

Nama : Daffa Pandu Dewanata
NPM : 202010225242
Kelas : TIF3A6

METODE NUMERIK

30-10-2021

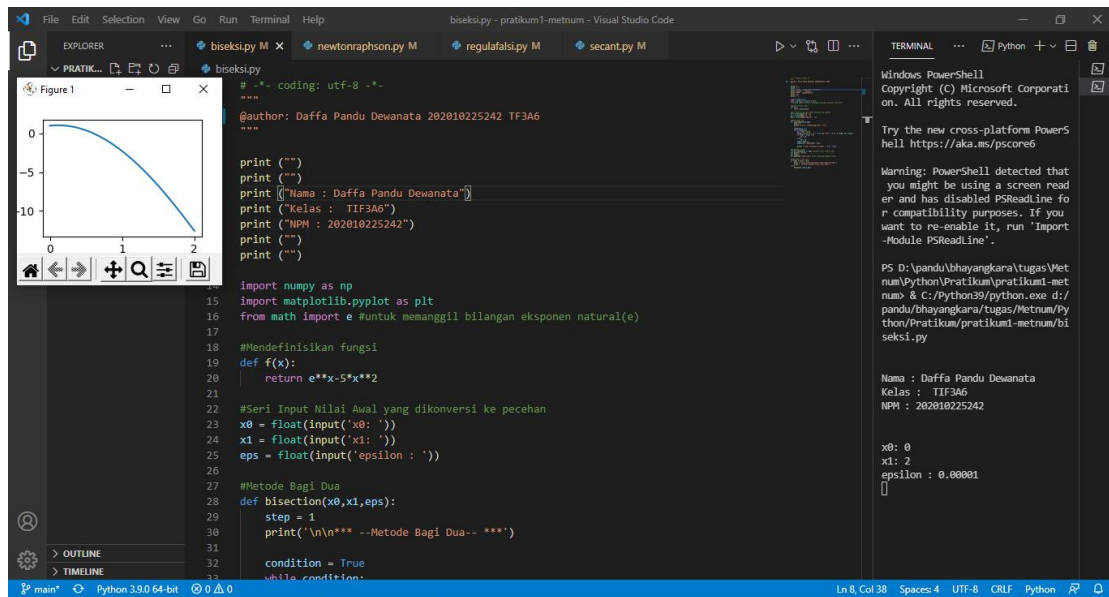
Tugas Pratikum 1 Modul 1

Github Daffa Pandu Dewanata :

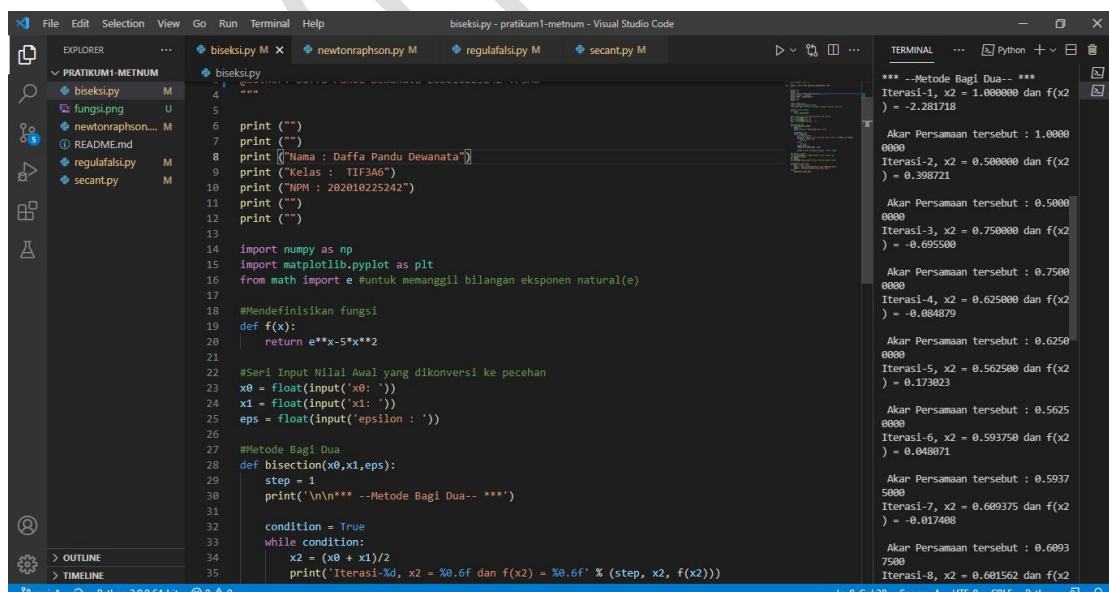
<https://github.com/daffapandu2304/pratikum1-metnum>

Tugas Pratikum Modul 1

A. Script Metode Bagi Dua

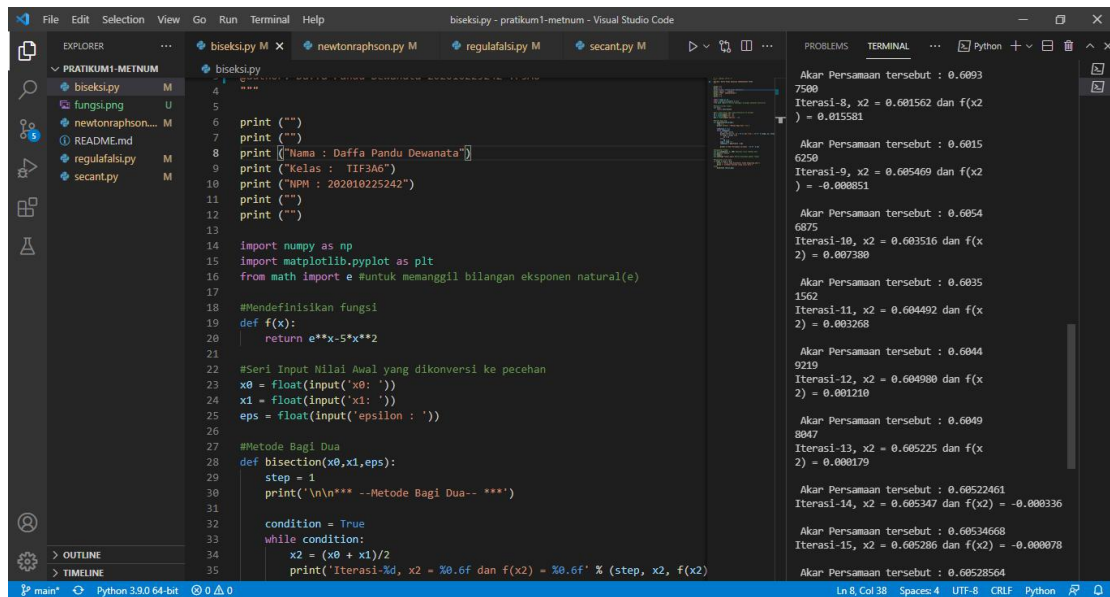


Gambar 1. Hasil Script Python Metode Bagi Dua



Gambar 2. Hasil Script Python Metode Bagi Dua

<https://github.com/daffapandu2304/pratikum1-metnum>

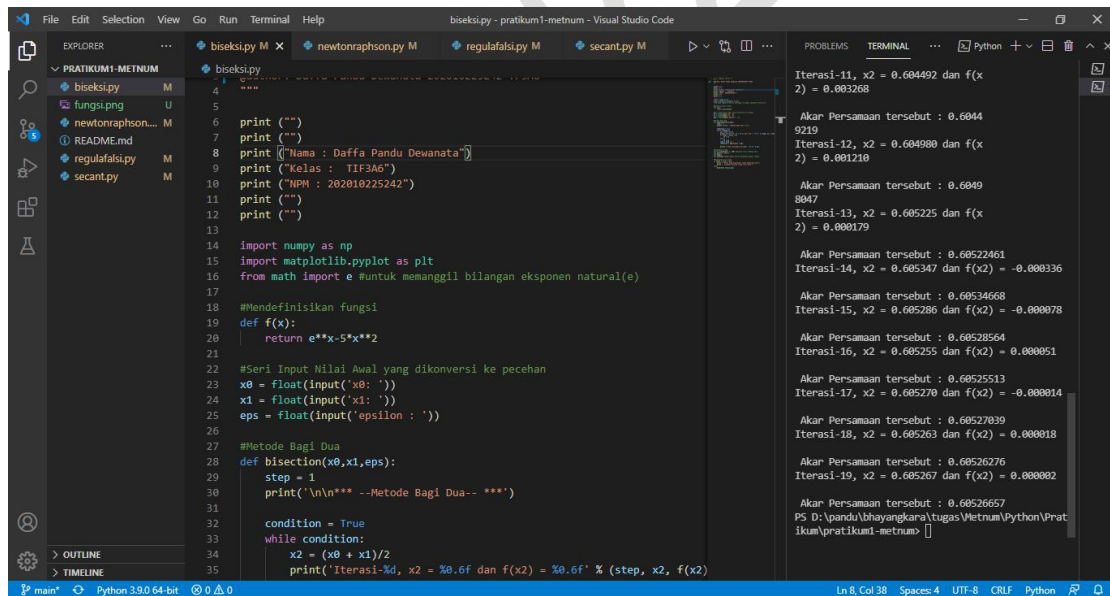


```
File Edit Selection View Go Run Terminal Help
bisiksi.py - pratikum1-metnum - Visual Studio Code

EXPLORER
PRATIKUM1-METNUM
bisiksi.py M
fungsi.png U
newtonraphson... M
README.md
regulafalsi.py M
secant.py M

bisiksi.py
1 # -*- coding: utf-8 -*-
2 """
3     Nama : Daffa Pandu Dewanata
4     Kelas : TIF3A6
5     NPM : 202010225242
6 """
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #untuk memanggil bilangan eksponen natural(e)
17
18 #Mendefinisikan fungsi
19 def f(x):
20     return e**x-5*x**2
21
22 #Seri Input Nilai Awal yang dikonversi ke pecahan
23 x0 = float(input('x0: '))
24 x1 = float(input('x1: '))
25 eps = float(input('epsilon: '))
26
27 #Metode Bagi Dua
28 def bisection(x0,x1,eps):
29     step = 1
30     print('\n\n*** --Metode Bagi Dua-- ***')
31
32     condition = True
33     while condition:
34         x2 = (x0 + x1)/2
35         print('Iterasi-%d, x2 = %.6f dan f(x2) = %.6f' % (step, x2, f(x2))
36
37         Akar Persamaan tersebut : 0.6093
38         Iterasi-8, x2 = 0.601562 dan f(x2) = 0.015581
39
40         Akar Persamaan tersebut : 0.6015
41         Iterasi-9, x2 = 0.605469 dan f(x2) = -0.000851
42
43         Akar Persamaan tersebut : 0.6054
44         Iterasi-10, x2 = 0.603516 dan f(x2) = 0.007380
45
46         Akar Persamaan tersebut : 0.6035
47         Iterasi-11, x2 = 0.604492 dan f(x2) = 0.003268
48
49         Akar Persamaan tersebut : 0.6044
50         Iterasi-12, x2 = 0.604980 dan f(x2) = 0.001210
51
52         Akar Persamaan tersebut : 0.6049
53         Iterasi-13, x2 = 0.605225 dan f(x2) = 0.000179
54
55         Akar Persamaan tersebut : 0.60522461
56         Iterasi-14, x2 = 0.605347 dan f(x2) = -0.000336
57
58         Akar Persamaan tersebut : 0.60534668
59         Iterasi-15, x2 = 0.605286 dan f(x2) = -0.000078
60
61         Akar Persamaan tersebut : 0.60528564
```

Gambar 3. Hasil Script Python Metode Bagi Dua



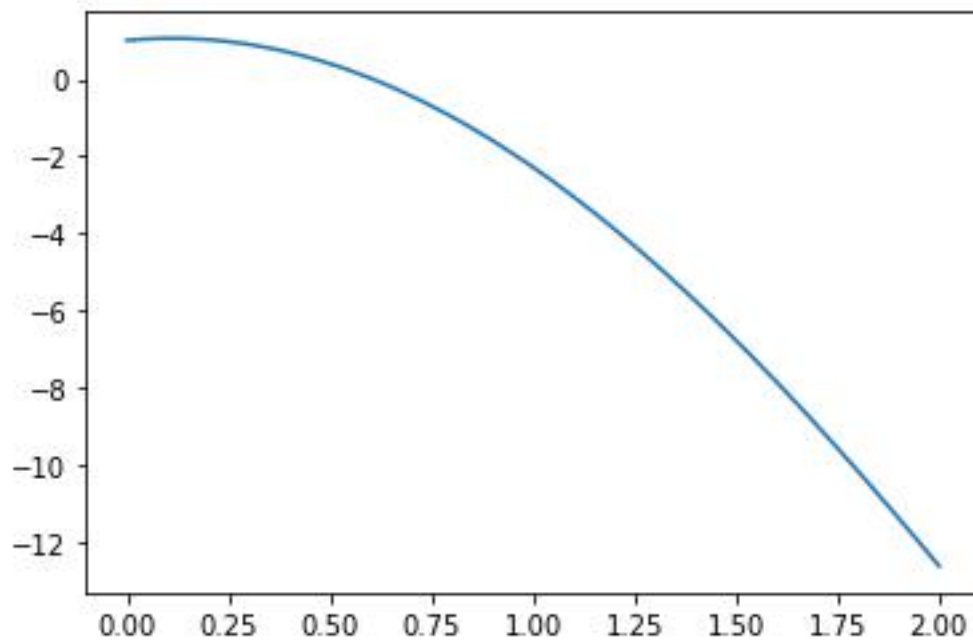
```
File Edit Selection View Go Run Terminal Help
bisiksi.py - pratikum1-metnum - Visual Studio Code

EXPLORER
PRATIKUM1-METNUM
bisiksi.py M
fungsi.png U
newtonraphson... M
README.md
regulafalsi.py M
secant.py M

bisiksi.py
1 # -*- coding: utf-8 -*-
2 """
3     Nama : Daffa Pandu Dewanata
4     Kelas : TIF3A6
5     NPM : 202010225242
6 """
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #untuk memanggil bilangan eksponen natural(e)
17
18 #Mendefinisikan fungsi
19 def f(x):
20     return e**x-5*x**2
21
22 #Seri Input Nilai Awal yang dikonversi ke pecahan
23 x0 = float(input('x0: '))
24 x1 = float(input('x1: '))
25 eps = float(input('epsilon: '))
26
27 #Metode Bagi Dua
28 def bisection(x0,x1,eps):
29     step = 1
30     print('\n\n*** --Metode Bagi Dua-- ***')
31
32     condition = True
33     while condition:
34         x2 = (x0 + x1)/2
35         print('Iterasi-%d, x2 = %.6f dan f(x2) = %.6f' % (step, x2, f(x2))
36
37         Akar Persamaan tersebut : 0.60492 dan f(x2) = 0.003268
38
39         Akar Persamaan tersebut : 0.6044
40         Iterasi-12, x2 = 0.604980 dan f(x2) = 0.001210
41
42         Akar Persamaan tersebut : 0.6049
43         Iterasi-13, x2 = 0.605225 dan f(x2) = 0.000179
44
45         Akar Persamaan tersebut : 0.60522461
46         Iterasi-14, x2 = 0.605347 dan f(x2) = -0.000336
47
48         Akar Persamaan tersebut : 0.60534668
49         Iterasi-15, x2 = 0.605286 dan f(x2) = -0.000078
50
51         Akar Persamaan tersebut : 0.60528564
52         Iterasi-16, x2 = 0.605255 dan f(x2) = 0.000051
53
54         Akar Persamaan tersebut : 0.60525513
55         Iterasi-17, x2 = 0.605270 dan f(x2) = -0.000014
56
57         Akar Persamaan tersebut : 0.60527039
58         Iterasi-18, x2 = 0.605263 dan f(x2) = 0.000018
59
60         Akar Persamaan tersebut : 0.60526276
61         Iterasi-19, x2 = 0.605267 dan f(x2) = 0.000002
62
63         Akar Persamaan tersebut : 0.60526657
64         PS D:\pandu\bhayangkara\tugas\Vietnum\Python\Pratikum\pratikum1-metnum> []
```

Gambar 4. Hasil Script Python Metode Bagi Dua

METODE NUMERIK



Gambar 5. Hasil Script Python Metode Bagi Dua

Script Metode Bagi Dua

```
# -*- coding: utf-8 -*-
"""
@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""

print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt
from math import e #untuk memanggil bilangan eksponen natural(e)

#Mendefinisikan fungsi
def f(x):
```

```
    return e**x-5*x**2

#Seri Input Nilai Awal yang dikonversi ke pecahan
x0 = float(input('x0: '))
x1 = float(input('x1: '))
eps = float(input('epsilon : '))

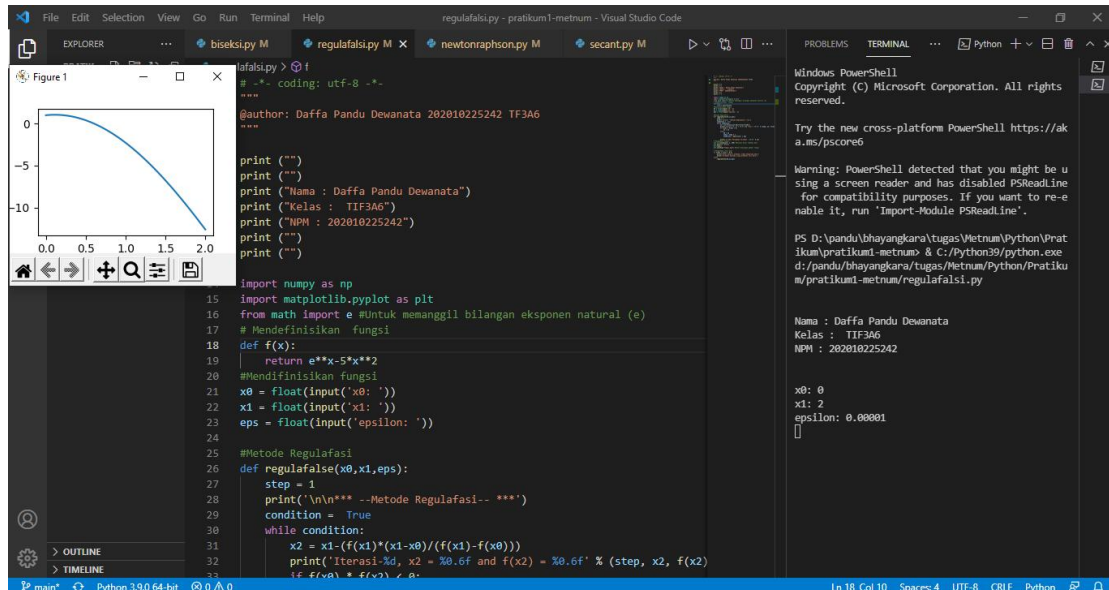
#Metode Bagi Dua
def bisection(x0,x1,eps):
    step = 1
    print('\n\n*** --Metode Bagi Dua-- ***')

    condition = True
    while condition:
        x2 = (x0 + x1)/2
        print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
        step = step + 1
        condition = abs(f(x2)) > eps

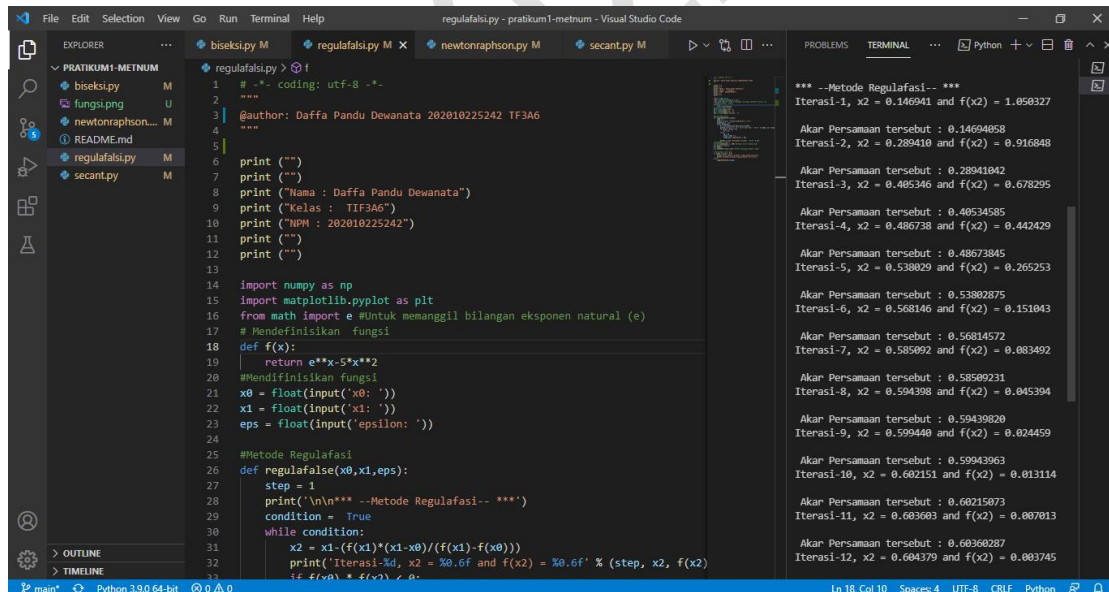
    print('\n Akar Persamaan tersebut : %0.8f' % x2)

#menggambar fungsi
rr= np.linspace(0, 2, 100) #masukan nilai tebakan awal
plt.plot(rr, f(rr))
plt.show()
plt.savefig("fungsi.png") #untuk menyimpan gambar fungsi

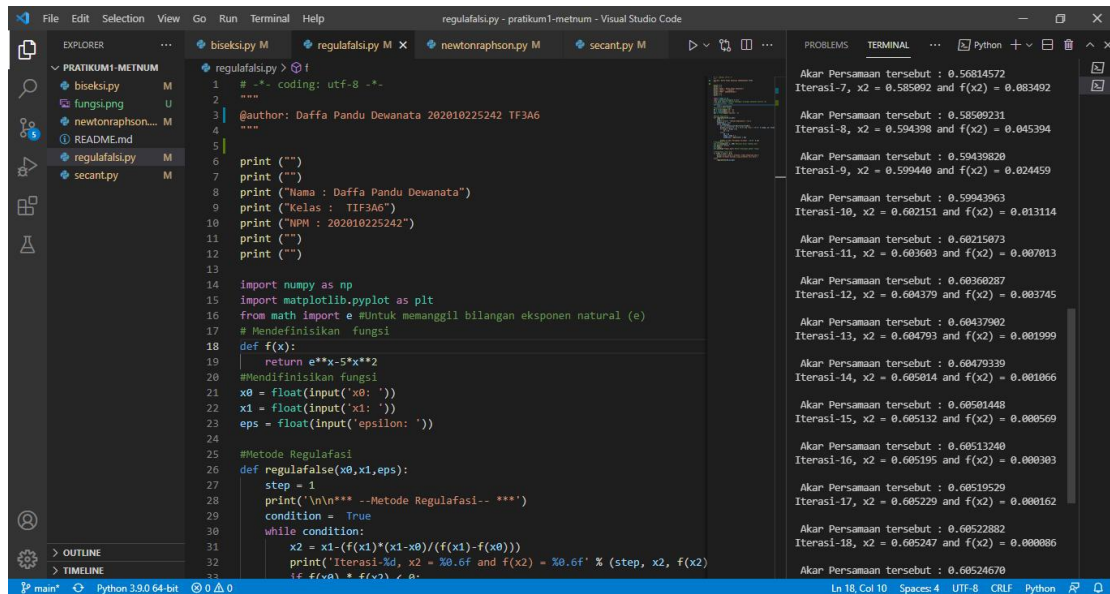
#Pengecekan nilai awal
if f(x0) * f(x1) > 0.0:
    print (' Nilai yang diprediksi tidak mengurung akar')
    print (' Silahkan mencoba ulang nilai baru')
else:
    bisection (x0,x1,eps)
```

Tugas Pratikum Modul 2**B. Script Metode Regulasfalsi**

Gambar 1. Hasil Script Python Metode Regulasfalsi



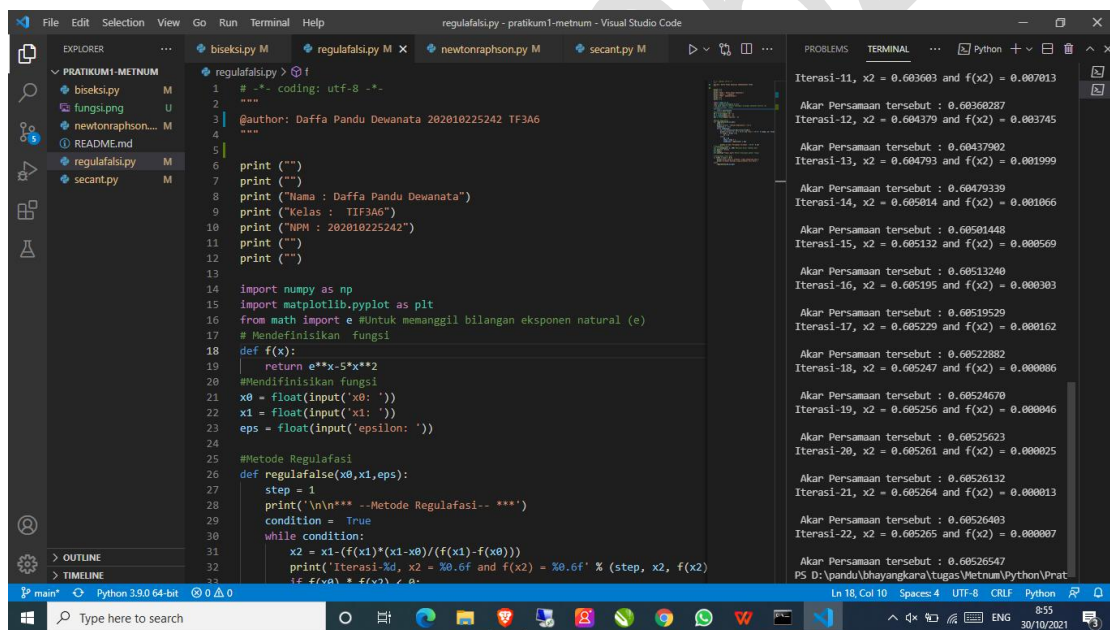
Gambar 2. Hasil Script Python Metode Regulasfalsi



```
1 # -*- coding: utf-8 -*-
2 """
3 @author: Daffa Pandu Dewanata 202010225242 TIF3A6
4 """
5
6 print("")
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #Untuk memanggil bilangan eksponen natural (e)
17 # Mendefinisikan fungsi
18 def f(x):
19     return e**x-5*x**2
20 #Mendefinisikan fungsi
21 x0 = float(input('x0: '))
22 x1 = float(input('x1: '))
23 eps = float(input('epsilon: '))
24
25 #Metode Regula-Falsi
26 def regufalse(x0,x1,eps):
27     step = 1
28     print('\n\n*** --Metode Regula-Falsi-- ***')
29     condition = True
30     while condition:
31         x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
32         print('Iterasi-%d, x2 = %.6f and f(x2) = %.6f' % (step, x2, f(x2)))
33         if f(x0) * f(x2) < 0:
```

Akar Persamaan tersebut : 0.56814572
Iterasi-7, x2 = 0.585092 and f(x2) = 0.083492
Akar Persamaan tersebut : 0.58509231
Iterasi-8, x2 = 0.594398 and f(x2) = 0.045394
Akar Persamaan tersebut : 0.59439820
Iterasi-9, x2 = 0.599448 and f(x2) = 0.024459
Akar Persamaan tersebut : 0.59943963
Iterasi-10, x2 = 0.602151 and f(x2) = 0.013114
Akar Persamaan tersebut : 0.60215073
Iterasi-11, x2 = 0.603603 and f(x2) = 0.007013
Akar Persamaan tersebut : 0.60360287
Iterasi-12, x2 = 0.604379 and f(x2) = 0.003745
Akar Persamaan tersebut : 0.60437902
Iterasi-13, x2 = 0.604793 and f(x2) = 0.001999
Akar Persamaan tersebut : 0.60479339
Iterasi-14, x2 = 0.605014 and f(x2) = 0.001066
Akar Persamaan tersebut : 0.60501448
Iterasi-15, x2 = 0.605132 and f(x2) = 0.000569
Akar Persamaan tersebut : 0.60513240
Iterasi-16, x2 = 0.605195 and f(x2) = 0.000303
Akar Persamaan tersebut : 0.60519529
Iterasi-17, x2 = 0.605229 and f(x2) = 0.000162
Akar Persamaan tersebut : 0.60522882
Iterasi-18, x2 = 0.605247 and f(x2) = 0.000086
Akar Persamaan tersebut : 0.60524670

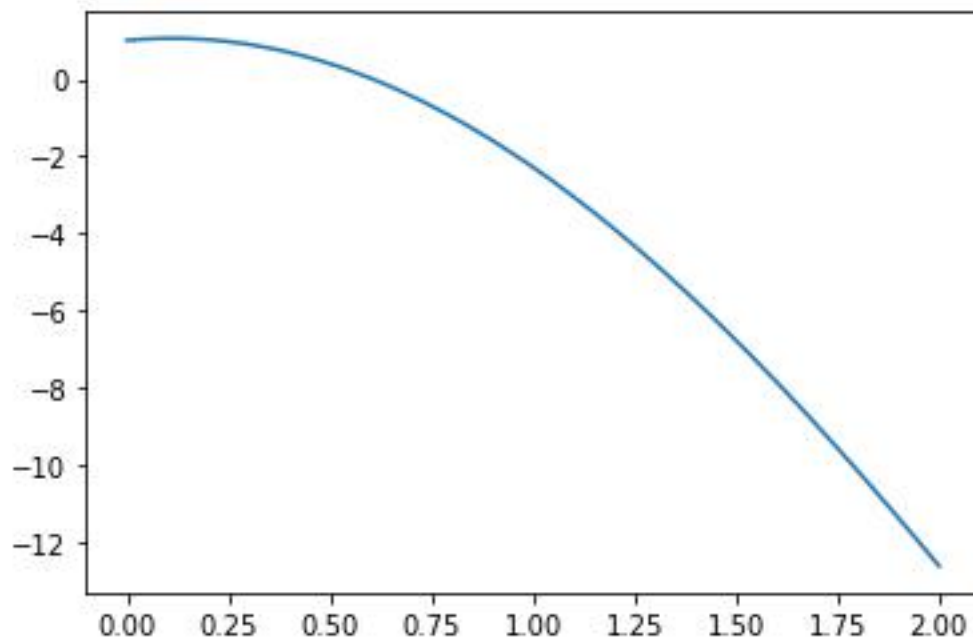
Gambar 3. Hasil Script Python Metode Regula-Falsi



```
24
25 #Metode Regula-Falsi
26 def regufalse(x0,x1,eps):
27     step = 1
28     print('\n\n*** --Metode Regula-Falsi-- ***')
29     condition = True
30     while condition:
31         x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
32         print('Iterasi-%d, x2 = %.6f and f(x2) = %.6f' % (step, x2, f(x2)))
33         if f(x0) * f(x2) < 0:
```

Iterasi-11, x2 = 0.603603 and f(x2) = 0.007013
Akar Persamaan tersebut : 0.60360287
Iterasi-12, x2 = 0.604379 and f(x2) = 0.003745
Akar Persamaan tersebut : 0.60437902
Iterasi-13, x2 = 0.604793 and f(x2) = 0.001999
Akar Persamaan tersebut : 0.60479339
Iterasi-14, x2 = 0.605014 and f(x2) = 0.001066
Akar Persamaan tersebut : 0.60501448
Iterasi-15, x2 = 0.605132 and f(x2) = 0.000569
Akar Persamaan tersebut : 0.60513240
Iterasi-16, x2 = 0.605195 and f(x2) = 0.000303
Akar Persamaan tersebut : 0.60519529
Iterasi-17, x2 = 0.605229 and f(x2) = 0.000162
Akar Persamaan tersebut : 0.60522882
Iterasi-18, x2 = 0.605247 and f(x2) = 0.000086
Akar Persamaan tersebut : 0.60524670
Iterasi-19, x2 = 0.605256 and f(x2) = 0.000046
Akar Persamaan tersebut : 0.60525623
Iterasi-20, x2 = 0.605261 and f(x2) = 0.000025
Akar Persamaan tersebut : 0.60526132
Iterasi-21, x2 = 0.605264 and f(x2) = 0.000013
Akar Persamaan tersebut : 0.60526403
Iterasi-22, x2 = 0.605265 and f(x2) = 0.000007
Akar Persamaan tersebut : 0.60526547
PS D:\pandu\bhayangkara\tugas\metnum\Python\Prat:

Gambar 4. Hasil Script Python Metode Regula-Falsi



Gambar 5. Hasil Script Python Metode RegulaFalsi

Script Metode RegulaFalsi

```
# -*- coding: utf-8 -*-
"""
@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""

print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt
from math import e #Untuk memanggil bilangan eksponen natural (e)
# Mendefinisikan fungsi
def f(x):
    return e**x-5*x**2
#Mendefinisikan fungsi
x0 = float(input('x0: '))
x1 = float(input('x1: '))
eps = float(input('epsilon: '))

#Metode RegulaFalsi
def regulafalse(x0,x1,eps):
```

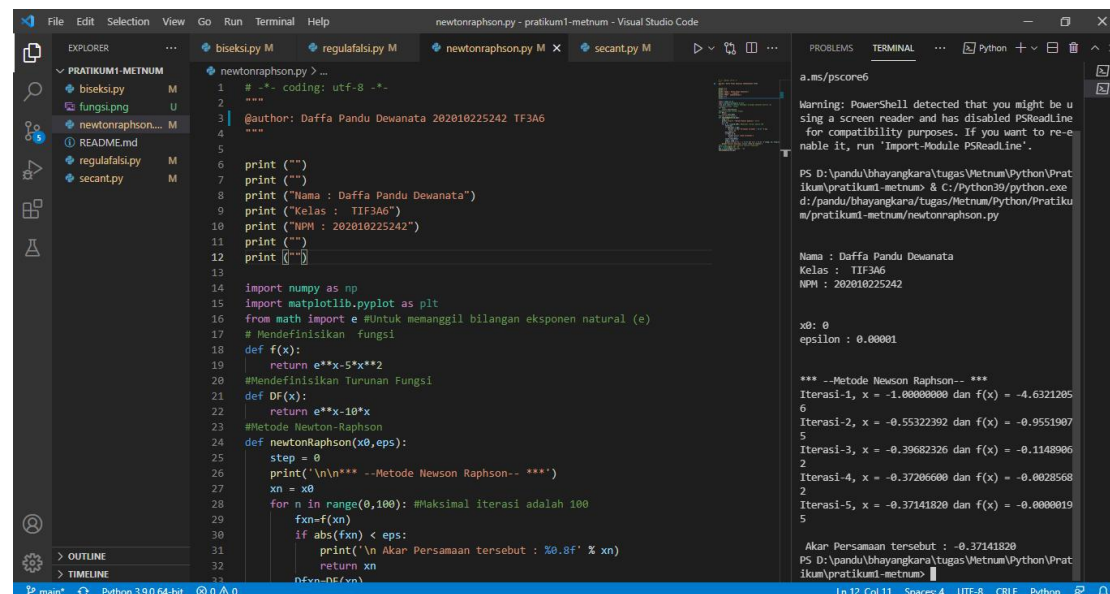
```
step = 1
print('\n\n*** --Metode Regulasi-- ***')
condition = True
while condition:
    x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
    print('Iterasi-%d, x2 = %0.6f and f(x2) = %0.6f' % (step, x2, f(x2)))
    if f(x0) * f(x2) < 0:
        x1 = x2
    else:
        x0 = x2
        step = step + 1
        condition = abs(f(x2)) > eps

print('\n Akar Persamaan tersebut : %0.8f' % x2)
# Menggambar Fungsi
rr= np.linspace(0, 2, 100) #Masukan Nilai tebakan Awal
plt.plot(rr, f(rr))
plt.show()
plt.savefig("fungsi.png") #Untuk menyimpan gambar fungsi

# Pengecekan nilai awal
if f(x0) * f(x1) > 0.0:
    print('Nilai yang di prediksi tidak mengurung akar')
    print('Silahkan mencoba ulang prediksi nili baru')
else:
    regulafalse(x0,x1,eps)
```


Tugas Pratikum Modul 3

C. Script Metode NewtonRaphson



```
newtonraphson.py - pratikum1-metnum - Visual Studio Code
EXPLORER
PRATIKUM1-METNUM
  biseksi.py M
  fungsi.png U
  newtonraphson... M
  README.md
  regulafalsi.py M
  secant.py M
  newtonraphson.py > ...
1 # -*- coding: utf-8 -*-
2 """
3 @author: Daffa Pandu Dewanata 202010225242 TF3A6
4 """
5
6 print("")
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #Untuk memanggil bilangan eksponen natural (e)
17 # Mendefinisikan fungsi
18 def f(x):
19     return e**x-5*x**2
20 #Mendefinisikan Turunan Fungsi
21 def DF(x):
22     return e**x-10*x
23 #Metode Newton-Raphson
24 def newtonRaphson(x0,eps):
25     step = 0
26     print('\n\n*** --Metode Newson Raphson-- ***')
27     xn = x0
28     for n in range(0,100): #Maksimal Iterasi adalah 100
29         fxn=f(xn)
30         if abs(fxn) < eps:
31             print('\n Akar Persamaan tersebut : %0.8f' % xn)
32             return xn
33         Dfxn=DF(xn)
34         xn=xn-fxn/Dfxn
35     return xn
36
37 #Main
38 x0 = 0
39 eps = 0.00001
40
41 #Metode Newson Raphson-- ***
42 Iterasi-1, x = -1.00000000 dan f(x) = -4.6321285
43 Iterasi-2, x = -0.55322392 dan f(x) = -0.9551987
44 Iterasi-3, x = -0.39682326 dan f(x) = -0.1148906
45 Iterasi-4, x = -0.37206600 dan f(x) = -0.0028568
46 Iterasi-5, x = -0.37141820 dan f(x) = -0.0000019
47
48 Akar Persamaan tersebut : -0.37141820
49 PS D:\pandu\bhayangkara\tugas\Vietnum\Python\Pratikum\pratikum1-metnum> & C:/Python39/python.exe d:/pandu/bhayangkara/tugas/Metnum/Python/Pratikum/pratikum1-metnum/newtonraphson.py
50
51 Nama : Daffa Pandu Dewanata
52 Kelas : TIF3A6
53 NPM : 202010225242
54
55 x0: 0
56 epsilon : 0.00001
57
58 *** --Metode Newson Raphson-- ***
59 Iterasi-1, x = -1.00000000 dan f(x) = -4.6321285
60 Iterasi-2, x = -0.55322392 dan f(x) = -0.9551987
61 Iterasi-3, x = -0.39682326 dan f(x) = -0.1148906
62 Iterasi-4, x = -0.37206600 dan f(x) = -0.0028568
63 Iterasi-5, x = -0.37141820 dan f(x) = -0.0000019
64
65 Akar Persamaan tersebut : -0.37141820
66 PS D:\pandu\bhayangkara\tugas\Vietnum\Python\Pratikum\pratikum1-metnum>
```

Gambar 1. Hasil Script Python Metode NewtonRaphson

Script Metode NewtonRapshon

```
# -*- coding: utf-8 -*-
"""
@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""
```

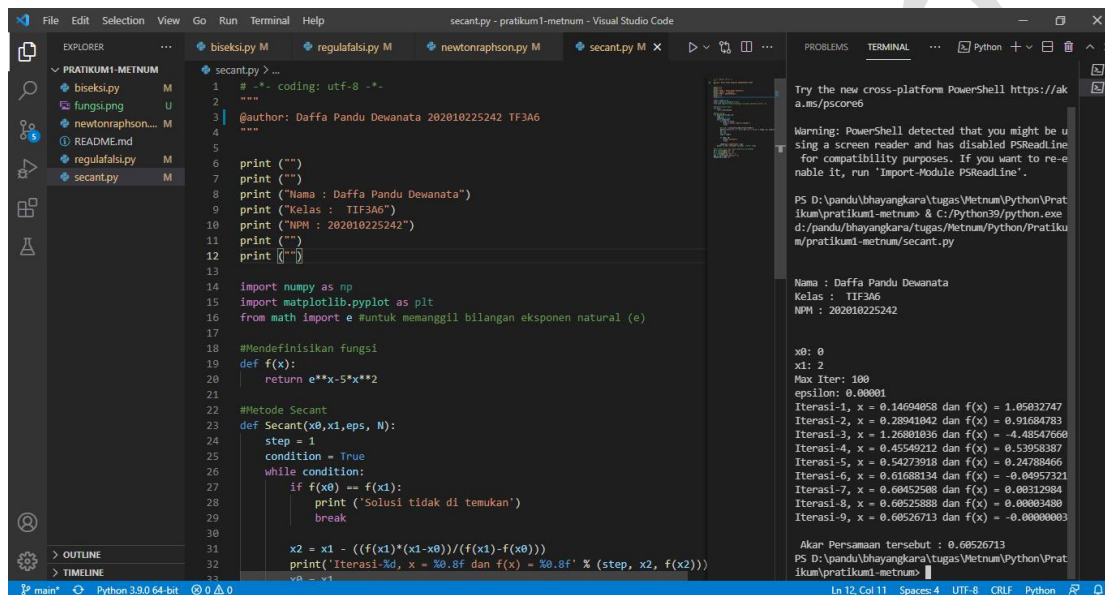
```
print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")
```

```
import numpy as np
import matplotlib.pyplot as plt
```

```
from math import e #Untuk memanggil bilangan eksponen natural (e)
# Mendefinisikan fungsi
def f(x):
    return e**x-5*x**2
#Mendefinisikan Turunan Fungsi
def DF(x):
    return e**x-10*x
#Metode Newton-Raphson
def newtonRaphson(x0,eps):
    step = 0
    print('\n\n*** --Metode Newson Raphson-- ***')
    xn = x0
    for n in range(0,100): #Maksimal iterasi adalah 100
        fxn=f(xn)
        if abs(fxn) < eps:
            print('\n Akar Persamaan tersebut : %0.8f' % xn)
            return xn
        Dfxn=DF(xn)
        if Dfxn == 0:
            print('Solusi tidak ditemukan')
            return None
        xn=xn-(fxn/Dfxn)
        step = step + 1
        print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, xn, f(xn)))
    print('Iterasi maksimum, solusi tidak di temukan')
#Sesi Input Nilai awal yang di konversi kepecahan
x0 = float(input('x0: '))
eps = float(input('epsilon : '))
newtonRaphson(x0,eps)
```

Tugas Pratikum Modul 4

D. Script Metode Secant



```
1 # -*- coding: utf-8 -*-
2 """
3 @author: Daffa Pandu Dewanata 202010225242 TF3A6
4 """
5
6 print("")
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #untuk memanggil bilangan eksponen natural (e)
17
18 #Mendefinisikan fungsi
19 def f(x):
20     return e**x-5*x**2
21
22 #Metode Secant
23 def Secant(x0,x1,eps,N):
24     step = 1
25     condition = True
26     while condition:
27         if f(x0) == f(x1):
28             print("Solusi tidak di temukan")
29             break
30         x2 = x1 - ((f(x1)*(x1-x0))/(f(x1)-f(x0)))
31         print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, x2, f(x2)))
32         x0 = x1
33         x1 = x2
```

Gambar 1. Hasil Script Python Metode Secant

Script Metode Secant

```
# -*- coding: utf-8 -*-
"""
@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""

print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt
from math import e #untuk memanggil bilangan eksponen natural (e)

#Mendefinisikan fungsi
```

```
def f(x):  
    return e**x-5*x**2  
  
#Metode Secant  
def Secant(x0,x1,eps, N):  
    step = 1  
    condition = True  
    while condition:  
        if f(x0) == f(x1):  
            print ('Solusi tidak di temukan')  
            break  
  
        x2 = x1 - ((f(x1)*(x1-x0))/(f(x1)-f(x0)))  
        print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, x2, f(x2)))  
        x0 = x1  
        x1 = x2  
        step = step+1  
  
    if step > N:  
        print('Divergen')  
        break  
  
    condition = abs(f(x2)) > eps  
    print("\n Akar Persamaan tersebut : %0.8f" % x2)  
  
#Sesi Input Nilai Awal yang dikonversi ke pecahan  
x0 = float(input('x0: '))  
x1 = float(input('x1: '))  
N = int(input('Max Iter: '))  
eps = float(input('epsilon: '))  
Secant(x0,x1,eps, N)
```

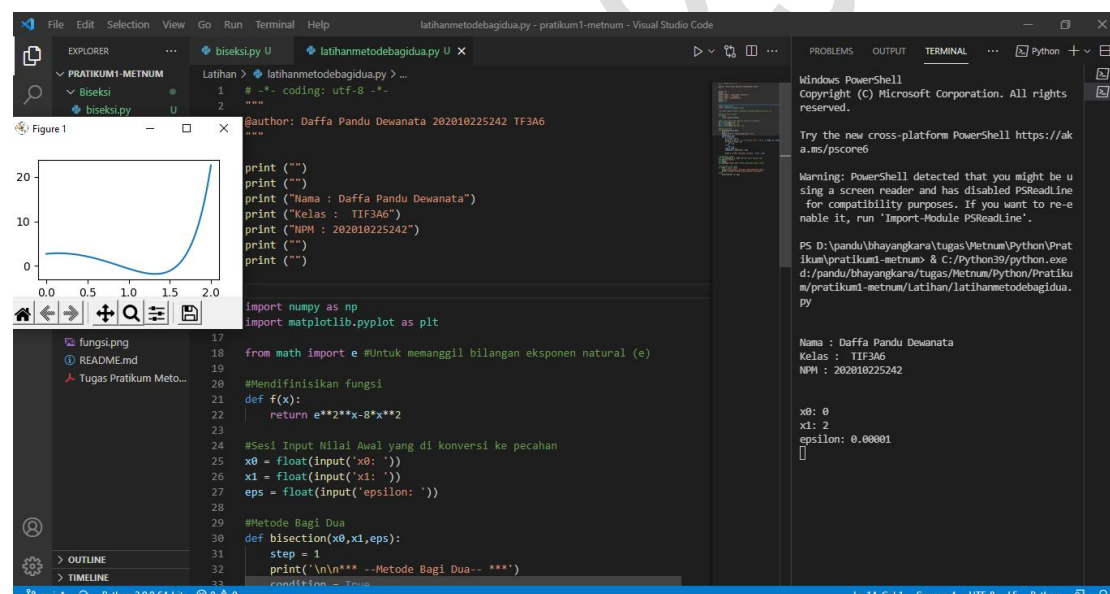
Tugas Pratkum Latihan Soal

Hitunglah akar persamaan non linear $f(x) = e^{x^2} - 8x^2$ dengan metode :

1. Bagi dua, gunakan $\epsilon = 0.00001$ dan tebakan awal $a = 0$ dan $b = 2$.
2. Regula falsi, gunakan $\epsilon = 0.00001$ dan tebakan awal $a = 0$ dan $b = 2$.
3. Newton Raphson, gunakan $\epsilon = 0.00001$ dan tebakan awal $x_0 = 0$.
4. Secant, gunakan $\epsilon = 0.00001$ dan tebakan awal $x_0 = 0$ dan $x_1 = 2$.
5. Berdasarkan perhitungan 1 hingga 4, metode mana yang terbaik, dan berikan alasannya.

Jawaban :

1. Latihan Biseksi Metode Bagi Dua



Gambar 1. Hasil Script Python Latihan Metode Bagi Dua

```

1 # -*- coding: utf-8 -*-
2 """
3 @author: Daffa Pandu Dewanata 202010225242 TF3A6
4 """
5
6 print("")
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14
15 import numpy as np
16 import matplotlib.pyplot as plt
17
18 from math import e #Untuk memanggil bilangan eksponen natural (e)
19
20 #Mendefinisikan fungsi
21 def f(x):
22     return e**2*x-8*x**2
23
24 #Sesi Input Nilai Awal yang di konversi ke pecahan
25 x0 = float(input('x0: '))
26 x1 = float(input('x1: '))
27 eps = float(input('epsilon: '))
28
29 #Metode Bagi Dua
30 def bisection(x0,x1,eps):
31     step = 1
32     print('\n\n*** --Metode Bagi Dua-- ***')
33     condition = True

```

Terminal Output:

```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

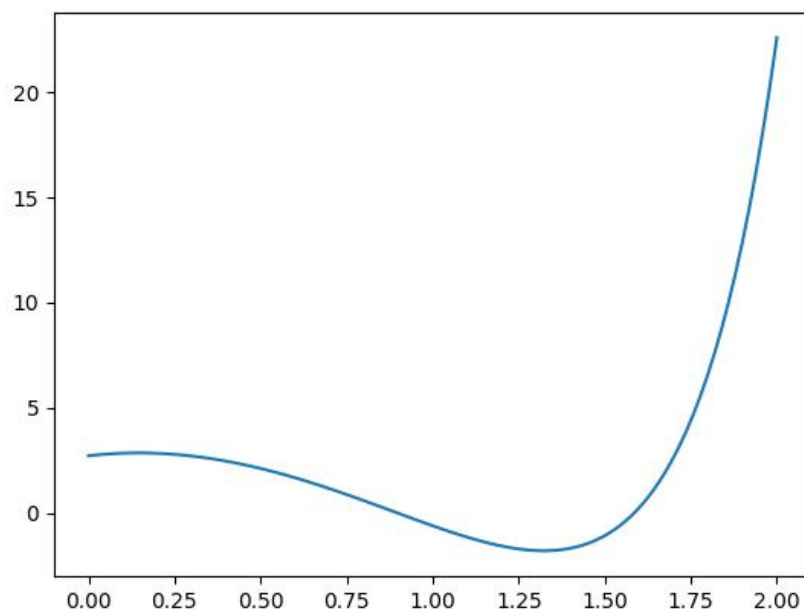
PS D:\pandu\bhayangkara\tugas\Metnum\Python\Pratikum\pratikum1-metnum> & C:/Python39/python.exe d:/pandu/bhayangkara/tugas/Metnum/Python/Pratikum/pratikum1-metnum/Latihan/latihanmetodebagidua.py

Nama : Daffa Pandu Dewanata
Kelas : TIF3A6
NPM : 202010225242

x0: 0
x1: 2
epsilon: 0.00001
Nilai yang di prediksi tidak mengurung akar
Silahkan mencoba ulang prediksi nilai baru
PS D:\pandu\bhayangkara\tugas\Metnum\Python\Pratikum\pratikum1-metnum>

```

Gambar 2. Hasil Script Python Latihan Metode Bagi Dua



Gambar 3. Hasil Script Python Latihan Metode Bagi Dua

Script Latihan Metode Bagi Dua

```

# -*- coding: utf-8 -*-
"""

```

```

@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""

```

```

print("")

```

<https://github.com/daffapandu2304/pratikum1-metnum>


```
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt

from math import e #Untuk memanggil bilangan eksponen natural (e)

#Mendefinisikan fungsi
def f(x):
    return e**2**x-8*x**2

#Sesi Input Nilai Awal yang di konversi ke pecahan
x0 = float(input('x0: '))
x1 = float(input('x1: '))
eps = float(input('epsilon: '))

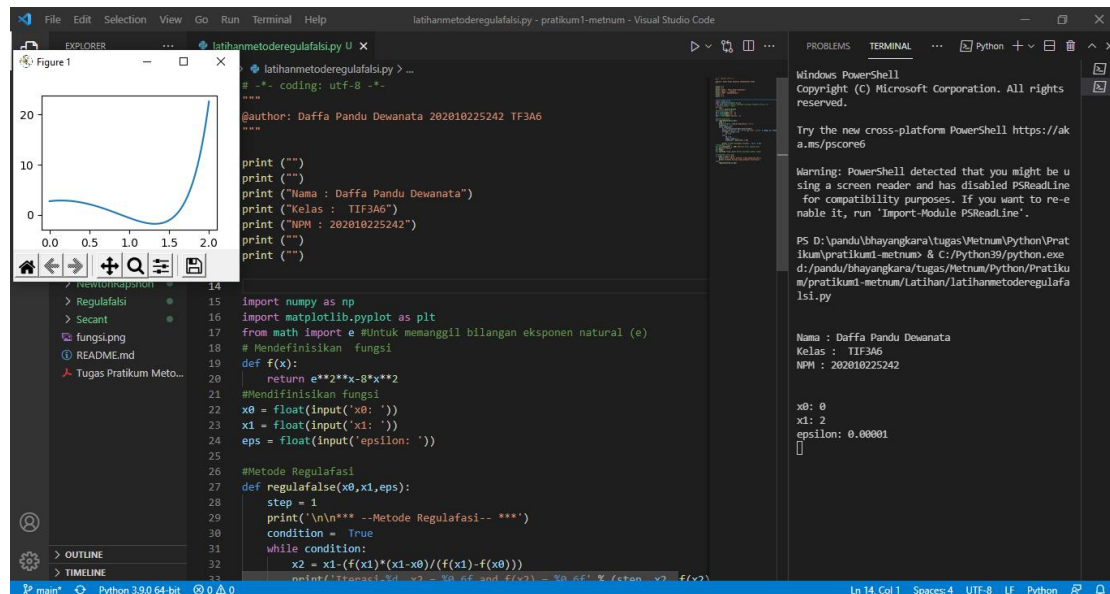
#Metode Bagi Dua
def bisection(x0,x1,eps):
    step = 1
    print('\n\n*** --Metode Bagi Dua-- ***')
    condition = True
    while condition:
        x2 = (x0 + x1)/2
        print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
        step = step + 1
        condition = abs(f(x2)) > eps

    print('\n Akar Persamaan tersebut : %0.8f' % x2)

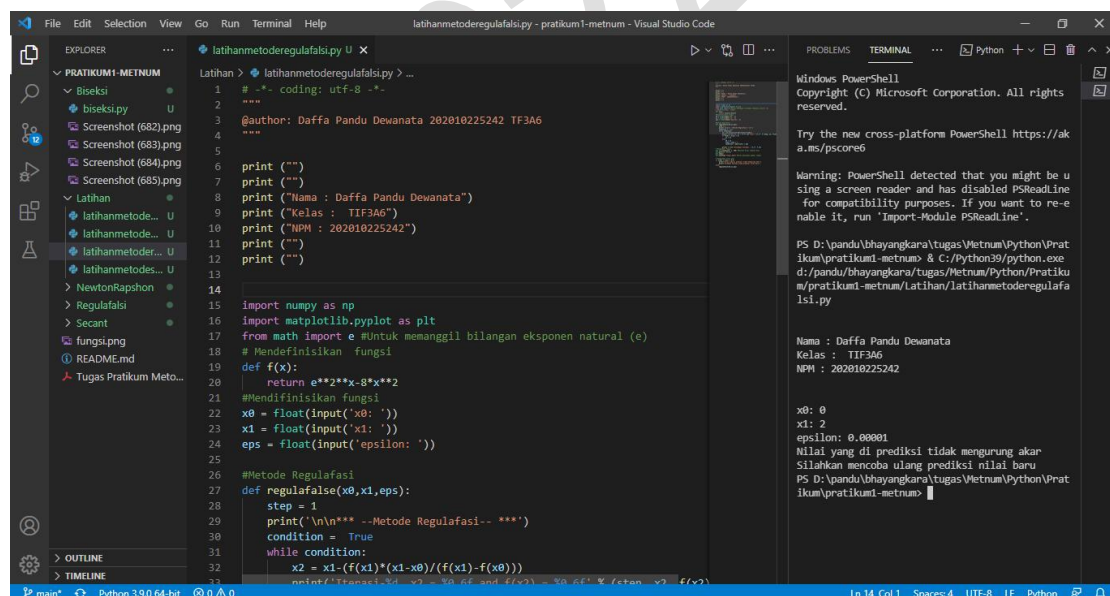
# Menggambar Fungsi
rr= np.linspace(0, 2, 100) #Masukan Nilai tebakan Awal
plt.plot(rr, f(rr))
plt.show()
plt.savefig("fungsi.png") #Untuk menyimpan gambar fungsi

# Pengecekan nilai awal
if f(x0) * f(x1) > 0.0:
    print('Nilai yang di prediksi tidak mengurung akar')
    print('Silahkan mencoba ulang prediksi nilai baru')
else:
    bisection(x0, x1, eps)
```

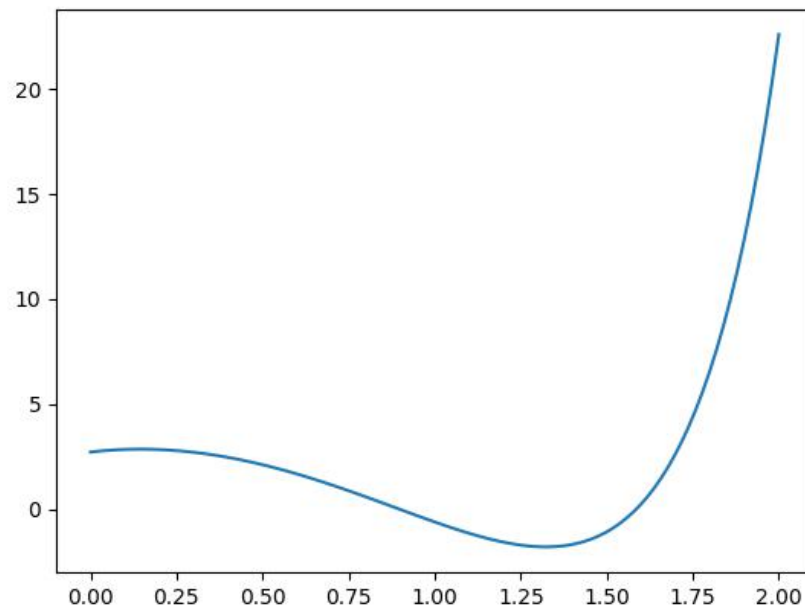
2. Latihan Biseksi Metode RegulaFalsi



Gambar 1. Hasil Script Python Latihan Metode RegulaFalsi



Gambar 2. Hasil Script Python Latihan Metode RegulaFalsi



Gambar 3. Hasil Script Python Latihan Metode RegulaFalsi

Script Latihan Metode RegulaFalsi

```
# -*- coding: utf-8 -*-
"""
@author: Daffa Pandu Dewanata 202010225242 TF3A6
"""

print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt
from math import e #Untuk memanggil bilangan eksponen natural (e)
# Mendefinisikan fungsi
def f(x):
    return e**(2*x-8)*x**2
#Mendefinisikan fungsi
x0 = float(input('x0: '))
x1 = float(input('x1: '))
eps = float(input('epsilon: '))

#Metode RegulaFalsi
def regufalse(x0,x1,eps):
    step = 1
    print('\n\n*** --Metode RegulaFalsi-- ***')
```

```
condition = True
while condition:
    x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
    print('Iterasi-%d, x2 = %0.6f and f(x2) = %0.6f' % (step, x2, f(x2)))
    if f(x0) * f(x2) < 0:
        x1 = x2
    else:
        x0 = x2
        step = step + 1
        condition = abs(f(x2)) > eps

print('\n Akar Persamaan tersebut : %0.8f' % x2)
# Menggambar Fungsi
rr= np.linspace(0, 2, 100) #Masukan Nilai tebakan Awal
plt.plot(rr, f(rr))
plt.show()
plt.savefig("fungsi.png") #Untuk menyimpan gambar fungsi

# Pengecekan nilai awal
if f(x0) * f(x1) > 0.0:
    print('Nilai yang di prediksi tidak mengurung akar')
    print('Silahkan mencoba ulang prediksi nilai baru')
else:
    regulafalse(x0,x1,eps)
```

3. Latihan Biseksi Metode Newton Rapshon

Gambar 1. Hasil Script Python Latihan Metode Newton Rapshon

```

1  # -*- coding: utf-8 -*-
2  """
3  @author: Daffa Pandu Dewanata 202010225242 TF3A6
4  """
5
6  print("")
7  print("")
8  print("Nama : Daffa Pandu Dewanata")
9  print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #Untuk memanggil bilangan eksponen natural (e)
17
18 # Mendefinisikan fungsi
19 def f(x):
20     return e**(2*x)-8*x**2
21
22 #Mendefinisikan Turunan Fungsi
23 def DF(x):
24     return e**(2*x)-16*x
25
26 #Metode Newton-Raphson
27 def newtonRaphson(x0,eps):
28     step = 0
29     print("\n\n*** --Metode Newton Raphson-- ***")
30     xn = x0
31     for n in range(0,100): #Maksimal iterasi adalah 100
32         fxn=f(xn)
33         if abs(fxn) < eps:
34             break
35         xn = xn - f(xn)/DF(xn)
36         step = step + 1
37
38     return xn, step
39
40 # Menampilkan hasil
41 print("Nilai x setelah 100 iterasi adalah: ", newtonRaphson(1, 1e-6))
42
43 # Menampilkan grafik
44 plt.plot(newtonRaphson(1, 1e-6))
45 plt.show()
46
47 # Menampilkan hasil iterasi
48 print("Iterasi 1, x = -0.50084711 dan f(x) = 0.82049009")
49 print("Iterasi 2, x = -0.5049672 dan f(x) = -0.012478")
50 print("Iterasi 3, x = -0.5028764 dan f(x) = 0.0075018")
51 print("Iterasi 4, x = -0.50362045 dan f(x) = -0.004544")
52 print("Iterasi 5, x = -0.50281474 dan f(x) = 0.0027483")
53 print("Iterasi 6, x = -0.50330123 dan f(x) = -0.001657")
54 print("Iterasi 7, x = -0.50300727 dan f(x) = 0.0010004")
55 print("Iterasi 8, x = -0.50318481 dan f(x) = -0.000604")
56 print("Iterasi 9, x = -0.50307756 dan f(x) = 0.0003651")
57 print("Iterasi 10, x = -0.50314234 dan f(x) = -0.000228")
58 print("Iterasi 11, x = -0.50310321 dan f(x) = 0.0001332")
59 print("Iterasi 12, x = -0.50312685 dan f(x) = -0.000080")
60 print("Iterasi 13, x = -0.50311257 dan f(x) = 0.0000486")
61 print("Iterasi 14, x = -0.50312119 dan f(x) = -0.000029")
62 print("Iterasi 15, x = -0.50311598 dan f(x) = 0.0000177")
63 print("Iterasi 16, x = -0.50311913 dan f(x) = -0.000010")
64 print("Iterasi 17, x = -0.50311723 dan f(x) = 0.0000064")
65
66 # Menampilkan hasil iterasi
67 print("Akan Persamaan tersebut : -0.50311723")
68 print("PS: D:\pandu\bhayangkara\tugas\MetNum\Python\Prat...")

```

Gambar 2. Hasil Script Python Latihan Metode Newton Rapshon

Script Latihan Metode Newton Rapshon

```
# -*- coding: utf-8 -*-
"""
```

@author: Daffa Pandu Dewanata 202010225242 TF3A6
 """"

<https://github.com/daffapandu2304/pratikum1-metnum>

```
print("")
print("")
print("Nama : Daffa Pandu Dewanata")
print("Kelas : TIF3A6")
print("NPM : 202010225242")
print("")
print("")

import numpy as np
import matplotlib.pyplot as plt
from math import e #Untuk memanggil bilangan eksponen natural (e)

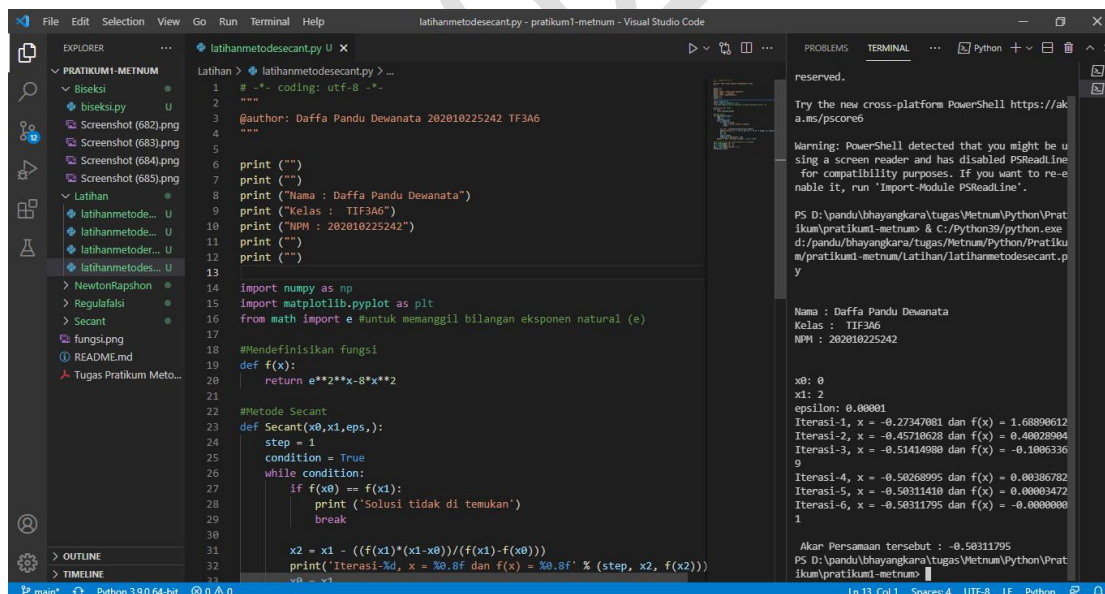
# Mendefinisikan fungsi
def f(x):
    return e**2*x-8*x**2

#Mendefinisikan Turunan Fungsi
def DF(x):
    return e**x-10*x

#Metode Newton-Raphson
def newtonRaphson(x0,eps):
    step = 0
    print('\n\n*** --Metode Newson Raphson-- ***')
    xn = x0
    for n in range(0,100): #Maksimal iterasi adalah 100
        fxn=f(xn)
        if abs(fxn) < eps:
            print('\n Akar Persamaan tersebut : %0.8f' % xn)
            return xn
        Dfxn=DF(xn)
        if Dfxn == 0:
            print('Solusi tidak ditemukan')
            return None
        xn=xn-(fxn/Dfxn)
        step = step + 1
        print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, xn, f(xn)))
    print('Iterasi maksimum, solusi tidak di temukan')

#Sesi Input Nilai awal yang di konversi kepecahan
x0 = float(input('x0: '))
eps = float(input('epsilon : '))
newtonRaphson(x0,eps)
```


4. Latihan Biseksi Metode Secant



```
1 # -*- coding: utf-8 -*-
2 """
3 @author: Daffa Pandu Dewanata 202010225242 TF3A6
4 """
5
6 print("")
7 print("")
8 print("Nama : Daffa Pandu Dewanata")
9 print("Kelas : TIF3A6")
10 print("NPM : 202010225242")
11 print("")
12 print("")
13
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from math import e #untuk memanggil bilangan eksponen natural (e)
17
18 #Mendefinisikan fungsi
19 def f(x):
20     return e**2*x-8*x**2
21
22 #Metode Secant
23 def Secant(x0,x1,eps,):
24     step = 1
25     condition = True
26     while condition:
27         if f(x0) == f(x1):
28             print("Solusi tidak di temukan")
29             break
30
31         x2 = x1 - ((f(x1)*(x1-x0))/(f(x1)-f(x0)))
32         print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, x2, f(x2)))
33         x0 = x1
34         x1 = x2
35         step = step + 1
36         condition = False
37     return x2
```

reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS D:\pandu\bhayangkara\tugas\Vietnum\Python\Pratikum\pratikum1-metnum> & C:/Python39/python.exe d:/pandu/bhayangkara/tugas/Metnum/Python/Pratikum/pratikum1-metnum/Latihan/latihanmetodesecant.py

Nama : Daffa Pandu Dewanata
Kelas : TIF3A6
NPM : 202010225242

x0: 0
x1: 2
epsilon: 0.00001
Iterasi-1, x = -0.27347081 dan f(x) = 1.68890612
Iterasi-2, x = -0.45710628 dan f(x) = 0.40028904
Iterasi-3, x = -0.51414908 dan f(x) = -0.1886336
9
Iterasi-4, x = -0.50268995 dan f(x) = 0.00386782
Iterasi-5, x = -0.50311410 dan f(x) = 0.00003472
Iterasi-6, x = -0.50311795 dan f(x) = -0.0000000
1

Akar Persamaan tersebut : -0.50311795
PS D:\pandu\bhayangkara\tugas\Vietnum\Python\Pratikum\pratikum1-metnum>

Gambar 1. Hasil Script Python Latihan Metode Secant

Script Latihan Metode Secant

```
# -*- coding: utf-8 -*-
"""
```

```
@author: Daffa Pandu Dewanata 202010225242 TF3A6
```

<https://github.com/daffapandu2304/pratikum1-metnum>

Nama : Daffa Pandu Dewanata
NPM : 202010225242
Kelas : TIF3A6

METODE NUMERIK

30-10-2021

"""

```
print ("")
print ("")
print ("Nama : Daffa Pandu Dewanata")
print ("Kelas : TIF3A6")
print ("NPM : 202010225242")
print ("")
print ("")
```

```
import numpy as np
import matplotlib.pyplot as plt
from math import e #untuk memanggil bilangan eksponen natural (e)
```

```
#Mendefinisikan fungsi
```

```
def f(x):
    return e**2**x-8*x**2
```

```
#Metode Secant
```

```
def Secant(x0,x1,eps,):
```

```
    step = 1
    condition = True
    while condition:
        if f(x0) == f(x1):
            print ('Solusi tidak di temukan')
            break
```

```
        x2 = x1 - ((f(x1)*(x1-x0))/(f(x1)-f(x0)))
        print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, x2, f(x2)))
        x0 = x1
        x1 = x2
        step = step+1
        condition = abs(f(x2)) > eps
        print('\n Akar Persamaan tersebut : %0.8f' % x2)
```

```
#Sesi Input Nilai Awal yang dikonversi ke pecahan
```

```
x0 = float(input('x0: '))
x1 = float(input('x1: '))
eps = float(input('epsilon: '))
Secant(x0,x1,eps,)
```