LAPORAN PRAKTIKUM MATA KULIAH

METODE NUMERIK

PRAKTIKUM 7 - SISTEM PERSAMAAN NONLINEAR

DISUSUN OLEH:

M0521003 – ADI PRASETYA

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

2022

**BAB I**

**ANALISIS *SOURCE CODE***

1. **Analisis *Source code* Soal 1**

   - **Grafik 1 (a)**

```
% M0521003 - Adi Prasetya
% Plot Kurva untuk Grafik pada Soal 1

x=linspace(0,4);
f1=@(x) x.^2 -5;
y=f1(x);
plot(x,y)

line([min(x),max(x)],[0,0]);
line([0,0],[min(y),max(y)]);

%Anotasi
legend('x^2 - 5')
title('Grafik untuk x^2 - 5')
xlabel('Sumbu X')
ylabel('Sumbu Y')

```

Pada *source code* di atas, terdapat ***function*** **linspace(0,4)** untuk membuat vektor baris berisi 100 titik yang terpisah merata secara linear antara 0 dan 4. Kemudian, terdapat pendeklarasian ***function*** **anonymous** untuk dengan parameter (x) dengan isi *function* adalah sebagai berikut.

$$f2(x) = x^2 - 5$$

Lalu, terdapat ***function*** **plot(x,y)** untuk merepresentasikan data dengan grafik yang terdiri atas sumbu X dan sumbu Y. Terdapat ***function*** **line([min(x), max(x)], [0,0] )** dan ***function*** **line([0,0],[min(y), max(y)]** untuk untuk menghasilkan garis sebagai sumbu X dan sumbu Y. Kemudian, terdapat anotasi berupa **legend( )** yang berisi penjelasan garis dilengkapi dengan sampel garis yang dijelaskan. Terdapat anotasi **title( )** untuk membuat judul dari plot grafik yang telah dibuat. Lalu, terdapat ***function*** **xlabel( )** dan **ylabel( )** untuk memberikan keterangan pada sumbu X dan sumbu Y.

- **Grafik 1 (b)**

```
% M0521003 - Adi Prasetya
% Plot Kurva untuk Grafik pada Soal 2

x=linspace(0,4);
f2=@(x) (log(4*sin(2*x)+5)/3)-(2*x/3);
y=f2(x);
plot(x,y)

line([min(x),max(x)],[0,0]);
line([0,0],[min(y),max(y)]);

%Anotasi
legend('(ln(4*sin(2*x)+5)/3)-(2*x/3)')
title('Grafik untuk (ln(4*sin(2*x)+5)/3)-(2*x/3)')
xlabel('Sumbu X')
ylabel('Sumbu Y')
```

Pada *source code* di atas, *source code* ini hampir sama seperti source untuk membuat grafik A, hanya terdapat perbedaan pada pendeklarasian *function* anonymous. Terdapat *function* linspace(0,4) untuk membuat vektor baris berisi 100 titik yang terpisah merata secara linear antara 0 dan 4. Kemudian, terdapat *function* anonymous untuk terdapat pendeklarasian *function* anonymous untuk dengan parameter (x) dengan isi *function* adalah sebagai berikut.

$$f1(x) = \frac{\ln(4\sin(2x) + 5)}{3} - \frac{2x}{3}$$

Lalu, terdapat *function* plot(x,y) untuk merepresentasikan data dengan grafik yang terdiri atas sumbu X dan sumbu Y. Terdapat *function* line([min(x), max(x)], [0,0] ) dan *function* line([0,0],[min(y), max(y)] untuk untuk menghasilkan garis sebagai sumbu X dan sumbu Y. Kemudian, terdapat anotasi berupa legend( ) yang berisi penjelasan garis dilengkapi dengan sampel garis yang dijelaskan. Terdapat anotasi title( ) untuk membuat judul dari plot grafik yang telah dibuat. Lalu, terdapat *function* xlabel( ) dan ylabel( ) untuk memberikan keterangan pada sumbu X dan sumbu Y.

- **Metode Biseksi**

```
% M0521003 - Adi Prasetya

function c = Biseksi(fungsi,a,b,jumlahIterasi,batasError);
tic;
  fa = fungsi(a);
  fb = fungsi(b);
  if fa*fb > 0
    error('Warning: sama tanda.')
  endif
  fprintf(' Iterasi\ta\t\tb\t\tfa\t\tfb\t\tc\t\tfc\n');
  for i=1:jumlahIterasi
    c = (a+b)/2;
    fc = fungsi(c);
    fprintf (' %d\t\t%f\t%f\t%f\t%f\t%f\n',
            i,a,b,fa,fb,c,fc);
    if abs(c-a) <= batasError || abs(c-b) <= batasError
      break;
    endif
    if fa*fc == 0
      break;
    elseif fa*fc < 0
      b = c;
      fb = fungsi(b);
    else
      a = c;
      fa = fungsi(a);
    endif
  endfor
toc;
end

```

Pada *source code* di atas, terdapat ***function*** **Biseksi(fungsi, a, b , jumlahIterasi, batasError)** dimana hasil dari operasi dalam *function* tersebut akan di-*return* dalam **variabel c**. Lalu, nilai awal dari a dan b akan dimasukkan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fa dan fb. Kemudian, dilakukan operasi fa*fb dengan *if condition*, dimana **jika nilai fa*fb lebih dari nol** maka akan menghasilkan error, yaitu terdapat kesamaan tanda. Selanjutnya dilakukan *for loop* yang dimulai dari i = 1 hingga i = jumlah iterasi, yaitu 20. Kemudian dalam *for loop* tersebut, dilakukan pencarian nilai c dengan rumus sebagai berikut

$$c = \frac{a+b}{2}$$

Kemudian, nilai dari c tersebut dimasukan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fc. Salah satu syarat berhentinya iterasi dari metode biseksi adalah **ketika nilai galat kurang dari batas toleransi galat yang sudah ditentukan** sehingga terdapat *if condition* yang menyatakan bahwa jika nilai *absolute* (c-a) atau *absolute* (c-b) kurang atau sama dengan nilai batas toleransi galat, maka *looping* akan berhenti. Syarat yang lain adalah **jika nilai fa*fc sama**

**dengan nol**, maka iterasi juga akan berhenti. Kemudian, terdapat *else if condition* dimana jika fa*fc kurang dari nol maka nilai dari c sebelumnya akan menggantikan nilai b, lalu nilai b yang baru akan dimasukkan pada f(b) yang nilainya akan disimpan dalam variabel fb. Terdapat *else condition* dimana nilai dari c sebelumnya akan menggantikan nilai a yang baru, lalu nilai a yang baru akan dimasukkan pada f(a) yang hasilnya akan disimpan dalam variabel fa. Iterasi ini baru akan berhenti jika terdapat kondisi dimana nilai galat kurang dari atau sama dengan batas error atau nilai fa*fc sama dengan nol.

- **Metode Regula Falsi**

```
% M0521003 - Adi Prasetya

function c = RegulaFalsi(fungsi,a,b,jumlahIterasi,batasError)
tic;
  fa = fungsi(a);
  fb = fungsi(b);
  if fa*fb > 0
    error('Warning: sama tanda.')
  endif
  fprintf(' Iterasi\ta\t\tb\t\tfa\t\tfb\t\tc\t\tfc\n');
  for i=1:jumlahIterasi
    c = (a*fb-b*fa)/(fb-fa);
    fc = fungsi(c);
    fprintf (' %d\t\t%f\t%f\t%f\t%f\t%f\t%f\n',
            i,a,b,fa,fb,c,fc);
    if abs(c-a) <= batasError || abs(c-b) <= batasError
      break;
    endif
    if fa*fc == 0
      break;
    elseif fa*fc < 0
      b = c;
      fb = fungsi(b);
    else
      a = c;
      fa = fungsi(a);
    endif
  endfor
toc;
end
```

Pada *source code* di atas, terdapat ***function*** **RegulaFalsi(fungsi, a, b , jumlahIterasi, batasError)** dimana hasil dari operasi dalam *function* tersebut akan di-*return* dalam **variabel c**. Pada dasarnya, algoritma pada metode biseksi dan regula falsi adalah sama, perbedaanya hanya pada pencarian nilai c. Lalu, nilai awal dari a dan b akan dimasukkan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fa dan fb. Kemudian, dilakukan operasi fa*fb dengan *if condition*, dimana **jika nilai fa*fb lebih dari nol** maka akan menghasilkan error, yaitu terdapat kesamaan tanda. Selanjutnya dilakukan *for loop* yang dimulai dari i = 1 hingga i = jumlah iterasi, yaitu 20. Kemudian dalam *for loop* tersebut, dilakukan pencarian nilai c dengan rumus sebagai berikut

$$c = \frac{a \times fb - b \times fa}{fa \times fb}$$

Kemudian, nilai dari c tersebut dimasukan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fc. Salah satu syarat berhentinya iterasi dari metode biseksi adalah **ketika nilai galat kurang dari batas toleransi galat yang sudah ditentukan** sehingga terdapat *if condition* yang menyatakan bahwa jika nilai *absolute* (c-a) atau *absolute* (c-b) kurang atau sama dengan nilai batas toleransi galat, maka *looping* akan berhenti. Syarat yang lain adalah **jika nilai fa*fc sama dengan nol**, maka iterasi juga akan berhenti. Kemudian, terdapat *else if condition* dimana jika fa*fc kurang dari nol maka nilai dari c sebelumnya akan menggantikan nilai b, lalu nilai b yang baru akan dimasukkan pada f(b) yang nilainya akan disimpan dalam variabel fb. Terdapat *else condition* dimana nilai dari c sebelumnya akan menggantikan nilai a yang baru, lalu nilai a yang baru akan dimasukkan pada f(a) yang hasilnya akan disimpan dalam variabel fa. Iterasi ini baru akan berhenti jika terdapat kondisi dimana nilai galat kurang dari atau sama dengan batas error atau nilai fa*fc sama dengan nol.

- **Metode Secant**

```
% M0521003 - Adi Prasetya

function c = Secant(f,a,b,jumlahIterasi,batasError)
  fprintf(' Iter\ta\t\tb\t\tfa\t\tfb\t\tc\t\tfc\n');
tic;
  for i=1:jumlahIterasi
    fa = f(a);
    fb = f(b);
    if fb==fa
      error("Pembagian dengan 0.")
    endif
    c = b-(fb*(b-a))/(fb-fa);
    fc = f(c);
    fprintf (' %d\t%f\t%f\t%f\t%f\t%f\t%f\n',
            i,a,b,fa,fb,c,fc);
    if abs(c-b)<=batasError || fc==0
      toc;
      return;
    else
      a=b;
      b=c;
    endif
  endfor
toc;
end
```

Pada *source code* di atas, terdapat *function* Secant(f, a, b, jumlahIterasi, batasError) dimana hasil dari operasi dalam *function* tersebut akan di-*return* dalam variabel c. Kemudian, terdapat for loop yang dimulai dari i = 1 hingga i = jumlahIterasi. Lalu, nilai awal dari a dan b akan dimasukkan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fa dan fb. Jika nilai dari fa sama dengan nilai fb, maka akan menyebabkan pembagian dengan nol sehingga terdapat *if condition* yang berisi error pada *source code* di atas. Setelah itu, dilakukan pencarian nilai c dengan rumus sebagai berikut.

$$c = b - \frac{fb \times (b - a)}{(fb - fa)}$$

Kemudian, nilai dari c tersebut dimasukan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* f1/f2 yang hasilnya akan disimpan ke dalam variabel fc. Iterasi akan berhenti jika absolute (c-b) kurang dari atau sama dengan batas error yang telah ditentukan sebelumnya atau nilai dari fc sama dengan nol. Maka dari itu, terdapat if *else condition* untuk memberhentikan looping tersebut dimana jika kondisi if terpenuhi maka akan mengembalikan nilai sedangkan jika kondisi else terpenuhi maka nilai a akan digantikan oleh nilai b dan nilai b akan digantikan oleh nilai c. Statement pada *else condition* akan terus dilakukan hingga iterasi berhenti, yaitu jika kondisi pada syarat-syarat berhentinya iterasi berhasil dipenuhi.

- **Metode Newton Raphson**

```
% M0521003 - Adi Prasetya

function b = NewtonRaphson_edited(f,a, jumlahIterasi, batasError)
tic;
  syms x;
  f_diff = matlabFunction(diff(sym(f),'x'));
  fprintf(' Iter\ta\t\tf(a)\t\tf_diff(b)\tb\t\tfb\n');
  for i=1:jumlahIterasi
    fa = f(a);
    fdiffa = f_diff(a);
    if fdiffa==0
      error("Pembagian dengan 0.")
    endif

    b = a-fa/fdiffa;
    fb = f(b);
    fprintf (' %d\t%f\t%f\t%f\t%f\t%f\n',
             i,a,fa,fdiffa,b,fb);
    if abs(b-a)<= batasError || fb==0
      toc;
      return;
    else
      a=b;
    endif
  endfor
end
```

Pada *source code* di atas, terdapat *function* NewtonRaphson(f, a, jumlahIterasi, batasError) dimana hasil dari operasi dalam *function* tersebut akan di-*return* ke dalam variabel c. Kemudian, terdapat pendeklarasian variabel x sebagai variabel symbolic/independent dari fungsi yang dimiliki. Kemudian, dilakukan pencarian fungsi turunan yang hasilnya disimpan dalam variabel f_diff. Lalu, dilakukan iterasi dari i = 1 hingga i = jumlah iterasi. Lalu, nilai awal dari a akan dimasukkan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* g yang hasilnya akan disimpan ke dalam variabel fa. Dalam rumus metode Newton Raphon akan terdapat operasi pembagian fa dengan fdiffa, maka dari itu hasil dari fdiffa tidak boleh sama dengan nol. Untuk mengantisipasi hal tersebut, terdapat *if condition* dimana jika fdiffa sama dengan nol, maka iterasi berhenti dan memunculkan pesan error. Setelah itu, dilakukan pencarian nilai b dengan rumus sebagai berikut.

$$b = a - \frac{fa}{fdiffa}$$

Kemudian, nilai dari b tersebut dimasukkan ke dalam *function* yang sudah dideklarasikan sebelumnya, yaitu *function* g1 atau g2 yang hasilnya akan disimpan dalam variabel fb . Iterasi akan berhenti jika absolute (b-a) kurang dari atau sama dengan batas error yang telah ditentukan sebelumnya atau nilai dari fb sama dengan nol. Maka dari itu, terdapat if *else condition* untuk memberhentikan looping tersebut dimana jika kondisi if terpenuhi maka akan mengembalikan nilai sedangkan jika kondisi else terpenuhi maka nilai a akan digantikan oleh nilai b Statement pada *else condition* akan terus dilakukan hingga iterasi berhenti, yaitu jika kondisi pada syarat-syarat berhentinya iterasi berhasil dipenuhi.

## 2. Analisis *Source code* Soal 2

### ➢ Metode Fixed Point Itteration (FPI)

```
% M0521003 - Adi Prasetya

function b = FPI(g,a,jumlahIterasi,batasError)
tic;
  syms x;
  g_diff = matlabFunction(diff(sym(g),'x'));
  c_score = abs(g_diff(a))
  if c_score < 0 || c_score > 1
    error("Fungsi tidak layak digunakan.")
  endif
  fprintf(' Iter\ta\t\tb\t\tgb\n');
  for i=1:jumlahIterasi
    b = g(a);
    gb = g(b);
    fprintf (' %d\t%f\t%f\t%f\n',
             i,a,b,gb);
    if abs(b-a)<=batasError || gb==0
      toc;
      return;
    else
      a=b;
    endif
  endfor
end
```

Pada *source code* di atas, terdapat *function* FPI(g, a, jumlahIterasi, batasError) dimana hasil dari operasi dalam *function* tersebut akan di-*return* ke dalam variabel b. Kemudian, terdapat pendeklarasian variabel x sebagai variabel symbolic/independent dari fungsi yang dimiliki. Kemudian, dilakukan pencarian fungsi turunan yang hasilnya disimpan dalam variabel g_diff. Pencarian nilai c_score dilakukan dengan cara memasukkan nilai awal ke dalam fungsi turunan yang sudah didapatkan sebelumnya kemudian hasilnya diabsolute dan dimasukkan ke dalam varibel c_score. Salah satu syarat fungsi layak digunakan adalah $0 < |g'(a)| < 1$ maka dari itu terdapat *if condition* dimana jika nilai c_score kurang dari nol atau lebih dari nol maka fungsi tersebut tidak layak digunakan. Kemudian terdapat for loop yang dimulai dari i = 1 hingga i = jumlah iterasi. Kemudian, nilai dari a dimasukkan ke dalam *function* g yang sudah dideklarasikan sebelumnya, yaitu *function* g1 atau g2 yang hasilnya akan disimpan dalam variabel b dan nilai dari b dimasukkan ke dalam *function* g yang sudah dideklarasikan sebelumnya, yaitu *function* g1 atau g2 yang hasilnya akan disimpan dalam variabel gb. Syarat berhentinya iterasi adalah jika nilai absolute (b-a) kurang dari nilai batas error atau nilai gb sama dengan nol. Jika syarat pada *if condition* ini terpenuhi maka nilai akan di-*return*, sedangkan jika tidak terpenuhi maka nilai b akan digantikan dengan nilai a yang akan terus dilakukan hingga *if condition* terpenuhi.

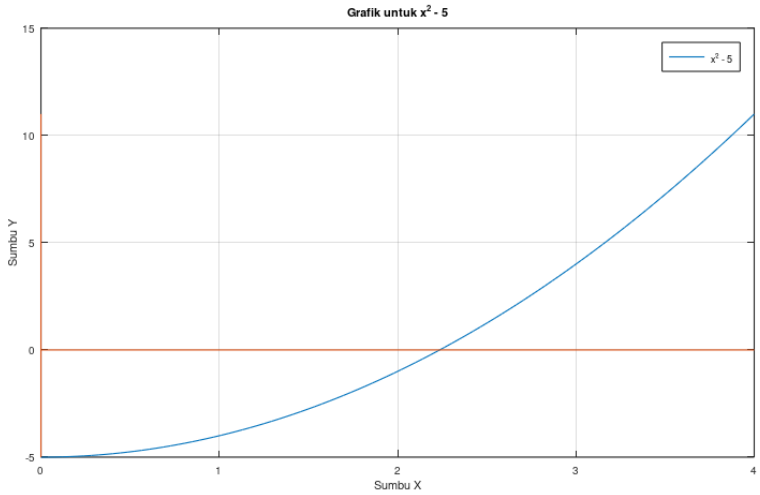
# BAB II

# ANALISIS PRAKTIKUM

## 1. Analisis Praktikum Soal 1

### a) Soal A

- Grafik A




Gambar di atas adalah plot yang dihasilkan oleh fungsi di bawah ini.

$$f2(x) = x^2 - 5$$

Berdasarkan grafik tersebut, maka dapat terlihat bahwa nilai akar mendekati 2.2. Maka dari itu, nilai awal yang digunakan untuk metode Biseksi, metode Regula Falsi, dan metode Secant adalah 2.1 untuk nilai a dan 2.4 untuk nilai b, sedangkan nilai awal yang digunakan untuk metode Newton Raphson adalah 2.

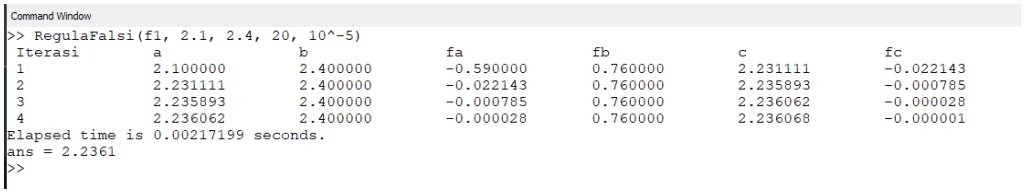
- Metode Biseksi

```
Command Window
>> Biseksi(f1, 2.1, 2.4, 20, 10^-5)
 Iterasi        a           b           fa          fb          c           fc
 1           2.100000    2.400000   -0.590000    0.760000    2.250000    0.062500
 2           2.100000    2.250000   -0.590000    0.062500    2.175000   -0.269375
 3           2.175000    2.250000   -0.269375    0.062500    2.212500   -0.104844
 4           2.212500    2.250000   -0.104844    0.062500    2.231250   -0.021523
 5           2.231250    2.250000   -0.021523    0.062500    2.240625    0.020400
 6           2.231250    2.240625   -0.021523    0.020400    2.235938   -0.000583
 7           2.235938    2.240625   -0.000583    0.020400    2.238281    0.009903
 8           2.235938    2.238281   -0.000583    0.009903    2.237109    0.004658
 9           2.235938    2.237109   -0.000583    0.004658    2.236523    0.002037
 10          2.235938    2.236523   -0.000583    0.002037    2.236230    0.000727
 11          2.235938    2.236230   -0.000583    0.000727    2.236084    0.000072
 12          2.235938    2.236084   -0.000583    0.000072    2.236011   -0.000256
 13          2.236011    2.236084   -0.000256    0.000072    2.236047   -0.000092
 14          2.236047    2.236084   -0.000092    0.000072    2.236066   -0.000010
 15          2.236066    2.236084   -0.000010    0.000072    2.236075    0.000031
Elapsed time is 0.00806403 seconds.
ans = 2.2361
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* Biseksi(f1, 2.1, 2.4, 20, $10^{-5}$) dimana f1 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 2.1 adalah nilai awal untuk titik

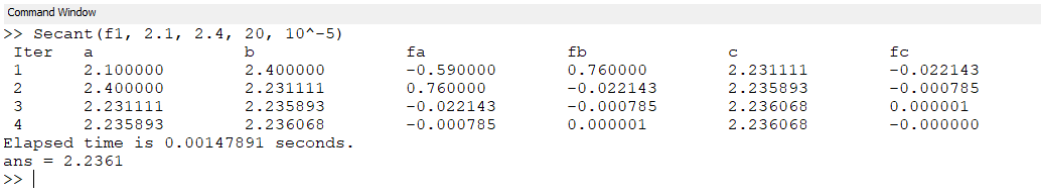
a, 2.4 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-14 dikarenakan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 2.236075 dengan waktu komputasi sebesar 0.00806403 s.

- Metode Regula Falsi




Pada *command window* di atas, dilakukan pemanggilan *function* RegulaFalsi(f1, 2.1, 2.4, 20, $10^{-5}$) dimana f1 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 2.1 adalah nilai awal untuk titik a, 2.4 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-4 dikarenakan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 2.236068 dengan waktu komputasi sebesar 0.00217199 s.

- Metode Secant




Pada *command window* di atas, dilakukan pemanggilan *function* Secant(f1, 2.1, 2.4, 20, $10^{-5}$) dimana f1 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 2.1 adalah nilai awal untuk titik a, 2.4 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-4 dikarenakan nilai fc sama dengan nol dan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 2.236068 dengan waktu komputasi sebesar 0.00147891 s.

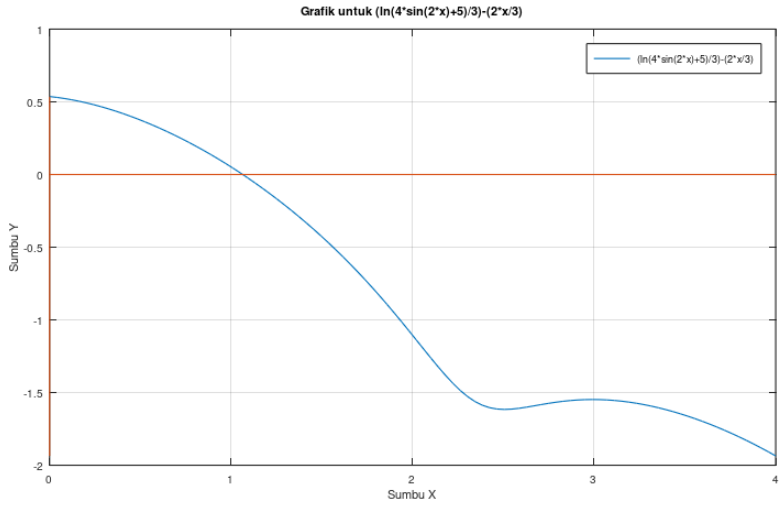
- Metode Newton Raphson

```
Command Window
>> NewtonRaphson_edited(f1, 2, 20,10^-5)
 Iter    a               f(a)            f_diff(b)       b               fb
 1       2.000000        -1.000000       4.000000        2.250000        0.062500
 2       2.250000        0.062500        4.500000        2.236111        0.000193
 3       2.236111        0.000193        4.472222        2.236068        0.000000
 4       2.236068        0.000000        4.472136        2.236068        0.000000
Elapsed time is 0.217268 seconds.
ans = 2.2361
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* NewtonRaphson_edited(f1, 2, 20, $10^{-5}$) dimana f1 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 2 adalah nilai awal untuk titik a, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-4 dikarenakan nilai fb sama dengan nol dan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 2.236068 dengan waktu komputasi sebesar 0.217268 s.

**b) Soal B**

- Grafik B




Gambar di atas adalah plot yang dihasilkan oleh fungsi di bawah ini.

$$f1(x) = \frac{\ln(4\sin(2x) + 5)}{3} - \frac{2x}{3}$$

Berdasarkan grafik tersebut, maka dapat terlihat bahwa nilai akar mendekati 1.1. Maka dari itu, nilai awal yang digunakan untuk metode Biseksi, metode Regula Falsi, dan metode Secant adalah 0.5 untuk nilai a dan 1.5 untuk nilai b, sedangkan nilai awal yang digunakan untuk metode Newton Raphson adalah 2.

- Metode Biseksi

```
Command Window
>> Biseksi(f2, 0.5, 1.5, 20, 10^-5)
 Iterasi        a               b               fa              fb              c
 1              0.500000        1.500000        0.374721        -0.427865       1.000000
 2              1.000000        1.500000        0.052026        -0.427865       1.250000
 3              1.000000        1.250000        0.052026        -0.166449       1.125000
 4              1.000000        1.125000        0.052026        -0.052206       1.062500
 5              1.062500        1.125000        0.001128        -0.052206       1.093750
 6              1.062500        1.093750        0.001128        -0.025231       1.078125
 7              1.062500        1.078125        0.001128        -0.011975       1.070312
 8              1.062500        1.070312        0.001128        -0.005404       1.066406
 9              1.062500        1.066406        0.001128        -0.002133       1.064453
 10             1.062500        1.064453        0.001128        -0.000502       1.063477
 11             1.063477        1.064453        0.000314        -0.000502       1.063965
 12             1.063477        1.063965        0.000314        -0.000094       1.063721
 13             1.063721        1.063965        0.000110        -0.000094       1.063843
 14             1.063843        1.063965        0.000008        -0.000094       1.063904
 15             1.063843        1.063904        0.000008        -0.000043       1.063873
 16             1.063843        1.063873        0.000008        -0.000018       1.063858
 17             1.063843        1.063858        0.000008        -0.000005       1.063850
Elapsed time is 0.017741 seconds.
ans = 1.0639
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* Biseksi(f2, 0.5, 1.5, 20, $10^{-5}$) dimana f2 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 0.5 adalah nilai awal untuk titik a, 1.5 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-17 dikarenakan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 1.063850 dengan waktu komputasi sebesar 0.017741 s.

- Metode Regula Falsi

```
Command Window
>> RegulaFalsi(f2, 0.5, 1.5, 20, 10^-5)
 Iterasi        a               b               fa              fb              c
 1              0.500000        1.500000        0.374721        -0.427865       0.966892
 2              0.966892        1.500000        0.078018        -0.427865       1.049108
 3              1.049108        1.500000        0.012237        -0.427865       1.061645
 4              1.061645        1.500000        0.001840        -0.427865       1.063523
 5              1.063523        1.500000        0.000275        -0.427865       1.063803
 6              1.063803        1.500000        0.000041        -0.427865       1.063845
 7              1.063845        1.500000        0.000006        -0.427865       1.063851
Elapsed time is 0.00422502 seconds.
ans = 1.0639
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* RegulaFalsi(f2, 0.5, 1.5, 20, $10^{-5}$) dimana f2 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 0.5 adalah nilai awal untuk titik a, 1.5 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-7 dikarenakan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 1.063851 dengan waktu komputasi sebesar 0.00422502 s.

- Metode Secant

```
Command Window
>> Secant(f2, 0.5, 1.5, 20, 10^-5)
 Iter    a              b              fa             fb             c              fc
 1       0.500000       1.500000       0.374721       -0.427865      0.966892       0.078018
 2       1.500000       0.966892       -0.427865      0.078018       1.049108       0.012237
 3       0.966892       1.049108       0.078018       0.012237       1.064403       -0.000459
 4       1.049108       1.064403       0.012237       -0.000459      1.063849       0.000003
 5       1.064403       1.063849       -0.000459      0.000003       1.063852       0.000000
Elapsed time is 0.00310993 seconds.
ans = 1.0639
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* Secant(f2, 0.5, 1.5, 20, $10^{-5}$) dimana f2 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 0.5 adalah nilai awal untuk titik a, 1.5 adalah nilai awal untuk titik b, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-5 dikarenakan nilai fc sama dengan nol dan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 1.063852 dengan waktu komputasi sebesar 0.00310993 s.

- Metode Newton Raphson

```
Command Window
>> NewtonRaphson_edited(f2, 2, 20, 10^-5)
 Iter    a              f(a)           f_diff(b)      b              fb
 1       2.000000       -1.106850      -1.550212      1.286001       -0.201296
 2       1.286001       -0.201296      -0.980431      1.080686       -0.014138
 3       1.080686       -0.014138      -0.845088      1.063957       -0.000088
 4       1.063957       -0.000088      -0.834619      1.063852       -0.000000
 5       1.063852       -0.000000      -0.834553      1.063852       0.000000
Elapsed time is 0.484264 seconds.
ans = 1.0639
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* NewtonRaphson_edited(f2, 2, 20, $10^{-5}$) dimana f2 merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 2 adalah nilai awal untuk titik a, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada iterasi yang ke-4 dikarenakan nilai fb sama dengan nol dan galat yang dihasilkan telah memenuhi syarat untuk berhentinya iterasi. Nilai akhir yang didapatkan adalah 1.063852 dengan waktu komputasi sebesar 0.484264 s.

**KESIMPULAN UNTUK SOAL 1 :**

- **SOAL A**

  Berikut ini adalah jenis metode dan jumlah iterasi berdasarkan hasil percobaan sebelumnya.

| Jenis Metode | Jumlah Iterasi |
|---|---|
| Regula Falsi | 4 |
| Secant | 4 |
| Newton Raphson | 4 |
| Biseksi | 15 |

  Berikut ini adalah jenis metode dan waktu komputasi berdasarkan hasil percobaan sebelumnya.

| Jenis Metode | Waktu Komputasi (s) |
|---|---|
| Secant | 0.00147891 |
| Regula Falsi | 0.00217199 |
| Biseksi | 0.00806403 |
| Newton Raphson | 0.217268 |

  Berdasarkan kedua tabel di atas, metode Secant merupakan metode dengan jumlah iterasi paling sedikit dan waktu komputasi paling cepat.

- **SOAL B**

  Berikut ini adalah jenis metode dan jumlah iterasi berdasarkan hasil percobaan sebelumnya.

| Jenis Metode | Jumlah Iterasi |
|---|---|
| Secant | 5 |
| Newton Raphson | 5 |
| Regula Falsi | 7 |
| Biseksi | 17 |

  Berikut ini adalah jenis metode dan waktu komputasi berdasarkan hasil percobaan sebelumnya.

| Jenis Metode | Waktu Komputasi (s) |
|---|---|
| Secant | 0.00310993 |
| Regula Falsi | 0.00422502 |
| Biseksi | 0.017741 |
| Newton Raphson | 0.484264 |

  Berdasarkan kedua tabel tabel di atas, metode Secant merupakan metode dengan jumlah iterasi paling sedikit dan waktu komputasi paling cepat.

## 2. Analisis Praktikum Soal 2

Diketahui terdapat sebuah fungsi sebagai berikut.

$$y = x^2 + \sin(x) - 10$$

Berikut ini adalah variasi fungsi dan hasil penerapan dengan metode *Fixed point iteration* pada kedua variasi fungsi tersebut.

### a) Variasi Fungsi 1

Berikut ini adalah hasil dari variasi fungsi 1.

$$y = x^2 + \sin(x) - 10$$
$$x^2 + \sin(x) - 10 = 0$$
$$x^2 = 10 - \sin(x)$$
$$x = \frac{10}{x} - \frac{\sin(x)}{x}$$

Lalu, dilakukan pendeklarasian *function* anonymous pada *command window*.

```
Command Window
>> g1 = @(x) (10/x) - (sin(x)/x)
g1 =

@(x) (10 / x) - (sin (x) / x)
```

Pada *command window* di atas dilakukan ***function*** **anonymous** untuk dengan parameter (x) dengan isi *function* adalah sebagai berikut.

$$x = \frac{10}{x} - \frac{\sin(x)}{x}$$

```
Command Window
>> FPI(g1, 2, 20, 10^-5)
c_score = 2.0646
error: Fungsi tidak layak digunakan.
error: called from
    FPI at line 7 column 5
>>
```

Pada *command window* di atas, terlihat bahwa **fungsi tidak layak digunakan**. Hal ini karena nilai c_score sebesar 2.0646 dimana seharusnya nilai dari c_score berada di antara 0 dan 1.

**b) Variasi Fungsi 2**

Berikut ini adalah hasil dari variasi fungsi 2.

$$y = x^2 + \sin(x) - 10$$
$$x^2 + \sin(x) - 10 = 0$$
$$x^2 = 10 - \sin(x)$$
$$x = \sqrt{10 - \sin(x)}$$

Lalu, dilakukan pendeklarasian *function* anonymous pada *command window*.

```
Command Window
>> g2 = @(x) sqrt(10-sin(x))
g2 =

@(x) sqrt (10 - sin (x))
```

Pada *command window* di atas dilakukan ***function*** **anonymous** untuk dengan parameter (x) dengan isi *function* adalah sebagai berikut.

$$x = \frac{10}{x} - \frac{\sin(x)}{x}$$

```
Command Window
>>  FPI(g2, 2, 20, 10^-5)
c_score = 0.069011
 Iter    a               b               gb
 1       2.000000        3.015079        3.142264
 2       3.015079        3.142264        3.162384
 3       3.142264        3.162384        3.165563
 4       3.162384        3.165563        3.166065
 5       3.165563        3.166065        3.166144
 6       3.166065        3.166144        3.166157
 7       3.166144        3.166157        3.166159
 8       3.166157        3.166159        3.166159
Elapsed time is 0.254586 seconds.
ans = 3.1662
>>
```

Pada *command window* di atas, dilakukan pemanggilan *function* FPI(g2, 2, 20, 10^-5) dimana g2 merupakan *anonymous function* yang sudah ditentukan sebelumnya, 2 adalah nilai untuk nilai awal, 20 adalah jumlah iterasi yang dilakukan, dan 10^-5 adalah batas toleransi error yang dapat diterima. Iterasi berhenti pada saat iterasi ke-8 dikarenakan nilai galat dari (b-a) sudah memenuhi batas toleransi error yang sudah ditentukan sebelumnya. Nilai akhir yang didapatkan adalah sebesar 3.166159 dengan waktu komputasi sebesar 0.254586 s.

**KESIMPULAN SOAL 2 :**

Berdasarkan hasil percobaan dengan metode *fixed point iteration*, didapatkan bahwa nilai akar terkecil yang memenuhi adalah 3.1662 dan waktu komputasi sebesar 0.254586. Jika dilihat dari waktu komputasinya maka dapat dsisimpulkan bahwa metode *fixed point iteration* cukup lambat dalam mencari nilai akar dalam suatu sistem persamaan non linear.