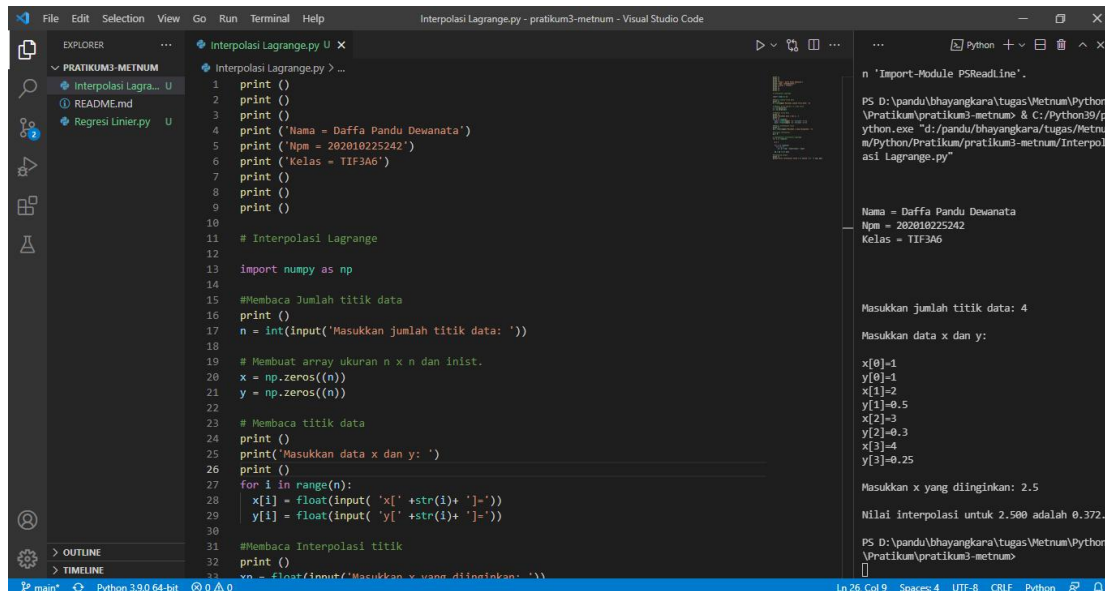


Pratikum 3

Metode Lagrange



```
1 print ()
2 print ()
3 print ()
4 print ('Nama = Daffa Pandu Dewanata')
5 print ('Npm = 202010225242')
6 print ('Kelas = TIF3A6')
7 print ()
8 print ()
9 print ()
10
11 # Interpolasi Lagrange
12
13 import numpy as np
14
15 #Membaca Jumlah titik data
16 print ()
17 n = int(input('Masukkan jumlah titik data: '))
18
19 # Membuat array ukuran n x n dan inist.
20 x = np.zeros((n))
21 y = np.zeros((n))
22
23 # Membaca titik data
24 print ()
25 print('Masukkan data x dan y: ')
26 print ()
27 for i in range(n):
28     x[i] = float(input('x[' + str(i) + ']=-'))
29     y[i] = float(input('y[' + str(i) + ']=-'))
30
31 #Membaca Interpolasi titik
32 print ()
33 n = float(input('Masukkan x yang diinginkan: '))
```

Output Console:

```
n 'Import-Module PSReadLine'.
PS D:\pandu\bhayangkara\tugas\Metnum\Python\Pratikum3-metnum> & C:/Python39/p
ython.exe "d:/pandu/bhayangkara/tugas/Metnu
m/Python/Pratikum/pratikum3-metnum/Interpol
asi Lagrange.py"

Nama = Daffa Pandu Dewanata
Npm = 202010225242
Kelas = TIF3A6

Masukkan jumlah titik data: 4

Masukkan data x dan y:

x[0]=-1
y[0]=-1
x[1]=2
y[1]=0.5
x[2]=3
y[2]=-0.3
x[3]=4
y[3]=0.25

Masukkan x yang diinginkan: 2.5

Nilai Interpolasi untuk 2.500 adalah 0.372.

PS D:\pandu\bhayangkara\tugas\Metnum\Python\Pratikum3-metnum>
```

Script Lagrange

```
print ()
print ()
print ()
print ('Nama = Daffa Pandu Dewanata')
print ('Npm = 202010225242')
print ('Kelas = TIF3A6')
print ()
print ()
print ()

# Interpolasi Lagrange

import numpy as np

#Membaca Jumlah titik data
print ()
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 ()
print('Masukkan data x dan y: ')
print ()
```

```
for i in range(n):
    x[i] = float(input( 'x[' +str(i)+ ']='))
    y[i] = float(input( 'y[' +str(i)+ ']='))

#Membaca Interpolasi titik
print ()
xp = float(input('Masukkan x yang diinginkan: '))

#Inisiasi interpolasi
yp = 0

# Implementasi Interpolasi Lagrange
for i in range(n):

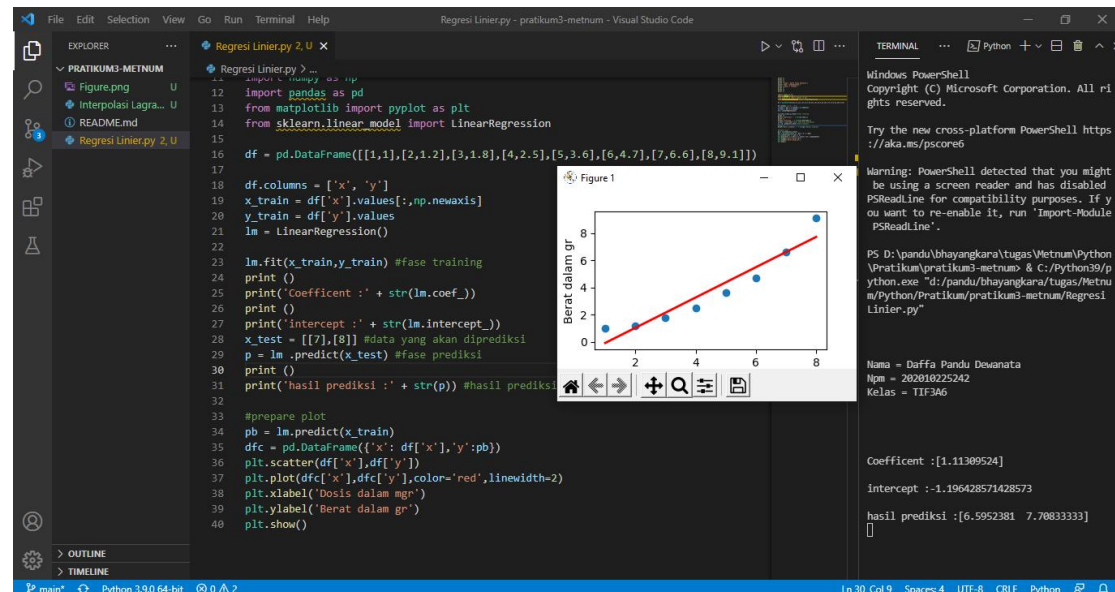
    p = 1

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

Metode Regresi Linear



Script Regresi Linear

```

print ()
print ()
print ()
print ('Nama = Daffa Pandu Dewanata')
print ('Npm = 202010225242')
print ('Kelas = TIF3A6')
print ()
print ()
print ()
  
```

```

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 ()  
print('Coefficient :' + str(lm.coef_))  
print ()  
print('intercept :' + str(lm.intercept_))  
x_test = [[7],[8]] #data yang akan diprediksi  
p = lm .predict(x_test) #fase prediksi  
print ()  
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()
```