Tugas Pemrograman 4

Relasi Antar Class dalam Permainan Ular Tangga



Nama :

Galih Aulia Al Hakim

4210161028

D4 TEKNOLOGI GAME

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

SURABAYA

2018

**METODE TABEL**

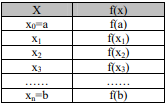
1. **DASAR TEORI**

Penyelesaian persamaan non linier adalah penentuan akar-akar persamaan non linier.Dimana akar sebuah persamaan f(x) =0 adalah nilai-nilai x yang menyebabkan nilai f(x) sama dengan nol. Dengan kata lain akar persamaan f(x) adalah titik potong antara kurva f(x) dan sumbu X.

Theorema 1.1.

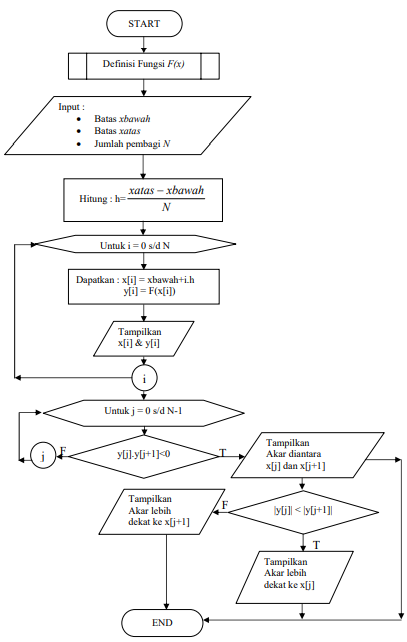
Suatu range x=[a,b] mempunyai akar bila f(a) dan f(b) berlawanan tanda atau memenuhi f(a).f(b)<0

Secara sederhana, untuk menyelesaikan persamaan non linier dapat dilakukan dengan menggunakan metode table atau pembagian area.Dimana untuk x = [ ] a,b atau x di antara a dan b dibagi sebanyak N bagian dan pada masing-masing bagian dihitung nilai f(x) sehingga diperoleh tabel :



Dari tabel ini, bila ditemukan f(xk)=0 atau mendekati nol maka dikatakan bahwa xk adalah penyelesaian persamaan f(xk)=0.Bila tidak ada f(xk) yang sama dengan nol, maka dicari nilai f(xk) dan f(xk+1) yang berlawanan tanda, bila tidak ditemukan maka dikatakan tidak mempunyai akar untuk x = [a,b], dan bila ditemukan maka ada 2 pendapat untuk menentukan akar persamaan, yaitu :

1. Akar persamaan ditentukan oleh nilai mana yang lebih dekat, bila |f(xk)|≤ |f(xk+1)| maka akarnya xk, dan bila |f(xk+1)|
2. Akarnya perlu di cari lagi, dengan range x = [ ] 1 , k k+ x x .
3. **ALGORITMA**
4. Defisikan fungsi f(x)
5. Tentukan range untuk x yang berupa batas bawah xbawahdan batas atas xatas
6. Tentukan jumlah pembagian N
7. Hitung step pembagi h H = N x x atas – bawah
8. Untuk i = 0 s/d N, hitung xi = xbawah + i.h yi = f(xi)
9. Untuk I = 0 s/d N dicari k dimana \*. Bila f(xk) = 0 maka xk adalah penyelesaian \*. Bila f(xk).f(xk+1) < 0 maka : - Bila |f(xk)|
10. **FLOWCHART**



1. **LIST PROGRAM**

#include <stdio.h>

#include <math.h>

float b,a,n,h,e,x[100],y[100];

int i,j;

float f(float x)

{

return(exp(-x)-x);

}

void tabel()

{

puts("\tMetode Tabel");

puts("\t==================\n");

printf("\t Fungsi\t= F(x)=exp(-x)-x\n");

printf("Masukkan batas bawah\t= ");

scanf("%f",&b);

printf("Masukkan batas atas\t= ");

scanf("%f",&a);

printf("Banyak iterasi\t\t= ");

scanf("%f", &n);

printf("Toleransi Error\t\t= ");

scanf("%f", &e);

h = (a - b)/n;

printf("i\t\tx\t\t\tfx\n");

for(i=0;i<n+1;i++)

{

x[i]=b+(i\*h);

y[i]=f(x[i]);

printf("%d\t\t%f\t\t%f\n",i,x[i],y[i]);

}

for(j=0;j<n-1;j++)

{

if((y[j]\*y[j+1])<0)

{

printf("\nAkar diantara %f & %f\n",x[j],x[j+1]);

if(fabs(y[j])<fabs(y[j+1]))

{

printf("Akar lebih dekat ke %f\n", x[j]);

printf("Toleransi error = %f", y[j]);

}

else

{

printf("Akar lebih dekat ke %f\n", x[j+1]);

printf("Toleransi error = %f", y[j+1]);

}

}

}

}

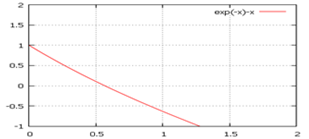
main()

{

tabel();

}

1. **PENGAMATAN AWAL**
2. Gambar GNU Plot

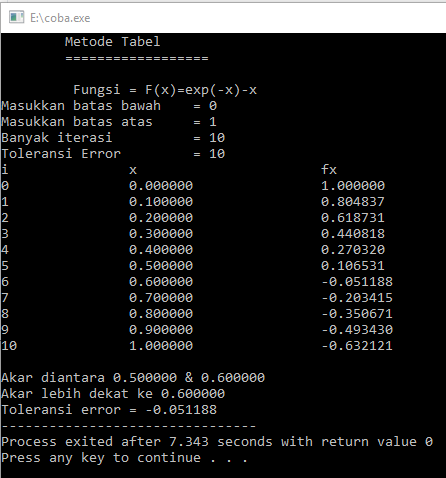


1. Perkiraan Batas Atas dan Batas Bawah

Batas Atas = 0

Batas Bawah = 1

1. **HASIL PERCOBAAN**



1. **KESIMPULAN**

* Metode table adalah metode yang lumayan sederhana.
* Besarnya jumlah iterasi berbanding terbalik dengan besarnya toleransi error yang didapat. Jadi semakin besar kita menginputkan jumlah iterasi maksimumnya, maka nilai toleransi errornya akan semakin kecil.
* Besarnya batas atas dan batas bawah berbanding lurus dengan besarnya toleransi error. Jadi semakin besar selisih batas atas dan bawah, maka nilai toleransi errornya juga semakin besar.