

DATA ANALYTICS

BAB 3 : DATA MANIPULATIONS

Praktikum 3

1.1 Tujuan

Mahasiswa mengenal materi Manipulasi Data menggunakan bahasa pemrograman Python antara lain mahasiswa akan mempelajari tentang penggunaan library Python seperti NumPy dan Pandas.

1.2 Ulasan Materi

A. Introduction to Pandas

1. Apa itu Pandas?

Pandas adalah library Python yang kuat untuk memanipulasi, menganalisis, dan membersihkan data.

2. Mengapa Pandas?

- Dibuat untuk Big Data
- Mudah Dipelajari
- Fungsionalitas yang Kuat

3. Memulai dengan Pandas

Mengimpor Pandas:

```
import pandas as pd
```

Membuat DataFrame:

```
data = {  
    "mobil": ["BMW", "Volvo", "Ford"],  
    "kelulusan": [3, 7, 2]  
}  
  
myvar = pd.DataFrame(data)
```

Memeriksa Versi Pandas:

```
print(pd.__version__)
```

4. Seri Pandas

Seri adalah seperti kolom dalam tabel.

Membuat Seri:

```
a = [1, 7, 2]

myvar = pd.Series(a)
```

Memberi Label pada Seri:

```
a = [1, 7, 2]

myvar = pd.Series(a, index=["x", "y", "z"])
```

Memilih Item dalam Seri:

```
# Akses item pertama
print(myvar[0])

# Akses item dengan label
print(myvar["y"])
```

Membuat Seri dari Dictionary:

```
calories = {"hari1": 420, "hari2": 380, "hari3": 390}
myvar = pd.Series(calories)
```

5. DataFrame Pandas

DataFrame adalah seperti tabel dengan baris dan kolom.

Membuat DataFrame:

```
import pandas as pd

data = {
    "kalori": [420, 380, 390],
    "durasi": [50, 40, 45]
}

df = pd.DataFrame(data)
```

Menemukan Baris:

```
# Akses baris 0
print(df.loc[0])

# Akses baris 0 dan 1
print(df.loc[[0, 1]])
```

Menemukan Baris dengan Indeks Bernama:

```
# Akses baris "hari2"  
print(df.loc["hari2"])
```

Memuat File ke dalam DataFrame:

```
df = pd.read_csv("data.csv")
```

6. Membaca File CSV

```
df = pd.read_csv('data.csv')  
  
print(df.to_string())
```

Tips: Gunakan `to_string()` untuk mencetak seluruh DataFrame.

7. Membaca File JSON

```
df = pd.read_json('data.json')  
  
print(df.to_string())
```

8. Menganalisis DataFrame dengan Pandas

Menampilkan Data:

```
df = pd.read_csv('data.csv')  
  
print(df.head(10)) #menampilkan 10 baris pertama dataset
```

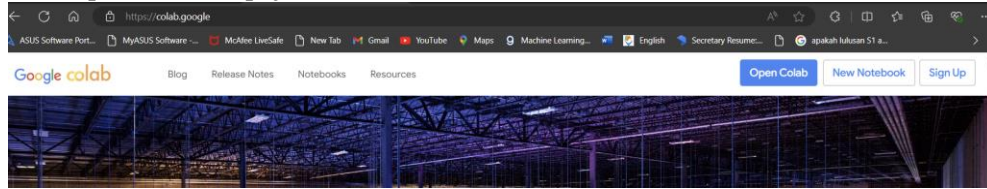
Informasi Tentang Data:

```
df = pd.read_csv('data.csv')  
  
print(df.info())
```

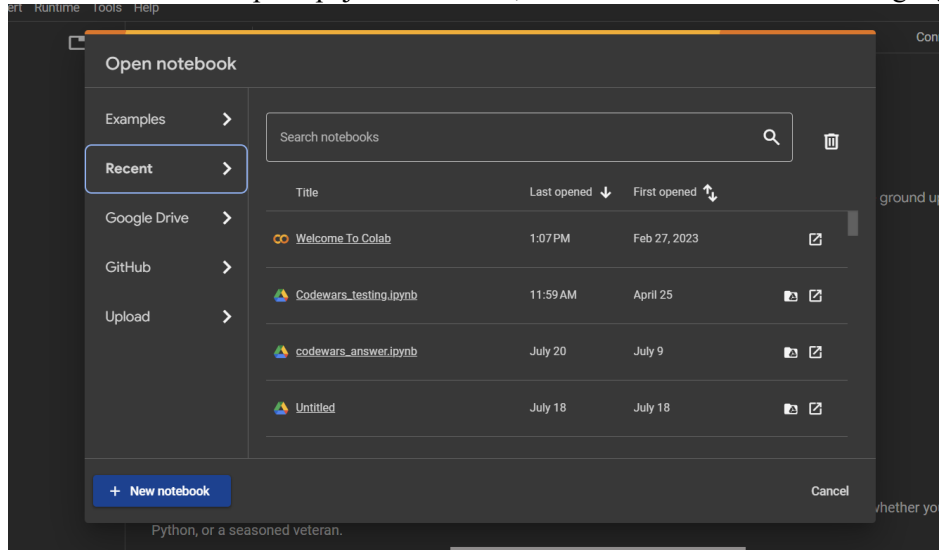
1.3 Langkah Persiapan

1. Membuka Google Colab

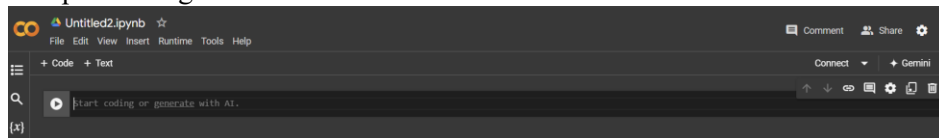
- Buka Google Colaboratory dengan link berikut <https://colab.research.google.com/>.
- Klik Open Colab di pojok kanan atas



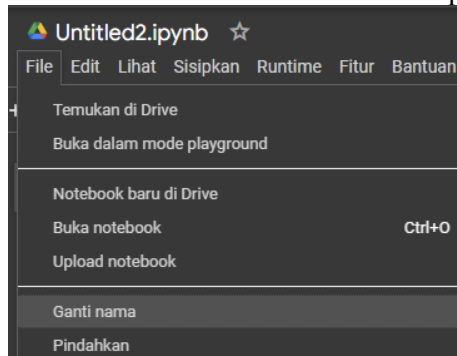
- Anda bisa login menggunakan akun Google.
- Klik New Notebook pada pojok kiri bawah, untuk membuka halaman baru google colab.



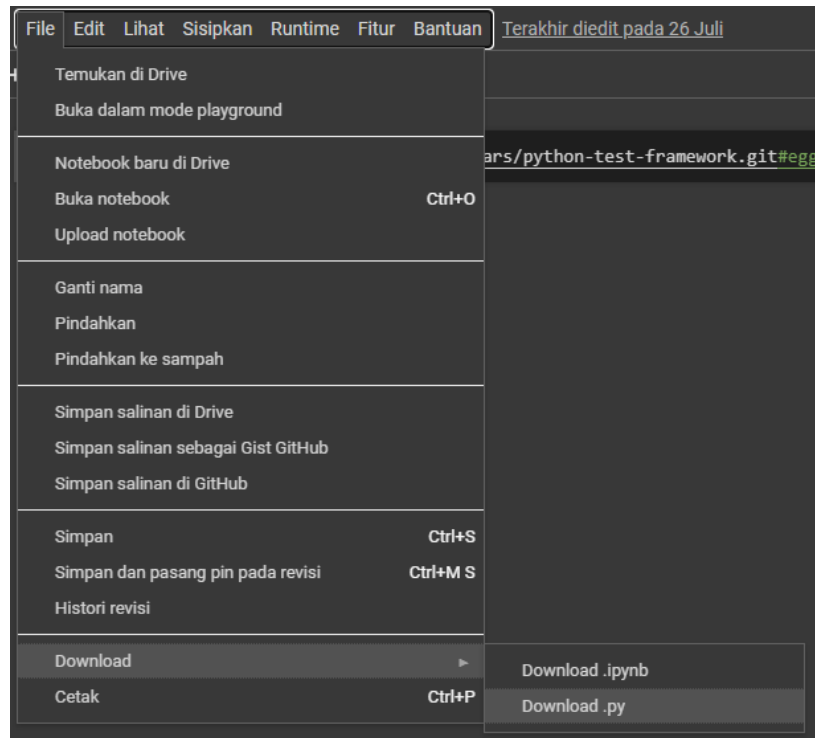
e. Tampilan Google Colab.



f. Ganti nama file sesuai arahan format pada praktikum



g. Setelah selesai mengerjakan praktikum, download file dengan format (.py)



1.4 Contoh Studi Kasus

Produk dengan Harga Rata-Rata Tertinggi

Pada contoh ini kita akan belajar menganalisa data dengan menemukan produk dengan garga rata-rata tertinggi. Berikut ini adalah langkah-langkahnya :

1. Buat fungsi **load_data()** untuk membaca file CSV. Simpan pada variable **data_toko**
Akses dataset pada url berikut:
https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/master/data_toko.csv

```
import pandas as pd

url =
"https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/master/data_toko.csv" # Replace with your actual URL
```

Mengembalikan nilai **data_toko** menggunakan **return**

```
# Read the CSV directly from the URL
def data_load():
    data_toko = pd.read_csv(url)
    return data_toko
```

2. Memeriksa Data:

- Buatlah fungsi bernama **head_rows()** untuk menampilkan 10 baris pertama dari DataFrame.

```
# Tampilkan 10 baris pertama DataFrame
def head_rows():
    return data_load().head(10)
```

- Tampilkan informasi tentang DataFrame, termasuk jumlah baris, kolom, dan tipe data. Menggunakan **info()**

```
data_load().info()
```

3. Analisis Data:

- Hitung pendapatan total dari penjualan semua produk. Buat fungsi **find_highest_average_price()**

```
def find_highest_average_price():
    df = data_load()

    # Kelompokkan data berdasarkan lini produk
    produk_terkelompokan = df.groupby('Product line')

    # Hitung harga rata-rata untuk setiap kelompok produk
    produk_terkelompokan = produk_terkelompokan[['Unit price']].mean()

    # Temukan produk dengan harga rata-rata tertinggi
    produk_harga_rata_tinggi = produk_terkelompokan.idxmax()
    harga_rata_tinggi = produk_terkelompokan.max()

    # Kembalikan produk dengan harga rata-rata tertinggi dan harga rata-ratanya
    return produk_harga_rata_tinggi, harga_rata_tinggi
```

4. Print produk dengan harga rata-rata tertinggi **print()**

```
def print_highest_average_price():
    produk_harga_rata_tinggi, harga_rata_tinggi = find_highest_average_price()
    print(f"Produk dengan harga rata-rata tertinggi: {produk_harga_rata_tinggi} dengan harga rata-rata {harga_rata_tinggi}")
```

```
print_highest_average_price()
```

Tampilan Keseluruhan Kode

```
import pandas as pd
```

```

url =
"https://raw.githubusercontent.com/noora20FH/skripsi_noor
a2023/master/data_toko.csv" # Replace with your actual
URL

def data_load():
    data_toko = pd.read_csv(url)
    return data_toko

def head_rows():
    return data_load().head(10)

def find_highest_average_price():
    df = data_load()

    # Kelompokkan data berdasarkan lini produk
    produk_terkelompokan = df.groupby('Product line')

    # Hitung harga rata-rata untuk setiap kelompok produk
    produk_terkelompokan = produk_terkelompokan[['Unit
price']].mean()

    # Temukan produk dengan harga rata-rata tertinggi
    produk_harga_rata_tinggi =
produk_terkelompokan.idxmax()
    harga_rata_tinggi = produk_terkelompokan.max()

    # Kembalikan produk dengan harga rata-rata tertinggi
    dan harga rata-ratanya
    return produk_harga_rata_tinggi, harga_rata_tinggi

def print_highest_average_price():
    produk_harga_rata_tinggi, harga_rata_tinggi =
find_highest_average_price()
    print(f"Produk dengan harga rata-rata tertinggi:
{produk_harga_rata_tinggi} dengan harga rata-rata
{harga_rata_tinggi}")

# Gunakan fungsi untuk menemukan produk dengan harga
rata-rata tertinggi
print_highest_average_price()

```

1.5 Praktikum

Menganalisis dataset film untuk menemukan 10 film Terbaik

Skenario:

Bayangkan Anda bekerja di platform streaming film dan ingin mempromosikan 10 film terbaik kepada pengguna berdasarkan skor kritikus. Anda memiliki akses ke dataset film yang berisi berbagai informasi, termasuk skor kritikus.

Objektif:

1. Menganalisis dataset film untuk menemukan 10 film dengan skor kritikus tertinggi.
2. Menyajikan daftar 10 film terbaik.

Langkah – Langkah :

1. Import library yang dibutuhkan
2. Buat fungsi **load_data()** untuk membaca file CSV. Simpan pada variable **data_toko**
Akses dataset pada url berikut:
https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/clean_movie_data.csv

Mengembalikan nilai **data_toko** menggunakan **return**

3. Buatlah fungsi bernama **critic_scores()**
4. Buatlah variable bernama **critic_scores** di dalam fungsi **critic_scores()** untuk merubah tipe data kolom **critic_scores** menjadi Numpy array menggunakan **to.numpy()**

`nama_variabel = nama_fungsi()["nama_kolom"].to_numpy()`

5. Return variabel **critic_scores**
6. Buatlah fungsi bernama **sorted_indices()**
7. Buatlah variable bernama **sorted_indices** di dalam fungsi **sorted_indices()** untuk mengurutkan index dari tinggi ke rendah.
 - Panggil variabel **critic_scores** dan gunakan **argsort()** beserta python slicing **[::-1]** untuk mengurutkan terbalik (tinggi ke rendah).
8. Return variabel **sorted_indices**
9. Buatlah fungsi **top_10_movies()**
10. Buatlah variable bernama **top_10_movies** di dalam fungsi **top_10_movies()**
 - Panggil fungsi **data_load()** untuk menggunakan DataFrame yang berisi data movie.
 - Gunakan metode **iloc()** pada **data_load()** untuk mengakses elemen dalam DataFrame
 - Potong array pada **sorted_indices()** menggunakan slicing python **[:]** untuk

skor 10 teratas

```
nama_variabel = nama_fungsi().iloc[nama_variabel[:10] ]
```

11. Print total 10 movie terbaik menggunakan metode **print()**

12. Submit

Beri file dengan nama **answer_bab3_percobaan3.py** pastikan file dalam format **Python (.py)**