PERSAMAAN DIFFERENSIAL

Oleh

Muh. Firdaus

F1D020054

1. Tujuan

Tujuan dari dilakukan pratikum ini adalah sebagai berikut:

a. Untuk memahami proses persamaan diferensial dengan metode Runge Kutta
b. Untuk proses penyelesaian persamaan diferensial orde tinggi

2. Algoritma Penyelesaian

Selesaikan persamaan diferensial berikut dengan metode RK orde 3:

$0.25y'' + 64\,xy' + y = 0$ dengan $y(0)=1$ dan $y'(0)=-8$

Metode Runge Kutta orde 3 dapat diselesaikan dengan rumus berikut:

$k_1 = hf(x_r, y_r)$

$k_2 = h(f(x_r+p_1h, y_r+q_{11}k_1)$

$k_3 = h(f(x_r+p_2h, y_r+q_{21}k_1+q_{22}k_2)$

$y_{r+1} = y_r + a_1k_1 + a_2k_2 + a_3k_3$

dengan menggunakan penurunan rumus yang ada didapatkan :

$k_1 = hf(x_r, y_r)$

$k_2 = h(f(x_r+1/2\,h, y_r+1/2\,k_1)$

$k_3 = h(f(x_r+h, y_r-k_1+2k_2)$

$y_{r+1} = y_r + 1/6(\,k_1 + 4k_2 + k_3)$

a. RungeKutta.java

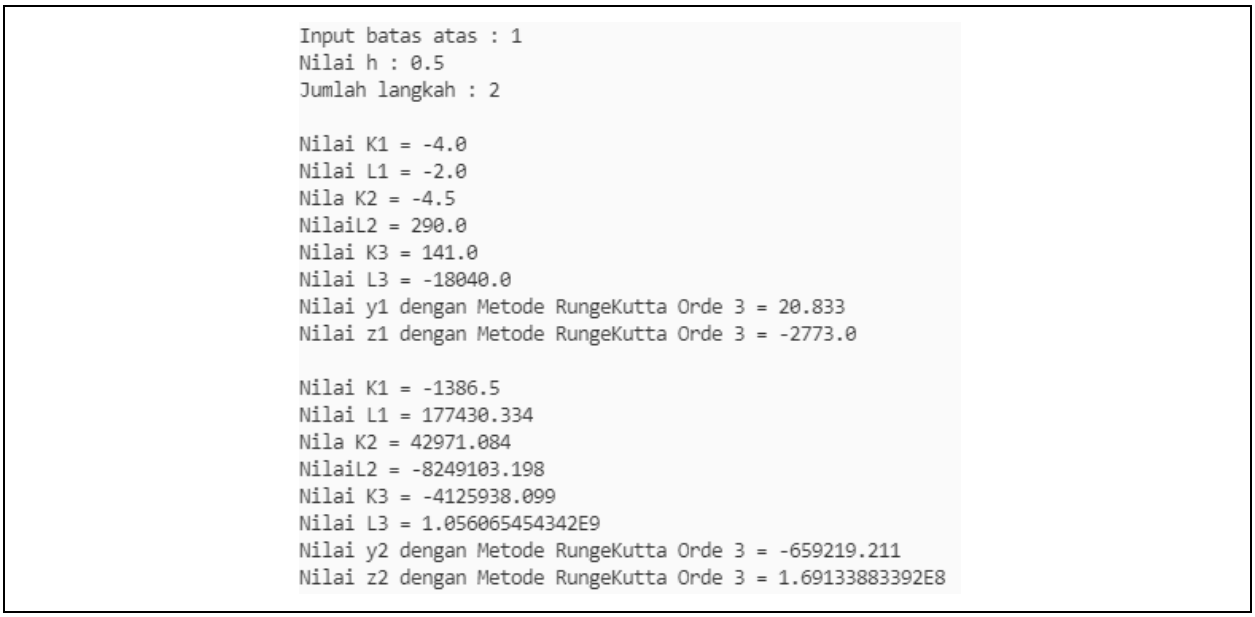
```
import java.text.DecimalFormat;
public class RungeKutta {
        DecimalFormat df = new DecimalFormat("#.###");
    public double gx(double x,double y,double z){
        double g = (4*((-64*x*z)-y));
        return g;
    }
    public double fx(double x,double y,double z){
        return z;
    }
    public void hitung(double[] x,double[] y,double[] z, double n, double h){
        double K1,K2,k3;
        double L1,L2,L3;
        for(int i=1; i<=n; i++){
            System.out.println();
            K1= Double.parseDouble(df.format( h * fx(x[i-1],y[i-1],z[i-1])));
            System.out.println("Nilai K1 = "+K1);
            L2 = Double.parseDouble(df.format( h * gx(x[i-1],y[i-1],z[i-1])));
            System.out.println("Nilai L1 = "+L2);
            K2 = Double.parseDouble(df.format( h * fx(x[i-1]+h/2,y[i-1]+K1/2,z[i-
            1]+L2/2)));
            System.out.println("Nila K2 = "+K2);
            L2 = Double.parseDouble(df.format( h * gx(x[i-1]+h/2,y[i-1]+K1/2,z[i-
            1]+L2/2)));
            System.out.println("NilaiL2 = "+L2);
            k3 = Double.parseDouble(df.format( h * fx(x[i-1]+h,
            y[i-1]-K1+2*K2,z[i-1]-L2+2*L2)));
```

```
            System.out.println("Nilai K3 = "+k3);
            L3 = Double.parseDouble(df.format( h * gx(x[i-1]+h,y[i-1]-K1+2*K2,z[i-
            1]-L2+2*L2)));
            System.out.println("Nilai L3 = "+L3);
            y[i] = Double.parseDouble(df.format( y[i-1] + ((K1 + (4*K2) + k3)/6 )));
            z[i]= Double.parseDouble(df.format( z[i-1] + ((L2 + (4*L2) + L3)/6 )));
            x[i]=x[i-1]+h;
            System.out.println("Nilai y"+i+" dengan Metode RungeKutta Orde 3 =
            "+y[i]);
            System.out.println("Nilai z"+i+" dengan Metode RungeKutta Orde 3 =
            "+z[i]);
        }
    }
}
```

b. Main.java

```
import java.text.DecimalFormat;
import java.util.Scanner;
    public class Main {
        public static void main(String[] args) {
        DecimalFormat df = new DecimalFormat("#.###");
        Scanner input = new Scanner(System.in);
        RungeKutta rk = new RungeKutta ();
        System.out.print("Input batas atas : ");
        double b=input.nextDouble();
        System.out.print("Nilai h : ");
        double h=input.nextDouble();
        int n=(int) (b/h);
        System.out.println("Jumlah langkah : "+n);
        double[] x = new double[n+1];
        double[] y = new double[n+1];
        double[] z = new double[n+1];
        x[0]=0;y[0]=1;z[0]=-8;
        rk.hitung(x, y, z, n, h);
    }
}
```

3. Hasil Uji Coba dan Analisa Singkat



Gambar 5.1 Hasil Percobaan Runge Kutta Orde 3

Pada **Gambar 5.1** merupakan hasil penyelesaian persamaan differensial menggunakan metode Runge Kutta (RK) orde tiga. Untuk menyelesaikan persaaman terebut dibutuhkan nilai batas atas dan h untuk mengetahui jumlah iterasi yang akan dilakukan. Pada gambar tersebut dimasukkan nilai batas atas = 1 dan nilai h = 0.5 sehingga didapatkan jumlah iterasinya adalah 2 menggunkan rumus rumus $n = (b - a)/h$. Pada orde pertama perlu ditentukan nilai $k1$ dengan rumus $k1$ = hf(xr,yr) sehingga diperoleh -4.0, mencari $k2$ dengan rumus $k2$ = h(f($xr + p1$h, $yr + q11k1$) sehingga diperoleh nilai -4.5 dan mencari $k3$ dengan rumus $k3$ = h(f($xr + p2$h, $yr + q21k2$) sehingga diperoleh nilai 141.0. Kemudian perlu dicari nilai $y1/(y')$ dengan rumus $yr + 1 = yr + a1k1 + a2k2 + a3k3$, dengan nilai $k1$, $k2$, $k3$ yang sudah ditemukan sebelumnya lalu masukan ke dalam rumus y sehingga nilai $y1/(y')$ = 20.833. Setelah mendapatkan nilai y1/(y'), lalu mencari nilai $z1/(y'')$ dengan mencari nilai L1, L2 dan L3 dengan rumus yang sama seperti di atas. Dengan mencari masing-masing nilainya lalu di dapatkan nilai $z1/(y'')$ = -2773.0. Kemudian langkah tersebut akan diulang sebanyak nilai n atau langkah yang ditentukan sesuai dengan nilai b, a dan nilai h.

4. Kesimpulan

1. Poses penyelesaian persamaan diferensial menggukanakn Runge Kutta dapat menggunakan program dengan Bahasa Java. Pada saat akan membuat program dari metode Runge Kutta orde 3, dijabarkan terlebih dahulu penurunan rumusnya sehingga akan menghasilkan nilai yang sama. Metode ini dapat diselesaikan dengan rumus yang memiliki prasyarat bahwa komponen-komponen lainnya telah juga diselesaikan menggunakan rumus, yaitu nilai k.
2. Metode PDB yang berorde rendah seperti metode Euler memperlihatkan hasil yang sangat menyimpang (divergen) dengan solusi sejatinya ketika jumlah langkahnya membesar, sedangkan solusi dengan metode Runge-Kutta memperlihatkan kestabilan pada setiap langkahnya. Ini disebabkan galat per langkah pada metode Euler semakin menumpuk dengan bertambahnya lebih sering dipakai.

5. Referensi

[1] Wijaya, IGP Suta, Persamaan Differensial. Mataram: Universitas Mataram.

[2] Wijaya, IGP Suta, Sistem Persamaan Differensial. Mataram: Universitas Mataram.

[3] Sasongko, Setia Budi. “Metode Numerik dengan Scilab”, ANDI, Yogyakarta, 2010.

[4] Rinaldi, Munir. “Metode Numerik”, INFORMATIKA, Bandung, 2003.

[5] I Ketut Adi Atmika.“Metode Numerik", Universitas Udayana 2016.