

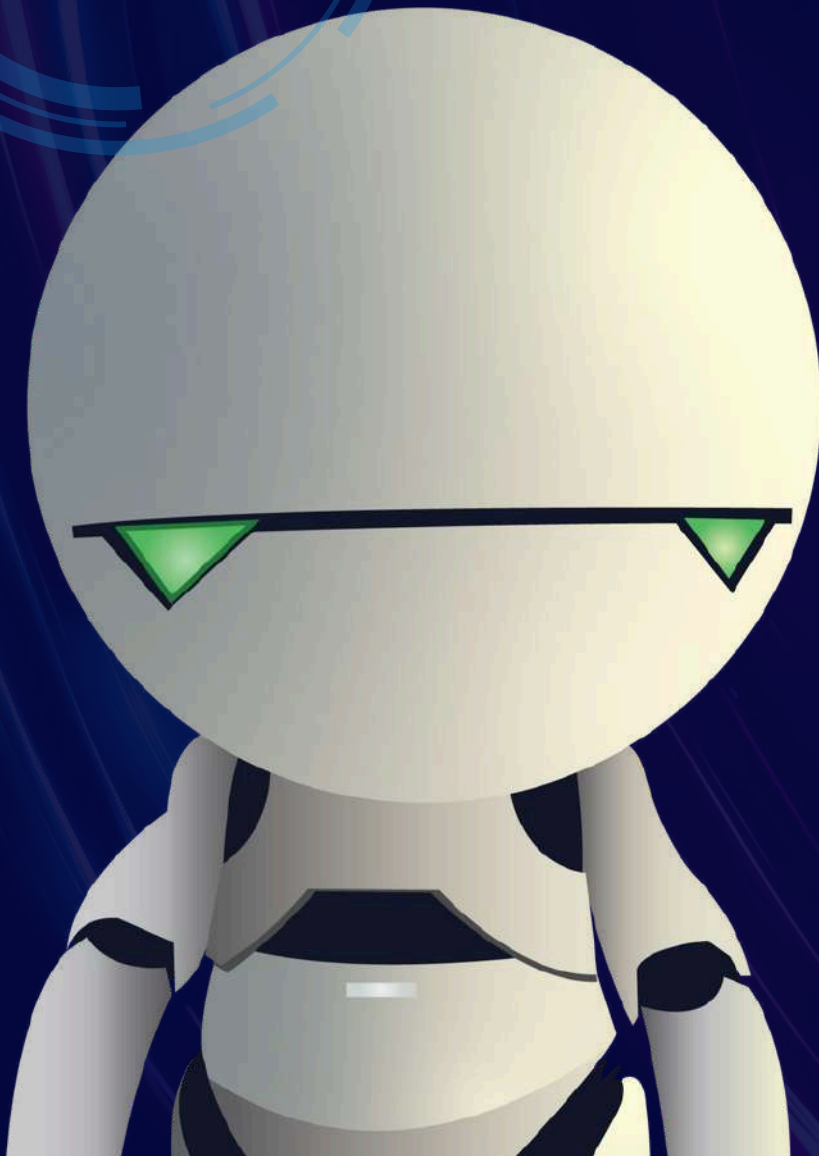
EL421-AL

BORDE DATA CLEANING & VISUALIZATION

A D V A N C E D P R O G R A M M I N G

DATA INFO

```
df = pd.read_csv('data/data_penjualan_borde.csv')  
  
print(df.info())  
print(df.describe(include='all'))
```



```
RangeIndex: 120 entries, 0 to 119  
Data columns (total 8 columns):  
#   Column                                Non-Null Count  Dtype  
---  -  
0   Tanggal                              120 non-null    object  
1   T_Kunjung                            120 non-null    int64  
2   T_Beli                               120 non-null    int64  
3   Penjualan                            120 non-null    object  
4   TK_Pesanan_Kunjung                   120 non-null    object  
5   Pesanan                              120 non-null    int64  
6   Penjualan_SiapKirim                  120 non-null    object  
7   TK_SiapKirim_Buat                    120 non-null    object  
dtypes: int64(3), object(5)
```




CHANGE FORMAT

```
#Ubah Format Tanggal Jadi (dd-mm-yyy)  
df['Tanggal'] = pd.to_datetime(df['Tanggal'], dayfirst=True)
```

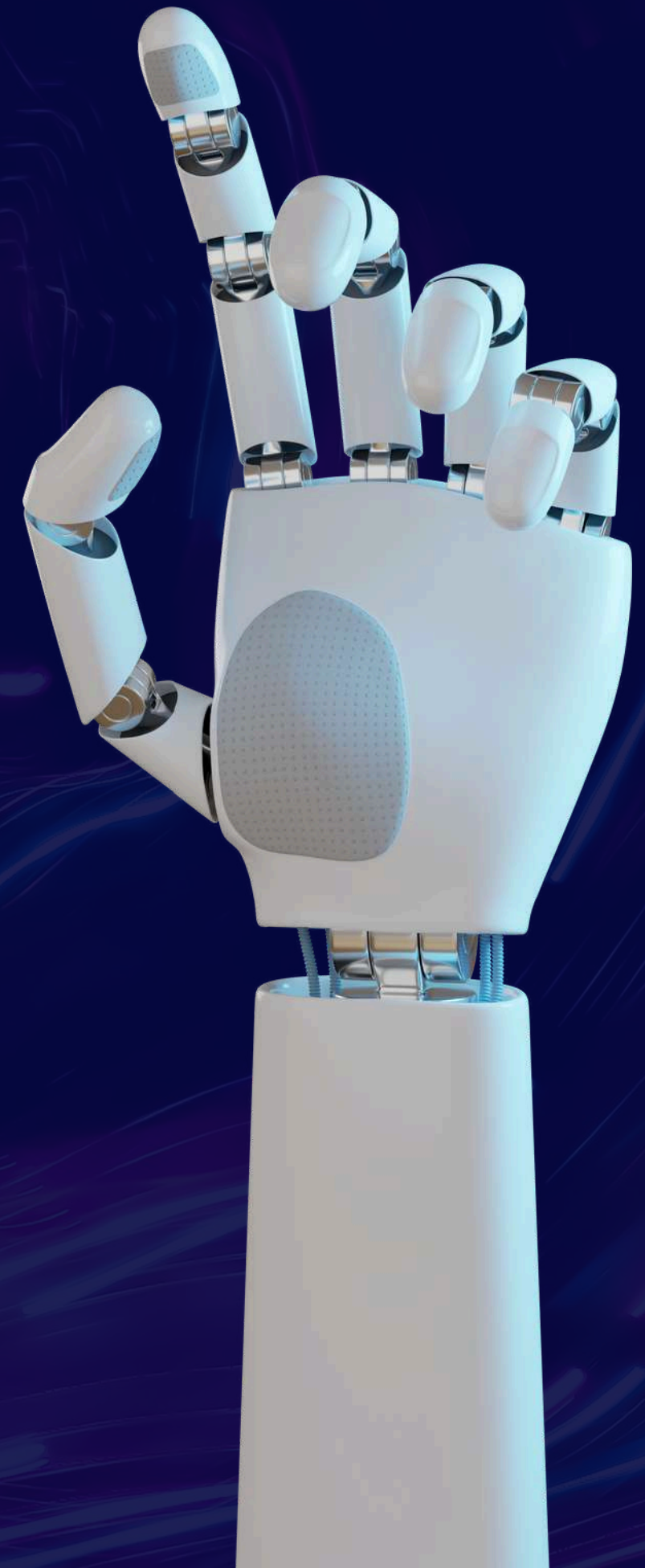
```
#Gabungkan Data per Bulan  
df['Bulan'] = df['Tanggal'].dt.to_period('M')  
df_bulan = df.groupby('Bulan')['T_Beli'].sum()
```

```
#Ubah Penjualan dari Desimal ke Ribuan  
df['Penjualan'] = df['Penjualan'].str.replace('.', '', regex=False)  
df['Penjualan_SiapKirim'] = df['Penjualan_SiapKirim'].str.replace('.', '', regex=False)
```


DELETE ROWS WITH NO SALES

```
#Hapus Baris yang Tidak Ada Penjualan
print("Data Penjualan Bersih")
df['T_Beli'] = pd.to_numeric(df['T_Beli'], errors='coerce')
df['Penjualan'] = pd.to_numeric(df['Penjualan'], errors='coerce')
df['Rata_Beli'] = (df['Penjualan']/df['T_Beli']).round(2)
df_clean = df[df['T_Beli'] != 0].dropna(subset=['T_Beli'])
print(df_clean)
df_clean.to_csv('data/data_penjualan_bersih_borde.csv', index=False)
```

data_penjualan_bersih_borde.csv





ORDER DELIVERED

```
#Tampilkan Data Pesanan yang Terkirim
print("\nData Pesanan Terkirim")
df_clean['Pesanan'] = pd.to_numeric(df_clean['Pesanan'], errors='coerce')
df_terkirim = df_clean[df_clean['Pesanan'] != 0].dropna(subset=['Pesanan'])
print(df_terkirim)
df_terkirim.to_csv('data/data_pesanan_terkirim.csv', index=False)
```

[data_pesanan_terkirim.csv](#)

ORDER NOT DELIVERED

```
#Tampilkan Data Pesanan yang Tidak Terkirim
print("\nData Pesanan Tidak Terkirim")
df_clean['Pesanan'] = pd.to_numeric(df_clean['Pesanan'], errors='coerce')
df_gagal_kirim = df_clean[df_clean['Pesanan'] == 0].dropna(subset=['Pesanan'])
print(df_gagal_kirim)
df_gagal_kirim.to_csv('data/data_pesanan_tidak_terkirim.csv', index=False)
```

data_pesanan tidak terkirim.csv

TOTAL SALES PER MONTH

```
#Hitung Total Penjualan per Bulan
total_penjualan_perbulan = df.groupby('Bulan')['Penjualan'].sum()
print("\nTotal Penjualan per Bulan")
print(total_penjualan_perbulan)
```

```
Total Penjualan per Bulan
Bulan
2025-01    4779273
2025-02    1955348
2025-03    2227735
2025-04     9817237
Freq: M, Name: Penjualan, dtype: int64
```



5 DAYS WITH THE MOST TRANSACTION

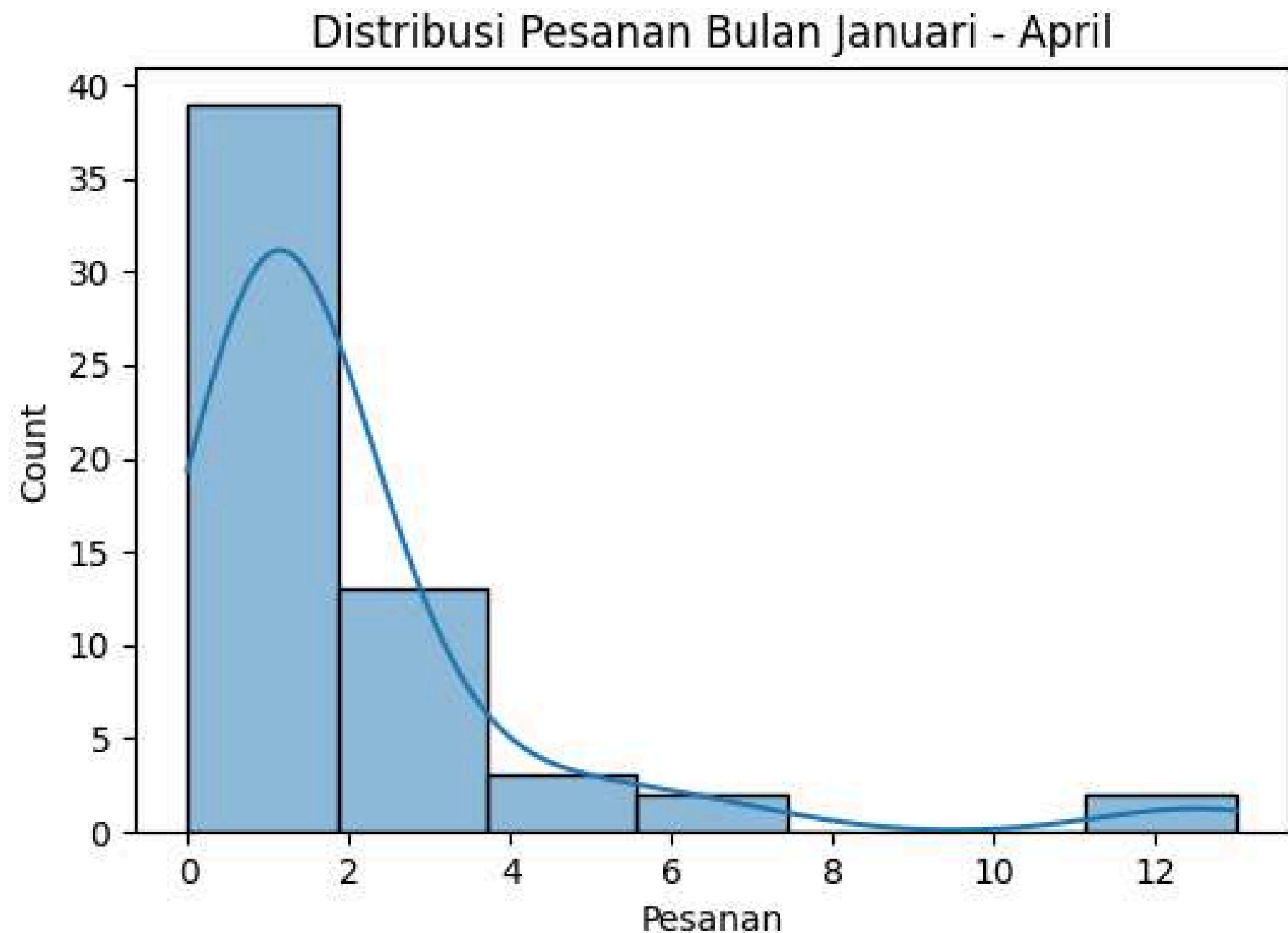
```
#Tampilkan 5 Hari dengan Transaksi Terbanyak
hari_transaksi_terbanyak = df.groupby('Tanggal')['Penjualan'].sum().sort_values(ascending=False)
print("\nHari Transaksi Terbanyak")
print(hari_transaksi_terbanyak.head(5))
```

Hari Transaksi Terbanyak	
Tanggal	
2025-04-28	2500470
2025-04-30	1773970
2025-04-25	1568524
2025-04-22	799329
2025-04-24	747125

Name: Penjualan, dtype: int64

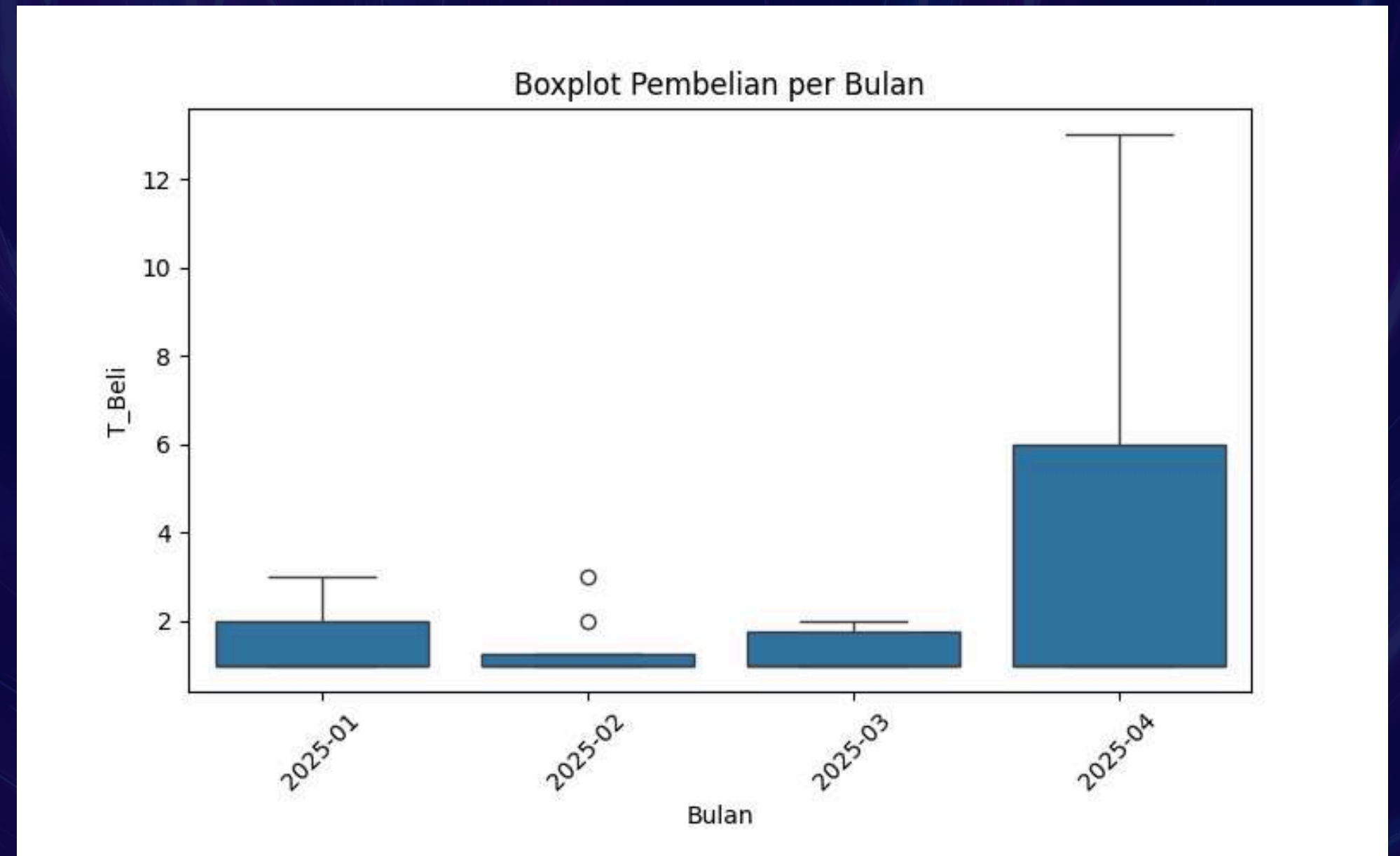
HISTOGRAM

```
#Histogram Pesanan  
plt.figure(figsize=(6,4))  
sns.histplot(df_clean['Pesanan'], bins=7, kde=True)  
plt.title('Distribusi Pesanan Bulan Januari - April')  
plt.show()
```

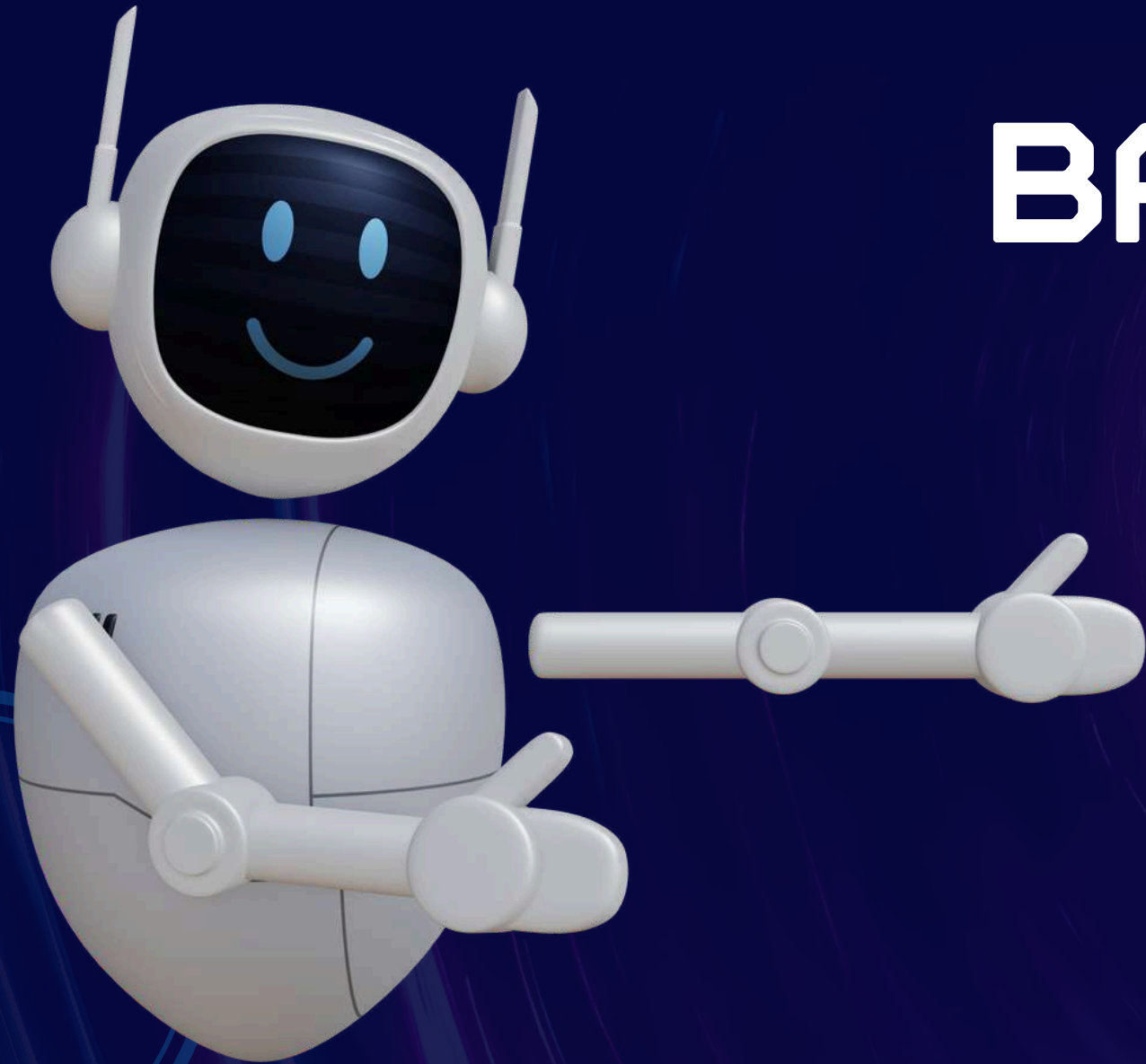



```
#Boxplot per Bulan
plt.figure(figsize=(8,5))
sns.boxplot(x='Bulan', y='T_Beli', data=df_clean)
plt.title('Boxplot Pembelian per Bulan')
plt.xticks(rotation=45)
plt.show()
```

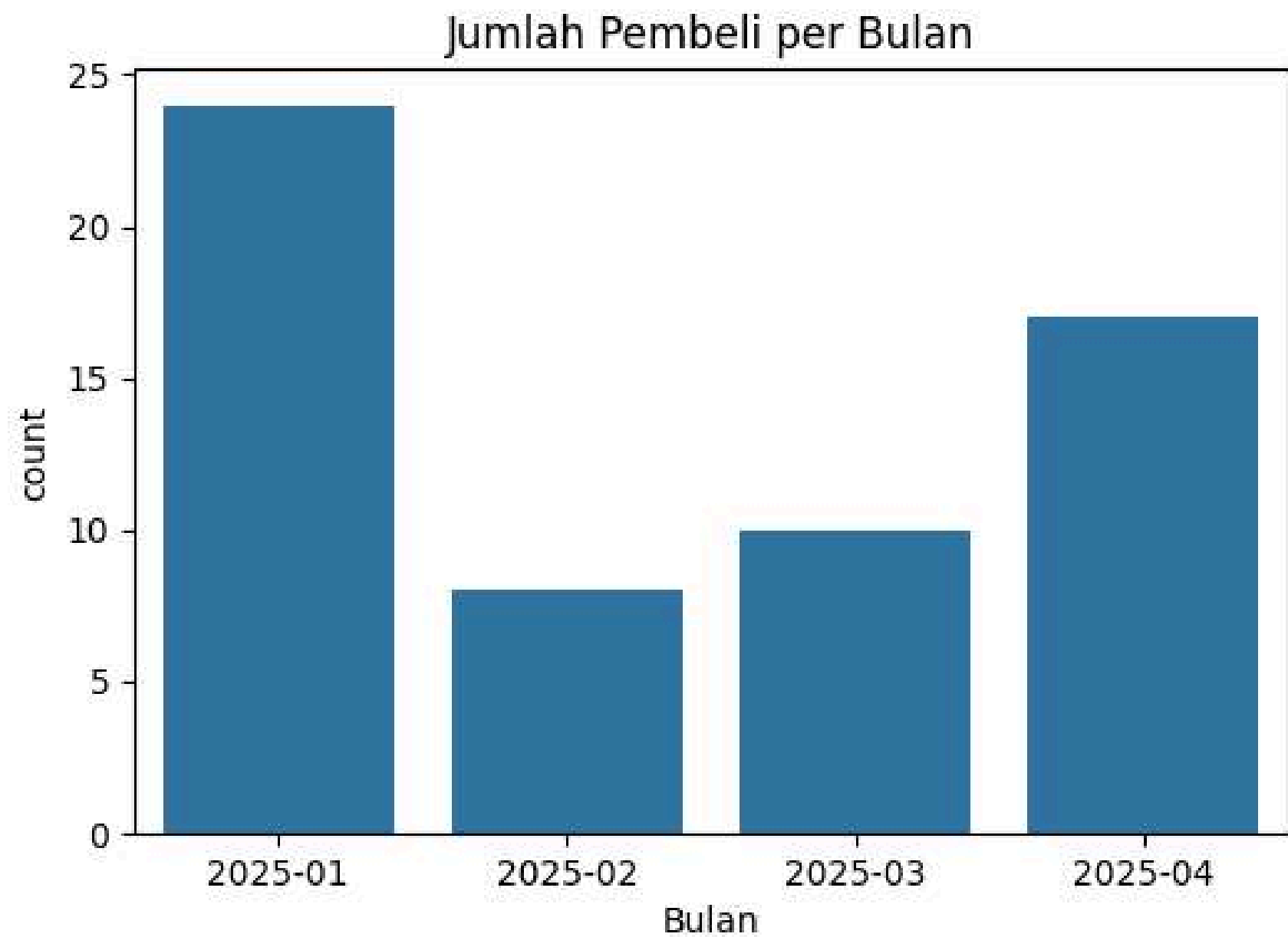
BOXPLOT

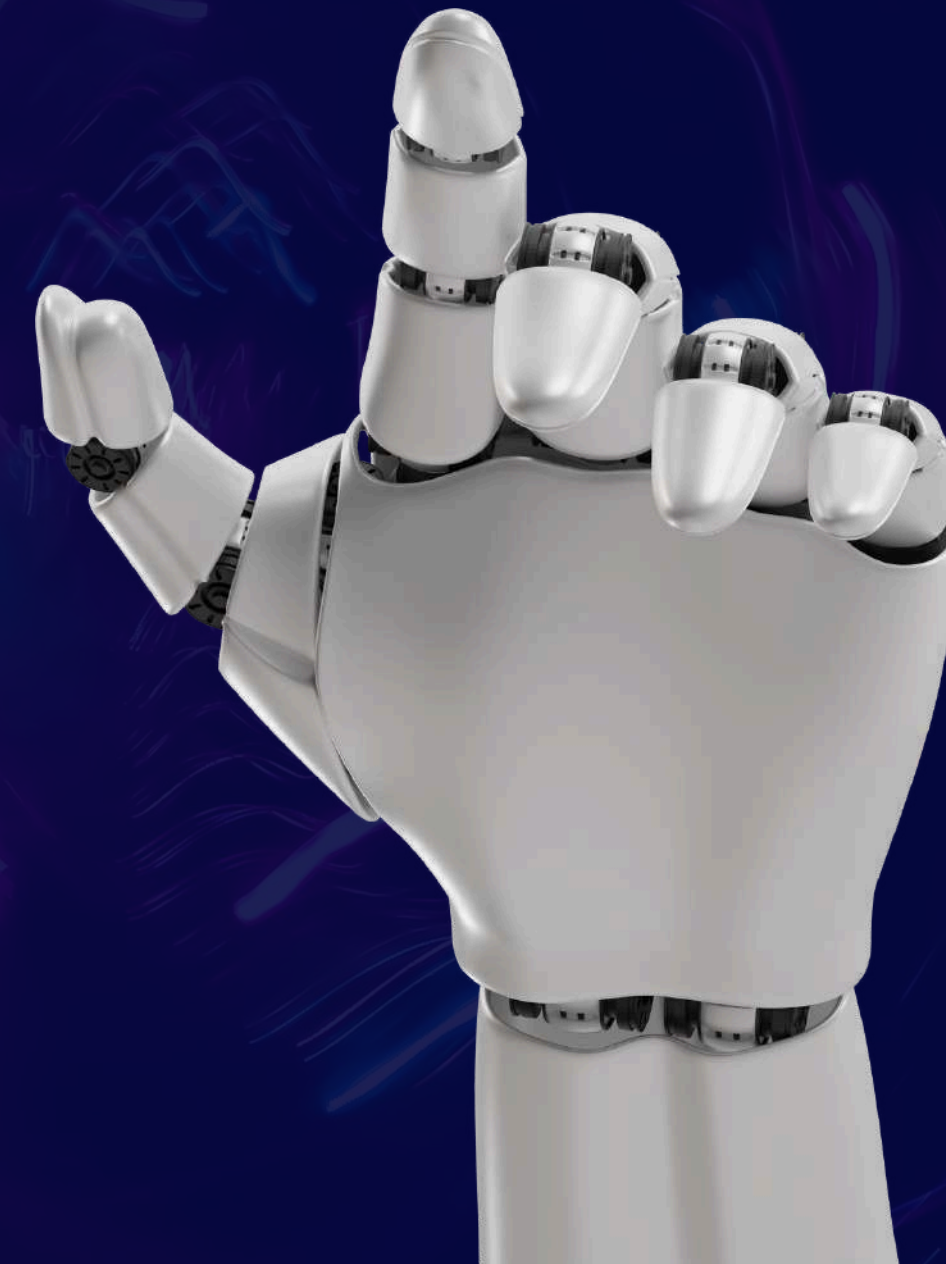
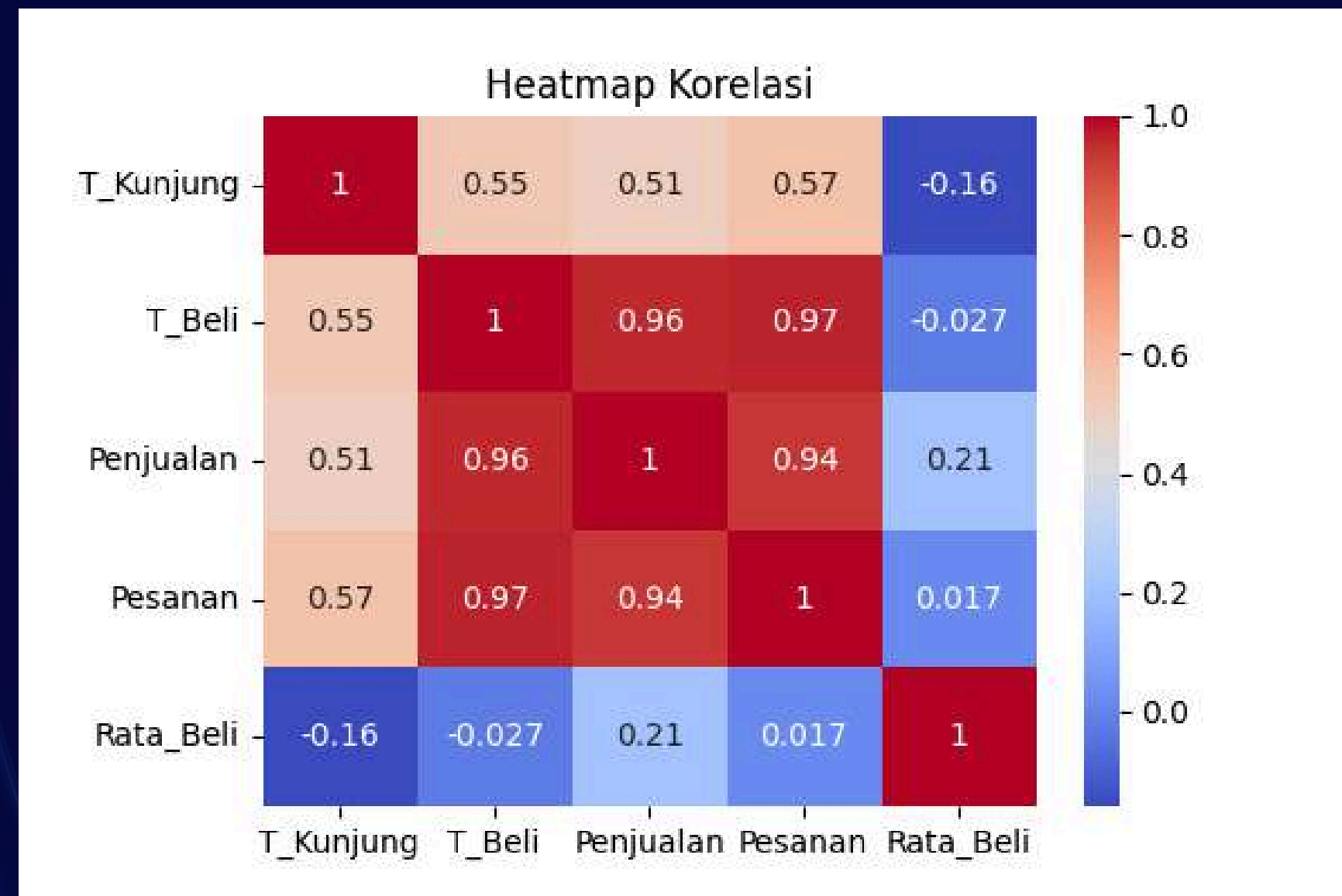


BARPLOT



```
#Barplot Jumlah Pesanan per Bulan  
plt.figure(figsize=(6,4))  
sns.countplot(x='Bulan', data=df_clean)  
plt.title('Jumlah Pembeli per Bulan')  
plt.show()
```





HEATMAP

```
#Heatmap Korelasi
plt.figure(figsize=(6,4))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Heatmap Korelasi')
plt.show()
```




THANK
YOU