

Praktikum Fisika Komputasi

Modul 9

Septian Tri Laksono

1227030032

```
from sklearn import tree
```

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

```
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]
```

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

```
# Prediction  
print("Logika AND Metode Decision Tree")  
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 AND Metode Decision Tree  
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]
```

```
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/format.txt' # Sesuaikan path file
Database = pd.read_csv(FileDB, sep=";", header=0)

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

Mounted at /content/drive
-----
   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
```

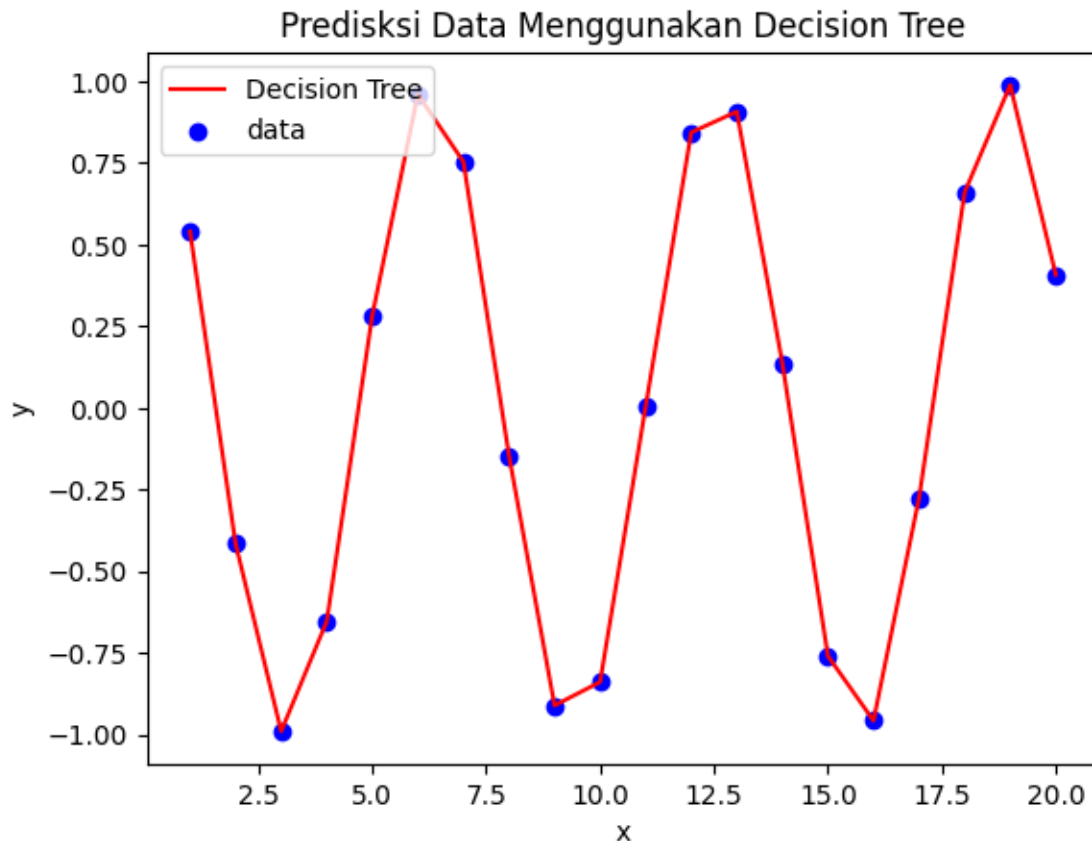
```
# x data, y target
x = Database[['Feature']] # replace with your actual column names
y = Database.Target
```

```
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()
```

```
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]
```



Pertama kita membuat kode program klasifikasi dengan memasukkan dataset berupa data (x) dan target (Y) dengan menggunakan logika gerbang AND, di mana input terdiri dari pasangan biner. Dibuat model dan melatihnya agar model dapat memprediksi sebuah data. Lalu model berhasil mengklasifikasikan kombinasi input sesuai dengan aturan logika AND. Hasil yang diperoleh menunjukkan bahwa output hanya akan bernilai 1 jika kedua input bernilai 1.

Kemudian dibuat kode program untuk memprediksi data, yang mana data yang ingin kita prediksi di simpan di drive yang kemudian di impor sesuai dengan path file, isi dari data tersebut meliputi kolom fitur dan target, di mana fitur adalah nilai x dan target adalah y. Kemudian data di prediksi dan model Decision Tree berhasil memprediksi nilai target berdasarkan fitur yang diberikan. Hasilnya mencerminkan nilai data yang tepat untuk setiap input, ditunjukkan dengan grafik yang memperlihatkan hubungan antara fitur dan target.

Metode Decision Tree dapat digunakan dalam berbagai aplikasi di dunia perkuliahan, khususnya di jurusan Fisika contohnya dapat digunakan untuk mengklasifikasikan data berdasarkan parameter yang diamati, memodelkan fenomena fisika kompleks, seperti gerakan proyektil atau perubahan energi dalam sistem, dan untuk menganalisis data yang dihasilkan oleh sensor seperti suhu, tekanan, atau radiasi