

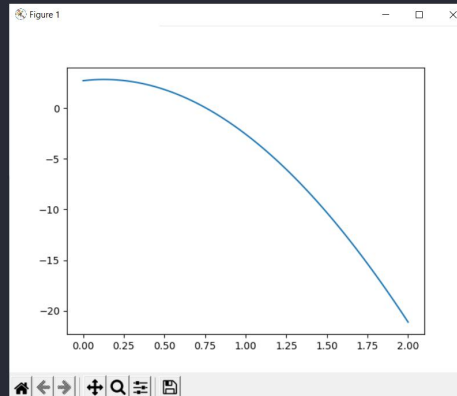
1. Metode Bagi Dua

```
biseksi.py > f
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 # sesi input nilai awal yang di konversi ke pecahan
10 x0 = float(input('x0: '))
11 x1 = float(input('x1: '))
12 eps = float(input('epsilon: '))
13
14 # metode bagi dua
15 def bisection(x0, x1, eps):
16     step = 1
17     print('\n\n*** --Metode Bagi Dua-- ***')
18     condition = True
19     while condition:
20         x2 = (x0 + x1)/2
21         print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
22         if f(x0) * f(x2) < 0:
23             x1 = x2
24         else:
25             x0 = x2
26         step = step + 1
27         condition = abs(f(x2)) > eps
28
29     print('\n Akar persamaan tersebut : %0.8f' % x2)
30
31 # pengecekan nilai awal
32 if f(x0) * f(x1) > 0.0:
33     print('Nilai yang di prediksi tidak mengurung akar')
34     print('Silahkan mencoba ulang prediksi nilai baru')
35 else:
36     bisection(x0, x1, eps)
37
38 # menggambar fungsi
39 rr = np.linspace(0, 2, 100) #masukkan nilai tebakan awal
40 plt.plot(rr, f(rr))
41 plt.show()
42 plt.savefig("fungsi.png") #untuk menyimpan gambar fungsi
43
```

```
py
/cygdrive/e/Data/Tri/Kuliah/Metode Numerik/Tugas/praktikum1 master ? py biseksi.py
x0: 0
x1: 2
epsilon: 0.00001
```

```
*** --Metode Bagi Dua-- ***
Iterasi-1, x2 = 1.000000 dan f(x2) = -2.563436
Iterasi-2, x2 = 0.500000 dan f(x2) = 1.844231
Iterasi-3, x2 = 0.750000 dan f(x2) = 0.071507
Iterasi-4, x2 = 0.625000 dan f(x2) = -1.139649
Iterasi-5, x2 = 0.812500 dan f(x2) = -0.247262
Iterasi-6, x2 = 0.781250 dan f(x2) = -0.211121
Iterasi-7, x2 = 0.765625 dan f(x2) = -0.068805
Iterasi-8, x2 = 0.757812 dan f(x2) = 0.002172
Iterasi-9, x2 = 0.761719 dan f(x2) = -0.032851
Iterasi-10, x2 = 0.759766 dan f(x2) = -0.015313
Iterasi-11, x2 = 0.758709 dan f(x2) = -0.005564
Iterasi-12, x2 = 0.758381 dan f(x2) = -0.002195
Iterasi-13, x2 = 0.758057 dan f(x2) = -0.000811
Iterasi-14, x2 = 0.757935 dan f(x2) = 0.001881
Iterasi-15, x2 = 0.757996 dan f(x2) = 0.000335
Iterasi-16, x2 = 0.758026 dan f(x2) = 0.000262
Iterasi-17, x2 = 0.758041 dan f(x2) = 0.000126
Iterasi-18, x2 = 0.758049 dan f(x2) = 0.000057
Iterasi-19, x2 = 0.758053 dan f(x2) = 0.000023
Iterasi-20, x2 = 0.758055 dan f(x2) = 0.000006

Akar persamaan tersebut : 0.75805473
```



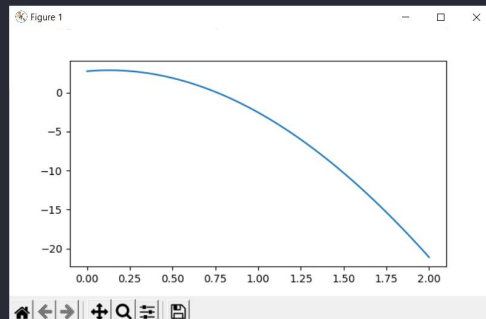
2. Metode Regula-falsi

```
regulafalsi.py > f
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import e
4
5 #mendefinisikan fungsi
6
7
8 def f(x):
9     return e**2*x-8*x**2
10
11
12 # sesi input nilai awal yang di konversi ke pecahan
13 x0 = float(input('x0: '))
14 x1 = float(input('x1: '))
15 eps = float(input('epsilon: '))
16
17 # metode regulafalse
18 def regulafalse(x0, x1, eps):
19     step = 1
20     print('\n\n*** --Metode Regulafalse-- ***')
21     condition = True
22     while condition:
23         x2 = x1-(f(x1)*(x1-x0)/(f(x1)-f(x0)))
24         print('Iterasi-%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))
25         if f(x0) * f(x2) < 0:
26             x1 = x2
27         else:
28             x0 = x2
29         step = step + 1
30         condition = abs(f(x2)) > eps
31
32     print('\n Akar persamaan tersebut : %0.8f' % x2)
33
34
35 # pengecekan nilai awal
36 if f(x0) * f(x1) > 0.0:
37     print('Nilai yang di prediksi tidak mengurung akar')
38     print('Silahkan mencoba ulang prediksi nilai baru')
39 else:
40     regulafalse(x0, x1, eps)
41
42 # menggambar fungsi
43 rr = np.linspace(0, 2, 100) # masukkan nilai tebakan awal
44 plt.plot(rr, f(rr))
45 plt.show()
46 plt.savefig("fungsi.png") # untuk menyimpan gambar fungsi
47
```

```
C:\pydrive\Drive\Tri\Kuliah\Metode Numerik\Tugas\praktikum1 master 7 py regulafalsi.py
x0: 0
x1: 2
epsilon: 0.00001
```

```
*** --Metode Regulafalse-- ***
Iterasi-1, x2 = 0.227994 dan f(x2) = 2.767815
Iterasi-2, x2 = 0.433253 dan f(x2) = 2.168763
Iterasi-3, x2 = 0.579113 dan f(x2) = 1.377959
Iterasi-4, x2 = 0.666113 dan f(x2) = 0.763699
Iterasi-5, x2 = 0.712648 dan f(x2) = 0.391888
Iterasi-6, x2 = 0.736888 dan f(x2) = 0.193111
Iterasi-7, x2 = 0.747536 dan f(x2) = 0.093384
Iterasi-8, x2 = 0.753043 dan f(x2) = 0.044648
Iterasi-9, x2 = 0.755673 dan f(x2) = 0.021267
Iterasi-10, x2 = 0.756924 dan f(x2) = 0.010107
Iterasi-11, x2 = 0.757219 dan f(x2) = 0.004799
Iterasi-12, x2 = 0.757881 dan f(x2) = 0.002277
Iterasi-13, x2 = 0.757935 dan f(x2) = 0.001080
Iterasi-14, x2 = 0.757998 dan f(x2) = 0.000512
Iterasi-15, x2 = 0.758028 dan f(x2) = 0.000243
Iterasi-16, x2 = 0.758043 dan f(x2) = 0.000115
Iterasi-17, x2 = 0.758049 dan f(x2) = 0.000055
Iterasi-18, x2 = 0.758053 dan f(x2) = 0.000026
Iterasi-19, x2 = 0.758054 dan f(x2) = 0.000012
Iterasi-20, x2 = 0.758055 dan f(x2) = 0.000006
```

```
Akar persamaan tersebut : 0.75805477
```



3. Metode Newton Raphson

```
newton-raphson.py > ...
1  from math import e
2
3  #mendefinisikan fungsi
4
5
6  def f(x):
7      return e*2**x-8*x**2
8
9  def Df(x):
10     return e*2**x-16*x
11
12 # metode newton rapshon
13 def newtonRapshon(x0, eps):
14     step = 0
15     print('\n\n** --Metode Newton Raphson-- **')
16     xn = x0
17     for n in range(0,100): # maksimal iterasi adalah 100
18         fxn=f(xn)
19         if abs(fxn) < eps:
20             print('\n Akar Persamaan tersebut : %0.8f' % xn)
21             return xn
22         Dfxn = Df(xn)
23         if Dfxn == 0:
24             print('Solusi tidak ditemukan')
25             return None
26         xn = xn - (fxn/Dfxn)
27         step = step + 1
28         print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, xn, f(xn)))
29     print('Iterasi maksimum, solusi tidak ditemukan')
30
31
32 # sesi input nilai awal yang di konversi ke pecahan
33 x0 = float(input('x0: '))
34 eps = float(input('epsilon: '))
35 newtonRapshon(x0, eps)
```

```

newton-raphson.py > ...
1  from math import e
2
3  #mendefinisikan fungsi
4
5
6  def f(x):
7      return e*2**x-8*x**2
8
9  def Df(x):
10     return e*2**x-16*x
11
12 # metode newton rapshon
13 def newtonRapshon(x0, eps):
14     step = 0
15     print('\n\n*** --Metode Newton Raphson-- ***')
16     xn = x0
17     for n in range(0,100): # maksimal iterasi adalah 100
18         fxn=f(xn)
19         if abs(fxn) < eps:
20             print('\n Akar Persamaan tersebut : %0.8f' % xn)
21             return xn
22         Dfxn = Df(xn)
23         if Dfxn == 0:
24             print('Solusi tidak ditemukan')
25             return None
26         xn = xn - (fxn/Dfxn)
27         step = step + 1
28         print('Iterasi-%d, x = %0.8f dan f(x) = %0.8f' % (step, xn, f(xn)))
29     print('Iterasi maksimum, solusi tidak ditemukan')
30
31
32 # sesi input nilai awal yang di konversi ke pecahan
33 x0 = float(input('x0: '))
34 eps = float(input('epsilon: '))
35 newtonRapshon(x0, eps)

```

4. Metode Secant

```

/cydrive/e/Data Tri/Kuliah/Metode Numerik/Tugas/praktikum1 master ? py secant.py
x0: 0
x1: 2
Max Iter: 100
epsilon: 0.00001

*** --Metode Secant-- ***
Iterasi-1, x = 0.22799448 dan f(x) = 2.767015
Iterasi-2, x = 0.4325202 dan f(x) = 2.168763
Iterasi-3, x = 1.17635408 dan f(x) = -4.927021
Iterasi-4, x = 0.66837567 dan f(x) = 0.887465
Iterasi-5, x = 0.73302934 dan f(x) = 0.219478
Iterasi-6, x = 0.76014758 dan f(x) = -0.018739
Iterasi-7, x = 0.75801428 dan f(x) = 0.000368
Iterasi-8, x = 0.75805535 dan f(x) = 0.000001
Akar persamaan tersebut : 0.75805535
/cydrive/e/Data Tri/Kuliah/Metode Numerik/Tugas/praktikum1 master ?

secant.py > Df
1  from math import e
2
3  # mendefinisikan fungsi
4  def f(x):
5      return e*2**x-8*x**2
6
7
8  def Df(x):
9      return e*2**x-16*x
10
11 # metode secant
12 def Secant(x0, x1, eps, N):
13     step = 1
14     print('\n\n*** --Metode Secant-- ***')
15     condition = True
16     while condition:
17         if f(x0) == f(x1):
18             print('Solusi tidak ditemukan')
19             break
20
21         x2 = x1 - ( ( f(x1)*(x1-x0) ) / (f(x1)-f(x0)) )
22         print('Iterasi-%d, x = %0.8f dan f(x) = %0.6f' % (step, x2, f(x2)))
23         x0 = x1
24         x1 = x2
25         step = step + 1
26
27         if step > N:
28             print('Divergen')
29             break
30
31         condition = abs(f(x2)) > eps
32         print('\n Akar persamaan tersebut : %0.8f' % x2)
33
34
35
36 # sesi input nilai awal yang di konversi ke pecahan
37 x0 = float(input('x0: '))
38 x1 = float(input('x1: '))
39 N = int(input('Max Iter: '))
40 eps = float(input('epsilon: '))
41 Secant(x0, x1, eps, N)
42
```

5. Metode terbaik itu metode bagi dua, karena sederhana