

VERSION 2.2  
17 NOVEMBER 2024



# [PEMROGRAMAN FUNGSIONAL]

Modul 5 - Functional Programming Implementation:  
for Data Visualization

DISUSUN OLEH:  
ALVIYA LAELA LESTARI  
RAFLI KHARISMA AKBAR

DIAUDIT OLEH:  
FERA PUTRI AYU L., S.KOM., M.T.

LAB. INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## [PEMROGRAMAN FUNGSIONAL]

---

### PERSIAPAN MATERI

1. Praktikan diharapkan telah memahami materi pada modul-modul sebelumnya.
2. Praktikan diharapkan juga mencari sumber belajar eksternal mengenai bahasa python.

---

### TUJUAN

Sub-CPMK 7: Mahasiswa mampu mendesain program dengan teknik yang tepat untuk menyelesaikan masalah dengan menggunakan paradigma pemrograman fungsional (P6)

---

### TARGET MODUL

1. Mampu memahami dan mengimplementasikan beberapa python library untuk visualisasi data dengan memanfaatkan teknik pemrograman fungsional
2. Mampu menentukan Function yang tepat dalam prosesing data hingga visualisasi berdasarkan teknik pemrograman fungsional

---

### PERSIAPAN SOFTWARE/APLIKASI

1. Komputer/Laptop
2. Sistem operasi Windows/Linux/Mac OS/Android
3. Pycharm/Google Collab/ Jupyter Notebook
4. [Source Code Google Colab Modul 5](#)

# 1. Library pada Python

Pada modul ini, kita akan menggunakan berbagai library yang berkaitan dengan statistika dan visualisasi data untuk mengolah data dan menampilkan informasi secara efektif. Python adalah salah satu bahasa pemrograman yang sangat populer di kalangan data scientist dan data analyst karena memiliki berbagai library yang kuat untuk statistik dan visualisasi data.

Python sudah memiliki library built-in, yang mana artinya library ini sudah siap digunakan setelah Python telah terinstall. Built-in library ini dapat kalian baca [di sini](#). Dalam penggunaannya, library ini harus kita panggil jika ingin digunakan. Untuk menggunakan library ini, kita bisa menggunakan sintaks **import**. Cara lain bisa kalian pelajari [di sini](#).

Dalam bidang data science, adanya library ini membuat pemrograman Python menjadi lebih sederhana dan nyaman bagi programmer karena tidak perlu menulis ulang kode yang sama untuk program yang berbeda.

## 1.1 Matplotlib

Matplotlib diciptakan untuk membuat visualisasi data yang menawarkan banyak pilihan untuk berbagai jenis grafik beserta penyesuaiannya. Library ini gratis dan merupakan open source library. Kita dapat langsung mencoba melakukan import untuk mengecek apakah library matplotlib sudah tersedia dalam built-in package kita atau belum dengan cara menjalankan kode berikut:

```
import matplotlib
```

Jika tidak terjadi error, maka library siap digunakan. Jika belum, maka kalian perlu menginstall library matplotlib terlebih dahulu. Untuk menginstall library ini, kalian bisa membuka command prompt atau CMD dan run command:

```
pip install matplotlib
```

### a. Percobaan 1: Plotting menggunakan pyplot

Sebuah package pyplot dalam library matplotlib pada kode berikut ini digunakan secara alias (as) sebagai plt dan dapat dimanfaatkan untuk

menggambar sebuah garis. Penggunaan alias ini sangat memudahkan kita dalam menyingkat nama package yang panjang menjadi lebih ringkas penulisannya.

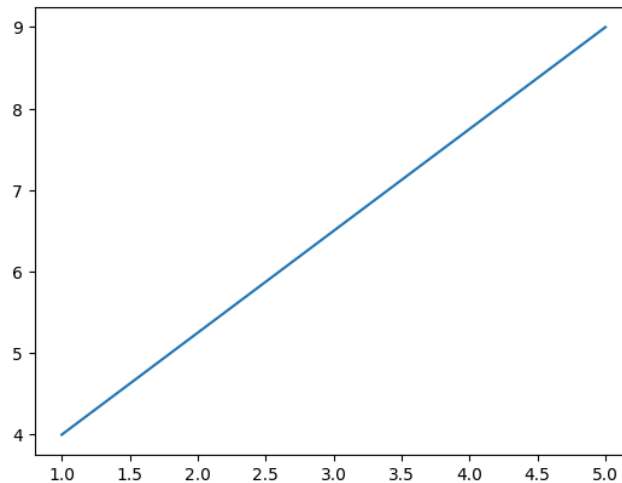
```
import matplotlib.pyplot as plt

plt.plot([1, 5], [4, 9])
plt.show()
```

```
# cara lain untuk melakukan import dan alias
from matplotlib import pyplot as plt

plt.plot([1, 5], [4, 9])
plt.show()
```

Output:



Pada kode di atas, setelah kita mengimport modul 'pyplot' as 'plt', kita menggunakan fungsi 'plt.plot()' untuk membuat grafik garis dan membuat argumen '[1, 5]' sebagai daftar koordinat x dan argumen '[4, 9]' sebagai daftar koordinat y. Fungsi ini membuat grafik dengan titik-titik (1, 4) dan (5, 9), dan menggambarkan garis yang melewati kedua titik tersebut. Fungsi 'plt.show()' digunakan untuk menampilkan grafik yang telah dibuat.

#### **b. Percobaan 2: Plotting menggunakan Scatter Plot**

Selain menggunakan library pyplot dalam matplotlib untuk membuat plot garis dan grafik seperti pada percobaan sebelumnya, kita juga dapat membuat visualisasi data menggunakan scatter plot.

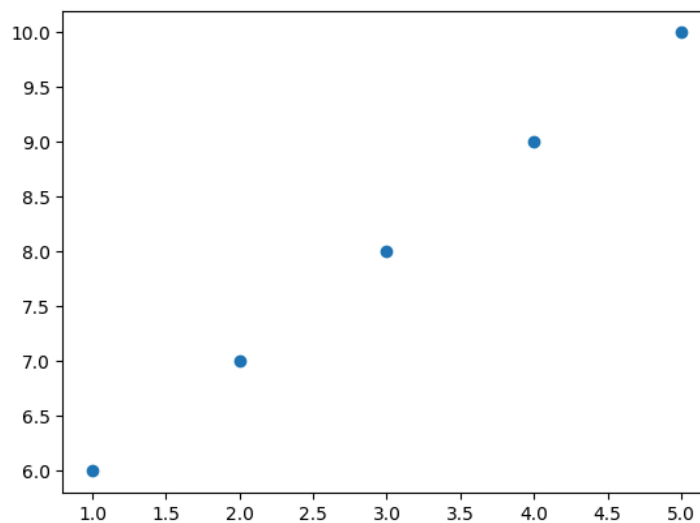
**Scatter plot** adalah salah satu jenis visualisasi data yang digunakan untuk menggambarkan hubungan antara dua variabel. Scatter plot memungkinkan kita untuk memahami apakah terdapat korelasi, pola atau hubungan lain di antara data yang diamati. Scatter plot efektif dalam menggambarkan data numerik dan membantu mengidentifikasi tren atau anomali yang mungkin tidak terlihat dari data mentah.

```
import matplotlib.pyplot as plt

garis_x = [1, 2, 3, 4, 5]
garis_y = [6, 7, 8, 9, 10]

plt.scatter(garis_x, garis_y)
plt.show()
```

Output:



Dalam kode ini, kita memiliki dua array yaitu 'garis\_x' dan 'garis\_y'. Masing-masing array berisi nilai koordinat x dan y untuk setiap titik yang akan ditampilkan pada fungsi plt.show().

### c. Percobaan 3: Visualisasi data menggunakan Pie Chart

Dalam dunia analisis data, visualisasi adalah salah satu cara yang paling efektif untuk menyampaikan informasi dengan cepat dan mudah dipahami. Selain dua plotting yang sudah dibahas sebelumnya terdapat juga Pie Chart atau diagram lingkaran.

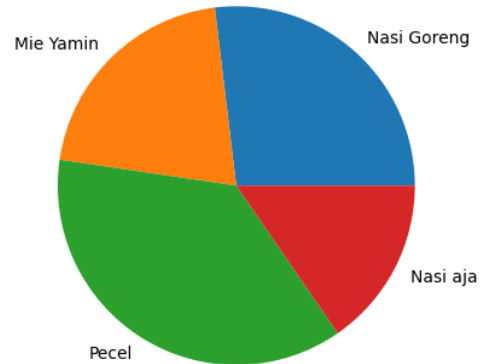
**Pie chart** adalah jenis diagram yang menggambarkan data dalam bentuk lingkaran, dimana lingkaran tersebut dibagi menjadi irisan sesuai proporsi masing-masing kategori. Setiap bagian dari lingkaran mewakili kontribusi suatu kategori terhadap total keseluruhan.

```
import matplotlib.pyplot as plt

favorite_food = {
    'Nasi Goreng' : 35,
    'Mie Yamin' : 27,
    'Pecel' : 48,
    'Nasi aja' : 20
}

colors = list(favorite_food.keys())
values = list(favorite_food.values())

plt.pie(values, labels=colors)
plt.show()
```



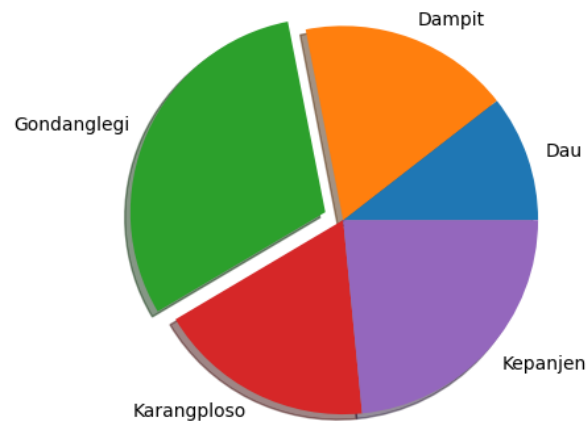
Dalam kode di atas, terdapat sebuah dictionary 'favorite\_food' yang berisi jumlah mahasiswa yang menyukai makanan sesuai yang telah disebutkan. Dictionary tersebut kemudian dipecah menjadi pasangan key-value agar lebih mudah untuk visualisasi. Setelah itu, pie chart dibuat dengan menggunakan 'plt.pie()'.

Dalam penggunaannya, kita juga dapat menonjolkan atau memisahkan salah satu atau lebih irisan dari lingkaran utama menggunakan keyword **explode**. Fitur ini sering digunakan untuk menyoroti kategori tertentu yang dianggap penting atau signifikan dalam data. Berikut adalah contoh kodenya:

```
jumlah_warga = [200, 333, 578, 342, 445]
kecamatan = ['Dau', 'Dampit', 'Gondanglegi', 'Karangploso',
             'Kapanjen']
explode = [0, 0, 0.1, 0, 0]

plt.pie(jumlah_warga, labels=kecamatan, explode=explode, shadow=True)
plt.show()
```

Output:



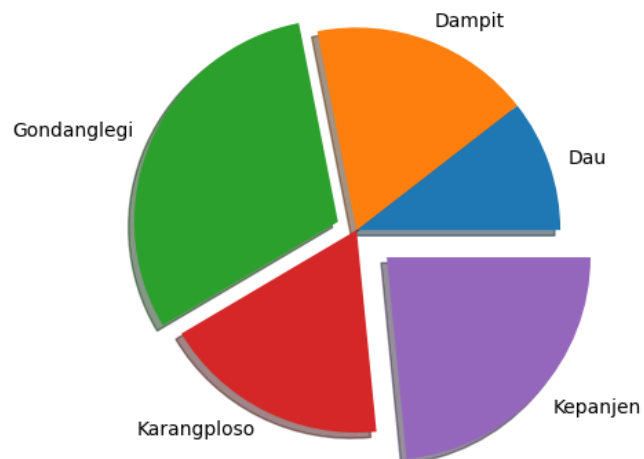
Dalam kode tersebut terdapat data jumlah warga setiap kecamatan yang ada di Kab. Malang. Dari data tersebut, diketahui bahwa Kec. Gondanglegi memiliki jumlah warga terbanyak dari 4 kecamatan lainnya sehingga bisa kategori ini bisa disorot. Selain Gondanglegi, kalian juga bisa menyorot kategori lain dengan menambah nilai positif pada indeks kategori yang akan disorot.

- Nilai 0 berarti irisan tetap berada di lingkaran.
- Nilai positif (misalnya 0.1 atau 0.2) akan menarik irisan keluar.

Coba ubah nilai explode menjadi:

```
explode = [0, 0, 0.1, 0, 0.2]
```

Dan perhatikan hasilnya!



#### d. Percobaan 4: Subplot dalam matplotlib

Subplot adalah fitur dalam matplotlib yang memungkinkan kita membuat beberapa plot dalam satu figur. Figur ini akan sangat berguna ketika kita ingin:

- Membandingkan beberapa dataset secara langsung
- Membuat visualisasi yang terdiri dari berbagai jenis grafik dalam satu tampilan

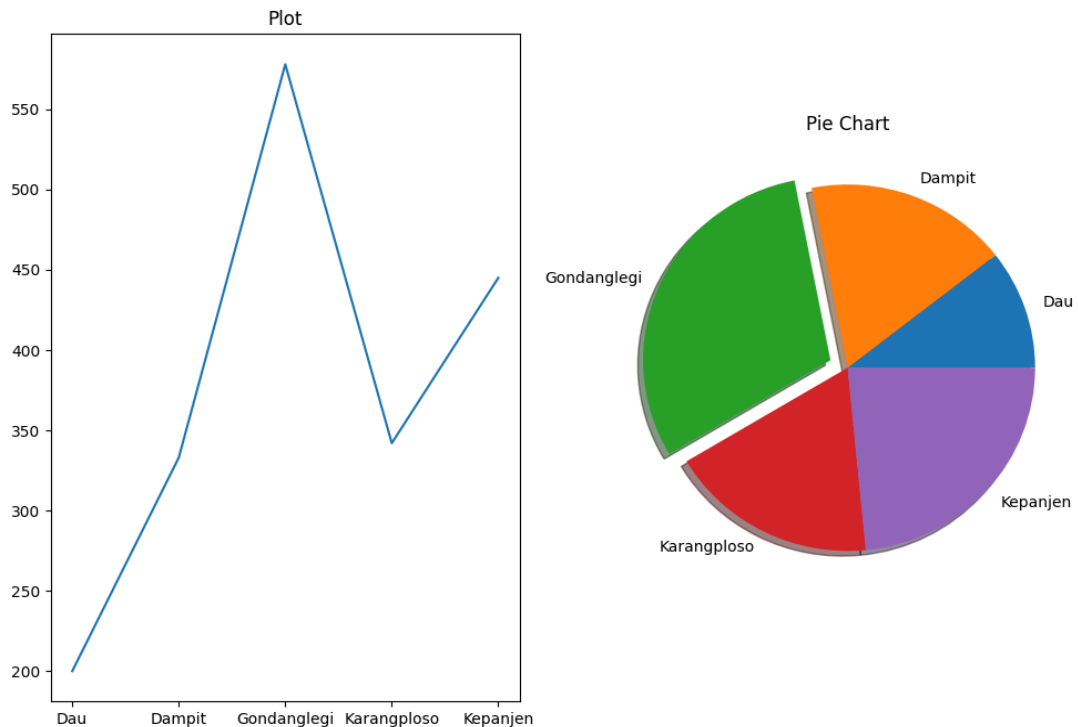
Berikut adalah contoh implementasinya:

```
jumlah_warga = [200, 333, 578, 342, 445]
kecamatan = ['Dau', 'Dampit', 'Gondanglegi', 'Karangploso', 'Kepanjen']
explode = [0, 0, 0.1, 0, 0]

plt.figure(figsize=(12, 8))
plt.subplot(1, 2, 1)
plt.title("Plot")
plt.plot(kecamatan, jumlah_warga)

plt.subplot(1, 2, 2)
plt.title("Pie Chart")
plt.pie(jumlah_warga, labels=kecamatan, explode=explode, shadow=True)
plt.show()
```

Output:





- `plt.subplot(1, 2, 1)` : the figure has 1 row, 2 columns, and this plot is the first plot.
- `plt.subplot(1, 2, 2)` : the figure has 1 row, 2 columns, and this plot is the second plot.

Kalian dapat menambahkan lebih banyak diagram dalam subplot dengan memodifikasi nilai parameter di dalamnya (jumlah baris, kolom, dan urutan gambar/diagram).

## 1.2 NumPy

NumPy (Numerical Python) adalah library yang digunakan untuk komputasi numerik. NumPy menyediakan objek array multidimensi yang efisien dan fleksibel, serta berbagai fungsi matematika yang dapat digunakan untuk melakukan operasi pada array tersebut. NumPy merupakan salah satu library dasar dalam ekosistem komputasi ilmiah Python, dan sering digunakan dalam analisis data, visualisasi, pemrosesan citra, dan berbagai aplikasi ilmiah lainnya.

NumPy juga sering digunakan dalam visualisasi data, terutama ketika digunakan bersama dengan library lain seperti Matplotlib, Seaborn, atau Plotly. NumPy menyediakan array yang kompatibel dengan banyak library visualisasi, sehingga memudahkan untuk memasukkan data dalam bentuk yang sesuai. Untuk penjelasan lebih lanjut dan dokumentasi resmi tentang NumPy, kalian dapat mengunjungi situs web resmi NumPy di [link berikut](#).

Untuk menginstall library ini, kalian bisa membuka command prompt atau CMD dan run command: `pip install numpy`

Berikut adalah contoh penggunaan NumPy dalam visualisasi data menggunakan Matplotlib:

### a. Percobaan 5: Variasi visualisasi data

Dalam program di bawah, kita akan menambahkan beberapa variasi tambahan untuk memperkaya visualisasi data kita.

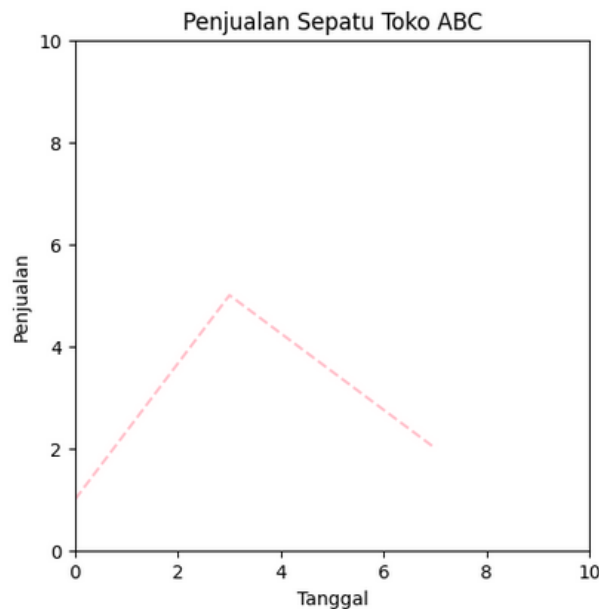
```
import matplotlib.pyplot as plt
import numpy as np
```

```

point_x = np.array([0, 3, 7])
point_y = np.array([1, 5, 2])
plt.figure(figsize = (10, 10))
plt.plot(point_x, point_y, color='pink', linestyle = '--')
plt.xlim([0, 10])
plt.ylim([0, 10])
plt.title("Penjualan Sepatu Toko ABC")
plt.xlabel("Tanggal")
plt.ylabel("Penjualan")
plt.show()

```

Output:



Seperti yang sudah dijelaskan sebelumnya, fungsi 'plt.plot()' digunakan untuk menggambar titik (marker) pada diagram. Secara default, fungsi ini menarik garis dari titik satu ke titik lain. Fungsi ini mengambil parameter untuk menentukan titik dalam diagram. Parameter pertama adalah larik (array) yang berisi titik-titik pada sumbu x. Parameter kedua adalah larik (array) yang berisi titik-titik pada sumbu y. Selain itu, kita juga bisa menambahkan beberapa atribut/variasi lain terhadap garis.

- plt.figure: Mengatur ukuran plot. Di sini kita atur menjadi 10 x 10 inci.
- color: Mengatur warna garis. Di sini kita atur menjadi berwarna pink.
- linestyle: Mengatur gaya garis. Di sini kita atur menjadi garis putus-putus ('--').

- xlim dan ylim: Mengatur batas sumbu x dan y. Di sini kita mengatur dari 0 hingga 10.
- xlabel dan ylabel: Memberi nama atau label pada tiap sumbu. Di sini kita memberi sumbu x label "Tanggal" dan sumbu y label "Penjualan".

Selengkapnya bisa kalian pelajari melalui [link berikut](#).

### b. Percobaan 6: Multiple lines

Kita dapat memplot garis sebanyak yang kita inginkan dengan menambahkan lebih banyak fungsi plt.plot() untuk setiap garis!

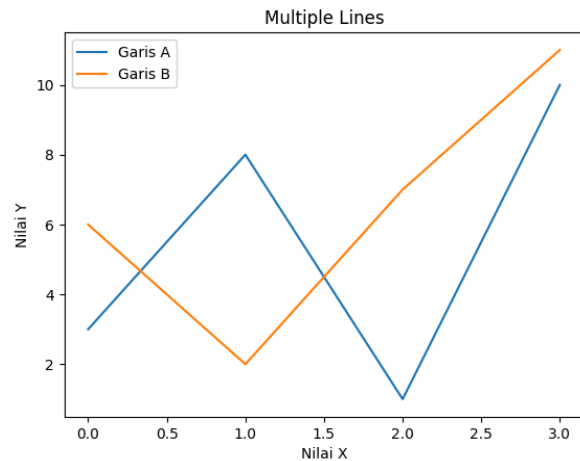
```
import matplotlib.pyplot as plt
import numpy as np

garis_a = np.array([3, 8, 1, 10])
garis_b = np.array([6, 2, 7, 11])

plt.plot(garis_a, label="Garis A")
plt.plot(garis_b, label="Garis B")

plt.title("Multiple Lines")
plt.xlabel("Nilai X")
plt.ylabel("Nilai Y")

plt.legend()
plt.show()
```



## 1.3 Pandas

Pandas adalah open-source library Python yang sangat powerful yang digunakan untuk manipulasi dan analisis data. Pandas berisi struktur dan fungsi data untuk melakukan operasi data yang efisien. Library ini digunakan bersama library lain untuk data science. Banyak struktur library NumPy digunakan atau direplikasi di Pandas.

Data yang dihasilkan oleh Pandas sering digunakan sebagai input untuk plotting di Matplotlib, analisis statistik di SciPy, dan algoritma pembelajaran di Scikit-learn.

Pandas memiliki dua struktur data, yaitu series dan dataframe.

- Series: Struktur satu dimensi yang bisa memuat data dari berbagai tipe (integer, string, float, dll.). Dapat diakses menggunakan indeks.
- DataFrame: Struktur data dua dimensi yang berbentuk tabel, dengan baris dan kolom. DataFrame merupakan gabungan dari beberapa Series.

Untuk menginstall library ini, kalian bisa membuka command prompt atau CMD dan run command: `pip install pandas`

### 1.3.1 Pandas Series

Pandas series adalah struktur data satu dimensi yang mampu menampung data jenis apapun (integer, string, float, objek Python, dll). Pandas series lebih sering hanya sebuah kolom di Excel. Label tidak harus unik tapi harus *hashable*.

Pandas series dibuat dengan memuat kumpulan data dari penyimpanan yang ada (bisa berupa database SQL, file CSV, atau file Excel). List, dictionary, nilai skalar juga dapat menggunakan Pandas Series.

#### a. Membuat Series Menggunakan Array Sederhana

Salah satu cara membuat Series adalah dengan menggunakan array sederhana, seperti array dari NumPy. Pandas Series memiliki indeks di sebelah kiri dan data di sebelah kanan, dimana indeks secara default berupa angka mulai dari 0, kecuali jika ditentukan secara eksplisit.

```
import pandas as pd
import numpy as np

# Membuat series kosong
series = pd.Series()
print("Pandas Series: ", series)

# Membuat array simple
data = np.array(['f', 'u', 'n', 'g', 's', 'i', 'o', 'n', 'a', 'l'])

# Memasukkan array ke dalam series
series = pd.Series(data)
print("Pandas Series:\n", series)

# Mengakses label berdasarkan indeks (Ingat! Indeks dimulai dari 0)
print("\nLabel ke-4: ", series[4])
```

Output:

```
Pandas Series: Series([], dtype: object)
Pandas Series:
0    f
1    u
2    n
3    g
4    s
5    i
6    o
7    n
8    a
9    l
dtype: object
Label ke-4: s
```

### b. Percobaan 7: Membuat Series dari Dictionary

Dalam Pandas, kita dapat dengan mudah membuat Series dari berbagai sumber data, salah satunya adalah dari dictionary. Dengan menggunakan dictionary untuk membuat Series, kita dapat dengan cepat dan efisien mengorganisir data yang terstruktur.

```
import pandas as pd

biodata_umur = {"Angela": 24, "Bianca": 21, "Chloe": 18}
result = pd.Series(biodata_umur)
print(result)
```

Output:

```
Angela    24
Bianca    21
Chloe     18
dtype: int64
```

### 1.3.2 Dataframe Pandas

Dataframe Pandas adalah struktur data dua dimensi dengan sumbu belabel (baris dan kolom). Dataframe ini dibuat dengan memuat kumpulan data dari storage yang ada (bisa berupa database SQL, file CSV, atau file Excel). Dataframe Pandas dapat dibuat dari list, dictionary, kamus, list of dictionaries, dll.

### a. Membuat Dataframe dari Array Sederhana

Membuat Dataframe dari Array Sederhana adalah salah satu cara dasar dalam menggunakan library Pandas untuk memanipulasi dan menganalisis data.

```
import pandas as pd

# Memanggil dataframe constructor
dataframe = pd.DataFrame()
print(dataframe)

# Membuat list berisi string
words = ['kita', 'bisa', 'kita', 'pasti', 'bisa']

# Memanggil dataframe constructor pada list
dataframe = pd.DataFrame(words)
print("\n", dataframe)
```

Output:

```
Empty DataFrame
Columns: []
Index: []

      0
0  kita
1  bisa
2  kita
3  pasti
4  bisa
```

### b. Percobaan 8: Membuat Dataframe dari Dictionary Pandas

Dictionary di Python adalah struktur data yang terdiri dari pasangan key dan value, di mana key dapat digunakan sebagai nama kolom di dalam DataFrame. Dengan menggunakan dictionary, pengguna dapat dengan mudah mengorganisasi data dan membuat DataFrame yang rapi, memungkinkan manipulasi dan analisis data yang lebih efisien.

```
import pandas as pd

# Memanggil dataframe constructor
myvar = pd.DataFrame()
print(myvar)

# Membuat dictionary
mydataset = {
    'Matkul': ["Mobile", "Fungsional", "Jaringan Komputer", "Piranti Cerdas"],
    'Nilai': [78, 90, 85, 95]
}
```

```
# Memanggil dataframe constructor pada list
myvar = pd.DataFrame(mydataset)
print(myvar)
```

Output:

```
Empty DataFrame
Columns: []
Index: []

   Matkul  Nilai
0   Mobile    78
1  Fungsional  90
2 Jaringan Komputer  85
3  Piranti Cerdas  95
```

### c. Percobaan 9: Membaca Data dengan Pandas

Salah satu langkah penting dalam analisis data adalah kemampuan untuk membaca dan mengolah data dari berbagai sumber. Pandas, sebagai library yang kuat untuk analisis data di Python, menyediakan berbagai fungsi untuk memudahkan proses ini. Salah satu format data yang paling umum digunakan adalah CSV (Comma-Separated Values), yang memungkinkan data disimpan dalam bentuk tabel yang dapat dibaca oleh berbagai aplikasi.

Dalam percobaan kali ini, silahkan download file CSV [berikut](#). Source: w3schools. Berikut cara untuk import file ke Google Collab:

1. Download file di atas, atau file lain yang akan kita import ke Google Collab

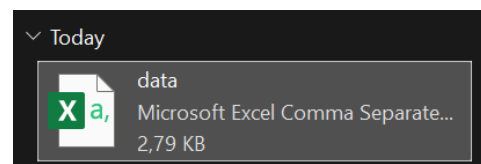
2. Klik icon berikut →



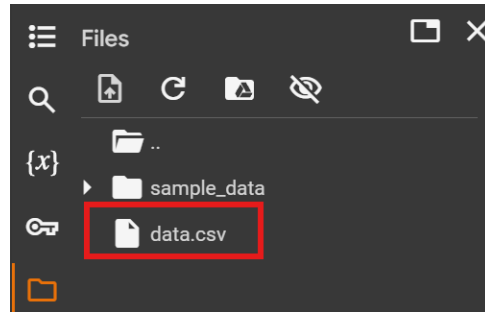
3. Kemudian klik icon paling kiri →



4. Pilih file yang akan diimpor →



5. File yang sudah diimpor akan terlihat seperti berikut:



Setelah berhasil mengimport file, selanjutnya kita akan membaca data dari file tersebut dengan kode berikut:

```
import pandas as pd

# Membaca file CSV
data = pd.read_csv('data.csv')

# Menampilkan 5 baris pertama
print(data.head())
```

Output:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0

#### d. Percobaan 10: Operasi Dasar pada DataFrame

Berikutnya kita akan mencoba beberapa operasi dasar pada DataFrame yang telah kita siapkan dari percobaan 4 sebelumnya.

##### - Menampilkan informasi umum tentang Dataframe

Kita dapat menggunakan fungsi `info()` untuk mendapatkan informasi

```
data.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Duration    169 non-null    int64
1   Pulse       169 non-null    int64
```



```

2    Maxpulse  169 non-null    int64
3    Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB

```

Dari syntax ini, kita dapat mengetahui bahwa terdapat 169 data entries yang terdapat pada file tersebut, mulai dari indeks 0 hingga 168. Terdapat 4 kolom beserta nama kolom. Ada tipe data. Ada jumlah non-null yang membantu memahami keberadaan data yang hilang (missing values). Selain itu juga terdapat informasi tentang penggunaan memori, yang menunjukkan seberapa banyak memori yang diperlukan untuk menyimpan data tersebut.

#### - Mengakses info statistik dasar dari data

```
print(data.describe())
```

Output:

	Duration	Pulse	Maxpulse	Calories
count	169.000000	169.000000	169.000000	164.000000
mean	63.846154	107.461538	134.047337	375.790244
std	42.299949	14.510259	16.450434	266.379919
min	15.000000	80.000000	100.000000	50.300000
25%	45.000000	100.000000	124.000000	250.925000
50%	60.000000	105.000000	131.000000	318.600000
75%	60.000000	111.000000	141.000000	387.600000
max	300.000000	159.000000	184.000000	1860.400000

Fungsi ini akan memberikan ringkasan statistik deskriptif dari kolom-kolom numerik di dalam DataFrame. Fungsi ini berguna untuk mendapatkan pemahaman awal tentang data, terutama untuk analisis data eksploratif. Memang apa saja sih artinya? Kalian bisa searching sendiri ya, yang buat modul capek :(

#### - Memilih kolom tertentu

Dalam analisis data menggunakan Pandas, seringkali kita perlu mengakses atau memanipulasi kolom tertentu dari DataFrame. Memilih kolom tertentu memungkinkan kita untuk fokus pada data yang relevan dan melakukan analisis yang lebih mendalam. Pandas menyediakan

cara yang mudah untuk memilih kolom, baik menggunakan nama kolom dalam tanda kurung siku ([]) atau menggunakan atribut DataFrame.

```
print(data['Calories'])
```

Output:

```
0      409.1
1      479.0
2      340.0
3      282.4
4      406.0
...
164    290.8
165    300.0
166    310.2
167    320.4
168    330.4
Name: Calories, Length: 169, dtype: float64
```

#### - Memilih baris dengan kondisi tertentu

Memilih baris dengan kondisi memungkinkan kita untuk fokus pada subset data yang relevan, sehingga analisis yang dilakukan menjadi lebih spesifik dan informatif. Dengan menggunakan teknik ini, kita dapat mengekstrak data yang memenuhi kriteria tertentu dan melakukan analisis lebih lanjut terhadap data tersebut.

```
filtered_data = data[data['Maxpulse'] > 135]
print(filtered_data)
```

Output:

```
   Duration  Pulse  Maxpulse  Calories
1         60    117      145     479.0
3         45    109      175     282.4
4         45    117      148     406.0
6         60    110      136     374.0
10        60    103      147     329.3
..      ...    ...      ...      ...
164        60    105      140     290.8
165        60    110      145     300.0
166        60    115      145     310.2
167        75    120      150     320.4
168        75    125      150     330.4

[67 rows x 4 columns]
```

Dari sini terpantau ada 67 baris data yang memenuhi kondisi yang kita inginkan.

## CODELAB

### CODELAB 1

Eksplorasilah **Percobaan 1-6** dan implementasikan pada data berikut:

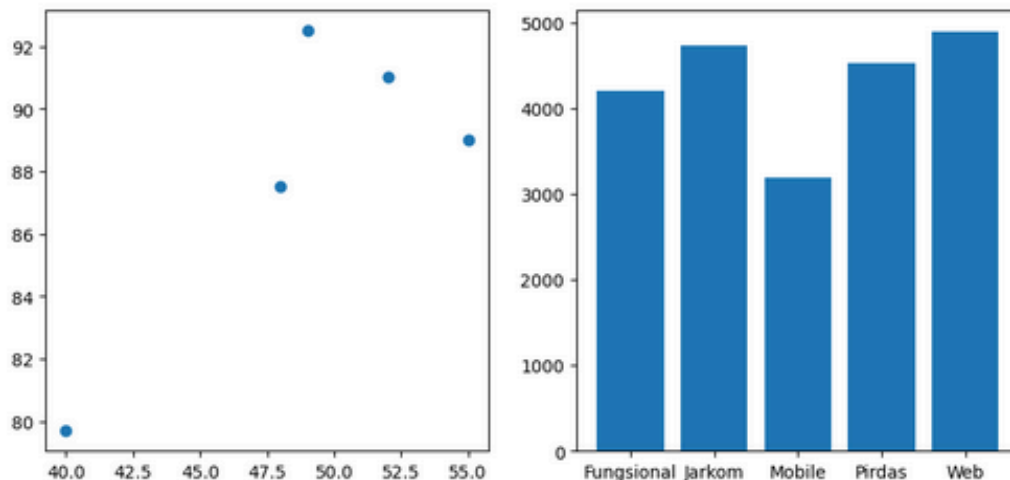
data sistem informasi mata kuliah dalam bentuk tuple, dimana setiap tuple berisi informasi 'mata\_kuliah', 'average\_nilai', 'jumlah\_mahasiswa'

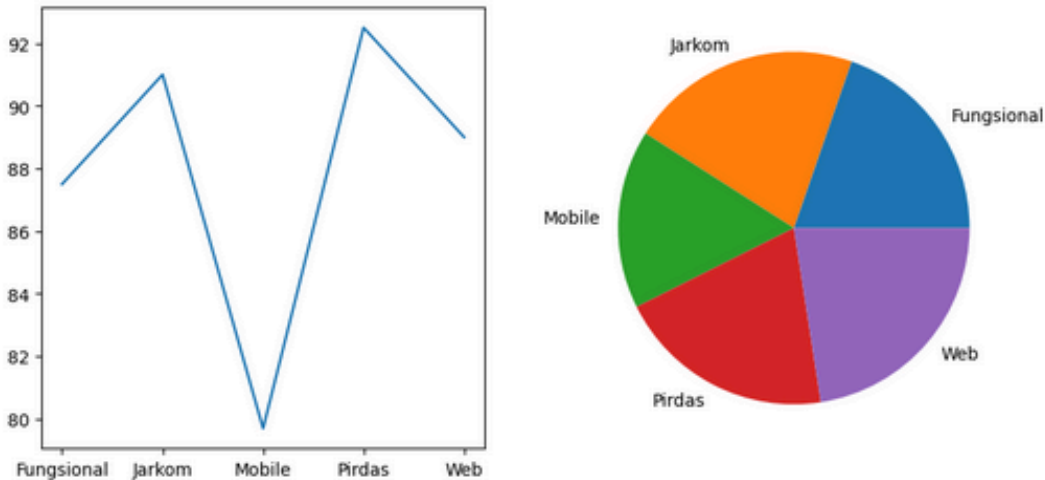
```
data_matkul = [  
    ("Fungsional", 87.5, 48),  
    ("Jarkom", 91, 52),  
    ("Mobile", 79.7, 40),  
    ("Pirdas", 92.5, 49),  
    ("Web", 89, 55)  
]
```

Kemudian modifikasilah **sesuai kreativitas kalian** menjadi satu kesatuan visualisasi data **yang menarik dan informatif!** Tunjukkan hasil modifikasi yang kalian lakukan pada asisten (program identik tidak akan dinilai). Jelaskan penggunaan masing-masing keyword yang kalian gunakan atau tambahkan.

Contoh Output:

```
Nama mata kuliah: ('Fungsional', 'Jarkom', 'Mobile', 'Pirdas', 'Web')  
Rata-rata nilai: (87.5, 91, 79.7, 92.5, 89)  
Jumlah mahasiswa: (48, 52, 40, 49, 55)
```





**\*) Output bisa divariasikan sesuai kreativitas masing-masing!**

Step:

1. Ekstrak data\_matkul untuk keperluan visualisasi data\*
2. Hitung total nilai untuk setiap mata kuliah\*
3. Buat scatter plot untuk menunjukkan hubungan antara rata-rata nilai dan jumlah mahasiswa
4. Buat diagram batang untuk menyajikan total nilai setiap mata kuliah
5. Buat diagram garis untuk menggambarkan rata-rata nilai tiap mata kuliah
6. Buat pie chart menggambarkan jumlah mahasiswa tiap mata kuliah dan explode data terbesar untuk nim ganjil dan data terkecil untuk nim genap.
7. Visualisasikan dengan subplot untuk menampilkan semua plot sekaligus.

Note\*): Manfaatkan list comprehension, map dan lambda untuk mengerjakan step 1&2

## CODELAB 2

1. Lakukan **Percobaan 7&8** dan implementasikan pada data yang pernah kalian miliki dari modul 1-4. Tunjukkan hasil modifikasi yang kalian lakukan pada asisten (program identik tidak akan dinilai). Jelaskan penggunaan masing-masing keyword yang kalian gunakan atau tambahkan.
2. Lakukan **Percobaan 9&10** sesuai di modul dan tunjukkan pada asisten setelah berhasil.

---

## TUGAS PRAKTIKUM

### TUGAS 1

Modifikasilah data yang telah kalian buat dari modul 1-4 dengan memvisualisasikannya sekreatif mungkin. Visualisasi data meliputi:

- Penggunaan plot garis, scatter plot, dan pie chart
- Penggunaan subplot untuk memvisualisasikan ketiga jenis plotting tersebut

**Poin penting:**

- Studi kasus yang sudah dipilih tidak boleh berubah
- Diperbolehkan untuk menambah atau memodifikasi data agar lebih variatif
- Poin plus untuk modifikasi plot (seperti mengubah warna, jenis garis, dll)
- Tidak boleh ada kesamaan dengan praktikan lain

### TUGAS 2

Unduh file data buah dan sayur melalui [link berikut](#). Lakukan operasi di bawah:

1. Bacalah data CSV tersebut menggunakan Pandas.
2. Tampilkan 10 data pertama.
3. Tampilkan informasi umum yang ada pada dataset tersebut. Jelaskan pada asisten!
4. Hitunglah rata-rata harga per tahun dari setiap produk!
5. Identifikasilah produk dengan harga tertinggi dan terendah!
6. Filterlah produk dengan harga antara 1.50 sampai 2.35!

## KRITERIA & DETAIL PENILAIAN

KETERANGAN		POIN	PROGRAM IDENTIK
Codelab 1		10	10
Codelab 2		5	5
Tugas 1	Pemahaman	20	10
	Kreativitas Kode/Kelengkapan Fitur	15	7
Tugas 2	Pemahaman	30	15
	Kreativitas Kode/Kelengkapan Fitur	20	10
Total		100	57

\*)Note: Program Identik berarti program sama persis dengan praktikan lain sehingga yang dinilai hanya pemahaman terhadap materi (code tidak mendapat bobot nilai sama sekali).

\*\*)Poin diatas merupakan poin maksimal yang bisa diperoleh. Asisten bisa memberikan nilai dibawah itu jika dirasa praktikan tidak maksimal saat demo (kurangnya pemahaman tentang apa yang di demokan).