



MySkill | *#RintisKarirImpian*

Portfolio - Intensive Bootcamp

# Exercise 2

## Phyton

**Owner:** Devita Eka Ratriningrum

Build your skill and portfolio via [myskill.id/bootcamp](https://myskill.id/bootcamp)

# Dataset



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

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

## customer\_detail:

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

## payment\_detail:

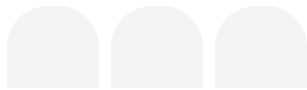
id	int64	angka unik dari metode pembayaran
payment_method	object	metode pembayaran yang digunakan



# Import pandas, numpy, matplotlib, seaborn, dan pandas.tseries.offsets



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



# Import Dataset

```
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)
```

## Menjalankan SQL di Google 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')
```

# Penggabungan Dataset Menggunakan SQL



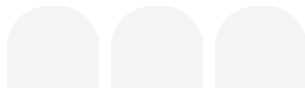
```
df = pd.read_sql("""
SELECT
    order_detail.*,
    payment_detail.payment_method,
    sku_detail.sku_name,
    sku_detail.base_price,
    sku_detail.cogs,
    sku_detail.category,
    customer_detail.registered_date
FROM order_detail
LEFT JOIN payment_detail
    on payment_detail.id = order_detail.payment_id
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)
```



# Penggabungan Dataset Menggunakan SQL



```
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)
```



# Mengubah Datatype

```
df = df.astype({"before_discount": 'int', "discount_amount": 'int', "after_discount": 'int', "base_price": 'int'})
df.dtypes
```

## Result:

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

# Mengubah Datatype

```
df['order_date'] = pd.to_datetime(df['order_date'])
df['registered_date'] = pd.to_datetime(df['registered_date'])
df.dtypes
```

## Result:

```
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
```



# Pertanyaan 1:

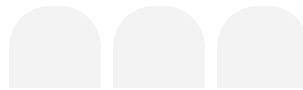


Akhir tahun ini, perusahaan akan memberikan hadiah bagi pelanggan yang memenangkan kompetisi **Festival Akhir Tahun**. Tim Marketing membutuhkan bantuan untuk menentukan perkiraan hadiah yang akan diberikan pada pemenang kompetisi nantinya. Hadiah tersebut akan diambil dari **TOP 5 Produk** dari Kategori **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**



# Pertanyaan 1:

## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

1. Membuat kolom baru untuk day, month, dan month number

### Code:

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

### Result:

```
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]  
day               object  
month             object  
month_num          int64  
year              int64  
dtype: object
```

# Pertanyaan 1:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

2. Memfilter data dengan valid = 1

**Code:** `valid = df['is_valid']==1`

1. Memfilter data dengan category = mobiles & tablets

**Code:** `category = df['category']=='Mobiles & Tablets'`

1. Memfilter data pada transaksi selama 2022

**Code:** `transaction = df['year']==2022`

1. Penggabuan

**Code:** `df_filter = df[category & valid & transaction]`

1. (
2. Gunakan sort\_values untuk mengurutkan data



# Pertanyaan 1:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

6. Gunakan groupby berdasarkan sku\_name

**Code:**

```
df_groupby = df_filter.groupby(by=["sku_id", "sku_name", "category", "year"])["qty_ordered"].sum()  
df_groupby_new = pd.DataFrame(df_groupby.reset_index(name='qty_total'))
```

1. Gunakan sort\_values untuk mengurutkan data

**Code:**

```
df_sort = df_groupby_new.sort_values(by=['qty_total'], ascending=[False])
```



# Pertanyaan 1:



## Answer:

### Code:

```
df_sort.head(5)
```

### Result:

	sku_id	sku_name	category	year	qty_total
11	P313303	IDROID_BALRX7-Gold	Mobiles & Tablets	2022	1000
36	P754831	IDROID_BALRX7-Jet black	Mobiles & Tablets	2022	31
37	P774890	Infinix Hot 4-Gold	Mobiles & Tablets	2022	15
31	P661542	samsung_Grand Prime Plus-Black	Mobiles & Tablets	2022	11
43	P926387	infinix_Zero 4-Grey	Mobiles & Tablets	2022	10

### Description:

Dari hasil coding diatas, dapat disimpulkan bahwa product **IDROID\_BALRX7-Gold** dari kategori Mobiles & Tablets merupakan produk yang paling banyak terjual pada tahun 2022



# Pertanyaan 2:



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

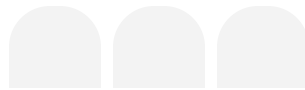
1. Kami mohon bantuan untuk melakukan pengecekan data penjualan kategori tersebut dengan tahun 2021 secara kuantitas penjualan. Dugaan sementara kami, telah terjadi penurunan kuantitas penjualan pada 2022 dibandingkan 2021. (Mohon juga menampilkan data ke-15 kategori)
1. Apabila memang terjadi penurunan 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

*#RintisKarirImpian*



# Pertanyaan 2:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

1. Memfilter data dengan valid = 1

**Code:** `valid = df['is_valid']==1`

1. Memfilter data pada variabel column 2021

**Code:** `trans_2021 = df['year']==2021`

1. Memfilter data pada variabel column 2022

**Code:** `trans_2022 = df['year']==2022`

1. Memfilter data pada variabel column Others

**Code:** `category = df['category']=='Others'`



# Pertanyaan 2:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

5. Menggabungkan filter untuk transaksi tahun 2022 dan 2021 untuk kategori Others

**Code:**

```
df_filter_2021 = df[category & valid & trans_2021]
df_filter_2022 = df[category & valid & trans_2022]
```

1. Menggabun

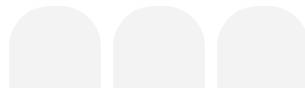
**Code:**

```
df_filter_category_2021 = df[valid & trans_2021]
df_filter_category_2022 = df[valid & trans_2022]
```

1. Groupby be

**Code:**

```
df_groupby_2021 = df_filter_2021.groupby(by=["sku_name"])[["qty_ordered"]].sum().sort_values(ascending=False).reset_index(name='qty_2021')
df_groupby_2022 = df_filter_2022.groupby(by=["sku_name"])[["qty_ordered"]].sum().sort_values(ascending=False).reset_index(name='qty_2022')
```





# Pertanyaan 2:



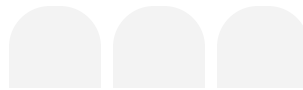
## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

7. Groupby berdasarkan category dan sort values

### Code:

```
df_groupby_category_2021 = df_filter_category_2021.groupby(by=["category"])[["qty_ordered"]].sum().sort_values(ascending=False).reset_index(name='qty_2021')
df_groupby_category_2022 = df_filter_category_2022.groupby(by=["category"])[["qty_ordered"]].sum().sort_values(ascending=False).reset_index(name='qty_2022')
```



# Pertanyaan 2:



## Answer 1.1:

### Code:

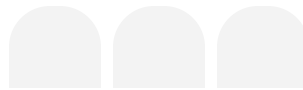
```
df_merge_others = df_groupby_2021.merge(df_groupby_2022, left_on = 'sku_name', right_on = 'sku_name')
df_merge_others
```

### Result:

	sku_name	qty_2021	qty_2022
0	RB_Dettol Germ Busting Kit-bf	200	45
1	Telemail_MM-DR-HB-L	23	2
2	kansai_NeverWet	10	1
3	emart_00-1	7	1
4	emart_Tyre Shape Air Compressor	5	34
5	MEGUIAR_G12711	4	1
6	Aladdin_bike_cover	3	2
7	kansai_Undercoating Aerosol	3	6
8	Entertainer Asia_Vouch 385-2017 Mobile App Lahore	2	1
9	Tmall_MM-DR-PAD	2	1
10	aw_Ultra Shine Wash & Wax-64oz./1893ml	2	1

### Description:

Dari hasil coding diatas, dapat terlihat product-product dalam kategori Others dan totl penjualan pada tahun 2021 dan 2022



# Pertanyaan 2:



## Answer 1.2:

### Code:

```
df_merge_category = df_groupby_category_2021.merge(df_groupby_category_2022, left_on = 'category', right_on = 'category')
df_merge_category
```

### Result:

	category	qty_2021	qty_2022
0	Soghaat	759	612
1	Others	426	263
2	Superstore	327	536
3	Men Fashion	237	175
4	Home & Living	193	250
5	School & Education	184	237
6	Health & Sports	173	200
7	Books	171	195

### Description:

Dari hasil coding diatas, dapat terlihat total penjualan dari tahun 2021 dan 2022 dari semua kategori.



# Pertanyaan 2:



## Answer 1.3:

**Code:**

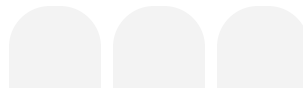
```
df_merge_category['pengurangan'] = df_merge_category['qty_2022'] - df_merge_category['qty_2021']
df_merge_category.sort_values(by=['pengurangan'], ascending=True, inplace=True)
df_merge_category
```

### Result:

	category	qty_2021	qty_2022	pengurangan
1	Others	426	263	-163
0	Soghaat	759	612	-147
3	Men Fashion	237	175	-62
9	Beauty & Grooming	168	153	-15
11	Appliances	124	148	24
7	Books	171	195	24
6	Health & Sports	173	200	27

### Description:

Dari hasil coding diatas, dapat terlihat bahwa kategori Others mengalami penurunan paling tinggi dalam penjualan antara tahun 2021 dan 2022



# Pertanyaan 2:



## Answer 2:

**Code:**

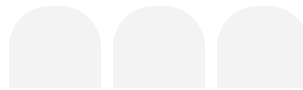
```
df_merge_others['pengurangan'] = df_merge_others['qty_2022'] - df_merge_others['qty_2021']
df_merge_others.sort_values(by=['pengurangan'], ascending=True, inplace=True)
df_merge_others
```

## Result:

	sku_name	qty_2021	qty_2022	pengurangan
0	RB_Dettol Germ Busting Kit-bf	200	45	-155
1	Telemall_MM-DR-HB-L	23	2	-21
2	kansai_NeverWet	10	1	-9
3	emart_00-1	7	1	-6
5	MEGUIAR_G12711	4	1	-3
10	aw_Ultra Shine Wash & Wax-64oz/1893ml	2	1	-1
8	Entertainer Asia_Vouch 365-2017 Mobile App Lahore	2	1	-1
9	Tmall_MM-DR-PAD	2	1	-1
6	Aladdin_bike_cover	3	2	-1

## Description:

Dari hasil coding diatas, dapat terlihat bahwa product RB\_Dettol Germ Busting Kit-bf pada kategori Others merupakan product yang mengalami penurunan paling drastis dari tahun 2021 ke 2022



# Pertanyaan 3:

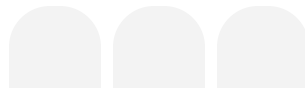


Terkait ulang tahun perusahaan pada 2 bulan mendatang, Tim Digital Marketing akan memberikan informasi promo bagi pelanggan pada akhir bulan ini. Kriteria pelanggan yang akan kami butuhkan adalah mereka yang sudah melakukan check-out namun belum melakukan pembayaran (is\_gross = 1) selama tahun 2022. 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



# Pertanyaan 3:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

1. Memfilter data dengan gross = 1

**Code:** `gross = df['is_gross']==1`

1. Memfilter c

**Code:** `valid = df['is_valid']==0`

1. Memfilter data dengan net = 0

**Code:** `net = df['is_net']==0`

1. Memfilter c

**Code:** `transaction = df['year']==2022`



# Pertanyaan 3:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

5. Menggabungkan filter untuk menampilkan ID Customer dan Registered Date

**Code:** `df_filter_promo = df[gross & valid & net & transaction]`





# Pertanyaan 3:



## Answer 3.1:

**Code:**

```
df_final = df_filter_promo[['customer_id', 'registered_date']]
df_final
```

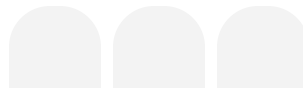
### Result:

	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
...	...	...
5856	C394076L	2021-10-12
5859	C248585L	2022-07-10
5865	C471304L	2022-05-13
5881	C265450L	2022-02-17
5883	C676393L	2021-07-27

1052 rows × 2 columns

### Description:

Dari hasil coding diatas, dapat disimpulkan bahwa sebanyak 1052 customers yang sudah melakukan check-out barang namun belum melakukan pembayaran selama tahun 2022.



# Pertanyaan 3:




## Answer 3.2:

### Code:

```
from google.colab import files
df_final.to_csv('audience_list.csv', encoding = 'utf-8-sig', index=False)
files.download('audience_list.csv')
```

### Result:

 audience\_list.csv  
20.6 KB • Done

### Description:

Dari hasil coding diatas, data yang sudah dicoding dapat di-download menjadi dokumen ber-ekstensi .csv



# Pertanyaan 4:



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:

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.

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



# Pertanyaan 4:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

1. Memfilter data dengan valid = 1

**Code:** `valid = df['is_valid']==1`

1. Memfilter data dengan day termasuk Saturday, Sunday dengan menggunakan fungsi isin dan simpan dalam variable 'weekends'

**Code:** `weekend = df["day"].isin(['Saturday', 'Sunday'])`

1. Memfilter data dengan day termasuk Monday, Tuesday, Wednesday, Thursday, Friday dengan menggunakan fungsi isin dan simpan dengan variabel 'weekday'

**Code:** `weekday = df["day"].isin(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'])`

1. Memfilter data pada tanggal Januari 2022 bulan Oktober sampai dengan Desember

**Code:** `tahun = df['year']==2022  
bulan = df["month"].isin(['October', 'November', 'December'])`



# Pertanyaan 4:



## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

5. Menggabungkan filter untuk menampilkan pencarian weekend dan weekday

**Code:**

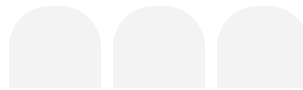
```
df_weekend = df[valid & weekend & tahun & bulan]
df_weekday = df[valid & weekday & tahun & bulan]
```

1. Groupby berdasarkan pada masing-masing variabel

**Code:**

```
df_groupby_weekend = df_weekend.groupby(by=["month_num", "month"])["before_discount"].mean()\
.round().sort_values(ascending=False).reset_index(name='avg_sales_weekend')

df_groupby_weekday = df_weekday.groupby(by=["month_num", "month"])["before_discount"].mean()\
.round().sort_values(ascending=False).reset_index(name='avg_sales_weekday')
```



# Pertanyaan 4:

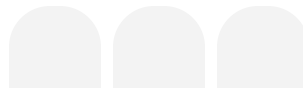


## Answer:

Untuk menjawab pertanyaan pertama, alangkah baiknya kita membuat beberapa kondisi yang sesuai dengan kondisi dari pertanyaan tersebut:

7. Menggabungkan kedua data dengan merge

**Code:** `df_merge = df_groupby_weekend.merge(df_groupby_weekday, left_on = 'month', right_on = 'month')`



# Pertanyaan 4:



## Answer 4.1:

**Code:** `df.groupby_weekend`

**Result:**

	month_num	month	avg_sales_weekend
0	10	October	634260.0
1	11	November	607794.0
2	12	December	410599.0

**Description:**

Dari hasil coding diatas, ditampilkan data penjualan Weekend pada bulan October, November dan Desember

**Code:** `df.groupby_weekday`

**Result:**

	month_num	month	avg_sales_weekday
0	10	October	874690.0
1	12	December	813574.0
2	11	November	641862.0

**Description:**

Dari hasil coding diatas, ditampilkan data penjualan Weekday pada bulan October, November dan Desember

# Pertanyaan 4:



## Answer 4.1:

### Code:

```
df_merge.sort_values(by='month_num_x',ascending=True, inplace=True)
df_merge = df_merge[["month", "avg_sales_weekend", "avg_sales_weekday"]]
df_merge
```

### Result:

	month	avg_sales_weekend	avg_sales_weekday
0	October	634260.0	874690.0
1	November	607794.0	641862.0
2	December	410599.0	813574.0

### Description:

Dari hasil coding diatas, dapat terlihat rata-rata penjualan weekend dan weekday pada bulan October, November dan December





# Pertanyaan 4:

## Answer 4.1:

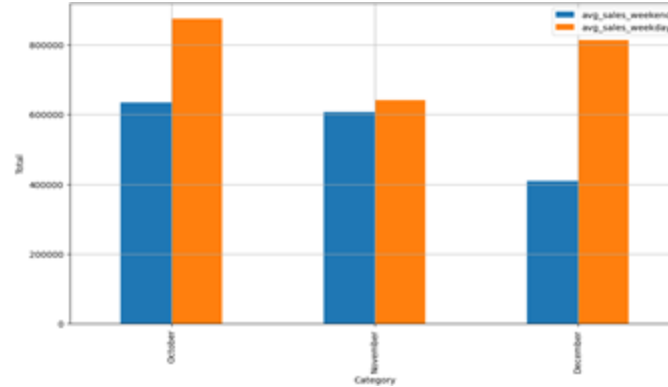
### Code:

```
df_merge.plot(x='month',  
              y=['avg_sales_weekend', 'avg_sales_weekday'],  
              kind='bar',  
              grid = True,  
              xlabel = 'Category',  
              ylabel = 'Total',  
              figsize=(12,7),  
              rot = 90,  
              table = False,  
              secondary_y = False)
```

### Description:

Dari hasil coding diatas, dapat melihat hasil perbandingan dari diagram batang

### Result:



# Pertanyaan 4:



## Answer 4.2:

### Code:

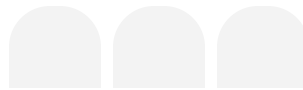
```
average_3_month = {'Periode': 'Total 3 months', \
                    'Avg Weekend': round(df_weekend['before_discount'].mean(), 2), \
                    'Avg Weekday': round(df_weekday['before_discount'].mean(), 2), \
                    'Diff (Value)': round(df_weekend['before_discount'].mean() - df_weekday['before_discount'].mean(), 2), \
                    'Diff (%)': pd.Series(round(((df_weekend['before_discount'].mean() - df_weekday['before_discount'].mean()) / df_weekend['before_discount'].mean()) * 100, 2), dtype=str) + '%'
                    }
pd.DataFrame(data=average_3_month, index=[0])
```

### Result:

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

### Description:

Dari hasil coding diatas, dapat terlihat rata-rata penjualan Weekend dan Weekday dalam 3 bulan (Oktober, November, dan Desember) serta hasil pengurangan dan dalam persen (%).



# Pertanyaan 4:



## Answer 4.2:

### Code:

```
df_merge['diff_value'] = df_groupby_weekend['avg_sales_weekend'] - df_groupby_weekday['avg_sales_weekday']
df_merge['diff_percent'] = pd.Series(round((df_merge['diff_value']/df_groupby_weekend['avg_sales_weekend'])*100),dtype=str) + '%'

df_merge_in_3_month = df_merge[["month","avg_sales_weekend","avg_sales_weekday","diff_value","diff_percent"]]
df_merge_in_3_month
```

### Result:

	month	avg_sales_weekend	avg_sales_weekday	diff_value	diff_percent
0	October	634260.0	874690.0	-240430.0	-38.0%
1	November	607794.0	641862.0	-205780.0	-34.0%
2	December	410599.0	813574.0	-231263.0	-56.0%

### Description:

Dari hasil coding diatas dapat terlihat rata-rata penjualan Weekday dan Weekend dalam bulan October, November, December serta pengurangan nilai dari Weekend dan Weekday dan dalam persen (%)

#RintisKarirImpian



# Pertanyaan 4:



## Answer 4.2:

### Code:

```
df_merge['diff_value'] = df_groupby_weekend['avg_sales_weekend'] - df_groupby_weekday['avg_sales_weekday']
df_merge['diff_percent'] = pd.Series(round((df_merge['diff_value']/df_groupby_weekend['avg_sales_weekend'])*100),dtype=str) + '%'

df_merge_in_3_month = df_merge[["month","avg_sales_weekend","avg_sales_weekday","diff_value","diff_percent"]]
df_merge_in_3_month
```

### Result:

	month	avg_sales_weekend	avg_sales_weekday	diff_value	diff_percent
0	October	634260.0	874690.0	-240430.0	-38.0%
1	November	607794.0	641862.0	-205780.0	-34.0%
2	December	410599.0	813574.0	-231263.0	-56.0%

### Description:

Dari hasil coding diatas dapat terlihat rata-rata penjualan Weekday dan Weekend dalam bulan October, November, December serta pengurangan nilai dari Weekend dan Weekday dan dalam persen (%)

#RintisKarirImpian





MySkill | *#RintisKarirImpian*

Portfolio - Intensive Bootcamp

# Thank You!