Nama: Ririn Anastasya

NIM: 1227030030

```
[31] from sklearn import tree
```

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

```
[33] x = [[0, 0, 0],
          [0, 5, 0],
          [0, 0, 0],
          [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]
```

```
[34] # 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 = [5]
10 5 7 = [10]
```

Kode program ini menggunakan *Decision Tree* untuk memprediksi hasil dari logika AND berdasarkan data yang digunakan. Kode program dapat menganalisis pola input dan membandingkannya dengan data untuk memberikan prediksi. Misalnya, untuk input [10, 10, 5] kode program memprediksi output [10] karena pola ini mirip dengan data [10, 5, 5] yang juga menghasilkan output [10]. Begitu pula, input [5, 10, 2] menghasilkan output [5] karena sama dengan pola [5, 10, 5]. Namun, jika input mengandung nilai nol seperti pada [2, 0, 10] dan [0, 0, 2], kode program cenderung memprediksi output [0] karena pola tersebut mengarah pada kombinasi yang lebih rendah. Untuk input lain seperti [5, 0, 2], [1, 12, 5], dan [2, 2, 6] kode program menghasilkan output [5] karena pola ini mendekati kombinasi yang menghasilkan output [5] seperti [5, 0, 5]. Sementara itu, input [10, 5, 7] juga menghasilkan output [10] sesuai dengan pola [10, 5, 5].

```
[19] 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/decision tree/dap.txt' # Sesuaikan path file
Database = pd.read_csv(FileDB, sep=",", header=0)

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
--------------------------
    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
```

Hasil dari kode program ini menunjukkan data yang diambil dari file dap.txt yang terdiri dari dua kolom, yaitu Feature dan Target. Kolom Feature berisi angka bulat dari 1 hingga 20 yang berfungsi sebagai input atau fitur dalam dataset, sedangkan kolom Target merupakan hasil perhitungan fungsi cosinus dari nilai-nilai di kolom Feature. Nilai-nilai di kolom Target berkisar antara -1 hingga 1 sehingga mencerminkan karakteristik fungsi cosinus yang bersifat periodik.

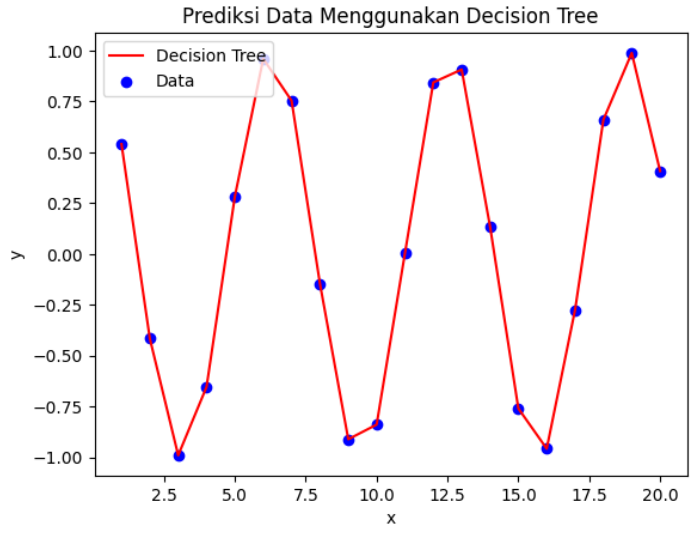
```
[28] # x data, y target
     x = Database[['Feature']] # replace with your actual column names
     y = Database.Target
```

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

```
[30] # 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
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
```



Hasil dari kode program ini menunjukkan bagaimana *Decision Tree* memprediksi hubungan antara Feature dan Target yang berasal dari dataset. Kode program menggunakan data dari kolom Feature dan hasil fungsi kosinusnya. Dalam grafik yang ditampilkan, garis merah mewakili prediksi model sementara titik biru menunjukkan data aktual. *Decision tree* bekerja dengan membagi nilai input menjadi beberapa segmen sehingga menghasilkan prediksi yang berbentuk langkah-langkah. Meskipun prediksi model mengikuti pola data kosinus, bentuknya tidak halus dan terlihat terputus-

putus. Nilai yang dicetak sebagai *xx(i) Decision Tree* menunjukkan nilai prediksi untuk setiap fitur dari 1 hingga 20 yang sejalan dengan pola target tetapi tidak selalu identik.

Metode *Decision Tree* memiliki beberapa kegunaan di perkuliahan terutama di jurusan fisika.

1. Klasifikasi Material: *Decision tree* bisa digunakan untuk mengelompokkan berbagai jenis material berdasarkan sifat-sifat fisiknya, seperti seberapa baik mereka menghantarkan listrik atau seberapa kuat mereka.
2. Analisis Data Eksperimen: Dalam penelitian fisika, *decision tree* dapat membantu menganalisis data yang dihasilkan dari eksperimen, seperti pengukuran suhu atau tekanan.
3. Prediksi Hasil Simulasi: *Decision tree* juga berguna untuk memprediksi hasil dari simulasi fisika, seperti simulasi aliran fluida atau interaksi antara partikel.