TUGAS 2

PERMASALAHAN IVP

KOMPUTASI NUKLIR

Anggota Kelompok:

Bagas Yadher Bima N.A.R.H	18/431318/TK/47911
Imam Bayu Prasetya	18/425231/TK/46926
Muhammad Farhan Ramadhany	18/431325/TK/47918
Muhammad Syafiq Fauzan	18/428979/TK/47481
Valentinus Elzha Widatama	18/425242/TK/46937

PROGRAM STUDI TEKNIK NUKLIR

DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA

FAKULTAS TEKNIK

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2021

A. DESKRIPSI MASALAH

Terdapat dua persoalan yang harus diselesaikan dengan menyusun program/*script* sehingga diperoleh hasil simulasi dengan menggunakan metode Runge-Kutta.

I. Bagian A

Mahasiswa diminta menyelesaikan permasalahan menggunakan metode Runge-Kutta, hasil dari simulasi kemudian membandingkan dengan hasil analitis yang sudah diketahui. Sehingga dapat dilakukan verifikasi program yang disusun sudah berjalan sesuai yang diharapkan.

Diketahui persamaan diferensial

$$y' + 1000(y - (t+2)) - 1 = 0 \text{ , dengan syarat } y(0) = 1$$

Solusi analitis terhadap persamaan diferensial tersebut adalah

$$y(t) = -\,e^{-1000t} + t + 2$$

Persoalan di atas merupakan persamaan diferensial Stiff sehingga penyelesaian dari persamaan tersebut dapat dilakukan dengan menggunakan dua metode, yaitu metode *fixed time step* dimana nilai interval waktu (Δt) tetap dan *adaptive time step* dimana interval waktu berubah-ubah.

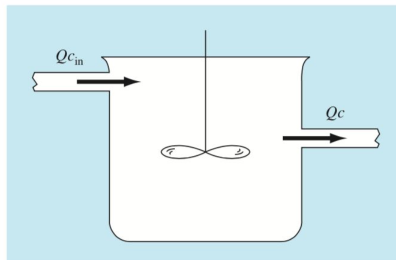
Dari permasalahan tersebut dapat disusun program/*script* menggunakan metode Runge-Kutta orde 4 (RK4) dengan *time step* hingga perhitungan sampai dengan t = 3 detik. Selanjutnya dibuat penyelesaian metode Runge-Kutta-Fehlberg (RKF) dengan *adaptive time step* menggunakan nilai batas toleransi maksimum (Tol_{max}). Nilai *safety* (S), dan nilai ekspansi (N). Nilai tersebut dapat divariasikan untuk mengetahui pengaruhnya terhadap nilai perhitungan.

Hasil dari kedua metode tersebut kemudian dibandingkan dengan hasil analitis. Kemudian akan dibandingkan galat yang dihasilkan antara metode RK4 dengan adaptif. Selain itu perbedaan dalam akurasi dan waktu komputasi antara kedua metode dibandingkan.

II. Bagian B

Pada bagian ini, mahasiswa diminta untuk menerapkan program yang telah dibuat di Bagian A untuk menyelesaikan permasalahan yang lebih realistis. Hasil simulasi tidak perlu dibandingkan dengan solusi analitis.

Gambar berikut ini menunjukkan skema reaktor kimia berpengaduk.



Neraca massa yang untuk reaktor tunggal tersebut dapat dituliskan sebagai berikut

$$V\,dc/dt = Q\,c_{in} - Q\,c$$

dengan Q adalah laju alir, V adalah volume dan c adalah konsentrasi. Jika konsentrasi bagian masukan (inlet) dinyatakan dengan relasi berikut

$$c_{in} = c_b(1 - e^{-0{,}12t})$$

Kemudian menghitung konsentrasi keluaran pada reaktor tersebut sebagai fungsi dari waktu. Diketahui nilai-nilai sebagai berikut: cb = 40 mg/m^3, Q = 6 m^3/menit, V = 100 m^3, dan c(0) = 20 mg/m^3. Perhitungan dimulai pada *timestep* dari t = 0 s.d. 100 menit. Hasil akhir akan dibuat plot konsentrasi *output* dan *input* sebagai fungsi dari waktu.

B. METODE PENYELESAIAN MASALAH

I. Bagian A

A. Runge-Kutta Orde 4

1. Penyusunan Model

Pada persamaan diferensial tersebut akan dilakukan penyelesaian numerik dengan menggunakan metode RK4. Dalam metode ini ditentukan terlebih dahulu kondisi awal (t dan y) dan lebar interval atau banyaknya segmen perhitungan. Kemudian dari variabel-variabel tersebut dilakukan perhitungan nilai koefisien K iteratif yang akan menentukan nilai variabel t dan y setiap titik atau interval tertentu.

2. Algoritma Program

1. Tentukan persamaan analitik dan turunannya,
   a. Y = -exp(-1000*x)+x+2
   b. Y' = 1 - 1000*(y-(t+2))
2. Definisikan waktu awal (w0) = 0, Syarat batas awal (y0) = 1, waktu akhir (t) = 3
3. Tentukan lebar interval (h) tiap segmen,
4. Hitung banyaknya segmen (n) = (t-x0)/h
5. Bentuk matriks variabel terikat dan variabel bebas awal dengan nilai tiap elemennya adalah 0
6. Untuk nilai i = sampai dengan i = banyaknya segmen, lakukan :
   a. Hitung nilai:
      i. k1 = h.f(x(i), y(i))
      ii. k2 = h.f(x(i) + 0,5h, y(i) + 0,5.k1)
      iii. k3 = h.f(x(i) + 0,5h, y(i) + 0,5.k2)
      iv. k4 = h.f(x(i) + h, y(i) + k3)
      v. y(i+1) = y(i) + 1/6 (k1 + 2k2 + 2k3 + k4)
7. Hitung nilai variabel terikat secara analitis (y_a)
   a. y_a = -exp(-1000 * x) + x +2

8. Hitung error relatif dari hasil numerik dan analitis
    a. error absolut = (ya - yrk4)
    b. error relatif = abs((ya - yrk4)./ya)*100
9. Buat plot perbandingan nilai numerik metode RK4 dengan analitis Runge-Kutta-Fehlberg

B. Runge-Kutta Adaptif

1. Penyusunan Model

Pada persamaan diferensial tersebut akan dilakukan penyelesaian numerik dengan menggunakan metode RKF. Dalam metode ini ditentukan terlebih dahulu kondisi awal (t dan y) dan lebar interval atau banyaknya segmen perhitungan. Kemudian dari variabel tersebut dilakukan perhitungan nilai koefisien K yang akan menentukan nilai variabel t dan y setiap titik. Penentuan koefisien tersebut dilakukan evaluasi secara iteratif sehingga *error* mencapai batas toleransi tertentu. Jika nilai *error* melebihi batas toleransi tertentu, maka lebar segmen kembali dihitung dan disesuaikan, kemudian iterasi berlanjut dengan menggunakan nilai lebar segmen yang baru.

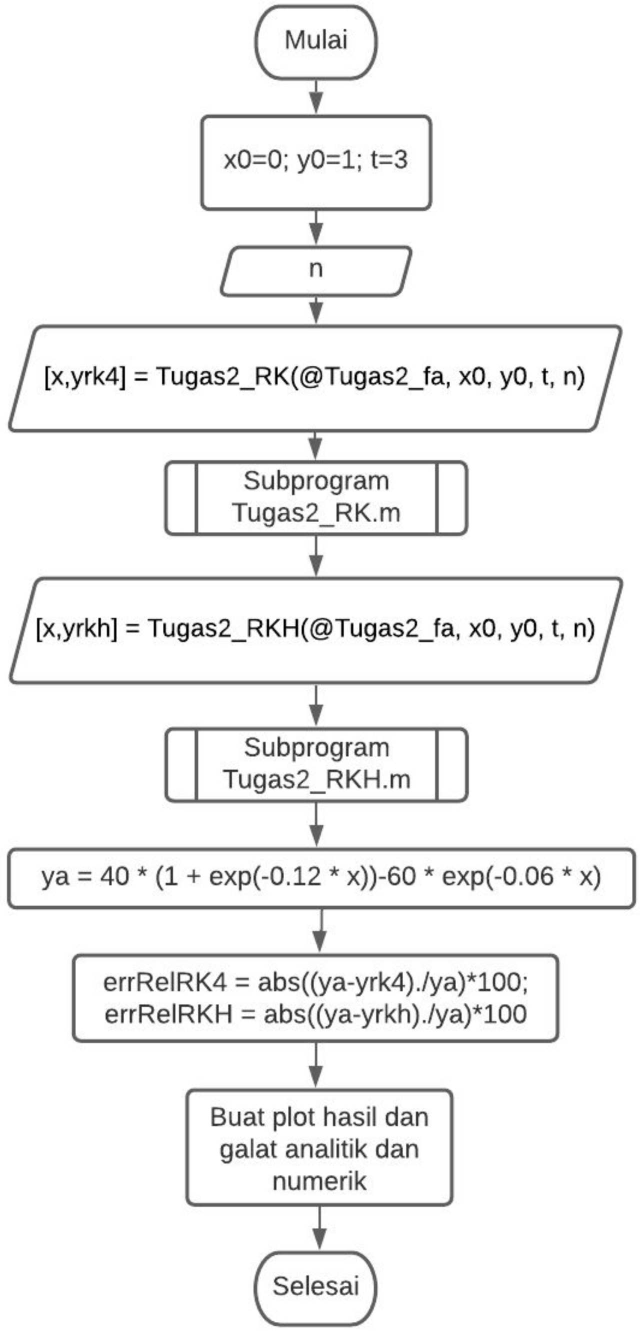
2. Algoritma Program
    1. Tentukan persamaan analitis dan turunannya,
        a. Y = -exp(-1000*x)+x+2
        b. Y' = 1 - 1000*(y-(t+2))
    2. Definisikan waktu awal (X0)=0, Syarat batas awal (y0) = 1, waktu akhir (t) =3
    3. Tentukan jumlah segment (n) metode,
    4. Hitung banyaknya segmen (n) = (t-x0)/h
    5. Bentuk matriks variabel terikat dan variabel bebas awal, dengan nilai tiap elemennya adalah 0
    6. Inputkan nilai toleransi, tol
    7. Inputkan nilai safety, S (0,8 s.d. 0,95)
    8. Inputkan nilai ekspansi, N (2 s.d. 5)
    9. Untuk nilai i = 1 sampai dengan i = banyaknya segmen, lakukan :

a. T(1) = X0
b. Y(1) = Y4(1) = Y5(1) = y0
c. Hitung nilai
   i. k1 = h(i).f(T(i), Y(i));
   ii. k2 = h(i).f(T(i) + h(i)/4, Y(i) + k1/4);
   iii. k3 = h(i).f((T(i) + 3/8. h(i)), (Y(i)+3/32.k1 + 9/32.k2));
   iv. k4 = h(i).f((T(i) + 12/13.h(i)),(Y(i) + 1932/2197.k1 - 7200/2197.k2 + 7296/2197.k3));
   v. k5 = h(i).f((T(i) + h(i)), (Y(i) + 439/216.k1 - 8.k2 + 3680/513.k3 - 845/4104.k4));
   vi. k6 = h(i).f((T(i) + 1/2.h(i)), (Y(i) - 8/27.k1 + 2.k2 - 3544/2565.k3 + 1859/4104.k4 - 11/40.k5));
d. Hitung Y4(i+1), Y5(i+1), T(i+1)
   i. Y4(i+1) = Y(i) + (25/216.k1 + 1408/2565.k3 + 2197/4104.k4 - 1/5.k5);
   ii. Y5(i+1) = Y(i) + 16/135.k1 + 6656/12825.k3 + 28561/56430.k4 - 9/50.k5 + 2/55.k6
   iii. T(i+1) = T(i) + h(i);
e. Hitung nilai error, err = 1/h(i) * abs(Y5(i+1) - Y4(i+1))
f. Jika error > toleransi, maka
   i. Hitung h baru, h(i) = S*h(i)*abs(tol/err)^0.25
   ii. Lakukan break dan kembali ke c dengan nilai h(i) yang baru
g. Alih-alih error <= toleransi, maka
   i. Jika error/toleransi > (N/S)^-5, maka hitung h(i+1) = S*h(i)*abs(tol/err)^0.2
   ii. Alih-alih hitung h(i+1) = N*h(i)

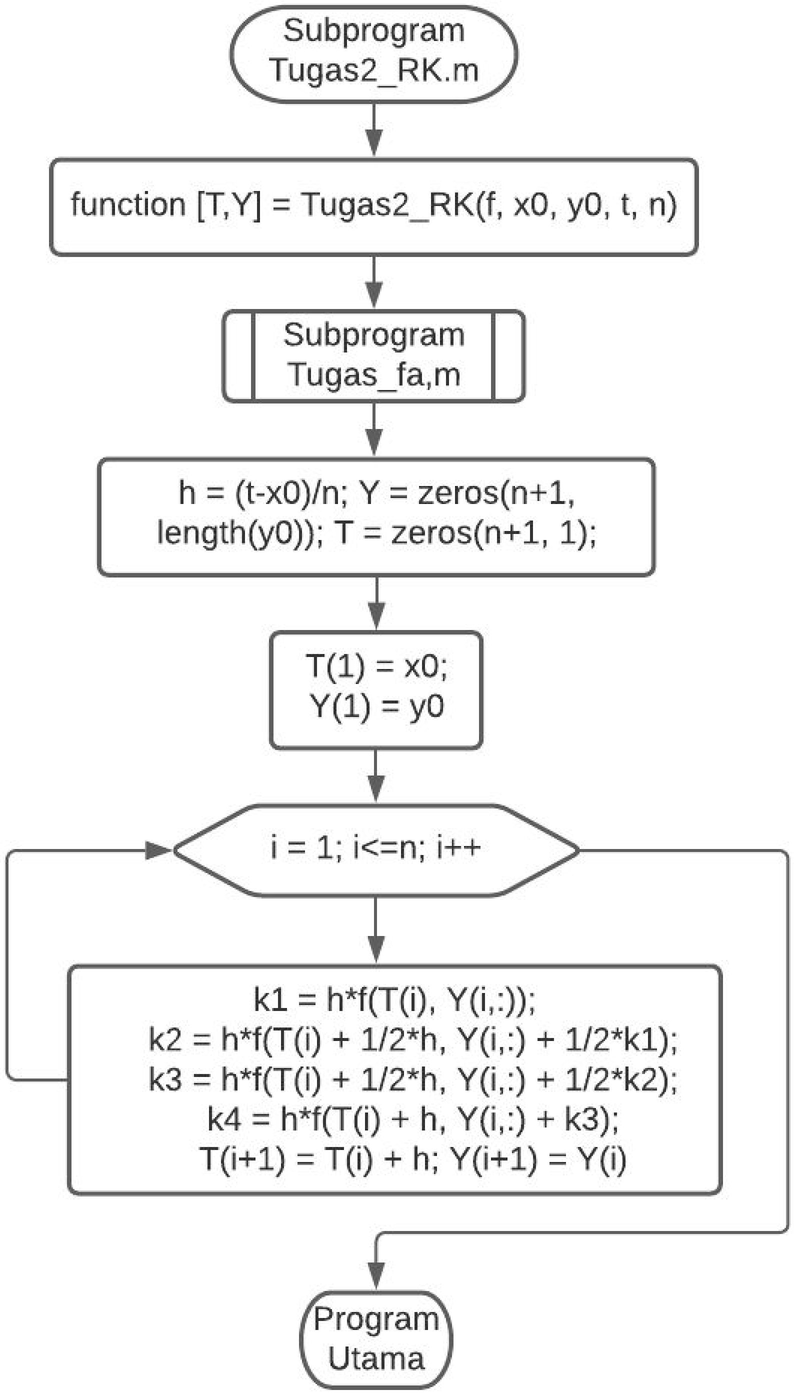
10. Hitung nilai variabel terikat, Ya, dengan menggunakan solusi analitis
11. Hitung *error* relatif dari hasil numerik dan analitis, error = abs((ya-yrkh)./ya)*100
12. Buat plot perbandingan hasil perhitungan dan error pada metode RKF dengan analitis

C. Diagram Alir

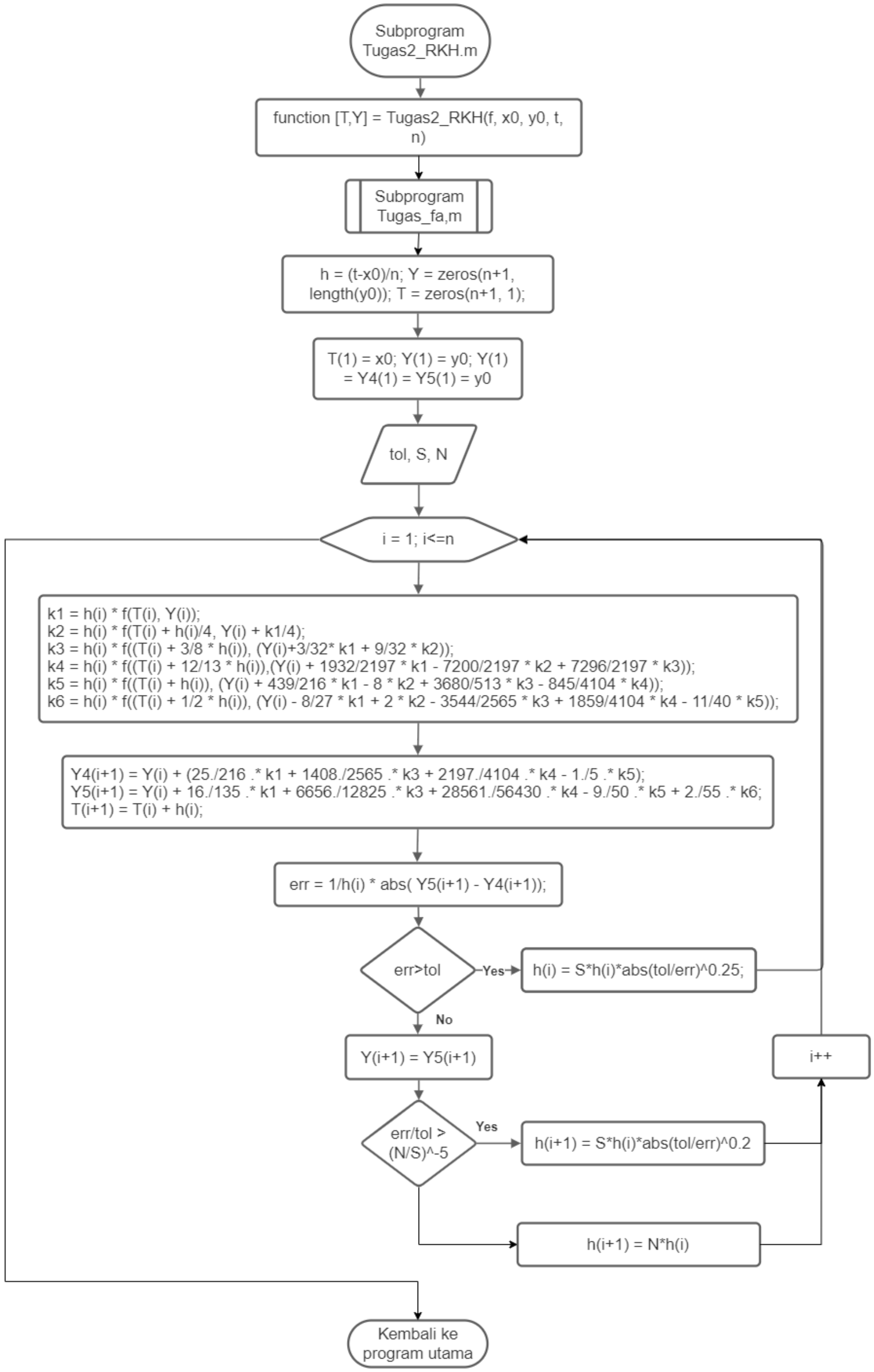
- Program Utama



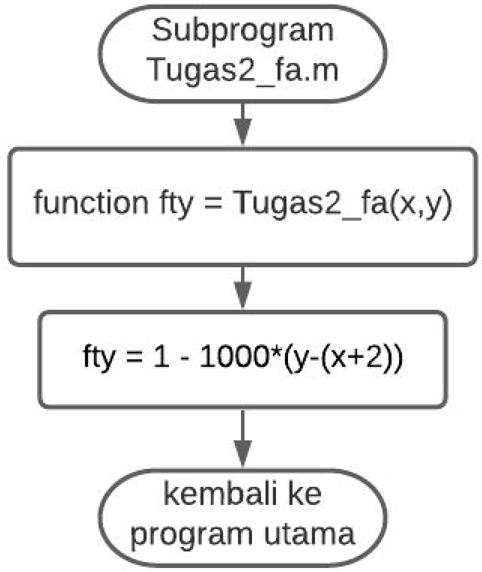
○ Subrogram Runge-Kutta Orde 4



○ Subprogram Runge-Kutta Adaptif



- Subprogram Fungsi



II. Bagian B

A. Persamaan

Persamaan neraca massa yang diberikan adalah sebagai berikut :

$$V \frac{dc}{dt} = Qc_{in} - Qc$$

dengan persamaan konsentrasi masukan yaitu

$$c_{in} = c_b (1 - e^{-0,12t})$$

Diketahui nilai $cb = 40\, mg/cm^3$, $Q = 6\, m^3/menit$, $V = 100\, m^3$, *dan* $c(0) = 20\, mg/m^3$. Persamaan diferensial untuk neraca massa dan konsentrasi masukan menjadi seperti berikut:

$$100c' + 6c - 240 (1 - e^{-0,12t}) = 0 \qquad \text{untuk } 0 \le t \le 100,\ c(0) = 20$$

Dari persamaan tersebut akan didapatkan persamaan untuk solusi analitik yaitu sebagai berikut :

$$c(t) = 40 (1 + e^{-0,12t}) - 60\, e^{-0,06t}$$

B. Algoritma Program

Algoritma bagian B adalah mirip seperti bagian A. Perbedaannya terletak pada persamaan analitis, persamaan diferensial, dan kondisi awal. Algoritma bagian B dapat dibentuk menjadi seperti berikut :

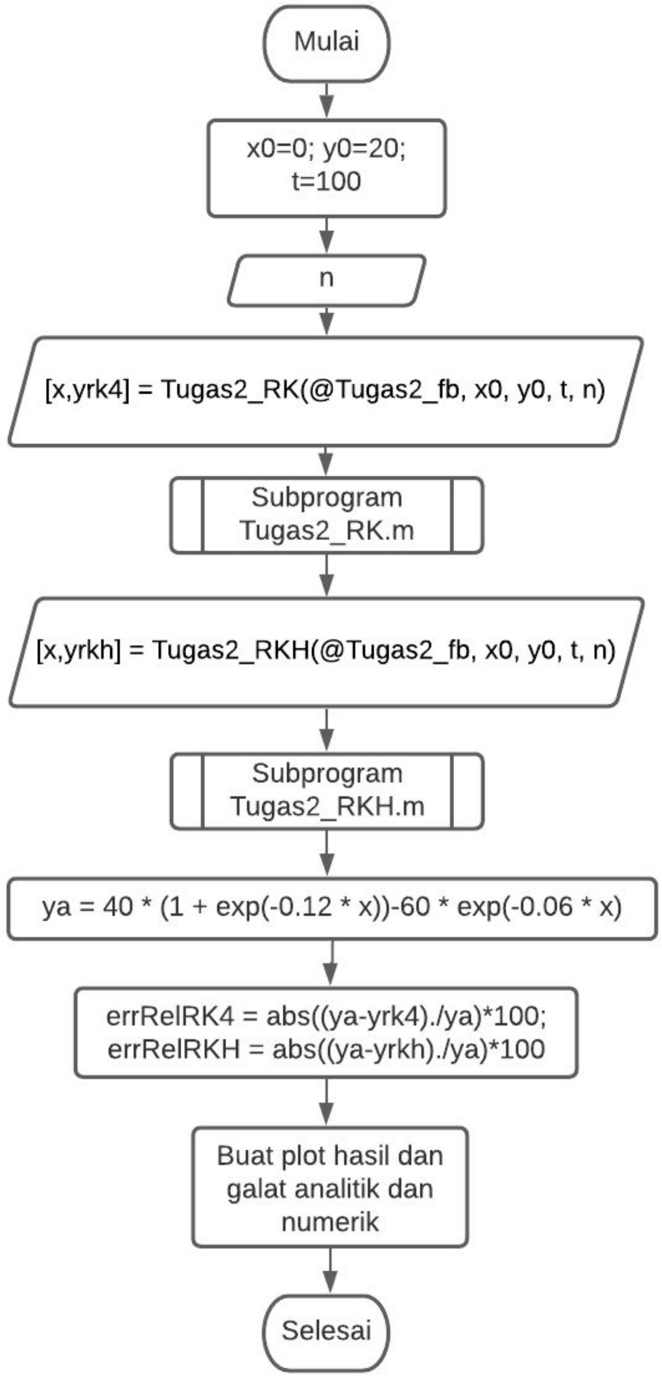
1. Tentukan persamaan analitik dan turunannya,
   a. $c(t) = 40\,(1 + e^{-0{,}1002t}) - 60e^{-0{,}06t}$
   b. $c'(t) = -0{,}06\,c(t) + 2{,}4\,(1 - e^{-0.12t})$
2. Definisikan waktu awal (x0) = 0, Syarat batas awal (y0) = 20, waktu akhir (t) = 100
3. Tentukan lebar interval (h) tiap segmen,
4. Hitung banyaknya segmen (n) = (t-x0)/h
5. Bentuk matriks variabel terikat dan variabel bebas awal, dengan nilai tiap elemennya adalah 0
6. Inputkan nilai toleransi, tol
7. Inputkan nilai *safety*, S
8. Inputkan nilai ekspansi, N
9. Untuk subprogram Runge-Kutta orde 4 (RK4), untuk nilai i = sampai dengan i = banyaknya segmen, lakukan :
   a. Hitung nilai k1, k2, k3, k4, variabel terikat dan variabel bebas
      i. k1 = h.f(x(i), y(i))
      ii. k2 = h.f(x(i) + 0,5h, y(i) + 0,5.k1)
      iii. k3 = h.f(x(i) + 0,5h, y(i) + 0,5.k2)
      iv. k4 = h.f(x(i) + h, y(i) + k3)
      v. y(i+1) = y(i) + 1/6 (k1 + 2k2 + 2k3 + k4)
      vi. T(i+1) = T(i) + h
10. Untuk subprogram Runge-Kutta Fehlberg (RKF), untuk nilai i = 1 sampai dengan i = banyaknya segmen, lakukan :
    a. T(1) = X0
    b. Y(1) = Y4(1) = Y5(1) = y0
    c. Hitung nilai k1, k2, k3, k4, k5, dan k6

        i. k1 = h(i).f(T(i), Y(i));
        ii. k2 = h(i).f(T(i) + h(i)/4, Y(i) + k1/4);
        iii. k3 = h(i).f((T(i) + 3/8. h(i)), (Y(i)+3/32.k1 + 9/32.k2));
        iv. k4 = h(i)f((T(i) + 12/13.h(i)),(Y(i) + 1932/2197.k1 - 7200/2197.k2 + 7296/2197.k3));
        v. k5 = h(i).f((T(i) + h(i)), (Y(i) + 439/216.k1 - 8.k2 + 3680/513.k3 - 845/4104.k4));
        vi. k6 = h(i).f((T(i) + 1/2.h(i)), (Y(i) - 8/27.k1 + 2.k2 - 3544/2565.k3 + 1859/4104.k4 - 11/40.k5));
    d. Hitung Y4(i+1), Y5(i+1), T(i+1)
        i. Y4(i+1) = Y(i) + (25/216.k1 + 1408/2565.k3 + 2197/4104.k4 - 1/5.k5);
        ii. Y5(i+1) = Y(i) + 16/135.k1 + 6656/12825.k3 + 28561/56430.k4 - 9/50.k5 + 2/55.k6
        iii. T(i+1) = T(i) + h(i);
    e. Hitung nilai error, err = 1/h(i) * abs(Y5(i+1) - Y4(i+1))
    f. Jika error > toleransi, maka
        i. Hitung h baru, h(i) = S*h(i)*abs(tol/err)^0.25
        ii. Lakukan iterasi ulang dengan kembali ke c dengan nilai h(i) yang baru
    g. Alih-alih error <= toleransi, maka
        i. Jika error/toleransi > (N/S)^-5, maka hitung h(i+1) = S*h(i)*abs(tol/err)^0.2
        ii. Alih-alih hitung h(i+1) = N*h(i)

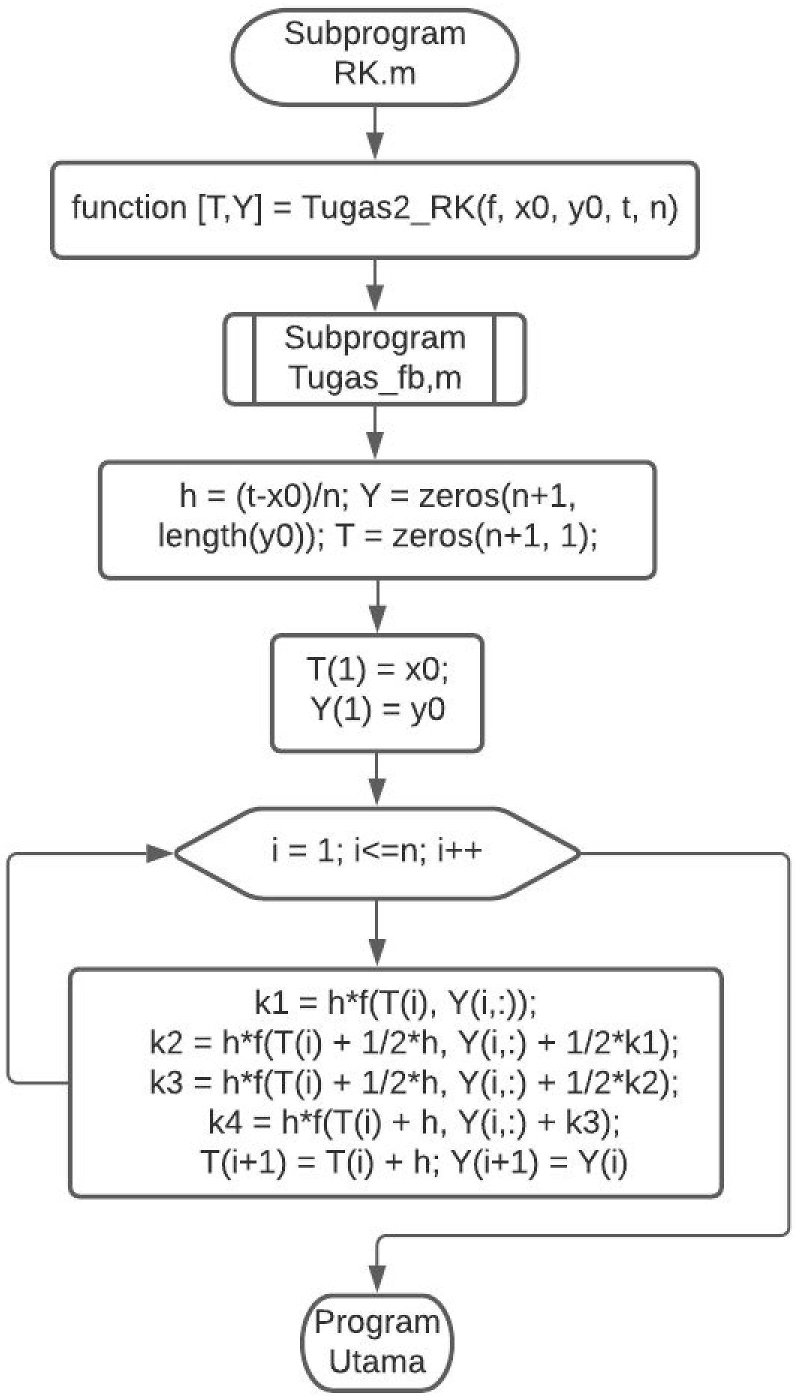
11. Hitung nilai variabel terikat secara analitis
12. Hitung error relatif dari metode RK4, RKF, dan analitis dengan rumus error_1 = abs(ya-yrkh)./ya*100, error_2 = abs(ya-yrk4)./ya*100
13. Buat plot untuk membandingkan nilai antara metode RK4, RKF, dan analitis.

C. Diagram Alir

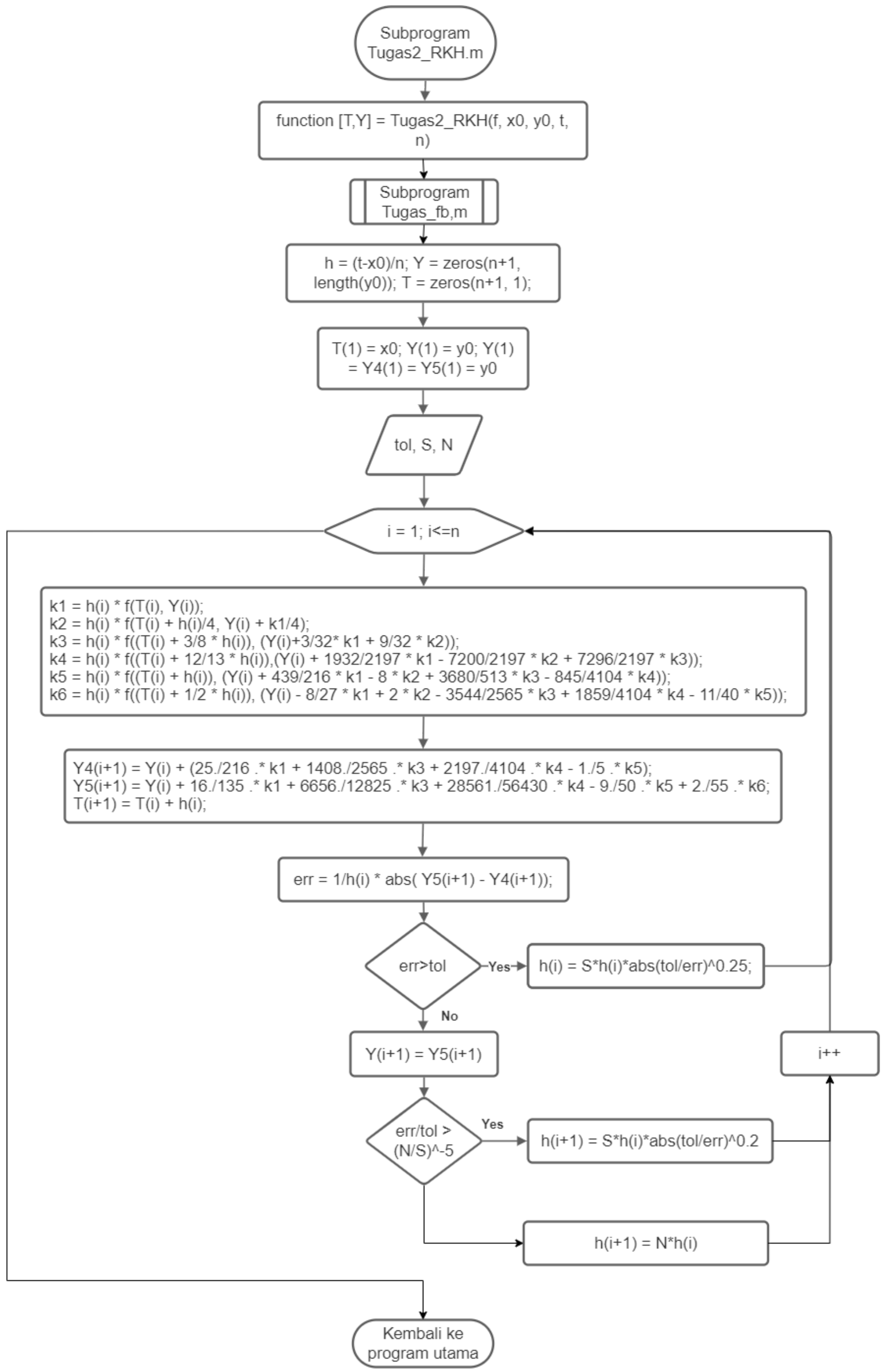
- Program Utama



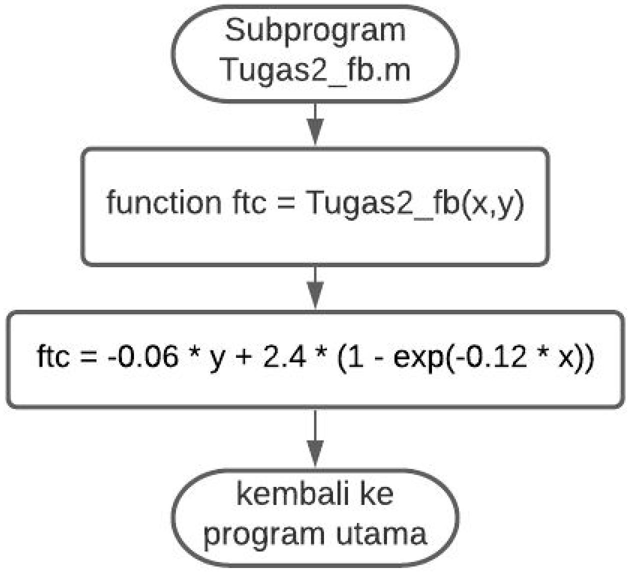
■ Subprogram Runge-Kutta Orde 4



■ Subprogram Runge-Kutta Adaptive



■ Subprogram Fungsi



C. IMPLEMENTASI PROGRAM

I. Bagian A

● Problem Utama Bagian A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Utama;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;clc;clf;

x0 = 0.0;       %Waktu awal / Batas awal
y0 = 1.0;        %Syarat batas awal
n  = input('Masukan jumlah segment (n) : ') %Lebar interval
t  = 3;          %Waktu akhir

tic;
[x,yrk4] = Tugas2_RK(@Tugas2_fa, x0, y0, t, n); %Masuk ke
subprogram RK.m
toc;
tic;
[hx, x, yrkh] = Tugas2_RKH(@Tugas2_fa, x0, y0, t, n); %Masuk ke
subprogram RKH.m
toc;

ya = -exp(-1000 * x) + x +2; %Menghitung solusi analitik
```

```
%Menentukan error relative tiap metode
errSejRK4 = (ya - yrk4);
errSejRKH = (ya - yrkh);
errRelRK4 = abs((ya-yrk4)./ya)*100;
errRelRKH = abs((ya-yrkh)./ya)*100;

%plot perbandingan analitik dan numerik
subplot(5,1,1);
hold on;
grid on;
plot(x,yrk4,'r--',x,yrkh,'b--',x,ya,'k')
hold off;
legend('RK Orde 4','RK-Felhberg','Analitik');
xlabel('Waktu (s)');
ylabel('Y(t)');
xlim([x0 0.05]);
ylim([0 3]);
title('Perbandingan Metode');

%plot error relatif
subplot(5,1,2);
hold on;
grid on;
plot(x,errRelRK4,'r', x, errRelRKH,'b');
hold off;
legend('ErrRel RK4','ErrRel RKF');
xlabel('Waktu (s)');
ylabel('%');
xlim([x0 0.05]);
ylim([0 100]);
title('Galat Relatif');

%plot error sejati RK4
subplot(5,1,3);
hold on;
grid on;
plot(x,errSejRK4,'r');
hold off;
legend('ErrSej RK4');
xlabel('Waktu (s)');
ylabel('-');
xlim([x0 0.05]);
title('Galat Sejati RK4');

%plot error sejati RKF
subplot(5,1,4);
```

```
hold on;
grid on;
plot(x,errSejRKH,'b');
hold off;
legend('ErrSej RKF');
xlabel('Waktu (s)');
ylabel('-');
xlim([x0 0.05]);
title('Galat Sejati RKF');

%plot timestep
subplot(5,1,5);
hold on;
grid on;
plot(x,hx,'g');
hold off;
xlabel('Waktu (s)');
ylabel('h(i)');
xlim([x0 0.05]);
%ylim([1e-3 10e-3]);
title('Timestep RKF');
```

- Persamaan Diferensial Bagian A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Fungsi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fty = fa(t,y)

fty = 1 - 1000*(y-(t+2)); %turunan pertama fungsi

end
```

- Program Runge Kutta Orde 4 Bagian A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Orde 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T,Y] = Tugas2_RK(f, x0, y0, t, n)

%fungsi Runge Kutta orde 4
%penyelesaian numerik persamaan diferensial dengan metode Runge Kutta
orde 4

h = (t-x0)/n;       %Interval segmen
Y = zeros(n+1, 1);  % variabel terikat awal
T = zeros(n+1, 1);  % variabel waktu awal
```

```
%syarat awal
T(1) = x0;
Y(1) = y0;

%jalankan sebanyak n langkah
for i = 1:n
    k1 = h*f(T(i), Y(i));
    k2 = h*f(T(i) + 1/2*h, Y(i) + 1/2*k1);
    k3 = h*f(T(i) + 1/2*h, Y(i) + 1/2*k2);
    k4 = h*f(T(i) + h, Y(i) + k3);
    T(i+1) = T(i) + h; %Menentukan T pada segmen berikutnya
    Y(i+1) = Y(i) + 1/6*(k1 + 2*k2 + 2*k3 + k4); %Menentukan Y pada
segmen berikutnya

end
end
```

- Program Runge Kutta Fehlberg Adaptive Bagian A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Fehlberg
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ht, T,Y] = Tugas2_RKH(f, x0, y0, t, n)

% Fungsi Runge-Kutta penyelesaian numerik persamaan diferensial dengan
metode
% Runge-Kutta Adaptive

h = (t-x0)/n;       %Menentukan banyak segment
ht(1, :) = h;       %Perubahan adaptif interval tiap segmen
Y = zeros(n+1, 1);  %Variabel terikat awal
T = zeros(n+1, 1);  %Variabel bebas awal

% syarat awal
T(1) = x0;
Y(1) = Y4(1) = Y5(1) = y0;

%Input parameter adaptive
tol  = input('Inputkan Batas Toleransi (T): ');
S    = input('Inputkan Nilai Safety (S)   : ' );
N    = input('Inputkan Nilai Ekspansi (N) : ' );

%Parameter untuk iterasi adaptif
i = 1;

% jalankan sebanyak n langkah
while i<=n
    k1 = h(i) * f(T(i), Y(i));
    k2 = h(i) * f(T(i) + h(i)/4, Y(i) + k1/4);
    k3 = h(i) * f((T(i) + 3/8 * h(i)), (Y(i)+3/32* k1 + 9/32 * k2));
```

```
    k4 = h(i) * f((T(i) + 12/13 * h(i)),(Y(i) + 1932/2197 * k1 -
7200/2197 * k2 + 7296/2197 * k3));
    k5 = h(i) * f((T(i) + h(i)), (Y(i) + 439/216 * k1 - 8 * k2 +
3680/513 * k3 - 845/4104 * k4));
    k6 = h(i) * f((T(i) + 1/2 * h(i)), (Y(i) - 8/27 * k1 + 2 * k2 -
3544/2565 * k3 + 1859/4104 * k4 - 11/40 * k5));

    %Menghitung variabel terikat dan variabel bebas pada kondisi
berikutnya
    %Y orde 4
    Y4(i+1) = Y(i) + (25./216 .* k1 + 1408./2565 .* k3 + 2197./4104 .*
k4 - 1./5 .* k5);

    %Y orde 5
    Y5(i+1) = Y(i) + 16./135 .* k1 + 6656./12825 .* k3 + 28561./56430
.* k4 - 9./50 .* k5 + 2./55 .* k6;
    T(i+1) = T(i) + h(i);

    %Menghitung perkiraan error untuk menentukan lebar interval
    err = 1/h(i) * abs( Y5(i+1) - Y4(i+1));

    if (err <= tol) %Diterima, dan gunakan interval berikut untuk
langkah berikutnya
        Y(i+1) = Y5(i+1);
          if (err/tol > (N/S)^-5)
            h(i+1) = S*h(i)*abs(tol/err)^0.2;
        else
            h(i+1) = N*h(i);
        endif
        i = i + 1;
    else
            h(i) = S*h(i)*abs(tol/err)^0.25; %Ditolak, dan gunakan
interval berikut untuk menghitung solusi sekarang
      endif
endwhile
end
```

II. Bagian B

- Program Utama Bagian B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Utama;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;clc;clf;

x0 = 0.0;       %Waktu awal / Batas awal
y0 = 20.0;        %Syarat batas awal
n  = input('Masukan jumlah segment (n) : ') %Lebar interval
```

```
t  = 100;          %Waktu akhir

tic;
[x,yrk4] = Tugas2_RK(@Tugas2_fb, x0, y0, t, n); %Masuk ke subprogram
RK.m
toc;
tic;
[hx, x,yrkh] = Tugas2_RKH(@Tugas2_fb, x0, y0, t, n); %Masuk ke
subprogram RKH.m
toc;

ya = 40 * (1 + exp(-0.12 * x))-60 * exp(-0.06 * x); %Menghitung solusi
analitik

%Menentukan error relative tiap metode
errSejRK4 = (ya - yrk4);
errSejRKH = (ya - yrkh);
errRelRK4 = abs((ya-yrk4)./ya)*100;
errRelRKH = abs((ya-yrkh)./ya)*100;

%plot perbandingan analitik dan numerik
subplot(5,1,1);
hold on;
grid on;
plot(x,yrk4,'r--',x,yrkh,'b--',x,ya,'k')
hold off;
legend('RK Orde 4','RK-Felhberg','Analitik');
xlabel('Waktu (s)');
ylabel('Y(t)');
xlim([x0 t]);
ylim([10 max(ya)]);
title('Perbandingan Metode');

%plot error relatif
subplot(5,1,2);
hold on;
grid on;
plot(x,errRelRK4,'r',x,errRelRKH,'b');
hold off;
legend('ErrRel RK4','ErrRel RKF');
xlabel('Waktu (s)');
ylabel('%');
xlim([x0 t]);
ylim([0 100]);
title('Galat Relatif');

%plot error sejati RK4
subplot(5,1,3);
hold on;
grid on;
plot(x,errSejRK4,'r');
```

```
hold off;
legend('ErrSej RK4');
xlabel('Waktu (s)');
ylabel('-');
xlim([x0 t]);
title('Galat Sejati RK4');

%plot error sejati RKF
subplot(5,1,4);
hold on;
grid on;
plot(x,errSejRKH,'b');
hold off;
legend('ErrSej RKF');
xlabel('Waktu (s)');
ylabel('-');
xlim([x0 t]);
title('Galat Sejati RKF');

%plot timestep
subplot(5,1,5);
hold on;
grid on;
plot(x,hx,'g');
hold off;
xlabel('Waktu (s)');
ylabel('h(i)');
xlim([x0 t]);
%ylim([1e-3 10e-3]);
title('Timestep RKF');
```

- Persamaan Diferensial Bagian B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Fungsi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ftc = fb(x,y)

ftc = -0.06 * y + 2.4 * (1 - exp(-0.12 * x));

end
```

- Program Runge Kutta Orde 4 Bagian B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Orde 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T,Y] = Tugas2_RK(f, x0, y0, t, n)

%fungsi Runge Kutta orde 4
%penyelesaian numerik persamaan diferensial dengan metode Runge Kutta
orde 4

h = (t-x0)/n;       %Interval segmen
Y = zeros(n+1, 1);  % variabel terikat awal
T = zeros(n+1, 1);  % variabel waktu awal

%syarat awal
T(1) = x0;
Y(1) = y0;

%jalankan sebanyak n langkah
for i = 1:n
    k1 = h*f(T(i), Y(i));
    k2 = h*f(T(i) + 1/2*h, Y(i) + 1/2*k1);
    k3 = h*f(T(i) + 1/2*h, Y(i) + 1/2*k2);
    k4 = h*f(T(i) + h, Y(i) + k3);
    T(i+1) = T(i) + h; %Menentukan T pada segmen berikutnya
    Y(i+1) = Y(i) + 1/6*(k1 + 2*k2 + 2*k3 + k4); %Menentukan Y pada
segmen berikutnya

end
end
```

- Program Runge Kutta Fehlberg Adaptive Bagian B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program Runge Kutta Fehlberg
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ht, T,Y] = Tugas2_RKH(f, x0, y0, t, n)

% Fungsi Runge-Kutta penyelesaian numerik persamaan diferensial dengan
metode
% Runge-Kutta Adaptive

h = (t-x0)/n;       %Menentukan banyak segment
ht(1, :) = h;       %Perubahan adaptif interval tiap segmen
Y = zeros(n+1, 1);  %Variabel terikat awal
T = zeros(n+1, 1);  %Variabel bebas awal

% syarat awal
T(1) = x0;
Y(1) = Y4(1) = Y5(1) = y0;
```

```
%Input parameter adaptive
tol  = input('Inputkan Batas Toleransi (T): ');
S    = input('Inputkan Nilai Safety (S)   : ' );
N    = input('Inputkan Nilai Ekspansi (N) : ' );

%Parameter untuk iterasi adaptif
i = 1;

% jalankan sebanyak n langkah
while i<=n
    k1 = h(i) * f(T(i), Y(i));
    k2 = h(i) * f(T(i) + h(i)/4, Y(i) + k1/4);
    k3 = h(i) * f((T(i) + 3/8 * h(i)), (Y(i)+3/32* k1 + 9/32 * k2));
    k4 = h(i) * f((T(i) + 12/13 * h(i)),(Y(i) + 1932/2197 * k1 -
7200/2197 * k2 + 7296/2197 * k3));
    k5 = h(i) * f((T(i) + h(i)), (Y(i) + 439/216 * k1 - 8 * k2 +
3680/513 * k3 - 845/4104 * k4));
    k6 = h(i) * f((T(i) + 1/2 * h(i)), (Y(i) - 8/27 * k1 + 2 * k2 -
3544/2565 * k3 + 1859/4104 * k4 - 11/40 * k5));

    %Menghitung variabel terikat dan variabel bebas pada kondisi
berikutnya
    %Y orde 4
    Y4(i+1) = Y(i) + (25./216 .* k1 + 1408./2565 .* k3 + 2197./4104 .*
k4 - 1./5 .* k5);

    %Y orde 5
    Y5(i+1) = Y(i) + 16./135 .* k1 + 6656./12825 .* k3 + 28561./56430
.* k4 - 9./50 .* k5 + 2./55 .* k6;
    T(i+1) = T(i) + h(i);

    %Menghitung perkiraan error untuk menentukan lebar interval
    err = 1/h(i) * abs( Y5(i+1) - Y4(i+1));

    if (err <= tol) %Diterima, dan gunakan interval berikut untuk
langkah berikutnya
        Y(i+1) = Y5(i+1);
          if (err/tol > (N/S)^-5)
            h(i+1) = S*h(i)*abs(tol/err)^0.2;
        else
            h(i+1) = N*h(i);
        endif
        i = i + 1;
    else
            h(i) = S*h(i)*abs(tol/err)^0.25; %Ditolak, dan gunakan
interval berikut untuk menghitung solusi sekarang
      endif
endwhile
end
```

D. HASIL DAN PEMBAHASAN

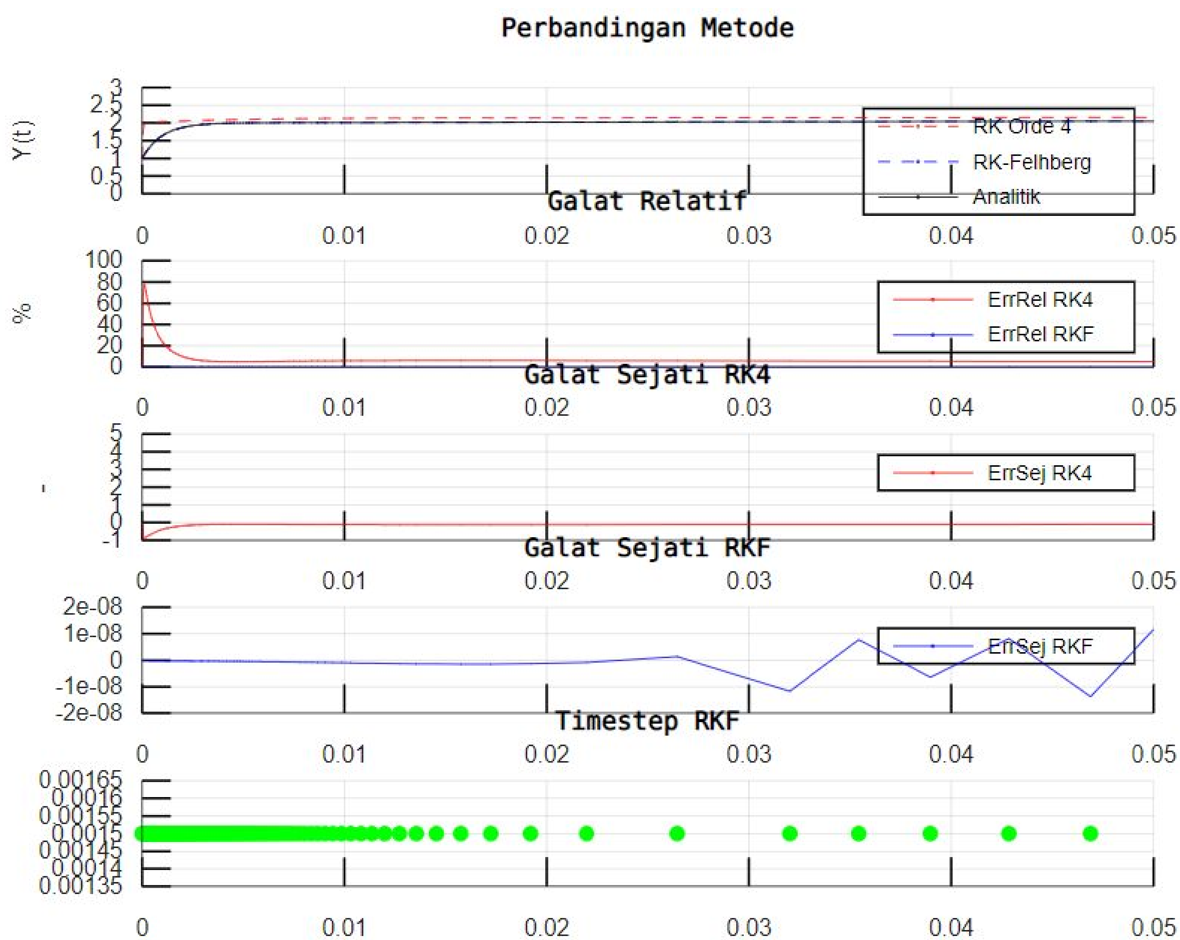
I. Bagian A

Pada persoalan yang diberikan, metode penyelesaian menggunakan metode Runge-Kutta. Metode ini dapat memperoleh akurasi dari deret ekspansi Taylor tanpa memerlukan diferensiasi orde yang lebih tinggi. Penggunaan metode ini ditentukan terlebih dahulu titik awal integrasi x0 dan y0 serta menentukan jumlah iterasi dengan interval tertentu.

Input program yang diberikan adalah sebagai berikut:

```
Masukan jumlah segment (n) : > 2000
n = 2000
Elapsed time is 0.253191 seconds.
Inputkan Batas Toleransi (T): > 1e-5
Inputkan Nilai Safety (S)    : > 0.89
Inputkan Nilai Ekspansi (N) : > 5
Elapsed time is 6.81848 seconds.
```

Kurva hasil simulasi dapat dilihat pada gambar berikut :



Simulasi dilakukan dengan menentukan nilai interval atau *time step* h pada awal *running program*. Untuk memperoleh hasil grafik dengan memiliki galat yang rendah maka nilai interval harus memenuhi syarat stabil dari metode RK4 yaitu $h < 2,785/\alpha$. Permasalahan soal memiliki persamaan diferensial dengan nilai $\alpha = 1000$ yang diperoleh dari persamaan $y' + 1000(y - (t+2)) - 1 = 0$, sehingga nilai interval $h < 0,002785$. Sedangkan perhitungan persamaan diferensial dengan metode RKF dengan *adaptive time step* perlu ditentukan nilai *error* dan nilai batas toleransi (Tol_{max}) untuk mengatur ukuran langkah. Selain itu ditentukan nilai *safety* (S) ($0,8 \leq S \leq 0,95$) dan nilai ekspansi (N) ($2 \leq S \leq 5$) untuk mengevaluasi interval sehingga nilai *time step* akan berubah/ beradaptasi sesuai dengan keadaan perhitungan hingga mencapai stabilitas. Dalam perhitungan RKF juga perlu ditentukan nilai interval (h) dengan memenuhi syarat stabil sebesar kurang dari 0,002785. Selanjutnya dilakukan perhitungan untuk menentukan perkiraan *error* (en) yang kemudian dibandingkan dengan nilai Tol_{max}. Jika nilai en <= Tol_{max}, nilai yn+1 diterima sebagai solusi pada langkah sekarang. Kemudian perhitungan dilanjutkan dengan menentukan ukuran interval atau *timestep* sesuai dengan syarat kondisi. Jika nilai en > Tol_{max}, nilai interval akan dihitung ulang yang kemudian digunakan untuk perhitungan RK4.

Hasil perhitungan dengan metode RK4 dengan *fixed time step* memiliki nilai galat cukup tinggi pada waktu t dengan orde kurang dari 0,01 s. Hal ini dikarenakan persamaan yang digunakan pada bagian ini merupakan persamaan diferensial Stiff. Secara sederhana, persamaan diferensial Stiff merupakan persamaan dengan perubahan nilai yang besar dalam jangka waktu yang sangat pendek sebelum menghasilkan perubahan nilai yang kecil pada waktu lainnya.

Hal tersebut menyebabkan proses komputasi dengan *timestep* konstan menghasilkan nilai yang tidak akurat karena hanya menggunakan satu jenis *timestep* yang terlalu besar atau terlalu kecil pada jangka waktu tertentu. Untuk perhitungan dengan metode RKF dengan *adaptive time step* diperoleh nilai galat cukup rendah dibandingkan nilai hasil analitis. Dari hasil grafik perhitungan nilai perhitungan metode RK4 mengalami kenaikan pada orde t 0,01 lebih besar dari hasil analitis sedangkan hasil perhitungan metode RKF hampir berhimpitan dengan nilai analitis. Dari segi waktu yang diperlukan saat menjalankan, metode Runge-Kutta-Fehlberg membutuhkan waktu yang lebih lama dibanding metode Runge-Kutta orde 4. Hal ini disebabkan adanya evaluasi terhadap nilai *error* terhadap toleransi yang akan mengubah lebar interval *time step* pada segmen sekarang dan berikutnya.

II. Bagian B

Pada bagian ini, permasalahan yang diberikan adalah menghitung konsentrasi keluaran dari suatu reaktor kimia berpengaduk dengan informasi neraca massa, konsentrasi masukan, dan kondisi awal yang telah diketahui.

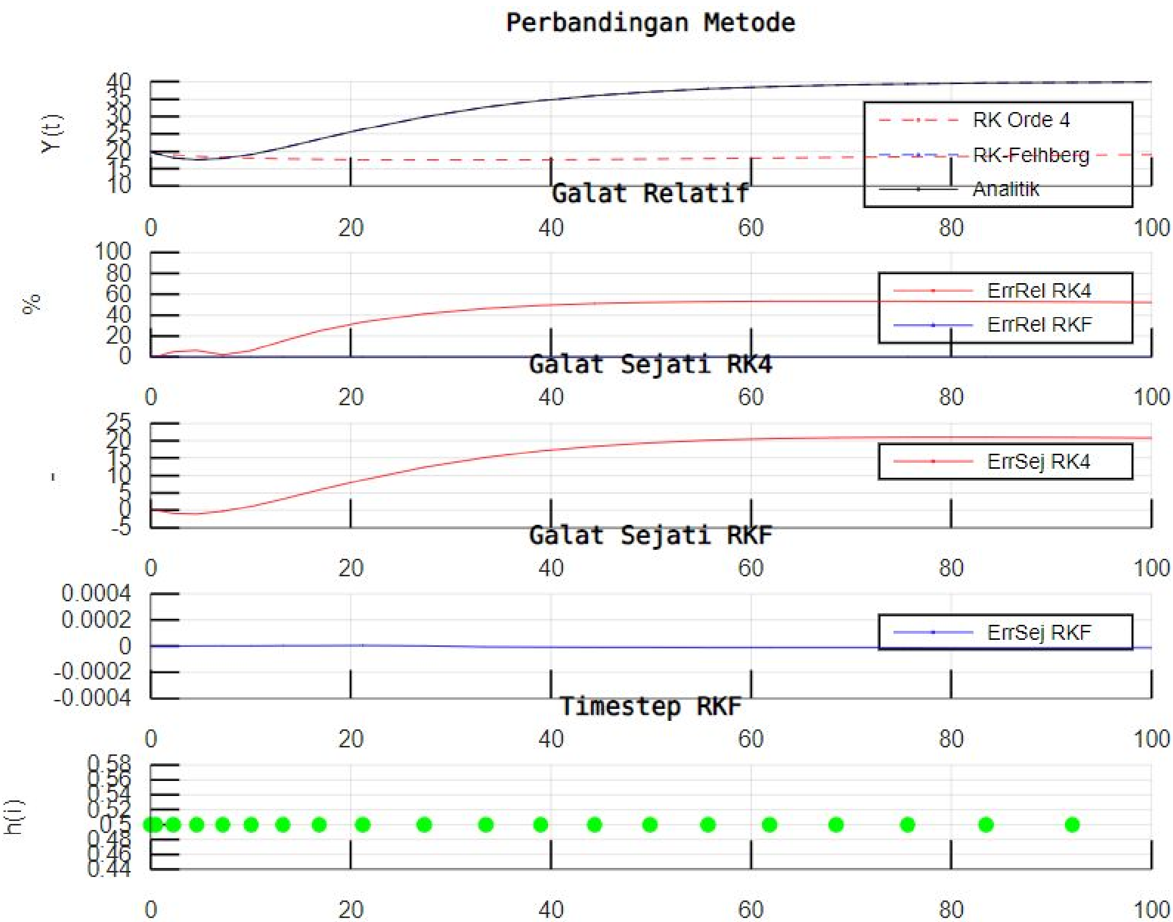
Neraca massa adalah persamaan diferensial ordiner dan konsentrasi masukan merupakan suatu fungsi waktu. Agar permasalahan yang diberikan dapat diselesaikan dengan kode yang telah dibuat untuk bagian A, maka dibutuhkan satu persamaan diferensial ordiner. Maka pertama, kedua persamaan digabung menjadi satu, dan setiap konstanta yang telah diketahui disubstitusikan dengan persamaan tersebut. Dalam persamaan diferensial ini hanya terdapat variabel c beserta turunannya dan variabel t (waktu). Selanjutnya dibutuhkan persamaan analitik untuk perbandingan galat. Persamaan analitik diperoleh melalui metode faktor integral karena persamaan diferensial yang digunakan adalah persamaan diferensial linier [1]**.** Waktu (t) yang diinginkan untuk permasalahan ini adalah 100 menit.

Sama seperti bagian A, simulasi dilakukan dengan menentukan nilai interval atau *time step* h pada awal *running program*. Untuk memperoleh hasil grafik dengan memiliki galat yang rendah maka nilai interval harus memenuhi syarat stabil dari metode RK4 yaitu $h < 2,785/\alpha$. Permasalahan soal memiliki persamaan diferensial dengan nilai $\alpha = 60$ yang diperoleh dari persamaan $100c' + 6c - 240(1 - e^{-0,12t}) = 0$, sehingga diperoleh nilai interval $h < 0,046416$. Sedangkan untuk mensimulasikan metode RKF perlu ditentukan variabel-variabel untuk mengevaluasi nilai interval *time step* sama seperti bagian A.

Input program yang diberikan adalah sebagai berikut :

```
Masukan jumlah segment (n) : > 200
n = 200
Elapsed time is 0.025533 seconds.
Inputkan Batas Toleransi (T): > 1e-5
Inputkan Nilai Safety (S)    : > 0.85
Inputkan Nilai Ekspansi (N) : > 4
Elapsed time is 6.38385 seconds.
```

Hasil yang diperoleh dari iterasi numerik dapat dilihat dalam plot berikut.



Persamaan diferensial yang digunakan adalah persamaan diferensial Stiff dapat dilihat dari perbedaan kurva pada interval awal dan interval sisanya pada persamaan analitis. Persamaan RK4 memiliki galat relatif yang sangat tinggi serta bentuk kurva yang jauh dari persamaan analitis yang diberikan. Sedangkan RKF memiliki bentuk kurva yang sangat mirip dengan persamaan analitik. Hal ini disebabkan karena RKF menggunakan *adaptive segmentation* dimana lebar segmen nilainya berubah-ubah, sedangkan RK4 menggunakan *constant segmentation* dimana lebar segmen nilainya terus sama selama proses iterasi. Dari informasi tersebut, diperoleh bahwa *constant segmentation* tidak dapat digunakan untuk menyelesaikan persamaan diferensial Stiff. Dari segi waktu yang diperlukan saat menjalankan program, metode Runge-Kutta-Fehlberg membutuhkan waktu yang lebih lama dibanding metode Runge-Kutta orde 4. Kondisi ini sama seperti permasalahan pada bagian A yang disebabkan adanya evaluasi terhadap nilai *error* terhadap toleransi yang akan mengubah lebar interval *time step* pada segmen sekarang dan berikutnya.

E. KESIMPULAN DAN SARAN

I. Kesimpulan

Dari program yang telah dibuat, dapat disimpulkan bahwa :

- Metode Runge-Kutta-Fehlberg lebih cocok digunakan dalam fenomena yang prosesnya menghasilkan persamaan diferensial Stiff (berubah secara drastis sebelum berubah dengan stabil),
- *Constant segmentation* tidak dapat digunakan untuk menyelesaikan persamaan diferensial Stiff. Penyelesaian dilakukan dengan menggunakan adaptive segmentation.
- Persamaan diferensial ordiner linier dapat diselesaikan dengan metode numerik/proses iterasi hanya dengan satu buah batas awal serta persamaan diferensial itu sendiri.
- Semakin banyak titik segmen maka proses iterasi membutuhkan waktu yang lebih lama.

II. Saran

Sebaiknya dilakukan pengkajian ulang terhadap kode program untuk melakukan plotting hasil komputasi agar gambar yang didapatkan lebih mudah dibaca dan dipahami.

F. DAFTAR ACUAN

[1] “Differential Equations - Linear Equations.” [Online] Available : https://tutorial.math.lamar.edu/Classes/DE/Linear.aspx (accessed Feb. 27, 2021).