

```
from sklearn import tree
```

```
# Database: Gerbang Logika AND  
# X = Data, y = Target
```

```
x = [[0, 0],  
      [0, 1],  
      [1, 0],  
      [1, 1]]  
y = [0, 0, 0, 1]
```

```
# Training and Classify  
clf = tree.DecisionTreeClassifier()  
clf = clf.fit(x,y)
```

```
# Prediction  
print("Logika AND Metode Decision Tree")  
print("Logika = Prediksi")  
print("0 0 = ", clf.predict([[0, 0]]))  
print("0 1 = ", clf.predict([[0, 1]]))  
print("1 0 = ", clf.predict([[1, 0]]))  
print("1 1 = ", clf.predict([[1, 1]]))
```

```
⇒ Logika AND Metode Decision Tree  
Logika = Prediksi  
0 0 = [0]  
0 1 = [0]  
1 0 = [0]  
1 1 = [1]
```

```
from google.colab import drive  
import pandas as pd  
import numpy as np  
from sklearn.tree import DecisionTreeRegressor  
import matplotlib.pyplot as plt
```

```
# Mount Google Drive
drive.mount('/content/drive')
# Path ke file di Google Drive
FileDB = '/content/drive/My Drive/MODUL 9 PRAKFIKSI/Sinus.txt' # Sesuaikan
Database = pd.read_csv(FileDB, sep=";", header=0)
# Lihat data
print("-----")
print(Database)
```

➡ Drive already mounted at /content/drive; to attempt to forcibly remount

```
-----
      Feature    Target
0         1  0.841471
1         2  0.989297
2         3  0.141120
3         4 -0.756802
4         5 -0.958924
5         6 -0.279415
6         7  0.656987
7         8  0.989358
8         9  0.412118
9        10 -0.544021
10       11 -0.999990
11       12 -0.536573
12       13  0.420167
13       14  0.990607
14       15  0.650288
15       16 -0.287903
16       17 -0.961397
17       18 -0.750987
18       19  0.149877
19       20  0.912945
```

```
# x data, y target
x = Database[['Feature']] # replace with your actual column names
y = Database.Target
```

```
reg = DecisionTreeRegressor(random_state=1)
reg = reg.fit(x, y)
```

```
# Display predicted data
xx = np.arange(1, 21, 1)
n = len(xx)
print("xx(i) Decision Tree")
for i in range(n):
    y_dct = reg.predict([[xx[i]]])
    print('{:.2f}'.format(xx[i]), y_dct)

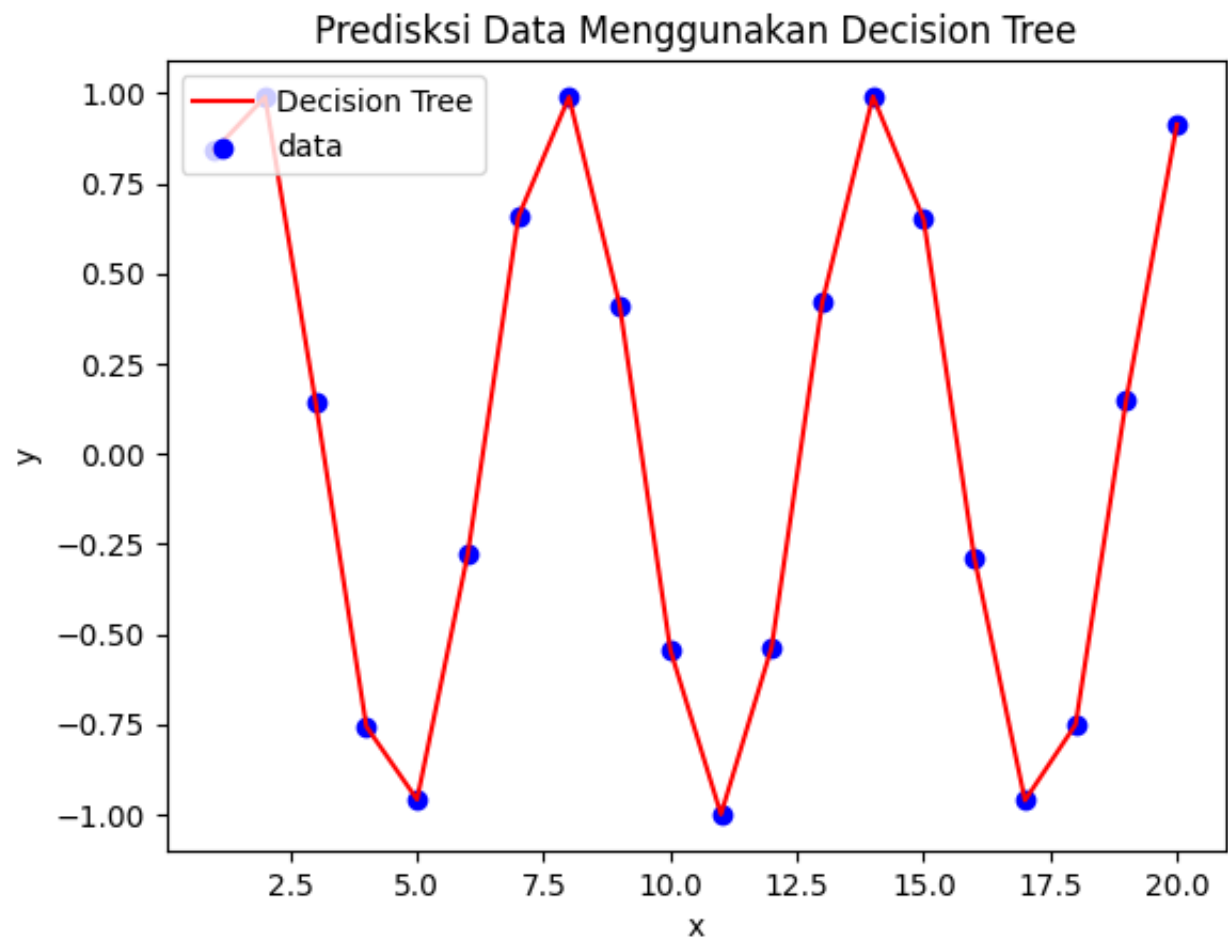
# Plot the predicted data
y_dct2 = reg.predict(x)
plt.figure()
plt.plot(x, y_dct2, color='red')
plt.scatter(x, y, color='blue')
plt.title('Prediksi Data Menggunakan Decision Tree')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(['Decision Tree', 'data'], loc=2)
```

```
plt.show()
```

[illegible]

```
warnings.warn(  
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarnin  
warnings.warn(  

```



```

# Mengimport library yang akan digunakan
from sklearn import tree

# Masukkan dataset berupa data (x) dan target (y)
# Dataset yang diberikan
x = [[0, 0, 0],
      [0, 5, 0],
      [0, 0, 5],
      [0, 5, 5],
      [5, 5, 0],
      [5, 0, 5],
      [5, 5, 5],
      [10, 5, 5],
      [5, 10, 5],
      [10, 10, 10]]

y = [0, 0, 0, 5, 5, 5, 10, 10, 5, 0]

# Membuat model dan melatihnya
clf = tree.DecisionTreeClassifier()
clf = clf.fit(x, y)

# Memprediksi data baru
print("Logika = Prediksi")
print("10 10 5 = ", clf.predict([[10, 10, 5]]))
print("5 10 2 = ", clf.predict([[5, 10, 2]]))
print("2 0 10 = ", clf.predict([[2, 0, 10]]))
print("5 0 2 = ", clf.predict([[5, 0, 2]]))
print("0 0 2 = ", clf.predict([[0, 0, 2]]))
print("2 10 2 = ", clf.predict([[2, 10, 2]]))
print("1 12 5 = ", clf.predict([[1, 12, 5]]))
print("2 2 6 = ", clf.predict([[2, 2, 6]]))
print("10 5 7 = ", clf.predict([[10, 5, 7]]))

```

```

⇒ Logika = Prediksi
10 10 5 = [10]
5 10 2 = [5]
2 0 10 = [0]
5 0 2 = [5]
0 0 2 = [0]
2 10 2 = [0]
1 12 5 = [5]
2 2 6 = [0]
10 5 7 = [10]

```

```

# Mengimport library yang digunakan
from google.colab import drive
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor
import matplotlib.pyplot as plt

```

```
import pandas as pd
```

```
# Memuat data pada google drive
# Mount Google Drive
drive.mount('/content/drive')

# Path ke file di Google Drive
FileDB = '/content/drive/My Drive/MODUL 9 PRAKFIKOM/Cosinus.txt' # Sesuaikan
# Dataset yang diberikan dalam format .txt
# Pastikan file Cosinus.txt sudah di-upload ke Google Drive dengan format yang sesuai
Database = pd.read_csv(FileDB, sep=",", header=0)

# Lihat data
print("-----")
print(Database)

# Menentukan fitur dan target
# x data, y target
x = Database[['Feature']] # Ganti dengan nama kolom yang sesuai
y = Database['Target']

# Membuat dan melatih model
reg = DecisionTreeRegressor(random_state=1)
reg = reg.fit(x, y)

# Memprediksi data
# Display predicted data
xx = np.arange(1, 21, 1)
n = len(xx)
print("xx(i) Decision Tree")
for i in range(n):
    y_dct = reg.predict([[xx[i]]])
    print('{:.2f}'.format(xx[i]), y_dct)

# Plot the predicted data
y_dct2 = reg.predict(x)
plt.figure()
plt.plot(x, y_dct2, color='red')
plt.scatter(x, y, color='blue')
plt.title('Prediksi Data Menggunakan Decision Tree')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(['Decision Tree', 'data'], loc=2)
plt.show()
```

➡ Drive already mounted at /content/drive; to attempt to forcibly remount

```
-----
      Feature      Target
0          1  0.540302
1          2 -0.416147
2          3 -0.989992
3          4 -0.653644
4          5  0.283662
5          6  0.960170
6          7  0.753902
7          8 -0.145500
8          9 -0.911130
9         10 -0.839072
10         11  0.004426
```

```

--      --      -----
11      12      0.843854
12      13      0.907447
13      14      0.136737
14      15      -0.759688
15      16      -0.957659
16      17      -0.275163
17      18      0.660317
18      19      0.988705
19      20      0.408082
xx(i) Decision Tree
1.00 [0.5403023]
2.00 [-0.41614684]
3.00 [-0.9899925]
4.00 [-0.65364362]
5.00 [0.28366219]
6.00 [0.96017029]
7.00 [0.75390225]
8.00 [-0.14550003]
9.00 [-0.91113026]
10.00 [-0.83907153]
11.00 [0.0044257]
12.00 [0.84385396]
13.00 [0.90744678]
14.00 [0.13673722]
15.00 [-0.75968791]
16.00 [-0.95765948]
17.00 [-0.27516334]
18.00 [0.66031671]
19.00 [0.98870462]
20.00 [0.40808206]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(

```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning:
  warnings.warn(
```

