Algoritma : Jurnal Matematika, Ilmu pengetahuan Alam, Kebumian dan Angkasa
Vol. 2, No. 6 November 2024
e-ISSN: 3046-5427; p-ISSN: 3032-6230, Hal 10-19
DOI : https://doi.org/10.62383/algoritma.v2i6.263
Available online at: https://journal.arimsi.or.id/index.php/Algoritma

Penerapan Metode Runge-Kutta Orde 3 Untuk Penyelesaian Persamaan Diferensial Biasa
(Studi Kasus di Matlab)

Asri Cahyati Sitorus Pane[1*], Novaria Br. Saragih[2] , Jadata Dompak Ambarita[3]
[1-3]PSM C 2023 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Negeri Medan,Indonesia
asrichyti.s@gmail.com[1*],novariasaragih5@gmail.com[2],jadata.ambarita098@gmail.com[3]

Alamat: Jl. William Iskandar Ps. V, Kenangan Baru, kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara
Korespondensi penulis: asrichyti.s@gmail.com*

Abstract.*This research studies the application of the nth order Runge-kutta method as a numerical solution to ordinary differential equations. This method was chosen because it is able to provide high accuracy and flexibility in various PDB problems. We implement the nth-order Runge-Kutta algorithm in MATLAB and compare with other numerical methods, such as Euler's method. The results show that the nth order Runge-Kutta method is able to produce more accurate solutions, especially for nonlinear systems. This research makes a significant contribution to the development of numerical solutions for PDB and shows the potential of MATLAB as an effective tool for numerical simulation. Sensitivity analyzes of parameters and time steps were also performed to understand the impact of variations on stability and convergence.*

Keywords*: Differential Equation, MATLAB, Numeric, Runge Kutta*

Abstrak.Penelitian ini mempelajari penerapan dari metode Runge-kutta orde n sebagai solusi numerik persamaan diferensial biasa. Metode ini dipilih karena mampu memberikan akurasi dan fleksibilitas yang tinggi dalam berbagai permasalahan PDB. Penulis mengimplementasikan algoritma Runge-Kutta orde n di MATLAB dan membandingkan dengan metode numerik lainnya, seperti metode Euler. Hasilnya menunjukkan bahwa metode Runge-Kutta orde n mampu menghasilkan solusi yang lebih akurat khususnya untuk sistem nonlinier. Penelitian ini memberikan kontribusi yang signifikan terhadap pengembangan solusi numerik untuk PDB dan menunjukkan potensi MATLAB sebagai alat yang efektif untuk simulasi numerik. Analisis sensitivitas parameter dan langkah waktu juga dilakukan untuk memahami dampak variasi terhadap stabilitas dan konvergensi.

Kata kunci: MATLAB, Numerik, Persamaan Diferensial, Runge Kutta

1. LATAR BELAKANG

Persamaan differensial merupakan persamaan yang identik dengan perubahan atau derivative. Di sisi lain, beberapa fenomena alam yang terjadi, erat sekali dengan perubahan sehingga fenomena tersebut sering kali dianalisa dengan pendekatan persamaan differensial sebagai alternatif pendekatan matematika. PDB dapat menjadi salah satu alternatif matematik yang dapat digunakan untuk menganalisis suatu persoalan alam, seperti penyelesaian dalam kasus rangkaian listrik (Rijoly & Rumlawang, 2020). Menghadapi kesulitan tersebut biasanya PDB dapat diselesaikan dengan metode numerik, metode numerik tentu saja memiliki kelebihan dan kekurangan tersendiri. Kelebihannya adalah bentuk persamaannya menjadi galat sehingga bentuk PDB seperti apapun dapat diselesaikan, namun kekurangannya diperlukan

Received: September 16,2024; Revised: September 30,2024; Accepted: Oktober 09, 2024; Oline Available : Oktober 11, 2024

jumlah pengulangan atau iterasi untuk menghasilkan solusi yang akurat layaknya solusi analitik yang bergantung terhadap bentuk dari PDB yang diselesaikan (Pandia & Sitepu, 2021). Metode numerik yang umum digunakan untuk mendapatkan solusi persamaan nonlinier adalah metode Newton, Bisection dan secant. Sedangkan untuk medapatkan solusi numerik dari persamaan diferensial biasa adalah metode Runge Kutta. Semua metode tersebut hanya berfokus pada persamaan dengan satu variabel saja dan tidak dapat digunakan jika persamaannya memiliki lebih dari satu variable (Sitompul & Siahaan, 2022b). Metode numerik juga seringkali dibuat dengan MATLAB untuk mempermudah dalam eksekusinya terutama sebagai media pembelajaran (Nugraha & Nurullaeli, 2023).

2. KAJIAN TEORITIS

Persamaan diferensial merupakan persamaan fungsi turunan yang ada dalam permasalahan matematika. Metode yang digunakan untuk solusi persamaan diferensial adalah metode analitik, tetapi ada persamaan yang tidak dapat diselesaikan dengan menggunakan metode analitik sehingga diperlukan adanya metode lain untuk mendekati nilai sebenarnya yaitu dengan menggunakan metode numerik. Persamaan diferensial (differential equation) adalah sebuah persamaan yang memuat satu atau lebih variabel tak-bebas dan turunan-turunannya terhadap variabel-variabel bebas (Rifandi & Abdy, 2023). Persamaan Diferensial Biasa (Ordinary Differential Equation) adalah sebuah persamaan diferensial yang hanya melibatkan satu variabel bebas. Suatu persamaan diferensial biasa dapat dinyatakan dalam bentuk :

$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^ny}{dx^n}\right) = 0$$

Permasalahan persamaan diferensial biasa dapat diselesaikan dengan metode numerik seperti Metode Runge Kutta dan Metode Euler.

Dengan Rumus Metode Runge Kutta :

$$y_{i+1} = y_i + a_1k_1 + a_2k_2 + \cdots + a_nk_n$$

$dimana$:

$$k_1 = hf(x_iy_i)$$

$$k_2 = hf(x_i{+}p_1h, y_i + q_{11}k_1)$$

$$k_3 = hf(x_i{+}p_2h, y_i + q_{21}k_1 + q_{22}k_2)$$

$$k_n = hf(x_i{+}p_{n-1}h, y_i + q_{n-1,1}k_1 + q_{n-2,2}k_2 + \cdots + q_{n-1,1}k_{n-1})$$

Rumus Metode Euler :

$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$

$y_1 = y_0 + hf(x_0, y_o)$

$y_2 = y_1 + hf(x_1, y_1)$

3. METODE PENELITIAN

Pada penelitian ini metode yang digunakan oleh peneliti dibagi menjadi tiga tahapan, yakni : Pertama, penyusunan penyelesaian persamaan numerik yang dilakukan sebagai solusi dari PDB baik orde satu, orde dua ataupun orde 3. Kedua, perancangan pada MATLAB yang disesuaikan dengan persamaan yang diinginkan. Ketiga, mengevaluasi hasil numerik dengan solusi analitik untuk persamaan PDB orde satu, orde dua ataupun orde tiga yang sederhana.Solusi numerik dengan metode runge kutta pada persamaan diferensial orde 3 dengan menggunakan algoritma runge kutta orde 3 :

$y_{i+1} = y_i + \frac{1}{6}(k_1 + 4k_2 + k_3)$

Dimana :

$k_1 = h.f(x_i, y_1)$

$k_2 = h.f\left(x_i + \frac{1}{2}h, y_1 + \frac{1}{2}k_1\right)$

$k_3 = h.f(x_i + h, y_1 - k_1 + 2k_2)$

Solusi numerik dengan metode euler pada persamaan diferensial dengan menggunakan algoritma euler :

$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$

$y_1 = y_0 + hf(x_0, y_o)$

$y_2 = y_1 + hf(x_1, y_1)$

Dengan diberikannya rumus Runge Kutta dan Euler yang merupakan Penyelesaian numerik dari permasalahan persamaan diferensial biasa yang akan dijadikan landasan teori dalam menyelesaikan persamaan diferensial biasa.

4. HASIL DAN PEMBAHASAN

Akan disajikan 1 contoh kasus dalam tulisan ini untuk menunjukkan perbandingan keefektifan metode runge kutta dengan metode euler dalam penyelesaian permasalahan persamaan diferensial biasa serta diimplementasikan ke dalam MATLAB.

Contoh Kasus :

Diberikan suatu fungsi $\frac{dy}{dx} = f(x,y) = (-2x^3 + 12x^2 - 20x + 8{,}5)$ dimana x = 0 dan y(0) = 1 dari x = 0 dan x = 1, panjang langkah h = 0,5

Penyelesaian :

1. Metode Runge Kutta Orde 3

$Dik:\ x_0 = 0 \rightarrow y(0) = 1\,;h = 0{,}5\,;\ x_n = 1$

$n = \frac{x_n - x_0}{h} = \frac{1-0}{0{,}5} = 2$

$x_1 = 0{,}5 \rightarrow y_1 = ?$

$k_1 = hf(x_i, y_i)\,; y_1 \rightarrow i = 0$

$= 0{,}5 \times f(x_0, y_0)$

$= 0{,}5(-2(0)^3 + 12(0)^2 - 20(0) + 8{,}5)$

$= 0{,}5(8{,}5)$

$= 4{,}25$

$$k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$

$$= 0{,}5f\left(0 + \frac{0{,}5}{2}, 1 + \frac{4{,}25}{2}\right)$$

$= 0{,}5f(0{,}25\,; 3{,}125)$

$= 0{,}5f(-2(0{,}25)^3 + 12(0{,}25)^2 - 20(0{,}25) + 8{,}5) = (0{,}5 \times 4{,}22)$

$= 2{,}11$

$k_3 = hf(x_i + h\,; y_i - k_1 + 2k_2)$

$= 0{,}5f\big(x_0 + 0{,}5\,; 1 - 4{,}25 + 2(2{,}11)\big)$

$= 0{,}5f\big(0 + 0{,}5\,; 1 - 4{,}25 + 2(2{,}11)\big)$

$= 0{,}5f(0{,}5\,; 0{,}97)$

$= 0{,}5f(-2(0{,}5)^3 + 12(0{,}5)^2 - 20(0{,}5) + 8{,}5)$

$= (0{,}625)$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

$$y_1 = 1 + \frac{1}{6}(4{,}25 + 4(2{,}11) + 0{,}625)$$

$$y_1 = 1 + \frac{1}{6}(13{,}315)$$

$= 3{,}22$

Untuk $x_2 = 1 \rightarrow y_2 = ?\,;$ $y_{1+1} = y_2 \rightarrow i = 1$

$k_1 = hf(x_i, y_i)$

e-ISSN: 3046-5427; p-ISSN: 3032-6230, Hal 10-19

$$= 0{,}5 \times f(x_1, y_1)$$

$$= 0{,}5(-2(0{,}5)^3 + 12(0{,}5)^2 - 20(0{,}5) + 8{,}5)$$

$$= 0{,}625$$

$$k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$

$$= 0{,}5f\left(0{,}5 + \frac{0{,}5}{2}, 3{,}22 + \frac{0{,}625}{2}\right)$$

$$= 0{,}5f(0{,}75\,; 3{,}533)$$

$$= 0{,}5f(-2(0{,}75)^3 + 12(0{,}75)^2 - 20(0{,}75) + 8{,}5)$$

$$= -0{,}297$$

$$k_3 = hf(x_i + h\,; y_i - k_1 + 2k_2)$$

$$= 0{,}5f\big(0{,}5 + 0{,}5\,; 3{,}22 - 0{,}625 + 2(-0{,}297)\big)$$

$$= 0{,}5f(1\,; 2{,}001)$$

$$= 0{,}5f(-2(1)^3 + 12(1)^2 - 20(1) + 8{,}5) = -0{,}75$$

$$y_{1+1} = y_1 + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

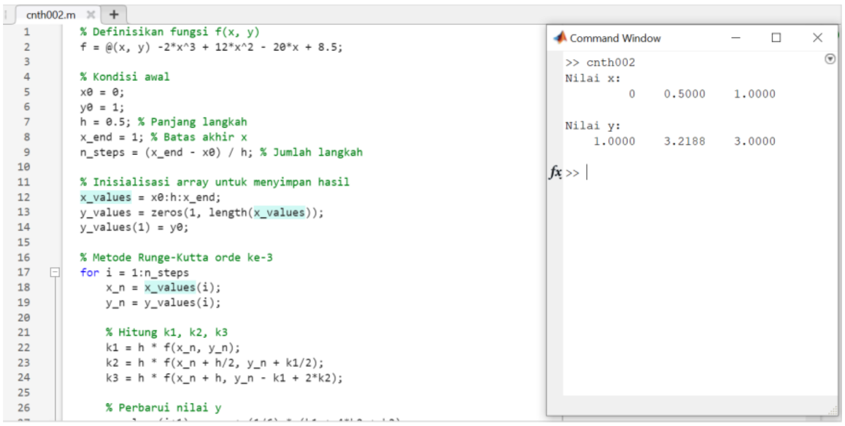
$$y_2 = 3{,}22 + \frac{1}{6}(0{,}625 + 4(-0{,}297) - 0{,}75)$$

$$y_2 = 3{,}22 + \frac{1}{6}(-1{,}313)$$

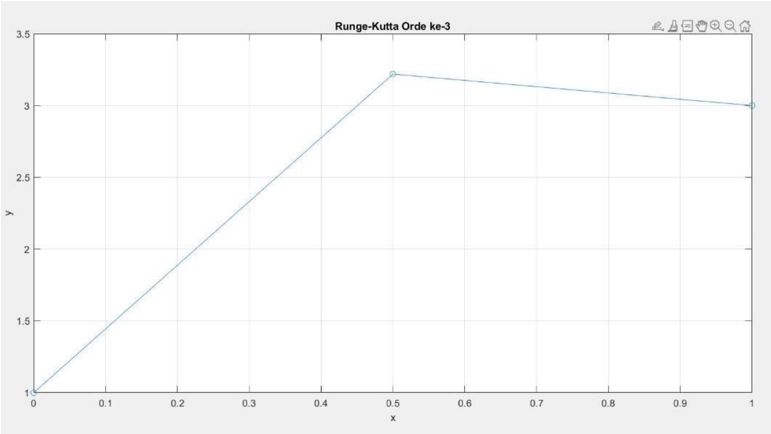
$$= 3{,}0012$$

Maka, ketika $x_1 = 0{,}5$, $y_1 = 3{,}22$ dan ketika $x_2 = 1$, $y_2 = 3{,}0012$

Pengaplikasian pada MATLAB



Gambar 1. Pengaplikasian Pada MATLAB



Gambar 2. Grafik Hasil MATLAB dari Runge Kutta Orde 3

Pemograman MATLAB untuk percobaan numerik diatas :

```
clc
close all
clear all
% Definisikan fungsi f(x, y)
f = @(x, y) -2*x^3 + 12*x^2 - 20*x + 8.5;
% Kondisi awal
x0 = 0;
y0 = 1;
h = 0.5; % Panjang langkah
x_end = 1; % Batas akhir x
n_steps = (x_end - x0) / h; % Jumlah langkah
% Inisialisasi array untuk menyimpan hasil
x_values = x0:h:x_end;
y_values = zeros(1, length(x_values));
y_values(1) = y0;
% Metode Runge-Kutta orde ke-3
for i = 1:n_steps
    x_n = x_values(i);
    y_n = y_values(i);
        % Hitung k1, k2, k3
    k1 = h * f(x_n, y_n);
    k2 = h * f(x_n + h/2, y_n + k1/2);
    k3 = h * f(x_n + h, y_n - k1 + 2*k2);
```

```
        % Perbarui nilai y
    y_values(i+1) = y_n + (1/6) * (k1 + 4*k2 + k3);
end
% Tampilkan hasil
disp('Nilai x:');
disp(x_values);
disp('Nilai y:');
disp(y_values);
% Plot solusi
plot(x_values, y_values, '-o');
xlabel('x');
ylabel('y');
title('Runge-Kutta Orde ke-3');
grid on;
```

Metode Euler

$Dik:\ x_0 = 0 \rightarrow y(0) = 1\,; h = 0{,}5\,;\ x_n = 1$

$y_{i+1} = y_i + hf(x_i\,, y_i)$

$x_1 = x_0 + h$

$\quad = 0 + 0{,}5$

$\quad = 0{,}5$

$y_1 = y_0 + hf(x_0\,, y_0)$

$\quad = 1 + 0{,}5(-2(0)^3 + 12(0)^2 - 20(0) + 8{,}5)$

$\quad = 1 + 0{,}5(8{,}5)$

$\quad = 5{,}25$

$x_2 = x_1 + h$

$\quad = 0{,}5 + 0{,}5$

$\quad = 1$

$y_2 = y_1 + hf(x_1\,, y_1)$

$\quad = 5{,}25 + 0{,}5(-2(0{,}5)^3 + 12(0{,}5)^2 - 20(0{,}5) + 8{,}5)$

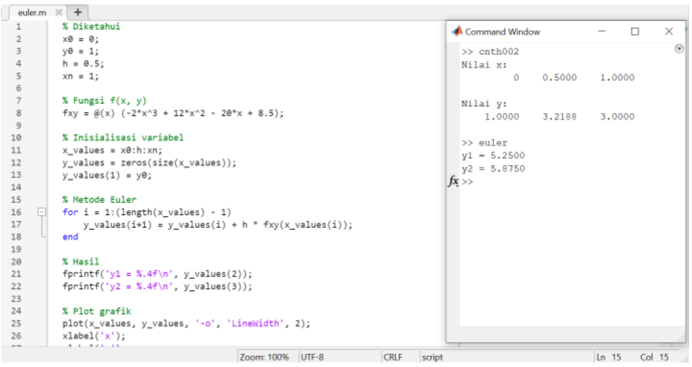
$\quad = 5{,}25 + 0{,}5(1{,}25)$

$\quad = 5{,}875$

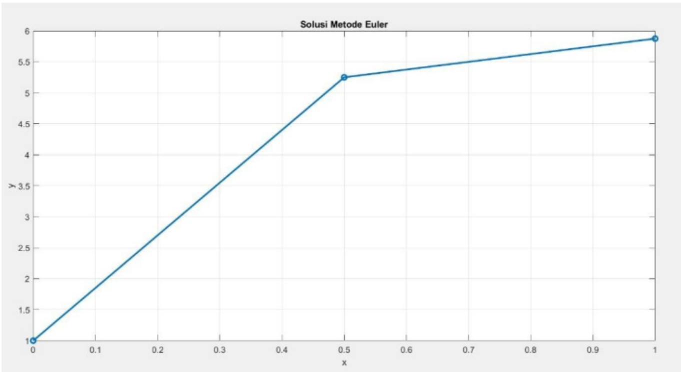
Maka, ketika $x_1 = 0{,}5$, $y_1 = 5{,}25$ dan ketika $x_2 = 1$, $y_2 = 5{,}875$

Terdapat Perbedaan hasil pada y_1 dan y_2 antara metode Runge kutta dan Euler. Hasil dari metode Euler dan Runge-Kutta tidak sama karena metode Runge-Kutta lebih akurat daripada metode Euler. Metode Runge-Kutta memiliki kesalahan pemotongan lokal yang lebih kecil daripada metode Euler untuk panjang langkah yang sama, sehingga memberikan hasil yang lebih akurat. Metode runge kutta lebih kompleks dan akurat dibandingkan metode Euler.

Pengaplikasian Pada MATLAB



Gambar 3. Pengaplikasian Pada MATLAB



Gambar 4. Grafik Hasil MATLAB dari Euler

Pemograman MATLAB untuk percobaan numerik diatas :

```
clc
close all
clear all
% Diketahui
x0 = 0;
y0 = 1;
h = 0.5;
xn = 1;

% Fungsi f(x, y)
```

```
fxy = @(x) (-2*x^3 + 12*x^2 - 20*x + 8.5);
% Inisialisasi variabel
x_values = x0:h:xn;
y_values = zeros(size(x_values));
y_values(1) = y0;
% Metode Euler
for i = 1:(length(x_values) - 1)
    y_values(i+1) = y_values(i) + h * fxy(x_values(i));
end
% Hasil
fprintf('y1 = %.4f\n', y_values(2));
fprintf('y2 = %.4f\n', y_values(3));
% Plot grafik
plot(x_values, y_values, '-o', 'LineWidth', 2);
xlabel('x');
ylabel('y');
title('Solusi Metode Euler');
grid on;
```

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa penyelesaian persamaan diferensial biasa menggunakan metode runge kutta lebih efektif dan akurat daripada menggunakan metode euler dikarenakan metode runge kutta menggunakan pendekatan lebih kompleks dalam menghitung nilai solusi tiap langkah, sehingga kesalahan (error) yang terjadi lebih kecil. Metode Runge Kutta menggunakan beberapa tahap perhitungan pada setiap langkah untuk menghitung rata-rata nilai kemiringan (turunan) dari beberapa titik antara titik awal dan akhir di interval tersebut. Karena menggabungkan informasi dari berbagai titik, hasilnya lebih mendekati solusi eksak dengan tingkat kesalahan yang lebih kecil dibandingkan metode Euler.

6. DAFTAR REFERENSI

Alda, R. I., & Muh. I. (2024). Solusi numerik model penyebaran penyakit ISPA menggunakan metode Runge Kutta orde lima (Studi kasus: Kabupaten Gowa). Jurnal MSA (Matematika dan Statistika serta Aplikasinya), 11(2), 87–92. https://doi.org/10.24252/msa.v11i2.43086

Fatahillah, A., Istiqomah, M., & Dafik, D. (2021). Pemodelan matematika pada kasus kecanduan game online menggunakan metode Runge-Kutta orde 14. Limits: Journal of Mathematics and Its Applications, 18(2), 129. https://doi.org/10.12962/limits.v18i2.6854

Hikmawati Pathuddin, R. I., & Rismayanti. (2022). Solusi numerik model penyebaran penyakit Covid-19 di Sulawesi Selatan dengan metode Runge-Kutta orde empat. Jurnal MSA (Matematika dan Statistika serta Aplikasinya), 10(2), 116–123. https://doi.org/10.24252/msa.v10i2.33817

Ludji, D. G., & Buan, F. C. H. (2023). Penerapan metode Runge-Kutta orde 4 pada pemodelan penularan penyakit cacar monyet. Journal of Mathematics Computations and Statistics, 6(1), 1. https://doi.org/10.35580/jmathcos.v6i1.37110

Munir, R. (2021). Metode numerik revisi kelima. Informatika Bandung.

Nugraha, A. M., & Nurullaeli, N. (2023). Graphical User Interface (GUI) Matlab untuk penyelesaian persamaan diferensial biasa orde satu. Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi), 7(1), 182–185. https://doi.org/10.30998/semnasristek.v7i1.6269

Rijoly, M. E., & Rumlawang, F. Y. (2020). Penyelesaian numerik persamaan diferensial orde dua dengan metode Runge-Kutta orde empat pada rangkaian listrik seri LC. Tensor: Pure and Applied Mathematics Journal, 1(1), 7–14. https://doi.org/10.30598/tensorvol1iss1pp7-14

Setiawan, L. I., & Mungkasi, S. (2021). Penyelesaian model epidemi SIR menggunakan metode Runge-Kutta orde empat dan metode Adams-Bashforth-Moulton. Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika, 18(2), 55–61. https://doi.org/10.33751/komputasi.v18i2.3623

Sharma, P. L., & Kumar, A. (2021). Demonstration study on Runge-Kutta fourth order method by using MATLAB programming. 17(4), 1–9. https://doi.org/10.9790/5728-1704030109

Sitompul, H. A., & Siahaan, E. (2022a). Akurasi solusi numerik pada persamaan gelombang berdimensi-satu. Jurnal Penelitian Fisikawan, 5, 54–63.

Sitompul, H. A., & Siahaan, E. W. B. (2022b). Solusi numerik persamaan diferensial biasa orde dua dengan sistem persamaan nonlinier. Jurnal Ilmiah Teknik Sipil, 11(2), 379. https://doi.org/10.46930/tekniksipil.v11i2.2841

Sitompul, H. A., & Siahaan, E. W. B. (2024). Solusi persamaan diferensial biasa orde tinggi dengan metode polinomial dan Runge Kutta. Jurnal Penelitian Fisikawan, 7(1), 32. https://doi.org/10.46930/jurnalpenelitianfisikawan.v7i1.4185

Ulfa, F., & Wartono. (2019). Modifikasi metode Runge-Kutta orde empat klasik menggunakan deret Lehmer dengan P=1 dan P=4. Journal of Chemical Information and Modeling, 1, 7–15.