiliki variasi cukup lebar dengan beberapa nilai ekstrem, yang kemungkinan dipengaruhi oleh kondisi atmosfer dan intensitas radiasi matahari.

Adapun nilai AQI sebagai indikator kualitas udara gabungan menunjukkan distribusi luas dengan sejumlah outlier di level tinggi, yang sejalan dengan temuan pada polutan utama seperti PM2.5 dan PM10. Keberadaan outlier ini tidak selalu berarti kesalahan data, melainkan bisa merefleksikan kondisi polusi ekstrem yang nyata. Namun demikian, dalam proses preprocessing, data-data ekstrem perlu dianalisis lebih lanjut untuk menentukan apakah akan dipertahankan sebagai representasi kondisi khusus atau dihapus guna menjaga kestabilan model prediksi. Berikut hasil dari proses pembersihan outlier dapat dilihat pada gambar 4.3.



Gambar 4. 3 Dataset setelah pembersihan outlier

Gambar 4.3 memperlihatkan distribusi data yang lebih terpusat dan homogen setelah melalui proses preprocessing. Nilai ekstrem yang tidak wajar telah dieliminasi menggunakan metode Interquartile Range (IQR) untuk penghapusan outlier. Proses ini menghasilkan pola hubungan antara AQI dan masing-masing parameter pencemar yang lebih jelas, di mana keterkaitan positif yang kuat antara PM2.5 dan PM10 dengan AQI tampak lebih konsisten dan terdefinisi. Selain itu, rentang nilai pada sumbu X dan Y menjadi lebih terbatas, menunjukkan bahwa dataset yang dihasilkan memiliki kualitas yang lebih baik, bersih, serta relevan untuk analisis lanjutan. Dataset awal terdiri dari 29.531 data. Proses imputasi nilai hilang pada fitur polutan tidak mengubah jumlah data. Setelah penghapusan data dengan nilai AQI kosong, jumlah data berkurang menjadi 24.850 data. Selanjutnya,

penghapusan outlier menggunakan metode IQR mengurangi data menjadi 6.898 data, dan proses penghilangan noise menggunakan Z-score menghasilkan 6.157 data sebagai dataset akhir yang digunakan dalam pemodelan.

4.3 Fitur Selection

Untuk mengukur tingkat hubungan antara setiap atribut dengan variabel target yang dalam penelitian ini dilakukan analisis korelasi. Tahapan ini merupakan prosedur umum yang banyak digunakan dalam penelitian sebelumnya karena mampu menunjukkan seberapa kuat keterkaitan suatu atribut terhadap variabel target. Melalui korelasi, peneliti dapat mengidentifikasi atribut mana yang memiliki nilai yang mendekati 1 atau −1, yang menandakan hubungan yang kuat. Sebaliknya, nilai mendekati 0 menunjukkan bahwa atribut tersebut memiliki keterkaitan yang lemah dengan target. Gambar 4.4 memperlihatkan hasil korelasi setiap atribut pada dataset AQI.

Gambar 4. 4. Korelasi atribut dengan target

Dari gambar 4.4 korelasi yang dihasilkan, terlihat bahwa fitur PM2.5 memiliki korelasi tertinggi terhadap AQI dengan nilai sebesar 0.81, yang menunjukkan bahwa peningkatan konsentrasi PM2.5 sangat memengaruhi kenaikan nilai AQI. Korelasi cukup kuat juga terlihat pada PM10 dengan nilai 0.56, menandakan bahwa kedua parameter partikel padat inilah yang memiliki peran paling dominan dalam pembentukan indeks kualitas udara. Sementara itu, polutan gas seperti NO2, NOx, NO, dan O3 memiliki korelasi sedang (0.25–0.35), menunjukkan bahwa meskipun berkontribusi, dampaknya tidak sebesar parameter partikulat.

Adapun polutan lain seperti CO, SO2, Benzene, Toluene, dan Xylene menunjukkan korelasi yang relatif rendah terhadap AQI. Selain itu, pola korelasi juga memperlihatkan hubungan kuat antar-fitur tertentu, misalnya antara NO–NO_2–NOx serta Benzene–Toluene, yang mengindikasikan bahwa variabel-variabel tersebut sering berasal dari sumber emisi yang sama. Hasil analisis korelasi ini sangat penting karena dapat menjadi dasar dalam pemilihan fitur serta meningkatkan interpretasi model prediksi AQI. Dari hasil korelasi maka dapat di urutkan fitur-fitur yang berpengaruh terhadap nilai target AQI seperti pada gambar 4.5.

```
=== Top Features with Highest Correlation to AQI ===
 1. PM2.5      → correlation = 0.806
 2. PM10       → correlation = 0.559
 3. NO2        → correlation = 0.348
 4. O3         → correlation = 0.320
 5. NH3        → correlation = 0.281
 6. NO         → correlation = 0.255
 7. NOx        → correlation = 0.249
 8. CO         → correlation = 0.221
 9. SO2        → correlation = 0.159
10. Benzene    → correlation = 0.103
11. Toluene    → correlation = 0.089
12. Xylene     → correlation = 0.032
```

Gambar 4. 5. Urutan fitur yang berpengaruh terhadap AQI

Berdasarkan Gambar 4.5 AQI diperkirakan akan sangat dipengaruhi oleh fitur-fitur dengan hubungan linear yang kuat terhadap target. Dengan demikian, PM2.5 dan PM10 menjadi dua fitur yang paling penting bagi model karena kontribusinya yang signifikan terhadap dinamika perubahan nilai AQI. Fitur-fitur

gas seperti NO_2, NOx, O_3, dan NO kemungkinan tetap memberikan informasi tambahan bagi GPR karena meskipun korelasinya sedang, nilai-nilai tersebut dapat membantu menjelaskan variasi non-linear pada pencemaran udara. Sementara itu, fitur dengan korelasi sangat rendah seperti Benzene, Toluene, dan Xylene memiliki kemungkinan lebih kecil untuk meningkatkan performa prediksi, bahkan dapat berpotensi menyebabkan noise apabila dimasukkan tanpa proses seleksi fitur. Dengan karakteristiknya yang sensitif terhadap pola non-linear dan dependensi antar-data, GPR akan memberikan hasil terbaik apabila fitur-fitur dominan (PM2.5, PM10, NO2, NOx) dijadikan prioritas, sedangkan fitur minor dipertimbangkan berdasarkan hasil uji performa model. Analisis fitur ini membantu memastikan bahwa model tidak kelebihan fitur (overfitting) dan mampu memberikan prediksi AQI yang lebih akurat dan stabil.

4.4 Algoritma GPR

Gaussian Process Regression (GPR) adalah metode regresi non-parametrik yang memodelkan data sebagai sebuah proses Gaussian, di mana setiap titik data saling berhubungan melalui fungsi kovarians atau *kernel*. GPR sangat fleksibel dalam menangkap pola non-linier karena model ini tidak mengasumsikan bentuk fungsi tertentu, melainkan membiarkan data membentuk fungsinya sendiri. Kinerja GPR sangat bergantung pada hyperparameter kernel seperti *length-scale*, *signal variance*, dan *noise level (alpha)* yang menentukan seberapa halus, sensitif, dan stabil prediksi yang dihasilkan.

4.5 Persiapan skenario

Tahap selanjutnya adalah penyusunan skenario penelitian. Pada tahap ini penelitian dirancang untuk beberapa skenario eksperimen yang memudahkan proses pengujian dan analisis performa model. Skenario ini disusun berdasarkan penggunaan dan perbandingan teknik pemilihan fitur, serta penerapan metode optimasi hyperparameter pada algoritma Gaussian Process Regression (GPR).

Pada penelitian ini, pemilihan fitur dilakukan menggunakan pendekatan korelasi untuk memilih variabel yang paling relevan terhadap target AQI. Tidak hanya pemilihan fitur, penelitian ini juga membandingkan performa model ketika dilakukan optimasi hyperparameter menggunakan Genetic Algorithm (GA) dan Grid Search. Oleh karena itu, skenario eksperimen mencakup beberapa konfigurasi jumlah fitur (12, 11, 10, 9, dan 8 fitur), model GPR tanpa optimasi (*baseline*), serta model GPR dengan optimasi GA dan Grid Search.

Tujuan dari perancangan skenario ini adalah untuk mengetahui sejauh mana pengurangan fitur dan penerapan metode optimasi dapat mempengaruhi performa prediksi GPR, sesuai dengan rumusan masalah dalam penelitian ini.

4.5.1 Skenario GPR dengan 12 fitur

Pada skenario ini, dataset dibagi dengan proporsi 70:15:15, sehingga diperoleh 4309 baris untuk training set, 923 baris untuk validation set, dan 925 baris untuk testing set. Pada skenario ini, tidak dilakukan pemilihan fitur maupun optimasi hyperparameter. Sehingga kolom yang digunakan Adalah yaitu: PM2.5, PM10, NO2, O3, NH3, NO, Nox, CO, SO2, Benzene, Toluene, dan Xylene. Berikut kode program dari GPR pada gambar 4.6 berikut.

```
kernel = C(1.0, constant_value_bounds="fixed") * RBF(1.0, length_scale_bounds="fixed")
gpr = GaussianProcessRegressor(
    kernel=kernel, alpha=1e-2,
    optimizer='fmin_l_bfgs_b',
    n_restarts_optimizer=0,
    normalize_y=False, random_state=42)
```

Gambar 4. 6 Kode program GPR

Pada gambar 4.6 tersebut menunjukkan konfigurasi Gaussian Process Regression (GPR) dengan parameter default tanpa proses optimasi hiperparameter. Kernel yang digunakan adalah kombinasi ConstantKernel dan Radial Basis Function (RBF) dengan nilai konstanta dan length-scale yang ditetapkan (fixed), sehingga parameter kernel tidak diperbarui selama proses pelatihan. Nilai alpha ditetapkan sebesar 1×10^{-2} sebagai tingkat noise pada data. Proses optimasi internal dinonaktifkan dengan menetapkan jumlah restart optimizer sebesar nol, sehingga model dilatih menggunakan nilai parameter awal tanpa pencarian parameter terbaik.

4.5.2 Skenario GPR dengan 11 fitur

Pada skenario ini, jumlah dataset yang digunakan sama dengan penelitian sebelumnya dan tetap dibagi dengan proporsi 70:15:15, sehingga diperoleh 4309 baris untuk training set, 923 baris untuk validation set, dan 925 baris untuk testing set. Pada skenario ini dilakukan pemilihan fitur menggunakan korelasi, sehingga diperoleh 11 fitur dengan nilai korelasi tertinggi terhadap variabel target, yaitu: PM2.5, PM10, NO2, O3, NH3, NO, Nox, CO, SO2, Benzene, dan Toluene, dengan mengeluarkan fitur Xylene karena memiliki korelasi paling rendah. Untuk kode program sendiri sama persis seperti pada gambar 4.6.

4.5.3 Skenario GPR dengan 10 fitur

Pada skenario ini dilakukan pengurangan satu fitur dari daftar 11 fitur sebelumnya. Fitur dengan kontribusi korelasi terendah di antara kelompok fitur tersebut dihapus sehingga model hanya menggunakan 10 fitur utama. Pembagian dataset tetap mengikuti proporsi 70:15:15 dengan jumlah baris yang sama, yaitu 4309 untuk training, 923 untuk validation, dan 925 untuk testing. Tujuan skenario ini adalah mengevaluasi apakah pengurangan satu fitur berdampak pada peningkatan performa model GPR atau justru menurunkan akurasi prediksi. Kode program yang digunakan sama dengan yang ditampilkan pada gambar 4.6.

4.5.4 Skenario GPR dengan 9 fitur

Pada skenario ini dilakukan reduksi fitur lebih lanjut dengan menghapus satu fitur tambahan dari daftar 10 fitur sebelumnya. Fitur yang dihilangkan dipilih berdasarkan nilai korelasi yang lebih rendah dibandingkan fitur lain, sehingga tersisa 9 fitur dengan kontribusi korelasi tertinggi terhadap variabel target. Pembagian dataset tetap sama, yaitu 4309 data training, 923 data validation, dan 925 data testing. Tujuan skenario ini adalah untuk mengevaluasi apakah pengurangan fitur secara bertahap dapat menghasilkan model GPR yang lebih sederhana namun tetap mempertahankan performa prediksi yang kompetitif. Untuk kode program GPR yang digunakan sama dengan yang ditampilkan pada gambar 4.6.

4.5.5 Skenario GPR dengan 8 fitur

Pada skenario ini jumlah fitur dikurangi kembali sehingga model hanya menggunakan 8 fitur dengan korelasi paling kuat terhadap variabel target. Dua

hingga tiga fitur dengan korelasi terendah dari skenario awal dieliminasi untuk menguji dampak dimensionality reduction yang lebih agresif terhadap performa model. Pembagian dataset tetap menggunakan rasio 70:15:15, dengan total 4309 baris untuk training, 923 untuk validation, dan 925 untuk testing. Skenario ini dirancang untuk menganalisis hubungan antara kompleksitas model yang semakin kecil dengan stabilitas serta akurasi prediksi pada GPR. Kode program untuk GPR yang digunakan sama dengan yang ditampilkan pada gambar 4.6.

4.5.6 Skenario Algoirtma Regresi

Pada skenario ini dilakukan pengujian dataset menggunakan beberapa algoritma regresi lain, yaitu regresi linier, neural network berupa Multilayer Perceptron (MLP), serta Random Forest Regressor. Pembagian dataset tetap menggunakan proporsi 70:15:15, dengan jumlah data sebanyak 4.309 baris untuk data pelatihan (training), 923 baris untuk data validasi (validation), dan 925 baris untuk data pengujian (testing). Jumlah fitur yang digunakan ada 12 fitur, berikut kode program regresi linier di tunjukkan pada gambar 4.7, Multilayer Perceptron (MLP) pada gambar 4.8, dan Random Forest Regressor pada gambar 4.9

```python
from sklearn.linear_model import LinearRegression

lr_model = LinearRegression()
LR_START = time.time()
lr_model.fit(X_trainval_s, Y_trainval_s.ravel())
LR_DURATION = time.time() - LR_START
print(f"✅ Linear Regression training completed in {LR_DURATION:.4f} s")
train_pred_lr = inv_y_log(lr_model.predict(X_train_s))
val_pred_lr   = inv_y_log(lr_model.predict(X_val_s))
test_pred_lr  = inv_y_log(lr_model.predict(X_test_s))
y_all_actual_lr = np.concatenate([y_train.ravel(), y_val.ravel(), y_test.ravel()])
y_all_pred_lr   = np.concatenate([train_pred_lr.ravel(), val_pred_lr.ravel(), test_pred_lr.ravel()])
lr_eval_data = {
    "Train": calculate_metrics_research(y_train, train_pred_lr),
    "Val":   calculate_metrics_research(y_val, val_pred_lr),
    "Test":  calculate_metrics_research(y_test, test_pred_lr),
    "ALL":   calculate_metrics_research(y_all_actual_lr, y_all_pred_lr)
}
```

Gambar 4. 7 Kode program regresi linier

Pada gambar 4.7 menunjukkan implementasi algoritma Linear Regression dengan parameter default. Model dilatih menggunakan data pelatihan dan validasi yang telah dinormalisasi, serta waktu pelatihan dicatat untuk mengukur efisiensi komputasi. Setelah proses pelatihan selesai, model digunakan untuk melakukan prediksi pada data pelatihan, validasi, dan pengujian. Hasil prediksi kemudian dikembalikan ke skala asli menggunakan fungsi invers transformasi. Selanjutnya, performa model dievaluasi menggunakan beberapa metrik evaluasi pada masing-masing subset data, yaitu data pelatihan, validasi, pengujian, serta seluruh data secara keseluruhan.

```python
from sklearn.neural_network import MLPRegressor

mlp_model = MLPRegressor(random_state=42)
MLP_START = time.time()
mlp_model.fit(X_trainval_s, Y_trainval_s.ravel())
MLP_DURATION = time.time() - MLP_START
train_pred_mlp = inv_y_log(mlp_model.predict(X_train_s))
val_pred_mlp   = inv_y_log(mlp_model.predict(X_val_s))
test_pred_mlp  = inv_y_log(mlp_model.predict(X_test_s))
y_all_actual_mlp = np.concatenate([y_train.ravel(), y_val.ravel(), y_test.ravel()])
y_all_pred_mlp   = np.concatenate([train_pred_mlp.ravel(), val_pred_mlp.ravel(), test_pred_mlp.ravel()])
```

Gambar 4. 8 Kode program MLP

Pada gambar 4.8 menunjukkan penerapan algoritma *Multilayer Perceptron (MLP) Regression*. Proses pelatihan dilakukan menggunakan data pelatihan dan validasi yang telah melalui tahap normalisasi, serta waktu pelatihan dicatat sebagai indikator efisiensi komputasi. Setelah pelatihan selesai, model digunakan untuk menghasilkan prediksi pada data pelatihan, validasi, dan pengujian. Nilai hasil prediksi kemudian dikonversi kembali ke skala semula melalui fungsi invers transformasi. Seluruh hasil prediksi dari masing-masing subset data selanjutnya digabungkan untuk keperluan evaluasi kinerja model secara menyeluruh.

```
from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor(random_state=42, n_jobs=-1)
RF_START = time.time()
rf_model.fit(X_trainval_s, Y_trainval_s.ravel())
RF_DURATION = time.time() - RF_START
train_pred_rf = inv_y_log(rf_model.predict(X_train_s))
val_pred_rf   = inv_y_log(rf_model.predict(X_val_s))
test_pred_rf  = inv_y_log(rf_model.predict(X_test_s))
y_all_actual_rf = np.concatenate([y_train.ravel(), y_val.ravel(), y_test.ravel()])
y_all_pred_rf   = np.concatenate([train_pred_rf.ravel(), val_pred_rf.ravel(), test_pred_rf.ravel()])
```

Gambar 4. 9 Kode program Random Forest Regresor

Pada gambar 4.9 merepresentasikan penerapan *Random Forest Regressor* dengan konfigurasi standar. Model dilatih menggunakan data pelatihan dan validasi yang telah melalui proses penskalaan, dengan pemanfaatan komputasi paralel untuk meningkatkan efisiensi waktu pelatihan. Lama proses pelatihan dicatat sebagai ukuran kinerja komputasi. Selanjutnya, model digunakan untuk menghasilkan prediksi pada data pelatihan, validasi, dan pengujian, yang kemudian dikembalikan ke skala data asli. Seluruh hasil prediksi dari masing-masing subset data dikombinasikan untuk mendukung evaluasi kinerja model secara menyeluruh.

4.5.7 Skenario GPR Optimasi GA

Pada skenario ini, setelah mendapatkan jumlah fitur dengan nilai evaluasi yang baik, selanjutnya model GPR dioptimasi menggunakan Genetic Algorithm (GA) dengan tujuan memperoleh konfigurasi parameter yang menghasilkan kesalahan prediksi minimum. Parameter yang dioptimasi meliputi constant kernel (C) yang berfungsi mengontrol skala variansi global model, length-scale pada kernel Radial Basis Function (RBF) yang menentukan tingkat kehalusan fungsi regresi, serta noise level (N) pada WhiteKernel yang merepresentasikan tingkat noise atau ketidakpastian pengukuran pada data kualitas udara. Ketiga parameter tersebut dioptimasi dalam ruang pencarian logaritmik agar GA mampu mengeksplorasi nilai parameter dalam rentang yang luas secara efisien.

Dataset yang digunakan sama dengan skenario sebelumnya dan tetap dibagi menggunakan proporsi 70:15:15, yaitu 4309 baris untuk training, 923 baris untuk validation, dan 925 baris untuk testing. Berikut gambar 4.10 menunjukkan gambaran proses pencarian parameter menggunakan Genetic Algorithm (GA).

```
============================================================
🚀 STARTING HYBRID GA OPTIMIZATION (RBF + WHITE KERNEL)
============================================================
Gen 01/15 | Best RMSE: 0.111208 | Time: 98.15s
Gen 02/15 | Best RMSE: 0.111208 | Time: 83.63s
Gen 03/15 | Best RMSE: 0.110972 | Time: 84.55s
Gen 04/15 | Best RMSE: 0.110972 | Time: 83.77s
Gen 05/15 | Best RMSE: 0.110972 | Time: 84.25s
Gen 06/15 | Best RMSE: 0.110733 | Time: 84.29s
Gen 07/15 | Best RMSE: 0.110683 | Time: 83.75s
Gen 08/15 | Best RMSE: 0.110683 | Time: 83.46s
Gen 09/15 | Best RMSE: 0.110683 | Time: 83.76s
Gen 10/15 | Best RMSE: 0.110683 | Time: 83.25s
Gen 11/15 | Best RMSE: 0.110683 | Time: 83.06s
Gen 12/15 | Best RMSE: 0.110683 | Time: 83.22s
Gen 13/15 | Best RMSE: 0.110683 | Time: 83.31s
Gen 14/15 | Best RMSE: 0.110683 | Time: 82.79s
Gen 15/15 | Best RMSE: 0.110683 | Time: 84.82s
```

Gambar 4. 10 Proses pencarian parameter GA

Pada GA proses optimasi diawali dengan pembangkitan populasi awal yang terdiri dari 20 kromosom, di mana setiap kromosom merepresentasikan satu kombinasi parameter C, L, dan N. Evaluasi fitness dilakukan dengan melatih model GPR menggunakan data latih dan menghitung nilai Root Mean Square Error (RMSE) pada data validasi, dengan fitness didefinisikan sebagai nilai negatif RMSE sehingga proses optimasi berfokus pada minimisasi kesalahan prediksi. Mekanisme seleksi menggunakan metode tournament selection untuk memilih individu dengan performa terbaik sebagai induk. Selanjutnya, operasi crossover dan mutasi diterapkan secara agresif untuk menghasilkan variasi solusi baru, sementara strategi elitisme digunakan untuk mempertahankan individu terbaik pada setiap generasi. Proses evolusi ini dijalankan selama 15 generasi seperti pada gambar 4.10

hingga diperoleh kombinasi hiperparameter yang optimal. Parameter terbaik hasil GA kemudian digunakan sebagai nilai awal pada tahap fine-tuning menggunakan algoritma optimasi L-BFGS-B yang disediakan oleh GPR, sehingga diperoleh model akhir dengan performa prediksi yang lebih optimal dan stabil.

4.5.8 Skenario GPR Optimasi Grid Search

Pada skenario ini, optimasi hiperparameter model Gaussian Process Regression (GPR) dilakukan menggunakan metode Grid Search untuk memperoleh kombinasi parameter kernel yang menghasilkan performa prediksi terbaik. Parameter yang dioptimasi meliputi constant kernel (C) yang mengatur skala variansi fungsi regresi, length-scale pada kernel Radial Basis Function (RBF) yang menentukan tingkat kehalusan fungsi prediksi, serta noise level pada WhiteKernel yang merepresentasikan tingkat noise pada data kualitas udara. Grid Search mengevaluasi seluruh kombinasi parameter yang telah ditentukan sebelumnya dalam ruang pencarian diskrit, sehingga setiap kandidat konfigurasi kernel diuji secara sistematis.

Pembagian dataset tetap menggunakan proporsi 70:15:15, dengan 4309 baris untuk training, 923 baris untuk validation, dan 925 baris untuk testing. Berikut gambar proses optimasi oleh *grid search* pada gambar 4.11

No	Mean Fit Time	RMSE (Val)	Kernel Configuration
1	292.5675 s	0.2239	1**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.001)
2	98.5647 s	0.1288	1**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.1)
3	228.0005 s	0.2239	1**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.001)
4	145.5478 s	0.1288	1**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.1)
5	236.2194 s	0.2239	1**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.001)
6	212.6104 s	0.1288	1**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.1)
7	175.7577 s	0.2239	3.16**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.001)
8	198.2267 s	0.1288	3.16**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.1)
9	228.8856 s	0.2239	3.16**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.001)
10	157.5900 s	0.1288	3.16**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.1)
11	210.0493 s	0.2239	3.16**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.001)
12	214.0704 s	0.1288	3.16**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.1)
13	204.1986 s	0.2239	10**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.001)
14	134.7626 s	0.1288	10**2 * RBF(length_scale=1) + WhiteKernel(noise_level=0.1)
15	212.2754 s	0.2239	10**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.001)
16	227.3883 s	0.1288	10**2 * RBF(length_scale=10) + WhiteKernel(noise_level=0.1)
17	370.5983 s	0.2239	10**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.001)
18	281.6184 s	0.1288	10**2 * RBF(length_scale=100) + WhiteKernel(noise_level=0.1)

Gambar 4. 11 Proses optimasi *Grid Search*

Berbeda dengan GA yang mengandalkan proses evolusi, Grid Search melakukan pencarian secara menyeluruh (brute-force) pada ruang parameter yang lebih terbatas namun terstruktur. Pada gambar 4.11 Proses optimasi dilakukan dengan menggabungkan data latih dan data validasi menggunakan skema PredefinedSplit agar evaluasi setiap kandidat parameter dilakukan secara konsisten pada data validasi yang sama. Setiap kombinasi kernel dilatih menggunakan data latih, kemudian dievaluasi berdasarkan nilai Root Mean Square Error (RMSE) pada data validasi, yang digunakan sebagai metrik utama dalam pemilihan model terbaik. Konfigurasi dengan nilai RMSE terendah dipilih sebagai model optimal, selanjutnya dilakukan pelatihan ulang pada gabungan data latih dan validasi dengan penambahan proses fine-tuning menggunakan algoritma optimasi internal GPR. Model akhir kemudian dievaluasi secara menyeluruh pada data latih, validasi, dan data uji untuk menilai kemampuan generalisasi serta stabilitas performa prediksi.

4.6 Evaluasi kinerja

Model dievaluasi untuk mengukur kinerja dan efektivitasnya, di mana hasil evaluasi tersebut digunakan untuk menilai sejauh mana kombinasi *hyperparameter* yang diperoleh melalui GA dan Grid Search berpengaruh terhadap GPR. Proses evaluasi kinerja ini dilakukan secara menyeluruh dengan menggunakan metrik kuantitatif.

4.6.1 Evaluasi Skenario GPR

Pada skenario ini, algoritma GPR digunakan untuk melakukan pemodelan menggunakan dataset yang telah dipersiapkan dengan pembagian data sebesar 75%:15%:15%. Pemodelan dilakukan dengan memanfaatkan seluruh fitur tanpa penerapan proses optimasi hyperparameter. Hasil evaluasi model ditunjukkan pada Gambar 4.12

```
Training GPR Default : 5211 samples.
Split sizes (train,val,test): 4309 923 925

========================================
FEATURE INFORMATION
----------------------------------------
Total Features Used : 12
Feature List : ['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene']
----------------------------------------
✅ Training completed in 7.9615 s

=== GPR Metrics ===
|                    |        R |     RMSE |      MAE |     MAPE |
|:-------------------|---------:|---------:|---------:|---------:|
| Train              | 0.887045 | 18.3925  | 11.5021  | 11.8064  |
| Validation         | 0.930222 | 15.6552  | 10.9586  | 14.5532  |
| Test               | 0.881613 | 13.7797  | 10.4065  | 14.166   |
| All Data (Train+Val) | 0.909414 | 19.8675  | 11.2883  | 12.3667  |
```

Gambar 4. 12. Evaluasi Skenario GPR

Pada gambar 4.12 Hasil evaluasi pada GPR baseline menunjukkan bahwa proses training membutuhkan waktu sekitar 7,9 detik. Performa model berada pada nilai korelasi (R) yaitu 0,88 pada data latih, 0,93 pada data validasi, dan 0,88 pada data uji. Nilai RMSE, MAE, dan MAPE pada setiap subset menunjukkan tingkat

error yang relatif stabil, yang menandakan bahwa model tidak mengalami overfitting signifikan. Bahkan, RMSE pada data uji tercatat lebih rendah dibandingkan data latih, mengindikasikan generalisasi yang baik.

4.6.2 Evaluasi Skenario GPR dengan 11 fitur

Pada evaluasi ini, performa GPR dievaluasi setelah dilakukan pemilihan fitur menggunakan metode korelasi, di mana 11 fitur dengan hubungan paling kuat terhadap variabel AQI dipertahankan untuk proses pemodelan. Pengurangan jumlah fitur ini bertujuan untuk meningkatkan efisiensi model serta mengurangi potensi overfitting tanpa mengorbankan akurasi prediksi. Proses pembagian dataset tetap menggunakan proporsi 75% untuk data latih, 15% untuk validasi, dan 15% untuk pengujian secara berurutan. Hasil evaluasi model ditunjukkan pada gambar 4.13

```
Training GPR Default : 5232 samples.
Split sizes (train,val,test): 4309 923 925

========================================
Features Used : 11
========================================
Features List : PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene
----------------------------------------
Training GPR (11 Features)...
✅ Training completed in 3.6838 s

=== GPR Baseline Metrics ===
|                      |        R |     RMSE |      MAE |     MAPE |
|:---------------------|---------:|---------:|---------:|---------:|
| Train                | 0.886277 | 18.4514  | 11.5284  | 11.9162  |
| Validation           | 0.930043 | 25.6762  | 15.948   | 14.5541  |
| Test                 | 0.885201 | 14.0954  | 10.5837  | 14.3368  |
| All Data (Train+Val) | 0.905901 | 19.9173  | 12.3081  | 12.3816  |
```

Gambar 4. 13. Evaluasi Skenario GPR dengan 11 fitur

Pada gambar 4.13 Proses pelatihan dilakukan pada dataset berukuran 4.309 data latih, 923 data validasi, dan 925 data uji, dengan waktu pelatihan sekitar 960 detik. Nilai korelasi (R) konsisten di seluruh subset data. Pada data latih, model memperoleh nilai R sebesar 0.8873, sedangkan pada data validasi dan uji masing-masing mencapai 0.9300 dan 0.8852, menunjukkan kemampuan generalisasi yang stabil. Nilai RMSE dan MAE juga relatif seimbang antar subset, dengan RMSE data uji sebesar 18.451 dan MAE 11.528, yang menandakan error prediksi yang masih dapat diterima. Secara keseluruhan, nilai R pada seluruh data mencapai 0.9059 dengan MAPE 12.3816

4.6.3 Evaluasi Skenario GPR dengan 10 fitur

Pada skenario ini, pemodelan GPR dilakukan menggunakan 10 fitur terpilih hasil analisis korelasi dan eliminasi bertahap. Penghapusan satu fitur dilakukan untuk menguji dampak reduksi fitur terhadap kestabilan dan akurasi prediksi AQI. Skenario ini bertujuan menilai konsistensi performa model ketika jumlah fitur semakin disederhanakan. Pembagian data, proses normalisasi, dan tahapan pelatihan model dilakukan dengan prosedur yang sama seperti skenario sebelumnya. Berikut hasil evaluasi pada gambar 4.14

```
Training GPR Default : 5232 samples.
Split sizes (train,val,test): 4309 923 925

========================================
Features Used : 10
========================================
Features List : PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene
----------------------------------------
Training GPR (10 Features)...
✅ Training completed in 3.5198 s

=== GPR Baseline Metrics ===
```

	R	RMSE	MAE	MAPE
Train	0.879948	18.9127	11.8459	12.2446
Validation	0.925698	26.4705	16.2413	14.7962
Test	0.886955	14.2899	10.721	14.4681
All Data (Train+Val)	0.900465	20.45	12.6213	12.6947

Gambar 4. 14. Evaluasi Skenario GPR dengan 10 fitur

Berdasarkan gambar 4.14 Hasil evaluasi menunjukkan bahwa model GPR dengan 10 fitur mampu memberikan performa prediksi yang cukup baik dan stabil pada seluruh skenario pengujian. Nilai koefisien korelasi (R) berada pada kisaran 0,88–0,93. Performa terbaik terlihat pada data validasi dengan nilai R sebesar 0,9257. Nilai RMSE, MAE, dan MAPE pada data latih dan data uji relatif seimbang, yang mengindikasikan bahwa model tidak mengalami overfitting secara signifikan. Pada data uji, RMSE sebesar 14,29 dan MAPE sebesar 14,47% menunjukkan tingkat kesalahan prediksi yang masih dalam batas wajar untuk data kualitas udara yang bersifat fluktuatif.

4.6.4 Evaluasi Skenario GPR dengan 9 fitur

Evaluasi pada skenario ini menguji performa GPR ketika jumlah fitur kembali dikurangi menjadi 9 fitur utama. Pemilihan fitur dilakukan dengan mempertimbangkan kekuatan hubungan masing-masing variabel terhadap AQI. Evaluasi ini penting untuk memahami apakah reduksi fitur berdampak positif, negatif, atau netral terhadap akurasi dan generalisasi model. Seluruh proses preprocessing dan pembagian data dilakukan secara konsisten agar hasil evaluasi memiliki kesetaraan metodologis dengan skenario-skenario sebelumnya. Berikut hasil dari evalusi ditunjukkan pada gambar 4.15

```
Training GPR Default : 5232 samples.
Split sizes (train,val,test): 4309 923 925

========================================
Features Used : 9
========================================
Features List : PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3
----------------------------------------
Training GPR (9 Features)...
✅ Training completed in 3.3508 s

=== GPR Baseline Metrics ===
|                      |        R |     RMSE |      MAE |     MAPE |
|:---------------------|---------:|---------:|---------:|---------:|
| Train                | 0.874103 | 19.3325  | 12.1737  | 12.62    |
| Validation           | 0.918127 | 27.7149  | 16.712   | 15.158   |
| Test                 | 0.889491 | 14.6167  | 10.9304  | 14.6512  |
| All Data (Train+Val) | 0.89405  | 21.0551  | 12.9743  | 13.0678  |
```

Gambar 4. 15. Evaluasi Skenario GPR dengan 9 fitur

Pada gambar 4.15 Hasil evaluasi menunjukkan bahwa model Gaussian Process Regression (GPR) dengan menggunakan 9 fitur polutan udara menghasilkan nilai korelasi (R) sebesar 0,8741 dengan RMSE 19,33, MAE 12,17,

dan MAPE 12,62%. Pada data validasi, nilai R meningkat menjadi 0,9181, meskipun disertai kenaikan nilai RMSE menjadi 27,71 dan MAPE 15,16%, yang mengindikasikan adanya peningkatan kesalahan prediksi ketika model dihadapkan pada data yang tidak digunakan dalam proses pelatihan.

Pada data uji, model tetap menunjukkan performa yang konsisten dengan nilai R sebesar 0,8895, RMSE 14,62, MAE 10,93, dan MAPE 14,65%. Secara keseluruhan, evaluasi pada gabungan data latih dan validasi menghasilkan nilai R sebesar 0,8941 dengan RMSE 21,06 dan MAPE 13,07%.

4.6.5 Evaluasi Skenario GPR dengan 8 fitur

Pada skenario ini, model GPR dibangun menggunakan 8 fitur paling dominan berdasarkan nilai korelasi dan relevansi terhadap AQI. Tahap ini dirancang untuk menilai seberapa jauh model masih mampu mempertahankan performa prediksi dengan jumlah input yang semakin minimal. Skenario ini juga memberikan gambaran mengenai titik optimal trade-off antara kompleksitas fitur dan akurasi prediksi. Mekanisme split data dan normalisasi tetap mengikuti prosedur standar untuk menjaga konsistensi evaluasi antar skenario. Berikut hasil evaluasi pada gambar 4.16

```
========================================
Features Used : 8
========================================
Features List : PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2
----------------------------------------
Training GPR (8 Features)...
✅ Training completed in 3.3448 s

=== GPR Baseline Metrics ===
```

	R	RMSE	MAE	MAPE
Train	0.8549	20.6841	13.5236	14.2961
Validation	0.908306	29.1536	18.1841	17.8363
Test	0.864964	15.4924	10.8089	14.034
All Data (Train+Val)	0.879091	22.412	14.3458	14.9206

Gambar 4. 16. Evaluasi Skenario GPR dengan 8 fitur

Pada gambar 4.16 Hasil evaluasi menunjukkan bahwa model Gaussian Process Regression (GPR) dengan menggunakan 8 fitur polutan udara masih mampu menghasilkan performa prediksi yang cukup baik, meskipun terjadi penurunan kinerja dibandingkan penggunaan jumlah fitur yang lebih banyak. Pada data latih, model memperoleh nilai korelasi (R) sebesar 0,8549 dengan RMSE 20,68, MAE 13,52, dan MAPE 14,30%. Pada data validasi, nilai R tercatat sebesar 0,9083, namun disertai peningkatan nilai kesalahan prediksi dengan RMSE mencapai 29,15 dan MAPE 17,84%, yang mengindikasikan bahwa model menjadi kurang stabil ketika dihadapkan pada data yang tidak digunakan saat pelatihan. Pada data uji, performa model tetap relatif konsisten dengan nilai R sebesar 0,8650, RMSE 15,49, MAE 10,81, dan MAPE 14,03%. Secara keseluruhan, evaluasi pada gabungan data latih dan validasi menghasilkan nilai R sebesar 0,8791 dengan

RMSE 22,41 dan MAPE 14,92%, yang menunjukkan adanya penurunan performa global akibat pengurangan jumlah fitur.

Berdasarkan hasil evaluasi dari seluruh skenario, dapat disimpulkan bahwa performa terbaik ditunjukkan oleh model GPR yang menggunakan semua fitur atau sebanyak 12 fitur, dengan nilai R tertinggi di antara skenario lainnya. Oleh karena itu, langkah selanjutnya adalah melakukan proses optimasi terhadap model GPR tersebut untuk memperoleh kinerja prediksi yang lebih maksimal.

4.6.6 Evaluasi Skenario Algoritma Regresi Lain

Pada tahap ini dilakukan evaluasi dan perbandingan performa beberapa algoritma regresi, yaitu *Linear Regression, Neural Network (Multilayer Perceptron/MLP),* dan *Random Forest Regressor*, dalam memprediksi nilai Air Quality Index (AQI). Pengujian dilakukan menggunakan 12 fitur hasil evaluasi pada model GPR, dengan jumlah data yang sama seperti pada pengujian model GPR. Evaluasi dilakukan pada data latih, validasi, dan uji menggunakan metrik koefisien korelasi (R), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), dan Mean Absolute Percentage Error (MAPE), serta mempertimbangkan waktu pelatihan model sebagai indikator efisiensi komputasi.

Pada model *Linear Regression* hasil evaluasi yang diperoleh ditinjukkan pada gambar 4.17

```
=== LINEAR REGRESSION TABLE ===
| Split   |        R |      RMSE |       MAE |      MAPE |
|:--------|---------:|----------:|----------:|----------:|
| Train   |   0.8059 |   23.9412 |   15.9176 |     17.02 |
| Val     |   0.8816 |   33.1792 |   21.9768 |     25.92 |
| Test    |   0.8609 |   14.8213 |   10.2032 |     12.99 |
| ALL     |   0.8424 |   24.4781 |   15.9674 |     17.75 |
------------------------------------------------------------
⏱ Training Time: 0.0066 s
============================================================
```

Gambar 4. 17. Evaluasi algoritma *linier regression*

Hasil evaluasi pada gambar 4.17 menunjukkan bahwa Linear Regression memiliki waktu pelatihan yang sangat cepat, yaitu hanya 0,0066 detik, menjadikannya model paling efisien secara komputasi. Namun, dari sisi akurasi, performa model ini relatif terbatas. Pada data latih diperoleh nilai R sebesar 0,8059 dengan RMSE 23,94, sedangkan pada data validasi nilai R meningkat menjadi 0,8816 tetapi disertai RMSE yang cukup besar, yaitu 33,18, serta MAPE 25,92%. Pada data uji, performa relatif stabil dengan RMSE 14,82 dan MAPE 12,99%. Secara keseluruhan, evaluasi pada seluruh data menghasilkan nilai R 0,8424 dan RMSE 24,48, yang menunjukkan bahwa Linear Regression kurang mampu menangkap hubungan nonlinier kompleks pada data kualitas udara.

Berikut hasil evaluasi dari model algoritma *Neural Network (Multilayer Perceptron/MLP)* pada gambar 4.18

```
=== NEURAL NETWORK (MLP) ===
| Split   |        R |     RMSE |      MAE |     MAPE |
|:--------|---------:|---------:|---------:|---------:|
| Train   |   0.8226 |  22.7991 |  15.0759 |    16.29 |
| Val     |   0.8917 |  31.8812 |  20.7155 |    23.30 |
| Test    |   0.8458 |  15.0008 |  10.7469 |    14.36 |
| ALL     |   0.8548 |  23.4512 |  15.2710 |    17.05 |
------------------------------------------------------------
⏱ Training Time: 0.7568 s
------------------------------------------------------------
```

Gambar 4. 18. Evaluasi model algoritma (MLP)

Model Neural Network (MLP) pada gambar 4.18 menunjukkan peningkatan performa dibandingkan Linear Regression, dengan waktu pelatihan 0,7568 detik. Pada data latih, nilai R mencapai 0,8226 dengan RMSE 22,80, sedangkan pada data validasi nilai R meningkat menjadi 0,8917 dengan RMSE 31,88. Pada data uji, model menghasilkan RMSE 15,00 dan MAPE 14,36%, yang masih berada pada kisaran wajar. Evaluasi pada seluruh data menunjukkan nilai R 0,8548 dan RMSE 23,45, menandakan bahwa MLP mampu memodelkan hubungan nonlinier dengan lebih baik dibandingkan Linear Regression, meskipun peningkatan performanya masih terbatas.

Untuk model *Random Forest Regressor* hasil evaluasi yang diperoleh ditinjukkan pada gambar 4.19.

```
=== RANDOM FOREST REGRESSOR TABLE ===
| Split   |        R |      RMSE |       MAE |      MAPE |
|:--------|---------:|---------:|---------:|---------:|
| Train   |   0.9782 |   8.7487 |   4.8954 |      4.84 |
| Val     |   0.9835 |  13.2731 |   6.8878 |      5.83 |
| Test    |   0.8808 |  14.1660 |  10.0956 |     13.43 |
| ALL     |   0.9741 |  10.4941 |   5.9754 |      6.28 |
------------------------------------------------------------
Total Training Time: 1.9340 s
```

Gambar 4. 19. Evaluasi algoritma *random forest regressor*

Pada gambar 4.19 menunjukkan bahwa algoritma *random forest regressor* berbeda dengan dua model sebelumnya, *random forest regressor* menunjukkan performa yang sangat unggul, meskipun menggunakan konfigurasi default. Waktu pelatihan tercatat sebesar 1,9340 detik, masih tergolong efisien. Pada data latih, model mencapai nilai R 0,9782 dengan RMSE 8,75, sedangkan pada data validasi nilai R meningkat menjadi 0,9835 dengan RMSE 13,27. Pada data uji, model tetap menunjukkan performa yang baik dengan RMSE 14,17 dan MAPE 13,43%. Evaluasi keseluruhan menghasilkan nilai R 0,9741, RMSE 10,49, dan MAPE 6,28%, yang mengindikasikan kemampuan prediksi yang sangat akurat dan stabil. Namun demikian, perbedaan yang cukup besar antara error data latih dan data uji menunjukkan adanya kecenderungan overfitting ringan.

4.6.7 Evaluasi Skenario GPR Optimasi GA

Setelah melaui skenario pemilihan fitur, evaluasi selanjutnya untuk menguji performa GPR setelah melalui proses optimasi hyperparameter menggunakan Genetic Algorithm (GA). GA digunakan untuk mencari kombinasi optimal dari tiga parameter utama GPR, yaitu constant kernel (C), length-scale RBF (L), dan noise

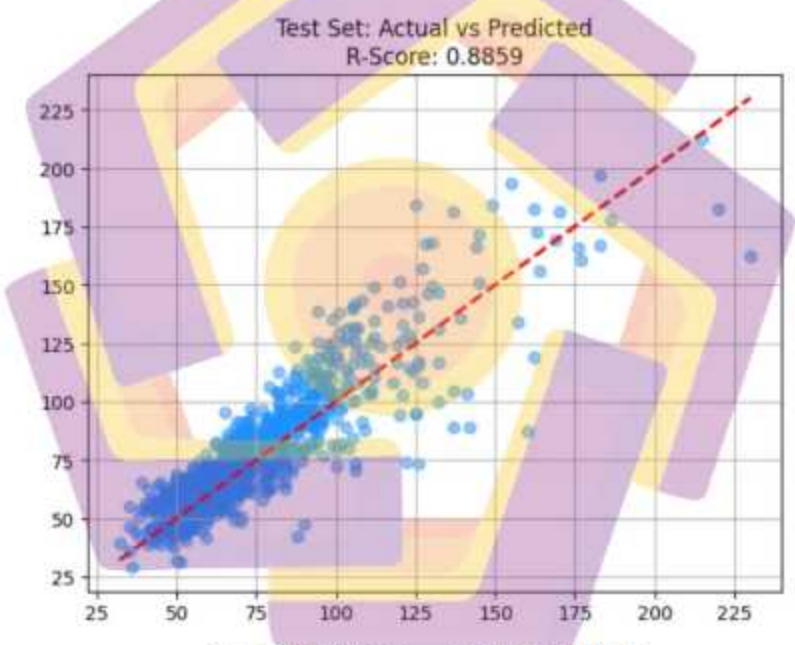
level (N). Berikut hasil Evaluasi dari algoritma GPR dengan optimasi GA pada gambar 4.20 Berikut.

```
=== FINAL EVALUATION TABLE ===
|       |      R |    RMSE |     MAE |    MAPE |
|:------|-------:|--------:|--------:|--------:|
| Train | 0.9009 | 17.331  | 10.7153 | 11.0924 |
| Val   | 0.9392 | 24.1563 | 14.9511 | 13.6508 |
| Test  | 0.8859 | 13.1573 |  9.7266 | 12.9842 |
| All   | 0.9181 | 18.7168 | 11.4625 | 11.5438 |
```

Gambar 4. 20. Evaluasi model GPR optimasi GA

Hasil evaluasi akhir pada gamabr 4.20 menunjukkan bahwa model Gaussian Process Regression (GPR) yang melalui proses optimasi Genetic Algorithm (GA) mampu menghasilkan performa prediksi yang lebih baik. Pada data latih, model mencapai nilai koefisien korelasi (R) sebesar 0,9009 dengan RMSE 17,33 dan MAPE 11,09%. Performa meningkat pada data validasi dengan nilai R mencapai 0,9392, meskipun disertai kenaikan RMSE menjadi 24,16, yang mencerminkan tantangan generalisasi pada data yang tidak digunakan selama pelatihan. Pada data uji, model tetap menunjukkan kemampuan generalisasi dengan nilai R sebesar 0,8859, RMSE 13,16, dan MAPE 12,98%. Evaluasi pada keseluruhan data menghasilkan nilai R 0,9181 dengan RMSE 18,72 dan MAPE 11,54%, menandakan bahwa model mampu mempertahankan akurasi yang konsisten secara global. Dari sisi komputasi, proses optimasi GA membutuhkan waktu yang cukup besar, yaitu sekitar 1.520 detik, dan pelatihan akhir dengan fine-tuning memerlukan waktu 2.584 detik, sehingga total waktu eksekusi mencapai 4.109 detik atau sekitar

1 jam, 8 menit, dan 29 detik. Konfigurasi kernel akhir yang diperoleh, yaitu ConstantKernel sebesar $0{,}927^2$, RBF dengan length-scale 0,534, serta WhiteKernel dengan noise level 0,149, menunjukkan bahwa kombinasi optimasi global GA menghasilkan model GPR dengan performa prediksi yang lebih optimal dan stabil. Berikut hasil scatter plot hubungan antara nilai AQI aktual dan AQI hasil prediksi model GPR yang telah dioptimasi menggunakan GA pada gambar 4.21



Gambar 4. 21. *Scatter plot* Evaluasi GPR-GA

Pada gambar 4.21 *scatter plot* tersebut menunjukkan hubungan antara nilai aktual dan nilai prediksi AQI pada data uji. Titik-titik data cenderung mengikuti garis diagonal merah, yang menandakan bahwa prediksi model cukup mendekati nilai aktual. Nilai R-Score sebesar 0,8859 mengindikasikan korelasi yang kuat

antara hasil prediksi dan data sebenarnya, sehingga model memiliki kemampuan generalisasi yang baik. Meskipun masih terlihat penyebaran titik yang lebih lebar pada nilai AQI tinggi, secara keseluruhan pola ini menunjukkan bahwa model mampu menangkap tren utama data kualitas udara dengan cukup akurat.

4.6.8 Evaluasi Skenario GPR Optimasi Grid Search

Pada skenario ini, optimasi hiperparameter Gaussian Process Regression (GPR) dilakukan menggunakan metode Grid Search, yaitu pendekatan pencarian ekshaustif yang mengevaluasi seluruh kombinasi parameter kernel yang telah ditentukan sebelumnya. Berbeda dengan Genetic Algorithm yang bersifat evolusioner dan adaptif, Grid Search melakukan pencarian secara sistematis untuk memastikan setiap kombinasi parameter diuji secara menyeluruh. Skenario ini dirancang sebagai pembanding langsung terhadap pendekatan GA guna menilai efektivitas metode pencarian ekshaustif dalam meningkatkan performa prediksi GPR. Hasil evaluasi performa model pada skenario ini ditunjukkan pada Gambar 4.22.

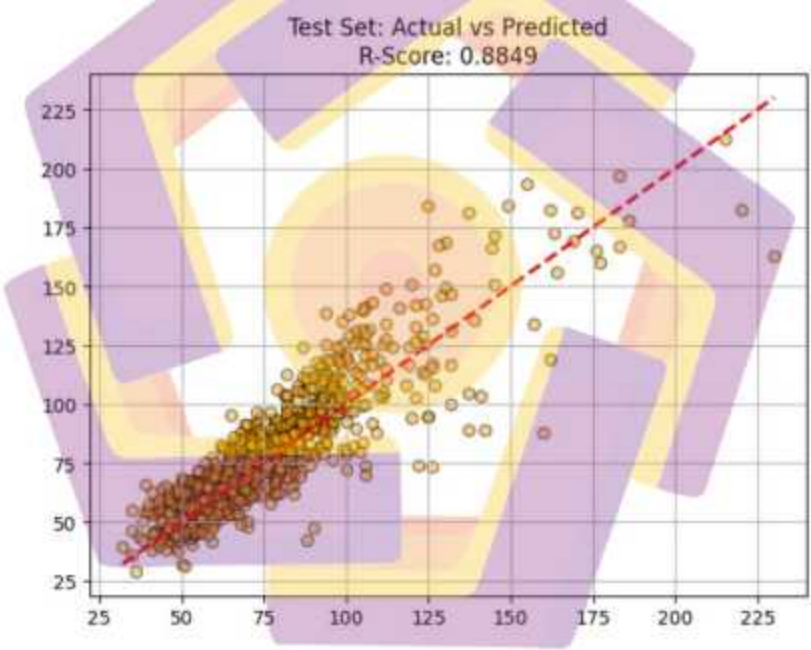
```
=== FINAL EVALUATION TABLE (GRID SEARCH) ===
|       |      R |    RMSE |     MAE |    MAPE |
|:------|-------:|--------:|--------:|--------:|
| Train | 0.9008 | 17.3393 | 10.7222 | 11.1    |
| Val   | 0.9391 | 24.1812 | 14.9649 | 13.6639 |
| Test  | 0.8849 | 13.2066 |  9.8024 | 13.1055 |
| All   | 0.9179 | 18.7287 | 11.4707 | 11.5523 |
```

Gambar 4. 22 Evaluasi Skenario GPR optimasi grid search

Berdasarkan Gambar 4.22 hasil evaluasi Gaussian Process Regression (GPR) dengan optimasi hiperparameter menggunakan Grid Search, model menunjukkan performa prediksi yang cukup baik dan konsisten pada seluruh skenario pengujian. Pada data latih, model memperoleh nilai koefisien korelasi (R) sebesar 0,9008 dengan RMSE 17,3393, MAE 10,7222, dan MAPE 11,10%. Pada data validasi, nilai R meningkat menjadi 0,9391, meskipun diikuti oleh peningkatan nilai kesalahan prediksi (RMSE 24,1812 dan MAE 14,9649). Selanjutnya, pada data uji, model tetap menunjukkan kemampuan generalisasi yang baik dengan nilai R sebesar 0,8849, serta nilai kesalahan yang relatif rendah, yaitu RMSE 13,2066, MAE 9,8024, dan MAPE 13,10%. Hasil ini mengonfirmasi bahwa konfigurasi kernel hasil Grid Search mampu menghasilkan prediksi yang stabil pada data yang belum pernah digunakan dalam proses pelatihan. Secara keseluruhan, performa model pada seluruh data (train dan validasi) menghasilkan nilai R sebesar 0,9179 dengan RMSE 18,7287, MAE 11,4707, dan MAPE 11,55%, yang menunjukkan efektivitas Grid Search dalam menemukan kombinasi hiperparameter yang optimal.

Dari sisi efisiensi komputasi, metode Grid Search membutuhkan waktu optimasi sebesar 5326,46 detik, diikuti dengan waktu pelatihan model akhir menggunakan fine-tuning L-BFGS-B selama 2158,04 detik, sehingga total waktu eksekusi mencapai 7491,97 detik. Waktu komputasi yang relatif tinggi ini mencerminkan karakteristik Grid Search yang mengevaluasi seluruh kombinasi kandidat secara menyeluruh, khususnya pada model GPR yang memiliki kompleksitas komputasi tinggi. Kernel akhir yang diperoleh adalah $0.927^2 \times \mathrm{RBF}(length\_scale = 0.537) + \mathrm{WhiteKernel}(noise\_level = 0.149)$, di mana

nilai *length-scale* yang kecil menunjukkan kemampuan model dalam menangkap pola lokal pada data, sedangkan nilai *noise level* yang cukup besar mengindikasikan keberadaan ketidakpastian atau noise yang signifikan pada data kualitas udara. Berikut hasil visualisasi scatter plot hubungan antara nilai AQI aktual dan AQI hasil prediksi model GPR yang telah dioptimasi menggunakan Grid Search pada gambar 4.23.



Gambar 4. 23. Scatter plot Evaluasi GPR-Grid search

Gambar 4.23 menunjukkan perbandingan nilai aktual dan nilai prediksi pada data uji (Test Set) untuk model Gaussian Process Regression (GPR) dengan optimasi Grid Search. Titik-titik sebar mayoritas berada dekat dengan garis diagonal merah (y = x), yang menandakan bahwa hasil prediksi model cukup

mendekati nilai aktual. Nilai R-score sebesar 0,8849 mengindikasikan adanya hubungan linear yang kuat antara nilai aktual dan prediksi, sehingga model memiliki kemampuan generalisasi yang baik pada data uji. Meskipun masih terlihat beberapa penyimpangan pada nilai ekstrem, secara keseluruhan model mampu menangkap pola utama data dengan cukup akurat.

4.6.9 Perbandingan Hasil Penelitian

Pada penelitian ini, hasil evaluasi kinerja model ditunjukkan pada Tabel 4.2 yang menyajikan perbandingan performa antara model GPR tanpa optimasi, GPR dengan optimasi Genetic Algorithm (GPR + GA), GPR dengan optimasi Grid Search (GPR + Grid Search) dan dengan algoritma regresi yang lain, dengan hasil pada keseluruhan data latih dan validasi (All).

Tabel 4. 2. Perbandingan Hasil Penelitian

Model	R	RMSE	MAPE	MAE	Waktu Total
GPR	0,9064	19,8675	11,898	11,5021	7,961 detik
GPR + GA	0,9181	18,7168	11,54	11,4625	4.109,25 detik
GPR + Grid Search	0,9179	18,7287	11,55	11,4707	7.491,96
Linier Regression	0,8424	23,4781	17,75	15,9674	0,007 detik
Neural Network	0,8548	23,4512	17,05	15,2710	0,757 detik
Random Forest	0,9741	10,4941	6,28	5,9754	1,934 detik

Berdasarkan Tabel 4.2 hasil perbandingan performa beberapa model prediksi, terlihat bahwa pendekatan Gaussian Process Regression (GPR) menunjukkan kinerja yang kompetitif dibandingkan metode konvensional maupun berbasis pembelajaran mesin lainnya. Model GPR tanpa optimasi menghasilkan nilai R sebesar 0,9064 dengan RMSE 19,8675, yang kemudian mengalami peningkatan performa setelah dilakukan optimasi hiperparameter. Penerapan Genetic Algorithm (GA) pada GPR mampu meningkatkan akurasi menjadi R = 0,9181 dengan

penurunan kesalahan prediksi (RMSE 18,7168; MAPE 11,54%) serta waktu komputasi yang lebih efisien dibandingkan Grid Search. Sementara itu, GPR dengan Grid Search juga menunjukkan performa yang sebanding (R = 0,9179; RMSE 18,7287), namun dengan biaya komputasi yang jauh lebih tinggi, sehingga dari sisi efisiensi GA lebih unggul sebagai metode optimasi hiperparameter.

Jika dibandingkan dengan model dasar, Linear Regression dan Neural Network menghasilkan performa yang lebih rendah, ditunjukkan oleh nilai R di bawah 0,86 dan tingkat error yang lebih tinggi, meskipun waktu komputasinya lebih singkat. Di sisi lain, Random Forest menunjukkan performa paling unggul dengan nilai R sebesar 0,9741, serta nilai RMSE, MAE, dan MAPE terendah, yang menandakan kemampuan model ini dalam menangkap hubungan nonlinier data secara lebih efektif dengan waktu komputasi yang masih relatif efisien. Meskipun demikian, hasil yang diperoleh menunjukkan bahwa GPR, khususnya dengan optimasi GA, masih memiliki potensi untuk terus dikembangkan, baik melalui perluasan ruang hiperparameter, pemilihan kernel yang lebih kompleks, maupun integrasi dengan metode ensemble, sehingga dapat mendekati atau bahkan menyaingi performa model Random Forest pada penelitian selanjutnya.

BAB 5
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

a. Genetic Algorithm (GA) maupun Grid Search mampu melakukan optimasi hyperparameter pada algoritma Gaussian Process Regression (GPR) secara efektif. Kedua metode optimasi menghasilkan konfigurasi kernel terbaik yang pada akhirnya konvergen pada struktur kernel yang sama, yaitu kombinasi Constant Kernel, RBF, dan WhiteKernel, setelah dilakukan proses pelatihan dan optimasi lanjutan. Hal ini menunjukkan bahwa baik pendekatan evolusioner (GA) maupun pencarian ekshaustif (Grid Search) mampu mengarahkan model GPR menuju solusi hiperparameter yang stabil dan optimal.

b. Akurasi prediksi, model GPR tanpa optimasi, GPR dengan optimasi GA, dan GPR dengan optimasi Grid Search menunjukkan performa yang sangat konsisten. Algoritma GPR dengan kombinasi GA mendapat nilai koefisien korelasi (R) berada pada kisaran 0,918, dengan nilai RMSE sekitar 18,7, MAE sekitar 11,46, dan MAPE sekitar 11,54%. Hasil ini menunjukkan bahwa penerapan optimasi hyperparameter tidak secara signifikan meningkatkan nilai evaluasi numerik dibandingkan GPR tanpa optimasi, namun berhasil memastikan bahwa model bekerja pada konfigurasi parameter yang optimal dan stabil. Selain itu, hasil evaluasi pada data uji menunjukkan bahwa model memiliki kemampuan

generalisasi yang baik dalam memprediksi kualitas udara pada data yang belum pernah dilihat sebelumnya.

Secara keseluruhan, penelitian ini menyimpulkan bahwa optimasi hyperparameter menggunakan Genetic Algorithm dan Grid Search memberikan kontribusi penting dalam menjamin kualitas dan kestabilan model GPR, meskipun peningkatan nilai evaluasi numerik relatif kecil. Dengan demikian, optimasi hyperparameter tetap menjadi tahap yang krusial dalam pengembangan model prediksi kualitas udara berbasis GPR, terutama untuk memastikan bahwa performa model yang diperoleh merupakan hasil dari konfigurasi parameter yang optimal.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa hal yang dapat menjadi arah pengembangan untuk penelitian selanjutnya. :

a. Penelitian selanjutnya disarankan untuk mengeksplorasi penggunaan algoritma regresi yang lain dan metode optimasi hiperparameter lain selain Genetic Algorithm dan Grid Search, seperti Bayesian Optimization, Particle Swarm Optimization (PSO), atau Differential Evolution, guna membandingkan efisiensi pencarian parameter dan waktu komputasi pada algoritma Gaussian Process Regression.

b. Pengembangan model dapat dilakukan dengan menambahkan variasi kernel GPR yang lebih kompleks, seperti kombinasi Matern Kernel, Rational Quadratic, atau kernel komposit multi-skala, guna menangkap karakteristik data kualitas udara yang bersifat non-linear dan dinamis secara lebih mendalam.

DAFTAR PUSTAKA

[1] Perdinan *et al.*, "Kajian Pemantauan Kualitas Udara di Provinsi DKI Jakarta Tahun 2023," *Dinas Lingkung. Hidup Provinsi DKI Jakarta*, pp. 1–127, 2023, [Online]. Available: https://lingkunganhidup.jakarta.go.id/files/laporan/LAPORAN_KAJIAN_PEMANTAUAN_KUALITAS_UDARA2023.pdf

[2] Health Effects Institute, "State of Global Air 2019," *Heal. Eff. Institute.*, p. 24, 2019, doi: https://www.stateofglobalair.org/sites/default/files/soga_2019_report.pdf.

[3] Y. Ristia, "Pengendalian Pencemaran Udara," *J. El-Thawalib*, vol. 3, no. 2, pp. 375–386, 2022, doi: 10.24952/el-thawalib.v3i2.5331.

[4] Z. Zhang, S. Zhang, C. Chen, and J. Yuan, "A systematic survey of air quality prediction based on deep learning," *Alexandria Eng. J.*, vol. 93, no. September 2023, pp. 128–141, 2024, doi: 10.1016/j.aej.2024.03.031.

[5] US EPA, "Technical Assistance Document for the Reporting of Daily Air Quality – the Air Quality Index (AQI)," *Environ. Prot.*, p. 22, 2018, [Online]. Available: https://airnowtest.epa.gov/sites/default/files/2018-05/aqi-technical-assistance-document-may2016.pdf

[6] A. Samad, S. Garuda, U. Vogt, and B. Yang, "Air pollution prediction using machine learning techniques – An approach to replace existing monitoring stations with virtual monitoring stations," *Atmos. Environ.*, vol. 310, no. April, p. 119987, 2023, doi: 10.1016/j.atmosenv.2023.119987.

[7] M. Méndez, M. G. Merayo, and M. Núñez, *Machine learning algorithms to forecast air quality: a survey*, vol. 56, no. 9. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10424-4.

[8] A. Yenkikar, V. P. Mishra, M. Bali, and T. Ara, "Explainable forecasting of air quality index using a hybrid random forest and ARIMA model," *MethodsX*, vol. 15, no. May, p. 103517, 2025, doi: 10.1016/j.mex.2025.103517.

[9] Y. Liang *et al.*, "AirFormer: Predicting Nationwide Air Quality in China with Transformers," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 14329–14337, 2023, doi: 10.1609/aaai.v37i12.26676.

[10] J. Luo and Y. Gong, "Air pollutant prediction based on ARIMA-WOA-LSTM model," *Atmos. Pollut. Res.*, vol. 14, no. 6, p. 101761, 2023, doi: 10.1016/j.apr.2023.101761.

[11] Z. Zhao, J. Wu, F. Cai, S. Zhang, and Y. G. Wang, "A hybrid deep learning framework for air quality prediction with spatial autocorrelation during the COVID-19 pandemic," *Scientific Reports*, vol. 13, no. 1. 2023. doi: 10.1038/s41598-023-28287-8.

[12] W. Zhang, Z. Chen, J. Miao, and X. Liu, "Research on Graph Neural Network in Stock Market," *Procedia Comput. Sci.*, vol. 214, no. C, pp. 786–

792, 2022, doi: 10.1016/j.procs.2022.11.242.

[13] Y. Huang, J. Yu, X. Dai, Z. Huang, and Y. Li, "Air-Quality Prediction Based on the EMD–IPSO–LSTM Combination Model," *Sustain.*, vol. 14, no. 9, 2022, doi: 10.3390/su14094889.

[14] D. Pruthi and Y. Liu, "Low-cost nature-inspired deep learning system for PM2.5 forecast over Delhi, India," *Environ. Int.*, vol. 166, no. June, p. 107373, 2022, doi: 10.1016/j.envint.2022.107373.

[15] D. Iskandaryan, F. Ramos, and S. Trilles, "Graph Neural Network for Air Quality Prediction: A Case Study in Madrid," *IEEE Access*, vol. 11, no. 2, pp. 2729–2742, 2023, doi: 10.1109/ACCESS.2023.3234214.

[16] R. S. Suri *et al.*, "Air Quality Prediction-A Study Using Neural Network Based Approach," *J. Soft Comput. Civ. Eng.*, vol. 7, no. 1, pp. 93–113, 2023, doi: 10.22115/SCCE.2022.352017.1488.

[17] H. Ahmadi, M. Rodehutscord, and W. Siegert, "Bi-objective optimization of nutrient intake and performance of broiler chickens using Gaussian process regression and genetic algorithm," *Front. Anim. Sci.*, vol. 4, no. March, pp. 1–11, 2023, doi: 10.3389/fanim.2023.1042725.

[18] M. Rice *et al.*, "Outdoor air pollution and your health," *Am. J. Respir. Crit. Care Med.*, vol. 204, no. 7, pp. P13–P14, 2021, doi: 10.1164/rccm.2046P13.

[19] M. S. Caywood, D. M. Roberts, J. B. Colombe, H. S. Greenwald, and M. Z. Weiland, "Gaussian process regression for predictive but interpretable machine learning models: An example of predicting mental workload across tasks," *Front. Hum. Neurosci.*, vol. 10, no. January, pp. 1–19, 2017, doi: 10.3389/fnhum.2016.00647.

[20] K. Isik and S. E. Alptekin, "A benchmark comparison of Gaussian process regression, support vector machines, and ANFIS for man-hour prediction in power transformers manufacturing," *Procedia Comput. Sci.*, vol. 207, pp. 2567–2577, 2022, doi: 10.1016/j.procs.2022.09.315.

[21] J. Eberhard and V. Geissbuhler, *Konservative und operative therapie bei harninkontinenz, deszensus und urogenital-beschwerden*, vol. 7, no. 5. 2000.

[22] N. Quadrianto, K. Kersting, and Z. Xu, "Gaussian Process," Encyclopedia of Machine Learning. [Online]. Available: https://scikit-learn.org/stable/modules/gaussian_process.html

[23] C. K. I. Williams and D. Barber, "Bayesian classification with gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1342–1351, 1998, doi: 10.1109/34.735807.

[24] S. Katoch, S. S. Chauhan, and V. Kumar, *Katoch2021_Article_AReviewOnGeneticAlgorithmPastP.pdf*, vol. 80. Multimedia Tools and Applications, 2021.

[25] T. Hai *et al.*, "Optimizing Gaussian process regression (GPR) hyperparameters with three metaheuristic algorithms for viscosity prediction

of suspensions containing microencapsulated PCMs," *Sci. Rep.*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-71027-9.

[26] T. Wang, X. Shao, D. Qin, K. Huang, M. Yao, and Y. Duan, "An Adaptive GPR-Based Multidisciplinary Design Optimization of Structural and Control Parameters of Intelligent Bus for Rollover Stability," *Mathematics*, vol. 13, no. 5, 2025, doi: 10.3390/math13050782.

[27] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[28] D. K. Barupal and O. Fiehn, "Generating the blood exposome database using a comprehensive text mining and database fusion approach," *Environ. Health Perspect.*, vol. 127, no. 9, pp. 2825–2830, 2019, doi: 10.1289/EHP4713.

[29] C. Ferreira, *Parameter optimization*, vol. 21. 2006. doi: 10.1007/3-540-32849-1_8.

[30] R. Kohavi and S. Edu, "10.1.1.76.1253," pp. 1–7, 2006, [Online]. Available: papers://5e3e5e59-48a2-47c1-b6b1-a778137d3ec1/Paper/p2015

[31] S. Manzhos and M. Ihara, "On the optimization of hyperparameters in Gaussian process regression with the help of low-order high-dimensional model representation," pp. 1–16, 2021, doi: 10.1007/s10910-022-01407-x.

[32] G. Kopsiaftis, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Mantoglou, "Gaussian process regression tuned by Bayesian optimization for seawater intrusion prediction," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/2859429.

[33] B. Ratner, "The correlation coefficient: Its values range between 1/1, or do they," *J. Targeting, Meas. Anal. Mark.*, vol. 17, no. 2, pp. 139–142, 2009, doi: 10.1057/jt.2009.5.

[34] S. Nakagawa, P. C. D. Johnson, and H. Schielzeth, "The coefficient of determination R^2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded," *J. R. Soc. Interface*, vol. 14, no. 134, 2024, doi: 10.1098/rsif.2017.0213.

[35] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *Int. J. Forecast.*, vol. 32, no. 3, pp. 669–679, 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[36] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Front. Bioinforma.*, vol. 2, no. June, pp. 1–17, 2022, doi: 10.3389/fbinf.2022.927312.

[37] K. P. Singh, N. Basant, and S. Gupta, "Support vector machines in water quality management," *Anal. Chim. Acta*, vol. 703, no. 2, pp. 152–162, 2011, doi: 10.1016/j.aca.2011.07.027.

[38] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality

reduction," *Proc. Natl. Conf. Artif. Intell.*, vol. 2, pp. 677–682, 2008.

[39] L. Fang *et al.*, "Development of a regional feature selection-based machine learning system (RFSML v1.0) for air pollution forecasting over China," *Geosci. Model Dev.*, vol. 15, no. 20, pp. 7791–7807, 2022, doi: 10.5194/gmd-15-7791-2022.