Berikut adalah penjelasan dari kode program yang telah di Buat:

1. **Import Library:**

**from sklearn.tree import DecisionTreeRegressor**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

Kode ini mengimpor library yang dibutuhkan untuk program. Memanggil DecisionTreeRegressor dari scikit-learn, numpy untuk operasi numerik, pandas untuk manipulasi data, dan matplotlib untuk plotting data.

1. **Load Database:**

**FileDB = 'cos.txt'**

**Database = pd.read\_csv(FileDB, sep=",", header=0)**

**print("---------------------")**

**print(Database)**

Program membaca data dari file 'cos.txt' menggunakan pandas dan menampilkannya pada python. Data ini kemudian disimpan dalam bentuk DataFrame.

1. **Prepare Data (Features and Target):**

**x = Database[['Feature']]**

**y = Database['Target']**

Memisahkan data menjadi dua bagian: x sebagai fitur (features) dan y sebagai target.

1. **Initialize and Train DecisionTreeRegressor:**

**reg = DecisionTreeRegressor(random\_state=0)**

**reg.fit(x, y)**

Inisialisasi model DecisionTreeRegressor dengan satu tetangga terdekat dan melatihnya menggunakan data x dan y.

1. **Generate and Display Predictions:**

**xx = np.arange(1, 21, 1)**

**n = len(xx)**

**print("----------------------------")**

**print("-Decision Tree Predictions-")**

**print("xx(i) Decision Tree")**

Membuat array xx dari 1 hingga 20 dan melakukan prediksi untuk setiap nilai menggunakan model DecissionTreeRegressor. Hasil prediksi ditampilkan dipython.

1. **Plot Predictions:**

**y\_dct2 = reg.predict(x)**

**plt.figure()**

**plt.plot(x, y\_dct2, color='red')**

**plt.scatter(x, y, color='blue')**

**plt.title('Prediction Using Decision Tree')**

**plt.xlabel('x')**

**plt.ylabel('y')**

**plt.legend(['Decision Tree', 'Data'], loc='upper right')**

**plt.show()**

Membuat plot dari data aktual dan hasil prediksi menggunakan matplotlib.