DATA ANALYTICS

BAB 2 : DATA CLEANING

Praktikum 2

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:

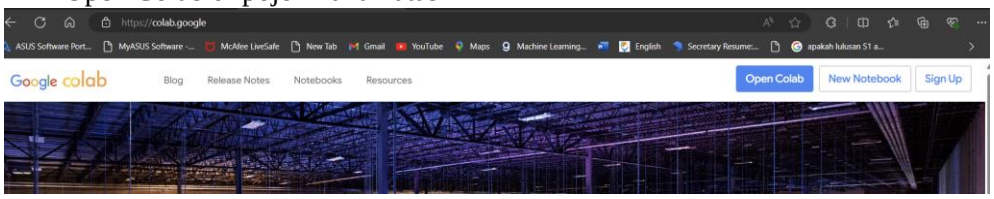
- Metode corr() menghitung hubungan antara setiap kolom dalam kumpulan data.

Contoh:

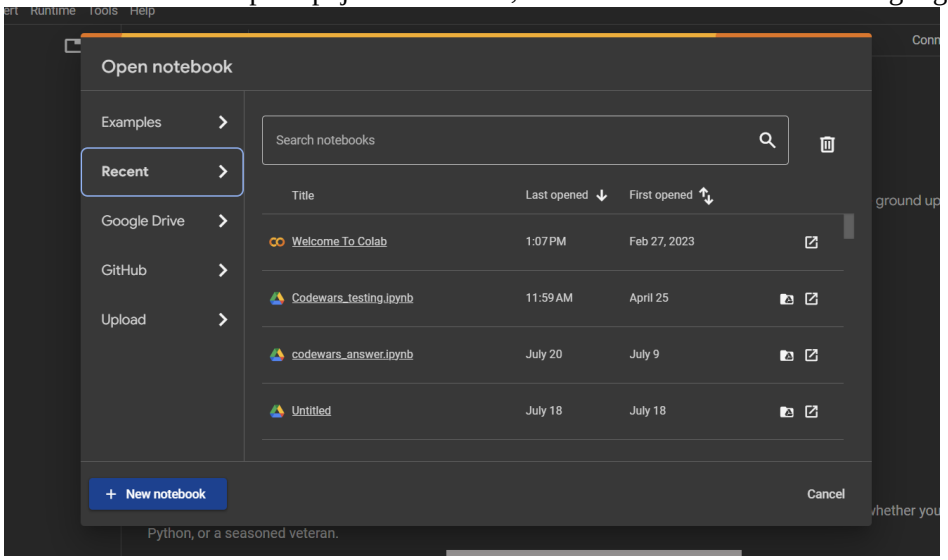
```
# Membaca data dari CSV
df = pd.read_csv('data.csv')

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

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

1.3 Langkah Persiapan

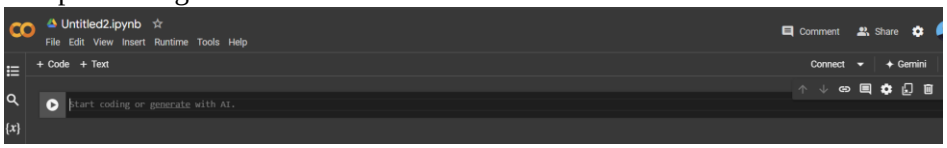
1. Membuka Google Colab
    a. Buka Google Colaboratory dengan link berikut https://colab.research.google.com/ .
    b. Klik Open Colab di pojok kanan atas



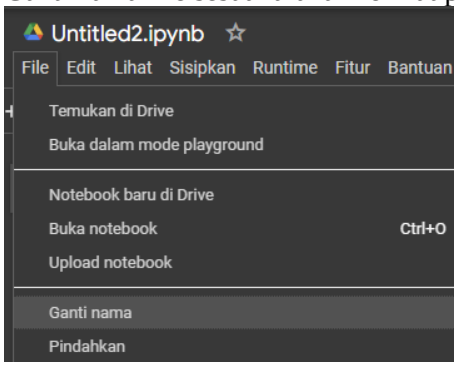
    c. Anda bisa login menggunakan akun Google.
    d. Klik New Notebook pada pojok kiri bawah, untuk membuka halaman baru google colab.



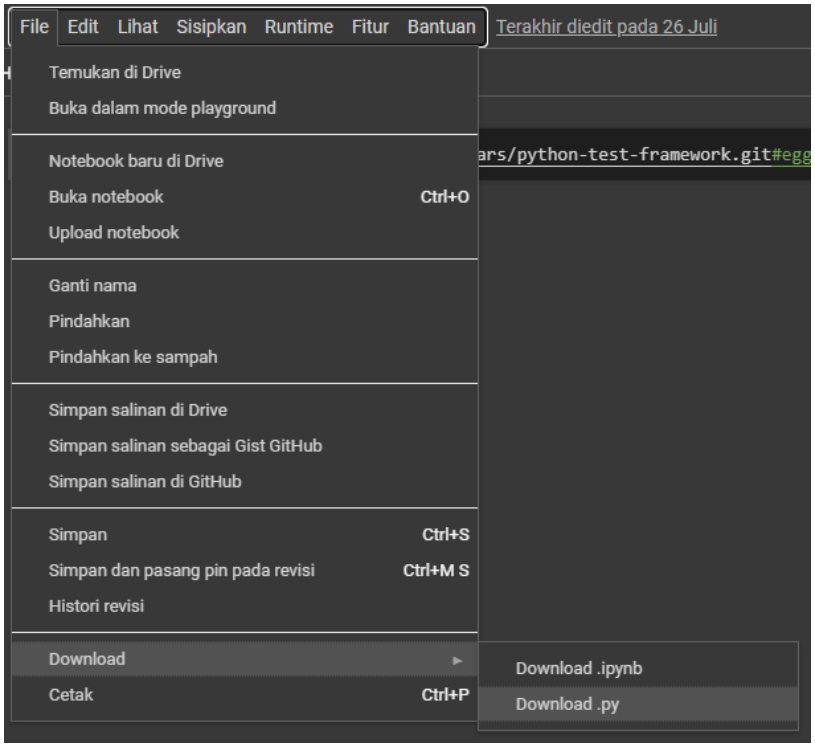
    e. Tampilan Google Colab.



    f. Ganti nama file sesuai arahan format pada praktikum



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



1.4 Contoh Studi Kasus

Contoh : Pembersihan data pada Dataset fitness_data.csv

Pada contoh ini kita akan belajar menampilkan tulisan Hello, World! menggunakan fungsi print() dalam bahasa python. Berikut ini adalah langkah-langkahnya :

Pembersihan Dataset Film

Skenario : Anda memiliki kumpulan data film dalam format CSV. Namun, data tersebut mungkin berisi kesalahan, nilai yang hilang (missing values), dan ketidakkonsistenan. Hal ini dapat mempersulit Anda untuk menganalisis data secara efektif.

Objektif : Tujuan dari studi kasus ini adalah untuk membersihkan kumpulan data film tersebut. Pembersihan data meliputi:

- Membersihkan nama kolom
- Mengubah tipe data kolom tertentu menjadi numerik
- Menangani nilai yang hilang (missing values)
- Menangani ketidakkonsistenan data, seperti karakter khusus atau format yang salah

Langkah – Langkah :

1. **Impor library:** Kita akan menggunakan library pandas untuk memanipulasi data dan NumPy untuk menangani nilai missing value (NaN).

```
import pandas as pd
```

```
import numpy as np  # Import NumPy for NaN handling
```

2. **Baca data film:** Membaca data film dari file CSV menggunakan fungsi pd.read_csv.

```
url =
"https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/
movie_data.csv"
def load_data():

  df = pd.read_csv(url)
  return df
```

3. **Membersihkan nama kolom:** Mengubah nama kolom yang tidak konsisten atau mengandung spasi menjadi lebih deskriptif dan menggunakan snake_case.

```
def clean_columns_name():
  clean_names = {
    "Movie Title": "Movie_Title",
    "Release Year": "Release_Year",
    "Genre": "Genre",
    "Director": "Director",
    "Critic Score (Rotten Tomatoes)": "Critic_Score",
    "User Rating (IMDb)": "User_Rating",
    "Studio": "Studio",
    "Running Time (Minutes)": "Running_Time",
    "Budget (Millions USD)": "Budget",
    "Box Office (Millions USD)": "Box_Office",
    # Handle potential empty string for "Unnamed: 10"
    "Unnamed": None  # This will rename the empty string to None
(effectively dropping it)
  }
  data = clean_columns().rename(columns=clean_names)
  return data
```

4. **Mengubah tipe data:** Mengubah tipe data kolom tertentu menjadi numerik untuk memudahkan perhitungan.

5. **Menangani missing values:**

- Mengganti nilai "N/A" pada kolom "Critic_Score" dengan NaN (Not a Number).
- Mengisi nilai NaN pada kolom numerik ("Budget" dan "Box_Office") dengan mean (rata-rata).
- Menghapus baris dengan nilai missing pada kolom "User_Rating" karena kolom ini penting untuk analisis rating film.

```
# Fix data types (assuming these columns contain numeric data)
def clean_data_types():
  df = clean_columns_name()
  df["Release_Year"] = pd.to_numeric(df["Release_Year"],
errors="coerce")
  df["Critic_Score"] =
pd.to_numeric(df["Critic_Score"].str.replace("%", ""),
errors="coerce")
  df["Budget"] = pd.to_numeric(df["Budget"], errors="coerce")
  df["Box_Office"] = pd.to_numeric(df["Box_Office"],
errors="coerce")

# Handle missing values (replace "N/A" with NaN and fill NaN with the mean for
numeric columns)

  df["Critic_Score"] = df["Critic_Score"].replace("N/A", np.nan)
  df["Budget"] = df["Budget"].fillna(df["Budget"].mean())
  df["Box_Office"] =
df["Box_Office"].fillna(df["Box_Office"].mean())
  df.dropna(subset=["User_Rating"], inplace=True)
  return df
```

6. **Simpan data bersih:** Setelah proses pembersihan selesai, simpan data yang sudah bersih ke file CSV baru.

```
def save_data():
  df = clean_data_types()
  df.to_csv("clean_movie_data.csv", index=False)
```

Tampilan Keseluruhan kode

```
import pandas as pd
import numpy as np

# Read the movie data
url =
"https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/
movie_data.csv"
def load_data():

  df = pd.read_csv(url)
  return df

def clean_columns_name():
  clean_names = {
```

```
    "Movie Title": "Movie_Title",
    "Release Year": "Release_Year",
    "Genre": "Genre",
    "Director": "Director",
    "Critic Score (Rotten Tomatoes)": "Critic_Score",
    "User Rating (IMDb)": "User_Rating",
    "Studio": "Studio",
    "Running Time (Minutes)": "Running_Time",
    "Budget (Millions USD)": "Budget",
    "Box Office (Millions USD)": "Box_Office",
    # Handle potential empty string for "Unnamed: 10"
    "Unnamed": None  # This will rename the empty string to None
(effectively dropping it)
  }
  data = clean_columns().rename(columns=clean_names)
  return data

def clean_data_types():
  df = clean_columns_name()
  df["Release_Year"] = pd.to_numeric(df["Release_Year"],
errors="coerce")
  df["Critic_Score"] =
pd.to_numeric(df["Critic_Score"].str.replace("%", ""),
errors="coerce")
  df["Budget"] = pd.to_numeric(df["Budget"], errors="coerce")
  df["Box_Office"] = pd.to_numeric(df["Box_Office"],
errors="coerce")
  df["Critic_Score"] = df["Critic_Score"].replace("N/A", np.nan)
  df["Budget"] = df["Budget"].fillna(df["Budget"].mean())
  df["Box_Office"] =
df["Box_Office"].fillna(df["Box_Office"].mean())
  df.dropna(subset=["User_Rating"], inplace=True)
  return df


def save_data():
  df = clean_data_types()
  df.to_csv("clean_movie_data.csv", index=False)

# Call the functions
save_data()
```

1.5 Praktikum

Pembersihan Data Perumahan NYC

Skenario:

- Analis data perlu menganalisis data perumahan NYC untuk mengidentifikasi tren pasar, nilai properti, dan rekomendasi investasi.
- Data tidak bersih dan konsisten, dengan nama kolom yang tidak jelas, tipe data yang salah, nilai yang hilang, dan ketidakkonsistenan format.

Objektif:

- Melakukan pembersihan data menyeluruh untuk meningkatkan kualitas, memudahkan analisis, dan mendapatkan wawasan yang lebih baik.

Langkah – Langkah

1. Import pustaka:
   import pandas as pd: Memuat pustaka Pandas untuk manipulasi data.
2. Define URL:
   url = https://raw.githubusercontent.com/noora20FH/skripsi_noora2023/main/nyc_perumahan.csv : Menyimpan URL file CSV data perumahan.
3. Fungsi ***load_data()***:
   3.1 Membaca data CSV dari URL menggunakan ***pd.read_csv(url)***.
   3.2 Mengembalikan DataFrame yang dimuat (***df***).
4. Fungsi ***clean_columns():***
   4.1 Membuat daftar unnecessary_columns berisi kolom yang akan dihapus ***unnecessary_columns*** *= ['BLOCK', 'LOT','EASE-MENT','TAX CLASS AT PRESENT','TAX CLASS AT TIME OF SALE']*
   4.2 Memanggil ***load_data()*** untuk mendapatkan DataFrame.
   4.3 Menghapus kolom yang tidak perlu dari DataFrame ***df.drop(unnecessary_columns, axis=1)***
   4.4 Mengembalikan DataFrame yang sudah dibersihkan (***df***).

5. Fungsi **clean_columns_name()**:
   5.1 Membuat kamus clean_names untuk memetakan nama kolom lama ke nama baru (misalnya, "BUILDING CLASS CATEGORY" menjadi "BUILDING_CLASS_CATEGORY").
   5.2 Memanggil **clean_columns()** untuk mendapatkan DataFrame yang sudah dibersihkan.
   5.3 Mengubah nama kolom dalam DataFrame ***df.rename(columns=clean_names)***

5.4 Mengembalikan nilai DataFrame (***df***).

6. Menampilkan nama kolom baru:
   6.1 Mendapatkan daftar nama kolom baru dari ***clean_columns_name().columns***
   6.2 Mencetak daftar nama kolom baru ***print(...)*** .
7. **Submit**

   Simpan file dengan nama **answer_bab2_percobaan2.py** pastikan menyimpan file dengan format file Python **(.py)**