LAPORAN PRAKTIKUM MATA KULIAH

METODE NUMERIK

PRAKTIKUM 8 - PERSAMAAN DIFFERENSIAL BIASA

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 Pencarian Fungsi Eksak**

   Di soal diketahui bahwa terdapat sebuah persamaan differensial sebagai berikut.

$$\frac{dx}{dy} + 2y = 0$$

   Dari persamaan diffensial di atas, dilakukan operasi untuk mencari fungsi eksak yang akan digunakan sebagai $f_{asli.}$ Proses pencarian fungsi asli adalah sebagai berikut.

$$\frac{dx}{dy} + 2y = 0$$

$$dx = -2y\, dy$$

$$\int dx = \int -2y\, dy$$

$$x = -y^2 + C_2$$

$$y^2 = C_2 - x$$

$$y = \sqrt{C_2 - x}$$

   Diketahui y(0) = 2, maka

$$y = \sqrt{C_2 - x}$$

$$2 = \sqrt{C_2 - 0}$$

$$2 = \sqrt{C_2}$$

$$C_2 = 4$$

   Sehingga didapatkan sebagai berikut.

$$y = \sqrt{C_2 - x}$$

$$y = \sqrt{4 - x}$$

2. Analisis Pencarian $\frac{dy}{dx}$ untuk Metode Euler, Metode Heun, dan Metode Runge Kutta Orde 4

Di soal diketahui bahwa terdapat sebuah persamaan differensial sebagai berikut.

$$\frac{dx}{dy} + 2y = 0$$

Maka,

$$\frac{dx}{dy} + 2y = 0$$

$$\frac{dx}{dy} = -2y$$

$$dx = -2y\, dy$$

$$\frac{1}{-2y} = \frac{dy}{dx}$$

$$\frac{dy}{dx} = \frac{1}{-2y}$$

Sehingga,

$$\frac{dy}{dx} = \frac{1}{-2y}$$

Turunan di atas akan digunakan sebagai fungsi f pada perhitungan dengan metode Euler, metode Heun, dan metode Runge Kutta Orde 4.

3. Analisis *Source code* Metode Euler

- Plot

```
#M0521003 - Adi Prasetya

function [x,y] = Euler (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Euler\n i\txn\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  fprintf(' %d\t%f\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    y(i+1) = y(i)+h*f(x(i),y(i));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\n',
            i, x(i+1), y(i+1));
  endfor
##  y=y(end);
end
```

Pada *source code* di atas, terdapat ***function* Euler (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel [x,y].** Dilakukan pencarian nilai dari h dengan rumus di bawah ini.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x dan nilai dari y0 dimasukkan ke dalam variabel y. Kemudian, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hingga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$y_{i+1} = y_i + h \times f(x_i, y_i)$$

$$x_{i+1} = x_i + h$$

Pada rumus untuk mencari y_{i+1} dan x_{i+1} akan selalu menggunakan nilai , y_i dan x_i sebelumnya. Setelah itu, nilai dari x dan y dari setiap iterasi akan disimpan dalam array pada workspace dengan variabel tertentu sehingga nilai-nilai x dan y dalam array tersebut dapat digunakan dalam *plotting*.

- Galat

```
#M0521003 - Adi Prasetya

##function [x,y] = Euler (f,a,b,y0,N)

function y = Euler (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Euler\n i\txn\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  fprintf(' %d\t%f\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    y(i+1) = y(i)+h*f(x(i),y(i));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\n',
            i, x(i+1), y(i+1));
  endfor
  y = y(end);
end
```

Pada *source code* di atas, terdapat *function* **Euler (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel y**. Dilakukan pencarian nilai dari h dengan rumus di bawah ini.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x dan nilai dari y_0 dimasukkan ke dalam variabel y. Kemudian, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hingga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$y_{i+1} = y_i + h \times f(x_i, y_i)$$

$$x_{i+1} = x_i + h$$

Pada rumus untuk mencari y_{i+1} dan x_{i+1} akan selalu menggunakan nilai , y_i dan x_i sebelumnya. Setelah itu, ketika iterasi sudah selesai maka nilai y pada iterasi terakhir akan dapat diketahui yang di-*return* dalam variabel y. Kemudian, nilai dalam variabel y tersebut akan disimpan ke dalam workspace dengan variabel tertentu yang akan digunakan dalam perhitungan galat.

4. Analisis *Source code* Metode Heun

- Plot

```
#M0521003 - Adi Prasetya

function [x,y] = Heun (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Heun\n i\txn\t\tk1\t\tk2\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  k = [];
  fprintf(' %d\t%f\t-\t\t-\t\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    k(1) = f(x(i),y(i));
    k(2) = f(x(i)+h,y(i)+k(1)*h);
    y(i+1) = y(i)+h/2*(k(1)+k(2));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\t%f\n',
            i, x(i+1), k(1), k(2), y(i+1));
  endfor
##  y=y(end);
end
```

Pada *source code* di atas, terdapat ***function* Heun (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel [x,y].** Dilakukan pencarian nilai dari h dengann rumus di bawah ini.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x, nilai dari y_0 dimasukkan ke dalam variabel y, dan dilakukan pendeklarasian variabel k. Lalu, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hinga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai k_1, k_2, y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + h, y_i + h \times k_1)$$

$$y_{i+1} = y_i + h(k_1 + k_2)$$

$$x_{i+1} = x_i + h$$

Pada setiap rumus di atas untuk akan selalu menggunakan nilai x_i dan y_i sebelumnya. Setelah itu, nilai dari x dan y dari setiap iterasi akan disimpan dalam array pada workspace dengan variabel tertentu sehingga nilai-nilai x dan y dalam array tersebut dapat digunakan dalam *plotting*.

- Galat

```
#M0521003 - Adi Prasetya

##function [x,y] = Heun (f,a,b,y0,N)

function y = Heun (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Heun\n i\txn\t\tk1\t\tk2\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  k = [];
  fprintf(' %d\t%f\t-\t\t-\t\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    k(1) = f(x(i),y(i));
    k(2) = f(x(i)+h,y(i)+k(1)*h);
    y(i+1) = y(i)+h/2*(k(1)+k(2));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\t%f\t%f\n',
            i, x(i+1), k(1), k(2), y(i+1));
  endfor
  y=y(end);
end
```

Pada *source code* di atas, terdapat ***function* Heun (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel y.** Dilakukan pencarian nilai dari h dengan rumus di bawah ini.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x, nilai dari y_0 dimasukkan ke dalam variabel y, dan dilakukan pendeklarasian variabel k. Lalu, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hinga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai k_1, k_2, y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + h, y_i + h \times k_1)$$

$$y_{i+1} = y_i + h(k_1 + k_2)$$

$$x_{i+1} = x_i + h$$

Pada setiap rumus di atas untuk akan selalu menggunakan nilai x_i dan y_i sebelumnya. Setelah itu, ketika iterasi sudah selesai maka nilai y pada iterasi terakhir akan dapat diketahui yang di-*return* dalam variabel y. Kemudian, nilai

dalam variabel y tersebut akan disimpan ke dalam workspace dengan variabel tertentu yang akan digunakan dalam perhitungan galat.

5. Analisis *Source code* Metode Runge Kutta

- Plot

```
#M0521003 - Adi Prasetya

function [x,y] = Runge4 (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Runge-Katta Orde 4\n i\txn\t\tk1\t\tk2\t\tk3\t\tk4\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  k = [];
  fprintf(' %d\t%f\t-\t\t-\t\t-\t\t-\t\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    k(1) = f(x(i),y(i));
    k(2) = f(x(i)+h/2,y(i)+k(1)*h/2);
    k(3) = f(x(i)+h/2,y(i)+k(2)*h/2);
    k(4) = f(x(i)+h,y(i)+k(3)*h);
    y(i+1) = y(i)+h/6*(k(1)+2*k(2)+2*k(3)+k(4));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\t%f\t%f\t%f\n',
            i, x(i+1), k(1), k(2), k(3), k(4), y(i+1));
  endfor
##  y=y(end);
end
```

Pada *source code* di atas, terdapat ***function* Runge4 (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel [x, y]**. Dilakukan pencarian nilai dari h dengan rumus sebagai berikut.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x, nilai dari y_0 dimasukkan ke dalam variabel y, dan dilakukan pendeklarasian variabel k. Lalu, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hinga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai k_1, k_2, y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2} \times k_1)$$

$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2} \times k_1)$$

$$k_4 = f(x_i + h, y_i + k_3 \times h)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$x_{i+1} = x_i + h$$

Pada setiap rumus di atas untuk akan selalu menggunakan nilai x_i dan y_i sebelumnya. Setelah itu, nilai dari x dan y dari setiap iterasi akan disimpan dalam array pada workspace dengan variabel tertentu sehingga nilai-nilai x dan y dalam array tersebut dapat digunakan dalam *plotting*.

- Galat

```
#M0521003 - Adi Prasetya

##function [x,y] = Runge4(f,a,b,y0,N)

function y = Runge4 (f,a,b,y0,N)
  h = (b-a)/N;

  fprintf('\n Hasil Perhitungan Metode Runge-Katta Orde 4\n i\txn\t\tk1\t\tk2\t\tk3\t\tk4\t\tyn\n')
  #Memulai iterasi ke-0
  x = a;
  y = y0;
  k = [];
  fprintf(' %d\t%f\t-\t\t-\t\t-\t\t-\t\t%f\n',
          0, x, y);
  #Memulai iterasi pertama sampai selesai
  for i = 1:N;
    k(1) = f(x(i),y(i));
    k(2) = f(x(i)+h/2,y(i)+k(1)*h/2);
    k(3) = f(x(i)+h/2,y(i)+k(2)*h/2);
    k(4) = f(x(i)+h,y(i)+k(3)*h);
    y(i+1) = y(i)+h/6*(k(1)+2*k(2)+2*k(3)+k(4));
    x(i+1) = x(i)+h;
    fprintf(' %d\t%f\t%f\t%f\t%f\t%f\n',
            i, x(i+1), k(1), k(2), k(3), k(4), y(i+1));
  endfor
  y=y(end);
end
```

Pada *source code* di atas, terdapat ***function* Runge4 (f, a, b, y0, N)** dimana hasil operasi dalam *function* tersebut akan di-*return* dalam **variabel y.** Dilakukan pencarian nilai dari h dengan rumus sebagai berikut.

$$h = \frac{(b-a)}{N}$$

Kemudian, dilakukan *assignment* untuk nilai dari a dimasukkan ke dalam variabel x, nilai dari y_0 dimasukkan ke dalam variabel y, dan dilakukan pendeklarasian variabel k. Lalu, terdapat *for loop* dimana di dalamnya dilakukan iterasi yang dimulai dari i = 1 hinga i = N. Pada setiap iterasinya akan dilakukan operasi untuk mencari nilai k_1, k_2, y_{i+1} dan x_{i+1} dengan rumus di bawah ini.

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2} \times k_1)$$

$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2} \times k_1)$$

$$k_4 = f(x_i + h, y_i + k_3 \times h)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$x_{i+1} = x_i + h$$

Pada setiap rumus di atas untuk akan selalu menggunakan nilai x_i dan y_i sebelumnya. Setelah itu, ketika iterasi sudah selesai maka nilai y pada iterasi terakhir akan dapat diketahui yang di-*return* dalam variabel y. Kemudian, nilai dalam variabel y tersebut akan disimpan ke dalam workspace dengan variabel tertentu yang akan digunakan dalam perhitungan galat.

6. Analisis *Source code Plotting*

```
% M0521003 - Adi Prasetya
% Plot Kurva untuk metode eksak dan ketiga metode lainnya

x=linspace(0,2);
fasli = @(x) sqrt(4-x)
y=fasli(x);

plot(x,y, 'r', x_euler, y_euler, 'g', x_heun, y_heun, 'y', x_runge4, y_runge4, 'b')

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

%Anotasi
legend ("eksak", "euler", "heun", "runge4")
title('Grafik Fungsi')
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 2. Kemudian, terdapat pendeklarasian *anonymous function* dengan parameter (x) yang memiliki isi sebagai berikut.

$$f_{asli}(x) = \sqrt{4 - x}$$

Lalu, terdapat *function* plot yang akan mempresentasikan hasil perhitungan dari metode eksak dengan warna merah, metode euler dengan warna hijau, metode heun dengan warna kuning, dan metode Runge kutta orde 4 dengan warna biru pada grafik yang terdiri atas sumbu X dan sumbuY. 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 warna garis dan keterangan dari metode pada warna garis tersebut. 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

7. Analisis *Source code* Galat

```
% M0521003 - Adi Prasetya
% Perhitungan Galat pada TItik Terakhir dan Keseluruhan Iterasi

eksak = fasli(2);
fprintf('Hasil Perhitungan dengan 3 Metode : \n')

digitsOld = digits(12);

eksak_8 = vpa(eksak)
euler_8 = vpa(euler)
heun_8 = vpa(heun)
runge4_8 = vpa(runge4)

fprintf('\nHasil Perhitungan Galat pada Titik Terakhir: \n')
err_euler = abs((eksak-euler)/eksak)
err_heun = abs((eksak-heun)/eksak)
err_runge4 = abs((eksak-runge4)/eksak)

fprintf('\nHasil Perhitungan Galat pada Keseluruhan Iterasi: \n')
all_error_euler = vpa(mean(abs((fasli(x_euler)-y_euler)-fasli(x_euler))))
all_error_heun = vpa(mean(abs((fasli(x_heun)-y_heun)-fasli(x_heun))))
all_error_runge4 = vpa(mean(abs((fasli(x_runge4)-y_runge4)-fasli(x_runge4))))
```

Pada *source code* di atas, terdapat *function* fasli(2) yang hasil dari operasi *function* tersebut akan disimpan dalam variabel eksak. Kemudian, terdapat digitsOld = digits(12) yang digunakan untuk menentukan jumlah digit yang akan ditampilkan. Lalu, terdapat *function* vpa ( ) yang di dalamnya terdapat variabel eksak, euler, heun, dan runge4 yang digunakan untuk melakukan konversi variabel-variabel tersebut ke dalam symbolic expression. Tujuan dilakukan konversi ini adalah agar jumlah digit yang dapat dari variabel eksak_8, euler_8, heun_8, dan runge4_8 adalah 12 digits.

Kemudian, terdapat *function* abs( ) yang di dalamnya terdapat operasi untuk mencari error relatif dari metode euler, metode heun, dan metode runge kutta orde 4. Lalu, terdapat *function* mean( ) yang di dalamnya terdapat operasi untuk mencari besar error dari keseluruhan iterasi yang kemudian hasilnya dikonversi ke dalam symbolic expression. Hasil perhitungan galat pada keseluruhan iterasi dari ketiga metode tersebut akan disimpan ke dalam variabel all_error_euler, all_error_heun, dan all_error_runge4.

BAB II

ANALISIS PRAKTIKUM (COMMAND WINDOW)

1. **Analisis Praktikum Pendahuluan**

- Pendeklarasian anonymous *function* f_{asli}

Dilakukan pendeklarasian anonymous *function* pada command window yang mana anonymous *function* ini akan digunakan dalam perhitungan pada metode eksak.

```
Command Window
>> fasli = @(x) (sqrt(4-x))
fasli =

@(x) (sqrt (4 - x))
```

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

$$f_{asli}(x) = \sqrt{4 - x}$$

- Pendeklarasian anonymous *function* f

Dilakukan pendeklarasian anonymous *function* pada command window yang mana anonymous *function* ini akan digunakan dalam perhitungan pada metode euler, metode heun, dan metode runge kutta orde 4.

```
Command Window
>> f = @(x,y) 1/ (-2*y)
f =

@(x, y) 1 / (-2 * y)

>>
```

Pada command window di atas, dilakukan pendeklarasian anonymous *function* dengan parameter (x,y) dengan isi *function* sebagai berikut.

$$f(x, y) = \frac{1}{-2y}$$

2. Analisis Praktikum Metode Euler

- Galat

```
Command Window
>> euler = Euler(f, 0, 2, 2, 10)

 Hasil Perhitungan Metode Euler
 i       xn              yn
 0       0.000000        2.000000
 1       0.200000        1.950000
 2       0.400000        1.898718
 3       0.600000        1.846051
 4       0.800000        1.791881
 5       1.000000        1.736074
 6       1.200000        1.678473
 7       1.400000        1.618895
 8       1.600000        1.557124
 9       1.800000        1.492903
 10      2.000000        1.425920
euler = 1.4259
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Euler(f, 0, 2, 2, 10) dimana f merupakan anonymous *function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Euler pada saat y di iterasi terakhir akan di-*return* dan disimpan ke dalam variabel euler yang nantinya akan digunakan dalam perhitungan Galat. Nilai y pada iterasi ke-10 adalah 1.425920.

- Plot

```
Command Window
>> [x_euler, y_euler] = Euler(f, 0, 2, 2, 10);

 Hasil Perhitungan Metode Euler
 i       xn              yn
 0       0.000000        2.000000
 1       0.200000        1.950000
 2       0.400000        1.898718
 3       0.600000        1.846051
 4       0.800000        1.791881
 5       1.000000        1.736074
 6       1.200000        1.678473
 7       1.400000        1.618895
 8       1.600000        1.557124
 9       1.800000        1.492903
 10      2.000000        1.425920
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Euler(f, 0, 2, 2, 10) dimana f merupakan anonymous *function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Euler pada setiap iterasinya akan di-*return* dan disimpan ke dalam variabel [x_euler, y_euler] yang nantinya akan digunakan dalam *plotting*.

3. Analisis Praktikum Metode Heun

- Galat

```
Command Window
>> heun = Heun(f, 0, 2, 2, 10)

 Hasil Perhitungan Metode Heun
 i       xn              k1              k2              yn
 0       0.000000        -               -               2.000000
 1       0.200000        -0.250000       -0.256410       1.949359
 2       0.400000        -0.256495       -0.263427       1.897367
 3       0.600000        -0.263523       -0.271052       1.843909
 4       0.800000        -0.271163       -0.279380       1.788855
 5       1.000000        -0.279508       -0.288525       1.732052
 6       1.200000        -0.288675       -0.298629       1.673321
 7       1.400000        -0.298807       -0.309874       1.612453
 8       1.600000        -0.310087       -0.322490       1.549196
 9       1.800000        -0.322748       -0.336781       1.483243
 10      2.000000        -0.337099       -0.353152       1.414218
heun = 1.4142
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Heun(f, 0, 2, 2, 10) dimana f merupakan anonymous *function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Heun pada saat y di iterasi terakhir akan di-*return* dan disimpan ke dalam variabel heun yang nantinya akan digunakan dalam perhitungan Galat. Nilai y pada iterasi ke-10 adalah 1.414218.

- Plot

```
Command Window
>> [x_heun, y_heun] = Heun(f, 0, 2, 2, 10);

 Hasil Perhitungan Metode Heun
 i       xn              k1              k2              yn
 0       0.000000        -               -               2.000000
 1       0.200000        -0.250000       -0.256410       1.949359
 2       0.400000        -0.256495       -0.263427       1.897367
 3       0.600000        -0.263523       -0.271052       1.843909
 4       0.800000        -0.271163       -0.279380       1.788855
 5       1.000000        -0.279508       -0.288525       1.732052
 6       1.200000        -0.288675       -0.298629       1.673321
 7       1.400000        -0.298807       -0.309874       1.612453
 8       1.600000        -0.310087       -0.322490       1.549196
 9       1.800000        -0.322748       -0.336781       1.483243
 10      2.000000        -0.337099       -0.353152       1.414218
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Heun(f, 0, 2, 2, 10) dimana f merupakan anonymous *function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Heun pada setiap iterasinya akan di-*return* dan disimpan ke dalam variabel [x_heun, y_heun] yang nantinya akan digunakan dalam *plotting*.

4. Analisis Praktikum Metode Runge Kutta

- Galat

```
Command Window
>> runge4 = Runge4(f, 0, 2, 2, 10)

 Hasil Perhitungan Metode Runge-Katta Orde 4
 i      xn             k1             k2             k3             k4             yn
 0      0.000000       -              -              -              -              2.000000
 1      0.200000       -0.250000      -0.253165      -0.253205      -0.256495      1.949359
 2      0.400000       -0.256495      -0.259915      -0.259961      -0.263523      1.897367
 3      0.600000       -0.263523      -0.267235      -0.267288      -0.271163      1.843909
 4      0.800000       -0.271163      -0.275210      -0.275272      -0.279508      1.788854
 5      1.000000       -0.279508      -0.283945      -0.284017      -0.288675      1.732051
 6      1.200000       -0.288675      -0.293568      -0.293652      -0.298807      1.673320
 7      1.400000       -0.298807      -0.304240      -0.304341      -0.310087      1.612452
 8      1.600000       -0.310087      -0.316167      -0.316289      -0.322749      1.549193
 9      1.800000       -0.322749      -0.329616      -0.329765      -0.337100      1.483240
 10     2.000000       -0.337100      -0.344939      -0.345126      -0.353553      1.414214
runge4 = 1.4142
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Runge4(f, 0, 2, 2, 10) dimana f merupakan anonymous *function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Runge4 pada saat y di iterasi terakhir akan di-*return* dan disimpan ke dalam variabel runge4 yang nantinya akan digunakan dalam perhitungan Galat. Nilai y pada iterasi ke-10 adalah 1.414214.

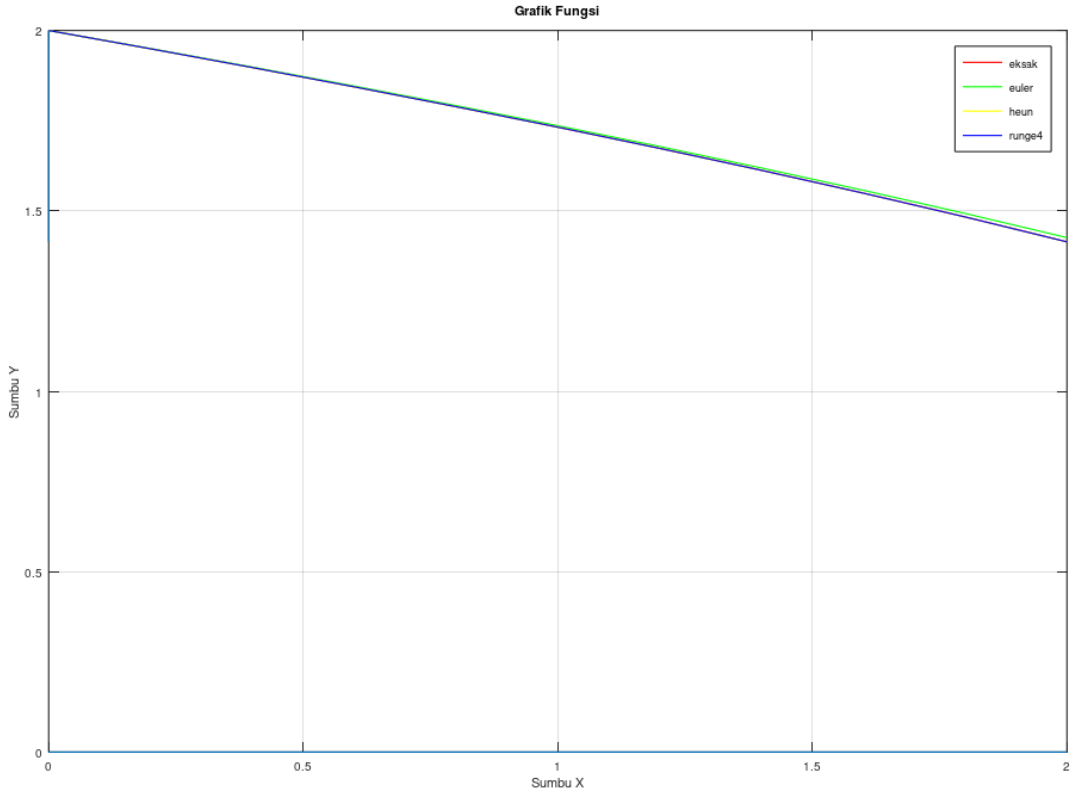
- Plot

```
Command Window
>> [x_runge4, y_runge4] = Runge4(f, 0, 2, 2, 10);

 Hasil Perhitungan Metode Runge-Katta Orde 4
 i      xn             k1             k2             k3             k4             yn
 0      0.000000       -              -              -              -              2.000000
 1      0.200000       -0.250000      -0.253165      -0.253205      -0.256495      1.949359
 2      0.400000       -0.256495      -0.259915      -0.259961      -0.263523      1.897367
 3      0.600000       -0.263523      -0.267235      -0.267288      -0.271163      1.843909
 4      0.800000       -0.271163      -0.275210      -0.275272      -0.279508      1.788854
 5      1.000000       -0.279508      -0.283945      -0.284017      -0.288675      1.732051
 6      1.200000       -0.288675      -0.293568      -0.293652      -0.298807      1.673320
 7      1.400000       -0.298807      -0.304240      -0.304341      -0.310087      1.612452
 8      1.600000       -0.310087      -0.316167      -0.316289      -0.322749      1.549193
 9      1.800000       -0.322749      -0.329616      -0.329765      -0.337100      1.483240
 10     2.000000       -0.337100      -0.344939      -0.345126      -0.353553      1.414214
>>
```

Pada command window di atas, dilakukan pemanggilan *function* Runge4(f, 0, 2, 2, 10) dimana f merupakan *anonymous function* yang sudah dideklarasikan sebelumnya, 0 adalah titik awal , 2 adalah titik akhir, 2 adalah nilai akhir, dan 10 adalah jumlah iterasi yang dilakukan. Hasil operasi dari *function* Runge4 pada setiap iterasinya akan di-*return* dan disimpan ke dalam variabel [x_runge4, y_runge4] yang nantinya akan digunakan dalam *plotting*.

5. Analisis Grafik



Pada grafik di atas, direpresentasikan hasil perhitungan dari metode eksak dengan warna merah, metode euler dengan warna hijau, metode heun dengan warna kuning, dan metode Runge kutta orde 4 dengan warna biru. Dapat terlihat bahwa semakin besar nilai (x) maka error yang dihasilkan juga akan semakin membesar dan grafik hasil perhitungan dari ketiga metode tersebut terlihat mulai menjauhi grafik dari metode eksak.

6. Analisis Galat

```
Command Window
>> Galat

Hasil Perhitungan dengan 3 Metode :
eksak_8 = (sym) 1.41421356237
euler_8 = (sym) 1.42591958031
heun_8 = (sym) 1.41421753860
runge4_8 = (sym) 1.41421353720

Hasil Perhitungan Galat pada Titik Terakhir:
err_euler = 8.2774e-03
err_heun = 2.8116e-06
err_runge4 = 1.7802e-08

Hasil Perhitungan Galat pada Keseluruhan Iterasi:
all_error_euler = (sym) 1.72691254053
all_error_heun = (sym) 1.72217925168
all_error_runge4 = (sym) 1.72217796983
>>
```

Pada command window di atas, terlihat bahwa pada hasil perhitungan dengan metode Runge Kutta Orde 4 merupakan perhitungan yang nilai akhirnya paling dekat dengan nilai akhir dari hasil dari perhitungan dengan metode Eksak. Hal ini dikarenakan hasil perhitungan dengan metode Runge kutta orde 4 memiliki error relative yang lebih kecil dibanding kedua metode lainnya, yaitu sebesar $1.7802e-8$ atau 0.00000000017802 %. Selain itu, hasil galat pada keseluruhan iterasi juga yang paling rendah, yaitu sebesar 1.72217796983.