

TUGAS 9 PRAKTIKUM FISIKA KOMPUTASI

“DECISION TREE METHOD”

A. Kode Program Klasifikasi Dataset

```
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]]))
```

dengan hasil prediksinya:

```
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]
```

B. Kode Program Prediksi Data Cosinus

```
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/Cosinus.txt' #Sesuaikan dengan
nama file
Database = pd.read_csv(FileDB, sep=",", header=0)

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

Dengan data cosinus:

```
-----
   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']]
y = Database.Target
reg = DecisionTreeRegressor(random_state=1)
reg = reg.fit(x,y)
#Menampilkan prediksi data
```

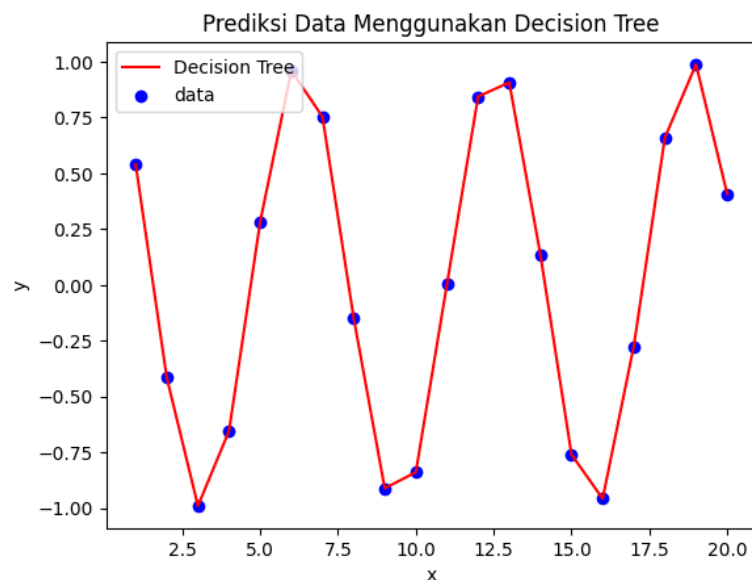
```

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 dari prediksi 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()

```

dengan grafik yang dihasilkan:



C. Penjelasan Hasil

1. Metode Klasifikasi, metode ini bertujuan untuk membuat dan menguji sebuah model pembelajaran mesin sederhana yang dapat mensimulasikan operasi logika AND. Metode ini terdapat 3 tahapan utama, yaitu pembuatan model, pelatihan model, serta pengujian model. Ketika semua input bernilai besar (misal 10, 10, 5), model memprediksi hasil 10 (yang dianggap sebagai true). Hal ini sesuai dengan operasi logika AND, di mana jika semua input bernilai true, maka hasilnya juga true. Ketika ada satu atau lebih input bernilai 0, model

memprediksi hasil 0 (false). Hal ini juga sesuai dengan operasi logika AND, di mana jika ada satu saja input yang bernilai false, maka hasilnya juga false.

2. Metode Prediksi Data, metode ini bertujuan untuk memvisualisasikan hasil prediksi melalui grafik dibandingkan dengan data asli. Dalam metode ini digunakan data cosinus yang telah dibuat untuk selanjutnya akan ditentukan prediksi data baru. Berdasarkan grafik yang didapatkan, sumbu x menunjukkan nilai fitur input (1 sampai 20), sedangkan sumbu y menunjukkan nilai prediksi (garis merah) dan nilai sebenarnya (titik biru) yang didapatkan dari model Decision Tree.

D. Aplikasi Decision Tree Method dalam Perkuliahan Fisika

1. Analisis Data Eksperimen

Decision Tree dapat digunakan untuk mengidentifikasi pola-pola yang kompleks dalam data eksperimen. Misalnya, untuk mengklasifikasikan jenis partikel berdasarkan sifat-sifatnya (massa, muatan, spin), atau untuk mengidentifikasi fase suatu zat berdasarkan data tekanan dan suhu.

Dengan menganalisis data eksperimen sebelumnya, metode ini dapat membantu merancang eksperimen baru yang lebih efisien dengan mengidentifikasi variabel-variabel yang paling berpengaruh terhadap hasil eksperimen.

2. Simulasi Fisika

Decision Tree dapat digunakan untuk membangun model simulasi sederhana dari sistem fisika. Misalnya, untuk mensimulasikan gerak partikel di bawah pengaruh gaya tertentu. Dengan mengubah parameter dalam model pohon keputusan, kita dapat menganalisis bagaimana perubahan parameter tersebut mempengaruhi hasil simulasi.

3. Prediksi Gerak Planet

Decision Tree dapat digunakan untuk memprediksi gerak serta posisi planet dengan baik. Dengan menggunakan data historis tentang posisi planet, pohon keputusan dapat digunakan untuk memprediksi posisi planet di masa depan.