

LAPORAN TUGAS 8
PENGUNAAN SYMBOLIC UNTUK NUMERIC PROCESSING
KOMPUTASI NUKLIR



Disusun Oleh :

Muhammad Farhan Ramadhany (18/431325/TK/47918)

PROGRAM STUDI TEKNIK NUKLIR
DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2021

Daftar Isi

Halaman Judul	i
Daftar Isi	ii
A. Deskripsi Masalah	1
B. Metode Penyelesaian Masalah.....	1
a. Algoritma	1
b. Diagram Alir.....	2
C. Implementasi Program.....	3
D. Hasil	4
a. Perbandingan Solusi Analitik dan Numerik	4
b. Perbandingan Error Absolut	5
E. Penjelasan.....	5

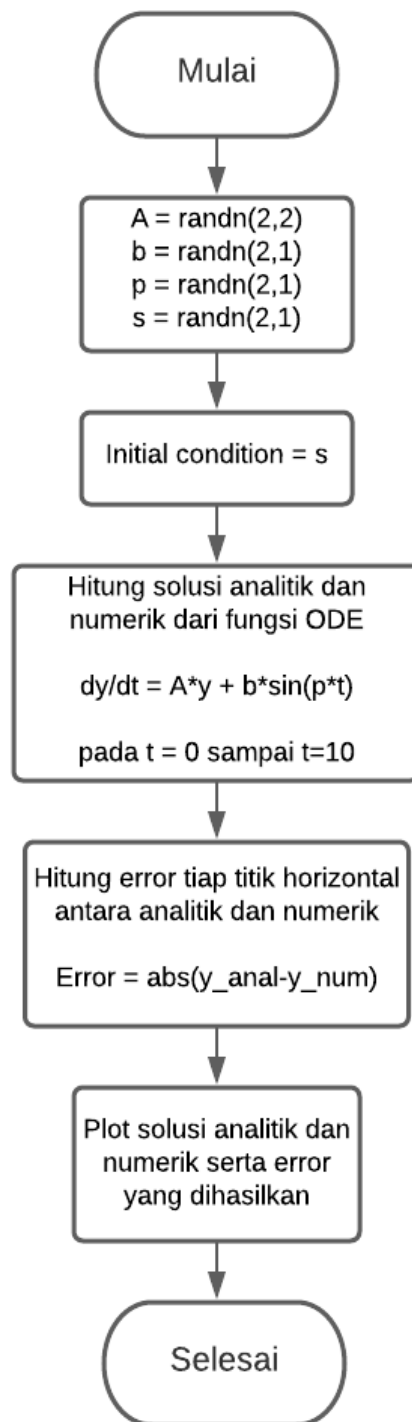
A. Deskripsi Masalah

- a. Buatlah matriks A (2×2) dan vektor kolom b , p , dan s dengan ukuran matriks (2×1) menggunakan fungsi `randn()`.
- b. Hitunglah solusi analitis $y(t) = (y_1(t), y_2(t))^T$, t adalah bilangan riil dari sistem persamaan diferensial $y'(t) = Ay(t) + (b_1 \sin(p_1 t), b_2 \sin(p_2 t))^T$, $y(0) = s$. Gunakan fungsi `diff()` dan `dsolve()`.
- c. Hitunglah solusi numerik $y_{num}(t)$ pada segmen $t=[0,10]$ menggunakan fungsi `ode45()`.
- d. Plot fungsi berikut pada segmen $t=[0,10]$,
 - Solusi analitis $y_1(t), y_2(t)$
 - Galat absolut $\Delta_{abs}(t) = |y_{num}(t) - y(t)|$

B. Metode Penyelesaian Masalah

- a. Algoritma
 1. Susun matriks A ukuran 2×2 dan matriks b , p , s dengan ukuran 2×1 dengan nilai tiap elemen berupa bilangan random.
 2. Definisikan hasil solusi analitis berupa matriks 2×2 .
 3. Hitung solusi analitis dengan menggunakan fungsi `dsolve()` dengan *initial condition* adalah matriks s .
 4. Ambil solusi analitis dengan 5 angka penting.
 5. Definisikan rentang horizontal dari 0 sampai 10.
 6. Hitung solusi numerik dengan masuk ke sub-fungsi ODE dengan menggunakan fungsi `ode45()`.
 7. Hitung error absolut yang dihasilkan dari hasil analitik dan numerik pada tiap titik.
 8. Plot hasil analitik, numerik, dan error yang dihasilkan.

b. Diagram Alir



C. Implementasi Program

```
clear all;clf;clc;

%soal a
%Bangkitkan bilangan random dengan fungsi randn()
A = randn(2,2);
b = randn(2,1);
p = randn(2,1);
s = randn(2,1);

%soal b
syms y1(t) y2(t)

%Definisi ulang ODE
ODE = [diff(y1) ==
A(1,1)*y1(t)+A(1,2)*y2(t)+(b(1)*sin(p(1)*t)),diff(y2) ==
A(2,1)*y1(t)+A(2,2)*y2(t)+(b(2)*sin(p(2)*t))];

%Definisi ulang IC
ic = [y1(0)==s(1),y2(0)==s(2)];

%Komputasi solusi analitik
[y_analitik1(t),y_analitik2(t)] = dsolve(ODE, ic)

%soal c
%Rentang sumbu horizontal dari 0 sampai 10
tspan = [0 10];

%Definisi ulang kondisi awal
y0 = [s(1);s(2)]

%Komputasi solusi numerik
[t,y_numerik] = ode45(@(t,y_numerik) pers(t,y_numerik,A,b,p), tspan,
y0);

%soal d
%Definisi error tiap segmen horizontal
error = zeros(size(t,1),2)

%Perhitungan error tiap segmen
for i=1:1:size(t,1)
    error(i,1) = norm(y_numerik(i,1) - y_analitik1(t(i)),2)
    error(i,2) = norm(y_numerik(i,2) - y_analitik2(t(i)),2)
end

%Plot solusi analitik dan numerik dalam 1 grafik
figure(1)
fplot(y_analitik1,'r')
hold on
fplot(y_analitik2,'b')
hold on
plot(t,y_numerik(:,1),'--r',t,y_numerik(:,2),'--b')
hold off
ylim([-10 10]);
xlim([0 10]);
legend('yAnalitis1','yAnalitis2','yNumerik1','yNumerik2')
title('Solusi Analitis dan Numerik');
xlabel('t')
ylabel('y')
```

```

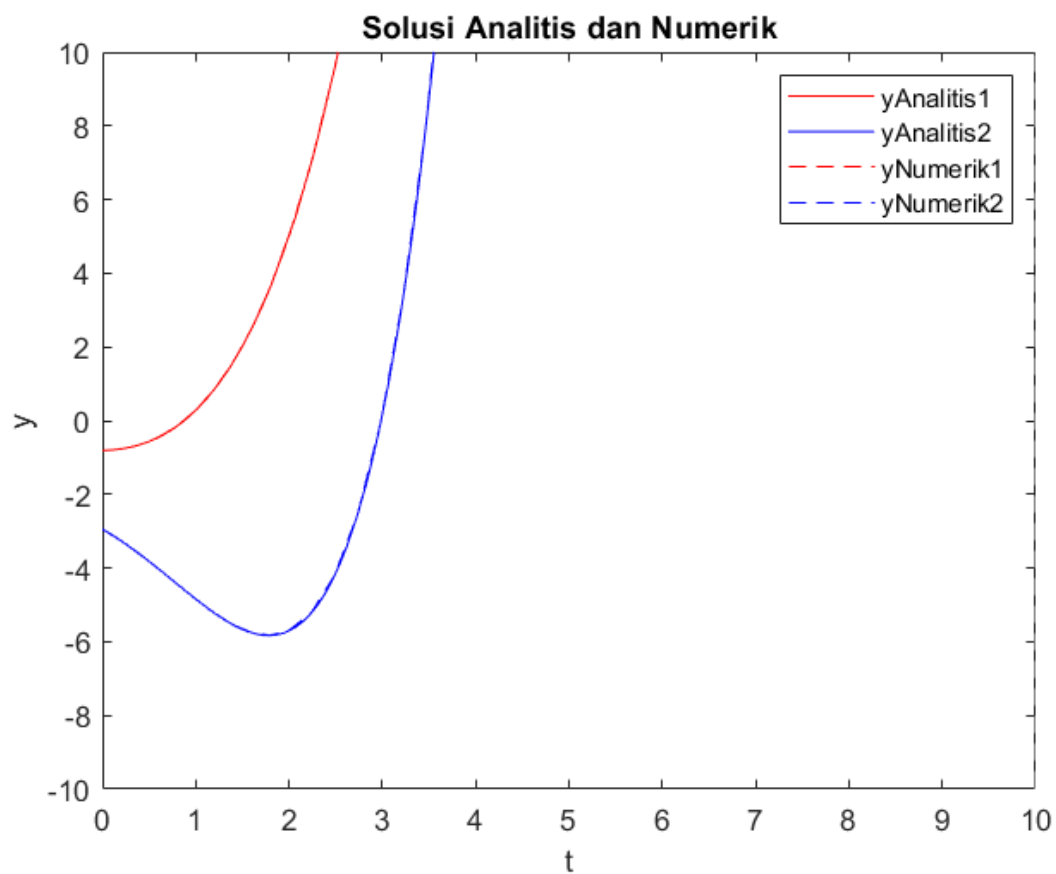
%Plot error yang dihasilkan
figure(2)
plot(t,error(:,1),t,error(:,2))
legend('y1','y2')
title('Galat Absolut');
xlabel('t')
ylabel('Galat absolut')

%Sub-fungsi untuk perhitungan numerik
function dydt = pers(t,y_numerik,A,b,p)
    dydt = [A(1,1)*y_numerik(1)+A(1,2)*y_numerik(2)+(b(1)*sin(p(1)*t))
; A(2,1)*y_numerik(1)+A(2,2)*y_numerik(2)+(b(2)*sin(p(2)*t))]
end

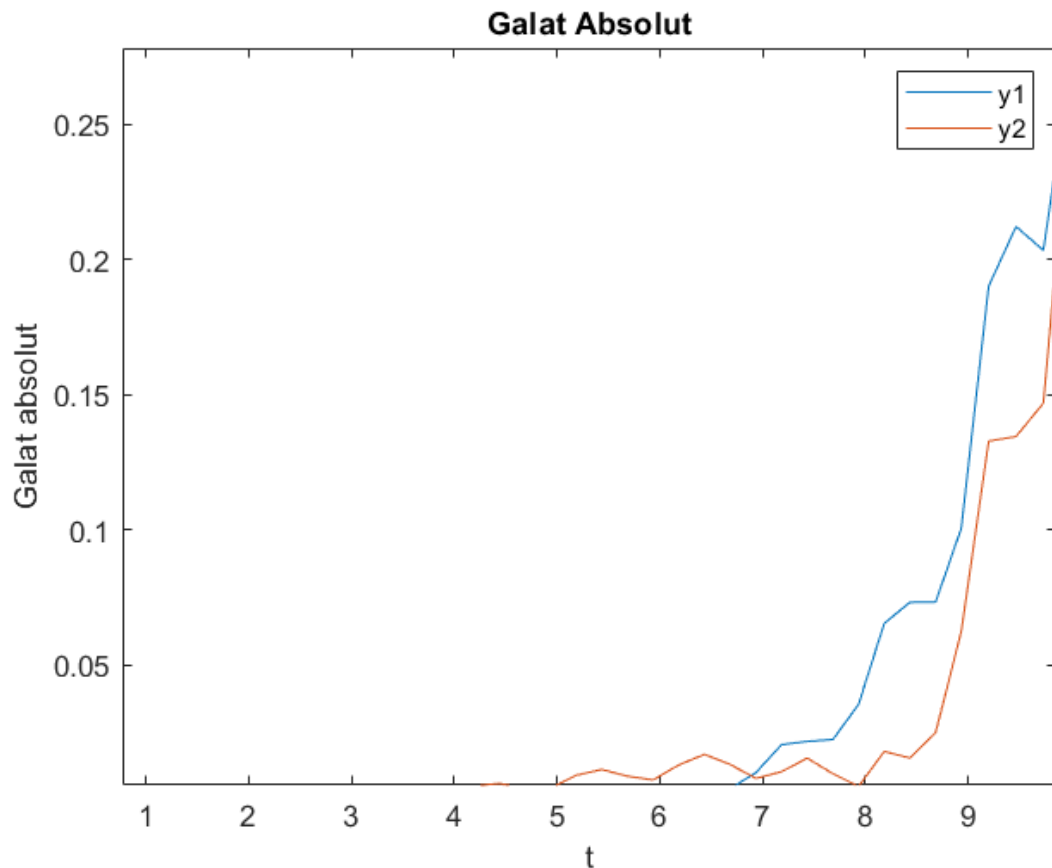
```

D. Hasil

a. Perbandingan Solusi Analitik dan Numerik



b. Perbandingan Error Absolut



E. Penjelasan

Fungsi MATLAB 'diff' digunakan untuk melakukan diferensiasi simbolik dan format fungsinya adalah sebagai berikut.

```
g(x) = diff(f(x), variabel penderivasi , orde derivasi);
```

Kemudian jika dalam suatu kasus hanya diketahui persamaan derivasi dan nilai di setiap titik-titik persamaan tersebut dapat digunakan fungsi 'dsolve'.

```
S = dsolve(Persamaan diferensial, kondisi awal);
```

Contohnya, `S = dsolve (eqn)` menyelesaikan persamaan diferensial `eqn`, di mana `eqn` adalah persamaan simbolik. Gunakan `diff` dan `==` untuk merepresentasikan persamaan diferensial. Misalnya, `diff (y, x) == y` mewakili persamaan $dy/dx = y$. Kemudian `S` akan menyelesaikan sistem persamaan diferensial dengan menetapkan `eqn` sebagai vektor persamaan tersebut.

Pada program yang dibuat, berdasarkan dari persamaan diferensial orde 1, tetapi dari persamaan tersebut memiliki 2 buah solusi sehingga perlu didefinisikan terlebih dahulu agar mempermudah dalam melakukan perhitungan. Kemudian baik solusi analitik maupun solusi numeriknya bergantung pada nilai random yang dibentuk pada definisi variabel awal. Nilai random tersebut akan mempengaruhi bentuk grafik yang dihasilkan.

Untuk melakukan komputasi secara analitik dapat dilakukan dengan menggunakan fungsi 'dsolve' dimana hasil tersebut akan berupa persamaan $y_1(t)$ dan $y_2(t)$. Karena hasil analitik berbentuk persamaan maka untuk mendapatkan hasil berupa bilangan tinggal melakukan substitusi nilai variabel t ke solusi analitik.

Untuk melakukan komputasi secara numerik dapat menggunakan beberapa perintah seperti 'ode45', 'ode23', dll. Contohnya adalah $[t, y] = \text{ode45}(\text{odefun}, \text{tspan}, y_0)$, di mana $\text{tspan} = [t_0 \text{ } t_f]$, mengintegrasikan sistem persamaan diferensial $y' = f(t, y)$ dari t_0 ke t_f dengan kondisi awal y_0 . Setiap baris dalam larik solusi y sesuai dengan nilai yang dikembalikan dalam vektor kolom t .

Semua solusi MATLAB ODE dapat menyelesaikan sistem persamaan bentuk $y' = f(t, y)$, atau masalah yang melibatkan matriks massa, $M(t, y) y' = f(t, y)$. Semua perhitungan numerik menggunakan sintaks yang serupa. Seperti ode23s hanya dapat menyelesaikan masalah dengan matriks massa jika matriks massa tersebut konstan. ode15s dan ode23t dapat menyelesaikan masalah dengan matriks massa yang tunggal, yang disebut persamaan aljabar-diferensial.

'ode45' adalah solver ODE serbaguna dan merupakan *solver* pertama yang dapat memecahkan Sebagian besar persamaan matematis. Namun, jika masalahnya persamaan diferensial yang *stiff* atau membutuhkan akurasi tinggi, maka ada *solver* ODE lain yang mungkin lebih cocok untuk masalah tersebut.

Jika dibandingkan analitik dan numerik, metode numerik memiliki waktu komputasi yang lebih lama karena dalam komputasinya diperlukan beberapa kali evaluasi dan dilakukan proses iterasi agar hasil yang didapatkan dapat menyerupai hasil analitik. Berdasarkan hasil yang didapatkan, error antara hasil analitik dan numerik tiap titik masih dapat diterima dan relative lebih kecil. Hal tersebut dipengaruhi bilangan random yang dibangkitkan, jika antara matriks-matriks random tersebut memiliki nilai yang bagus, maka error yang dihasilkan juga akan bagus. Begitupun sebaliknya, jika nilai random yang dibangkitkan membuat persamaan diferensial menjadi rumit maka hasil numerik akan jauh dari nilai analitik.