##### **PEMROGRAMAN COMPUTER AIDED**

**“PRAKTIKUM 4”**

**“PENYELESAIAN PERSAMAAN NON LINIER METODE ITERASI”**



Oleh :

Achmad Robith Fuadi 4210161027

**Program Studi Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2018**

**LAPORAN PRAKTIKUM 4**

|  |  |  |
| --- | --- | --- |
| Judul | : | **Penyelesaian Persamaan Non Linier Metode Iterasi** |
| Tujuan | : | Mempelajari metode iterasi untuk penyelesaian persamaan non linier. |
| Dasar Teori | : | Metode iterasi sederhana adalah metode yang memisahkan x dengan sebagian x yang lain sehingga diperoleh : x = g(x). Sebagai contoh untuk menyelesaikan persamaan x – ex = 0 maka persamaan di ubah menjadi : x = ex atau g(x) = ex. g(x) inilah yang menjadi dasar iterasi pada metode iterasi sederhana ini. |
| Algoritma | : | 1. Defisikan fungsi f(x) dan g(x) 2. Tentukan toleransi error (e) dan iterasi maksimum (n) 3. Tentukan pendekatan awal x[0] 4. Untuk iterasi = 1 s/d n atau F(x[*iterasi*]) ≥ *e*   Xi = g(xi-1)  Hitung F(xi)   1. Akar adalah x terakhir yang diperoleh. |
| Flowchart | : |  |

**FORM LAPORAN AKHIR**

**Dini Ayu Tri Maryani / 4210161011**

**Algoritma :**

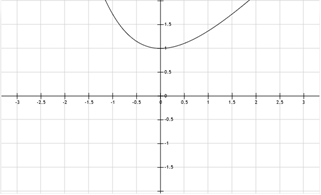
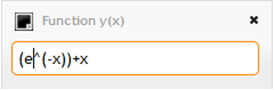
|  |
| --- |
| Judul Percobaan: Metode Iterasi  Algoritma:   1. Defisikan fungsi f(x) dan g(x) 2. Tentukan toleransi error (e) dan iterasi maksimum (n) 3. Tentukan pendekatan awal x[0] 4. Untuk iterasi = 1 s/d n atau F(x[*iterasi*]) ≥ *e*   Xi = g(xi-1)  Hitung F(xi)   1. Akar adalah x terakhir yang diperoleh. |

**Listing program yang sudah benar :**

|  |
| --- |
| **#include<iostream>**  **#include <cmath>**  **using namespace std;**  **int main(){**  **int interval;**  **int i = 1;**    **double x0 = 1;**  **double fx = 0;**  **double gx;**  **double error;**  **bool cont = true;**    **cout << "f(x) = e^-x+x\n"<<endl;**  **cout << "cari akar menggunakan metode iterasi sederhana\n"<<endl;**    **while(fx == 0){**  **cout << "Pendekatan awal:\nx0 = ";**  **cin >> x0;**  **fx = pow(1/2.718, x0) + x0;**  **gx = -pow(1/2.718,x0);**  **if(fx == 0){**  **cout << "masukkan lagi\n"<<endl;**  **}**  **}**  **cout << "Masukkan toleransi galat :";**  **cin >>error;**  **cout << "Masukkan iterasi maksimal : ";**  **cin >>interval;**  **cout << "Iterasi\t <\t\t\tg(x)\t\t\tf(x)\n";**  **for (i = 1;i<=interval, abs(fx) >= error;i++){**  **cout << i << "\t" << x0 << "\t\t" << gx << "\t\t\t" << fx <<endl;**  **x0 = gx;**  **gx = -pow(1/2.718, x0);**  **fx = pow(1/2.718, x0) + x0;**  **}**  **cout << i << "\t" << x0 << "\t\t" << gx << "\t\t" << fx <<endl;**  **cout << "akar = " << x0 <<endl;**  **return 0;**  **}**  Output : |

**Pengamatan awal**

1. Gambar kurva fungsi dengan Gnu Plot

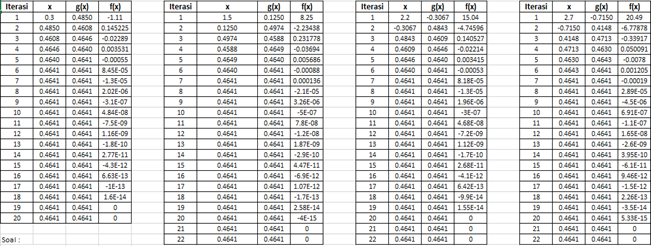
 

1. Perkiraan nilai x0

|  |
| --- |
| X0 |
| -1 |
| -0.75 |
| -0.6 |
| -0.5 |

**Hasil percobaan :**

* 1. Tabel hasil iterasi, xi, f(xi)



* 1. Pengamatan terhadap parameter

1. Toleransi error(e) terhadap jumlah iterasi (N)

|  |  |
| --- | --- |
| Toleransi Error (e) | Jumlah Iterasi (N) |
| 0,1 | 5 |
| 0,01 | 5 |
| 0,001 | 5 |
| 0,0001 | 5 |

1. Pengubahan nilai awal x0 terhadap iterasi (N)

|  |  |
| --- | --- |
| X0 | Iterasi |
| -1 | 5 |
| -0.75 | 5 |
| -0.6 | 6 |
| -0.5 | 6 |

**Kesimpulan :**

|  |
| --- |
| Dengan menggunakan metode iterasi sederhana akar yang diperoleh dapat akurat, hanya saja fungsi yang diterapkan disini tidak ada solusinya (tidak berpotongan pada sumbu x). |