

2025

FINAL PROJECT - Python

E-Commerce Python Performance Insights

#RintisKarirImpian

#MySkill

Profil Analyst

ROZALINDA TITALIA PUTRI



Surabaya, Jawa Timur



lindaroza509@gmail.com



Rozalinda Titalia Putri

Saya adalah mahasiswa sistem informasi semester 3 yang memiliki semangat tinggi dibidang IT dan Data Analyst. Sebagai calon Data Analyst, saya sangat antusias untuk menerapkan kemampuan analisis dan pengetahuan saya dalam menyelesaikan permasalahan terutama dibidang data.



#MySkill

List of Contents

Overview

Issue

Result & Interpretation

Overview

Data yang digunakan berasal dari Tokopedia (bukan data sesungguhnya). Dataset yang digunakan terdiri dari:

1. **order_detail**
2. **sku_detail**
3. **customer_detail**
4. **payment_detail**

Penjelasan dataset adalah sebagai berikut:

- **order_detail** : berisi informasi detail dari setiap transaksi
- **customer_detail** : berisi informasi pelanggan/customer
- **sku_detail** : berisi informasi detail terkait produk
- **payment_detail** : berisi daftar metode pembayaran

order_detail	
id	angka unik dari order/order_id
customer_id	angka unik dari pelanggan
order_date	tanggal saat dilakukan transaksi
sku_id	angka unik dari produk (SKU : 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 (price * 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

Overview

#MySkill

sku_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 if Goods Sold : total biaya untuk menjual 1 produk
catgeory	kategori produk

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_methode	metode pembayaran yang digunakan

```
#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)
```

Mengambil file dari link Github, kemudian membaca file yang ada pada Github dalam bentuk file csv.
juga memberi nama alias

```
df_od.head()  
df_pd.head()  
df_cd.head()  
df_sd.head()
```

Menampilkan 5 data teratas yang ada pada order_detail, payment_detail, customer_detail, sku_deatil



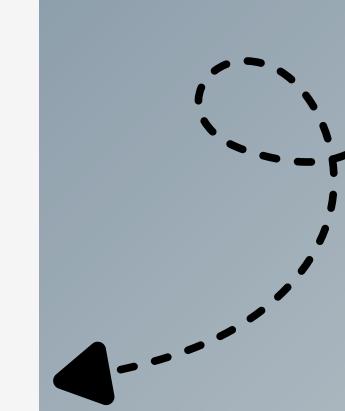
```
[ ] #Menjalankan SQL di Colab
from sqlite3 import connect
conn = connect(':memory:')
df_od.to_sql('order_detail', conn, index=False, if_exists='replace')
df_pd.to_sql('payment_detail', conn, index=False, if_exists='replace')
df_sd.to_sql('sku_detail', conn, index=False, if_exists='replace')
df_cd.to_sql('customer_detail',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



Membuat database SQLite sementara pada Google Colab, kemudian keempat dataset akan disimpan dalam bentuk tabel SQL. Hal tersebut diperlukan untuk mempermudah analisa data menggunakan query SQL pada Google Colab

```
▶ #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)
```

Tampilan query SQL yang digunakan pada Goole Colab, query tersebut digunakan untuk menggabungkan keempat dataset menjadi satu dengan nama data frame read sql



▶ #Menampilkan tipe data tiap kolom
df.dtypes

Menampilkan tipe data yang ada pada tiap kolom

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

Mengubah tipe data yang ada pada read_sql untuk mempermudah pengolahan data

▶ #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

Mengubah tipe data yang ada pada read_sql untuk mempermudah pengolahan data





Question 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

```

# Memfilter data
filtered = df[
    (df['is_valid'] == 1) &
    (df['category'] == 'Mobiles & Tablets') &
    (df['order_date'].dt.year == 2022)]

# Grouping 'filtered' berdasarkan sku_name
# Hitung jumlah kuantitas penjualan (qty_ordered)
# Urutkan berdasarkan sum qty_ordered DESC

grouped = (
    filtered.groupby('sku_name')['qty_ordered']
    .sum()
    .reset_index()
    .sort_values(by='qty_ordered', ascending=False)
    .rename(columns={
        'sku_name': 'produk',
        'qty_ordered': 'kuantitas'
    }))
# Menampilkan 5 top product
top5_product = grouped.head()

```

Filter data:

Membuat variabel '**filtered**' dengan values:

- filter data dengan 'is_valid' adalah 1,
- filter data dengan 'category' adalah Mobiles & Tablets
- Filter data 'order_date' berdasarkan tahun menggunakan fungsi **.dt.year** adalah tahun 2022

Grouped data:

Membuat variabel '**grouped**' dengan value yang berisi data '**filtered**' setelah diberikan fungsi **.groupby**

- **.groupby** digunakan untuk mengelompokkan data berdasarkan **sku_name** kemudian **qty_ordered**
- **.sum()** digunakan untuk menjumlah kolom **'qty_ordered'**

```
grouped = (
    filtered.groupby('sku_name')['qty_ordered']
    .sum()
    .reset_index()
    .sort_values(by='qty_ordered', ascending=False)
    .rename(columns={
        'sku_name': 'produk',
        'qty_ordered': 'kuantitas'
    }))
```

```
# Menampilkan 5 top product
top5_product = grouped.head()
```

```
print(top5_product)
```

- saat menggunakan fungsi **.groupby()**, kolom yang dipakai untuk mengelompokkan (**sku_name**) akan menjadi **index**, bukan kolom biasa, sehingga **.reset_index()** akan mengubah **index** tersebut kembali menjadi **kolom biasa**.
- mengurutkan data menggunakan fungsi **.sort_values** berdasarkan '**qty_ordered**', dengan urutan dari nilai terbesar ke nilai terkecil (**ascending=False**)
- mengganti nama kolom output dengan fungsi **.rename**, yaitu mengganti nama kolom '**sku_name**' menjadi '**produk**' dan '**qty_ordered**' menjadi '**kuantitas**'. Rename ini hanya berlaku di data '**grouped**' bukan data frame

Top 5 product:

- membuat variabel '**top5_product**' dengan value yang menampilkan 5 produk teratas

Menampilkan top 5 product:

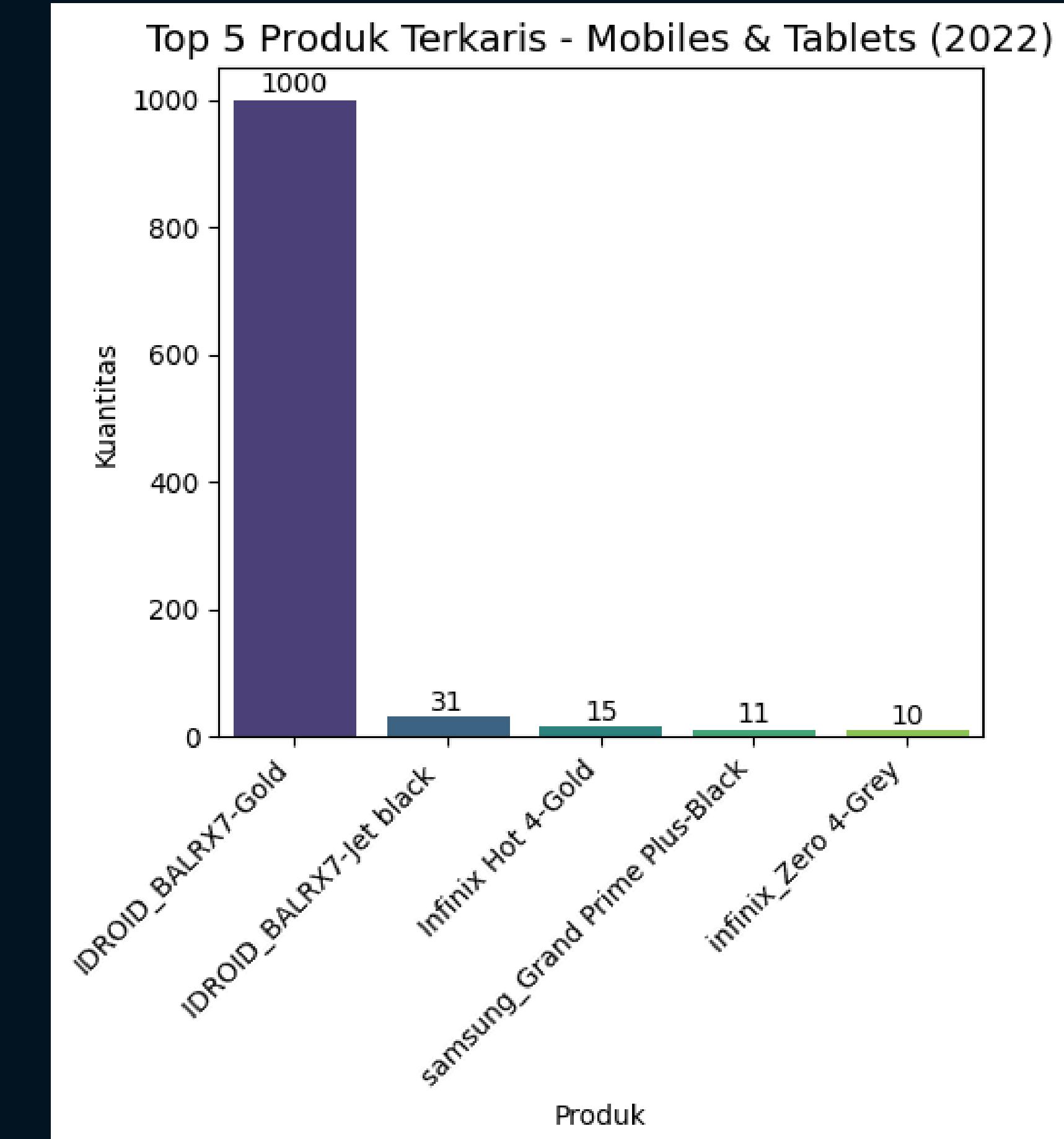
- menampilkan 5 product teratas dengan penjualan tertinggi menggunakan fungsi **.head()**

Result



hadiah yang dapat diberikan kepada pemenang kompetisi untuk Festival Akhir Tahun adalah:

	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		



Insight

IDROID_BALRX7-Gold adalah produk dengan penjualan paling laris yaitu mencapai 1.000 unit, sementara 4 produk dibawahnya hanya terjual tidak sampai 50 unit. Jarak penjualan yang sangat besar mengindikasikan bahwa hanya satu produk yang mendominasi pasar di tahun 2022, sedangkan produk lainnya memiliki performa penjualan yang paling rendah.



Recommendation

Perusahaan bisa meniru strategi IDROID_BALRX-GOLD, misalnya dari cara promosinya, diskon, atau penempatan produk yang menarik. Strategi ini bisa dicoba untuk produk lain yang kurang laku agar penjualannya ikut naik. Selain itu, pemasaran dan distribusi juga perlu dievaluasi supaya lebih fokus ke produk yang berpeluang tinggi laku di pasaran





Question 2

Menindaklanjuti meeting gabungan Tim Werehouse 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 Werehouse

```
1 #Menfilter data frame 'is_valid' jika bernilai 1(true) dan tahun 2021,2022
2 df_filter=df[
3     |     (df['is_valid']==1) &
4     |     (df['years'].between(2021,2022)))
5 ]
6 print(df_filter)
```

Filter data:

Membuat variabel '**filtered**' dengan values:

- Filter data dengan '**is_valid**' adalah 1,
- Filter data dengan '**years**' berdasarkan transaksi pada tahun 2021 dan 2022



```
1 #Mengelompokkan dan Menjumlahkan Data
2 grouped = df_filter.groupby(['category', 'years'])['qty_ordered'].sum().reset_index()
```

Groupby

- Mengelompokkan df_filter berdasarkan nama kategori (category) dan tahun (years), lalu menjumlahkan kuantitas pesanan (qty_ordered) untuk setiap grup, dan mengubah hasilnya kembali menjadi DataFrame biasa.

```
#Memisahkan Data per Tahun
```

```
df_2021 = grouped[grouped['years']==2021]
df_2022 = grouped[grouped['years']==2022]
```

Groupby

- Baris pertama membuat DataFrame baru bernama df_2021 yang hanya berisi data dari grouped di mana kolom years bernilai 2021.
- Baris kedua melakukan hal yang sama untuk tahun 2022 dan menyimpannya di df_2022.

```
#Menggabungkan Data 2021 dan 2022
```

```
df_merge = pd.merge(df_2021, df_2022, on='category', suffixes=('_2021', '_2022'))
```

Merge

- pd.merge(_): Perintah yang menggabungkan dua DataFrame (df_2021 dan df_2022) menjadi satu tabel baru.
- on='category': Penggabungan dilakukan dengan mencocokkan nilai pada kolom category. Jadi, baris untuk produk yang sama dari kedua tahun akan disatukan.
- suffixes=('_2021', '_2022'): Karena kedua DataFrame memiliki nama kolom yang sama (misalnya qty_ordered), suffixes menambahkan akhiran untuk membedakannya. Kolom qty_ordered dari df_2021 akan menjadi qty_ordered_2021, dan dari df_2022 menjadi qty_ordered_2022.



```
#Menghitung Selisih Pertumbuhan  
df_merge['selisih'] = df_merge['qty_ordered_2022'] - df_merge['qty_ordered_2021']
```

Merge

- Membuat kolom baru bernama selisih.
- Nilainya dihitung dengan mengurangi jumlah penjualan tahun 2021 dari jumlah penjualan tahun 2022 untuk setiap produk. Hasil positif berarti penjualan naik, hasil negatif berarti turun.



```
#Membuat Kolom Status  
df_merge['status'] = df_merge['selisih'].apply(lambda x: 'Naik' if x > 0 else ('Turun' if x < 0 else 'Tetap'))
```

Merge

- `df_merge['selisih'].apply(_)`: Menerapkan sebuah fungsi pada setiap nilai di kolom selisih.
- `lambda x_`: Ini adalah fungsi singkat (fungsi lambda) yang memeriksa nilai x (nilai selisih):
 - a.Jika $x > 0$ (selisihnya positif), hasilnya adalah string 'Naik'.
 - b.Jika $x < 0$ (selisihnya negatif), hasilnya adalah string 'Turun'.
 - c.Jika tidak keduanya (berarti x adalah 0), hasilnya adalah 'Tetap'.
- Hasil dari fungsi ini disimpan dalam kolom baru bernama status.



#Mengurutkan Data

```
df_merge = df_merge.sort_values(by='selisih', ascending=False)
print(df_merge.head(15))
```

Merge

- Baris ini mengurutkan seluruh DataFrame df_merge berdasarkan nilai pada kolom selisih.
- ascending=False: Mengurutkan dari nilai terbesar ke terkecil. Artinya, produk dengan kenaikan penjualan paling tinggi akan berada di paling atas.
- Mencetak atau menampilkan 15 baris pertama dari DataFrame yang sudah diurutkan. Ini secara efektif menunjukkan 15 produk dengan pertumbuhan penjualan tertinggi dari tahun 2021 ke 2022.



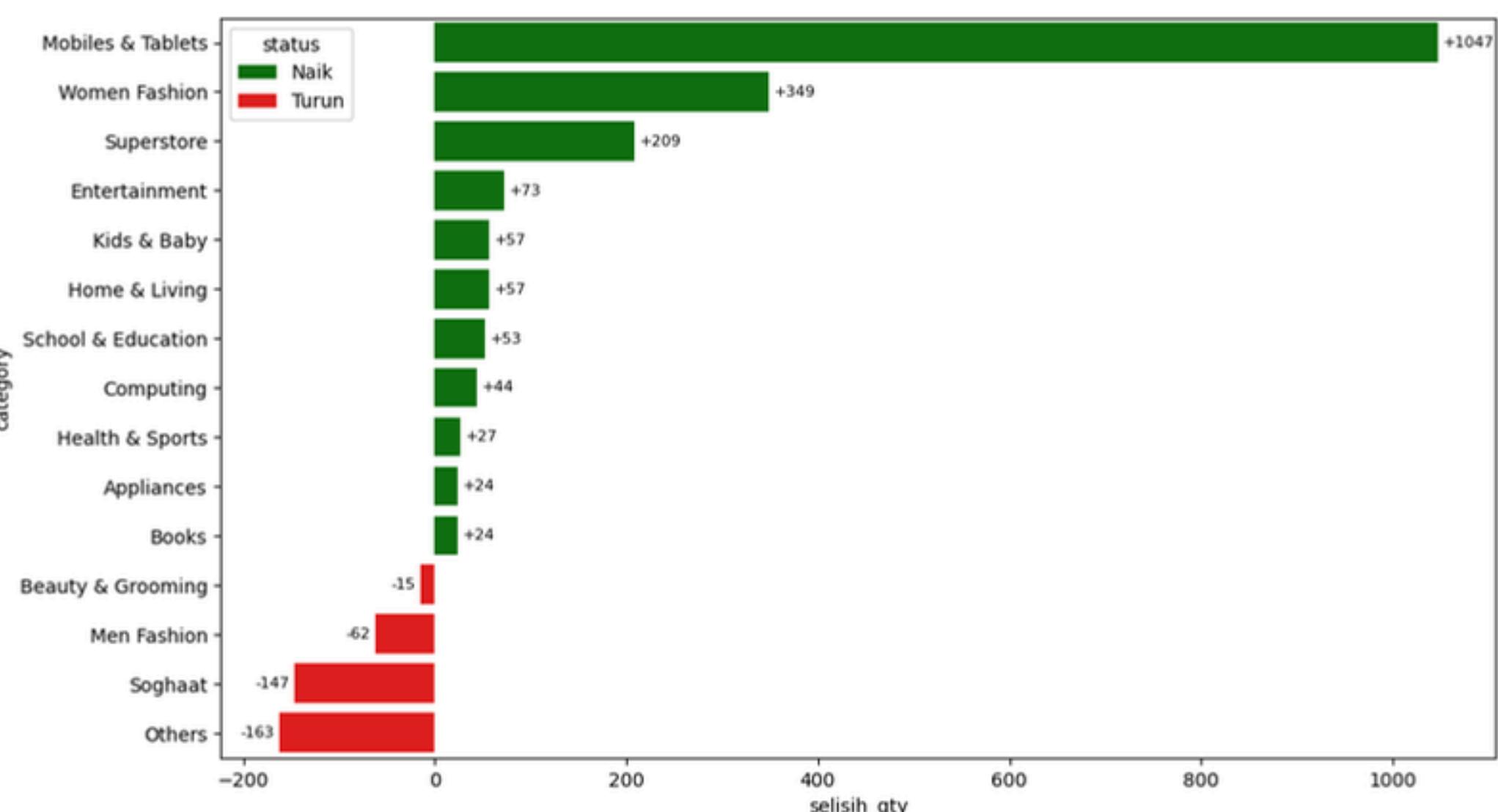
Urutan pengecekan data penjualan

```
1 #Menfilter data frame 'is_valid' jika bernilai 1(true) dan tahun 2021,2022
2 df_filter=df[
3     |  (df['is_valid']==1) &
4     |  (df['years'].between(2021,2022))
5 ]
6 print(df_filter)
```

```
1 #Mengelompokkan dan Menjumlahkan Data
2 grouped = df_filter.groupby(['category', 'years'])['qty_ordered'].sum().reset_index()
3 #Memisahkan Data per Tahun
4 df_2021 = grouped[grouped['years']==2021]
5 df_2022 = grouped[grouped['years']==2022]
6 #Menggabungkan Data 2021 dan 2022
7 df_merge = pd.merge(df_2021, df_2022, on='category',suffixes=('_2021', '_2022'))
8 #Menghitung Selisih Pertumbuhan
9 df_merge['selisih'] = df_merge['qty_ordered_2022'] - df_merge['qty_ordered_2021']
10 #Membuat Kolom Status
11 df_merge['status'] = df_merge['selisih'].apply(lambda x: 'Naik' if x > 0 else ('Turun' if x < 0 else 'Tetap'))
12 #Mengurutkan Data
13 df_merge = df_merge.sort_values(by='selisih', ascending=False)
14 display(df_merge.head(15))
15
```

Result

	category	years_2021	qty_ordered_2021	years_2022	qty_ordered_2022	selisih	status
9	Mobiles & Tablets	2021	107	2022	1154	1047	Naik
14	Women Fashion	2021	140	2022	489	349	Naik
13	Superstore	2021	327	2022	536	209	Naik
4	Entertainment	2021	77	2022	150	73	Naik
7	Kids & Baby	2021	170	2022	227	57	Naik
6	Home & Living	2021	193	2022	250	57	Naik
11	School & Education	2021	184	2022	237	53	Naik
3	Computing	2021	109	2022	153	44	Naik
5	Health & Sports	2021	173	2022	200	27	Naik
0	Appliances	2021	124	2022	148	24	Naik
2	Books	2021	171	2022	195	24	Naik
1	Beauty & Grooming	2021	168	2022	153	-15	Turun
8	Men Fashion	2021	237	2022	175	-62	Turun
12	Soghaat	2021	759	2022	612	-147	Turun
10	Others	2021	426	2022	263	-163	Turun



Insight

Pertumbuhan Kuat: Kategori seperti Mobiles & Tablets (+1047), Women Fashion (+349), dan Superstore (+209) menunjukkan peningkatan penjualan yang sangat baik. Kategori lain seperti Entertainment, Kids & Baby, Home & Living, School & Education, Computing, Health & Sports, Appliances, dan Books juga tumbuh positif



Potensi

Optimalisasi Strategi untuk Kategori

Berkinerja Tinggi: Peluang dan Tindakan dari kategori Mobiles & Tablets, bisa menjadi patokan dengan meningkatkan investasi dan memperluas jangkauan penjualan produk.

Optimalisasi Stok dan Distribusi: Dengan adanya pertumbuhan yang jelas, pastikan ketersediaan stok dan efisiensi distribusi untuk Tambah stok untuk kategori yang naik, kurangi untuk yang turun. agar tetap terjaga tidak kehilangan momentum penjualan.



```
1 df_filter2=df[  
2     (df['is_valid']==1) &  
3     (df['years'].between(2021,2022)) &  
4     (df['category'].str.lower()=='others')  
5 ]  
6 print(df_filter2)
```

Filter data:

Membuat variabel '**filtered**' dengan values:

- Filter data dengan 'is_valid' adalah 1,
- Filter data dengan 'years' berdasarkan transaksi pada tahun 2021 dan 2022
- Filter data dengan 'catagory' dengan mengubah semua huruf menjadi kecil



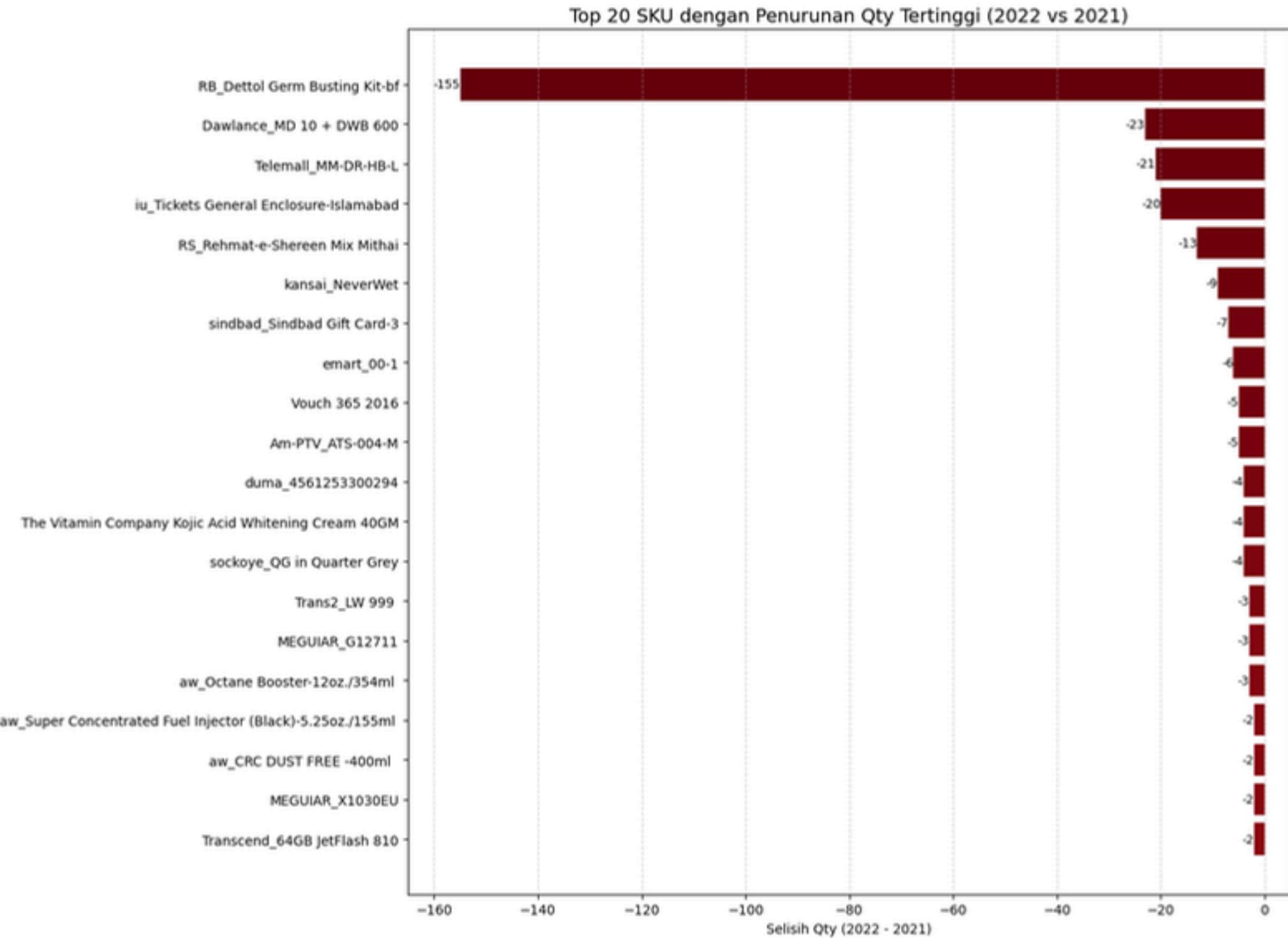
```
1 grouped = df_filter2.groupby(['sku_name', 'years'])['qty_ordered'].sum().reset_index()
2
3 df_2021 = grouped[grouped['years'] == 2021][['sku_name', 'qty_ordered']].rename(columns={'qty_ordered': 'qty_2021'})
4 df_2022 = grouped[grouped['years'] == 2022][['sku_name', 'qty_ordered']].rename(columns={'qty_ordered': 'qty_2022'}) # Calculate quantities for 2022
5
6 merged = pd.merge(df_2021, df_2022, on='sku_name', how='outer').fillna(0) # Merge df_2021 and df_2022
7 merged['selisih'] = merged['qty_2022'] - merged['qty_2021']
8
9 penurunan = merged[merged['selisih'] < 0] # Filter for products with a decrease in quantity
10 top_20_penurunan = penurunan.sort_values(by='selisih', ascending=True).head(20) # Sort by difference and get the top 20
11
12 top_20_penurunan
```

Merge

- `ascending=True`: Mengurutkan dari nilai terkecil ke tebesar. Artinya, produk dengan kenaikan penjualan paling rendah akan berada di paling atas.
- Mencetak atau menampilkan 20 baris pertama dari DataFrame yang sudah diurutkan. Ini secara efektif menunjukkan 15 produk dengan pertumbuhan penjualan tertinggi dari tahun 2021 ke 2022.

Result

	sku_name	qty_2021	qty_2022	selisih
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
85	The Vitamin Company Kojic Acid Whitening Cream...	4.0	0.0	-4.0
155	sockeye_QG in Quarter Grey	4.0	0.0	-4.0
88	Trans2_LW 999	3.0	0.0	-3.0
43	MEGUIAR_G12711	4.0	1.0	-3.0
115	aw_Octane Booster-12oz./354ml	3.0	0.0	-3.0
119	aw_Super Concentrated Fuel Injector (Black)-5....	2.0	0.0	-2.0
109	aw_CRC DUST FREE -400ml	2.0	0.0	-2.0
60	MEGUIAR_X1030EU	2.0	0.0	-2.0
89	Transcend_64GB JetFlash 810	2.0	0.0	-2.0



Insight

Penurunan Paling Drastis pada "RB_Dettol Germ Busting Kit-bf": Produk "RB_Dettol Germ Busting Kit-bf" menunjukkan penurunan kuantitas penjualan yang paling signifikan, yaitu -155 unit (dari 200 unit di 2021 menjadi hanya 45 unit di 2022). Ini adalah performer yang paling mengkhawatirkan dan memerlukan perhatian segera



Rekomendasi

Fokus pada Penurunan Terbesar: Prioritaskan investigasi mendalam terhadap "RB_Dettol Germ Busting Kit-bf" dan SKU lain dengan penurunan signifikan. Cari tahu penyebabnya: Apakah ada masalah kualitas, perubahan harga, munculnya kompetitor, promosi yang berakhir, atau perubahan tren pasar





Question 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

Filter data:

```
df['years'] = df['order_date'].dt.year  
  
df_filtered = df[  
    (df['is_gross'] == 1) &  
    (df['is_valid'] == 0) &  
    (df['is_net'] == 0) &  
    (df['order_date'].dt.year == 2022)  
]
```



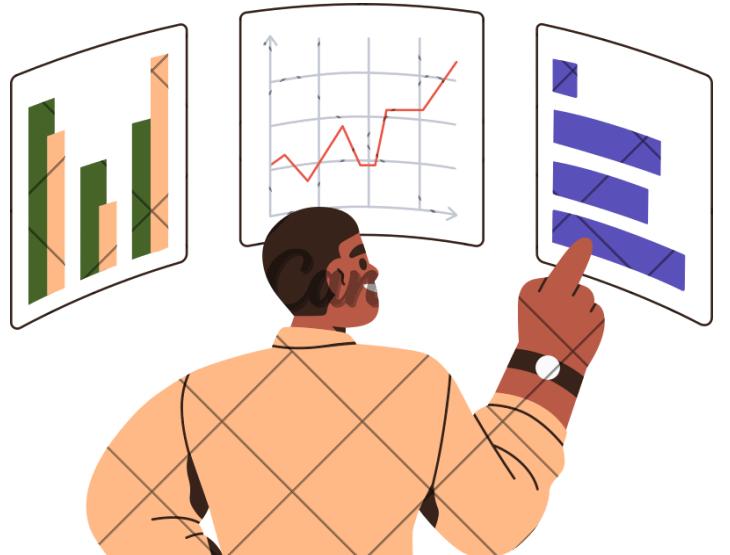
Membuat variabel '**filtered**' dengan values:

- filter data dengan 'is_gross' adalah 1,
→ Artinya masih dalam status pesanan tapi belum dibayar.
- filter data dengan 'is_valid' adalah 0
→ Artinya transaksi tersebut belum diverifikasi atau belum dianggap resmi oleh sistem.
- filter data dengan 'is_net' adalah 0
→ Menyaring transaksi yang belum masuk perhitungan akhir, bisa juga dimaknai belum settle secara sistem.
- Filter data 'order_date' berdasarkan tahun menggunakan fungsi **.dt.year** adalah tahun 2022
→ Hanya ambil transaksi yang terjadi di tahun 2022 saja.

Result

```
▶ result_df = filtered_df.loc[:, ['customer_id', 'registered_date']].drop_duplicates()  
result_df
```

Membuat variabel “**result_df**” untuk menampilkan data berdasarkan ‘**customer_id**’, ‘**registered_date**’. Kemudian membuang data ‘**customer_id**’ yang sama



	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

Kriteria Pelanggan (Check-out namun Belum Bayar di 2022): Pelanggan dengan niat beli tinggi menyatakan minat beli yang kuat terhadap produk, sehingga ini menjadi segmen yang penting untuk ditinjau lagi karena menjadi salah satu faktor pertimbangan penjualan.

Identifikasi Hambatan Pembeli: Adanya hambatan yang kemungkinan terjadi pada tahapan pembelian bisa menjadi indikasi bahwa transaksi belum selesai atau belum dibayar, hal ini bisa terjadi dengan kondisi tertentu, dari metode pembayaran ditolak, gangguan jaringan, atau harga yang tidak sesuai dengan perkiraan pembeli sehingga terjadi keraguan.

Potensi

Targeting Promo yang Sangat Efektif: Memberikan promo ulang tahun kepada segmen menjadi strategi yang baik. Mereka sudah familiar dengan produk dan hanya perlu insentif untuk menyelesaikan pembelian yang tertunda.

Umpulan Potensial: Selain promo, bisa menyertakan opsi untuk memberikan umpan balik tentang mengapa mereka tidak melanjutkan pembayaran. Ini bisa menjadi sumber informasi berharga untuk perbaikan user experience di masa mendatang.

Result

 #Jalankan kode ini untuk mendownload file
#32
from google.colab import files
result_df.to_csv('audience_list.csv', encoding='utf-8-sig', index=False)
files.download('audience_list.csv')

Mengimpor modul files dari penyimpanan google colab, kemudian menyimpan “**result_df**” ke bentuk file csv dengan nama “**audience_list**”. Hasil tersebut akan didownload menggunakan nama “**audience_list**”





Question 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

Answer 4.1

```
[51] print(df.columns)
    Show hidden output

[52] df['day'] = df['order_date'].dt.day_name()
df['month'] = df['order_date'].dt.month
df['month_number'] = df['order_date'].dt.month
df['years'] = df['order_date'].dt.year
df['day_type'] = df['day'].apply(lambda x: 'weekend' if x in ['Saturday', 'Sunday'] else 'weekday')
df['total_sales'] = df['price'] * df['qty_ordered']
df['day_type']

    Show hidden output
+ Code + Text

[53] df_filter1 = df[
    (df['is_valid']==1) &
    (df['month_number'].isin([10,11,12])) &
    (df['years']==2022)
]
```

Interpretation

Menampilkan daftar semua kolom dalam data df.

mengubah tanggal jadi informasi hari, bulan, dan tahun, lalu menentukan apakah hari itu weekend atau weekday. Setelah itu, dihitung total penjualan dari harga × jumlah barang. Hasilnya dipakai untuk analisis penjualan berdasarkan waktu.

menyaring data df untuk mengambil transaksi yang valid, terjadi pada bulan Oktober, November, atau Desember, dan di tahun 2022. Hasilnya disimpan di df_filter1.

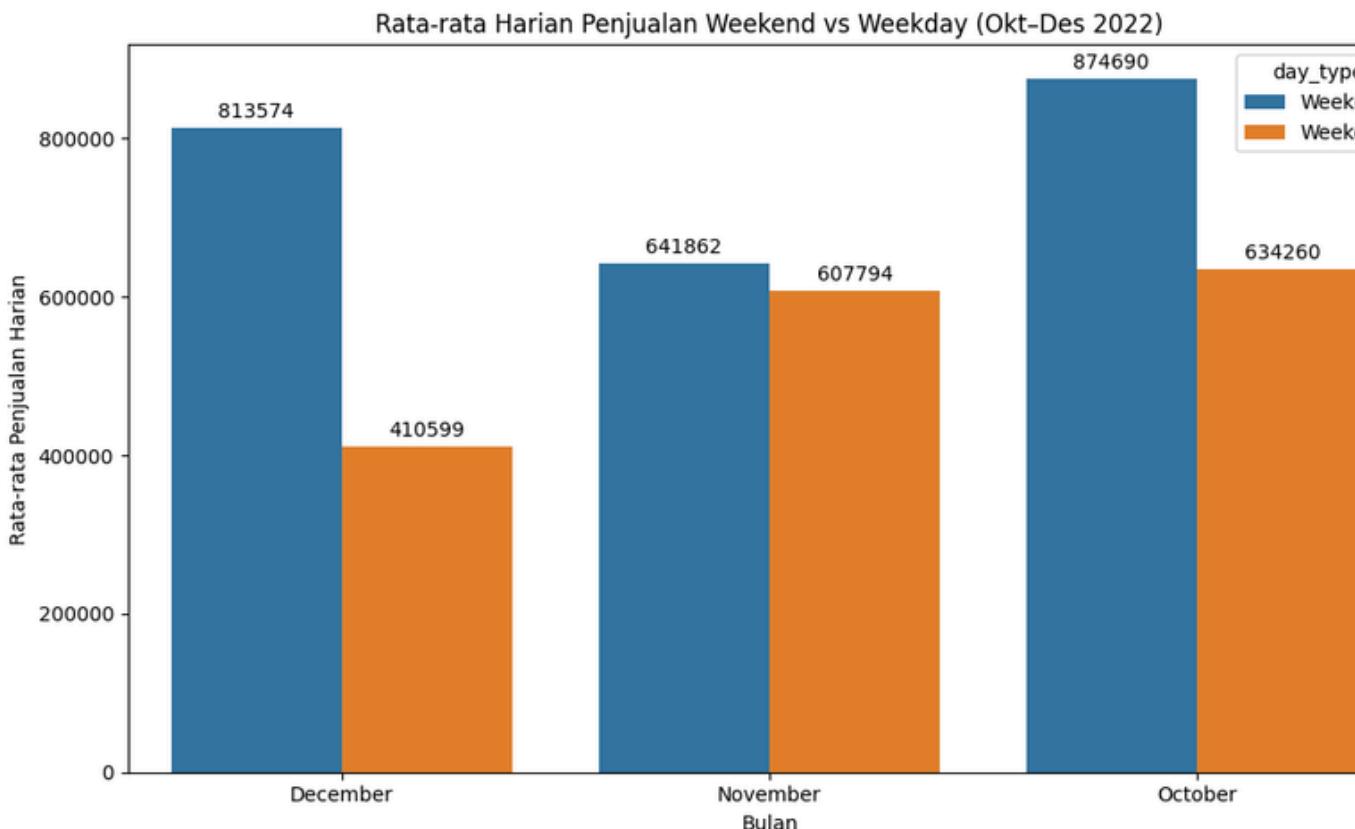
menjumlahkan total penjualan per hari, lalu dikelompokkan berdasarkan tanggal transaksi, bulan, dan tipe hari (weekday/weekend). Hasilnya disimpan di df_result1 dan ditampilkan dengan print().

menghitung rata-rata penjualan harian dari data valid selama Oktober–Desember 2022, berdasarkan bulan dan jenis hari (weekday/weekend). Hasilnya ditampilkan dalam urutan bulan dan tipe hari, dengan format angka tanpa desimal.

Answer 4.1

```
Commands + Code + Text + Run all ↗
In [34]: plt.figure(figsize=(10, 6))
ax = sns.barplot(data=df_results, x='month', y='df_results_avg', hue='day_type')
plt.title('Rata-rata Penjualan Weekend vs Weekday (Okt-Des 2022)')
plt.xlabel('Bulan')
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f', label_type='edge', padding=2)
plt.tight_layout()
plt.show()
```

Result



Interpretation

membuat grafik batang yang membandingkan rata-rata penjualan harian antara weekend dan weekday di setiap bulan (Oktober–Desember 2022). Grafik ditampilkan dengan ukuran 10×6 dan label yang rapih.

Insight

terlihat bahwa rata-rata penjualan harian di weekdays (Senin sampai Jumat) lebih tinggi dibandingkan hari weekend (Sabtu dan Minggu) di bulan Oktober, November, dan Desember 2022.

Padahal, perusahaan sudah menjalankan campaign khusus di akhir pekan, tapi hasilnya belum terlihat efektif. Bahkan, penjualan weekend paling rendah terjadi di bulan Desember.

Artinya, campaign yang dijalankan di hari Sabtu dan Minggu belum berhasil meningkatkan penjualan. Mungkin strategi promonya perlu dievaluasi atau disesuaikan dengan kebiasaan belanja konsumen

Answer 4.2

Interpretation

```
(37) # Tulis kode Anda di bawah ini. Dapat menggunakan lebih dari 1 blok kode
df_filter1 = df[
    (df['is_valid']==1) &
    (df['month_number'].isin([10,11,12])) &
    (df['year']==2022)
]

df_result1 = (
    df_filter1.groupby(['order_date','month','day_type'])['total_sales']
    .sum()
    .reset_index()
)
print(df_result1)
```

```
avg_3month_sales = (
    df_filter1
    .groupby('day_type')[['total_sales']]
    .mean()
    .reset_index()
    .rename(columns={'total_sales': 'avg_3month_sales'})
)
print(avg_3month_sales)

plt.figure(figsize=(10, 6))
ax = sns.barplot(data=avg_3month_sales, x='day_type', y='avg_3month_sales', hue='day_type')
plt.title('Rata-rata Penjualan Weekend vs Weekday (Okt-Des 2022)')
plt.xlabel('Rata-rata Penjualan Harian')
plt.ylabel('Jumlah')
for container in ax.containers:
    ax.bar_label(container, fmt='%.2f', label_type='edge', padding=10)
plt.tight_layout()
plt.show()
```

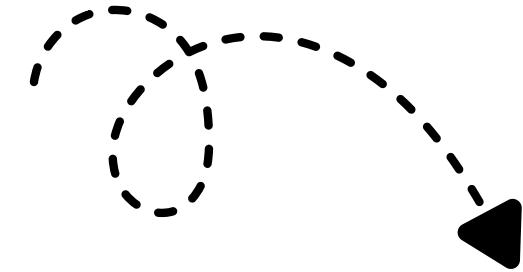
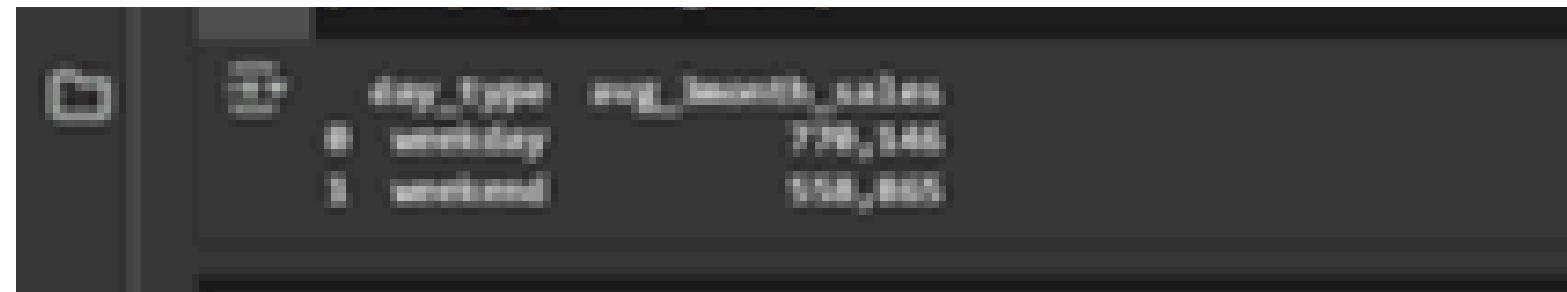
menyaring data untuk mengambil transaksi yang valid, terjadi di bulan Oktober–Desember, dan di tahun 2022. Hasilnya disimpan di df_filter1.

menghitung total penjualan per hari berdasarkan tanggal, bulan, dan tipe hari (weekday/weekend). Hasilnya disimpan di df_result1 dan ditampilkan.

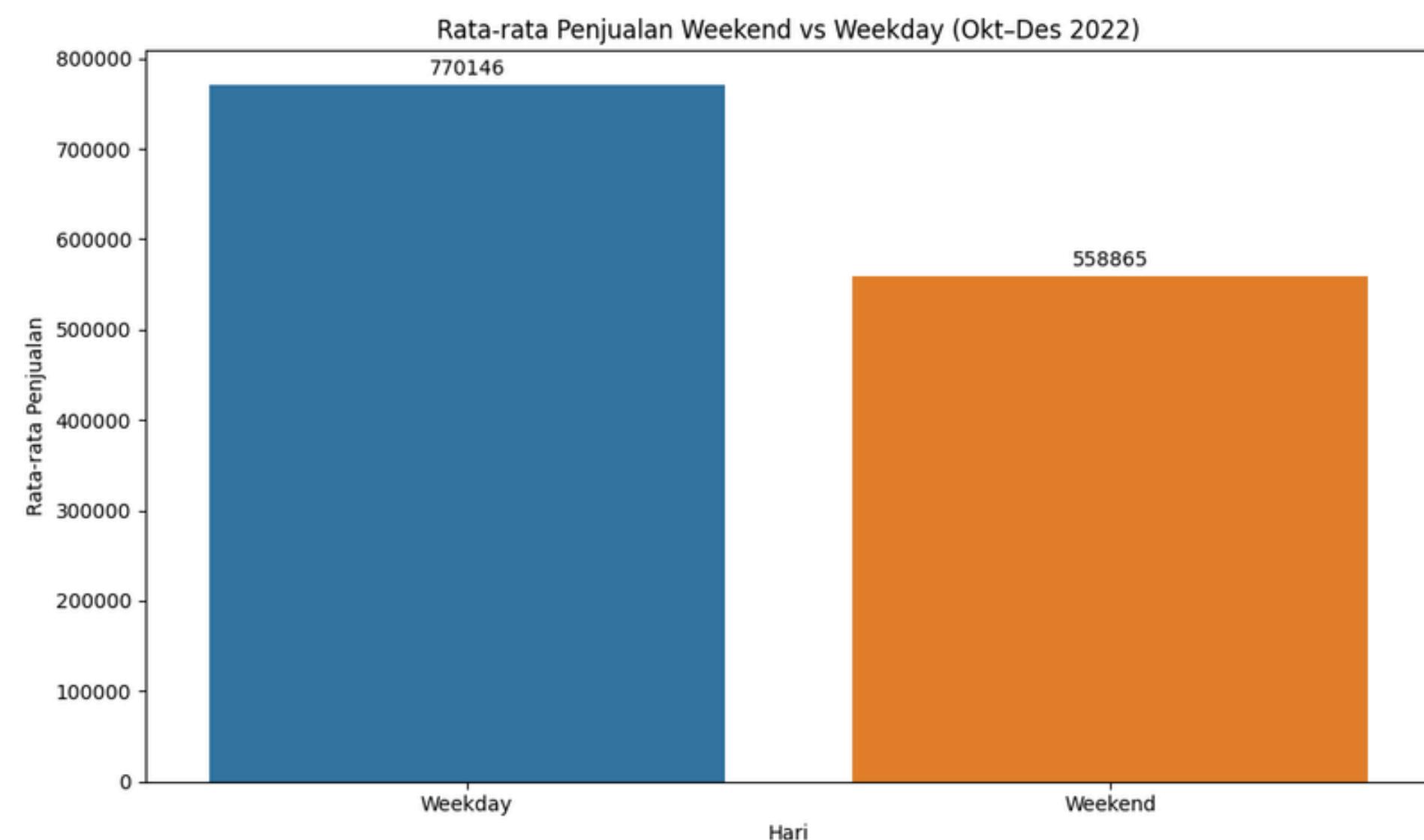
menghitung rata-rata penjualan harian berdasarkan tipe hari (weekday atau weekend) saja. Hasilnya disimpan di avg_3month_sales dan ditampilkan

membuat grafik batang untuk membandingkan rata-rata penjualan harian antara weekend dan weekday. Grafik ditampilkan dengan ukuran 10×6

Result



Insight



Selama periode Oktober hingga Desember 2022, rata-rata penjualan harian pada hari kerja (Senin-Jumat) tercatat lebih tinggi dibandingkan akhir pekan (Sabtu-Minggu). Meskipun telah dilakukan campaign khusus setiap weekend, hasil penjualannya justru tidak menunjukkan peningkatan yang signifikan. Hal ini menunjukkan bahwa campaign belum memberikan dampak yang efektif terhadap penjualan di akhir pekan, sehingga strategi promosi pada hari Sabtu dan Minggu perlu dievaluasi agar lebih sesuai dengan perilaku belanja konsumen.





Thank You.

#RintisKarirImpian

#MySkill