

Implementasi Artificial Neural Network pada Kendali Bola Pendulum

Danu Arrival Hidayat¹⁾, Farras Fauzan²⁾, Andy Dwi Cahyo³⁾,
Dhany Eka Yulian⁴⁾, Muhammad Thaifur⁵⁾, Wahyu S Pambudi⁶⁾

^{1),2),3),4),5),6)} Teknik Elektro, Institut Teknologi Adhi Tama Surabaya
Jl. Arief Rahman Hakim No.100, Surabaya
Email : danuarrival25@gmail.com

Abstrak. Pada era revolusi industri 4.0 ini proses dari produksi suatu barang yang dibutuhkan memiliki peningkatan sehingga diperlukan sistem yang cerdas (*machine learning*). Salah satu *machine learning* yang dapat digunakan adalah Artificial Neural Network (ANN) dengan *error backpropagation*. Proses implementasi ANN perlu studi awal dalam bentuk simulasi, dimana simulasi yang bisa dibangun adalah aplikasi kendali untuk mengikuti target gerak natural pendulum dengan menggunakan metode Runge-kutta. Parameter dasar yang ada pada aplikasi ini meliputi massa, length, damping, theta, alpha, dan gain. Pengujian yang dilakukan 3 kali dengan parameter nilai yang diubah adalah massa, length, dan gain. Hasil percobaan pertama nilai parameter massa diubah dengan nilai 2, 5, dan 10 dengan hasil error average yang didapatkan adalah 0.51%, 1.46%, dan 5.1%. Hasil pengujian kedua dilakukan perubahan pada parameter length dengan nilai variasi 1, 4, dan 6 dengan hasil error average 0.6%, 4.78%, dan 5.4%. Hasil pengujian ketiga parameter yang diubah adalah gain nilai variasi 10000, 14000, dan 20000 dengan hasil error average yang didapatkan adalah 4.93%, 5.62%, dan 2.05%.

Katakunci: Industri 4.0, Artificial Neural Network, Pendulum.

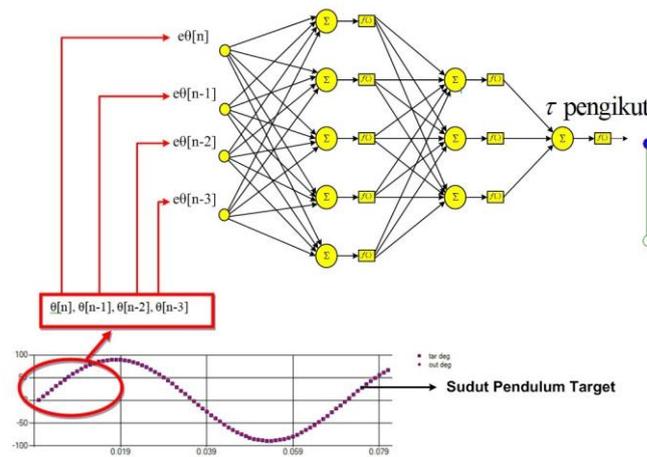
1. Pendahuluan

Revolusi industri 4.0 merupakan revolusi yang melibatkan sistem cerdas dan otomasi sebagai pengganti dari tenaga manusia dalam bidang industri. Hal ini buktikan dengan efisiensi yang dilakukan oleh sistem cerdas dalam melakukan hal yang mendasar yang biasa dikerjakan oleh manusia melalui teknologi AI[1]. Salah satu sistem AI (*Artificial Intelligence*) adalah ANN (*Artificial Neural Network*), yang dimana *Artificial Neural Network* memiliki konsep yang sama seperti otak pada manusia yang dimana memiliki neuron-neuron dan *layer-layer* tertentu agar dapat melakukan proses yang dapat dilakukan oleh manusia [2]. Penerapan ANN bisa diterapkan dalam berbagai hal seperti melakukan mengklarifikasi bentuk, memetakan bentuk dari input menjadi bentuk baru pada *output*, menyimpan bentuk yang kemudian digunakan/dipanggil kembali, memetakan berbagai bentuk yang sejenis, mengoptimisasi masalah, dan melakukan prediksi. Untuk aplikasi yang digunakan kali ini adalah memetakan pola *input* ke *output* dengan perantara aplikasi pendulum[3]. Pendulum merupakan suatu benda yang memiliki *massa* dan terikat pada sebuah tali yang dapat berayun secara bebas dan memiliki nilai periodik yang dapat dijadikan dasar dari pergerakan jam pada masa kuno. Pada sistem ANN ini dilakukan pemetaan pola *input* yang berupa pendulum, menjadi pola baru yang sama berupa *output*[4][5].

2. Pembahasan

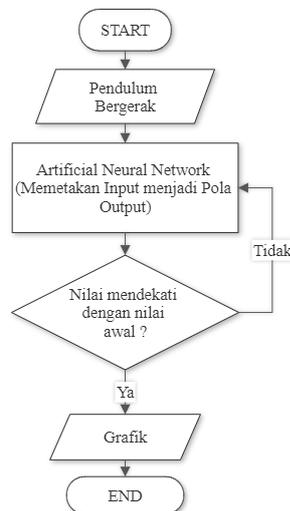
2.1. Blok Diagram

Pada penelitian ini menggunakan *Artificial Neural Network* seperti pada Gambar 1, dimana pendulum melakukan pergerakan dan menghasilkan *output*. Tugas dari *Artificial Neural Network* adalah melakukan pemrosesan *input* dan *output* dari pergerakan pendulum lalu melakukan pembuatan penyamaan antara pergerakan pendulum dengan hasil *output* dari *Artificial Neural Network* yang sudah diproses.



Gambar 1. Blok Diagram *Neural Network*

2.2. Flowchart



Gambar 2. *Flowchart* Simulasi *Neural Network* Pendulum

Awal dari Gambar 2 ini adalah melakukan setting parameter yang digunakan untuk pergerakan pendulum yaitu dengan menambah *mass*, *length*, dan *damping* dari pendulum. Sistem melakukan pembacaan bentuk pola yang dibuat oleh pendulum dan menyesuaikan nilai awal dari pendulum, berikutnya hasil dari pembacaan yang dilakukan *Neural Network* di tampilkan melalui perantara grafik dengan tujuan agar pembacaan dari grafik dapat di analisa dengan baik.

2.3. *Artificial Neural Network*

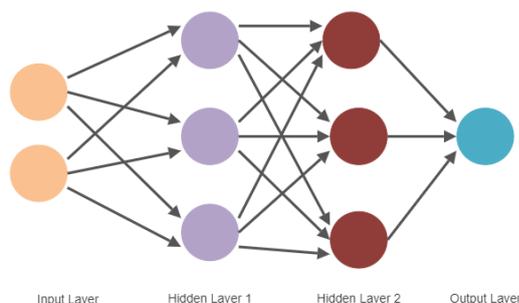
Artificial Neural Network adalah sekelompok dari suatu unit pemrosesan kecil yang dibentuk dan di proses yang didasarkan pada sistem saraf manusia. *Neural Network* sendiri adalah sistem yang adaptif dimana sistem dapat mengubah stuktur yang memiliki kegunaan agar dapat memecahkan masalah dengan informasi yang didapat dari eksternal maupun internal yang ada pada jaringan pada sistem tersebut, dimana simulasi dari strukturnya sama dengan dengan otak manusia yang dimana memiliki neuron-neuron yang bercabang. Data yang sudah dilakukan pemrosesan lalu dikirimkan ke perangkat lunak agar bias mengenali pola bentuk yang kompleks serta melakukan iterasi dari data-data yang sudah di lakukan sebelumnya beberapa kali yang dilakukan. Ada 3 jenis pembelajaran atau lebih dikenal dengan pelatihan dari sistem *Neural Network* yaitu pembelajaran terawasi, pembelajaran tak terawasi dan pembelajaran yang diperkuat. Dalam pembelajaran terawasi apabila proses hasil dari yang sudah di ambil dari data yang sudah diketahui sebelumnya, lapisan dari keluaran yang akan di bangkitkan

nantinya akan dilakukan perbandingan berupa hasil *error* yang apabila hasil *error* tersebut cukup besar maka perlu dilakukan iterasi lagi. Pada pembelajaran tak terawasi tidak memerlukan keluaran target yang dimana target dari keluaran ini adalah dengan mengelompokkan beberapa unit yang mendekati dari suatu area yang sudah ditentukan. Metode berikutnya adalah pembelajaran yang diperkuat, dimana metode ini merupakan metode yang mengambil data dari data yang sudah di ambil sebelumnya dan digunakan untuk pembelajaran dengan hasil yang memungkinkan mendekati dari data target.

2.4. Backpropagation

Metode ini merupakan metode pembelajaran yang ada pada Neural Network, yang dimana arsitektur jaringannya tidak memiliki koneksi *feedback*/umpan balik akan tetapi *error*nya dipropagasikan selama melakukan *training*. *Backpropagation* merupakan suatu metode yang pada umumnya digunakan untuk proses neural lebih dari satu, biasanya pada *backpropagation* ini memiliki tiga lapis layer yang hampir bisa menyelesaikan masalah yang ada pada dunia industri modern. Metode ini mempunyai dasar sistematis yang cukup kuat dalam pembuatan objektif dan algoritma, dimana metode ini mendapatkan Persamaan dan keluaran yang koefisien dalam meminimalkan jumlah *error* yang terjadi melalui pembelajaran yang bisa dilakukan dalam model yang bisa dikembangkan (*training set*):

1. Awalnya dengan *input layer*, melakukan perhitungan dari setiap layer *output* dan melakukan pemrosesan melalui lapisan luar
2. Kemudian menghitung *error* yang terjadi pada *input layer* yang merupakan selisih antara data aktual dan target yang akan dicapai
3. Berikutnya ubah *error* yang terjadi tersebut kepada kesalahan pada masukan elemen pemrosesan
4. Lakukan *backpropagation* dari *error* yang telah dilakukan pada *output* pada setiap elemen pemroses ke *error* yang ada pada *input*. Lakukan pengulangan proses ini sampai *input* target tercapai
5. Selanjutnya mengubah keseluruhan bobot dengan cara menggunakan *error* yang ada pada sisi *input* elemen dan *output* elemen yang melakukan proses



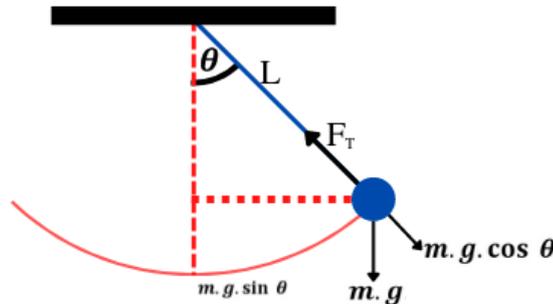
Gambar 3. Pemodelan *Neural Network*

Gambar 3 merupakan contoh dari pemodelan sistem *Neural Network* yang dimana memiliki *input*, *hidden* 1 dan 2, *output layer*. Berdasarkan dari pemodelan di atas bisa di ambil bahwa proses *Neural Network* selalu melakukan iterasi berkali kali agar mendapatkan nilai yang ideal dengan akurasi yang tinggi, maka data yang diambil dan diproses harus jelas dengan ukuran yang dapat dihitung secara matematis. Didalam metode ini, setiap unit *layer* akan berhubungan dengan setiap unit di *hidden layer* akan terhubung dengan unit *output layer*. Ketika jaringan ini diberi masukan pola yang digunakan sebagai *training set*, maka bentuk pola tersebut akan mengarah ke unit-unit *hidden layer* yang lain untuk melakukan tahapan berikut yang akan di lanjutkan pada *output layer*, dan *output layer* akan melakukan respon dari keluaran pada *nural network*. Hasil dari keluaran jika tidak sesuai dengan yang ditargetkan, maka hasil yang akan dilakukan aktifasi secara *backpropagation* pada *hidden layer* dengan tujuan menuju *input layer*.

2.5. Pendulum

Pendulum adalah suatu benda yang digantungkan dan diikat dengan tali yang dimana dapat berayun dengan gaya bebas dan memiliki periodik yang dapat menjadi dasar dari kerja jam dinding kuno yang memiliki pergerakan ayunan di dalamnya. Pada bidang fisika, prinsip dari pendulum ini pertama ditemukan oleh Galileo Galilei pada tahun 1602 bahwa periode/ lama gerak dari suatu bandul ditentukan dari panjang tali dan percepatan gravitasi yang terjadi pada benda dengan Persamaan (1).

$$F = -m \cdot g \cdot \sin \theta \dots \dots \dots (1)$$



Gambar 4. Pendulum Sederhana

Dari $F = m \cdot a$ yang dimana sama dengan $F = -m \cdot g \cdot \sin \theta$ maka runtutan bertahap pada Persamaan (2),

$$m \cdot a = -m \cdot g \cdot \sin \theta \dots \dots \dots (2)$$

$$m(\omega^2 y) = -m \cdot g \cdot \frac{y}{l} \dots \dots \dots (3)$$

$$\omega^2 = \frac{g}{l} \dots \dots \dots (4)$$

$$\left[\frac{2\pi}{T}\right]^2 = \frac{g}{l} \dots \dots \dots (5)$$

$$\frac{4\pi^2}{T^2} = \frac{g}{l} \dots \dots \dots (6)$$

$$T^2 = 4\pi^2 \frac{l}{g} \dots \dots \dots (7)$$

$$T = \sqrt{4\pi^2 \frac{l}{g}} \dots \dots \dots (8)$$

Maka didapatkan ,

$$T = 2\pi \sqrt{\frac{l}{g}}$$

Dimana ,

- l = Panjang dari bandul dalam satuan meter
- g = Percepatan gravitasi yang terjadi
- T = Periode

Sehingga frekuensi : $f = \frac{1}{T}$ maka , $f = \frac{1}{2\pi} \sqrt{\frac{g}{l}}$

2.6. Metode Runge-Kutta

Persamaan differensial merupakan suatu gabungan dari beberapa fungsi yang tidak dapat diketahui secara jelas turunannya. Persamaan ini hanya memiliki perubah tunggal bebas. Perubah bebas biasanya dapat disimbolkan dengan huruf x. Metode Runge-Kutta yang dimana salah satu perangkat dengan metode yang biasanya digunakan agar dapat melakukan penyelesaian Persamaan diferensial dengan persyaratan awal yaitu [5]:

$$y' = f(x, y)$$

$y(x_0)$ diberikan agar dapat melakukan pemecahan persoalan yang ada, pada sumbu waktu x dipilih simpul dari waktu diskrit $x_0, x_1, x_2, \dots, x_r, x_{r+1}$ dan seterusnya dengan $x_{r+1} = x_r + h_r$, dengan h_r merupakan "step length" keadaan ke-r ke saat ke-(r+1). Biasanya "step length" ini tergantung dengan r, akan tetapi untuk membuatnya lebih mudah dapat agar mendapatkan data secara konstan, yaitu h. Memiliki bentuk umum dari Runge-Kutta orde-n adalah:

$$y_{r+1} = y_r + a_1k_1 + a_2k_2 + \dots + a_nk_n \dots \dots \dots (9)$$

Dengan tetapan a_1, a_2, \dots, a_n dan

$$k_1 = hf(x_r, y_r) \dots \dots \dots (10)$$

$$k_2 = hf(x_r + p_1h, y_r + q_{11}k_1) \dots \dots \dots (11)$$

$$k_3 = hf(x_r + p_1h, y_r + q_{21}k_1 + q_{22}k_2) \dots \dots \dots (12)$$

$$k_n = hf(x_r + p_{n-1}h, y_r + q_{n-1,1}k_1 + q_{n-1,2}k_2 + \dots + q_{n-1,n-1}k_{n-1}) \dots \dots \dots (13)$$

Runge-Kutta Orde Pertama

Orde pertama ini berbentuk: $k_1 = hf(x_r, y_r)$ seperti pada Persamaan berikut (14).

$$y_{r+1} = y_r + (a_1k_1) \dots \dots \dots (14)$$

Runge-Kutta Orde Kedua

Orde kedua berbentuk seperti pada Persamaan (15).

$$k_1 = hf(x_r, y_r)$$

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

$$y_{r+1} = y_r + (a_1k_1 + a_2k_2) \dots \dots \dots (15)$$

Runge-Kutta Orde Ketiga

Orde yang sering digunakan dalam penggunaan pada umumnya adalah orde ketiga dan orde keempat. Kedua metode tersebut sangat sering digunakan karena ketelitian solusinya yang sangat tinggi jika dibandingkan dengan orde-orde sebelumnya. Orde ketiga berbentuk seperti Persamaan (16).

$$k_1 = hf(x_r, y_r)$$

$$k_2 = hf\left(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_1\right)$$

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

$$y_{r+1} = y_r + \left(\frac{k_1 + 4k_2 + k_3}{6}\right) \dots \dots \dots (16)$$

Runge-Kutta Orde Keempat

Orde keempat menghasilkan sesuatu yang eksak yang menghasilkan orde keempat berbentuk (17) :

$$\begin{aligned}
 k_1 &= hf(x_r, y_r) \\
 k_2 &= hf\left(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_1\right) \\
 k_3 &= hf\left(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_2\right) \\
 k_4 &= hf(x_r + h, y_r + k_3) \\
 y_{r+1} &= y_r + \left(\frac{k_1+4k_2+k_3}{6}\right) \dots\dots\dots(17)
 \end{aligned}$$

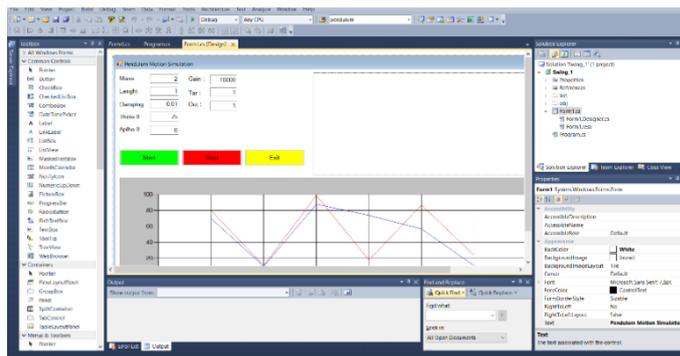
Runge-Kutta Orde Kelima

K_5 adalah orde kelima dengan bentuk Persamaan (18).

$$\begin{aligned}
 k_1 &= hf(x_r, y_r) \\
 k_2 &= hf\left(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_1\right) \\
 k_3 &= hf\left(x_r + \frac{1}{4}h, y_r + \frac{3}{16}k_1 + \frac{1}{16}k_2\right) \\
 k_4 &= hf\left(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_3\right) \\
 k_5 &= hf\left(x_r + \frac{3}{4}h, y_r + \frac{3}{16}k_2 + \frac{1}{16}k_3 + \frac{9}{16}k_4\right) \\
 y_{r+1} &= y_r + \frac{1}{9}(7k_1 + 32k_3 + 12k_4 + 7k_5)O(h^6) \dots\dots\dots(18)
 \end{aligned}$$

2.8. Visual Studio

Software yang digunakan untuk melakukan penelitian adalah Visual Studio seperti berikut :

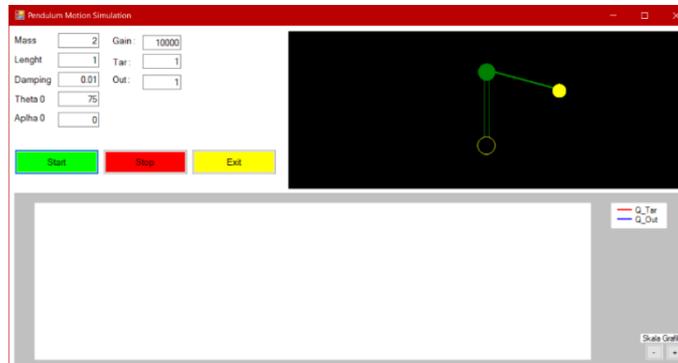


Gambar 5. Tampilan Software

Tampilan pada Gambar 5 akan menjadi awal dari aplikasi yang akan digunakan, visual basic sendiri adalah suatu software pemrograman yang dapat memudahkan pengguna untuk membuat suatu rancangan aplikasi meskipun programmer tersebut adalah pemula, dikarenakan software ini memudahkan pengguna untuk memahami apa saja yang bias dibuat untuk aplikasi yang diinginkan. Kelebihan dari visual basic sendiri adalah hasil output file yang akan disimpan memiliki ukuran yang cukup kecil dan kebanyakan digunakan oleh suatu komunitas pemrograman, dan tidak bias di hindari lagi apabila suatu software memiliki kekurangan yaitu disaat menjalankan program memori yang dibutuhkan besar dan software ini hanya bias digunakan pada Operating Sistem Windows.

2.9. Hasil

Pada penelitian ini melakukan analisa mengenai pendulum pengikut yang dimana pergerakannya diikuti oleh pergerakan yang dihasilkan oleh sistem *Neural Network*, dengan perantara *software* Visual Basic sebagai basis dasar program yang akan di uji cobakan pada Gambar berikut.



Gambar 6. Tampilan Awal saat Program Dijalankan

Dari tampilan pada Gambar 6 terdapat 3 bagian penting yaitu data pendulum, gambar pendulum, dan grafik pendulum.

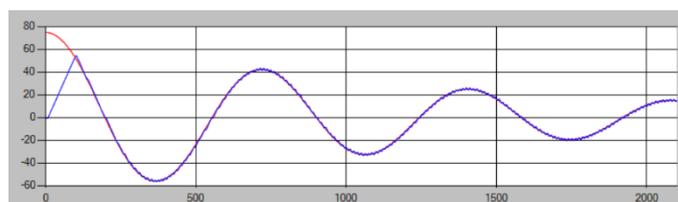
- Mass atau berat pendulum → satuan Kg
- Length atau panjang lengan pendulum → satuan meter
- *Damping* atau redaman
- *Theta* merupakan nilai awal sudut pada pendulum merah ketika memulai
- *Alpha* merupakan nilai awal sudut pada pendulum biru ketika memulai
- *Gain* atau nilai respon pendulum biru terhadap merah
- *Q_Tar* adalah besar sudut yang dihasilkan oleh pendulum merah
- *Q_Out* adalah besar sudut yang dihasilkan oleh pendulum biru

2.10. Analisa Data

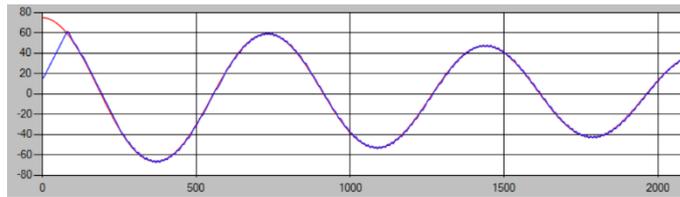
Berikut pembahasan dari analisa yang dilakukan yaitu dengan mengamati respon yang dikeluarkan melalui grafik yang sudah tersediakan dari pendulum pengikut terhadap target yang masing masing divisualisasikan dengan garis merah dan garis biru. Ada 3 pengujian yang dilakukan, pengujian pertama dilakukan perubahan pada parameter *massa* yang berpengaruh pada besar tampilan dari bola pendulum, pengujian kedua dilakukan perubahan pada nilai *length* yang dimana mengubah panjang dari tali pendulum, dan untuk pengujian ketiga dilakukan perubahan pada nilai *gain* yang mempengaruhi kecepatan awal dari pembacaan *neural network*.

Pengujian Pertama

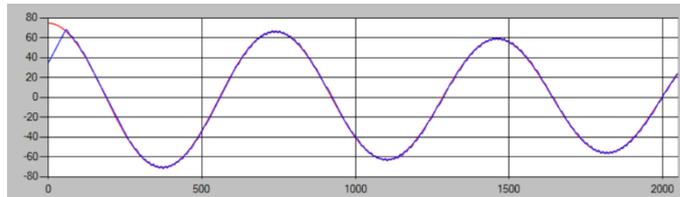
Pada pengujian pertama ini nilai pada *massa* akan diubah sementara untuk nilai parameter yang lain diantaranya nilai *length*, *damping*, *Theta*, *Alpha* dan *Gain* dibuat konstan masing-masing sebesar 1, 0.5, 75, 0 dan 10000.



Gambar 7. Grafik ketika *Massa* = 2 Kg



Gambar 8. Grafik ketika $Massa = 5$ Kg



Gambar 9. Grafik ketika $Massa = 10$ Kg

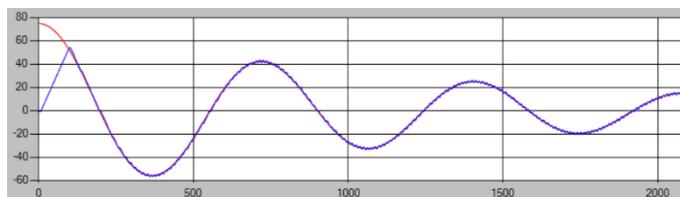
Tabel 1. Perbandingan nilai $Massa$

$Massa$	$Lenght$	$Damping$	$Theta$	$Alpha$	$Gain$	Q_Tar	Q_Out	$Error\ Average$ (%)
2	1	0.5	75	0	10000	14,83599655	14,91313355	0,519931381
5	1	0.5	75	0	10000	35,2048456	34,68983626	1,462893347
10	1	0.5	75	0	10000	24,38850814	23,14272672	5,108067333

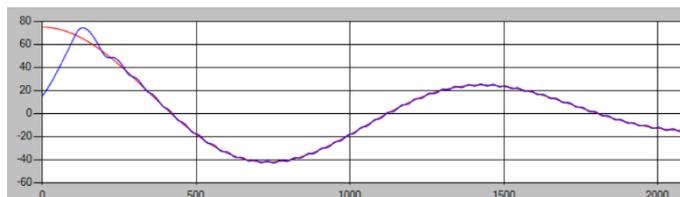
Dari perbandingan tabel 1 diatas bisa didapatkan bahwa semakin tinggi $massa$ yang di $inputkan$ maka akurasi nya lama kelamaan akan menjadi berkurang bisa dibuktikan dengan data pada table 1 dengan hasil yang sebagaimana bisa dilihat pada Gambar 8, 9, dan 10. $Massa$ pada 2 Kg memiliki $error$ yang kecil dibandingkan dengan $massa$ pada 5 dan 10 Kg.

Pengujian Kedua

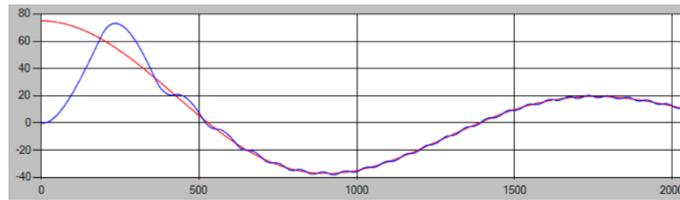
Pada pengujian kedua ini parameter yang akan diubah adalah parameter $Lenght$ dengan nilai $Massa$, $Damping$, $Theta$, $Alpha$ dan $Gain$ diatur secara konstan yaitu 2 Kg, 1, 75, 0, dan 10000.



Gambar 10. Grafik ketika $Lenght = 1$



Gambar 11. Grafik ketika $Lenght = 4$



Gambar 12. Grafik ketika $Length = 6$

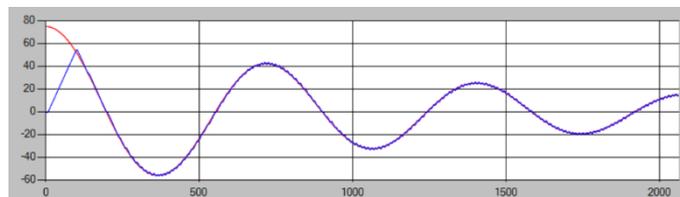
Tabel 2. Perbandingan nilai $Length$

$Massa$	$Length$	$Damping$	$Theta$	$Alpha$	$Gain$	Q_Tar	Q_Out	$Error\ Average\ (%)$
2	1	0.5	75	0	10000	15,0507155	14,95374767	0,644273873
2	4	0.5	75	0	10000	14,63063724	15,33061261	4,784312216
2	6	0.5	75	0	10000	10,52684904	11,09613817	5,407972788

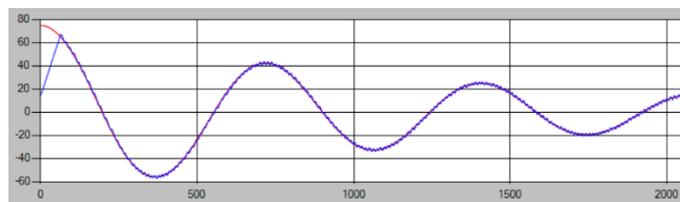
Pada tabel 2 dapat dilihat nilai $error$ pada panjang $Length$ 1 yaitu 0.64% dan nilai $error$ semakin bertambah setiap panjang $Length$ ditambahkan sampai 5.4%. $Overshoot$ dari perubahan panjang $Length$ apabila diamati memiliki perbedaan, dimana ketika lengan $Length$ 6 $overshoot$ yang dihasilkan sangat besar apabila dibandingkan dengan lengan $Length$ 1.

Pengujian Ketiga

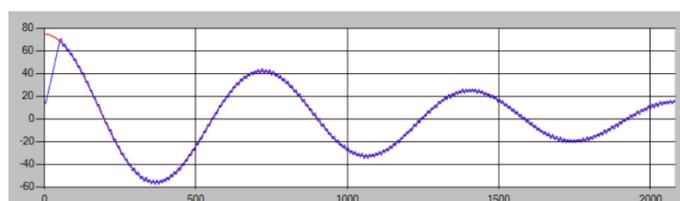
Pada pengujian ini parameter yang akan dirubah adalah $Gain$ dengan nilai $Massa$, $Length$, $Theta$, $Alpha$ dan $damping$ diatur secara konstan yaitu 2 Kg, 1, 75, 0, dan 0.5



Gambar 13. Grafik ketika $Gain = 10000$



Gambar 14. Grafik ketika $Gain = 14000$



Gambar 15. Grafik ketika $Gain = 20000$

Tabel 3. Perbandingan nilai $Gain$

<i>Massa</i>	<i>Lenght</i>	<i>Damping</i>	<i>Theta</i>	<i>Alpha</i>	<i>Gain</i>	Q_Tar	Q_Out	<i>Error Average (%)</i>
2	1	0.5	75	0	10000	14,8938791	14,15916184	4,933014816
2	1	0.5	75	0	14000	14,59842641	15,41896889	5,620759804
2	1	0.5	75	0	20000	15,09037441	15,40012967	2,052667791

Pada tabel 3 dapat didapatkan bahwa semakin besar *gain* yang di berikan maka nilai awalnya menjadi lebih cepat apabila membandingkan *Gain* 10000, 140000, dan 20000. Eksekusi dari *gain* 10000 lebih besar nilai *error averagenya* dari pada *gain* 20000 dengan *gain* 1000 mendapatkan *error average* 4,93% dan *gain* dari 20000 mendapatkan *nilai average* 2,05%.

3. Simpulan

Hasil pada pengujian 1, 2, dan 3 tersebut dapat menunjukkan bahwa sistem *Neural Network* dapat melakukan pemetaan pola dari input ke pola baru pada output yang dimana perantara dari penelitian ini adalah pergerakan pendulum. Percobaan pertama merubah parameter *Massa* dimana diberi variasi 2 Kg, 5 Kg, dan 10 Kg dengan hasil yang bisa dilihat yaitu semakin besar benda atau berat *massa* maka akurasi yang didapatkan akan menurun. Percobaan kedua dengan mengubah parameter panjang *Lenght* dan hasil yang didapatkan adalah semakin besar nilai panjang *Lenght* maka semakin besar juga *error* yang dihasilkan.

Ucapan Terima Kasih

Penulis berterima kasih kepada Tuhan Yang Maha Esa atas restunya dan bantuan kerja sama dari anggota yang sudah membantu untuk menyelesaikan proses dari pembuatan awal sampai akhir.

Daftar Pustaka

- [1]. L. Rohida, S. Sos, M. Si, D. Sumber, and D. Manusia, 2018 , “Pengaruh Era Revolusi Industri 4.0 terhadap Kompetensi Sumber Daya Manusia,”.
- [2]. Euis Saraswati, Yuyun Umaidah, and Apriade Voutama, 2021 , “Penerapan Algoritma Artificial Neural Network untuk Klasifikasi Opini Publik Terhadap Covid-19,”.
- [3]. Wahyu S. Pambudi, Enggar Alfianto, Andy Rachman, and Dian Puspita Hapsari, Mar. 2019, “Simulation design of trajectory planning robot manipulator,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 196–205, doi: 10.11591/eei.v8i1.1179.
- [4]. Kintan Limiansih and Ign Edi Santosa, Dec. 2013, “Redaman Pada Pendulum Sederhana,” *Jurnal Fisika Indonesia*, vol. 17.
- [5]. Rahayu Puji Utami, 2005, “METODE RUNGE-KUTTA UNTUK SOLUSI Persamaan PENDULUM ,” *Jurusan Matematika*.