BAB V
DATA PREPROCESSING

I. TUJUAN

- Mampu mendeteksi dan menghapus (atau memperbaiki) data set yang tidak akurat atau korup menggunakan Python dengan library *pandas* dan *numpy*.

II. PENDAHULUAN

Data Cleaning

Data cleaning merupakan serangkaian proses untuk mengidentifikasi kesalahan pada data baik data tidak akurat maupun korup, kemudian dapat melakukan beberapa koreksi seperti perbaikan atau menghapus data jika diperlukan.

Beberapa fungsi yang terdapat pada pandas untuk melakukan data cleaning:

Fungsi	Keterangan
df.astype(tipe_baru)	Mengkonversi tipe data kolom DataFrame ke tipe
pd.to_numeric(kolom)	Mengkonversi kolom ke tipe numerik
pd.isna() atau pd.isnull()	Mendeteksi nilai yang hilang (NaN, None)
df.fillna(value)	Mengisi nilai yang hilang dengan nilai tertentu
df.dropna()	Menghapus baris yang mengandung nilai yang hilang
df.replace(nilai_lama, nilai_baru)	Mengganti nilai tertentu dengan nilai baru

Nilai hilang default dalam *pandas*:

- NaN - Floating point
- None - Python object
- NaT - Timestamps

III. LANGKAH PRAKTIKUM

1. Install Library yang Dibutuhkan. Jalankan perintah berikut di terminal atau di *cell* Google Colab:

```
pip install pandas numpy
```

2. Buka dan unduh dataset messydata.csv pada link berikut https://s.id/messydata. Di dalam file tersebut memiliki beberapa data hilang seperti NaN, -99, karakter kosong.
3. Langkah berikutnya membuka dataset dan tampilkan isinya menggunakan kode berikut:

```
import pandas as pd
# Membaca file CSV
df = pd.read_csv('messydata.csv')
print(df)
```

```
          A      B    C      D
0   maldimz    3.0  yes    2.0
1      afe2    NaN   no    2.0
2      wth4    3.0  yes    4.0
3      atn2    2.0   no   24.6
4        21   25.0  yes    6.0
5      ple2   45.6  yes    7.0
6      snn3   41.1   no   10.0
7      avg2  234.0  yes   22.0
8      ikk1    2.0  NaN    3.0
9      pkn3    5.0  yes    NaN
10     gry5    5.0   no    3.0
11     1arg  145.0   no   56.0
12     0opw  556.0   no    7.0
13      khj   34.0  yes   12.0
14     lasd  234.0   no    NaN
15    ujgh3    NaN  yes   45.0
16     asd3   12.0  yes   22.0
17     njk4   10.0   no   45.0
18     lkj2    2.0   no  -99.0
19    mn02f   15.0   no  234.0
20    ik67j    1.0  yes    5.0
```

Gambar 5.1 Hasil menampilkan data

4. Mengubah Tipe Data Kolom Mengubah tipe data kolom tertentu menggunakan *astype*():

   Mengubah tipe data *double* menjadi *String*

```
df['B'] = df['B'].astype('string')
print(df)
```

```
          A      B    C      D
0   maldimz    3.0  yes    2.0
1      afe2   <NA>   no    2.0
2      wth4    3.0  yes    4.0
3      atn2    2.0   no   24.6
4        21   25.0  yes    6.0
5      ple2   45.6  yes    7.0
6      snn3   41.1   no   10.0
7      avg2  234.0  yes   22.0
8      ikk1    2.0  NaN    3.0
9      pkn3    5.0  yes    NaN
10     gry5    5.0   no    3.0
11     1arg  145.0   no   56.0
12     0opw  556.0   no    7.0
13      khj   34.0  yes   12.0
14     lasd  234.0   no    NaN
15    ujgh3   <NA>  yes   45.0
16     asd3   12.0  yes   22.0
17     njk4   10.0   no   45.0
18     lkj2    2.0   no  -99.0
19    mn02f   15.0   no  234.0
20    ik67j    1.0  yes    5.0
```

Gambar 5.2 Hasil setelah mengubah menjadi string

Mengubah tipe data *String* menjadi *double*

```
df['B'] = df['B'].astype('float')
print(df)
```

```
           A      B    C      D
0   maldimz    3.0  yes    2.0
1      afe2    NaN   no    2.0
2      wth4    3.0  yes    4.0
3      atn2    2.0   no   24.6
4        21   25.0  yes    6.0
5      ple2   45.6  yes    7.0
6      snn3   41.1   no   10.0
7      avg2  234.0  yes   22.0
8      ikk1    2.0  NaN    3.0
9      pkn3    5.0  yes    NaN
10     gry5    5.0   no    3.0
11     1arg  145.0   no   56.0
12     0opw  556.0   no    7.0
13      khj   34.0  yes   12.0
14     lasd  234.0   no    NaN
15    ujgh3    NaN  yes   45.0
16     asd3   12.0  yes   22.0
17     njk4   10.0   no   45.0
18     lkj2    2.0   no -99.0
```

Gambar 5.3 Hasil setelah mengubah menjadi float

5. Mengecek tipe data dan statistik dataset gunakan fungsi berikut untuk melihat tipe data dan statistik dataset:

```
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   A       21 non-null     object
 1   B       19 non-null     float64
 2   C       20 non-null     object
 3   D       19 non-null     float64
dtypes: float64(2), object(2)
memory usage: 804.0+ bytes
None
                B           D
count   19.000000   19.000000
mean    72.352632   21.610526
std    138.706206   60.040670
min      1.000000  -99.000000
25%      3.000000    3.500000
50%     12.000000    7.000000
75%     43.350000   23.300000
max    556.000000  234.000000
```

Gambar 5.4 Hasil untuk melihat info dan statistic data

6. Mendeteksi dan mengecek nilai null/hilang dalam dataset:

```
missing_values = df.isnull()
print(missing_values)
```

	A	B	C	D
0	False	False	False	False
1	False	True	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
5	False	False	False	False
6	False	False	False	False
7	False	False	False	False
8	False	False	True	False
9	False	False	False	True
10	False	False	False	False
11	False	False	False	False
12	False	False	False	False
13	False	False	False	False
14	False	False	False	True
15	False	True	False	False
16	False	False	False	False
17	False	False	False	False
18	False	False	False	False
19	False	False	False	False
20	False	False	False	False

Gambar 5.5 Pengecekan nilai null/hilang pada data

Menampilkan baris yang mengandung nilai null:

```
rows_with_missing = df[df.isnull().any(axis=1)]
print(rows_with_missing)
```

	A	B	C	D
1	afe2	NaN	no	2.0
8	ikk1	2.0	NaN	3.0
9	pkn3	5.0	yes	NaN
14	lasd	234.0	no	NaN
15	ujgh3	NaN	yes	45.0

Gambar 5.6 Hasil untuk menampilkan baris yang berisi null

7. Mengganti nilai hilang atau mengganti nilai tertentu (misalnya -99) dengan NaN menggunakan *replace()*:

```
import numpy as np

df.replace(-99, np.nan, inplace=True)
print(df)
```

	A	B	C	D
0	maldimz	3.0	yes	2.0
1	afe2	NaN	no	2.0
2	wth4	3.0	yes	4.0
3	atn2	2.0	no	24.6
4	21	25.0	yes	6.0
5	ple2	45.6	yes	7.0
6	snn3	41.1	no	10.0
7	avg2	234.0	yes	22.0
8	ikk1	2.0	NaN	3.0
9	pkn3	5.0	yes	NaN
10	gry5	5.0	no	3.0
11	1arg	145.0	no	56.0
12	0opw	556.0	no	7.0
13	khj	34.0	yes	12.0
14	lasd	234.0	no	NaN
15	ujgh3	NaN	yes	45.0
16	asd3	12.0	yes	22.0
17	njk4	10.0	no	45.0
18	lkj2	2.0	no	NaN
19	mn02f	15.0	no	234.0
20	ik67j	1.0	yes	5.0

Gambar 5.7 Hasil setelah mengganti nilai tertentu

8. Memperbaiki nilai hilang atau mengisi nilai hilang dengan metode tertentu:

```
# Mengisi nilai hilang dengan nilai sebelumnya
df.fillna(method='ffill', inplace=True)
print(df)
```

	A	B	C	D
0	maldimz	3.0	yes	2.0
1	afe2	3.0	no	2.0
2	wth4	3.0	yes	4.0
3	atn2	2.0	no	24.6
4	21	25.0	yes	6.0
5	ple2	45.6	yes	7.0
6	snn3	41.1	no	10.0
7	avg2	234.0	yes	22.0
8	ikk1	2.0	yes	3.0
9	pkn3	5.0	yes	3.0
10	gry5	5.0	no	3.0
11	1arg	145.0	no	56.0
12	0opw	556.0	no	7.0
13	khj	34.0	yes	12.0
14	lasd	234.0	no	12.0
15	ujgh3	234.0	yes	45.0
16	asd3	12.0	yes	22.0
17	njk4	10.0	no	45.0
18	lkj2	2.0	no	45.0
19	mn02f	15.0	no	234.0
20	ik67j	1.0	yes	5.0

Gambar 5.8 Hasil setelah memperbaiki atau mengisi nilai hilang

9. Menghapus nilai hilang atau baris yang mengandung nilai null:

```
df_cleaned = df.dropna()
print(df_cleaned)
```

	A	B	C	D
0	maldimz	3.0	yes	2.0
1	afe2	3.0	no	2.0
2	wth4	3.0	yes	4.0
3	atn2	2.0	no	24.6
4	21	25.0	yes	6.0
5	ple2	45.6	yes	7.0
6	snn3	41.1	no	10.0
7	avg2	234.0	yes	22.0
8	ikk1	2.0	yes	3.0
9	pkn3	5.0	yes	3.0
10	gry5	5.0	no	3.0
11	1arg	145.0	no	56.0
12	0opw	556.0	no	7.0
13	khj	34.0	yes	12.0
14	lasd	234.0	no	12.0
15	ujgh3	234.0	yes	45.0
16	asd3	12.0	yes	22.0
17	njk4	10.0	no	45.0
18	lkj2	2.0	no	45.0
19	mn02f	15.0	no	234.0
20	ik67j	1.0	yes	5.0

Gambar 5.9 Hasil setelah menghapus baris dengan nilai null

10. Menyimpan data setelah *cleaning* ke file yang baru:

```
df_cleaned.to_csv('cleaned_data.csv', index=False)
```

IV. TUGAS PRAKTIKUM

1. Unduh dataset dari tautan yang diberikan, buka dataset tersebut, dan tampilkan data sebelum proses preprocessing.
2. Identifikasi nilai hilang dalam dataset, ganti nilai hilang dengan metode yang sesuai (misalnya, nilai sebelumnya atau rata-rata).
3. Hapus baris dengan nilai hilang, jika diperlukan.
4. Tampilkan dataset sebelum dan setelah proses preprocessing.
5. Gunakan Google Colab atau VSCode untuk menyelesaikan tugas ini.
6. Jelaskan setiap langkah proses preprocessing data dengan benar!