Nama : **NUR ANISAH**

Nim : **200170015**

Kelas : A2 – Metode Numerik

1. Newton Methode

a. Sourcecode

Newton_methode003.cpp

```c
#include<stdio.h>
#include<math.h>
#include<conio.h>

float Fungsi(float x);
float FungsiTurunan(float x);
float e;

main(){
    int n = 0;
    float x[100];
    printf("PROGRAM NUMERIK METODE NEWTON \n\n");
    printf("Persamaan Fungsi :  f(x)=x^3+x^2-3x-3 \n");
    printf("dengan x(n+1) = xn - (f(xn)/f'(xn))\n\n");

//  inputan
    printf("Masukkan Nilai x Awal   : "); scanf("%f", &x[0]);
    printf("Masukkan Nilai Toleransi Error  : "); scanf("%f", &e);


//  menampilkan tabel hasil iterasi
    printf("=================================================================\n");
    printf("! n ! xn ! f(xn) ! f'(xn) ! f(xn)/f'(xn) ! x(n+1) !\n");
    printf("=================================================================\n");

//  struktur kondisi metode Newton
    do{
        x[n+1]= x[n] - (Fungsi(x[n])/FungsiTurunan(x[n]));
        printf("! %d ! %2.4f ! %2.4f ! %2.4f ! %2.4f ! %2.4f !\n", n, x[n], Fungsi(x[n]), FungsiTurunan(x[n]), Fungsi(x[n])/FungsiTurunan(x[n]), x[n+1]);
        n++;
    } while (abs(x[n-1])-(x[n-2])>e);

    printf("! %d ! %2.4f ! %2.4f ! %2.4f ! %2.4f ! %2.4f !\n", n, x[n], Fungsi(x[n]), FungsiTurunan(x[n]), Fungsi(x[n])/FungsiTurunan(x[n]), x[n+1]);
    printf("=================================================================\n");
    printf("Jadi, hasil yang memnuhi dari persamaan tersebut adalah x = %2.4f", x[n]);
    getch();
}

//  membuat function Fungsi
float Fungsi(float x){
    return pow (x,3)+(x,2)-(3*x)-3;
}

//  membuat function FungsiTurunan
float FungsiTurunan(float x){
    return pow (3*x,2)+(2*x)-3;
}
```

b. Hasil

```
D:\Materi Kuliah\Semester 4\Metode Numerik\Newton Methode\Newton_methode003.exe
PROGRAM NUMERIK METODE NEWTON

Persamaan Fungsi :  f(x)=x^3+x^2-3x-3
dengan x(n+1) = xn - (f(xn)/f'(xn))

Masukkan Nilai x Awal   : 3
Masukkan Nilai Toleransi Error  : 0.000001
=================================================================
! n ! xn ! f(xn) ! f'(xn) ! f(xn)/f'(xn) ! x(n+1) !
=================================================================
! 0 ! 3.0000 ! 17.0000 ! 84.0000 ! 0.2024 ! 2.7976 !
! 1 ! 2.7976 ! 12.5032 ! 73.0353 ! 0.1712 ! 2.6264 !
! 2 ! 2.6264 ! 9.2381 ! 64.3358 ! 0.1436 ! 0.0000 !
=================================================================
Jadi, hasil yang memnuhi dari persamaan tersebut adalah x = 2.6264
```

2. Secan Method

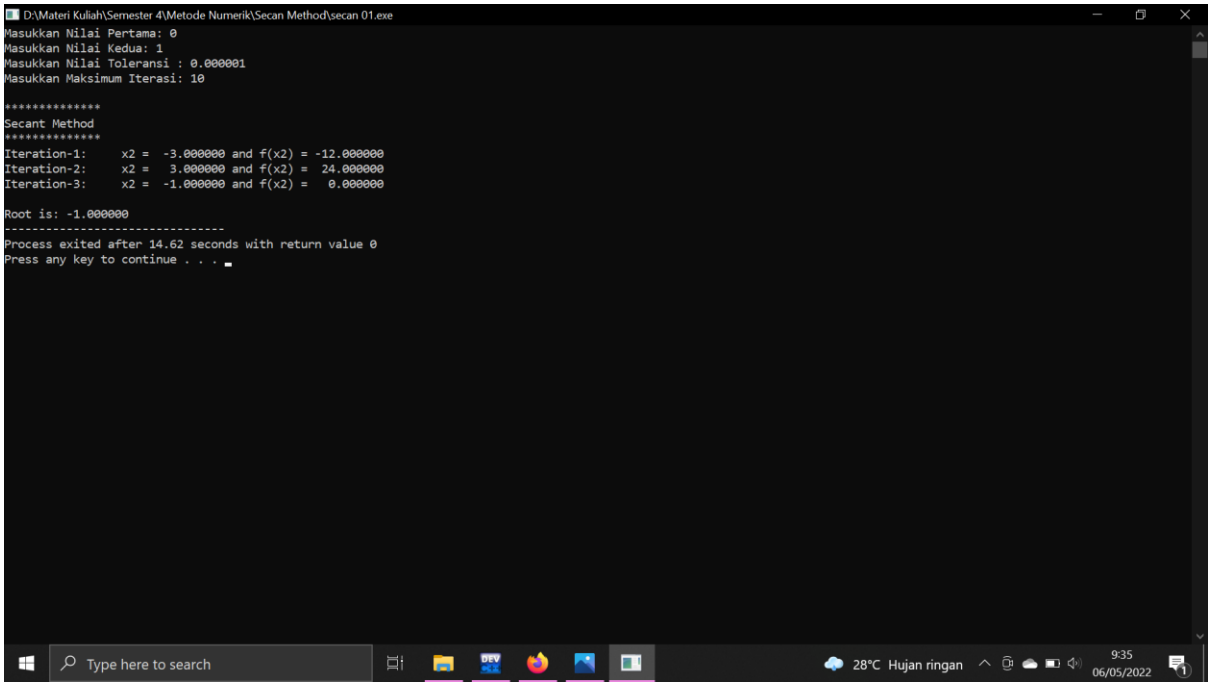
a. Sourcecode

[*] secan 01.cpp

```cpp
#include<iostream>
#include<iomanip>
#include<math.h>
#include<stdlib.h>
#define    f(x)    x*x*x + x*x - 3*x - 3

using namespace std;

int main()
{
    float x0, x1, x2, f0, f1, f2, e;
    int step = 1, N;

   cout<< setprecision(6)<< fixed;

    /* Input nilai */
    cout<<"Masukkan Nilai Pertama: ";
    cin>>x0;
    cout<<"Masukkan Nilai Kedua: ";
    cin>>x1;
    cout<<"Masukkan Nilai Toleransi : ";
    cin>>e;
    cout<<"Masukkan Maksimum Iterasi: ";
    cin>>N;

    /* Implementasi Secant Method */
    cout<< endl<<"**************"<< endl;
    cout<<"Secant Method"<< endl;
    cout<<"**************"<< endl;
```

```cpp
    do
    {
        f0 = f(x0);
        f1 = f(x1);
        if(f0 == f1)
        {
            cout<<"Mathematical Error.";
            exit(0);
        }

        x2 = x1 - (x1 - x0) * f1/(f1-f0);
        f2 = f(x2);

        cout<<"Iteration-"<< step<<":\t x2 = "<< setw(10)<< x2<<" and f(x2) = "<< setw(10)<< f(x2)<< endl;

        x0 = x1;
        f0 = f1;
        x1 = x2;
        f1 = f2;

        step = step + 1;

        if(step > N)
        {
            cout<<"Not Convergent.";
            exit(0);
        }
    }while(fabs(f2)>e);

    cout<< endl<<"Root is: "<< x2;

    return 0;
}
```

b. Hasil


D:\Materi Kuliah\Semester 4\Metode Numerik\Secan Method\secan 01.exe
Masukkan Nilai Pertama: 0
Masukkan Nilai Kedua: 1
Masukkan Nilai Toleransi : 0.000001
Masukkan Maksimum Iterasi: 10
****
Secant Method
****
Iteration-1: x2 = -3.000000 and f(x2) = -12.000000
Iteration-2: x2 =   3.000000 and f(x2) = 24.000000
Iteration-3: x2 = -1.000000 and f(x2) =   0.000000
Root is: -1.000000

Process exited after 14.62 seconds with return value 0
Press any key to continue . . .
Type here to search
28°C Hujan ringan
9:35
06/05/2022