LAPORAN PRAKTIKUM
METODE NUMERIK

Judul: Persamaan Non-Linier II

DISUSUN OLEH

ILHAM NUR ROMDONI M0520038

PROGRAM INFORMATIKA

FAKULTAS MIPA

UNIVERSITAS SEBELAS MARET

2021

SCREENSHOT

A. Screenshot Praktikum

1. $y = x^3 - 2x^2 + 3$ dengan $x_0 = 3$, (n = 20, ε = 0,0001)

   a. Metode Newton Raphson

```
>> syms x
>> f = x^3 - 2*x^2 + 3

f =

x^3 - 2*x^2 + 3

>> y = Metode_Newton_Raphson(f,3,20,0.0001)

f_asli =

x^3 - 2*x^2 + 3


f_turunan =

3*x^2 - 4*x

Iter     fa           fb          y
  0   12.0000     15.0000     2.2000
  1    3.9680     5.7200     1.5063
  2    1.8798     0.7816    -0.8988
  3    0.6580     6.0191    -1.0082
  4   -0.0574      7.0818     -1.0000
  5   -0.0003      7.0005     -1.0000
  6   -0.0000      7.0000     -1.0000

y =

    -1
```

   b. Metode Secant

```
>> f = inline('x^3 - 2*x^2 + 3')

f =

     Inline function:
     f(x) = x^3 - 2*x^2 + 3

>> y = Metode_Secant(f,3,20,0.0001)
Iter    x0       xl         x        abs(x0-xl)    fa
  0    3.00     2.00      1.00        1.6667     12.0000
  1    2.00     1.67      0.33        0.9200     3.0000
  2    1.67     0.92      0.75        132.7528     2.0741
  3    0.92     132.75      131.83        0.9199     2.0859
  4    132.75     0.92      131.83        0.9198     2304300.0632
  5    0.92     0.92      0.00        2.7480     2.0860
  6    0.92     2.75      1.83        0.3386     2.0862
  7    2.75     0.34      2.41        -0.8208     8.6486
  8    0.34     -0.82      1.16        -1.5665     2.8095
  9   -0.82     -1.57      0.75        -0.9405     1.0998
 10   -1.57     -0.94      0.63        -0.9811     -5.7521
 11   -0.94     -0.98      0.04        -1.0008     0.3993
 12   -0.98     -1.00      0.02        -1.0000     0.1306
 13   -1.00     -1.00      0.00        -1.0000     -0.0059

y =

   -1.0000
```

2. $y = e^x - sinx$ dengan $x_0 = 1$, (n = 20, ε = 0,0001)

   a. Metode Newton Raphson

```
>> syms x
>> f = exp(x) - sin(x)

f =

exp(x) - sin(x)

>> y = Metode_Newton_Raphson(f,1,20,0.0001)

f_asli =

exp(x) - sin(x)


f_turunan =

exp(x) - cos(x)

Iter     fa           fb          y
  0    1.8768     2.1780     0.1383
  1    1.0105     0.1578    -6.2635
  2   -0.0178     -0.9979    -6.2813
  3    0.0000     -0.9981    -6.2813

y =

   -6.2813
```

   b. Metode Secant

```
>> f = inline('exp(x) - sin(x)')

f =

     Inline function:
     f(x) = exp(x) - sin(x)

>> y = Metode_Secant(f,1,20,0.0001)
Iter    x0       xl         x        abs(x0-xl)    fa
  0    1.00     0.00      1.00        -1.1405     1.8768
  1    0.00    -1.14      1.14         4.9912     1.0000
  2   -1.14     4.99      6.13        -1.1918     1.2285
  3    4.99    -1.19      6.18        -1.2437     148.0759
  4   -1.19    -1.24      0.05         23.5648     1.2327
  5   -1.24    23.56      24.81         -1.2437     1.2353
  6   23.56    -1.24      24.81         -1.2437     17142435237.4305

y =

   -1.2437
```

B. Screenshot Source Code

1. Metode Newton Raphson

```
% Ilham Nur Romdoni, M0520038

function y = Metode_Newton_Raphson (f, x0, n, tol)
f_asli = sym(f)
f_turunan = diff(f_asli,'x')
y = double(x0);
i = 0;
fa = subs (f_asli,y);
fprintf('Iter    fa          fb         y\n');
while (abs(fa) > tol)
    fa = subs (f_asli,y);
    fb = subs (f_turunan,y);
    if fa == 0 || i == n
        break
```

```
    end
    y = double (y - fa./fb);
    nilai_y = double(y);
    nilai_fa = double(fa);
    nilai_fb = double(fb);
    fprintf('%3.0f   %5.4f    %5.4f    %5.4f\n',i,nilai_fa,nilai_fb,nilai_y);
    %disp ([i nilai_y nilai_fa])
    i = i + 1;
end
```

2. Metode Secant

```
% Ilham Nur Romdoni, M0520038

function x = Metode_Secant (f, x0, n, tol)
fa = f(x0);
x1 = x0-1;
fb = f(x1);
i = 0;
fprintf('Iter   x0      x1        x       abs(x0-x1)   fa\n');
while (abs (x0 - x1) > tol)
    fa = f(x0);
    fb = f(x1);
    if (fa == 0 || i == n)
        return
    end
    x = x1 - fb.*(x1-x0)./(fb-fa);
    %disp([i x fa)])
    fprintf('%3.0f   %4.2f    %4.2f     %4.2f       %5.4f    %5.4f\n',i,x0,x1,abs(x0-x1),x,fa);
    x0=x1;
    x1=x;
    i=i+1;
end
```

ANALISIS

A. Analisis Source Code

1. Metode Newton Raphson
    - Dideklarasikan fungsi dari Metode_Newton_Raphson dengan parameter *input*-an (f, x0, n, tol). f merupakan fungsi, x0 adalah nilai awal, n sebagai batas iterasi sedangkan tol yaitu batas *error*.
    - Diinisialisasikan f_asli yang akan diisi oleh symbolic dari f yang *input*-kan.
    - Lalu f_turunan adalah *differensial* dari f_asli terhadap x.
    - y diinisialisasi dengan nilai x0 dengan tipe data double.
    - fa adalah substitasi dari f_asli terhadap nilai y. Substitusi akan mengubah nilai x default dari f_asli dengan nilai y.
    - Perulangan while dengan batas nilai mutlak fa lebih besar dari toleransi.
    - fa akan diisi f_asli sedangkan fb akan diisi nilai dari f_turunan.
    - Ketika fa bernilai 0 atau i sudah bernilai n maka break. Ini berarti kasus sudah berada pada batas iterasi tetapi belum pada batas *error* maka proses dihentikan.
    - Dimasukkan rumus Newton Raphson untuk mendapatkan nilai y baru dengan tipe data double.
    - Nilai baru dari y disimpan dalam nilai_y sedangkan nilai_fa menyimpan nilai fa yang baru.
    - Tampilkan hasil dari i, nilai_fa, nilai_fb, dan nilai_y. Dengan i = i +1 di mana berarti nilai i bertambah pada setiap perulangan untuk menunjukkan iterasi.

2. Metode Secant
    - Dideklarasikan fungsi Metode_Secant dengan parameter *input*-an (f, x0, n, tol). f merupakan fungsi, x0 adalah nilai awal, n sebagai batas iterasi sedangkan tol yaitu batas *error*.
    - fa diinisialisasi dengan nilai dari fungsi dari f(x0) dan fb adalah nilai dari f(x1). x1 didapatkan dengan mengurangkan x0 dengan 1.
    - Perulangan while dengan batas nilai mutlak dari x0 dikurangi x1 lebih besar dari toleransi
    - fa akan diisi f(x0) sedangkan fb akan diisi nilai dari f(x1).

- Ketika fa bernilai 0 atau i sudah bernilai n maka return. Ini berarti kasus sudah berada pada batas iterasi tetapi belum pada batas *error* maka proses dihentikan.
- Dimasukkan rumus Secant untuk mendapatkan nilai x baru.
- Nilai baru dari x1 disimpan dalam x0 yang didapatkan dari nilai baru dari x.
- Tampilkan hasil dari i, x0, x1, x, abs(x0-x1) dan fa. Dengan i = i +1 di mana berarti nilai i bertambah pada setiap perulangan untuk menunjukkan iterasi.

B. Analisis Jalannya Program

1. Tentukan nilai akar yang diperoleh dari bentuk persamaan linier di bawah ini dengan menggunakan metode Newton Raphson dan metode Secant

   a. $y = e^x - sinx$

   1) Metode Newton Raphson

```
>> syms x
>> f = exp(x) - sin(x)

f =

exp(x) - sin(x)

>> y = Metode_Newton_Raphson(f,1,20,0.0001)

f_asli =

exp(x) - sin(x)


f_turunan =

exp(x) - cos(x)

Iter    fa          fb          y
  0   1.8768    2.1780    0.1383
  1   1.0105    0.1578   -6.2635
  2   -0.0178    -0.9979    -6.2813
  3   0.0000    -0.9981    -6.2813

y =

   -6.2813
```

   2) Metode Secant

```
>> f = inline('exp(x) - sin(x)')

f =

     Inline function:
     f(x) = exp(x) - sin(x)

>> y = Metode_Secant(f,1,20,0.0001)
Iter   x0      x1        x       abs(x0-x1)    fa
  0   1.00    0.00     1.00       -1.1405    1.8768
  1   0.00    -1.14     1.14        4.9912    1.0000
  2   -1.14    4.99     6.13       -1.1918    1.2285
  3   4.99    -1.19     6.18       -1.2437    148.0759
  4   -1.19    -1.24     0.05        23.5648    1.2327
  5   -1.24    23.56     24.81        -1.2437    1.2353
  6   23.56    -1.24     24.81        -1.2437    17142435237.4305

y =

   -1.2437
```

b. $y = 2x^3 - 2x^2 - 2$

1) Metode Newton Raphson

```
>> syms x
>> f = 2*x^3 - 2*x^2 - 2

f =

2*x^3 - 2*x^2 - 2

>> y = Metode_Newton_Raphson(f,3,20,0.0001)

f_asli =

2*x^3 - 2*x^2 - 2


f_turunan =

6*x^2 - 4*x

Iter    fa           fb          y
  0   34.0000    42.0000    2.1905
  1   9.4243    20.0272    1.7199
  2   2.2591    10.8688    1.5121
  3   0.3414    7.6697    1.4675
  4   0.0138    7.0518    1.4656
  5   0.0000    7.0252    1.4656

y =

    1.4656
```

2) Metode Secant

```
>> f = inline('2*x^3 - 2*x^2 - 2')

f =

     Inline function:
     f(x) = 2*x^3 - 2*x^2 - 2

>> y = Metode_Secant(f,3,20,0.0001)
Iter   x0      xl        x       abs(x0-xl)   fa
  0   3.00    2.00     1.00       1.7857    34.0000
  1   2.00    1.79     0.21       1.5699    6.0000
  2   1.79    1.57     0.22       1.4906    3.0109
  3   1.57    1.49     0.08       1.4679    0.8088
  4   1.49    1.47     0.02       1.4656    0.1800
  5   1.47    1.47     0.00       1.4656    0.0163

y =

    1.4656
```

c. $y = 2 \sin x \cos 3x$

1) Metode Newton Raphson

```
>> syms x
>> f = 2*sin(x)*cos(3*x)

f =

2*cos(3*x)*sin(x)

>> y = Metode_Newton_Raphson(f,3,20,0.0001)

f_asli =

2*cos(3*x)*sin(x)
```

```
f_turunan =

2*cos(3*x)*cos(x) - 6*sin(3*x)*sin(x)

Iter    fa          fb          y
  0   -0.2572    1.4551    3.1767
  1    0.0699    1.9655    3.1412
  2   -0.0008    2.0000    3.1416
  3    0.0000    2.0000    3.1416

y =

    3.1416
```

2) Metode Secant

```
>> f = inline('2*sin(x)*cos(3*x)')

f =

     Inline function:
     f(x) = 2*sin(x)*cos(3*x)

>> y = Metode_Secant(f,3,20,0.0001)
Iter    x0       x1        x        abs(x0-x1)    fa
  0   3.00     2.00      1.00        2.8716    -0.2572
  1   2.00     2.87      0.87        2.7200     1.7462
  2   2.87     2.72      0.15        2.4117    -0.3678
  3   2.72     2.41      0.31        2.6455    -0.2465
  4   2.41     2.65      0.23        2.6240     0.7735
  5   2.65     2.62      0.02        2.6177    -0.0784
  6   2.62     2.62      0.01        2.6180    -0.0177
  7   2.62     2.62      0.00        2.6180     0.0009

y =

    2.6180

>> y = Metode_Secant(f,4,20,0.0001)
Iter    x0       x1        x        abs(x0-x1)    fa
  0   4.00     3.00      1.00        2.7479    -1.2773
  1   3.00     2.75      0.25        4.8878    -0.2572
  2   2.75     4.89      2.14        3.2349    -0.2915
  3   4.89     3.23      1.65        2.8694     0.9893
  4   3.23     2.87      0.37        3.1153     0.1792
  5   2.87     3.12      0.25        3.1561    -0.3682
  6   3.12     3.16      0.04        3.1416    -0.0525
  7   3.16     3.14      0.01        3.1416     0.0291

y =

    3.1416
```

d. $y = \frac{\sin(x)}{x^2 - e^x}$

1) Metode Newton Raphson

```
>> syms x
>> f = sin(x)/(x^2-exp(x))

f =

-sin(x)/(exp(x) - x^2)

>> y = Metode_Newton_Raphson(f,2,20,0.0001)

f_asli =

-sin(x)/(exp(x) - x^2)
```

```
f_turunan =

- cos(x)/(exp(x) - x^2) - (sin(x)*(2*x - exp(x)))/(exp(x) - x^2)^2

Iter    fa          fb          y
  0   -0.2683     0.3911     2.6860
  1   -0.0590     0.1940     2.9902
  2   -0.0138     0.1078     3.1180
  3   -0.0018     0.0800     3.1409
  4   -0.0001     0.0755     3.1416

y =

    3.1416
```

2) Metode Secant

```
>> f = inline('sin(x)/(x^2-exp(x))')

f =

     Inline function:
     f(x) = sin(x)/(x^2-exp(x))

>> y = Metode_Secant(f,2,20,0.0001)
Iter    x0       x1         x        abs(x0-x1)    fa
  0   2.00     1.00      1.00        3.2118     -0.2683
  1   1.00     3.21      2.21        3.1902     -0.4897
  2   3.21     3.19      0.02        3.1368     0.0048
  3   3.19     3.14      0.05        3.1419     0.0034
  4   3.14     3.14      0.01        3.1416     -0.0004
  5   3.14     3.14      0.00        3.1416     0.0000

y =

    3.1416
```

Untuk menjalankan program fungsi metode Newton Raphson, deklarasikan nilai x terlebih dahulu dengan menuliskan syms x pada *command window*. Tuliskan bentuk persamaan ke variabel f. Lalu panggil fungsi Metode_Newton_Raphson dengan parameter *input*-an yang sesuai pada *source code*.

Output pada fungsi Metode_Newton_Raphson menampilkan f_asli yang merupakan nilai dari persamaan yang di-*input*-kan dan f_turunan yang didapatkan dari *differensial* persamaan. Program akan menampilkan hasil dengan bentuk seperti tabel ber-*headline* iter, fa, fb, dan y. Di mana setiap baris akan menunjukkan nilai yang baru dari masing-masing variabel karena menjalankan perulangan. Perulangan akan berhenti saat nilai fa yang baru melebihi toleransi atau nilai iter yang baru melebihi batas iterasi yang ditentukan. Nilai y pada iterasi terakhir adalah nilai akar penyelesaian.

Sedangkan pada program fungsi metode Secant, deklarasikan persamaan ke variabel f dengan fungsi inline pada *command window*. Inline serupa dengan *anonymous function* yaitu *function* yang tidak perlu tersimpan pada *file* .m. Lalu panggil fungsi Metode_Secant dengan parameter *input*-an yang sesuai *source code*.

Output pada fungsi Metode_Secant menampilkan hasil dengan bentuk seperti tabel ber-*headline* iter, x0, x1, x, abs(x0-x1) dan fa. Di mana setiap baris akan menunjukkan nilai yang baru dari masing-masing variabel karena menjalankan perulangan. Perulangan akan berhenti saat nilai abs(x0-x1) yang baru melebihi toleransi atau nilai iter yang baru melebihi batas iterasi yang ditentukan. Nilai x1 pada iterasi terakhir adalah nilai akar penyelesaian.

Kesimpulan dari penentuan nilai akar penyelesaian dari 4 persamaan di atas adalah terdapat perbedaan nilai akar penyelesaian dari metode secant dan metode newton Raphson yang signifikan pada persamaan 1. Sedangkan pada ketiga persamaan lain menunjukkan nilai akar penyelesaian yang sama. Praktikan berhipotesis, hal ini terjadi karena terdapat kesalahan pada *source code* yang dibuat meskipun praktikan tidak menemukan kesalahan tersebut.