

**TUGAS AKHIR - EF234801**

# **PENGEMBANGAN MODEL PREDIKSI HARGA PASAR PUPIK DENGAN METODE STATISTIK DAN *MACHINE LEARNING***

**HILMI ZHARFAN RACHMADI**

**NRP 5025201268**

**Dosen Pembimbing 1**

**Dr. Agus Budi Raharjo, S.Kom, M.Kom.**

**NIP 1990202011022**

**Dosen Pembimbing 2**

**Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.**

**NIP 197804102003122001**

**Program Studi S1 Teknik Informatika**

**Departemen Teknik Informatika**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya**

*(Halaman ini Sengaja Dikосongkan)*



**TUGAS AKHIR - EF234801**

# **PENGEMBANGAN MODEL PREDIKSI HARGA PASAR PUPUK DENGAN METODE STATISTIK DAN *MACHINE LEARNING***

**HILMI ZHARFAN RACHMADI  
NRP 5025201268**

**Dosen Pembimbing  
Dr. Agus Budi Raharjo, S.Kom, M.Kom.  
NIP 1990202011022**

**Dosen Pembimbing 2  
Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.  
NIP 197804102003122001**

**Program Studi S1 Teknik Informatika**

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024

*(Halaman ini Sengaja Dikosongkan)*



**UNDERGRADUATE THESIS - EF234801**

**DEVELOPMENT OF FERTILIZER MARKET PRICE  
FORECASTING MODEL WITH STATISTICAL AND MACHINE  
LEARNING METHODS**

**HILMI ZHARFAN RACHMADI  
NRP 5025201268**

**Advisor  
Dr. Agus Budi Raharjo, S.Kom, M.Kom.  
NIP 1990202011022**

**Co-Advisor  
Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.  
NIP 197804102003122001**

**Study Program Bachelor of Informatics  
Department of Informatics  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2024**

*(Halaman ini Sengaja Dikosongkan)*

## LEMBAR PENGESAHAN

### PENGEMBANGAN MODEL PREDIKSI HARGA PASAR PUPUK DENGAN METODE STATISTIK DAN *MACHINE LEARNING*

#### TUGAS AKHIR


Diajukan untuk memenuhi salah satu syarat  
Memperoleh gelar Sarjana Komputer pada  
Program Studi S-1 Teknik Informatika  
Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh : **HILMI ZHARFAN RACHMADI**


NRP. 5025201268

Disetujui oleh Tim Penguji Tugas Akhir:


1. Dr. Agus Budi Raharjo, S.Kom, M.Kom.

Pembimbing 

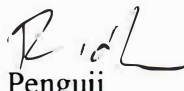
2. Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.

Ko-pembimbing 

3. Fajar Baskoro, S.Kom., M.T.

Penguji 

4. Ridho Rahman Hariadi, S.Kom, M.Sc.

Penguji 

**SURABAYA**  
**Juli, 2024**

*(Halaman ini Sengaja Dikosongkan)*



# APPROVAL SHEET

## DEVELOPMENT OF FERTILIZER MARKET PRICE FORECASTING MODEL WITH STATISTICAL AND MACHINE LEARNING METHODS

### UNDERGRADUATE THESIS

Submitted to fulfill one of the requirements  
for obtaining a degree Bachelor of Informatics at  
Undergraduate Study Program of Informatics Engineering  
Department of Informatics  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember

By: **HILMI ZHARFAN RACHMADI**

NRP. 5025201268

Approved by Final Project Examiner Team:

1. Dr. Agus Budi Raharjo, S.Kom, M.Kom.
2. Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.
3. Fajar Baskoro, S.Kom., M.T.
4. Ridho Rahman Hariadi, S.Kom, M.Sc.

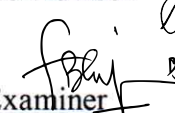
Advisor



Co-Advisor



Examiner



Examiner



**SURABAYA**  
**July, 2024**

*(Halaman ini Sengaja Dikosongkan)*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Hilmi Zharfan Rachmadi / 5025201268

Departemen : Teknik Informatika

Dosen Pembimbing I / NPP : Dr. Agus Budi Raharjo, S.Kom., M.Kom. /  
1990202011022

Dosen Pembimbing II / NIP : Prof.Ir.Dr. Diana Purwitasari, S.Kom., M.Sc. /  
197804102003122001

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Pengembangan Model Prediksi Harga Pasar Pupuk dengan Metode Statistik dan *Machine Learning*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima saksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 25 Juli 2024

Mahasiswa



(Hilmi Zharfan Rachmadi)

NRP. 5025201268

Mengetahui

Dosen Pembimbing I



(Dr. Agus Budi Raharjo, S.Kom., M.Kom.)

NPP. 1990202011022

Dosen Pembimbing II



(Prof.Ir.Dr. Diana Purwitasari, S.Kom., M.Sc.)

NIP. 197804102003122001

*(Halaman ini Sengaja Dikosongkan)*

## STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Hilmi Zharfan Rachmadi / 5025201268  
Department : Informatics  
Advisor I / NIP : Dr. Agus Budi Raharjo, S.Kom., M.Kom. /  
1990202011022  
Advisor II / NIP : Prof.Ir.Dr. Diana Purwitasari, S.Kom., M.Sc. /  
197804102003122001

Hereby declare that Final Project with the title of “Development of Fertilizer Market Price Forecasting Model with Statistical and Machine Learning Methods” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, 25<sup>th</sup> July 2024

Student



(Hilmi Zharfan Rachmadi)

NRP. 5025201268

Acknowledge

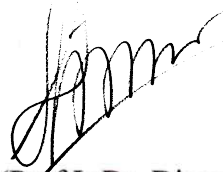
Advisor I



(Dr. Agus Budi Raharjo, S.Kom., M.Kom.)

NPP. 1990202011022

Advisor II



(Prof.Ir.Dr. Diana Purwitasari, S.Kom., M.Sc.)

NIP. 197804102003122001

*(Halaman ini Sengaja Dikosongkan)*

## PENGEMBANGAN MODEL PREDIKSI HARGA PASAR PUPUK DENGAN METODE STATISTIK DAN *MACHINE LEARNING*

**Nama Mahasiswa / NRP** : Hilmi Zharfan Rachmadi / 5025201268  
**Departemen** : Informatika FTEIC - ITS  
**Dosen Pembimbing** : 1. Dr. Agus Budi Raharjo, S.Kom, M.Kom.  
2. Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.

### Abstrak

Industri pupuk global telah menghadapi volatilitas harga yang signifikan selama lebih dari satu dekade terakhir. Fluktuasi harga bahan baku, dinamika geopolitik, kebijakan ekspor, dan gangguan rantai pasokan menjadi faktor-faktor eksternal yang memengaruhi harga pupuk. Rusia, sebagai eksportir pupuk terbesar di dunia, berperan penting sebagai pemasok lima jenis pupuk: *potassium chloride* (KCL), *rock phosphate* (RP), *diammonium phosphate* (DAP), urea, dan *triple superphosphate* (TSP). Konflik geopolitik yang melibatkan Rusia telah menghambat pasokan pupuk-pupuk tersebut, sehingga meningkatkan fluktuasi harga. Dalam kondisi ini, PT X, salah satu produsen pupuk terbesar di Asia Tenggara, menghadapi tantangan dalam memprediksi harga pasar pupuk dengan akurat sebagai bahan pertimbangan dalam menetapkan harga jual yang kompetitif setiap bulan.

Penelitian ini mengimplementasikan model prediksi harga pasar pupuk DAP, KCL, RP, TSP, dan urea menggunakan model statistik AutoARIMA dan model *machine learning* XGBoost. Kedua model digunakan untuk memprediksi harga kelima jenis pupuk selama 3, 6, dan 12 bulan mendatang. Hasil prediksi dievaluasi menggunakan metrik kesalahan *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE). Untuk meminimalkan metrik kesalahan, dilakukan eksplorasi terkait pemrosesan dan transformasi data yang meliputi penanganan data pencilon dengan imputasi median, *power transform*, *differencing*, *scaling*, dan pembuatan variabel *lag*.

Setelah uji coba pada kelima dataset harga pupuk, model XGBoost menghasilkan metrik kesalahan yang lebih rendah dibandingkan AutoARIMA dalam sebagian besar skenario, baik untuk rentang prediksi pendek, maupun panjang. Kedua model menunjukkan peningkatan metrik kesalahan seiring bertambahnya rentang waktu prediksi dengan AutoARIMA mengalami peningkatan yang lebih signifikan. Berdasarkan kriteria MAPE, model XGBoost konsisten menghasilkan prediksi yang tergolong "Sangat Baik" untuk semua jenis pupuk dan rentang waktu, sedangkan performa AutoARIMA bervariasi tergantung jenis pupuk dan rentang waktu prediksi. Keunggulan XGBoost mungkin disebabkan oleh kemampuannya

dalam menangani hubungan dan pola yang kompleks serta nonlinear dalam data harga pupuk. Meskipun demikian, AutoARIMA masih menunjukkan potensi untuk prediksi jangka pendek, terutama untuk pupuk RP yang cenderung lebih mudah diprediksi. Parameter terkait imputasi data pencilan dan orde *differencing* menjadi dua parameter terpenting untuk model XGBoost, sedangkan parameter imputasi data dan *power transform* menjadi yang terpenting untuk AutoARIMA. Menariknya, penggunaan data pelatihan tanpa imputasi data pencilan lebih sering menghasilkan metrik kesalahan terendah. Hal ini mengindikasikan bahwa data harga pupuk mengandung kenaikan eksponensial yang membuat penerapan imputasi data pencilan cenderung berdampak negatif terhadap akurasi prediksi.

**Kata kunci:** *Multi-step time series forecasting*, ARIMA, XGBoost, prediksi harga pasar pupuk



# DEVELOPMENT OF FERTILIZER MARKET PRICE FORECASTING MODEL WITH STATISTICAL AND MACHINE LEARNING METHODS

**Student Name / NRP** : Hilmi Zharfan Rachmadi / 5025201268  
**Department** : Informatika FTEIC - ITS  
**Advisor** : 1. Dr. Agus Budi Raharjo, S.Kom, M.Kom.  
2. Prof. Dr. Ir. Diana Purwitasari, S.Kom., M.Sc.

## Abstract

The global fertilizer industry has experienced significant price volatility over the past decade. Factors such as raw material price fluctuations, geopolitical dynamics, export policies, and supply chain disruptions influence fertilizer prices. Russia, the world's largest fertilizer exporter, plays a crucial role in supplying five types of fertilizers: potassium chloride (KCL), rock phosphate (RP), diammonium phosphate (DAP), urea, and triple superphosphate (TSP). Geopolitical conflicts involving Russia have disrupted the supply of these fertilizers, leading to increased price volatility. In this context, PT X, one of the largest fertilizer producers in Southeast Asia, faces the challenge of accurately predicting fertilizer market prices to help set competitive selling prices each month.

This research implements a market price prediction model for DAP, KCL, RP, TSP, and urea fertilizers using AutoARIMA and XGBoost, representing a statistical model and a machine learning model. Both models predict the prices of these fertilizers for the next 3, 6, and 12 months. The prediction results were evaluated using MAE, RMSE, and MAPE error metrics. To minimize these errors, various data processing and transformation techniques were explored, including handling outliers with median imputation, power transform, differencing, scaling, and generating lag variables.

After testing on all five fertilizer price datasets, the XGBoost model produced lower error metrics than AutoARIMA in most scenarios for both short and long prediction ranges. Both models showed increased error metrics as the prediction time span increased, with AutoARIMA experiencing a more significant rise. Based on the MAPE criterion, the XGBoost model consistently produced predictions classified as "Excellent" for all fertilizer types and timescales, while AutoARIMA's performance varied depending on the fertilizer type and prediction timescale. The superiority of XGBoost may be due to its ability to handle complex and nonlinear relationships and patterns in fertilizer price data. Nonetheless, AutoARIMA still shows potential for short-term predictions, especially for RP fertilizers, which tend to be more predictable. Parameters related to outlier data imputation and differencing order are the most

important for the XGBoost model, while data imputation and power transform parameters are the most crucial for AutoARIMA. Interestingly, using training data without outlier data imputation often resulted in the lowest error metric. This might suggest that the fertilizer price data contains an exponential increase, making the application of outlier data imputation likely to negatively impact prediction accuracy.

**Keywords:** *Multi-step* time series forecasting, ARIMA, XGBoost, fertilizer market price forecasting.

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah subhana wa ta'ala atas segala rahmat dan karunia-Nya, sehingga penulis mampu menyelesaikan Tugas Akhir yang berjudul:

### **“PENGEMBANGAN MODEL PREDIKSI HARGA PASAR PUPUK DENGAN METODE STATISTIK DAN *MACHINE LEARNING*”**

Penyusunan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi S-1 Teknik Informatika, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Melalui buku ini, penulis juga hendak menyampaikan rasa terima kasih kepada mereka yang telah membantu, baik langsung, maupun tidak langsung, dalam pengerjaan Tugas Akhir:

1. Allah subhana wa ta'ala yang telah menciptakan penulis beserta keluarga dan memudahkan penulis dalam menyelesaikan Tugas Akhir ini.
2. Nabi Muhammad shallallahu 'alaihi wasallam yang telah menjadi teladan kehidupan penulis.
3. Eyang Kakung, Ir. Adjitomo Dwidjodarsono rahimahullah, yang telah berjasa merawat dan membesarkan penulis.
4. Keluarga penulis, terutama Bapak, Ibu, dan Adik yang senantiasa memberikan dukungan, doa, dan semangat.
5. Bapak Dr. Agus Budi Raharjo, S.Kom, M.Kom. dan Ibu Prof. Ir. Dr. Diana Purwitasari, S.Kom., M.Sc. selaku dosen pembimbing dan ko-pembimbing atas bimbingan, arahan, dan pengawasannya dalam proses pengerjaan Tugas Akhir ini.
6. Anak Agung Yatestha Parwata, Selomita Zhafīrah, dan Tegar Ganang Satrio Priambodo yang telah menjadi teman diskusi selama pengerjaan.
7. Teman-teman dari Lab Alpro dan GIGa yang selalu mendukung dan menemani penulis selama pengerjaan Tugas Akhir, khususnya Andhika Ditya Bagaskara, Elshe Erviana Angely, Tegar Ganang Satrio Priambodo, Ferry Nur Alfian Eka Putra, dan Hemakesha Ramadhani, Ridzki Raihan Alfaza.

Kami menyadari bahwa Tugas Akhir ini mungkin masih memiliki ruang untuk peningkatan. Oleh karena itu, kami sangat menghargai kritik, saran, dan masukan yang konstruktif untuk perbaikan di masa depan. Kami berharap Tugas Akhir ini dapat memberikan kontribusi signifikan bagi perkembangan ilmu pengetahuan dan teknologi, khususnya dalam bidang Teknik Informatika.

Surabaya, 25 Juli 2024

Hilmi Zharfan Rachmadi

*Halaman ini Sengaja Dikosongkan*

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN .....	v
PERNYATAAN ORISINALITAS .....	ix
Abstrak .....	xiii
KATA PENGANTAR .....	xvii
DAFTAR ISI.....	xix
DAFTAR GAMBAR .....	xxi
DAFTAR TABEL.....	xxiii
DAFTAR KODE SEMU .....	xxv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
BAB 2 TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terkait.....	5
2.2 Dasar Teori .....	6
2.2.1 Deteksi Data Pencilan dengan IQR dan Imputasi dengan Median.....	6
2.2.2 Normalisasi Distribusi Data dengan <i>Power Transform</i> .....	7
2.2.3 Transformasi Data Stasioner dengan <i>Differencing</i> .....	8
2.2.4 Penerapan <i>Scaling</i> untuk Menyesuaikan Skala Data.....	10
2.2.5 Pemodelan Prediksi Deret Waktu dengan AutoARIMA dan XGBoost .....	11
2.2.6 <i>Hyperparameter Tuning</i> dengan Optuna.....	14
2.2.7 Pengujian Evaluasi dengan MAE, RMSE, dan MAPE .....	17
BAB 3 METODOLOGI.....	19
3.1 Metode yang Digunakan.....	19
3.2 Bahan dan Peralatan yang Digunakan .....	21
3.2.1 Spesifikasi Perangkat Keras .....	21
3.2.2 Spesifikasi Perangkat Lunak .....	21
3.3 Implementasi .....	21
3.3.1 Tahap Persiapan dan Pemrosesan Data .....	21
3.3.2 Pemodelan dan Optimasi Parameter.....	24

3.3.3	Evaluasi .....	42
BAB 4	HASIL DAN PEMBAHASAN .....	43
4.1	Uji Coba 1 untuk Membandingkan Performa Model Prediksi Harga DAP .....	43
4.2	Uji Coba 2 untuk Membandingkan Performa Model Prediksi Harga KCL .....	44
4.3	Uji Coba 3 untuk Membandingkan Performa Model Prediksi Harga RP .....	46
4.4	Uji Coba 4 untuk Membandingkan Performa Model Prediksi Harga TSP .....	47
4.5	Uji Coba 5 untuk Membandingkan Performa Model Prediksi Harga Urea .....	49
4.6	Pembahasan .....	50
BAB 5	KESIMPULAN DAN SARAN .....	55
5.1	Kesimpulan.....	55
5.2	Saran .....	55
DAFTAR PUSTAKA	.....	57
BIODATA PENULIS	.....	61

## DAFTAR GAMBAR

Gambar 2.2 Diagram <i>First Order Differencing</i> (Peixeiro, 2022).....	9
Gambar 3.1 Diagram Alir Uji Coba dengan AutoARIMA .....	19
Gambar 3.2 Diagram Alir Uji Coba dengan <i>Machine Learning</i> .....	20
Gambar 3.3 Plot Harga Kelima Jenis Pupuk .....	22
Gambar 3.4 Line Plot Pemisahan Data Latih dan Uji untuk Prediksi 3 Bulan.....	23
Gambar 3.5 Line Plot Pemisahan Data Latih dan Uji untuk Prediksi 6 Bulan.....	23
Gambar 3.6 Line Plot Pemisahan Data Latih dan Uji untuk Prediksi 12 Bulan.....	24
Gambar 3.7 Bar Plot Jumlah Data Pencilan pada Kolom Harga Pupuk.....	25
Gambar 3.8 Box Plot Data Pencilan pada Kolom Harga Pupuk .....	26
Gambar 3.9 Line Plot Visualisasi Data Pencilan pada Kolom Harga Pupuk .....	27
Gambar 3.10 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 3 Bulan).....	28
Gambar 3.11 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 6 Bulan).....	29
Gambar 3.12 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 12 Bulan).....	29
Gambar 3.13 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 3 Bulan) .....	30
Gambar 3.14 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 6 Bulan) .....	31
Gambar 3.15 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 12 Bulan) .....	32
Gambar 3.16 Histogram Distribusi Data Harga DAP .....	34
Gambar 3.17 Histogram Harga DAP Sesudah <i>Power Transform</i> .....	34
Gambar 3.18 Histogram Distribusi Data Harga KCL .....	34
Gambar 3.19 Histogram Harga KCL Sebelum dan Sesudah <i>Power Transform</i> .....	34
Gambar 3.20 Histogram Distribusi Data Harga RP .....	35
Gambar 3.21 Histogram Harga RP Sebelum dan Sesudah <i>Power Transform</i> .....	35
Gambar 3.22 Histogram Distribusi Data Harga TSP .....	35
Gambar 3.23 Histogram Harga TSP Sebelum dan Sesudah <i>Power Transform</i> .....	36
Gambar 3.24 Histogram Harga Urea Sebelum dan Sesudah <i>Power Transform</i> .....	36
Gambar 3.25 Histogram Harga Urea Sebelum dan Sesudah <i>Power Transform</i> .....	36
Gambar 3.26 Contoh Visualisasi Tingkat Kepentingan Parameter untuk XGBoost.....	41
Gambar 3.27 Contoh Visualisasi Riwayat Optimasi Parameter untuk XGBoost.....	42

*(Halaman ini Sengaja Dikosongkan)*



## DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu tentang Prediksi Pupuk dan Perbandingan Model Prediksi Deret Waktu.....	5
Tabel 2.2 Nilai Lambda dan Jenis Transformasi Box-Cox .....	8
Tabel 2.3 Parameter SARIMA .....	12
Tabel 3.1 Skenario Uji Coba .....	20
Tabel 3.2 Spesifikasi Perangkat Keras .....	21
Tabel 3.3 Spesifikasi Perangkat Lunak .....	21
Tabel 3.4 Penjelasan Kolom-kolom Data.....	22
Tabel 3.5 Perbandingan Jumlah Data Pencilan Tiap Jenis Pupuk.....	25
Tabel 3.6 Perbandingan Jumlah Data Pencilan Tiap Jenis Pupuk setelah Imputasi.....	27
Tabel 3.7 Parameter-parameter yang Dipertimbangkan untuk Metode <i>Machine Learning</i> .....	37
Tabel 3.8 Parameter-parameter yang Dipertimbangkan untuk Metode Statistik .....	39
Tabel 4.1 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar DAP .....	43
Tabel 4.2 Analisis Persentase Kepentingan Parameter untuk Prediksi Harga Pasar DAP .....	44
Tabel 4.3 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar KCL .....	44
Tabel 4. 4 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar KCL.....	45
Tabel 4.5 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar RP.....	46
Tabel 4.6 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar RP.....	47
Tabel 4.7 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar TSP.....	47
Tabel 4.8 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar TSP.....	48
Tabel 4.9 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar Urea.....	49
Tabel 4.10 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar Urea .....	50
Tabel 4.11 Rentang Kriteria MAPE .....	52
Tabel 4.12 Rangkuman Jenis Pupuk dengan Metrik Kesalahan Terendah .....	53
Tabel 4.13 Rangkuman Jenis Pupuk dengan Metrik Kesalahan Tertinggi.....	53
Tabel 4.14 Rangkuman Analisis Kepentingan Parameter .....	53

*(Halaman ini Sengaja Dikosongkan)*

## DAFTAR KODE SEMU

Kode Semu 2.1 Contoh Fungsi Objective Optuna .....	15
Kode Semu 2.2 Contoh Fungsi Objective Optuna dengan <i>GridSampler</i> .....	16
Kode Semu 2.3 Contoh Fungsi Objective Optuna dengan <i>SuccessiveHalvingPruner</i> .....	17
Kode Semu 3.1 Pemuatan dan Persiapan Data.....	22
Kode Semu 3.2 Fungsi Deteksi dan Imputasi Data Pencilan .....	25
Kode Semu 3.3 Fungsi Penerapan <i>Power Transform</i> .....	33
Kode Semu 3.4 Fungsi Inversi <i>Power Transform</i> .....	33
Kode Semu 3.5 Fungsi Objective Optimasi Parameter untuk Metode <i>Machine Learning</i> .....	38
Kode Semu 3.6 Fungsi Objective Optimasi Parameter untuk Metode Statistik.....	39
Kode Semu 3.7 Fungsi Manajemen Proses Optimasi Parameter .....	41
Kode Semu 3.8 Analisis Hasil Optimasi Parameter .....	41
Kode Semu 3.9 Fungsi Evaluasi Hasil Prediksi .....	42

*(Halaman ini Sengaja Dikosongkan)*

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Selama lebih dari satu dekade terakhir, industri pupuk global telah menghadapi volatilitas harga yang signifikan (Ilinova et al., 2021). Harga pupuk dipengaruhi oleh berbagai faktor eksternal yang kompleks, termasuk fluktuasi harga bahan baku, dinamika geopolitik, perubahan kebijakan ekspor, serta gangguan rantai pasokan. Beberapa contoh peristiwa yang terjadi, antara lain, perang antara Rusia dan Ukraina, sanksi internasional terhadap Belarus dan Rusia, pembatasan ekspor oleh Rusia dan Cina, kebijakan subsidi pupuk di India, serta gangguan rantai pasokan akibat pandemi COVID-19 (Baffes & Wee, 2022; Jones & Nti, 2022). Faktor-faktor eksternal ini bersifat rumit dan saling terkait, sehingga sulit diprediksi.

Rusia adalah eksportir pupuk terbesar di dunia. Ada lima jenis pupuk utama yang diekspor oleh Rusia: *potassium chloride* (KCL), *rock phosphate* (RP), *diammonium phosphate* (DAP), urea, dan *triple superphosphate* (TSP). DAP adalah pupuk komersial yang umumnya diproduksi dalam jumlah besar melalui reaksi antara amonia gas dan asam fosfat hijau (Gargouri et al., 2011). KCL dikenal sebagai pupuk komersial dengan indeks garam yang tinggi (Hussain et al., 2015). RP adalah sumber alami fosfor yang dapat berfungsi sebagai alternatif untuk pupuk kimia (Maharana et al., 2021). RP juga berperan sebagai bahan baku dasar yang diperlukan dalam produksi pupuk berbasis fosfat, seperti DAP dan TSP (Kumari & Phogat, 2008). TSP adalah pupuk kimia yang diperoleh dengan menambahkan asam fosfat ke RP (Seesanong et al., 2021). Urea adalah pupuk yang diproduksi melalui reaksi karbon dioksida dengan amonia (Ding et al., 2023). Akibat konflik geopolitik yang melibatkan Rusia, pasokan kelima jenis pupuk tersebut terhambat, sehingga meningkatkan volatilitas harga (Kee et al., 2023). Volatilitas harga tiap jenis pupuk terlihat dari perubahan harga yang tajam dalam waktu singkat (Ilinova et al., 2021).

Dalam kondisi volatil ini, PT X, salah satu produsen pupuk urea dan NPK terbesar di Asia Tenggara, menghadapi tantangan memprediksi harga pasar pupuk KCL, RP, DAP, urea, dan TSP dengan tepat. Harga pasar kelima jenis pupuk tersebut menjadi bahan pertimbangan PT X dalam menetapkan harga jual yang tetap setiap bulan. Jika harga jual lebih rendah dari harga pasar, PT X tidak dapat memaksimalkan keuntungan. Sebaliknya, jika harga jual lebih tinggi dari harga pasar, PT X akan kalah bersaing dengan kompetitor. Secara umum, pentingnya prediksi harga pupuk tidak hanya terbatas pada kebutuhan PT X, tetapi juga berpotensi memberi manfaat bagi industri pupuk secara keseluruhan. S. W. Kim & Brorsen (2017) menyampaikan bahwa prediksi harga pupuk adalah alat penting untuk mengurangi risiko di tengah volatilitas industri yang meningkat. Ibendahl (2020) dan Newton (2019) menambahkan bahwa prediksi harga dapat membantu mengendalikan biaya produksi dengan memberikan informasi tentang waktu yang tepat untuk membeli bahan baku. Selain itu, Borkar (2023) menyoroti bahwa hasil prediksi dapat membantu merencanakan alokasi sumber daya, menghindari pemborosan, dan meningkatkan kualitas operasional.

Kebermanfaatan prediksi terkait pupuk bergantung pada tingkat akurasi. Prediksi yang tidak akurat dapat membawa kerugian bagi industri pupuk. Bahkan, ketidakakuratan prediksi menjadi salah satu penyebab tingginya volatilitas yang dihadapi produsen pupuk. Prediksi yang tidak akurat dapat bersifat terlalu pesimis atau terlalu optimis. Prediksi yang terlalu pesimis dapat menghambat investasi dalam produksi pupuk, sehingga menyebabkan kekurangan pasokan untuk kebutuhan pertanian. Sebaliknya, prediksi yang terlalu optimis dapat mendorong investasi berlebihan dalam kapasitas produksi pupuk. Sebagai contoh, pada tahun 1980-an, prediksi tingginya permintaan pupuk menyebabkan kelebihan pasokan dan ketidakstabilan pasar. Selain itu, hasil prediksi yang tidak akurat dapat mengakibatkan

keputusan kebijakan dan strategi perusahaan yang tidak tepat (International Fertilizer Industry Association (IFA), 2007).

Literatur mengenai prediksi harga pupuk menunjukkan kecenderungan yang kuat terhadap penggunaan metode-metode statistik. S. W. Kim & Brorsen (2017) melakukan prediksi harga butiran urea di pasar New Orleans dengan menggunakan variasi-variasi dari model *Autoregressive Moving Average* (ARMA) dan *Autoregressive* (AR). Variasi-variasi tersebut meliputi *Autoregressive* dengan variabel eksogen (ARX), *Seasonal Autoregressive* (SAR), *Seasonal Autoregressive* dengan variabel eksogen (SARX), *Autoregressive* dengan GARCH (ARG), serta *Autoregressive* dengan variabel eksogen dan GARCH (ARXG). Sementara itu, Newton (2019) menggunakan model *Vector Autoregression* (VAR) untuk memprediksi harga pupuk grosir dan eceran di wilayah Corn Belt dan Southern Plains Amerika Serikat. Meskipun tidak secara langsung memprediksi harga, Borkar (2023) menggunakan model *Autoregressive Integrated Moving Average* (ARIMA) dan pendekatan Box-Jenkins untuk memprediksi konsumsi pupuk nitrogen, fosfor, dan kalium di India, yang dapat memberikan wawasan tentang tren permintaan dan potensi pergerakan harga.

Dalam perkembangan metode prediksi deret waktu, metode machine learning, khususnya XGBoost, telah menunjukkan potensi yang menjanjikan sebagai alternatif metode statistik. Henzel et al. (2021) menemukan bahwa XGBoost memberikan hasil terbaik dalam memprediksi penjualan bensin dan solar di 25 SPBU dibandingkan dengan metode ARIMA, *Moving Average* (MA), dan *K-Nearest Neighbor* (KNN). Senada dengan penelitian tersebut, Durgapal & Vimal (2021) melaporkan bahwa XGBoost menghasilkan performa terbaik berdasarkan RMSE dan MAPE dalam memprediksi harga saham Google, mengungguli model ARIMA dan *Random Forest*. Keunggulan XGBoost juga terlihat dalam studi Rahman & Chowdhury (2022) yang membandingkan akurasi prediksi model *Autoregressive Integrated Moving Average* dengan variabel eksogen (ARIMAX) dan XGBoost untuk memodelkan insiden COVID-19 di negara-negara *South Asian Association for Regional Cooperation* (SAARC), di mana XGBoost menunjukkan performa yang lebih baik di sebagian besar negara. Lebih lanjut, Zhang et al. (2021) membandingkan performa XGBoost dengan beberapa model lain termasuk ARIMA, LSTM, Prophet, dan *Gradient-Boosted Decision Trees* (GBDT) dalam memprediksi volume penjualan toko ritel, dan menemukan bahwa XGBoost memberikan hasil terbaik dengan RMSE dan MAE terendah, serta komputasi yang lebih cepat. Alim et al. (2020) juga melaporkan performa XGBoost yang lebih baik dibandingkan ARIMA dalam memprediksi tren dan musiman kejadian bruselosis manusia di daratan China, dengan nilai MAE, RMSE, dan MAPE yang lebih rendah pada data pelatihan dan pengujian.

Penggunaan XGBoost dan metode machine learning lainnya dalam memprediksi harga pupuk masih terbatas, meskipun telah menunjukkan potensi yang menjanjikan dalam berbagai aplikasi prediksi deret waktu. Di antara sedikit penelitian yang menggunakan metode machine learning untuk prediksi harga pupuk, Ibendahl (2020) mengembangkan model prediksi harga pupuk anhydrous ammonia menggunakan regresi *Ordinary Least Squares* (OLS) dengan memanfaatkan harga jagung dan harga minyak mentah dengan variabel *lag* 9 bulan. Sementara itu, K. Jana et al. (2023) menggunakan pendekatan yang lebih kompleks dengan mengintegrasikan algoritma Boruta, *Maximal Overlap Discrete Wavelet Transformation* (MODWT), dan regresi *Random Forest* (RF) untuk memprediksi pergerakan harga pupuk urea dan DAP. Mengingat keunggulan XGBoost dalam berbagai aplikasi prediksi deret waktu, terdapat peluang untuk mengeksplorasi penggunaan XGBoost dalam prediksi harga pupuk di masa depan.

Akurasi prediksi deret waktu tidak hanya dipengaruhi oleh jenis model yang digunakan, tetapi juga oleh kualitas data deret waktu dan jenis transformasi yang diterapkan. Salah satu aspek dari kualitas data adalah penanganan outlier atau data pencilan. Romanuke (2022)

menemukan bahwa penghapusan data pencilan di titik awal dapat meningkatkan akurasi prediksi, terutama pada deret waktu tanpa tren atau musiman. Namun, penghapusan data pencilan pada deret waktu dengan kenaikan eksponensial berdampak pada pengurangan akurasi prediksi. Penelitian Parra-Plazas et al. (2023) menghapus data pencilan yang terdeteksi dan mengisi data yang hilang menggunakan metode *Discrete Fourier Transform* (DFT) dan *Inverse Fast Fourier Transform* (IFFT). Proses penghapusan data pencilan tersebut membuat deret waktu menjadi lebih stasioner berdasarkan tes *Augmented Dickey-Fuller* (ADF). Stasioneritas merujuk pada kondisi di mana rata-rata, varian, dan kovarian deret waktu konstan seiring waktu. Van Greunen et al. (2014) dan Dixit & Jain (2021) menunjukkan bahwa deret waktu yang stasioner cenderung menghasilkan kesalahan prediksi yang lebih rendah. Untuk mencapai stasioneritas, berbagai teknik transformasi data lain dapat diterapkan, termasuk *power transform*, penghapusan tren, dan *differencing*.

Berdasarkan literatur-literatur yang telah disebutkan di atas, penelitian ini akan mengimplementasikan model prediksi harga pasar pupuk DAP, KCL, RP, TSP, dan urea. Akan digunakan metode statistik dan metode *machine learning*. Metode statistik diwakili oleh model AutoARIMA, sedangkan metode *machine learning* diwakili oleh model XGBoost. Kedua model akan digunakan untuk memprediksi harga kelima jenis pupuk selama 3, 6, dan 12 bulan mendatang. Kemudian, hasil prediksi model akan dievaluasi dengan metrik kesalahan MAE, RMSE, MAPE. Untuk membantu meminimalkan metrik-metrik kesalahan, akan dilakukan eksplorasi terkait pemrosesan dan transformasi data yang meliputi penanganan data pencilan dengan imputasi median, *power transform*, *differencing*, *scaling*, dan pembuatan variabel *lag*.

## 1.2 Rumusan Permasalahan

Permasalahan pada tugas akhir ini memiliki rumusan sebagai berikut:

1. Bagaimana pemrosesan data yang diperlukan untuk data harga pasar pupuk?
2. Bagaimana cara mengembangkan model prediksi harga pasar pupuk?
3. Bagaimana cara mengevaluasi kinerja model prediksi harga pasar pupuk?

## 1.3 Batasan Masalah

Batasan-batasan yang ditentukan dalam pengerjaan tugas akhir ini sebagai berikut.

1. Dataset yang digunakan untuk melatih model adalah data historis harga pasar pupuk DAP, KCL, RP, TSP, dan urea dari November 1993 hingga November 2023. Data ini diambil dari situs portal data Index Mundi.
2. Parameter yang digunakan dalam proses tuning berfokus pada jenis transformasi dan pemrosesan yang diterapkan pada data.
3. Bahasa pemrograman Python akan digunakan dalam pemrosesan data dan pengembangan model prediksi harga pasar untuk kelima jenis pupuk tersebut.
4. Model prediksi harga pasar pupuk akan diimplementasikan menggunakan AutoARIMA dan XGBoost. Kedua model hanya memprediksi berdasarkan data historis harga pasar pupuk tanpa mempertimbangkan faktor eksternal dan harga pokok penjualan (HPP).

## 1.4 Tujuan

Tujuan dari tugas akhir ini adalah mengembangkan model prediksi menggunakan AutoARIMA dan XGBoost yang memanfaatkan data historis untuk memprediksi harga pasar pupuk DAP, KCL, RP, TSP, dan urea.

### **1.5 Manfaat**

Hasil dari pengerjaan tugas akhir ini diharapkan dapat membantu memberikan bahan pertimbangan bagi PT X dalam menetapkan harga jual produk perusahaan supaya dapat memaksimalkan keuntungan.



## BAB 2 TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Tugas akhir ini menggunakan berbagai referensi dari penelitian sebelumnya yang membahas prediksi pupuk serta membandingkan model statistik dan machine learning untuk prediksi deret waktu. Dalam rangka memahami penelitian terkait prediksi deret waktu harga pupuk, dilakukan pencarian dengan kata kunci *forecasting fertilizer price*. Selain itu, untuk mengetahui penelitian terkait prediksi deret waktu dengan XGBoost, dilakukan pencarian dengan kata kunci XGBoost *time series forecasting*. Tabel 2.1 merangkum penelitian-penelitian tersebut dengan rincian tujuan, metodologi, hasil, nama peneliti, dan tahun penelitian. Dari tabel tersebut, terlihat bahwa prediksi harga pupuk kebanyakan masih menggunakan metode statistik, sedangkan metode machine learning belum banyak digunakan. Hal ini menunjukkan peluang untuk mengembangkan model prediksi harga pupuk dengan metode machine learning, seperti XGBoost.

Tabel 2.1 Penelitian Terdahulu tentang Prediksi Pupuk dan Perbandingan Model Prediksi Deret Waktu

No.	Tujuan	Metodologi	Hasil
1	Mengembangkan model prediksi harga urea granular di pasar New Orleans untuk mengurangi risiko dalam industri pupuk	Mengimplementasikan model ARMA, AR, ARX, SAR, SARX, ARG, dan ARXG. Teknik <i>rolling window regression</i> juga digunakan dengan ukuran <i>window</i> 36, 48, 60, 72, dan 81 bulan. Metrik evaluasi yang digunakan meliputi MAE, RMSE, dan tes MDM.	Model ARX dengan <i>window</i> 48 bulan menunjukkan kinerja terbaik. Tidak ada perbedaan signifikan antara hasil prediksi model ARX dan hasil prediksi komersial. Kombinasi hasil prediksi ARX dan komersial menghasilkan metrik kesalahan terendah (S. W. Kim & Brorsen, 2017).
2	Mengembangkan kerangka kerja machine learning ensemble untuk memprediksi pergerakan harga pupuk DAP dan urea	Mengintegrasikan algoritma Boruta, Transformasi Wavelet Diskrit Overlap Maksimal (MODWT), <i>Random Forest</i> (RF), dan AI yang dapat dijelaskan untuk mengatasi masalah keterbatasan jumlah observasi dan perilaku nonlinear pada data harga pupuk. Kerangka kerja ini digunakan untuk memprediksi harga bulanan urea dan DAP selama hampir 30 tahun terakhir. Metrik yang digunakan antara lain MAE, RMSE, MAPE.	Kinerja prediksi lebih baik ketika residual digunakan sebagai fitur tambahan dalam model. Untuk prediksi harga urea, ketika residual dimasukkan sebagai fitur, model menghasilkan MAE 12.070, RMSE 25.916, MAPE 5.250. Sementara untuk harga DAP, hasilnya adalah MAE 11.332, RMSE 26.615, MAPE 3.192 (K. Jana et al., 2023).
3	Meneliti dampak pengelompokan data dalam proses prediksi penjualan bahan bakar di SPBU.	Menggunakan metode XGBoost, ARIMA, MA, dan K-NN untuk memprediksi penjualan bensin dan solar di 25 SPBU selama periode 4 jam. Metode clustering seperti <i>partitional clustering</i> dan <i>k-means</i> juga digunakan untuk mengelompokkan data penjualan yang serupa.	Metode terbaik adalah XGBoost, yang menggunakan data dari seluruh SPBU yang telah dinormalisasi dan digabungkan. Metode ini menghasilkan kesalahan terendah dengan RMSE 347.97, MAE 207.13, $R^2$ 0.65, dan WMAPE 0.32% (Henzel et al., 2021).

No.	Tujuan	Metodologi	Hasil
		Metrik evaluasi yang digunakan mencakup RMSE, MAE, R <sup>2</sup> , dan WMAPE.	
4	Membandingkan akurasi prediksi model ARIMAX dan XGBoost untuk memodelkan insiden COVID-19 di negara-negara SAARC.	Metode yang digunakan adalah model ARIMAX dan XGBoost. Data yang digunakan meliputi jumlah kasus COVID-19 harian, suhu minimum dan maksimum, kelembaban relatif, tekanan permukaan, curah hujan, dan kecepatan angin maksimum dari awal pandemi hingga 29 Januari 2022 untuk setiap negara SAARC. Metrik yang digunakan untuk mengevaluasi kinerja model meliputi MAPE, MPE, MAE, dan RMSE.	Model XGBoost memiliki performa yang lebih baik dalam memprediksi kasus COVID-19 di sebagian besar negara SAARC dibandingkan model ARIMAX (Rahman & Chowdhury, 2022).
5	Mengembangkan model prediksi kasus brucellosis pada manusia di Tiongkok.	Membandingkan kinerja model ARIMA dan XGBoost untuk memprediksi tren dan musiman kejadian brucellosis. Data yang digunakan mencakup kasus bulanan brucellosis dari Januari 2008 hingga Juni 2019. Metrik evaluasi yang digunakan adalah MAE, RMSE, dan MAPE.	Model XGBoost menunjukkan performa lebih baik dibandingkan ARIMA, dengan nilai MAE, RMSE, dan MAPE yang lebih rendah pada data pelatihan dan pengujian. XGBoost dianggap lebih cocok untuk memprediksi kasus brucellosis karena kemampuannya menangani data nonlinear (Alim et al., 2020).
6	Mengembangkan model prediksi beras tahunan di Bangladesh.	Membandingkan performa model ARIMA dan XGBoost untuk memprediksi produksi beras dalam 10 tahun ke depan. Penelitian ini menggunakan data produksi beras tahunan Bangladesh dari tahun 1961 hingga 2020. Metrik evaluasi yang digunakan meliputi MAE, MPE, RMSE, dan MAPE.	Model XGBoost memiliki performa yang lebih baik dibandingkan ARIMA, dengan nilai MAPE yang lebih rendah pada data pengujian (Noorunnahar et al., 2023).

## 2.2 Dasar Teori

### 2.2.1 Deteksi Data Pencilan dengan IQR dan Imputasi dengan Median

Data pencilan atau *outlier* adalah yang berbeda secara signifikan dari mayoritas data. Data ini dapat merepresentasikan fluktuasi karena peristiwa yang tidak biasa atau tidak terduga. Salah satu metode untuk mendeteksi data pencilan adalah dengan rentang interkuartil (IQR). IQR adalah ukuran statistik untuk sebaran data yang dihitung berdasarkan jarak antara kuartil bawah (Q1) dan kuartil atas (Q3).

$$IQR = Q3 - Q1$$

Di mana

- $IQR$  : Rentang interkuartil data
- $Q3$  : Kuartil atas
- $Q1$  : Kuartil bawah

Data yang dianggap sebagai pencilan adalah data yang berada di luar batas bawah dan batas atas. Batas bawah didefinisikan sebagai  $Q1 - 1.5 \times IQR$ , sedangkan batas atas didefinisikan sebagai  $Q3 + 1.5 \times IQR$ . Data pencilan yang terdeteksi akan diganti dengan nilai median (Yaro et al., 2023; Zhang et al., 2019). Berdasarkan penelitian Lee & Yun (2024), menggantikan data pencilan dengan nilai median menunjukkan hasil yang cukup baik. Tingkat kesalahan rata-rata relatifnya cukup rendah, yaitu 16,91% dalam prediksi model machine learning. Metode penggantian dengan median ini juga efektif dalam mempertahankan relasi antara faktor-faktor yang berperan dalam data.

### 2.2.2 Normalisasi Distribusi Data dengan *Power Transform*

*Power transform* adalah transformasi berbasis pangkat yang digunakan untuk menstabilkan variansi dan membuat distribusi data lebih normal. Tiga jenis *power transform* yang sering digunakan adalah transformasi logaritma, akar kuadrat, dan Box-Cox (Brownlee, 2020; Guerrero & Perera, 2004).

#### a. Transformasi Logaritma

Transformasi logaritma berguna untuk mengatasi tren eksponensial yang ekstrem dalam deret waktu. Transformasi ini dapat membuat distribusi data lebih normal. Namun, perlu diperhatikan bahwa transformasi logaritma hanya dapat diterapkan jika semua data lebih dari nol (Brownlee, 2020).

$$y = \log_b(x) \quad (2.1)$$

$$x = b^y \quad (2.2)$$

Dimana

- $y$  : Nilai data hasil transformasi logaritma
- $x$  : Nilai data asli
- $b$  : Basis logaritma; biasanya bernilai 10

#### b. Transformasi Akar Kuadrat

Transformasi akar kuadrat berguna untuk mengatasi tren pertumbuhan kuadratik dalam deret waktu. Seperti transformasi logaritma, transformasi akar kuadrat juga dapat membuat distribusi data lebih normal (Brownlee, 2020).

$$y = \sqrt{x} \quad (2.3)$$

$$x = y^2 \quad (2.4)$$

Dimana

- $y$  : Nilai data hasil transformasi akar kuadrat
- $x$  : Nilai data asli

**c. Transformasi Box-Cox**

Transformasi Box-Cox mencakup dua jenis transformasi sebelumnya, yaitu logaritma dan akar kuadrat. Keunggulan utama Box-Cox adalah mampu memilih *power transform* yang paling sesuai dengan data berdasarkan parameter  $\lambda$ . Box-Cox mencari nilai  $\lambda$  yang menghasilkan distribusi data paling normal. Dalam Rumus (2.5) dan (2.6), terlihat bahwa Box-Cox memiliki dua persamaan yang bergantung pada nilai  $\lambda$  (Brownlee, 2020; Proietti & Lütkepohl, 2013).

$$y(\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \text{jika } \lambda \neq 0 \\ \ln(x), & \text{jika } \lambda = 0 \end{cases} \tag{2.5}$$

$$x = \begin{cases} (1 + \lambda y)^{1/\lambda}, & \text{jika } \lambda \neq 0 \\ e^y, & \text{jika } \lambda = 0 \end{cases} \tag{2.6}$$

Dimana

- $y$  : Nilai data hasil transformasi akar kuadrat
- $x$  : Nilai data asli
- $\lambda$  : Parameter Box-Cox

Jenis-jenis *power transform* berdasarkan parameter  $\lambda$  Box-Cox ditampilkan pada Tabel 2.1 (Brownlee, 2020).

Tabel 2.2 Nilai Lambda dan Jenis Transformasi Box-Cox

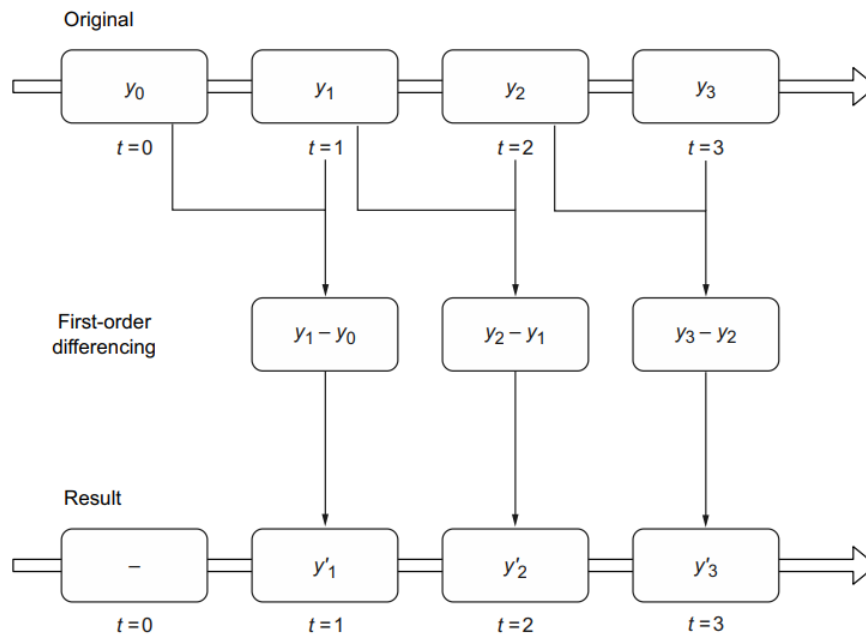
Nilai Lambda	Jenis Transformasi	Persamaan
-1.0	<i>Reciprocal</i>	$y = \frac{1}{x}$
-0.5	<i>Reciprocal Square Root</i>	$y = \frac{1}{\sqrt{x}}$
0.0	<i>Log</i>	$y = \log_b(x)$
0.5	<i>Square Root</i>	$y = \sqrt{x}$
1.0	<i>None</i>	$y = x$

**2.2.3 Transformasi Data Stasioner dengan Differencing**

Sebelum melatih model prediksi dengan data deret waktu, data tersebut sebaiknya dibuat stasioner. Salah satu metode untuk membuat data deret waktu menjadi stasioner adalah differencing (Fufa, 2021). *Differencing* adalah operasi menghitung selisih antara nilai saat ini dengan nilai-nilai sebelumnya. (Hyndman & Athanasopoulos, 2021; National Institute of Standards and Technology, 2012). Di antara jenis *differencing* yang sering digunakan adalah *first order differencing*, dan *second order differencing* (Hyndman & Athanasopoulos, 2021).

**a. First Order Differencing**

*First order differencing* menghitung selisih antara dua observasi yang berurutan. Setelah *first order differencing*, jumlah nilai pada deret waktu akan berkurang satu karena tidak mungkin menghitung selisih untuk nilai pertama (Hyndman & Athanasopoulos, 2021).



Gambar 2.1 Diagram First Order *Differencing* (Peixeiro, 2022)

$$y'_t = y_t - y_{t-1} \quad (2.7)$$

Dimana

- $y'_t$  : Nilai deret waktu hasil *first order differencing* pada waktu  $t$
- $y_t$  : Nilai deret waktu pada waktu  $t$
- $y_{t-1}$  : Nilai deret waktu pada satu observasi sebelumnya ( $t - 1$ )

**b. *Second Order Differencing***

Terkadang, deret waktu yang sudah melalui *first order differencing* masih belum stasioner, sehingga perlu dilakukan *differencing* lagi. *Second order differencing* pada dasarnya melakukan *first order differencing* dua kali. Setelah *second order differencing*, jumlah nilai pada deret waktu akan berkurang dua. Dalam praktiknya, *differencing* di atas orde dua hampir tidak pernah diperlukan (Hyndman & Athanasopoulos, 2021).

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned} \quad (2.8)$$

Dimana

- $y''_t$  : Nilai deret waktu hasil *second order differencing* pada waktu  $t$
- $y'_t, y'_{t-1}$  : Nilai deret waktu hasil *first order differencing* pada waktu  $t$  dan  $t - 1$
- $y_t, y_{t-1}, y_{t-2}$  : Nilai deret waktu pada waktu  $t, t - 1$ , dan  $t - 2$

**c. *Seasonal Differencing***

*Seasonal differencing* menghitung selisih antara suatu nilai dengan nilai observasi sebelumnya dalam musim yang sama. Sebagai contoh, jika  $m$  adalah jumlah periode waktu

dalam satu musim dan data mempunyai frekuensi bulanan, maka *Seasonal differencing* akan menghitung selisih antara nilai bulan ini dengan nilai  $m$  bulan sebelumnya. Data bulanan biasanya menggunakan  $m = 12$  yang mewakili 12 bulan atau 1 tahun. Terkadang diperlukan *Seasonal differencing* dan *first order differencing* untuk membuat deret waktu stasioner (Hyndman & Athanasopoulos, 2021; Peixeiro, 2022).

$$y'_t = y_t - y_{t-m} \quad (2.9)$$

Dimana

- $y'_t$  : Nilai deret waktu hasil *Seasonal differencing* pada waktu  $t$
- $y_t$  : Nilai deret waktu pada waktu  $t$
- $m$  : Jumlah periode untuk satu musim
- $y_{t-m}$  : Nilai deret waktu pada satu musim sebelumnya ( $t - m$ )

#### 2.2.4 Penerapan Scaling untuk Menyesuaikan Skala Data

*Scaling* adalah proses transformasi data supaya mempunyai rentang atau distribusi tertentu. Misalnya, rentang  $[0,1]$  atau distribusi normal. *Scaling* dapat membantu meningkatkan performa algoritma *machine learning*, terutama jika fitur-fiturnya mempunyai rentang atau skala yang berbeda. *Scaling* juga dapat membantu mengurangi dampak data pencilan (*outlier*), *noise*, dan kecondongan (*skewness*) dalam data (Raju et al., 2020). Tiga teknik *scaling* yang digunakan dalam tugas akhir ini adalah *MinMaxScaler*, *StandardScaler*, dan *RobustScaler*.

##### a. *MinMaxScaler*

*MinMaxScaler* mengubah rentang data menjadi  $[0,1]$ . Kelemahannya, teknik *scaling* ini sensitif terhadap *outlier* karena tidak merubah distribusi data asli (De Amorim et al., 2022; Raju et al., 2020).

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2.10)$$

Dimana

- $x'_i$  : Nilai data yang sudah di-*scaling*
- $x_{min}$  : Nilai minimum dalam data
- $x_{max}$  : Nilai maksimum dalam data

##### b. *StandardScaler*

*StandardScaler* mengimplementasikan normalisasi *Z-score* untuk memusatkan distribusi data di sekitar 0 dengan standar deviasi 1. *StandardScaler* mengurangi data dengan rata-rata dan membaginya dengan standar deviasi. *StandardScaler* mampu menormalisasi nilai positif dan negatif menjadi distribusi yang serupa. Akan tetapi, teknik ini berasumsi bahwa data mempunyai distribusi normal (De Amorim et al., 2022; Raju et al., 2020). Kelemahannya, jika terdapat *outlier* dalam data, rentang data hasil *StandardScaler* akan menjadi terlalu kecil (De Amorim et al., 2022).

$$x'_i = \frac{x_i - \bar{x}}{s} \quad (2.11)$$

Dimana

- $x'_i$  : Nilai data yang sudah di-*scaling*
- $\bar{x}$  : Rata-rata data
- $s$  : Standar deviasi data

**c. RobustScaler**

*RobustScaler* mengurangi data dengan median dan membaginya dengan rentang interkuartil (IQR). Teknik *scaling* ini dapat mengurangi pengaruh data pencilan (De Amorim et al., 2022; Raju et al., 2020).

$$x'_i = \frac{x_i - Q2(x)}{Q3(x) - Q1(x)} \quad (2.12)$$

Dimana

- $x'_i$  : Nilai data yang sudah di-*scaling*
- $Q2(X)$  : Kuartil kedua atau median data
- $Q1(x), Q3(X)$  : Kuartil pertama dan kuartil ketiga data
- $Q3(x) - Q1(x)$  : Rentang interkuartil data (IQR)

### 2.2.5 Pemodelan Prediksi Deret Waktu dengan AutoARIMA dan XGBoost

Dalam melakukan prediksi deret waktu, terdapat dua metode yang digunakan, yaitu metode statistik dan metode machine learning. Metode statistik menggunakan implementasi *Seasonal Autoregressive Integrated Moving Average* (SARIMA) melalui AutoARIMA untuk memodelkan pola musiman dan tren dalam data deret waktu. Sementara itu, metode machine learning menggunakan model *Extreme Gradient Boosting* (XGBoost) dengan bantuan variabel *lag* untuk melakukan prediksi. Kedua pendekatan memiliki karakteristik, kelebihan, dan tantangan masing-masing dalam menangani kompleksitas data deret waktu.

**a. Pemodelan Statistik dengan SARIMA dan AutoARIMA**

*Seasonal Autoregressive Integrated Moving Average* (SARIMA) adalah model prediksi statistik yang mempertimbangkan aspek musiman (*Seasonal*) dan nonmusiman (*non-Seasonal*) dari deret waktu. Rumus (2.13) menampilkan notasi SARIMA.

$$(p, d, q) x (P, D, Q)S \quad (2.13)$$

Parameter  $(p, d, q)$  mewakili aspek *non-Seasonal* dari model.  $p$  mewakili bagian *Autoregressive* (AR), yakni jumlah data masa lalu yang digunakan untuk memprediksi nilai berikutnya.  $d$  mewakili bagian *integrated* (I), yakni jumlah orde *differencing* yang dibutuhkan untuk membuat deret waktu stasioner.  $q$  mewakili bagian *Moving Average* (MA), yakni jumlah kesalahan prediksi masa lalu yang mempengaruhi prediksi nilai berikutnya (Abdulqader et al., 2020; Peixeiro, 2022). Sebagai contoh, jika  $p = 2$ , maka model akan melibatkan nilai  $y_{t-1}$  dan  $y_{t-2}$  dalam proses prediksi. Jika  $q = 2$ , maka model akan melibatkan kesalahan  $\epsilon_{t-1}$  dan  $\epsilon_{t-2}$ . Kemudian, jika  $d = 1$ , maka akan dilakukan *first order differencing* dengan Rumus (2.14).

$$y'_t = y_t - y_{t-1} \quad (2.14)$$

Parameter  $(P, D, Q)S$  mewakili aspek *Seasonal* dari model.  $P, D, Q$  pada dasarnya sama dengan  $p, d, q$ .  $P$  mewakili bagian *Seasonal Autoregressive*,  $D$  mewakili bagian *Seasonal differencing*, dan  $Q$  mewakili bagian *Seasonal Moving Average*. Parameter tambahan  $S$  adalah jumlah periode waktu dalam satu musim (Abdulqader et al., 2020; Peixeiro, 2022). Sebagai contoh, pada umumnya  $S = 12$  untuk data bulanan dengan siklus tahunan. Jika  $S = 12$  dan  $P = 2$ , maka model akan melibatkan nilai  $y_{t-12}$  dan  $y_{t-24}$  dalam proses prediksi. Jika  $Q = 2$ , maka model akan melibatkan kesalahan  $\epsilon_{t-12}$  dan  $\epsilon_{t-24}$ . Kemudian, jika  $D = 1$ , akan dilakukan *Seasonal differencing* dengan Rumus (2.15). Secara keseluruhan, parameter-parameter SARIMA tercantum pada Tabel 2.4 (Abdulqader et al., 2020; Peixeiro, 2022).

$$y'_t = y_t - y_{t-12} \quad (2.15)$$

Tabel 2.3 Parameter SARIMA

Parameter	Keterangan
p	Orde <i>Autoregressive</i> (AR)
d	Orde <i>non-Seasonal differencing</i> atau komponen <i>integrated</i> (I)
q	Orde <i>Moving Average</i> (MA)
P	Orde <i>Seasonal Autoregressive</i> (AR)
D	Orde <i>Seasonal differencing</i>
Q	Orde <i>Seasonal Moving Average</i> (MA)
S	Jumlah periode waktu dalam satu musim

Algoritma SARIMA diimplementasikan dengan model AutoARIMA dari pustaka pmdarima dan Statsforecast. AutoARIMA mengotomasi proses identifikasi parameter model yang optimal termasuk  $(p, d, q)$  dan  $(P, D, Q)$ . Otomasi proses identifikasi parameter model dilakukan melalui pengujian stasioneritas dan minimalisasi Akaike Information Criterion (AIC) (James & Tripathi, 2021). Parameter  $(d)$  dan  $(D)$  ditentukan oleh pengujian stasioneritas, sedangkan parameter  $(p, q)$  dan  $(P, Q)$  ditentukan berdasarkan AIC terendah (Ospina et al., 2023). James & Tripathi (2021) mendefinisikan AIC sebagai ukuran yang digunakan secara luas untuk sebuah model statistik. Pada dasarnya, AIC mengukur kesesuaian dan kesederhanaan model ke dalam sebuah statistik tunggal.

Untuk menentukan jumlah orde  $d$  atau *non seasonal differencing* yang dibutuhkan data, AutoARIMA menggunakan tes *Augmented Dickey-Fuller* (ADF). Hipotesis nol ( $H_0$ ) dalam tes ADF adalah deret waktu mempunyai akar unit nonstasioneritas. Akar unit nonstasioneritas adalah karakteristik yang membuat rata-rata, varians, dan autokorelasi menjadi bergantung pada waktu. Ada dua nilai yang dihasilkan oleh tes ADF, yakni nilai  $p$  dan statistik ADF. Jika nilai  $p \leq 0.05$ , maka hipotesis nol dapat ditolak, yang menandakan bahwa deret waktu stasioner. Sementara itu, nilai statistik ADF berupa bilangan negatif. Semakin negatif nilainya, maka semakin kuat penolakan terhadap hipotesis nol (Peixeiro, 2022).

Sementara itu, AutoARIMA menentukan orde  $D$  atau *seasonal differencing* yang dibutuhkan data melalui tes *Osborn, Chui, Smith, and Birchenhall* (OCSB).  $H_0$  dari tes OCSB adalah keberadaan *seasonal* unit root dalam data, yang mengimplikasikan bahwa deret waktu tidak stasioner pada periode musiman tertentu (Rice et al., 2019). Jumlah bulan dalam satu periode musim ditentukan oleh parameter *season\_length* dalam AutoARIMA. Tes OCSB mengembalikan nilai  $p$  ( $p$ -value) yang digunakan untuk menguji  $H_0$ . Jika nilai  $p \leq 0,05$ , maka hipotesis nol dapat ditolak, yang berarti deret waktu tidak memiliki *Seasonal* unit root. Sebaliknya, jika nilai  $p > 0,05$ , maka hipotesis nol tidak dapat ditolak, yang berarti deret waktu memiliki *Seasonal* unit root (Rice et al., 2019).



James & Tripathi (2021) juga menemukan bahwa dalam beberapa kasus, model AutoARIMA (terutama yang menyertakan faktor musiman) memiliki kinerja yang lebih baik daripada model yang disesuaikan secara manual. Beberapa alasan ini menjadikan AutoARIMA andal dalam memilih model ARIMA yang paling sesuai untuk deret waktu yang diberikan (Ospina et al., 2023). Akan tetapi, seperti model ARIMA pada umumnya, AutoARIMA bergantung pada pemodelan linier, sehingga sulit menangkap pola nonlinier dalam data (Krishnan et al., 2017). Ospina et al. (2023) pun menambahkan bahwa performa AutoARIMA berpotensi menurun seiring meningkatnya periode waktu yang perlu diprediksi.

#### **b. Pemodelan *Machine Learning* dengan Extreme Gradient Boosting (XGBoost)**

*Gradient boosting* adalah algoritma ensemble *machine learning* yang menggabungkan beberapa model pembelajaran lemah untuk membentuk model yang lebih kuat. Model-model pembelajaran lemah ini bisa berbasis linear atau pohon keputusan (*decision tree*). Pelatihan model *gradient boosting* dilakukan secara bertahap. Pada setiap tahap, model pembelajaran baru ditambahkan untuk memperbaiki kesalahan dari model sebelumnya (Paliari et al., 2021).

*Extreme Gradient Boosting* (XGBoost) merupakan algoritma hasil optimasi *gradient boosting*. Beberapa optimasi tersebut, antara lain, penambahan teknik regularisasi, penanganan nilai kosong, pemrosesan paralel, dan perhitungan gradien orde dua dari fungsi kerugian (*loss function*). Regularisasi membantu mencegah *overfitting* dengan memberikan penalti ketika model menjadi terlalu kompleks. Penanganan nilai kosong secara otomatis oleh XGBoost membantu menghemat waktu *preprocessing*. Pemrosesan paralel dan perhitungan gradien orde dua dari fungsi kerugian berguna untuk mempercepat performa XGBoost, sehingga melampaui *gradient boosting* biasa (Paliari et al., 2021; Zhai et al., 2020).

Dalam prediksi deret waktu, model *machine learning*, seperti XGBoost, menggunakan pendekatan *supervised*. Pendekatan supervised terdiri dari variabel independen dan variabel dependen yang menjadi target prediksi. Model supervised *machine learning* akan mempelajari hubungan antara kedua variabel tersebut (Korstanje, 2021). Untuk menerapkan supervised *machine learning*, deret waktu perlu diubah menjadi format matriks. Dalam format matriks, tiap nilai dalam deret waktu akan dihubungkan dengan nilai-nilai sebelumnya yang disebut dengan variabel *lag*. Sebagai contoh, *lag* 1 adalah nilai pada waktu  $t - 1$ , dan *lag*  $m$  adalah nilai pada waktu  $t - m$ . Penggunaan variabel *lag* sebagai input prediksi memungkinkan model *machine learning* untuk mempelajari pola dan ketergantungan antara nilai masa lalu dan masa depan. Oleh karena itu, jumlah variabel *lag* yang digunakan sebagai fitur dalam matriks perlu diatur untuk mengoptimalkan performa model.

Jika deret waktu sudah diubah menjadi matriks, model XGBoost dapat dilatih untuk memprediksi nilainya. Selama proses pelatihan, setiap baris dianggap sebagai potongan data terpisah. Di mana nilai variabel *lag* 1, 2, ...,  $m$  akan digunakan sebagai input untuk memprediksi nilai pada waktu  $m + 1$  (Rodrigo & Ortiz, 2023). Dalam hal ini, nilai variabel *lag* 1, 2, ...,  $m$  akan berperan sebagai variabel independen dan nilai pada waktu  $m + 1$  berperan sebagai variabel dependen. Pada dasarnya, variabel *lag* hanya dapat digunakan untuk memprediksi satu nilai selanjutnya (Korstanje, 2021). Untuk memprediksi beberapa nilai masa depan dengan variabel *lag*, diperlukan pendekatan tambahan, yakni prediksi *multi-step* secara rekursif.

Prediksi *multi-step* secara rekursif melibatkan proses di mana tiap prediksi baru didasarkan pada prediksi sebelumnya. Model menggunakan hasil prediksinya sendiri sebagai input untuk prediksi berikutnya. Sebagai contoh, hasil prediksi pada waktu  $t + 1$  akan digunakan sebagai input untuk memprediksi  $t + 2$ , dan seterusnya. Proses rekursif berlanjut hingga mencakup seluruh rentang prediksi yang diinginkan. Akan tetapi, rentang prediksi

jangka panjang dapat menjadi tantangan untuk metode rekursif. Akurasinya bisa jadi berkurang karena akumulasi eror (In & Jung, 2022; Rodrigo & Ortiz, 2023).

Model *machine learning* berbasis pohon keputusan, termasuk XGBoost, telah terbukti mampu memahami hubungan kompleks dan pola dalam data. Namun, kemampuan tersebut hanya terbatas pada interpolasi. Model-model ini sulit melakukan ekstrapolasi, yakni membuat prediksi berdasarkan situasi yang tidak ditemui dalam data pelatihan. Kesulitan ini bersumber dari proses pengambilan keputusan di setiap pohon yang ditentukan oleh ambang batas pemisahan (split threshold). Ambang batas pemisahan ditetapkan berdasarkan nilai-nilai fitur dalam rentang data pelatihan. Jika model menemui nilai fitur di luar rentang tersebut, model akan mengambil nilai yang terdekat dari data pelatihan. Sebagai contoh, jika data pelatihan berisi fitur waktu dengan nilai  $t_1$  dan  $t_2$ , maka tiap prediksi untuk  $t > t_2$  akan sama dengan prediksi untuk  $t = t_2$ . Konsekuensinya, jika data pelatihan menunjukkan suatu tren, model mungkin tidak bisa melanjutkan tren tersebut. Fitur waktu seringkali mempengaruhi fitur lain. Jadi, meskipun fitur waktu dihilangkan dari data pelatihan, model masih dapat mempelajari pola terkait waktu dari fitur lainnya (Gao et al., 2021; Malistov & Trushin, 2019).

### 2.2.6 Hyperparameter *Tuning* dengan Optuna

Hyperparameter *tuning* adalah langkah penting dalam optimasi model machine learning. Menurut studi Moya (2021), kinerja model machine learning dapat berubah signifikan karena hyperparameter. Akurasinya bisa berkisar dari 25% hingga 90%, tergantung pada parameter yang dipilih. Metode penyetulan hyperparameter yang tepat sangat mempengaruhi kinerja model dan efisiensi komputasi. Mengingat kompleksitas dan sifat pendekatan manual yang membutuhkan waktu, disarankan untuk menggunakan teknik hyperparameter *tuning* yang otomatis (Moya, 2021)

Sebagian besar kerangka kerja hyperparameter *tuning* yang tersedia mengharuskan pengguna untuk mendefinisikan ruang pencarian parameter secara statis, yang dapat menjadi tantangan bagi eksperimen berskala besar. Selain itu, arsitektur kerangka kerja yang ada seringkali tidak mampu mengakomodasi berbagai model. Untuk mengatasi tantangan ini, Akiba et al. (2019) memperkenalkan kerangka kerja hyperparameter *tuning* otomatis bernama Optuna. Optuna memungkinkan pengguna untuk membangun ruang pencarian secara dinamis selama berlangsungnya runtime. Dalam Optuna, tiap proses optimasi disebut sebagai sebuah objek *study*. Objek *study* memanggil method *optimize* yang menerima suatu fungsi *objective* sebagai parameter. Optuna merumuskan optimasi parameter sebagai proses meminimalkan atau memaksimalkan suatu fungsi *objective* yang didefinisikan oleh pengguna. Fungsi ini merangkum logika untuk melatih dan mengevaluasi model dengan hyperparameter yang diberikan. Fungsi *objective* biasanya mengambil satu set hyperparameter sebagai input dan mengembalikan skor yang menunjukkan performa model. Dalam prosesnya, fungsi *objective* menerima objek *trial* sebagai argumen. *Trial* digunakan untuk mengajukan nilai hyperparameter dan mengevaluasi hasil kinerjanya, sehingga memungkinkan penyesuaian yang dinamis selama proses optimasi. Pembangunan ruang pencarian secara dinamis dilakukan dengan memanggil metode *suggest* dalam objek *trial*. Metode ini memungkinkan pengguna untuk menyesuaikan ruang pencarian hyperparameter dengan kebutuhan spesifik dari model dan data yang digunakan. Selain itu, fungsi *objective* dalam Optuna bisa dibuat secara modular, dengan fungsi-fungsi terpisah untuk masing-masing model dan *optimizer*. Pendekatan modular ini menghasilkan kode yang lebih terorganisir dan memudahkan pemeliharaan serta pengembangan lebih lanjut (Akiba et al., 2019).

Pada Kode Semu 2.1, definisi fungsi *objective* dimulai pada baris ke-2 dengan objek *trial* sebagai argumen. Objek *trial* digunakan untuk memanggil method *suggest\_int* pada baris ke-3 dan baris ke-6 untuk membangun ruang pencarian bagi parameter jumlah *hidden layer* dan

jumlah unit pada tiap *hidden layer*. Parameter-parameter hasil method *suggest* digunakan untuk membuat *instance* model MLPClassifier pada baris ke-7. Setelah proses pelatihan model pada baris ke-10, fungsi *objective* akan mengembalikan nilai error rate (1.0 – akurasi) pada baris ke-11. Nilai ini akan diminimalisasi oleh Optuna dalam proses optimasi. Proses optimasi direpresentasikan oleh objek *study* yang dibuat pada baris ke-12. Pada baris ke-13, Objek *study* akan memanggil method *optimize* dengan parameter fungsi *objective* dan *n\_trials*. Setelah pemanggilan method *optimize*, proses optimasi akan berjalan dengan jumlah uji coba sebanyak *n\_trials*.

Pencarian konfigurasi parameter semakin dipercepat oleh Optuna dengan bantuan algoritma pengambilan sampel dan pemangkasan uji coba. Algoritma pengambilan sampel dalam Optuna diimplementasikan dalam objek *Sampler*. *Sampler* bertanggung jawab menentukan nilai parameter yang akan dievaluasi selanjutnya dalam proses optimasi. Tujuannya adalah mempersempit ruang pencarian untuk menemukan nilai optimal. *Sampler* default dalam Optuna adalah *TPESampler* yang menggunakan Tree-structured Parzen Estimator (TPE). Metode optimasi Bayesian ini membangun model probabilitas berdasarkan satu set hyperparameter dan data uji. Model ini kemudian digunakan untuk menyarankan konfigurasi baru yang berpotensi menghasilkan hasil baik (Akiba et al., 2019). Jenis *Sampler* Optuna yang digunakan dalam penelitian ini adalah *GridSampler*, yang mengeksplorasi semua kemungkinan kombinasi parameter dalam ruang pencarian yang ditentukan (Xing et al., 2024).

Kode sumber 2.2 mendemonstrasikan penggunaan objek *Sampler*, khususnya *GridSampler* dalam proses optimasi parameter dengan Optuna. Baris ke-32 hingga ke-65 mendefinisikan fungsi *objective* yang bertujuan mengoptimalkan dua variabel *x* dan *y*. Selain dibangun melalui method *suggest* dalam fungsi *objective*, ruang pencarian parameter juga perlu didefinisikan secara terpisah sebagai *dictionary search\_space* pada baris ke-9 hingga ke-12. Selanjutnya, baris ke-15 membuat sebuah *study* Optuna dengan *GridSampler* yang menerima *dictionary search\_space* sebagai parameter. *GridSampler* akan mencoba semua kombinasi parameter yang mungkin dari *search\_space*. Setelah optimasi dijalankan dengan 25 *trial* pada baris ke-18, hasil parameter dan nilai fungsi *objective* terbaik akan ditampilkan pada baris ke-21 dan ke-22.

Contoh Penggunaan Fungsi Objective
Input: Trial
Output: Skor fungsi objective yang dihasilkan oleh trial
<pre> 1. IMPORT LIBRARY optuna, sklearn.datasets, sklearn.model_selection,    sklearn.neural_network 2. <b>FUNCTION objective(trial):</b> 3.     n_layers = trial.SUGGEST_INT('n_layers', 1, 4) 4.     ARRAY layers 5.     <b>FOR</b> i <b>IN</b> RANGE(n_layers): 6.         layers.APPEND(trial.SUGGEST_INT(f'n_units_{i}', 1, 128)) 7.     clf = MLPClassifier(hidden_layer_sizes=layers) 8.     mnist = fetch_mldata('MNIST original') 9.     x_train, x_test, y_train, y_test = train_test_split(mnist.data,    mnist.target) 10.    clf.FIT(x_train, y_train) 11.    <b>RETURN</b> 1.0 - clf.SCORE(x_test, y_test) 12.    study = optuna.create_study() 13.    study.OPTIMIZE(objective, n_trials=100) </pre>

Kode Semu 2.1 Contoh Fungsi Objective Optuna

Contoh Penggunaan Fungsi Objective dengan GridSampler
Input: Trial
Output: Skor fungsi objective yang dihasilkan oleh trial
<pre> 1. IMPORT LIBRARY optuna </pre>

```

2. FUNCTION objective(trial):
3.     x = trial.SUGGEST_FLOAT("x", -5, 5)
4.     y = trial.SUGGEST_INT("y", -5, 5)
5.     RETURN x^2 + y^2
6. DEFINE search_space:
7.     x: [-5, -2.5, 0, 2.5, 5]
8.     y: [-5, -2, 0, 2, 5]
9. study =
    optuna.CREATE_STUDY(sampler=optuna.samplers.GridSampler(search_space))
10. study.OPTIMIZE(objective, n_trials=25)
11. PRINT "Best parameters:", study.BEST_PARAMS
12. PRINT "Best value:", study.BEST_VALUE
13. FOR trial IN study.trials:
14.     PRINT "Trial", trial.number, ": params =", trial.params, ",
        value =", trial.value

```

Kode Semu 2.2 Contoh Fungsi Objective Optuna dengan *GridSampler*

Sementara itu, algoritma pemangkasan uji coba dalam Optuna diimplementasikan dalam objek Pruner. Pruner berfungsi untuk mengeliminasi uji coba yang dianggap tidak menjanjikan. Pruner bekerja dengan memantau nilai objektif optimasi secara berkala dan memangkas uji coba yang tidak memenuhi kondisi tertentu. Pruner memainkan peran penting dalam mengendalikan biaya komputasi setiap uji coba. Dengan memangkas uji coba yang kurang menjanjikan, Pruner memastikan bahwa sumber daya komputasi dapat dioptimalkan untuk kandidat yang paling potensial, sehingga meningkatkan efisiensi proses optimasi hyperparameter secara keseluruhan (Akiba et al., 2019). Algoritma pemangkasan utama yang digunakan dalam Optuna adalah varian dari Asynchronous Successive Halving Algorithm (ASHA), yang diimplementasikan dalam *SuccessiveHalvingPruner*. Algoritma ini mendukung eksekusi pekerjaan secara asinkron dan paralel dengan kemampuan untuk *early stopping* berdasarkan peringkat sementara. Pendekatan ini sangat efektif untuk aplikasi yang proses optimasi parameternya tersebar di beberapa unit komputasi (Akiba et al., 2019).

Kode Semu 2.3 menyediakan contoh cara penggunaan *pruning* atau pemangkasan dalam Optuna. Pada baris ke-7 hingga ke-13, model dilatih secara bertahap dengan metode partial-fit selama 100 iterasi. Dalam setiap iterasi, hasil performa model akan dilaporkan melalui method *report* milik objek *trial* pada baris ke-10. Mekanisme *pruning* mulai diterapkan pada baris ke-11 dan ke-12. Jika Optuna menentukan bahwa *trial* saat ini tidak menjanjikan, *trial* tersebut akan dipangkas untuk menghemat sumber daya komputasi. Penentuan *trial* mana yang tidak menjanjikan dilakukan oleh objek Pruner yang didefinisikan pada baris ke-16 sampai ke-19. Khususnya, jenis Pruner yang digunakan dalam Kode Semu 2.3 adalah *SuccessiveHalvingPruner*. *SuccessiveHalvingPruner* menganggap suatu *trial* tidak menjanjikan jika memiliki nilai *objective* sementara berada di bawah kinerja *trial-trial* lain pada tahap evaluasi yang sama.

Contoh Penggunaan Fungsi <i>Objective</i> dengan <i>SuccessiveHalvingPruner</i>
Input: Trial
Output: Skor fungsi <i>objective</i> yang dihasilkan oleh trial
<pre> 1. IMPORT LIBRARY optuna, sklearn.datasets, sklearn.model_selection,     sklearn.linear_model, numpy 2. <b>FUNCTION</b> objective(trial): 3.     X, y = load_iris(return_X_y=True) 4.     X_train, X_valid, y_train, y_valid = train_test_split(X, y,         test_size=0.25, random_state=0) 5.     alpha = trial.SUGGEST_FLOAT("alpha", 1e-5, 1e-1, log=True) 6.     clf = SGDClassifier(alpha=alpha) 7.     <b>FOR</b> step IN RANGE(100): </pre>

```

8.         clf.partial_fit(X_train, y_train, classes=np.unique(y))
9.         intermediate_value = clf.SCORE(X_valid, y_valid)
10.        trial.REPORT(intermediate_value, step)
11.        IF trial.SHOULD_PRUNE():
12.            RAISE optuna.TrialPruned()
13.        RETURN clf.SCORE(X_valid, y_valid)
14.    study = optuna.CREATE_STUDY(
15.        direction="maximize",
16.        pruner=optuna.pruners.SuccessiveHalvingPruner(
17.            min_resource=1,
18.            reduction_factor=4,
19.            min_early_stopping_rate=0
20.        )
21.    )
22.    study.OPTIMIZE(objective, n_trials=100)

```

Kode Semu 2.3 Contoh Fungsi Objective Optuna dengan *SuccessiveHalvingPruner*

## 2.2.7 Pengujian Evaluasi dengan MAE, RMSE, dan MAPE

Metode evaluasi kinerja model prediksi deret waktu meliputi dua kategori, yakni tergantung skala (*scale-dependent*) dan tidak tergantung skala (*scale-independent*) - Referensi. Metode evaluasi *scale-dependent* menggunakan skala dan satuan yang sama dengan data. Tugas akhir ini akan menggunakan *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) sebagai metode *scale-dependent*.

### a. *Scale Dependent: Mean Absolute Error (MAE)*

MAE mengukur rata-rata selisih absolut antara nilai hasil prediksi dan nilai aktual. Rentang nilainya adalah  $(0, +\infty)$ . Nilai MAE yang lebih rendah menunjukkan akurasi yang lebih tinggi. Keuntungan dari penggunaan MAE adalah memiliki satuan yang sama dengan data asli, sehingga mudah dihitung dan dipahami (Jierula et al., 2021).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |D_{\text{pre},i} - D_{\text{act},i}| \quad (2.16)$$

Di mana

- $n$  : Jumlah data
- $D_{\text{pre},i}$  : Hasil prediksi ke- $i$
- $D_{\text{act},i}$  : Nilai aktual ke- $i$

### b. *Scale Dependent: Root Mean Squared Error (RMSE)*

Perhitungan RMSE didasarkan pada Mean Squared Error (MSE). MSE mengukur rata-rata selisih kuadrat antara nilai prediksi dan nilai aktual. Karena perhitungannya menggunakan kuadrat, selisih besar antara nilai prediksi dan nilai aktual mendapat bobot lebih tinggi. Namun, hal ini juga membuat satuan MSE menjadi kuadrat dari satuan data asli. Untuk menyamakan satuannya dengan data asli, MSE diubah menjadi RMSE dengan diambil akar kuadratnya. Sama seperti MAE, jangkauan nilai RMSE adalah  $(0, +\infty)$  dan nilai RMSE yang lebih rendah menunjukkan akurasi yang lebih tinggi. (Jierula et al., 2021). Secara matematis RMSE didefinisikan sebagai berikut:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (D_{\text{pre},i} - D_{\text{act},i})^2} \quad (2.17)$$

Di mana

- $n$  : Jumlah data
- $D_{\text{pre},i}$  : Hasil prediksi ke- $i$
- $D_{\text{act},i}$  : Nilai aktual ke- $i$

Kelemahan metode dalam kategori ini adalah sulit membandingkan akurasi model prediksi pada data dengan beragam skala dan satuan. (Jierula et al., 2021; S. Kim & Kim, 2016). Sebagai contoh, kesalahan prediksi sebesar 10 satuan dapat mempunyai tingkat keparahan yang berbeda-beda, tergantung pada skala data (Jierula et al., 2021). Oleh karena itu, metode *scale-independent* juga perlu dipertimbangkan.

**c. *Scale Independent: Mean Absolute Percentage Error (MAPE)***

Metode evaluasi *scale-independent* mampu membandingkan akurasi model prediksi pada data dengan beragam skala dan satuan. Kesalahan dalam bentuk persentase juga lebih mudah dipahami dan ditafsirkan, khususnya jika skala data asli tidak diketahui. Sebagai contoh, jika peninjau tidak mengetahui skala data, laporan yang menyatakan "kesalahan prediksi 5%" akan lebih bermakna daripada "kesalahan prediksi 50 cm" (Jierula et al., 2021; S. Kim & Kim, 2016). Tugas akhir ini akan menggunakan *Mean Absolute Percentage Error* (MAPE) sebagai metode *scale-independent*. MAPE mengukur persentase rata-rata selisih absolut antara nilai hasil prediksi dan nilai aktual (Jierula et al., 2021). Secara matematis MAPE didefinisikan sebagai berikut:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{D_{\text{pre},i} - D_{\text{act},i}}{D_{\text{act},i}} \right| \quad (2.18)$$

Di mana

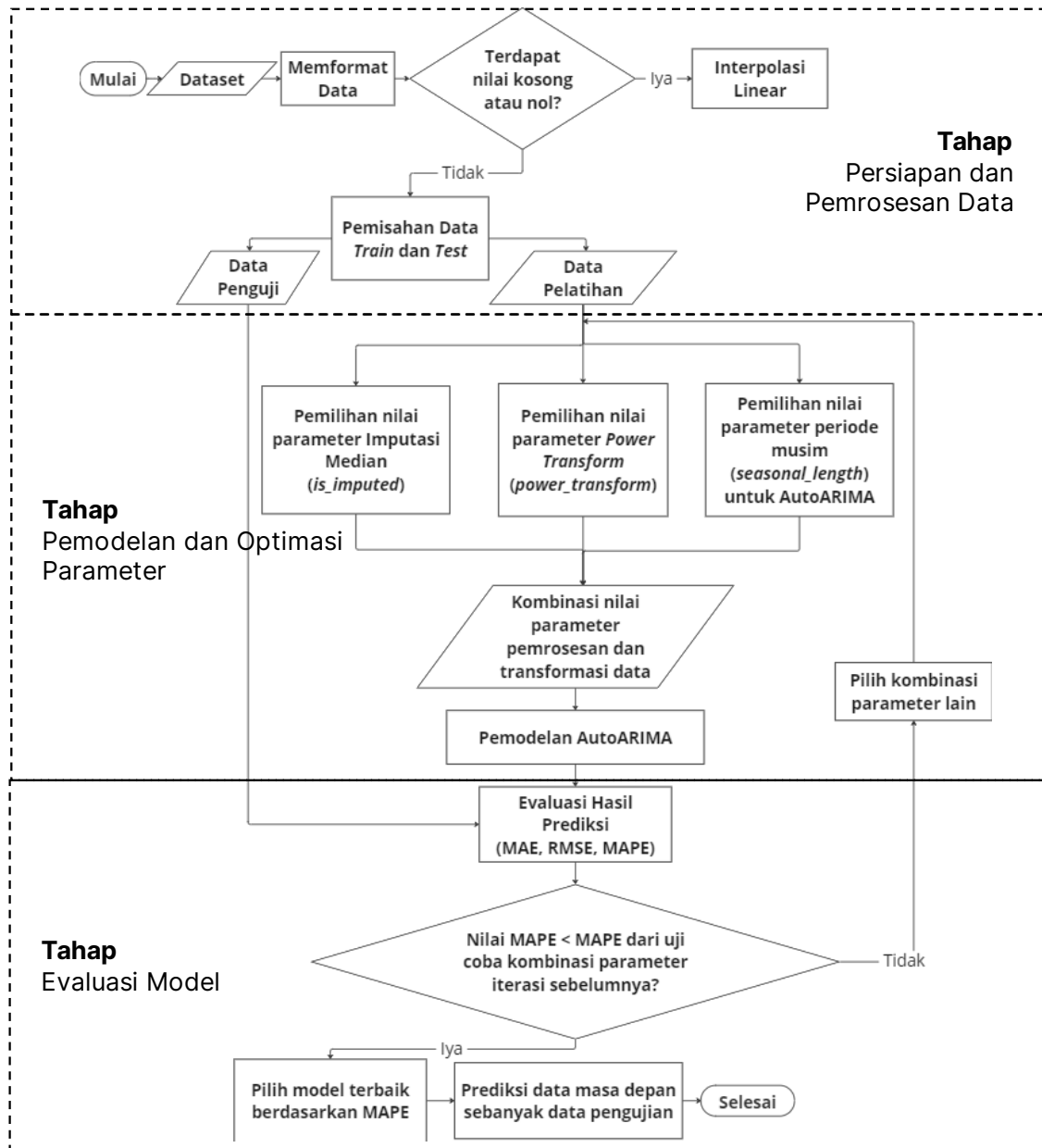
- $n$  : Jumlah data
- $D_{\text{pre},i}$  : Hasil prediksi ke- $i$
- $D_{\text{act},i}$  : Nilai aktual ke- $i$

MAPE adalah salah satu ukuran akurasi model prediksi yang paling populer di kalangan praktisi industri. MAPE juga direkomendasikan di banyak buku dan menjadi ukuran akurasi utama dalam kompetisi peramalan Makridakis (S. Kim & Kim, 2016). Meskipun mudah dipahami dan ditafsirkan, metode dalam kategori *scale-independent* sensitif terhadap data pencilan. Jika nilai aktualnya nol atau mendekati nol, MAPE akan menjadi tak terhingga atau tak terdefinisi (Jierula et al., 2021; S. Kim & Kim, 2016; Sukarsa et al., 2021).

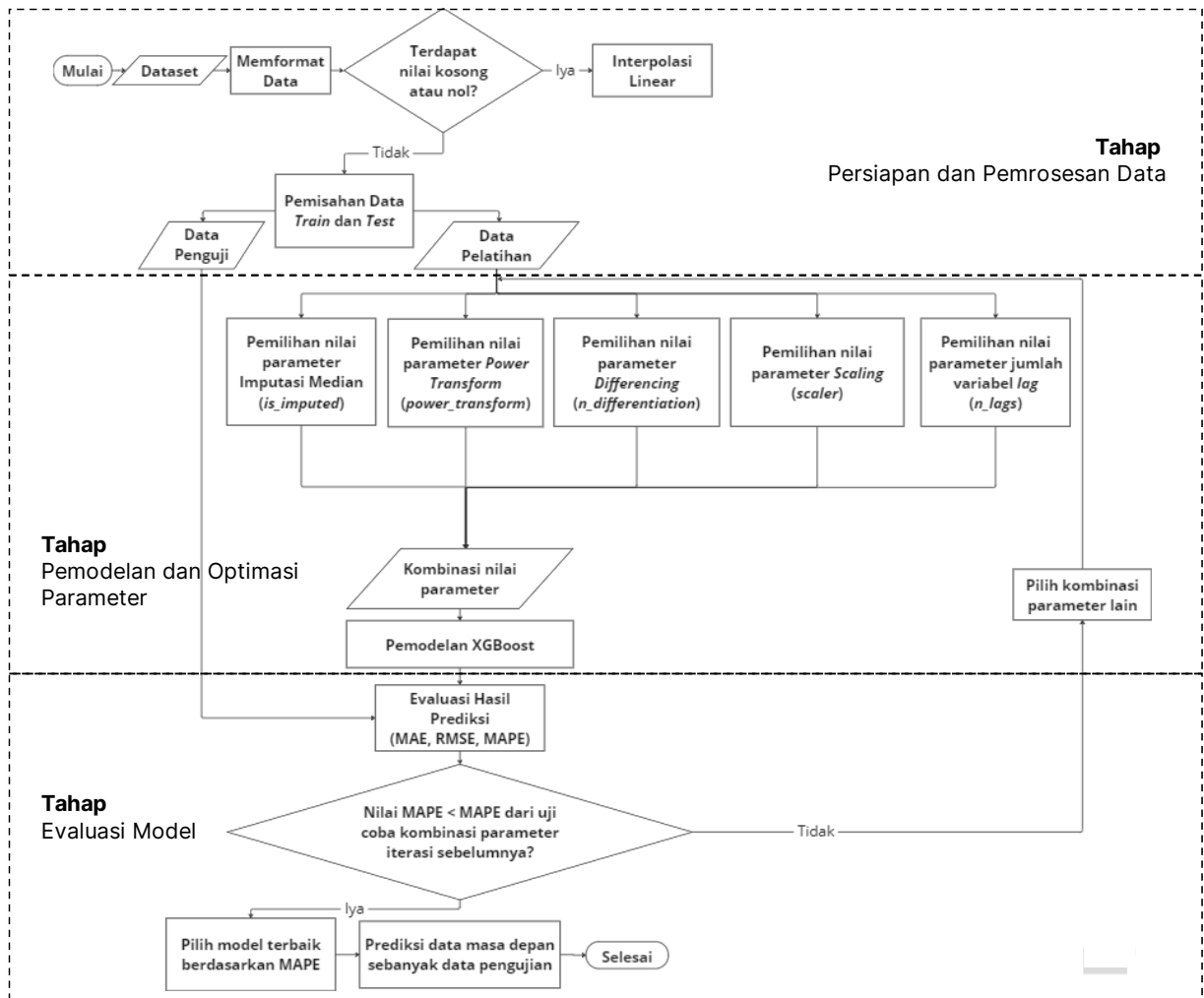
## BAB 3 METODOLOGI

### 3.1 Metode yang Digunakan

Dalam tugas akhir ini, digunakan dua pendekatan dalam merancang model prediksi, yaitu metode statistik dan metode *machine learning*. Metode statistik menggunakan model AutoARIMA, sedangkan metode *machine learning* menggunakan model XGBoost. Setiap pendekatan melibatkan tahap persiapan dan pemrosesan data, pemodelan dan optimasi parameter, serta evaluasi model. Diagram alir metode statistik ditampilkan dalam Gambar 3.1. Sementara itu, diagram alir metode *machine learning* ditampilkan pada Gambar 3.2. Perbedaan utama kedua diagram alir terletak pada parameter yang dipertimbangkan dalam proses *tuning*.



Gambar 3.1 Diagram Alir Uji Coba dengan AutoARIMA



Gambar 3.2 Diagram Alir Uji Coba dengan *Machine Learning*

Tabel 3.1 Skenario Uji Coba

Uji Coba	Data Harga Pupuk yang Digunakan	Tujuan
Uji Coba 1	DAP	Mengidentifikasi model dengan metrik kesalahan terendah untuk rentang prediksi 3 Bulan, 6 Bulan, dan 12 Bulan. Melakukan analisis terhadap tingkat kepentingan parameter pemrosesan dan transformasi data.
Uji Coba 2	KCL	
Uji Coba 3	RP	
Uji Coba 4	TSP	
Uji Coba 5	Urea	

Penelitian ini melibatkan serangkaian uji coba yang dirancang untuk menganalisis dan membandingkan performa model AutoARIMA dan XGBoost dalam memprediksi harga lima jenis pupuk, yakni DAP, KCL, RP, TSP, dan urea. Tabel 3.1 menyajikan rincian skenario untuk masing-masing uji coba yang dilakukan. Setiap uji coba bertujuan membandingkan performa kedua model dalam memprediksi harga pupuk tertentu. Uji Coba 1 berfokus pada pupuk DAP, Uji Coba 2 pada pupuk KCL, Uji Coba 3 pada pupuk RP, Uji Coba 4 pada pupuk TSP, dan Uji Coba 5 pada pupuk urea. Melalui serangkaian uji coba ini, akan diidentifikasi model yang menghasilkan metrik kesalahan terendah, khususnya MAPE, untuk setiap rentang prediksi 3 bulan, 6 bulan, dan 12 bulan. Selain itu, analisis terhadap tingkat kepentingan parameter pemrosesan dan transformasi data juga akan dilakukan dalam setiap uji coba.



## 3.2 Bahan dan Peralatan yang Digunakan

### 3.2.1 Spesifikasi Perangkat Keras

Tugas akhir ini menggunakan Google Colaboratory untuk melakukan implementasi pada metode yang telah dirancang, dan menggunakan perangkat keras berupa sebuah laptop dengan spesifikasi yang memadai, sehingga penelitian dapat berjalan dengan baik. Spesifikasi perangkat keras tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Keras

1.	<b>Sistem Operasi</b>	Windows 11 Home 64-bit (10.0, Build 22621)
2.	<b>Memori</b>	16384 MB RAM
3.	<b>CPU</b>	13 <sup>th</sup> Gen Intel(R) Core(TM) i7-13620H
4.	<b>GPU</b>	NVIDIA GeForce RTX 4060

### 3.2.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan untuk penyelesaian tugas akhir ini mencakup bahasa pemrograman dan pustaka. Bahasa pemrograman yang digunakan untuk pembangunan dan pelatihan model prediksi adalah Python 3.10.12. Pustaka Python yang digunakan serta penggunaannya dijelaskan pada Tabel 3.3.

Tabel 3.3 Spesifikasi Perangkat Lunak

<b>Tahapan</b>	<b>Pustaka yang Digunakan</b>
Persiapan dan Pemrosesan Data	Pandas v1.5.3, Sklearn v1.2.2, Statsmodels v0.13.2, Scipy v1.11.4
Pemodelan dan Optimasi Parameter	Skforecast v0.11.0, XGBoost v2.0.3, Pmdarima v2.0.4, Optuna v3.6.1
Evaluasi Model	Pandas v1.5.3, Numpy v1.23.5, Sklearn v1.2.2

## 3.3 Implementasi

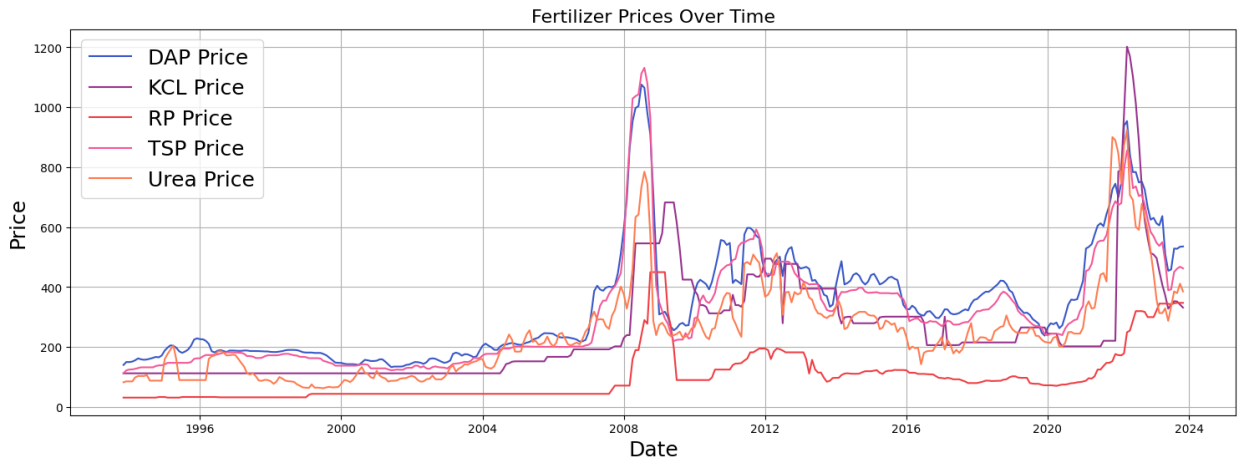
Penelitian tentang prediksi harga pasar pupuk diimplementasikan dengan bahan dan peralatan yang telah disebutkan di Bagian 3.2. Implementasi penelitian dibagi menjadi tiga tahapan, yaitu tahap persiapan dan pemrosesan data, tahap pemodelan dan optimasi parameter, serta tahap evaluasi.

### 3.3.1 Tahap Persiapan dan Pemrosesan Data

Data yang digunakan berupa data harga lima jenis pupuk dari situs portal data Index Mundi. Lima jenis pupuk yang menjadi fokus, yaitu pupuk *diammonium phosphate* (DAP), *potassium chloride* (KCL), *rock phosphate* (RP), *triple superphosphate* (TSP), dan urea. Data harga kelima jenis pupuk dikumpulkan oleh Index Mundi dari *Fertilizer Week*, *Fertilizer International*, dan Bank Dunia. Masing-masing jenis pupuk mempunyai data historis dengan total 361 baris dan dua kolom, yakni kolom *date* dan kolom *price*, seperti yang ditampilkan dalam Tabel 3.4. Kolom *date* mempunyai frekuensi bulanan mulai November 1993 sampai November 2023. Sementara itu, harga pupuk pada kolom *price* mempunyai satuan Dolar AS per metrik ton. Harga kelima jenis pupuk seiring waktu divisualisasikan dalam Gambar 3.3.

Tabel 3.4 Penjelasan Kolom-kolom Data

Kolom	Type Data	Penjelasan
date	DateTime	Bulan pencatatan harga mulai November 1993 hingga November 2023.
price	Float	Harga pupuk dengan satuan Dolar AS per metrik yang digunakan sebagai variable target prediksi.



Gambar 3.3 Plot Harga Kelima Jenis Pupuk

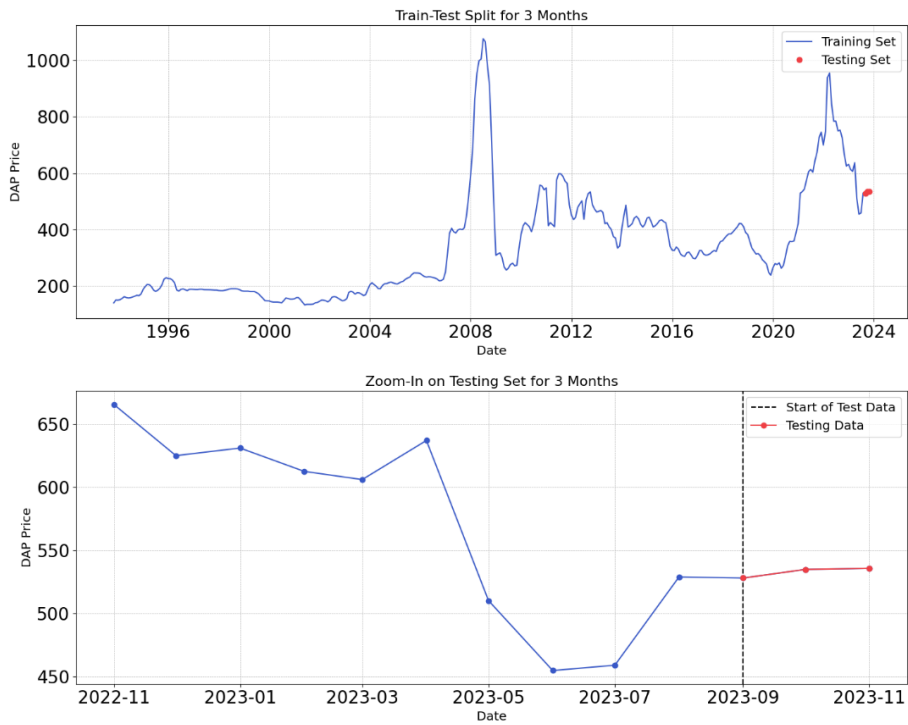
```

Pemuatan dan Persiapan Data
Input: Data harga pupuk
Output: Data pelatihan dan data pengujian
1. IMPORT LIBRARY pandas, numpy, matplotlib.pyplot
2. IMPORT FUNCTION train_test_split FROM sklearn.model_selection
3. DOWNLOAD DATASET EXCEL FILE
4. LOAD EXCEL FILE INTO VARIABLE data
5. SET date_col TO 'date'
6. SET target_col TO 'dap_price'
7. CONVERT date_col TO DATETIME IN data
8. SET date_col AS INDEX IN data
9. SET FREQUENCY OF data TO MONTH START
10. REPLACE 0 VALUES IN target_col WITH NaN
11. IF ANY VALUES IN target_col ARE NaN:
12.     INTERPOLATE MISSING VALUES IN target_col USING LINEAR METHOD
13. SET n_test TO 12
14. SPLIT data INTO train_data AND test_data WITH test_data SIZE OF
    n_test WITHOUT SHUFFLING
    
```

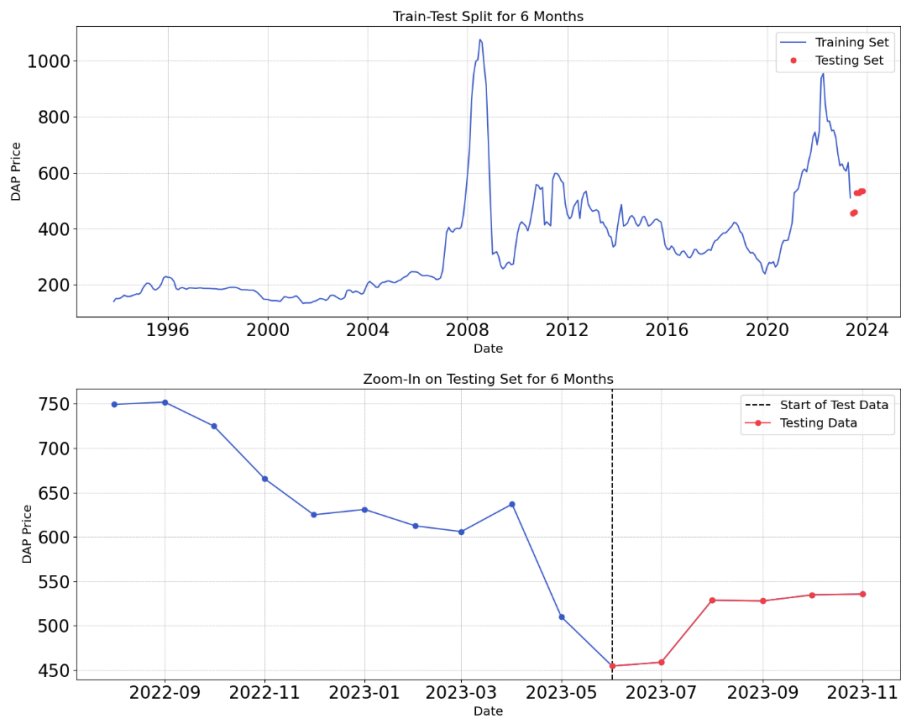
Kode Semu 3.1 Pemuatan dan Persiapan Data

Setelah data diunduh dan dimuat dari Index Mundi, dijalankan tahap persiapan dan pemrosesan data pada Kode Semu 3.1. Tahap ini dimulai dengan penentuan kolom tanggal dan kolom target pada baris ke-5 dan ke-6. Pada baris ke-7 hingga ke-9, kolom tanggal dikonversi menjadi tipe data *datetime* dan dijadikan sebagai indeks data. Indeks data lalu diatur frekuensinya menjadi awal bulanan atau *month start* (MS). Baris ke-11 memeriksa keberadaan nilai-nilai nol dalam kolom target. Nilai-nilai nol yang terdeteksi akan digantikan dengan nilai NaN. Nilai NaN atau *missing values* akan diisi dengan metode interpolasi linier pada baris ke-12. Setelah penanganan *missing values*, data dipisah menjadi set pelatihan dan pengujian pada baris ke-14. Ukuran set pengujian diatur oleh variabel *n\_test*. Variabel ini juga mengatur rentang prediksi model. Sebagai contoh, dalam Kode Semu 3.1, variabel *n\_test* bernilai 12, yang berarti model akan memprediksi 12 bulan setelah tanggal terakhir set pelatihan. Dalam penelitian ini, digunakan tiga rentang prediksi, yakni 3 bulan, 6 bulan, dan 12 bulan. Gambar

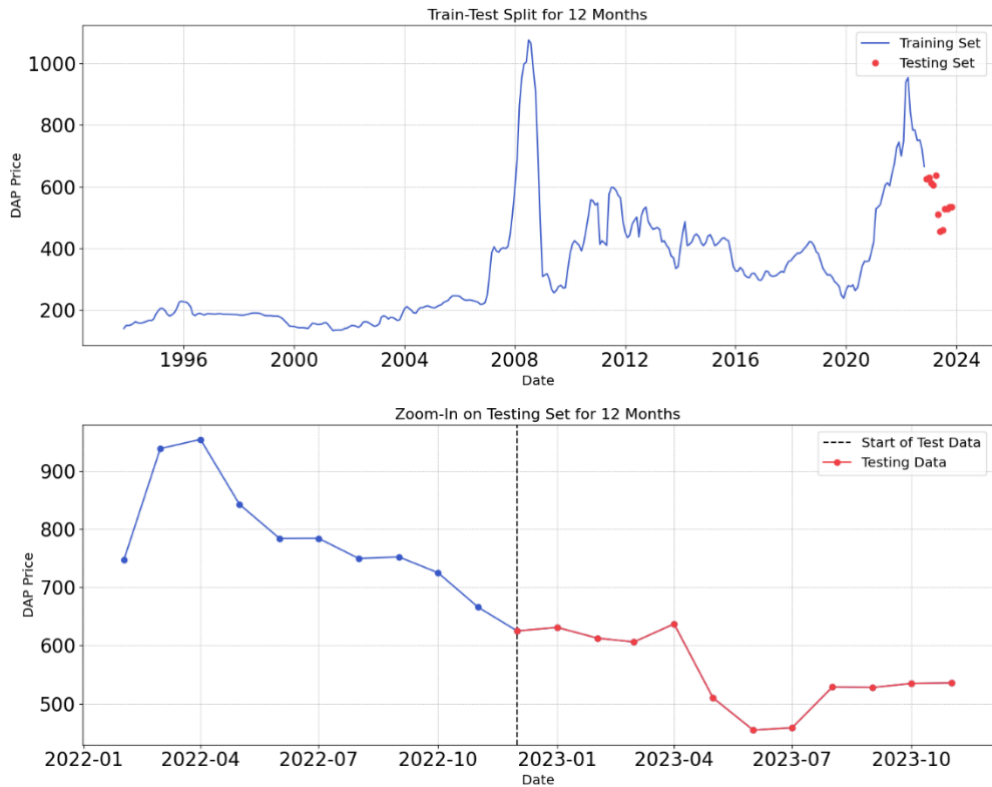
3.4, Gambar 3.5, dan Gambar 3.6 memperlihatkan contoh visualisasi pemisahan set pelatihan dan pengujian pada data harga pupuk DAP untuk prediksi 3 bulan sampai 12 bulan.



Gambar 3.4 Line Plot Pemisahan Data Latih dan Uji untuk Prediksi 3 Bulan



Gambar 3.5 Line Plot Pemisahan Data Latih dan Uji untuk Prediksi 6 Bulan



Gambar 3.6 *Line Plot* Pemisahan Data Latih dan Uji untuk Prediksi 12 Bulan

### 3.3.2 Pemodelan dan Optimasi Parameter

Tahap pemodelan mempunyai beberapa fungsi pendukung yang dijelaskan dalam Kode Semu 3.2 hingga 3.5. Fungsi-fungsi pendukung tersebut meliputi proses deteksi dan imputasi data pencilan, aplikasi dan inversi *power transform*, serta optimasi parameter.

#### a. Deteksi dan Imputasi Data Pencilan

Kode Semu 3.2 menjelaskan fungsi *detect\_and\_impute\_outlier* untuk deteksi dan imputasi data pencilan yang menerima *dataframe* dan kolom target sebagai parameter. Deteksi data pencilan dilakukan dengan menggunakan rentang interkuartil (IQR). Perhitungan IQR dimulai pada baris ke-3 yang menghitung kuartil pertama (Q1) dan kuartil ketiga (Q3) dari kolom target. Kemudian, pada baris ke-4, IQR dihitung sebagai selisih antara Q3 dan Q1. IQR lalu digunakan untuk menetapkan batas atas dan batas bawah data pencilan. Pada baris ke-5, ditetapkan  $Q1 - 1.5 * IQR$  sebagai batas bawah dan  $Q3 + 1.5 * IQR$  sebagai batas atas. Titik-titik data yang berada di luar kedua batas tersebut dianggap sebagai data pencilan. Proses imputasi atau penggantian data pencilan dimulai pada baris ke-6 dimana objek *SimpleImputer* diinisialisasi dengan strategi median. Pada baris ke-8, nilai pada indeks-indeks data pencilan akan diganti dengan NaN. Untuk mengisi nilai NaN dengan nilai median dari kolom target, *SingleImputer* diterapkan pada baris ke-9. Kemudian, baris ke-10 membuat *dataframe* baru bernama *result* sebagai salinan dari *dataframe* input. Data hasil imputasi akhirnya disimpan dalam *result* pada baris ke-11 dan dikembalikan oleh fungsi pada baris ke-12.

```

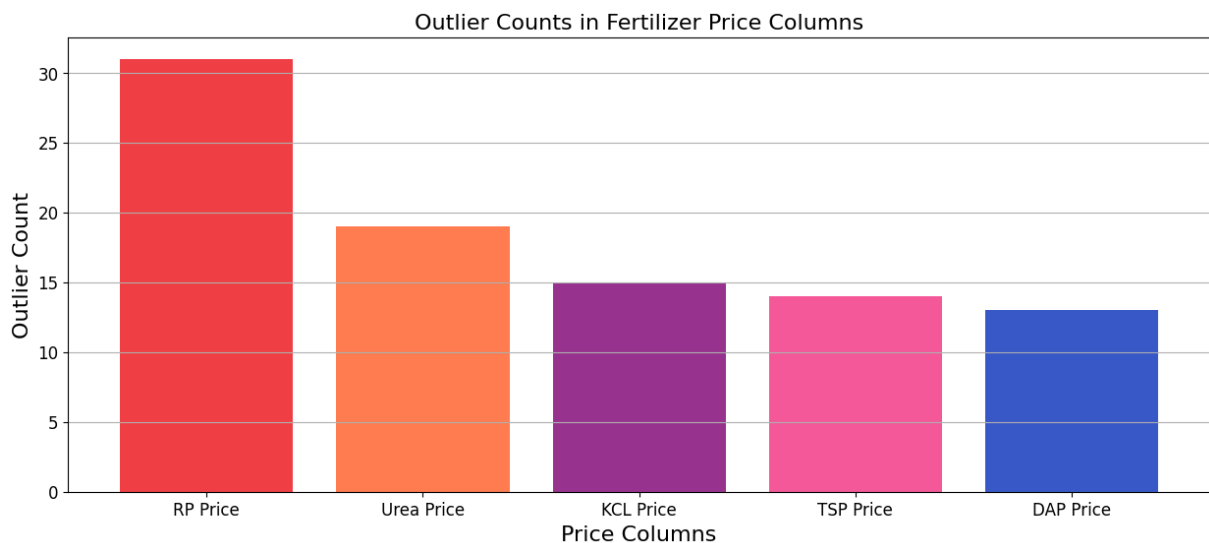
Fungsi Deteksi dan Imputasi Data Pencilan
Input: Data harga pupuk dan nama kolom target prediksi
Output: Data harga pupuk dengan data pencilan yang sudah diganti median
1. IMPORT FUNCTION SimpleImputer FROM sklearn.impute
2. FUNCTION detect_and_impute_outlier(data, target_col):
3.     CALCULATE Q1 AND Q3 AS 25TH AND 75TH PERCENTILES OF data[target_col]
4.     CALCULATE IQR AS Q3 - Q1
5.     FIND INDICES OF OUTLIERS IN data WHERE data[target_col] IS LESS
   THAN (Q1 - 1.5 * IQR) OR GREATER THAN (Q3 + 1.5 * IQR)
6.     INITIALIZE SimpleImputer WITH MEDIAN STRATEGY
7.     CREATE imputed_data AS A DATAFRAME WITH THE SAME INDEX AS data AND
   data[target_col]
8.     SET VALUES OF target_col TO NaN IN imputed_data FOR OUTLIER INDICES
9.     APPLY IMPUTER TO imputed_data[target_col] TO REPLACE NaNs WITH
   MEDIAN VALUES
10.    CREATE result AS A DEEP COPY OF data
11.    SET result[target_col] TO imputed_data[target_col]
12.    RETURN result

```

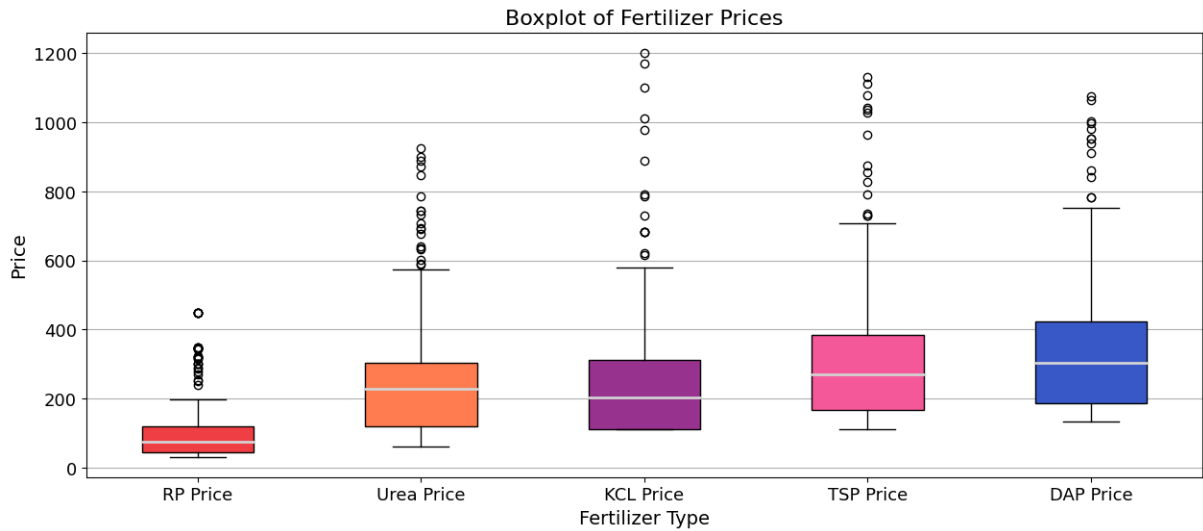
Kode Semu 3.2 Fungsi Deteksi dan Imputasi Data Pencilan

Tabel 3.5 Perbandingan Jumlah Data Pencilan Tiap Jenis Pupuk

Jenis Pupuk	Jumlah Data Pencilan
RP	31
Urea	19
KCL	15
TSP	14
DAP	13



Gambar 3.7 Bar Plot Jumlah Data Pencilan pada Kolom Harga Pupuk



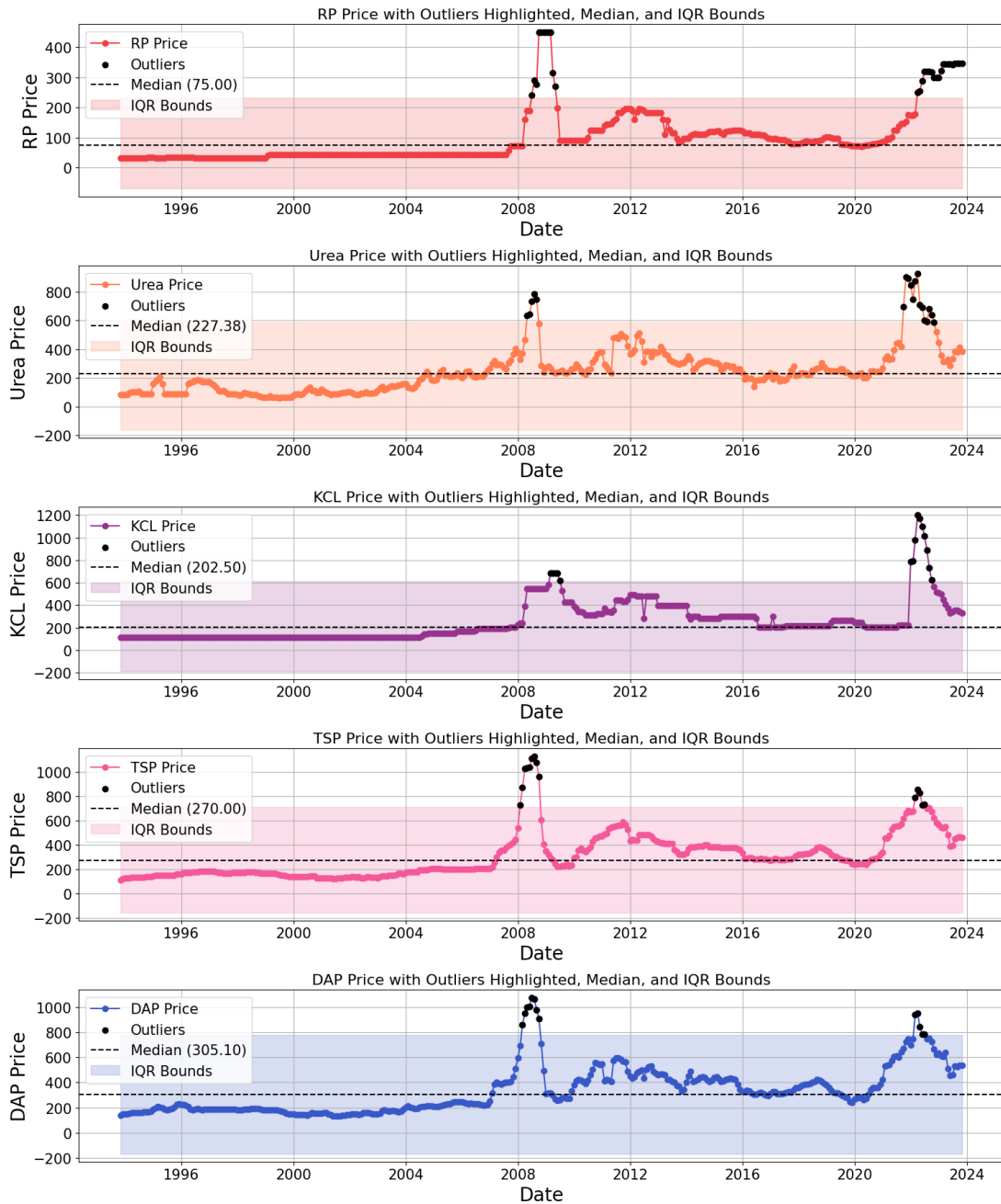
Gambar 3.8 Box Plot Data Pencilan pada Kolom Harga Pupuk

Tabel 3.5 dan Gambar 3.7 menunjukkan perbandingan jumlah data pencilan untuk tiap kolom harga pupuk. Terlihat bahwa pupuk RP memiliki jumlah data pencilan terbanyak, yakni sejumlah 31 titik data. Pupuk urea berada di posisi kedua dengan 19 titik data pencilan. Sementara itu, pupuk KCL, TSP, dan DAP memiliki jumlah data pencilan yang relatif serupa. Pupuk DAP memiliki jumlah data pencilan paling sedikit di antara ketiganya, yakni sejumlah 13 titik data.

Gambar 3.8 menyajikan visualisasi Box Plot dari kelima kolom harga pupuk. Garis di dalam setiap box menandai harga median, yang terlihat meningkat dari kiri ke kanan. Pupuk RP menunjukkan harga median terendah, mengindikasikan harga pasar yang paling murah. Sebaliknya, pupuk DAP memiliki median tertinggi, menandakan harga pasar yang lebih mahal dibandingkan jenis pupuk lainnya.

Panjang setiap box dalam Box Plot menunjukkan IQR, yang merepresentasikan variabilitas harga pupuk. Garis di bawah dan di atas setiap box menunjukkan batas bawah dan batas atas data normal. Batas bawah umumnya bernilai  $Q1 - 1.5 * IQR$  dan batas atas bernilai  $Q3 + 1.5 * IQR$ . Titik data yang berada di luar batas ini dianggap sebagai data pencilan.

Pupuk RP memiliki IQR terkecil yang menandakan variabilitas harga terendah di antara pupuk lainnya. IQR yang kecil pada pupuk RP menghasilkan rentang nilai yang sempit antara batas bawah dan batas atas. Konsekuensinya, lebih banyak titik data yang berada di luar batas dan dianggap sebagai data pencilan. Di sisi lain, pupuk urea, KCL, TSP, dan DAP memiliki rentang nilai yang lebih luas antara batas bawah dan batas atas. Rentang nilai yang lebih luas ini menyebabkan jumlah data pencilan lebih sedikit pada keempat jenis pupuk tersebut. Meskipun memiliki jumlah data pencilan yang lebih sedikit, nilai data pencilan pada pupuk urea, KCL, TSP, dan DAP cenderung lebih tinggi dibandingkan dengan RP. Hal ini terlihat pada pupuk KCL yang memiliki data pencilan dengan nilai tertinggi, mencapai sekitar \$1200.



Gambar 3.9 Line Plot Visualisasi Data Pencilan pada Kolom Harga Pupuk

Tabel 3.6 Perbandingan Jumlah Data Pencilan Tiap Jenis Pupuk setelah Imputasi

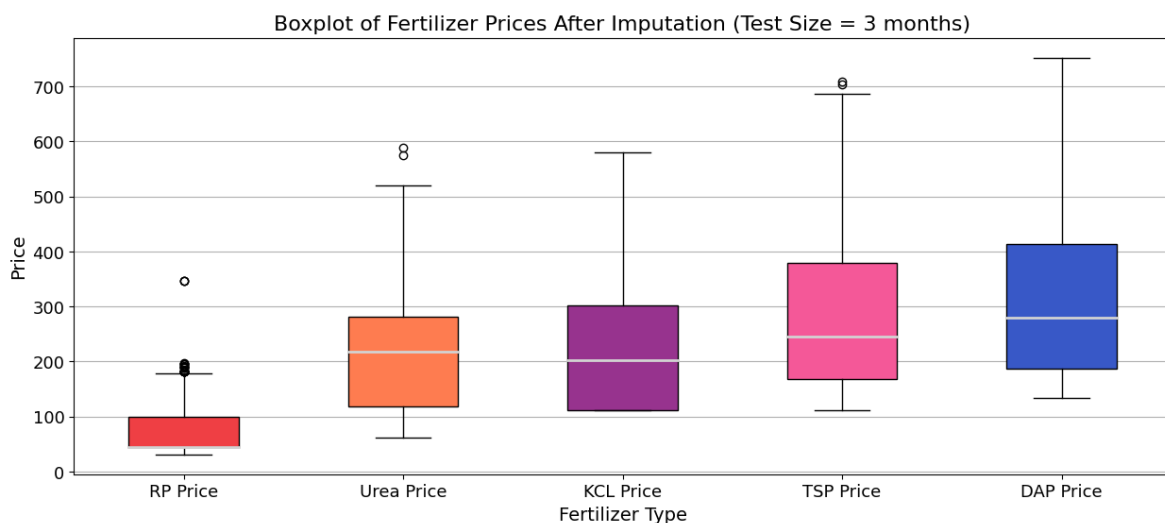
Fertilizer Type	Jumlah Data Pencilan Sebelum Imputasi	Jumlah Data Pencilan Setelah Imputasi		
		Prediksi 3 Bulan	Prediksi 6 Bulan	Prediksi 12 Bulan
RP	31	24	12	12
Urea	19	2	1	0
KCL	15	0	0	0
TSP	14	2	2	1
DAP	13	0	0	0

Gambar 3.9 memvisualisasikan perubahan harga lima jenis pupuk seiring waktu, disertai penandaan titik-titik data pencilan. Titik-titik data pencilan ditandai dengan warna hitam di luar daerah berwarna. Daerah berwarna menunjukkan rentang nilai antara batas bawah dan batas atas, sedangkan garis hitam dalam daerah berwarna merepresentasikan nilai median untuk tiap jenis pupuk. Secara keseluruhan, semua jenis pupuk menunjukkan pola naik turun harga yang serupa, terutama pada TSP dan DAP. Fluktuasi harga yang serupa mengindikasikan bahwa seluruh industri pupuk dipengaruhi oleh kekuatan pasar yang sama. Observasi ini menyoroti keterhubungan antar pasar pupuk global, di mana faktor-faktor ekonomi makro dan geopolitik dapat berdampak luas pada berbagai jenis produk pupuk.

Analisis tren mengungkapkan bahwa semua jenis pupuk memiliki harga yang relatif stabil dari tahun 1996 hingga 2007. Namun, stabilitas ini terganggu oleh dua lonjakan besar yang memunculkan titik-titik data pencilan. Lonjakan pertama terjadi sekitar tahun 2008-2009 dan lonjakan kedua terjadi sekitar tahun 2022-2023. Kedua lonjakan harga tersebut mencerminkan sensitivitas pasar pupuk terhadap guncangan ekonomi global dan geopolitik. Krisis moneter global 2008 menyebabkan peningkatan harga energi dan komoditas, yang berdampak langsung pada biaya produksi dan distribusi pupuk. Lonjakan harga tahun 2022-2023 dapat dikaitkan dengan pemulihan ekonomi pascapandemi yang meningkatkan permintaan pupuk, serta konflik Rusia-Ukraina yang mengganggu rantai pasokan global. Setelah kedua lonjakan harga tersebut, mayoritas harga pupuk kembali turun. Namun, harga pupuk RP justru cenderung stagnan pada level yang tinggi.

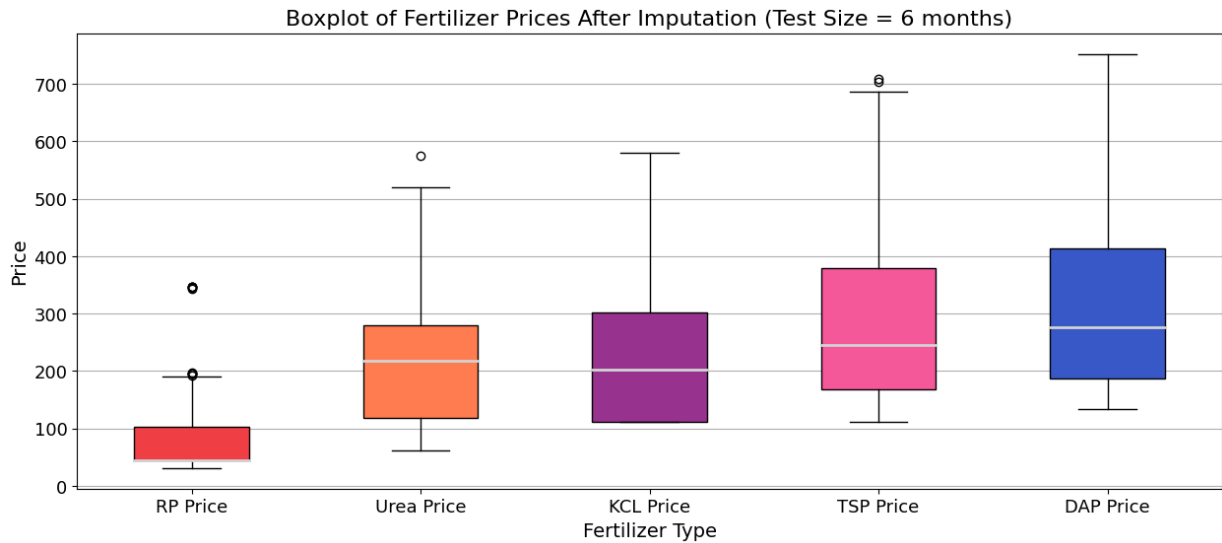
Mengikuti alur pada Kode Semu 3.2, nilai data pencilan yang terdeteksi akan diimputasi atau diganti dengan nilai median masing-masing kolom. Imputasi ini hanya diterapkan pada set pelatihan dan tidak pada set pengujian. Sebagai konsekuensi, jumlah data pencilan yang tersisa setelah proses imputasi dipengaruhi oleh ukuran set pelatihan dan set pengujian. Tabel 3.6 menunjukkan bahwa jumlah data pencilan yang tersisa setelah imputasi cenderung semakin sedikit seiring bertambahnya ukuran set pengujian atau rentang waktu prediksi.

Pupuk RP masih memiliki jumlah data pencilan terbanyak, bahkan setelah imputasi. Untuk prediksi 3 bulan, pupuk RP memiliki sisa data pencilan sebanyak 24 titik data. Sementara itu, untuk prediksi 6 bulan dan 12 bulan, pupuk RP memiliki sisa data pencilan sebanyak 12 titik data. Pupuk urea memiliki sisa data pencilan sebanyak 2 titik data untuk prediksi 3 bulan dan sebanyak 1 titik data untuk prediksi 6 bulan. Untuk prediksi 12 bulan, tidak ada data pencilan lagi yang tersisa untuk pupuk urea. Pupuk TSP memiliki sisa data pencilan sebanyak 2 titik data untuk prediksi 3 bulan dan 6 bulan, serta sebanyak 1 titik data untuk prediksi 12 bulan.

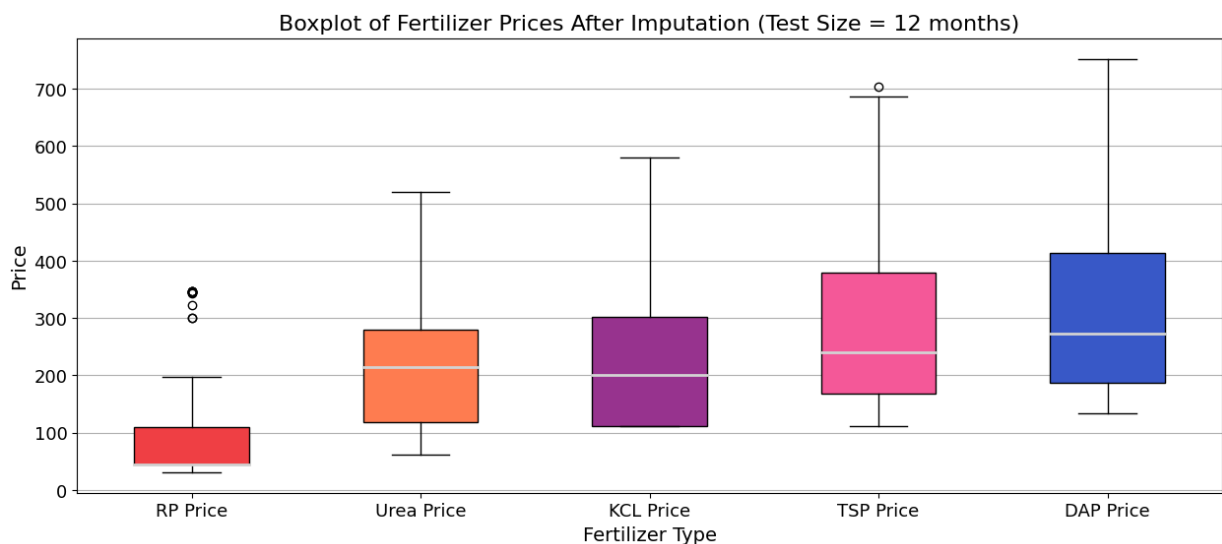


Gambar 3.10 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 3 Bulan)



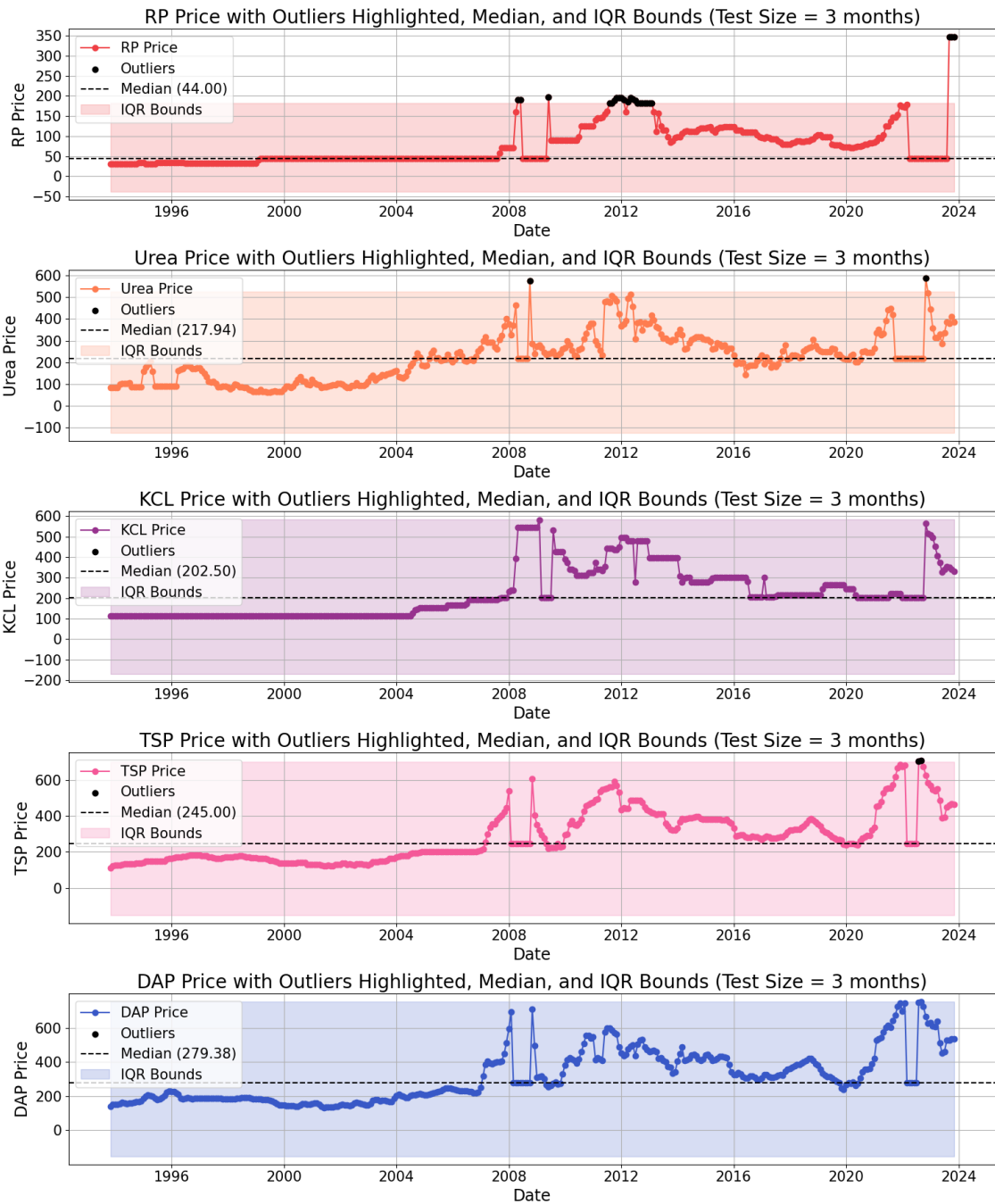


Gambar 3.11 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 6 Bulan)

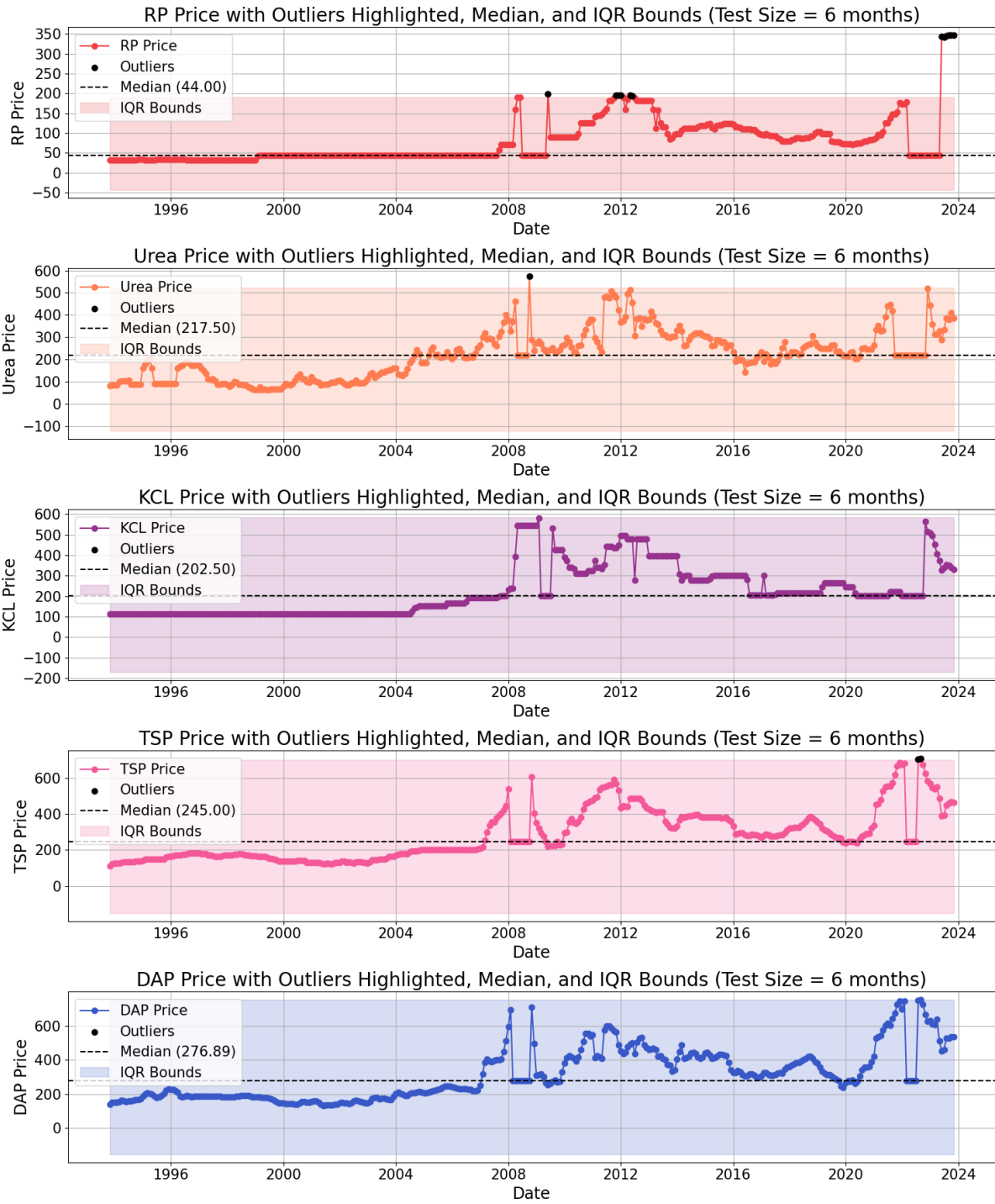


Gambar 3.12 Box Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 12 Bulan)

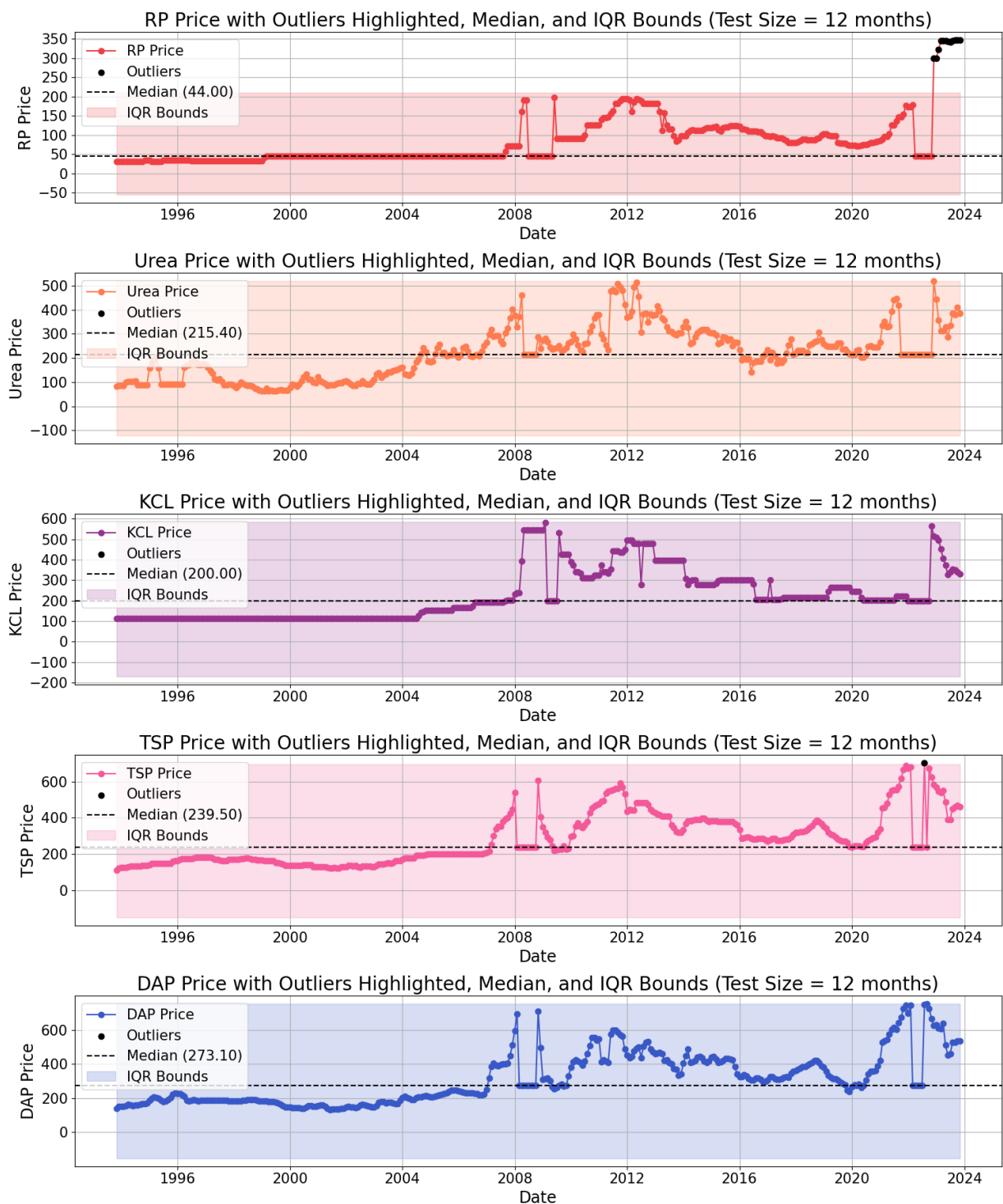
Gambar 3.10, Gambar 3.11, dan Gambar 3.12 menampilkan visualisasi Box Plot dari kelima kolom harga pupuk setelah proses imputasi. Letak garis median dalam Box Plot mengalami kenaikan pada sebagian besar jenis pupuk. Panjang box juga bertambah seiring dengan kenaikan letak garis median. Hal ini menunjukkan perbesaran IQR yang memperluas rentang nilai antara batas bawah dan batas atas. Semakin luas rentang nilai antara batas bawah dan batas atas, maka semakin sedikit data yang dianggap sebagai pencilan. Terlihat pula bahwa semua data pencilan yang tersisa melebihi batas atas, tanpa ada yang kurang dari batas bawah. Maka dari itu, dapat dikatakan bahwa semua data pencilan yang tersisa merupakan nilai-nilai ekstrem yang disebabkan oleh lonjakan harga.



Gambar 3.13 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 3 Bulan)



Gambar 3.14 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 6 Bulan)



Gambar 3.15 Line Plot Hasil Imputasi pada Kolom Harga Pupuk (Prediksi 12 Bulan)

Gambar 3.13, Gambar 3.14, dan Gambar 3.15 menampilkan Line Plot hasil imputasi dari lima jenis pupuk untuk tiga rentang prediksi. Garis hitam vertikal menunjukkan titik pemisahan antara set pelatihan dan set pengujian. Berdasarkan ketiga gambar, data pencilan yang tersisa setelah imputasi pada harga pupuk RP, urea, dan TSP terlihat sekitar tahun 2008-2013 dan 2023-2024. Khusus untuk pupuk RP, terdapat data pencilan yang tersisa pada set pengujian. Data pencilan tersebut tidak terpengaruh oleh proses imputasi karena proses ini

hanya diterapkan pada set pelatihan. Oleh sebab itu, jumlah data pencilan yang tersisa pada pupuk RP paling banyak.

### b. Aplikasi dan Inversi *Power Transform*

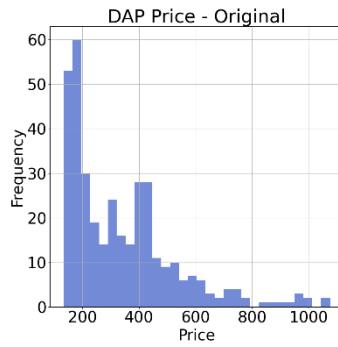
Kode Semu 3.3 menampilkan fungsi *apply\_transformation* untuk menerapkan *power transform* yang meliputi transformasi logaritma, akar kuadrat, dan Box-Cox. Semua transformasi diimplementasikan dengan fungsi *boxcox* dan *boxcox\_normmax* dari pustaka *scipy*. Jenis transformasi diatur dengan nilai parameter lambda ( $\lambda$ ). Pada baris ke-3 hingga ke-5, lambda = 0 digunakan untuk menerapkan transformasi logaritma. Kemudian, pada baris ke-6 hingga ke-8, lambda = 0.5 digunakan untuk menerapkan transformasi akar kuadrat. Jika lambda tidak ditetapkan, Box-Cox mencari nilai lambda yang membuat distribusi data paling normal dengan metode MLE (Maximum Likelihood Estimation) pada baris ke-9 hingga ke-10 atau Pearson R pada baris ke-11 hingga ke-13. Pada baris ke-17, fungsi *apply\_transformation* akan mengembalikan data hasil transformasi beserta nilai lambda yang dibutuhkan sebagai parameter fungsi *inverse\_transformation* pada Kode Semu 3.4 untuk mengembalikan data ke skala awal.

Fungsi Penerapan <i>Power Transform</i>
Input: Data dan jenis transformasi pilihan Output: Data hasil transformasi pilihan, nilai lambda
<pre> 1. IMPORT FUNCTIONS boxcox, boxcox_normmax FROM scipy.stats 2. FUNCTION apply_transformation(data, transform_type): 3.     IF transform_type IS 'Log': 4.         SET transformed_data TO BoxCox TRANSFORMATION OF data WITH lambda            = 0 5.         SET lambda_value TO 0 6.     ELSE IF transform_type IS 'Square Root': 7.         SET transformed_data TO BoxCox TRANSFORMATION OF data WITH lambda            = 0.5 8.         SET lambda_value TO 0.5 9.     ELSE IF transform_type IS 'BoxCox_MLE': 10.        SET transformed_data AND lambda_value TO BoxCox TRANSFORMATION            OF data WITH BEST FIT lambda 11.        ELSE IF transform_type IS 'BoxCox_PearsonR': 12.        CALCULATE lambda_value AS BoxCox NORMALIZATION MAXIMUM OF data            USING 'pearsonr' METHOD 13.        SET transformed_data TO BoxCox TRANSFORMATION OF data WITH            lambda_value 14.        ELSE: 15.        SET transformed_data TO data 16.        SET lambda_value TO None 17.        RETURN transformed_data, lambda_value </pre>

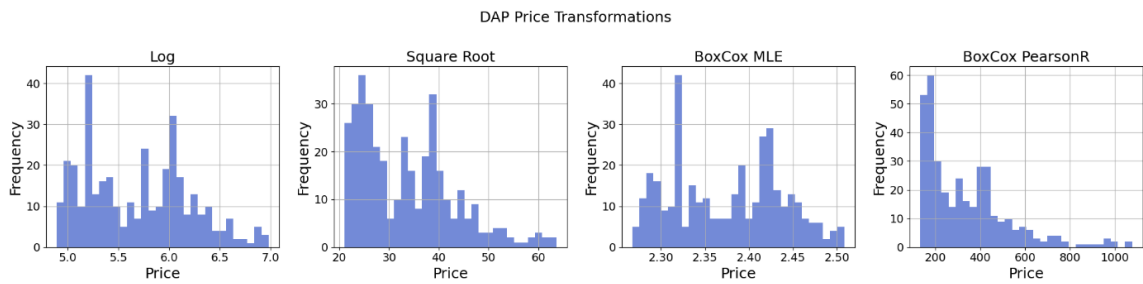
Kode Semu 3.3 Fungsi Penerapan *Power Transform*

Fungsi Inversi <i>Power Transform</i>
Input: Data hasil transformasi, nilai lambda Output: Data dalam skala awal
<pre> 1. IMPORT FUNCTION inv_boxcox FROM scipy.special 2. FUNCTION inverse_transformation(transformed_data, lambda_value): 3.     IF lambda_value IS NOT None: 4.         RETURN INVERSE BoxCox TRANSFORMATION OF transformed_data USING            lambda_value 5.     ELSE: 6.         RETURN transformed_data </pre>

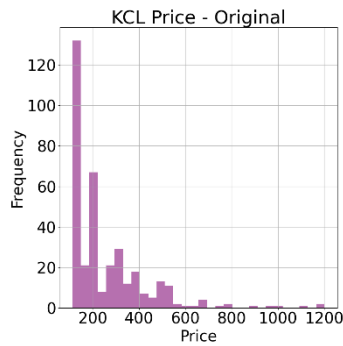
Kode Semu 3.4 Fungsi Inversi *Power Transform*



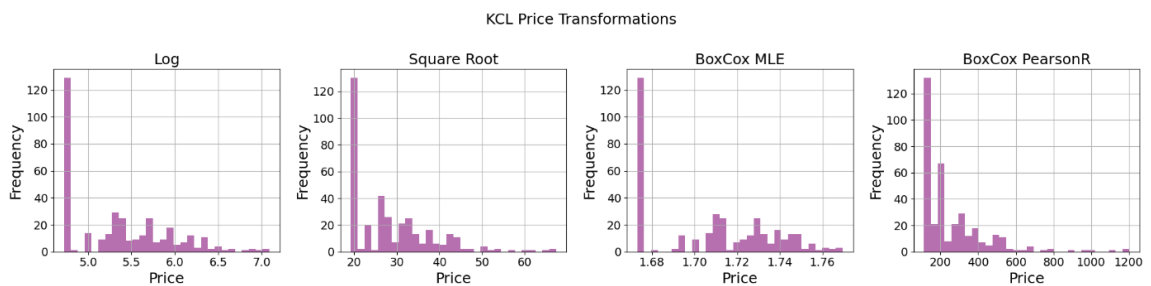
Gambar 3.16 Histogram Distribusi Data Harga DAP



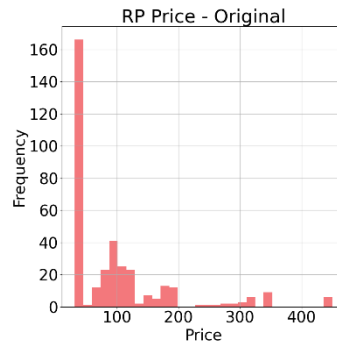
Gambar 3.17 Histogram Harga DAP Sesudah *Power Transform*



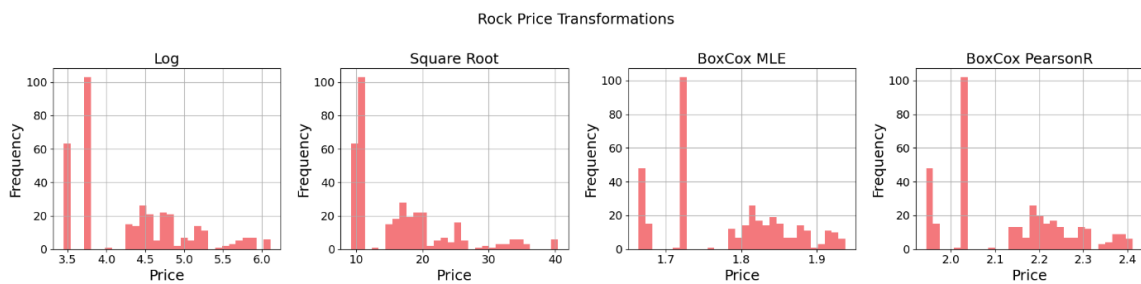
Gambar 3.18 Histogram Distribusi Data Harga KCL



Gambar 3.19 Histogram Harga KCL Sebelum dan Sesudah *Power Transform*



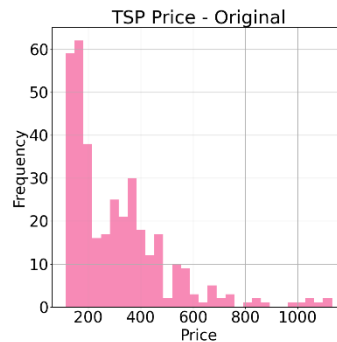
Gambar 3.20 Histogram Distribusi Data Harga RP



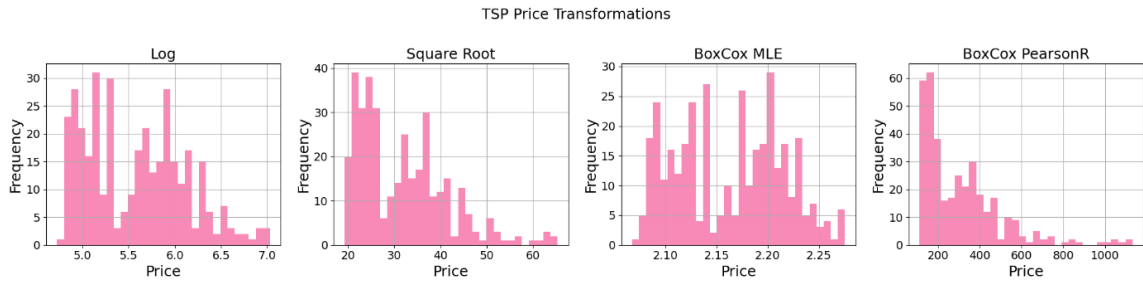
Gambar 3.21 Histogram Harga RP Sebelum dan Sesudah *Power Transform*

Gambar 3.16 dan Gambar 3.17 menampilkan histogram yang memvisualisasikan distribusi data harga DAP sebelum dan sesudah power transform. Data harga DAP sebelum transformasi menunjukkan distribusi miring ke kanan, menandakan mayoritas harga DAP terkumpul di kisaran harga rendah, sekitar \$100-\$400. Namun, terdapat beberapa harga tinggi yang secara signifikan menyimpang dari norma, mencapai kisaran \$1000. Untuk membuat distribusi data yang menjadi lebih mendekati distribusi normal, dilakukan power transform. Di antara keempat power transform, transformasi logaritma dan Box-Cox MLE membuat distribusi data harga DAP terlihat paling mendekati distribusi normal. Sebaliknya, transformasi akar kuadrat dan Box-Cox PearsonR tidak banyak berpengaruh pada kemiringan distribusi data harga DAP.

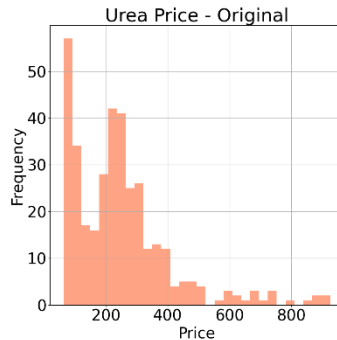
Gambar 3.18 dan Gambar 3.19 menampilkan histogram yang memvisualisasikan distribusi data harga KCL sebelum dan sesudah power transform. Data harga KCL sebelum transformasi menunjukkan distribusi yang miring ke kanan dengan mayoritas harga terkumpul di kisaran \$100-\$200. Harga tertinggi mencapai sekitar \$1200. Dari keempat power transform yang digunakan, transformasi logaritma dan Box-Cox MLE terlihat paling efektif dalam mengurangi kemiringan distribusi data harga KCL.



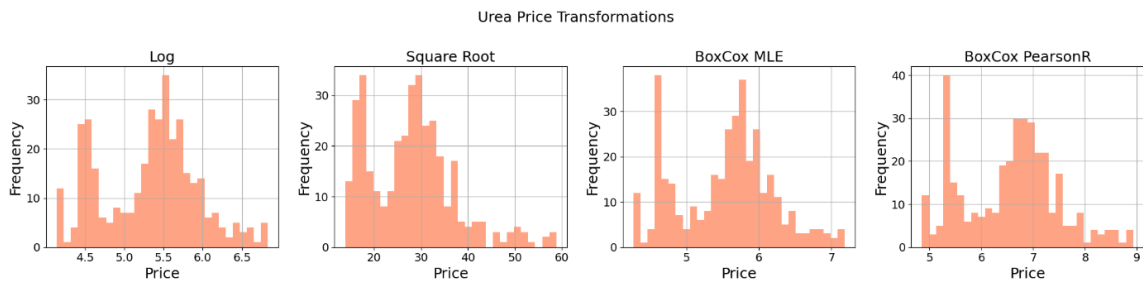
Gambar 3.22 Histogram Distribusi Data Harga TSP



Gambar 3.23 Histogram Harga TSP Sebelum dan Sesudah *Power Transform*



Gambar 3.24 Histogram Harga Urea Sebelum dan Sesudah *Power Transform*



Gambar 3.25 Histogram Harga Urea Sebelum dan Sesudah *Power Transform*

Gambar 3.20 dan Gambar 3.21 menyajikan histogram yang menggambarkan distribusi data harga RP sebelum dan setelah power transform. Data harga RP sebelum transformasi menunjukkan distribusi yang miring ke kanan dengan mayoritas harga terkonsentrasi di kisaran \$100. Terdapat sebagian kecil data harga yang melampaui \$400. Hasil dari transformasi logaritma, transformasi Box-Cox MLE, dan transformasi Box-Cox PearsonR membuat distribusi terlihat paling mendekati normal.

Gambar 3.22 dan Gambar 3.23 menunjukkan histogram yang menggambarkan distribusi data harga TSP sebelum dan setelah power transform. Sebelum transformasi, data harga TSP memperlihatkan distribusi miring ke kanan dengan sebagian besar harga berada di kisaran \$100-\$400, dan sebagian kecil harga mencapai kisaran \$1000. Transformasi logaritma dan Box-Cox MLE membuat distribusi data harga TSP terlihat paling mendekati normal. Di sisi lain, transformasi Box-Cox Pearson R tampak tidak mengubah distribusi data harga TSP sama sekali. Gambar 3.24 dan Gambar 3.25 menampilkan histogram yang memvisualisasikan distribusi data harga urea sebelum dan setelah transformasi. Data harga urea sebelum transformasi menunjukkan distribusi miring ke kanan dengan kebanyakan harga terkumpul di kisaran \$100-\$300 dan beberapa data harga terletak di kisaran tinggi \$800-\$900. Hasil transformasi logaritma, transformasi Box-Cox MLE, dan transformasi Box-Cox PearsonR membuat distribusi data harga urea terlihat paling mendekati normal. Secara keseluruhan, sebelum proses transformasi, data harga pupuk menunjukkan distribusi yang miring kanan. Histogram-histogram pada Gambar 3.16 hingga Gambar 3.25 menampilkan bagaimana power



transform dapat mengurangi kemiringan distribusi tersebut dengan kekuatan pengaruh yang bervariasi.

**c. Optimasi Parameter**

Optimasi parameter dengan pustaka Optuna membutuhkan fungsi *objective*, objek *study*, dan pemanggilan method *optimize* milik objek *study* tersebut. Fungsi *objective* untuk metode *machine learning* dan metode statistik perlu dipisah karena ruang pencarian parameter keduanya berbeda. Kode Semu 3.4 menampilkan fungsi *objective* untuk metode machine learning. Baris ke-8 hingga ke-12 membangun ruang pencarian parameter dengan method *suggest*. Parameter-parameter tersebut, antara lain, *scaler*, *n\_lags*, *n\_differentiation*, *power\_transform*, dan, *is\_imputed*. Keterangan dan rentang pilihan nilai untuk tiap parameter dicantumkan pada Tabel 3.3.

Tabel 3.7 Parameter-parameter yang Dipertimbangkan untuk Metode *Machine Learning*

Parameter	Rentang Pilihan Nilai	Keterangan
scaler	StandardScaler, RobustScaler, MinMaxScaler	Jenis scaler yang diterapkan pada data untuk mengubah skala data.
n_lags	1 sampai 12	Jumlah variabel lag atau data masa lalu yang digunakan untuk memprediksi nilai masa depan.
n_differentiation	None, 1, 2	Jumlah orde <i>differencing</i> yang diterapkan pada data untuk membuat data stasioner.
power_transform	None, Log, Square Root, BoxCox_MLE, BoxCox_PearsonR	Jenis <i>power transform</i> yang diterapkan pada data. Meliputi transformasi BoxCox, logaritma, dan akar kuadrat.
is_imputed	True, False	Menentukan apakah data-data pencilan diganti dengan median atau tidak.

Penerapan parameter pilihan pada proses pelatihan model *machine learning* dimulai pada baris ke-14 yang memeriksa nilai parameter *is\_imputed*. Jika *is\_imputed* bernilai True, maka akan dilakukan deteksi dan imputasi data pencilan. Kemudian, baris ke-33 memeriksa nilai parameter *power\_transform* dan menerapkan jenis transformasi yang sesuai pilihan pada data. Setelah data melalui transformasi, baris ke-35 hingga ke-40 mulai membuat model prediksi *machine learning* yang diimplementasikan dengan objek *ForecasterAutoreg* dari pustaka *Skforecast*. Objek *ForecasterAutoreg* berisi model regresi yang disertai parameter hasil method *suggest*, khususnya *transformer\_y*, *differentiation*, dan *lags*.

Parameter *transformer\_y* menentukan jenis *scaling*, parameter *differentiation* menentukan jumlah orde *differencing*, sedangkan parameter *lags* menentukan jumlah variabel *lag* yang digunakan untuk prediksi. Model lalu dilatih dengan metode fit pada baris ke-42. Selanjutnya, pada baris ke-43, model melakukan prediksi dengan rentang waktu sebanyak *n\_test*. Sebelum dievaluasi, transformasi data pada hasil prediksi perlu dibalikkan terlebih dahulu pada baris ke-45. Proses evaluasi dilakukan pada baris ke-50 dengan fungsi *evaluate\_forecast*. Nilai MAPE hasil evaluasi akhirnya digunakan sebagai *score* fungsi *objective* yang dikembalikan pada baris ke-60.

Selain itu, dalam fungsi *objective* Kode Semu 3.4, terdapat beberapa kondisi pemangkasan atau penghentian awal *trial*. Pertama, pada baris ke-17 hingga ke-21, *trial* akan dihentikan jika kombinasi parameter sudah digunakan sebelumnya. Kedua, pada baris ke-24 dan ke-25, *trial* akan dihentikan jika data pelatihan mengandung nilai NaN atau Inf. Ketiga, pada baris ke-53 dan ke-54, *trial* akan dihentikan jika data hasil prediksi mengandung nilai NaN atau Inf. Keempat, pada baris ke-62 dan ke-64, *trial* akan dihentikan jika mengalami eror dalam proses optimasi. Terakhir, *trial* akan dihentikan jika method *should\_prune* mengembalikan nilai

True yang berarti trial dianggap tidak menjanjikan. Syarat *trial* dianggap tidak menjanjikan akan ditentukan oleh objek *Pruner* yang digunakan dalam method *optimize*.

Fungsi Objective Optimasi Parameter untuk Metode <i>Machine Learning</i>
Input: Trial, jenis model, data pelatihan, data pengujian, ukuran data pengujian, kolom tanggal, kolom target Output: Skor MAPE untuk suatu trial
<pre> 1. <b>FUNCTION</b> objective(trial, regressor, train_data, test_data, n_test,    date_col, target_col): 2.     SET scaler_mapping TO {'StandardScaler': StandardScaler(),    'RobustScaler': RobustScaler(), 'MinMaxScaler': MinMaxScaler(), 'None':    None} 3.     SET scaler TO scaler_mapping[scaler_choice] 4.     <b>FOR</b> t IN trial.study.trials: 5.         <b>IF</b> t.state IS NOT optuna.trial.TrialState.COMPLETE: CONTINUE 6.         <b>IF</b> t.params IS trial.params: RAISE TrialPruned('Duplicate    parameter set') 7.     <b>TRY</b>: 8.         <b>IF</b> train_data CONTAINS NaN OR inf: RAISE TrialPruned() 9.         <b>IF</b> impute_choice: SET imputed_train_data TO    detect_and_impute_outlier(train_data, target_col) 10.        <b>ELSE</b>: SET imputed_train_data TO train_data 11.        SET transformed_train_data TO DEEP COPY OF imputed_train_data 12.        SET transformed_train_data[target_col], lambda_value TO    apply_transformation(imputed_train_data[target_col],    power_transform_choice) 13.        INITIALIZE forecaster WITH    ForecasterAutoreg(regressor=regressor, transformer_y=scaler,    differentiation=n_differentiation, lags=n_lags) 14.        CALL forecaster.fit(y=transformed_train_data[target_col]) 15.        SET predictions TO forecaster.predict(n_test) 16.        CREATE predictions_df WITH {date_col: predictions.index,    target_col: predictions.values} 17.        SET inversed_predictions_df TO    inverse_transformation(predictions_df[target_col], lambda_value) 18.        SET actual TO test_data[target_col].values 19.        SET pred TO inversed_predictions_df 20.        CREATE comparison_df WITH {'Actual': actual, 'Predicted': pred} 21.        SET evaluation_results TO    evaluate_forecast(comparison_df['Actual'], comparison_df['Predicted']) 22.        SET score TO evaluation_results['MAPE'] 23.        <b>IF</b> predictions_df OR inversed_predictions_df CONTAINS NaN OR    inf: RAISE TrialPruned() 24.        CALL trial.report(score, trial.number) 25.        <b>IF</b> trial.should_prune(): RAISE TrialPruned() 26.        <b>RETURN</b> score 27.    <b>EXCEPT</b> Exception AS e: 28.        PRINT f"Trial failed due to error: {e}" 29.        RAISE TrialPruned()           </pre>

Kode Semu 3.5 Fungsi Objective Optimasi Parameter untuk Metode *Machine Learning*

Tabel 3.8 Parameter-parameter yang Dipertimbangkan untuk Metode Statistik

Parameter	Rentang Pilihan Nilai	Keterangan
power_transform	None, Log, Square Root, BoxCox_MLE, BoxCox_PearsonR	Jenis <i>power transform</i> yang diterapkan pada data. Meliputi transformasi BoxCox, logaritma, dan akar kuadrat.
is_imputed	True, False	Menentukan apakah data-data pencilan diganti dengan median atau tidak.
Seasonal_length	3, 4, 6, 12	Biasa disebut S. Menentukan jumlah bulan dalam satu periode musim untuk SARIMA.

Fungsi Objective Optimasi Parameter untuk Metode Statistik

Input: Trial, jenis model, data pelatihan, data pengujian, ukuran data pengujian, kolom tanggal, kolom target

Output: Skor MAPE untuk suatu trial

```

1. FUNCTION objective_autoARIMA(trial, train_data, test_data, n_test,
   date_col, target_col, regressor):
2.   FOR t IN trial.study.trials:
3.     IF t.state IS NOT optuna.trial.TrialState.COMPLETE: CONTINUE
4.     IF t.params IS trial.params: RAISE TrialPruned('Duplicate
   parameter set')
5.     TRY:
6.       IF train_data CONTAINS NaN OR inf: RAISE TrialPruned()
7.       IF impute_choice: SET imputed_train_data TO
   detect_and_impute_outlier(train_data, target_col)
8.       ELSE: SET imputed_train_data TO train_data
9.       SET transformed_train_data TO DEEP COPY OF imputed_train_data
10.      SET transformed_train_data[target_col], lambda_value TO
   apply_transformation(imputed_train_data[target_col],
   power_transform_choice)
11.      INITIALIZE auto_arima WITH
   StatsForecast(models=[AutoARIMA(max_p=12, max_d=2, max_q=12, max_P=4,
   max_D=2, max_Q=4, seasonal=True, season_length=seasonal_length_choice)],
   n_jobs=-1, freq='MS')
12.      CALL auto_arima.fit(transformed_train_data)
13.      SET predictions TO auto_arima.predict(h=n_test)
14.      CREATE predictions_df WITH {date_col: predictions['ds'],
   target_col: predictions[regressor]}
15.      SET inversed_predictions_df TO
   inverse_transformation(predictions_df[target_col], lambda_value)
16.      SET actual TO test_data[target_col].values
17.      SET pred TO inversed_predictions_df.values
18.      CREATE comparison_df WITH {'Actual': actual, 'Predicted': pred}
19.      SET comparison_df.index TO test_data['ds']
20.      SET evaluation_results TO
   evaluate_forecast(comparison_df['Actual'], comparison_df['Predicted'])
21.      SET score TO evaluation_results['MAPE']
22.      IF predictions_df OR inversed_predictions_df CONTAINS NaN OR
   inf: RAISE TrialPruned()
23.      CALL trial.report(score, trial.number)
24.      IF trial.should_prune(): RAISE TrialPruned()
25.      RETURN score
26.   EXCEPT Exception AS e:
27.     PRINT f"Trial failed due to error: {e}"
28.     RAISE TrialPruned()

```

Kode Semu 3.6 Fungsi Objective Optimasi Parameter untuk Metode Statistik

Sementara itu, Kode Semu 3.5 menunjukkan fungsi *objective* untuk metode statistik. Kode Semu 3.5 mempunyai banyak kesamaan dengan Kode Semu 3.4. Perbedaan utamanya terletak pada jenis model yang digunakan dan parameter yang dioptimasi. Mulai dari baris ke-7 hingga ke-9, fungsi ini membangun ruang pencarian parameter model statistik dengan method *suggest*. Parameter yang dimaksud meliputi *power\_transform*, *is\_imputed*, dan *Seasonal\_length*. Keterangan dan rentang pilihan nilai untuk tiap parameter dicantumkan pada Tabel 3.4.

Selain parameter, perbedaan Kode Semu 3.5 dengan Kode Semu 3.4 terlihat pada jenis model yang digunakan. Pada baris ke-29 hingga ke-33, Kode Semu 3.5 menginisialisasi model AutoARIMA dengan parameter *max\_p*, *max\_d*, *max\_q*, *max\_P*, *max\_D*, *max\_Q*, *Seasonal*, dan *Seasonal\_length*. Keenam parameter pertama menetapkan nilai maksimum parameter (*p*, *d*, *q*) dan (*P*, *D*, *Q*) selama pencarian orde optimal dalam model AutoARIMA. Model AutoARIMA mencari orde parameter yang optimal melalui pengujian statistik. Uji stasioneritas dengan ADF menentukan orde parameter (*d*), sedangkan uji stasioneritas dengan OCSB menentukan orde parameter (*D*). Parameter (*p*, *q*) dan (*P*, *Q*) ditentukan melalui minimalisasi AIC.

Parameter *Seasonal* yang bernilai True memungkinkan model AutoARIMA untuk mempertimbangkan faktor musiman, sehingga mengimplementasikan algoritma SARIMA. Jumlah bulan dalam satu periode musim atau *S* diatur oleh parameter *Seasonal\_length*. Parameter *Seasonal\_length* diambil dari pilihan method *suggest* milik *trial* Optuna. Setelah inisialisasi model dengan parameter, pelatihan model AutoARIMA dilakukan pada baris ke-35 diikuti dengan prediksi pada baris ke-37 dan evaluasi pada baris ke-46. Detail implementasi pelatihan, prediksi, dan evaluasi dalam Kode Sumber 2.5 sama dengan Kode Sumber 2.4.

Kode Semu 3.6 menampilkan fungsi *tune\_forecaster* yang akan membuat objek *study* dan memanggil method *optimize*-nya. Objek *study* dibuat pada baris ke-8 dengan *GridSampler* dan *SuccessiveHalvingPruner*. Penggunaan *GridSampler* akan memastikan bahwa optimasi akan mencoba semua kombinasi parameter yang didefinisikan dalam ruang pencarian. Sementara itu, *SuccessiveHalvingPruner* akan secara progresif memangkas *trial* yang berkinerja buruk selama proses optimasi. Selain *sampler* dan *pruner*, objek *study* mempunyai parameter *direction* bernilai *minimize* yang berarti proses optimasi akan berusaha meminimalkan nilai fungsi *objective*. Dalam konteks penelitian ini, nilai fungsi *objective* berupa nilai metrik MAPE. Pemanggilan method *optimize* milik objek *study* dilakukan pada baris ke-10. Hasil parameter terbaik akan disimpan dalam variabel *best\_params* pada baris ke-12 dan hasil nilai metrik terendah akan disimpan dalam variabel *lowest\_metric* pada baris ke-13. Kedua variabel akan dikembalikan bersama objek *study* pada baris ke-17. Kode Sumber 3.7 memvisualisasikan hasil optimasi parameter. Baris ke-8 hingga ke-10 menghasilkan visualisasi tingkat kepentingan parameter, seperti yang ditampilkan dalam Gambar 3.17. Kemudian, baris ke-17 hingga ke-19 menampilkan riwayat optimasi parameter dalam bentuk nilai fungsi *objective* seiring bertambahnya *trial*. Gambar 3.18 menunjukkan contoh visualisasi riwayat optimasi tersebut.

```

Fungsi Manajemen Proses Optimasi Parameter
Input: Data pelatihan, data pengujian, ukuran data pengujian, kolom tanggal,
kolom target, jenis model, ruang pencarian, jumlah trial
Output: Hasil semua trial, parameter terbaik, metrik kesalahan terendah
1. IMPORT optuna
2. IMPORT GridSampler FROM optuna.samplers
3. IMPORT SuccessiveHalvingPruner FROM optuna.pruners
4. FUNCTION tune_forecaster(train_data, test_data, n_test, date_col,
target_col, regressor, search_space, n_trials=None):
5.     CREATE sampler WITH GridSampler(search_space=search_space)
6.     CREATE pruner WITH SuccessiveHalvingPruner()
7.     CREATE study WITH optuna.create_study(direction='minimize',
sampler=sampler, pruner=pruner)
8.     CALL study.optimize(lambda trial: objective (trial, train_data,
test_data, n_test, date_col, target_col, regressor), n_trials=n_trials)
9.     SET best_params TO study.best_params
10.    SET lowest_metric TO study.best_value
11.    PRINT "Best parameters: {best_params}, Best MAPE: {lowest_metric}"
12.    RETURN study, best_params, lowest_metric

```

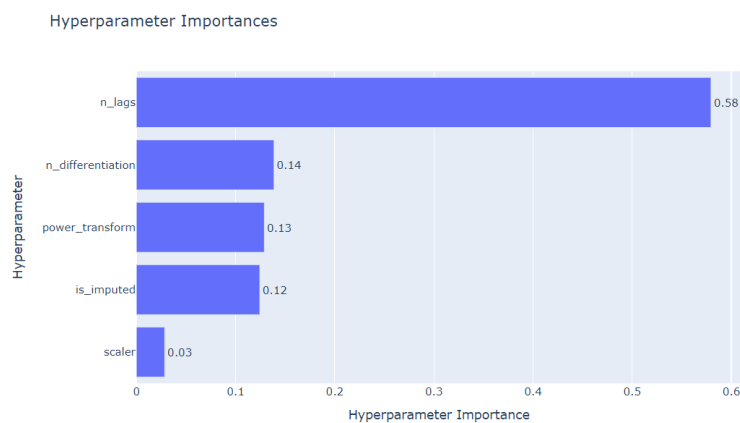
Kode Semu 3.7 Fungsi Manajemen Proses Optimasi Parameter

```

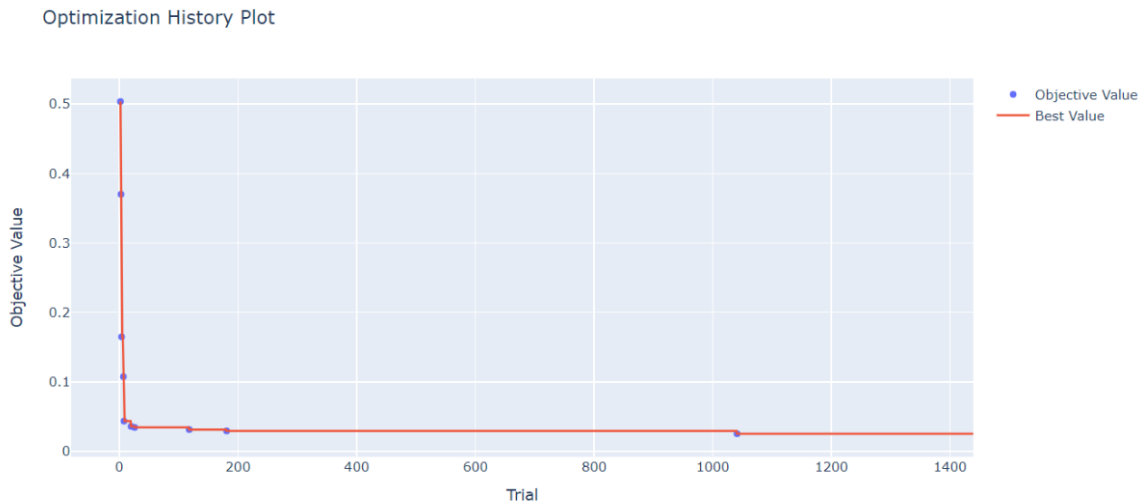
Analisis Hasil Optimasi Parameter
Input: Hasil semua trial
Output: Visualisasi tingkat kepentingan parameter dan riwayat optimasi
1. IMPORT optuna
2. IMPORT pandas, matplotlib.pyplot
3. RUN tune_forecaster_with_optuna AND STORE RESULTS IN study, best_params,
lowest_metric
4. DISPLAY Hyperparameter Importances:
5.     CREATE fig WITH optuna.visualization.plot_param_importances(study)
6.     CALL fig.show()
7. CALCULATE importances WITH
optuna.importance.get_param_importances(study)
8. CREATE df_importances AS DataFrame WITH [dict(importances)], index=[0]
9. SET df_importances['data'] TO target_col
10. SET df_importances['horizon'] TO n_test
11. DISPLAY Optimization History:
12.    CREATE fig WITH
optuna.visualization.plot_optimization_history(study)
13.    CALL fig.show()
14. GET study trials dataframe WITH study.trials_dataframe()

```

Kode Semu 3.8 Analisis Hasil Optimasi Parameter



Gambar 3.26 Contoh Visualisasi Tingkat Kepentingan Parameter untuk XGBoost



Gambar 3.27 Contoh Visualisasi Riwayat Optimasi Parameter untuk XGBoost

Gambar 3.26 menunjukkan contoh visualisasi tingkat kepentingan parameter yang disediakan oleh pustaka Optuna. Dari gambar tersebut, terlihat seberapa besar kontribusi tiap parameter dalam mengurangi kesalahan model XGBoost untuk memprediksi harga DAP dengan rentang waktu 3 bulan. Parameter `n_lags` memiliki tingkat kepentingan tertinggi yaitu 0.58. Parameter `n_differentiation`, `power_transform`, dan `is_imputed` menunjukkan tingkat kepentingan yang hampir sama. Terakhir, parameter `scaler` memiliki tingkat kepentingan paling rendah yaitu senilai 0.03. Gambar 3.27 memperlihatkan visualisasi proses optimasi parameter XGBoost untuk memprediksi harga DAP tiga bulan ke depan. *Objective Value* mengacu pada nilai MAPE dari berbagai uji coba kombinasi parameter. Selama 1400 uji coba, nilai MAPE terlihat terus menurun, dari sekitar 50% hingga mencapai kurang dari 10%.

### 3.3.3 Evaluasi

Kode Semu 3.9 menampilkan fungsi `evaluate_forecast` untuk evaluasi hasil prediksi. Terlihat pada baris ke-3 bahwa fungsi ini menerima nilai sebenarnya (`y_true`) dan nilai hasil prediksi (`y_pred`) dari kolom target sebagai parameter. Hasil prediksi dievaluasi dengan tiga metrik, yakni MAPE pada baris ke-4, RMSE pada baris ke-5, dan MAE pada baris ke-6. Fungsi mengembalikan pandas Series berisi nilai ketiga metrik evaluasi pada baris ke-8. Di antara ketiga metrik evaluasi, MAPE berperan sebagai metrik utama dalam menentukan *trial* dan kombinasi parameter terbaik.

Fungsi Evaluasi Hasil Prediksi
Input: Nilai sebenarnya dan hasil prediksi
Output: Series berisi MAPE, RMSE, dan MAE
<pre> 1.  IMPORT numpy, pandas 2.  IMPORT mean_absolute_percentage_error, mean_squared_error,     mean_absolute_error FROM sklearn.metrics 3.  <b>FUNCTION</b> evaluate_forecast(y_true, y_pred): 4.      CALCULATE MAPE WITH mean_absolute_percentage_error(y_true, y_pred) 5.      CALCULATE RMSE WITH np.sqrt(mean_squared_error(y_true, y_pred)) 6.      CALCULATE MAE WITH mean_absolute_error(y_true, y_pred) 7.      CREATE results DICTIONARY WITH 'MAPE', 'RMSE', 'MAE' 8.      <b>RETURN</b> pd.Series(results) </pre>

Kode Semu 3.9 Fungsi Evaluasi Hasil Prediksi

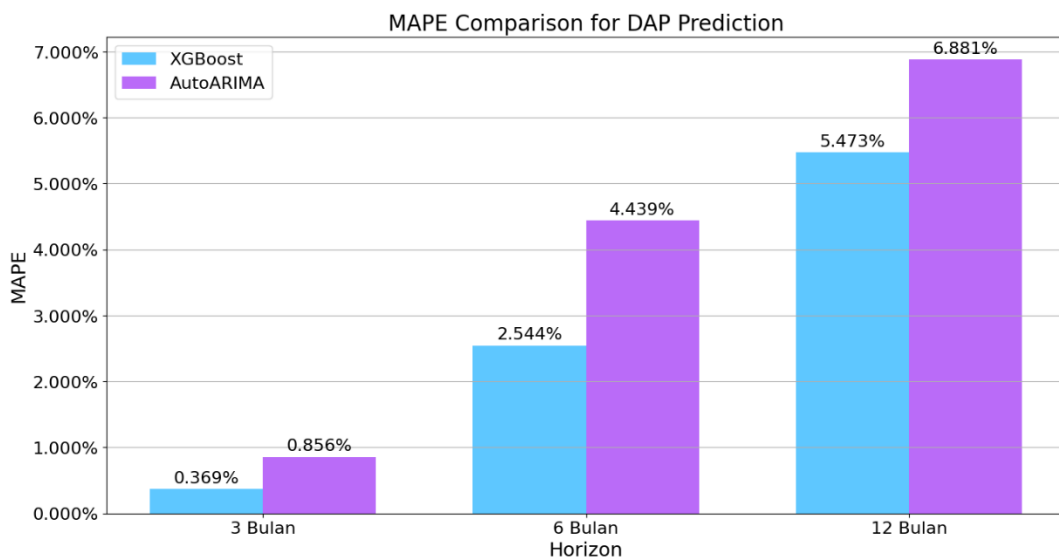
## BAB 4 HASIL DAN PEMBAHASAN

### 4.1 Uji Coba 1 untuk Membandingkan Performa Model Prediksi Harga DAP

Uji coba pertama dilakukan untuk mengevaluasi performa model XGBoost dan AutoARIMA dalam memprediksi harga pasar pupuk DAP. Performa diukur dengan menggunakan metrik evaluasi MAPE, MAE, dan RMSE.

Tabel 4.1 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar DAP

Metrik Evaluasi	Model	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
MAPE	XGBoost	0.369%	2.544%	5.473%
	AutoARIMA	0.856%	4.439%	6.881%
MAE	XGBoost	1.969	12.368	29.877
	AutoARIMA	4.577	20.657	37.857
RMSE	XGBoost	2.195	16.225	40.092
	AutoARIMA	5.293	31.252	49.764



Gambar 4.1 Bar Plot Perbandingan MAPE Hasil Prediksi Harga Pasar DAP

Tabel 4.1 dan Gambar 4.1 menunjukkan perbandingan performa model XGBoost dan AutoARIMA dalam melakukan prediksi harga pasar DAP (Diammonium Phosphate) dengan rentang waktu 3 bulan, 6 bulan, dan 12 bulan. Terdapat tiga metrik evaluasi yang digunakan, yaitu Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), dan Root Mean Square Error (RMSE). MAPE berperan sebagai metrik utama. Pada rentang prediksi 3 bulan, XGBoost menunjukkan kinerja yang lebih baik dengan MAPE sebesar 0.369%, sedangkan AutoARIMA memiliki MAPE sebesar 0.856%. Kinerja XGBoost tetap unggul pada rentang prediksi 6 bulan dan 12 bulan dengan MAPE masing-masing 2.544% dan 5.473%, dibandingkan dengan AutoARIMA yang memiliki MAPE sebesar 4.439% dan 6.881%.

Kedua model menunjukkan peningkatan MAPE seiring dengan meningkatnya rentang waktu prediksi. Namun, laju peningkatan tersebut berbeda antara kedua model. Pada model XGBoost, MAPE meningkat dari 0.369% untuk prediksi 3 bulan menjadi 2.544% untuk prediksi 6 bulan, menunjukkan peningkatan sebesar 2.175%. Kemudian, MAPE kembali meningkat menjadi 5.473% untuk prediksi 12 bulan, dengan peningkatan sebesar 2.929% dari prediksi 6 bulan. Secara keseluruhan, dari prediksi 3 bulan ke 12 bulan, terdapat peningkatan

total sebesar 5.104%. Sementara itu, pada model AutoARIMA, MAPE awalnya berada di angka 0.856% untuk prediksi 3 bulan. Nilai ini meningkat menjadi 4.439% untuk prediksi 6 bulan, dengan peningkatan sebesar 3.583%. Selanjutnya, MAPE untuk prediksi 12 bulan mencapai 6.881%, yang berarti ada peningkatan sebesar 2.442% dari prediksi 6 bulan. Secara keseluruhan, dari prediksi 3 bulan ke 12 bulan, terdapat peningkatan total sebesar 6.025%. Dari hasil di atas, terlihat bahwa model AutoARIMA mengalami kenaikan MAPE yang lebih besar dibandingkan model XGBoost seiring dengan meningkatnya rentang waktu prediksi

Tabel 4.2 Analisis Persentase Kepentingan Parameter untuk Prediksi Harga Pasar DAP

Metode	Parameter	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
XGBoost	scaler	16.312%	0.833%	10.843%
	n lags	11.774%	12.000%	16.819%
	n differentiation	39.029%	68.113%	53.215%
	power transform	28.390%	16.156%	13.125%
	is imputed	4.496%	2.898%	5.998%
AutoARIMA	Seasonal length	77.429%	28.229%	23.498%
	power transform	22.571%	16.793%	19.571%
	is imputed	0.000%	54.978%	56.930%

Tabel 4.2 memaparkan analisis kepentingan parameter dalam proses optimasi model XGBoost dan AutoARIMA untuk prediksi harga pasar DAP. Untuk model XGBoost, parameter *n\_differentiation* secara konsisten menunjukkan tingkat kepentingan tertinggi di semua rentang prediksi, dengan nilai 39,029% untuk prediksi 3 bulan, meningkat signifikan menjadi 68,113% untuk prediksi 6 bulan, dan sedikit menurun menjadi 53,215% untuk prediksi 12 bulan. Sementara itu, untuk model AutoARIMA, parameter *Seasonal\_length* memiliki tingkat kepentingan yang sangat tinggi (77,429%) untuk prediksi 3 bulan. Namun, tingkat kepentingannya menurun drastis untuk prediksi 6 dan 12 bulan menjadi 28.229% dan 23.498% berturut-turut. Sebaliknya, parameter *is\_imputed* menunjukkan peningkatan kepentingan yang signifikan, dari 0,000% untuk prediksi 3 bulan menjadi 54.978% untuk prediksi 6 bulan dan 56.930% untuk prediksi 12 bulan.

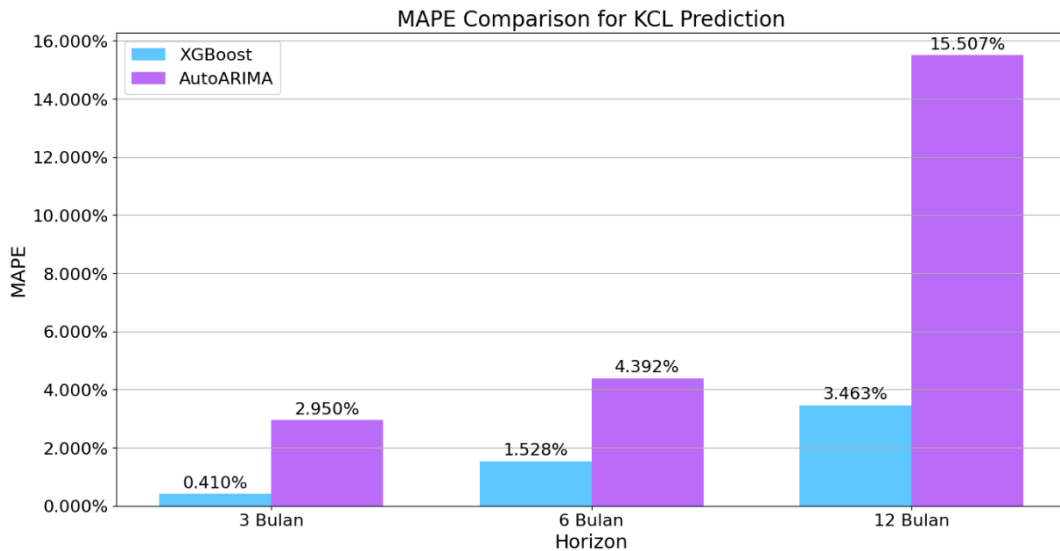
#### 4.2 Uji Coba 2 untuk Membandingkan Performa Model Prediksi Harga KCL

Uji coba kedua dilakukan untuk mengevaluasi performa model XGBoost dan AutoARIMA dalam memprediksi harga pasar pupuk KCL. Performa diukur dengan menggunakan metrik evaluasi MAPE, MAE, dan RMSE.

Tabel 4.3 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar KCL

Metrik Evaluasi	Model	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
MAPE	XGBoost	0.410%	1.528%	3.463%
	AutoARIMA	2.950%	4.392%	15.507%
MAE	XGBoost	1.412	5.154	13.713
	AutoARIMA	9.896	14.643	60.916
RMSE	XGBoost	1.462	5.944	17.718
	AutoARIMA	12.580	19.576	69.424





Gambar 4.2 Bar Plot Perbandingan MAPE Hasil Prediksi KCL

Tabel 4.3 dan Gambar 4.2 memperlihatkan perbandingan kinerja model XGBoost dan AutoARIMA dalam memprediksi harga pasar KCL (Kalium Klorida). Digunakan rentang waktu prediksi dan metrik evaluasi yang sama dengan uji coba sebelumnya. Pada rentang prediksi 3 bulan, model XGBoost menunjukkan performa yang lebih baik dengan MAPE sebesar 0.410%, sedangkan AutoARIMA memiliki MAPE sebesar 2.950%. Kinerja XGBoost tetap unggul pada prediksi 6 bulan dan 12 bulan dengan MAPE masing-masing sebesar 1.528% dan 3.463%, dibandingkan AutoARIMA yang memiliki MAPE sebesar 4.392% dan 15.507%.

Pada model XGBoost, MAPE meningkat dari 0.410% untuk prediksi 3 bulan menjadi 3.463% untuk prediksi 12 bulan, dengan peningkatan total sebesar 3.053%. Di sisi lain, pada model AutoARIMA, MAPE awalnya berada di angka 2.950% untuk prediksi 3 bulan. Nilai ini meningkat menjadi 15.507% untuk prediksi 12 bulan, dengan peningkatan total sebesar 12.557%. Baik model XGBoost, maupun AutoARIMA, mengalami kenaikan nilai metrik eror seiring meningkatnya rentang waktu prediksi. Namun, AutoARIMA kembali menunjukkan kenaikan metrik kesalahan yang lebih drastis.

Tabel 4. 4 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar KCL

Metode	Parameter	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
XGBoost	scaler	23.751%	46.507%	2.327%
	n_lags	17.811%	13.391%	3.276%
	n_differentiation	32.565%	1.740%	75.795%
	power_transform	18.225%	36.620%	17.408%
	is_imputed	7.648%	1.742%	1.194%
AutoARIMA	Seasonal length	10.783%	22.810%	52.500%
	power_transform	36.675%	15.121%	40.480%
	is_imputed	52.541%	62.070%	7.020%

Tabel 4.4 menunjukkan hasil analisis persentase kepentingan parameter untuk prediksi harga pasar KCL menggunakan metode XGBoost dan AutoARIMA. Untuk model XGBoost, parameter n\_differentiation memiliki tingkat kepentingan tertinggi pada rentang prediksi 3 bulan dan 12 bulan dengan nilai persentase 32.565% dan 75.795%. Namun, tingkat kepentingan parameter ini menurun untuk rentang prediksi 6 bulan, menjadi hanya 1.740%. Pada rentang prediksi 6 bulan, kedudukan parameter dengan tingkat kepentingan tertinggi digantikan oleh

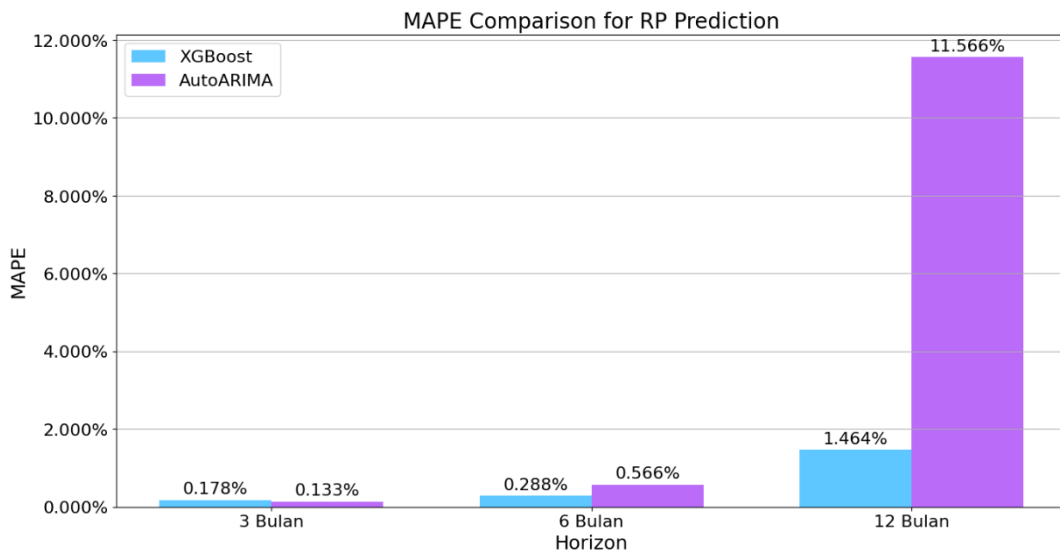
scaler dengan persentase 46.507%. Sementara itu, untuk model AutoARIMA, parameter *is\_imputed* memiliki tingkat kepentingan yang tinggi pada rentang prediksi 3 dan 6 bulan, dengan nilai 52.541% dan 62.070%. Akan tetapi, tingkat kepentingan ini menurun menjadi hanya 7.020% pada rentang prediksi 12 bulan. Pada rentang prediksi 12 bulan, parameter dengan tingkat kepentingan tertinggi adalah *Seasonal\_length*. Parameter *Seasonal\_length* menunjukkan peningkatan kepentingan yang signifikan dari 10.783% pada prediksi 3 bulan menjadi 52.500% pada prediksi 12 bulan.

### 4.3 Uji Coba 3 untuk Membandingkan Performa Model Prediksi Harga RP

Uji coba ketiga bertujuan untuk mengevaluasi kinerja model XGBoost dan AutoARIMA dalam memprediksi harga pasar pupuk RP. Performa diukur dengan menggunakan metrik evaluasi MAPE, MAE, dan RMSE.

Tabel 4.5 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar RP

Metrik Evaluasi	Model	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
MAPE	XGBoost	0.178%	<b>0.288%</b>	<b>1.464%</b>
	AutoARIMA	<b>0.133%</b>	0.566%	11.566%
MAE	XGBoost	0.618	<b>0.996</b>	<b>4.873</b>
	AutoARIMA	<b>0.461</b>	1.958	39.741
RMSE	XGBoost	0.671	<b>1.185</b>	<b>6.211</b>
	AutoARIMA	<b>0.465</b>	2.114	43.439



Gambar 4.3 Bar Plot Perbandingan MAPE Hasil Prediksi RP

Tabel 4.5 dan Gambar 4.3 menyajikan perbandingan kinerja kedua model dalam memprediksi harga pasar RP. Digunakan rentang waktu prediksi dan metrik evaluasi yang sama seperti uji coba sebelumnya. Kedua model menunjukkan performa yang bervariasi tergantung pada rentang waktu prediksi. Pada prediksi jangka pendek 3 bulan, model AutoARIMA menunjukkan performa yang sedikit lebih baik dengan MAPE sebesar 0.133% dibandingkan XGBoost yang memiliki MAPE sebesar 0.178%. Namun, pada rentang prediksi yang lebih panjang, XGBoost mulai mengungguli AutoARIMA. Pada prediksi 6 bulan, XGBoost menghasilkan MAPE sebesar 0,288%, sementara AutoARIMA menghasilkan MAPE 0,566%. Perbedaan performa menjadi semakin signifikan pada prediksi 12 bulan, di mana XGBoost menghasilkan MAPE 1,464% dibandingkan dengan 11,566% untuk AutoARIMA.

Seiring dengan peningkatan rentang waktu prediksi, XGBoost menunjukkan peningkatan MAPE yang relatif konsisten, dari 0,178% pada prediksi 3 bulan, menjadi 0,288% pada 6 bulan, dan 1,464% pada 12 bulan. Peningkatan ini menunjukkan bahwa XGBoost mampu mempertahankan tingkat akurasi yang cukup stabil meskipun rentang prediksi bertambah. Di sisi lain, AutoARIMA menunjukkan pola yang sangat berbeda. Meski awalnya unggul dengan nilai MAPE 0,133% pada prediksi 3 bulan, nilainya meningkat menjadi 0,566% pada prediksi 6 bulan, dan melonjak drastis menjadi 11,566% pada prediksi 12 bulan.

Tabel 4.6 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar RP

Metode	Parameter	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
XGBoost	scaler	22.874%	8.849%	10.403%
	n_lags	38.859%	4.687%	2.133%
	n_differentiation	11.476%	16.426%	26.154%
	power_transform	26.791%	30.908%	17.459%
	is_imputed	0.000%	39.130%	43.851%
AutoARIMA	Seasonal_length	30.185%	40.416%	38.231%
	power_transform	13.181%	17.149%	61.769%
	is_imputed	56.634%	42.435%	0.000%

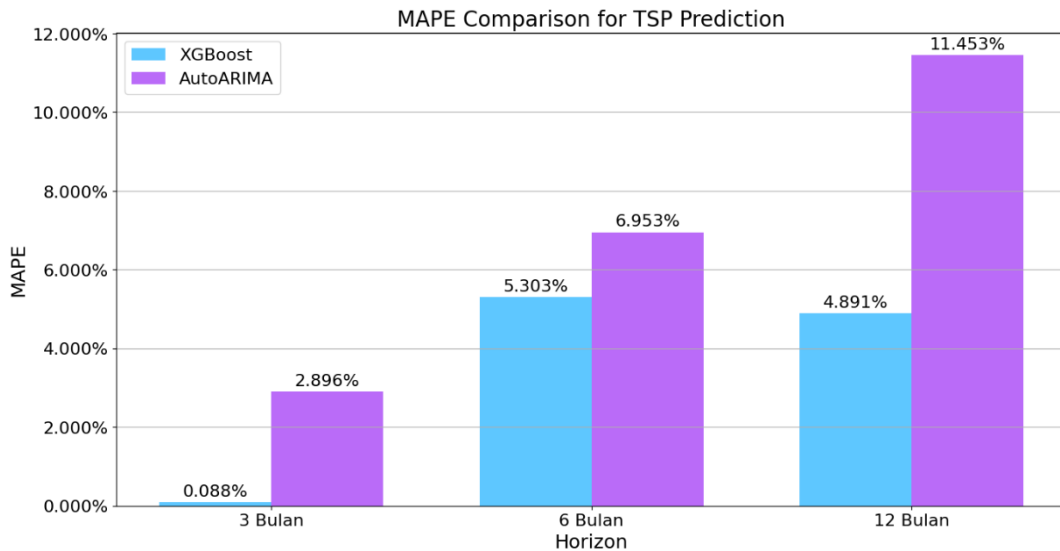
Tabel 4.6 menunjukkan hasil analisis peringkat kepentingan dari berbagai parameter dalam memprediksi harga pasar RP. Untuk model XGBoost, parameter *n\_lags* memiliki peringkat tertinggi pada prediksi 3 bulan, dengan nilai 38.859%. Namun, kepentingannya berkurang untuk rentang prediksi yang lebih panjang, dengan nilai 4.687% pada prediksi 6 bulan dan 2.133% pada prediksi 12 bulan. Parameter *is\_imputed*, yang nilainya 0.000% untuk prediksi 3 bulan, justru menjadi parameter paling penting untuk prediksi 6 bulan dan 12 bulan, dengan nilai persentase 39.130% dan 43.851%. Dalam model AutoARIMA, parameter *is\_imputed* masih menjadi parameter paling penting, khususnya untuk prediksi 3 bulan dan 6 bulan, dengan nilai persentase kepentingan masing-masing 56.634% dan 42.435%. Namun, untuk prediksi 12 bulan, tingkat kepentingan dari *is\_imputed* turun menjadi 0.000%, dan parameter *power\_transform* mengambil alih sebagai parameter paling penting dengan persentase 61.769%.

#### 4.4 Uji Coba 4 untuk Membandingkan Performa Model Prediksi Harga TSP

Uji coba keempat bertujuan untuk mengevaluasi kinerja model XGBoost dan AutoARIMA dalam memprediksi harga pasar TSP (Triple Super Phosphate). Uji coba dilakukan menggunakan metrik evaluasi MAPE, MAE, dan RMSE.

Tabel 4.7 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar TSP

Metrik Evaluasi	Model	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
MAPE	XGBoost	<b>0.088%</b>	<b>5.303%</b>	<b>4.891%</b>
	AutoARIMA	2.896%	6.953%	11.453%
MAE	XGBoost	<b>0.407</b>	<b>21.513</b>	<b>22.287</b>
	AutoARIMA	13.457	28.216	49.375
RMSE	XGBoost	<b>0.490</b>	<b>29.588</b>	<b>32.871</b>
	AutoARIMA	13.765	38.530	66.979



Gambar 4.4 Bar Plot Perbandingan MAPE Hasil Prediksi TSP

Tabel 4.7 dan Gambar 4.4 membandingkan performa kedua model ini dengan rentang waktu dan metrik evaluasi yang sama seperti pengujian sebelumnya. Pada rentang prediksi 3 bulan, XGBoost menunjukkan performa yang jauh lebih baik dengan MAPE sebesar 0.088%, sementara AutoARIMA memiliki MAPE sebesar 2.896%. Pada prediksi 6 bulan, MAPE XGBoost naik menjadi 5.303%, tetapi masih lebih rendah dibandingkan AutoARIMA yang memiliki MAPE sebesar 6.953%. Pada prediksi 12 bulan, MAPE XGBoost sedikit menurun menjadi 4.891%, sedangkan AutoARIMA menunjukkan peningkatan signifikan menjadi 11.453%.

Kedua model menunjukkan peningkatan nilai MAPE seiring dengan bertambahnya rentang waktu prediksi, kecuali dalam satu kasus. Pada model XGBoost, MAPE meningkat dari 0.088% untuk prediksi 3 bulan menjadi 5.303% untuk prediksi 6 bulan, menunjukkan peningkatan sebesar 5.215%. Namun, MAPE sedikit turun menjadi 4.891% untuk prediksi 12 bulan, dengan penurunan sebesar 0.412% dari prediksi 6 bulan. Penurunan ini tidak biasa, karena umumnya MAPE akan meningkat seiring bertambahnya rentang waktu prediksi. Secara total, dari prediksi 3 bulan ke 12 bulan, terjadi kenaikan sebesar 4.803%. Sementara itu, pada model AutoARIMA, MAPE awalnya berada di angka 2.896% untuk prediksi 3 bulan. Nilai ini meningkat menjadi 6.953% untuk prediksi 6 bulan, dengan peningkatan sebesar 4.057%. Selanjutnya, MAPE untuk prediksi 12 bulan mencapai 11.453%, yang berarti ada peningkatan sebesar 4.500% dari prediksi 6 bulan. Secara keseluruhan, dari prediksi 3 bulan ke 12 bulan, terdapat peningkatan total sebesar 8.557%.

Tabel 4.8 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar TSP

Metode	Parameter	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
XGBoost	scaler	3.293%	24.863%	13.336%
	n_lags	35.829%	49.304%	11.913%
	n_differentiation	5.026%	6.366%	2.887%
	power_transform	40.416%	14.791%	39.882%
	is_imputed	15.436%	4.676%	31.982%
AutoARIMA	Seasonal_length	7.273%	10.315%	9.638%
	power_transform	92.727%	29.895%	32.255%
	is_imputed	0.000%	59.790%	58.107%

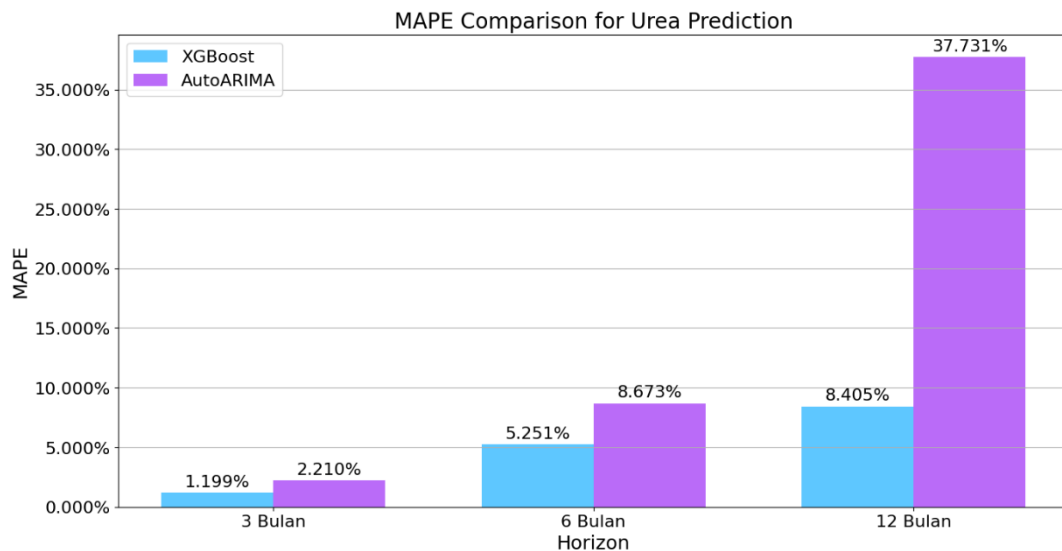
Tabel 4.8 menampilkan hasil analisis persentase kepentingan parameter untuk prediksi harga pasar TSP. Untuk metode XGBoost, parameter `power_transform` memiliki peran terpenting dalam prediksi 3 bulan dan 12 bulan dengan persentase yang konsisten, yaitu 40.416% dan 39.882%. Namun, pada prediksi 6 bulan, `power_transform` hanya memiliki tingkat kepentingan sebesar 14.791%. Parameter yang terpenting pada prediksi 6 bulan adalah `n_lags` dengan persentase sebanyak 49.304%. Sementara itu, untuk metode AutoARIMA, parameter `power_transform` menjadi yang terpenting pada prediksi 3 bulan, dengan nilai mencapai 92,727%. Akan tetapi, kepentingannya menurun drastis pada prediksi 6 bulan dan 12 bulan, menjadi 29,895% dan 32,255%. Di sisi lain, parameter `'is_imputed'` yang mempunyai tingkat kepentingan 0.000% pada prediksi 3 bulan, menunjukkan peningkatan yang signifikan pada prediksi 6 bulan dan 12 bulan, dengan nilai masing-masing 59,790% dan 58,107%.

#### 4.5 Uji Coba 5 untuk Membandingkan Performa Model Prediksi Harga Urea

Uji coba kelima dilakukan untuk mengukur kinerja model XGBoost dan AutoARIMA dalam memprediksi harga pasar Urea

Tabel 4.9 Perbandingan Metrik Evaluasi Hasil Prediksi Harga Pasar Urea

Metrik Evaluasi	Model	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
MAPE	XGBoost	<b>1.199%</b>	<b>5.251%</b>	<b>8.405%</b>
	AutoARIMA	2.210%	8.673%	37.731%
MAE	XGBoost	<b>4.853</b>	<b>16.801</b>	<b>32.187</b>
	AutoARIMA	8.949	30.714	146.317
RMSE	XGBoost	<b>6.805</b>	<b>23.785</b>	<b>38.873</b>
	AutoARIMA	12.731	33.832	159.587



Gambar 4.5 Bar Plot Perbandingan MAPE Hasil Prediksi Urea

Tabel 4.9 dan Gambar 4.5 memperlihatkan perbandingan kinerja kedua model dalam memprediksi harga pasar Urea. Digunakan metrik evaluasi dan rentang prediksi yang sama dengan uji coba sebelumnya. Pada rentang prediksi 3 bulan, XGBoost menunjukkan performa yang lebih baik dengan MAPE sebesar 1.199%, sementara AutoARIMA memiliki MAPE sebesar 2.210%. Pada prediksi 6 bulan, MAPE XGBoost meningkat menjadi 5.251%, namun masih lebih rendah dibandingkan AutoARIMA yang memiliki MAPE sebesar 8.673%. Pada

prediksi 12 bulan, MAPE XGBoost mencapai 8.405%, sedangkan AutoARIMA menunjukkan peningkatan signifikan menjadi 37.731%.

Dari Tabel 4.9, terlihat bahwa kedua model mengalami kenaikan nilai MAPE seiring dengan meningkatnya rentang waktu prediksi. Pada model XGBoost, MAPE naik dari 1.199% untuk prediksi 3 bulan menjadi 5.251% untuk prediksi 6 bulan, menunjukkan peningkatan sebesar 4.052%. Kemudian, MAPE kembali meningkat menjadi 8.405% untuk prediksi 12 bulan, dengan peningkatan sebesar 3.154% dari prediksi 6 bulan. Secara keseluruhan, dari prediksi 3 bulan ke 12 bulan, terdapat peningkatan total sebesar 7.206%. Di sisi lain, peningkatan yang jauh lebih drastis terlihat pada model AutoARIMA. MAPE awalnya berada di angka 2.210% untuk prediksi 3 bulan. Nilai ini meningkat menjadi 8.673% untuk prediksi 6 bulan, dengan peningkatan sebesar 6.463%. Selanjutnya, MAPE untuk prediksi 12 bulan mencapai 37.731%, yang berarti ada peningkatan sebesar 29.058% dari prediksi 6 bulan. Secara keseluruhan, dari prediksi 3 bulan ke 12 bulan, terdapat peningkatan total sebesar 35.521%.

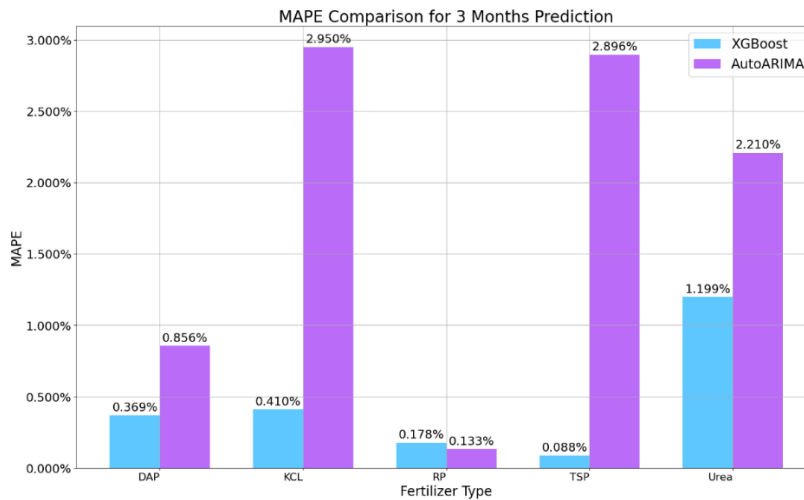
Tabel 4.10 Analisis Kepentingan Parameter untuk Prediksi Harga Pasar Urea

Metode	Parameter	Rentang Prediksi		
		3 Bulan	6 Bulan	12 Bulan
XGBoost	scaler	29.874%	66.252%	2.442%
	n_lags	29.198%	9.391%	3.801%
	n_differentiation	11.729%	13.348%	69.024%
	power_transform	25.092%	9.856%	15.244%
	is_imputed	4.107%	1.153%	9.488%
AutoARIMA	Seasonal_length	6.941%	6.456%	0.369%
	power_transform	87.837%	79.500%	9.948%
	is_imputed	5.221%	14.044%	89.683%

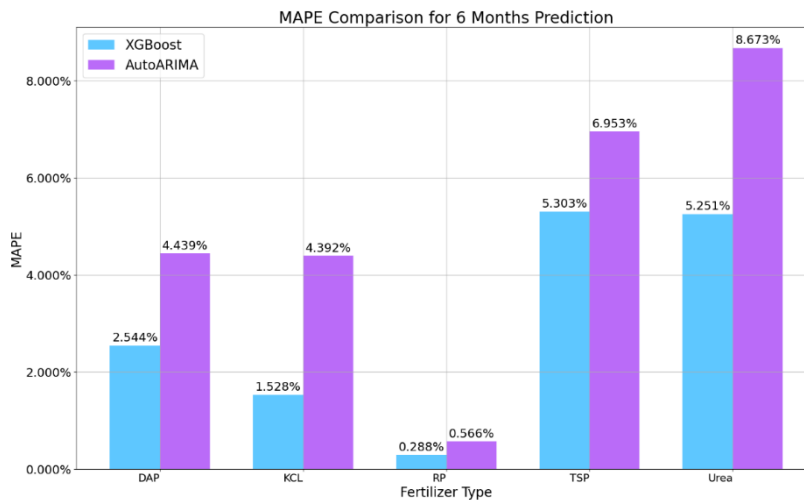
Tabel 4.10 menampilkan hasil analisis persentase kepentingan parameter dalam memprediksi harga pasar Urea. Untuk metode XGBoost, pada rentang prediksi 3 bulan, parameter scaler dan n\_lags memiliki tingkat kepentingan yang hampir sama, yaitu 29.874% dan 29.198%. Pada rentang prediksi 6 bulan, kepentingan parameter scaler meningkat menjadi 66.252%. Kemudian, pada rentang prediksi 12 bulan, n\_differentiation menjadi parameter paling penting dengan nilai sebesar 69.024%. Sementara itu, untuk metode AutoARIMA, parameter power\_transform memiliki tingkat kepentingan yang sangat tinggi pada rentang prediksi 3 bulan dan 6 bulan, dengan nilai masing-masing 87,837% dan 79,500%. Namun, pada rentang prediksi 12 bulan, tingkat kepentingannya turun menjadi 9,948%. Parameter paling krusial pada rentang prediksi 12 bulan berganti menjadi is\_imputed dengan nilai 89.683%.

#### 4.6 Pembahasan

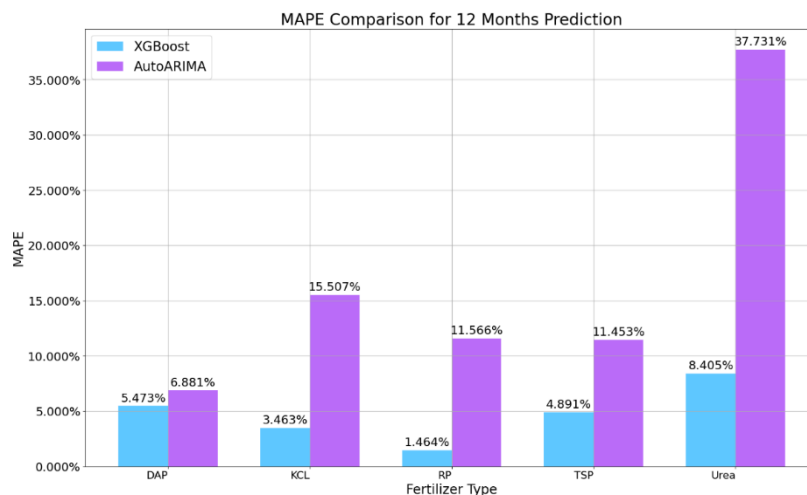
Setelah uji coba pada lima dataset harga pupuk, metode machine learning dengan model XGBoost menghasilkan metrik kesalahan MAPE, MAE, dan RMSE yang lebih rendah dibandingkan dengan metode statistik menggunakan model AutoARIMA. XGBoost mengungguli performa AutoARIMA dalam sebagian besar skenario uji coba, baik untuk rentang prediksi pendek, maupun panjang. Namun, ada satu pengecualian pada prediksi 3 bulan untuk harga RP, di mana AutoARIMA menunjukkan performa yang sedikit lebih baik dibandingkan XGBoost. Kedua model cenderung menunjukkan peningkatan metrik kesalahan seiring bertambahnya rentang waktu prediksi. Akan tetapi, AutoARIMA mengalami peningkatan metrik kesalahan yang lebih signifikan, terutama pada rentang waktu prediksi yang lebih panjang.



Gambar 4.6 Bar Plot Perbandingan MAPE Hasil Prediksi 3 Bulan



Gambar 4.7 Bar Plot Perbandingan MAPE Hasil Prediksi 6 Bulan



Gambar 4.8 Bar Plot Perbandingan MAPE Hasil Prediksi 12 Bulan

Keunggulan XGBoost mungkin karena kemampuannya dalam menangani hubungan dan pola yang kompleks serta nonlinear dalam data harga pupuk (Noorunnahar et al., 2023). Hubungan ini merujuk pada keterkaitan antara harga pupuk masa lalu dan harga pupuk masa depan, sedangkan pola merujuk pada fluktuasi harga pupuk. AutoARIMA adalah model

berbasis linear, sehingga cenderung tidak mampu menangkap hubungan dan pola tersebut sebaik XGBoost. Meskipun demikian, kasus pengecualian pada prediksi 3 bulan harga RP menunjukkan bahwa AutoARIMA masih memiliki potensi untuk prediksi jangka pendek, mengingat kompleksitas prediksi jangka pendek yang lebih rendah (Ospina et al., 2023).

Menurut Sukarsa et al. (2021), hasil MAPE dapat dibagi menjadi 4 kriteria, antara lain, “Buruk”, “Wajar”, “Baik”, dan “Sangat Baik”. Tabel 4.11 berisi rentang nilai MAPE dan kriterianya. Merujuk pada Tabel 4.11, hasil prediksi model XGBoost dan AutoARIMA menunjukkan kinerja yang beragam pada berbagai rentang waktu prediksi. Untuk prediksi 3 bulan dan 6 bulan, kedua model menunjukkan MAPE yang masuk dalam kriteria “Sangat Baik”. Pada prediksi 12 bulan, model XGBoost konsisten menunjukkan MAPE yang “Sangat Baik” untuk semua jenis pupuk. Di sisi lain, performa AutoARIMA bergantung pada jenis pupuk yang diprediksi. Untuk pupuk DAP, AutoARIMA masih mampu menghasilkan prediksi dengan MAPE dalam kriteria "Sangat Baik". Namun, untuk pupuk KCL, RP, dan TSP, MAPE hasil prediksi AutoARIMA berada dalam rentang 11% hingga 15% yang masuk dalam kriteria "Baik". Sementara itu, untuk pupuk urea, MAPE hasil prediksi AutoARIMA mencapai sekitar 37% yang masuk dalam kriteria "Wajar". Hasil ini menunjukkan bahwa model XGBoost cenderung lebih konsisten menghasilkan prediksi akurat untuk berbagai jenis pupuk dan rentang waktu prediksi. Sebaliknya, performa AutoARIMA cenderung menurun seiring dengan bertambahnya rentang waktu prediksi, terutama untuk pupuk urea.

Berdasarkan Tabel 4.12, pupuk RP cenderung lebih mudah diprediksi karena paling sering muncul sebagai jenis pupuk dengan metrik kesalahan terendah. Misalnya, pada prediksi 6 bulan menggunakan XGBoost, RP memiliki MAPE terendah sebesar 0,288%. Pada prediksi 3 bulan dengan AutoARIMA, RP juga menunjukkan MAPE terendah sebesar 0,133%. Karakteristik pergerakan harga yang berbeda dibandingkan dengan empat jenis pupuk lainnya mungkin berkontribusi terhadap kemudahan prediksi harga pupuk RP. Perbedaan ini terlihat pada Gambar 3.8 dan Gambar 3.9. Pada Box Plot Gambar 3.8, pupuk RP menunjukkan variabilitas harga paling rendah. Selain itu, pada Gambar 3.9, setelah lonjakan harga tahun 2022-2023, harga pupuk RP menetap di level tinggi, sedangkan mayoritas harga pupuk lain kembali turun. Di sisi lain, berdasarkan Tabel 4.13, pupuk Urea menjadi jenis pupuk yang paling sulit diprediksi dengan metrik kesalahan tertinggi pada berbagai uji coba. Hal ini terutama terlihat dari tingginya MAPE Urea pada prediksi 12 bulan dengan nilai 8.405% untuk XGBoost dan bahkan mencapai 37.731% untuk AutoARIMA. Kesulitan prediksi harga pupuk urea mungkin terkait dengan perannya sebagai pupuk nitrogen utama yang dibutuhkan untuk memenuhi permintaan pangan global (Ding et al., 2023). Peran ini bisa membuat fluktuasi harga pupuk urea menjadi lebih kompleks dan dipengaruhi oleh faktor-faktor yang lebih sulit dimodelkan.

Tabel 4.11 Rentang Kriteria MAPE

<b>Rentang MAPE</b>	<b>Kriteria</b>
<10%	Sangat Baik
10% - 20%	Baik
20% - 50%	Wajar
>50%	Buruk



Tabel 4.12 Rangkuman Jenis Pupuk dengan Metrik Kesalahan Terendah

Metode	Rentang Prediksi	Jenis Pupuk dengan Metrik Kesalahan Terendah	Metrik Kesalahan		
			MAPE	MAE	RMSE
XGBoost	3	TSP	0.088%	0.490	0.407
	6	RP	<b>0.288%</b>	<b>1.185</b>	<b>0.996</b>
	12	RP	<b>1.464%</b>	<b>6.211</b>	<b>4.873</b>
AutoARIMA	3	RP	<b>0.133%</b>	<b>0.465</b>	<b>0.461</b>
	6	RP	<b>0.566%</b>	<b>2.114</b>	<b>1.958</b>
	12	DAP	6.881%	49.764	37.857

Tabel 4.13 Rangkuman Jenis Pupuk dengan Metrik Kesalahan Tertinggi

Metode	Rentang Prediksi	Jenis Pupuk dengan Metrik Kesalahan Tertinggi	Metrik Kesalahan		
			MAPE	MAE	RMSE
XGBoost	3	Urea	<b>1.199%</b>	<b>6.805</b>	<b>4.853</b>
	6	TSP	5.303%	29.588	21.513
	12	Urea	<b>8.405%</b>	<b>38.873</b>	<b>32.187</b>
AutoARIMA	3	KCL	2.950%	12.580	9.896
	6	Urea	<b>8.673%</b>	<b>33.832</b>	<b>30.714</b>
	12	Urea	<b>37.731%</b>	<b>159.587</b>	<b>146.317</b>

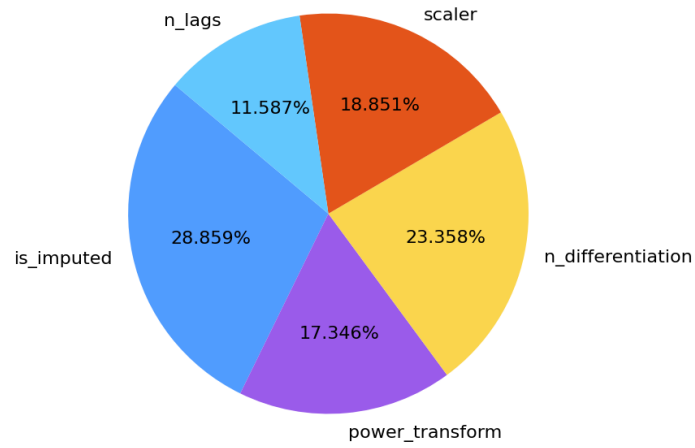
Tabel 4.14 Rangkuman Analisis Kepentingan Parameter

Metode	Parameter	Rata-rata Persentase Tingkat Kepentingan	Nilai Paling Sering Digunakan untuk Hasil Terbaik
XGBoost	<b>is_imputed</b>	<b>28.860%</b>	False
	power_transform	17.346%	Log
	<b>n_differentiation</b>	<b>23.358%</b>	<b>1</b>
	Scaler	18.851%	MinMaxScaler
	n_lags	11.587%	7
AutoARIMA	<b>is_imputed</b>	<b>37.297%</b>	False
	<b>power_transform</b>	<b>38.365%</b>	None
	Seasonal length	24.338%	4

Tabel 4.14, Gambar 4.9, dan Gambar 4.10 merangkum semua rata-rata persentase kepentingan parameter dalam memprediksi harga kelima jenis pupuk. Untuk model XGBoost, parameter *is\_imputed* dan *n\_differentiation* menjadi dua parameter terpenting dengan rata-rata persentase tingkat kepentingan sebesar 28.860% dan 23.358%. Sementara itu, untuk model AutoARIMA, parameter *is\_imputed* dan *power\_transform* menjadi dua parameter terpenting dengan rata-rata persentase tingkat kepentingan sebesar 38.365% dan 37.297%. Tingginya kepentingan parameter-parameter ini menandakan bahwa pertimbangan terhadap keberadaan data pencilan, stasioneritas, dan normalitas distribusi dapat membantu membangun model prediksi yang akurat. Menariknya, penggunaan nilai False untuk parameter *is\_imputed* lebih sering menghasilkan hasil terbaik, mengindikasikan bahwa penerapan imputasi data pencilan pada harga pupuk cenderung berdampak negatif terhadap akurasi prediksi. Hal ini mungkin

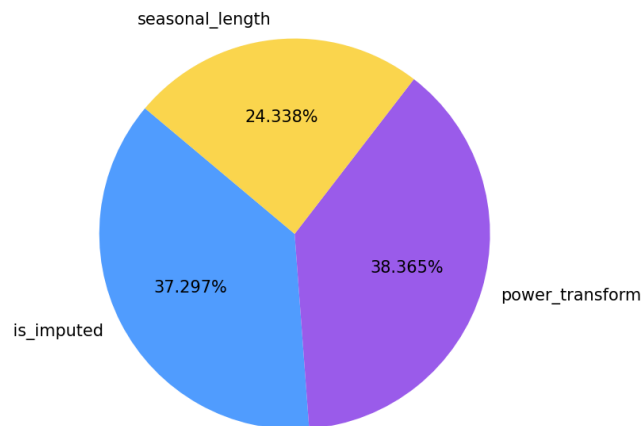
disebabkan oleh karakteristik data deret waktu harga pupuk yang mengandung kenaikan eksponensial (Romanuke, 2022).

Rata-rata Persentase Tingkat Kepentingan Parameter XGBoost



Gambar 4.9 Rata-rata Persentase Tingkat Kepentingan Parameter untuk XGBoost

Rata-rata Persentase Tingkat Kepentingan Parameter AutoARIMA



Gambar 4.10 Rata-rata Persentase Tingkat Kepentingan Parameter untuk AutoARIMA

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dalam pengerjaan tugas akhir ini, berdasarkan tahap perancangan, implementasi metode, dan uji coba, diperoleh kesimpulan sebagai berikut:

- a. Proses pemrosesan data untuk data harga pasar pupuk dilakukan dengan imputasi median, *power transform*, *differencing*, *scaling*, dan pembuatan variabel lag. Teknik-teknik pemrosesan data tersebut memperhatikan sifat-sifat statistik dari data deret waktu, khususnya keberadaan data pencilan, kenormalan distribusi data, dan stasioneritas.
- b. Pengimplementasian model prediksi harga pasar pupuk dilakukan dengan model statistik, seperti AutoARIMA, dan model machine learning, seperti XGBoost. Untuk memprediksi nilai masa depan deret waktu, model machine learning menggunakan teknik *multi-step recursive forecasting* dengan bantuan variabel lag.
- c. Pengevaluasian kinerja model prediksi harga pasar pupuk dilakukan dengan menggunakan metrik evaluasi MAPE, RMSE, dan MAE. MAPE menyajikan tingkat kesalahan hasil prediksi dalam bentuk persentase. Sementara itu, MAE dan RMSE menyajikan tingkat kesalahan hasil prediksi dalam skala yang sama dengan data pelatihan.

### 5.2 Saran

Saran yang dapat diberikan berdasarkan implementasi model prediksi harga pupuk pada tugas akhir ini adalah sebagai berikut:

1. Melakukan hyperparameter *tuning* yang lebih ekstensif pada model XGBoost dan AutoARIMA dengan didukung oleh teknik optimasi parameter lain seperti Bayesian Optimization.
2. Mengintegrasikan faktor-faktor eksternal seperti kondisi pasar global, kebijakan pemerintah, dan tren komoditas ke dalam model prediksi untuk meningkatkan akurasi, terutama untuk prediksi jangka panjang.

*(Halaman ini Sengaja Dikosongkan)*

## DAFTAR PUSTAKA

- Abdulqader, Q., Tareq, M., Mustafa Abdulqader, Q., Tariq Hassan, M., & Hasan Ahmad, K. (2020). *Building a Mathematical SARIMA Model for Forecasting the Number of Monthly Injured People by Traffic Accidents in Erbil City* (Vol. 62).  
<https://www.researchgate.net/publication/341099823>
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework*. <http://arxiv.org/abs/1907.10902>
- Alim, M., Ye, G. H., Guan, P., Huang, D. S., Zhou, B. Sen, & Wu, W. (2020). Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: A time-series study. *BMJ Open*, 10(12). <https://doi.org/10.1136/bmjopen-2020-039676>
- Armstrong, J. S. (2001). *Extrapolation for Time-Series and Cross-Sectional Data* (pp. 217–243). [https://doi.org/10.1007/978-0-306-47630-3\\_11](https://doi.org/10.1007/978-0-306-47630-3_11)
- Baffes, J., & Wee, C. K. (2022). *Fertilizer prices expected to remain higher for longer*. <https://blogs.worldbank.org/en/opendata/fertilizer-prices-expected-remain-higher-longer>
- Borkar, P. (2023). Statistical Modeling for Forecasting Fertilizer Consumption in India. *Plant Science Today*, 10. <https://doi.org/10.14719/pst.1982>
- Brownlee, J. (2020). *Introduction to Time Series Forecasting with Python*.
- De Amorim, L. B. V, Cavalcanti, G. D. C., & Cruz, R. M. O. (2022). *The choice of scaling technique matters for classification performance*. <https://github.com/amorimlb/scaling>
- Ding, J., Ye, R., Fu, Y., He, Y., Wu, Y., Zhang, Y., Zhong, Q., Kung, H. H., & Fan, M. (2023). Direct synthesis of urea from carbon dioxide and ammonia. In *Nature Communications* (Vol. 14, Issue 1). Nature Research. <https://doi.org/10.1038/s41467-023-40351-5>
- Dixit, A., & Jain, S. (2021). Effect of stationarity on traditional machine learning models: Time series analysis. *2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)*, 303–308. <https://doi.org/10.1145/3474124.3474167>
- Durgapal, A., & Vimal, V. (2021). Prediction of Stock Price Using Statistical and Ensemble learning Models: A Comparative Study. *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2021*. <https://doi.org/10.1109/UPCON52273.2021.9667644>
- Fufa, G. B. (2021). Determinants of Pro-Poor Growth and Its Impacts on Income Share: Evidence from Ethiopian Time Series Data. *Scientific World Journal*, 2021. <https://doi.org/10.1155/2021/6645789>
- Gao, K., Yang, Y., Zhang, T., Li, A., & Qu, X. (2021). Extrapolation-enhanced model for travel decision making: An ensemble machine learning approach considering behavioral theory. *Knowledge-Based Systems*, 218. <https://doi.org/10.1016/j.knosys.2021.106882>
- Gargouri, M., Chtara, C., Charrock, P., Nzihou, A., & El Feki, H. (2011). Synthesis and Physicochemical Characterization of Pure Diammonium Phosphate from Industrial Fertilizer. *Industrial & Engineering Chemistry Research*, 50(11), 6580–6584. <https://doi.org/10.1021/ie102237n>
- Gok, H. (2018). *Tree-based Models*. <https://hrngok.github.io/posts/tree-based%20models/>
- Guerrero, V. M., & Perera, R. (2004). Variance stabilizing power transformation for time series. *Journal of Modern Applied Statistical Methods*, 3(2), 357–369. <https://doi.org/10.22237/jmasm/1099267740>

- Guo, Z. (2023). Research on the Augmented Dickey-Fuller Test for Predicting Stock Prices and Returns. *Advances in Economics, Management and Political Sciences*, 44(1), 101–106. <https://doi.org/10.54254/2754-1169/44/20232198>
- Henzel, J., Bularz, J., & Sikora, M. (2021). Impact of time series clustering on fuel sales prediction results. *Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems*, 26, 13–21. <https://doi.org/10.15439/2021f129>
- Hussain, A., Ahmad, H. T., Nazir, Q., Arshad, M., Mustafa, A., Afzal, A., & Zeb, H. (2015). *Comparative effectiveness of SOP and MOP for crop productivity in Pakistani soils-A review Comparative effectiveness of SOP and MOP for crop productivity in Pakistani soils-A review International Journal of Agronomy and Agricultural Research (IJAAR)*. <http://www.innspub.net>
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice (3rd Edition)*. <https://otexts.com/fpp3/>
- Ibendahl, G. (2020). *Predicting Nitrogen Fertilizer Prices*. [www.dtnpf.com/agriculture/web/ag/crops](http://www.dtnpf.com/agriculture/web/ag/crops).
- Ilinova, A., Dmitrieva, D., & Kraslawski, A. (2021). Influence of COVID-19 pandemic on fertilizer companies: The role of competitive advantages. *Resources Policy*, 71, 102019. <https://doi.org/10.1016/J.RESOURPOL.2021.102019>
- In, Y. J., & Jung, J. Y. (2022). Simple averaging of direct and recursive forecasts via partial pooling using machine learning. *International Journal of Forecasting*, 38(4), 1386–1399. <https://doi.org/10.1016/j.ijforecast.2021.11.007>
- International Fertilizer Industry Association (IFA). (2007). *Developing a Crop-Based, Expert-Based Fertilizer Demand Forecast*.
- James, A., & Tripathi, V. (2021). Time series data analysis and ARIMA modeling to forecast the short-term trajectory of the acceleration of fatalities in Brazil caused by the corona virus (COVID-19). *PeerJ*, 9. <https://doi.org/10.7717/peerj.11748>
- Jierula, A., Wang, S., OH, T.-M., & Wang, P. (2021). Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data. *Applied Sciences*, 11(5), 2314. <https://doi.org/10.3390/app11052314>
- Jones, K., & Nti, F. (2022). *Impacts and Repercussions of Price Increases on the Global Fertilizer Market Prepared By: Global Market Analysis, Modeling Team*.
- K. Jana, R., Ghosh, I., & Kumar, P. N. R. (2023). An explainable AI-enabled granular ensemble machine learning framework to demystify fertilizer price movements. *Journal of the Operational Research Society*. <https://doi.org/10.1080/01605682.2023.2260908>
- Kee, J., Cardell, L., & Zereyesus, Y. A. (2023). *USDA ERS - Global Fertilizer Market Challenged by Russia's Invasion of Ukraine*. <https://www.ers.usda.gov/amber-waves/2023/september/global-fertilizer-market-challenged-by-russia-s-invasion-of-ukraine/>
- Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3), 669–679. <https://doi.org/10.1016/j.ijforecast.2015.12.003>
- Kim, S. W., & Brorsen, B. W. (2017). Forecasting urea prices. *Applied Economics*, 49(49), 4970–4981. <https://doi.org/10.1080/00036846.2017.1296554>
- Korstanje, J. (2021). Advanced forecasting with python: With state-of-the-art-models including LSTMs, Facebook's prophet, and Amazon's DeepAR. In *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's*

- Prophet, and Amazon's DeepAR*. Apress Media LLC. <https://doi.org/10.1007/978-1-4842-7150-6>
- Krishnan, N., Karthikeyan, M., Tamilnadu College of Engineering, Institute of Electrical and Electronics Engineers. Madras Section. Podhigai Subsection, Institute of Electrical and Electronics Engineers. Madras Section. Signal Processing/Computational Intelligence/Computer Joint Societies Chapter, & Institute of Electrical and Electronics Engineers. (2017). *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICIC) : 2017 December 14-16 : venue: Tamilnadu College of Engineering, Coimbatore-641 659, Tamilnadu, India*.
- Kumari, K., & Phogat, V. K. (2008). *Rock Phosphate: Its Availability And Solubilization In The Soil-A Review See Profile*. <https://www.researchgate.net/publication/265469230>
- Lepot, M., Aubin, J. B., & Clemens, F. H. L. R. (2017). Interpolation in time series: An introductory overview of existing methods, their performance criteria and uncertainty assessment. In *Water (Switzerland)* (Vol. 9, Issue 10). MDPI AG. <https://doi.org/10.3390/w9100796>
- Maharana, R., Basu, A., Dhal, N. K., & Adak, T. (2021). Biosolubilization of rock phosphate by *Pleurotus ostreatus* with brewery sludge and its effect on the growth of maize (*Zea mays* L.). *Journal of Plant Nutrition*, 44(3). <https://doi.org/10.1080/01904167.2020.1822397>
- Malistov, A., & Trushin, A. (2019). Gradient Boosted Trees with Extrapolation. *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019, 2019-January*, 783–789. <https://doi.org/10.1109/ICMLA.2019.00138>
- Moya, H. (2021). *Machine learning model optimization with hyper-parameter tuning approach*. <https://www.researchgate.net/publication/354495368>
- National Institute of Standards and Technology. (2012). *NIST/SEMATECH e-Handbook of Statistical Methods*. <https://www.itl.nist.gov/div898/handbook/>
- Nau, D. (2020). *Statistical forecasting: notes on regression and time series analysis*. <https://people.duke.edu/~rnau/411home.htm>
- Newton, M. J. (2019). *Develop A Fertilizer Price Forecasting Model To Assist With Farm Management Decisions*.
- Noorunnahar, M., Chowdhury, A. H., & Mila, F. A. (2023). A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh. *PLoS ONE*, 18(3 March). <https://doi.org/10.1371/journal.pone.0283452>
- Ospina, R., Gondim, J. A. M., Leiva, V., & Castro, C. (2023). An Overview of Forecast Analysis with ARIMA Models during the COVID-19 Pandemic: Methodology and Case Study in Brazil. In *Mathematics* (Vol. 11, Issue 14). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/math11143069>
- Paliari, I., Karanikola, A., & Kotsiantis, S. (2021, July 12). A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting. *IISA 2021 - 12th International Conference on Information, Intelligence, Systems and Applications*. <https://doi.org/10.1109/IISA52424.2021.9555520>
- Parra-Plazas, J., Gaona-Garcia, P., & Plazas-Nossa, L. (2023). Time series outlier removal and imputing methods based on Colombian weather stations data. *Environmental Science and Pollution Research*, 30(28), 72319–72335. <https://doi.org/10.1007/s11356-023-27176-x>
- Peixeiro, M. (2022). *Time Series Forecasting in Python*.
- Proietti, T., & Lütkepohl, H. (2013). Does the Box-Cox transformation help in forecasting macroeconomic time series? *International Journal of Forecasting*, 29(1), 88–99. <https://doi.org/10.1016/j.ijforecast.2012.06.001>

- Rahman, M. S., & Chowdhury, A. H. (2022). A data-driven eXtreme gradient boosting machine learning model to predict COVID-19 transmission with meteorological drivers. *PLoS ONE*, 17(9 September). <https://doi.org/10.1371/journal.pone.0273319>
- Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., & Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 729–735. <https://doi.org/10.1109/ICSSIT48917.2020.9214160>
- Rice, W. L., Park, S. Y., Pan, B., & Newman, P. (2019). Forecasting campground demand in US national parks. *Annals of Tourism Research*, 75, 424–438. <https://doi.org/10.1016/j.annals.2019.01.013>
- Rodrigo, J. A., & Ortiz, J. E. (2023). *skforecast (Version 0.11.0)*. <https://doi.org/10.5281/zenodo.8382788>
- Rodrigues, P. M. M., & Osborn, D. R. (1999). Performance of seasonal unit root tests for monthly data. *Journal of Applied Statistics*, 26(8), 985–1004. <https://doi.org/10.1080/02664769921981>
- Romanuke, V. (2022). Impact of Starting Outlier Removal on Accuracy of Time Series Forecasting. *Maritime Technical Journal*, 224(1), 1–15. <https://doi.org/10.2478/sjpna-2022-0001>
- Seesanong, S., Seangarun, C., Boonchom, B., Laohavisuti, N., Chaiseeda, K., & Boonmee, W. (2021). Composition and Properties of Triple Superphosphate Obtained from Oyster Shells and Various Concentrations of Phosphoric Acid. *ACS Omega*, 6(34), 22065–22072. <https://doi.org/10.1021/acsomega.1c02660>
- Sukarsa, I. M., Pandika Pinata, N. N., Kadek Dwi Rusjyanthi, N., & Wisswani, N. W. (2021). Estimation of Gourami Supplies Using Gradient Boosting Decision Tree Method of XGBoost. *TEM Journal*, 10(1), 144–151. <https://doi.org/10.18421/TEM101-17>
- Van Greunen, J., Heymans, A., Van Heerden, C., & Van Vuuren, G. (2014). The prominence of stationarity in time series forecasting. *Journal for Studies in Economics and Econometrics*, 38(1), 1–16. <https://doi.org/10.1080/10800379.2014.12097260>
- Yaro, A. S., Maly, F., & Prazak, P. (2023). Outlier Detection in Time-Series Receive Signal Strength Observation Using Z-Score Method with Sn Scale Estimator for Indoor Localization. *Applied Sciences (Switzerland)*, 13(6). <https://doi.org/10.3390/app13063900>
- Zhai, N., Yao, P., & Zhou, X. (2020). *Multivariate time series forecast in industrial process based on XGBoost and GRU*. 1397–1400. <https://doi.org/10.1109/ITAIC49862.2020.9338878>
- Zhang, L., Bian, W., Qu, W., Tuo, L., & Wang, Y. (2021). Time series forecast of sales volume based on XGBoost. *Journal of Physics: Conference Series*, 1873(1). <https://doi.org/10.1088/1742-6596/1873/1/012067>
- Zhang, L., Elghazoly, S., & Tweedie, B. (2019). *Introducing AnomDB: An Unsupervised Anomaly Detection Method for CNC Machine Control Data*.
- Zuo, X. (2019). *Several Important Unit Root Tests*.



## **BIODATA PENULIS**



Penulis dilahirkan di Surabaya, 23 April 2003, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal di TK Kuncup Bunga Surabaya, SD Labschool UNESA Surabaya, SMPN 12 Surabaya, dan SMAN 5 Surabaya. Setelah lulus dari SMAN tahun 2020, penulis diterima di Departemen Teknik Informatika FTEIC – ITS pada tahun 2020 dan terdaftar dengan NRP 505201268 di Departemen Teknik Informatika. Penulis aktif sebagai salah satu administrator Lab Algoritma dan Pemrograman.