



MYSKILL

FINAL PROJECT

DATA ANALYSIS USING PYTHON IN GOOGLE COLAB

Mochamad Syahrizal

MySkill Final Project Data Analysis Batch 18



TABLE OF CONTENTS



Data Preparation & Preprocessing



Problem Statements & Analysis



Insights & Recommendations





DATA PREPARATION



Import Library

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

Pada awal proyek, kita mengimpor beberapa library penting yang akan digunakan untuk analisis data dan visualisasi:

1. **Pandas (import pandas as pd)**: Digunakan untuk memanipulasi dan menganalisis data dalam bentuk DataFrame.
2. **NumPy (import numpy as np)**: Digunakan untuk perhitungan numerik dan bekerja dengan array.
3. **Matplotlib (import matplotlib.pyplot as plt)**: Library utama untuk membuat visualisasi dasar seperti grafik garis, batang, dll.
4. **Seaborn (import seaborn as sns)**: Library yang mempermudah pembuatan visualisasi statistik yang lebih menarik dan informatif.
5. **BDay dari pandas.tseries.offsets**: Digunakan untuk operasi manipulasi tanggal, khususnya untuk bekerja dengan hari kerja (business days).



DATA PREPARATION



Import Datasets

```
#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)
```

path_od, path_pd, path_cd, path_sd: Menyimpan URL yang mengarah ke dataset CSV yang akan digunakan dalam analisis. Setiap variabel menyimpan data yang berbeda:

- path_od: Menyimpan data transaksi (order detail).
- path_pd: Menyimpan data metode pembayaran (payment detail).
- path_cd: Menyimpan data pelanggan (customer detail).
- path_sd: Menyimpan data produk (SKU detail).

pd.read_csv(): Fungsi dari Pandas yang digunakan untuk membaca file CSV dari URL yang diberikan, kemudian data disimpan dalam DataFrame:

- df_od: Menyimpan data dari order_detail.csv.
- df_pd: Menyimpan data dari payment_detail.csv.
- df_cd: Menyimpan data dari customer_detail.csv.
- df_sd: Menyimpan data dari sku_detail.csv.

Sumber data berasal dari Github, Data yang digunakan adalah data yang berasal dari Tokopedia (bukan data sesungguhnya)

DATASETS DESCRIPTION

| 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 produk 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



Initial Data Inspection

```
#Mengambil 5 baris pertama
df_od.head()

  id customer_id order_date sku_id price qty_ordered before_discount discount_amount after_discount is_gross is_valid is_net payment_id
0 ODR9939707760w C713589L 2021-11-19 P858068 26100 200 5220000.0 2610000.0 2610000.0 1 1 0 5
1 ODR7448356649d C551551L 2021-11-19 P886455 1971942 5 9859710.0 2464927.50 7394782.50 1 0 0 5
2 ODR4011281866z C685596L 2021-11-25 P678648 7482000 1 7482000.0 2065344.62 5416655.38 1 0 0 4
3 ODR3378927994s C830683L 2021-11-22 P540013 3593680 1 3593680.0 1455440.40 2138239.60 1 1 1 5
4 ODR4904430099k C191766L 2021-11-21 P491032 4413220 1 4413220.0 1059172.80 3354047.20 1 1 1 4
```

```
#Mengambil 5 baris pertama
df_sd.head()

      id          sku_name base_price   cogs category
0  P798444        AT-FSM-35  57631.70 46052  Kids & Baby
1  P938347  AYS_Haier-18HNF 3931789.26 3499256    Appliances
2  P826364 Atalian _DV206A-Brown-41  324597.00 243426   Men Fashion
3  P467533  Darul_Sakoon_Food_Bundle  2870.42  2378    Superstore
4  P229955  HP_15AY-15-Ay072NIA-ci3 2265625.00 1631250 Computing
```

KETERANGAN :

Tahap ini bertujuan untuk melihat langsung struktur dan contoh value dalam masing-masing kolom dalam setiap table data agar memberikan gambaran terkait dataset yang akan digunakan



DATA PREPARATION



Initial Data Inspection

```
#Mengambil 5 baris pertama  
df_pd.head()
```

| | <code>id</code> | <code>payment_method</code> |
|---|-----------------|-----------------------------|
| 0 | 1 | cod |
| 1 | 2 | jazzvoucher |
| 2 | 3 | customercredit |
| 3 | 4 | Payaxis |
| 4 | 5 | jazzwallet |

```
#Mengambil 5 baris pertama  
df_cd.head()
```

| | <code>id</code> | <code>registered_date</code> |
|---|-----------------|------------------------------|
| 0 | C996508L | 2021-07-10 |
| 1 | C180415L | 2021-07-18 |
| 2 | C535451L | 2021-07-23 |
| 3 | C177843L | 2021-07-12 |
| 4 | C951682L | 2021-07-27 |

KETERANGAN :

Tahap ini bertujuan untuk melihat langsung struktur dan contoh value dalam masing-masing kolom dalam setiap table agar memberikan gambaran terkait dataset yang akan digunakan



DATA PREPROCESSING

IMPORT SQL TO COLAB

```
#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')
```

KETERANGAN :

1. **from sqlite3 import connect**: Mengimpor fungsi connect dari library SQLite untuk membuat koneksi database di memori (tanpa file fisik).
2. **conn = connect(':memory:')**: Membuat koneksi database sementara di memori (RAM) yang akan digunakan untuk menjalankan SQL.
3. **to_sql()**: Fungsi dari Pandas untuk menyimpan DataFrame sebagai tabel di dalam database SQLite.
 - **df_od, df_pd, df_sd, df_cd**: DataFrame yang diubah menjadi tabel SQL bernama order_detail, payment_detail, sku_detail, dan customer_detail.
 - **index=False**: Menghindari penyimpanan index DataFrame.
 - **if_exists='replace'**: Jika tabel sudah ada, maka akan diganti dengan data baru.



DATA PREPROCESSING

```
] #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)
```

MERGE DATA WITH JOIN

KETERANGAN :

1. **pd.read_sql()**: Mengambil hasil query SQL dan menyimpannya sebagai DataFrame Pandas.
2. **SELECT**: Memilih kolom yang akan diambil dari beberapa tabel:
 - **order_detail.***: Memilih semua kolom dari tabel order_detail.
 - Kolom-kolom spesifik dari tabel lain, seperti payment_method, sku_name, base_price, cogs, category, dan registered_date.
3. **LEFT JOIN**: Bergabung dengan tabel lain menggunakan metode LEFT JOIN, yang memastikan semua baris dari order_detail diambil, meskipun tidak ada kecocokan di tabel yang digabungkan:
 - payment_detail dihubungkan melalui payment_id.
 - sku_detail dihubungkan melalui sku_id.
 - customer_detail dihubungkan melalui customer_id.
4. **conn**: Menggunakan koneksi SQLite yang telah dibuat sebelumnya untuk menjalankan query SQL.



DATA PREPROCESSING

CHECKING VARIABLE VALUE AFTER DATA MERGED

```
#Mengambil 5 baris pertama
df.head()

  id customer_id order_date sku_id price qty_ordered before_discount discount_amount after_discount is_gross is_valid is_net payment_id payment
0 ODR9939707760w C713589L 2021-11-19 P858068 26100 200 5220000.0 2610000.00 2610000.00 1 1 0 5
1 ODR7448356649d C551551L 2021-11-19 P886455 1971942 5 9859710.0 2464927.50 7394782.50 1 0 0 5
2 ODR4011281866z C685596L 2021-11-25 P678648 7482000 1 7482000.0 2065344.62 5416655.38 1 0 0 4
3 ODR3378927994s C830683L 2021-11-22 P540013 3593680 1 3593680.0 1455440.40 2138239.60 1 1 1 5
4 ODR4904430099k C191766L 2021-11-21 P491032 4413220 1 4413220.0 1059172.80 3354047.20 1 1 1 4
```

KETERANGAN : Mengecek 5 teratas bentuk nilai/data dalam variable df



DATA PREPROCESSING

DATA TYPE CHECKING

```
#Menampilkan tipe data tiap kolom
df.dtypes
```

| | 0 |
|-----------------|---------|
| 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 |
| dtype: | object |

Hal ini bertujuan untuk mengetahui setiap tipe data yang ada pada setiap kolom, serta mengidentifikasi kolom mana yang perlu diubah tipe datanya agar mempermudah proses analisa.

Dari sini diketahui beberapa kolom yang perlu diubah adalah **before_discount**, **discount_amount**, **after_discount**, **base_price**, **order_date** dan **registered_date**.



DATA PREPROCESSING

DATA TYPE CONVERSION

```
#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
```

| | 0 |
|-----------------|--------|
| id | object |
| customer_id | object |
| order_date | object |
| 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 | object |
| dtype: | object |

```
#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
```

| | 0 |
|-----------------|----------------|
| 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] |
| dtype: | object |

Mengubah tipe data kolom-kolom yang sebelumnya telah ditentukan menjadi tipe data yang lebih mudah untuk diproses dan dianalisa, yaitu menjadi tipe data int dan datetime.



PROBLEM STATEMENT 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



ANALYSIS & INSIGHT



SYNTAX

PROBLEM STATEMENT 1

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Filter data untuk kategori 'Mobiles & Tablets', transaksi valid (is_valid = 1),
# dan pesanan yang terjadi di tahun 2022
top5 = pd.DataFrame(
    df[
        (df['category'] == 'Mobiles & Tablets') & # Memfilter kategori 'Mobiles & Tablets'
        (df['is_valid'] == 1) & # Memfilter transaksi valid (is_valid = 1)
        (df['order_date'].dt.year == 2022) # Memfilter transaksi yang terjadi di tahun 2022
    ]
    # Mengelompokkan data berdasarkan SKU (sku_name) dan menjumlahkan qty_ordered
    .groupby(by=['sku_name'])['qty_ordered']
    .sum() # Menjumlahkan total qty_ordered untuk setiap sku_name
    .sort_values(ascending=False) # Mengurutkan hasil dari terbesar ke terkecil
    .head() # Mengambil 5 SKU dengan jumlah qty_ordered tertinggi
    .reset_index(name='quantity_order_2022') # Mengatur ulang indeks dan mengganti nama kolom hasil agregasi
)

# Menampilkan DataFrame top 5 produk dengan qty_ordered tertinggi di kategori 'Mobiles & Tablets' untuk tahun 2022
top5
```

memfilter data transaksi dari kategori "Mobiles & Tablets" yang valid (`is_valid == 1`) dan terjadi di tahun 2022, kemudian mengelompokkan data berdasarkan nama produk (`sku_name`) dan menghitung total kuantitas yang dipesan (`qty_ordered`) untuk setiap produk.

Hasilnya diurutkan dari yang terbesar ke yang terkecil, dan hanya 5 produk teratas dengan kuantitas pemesanan tertinggi yang diambil. Data akhir dimasukkan ke dalam DataFrame `top5` dengan kolom yang diberi nama `quantity_order_2022`.

```
# Mengurutkan DataFrame terlebih dahulu (jika belum)
top5 = top5.sort_values(by='quantity_order_2022', ascending=False)

# Membuat horizontal bar chart
top5.plot(
    x='sku_name',
    y='quantity_order_2022',
    kind='barh',
    grid=True,
    xlabel='Quantity',
    ylabel='Product Name',
    figsize=(12, 6),
    title='TOP 5 Product'
)

# Membalik sumbu y agar nilai terbesar di atas
plt.gca().invert_yaxis()

plt.show()
```

Membuat visualisasi bar chart dari dataframe



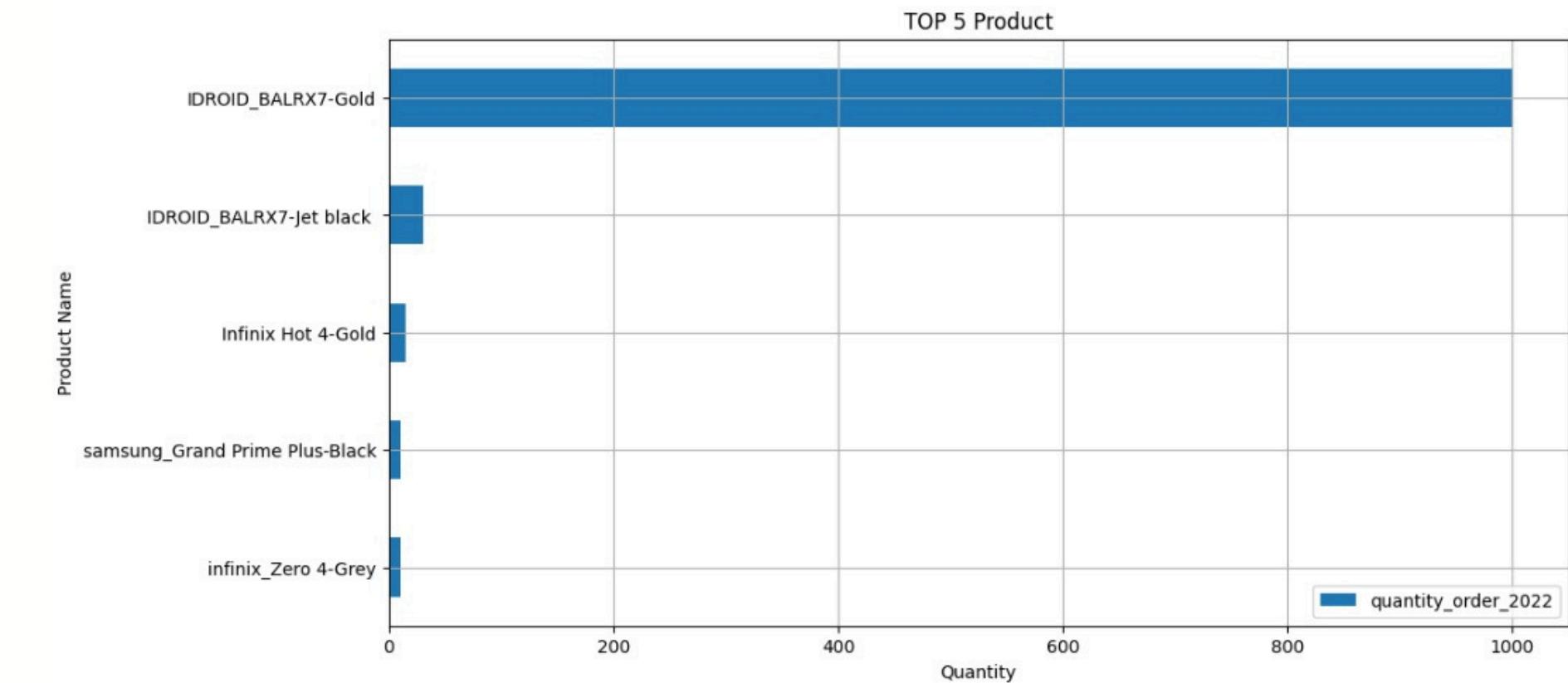
ANALYSIS & INSIGHT



RESULT & INSIGHT

PROBLEM STATEMENT 1

| | sku_name | quantity_order_2022 |
|---|--------------------------------|---------------------|
| 0 | IDROID_BALRX7-Gold | 1000 |
| 1 | IDROID_BALRX7-Jet black | 31 |
| 2 | Infinix Hot 4-Gold | 15 |
| 3 | samsung_Grand Prime Plus-Black | 11 |
| 4 | infinix_Zero 4-Grey | 10 |



INSIGHT

Analisis data menunjukkan bahwa lima produk terlaris dalam kategori Mobiles & Tablets adalah Idroid BALRX7-Gold, Idroid BALRX7-Jet Black, Infinix Hot 4-Gold, Samsung Grand Prime Plus-Black, dan Infinix Zero 4-Grey. Di antara semua produk tersebut, **Idroid BALRX7-Gold menempati posisi teratas dengan sales mencapai 1000 dan merupakan pilihan favorit di kalangan konsumen, sehingga bisa menjadi hadiah yang menarik untuk pemenang Festival Akhir Tahun.**



PROBLEM STATEMENT 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



ANALYSIS & INSIGHT



SYNTAX FOR ANALYSIS

PROBLEM STATEMENT 2.1

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode

# Menghitung total qty_ordered per kategori untuk tahun 2021
dp2021 = pd.DataFrame(
    df[
        (df['is_valid'] == 1) & # Memfilter transaksi yang valid (is_valid = 1)
        (df['order_date'].dt.year == 2021) # Memfilter transaksi yang terjadi di tahun 2021
    ]
    # Mengelompokkan data berdasarkan kategori dan menjumlahkan qty_ordered
    .groupby(by='category')['qty_ordered']
    .sum() # Menjumlahkan 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 menjadi 'qty_2021'
)

# Menghitung total qty_ordered per kategori untuk tahun 2022
dp2022 = pd.DataFrame(
    df[
        (df['is_valid'] == 1) & # Memfilter transaksi yang valid (is_valid = 1)
        (df['order_date'].dt.year == 2022) # Memfilter transaksi yang terjadi di tahun 2022
    ]
    # Mengelompokkan data berdasarkan kategori dan menjumlahkan qty_ordered
    .groupby(by='category')['qty_ordered']
    .sum() # Menjumlahkan total qty_ordered untuk 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 menjadi 'qty_2022'
)
```

menghitung total kuantitas yang dipesan (qty_ordered) per kategori produk untuk dua tahun berbeda, yaitu 2021 dan 2022. Pada setiap blok, data difilter untuk menyertakan hanya transaksi yang valid (is_valid == 1) dan yang terjadi pada tahun yang relevan. Kemudian, data dikelompokkan berdasarkan kategori produk, dan total kuantitas yang dipesan dijumlahkan untuk setiap kategori. Hasilnya diurutkan dari yang tertinggi ke terendah dan disimpan dalam DataFrame baru, dp2021 untuk tahun 2021 dan dp2022 untuk tahun 2022

```
# Merge dataframes
dpgrowth = dp2021.merge(dp2022, on='category')

# Calculate growth
dpgrowth['qty_growth'] = dpgrowth['qty_2022'] - dpgrowth['qty_2021']

# Sort the growth data
dpgrowth.sort_values(by='qty_growth', ascending=True, inplace=True)

# Display the result
dpgrowth
```

menggabungkan dua DataFrame (dp2021 dan dp2022) yang berisi total kuantitas yang dipesan per kategori untuk tahun 2021 dan 2022. Penggabungan dilakukan berdasarkan kolom category untuk menghasilkan DataFrame baru bernama dpgrowth. Setelah itu, kolom baru qty_growth dihitung dengan mengurangi qty_2021 dari qty_2022, yang menunjukkan perubahan dalam kuantitas yang dipesan antara kedua tahun tersebut. Data kemudian diurutkan berdasarkan qty_growth untuk menampilkan kategori yang mengalami penurunan kuantitas paling signifikan terlebih dahulu, sebelum akhir ditampilkan



ANALYSIS & INSIGHT



SYNTAX FOR VISUALIZATION

PROBLEM STATEMENT 2.1

```
# Mengurutkan data berdasarkan qty_growth
dpgrowth.sort_values(by='qty_growth', ascending=False, inplace=True)

# Membuat horizontal bar chart
ax = dpgrowth.plot(
    x='category', # Sumbu x adalah kategori produk
    y='qty_growth', # Sumbu y adalah pertumbuhan kuantitas
    kind='barh', # Membuat bar chart horizontal
    color='skyblue', # Warna bar chart
    grid=True, # Menambahkan grid
    xlabel='Product Category', # Label sumbu x
    ylabel='Quantity Growth', # Label sumbu y
    figsize=(12, 7), # Ukuran figure
    title='Growth in Quantity by Category (2022 vs 2021)' # Judul chart
)

# Menambahkan label nilai di sebelah kanan bar
for i in ax.patches:
    ax.text(i.get_width() + 0.2, # Sedikit jarak dari bar
            i.get_y() + i.get_height()/2, # Menempatkan di tengah bar
            round(i.get_width(), 2), # Menampilkan nilai pertumbuhan kuantitas
            ha='left', va='center', fontsize=11) # Menempatkan label di kanan

# Menampilkan bar chart
plt.tight_layout()
plt.show()
```



membuat horizontal bar chart yang menggambarkan pertumbuhan kuantitas produk dari tahun 2021 ke 2022. Data diurutkan berdasarkan pertumbuhan (qty_growth) dari yang tertinggi ke terendah, dengan kategori sebagai sumbu x dan pertumbuhan kuantitas sebagai sumbu y. Label nilai ditambahkan di sebelah kanan setiap bar untuk menunjukkan besarnya pertumbuhan



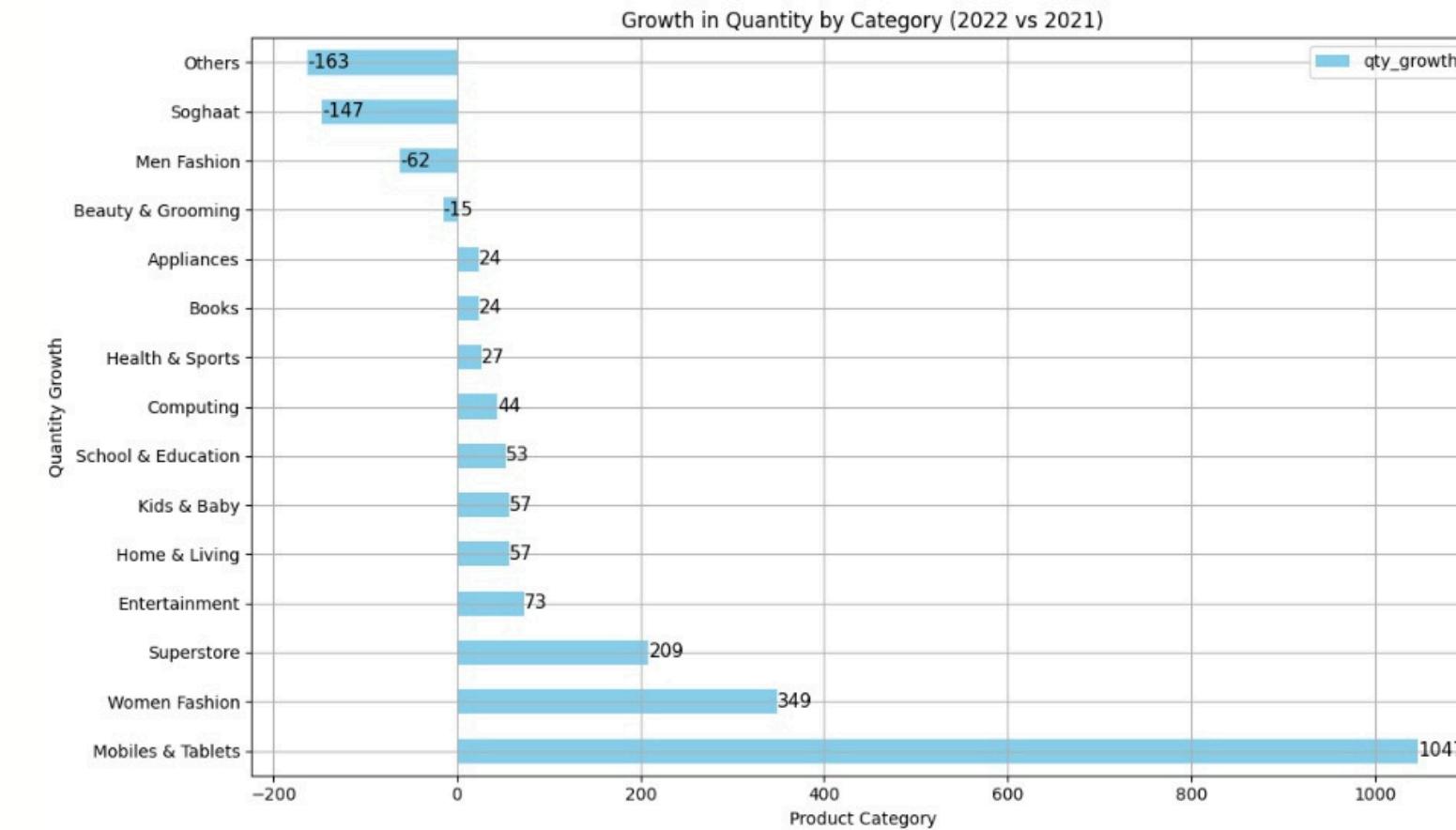
ANALYSIS & INSIGHT



RESULT & INSIGHT

PROBLEM STATEMENT 2.1

| | category | qty_2021 | qty_2022 | qty_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

Berdasarkan data yang diperoleh, ternyata dugaan tim warehouse benar bahwa **terjadi penurunan kuantitas penjualan pada kategori Others di tahun 2022 jika dibandingkan dengan tahun sebelumnya, yaitu sebesar -163**. Oleh karena itu, diperlukan rencana lebih lanjut untuk melakukan analisis lebih mendalam untuk mengidentifikasi produk-produk yang mengalami penurunan tersebut.



ANALYSIS & INSIGHT



SYNTAX FOR ANALYSIS

PROBLEM STATEMENT 2.2

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Menghitung kuantitas untuk kategori 'Others' pada tahun 2021
others_qty_2021 = pd.DataFrame(
    df[
        (df['is_valid'] == 1) &
        (df['category'] == 'Others') &
        (df['order_date'].dt.year == 2021)
    ]
    .groupby(by='sku_name')['qty_ordered']
    .sum()
    .sort_values(ascending=False)
    .reset_index(name='qty_2021')
)

# Menghitung kuantitas untuk kategori 'Others' pada tahun 2022
others_qty_2022 = pd.DataFrame(
    df[
        (df['is_valid'] == 1) &
        (df['category'] == 'Others') &
        (df['order_date'].dt.year == 2022)
    ]
    .groupby(by='sku_name')['qty_ordered']
    .sum()
    .sort_values(ascending=False)
    .reset_index(name='qty_2022')
)

# Menggabungkan data
data_check = others_qty_2021.merge(others_qty_2022, how='outer')
data_check.fillna(0, inplace=True)

# Top 20 nama produk yang mengalami penurunan paling tinggi pada tahun 2022 dibandingkan dengan tahun 2021
data_check['qty_growth'] = data_check['qty_2022'] - data_check['qty_2021']
data_check.sort_values(by='qty_growth', ascending=True, inplace=True)
data_check = data_check.head(20)

data_check
```

menghitung total kuantitas yang dipesan (qty_ordered) untuk kategori "Others" pada tahun 2021 dan 2022. Data difilter untuk transaksi yang valid, kemudian dikelompokkan berdasarkan nama produk (sku_name) dan dijumlahkan. Hasilnya diurutkan dari yang tertinggi ke terendah dan disimpan dalam dua DataFrame terpisah: others_qty_2021 untuk tahun 2021 dan others_qty_2022 untuk tahun 2022.

menggabungkan DataFrame others_qty_2021 dan others_qty_2022 berdasarkan kategori produk dengan metode penggabungan outer. Nilai yang hilang diisi dengan nol menggunakan fillna(0). Selanjutnya, kolom qty_growth dihitung untuk menunjukkan perbedaan kuantitas antara tahun 2022 dan 2021. Data kemudian diurutkan berdasarkan qty_growth dari yang terendah ke tertinggi, dan hanya 20 produk dengan penurunan terbesar yang diambil untuk analisis lebih lanjut, disimpan dalam DataFrame data_check.



ANALYSIS & INSIGHT



SYNTAX FOR VISUALIZATION

PROBLEM STATEMENT 2.2

```
# Mengurutkan dan memilih 20 produk teratas dengan penurunan terbesar
data_check = data_check.sort_values(by='qty_growth', ascending=True).head(20)

# Membuat horizontal bar chart
data_check.plot(
    x='sku_name', # Menggunakan nama produk sebagai sumbu x
    y='qty_growth', # Menggunakan pertumbuhan kuantitas sebagai sumbu y
    kind='barh', # Membuat horizontal bar chart
    grid=True, # Menambahkan grid
    xlabel='Product Name', # Label sumbu x
    ylabel='Quantity Growth', # Label sumbu y
    figsize=(12, 7), # Ukuran plot
    title='Top 20 Products with the Largest Decline in Quantity (2022 vs 2021)' # Judul
)

# Membalik sumbu y agar produk dengan penurunan terbesar berada di atas
plt.gca().invert_yaxis()

# Menampilkan bar chart
plt.tight_layout()
plt.show()
```



membuat visualisasi horizontal bar chart yang menunjukkan 20 produk dengan penurunan kuantitas terbesar antara tahun 2021 dan 2022. Data diurutkan berdasarkan qty_growth dalam urutan menaik, kemudian bar chart dibuat dengan nama produk sebagai sumbu x dan pertumbuhan kuantitas sebagai sumbu y.



ANALYSIS & INSIGHT



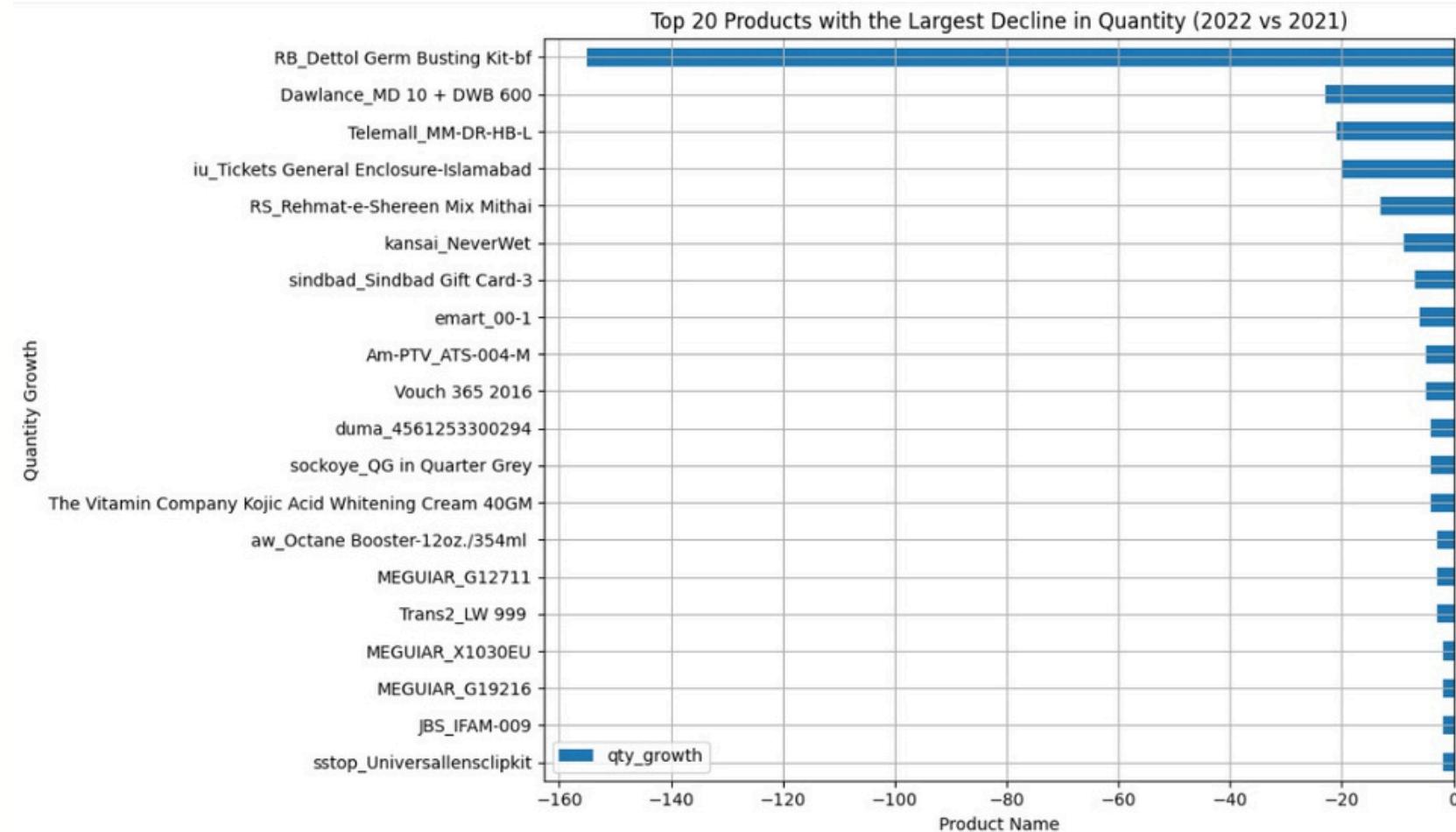
RESULT & INSIGHT

PROBLEM STATEMENT 2.2

| | sku_name | qty_2021 | qty_2022 | qty_growth |
|-----|---|----------|----------|------------|
| 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 | sockeye_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

Berdasarkan analisis lebih lanjut, kami dapat memastikan bahwa terjadi penurunan kuantitas penjualan dalam kategori Others pada tahun 2022. Produk **yang mengalami penurunan paling signifikan adalah RB_Dettol Germ Busting Kit-bf, dengan penurunan mencapai -155 unit. Selain itu, terdapat beberapa produk yang tidak terjual sama sekali sepanjang tahun 2022.**



RECOMMENDATIONS

- Evaluasi Produk Tidak Terjual:** Produk yang tidak mencatat penjualan pada tahun 2022 perlu dievaluasi untuk menentukan apakah produk tersebut masih relevan untuk pasar atau apakah perlu dihapus dari daftar inventaris.
- Rebranding atau Penawaran Promosi:** Pertimbangkan untuk melakukan rebranding atau memberikan penawaran khusus seperti bundling produk, diskon eksklusif, atau insentif lainnya untuk meningkatkan penjualan produk yang mengalami penurunan tajam.



PROBLEM STATEMENT 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



ANALYSIS & INSIGHT



SYNTAX

PROBLEM STATEMENT 3

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Filtering data untuk dt_promo
dt_promo = df[
    (df['is_gross'] == 1) &
    (df['is_valid'] == 0) &
    (df['is_net'] == 0) &
    (df['order_date'].dt.year == 2022)
]

# Memilih kolom customer_id dan registered_date
dt_custpromo = dt_promo[['customer_id', 'registered_date']].drop_duplicates()
dt_custpromo
```

memfilter data dari DataFrame df berdasarkan beberapa kriteria untuk menghasilkan DataFrame baru dt_promo. Data difilter untuk transaksi dengan is_gross sama dengan 1, is_valid sama dengan 0, is_net sama dengan 0, dan terjadi di tahun 2022. Setelah itu, kolom customer_id dan registered_date dipilih dari DataFrame yang telah difilter dan tidak ada duplicates, dan hasilnya disimpan dalam DataFrame dt_custpromo

```
#Jalankan kode ini untuk mendownload file
from google.colab import files
dt_custpromo.to_csv('audience_list.csv', encoding = 'utf-8-sig', index=False) #ganti [nama variabel file] dengan nama variabel yang digunakan
files.download('audience_list.csv')
```

mendownload DataFrame dt_custpromo sebagai file CSV. Pertama, modul files dari Google Colab diimpor untuk mengakses fungsi pengunduhan. Kemudian, DataFrame dt_custpromo disimpan sebagai file CSV bernama audience_list.csv dengan encoding UTF-8. File ini yang akan dikirimkan untuk tim Digital Marketing



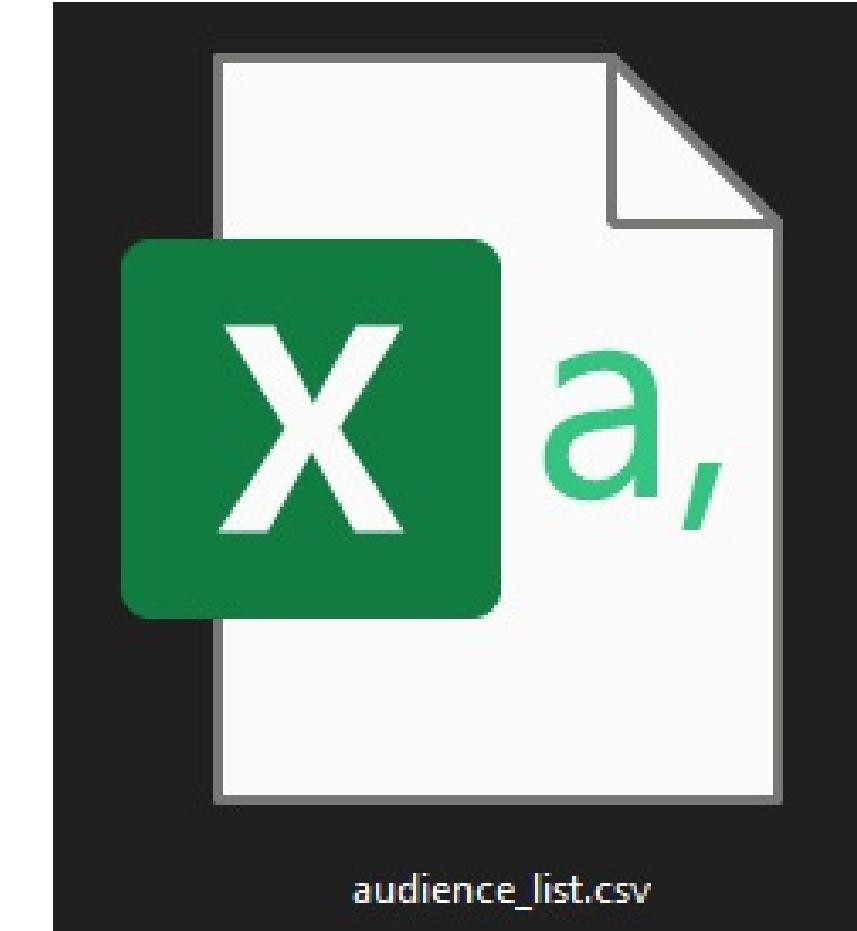
ANALYSIS & INSIGHT



RESULT & INSIGHT

PROBLEM STATEMENT 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 × 2 columns | | |



INSIGHT

Berdasarkan data yang dianalisis, **terdapat 820 pelanggan yang memenuhi syarat untuk menerima informasi mengenai promosi ulang tahun perusahaan**. File tersebut telah berhasil diunduh dan siap untuk diserahkan kepada tim pemasaran digital.



PROBLEM STATEMENT 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



ANALYSIS & INSIGHT



SYNTAX FOR ANALYSIS

PROBLEM STATEMENT 4.1

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Membuat kolom tambahan untuk day, month, month_num
df['day'] = df['order_date'].dt.day_name()
df['month'] = df['order_date'].dt.month_name() # Memperbaiki typo
df['month_num'] = df['order_date'].dt.month # Memperbaiki typo

# Filter penjualan weekends Oktober - Desember
weekends = pd.DataFrame( # Memperbaiki typo
    df[
        (df['is_valid'] == 1) &
        (df['day'].isin(['Saturday', 'Sunday'])) & # Menambahkan kurung
        (df['order_date'] >= '2022-10-01') &
        (df['order_date'] <= '2022-12-31') # Memperbaiki batas tanggal
    ].groupby(by=["month_num", "month"])["before_discount"]
    .mean()
    .round()
    .sort_values(ascending=False)
    .reset_index(name='avg_sales_weekend')
)

# Filter penjualan weekdays Oktober - Desember
weekdays = pd.DataFrame(
    df[
        (df['is_valid'] == 1) &
        (df['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])) & # Menambahkan kurung dan memperbaiki huruf kapital
        (df['order_date'] >= '2022-10-01') &
        (df['order_date'] <= '2022-12-31') # Memperbaiki batas tanggal
    ].groupby(by=["month_num", "month"])["before_discount"]
    .mean()
    .sort_values(ascending=False)
    .round()
    .reset_index(name='avg_sales_weekdays')
)

# Menggabungkan data penjualan weekends & weekdays
monthly = weekends.merge(weekdays, on='month') # Memperbaiki nama variabel
monthly.sort_values(by='month_num_x', ascending=True, inplace=True)

# Menampilkan output akhir
monthly = monthly[['month', 'avg_sales_weekend', 'avg_sales_weekdays']]
df_groupby_weekend = monthly

# Membuat kolom tambahan untuk diff_value dan diff_percent
df_groupby_weekend['diff_value'] = df_groupby_weekend['avg_sales_weekend'] - df_groupby_weekend['avg_sales_weekdays']
df_groupby_weekend['diff_percent'] = round(df_groupby_weekend['diff_value'] / df_groupby_weekend['avg_sales_weekend'] * 100) # Memperbaiki nama variabel

# Menampilkan hasil
df_groupby_weekend
```

menambahkan kolom baru ke DataFrame df, termasuk nama hari, nama bulan, dan nomor bulan berdasarkan kolom order_date. Kemudian, dua DataFrame baru dibuat: weekends untuk menghitung rata-rata penjualan pada akhir pekan dan weekdays untuk menghitung rata-rata penjualan pada hari kerja, masing-masing dari Oktober hingga Desember 2022. Hasilnya diurutkan berdasarkan rata-rata penjualan tertinggi ke terendah.

menggabungkan data penjualan akhir pekan dan hari kerja ke dalam DataFrame monthly, lalu diurutkan berdasarkan nomor bulan. Data ditampilkan dengan kolom avg_sales_weekend dan avg_sales_weekdays. Dua kolom tambahan dibuat: diff_value untuk menunjukkan selisih penjualan antara akhir pekan dan hari kerja, serta diff_percent untuk menghitung persentase perbedaan tersebut. Hasil akhir disimpan dalam df_groupby_weekend.



ANALYSIS & INSIGHT



SYNTAX FOR VISUALIZATION

PROBLEM STATEMENT 4.1

```
# Membuat visualisasi
df_groupby_weekend.plot(
    x='month',
    y=['avg_sales_weekend', 'avg_sales_weekdays'],
    kind='bar',
    grid=True,
    xlabel='Bulan',
    ylabel='Total',
    figsize=(12, 7),
    table=False,
    title='Average Sales Weekend vs Weekday October - December 2022'
)
```



membuat visualisasi bar chart yang membandingkan rata-rata penjualan antara akhir pekan dan hari kerja berdasarkan bulan. Data diambil dari DataFrame df_groupby_weekend



ANALYSIS & INSIGHT



RESULT & INSIGHT

PROBLEM STATEMENT 4.1

| | month | avg_sales_weekend | avg_sales_weekdays | diff_value | diff_percent |
|---|----------|-------------------|--------------------|------------|--------------|
| 0 | October | 634260.0 | 874690.0 | -240430.0 | -38.0 |
| 1 | November | 607794.0 | 641862.0 | -34068.0 | -6.0 |
| 2 | December | 410599.0 | 813574.0 | -402975.0 | -98.0 |



INSIGHT

Berdasarkan analisis data, terlihat bahwa rata-rata penjualan akhir pekan mengalami penurunan dibandingkan dengan hari kerja selama periode Oktober hingga Desember 2022. Pada bulan Oktober, rata-rata penjualan akhir pekan mencapai 634,260.0, sementara rata-rata penjualan hari kerja adalah 874,690.0, menghasilkan selisih sebesar -240,430.0 atau penurunan 38.0%. Di bulan November, rata-rata penjualan akhir pekan adalah 607,794.0, dengan rata-rata penjualan hari kerja sebesar 641,862.0, menciptakan selisih yang lebih kecil yaitu -34,068.0, setara dengan penurunan 6.0%.

Namun, bulan Desember mencatatkan rata-rata penjualan akhir pekan terendah, yaitu 410,599.0, dibandingkan dengan 813,574.0 untuk hari kerja, yang mengakibatkan selisih sebesar -402,975.0 atau penurunan dramatis sebesar 98.0%.



ANALYSIS & INSIGHT



SYNTAX FOR ANALYSIS

PROBLEM STATEMENT 4.2

```
# Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
# Filter penjualan weekends Oktober - Desember
weekend_2022 = df[
    (df['is_valid'] == 1) &
    (df['day'].isin(['Saturday', 'Sunday'])) &
    (df['order_date'] >= '2022-10-01') & (df['order_date'] <= '2022-12-31')
]

# Filter penjualan weekdays Oktober - Desember
weekday_2022 = df[
    (df['is_valid'] == 1) &
    (df['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])) &
    (df['order_date'] >= '2022-10-01') & (df['order_date'] <= '2022-12-31')
]

# Mencari rata-rata penjualan weekend dan weekday selama bulan Oktober sampai Desember 2022
total_months = {
    "Periode": "Total 3 Months",
    "Avg Weekend Sales": round(weekend_2022['before_discount'].mean(), 2),
    "Avg Weekday Sales": round(weekday_2022['before_discount'].mean(), 2),
    "Diff (Value)": round(weekend_2022['before_discount'].mean(), 2) - round(weekday_2022['before_discount'].mean(), 2),
    "Diff(%)": round((weekend_2022['before_discount'].mean() - weekday_2022['before_discount'].mean()) / weekend_2022['before_discount'].mean() * 100, 2)
}

# Mengonversi dictionary ke DataFrame
pd.DataFrame([total_months])
```

memfilter data penjualan dari DataFrame df berdasarkan kriteria tertentu selama periode Oktober hingga Desember 2022. Pada bagian pertama, weekend_2022 menyimpan transaksi yang valid (is_valid == 1) yang terjadi pada hari Sabtu dan Minggu. Pada bagian kedua, weekday_2022 menyimpan transaksi yang valid yang terjadi pada hari kerja (Senin hingga Jumat) dalam periode yang sama. Kedua DataFrame ini memungkinkan analisis terpisah untuk penjualan akhir pekan dan hari kerja selama kuartal terakhir tahun 2022

menghitung rata-rata penjualan selama periode Oktober hingga Desember 2022 untuk penjualan akhir pekan dan hari kerja. Sebuah dictionary bernama total_months dibuat untuk menyimpan informasi seperti periode, rata-rata penjualan akhir pekan (Avg Weekend Sales), rata-rata penjualan hari kerja (Avg Weekday Sales), selisih nilai antara keduanya (Diff (Value)), dan persentase perbedaan (Diff(%)). Rata-rata dan selisih dihitung menggunakan metode .mean() dan diubah menjadi dua angka desimal.



ANALYSIS & INSIGHT



SYNTAX FOR VISUALIZATION

PROBLEM STATEMENT 4.2

```
# Membuat DataFrame untuk visualisasi
df_groupby_weekend = pd.DataFrame({
    'Periode': ['Weekend', 'Weekday'],
    'Avg Sales': [weekend_2022['before_discount'].mean(), weekday_2022['before_discount'].mean()]
})

# Membuat visualisasi bar chart
df_groupby_weekend.plot(
    x='Periode', # Menggunakan 'Periode' sebagai sumbu x
    y='Avg Sales', # Menampilkan rata-rata penjualan
    kind='bar', # Menggunakan bar chart vertikal
    grid=True, # Menambahkan grid
    xlabel='Periode', # Label untuk sumbu x
    ylabel='Average Sales', # Label untuk sumbu y
    figsize=(12, 6), # Ukuran plot
    title='Average Sales: Weekend vs Weekday (Oct-Dec 2022)' # Judul plot
)
```



membuat visualisasi bar chart yang membandingkan rata-rata penjualan akhir pekan dan hari kerja dari Oktober hingga Desember 2022. Sebuah DataFrame bernama df_groupby_weekend dibuat dengan kolom Periode untuk label "Weekend" dan "Weekday", serta kolom Avg Sales untuk rata-rata penjualan.



ANALYSIS & INSIGHT

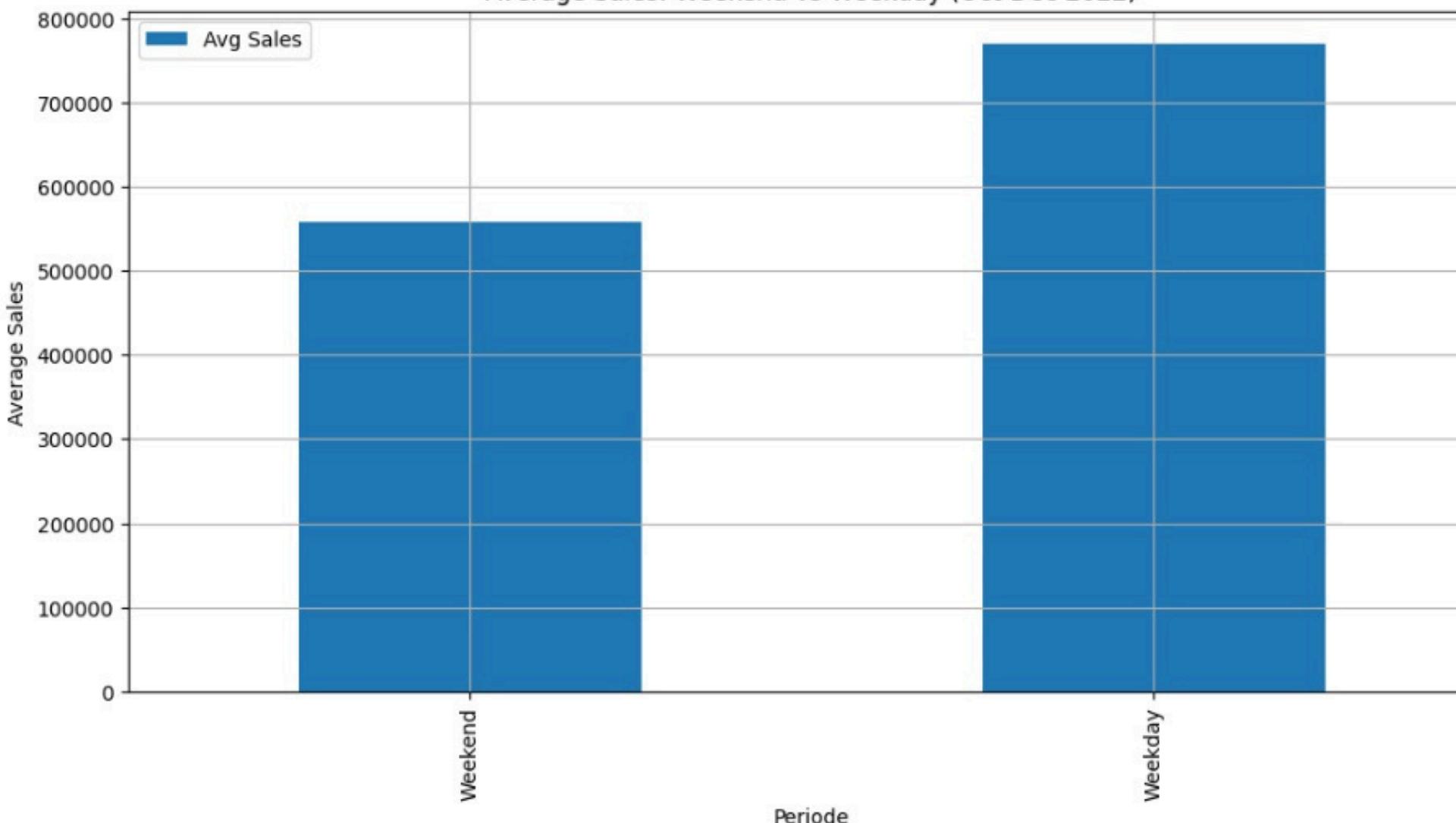


RESULT & INSIGHT

PROBLEM STATEMENT 4.2

| Periode | Avg Weekend Sales | Avg Weekday Sales | Diff (Value) | Diff(%) |
|------------------|-------------------|-------------------|--------------|---------|
| 0 Total 3 Months | 558865.15 | 770146.01 | -211280.86 | -37.81 |

Average Sales: Weekend vs Weekday (Oct-Dec 2022)



INSIGHT

Berdasarkan analisis data yang diperoleh, rata-rata penjualan akhir pekan selama **periode Oktober hingga Desember 2022 berada di bawah rata-rata penjualan hari kerja, dengan perbedaan sebesar -211.280,89 (-37.81%)**. Hal ini menunjukkan bahwa **kampanye yang dilakukan oleh Tim Campaign belum berhasil mencapai atau melebihi penjualan pada hari kerja**.

RECOMMENDATIONS

- **Evaluasi Kampanye Lebih Lanjut:** Adakan diskusi lintas tim (marketing dan produk) untuk mengevaluasi kampanye yang telah dilakukan dan mengidentifikasi faktor-faktor yang menyebabkan penjualan akhir pekan tidak optimal.
- **Kampanye Khusus Akhir Pekan:** Luncurkan kampanye khusus yang berfokus pada akhir pekan, seperti program loyalitas dengan hadiah untuk pembelian di hari Sabtu dan Minggu, atau promo eksklusif dengan batasan waktu selama akhir pekan.



ANALYSIS & INSIGHT



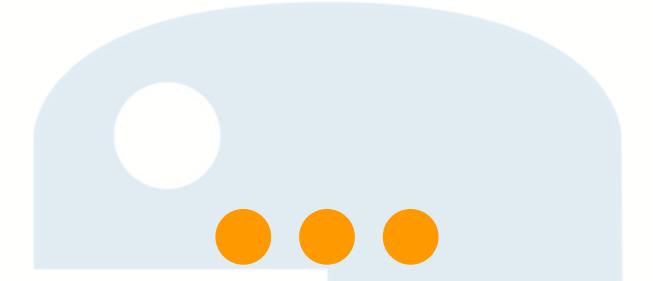
CONCLUSION

Dalam proyek ini, analisis data penjualan dilakukan untuk mengevaluasi kinerja produk dalam kategori Mobiles & Tablets serta kategori Others selama tahun 2022. Insight yang diperoleh menunjukkan bahwa produk terlaris, seperti Idroid BALRX7-Gold, memiliki potensi tinggi untuk promosi, sementara penurunan signifikan dalam kategori Others, terutama pada produk RB_Dettol Germ Busting Kit-bf, perlu ditangani dengan strategi pemasaran yang lebih efektif.

Rata-rata penjualan akhir pekan juga ditemukan lebih rendah dibandingkan dengan hari kerja, dengan selisih mencapai -37.81%. Hal ini mengindikasikan bahwa kampanye promosi yang diterapkan belum optimal dalam meningkatkan penjualan akhir pekan. Oleh karena itu, revisi dan pengujian strategi promosi yang lebih agresif pada akhir pekan sangat diperlukan untuk menarik lebih banyak pelanggan.

Secara keseluruhan, proyek ini menekankan pentingnya penggunaan Python dan library seperti Pandas dan Matplotlib dalam analisis data dan visualisasi. Pembelajaran yang diperoleh dari proyek ini dapat dimanfaatkan untuk merumuskan strategi pemasaran yang lebih tepat dan mengoptimalkan penjualan, sehingga perusahaan dapat mencapai tujuan bisnisnya dengan lebih baik di masa depan.





THANK YOU!



MOCHAMAD SYAHRIZAL



[Mochamad Syahrizal](#)



[MochSyahrizal](#)