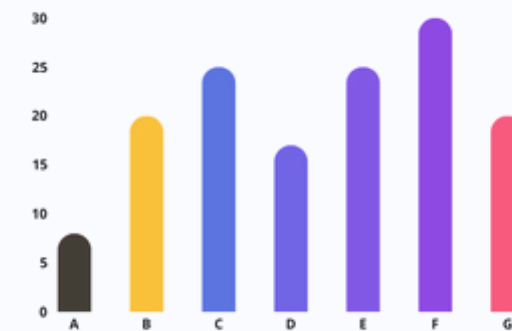


FINAL PROJECT DATA ANALYSIS - PYTHON

Batch 25 | Final Project Data Analyst MySkill

KELOMPOK C6

Annual
Revenue



Equity



from last month

Current YTD
(Q3)



Today's
Expenses

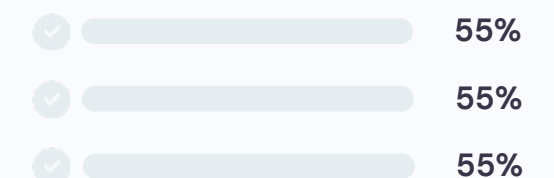
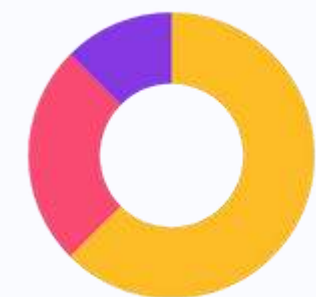


\$150K



\$100K

Total
Income



Loan Approval Rate



Project Team

A60 – BINTAN SANIA



A130 – FIKRI NURRAHMAN
HAKIM



A240 – MUHAMMAD
RAFLI IHSANUDDIN



A314 – SHINTIA BELLA



A16 – AHMAD FATHANNAFI



A49 – AVE LISA SIHOMBING



A96 – DWISTA ADELIA
PUTRI



A125 – FELISITAS
YOHANIA RENYA T. PULI



A185 – M YOGA IRVANDI



A256 – NAUFAL AKBAR



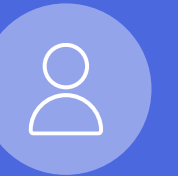
A332 – VANIA ZAHRA
ARDELIA



A237 – MUHAMMAD RAFA
ZULFIKAR MUSYARAF



A298 – RIZKY AMALIA DWI
CAHYANI



Training hours

100%

Women in leadership

100%

Table of Content



Project Background

Introduction
Expected Output



Data Understanding

Data set overview
Schema
Data Preparation



Data Analysis

Case Problem
Solution
Explanation & Insight

Project **Background**

Menjelang akhir tahun dan dalam rangka evaluasi serta perencanaan strategi bisnis, perusahaan membutuhkan dukungan analisis data penjualan dan perilaku pelanggan. Beberapa inisiatif strategis tengah dijalankan, mulai dari penentuan hadiah kompetisi Festival Akhir Tahun, evaluasi kinerja penjualan kategori tertentu, perencanaan promo ulang tahun perusahaan, hingga pengukuran efektivitas campaign akhir pekan.

Tim Marketing, Warehouse, Digital Marketing, dan Campaign memerlukan data yang akurat dan berbasis historis (tahun 2021–2022) untuk mendukung pengambilan keputusan yang tepat. Oleh karena itu, analisis data dilakukan untuk mengidentifikasi produk dengan performa penjualan tertinggi, mendeteksi penurunan kuantitas penjualan pada kategori tertentu, menyusun daftar pelanggan potensial untuk aktivitas promosi, serta mengevaluasi dampak campaign terhadap peningkatan penjualan. Hasil analisis ini diharapkan dapat menjadi dasar diskusi, perencanaan, dan penentuan strategi bisnis selanjutnya.

Data Understanding

Dataset yang digunakan

- order_detail
- sku_detail
- customer_detail
- payment_detail

customer_detail

id → angka unik dari produk (dapat digunakan untuk key saat join)

sku_name → nama dari produk

base_price → harga barang yang tertera pada tagging harga/price

cogs → cost of goods sold/total biaya untuk menjual 1 produk

category → kategori produk

order_detail

id → angka unik dari order/id_order

customer_id → angka unik dari pelanggan

order_date → tanggal saat dilakukan transaksi

sku_id → angka unik dari produk (sku adalah stock keeping unit)

price → harga yang tertera pada tagging harga

qty_ordered → jumlah barang yang dibeli oleh pelanggan

before_discount → nilai harga total dari produk ($\text{price} * \text{qty_ordered}$)

discount_amount → nilai diskon product total

after_discount → nilai harga total produk ketika sudah dikurangi dengan diskon

is_gross → menunjukkan pelanggan belum membayar pesanan

is_valid → menunjukkan pelanggan sudah melakukan pembayaran

is_net → menunjukkan transaksi sudah selesai

payment_id → angka unik dari metode pembayaran

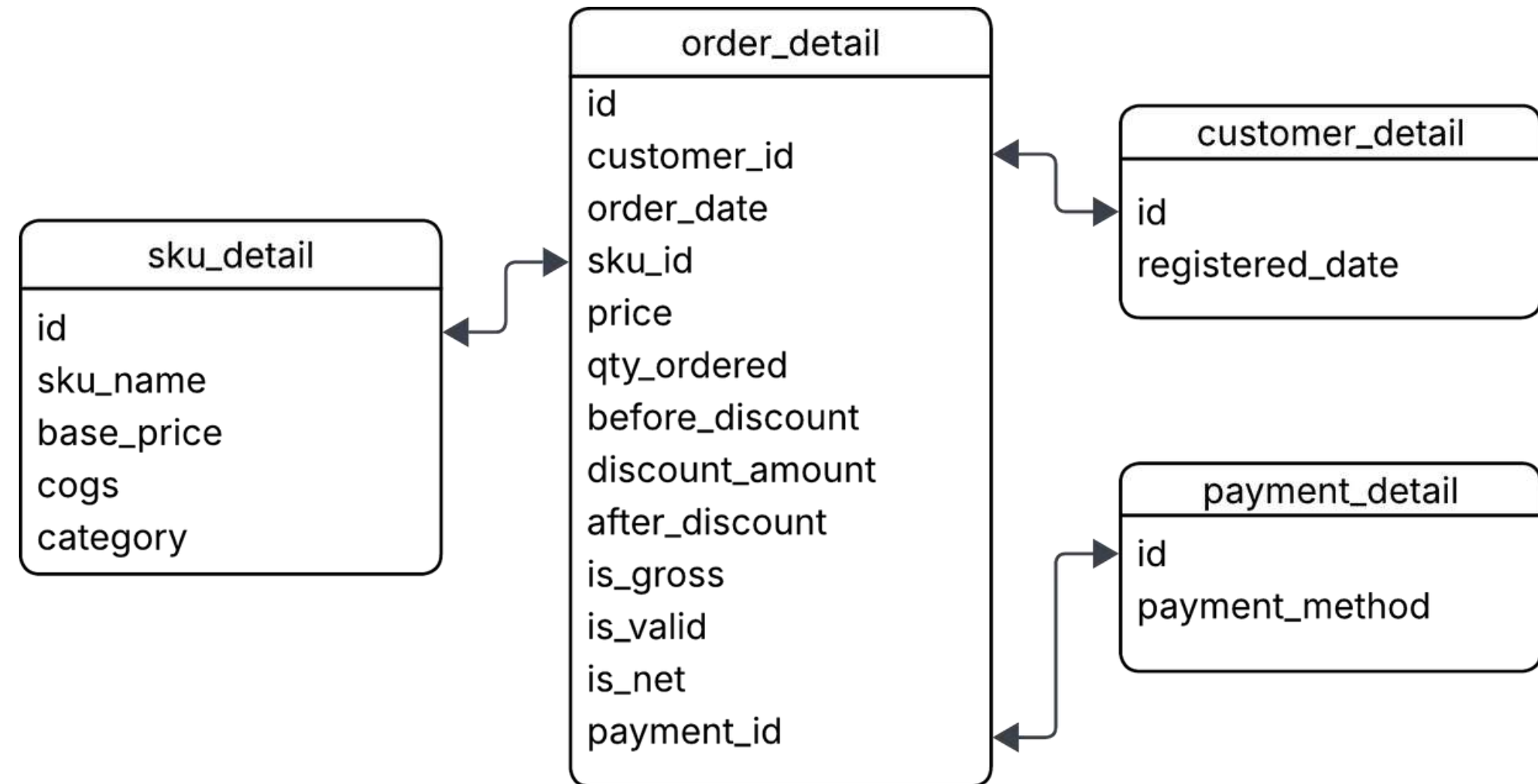
Schema

customer_detail

id → angka unik dari pelanggan
registered_date → tanggal pelanggan mulai mendaftarkan diri sebagai anggota

payment_detail

id → angka unik dari metode pembayaran
payment_method → metode pembayaran yang digunakan



Data Preparation

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

- **pandas as pd** → mengolah dan menganalisis data dalam bentuk tabel (DataFrame, Series)
- **numpy as np** → perhitungan operasi numerik dan manipulasi array
- **matplotlib.pyplot as plt & seaborn as sns** → membuat visualisasi data
- **BDay** dari **pandas.tseries.offsets** → operasi tanggal berbasis hari kerja

- **path_...** → menyimpan alamat file data yang berupa URL GitHub
- **pd.read_csv(...)** → membaca file CSV dari URL menjadi DataFrame

```
#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)
```

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

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

- **df_od.head()** → menampilkan 5 baris teratas data secara default

Data Preparation

```
#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')
```

3998

```
#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)
```

- **from sqlite3 import connect** → mengimport fungsi **connect** dari library sqlite3 untuk membuat koneksi database SQLite
- **conn = connect(':memory:')** → membuat database SQLite sementara di memori
- **df_od.to_sql(...)** → menyimpan DataFrame ke tabel SQL dengan DataFrame tidak disimpan sebagai kolom (**index=False**) dan jika tabel sudah ada, akan ditimpa (**if_exists='replace'**)
- **pd.read_sql ()** → menjalankan query SQL dan menyimpan hasilnya ke DataFrame pandas
- **conn** → koneksi database yang dibuat sebelumnya

Data Preparation

```
#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

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
```

- **pd.to_datetime(...)** → mengonversi data menjadi format tanggal/waktu (datetime)

Struktur kolom dan tipe data telah sesuai dengan kebutuhan analisis ditambah dengan dengan penyesuaian tipe tanggal untuk kebutuhan visualisasi data berbasis waktu

- **df.astype(...)** → mengubah tipe data kolom
- **df.dtypes** → menampilkan tipe data kolom

```
#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        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
```

Data Preparation

```
#Cek Duplikasi Data
df.duplicated(keep=False).sum()

np.int64(0)
```

`df.duplicated(keep=False).sum()` → menghitung baris yang memiliki duplikat

`df.isna().sum()` → menghitung nilai NULL yang terdapat di tiap kolom

Pengecekan duplikasi dan nilai kosong dilakukan pada seluruh kolom utama.
Tidak ditemukan duplikasi data maupun nilai kosong yang mempengaruhi perhitungan metrik utama, sehingga tidak diperlukan penanganan tambahan.

```
#Mengecek apakah ada nilai NULL di tiap kolom
print(df.isna().sum())

id                0
customer_id       0
order_date        0
sku_id            0
price             0
qty_ordered       0
before_discount   0
discount_amount   0
after_discount    0
is_gross          0
is_valid          0
is_net            0
payment_id        0
payment_method    0
sku_name          0
base_price        0
cogs              0
category          0
registered_date   0
dtype: int64
```

Case Problem

Case 1

Tentukan TOP 5 produk kategori Mobiles & Tablets dengan kuantitas penjualan tertinggi (valid = 1) selama tahun 2022 sebagai perkiraan hadiah Festival Akhir Tahun.

Case 2

1. Cek dan bandingkan kuantitas penjualan tahun 2021 vs 2022 untuk kategori tersebut, serta tampilkan 15 kategori, guna memverifikasi dugaan penurunan penjualan.
2. Menyediakan data TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 jika dibanding dengan 2021, apabila terjadi penurunan kuantitas penjualan pada kategori Others.

Case 3

Tampilkan ID Customer dan Registered Date untuk pelanggan yang sudah check-out namun belum membayar (is_gross = 1) selama tahun 2022.

Case 4

1. Tampilkan rata-rata harian penjualan (before_discount) antara weekend vs weekday per bulan Oktober–Desember 2022 untuk melihat peningkatan tiap bulan.
2. Tampilkan rata-rata harian penjualan weekend vs weekday secara keseluruhan selama Oktober–Desember 2022.

Case 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

CASE 1

TOP 5 produk kategori Mobiles & Tablets dengan kuantitas penjualan tertinggi (valid = 1) selama tahun 2022

SYNTAX CODE:

```
#Filter data valid = 1, tahun 2022 saja, dan Category = Mobiles & Tablets
df_filtered = df[
    (df["is_valid"] == 1) &
    (df["order_date"].dt.year == 2022) &
    (df["category"] == "Mobiles & Tablets")
]

#Groupby, urutkan data, dan ambil top 5
top_5_products = (
    df_filtered
    .groupby("sku_name", as_index=False)
    .agg(total_qty=("qty_ordered", "sum"))
    .sort_values(by="total_qty", ascending=False)
    .head(5)
)

#Output
top_5_products
```

Filtering

Memfilter dataframe dengan kondisi:

- Hanya mengambil data transaksi yang dianggap valid (valid = 1)
- Hanya mengambil transaksi pada tahun 2022
- Hanya untuk produk dalam kategori Mobiles & Tablets

Grouping, Agregation, Sorting

- Mengelompokkan data berdasarkan nama SKU (produk)
- Menghitung total kuantitas terjual setiap SKU
- Mengurutkan penjualan dari yang terbesar ke yang terkecil

Output

Menampilkan hasil akhir 5 SKU (produk) terlaris selama tahun 2022, dengan transaksi yang valid dan masuk dalam kategori Mobile & Tablets.

CASE 1

TOP 5 produk kategori Mobiles & Tablets dengan kuantitas penjualan tertinggi (valid = 1) selama tahun 2022

SYNTAX CODE CHART:

```
colors = ['#4C72B0', '#55A868', '#C44E52', '#8172B3', '#CCB974']
top_5_products.plot(
    kind='bar',
    x='sku_name',
    y='total_qty',
    legend=False,
    color=colors
)
plt.title('Top 5 Produk Mobiles & Tablets Tahun 2022')
plt.xlabel('Nama Produk')
plt.ylabel('Total Kuantitas Terjual')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Coloring

Mendefinisikan warna yang akan digunakan untuk mewarnai setiap bar pada chart.

Plotting Bar Chart

Menggunakan fungsi .plot() dari Pandas untuk membuat grafik.

Judul Chart

Menambahkan judul pada chart.

Label Sumbu x dan y

Sumbu x untuk nama produk dan sumbu y untuk total kuantitas yang terjual.

Rotasi Label Sumbu x

Merotasi teks label sumbu X sebesar 45 derajat agar tidak bertumpuk dan lebih mudah dibaca.

Layout Otomatis

Mengatur layout secara otomatis agar elemen chart tidak terpotong.

Output

Menampilkan chart sebagai hasil visualisasi.

Result Insight



Penjualan Sangat Terkonsentrasi Pada Satu Produk

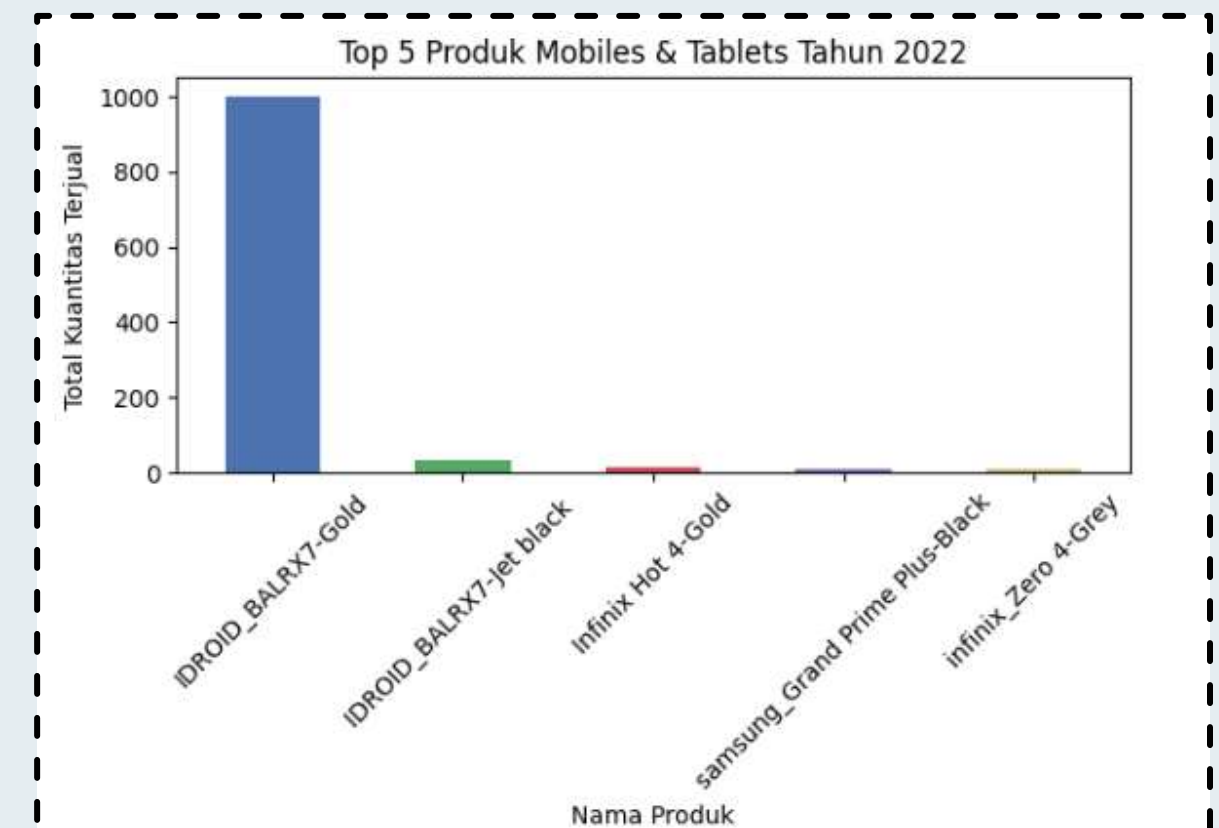
- Produk IDROID_BALRX7–Gold mencatat penjualan 1.000 unit, jauh melampaui produk lainnya.
- Produk peringkat kedua hanya terjual 31 unit, menunjukkan kesenjangan penjualan yang sangat besar.
- Produk ini menjadi key revenue driver dan sangat menentukan performa total penjualan.



Distribusi Penjualan Tidak Merata (Skewed Distribution)

- Lebih dari 90% total kuantitas Top 5 berasal dari satu produk saja.
- Empat produk lainnya memiliki penjualan relatif rendah (10–31 unit).
- Produk ini menjadi key revenue driver dan sangat menentukan performa total penjualan.

	sku_name	total_qty
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



Result Insight



Indikasi Preferensi Konsumen Terhadap “Color” Tertentu

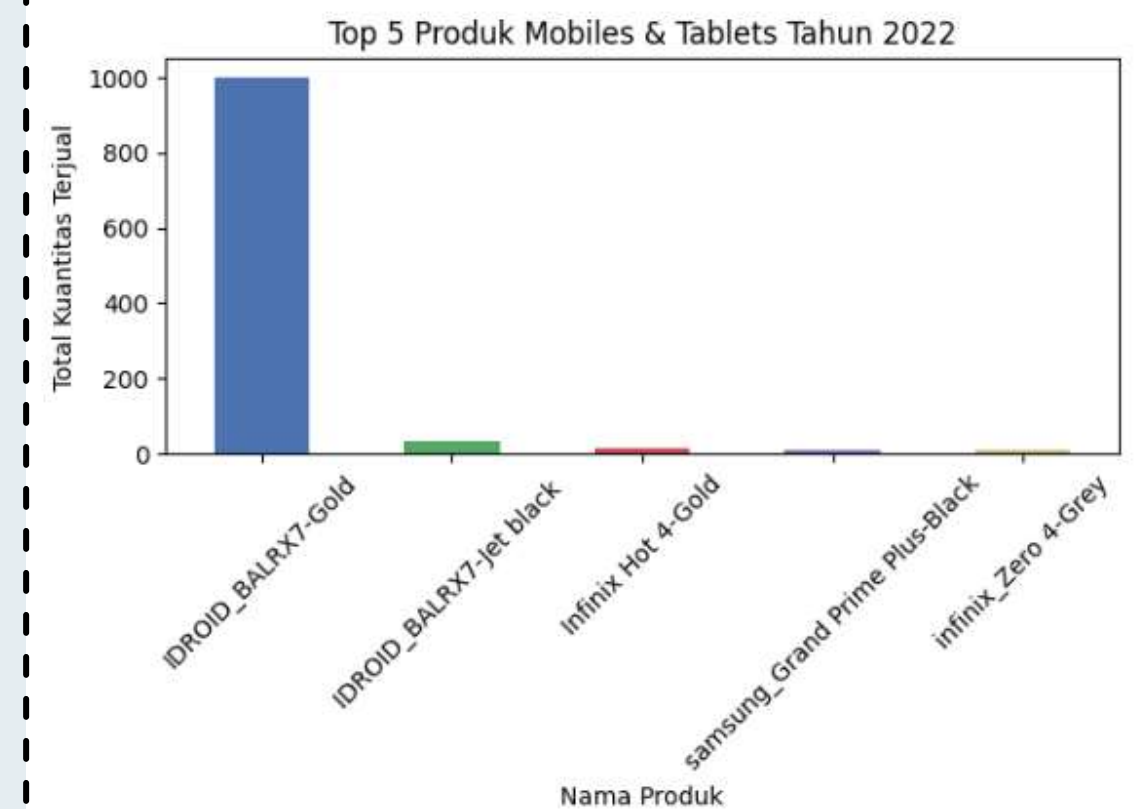
- Warna Gold muncul pada dua produk dengan performa relatif lebih baik dibanding warna lain.
- Warna dapat menjadi salah satu faktor yang memengaruhi keputusan pembelian



Produk Dengan Penjualan Rendah Berpotensi Menjadi Slow-Moving Item

- Beberapa produk berada pada level penjualan < 20 unit, yang berisiko:
 - Menumpuk stok
 - Menurunkan perputaran inventory
 - Membebani biaya penyimpanan

	sku_name	total_qty
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



Recommendations



**Prioritaskan
produk unggulan**

- Pastikan ketersediaan stok IDROID_BALRX7–Gold
- Fokuskan distribusi dan monitoring stok untuk mencegah out-of-stock
- Jadikan produk ini sebagai benchmark untuk produk lain



**Kurangi risiko
ketergantungan
pada satu produk**

- Kembangkan strategi untuk menaikkan penjualan produk lain
- Replikasi faktor sukses (harga, promo, positioning) dari produk unggulan



**Optimalkan produk
dengan penjualan rendah**

- Terapkan promo, diskon, atau bundling
- Evaluasi ulang strategi pricing
- Pertimbangkan pengurangan reorder untuk produk slow-moving



**Manfaatkan insight
preferensi warna**

- Fokuskan stok dan promosi pada warna yang lebih diminati
- Evaluasi ulang varian warna dengan performa rendah



**Rekomendasi
analisis lanjutan**

- Untuk mendukung pengambilan keputusan yang lebih kuat:
- Analisis penjualan per periode (trend)
- Kontribusi (%) tiap produk terhadap total penjualan
- Perbandingan margin vs kuantitas
- Analisis stok vs kecepatan penjualan (inventory turnover)

Conclusion

Penjualan Top 5 produk sangat didominasi oleh satu produk utama, sehingga kinerja penjualan saat ini belum seimbang dan berisiko jika terjadi gangguan pada produk tersebut. Beberapa produk lain menunjukkan penjualan rendah dan berpotensi menjadi slow-moving item. Di sisi lain, terdapat indikasi preferensi konsumen terhadap warna tertentu (Gold). Oleh karena itu, perusahaan perlu mengamankan produk unggulan, sekaligus mengoptimalkan dan menyeimbangkan portofolio produk agar pertumbuhan penjualan lebih berkelanjutan.

Case 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

CASE 2.1

SYNTAX CODE:

```
#Filter valid = 1 untuk tahun 2021 dan 2022
df_2021_soal2 = df_soal2[(df_soal2['is_valid'] == 1) & (df_soal2['order_date'].dt.year == 2021)]
df_2022_soal2 = df_soal2[(df_soal2['is_valid'] == 1) & (df_soal2['order_date'].dt.year == 2022)]

#Mengelompokkan (groupby) berdasarkan sku_name tahun 2021
penjualan_2021 = df_2021_soal2.groupby('category')['qty_ordered'].sum().reset_index()
penjualan_2021.columns = ['category', 'qty_ordered_2021']

#Mengelompokkan (groupby) berdasarkan sku_name tahun 2021
penjualan_2022 = df_2022_soal2.groupby('category')['qty_ordered'].sum().reset_index()
penjualan_2022.columns = ['category', 'qty_ordered_2021']

#Menggabungkan data penjualan_2021 dan penjualan_2022
compare_penjualan_category = pd.merge (penjualan_2021, penjualan_2022, on = 'category', how = 'outer')

#Melakukan pengurangan kolom qty 2022 dengan qty 2021 dengan menambahkan kolom baru
compare_penjualan_category ['selisih_qty'] = compare_penjualan_category ['qty_ordered_2022'] - compare_penjualan_category ['qty_ordered_2021']

#Mengurutkan selisih dari yang paling besar
compare_penjualan_category = compare_penjualan_category.sort_values (by = 'selisih_qty', ascending = True).head(15)

#Menampilkan 15 kategori
print(compare_penjualan_category)
```

CASE 2.1

FILTERING

- Tujuan → memisahkan data mentah menjadi dua kumpulan data (tahun 2021 dan 2022)
- `is_valid == 1` → hanya mengambil transaksi yang berhasil/valid (bukan canceled atau fraud)
- `.dt.year` → mengambil informasi tahun dari kolom tanggal untuk memisahkan transaksi tahun 2021 dan 2022

GROUPING

- Tujuan → menghitung total barang yang terjual per kategori.
- `groupby('category')` → mengelompokkan data berdasarkan kategori produk.
- `['qty_ordered'].sum()` → menjumlahkan seluruh kuantitas pesanan dalam kelompok tersebut.
- `reset_index()` & `.columns` → mengubah hasil perhitungan menjadi tabel (DataFrame) yang rapi dan memberi nama kolom yang spesifik agar tidak tertukar antara tahun 2021 dan 2022.

MERGING

- Tujuan → menggabungkan data tahun 2021 dan 2022 ke dalam satu tabel
- Logika → menggunakan fungsi `merge` dengan kunci `category` dan parameter `how='outer'` untuk memastikan jika ada kategori yang hanya muncul di satu tahun, kategori tersebut tetap muncul di tabel hasil

PERHITUNGAN SELISIH

- Tujuan → mencari angka pertumbuhan atau penurunan
- Logika → membuat kolom baru bernama `selisih_qty` dengan rumus: `Qty 2022 - Qty 2021`. Jika hasilnya negatif maka terjadi penurunan penjualan

SORTING

- Tujuan → menemukan kategori yang mengalami penurunan paling tinggi
- Logika → `ascending=True` untuk mengurutkan dari angka terkecil (paling negatif) dan `.head(15)` untuk mengambil 15 baris teratas

Result Insight



BOTTOM PERFORMERS

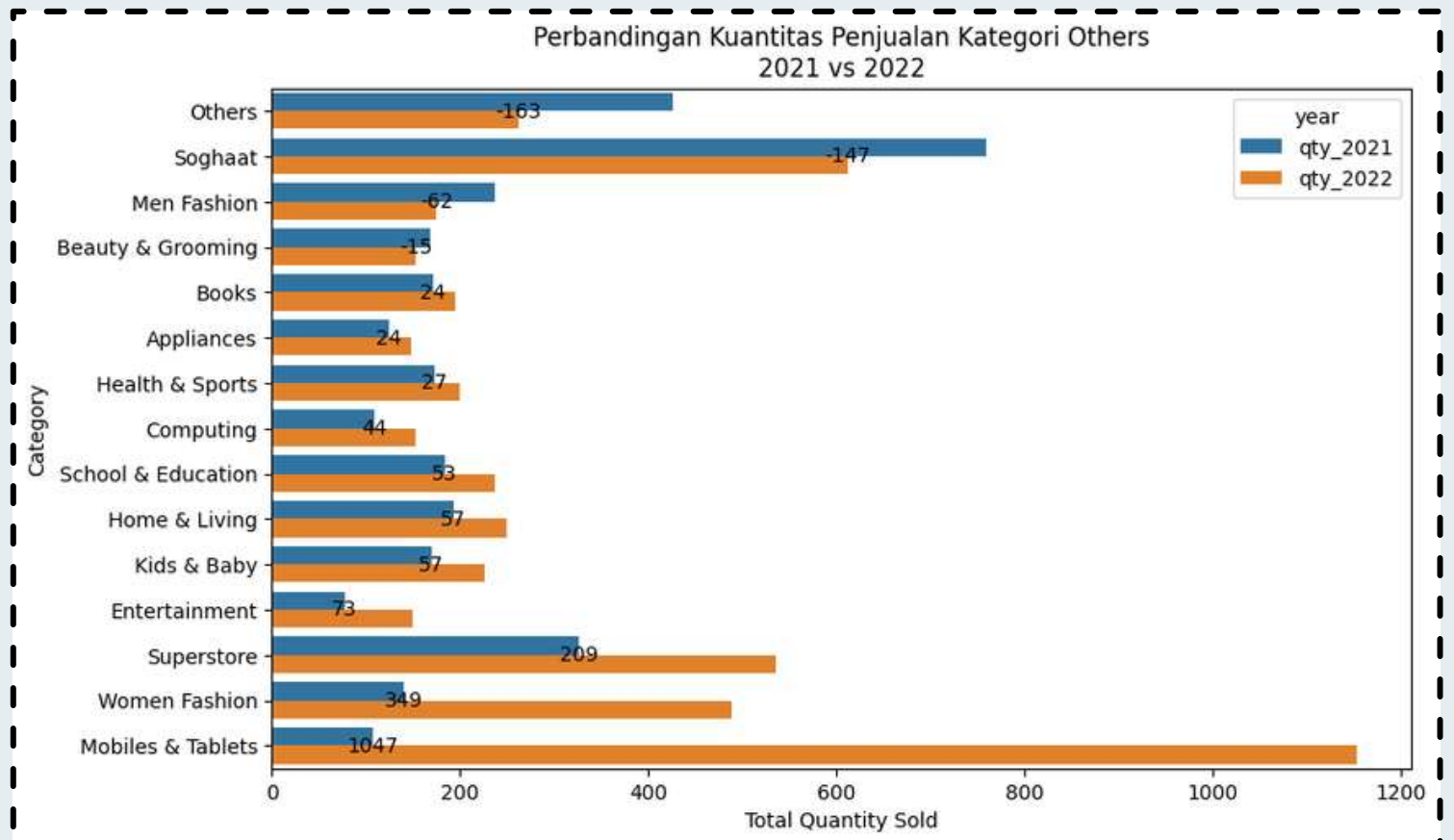
- Kategori "Others" & "Soghaat": Dua kategori ini adalah penyumbang penurunan terbesar. mengindikasikan adanya masalah pada stok yang tidak laku atau hilangnya minat pasar secara mendadak.
- Kategori Fashion & Beauty: Men Fashion (-62) dan Beauty & Grooming (-15) juga mengalami penurunan



TOP PERFORMERS

Terjadi lonjakan yang signifikan pada kategori Superstore, Women Fashion, dan Mobiles & Tablets. Membuktikan bahwa customer di tahun 2022 sangat meminati teknologi terbaru dan kebutuhan gaya hidup.

	category	qty_ordered_2021	qty_ordered_2022	selisih_qty
10	Others	426	263	-163
12	Soghaat	759	612	-147
8	Men Fashion	237	175	-62
1	Beauty & Grooming	168	153	-15
2	Books	171	195	24
0	Appliances	124	148	24
5	Health & Sports	173	200	27
3	Computing	109	153	44
11	School & Education	184	237	53
6	Home & Living	193	250	57
7	Kids & Baby	170	227	57
4	Entertainment	77	150	73
13	Superstore	327	536	209
14	Women Fashion	140	489	349
9	Mobiles & Tablets	107	1154	1047



Result Insight

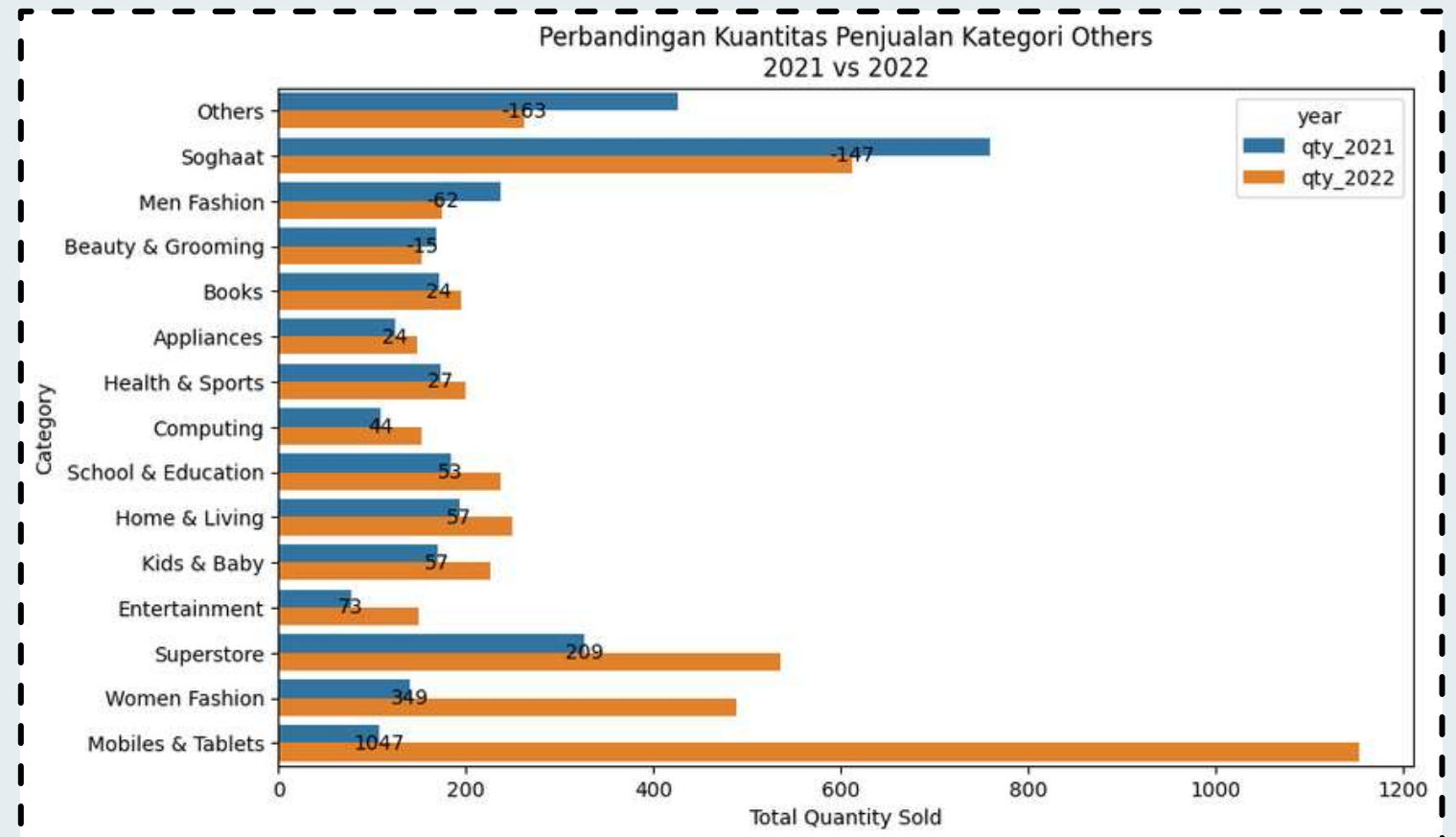
+ KATEGORI STABIL

- Produk pada kategori Books, Appliances, serta School & Education menunjukkan performa yang stabil karena sifatnya yang fungsional dan berbasis pada kebutuhan primer serta siklus rutin tahunan, sehingga tidak terlalu terpengaruh oleh tren pasar.
- Kategori Health & Sports dan Computing terus menunjukkan pertumbuhan stabil yang didorong oleh loyalitas pelanggan.

L PERBANDINGAN TREN

Selisih penjualan kategori Women Fashion dan Men Fashion berbanding terbalik. menunjukkan adanya pergeseran target audiens atau efektivitas promosi yang timpang antara segmen pria dan wanita.

	category	qty_ordered_2021	qty_ordered_2022	selisih_qty
10	Others	426	263	-163
12	Soghaat	759	612	-147
8	Men Fashion	237	175	-62
1	Beauty & Grooming	168	153	-15
2	Books	171	195	24
0	Appliances	124	148	24
5	Health & Sports	173	200	27
3	Computing	109	153	44
11	School & Education	184	237	53
6	Home & Living	193	250	57
7	Kids & Baby	170	227	57
4	Entertainment	77	150	73
13	Superstore	327	536	209
14	Women Fashion	140	489	349
9	Mobiles & Tablets	107	1154	1047



Recommendations



Penyesuaian Inventory Tahun 2023

- Likuidasi stok Others dan Soghaat dengan melakukan clearance sale, diskon,
- Mengurangi kuota pengadaan barang untuk kategori yang mengalami penurunan paling tinggi



Maksimalkan Momentun Pertumbuhan

- Pastikan ketersediaan stok selalu terjaga agar tidak terjadi kehilangan potensi penjualan (lost sales)
- Mengadakan program membership untuk kategori dengan pertumbuhan penjualan tertinggi



Evaluasi Strategi Pemasaran

- Lakukan pengujian iklan secara terpisah untuk segmen pria dan wanita guna memahami pesan mana yang lebih efektif mengonversi penjualan, sehingga anggaran promosi tidak terbuang sia-sia pada segmen yang kurang responsif.

Conclusion

Prioritaskan pengadaan pada kategori berkontribusi besar dengan restock lebih sering dan stok aman lebih tinggi untuk produk inti yang cepat habis, sambil tetap menjaga rotasi dan menghindari penumpukan varian spekulatif. Untuk kategori yang bertumbuh, lakukan penambahan stok bertahap berbasis tren dengan kombinasi produk inti dan musiman agar tetap adaptif terhadap perubahan permintaan. Untuk kategori yang stabil bervolume rendah dan kategori yang menurun, terapkan pendekatan konservatif dan defensif melalui stok kecil-ketat, pembelian berbasis permintaan aktual, serta percepatan clearance atau promo untuk barang lama guna menekan risiko overstock dan barang usang.

CASE 2.2

Menyediakan TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 dibanding dengan 2021, apabila terjadi penurunan kuantitas penjualan pada kategori Others.

SYNTAX CODE:

```
# Filter data Others, valid = 1, tahun 2021 & 2022
df_others = df_valid[
    (df_valid['is_valid'] == 1) &
    (df_valid['category'] == 'Others') &
    (df_valid['order_date'].dt.year.isin([2021, 2022]))
]

# Grouping qty per SKU & tahun
grouped = (
    df_others
    .groupby(['sku_name', df_others['order_date'].dt.year])['qty_ordered']
    .sum()
    .unstack(fill_value=0)
    .reset_index()
)

# Rename kolom
grouped.columns = ['sku_name', 'qty_ordered_21', 'qty_ordered_22']

# Hitung selisih
grouped['selisih_qty_others'] = grouped['qty_ordered_22'] - grouped['qty_ordered_21']

# Sort & ambil top 20 penurunan
compare_produk_others = grouped.sort_values(by='selisih_qty_others', ascending=True)

compare_produk_others.head(20)
```

Filter Data

Menyaring transaksi yang valid, kategori Others, dan hanya tahun 2021–2022

Groupby SKU dan tahun

Mengelompokkan data berdasarkan nama produk (SKU) dan tahun untuk melihat total penjualan tiap tahun

SUM Quantity

Menjumlahkan total quantity (qty_ordered) per SKU per tahun

Unstack

Mengubah format data dari vertikal menjadi horizontal sehingga tahun 2021 dan 2022 menjadi kolom terpisah

Rename Kolom

Mengganti nama kolom agar lebih mudah dibaca (qty_ordered_21, qty_ordered_22)

Hitung Selisih

Menghitung perubahan penjualan dengan rumus: 2022 – 2021

Sorting

Mengurutkan data dari penurunan terbesar ke terkecil

Top 20

Menampilkan 20 produk dengan penurunan penjualan terbesar

Result Insight



Penurunan Tinggi

Penurunan Tertinggi berada pada produk RB_Dettol Germ Busting Kit-bf mengalami penurunan yang sangat drastis, yaitu berkurang sebesar 155 unit, yakni dari 200 unit di 2021 menjadi hanya 45 unit di 2022.



Dead Stock

Dari 20 produk teratas yang mengalami penurunan, terdapat 16 produk yang sama sekali tidak terjual (0 penjualan) pada tahun 2022. Beberapa di antaranya adalah:

- Dawlance_MD 10 + DWB 600
- iu_Tickets General Enclosure-Islamabad
- RS_Rehmat-e-Shereen Mix Mithai
- sindbad_Sindbad Gift Card-3
- Vouch 365 2016
- Am-PTV_ATS-004-M
- duma_4561253300294
- sockoye_QG in Quarter Grey
- The Vitamin Company Kojic Acid Whitening Cream
- aw_Octane Booster-12oz./354ml
- Trans2_LW 999
- aw_Super Concentrated Fuel Injector (Black)-5...
- sstop_Universallensclipkit
- Transcend_64GB JetFlash 810
- JBS_IFAM-009

	sku_name	qty_ordered_21	qty_ordered_22	selisih_qty_others
69	RB_Dettol Germ Busting Kit-bf	200	45	-155
18	Dawlance_MD 10 + DWB 600	23	0	-23
84	Telemall_MM-DR-HB-L	23	2	-21
137	iu_Tickets General Enclosure-Islamabad	20	0	-20
70	RS_Rehmat-e-Shereen Mix Mithai	13	0	-13
141	kansai_NeverWet	10	1	-9
154	sindbad_Sindbad Gift Card-3	7	0	-7
133	emart_00-1	7	1	-6
93	Vouch 365 2016	5	0	-5
4	Am-PTV_ATS-004-M	5	0	-5
129	duma_4561253300294	4	0	-4
155	sockoye_QG in Quarter Grey	4	0	-4
85	The Vitamin Company Kojic Acid Whitening Cream...	4	0	-4
115	aw_Octane Booster-12oz./354ml	3	0	-3
88	Trans2_LW 999	3	0	-3
43	MEGUIAR_G12711	4	1	-3
119	aw_Super Concentrated Fuel Injector (Black)-5....	2	0	-2
156	sstop_Universallensclipkit	2	0	-2
89	Transcend_64GB JetFlash 810	2	0	-2
26	JBS_IFAM-009	2	0	-2

Result Insight

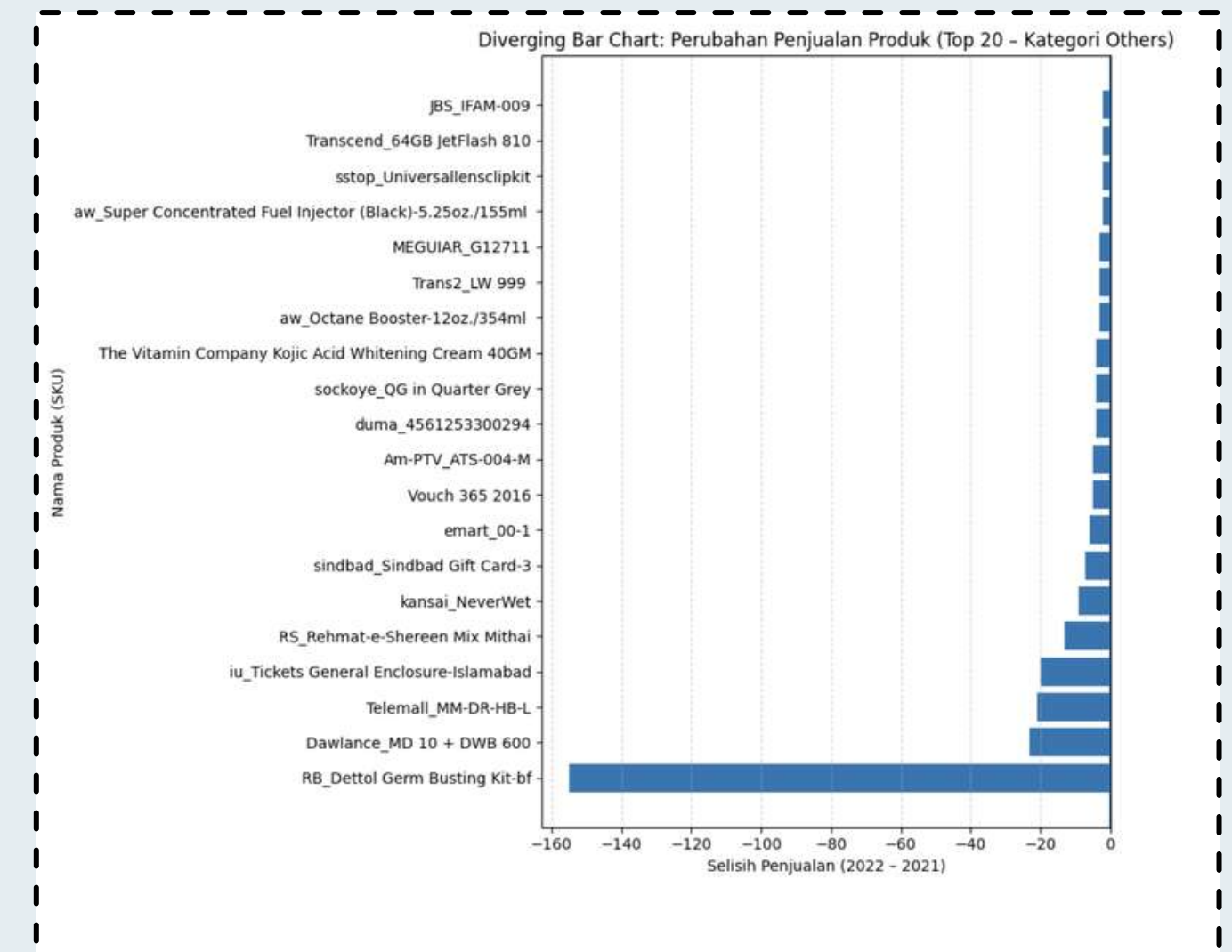
+ Penurunan Tinggi

Penurunan Tertinggi berada pada produk RB_Dettol Germ Busting Kit-bf mengalami penurunan yang sangat drastis, yaitu berkurang sebesar 155 unit, yakni dari 200 unit di 2021 menjadi hanya 45 unit di 2022.

L Dead Stock

Dari 20 produk teratas yang mengalami penurunan, terdapat 16 produk yang sama sekali tidak terjual (0 penjualan) pada tahun 2022. Beberapa di antaranya adalah:

- Dawlance_MD 10 + DWB 600
- iu_Tickets General Enclosure-Islamabad
- RS_Rehmat-e-Shereen Mix Mithai
- sindbad_Sindbad Gift Card-3
- Vouch 365 2016
- Am-PTV_ATS-004-M
- duma_4561253300294
- sockoye_QG in Quarter Grey
- The Vitamin Company Kojic Acid Whitening Cream
- aw_Octane Booster-12oz./354ml
- Trans2_LW 999
- aw_Super Concentrated Fuel Injector (Black)-5...
- sstop_Universallensclipkit
- Transcend_64GB JetFlash 810
- JBS_IFAM-009



Recommendations



Strategi Likuidasi

Lakukan penghabisan sisa stok produk melalui program seperti Flash Sale, diskon besar, atau bundling dengan produk populer. Fokus utama strategi ini adalah mengosongkan gudang dan mengembalikan cash flow, daripada mencari margin keuntungan.



Penghentian Restock

Lakukan penghentian pembelian baru untuk seluruh SKU yang masuk dalam daftar 20 produk dengan penurunan terbesar. Berdasarkan data yang ada, penambahan stok baru berpotensi memperburuk beban biaya operasional gudang, dengan pertimbangan tidak ada stok yang berkurang.



Audit produk

Lakukan pengecekan fisik segera untuk memisahkan produk yang sudah tidak layak jual (seperti produk kadaluwarsa atau voucher tahun lama yang sudah tidak berlaku). Produk yang sudah tidak punya nilai jual harus segera dihapus dari sistem (write-off) agar laporan inventaris perusahaan menunjukkan nilai aset yang akurat.

Conclusion

Berdasarkan analisis data, ditemukan bahwa penurunan kuantitas penjualan kategori Others pada tahun 2022 sangat signifikan, di mana 80% dari TOP 20 produk yang mengalami penurunan terdalam mencatatkan nol penjualan (dead stock), dengan penurunan tertinggi pada produk RB_Dettol Germ Busting Kit-bf sebesar 155 unit. Untuk menanggulangi penumpukan stok tersebut, tim direkomendasikan untuk segera melakukan likuidasi melalui program diskon besar atau bundling, menghentikan seluruh rencana pengadaan ulang (restock) untuk SKU terdampak, serta melakukan audit fisik dan write-off terhadap produk yang sudah kadaluwarsa atau tidak relevan guna menjaga akurasi nilai aset dan efisiensi ruang gudang.

Case 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 Marketing

CASE 3

Customer check-out namun belum melakukan pembayaran selama tahun 2022. Data yang dibutuhkan ID Customer dan Registered Date

SYNTAX CODE:

```
#Filter data sesuai kriteria
df_tokopaedi = (
    df[
        (df['is_gross'] == 1) &
        (df['is_valid'] == 0) &
        (df['is_net'] == 0) &
        (df['order_date'].dt.year == 2022)
    ]
)

#Ambil data customer_id dan registered_date
customer_checkout = (
    df_tokopaedi[['customer_id', 'registered_date']]
    .drop_duplicates()
    .sort_values(by='registered_date')
    .reset_index(drop=True)
)

#Menampilkan data
customer_checkout
```

Filter Data Sesuai Kriteria

- `df_tokopaedi = (...)` → menyimpan hasil filter data ke dalam DataFrame bernama tokopaedi
- `df[...]` → melakukan filter baris pada DataFrame df
- `(df['is_gross'] == 1)` → memilih transaksi yang sudah check-out
- `(df['is_valid'] == 0)` → memilih transaksi yang belum melakukan pembayaran (belum valid)
- `(df['is_net'] == 0)` → memilih transaksi yang belum menjadi transaksi bersih (belum dibayar)
- `& (AND operator)` → menggabungkan seluruh kondisi
- `(df['order_date'].dt.year == 2022)` → memilih transaksi yang terjadi pada tahun 2022

Ambil Data customer_id dan registered_date

- `customer_checkout = (...)` → menyimpan hasil akhir ke DataFrame customer_checkout
- `df_tokopaedi[['customer_id', 'registered_date']]` → mengambil kolom customer_id dan registered_date dari DataFrame saja
- `.drop_duplicates()` → menghapus data customer yang duplikat, sehingga setiap customer hanya muncul sekali
- `.sort_values(by='registered_date')` → mengurutkan data berdasarkan tanggal registrasi
- `.reset_index(drop=True)` → mengatur ulang index agar rapi berurutan dan menghapus index lama

CASE 3

Customer check-out namun belum melakukan pembayaran selama tahun 2022. Data yang dibutuhkan ID Customer dan Registered Date

SYNTAX CODE:

```
#Trend pelanggan yang hanya melakukan checkout per bulan (2022)
monthly_trend = (
    df_tokopaedi
    .groupby(df_tokopaedi['order_date'].dt.to_period('M'))['customer_id']
    .nunique()
    .reset_index()
    .rename(columns={'customer_id': 'Total_cust', 'order_date': 'Bulan'})
)

print('Tren pelanggan yang hanya checkout per bulan 2022:')
print(monthly_trend)
```

Tren Pelanggan Yang Hanya Melakukan Check-out Per Bulan (2022)

- `monthly_trend = (...)` → membuat variabel `monthly_trend` untuk menyimpan hasil analisis tren bulanan
- `df_tokopaedi` → data sumber, berisi pelanggan yang sudah check-out tetapi belum membayar
- `.groupby(df_tokopaedi['order_date'].dt.to_period('M'))['customer_id']` → mengelompokkan data berdasarkan bulan (Year-Month) dari `order_date`
- `.nunique()` → menghitung jumlah customer unik tiap bulan
- `.reset_index()` → mengubah hasil `groupby` menjadi DataFrame biasa
- `.rename(columns={'customer_id': 'Total_cust', 'order_date': 'Bulan'})` → mengganti nama kolom agar lebih deskriptif
- `print('Tren pelanggan yang hanya checkout per bulan 2022:')` → menampilkan judul output
- `print(monthly_trend)` → menampilkan tabel tren pelanggan bulanan

CASE 3

Customer check-out namun belum melakukan pembayaran selama tahun 2022. Data yang dibutuhkan ID Customer dan Registered Date

SYNTAX CODE CHART:

```
import matplotlib.pyplot as plt

plt.figure()
plt.plot(monthly_trend['Bulan'].astype(str),
         monthly_trend['Total_cust'],
         marker='o')
plt.xticks(rotation=45)
plt.title('Tren Customer Checkout Tanpa Bayar - 2022')
plt.xlabel('Bulan')
plt.ylabel('Jumlah Customer')
plt.tight_layout()
plt.show()
```

Visualisasi Grafik

- `import matplotlib.pyplot as plt` → mengimpor pustaka Matplotlib untuk membuat visualisasi grafik
- `plt.figure()` → membuat canvas/figure baru untuk grafik
- `plt.plot(...)` → membuat grafik garis
- `monthly_trend['Bulan'].astype(str)` → sumbu X (bulan, diubah ke string agar terbaca jelas)
- `monthly_trend['Total_cust']` → sumbu Y (jumlah customer)
- `marker='o'` → menampilkan titik pada setiap bulan
- `plt.xticks(rotation=45)` → memutar label bulan 45° agar tidak saling bertumpuk
- `plt.title('Tren Customer Checkout Tanpa Bayar - 2022')` → memberi judul grafik
- `plt.xlabel('Bulan')` → memberi label sumbu X
- `plt.ylabel('Jumlah Customer')` → memberi label sumbu Y
- `plt.tight_layout()` → mengatur tata letak agar elemen grafik tidak saling terpotong
- `plt.show()` → menampilkan grafik ke layar

Result **Insight**



High Potential Customers – Konversi Gagal

Niat beli tinggi di awal tahun, tetapi banyak check-out gagal bayar akibat hambatan di tahap payment.



Penurunan Niat Beli & Lost Revenu

Sejak pertengahan tahun niat beli turun tajam, memicu banyak transaksi gagal dan hilangnya potensi revenue serta repeat buyer.



High Season & Strategi Retensi Belum Optimal

Momentum high season belum dimanfaatkan maksimal, padahal registered date dapat digunakan untuk segmentasi dan promo yang lebih tepat sasaran.

	customer_id	registered_date
0	C849967L	2021-07-02
1	C349980L	2021-07-02
2	C831683L	2021-07-03
3	C893754L	2021-07-03
4	C378684L	2021-07-03
...
815	C877563L	2022-08-24
816	C741725L	2022-08-25
817	C788348L	2022-08-25
818	C466867L	2022-08-28
819	C925840L	2022-08-28
820 rows × 2 columns		

Result Insight



High Potential Customers – Konversi Gagal

Niat beli tinggi di awal tahun, tetapi banyak check-out gagal bayar akibat hambatan di tahap payment.



Penurunan Niat Beli & Lost Revenu

Sejak pertengahan tahun niat beli turun tajam, memicu banyak transaksi gagal dan hilangnya potensi revenue serta repeat buyer.

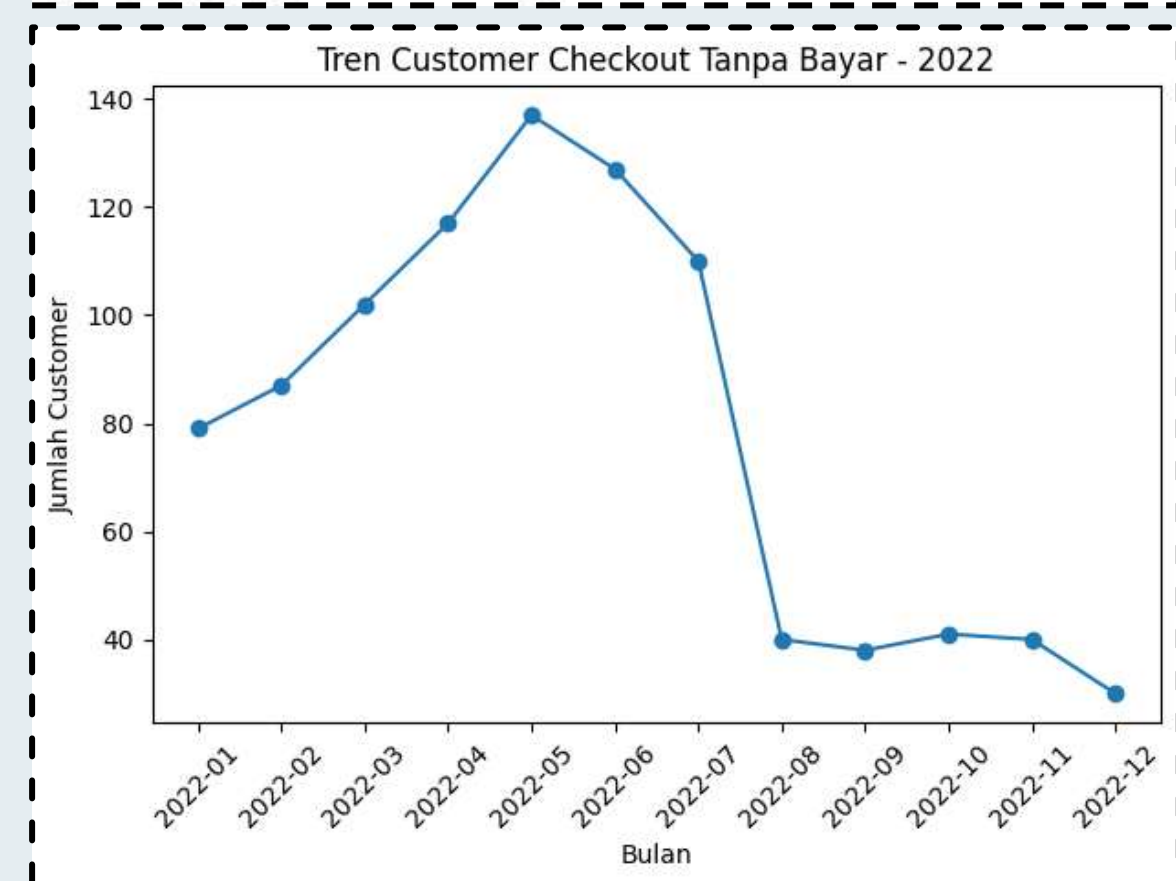


High Season & Strategi Retensi Belum Optimal

Momentum high season belum dimanfaatkan maksimal, padahal registered date dapat digunakan untuk segmentasi dan promo yang lebih tepat sasaran.

Tren pelanggan yang hanya checkout per bulan 2022:

	Bulan	Total_cust
0	2022-01	79
1	2022-02	87
2	2022-03	102
3	2022-04	117
4	2022-05	137
5	2022-06	127
6	2022-07	110
7	2022-08	40
8	2022-09	38
9	2022-10	41
10	2022-11	40
11	2022-12	30



Recommendations



Strategi Promo Terfokus

Fokuskan promo di periode kritis dengan voucher, urgensi waktu, dan segmentasi personal untuk meningkatkan konversi.



Payment Reminder & Opsi Bayar

Aktifkan reminder otomatis 1x24 jam dan perluas metode pembayaran untuk mendorong penyelesaian transaksi.



Evaluasi Payment Issue

Lakukan investigasi penyebab checkout gagal bayar untuk mengurangi hambatan pembayaran.

Conclusion

Masalah utama terletak pada kegagalan konversi di tahap pembayaran dan penurunan niat beli setelah pertengahan tahun, yang diperparah oleh kurang optimalnya pemanfaatan momentum high season. Dengan strategi promo yang lebih terfokus, reminder pembayaran yang efektif, perluasan metode pembayaran, serta evaluasi menyeluruh atas kendala payment, perusahaan berpeluang besar memulihkan lost revenue dan meningkatkan konversi serta retensi pelanggan.

Case 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

CASE 4.1

Tampilkan rata-rata harian penjualan (before_discount) antara weekend vs weekday per bulan Oktober–Desember 2022 untuk melihat peningkatan tiap bulan

SYNTAX CODE:

```
1 import numpy as np
2
3 # 1. Pastikan order_date bertipe datetime
4 df_od['order_date'] = pd.to_datetime(df_od['order_date'])
5
6 # 2. Filter data valid: Oktober–Desember 2022
7 df_4_1 = df_od
8 [(df_od['order_date'].dt.year == 2022) &
9  (df_od['order_date'].dt.month.isin([10, 11, 12])) &
10 (df_od['is_valid'] == 1)].copy()
11
12 # 3. Tentukan day type (Weekday vs Weekend)
13 df_4_1['day_type'] = df_4_1['order_date'].dt.day_name().apply(
14     lambda x: 'Weekend' if x in ['Saturday', 'Sunday'] else 'Weekday')
15
16 # 4. Tambahkan nama bulan
17 df_4_1['month'] = df_4_1['order_date'].dt.month_name()
18
19 # 5. TOTAL PENJUALAN PER HARI
20 daily_sales =
21 (df_4_1.groupby(['order_date', 'month', 'day_type'], as_index=False)
22  .agg(total_daily_sales=('before_discount', 'sum')))
23
24 # 6. RATA-RATA HARIAN
25 avg_daily_sales =
26 (daily_sales
27  .groupby(['month', 'day_type'], as_index=False)
28  .agg(avg_daily_sales=('total_daily_sales', 'mean'))
29  .round(1))
30
```

Fungsi pada Data Preparation (Oktober–Desember 2022)

- Data pada dataframe **df_od** dikonversi ke format datetime agar dapat dilakukan filter waktu.
- Data difilter hanya untuk transaksi valid (**is_valid = 1**) pada bulan Oktober–Desember 2022.
- Kolom **day_type** dibuat untuk mengelompokkan hari menjadi Weekday dan Weekend berdasarkan order_date.
- Kolom **month** dibuat untuk mengambil nama bulan dari **order_date**.

Fungsi pada daily_sales

- **.groupby()** digunakan untuk mengelompokkan data berdasarkan tanggal, bulan, dan jenis hari (**day_type**).
- **.agg()** digunakan untuk menghitung total penjualan per hari (**before_discount**) sehingga setiap tanggal memiliki satu nilai total penjualan.

Fungsi pada avg_daily_sales

- Data dikelompokkan kembali berdasarkan bulan dan **day_type**.
- Rata-rata dihitung dari total penjualan harian, bukan per transaksi, sehingga hasil lebih representatif.
- **.round(1)** digunakan untuk membulatkan nilai agar mudah dibaca.

CASE 4.1

Tampilkan rata-rata harian penjualan (before_discount) antara weekend vs weekday per bulan Oktober–Desember 2022 untuk melihat peningkatan tiap bulan

SYNTAX CODE:

```
31 # 7. Urutan bulan
32 month_order = ['October', 'November', 'December']
33 avg_daily_sales['month'] = pd.Categorical
34 ( avg_daily_sales['month'],
35   categories=month_order,
36   ordered=True)
37
38 # 8. Pivot tabel (Weekday vs Weekend)
39 final_daily_sales =
40 ( avg_daily_sales
41   .pivot(index='month', columns='day_type', values='avg_daily_sales')
42   .sort_index())
43
44 # 9. Uplift / selisih Weekend vs Weekday
45 final_daily_sales['uplift_weekend'] =
46 (final_daily_sales['Weekend'] - final_daily_sales['Weekday'])
47
48 # 10. Impact
49 final_daily_sales['impact'] = np.where
50 ( final_daily_sales['uplift_weekend'] > 0,
51   'Naik', 'Tidak Naik')
52
53 final_daily_sales
54
```

Fungsi Pengurutan Bulan

- `pd.Categorical()` digunakan untuk mengatur urutan bulan secara kronologis: October → November → December.
- Hal ini mencegah urutan bulan tampil secara alfabet.

Fungsi pada final_daily_sales

- `.pivot()` digunakan untuk mengubah day_type menjadi kolom (Weekday vs Weekend) agar mudah dibandingkan.
- Tabel diurutkan berdasarkan bulan sesuai urutan yang telah ditentukan.

Fungsi Uplift & Impact

Uplift Weekend dihitung sebagai selisih rata-rata penjualan harian:

- Weekend - Weekday

Impact digunakan untuk menilai dampak campaign:

- “Naik” → jika uplift > 0
- “Tidak Naik” → jika uplift ≤ 0

Result Insight



Perbandingan Penjualan Weekday vs Weekend

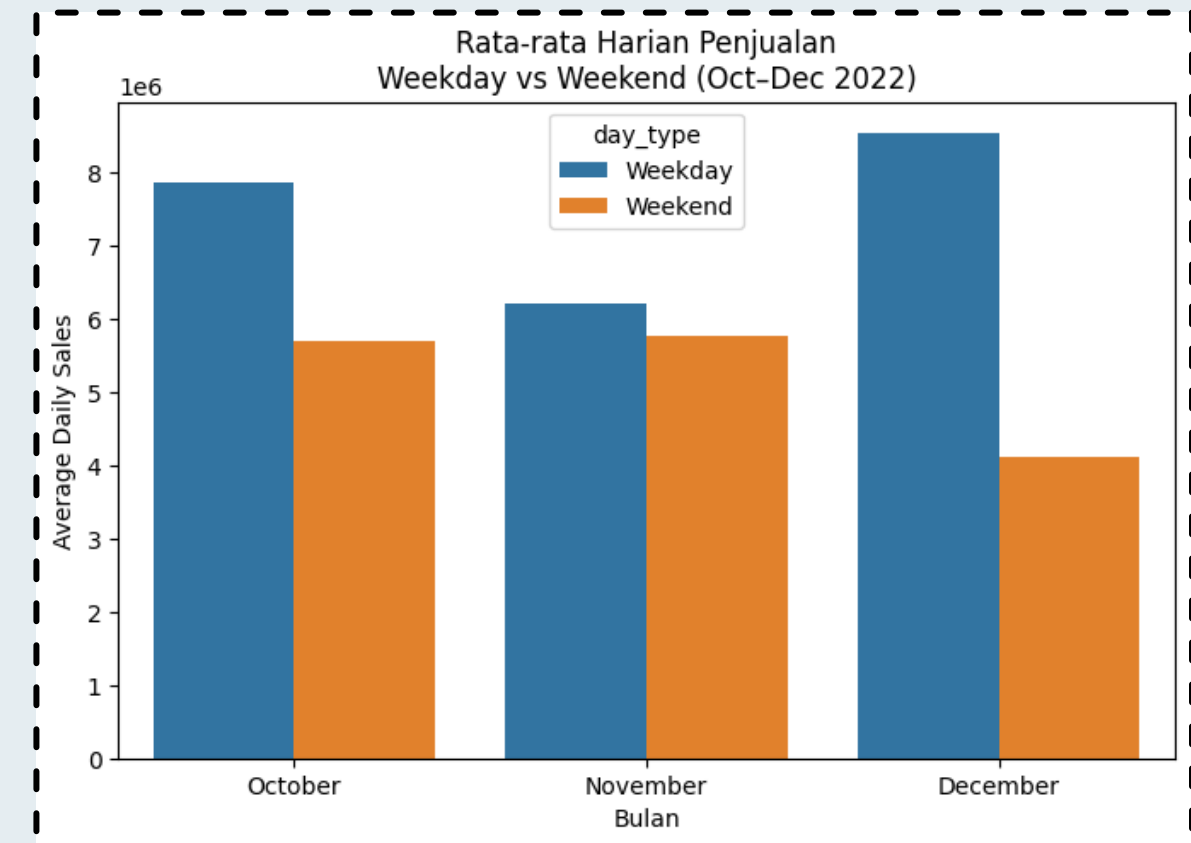
- Pada seluruh bulan (Oktober–Desember 2022), rata-rata penjualan harian weekday selalu lebih tinggi dibandingkan weekend.
- Hal ini menunjukkan bahwa aktivitas transaksi lebih kuat di hari kerja dibandingkan akhir pekan

day_type	Weekday	Weekend	uplift_weekend	impact
month				
October	7872212.4	5708340.7	-2163871.7	Tidak Naik
November	6204666.0	5774045.0	-430621.0	Tidak Naik
December	8542530.0	4105994.0	-4436536.0	Tidak Naik



Dampak Campaign Weekend Terhadap Penjualan

- Nilai uplift weekend bernilai negatif di semua bulan, artinya: Rata-rata penjualan weekend < rata-rata penjualan weekday
- Dengan demikian, campaign yang dijalankan setiap Sabtu dan Minggu belum berhasil meningkatkan penjualan harian.



Bulan Dengan Selisih Terbesar

- Desember menunjukkan selisih (uplift negatif) terbesar antara weekday dan weekend.
- Ini mengindikasikan bahwa meskipun Desember biasanya periode ramai, penjualan tetap lebih terkonsentrasi di weekday.

Recommendations



Optimalisasi Strategi Campaign

- Campaign weekend perlu dievaluasi dari sisi mekanisme promo, seperti besaran diskon, jenis penawaran, maupun syarat dan ketentuan. Uplift yang konsisten negatif menunjukkan bahwa promo yang diberikan belum cukup kuat untuk mendorong peningkatan transaksi di akhir pekan.



Penyesuaian Fokus Waktu dan Target

- Mengingat penjualan lebih tinggi terjadi pada weekday, perusahaan dapat mempertimbangkan untuk memperkuat campaign di hari kerja atau mengombinasikan strategi weekday-weekend. Selain itu, campaign weekend sebaiknya lebih tersegmentasi agar lebih relevan bagi target pelanggan tertentu.



Pendalaman Analisis Perilaku dan Musiman

- Perbedaan penjualan yang paling besar terjadi di bulan Desember, sehingga diperlukan analisis lanjutan terhadap faktor musiman, perilaku konsumen, serta kondisi eksternal (libur atau kompetisi promo) yang dapat memengaruhi efektivitas campaign.

Conclusion

Berdasarkan analisis rata-rata penjualan harian periode Oktober–Desember 2022, dapat disimpulkan bahwa campaign yang dijalankan pada akhir pekan (Sabtu–Minggu) belum memberikan dampak positif terhadap peningkatan penjualan. Pada seluruh bulan, rata-rata penjualan weekday tetap lebih tinggi dibandingkan weekend, dengan uplift weekend yang konsisten bernilai negatif. Oleh karena itu, strategi campaign weekend perlu dievaluasi dan disesuaikan agar dapat memberikan kontribusi yang lebih optimal terhadap peningkatan penjualan ke depannya.

CASE 4.2

Rata-rata harian penjualan (before_discount)
weekend vs weekday secara keseluruhan selama
Oktober–Desember 2022

SYNTAX CODE:

```
1 total_sales_weekend = weekend['before_discount'].sum()
2 unique_days_weekend = weekend['order_date'].nunique()
3 avg_daily_sales_weekend_overall = total_sales_weekend / unique_days_weekend
4
5 total_sales_weekday = weekday['before_discount'].sum()
6 unique_days_weekday = weekday['order_date'].nunique()
7 avg_daily_sales_weekday_overall = total_sales_weekday / unique_days_weekday
8
9 print(f"Overall Average Daily Sales for Weekends (Oct-Dec 2022): {avg_daily_sales_weekend_overall:,.2f}")
10 print(f"Overall Average Daily Sales for Weekdays (Oct-Dec 2022): {avg_daily_sales_weekday_overall:,.2f}")
11
12 summary_overall_3months = pd.DataFrame({
13     'day_type': ['Weekday', 'Weekend'],
14     'avg_daily_sales': [avg_daily_sales_weekday_overall, avg_daily_sales_weekend_overall]
15 })
16
17 plt.figure(figsize=(8, 5))
18 sns.barplot(x='day_type', y='avg_daily_sales', data=summary_overall_3months, palette={'Weekday': 'lightcoral', 'Weekend': 'skyblue'})
19 plt.title('Rata-rata Harian Penjualan Keseluruhan (Okt–Des 2022)')
20 plt.ylabel('Rata-rata Penjualan Harian')
21 plt.xlabel('')
22 plt.xticks(rotation=0)
23 plt.grid(axis='y', linestyle='--', alpha=0.7)
24 plt.tight_layout()
25 plt.show()
```

Rata-rata Penjualan Harian

- **total_sales_weekend** → digunakan untuk menghitung total penjualan sebelum diskon pada akhir pekan.
- **unique_days_weekend** → menghitung jumlah hari transaksi unik pada akhir pekan.
- **avg_daily_sales_weekend_overall** → menghitung rata-rata penjualan harian akhir pekan dari total penjualan dibagi jumlah hari
- **total_sales_weekday** → menghitung total penjualan sebelum diskon pada hari kerja.
- **unique_days_weekday** → menghitung jumlah hari transaksi unik pada hari kerja.
- **avg_daily_sales_weekday_overall** → menghitung rata-rata penjualan harian hari kerja.

Ringkasan & Visualisasi

- **summary_overall_3months** → membuat DataFrame ringkasan rata-rata penjualan harian Weekday dan Weekend.
- **sns.barplot()** → menampilkan perbandingan rata-rata penjualan harian dalam bentuk grafik batang.
- **plt.show()** → menampilkan hasil visualisasi grafik.

Result Insight



Pendapatan Weekday Lebih Tinggi Dibandingkan Weekend

Rata-rata pendapatan harian di weekday sekitar 7,52 juta, sedangkan di weekend hanya sekitar 5,27 juta. Ini menunjukkan bahwa aktivitas utama masih terjadi pada weekday, sehingga weekday menjadi penopang utama pendapatan. Bisnis masih belum seimbang antara weekday dan weekend.



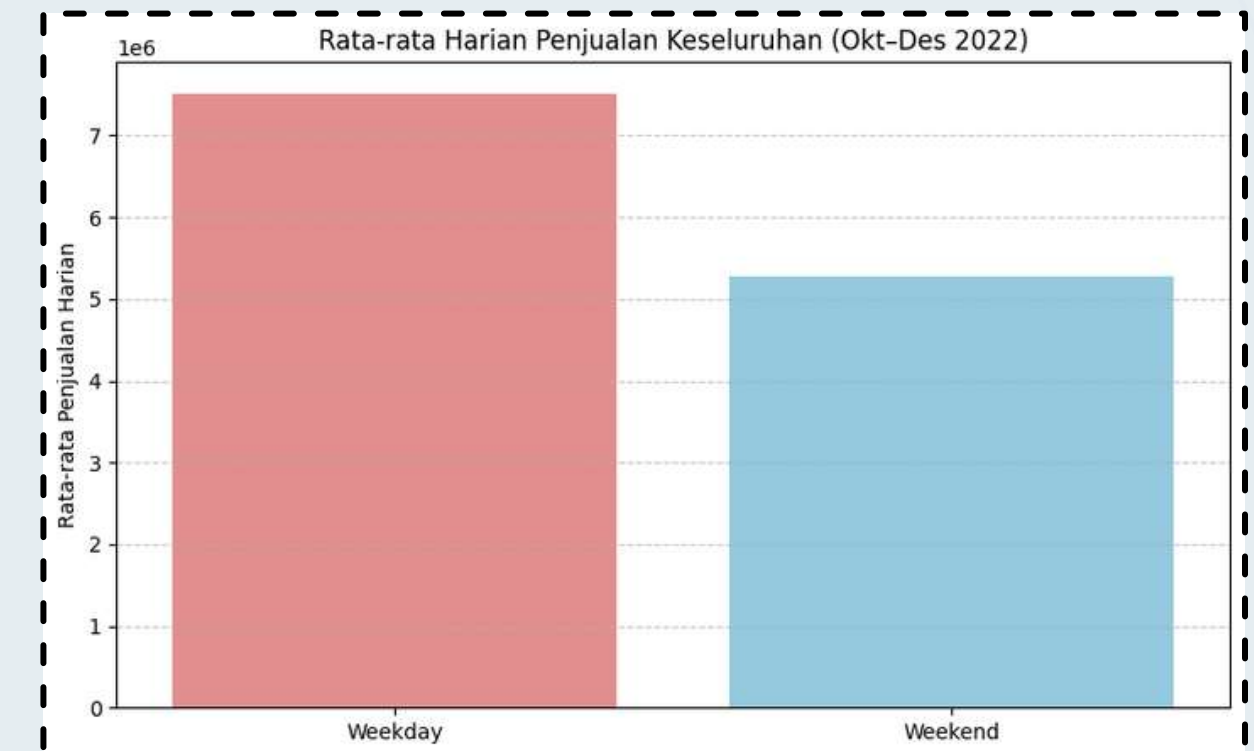
Potensi Weekend Belum Dimaksimalkan

Selisih pendapatan yang cukup besar antara weekday dan weekend menandakan bahwa weekend bukan tidak diminati, tetapi belum didorong dengan strategi yang tepat. Padahal, di weekend orang cenderung memiliki waktu luang lebih banyak.



Perilaku Konsumen Berbeda Antara Weekday dan Weekend

Di weekday, konsumen biasanya bertransaksi karena aktivitas operasional, kebutuhan rutin, atau keperluan mendesak. Sementara di weekend, keputusan membeli lebih dipengaruhi oleh promo, suasana santai, dan dorongan emosional.



Overall Average Daily Sales for Weekends (Oct-Dec 2022): 5,269,300.00
Overall Average Daily Sales for Weekdays (Oct-Dec 2022): 7,520,249.29



Rata-Rata Penjualan Harian Periode Okt-Des 2022

- Rata-rata harian Weekday: Rp 7.520.249
- Rata-rata harian Weekend: Rp 5.269.300

Recommendations



Meningkatkan Penjualan Weekend

Memanfaatkan weekend untuk meningkatkan penjualan, karena orang biasanya punya lebih banyak waktu luang dan memastikan stok, layanan, dan operasional tetap optimal seperti saat weekday.



Menjalankan Strategi Promosi

Mengadakan promo khusus weekend, seperti flashsale, bundling, atau cashback untuk mendorong pembelian dengan menggunakan bahasa yang ringan, mudah dimengerti, dan menarik.



Menarik Pelanggan Baru

Mengenalkan produk dan membangun ketertarikan dengan mengadakan trial atau bonus pembelian pertama yang hanya berlaku di weekend khusus pelanggan baru.

Conclusion

Berdasarkan analisis rata-rata pendapatan harian periode Oktober–Desember 2022, kinerja weekday terbukti lebih kuat dan menjadi sumber utama pendapatan, sementara kinerja weekend masih tertinggal cukup jauh yang menunjukkan potensi weekend belum dimanfaatkan secara optimal.

Weekend memiliki peluang besar untuk mendorong pertumbuhan karena konsumen memiliki lebih banyak waktu luang, namun membutuhkan dorongan berupa promo, komunikasi yang menarik, dan pendekatan yang lebih santai. Oleh karena itu, strategi yang paling tepat adalah mengembangkan strategi khusus untuk meningkatkan transaksi di weekend dan menjaga kestabilan performa weekday, sehingga secara bertahap bisnis dapat mencapai pertumbuhan yang lebih seimbang, mengurangi ketergantungan pada weekday, dan meningkatkan total pendapatan secara keseluruhan.

Presented by Group C6



Thank You

*We appreciate your partnership and trust. Let's continue
creating impact together.*

