



OPTIMALISASI METODE *NEURAL NETWORK* PADA AYUNAN TEREDAM HARMONIS

F.H. Faturahman¹, F. Armawanto, D.A. Faradita, T. Wahyuni, W. Kurniawan, N. Khoiri, D.E. Handayani, dan J. Saefan

¹Pendidikan Fisika, Universitas PGRI Semarang, Kota Semarang
e-mail: fikrihaiqalfaturahman@gmail.com

Abstrak

Analisis pendulum teredam yang dinyatakan dalam persamaan diferensial dianalisis menggunakan metode numerik. Metode numerik yang digunakan di antaranya metode *Euler*, *Runge-Kutte* orde ke-4, *Odeint* yang disediakan dalam bahasa *Python*, dan skema *Neural Network*. Semua jawaban yang diperoleh merupakan perkiraan atau prediksi yang mengandung kesalahan. *Error* dari metode-metode tersebut akan dibandingkan dengan solusi analitis dari kasus tersebut, yang dikenal sebagai *error global*. Metode *Odeint* dan *Runge-Kutta* orde ke-4t lebih akurat dibandingkan dengan metode lainnya. *Odeint* dibangun dari metode *Runge-Kutta* orde ke-4. Metode *Neural Network* memiliki akurasi yang lebih rendah dibandingkan *Runge-Kutta*, *error* yang diperoleh dari metode *Neural Network* masih dalam batas toleransi yang dapat diterima. Penelitian ini diperoleh luaran bahwa metode *Runge-Kutta* merupakan metode yang paling presisi. Namun dengan menggunakan metode *Neural Network* bisa mendapatkan jawaban yang lebih presisi apabila menggunakan *Epochs* yang tinggi.

Kata kunci: Analisis *Error*, Ayunan Teredam, *Runge-Kutta*, *Odeint*, *Neural Network*

Abstract

Analysis of the damped pendulum expressed in differential equations analyzed using numerical methods. Numerical methods are used to solve the differential equations in the case of a damped pendulum system. In this paper, we aim to solve the equation of dumped pendulum motion using some numerical. Four numerical methods are decomposed, each the Euler method, fourth-order Runge-Kutte, Odeint provided in Python, and Neural Network scheme. All answers obtained are approximations or predictions that include errors. The error of those methods will be compared with the analytical solution of the case, known as the global error. The Odeint and fourth-order Runge-Kutta methods are more accurate than the other methods. The Odeint is built from the Runge-Kutta fourth-order method. The Neural Network method has less accuracy than Runge-Kutta, but the error obtained is still within acceptable tolerance limits. This research obtained the output that the Runge-Kutta method is the most precise method. However, using the Neural Network method can get a more precise answer when using high Epochs.

Keywords : *Error Analysis, Damped Pendulum, Runge-Kutta, Odeint, Neural Network*

1. Pendahuluan

Ayunan teredam harmonik merupakan sistem dinamika gerak osilasi yang banyak dijumpai dalam kasus mekanika (Hamadi & Nugroho, 2015). Gerak harmonik teredam merupakan gerak bolak-balik suatu benda yang dipengaruhi gaya luar sehingga gerak osilasi semakin lama akan berhenti (Variastuti et al., 2019). Karakteristik gerak harmonik teredam biasanya dinyatakan dalam persamaan diferensial yang secara umum diselesaikan menurut analisis analitik. Ayunan teredam harmonik ini mempunyai sistem gerak yang kompleks berupa persamaan diferensial orde 2 maka akan sulit apabila diselesaikan dengan persamaan analitik, sehingga salah satu cara yang dapat digunakan dalam menyelesaikan persamaan diferensial ini dengan metode numerik.

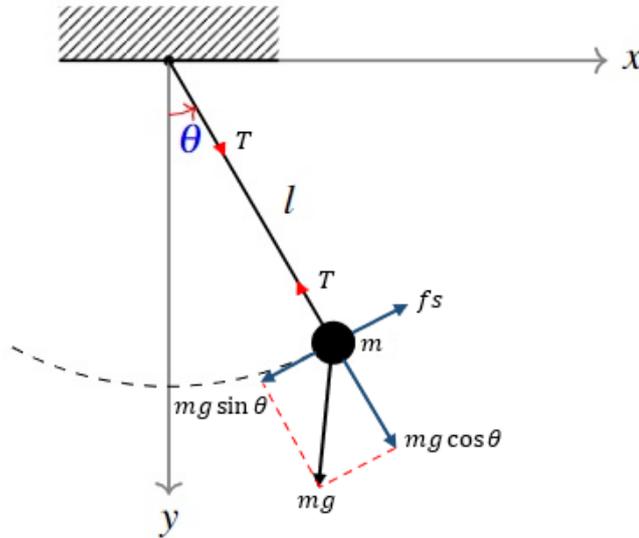
Beberapa metode yang dapat digunakan dalam penyelesaian analisis numerik adalah metode *Euler* (Fauzi, 2011), metode *Runge-Kutta* orde 4 (Chapra & Canale, 2015), *Odeint* (Ahnert & Mulansky, 2011)(Ghosh et al., 2021), dan *Neural Network* (Nagy et al., 2020). Metode *Euler* sering digunakan dalam menyelesaikan masalah persamaan diferensial orde 2. Dalam penyelesaian masalah persamaan diferensial orde 2 menggunakan persamaan *Lagrange* dengan persamaan yang meninjau pada energi kinetik dan energi potensial sehingga metode *Euler* merupakan metode yang paling sederhana (Dharmawan et al., 2014). Metode *Runge-Kutta* merupakan metode yang paling akurat dibandingkan dengan metode lainnya (Sari et al., 2021). Metode *Runge-Kutta* memiliki orde yang berbeda, dengan semakin tinggi orde yang digunakan maka derajat ketelitian semakin tinggi.

Odeint adalah sebuah sumber pustaka C++ yang memiliki kualitas tinggi (Ahnert & Mulansky, 2011). Metode *Odeint* digunakan untuk menyelesaikan berbagai macam algoritma numerik dari suatu masalah nilai awal pada persamaan diferensial biasa (ode) (Ahnert et al., 2014). *Neural Network* merupakan suatu sistem jaringan komputasi yang terinspirasi dari otak manusia dimana jaringan komputer yang terdiri dari elemen elemen untuk menyelesaikan berbagai masalah secara dinamis. Dengan menggunakan *Neural Network* akan menghasilkan solusi yang paling efisien untuk berbagai algoritma pembelajaran (Akthar, 2016).

Kekuatan metode numerik dalam penyelesaian persamaan diferensial dapat diukur dengan *global error*. Analisis *global error* sangat penting dalam menentukan seberapa baik metode numerik yang digunakan. Analisis *global error* dengan membandingkan metode numerik dengan jawaban analitik (Shampine & Watts, 1976). Tujuan dari penelitian ini adalah mencari *global error* dari empat metode diantaranya metode *Euler*, *Runge-Kutta* orde 4, *Odeint*, dan *Neural Network* dengan cara membandingkan hasil antara analitik dan numeriknya pada kasus ayunan teredam harmonis.

a. Sistem Ayunan Teredam

Ayunan sederhana merupakan salah satu dinamika osilasi yang banyak dijumpai dalam fisika. Pada system ayunan teredam terdapat pada koordinat cartesian (x, y) . Ayunan sederhana terdiri dari sebuah massa m yang terikat pada tali dengan panjang l berayun bebas dengan sudut θ di bidang vertikal yang mempunyai gaya gesek f_s yang sebanding dengan percepatan sudut ω dan tegangan tali berupa T , yang diilustrasikan pada gambar 1.



Gambar 1. Sistem Ayunan Teredam Harmonik

Persamaan gerak untuk system ini berupa persamaan diferensial orde dua yang dirumuskan menurut

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta - \frac{\mu}{ml}\omega \tag{1}$$

dengan μ adalah koefisien gerak, dengan g adalah pengaruh gravitasi.

b. Metode Euler

Metode numerik digunakan untuk menyelesaikan suatu permasalahan matematis dengan persamaan diferensial yang tidak dapat diselesaikan dengan metode analitik. Metode numerik sering digunakan untuk menyelesaikan persamaan *differential continue* menjadi diferensial diskret. Pendekatan numerik digunakan untuk menyelesaikan Persamaan Diferensial Biasa (PDB) orde satu dan orde dua dengan metode *Euler* dan *Runge-Kutta*.

Metode *Euler* merupakan metode yang diambil dari suku pertama deret *Taylor* (Rahmatullah et al., 2020). Metode ini cukup sederhana untuk menyelesaikan persamaan diferensial biasa. Metode *Euler* biasa digunakan dalam persamaan diferensial karena lebih mudah. Namun untuk hasil ketelitiannya masih kurang dikarenakan hanya dua suku perhiungan yang diambil. Analisis dengan metode ini dapat dilakukan dengan mudah karena akurasi yang kurang akurat, dengan itu rumus umum untuk pendekatan *Euler* sebagai berikut:

$$y_{x+h} = y_x + hy'_x \tag{2}$$

c. Runge-Kutta

Metode *Runge-Kutta* metode umum yang digunakan dalam penyelesaian persamaan diferensial biasa, karena Metode *Runge-Kutta* memiliki akurasi yang lebih tinggi dari pada metode *Euler* dan tidak memerlukan turunan dari fungsi. Metode *Runge-Kutta* menggunakan empat pendekatan. Metode *Runge-Kutta* terdapat beberapa tipe yang bergantung pada nilai n (orde) yang digunakan.

Metode *Runge-Kutta* yang akan ditinjau adalah metode *Runge-Kutta* orde ke-4. Apabila suatu persamaan diferensial $f' = dy/dx$, tentu dapat diperkirakan bahwa jawaban numerik dapat berupa $f(y, x)$ yaitu suatu fungsi yang dapat bergantung pada y atau x . Metode *Runge-*

Kutta ini dapat dikatakan sebagai pengembangan metode *Euler* dengan menambahkan beberapa suku tambahan pada satu grid yang dibuat. Suku-suku tersebut adalah

$$\begin{aligned} j_1 &= hf'(y, x) & j_3 &= hf' \left(y + \frac{j_2}{2}, x + \frac{h}{2} \right) \\ j_2 &= hf' \left(y + \frac{j_1}{2}, x + \frac{h}{2} \right) & j_4 &= hf'(y + j_3, x + h) \end{aligned}$$

selanjutnya, jawaban metode ini dirumuskan menurut

$$f_{n+h} = f_n + \frac{1}{6}(j_1 + 2j_2 + 2j_3 + j_4) \tag{3}$$

Apabila persamaan diferensial merupakan persamaan diferensial orde 2. Metode *Runge-Kutta* ini saling terkait, tidak seperti metode *Euler* yang dapat dipisah. Andaikan $g' = d^2y/dx^2$ dan $f' = g = dy/dx$, tentu jawaban f berupa fungsi yang dapat bergantung pada y , $g = f'$ atau x . Jawaban numerik akan diberikan menurut

$$f_{n+h} = f_n + \frac{1}{6}(j_1 + 2j_2 + 2j_3 + j_4) \tag{4}$$

$$g_{n+h} = g_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5}$$

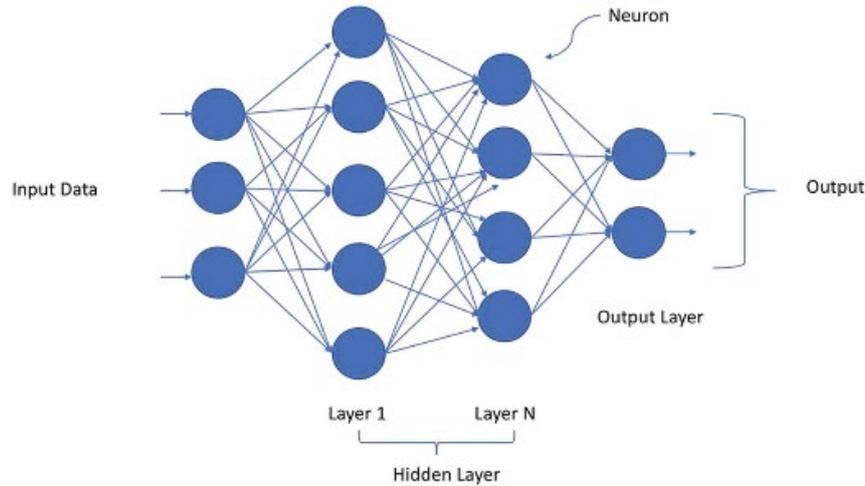
Dengan

$$\begin{aligned} j_1 &= hg(y, g, x) & k_1 &= hg'(y, g, x) \\ j_2 &= hg \left(y + \frac{j_1}{2}, g + \frac{k_1}{2}, x + \frac{h}{2} \right) & k_2 &= hg' \left(y + \frac{j_1}{2}, g + \frac{k_1}{2}, x + \frac{h}{2} \right) \\ j_3 &= hg \left(y + \frac{j_2}{2}, g + \frac{k_2}{2}, x + \frac{h}{2} \right) & k_3 &= hg' \left(y + \frac{j_2}{2}, g + \frac{k_2}{2}, x + \frac{h}{2} \right) \\ j_4 &= hg(y + j_3, g + k_3, x + h) & k_4 &= hg'(y + j_3, g + k_3, x + h) \end{aligned}$$

Metode *Runge-Kutta* orde ke-4 mempunyai ketelitian lebih tinggi dibanding dengan *Runge-Kutta* Orde sebelumnya. Oleh karena itu, semakin tinggi orde yang digunakan maka akurasi yang didapatkan akan semakin tinggi. Metode *Runge-Kutta* orde ke-4 mudah deprogram dan memiliki kecil kesalahan pemotongan dan juga kecil kesalahan pembulatan (Triadmidjo, 2002).

d. Neural Network

Neural Network merupakan sistem jaringan pemrosesan yang terdiri dari beberapa *neuron* yang saling terkait. *Neuron* berfungsi untuk memecahkan masalah dari sebuah data. *Neural Network* sendiri digambarkan sebagai kemampuan menganalisis dan mengolah data pelatihan yang memiliki kemiripan dengan otak manusia. *Neural Network* digunakan untuk mendapatkan solusi numerik dari permasalahan persamaan diferensial dan masalah nilai awal di berbagai bidang matematika, fisika, dan ilmu teknik.



Dari gambar di atas dapat dijelaskan bahwa *Neural Network* terdiri dari tiga jenis lapisan yaitu *Input layer*, *Hidden layer* dan *output layer*. *Input layer* merupakan lapisan yang berkomunikasi dengan lingkungan eksternal yang menyajikan pola ke *Neural Network* (Karsoliya, 2012). *Input layer* memiliki pola yang kemudian akan diolah oleh *neuron-neuron* yang berada pada *hidden layer* yang kemudian memberikan *output* ke lapisan berikutnya dan seterusnya, sehingga menghasilkan *output layer* dengan pola lain. *Input layer* harus mewakili kondisi yang dilatih untuk *Neural Network*. Setiap *neuron input* harus mewakili beberapa variabel independen yang memiliki pengaruh terhadap *output* jaringan saraf. *Hidden layer* pada *Neural Network* merupakan kumpulan *neuron* yang memiliki fungsi aktivasi yang diterapkan menghubungkan antara layer *input* dan layer *output* (Akthar, 2016).

Fungsi aktivasi adalah parameter yang berguna untuk menentukan fungsi matematika yang akan digunakan suatu model *machine learning* untuk mengelola data *input* sehingga bisa didapatkan nilai *output* (Dananjaya et al., 2022). Setiap lapisan pada *hidden layer* memiliki satu atau banyak *neuron*, dan masing-masing *neuron* akan menghitung sebuah fungsi kecil (misalnya, fungsi aktivasi). Setiap *neuron* yang tersimpan pada *hidden layer* harus dihitung. Jika datanya dipisahkan secara linier maka tidak menggunakan *hidden layer* untuk fungsi aktivasi. Pada kasus batas keputusan dan akurasi yang berubah-ubah dengan fungsi aktivasi rasional maka menggunakan dua atau tiga *hidden layer* (Wibawa & Dendi Maysanjaya, 2018). Jika *neuron* yang berada pada *hidden layer* sedikit dibandingkan dengan kompleksitas maka akan terjadi "*Underfitting*", dan sebaliknya apabila jumlah *neuron* yang berada pada *hidden layer* lebih banyak maka akan terjadi "*Overfitting*". *Underfitting* terjadi ketika *neuron* pada *hidden layer* yang berguna untuk memadai mendeteksi sinyal dalam kumpulan data terlalu sedikit.

Output layer neuron network merupakan lapisan yang menyajikan pola dari *hidden layer* ke lingkungan eksternal. Jumlah *neuron* yang keluar harus terkait secara langsung dengan pekerjaan yang dilakukann oleh *Neural Network*. Dalam menentukan jumlah *neuron* yang akan digunakan pada *output layer* maka hal yang harus dilakukan yaitu mempertimbangkan tujuan penggunaan *Neural Network* tiruan. Apabila akan mengklasifikasikan item ke dalam kelompok menggunakan *Neural Network*, maka setiap kelompok yang ditugaskan ke dalam *input* memiliki satu *neuron* keluaran. Jika *Neural Network* melakukan pengurangan noise pada sinyal, maka jumlah *neuron* akan sama dengan jumlah *neuron output*. Dalam *Neural Network* pola-pola pada *output layer* memiliki pola yang sama dengan *input* (Gupta, 2013).

Koneksi antara dua *neuron* dari lapisan yang berurutan akan memiliki bobot yang terkait. Bobot mendefinisikan pengaruh *input* ke *output* untuk *neuron* berikutnya dan akhirnya untuk *output* akhir secara keseluruhan. Pada *Neural Network*, bobot awal akan bersifat acak selama pelatihan model, tetapi bobot ini diperbarui secara iteratif untuk belajar memprediksi *output* yang benar. Dengan menguraikan jaringan dapat didefinisikan beberapa blok bangunan logis

seperti *neuron*, lapisan, bobot, *input*, *output*, fungsi aktivasi di dalam *neuron* untuk menghitung proses pembelajaran, dan sebagainya.

Metode *Neural Network* terdapat komponen *loss function* yang digunakan untuk mengevaluasi kinerja model dan memandu proses optimasi. Fungsi *loss function* untuk menghitung kuantitas kerugian pada *Neural Network*. Salah satu *loss function* yang umum digunakan pada *Neural Network* adalah kerugian MSE (*Mean Squared Error*). MSE atau kerugian kuadrat merupakan rata-rata perbedaan nilai actual dan nilai yang diprediksi.

Penyelesaian persamaan diferensial menggunakan *Neural Network* dengan merepresentasikan solusi *Neural Network* dan melatih jaringan yang dihasilkan untuk memenuhi kondisi yang diperlukan oleh persamaan diferensial. Persamaan umum *Neural Network* untuk mempresentasikan sebuah solusi persamaan diferensial yang diberikan menurut:

$$y \approx N(x) \tag{6}$$

Loss function

$$L = L_f + L_u \tag{7}$$

dengan

$$L_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\frac{dN(x)}{dx} \Big|_{x=x_i} - f(x_i) \right)^2 \tag{8}$$

dan

$$L_u = \frac{1}{N_u} \sum_{i=1}^{N_u} \left(\frac{dN(x)}{dx} N(x_i) - y(x_i) \right)^2 \tag{9}$$

ada *Neural Network* terdapat *Epochs* yang ditentukan berdasarkan *hit n trail* pada masalah rumit dalam *Neural Network*. Satu *Epochs* digunakan dalam satu siklus dengan semua data pelatihan. Dalam satu lintasan terdiri dari lintasan maju dan lintasan mundur. Satu *Epochs* terdiri dari satu atau lebih batch. Iterasi merupakan jumlah batch yang dibutuhkan untuk menyelesaikan satu *Epochs*, dengan contoh 1000 batch dapat diselesaikan satu *Epochs* dengan menggunakan satu kali iterasi (Yuliani et al., 2020).

e. Metode Analitik

Metode Analitik merupakan persamaan diferensial yang melibatkan fungsi dan turunan orde 2, tanpa menurunkan orde yang lebih tinggi dari fungsi tersebut. Persamaan umum analitik dapat ditulis.

$$\frac{d^2y}{dx^2} + b \frac{dy}{dx} + cy = 0 \tag{10}$$

Jawaban umum dari persamaan (6) ada tiga macam, masing-masing:

Jika $b^2 > 4c$, maka

$$y = c_1 e^{\alpha x} + c_2 e^{\beta x} \tag{11}$$

Jika $b^2 = 4c$,

$$y = (A\alpha + B e^{\alpha x}) \tag{12}$$

Jika $b^2 < 4c$

$$y = e^{\alpha x} (A e^{i\beta x} + B e^{-i\beta x}) \tag{13}$$

dengan nilai $\alpha = (-b + \sqrt{b^2 - 4c})/2$ dan $\beta = (-b - \sqrt{b^2 - 4c})/2$

2. Metode

Ayunan teredam merupakan suatu sistem mekanika dengan menggunakan penyelesaian persamaan diferensial orde 2. Dari persamaan orde 2 menggunakan metode analitik didapatkan solusi berupa sudut θ dan kecepatan sudu ω dengan persamaan (5), (6), dan (7). Dalam menentukan solusi dari permasalahan pada θ dan ω dapat diselesaikan menggunakan investigasi numerik berupa metode *Euler* dan metode *Runge-Kutta*. Selain menggunakan dua metode tersebut permasalahan tersebut dapat diselesaikan menggunakan orde *solver* pada *Odeint* yang diselesaikan menggunakan *Python*. Solusi persamaan orde 2 dapat juga diselesaikan menggunakan *Neural Network* yang akan digunakan untuk memprediksi solusi dari θ dan ω .

Solusi untuk memperoleh nilai dari θ dan ω mempunyai lima macam penyelesaian, maka persamaan yang dari solusi diferensial tersebut dapat membandingkan beberapa metode numerik terhadap metode analitik. Dari perbandingan solusi diferensial tersebut solusi analitik merupakan solusi yang paling valid dalam penyelesaian persamaan diferensial tersebut, sehingga solusi analitik dijadikan sebagai pembanding dari metode yang ada. Dengan membandingkan solusi analitik dengan empat metode berupa metode *Euler*, *Runge-Kutta*, *Odeint*, dan *Neural Network*, maka dapat ditentukan *error* global yang terjadi empat metode dalam persamaan diferensial ayunan teredam harmonis. Untuk mendapatkan perbandingan yang akurat secara khusus metode *Neural Network* yang berupa prediksi akan ditinjau dari perubahan *Epochs* yang digunakan.

3. Hasil dan Pembahasan

Dari penyelesaian persamaan diferensial ayunan teredam harmonis diperoleh jawaban secara analitik dan numerik dengan parameter pada tabel 1.

Tabel 1. Parameter

No.	Parameter	Dimensi	Satuan
1.	g	Percepatan gravitasi	9.8 ms^{-2}
2.	l	Panjang tali	1 m
3.	m	Massa benda	0.05 kg
4.	μ	Koefisien gesek	0.1 kgm/s
5.	t_0	Waktu awal	0 detik
6.	t_{mak}	Waktu akhir	10 detik
7.	h	Lebar grid	0.01

dari parameter tabel 1 dapat diperoleh solusi persamaan diferensial secara analitik dan numerik. Dalam mendapatkan solusi analitik dari persamaan diferensial ayunan teredam, dengan persamaan awal

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} - \frac{\mu}{ml}\omega \tag{14}$$

Dengan kasus ayunan teredam maka sudut yang digunakan harus sudut kecil. Dengan menggunakan sudut kecil kita memperoleh solusi analitik pada persamaan (5). Dari persamaan tersebut dihasilkan penyelesaian pada persamaan gerak ayunan teredam secara analitik menurut

$$\theta = \theta_0 e^{\alpha t} \cos \beta t \tag{15}$$

$$\omega = \theta_0 (\alpha e^{\alpha t} \sin \beta t - \beta e^{\alpha t} \cos \beta t) \tag{16}$$

Selain mendapatkan jawaban analitik dari parameter tersebut kasus ayunan teredam juga dapat diselesaikan secara numerik. Penyelesaian secara numerik pada kasus ayunan teredam didapatkan persamaan gerak berupa

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\theta - \frac{\mu}{ml}\omega \tag{17}$$

Dari persamaan gerak (2) dapat diselesaikan menggunakan metode *Euler* dengan persamaan yang didapat

$$\theta_{t+h} = \theta_t + h\dot{\theta} \tag{18}$$

$$\dot{\theta}_{t+h} = \dot{\theta}_t + h\ddot{\theta} \tag{19}$$

Selain dengan metode *Euler* kasus ayunan teredam juga dapat diselesaikan menggunakan metode *Runge-Kutta*. Dari metode *Runge-Kutta* mendapatkan solusi numerik berupa persamaan berikut

$$\theta_{n+1} = \theta_n + \frac{1}{6}(j_1 + 2j_2 + 2j_3 + j_4) \tag{20}$$

$$\omega_{n+1} = \omega_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{21}$$

Metode *Odeint* merupakan suatu solusi dengan pemanfaatan program python untuk menyelesaikan persamaan diferensial biasa. Metode *Odeint* dapat digunakan dalam menyelesaikan persamaan gerak ayunan teredam. Metode *Odeint* merupakan penerapan *Runge-Kutta* dalam pemrograman. *Odeint* tidak dapat diterapkan untuk semua kasus, ada beberapa kasus hanya dapat diselesaikan menggunakan metode *Runge-Kutta*. Dalam kasus ayunan teredam ini dapat diselesaikan menggunakan persamaan *Odeint* dengan fungsi

$$(u[1], -\frac{g}{l}(u[0]) - \frac{\mu}{ml}u[1]) \tag{22}$$

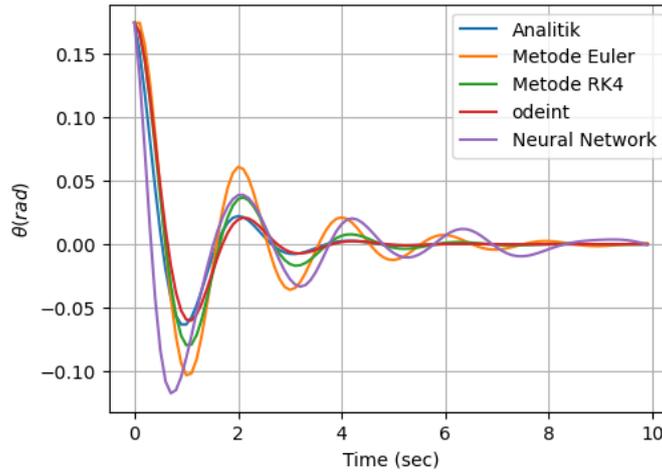
dengan (u) merupakan vektor yang menggambarkan keadaan sistem. Sedangkan $u[0]$ merupakan sudut (θ) bandul pada posisi vertikal sedangkan $u[1]$ merupakan kecepatan sudut (ω).

Penyelesaian persamaan diferensial ayunan teredam dengan metode *Neural Network* dapat diselesaikan dengan mengimplementasikan persamaan 6 dengan perumusan sebagai berikut

$$\theta \approx N(t) \tag{23}$$

Dengan (θ) merupakan sudut dari ayunan dan (t) sebagai satuan waktu.

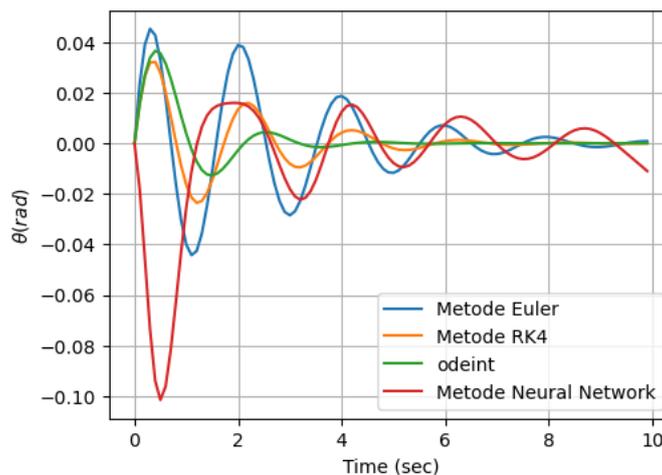
Seluruh metode yang telah digunakan untuk menyelesaikan persamaan numerik akan di dapatkan tingkat ketelitian dengan solusi analitik sebagai solusi yang paling valid. Hasil dari lima metode dapat ditampilkan berupa grafik pada gambar 3.



Gambar 3. Grafik θ Terhadap t

Dari grafik pada gambar 3 dapat diketahui ketelitian setiap metode. Dengan metode *Odeint* yang memiliki ketelitian yang hampir mendekati metode analitik (Nagy et al., 2020). Hal ini dibuktikan pada grafik metode *Odeint* memiliki grafik yang paling mendekati grafik analitik, dibandingkan dengan metode lainnya. Metode *Runge-Kutta* mendekati grafik metode *Odeint*, karena *Odeint* dan *Runge-Kutta* memiliki pendekatan yang sama dalam memecahkan persamaan diferensial biasa. Pada grafik metode *Euler* mempunyai ketelitian yang kurang apabila dibandingkan dengan analitik. Metode *Euler* mempunyai ketelitian kurang akurat karena dalam menyelesaikan kasus persamaan diferensial yang kompleks akan mendapatkan hasil yang kurang akurat, namun metode ini merupakan metode yang paling sederhana dan mudah dipahami. Sedangkan pada metode *Neural Network* mempunyai hasil grafik yang paling jauh dengan grafik analitik. Hal ini dikarenakan pada *Neural Network* merupakan sebuah prediksi yang dipengaruhi *Epochs*, dengan semakin besar *Epochs* semakin teliti solusi yang didapatkan seperti dalam penelitian.

Analisis global *error* digunakan untuk membuktikan metode numerik, yang dibandingkan dengan metode analitik. Dimana analitik merupakan solusi paling valid sehingga dijadikan acuan dari solusi numerik. Hasil penyelesaian secara numerik dapat ditampilkan dengan grafik pada gambar 4



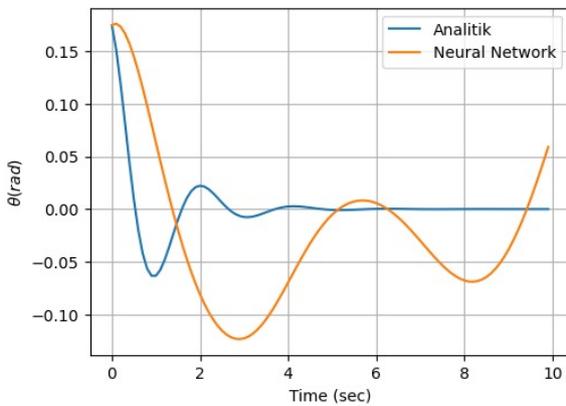
Gambar 4. Grafik Global *Error*

Dengan penyelesaian secara analitik dan numerik maka akan didapatkan hasil stabilitas dari nilai *error* pada hasil penyelesaian tersebut. Dengan menggunakan global *error* membandingkan antara 4 metode dengan metode analitik, maka didapatkan hasil nilai Maksimum *Error* sebagai berikut.

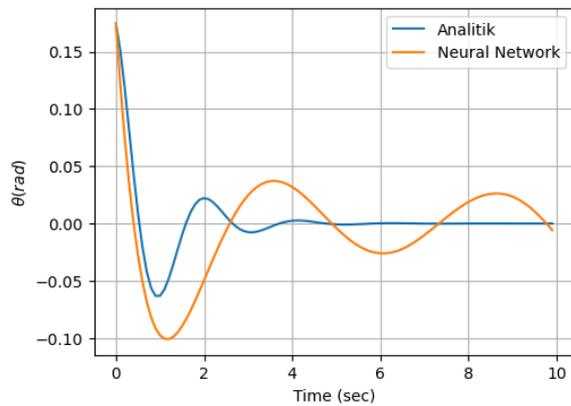
Tabel 2. Analisis *Error Global*

Step	t = 20	t = 40	t = 60	t = 80	t = 100	Max Error
<i>Euler</i>	0.0208	0.0019	0.0001	8.18e-06	4.02e-06	0.0452
<i>Runga-kutta</i>	0.0304	0.0048	0.0005	2.83e-06	2.80e-05	0.0321
<i>Odeint</i>	0.0155	0.0009	5.12e-05	2.98e-05	6.64e-06	0.0365
<i>Neural Network</i>	3.14e-02	6.78e-03	1.04e-03	1.51e-03	1.02e-02	0.1046

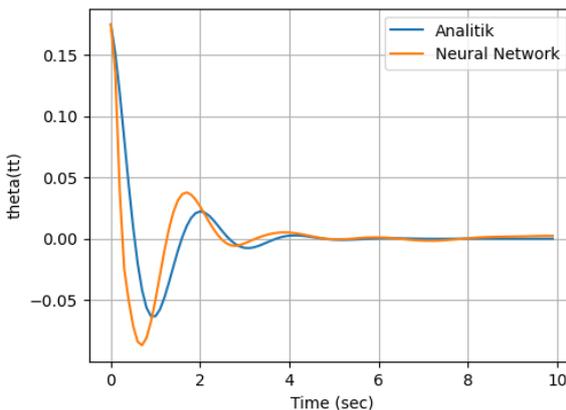
Untuk membuktikan metode *Neural Network* itu akurat dapat ditinjau dengan melakukan perubahan *Epochs* algoritma numerik pada persamaan diferensial ayunan teredam. Dengan meningkatkan jumlah *Epochs* dapat berpotensi mengurangi kesalahan atau meningkatkan akurasi, tetapi juga membutuhkan proses pelatihan yang lama. Dalam kasus ini pada solusi menggunakan *Neural Network* apabila semakin besar *Epochs* yang digunakan maka semakin akurat solusi yang dihasilkan dibuktikan dengan hasil grafik pada gambar 5,6,7, dan 8 seperti dalam penelitian (Mall & Chakraverty, 2016).



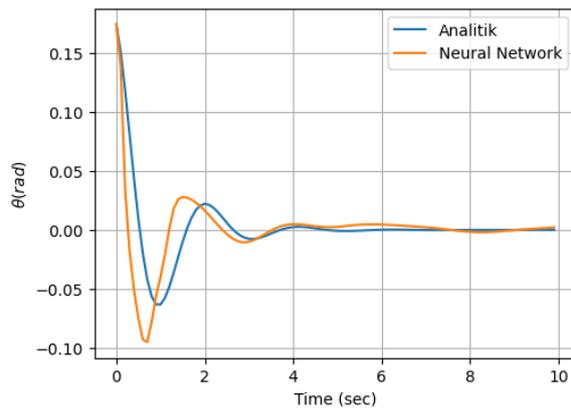
Gambar 5. Grafik dengan *Epoch 5*



Gambar 6. Grafik dengan *Epoch 100*



Gambar 7. Grafik dengan *Epoch 10000*



Gambar 8. Grafik dengan *Epoch 15000*

Pada gambar 5 dan 6 terlihat perbedaan grafik yang cukup signifikan antara jumlah *epoch* 5 dan *epoch* 100. Kemudian dilakukan percobaan kembali dengan jumlah *epoch* semakin besar. Pada gambar 7 dengan jumlah *epoch* 10000 dan gambar 8 dengan jumlah *epoch* 15000 grafik mulai stabil. Perubahan grafik tidak terlihat signifikan. Berdasarkan hasil grafik pada gambar 5, 6, 7, dan 8 semakin besar jumlah *epoch* maka grafik hasil dari persamaan

diferensial ayunan teredam dengan menggunakan metode *Neural Network* akan mendapatkan hasil akurasi yang lebih tinggi.

4. Kesimpulan dan Saran

Sistem pendulum teredam dapat diselesaikan menggunakan beberapa metode seperti metode analitik, metode *Euler*, metode *Runge-Kutta*, metode *Odeint*, dan metode *Neural Network*. Dari lima metode yang digunakan dalam menyelesaikan masalah metode *Runge-Kutta* merupakan metode yang paling presisi. Namun dengan menggunakan metode *Neural Network* bisa mendapatkan jawaban yang presisi apabila menggunakan *Epochs* yang tinggi.

Daftar Pustaka

- Ahnert, K., Demidov, D., & Mulansky, M. (2014). *Solving Ordinary Differential Equations on GPUs*. https://doi.org/10.1007/978-3-319-06548-9__7
- Ahnert, K., & Mulansky, M. (2011). *Odeint - Solving ordinary differential equations in C++*. *AIP Conference Proceedings*, 1389, 1586–1589. <https://doi.org/10.1063/1.3637934>
- Akthar, S. (2016). *A Study on Neural Network Architectures* (Vol. 7, Issue 9). www.iiste.org
- Chapra, S. C., & Canale, R. P. (2015). *Numerical methods for engineers*.
- Dananjaya, R. H., Sutrisno, S., & Fitriady, S. (2022). PENERAPAN ARTIFICIAL NEURAL NETWORK (ANN) DALAM MEMREDIKSI KAPASITAS DUKUNG FONDASI TIANG. *Matriks Teknik Sipil*, 10(4), 419. <https://doi.org/10.20961/mateksi.v10i4.65034>
- Dharmawan, A., Simanungkalit, Y. Y., Megawati, N. Y., Kunci-Pemodelan, K., & Lagrange, E. (2014). Pemodelan Sistem Kendali PID pada Quadcopter dengan Metode *Euler Lagrange*. *IJEIS*, 4(1), 13–24.
- Fauzi, A. (2011). ANALISIS GERAK HARMONIK TEREDAM (DAMPED HARMONIC MOTION) DENGAN SPREADSHEET EXCEL. 7(2), 318–322.
- Ghosh, S., Birrell, P. J., & De Angelis, D. (2021). *Variational inference for nonlinear ordinary differential equations*.
- Gupta, N. (2013). *Network and Complex Systems Artificial Neural Network*. 3(1). www.iiste.org
- Hamadi, H., & Nugroho, F. (2015). Penyelesaian Numerik Model Ayunan Terpaksa Menggunakan Metode Exponential Time Differencing (ETD) dan Karakteristik Dinamika. *Jurnal Materi Dan Pembelajaran Fisika (JMPPF)*, 5, 56–64.
- Karsoliya, S. (2012). Approximating Number of Hidden layer *neurons* in Multiple Hidden Layer BPNN Architecture. *International Journal of Engineering Trends and Technology*. <http://www.internationaljournalsrg.org>
- Mall, S., & Chakraverty, S. (2016). Application of Legendre *Neural Network* for solving ordinary differential equations. *Applied Soft Computing Journal*, 43, 347–356. <https://doi.org/10.1016/j.asoc.2015.10.069>
- Moolayil, J. (2018). Learn Keras for Deep *Neural Networks*: A Fast-Track Approach to Modern Deep Learning with *Python*. In *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. Apress Media LLC. <https://doi.org/10.1007/978-1-4842-4240-7>
- Nagy, D., Plavec, L., & Hegedűs, F. (2020). *Solving large number of non-stiff, low-dimensional ordinary differential equation systems on GPUs and CPUs: performance comparisons of MPGOS, ODEINT and DifferentialEquations.jl*. <http://arxiv.org/abs/2011.01740>
- Rahmatullah, S., Arman, Y., Prodi Fisika FMIPA Universitas Tanjungpura, A., Hadari Nawawi Pontianak Prodi Ilmu Kelautan FMIPA Universitas Tanjungpura, J., & Hadari Nawawi

- Pontianak, J. (2020). *Simulasi Gerak Osilasi Model Pegas Bergandeng Menggunakan Metode Runge-Kutta*. 8(3), 180–184.
- Sari, U., Indonesia, M., Pandia, W., & Sitepu, I. (2021). PENENTUAN GALAT PERSAMAAN DIFERENSIAL BIASA ORDE 1 DENGAN METODE NUMERIK. *Jurnal Mutiara Pendidikan Indonesia* , 6(1). <https://doi.org/10.51544/mutiara%20pendidik.v6i1.1907>
- Shampine, L. F., & Watts, H. A. (1976). Global Error Estimation for Ordinary Differential Equations. In *ACM Trans. Math. Software* ~ (Vol. 2).
- Triadmidjo, B. (2002). *Metode Numerik Dilengkapi Dengan Program Komputasi*.
- Variastuti, M., Ali, F., Viridi, S., Satya Perkasa, Y., Sarjana Fisika, P., Keilmuan Fisika Nuklir dan Komputasi, K., & Keilmuan Fisika Nuklir Teori, K. (2019). *Simulasi Gerak Pendulum Teredam dengan Metode Euler Berbasis Javascript*.
- Wibawa, M. S., & Dendi Maysanjaya, M. (2018). MULTI LAYER PERCEPTRON DAN PRINCIPAL COMPONENT ANALYSIS UNTUK DIAGNOSA KANKER PAYUDARA. *Jurnal Nasional Pendidikan Teknik Informatika*, 7(1), 90–99.
- Yuliani, E., Aini, A. N., & Khasanah, C. U. (2020). Perbandingan Jumlah *Epoch* Dan Steps Per *Epoch* Pada Convolutional *Neural Network* Untuk Meningkatkan Akurasi Dalam Klasifikasi Gambar. *Jurnal Informa : Jurnal Penelitian Dan Pengabdian Masyarakat*, 5(3), 23–27. <https://doi.org/10.46808/informa.v5i3.140>