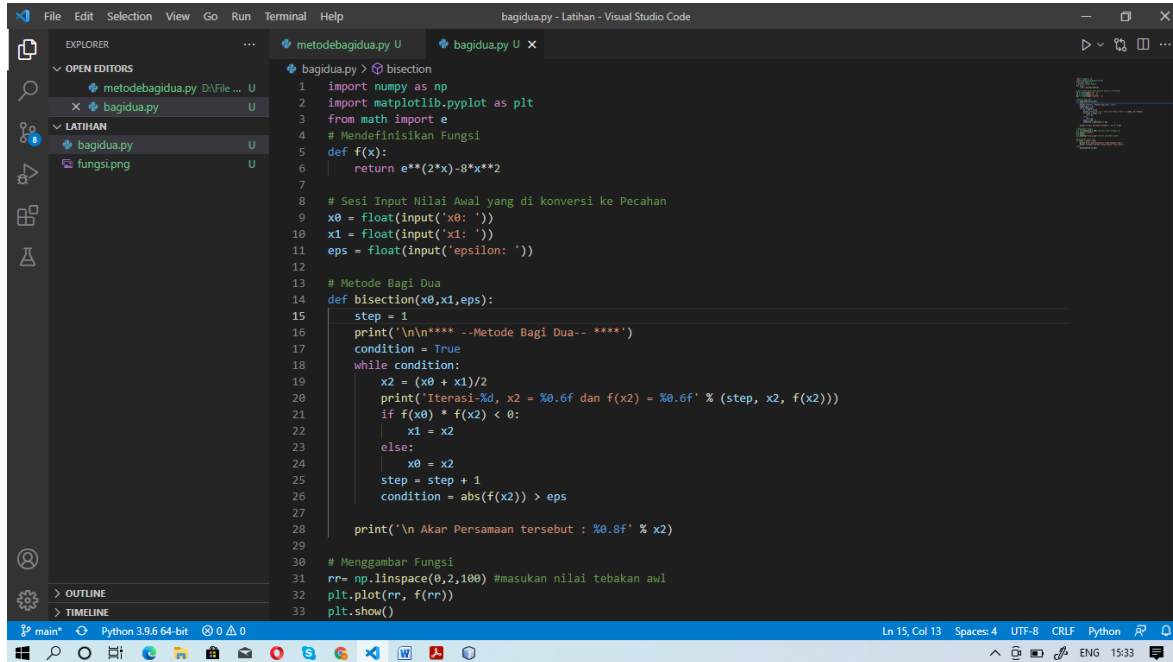


Nama : Muhamad Bagas Maulana
NPM : 202010225336
Kelas : TIF3A5

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

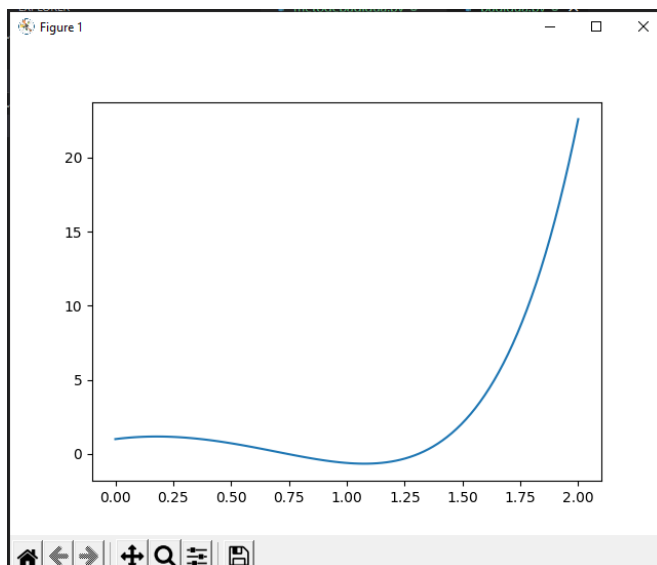
1. Bagi dua, gunakan $\epsilon = 0.00001$ dan tebakan awal $a = 0$ dan $b = 2$.

Source Code.

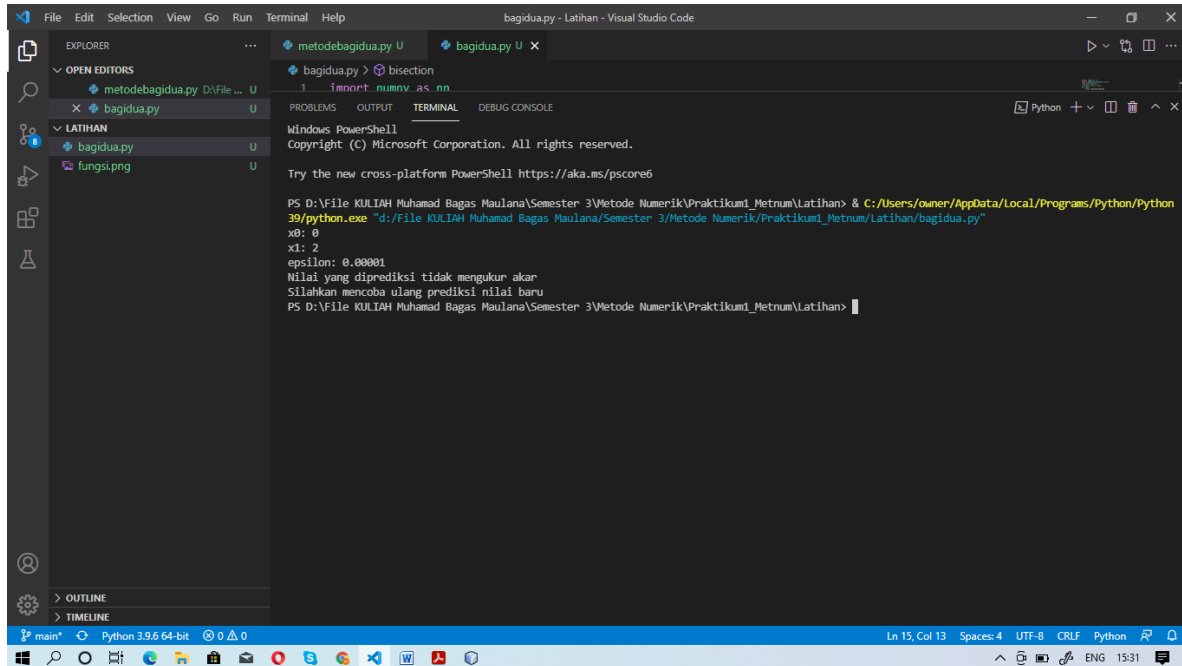


```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import e
4 # Mendefinisikan Fungsi
5 def f(x):
6     return e**(2*x)-8**x**2
7
8 # Sesi Input Nilai Awal yang di konversi ke Pecahan
9 x0 = float(input('x0: '))
10 x1 = float(input('x1: '))
11 eps = float(input('epsilon: '))
12
13 # Metode Bagi Dua
14 def bisection(x0,x1,eps):
15     step = 1
16     print('\n\n***** --Metode Bagi Dua-- *****')
17     condition = True
18     while condition:
19         x2 = (x0 + x1)/2
20         print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
21         if f(x0) * f(x2) < 0:
22             x1 = x2
23         else:
24             x0 = x2
25         step = step + 1
26         condition = abs(f(x2)) > eps
27     print('\n Akar Persamaan tersebut : %0.8f' % x2)
28
29
30 # Menggambar Fungsi
31 rr= np.linspace(0,2,100) #masukan nilai tebakan awl
32 plt.plot(rr, f(rr))
33 plt.show()
```

Grafik.



Hasil Output.



```
File Edit Selection View Go Run Terminal Help
bagidua.py - Latihan - Visual Studio Code

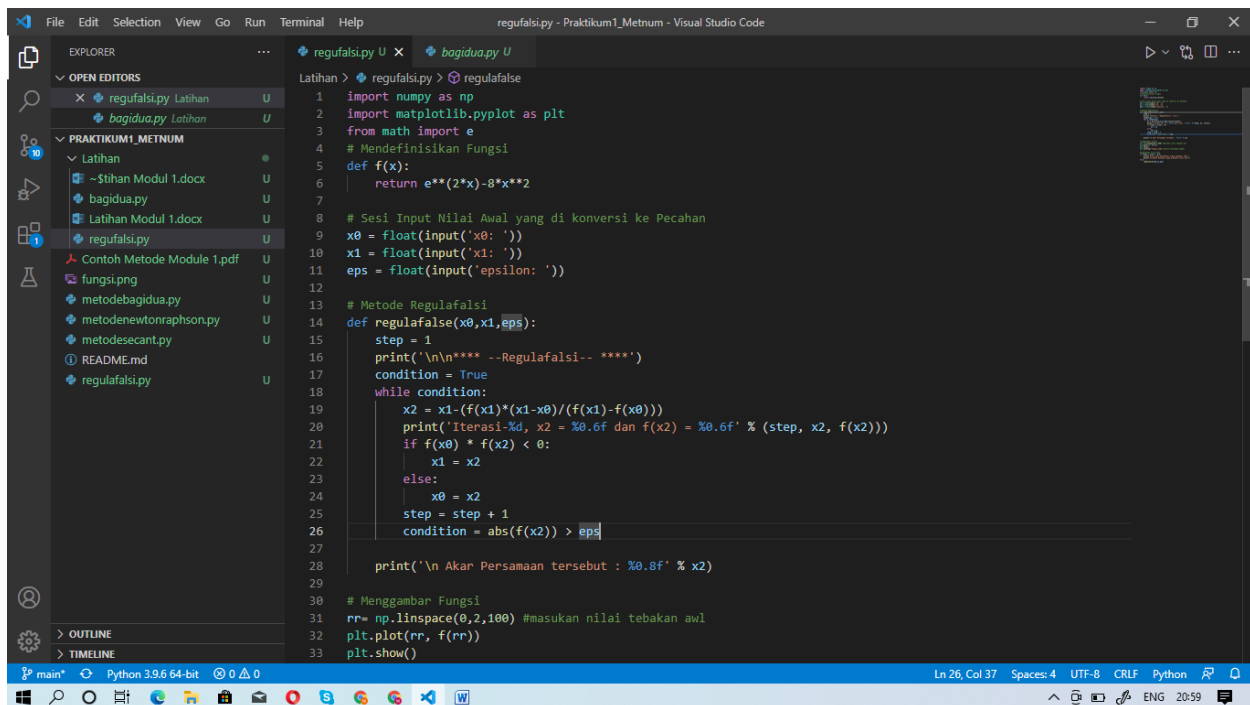
EXPLORER
metodebagidua.py D:\File ... U
bagidua.py U
LATIHAN
bagidua.py U
fungsi.png U

metodebagidua.py U
bagidua.py U

1 import numpy as np
2
3 def f(x):
4     return e**(2*x)-8*x**2
5
6 # Sesi Input Nilai Awal yang di konversi ke Pecahan
7 x0 = float(input('x0: '))
8 x1 = float(input('x1: '))
9 eps = float(input('epsilon: '))
10
11 # Metode Regulasi
12 def regulasi(x0,x1,eps):
13     step = 1
14     print('\n\n*** --Regulasi-- ***')
15     condition = True
16     while condition:
17         x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
18         print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
19         if f(x0) * f(x2) < 0:
20             x1 = x2
21         else:
22             x0 = x2
23         step = step + 1
24         condition = abs(f(x2)) > eps
25
26 print('\n Akan Persamaan tersebut : %0.8f' % x2)
27
28 # Menggambar Fungsi
29 rr= np.linspace(0,2,100) #masukan nilai tebakan awl
30 plt.plot(rr, f(rr))
31 plt.show()
```

2. Regula-falsi, gunakan $e = 0.00001$ dan tebakan awal $a = 0$ dan $b = 2$.

Source Code.

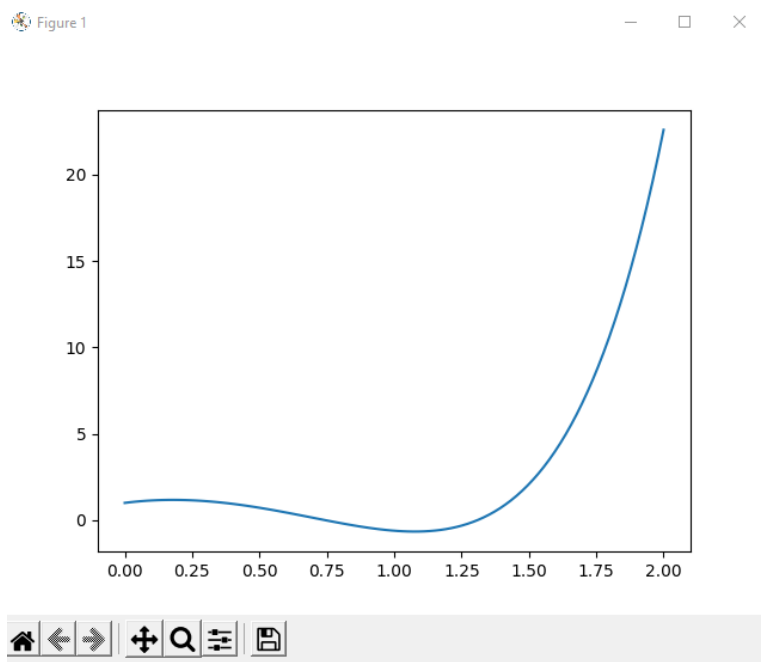


```
File Edit Selection View Go Run Terminal Help
regufalsi.py - Praktikum1_Metnum - Visual Studio Code

EXPLORER
regufalsi.py Latihan U
bagidua.py Latihan U
PRAKTIKUM1_METNUM
Latihan
~$tihan Modul 1.docx U
bagidua.py U
Latihan Modul 1.docx U
regufalsi.py U
Contoh Metode Module 1.pdf U
fungsi.png U
metodebagidua.py U
metodenewtonraphson.py U
metodesecant.py U
README.md
regufalsi.py U

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import e
4 # Mendefinisikan Fungsi
5 def f(x):
6     return e**(2*x)-8*x**2
7
8 # Sesi Input Nilai Awal yang di konversi ke Pecahan
9 x0 = float(input('x0: '))
10 x1 = float(input('x1: '))
11 eps = float(input('epsilon: '))
12
13 # Metode Regula-falsi
14 def regufalsi(x0,x1,eps):
15     step = 1
16     print('\n\n*** --Regula-falsi-- ***')
17     condition = True
18     while condition:
19         x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
20         print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
21         if f(x0) * f(x2) < 0:
22             x1 = x2
23         else:
24             x0 = x2
25         step = step + 1
26         condition = abs(f(x2)) > eps
27
28 print('\n Akan Persamaan tersebut : %0.8f' % x2)
29
30 # Menggambar Fungsi
31 rr= np.linspace(0,2,100) #masukan nilai tebakan awl
32 plt.plot(rr, f(rr))
33 plt.show()
```

Grafik



Hasil Output.

```
regufalsi.py - Praktikum1_Metnum - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    regufalsi.py Latihan U
    bagidua.py Latihan U
  PRAKTIKUM1_METNUM
    Latihan
    ~Stihan Modul 1.docx U
    bagidua.py U
    Latihan Modul 1.docx U
    regufalsi.py U
    Contoh Metode Module 1.pdf U
    fungsi.png U
    metodebagidua.py U
    metodewtonraphson.py U
    metodesecant.py U
    README.md U
    regufalsi.py U

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

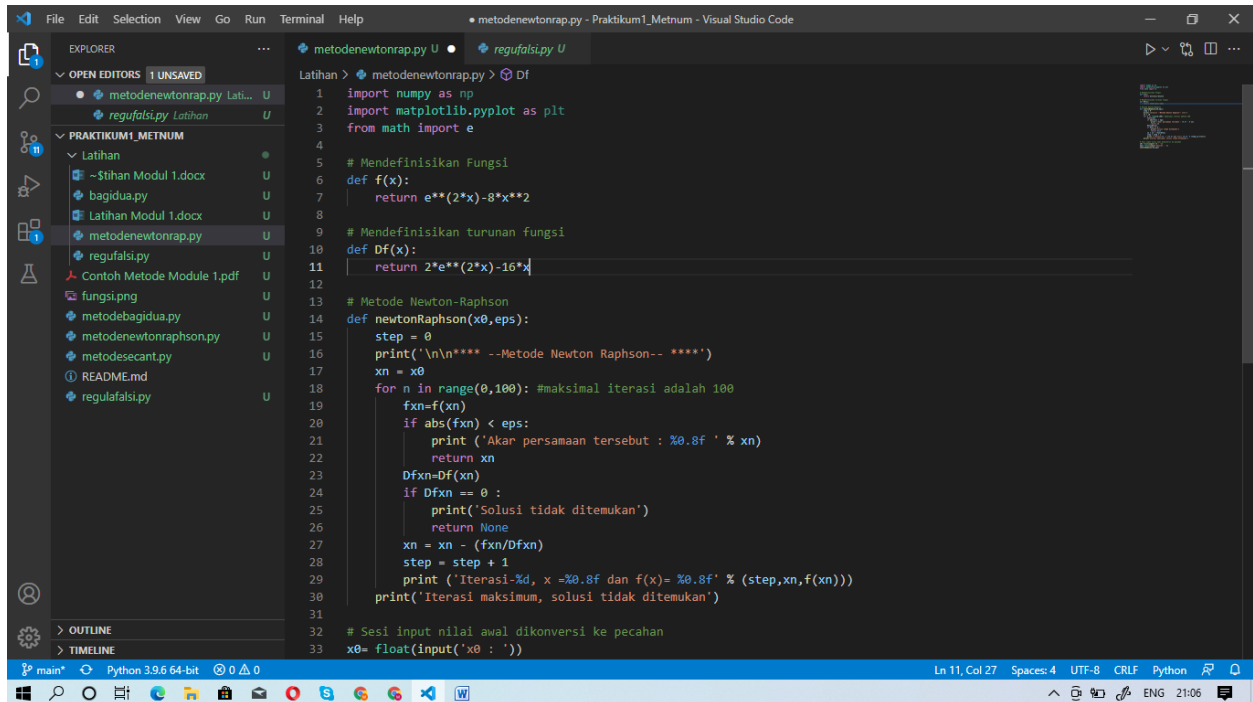
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum> & C:/Users/owner/AppData/Local/Programs/Python/Python39/python.exe "d:/File KULIAH Muhamad Bagas Maulana/Semester 3/Metode Numerik/Praktikum1_Metnum/Latihan/regufalsi.py"
x0: 0
x1: 2
epsilon: 0.00001
Nilai yang diprediksi tidak mengukur akar
Silahkan mencoba ulang prediksi nilai baru
PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum>
```

3. Newton Raphson, gunakan $e = 0.00001$ dan tebakan awal $0 \leq x \leq 0$.

Source Code.

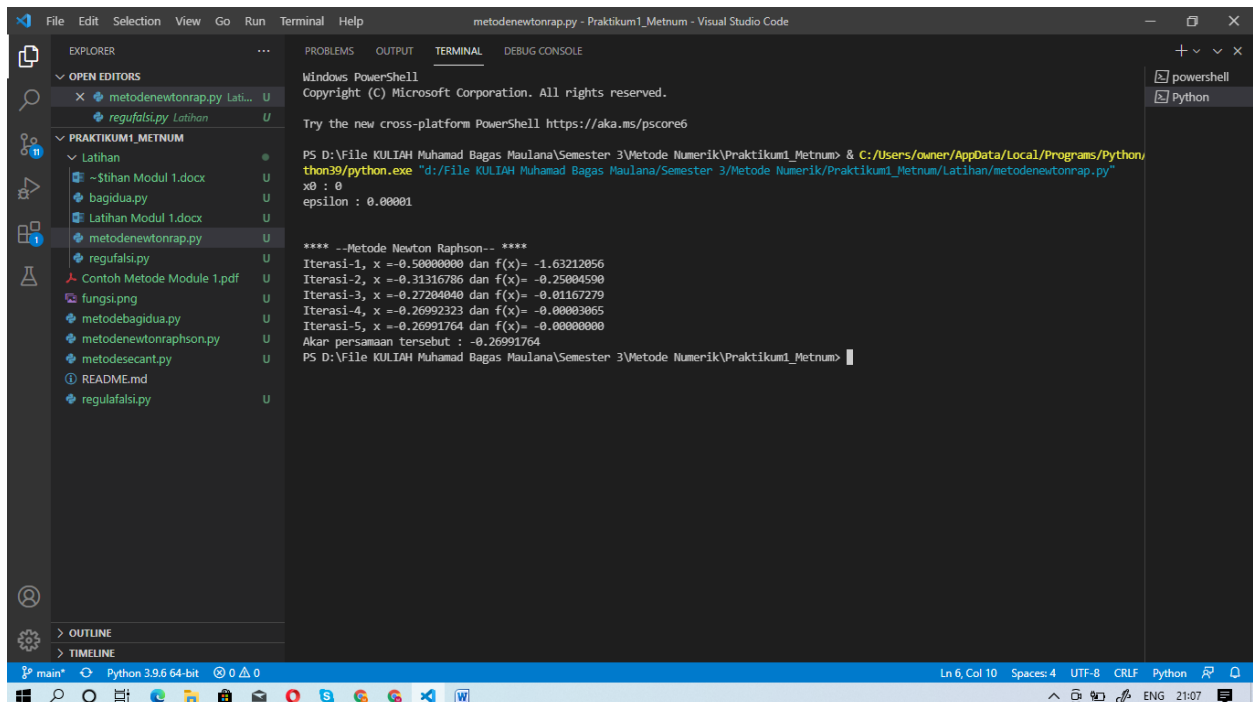


```
File Edit Selection View Go Run Terminal Help
metodenewtonrap.py - Praktikum1_Metnum - Visual Studio Code

EXPLORER
metodenewtonrap.py U
regufalsi.py Latihan U
PRAKTIKUM1_METNUM
Latihan
~$tihan Modul 1.docx U
bagidua.py U
Latihan Modul 1.docx U
metodenewtonrap.py U
regufalsi.py U
Contoh Metode Module 1.pdf U
fungsi.png U
metodebagidua.py U
metodenewtonraphson.py U
metodesecant.py U
README.md U
regulafalsi.py U

metodenewtonrap.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import e
4
5 # Mendefinisikan Fungsi
6 def f(x):
7     return e**(2*x)-8*x**2
8
9 # Mendefinisikan turunan fungsi
10 def Df(x):
11     return 2*e**(2*x)-16*x
12
13 # Metode Newton-Raphson
14 def newtonRaphson(x0,eps):
15     step = 0
16     print('\n\n**** --Metode Newton Raphson-- ****')
17     xn = x0
18     for n in range(0,100): #maksimal iterasi adalah 100
19         fxn=f(xn)
20         if abs(fxn) < eps:
21             print('Akar persamaan tersebut : %0.8f ' % xn)
22             return xn
23         Dfxn=Df(xn)
24         if Dfxn == 0 :
25             print('Solusi tidak ditemukan')
26             return None
27         xn = xn - (fxn/Dfxn)
28         step = step + 1
29         print('Iterasi-%d, x =%0.8f dan f(x)= %0.8f' % (step,xn,f(xn)))
30     print('Iterasi maksimum, solusi tidak ditemukan')
31
32 # Sesi input nilai awal dikonversi ke pecahan
33 x0= float(input('x0 : '))
```

Hasil Output.



```
File Edit Selection View Go Run Terminal Help
metodenewtonrap.py - Praktikum1_Metnum - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

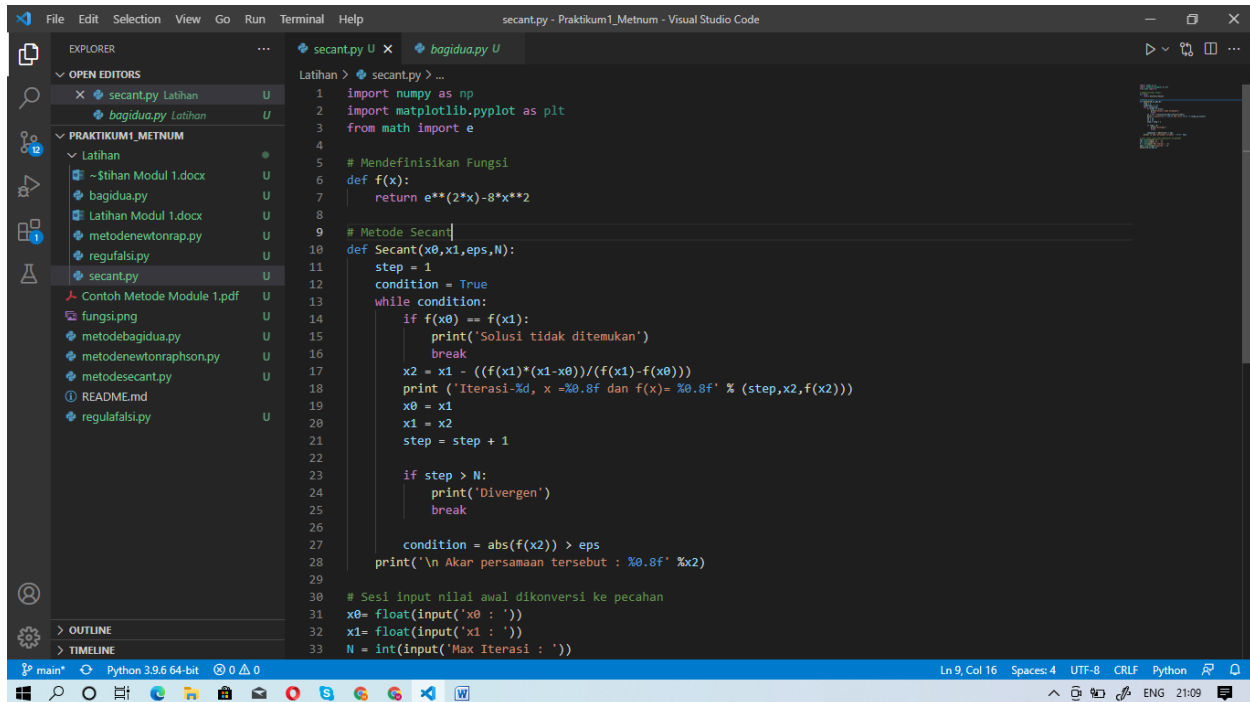
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum> & C:/Users/owner/AppData/Local/Programs/Python/Python39/python.exe "d:/File KULIAH Muhamad Bagas Maulana/Semester 3/Metode Numerik/Praktikum1_Metnum/Latihan/metodenewtonrap.py"
x0 : 0
epsilon : 0.00001

**** --Metode Newton Raphson-- ****
Iterasi-1, x =-0.50000000 dan f(x)= -1.63212056
Iterasi-2, x =-0.31316786 dan f(x)= -0.25004590
Iterasi-3, x =-0.27204040 dan f(x)= -0.01167279
Iterasi-4, x =-0.26992323 dan f(x)= -0.00003065
Iterasi-5, x =-0.26991764 dan f(x)= -0.00000000
Akar persamaan tersebut : -0.26991764
PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum>
```

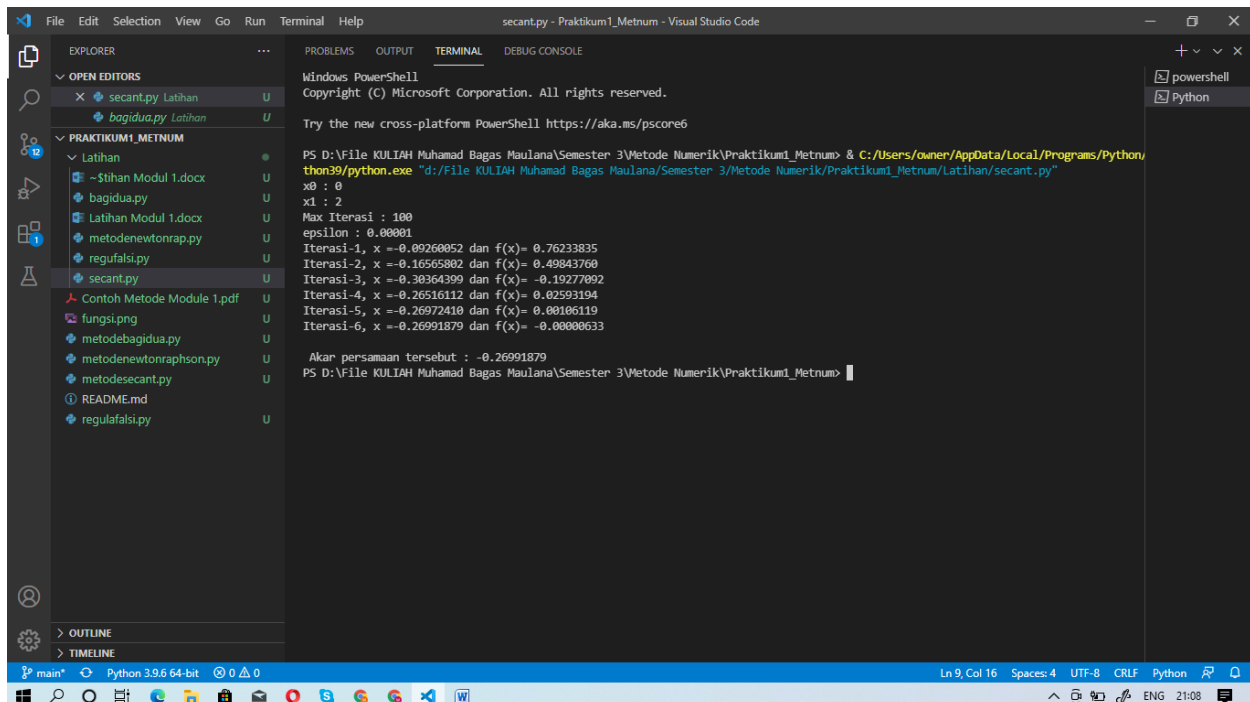
4. Secant, gunakan $\epsilon = 0.00001$ dan tebakan awal $x_0 = 0$ dan $x_1 = 2$.

Source Code.



```
secant.py - Praktikum1_Metnum - Visual Studio Code
File Edit Selection View Go Run Terminal Help
secant.py U bagidua.py U
EXPLORER
  OPEN EDITORS
    secant.py Latihan U
    bagidua.py Latihan U
  PRAKTIKUM1_METNUM
    Latihan
    -Stihan Modul 1.docx U
    bagidua.py U
    Latihan Modul 1.docx U
    metodenewtonrap.py U
    regufalsi.py U
    secant.py U
    Contoh Metode Module 1.pdf U
    fungsi.png U
    metodebagidua.py U
    metodenewtonraphson.py U
    metodeseccant.py U
    README.md
    regufalsi.py U
  OUTLINE
  TIMELINE
  main* Python 3.9.6 64-bit 0 0 0
Ln 9, Col 16 Spaces: 4 UTF-8 CRLF Python
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import e
4
5 # Mendefinisikan Fungsi
6 def f(x):
7     return e**(2*x)-8*x**2
8
9 # Metode Secant
10 def Secant(x0,x1,eps,N):
11     step = 1
12     condition = True
13     while condition:
14         if f(x0) == f(x1):
15             print('Solusi tidak ditemukan')
16             break
17         x2 = x1 - ((f(x1)*(x1-x0))/(f(x1)-f(x0)))
18         print('Iterasi-%d, x =%.8f dan f(x)= %.8f' % (step,x2,f(x2)))
19         x0 = x1
20         x1 = x2
21         step = step + 1
22
23     if step > N:
24         print('Divergen')
25         break
26
27     condition = abs(f(x2)) > eps
28     print('\n Akar persamaan tersebut : %.8f' %x2)
29
30 # Sesi input nilai awal dikonversi ke pecahan
31 x0= float(input('x0 : '))
32 x1= float(input('x1 : '))
33 N = int(input('Max Iterasi : '))
```

Hasil Output.



```
secant.py - Praktikum1_Metnum - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

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

PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum> & C:/Users/owner/AppData/Local/Programs/Python/Python39/Python.exe "d:/File KULIAH Muhamad Bagas Maulana/Semester 3/Metode Numerik/Praktikum1_Metnum/Latihan/secant.py"
x0 : 0
x1 : 2
Max Iterasi : 100
epsilon : 0.00001
Iterasi-1, x =-0.09260052 dan f(x)= 0.76233835
Iterasi-2, x =-0.16565802 dan f(x)= 0.49843760
Iterasi-3, x =-0.30364399 dan f(x)= -0.19277092
Iterasi-4, x =-0.26516112 dan f(x)= 0.02593194
Iterasi-5, x =-0.26972410 dan f(x)= 0.00106119
Iterasi-6, x =-0.26991879 dan f(x)= -0.00000633

Akar persamaan tersebut : -0.26991879
PS D:\File KULIAH Muhamad Bagas Maulana\Semester 3\Metode Numerik\Praktikum1_Metnum>
```

5. Berdasarkan perhitungan 1 hingga 4, metode mana yang terbaik, dan berikan alasannya.

➔ Menurut saya metode Newton-Raphson yang terbaik karena dapat menunjukkan hasil dari soal tersebut dan memiliki nilai error yang kecil.