Laporan Praktikum Pertemuan 4

Data Science Lanjut

Data Visualization With Python Matplotlib For Beginner

Dibuat oleh

Nama	: Muhamad faisal halim
Nim	: 19.240.0163
Kelas	: -
Mata kuliah	: Data science lanjut

Mahasiswa pertukaran mahasiswa.

Universitas muhammadiyah kalimantan timur

~ stmik widya pratama pekalongan

Note

Data pada praktikum ini disamakan dengan data yang ada pada contoh yang diberikan di openlearning umkt.

Materi dan praktikum

TITLE : Data Visualization With Python Matplotlib For Beginner

Data Visualization Python

Visualisasi data adalah upaya untuk memahami data dengan menempatkannya dalam konteks visual sehingga pola, tren, dan korelasi yang mungkin tidak terdeteksi dapat diekspos.

Python menawarkan beberapa library grafik hebat yang dikemas dengan banyak fitur berbeda. Tidak masalah jika Anda ingin membuat plot interaktif, langsung, atau sangat disesuaikan, python memiliki perlibraryan yang sangat baik untuk Anda.

Untuk mendapatkan sedikit gambaran, berikut adalah beberapa library plot yang populer:

- **Matplotlib**: level rendah, memberikan banyak kebebasan
- **Pandas Visualization**: antarmuka yang mudah digunakan, dibangun di atas Matplotlib
- **Seaborn**: antarmuka tingkat tinggi, gaya default yang bagus
- **ggplot**: berdasarkan ggplot2 R, menggunakan Tata Bahasa Grafik
- **Plotly**: dapat membuat plot interaktif

Dataset

Dataset/Himpunan Data/Data Latih adalah sebuah himpunan data yang berasal dari informasi masa-masa lampau dan dikelola menjadi sebuah informasi untuk melakukan teknik dari ilmu **data mining.**

Importing Dataset

Sebelum melakukan praktik kita harus mengimport dulu library dan dataset yang akan kita gunakan dalam praktik ini.

```
import pandas as pd

dataset = pd.read_csv('https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/retail_raw_reduced.csv')

print('Ukuran dataset: %d baris dan %d kolom\n' % dataset.shape)
print('Lima data teratas:')
print(dataset.head())
```

✓ 1.3s

```
Ukuran dataset: 5000 baris dan 9 kolom

Lima data teratas:
   order_id  order_date  customer_id  ...     brand quantity item_price
0   1703458  2019-10-17        14004  ...   BRAND_J       10     740000
1   1706815  2019-10-24        17220  ...   BRAND_R        2     604000
2   1710718  2019-11-03        16518  ...   BRAND_C        8    1045000
3   1683592  2019-08-19        16364  ...   BRAND_A        4     205000
4   1702573  2019-10-16        15696  ...   BRAND_R        2    4475000

[5 rows x 9 columns]
```

Kita juga bisa menambahkan kolom baru kedalam dataset yang kita insertkan diatas. Dengan cara berikut. Dalam contoh ini menambahkan kolom order_month.

```
import datetime

dataset['order_month'] = dataset['order_date'].apply(lambda x: datetime.datetime.strptime(x, "%Y-%m-%d").strftime('%Y-%m'))
print(dataset.head())
```

✓ 0.5s

```
   order_id  order_date  customer_id  ... quantity item_price order_month
0   1703458  2019-10-17        14004  ...       10     740000     2019-10
1   1706815  2019-10-24        17220  ...        2     604000     2019-10
2   1710718  2019-11-03        16518  ...        8    1045000     2019-11
3   1683592  2019-08-19        16364  ...        4     205000     2019-08
4   1702573  2019-10-16        15696  ...        2    4475000     2019-10

[5 rows x 10 columns]
```

Contoh lain penambaha kolom GMV, Gross Marchendise Value adalah istilah yang digunakan dalam ritel online untuk menunjukkan total nilai uang penjualan untuk barang dagangan yang dijual melalui pasar tertentu selama jangka waktu tertentu.

```
dataset['gmv'] = dataset['item_price'] * dataset['quantity']

print(dataset.head())
```

✓ 0.7s

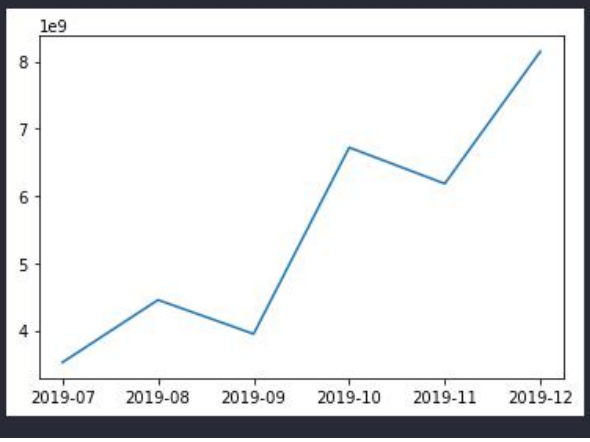
```
   order_id  order_date  customer_id  ... item_price order_month      gmv
0   1703458  2019-10-17        14004  ...     740000     2019-10  7400000
1   1706815  2019-10-24        17220  ...     604000     2019-10  1208000
2   1710718  2019-11-03        16518  ...    1045000     2019-11  8360000
3   1683592  2019-08-19        16364  ...     205000     2019-08   820000
4   1702573  2019-10-16        15696  ...    4475000     2019-10  8950000

[5 rows x 11 columns]
```

Plot Pertama : Trend Pertumbuhan GMV

Cara standar untuk menggunakan matplotlib yaitu dengan memanggil function plt.plot lalu definisikan nilai di sumbu-x dan sumbu-y. Dalam hal ini, definisikan kolom order_month di sumbu-x (parameter pertama), dan kolom gmv di sumbu-y (parameter kedua). Setelah selesai mendefinisikan komponen chart-nya, lalu panggil plt.show()untuk menampilkan grafiknya.

```
import matplotlib.pyplot as plt

monthly_amount = dataset.groupby('order_month')['gmv'].sum().reset_index()
plt.plot(monthly_amount['order_month'], monthly_amount['gmv'])
plt.show()
```

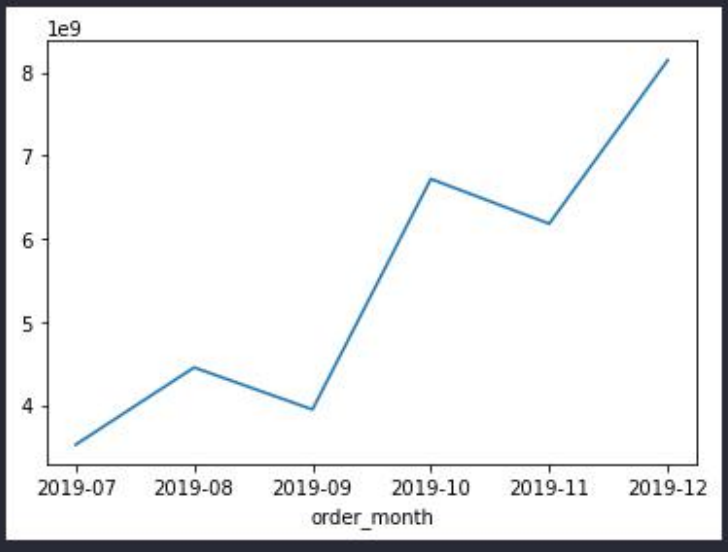
✓ 2.3s



Cara Pendekatan lain dengan menggunakan fungsi .plot()

```
import matplotlib.pyplot as plt
dataset.groupby(['order_month'])['gmv'].sum().plot()
plt.show()
```

✓ 0.7s



Dan masih banyak sekali yang dapat dilakukan oleh matplotlib ini.

Mengubah figure size	`plt.figure(figsize=(15,5))`

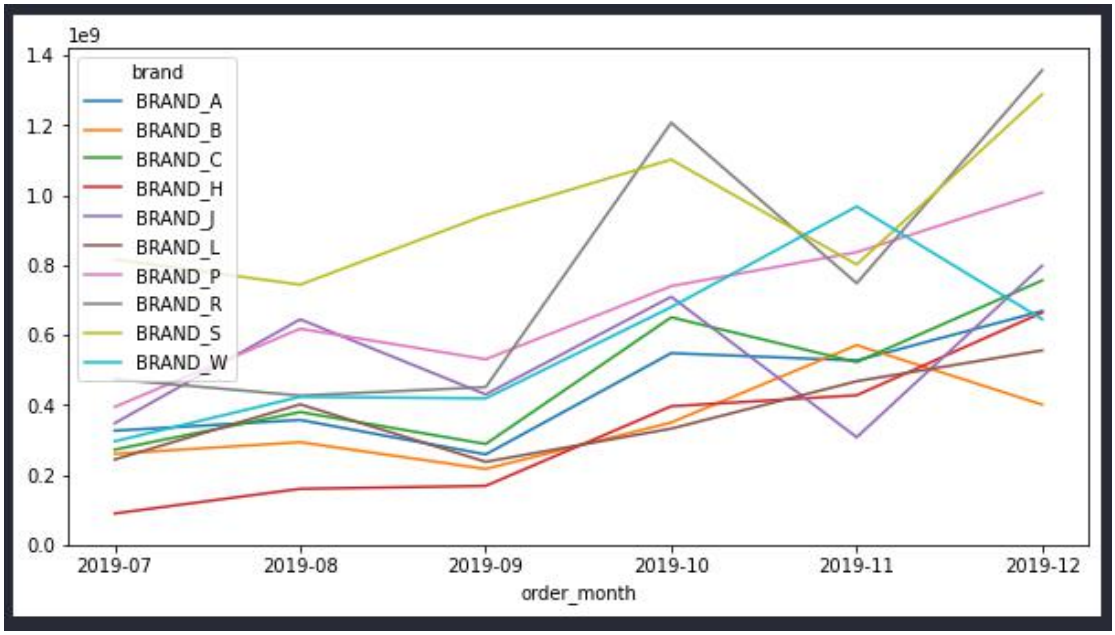
Menambah title dan axis label	`plt.title('Monthly GMV Year 2019')`
	`plt.xlabel('Order Month')`
	`plt.ylabel('Total GMV')`

Custom title dan axis label	`plt.title('Monthly GMV Year 2019', loc='center', pad=40, fontsize=20, color='blue')`
	`plt.xlabel('Order Month', fontsize=15)`
	`plt.ylabel('Total Amount', fontsize=15)`

Custom line point	`dataset.groupby(['order_month'])['gmv'].sum().plot(color='green', marker='o', linestyle='-.', linewidth=2)`
Custom grid	`plt.grid(color='black', linestyle='dotted', linewidth=0.5)`
Custom axis ticks	`labels, locations = plt.yticks()` `plt.yticks(labels, (labels/1000000000).astype(int))`
Menentukan batas minimum dan maksimum axis ticks	`plt.ylim(ymin=0)`
Menambah informasi pada plot	`plt.text(0.45,0.72, 'The GMV increased significantly on October 2019 ', transform=fig.transFigure, color='red')`
Menyimpan hasil plot kedalam file image	`plt.savefig('monthly_gmv.png')`
Custom kualitas menyimpan gambar	`plt.savefig('monthly_gmv.png', quality=95)`

Membuat Multi Line chart

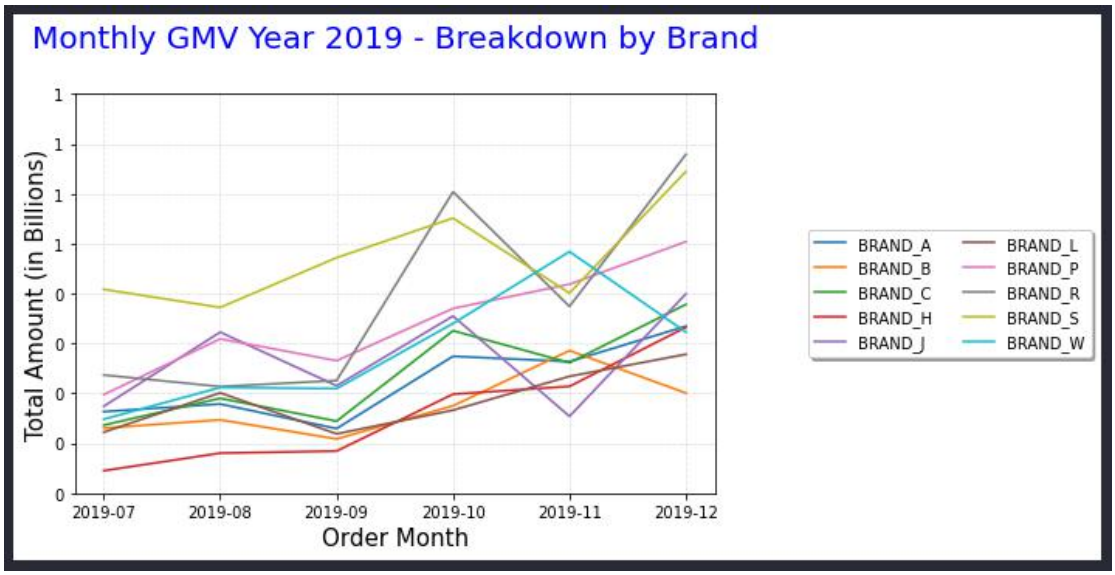
```
import datetime
import pandas as pd
import matplotlib.pyplot as plt

dataset = pd.read_csv('https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/retail_raw_reduced.csv')

dataset['order_month'] = dataset['order_date'].apply(lambda x: datetime.datetime.strptime(x, "%Y-%m-%d").strftime('%Y-%m'))
dataset['gmv'] = dataset['item_price']*dataset['quantity']

dataset.groupby(['order_month','brand'])['gmv'].sum().unstack().plot()
plt.gcf().set_size_inches(10, 5)
plt.ylim(ymin=0)
plt.show()
✓ 2.8s
```



Custom Legends

```
import matplotlib.pyplot as plt
dataset.groupby(['order_month','brand'])['gmv'].sum().unstack().plot()
plt.ylim(ymin=0)
plt.legend(loc='right',bbox_to_anchor=(1.25, 0.5), shadow=True, ncol=2)
plt.gcf().set_size_inches(13, 5)
plt.tight_layout()
plt.show()
```

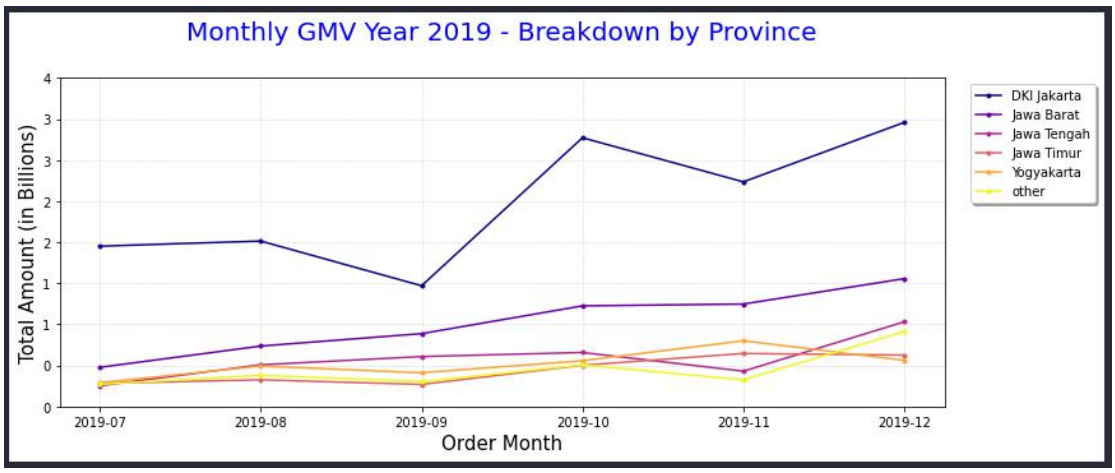
✓ 1.4s



Custom color map dan GMV by top Provinces

Untuk custom color map bisa menggunakan cmap='…' pada .plot() selengkapnya bisalihat kode dibawah

```
import matplotlib.pyplot as plt

top_provinces = (dataset.groupby('province')['gmv'].sum().reset_index().sort_values(by='gmv',ascending=False).head(5))

dataset['province_top'] = dataset['province'].apply(lambda x: x if (x in top_provinces['province'].to_list()) else 'other')
dataset.groupby(['order_month','province_top'])['gmv'].sum().unstack().plot(marker='.', cmap='plasma')

plt.title('Monthly GMV Year 2019 - Breakdown by Province',loc='center',pad=30, fontsize=20, color='blue')
plt.xlabel('Order Month', fontsize = 15)
plt.ylabel('Total Amount (in Billions)',fontsize = 15)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.ylim(ymin=0)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000000000).astype(int))
plt.legend(loc='upper center', bbox_to_anchor=(1.1, 1), shadow=True, ncol=1)
plt.gcf().set_size_inches(12, 5)
plt.tight_layout()
plt.show()
```

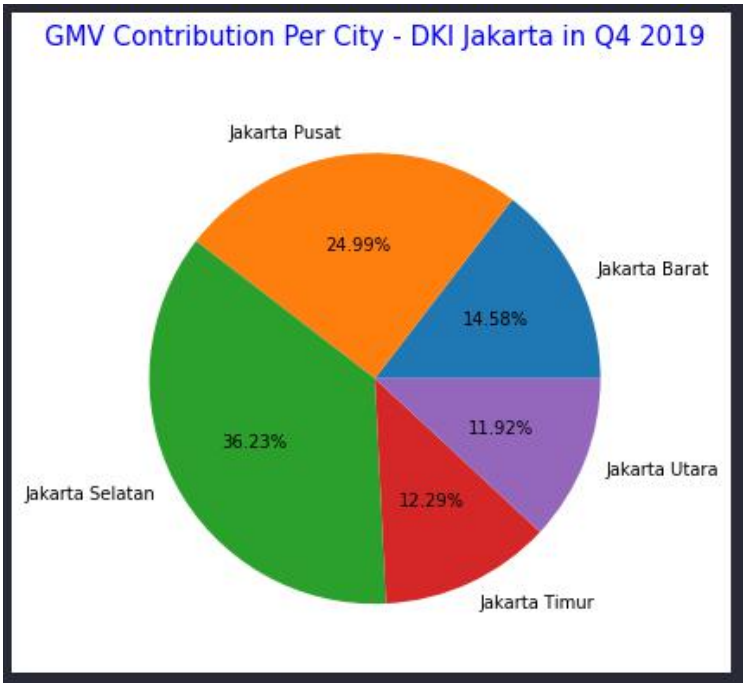


Pie Chart

Beberapa parameter yang bisa dimodifikasi:

- **labels**: array yang berisikan label/tulisan yang ditunjukkan untuk masing-masing bagian pie.
- **colors**: array yang berisikan warna untuk masing-masing bagian pie.
- **autopct**: format untuk nilai persentasi yang ditampilkan, bisa berupa string atau function.
- **shadow**: jika diisi True, maka ada bayangan untuk pie chart-nya. Defaultnya adalah False.
- **radius**: jari-jari dari pie-chart

```
import matplotlib.pyplot as plt

dataset_dki_q4 = dataset[(dataset['province']=='DKI Jakarta') & (dataset['order_month'] ≥ '2019-10')]
gmv_per_city_dki_q4 = dataset_dki_q4.groupby('city')['gmv'].sum().reset_index()

plt.figure(figsize=(6,6), facecolor='white')
plt.pie(gmv_per_city_dki_q4['gmv'], labels = gmv_per_city_dki_q4['city'],autopct='%1.2f%%')
plt.title('GMV Contribution Per City - DKI Jakarta in Q4 2019',loc='center',pad=30, fontsize=15, color='blue')
plt.show()
```

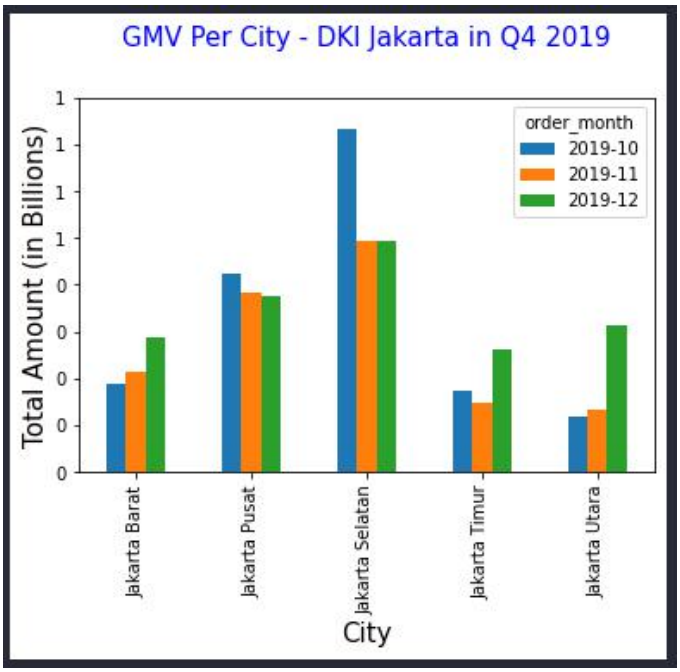


Bar Chart

```
import matplotlib.pyplot as plt

plt.clf()

dataset_dki_q4.groupby('city')['gmv'].sum().sort_values(ascending=False).plot(kind='bar', color='green')
labels, locations = plt.yticks()

plt.title('GMV Per City - DKI Jakarta in Q4 2019',loc='center',pad=30, fontsize=15, color='blue')
plt.xlabel('City', fontsize = 15)
plt.ylabel('Total Amount (in Billions)',fontsize = 15)
plt.ylim(ymin=0)
plt.yticks(labels, (labels/1000000000).astype(int))
plt.xticks(rotation=90)
plt.show()
```

Multi Bar Chart

```
import matplotlib.pyplot as plt

plt.clf()

dataset_dki_q4.groupby(['city','order_month'])['gmv'].sum().sort_values(ascending=False).unstack().plot(kind='bar')
labels, locations = plt.yticks()

plt.title('GMV Per City - DKI Jakarta in Q4 2019',loc='center',pad=30, fontsize=15, color='blue')
plt.xlabel('City', fontsize = 15)
plt.ylabel('Total Amount (in Billions)',fontsize = 15)
plt.ylim(ymin=0)
plt.yticks(labels, (labels/1000000000).astype(int))
plt.xticks(rotation=90)
plt.show()
```



Selain dari chart yang sudah dipraktikan diatas kita dapat membuat chart lain misal

1. Histogram
2. Scatterplot