



**TUGAS AKHIR - MT141585**

**PEMODELAN SISTEM *TURRET GUN* DENGAN  
METODE JARINGAN SYARAF TIRUAN  
*BACKPROPAGATION***

**AGUNG KARTIKA FIBRIANTO  
NRP 2112100062**

**Dosen Pembimbing  
Arif Wahjudi, S.T., M.T., Ph.D.**

**JURUSAN TEKNIK MESIN  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*(Halaman ini sengaja dikosongkan)*



**FINAL PROJECT - MT141585**

***TURRET GUN SYSTEM MODELLING WITH  
NEURAL NETWORK BACKPROPAGATION  
METHODE***

**AGUNG KARTIKA FIBRIANTO  
NRP 2112100062**

*Academic Supervisor*  
**Arif Wahjudi, S.T., M.T., Ph.D.**

**MECHANICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technologi  
Sepuluh Nopember Institute of Technology  
Surabaya 2017**

*(Halaman ini sengaja dikosongkan)*

**PEMODELAN SISTEM *TURRET GUN* DENGAN  
METODE JARINGAN SYARAF TIRUAN  
*BACKPROPAGATION***

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada  
Program Studi S-1 Jurusan Teknik Mesin  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

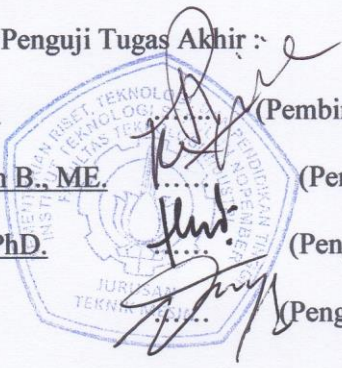
Oleh :

**AGUNG KARTIKA FIBRIANTO**

NRP. 2112 100 062

Disetujui oleh Tim Penguji Tugas Akhir :

1. Arif Wahjudi, ST, MT, PhD.  
NIP. 197303222001121001 (Pembimbing)
2. Prof. Dr. Ing. I Made Londen B., ME.  
NIP. 195811061986011002 (Penguji I)
3. Hendro Nurhadi, Dipl. Ing., PhD.  
NIP. 197511202002121002 (Penguji II)
4. Dinny Harnany, ST, MSc.  
NIP. 2100201405001 (Penguji III)



**SURABAYA  
JANUARI, 2017**

*(Halaman ini sengaja dikosongkan)*

# PEMODELAN SISTEM *TURRET GUN* DENGAN METODE JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

**Nama** : Agung Kartika Fibrianto  
**NRP** : 2112100062  
**Jurusan** : Teknik Mesin FTI-ITS  
**Dosen Pembimbing** : Arif Wahjudi, S.T., M.T., Ph.D.

## ABSTRAK

Saat ini, senjata otomatis tengah dikembangkan untuk bisa menggantikan tugas manusia dalam menjaga kedaulatan negara. Penelitian tentang pemodelan sistem *turret gun* senjata otomatis penting untuk menunjang penelitian-penelitian berikutnya. Penelitian sebelumnya telah memodelkan *turret gun* dengan metode penurunan persamaan matematis. Akan tetapi, dengan metode tersebut, sistem harus disederhanakan dan persamaannya diselesaikan dengan tepat. Di era yang modern ini, metode pemodelan sistem telah berkembang dengan memanfaatkan sistem syaraf manusia atau biasa disebut Jaringan Syaraf Tiruan (JST). Dengan metode ini, sistem tidak perlu disederhanakan dan persamaannya tidak perlu diselesaikan. Maka dari itu, pada penelitian ini, sistem *turret gun* dimodelkan dengan JST dan kemudian akan dibandingkan dengan pemodelan dari metode penurunan persamaan matematis.

Pemodelan sistem *turret gun* metode JST dimulai dengan pengambilan data dari model referensi. Kemudian model JST dibangun dengan menentukan *layer* dan *neuron*. JST tersebut dilatih untuk memperoleh *weight* dan *bias*. Jika *mean square error* dari JST lebih kecil dari  $10^{-3}$ , maka model JST diverifikasi dengan membandingkan hasil keluaran model JST dengan model penurunan persamaan matematis. Jika error lebih dari 2 % maka model JST disusun ulang dengan merubah *layer* dan *neuron*.

Hasil yang diperoleh pada penelitian (tugas akhir) ini adalah suatu pemodelan sistem *turret gun* dengan metode Jaringan Syaraf Tiruan, yang mempunyai 4 *hidden layer* dan 1 *output layer*. Masing-masing *hidden layer* mempunyai 20 *neuron* dan *ouput layer* mempunyai 3 *neuron*, dengan performa Mean Square Error sebesar 0,000684. Output dari model jaringan syaraf tiruan ini dibandingkan dengan respon model yang diturunkan dari persamaan matematis mempunyai error 1,803 %. Model jaringan syaraf tiruan ini mempunyai respon dengan tendensi yang baik terhadap respon model penurunan persamaan matematis ketika diberi input yang konstan. Pada input yang berubah-ubah, respon model jaringan syaraf tiruan tidak bisa mengikuti model penurunan persamaan matematis. Hal ini disebabkan karena model jaringan syaraf tiruan tidak bisa memperhitungkan output sebelumnya.

**Kata Kunci : Pemodelan Sistem, *Turret Gun*, Jaringan Syaraf Tiruan**



# ***TURRET GUN SYSTEM MODELLING WITH NEURAL NETWORK BACKPROPAGATION METHODE***

**Name** : Agung Kartika Fibrianto  
**Student ID** : 2112100062  
**Department** : Teknik Mesin FTI-ITS  
**Academic Supervisor** : Arif Wahjudi, S.T., M.T., Ph.D.

## ***ABSTRACT***

*Nowdays, automatic weapons is developed to replace the human duty for maintain the state sovereignty. Research about automatic turret gun modelling is very important to support the following research. Previews research has modeled turret gun system with derivatif mathematical equation methode. However, on this methode, system must be simplified and the equation must be solved. In this modern era, sytem modelling methode evolve with utilize the human neural system or Artificial Neural Network (ANN). On this methode, the system didn't need to simplified and the complicated equation didn't need to solved. Thus in this research, the turret gun system was modelled with ANN and then compared with the matematical derivative equation methode model.*

*Turret gun system modelling was started with taking data from the reference model. Then, the ANN model was build by determine the layer and neuron. This ANN was trained to gain the wight and bias for each neuron. If the Mean Square Error (MSE) from the ANN was below than 10-3, so the ANN model can be verified by comparing the output from ANN model and output from matematical model. If the error between them was more than 2%, so the strukture of ANN must be rearranged.*

*The result from this research was a model of turret gun system with ANN methode, that contained 4 hidden layers and 1 output layer. For each hidden layer contained 20 neurons and output layer contained 3 neurons, with performed 0,000684 in*

*MSE. Outputs from ANN model were compared with outputs from mathematical model and this comparison had 1,803% error. Respond from ANN model had same tendency with respond from mathematical model at the constant input. However, if the inputs were permuted, respond from ANN model cann't follow the respond from mathematical model. This problem appeared because ANN model didn't calculate the previews output.*

***Keyword : System Identification, turre gun, neural network***

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas berkat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan laporan tugas akhir yang berjudul “Pemodelan Sistem *Turret gun* dengan metode Jaringan Syaraf Tiruan *Backpropagation*”. Laporan tugas akhir ini merupakan syarat untuk memperoleh gelar Sarjana Teknik (S.T.) di Jurusan Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan Laporan Tugas Akhir ini, penulis memperoleh banyak dukungan dan bantuan. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan keluarga penulis yang senantiasa mendukung segala aktivitas dan selalu mendoakan penulis selama proses kuliah dan pengerjaan tugas akhir ini.
2. Bapak Arif Wahjudi, ST., MT., Ph.D. selaku dosen pembimbing atas bimbingan, kesabaran, ketulusan dan totalitas kepada penulis selama proses pengerjaan tugas akhir ini.
3. Bapak Hendro Nurhadi, Dipl-Ing., Ph.D. , Bapak Prof. Dr. Ing. I Made Londen Batan, M.Eng., dan Ibu Dinny Harnany selaku dosen penguji atas wawasan, bimbingan , saran dan kritik yang penting dalam proses penyelesaian tugas akhir ini.
4. Bapak Ir. Bambang Pramujari, M.Eng., Ph.D. selaku ketua jurusan Teknik Mesin, Bapak Prof. Dr. Ir. Djatmiko Ichسانی, M.Eng. selaku dosen wali dan seluruh dosen Jurusan Teknik Mesin yang telah memberikan ilmu dan mendidik penulis selama proses perkuliahan.
5. Seluruh tenaga kependidikan dan karyawan Jurusan Teknik Mesin yang telah membantu penulis selama proses kuliah dan pengerjaan tugas akhir.
6. Teman-teman seperjuangan Lab P3: Deris, Febri, Wahyu, Candra, Tubagus, Tedi dan seluruh penghuni atas segala

dukungan, semangat, waktu luang, dan kebersamaan selama proses pengerjaan tugas akhir.

7. Seluruh teman-teman Mahasiswa Teknik Mesin Angkatan 2012 (M55) atas bantuan dan kebersamaan selama kuliah.
8. Serta semua pihak yang telah memberikan bantuan dan dukungan yang tidak bisa penulis sebutkan satu-persatu pada halaman ini.

Penulis menyadari bahwa laporan tugas akhir ini masih banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran demi perbaikan dan penyempurnaan tugas akhir ini. Semoga tulisan ini dapat bermanfaat bagi semua. Terima kasih.

Surabaya, Januari 2017

Penulis

# DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	v
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	5
2.1. Penelitian Terdahulu.....	5
2.2. Model Turret Gun Metode Penurunan Persamaan Matematis .....	6
2.3. Pemodelan Sistem .....	8
2.4. Jaringan Syaraf Tiruan .....	11
2.5. Jaringan Syaraf Tiruan Backpropagation .....	17
2.6. Jaringan Syaraf Tiruan dalam Pemodelan Sistem .....	18
<b>BAB III METODE PENELITIAN</b> .....	21
3.1. Diagram Alir Penelitian.....	21
3.2. Langkah-Langkah Penelitian.....	22
3.3. Diagram Alir Penentuan Jaringan Syaraf Tiruan .....	24
3.4. Langkah-Langkah Pelatihan Jaringan Syaraf Tiruan .....	25
<b>BAB IV PENGAMBILAN DATA DAN PELATIHAN     JARINGAN</b> .....	27
4.1. Pengambilan Data Training.....	27
4.2. Pelatihan Jaringan Syaraf Tiruan.....	29

**BAB V VERIFIKASI DAN UJI COBA RESPON MODEL .37**  
5.1. Verifikasi Model Sistem.....37  
5.2. Pengujian Respon Model.....39  
**BAB VI KESIMPULAN DAN SARAN.....49**  
6.1. Kesimpulan.....49  
6.2. Saran.....49

**DAFTAR PUSTAKA**  
**LAMPIRAN**  
**BIODATA PENULIS**

## DAFTAR GAMBAR

Gambar 2.1. Koordinat dari sistem senjata otomatis dua sumbu putar [1].....	5
Gambar 2.2. Diagram Simulink Model Penurunan Persamaan Matematis [1]. .....	8
Gambar 2.3. Strktur dasar jaringan syarf biologis dianalogikan sebagai struktur sederhana jaringan syaraf tiruan [6].....	12
Gambar 2.4. Neuron dengan satu input.....	14
Gambar 2.5. Fungsi Aktivasi (a) Linier, (b) Logsig, dan (c) Tansig [8]. .....	14
Gambar 2.6. Neuron dengan R input [8]. .....	15
Gambar 2.7. Jaringan yang memiliki neuron sejumlah S, input sejumlah R dalam satu layer[8]. .....	16
Gambar 2.8. Jaringan dengan tiga layer[8]. .....	16
Gambar 2.9. Blok diagram pemodelan sistem dengan jaringan syaraf tiruan[8] .....	18
Gambar 3.1. Diagram Alir Penelitian.....	22
Gambar 3.2. Diagram Alir Penentuan Model JST .....	24
Gambar 4.1. Diagram Simulink untuk Pengambilan Data Training. ....	27
Gambar 4.2. Tampilan Neural Network Tool. ....	30
Gambar 4.3. Tampilan untuk membuat Network Baru. ....	31
Gambar 4.4. Struktur Jaringan network28 .....	36
Gambar 4.5. Diagram Simulink Perbandingan Respon Model ...	39
Gambar 4.6. Grafik Respon torsi azimuth 4 Nm dan torsi Elevasi 2 Nm.....	41
Gambar 4.7. Grafik Respon torsi azimuth 2 Nm dan torsi elevasi 4 Nm.....	43
Gambar 4.8. Grafik Respon torsi azimuth 6 Nm dan torsi elevasi 6 Nm.....	45
Gambar 4.9. Grafik perbandingan dengan input berubah-ubah. .	47

*(Halaman ini sengaja dikosongkan)*



## DAFTAR TABEL

Tabel 4.1. Data Training.....	28
Tabel 4.2. Pelatihan dengan variasi fungsi aktivasi. ....	32
Tabel 4.3. Pelatihan dengan variasi jumlah hidden layer dan jumlah neuron tiap hidden layer.....	34
Tabel 4.5. Perhitungan Mean Absolute Error.....	38
Tabel 1. Weight dari input layer ke layer 1 .....	53
Tabel 2. Weight dari layer 1 ke layer 2 .....	54
Tabel 3. Weight dari layer 2 ke layer 3 .....	55
Tabel 4. Weight dari layer 3 ke layer 4 .....	57
Tabel 5. Weight dari layer 4 ke Output layer .....	58
Tabel 6. Bias untuk layer 1.....	61
Tabel 7. Bias untuk layer 2.....	62
Tabel 8. Bias untuk Layer 3 .....	63
Tabel 9. Bias untuk Layer 4 .....	64
Tabel 10. Bias untuk Layer 5 .....	65

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Saat ini, senjata otomatis tengah dikembangkan untuk bisa menggantikan tugas manusia dalam menjaga kedaulatan negara. Senjata otomatis ini diharapkan bisa mengenali target dengan tepat dan membidik dengan cepat dan tepat. Pengembangan dari senjata otomatis tersebut meliputi rancang bangun senjata, rancangan pengenalan target, rancangan pemodelan sistem, dan rancangan sistem kendali. Penelitian tentang pemodelan sistem *turret gun* senjata otomatis ini penting untuk menunjang penelitian-penelitian selanjutnya.

Penelitian mengenai pemodelan sistem *turret gun* telah dilakukan beberapa kali, salah satunya adalah pemodelan sistem dengan menurunkan persamaan matematis [1]. Dalam penelitian tersebut, *turret gun* dimodelkan dengan menurunkan persamaan dinamis dasar secara kompleks. Pemodelan yang dilakukan secara matematis tersebut memiliki kekurangan antara lain perlu menyederhanakan sistem, perlu menjadikan sistem ke dalam persamaan matematis dan perlu menurunkan serta menyelesaikan persamaan matematis yang rumit dan kompleks.

Di era yang modern ini, metode pemodelan sistem telah berkembang dengan memanfaatkan sistem syaraf manusia atau biasa disebut Jaringan Syaraf Tiruan (JST), dalam bahasa inggris *Artificial Neural Network* (ANN). Sebagai contoh, penelitian referensi [2] menggunakan jaringan syaraf tiruan *recurrent* untuk memodelkan motor DC. Dengan menggunakan JST, pemodelan suatu sistem tidak perlu menyederhanaan sistem dan menurunkan serta menyelesaikan persamaan matematis, tetapi perlu mengetahui input dan output dari sistem yang akan dimodelkan.

Oleh karena uraian tersebut, sistem *turret gun* akan dimodelkan dengan metode jaringan syaraf tiruan pada penelitian (tugas akhir) ini. Model JST yang akan dibuat menggunakan jenis

*backpropagation*. Penelitian ini juga akan membandingkan hasil pemodelan jaringan syaraf tiruan dengan pemodelan dengan metode penurunan persamaan matematis.

### **1.2. Rumusan Masalah**

Rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana memodelkan sistem *turret gun* dengan menggunakan metode Jaringan Syaraf Tiruan *backpropagation* ?
2. Bagaimana performa model Jaringan Syaraf Tiruan yang telah dibuat dibandingkan dengan model dari metode penurunan persamaan matematis ?

### **1.3. Batasan Masalah**

Batasan masalah yang digunakan dalam tugas akhir ini adalah sebagai berikut:

1. Rancang bangun *turret gun* tidak dibahas.
2. Tidak ada gangguan yang masuk dalam sistem *turret gun*.
3. Parameter *weight* dan *bias* ditentukan melalui pelatihan jaringan syaraf tiruan.
4. Jumlah *hidden layer* jaringan syaraf tiruan ditentukan dengan cara trial and error.
5. Pengambilan data input dan output mengacu pada model penurunan persamaan matematis.

### **1.4. Tujuan Penelitian**

Tujuan penelitian dalam tugas akhir ini adalah sebagai berikut:

1. Memodelkan sistem *turret gun* dengan menggunakan metode Jaringan Syaraf Tiruan *backpropagation*.
2. Mengetahui performa model Jaringan Syaraf Tiruan yang telah dibuat dibandingkan dengan model dari metode penurunan persamaan matematis.

### **1.5. Manfaat Penelitian**

Penelitian ini diharapkan memberi manfaat sebagai berikut:

1. Dapat menambah wawasan terkait jaringan syaraf tiruan dalam penggunaannya dalam pengenalan sistem.
2. Sebagai referensi bagi penelitian selanjutnya dalam sistem kendali turret gun.

*(Halaman ini sengaja dikosongkan)*

## BAB II

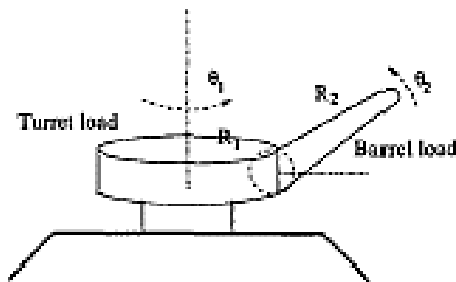
### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Dewasa ini, *turret gun* otomatis banyak dikembangkan karena banyaknya kebutuhan senjata dengan teknologi otomatis. Penelitian-penelitian yang telah dilakukan meliputi pengenalan target secara otomatis, rancang bangun prototipe pelontar peluru otomatis, dan pemodelan serta perancangan sistem kendali penggerak senjata otomatis.

Salah satu penelitian tentang pengenalan target otomatis yaitu, kita bisa menggunakan dua kamera untuk memperoleh koordinat ruang suatu benda yang kita deteksi. Caranya yaitu dengan kamera menangkap gambar secara *real time*, kemudian dari gambar tersebut kita mencari nilai tengah benda disetiap gambar. Hasil titik tengah tersebut dihitung dengan suatu persamaan dan kemudian di transformasikan ke koordinat sebenarnya dengan nilai intrinsik kamera yang diperoleh [3].

Penelitian selanjutnya yaitu penelitian yang membahas tentang pemodelan dan kendali sistem *turret gun*. Pada penelitian ini, pemodelan sistem dilakukan dengan mencari persamaan matematis sistem. Pemodelan dari sistem mengacu pada gambar 2.1. berikut ini.



Gambar 2.1. Koordinat dari sistem senjata otomatis dua sumbu putar [1].

Pada gambar 2.1, senjata otomatis dibagi menjadi 2 bagian yaitu *turret* dan *gun*. Setiap bagian digerakkan oleh motor dan memiliki parameter massa, radius dan posisi angular. Dalam mencari persamaan matematisnya, peneliti menurunkan dari persamaan dinamis dasar yang kemudian dibuat menjadi suatu bentuk *state space*. Kemudian, *State space* ini dirubah menjadi bentuk blok Simulink yang bisa disimulasikan. Peneliti pada penelitian ini membandingkan hasil respon dari beberapa sistem kendali [1].

**2.2. Model Turret Gun Metode Penurunan Persamaan Matematis**

*Turret gun* bisa diwujudkan sebagai robot dengan dua sumbu putar. *Turret gun* secara umum mempunyai dua bagian utama yaitu landasan putar (*turret*) dan lengan ayun (*gun*). Massa dari *turret* disimbolkan dengan  $m_1$ , massa dari *gun* disimbolkan dengan  $m_2$ ,  $R_1$  adalah jari-jari *turret*,  $R_2$  adalah panjang dari *gun*,  $\theta_1$  adalah sudut angular dari *turret*, dan  $\theta_2$  adalah sudut angular dari *gun* seperti terlihat pada gambar 2.1. [1].

Persamaan dinamis dari *turret gun* diturunkan dengan metode La-grangian. Persamaan dinamis dari sistem dapat dilihat pada persamaan (1).

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \dots\dots\dots(1)$$

Dimana  $\tau$  adalah torsi,  $D(\theta)$  adalah matriks inersia,  $C(\theta, \dot{\theta})$  adalah gaya *coriolis* dan gaya sentrifugal, dan  $G(\theta)$  adalah gaya gravitasi. Persamaan yang lebih jelas bisa dilihat pada persamaan berikut.

$$D(\theta) = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix} \dots\dots\dots(2)$$

$$C(\theta, \dot{\theta}) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \dots\dots\dots(3)$$

$$G(\theta) = \left( 0, \frac{1}{2} m_2 g R_2 \cos(\theta_2) \right)^T \dots\dots\dots(4)$$



Dimana

$$D_{11} = \frac{1}{2}m_1R_1^2 + m_2R_1^2 + m_2R_1R_2\cos\theta_2 + \frac{1}{3}m_2R_2^2\cos^2\theta_2$$

$$D_{22} = \frac{1}{3}m_2R_2^2$$

$$C_{11} = -m_2R_1R_2\cos\theta_2\dot{\theta}_2$$

$$C_{12} = \frac{1}{3}m_2R_2^2\sin(2\theta_2)\dot{\theta}_1$$

$$C_{21} = \left[ \frac{1}{2}m_2R_1R_2\sin\theta_2 + \frac{1}{6}m_2R_2^2\sin(2\theta_2) \right] \dot{\theta}_1$$

$$C_{22} = 0$$

Dari persamaan dimanis (1), kita dapat menurunkan persamaan untuk menghitung torsi untuk setiap komponen.

$$\tau_1 = D_{11}\ddot{\theta}_1 + C_{11}\dot{\theta}_1 + C_{12}\dot{\theta}_2$$

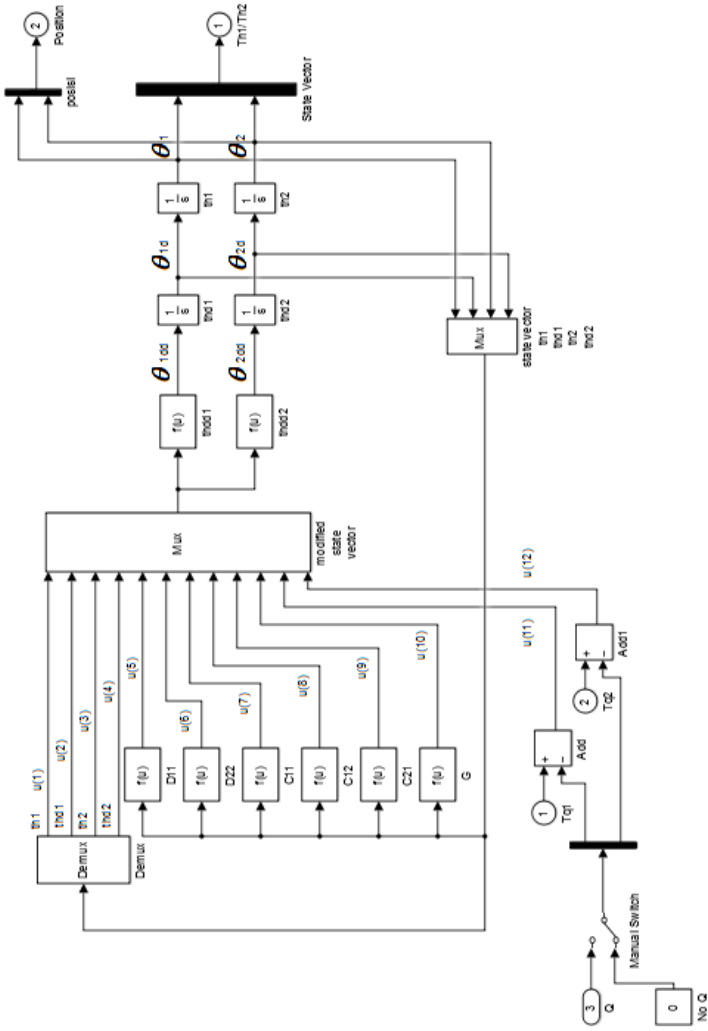
$$\tau_2 = D_{22}\ddot{\theta}_2 + C_{21}\dot{\theta}_1 + C_{22}\dot{\theta}_2 + G(\theta)$$

Untuk memperoleh percepatan, kecepatan dan posisi kita dapat merubah ke persamaan berikut.

$$\ddot{\theta}_1 = \frac{\tau_1 - C_{11}\dot{\theta}_1 - C_{12}\dot{\theta}_2}{D_{11}} \dots\dots\dots(5)$$

$$\ddot{\theta}_2 = \frac{\tau_2 - C_{21}\dot{\theta}_1 - C_{22}\dot{\theta}_2 - G(\theta)}{D_{22}} \dots\dots\dots(6)$$

Persamaan (5) dan (6) dibuat dalam bentuk diagram simulink seperti gambar 2.2



Gambar 2.2. Diagram Simulink Model Penurunan Persamaan Matematis [1]

### 2.3. Pemodelan Sistem

Model adalah suatu representasi sederhana dari sebuah sistem (atau proses atau teori), bukan sistem itu sendiri. Suatu model tidak harus memiliki seluruh atribut yang ada di dalam sistem, tetapi model tersebut disederhanakan, dikontrol, digeneralisasi, atau diidealkan. Untuk sebuah model yang akan digunakan, seluruh sifat-sifat relevannya harus ditetapkan dalam suatu cara yang praktis, dinyatakan dalam suatu set deskripsi terbatas yang masuk akal (*reasonably*). Sebuah model harus divalidasi. Setelah divalidasi, sebuah model dapat digunakan untuk menyelidiki dan memprediksi perilaku-perilaku (sifat) sistem, atau menjawab "*what if questions*" untuk mempertajam pemahaman, pelatihan, prediksi, dan evaluasi alternatif [4].

Secara umum model dapat dikategorikan menjadi 3 yaitu ikonik (model fisik), model analog (model diagramatik), dan model simbolik (model matematik) [5]. Penjabarannya adalah sebagai berikut :

1. **Model ikonik** ini mempunyai karakteristik yang sama dengan hal yang diwakilinya, dan terutama amat sesuai untuk menerangkan kejadian pada waktu yang spesifik.
2. **Model analog atau model diagramatik** dapat digunakan untuk mewakili situasi dinamik, yaitu keadaan yang berubah menurut waktu. Model ini lebih sering digunakan daripada model ikonik karena kemampuannya untuk mengetengahkan karakteristik dari kejadian yang dikaji.
3. **Model Simbolik atau model matematis** merupakan salah satu bentuk model yang paling banyak digunakan. Pada hakekatnya, ilmu sistem memusatkan perhatian pada model simbolik sebagai perwakilan dari realita yang dikaji. Format model simbolik dapat berupa bentuk angka, simbol dan rumus. Jenis model simbolik yang umum dipakai adalah suatu persamaan .

Pemodelan sistem mempunyai 3 tahapan besar yang menjadi acuan dalam menentukan langkah [5]. Tahap-tahap tersebut meliputi :

### **1. Tahap Seleksi Konsep**

Langkah awal dari pemodelan adalah melakukan seleksi alternatif. Seleksi ini dilakukan untuk menentukan alternatif-alternatif mana yang bermanfaat dan bernilai cukup besar untuk dilakukan pemodelan.

### **2. Tahap Pemodelan**

Sebagai langkah awal dari pemodelan adalah menetapkan jenis model yang akan digunakan, sejalan dengan tujuan dan karakteristik sistem. Setelah itu, aktivitas pemodelan terpusat pada pembentukan model yang realistik. Dalam hal ini ada dua cara pendekatan untuk membentuk suatu model, yaitu:

#### **a. Pendekatan Kotak Hitam (Black Box)**

Metode ini digunakan untuk melakukan identifikasi model sistem dari data yang menggambarkan perilaku masa lalu dari sistem. Melalui berbagai teknik statistik dan matematik, maka model yang paling cocok dengan data operasional dapat diturunkan. Metoda ini tidak banyak berguna pada perancangan sistem yang kenyataannya belum ada, dimana tujuan sistem masih berupa konsep.

#### **b. Pendekatan Struktural**

Metode ini mempelajari secara teliti struktur sistem untuk menentukan komponen basis sistem serta keterkaitannya. Melalui pemodelan karakteristik dari komponen sistem serta kendala-kendala yang disebabkan oleh adanya keterkaitan antara komponen, maka model sistem keseluruhan dapat disusun secara berantai. Pendekatan struktural ini banyak digunakan dalam rancang-bangun dan pengendalian sistem fisik dan non fisik.

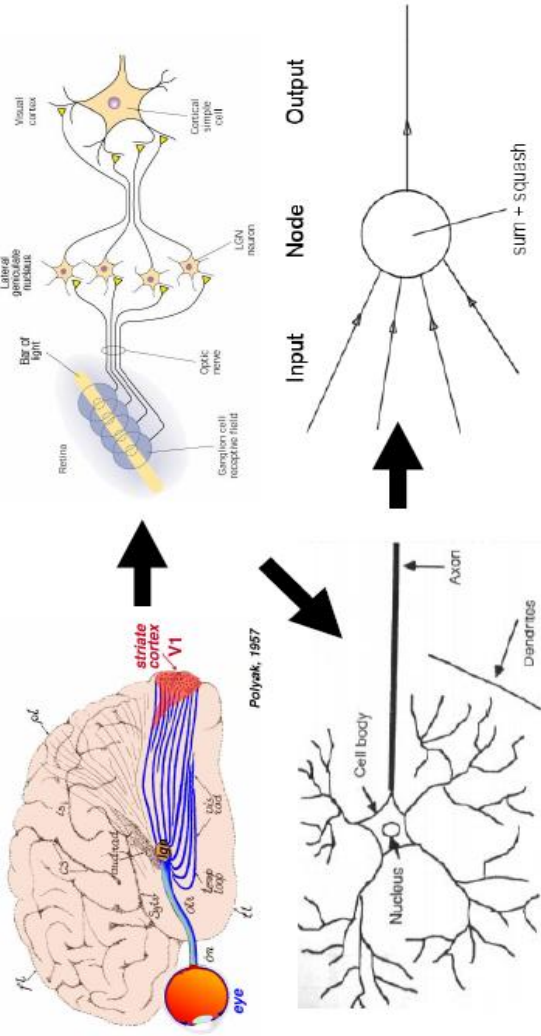
### **3. Tahap Validasi**

Validasi model pada hakekatnya merupakan usaha untuk menyimpulkan apakah model sistem tersebut merupakan perwakilan yang sah dari realitas yang dikaji sehingga dapat dihasilkan kesimpulan yang meyakinkan. Validasi merupakan proses iteratif yang berupa pengujian berturut-turut sebagai proses penyempurnaan model. Umumnya validasi dimulai dengan uji sederhana seperti pengamatan atas :

- a. tanda aljabar (sign)
- b. kejangkatan dari besaran (order of magnitude)
- c. format respon (linear, eksponensial, logaritmik)
- d. arah perubahan peubah apabila input atau parameter diganti-ganti
- e. nilai batas peubah sesuai dengan nilai batas parameter sistem [5].

### **2.4. Jaringan Syaraf Tiruan**

Jaringan Syaraf Tiruan (JST) adalah suatu sistem pemrosesan informasi yang memiliki karakteristik seperti dengan jaringan syaraf biologi yang dimiliki manusia seperti ditunjukkan pada gambar 2.3.



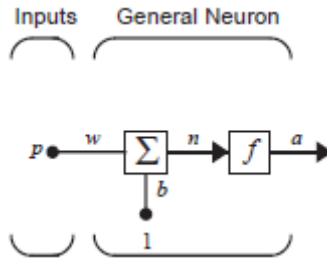
Gambar 2.3. Struktur dasar jaringan syaraf biologis dianalogikan sebagai struktur sederhana jaringan syaraf tiruan [6].

JST dibentuk sebagai generalisasi model matematika dari syaraf biologi, dengan mengasumsikan bahwa :

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
2. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
4. Dalam menentukan output, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

Jaringan syaraf tiruan sederhana pertama kali diperkenalkan oleh Mc Culloch dan Pitts di tahun 1943. Mc Culloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan Mc Culloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*. Tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan jaringan yang disebut *Perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya. Widrow dan Hoff (1960) mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan yang dikenal sebagai aturan delta (atau sering disebut kuadrat rata-rata kecil). Aturan ini akan mengubah bobot perceptron apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Yang dilakukan oleh peneliti terdahulu hanya menggunakan *layer* tunggal (*single layer*). Rumelhart (1986) mengembangkan *perceptron* menjadi *Backpropagation* yang memungkinkan jaringan diproses melalui beberapa *layer*. Selain itu, model jaringan syaraf tiruan yang lain juga dikembangkan oleh Kohonen (1972) dan Hopfield (1982). Pengembangan yang ramai dibicarakan sejak tahun 1990-an adalah aplikasi model-model jaringan syaraf tiruan untuk menyelesaikan masalah di dunia nyata [7].

Jaringan syaraf tiruan mula-mula terbentuk dari struktur yang sederhana yaitu struktur dengan neuron seperti pada gambar 2.4. Pada gambar tersebut,  $p$  adalah input skalar dikalikan dengan *weight* ( $w$ ) dan ditambahkan dengan *bias* ( $b$ ). Perkalian dan penjumlahan tadi yang dinotasikan ( $n$ ) selanjutnya akan masuk ke fungsi aktivasi ( $f$ ) sehingga diperoleh output ( $a$ ).

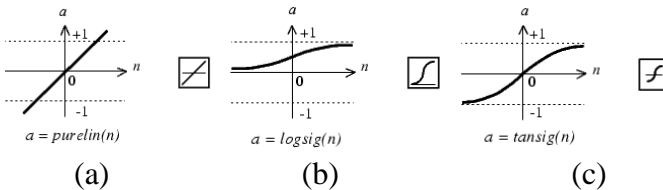


Gambar 2.4. Neuron dengan satu input

Sehingga nilai keluaran neuron dapat ditulis sebagai berikut :

$$a = f(wp + b) \dots \dots \dots (7)$$

Nilai  $w$  dan  $b$  adalah nilai parameter neuron yang dapat diatur. Parameter  $w$  dan  $b$  dapat diatur dengan learning rule sehingga hubungan input-output dan target dapat terpenuhi. Fungsi aktivasi ( $f$ ) adalah fungsi yang bisa dipilih oleh penyusun. Fungsi aktivasi yang biasa digunakan dalam jaringan syaraf tiruan adalah sebagai berikut :

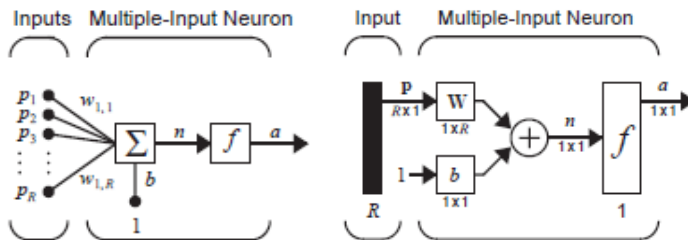


Gambar 2.5. Fungsi Aktivasi (a) Linier, (b) Logsig, dan (c) Tansig [8].



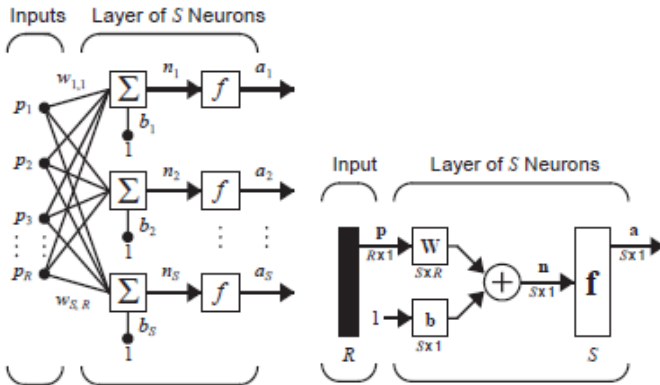
Fungsi aktivasi pada gambar 2.5. akan mengaktifkan jaringan yang telah dibuat. Fungsi aktivasi linier (a) akan membawa nilai input ke output yang sebanding. Fungsi aktivasi Logsig (b) akan membawa input ke output dengan penghitungan log-sigmoid yang outputnya 0 sampai 1 dan fungsinya  $f(net) = \frac{1}{1+e^{-net}}$ . Fungsi aktivasi Tansig (c) akan membawa input ke output dengan penghitungan tan-sigmoid yang outputnya 1 sampai -1 dan fungsinya  $f(net) = \frac{2}{1+e^{-net}} - 1$ .

Suatu neuron pastinya tidak selalu memiliki hanya 1 input. Jika input memiliki sejumlah  $R$  input, maka inputnya adalah  $p_1, p_2, \dots, p_R$  dan weight mempunyai notasi  $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ . Nilai  $n$  yang terjadi adalah  $n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$ . Struktur jaringan yang dibentuk menjadi seperti pada gambar 2.6.



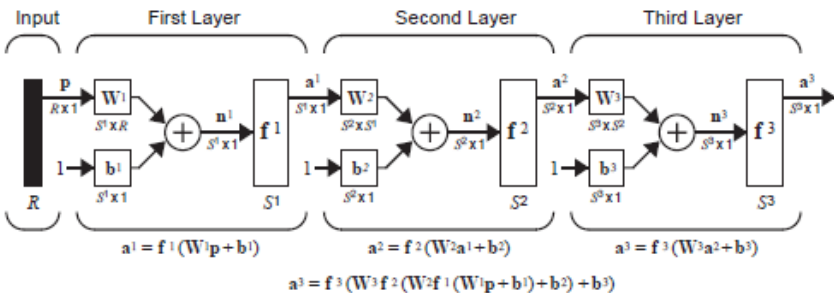
Gambar 2.6. Neuron dengan  $R$  input [8].

Biasanya satu neuron dengan banyak input belum cukup untuk memodelkan suatu sistem. Kita harus menambahkan beberapa neuron dalam satu *layer* yang akan bekerja secara paralel. Misalnya pada layer terdapat input sejumlah  $R$  buah dan kita tambahkan neuron sejumlah  $S$ . Dengan begitu strukturnya mempunyai matriks weight  $\mathbf{W}$  menjadi seperti pada gambar 2.7.



Gambar 2.7. Jaringan yang memiliki neuron sejumlah  $S$ , input sejumlah  $R$  dalam satu *layer*[8].

Untuk memaksimalkan hasil yang diperoleh, *layer* penyusun jaringan dapat ditambah beberapa *layer*. *Layer* yang langsung berhubungan dengan output disebut *output layer*, sedangkan *layer* yang lain disebut *hidden layer*. Dengan demikian, output dari layer 1 akan menjadi input pada layer 2 dan seterusnya. Setiap layer akan memiliki masing-masing matriks weight  $\mathbf{W}$  dan vektor bias  $\mathbf{b}$ . Secara notasi bisa digambarkan sebagai gambar 2.8. berikut[8].



Gambar 2.8. Jaringan dengan tiga *layer*[8].

## 2.5. Jaringan Syaraf Tiruan Backpropagation

Jaringan syaraf tiruan memiliki beberapa model yang bisa digunakan. Salah satu model yang bisa digunakan untuk pemodelan yang kompleks adalah *backpropagation*[7]. Algoritma *backpropagation* ini akan menghasilkan unjuk kerja yang lebih baik. Hal ini berarti bahwa *wegiht* dan *bias* akan mendekati nilai yang seharusnya. Kelebihan dari model *backpropagation* ini adalah kemampuannya untuk belajar (adaptif) dan mempunyai toleransi kesalahan (*fault tolerance*). Dengan begitu kita dapat mewujudkan sistem yang tahan terhadap kerusakan (*robust*) dan konsisten bekerja dengan baik.

Metode algoritma *backpropagation* ini merupakan *supervised training* dimana untuk tiap pola input terdapat pasangan target output untuk masing-masing pola input. Ide dasarnya dapat dijelaskan dengan pola hubungan sederhana yaitu : jika output memberikan hasil yang tidak sesuai dengan target yang diinginkan, maka pembobot akan dikoreksi agar *error*nya dapat diperkecil dan selanjutnya respon jaringan diharapkan akan lebih mendekati harga yang sesuai.

Pelatihan *backpropagation* meliputi 3 tahapan dalam proses pelatihan yaitu : propagasi maju, propagasi mundur dan perubahan bobot. [7]

- **Fase 1 : Propagasi maju (*feedforward*)**

Pada fase ini pola input dihitung maju mulai dari input *layer* sampai output layer menggunakan fungsi aktivasi. Kemudian diperoleh keluaran jaringan yang akan dibandingkan dengan target yang harus dicapai. Selisih antara keluaran dan target dihitung dan menjadi pertimbangan untuk fase selanjutnya.

- **Fase 2 : Propagasi mundur (*backpropagation*)**

Selisih yang telah diperoleh dari fase 1 merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur dengan menghitung faktor delta ( $\delta$ ). Faktor tersebut digunakan untuk mendistribusikan kesalahan pada setiap *neuron* yang berhubungan dengan keluaran jaringan.

Dengan demikian diperoleh faktor delta ( $\delta$ ) pada setiap *neuron* yang berhubungan dengan keluaran jaringan. Faktor ini akan digunakan sebagai dasar untuk mengubah bobot.

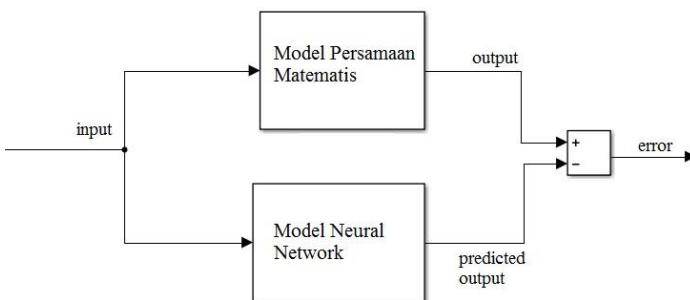
- **Fase 3 : Perubahan Bobot**

Faktor delta ( $\delta$ ) pada masing-masing neuron akan dijadikan dasar seberapa besar atau kecilnya perubahan bobot yang terjadi. Perubahan bobot untuk satu garis keluaran dilakukan bersamaan.

Dari fase 1 sampai 3 akan dilakukan secara berulang-ulang sampai kondisi penghentian dipenuhi. Kondisi penghentian yang biasa dipakai adalah jumlah iterasi (*epoch*) dan error. Jika iterasi sudah mencapai batas maksimal, maka pelatihan akan dihentikan, atau jika error sudah lebih kecil dari yang ditentukan maka pelatihan akan dihentikan.

## 2.6. Jaringan Syaraf Tiruan dalam Pemodelan Sistem

Pemodelan dengan menggunakan jaringan syaraf tiruan bisa disebut dengan *black box modelling*. Dengan metode ini kita tidak perlu mengetahui bagaimana fungsi yang terjadi pada sistem yang akan dimodelkan. Akan tetapi kita harus mempunyai sekumpulan input dan output dari sistem yang akan dimodelkan.



Gambar 2.9. Blok diagram pemodelan sistem dengan jaringan syaraf tiruan[8]

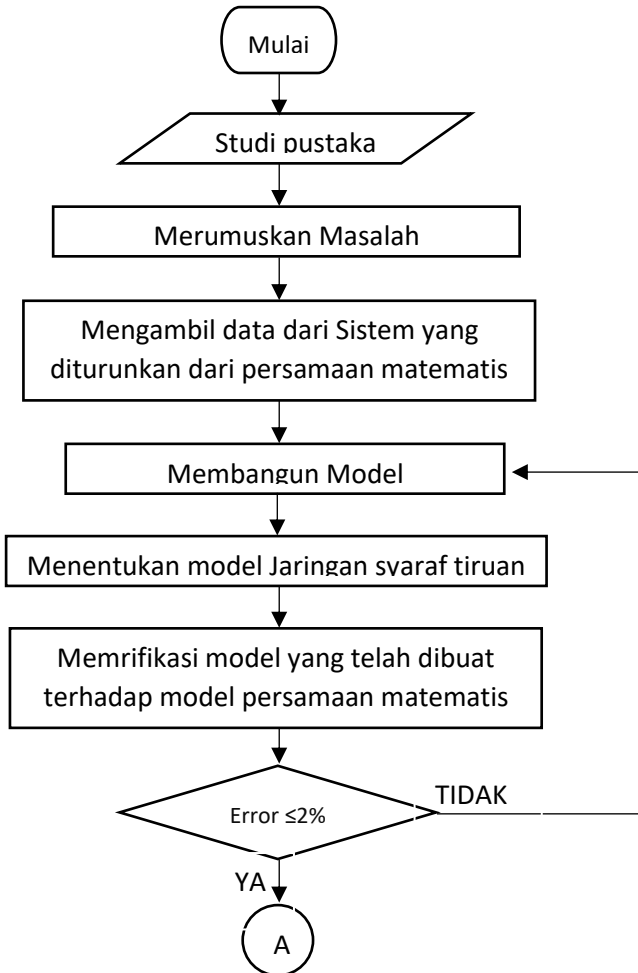
Pada gambar 2.9., sistem digambarkan sebagai *model referensi* akan menghasilkan output tertentu ketika dimasuki input tertentu. Input yang masuk pada sistem akan masuk juga pada blok jaringan syaraf tiruan dan akan akan menghasilkan *predicted output*. *Predicted output* ini akan dibandingkan dengan output keluaran dari sistem. Hasil perbandingan ini adalah *error* yang kemudian akan digunakan untuk membuat blok JST yang lebih akurat. Error yang diperoleh akan digunakan blok jaringan syaraf tiruan untuk merubah *weight* dan *bias* nya sehingga mendekati target output dari sistem dan *error* menjadi lebih kecil atau bisa sama dengan nol.

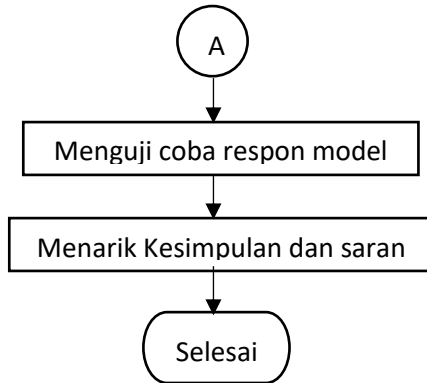
*(Halaman ini sengaja dikosongkan)*

## BAB III METODE PENELITIAN

### 3.1. Diagram Alir Penelitian

Penelitian pada tugas akhir ini dilaksanakan dengan mengikuti diagram alir sebagai berikut.





Gambar 3.1. Diagram Alir Penelitian

### 3.2. Langkah-Langkah Penelitian

Berdasarkan diagram alir tersebut, langkah-langkah penelitian secara detail dijabarkan sebagai berikut:

#### 1. Studi pustaka

Tahap studi pustaka adalah tahap dimana penulis melakukan studi literatur mengenai apa dan bagaimana jaringan syaraf tiruan bekerja, selain itu juga bagaimana penelitian sebelumnya mengenai jaringan syaraf tiruan dan penelitian tentang *turret gun*. Literatur yang digunakan berupa buku, jurnal ilmiah dan makalah mengenai topik yang dibicarakan

#### 2. Merumuskan masalah

Langkah ini dilakukan untuk menentukan permasalahan yang akan dikaji dan dicari solusi terbaiknya. Pada tugas akhir ini diperoleh perumusan masalah yang akan dibahas.

#### 3. Mengambil data dari sistem yang diturunkan dari persamaan matematis

Pada tahap ini dilakukan pengambilan data masukan dan keluaran sistem. Sistem *turret gun* ini sebelumnya sudah dimodelkan sehingga pengambilan data dilakukan



dengan mensimulasikan model pada referensi [1]. Data yang diperoleh disini berupa 3 masukan yaitu torsi azimut, torsi elevasi dan waktu pengambilan data. Sedangkan data keluaran yang diperoleh yaitu sudut azimut dan sudut elevasi untuk tiap waktu keluaran.

#### **4. Membangun model JST dengan algoritma *backpropagation***

Pembuatan JST pada langkah ini dilakukan dengan bantuan perangkat lunak MATLAB R2014a pada perangkat keras laptop ASUS A43s dengan kecepatan prosesor 2,5 GHz dan memori 4 GB. Pembuatan JST ini menggunakan bantuan Neural Network Toolbox yang ada pada perangkat lunak tersebut.

#### **5. Menentukan model jaringan syaraf tiruan (JST)**

Pada langkah ini pelatihan dilakukan untuk menentukan *weight* dan *bias* yang akan digunakan pada model JST. Secara lebih lengkap akan dibahas pada diagram alir pelatihan jaringan syaraf tiruan.

#### **6. Memverifikasi model yang telah dibuat terhadap model persamaan matematis**

Model JST yang telah dibuat akan diverifikasi hasil outputnya dengan hasil output dari model persamaan matematis. Hasil yang diperoleh dari verifikasi ini adalah berupa perbedaan atau error terbesar antara hasil output model JST dengan hasil output dari model matematis. Jika error lebih besar dari 2%, maka kembali lagi ke langkah 5.

#### **7. Menguji coba respon model**

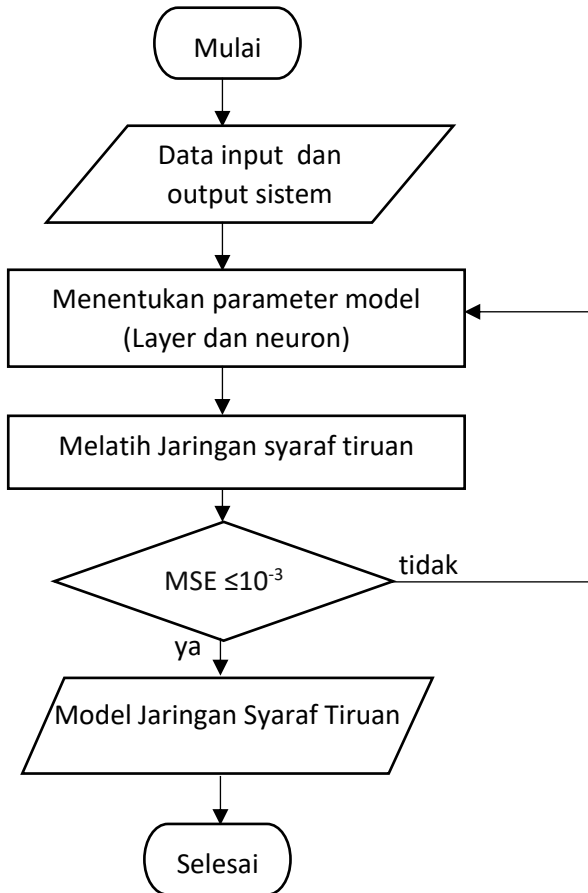
Model yang telah dibuat akan diuji coba dengan memberikan masukan berupa torsi konstan dan juga torsi yang berubah-ubah.

#### **8. Menarik kesimpulan dan saran**

Langkah terakhir adalah menarik kesimpulan dari penelitian ini yang akan menjawab dari rumusan masalah pada awal penelitian. Saran untuk penelitian selanjutnya dirumuskan agar bisa memperoleh hasil yang lebih baik.

### 3.3. Diagram Alir Penentuan Jaringan Syaraf Tiruan

Penentuan Jaringan syaraf tiruan pada penelitian ini mengikuti diagram alir sebagai berikut.



Gambar 3.2. Diagram Alir Penentuan Model JST

### **3.4. Langkah-Langkah Pelatihan Jaringan Syaraf Tiruan**

Berdasarkan diagram alir tersebut, langkah-langkah pelatihan JST secara detail dijabarkan sebagai berikut:

#### **1. Menentukan parameter model (layer dan neuron)**

Layer dan neuron untuk jaringan syaraf tiruan ditentukan dengan dimulai dari jumlah layer dan neuron kecil.

#### **2. Melatih Jaringan Syaraf Tiruan**

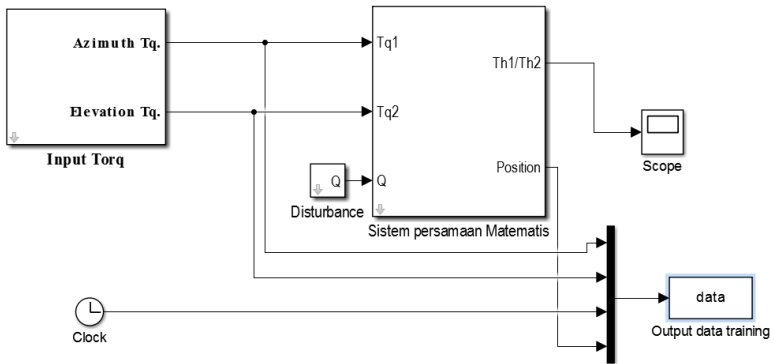
Pelatihan jaringan syaraf tiruan dilakukan dengan metode backpropagation. Jika  $mse \geq 10^{-3}$  maka akan dilakukan pelatihan lagi terus menerus. Jika perubahan mse sudah tidak signifikan dan mse masih  $\geq 10^{-3}$ , maka harus dilakukan penentuan jumlah layer dan neuron kembali.

*(Halaman ini sengaja dikosongkan)*

## BAB IV PENGAMBILAN DATA DAN PELATIHAN JARINGAN

### 4.1. Pengambilan Data *Training*

Data *training* adalah data yang akan digunakan untuk melatih jaringan syaraf tiruan supaya dapat mendekati sistem yang diinginkan. Pada penelitian ini data *training* diambil dengan cara mensimulasikan model dari penurunan persamaan matematis pada referensi [1]. Simulasi untuk memperoleh data *training* membutuhkan *software* Matlab Simulink dan diagram Simulink seperti pada gambar 4.1.



Gambar 4.1. Diagram Simulink untuk Pengambilan Data *Training*.

Pada gambar 4.1., blok “*Input torq*” adalah blok input. Input berupa torsi azimuth dan torsi elevasi dimasukkan dalam blok tersebut. Torsi azimuth dan elevasi ini akan masuk ke dalam blok ke “Sistem persamaan matematis”. Setelah itu, hasil pemrosesan masuk ke blok “Output data *training*”. Pada blok ini, hasil yang berupa sudut azimuth dan sudut elevasi keluaran dari blok sistem dan juga waktu akan dicatat dan dimasukkan pada lembar kerja *software* Matlab.

Input dari torsi azimuth dan elevasi yang dimasukkan mempunyai *range* data dari -10 Nm sampai 10 Nm. *Range* data ini kemudian dicacah dengan kenaikan setiap 2 Nm agar data training lebih akurat. Waktu yang dicatat untuk memperoleh data training ini adalah 0 sampai 9 detik dengan kenaikan 0,2 detik. Data training yang diperoleh dari simulasi ini bisa dilihat pada tabel 4.1. berikut.

Tabel 4.1. Data Training.

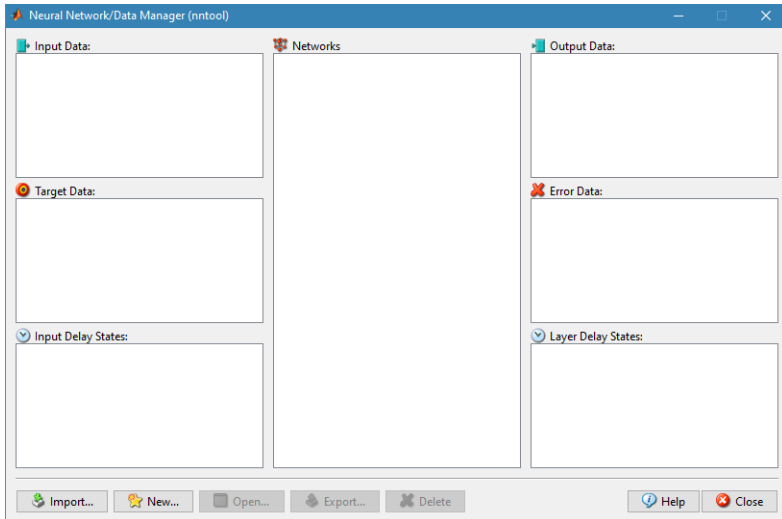
No	Torsi Azimuth (Nm)	Torsi Elevasi (Nm)	Waktu (Detik)	Sudut Azimuth (derajat)	Sudut Elevasi (derajat)	Waktu (detik)
1	10	10	0	0	1,1071	0
2	10	10	0,2	0,0429	1,7041	0,2
3	10	10	0,4	0,1577	5,9682	0,4
4	10	10	0,6	0,3633	13,114	0,6
5	10	10	0,8	0,6214	23,174	0,8
6	10	10	1	0,7476	37,855	1
7	10	10	1,2	0,8610	55,720	1,2
8	10	8	0	0	1,1071	0
9	10	8	0,2	0,0424	1,3281	0,2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
2276	-10	-8	0,4	-0,1763	-6,9646	0,4
2277	-10	-8	0,6	-0,4047	-15,058	0,6
2278	-10	-10	0	0	1,1071	0
2279	-10	-10	0,2	-0,0361	-1,7382	0,2
2280	-10	-10	0,4	-0,1681	-8,1188	0,4
2281	-10	-10	0,6	-0,3875	-17,716	0,6

Data *training* pada tabel 4.1. pada kolom 1 adalah input torsi azimuth, pada kolom 2 adalah input torsi elevasi, pada kolom 3 adalah input waktu, pada kolom 4 output sudut azimuth, pada kolom 5 output sudut elevasi, dan pada kolom 6 adalah output waktu. Jumlah data keseluruhan berjumlah 2281 pasang data.

Data *training* pada tabel 4.1. sudah disaring dengan data yang mempunyai sudut azimuth lebih dari 65 derajat dan lebih kecil dari -65 derajat tidak diikutkan pelatihan. Begitu pula dengan sudut elevasi yang lebih besar dari 65 derajat dan lebih kecil dari -25 derajat tidak diikutkan dalam pelatihan. Hal ini mengacu pada desain sistem yang hanya mempunyai range sudut azimuth -60 derajat sampai 60 derajat dan sudut elevasi -20 derajat 60 derajat. Selain hal ini, pembatasan data pelatihan ini berguna untuk memperkecil data pelatihan agar pelatihan menjadi lebih cepat. Penambahan 5 derajat pada masing-masing derajat dimaksudkan agar overshoot 5% *design criteria* pada sistem kendali dapat terpenuhi.

#### **4.2. Pelatihan Jaringan Syaraf Tiruan**

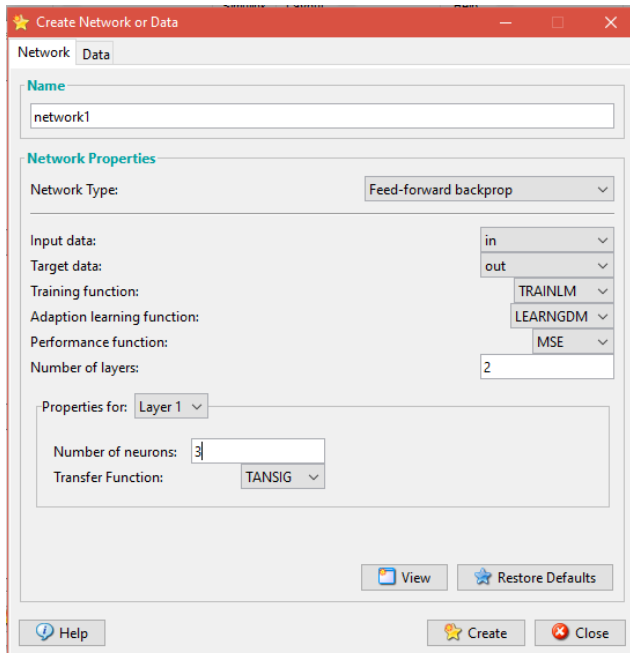
Pelatihan Jaringan Syaraf Tiruan dilakukan dengan menggunakan *software* Matlab. Di dalam *software* Matlab ini terdapat suatu fasilitas yang bisa digunakan untuk membuat suatu sistem Jaringan Syaraf tiruan yaitu *Neural Network Tool* (nntool). Kelebihan *Neural Network Tool* pada Matlab ini bisa digunakan untuk membandingkan masing-masing performa Jaringan sebelum dijadikan blok Simulink. Tampilan *Neural Network Tool* dapat dilihat pada gambar 4.2.



Gambar 4.2. Tampilan Neural Network Tool.

Pada gambar 4.2., Neural Network Tool mempunyai beberapa kolom fungsi masing-masing. Kolom kiri atas “*input data*” berfungsi untuk memasukkan data *training input*. Pada bagian kiri baris kedua terdapat kolom “*target data*” berfungsi untuk memasukkan data *training output*. Kolom yang ditengah yaitu “*networks*” berfungsi untuk melihat *Neural Network* yang telah dibuat. Jaringan atau *networks* yang akan dilatih dapat dibuat dengan meng-klik pada tombol “new pada tampilan gambar 4.2. Setelah itu akan muncul tampilan seperti pada gambar 4.3.





Gambar 4.3. Tampilan untuk membuat Network Baru.

Jaringan atau *Network* baru dibuat dengan cara mengisi nama *network* pada kolom "Name". Kemudian tipe "Feed-Foward Back Propagation" dipilih pada kolom "*Network Type*". Lalu input data dan target data dipilih dari data yang telah dimasukkan. Fungsi TRAINLM dipilih pada kolom "*Training function*". MSE dipilih pada kolom "*Performance Function*" untuk menjadikan MSE sebagai indeks performa dari Network. Jumlah layer yang sesuai dengan kombinasi dimasukkan pada kolom "*Number of layer*". Jumlah neuron pada tiap layer dan fungsi aktivasi tiap layer diisikan pada kolom "*properties for layer*". Jika semua kolom sudah terisi semua, tombol "create" diklik.

Dalam pelatihan ini, jumlah *hidden layer* dan jumlah *neuron* dalam *hidden layer* dilakukan variasi. Pemilihan kombinasi dari jumlah *neuron* pada *hidden layer* menggunakan cara yaitu : jika  $n$  adalah jumlah input, maka *neuron* yang digunakan adalah  $2n$  atau  $2n + 1$  atau 5. Selain itu pemilihan jumlah *neuron* pada *hidden layer* menggunakan angka kelipatannya jika  $mse \leq 10^{-3}$  yang diperoleh tidak juga memenuhi target ketika *training*.

Pelatihan jaringan syaraf tiruan memiliki parameter batas *training*. Pada pelatihan Jaringan Syaraf Tiruan ini menggunakan parameter batas jumlah iterasi atau *epoch*. Jumlah iterasi atau *epoch* maksimal yang digunakan pada training ini adalah 10000 kali iterasi. Jumlah iterasi atau *epoch* ini berpengaruh pada berapa kali *software* mencoba mengganti *weight* dan *bias* dari jaringan syaraf tiruan hingga tercapai performa yang diinginkan. Jika performa belum tercapai tetapi jumlah iterasi sudah tercapai, maka pelatihan akan dihentikan.

Pelatihan Jaringan syaraf tiruan dilakukan dengan mencoba dahulu dengan beberapa kondisi dimana terdapat perbedaan fungsi aktivasi. Hasil pelatihan jaringan syaraf tiruan dengan variasi fungsi aktivasi dapat dilihat pada tabel 4.2.

Tabel 4.2. Pelatihan dengan variasi fungsi aktivasi.

name	Jml. layer	neuron per layer					AF per layer					MSE
		1	2	3	4	5	1	2	3	4	5	
net1	2	3	3				tansig	tansig				10,55
net2	2	3	3				logsig	logsig				146
net3	2	6	3				tansig	tansig				3,15
net4	2	6	3				logsig	logsig				148
net5	5	7	7	7	7	3	tansig	logsig	tansig	logsig	tansig	421

Tabel 4.2. menunjukkan bahwa jaringan *network1* dan *network 3* dengan fungsi aktivasi TANSIG mempunyai MSE yang jauh lebih kecil daripada *network2* dan *network 4* yang mempunyai fungsi aktivasi LOGSIG. Selain itu, *network1* dan *network3* juga mempunyai performa lebih baik dari kombinasi fungsi aktivasi antara TANSIG dan LOGSIG pada *network5*. Dengan begitu, fungsi aktivasi yang tepat pada sistem ini adalah fungsi aktivasi TANSIG. Pelatihan dengan variasi *hidden layer* dan *neuron* akan menggunakan fungsi aktivasi TANSIG seluruhnya.

Tahap pelatihan jaringan syaraf tiruan selanjutnya dilakukan dengan memvariasikan jumlah *hidden layer* dan jumlah *neuron* dari setiap *hidden layer*. Hasil dari pelatihan tersebut dapat dilihat pada tabel 4.3.

Tabel 4.3. Pelatihan dengan variasi jumlah hidden layer dan jumlah neuron tiap hidden layer.

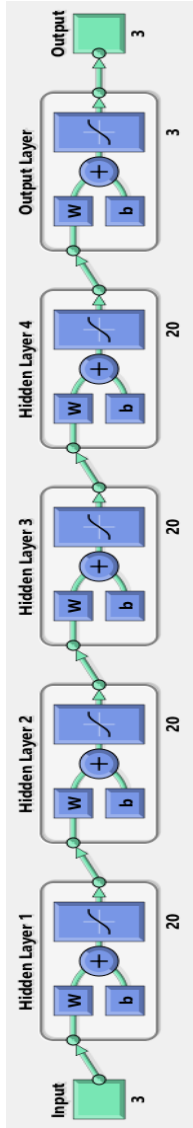
Nama	Jumlah Layer	Jumlah Neuron per Layer						MSE
		1	2	3	4	5	6	
network1	2	3	3					10,552
network2	2	6	3					3,15
network3	2	7	3					17,6955
network4	2	10	3					2,2119
network5	2	12	3					1,5896
network6	2	14	3					0,72804
network7	2	20	3					0,40738
network8	3	3	3	3				2,4402
network9	3	6	6	3				228,769
network10	3	7	7	3				0,19001
network11	3	10	10	3				0,038442
network12	3	12	12	3				0,011737
network13	3	14	14	3				0,012232
network14	3	20	20	3				0,057754
network15	4	3	3	3	3			2,5761
network16	4	6	6	6	3			2,9839
network17	4	7	7	7	3			0,081698
network18	4	10	10	10	3			0,084921
network19	4	12	12	12	3			0,40478
network20	4	14	14	14	3			0,044703
network21	4	20	20	20	3			0,004709
network22	5	3	3	3	3	3		2,6143
network23	5	6	6	6	6	3		0,012908
network24	5	7	7	7	7	3		111,286
network25	5	10	10	10	10	3		0,014401
network26	5	12	12	12	12	3		0,0031484
network27	5	14	14	14	14	3		44,1084
network28	5	20	20	20	20	3		0,000684
network29	6	3	3	3	3	3	3	2,1914
network30	6	6	6	6	6	6	3	2,0143
network31	6	7	7	7	7	7	3	0,023843
network32	6	10	10	10	10	10	3	0,040274
network33	6	12	12	12	12	12	3	9,865
network34	6	14	14	14	14	14	3	0,22671

Tabel 4.3. mempunyai beberapa kolom. Kolom ke-1 yaitu nama jaringan yang dibangun. Kolom ke-2 adalah jumlah *layer* yang digunakan pada satu jaringan yang terdiri *hidden layer* dan *output layer*. Kolom ke-3 adalah kolom jumlah *neuron* pada setiap *layer*. Kolom ke-3 ini mempunyai 6 buah sub-kolom yang mewakili *layer* yang berisi jumlah *neuron* tiap *layer*. Kolom ke-4 adalah kolom MSE yang menunjukkan performa dari setiap jaringan yang telah dibangun.

Tabel 4.3. menunjukkan bahwa semakin banyak *layer* yang digunakan maka MSE akan semakin turun. Tetapi ketika 6 *layer* digunakan, MSE kembali naik dan pelatihan memakan waktu yang lama, sehingga pengujian dihentikan hanya sampai 6 *layer*. Penggunaan jumlah *neuron* pada tiap *hidden layer* juga berpengaruh terhadap performa dari jaringan yaitu semakin banyak *neuron* pada tiap *layer* maka MSE cenderung semakin turun, tetapi waktu pelatihan semakin lama. Karena waktu pelatihan terlalu lama, maka *neuron* hanya dibatasi sampai 20 *neuron* pada tiap *layer*.

Dari tabel 4.3., pelatihan jaringan syaraf tiruan memperoleh jaringan yang mempunyai MSE kurang dari  $10^{-3}$  yaitu *network28*. Jaringan tersebut memenuhi target untuk dilakukan verifikasi terhadap model persamaan matematis. *Network28* ini mempunyai 5 *layer* yang terdiri dari 4 *hidden layer* dan 1 *output layer*. Masing-masing *hidden layer* mempunyai 20 *neuron* dan *output layer* mempunyai 3 *neuron*. Struktur jaringan *network28* dapat dilihat pada gambar 4.4.

Setiap *neuron* dalam *hidden layer* maupun *output layer* mempunyai *weight* dan *bias* masing-masing. *Weight* dan *bias* masing-masing dapat dilihat pada Lampiran 1.



Gambar 4.4. Struktur Jaringan *network28*

## BAB V

### VERIFIKASI DAN UJI COBA RESPON MODEL

#### 5.1. Verifikasi Model Sistem

Jaringan syaraf tiruan yang telah diperoleh dari pelatihan harus melalui tahap verifikasi. Verifikasi model sistem ini dilakukan dengan cara membandingkan output dari model jaringan syaraf tiruan dengan output dari model penurunan persamaan matematis pada referensi [1]. Output yang akan dibandingkan ini mempunyai input yang sama pada masing-masing model. Data output dari model penurunan persamaan matematis diperoleh dari *data training*. Data output dari model jaringan syaraf tiruan diperoleh dengan cara mensimulasikan data input dengan fasilitas dari Neural Network Tool.

Output dari model penurunan persamaan matematis dan output dari model jaringan syaraf tiruan dibandingkan dengan metode *Mean Absolute Error* (MAE). Dalam statistik, MAE adalah cara yang digunakan untuk mengukur seberapa dekat antara prediksi dan hasil sebenarnya dari suatu sistem [9]. Pada metode ini akan dicari error dengan persamaan sebagai berikut.

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \dots\dots\dots (8)$$

Pada persamaan (8), *Mean Absolute Error* (MAE) dicari dengan cara menghitung selisih antara data output dari model jaringan syaraf tiruan ( $f_i$ ) dan data output dari model penurunan persamaan matematis ( $y_i$ ). Kemudian selisih atau error dari keduanya dicari nilai absolutnya. Setelah itu semua data absolut dijumlahkan dan kemudian dibagi dengan jumlah data keseluruhan ( $n$ ). Perhitungan dari *Mean Absolute Error* dapat dilihat pada tabel 4.5.

Tabel 4.5. Perhitungan *Mean Absolute Error*

Data nomor	1	2	3	4	5	6	...	2281	Jumlah	
target $\theta$ az ( $f_i$ )	0	-0,036	-0,168	-0,388	0	-0,037	...	0,861		
target $\theta$ el ( $f_i$ )	1,107	-1,738	-8,119	-17,72	1,107	-1,434	...	55,72		
out $\theta$ az ( $y_i$ )	-0,02	-0,05	-0,154	-0,398	-0	-0,048	...	0,843		
out $\theta$ el ( $y_i$ )	1,075	-1,735	-8,158	-17,72	1,094	-1,438	...	55,84		
err $\theta$ az ( $f_i - y_i$ )	0,015	0,0138	-0,014	0,0102	0,003	0,0109	...	0,018		
err $\theta$ el ( $f_i - y_i$ )	0,032	-0,003	0,0389	0,0057	0,013	0,0033	...	-0,12		
abs err $\theta$ az $ f_i - y_i $	0,015	0,0138	0,0142	0,0102	0,003	0,0109	...	0,018		30,180
abs err $\theta$ el $ f_i - y_i $	0,032	0,0033	0,0389	0,0057	0,013	0,0033	...	0,123		52,072
Jumlah total absolut error							$\sum_{i=1}^n  f_i - y_i $	82,252		
MAE									0,01803	

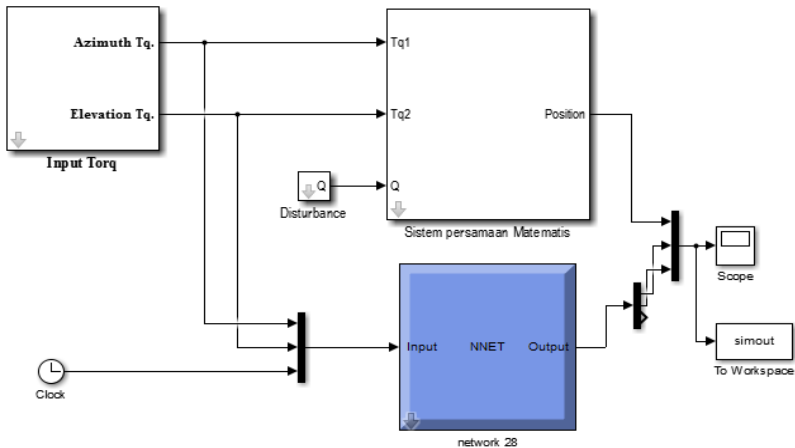
Baris pertama dan kedua pada tabel 4.5. adalah data output dari model penurunan persamaan matematis ( $f_i$ ). Baris ketiga dan keempat adalah data output dari model jaringan syaraf tiruan ( $y_i$ ). Baris kelima dan keenam adalah selisih dari data output ( $f_i - y_i$ ). Baris ke tujuh dan delapan adalah nilai absolut dari selisih data output ( $|f_i - y_i|$ ). Kolom yang paling kanan dari baris ke tujuh dan delapan adalah penjumlahan dari nilai absolut selisih. Dan pada baris ke 9 kolom paling kanan adalah jumlah total absolut error. Jumlah total dari nilai absolut tersebut akan dibagi dengan total jumlah data ( $n$ ) yaitu sejumlah 4562 data. Pada baris terakhir adalah nilai *Mean Absolute Error* dengan nilai 0,018 atau jika dijadikan persentase adalah 1,8%.



Hasil perhitungan *Mean Absolute Error* menunjukkan angka 1,8 %. Hal ini berarti bahwa model jaringan syaraf tiruan dapat mendekati model penurunan persamaan matematis. Error yang bernilai 1,8% ini juga sudah memenuhi target dari penelitian (2%) sehingga tidak perlu perhitungan ulang dalam mencari struktur jaringan syaraf tiruan yang lebih tepat.

## 5.2. Pengujian Respon Model

Model persamaan matematis dan model jaringan syaraf tiruan seharusnya mempunyai respon yang tidak terlalu berbeda jika diberikan input yang sama. Untuk membandingkan respon kedua model, *network28* jaringan syaraf tiruan diubah menjadi blok simulink terlebih dahulu. Jika kedua buah model telah menjadi blok simulink, maka akan dengan mudah membandingkan kedua respon model dalam bentuk grafik. Respon dari kedua model dibandingkan dengan diagram simulink pada gambar 4.5.



Gambar 4.5. Diagram Simulink Perbandingan Respon Model

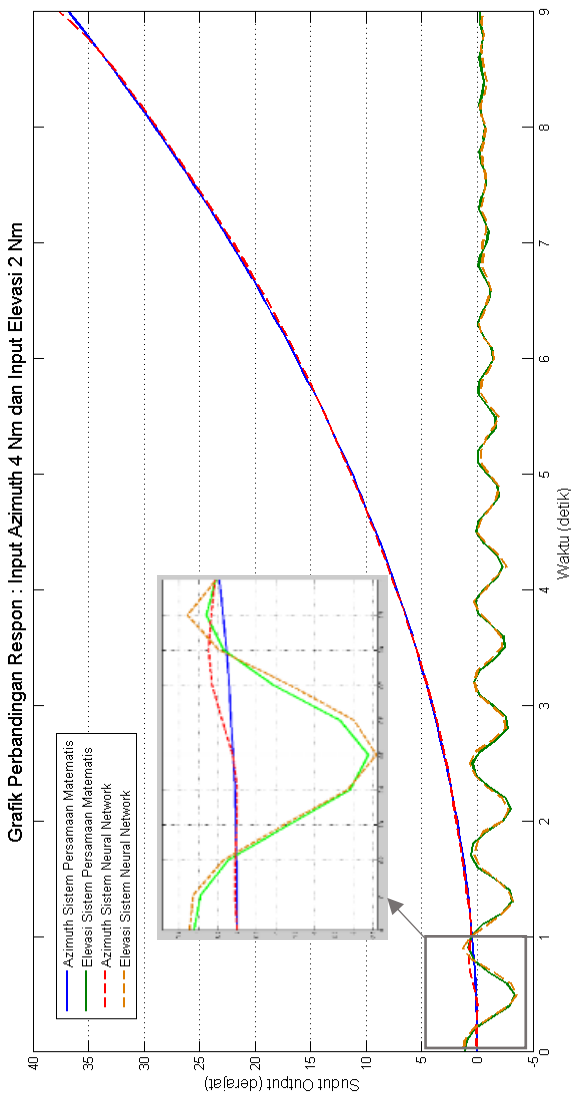
Pada gambar 4.5., input dimasukkan pada blok “input torq” berupa torsi azimuth dan torsi elevasi yang bernilai tetap/konstan.

Kemudian input akan masuk ke dalam blok kedua model. Blok yang berwarna putih “Sistem persamaan Matematis” adalah model yang diturunkan dari persamaan matematis, dan yang berwarna biru “network28” adalah model dari jaringan syaraf tiruan. Hasil dari kedua blok akan keluar menuju “scope” untuk dilihat grafiknya dan juga akan masuk ke simout sebagai data pada Matlab.

Pada percobaan pertama, model persamaan matematis dan model jaringan syaraf tiruan dibandingkan menggunakan input torsi azimuth lebih besar dari input torsi elevasi. Input yang dipilih pada pengujian ini adalah input torsi azimuth 4 Nm dan torsi elevasi 2 Nm. Hasil dari perbandingan respon grafik dapat dilihat pada gambar 4.6.

Gambar 4.6. menunjukkan bahwa sudut keluaran azimuth baik respon dari model penurunan persamaan matematis maupun respon dari model jaringan syaraf tiruan naik terus. Hal ini menunjukkan bahwa pada sumbu azimuth sistem akan bertambah sudutnya secara terus menerus jika dikenai torsi secara konstan. Selain itu, respon dari kedua sistem yang terus beriringan menunjukkan bahwa sistem jaringan syaraf tiruan dan sistem penurunan persamaan matematis memiliki error yang kecil.

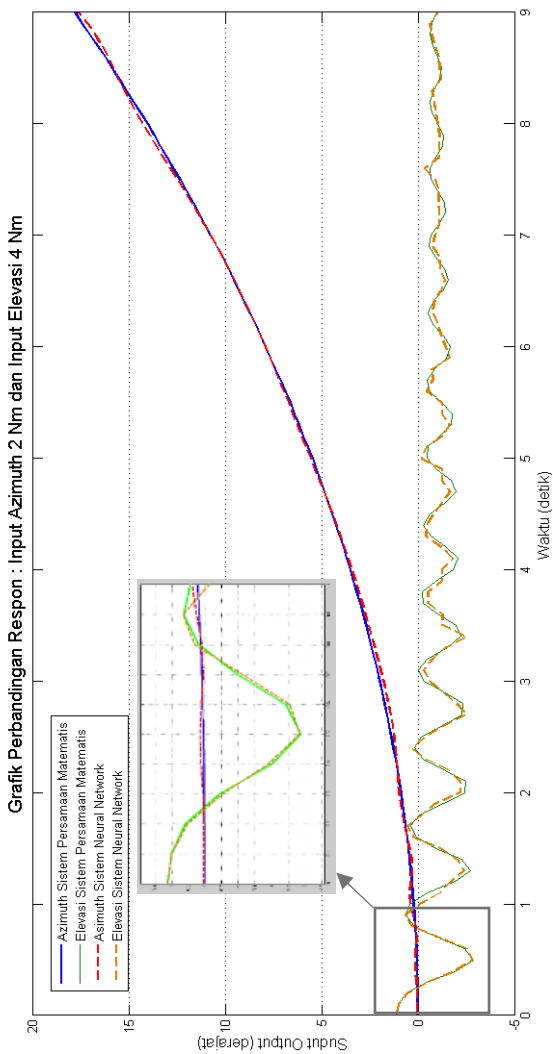
Gambar 4.6. menunjukkan pula bahwa respon sudut elevasi dari kedua model membentuk grafik yang berosilasi dengan amplitudo semakin lama semakin mengecil. Respon yang berosilasi ini disebabkan karena torsi input yang bernilai positif akan melawan torsi dari pengaruh berat *gun* yang bernilai negatif. Saat awal dimulainya input, resultan dari torsi ini bernilai negatif sehingga sudut akan turun. Ketika pada sudut tertentu, resultan dari torsi ini bernilai positif sehingga sudut akan naik kembali sampai resultan dari torsi bernilai negatif kembali. Respon dari kedua sistem juga terus beriringan yang menunjukkan error dari kedua sistem yang kecil.



Gambar 4.6. Grafik Respon torsi azimuth 4 Nm dan torsi Elevasi 2 Nm.

Pada pengujian kedua, model persamaan matematis dan model jaringan syaraf tiruan dibandingkan menggunakan input torsi azimuth lebih kecil dari input torsi elevasi. Input yang dipilih pada pengujian ini adalah torsi azimuth 2 Nm dan torsi elevasi 4 Nm. Hasil dari perbandingan respon grafik dapat dilihat pada gambar 4.7.

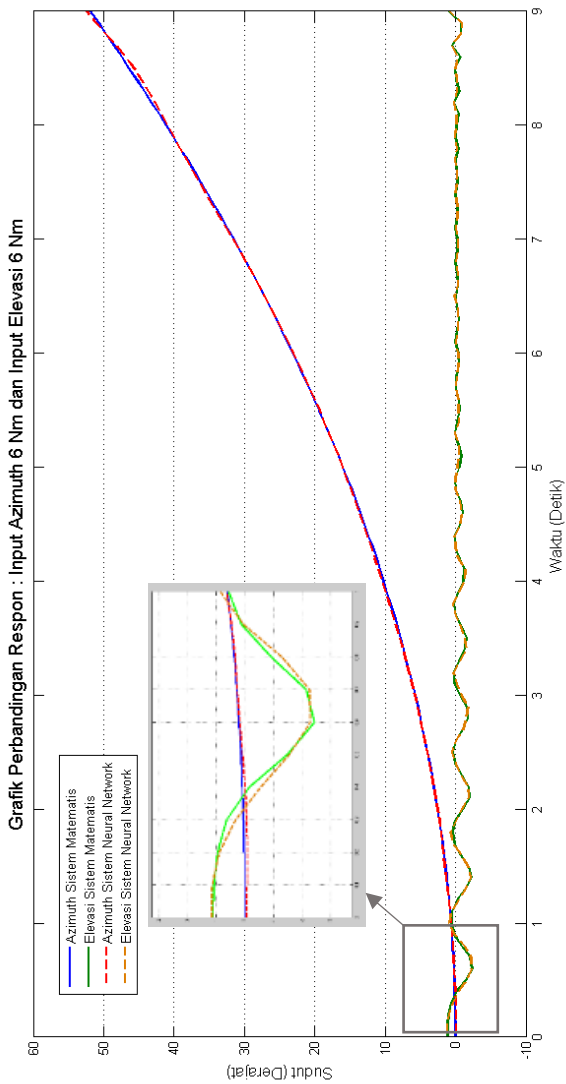
Gambar 4.7. menunjukkan bahwa keluaran sudut azimuth dari kedua model naik terus. Kenaikan dari sudut azimuth lebih lambat dari grafik yang sebelumnya dikarenakan torsi masukan yang lebih kecil. Respon dari kedua model yang saling beriringan menunjukkan bahwa kedua model memiliki error yang kecil. Respon sudut elevasi masih sama dengan grafik sebelumnya yaitu mengalami osilasi. Osilasi yang terjadi pada grafik ini memiliki amplitudo yang lebih kecil dari grafik sebelumnya karena torsi yang diberikan lebih besar. Respon dari kedua model yang saling beriringan menunjukkan bahwa kedua model memiliki error yang kecil.



Gambar 4.7. Grafik Respon torsi azimuth 2 Nm dan torsi elevasi 4 Nm.

Pada pengujian ketiga, model persamaan matematis dan model jaringan syaraf tiruan dibandingkan menggunakan input torsi azimuth sama dengan input torsi elevasi. Input yang dipilih pada pengujian ini adalah torsi azimuth 6 Nm dan torsi elevasi 6 Nm. Hasil dari perbandingan respon grafik dapat dilihat pada gambar 4.8.

Gambar 4.8. menunjukkan bahwa output sudut azimuth dari kedua model naik terus. Kenaikan dari sudut azimuth lebih cepat dari grafik yang sebelumnya dikarenakan torsi masukan yang lebih besar. Respon dari kedua model yang saling beriringan menunjukkan bahwa kedua model memiliki error yang kecil. Pada respon sudut elevasi masih sama dengan grafik sebelumnya yaitu mengalami osilasi. Osilasi yang terjadi pada grafik ini memiliki amplitudo yang lebih kecil dari grafik sebelumnya karena torsi yang diberikan lebih besar. Amplitudo yang lebih kecil ini disebabkan karena titik balik ke atas dicapai ketika sudut lebih kecil. Respon dari kedua model yang saling beriringan menunjukkan bahwa kedua model memiliki error yang kecil.



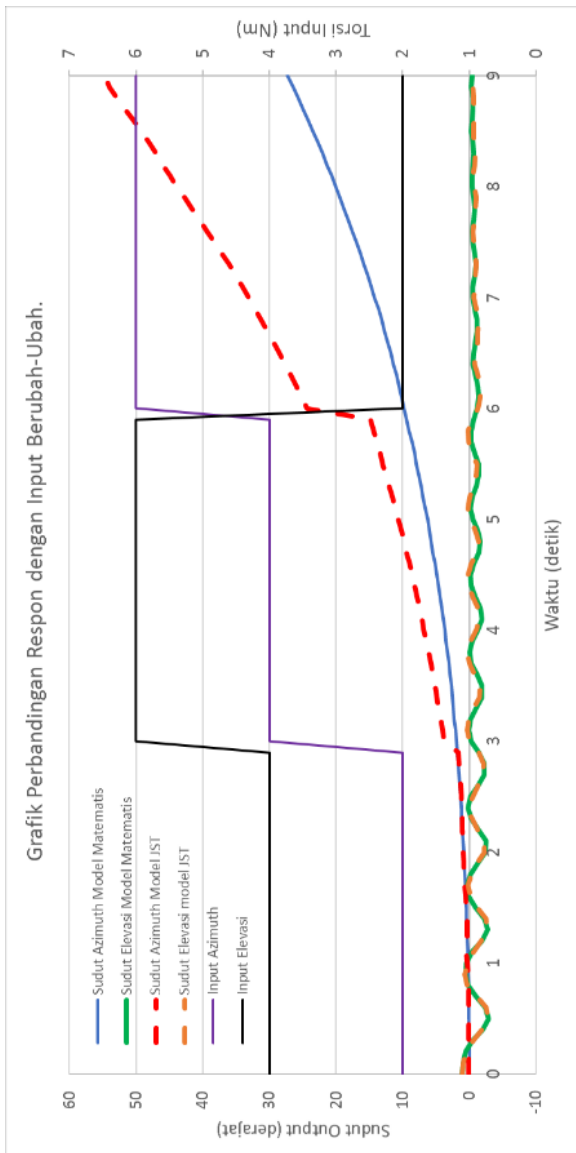
Gambar 4.8. Grafik Respon torsi azimuth 6 Nm dan torsi elevasi 6 Nm.

Pada pengujian selanjutnya, model persamaan matematis dan model jaringan syaraf tiruan dibandingkan menggunakan input torsi azimuth dan input torsi elevasi yang berubah 2 kali. Input yang dipilih pada pengujian ini adalah torsi azimuth 2 Nm dan torsi elevasi 4 Nm untuk detik 0 sampai detik ke-3, torsi azimuth 4 Nm dan torsi elevasi 6 Nm untuk detik ke-3 sampai detik ke-6 serta torsi azimuth 6 Nm dan torsi elevasi 2 Nm untuk detik ke-6 sampai detik ke-9. Hasil dari perbandingan respon grafik dapat dilihat pada gambar 4.9.

Gambar 4.9. menunjukkan bahwa respon kedua model dapat tetap berjalan secara beriringan sampai pada perubahan input. Ketika input berubah, azimuth pada model Jaringan Syaraf Tiruan naik melebihi dari model persamaan matematis. Ketika input berubah kembali pada detik ke-6, input respon dari model jaringan syaraf tiruan semakin naik menjauhi respon dari model persamaan matematis. Pada sumbu elevasi hal yang terjadi sebenarnya juga sama akan tetapi perbedaannya tidak terlalu kelihatan karena hanya di sekitar angka nol. Jika dilihat dari data, saat perubahan input detik ke-3 respon model jaringan syaraf tiruan mempunyai nilai yang sedikit lebih tinggi. Saat perubahan input detik ke-6 respon model jaringan syaraf tiruan mempunyai nilai yang sedikit lebih rendah dari respon model jaringan syaraf tiruan.

Ketidak sesuaian antara respon model jaringan syaraf tiruan dan respon model persamaan matematis saat input berubah ini disebabkan oleh model jaringan syaraf tiruan yang tidak bisa memperhitungkan output sebelumnya. Model jaringan syaraf tiruan langsung memasukkan input berupa torsi azimuth, torsi elevasi dan waktu dan memprosesnya. Sedangkan model persamaan matematis melakukan perhitungan dengan melakukan *feedback* berupa output yang baru keluar.





Gambar 4.9. Grafik perbandingan dengan input berubah-ubah.

*(Halaman ini sengaja dikosongkan)*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1. Kesimpulan**

Dari penelitian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut :

1. Sistem *turret gun* dapat dimodelkan dengan metode Jaringan Syaraf Tiruan yang terdiri dari 5 *layer* yang terdiri dari 4 *Hidden Layer* dan 1 *Output Layer*. *Neuron* pada masing-masing *hidden layer* adalah 20 buah dan *neuron* pada *output layer* adalah 3 buah. Struktur Jaringan Syaraf Tiruan tersebut dilengkapi dengan *weight* dan *bias* seperti pada lampiran 1.
2. Model jaringan syaraf tiruan dari sistem *turret gun* mempunyai performa MSE 0,000648, dan dibandingkan dengan model yang diturunkan dari persamaan matematis, model jaringan syaraf tiruan ini mempunyai error 1,8%.
3. Respon model jaringan syaraf tiruan mempunyai tendensi yang sama dengan respon model yang diturunkan dari persamaan matematis saat diberikan input yang konstan. Tetapi saat diberikan input yang berubah-ubah, respon model jaringan syaraf tiruan tersebut tidak bisa mengikuti *trendline* dari respon model persamaan matematis, karena model jaringan syaraf tiruan tidak bisa memperhitungkan nilai output sebelumnya.

#### **6.2. Saran**

Saran untuk penelitian selanjutnya yaitu :

1. Pengambilan data training dilakukan dengan jangkauan yang lebih luas dan pencacahan yang lebih kecil agar jaringan syaraf tiruan yang dibangun bisa lebih akurat.
2. Pengembangan model jaringan syaraf tiruan lebih lanjut agar bisa memperhitungkan nilai output sebelumnya.

*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1]. Mushonnifah, S. Subchan. Nurhadi, H. **Numerical Simulation of Acceleration and Force Control in Turret-Gun System With Disturbance**. IEEE Xplore Digital Library 1,1:13-18.
- [2]. Ismeal, G.M. Kyslan, K. Fedak, V. **DC Motor Identification Based on Recurrent Neural Networks**. Košice: Technical University of Košice. 2012.
- [3]. Wardah, C. L. **Kalibrasi Stereoo Kamera dan Penentuan Koordinat 3 Dimensi untuk Target Tunggal pada Sistem Pelontar Peluru Autotracking**. Surabaya: Institut Teknologi Sepuluh Nopember. 2015.
- [4]. Tim Dosen Simulasi dan Pemodelan. **Simulasi dan Pemodelan**. Jakarta:Universitas Gunadarma, 2014.
- [5]. Soemarno. **Prinsip-Prinsip Pemodelan Sistem**. Malang:Universitas Brawijaya, 2012.
- [6]. Kriesel, D. **A Brief Introduction to Neural Networks**. Bonn : dkriesel.com,2005.
- [7]. Siang, J.J. **Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab**. Yogyakarta: Andi, 2005.
- [8]. Martin T. Hagan, Howard B. Demuth, Orlando De Jesus. **An Introduction to Use of Neural Networks in Control Systems**. Oklahoma : Oklahoma State University, 2009.
- [9]. Cydebot, 2016. **Mean Absolute Error**, <URL:[http://en.wikipedia.org/wiki/mean\\_absolute\\_error](http://en.wikipedia.org/wiki/mean_absolute_error)>

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN

Tabel 1. *Weight* dari *input layer* ke *layer 1*

		Input		
		neuron 1	neuron 2	neuron 3
Layer 1	neuron 1	0,27949	-2,6277	-3,6192
	neuron 2	0,76272	1,4155	2,765
	neuron 3	0,16914	-1,8386	-2,9442
	neuron 4	1,1478	1,4958	-0,78642
	neuron 5	0,28925	3,3224	0,11429
	neuron 6	0,53673	-2,8196	-1,2666
	neuron 7	0,49981	2,561	1,3848
	neuron 8	0,31588	1,1732	3,3232
	neuron 9	-0,74037	1,2634	1,5604
	neuron 10	0,096498	-2,0581	1,2816
	neuron 11	0,19703	-2,1778	-1,7902
	neuron 12	-0,10475	2,5033	2,682
	neuron 13	0,002678	-1,2587	3,4178
	neuron 14	-0,87099	-0,76119	-1,1753
	neuron 15	0,6254	-0,11016	-1,0524
	neuron 16	-0,26591	3,1766	1,0794
	neuron 17	0,64635	-1,4802	-0,8968
	neuron 18	-0,00645	0,30636	3,5206
	neuron 19	-0,03298	2,7743	4,6221
	neuron 20	-1,4428	-0,76233	-1,9905

Tabel 2. *Weight* dari *layer 1* ke *layer 2*

		Layer 1									
		neuron 1	neuron 2	neuron 3	neuron 4	neuron 5	neuron 6	neuron 7	neuron 8	neuron 9	neuron 10
Layer 2	neuron 1	-0,34693	0,11248	-0,07652	-0,14253	-0,15444	0,072413	-0,72878	0,61629	0,62653	0,11023
	neuron 2	0,001619	0,96411	-0,72707	0,020032	1,0446	0,21422	-0,35823	1,1283	-0,221368	1,8576
	neuron 3	0,14201	-0,51263	0,18921	0,15904	0,72573	0,29851	0,3056	0,35343	-0,22523	-0,37048
	neuron 4	0,033396	-0,36794	-0,10301	-0,0569	-0,4238	-0,00727	0,29205	-0,26382	0,6012	0,26145
	neuron 5	0,37198	-0,47273	0,74905	0,008653	-0,74313	-0,2004	0,0633379	-0,08831	0,014585	-0,61776
	neuron 6	-0,15407	-0,28009	0,26373	0,095481	-0,00205	0,12787	0,36308	0,41982	-0,2749	-0,37409
	neuron 7	-0,08218	0,36251	-0,2644	-0,0177	-0,02081	-0,88755	-0,33022	-0,98405	0,098664	-0,0822
	neuron 8	0,026309	0,76938	0,34335	0,12213	0,96365	-0,38191	1,0464	0,75713	-0,16906	0,16847
	neuron 9	-0,00589	-0,0708	0,55807	-0,05726	-0,27415	-0,07405	0,24001	0,22978	0,099453	0,19481
	neuron 10	0,032331	0,33031	0,29679	0,031193	-0,1381	0,52777	0,61725	0,93899	0,43777	-0,51857
	neuron 11	0,48421	0,56356	0,26742	0,17959	-0,84758	0,62794	-0,03678	1,0466	0,055464	-0,34502
	neuron 12	-0,24376	-0,49667	0,70458	-0,11188	-0,88287	0,28855	-0,14321	0,54613	0,32765	0,46144
	neuron 13	-0,13625	-0,10653	-0,29867	-0,10349	0,043129	0,25282	-0,08596	0,86654	0,34038	0,35328
	neuron 14	0,12925	0,67513	0,44319	-0,06042	-0,19533	0,83958	0,31088	0,56953	0,52822	0,88199
	neuron 15	-0,55977	-0,82302	-0,91453	-0,06118	0,73748	-0,11267	-0,64682	0,049661	0,32504	0,60183
	neuron 16	0,67998	0,085	-0,28102	-0,08045	-0,4454	-0,75918	0,20461	0,76825	0,2464	0,24234
	neuron 17	0,13753	-0,72302	0,16794	-0,05769	-0,99965	0,26518	0,96008	0,52111	0,087532	-1,2809
	neuron 18	0,14086	-1,3438	-0,65181	-0,73208	0,52714	-0,75303	-0,41694	-0,82287	0,33595	-0,63403
	neuron 19	-0,61296	0,34503	-0,61925	-0,00469	1,4157	-0,71769	0,22469	0,24254	0,079412	1,4995
	neuron 20	0,3065	-0,64608	-0,64897	-0,00746	0,13877	-0,82796	-0,3112	-0,81534	-0,21541	-0,13284



Tabel 2. *Weight* dari *layer 1* ke *layer 2* (lanjutan)

		layer 1																			
		neuron 11	neuron 12	neuron 13	neuron 14	neuron 15	neuron 16	neuron 17	neuron 18	neuron 19	neuron 20										
Layer 2	neuron 1	-0,94268	-0,10713	0,38751	-0,2122	-0,19743	-0,22357	0,72336	-0,45738	0,072491	-0,01696										
	neuron 2	0,21956	0,94766	-0,45452	0,30218	-0,1375	-0,60476	-1,233	0,35444	-0,00114	-0,43216										
	neuron 3	-0,02514	-0,18561	-0,41556	0,32378	-0,2132	-0,81085	0,50216	-0,02289	-0,69992	-0,01134										
	neuron 4	0,14922	-0,50223	0,312	-0,01479	0,057043	0,17692	-1,0953	-0,05224	-0,49246	-0,09479										
	neuron 5	0,67623	-0,4095	0,28598	0,026213	-0,10197	-0,1751	0,30392	-0,69411	0,30454	0,12272										
	neuron 6	-0,58541	0,53113	0,44929	0,04832	0,51725	-0,02743	-0,5922	0,33101	0,58405	-0,05224										
	neuron 7	-0,38246	-0,10471	-0,50047	-0,23682	-0,36762	-0,17732	-0,55353	-0,16602	0,5056	-0,09917										
	neuron 8	0,12744	1,6637	-1,0782	-0,18486	0,79022	2,0075	0,39553	0,79704	-0,31313	0,033141										
	neuron 9	-0,79595	-0,54177	-0,20166	-0,01142	-0,91582	-0,25479	0,17693	1,4837	-0,01974	-0,04501										
	neuron 10	0,42247	0,8743	-1,8809	0,20087	0,31557	-0,22248	-1,2149	-0,04948	1,0291	-0,00068										
	neuron 11	-0,02603	0,51423	0,34102	0,12286	-0,05302	-0,66454	0,59752	0,44805	-0,91606	0,040613										
	neuron 12	-0,89802	0,44674	-0,93305	0,020218	-0,68848	0,16111	-0,29212	-0,34504	-0,37324	0,043116										
	neuron 13	0,43325	-1,0217	-0,23093	-0,0698	0,13778	0,85286	-0,57855	1,9667	-0,2404	-0,03801										
	neuron 14	0,15801	-0,01056	1,2626	-0,36703	0,44724	-1,2951	-0,05771	-0,74668	0,15153	0,047585										
	neuron 15	0,22676	-0,79604	1,1061	-0,22322	-0,0138	0,99791	-0,4131	0,52264	-0,25853	-0,26778										
	neuron 16	-0,46641	0,27878	-0,30029	-0,17938	0,51845	-0,14933	-0,18542	-0,58787	-0,72205	-0,00018										
	neuron 17	0,31008	-0,448	-0,58277	-1,1446	0,43229	-0,23114	0,21674	0,67074	-0,37889	-0,38888										
	neuron 18	0,34548	-0,02801	0,26092	0,23472	-0,05964	0,56186	-0,4122	0,42969	-0,72635	-0,24759										
	neuron 19	-1,1666	-0,62505	1,0375	0,086864	0,46136	1,5246	-1,0387	-1,3854	-0,83984	-0,01906										
	neuron 20	0,51546	-1,0313	0,446	0,39073	0,28578	0,61667	-0,29063	-0,51506	0,17319	0,4071										

Tabel 3. *Weight* dari layer 2 ke layer 3

		Layer 2											
		neuron 1	neuron 2	neuron 3	neuron 4	neuron 5	neuron 6	neuron 7	neuron 8	neuron 9	neuron 10		
Layer 3	neuron 1	0,15997	0,16506	-0,17874	1,0778	0,35174	1,6042	0,1681	-0,29433	1,2307	-1,0203		
	neuron 2	-0,00132	0,83756	0,33002	-0,4067	-0,85672	1,0209	0,83328	-0,53195	-0,01083	0,64096		
	neuron 3	-0,2119	0,34016	-0,05108	-0,22029	0,55582	0,62753	0,083985	-0,05224	0,13744	0,40469		
	neuron 4	0,062212	0,57752	0,21284	-0,04264	-0,93428	-0,40472	-0,74644	-0,28949	0,48075	-0,24385		
	neuron 5	-0,09032	-0,30543	-0,01485	-0,27745	-1,2096	0,22894	0,16477	-0,20956	-0,52255	0,77723		
	neuron 6	-0,27652	-0,81832	0,25724	-0,08883	-0,40504	0,76972	0,092423	-0,15686	0,45317	-0,71033		
	neuron 7	0,37927	-0,80298	-0,00156	-0,59964	-0,86788	-0,42112	-0,34031	0,084585	0,42997	0,029471		
	neuron 8	-0,18899	1,0161	-0,23292	-0,22098	0,94851	-0,83709	-0,00639	0,32733	-0,28579	0,1929		
	neuron 9	0,10148	-1,4855	0,46533	-0,21525	-1,0998	-0,00083	-0,42837	-0,11333	-1,2297	0,75945		
	neuron 10	0,11577	1,2617	0,76956	0,91569	-0,05244	0,51929	0,93289	0,97458	0,61109	0,12751		
	neuron 11	-0,70886	-0,1221	-0,64202	1,3005	1,2082	-2,4947	-0,75567	-0,44784	-0,31663	-0,42366		
	neuron 12	-0,01952	-1,1918	-0,83999	0,92428	0,82809	-0,51607	0,44185	-0,37714	-1,3865	-0,42855		
	neuron 13	-0,09732	0,60588	-0,2233	-0,10917	1,9108	0,20787	-0,58483	0,2115	0,44876	-0,1528		
	neuron 14	0,03162	-0,40548	0,13522	0,64235	-0,34281	-0,08517	-0,1278	0,032587	-1,5611	0,27787		
	neuron 15	0,48795	0,20091	0,38049	-0,41381	0,018061	-0,02302	-0,35842	0,11704	-0,36516	0,95543		
	neuron 16	0,17379	-0,63591	-0,40593	-0,53319	-0,12252	-0,38269	0,34053	0,28013	-0,81853	-0,27182		
	neuron 17	-0,15876	-0,31717	-0,25425	-0,55244	1,2794	0,41864	0,24076	0,06465	-0,40484	-1,5605		
	neuron 18	0,11012	-0,1458	-0,27536	-0,47894	-0,71166	0,64889	-0,10875	-0,15303	-0,07985	0,39391		
	neuron 19	0,40761	0,21376	-0,20987	-0,09567	-0,25789	0,042198	-0,98026	-0,21381	-0,29319	-1,001		
	neuron 20	-0,1361	0,484	-0,17463	-0,09807	0,28474	0,036618	0,7166	-0,19246	0,90163	0,28175		

Tabel 3. *Weight* dari *layer 2* ke *layer 3* (lanjutan)

		layer 2																		
		neuron 11	neuron 12	neuron 13	neuron 14	neuron 15	neuron 16	neuron 17	neuron 18	neuron 19	neuron 20									
Layer 3	neuron 1	-0,27179	-0,21785	0,62382	-0,2618	0,1052	-0,01987	-0,47754	0,41991	0,13185	0,070614									
	neuron 2	0,66108	0,13686	0,33762	-0,08084	-0,22784	0,21672	-0,08257	-0,04885	-0,24879	-0,53702									
	neuron 3	0,43077	-0,85271	-0,23252	0,043339	0,19432	-0,94858	-0,61021	0,090539	0,2731	0,36209									
	neuron 4	0,65611	-0,76246	-0,62988	0,38282	-0,857	0,32203	-0,71669	-1,3925	0,17624	-0,12218									
	neuron 5	0,45979	-0,11621	1,1605	0,095112	-0,11651	0,23242	0,041836	0,93302	0,57387	-0,89808									
	neuron 6	0,13213	-0,12086	0,70845	-0,3326	0,63062	0,34027	-1,396	-0,62255	-0,74778	-0,67713									
	neuron 7	0,48569	0,18916	0,01216	-0,47039	-0,27583	0,68277	0,59536	0,87759	-0,5161	0,12386									
	neuron 8	0,91841	-0,19238	0,63524	0,45015	-0,51527	0,89086	0,59186	-0,05797	-0,84494	0,32289									
	neuron 9	-0,18173	0,13204	-0,13496	-0,14273	-0,40557	0,53797	0,17889	0,094596	-0,33789	-0,10107									
	neuron 10	0,098883	0,17804	0,29271	0,27359	-0,37822	0,20624	-0,24551	0,031631	1,2382	0,7079									
	neuron 11	0,52193	-0,81868	0,69231	-0,23388	0,80719	0,3921	-0,4204	-0,42623	0,58281	-0,38824									
	neuron 12	0,40401	-0,45551	-0,10787	-0,14547	0,53668	-0,22702	0,8213	0,4865	-1,5901	-0,64413									
	neuron 13	0,26586	-0,08762	-0,59234	-0,15066	0,033392	0,30059	0,15216	-0,4909	0,15303	-0,7874									
	neuron 14	0,20337	0,051799	0,92641	-0,18444	-0,32231	0,64029	-0,17378	-0,66843	-0,11529	0,33552									
	neuron 15	0,33945	0,062266	-0,19783	-0,34196	-0,26342	0,69142	0,41895	0,11317	-1,2031	-1,5146									
	neuron 16	0,10132	-0,77438	-0,59752	0,65691	-0,62473	0,45924	0,25774	0,46571	-0,53989	-0,544									
	neuron 17	-0,1742	-0,23355	-0,66131	0,42569	-0,05962	-0,16531	0,24435	-0,25304	-0,47647	-0,04278									
	neuron 18	-0,02432	-0,02041	-0,23781	-0,28416	-0,48917	0,87024	0,49686	0,33192	-0,02915	0,21701									
	neuron 19	0,42639	0,022063	-0,09067	0,60248	0,63643	-0,52729	0,28365	0,083996	-0,09436	0,20069									
	neuron 20	-1,2759	0,037213	-0,47621	-0,59193	0,30319	-0,197	-0,09372	-0,32179	1,5474	1,6251									

Tabel 4. *Weight* dari *layer 3* ke *layer 4*

		Layer 3									
		neuron 1	neuron 2	neuron 3	neuron 4	neuron 5	neuron 6	neuron 7	neuron 8	neuron 9	neuron 10
Layer 4	neuron 1	0,63373	-0,21024	-0,37408	0,023587	-0,35579	0,66377	-0,15768	-0,59931	-0,50826	-0,63118
	neuron 2	-0,15296	0,31091	-0,0269	1,2975	0,80807	-0,28639	0,43026	-0,01856	-0,10706	-0,10604
	neuron 3	0,20003	-0,33938	-0,83376	-0,84819	0,6094	0,48112	0,54896	0,074073	-0,45288	-0,25389
	neuron 4	-0,22407	1,0404	0,62965	0,49618	0,085894	0,12767	-0,34339	-0,88467	0,079483	0,26622
	neuron 5	-1,6477	-0,92255	-0,18849	-1,2441	0,66745	-1,0212	1,242	-0,09404	0,027987	-1,657
	neuron 6	0,4322	-0,28697	-0,08415	0,56125	0,44787	0,1112	0,088404	-0,48659	0,23674	0,075217
	neuron 7	-0,74816	0,51307	-0,06714	0,50358	0,40328	-0,57688	0,30886	-0,30492	0,32947	0,11107
	neuron 8	-1,044	-0,31385	-0,48893	0,01812	0,32922	-0,71862	-0,25892	-0,76494	0,34982	0,5271
	neuron 9	-0,45833	0,033374	-0,17359	0,26746	0,39424	-0,40021	0,51207	-1,6224	-0,27873	-0,50826
	neuron 10	0,14776	0,63553	0,25788	0,42527	0,353	-1,3879	-0,13895	0,24964	-0,63612	-1,7041
	neuron 11	0,073009	0,38101	-0,38003	0,32288	0,2365	0,16783	0,54691	-0,24083	0,12141	-0,02028
	neuron 12	0,47111	0,19203	0,744	0,087009	1,1165	0,53085	-0,60981	1,0286	0,60468	0,38425
	neuron 13	0,025706	-0,19551	0,091124	-0,32695	0,02354	-0,08176	-0,51557	0,14877	-0,17427	0,21606
	neuron 14	-0,12675	0,507	0,90152	0,096688	0,4592	-0,42155	-0,52646	-0,03666	0,095958	-0,32843
	neuron 15	0,344	0,65819	0,67227	0,79276	0,2506	0,001435	0,10326	-0,25354	0,056864	0,039076
	neuron 16	1,3131	0,68045	-0,24431	-0,32002	0,52389	-1,3428	0,21392	0,26811	0,57873	-0,17893
	neuron 17	-0,03498	0,24263	0,25667	-0,49961	-0,34732	-0,05837	0,090586	0,15393	-0,03979	0,057702
	neuron 18	-1,4593	-0,62274	0,057595	0,083032	-0,10961	-0,2063	0,73246	-0,4289	2,119	0,26402
	neuron 19	-0,05418	-0,3819	-0,39823	0,71564	-0,12011	0,12958	-0,2165	0,31262	0,47173	0,45584
	neuron 20	-0,65276	-0,93622	-0,60965	-0,09532	-0,17338	-0,75806	0,3349	0,084174	-0,04384	0,20175

Tabel 4. *Weight* dari *layer 3* ke *layer 4* (lanjutan)

		layer 2																		
		neuron 11	neuron 12	neuron 13	neuron 14	neuron 15	neuron 16	neuron 17	neuron 18	neuron 15	neuron 18	neuron 15	neuron 20							
layer 4	neuron 1	-0,19478	-0,29015	-0,47588	0,046028	-0,11056	-0,47396	0,57842	-0,52094	-0,33929	0,9481									
	neuron 2	-0,21251	-0,14947	0,28786	-0,28977	-0,00565	-0,37945	-0,07018	0,34923	0,26338	-0,1362									
	neuron 3	0,17299	-0,12097	0,29519	0,044601	-0,1586	0,50099	-0,17724	0,56933	0,51931	-0,05187									
	neuron 4	0,18591	0,4394	-1,2245	0,29622	-0,42477	-0,50937	0,5928	-0,18359	-1,2568	0,60081									
	neuron 5	-0,14411	0,27279	0,4551	0,67593	1,4451	1,0694	-1,0367	2,0661	1,9794	-1,4626									
	neuron 6	0,12291	-0,17898	0,1724	-0,68938	-0,38878	-0,31176	0,51472	0,25382	-0,60286	0,27435									
	neuron 7	-0,29016	-0,6856	-0,73189	0,13364	0,48081	-0,00917	0,48597	1,0588	0,47361	0,59542									
	neuron 8	-0,19293	-1,0029	-0,91384	0,21911	0,82703	0,19123	0,4817	0,06574	0,45074	0,58687									
	neuron 9	-0,47318	1,031	-0,11471	0,29242	0,53752	0,63768	-1,08	0,70468	-0,0984	0,46439									
	neuron 10	1,5475	-0,49305	-0,90018	0,47988	2,3226	1,0009	-1,2575	-0,35984	-0,58267	1,7863									
	neuron 11	-0,52334	-0,02953	-0,10704	0,25621	-0,32295	0,30874	-0,14764	0,306	-0,03401	0,40471									
	neuron 12	-0,78909	-0,27633	-0,03117	0,63584	-0,77443	0,095156	0,83206	0,079652	-0,8422	0,80469									
	neuron 13	0,3373	-0,0074	0,008557	-1,161	0,33269	-0,22963	-0,0842	-0,15155	-0,08578	-0,22001									
	neuron 14	0,37195	0,14061	-0,09129	0,28609	-0,31524	-0,00231	0,48274	0,33222	-0,48263	0,81306									
	neuron 15	-0,48803	-0,25639	-0,11925	-0,79432	-0,37543	-0,40731	0,38055	0,14254	-0,20527	0,20466									
	neuron 16	1,186	-0,91891	-0,07119	-0,0076	-0,52621	-0,18549	0,95977	-1,1581	0,057029	0,17674									
	neuron 17	0,45857	-0,13181	0,075126	-0,23859	0,17947	-0,37282	0,85714	-0,12239	-0,16954	-0,40364									
	neuron 18	-0,1182	-0,36218	0,02493	-0,57688	0,090659	1,1253	0,34638	-0,07077	0,25326	-1,0449									
	neuron 19	0,182	-0,05828	0,64061	0,34187	0,15965	-0,64957	0,026071	-0,49679	-0,11715	0,12761									
	neuron 20	0,53629	0,44442	-0,07021	0,29578	0,41306	0,29269	-0,72779	0,85502	0,44967	-0,78135									

Tabel 5. *Weight* dari *layer* 4 ke *layer* 5

		Layer 4									
		neuron 1	neuron 2	neuron 3	neuron 4	neuron 5	neuron 6	neuron 7	neuron 8	neuron 9	neuron 10
layer 5	neuron 1	3,6077	4,3569	3,7191	-5,1499	0,79235	2,5471	3,0985	-2,9299	3,2986	0,10647
	neuron 2	-0,96468	-0,51505	1,111	0,27442	-0,66801	1,0129	1,6369	2,5408	0,27586	4,6524
	neuron 3	0,054016	0,36381	0,47521	-0,27511	0,024109	0,082971	-0,01529	0,071458	0,030599	-0,00069

Tabel 5. *Weight* dari *layer* 4 ke *layer* 5 (lanjutan)

		layer 4									
		neuron 11	neuron 12	neuron 13	neuron 14	neuron 15	neuron 16	neuron 17	neuron 18	neuron 19	neuron 20
layer 5	neuron 1	2,0123	1,8872	-0,02466	3,6622	-3,102	-0,03737	3,1027	4,3231	3,8408	3,4357
	neuron 2	-1,9059	1,6106	0,15627	0,13356	1,2021	-1,5221	-0,80269	0,19111	0,15781	0,74289
	neuron 3	0,51089	-0,81787	3,1213	-0,65344	0,92886	-0,1376	-1,5529	-0,16729	-0,10105	-0,3377

Tabel 6. *Bias* untuk *layer* 1

Layer 1	neuron 1	-3,1052
	neuron 2	-3,3769
	neuron 3	-2,1958
	neuron 4	-0,87313
	neuron 5	-2,4757
	neuron 6	-0,70068
	neuron 7	-1,3602
	neuron 8	-1,4373
	neuron 9	-0,01415
	neuron 10	-0,02453
	neuron 11	-0,15318
	neuron 12	0,24707
	neuron 13	0,24286
	neuron 14	-0,14624
	neuron 15	1,1059
	neuron 16	-0,74702
	neuron 17	2,2405
	neuron 18	3,4679
	neuron 19	-2,8692
	neuron 20	-3,1962

Tabel 7. *Bias untuk layer 2*

Layer 2	neuron 1	1,8775
	neuron 2	-1,3434
	neuron 3	-1,5565
	neuron 4	-1,2121
	neuron 5	0,47278
	neuron 6	1,042
	neuron 7	0,072962
	neuron 8	0,13863
	neuron 9	0,31284
	neuron 10	-0,42849
	neuron 11	-0,04744
	neuron 12	0,56659
	neuron 13	-1,261
	neuron 14	0,15735
	neuron 15	-1,3958
	neuron 16	0,99473
	neuron 17	1,5663
	neuron 18	1,443
	neuron 19	-1,0283
	neuron 20	1,4871



Tabel 8. *Bias* untuk *Layer 3*

Layer 3	neuron 1	-2,1215
	neuron 2	-1,6376
	neuron 3	-1,5272
	neuron 4	-1,174
	neuron 5	1,5251
	neuron 6	-1,3473
	neuron 7	-0,14063
	neuron 8	0,41006
	neuron 9	0,099727
	neuron 10	-0,04625
	neuron 11	-0,44426
	neuron 12	-0,55784
	neuron 13	0,70218
	neuron 14	0,32642
	neuron 15	0,67292
	neuron 16	0,85989
	neuron 17	-0,66195
	neuron 18	1,5485
	neuron 19	1,6075
	neuron 20	-1,4747

Tabel 9. *Bias* untuk *Layer 4*

Layer 4	neuron 1	-1,7675
	neuron 2	-1,6788
	neuron 3	-0,81295
	neuron 4	0,51922
	neuron 5	2,6883
	neuron 6	-0,87114
	neuron 7	0,024379
	neuron 8	0,6082
	neuron 9	0,84199
	neuron 10	-0,22689
	neuron 11	0,16855
	neuron 12	-0,11497
	neuron 13	-0,51535
	neuron 14	-0,71004
	neuron 15	0,46115
	neuron 16	0,58808
	neuron 17	-1,0146
	neuron 18	-0,76106
	neuron 19	-1,5082
	neuron 20	-0,8334

Tabel 10. *Bias* untuk *Layer 5*

Output Layer	neuron 1	-2,7819
	neuron 2	0,26371
	neuron 3	-1,3878

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



Penulis bernama Agung Kartika Fibrianto dilahirkan di Kabupaten Magetan pada tanggal 11 Februari 1994. Penulis yang merupakan putra pertama dari pasangan Bapak Hariyanto dan Ibu Sukarti ini, memulai pendidikannya di TK Marsudisiwi Candirejo pada tahun 1999 sampai 2000. Dilanjutkan di SDN Candirejo 1 (2000-2006), SMP Negeri 1 Magetan (2006-2009), dan SMA Negeri 1 Magetan (2009-2012). Kemudian, penulis melanjutkan pendidikan ke jenjang perguruan tinggi di Institut Teknologi Sepuluh Nopember (ITS) Surabaya jurusan S1 Reguler Teknik Mesin.

Penampilan yang ramah dan santun membuat pria berkulit sawo matang ini mudah bergaul. Selama jenjang perkuliahan, penulis aktif dalam kegiatan organisasi khususnya kerohanian. Jabatan organisasi yang pernah diemban penulis adalah sebagai sekretaris umum LDJ Ash-Shaff Teknik Mesin ITS. Di jurusan Teknik Mesin, penulis mengambil bidang studi Manufaktur dan bergabung dalam Laboratorium Perancangan dan Pengembangan Produk. Penulis juga aktif dalam kegiatan laboratorium yaitu sebagai grader dan asisten praktikum Pengukuran Teknik. Email penulis yang bisa dihubungi adalah [agung.kartika1994@gmail.com](mailto:agung.kartika1994@gmail.com).

*(Halaman ini sengaja dikosongkan)*