

DATA ANALYTICS

BAB 4 : VISUALISASI DATA

1.1 Tujuan

Mahasiswa mengenal materi visualisasi data menggunakan Matplotlib pada bahasa pemrograman python.

1.2 Ulasan Materi

- ✚ **Visualisasi data** adalah proses mengubah data menjadi representasi grafis. Ini membantu dalam memahami pola, tren, dan hubungan dalam data lebih mudah. Python menawarkan beberapa pustaka yang kuat untuk melakukan ini.
- ✚ Berikut adalah beberapa pustaka Python populer untuk visualisasi data:

Pustaka Utama:

- **Matplotlib:** Ini adalah pustaka dasar untuk membuat berbagai jenis plot seperti garis, batang, scatter, pie, histogram, dan lainnya. Matplotlib memberikan kontrol penuh atas tampilan visualisasi.
- **Seaborn:** Dibangun di atas Matplotlib, Seaborn menyediakan antarmuka tingkat tinggi untuk membuat visualisasi statistik yang menarik dan informatif.
- **Plotly:** Memungkinkan pembuatan visualisasi interaktif yang dapat dibagikan secara online atau disematkan dalam aplikasi web.

Pustaka Lainnya:

- **ggplot2 (Plotnine):** Berdasarkan ggplot2 dari R, ini menawarkan sintaks yang mirip dan fokus pada estetika.
- **Bokeh:** Untuk membuat visualisasi interaktif yang dapat dijalankan di browser.
- **Altair:** Berfokus pada deklaratif visualisasi, memungkinkan pembuatan visualisasi dengan kode yang lebih ringkas.
- **Geoplotlib:** Khusus untuk visualisasi data geografis.

Pilihan Pustaka Berdasarkan Kebutuhan:

- **Visualisasi sederhana:** Matplotlib atau Seaborn
- **Visualisasi statistik:** Seaborn
- **Visualisasi interaktif:** Plotly atau Bokeh
- **Visualisasi geografis:** Geoplotlib

Banyak pustaka lain juga tersedia, masing-masing dengan kekuatan dan kelemahannya sendiri. Pemilihan pustaka terbaik tergantung pada jenis data, kompleksitas visualisasi yang diinginkan, dan tingkat pengalaman pengguna dengan Python.

Matplotlib Bars

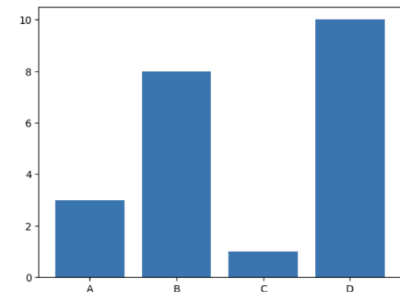
Dengan Pyplot, Anda dapat menggunakan fungsi `bar()` untuk menggambar grafik batang. Contoh :

Gambar 4 batang

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

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x, y)
plt.show()
```



- Fungsi `bar()` menerima argumen yang menggambarkan tata letak batang grafik. Kategori dan nilainya direpresentasikan oleh argumen pertama dan kedua dalam bentuk array.

- Nama Warna (Color Names)

Anda dapat menggunakan salah satu dari [140 warna yang didukung](#).

```
plt.bar(x, y, color = "hotpink")
```

- Kode Warna (Color Hex)

Or you can use [Hexadecimal color values](#):

```
plt.bar(x, y, color = "#4CAF50")
```

- Lebar Bar

Fungsi `bar()` menerima argumen kata kunci `width` untuk mengatur lebar batang.

```
plt.bar(x, y, width = 0.1)
```

Nilai lebar standar adalah 0,8

Catatan: Untuk batang horizontal, gunakan tinggi daripada lebar

```
plt.barh(x, y, height = 0.1)
```

Matplotlib Histograms

Histogram adalah grafik yang menunjukkan distribusi frekuensi.

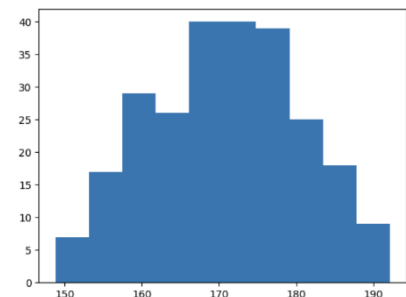
Grafik ini menampilkan jumlah pengamatan dalam setiap interval tertentu.

Example:

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

x = np.random.normal(170, 10,
250)

plt.hist(x)
plt.show()
```



Matplotlib Pie Charts

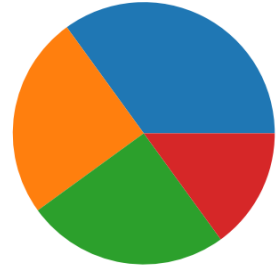
Dengan Pyplot, Anda dapat menggunakan fungsi `pie()` untuk menggambar diagram lingkaran.

Contoh :

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

y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```

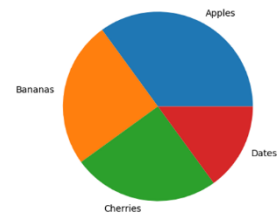


Seperti yang Anda lihat, grafik lingkaran (pie chart) menggambar satu bagian (disebut irisan atau segmen) untuk setiap nilai dalam deretan data (dalam kasus ini [35, 25, 25, 15]). Secara default, penggambaran irisan pertama dimulai dari sumbu x dan bergerak berlawanan arah jarum jam.

- Label

Tambahkan label pada grafik lingkaran dengan parameter `label`. Parameter label harus berupa array dengan satu label untuk setiap irisan.

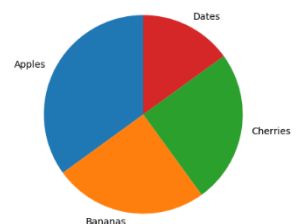
```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
plt.pie(y, labels = mylabels)
plt.show()
```



- Sudut Awal

Sudut awal standar berada pada sumbu x, tetapi Anda dapat mengubah sudut awal dengan menentukan parameter `startangle`. Sudut awal default adalah 0:

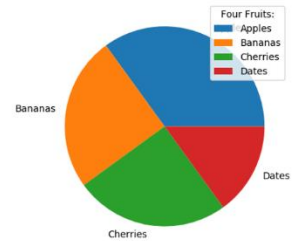
```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
plt.pie(y, labels = mylabels,
startangle = 90)
plt.show()
```



- Legend

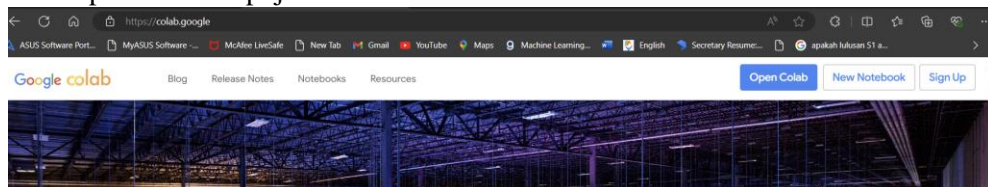
Untuk menambahkan daftar penjelasan untuk setiap bagian grafik, gunakan fungsi `legend()`. Untuk menambahkan judul pada legend, tambahkan parameter `title` pada fungsi `legend()`.

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas",
"Cherries", "Dates"]
plt.pie(y, labels = mylabels)
plt.legend(title = "Four Fruits:")
plt.show()
```

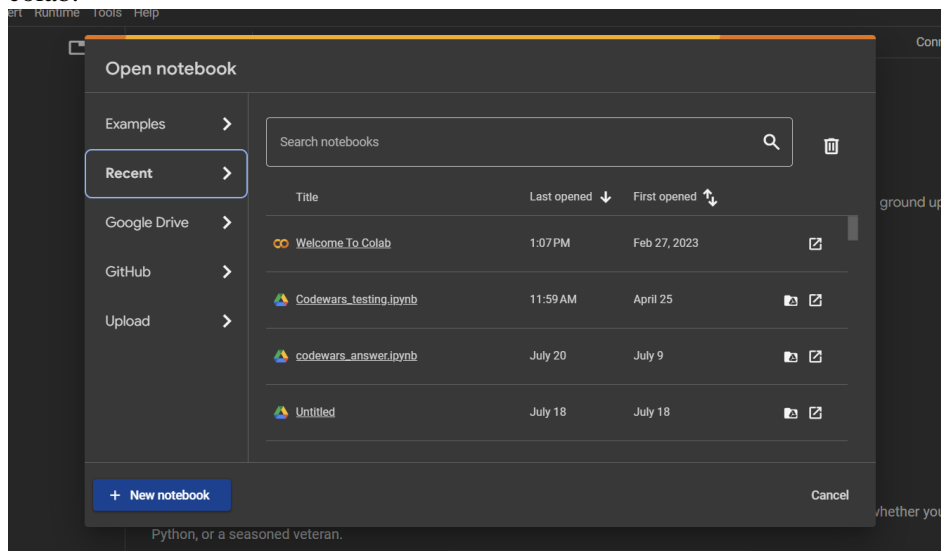


1.3 Langkah Persiapan

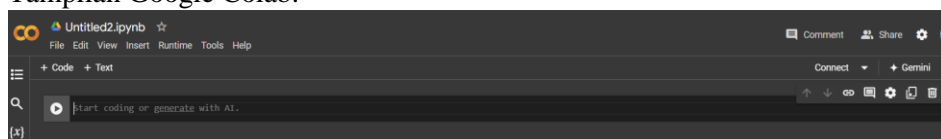
1. Unduh file testing dengan format Python (.py) pada link Google drive berikut ini, <https://drive.google.com/drive/folders/1vELIyfN3MQDoTSnug9ChqyUuz61Q0X35?usp=sharing>
2. Membuka Google Colab
 - a. Buka Google Colaboratory dengan link berikut <https://colab.research.google.com/> .
 - b. Klik Open Colab di pojok kanan atas



- c. Anda bisa login menggunakan akun Google.
- d. Klik New Notebook pada pojok kiri bawah, untuk membuka halaman baru google colab.



- e. Tampilan Google Colab.

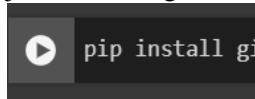


3. Menjalankan install codewars_test pada cell google colab
codewars_test adalah framework yang digunakan untuk menjalankan validasi kode jawaban.

- a. Jalankan command berikut:

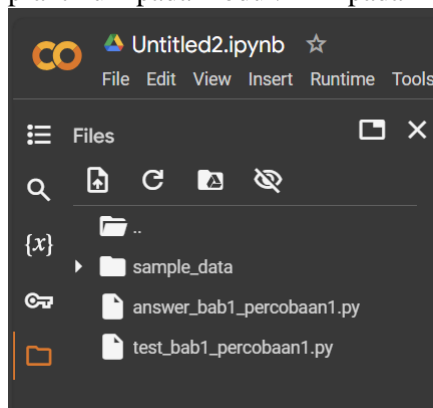
```
pip install git+https://github.com/codewars/python-test-  
framework.git#egg=codewars_test
```

- b. jalankan dengan menekan tombol play



4. Menjalankan test Validasi untuk hasil kode jawaban

- a. Upload file testing file untuk validasi jawaban dan file berisi kode hasil dari jawaban praktikum pada modul. Klik pada ikon files.



- b. Jalankan file testing pada cell google colab untuk memvalidasi kode hasil jawaban praktikum.
c. Ketikkan command berikut, lalu jalankan cell:

```
!python /content/test_bab1_percobaan1.py
```

Pada nama file testing test_bab1_percobaan 1.py, sesuaikan dengan nama file test yang akan dijalankan sesuai dengan modul yang telah dikerjakan.

- d. Berikut adalah output hasil dari validasi
PASSED menandakan case yang dikerjakan berhasil, sedangkan FAILED menandakan case yang dikerjakan gagal dan perlu perbaikan.

```
<DESCRIBE::>BAB 1 | Percobaan 1  
=====
```

```
<IT::>1. Test Memuat Data  
<PASSED::>Test Passed  
<PASSED::>Test Passed  
<COMPLETEDIN::>50.77  
<IT::>2. Test Print Nilai Fungsi data_load()  
=====
```

```
<FAILED::>====> Error :Tidak Menampilkan nilai fungsi data_load() menggunakan print(): False should equal True  
<COMPLETEDIN::>0.03  
<COMPLETEDIN::>50.88
```

1.4 Contoh Studi Kasus

Membuat Label, Legenda, dan Sudut Awal

Skenario: Sebuah perusahaan menganalisis data penjualan untuk berbagai kategori produk. Mereka ingin membuat diagram lingkaran untuk memvisualisasikan distribusi penjualan di antara kategori-kategori tersebut.

Objektif:

1. Memberi label persentase pada setiap irisan diagram lingkaran dengan kategori produk yang sesuai.
2. Menambahkan legenda yang dengan jelas mengidentifikasi setiap kategori.
3. Memutar diagram lingkaran untuk memulai pada sudut tertentu, sehingga menekankan kategori yang paling signifikan.

Langkah-langkah:

1. Import Pustaka:

1.1 import pandas as pd: Mengimpor pustaka Pandas untuk manipulasi data.

1.2 import matplotlib.pyplot as plt: Mengimpor pustaka Matplotlib untuk visualisasi data.

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

2. Fungsi Pemuatan Data:

2.1 def load_data(): Mendefinisikan fungsi load_data untuk memuat data dari file CSV.

2.2 Membaca file CSV dengan URL yang diberikan dan menyimpannya dalam DataFrame df.

https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/product_category.csv

2.3 Mengembalikan DataFrame df.

```
def load_data():
    url =
    "https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main
    /product_category.csv "
    df = pd.read_csv(url)
    return df
```

3. Fungsi Penentuan Ukuran Gambar:

3.1 definisikan variabel **fig_size**

3.2 Membuat objek gambar dengan ukuran **8x6** inci menggunakan `plt.subplots()`.

```
fig_size = plt.subplots(figsize=(8, 6))
```

4. Membuat Pie Chart:

4.2 Menggunakan `plt.pie()` untuk membuat pie chart dengan:

- Data penjualan (`load_data()['Sales']`) sebagai ukuran potongan pie.
- Label kategori produk (`load_data()['Product Category']`) untuk setiap potongan.
- Persentase pada setiap potongan (`autopct='%1.1f%%'`).
- Sudut awal (`startangle=90`) untuk memulai pie chart dari posisi jam 12.

```
plt.pie(load_data()['Sales'], labels=load_data()['Product Category'], autopct='%1.1f%%', startangle=90)
```

4.3 Menambahkan legenda menggunakan `plt.legend(title='Product Category')`.

```
plt.legend(title='Product Category')
```

4.4 Memberikan judul pada grafik menggunakan `plt.title('Sales Distribution by Product Category')`.

```
plt.title('Sales Distribution by Product Category')
```

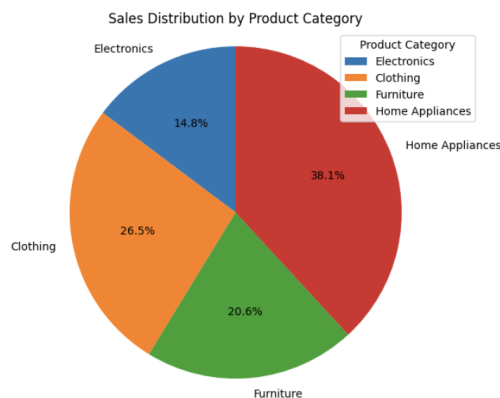
4.5 Menyesuaikan tampilan grafik menggunakan `plt.axis('equal')`.

```
plt.axis('equal') # Memastikan diagram lingkaran berbentuk bulat
```

4.6 Menampilkan grafik menggunakan `plt.show()`.

```
plt.show()
```

Output:



Tampilan Keseluruhan Kode

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

def load_data():
    url =
    "https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main
    /product_category.csv "
    df = pd.read_csv(url)
    return df
def create_pie_chart():

    fig_size = plt.subplots(figsize=(8, 6))

    plt.pie(load_data()['Sales'], labels=load_data()['Product
    Category'], autopct='%1.1f%%', startangle=90)
    plt.legend(title="Product Category")
    plt.title('Sales Distribution by Product Category')
    plt.axis('equal')
    plt.show()
```

1.4 Praktikum

Skenario:

Departemen perencanaan kota ingin menganalisis distribusi usia penduduknya untuk memahami komposisi demografi dan merencanakan layanan masa depan.

Tujuan:

- Persiapan data: Membuat dataset sampel dengan kategori dan nilai yang sesuai.
- Visualisasi data: Membuat diagram lingkaran untuk merepresentasikan data secara visual.
- Kustomisasi grafik: Menambahkan label, legenda, dan judul untuk meningkatkan keterbacaan dan informativitas grafik.

Langkah – langkah:

1. Import Pustaka:

1.1 *import pandas as pd*: Mengimpor pustaka Pandas untuk manipulasi data.

1.2 *import matplotlib.pyplot as plt*: Mengimpor pustaka Matplotlib untuk visualisasi data.

2. Fungsi Pemuatan Data:

2.1 **def load_data()** : definisikan fungsi load_data untuk memuat data dari file CSV.

2.2 buat variabel **df** untuk menyimpan dataframe berupa file csv yang di dapat dari link berikut:

https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/age_group.csv

2.3 Mengembalikan DataFrame **df** menggunakan return.

3. Membuat Pie Chart:

3.1 Buatlah fungsi bernama **create_pie_chart()**

3.2 Definisikan variabel **df** yang akan menyimpan *data frame* dari load_data()

3.3 Definisikan variabel **get_age_col** untuk mengambil kolom 'Age Group' dari *data frame*. **df['Age Group']**

3.4 Definisikan variabel **get_population_col** untuk mengambil kolom 'Population' dari *data frame*. **df['Population']**

3.5 Definisikan variabel **fig_size** untuk mengatur ukuran gambar.

3.6 Buat objek gambar dengan ukuran **10x8** inci menggunakan **plt.subplots()**.

3.7 Menggunakan **plt.pie()** untuk membuat pie chart dengan data populasi, label kelompok umur, persentase, dan sudut awal.

- Ambil data kolom populasi, gunakan variabel **get_population_col** yang telah didefinisikan sebelumnya sebagai ukuran potongan pie.
- Ambil label kategori grup umur, gunakan variabel **get_age_col** yang telah didefinisikan sebelumnya untuk setiap potongan.
- Persentase pada setiap potongan (**autopct='%1.1f%%'**).
- Sudut awal (**startangle=90**) untuk memulai pie chart dari posisi jam 12.

3.8 Menambahkan legenda menggunakan **plt.legend()** beri nama "Rentang usia".

3.9 Memberikan judul pada grafik menggunakan **plt.title()** beri judul "Distribusi Usia Penduduk pada Kota A".

3.10 Menyesuaikan tampilan grafik menggunakan **plt.axis('equal')**.

3.11 Menampilkan grafik menggunakan **plt.show()**.

4. Panggil fungsi **create_pie_chart()** untuk menampilkan pie chart.

5. Submit

Simpan file dengan nama **answer_bab4_percobaan1.py** pastikan file disimpan dalam format Python file (.py).