



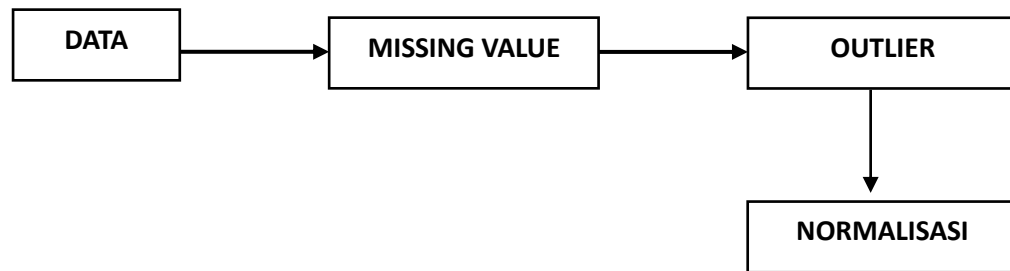
# Pembelajaran Mesin (Praktikum) TI –B4

FAKULTTAS VOKASI  
UNIVERSITAS AIRLANGGA

[152111283042] | [Nela Anjani] | [21 September 2023]

**Latihan :**

Kerjakan semua program diatas

**TUGAS :**

1. Tugas dikerjakan secara mandiri
2. Carilah data bebas
3. Lakukan Transformasi Data berdasarkan Langkah-langkah yang terdapat pada gambar diatas
4. Tugas terdiri dari Laporan, file python, dan data aslinya dan dikumpulkan dengan nama "Tugas Transformasi Data\_NIM. Zip
5. Tugas dikumpulkan paling lambat hari Kamis / 21 September 2023 pukul 20.00 Wib

- PROGRAM

```
import numpy as np
import pandas as pd
from statistics import mean
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
import scipy.stats as stats

# Membaca Data
data = pd.read_excel("kidneydisease.xlsx")
# data = pd.read_csv('kidneydisease.csv')
print(data.head())

# Mengambil beberapa atribut / variabel
data1 = data.loc[:, ['blood_glucose_random', 'blood_urea', 'serum_creatinine']]
print(data1.head())
```

- OUTPUT

```
id  age  blood_pressure  specific_gravity  albumin  sugar  ...  sodium  potassium  hemoglobin  packed_cell_volume  whitebloodcell_count  redbloodcell_count
0  0  48.0             80.0             1.020    1.0    0.0  ...    NaN      NaN      15.4         44             7800             5.2
1  1   7.0             50.0             1.020    4.0    0.0  ...    NaN      NaN      11.3         38             6000             NaN
2  2  62.0             80.0             1.010    2.0    3.0  ...    NaN      NaN      9.6         31             7500             NaN
3  3  48.0             70.0             1.005    4.0    0.0  ...   111.0    2.5      11.2         32             6700             3.9
4  4  51.0             80.0             1.010    2.0    0.0  ...    NaN      NaN      11.6         35             7300             4.6

[5 rows x 15 columns]
blood_glucose_random  blood_urea  serum_creatinine
0             121.0           36.0             1.2
1              NaN           18.0             0.8
2             423.0           53.0             1.8
3             117.0           56.0             3.8
4             106.0           26.0             1.4
```

- DETEKSI MISSING VALUE

```
# MENDETEKSI DATA MISSING
print("Deteksi Missing Value")
print(data1.isna().sum())
```

- OUTPUT

```
4             106.0           26.0             1.4
Deteksi Missing Value
blood_glucose_random    44
blood_urea              19
serum_creatinine        17
dtype: int64
```

- PENANGANAN MISSING VALUE

```
# Penanganan Data Missing Value
## MENGHAPUS DATA MISSING VALUE
print("Penanganan Missing Value")
data_cleaned = data1.dropna()
print("Data tanpa missing value")
print(data_cleaned)
```

- OUTPUT

```

Penanganan Missing Value
Data tanpa missing value
   blood_glucose_random  blood_urea  serum_creatinine
0                    121.0        36.0             1.2
2                    423.0        53.0             1.8
3                    117.0        56.0             3.8
4                    106.0        26.0             1.4
5                     74.0        25.0             1.1
..                   ...         ...             ...
395                   140.0        49.0             0.5
396                    75.0        31.0             1.2
397                   100.0        26.0             0.6
398                   114.0        50.0             1.0
399                   131.0        18.0             1.1

```

- Penanganan Missing Value (mengganti missing value dengan nilai rata-rata (mean

```

# Penanganan Data Missing Value
## MENGGANTI DATA MISSING VALUE DENGAN MEAN
print("Penanganan Missing Value 2")
data1['blood_glucose_random'].fillna(data1['blood_glucose_random'].mean(), inplace=True)
data1['blood_urea'].fillna(data1['blood_urea'].mean(), inplace=True)
data1['serum_creatinine'].fillna(data1['serum_creatinine'].mean(), inplace=True)
print("Missing data pada blood glucose =", data1['blood_glucose_random'].isna().sum())
print("Missing data pada blood urea =", data1['blood_urea'].isna().sum())
print("Missing data pada serum creatinine =", data1['serum_creatinine'].isna().sum())

# Menampilkan nilai mean setelah penanganan missing value
mean_blood_glucose_random = data1['blood_glucose_random'].mean()
mean_blood_urea = data1['blood_urea'].mean()
mean_serum_creatinine = data1['serum_creatinine'].mean()

print("Mean untuk 'blood_glucose_random':", mean_blood_glucose_random)
print("Mean untuk 'blood_urea':", mean_blood_urea)
print("Mean untuk 'serum_creatinine':", mean_serum_creatinine)

```

- OUTPUT

```

Penanganan Missing Value 2
Missing data pada blood glucose = 0
Missing data pada blood urea = 0
Missing data pada serum creatinine = 0
Mean untuk 'blood_glucose_random': 148.0365168539326
Mean untuk 'blood_urea': 57.4257217847769
Mean untuk 'serum_creatinine': 3.072454308093995
Deteksi Outlier

```

- DETEKSI OUTLIER

```

# Mendeteksi Outlier
print("Deteksi Outlier")
outliers = []

def detect_outlier(data):
    threshold = 3
    mean_value = data.mean()
    std_dev = data.std()

    for x in data:
        z_score = (x - mean_value) / std_dev
        if np.abs(z_score) > threshold:
            outliers.append(x)
    return outliers

# Mencetak Outlier
outlier1 = detect_outlier(data1['blood_glucose_random'])
print("Outlier kolom blood_glucose_random : ", outlier1)
print("Banyak outlier blood_glucose_random : ", len(outlier1))
print()

```

- OUTPUT

```

Deteksi Outlier
Outlier kolom blood_glucose_random : [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0]
Banyak outlier blood_glucose_random : 11

Outlier kolom blood_urea : [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0]
Banyak outlier blood_urea : 21
Outlier kolom serum_creatinine : [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0, 24.0, 76.0, 32.0, 48.1]
Banyak outlier serum_creatinine : 25

Outlier blood_glucose_random = [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0, 24.0, 76.0, 32.0, 48.1, 423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0]
Outlier blood_urea = [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0, 24.0, 76.0, 32.0, 48.1, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0]
Outlier serum_creatinine = [423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0, 24.0, 76.0, 32.0, 48.1, 423.0, 410.0, 490.0, 380.0, 425.0, 415.0, 424.0, 447.0, 490.0, 463.0, 424.0, 391.0, 217.0, 219.0, 208.0, 322.0, 235.0, 223.0, 241.0, 215.0, 309.0, 24.0, 76.0, 32.0, 48.1]

```

- PENANGANAN OUTLIER

```

# Penanganan Outlier
variabel = ['blood_glucose_random', 'blood_urea', 'serum_creatinine']
for var in variabel:
    outlier_datapoints = detect_outlier(data1[var])
    print("Outlier ", var, " = ", outlier_datapoints)

```

- DATA HASIL SETELAH PENANGANAN OUTLIER

```

Data setelah penanganan outlier:
   blood_glucose_random  blood_urea  serum_creatinine
0             121.000000         36.0             1.2
1             148.036517         18.0             0.8
2             148.036517         53.0             1.8
3             117.000000         56.0             3.8
4             106.000000         26.0             1.4
..                 ...           ...             ...
395            140.000000         49.0             0.5
396             75.000000         31.0             1.2
397            100.000000         26.0             0.6
398            114.000000         50.0             1.0
399            131.000000         18.0             1.1

```

- NORMALISASI

```
# Normalisasi
#Feature Scaling or Standardization
scaler1 = StandardScaler()
Normalisasi1 = scaler1.fit_transform(data1)
print("Hasil Feature Scaling = " )
print(Normalisasi1)

#Min Max
scaler2 = MinMaxScaler()
Normalisasi2 = scaler2.fit_transform(data1)
print("Hasil Min-Max Scaler = ")
print(Normalisasi2)

#Z-Score
zscores = stats.zscore(data1, axis=1)
print("Hasil Z Score = ")
print(zscores)
```

```
Hasil Feature Scaling =
[[-0.33294189 -0.48462684 -0.46531861]
 [ 0.16121333 -0.98669123 -0.59342361]
 [ 0.16121333 -0.01045492 -0.27316112]
 ...
 [-0.71676573 -0.7635515 -0.65747611]
 [-0.46088317 -0.09413231 -0.52937111]
 [-0.15016864 -0.98669123 -0.49734486]]
Hasil Min-Max Scaler =
[[0.29289941 0.17206983 0.04519774]
 [0.3728891 0.08229426 0.02259887]
 [0.3728891 0.25685786 0.07909605]
 ...
 [0.23076923 0.12219451 0.01129944]
 [0.27218935 0.24189526 0.03389831]
 [0.32248521 0.08229426 0.03954802]]
Hasil Z Score =
      blood_glucose_random  blood_urea  serum_creatinine
0          1.356676      -0.332545      -1.024131
1          1.406121      -0.572222      -0.833899
2          1.327390      -0.241171      -1.086219
3          1.255218      -0.063409      -1.191808
4          1.377983      -0.413544      -0.964439
..          ...          ...          ...
395         1.328723      -0.244992      -1.083731
396         1.295329      -0.156143      -1.139185
397         1.370782      -0.384198      -0.986584
398         1.275198      -0.108068      -1.167130
399         1.404055      -0.555495      -0.848560
```