

LAPORAN PRAKTIKUM
Praktikum Eksplorasi dan Visualisasi Data
Explorasi Data Analisis dan Visualisasi

Dosen Pengampu : Edi Satriyanto S.Si., M.Si



Disusun Oleh :
Dinda Ayu Permatasari – 3323600012
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
PROGRAM STUDI SAINS DATA TERAPAN
TAHUN 2024/2025

A. Explorasi Data Analisis

1. Memahami Struktur Data

a) Import Library

Script

```
1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4 import numpy as np
```

Analisis

Pada Langkah pertama, kita menyiapkan library yang diperlukan, dalam hal ini memakai library pandas, seaborn, matplotlib dan numpy untuk kebutuhan analysis data dan visualisasi data.

b) Loading Data

Script

```
1 df = pd.read_excel("Tokokita.xlsx")
```

Analisis

Selanjutnya, saya memuat data dari file Excel (Tokokita.xlsx) ke dalam objek DataFrame Pandas yang disimpan dalam variabel df.

c) Menampilkan Sample Data

Script

```
df.head()
```

Output

	Tanggal	Kode Barang	Nama Barang	Kategori	Sat	Qty	Jual(Rp)	Beli(Rp)	Penjualan	Pembelian	Region	City
0	2020-01-03	8.9927779e+12	GLADE GANTUNG MOUNTAIN PINE 75+10GR	PEWANGI	pcs	77.0	11000.0	9600.0	847000.0	739200.0	West Java	Depok
1	2020-01-04	2.011040e+05	SARDEN ABC KECIL	DAGING	can	244.0	10000.0	7500.0	2440000.0	1830000.0	Jakarta	North Jakarta
2	2020-01-06	2.028080e+05	BANGO KECAP BOTOL 135ml	KECAP	btl	100.0	11000.0	9570.0	1100000.0	957000.0	Jakarta	South Jakarta
3	2020-01-08	8.992747e+12	VIXAL HARUM SEGAR 780ML	PEMBERSIH LANTAI	pcs	169.0	17000.0	15290.0	2873000.0	2584010.0	Jakarta	North Jakarta
4	2020-01-10	7.000090e+05	RINSO DETERGENT ANTI NODA 800GR	DETERGEN	pcs	50.0	21800.0	19580.0	1090000.0	979000.0	West Java	Bekasi

Analisis

Lalu saya menggunakan fungsi `head()` untuk menampilkan 5 baris pertama dari dataset yang sudah saya load dari langkah sebelumnya. Dari output diatas kita bisa melihat apa saja kolom, baris dan isi dari dataset yang kita miliki yaitu 'Tokokita.xlsx'.

d) Menampilkan Informasi Dataset

Script

```
df.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 801 entries, 0 to 800
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   Tanggal     400 non-null   datetime64[ns]
 1   Kode Barang 400 non-null   float64
 2   Nama Barang 400 non-null   object
 3   Kategori    400 non-null   object
 4   Sat         400 non-null   object
 5   Qty         400 non-null   float64
 6   Jual(Rp)    399 non-null   float64
 7   Beli(Rp)    399 non-null   float64
 8   Penjualan   400 non-null   float64
 9   Pembelian   400 non-null   float64
10   Region      801 non-null   object
11   City        801 non-null   object
dtypes: datetime64[ns](1), float64(6), object(5)
```

Analisis

Disini saya juga menggunakan fungsi `info()` untuk menampilkan informasi dari dataset 'Tokokita.xlsx'. Kita bisa mengetahui jumlah datanya (ada 801 baris dan 12 kolom) serta tipe data untuk setiap kolomnya.

2. Memeriksa Nilai yang Hilang (Missing Values) dan Data Duplikat

a) Identifikasi jumlah nilai yang hilang di setiap kolom

Script	
<pre>df.isnull().sum()</pre>	
Output	
<pre>Tanggal 401 Kode Barang 401 Nama Barang 401 Kategori 401 Sat 401 Qty 401 Jual(Rp) 402 Beli(Rp) 402 Penjualan 401 Pembelian 401 Region 0 City 0 dtype: int64</pre>	
Analisis	
<p>Kemudian untuk proses eksplorasi data analisis selanjutnya, yaitu memeriksa apakah ada data yang hilang dalam dataset kita (Missing Value). Hal ini perlu diperhatikan sebelum kita menggunakan data untuk analisis lebih lanjut. Oleh karena itu, pada kode diatas saya memeriksa jumlah nilai yang hilang (NaN) di setiap kolom DataFrame df. Dan hasilnya menunjukkan bahwa kolom Region dan City memiliki nilai di kolom yang lengkap. Ini menunjukkan bahwa tidak ada data yang hilang dalam kolom ini, sehingga kolom ini bisa dianggap terisi sepenuhnya. Sedangkan hampir semua kolom lainnya memiliki nilai yang hilang. Ini mencakup banyak kolom, sehingga penanganan nilai hilang menjadi sangat penting.</p>	

b) Menangani nilai yang hilang

Script	
<pre># remove all the rows that contain a missing value df = df.dropna() df.isnull().sum()</pre>	
Output	
<pre>Tanggal 0 Kategori 0 Qty 0 Jual(Rp) 0 Beli(Rp) 0 Penjualan 0 Pembelian 0 Region 0 City 0 Month-Year 0 dtype: int64</pre>	
Analisis	
<p>Lalu untuk menangani nilai yang hilang, saya menggunakan dropna() untuk menghapus semua baris dalam DataFrame df yang mengandung nilai hilang (NaN). Dengan demikian, setelah baris-baris ini dihapus, tidak ada baris yang memiliki nilai hilang di DataFrame yang tersisa. Dan fungsi isnull().sum() untuk memeriksa jumlah nilai yang hilang di setiap kolom. Output menunjukkan bahwa semua kolom sekarang memiliki 0 nilai yang hilang.</p>	

c) Memeriksa data duplikat

Script	
<pre>df[df.duplicated()]</pre>	
Output	

Tanggal	Kode Barang	Nama Barang	Kategori	Sat	Qty	Jual(Rp)	Beli(Rp)	Penjualan	Pembelian	Region	City
Analisis											
<p>Terakhir saya menggunakan fungsi duplicated() untuk memeriksa apakah ada data yang duplikat dalam dataset. Dan hasil dari outputnya menunjukkan bahwa tidak ada baris yang memiliki nilai duplikat.</p>											

3. Memahami Distribusi Data

a) Descriptive Statistics

Script							
<pre>df.describe()</pre>							
Output							
	Tanggal	Kode Barang	Qty	Jual(Rp)	Beli(Rp)	Penjualan	Pembelian
count	399	3.990000e+02	399.000000	399.000000	399.000000	3.990000e+02	3.990000e+02
mean	2020-10-03 05:50:04.511278080	5.034385e+12	135.255639	22305.764411	18796.548037	3.216136e+06	2.713075e+06
min	2020-01-03 00:00:00	1.000010e+05	15.000000	750.000000	300.000000	1.875000e+04	1.010000e+04
25%	2020-06-11 12:00:00	5.012245e+05	81.500000	3750.000000	2958.333333	4.210000e+05	3.168750e+05
50%	2020-10-24 00:00:00	8.886008e+12	134.000000	12000.000000	9300.000000	1.155000e+06	9.646000e+05
75%	2021-02-05 00:00:00	8.996007e+12	192.500000	23000.000000	17812.500000	2.965050e+06	2.377072e+06
max	2021-04-20 00:00:00	8.993189e+13	250.000000	750000.000000	675000.000000	1.695000e+08	1.525500e+08
std	NaN	6.290020e+12	66.288010	62102.078630	55438.083715	1.202097e+07	1.078004e+07
Analisis							
<p>Untuk memahami distribusi data yang kita miliki, bisa memakai fungsi describe() untuk memberikan ringkasan statistik deskriptif dari kolom numerik dalam DataFrame df. Didalamnya mencakup metrik-metrik seperti count, mean, min, max, serta kuartil (25%, 50%, 75%).</p>							

4. Menghapus Kolom Yang Tidak Perlu

Script	
<pre>df.drop(['Kode Barang', 'Nama Barang', 'Sat'], axis=1, inplace=True) df</pre>	

Output

	Tanggal	Kategori	Qty	Jual(Rp)	Beli(Rp)	Penjualan	Pembelian	Region	City
0	2020-01-03	PEWANGI	77.0	11000.0	9600.000000	847000.0	7.392000e+05	West Java	Depok
1	2020-01-04	DAGING	244.0	10000.0	7500.000000	2440000.0	1.830000e+06	Jakarta	North Jakarta
2	2020-01-06	KECAP	100.0	11000.0	9570.000000	1100000.0	9.570000e+05	Jakarta	South Jakarta
3	2020-01-08	PEMBERSIH LANTAI	169.0	17000.0	15290.000000	2873000.0	2.584010e+06	Jakarta	North Jakarta
4	2020-01-10	DETERGEN	50.0	21800.0	19580.000000	1090000.0	9.790000e+05	West Java	Bekasi
...
395	2021-04-17	SNACK	132.0	9000.0	7500.000000	1188000.0	9.900000e+05	Jakarta	North Jakarta
396	2021-04-18	SNACK	131.0	2000.0	1630.000000	262000.0	2.135300e+05	Jakarta	Central Jakarta
397	2021-04-18	PEWANGI	206.0	1500.0	916.666667	309000.0	1.888333e+05	Jakarta	West Jakarta
398	2021-04-20	POPOK	67.0	20000.0	14300.000000	1340000.0	9.581000e+05	Jakarta	East Jakarta
399	2021-04-20	TISU	207.0	11000.0	7800.000000	2277000.0	1.614600e+06	Jakarta	East Jakarta

399 rows × 9 columns

Analisis

Kode `drop()` diatas untuk menghapus kolom-kolom tertentu dari DataFrame berfungsi untuk membersihkan data dan menyederhanakan analisis dengan hanya menyimpan kolom yang dianggap relevan. DataFrame yang tersisa kini siap untuk analisis lebih lanjut dan visualisasi data.

5. Mengubah Tipe Data

Script

```
1 # Mengubah tipe data dari float64 ke int
2 df['Qty'] = df['Qty'].astype(int)
3 df['Jual(Rp)'] = df['Jual(Rp)'].astype(int)
4 df['Beli(Rp)'] = df['Beli(Rp)'].astype(int)
5 df['Penjualan'] = df['Penjualan'].astype(int)
6 df['Pembelian'] = df['Pembelian'].astype(int)
7
8 print(df.dtypes)
```

Output

Tanggal	datetime64[ns]
Kategori	object
Qty	int32
Jual(Rp)	int32
Beli(Rp)	int32
Penjualan	int32
Pembelian	int32
Region	object
City	object
dtype:	object

Analisis

Kode diatas digunakan untuk mengubah tipe data dari float64 ke int32 pada beberapa kolom numerik: 'Qty', 'Jual(Rp)', 'Beli(Rp)', 'Penjualan', dan 'Pembelian'. Ini penting karena dalam konteks tertentu, nilai numerik seperti kuantitas dan harga lebih cocok disimpan sebagai bilangan bulat (int) dibandingkan bilangan desimal (float), terutama jika data tidak memerlukan representasi desimal.

6. Analisis Korelasi Antar Variabel

a) Menghitung korelasi

Script

```

1  # Pilih kolom numerik
2  df_numerik = df[['Qty', 'Jual(Rp)', 'Beli(Rp)', 'Penjualan', 'Pembelian']]
3
4  # Menghitung matriks korelasi
5  korelasi_matrix = df_numerik.corr()
6
7  # Menampilkan matriks korelasi
8  print("Matriks Korelasi:")
9  print(korelasi_matrix)

```

Output

Matriks Korelasi:

	Qty	Jual(Rp)	Beli(Rp)	Penjualan	Pembelian
Qty	1.000000	0.048500	0.046577	0.188084	0.180143
Jual(Rp)	0.048500	1.000000	0.997641	0.903781	0.901388
Beli(Rp)	0.046577	0.997641	1.000000	0.905602	0.907216
Penjualan	0.188084	0.903781	0.905602	1.000000	0.998179
Pembelian	0.180143	0.901388	0.907216	0.998179	1.000000

Analisis

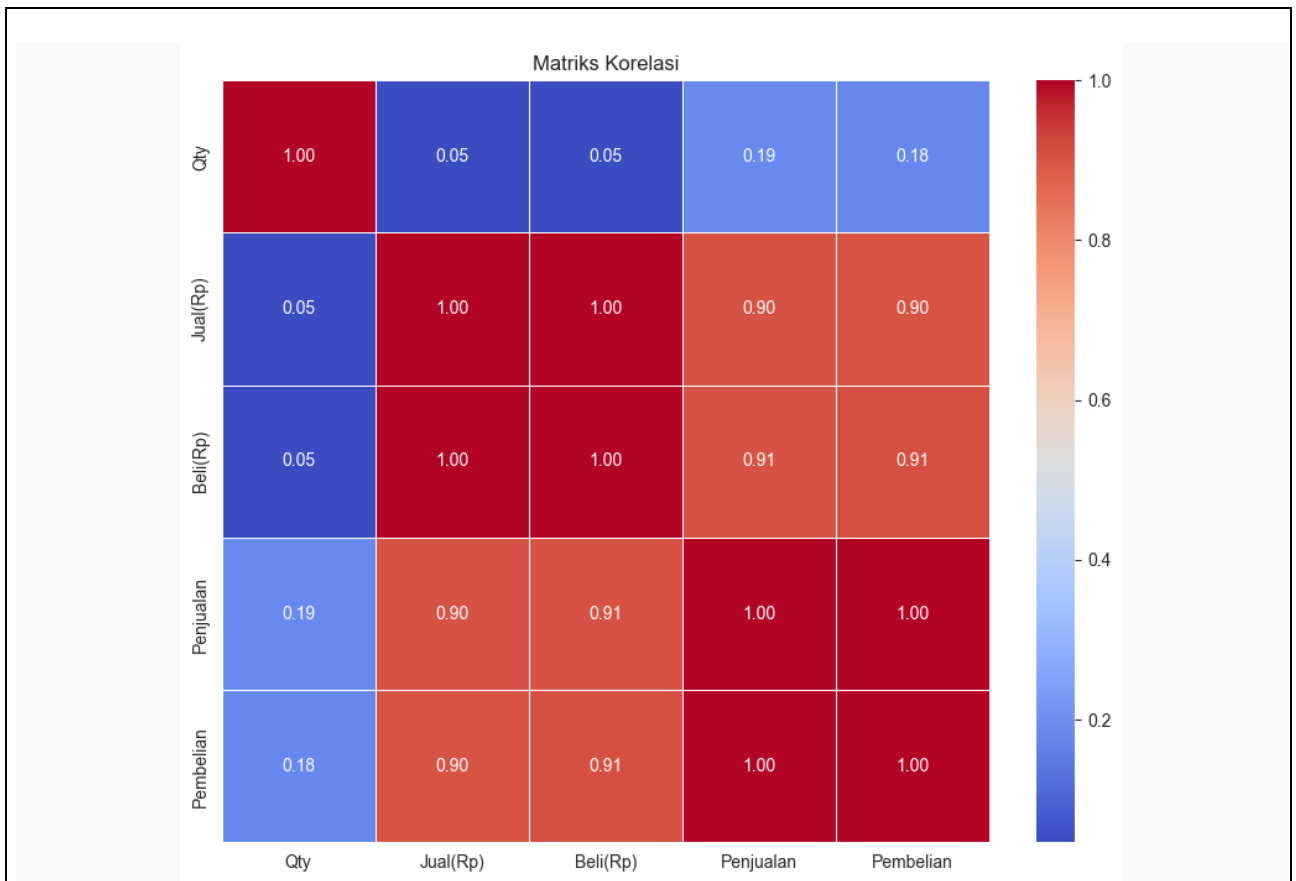
Kode diatas digunakan untuk menghitung dan menampilkan matriks korelasi untuk beberapa variabel numerik dalam dataset. Output menunjukkan bahwa terdapat hubungan yang sangat kuat antara harga jual, harga beli, penjualan, dan pembelian. Korelasi yang lebih rendah antara kuantitas dan variabel lainnya mengindikasikan bahwa jumlah barang yang terjual tidak selalu berkorelasi erat dengan harga atau total nilai transaksi.

b) Visualisasi korelasi (heatmap)

Script

```
1 plt.figure(figsize=(10, 8))
2 sns.heatmap(korelasi_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
3 plt.title('Matriks Korelasi')
4 plt.show()
```

Output



Analisis

Visualisasi heatmap ini efektif dalam memberikan gambaran visual dari hubungan korelasi antar variabel. Warna yang lebih merah menunjukkan korelasi positif yang kuat, sedangkan warna biru menunjukkan korelasi negatif. Dalam heatmap ini, ada warna yang sangat merah antara Jual(Rp) dan Beli(Rp), Penjualan dan Pembelian, serta hubungan lainnya yang kuat.

7. Menyimpan Data Setelah Proses Pembersihan

Script

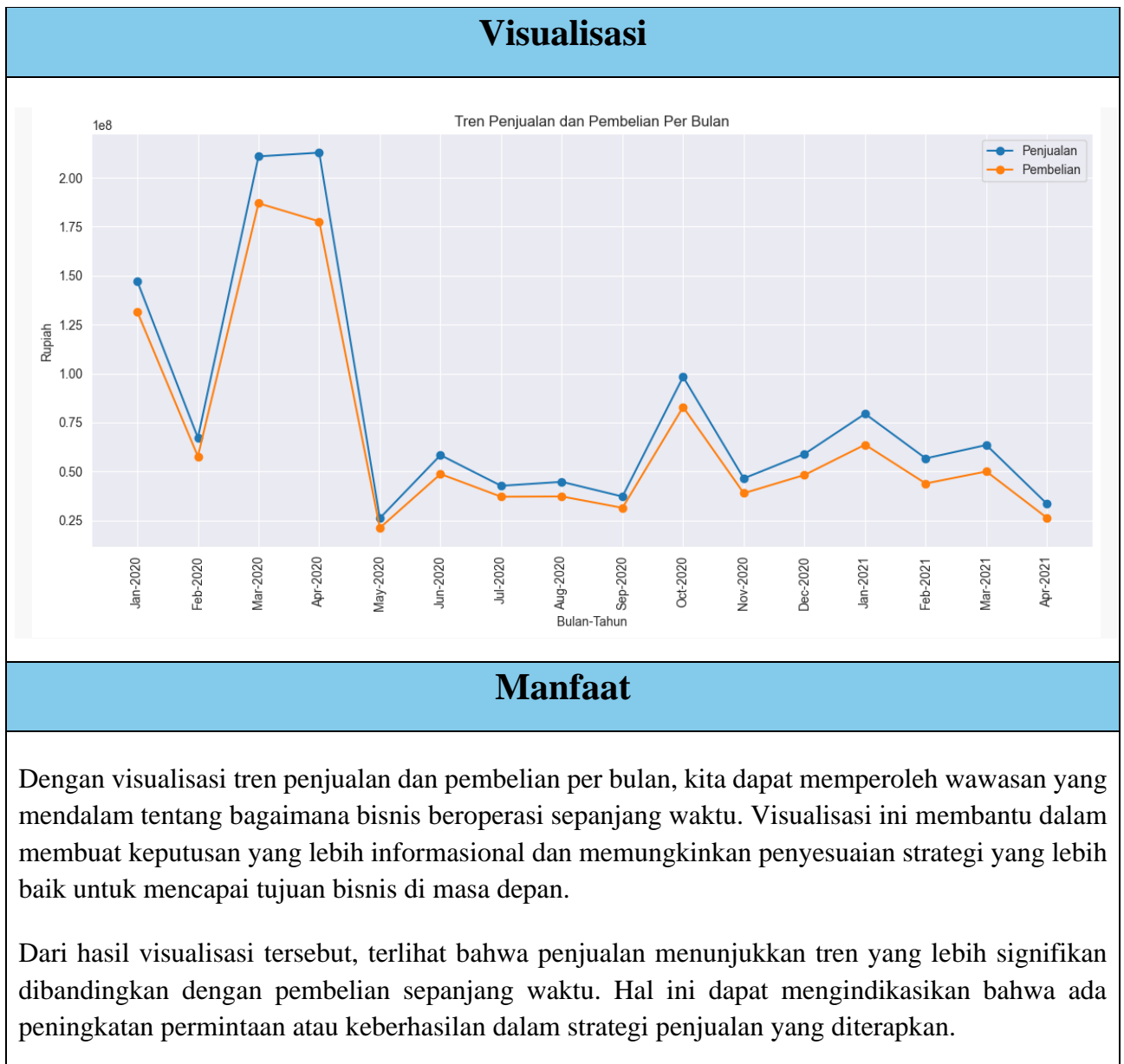
```
df.to_excel('Tokokita_clean.xlsx', index=False)
```

Analisis

Karna proses eksplorasi data analisis sudah kita lakukan sebelumnya, maka kita perlu menyimpan file yang sudah dibersihkan tersebut dan membuat file baru Bernama 'Tokokita_clean.xlsx'. File ini akan dibuat di direktori kerja saat ini. File ini akan berisi data yang telah dibersihkan dan diformat sesuai dengan perubahan yang telah dilakukan pada DataFrame df. Menyimpan data yang telah dibersihkan sangat penting untuk menjaga hasil analisis dan menghindari melakukan pembersihan ulang setiap kali data tersebut diperlukan.

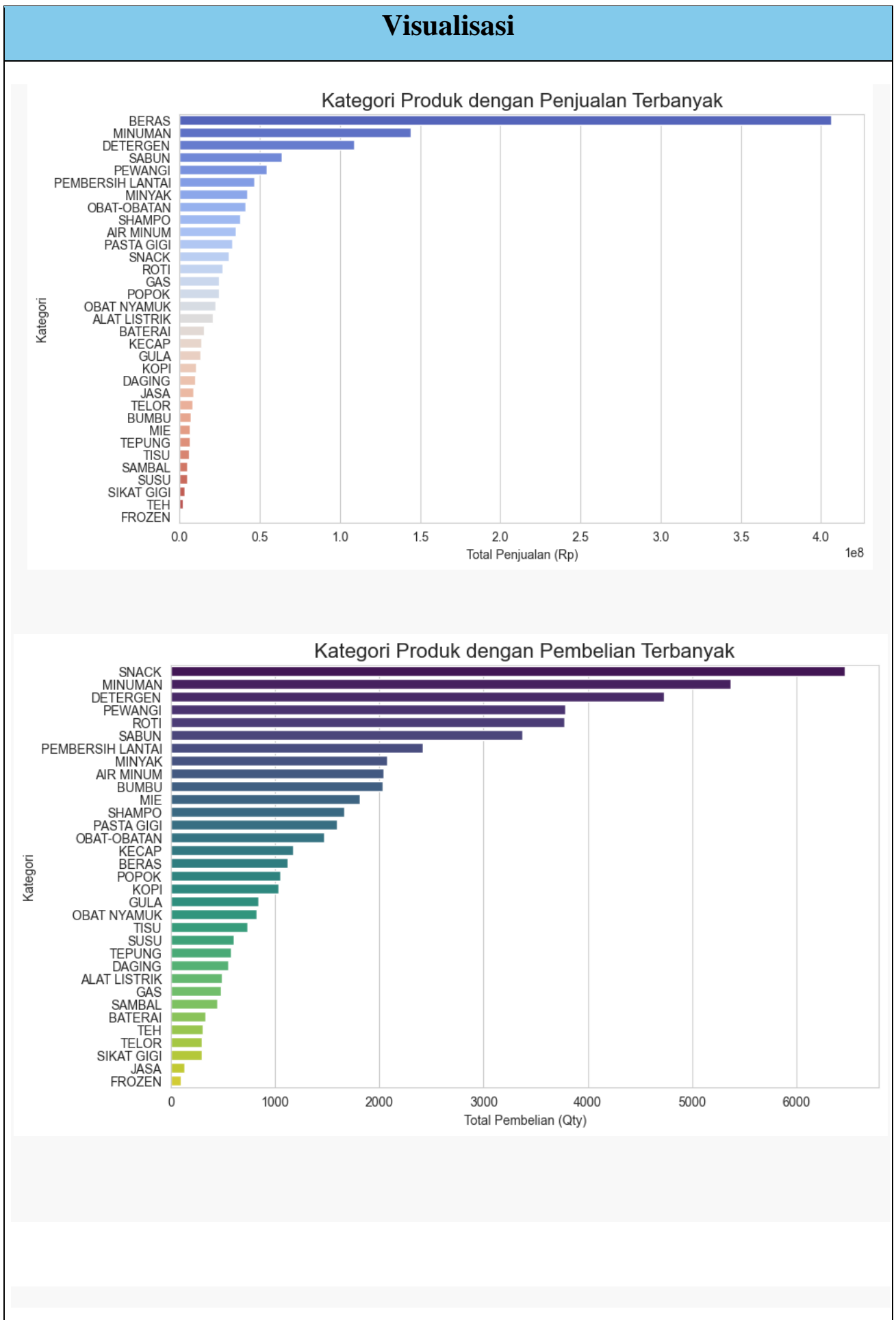
B. Visualisasi Data

1. Tren penjualan dan pembelian dalam per bulan



2. Produk dengan penjualan dan pembelian terbanyak

Visualisasi



Manfaat
<p>Visualisasi ini memberikan pandangan yang jelas dan terperinci tentang produk mana yang paling berkontribusi terhadap penjualan dan pembelian, yang dapat membantu dalam pengambilan keputusan dan perencanaan bisnis yang lebih baik.</p> <p>Produk seperti beras lalu diikuti dengan minuman, detergen, sabun, dan pewangi menunjukkan angka penjualan yang tinggi. Ini menunjukkan bahwa produk-produk ini sangat dibutuhkan dan populer di kalangan konsumen. Sedangkan snack, minuman, detergen, pewangi, dan roti muncul sebagai produk dengan pembelian terbanyak. Ini menunjukkan bahwa produk-produk ini sering dibeli dalam jumlah besar atau lebih sering dibandingkan produk lainnya.</p> <p>Hal ini bisa dipertimbangkan agar perusahaan mempertimbangkan untuk meningkatkan stok dan memperluas variasi produk dalam kategori dengan penjualan tertinggi seperti beras dan minuman. Atau berdasarkan pola penjualan dan pembelian tersebut, perusahaan dapat merencanakan diversifikasi produk dan inovasi yang sesuai dengan tren pasar dan kebutuhan konsumen.</p>

3. Distribusi penjualan dan pembelian berdasarkan kota

Visualisasi

Distribusi Penjualan dan Pembelian Berdasarkan Kota

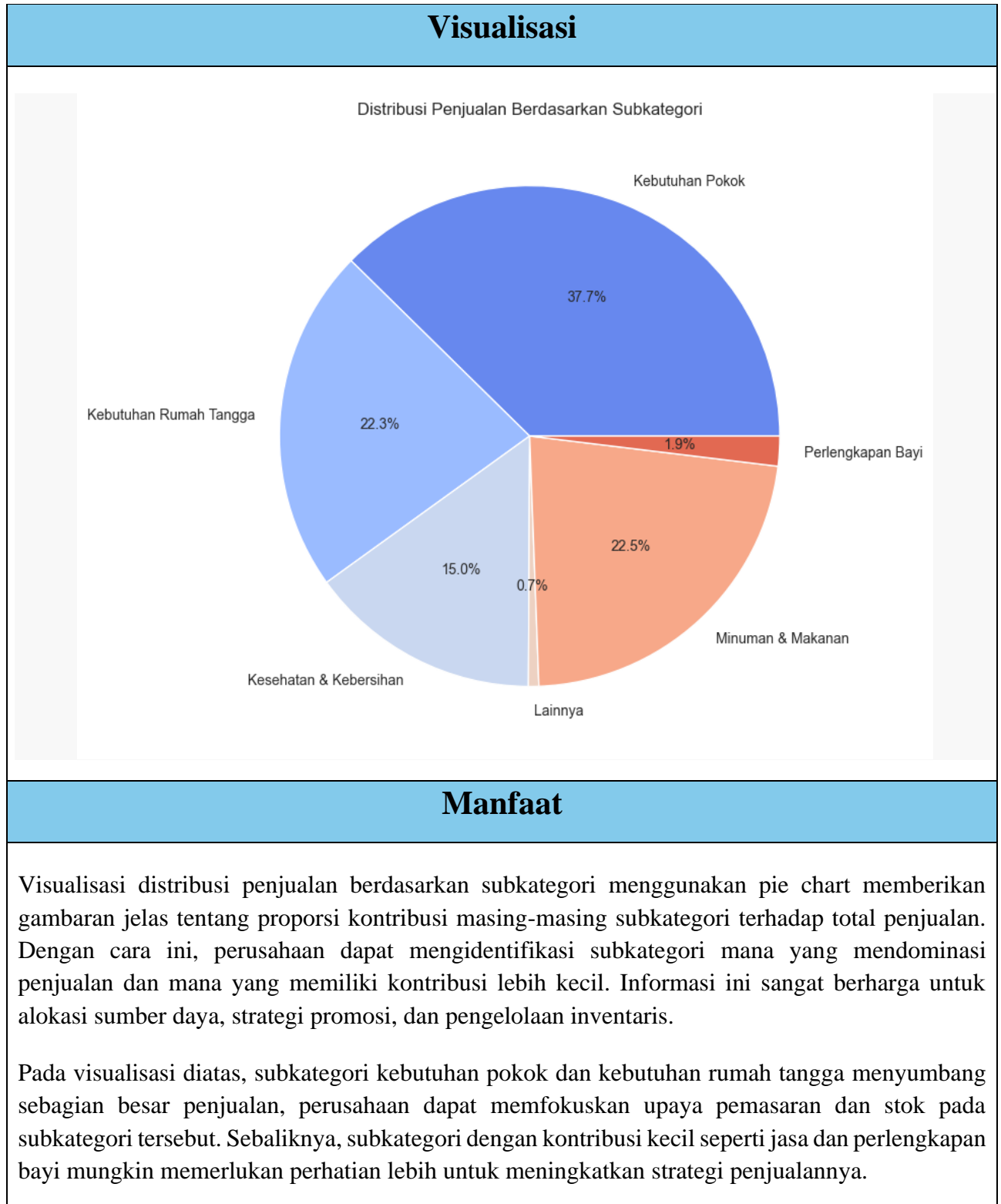
Kota	Penjualan (Rp)	Pembelian (Rp)
Bekasi	2.3e8	2.0e8
Central Jakarta	0.4e8	0.3e8
Depok	2.1e8	1.8e8
East Jakarta	3.8e8	3.2e8
North Jakarta	1.8e8	1.4e8
South Jakarta	0.2e8	0.2e8
South Tangerang	0.3e8	0.2e8
Tangerang	1.2e8	1.0e8
West Jakarta	0.5e8	0.4e8

Manfaat

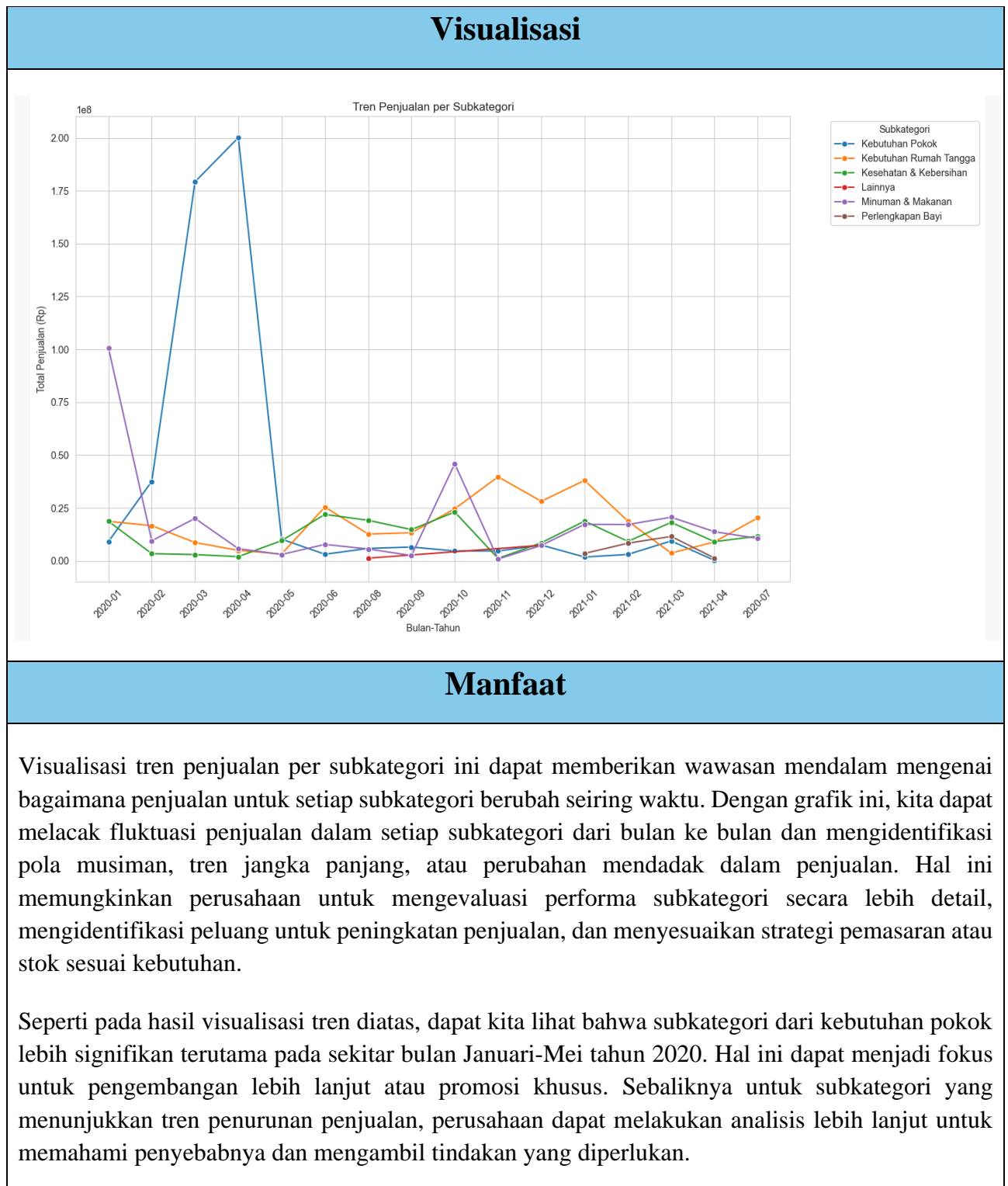
Visualisasi distribusi penjualan dan pembelian berdasarkan kota dapat memberikan wawasan yang mendalam mengenai performa pasar di berbagai lokasi. Dengan mengidentifikasi kota-kota dengan total penjualan dan pembelian tertinggi, perusahaan dapat merencanakan strategi pemasaran, pengelolaan stok, dan logistik yang lebih efisien. Selain itu, data ini dapat membantu dalam

menyesuaikan penawaran produk sesuai dengan kebutuhan lokal, serta merencanakan ekspansi pasar dengan lebih strategis.

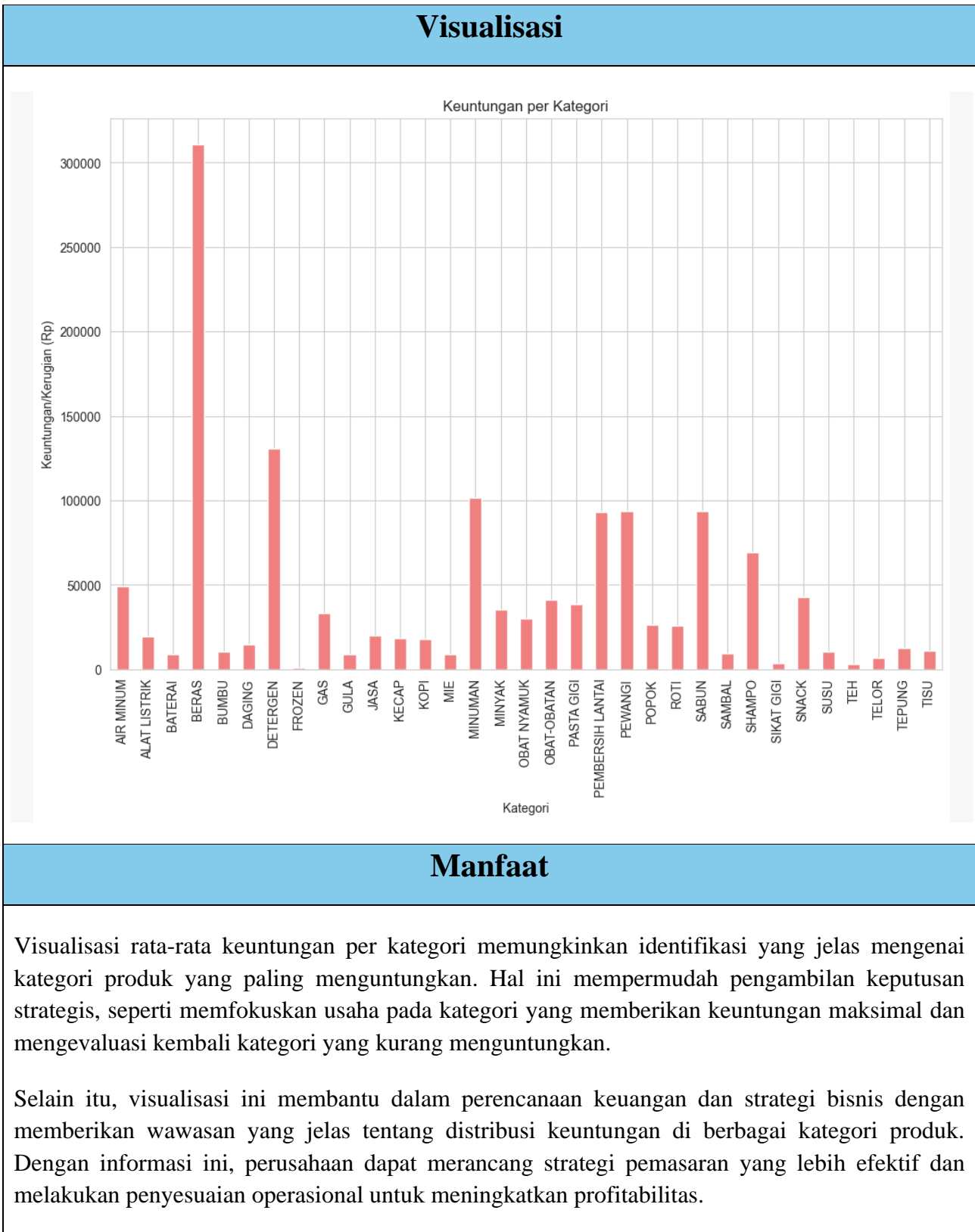
4. Distribusi penjualan berdasarkan sub-kategori



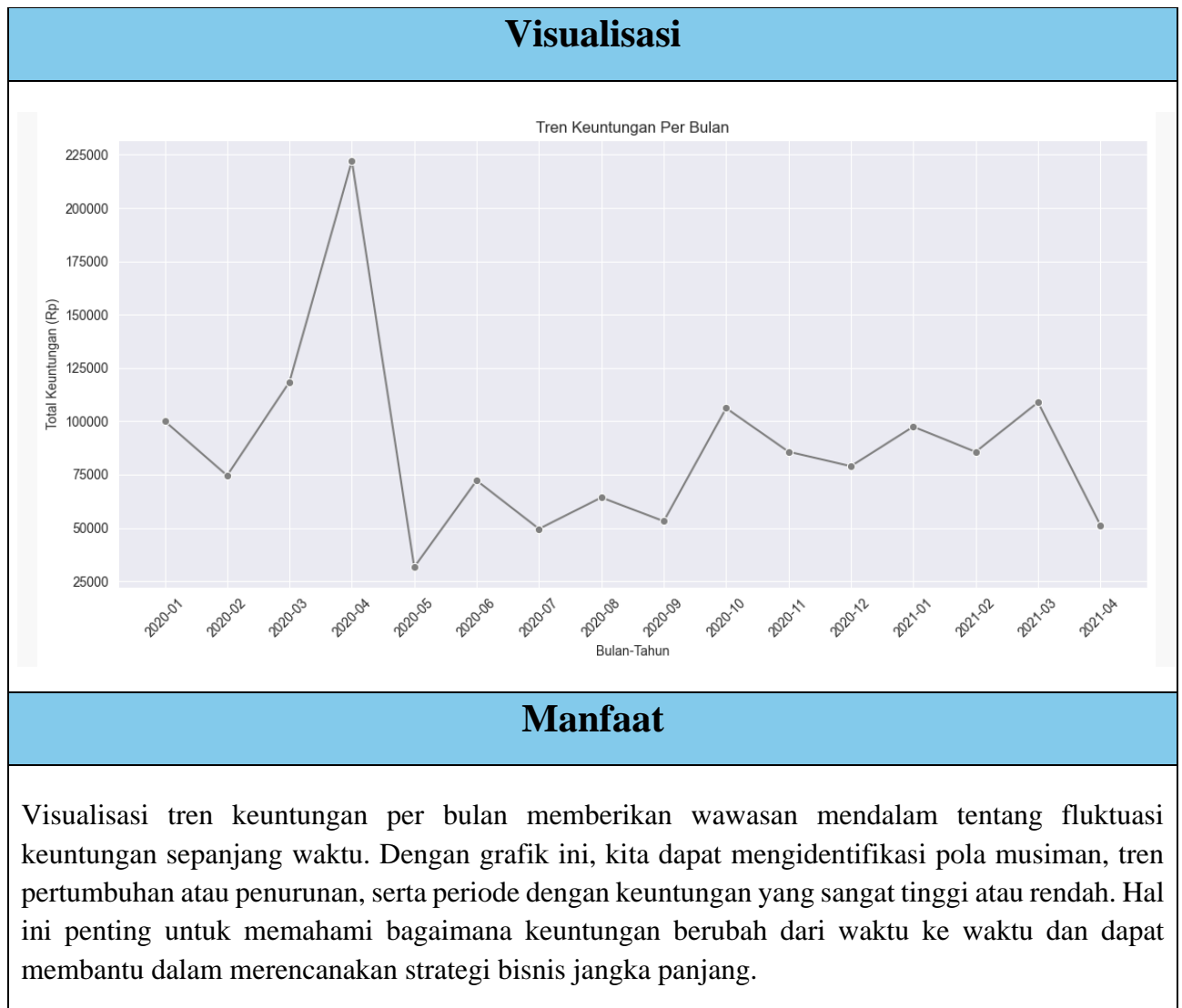
5. Tren penjualan per sub-kategori



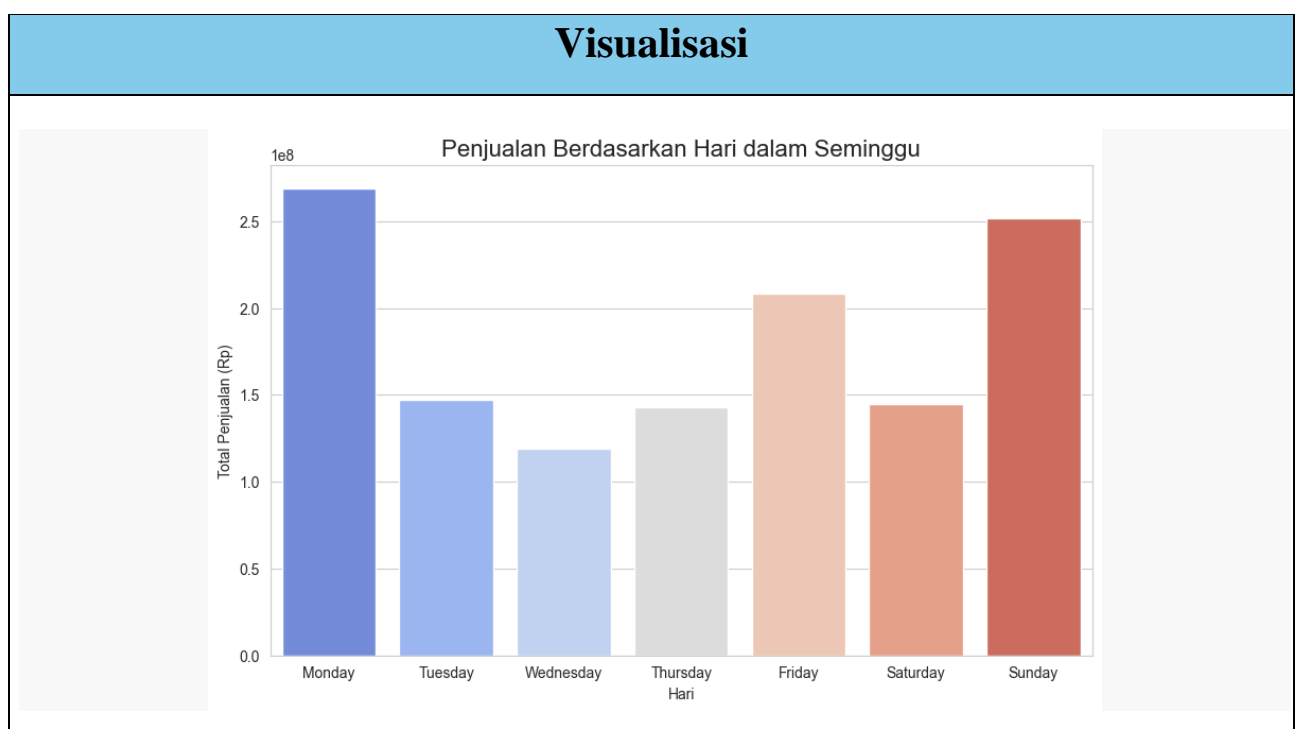
6. Analisis rata-rata keuntungan per sub-kategori



7. Analisis tren keuntungan per bulan



8. Analisis penjualan berdasarkan hari dalam seminggu



Manfaat

Visualisasi penjualan berdasarkan hari dalam seminggu membantu dalam memahami pola penjualan yang mungkin dipengaruhi oleh hari tertentu. Dengan informasi ini, perusahaan dapat mengidentifikasi hari-hari dengan penjualan tertinggi dan terendah, memungkinkan mereka untuk mengoptimalkan strategi pemasaran dan operasional sesuai dengan pola penjualan.

Pada visualisasi diatas menunjukkan bahwa penjualan tertinggi pada hari Senin dan Minggu, dimana kedua hari tersebut merupakan waktu-waktu yang signifikan untuk penjualan. Hal ini mengindikasikan adanya permintaan yang tinggi pada awal dan akhir minggu, yang dapat mempengaruhi strategi bisnis. Perusahaan harus mempertimbangkan untuk meningkatkan promosi dan persediaan produk pada hari-hari ini, serta memastikan layanan pelanggan yang optimal.