Nama:Haris Himawan

Npm:202010225264

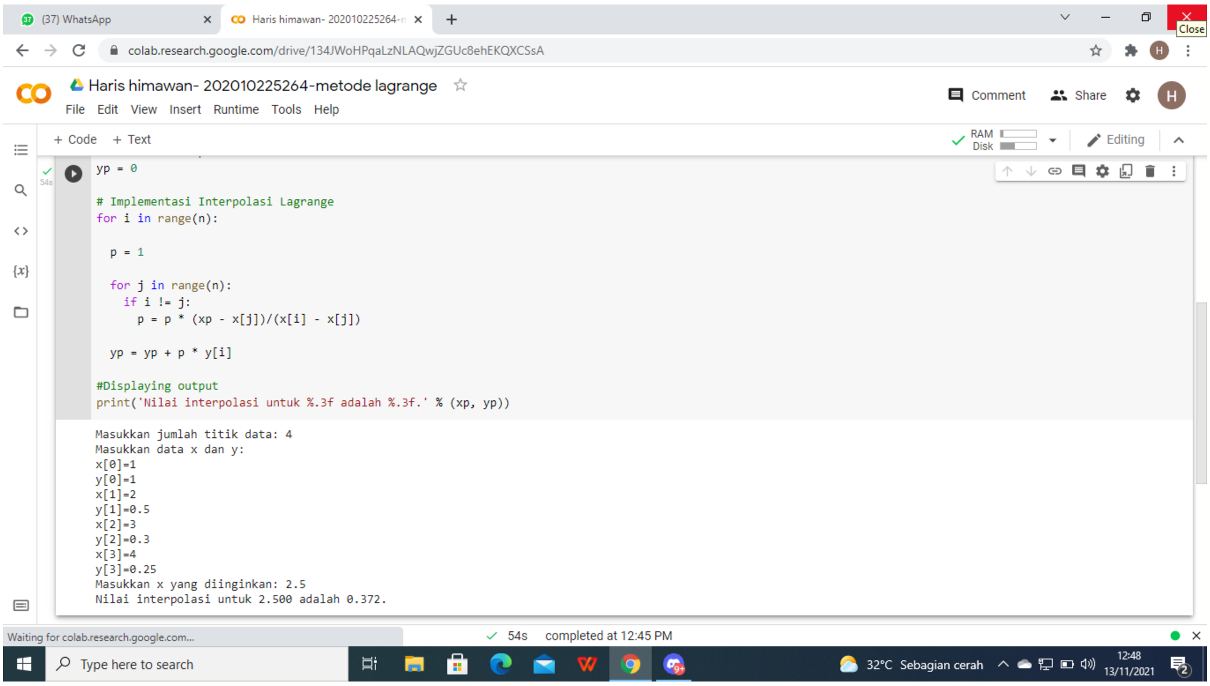
Kelas:TF3A6

Praktikum metode numerik 3

1.metode lagrange

#SCRIPT

```
# 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))
```


(37) WhatsApp
Haris himawan- 202010225264-
colab.research.google.com/drive/134JWoHPqaLzNLAQwjZGUc8ehEKQXCSsA
Close
Haris himawan- 202010225264-metode lagrange
File
Edit
View
Insert
Runtime
Tools
Help
Comment
Share
+ Code
+ Text
RAM
Disk
Editing
54s
Waiting for colab.research.google.com...
54s completed at 12:45 PM
Type here to search
32°C Sebagian cerah
12:48
13/11/2021

```
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))
```

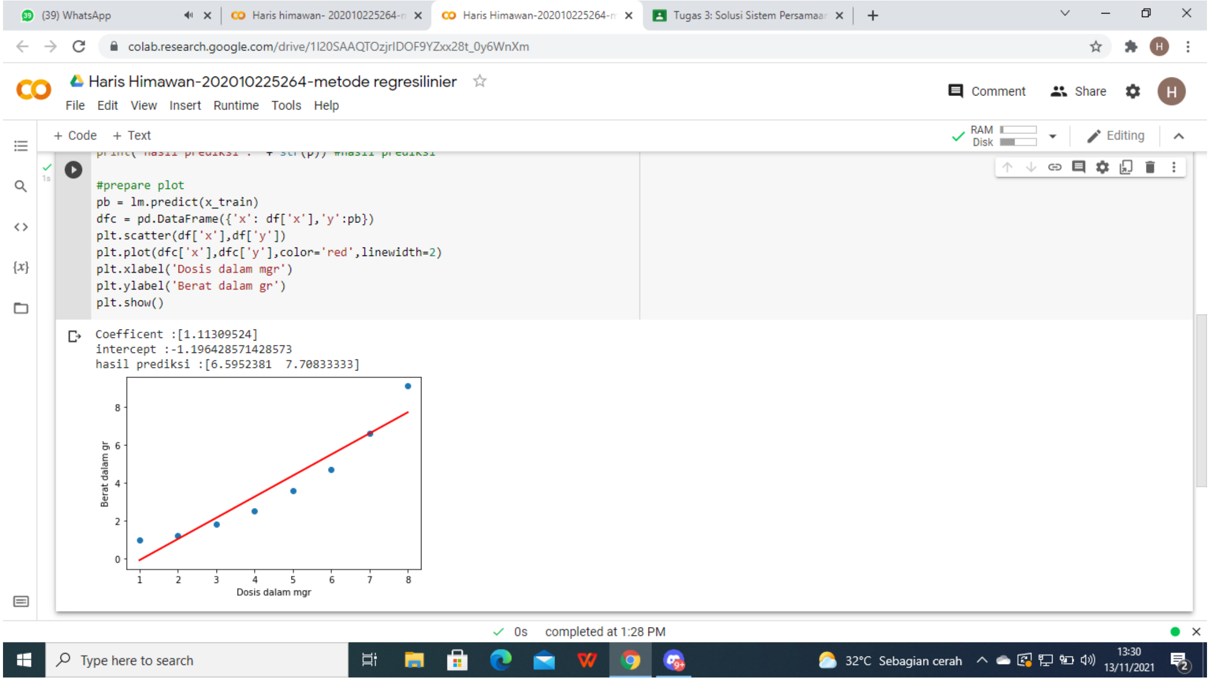
```
Masukkan jumlah titik data: 4
Masukkan data x dan y:
x[0]=1
y[0]=1
x[1]=2
y[1]=0.5
x[2]=3
y[2]=0.3
x[3]=4
y[3]=0.25
Masukkan x yang diinginkan: 2.5
Nilai interpolasi untuk 2.500 adalah 0.372.
```

2.Metode regresi linear

#script

```
# -*- coding: utf-8 -*-
"""MetodeRegresiLinear.py
Automatically generated by Colaboratory.
Original file is located at
    https://colab.research.google.com/drive/1owu8NXWIar9NROWlCXzPQFuectzAzZs5
"""

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()
```



```
#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()
```

```
Coefficent :[1.11309524]
intercept :-1.196428571428573
hasil prediksi :[6.5952381  7.70833333]
```