



FINAL PROJECT PYTHON

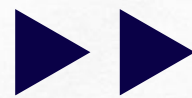
Analisis Data Penjualan E-Commerce:
Optimalisasi Strategi Berbasis Data

Kelompok 4B





ANGGOTA



Amos Christ Kevin

Dani Harsalisman Febrian Putra

Irfan Rasyid

Muhamad Fatih Praga Ilhaq

Reza Muhamad Ricky

Adinda Aulia Hudianti

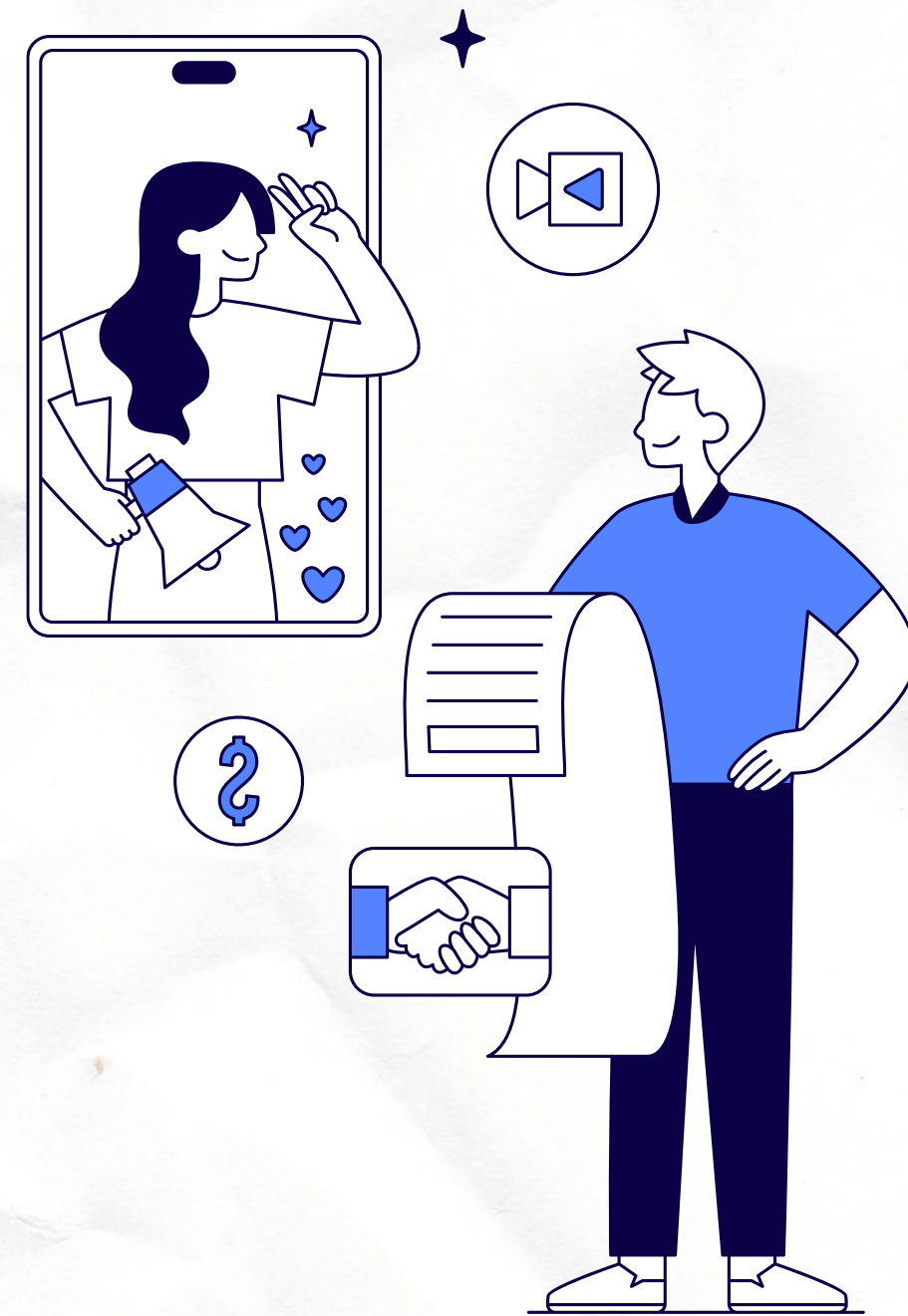
Dickna Nenden Woro
Ramadityawati

Hanif Dwi Satria

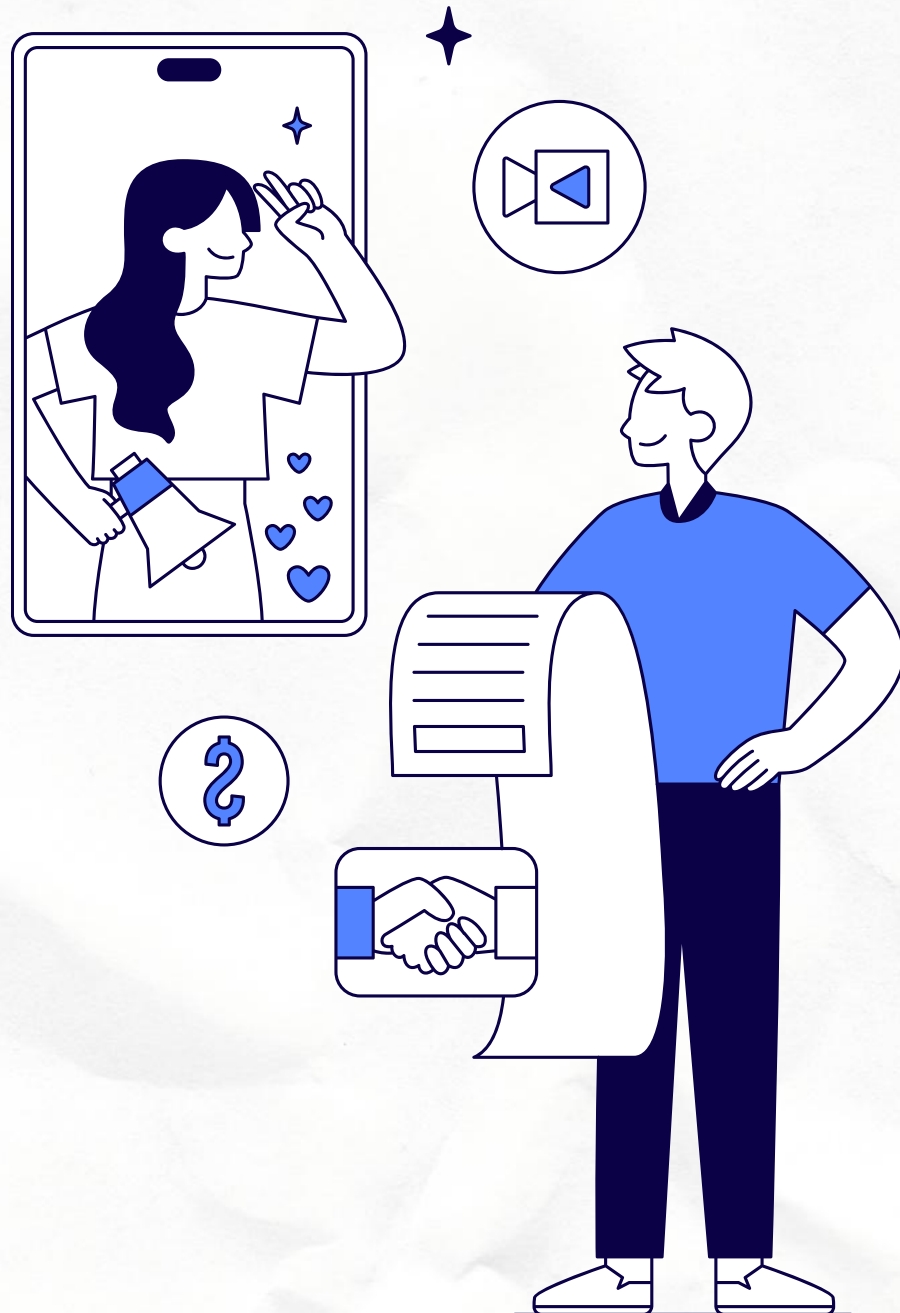
Marchenko Svesda

Nando Rifki Utama Putra

Sharfina Zahra Akbar



PROJECT OVERVIEW



Objective

Project ini bertujuan untuk menganalisis data penjualan e-commerce untuk memberikan wawasan strategis bagi pengambilan keputusan bisnis. Analisis mencakup evaluasi performa produk, perilaku pelanggan, dan efektivitas strategi promosi serta membantu penyelesaian permasalahan bisnis yang spesifik.

Description

Dataset yang digunakan mencakup informasi terkait transaksi, produk, pelanggan, dan metode pembayaran. Proses pengolahan data dilakukan menggunakan Python untuk analisis mendalam dan SQL untuk penggabungan data yang presisi. Dengan pendekatan ini, proyek bertujuan untuk mengidentifikasi pola penjualan serta perilaku pelanggan guna mendukung pengambilan keputusan strategis yang berbasis data.

Goals

1. Membersihkan dan mengintegrasikan dataset untuk memudahkan analisis.
2. Mengidentifikasi tren utama dalam penjualan, seperti produk populer dan pola perilaku pelanggan.
3. Memberikan rekomendasi strategis yang relevan berdasarkan temuan analisis data.



DATASET OVERVIEW

Data yang digunakan adalah data yang berasal dari Tokopedia (bukan data sesungguhnya). Mengenai penjelasan dataset adalah sebagai berikut:

variable	class	description
order_detail:		
id	object	angka unik dari order / id_order
customer_id	object	angka unik dari pelanggan
order_date	object	tanggal saat dilakukan transaksi
sku_id	object	angka unik dari produk (sku adalah stock keeping unit)
price	int64	harga yang tertera pada tagging harga
qty_ordered	int64	jumlah barang yang dibeli oleh pelanggan
before_discount	float64	nilai harga total dari produk (price * qty_ordered)
discount_amount	float64	nilai diskon product total
after_discount	float64	nilai harga total produk ketika sudah dikurangi dengan diskon
is_gross	int64	menunjukkan pelanggan belum membayar pesanan
is_valid	int64	menunjukkan pelanggan sudah melakukan pembayaran
is_net	int64	menunjukkan transaksi sudah selesai
payment_id	int64	angka unik dari metode pembayaran

sku_detail:		
id	object	angka unik dari produk (dapat digunakan untuk key saat join)
sku_name	object	nama dari produk
base_price	float64	harga barang yang tertera pada tagging harga / price
cogs	int64	cost of goods sold / total biaya untuk menjual 1 produk
category	object	kategori produk

customer_detail:		
id	object	angka unik dari pelanggan
registered_date	object	tanggal pelanggan mulai mendaftarkan diri sebagai anggota

payment_detail:		
id	int64	angka unik dari metode pembayaran
payment_method	object	metode pembayaran yang digunakan





DATA PREPARATION

1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas.tseries.offsets import BDay
```

Sebelum melakukan analisis data, kita perlu mengimpor beberapa library yang akan digunakan dalam pemrosesan data dan visualisasi. Berikut adalah library yang diperlukan:

- **import pandas as pd**: untuk manipulasi dan analisis data, seperti membaca data dari file CSV dan mengelola data dalam bentuk tabel (DataFrame).
- **import numpy as np**: untuk komputasi numerik dan operasi array, seperti perhitungan statistik atau operasi matematika lainnya.
- **import matplotlib.pyplot as plt**: untuk mengimpor modul pyplot dari matplotlib yang digunakan dalam pembuatan visualisasi data, seperti grafik dan plot.
- **import seaborn as sns**: untuk mengimpor library seaborn yang menyediakan cara mudah untuk membuat visualisasi statistik, seperti heatmap atau scatterplot.
- **from pandas.tseries.offsets import BDay**: untuk mengimpor BDay dari pandas.tseries.offsets yang digunakan dalam manipulasi tanggal, khususnya untuk menambahkan atau mengurangi hari kerja dalam perhitungan waktu.

2

```
#Sumber data yang digunakan
path_od = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/order_detail.csv"
path_pd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/payment_detail.csv"
path_cd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/customer_detail.csv"
path_sd = "https://raw.githubusercontent.com/dataskillsboost/FinalProjectDA11/main/sku_detail.csv"
df_od = pd.read_csv(path_od)
df_pd = pd.read_csv(path_pd)
df_cd = pd.read_csv(path_cd)
df_sd = pd.read_csv(path_sd)
```

Mendefinisikan dan menyimpan URL dari file CSV yang berisi data order detail, payment detail, customer detail, sku detail. kemudian menggunakan fungsi `pd.read_csv()` dari pandas untuk mengimpor data dari file CSV yang telah didefinisikan, lalu didefinisikan kembali sebagai DataFrame.

3

```
#Menjalankan SQL di Colab
from sqlite3 import connect
conn = connect(':memory:')
df_od.to_sql('order_detail', conn, index=False, if_exists='replace')
df_pd.to_sql('payment_detail', conn, index=False, if_exists='replace')
df_sd.to_sql('sku_detail', conn, index=False, if_exists='replace')
df_cd.to_sql('customer_detail', conn, index=False, if_exists='replace')
```

Menjalankan SQL di Colab

- **from sqlite3 import connect**: Mengimpor modul untuk koneksi ke database SQLite.
- **conn = connect(':memory:')**: Membuat koneksi ke database SQLite yang disimpan dalam memori.
- **df_od.to_sql('order_detail', conn, index=False, if_exists='replace')**: Mengimpor DataFrame `df_od` ke dalam tabel SQL `order_detail`. Langkah serupa dilakukan untuk DataFrame lainnya: `df_pd`, `df_sd`, dan `df_cd`.



DATA PREPARATION

4

```
#Query SQL untuk menggabungkan data
df = pd.read_sql("""
SELECT
    order_detail.*,
    payment_detail.payment_method,
    sku_detail.sku_name,
    sku_detail.base_price,
    sku_detail.cogs,
    sku_detail.category,
    customer_detail.registered_date
FROM order_detail
LEFT JOIN payment_detail
    on payment_detail.id = order_detail.payment_id
LEFT JOIN sku_detail
    on sku_detail.id = order_detail.sku_id
LEFT JOIN customer_detail
    on customer_detail.id = order_detail.customer_id
""", conn)
```

Query SQL untuk menggabungkan data

1. Menjalankan SQL Query `df = pd.read_sql(""" ... """, conn)`: untuk menggabungkan data dari beberapa tabel dan menyimpannya dalam DataFrame df.

2. Penjelasan Query:

- **SELECT**: Memilih kolom yang diperlukan dari tabel:
 - **order_detail.***: Semua kolom dari tabel order_detail.
 - **payment_detail.payment_method**: Kolom metode pembayaran dari tabel payment_detail.
 - **sku_detail.sku_name, base_price, cogs, category**: Kolom terkait produk dari tabel sku_detail.
 - **customer_detail.registered_date**: Tanggal pendaftaran pelanggan dari tabel customer_detail.
- **LEFT JOIN**: Digunakan untuk menggabungkan tabel berdasarkan ID yang sesuai:
 - payment_detail dengan order_detail berdasarkan payment_id.
 - sku_detail dengan order_detail berdasarkan sku_id.
 - customer_detail dengan order_detail berdasarkan customer_id.

Final Project Python Kelompok B4

5

```
[11] #Mengubah tipe data agar mudah dilakukan pengolahan data
df = df.astype({"before_discount": 'int', "discount_amount": 'int', "after_discount": 'int', "base_price": 'int'})
df.dtypes

#Mengubah tipe kolom Date menjadi Datetime
df['order_date'] = pd.to_datetime(df['order_date'])
df['registered_date'] = pd.to_datetime(df['registered_date'])
df.dtypes
```

id	object
customer_id	object
order_date	object
sku_id	object
price	int64
qty_ordered	int64
before_discount	float64
discount_amount	float64
after_discount	float64
is_gross	int64
is_valid	int64
is_net	int64
payment_id	int64
payment_method	object
sku_name	object
base_price	float64
cogs	int64
category	object
registered_date	object



id	object
customer_id	object
order_date	datetime64[ns]
sku_id	object
price	int64
qty_ordered	int64
before_discount	int64
discount_amount	int64
after_discount	int64
is_gross	int64
is_valid	int64
is_net	int64
payment_id	int64
payment_method	object
sku_name	object
base_price	int64
cogs	int64
category	object
registered_date	datetime64[ns]

1. **Mengubah tipe data** kolom yang sebelumnya bertipe **float64** menjadi **int** (integer). Mengubah tipe data ke int lebih efisien untuk operasi matematis, terutama jika nilai desimal tidak diperlukan (misalnya, diskon dan harga dasar biasanya berupa angka bulat).

2. **Mengubah tipe data** kolom **date** menjadi **datetime**. Dengan tipe datetime, kita dapat dengan mudah melakukan operasi pada tanggal seperti perhitungan selisih tanggal, ekstraksi bagian tertentu dari tanggal (misalnya, bulan atau tahun), dan filter data berdasarkan tanggal.

► PROBLEM - 1

Dear Data Analyst,



Akhir tahun ini, perusahaan akan memberikan hadiah bagi pelanggan yang memenangkan kompetisi Festival Akhir Tahun. Tim Marketing membutuhkan bantuan untuk menentukan perkiraan hadiah yang akan diberikan pada pemenang kompetisi nantinya. Hadiah tersebut akan diambil dari TOP 5 Produk dari Kategori Mobiles & Tablets selama tahun 2022, dengan jumlah kuantitas penjualan (valid = 1) paling tinggi.

Mohon bantuan, untuk mengirimkan data tersebut sebelum akhir bulan ini ke Tim Marketing. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards

Tim Marketing





CODE PROBLEM - 1

Problem Approach

Untuk menentukan 5 produk teratas di kategori "Mobiles & Tablets" tahun 2022, pertama, kita memfilter transaksi yang valid dan sesuai kategori. Setelah itu, data difilter untuk transaksi di tahun 2022, dikelompokkan berdasarkan nama produk, dan dihitung total kuantitas yang terjual. Data kemudian diurutkan berdasarkan kuantitas penjualan tertinggi dan 5 produk teratas dipilih.

Table

```
# Mengubah tipe data order_date
df_od['order_date'] = pd.to_datetime(df_od['order_date'])

# 1. Memfilter data dengan valid = 1
df_valid = df_od[df_od['is_valid'] == 1]

# 2. Memfilter data dengan Category = Mobiles & Tablets
df_merged = df_valid.merge(df_sd, left_on = 'sku_id', right_on = 'id')
df_mobiles = df_merged[df_merged['category'] == 'Mobiles & Tablets']

# 3. Memfilter data pada transaksi selama 2022
df_2022 = df_mobiles[df_mobiles['order_date'].dt.year == 2022]

# 4. Gunakan groupby berdasarkan sku_name
grouped_data = df_2022.groupby('sku_name')['qty_ordered'].sum().reset_index()

# 5. Gunakan sort_values untuk mengurutkan data
sorted_data = grouped_data.sort_values(by='qty_ordered', ascending=False)

# 6. Gunakan head untuk menampilkan top 5 produk
top_5_products = sorted_data.head(5)

print(top_5_products)
```

Syntax Breakdown

- **df_od['is_valid'] == 1:** Memilih transaksi yang valid (is_valid = 1).
- **df_merged['category'] == 'Mobiles & Tablets':** Memilih transaksi dari kategori yang relevan.
- **df_mobiles['order_date'].dt.year == 2022:** Memilih transaksi yang terjadi pada tahun 2022.
- **.groupby('sku_name')['qty_ordered'].sum():** Mengelompokkan data berdasarkan nama produk dan menghitung total kuantitas yang terjual.
- **.sort_values(by='qty_ordered', ascending=False):** Mengurutkan produk berdasarkan kuantitas terjual tertinggi.
- **.head(5):** Mengambil 5 produk dengan kuantitas penjualan tertinggi.



CODE PROBLEM - 1

Problem Approach

Setelah memperoleh 5 produk teratas berdasarkan kuantitas penjualan, data divisualisasikan dengan bar plot. Grafik ini menampilkan kuantitas yang terjual pada sumbu X dan nama produk pada sumbu Y untuk memudahkan pemahaman perbandingan antara produk.

Visualization

```
# Set the Seaborn style (optional for aesthetics)
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(10, 6)) # Set the figure size
ax = sns.barplot(x='qty_ordered', y='sku_name', data= top_5_products, palette='viridis')

# Add labels and title
plt.title('Top 5 Products by Quantity Ordered', fontsize=16)
plt.xlabel('Quantity Ordered', fontsize=12)
plt.ylabel('Product SKU', fontsize=12)

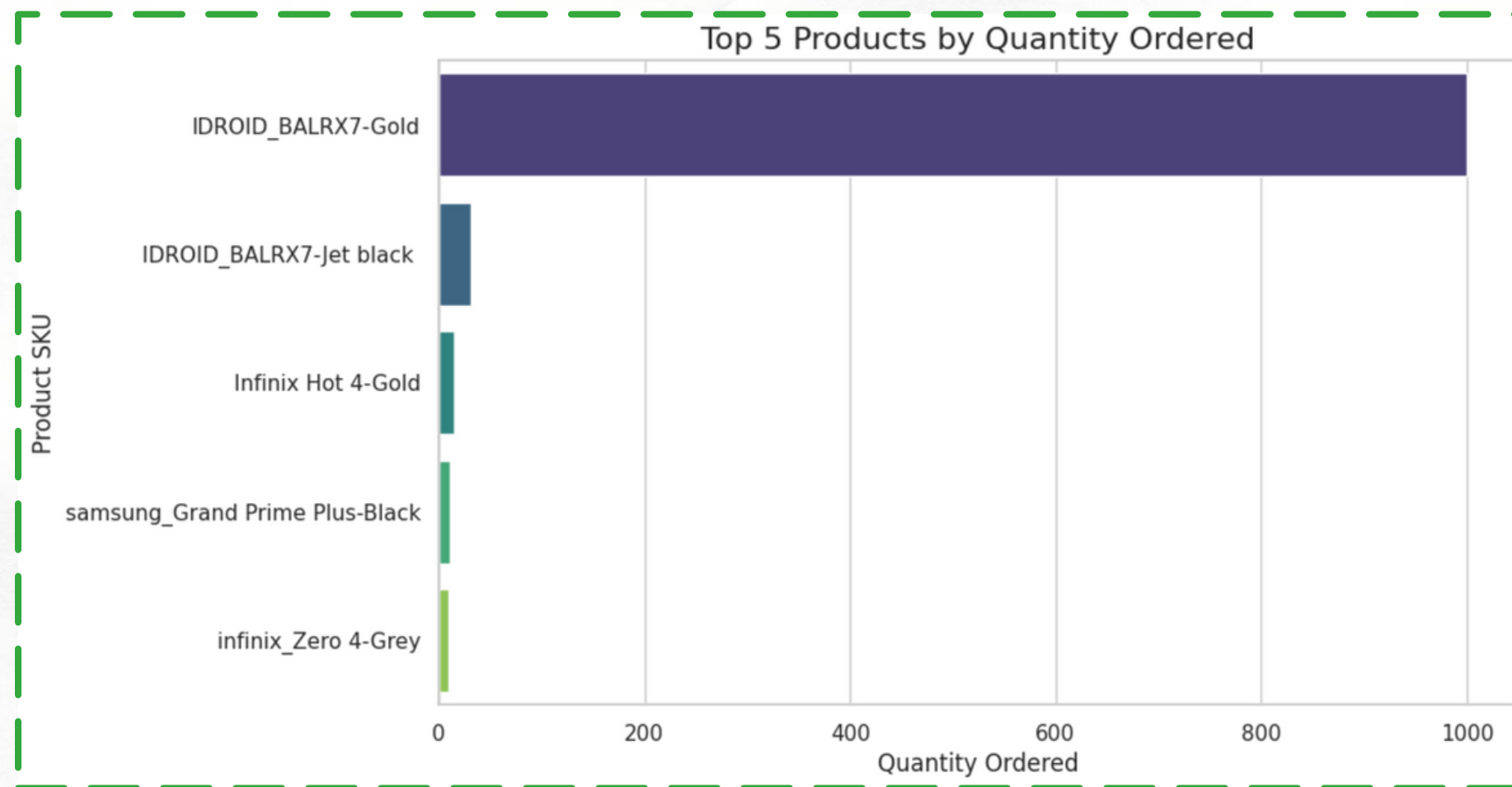
# Show the plot
plt.show()
```

Syntax Breakdown

- **sns.set(style="whitegrid"):** Mengatur tampilan visualisasi dengan grid putih.
- **sns.barplot(x='qty_ordered', y='sku_name', data=top_5_products, palette='viridis'):** Membuat bar plot dengan kuantitas yang terjual pada sumbu X dan nama produk pada sumbu Y. Palet warna 'viridis' digunakan untuk tampilan yang menarik.
- **plt.title(), plt.xlabel(), plt.ylabel():** Menambahkan judul, label sumbu X (kuantitas terjual), dan label sumbu Y (nama produk) untuk memberi konteks pada grafik.
- **plt.show():** Menampilkan grafik yang telah dibuat.



RESULT PROBLEM - 1



IDROIS_BALRX7-Gold merupakan produk teratas dalam penjualan kategori Mobiles & Tablets pada tahun 2022. Oleh karena itu, akan menjadi pilihan yang bagus sebagai hadiah utama. Produk lainnya, meskipun dengan angka penjualan yang lebih rendah, tetap dapat digunakan sebagai hadiah sekunder/pilihan sehingga dapat membuat struktur pembagian hadiah yang menarik dan adil untuk kompetisi Festival Akhir Tahun.

	sku_name	qty_ordered
1	IDROID_BALRX7-Gold	1000
2	IDROID_BALRX7-Jet black	31
3	Infinix Hot 4-Gold	15
43	samsung_Grand Prime Plus-Black	11
34	infinix_Zero 4-Grey	10



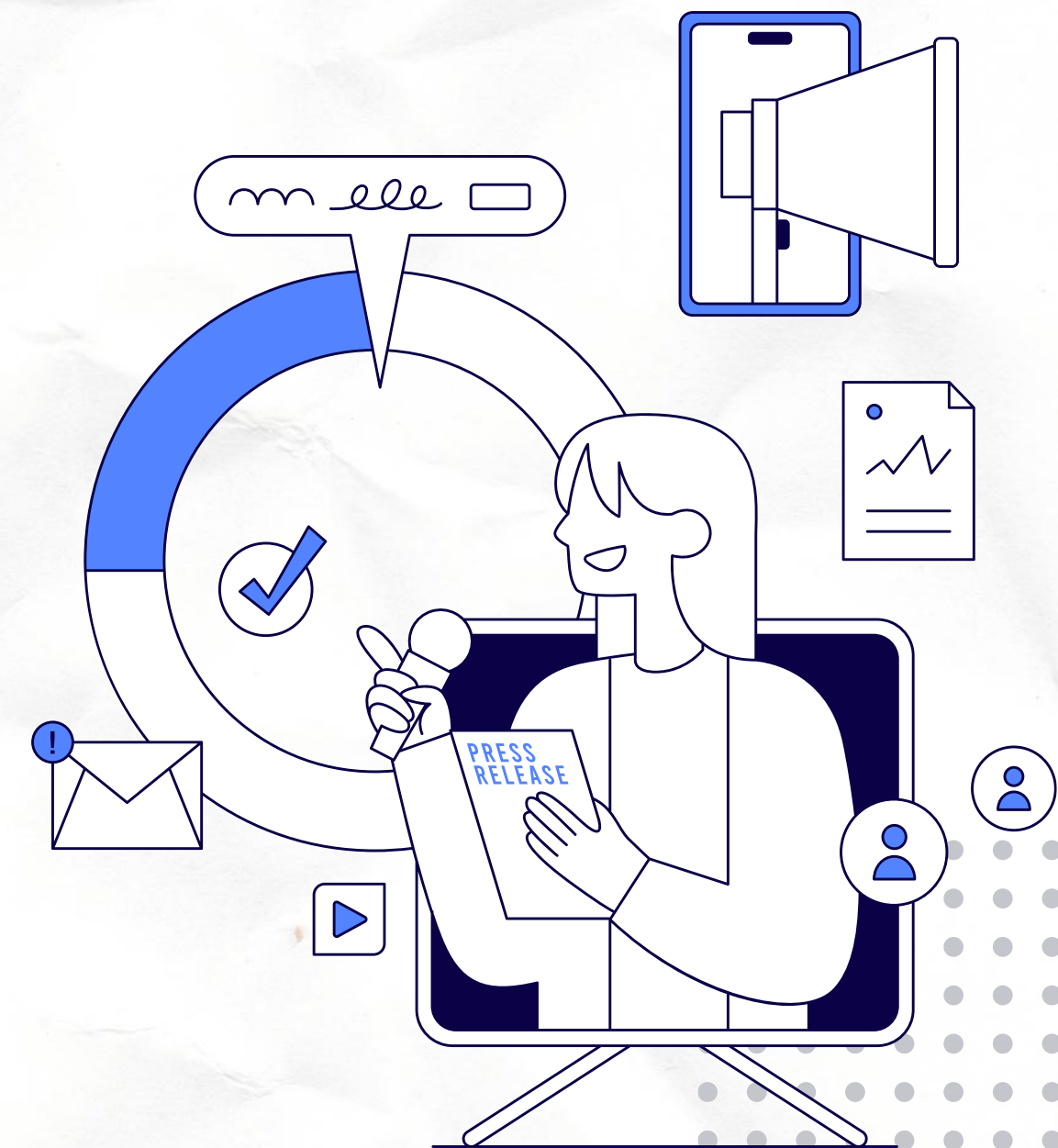


INSIGHT & RECOMMENDATION PROBLEM

-1

Terdapat perbedaan yang sangat signifikan antara produk pertama, IDROID_BALRX7-Gold dengan total penjualan 1000 unit, dibandingkan dengan produk lain (dalam top 5) yang penjualannya lebih rendah. Hal tersebut menunjukkan bahwa IDROID_BALRX7-Gold merupakan produk yang paling populer dan diminati dalam kategori Mobiles & Tablets selama tahun 2022 sehingga dapat dijadikan hadiah utama kompetisi festival tersebut

Pastikan tim pengadaan memeriksa stok yang tersedia dari 5 produk Mobiles & Tablets teratas untuk menjamin bahwa produk tersebut dapat didistribusikan untuk hadiah kompetisi Festival Akhir Tahun tanpa ada kendala.



► PROBLEM - 2

Dear Data Analyst,

Menindaklanjuti meeting gabungan Tim Warehouse dan Tim Marketing, kami menemukan bahwa ketersediaan stock produk dengan Kategori Others pada akhir 2022 kemarin masih banyak.

1. Kami mohon bantuan untuk melakukan pengecekan data penjualan kategori tersebut dengan tahun 2021 secara kuantitas penjualan. Dugaan sementara kami, telah terjadi penurunan kuantitas penjualan pada 2022 dibandingkan 2021. (Mohon juga menampilkan data ke-15 kategori)
2. Apabila memang terjadi penurunan kuantitas penjualan pada kategori Others, kami mohon bantuan untuk menyediakan data TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 jika dibanding dengan 2021. Hal ini kami gunakan sebagai bahan diskusi pada meeting selanjutnya.

Mohon bantuan untuk mengirimkan data tersebut paling lambat 4 hari dari hari ini. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards

Tim Warehouse



CODE PROBLEM - 2.1

Problem Approach

Membandingkan total kuantitas penjualan untuk masing-masing kategori produk pada tahun 2021 dan 2022. Data dikelompokkan berdasarkan kategori, dihitung total kuantitasnya, kemudian hasilnya dibandingkan untuk melihat pertumbuhan atau penurunan.

Table

```
# Membuat perbandingan data penjualan produk berdasarkan kategori pada tahun 2021 dan 2022
# tampilkan data ke-15 kategori
# Filter data penjualan tahun 2021 dan menghitung total qty_ordered per kategori
sales_2021_category = pd.DataFrame(
    df[
        (df['order_date'].dt.year == 2021) & # Filter data transaksi tahun 2021
        (df['is_valid'] == 1) # Memfilter data dengan is_valid=1
    ]
    .groupby('category')['qty_ordered'] # Mengelompokkan data berdasarkan kategori
    .sum() # Menjumlahkan total qty_ordered untuk setiap kategori
    .sort_values(ascending=False) # Mengurutkan hasil dari terbesar ke terkecil
    .reset_index(name='Total_Quantity_2021') # Mengatur ulang indeks dan mengganti nama kolom hasil agregasi
)

# Filter data penjualan tahun 2022 dan menghitung total qty_ordered per kategori
sales_2022_category = pd.DataFrame(
    df[
        (df['order_date'].dt.year == 2022) & # Filter data transaksi tahun 2022
        (df['is_valid'] == 1) # Memfilter data dengan is_valid=1
    ]
    .groupby('category')['qty_ordered'] # Mengelompokkan data berdasarkan kategori
    .sum() # Menjumlahkan total qty_ordered untuk setiap kategori
    .sort_values(ascending=False) # Mengurutkan hasil dari terbesar ke terkecil
    .reset_index(name='Total_Quantity_2022') # Mengatur ulang indeks dan mengganti nama kolom hasil agregasi
)

# Menggabungkan data penjualan tahun 2021 dan 2022 berdasarkan kategori
sales_category_growth = sales_2021_category.merge(sales_2022_category, on='category', how='inner')

# Menghitung pertumbuhan (growth) dari 2021 ke 2022
sales_category_growth['Quantity_Growth'] = sales_category_growth['Total_Quantity_2022'] - sales_category_growth['Total_Quantity_2021']

# Mengurutkan data berdasarkan pertumbuhan (growth)
sales_category_growth.sort_values(by='Quantity_Growth', ascending=True, inplace=True)

# Menampilkan 15 kategori teratas dengan pertumbuhan
sales_category_growth.head(15)
```

Syntax Breakdown

- Filter Data Kategori Tahun 2021 dan 2022
 - `df['order_date'].dt.year == 2021/2022`: Memfilter data berdasarkan tahun transaksi.
 - `df['is_valid'] == 1`: Menyertakan hanya transaksi yang valid.
- Groupby dan Aggregation
 - `.groupby('category')`: Mengelompokkan data berdasarkan kategori produk.
 - `.sum()`: Menjumlahkan total kuantitas (`qty_ordered`) untuk setiap kategori.
- Reset Index
 - `.reset_index(name='Total_Quantity_2021/2022')`: Mengatur ulang indeks dan memberi nama baru pada kolom hasil agregasi.
- Merge dan Decline Calculation
 - `merge(sales_2021_category, sales_2022_category, on='category', how='inner')`: Menggabungkan data tahun 2021 dan 2022 berdasarkan kategori.
- Calculate Growth dan Sort Values
 - `sales_growth['Quantity_Growth']`: Menghitung selisih kuantitas penjualan antara 2021 dan 2022.
 - `.sort_values('Quantity_Growth', ascending=True)`: Mengurutkan data berdasarkan pertumbuhan kuantitas, dari yang paling rendah ke tinggi.



CODE PROBLEM - 2.1

Problem Approach

Visualisasi menggunakan horizontal bar chart untuk mempermudah identifikasi kategori dengan perubahan signifikan. Visualisasi dilakukan dengan menampilkan data secara terurut dari penurunan terbesar ke pertumbuhan terbesar, serta menambahkan garis vertikal pada titik nol untuk membedakan tren positif dan negatif.

Visualization

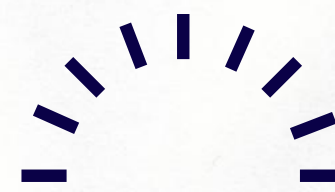
```
[ ] # Create horizontal barplot
plt.figure(figsize=(12, 8))
ax = sns.barplot(data=sales_category_growth, x='Quantity_Growth', y='category', palette='Blues_r')

# Add text labels for each bar
for i, (value, name) in enumerate(zip(sales_category_growth['Quantity_Growth'], sales_category_growth['category'])):
    ax.text(value + 5, i, f'{value}', color='black', va='center', fontsize=10)

# Add titles and labels
plt.title('Growth in Quantity by Category (2022 vs 2021)', fontsize=14)
plt.xlabel('Quantity Growth', fontsize=12)
plt.ylabel('Product Category', fontsize=12)
plt.axvline(0, color='gray', linestyle='--', linewidth=1) # Add a vertical line at 0 for reference
plt.tight_layout()
plt.show()
```

Syntax Breakdown

- `sns.barplot(data=sales_growth.head(15), x='Quantity_Growth', y='category')`: Membuat diagram batang horizontal untuk menampilkan perubahan kuantitas.
- `ax.text(value + 5, i, f'{value}')`: Menambahkan label angka pada setiap batang diagram.
- `plt.axvline(0, color='gray', linestyle='--')`: Menambahkan garis vertikal di titik nol.
- `palette='Blues_r'`: Mengatur warna diagram agar estetik dan konsisten.





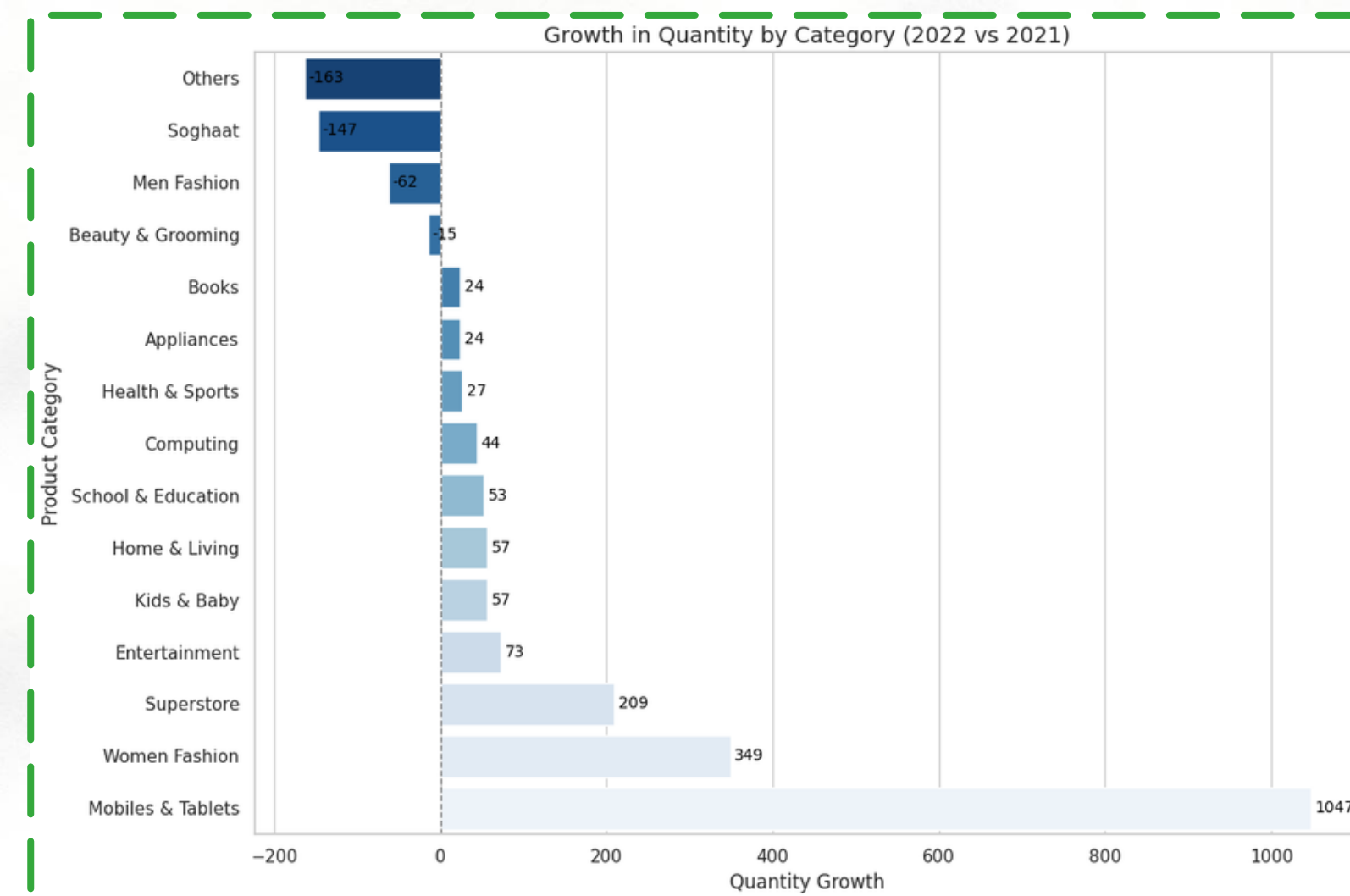
RESULT PROBLEM - 2.2

Output

	category	Total_Quantity_2021	Total_Quantity_2022	Quantity_Growth
1	Others	426	263	-163
0	Soghaat	759	612	-147
3	Men Fashion	237	175	-62
9	Beauty & Grooming	168	153	-15
7	Books	171	195	24
11	Appliances	124	148	24
6	Health & Sports	173	200	27
12	Computing	109	153	44
5	School & Education	184	237	53
4	Home & Living	193	250	57
8	Kids & Baby	170	227	57
14	Entertainment	77	150	73
2	Superstore	327	536	209
10	Women Fashion	140	489	349
13	Mobiles & Tablets	107	1154	1047

Insight

- Kuantitas penjualan kategori seperti Mobiles & Tablets, Women Fashion, dan Superstore mengalami peningkatan yang cukup signifikan pada tahun 2022.
- Kategori Others mengalami penurunan kuantitas terbesar, yaitu sebesar 163 unit dari tahun 2021 ke 2022.
- Mayoritas kategori menunjukkan pertumbuhan positif, namun beberapa kategori penting seperti Others mengalami penurunan yang cukup besar, mengindikasikan perlunya perhatian khusus.



Recommendation

- Analisis lebih mendalam terkait penyebab penurunan, seperti kurangnya promosi, tren pasar yang berubah, atau stok yang tidak sesuai dengan permintaan.
- Pertimbangkan strategi promosi khusus untuk meningkatkan penjualan pada kategori yang mengalami penurunan kuantitas penjualan.
- Memperkuat strategi penjualan untuk kategori yang sedang tumbuh pesat seperti Mobiles & Tablets dan Women Fashion.
- Melakukan survei untuk memahami perubahan kebutuhan atau preferensi konsumen yang memengaruhi kinerja kategori.



CODE PROBLEM - 2.2

Problem Approach

Memfilter data transaksi dari kategori Others, menghitung total kuantitas penjualan per produk untuk tahun 2021 dan 2022, lalu membandingkan selisihnya. Produk dengan penurunan kuantitas tertinggi diurutkan dan disajikan dalam daftar 20 teratas.

Table

```
[21] # Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Menyediakan data TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 jika dibanding dengan 2021
sales_2021_others = pd.DataFrame(
    df[
        (df['order_date'].dt.year == 2021) & # Filter transaksi tahun 2021
        (df['category'] == 'Others') &      # Memfilter data category = Others
        (df['is_valid'] == 1)                # Memfilter data is_valid=1
    ]
    .groupby('sku_name')['qty_ordered']
    .sum() # menjumlah total qty_ordered untuk setiap kategori
    .sort_values(ascending=False) #mengurutkan hasil dari terbesar ke terkecil
    .reset_index(name='Qty_2021') # mengatur ulang indeks dan mengganti nama kolom hasil agregasi
)

sales_2022_others = pd.DataFrame(
    df[
        (df['order_date'].dt.year == 2022) & # Filter transaksi tahun 2022
        (df['category'] == 'Others') &      # Memfilter data category = Others
        (df['is_valid'] == 1)                # Memfilter data is_valid=1
    ]
    .groupby('sku_name')['qty_ordered']
    .sum() # menjumlah total qty_ordered setiap kategori
    .sort_values(ascending=False) #mengurutkan hasil dari terbesar ke terkecil
    .reset_index(name='Qty_2022') # mengatur ulang indeks dan mengganti nama kolom hasil agregasi
)

# Gabungkan data penjualan tahun 2021 dan 2022
sales_comparison = pd.merge(sales_2021_others, sales_2022_others, on='sku_name', how='outer')
sales_comparison.fillna(0, inplace=True)

# Hitung penurunan kuantitas
sales_comparison['Qty_Decline'] = sales_comparison['Qty_2022'] - sales_comparison['Qty_2021']

# Urutkan berdasarkan penurunan kuantitas terbanyak
sales_comparison_sorted = sales_comparison.sort_values(by='Qty_Decline', ascending=True)

# Ambil 20 produk teratas yang mengalami penurunan kuantitas terbanyak
top_20_decline = sales_comparison_sorted.head(20)

# Menampilkan hasil
top_20_decline
```

Syntax Breakdown

- Filter Data Kategori Others
 - `df['category'] == 'Others'`: Memfilter hanya transaksi dengan kategori Others.
 - `df['is_valid'] == 1`: Menyertakan hanya transaksi yang valid.
- Groupby dan Aggregation
 - `.groupby('sku_name')`: Mengelompokkan data berdasarkan nama produk.
 - `.sum()`: Menghitung total kuantitas yang terjual untuk setiap produk.
- Merge dan Decline Calculation
 - `merge(...)`: Menggabungkan data 2021 dan 2022 berdasarkan nama produk.
 - `fillna(0)`: Mengisi nilai kosong dengan nol untuk menangani produk tanpa penjualan di salah satu tahun.
 - `sales_comparison['Qty_Decline']`: Menghitung selisih kuantitas antara 2021 dan 2022.
- Sorting dan Filtering
 - `sort_values(by='Qty_Decline')`: Mengurutkan data berdasarkan penurunan terbesar ke terkecil.
 - `head(20)`: Memilih 20 produk dengan penurunan tertinggi.



CODE PROBLEM - 2.2

Problem Approach

Visualisasi dilakukan dengan menggunakan horizontal bar chart untuk mempermudah membandingkan penurunan penjualan produk berdasarkan SKU, mengutamakan visualisasi produk dengan perubahan signifikan.

Visualization

```
# Membuat bar chart horizontal
plt.figure(figsize=(12, 8)) # Mengatur ukuran figure
sns.barplot(
    data=top_20_decline, # Data untuk bar chart
    y='sku_name',        # Nama produk sebagai sumbu-y
    x='Qty_Decline',     # Penurunan kuantitas sebagai sumbu-x
    palette='coolwarm'   # Palet warna untuk estetika
)

# Menambahkan label nilai pada setiap bar
for index, value in enumerate(top_20_decline['Qty_Decline']):
    plt.text(value, index, f'{value:.0f}', va='center', ha='right' if value < 0 else 'left', fontsize=10)

# Menambahkan judul dan label sumbu
plt.title('Top 20 Products with the Highest Decline in Quantity (2022 vs 2021)', fontsize=16, fontweight='bold')
plt.xlabel('Quantity Decline', fontsize=12)
plt.ylabel('Product SKU Name', fontsize=12)

# Menambahkan grid untuk tampilan yang lebih rapi
plt.grid(axis='x', linestyle='--', alpha=0.7)

# Menampilkan plot
plt.tight_layout()
plt.show()
```

Syntax Breakdown

- Barplot
 - `sns.barplot(data=top_20_decline, x='Qty_Decline', y='sku_name')`: Menampilkan produk dengan penurunan kuantitas tertinggi.
- Annotation
 - `plt.text(value, index, f'{value:.0f}')`: Menambahkan label angka pada setiap batang diagram.
- Customization
 - `plt.grid(axis='x', linestyle='--')`: Menambahkan grid di sumbu x untuk mempermudah pembacaan visual.





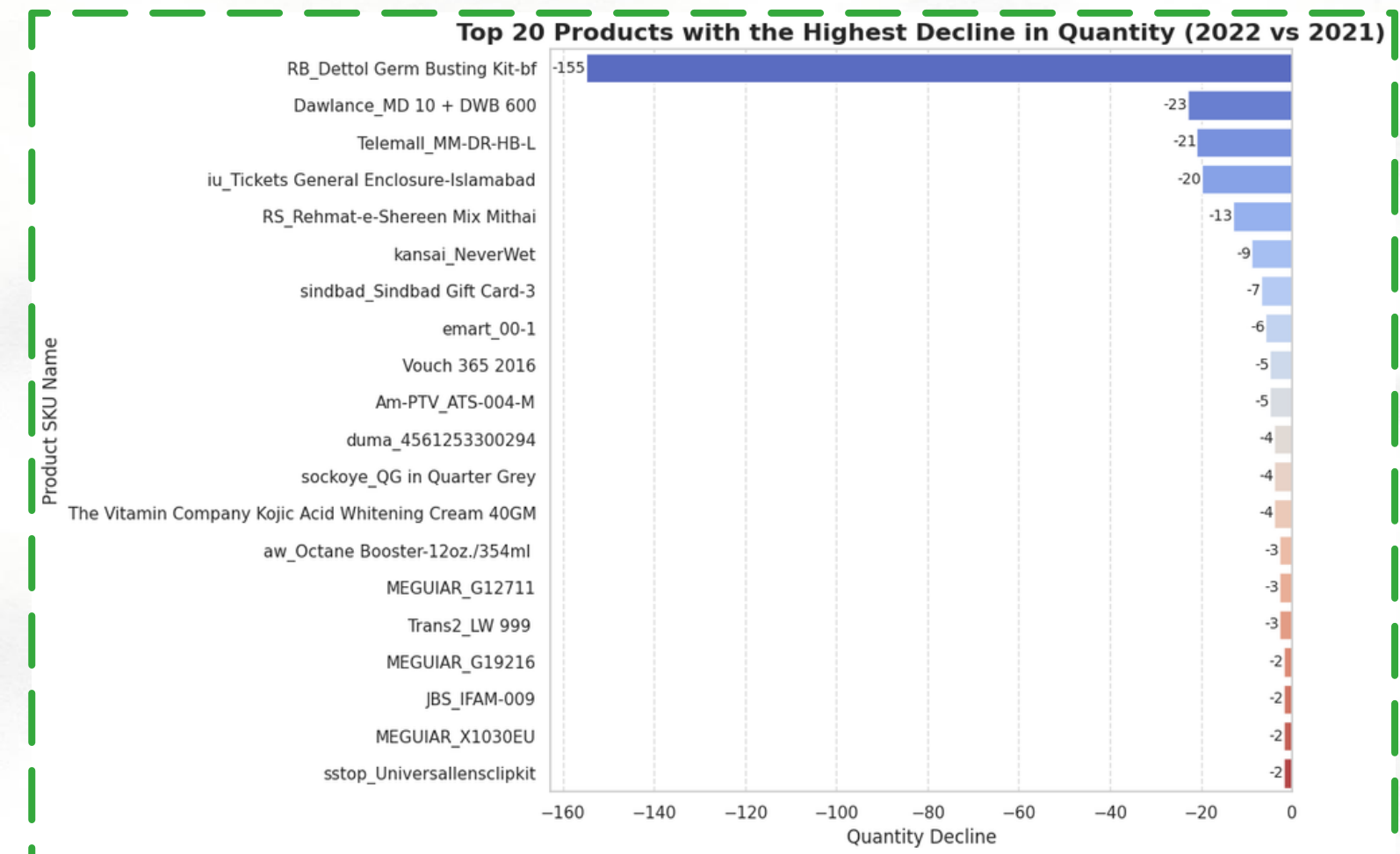
RESULT PROBLEM - 2.2

Output

	sku_name	Qty_2021	Qty_2022	Qty_Decline
69	RB_Dettol Germ Busting Kit-bf	200.0	45.0	-155.0
18	Dawlance_MD 10 + DWB 600	23.0	0.0	-23.0
84	Telemall_MM-DR-HB-L	23.0	2.0	-21.0
137	iu_Tickets General Enclosure-Islamabad	20.0	0.0	-20.0
70	RS_Rehmat-e-Shereen Mix Mithai	13.0	0.0	-13.0
141	kansai_NeverWet	10.0	1.0	-9.0
154	sindbad_Sindbad Gift Card-3	7.0	0.0	-7.0
133	emart_00-1	7.0	1.0	-6.0
93	Vouch 365 2016	5.0	0.0	-5.0
4	Am-PTV_ATS-004-M	5.0	0.0	-5.0
129	duma_4561253300294	4.0	0.0	-4.0
155	sockoye_QG in Quarter Grey	4.0	0.0	-4.0
85	The Vitamin Company Kojic Acid Whitening Cream...	4.0	0.0	-4.0
115	aw_Octane Booster-12oz./354ml	3.0	0.0	-3.0
43	MEGUIAR_G12711	4.0	1.0	-3.0
88	Trans2_LW 999	3.0	0.0	-3.0
50	MEGUIAR_G19216	2.0	0.0	-2.0
26	JBS_IFAM-009	2.0	0.0	-2.0
60	MEGUIAR_X1030EU	2.0	0.0	-2.0
156	sstop_Universallensclipkit	2.0	0.0	-2.0

Insight

Hasil output menunjukkan bahwa produk dengan penurunan jumlah penjualan tertinggi antara tahun 2021 dan 2022 adalah produk RB_Dettol Germ Busting Kit-bf, dengan penurunan signifikan sebesar 155 unit. penurunan ini jauh lebih besar dibandingkan dengan produk lainnya yang turun berkisar antara 2 hingga 23 unit. Selain itu, beberapa produk mengalami penurunan ke angka nol pada tahun 2022 atau tidak ada penjualan sama sekali.



Recommendation

- Evaluasi faktor penyebab penurunan penjualan produk signifikan
- Melakukan kampanye promosi untuk produk dengan penurunan moderat hingga tinggi untuk meningkatkan daya tarik pelanggan.
- Revitalisasi strategi pemasaran untuk produk yang tidak terjual sama sekali pada tahun 2022
- Melakukan riset pasar atau survei pelanggan untuk memahami alasan di balik rendahnya penjualan produk-produk ini. Identifikasi apakah ada produk substitusi atau perubahan preferensi pelanggan yang menjadi penyebab.



PROBLEM - 3



Dear Data Analyst,

Terkait ulang tahun perusahaan pada 2 bulan mendatang, Tim Digital Marketing akan memberikan informasi promo bagi pelanggan pada akhir bulan ini. Kriteria pelanggan yang akan kami butuhkan adalah mereka yang sudah melakukan check-out namun belum melakukan pembayaran (`is_gross = 1`) selama tahun 2022. Data yang kami butuhkan adalah ID Customer dan Registered Date.

Mohon bantuan, untuk mengirimkan data tersebut sebelum akhir bulan ini ke Tim Digital Marketing. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards

Tim Digital Marketing



CODE PROBLEM - 3

```
✓ [20] import pandas as pd
0s

# Memfilter data berdasarkan kriteria yang diberikan
filtered_data = df[
    (df['is_gross'] == 1) & # gross = 1
    (df['is_valid'] == 0) & # valid = 0
    (df['is_net'] == 0) &   # net = 0
    (df['order_date'].dt.year == 2022) # Transaksi selama tahun 2022
]

# Mengambil kolom ID Customer dan Registered Date
result = filtered_data[['customer_id', 'registered_date']].drop_duplicates()

# Menampilkan hasil
print(result)
```

- `df['is_gross'] == 1` : Memilih baris di mana kolom `is_gross` memiliki nilai 1
- `df['is_valid'] == 0` : Memilih baris di mana kolom `is_valid` memiliki nilai 0
- `df['is_net'] == 0` : Memilih baris di mana kolom `is_net` memiliki nilai 0
- `df['order_date'].dt.year == 2022`: Memilih transaksi yang terjadi pada tahun 2022. `df['order_date']` diasumsikan berupa objek `datetime`, dan `.dt.year` digunakan untuk mengekstrak tahun dari tanggal tersebut.
- `filtered data` : Mengambil dua kolom dari data yang sudah difilter, yaitu `customer_id` dan `registered_date`.
- `drop duplicates` : Menghapus baris yang duplikat berdasarkan kedua kolom tersebut, sehingga hanya satu entri unik per kombinasi `customer_id` dan `registered_date`.





RESULT PROBLEM - 3

```
[↕]
  customer_id registered_date
9      C246762L 2022-05-08
18     C848774L 2021-11-07
19     C693415L 2022-04-12
21     C180595L 2022-04-22
22     C587425L 2022-03-22
...      ...      ...
5855    C653797L 2022-04-03
5856    C394076L 2021-10-12
5859    C248585L 2022-07-10
5865    C471304L 2022-05-13
5881    C265450L 2022-02-17
```

[820 rows x 2 columns]

Hasil query ini menunjukkan data pelanggan yang terdaftar dengan dua kolom informasi: customer_id yang merupakan ID unik setiap pelanggan, dan registered_date yang menunjukkan tanggal mereka terdaftar. Dalam data yang ditampilkan, terdapat 820 baris, dan beberapa contoh data yang terambil mencakup ID pelanggan seperti C246762L yang terdaftar pada 2022-05-08, serta C848774L yang terdaftar pada 2021-11-07. Secara keseluruhan, data ini memberikan gambaran mengenai waktu pendaftaran pelanggan selama periode tertentu.



INSIGHT & RECOMMENDATION PROBLEM

- 3

1. Sebanyak 820 pelanggan memenuhi kriteria yang diminta, yaitu telah melakukan check-out tetapi belum melakukan pembayaran (`is_gross = 1`) selama tahun 2022. Ini menunjukkan bahwa ada potensi besar bagi tim marketing untuk mengonversi pelanggan-pelanggan ini menjadi pembeli dengan pendekatan yang tepat.
2. Pelanggan yang sudah berada pada tahap check-out tetapi belum membayar dapat dikategorikan sebagai pelanggan yang memiliki ketertarikan terhadap produk, namun belum menyelesaikan transaksi. Ini menjadi peluang besar bagi Tim Digital Marketing untuk memberikan insentif seperti diskon atau reminder untuk mendorong mereka menyelesaikan pembayaran.
3. Kolom `registered_date` memberi wawasan tentang kapan pelanggan pertama kali mendaftar, yang bisa digunakan untuk menganalisis apakah waktu pendaftaran mempengaruhi kemungkinan mereka untuk melakukan pembelian atau tidak.
4. Meskipun ada 820 pelanggan, perlu diingat bahwa data ini hanya mencakup pelanggan yang berhubungan dengan transaksi tahun 2022. Tidak ada informasi terkait dengan transaksi atau pelanggan yang mungkin memiliki aktivitas setelah tahun tersebut, sehingga perlu dipertimbangkan untuk mengevaluasi data lebih lanjut jika ingin memperluas jangkauan promosi.
5. Dengan data `registered_date`, ada peluang untuk mengelompokkan pelanggan berdasarkan periode waktu pendaftaran mereka, misalnya pelanggan yang mendaftar lebih awal atau lebih akhir di 2022. Segmentasi ini dapat membantu Tim Digital Marketing merancang pesan yang lebih terpersonalisasi, seperti penawaran yang berbeda untuk pelanggan yang baru mendaftar versus pelanggan yang sudah lebih lama bergabung.



PROBLEM - 4

Dear Data Analyst,

Pada bulan October hingga Desember 2022, kami melakukan campaign setiap hari Sabtu dan Minggu. Kami hendak menilai, apakah campaign tersebut cukup berdampak pada kenaikan penjualan (before_discount). Mohon bantuan untuk menampilkan data:

1. Rata-rata harian penjualan weekends (Sabtu dan Minggu) vs rata-rata harian penjualan weekdays (Senin-Jumat) per bulan tersebut. Apakah ada peningkatan penjualan pada masing-masing bulan tersebut.
2. Rata-rata harian penjualan weekends (Sabtu dan Minggu) vs rata-rata harian penjualan weekdays (Senin-Jumat) keseluruhan 3 bulan tersebut.

Mohon bantuan untuk mengirimkan data tersebut paling lambat minggu depan. Atas bantuan yang diberikan, kami mengucapkan terima kasih.

Regards

Tim Campaign



CODE PROBLEM - 4.1

Problem Approach

membandingkan rata-rata harian penjualan antara weekends dan weekdays di bulan Oktober, November, dan Desember 2022, pertama-tama data akan difilter untuk memastikan hanya transaksi valid di tahun 2022 dan bulan yang relevan. Kemudian, kita akan menghitung rata-rata penjualan untuk weekdays (Senin-Jumat) dan weekends (Sabtu dan Minggu) per bulan. Perbandingan antara rata-rata penjualan weekends dan weekdays akan dihitung untuk melihat apakah ada peningkatan pada weekends.

Table

```
# Tambahkan kolom day, month, dan month_number
df['day'] = df['order_date'].dt.day_name() # Menambahkan nama hari
df['month'] = df['order_date'].dt.month_name() # Nama bulan
df['month_number'] = df['order_date'].dt.month # Angka bulan

# Filter data berdasarkan kriteria
filtered_data = df[
    (df['is_valid'] == 1) & # Hanya valid data
    (df['order_date'].dt.year == 2022) & # Tahun 2022
    (df['order_date'].dt.month.isin([10, 11, 12])) # Oktober-Desember
]

# Menghitung rata-rata harian 'weekdays'
weekdays_avg = (
    filtered_data[filtered_data['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])]
    .groupby('month_number')['before_discount'] # Mengelompokkan berdasarkan bulan
    .mean() # Menghitung rata-rata
    .round(2) # Pembulatan ke 2 angka desimal
    .reset_index(name='avg_sales_weekdays') # mengatur ulang indeks dan mengganti nama kolom hasil ag
)

# Menghitung rata-rata harian 'weekends'
weekends_avg = (
    filtered_data[filtered_data['day'].isin(['Saturday', 'Sunday'])] # Filter weekends
    .groupby('month_number')['before_discount'] # Mengelompokkan berdasarkan bulan
    .mean() # Menghitung rata-rata
    .round(2) # Pembulatan ke 2 angka desimal
    .reset_index(name='avg_sales_weekends') # mengatur ulang indeks dan mengganti nama kolom hasil agr
)
```

Syntax Breakdown

- Tambahkan Kolom Hari, Bulan, dan Nomor Bulan
 - `df['day'] = df['order_date'].dt.day_name()`: Menambahkan kolom yang berisi nama hari berdasarkan kolom `order_date`.
 - `df['month'] = df['order_date'].dt.month_name()`: Menambahkan kolom yang berisi nama bulan.
 - `df['month_number'] = df['order_date'].dt.month`: Menambahkan kolom yang berisi nomor bulan untuk mempermudah analisis per bulan.
- Filter Data berdasarkan Kriteria
 - `(df['is_valid'] == 1)`: Memilih hanya transaksi yang valid.
 - `(df['order_date'].dt.year == 2022)`: Memilih transaksi hanya untuk tahun 2022.
 - `(df['order_date'].dt.month.isin([10, 11, 12]))`: Memilih transaksi hanya untuk bulan Oktober hingga Desember 2022.
- Hitung Rata-Rata Penjualan Weekdays
 - `filtered_data[filtered_data['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])]`: Memfilter data hanya untuk weekdays.
 - `.groupby('month_number')['before_discount'].mean()`: Mengelompokkan data berdasarkan bulan dan menghitung rata-rata penjualan untuk weekdays.
- Hitung Rata-Rata Penjualan Weekends
 - `filtered_data[filtered_data['day'].isin(['Saturday', 'Sunday'])]`: Memfilter data hanya untuk weekends.
 - `.groupby('month_number')['before_discount'].mean()`: Mengelompokkan data berdasarkan bulan dan menghitung rata-rata penjualan untuk weekends.



CODE PROBLEM - 4.1

Table

```
# Merger hasil agregasi weekdays dan weekends berdasarkan 'mont_number'
monthly_avg = pd.merge weekdays_avg, weekends_avg, on='month_number')
monthly_avg['month'] = monthly_avg['month_number'].map({10: 'October', 11: 'November', 12: 'December'})
monthly_avg

# Tambahkan kolom perbedaan dan persentase perubahan
monthly_avg['difference'] = (monthly_avg['avg_sales_weekends'] - monthly_avg['avg_sales_weekdays']).round(2)
monthly_avg['%diff'] = (monthly_avg['difference'] / monthly_avg['avg_sales_weekdays'] * 100).round(2) # Pers

# Tampilkan Tabel
print(monthly_avg[['month', 'avg_sales_weekdays', 'avg_sales_weekends', 'difference', '%diff']])
```

Syntax Breakdown

- Gabungkan Data Weekdays dan Weekends
 - `pd.merge weekdays_avg, weekends_avg, on='month_number')`: Menggabungkan hasil rata-rata weekdays dan weekends berdasarkan nomor bulan.
- Hitung Perbedaan dan Persentase Perubahan
 - `monthly_avg['difference'] = (monthly_avg['avg_sales_weekends'] - monthly_avg['avg_sales_weekdays'])`: Menghitung selisih antara rata-rata penjualan weekends dan weekdays.
 - `monthly_avg['%diff'] = (monthly_avg['difference'] / monthly_avg['avg_sales_weekdays'] * 100)`: Menghitung persentase perubahan.



CODE PROBLEM - 4.1

Problem Approach

Visualisasi akan digunakan untuk membandingkan rata-rata penjualan harian antara weekdays dan weekends di bulan Oktober hingga Desember 2022. Grafik batang (bar plot) akan menampilkan perbandingan ini untuk setiap bulan dengan warna yang berbeda untuk weekdays dan weekends, untuk mempermudah analisis perbandingan.

Visualization

```
# Visualisasi Grafik
plt.figure(figsize=(8, 6))
sns.barplot(data=monthly_avg.melt(id_vars='month',
                                   value_vars=['avg_sales_weekdays', 'avg_sales_weekends'],
                                   var_name='Day Type',
                                   value_name='Average Sales'),
            x='month', y='Average Sales', hue='Day Type', palette = 'viridis') # Grafik batang
plt.title("Average Sales 2021 Vs 2022")
plt.ylabel("Average Sales")
plt.xlabel("Month")
plt.legend(title="Day Type", loc="upper right")
plt.show()
```

Syntax Breakdown

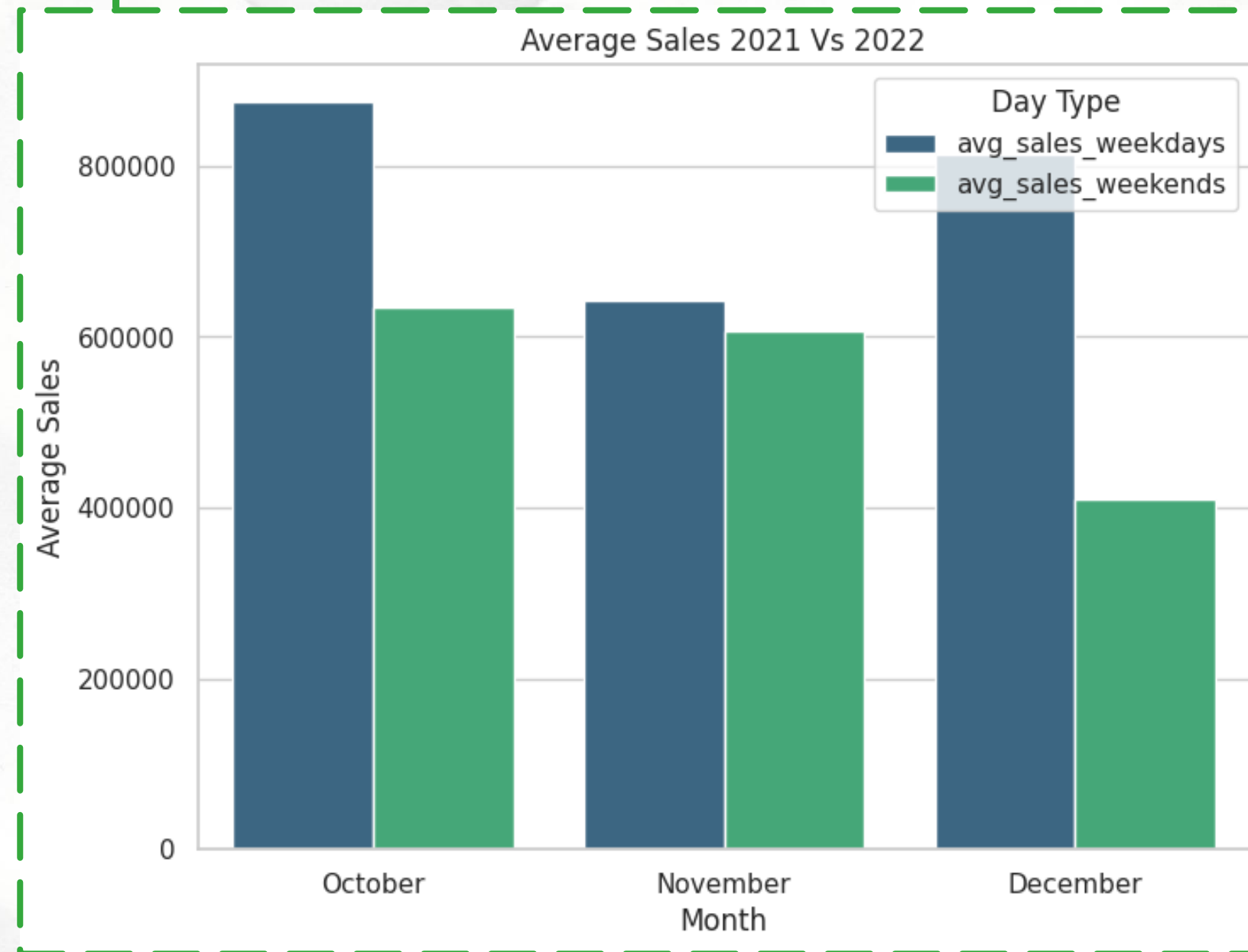
- Mengatur Ukuran Grafik
 - `plt.figure(figsize=(8, 6))`: Mengatur ukuran grafik menjadi 8x6 inci.
- Buat Bar Plot
 - `sns.barplot()`: Membuat grafik batang untuk membandingkan rata-rata penjualan weekdays dan weekends per bulan.
 - `data=monthly_avg.melt(...)`: Mengubah format data agar dapat digunakan dengan `sns.barplot()` (melting data).
- Tambahkan Judul dan Label
 - `plt.title()`: Menambahkan judul pada grafik.
 - `plt.ylabel()`, `plt.xlabel()`: Menambahkan label pada sumbu Y dan X.
- Menampilkan Grafik
 - `plt.show()`: Menampilkan grafik yang telah dibuat.





RESULT PROBLEM - 4.1

Output



	month	avg_sales_weekdays	avg_sales_weekends	difference	%diff
0	October	874690.27	634260.07	-240430.20	-27.49
1	November	641862.00	607794.21	-34067.79	-5.31
2	December	813574.29	410599.40	-402974.89	-49.53

Insight

Penjelasan Berdasarkan Bulan:

- Oktober: Penjualan pada hari kerja (weekdays) jauh lebih tinggi dibandingkan akhir pekan (weekends), dengan selisih 240,430.20 atau -27.49% lebih rendah pada weekends.
- November: Perbedaan antara weekdays dan weekends jauh lebih kecil. Rata-rata penjualan pada weekdays sedikit lebih tinggi daripada weekends, dengan selisih 34,067.79 atau -5.31%.
- Desember: Desember menunjukkan perbedaan terbesar. Penjualan weekdays jauh lebih tinggi daripada weekends dengan selisih 402,974.89 atau -49.53%. Penurunan penjualan pada weekends di bulan ini sangat signifikan, sekitar 49.53% lebih rendah dibandingkan hari kerja.

Kesimpulan:

- Secara umum, penjualan pada hari kerja (weekdays) lebih tinggi dibandingkan dengan akhir pekan (weekends) di ketiga bulan tersebut, dengan Oktober dan Desember menunjukkan perbedaan yang lebih besar.
- Hal ini dapat mengindikasikan bahwa konsumen lebih cenderung berbelanja pada hari kerja daripada pada akhir pekan, atau mungkin dipengaruhi oleh faktor lain seperti promosi atau jenis produk yang lebih sering dibeli di hari kerja.



IRECOMMENDATION PROBLEM - 4.1



Recommendation

- Fokus pada Pemasaran Weekends:
 - Perlu adanya peningkatan pemasaran yang lebih fokus pada akhir pekan, dengan menawarkan penawaran atau promo khusus untuk menarik konsumen berbelanja.
 - Bisa menciptakan “Weekend Specials” atau diskon khusus yang berlaku hanya di akhir pekan untuk mengurangi perbedaan yang signifikan antara weekdays dan weekends.
- Kampanye Liburan atau Musiman di Desember:
 - Meningkatkan pemasaran khusus untuk bulan Desember dengan fokus pada strategi belanja musiman. Misalnya, mengadakan kampanye dengan tema liburan atau menawarkan paket bundling yang menarik pada akhir pekan.
- Analisis Lebih Lanjut terhadap Perilaku Konsumen:
 - Melakukan survei atau riset pasar lebih mendalam untuk memahami perubahan dalam perilaku konsumen, apakah mereka lebih suka berbelanja di weekdays karena kenyamanan atau kemudahan tertentu, atau apakah ada faktor lain yang mempengaruhi preferensi mereka.
- Peningkatan Pengalaman Berbelanja di Weekdays:
 - Mengingat penjualan yang lebih tinggi pada weekdays, perusahaan dapat terus meningkatkan pengalaman berbelanja selama hari kerja dengan menawarkan pengalaman belanja yang lebih personal atau efisien.





CODE PROBLEM - 4.2

Problem Approach

Untuk membandingkan rata-rata penjualan antara weekdays dan weekends secara keseluruhan dari Oktober hingga Desember 2022, kita akan menghitung rata-rata penjualan untuk seluruh periode tersebut, baik untuk weekdays maupun weekends. Selanjutnya, kita akan membuat tabel perbandingan antara kedua jenis hari dan menghitung selisih serta persentase perubahan antara weekdays dan weekends.

Table

```
# Rata-rata harian penjualan weekends (Sabtu dan Minggu) vs rata-rata harian penjualan weekdays (Senin-Jumat)
# keseluruhan 3 bulan tersebut.
# Tambahkan kolom day, month, dan month_number
# Filter data untuk weekends dan weekdays
weekends = filtered_data[filtered_data['day'].isin(['Saturday', 'Sunday'])] # Filter weekends
weekdays = filtered_data[filtered_data['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])]

# Menghitung rata-rata harian untuk keseluruhan data (weekends & weekdays)
weekends_avg_overall = weekends['before_discount'].mean() # Menghitung rata-rata penjualan 'before_discount'
weekdays_avg_overall = weekdays['before_discount'].mean() # Menghitung rata-rata penjualan 'before_discount'

# Menggabungkan data weekends dan weekdays
overall_avg = pd.DataFrame({
    'Day Type': ['Weekdays', 'Weekends'], # Menambahkan kolom 'Day Type' dengan nilai 'Weekdays' dan 'Weekends'
    'Average Sales': [weekdays_avg_overall, weekends_avg_overall], # Menambahkan kolom 'Average Sales' dengan rata-rata penjualan
    'Month': ['October - December', 'October - December'], # Menambahkan kolom 'Month' dengan label untuk periode
})

# Membuat tabel perbandingan selisih dan persentase perubahan
overall_comparison = {
    "Quarter": "Q4",
    "Average Weekends": weekends_avg_overall.round(2),
    "Average Weekdays": weekdays_avg_overall.round(2),
    "Difference": (weekends_avg_overall - weekdays_avg_overall).round(2),
    "%Diff": ((weekends_avg_overall - weekdays_avg_overall) / weekdays_avg_overall * 100).round(2)
}

# Tampilkan hasil tabel dengan selisih dan persentase
pd.DataFrame([overall_comparison])
```

Syntax Breakdown

- Filter Data Weekends dan Weekdays
 - weekends = filtered_data[filtered_data['day'].isin(['Saturday', 'Sunday'])]: Memfilter data hanya untuk weekends.
 - weekdays = filtered_data[filtered_data['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])]: Memfilter data hanya untuk weekdays.
- Hitung Rata-Rata Penjualan Weekends dan Weekdays
 - weekends_avg_overall = weekends['before_discount'].mean(): Menghitung rata-rata penjualan untuk weekends secara keseluruhan.
 - weekdays_avg_overall = weekdays['before_discount'].mean(): Menghitung rata-rata penjualan untuk weekdays secara keseluruhan.
- Gabungkan Data Weekdays dan Weekends
 - overall_avg = pd.DataFrame(...): Membuat DataFrame baru dengan informasi rata-rata penjualan untuk weekdays dan weekends.
- Tabel Perbandingan Selisih dan Persentase Perubahan
 - overall_comparison = {...}: Membuat dictionary yang berisi perbandingan rata-rata penjualan antara weekdays dan weekends, serta selisih dan persentase perubahan.



CODE PROBLEM - 4.2

Problem Approach

Visualisasi akan membandingkan rata-rata penjualan antara weekdays dan weekends untuk keseluruhan periode Oktober hingga Desember 2022. Grafik batang akan menampilkan kedua kategori tersebut, dan nilai rata-rata akan ditampilkan di atas setiap batang untuk menunjukkan angka dengan jelas.

Visualization

```
# Visualisasi grafik
plt.figure(figsize=(8, 6)) # Membuat ukuran grafik 8x6 inci
sns.barplot(data=overall_avg, x='Day Type', y='Average Sales', palette='viridis') #
plt.title("Average Sales Weekends vs Weekdays (October - December 2022)") # Menamba
plt.ylabel("Average Sales (before_discount)") # Menambahkan label pada sumbu Y
plt.xlabel("Day Type") # Menambahkan label pada sumbu X

# Menambahkan nilai rata-rata pada grafik untuk memperjelas
for i, v in enumerate(overall_avg['Average Sales']): # Melakukan iterasi untuk seti
    plt.text(i, v + 5000, f"{v:,.2f}", ha='center', va='bottom', fontweight='bold')

plt.show()
```

Syntax Breakdown

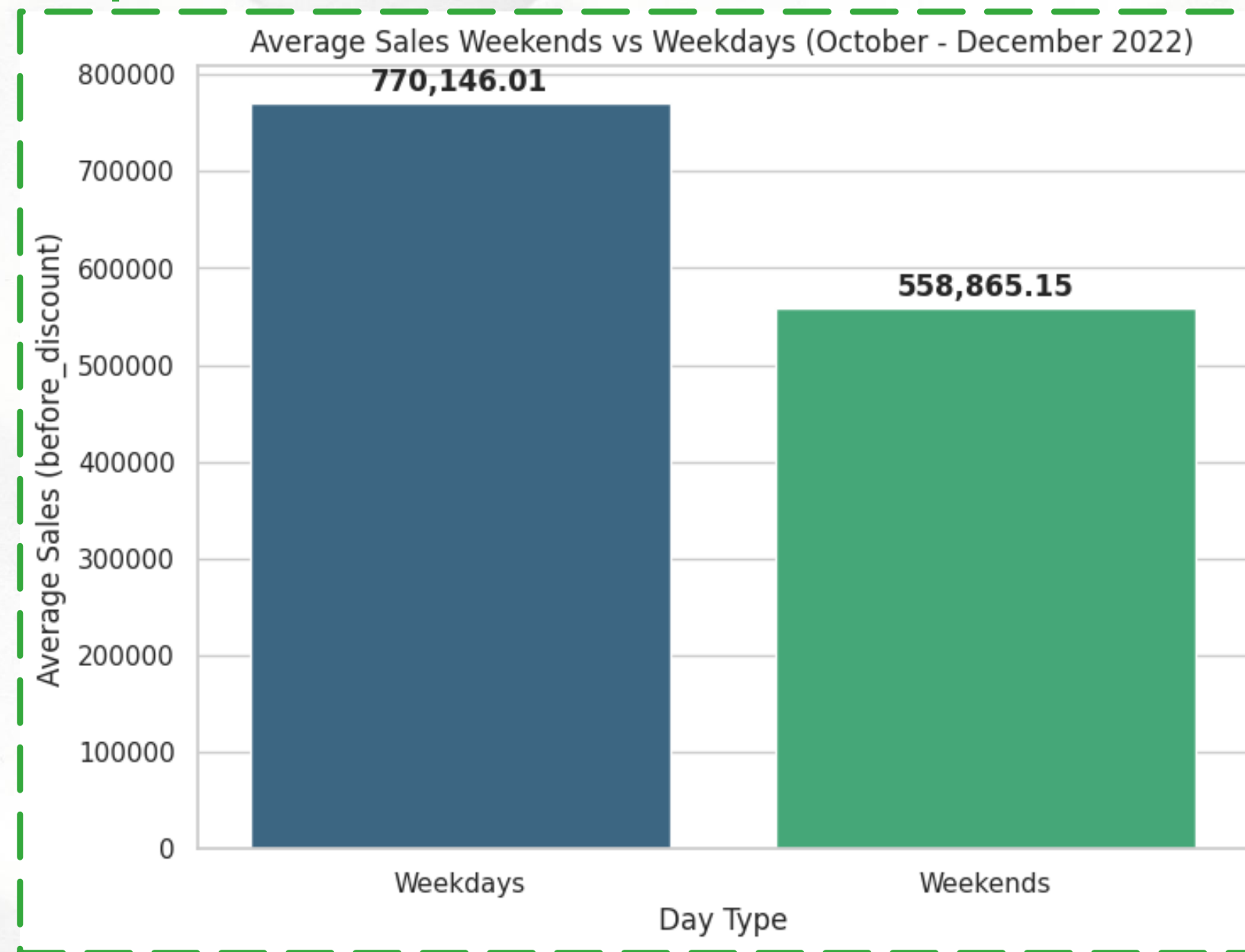
- Mengatur Ukuran Grafik
 - `plt.figure(figsize=(8, 6))`: Mengatur ukuran grafik menjadi 8x6 inci.
- Buat Bar Plot
 - `sns.barplot()`: Membuat grafik batang untuk membandingkan rata-rata penjualan weekdays dan weekends.
- Tambahkan Judul dan Label
 - `plt.text()`: Menambahkan nilai rata-rata pada batang grafik untuk memperjelas perbandingan.
- Menampilkan Grafik
 - `plt.show()`: Menampilkan grafik yang telah dibuat.





RESULT PROBLEM - 4.2

Output



	Quarter	Average Weekends	Average Weekdays	Difference	%Diff
0	Q4	558865.15	770146.01	-211280.86	-27.43

Insight

- Penjualan rata-rata harian pada weekdays lebih tinggi dibandingkan weekends.
- Pada Q4, penjualan pada hari kerja (weekdays) lebih tinggi daripada penjualan akhir pekan (weekends). Selisihnya adalah sekitar 211,280.86, dengan penjualan weekdays lebih tinggi.
- Penurunan penjualan di weekends cukup signifikan, yaitu sekitar 27.43% lebih rendah dibandingkan dengan penjualan di weekdays. Ini berarti bahwa konsumen cenderung lebih banyak berbelanja pada hari kerja daripada pada akhir pekan selama kuartal keempat.
- Campaign yang dilakukan setiap Sabtu dan Minggu (weekends) tidak cukup efektif untuk meningkatkan penjualan hingga menyamai atau mendekati level penjualan pada weekdays.

Recommendation

- Evaluasi Campaign Weekends: Perbaiki strategi promosi dengan penawaran lebih menarik seperti diskon khusus, bundling, atau gratis ongkir hanya di akhir pekan.
- Optimalkan Weekdays: Pertahankan performa tinggi di weekdays dengan promosi tambahan seperti flash sale atau loyalty program.
- Target Produk Tepat: Promosikan produk unggulan weekdays di weekends untuk menarik minat pelanggan.
- Segmentasi Pelanggan: Identifikasi pola belanja untuk menciptakan urgensi pada weekends, seperti promo waktu terbatas atau eksklusif.

CONCLUSIO

N Pada **problem 1**, didapatkan bahwa produk dengan kuantitas penjualan terbanyak adalah produk IDROID_BALRX7-Gold dengan total penjualan sebanyak 1000 unit pada tahun 2022 untuk kategori Mobiles & Tablets

- Pada **problem 2**, ditemukan bahwa kategori Others mengalami penurunan kuantitas terbesar, yaitu sebesar 163 unit dari tahun 2021 ke 2022. Sedangkan produk dengan penurunan jumlah penjualan tertinggi antara tahun 2021 dan 2022 adalah produk RB_Dettol Germ Busting Kit-bf, dengan penurunan signifikan sebesar 155 unit.
- Pada **problem 3**, didapatkan bahwa ada 820 pelanggan memenuhi kriteria yang diminta, yaitu telah melakukan check-out tetapi belum melakukan pembayaran selama tahun 2022.
- Pada **problem 4**, didapatkan bahwa penjualan pada hari kerja (weekdays) lebih tinggi dibandingkan dengan akhir pekan (weekends) di ketiga bulan tersebut (Oktober, November, dan Desember) serta penjualan rata-rata harian pada weekdays lebih tinggi dibandingkan weekends.



THANK YOU



Kelompok B4

