

Modul 2 STK - Week #3 - Pengantar Bahasa Python (Library)

September 2, 2024

1 Modul 2 - Pengantar Bahasa Python

Modul ini bertujuan untuk memperkenalkan mahasiswa pada penggunaan library Python yang penting untuk analisis data dan visualisasi. Python, dengan ekosistem library-nya yang luas seperti NumPy, Pandas, dan Matplotlib, menjadi alat yang sangat berguna dalam berbagai bidang, termasuk analisis data dan ilmu data.

Dalam konteks Sistem Temu Kembali (Information Retrieval), kemampuan untuk mengolah dan menganalisis data dalam jumlah besar dengan cepat dan efisien sangatlah penting. NumPy memungkinkan operasi matematis dan komputasi yang cepat pada array multidimensi, Pandas menyediakan struktur data yang kuat dan fleksibel untuk memanipulasi data dalam format tabel, dan Matplotlib memungkinkan visualisasi data yang informatif untuk interpretasi dan komunikasi hasil analisis.

1.1 Tujuan Praktikum

- 1) Mahasiswa memahami konsep dasar library pada Python
- 2) Mahasiswa mampu membedakan beberapa library penting dalam Pengolahan Data seperti Numpy, Pandas, dan Matplotlib
- 3) Mahasiswa mampu mengimplementasikan penggunaan library pada studi kasus yang diberikan
- 4) Mahasiswa mampu memvisualisasikan suatu studi kasus menggunakan library Python

1.2 Alat dan Bahan:

- 1) Laptop atau Komputer
- 2) Akses ke Jupyter Notebook/Google Colab/VSC
- 3) Instalasi Python 3.x
- 4) Berkas teks
- 5) Akses Internet

1.3 Materi

Dalam dunia pemrograman, library dan module merupakan komponen penting yang memberikan kekuatan tambahan bagi bahasa pemrograman, termasuk Python. Library adalah kumpulan fungsi, metode, dan class yang dapat digunakan untuk melakukan tugas-tugas tertentu, sementara module adalah file yang berisi kode Python yang dapat diimpor dan digunakan dalam skrip lain. Kedua komponen ini memungkinkan programmer untuk memanfaatkan kode yang sudah ada dan teruji, sehingga menghemat waktu dan usaha dalam pengembangan aplikasi.

Python dikenal karena ekosistem library-nya yang luas dan kaya, yang memungkinkan pengguna untuk melakukan berbagai macam tugas mulai dari analisis data hingga pengembangan web dan pembelajaran mesin. Dalam konteks analisis data dan visualisasi, Python memiliki beberapa library utama yang sangat kuat seperti NumPy, Pandas, dan Matplotlib.

1.3.1 NumPy

Pengenalan Numpy NumPy (Numerical Python) adalah library Python yang digunakan untuk komputasi ilmiah. NumPy menyediakan objek array multidimensi, berbagai fungsi derivatif (matematika, aljabar linier, dan transformasi Fourier), dan alat untuk bekerja dengan array secara efisien.

Instalasi NumPy Untuk menggunakan NumPy, kita harus menginstalnya terlebih dahulu (jika belum terinstal) dan mengimpornya ke dalam proyek Python.

```
[1]: pip install numpy
```

```
Requirement already satisfied: numpy in  
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages  
(1.25.2)  
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.1 -> 24.2
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Mengimpor Library NumPy

```
[2]: # Import NumPy  
import numpy as np
```

Membuat dan Memanipulasi Array

```
[3]: # Membuat array 1 dimensi  
array_1d = np.array([1, 2, 3, 4, 5])  
print("Array 1D:", array_1d)  
  
# Membuat array 2 dimensi  
array_2d = np.array([[1, 2, 3], [4, 5, 6]])  
print("Array 2D:\n", array_2d)  
  
# Mengakses elemen array  
print("Elemen pertama array_1d:", array_1d[0])  
print("Elemen di baris ke-2, kolom ke-3 dari array_2d:", array_2d[1, 2])  
  
# Operasi dasar pada array  
print("Penjumlahan array:", array_1d + 5)  
print("Perkalian array:", array_1d * 2)  
  
# Fungsi matematika
```

```
print("Akar kuadrat dari array_1d:", np.sqrt(array_1d))
print("Sine dari array_1d:", np.sin(array_1d))

# Reshape array
array_reshaped = array_1d.reshape((1, 5))
print("Array yang telah di reshape:\n", array_reshaped)
```

```
Array 1D: [1 2 3 4 5]
Array 2D:
[[1 2 3]
 [4 5 6]]
Elemen pertama array_1d: 1
Elemen di baris ke-2, kolom ke-3 dari array_2d: 6
Penjumlahan array: [ 6  7  8  9 10]
Perkalian array: [ 2  4  6  8 10]
Akar kuadrat dari array_1d: [1.          1.41421356  1.73205081  2.
 2.23606798]
Sine dari array_1d: [ 0.84147098  0.90929743  0.14112001 -0.7568025
-0.95892427]
Array yang telah di reshape:
[[1 2 3 4 5]]
```

1.3.2 Pandas

Pengenalan Pandas Pandas adalah library Python yang digunakan untuk manipulasi data dan analisis data. Pandas menyediakan struktur data yang disebut Series dan DataFrame, yang sangat berguna untuk menangani data tabular (seperti tabel dari database atau file Excel).

Instalasi Pandas

[4]: `pip install pandas`

```
Requirement already satisfied: pandas in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages
(2.1.0)
Requirement already satisfied: numpy>=1.22.4 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
pandas) (1.25.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
pandas) (2023.3)
Requirement already satisfied: six>=1.5 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
```

```
python-dateutil>=2.8.2->pandas) (1.16.0)
```

Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 24.1 -> 24.2

[notice] To update, run: python.exe -m pip install --upgrade pip

Mengimpor Library Pandas

```
[5]: # Import Pandas
import pandas as pd
```

Membuat Series dan DataFrame

```
[6]: # Membuat Series
series = pd.Series([1, 2, 3, 4, 5])
print(series)

# Membuat DataFrame
data = {
    'Nama': ['Ayu', 'Budi', 'Cici'],
    'Umur': [23, 25, 22]
}
df = pd.DataFrame(data)
print(df)
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
   Nama  Umur
0  Ayu   23
1  Budi   25
2  Cici   22
```

Membuat dan Manipulasi DataFrame

```
[7]: # Membuat DataFrame dari dictionary
data = {
    'Nama': ['Andi', 'Budi', 'Cici', 'Dodi'],
    'Umur': [21, 23, 19, 22],
    'Kota': ['Jakarta', 'Bandung', 'Surabaya', 'Medan']
}
df = pd.DataFrame(data)
print("DataFrame:\n", df)

# Menampilkan informasi DataFrame
print("\nInfo DataFrame:")
print(df.info())
```

DataFrame:

	Nama	Umur	Kota
0	Andi	21	Jakarta
1	Budi	23	Bandung
2	Cici	19	Surabaya
3	Dodi	22	Medan

Info DataFrame:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 4 entries, 0 to 3

Data columns (total 3 columns):

#	Column	Non-Null Count	Dtype
0	Nama	4 non-null	object
1	Umur	4 non-null	int64
2	Kota	4 non-null	object

dtypes: int64(1), object(2)

memory usage: 224.0+ bytes

None

```
[8]: # Mengakses kolom DataFrame
print("\nKolom 'Nama':\n", df['Nama'])

# Mengakses baris DataFrame
print("\nBaris ke-2:\n", df.iloc[1])

# Menambahkan kolom baru
df['Skor'] = [88, 92, 95, 89]
print("\nDataFrame dengan kolom baru 'Skor':\n", df)

# Menghapus kolom
df = df.drop('Kota', axis=1)
print("\nDataFrame setelah kolom 'Kota' dihapus:\n", df)

# Operasi agregasi
print("\nRata-rata umur:", df['Umur'].mean())
print("Umur maksimal:", df['Umur'].max())
```

Kolom 'Nama':

0	Andi
1	Budi
2	Cici
3	Dodi

Name: Nama, dtype: object

Baris ke-2:

Nama	Budi
------	------

```
Umur      23
Kota      Bandung
Name: 1, dtype: object
```

DataFrame dengan kolom baru 'Skor':

	Nama	Umur	Kota	Skor
0	Andi	21	Jakarta	88
1	Budi	23	Bandung	92
2	Cici	19	Surabaya	95
3	Dodi	22	Medan	89

DataFrame setelah kolom 'Kota' dihapus:

	Nama	Umur	Skor
0	Andi	21	88
1	Budi	23	92
2	Cici	19	95
3	Dodi	22	89

Rata-rata umur: 21.25

Umur maksimal: 23

Jika ingin mengimplementasikan Pandas pada File CSV, maka terlebih dahulu lakukan lah pembacaan data CSV yang dimaksud lalu simpan ke dalam DataFrame seperti pada contoh berikut:

```
[9]: # Membaca data dari file CSV
      #df_csv = pd.read_csv('path_to_file.csv')
```

df_csv adalah dataframe CSV yang dibuat melalui penggunaan library Pandas setelah membaca file CSV 'path_to_file.csv'. Pastikan untuk menyimpan file CSV ke dalam folder yang sama

Mengganti Nilai NaN

```
[10]: import pandas as pd
      import numpy as np

      data = {
          'Nama': ['Ayu', 'Budi', np.nan],
          'Umur': [23, np.nan, 22]
      }
      df = pd.DataFrame(data)

      # Mengganti nilai NaN dengan nilai tertentu
      df_filled = df.fillna('Tidak diketahui')
      print(df_filled)
```

	Nama	Umur
0	Ayu	23.0
1	Budi	Tidak diketahui
2	Tidak diketahui	22.0

Penjelasan: fillna() digunakan untuk mengganti nilai NaN dengan nilai yang diinginkan.

1.3.3 Matplotlib

Pengenalan Matplotlib Matplotlib adalah library plotting 2D yang digunakan untuk membuat visualisasi data dalam bentuk grafik dan plot. Dengan Matplotlib, kita bisa membuat berbagai jenis grafik seperti line plot, scatter plot, bar plot, dll.

Instalasi Matplotlib

```
[11]: pip install matplotlib
```

```
Requirement already satisfied: matplotlib in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages
(3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (1.1.0)
Requirement already satisfied: cyclor>=0.10 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (4.42.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (1.4.5)
Requirement already satisfied: numpy<2,>=1.21 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (1.25.2)
Requirement already satisfied: packaging>=20.0 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (10.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
c:\users\lenovo\appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.1 -> 24.2
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Mengimpor Library Matplotlib

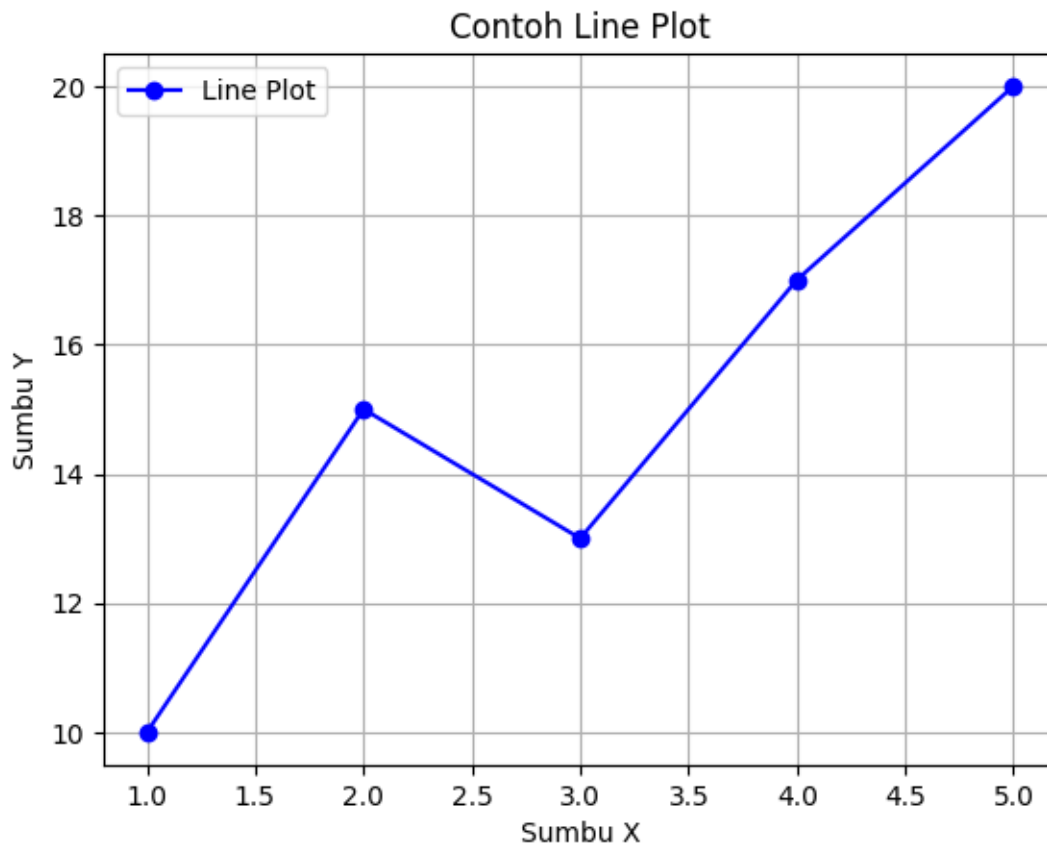
```
[12]: # Import Matplotlib
import matplotlib.pyplot as plt
```

Membuat Plot Dasar dengan Matplotlib

```
[13]: # Membuat data untuk plot
x = [1, 2, 3, 4, 5]
y = [10, 15, 13, 17, 20]
```

a. Membuat Grafik Garis

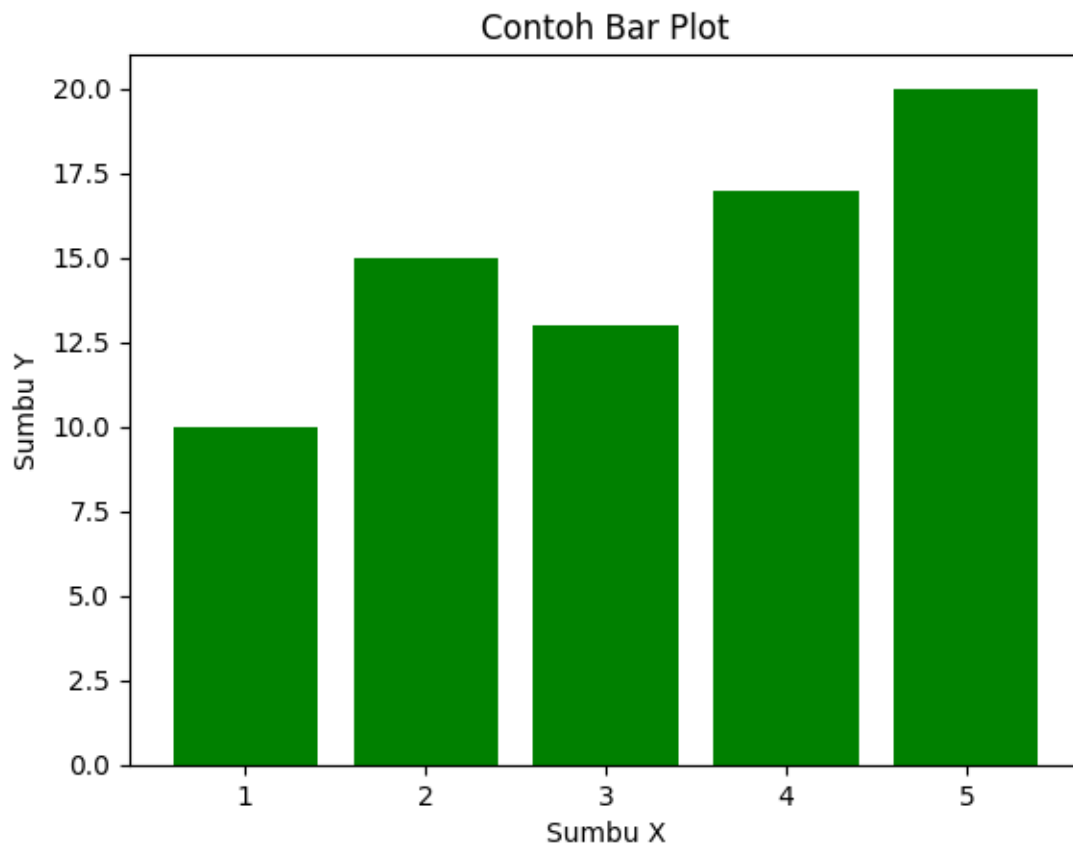
```
[14]: # Membuat line plot
plt.plot(x, y, label='Line Plot', color='blue', marker='o')
plt.title('Contoh Line Plot')
plt.xlabel('Sumbu X')
plt.ylabel('Sumbu Y')
plt.legend()
plt.grid(True)
plt.show()
```



Penjelasan: `plt.plot()` digunakan untuk membuat grafik garis. `plt.title()`, `plt.xlabel()`, dan `plt.ylabel()` digunakan untuk menambahkan judul dan label sumbu.

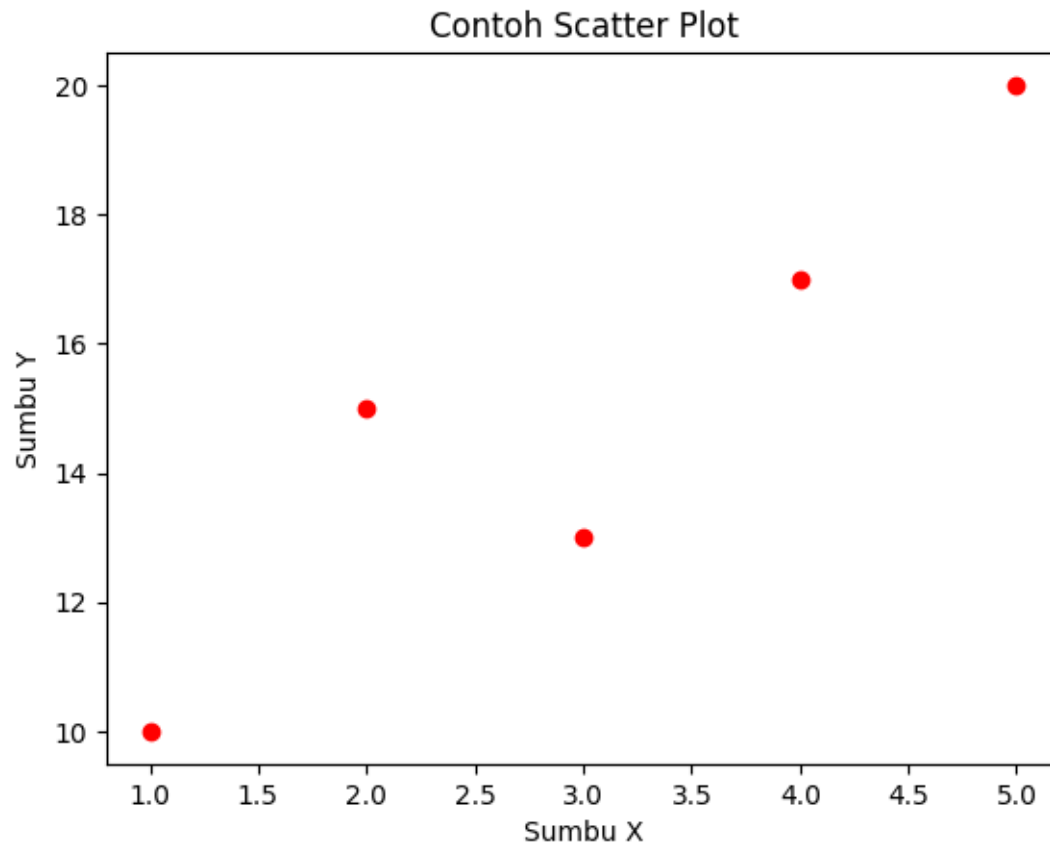
b. Membuat grafik batang

```
[15]: # Membuat bar plot
plt.bar(x, y, color='green')
plt.title('Contoh Bar Plot')
plt.xlabel('Sumbu X')
plt.ylabel('Sumbu Y')
plt.show()
```



c. Membuat Scatter Plot

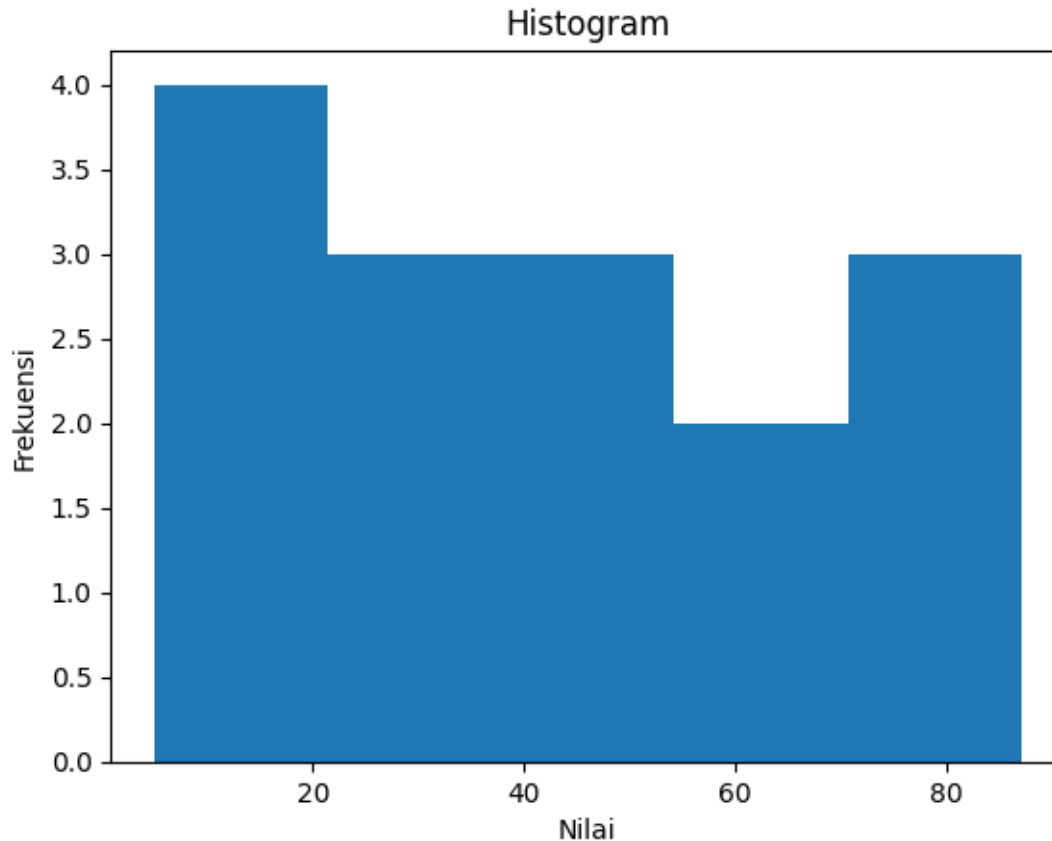
```
[16]: # Membuat scatter plot
plt.scatter(x, y, color='red')
plt.title('Contoh Scatter Plot')
plt.xlabel('Sumbu X')
plt.ylabel('Sumbu Y')
plt.show()
```



d. Membuat Histogram

```
[17]: data = [22, 87, 5, 43, 56, 73, 55, 54, 11, 20, 51, 5, 79, 31, 27]

plt.hist(data, bins=5)
plt.title('Histogram')
plt.xlabel('Nilai')
plt.ylabel('Frekuensi')
plt.show()
```



Penjelasan: `plt.hist()` digunakan untuk membuat histogram. Parameter `bins` menentukan jumlah interval.

1.3.4 Latihan: Studi Kasus 1

Studi Kasus ini bertujuan untuk mengimplementasikan `numpy`, `pandas`, dan `matplotlib`. Pada studi kasus ini, kita menggunakan file CSV `BigmacPrice.csv` sebagai dataset. Catatan: Dataset akan dikirim terpisah

Import Modules Langkah pertama adalah melakukan import Modules atau library yang dibutuhkan, yakni `Numpy`, `Pandas`, dan `Matplotlib`

```
[18]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Menggunakan Sample Dataset Selanjutnya, gunakan sample dataset `BigmacPrice.csv` yang disimpan pada folder yang sama letak file Python berada.

```
[19]: df = pd.read_csv('BigmacPrice.csv')
```

```
[20]: df
```

```
[20]:
```

	date	currency_code	name	local_price	dollar_ex \
0	2000-04-01	ARS	Argentina	2.50	1
1	2000-04-01	AUD	Australia	2.59	1
2	2000-04-01	BRL	Brazil	2.95	1
3	2000-04-01	GBP	Britain	1.90	1
4	2000-04-01	CAD	Canada	2.85	1
...
1941	2022-07-01	AED	United Arab Emirates	18.00	3
1942	2022-07-01	USD	United States	5.15	1
1943	2022-07-01	UYU	Uruguay	255.00	41
1944	2022-07-01	VES	Venezuela	10.00	5
1945	2022-07-01	VND	Vietnam	69000.00	23417

	dollar_price
0	2.50
1	2.59
2	2.95
3	1.90
4	2.85
...	...
1941	6.00
1942	5.15
1943	6.22
1944	2.00
1945	2.95


```
[1946 rows x 6 columns]
```

```
[21]: #Melihat ukuran baris dan kolom
df.shape
```

```
[21]: (1946, 6)
```

```
[22]: #Melihat daftar kolom
df.columns
```

```
[22]: Index(['date', 'currency_code', 'name', 'local_price', 'dollar_ex',
        'dollar_price'],
        dtype='object')
```

```
[23]: #Memfilter salah satu data dengan karakter tertentu
data_indonesia = df[df['name'] == 'Indonesia']
data_indonesia

#Kode di atas berarti hanya menampilkan data yang atribut namanya adalah
↳Indonesia
```

```

[23]:      date currency_code      name local_price dollar_ex \
12    2000-04-01      IDR Indonesia    14500.0      7945
40    2001-04-01      IDR Indonesia    14700.0     10855
68    2002-04-01      IDR Indonesia    16000.0      9430
102   2003-04-01      IDR Indonesia    16100.0      8740
136   2004-05-01      IDR Indonesia    16100.0      9120
176   2005-06-01      IDR Indonesia    14600.0      9545
215   2006-01-01      IDR Indonesia    14600.0      9460
249   2006-05-01      IDR Indonesia    14600.0      9325
290   2007-01-01      IDR Indonesia    15900.0      9090
331   2007-06-01      IDR Indonesia    15900.0      9015
372   2008-06-01      IDR Indonesia    18700.0      9152
412   2009-07-01      IDR Indonesia    20900.0     10200
453   2010-01-01      IDR Indonesia    20900.0      9320
494   2010-07-01      IDR Indonesia    22780.0      9063
543   2011-07-01      IDR Indonesia    22534.0      8523
598   2012-01-01      IDR Indonesia    22534.0      9160
652   2012-07-01      IDR Indonesia    24200.0      9482
706   2013-01-01      IDR Indonesia    27939.0      9767
761   2013-07-01      IDR Indonesia    27939.0      9965
816   2014-01-01      IDR Indonesia    27939.0     12140
872   2014-07-01      IDR Indonesia    27939.0     11505
929   2015-01-01      IDR Indonesia    27939.0     12480
985   2015-07-01      IDR Indonesia    30500.0     13344
1041  2016-01-01      IDR Indonesia    30500.0     13947
1097  2016-07-01      IDR Indonesia    31000.0     13112
1153  2017-01-01      IDR Indonesia    31000.0     13329
1209  2017-07-01      IDR Indonesia    32126.0     13369
1265  2018-01-01      IDR Indonesia    35750.0     13359
1326  2018-07-01      IDR Indonesia    31500.0     14360
1398  2019-01-01      IDR Indonesia    33000.0     14090
1470  2019-07-09      IDR Indonesia    32000.0     14130
1542  2020-01-14      IDR Indonesia    33000.0     13670
1614  2020-07-01      IDR Indonesia    34000.0     14435
1686  2021-01-01      IDR Indonesia    34000.0     14125
1758  2021-07-01      IDR Indonesia    34000.0     14517
1831  2022-01-01      IDR Indonesia    34000.0     14382
1903  2022-07-01      IDR Indonesia    35000.0     14977

```

```

      dollar_price
12          1.83
40          1.35
68          1.70
102         1.84
136         1.77
176         1.53
215         1.54

```

249	1.57
290	1.75
331	1.76
372	2.04
412	2.05
453	2.24
494	2.51
543	2.64
598	2.46
652	2.55
706	2.86
761	2.80
816	2.30
872	2.43
929	2.24
985	2.29
1041	2.19
1097	2.36
1153	2.33
1209	2.40
1265	2.68
1326	2.19
1398	2.34
1470	2.26
1542	2.41
1614	2.36
1686	2.41
1758	2.34
1831	2.36
1903	2.34

```
[24]: #Menampilkan detail statistik dari data yang telah difilter
data_indonesia.describe()
```

```
[24]:
```

	local_price	dollar_ex	dollar_price
count	37.000000	37.000000	37.000000
mean	25314.027027	11416.972973	2.189730
std	7488.014521	2308.769483	0.373054
min	14500.000000	7945.000000	1.350000
25%	16100.000000	9320.000000	1.840000
50%	27939.000000	10855.000000	2.300000
75%	32000.000000	13670.000000	2.410000
max	35750.000000	14977.000000	2.860000

```
[25]: #Menampilkan data sesuai kriteria
harga_murah = data_indonesia[data_indonesia['local_price'] < 20000]
harga_murah
```

```
[25]:
```

	date	currency_code	name	local_price	dollar_ex	dollar_price
12	2000-04-01	IDR	Indonesia	14500.0	7945	1.83
40	2001-04-01	IDR	Indonesia	14700.0	10855	1.35
68	2002-04-01	IDR	Indonesia	16000.0	9430	1.70
102	2003-04-01	IDR	Indonesia	16100.0	8740	1.84
136	2004-05-01	IDR	Indonesia	16100.0	9120	1.77
176	2005-06-01	IDR	Indonesia	14600.0	9545	1.53
215	2006-01-01	IDR	Indonesia	14600.0	9460	1.54
249	2006-05-01	IDR	Indonesia	14600.0	9325	1.57
290	2007-01-01	IDR	Indonesia	15900.0	9090	1.75
331	2007-06-01	IDR	Indonesia	15900.0	9015	1.76
372	2008-06-01	IDR	Indonesia	18700.0	9152	2.04

```
[26]: #Menampilkan data sesuai kriteria
harga_mahal = data_indonesia[data_indonesia['local_price'] >= 20000]
harga_mahal
```

```
[26]:
```

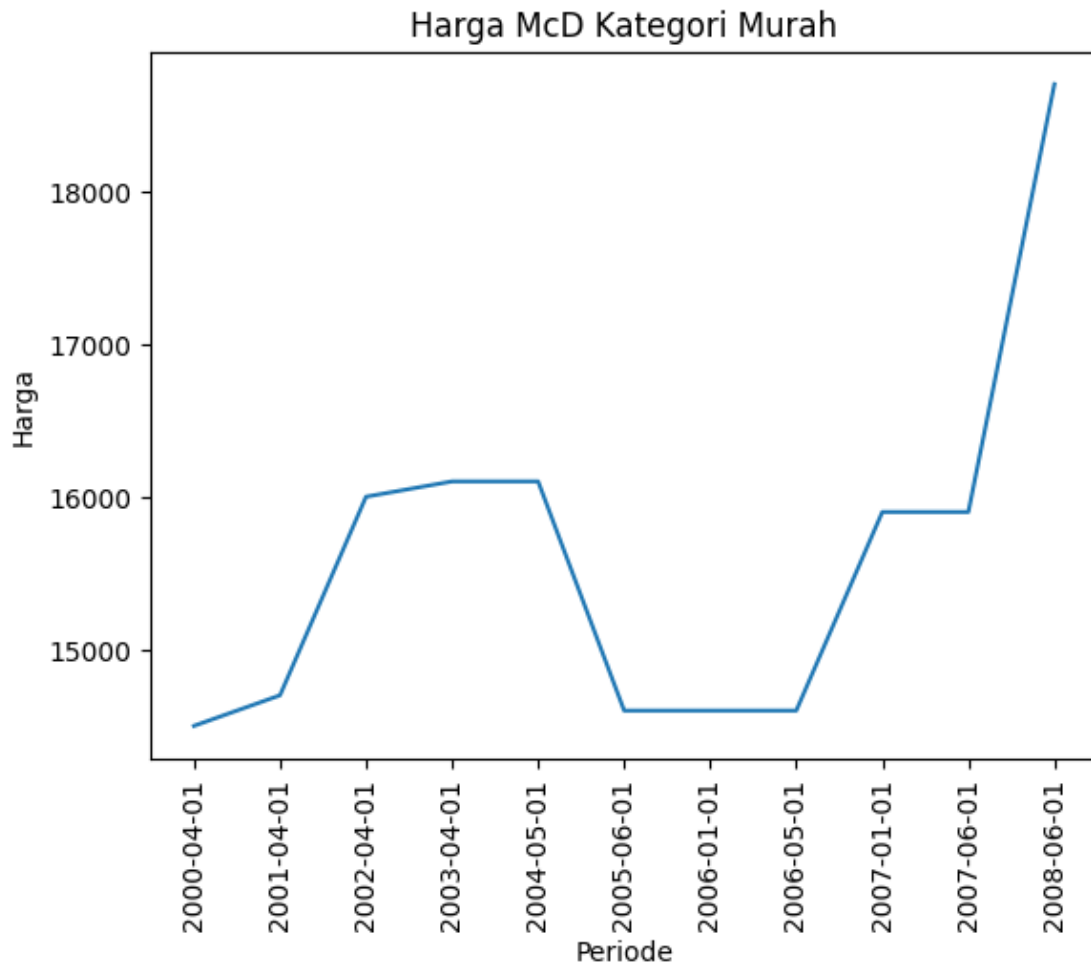
	date	currency_code	name	local_price	dollar_ex	\
412	2009-07-01	IDR	Indonesia	20900.0	10200	
453	2010-01-01	IDR	Indonesia	20900.0	9320	
494	2010-07-01	IDR	Indonesia	22780.0	9063	
543	2011-07-01	IDR	Indonesia	22534.0	8523	
598	2012-01-01	IDR	Indonesia	22534.0	9160	
652	2012-07-01	IDR	Indonesia	24200.0	9482	
706	2013-01-01	IDR	Indonesia	27939.0	9767	
761	2013-07-01	IDR	Indonesia	27939.0	9965	
816	2014-01-01	IDR	Indonesia	27939.0	12140	
872	2014-07-01	IDR	Indonesia	27939.0	11505	
929	2015-01-01	IDR	Indonesia	27939.0	12480	
985	2015-07-01	IDR	Indonesia	30500.0	13344	
1041	2016-01-01	IDR	Indonesia	30500.0	13947	
1097	2016-07-01	IDR	Indonesia	31000.0	13112	
1153	2017-01-01	IDR	Indonesia	31000.0	13329	
1209	2017-07-01	IDR	Indonesia	32126.0	13369	
1265	2018-01-01	IDR	Indonesia	35750.0	13359	
1326	2018-07-01	IDR	Indonesia	31500.0	14360	
1398	2019-01-01	IDR	Indonesia	33000.0	14090	
1470	2019-07-09	IDR	Indonesia	32000.0	14130	
1542	2020-01-14	IDR	Indonesia	33000.0	13670	
1614	2020-07-01	IDR	Indonesia	34000.0	14435	
1686	2021-01-01	IDR	Indonesia	34000.0	14125	
1758	2021-07-01	IDR	Indonesia	34000.0	14517	
1831	2022-01-01	IDR	Indonesia	34000.0	14382	
1903	2022-07-01	IDR	Indonesia	35000.0	14977	

	dollar_price
412	2.05

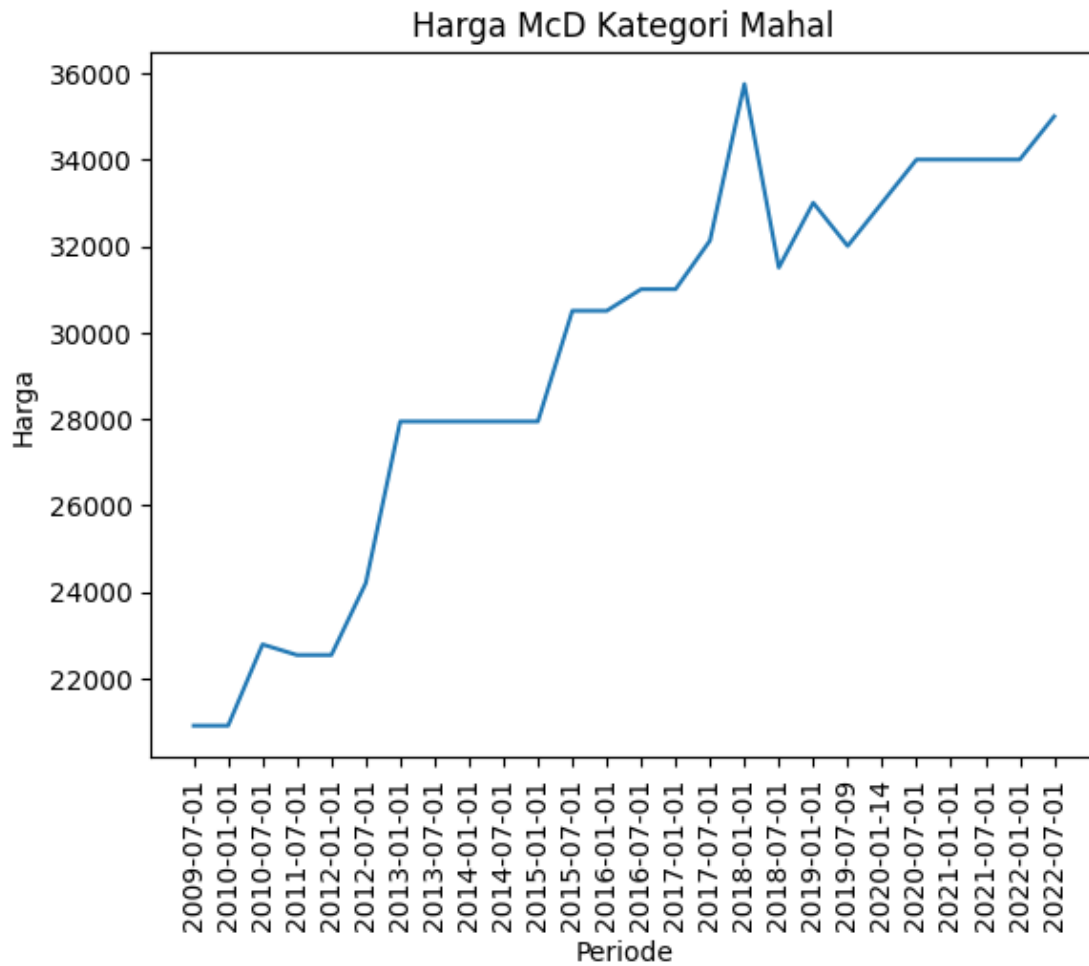
453	2.24
494	2.51
543	2.64
598	2.46
652	2.55
706	2.86
761	2.80
816	2.30
872	2.43
929	2.24
985	2.29
1041	2.19
1097	2.36
1153	2.33
1209	2.40
1265	2.68
1326	2.19
1398	2.34
1470	2.26
1542	2.41
1614	2.36
1686	2.41
1758	2.34
1831	2.36
1903	2.34

Plotting Menggunakan Matplotlib Selanjutnya, lakukan plotting untuk visualisasi data menggunakan matplotlib. Untuk contoh ini, kita menggunakan variabel atau atribut `local_price` dan `date` untuk melihat laju perubahan harga per waktu.

```
[27]: #Melakukan plotting terhadap dua variabel, misalnya kategori murah berdasarkan
      ↪waktu
plt.plot(harga_murah['date'], harga_murah['local_price'])
plt.xlabel('Periode')
plt.ylabel('Harga')
plt.xticks(rotation='vertical')
plt.title('Harga McD Kategori Murah')
plt.show()
```

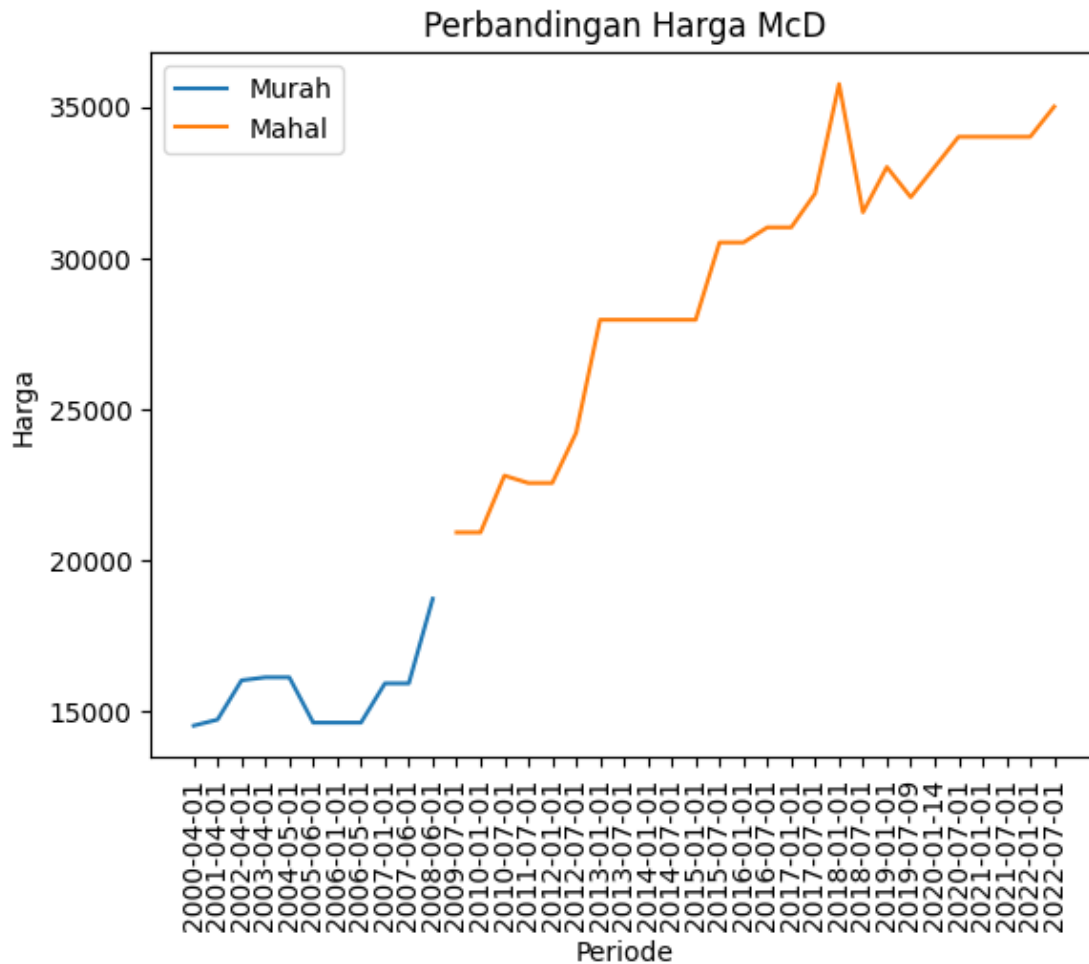



```
[28]: #Melakukan plotting terhadap dua variabel, misalnya kategori mahal berdasarkan waktu
plt.plot(harga_mahal['date'], harga_mahal['local_price'])
plt.xlabel('Periode')
plt.ylabel('Harga')
plt.xticks(rotation='vertical')
plt.title('Harga McD Kategori Mahal')
plt.show()
```



Plotting dengan menggabungkan dua grafik Visualisasi juga dapat dilakukan dengan menggunakan dua grafik yang tersedia dalam matplotlib, seperti pada contoh berikut:

```
[29]: plt.plot(harga_murah['date'], harga_murah['local_price'])
plt.plot(harga_mahal['date'], harga_mahal['local_price'])
plt.xticks(rotation='vertical')
plt.xlabel('Periode')
plt.ylabel('Harga')
plt.title('Perbandingan Harga McD')
plt.legend(['Murah', 'Mahal'])
plt.show()
```



1.3.5 Latihan 2: Studi Kasus 2

Pada Studi Kasus 2 ini, kita mencoba menggunakan dataset yang lain yakni `sample_data_plot.csv`. Catatan: Dataset akan dikirim terpisah

Import Sample Dataset Langkah pertama adalah lakukan pembacaan dataset terlebih dahulu. Sebenarnya, sebelum melakukan pembacaan dataset, maka kita perlu melakukan import library, namun karena pada contoh Studi Kasus 1 kita sudah melakukan import library, selama codenya dilakukan pada file yang sama, maka tidak perlu lagi melakukan import library.

```
[30]: df = pd.read_csv('sample_data_plot.csv')
      df.head()
```

```
[30]: Country name Regional indicator Ladder score \
0      Finland      Western Europe      7.8087
1      Denmark      Western Europe      7.6456
2  Switzerland      Western Europe      7.5599
```

3	Iceland	Western Europe	7.5045
4	Norway	Western Europe	7.4880

	Standard error of ladder score	upperwhisker	lowerwhisker \
0	0.031156	7.869766	7.747634
1	0.033492	7.711245	7.579955
2	0.035014	7.628528	7.491272
3	0.059616	7.621347	7.387653
4	0.034837	7.556281	7.419719

	Logged GDP per capita	Social support	Healthy life expectancy \
0	10.639267	0.954330	71.900825
1	10.774001	0.955991	72.402504
2	10.979933	0.942847	74.102448
3	10.772559	0.974670	73.000000
4	11.087804	0.952487	73.200783

	Freedom to make life choices	Generosity	Perceptions of corruption \
0	0.949172	-0.059482	0.195445
1	0.951444	0.066202	0.168489
2	0.921337	0.105911	0.303728
3	0.948892	0.246944	0.711710
4	0.955750	0.134533	0.263218

	Ladder score in Dystopia	Explained by: Log GDP per capita \
0	1.972317	1.285190
1	1.972317	1.326949
2	1.972317	1.390774
3	1.972317	1.326502
4	1.972317	1.424207

	Explained by: Social support	Explained by: Healthy life expectancy \
0	1.499526	0.961271
1	1.503449	0.979333
2	1.472403	1.040533
3	1.547567	1.000843
4	1.495173	1.008072

	Explained by: Freedom to make life choices	Explained by: Generosity \
0	0.662317	0.159670
1	0.665040	0.242793
2	0.628954	0.269056
3	0.661981	0.362330
4	0.670201	0.287985

	Explained by: Perceptions of corruption	Dystopia + residual
0	0.477857	2.762835

1	0.495260	2.432741
2	0.407946	2.350267
3	0.144541	2.460688
4	0.434101	2.168266

Pemanggilan `df.head` digunakan untuk menampilkan 5 baris data pertama

Mengganti Nama Kolom/Atribut Dalam Python, dengan menggunakan `DataFrame`, maka kita dapat mengganti nama atribut atau nama kolom menjadi lebih singkat atau bentuk apapun yang diinginkan. Hal ini biasanya dilakukan untuk menyederhanakan penamaan nama atribut atau kolom dari dataset asli sebelumnya.

```
[31]: #Mengganti nama kolom
df.rename(columns={
    'Country name' : 'Country',
    'Perceptions of corruption' : 'Corruption',
    'Freedom to make life choices' : 'Freedom'
}, inplace = True)

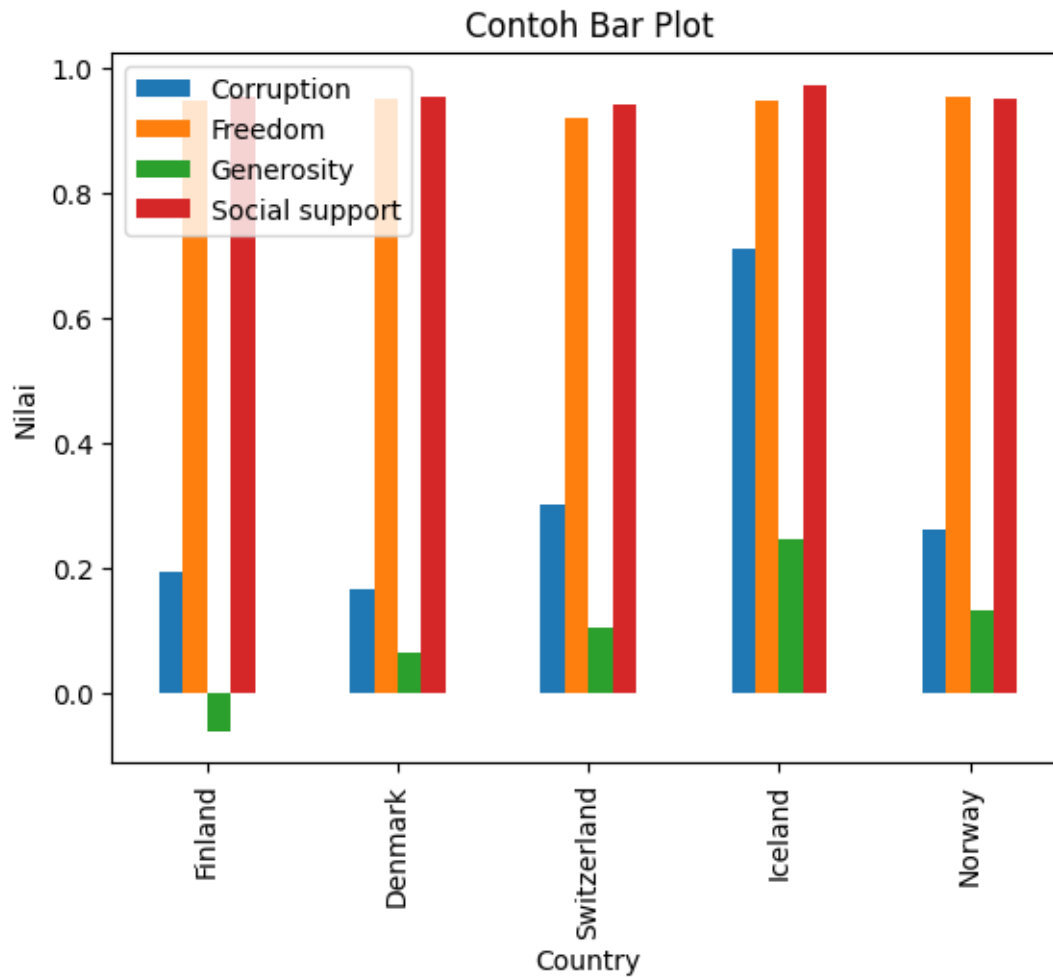
df.columns
```

```
[31]: Index(['Country', 'Regional indicator', 'Ladder score',
    'Standard error of ladder score', 'upperwhisker', 'lowerwhisker',
    'Logged GDP per capita', 'Social support', 'Healthy life expectancy',
    'Freedom', 'Generosity', 'Corruption', 'Ladder score in Dystopia',
    'Explained by: Log GDP per capita', 'Explained by: Social support',
    'Explained by: Healthy life expectancy',
    'Explained by: Freedom to make life choices',
    'Explained by: Generosity', 'Explained by: Perceptions of corruption',
    'Dystopia + residual'],
    dtype='object')
```

Membentuk Bar Plot

```
[32]: df[:5].plot(x='Country',
    y=['Corruption', 'Freedom', 'Generosity', 'Social support'],
    kind='bar')

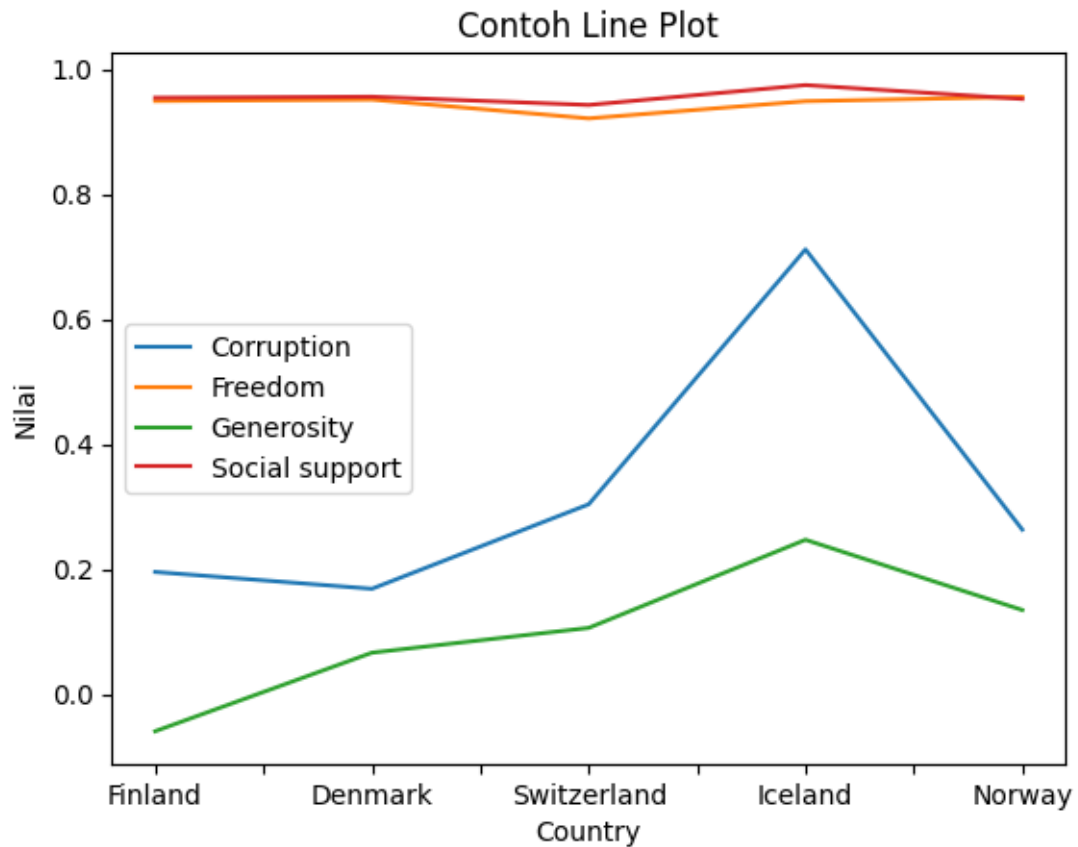
plt.title('Contoh Bar Plot')
plt.ylabel('Nilai')
plt.show()
```



Membentuk Line Plot

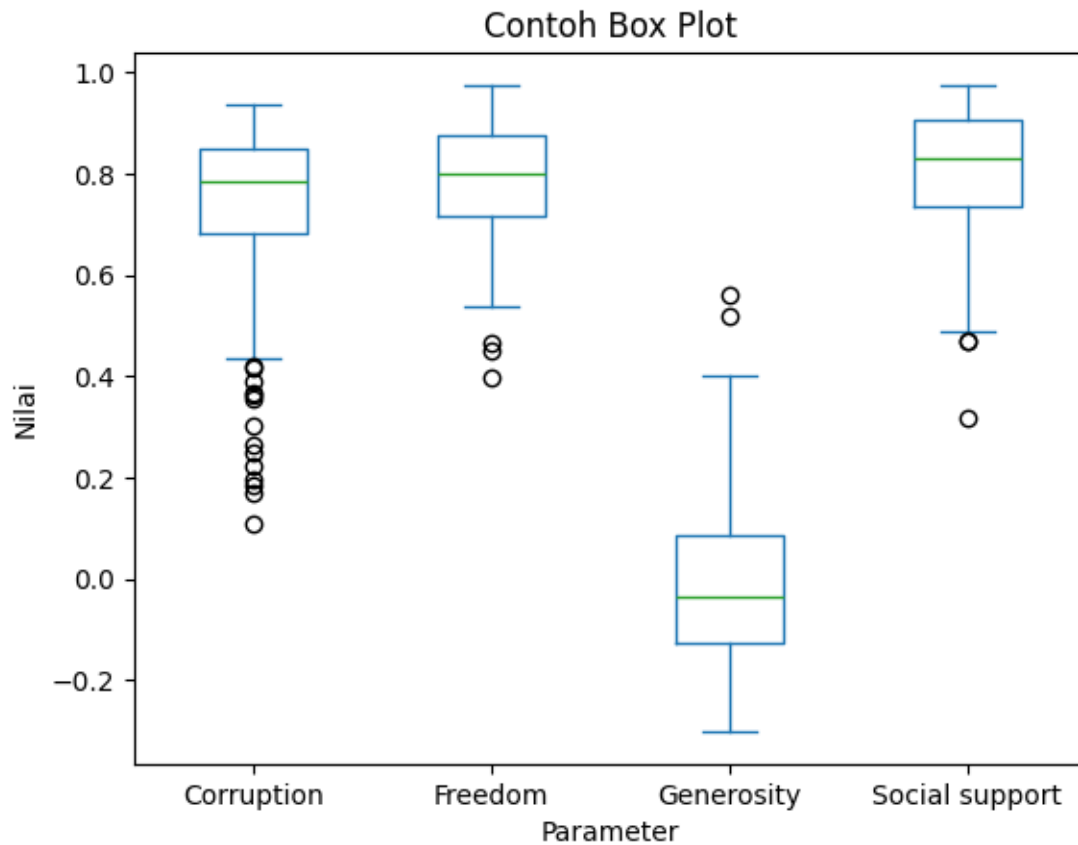
```
[33]: df[:5].plot(x='Country',
                 y=['Corruption', 'Freedom', 'Generosity', 'Social support'],
                 kind='line')

plt.title('Contoh Line Plot')
plt.ylabel('Nilai')
plt.show()
```



Membentuk Box Plot

```
[34]: df.plot(y=['Corruption', 'Freedom', 'Generosity', 'Social support'],  
           kind='box')  
  
plt.title('Contoh Box Plot')  
plt.xlabel('Parameter')  
plt.ylabel('Nilai')  
plt.show()
```

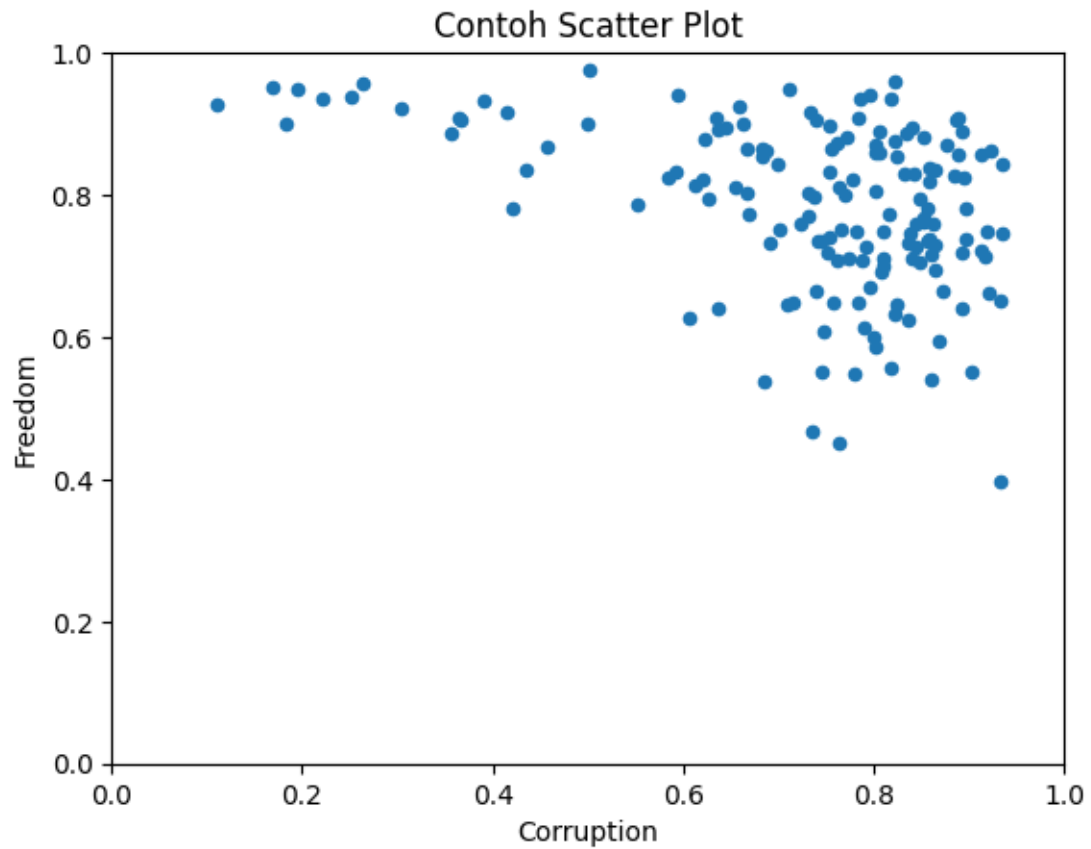


Membentuk Scatter Plot

```
[35]: df.plot(x='Corruption', y='Freedom', kind='scatter')

plt.xlim((0,1))
plt.ylim((0,1))

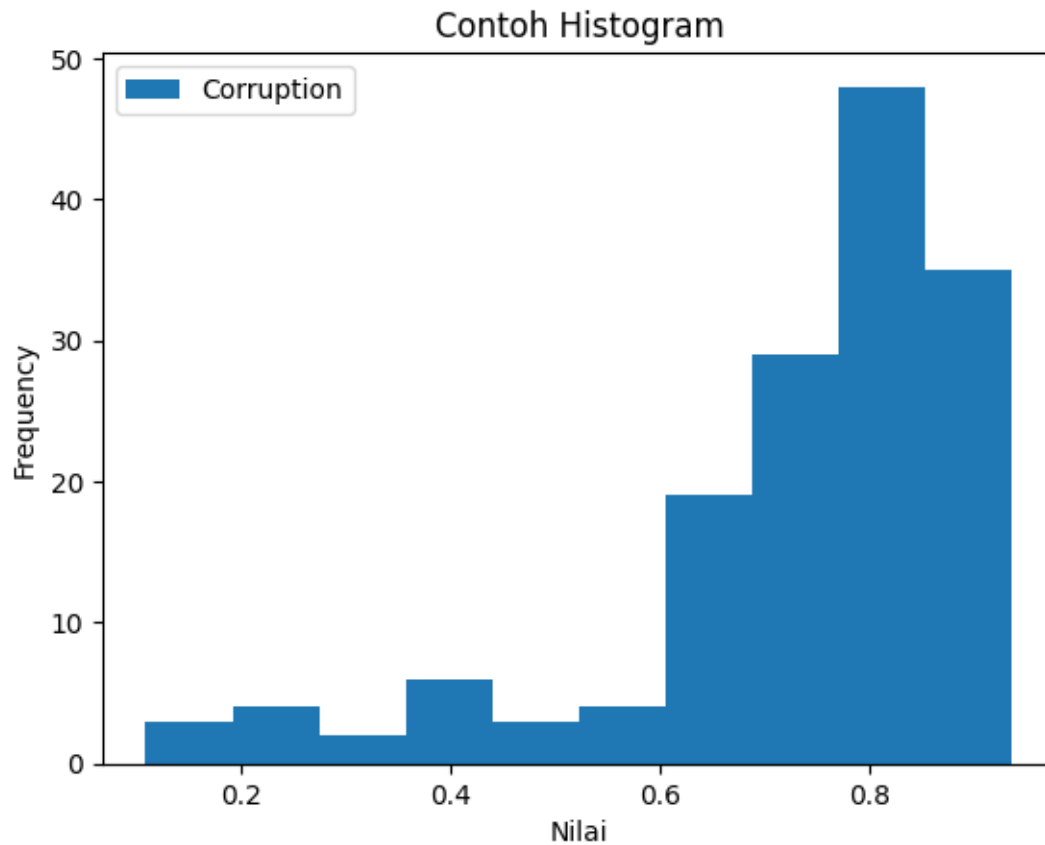
plt.title('Contoh Scatter Plot')
plt.show()
```

Membentuk Histogram

```
[36]: df.plot(x='Country', y='Corruption', kind='hist', bins=10)

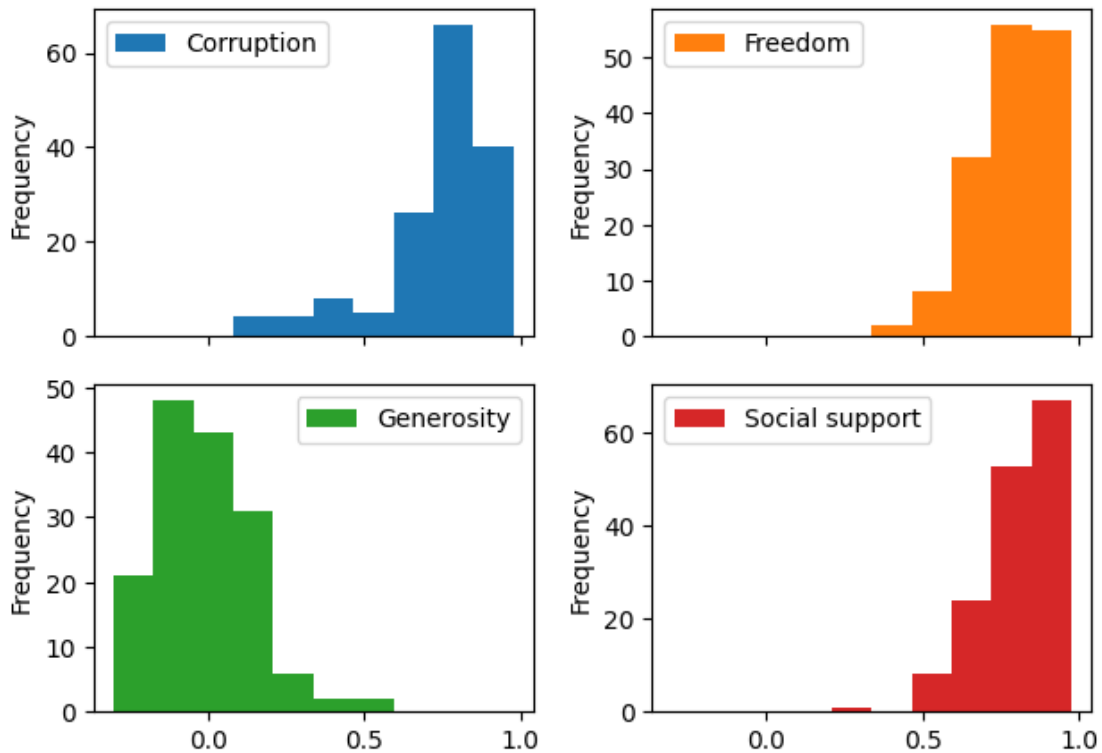
plt.title('Contoh Histogram')
plt.xlabel('Nilai')
plt.show()
```



Membentuk Subplots

```
[37]: df.plot(x='Country',  
             y=['Corruption', 'Freedom', 'Generosity', 'Social support'],  
             kind='hist',  
             subplots=True,  
             layout=(2,2))  
  
plt.suptitle('Contoh Subplots')  
plt.tight_layout()  
plt.show()
```

Contoh Subplots



1.4 Soal Latihan/Tugas Praktikum

1.4.1 Soal

1. Cari salah satu dataset pada kaggle.com, misalnya Netflix Movies and TV Shows (<https://www.kaggle.com/datasets/shivamb/netflix-shows>), atau jenis dataset lainnya
2. Lakukan proses pembacaan file CSV
3. Manipulasi data dan kolom pada file tersebut menggunakan numpy dan pandas
4. Visualisasikan apa yang anda peroleh dari dataset tersebut
5. Lakukan sekreatif mungkin

1.4.2 Instruksi

1. Jawaban dikumpul dalam format pdf
2. Jabarkan output dan hasil eksperimen yang telah diperoleh
3. Tambahkan referensi yang digunakan dalam memperkaya analisa saudara
4. Tambahkan pula penjelasan mengenai hal-hal apa yang baru anda pelajari melalui Modul 2 ini dan apa tantangan yang didapatkan selama mengerjakan materi di Modul dan Tugas