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


df_cd = pd.read_csv('customer_detail.csv')
df_od = pd.read_csv('order_detail.csv')
df_pd = pd.read_csv('payment_detail.csv')
df_sd = pd.read_csv('sku_detail.csv')

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   id               3998 non-null   object
 1   registered_date  3998 non-null   object
dtypes: object(2)
memory usage: 62.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5884 entries, 0 to 5883
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               5884 non-null   object 
 1   customer_id      5884 non-null   object 
 2   order_date       5884 non-null   object 
 3   sku_id           5884 non-null   object 
 4   price            5884 non-null   int64  
 5   qty_ordered      5884 non-null   int64  
 6   before_discount  5884 non-null   float64
 7   discount_amount  5884 non-null   float64
 8   after_discount   5884 non-null   float64
 9   is_gross         5884 non-null   int64  
 10  is_valid         5884 non-null   int64  
 11  is_net           5884 non-null   int64  
 12  payment_id       5884 non-null   int64  
dtypes: float64(3), int64(6), object(4)
memory usage: 597.7+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   id              16 non-null     int64 
 1   payment_method  16 non-null     object
dtypes: int64(1), object(1)
memory usage: 388.0+ bytes
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3206 entries, 0 to 3205
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   id          3206 non-null   object 
 1   sku_name    3206 non-null   object 
 2   base_price  3206 non-null   float64
 3   cogs        3206 non-null   int64  
 4   category    3206 non-null   object 
dtypes: float64(1), int64(1), object(3)
memory usage: 125.4+ KB
```

```
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
```

```
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)
```

```
df.head()
```

	id	customer_id	order_date	sku_id	price	qty_ordered	before_discount	discount_amount	after_discount	is_gross	is
0	ODR1111336599a	C993531L	2022-02-23	P569750	62002	1	62002.0	0.0	62002.0	1	
1	ODR1116457650x	C517907L	2022-10-08	P320550	159500	1	159500.0	0.0	159500.0	1	
2	ODR1119282607l	C192625L	2021-08-07	P146984	919300	1	919300.0	0.0	919300.0	1	
3	ODR1119964698b	C225180L	2021-08-11	P523149	59856	1	59856.0	0.0	59856.0	1	
4	ODR1120310291y	C708473L	2021-09-18	P692619	34510	1	34510.0	0.0	34510.0	1	

Langkah berikutnya: Buat kode dengan df | Lihat plot yang direkomendasikan | New interactive sheet

```
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

```
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

```
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

Top 5 Best-Selling Mobile Products in 2022

No 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.

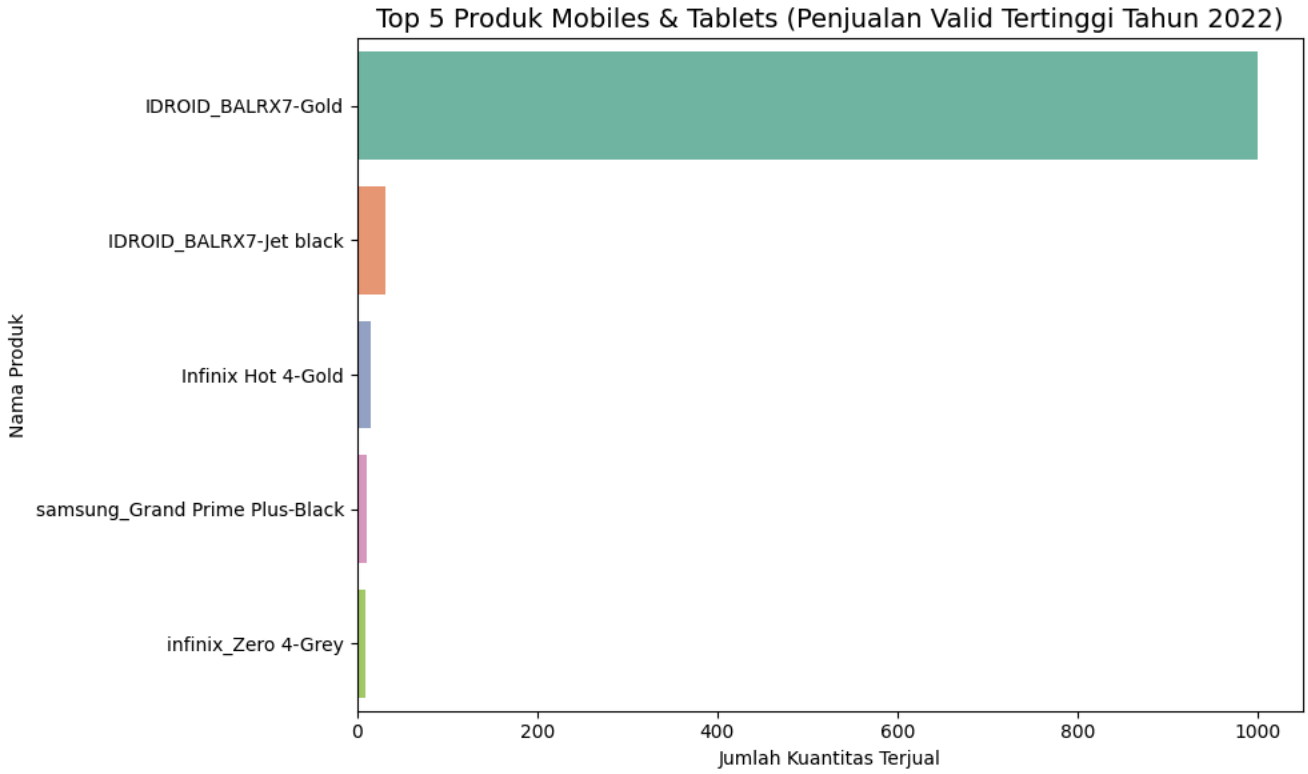
Regards

Tim Marketing

```
# Filter data tahun 2022, valid, dan kategori Mobiles & Tablets
df_2022_valid = df[
    (df['category'] == 'Mobiles & Tablets') &
    (df['is_valid'] == 1) &
    (df['order_date'].dt.year == 2022)
]

# Hitung total kuantitas penjualan tiap produk
top_5_mobiles = (
    df_2022_valid.groupby('sku_name')['qty_ordered']
    .sum()
    .sort_values(ascending=False)
    .head(5)
    .reset_index()
)

# Tampilkan Top 5 Produk Mobiles & Tablets (Penjualan Valid Tertinggi Tahun 2022
```

```
print('Top 5 Produk Mobiles & Tablets (Penjualan Valid Tertinggi Tahun 2022')
print(top_5_mobiles)

# Visualisasi
plt.figure(figsize=(10,6))
sns.barplot(
    data=top_5_mobiles,
    y='sku_name',
    x='qty_ordered',
    hue='sku_name',
    palette='Set2',
    legend=False
)
plt.title('Top 5 Produk Mobiles & Tablets (Penjualan Valid Tertinggi Tahun 2022)', fontsize=14)
plt.xlabel('Jumlah Kuantitas Terjual')
plt.ylabel('Nama Produk')
plt.tight_layout()
plt.show()
```

```
Top 5 Produk Mobiles & Tablets (Penjualan Valid Tertinggi Tahun 2022
                        sku_name  qty_ordered
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
```



To support the company's year-end gift program, an analysis was conducted to identify the top 5 best-selling products from the Mobiles & Tablets category in 2022 based on valid transactions. The results showed that IDROID_BALRX7-Gold dominated the sales significantly with 1000 units sold, while the other products in the top five had less than 35 units sold. This highlights the strong customer preference for this product, making it a strong candidate for a promotional gift.

Sales Decline in the "Others" Category (2021 vs 2022)

No 2

Dear Data Analyst,

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)

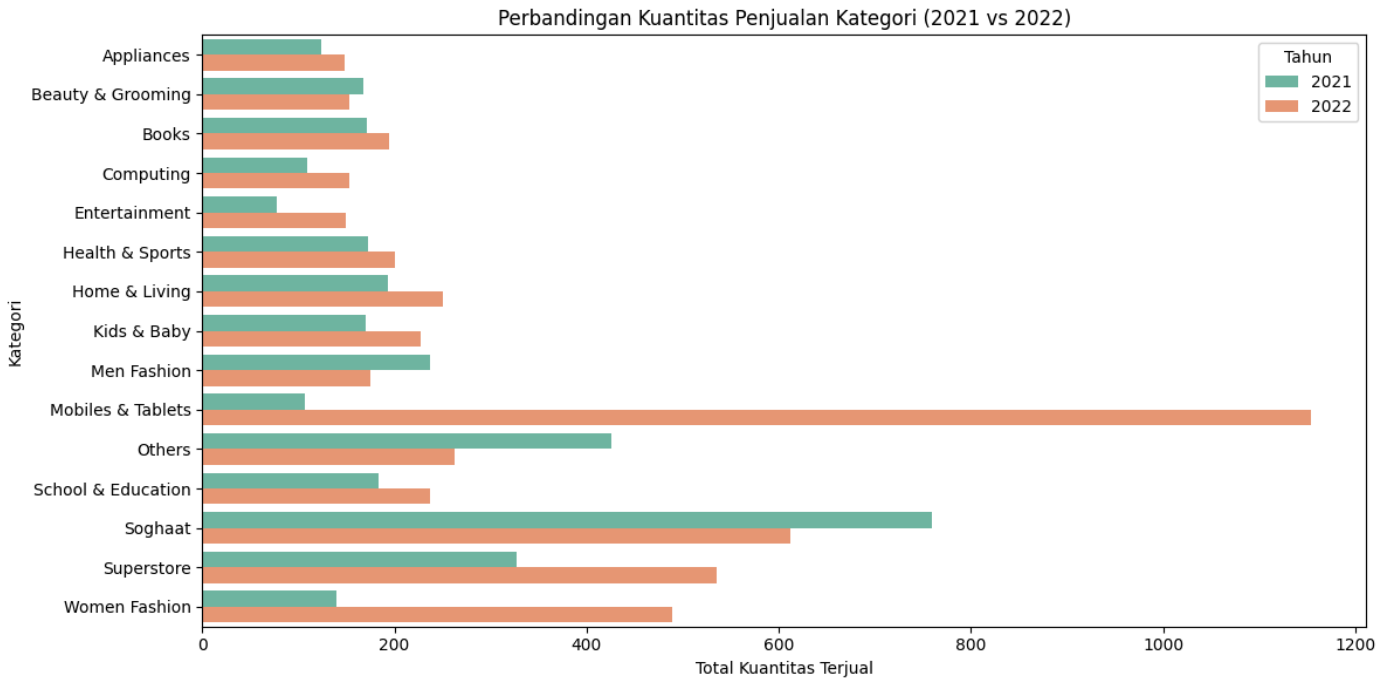
2. Apabila memang terjadi penurunan kuantitas penjualan pada kategori Others, kami mohon bantuan untuk menyediakan data TOP 20 nama produk yang mengalami penurunan paling tinggi pada 2022 jika dibanding dengan 2021. Hal ini kami gunakan sebagai bahan diskusi pada meeting selanjutnya.

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

Regards

Tim Werehouse

```
# Tambahkan kolom tahun
df['year'] = df['order_date'].dt.year

# Filter data valid saja
df_valid = df[df['is_valid'] == 1]

# Grup kategori dan tahun
sales_by_category = (
    df_valid.groupby(['category', 'year'])['qty_ordered']
    .sum()
    .reset_index()
)

# Ambil hanya data tahun 2021 dan 2022
sales_by_category = sales_by_category[sales_by_category['year'].isin([2021, 2022])]

# Tampilkan data 15 kategori teratas berdasarkan total penjualan dua tahun
top_15_categories = (
    sales_by_category.groupby('category')['qty_ordered']
    .sum()
    .sort_values(ascending=False)
    .head(15)
    .reset_index()
)

# Gabungkan dengan data per tahun untuk plot
top_categories_list = top_15_categories['category'].tolist()
filtered_sales = sales_by_category[sales_by_category['category'].isin(top_categories_list)]

plt.figure(figsize=(12, 6))
sns.barplot(data=filtered_sales, x='qty_ordered', y='category', hue='year', palette='Set2')
plt.title('Perbandingan Kuantitas Penjualan Kategori (2021 vs 2022)')
plt.xlabel('Total Kuantitas Terjual')
plt.ylabel('Kategori')
plt.legend(title='Tahun')
plt.tight_layout()
plt.show()
```




```
others_sales = (
    df_valid[df_valid['category'] == 'Others']
    .groupby('year')['qty_ordered']
    .sum()
    .reset_index()
)
print(others_sales)
```

```
   year  qty_ordered
0  2021          426
1  2022          263
```

```
# Filter data kategori Others
others_data = df_valid[df_valid['category'] == 'Others']

# Hitung qty per produk per tahun
others_by_sku = (
    others_data.groupby(['sku_name', 'year'])['qty_ordered']
    .sum()
    .unstack(fill_value=0)
    .reset_index()
)

# Tambahkan kolom selisih
others_by_sku['difference'] = others_by_sku[2022] - others_by_sku[2021]

# Ambil produk dengan penurunan tertinggi
top_20_decline = others_by_sku.sort_values('difference').head(20)

# Format tabel output yang rapi
top_20_decline_formatted = top_20_decline[['sku_name', 2021, 2022, 'difference']].copy()
top_20_decline_formatted.columns = ['Produk', 'Qty 2021', 'Qty 2022', 'Penurunan']

# Reset index dan cetak dengan tabulate
from tabulate import tabulate

print(tabulate(top_20_decline_formatted, headers='keys', tablefmt='fancy_grid', showindex=True))
```

```
╒════╤═══════════════════════════════════════╤═══════════╤═══════════╤═══════════╕
│    │ Produk                                │  Qty 2021 │  Qty 2022 │ Penurunan │
╞════╪═══════════════════════════════════════╪═══════════╪═══════════╪═══════════╡
│ 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 40GM	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)-5.25oz./155ml	2	0	-2
156	sstop_Universallensclipkit	2	0	-2
89	Transcend_64GB JetFlash 810	2	0	-2
26	JBS_IFAM-009	2	0	-2

The warehouse team suspected a drop in sales for the “Others” category in 2022 compared to 2021. The data confirmed this with sales declining from 426 units in 2021 to 263 units in 2022. Additionally, the analysis identified the top 20 products with the steepest decline in sales, such as RB_Dettol Germ Busting Kit-bf, which dropped by 155 units. This insight is useful for inventory optimization and planning future marketing strategies.

Customers with Unpaid Orders in 2022

No 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

```
dtpromo = df[
    (df['is_gross']==1) &
    (df['is_valid']==0) &
    (df['is_net']==0) &
    (df['order_date'].dt.year==2022)
]
dtpromo = dtpromo[['customer_id','registered_date']]
dtpromo.sort_values(by=['customer_id'])
answer3 = dtpromo.drop_duplicates(subset=['customer_id'])
answer3
```

	customer_id	registered_date
0	C993531L	2021-11-08
8	C652751L	2022-07-19
19	C189144L	2022-07-01
22	C224183L	2022-04-07
23	C907186L	2022-05-08
...	...	...
5842	C118563L	2022-06-07
5849	C217246L	2022-08-04
5854	C577176L	2022-06-01
5862	C744990L	2022-05-21
5877	C417245L	2022-07-26

820 rows × 2 columns

Langkah berikutnya: Buat kode dengan answer3 | Lihat plot yang direkomendasikan | New interactive sheet

```
from google.colab import files
dtpromo.to_csv('List Promo Customer.csv', encoding = 'utf-8-sig', index=False)
files.download('List Promo Customer.csv')
```

As part of a marketing initiative for the company's upcoming anniversary, the team identified customers who had checked out but did not complete payment in 2022 (is_gross = 1, is_valid = 0, is_net = 0). A total of 820 unique customers were found. Their customer IDs and registration dates were compiled and can be used for targeted promotional campaigns aimed at encouraging conversions.

Impact of Weekend Campaigns (Oct–Dec 2022) on Sales

No 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

```
df['day'] = df['order_date'].dt.day_name()
df['month'] = df['order_date'].dt.month
df['month_num'] = df['order_date'].dt.month_name()

daily_weekend = (
    df.loc[
        (df['is_valid'] == 1) &
        (df['day'].isin(['Saturday', 'Sunday'])) &
        (df['order_date'].between('2022-10-01', '2022-12-31'))
    ]
    .groupby(['month', 'month_num', 'order_date'])
    .agg(daily_sales_weekend=('before_discount', 'sum'))
    .round()
```

```
    .reset_index()
)

daily_weekend
```

	month	month_num	order_date	daily_sales_weekend
0	10	October	2022-10-01	7603220
1	10	October	2022-10-02	2233348
2	10	October	2022-10-08	7288454
3	11	November	2022-11-05	6545822
4	11	November	2022-11-06	5002268
5	12	December	2022-12-03	2121234
6	12	December	2022-12-04	6090754

Langkah berikutnya: Buat kode dengan daily_weekend | Lihat plot yang direkomendasikan | New interactive sheet

```
df_weekend = (
    daily_weekend
    .groupby(by=["month_num", "month"], as_index=False)
    .agg(avg_sales_weekend=('daily_sales_weekend', 'mean'))
    .round()
    .sort_values(by='avg_sales_weekend', ascending=False)
    .reset_index(drop=True)
)

df_weekend
```

	month_num	month	avg_sales_weekend
0	November	11	5774045.0
1	October	10	5708341.0
2	December	12	4105994.0

Langkah berikutnya: Buat kode dengan df_weekend | Lihat plot yang direkomendasikan | New interactive sheet

```
daily_weekday = (
    df.loc[
        (df['is_valid'] == 1) &
        (df['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])) &
        (df['order_date'].between('2022-10-01', '2022-12-31'))
    ]
    .groupby(['month', 'month_num', 'order_date'])
    .agg(daily_sales_weekday=('before_discount', 'sum'))
    .round()
    .reset_index()
)

daily_weekday
```

	month	month_num	order_date	daily_sales_weekday
0	10	October	2022-10-03	5111076
1	10	October	2022-10-04	12259634
2	10	October	2022-10-05	10050704
3	10	October	2022-10-06	4050430
4	10	October	2022-10-07	7889218
5	11	November	2022-11-01	8999106
6	11	November	2022-11-02	781782
7	11	November	2022-11-03	4296930
8	11	November	2022-11-04	11829274
9	11	November	2022-11-07	3706084
10	11	November	2022-11-08	7614820
11	12	December	2022-12-01	8811882
12	12	December	2022-12-02	2678962
13	12	December	2022-12-05	2699668
14	12	December	2022-12-06	9767606
15	12	December	2022-12-07	8352754
16	12	December	2022-12-08	18944308

Langkah berikutnya: Buat kode dengan daily_weekday | Lihat plot yang direkomendasikan | New interactive sheet

```
df_weekday = (
    daily_weekday
    .groupby(by=["month_num", "month"], as_index=False)
    .agg(avg_sales_weekday=('daily_sales_weekday', 'mean'))
    .round()
    .sort_values(by='avg_sales_weekday', ascending=False)
    .reset_index(drop=True)
)

df_weekday
```

	month_num	month	avg_sales_weekday
0	December	12	8542530.0
1	October	10	7872212.0
2	November	11	6204666.0

Langkah berikutnya: Buat kode dengan df_weekday | Lihat plot yang direkomendasikan | New interactive sheet

```
merged_avg_sales = pd.merge(df_weekend, df_weekday, on=['month', 'month_num'])
merged_avg_sales['difference'] = merged_avg_sales['avg_sales_weekend'] - merged_avg_sales['avg_sales_weekday']
merged_avg_sales['percent_diff'] = (merged_avg_sales['difference'] / merged_avg_sales['avg_sales_weekend']) * 100

merged_avg_sales
```

	month_num	month	avg_sales_weekend	avg_sales_weekday	difference	percent_diff
0	November	11	5774045.0	6204666.0	-430621.0	-7.457874
1	October	10	5708341.0	7872212.0	-2163871.0	-37.907178
2	December	12	4105994.0	8542530.0	-4436536.0	-108.050231

Langkah berikutnya: Buat kode dengan merged_avg_sales | Lihat plot yang direkomendasikan | New interactive sheet

```
import matplotlib.pyplot as plt
import numpy as np

# Data untuk chart
bulan = merged_avg_sales['month_num']
```

```
    weekend_sales = merged_avg_sales['avg_sales_weekend']
    weekday_sales = merged_avg_sales['avg_sales_weekday']

    # Menambahkan total 3 bulan di akhir
    bulan = list(bulan) + ['Oct-Dec']
    weekend_sales = list(weekend_sales) + [df[(df['day'].isin(['Saturday', 'Sunday'])) & (df['order_date'].between('2022-10-01', '2022-12-31'))][
    weekday_sales = list(weekday_sales) + [df[(df['day'].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])) & (df['order_date'].betw

    # Membuat posisi bar
    x = np.arange(len(bulan))
    width = 0.35

    # Membuat figure
    fig, ax = plt.subplots(figsize=(10, 6))

    # Bar chart
    bars1 = ax.bar(x - width/2, weekend_sales, width, label='Weekend', color='#72b6a1')
    bars2 = ax.bar(x + width/2, weekday_sales, width, label='Weekday', color='#e99675')

    # Menambahkan label angka di atas bar
    for bar in bars1:
        height = bar.get_height()
        ax.annotate(f'{height/1e6:.1f}M',
                    xy=(bar.get_x() + bar.get_width() / 2, height),
                    xytext=(0, 3),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

    for bar in bars2:
        height = bar.get_height()
```