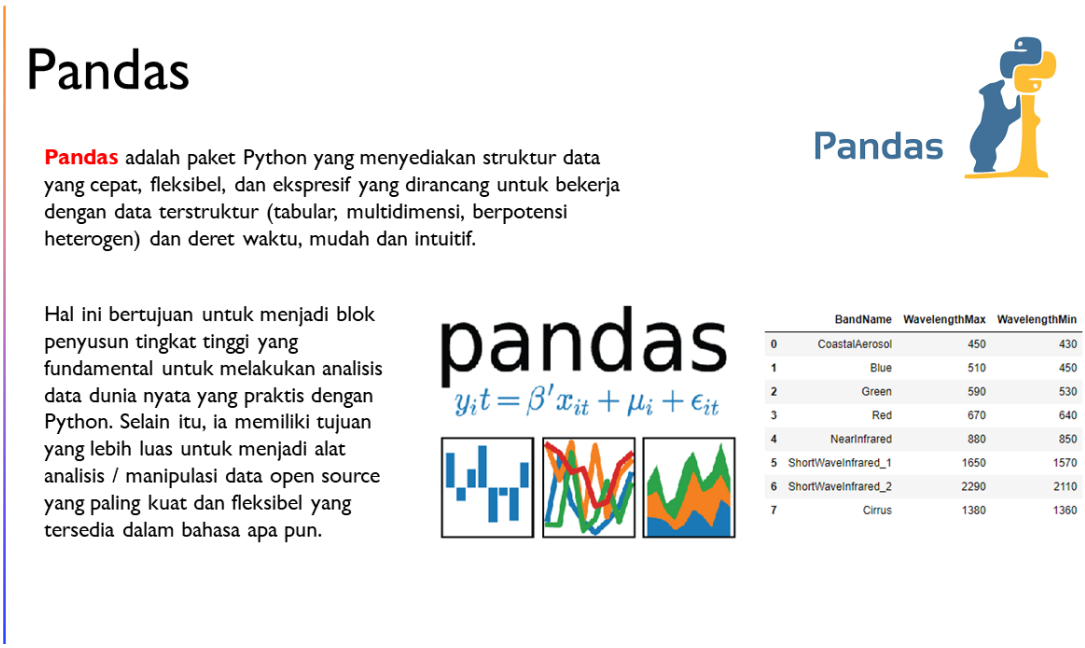
Python Table Management and Data Preprocessing

Pandas memiliki banyak kegunaan, pandas mampu menyajikan hal-hal pengolahan data yang rumit menjadi sederhana, membantu mempercepat proses penyajian data dan analasis data



Pandas DataFrame

Membuat Tabel dengan dataFrame

In [1]:
```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50,40,45],
  "size": [80,75,60]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

```
   calories  duration  size
0       420        50    80
1       380        40    75
2       390        45    60
```

Untuk melihat data pada baris ke 0

In [2]:
```
print(df.loc[1])
```

```
calories    380
duration     40
size         75
Name: 1, dtype: int64
```

Untuk melihat baris ke 0 dan 1

In [3]:
```
print(df.iloc[0:2])
```

```
   calories  duration  size
0       420        50    80
1       380        40    75
```

Mengganti nama baris 0, 1 dan 2 menjadi day1, day2, dan day3

In [4]:
```
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])

print(df)
```

```
      calories  duration  size
day1       420        50    80
day2       380        40    75
day3       390        45    60
```

Memanggil data baris day2

In [5]:
```
print(df.loc["day2"])
```

```
calories    380
duration     40
size         75
Name: day2, dtype: int64
```

Memisahkan beberapa kolom dari tabel

In [6]:
```
result = df.loc[:, ["calories","duration" ]]
result
```

Out[6]:

	calories	duration
day1	420	50
day2	380	40
day3	390	45

Load File

Pastikan file data CSV ada di folder yang sama dengan file Python.

In [7]:
```
import pandas as pd

df = pd.read_csv('california_housing_test.csv')

print(df)
```

```
      longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0       -122.05     37.37                27.0       3885.0           661.0
1       -118.30     34.26                43.0       1510.0           310.0
2       -117.81     33.78                27.0       3589.0           507.0
3       -118.36     33.82                28.0         67.0            15.0
4       -119.67     36.33                19.0       1241.0           244.0
...         ...       ...                 ...          ...             ...
2995    -119.86     34.42                23.0       1450.0           642.0
2996    -118.14     34.06                27.0       5257.0          1082.0
2997    -119.70     36.30                10.0        956.0           201.0
2998    -117.12     34.10                40.0         96.0            14.0
```

```
2999    -119.63     34.42                42.0       1765.0           263.0

      population  households  median_income  median_house_value
0         1537.0       606.0         6.6085            344700.0
1          809.0       277.0         3.5990            176500.0
2         1484.0       495.0         5.7934            270500.0
3           49.0        11.0         6.1359            330000.0
4          850.0       237.0         2.9375             81700.0
...          ...         ...            ...                 ...
2995      1258.0       607.0         1.1790            225000.0
2996      3496.0      1036.0         3.3906            237200.0
2997       693.0       220.0         2.2895             62000.0
2998        46.0        14.0         3.2708            162500.0
2999       753.0       260.0         8.5608            500001.0

[3000 rows x 9 columns]
```

In [8]:

```python
print(df.head(10))
```

```
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -122.05     37.37                27.0       3885.0           661.0
1    -118.30     34.26                43.0       1510.0           310.0
2    -117.81     33.78                27.0       3589.0           507.0
3    -118.36     33.82                28.0         67.0            15.0
4    -119.67     36.33                19.0       1241.0           244.0
5    -119.56     36.51                37.0       1018.0           213.0
6    -121.43     38.63                43.0       1009.0           225.0
7    -120.65     35.48                19.0       2310.0           471.0
8    -122.84     38.40                15.0       3080.0           617.0
9    -118.02     34.08                31.0       2402.0           632.0

   population  households  median_income  median_house_value
0      1537.0       606.0         6.6085            344700.0
1       809.0       277.0         3.5990            176500.0
2      1484.0       495.0         5.7934            270500.0
3        49.0        11.0         6.1359            330000.0
4       850.0       237.0         2.9375             81700.0
5       663.0       204.0         1.6635             67000.0
6       604.0       218.0         1.6641             67000.0
7      1341.0       441.0         3.2250            166900.0
8      1446.0       599.0         3.6696            194400.0
9      2830.0       603.0         2.3333            164200.0
```

In [9]:

```python
print(df.tail())
```

```
      longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
2995    -119.86     34.42                23.0       1450.0           642.0
2996    -118.14     34.06                27.0       5257.0          1082.0
2997    -119.70     36.30                10.0        956.0           201.0
2998    -117.12     34.10                40.0         96.0            14.0
2999    -119.63     34.42                42.0       1765.0           263.0

      population  households  median_income  median_house_value
2995      1258.0       607.0         1.1790            225000.0
2996      3496.0      1036.0         3.3906            237200.0
2997       693.0       220.0         2.2895             62000.0
2998        46.0        14.0         3.2708            162500.0
2999       753.0       260.0         8.5608            500001.0
```

In [10]:

```python
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           3000 non-null   float64
 1   latitude            3000 non-null   float64
 2   housing_median_age  3000 non-null   float64
 3   total_rooms         3000 non-null   float64
 4   total_bedrooms      3000 non-null   float64
 5   population          3000 non-null   float64
 6   households          3000 non-null   float64
 7   median_income       3000 non-null   float64
 8   median_house_value  3000 non-null   float64
dtypes: float64(9)
memory usage: 211.1 KB
None
```

In [11]:

```
df.describe()
```

Out[11]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	h
count	3000.000000	3000.00000	3000.000000	3000.000000	3000.000000	3000.000000	
mean	-119.589200	35.63539	28.845333	2599.578667	529.950667	1402.798667	
std	1.994936	2.12967	12.555396	2155.593332	415.654368	1030.543012	
min	-124.180000	32.56000	1.000000	6.000000	2.000000	5.000000	
25%	-121.810000	33.93000	18.000000	1401.000000	291.000000	780.000000	
50%	-118.485000	34.27000	29.000000	2106.000000	437.000000	1155.000000	
75%	-118.020000	37.69000	37.000000	3129.000000	636.000000	1742.750000	
max	-114.490000	41.92000	52.000000	30450.000000	5419.000000	11935.000000	

Data pre-Processing

Dalam pre processing data terdapat beberapa aspek berikut ini:

1. missing values
2. data standardization
3. data normalization
4. data binning

Import Data

In [12]:

```
import pandas as pd
df = pd.read_csv('hepatitis_csv.csv')
df.head(10)
```

Out[12]:

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable
0	30	male	False	False	False	False	False	False	False	False
1	50	female	False	False	True	False	False	False	False	False
2	78	female	True	False	True	False	False	True	False	False

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable
3	31	female	NaN	True	False	False	False	True	False	False
4	34	female	True	False	False	False	False	True	False	False
5	34	female	True	False	False	False	False	True	False	False
6	51	female	False	False	True	False	True	True	False	True
7	23	female	True	False	False	False	False	True	False	False
8	39	female	True	False	True	False	False	True	True	False
9	30	female	True	False	False	False	False	True	False	False

Identifikasi Missing Value

In [13]:
```
df.isna().sum()
```

Out[13]:
```
age                 0
sex                 0
steroid             1
antivirals          0
fatigue             1
malaise             1
anorexia            1
liver_big          10
liver_firm         11
spleen_palpable     5
spiders             5
ascites             5
varices             5
bilirubin           6
alk_phosphate      29
sgot                4
albumin            16
protime            67
histology           0
class               0
dtype: int64
```

Cetak dalam bentuk presentase

In [14]:
```
df.isna().sum()/len(df)*100
```

Out[14]:
```
age                 0.000000
sex                 0.000000
steroid             0.645161
antivirals          0.000000
fatigue             0.645161
malaise             0.645161
anorexia            0.645161
liver_big           6.451613
liver_firm          7.096774
spleen_palpable     3.225806
spiders             3.225806
ascites             3.225806
varices             3.225806
bilirubin           3.870968
alk_phosphate      18.709677
sgot                2.580645
```

```
albumin            10.322581
protime            43.225806
histology           0.000000
class               0.000000
dtype: float64
```

Drop Missing Values

Menghampus semua kolom yang memiliki missing value

In [15]:

```
df.dropna( )
```

Out[15]:

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpab
5	34	female	True	False	False	False	False	True	False	Fal
10	39	female	False	True	False	False	False	False	True	Fal
11	32	female	True	True	True	False	False	True	True	Fal
12	41	female	True	True	True	False	False	True	True	Fal
13	30	female	True	False	True	False	False	True	True	Fal
...	...	...	...	...	...	...	...	...	...	
139	45	female	True	True	False	False	False	True	False	Fal
143	49	female	False	False	True	True	False	True	False	Tru
145	31	female	False	False	True	False	False	True	False	Fal
153	53	male	False	False	True	False	False	True	False	Tru
154	43	female	True	False	True	False	False	True	False	Tru

80 rows × 20 columns

Drop baris dari spesifik kolom

In [16]:

```
df = pd.read_csv('hepatitis_csv.csv')
df.dropna(subset=['protime'],axis=0,inplace=True)
df.isna().sum()/len(df)*100
```

Out[16]:

```
age                0.000000
sex                0.000000
steroid            1.136364
antivirals         0.000000
fatigue            0.000000
malaise            0.000000
anorexia           0.000000
liver_big          2.272727
liver_firm         2.272727
spleen_palpable    1.136364
spiders            1.136364
ascites            1.136364
varices            1.136364
bilirubin          0.000000
alk_phosphate      4.545455
sgot               0.000000
albumin            1.136364
protime            0.000000
histology          0.000000
```

```
class               0.000000
dtype: float64
```

In [17]:

```
df.head(10)
```

Out[17]:

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable
3	31	female	NaN	True	False	False	False	True	False	False
5	34	female	True	False	False	False	False	True	False	False
10	39	female	False	True	False	False	False	False	True	False
11	32	female	True	True	True	False	False	True	True	False
12	41	female	True	True	True	False	False	True	True	False
13	30	female	True	False	True	False	False	True	True	False
15	38	female	False	False	True	True	True	True	False	False
17	40	female	False	False	True	False	False	True	True	False
18	38	female	True	False	False	False	False	True	False	False
19	38	female	False	True	False	False	False	False	True	False

Mengganti data Missing Values

Strategi yang baik saat menangani nilai yang hilang melibatkan penggantiannya dengan nilai lain. Biasanya, strategi berikut ini:

1. untuk nilai numerik ganti nilai yang hilang dengan nilai rata-rata kolom
2. untuk nilai kategorial ganti nilai yang hilang dengan nilai kolom yang paling sering
3. gunakan fungsi lain

Untuk mengganti nilai yang hilang, bisanya digunakan tiga fungsi seperti: fillna(), replace() dan interpolate().

In [18]:

```
df = pd.read_csv('hepatitis_csv.csv')
df.dtypes
```

Out[18]:

```
age                 int64
sex                object
steroid            object
antivirals           bool
fatigue            object
malaise            object
anorexia           object
liver_big          object
liver_firm         object
spleen_palpable    object
spiders            object
ascites            object
varices            object
bilirubin         float64
alk_phosphate     float64
sgot              float64
albumin           float64
protime           float64
histology            bool
```

```
class                object
dtype: object
```

Kolom Numerik

In [19]:
```
import numpy as np
numeric = df.select_dtypes(include=np.number)
numeric_columns = numeric.columns
```

Isi kolom numerik dengan nilai rataan

In [20]:
```
df[numeric_columns] = df[numeric_columns].fillna(df.mean())
```

```
C:\Users\USER\AppData\Local\Temp/ipykernel_10040/3464485706.py:1: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  df[numeric_columns] = df[numeric_columns].fillna(df.mean())
```

Melihat hasilnya adalah

In [21]:
```
df.isna().sum()/len(df)*100
```

Out[21]:
```
age                 0.000000
sex                 0.000000
steroid             0.645161
antivirals          0.000000
fatigue             0.645161
malaise             0.645161
anorexia            0.645161
liver_big           6.451613
liver_firm          7.096774
spleen_palpable     3.225806
spiders             3.225806
ascites             3.225806
varices             3.225806
bilirubin           0.000000
alk_phosphate       0.000000
sgot                0.000000
albumin             0.000000
protime             0.000000
histology           0.000000
class               0.000000
dtype: float64
```

Kolom Kategorial

In [22]:
```
boolean_columns = df.select_dtypes(include=np.object).columns.tolist()
boolean_columns.remove('class')
df[boolean_columns] = df[boolean_columns].astype('bool')
```

```
C:\Users\USER\AppData\Local\Temp/ipykernel_10040/2487696438.py:1: DeprecationWarnin
g: `np.object` is a deprecated alias for the builtin `object`. To silence this warni
ng, use `object` by itself. Doing this will not modify any behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
elease/1.20.0-notes.html#deprecations
  boolean_columns = df.select_dtypes(include=np.object).columns.tolist()
```

Gunakan fungsi mode() untuk mencari nilai yang sering muncul.

In [23]:

```
df[boolean_columns].fillna(df.mode())
```

Out[23]:

	sex	steroid	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable	spiders	ascites
0	True	False	False	False	False	False	False	False	False	False
1	True	False	True	False	False	False	False	False	False	False
2	True	True	True	False	False	True	False	False	False	False
3	True	True	False	False	False	True	False	False	False	False
4	True	True	False	False	False	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...	..
150	True	True	True	True	True	True	False	False	True	True
151	True	True	True	False	False	True	True	False	False	False
152	True	False	True	True	False	False	True	False	True	False
153	True	False	True	False	False	True	False	True	True	False
154	True	True	True	False	False	True	False	True	True	True

155 rows × 11 columns

Melihat hasilnya

In [24]:

```
df.isna().sum()/len(df)*100
```

Out[24]:

```
age                 0.0
sex                 0.0
steroid             0.0
antivirals          0.0
fatigue             0.0
malaise             0.0
anorexia            0.0
liver_big           0.0
liver_firm          0.0
spleen_palpable     0.0
spiders             0.0
ascites             0.0
varices             0.0
bilirubin           0.0
alk_phosphate       0.0
sgot                0.0
albumin             0.0
protime             0.0
histology           0.0
class               0.0
dtype: float64
```

Interpolasi

Solusi lain untuk mengganti nilai yang hilang melibatkan penggunaan fungsi lain, seperti interpolasi linier. Dalam kasus ini, misalnya, kita bisa mengganti nilai yang hilang di atas kolom, dengan interpolasi antara yang sebelumnya dan yang berikutnya. Ini dapat dicapai melalui penggunaan fungsi interpolate ().

In [25]:

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

```
df.isna().sum()/len(df)*100
```

Out[25]:

```
age                 0.000000
sex                 0.000000
steroid             0.645161
antivirals          0.000000
fatigue             0.645161
malaise             0.645161
anorexia            0.645161
liver_big           6.451613
liver_firm          7.096774
spleen_palpable     3.225806
spiders             3.225806
ascites             3.225806
varices             3.225806
bilirubin           3.870968
alk_phosphate      18.709677
sgot                2.580645
albumin            10.322581
protime            43.225806
histology           0.000000
class               0.000000
dtype: float64
```

Pilih kolom yang bertipe numerik

In [26]:

```
numeric = df.select_dtypes(include=np.number)
numeric_columns = numeric.columns
df.head(10)
```

Out[26]:

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable
0	30	male	False	False	False	False	False	False	False	False
1	50	female	False	False	True	False	False	False	False	False
2	78	female	True	False	True	False	False	True	False	False
3	31	female	NaN	True	False	False	False	True	False	False
4	34	female	True	False	False	False	False	True	False	False
5	34	female	True	False	False	False	False	True	False	False
6	51	female	False	False	True	False	True	True	False	True
7	23	female	True	False	False	False	False	True	False	False
8	39	female	True	False	True	False	False	True	True	False
9	30	female	True	False	False	False	False	True	False	False

Sekarang kita dapat menerapkan fungsi interpolate () ke kolom numerik, dengan mengatur juga arah batas ke depan. Artinya interpolasi linier diterapkan mulai dari baris pertama hingga baris terakhir.

In [27]:

```
df[numeric_columns] = df[numeric_columns].interpolate(method ='linear', limit_direct
```

In [28]:

```
df.head(10)
```

Out[28]:

	age	sex	steroid	antivirals	fatigue	malaise	anorexia	liver_big	liver_firm	spleen_palpable
0	30	male	False	False	False	False	False	False	False	False
1	50	female	False	False	True	False	False	False	False	False
2	78	female	True	False	True	False	False	True	False	False
3	31	female	NaN	True	False	False	False	True	False	False
4	34	female	True	False	False	False	False	True	False	False
5	34	female	True	False	False	False	False	True	False	False
6	51	female	False	False	True	False	True	True	False	True
7	23	female	True	False	False	False	False	True	False	False
8	39	female	True	False	True	False	False	True	True	False
9	30	female	True	False	False	False	False	True	False	False

In [29]:

```
df.isna().sum()/len(df)*100
```

Out[29]:

```
age                 0.000000
sex                 0.000000
steroid             0.645161
antivirals          0.000000
fatigue             0.645161
malaise             0.645161
anorexia            0.645161
liver_big           6.451613
liver_firm          7.096774
spleen_palpable     3.225806
spiders             3.225806
ascites             3.225806
varices             3.225806
bilirubin           0.000000
alk_phosphate       0.000000
sgot                0.000000
albumin             0.000000
protime             1.935484
histology           0.000000
class               0.000000
dtype: float64
```

Excerise

Life Expectancy (WHO)

https://www.kaggle.com/datasets/augustus0498/life-expectancy-who

In [30]:

```
import pandas as pd
df = pd.read_csv('Life Expectancy Data.csv')
df
```

Out[30]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0
...	...	...	...	...	...	...	...	...	...
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0

2938 rows × 22 columns

In [31]:

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year                             2938 non-null   int64
 2   Status                           2938 non-null   object
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years              2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
None
```

In [32]:

```
df.describe()
```

Out[32]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000

Data pre-Processing

Identifikasi Missing Value

In [33]:
```
df.isna().sum()
```

Out[33]:
```
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

Cetak dalam bentuk presentase

In [34]:
```
df.isna().sum()/len(df)*100
```

Out[34]:
```
Country                            0.000000
Year                               0.000000
Status                             0.000000
Life expectancy                    0.340368
Adult Mortality                    0.340368
infant deaths                      0.000000
Alcohol                            6.603131
```

```
percentage expenditure              0.000000
Hepatitis B                        18.822328
Measles                             0.000000
 BMI                                1.157250
under-five deaths                   0.000000
Polio                               0.646698
Total expenditure                   7.692308
Diphtheria                          0.646698
 HIV/AIDS                           0.000000
GDP                                15.248468
Population                         22.191967
 thinness  1-19 years               1.157250
 thinness 5-9 years                 1.157250
Income composition of resources     5.684139
Schooling                           5.547992
dtype: float64
```

Drop Missing Values

Drop baris dari spesifik kolom

In [35]:

```
df.dropna(subset=['Total expenditure'],axis=0,inplace=True)
df.isna().sum()/len(df)*100
```

Out[35]:

```
Country                             0.000000
Year                                0.000000
Status                              0.000000
Life expectancy                     0.368732
Adult Mortality                     0.368732
infant deaths                       0.000000
Alcohol                             0.184366
percentage expenditure              0.000000
Hepatitis B                        19.026549
Measles                             0.000000
 BMI                                0.737463
under-five deaths                   0.000000
Polio                               0.294985
Total expenditure                   0.000000
Diphtheria                          0.294985
 HIV/AIDS                           0.000000
GDP                                14.011799
Population                         21.423304
 thinness  1-19 years               0.737463
 thinness 5-9 years                 0.737463
Income composition of resources     4.682891
Schooling                           4.535398
dtype: float64
```

In [36]:

```
df.head(10)
```

Out[36]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
5	Afghanistan	2010	Developing	58.8	279.0	74	0.01	79.679367	66.0	
6	Afghanistan	2009	Developing	58.6	281.0	77	0.01	56.762217	63.0	
7	Afghanistan	2008	Developing	58.1	287.0	80	0.03	25.873925	64.0	
8	Afghanistan	2007	Developing	57.5	295.0	82	0.02	10.910156	63.0	
9	Afghanistan	2006	Developing	57.3	295.0	84	0.03	17.171518	64.0	

10 rows × 22 columns

Mengganti data Missing Values

In [37]:
```
df.dtypes
```

Out[37]:
```
Country                              object
Year                                  int64
Status                               object
Life expectancy                     float64
Adult Mortality                     float64
infant deaths                         int64
Alcohol                             float64
percentage expenditure              float64
Hepatitis B                         float64
Measles                               int64
 BMI                                float64
under-five deaths                     int64
Polio                               float64
Total expenditure                   float64
Diphtheria                          float64
 HIV/AIDS                           float64
GDP                                 float64
Population                          float64
 thinness  1-19 years               float64
 thinness 5-9 years                 float64
Income composition of resources     float64
Schooling                           float64
dtype: object
```

Kolom Numerik

In [38]:
```
import numpy as np
numeric = df.select_dtypes(include=np.number)
numeric_columns = numeric.columns
```

In [39]:
```
df[numeric_columns] = df[numeric_columns].fillna(df.mean())
```

```
C:\Users\USER\AppData\Local\Temp/ipykernel_10040/3464485706.py:1: FutureWarning: Dro
pping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depr
ecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  df[numeric_columns] = df[numeric_columns].fillna(df.mean())
```

Melihat hasilnya adalah

In [40]:
```
df.isna().sum()/len(df)*100
```

Out[40]:
```
Country                            0.0
Year                               0.0
Status                             0.0
Life expectancy                    0.0
Adult Mortality                    0.0
infant deaths                      0.0
Alcohol                            0.0
percentage expenditure             0.0
Hepatitis B                        0.0
Measles                            0.0
 BMI                               0.0
under-five deaths                  0.0
Polio                              0.0
Total expenditure                  0.0
Diphtheria                         0.0
 HIV/AIDS                          0.0
GDP                                0.0
Population                         0.0
 thinness  1-19 years              0.0
 thinness 5-9 years                0.0
Income composition of resources    0.0
Schooling                          0.0
dtype: float64
```

Kolom Kategorial

In [41]:
```
boolean_columns = df.select_dtypes(include=np.object).columns.tolist()
boolean_columns.remove('Status')
df[boolean_columns] = df[boolean_columns].astype('bool')
```

```
C:\Users\USER\AppData\Local\Temp/ipykernel_10040/1529296272.py:1: DeprecationWarnin
g: `np.object` is a deprecated alias for the builtin `object`. To silence this warni
ng, use `object` by itself. Doing this will not modify any behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
elease/1.20.0-notes.html#deprecations
  boolean_columns = df.select_dtypes(include=np.object).columns.tolist()
```

In [42]:
```
df[boolean_columns].fillna(df.mode())
```

Out[42]:

	Country
0	True
1	True
2	True
3	True
4	True
...	...
2933	True
2934	True
2935	True
2936	True
2937	True

2712 rows × 1 columns

Melihat hasilnya

In [43]:
```
df.isna().sum()/len(df)*100
```

Out[43]:
```
Country                           0.0
Year                              0.0
Status                            0.0
Life expectancy                   0.0
Adult Mortality                   0.0
infant deaths                     0.0
Alcohol                           0.0
percentage expenditure            0.0
Hepatitis B                       0.0
Measles                           0.0
 BMI                              0.0
under-five deaths                 0.0
Polio                             0.0
Total expenditure                 0.0
Diphtheria                        0.0
 HIV/AIDS                         0.0
GDP                               0.0
Population                        0.0
 thinness  1-19 years             0.0
 thinness 5-9 years               0.0
Income composition of resources   0.0
Schooling                         0.0
dtype: float64
```