

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
```

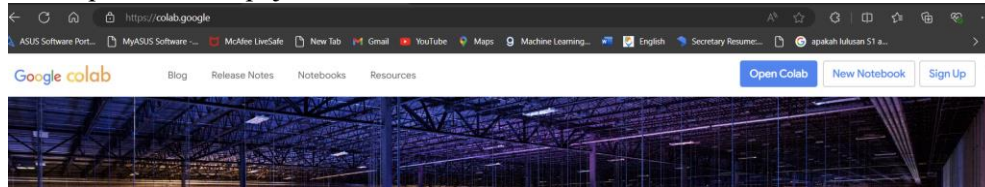
### Informasi Tentang Data:

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

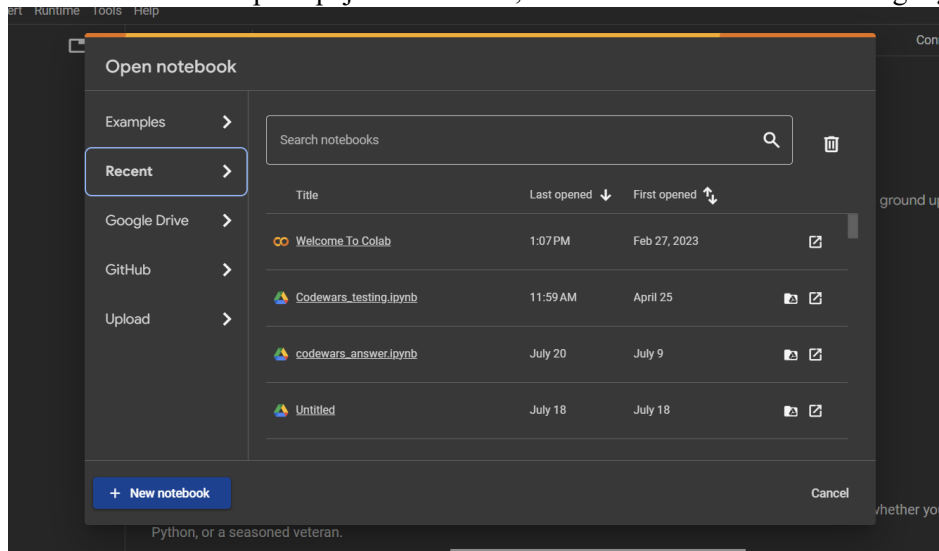
print(df.info())
```

### 1.3 Langkah Persiapan

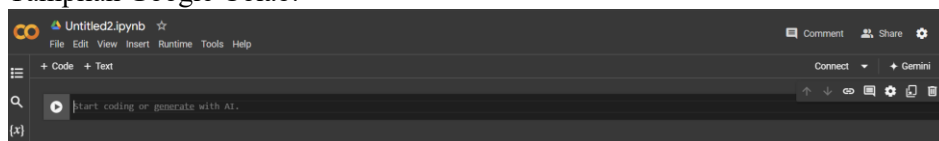
1. Unduh file testing dengan format Python (.py) pada link Google drive berikut ini,  
<https://drive.google.com/drive/folders/1vELIyfN3MQDoTSnug9ChqyUuz61Q0X35?usp=sharing>
2. Membuka Google Colab
  - a. Buka Google Colaboratory dengan link berikut <https://colab.research.google.com/> .
  - b. Klik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



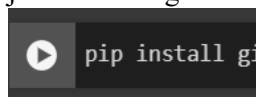
- e. Tampilan Google Colab.



3. Menjalankan install codewars\_test pada cell google colab  
codewars\_test adalah framework yang digunakan untuk menjalankan validasi kode jawaban.
  - a. Jalankan command berikut:

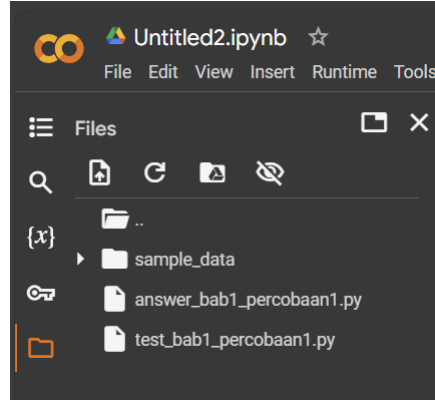
```
pip install git+https://github.com/codewars/python-test-framework.git#egg=codewars_test
```

- 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

### 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. Import library yang dibutuhkan

```
import pandas as pd
```

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/master/data\\_toko.csv](https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/master/data_toko.csv)

```
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
```

3. **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()
```

4. **Analisis Data:**

- Hitung pendapatan total dari penjualan semua produk. Buat fungsi **find\_highest\_average\_price()**

```
# Definisikan fungsi untuk menemukan produk dengan harga rata-rata tertinggi  
  
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
# Gunakan fungsi untuk menemukan produk dengan harga rata-rata tertinggi
produk_harga_rata_tinggi, harga_rata_tinggi =
find_highest_average_price()
```

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

```
# Cetak produk dengan harga rata-rata tertinggi
print(f"Produk dengan harga rata-rata tertinggi:
{produk_harga_rata_tinggi} dengan harga rata-rata
{harga_rata_tinggi}")
```

## 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](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()**