DATA ANALYTICS

BAB 2 : DATA CLEANING

1.1 Tujuan

Mahasiswa mengenal materi pembersihan data (data cleaning). Pembersihan data adalah proses persiapan data yang melibatkan identifikasi, memahami, dan menangani kesalahan (errors), missing values, dan inkonsistensi dalam kumpulan data.

1.2 Ulasan Materi

Pembersihan Data

Pembersihan data berarti memperbaiki data yang buruk dalam kumpulan data Anda. Data buruk bisa berupa:

- Sel kosong
- Data dalam format yang salah
- Data salah
- Duplikat

Dalam tutorial ini Anda akan belajar bagaimana menangani semuanya.

Kumpulan Data Kami

Dalam bab selanjutnya kita akan menggunakan kumpulan data ini:

	Durasi	Tanggal	Denyut Nadi	Denyut Nadi Maksimum	Kalori
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300.0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253.3
8	30	'2020/12/09'	109	133	195.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329.3
11	60	'2020/12/12'	100	120	250.7
12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379.3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300.0
18	45	'2020/12/18'	90	112	NaN
19	60	'2020/12/19'	103	123	323.0

20	45	'2020/12/20'	97	125	243.0
21	60	'2020/12/21'	108	131	364.2

1. **Import Library yang dibutuhkan**

```
import pandas as pd
```

2. **Pembersihan Data Sel Kosong dengan Pandas**

Hapus Baris:

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

print(new_df.to_string())
```

Ganti Nilai Kosong

```
df = pd.read_csv('data.csv')
df.fillna(130, inplace=True)

print(df.to_string())
```

Ganti Hanya untuk Kolom Tertentu:

```
df = pd.read_csv('data.csv')
df["Kalori"].fillna(130, inplace=True)

print(df.to_string())
```

Ganti Menggunakan Mean, Median, atau Mode:

```
df = pd.read_csv('data.csv')
x = df["Kalori"].mean()
df["Kalori"].fillna(x, inplace=True)

print(df.to_string())
```

3. **Pembersihan Data - Format Data yang Salah dengan Pandas**

Format Data yang Salah:

- Sel dengan format data yang salah dapat mempersulit atau bahkan tidak memungkinkan analisis data.
- Dua opsi untuk memperbaikinya: hapus baris atau konversi semua sel di kolom ke format yang sama.

Contoh:

```
# Membaca data dari CSV
df = pd.read_csv('data.csv')

# Menemukan sel dengan format tanggal yang salah
tanggal_salah = df[df['Tanggal'].str.contains('[^0-9-/]+')]['Tanggal'].unique()

print("Tanggal yang salah:", tanggal_salah)

# Menghapus baris dengan tanggal yang salah
df_baru = df.drop(df[df['Tanggal'].str.contains('[^0-9-/]+')].index)

# Menampilkan data yang dihapus
print(df_baru.to_string())
```

4. **Pandas - Memperbaiki Data yang Salah**

   Data yang Salah:

   - Data yang salah tidak selalu berarti "sel kosong" atau "format yang salah".
   - Data bisa saja salah secara nilai, seperti "199" alih-alih "1.99".

   Contoh:

```
# Membaca data dari CSV
df = pd.read_csv('data.csv')

# Menemukan durasi yang tidak masuk akal
durasi_salah = df[df['Durasi'] > 300]

print("Durasi yang salah:", durasi_salah)

# Mengubah durasi yang salah menjadi nilai yang wajar
df['Durasi'].replace(to_replace=durasi_salah['Durasi'].max(),
                     method='fillna', inplace=True)

# Menampilkan data yang diubah
print(df.to_string())
```

5. **Pandas - Menghilangkan Duplikat**

   Mengenal Duplikat:

   - Baris duplikat adalah baris yang tercatat lebih dari sekali.

   Contoh:

```
# Membaca data dari CSV
```

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

# Menemukan baris duplikat
duplikat = df.duplicated()

print("Baris duplikat:", df[duplikat])

# Menghapus baris duplikat
df_baru = df.drop_duplicates()

# Menampilkan data yang dihapus duplikatnya
print(df_baru.to_string())
```

6. **Pandas - Korelasi Data**

Menemukan Hubungan Antar Kolom:

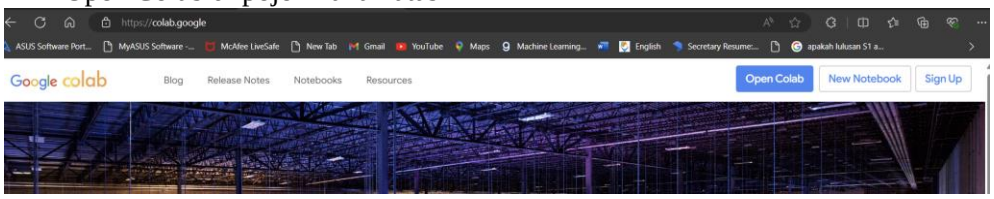
- Metode `corr()` menghitung hubungan antara setiap kolom dalam kumpulan data.

Contoh:

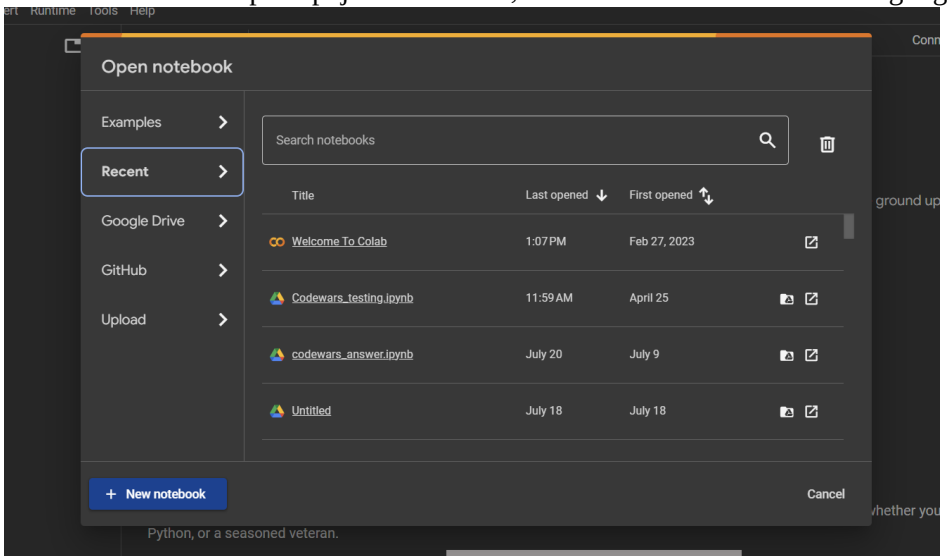
```
# Membaca data dari CSV
df = pd.read_csv('data.csv')

# Menghitung korelasi antar kolom
korelasi = df.corr()

# Menampilkan matriks korelasi
print(korelasi.to_string())
```

1.3 Langkah Persiapan

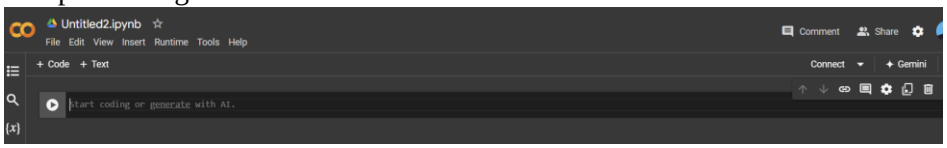
1. Membuka Google Colab
    a. Buka Google Colaboratory dengan link berikut https://colab.research.google.com/ .
    b. Klik Open Colab di pojok kanan atas



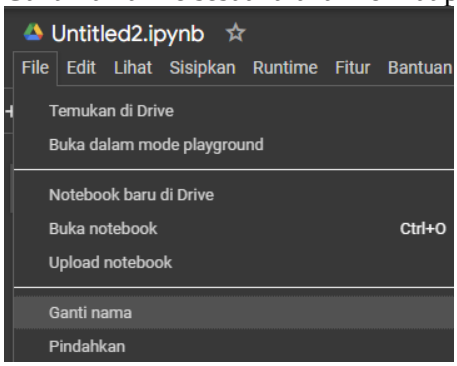
    c. Anda bisa login menggunakan akun Google.
    d. Klik New Notebook pada pojok kiri bawah, untuk membuka halaman baru google colab.



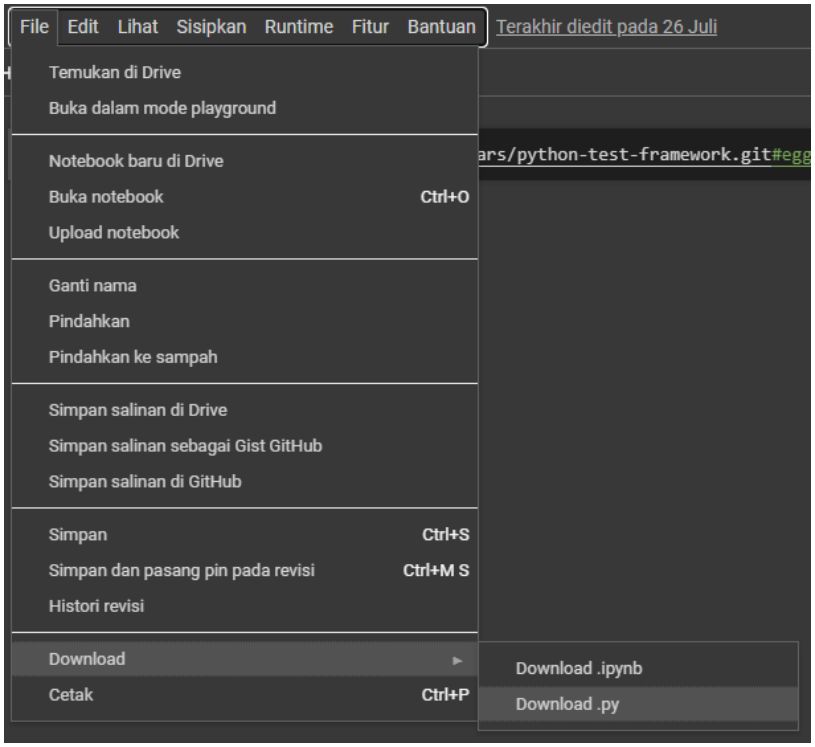
    e. Tampilan Google Colab.



    f. Ganti nama file sesuai arahan format pada praktikum



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


File
Edit
Lihat
Sisipkan
Runtime
Fitur
Bantuan
Terakhir diedit pada 26 Juli
Temukan di Drive
Buka dalam mode playground
Notebook baru di Drive
Buka notebook Ctrl+O
Upload notebook
Ganti nama
Pindahkan
Pindahkan ke sampah
Simpan salinan di Drive
Simpan salinan sebagai Gist GitHub
Simpan salinan di GitHub
Simpan Ctrl+S
Simpan dan pasang pin pada revisi Ctrl+M S
Histori revisi
Download
Cetak Ctrl+P
Download .ipynb
Download .py
ars/python-test-framework.git#egg

1.4 Contoh Studi Kasus

Preprocessing Data untuk Analisis Statistik

Langkah - Langkah

1. Memuat Pustaka Pandas `import pandas as pd`.

```
import pandas as pd
```

2. Mendefinisikan Fungsi `load_data()`:`def load_data()`. Perintah ini mendefinisikan sebuah fungsi bernama `load_data()`. Fungsi ini akan digunakan untuk memuat data dari file CSV ke dalam DataFrame pandas.
3. Menetapkan URL Data, `url = "https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/data.csv"`. Perintah ini menetapkan URL file CSV yang berisi data yang ingin dimuat. Pastikan untuk mengganti URL ini dengan URL file CSV Anda yang sebenarnya.
4. Membaca Data CSV. `df = pd.read_csv(url)`. Perintah ini menggunakan fungsi `pd.read_csv()` dari pustaka pandas untuk membaca data dari file CSV yang ditentukan oleh variabel `url`. Hasilnya disimpan dalam objek DataFrame dengan nama `df`.

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

5. Menampilkan Jumlah Nilai Hilang: `print(load_data().isnull().sum())`. Perintah ini memanggil fungsi `load_data()` untuk memuat data dan kemudian menghitung jumlah nilai yang hilang di setiap kolom DataFrame. Hasilnya dicetak ke konsol, menunjukkan berapa banyak nilai yang hilang untuk setiap variabel dalam dataset.

```
print(load_data().isnull().sum())
```

Output:

```
Duration    0
Pulse       0
Maxpulse    0
Calories    5
dtype: int64
```

6. Memuat Data ke dalam DataFrame Baru: `new_df = load_data()`. Perintah ini membuat DataFrame baru bernama `new_df` dan memuat data dari fungsi `load_data()`.

7. Mengisi Nilai Hilang pada Kolom Kalori:
`new_df['Calories'].fillna(new_df['Calories'].mean())`. menggunakan metode `fillna()` pada kolom `Calories` dari DataFrame `new_df`. Metode ini mengisi nilai yang hilang dengan nilai rata-rata dari kolom `Calories`.

```
def updated_data():
  updated_df =
load_data().fillna(load_data()['Calories'].mean())
  return updated_df
```

8. Menampilkan DataFrame Baru `print(new_df)`. Perintah ini mencetak DataFrame `new_df` ke konsol.
9. Menampilkan Jumlah Nilai Hilang Setelah Pengisian Perhatikan bahwa pada kolom `Calories` sekarang tidak memiliki null

```
print(updated_data().isnull().sum())
```

```
Duration    0
Pulse       0
Maxpulse    0
Calories    0
dtype: int64
```

Tampilan keseluruhan kode

```
import pandas as pd

def load_data():
  url =
"https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/da
ta.csv"
  df = pd.read_csv(url)
  return df
print(load_data().isnull().sum())

def updated_data():
  updated_df = load_data().fillna(load_data()['Calories'].mean())
  return updated_df

print(updated_data().isnull().sum())
```

1.6 Praktikum

Mengisi Nilai Hilang pada Kolom Kalori

Skenario:

Bayangkan Anda adalah seorang ahli gizi yang ingin menganalisis data makanan untuk membantu klien Anda mencapai tujuan kesehatan mereka. Anda memiliki kumpulan data CSV yang berisi informasi tentang berbagai makanan, termasuk nama makanan, kalori, lemak, protein, dan karbohidrat. Namun, Anda menemukan bahwa beberapa nilai kalori hilang dalam dataset. Hal ini dapat membuat analisis data menjadi tidak akurat dan tidak dapat diandalkan.

Objektif:

- Isi nilai kalori hilang untuk tingkatkan kualitas data dan analisis.
- Bantu klien buat pilihan makanan dan rekomendasi diet lebih baik.

Langkah – Langkah:

1. Import library untuk manipulasi data.
2. Define fungsi `load_data()`:
    a. Menetapkan URL file CSV ke variabel `url`.

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

    b. Membaca data CSV menggunakan `pd.read_csv(url)`, menyimpannya dalam DataFrame `df`.
    c. Mengembalikan DataFrame `df`.

3. Mengisi Nilai yang hilang dengan nilai rata-rata:
    a. Define fungsi `updated_data()`:
    b. Buat variabel `updated_data`
    c. Memuat data dengan `load_data()`.
    d. Mengisi nilai yang hilang pada kolom 'Calories' dengan rata-rata nilai 'Calories' dalam DataFrame menggunakan .`fillna(load_data['Calories'].mean())`
    e. Mengembalikan nilai `updated_data`.

4. Print string berisikan “Missing value: ”
5. Print menggunakan fungsi **print(),** total nilai yang hilang per kolom dengan `updated_data().isnull().sum()`.
6. Submit

   Simpan nama file dengan **answer_bab2_percobaan3.py** pastikan file tersimpan dalam format Python file **(.py)**