LAPORAN PEMROGRAMAN AIDED

Oleh :

Julian Akbar Renaldi 4210161029

D4 TEKNOLOGI GAME

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

SURABAYA

2017

LAPORAN PRAKTIKUM 6

Judul	:	**Penyelesaian Persamaan Non Linier Metode Newton Raphson dengan Modifikasi Tabel**
Tujuan	:	Mempelajari metode Newton Raphson dengan modifikasi Tabel untuk penyelesaian persamaan non linier.
Dasar Teori	:	Permasalahan pada pemakaian metode newton raphson adalah : 1. Metode ini tidak dapat digunakan ketika titik pendekatannya berada pada titik ekstrim atau titik puncak, karena pada titik ini nilai $F_1(x) = 0$ sehingga nilai penyebut dari $\frac{F(x)}{F^1(x)}$ sama dengan nol, Bila titik pendekatan berada pada titik puncak, maka titik selanjutnya akan berada di tak berhingga. 2. Metode ini menjadi sulit atau lama mendapatkan penyelesaian ketika titik pendekatannya berada di antara dua titik stasioner. Bila titik pendekatan berada pada dua tiitik puncak akan dapat mengakibatkan hilangnya penyelesaian (*divergensi*). Hal ini disebabkan titik selanjutnya berada pada salah satu titik puncak atau arah pendekatannya berbeda. Untuk dapat menyelesaikan kedua permasalahan pada metode newton raphson ini, maka metode newton raphson perlu dimodifikasi dengan : a. Bila titik pendekatan berada pada titik puncak maka titik pendekatan tersebut harus di geser sedikit, $x_i = x_i \pm \delta$ dimana δ adalah konstanta yang ditentukan dengan demikian $F_1(x_i) \neq 0$ dan metode newton raphson tetap dapat berjalan. b. Untuk menghindari titik-titik pendekatan yang berada jauh, sebaiknya pemakaian metode newton raphson ini didahului oleh metode tabel, sehingga dapat di jamin konvergensi dari metode newton raphson.

Algoritma	:	(1) Defisikan fungsi f(x) (2) Ambil range nilai x =□*a,b*□□dengan jumlah pembagi p (3) Masukkan torelansi error (e) dan masukkan iterasi n (4) Gunakan algoritma tabel diperoleh titik pendekatan awal x_0 dari: $F(x_k) \, . \, F(x_{k+1})<0$ maka $x_0 = x_k$ (5) Hitung $F(x_0)$ dan $F_1(x_0)$ (6) Bila *F*□*abs*□*F1* □*x*0 □□□□□*e* maka pendekatan awal x_0 digeser sebesar dx (dimasukkan) $x_0 = x_0 + dx$ hitung $F(x_0)$ dan $F_1(x_0)$ (7) Untuk iterasi I= 1 s/d n atau $\|F(x_i)\| \geq$□*e* $x_1 = x_{i\text{-}1} - \frac{F(x_{i-1})}{F^1(x_{i-1})}$ hitung $F(x_i)$ dan $F_1(x_i)$ bila $\|F(x_i)\| < e$ maka $x_i = x_i + dx$ hitung $F(x_i)$ dan $F_1(x_0)$ (8) Akar persamaan adalah x terakhir yang diperoleh.

Algoritma :

Judul Percobaan: Metode Newton Raphson dengan Modifikasi Tabel

Algoritma:

1. Defisikan fungsi f(x)
2. Ambil range nilai x =□*a,b*□□dengan jumlah pembagi p
3. Masukkan torelansi error (e) dan masukkan iterasi n
4. Gunakan algoritma tabel diperoleh titik pendekatan awal x_0 dari:
   $F(x_k) \, . \, F(x_{k+1})<0$ maka $x_0 = x_k$
5. Hitung $F(x_0)$ dan $F_1(x_0)$
6. Bila *F*□*abs*□*F1* □*x*0 □□□□□*e* maka pendekatan awal x_0 digeser sebesar dx (dimasukkan)
   $x_0 = x_0 + dx$
   hitung $F(x_0)$ dan $F_1(x_0)$
7. Untuk iterasi I= 1 s/d n atau $|F(x_i)| \geq$□*e*

$$x_i = x_{i-1} - \frac{F(x_{i-1})}{F^1(x_{i-1})}$$

hitung $F(x_i)$ dan $F_1(x_i)$

bila $|F(x_i)| < e$ maka

$x_i = x_i + dx$

hitung $F(x_i)$ dan $F_1(x_0)$

8. Akar persamaan adalah x terakhir yang diperoleh.

Program :

Metode Newton Raphson dengan Modifikasi Tabel.cpp

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int iterasi, i = 1;
    double ra, rb, pembagi, fxa, fxb, fx, fdx, error, x0, x[20], y[20];
    bool cont = true;
    cout<<"Metode Newton Raphson dengan Modifikasi Tabel"<<endl
        <<"f(x) = x * e ^ -x + cos(2 * x)"<<endl
        <<"f'(x) = (1 - x) * e ^ -x - 2 * sin(2 * x)"<<endl
        <<"Cari akar menggunakan metode newton raphson dengan modifikasi tabel"<<endl;

    cout<<"Batas bawah : "<<endl
        <<"rbawah : ";
    cin>>rb;
    fxb = rb * pow(1 / 2.718, rb) + cos((2 * rb) * 3.1415 / 180.0);

    cout<<"Batas atas : "<<
        endl<<"ratas = ";
    cin>>ra;
    fxa = ra * pow(1 / 2.718, ra) + cos((2 * ra) * 3.1415 / 180.0);

    cout<<"Masukan toleransi galat : ";
    cin>>error;
    cout<<"Masukan iterasi maksimal : ";
    cin>>iterasi;

    pembagi = (ra - rb) / iterasi;
    for(i = 0; i <= iterasi; i++){
        x[i] = rb + i * pembagi;
```

```cpp
            y[i] = x[i] * pow(1 / 2.718, x[i]) + cos(2 * x[i]);
        }
        for(i = 0; i <= iterasi; i++){
            if(y[i] == 0){
                x0 = x[i];
                cout<<"Titik pendekatan awal x0 = "<<x0<<"\n";
            }
            if(y[i] * y[i + 1] < 0){
                if(abs(y[i]) < abs(y[i + 1])){
                    x0 = x[i];
                    cout<<"Titik pendekatan awal x0 = "<<x0<<endl;
                }
                else{
                    x0 = x[i + 1];
                    cout<<"Titik pendekatan awal x0 = "<<x0<<endl;
                }
            }
        }
        cout<<"Diperoleh x0 dari metode tabel = "<<x0<<endl;
        fx = x0 * pow(1 / 2.718, x0) + cos(2 * x0);
        fdx = (1 - x0) * pow(1 / 2.718, x0) - 2 * sin(2 * x0);
        cout<<"Iterasi\tx\t\tf(x)\t\tf'(x)"<<endl
            <<"--------------------------------"
            <<"--------------------------------"<<endl;
        for(i = 1; i <= iterasi, abs(fx) >= error; i++){
            cout<<i<<"\t"<<x0<<"\t\t"<<fx<<"\t\t"<<fdx<<endl;
            x0 = x0 - (fx / fdx);
            fx = x0 * pow(1 / 2.718, x0) + cos(2 * x0);
            fdx = (1 - x0) * pow(1 / 2.718, x0) - 2 * sin(2 * x0);
        }
        cout<<i<<"\t"<<x0<<"\t\t"<<fx<<"\t\t"<<fdx<<endl;
```

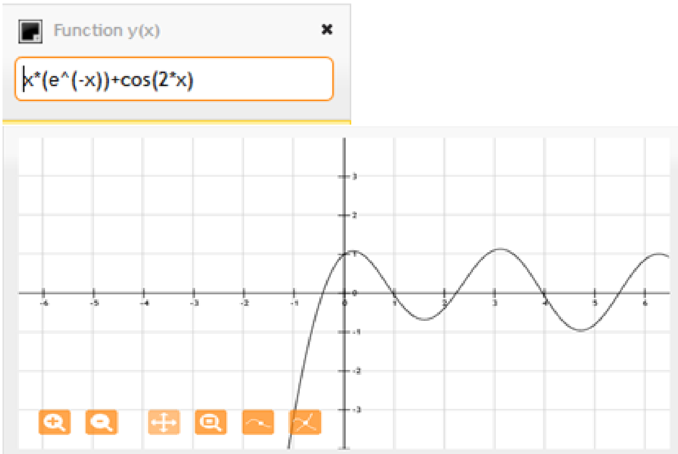
Output :

```
D:\Private\Academy PENS\Semester 4\Pemrograman Aided\Metode Newt...
Metode Newton Raphson dengan Modifikasi Tabel
f(x) = x * e ^ -x + cos(2 * x)
f'(x) = (1 - x) * e ^ -x - 2 * sin(2 * x)
Cari akar menggunakan metode newton raphson dengan modifikasi tabel
Batas bawah :
rbawah : 0
Batas atas :
ratas = 1
Masukan toleransi galat : 0.0001
Masukan iterasi maksimal : 10
Titik pendekatan awal x0 = 1
Titik pendekatan awal x0 = 1
Diperoleh x0 dari metode tabel = 1
Iterasi x               f(x)            f'(x)
--------------------------------------------------------------
1       1               -0.0482293              -1.81859
2       0.97348         0.000429914             -1.85014
3       0.973712                3.80785e-008            -1.84989

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Pengamatan awal

a. Gambar kurva fungsi dengan Gnu Plot



b. Perkiraan nilai x0

X0
0
0.25
0.55
0.75

Hasil percobaan :

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

Iterasi	x	f(x)	f'(x)
0	1.000000000000000	-0.801359446372008	0.491995113722046
1	2.628795538861260	0.566016797799346	1.027432001976020
2	2.077891151309820	0.017171773221860	0.951423550138314
3	2.059842645763200	0.000035255147925	0.947422078478384
4	2.059805434103560	-0.000000001250600	0.947413712915499
5	2.059805435423570	0.000000000000050	0.947413713212260
6	2.059805435423520	0.000000000000000	0.947413713212248
7	2.059805435423520	0.000000000000000	0.947413713212248
8	2.059805435423520	0.000000000000000	0.947413713212248
9	2.059805435423520	0.000000000000000	0.947413713212248
10	2.059805435423520	0.000000000000000	0.947413713212248
11	2.059805435423520	0.000000000000000	0.947413713212248
12	2.059805435423520	0.000000000000000	0.947413713212248
13	2.059805435423520	0.000000000000000	0.947413713212248

2) Pengamatan terhadap parameter

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

Toleransi Error (e)	Jumlah Iterasi (N)
0,1	4
0,01	5
0,001	5
0,0001	6

b. Pengubahan nilai awal x0 terhadap iterasi (N)

X0	Iterasi
0	6
0.25	6
0.75	6
0.55	6

Kesimpulan :

Dengan modifikasi tabel, fungsi yang memotong sumbu x lebih dari sekali dapat dicari titik yang medekati pertama kali.