**Tugas Modul I**

**Solusi Persamaan Nonlinear**

****

Dibuat Oleh :

Nashirudin Baqiy

24060119130045

Asisten Praktikum :

Muhammad Rizqi Arya Pradana

Ibnu Nahwitama

**DEPARTEMEN INFORMATIKA**

**FAKULTAS SAINS DAN MATEMATIKA**

**UNVERSITAS DIPONEGORO**

**2020**

# BAB I PENDAHULUAN

## Tujuan

1. Dapat menghitung akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson dan metode Secant
2. Mencari besarnya kesalahan dari suatu perhitungan akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson, dan metode Secant

## Rumusan Permasalahan

1. Diberikan fungsi 𝑓(𝑥) = 𝑒 𝑥 + 𝑥 2 − 3𝑥 − 2 = 0 terdapat sebuah akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Biseksi dengan toleransi kesalahan 1e-5
2. Diberikan fungsi 𝑓(𝑥) = 𝑒 𝑥 + 𝑥 2 − 3𝑥 − 2 = 0 mempunyai akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Newton Raphson dengan toleransi kesalahan 1e-5.

# BAB II DASAR TEORI

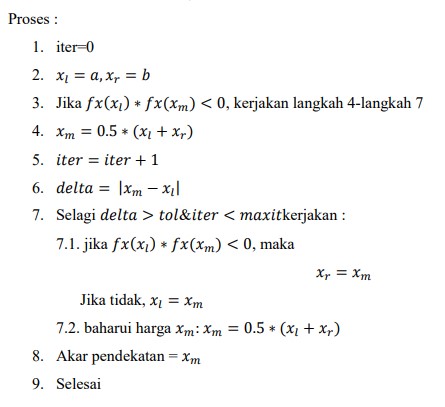
## 2.1 Pendahuluan Teori

Pencarian akar(penyelesaian) suatu persamaan non-linear, , adalah mencari suatu harga , yang apabila disubstitusikan ke dalam persamaan itu, akan memberikan harga fungsi nol. Secara matematika

Sebagai contoh, akar dari persamaan adalah 1.763223 karena, apabila harga 𝑥 ∗ = 1.763223 disubstitusikan ke dalam persamaan itu, harga fungsi itu menjadi nol. Metode-metode numeric yang banyak digunakan untuk menyelesaikan persamaan non-linear adalah metode biseksi, metode regula falsi, secant, newton -Raphson dan titik tetap. Tiga metode akan dibicarakan di sini adalah metode biseksi , metode Newton-Raphson dan metode Secant

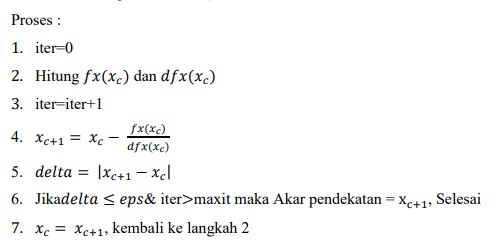
## 2.2 Metode Biseksi

Dalam metode Biseksi, interval yang mengandung akar dibagi menjadi dua secara berurutan hingga ukuran interval mengecil dan akhirnya mencapai harga toleransi kesalahan yang diinginkan. Dalam interval [a,b] terdapat sebuah akar (yang akan dicari), apabila dipenuhi :



## 2.3 Metode Newton-Raphson

Metode Newton Raphson, dalam mencari akar suatu fungsi nonlinear memerlukan evaluasi harga fungsi dan turunannya pada sembarang titik x yang merupakan harga awal tebakan akar fungsi tersebut. Metode ini didasarkan atas perluasan deret Taylor di sekitar suatu titik. Karenanya, apabila harga awal tebakan jauh dari akar sebenarnya, konvergensi akan lambat atau mungkin tidak dicapai sama sekali.



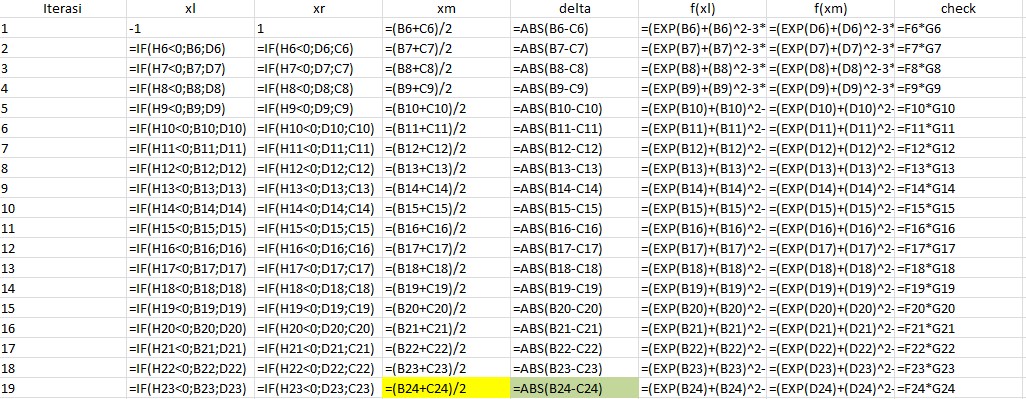
# BAB III PEMBAHASAN

## 3.1 Penyelesaian dengan Metode Biseksi

Syarat dari Metode Biseksi ialah maka harus . Selanjut-selanjutnya disebut dengan check ditabel.

**3.1.1 Dengan Excel**

Isi



Hasil



**3.1.2 Dengan Python**

Source Code

|  |
| --- |
| # BISECTION METHOD  # 24060119130045 - Nashirudin Baqiy  def bisect(f,a,b,tol=0.00001,maxiter=50):  iterator = 0    xl=a  xr=b  xm=(xl + xr) / 2  if f(a)\*f(b)==0:  if f(a)==0:  return a  elif f(b)==0:  return b  elif f(a)\*f(b)>0:  raise ValueError('Nilai f(a) dan f(b) harus mempunyai tanda yang berbeda')  while iterator < maxiter:  if f(xl)\*f(xm)<0:  xr = xm    elif f(xl)\*f(xm)>0:  xl = xm    elif f(xl)\*f(xm)==0:  return xm  xm =(xl+xr)/2  iterator=iterator+1  delta=abs(xr-xl)  if delta<tol:  return xm  return xm  def persamaan(x):  import math  return math.e\*\*x + x\*\*2 - 3\*x - 2  print('Akarnya dengan Biseksi ='+str(bisect(persamaan,-1,1))) |

Hasil

F:\TUGAS\Praktikum\Metnum\Tugas1\ss\hasil python biseksi fix.jpg

**3.1.3 Penjelasan**

Kedua cara diatas baik excel maupun python menggunakan langkah-langkah yang sama yang telah dilampirkan di BAB II DASAR TEORI. Iter 0 yaitu mengecek apakah memenuhi syarat menggunakan metode biseksi atau tidak dengan cara . Jika memenuhi, maka masuk ke langkah 4 dst dari iter 1 selagi delta > tol & iter < maxit. Jika pada iter 1, maka dan jika bukan kemudian diperbaru begitu juga dengan deltanya. Setelah stop maka didapat akarnya yaitu .

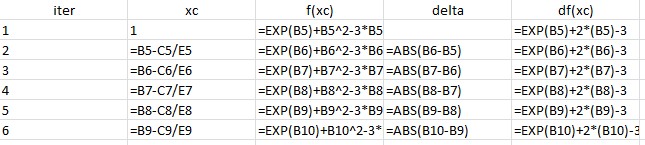
Akar pendekatannya : -0.390266

Dengan Toleransi : 7.63e-6

## 3.2 Penyelesaian dengan Metode Newton-Raphson

**3.2.1 Dengan Excel**

Isi



Hasil

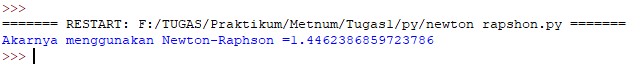


**3.2.2 Dengan Python**

Source Code

|  |
| --- |
| # NEWTON-RAPHSON METHOD  # 24060119130045 - Nashirudin Baqiy  from math import \*  def newton\_raphson(f,fdef,x0,eps=0.00001,maxiter=100):  iterator=0  while iterator < maxiter:  xc = f(x0)  xcdef = fdef(x0)  iterator=iterator+1  x1 = x0- (xc/xcdef)  delta = abs(x1-x0)  if delta<=eps:  return x1  x0 = x1  return x0  def persamaan(x):  return e\*\*x + x\*\*2 - 3\*x - 2  def turunan(x):  return e\*\*x + 2\*x - 3  print('Akarnya menggunakan Newton-Raphson =  '+str(newton\_raphson(persamaan,turunan,1))) |

Hasil



**3.1.3 Penjelasan**

Mencari f(xc) dan df(xc) terlebih dahulu, kemudian dilanjutkan xc menjadi dan mencari delta dari absolut . Langkah terus dilakukan hingga kondisi berhenti yaitu delta eps & iter > maxit. Jika berhenti, didapatkan akar pendekatan yaitu .

Akar pendekatannya : 1.4462

Dengan Toleransi : 2.81e-6

# BAB IV PENUTUPAN

## 4.1 Kesimpulan

Akar yang terdapat dari suatu persamaan nonlinear dapat dicari dengan Metode *Bisection* (Biseksi) dan Metode Newton-Raphson. Langkah-langkah yang tersedia dapat dibuat dengan menggunakan Bahasa *Python* atau menggunakan Spreadsheet seperti excel sehingga tidak perlu melakukan iterasi secara sistem manual seperti yang ada pada *proses* di atas. Hasil kerja dari sistem program tersebut adalah sebagai berikut.

1. Pada soal pertama tentang implementasi Metode *Bisection*, dicari akar dari persamaan dengan batas kiri -1 dan batas kanan 1. Hasil yang diperoleh menggunakan metode biseksi yang telah dibuat dalam Bahasa *Python* adalah -0.39026641845703125. Excel juga menghasilkan nilai yang sama namun ditampilkan dalam pembulatan.
2. Pada soal kedua tentang implementasi Metode Newton-Raphson, dicari akar dari persamaan yang memiliki turunan dan harga taksiran 1. Hasil yang diperoleh menggunakan Metode Newton-Raphson yang telah dibuat dalam Bahasa *Python* adalah 1.4462386859723786. Excel juga menghasilkan nilai yang sama namun ditampilkan dalam pembulatan.