

# **FINAL PROJECT**

- **Analisis Peramalan Penjualan Produk Multi-Purpose Vehicle (MPV) Dengan Model Hibrida Time-Series dan Machine Learning**

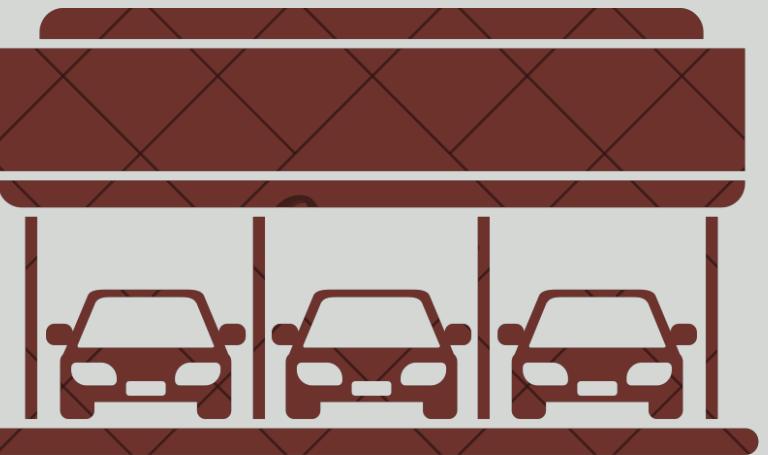
Eriel Matthew Gerald Morado

# Gambaran Umum Objek



Perusahaan Z

Distribusi  
→

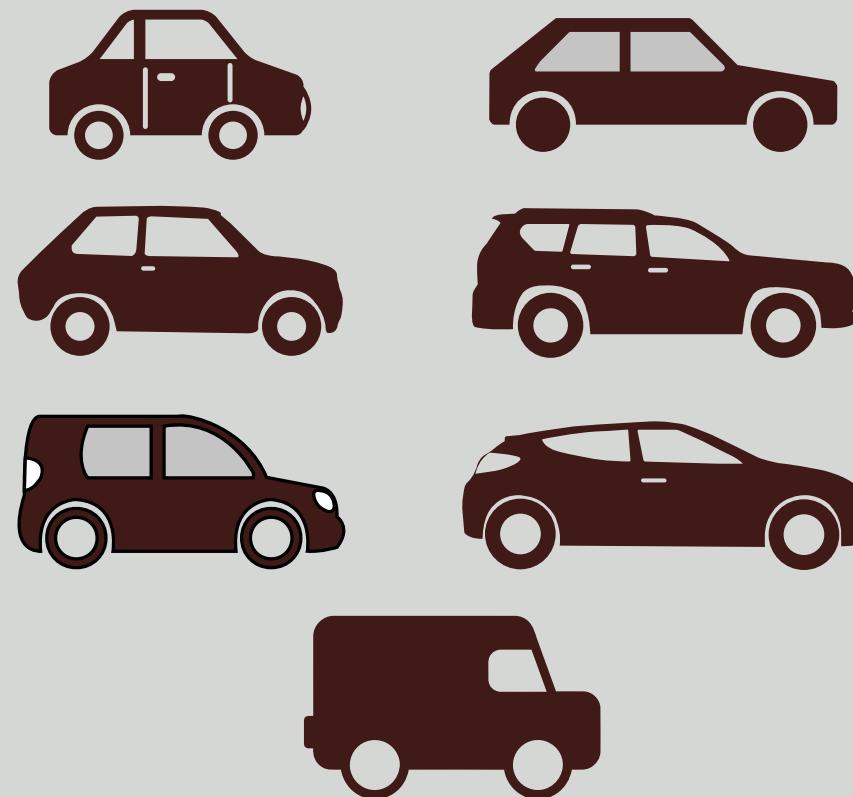


Dealer U



Kota Malang

## PASSENGER CAR



## COMMERCIAL CAR



# Latar Belakang

## Penjualan Produk Otomotif yang Fluktuatif

Industri otomotif, sama seperti industri lainnya, memiliki **ketidakpastian** yang mengakibatkan **fluktuasi penjualan** seiring waktu.

Penjualan produk otomotif di Indonesia mengalami **penjualan yang “naik-turun” sepanjang tahun 2014 - 2024**, dengan penjualan yang **cenderung menurun** dalam 2 tahun terakhir.

### Penjualan Retail Otomotif di Indonesia



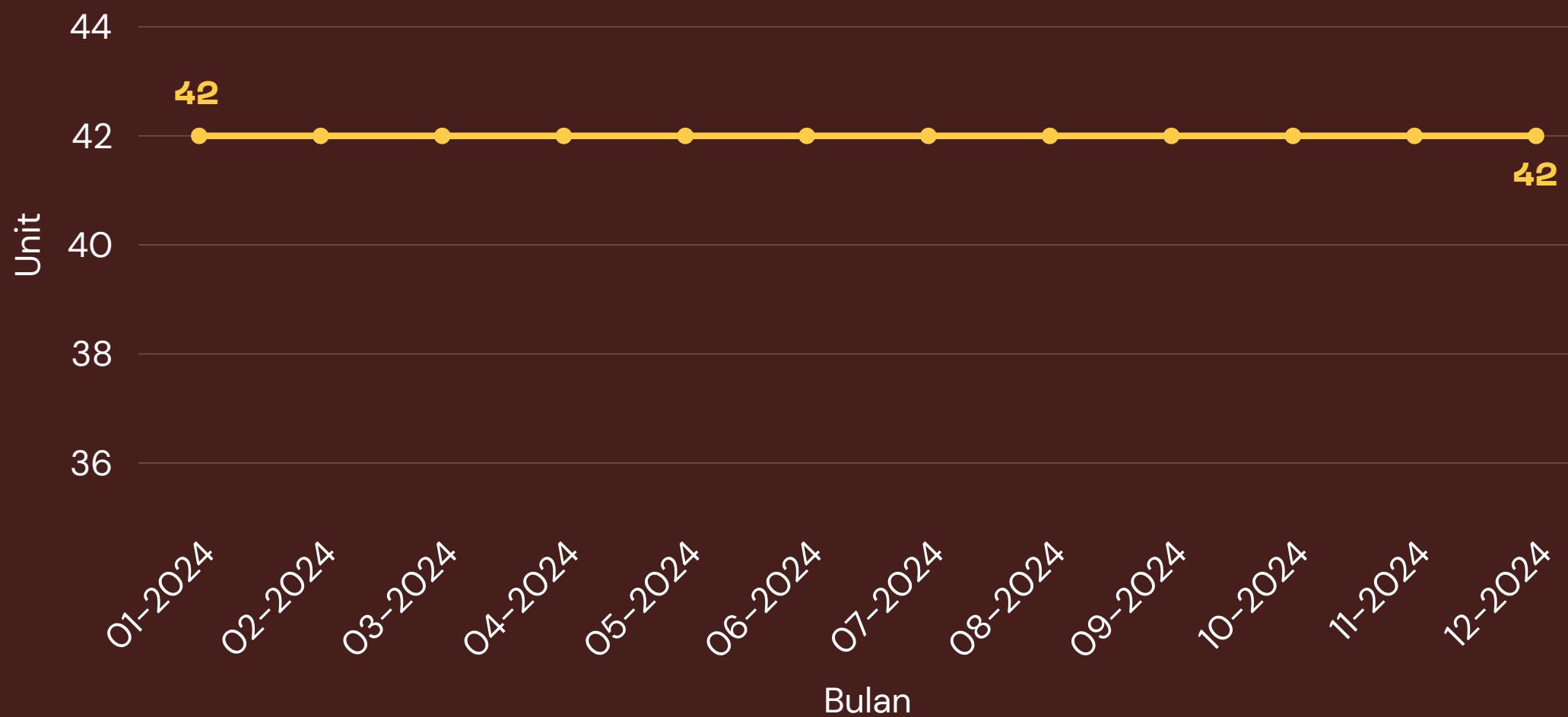
# Latar Belakang

## Penetapan Target Tak Berdasar

Dealer U tidak memiliki dasar yang kuat dalam penetapan target penjualan.

Target bulanan ditetapkan melalui jumlah karyawan sales pada dealer yang berjumlah **21 orang**, yakni **2 unit per orang**, sehingga target penjualan bulanan adalah sebanyak **42 unit**.

### Target Penjualan Bulanan pada Dealer U

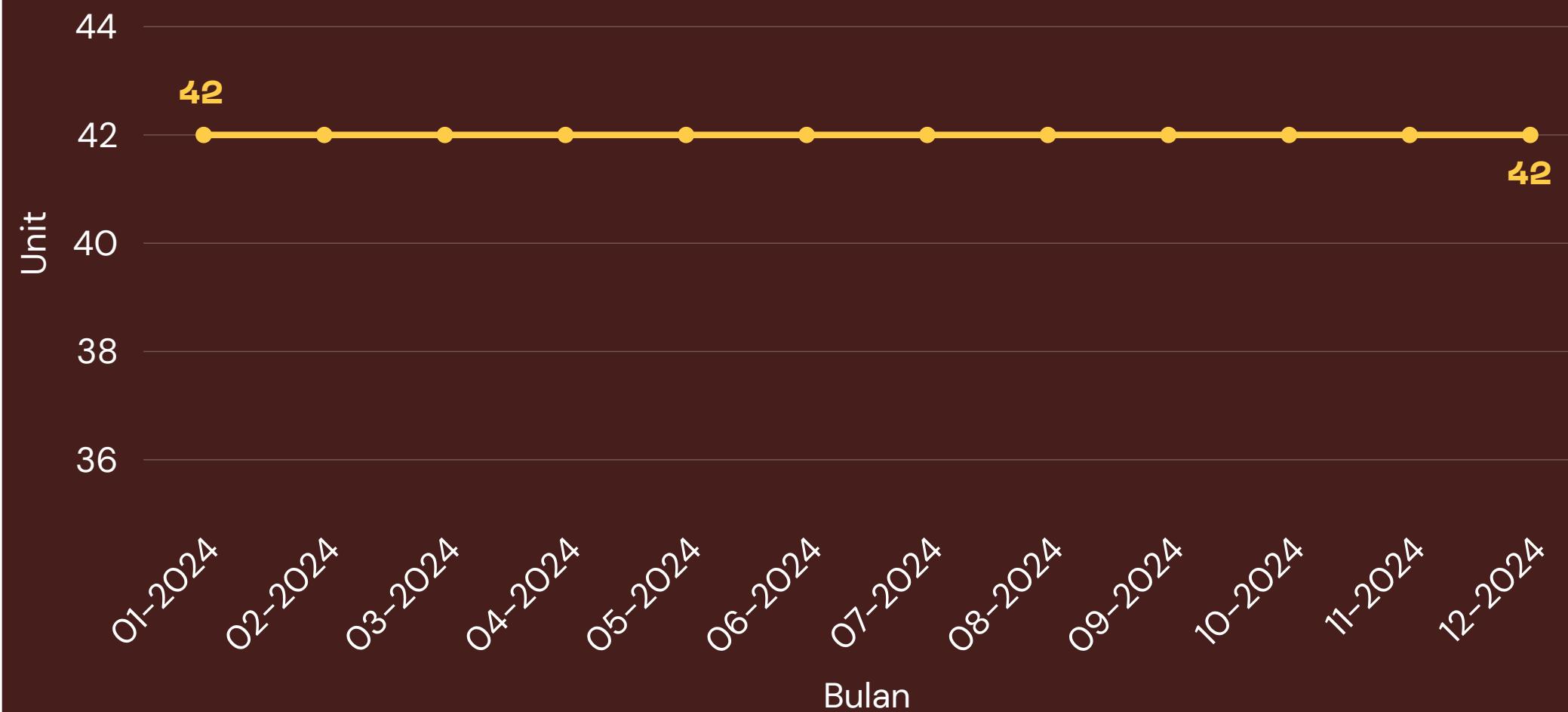


# Latar Belakang

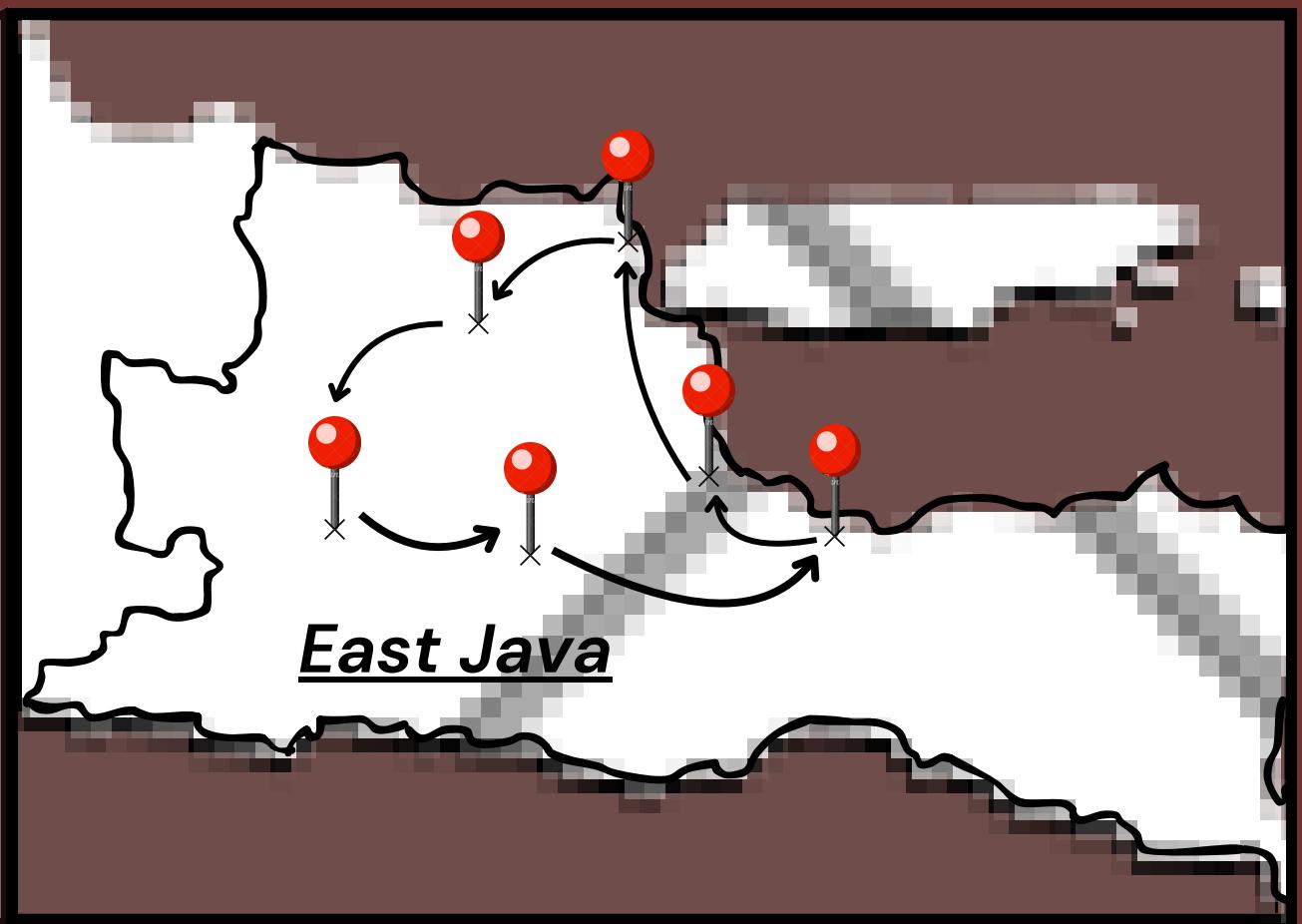
## Penetapan Target Tak Berdasar

Perusahaan Z menetapkan **punishment** terhadap *branch manager* dari dealer terkait jika **tidak dapat mencapai 85%** dari target tahunan sebanyak 500 unit, yakni sebanyak **425 unit**.

### Target Penjualan Bulanan pada Dealer U



# Latar Belakang



Branch Manager (BM) dari dealer yang tidak mencapai target penjualan sebanyak 425 unit akan dipindahkan ke dealer pada daerah yang lebih kecil, baik secara geografi maupun demografi.

# Latar Belakang

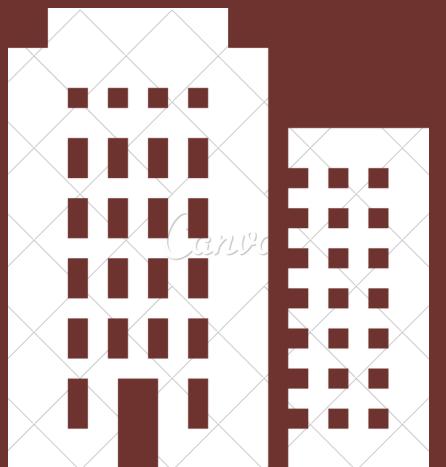


Januari 2025



Branch Manager  
Dealer U

→  
*rolled-in*



Dealer U

BM dari Dealer U saat ini merupakan BM yang sebelumnya mengelola dealer pada daerah Mojokerto, Jawa Timur.

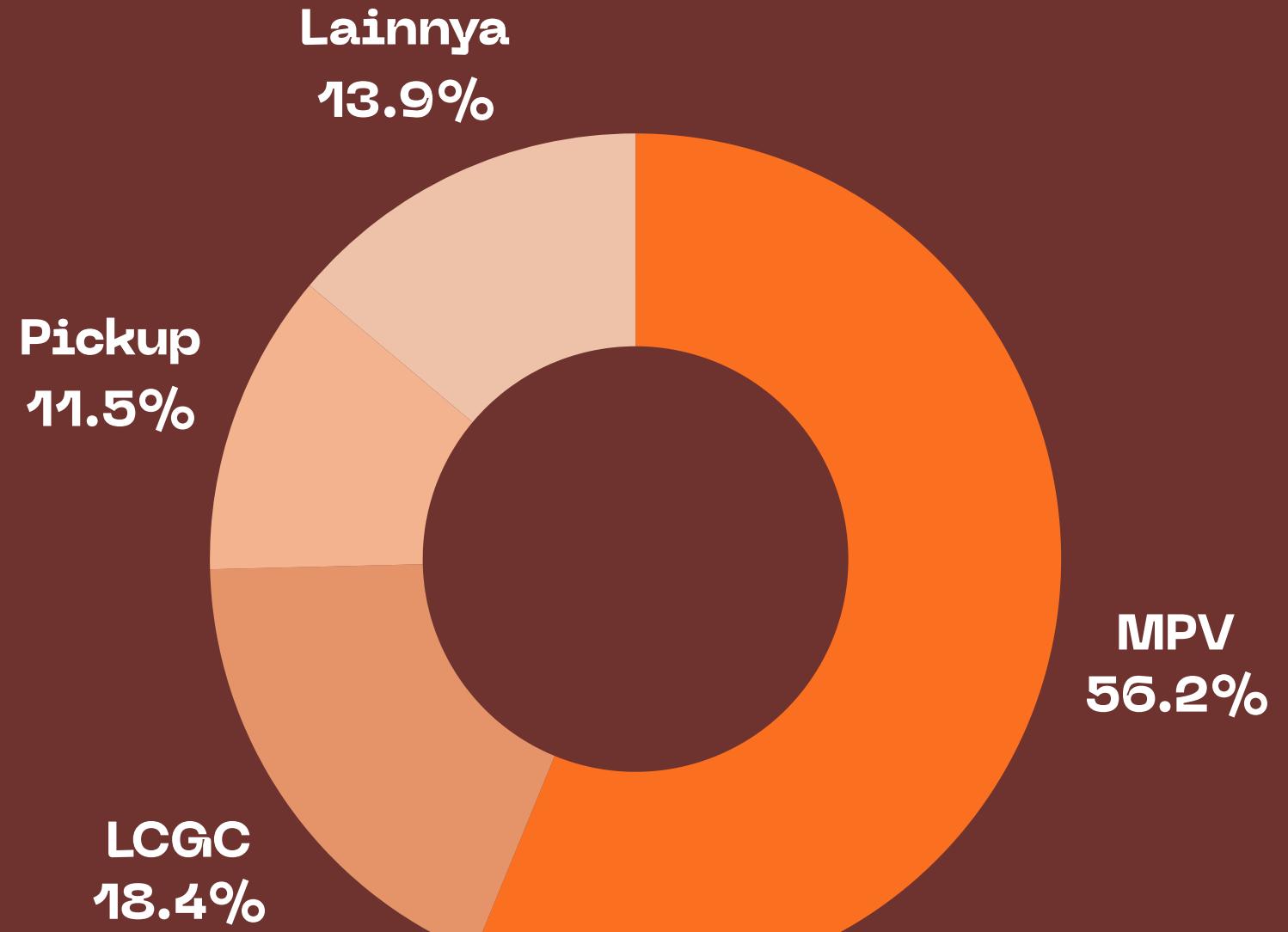


BM dari Dealer U menyampaikan bahwa mereka membutuhkan target penjualan dengan dasar yang lebih kuat dan komprehensif, karena kondisi *existing* tidak menerapkan analisis mendalam untuk menetapkan target penjualan bulanan.



Peramalan menjadi sebuah alternatif sebagai jawaban dalam penetapan target dengan dasar yang lebih kuat.

# Latar Belakang



Penjualan Retail Otomotif di Indonesia  
Januari 2025 – Maret 2025

Mobil dengan tipe *Multi-Purpose Vehicle* (MPV) menyumbang lebih dari 50% total penjualan mobil di Indonesia, sehingga tipe ini menjadi fokus pada penelitian ini.

Perusahaan Z menyediakan 3 tipe MPV, salah satunya diluncurkan di Indonesia pada tahun 2012 dan memiliki penjualan tertinggi dibandingkan 2 tipe lainnya, sehingga menjadi objek dari penelitian ini.

Tipe tersebut akan diberikan alias sebagai Produk X.

# Latar Belakang

Peramalan pada penelitian ini akan memanfaatkan data penjualan bulanan dari Produk X di Dealer U pada 11 tahun terakhir, yakni tahun 2014 - 2025, menggunakan sebuah model dengan pendekatan hibrida.

Model hibrida yang digunakan merupakan gabungan dari 2 model, yaitu model *time-series* dan model *machine learning*.

# Identifikasi Masalah

Penentuan target penjualan bulanan tidak memiliki dasar yang kuat, dan hanya ditentukan dari jumlah karyawan yang ada pada Dealer U

# Rumusan Masalah

01 —  
Bagaimana model yang sesuai untuk meramalkan penjualan dari Produk X pada Dealer U?

02 —  
Bagaimana penjualan dari Produk X pada Dealer U untuk tahun 2025?

# Tujuan Penelitian

01 —  
Membangun model dengan performa yang memadai untuk meramalkan permintaan dari Produk X

02 —  
Meramalkan penjualan Produk X pada Dealer U untuk tahun 2025

# Manfaat Penelitian

01

Sebagai acuan dalam menentukan target penjualan bulanan dari Produk X pada Dealer U untuk tahun 2025

02

Model dapat digunakan untuk meramalkan penjualan dari Produk X untuk periode waktu lainnya

# Batasan Penelitian

01

Peramalan permintaan dibuat berdasarkan data historis penjualan Produk X pada Dealer U dari periode Januari 2014 – Februari 2025

02

Penelitian hanya berfokus pada satu dealer di Malang, sehingga hasil penelitian tidak dapat digeneralisasikan untuk dealer ataupun tipe produk lain

03

Peramalan permintaan hanya mengandalkan variabel inflasi, indeks harga konsumen, dan variabel pencarian online

# Asumsi Penelitian

01

Data yang digunakan dianggap valid, akurat, dan representatif terhadap kondisi aktual

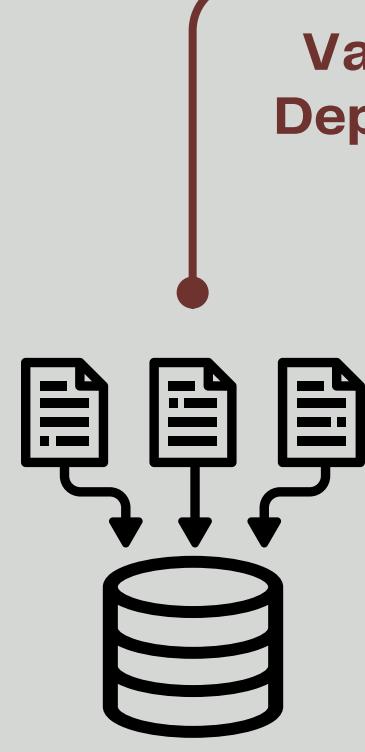
02

Peramalan nilai variabel prediktor di masa mendatang dianggap tidak memiliki error yang signifikan

# **HASIL DAN PEMBAHASAN**

---

# Pengumpulan Data



Variabel  
Dependen

## Data Penjualan Bulanan Produk X pada Dealer U

Diperoleh melalui database Dealer U.

## Data Pencarian Online

Diperoleh melalui Google Trends.

'...' adalah pencarian mengenai istilah tersebut.

Variabel  
Independen

## Data Ekonomi Regional

Diperoleh melalui Badan Pusat Statistik Kota Malang.

- 'Produk X'
- 'Harga Produk X (+merek)'
- 'Harga Produk X'
- 'Perusahaan Z malang'
- 'Perusahaan Z'

Inflasi

Indeks Harga Konsumen (IHK)

# Pengolahan Data

## 01 Persiapan Data

**1. Identifikasi variabel input dan output**

---

**2. Integrasi data**

---

**3. Eksplorasi data**

---

# Pengolahan Data

## 1. Identifikasi variabel input dan output

### Variabel Input:

- Inflasi
- Indeks Harga Konsumen
- Produk X
- 'Harga Produk X (+merek)'
- 'Harga Produk X'
- 'Perusahaan Z malang'
- 'Perusahaan Z'

### Variabel Output:

Penjualan Produk X

## 2. Integrasi data

Integrasi data dilakukan untuk menggabungkan data-data yang telah dikumpulkan sebelumnya.

Data yang digabungkan adalah data dari masing-masing variabel input dan data variabel output (penjualan).

## 3. Eksplorasi data

Eksplorasi data dilakukan untuk merangkum data dan mengambil karakteristik utama dari data, serta memeriksa asumsi yang dibutuhkan dalam pelatihan model dan pengujian hipotesis. **Eksplorasi data melibatkan proses-proses berikut.**

- Pemeriksaan Missing Values
- Pemeriksaan Statistik Deskriptif
- Identifikasi Outlier
- Visualisasi Data
- Decomposition

# Pengolahan Data

## Dataset

Meng-import seluruh data dari Excel ke Jupyter Notebook, kemudian dilakukan **merge** dengan syntax berikut:

```
In [15]: df_search = pd.merge(Harga_Produk_X, Harga_Produk_X_Merek, on='Month')
df_search = pd.merge(df_search, Produk_X, on='Month')
df_search = pd.merge(df_search, Perusahaan_Z_malang, on='Month')
df_search = pd.merge(df_search, Perusahaan_Z, on='Month')
df_search
```

```
In [19]: df_search = pd.merge(df_search, sales, on='Month')
```

```
In [22]: df_eco = pd.merge(inflasi, ihk, on='Month')
df_eco
```

```
In [25]: df_full = pd.merge(df_eco, df_search, on='Month')
df_full
```

## Hasil

Bulan	Inflasi	IHK	'Harga Produk X'	...	'Perusahaan Z'	Penjualan
2014-01	0,25	86,79	50	...	71	50
2014-02	0,22	86,98	50	...	66	26
2014-03	0,98	87,84	57	...	65	48
2014-04	1,2	88,89	52	...	67	34
2014-05	0,71	89,52	45	...	66	32
2014-06	0,05	89,57	60	...	62	51
...	...	...	...	...	...	...
2024-10	-0,45	143,40	19	...	31	4
2024-11	0,05	143,48	20	...	32	1
2024-12	0,06	143,57	23	...	37	5
2025-01	0,28	143,97	25	...	37	7
2025-02	0,10	144,11	24	...	34	1

Data yang digunakan berada pada rentang waktu Januari 2014 – Februari 2025, sehingga terdapat **134 datapoints** dalam dataset. Selain itu, terdapat **8 kolom**, yang merepresentasikan **7 fitur** dan **1 target**.

# Pengolahan Data

## Eksplorasi Data

### Statistik Deskriptif

Pemeriksaan statistik deskriptif (mean, median, standar deviasi, dll.) dari data dilakukan dengan syntax berikut:

```
In [33]: df_full.describe()
```

### Hasil

	Inflasi	IHK	'Harga Produk X'	'Harga Produk X (+merek)'	'Produk X'	'Perusahaan Z malang'	'Perusahaan Z'	Penjualan Produk X
count	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000
mean	0.318134	109.935907	43.947761	18.649254	11.126866	53.268657	49.216418	13.731343
std	1.426529	15.755314	11.965689	9.473554	4.751607	20.859777	13.435651	10.607250
min	-3.210000	86.792877	17.000000	0.000000	3.000000	0.000000	27.000000	0.000000
25%	-0.300000	100.663805	36.000000	11.000000	7.000000	39.000000	36.000000	6.000000
50%	0.155000	105.358372	44.500000	18.000000	10.500000	53.000000	51.500000	12.000000
75%	0.707500	116.163567	52.000000	24.750000	14.000000	67.500000	61.000000	17.750000
max	8.890000	144.292273	84.000000	43.000000	23.000000	100.000000	77.000000	55.000000

- Persiapan Data
- Variabel **Inflasi** memiliki **mean 0,32**, yang menunjukkan rata-rata inflasi yang relatif rendah. Nilai **minimum** menunjukkan nilai **-3,21%**, mengindikasikan **terjadinya deflasi** pada periode tertentu.
  - Pada variabel IHK, nilai **minimum** dan **maksimum** adalah sebesar **86,79** dan **144,29**, secara berturut-turut, sehingga kemungkinan terdapat **tren yang cenderung meningkat** pada variabel IHK.

- Variabel '**Harga Produk X**' memiliki **mean 43,95** dan **range 17-84**, yang menandakan **masarakat cukup sering mencari informasi dengan kata kunci ini**.
- Variabel '**Produk X**' memiliki **mean 11,13** dan **range 3-23**, mengindikasikan **pencarian yang stabil** mengenai kata kunci ini dan **ketertarikan tetap terhadap produk**.
- Variabel '**Perusahaan Z malang**' memiliki **standar deviasi** sebesar **20,86** dan **range 0-100**, mengindikasikan **pencarian yang fluktuatif**.

- Variabel Penjualan Produk X memiliki **mean 13,73** dan **standar deviasi 10,61**, menunjukkan **variabilitas yang cukup tinggi**. Selain itu, **median** berada di nilai **12**, sehingga dapat disimpulkan bahwa **lebih dari 50% data berada di bawah rata-rata** dan **distribusi bersifat condong ke kanan**.

# Pengolahan Data

## Eksplorasi Data

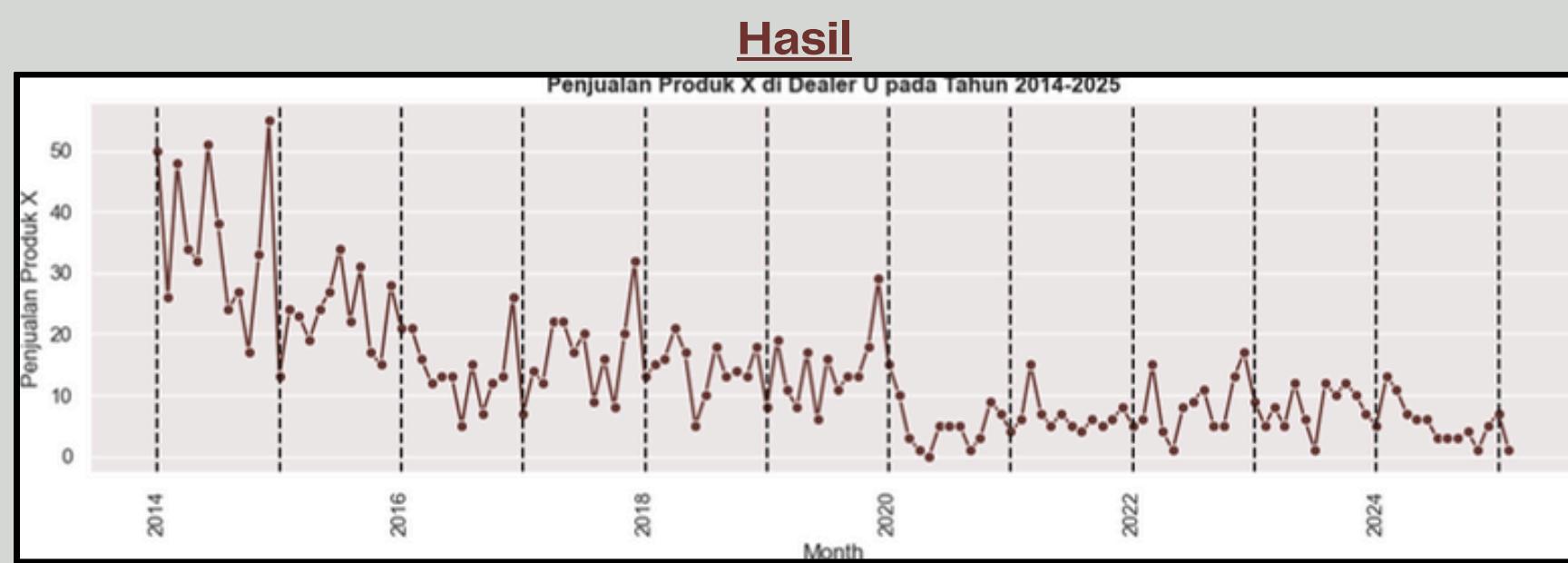
### Visualisasi Data: Pergerakan Penjualan

Visualisasi data penjualan dilakukan dengan syntax berikut:

```
In [35]: plt.figure(figsize=(12,4))
plt.xticks(rotation=90)
sns.lineplot(data=df_full, x='Month', y='Penjualan Produk X', marker='o')

for year in range(2014,2026):
    jan_date = pd.Timestamp(f'{year}-01-01')
    plt.axvline(jan_date, color='k', linestyle='--')

plt.title('Penjualan Produk X di Dealer U pada Tahun 2014-2025', fontweight='bold')
plt.ylabel('Penjualan Produk X')
plt.tight_layout()
plt.show()
```



### Interpretasi:

Visualisasi tersebut menunjukkan penjualan bulanan dari Produk X pada periode **Januari 2014 sampai dengan Februari 2025**, di mana setiap titik merepresentasikan **Jumlah unit yang terjual per bulan**.

- Penjualan **menurun secara signifikan dalam 11 tahun terakhir**.
- Tahun **2014-2015** memiliki **penjualan tertinggi yang fluktuatif**, dengan beberapa bulan melebihi 50 unit.
- **Tren menurun sejak tahun 2016**, dengan **penurunan tajam pada awal tahun 2020**, yang kemungkinan diakibatkan oleh pandemi.
- Terdapat **siklus tahunan** pada data.

- Periode awal menunjukkan potensi pasar yang tinggi, tetapi **tidak dapat dipertahankan**.
- Tingkat **fluktuasi yang tinggi** di tahun **2014-2025** berpotensi **menyebabkan masalah persediaan** (overstock atau stockout).

- **Forecasting** perlu dilakukan untuk melihat **penjualan dalam masa mendatang**.
- Perlu tindakan lebih lanjut untuk **mencegah penjualan produk menurun lebih lanjut** dan menyebabkan **produk menjadi tidak relevan**. Tindakan yang dapat dilakukan adalah melakukan **inovasi produk** atau **evaluasi harga jual**.

# Pengolahan Data

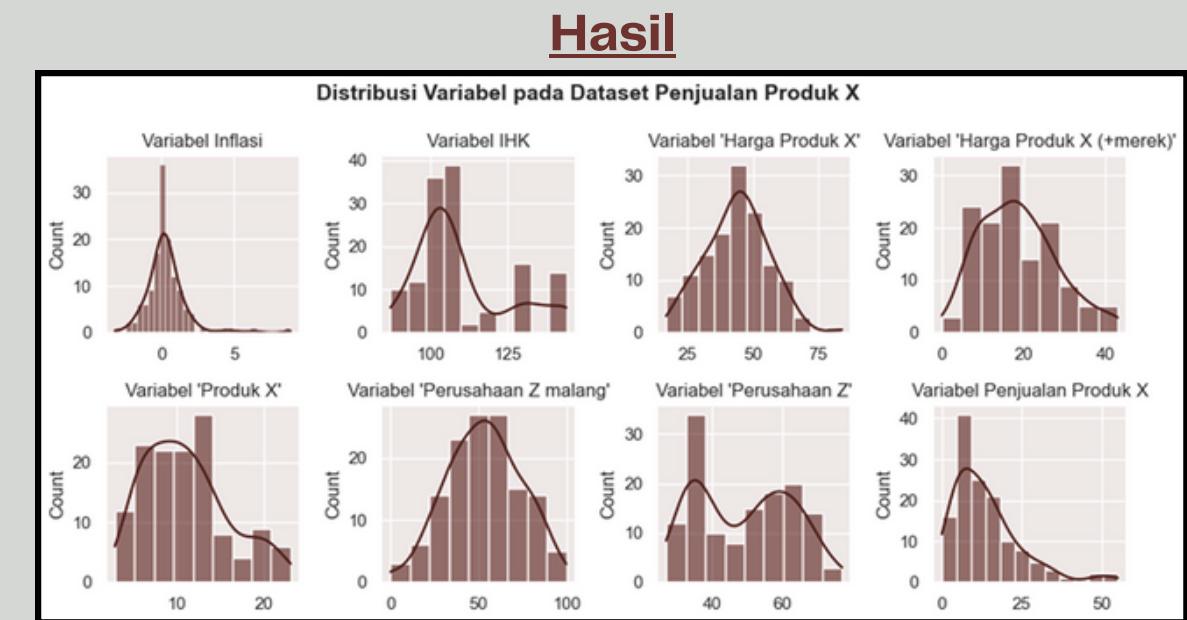
## Persiapan Data

Visualisasi distribusi variabel dilakukan dengan syntax berikut:

### Visualisasi Data: Distribusi Variabel

```
In [34]: df_cols = df_full.columns  
df_cols  
  
In [37]: fig, axes = plt.subplots(2, 4, figsize=(10,5))  
axes=axes.flatten()  
  
for i, col in enumerate(df_cols):  
    sns.histplot(df_full, x=df_full[col], ax=axes[i], kde=True, color="#004b66")  
    axes[i].set_title(f'Variabel {df_cols[i]}')  
    axes[i].set_xlabel('')  
    plt.tight_layout()  
  
plt.suptitle("Distribusi Variabel pada Dataset Penjualan Produk X", fontsize=14, fontweight='bold')  
plt.tight_layout(rect=[0,0,1,1])  
plt.show()
```

## Eksplorasi Data



### Interpretasi:

- Berdasarkan visualisasi tersebut, setiap variabel memiliki kecondongan yang berbeda-beda.
  - Variabel **Inflasi** memiliki **kecondongan ke kanan (positively skewed)**, dengan sebagian besar data berada di sekitar 0.
  - Variabel **IHK** memiliki **sedikit kecondongan ke kanan (positively skewed)**, dengan sebagian besar data berada di sekitar nilai **100 - 110**.
  - Variabel **'Harga Produk X'** memiliki distribusi yang cenderung normal, dengan **modus di sekitar median**.
  - Variabel **'Harga Produk X (+merek)'** memiliki sedikit **kecondongan ke kanan (positively skewed)**.
  - Variabel **'Produk X'** memiliki sedikit **kecondongan ke kanan (positively skewed)**.
  - Variabel **'Perusahaan Z malang'** memiliki **distribusi** yang cenderung **normal**.
  - Variabel **'Perusahaan Z'** memiliki distribusi dengan 2 puncak, sehingga memiliki **distribusi bimodal**.
  - Variabel **Penjualan** memiliki distribusi yang memiliki **kecondongan ke kanan**, di mana sebagian besar penjualan bernilai rendah.
- Beberapa variabel menunjukkan adanya distribusi yang tidak normal, yang mengindikasikan adanya outlier dan variabilitas tinggi.

# Persiapan Data

# Pengolahan Data

## Eksplorasi Data

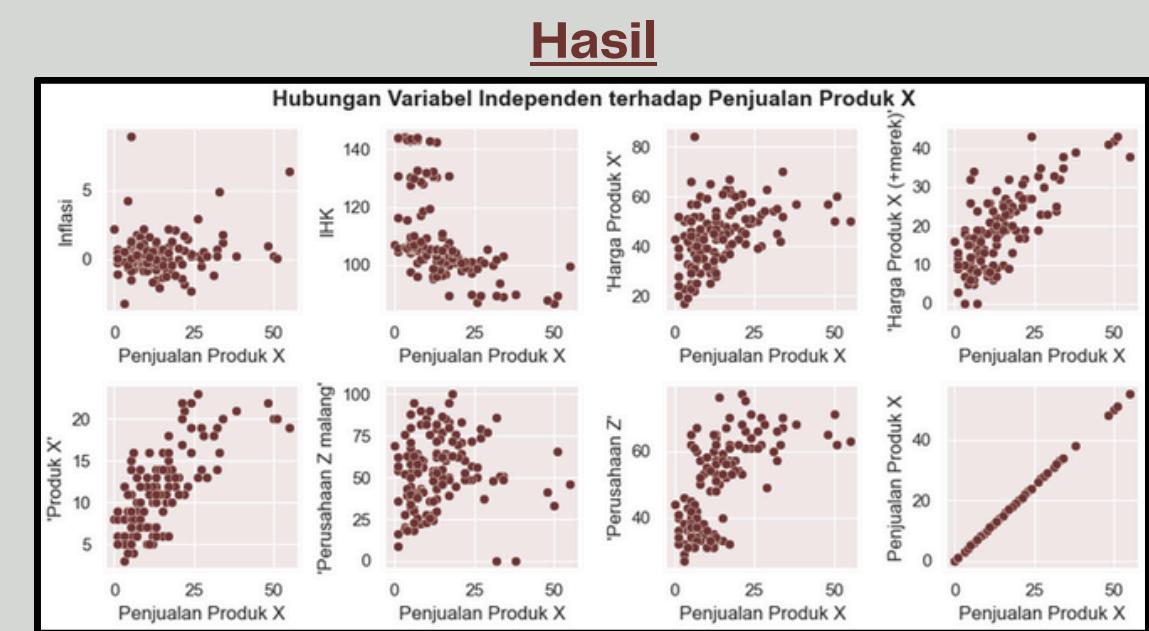
### Visualisasi Data: Distribusi Variabel

Visualisasi distribusi variabel dilakukan dengan syntax berikut:

```
In [41]: fig, axes = plt.subplots(2, 4, figsize=(10,5))
axes=axes.flatten()

for i, col in enumerate(df_cols):
    sns.scatterplot(df_full,
                    x=df_full['Penjualan Produk X'],
                    y=df_full[col],
                    ax=axes[i],
                    color="#6600cc")
    plt.tight_layout()
    axes[i].set_xlabel('Penjualan Produk X')

plt.suptitle("Hubungan Variabel Independen terhadap Penjualan Produk X", fontsize=14, fontweight='bold')
plt.tight_layout(rect=[0,0,1,1.02])
plt.show()
```



### Interpretasi:

- Scatterplot digunakan untuk menggambarkan hubungan antara masing-masing predictor variable terhadap predicted variable.
- Variabel **Inflasi** memiliki **hubungan lemah** dengan variabel penjualan, ditunjukkan dengan **persebaran acak** pada scatterplot-nya.
- Variabel **IHK** terlihat memiliki hubungan **negatif non-linier** dengan variabel penjualan.
- Variabel '**Harga Produk X**' memiliki hubungan **negatif lemah** dengan variabel penjualan.
- Variabel '**Harga Produk X (+merek)**' memiliki hubungan **positif kuat** dengan variabel penjualan.

- Variabel '**Produk X**' memiliki hubungan **positif kuat** dengan variabel penjualan
- Variabel '**Perusahaan Z malang**' memiliki **hubungan lemah** dengan variabel penjualan.
- Variabel '**Perusahaan Z**' memiliki hubungan **positif cukup kuat**, kemungkinan non-linier, dengan variabel penjualan

**Variabel yang memiliki hubungan kuat dengan penjualan adalah variabel IHK, 'Harga Produk X (+merek)', 'Produk X', dan variabel 'Perusahaan Z'.**

# Pengolahan Data

## Eksplorasi Data

### Identifikasi Outlier (Interquartile Range)

Identifikasi outlier dilakukan dengan metode Interquartile Range (IQR) dengan syntax berikut:

```
In [45]: Q1_sales = df_full['Penjualan Produk X'].quantile(0.25)
Q3_sales = df_full['Penjualan Produk X'].quantile(0.75)
IQR_sales = Q3_sales - Q1_sales

outlier_sales = df_full[(df_full['Penjualan Produk X'] > Q3_sales + 1.5*IQR_sales) | 
                       (df_full['Penjualan Produk X'] < Q1_sales - 1.5*IQR_sales)]
outlier_sales[['Month', 'Penjualan Produk X']]
```

```
In [44]: fig, axes = plt.subplots(2, 4, figsize=(10,4))
axes=axes.flatten()

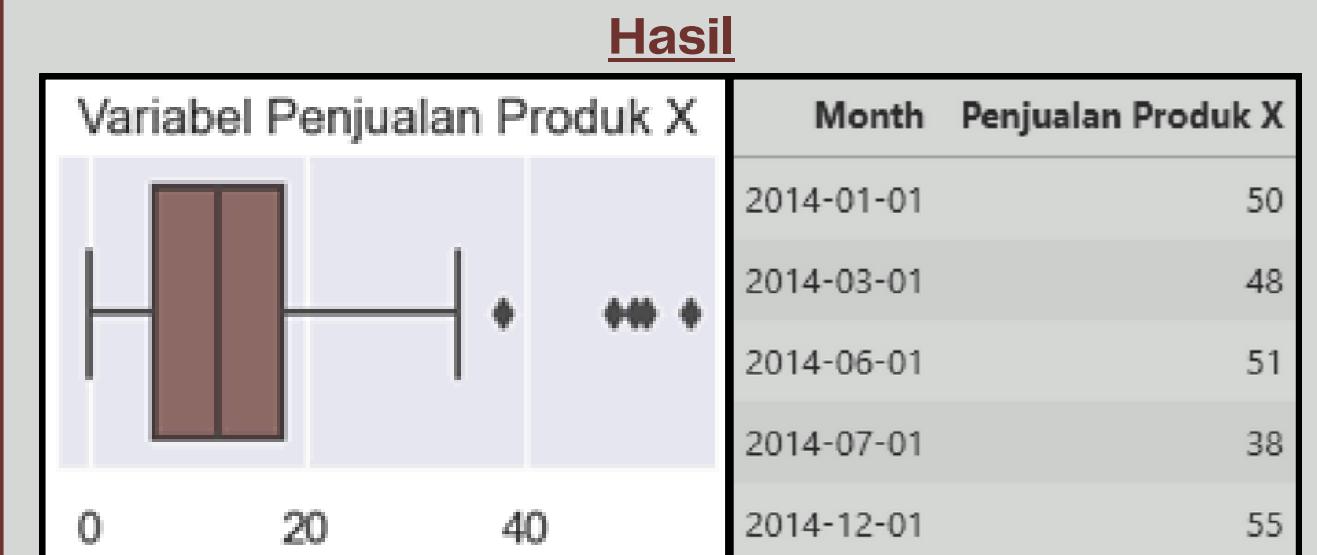
for i, col in enumerate(df_cols):
    sns.boxplot(data=df_full,
                x=col,
                ax=axes[i],
                color="#00ff99")
    axes[i].set_title(f'Variabel {df_cols[i]}')
    axes[i].set_xlabel('')
    plt.tight_layout()

plt.suptitle("Boxplot dari Seluruh Variabel pada Dataset Penjualan Produk X", fontsize=14, fontweight='bold')
plt.tight_layout(rect=[0,0,1,1])
plt.show()
```

Nilai yang berada **di luar**:

- **batas atas ( $Q3 + 1,5 \cdot IQR$ )**; dan
- **batas bawah ( $Q1 - 1,5 \cdot IQR$ )**

akan dianggap sebagai **outlier**.



Proses matematis dengan IQR Method menghasilkan 3 nilai, yaitu:

- **Q1 = 6**
- **Q3 = 17,75**
- **IQR = 11,75**

Berdasarkan proses identifikasi outlier dengan IQR Method dan visualisasi Boxplot, teridentifikasi **5 datapoints** pada penjualan yang merupakan **outlier**.

Outlier-outlier tersebut **tidak akan dihilangkan**, karena **jika dihilangkan**, maka **dapat merusak aspek temporal** dari data karena data bersifat **time-series**.

# Persiapan Data

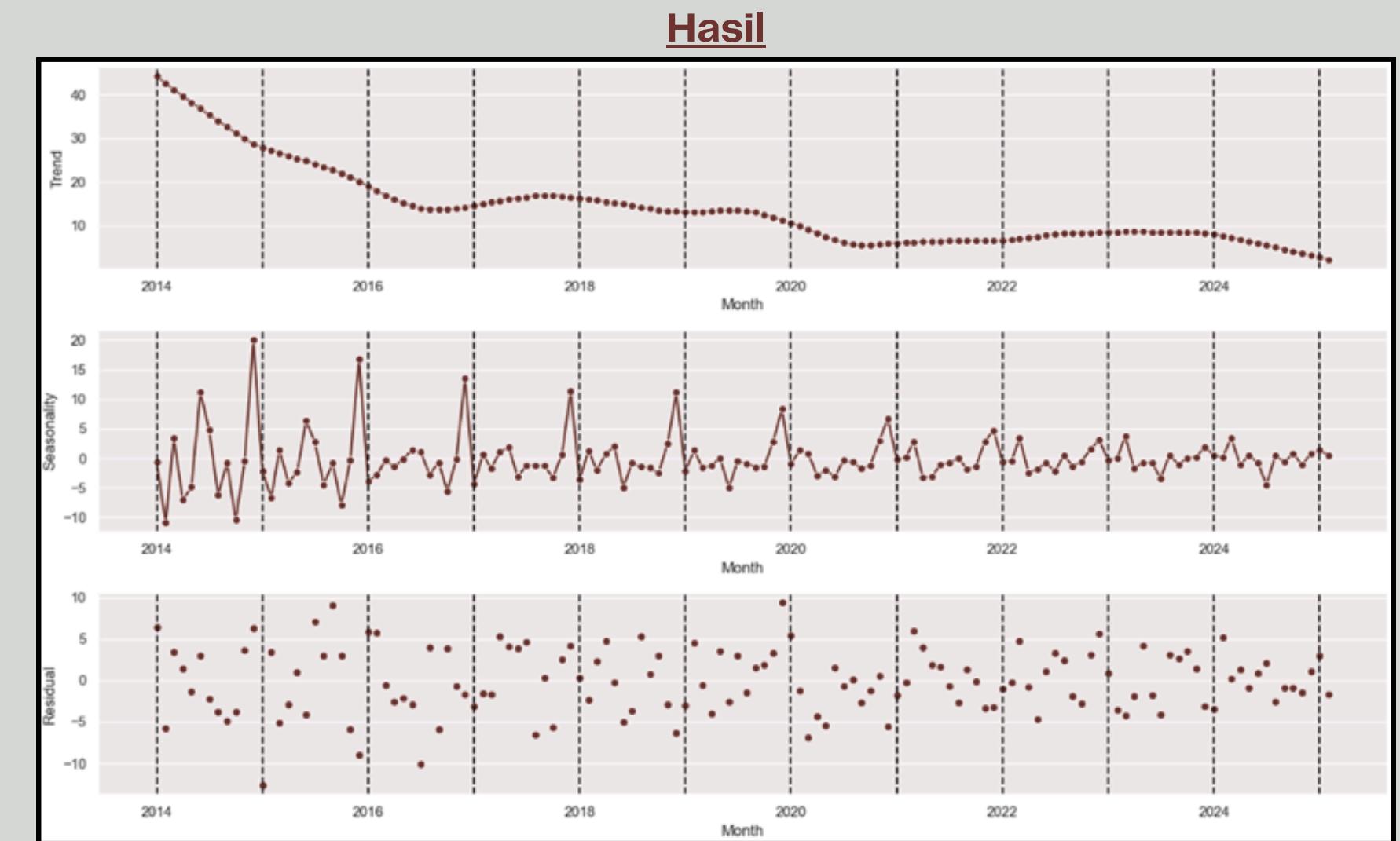
# Pengolahan Data

## Eksplorasi Data

### Dekomposisi (STL)

Dekomposisi dilakukan dengan syntax berikut:

```
In [46]: from statsmodels.tsa.seasonal import STL  
  
In [48]: target = df_full[['Month', 'Penjualan Produk X']].set_index(keys='Month')  
target  
  
In [49]: stl = STL(target, period=12)  
result = stl.fit()  
  
In [50]: seasonal, trend, resid = result.seasonal, result.trend, result.resid  
  
In [51]: stl_element = [target, trend, seasonal]  
  
In [52]: plt.figure(figsize=(15,12))  
ylabel=['Series', 'Trend', 'Seasonality', 'Residual']  
for index, element in enumerate(stl_element):  
    plt.subplot(4,1,index+1)  
    sns.lineplot(element, marker='o')  
    plt.ylabel(ylabel[index])  
  
    for year in range(2014,2026):  
        jan_date = pd.Timestamp(f'{year}-01-01')  
        plt.axvline(jan_date, color='k', linestyle='--')  
  
plt.subplot(4,1,4)  
sns.scatterplot(resid)  
plt.ylabel(ylabel[3])  
for year in range(2014,2026):  
    jan_date = pd.Timestamp(f'{year}-01-01')  
    plt.axvline(jan_date, color='k', linestyle='--')  
  
plt.suptitle("Dekomposisi Data Penjualan Produk X", fontsize=16, fontweight='bold')  
plt.tight_layout(rect=[0, 0, 1, 1])  
plt.show()
```



Dekomposisi menghasilkan 2 informasi berguna dalam analisis time-series, yaitu:

- **Terdapat tren**, yang bersifat menurun
- **Terdapat seasonality** yang cenderung berulang setiap 12 data, atau setiap 12 bulan sekali.

Berdasarkan informasi tersebut, model time-series Holt-Winters' digunakan karena dapat mempertimbangkan tren dan musiman.

# Pengolahan Data

## 02 Pembentukan Model Holt-Winters' *Exponential Smoothing*

- 1. Pemisahan data pelatihan dan pengujian**
- 2. Penetapan model Holt-Winters'**
- 3. Pelatihan model awal**
- 4. Evaluasi model awal**
- 5. Tuning model**
- 6. Evaluasi model**
- 7. Perhitungan residual model**

# Pengolahan Data

## 1. Pemisahan data pelatihan dan pengujian

Data penjualan dipisah menjadi data pelatihan dan data pengujian. Data penjualan pada Januari 2014 – Desember 2023 digunakan sebagai data pelatihan, sedangkan data penjualan pada Januari 2024 – Februari 2025 digunakan sebagai data pengujian

## 2. Penetapan model Holt-Winters'

Model Holt-Winters' (HW) awal ditetapkan sebagai base model untuk pelatihan dan pengujian terhadap data. Model ini akan dikembangkan pada proses-proses selanjutnya.

Model HW awal ditetapkan dengan hyperparameter berikut: (**trend='additive', seasonal='multiplicative', seasonal\_periods=12**)

## 3. Pelatihan model awal

Model HW awal dilatih terhadap data penjualan pelatihan yang telah ditetapkan pada langkah 1.

## 4. Evaluasi model awal

Model HW awal yang telah dilatih dievaluasi dengan melihat pola prediksi model dan performance metric dari model.

Performance metric yang digunakan adalah **Root Mean Squared Error (RMSE)** sebagai indikator utama, dan **Mean Absolute Error (MAE)** sebagai indikator pendukung.

## 5. Tuning model

Tuning dilakukan dengan menetapkan nilai untuk hyperparameter trend dan seasonal menjadi 'additive' atau 'multiplicative', sedangkan hyperparameter seasonal\_periods akan ditetapkan nilainya menjadi 12.

## 6. Evaluasi model

Model dari proses tuning dievaluasi kembali dengan metode yang sama seperti evaluasi sebelumnya. Expanding window forecast dilakukan untuk meningkatkan performa model. Model dengan performa terbaik dipilih sebagai model optimal.

## 7. Perhitungan residual model

Residual model digunakan untuk pelatihan model random forest pada proses selanjutnya.

# Pengolahan Data

# Pemisahan data

**Data Pelatihan:** Data Januari 2014 – Desember 2023

# Data Pengujian: Data Januari 2024 – Februari 2025

```
In [59]: hw_train = df_full['Penjualan Produk X'].iloc[:120].copy()
        hw_test = df_full['Penjualan Produk X'].tail(14).copy().reset_index(drop=True)
```

## Data Pelatihan:

Bulan	Penjualan Produk X
Januari 2014	50
Februari 2014	26
...	...
Desember 2023	7

# Data Pengujian:

Bulan	Penjualan Produk X
Januari 2024	5
Februari 2024	13
...	...
Februari 2025	1

### **Penetapan model Holt-Winters' awal**

Model HW awal yang digunakan adalah model dengan **additive trend** dan **multiplicative seasonality**.

Proses pelatihan model HW awal dilakukan dengan syntax di samping.

# Pengolahan Data

## Pelatihan model awal

Parameter dari model yang telah dilatih, diperoleh melalui syntax berikut:

```
print("Smoothing Parameters")
print("Alpha (level):", hw_model_awal.params['smoothing_level'])
print("Beta (trend):", hw_model_awal.params.get('smoothing_trend'))
print("Gamma (seasonal):", hw_model_awal.params.get('smoothing_seasonal'))

print("\nInitial State")
print("Initial level:", hw_model_awal.params['initial_level'])
print("Initial trend:", hw_model_awal.params.get('initial_trend'))
print("Initial seasonal:", hw_model_awal.params.get('initial_seasonal'))
```

Parameter model HW awal adalah sebagai berikut:

$$L_t = 0,3308 \left( \frac{Y_t}{S_{t-s}} \right) + 0,6692(L_{t-1} + b_{t-1})$$

$$b_t = 0,0244(L_t - L_{t-1}) + 0,9756(b_{t-1})$$

$$S_t = 0,0796 \left( \frac{Y_t}{L_t} \right) + 0,9204(S_{t-s})$$

## Evaluasi model awal

Model HW awal digunakan untuk memprediksi data secara **in-sample (data pelatihan)** dan **out-of-sample (data pengujian)** melalui syntax berikut:

```
In [72]: hw_awal_insample = hw_model_awal.predict(start=0,
                                                end=len(hw_train) - 1)

hw_awal_outsample = hw_model_awal.predict(start=len(hw_train),
                                           end = len(hw_train)
                                                 + len (hw_test)
                                                 - 1)
```

Evaluasi dilakukan dengan **performance metric** berupa **Root Mean Squared Error (RMSE)** dan **Mean Absolute Error (MAE)**, dengan syntax berikut:

```
In [96]: HW_ori_ins_rmse = mean_squared_error(hw_train, hw_awal_insample, squared=False)
HW_ori_ins_mae = mean_absolute_error(hw_train, hw_awal_insample)

HW_ori_outs_rmse = mean_squared_error(hw_test, hw_awal_outsample, squared=False)
HW_ori_outs_mae = mean_absolute_error(hw_test, hw_awal_outsample)
```

Di bawah ini adalah hasil evaluasi model HW awal.

Metric	In-Sample	Out-of-Sample
RMSE	5,475	3,627
MAE	4,447	3,043

Model memiliki performa out-of-sample yang tidak buruk, tetapi dapat ditingkatkan lebih lagi melalui proses **hyperparameter tuning**.

# Pengolahan Data

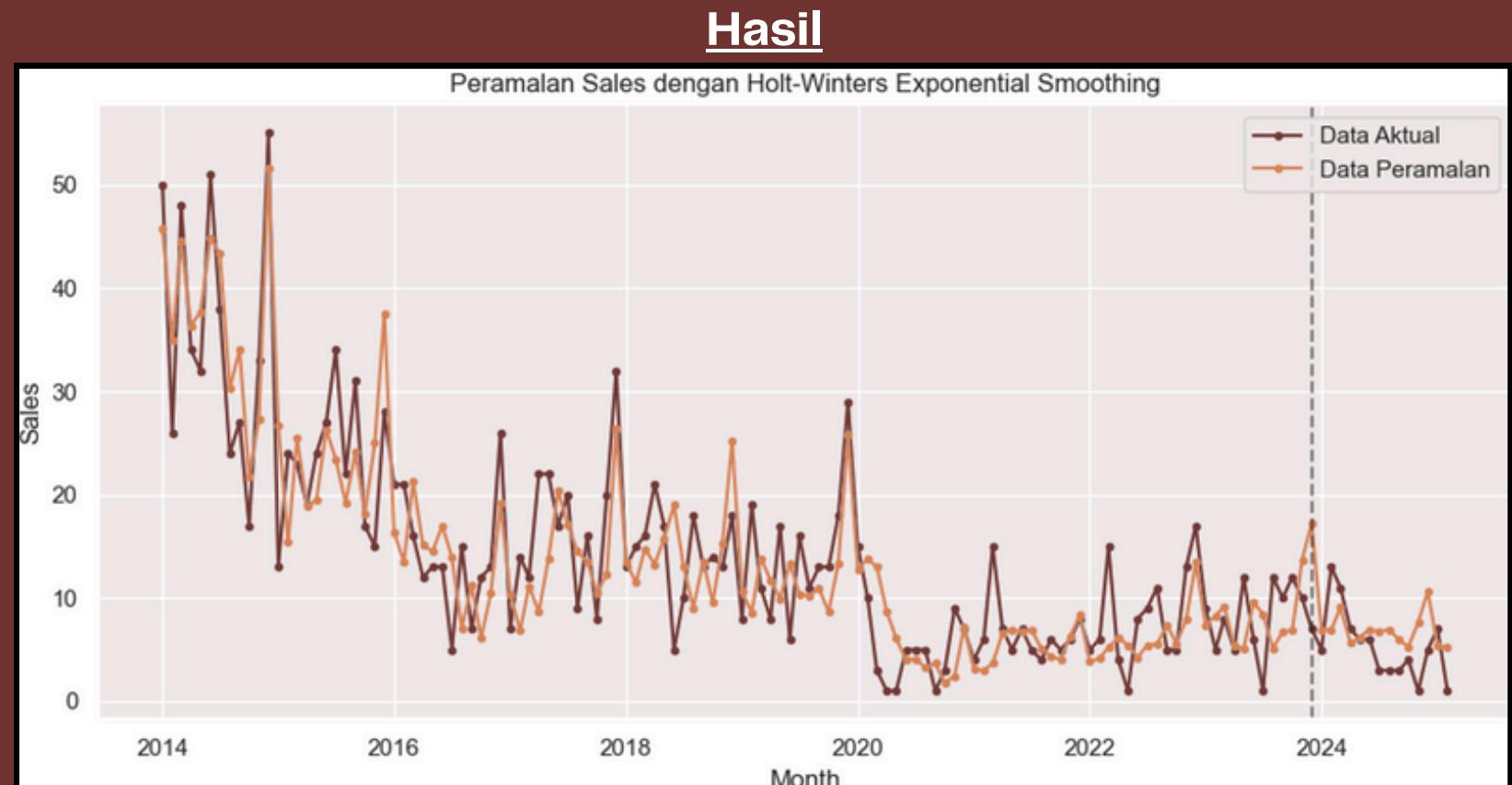
## Pembentukan Model Holt-Winters' ES

### Evaluasi model awal

Evaluasi juga dilakukan dengan melihat pola peramalan dari model menggunakan visualisasi garis. Visualisasi dilakukan dengan syntax berikut.

```
In [97]: plt.figure(figsize=(10,5))
plt.plot(sales, label='Data Aktual', marker='.')
plt.plot(pd.concat([hw_awal_insample, hw_awal_outsample]),
         label='Data Peramalan', marker='.')

plt.axvline(pd.Timestamp('2023-12-01'), linestyle='--',
            color='k', alpha=0.5)
plt.title("Peramalan Sales dengan Holt-Winters Exponential Smoothing")
plt.xlabel('Month')
plt.ylabel("Sales")
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()
```



Garis putus-putus menunjukkan batas periode antara data pelatihan dan data pengujian.

Model HW awal terlihat mampu meng-capture fluktuasi dari data penjualan aktual. Namun, terdapat **gap** yang cukup besar antara performa **in-sample** dan **out-of-sample**-nya, yang mampu menandakan bahwa **model memiliki kemampuan generalisasi yang kurang baik**.

# Pengolahan Data

## Tuning model

Tuning dilakukan dengan mencoba segala kombinasi hyperparameter dari model Holt-Winters. Kombinasi yang mungkin diuji adalah sebagai berikut:

- 1.trend = 'additive', seasonal = 'additive'
- 2.trend = 'additive', seasonal = 'multiplicative'
- 3.trend = 'multiplicative', seasonal = 'additive'
- 4.trend = 'multiplicative', seasonal = 'multiplicative'

## Evaluasi model

Evaluasi dilakukan dengan memilih kombinasi hyperparameter dengan nilai **RMSE out-of-sample** terendah. Performa terbaik diperoleh oleh model dengan kombinasi **trend = 'multiplicative', seasonal = 'multiplicative'**.

Nilai performance metric dari model tersebut adalah sebagai berikut.

Metrik	Tuned Model
RMSE	3,048
MAE	2,364

## Penghitungan residual model

Residual merupakan selisih antara nilai aktual dengan nilai peramalan.

$$r_t = y_t - \hat{y}_t$$

Dimana:

$r_t$  = residual pada observasi ke- $t$

$y_t$  = nilai aktual pada observasi ke- $t$

$\hat{y}_t$  = nilai peramalan pada observasi ke- $t$

Bulan	Aktual	Peramalan	Residual
Januari 2014	50	47	3
Februari 2014	26	34	-8
...	...	...	...
Januari 2025	7	3	4
Februari 2025	1	4	-3

Residual akan digunakan untuk **melatih model Random Forest**.

# Pengolahan Data

## 03 Pembentukan Model *Random Forest*

- 1. Pemilihan variabel prediktor**
- 2. Pemisahan data pelatihan dan pengujian**
- 3. Inisiasi model awal**
- 4. Pelatihan model awal**
- 5. Evaluasi model awal**
- 6. Tuning model**
- 7. Evaluasi model**

# Pengolahan Data

## 1. Pemilihan variabel prediktor

Variabel prediktor adalah variabel yang digunakan untuk memprediksi variabel target, yaitu residual model HW.

Proses ini melibatkan tahap pelatihan model awal, pemeriksaan kepentingan relatif fitur, dan pemeriksaan Mean Absolute SHAP Value.

## 2. Pemisahan data pelatihan dan pengujian

Dataset (dengan fitur yang telah diseleksi) pada Januari 2014 – Desember 2023 digunakan sebagai data pelatihan, dan dataset pada Januari 2024 – Februari 2025 digunakan sebagai data pengujian.

## 3. Inisiasi model awal

Model awal yang digunakan sebagai base model adalah model random forest dengan 100 pohon keputusan ( $n\_estimators=100$ ).

## 4. Pelatihan model awal

Model awal dilatih terhadap data pelatihan.

## 5. Evaluasi model awal

Evaluasi dilakukan secara in-sample dan out-of-sample dengan melihat pola prediksi dan MAE dari model.

## 6. Tuning model

Hyperparameter terbaik dari model random forest ditemukan dengan metode Grid Search dan dilanjut dengan trial-and-error untuk menemukan model terbaik, yakni model dengan MAE terkecil, serta selisih performa in-sample dan out-of-sample terkecil.

Tuning dilakukan dengan menentukan parameter grid untuk masing-masing hyperparameter, kemudian dilakukan grid search untuk menemukan kombinasi dengan potensi performa terbaik, dan dilanjut dengan trial-and-error untuk menemukan model dengan performa yang lebih baik.

## 7. Evaluasi model

Evaluasi model dilakukan kembali dengan membandingkan performa model hasil tuning dengan model awal.

# Pengolahan Data

## Pemilihan variabel prediktor

### Fitting model awal

Model random forest (RF) awal yang digunakan adalah model dengan 100 pohon keputusan (n\_estimators=100). Berikut proses fitting model RF awal terhadap dataset pelatihan.

```
In [139]: RF = RandomForestRegressor(n_estimators=100, random_state=42)
RF.fit(X_RF_train, Y_RF_train)
```

### Kepentingan Fitur (Feature Importance)

Untuk melihat **10 fitur dengan nilai feature importance tertinggi**, dijalankan syntax sebagai berikut:

```
In [145]: plt.figure(figsize=(10, 6))
ax = sns.barplot(data=feature_importance_RF.head(10),
                  x='RF Importance',
                  y='Feature',
                  palette="Blues_r")

for index, value in enumerate(feature_importance_RF.head(10)[ "RF Importance"]):
    ax.text(value, index, f"{value:.3f}", va='center', fontsize=10)

plt.axvline(0.02, linestyle='--', c='r')
plt.axvline(0.05, linestyle='--', c='g')
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Feature Importance from Random Forest")
plt.show()
```

### Mean Absolute SHAP Value (MASV)

Untuk melihat **10 fitur dengan nilai MASV tertinggi**, dijalankan syntax sebagai berikut:

```
In [147]: explainer = shap.TreeExplainer(RF)
shap_values = explainer.shap_values(X_RF_train)

In [148]: shap_df = pd.DataFrame(shap_values, columns=X_RF_train.columns)
mean_abs_shap = shap_df.abs().mean().sort_values(ascending=False)
mean_abs_shap.head(10)

In [154]: plt.figure(figsize=(10, 6))
ax = sns.barplot(data=mean_abs_shap.head(10),
                  x='MASV',
                  y='Feature',
                  palette="Greens_r")

for index, value in enumerate(mean_abs_shap.head(10)[ "MASV"]):
    ax.text(value, index, f"{value:.3f}", va='center', fontsize=10)

plt.xlabel("MASV")
plt.ylabel("Features")
plt.title("Mean Absolute SHAP Value")
plt.show()
```

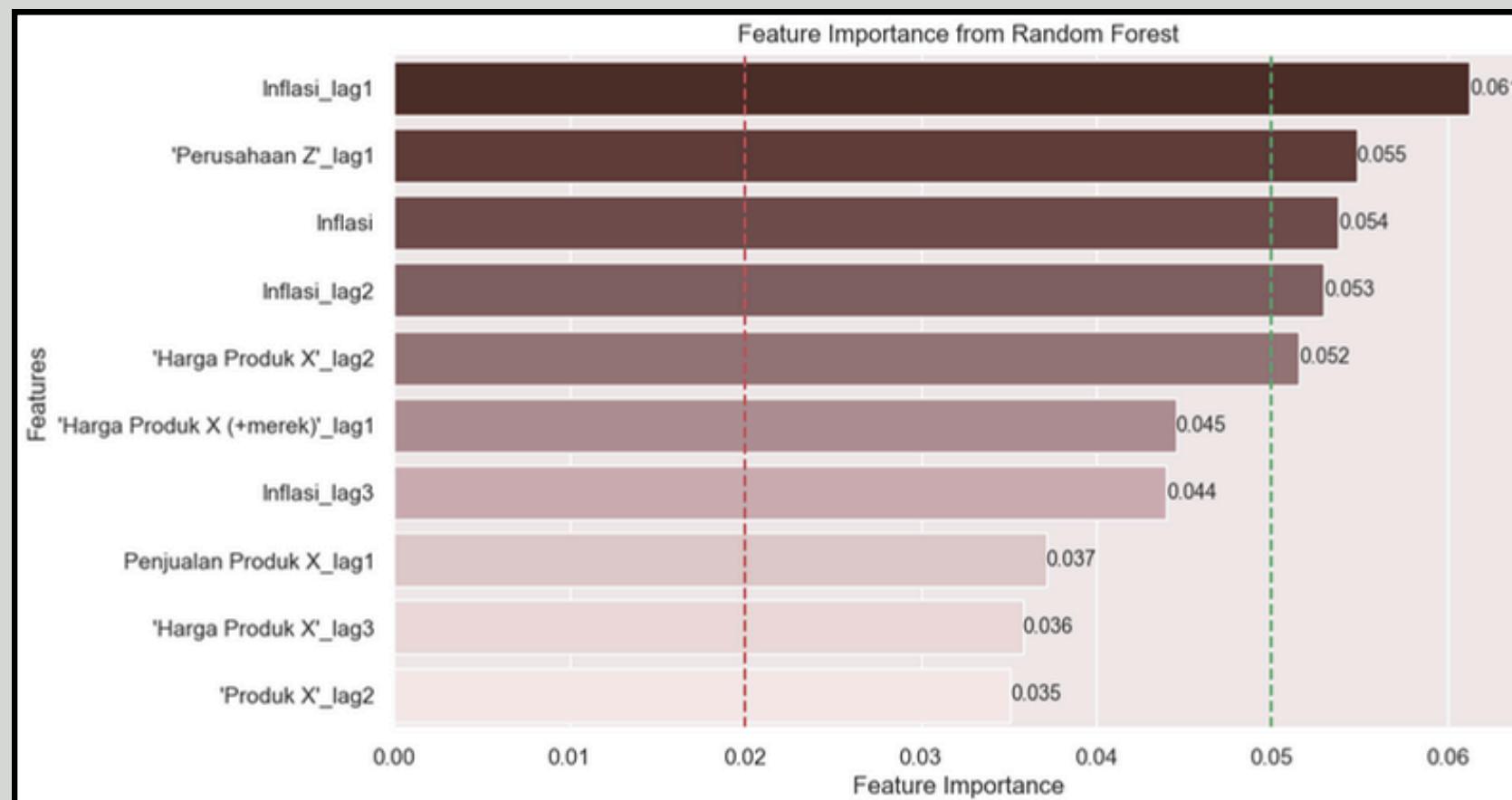
# Pengolahan Data

## Pembentukan Model Random Forest

### Pemilihan variabel prediktor

#### Kepentingan Fitur (Feature Importance)

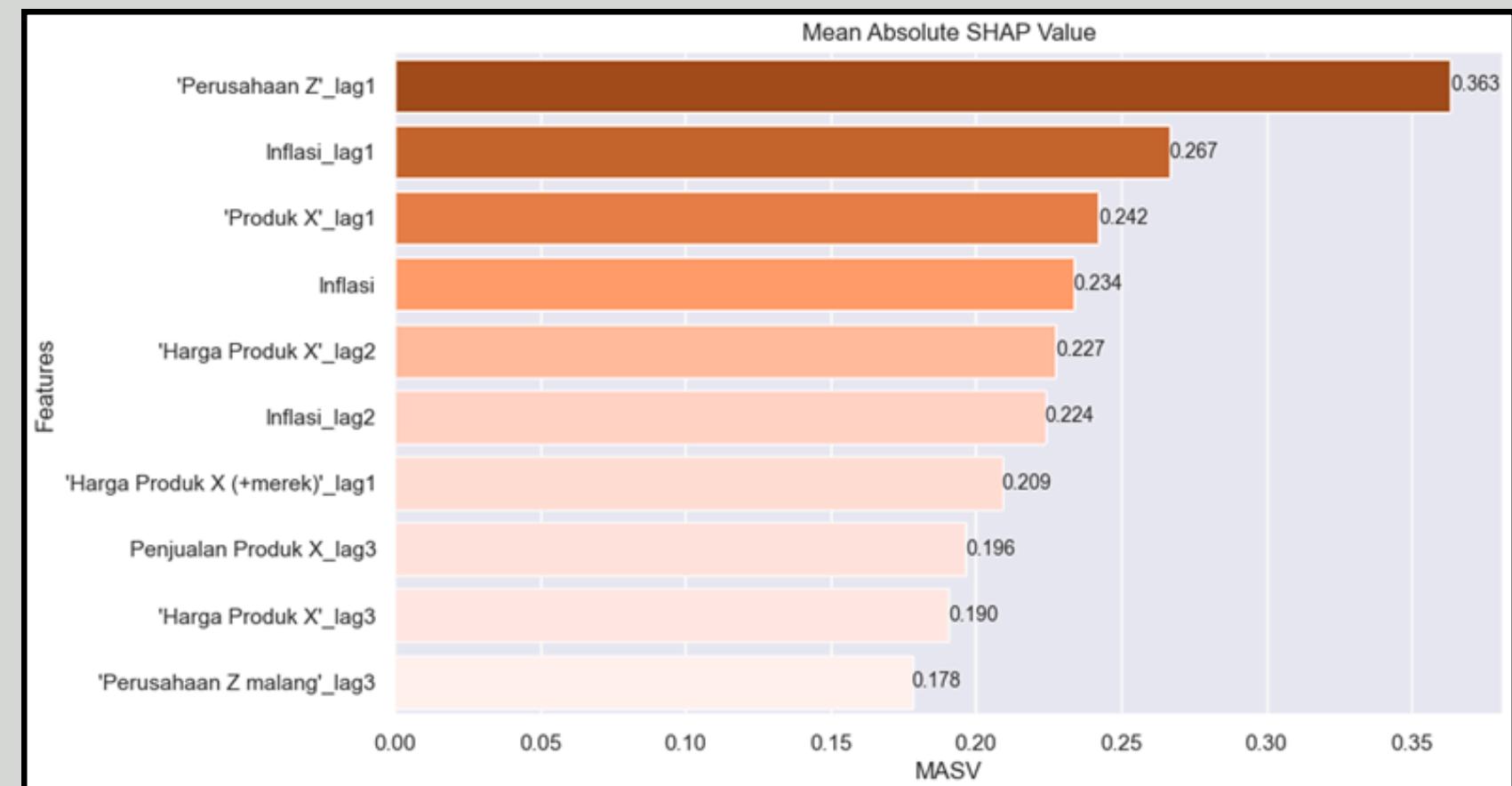
Hasil dari visualisasi feature importance adalah sebagai berikut.



Variabel-varibel di atas merupakan 10 variabel dengan nilai kepentingan relatif tertinggi. **Variabel Inflasi\_lag1 memiliki nilai kepentingan tertinggi, yakni sebesar 0,061.**

#### Mean Absolute SHAP Value (MASV)

Hasil dari visualisasi MASV fitur adalah sebagai berikut.



Variabel-varibel di atas merupakan 10 variabel dengan MASV tertinggi. **Variabel 'Perusahaan Z'\_lag1 memiliki nilai kontribusi global tertinggi, yakni sebesar 0,363.**

# Pembentukan Model Random Forest

# Pengolahan Data

## Pemisahan data

### Data Pelatihan

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag3	Residual HW
2014-01	72	17	0,52	...	32	3
2014-02	71	33	0,25	...	38	-8
2014-03	66	55	0,22	...	41	3
2014-04	65	50	0,98	...	50	-1
2014-05	67	26	1,20	...	50	-1
...	...	...	...	...	...	...
2023-08	34	12	0,48	...	34	7
2023-09	35	6	0,09	...	31	4
2023-10	36	1	0,65	...	28	6
2023-11	31	12	0,59	...	45	-2
2023-12	31	10	-0,30	...	28	-9

### Data Pengujian

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag3	Residual HW
2024-01	34	12	0,43	...	30	-2
2024-02	34	10	-0,85	...	32	7
2024-03	34	7	-0,06	...	35	1
2024-04	33	5	0,17	...	29	0
2024-05	33	13	0,18	...	29	-1
2024-06	31	11	0,12	...	25	-2
2024-07	31	7	0,06	...	40	-4
2024-08	32	6	0,07	...	25	-3
2024-09	29	6	0,52	...	22	-1
2024-10	27	3	-0,17	...	23	1
2024-11	31	3	-0,45	...	17	-4
2024-12	32	3	0,05	...	23	0
2025-01	37	4	0,06	...	19	4
2025-02	37	1	0,28	...	20	-3

## Inisiasi model awal



## Pelatihan model awal

Pelatihan model RF awal (`n_estimators=100`) setelah proses feature selection dilakukan dengan menjalankan syntax berikut.

```
In [193]: RF_res = RandomForestRegressor(n_estimators=100, random_state=42)
RF_res.fit(X_RF_train_red, Y_RF_train)
```

## Evaluasi model awal

Evaluasi model RF dilakukan dengan **melihat pola peramalan** dan **MAE dari model**, seperti berikut:

- Pola Peramalan Model

```
In [199]: plt.figure(figsize=(10,4))
plt.xticks(rotation=90)
sns.lineplot(data=RF_res_ins_df, marker='o')
plt.title('In Sample Actual vs Predicted Residual Value using Random Forest')
plt.show()
```

```
In [201]: plt.figure(figsize=(10,4))
plt.xticks(rotation=90)
sns.lineplot(data=RF_res_outs_df, marker='o')
plt.title('Out of Sample Actual vs Predicted Residual Value using Random Forest'
plt.show()
```

- Mean Absolute Error (MAE) Model

```
In [ ]: RF_res_ins_mae = mean_absolute_error(RF_res_ins_df['Y_Actual'], RF_res_ins_df['Y_Predicted'])
RF_res_outs_mae = mean_absolute_error(RF_res_outs_df['Y_Actual'], RF_res_outs_df['Y_Predicted'])
```

# Pembentukan Model Random Forest

# Pengolahan Data

## Pemisahan data

### Data Pelatihan

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag3	Residual HW
2014-01	72	17	0,52	...	32	3
2014-02	71	33	0,25	...	38	-8
2014-03	66	55	0,22	...	41	3
2014-04	65	50	0,98	...	50	-1
2014-05	67	26	1,20	...	50	-1
...	...	...	...	...	...	...
2023-08	34	12	0,48	...	34	7
2023-09	35	6	0,09	...	31	4
2023-10	36	1	0,65	...	28	6
2023-11	31	12	0,59	...	45	-2
2023-12	31	10	-0,30	...	28	-9

### Data Pengujian

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag3	Residual HW
2024-01	34	12	0,43	...	30	-2
2024-02	34	10	-0,85	...	32	7
2024-03	34	7	-0,06	...	35	1
2024-04	33	5	0,17	...	29	0
2024-05	33	13	0,18	...	29	-1
2024-06	31	11	0,12	...	25	-2
2024-07	31	7	0,06	...	40	-4
2024-08	32	6	0,07	...	25	-3
2024-09	29	6	0,52	...	22	-1
2024-10	27	3	-0,17	...	23	1
2024-11	31	3	-0,45	...	17	-4
2024-12	32	3	0,05	...	23	0
2025-01	37	4	0,06	...	19	4
2025-02	37	1	0,28	...	20	-3

## Inisiasi model awal

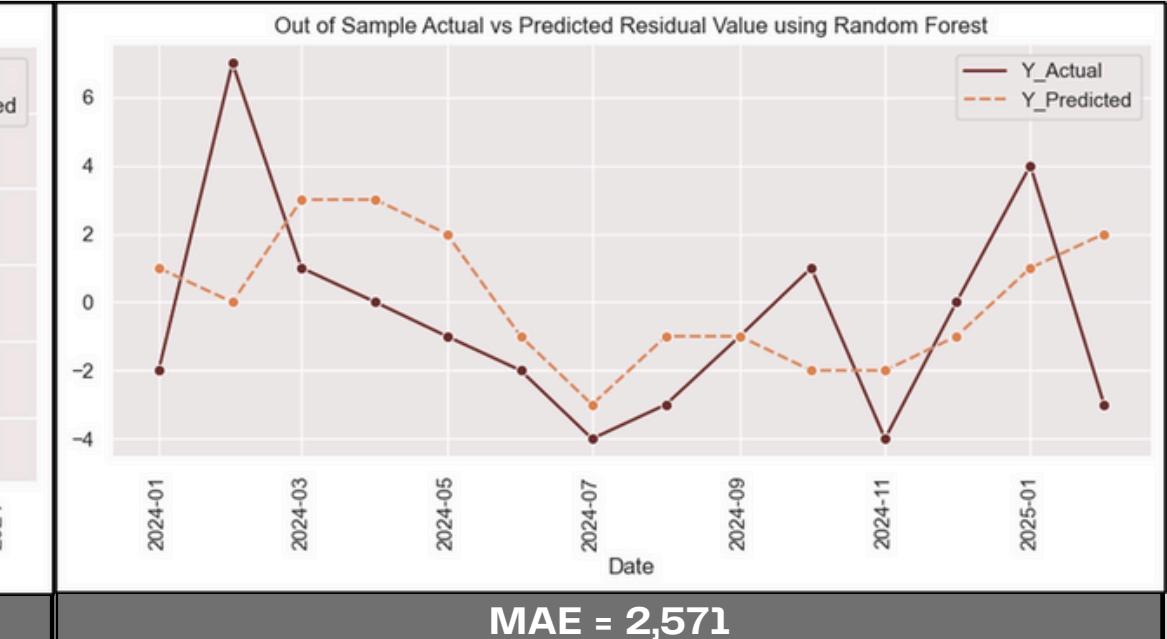
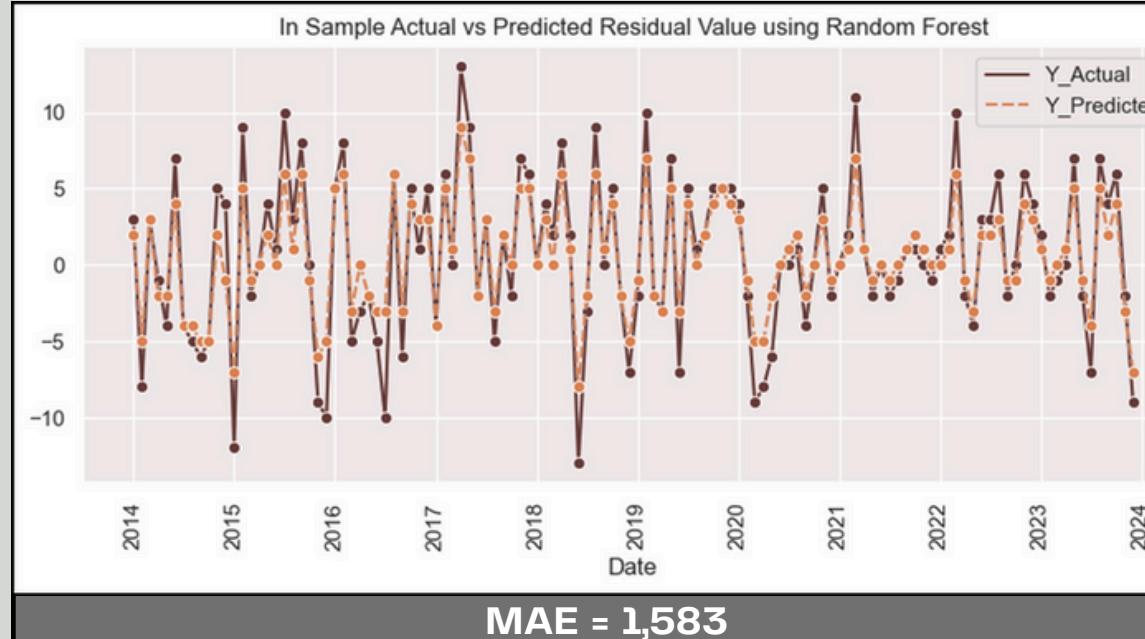


## Pelatihan model awal

Pelatihan model RF awal (`n_estimators=100`) setelah proses feature selection dilakukan dengan menjalankan syntax berikut.

```
In [193]: RF_res = RandomForestRegressor(n_estimators=100, random_state=42)
RF_res.fit(X_RF_train_red, Y_RF_train)
```

## Evaluasi model awal



Model mengindikasikan adanya overfitting dan masih bisa ditingkatkan lagi.

# Pengolahan Data

## Tuning model (Grid Search + Trial and Error)

```
In [214]: param_grid = {'n_estimators': [2, 50, 100, 150, 200, 250],
                     'max_depth': [5, 10, 15, 20],
                     'min_samples_split': [2, 4, 6, 8, 10],
                     'min_samples_leaf': [1, 2, 3, 4, 5],
                     'max_features': [1, 3, 5, 10]}

best_mae = float('inf')
best_params = None
results = []

for params in ParameterGrid(param_grid):
    RF_GS_model = RandomForestRegressor(**params, random_state=42)
    RF_GS_model.fit(X_RF_train_red, Y_RF_train)

    RF_GS_insample = RF_GS_model.predict(X_RF_train_red)
    RF_GS_outsample = RF_GS_model.predict(X_RF_test_red)

    ins_mae = mean_absolute_error(Y_RF_train, RF_GS_insample)
    outs_mae = mean_absolute_error(Y_RF_test, RF_GS_outsample)

    results.append({'params': params,
                   'insample_mae': ins_mae,
                   'outsample_mae': outs_mae})

    if outs_mae < best_mae:
        best_mae = outs_mae
        best_params = params

print(f"Best parameters: {best_params}")
print(f"Best out-of-sample MAE: {best_mae}")
```

### Hasil Grid Search

Hyperparameter	Value
n_estimators	50
max_depth	5
min_samples_split	8
min_samples_leaf	1
max_features	5

### Performance

Metrik	RF GridSearch	
	In-Sample	Out-of-Sample
MAE	2,867	2,143

Model hasil Grid Search memiliki performa out-of-sample yang cukup baik, tetapi memiliki nilai in-sample yang lebih tinggi, sehingga **gap in-sample dengan out-of-sample menjadi lebih besar**. Hal ini mengindikasikan **model memiliki performa yang kurang stabil**.

### Hasil Grid Search + Trial and Error

Hyperparameter	Value
n_estimators	65
max_depth	13
min_samples_split	8
min_samples_leaf	1
max_features	2

### Performance

Metrik	RF GridSearch	
	In-Sample	Out-of-Sample
MAE	2,333	2,0

Setelah di-adjust, model memiliki performa in-sample dan out-of-sample yang lebih baik, serta **gap yang lebih kecil**, sehingga **model ini dapat disimpulkan memiliki performa yang lebih baik dibandingkan dengan model Grid Search sebelum di-adjust**.

# Pembentukan Model Random Forest

# Pengolahan Data

## Evaluasi model

Evaluasi model dilakukan dengan menjalankan syntax berikut.

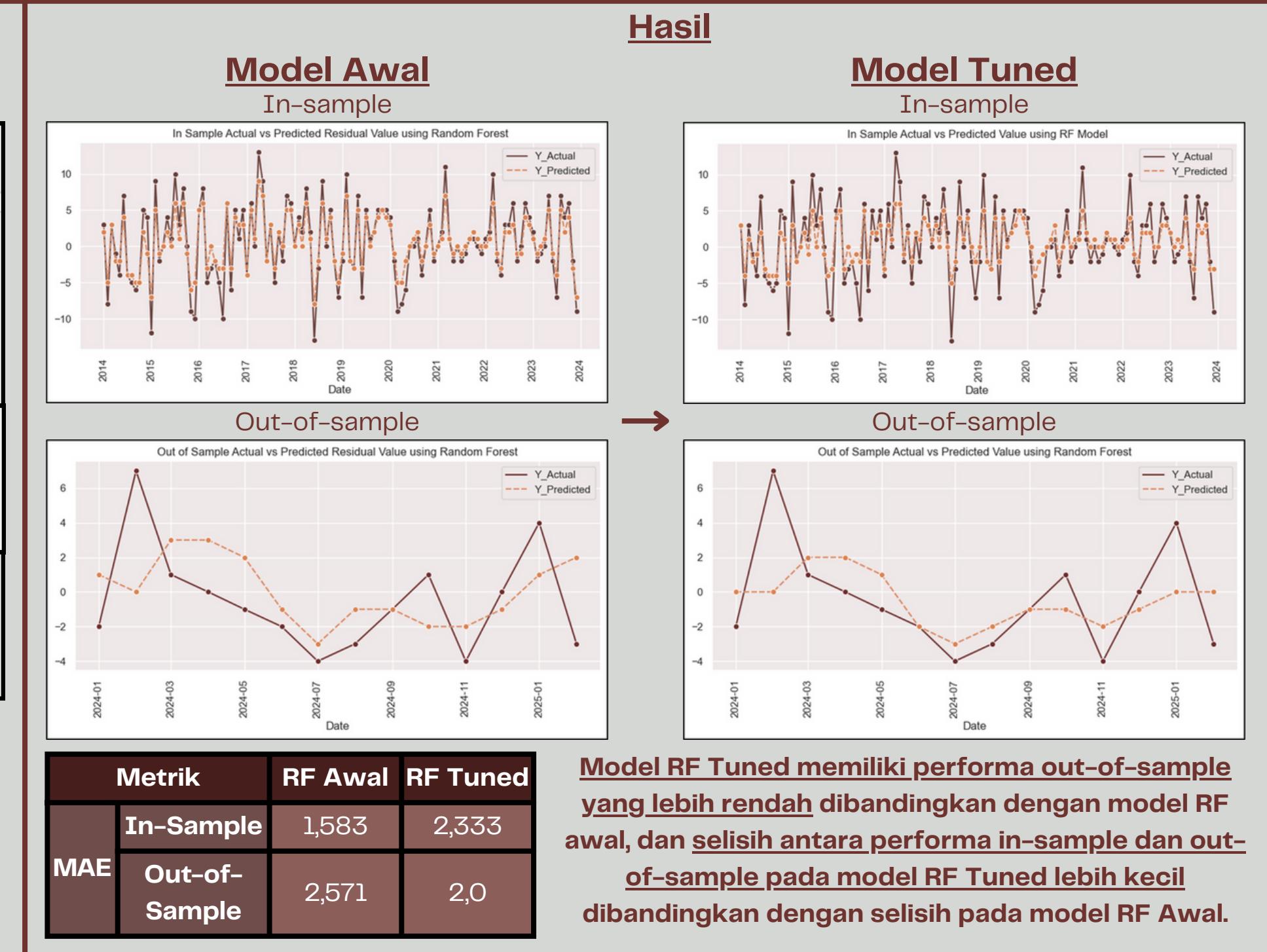
```
In [227]: RF_opt_ins = RF_opt.predict(X_RF_train_red)
RF_opt_outs = RF_opt.predict(X_RF_test_red)

In [228]: RF_opt_ins_df = pd.DataFrame({'Y_Actual': Y_RF_train,
                                     'Y_Predicted': RF_opt_ins,
                                     'Date': train_date})

RF_opt_outs_df = pd.DataFrame({'Y_Actual': Y_RF_test,
                               'Y_Predicted': RF_opt_outs,
                               'Date': test_date})

In [232]: plt.figure(figsize=(10,4))
plt.xticks(rotation=90)
sns.lineplot(data=RF_opt_ins_df, marker='o')
plt.title('In Sample Actual vs Predicted Residual Value using Random Forest')
plt.show()

In [232]: plt.figure(figsize=(10,4))
plt.xticks(rotation=90)
sns.lineplot(data=RF_opt_outs_df, marker='o')
plt.title('Out of Sample Actual vs Predicted Value using Random Forest')
plt.show()
```



# Pengolahan Data

## 04 Pembentukan Model Hibrida

- 1. Peramalan penjualan oleh model Holt-Winters'**
- 2. Peramalan residual oleh model Random Forest**
- 3. Penghitungan nilai peramalan akhir**
- 4. Evaluasi model hibrida**

# Pengolahan Data

## **1. Peramalan penjualan oleh model Holt-Winters'**

Model Holt-Winters' optimal pada tahap pembentukan model digunakan untuk peramalan data penjualan pelatihan dan pengujian pada tahap ini.

Setelah peramalan dilakukan, dihitung nilai residual untuk digunakan dalam pelatihan model random forest.

---

## **2. Peramalan residual oleh model Random Forest**

Nilai residual dan variabel prediktor dimasukkan sebagai input ke dalam model random forest optimal (`n_estimators=65, max_depth=13, min_samples_split=8, min_samples_leaf=1, max_features=2`) untuk diramalkan.

---

## **3. Penghitungan nilai peramalan akhir**

Nilai peramalan akhir diperoleh melalui penjumlahan nilai peramalan penjualan dari model Holt-Winters' dengan nilai peramalan residual dari model random forest pada setiap observasi.

---

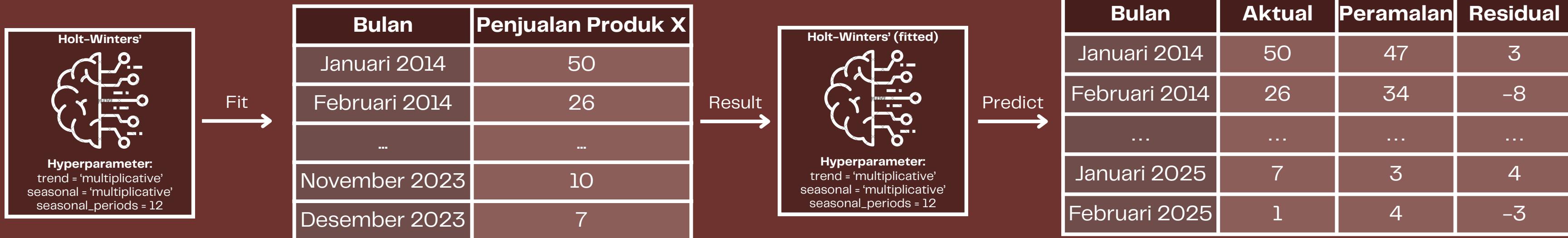
## **4. Evaluasi model hibrida**

Evaluasi model dilakukan dengan melihat pola peramalan dan metrik performa model (RMSE & MAE).

Selain itu, dilakukan juga perbandingan antara performa model hibrida terhadap model Holt-Winters' tunggal.

# Pengolahan Data

## Peramalan penjualan oleh model Holt-Winters'



## Peramalan residual oleh model Random Forest

Syntax di bawah ini dijalankan untuk melakukan peramalan dengan model Random Forest.

```
In [239]: RF_model = RandomForestRegressor(n_estimators=65, max_depth=13,
                                         min_samples_split=8, min_samples_leaf=1,
                                         max_features=2, random_state=42)

In [240]: RF_model.fit(X_RF_train_red, Y_RF_train)
```

```
In [241]: RF_insample = RF_model.predict(X_RF_train_red)
RF_outsample = RF_model.predict(X_RF_test_red)

In [242]: RF_insample_df = pd.DataFrame({'Y_Predicted': RF_insample,
                                         'Date': train_date})

RF_outsample_df = pd.DataFrame({'Y_Predicted': RF_outsample,
                                 'Date': test_date})
```

# Pengolahan Data

## Penghitungan nilai peramalan akhir

Peramalan Penjualan Holt-Winters' + Peramalan Residual Random Forest

Coded →

```
In [248]: hw_rf_ins_df = hw_insample_df[['Y_Actual']]
hw_rf_outs_df = hw_outs_exp_df[['Y_Actual']]

In [249]: hw_rf_ins_df['Y_Predicted'] = hw_insample_df['Y_Predicted']
           + RF_insample_df['Y_Predicted']

           hw_rf_outs_df['Y_Predicted'] = hw_outs_exp_df['Y_Predicted']
           + RF_outsample_df['Y_Predicted']
```

## Evaluasi model hibrida

Evaluasi model hibrida dilakukan dengan syntax berikut.

```
In [251]: plt.figure(figsize=(10,4))
sns.lineplot(hw_rf_ins_df, marker='o', dashes=False)

plt.title('Actual vs Predicted In Sample using Hybrid HW + RF Model',
          fontweight='bold')
plt.tight_layout()
plt.show()
```

```
In [253]: plt.figure(figsize=(10,4))
sns.lineplot(hw_rf_outs_df, marker='o', dashes=False)

plt.title('Actual vs Predicted Out of Sample using Hybrid HW + RF Model',
          fontweight='bold')
plt.tight_layout()
plt.show()
```

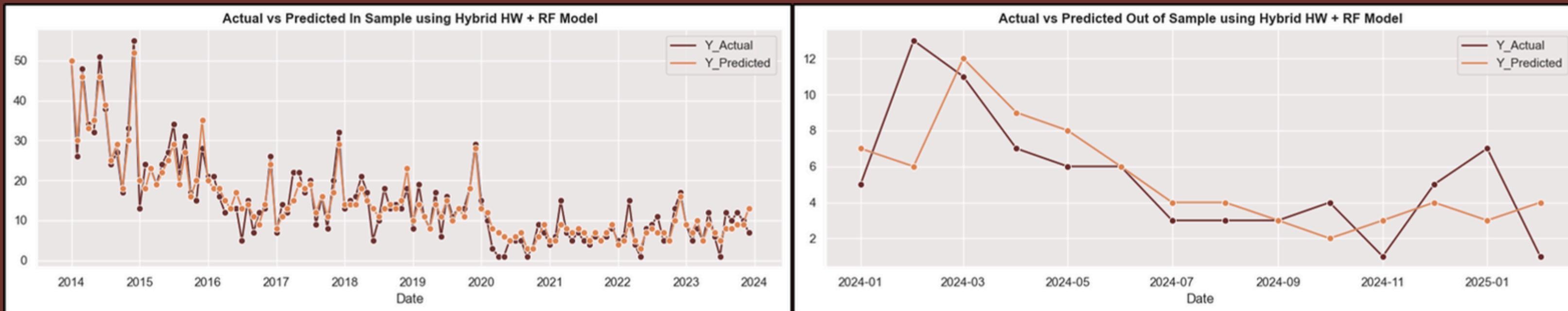
# Pengolahan Data

## Penghitungan nilai peramalan akhir (Hasil)

Peramalan Penjualan Holt-Winters' + Peramalan Residual Random Forest

Bulan	Peramalan Penjualan Holt-Winters'	Peramalan Residual Random Forest	Peramalan Model Hibrida
Januari 2014	47	3	50
Februari 2014	34	-4	30
...	...	...	...
Januari 2025	3	0	3
Februari 2025	4	0	4

## Evaluasi model hibrida (Hasil)



# Pengolahan Data

## Penghitungan nilai peramalan akhir (Hasil)

Peramalan Penjualan Holt-Winters' + Peramalan Residual Random Forest

Bulan	Peramalan Penjualan Holt-Winters'	Peramalan Residual Random Forest	Peramalan Model Hibrida
Januari 2014	47	3	50
Februari 2014	34	-4	30
...	...	...	...
Januari 2025	3	0	3
Februari 2025	4	0	4

## Evaluasi model hibrida (Hasil)

Metrik		Holt-Winters'	Hibrida	%Improvement
RMSE	In-Sample	5,287	3,017	42,9%
	Out-of-Sample	3,048	2,646	13,2%
MAE	In-Sample	4,268	2,333	45,3%
	Out-of-Sample	2,364	2,0	15,3%

Model hibrida mampu meningkatkan performa model sebesar 13,2% (RMSE out-of-sample).

Model hibrida juga terlihat mampu meningkatkan stabilitas model, terlihat dari selisih skor in-sample dan out-of-sample yang menurun.

$$\begin{array}{ccc} \text{Performance Gap} & & \\ |5,287 - 3,048| & \longrightarrow & |3,017 - 2,646| \\ = 2,239 & & = 0,371 \end{array}$$

# Pengolahan Data

## 05 Peramalan Penjualan Tahun 2025

- 1. Persiapan data input tahun 2025**
- 2. Peramalan penjualan dari model Holt-Winters'**
- 3. Prediksi residual dari model Random Forest**
- 4. Perhitungan peramalan penjualan tahun 2025**

# Pengolahan Data

## 1. Persiapan data input tahun 2025

Data input berisi nilai dari tiap fitur pada tahun 2025 untuk model random forest. Nilai fitur pada tahun 2025 diramalkan menggunakan model SARIMA.

## 2. Peramalan penjualan dari model Holt-Winters'

Peramalan dengan model Holt-Winters' dilakukan dengan menerapkan metode **expanding window forecast**.

## 3. Prediksi residual dari model Random Forest

Peramalan dilakukan untuk 3 periode mendatang, kemudian nilai pada 3 periode tersebut dijumlahkan dengan nilai peramalan Holt-Winters' pada periode terkait untuk menghasilkan nilai peramalan sesungguhnya. Nilai sesungguhnya pada 3 periode tersebut dimasukkan ke dalam data pelatihan dan model random forest meramalkan untuk 3 periode selanjutnya. Proses dilakukan berulang kali.

## 4. Perhitungan peramalan penjualan pada tahun 2025

Penjualan sesungguhnya diperoleh melalui penjumlahan nilai dari model Holt-Winter's dengan nilai residual dari model random forest.

# Pengolahan Data

## Persiapan data input tahun 2025

Nilai fitur untuk tahun 2025 (untuk model Random Forest) dilakukan **menggunakan model SARIMA**. Misalnya untuk **fitur 'Perusahaan Z'**, peramalan dilakukan dengan menjalankan syntax berikut:

```
In [130]: n_forecast = 9
history = list(sm_full)
forecast = []

for i in range(n_forecast):
    model = SARIMAX(history, order=auto_model_sm.order,
                    seasonal_order=auto_model_sm.seasonal_order,
                    enforce_stationarity=True, enforce_invertibility=True)
    model_fit = model.fit(disp=False, method='lbfgs')

    Y_forecast = model_fit.forecast(steps=1)
    forecast.append(Y_forecast[0])
    history.append(Y_forecast[0])

start_date = pd.to_datetime('2025-04-01')
forecast_index = pd.date_range(start=start_date, periods=n_forecast, freq='MS')

forecast_series = pd.Series(forecast, index=forecast_index)
forecast_series
```

Untuk fitur lainnya, proses peramalan dilakukan dengan cara yang serupa.

# Pengolahan Data

## Persiapan Penjualan Tahun 2025

### Persiapan data input tahun 2025

Hasil peramalan dari setiap fitur yang dibutuhkan pada tahun 2025 ditampilkan pada tabel di bawah ini.

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag2	'Harga Produk X'_lag3
2025-01	37	4	0,06	...	20	19
2025-02	37	1	0,28	...	23	20
2025-03	34	5	0,10	...	25	23
2025-04	33	7	0,43	...	24	25
2025-05	35	1	0,57	...	24	24
2025-06	31	NaN	-0,94	...	38	24
2025-07	31	NaN	-1,75	...	19	38
2025-08	35	NaN	0,01	...	15	19
2025-09	33	NaN	0,51	...	19	15
2025-10	33	NaN	0,59	...	20	19
2025-11	34	NaN	-0,85	...	22	20
2025-12	35	NaN	-2,15	...	14	22

### Peramalan penjualan dari model Holt-Winters'

```
In [278]: n_forecast = 10
history = list(forecast_train)
predictions = []

for t in range(n_forecast):
    model = ExponentialSmoothing(history, trend='mul', seasonal='mul',
                                  seasonal_periods=12)
    model_fit = model.fit()

    forecast = model_fit.forecast(steps=1)
    predictions.append(forecast[0])
    history.append(forecast[0])

forecast_series = pd.Series(predictions, index=forecast_index)
forecast_series
```

### Prediksi residual dari model Random Forest

```
In [282]: MarMay2025_res = RF_final.predict(X_RF_MarMay2025)
MarMay2025_res = pd.Series(MarMay2025_res, index=forecast_index[0:3])
MarMay2025_res

In [287]: JunAug2025_res = RF_final.predict(X_RF_JunAug2025)
JunAug2025_res = pd.Series(JunAug2025_res, index=forecast_index[3:6])
JunAug2025_res

In [292]: SepNov2025_res = RF_final.predict(X_RF_SepNov2025)
SepNov2025_res = pd.Series(SepNov2025_res, index=forecast_index[6:9])
SepNov2025_res

In [299]: NovDec2025_res = RF_final.predict(X_RF_NovDec2025)
NovDec2025_res = pd.Series(NovDec2025_res, index=forecast_index[8:10])
NovDec2025_res
```

# Pengolahan Data

## Peramalan Penjualan Tahun 2025

### Persiapan data input tahun 2025

Hasil peramalan dari setiap fitur yang dibutuhkan pada tahun 2025 ditampilkan pada tabel di bawah ini.

Bulan	'Perusahaan Z'_lag1	Penjualan Produk X'_lag3	Inflasi_lag1	...	'Harga Produk X'_lag2	'Harga Produk X'_lag3
2025-01	37	4	0,06	...	20	19
2025-02	37	1	0,28	...	23	20
2025-03	34	5	0,10	...	25	23
2025-04	33	7	0,43	...	24	25
2025-05	35	1	0,57	...	24	24
2025-06	31	NaN	-0,94	...	38	24
2025-07	31	NaN	-1,75	...	19	38
2025-08	35	NaN	0,01	...	15	19
2025-09	33	NaN	0,51	...	19	15
2025-10	33	NaN	0,59	...	20	19
2025-11	34	NaN	-0,85	...	22	20
2025-12	35	NaN	-2,15	...	14	22

### Peramalan penjualan dari model Holt-Winters' (Hasil)

Bulan	Peramalan Holt-Winters'
2025-03	3,851
2025-04	2,379
2025-05	2,528
2025-06	2,698
2025-07	2,469
2025-08	2,563
2025-09	2,212
2025-10	1,983
2025-11	2,528
2025-12	3,653

### Prediksi residual dari model Random Forest (Hasil)

Bulan	Peramalan Residual
2025-03	1,481
2025-04	1,061
2025-05	-0,377
2025-06	-0,154
2025-07	-2,415
2025-08	-0,804
2025-09	-0,528
2025-10	-0,349
2025-11	0,782
2025-12	-1,083

# Pengolahan Data

## Perhitungan peramalan penjualan pada tahun 2025

Bulan	Peramalan Penjualan Holt-Winters'	Peramalan Residual Random Forest	Peramalan Penjualan Akhir	Peramalan Penjualan Akhir (round)
Maret 2025	3,851	1,481	5,332	5
April 2025	2,379	1,061	3,439	3
Mei 2025	2,528	-0,377	2,151	2
Juni 2025	2,698	-0,154	2,543	3
Juli 2025	2,469	-2,415	0,054	0
Agustus 2025	2,563	-0,804	1,759	2
September 2025	2,212	-0,528	1,684	2
Okttober 2025	1,983	-0,349	1,634	2
November 2025	2,528	0,782	3,310	3
Desember 2025	3,653	-1,083	2,571	3

Peramalan  
Penjualan  
Akhir

# THANK you

Eriel Matthew Gerald Morado

