

Visualisasi Data dengan Python Matplotlib

Topik

- Pengenalan Matplotlib dan Persiapan Dataset
 - Pengenalan Matplotlib
 - Pengenalan Dataset
 - Penjelasan Dataset
 - Penambahan Kolom Order Month pada Dataset
 - Penambahan Kolom GMV pada Dataset
- Plot Pertama dengan Matplotlib
 - Membuat Data Agregat
 - Plot Pertama: Membuat Line Chart Trend Pertumbuhan GMV
 - Cara Alternatif: Fungsi `.plot()` pada pandas Dataframe
- Kustomisasi Grafik
 - Anatomi dari Figure
 - Mengubah Figure Size
 - Menambahkan Title and Axis Labels
 - Kustomisasi Title and Axis Labels
 - Kustomisasi Line dan Point
 - Kustomisasi Grid
 - Kustomisasi Axis Ticks
 - Menentukan Batas Minimum dan Maksimum Axis Ticks
 - Menambahkan Informasi Pada Plot
 - Menyimpan Hasil Plot Menjadi File Image
 - Pengaturan Parameter untuk Menyimpan Gambar

Pengenalan Matplotlib

- Dalam menganalisa data, sering kali menggunakan visualisasi data untuk memahami karakteristik maupun trend di dalam data yang dimiliki.
- Salah satu library di python yang biasa digunakan untuk keperluan visualisasi data adalah matplotlib.

Hal yang dilakukan di modul ini:

- Pengolahan dataset (menggunakan library NumPy & Pandas).
- Membuat grafik dasar menggunakan matplotlib.
- Melakukan modifikasi komponen visualisasi, seperti:
 - axis
 - labels
 - title
 - legend
- Menyimpan plot visualisasi yang sudah dibuat

Pengenalan Dataset

- Gunakan dataset yang telah disediakan: retail_raw_reduced.csv.
- Tampilkan dataset nya dengan menggunakan kode berikut:

```
import pandas as pd
dataset = pd.read_csv('retail_raw_reduced.csv')
print('Ukuran dataset: %d baris dan %d kolom\n' % dataset.shape)
print('Lima data teratas:')
print(dataset.head())
```

output:

Ukuran dataset: 5000 baris dan 9 kolom

Lima data teratas:

	order_id	order_date	customer_id	city	province	product_id	\
0	1703458	2019-10-17	14004	Jakarta Selatan	DKI Jakarta	P1910	
1	1706815	2019-10-24	17220	Jakarta Selatan	DKI Jakarta	P2934	
2	1710718	2019-11-03	16518	Jakarta Utara	DKI Jakarta	P0908	
3	1683592	2019-08-19	16364	Jakarta Barat	DKI Jakarta	P0128	
4	1702573	2019-10-16	15696	Jakarta Timur	DKI Jakarta	P2968	

	brand	quantity	item_price
0	BRAND_J	10	740000
1	BRAND_R	2	604000
2	BRAND_C	8	1045000
3	BRAND_A	4	205000
4	BRAND_R	2	4475000

Penjelasan Dataset

- **order_id** : ID dari order/transaksi, 1 transaksi bisa terdiri dari beberapa produk, tetapi hanya dilakukan oleh 1 customer
- **order_date** : tanggal terjadinya transaksi
- **customer_id** : ID dari pembeli, bisa jadi dalam satu hari, 1 customer melakukan transaksi beberapa kali
- **city** : kota tempat toko terjadinya transaksi
- **province** : provinsi (berdasarkan city)
- **product_id** : ID dari suatu product yang dibeli
- **brand** : brand/merk dari product. Suatu product yang sama pasti memiliki brand yang sama
- **quantity** : Kuantitas / banyaknya product yang dibeli
- **item_price** : Harga dari 1 product (dalam Rupiah). Suatu product yang sama, bisa jadi memiliki harga yang berbeda saat dibeli.
- Catatan: Dalam dataset **retail_raw.csv**, satu **order_id** pasti hanya memiliki satu **customer_id**

Penambahan Kolom Order Month pada Dataset

- Untuk membuat grafik Gross Merchandise Value (GMV) bulanan, bayangan kita tentu adalah sebuah line chart dengan sumbu-x berupa bulan, dan sumbu-y adalah total GMV di bulan tersebut.
- Oleh karena itu, kita akan menambahkan kolom order month pada dataset.
- Dengan menggunakan kolom order date yang ada pada dataset, kita bisa membuat kolom order month dengan cara berikut :

Setelah dijalankan kode akan menambahkan kolom order month pada dataset

```
In [3]: import datetime
dataset['order_month'] = dataset['order_date'].apply(lambda x:
datetime.datetime.strptime(x, "%Y-%m-%d").strftime('%Y-%m'))
print(dataset.head())
```

	order_id	order_date	customer_id	city	province	product_id	\
0	1703458	2019-10-17	14004	Jakarta Selatan	DKI Jakarta	P1910	
1	1706815	2019-10-24	17220	Jakarta Selatan	DKI Jakarta	P2934	
2	1710718	2019-11-03	16518	Jakarta Utara	DKI Jakarta	P0908	
3	1683592	2019-08-19	16364	Jakarta Barat	DKI Jakarta	P0128	
4	1702573	2019-10-16	15696	Jakarta Timur	DKI Jakarta	P2968	

	brand	quantity	item_price	order_month
0	BRAND_J	10	740000	2019-10
1	BRAND_R	2	604000	2019-10
2	BRAND_C	8	1045000	2019-11
3	BRAND_A	4	205000	2019-08
4	BRAND_R	2	4475000	2019-10

Ada beberapa function yang digunakan:

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

- Apply & lambda biasa digunakan untuk membuat kolom baru, berdasarkan suatu kolom lain yang sudah ada (misal `.apply(lambda x: x*2)` berarti setiap input `x` di dalam kolom, akan diubah menjadi `x*2`). Dalam hal ini kolom yang sudah ada adalah `dataset['order_date']`, lalu tiap nilai di dalamnya kita proses agar menjadi month-nya saja.
- Function `datetime.datetime.strptime` digunakan untuk mengubah date/time dalam bentuk string menjadi tipe data `datetime`.
- Function `strftime` digunakan untuk mengubah format suatu data bertipe `datetime`, dalam hal ini diubah menjadi `'%Y-%m'`, yang berarti outputnya adalah waktu dengan bentuk YYYY-MM atau tahun dan bulan saja, tanggalnya sudah tidak ada.

Penambahan Kolom GMV pada Dataset

- Selanjutnya, bagaimana dengan GMV?
- Definisikan GMV sebagai perkalian(*) setiap entri di kolom item_price dan kolom quantity.
- Bisa menggunakan fungsi apply & lambda seperti sebelumnya, tetapi ada cara yang lebih mudah yaitu dengan mengalikan kedua kolom yang disebutkan secara langsung.

Kode Penambahan Kolom GMV pada Dataset

```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
+ < > Run Code
In [4]: dataset['gmv'] = dataset['item_price']*dataset['quantity']
print('Ukuran dataset: %d baris dan %d kolom\n' % dataset.shape)
print('Lima data teratas:')
print(dataset.head())

Ukuran dataset: 5000 baris dan 11 kolom

Lima data teratas:
  order_id order_date customer_id city province product_id \
0  1703458  2019-10-17      14004 Jakarta Selatan DKI Jakarta P1910
1  1706815  2019-10-24      17220 Jakarta Selatan DKI Jakarta P2934
2  1710718  2019-11-03      16518 Jakarta Utara DKI Jakarta P0908
3  1683592  2019-08-19      16364 Jakarta Barat DKI Jakarta P0128
4  1702573  2019-10-16      15696 Jakarta Timur DKI Jakarta P2968

  brand quantity item_price order_month gmv
0  BRAND_J      10     740000    2019-10  7400000
1  BRAND_R       2     604000    2019-10  1208000
2  BRAND_C       8    1045000    2019-11  8360000
3  BRAND_A       4     205000    2019-08   820000
4  BRAND_R       2    4475000    2019-10  8950000
```

Sekarang pada dataframe terdapat penambahan kolom **order_month** dan **gmv**.

Membuat Data Agregat

- Dengan menggunakan fungsi **groupby** dan **sum**, akan menampilkan total gmv dari tiap bulan.
- Berikut code dan outputnya :



The screenshot shows a Jupyter Notebook interface. At the top, there is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. On the right side of the menu bar, there is a 'Trusted' button and a 'Python 3' indicator. Below the menu bar is a toolbar with various icons for file operations, navigation, and execution. The main area of the notebook contains a code cell with the following Python code:

```
In [5]: monthly_amount = dataset.groupby('order_month')['gmv'].sum().reset_index()
print(monthly_amount)
```

The output of the code is a table with two columns: 'order_month' and 'gmv'. The table contains six rows of data, indexed from 0 to 5.

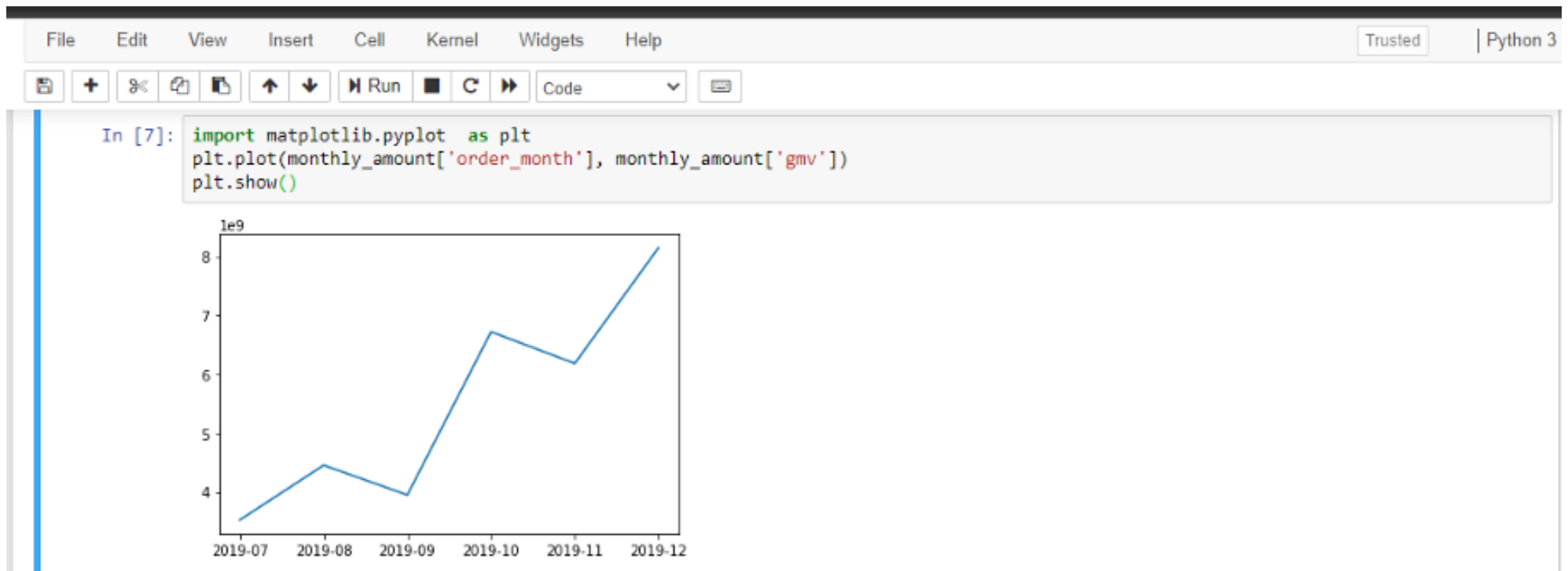
	order_month	gmv
0	2019-07	3524041000
1	2019-08	4452923000
2	2019-09	3947002000
3	2019-10	6719937000
4	2019-11	6182229000
5	2019-12	8148235000

Plot Pertama:

Membuat Line Chart Trend Pertumbuhan GMV

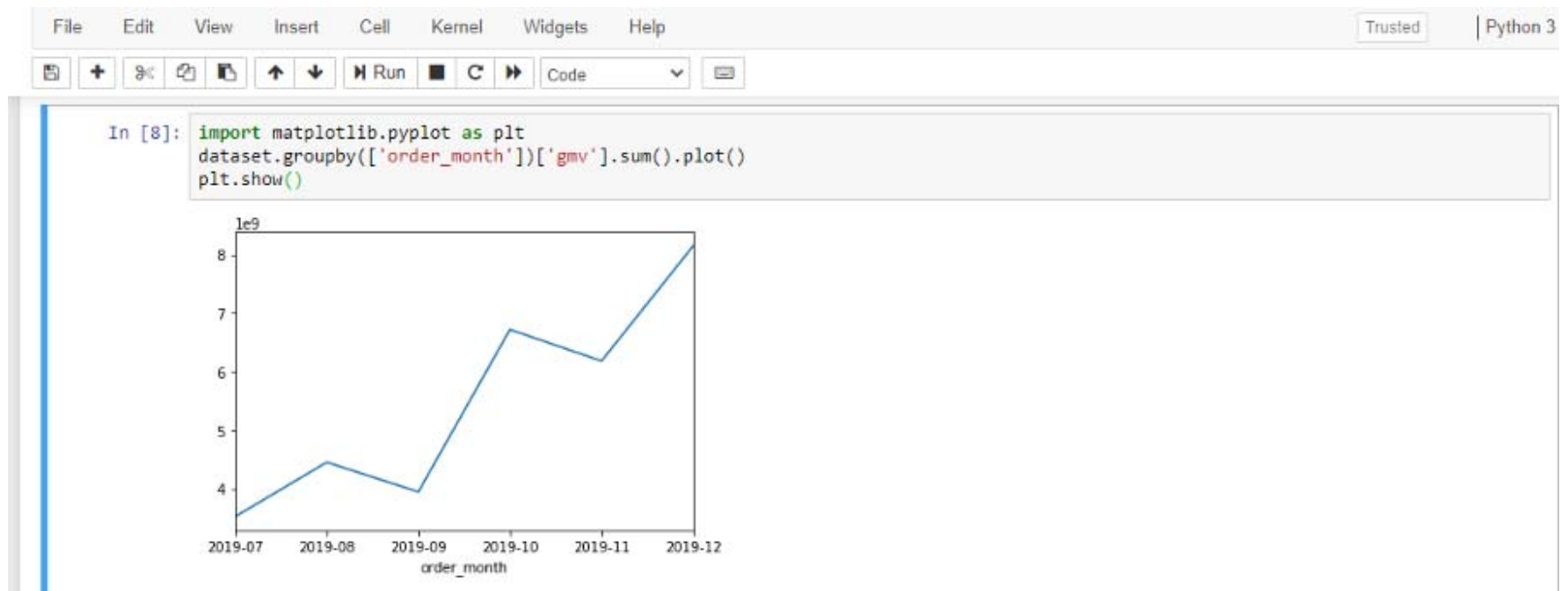
- Untuk menggunakan matplotlib, library harus dipanggil terlebih dahulu dengan menggunakan fungsi **import**.
- Cara standar untuk menggunakan matplotlib yaitu dengan memanggil function `plt.plot` lalu definisikan nilai di sumbu-x dan sumbu-y.
- Dalam hal ini, definisikan kolom `order_month` di sumbu-x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

Berikut Kode beserta grafik yang ditampilkan

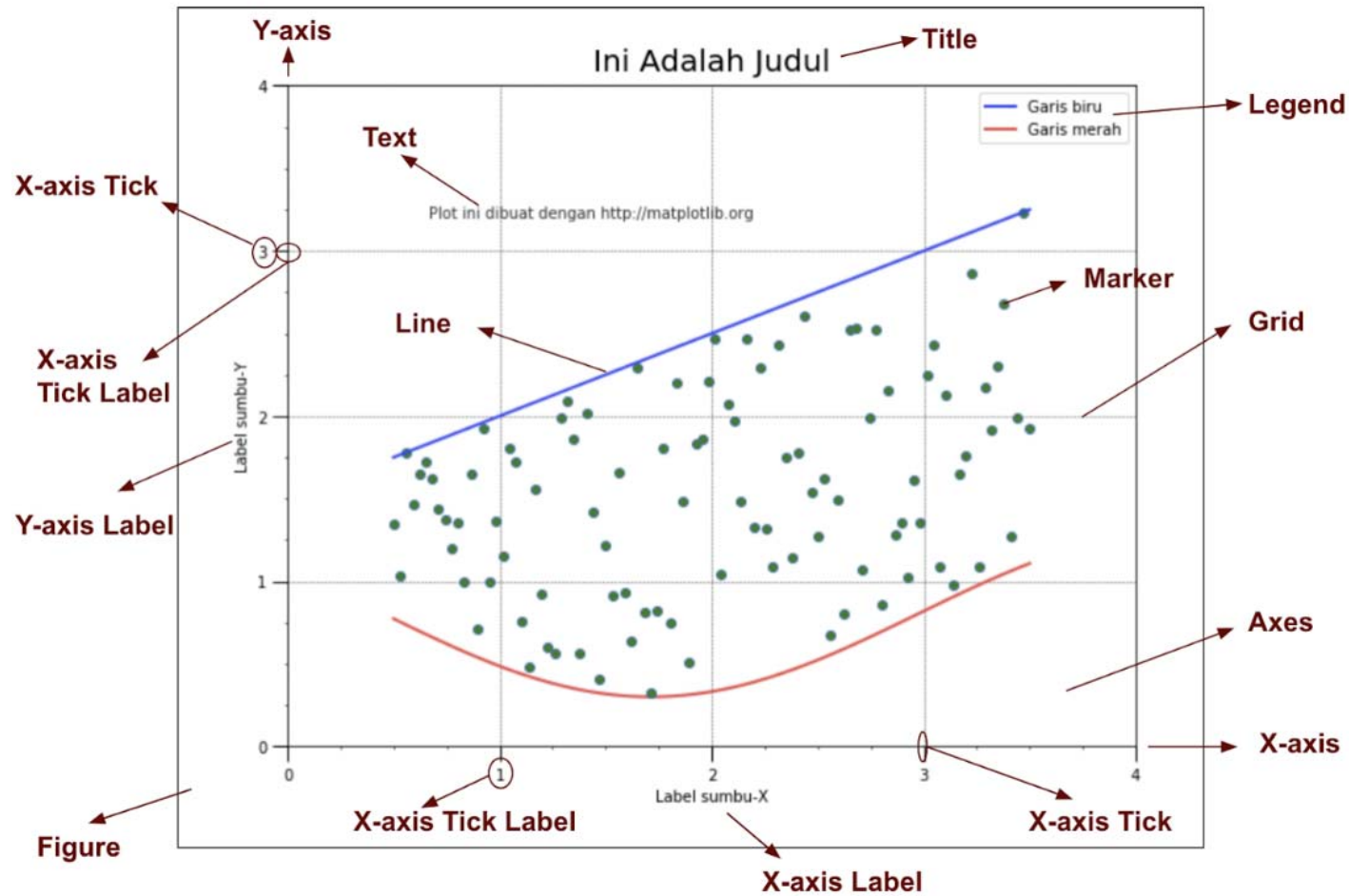


Cara Alternatif: Fungsi .plot() pada pandas Dataframe

- Dengan pendekatan lain, dimungkinkan juga melakukan visualisasi dengan matplotlib tanpa membuat variabel baru.
- Misalnya dengan code seperti ini: Grafik yang dihasilkan serupa.



Anatomi dari Figure

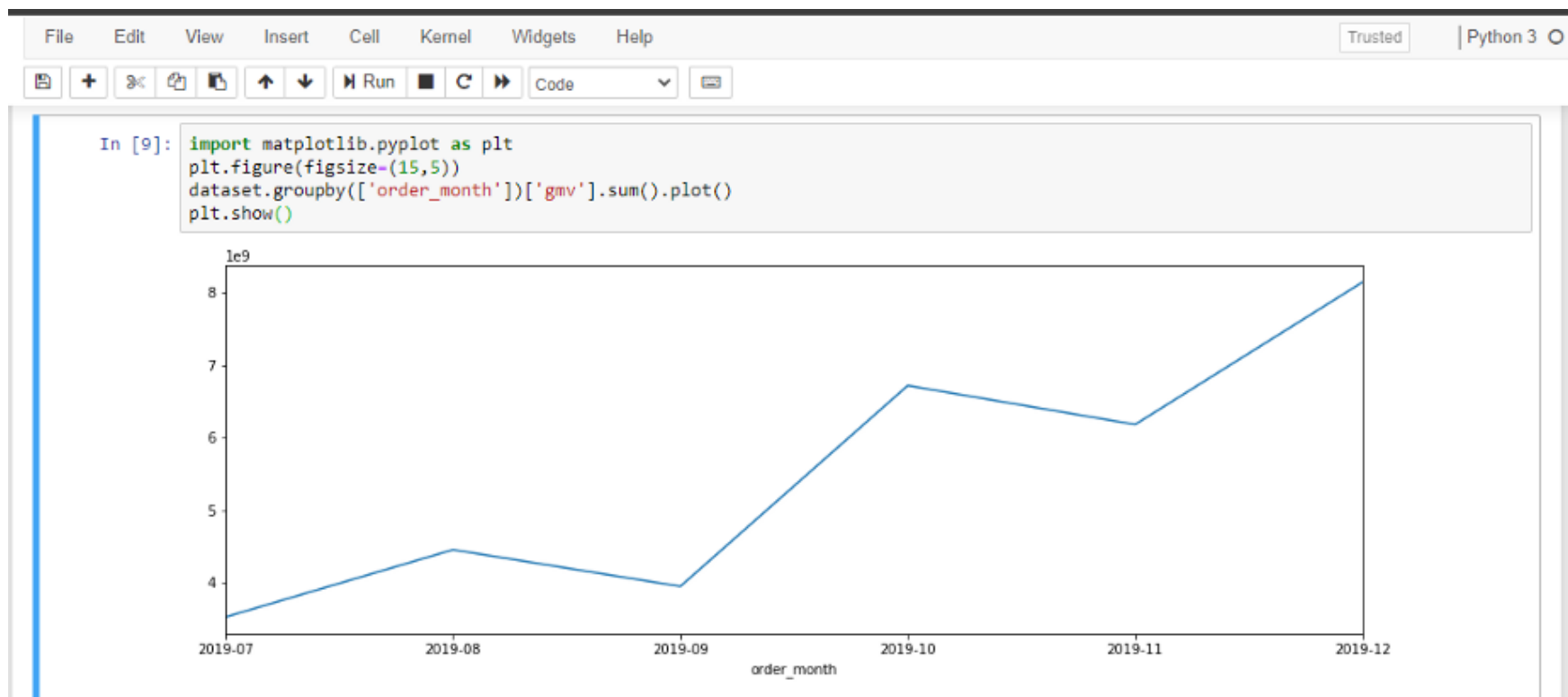


Istilah dan komponen di sebuah plot

- **Figure** adalah keseluruhan visualisasi yang kita plot dalam 1 kali menjalankan code.
- Sedangkan satu plot (yang dibuat saat memanggil `.plot()` atau `.scatter()`) disebut **Axes**. Sebuah Figure bisa terdiri dari beberapa Axes.
- Setiap Axes biasanya memiliki sumbu-X (**X-axis**) dan sumbu-Y (**Y-axis**). Masing-masing sumbu memiliki komponen sebagai berikut:
- **Axis Label**: Nama dari sumbu yang ditampilkan.
- **Tick**: Penanda berupa titik/garis kecil yang berjajar di sumbu, sebagai referensi skala nilai.
- **Tick Label**: Tulisan di tiap tick yang menyatakan nilainya.
- Untuk isi grafiknya sendiri, bisa berupa **line** (untuk line plot), atau **marker** (untuk scatter plot), bisa juga bentuk lain seperti **bar** (untuk bar plot / histogram).
- Aksesoris lain yang bisa ditambahkan, di antaranya **Grid** untuk mempermudah melihat tick yang sejajar, dan **Text** untuk memberikan informasi tambahan berbentuk teks di grafik.
- Untuk anatomi yang lebih lengkap, bisa dilihat di link berikut: <https://matplotlib.org/3.1.3/gallery/showcase/anatomy.html>

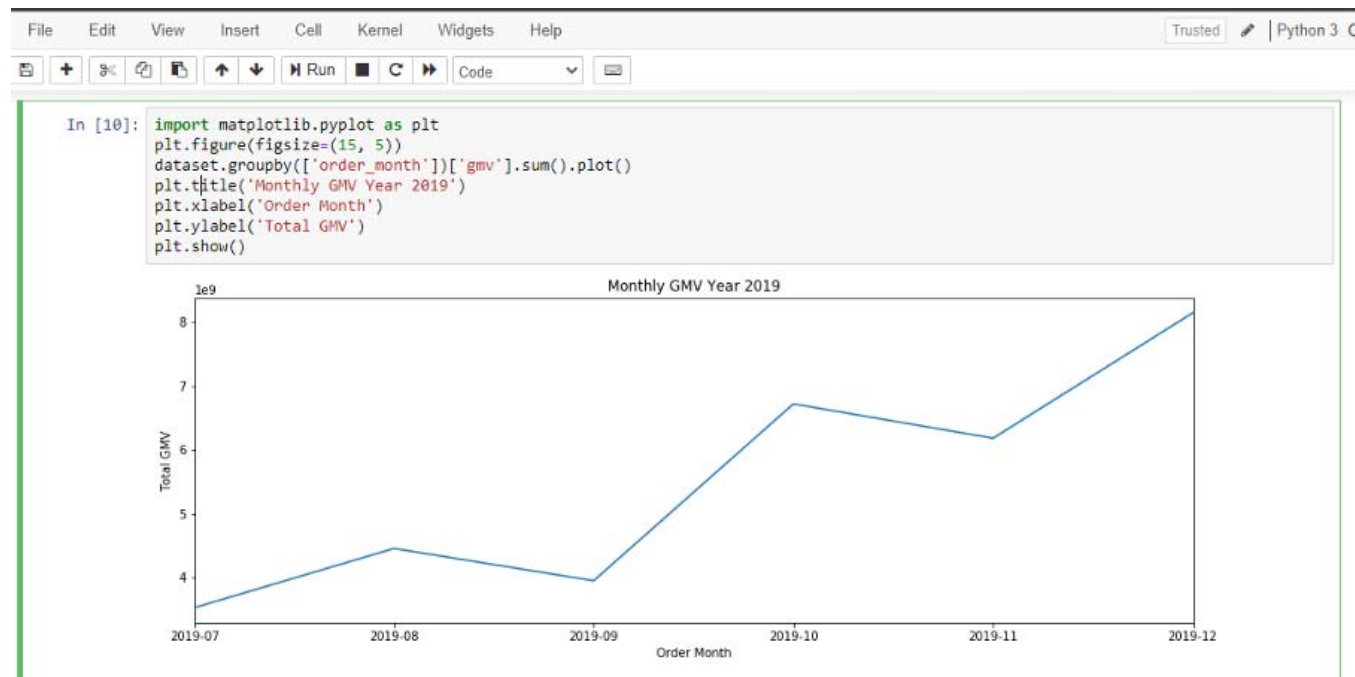
Mengubah Figure Size

- Mengganti ukuran dari figure sesuai kebutuhan dengan menggunakan fungsi **figsize**.
- Mengganti ukuran dari figure nya sehingga tampilan grafik tampak lebih lebar.



Menambahkan Title and Axis Labels

- Beberapa hal yang lazim dimodifikasi pada sebuah grafik adalah judul (title) dan label di sumbu-x & sumbu-y (axis labels).
- Untuk menambahkannya, tambahkan `plt.title`, `plt.xlabel`, dan `plt.ylabel` di code sebelum `plt.show()`.
- Tampak grafik sudah ditambahkan judul(title). Kemudian ditambahkan juga label pada sumbu-x dan sumbu-y.



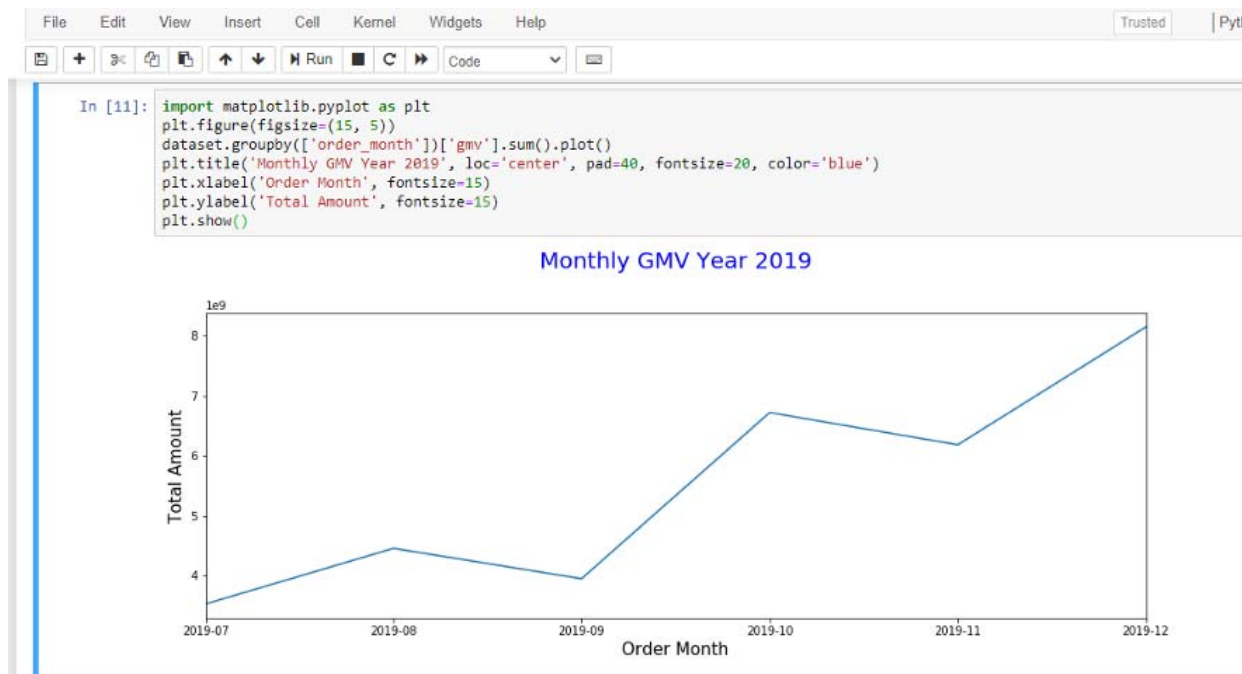
Kustomisasi Title and Axis Labels

- Kita bisa melakukan berbagai modifikasi dengan menambahkan parameter ke dalam function-nya.

Misalnya, untuk judul/title, parameter yang bisa ditambahkan:

- **loc**: digunakan untuk menentukan posisi title, misalnya 'left' untuk membuat rata kiri, 'right' untuk rata kanan, dan 'center' untuk meletakkannya di tengah. Jika tidak didefinisikan, maka defaultnya title ada di tengah.
 - **pad**: digunakan untuk menambahkan jarak antara judul ke grafik (dalam satuan px), misalnya kita tidak ingin judulnya terlalu menempel dengan grafiknya, jadi kita beri jarak.
 - **fontsize**: digunakan untuk mengganti ukuran font/huruf (dalam satuan px).
 - **color**: digunakan untuk mengganti warna huruf judul. Kita bisa menggunakan warna dasar dengan kata seperti 'blue', 'red', 'orange', dsb. Bisa juga dengan hex string, misalnya '#42DDF5' untuk warna biru muda.
- Untuk xlabel dan ylabel, kita bisa mengganti fontsize dan color, tetapi tidak bisa mengganti loc.

Modifikasi judul, xlabel, dan ylabel dengan parameter-parameter



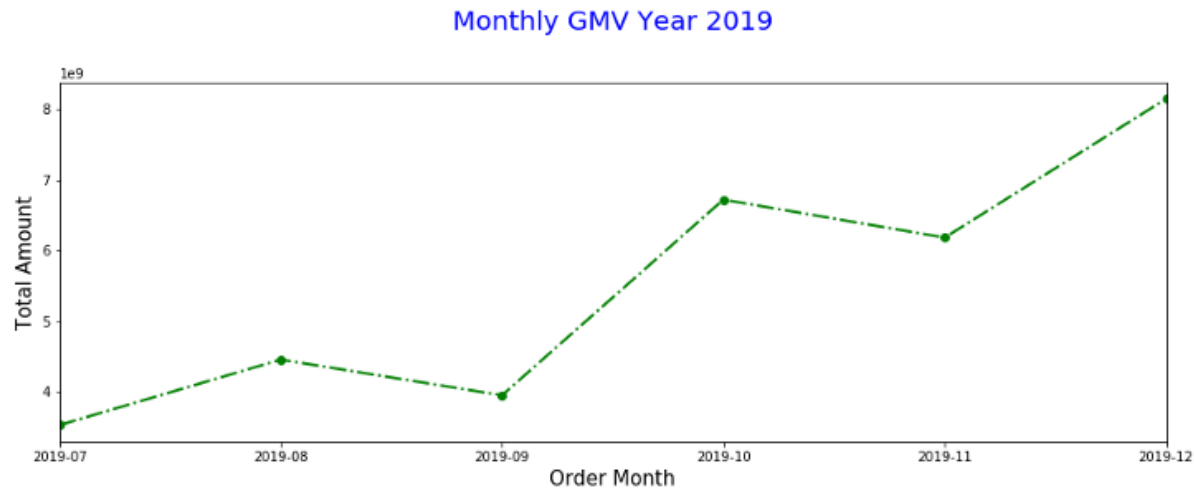
- Bisa dilihat bahwa judulnya sekarang menjadi lebih besar, agak berjarak ke atas, dan berwarna biru.
- Label di sumbu x maupun y juga menjadi lebih besar dari sebelumnya.

Kustomisasi Line dan Point

- Untuk mengubah style dari garis maupun titik di chart, lakukan dengan menambahkan parameter di function `.plot()`. Beberapa parameter yang bisa dikustomisasi:
 - **color**: mengubah warnanya (sama seperti di title)
 - **linewidth**: mengubah ketebalan line/garisnya (dalam satuan px)
 - **linestyle**: mengubah jenis dari garis. Misalnya '-' atau 'solid' untuk garis tak terputus (seperti pada default), '—' atau 'dashed' untuk garis putus-putus, ':' atau 'dotted' untuk garis berupa titik-titik, bisa juga '-.' atau 'dashdot' untuk garis dan titik bergantian.
 - **marker**: mengubah tipe points/titik data di chart. Ada banyak sekali kemungkinan nilai untuk marker ini, yang biasanya digunakan yaitu '.' untuk bulatan kecil/titik, 'o' untuk bulatan agak besar, 's' untuk persegi, 'D' untuk diamond/wajik, dan bentuk-bentuk lain seperti '+', 'x', '|', '*'.

Modifikasi line dan point

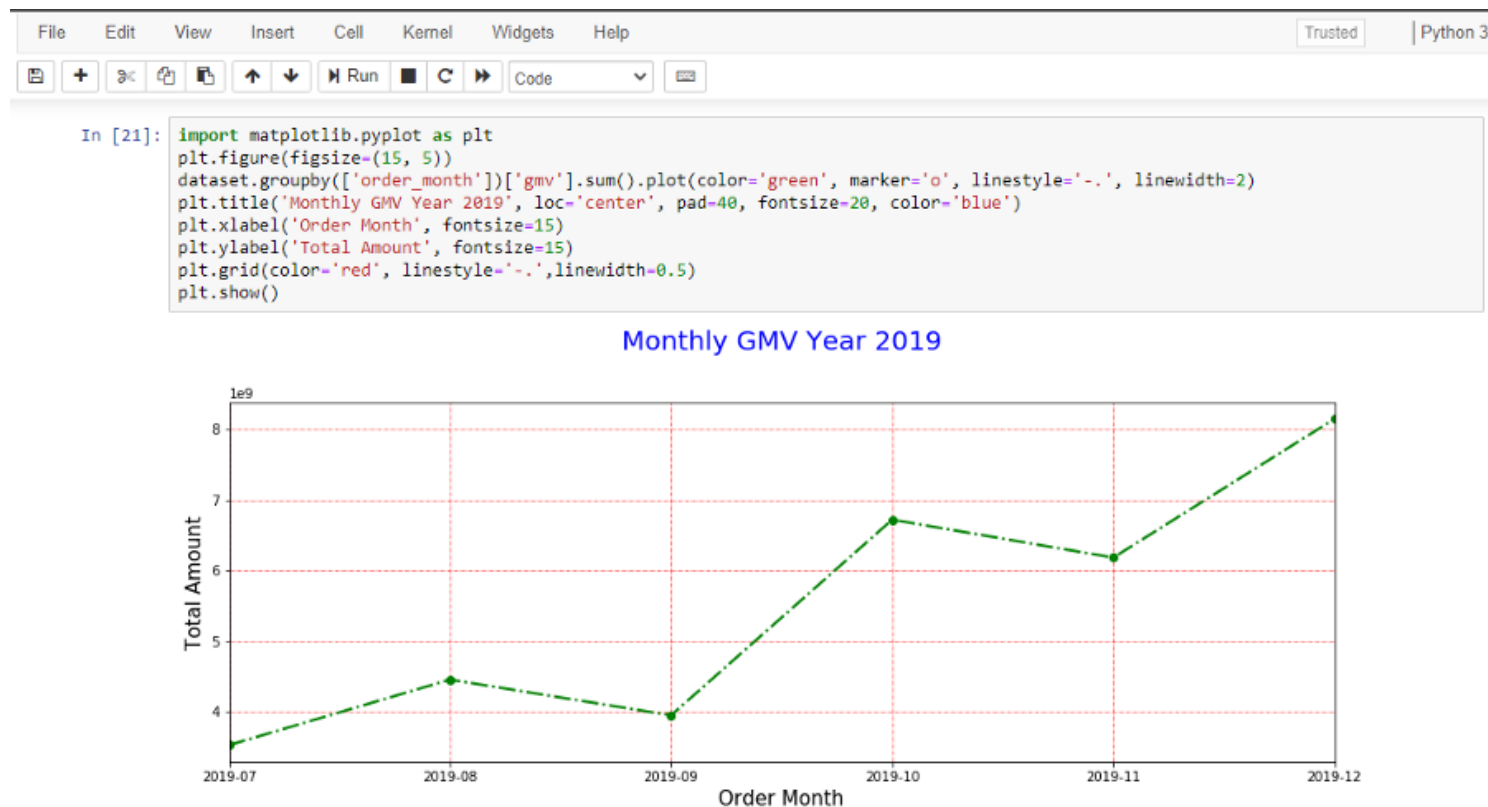
```
In [14]: import matplotlib.pyplot as plt
plt.figure(figsize=(15, 5))
dataset.groupby(['order_month'])['gmv'].sum().plot(color='green', marker='o', linestyle='--', linewidth=2)
plt.title('Monthly GMV Year 2019', loc='center', pad=40, fontsize=20, color='blue')
plt.xlabel('Order Month', fontsize=15)
plt.ylabel('Total Amount', fontsize=15)
plt.show()
```



Kita lihat line dari grafik diatas lebih tebal dan berwarna hijau, serta setiap titik data ditandai dengan tanda bulat 'o'.

Kustomisasi Grid

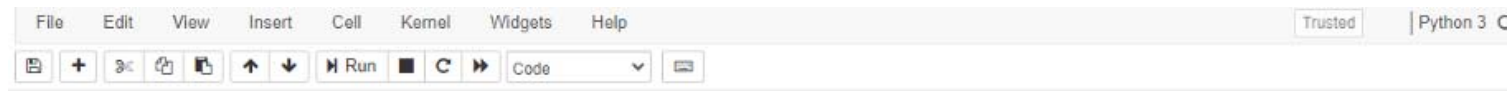
- Memodifikasi grid agar lebih mudah memetakan titik ke sumbu-y.
- Caranya dengan menambahkan plt.grid() dengan parameter yang mirip dengan line.



Terdapat grid dengan warna merah untuk memudahkan melihat kordinat dari setiap titik.

Kustomisasi Axis Ticks

- Nilai-nilai di sumbu x dan y bisa diakses melalui function `plt.xticks()` dan `plt.yticks()`.



```
In [23]: import matplotlib.pyplot as plt
plt.figure(figsize=(15, 5))
dataset.groupby(['order_month'])['gmv'].sum().plot(color='green', marker='o', linestyle='--', linewidth=2)
plt.title('Monthly GMV Year 2019', loc='center', pad=40, fontsize=20, color='blue')
plt.xlabel('Order Month', fontsize=15)
plt.ylabel('Total Amount (in Billions)', fontsize=15)
plt.grid(color='red', linestyle='--', linewidth=0.5)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000000000).astype(int))
plt.show()
```

Monthly GMV Year 2019

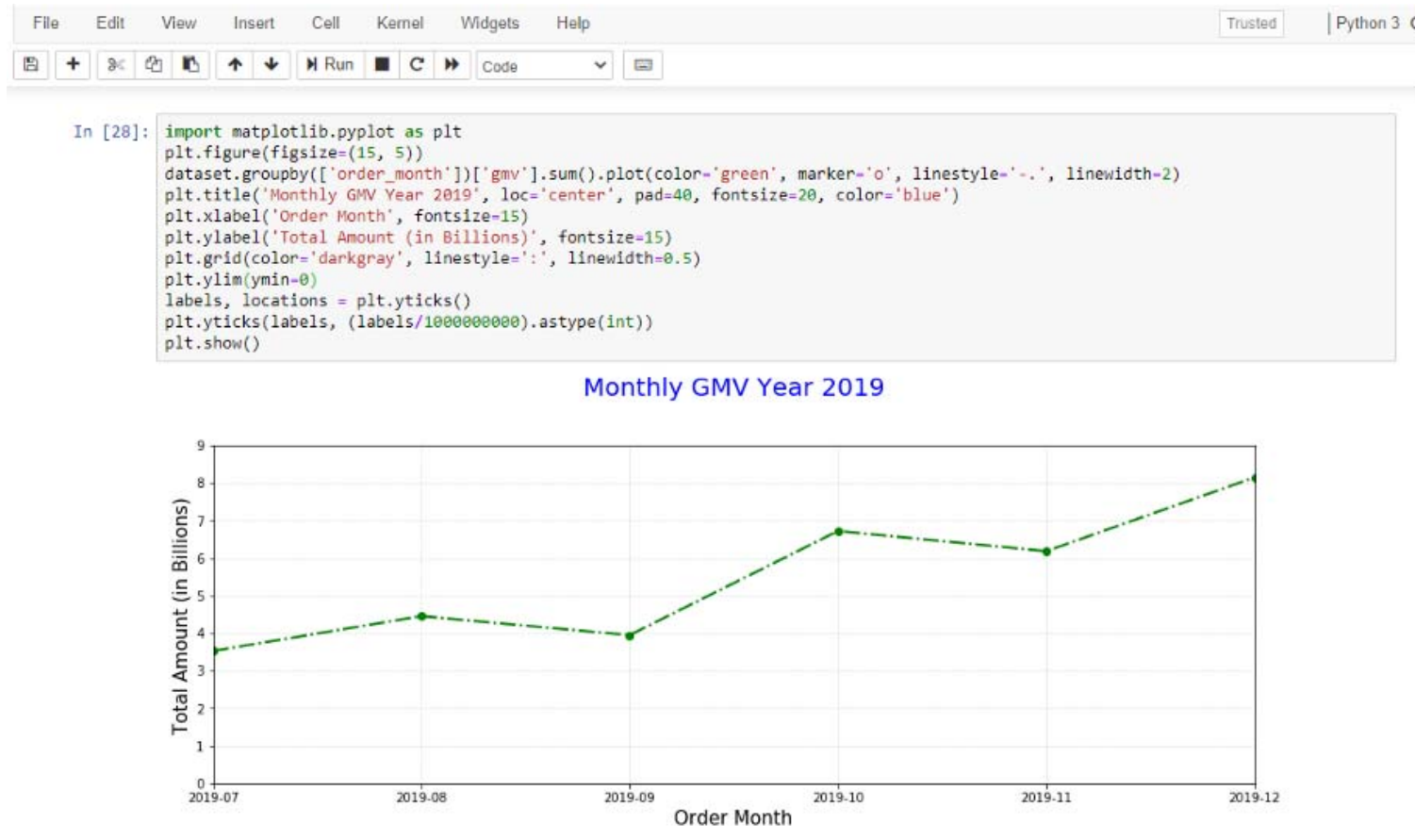


Dapat dilihat bahwa `plt.ticks()` yang sebelumnya, digantikan dengan nilai baru yaitu nilai awal dibagi dengan 1 milyar (1000000000).

Menentukan Batas Minimum dan Maksimum Axis Ticks

- Di sini dapat dilihat bahwa nilai terkecil di sumbu-y adalah 150.
- Hal ini kadang bisa membuat orang tertipu, seakan-akan GMV di bulan Oktober naik menjadi hampir 3 kali lipat dari September.
- Untuk itu sebaiknya diset agar sumbu-y nya dimulai dari 0, cukup tambahkan `plt.ylim(ymin=0)`.

Menentukan Batas Minimum dan Maksimum Axis Ticks

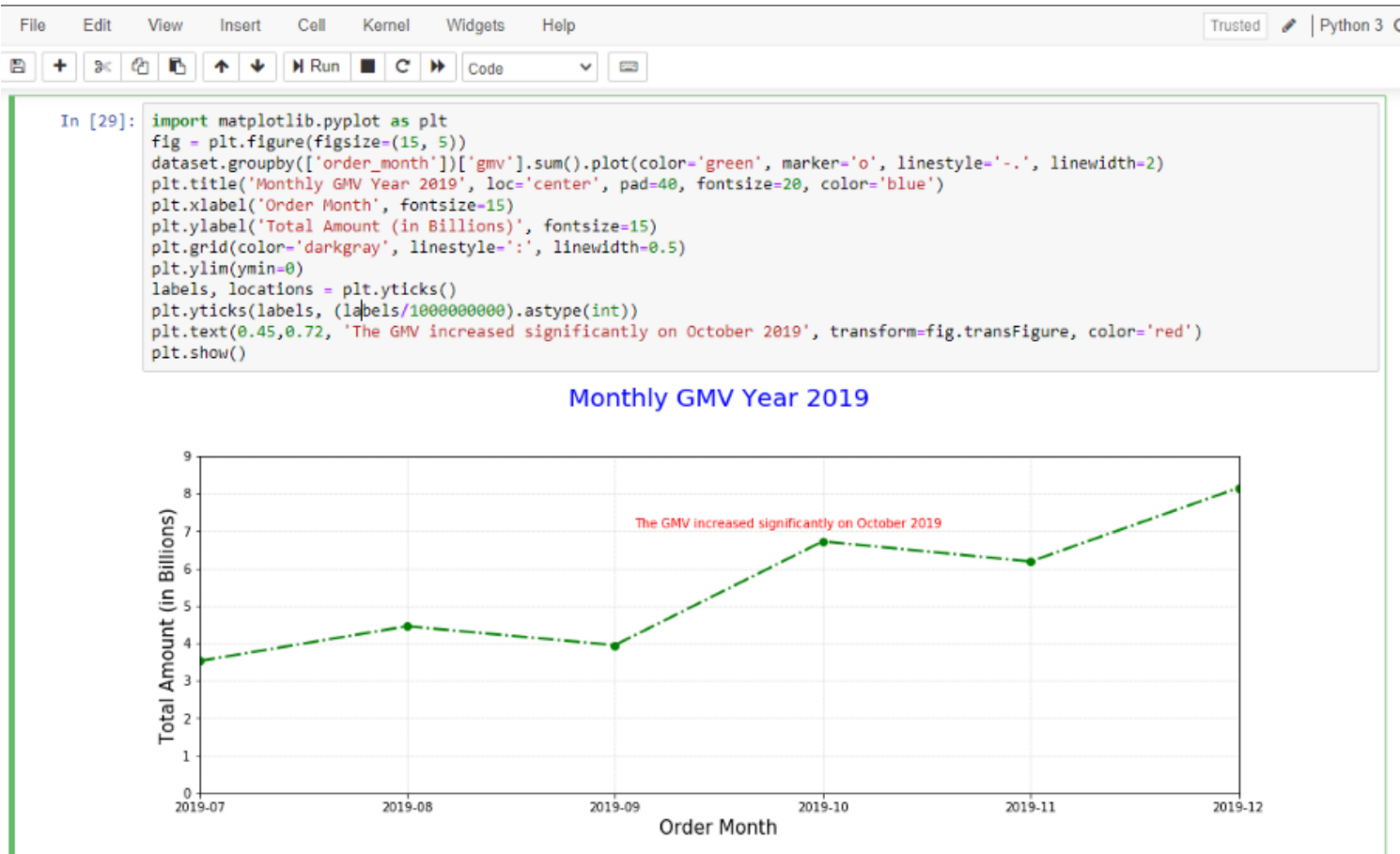


- Untuk mengatur batas maksimum, dapat ditambahkan parameter ymax.
- Mengatur batas minimum dan maksimum sumbu-x dapat dilakukan dengan function plt.xlim.

Menambahkan Informasi Pada Plot

- Dilihat dari trend-nya, mulai bulan Oktober 2019, GMV naik drastis setelah sebelumnya stagnan di angka 200–300 milyar per bulan.
- Menambahkan info ini di grafik dengan menambahkan `plt.text`.

Menambahkan Informasi Pada Plot



- Perhatikan bahwa ada beberapa parameter yang diset saat menggunakan `plt.text`. Dua angka pertama itu adalah koordinat, x dan y. Saat set `transform=fig.transFigure`, maka koordinatnya berkisar 0 sampai 1 (untuk x dari kanan ke kiri, dan untuk y, dari bawah ke atas).
- Jika parameter `transform` tidak diisi, maka koordinatnya dalam satuan inch (Dalam contoh ini, dari 0–15 dari kiri ke kanan, dan 0–5 dari bawah ke atas). Seperti halnya title atau label, dimungkinkan juga untuk set warna dan ukuran hurufnya.

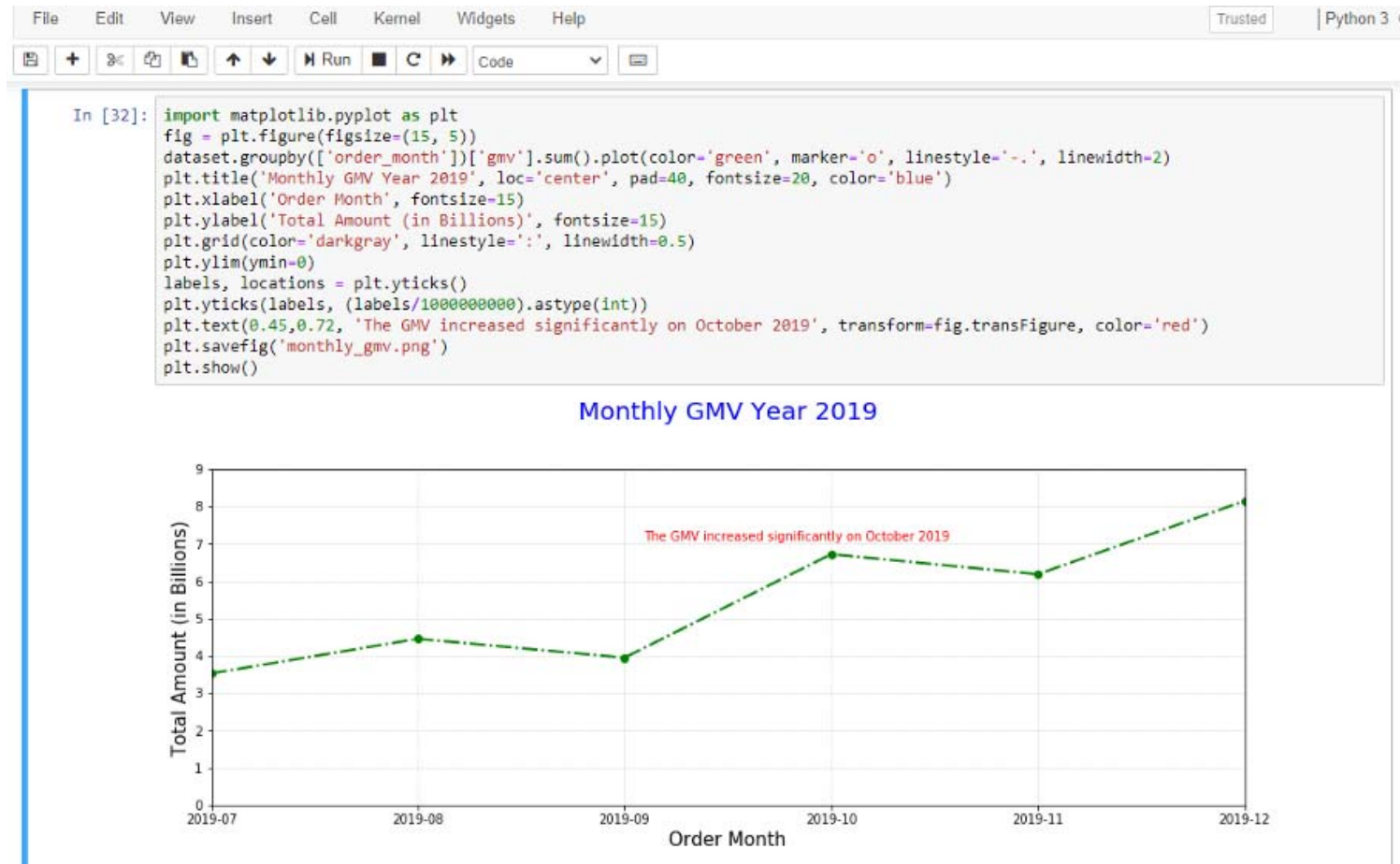
Menyimpan Hasil Plot Menjadi File Image

- Menyimpan sebagai file gambar dengan *function* **savefig** sebelum `plt.show()`.
- Menyimpannya ke berbagai tipe file: jpg, png, atau pdf.
- Untuk mengetahui format lengkapnya, kita bisa menggunakan code berikut:

`plt.gcf().canvas.get_supported_filetypes()`

```
{'ps': 'Postscript',  
 'eps': 'Encapsulated Postscript',  
 'pdf': 'Portable Document Format',  
 'pgf': 'PGF code for LaTeX',  
 'png': 'Portable Network Graphics',  
 'raw': 'Raw RGBA bitmap',  
 'rgba': 'Raw RGBA bitmap',  
 'svg': 'Scalable Vector Graphics',  
 'svgz': 'Scalable Vector Graphics',  
 'jpg': 'Joint Photographic Experts Group',  
 'jpeg': 'Joint Photographic Experts Group',  
 'tif': 'Tagged Image File Format',  
 'tiff': 'Tagged Image File Format'}
```

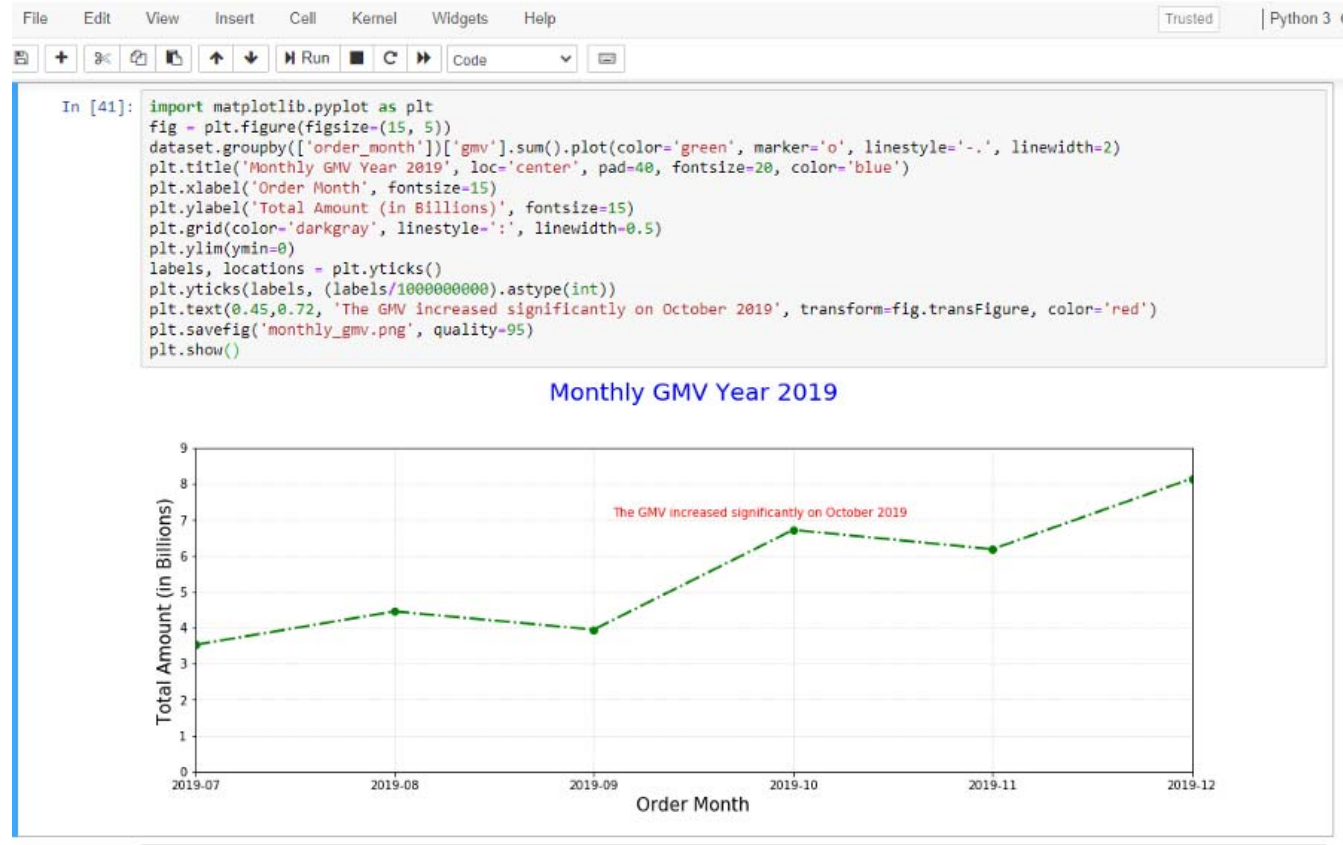
Menyimpan figure sebagai tipe png



Pengaturan Parameter untuk Menyimpan Gambar

- Hasil plot dapat diatur lagi bagaimana hasil dari file tersebut setelah di save. Misal dari kualitas gambar dan lainnya. Berbagai parameter yang bisa diatur saat menyimpan gambar, antara lain:
 - **dpi:** Resolusi gambar (dots per inch).
 - **quality:** Kualitas gambar (hanya berlaku jika formatnya jpg atau jpeg), bisa diisi nilai 1 (paling buruk) hingga 95 (paling bagus).
 - **facecolor:** Memberikan warna bagian depan figure, di luar area plot
 - **edgecolor:** Memberikan warna pinggiran gambar
 - **transparent:** Jika nilainya True, maka gambarnya jadi transparan (jika filenya png)

Pengaturan Parameter untuk Menyimpan Gambar



File akan menghasilkan kualitas gambar yang tersimpan sebesar 95% dari awal.

Tugas

- Dengan menggunakan dataset yang sama buatlah sebuah line chart dengan Matplotlib, yang menunjukkan **jumlah pembelian harian selama bulan Desember**.
- Beberapa spesifikasi yang harus diperhatikan:
 - Ukuran figure adalah 10x5
 - Sumbu-x adalah tanggal pembelian, dari tanggal 1–31 Desember 2019
 - Sumbu-y adalah jumlah unique customers di tiap tanggal
 - Title dan axis label harus ada, tulisan dan style-nya silakan disesuaikan sendiri

Petunjuk

- Import dulu library yang dibutuhkan untuk membuat grafik, seperti library datetime, pandas dan tentunya matplotlib.
- Baca dataset nya.
- Buat kolom order_month. Kolom order_month berisi data pemesanan dalam bulanan.
- Buat kolom gmv. Kolom gmv didapat dari perkalian kolom item price dan kolom quantity.
- Buat grafiknya dengan tahapan:
 - Atur ukuran figure, disini kita buat 10x5.
 - buat grafiknya, caranya dengan groupby kolom order month dan order date di sumbu x dan kolom customer id di sumbu y.
 - Beri title grafik sesuai kode diatas lengkap beserta parameteranya.
 - Beri label pada sumbu x dan y, lengkap beserta parameteranya.
 - Atur grid nya lengkap beserta parameteranya.
 - Atur batas minimum sumbu-y (mulai dari 0)
 - Tampilkan grafik