Import Library

```
import pandas as pd
```

Mean (Rata-Rata)

```
#Membuat Dictionary dari series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
'Lee','Chanchal','Gasper','Naviya','Andres']),
'Umur':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
'Nilai':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])}


# Membuat a DataFrame
df = pd.DataFrame(d)
df
```

	Name	Umur	Nilai
0	Tom	25	4.23
1	James	26	3.24
2	Ricky	25	3.98
3	Vin	23	2.56
4	Steve	30	3.20
5	Smith	29	4.60
6	Jack	23	3.80
7	Lee	34	3.78
8	Chanchal	40	2.98
9	Gasper	30	4.80
10	Naviya	51	4.10
11	Andres	46	3.65

```
# Mencari Mean
print ("Nilai rata-rata usia")
print (df['Umur'].mean())

print ("Rata-rata nilai")
print (df['Nilai'].mean())
```

```
Nilai rata-rata usia
31.833333333333332
Rata-rata nilai
3.7433333333333327
```

Median (Nilai Tengah)

```
# Membuat Dictionary dari series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
'Lee','Chanchal','Gasper','Naviya','Andres']),
'Umur':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
'Nilai':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])}


# Membuat a DataFrame
df = pd.DataFrame(d)

# Mencari Mean
print ("Nilai Median usia")
print (df['Umur'].median())

print ("Median nilai")
print (df['Nilai'].median())
```

```
     Nilai Median usia
     29.5
     Median nilai
     3.79
```

Modus (Nilai yang paling sering muncul)

```
# Kode python untuk demonstrasi fungsi mode()

# Import modul statistik
from statistics import mode

# Import modul fraction as fr
from fractions import Fraction as fr


# tuple dari angka integer positive
data1 = (2, 3, 3, 4, 5, 5, 5, 5, 6, 6, 6, 7)


# tuple dari set angka desimal
data2 = (2.4, 1.3, 1.3, 1.3, 2.4, 4.6)


# tuple dari angka pecahan (fr)
data3 = (fr(1, 2), fr(1, 2), fr(10, 3), fr(2, 3))


# tuple dari angka negatif
data4 = (-1, -2, -2, -2, -7, -7, -9)


# tuple dari strings
data5 = ("red", "blue", "black", "blue", "black", "black", "brown")


# Print mode dari dataset diatas
print("Mode dari data set 1 is % s" % (mode(data1)))
print("Mode dari data set 2 is % s" % (mode(data2)))
print("Mode dari data set 3 is % s" % (mode(data3)))
print("Mode dari data set 4 is % s" % (mode(data4)))
print("Mode dari data set 5 is % s" % (mode(data5)))
```

```
     Mode dari data set 1 is 5
     Mode dari data set 2 is 1.3
     Mode dari data set 3 is 1/2
     Mode dari data set 4 is -2
     Mode dari data set 5 is black
```

Menghitung Standar Deviasi dari dataframe atau series

```
# Import Library
import pandas as pd


# Membuat DataFrame contoh
data = {'A': [1, 2, 3, 4, 5],
       'B': [5, 6, 7, 8, 9]}
df = pd.DataFrame(data)
df
```

	A	B
0	1	5
1	2	6
2	3	7
3	4	8
4	5	9

```
# Menghitung deviasi standar keseluruhan DataFrame
std_dev = df.std()
print("deviasi standar Keseluruhan DataFrame:")
print(std_dev)
```

```
deviasi standar Keseluruhan DataFrame:
A    1.581139
B    1.581139
dtype: float64
```

```
# Menghitung deviasi standar kolom 'A'
std_dev_col_A = df['A'].std()
print("deviasi standar Kolom 'A':", std_dev_col_A)
```

```
deviasi standar Kolom 'A': 1.5811388300841898
```

```
# Menghitung deviasi standar baris 1
std_dev_row_1 = df.loc[1].std()
print("deviasi standar Baris 1:", std_dev_row_1)
```

```
deviasi standar Baris 1: 2.8284271247461903
```

Menghitung varian Dataframe atau Series

```
# Import Library
import pandas as pd


# Membuat DataFrame contoh
data = {'A': [1, 2, 3, 4, 5],
       'B': [5, 6, 7, 8, 9]}
df = pd.DataFrame(data)


# Menghitung variansi keseluruhan DataFrame
varian = df.var()
print("Variansi Keseluruhan DataFrame:")
print(varian)
```

```
Variansi Keseluruhan DataFrame:
A    2.5
B    2.5
dtype: float64
```

```
# Menghitung variansi kolom 'A'
varian_col_A = df['A'].var()
print("Variansi Kolom 'A':", varian_col_A)
```

```
Variansi Kolom 'A': 2.5
```

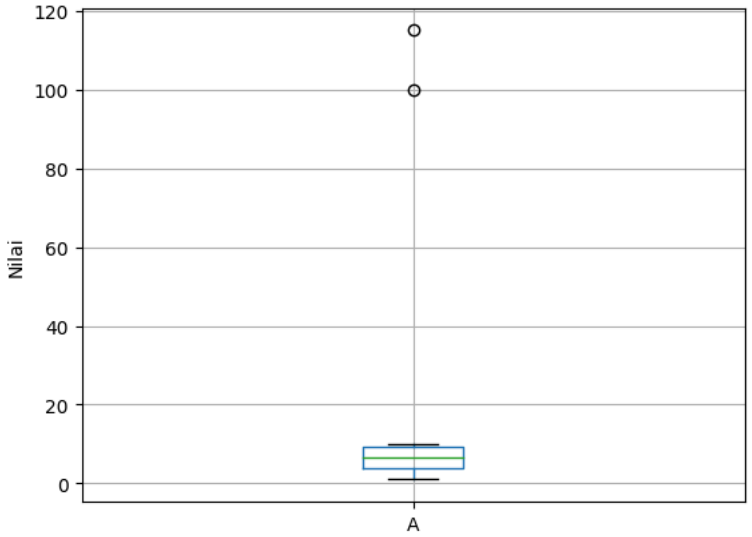
Menghitung Kuratil Dengan Pandas

```
# Import Library
import pandas as pd


# Membuat DataFrame contoh
data = {'A': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 115]}
df = pd.DataFrame(data)


# Menghitung quartil menggunakan method quantile()
Q1 = df['A'].quantile(0.25) # Q1
Q2 = df['A'].quantile(0.50)  # Q2 atau Median
Q3 = df['A'].quantile(0.75) # Q3


# Menghitung IQR
IQR = Q3 - Q1


print("Quartil Pertama (Q1):", Q1)
print("Quartil Kedua atau Median (Q2):", Q2)
print("Quartil Ketiga (Q3):", Q3)
print("Interquartile Range (IQR):", IQR)
```

```
Quartil Pertama (Q1): 3.75
Quartil Kedua atau Median (Q2): 6.5
Quartil Ketiga (Q3): 9.25
Interquartile Range (IQR): 5.5
```

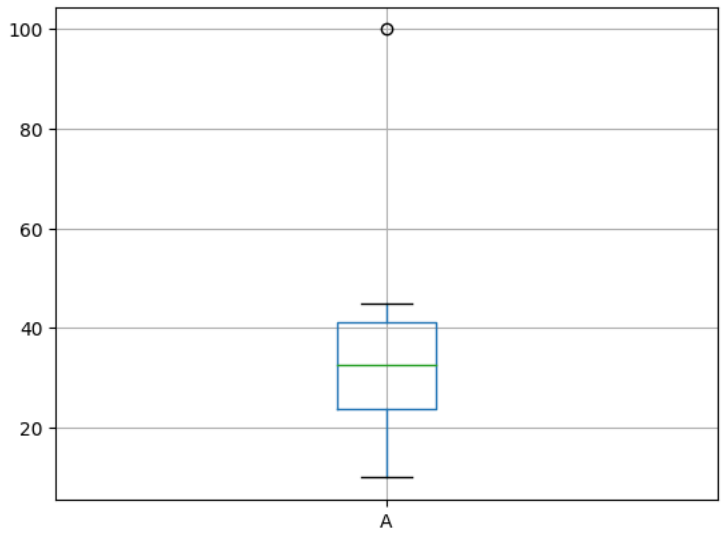
Menampilkan Quratil dengan boxplot

```
# Import Library
import matplotlib.pyplot as plt


# Membuat boxplot
df.boxplot(column='A')


# Menambahkan label sumbu
plt.ylabel('Nilai')


# Menampilkan plot
plt.show()
```



Implementasi identifikasi outlier dengan boxplot

```
# Import library
import pandas as pd
import matplotlib.pyplot as plt


# Membuat DataFrame contoh
data = {'A': [10, 20, 25, 30, 35, 40, 45, 100]}
df = pd.DataFrame(data)


# Visualisasi data dengan boxplot
df.boxplot(column=['A'])
plt.show()
```



Implementasi identifikasi outlier dengan statistik deskriptif

```
# Import library
import pandas as pd

# Membuat DataFrame contoh
data = {'A': [10, 20, 25, 30, 35, 40, 45, 100]}
df = pd.DataFrame(data)


# Menggunakan fungsi describe untuk mendapatkan statistik deskriptif
desc = df['A'].describe()
print(desc)
```

```
count      8.000000
mean      38.125000
std       27.377976
min       10.000000
25%       23.750000
50%       32.500000
75%       41.250000
max      100.000000
Name: A, dtype: float64
```

Penangan Outlier dengan penghapusan

```
# Import Library
import pandas as pd

# Membuat DataFrame contoh
data = {'A': [10, 20, 25, 30, 35, 40, 45, 100]}
df = pd.DataFrame(data)


# Hitung IQR
Q1 = df['A'].quantile(0.25)
Q3 = df['A'].quantile(0.75)
IQR = Q3 - Q1


print("Data awal")
print(df)

# Lower Bound
lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR

# Menghapus outlier

df = df[(df['A'] >= lower_bound) & (df['A'] <= upper_bound)]
print("Data setelah penghapusan outlier")
print(df)
```

```
Data awal
     A
0   10
1   20
2   25
3   30
4   35
5   40
6   45
7  100
Data setelah penghapusan outlier
    A
0  10
1  20
2  25
3  30
4  35
5  40
6  45
```

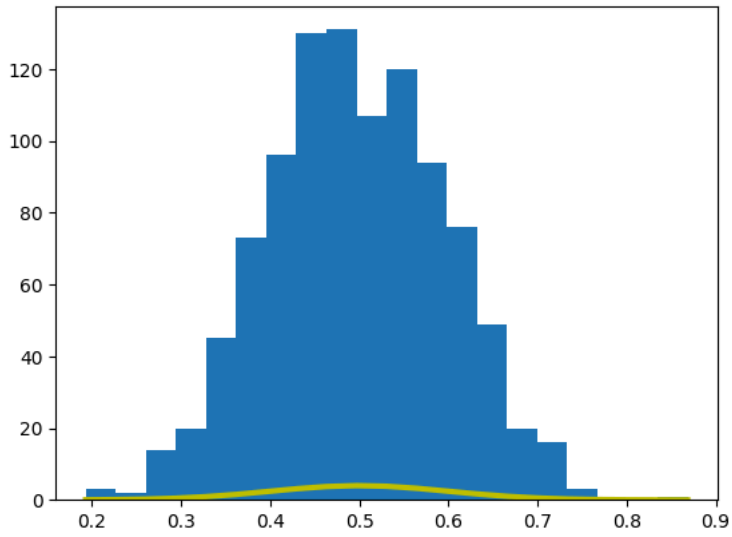
Penanganan Outlier dengan Transformasi Data

```
# Import Library
import pandas as pd
import numpy as np


# Membuat DataFrame contoh
data = {'A': [10, 20, 25, 30, 35, 40, 45, 100]}
df = pd.DataFrame(data)


# Hitung IQR
Q1 = df['A'].quantile(0.25)
Q3 = df['A'].quantile(0.75)
IQR = Q3 - Q1


print("Data awal")
print(df)


# Menghapus outlier
df['A_log'] = np.log(df['A'])
print("Data setelah modifikasi")
print(df)
```

```
Data awal
     A
0   10
1   20
2   25
3   30
4   35
5   40
6   45
7  100
Data setelah modifikasi
     A     A_log
0   10  2.302585
1   20  2.995732
2   25  3.218876
3   30  3.401197
4   35  3.555348
5   40  3.688879
6   45  3.806662
7  100  4.605170
```

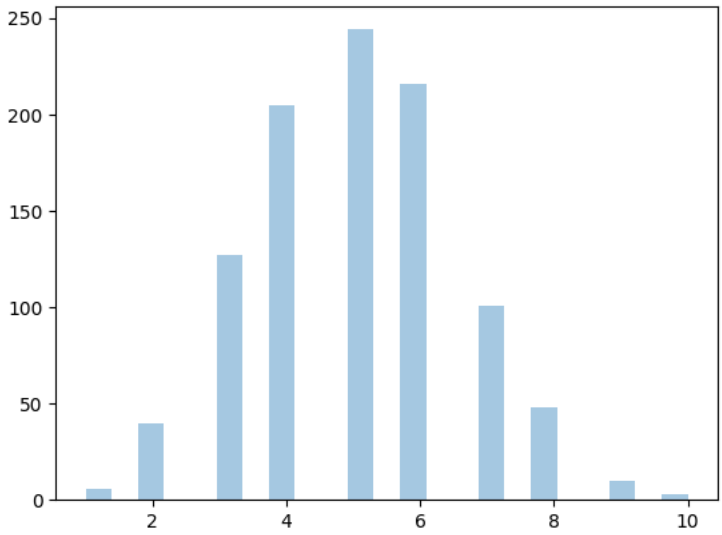
Melakukan Plot distribusi data dengan grafik batang

```
# Import Libray
import matplotlib.pyplot as plt
import numpy as np

# Membuat data dummy
mu, sigma = 0.5, 0.1
s = np.random.normal(mu, sigma, 1000)

# Membuat histogram
count, bins, ignored = plt.hist(s, 20)

# Plot grafik distribusi
plt.plot(bins, 1/(sigma * np.sqrt(2 * np.pi)) * np.exp( - (bins - mu)**2 / (2 * sigma**2) ), linewidth=3, color='y')
plt.show()
```



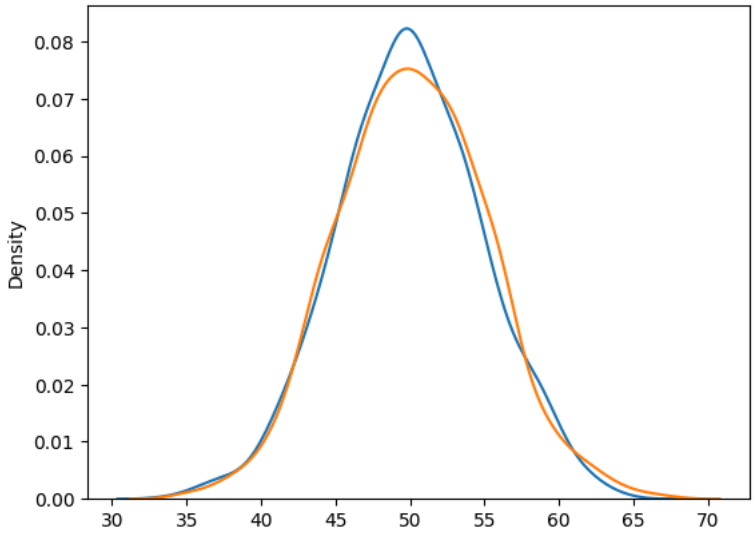
Melakukan Plot distribusi Binomial

```
# Import Library
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

# Abaikan peringatan
warnings.filterwarnings("ignore")

# Membuat grafik distribusi
sns.distplot(random.binomial(n=10, p=0.5, size=1000), hist=True, kde=False)
plt.show()
```



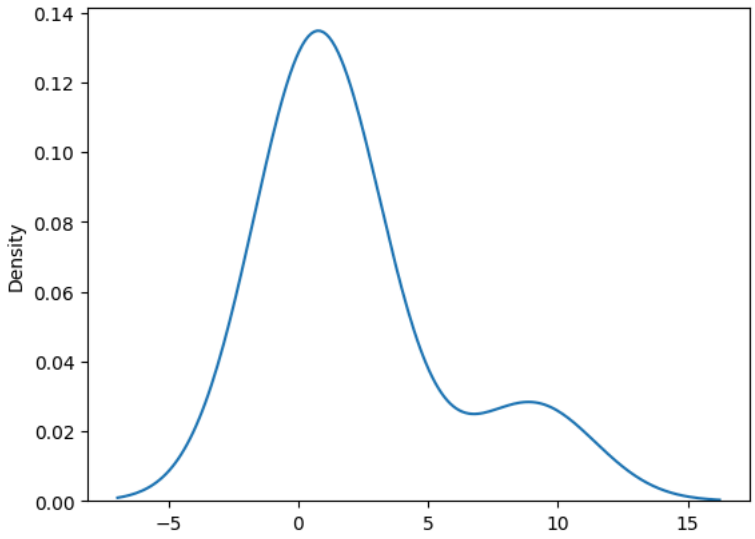
Perbandingan Distribusi Normal dan Binomial

```
# Import Library
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

# Membuat grafik
sns.distplot(random.normal(loc=50, scale=5, size=1000), hist=False, label='normal')
sns.distplot(random.binomial(n=100, p=0.5, size=1000), hist=False, label='binomial')
plt.show()
```



Distribusi Chi Square

```
# Import Library
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

# Membuat grafik
sns.distplot(random.chisquare(df=2, size=(2, 3)), hist=False)
plt.show()
```



Identifikasi Korelasi dengan Heatmap

```
# Import library
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes


# Load dataset diabetes dari scikit-learn - diabetes dataset
diabetes = load_diabetes()

diabetes
```

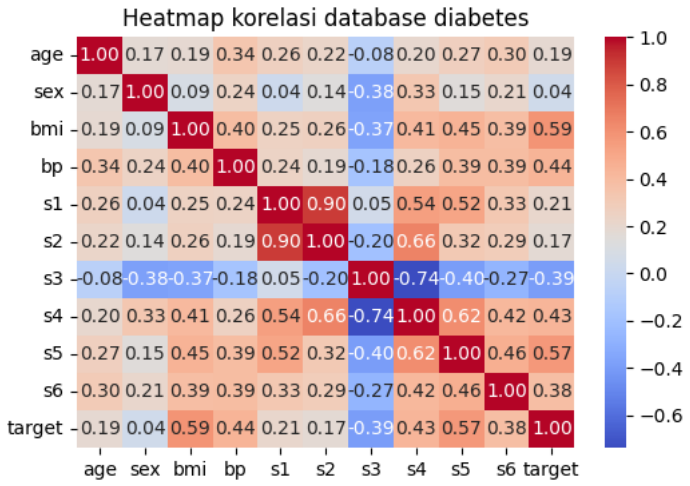

```
{'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
          0.01990749, -0.01764613],
        [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
         -0.06833155, -0.09220405],
        [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
          0.00286131, -0.02593034],
        ...,
        [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
         -0.04688253,  0.01549073],
        [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
          0.04452873, -0.02593034],
        [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
         -0.00422151,  0.00306441]]),
 'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
        69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
        68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
        87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
       259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
       128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
       150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
       200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
        42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
        83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
       104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
       173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
       107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
        60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
       197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
        59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
       237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
       143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
       142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
        77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
        78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
       154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
        71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
       150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
       145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
        94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
        60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
        31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
       114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
       191., 122., 230., 242., 248., 249., 192., 131., 237.,  78., 135.,
       244., 199., 270., 164.,  72.,  96., 306.,  91., 214.,  95., 216.,
       263., 178., 113., 200., 139., 139.,  88., 148.,  88., 243.,  71.,
        77., 109., 272.,  60.,  54., 221.,  90., 311., 281., 182., 321.,
        58., 262., 206., 233., 242., 123., 167.,  63., 197.,  71., 168.,
       140., 217., 121., 235., 245.,  40.,  52., 104., 132.,  88.,  69.,
       219.,  72., 201., 110.,  51., 277.,  63., 118.,  69., 273., 258.,
        43., 198., 242., 232., 175.,  93., 168., 275., 293., 281.,  72.,
       140., 189., 181., 209., 136., 261., 113., 131., 174., 257.,  55.,
        84.,  42., 146., 212., 233.,  91., 111., 152., 120.,  67., 310.,
        94., 183.,  66., 173.,  72.,  49.,  64.,  48., 178., 104., 132.,
       220.,  57.]),
 'frame': None,
 'DESCR': '.. _diabetes_dataset:\n\nDiabetes dataset\n----------------\n\nTen baseline variables, age, sex, body mass index, average
blood\npressure, and six blood serum measurements were obtained for each of n =\n442 diabetes patients, as well as the response of
interest, a\nquantitative measure of disease progression one year after baseline.\n\n**Data Set Characteristics:**\n\n  :Number of
```


```
# Konversi dataset ke dalam DataFrame pandas
df_diabetes = pd.DataFrame(data=diabetes.data, columns=diabetes.feature_names)
df_diabetes['target'] = diabetes.target


df_diabetes
```

	age	sex	bmi	bp	s1	s2	s3	s4	
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.
...	...	...	...	...	...	...	...	...	
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.

442 rows × 11 columns

```
# Hitung korelasi antara setiap fitur dan target
correlation = df_diabetes.corr()


# Buat heatmap untuk menampilkan korelasi
plt.figure(figsize=(6, 4))
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap korelasi database diabetes')
plt.show()
```



Penanganan Value yang Hilang

```
# Import Library
import pandas as pd

# Import Dataset
titanic_data = pd.read_csv('https://ngodingdata.com/wp-content/uploads/2020/05/titanic.csv')
titanic_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	[illegible]	female	38.0	1	0	PC 17599	71.2833	C85	C

Cek Data Hilang

```
titanic_data.isna()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	False	False	False	False	False	False	False	False	False	False	True
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True
887	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True
889	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True

891 rows × 12 columns

```
titanic_data.isna().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Handling missing value pada kolom age dengan nilai rata-rata

```
# Langkah 1
rata_umur = titanic_data['Age'].mean()
# Langkah 2
titanic_data['Age'] = titanic_data['Age'].fillna(rata_umur)
```

Cek kembali apakah masih terdapat missing value

```
# Langkah 3
titanic_data['Age'].isna().sum()
```