SOAL KE-1

1.A. Penanganan missing value

membaca data dari **datamhs.xlsx**

```
import pandas as pd
import numpy as np

df= pd.read_excel("datamhs.xlsx")
df
```

	NPM	nama	jenis_kelamin	jenis_sekolah_lanjutan	status_sma/k	ipk_sem_1	ipk_s
0	1901	Solpo	L	SMK	S	3.0	
1	1902	Sopli	L	SMA	N	2.0	
2	1903	NaN	L	SMA	S	2.5	
3	1904	Soplu	P	SMA	N	3.1	
4	1905	Soplo	NaN	SMK	N	2.0	
5	1906	Sopli	L	SMA	S	2.0	
6	1907	Sopla	L	SMA	S	NaN	
7	1908	Soplu	P	SMA	N	NaN	

mengubah nilai kosong menjadi **none/Null/NA/Na** agar dikenali oleh pandas

```
df = df.replace('', np.NaN)
df
```

	NPM	nama	jenis_kelamin	jenis_sekolah_lanjutan	status_sma/k	ipk_sem_1	ipk_s
0	1901	Solpo	L	SMK	S	3.0	
1	1902	Sopli	L	SMA	N	2.0	
2	1903	NaN	L	SMA	S	2.5	
3	1904	Soplu	P	SMA	N	3.1	
4	1905	Soplo	NaN	SMK	N	2.0	
5	1906	Sopli	L	SMA	S	2.0	

7	1908	Soplu	P	SMA	N	NaN

menghitung jumlah nilai kosong pada tabel data di tiap kolom

```
df.isnull().sum()
```

```
NPM                       0
nama                      1
jenis_kelamin             1
jenis_sekolah_lanjutan    0
status_sma/k              0
ipk_sem_1                 2
ipk_sem_2                 0
status_kelulusan          0
dtype: int64
```

melakukan Imputasi untuk data kategorik dan numerik

```
dfcopy = df
subs = "null"
subs1 = 0
dfcopy[['nama','jenis_kelamin']] = dfcopy[['nama','jenis_kelamin']].replace(np.NAN, subs)
dfcopy[['ipk_sem_1','ipk_sem_2']] = dfcopy[['ipk_sem_1','ipk_sem_2']].replace(np.NAN, subs1
dfcopy
```

	NPM	nama	jenis_kelamin	jenis_sekolah_lanjutan	status_sma/k	ipk_sem_1	ipk_s
0	1901	Solpo	L	SMK	S	3.0	
1	1902	Sopli	L	SMA	N	2.0	
2	1903	null	L	SMA	S	2.5	
3	1904	Soplu	P	SMA	N	3.1	
4	1905	Soplo	null	SMK	N	2.0	
5	1906	Sopli	L	SMA	S	2.0	
6	1907	Sopla	L	SMA	S	0.0	
7	1908	Soplu	P	SMA	N	0.0	

1.B. Statistik Deskriptif

statistik deskriptif untuk data kategorik

```
df[['nama','jenis_kelamin','jenis_sekolah_lanjutan','status_sma/k','status_kelulusan']].des
```

	nama	jenis_kelamin	jenis_sekolah_lanjutan	status_sma/k	status_kelulusan
count	8	8	8	8	8
unique	6	3	2	2	2
top	Sopli	L	SMA	S	Tepat
freq	2	5	6	4	4

Dari data tersebut dapat diketahui bahwa, kolom **nama** memiliki :

- jumlah data 8
- jumlah nilai unik 6
- nilai paling banyak (modus) adalah Sopli
- jumlah nilai paling banyak adalah 2

kolom **jenis_kelamin** memiliki :

- jumlah data 8
- jumlah nilai unik 3
- nilai paling banyak (modus) adalah L
- jumlah nilai paling banyak adalah 5

kolom **jenis_sekolah_lanjutan** memiliki :

- jumlah data 8
- jumlah nilai unik 2
- nilai paling banyak (modus) adalah SMA
- jumlah nilai paling banyak adalah 6

kolom **status_sma/k** memiliki :

- jumlah data 8
- jumlah nilai unik 2
- nilai paling banyak (modus) adalah S
- jumlah nilai paling banyak adalah 4

kolom **status_kelulusan** memiliki :

- jumlah data 8
- jumlah nilai unik 2
- nilai paling banyak (modus) adalah Tepat
- jumlah nilai paling banyak adalah 4

statistik deskriptif untuk data numerik

```
df[['NPM','ipk_sem_1','ipk_sem_2']].describe()
```

	NPM	ipk_sem_1	ipk_sem_2
count	8.00000	8.000000	8.00000
mean	1904.50000	1.825000	2.52500
std	2.44949	1.208009	0.56252
min	1901.00000	0.000000	2.00000
25%	1902.75000	1.500000	2.00000
50%	1904.50000	2.000000	2.50000
75%	1906.25000	2.625000	3.02500
max	1908.00000	3.100000	3.10000

Deskripsi statistik untuk kolom **NPM** adalah

- jumlah data = 8
- rata-rata = 1904
- standar deviasi = 2.449
- nilai minimal = 1901
- data ke 25% (kuartil 1) = 1902
- data ke 50% (kuartil 2/median) = 1904
- data ke 75% (kuartil 3) = 1906
- nilai maksimum = 1908

Deskripsi statistik untuk kolom **ipk_sem_1** adalah

- jumlah data = 8
- rata-rata = 1.825
- standar deviasi = 1.208
- nilai minimal = 0
- data ke 25% (kuartil 1) = 1.5
- data ke 50% (kuartil 2/median) = 2
- data ke 75% (kuartil 3) = 2.625
- nilai maksimum = 3.025

Deskripsi statistik untuk kolom **ipk_sem_2** adalah

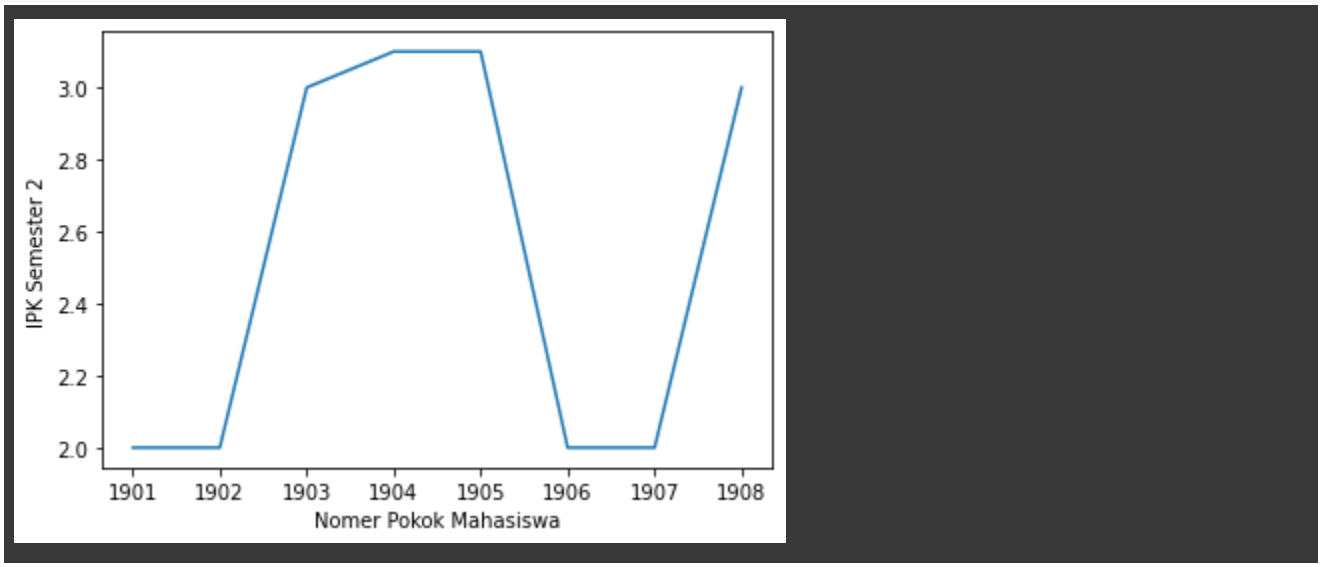
Deskripsi statistik untuk kolom **ipk_sem_2** adalah:

- jumlah data = 8
- rata-rata = 2.525
- standar deviasi = 0.5625
- nilai minimal = 2
- data ke 25% (kuartil 1) = 2
- data ke 50% (kuartil 2/median) = 2.5
- data ke 75% (kuartil 3) = 3.025
- nilai maksimum = 3.1

1.C. Visualisasi

```
import matplotlib.pyplot as plt

plt.plot(dfcopy['NPM'],dfcopy['ipk_sem_2'])
plt.xlabel('Nomer Pokok Mahasiswa')
plt.ylabel('IPK Semester 2')
plt.show()
```



SOAL KE-2

2.A. Mencari Nilai K

Membaca data

```
import pandas as pd

mobil = pd.read_csv("mobil.csv")
```

```
mobil
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmissi
0	0	Maruti Alto K10 LXI CNG	Delhi	2014	40929	CNG	Manu
1	1	Maruti Alto 800 2016-2019 LXI	Coimbatore	2013	54493	Petrol	Manu
2	2	Toyota Innova Crysta Touring Sport 2.4 MT	Mumbai	2017	34000	Diesel	Manu
3	3	Toyota Etios Liva GD	Hyderabad	2012	139000	Diesel	Manu
4	4	Hyundai i20 Magna	Mumbai	2014	29000	Petrol	Manu
...	...	...	...	...	...	...	
1229	1229	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manu
1230	1230	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automa

```
features = mobil[['Year','Kilometers_Driven']]
target = mobil[['Mileage']]
```

Membagi data ke dalam data latih (training data) dan data uji (testing data)

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(features, target, test_size=0.25, rando
```

Membuat classifier (K-Nearest Neighbors)

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neighbors/_classification.py:198: Data
  return self._fit(X, y)
KNeighborsClassifier(n_neighbors=7)
```

Membuat classifier (K-Nearest Neighbors)

```
knn.score(x_test,y_test)
```

```
0.016181229773462782
```

2.B. Menampilkan nilai jarak setiap data yang dihasilkan

```
from sklearn.metrics import DistanceMetric

dist = DistanceMetric.get_metric('euclidean')
dist.pairwise(features)
```

```
array([[    0.        , 13564.00003686,  6929.00064944, ...,
        12929.00015469, 11333.00004412, 31514.        ],
       [13564.00003686,     0.        , 20493.00039038, ...,
        26493.00001887,  2231.        , 17950.00002786],
       [ 6929.00064944, 20493.00039038,     0.        , ...,
         6000.00208333, 18262.00043807, 38443.00011706],
       ...,
       [12929.00015469, 26493.00001887,  6000.00208333, ...,
            0.        , 24262.00002061, 44443.000045  ],
       [11333.00004412,  2231.        , 18262.00043807, ...,
        24262.00002061,     0.        , 20181.00002478],
       [31514.        , 17950.00002786, 38443.00011706, ...,
        44443.000045  , 20181.00002478,     0.        ]])
```

2.C. Nilai akurasi yang dihasilkan

```
knn.score(x_test,y_test)
```

```
0.016181229773462782
```