**LAPORAN TUGAS 8**

# **PENGGUNAAN 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](#_Toc72697990)

[**Daftar Isi** ii](#_Toc72697991)

[**A.** **Deskripsi Masalah** 1](#_Toc72697992)

[**B.** **Metode Penyelesaian Masalah** 1](#_Toc72697993)

[a. Algoritma 1](#_Toc72697994)

[b. Diagram Alir 2](#_Toc72697995)

[**C.** **Implementasi Program** 3](#_Toc72697996)

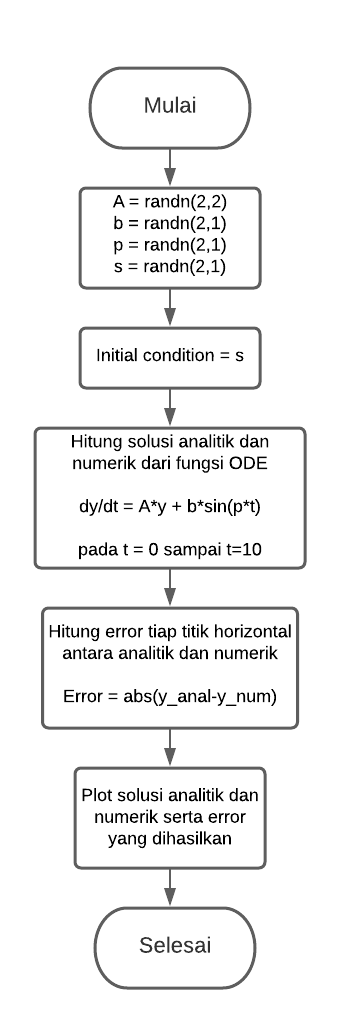
[**D.** **Hasil** 4](#_Toc72697997)

[a. Perbandingan Solusi Analitik dan Numerik 4](#_Toc72697998)

[b. Perbandingan Error Absolut 5](#_Toc72697999)

[**E.** **Penjelasan** 5](#_Toc72698000)

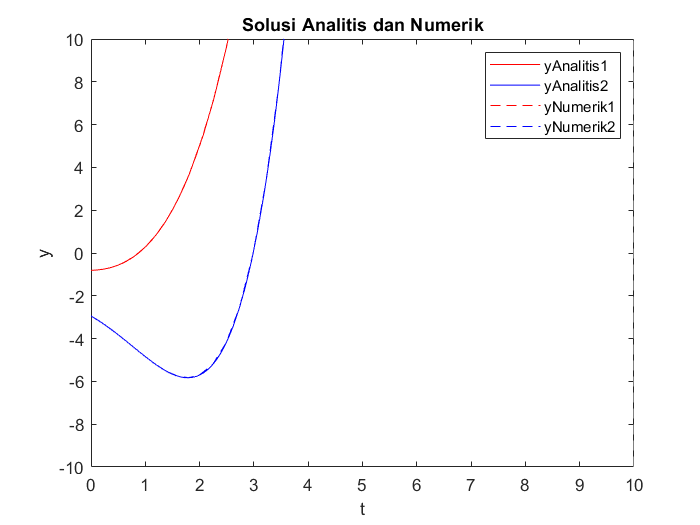
1. **Deskripsi Masalah**
2. Buatlah matriks A (2x2) dan vektor kolom b, p, dan s dengan ukuran matriks (2x1) menggunakan fungsi randn().
3. Hitunglah solusi analitis 𝑦(𝑡) = (𝑦1(𝑡) , 𝑦2(𝑡))𝑇, t adalah bilangan riil dari sistem persamaan diferensial 𝑦′(𝑡) = 𝐴𝑦(𝑡) + (𝑏1sin(𝑝1𝑡),𝑏2sin(𝑝2𝑡))𝑇, 𝑦(0) = 𝑠. Gunakan fungsi diff() dan dsolve().
4. Hitunglah solusi numerik 𝑦𝑛𝑢𝑚(𝑡) pada segmen t=[0,10] menggunakan fungsi ode45().
5. Plot fungsi berikut pada segmen t=[0,10],
   * Solusi analitis y1(t),y2(t)
   * Galat absolut
6. **Metode Penyelesaian Masalah**
7. Algoritma
8. Susun matriks A ukuran 2x2 dan matriks b, p, s dengan ukuran 2x1 dengan nilai tiap elemen berupa bilangan random.
9. Definisikan hasil solusi analitis berupa matriks 2x2.
10. Hitung solusi analitis dengan menggunakan fungsi dsolve() dengan *initial condition* adalah matriks s.
11. Ambil solusi analitis dengan 5 angka penting.
12. Definisikan rentang horizontal dari 0 sampai 10.
13. Hitung solusi numerik dengan masuk ke sub-fungsi ODE dengan menggunakan fungsi ode45().
14. Hitung error absolut yang dihasilkan dari hasil analitik dan numerik pada tiap titik.
15. Plot hasil analitik, numerik, dan error yang dihasilkan.
16. Diagram Alir



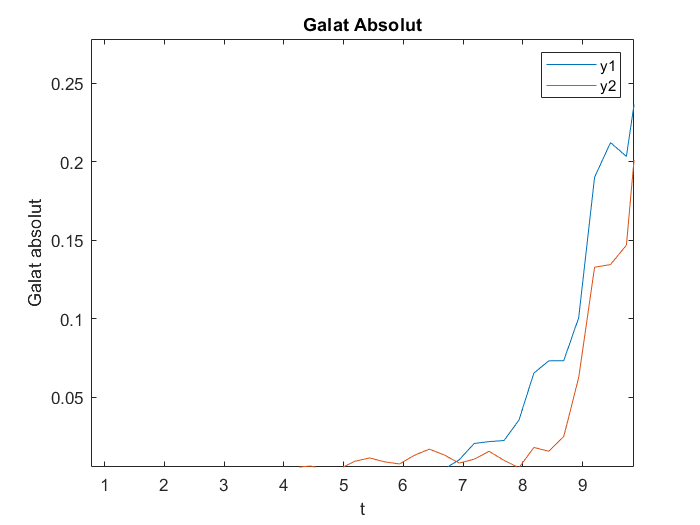
1. **Implementasi Program**

|  |
| --- |
| 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 bsyms y1(t) y2(t)%Definisi ulang ODEODE = [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 ICic = [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 10tspan = [0 10];%Definisi ulang kondisi awaly0 = [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 horizontalerror = zeros(size(t,1),2)%Perhitungan error tiap segmenfor 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 grafikfigure(1)fplot(y\_analitik1,'r')hold onfplot(y\_analitik2,'b')hold onplot(t,y\_numerik(:,1),'--r',t,y\_numerik(:,2),'--b')hold offylim([-10 10]);xlim([0 10]);legend('yAnalitis1','yAnalitis2','yNumerik1','yNumerik2')title('Solusi Analitis dan Numerik');xlabel('t')ylabel('y')%Plot error yang dihasilkanfigure(2)plot(t,error(:,1),t,error(:,2))legend('y1','y2')title('Galat Absolut');xlabel('t')ylabel('Galat absolut')%Sub-fungsi untuk perhitungan numerikfunction 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 |

1. **Hasil**
   1. Perbandingan Solusi Analitik dan Numerik



* 1. Perbandingan Error Absolut



1. **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 persaman 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 benut grafik yang dihasilkan.

Untuk melakukan komputasi secara analitik dapat dilakukan dengan menggunakan fungsi ‘dsolve’ dimana hasil tersebut akan berupa persaman y1(t) dan y2(t). Karena hasil analitik berbentuk persamaan maka untuk mendapatkan hasil berupa bilangan tinggal melakukan subtitusi nilai variabel t ke solusi analitik.

Untuk melakukan komputasi secara numerik dapat menggunakan beberapa perintah seperti ‘ode45’, ‘ode23’, dll. Contohnya adalah [t, y] = ode45 (odefun, tspan, y0), di mana tspan = [t0 tf], mengintegrasikan sistem persamaan diferensial y ′ = f (t, y) dari t0 ke tf dengan kondisi awal y0. 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 beberapakali 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.