Fariid Abyasta Suheri

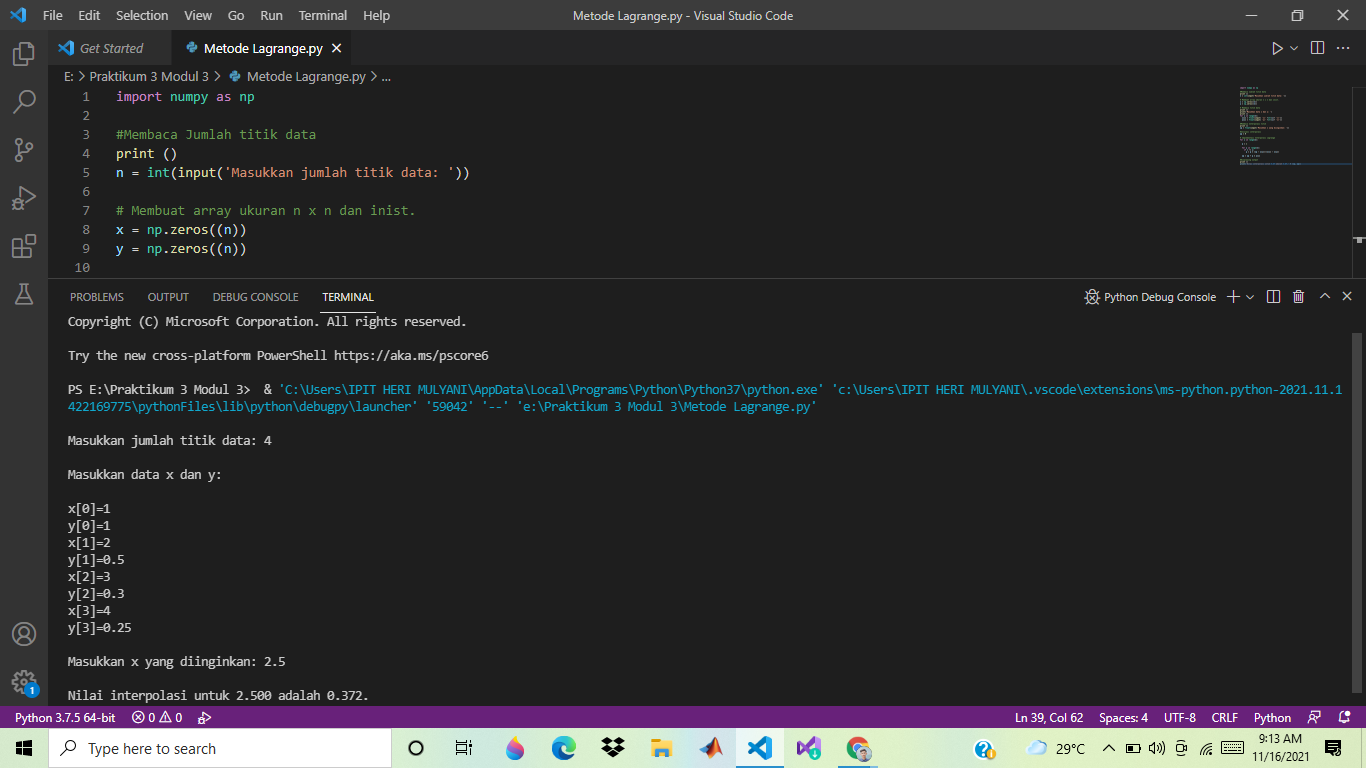
202010225333

TF3A6

PRAKTIKUM 3 METODE NUMERIK

PRAKTIKUM METNUM 3

1. Metode Lagrange



# Interpolasi Lagrange

import numpy as np

#Membaca Jumlah titik data

n = int(input('Masukkan jumlah titik data: '))

# Membuat array ukuran n x n dan inist.

x = np.zeros((n))

y = np.zeros((n))

# Membaca titik data

print('Masukkan data x dan y: ')

for i in range(n):

x[i] = float(input( 'x[' +str(i)+ ']='))

y[i] = float(input( 'y[' +str(i)+ ']='))

#Membaca Interpolasi titik

xp = float(input('Masukkan x yang diinginkan: '))

#Inisiasi interpolasi

yp = 0

# Implementasi Interpolasi Lagrange

for i in range(n):

p = 1

for j in range(n):

if i != j:

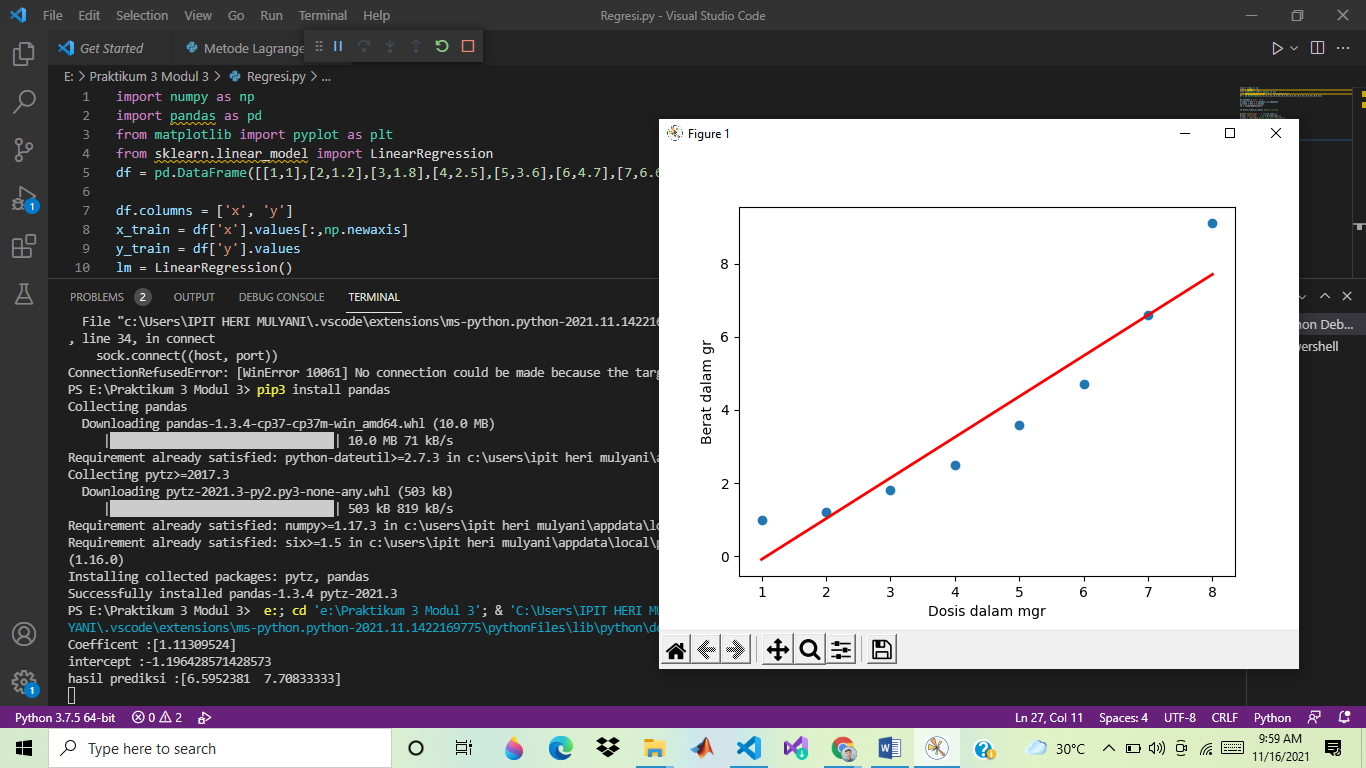
p = p \* (xp - x[j])/(x[i] - x[j])

yp = yp + p \* y[i]

#Displaying output

print('Nilai interpolasi untuk %.3f adalah %.3f.' % (xp, yp)

1. Metode Linear



import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

from sklearn.linear\_model import LinearRegression

df = pd.DataFrame([[1,1],[2,1.2],[3,1.8],[4,2.5],[5,3.6],[6,4.7],[7,6.6],[8,9.1]])

df.columns = ['x', 'y']

x\_train = df['x'].values[:,np.newaxis]

y\_train = df['y'].values

lm = LinearRegression()

lm.fit(x\_train,y\_train) #fase training

print('Coefficent :' + str(lm.coef\_))

print('intercept :' + str(lm.intercept\_))

x\_test = [[7],[8]] #data yang akan diprediksi

p = lm .predict(x\_test) #fase prediksi

print('hasil prediksi :' + str(p)) #hasil prediksi

#prepare plot

pb = lm.predict(x\_train)

dfc = pd.DataFrame({'x': df['x'],'y':pb})

plt.scatter(df['x'],df['y'])

plt.plot(dfc['x'],dfc['y'],color='red',linewidth=2)

plt.xlabel('Dosis dalam mgr')

plt.ylabel('Berat dalam gr')

plt.show()