Berikut adalah penjelasan dari kode program yang telah Kita Buat:

1. **Import Library:**

**python**

**Copy code**

**from sklearn.neighbors import KNeighborsRegressor**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

Kode ini mengimpor library yang dibutuhkan untuk program, termasuk model K-Nearest Neighbors Regressor dari scikit-learn, numpy untuk operasi numerik, pandas untuk manipulasi data, dan matplotlib untuk plotting.

1. **Load Database:**

**python**

**Copy code**

**FileDB = 'sinus.txt'**

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

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

**print(Database)**

Program membaca data dari file 'sinus.txt' menggunakan pandas dan menampilkannya di console. Data ini kemudian disimpan dalam bentuk DataFrame.

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

**python**

**Copy code**

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

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

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

1. **Initialize and Train K-Nearest Neighbors Regressor:**

**python**

**Copy code**

**reg = KNeighborsRegressor(n\_neighbors=1)**

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

Inisialisasi model K-Nearest Neighbors Regressor dengan satu tetangga terdekat dan melatihnya menggunakan data x dan y.

1. **Generate and Display Predictions:**

**python**

**Copy code**

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

**n = len(xx)**

**print("xx(i) K-NN")**

**for i in range(n):**

**y\_neighbor = reg.predict([[xx[i]]])**

**print('{:.2f}'.format(xx[i]), y\_neighbor[0])**

Membuat array xx dari 1 hingga 20 dan melakukan prediksi untuk setiap nilai menggunakan model K-Nearest Neighbors. Hasil prediksi ditampilkan di console.

1. **Plot Predictions:**

**python**

**Copy code**

**y\_neighbor2 = reg.predict(x)**

**plt.figure()**

**plt.plot(x, y\_neighbor2, color='red')**

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

**plt.title('Prediksi Data Menggunakan K-Nearest Neighbors')**

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

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

**plt.legend(['nearest neighbors', 'data'], loc=2)**

**plt.show()**

Membuat plot dari data aktual (biru) dan hasil prediksi (merah) menggunakan matplotlib.