Limits: Journal of Mathematics and Its Applications
E-ISSN: 2579-8936
P-ISSN: 1829-605X
Vol. 22, No. 2, Juli 2025, 73-84
DOI: https://doi.org/10.12962/limits.v22i2.3467

Perbandingan Metode Euler - Estimasi Galat *Neural Network* dan Metode Runge Kutta Orde 4 dalam Menyelesaikan Persamaan Diferensial Biasa Linear

Ratna Herdiana [1] *, Clarine Alfiani [2*]
[1,2]Departemen Matematika, Fakultas Sains dan Matematika,
Universitas Diponegoro, Semarang Indonesia
ratnaherdiana@lecturer.undip.ac.id
clarine.alfiani07@gmail.com

Diajukan: 12 Oktober 2024, Diperbaiki: 13 Juli 2025, Diterima: 14 Juli 2025

Abstrak

Persamaan diferensial biasa linear merupakan jenis persamaan diferensial yang umumnya mudah diselesaikan secara analitik ketika fungsi pada integral parsial memiliki bentuk sederhana. Namun, saat fungsi tersebut merupakan fungsi yang sulit maka membutuhkan metode lain seperti metode numerik dan metode yang diadaptasi dari *neural network* karena metode analitik hanya dapat digunakan saat permasalahan memiliki tafsiran geometri yang sederhana. Penelitian ini melibatkan metode Euler yang dilanjutkan dengan estimasi galat menggunakan *neural network* dan metode Runge-Kutta Orde-4 sebagai pembanding. Perbandingan dilakukan dengan menyelesaikan empat persamaan yang selanjutnya dilakukan analisis terhadap hasil dan galat pada masing-masing metode berdasarkan grafik yang dihasilkan dan kriteria MAPE. Hasil penelitian berdasarkan grafik menunjukkan bahwa galat yang dihasilkan oleh metode dengan estimasi galat menggunakan *neural network* lebih stabil daripada metode Runge Kutta Orde-4. Selain itu, berdasarkan hasil perhitungan dengan kriteria MAPE, metode estimasi galat menggunakan *neural network* menghasilkan tingkat akurasi dalam kategori sangat tinggi, sedangkan untuk metode Runge-Kutta Orde 4 menghasilkan tingkat akurasi yang berada dalam dua kategori, yaitu kategori sangat tinggi dan wajar.

Kata Kunci: Persamaan Diferensial Biasa, Metode Estimasi Galat Menggunakan *Neural Network,* Metode Runge Kutta Orde 4

Abstract

Linear ordinary differential equations are a type of differential equation that is generally easy to solve analytically when the function on a partial integral has a simple form. However, when the function is a difficult function, it requires other methods such as numerical methods and methods adapted from neural networks because analytical methods can only be used when the problem has a simple geometric interpretation. This study involves the Euler method followed by error estimation using neural networks and the Runge-Kutta Orde-4 method as a comparison. The comparison was carried out by solving four equations which were then analyzed for the results and errors in each method based on the graphs generated and the MAPE criteria. The results of the study based on graphs show that the error generated by the method with error estimation using neural networks is more stable than the 4^{th} Order Runge-Kutta method. In addition, based on the results of calculations with the MAPE criteria, the error estimation method using neural networks produces a very high level of accuracy in the category, while the 4^{th} Order Runge-Kutta method produces a level of accuracy in two categories, namely the very high and reasonable categories.

Keywords*: Ordinary Differential Equation, Error Estimation using Neural Network Method, 4^{th} Order Runge-Kutta Method*

1 Pendahuluan

Perubahan perilaku suatu fenomena dapat dijelaskan secara matematis melalui persamaan diferensial. Persamaan diferensial berperan membantu dalam mempelajari perilaku suatu fenomena karena di dunia nyata hanya sebatas mengungkapkan perubahannya. Pada artikel ini, pembahasan difokuskan pada jenis persamaan diferensial biasa linear.

Definisi 1 [1] *Suatu persamaan diferensial orde pertama dikatakan linear jika dapat dituliskan dalam bentuk:*

$$y'(x) + p(x)y(x) = r(x) \tag{1}$$

dengan p *dan* r *adalah fungsi-fungsi dari x yang diketahui.*

Persamaan diferensial (1) dapat ditulis dalam bentuk umum

$$y'(x) = f(x, y). \tag{2}$$

Secara umum, dalam menyelesaikan persamaan diferensial (1) cukup dengan diselesaikan secara analitik. Namun, pada banyak kasus, fungsi ruas kanan $r(x)$ memiliki bentuk yang rumit sehingga sulit untuk diintegralkan secara parsial. Hal ini mengakibatkan diperlukan metode lain dalam menyelesaikan persamaan diferensial tersebut. Metode lain yang bisa digunakan adalah metode numerik dan metode yang diadaptasi dari *neural netwok.* Metode numerik menghasilkan solusi hampiran atau pendekatan, yaitu solusi yang menghampiri solusi sebenarnya. Dalam menyelesaikan suatu persamaan diferensial secara numerik, diperlukan diskritisasi domain menjadi n bagian, selanjutnya solusi hampiran pada titik-titik diskritisasi (titik ke-i, $i = 1, 2, \dots, n$) tersebut merupakan solusi numerik dari persamaan diferensial. Metode numerik dapat digunakan dalam menyelesaikan persamaan diferensial melalui perhitungan solusi di titik $x_{i+1} = x_i + h$ dengan h merupakan ukuran langkah dari setiap iterasi.

Hal ini berbeda dengan penyelesaian persamaan diferensial dengan *neural networks (NN)* yang secara simultan mencari solusi pada setiap titik [2]. *Neural networks* dapat digunakan untuk memetakan secara nonlinear suatu masukan terhadap keluaran berdasarkan sistem neuron sehingga sangat sesuai untuk menganalisis suatu permasalahan persamaan diferensial yang memiliki nilai awal maupun syarat batas dibandingkan solusi secara analitik yang tidak dapat dihitung dengan mudah. Solusi persamaan diferensial menggunakan *NN* terbagi menjadi dua bagian, pertama memenuhi kondisi nilai awal atau syarat batas kemudian bagian kedua merupakan hasil dari *neural network* tepatnya *feedforward neural network* [3][4].

Pada penelitian ini, *neural network* (*NN*) diterapkan untuk mengkoreksi galat yang dihasilkan dari penyelesaian persamaan diferensial biasa menggunakan skema numerik yang umum dipakai, berbeda dengan metode numerik dimana galat solusi numerik langsung diperoleh bersama dengan solusi numeriknya. Pada metode estimasi-galat NN, aproksimasi solusi persamaan diferensial biasa lebih dahulu dihitung menggunakan metode numerik kemudian dilanjutkan dengan koreksi galat menggunakan algoritma NN. Metode numerik yang digunakan untuk mendapatkan solusi sementara dipilh berorde rendah agar komputasinya mudah dan murah, seperti metode Euler orde 1 atau 2. Skema seperti ini telah banyak dilakukan dan diketahui mempunyai akurasi dan orde konvergensi yang tinggi tanpa mengurangi stabilitas [5][6].

Beberapa peneliti telah melakukan penelitian terkait penyelesaian persamaan diferensial yang mengadaptasi teknik *neural networks,* seperti penggunaan algoritma *shooting neural networks,* yaitu gabungan dari metode *shooting* dan algoritma *neural networks* yang digunakan dalam menyelesaikan masalah syarat batas dalam persamaan diferensial biasa orde dua [7], penggunaan metode estimasi *error* dengan *neural network* untuk menyelesaikan persamaan diferensial biasa [2], penggunaan metode *feedforward neural networks* dalam menyelesaikan persamaan diferensial biasa dan parsial [8]. Kemudian Jiang dan Xuan [6] membandingkan beberapa *hyperparameter* dan *optimizer* berbeda menggunakan metode *variable control* yang diaplikasikan pada persamaan diferensial orde dua, selanjutnya *polynomial neural network* yang berkorespondensi dengan system persamaan diferensial telah dikenalkan pada [9] dimana dibahas proses iterasi pembentukan solusi umum untuk system persamaan difernsial biasa yang tak stasioner dalam bentuk polynomial. Selain itu, telah dilakukan penelitian tentang perbandingan penyelesaian persamaan diferensial biasa antara metode numerik (Euler, Heun, dan Runge Kutta) dengan *neural networks* metode *backpropagation* yang menghasilkan bahwa penggunakan metode *backpropagation* memiliki tingkat akurasi yang lebih tinggi (galat yang lebih kecil) dibandingkan metode numerik lain [10].

Tujuan penelitian ini adalah untuk menilai efektifitas dari hasil implementasi algoritma skema estimasi-galat NN dibandingkan dengan mentode numerik dengan akurasi tinggi khususnya RK4 dalam menyelesaikan beberapa masalah nilai awal persamaan diferensial linier biasa yang diketahui solusi analitiknya.

2 Metode Penelitian

2.1 Metode Estimasi Galat Menggunakan *Neural Network*

Sesuai namanya, metode estimasi galat menggunakan *neural network* lebih berfokus pada perhitungan galat dengan *neural network.* Metode ini memerlukan solusi numerik sementara, $\hat{y}(\boldsymbol{x}), \boldsymbol{x} = (x_1, x_2, \dots, x_n)$, hasil dari perhitungan metode numerik orde rendah, yaitu metode Euler orde 1 yang dilanjutkan dengan melakukan estimasi galat menggunakan *neural network* [2]. Solusi dari persamaan diferensial biasa menggunakan metode estimasi galat dengan *neural network* dapat ditulis:

$$y(\boldsymbol{x}, w) = \hat{y}(\boldsymbol{x}) + F\big(\boldsymbol{x}, N(\boldsymbol{x}, w)\big) \tag{3}$$

dimana $F\big(\boldsymbol{x}, N(\boldsymbol{x}, w)\big)$ merupakan galat yang diestimasi dengan *neural network*.

Diketahui Teorema Taylor sebagai berikut:

Teorema Taylor [11] Misalkan f suatu fungsi yang terdiferensial sebanyak $(p+1)$ kali, yaitu $f^{(p+1)}$ pada interval terbuka β yang memuat titik x_0 dan x, maka

$$f(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + .. + \frac{(x-x_0)^p}{p!} f^p(x_0) + R_p(x) \tag{4}$$

dengan $R_p(x)$ merupakan suku sisa (galat) ke-n yang dinyatakan oleh

$$R_p(x) = \frac{f^{(p+1)}(c)}{(p+1)!} (x - x_0)^{p+1} \tag{5}$$

untuk sebarang bilangan c antara titik x_0 dan x.

Misal terdapat fungsi $f(x)$ yang dihampiri oleh deret Taylor di $x_{i+1} = x_i + h$, dengan $i = 0,1,2, ...$ maka nilai hampiran yang dihasilkan oleh $f(x_{i+1})$ disekitar x_i adalah

$$f(x_{i+1}) = f(x_i) + \frac{(x_{i+1}-x_i)}{1!} f'(x_i) + \frac{(x_{i+1}-x_1)^2}{2!} f''(x_i) + \cdots + \frac{(x_{i+1}-x_i)^p}{p!} f^{(p)}(x_i) + R_p(x_{i+1})$$

$$f(x_{i+1}) = f(x_i) + \frac{h}{1!} f'(x_i) + \frac{h^2}{2!} f''(x_i) + .. + \frac{h^p}{p!} f^{(p)}(x_i) + R_p(x_{i+1}) \tag{6}$$

dengan

$$R_p(x_{i+1}) = \frac{h^{p+1}}{(p+1)!} f^{(p+1)}(t) = O(h^{p+1}), \quad x_i < t < x_{i+1}.$$

Selanjutnya, Persamaan (11) dapat ditulis ulang ke dalam bentuk:

$$f(x_{i+1}) = \sum_{k=0}^{p} \frac{h^k}{k!} f^{(k)}(x_i) + O(h^{p+1}). \tag{7}$$

Karena $\hat{y}(\boldsymbol{x})$ pada Persamaan (3) merupakan hasil dari metode Euler orde 1 dan memiliki galat pemotongan berorde dua, maka $F\big(\boldsymbol{x}, N(\boldsymbol{x}, w)\big)$ harus memiliki orde $O(h^2)$ karena berlaku sebagai galat pemotongan. Ambil $p = 1$ untuk mendapatkan galat orde kedua, maka persamaaan (7) berbentuk berikut:

$$f(x_{i+1}) = f(x_i) + \frac{h}{1!} f'(x_i) + O(h^2)$$

$$f(x_{i+1}) = f(x_i) + \frac{h}{1!} f'(x_i) + \frac{h^2}{2!} f''(t). \tag{8}$$

Persamaan (8) diketahui sebagai persamaan metode Euler orde 1 dengan galat pemotongan. Sehingga, disekitar titik awal $x_i = 0$ dan $0 \leq t < x$ diperoleh deret Taylor disekitar x_i

$$f(x) = f(0) + \frac{x}{1!} f'(0) + \frac{x^2}{2!} f''(t). \tag{9}$$

Berdasarkan Persamaan (9), dapat dilihat bahwa suku terakhir ruas kanan merupakan galat orde kedua. Sehingga diperoleh

$$F(\boldsymbol{x}, N(\boldsymbol{x}_i, w)) = \frac{x^2}{2!} f''(t). \tag{10}$$

Diketahui $F(\boldsymbol{x}, N(\boldsymbol{x}, w))$ merupakan fungsi dari $\boldsymbol{x}$ dan $N(\boldsymbol{x}, w)$, dengan $N(\boldsymbol{x}, w)$ merupakan merupakan estimasi galat dan hasil dari sebuah proses *single-output feedforward neural network* dengan parameter *w* dan *n* input dari vector $\boldsymbol{x}$. Persamaan (10) dapat dibawa ke bentuk $F(x_i, N(x_i, w)) = x_i{}^2 \frac{f''(t)}{2}$, dengan $\frac{f''(t)}{2}$ akan dihitung melalui proses *neural network* yang selanjutnya dituliskan sebagai $N(x_i, w)$. Maka Solusi di titik x_i dari persamaan diferensial biasa menggunakan metode estimasi galat dengan *neural network* dapat ditulis sebagai berikut:

$$y(x_i, w) = \hat{y}(x_i) + x_i{}^2 N(x_i, w), \tag{11}$$

dimana $N(x_i, w)$ didefinisikan

$$N(x_i, w) = \sum_{j=1}^{m} w_{jk}\, \sigma(z_{in_j})\,; \quad \text{dan} \qquad z_{in_j} = v_{0_j} + \sum_{i=1}^{n} x_i\, v_{ij}$$

dengan

w_{jk} : Bobot antara unit *hidden layer* ke-*j* dengan unit *output layer* ke-*k*

σ : Fungsi aktivasi (sigmoid biner)

z_{in_j} : Sinyal informasi dari *input layer* ke unit *hidden layer* ke -*j*

v_{0_j} : Bias pada unit *hidden layer* ke-*j*

x_i : Unit *input layer* ke-*i*

v_{ij} : Bobot antara unit *input layer* ke-*i* dengan unit *hidden layer* ke-*j*

Berikut algoritma metode Euler - estimasi galat menggunakan *neural network* untuk menyelesaikan persamaan diferensial (1) dimana diketahui $x \in [0,1]$:

Tetapkan (input):

- Banyaknya lapisan: yaitu lapisan input, lapisan tersembunyi dan lapisan luaran (dalam penelitian ini masing-masing satu lapisan).
- Orde konvergensi adalah 2, yaitu sama dengan orde galat metode Euler orde 1.
- Jumlah neuron dalm lapisan tersembunyi *m.*

Output: Solusi hampiran persamaan diferensial pada titik-titik diskritisasi.

Langkah algoritma

1. Inisialisasi parameter untuk NN (learning rate α, bobot w, dan toleransi *tol* = 0,01)

2. Hitung solusi numerik sementara masalah nilai awal, $\hat{y}(x_i),\ i = 1, \cdots, 10$, menggunakan metode Euler orde 1
3. Definisikan galat absolut $E(\boldsymbol{x}) = \|y(\boldsymbol{x}) - \hat{y}(\boldsymbol{x})\| = \max_i |y(x_i) - \hat{y}(x_i)|$
4. Lakukan proses neural network menghitung $N(x, w)$ untuk estimasi $E(\boldsymbol{x})$ sehingga $E(\boldsymbol{x}) = \boldsymbol{x}^2\, N(\boldsymbol{x}, \boldsymbol{w})$ dengan *loss function* = $L(\boldsymbol{w}) = \|f(\boldsymbol{x}, y) - y'(\boldsymbol{x})\|$
5. Lakukan iterasi selama *loss function >tol* (proses *backpropagation neural network* pelatihan (training) terhadap data 11 titik pada selang [0, 1])
   i. Satu Lapisan input: banyak unit *n* =11, yaitu 11 titik hasil diskritisasi
   ii. Satu Lapisan tersembunyi dengan *m* neuron.
   Nilai bobot dan bias dicari acak menggunakan fungsi *random seed* ; hitung fungsi aktivasi yaitu fungsi *sigmoid biner* $\sigma(z) = \frac{1}{1+e^{-z}}$, kemudian parameter bobot *w* dioptimasi menggunakan teknik dasar algoritma *gradient descent*, dengan learning rate α.
   iii. Lapisan luaran (output) tunggal, $N(\boldsymbol{x}, \boldsymbol{w}^*)$, dengan parameter optimal $\boldsymbol{w}^*$.
6. Dapatkan solusi hampiran akhir: $y(\boldsymbol{x}) = \hat{y}(\boldsymbol{x}) + \boldsymbol{x}^2\, N(\boldsymbol{x}, \boldsymbol{w}^*)$

Pengoptimalan *loss function* menggunakan *gradient descent* yaitu memperbaharui setiap bobot pada neuron ke-*j* lapisan ke-*k* (w_{jk}) dengan mengurangi nilai bobot awal dengan *learning rate* (*α*) dan nilai *gradient loss function*, dapat ditulis $w_{jk}(baru) = w_{jk}(lama) - \alpha\ \frac{\partial L(w)}{\partial w}$.

Tingkat pembelajaran (*learning rate), α*, merupakan parameter penyetelan dalam proses minimalisasi dan menentukan panjang langkah pada proses *gradient descent*. Ini berarti bahwa jika tingkat pembelajaran terlalu tinggi, maka mungkin minimum lokal terlampaui dan tidak mencapai minimum yang diinginkan, sedangkan jika tingkat pembelajaran terlalu kecil, proses pelatihan membutuhkan waktu terlalu lama, sehingga biaya komputasi menjadi terlalu besar.

2.2 Metode Runge-Kutta Orde 4 (RK4)

Metode Runge-Kutta merupakan metode numerik satu langkah yang artinya saat akan mencapai titik y_{i+1}, maka diperlukan keterangan pada titik sebelumnya, yaitu x_i dan y_i. Metode RK4 merupakan jenis metode Runge-Kutta yang memiliki ketilitian yang terbilang cukup tinggi karena galat pemotogan lokal dari deret Taylor ada di urutan $O(h^5)$. Berikut merupakan skema iterasi metode RK4:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

dengan nilai masing-masing k:

$k_1 = f(x_i, y_i)$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1)$$

$$k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2)$$

$$k_4 = f(x_i + \frac{1}{2}h, y_i + hk_3)$$

Galat per langkah metode Runge-Kutta orde empat adalah $O(h^5)$ dan galat kumulatif metode RK4 adalah $O(h^4)$. Galat kumulatif ini sebanding dengan pangkat empat ukuran langkah (h^4), artinya jika *h* diperkecil setengahnya (*h*/2), maka galat akan mengecil signifikan dengan factor 16. Metode RK4 berorde tinggi yang tentu memberikan solusi yang semakin teliti. Namun ketelitian ini disertai dengan jumlah komputasi yang semakin banyak.

2.3 Tahapan Penelitian

Berikut merupakan tahap-tahap yang dilakukan dalam menganalisis perbedaan metode dalam menyelesaikan persamaan diferensial biasa linear:

1. Menentukan jumlah masukan yaitu sebanyak 11 titik berjarak seragam pada interval domain [0,1].
2. Menghitung solusi $y(x)$ persamaan diferensial secara analitik.
3. Menghitung solusi numerik persamaan diferensial menggunakan metode RK4 dengan besar langkah (h) sebesar 0,1 pada setiap titik $x_{i+1},\ i = 0, 1, \cdots, 10$.
4. Menghitung solusi persamaan diferensial menggunakan metode Euler - estimasi galat menggunakan *neural network.*
   a. Konstruksi model *neural network* terdiri dari sebuah lapisan *input* sebelas titik x yang berada dalam selang[0, 1], sebuah lapisan tersembunyi (hidden kayer) dengan satu neuron, dan sebuah lapisan *output.*
   b. Nilai bobot dicari dengan fungsi *random seed,* sedangkan nilai bias sebesar 0.
   c. Untuk empat contoh ini, digunakan metode gradient descent dasar sebagai metode minimalisasi dengan nilai *learning rate* ditetapkan sebesar 0.001.
5. Menghitung nilai galat absolut pada masing-masing metode.
6. Membuat grafik perbandingan solusi hampiran dan solusi eksak dan grafik galat absolut masing-masing kasus persamaan diferensial dari hasil metode yang digunakan.
7. Menghitung nilai MAPE masing-masing metode.
8. Menganalisis hasil solusi hampiran dan galat absolut kedua metode berdasarkan grafik yang terbentuk dan nilai MAPE yang dihasilkan.

Penelitian ini berkonsentrasi pada efisiensi algoritma untuk memecahkan persamaan diferensial biasa dengan menerapkan jaringan syaraf sederhana dalam estimasi galat metode numerik berorde rendah, dengan demikian tidak dilakukan analisis efek yang disebabkan oleh teknik atau faktor lain seperti pilihan jaringan saraf, atau skema minimalisasi.

3 Hasil dan Pembahasan

Pada bagian ini, disajikan empat contoh masalah nilai awal persamaan diferensial biasa linear. Masing-masing permasalahan tersebut diselesaikan dengan metode RK4 dan metode estimasi galat menggunakan *neural network.* Seluruh hasil perhitungan diperoleh dengan bantuan *Google Collaboration* dan Microsoft Excel. Berikut masalah nilai awal yang digunakan:

a. $\frac{dy}{dx} = x^3 + 2x + \frac{x^2(1+3x^2)}{1+x+x^3} - \left(x + \frac{1+3x^2}{1+x+x^3}\right) y$ dengan nilai awal $y(0) = 1$

b. $\frac{dy}{dx} = \frac{x-4xy}{x^2+1}$ dengan nilai awal $y(0) = \frac{1}{3}$

c. $\frac{dy}{dx} = e^{-x} + \frac{y}{2}$ dengan nilai awal $y(0) = -1$

d. $\frac{dy}{dx} = \frac{\cos x - y \sin 2x}{1+ sin^2 x}$ dengan nilai awal $y(0) = 1$

Solusi analitik dari masing-masing persamaan diatas dihitung dan diperoleh sebagai berikut:

a. $y(x) = x^2 + \frac{e^{-\left(\frac{1}{2}x^2\right)}}{1+x+x^3}$

b. $y(x) = \frac{1}{(x^2+1)^2}\left[\frac{x^4}{4} + \frac{x^2}{2} + \frac{1}{3}\right]$

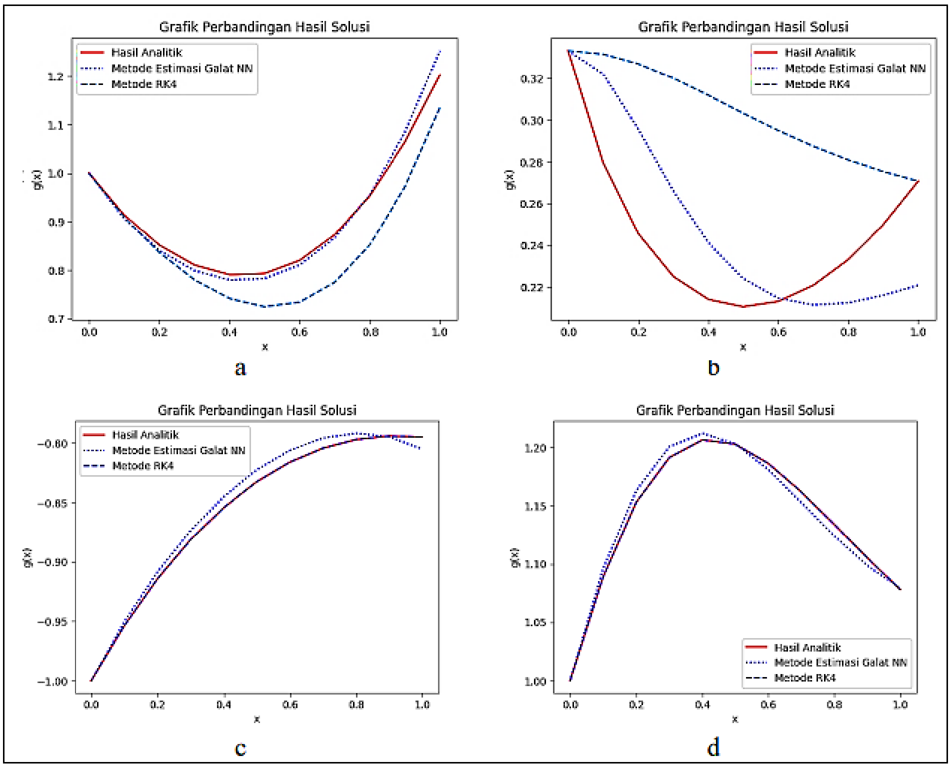
c. $y(x) = e^{\frac{1}{2}x}\left[-\frac{2}{3}e^{-\frac{3}{2}x} - \frac{1}{3}\right]$

d. $y(x) = \frac{1}{1+ sin^2 x}[\sin x + 1]$

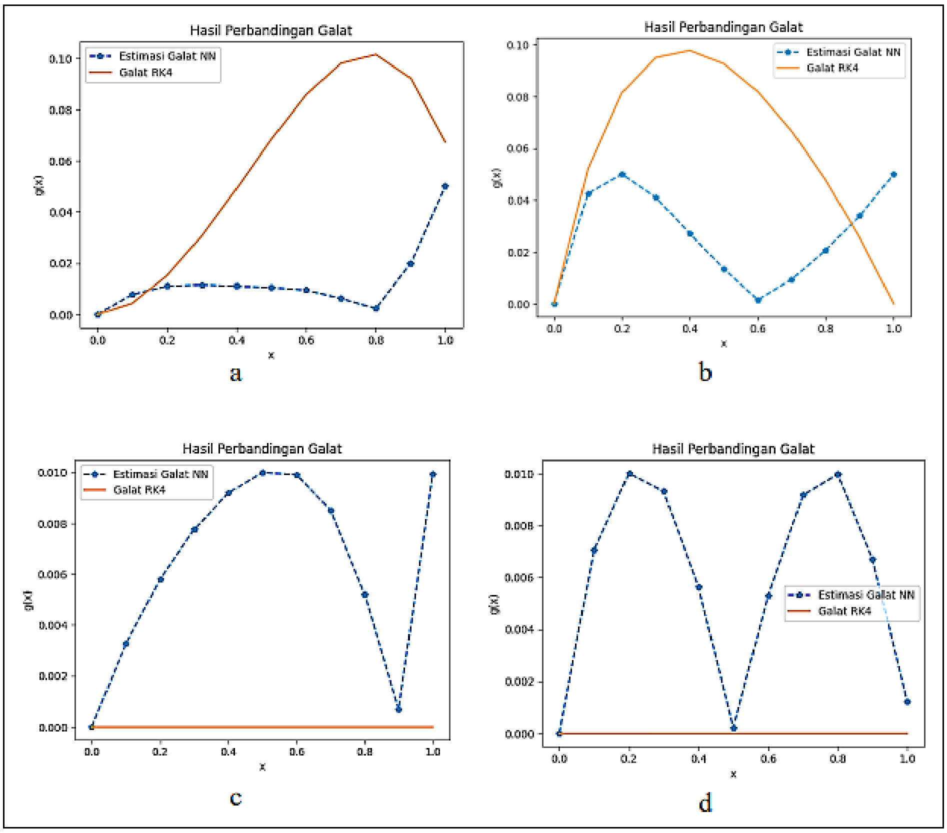
Gambar 1 dan Gambar 2 menunjukkan grafik perbandingan hasil solusi serta galat absolut berturut-turut dari metode RK4 dan metode estimasi galat menggunakan *neural network* terhadap solusi analitik masing-masing persamaan diferensial a, b, c, dan d.

Pada Gambar 1 dapat dilihat metode estimasi galat menggunakan *neural network* selalu menghasilkan solusi yang mendekati solusi analitik untuk setiap persamaan diferensial. Hal ini berbeda dengan metode RK4 terlihat memiliki solusi numerik yang sama dengan solusi analitiknya untuk kasus persamaan diferensial (c) dan (d), namun hal ini tidak berlaku untuk kasus persamaan diferensial (a) dan (b) yang solusi numeriknya terlihat menjauh dari solusi analitik.

Untuk melihat perbedaan lebih jelas dari kedua metode, telah dilakukan perhitungan galat absolut pada masing-masing metode dan grafik galat dapat dilihat pada Gambar 2.



Gambar 1. Grafik Perbandingan Solusi Analitik dan Numerik



Gambar 2. Grafik Hasil Perhitungan Galat Absolut Metode Numerik terhadap Solusi Analitik

Berdasarkan Gambar 2, dapat dilihat bahwa metode estimasi galat menggunakan *neural network* cenderung memiliki nilai galat yang stabil. Hal ini dikarenakan, pada metode ini terdapat nilai batas maksimum *error* yang dihasilkan (nilai toleransi) pada saat tahap pengoptimuman *loss function.* Dapat dilihat, kasus persamaan diferensial (a) dan (b) memiliki nilai galat tertinggi 0.05, yang merupakan nilai maksimum toleransi *error*. Begitu juga pada persamaan diferensial untuk kasus (c) dan (d) yang memiliki nilai galat maksimum pada angka 0.01. Sedangkan untuk metode RK4 terlihat bahwa pada persamaan (a) dan (b) memiliki galat yang lebih besar dibandingkan metode estimasi galat dengan *neural network.* Namun terjadi sebaliknya pada persamaan (c) dan (d), galat yang dihasilkan oleh metode RK4 hampir mendekati nol dengan sempurna.

Selanjutnya, dalam menentukan apakah galat yang dihasilkan oleh suatu metode memiliki akurasi yang baik, digunakan kriteria yang diusulkan oleh Lewis (1982) berdasarkan nilai *Mean Absolute Percentage Error* (MAPE) [12].

Pada Tabel 1, terlihat bahwa nilai MAPE metode estimasi galat menggunakan *neural network* termasuk kategori dengan akurasi yang sangat tinggi untuk setiap kasus persamaan diferensial yang digunakan. Selanjutnya, nilai MAPE yang dihasilkan oleh metode Runge-Kutta Orde-4.

Tabel 1. Tabel MAPE Metode Penyelesaian Persamaan Diferensial Biasa

Persamaan Diferensial	MAPE	
	Metode Estimasi Galat *Neural Network*	Metode Runge Kutta Orde Empat
a	1,31%	6,13%
b	9,80%	25,72%
c	0,76%	0%
d	0,51%	0%

berada pada dua kategori, yaitu nilai MAPE memiliki keakurasian yang sangat tinggi untuk persamaan diferensial (a), (c), dan (d) serta nilai MAPE dengan keakurasian yang wajar untuk kasus persamaan diferensial (b).

4 Simpulan

Berdasarkan empat contoh penyelesaian masalah nilai awal yang telah dipaparkan, dapat dilihat bahwa berdasarkan Gambar 2, nilai galat absolut untuk metode estimasi galat menggunakan *neural network* lebih stabil dibandingkan dengan metode Runge-Kutta Orde 4. Hal ini berarti bahwa nilai galat dari metode estimasi galat menggunakan *neural network* masih bisa ditoleransi. Selain itu, nilai MAPE yang dihasilkan oleh metode estimasi galat menggunakan *neural network* termasuk kedalam kategori dengan tingkat akurasi yang sangat tinggi, berbeda dengan metode Runge-Kutta Orde-4 dimana nilai MAPE tergolong pada dua kategori, yaitu akurasi yang sangat tinggi dan akurasi wajar. Skema metode Euler dengan Galat-NN dapat mengestimasi besaran galat yang sesuai dengan orde konvergensi dari metode Euler orde-1 yang digunakan untuk solusi sementara. Beberapa hasil numerik menunjukkan bahwa skema bisa mendapatkan akurasi yang sangat baik sebagaimana metode RK4.

Efisiensi metode dapat ditingkatkan dengan mempertimbangkan optimasi beberapa parameter yang dapat dikontrol dalam implementasi *neural network*. Lebih lanjut dapat dilakukan pengembangan contoh persamaan diferensial yang berupa sistem atau persamaan diferensial jenis parsial. Selain itu, dapat juga menggolongkan jenis persamaan yang digunakan apakah termasuk dalam implisit atau eksplisit, periodik atau tidak, agar jenis metode numerik yang digunakan bisa sesuai.

5 Daftar Pustaka

[1] S. L. Ross, *DIfferential Equation (third edition)*. New York: John Wiley&SonsInc., 2004.

[2] H. Nam, K. R. Baek, and S. Bu, “Error estimation using neural network technique for solving ordinary differential equations,” *Advances in Continuous and Discrete Models*, vol. 2022, no. 1, Dec. 2022, doi: 10.1186/s13662-022-03718-4.

[3] I. E. Lagaris, A. Likas, and D. I. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations,” *IEEE Trans Neural Netw*, vol. 9, no. 5, pp. 987–1000, 1998, doi: 10.1109/72.712178.

[4] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” in *NIPS’18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, New York: Curran Associates Inc, Jun. 2018, pp. 6572–6583.

[5] P. Kim, X. Piao, W. Jung, and S. Bu, “A new approach to estimating a numerical solution in the error embedded correction framework,” *Adv Differ Equ*, vol. 2018, no. 1, p. 168, Dec. 2018, doi: 10.1186/s13662-018-1619-6.

[6] W. Jiang and C. Xuan, “Neural Network for Solving Ordinary Differential Equations,” in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, Dec. 2022, pp. 261–265. doi: 10.1109/CICN56167.2022.10008361.

[7] Dewi Erla Mahmudah, Ratna Dwi Christyanti, Moh. Khoridatul Huda, and Fidia Deny Tisna Amijaya, "Penyelesaian Masalah Syarat Batas dalam Persamaan Diferensial Biasa Orde Dua dengan Menggunakan Algoritma Shooting Neural Networks," *Teknikom: Teknologi Informasi, Ilmu Komputer Dan Manajemen*, vol. 1, no. 2, pp. 67–78, 2017.

[8] W. Guasti Junior and I. P. Santos, "Solving Differential Equations Using Feedforward Neural Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 385–399. doi: 10.1007/978-3-030-86973-1_27.

[9] A. Golovkina and V. Kozynchenko, "Neural Network Representation for Ordinary Differential Equations," 2023, pp. 39–55. doi: 10.1007/978-3-031-22938-1_3.

[10] Jayme Yeremia Wijaya, The Houw Liong, and Ken Ratrri Retno Wardani, "Perbandingan Penyelesaian Persamaan Diferensial Biasa Menggunakan Metode Backpropagation, Euler, Heun, dan Runge-Kutta Orde 4," *Jurnal Telematika*, vol. 11, no. 1, pp. 1–6, 2016.

[11] Fardinah F, "Solusi Persamaan Diferensial Biasa dengan Metode Runge-Kutta Orde Lima," *Jurnal Matematika Dan Statistika Serta Aplikasinya*, vol. 5, no. 1, pp. 30–36, 2017.

[12] R. J. C. Chen, P. Bloomfield, and F. W. Cubbage, "Comparing Forecasting Models in Tourism," *Journal of Hospitality and Tourism Research*, vol. 32, no. 1, pp. 3–21, 2008, doi: 10.1177/1096348007309566.