

Penelitian TukuPos

Analisis Deret Waktu & Apriori pada Data TukuPos

Disusun Oleh:

Andi Hisyam H

Yusnia Auliya A

Table of Contents

1 Dataset

2 EDA

3 Preprocessing

4 Deret Waktu

5 Apriori





Dataset

Dataset

TukuPos

	ID	Tanggal	Trans ID	Konsumen	Nama Produk	Kategori	Qty
0	3088.0	2024-01-30 23:25:00	TR-240130-2	1012	11237.0	1217.0	2.0
1	3089.0	2024-01-30 23:25:00	TR-240130-2	1012	11304.0	1217.0	1.0
2	3090.0	2024-01-30 23:25:00	TR-240130-2	1012	11234.0	1217.0	1.0
3	3091.0	2024-01-30 23:31:00	TR-240130-3	1013	12182.0	1228.0	10.0
4	3092.0	2024-01-30 23:33:00	TR-240130-4	1014	9669.0	1207.0	1.0
...
19308	23240.0	2024-09-18 17:02:00	TR-240918-54	9995	12247.0	1208.0	5.0
19309	23241.0	2024-09-18 17:16:00	TR-240918-55	9996	11411.0	1217.0	1.0
19310	23242.0	2024-09-18 17:16:00	TR-240918-55	9996	11388.0	1217.0	5.0
19311	23243.0	2024-09-18 17:16:00	TR-240918-55	9996	11386.0	1217.0	6.0
19312	23244.0	2024-09-18 17:31:00	TR-240918-56	9997	11141.0	1217.0	1.0

19313 rows × 7 columns

Dataset TukuPos pada sheet
“Transaksi Penjualan”

- 19.313 baris
- 7 Kolom

Artinya total penjualan ada 19.313

Awal Transaksi:
30 Januari 2024 pukul 23:25

Akhir Transaksi:
18 September 2024 pukul 17:31



EDA **(Exploratory Data Analysis)**

EDA

Exploratory Data Analysis

Informasi Data

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19313 entries, 0 to 19312
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   ID               19303 non-null  float64
1   Tanggal          19313 non-null  datetime64[ns]
2   Trans ID         19313 non-null  object  
3   Konsumen         19313 non-null  int64   
4   Nama Produk      19303 non-null  float64
5   Kategori         19303 non-null  float64
6   Qty              19303 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 1.0+ MB
```

Perlu diubah pada tipe data
“**Nama Produk**” menjadi **Object**

EDA

Exploratory Data Analysis

Statistik Deskriptif

data.describe()						
	ID	Tanggal	Konsumen	Nama Produk	Kategori	Qty
count	19303.000000	19313	19313.000000	19303.000000	19303.000000	19303.000000
mean	13213.163550	2024-06-04 03:15:45.522704896	5574.265831	11224.731233	1219.058955	2.747449
min	3088.000000	2024-01-30 23:25:00	1012.000000	9472.000000	1207.000000	1.000000
25%	8144.500000	2024-04-06 13:35:00	3383.000000	10611.000000	1211.000000	1.000000
50%	13286.000000	2024-06-08 15:39:00	5646.000000	11375.000000	1217.000000	1.000000
75%	18258.500000	2024-07-30 18:47:00	7761.000000	11754.000000	1225.000000	2.000000
max	23244.000000	2024-09-18 17:31:00	9997.000000	12445.000000	1321.000000	1000.000000
std	5824.122308	NaN	2580.570371	712.028923	14.002980	13.908563

EDA

Exploratory Data Analysis

Missing Value

```
# Check for missing values
missing_data = data.isnull().sum()
print(missing_data)

# If necessary, drop missing values
data = data.dropna()
```

```
ID          10
Tanggal      0
Trans ID     0
Konsumen     0
Nama Produk  0
Kategori    10
Qty          10
dtype: int64
```

Missing Value sebanyak **10 data**,
maka akan dihapus

Duplicate Data

```
# Check for duplicate rows
duplicates = data.duplicated()
print(duplicates.sum())

# Remove duplicates if necessary
data = data.drop_duplicates()
```

```
0
```

Tidak Ada data yang duplikat

EDA

Exploratory Data Analysis

Total Penjualan Harian

Berikut jumlah total transaksi penjualan perharinya:

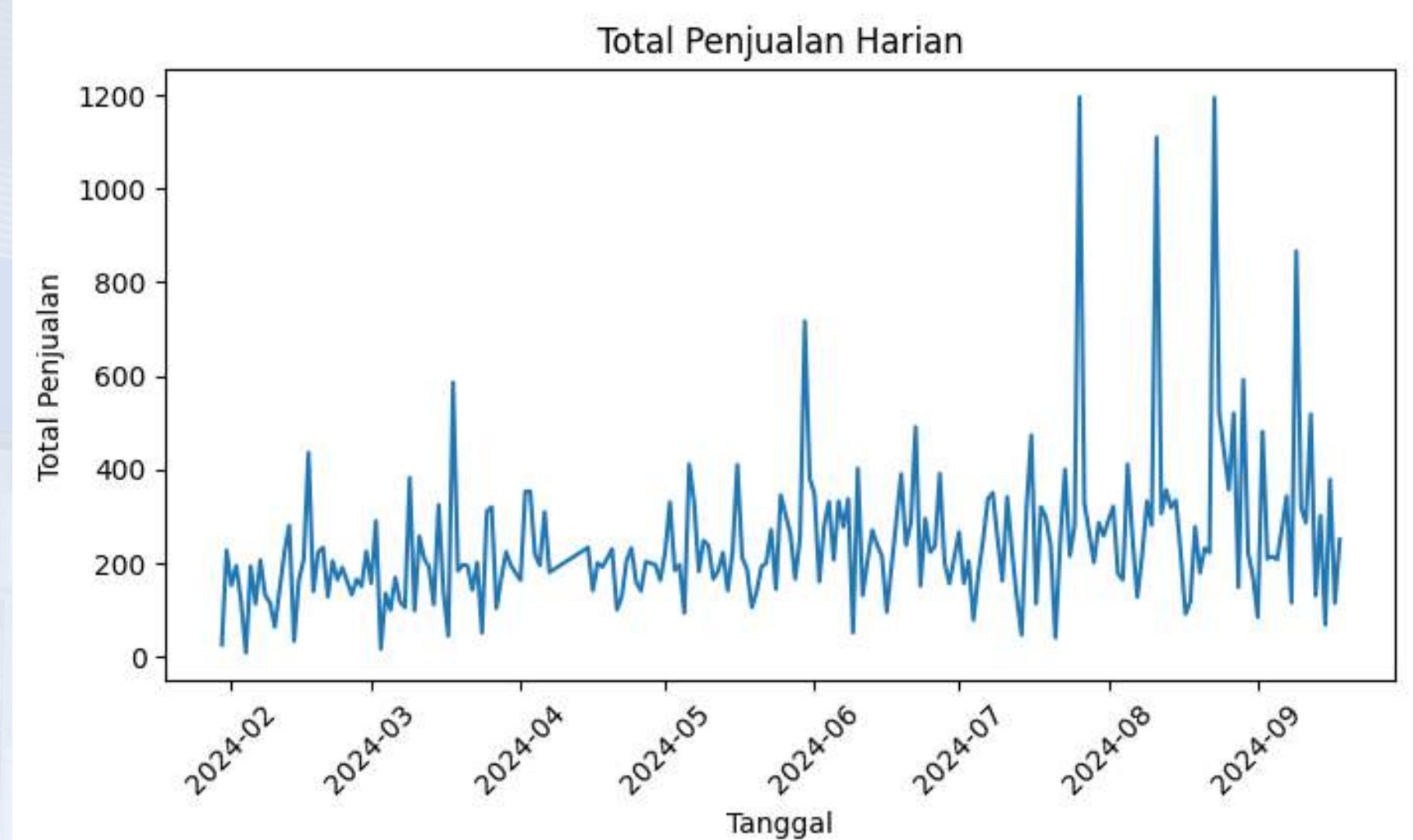
```
# Agregasi total penjualan harian
data_daily_sales = data.groupby(data['Tanggal'].dt.date)['Qty'].sum().reset_index()

# Rename columns
data_daily_sales.columns = ['Tanggal', 'Total Penjualan']

# Tampilkan hasil
print(data_daily_sales)
```

	Tanggal	Total Penjualan
0	2024-01-30	25.0
1	2024-01-31	227.0
2	2024-02-01	152.0
3	2024-02-02	194.0
4	2024-02-03	118.0
..
212	2024-09-14	301.0
213	2024-09-15	68.0
214	2024-09-16	378.0
215	2024-09-17	114.0
216	2024-09-18	250.0

[217 rows x 2 columns]



EDA

Exploratory Data Analysis

Total Penjualan Terbanyak

```
# Hitung total penjualan per bulan, hari, dan jam
total_sales = data.groupby(['month', 'day', 'hour'])['Qty'].sum().reset_index()

# Temukan total penjualan terbanyak
max_sales = total_sales.loc[total_sales['Qty'].idxmax()]

# Tampilkan hasil
print("Total Penjualan Terbanyak:")
print(f"Bulan: {max_sales['month']}, Hari: {max_sales['day']}, Jam: {max_sales['hour']}, Total Penjualan: {max_sales['Qty']}")
```

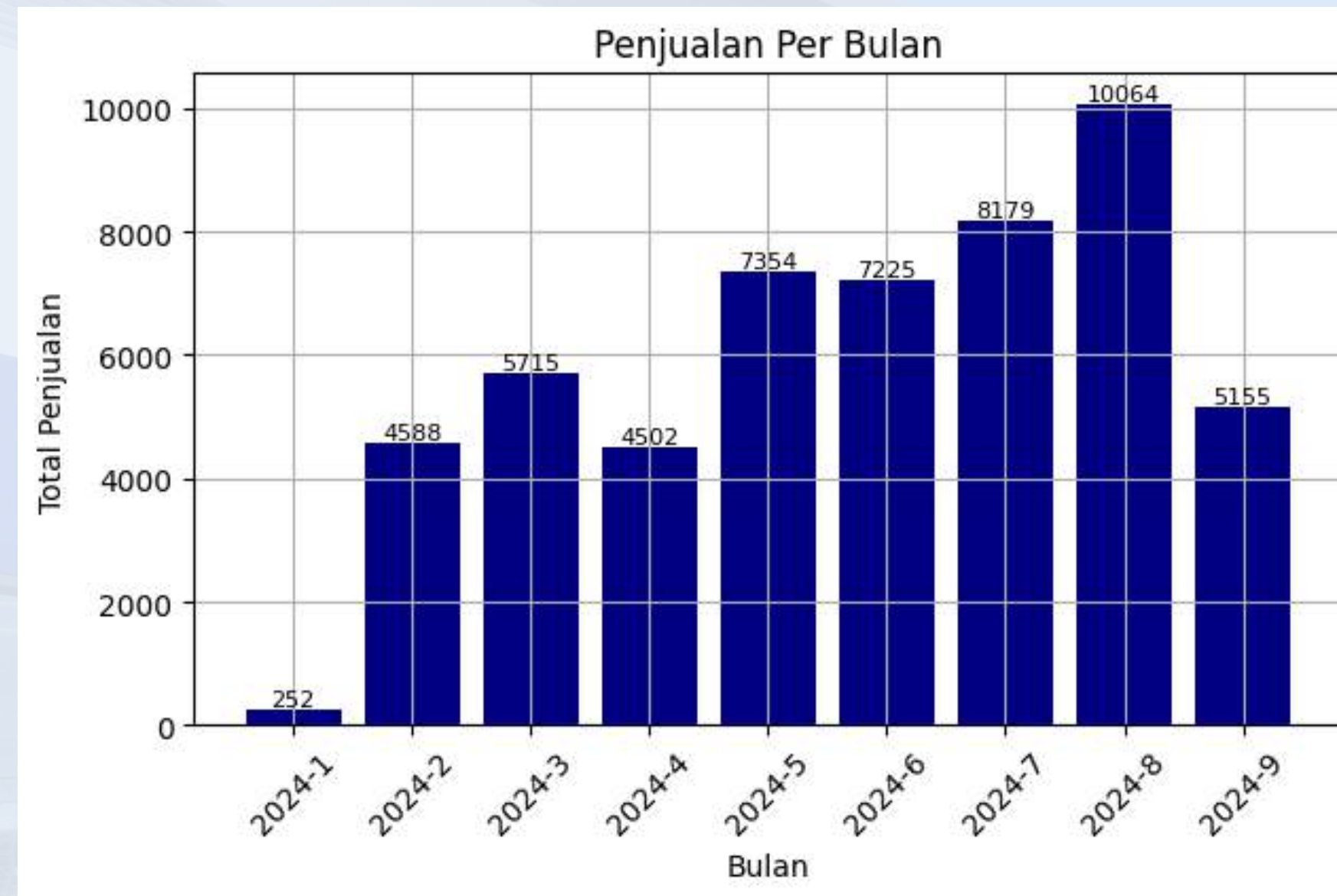
```
Total Penjualan Terbanyak:
Bulan: 8, Hari: Friday, Jam: 9, Total Penjualan: 1185.0
```

Penjualan Terbanyak terjadi pada bulan **Agustus**, hari **Jum'at**, pukul **9**, dan dengan total penjualan sebesar **1185** produk

EDA

Exploratory Data Analysis

Penjualan perBulan

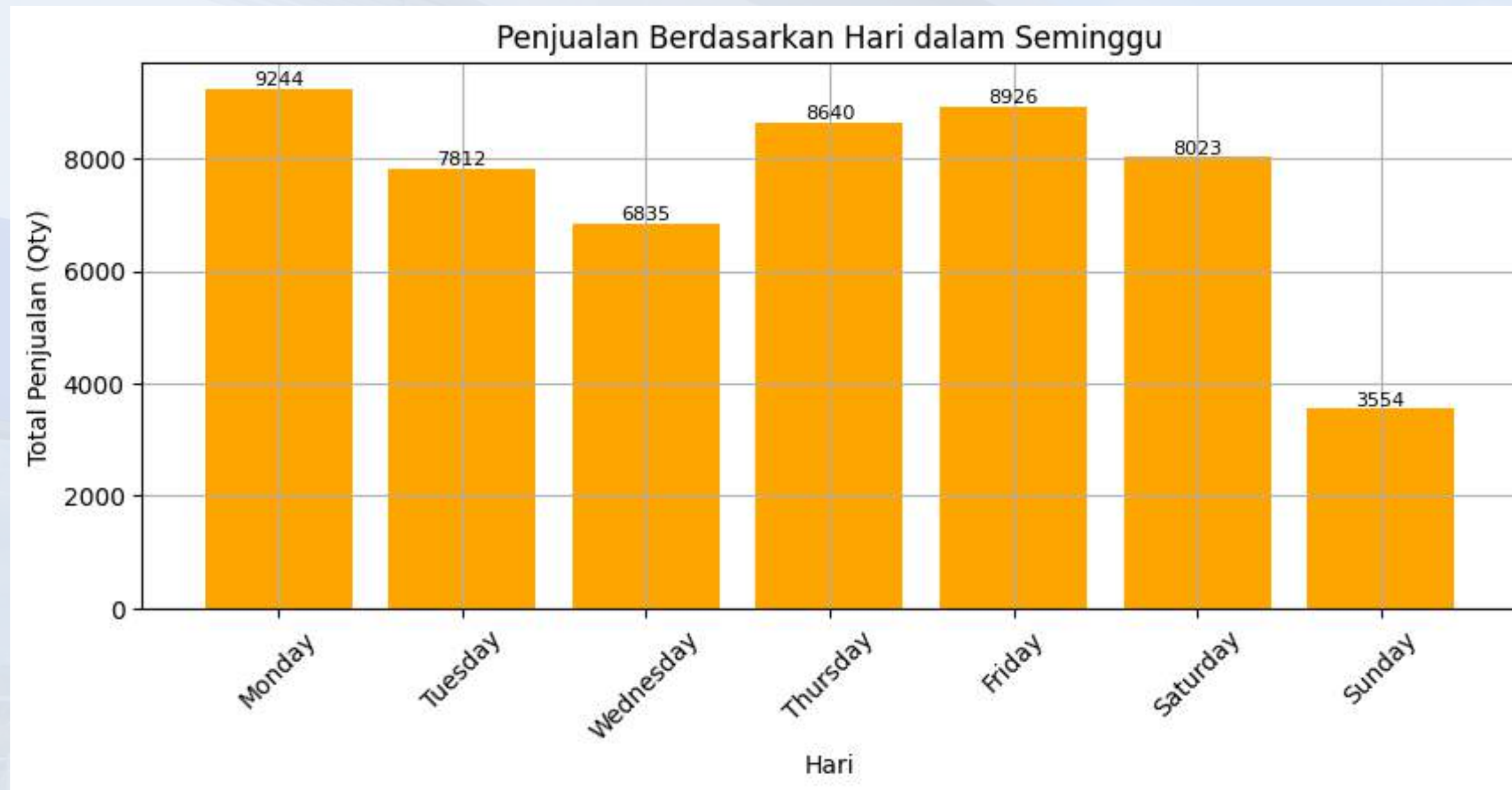


Penjualan perbulan terbanyak berturut-turut yaitu bulan **Agustus, Juli, Mei**, Juni, Maret, September, Februari, April, Januari

EDA

Exploratory Data Analysis

Penjualan berdasarkan Hari dalam Seminggu

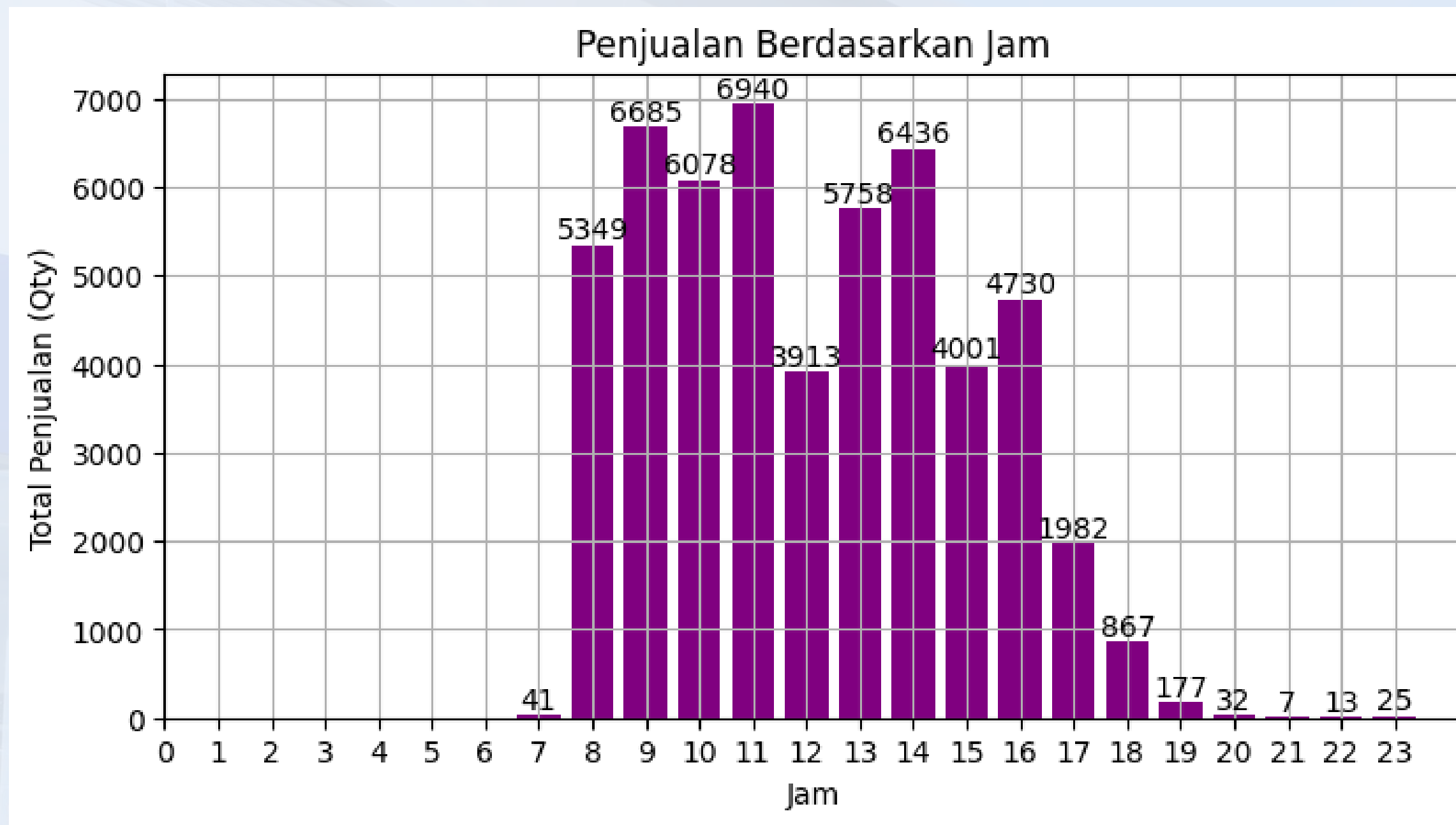


Penjualan berdasarkan hari dalam seminggu terbanyak berturut-turut yaitu hari **Senin, Jum'at, Kamis**, Sabtu, Selasa, Rabu, Minggu

EDA

Exploratory Data Analysis

Penjualan berdasarkan Jam

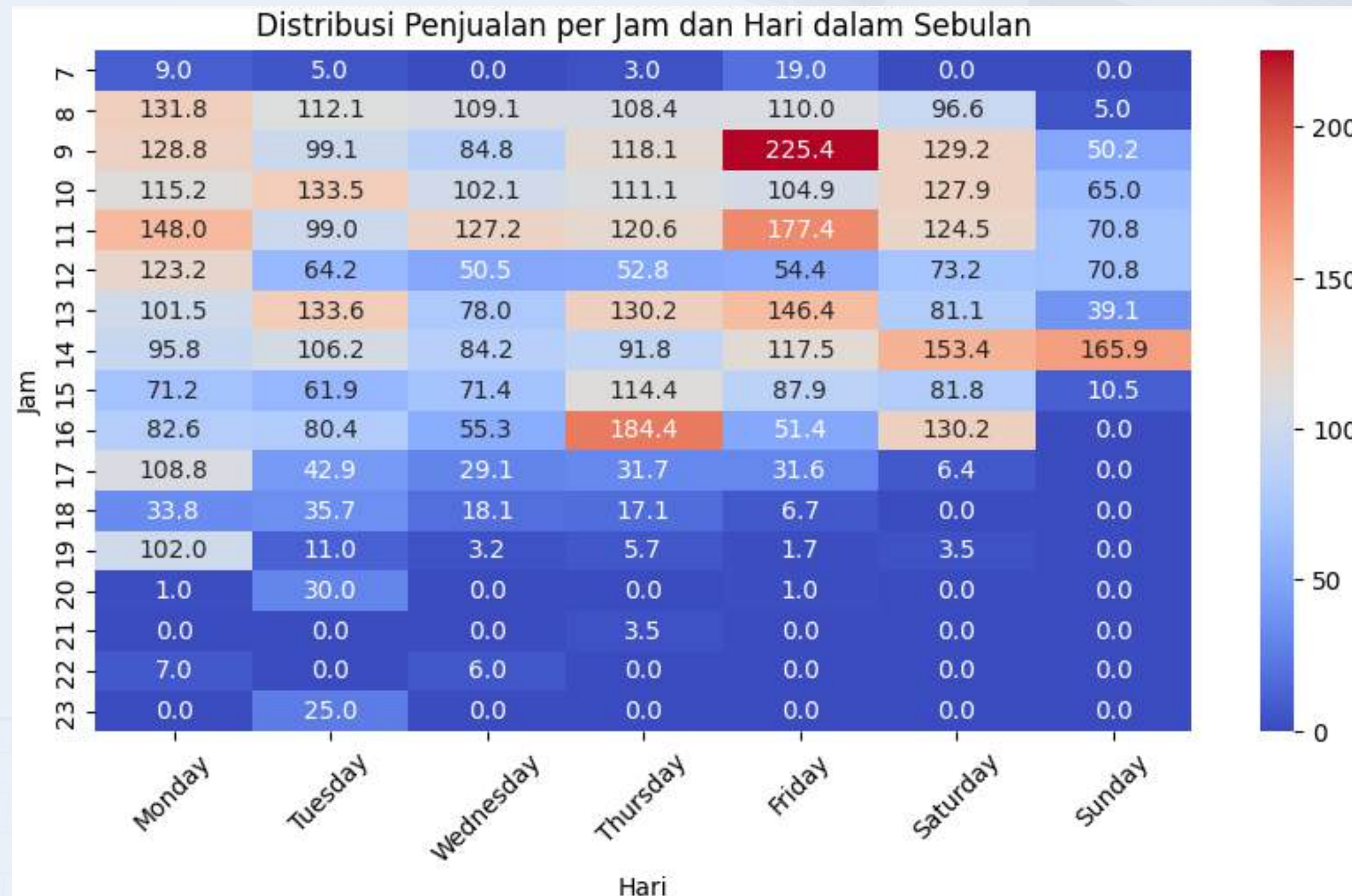


Penjualan berdasarkan jam terbanyak berturut-turut yaitu jam **11 siang**, jam **9 pagi**, jam **2 siang**, jam 10, jam 1 siang, dst.

EDA

Exploratory Data Analysis

Distribusi Penjualan perJam dan Hari dalam Sebulan

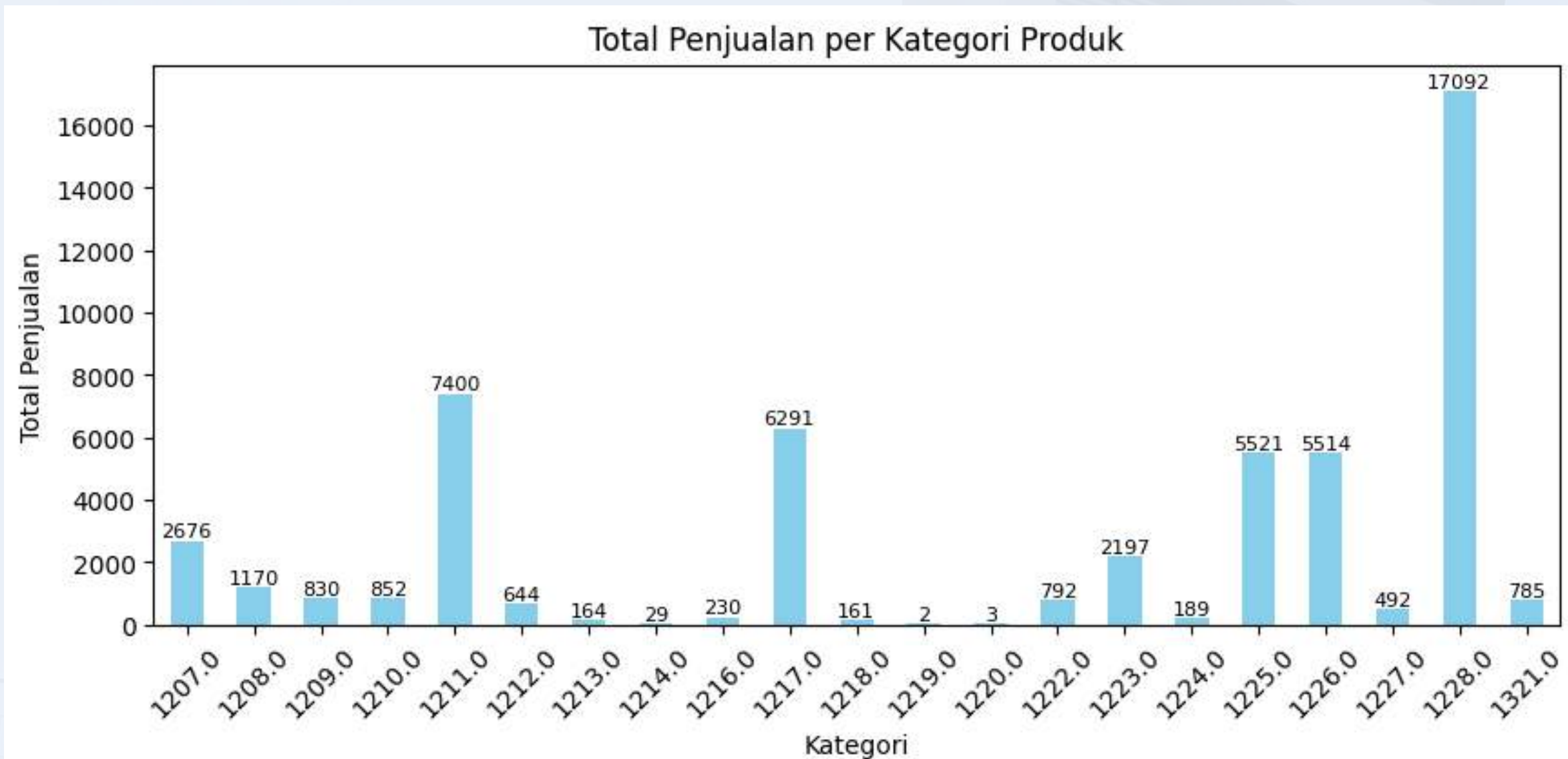


Semakin merah warnanya, maka hubungan antar kedua variabel semakin kuat

EDA

Exploratory Data Analysis

Total Penjualan perKategori Produk

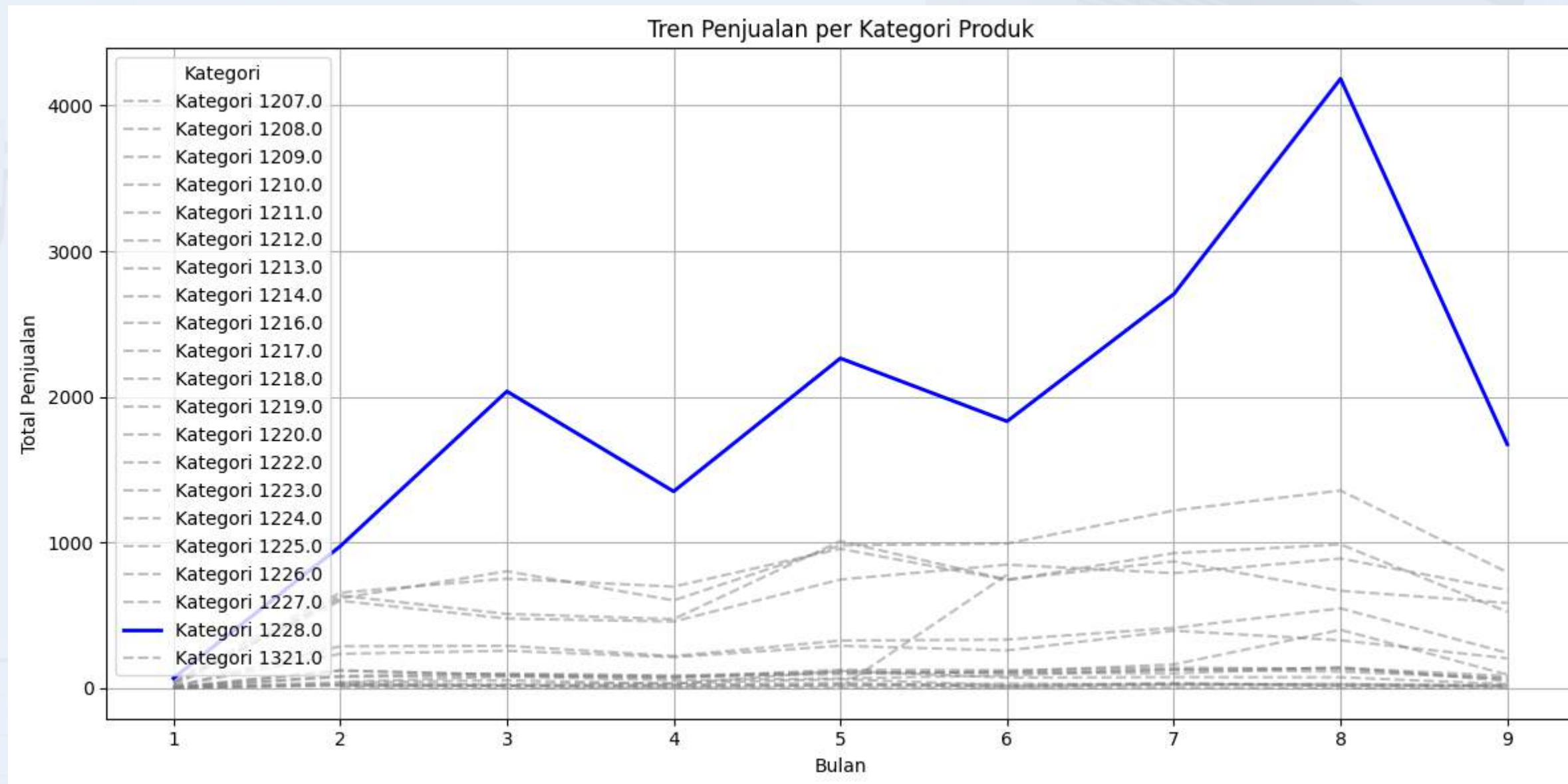


Penjualan terbanyak pada **kategori 1228** yaitu sebanyak 17.092 produk terjual

EDA

Exploratory Data Analysis

Tren Penjualan perKategori Produk

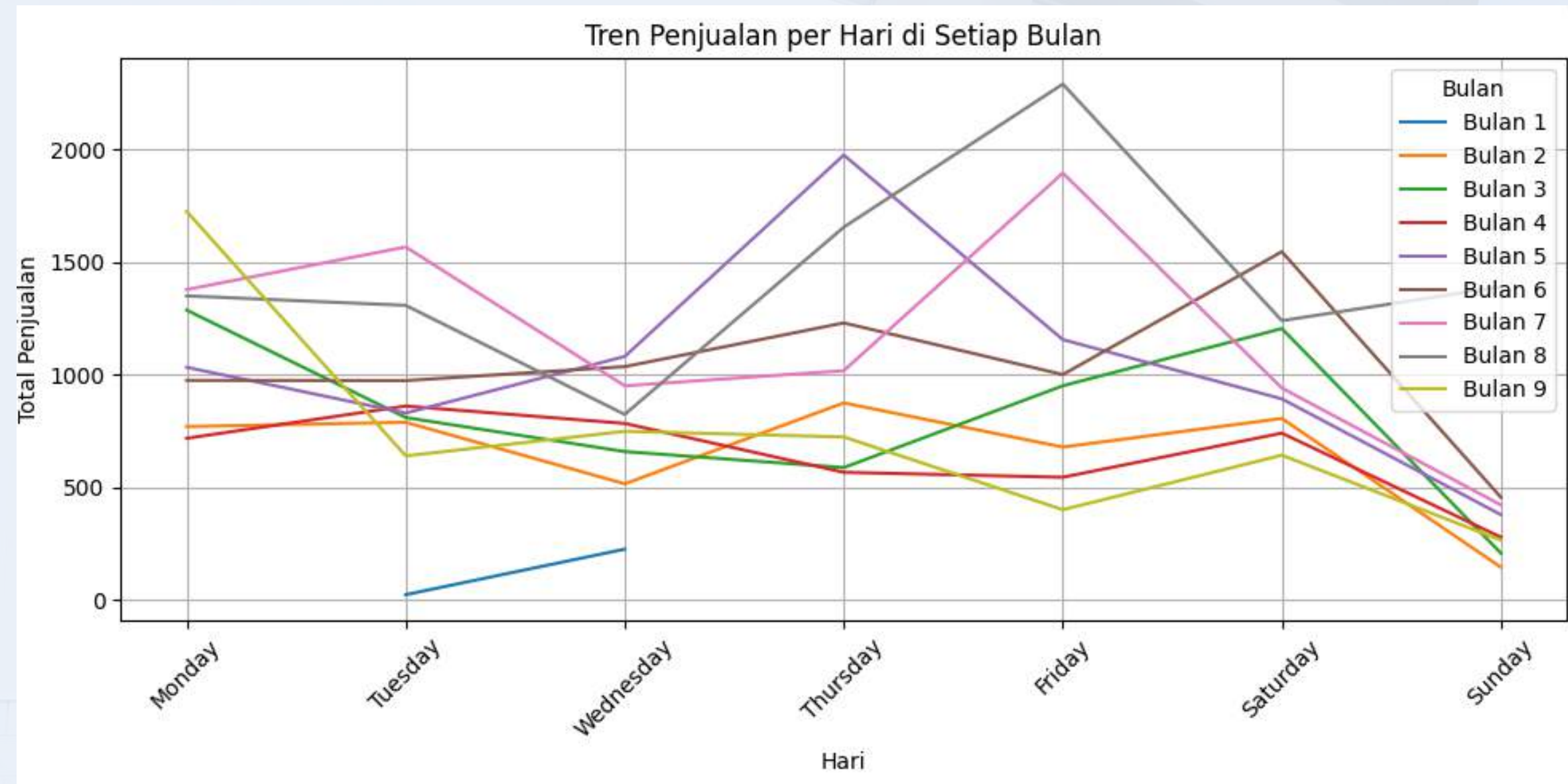


Kategori produk **1228.0** menunjukkan tren penjualan tertinggi dibandingkan kategori lainnya. Penjualan di kategori ini mencapai puncak yang sangat tinggi dibandingkan dengan kategori-kategori lain yang cenderung lebih stabil.

EDA

Exploratory Data Analysis

Tren Penjualan perHari di Setiap Bulan



Pada bulan 5 hingga 8, terdapat pola dimana penjualan selalu meningkat pada hari Kamis dan Jumat, namun mengalami penurunan di akhir pekan (Sabtu dan Minggu). Hal ini mengindikasikan adanya tren penjualan yang kuat di pertengahan minggu, tetapi cenderung menurun saat memasuki akhir pekan.

EDA

Exploratory Data Analysis

10 Konsumen Teratas berdasarkan Jumlah Transaksi

```
# Hitung jumlah transaksi per konsumen
transaction_counts = data['Konsumen'].value_counts().reset_index()
transaction_counts.columns = ['Konsumen', 'Jumlah_Transaksi']

# Ambil 10 konsumen teratas berdasarkan jumlah transaksi
top_consumers = transaction_counts.head(10)

# Tampilkan hasil
print("10 Konsumen Teratas Berdasarkan Jumlah Transaksi:")
print(top_consumers)
```

```
10 Konsumen Teratas Berdasarkan Jumlah Transaksi:
   Konsumen  Jumlah_Transaksi
0      9677                 32
1      7571                 30
2      4367                 28
3      6179                 25
4      8020                 23
5      5316                 21
6      9657                 20
7      2842                 19
8      1035                 19
9      5799                 18
```




Preprocessing Data

Preprocessing Data

Hapus Missing Value

```
# Check for missing values
missing_data = data.isnull().sum()
print(missing_data)

# If necessary, drop missing values
data = data.dropna()
```

```
ID          10
Tanggal      0
Trans ID     0
Konsumen     0
Nama Produk  0
Kategori    10
Qty          10
dtype: int64
```

10 Missing Value **sudah terhapus**

Ubah Tipe Data

```
# Convert 'Tanggal' to datetime
data['Tanggal'] = pd.to_datetime(data['Tanggal'])

# Convert 'Nama Produk' to string
data['Nama Produk'] = data['Nama Produk'].astype(str)
```

Ubah tipe data Tanggal ke datetime dan
Nama Produk ke string

Preprocessing Data

Hapus Duplicate Value

```
# Check for duplicate rows
duplicates = data.duplicated()
print(duplicates.sum())

# Remove duplicates if necessary
data = data.drop_duplicates()
```

0

Tidak Terdapat Data Duplikat

Preprocessing Data

Memecah kolom “Tanggal”

```
# Memecah kolom tanggal menjadi komponen
data['year'] = data['Tanggal'].dt.year
data['month'] = data['Tanggal'].dt.month
data['day'] = data['Tanggal'].dt.day_name()
data['hour'] = data['Tanggal'].dt.hour
data['minute'] = data['Tanggal'].dt.minute
data['second'] = data['Tanggal'].dt.second
```

Kodingan

	ID	Tanggal	Trans ID	Konsumen	Nama Produk	Kategori	Qty
0	3088.0	2024-01-30 23:25:00	TR-240130-2	1012	11237.0	1217.0	2.0
1	3089.0	2024-01-30 23:25:00	TR-240130-2	1012	11304.0	1217.0	1.0
2	3090.0	2024-01-30 23:25:00	TR-240130-2	1012	11234.0	1217.0	1.0
3	3091.0	2024-01-30 23:31:00	TR-240130-3	1013	12182.0	1228.0	10.0
4	3092.0	2024-01-30 23:33:00	TR-240130-4	1014	9669.0	1207.0	1.0

Sebelum



	ID	Tanggal	Trans ID	Konsumen	Nama Produk	Kategori	Qty	year	month	day	hour	minute	second
0	3088.0	2024-01-30 23:25:00	TR-240130-2	1012	11237.0	1217.0	2.0	2024	1	Tuesday	23	25	0
1	3089.0	2024-01-30 23:25:00	TR-240130-2	1012	11304.0	1217.0	1.0	2024	1	Tuesday	23	25	0
2	3090.0	2024-01-30 23:25:00	TR-240130-2	1012	11234.0	1217.0	1.0	2024	1	Tuesday	23	25	0
3	3091.0	2024-01-30 23:31:00	TR-240130-3	1013	12182.0	1228.0	10.0	2024	1	Tuesday	23	31	0
4	3092.0	2024-01-30 23:33:00	TR-240130-4	1014	9669.0	1207.0	1.0	2024	1	Tuesday	23	33	0

Sesudah

Preprocessing Data

Agregasi Total Penjualan Harian

```
# Agregasi total penjualan harian
data_daily_sales = data.groupby(data['Tanggal'].dt.date)['Qty'].sum().reset_index()

# Rename columns
data_daily_sales.columns = ['Tanggal', 'Total Penjualan']

# Tampilkan hasil
print(data_daily_sales)
```

	Tanggal	Total Penjualan
0	2024-01-30	25.0
1	2024-01-31	227.0
2	2024-02-01	152.0
3	2024-02-02	194.0
4	2024-02-03	118.0
..
212	2024-09-14	301.0
213	2024-09-15	68.0
214	2024-09-16	378.0
215	2024-09-17	114.0
216	2024-09-18	250.0



Analisis Deret Waktu

Untuk melihat bagaimana penjualan kedepannya

Analisis Deret Waktu

Uji Stasioneritas

```
from statsmodels.tsa.stattools import adfuller

# ADF Test for stationarity
result = adfuller(data_daily_sales['Total Penjualan'])
print('ADF Statistic:', result[0])
print('p-value:', result[1])
```

```
ADF Statistic: -6.678089076180782
p-value: 4.412711842531713e-09
```

Nilai ADF sebesar -6,678
dan nilai p-value
0.0000000004412 artinya
data ini **sudah stasioner**

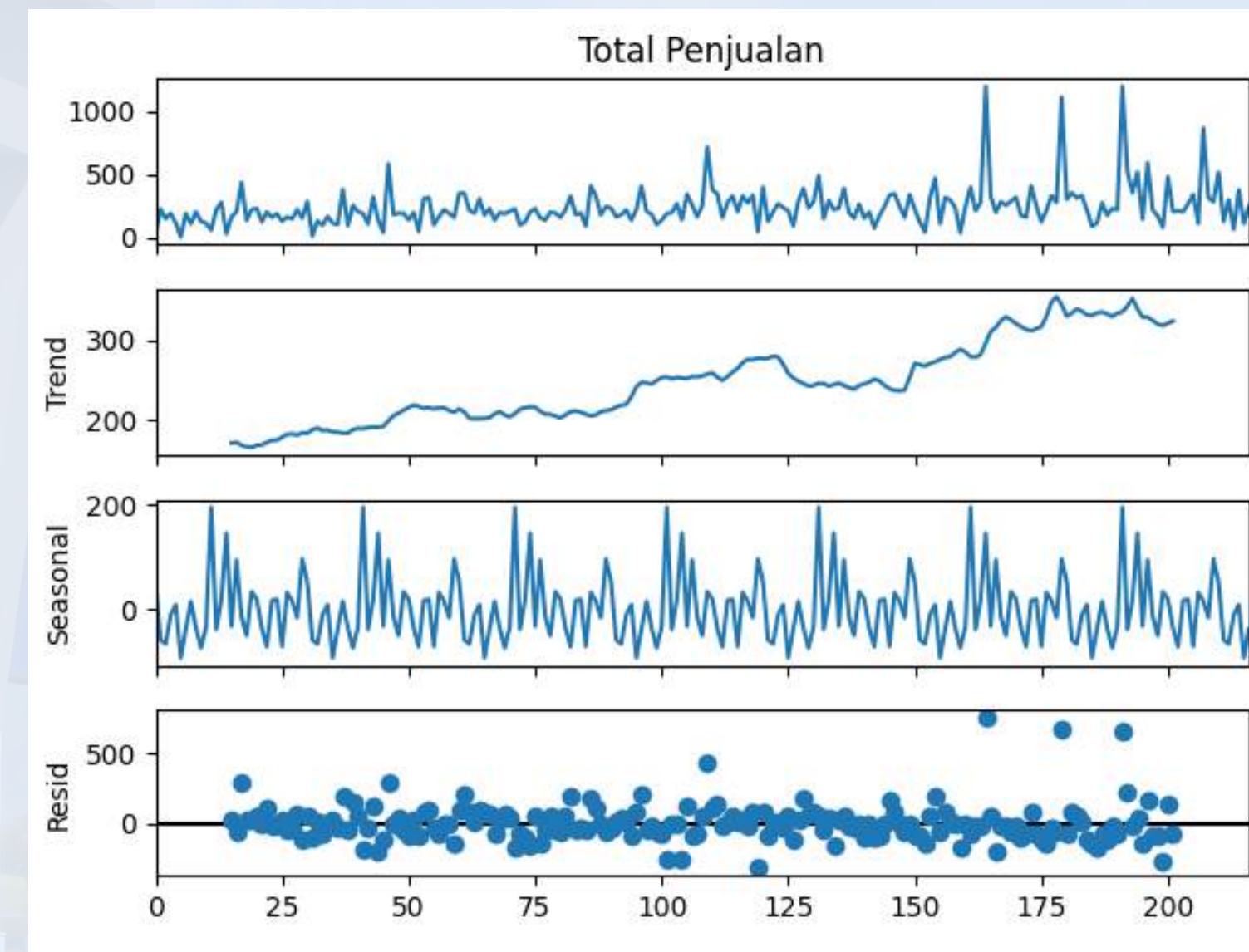
Analisis Deret Waktu

Dekomposisi

```
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

# Dekomposisi
decomposition = seasonal_decompose(data_daily_sales['Total Penjualan'], model='additive', period=30)

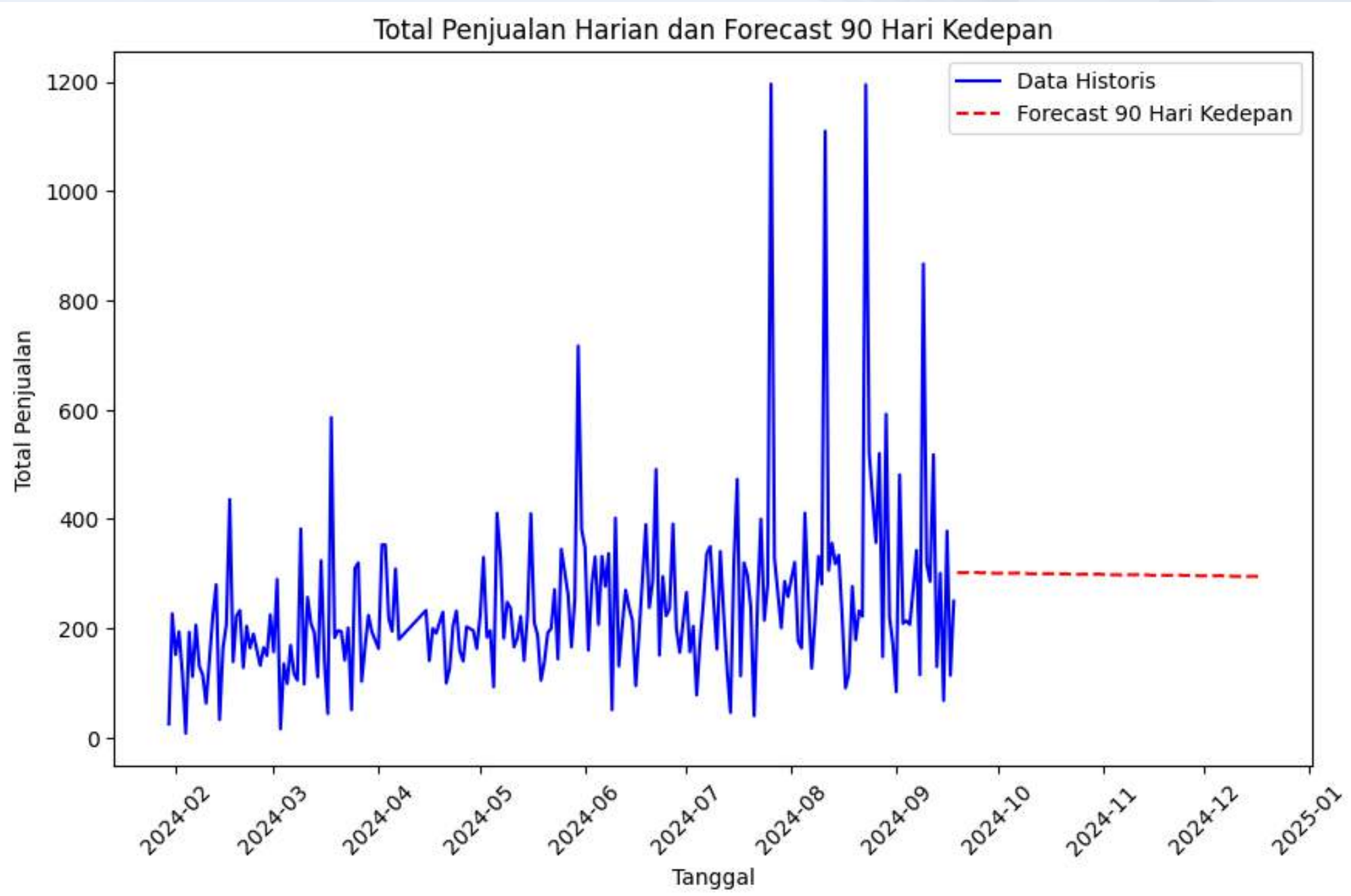
# Plot hasil dekomposisi
decomposition.plot()
plt.show()
```



Dari grafik disamping, menunjukkan bahwa data ini termasuk **Seasonal**

Analisis Deret Waktu

Prediksi Penjualan Menggunakan Model SARIMA



Berdasarkan hasil prediksi yang diperoleh, terlihat bahwa untuk penjualan 90 hari ke depan bersifat stasioner. Hal ini berbanding lurus dengan pengecekan data yang sudah dilakukan sebelumnya, di mana data yang diberikan memang bersifat stasioner atau penjualannya tetap dan hanya bersifat musiman saja. Dengan demikian, model memprediksi bahwa penjualan ke depannya akan tetap stasioner atau mendatar, karena hanya belajar dari data bulan-bulan sebelumnya.

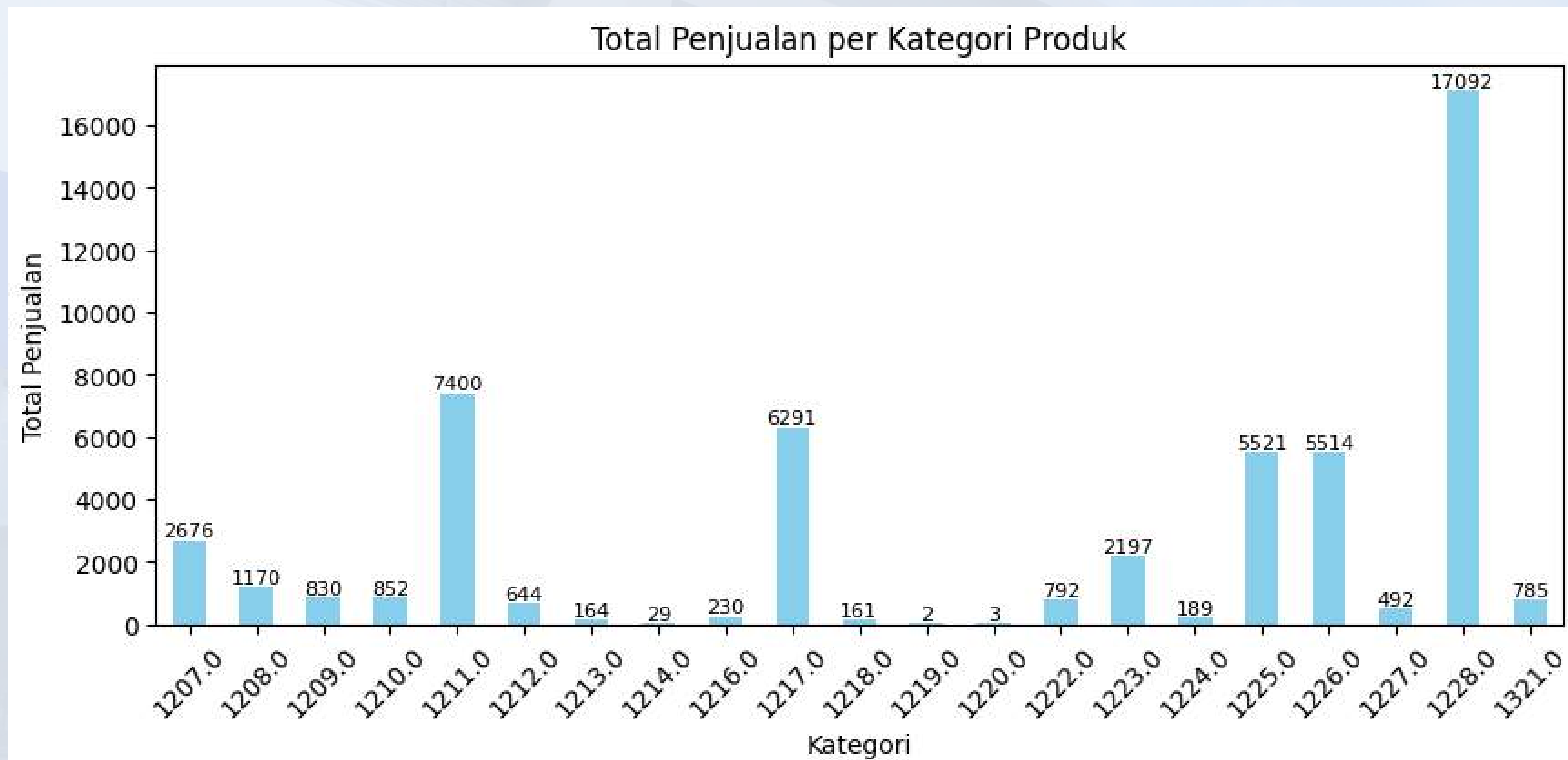


Analisis Algoritma Apriori

Untuk menentukan penataan barang di rak penjualan

Analisis Algoritma Apriori

Kategori Produk



Terdapat **21 Kategori**

Analisis Algoritma Apriori

Kategori 1207

1. Pembelian produk di Kategori 1207

```
kategori_1207 = (data[data['Kategori']== 1207].groupby(['Trans ID', 'Nama Produk'])['Qty'].sum().unstack().reset_index().fillna(0).set_index('Trans ID'))
kategori_1207.head(3)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in and should_run_async(code)

Nama Produk	10000.0	10001.0	10005.0	10007.0	10009.0	10013.0	10014.0	10015.0	10017.0	10025.0	...	9973.0	9979.0	9981.0	9982.0	9983.0	9991.0	9996.0
Trans ID																		
TR-240130-4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TR-240131-13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TR-240131-14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

3 rows × 423 columns

Analisis Algoritma Apriori

Kategori 1207

2. Encoding data pembelian produk

```
def hot_encode(x):  
    if(x<= 0): #jika yang dibeli lebih dari atau sama dengan 0 maka ditulis 0 (tidak dibeli)  
        return 0  
    if(x>= 1): #jika yang dibeli lebih dari atau sama dengan 1 maka ditulis 1 (dibeli)  
        return 1  
encoded_1207 = kategori_1207.applymap(hot_encode)  
kategori_1207 = encoded_1207  
encoded_1207.head(3)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically and should_run_async(code)
<ipython-input-199-06e3dc69a278>:6: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
encoded_1207 = kategori_1207.applymap(hot_encode)

Nama Produk	10000.0	10001.0	10005.0	10007.0	10009.0	10013.0	10014.0	10015.0	10017.0	10025.0	...	9973.0	9979.0	9981.0	9982.0	9983.0
Trans ID																
TR-240130-4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
TR-240131-13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
TR-240131-14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

3 rows x 423 columns

Analisis Algoritma Apriori

Kategori 1207

minimal nilai support 4%

3. Hasil Apriori

Hasil Analisis Apriori pada Kategori 1207

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(10242.0)	(10050.0)	0.034868	0.039749	0.010460	0.300000	7.547368	0.009074	1.371787	0.898844
1	(10050.0)	(10242.0)	0.039749	0.034868	0.010460	0.263158	7.547368	0.009074	1.309823	0.903413
3	(10051.0)	(10249.0)	0.028591	0.044630	0.004881	0.170732	3.825457	0.003605	1.152063	0.760332
5	(9474.0)	(9478.0)	0.035565	0.050209	0.004881	0.137255	2.733660	0.003096	1.100894	0.657577
6	(9475.0)	(9657.0)	0.038354	0.065551	0.004881	0.127273	1.941586	0.002367	1.070723	0.504299
11	(9479.0)	(9657.0)	0.041144	0.065551	0.004881	0.118644	1.809953	0.002184	1.060240	0.466701
2	(10249.0)	(10051.0)	0.044630	0.028591	0.004881	0.109375	3.825457	0.003605	1.090704	0.773097
4	(9478.0)	(9474.0)	0.050209	0.035565	0.004881	0.097222	2.733660	0.003096	1.068297	0.667716
9	(9478.0)	(9656.0)	0.050209	0.055091	0.004184	0.083333	1.512658	0.001418	1.030810	0.356828
8	(9656.0)	(9478.0)	0.055091	0.050209	0.004184	0.075949	1.512658	0.001418	1.027856	0.358672
7	(9657.0)	(9475.0)	0.065551	0.038354	0.004881	0.074468	1.941586	0.002367	1.039020	0.518977
10	(9657.0)	(9479.0)	0.065551	0.041144	0.004881	0.074468	1.809953	0.002184	1.036006	0.478891

Nilai **confidence** dan **lift** yang tinggi menunjukkan perilaku konsumen yang kuat dalam membeli produk secara bersamaan atau saling terkait. Artinya, ketika suatu produk (**antecedent**) **dibeli**, ada **kecenderungan tinggi** bahwa produk lain (**consequent**) juga akan **dibeli**

minimial nilai lift adalah 1

Analisis Algoritma Apriori

Kategori 1208

minimal nilai support 1%

Hasil Apriori

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
14	(10507.0, 10515.0)	(12247.0)	0.001499	0.032984	0.001499	1.000000	30.318182	0.001450	inf	0.968468
3	(10517.0)	(10507.0)	0.001499	0.256372	0.001499	1.000000	3.900585	0.001115	inf	0.744745
13	(12247.0, 10515.0)	(10507.0)	0.001499	0.256372	0.001499	1.000000	3.900585	0.001115	inf	0.744745
12	(12247.0, 10507.0)	(10515.0)	0.004498	0.097451	0.001499	0.333333	3.420513	0.001061	1.353823	0.710843
8	(10518.0)	(10510.0)	0.023988	0.097451	0.004498	0.187500	1.924038	0.002160	1.110829	0.492063
10	(12367.0)	(10510.0)	0.008996	0.097451	0.001499	0.166667	1.710256	0.000623	1.083058	0.419062
4	(10512.0)	(10509.0)	0.010495	0.064468	0.001499	0.142857	2.215947	0.000823	1.091454	0.554545
1	(10522.0)	(10505.0)	0.020990	0.040480	0.001499	0.071429	1.764550	0.000650	1.033329	0.442573
6	(10522.0)	(10509.0)	0.020990	0.064468	0.001499	0.071429	1.107973	0.000146	1.007496	0.099541
9	(10510.0)	(10518.0)	0.097451	0.023988	0.004498	0.046154	1.924038	0.002160	1.023238	0.532115
15	(12247.0)	(10507.0, 10515.0)	0.032984	0.001499	0.001499	0.045455	30.318182	0.001450	1.046048	1.000000
0	(10505.0)	(10522.0)	0.040480	0.020990	0.001499	0.037037	1.764550	0.000650	1.016665	0.451563
5	(10509.0)	(10512.0)	0.064468	0.010495	0.001499	0.023256	2.215947	0.000823	1.013065	0.586538
7	(10509.0)	(10522.0)	0.064468	0.020990	0.001499	0.023256	1.107973	0.000146	1.002320	0.104167
17	(10515.0)	(12247.0, 10507.0)	0.097451	0.004498	0.001499	0.015385	3.420513	0.001061	1.011057	0.784053

Perilaku Konsumen:

- Pembelian pada produk 10507 dan 10515 cenderung akan membeli produk 12247

Sehingga, penataan barang di rak pada produk 10507 dan 10515 didekatkan dengan produk 12247

minimial nilai lift adalah 1

Analisis Algoritma Apriori

Kategori 1209

minimal nilai support 1%

Hasil Apriori

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
10	(10538.0, 10530.0)	(10531.0)	0.001724	0.091379	0.001724	1.000000	10.943396	0.001567	inf	0.910190
12	(10530.0, 10531.0)	(10538.0)	0.001724	0.174138	0.001724	1.000000	5.742574	0.001424	inf	0.827288
11	(10538.0, 10531.0)	(10530.0)	0.003448	0.034483	0.001724	0.500000	14.500000	0.001605	1.931034	0.934256
16	(10537.0, 10546.0)	(10538.0)	0.003448	0.174138	0.001724	0.500000	2.871287	0.001124	1.651724	0.653979
18	(10546.0, 10538.0)	(10537.0)	0.003448	0.227586	0.001724	0.500000	2.196970	0.000939	1.544828	0.546713
7	(10545.0)	(10537.0)	0.018966	0.227586	0.005172	0.272727	1.198347	0.000856	1.062069	0.168717
17	(10537.0, 10538.0)	(10546.0)	0.006897	0.043103	0.001724	0.250000	5.800000	0.001427	1.275862	0.833333
3	(10543.0)	(10529.0)	0.008621	0.015517	0.001724	0.200000	12.888889	0.001590	1.230603	0.930435
2	(10529.0)	(10543.0)	0.015517	0.008621	0.001724	0.111111	12.888889	0.001590	1.115302	0.936953
8	(10545.0)	(10541.0)	0.018966	0.081034	0.001724	0.090909	1.121857	0.000187	1.010862	0.110721
4	(10550.0)	(10530.0)	0.025862	0.034483	0.001724	0.066667	1.933333	0.000832	1.034483	0.495575
0	(10550.0)	(10525.0)	0.025862	0.058621	0.001724	0.066667	1.137255	0.000208	1.008621	0.123894
14	(10530.0)	(10538.0, 10531.0)	0.034483	0.003448	0.001724	0.050000	14.500000	0.001605	1.049002	0.964286
5	(10530.0)	(10550.0)	0.034483	0.025862	0.001724	0.050000	1.933333	0.000832	1.025408	0.500000
20	(10546.0)	(10537.0, 10538.0)	0.043103	0.006897	0.001724	0.040000	5.800000	0.001427	1.034483	0.864865

Perilaku Konsumen:

- Pembelian pada produk 10538 dan 10530 cenderung akan membeli produk 10531

Sehingga, penataan barang di rak pada produk 10538 dan 10530 didekatkan dengan produk 10531

minimial nilai lift adalah 1

Analisis Algoritma Apriori

Kategori 1210

minimal nilai support 2%

Hasil Apriori

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
12	(10598.0)	(10597.0)	0.006568	0.006568	0.006568	1.000000	152.250000	0.006525	inf	1.000000
13	(10597.0)	(10598.0)	0.006568	0.006568	0.006568	1.000000	152.250000	0.006525	inf	1.000000
2	(12243.0)	(10581.0)	0.014778	0.027915	0.003284	0.222222	7.960784	0.002872	1.249824	0.887500
8	(12243.0)	(10588.0)	0.014778	0.054187	0.003284	0.222222	4.101010	0.002483	1.216045	0.767500
14	(12243.0)	(10600.0)	0.014778	0.070608	0.003284	0.222222	3.147287	0.002241	1.194933	0.692500
10	(12317.0)	(10588.0)	0.029557	0.054187	0.004926	0.166667	3.075758	0.003325	1.134975	0.695431
3	(10581.0)	(12243.0)	0.027915	0.014778	0.003284	0.117647	7.960784	0.002872	1.116585	0.899493
4	(12317.0)	(10583.0)	0.029557	0.083744	0.003284	0.111111	1.326797	0.000809	1.030788	0.253807
6	(10586.0)	(10588.0)	0.032841	0.054187	0.003284	0.100000	1.845455	0.001505	1.050903	0.473684
11	(10588.0)	(12317.0)	0.054187	0.029557	0.004926	0.090909	3.075758	0.003325	1.067488	0.713542
1	(10588.0)	(10565.0)	0.054187	0.072250	0.004926	0.090909	1.258264	0.001011	1.020525	0.217014
0	(10565.0)	(10588.0)	0.072250	0.054187	0.004926	0.068182	1.258264	0.001011	1.015019	0.221239
9	(10588.0)	(12243.0)	0.054187	0.014778	0.003284	0.060606	4.101010	0.002483	1.048784	0.799479
7	(10588.0)	(10586.0)	0.054187	0.032841	0.003284	0.060606	1.845455	0.001505	1.029557	0.484375
15	(10600.0)	(12243.0)	0.070608	0.014778	0.003284	0.046512	3.147287	0.002241	1.033281	0.734099

Perilaku Konsumen:

- Pembelian pada produk 10598 cenderung akan membeli produk 10597

Sehingga, penataan barang di rak pada produk 10598 didekatkan dengan produk 10597

minimial nilai lift adalah 1

Analisis Algoritma Apriori

Kategori 1211

minimal nilai support 2%

Hasil Apriori

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
13	(10624.0, 10609.0)	(10611.0)	0.005791	0.420490	0.002673	0.461538	1.097621	0.000238	1.076233	0.089456
9	(12395.0)	(10615.0)	0.009354	0.170156	0.003118	0.333333	1.958988	0.001526	1.244766	0.494155
5	(10616.0)	(10615.0)	0.022272	0.170156	0.006236	0.280000	1.645550	0.002446	1.152561	0.401237
3	(12395.0)	(10613.0)	0.009354	0.170156	0.002227	0.238095	1.399277	0.000636	1.089170	0.288040
6	(10622.0)	(10615.0)	0.011581	0.170156	0.002227	0.192308	1.130185	0.000257	1.027426	0.116539
12	(10624.0, 10611.0)	(10609.0)	0.015145	0.065479	0.002673	0.176471	2.695078	0.001681	1.134776	0.638625
14	(10609.0, 10611.0)	(10624.0)	0.015590	0.062361	0.002673	0.171429	2.748980	0.001700	1.131634	0.646305
11	(10626.0)	(10624.0)	0.016927	0.062361	0.002227	0.131579	2.109962	0.001172	1.079706	0.535116
18	(10615.0, 10611.0)	(10624.0)	0.020935	0.062361	0.002227	0.106383	1.705927	0.000922	1.049263	0.422657
0	(10624.0)	(10609.0)	0.062361	0.065479	0.005791	0.092857	1.418124	0.001707	1.030181	0.314453
1	(10609.0)	(10624.0)	0.065479	0.062361	0.005791	0.088435	1.418124	0.001707	1.028604	0.315502
15	(10624.0)	(10609.0, 10611.0)	0.062361	0.015590	0.002673	0.042857	2.748980	0.001700	1.028488	0.678543
17	(10609.0)	(10624.0, 10611.0)	0.065479	0.015145	0.002673	0.040816	2.695078	0.001681	1.026764	0.673022
4	(10615.0)	(10616.0)	0.170156	0.022272	0.006236	0.036649	1.645550	0.002446	1.014924	0.472740
10	(10624.0)	(10626.0)	0.062361	0.016927	0.002227	0.035714	2.109962	0.001172	1.019484	0.561045
16	(10624.0)	(10615.0, 10611.0)	0.062361	0.020935	0.002227	0.035714	1.705927	0.000922	1.015326	0.441330

Perilaku Konsumen:

- Pembelian pada produk 10624 dan 10609 cenderung akan membeli produk 10611

Sehingga, penataan barang di rak pada produk 10624 dan 10609 didekatkan dengan produk 10611

minimial nilai lift adalah 1



Kesimpulan

Analisis Deret Waktu

1. Memahami pola musiman penjualan:

- Menganalisis data historis penjualan untuk mengidentifikasi pola musiman (misalnya, apakah penjualan meningkat saat musim remodeling rumah).
- Memahami faktor-faktor yang memengaruhi pola musiman tersebut.

2. Fokus pada penjualan saat musim tinggi:

- Mengoptimalkan strategi pemasaran dan operasional saat penjualan meningkat di musim tertentu.
- Memastikan ketersediaan stok dan kapasitas produksi yang memadai.
- Mengalokasikan sumber daya secara efektif untuk memenuhi permintaan yang tinggi.

Analisis Deret Waktu

3. Menciptakan permintaan di luar musim tinggi:

- Menawarkan diskon atau promosi untuk mendorong penjualan di luar musim.
- Memanfaatkan media sosial dan digital marketing untuk menjangkau pelanggan potensial.
- Mengembangkan program loyalitas pelanggan untuk mendorong pembelian berulang.

4. Mengoptimalkan operasional dan efisiensi biaya:

- Mengatur inventori dan rantai pasokan secara efisien untuk meminimalkan biaya.
- Mengautomasi proses-proses operasional untuk meningkatkan produktivitas.

Analisis Algoritma Apriori

- Dalam melakukan **penataan barang** pada rak penjualan, cukup dengan melihat **antecedent** dan **consequent** pada setiap kategori produk hasil dari analisis asosiasi (Apriori) yang mana dapat **meningkatkan penjualan** berdasarkan perilaku konsumen
- **Strategi Bundling** dan **Promosi** Berdasarkan Pola Pembelian Konsumen
- Toko dapat mendesain penataan rak yang lebih efisien dan strategis. Ini tidak hanya mempermudah konsumen dalam menemukan dan membeli produk yang mereka butuhkan, tetapi juga **meningkatkan penjualan** melalui **cross-selling** dan **strategi promosi** yang lebih tepat sasaran



Thank You

Any Question?

