TUGAS AKHIR – TM 141585

DESAIN SISTEM PENGENDALI *PROPORTIONAL – INTEGRAL – NEURAL NETWORK* PADA *PROTOTYPE TURRET GUN*

MAS'UD ASADULLAH
NRP 02111340000154

DOSEN PEMBIBING
ARIF WAHJUDI, S.T., M.T., Ph.D.

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – TM 141585

DESAIN SISTEM PENGENDALI *PROPORTIONAL-INTEGRAL- NEURAL NETWORK* PADA *PROTOTYPE TURRET GUN*

MAS'UD ASADULLAH
NRP. 02111340000154

DOSEN PEMBIMBING
ARIF WAHJUDI, S.T., M.T., Ph.D.

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – TM 141585

THE CONTROL SYSTEM DESIGN OF PROTOTYPE TURRENT GUN WITH PROPORTIONAL - INTEGRAL- NEURAL NETWORK

MAS'UD ASADULLAH
NRP. 02111340000154

ADVISOR
ARIF WAHJUDI, S.T., M.T., Ph.D.

MECHANICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute Of Technology
Surabaya 2018

DESAIN SISTEM PENGENDALI *PROPORTIONAL-INTEGRAL- NEURAL NETWORK* PADA *PROTOTYPE TURRET GUN*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Pada
Program studi S-1 Departemen Teknik Mesin
Fakultas Teknik Industri
Institut Teknologi Sepuluh Nopember

Oleh :

MAS'UD ASADULLAH
NRP. **02111340000154**

Disetujui oleh tim penguji tugas akhir :

1. Arif Wahjudi, S.T., M.Sc., Ph.D. ..... (Pembimbing)
   NIP. 195811061986011002
2. Dinny Harnany, S.T., M.Sc. ..... (Penguji I)
   NIP. 2100201405001
3. Ari Kurniawan Saputra, S.T., M.T. ..... (Penguji II)
   NIP. 198604012015041001

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS TEKNOLOGI INDUSTRI DEPARTEMEN TEKNIK MESIN

SURABAYA
JULI, 2018

[Halaman ini sengaja dikosongkan]

DESAIN SISTEM PENGENDALI *PROPORTIONAL-INTEGRAL- NEURAL NETWORK* PADA *PROTOTYPE TURRET GUN*

Nama : Mas'ud Asadullah
NRP : 02111340000154
Departemen : Teknik Mesin FTI-ITS
Dosen Pembimbing : Arif Wahjudi, S.T., M.T., Ph.D.

ABSTRAK

Dalam penggunaan senjata otomatis, objek yang terdeteksi menjadi sasaran tembak. Oleh karena itu kecepatan dan akurasi menjadi faktor utama untuk diperhatikan. Berat dari senjata, mekanisme senjata, dan sensor pendeteksi gerak menjadi hal utama dalam mempengaruhi kecepatan dan akurasi. Selain 3 hal di atas terdapat faktor lain yang bisa meningkatkan kecepatan dan akurasi yaitu mengatur sistem pengendali yang sesuai untuk senjata otomatis.

Pada tugas akhir ini, desain sistem pengendalian dirancang untuk pengendalian senjata otomatis terhadap sudut elevasi dan sudut rotasi. Sistem pengendalian yang digunakan adalah proportional-integral (PI). Sistem pengendalian ini membutuhkan nilai proportional gain (Kp) dan integral gain (Ki) untuk dijadikan sebagai nilai pengali error, nilai ini menjadi input sistem pengendali. Pada sistem pengendali PI ini hanya ditetapkan satu nilai Kp dan Ki untuk satu sistem. Karena itu metode neural network merupakan sistem yang dapat mengubah strukturnya untuk memacahkan masalah berdasarkan informasi yang ada. metode ini dapat membuat nilai Kp dan Ki bisa menyesuaikan terhadap input sistem yang masuk. Persamaan nilai Kp dan Ki tersebut diperoleh dari pelatihan neural nework.

Dari tugas akhir ini, hasil rancangan sistem pengendali PINN yang telah dibangun mampu mengendalikan prototype

turret gun. sudut rotasi didapatkan nilai %overshoot (%OS) tertinggi adalah 12,71% dan terendah adalah 0,00% Sedangkan untuk nilai rise time (RT) tertinggi adalah 0,45 detik dan terendah 0,14 detik. Untuk nilai steady-state error (SSE) tertinggi adalah 5,38 derajat dan terendah 0,87 derajat. Pada sudut elevasi didapatkan nilai %overshoot (%OS) tertinggi adalah 9,04% dan terendah adalah 0,00%. Sedangkan untuk nilai rise time (RT) tertinggi adalah 0,79 detik dan terendah 0,21 detik. Untuk nilai steady-state error (SSE) tertinggi adalah 4,52 derajat dan terendah 0,10 derajat.

Kata kunci : senjata otomatis; neural network; proportional-integral control; turret gun; sistem pengendali;

THE CONTROL SYSTEM DESIGN OF PROTOTYPE TURRENT GUN WITH PROPORTIONAL - INTEGRAL-NEURAL NETWORK

Name	**: Mas'ud Asadullah**
NRP	**: 02111340000154**
Department	**: Teknik Mesin FTI-ITS**
Advisor	**: Arif Wahjudi, S.T., M.T., Ph.D.**

ABSTRACT

Maximun function of automatic weapon, can be realize if can detected object became great subject firing. Therefore speed and accuracy are the main factors to be considered. The weight, mechanism, and motion detection sensors from weapon are the main things in affecting speed and accuracy. In addition that 3 things are factors to improve the speed and accuracy of setting up the appropriate control system for automatic weapons.

In this final project, control system design is for automatic weapon control design of elevation angle and rotation angle. The control system used is proportional-integral (PI). This control system requires the value of proportional gain (Kp) and integral gain (Ki) to serve as the error multiplier value. This value becomes the control system input. In this PI controller system only one value of Kp and Ki that determined for one system. Therefore the neural network method is a system that can change its structure to mate the problem based on existing information. This method can make the Kp and Ki values adjust the input of the incoming system. The equation of Kp and Ki values are obtained from neural nework training.

From this final task, the design of PINN control system that already been built is able to control the prototype turret gun. The highest presentase of Rotation angle obtained in overshoot (% OS) is 12.71% and the lowest is 0.00%. But for

the highest rise time (RT) value is 0.45 seconds and the lowest 0.14 seconds. The highest steady-state error (SSE) is 5.38 degrees and the lowest is 0.87 degrees. At the highest persentase % of elevation angle overshoot (% OS) is 9.04% and the lowest is 0.00%. But in other hand the highest rise time (RT) is 0.79 seconds and the lowest 0.21 seconds. The highest steady-state error (SSE) is 4.52 degrees and the lowest is 0.10 degrees.

Keywords: automatic weapons; neural network; proportional-integral control; turret gun; control system.

KATA PENGANTAR

Puji dan syukur marilah kita panjatkan kehadirat Tuhan Yang Maha Esa. Karena setiap hidayah dan karunia-Nya, penulis mampu menyelesaikan Tugas Akhir yang berjudul **Desain Sistem Pengendali *Proportional-Integral- Neural Network* Pada *Prototype Turret Gun*.**

Tidak lupa ucapan terima kasih penulis sampaikan kepada berbagai pihak yang telah membantu dalam penyelesaian Tugas Akhir ini. Adapun ucapan terima kasih ini, penulis sampaikan kepada:

1. **Allah SWT** yang telah memberi semua anugrah, berkah, dan hidayahnya kepada penulis.
2. **Kedua orang tua** penulis yang selalu mendoakan, mendidik, dan memberi semangat putra-putrinya untuk melakukan yang terbaik.
3. **Kakak** dan **Adik** penulis yang telah memberikan banyak nasehat dan dukungan terbaiknya kepada penulis.
4. **Bapak Arif Wahjudi, S.T., M.T., Ph.D.,** selaku dosen pembimbing Tugas Akhir yang selalu memberikan ilmu, kritik dan saran, serta bimbingannya selama proses penyelesaian Tugas Akhir ini.
5. **Ibu Dinny Harnany, S.T., M.Sc.**, dan **Bapak Ari Kurniawan Saputra, S.T, M.T.** selaku dosen penguji Tugas Akhir, yang telah menyempatkan berbagi ilmu, sehingga tugas akhir ini dapat terselesaikan dengan baik.
6. **Ibu Ika Dewi Wijayanti, S.T., M.Sc.,** dan **Bapak Achmad Syaifudin, S.T, M.Eng, Ph.D.**, selaku dosen wali yang telah membimbing penulis selama masa studi di Teknik Mesin ITS.
7. **Hafizh Nazhar Pahlevi** dan **Ikhsan Abi Nubli** selaku teman seperjuangan dalam pembuatan Tugas Akhir ini.
8. **Ahmad Syauqi**, **Muhammad Shofiyurrohman Basyir**, **Raudhatul Jannah**, **Farah Nadia**, dan **Laila Nazilah**

selaku sahabat yang selalu setia mendengar curahan hati penulis selama ini.

9. Segenap keluarga besar Laboratorium Perancangan dan Pengembangan Produk yang selalu memberikan semangat, motivasi, suka, dan duka kepada penulis.
10. Segenap teman-teman M56, HMM 14/15, ITS TV, BEM ITS 16/18, Kementerian Inkubator Kajian 16/18, 1000 Guru Surabaya, Alumni SDIT IQRO angkatan 10 serta teman-teman organisasi lainnya yang tidak bisa saya sebutkan satu persatu. Terima kasih telah memberikan pengalaman dan mendukung saya di setiap keputusan yang telah saya pilih.
11. Dosen dan karyawan Jurusan Teknik Mesin FTI-ITS yang telah memberikan ilmu, pengalaman, dan bantuannya selama masa studi.
12. Serta semua pihak yang secara langsung ataupun tidak langsung terlibat dalam proses penyelesaian tugas akhir ini.

Dengan selesainya Tugas Akhir ini, penulis berharap Tugas Akhir ini dapat bermanfaat khususnya bagi penulis sendiri dan umumnya untuk kita semua.

Surabaya, Juli 2018

Mas'ud Asadullah

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam sejarah dunia, peperangan menjadi hal yang sering terjadi dalam meraih kemerdekaan negara, pada peperangan yang telah terjadi banyak menimbulkan korban jiwa. Sejarah mencatat pada perang dunia 1 menyebabkan korban jiwa sebanyak 17.661.000 jiwa dan pada perang dunia 2 menyebabkan korban jiwa sebanyak 62.537.400 jiwa.

Indonesia adalah salah satu negara yang mendapakan kemerdekaan dari peperangan melawan penjajah saat perang dunia 2 terjadi dengan korban jiwa mencapai 4.000.000 jiwa. Karena itu, kemerdekaan indonesia dijadikan sebagai suatu hal yang patut di jaga kedaulatan. Kedaulatan suatu negara haruslah dijaga dengan baik. Sumber daya yang banyak dibutuhkan untuk menjaga kedaulatan indonesia itu sendiri, terutama dalam persenjataan.

Senjata otomatis menjadi suatu hal yang tepat untuk menjaga kedaulatan negara. Tanpa adanya kontrol manual dan mampu mendeteksi musuh sendiri, senjata otomatis diharapkan mampu meminimalisir jumlah korban manusia. Agar senjata otomatis ini memiliki akurasi yang baik, sistem kontrol diharuskan mampu bereaksi dengan cepat dan tepat. Salah satu sistem kontrol yang memiliki karakteristik tersebut adalah sistem pengendalian PI.

Sistem pengendalian PI adalah sistem dengan 2 parameter, yaitu parameter *proportional gain* (K_P), *integral gain* (K_I). 2 parameter tersebut memiliki nilai yang tetap pada tiap persamaannya, mengakibatkan persamaan tersebut tidak dapat mewakili seluruh sistem yang ingin di kendalikan. Metode *neural network* (NN) menjadi salah satu solusi dari permasalahan diatas. Metode *neural network* berfungsi untuk mendapatkan persamaan *gain* yang sesuai dengan ouput yang diinginkan oleh sistem pengendali.

Penelitian tentang PI dan NN pada sistem pengendali *turret gun* telah dilakukan pada Penelitian sebelumnya. Penelitian tersebut meneliti tentang sistem pengendali PI dan NN yang disimulasikan menggunakan matlab [1]. Hasil penelitian ini diperoleh bahwa PI-NN memiliki respon yang lebih stabil. Oleh karena itu, tugas akhir ini berfokus untuk menghasilkan rancangan sistem pengendalian menggunakan PI-NN pada *prototype turret gun* secara langsung.

1.2 Rumusan Masalah

Rumusan masalah dalam laporan tugas akhir ini adalah sebagai berikut:

1. Bagaimana hasil rancangan pengendalian PI-NN untuk *Prototype Turrent Gun.*

1.3 Batasan Masalah

Batasan masalah pada penulisan laporan tugas akhir ini adalah sebagai berikut;

1. Rancang bangun alat sistem *turret gun* tidak dibahas.
2. Gesekan dan *losses* pada sistem mekanik diabaikan.

1.4 Tujuan Penelitian

Adapun tujuan dari penulisan tugas akhir ini adalah:

1. Mendapatkan hasil rancangan pengendalian PI-NN untuk *prototype turret gun.*

1.5 Manfaat Penelitian

Tugas akhir ini diharapkan dapat memberi manfaat diantaranya sebagai berikut:

1. Memperkaya dan memperdalam wacana dalam bidang ilmu mekatronika khususnya mengenai sistem *auto-tuning controller*.
2. Diperoleh pengetahuan sistem kendali *auto-tuning* yang diterapkan pada sistem *turret gun.*
3. Hasil rancang bangun alat ini dapat menjadi media dalam membantu pembelajaran mahasiswa Teknik Mesin untuk mata kuliah Sistem Dinamik dan Pengendalian Otomatis.

[Halaman ini sengaja dikosongkan]

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Perkembangan teknologi yang cukup pesat menyebabkan banyaknya penelitian yang dilakukan dibidang ini. Salah satunya adalah penelitian tentang sistem pengendali. Penelitian sebelumnya melakukan penelitian terhadap *permanent magnet synchronous motor* (PMSM) [2]. Karakter alat ini memiliki kecepatan tinggi, akurasi tinggi, dan kehandalan tinggi. Penelitian ini mencoba menggabungkan antara pengendali PID control dengan *neural network* (NN). Hal ini bertujuan agar kontrol PMSM bisa mengikuti waktu nyata. Terdapat 3 kesimpulan dari penelitian ini. PIDNN memanfaatkan keunggulan dari PID dan NN maka sistem akan lebih stabil. PIDNN memiliki respon cepat dan bisa menyesuaikan kecepatan referensi dengan akurasi steady state yang tinggi. Terakhir, pada kondisi kecepatan rendah PIDNN memiliki peforma yang lebih baik dari PID konventional.

Penelitian sistem pengendalian senjata juga pernah dilakukan pada tahun 2017. Penggabungan sistem pengendalian PI dengan metode NN dilakukan pada penelitian ini [1]. Penelitian ini dilakukan dengan mensimulasikan sistem pengendali PI-NN pada software matlab. 3 hasil penelitian diperoleh dari penelitian ini. Pertama, sistem *turret gun* dapat diberikan *controller* dengan menggunakan metode *neural network*. Berdasarkan hasil simulasi diperoleh grafik respon dengan *root mean square error* 0.0393063 untuk sudut rotasi dan 1.1621708 untuk sudut elevasi.

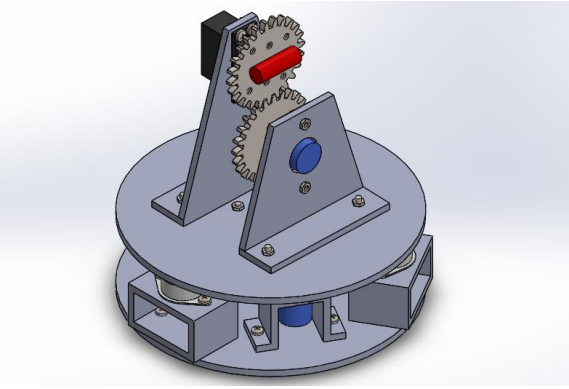
Kedua, nilai K_P dan K_I untuk sudut Rotasi dan sudut elevasi dapat berubah setiap waktu. Ketiga Respon sistem *turret-gun* dengan *Neural Network Proportional-Integral controller* dapat menurunkan *%overshoot* 63,493%, *settling time* 19,183%, dan *steady-state error* 27,064%. Sementara untuk sudut elevasi, *Neural Network Proportional-Integral controller* hanya dapat menurunkan *steady state error* yaitu 34,168 %. Sedangkan untuk

settling time mengalami kenaikan 16,463% dan *%overshoot* juga mengalami kenaikan 75,465%.

Penelitian tentang sistem kontrol PI dengan neural network juga dilakukan untuk mengetahui hubungan antara kedua teori pada tahun 2001 [3]. Hasil dari penelitian ini diperoleh rumus *performance index* untuk sistem kontrol PI. Fungsi rumus ini untuk menentukan data dengan nilai *performance index* rendah. Data tersebut akan dijadikan data pelatihan pada *neural network*.

2.2 *Prototype Turrent Gun*

Prototype turret-gun terdiri dari sistem *turret* dan *gun*. Untuk pengerak *turret-gun* menggunakan 2 motor servo, 1 motor servo untuk sudut rotasi dan 1 motor servo untuk sudut elevasi. *Turret* adalah bagian sistem yang bergerak pada sumbu rotasi. *Gun* adalah bagian sistem yang bergerak pada sumbu elevasi. Prototype ini juga terdapat sistem feedback yang menggunakan potensiometer sebagai pembacanya. Berikut bentuk dari *prototype turret-gun* pada gambar 2.1.



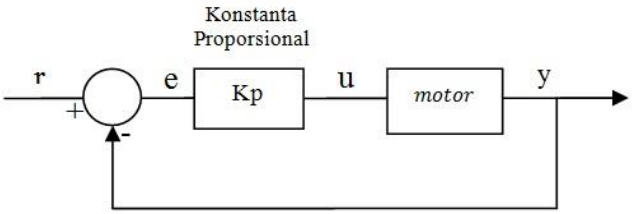
Gambar 2.1 Prototype *turret- gun*

Sudut maksimal *Prototype turret-gun* adalah 0 sampai 70 derajat untuk sudut elevasi dengan titik tengah ada di sudut 43 derajat. Sedangkan sudut rotasi memiliki 0 sampai 105 derajat dengan titik tengah ada di sudut 53 derajat. Sudut ini didapatkan dari penelitian sebelumnya.

2.3 *Proportional Control*

Pada jenis kontrol ini terdapat hubungan yang proporsional antara keluaran terhadap kesalahan (error). Memperbesar nilai kontrol ini dapat meningkatkan respon transient (memperkecil waktu penetapan proses) dari *prototype turret gun*. Tetapi pembesaran nilai Kp secara praktis tidak dapat dilakukan secara terus menurus karena pada penguatan tertentu, output proses akan menjadi kurang stabil. Persamaan *proportional control* dapat dilihat sebagai berikut:

$$u = K_p . e \qquad 2.1$$



Gambar 2.2 Diagram control proporsional [4]

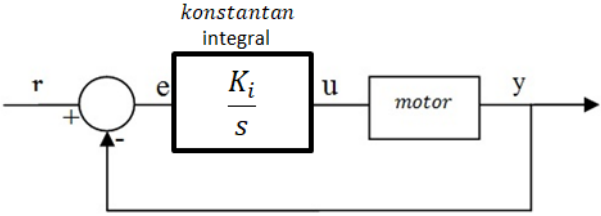
Gambar 2.2 menunjukkan blok diagram yang menggambarkan hubungan antara besaran referensi, besaran aktual dengan besaran input kontroler. Jika hasil besaran aktual telah sama dengan besaran referensi maka input kontroler akan nol. Artinya kontroler tidak akan lagi memberikan sinyal aktuasi kepada motor karena target besaran referensi telah diperoleh.

2.4 *Integral Control*

Fungsi dasar kontrol I adalah menurunkan *steady-state error*. Kontrol I jarang digunakan sendirian dalam aplikasi. Biasanya selalu dikombinasikan dengan kontrol P untuk memperbaiki respon guna mencapai eror minimum. Pengontrol *integral* berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan stabil nol. Jika sebuah Plant tidak memiliki unsur integrator (1/s), pengontrol proposional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan stabilnya nol. Dengan pengontrol *integral*, respon sistem dapat diperbaiki, yaitu mempunyai kesalahan keadaan stabilnya nol. Pengontrol *Integral* memiliki karaktiristik seperti halnya sebuah *integral*[1].

Kontrol I merupakan penjumlahan dari nilai *error* selama waktu tertentu, sehingga Kontrol I dapat memperbaiki nilai *error* yang didapat. Nilai dari *error* yang telah terakumulasi kemudian dikalikan dengan Ki dan ditambahkan pada *output* pengendali, hal ini dapat dilihat dari persamaan berikut:

$$u = K_i \int_0^t e(t)dt \qquad 2.2$$



Gambar 2.3 Diagram *control integral* [4]

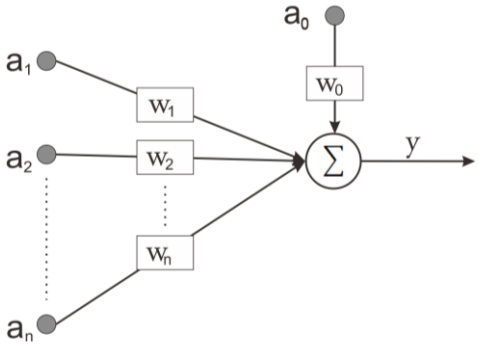
Gambar 2.3 menunjukkan blok diagram yang menggambarkan hubungan antara besaran referensi, besaran aktual dengan besaran input kontroler. Jika hasil besaran aktual telah sama dengan besaran referensi maka input kontroler akan nol. Artinya kontroler tidak akan lagi memberikan sinyal aktuasi kepada motor karena target besaran referensi telah diperoleh.

2.5 *Neural Network*

Neural network adalah salah satu model komputasi yang terdiri dari *neuron-neuron* buatan untuk memproses informasi dari input ke output. *Neuron-neuron* terdiri dari *synapses* dan *nodes*. Kumpulan dari node-node disebut *layer*. *Layer* dapat dibuat lebih dari satu. *Node* digunakan untuk menghimpun fungsi-fungsi.

Salah satu penggunaan *neural network* adalah untuk menentukan sebuah fungsi yang mendekati fungsi aslinya. Fungsi asli tersebut biasanya tidak diketahui karena sistem terlalu rumit untuk diturunkan ke dalam sebuah persamaan. Penggunaan *neural network* dimaksudkan untuk memperoleh persamaan *gain* dari sistem *control*. *Gain* tersebut akan mempengaruhi sistem sehingga kecepatan respon dari sistem akan meningkat[1].

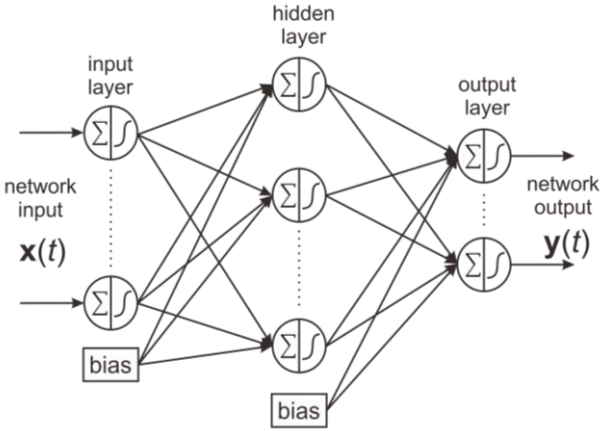
Selain itu, Penggunaan *neural network* tidak dibatasi dengan jumlah *layer* pada setiap pelatihannya. Jumlah *layer* ini mempengaruhi dalam permecahan masalah yang diberikan. Semakin banyak jumlah *layer* maka akan memperluas kemampuan *neural network*. Tetapi hal ini akan menyebabkan lamanya pemrosesan data pelatihan dan memperbanyak memori yang digunakan pada sistem.



Gambar 2.4 Dasar *Adaptive Linear Combiner* [6]

Gambar 2.4 menunjukkan struktur dari *Adaptive Linear Combiner* (ALC). Dengan input jaringan (*synapses* dan *nodes*) dari struktur di lambangkan dengan a, w sebagai nilai *weight*, dan output dari struktur ALC adalah y, serta secara umum a_0 memiliki nilai 1 dan disebut sebagai *bias*, maka persamaan output sama dengan berikut:

$$y = \sum_{i=1}^{n} a_i w_i = a^T w \qquad 2.3$$



Gambar 2.5 Jaringan *Multi-Layer Feedforward* [6]

Gambar 2.5 menunjukkan *Multi-Layer Artificial Neural Neworks* adalah jaringan *Feedforward* dengan sinyal input disebarkan ke depan melalui beberapa lapisan pemrosesan sebelum keluaran jaringan dihitung. Setiap lapisan terdiri dari sejumlah node, dan masing-masing node (umumnya) terdiri dari ALC sederhana, dengan fungsi transfer yang sesuai yang menghitung keluaran simpul dari sinyal masukan *weight*. Setiap node memiliki koneksi input dengan node pada layer sebelumnya saja, dan output node ditransmisikan ke node pada layer berikutnya, seperti yang ditunjukkan pada gambar 2.5. setiap node memiliki *weight* terkait yang secara linear mengubah vektor

inputnya. Maka persamaan yang biasa digunakan adalah *sigmoidal transfer function* yaitu:

$$f(u) = \frac{1}{1+\exp(-u)} \quad 2.4$$

Maka jika jumlah lines sama dengan 3 dimensi, persamaan menjadi.

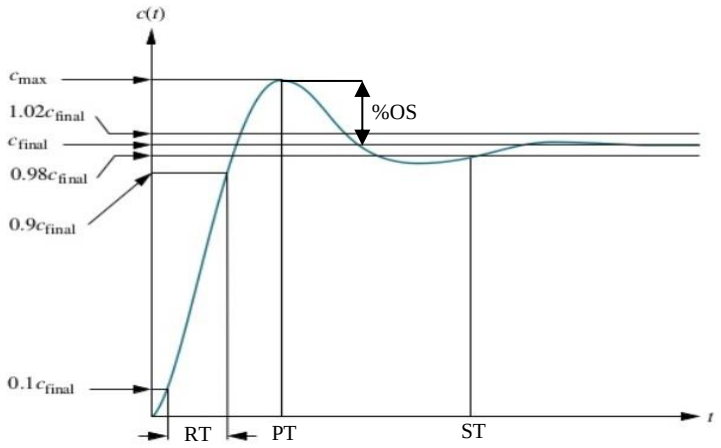
$$w_0 + a_1 w_1 + a_2 w_2 = c \quad 2.5$$

Dan jika menjadi n dimensi maka persamaan menjadi:

$$w_0 + a^T w = c \quad 2.6$$

2.6 Respon Sistem

Respon sistem atau tanggapan sistem merupakan perubahan perilaku output terhadap perubahan sinyal input. Respon sistem berupa kurva dan dapat menjadi dasar untuk menganalisa karakteristik sistem. Bentuk kurva respon sistem dapat dilihat setelah mendapatkan sinyal input.



Gambar 2.6 *Respon transient* [5]

Gambar 2.6 memperlihatkan contoh dari grafik *respon transient*. *Risetime*, *settling time*, dan *peak time* memberikan informasi mengenai kecepatan dan 'kualitas" respon transien. Besaran - besaran ini dapat membantu perancangan untuk mencapai kecepatan yang diinginkan tanpa osilasi atau *overshoot* yang berlebihan. Parameter yang menandakan kualitas respon *transient* antara lain:

- *Rise time (RT)*

Yaitu waktu yang diperlukan kurva untuk naik dari 0.1 ke 0.9 dari respon *steady state*.

- *Settling Time (ST)*

Yaitu waktu yang diperlukan kurva respon untuk mencapai 2% dari nilai *steady state*-nya.

- *Persen Overshoot (OS)*

Yaitu nilai relatif yang menyatakan perbandingan antara nilai tertinggi respon yang melebihi nilai *steady state* dibandingkan dengan nilai *steady state*.

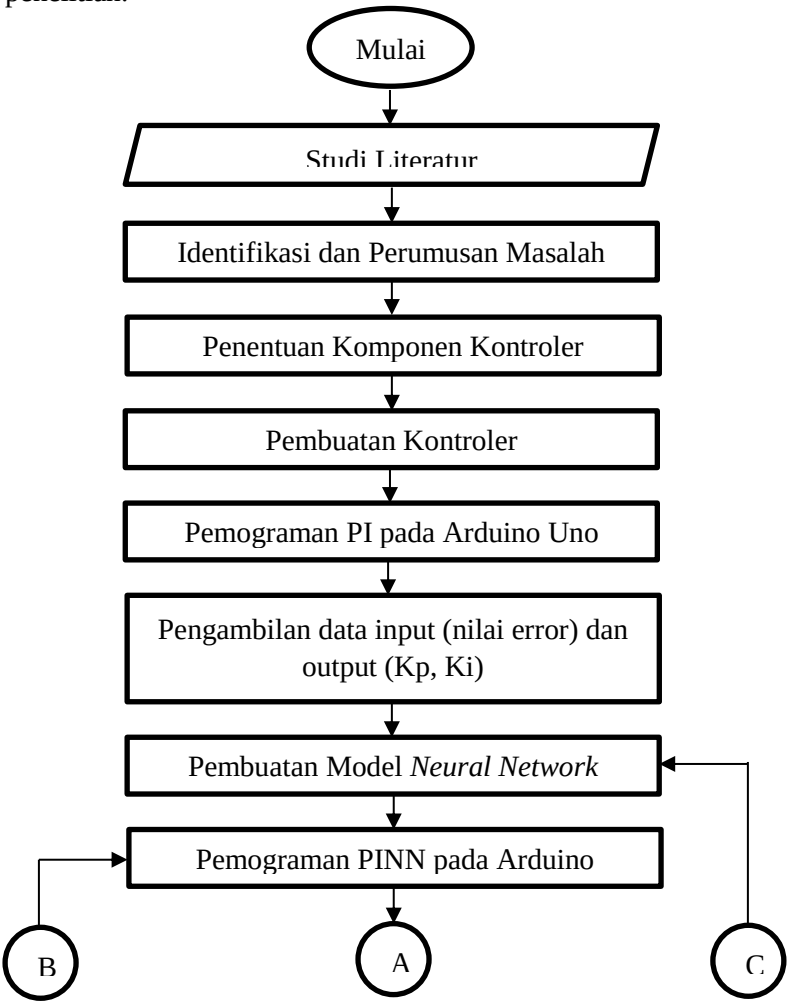
- *Time Peak (PT)*

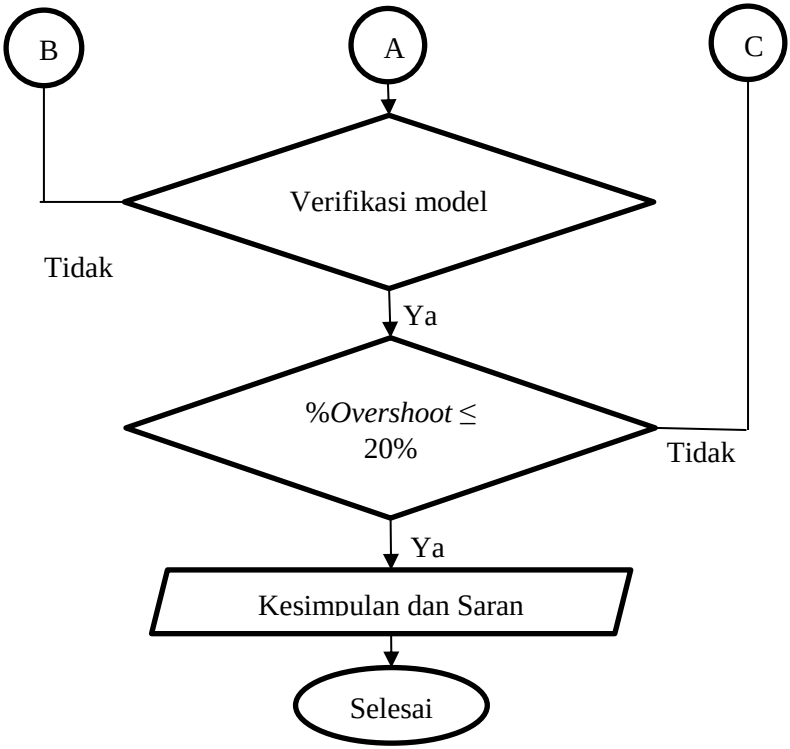
Yaitu waktu yang diukur mulai t = 0 hingga respon pertama kali mencapai puncak maksimum [5].

BAB III
METODE PENELITIAN

3.1 Diagram Alir Penelitian

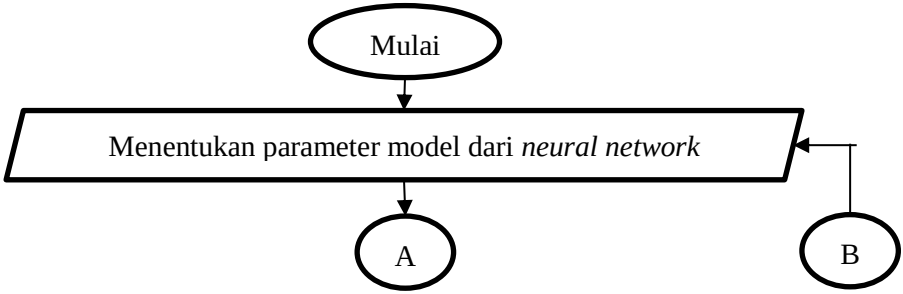
Berikut ini gambar 3.1 memperlihatkan *flowchart* diagram alir penelitian.

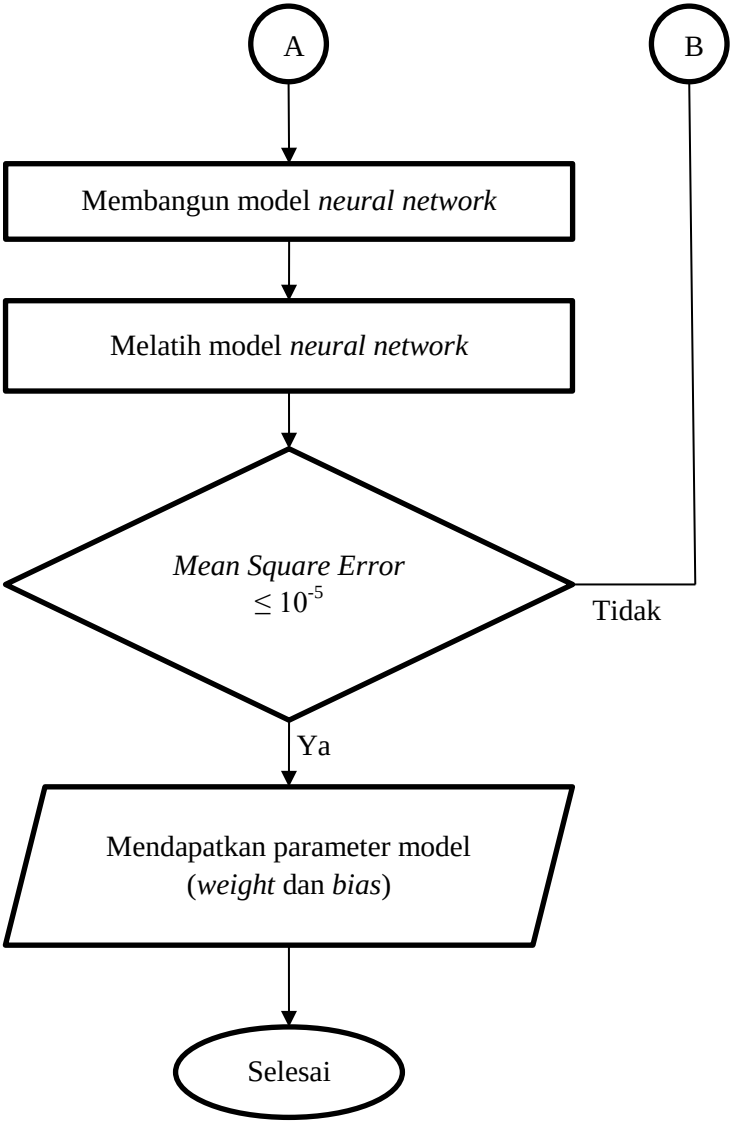




Gambar 3.1 Diagram Alir Penelitian

Dan berikut ini adalah gambar 3.2 memperlihatkan flowchart Diagram alir untuk membangun model *Neural network*.





Gambar 3.2 Diagram alir untuk pelatihan model *Neural Network*

3.2 Langkah-langkah Penelitian

Langkah-langkah penelitian secara garis besar pada tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

   Studi literatur bertujuan untuk meningkatkan pemahaman mengenai permasalahan yang dibahas pada tugas akhir ini. literatur-literatur yang digunakan terlebih dahulu disesuaikan dengan permasalahan yang dibahas, yaitu tentang sistem kontrol dan *neural network*. Studi literatur dilakukan dengan membaca buku teks, *e-book*, jurnal, dan penelitian terdahulu.

2. Perumusan Masalah

   Perumusan masalah ini disesuaikan dengan studi literatur yang telah dilakukan pada langkah sebelumnya. Kemudian menentuan permasalahan utama yang akan dibahas serta penguraian batasan-batasan masalah yang digunakan, Hal ini bertujuan agar penelitian dapat dilakukan dengan tujuan yang jelas dan sistematis.

3. Penentuan Komponen Kontroler

   Penentuan komponen kontroler bertujuan untuk menentukan komponen-komponen yang akan digunakan dalam penelitian ini. Penentuan komponen ini dilakukan dengan melihat dari segi fungsi, kualitas, harga dan sesuai dengan kebutuhan penelitian. Untuk itu diperlukan survey komponen.

4. Pembuatan Kontroler

   Setelah komponen yang dibutuhkan sudah ditentukan. Pembuatan kontroler dari komponen dilakukan dengan menyesuaikan dari mekanisme alat yang telah dibuat pada penelitian sebelumnya.

5. Pemrograman PI pada Arduino Uno

   Pemrograman arduino menjadi sistem yang menggerakkan seluruh *prototype turret gun*. Pemrograman pertama ini hanya menggunakan sistem pengendali PI saja.

Tujuannya adalah mendapatkan data sistem pengendali PI untuk dijadikan data pelatihan bagi *neural network*. Pada pemrograman PI ini difokuskan untuk menunjukkan hasil dari sistem respon sementara pada *prototype turret gun*.

6. Pengambilan data input dan output

Saat pemrograman Arduino sudah selesai, pengambilan data ouput dilakukan terhadap input yang dimasukkan. Pengambilan data input dan ouput dilakukan untuk 2 sudut, sudut rotasi dan sudut elevasi. Data sudut rotasi yang diambil adalah sudut dari 0 sampai 105 derajat dengan interval sudut pengembalian 15 derajat. Sedangkan untuk sudut elevasi diambil sudut dari 0 sampai 70 derajat dengan interval sudut pengambilan 10 derajat.

Pengambilan data input dan ouput juga menggunakan interval nilai untuk nilai Kp dan nilai Ki. Pengambilan nilai Kp antara 5 sampai 15 dengan interval nilai sebesar 5. Sedangkan untuk nilai Ki pada nilai 1 dan nilai 3. Setelah itu pengambilan data input dan ouput membutuhkan persamaan yang menyatakan peforma dari respon tersebut. Persamaan ini bertujuan untuk mendapatkan data input dan output yang akan digunakan dalam pembuatan model *neural network* agar mendapatkan nilai Kp dan Ki optimum, Berikut persamaan untuk adalah *performance index* (F) [6]:

$$\mathrm{F} = (k_1 \times \%Overshoot + k_2 \times RiseTime + k_3 \times steady\ state\ error) \qquad 3.1$$

Dimana:
$k_1 = 50$
$k_2 = 10$
$k_3 = 30$

Performance index tersebut tidak mutlak seperti persamaan 7, tetapi dapat diubah sesuai yang diinginkan. Pada persamaan diatas menitikberatkan peningkatan peforma

pada *risetime*. Namun, jika ingin meningkatkan peforma *%overshoot* dan *steady state error* maka nilai $k_1$dan k_3 dapat dibesarkan melebih nilai k_2. nilai k_1, k_2 dan k_3 adalah nilai kali dan dapat diubah sesuai dengan kondisi yang diinginkan.

7. Pembuatan Model *Neural network*

Untuk pembuatan model *neural network* terdapat langkah-langkah yang harus dilakukan yaitu:

a. Menentukan parameter model dari *neural network*

Pada tahap ini, dilakukan penentuan parameter-parameter dari model *neural network*. Parameter penentuan tersebut berupa *layer* dan *neuron* yang menjadi parameter utama dalam mempengaruhi ouput yang dihasilkan oleh *neural network.*

b. Membangun model *neural network*

Pada tahap ini, model *neural network* dibangun dengan memasukkan dan menentukan jumlah parameter yang telah ditentukan dalam langkah sebelumnya. Pada parameter *layer* menggunakan 1 sampai 3 *layer*. Sedangkan untuk parameter *neuron* menggunakan 5 sampai 15 *neuron* dengan interval 5 *neuron.*

c. Melatih model *neural network.*

Pelatihan model *neural network* dimaksudkan untuk mendapatkan nilai K_P dan K_I. Pada langkah ini, data input dan output dimasukkan kedalam *neural network* untuk diproses. Input untuk dimasukkan ke dalam pelatihan *neural network* diperoleh dari data error masing-masing sudut dari sistem yang sudah ada. Sementara output yang dimasukkan ke dalam pelatihan *neural network* adalah nilai K_P Rotasi, K_I Rotasi, K_P Elevasi, dan K_I Elevasi.

d. Mendapatkan parameter *weight* dan *bias*

Output yang dihasilkan dari pelatihan *neural network* adalah *weight* dan *bias* untuk *hidden layer* dan *layer* output. Kedua parameter tersebut adalah parameter yang

mempengaruhi persamaan dari *neural network* untuk menghasilkan nilai K_P dan K_I.

8. Pemrograman PINN pada Arduino

Pemrograman PINN pada arduino ini dilakukan untuk memasukkan hasil dari pelatihan *neural network*. Hasil pelatihan *neural network* berupa persamaan yang memiliki output berupa nilai kp dan nilai ki pada pemrograman PI sebelumnya. Bentuk persamaannya adalah persamaan matriks pada program arduino.

9. Verifikasi Model

Untuk menyempurnakan model, dibutuhkan verifikasi yang menyatakan jika model tersebut dapat digunakan. Verifikasi tersebut dilakukan dengan mencoba memberikan nilai input pada program. Nilai input yang dimasukkan adalah nilai input terkecil dan nilai terbesar pada masing-masing sudut. Jika program berjalan dengan baik dan hasil menunjukkan kurva respon sistem, maka verifikasi model berhasil.

10. Kesimpulan dan Saran

Kesimpulan dari penelitian Tugas Akhir dapat diambil jika tujuan dari tugas akhir ini sudah tercapai. Kesimpulan berupa hasil dari perancangan sistem pengendali PI-NN, bisa berbentuk persamaan ataupun respon sistem. Adapun saran dapat berupa masukan untuk penelitian selanjutnya yang terkait dengan sistem pengendalian pada *prototype turret gun*.

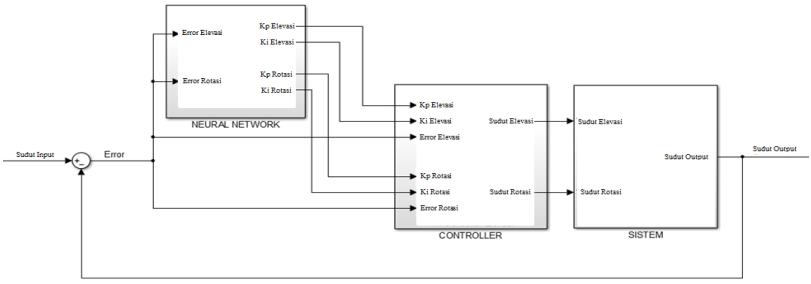
[Halaman ini sengaja dikosongkan]

BAB IV
PERANCANGAN SISTEM KENDALI *PROTOTYPE TURRET GUN*

4.1 Konfigurasi Sistem Pengendali *Prototype Turret Gun*

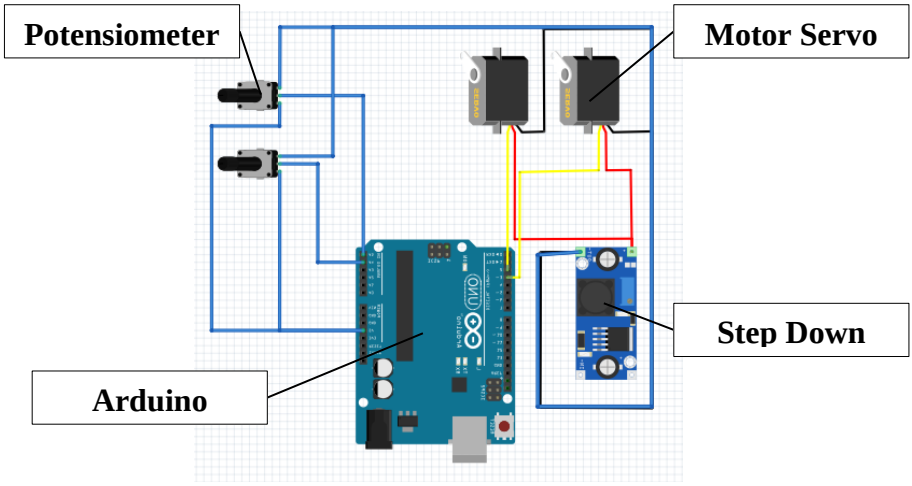
Pengaturan sudut elevasi dan sudut rotasi dilakukan dengan menggunakan motor servo, sedangkan untuk pembacaan keluaran dari motor servo menggunakan potensiometer. Untuk mengontrol motor servo dan potensiometer ini digunakan perangkat-perangkat pendukung seperti laptop, arduino uno, *step down*, dan program pendukung.

Pada pengoperasian arduino uno digunakan satu fungsi program yang telah disediakan dalam *library function* arduino yaitu *Library* servo.h. *Library* servo.h adalah pustaka pada bahasa C++ yang digunakan untuk mengoperasikan motor servo secara tepat. Tanpa menggunakan *library* ini maka perintah-perintah untuk motor servo tidak dapat dieksekusi. penggunaan *library-library* tersebut akan terlihat peran dan fungsinya ketika dikonfigurasikan dengan perangkat keras pendukungnya. Berikut adalah skema sistem kendali *prototype turret gun* pada gambar 4.1.



Gambar 4.1 Skema sistem kendali *Prototype turret gun*

Gambar 4.1 menunjukkan bahwa sistem kontrol yang digunakan pada *prototype turret gun* adalah sistem *close loop* dimana *input* berupa sudut yang didapatkan oleh kamera untuk diolah oleh *personal computer* (PC) melalui *software* sehingga dapat dikomunikasikan dengan mikrokontroller pada *prototype turret gun.* Setelah mendapatkan perintah, *prototype turret gun* akan bergerak sesuai dengan perintah yang diberikan. Setiap pergerakan yang dilakukan motor servo akan terbaca oleh potensiometer. Pembacaan potensiometer akan menjadi nilai output yang digunakan untuk mendapatkan *feedback* selisih sudut aktual dari nilai input dan output. Sementara konfigurasi sistem kontrol *prototype turret gun* dan perangkat kerasnya dapat dilihat pada gambar 4.2.



Gambar 4.2 Konfigurasi sistem kendali *prototype turret gun* dan perangkat kerasnya

Gambar 4.2 menunjukkan perangkat keras yang digunakan, yaitu 2 potensiometer, 2 motor servo, 1 arduino uno, dan 1 step down. Arduino uno sebagai mikrokontroler yang digunakan untuk menegndalikan motor servo dan membaca potensiometer. Motor

servo dihubungkan dengan arduino melalui pin 2 untuk rotasi dan pin 3 untuk elevasi. Sedangkan untuk potensiometer dihubungkan dengan Arduino melalui pin A4 untuk rotasi dan pin A5 untuk elevasi.

4.2 Penentuan Komponen Kontroler

Proses pembuatan perangkat keras (*hardware*) unit kontrol untuk alat *prototype turret gun* dengan dua motor servo dan dua potensiometer ini memerlukan beberapa macam komponen. Komponen yang digunakan dalam tugas akhir ini memiliki spesifikasi seperti pada tabel berikut ini.

1. Mikrokontroller Arduino Uno

Arduino Uno adalah arduino board yang menggunakan mikrokontroler ATmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menhubungkannya ke sebuah komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuanya bekerja. Arduino Uno menggunakan ATmega16U2 yang diprogram sebagai USB-to-serial converter untuk komunikasi serial ke computer melalui port USB. Tampak atas dari arduino uno dapat dilihat pada gambar 4.3.

Gambar 4.3 Papan Arduino Uno

Uno berbeda dari semua board mikrokontroler diawal-awal yang tidak menggunakan chip khusus driver FTDI USB-to-serial. Sebagai gantinya penerapan USB-to-serial adalah ATmega16U2 versi R2 (versi sebelumnya ATmega8U2). Versi Arduino Uno Rev.2 dilengkapi resistor ke garis ground. Berikut adalah spesifikasi arduino uno pada table 4.1.

Tabel 4.1 Spesifikasi Arduino Uno

Mikrokontroller	Atmega328
Operasi *Voltage*	5V
Input *Voltage*	7-12 V (Rekomendasi)
Input *Voltage*	6-20 V (limits)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

2. Potensiometer

Potensiometer (POT) adalah salah satu jenis Resistor yang Nilai Resistansinya dapat diatur sesuai dengan kebutuhan Rangkaian Elektronika ataupun kebutuhan pemakainya. Potensiometer merupakan Kelompok Resistor yang tergolong dalam Kategori Variable Resistor. Secara struktur, potensiometer terdiri dari 3 kaki terminal dengan sebuah tuas yang berfungsi sebagai pengaturnya. Potensiometer ini digunakan untuk menjadi sensor pergerakan dari motor servo. Gambar 4.4 dibawah ini menunjukkan bentuk potensiometer multitrun.

Gambar 4.4 Potensiometer

Sebuah Potensiometer (POT) terdiri dari sebuah elemen resistif yang membentuk jalur (track) dengan terminal di kedua ujungnya. Sedangkan terminal lainnya (biasanya berada di tengah) adalah Penyapu (Wiper) yang dipergunakan untuk menentukan pergerakan pada jalur elemen resistif (Resistive). Pergerakan Penyapu (Wiper) pada Jalur Elemen Resistif inilah yang mengatur naik-turunnya Nilai Resistansi sebuah Potensiometer. Elemen Resistif pada Potensionmeter umumunya terbuat dari bahan campuran metal dan keramik ataupun bahan karbon (*carbon*). Berdasarkan jalur elemen resistif-nya, potensiometer dapat digolongkan menjadi 2 jenis yaitu potensiometer linier dan potensiometer logaritmik.

3. Motor Servo

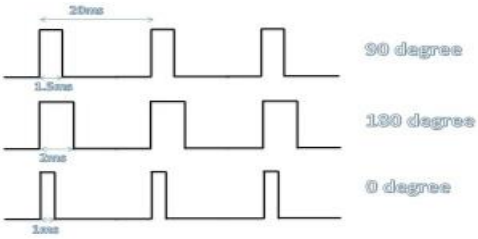
Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotor-nya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo. Gambar 4.5 dibawah ini menunjukkan motor servo yang digunakan.

Gambar 4.5 Motor Servo

Motor servo pada dasarnya dibuat menggunakan motor DC yang dilengkapi dengan controler dan sensor posisi sehingga dapat memiliki gerakan 0^o, 90^o, 180^o atau 360^o. Berikut adalah

komponen internal sebuah motor servo 180^{o}. Tiap komponen pada motor servo diatas masing-masing memiliki fungsi sebagai controler, driver, sensor, girbox dan aktuator. Pada gambar diatas terlihat beberapa bagian komponen motor servo. Motor pada sebuah motor servo adalah motor DC yang dikendalikan oleh bagian controler, kemudian komponen yang berfungsi sebagai sensor adalah potensiometer yang terhubung pada sistem girbox pada motor servo.

Untuk menjalankan atau mengendalikan motor servo berbeda dengan motor DC. Karena untuk mengedalikan motor servo perlu diberikan sumber tegangan dan sinyal kontrol. Besarnya sumber tegangan tergantyung dari spesifikasi motor servo yang digunakan. Sedangkan untuk mengendalikan putaran motor servo dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 5o Hz dengan periode 20ms dan duty cycle yang berbeda. Dimana untuk menggerakan motor servo sebesar 90^{o} diperlukan pulsa dengan ton duty cycle pulsa posistif 1,5ms dan unjtuk bergerak sebesar 180^{o} diperlukan lebar pulsa 2ms. Berikut bentuk pulsa kontrol motor servo dimaksud pada Gambar 4.6.



Gambar 4.6 Bentuk pulsa kendali motor servo

4. *Step Down* XL4005

Step down XL4005 yang ditunjukkan pada gambar 4.7 adalah jenis Step Down DC to DC seri XL4005 yang berfungsi untuk menurunkan dan menstabilkan tegangan DC. Kelebihan modul XL4005 ini adalah di kemampuannya menahan beban arus cukup besar, hingga 5A. sehingga bisa digunakan untuk sumber

power bertenaga besar. Penggunaan *Step down* pada tugas akhir ini bertujuan untuk menstabilkan tegangan dari *power supply* yang akan masuk kedalam motor servo. Besar tegangan untuk 2 motor servo adalah 5 volt untuk tiap motor servo.

Gambar 4.7 *Step down* XL4005

5. *Power Supply*

Power Supply atau dalam bahasa indonesia disebut dengan catu daya adalah suatu alat listrik yang dapat menyediakan energi listrik untuk perangkat listrik ataupun elektronika lainnya yang ditunjukkan pada gambar 4.8. Oleh karena itu, *power supply* kadang-kadang disebut juga dengan istilah *Electric Power Converter*. Pada tugas akhir ini digunakan 1 buah *power supply* dengan arus 3A dan voltase 12 Volt.

Gambar 4.8 Power supply

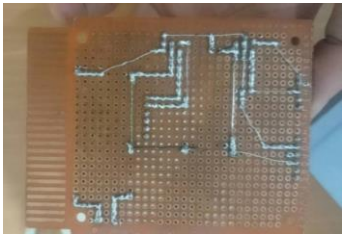
4.3 Pembuatan Kontroler

Dalam pembuatan alat *prototype turret gun* ada beberapa tahapan yang dilakukan untuk merakit seluruh komponen yang telah dipilih. Tahapan-tahapan ini diharapkan dapat

mempermudah dalam pembuatan dan pengecekan alat di setiap kondisinya. Tahapan-tahapan perakitan alat yaitu:

1. Pembuatan PCB.

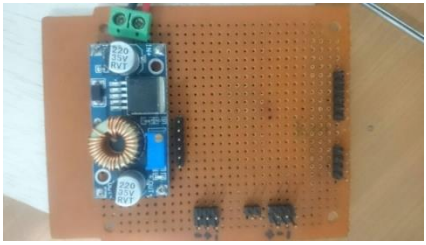
Pembuatan PCB (*Printed Circuit Board*) bertujuan untuk tempat peletakan komponen elektrik yang akan di gunakan seperti Arduino, *Step down*, dan lain-lain. Serta untuk membuat rangkaian jalur-jalur konduktor sebagai penghubung antara satu komponen dengan komponen lainnya yang terlihat pada gambar 4.9.



Gambar 4.9 Papan PCB

2. Pemasangan arduino dan *step down* pada PCB.

Pada PCB yang telah dibuat terdapat *pin header* yang dipasang sesuai bagian komponen yang ingin dipasang. 2 pin PWM untuk output motor servo, 2 pin analog untuk input potensiometer, dan 2 pin power untuk potensiomer. Sedangkan untuk *step down* dilakukan solder langsung pada papan PCB yang terlihat pada gambar 4.10 dan gambar 4.11.



Gambar 4.10 PCB sebelum dipasang komponen

Gambar 4.11 PCB setelah dipasang komponen.

3. Menghubungkan motor servo dan potensiometer pada PCB.

Motor servo dan potensiometer dihubungkan ke Arduino seperti yang ditunjukkan gambar 4.12. dari 3 pin yang ada pada motor servo dan potensiometer terbagi menjadi 1 pin untuk tegangan, 1 pin untuk *ground*, dan 1 pin untuk output/input. Pin tersebut dipasang ditempat yang telah disediakan pada PCB.

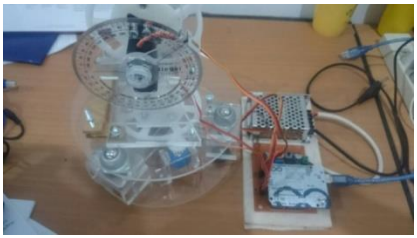
Gambar 4.12 Motor servo dan potensiometer dihubungkan di PCB.

4. Menghubungkan *power supply* dengan *step down*.

Motor servo memerlukan catu daya sebesar 5 volt dan arus sebesar 2 amper untuk menggerakkan dua buah motor servo. Pada kutub negatif dihubungkan ke pin ground pada *step down* dan kutub positif dihubungkan ke pin tegangan. Pemasangan komponen ini dituntukkan pada gambar 4.13.

Gambar 4.13 Power supply dihubungkan dengan *step down*.

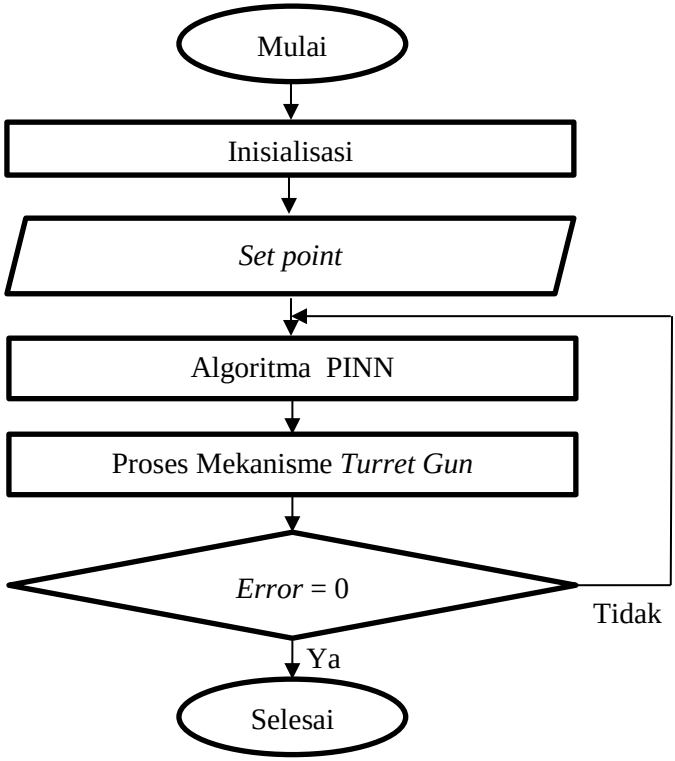
Seluruh rangkaian *Prototype turret gun* dapat dilihat pada gambar 4.14, namun *Prototype turret gun* belum mampu bekerja karena dibutuhkan kode pemrograman untuk membuatnya bergerak sesuai dengan yang diinginkan, kode pemrograman ini akan dibahas pada subab berikutnya.



Gambar 4.14 Rangkaian *Prototype turret gun*

4.4 Pemrograman PI pada Arduino Uno

Sebelum membuat program pada arduino uno, pembuatan *flowchart* proses pemrograman harus dilakukan terlebih dahulu. *Flowchart* ini berfungsi sebagai acuan dasar dalam pemrograman yang akan dilakukan lihat gambar 4.15. Selain itu dalam merancang suatu program perlu diperhatikan banyaknya proses yang dilakukan oleh program. Hal ini akan mempengaruhi lamanya proses pengerjaan. Karena itu meminimalkan program harus diutamakan.



Gambar 4.15 Flowchart pemrograman arduino

1. Deklarasi Variabel

Tahap pertama yang harus dilakukan adalah menginput variabel yang akan digunakan dalam pemrograman. Deklarasi variabel bertujuan untuk mengenalkan program dengan variabel-variabel yang akan digunakan dalam program. Variabel yang dimasukkan dalam program ini terdapat variabel untuk potensiometer, PID, serta input dan output untuk motor servo. Ada beberapa macam tipe variabel yang digunakan, seperti int, string, boolean, dan lain-lain tergantung kebutuhan. Berikut gambar 4.16 menunjukkan variabel-variabel yang digunakan:

```
#include <Servo.h>

Servo servo_rot, servo_elev;

int pos_rot, pos_elev,pos_rotl, pos_elevl;
int data;
String dataIn;
String dt[10];
int k;
boolean parsing=false;
double kp, ki,kpl,kil;
float error, error_i, p, i, pi,error_last;
float errorl, error_il, pl, il, pil,error_lastl;
int  as,rotasi=0,elevasi=0,prev_rotasi=0,prev_elevasi=0;
int simpan_error=0,simpan_errorl=0, ti, til;
int sel1,sel2,tet1,tet2,tet11,tet22,min_s,min_s1,P,O;
int h=0;
int x=0;
```

Gambar 4.16 Dekralasi variabel dan tipe variabelnya.

2. Pembacaan Potensiometer

Pada tahap ini dilakukan pemrograman untuk potensiometer. Potensiometer untuk rotasi dihubungkan dengan pin A4 arduino, sedangkan potensiometer untuk elevasi dihubungkan dengan pin A5 arduino. Dalam pemrograman potensiometer juga dilakukan kalibrasi pada tiap potensiometer. Karena potensiometer yang digunakan adalah tipe *multitrun* yaitu potensiometer dengan 10 putaran, maka jumlah derajatnya adalah 3600 derajat. Hal ini dapat dilihat pada gambar 4.17, yaitu :

```
void read_position(){
  P=(analogRead(A4));
  O=(analogRead(A5));
  pos_rot=(802-analogRead(A4));  //7
  pos_elev=(296 -analogRead(A5))*-1;
  pos_rotl=(pos_rot/1.2571429);
  if(pos_rotl<0){pos_rotl=0;}
  pos_elevl=(pos_elev/1.5);
  if(pos_elevl<0){pos_elevl=0;}
}
```

Gambar 4.17 Pemrograman potensiometer

3. Program Parsing Data

Parsing data adalah program untuk mengurai data masukan yang akan menjadi inputan bagi program arduino. Parsing data dilakukan dengan cara mengecek tiap karakter yang ada pada variabel dataIn. Tiap karakter tersebut akan dibandingkan dengan *tail* (karakter terakhir paket data) yang berupa karakter pagar (#) dan karakter pemisah antar data yang berupa karakter koma(,). Pengecekan dimulai dari index 1 (karakter kedua) variabel dataIn, hal ini dikarenakan index 0 (karakter pertama) merupakan *header* (karakter pertama) dalam paket data.

Setiap karakter yang telah melalui proses pengecekan, akan ditampung kedalam sebuah variabel string yang dikelompokkan dalam sebuah array yang diberi nama dt. Jika pada proses pengecekan ditemukan karakter koma (,), maka dengan otomatis variabel penampung data setelah dicek akan berpindah dengan cara increment index array nya. Proses tersebut akan terus dilakukan berulangkali sampai perulangan berakhir. Setelah proses parsing data selesai, data hasil proses parsing yang terdapat pada array dt akan ditampilkan. Dengan begitu, proses parsing data pun selesai dilakukan. Hal ini dapat dilihat pada gambar 4.18, yaitu :

```
void parsingData(){
  int j=0;
  dt[j]="";
  for(k=1;k<dataIn.length();k++){
    if ((dataIn[k] == ',')){
      j++;
      dt[j]="";
    }
    else{
      dt[j] = dt[j] + dataIn[k];
    }
  }
  rotasi=(52 - dt[0].toInt());
  if(rotasi<=-52){rotasi=-52;}
  else if(rotasi>=52){rotasi=52;}
  elevasi= 35-dt[1].toInt();
  if(elevasi<=-35){elevasi=-35;}
  else if(elevasi>=35){elevasi=35;}
```

Gambar 4.18 Pemrograman parsing data

4. Program *Neural Network* (NN) dan PI

Gambar 4.19 menunjukkan pemrograman untuk persamaan *neural network* yang akan digunakan. Persamaan *neural network* didapatkan dari pengelolaan data pada matlab. *Layer* dan *neuron* yang digunakan akan mempengarauhi bentuk dari persamaan *neural network*. Gambar 4.19 menunjukkan persamaan *neural network* yang menggunakan 2 layer, yaitu 1 *hidden layer* dengan 1 *layer output*. Banyaknya *layer* dan *neuron* akan menyebabkan persamaan *neural network* menjadi panjang.

```
void NN () {
    //Rotasi 13,8
    M11 = 1/(1+exp(-((simpan_error*0.2098)+(-13.9853))));
    M12 = 1/(1+exp(-((simpan_error*0.2459)+(-6.9377))));
    M13 = 1/(1+exp(-((simpan_error*1.1965)+(0.0097583))));
    M14 = 1/(1+exp(-((simpan_error*0.3)+(6.9004))));
    M15 = 1/(1+exp(-((simpan_error*-0,8741)+(15.0562))));

    M16 = (M11*1.2106)+(M12*3.2238)+(M13*2.10943)+(M14*4.4342)+(M15*2.09457)+1.1328;
    M17 = (M11*(-1.0668))+(M12*0.039197)+(M13*0.021397)+(M14*(0.001137))+(M15*(2.0445))+0.9163;

    //Elevasi 13,8
    M21 = 1/(1+exp(-((simpan_error1*0.7862)+(-13.0858))));
    M22 = 1/(1+exp(-((simpan_error1*0.5961)+(-5.246))));
    M23 = 1/(1+exp(-((simpan_error1*0.0405)+(-5.101))));
    M24 = 1/(1+exp(-((simpan_error1*0.3932)+(-6.2996))));
    M25 = 1/(1+exp(-((simpan_error1*0.145)+(14.0498))));

    M26 = (M21*2.035687)+(M22*2.333)+(M23*1.70936)+(M24*3.0569)+(M25*2.591377)+1.93093;
    M27 = (M21*(-1.5594))+(M22*2.778)+(M23*2.9177)+(M24*(-1.0020563))+(M25*(-1.00017516))+1.85618;

}
```

Gambar 4.19 Pemrograman Persamaan *Neural Network*

Gambar 4.20 menunjukkan pemograman untuk sistem kontrol PI. Program sistem pengendali PI di bagi menjadi 2 macam, program PI untuk sudut rotasi dan program PI untuk sudut elevasi. Program PI ini disesuaikan dengan rumus dari masing - masing sistem pengendali PI yaitu sistem pengendali *proportional* dan sistem pengendali *integral* dengan nilai Kp dan Ki yang telah ditentukan.

Program PINN merupakan gabungan dari program NN dan PI. Output dari program NN akan diberikan pada program PI. Output tersebut adalah nilai Kp dan Ki yang telah diolah pada program NN. nilai Kp dan Ki yang diberikan akan dikalikan

dengan nilai error dari selisih antara *set point* dengan hasil pembacaan potensiometer.

```
//Algoritma PID
void pid(){
//rotasi
  kp = output2[0][0];
  ki = output2[1][0];
  error = simpan_error;
  error_i = error-error_last ;
  p = kp * error;
  i = ki * error_i;
  pi=abs(p+i);
  error_last = error;

//elevasi
  kp1 = output21[0][0];
  ki1 = output21[1][0];
  error1 = simpan_error1;
  error_i1 = error1-error_last1 ;
  p1 = kp1 * error1;
  i1 = ki1 * error_i1;
  pi1=abs(p1+i1);
  error_last1 = error1;
}
```

Gambar 4.20 Pemrograman PI

5. Program Utama

Program utama adalah program yang mejalankan program secara berulang pada program arduino biasa ditulis dengan *void loop*. Program ini berisi seluruh program yang sebelumnya sudah di jelaskan dengan memberikan istilah untuk setiap program. Misalanya untuk program parsing data diwakilkan dengan istilah parsingData(). Begitu juga dengan program PI dan pembacaan potensiometer yang masing-masing diwakili dengan pid() dan read_position().

Selain itu, program ini juga berisi rumus untuk mencari nilai error pada sudut rotasi dan elevasi disetiap pengulangan yang terjadi. Program ini juga mengatur pergerkan dari sudut rotasi dan sudut elevasi pada *prototype turret gun*. Pengaturan ini dilakukan agar pergerakan dari sudut rotasi dan sudut elevasi terlihat bergerak secara bersamaan. Jadi pergerakan yang dihasilkan adalah pergerakan yang bergantian dari sudut rotasi ke sudut

elevasi dengan rentang sudut tertentu. Hal ini dapat dilihat pada gambar 4.21, yaitu :

```
void loop(){
  prev_rotasi=rotasi;
  prev_elevasi=elevasi;
  if(Serial.available()>0) {
    char inChar = (char)Serial.read();
    dataIn += inChar;
    if (inChar == '\n') {
      parsing = true;
    }
  }
  if(parsing){
    parsingData();
    parsing=false;
    dataIn="";
  }
  for(h=0;h<=x;h++){
  //sel1=rotasi- prev_rotasi;
  //sel2=elevasi- prev_elevasi;
  sel1=rotasi- pos_rot1;
  sel2=elevasi-pos_elev1;
  if(sel1<0){sel1=sel1*(-1);}
  if(sel2<0){sel2=sel2*(-1);}
  min_s=sel1;
  min_s1=sel2;;
  simpan_error=min_s;
  simpan_error1=min_s1;
  if(min_s==0){min_s=1;}
  if(min_s1==0){min_s1=1;}
  x=min(min_s,min_s1);
  //
   pid();

     //tet1=prev_rotasi+(((rotasi-prev_rotasi)/x)*h);
     //tet2=prev_elevasi+(((elevasi-prev_elevasi)/x)*h);
     tet1=prev_rotasi+(((rotasi- pos_rot1)/x)*h);
     tet2=prev_elevasi+(((elevasi-pos_elev1)/x)*h);
     tet11=50+tet1;
     tet22=tet2;
     ti=5000/(pi+1);
     if(ti==5000){ti=1;}
     ti1=5000/(pi1+1);
     if(ti1==5000){ti1=1;}
     delay(ti);
     servo_rot.write(tet11);
     delay(ti1);
     servo_elev.write(tet22);
     read_position();
```

Gambar 4.21 Pemrograman Utama

BAB V
PENENTUAN PARAMETER GAIN DENGAN NEURAL NETWORK

5.1 Pengambilan Data Input dan Output

Pengambilan data dilakukan dengan melakukan uji coba langsung pada prototy*pe turret gun* yang telah dibuat. *prototype* ini memiliki input referensi berupa sudut yaitu sudut rotasi dan sudut elevasi. Sistem pengendali yang terdapat pada *prototype* tersebut adalah PID *controller*. Input yang digunakan untuk PID *controller* adalah error yang terjadi pada *prototype*. Output dari *controller* adalah sudut. Sudut tersebut digunakan sebagai input untuk *prototype turret gun*. Output dari sistem adalah sudut rotasi dan sudut elevasi.

Blok PID *controller* memiliki persamaan PID. Persamaan tersebut terdapat parameter *proportional gain* (Kp), *Integral gain* (Ki), dan *derivative gain* (Kd). Nilai tersebut adalah faktor pengali untuk nilai error yang terjadi pada sistem. Untuk kebutuhan pengambilan data pada *prototype*, K nilai yang diubah adalah nilai Kp dan Ki.

Pengambilan data dilakukan dengan menyusun kemungkinan error yang dihasilkan oleh *prototype*. Untuk sudut rotasi terdapat error maksimal yaitu 105, sementara untuk sudut elevasi terdapat error maksimum yaitu 70.Sehingga, pengambilan data sudut rotasi diberikan rentang nilai error dari 0 sampai 105 dengan interval 15. Sedangkan untuk sudut elevasi diberikan rentang nilai error dari 0 sampai 70 dengan interval 10.

Masing-masing nilai error tersebut disimulasikan dengan memberikan nilai Kp dan Ki pada masing-masing sudut. Nilai Kp ditentukan pada rentang nilai 5 sampai 15 dengan interval 5 dan nilai Ki antara 1 dan 3. Karakteristik respon yang diambil pada setiap pengambilan data adalah *Settling Time*, *%overshoot*, *Rise time*, dan *Steady-state Error*. Pengambilan data di ulang sebanyak 5 kali untuk setiap kondisi. Kemudian nilai-nilai tersebut dihitung

dengan menggunakan persamaan 3.1 sehingga diperoleh data *performance index* (F). Data dari nilai F tersebut diambil data yang memiliki nilai F terkecil pada masing-masing nilai error di setiap percobaan. Data tersebut menjadi input untuk pelatihan *neural network*. Output yang digunakan adalah nilai Kp rotasi, Ki rotasi, Kp Elevasi, dan Ki Elevasi. Data output diperoleh dari nilai Kp dan Ki pada masing-masing nilai error yang mempunyai nilai F paling kecil.

5.2 Hasil Pengambilan Data

Sesuai dengan persamaan 3.1, parameter yang diambil adalah *settling time* (ST), %*overshoot* (OS), *rise time* (RT), dan *steady-state error* (SSE). Dengan kombinasi nilai error pada masing-masing sudut, nilai Kp rotasi, Ki rotasi, Kp elevasi, dan Ki elevasi, diperoleh data sebanyak 84 data. Data tersebut kemudian dipilih sesuai dengan nilai *performance index* (F) yang mempunyai nilai terkecil pada setiap nilai *set point*. Sehingga diperoleh data yang ditunjukkan pada tabel 5.1 dan tabel 5.2.

Tabel 5.1 Hasil Pengambilan Data untuk Sudut Rotasi

ROTASI							
Set Point	Kp	Ki	ST	OS	RT	SSE	F
15	5	1	0,76	0,00	0,28	0,00	2,81
	5	3	1,02	0,00	0,40	5,00	153,99
	10	1	0,27	0,00	0,13	1,00	31,32
	10	3	0,50	26,67	0,17	3,00	1425,02
	15	1	0,79	13,33	0,11	2,00	727,81
	15	3	0,33	6,67	0,17	0,00	335,06
30	5	1	1,03	20,00	0,43	5,00	1154,26
	5	3	0,87	0,00	0,48	4,00	124,84
	10	1	0,30	20,00	0,20	3,00	1092,01

	10	3	0,37	10,00	0,21	2,00	562,06
	15	1	1,53	40,00	0,26	1,00	2032,63
	15	3	0,64	10,00	0,18	2,00	561,81
45	5	1	3,50	22,22	0,54	0,00	1116,54
	5	3	2,01	26,67	0,54	2,00	1398,74
	10	1	1,65	26,67	0,27	4,00	1456,06
	10	3	1,57	26,67	0,31	4,00	1456,44
	15	1	1,21	28,89	0,19	1,00	1476,37
	15	3	0,83	8,89	0,19	2,00	506,35
60	5	1	3,06	16,67	0,59	3,00	929,22
	5	3	1,83	26,67	0,61	1,00	1369,47
	10	1	1,36	20,00	0,31	2,00	1063,12
	10	3	1,21	20,00	0,32	1,00	1033,18
	15	1	2,17	18,33	0,20	2,00	978,67
	15	3	0,47	3,33	0,19	2,00	228,57
75	5	1	2,02	17,33	0,67	2,00	933,33
	5	3	1,25	13,33	0,57	1,00	702,37
	10	1	2,09	20,00	0,29	4,00	1122,93
	10	3	1,67	18,67	0,30	0,00	936,28
	15	1	1,34	20,00	0,20	2,00	1062,01
	15	3	0,43	4,00	0,19	2,00	261,90
90	5	1	1,78	22,22	0,66	4,00	1237,70
	5	3	1,49	14,44	0,61	2,00	788,27
	10	1	2,30	21,11	0,32	4,00	1178,78
	10	3	3,44	4,44	0,36	1,00	255,83
	15	1	0,95	16,67	0,22	3,00	925,52
	15	3	1,21	6,67	0,21	3,00	425,38
105	5	1	1,36	0,00	0,66	0,00	6,57
	5	3	0,73	0,00	0,59	1,00	35,86

	10	1	1,45	1,90	0,31	2,00	158,38
	10	3	1,19	0,00	0,32	0,00	3,20
	15	1	0,88	3,81	0,20	5,00	342,49
	15	3	0,41	0,00	0,20	1,00	32,03

Tabel 5.2 Hasil Pengambilan Data untuk Sudut Elevasi

Elevasi							
Set Point	Kp	Ki	ST	OS	RT	SSE	F
10	5	1	0,55	0,00	0,46	5,00	154,55
	5	3	1,37	0,00	0,36	3,00	93,61
	10	1	0,57	0,00	0,35	1,00	33,50
	10	3	0,57	0,00	0,29	3,00	92,86
	15	1	0,58	0,00	0,27	2,00	62,67
	15	3	0,43	0,00	0,24	4,00	122,42
20	5	1	2,38	0,00	0,41	3,00	94,14
	5	3	0,93	0,00	0,47	4,00	124,65
	10	1	0,51	0,00	0,29	3,00	92,89
	10	3	1,31	0,00	0,29	0,00	2,89
	15	1	0,82	5,00	0,13	0,00	251,29
	15	3	0,78	0,00	0,17	4,00	121,69
30	5	1	0,63	0,00	0,45	2,00	64,47
	5	3	0,66	0,00	0,37	4,00	123,67
	10	1	1,78	0,00	0,26	0,00	2,58
	10	3	0,85	3,33	0,24	1,00	199,09
	15	1	1,00	6,67	0,17	1,00	365,01
	15	3	0,79	16,67	0,23	2,00	895,60
40	5	1	0,72	0,00	0,49	0,00	4,92
	5	3	0,788	2,50	0,53	1,00	160,25

	10	1	1,62	2,50	0,25	0,00	127,52
	10	3	1,06	10,00	0,31	2,00	563,12
	15	1	0,73	2,50	0,20	1,00	156,96
	15	3	1,12	10,00	0,20	3,00	591,98
50	5	1	1,18	14,00	0,55	2,00	765,46
	5	3	1,80	10,00	0,58	2,00	565,76
	10	1	1,60	4,00	0,27	1,00	232,66
	10	3	2,47	16,00	0,30	2,00	862,96
	15	1	0,86	10,00	0,20	2,00	562,03
	15	3	0,58	14,00	0,19	4,00	821,89
60	5	1	1,89	5,00	0,58	1,00	285,84
	5	3	0,89	20,00	0,56	0,00	1005,60
	10	1	1,78	8,33	0,30	1,00	449,70
	10	3	1,53	5,00	0,32	2,00	313,22
	15	1	1,52	5,00	0,20	1,00	281,97
	15	3	0,95	18,33	0,21	2,00	978,79
70	5	1	2,20	11,43	0,54	4,00	696,78
	5	3	1,86	4,29	0,59	0,00	220,18
	10	1	1,03	2,86	0,31	1,00	175,94
	10	3	2,25	5,71	0,32	1,00	318,90
	15	1	1,41	2,86	0,19	0,00	144,75
	15	3	1,40	10,00	0,20	0,00	501,99

Hasil dari Tabel 5.1 dan 5.2 akan diambil untuk masing-masing set point yang memiliki nilai F terkecil. Pengambilan nilai F terkecil ini diharapkan bias memberikan nilai *settling time* (ST), *%overshoot* (OS), *rise time* (RT), dan *steady-state error* (SSE) yang kecil juga untuk pelatihan *neural network*. Hasil pemilahan dapat di lihat pada tabel 5.3 dan tabel 5.4.

Tabel 5.3 Data untuk pelatihan *neural network* untuk sudut rotasi

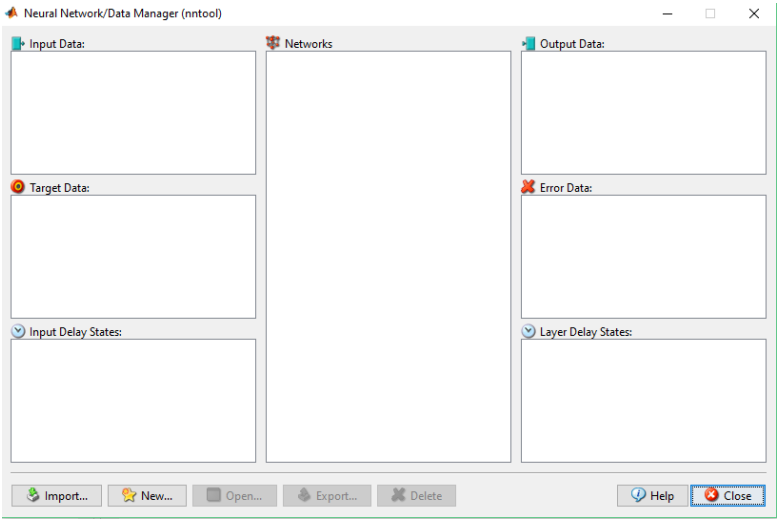
ROTASI							
Set Point	Kp	Ki	ST	OS	RT	SSE	F
15	5	1	0,76	0,00	0,28	0,00	2,81
30	5	3	0,87	0,00	0,48	4,00	124,84
45	15	3	0,83	8,89	0,19	2,00	506,35
60	15	3	0,47	3,33	0,19	2,00	228,57
75	15	3	0,43	4,00	0,19	2,00	261,90
90	10	3	3,44	4,44	0,36	1,00	255,83
105	10	3	1,19	0,00	0,32	0,00	3,20

Tabel 5.4 Data untuk pelatihan *neural network* untuk sudut elevasi

ELEVASI							
Set Point	Kp	Ki	ST	OS	RT	SSE	F
10	10	1	0,57	0,00	0,35	1,00	33,50
20	10	3	1	0,00	0,29	0,00	2,89
30	10	1	1,78	0,00	0,26	0,00	2,58
40	5	1	0,72	0,00	0,49	0,00	4,92
50	10	1	1,60	4,00	0,27	1,00	232,66
60	15	1	1,52	5,00	0,20	1,00	281,97
70	15	1	1,41	2,86	0,19	0,00	144,75

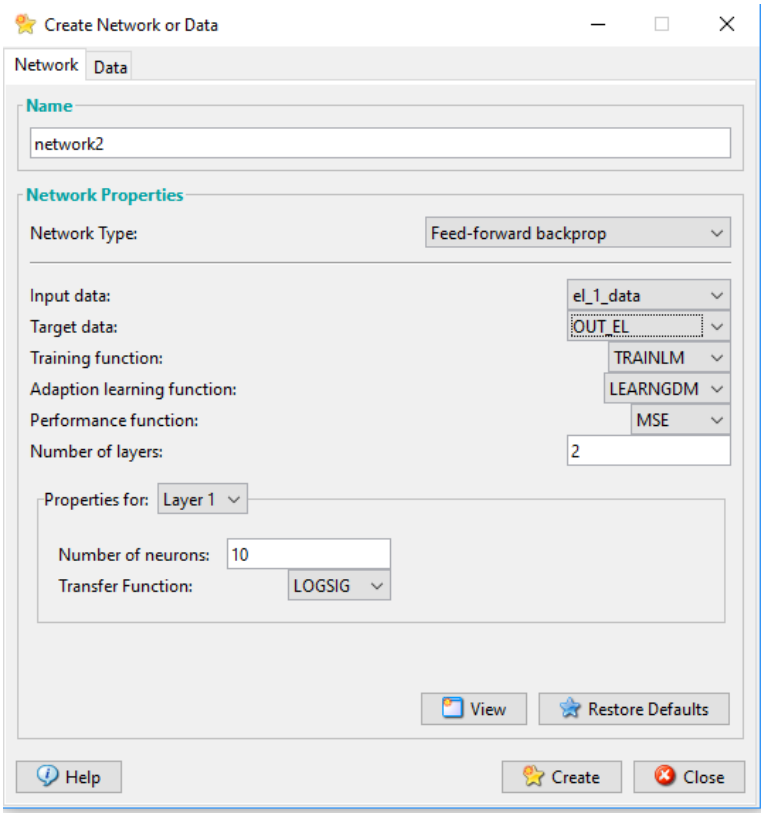
5.3 Pembuatan Model *Neural Network*

Pembuatan *neural network* dilakukan dengan *software* matlab. Pada *software* matlab terdapat *tool* berupa *data manager* untuk mengambil, membuat, menggunakan, dan memindahakan networks dan data. Dengan menggunakan syntax nntool, *data manager* dapat dipanggil seperti gambar dibawah ini:



Gambar 5.1 Toolbox Data Manager

Langkah selanjutnya yaitu memasukkan data untuk pelatihan *neural network*. Data yang digunakan adalah Data pada tabel 5.3 dan 5.4. Nilai *set point* tersebut dimasukkan sebagai data input, sedangkan nilai Kp dan Ki dimasukkan sebagai data target. Untuk membuat *neural network* dilakukan dengan memilih *new* pada *toolbox data manager*, seperti pada gambar 5.1. Kemudian akan muncul jendela *toolbox create network or data* untuk menentukan model *neural network* yang diinginkan, seperti menentukan tipe dari *neural network*, jumlah *layer* dan *neuron*, dan lain-lain, seperti pada gambar 5.2.



Gambar 5.2 Toolbox Create Network or Data

5.4 Pelatihan Neural Network untuk Sudut Rotasi

Untuk sudut rotasi digunakan *neural network feed-forward back Propagation* dengan satu input *layer*, satu *hidden layer* dan satu output *layer*. Pada *hidden layer* terdapat 5 *neuron* dan pada output *layer* terdapat 2 *neuron*. Kombinasi *layer* tersebut diperoleh dengan *trial* dan *error* pada pelatihan *neural network*. Hasil *trial* dan *error* adalah sebagai berikut:

Tabel 5.5 Hasil *trial* dan *error Neural Netwrok* untuk sudut rotasi

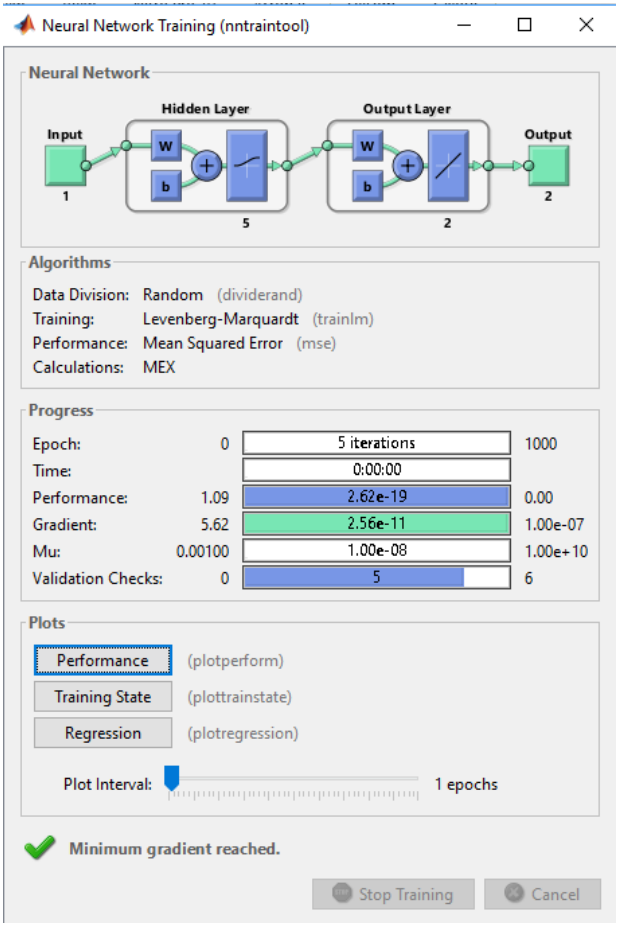
ROTASI						
No.	Input Layer	Hidden Layer	Neuron Hidden Layer	Output Layer	Neuron output layer	MSE
1	1	1	5	1	2	2,62E-19
2	1	1	10	1	2	6,66E-20
3	1	1	15	1	2	7,94E-21
4	1	2	5	1	2	1,65E-18
5	1	2	10	1	2	6,88E-20
6	1	2	15	1	2	7,27E-22
7	1	3	5	1	2	4,23E-14
8	1	3	10	1	2	7,00E-22
9	1	3	15	1	2	3,86E-25

Tabel 5.5 adalah hasil *trial* dan *error* untuk sudut rotasi yang dilakukan sebanyak sembilan kali. Percobaan tersebut dilakukan dengan mengubah jumlah *hidden layer* dan jumlah *neuron* pada *hidden layer*. Pada percobaan ini, *hidden layer* yang digunakan adalah 1 *hidden layer* sampai 3 *hidden layer*. *Neuron hidden layer* yang digunakan adalah 5 sampai 15 dengan interval 5 *neuron*. Dari percobaan ini diperoleh *mean square error* (MSE) terkecil adalah $3{,}86 \times 10^{-25}$.

Dengan kombinasi *layer* dan *neuron* diatas, diperoleh performa terbaik dari *neural network* yang diukur dengan *mean square error* (MSE) yaitu $3{,}86 \times 10^{-25}$. Karena nilai MSE pada 9 percobaan ini sudah berada dalam batas penelitian, maka penelitian ini menggunakan 1 *hidden layer* dengan 5 *neuron*, hal ini bertujuan untuk mengurangi besar memori arduino yang digunakan agar dapat mempercepat pemrosesan informasi pada program arduino.

Adapun parameter-parameter yang membatasi pelatihan *neural network* adalah gradien maksimum yang diatur pada nilai 10^{-10}, jumlah iterasi dibatasi hingga mencapai nilai 1000, dan

validasi diatur di angka enam. Hasil dari performa *neural network* setelah pelatihan dapat dilihat pada gambar 5.3.



Gambar 5.3 Performa dari neural network sudut rotasi setelah pelatihan

Gambar 5.3 adalah hasil pengujian dari *neural network* untuk sudut rotasi. Perfoma yang diukur adalah *mean square error*.

Setelah menempuh 5 kali iterasi, diperoleh MSE sebesar 2,62 × 10^{-19}, gradien sebesar 2,56 × 10^{-11}, dan validasi *neural network* sebanyak 5 kali. Hasil dari pelatihan tersebut diperoleh persamaan *neural network*. Persamaan tersebut terdapat nilai *weight (w)* dan *bias* (w_0). nilai *weight* dan *bias* berbentuk nilai matriks, yaitu:

$$w^{(1)} = \begin{bmatrix} 0{,}2098 \\ 0{,}2459 \\ 1{,}1965 \\ 0{,}3 \\ -0{,}8741 \end{bmatrix}$$

$$w_0{}^{(1)} = \begin{bmatrix} -13{,}9853 \\ -6{,}9377 \\ 0{,}0097583 \\ 6{,}9004 \\ 15{,}0562 \end{bmatrix}$$

Dari perhitungan nilai *weight* dan *bias* pada *hidden layer* dihasilkan matriks 5×1. Perhitungan tersebut diperoleh dari persamaan berikut:

$$u = \left(w^{(1)} \times input\right) + w_0{}^{(1)} \qquad 5.1$$

Nilai z akan melewati fungsi aktivasi persamaan 2.4. persamaan sebagai berikut:

$$f(u) = \frac{1}{1+\exp(u)} \qquad 5.2$$

Sedangkan pada output *layer*, *weight* dan *bias neural network* adalah sebagai berikut:

$$w^{(2)} = \begin{bmatrix} 1{,}2106 & 3{,}2238 & 2{,}10943 & 4{,}4342 & 2{,}09457 \\ -1{,}0668 & 0{,}039197 & 0{,}021397 & 0{,}001137 & 2{,}0445 \end{bmatrix}$$

$$w_0{}^{(2)} = [1{,}1328 \quad 0{,}9163]$$

Dari perhitungan nilai *weight* dan *bias* pada output *layer* dihasilkan matriks 2×1. Perhitungan tersebut diperoleh dari persamaan berikut:

$$y = \left(w^{(2)} \times f(u)\right) + w_0{}^{(2)} \qquad 5.3$$

Sehingga persamaan *neural network controller* untuk sudut rotasi adalah:

$$\begin{bmatrix} k_p \\ k_i \end{bmatrix} = \left(w^{(2)} \times f(u)\right) + w_0{}^{(2)} \qquad 5.4$$

5.5 Pelatihan Neural Network untuk Sudut Elevasi

Untuk sudut elevasi digunakan *neural network feed-forward back Propagation* dengan satu input *layer*, satu *hidden layer* dan satu output *layer*. Pada *hidden layer* terdapat 5 *neuron* dan pada output *layer* terdapat 2 *neuron*. Kombinasi *layer* tersebut diperoleh dengan *trial* dan *error* pada pelatihan *neural network*. Hasil *trial* dan *error* adalah sebagai berikut:

Tabel 5.6 Hasil *trial* dan *error Neural Network* untuk sudut elevasi

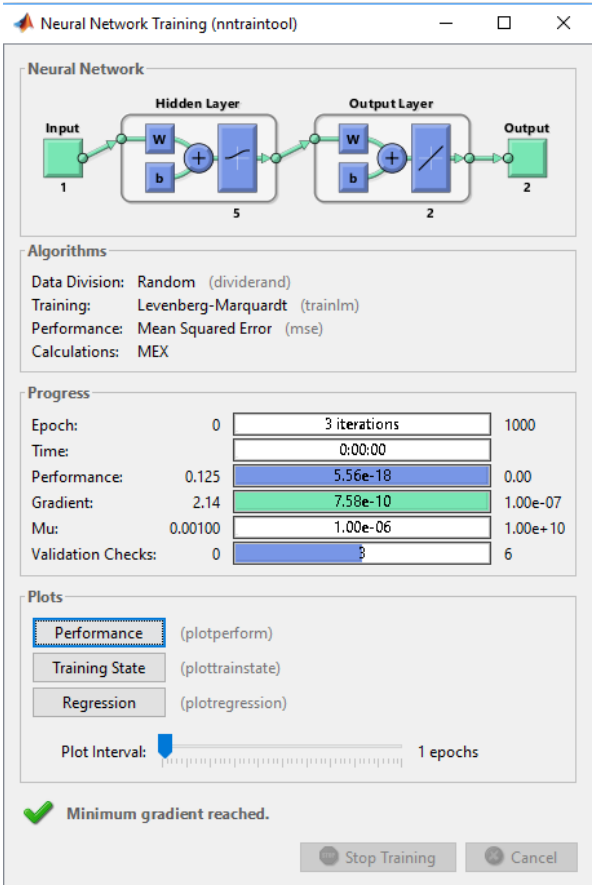
ELEVASI						
No.	Input Layer	Hidden Layer	Neuron Hidden Layer	Output Layer	Neuron output layer	MSE
1	1	1	5	1	2	5,56E-18

2	1	1	10	1	2	1,79E-20
3	1	1	15	1	2	8,38E-20
4	1	2	5	1	2	1,78E-17
5	1	2	10	1	2	3,71E-22
6	1	2	15	1	2	8,91E-22
7	1	3	5	1	2	3,08E-19
8	1	3	10	1	2	1,95E-21
9	1	3	15	1	2	3,14E-24

Tabel 5.6 adalah hasil *trial* dan *error* untuk sudut elevasi yang dilakukan sebanyak sembilan kali. Percobaan tersebut dilakukan dengan mengubah jumlah *hidden layer* dan jumlah *neuron* pada *hidden layer*. Pada percobaan ini, *hidden layer* yang digunakan adalah 1 *hidden layer* sampai 3 *hidden layer*. *Neuron hidden layer* yang digunakan adalah 5 sampai 15 dengan interval 5 *neuron*. Dari percobaan ini diperoleh *mean square error* (MSE) terkecil adalah $3{,}14 \times 10^{-24}$.

Dengan kombinasi *layer* dan *neuron* diatas, diperoleh performa terbaik dari *neural network* yang diukur dengan *mean square error* (MSE) yaitu $3{,}14 \times 10^{-24}$. Karena nilai MSE pada 9 percobaan ini sudah berada dalam batas penelitian, maka penelitian ini menggunakan 1 *hidden layer* dengan 5 *neuron*, hal ini bertujuan untuk mengurangi besar memori arduino yang digunakan agar dapat mempercepat pemrosesan informasi pada program arduino.

Adapun parameter-parameter yang membatasi pelatihan *neural network* adalah gradien maksimum yang diatur pada nilai 10^{-10}, jumlah iterasi dibatasi hingga mencapai nilai 1000, dan validasi diatur di angka enam. Hasil dari performa *neural network* setelah pelatihan dapat dilihat pada gambar 5.4.



Gambar 5.4 Performa dari neural network untuk sudut elevasi setelah pelatihan

Gambar 5.4 adalah hasil pengujian dari *neural network* untuk sudut elevasi. Perfoma yang diukur adalah *mean square error*. Setelah menempuh 3 kali iterasi, diperoleh MSE sebesar $5{,}56 \times 10^{-18}$, gradien sebesar $7{,}58 \times 10^{-10}$, dan validasi *neural network* sebanyak 3 kali. Hasil dari pelatihan tersebut diperoleh persamaan

neural network. Persamaan tersebut terdapat nilai *weight (w)* dan *bias* (w_0). nilai *weight* dan *bias* berbentuk nilai matriks, yaitu:

$$w^{(1)} = \begin{bmatrix} 0{,}7862 \\ 0{,}5961 \\ 0{,}0405 \\ 0{,}3932 \\ 0{,}145 \end{bmatrix}$$

$$w_0{}^{(1)} = \begin{bmatrix} -13{,}0858 \\ -5{,}246 \\ -5{,}101 \\ -6{,}2996 \\ 14{,}0498 \end{bmatrix}$$

Dari perhitungan nilai *weight* dan *bias* pada *hidden layer* dihasilkan matriks 5×1. Perhitungan tersebut diperoleh dari persamaan berikut:

$$u = \left(w^{(1)} \times input\right) + w_0{}^{(1)} \qquad 5.5$$

Nilai z akan melewati fungsi aktivasi persamaan 2.4. persamaan sebagai berikut:

$$f(u) = \frac{1}{1+\exp(u)} \qquad 5.6$$

Sedangkan pada output *layer*, *weight* dan *bias neural network* adalah sebagai berikut:

$$w^{(2)} = \begin{bmatrix} 2{,}035687 & 2{,}333 & 1{,}70936 & 3{,}0569 & 2{,}591377 \\ -1{,}5594 & 2{,}773 & 2{,}9177 & -1{,}0020563 & -1{,}00017516 \end{bmatrix}$$

$$w_0^{(2)} = [1{,}93093 \quad 1{,}85618]$$

Dari perhitungan nilai *weight* dan *bias* pada output *layer* dihasilkan matriks 2×1. Perhitungan tersebut diperoleh dari persamaan berikut:

$$y = \left(w^{(2)} \times f(u)\right) + w_0^{(2)} \qquad 5.7$$

Sehingga persamaan *neural network controller* untuk sudut elevasi adalah:

$$\begin{bmatrix} k_p \\ k_i \end{bmatrix} = \left(w^{(2)} \times f(u)\right) + w_0^{(2)} \qquad 5.8$$

5.6 Pengujian dan Verifikasi Model PINN Controller pada Prototype

Pengujian program pada *prototype turret gun* dilakukan dengan memberikan input dari serial monitor. Serial monitor disediakan dari software arduino. Pengambilan data dimulai dengan memberikan program pada arduino untuk memunculkan data pengulangan dari program. Data tersebut terdiri dari nilai error, waktu perubahan, dan sudut output dari potensiometer. Setiap pengujian pengambilan data yang dilakukan akan diambil 100 data pengulangan yang terjadi, lihat tabel 5.7.

Tabel 5.7 Contoh tabel 100 Data Pengulangan pada 1 kali Pengambilan Data

no, Data	ROTASI 45					
	error	waktu per-data (ms)	waktu gerak (s)	Kp	Ki	Sudut Potensiometer
1	0,00	5,00	0,01	8,72	2,92	0,00
2	0,00	5,00	0,01	8,72	2,92	0,00

3	45,00	5,17	0,02	12,96	2,96	0,00
4	45,00	6,85	0,02	12,96	2,96	0,00
5	45,00	6,85	0,03	12,96	2,96	0,00
6	45,00	6,85	0,04	12,96	2,96	0,00
7	45,00	6,85	0,04	12,96	2,96	0,00
8	45,00	6,85	0,05	12,96	2,96	0,00
9	45,00	6,85	0,06	12,96	2,96	0,00
10	45,00	6,85	0,06	12,96	2,96	0,00
11	45,00	6,85	0,07	12,96	2,96	0,00
12	45,00	6,85	0,08	12,96	2,96	0,00
13	45,00	6,85	0,08	12,96	2,96	0,00
14	45,00	6,85	0,09	12,96	2,96	0,00
15	45,00	6,85	0,10	12,96	2,96	0,00
16	45,00	6,85	0,10	12,96	2,96	0,00
17	45,00	6,85	0,11	12,96	2,96	0,00
18	45,00	6,85	0,12	12,96	2,96	0,00
19	45,00	6,85	0,12	12,96	2,96	0,82
20	44,18	7,03	0,13	12,94	2,96	2,46
21	42,54	7,36	0,14	12,91	2,96	2,46
22	42,54	7,27	0,15	12,91	2,96	4,92
23	40,08	7,92	0,15	12,83	2,97	4,92
24	40,08	7,76	0,16	12,83	2,97	6,57
25	38,43	8,26	0,17	12,76	2,97	6,57
26	38,43	8,14	0,18	12,76	2,97	9,03
27	35,97	9,03	0,19	12,58	2,97	9,03
28	35,97	8,82	0,20	12,58	2,97	11,49
29	33,51	9,92	0,21	12,31	2,96	11,49
30	33,51	9,68	0,22	12,31	2,96	13,13
31	31,87	10,57	0,23	12,06	2,95	13,13

32	31,87	10,38	0,24	12,06	2,95	15,59
33	29,41	12,04	0,25	11,62	2,95	16,41
34	28,59	12,31	0,26	11,46	2,95	18,05
35	26,95	13,60	0,27	11,13	2,95	34,47
36	10,53	112,16	0,39	9,81	2,85	43,50
37	1,50	163,35	0,55	9,48	2,93	42,68
38	2,32	148,87	0,70	9,65	2,93	43,50
39	1,50	338,41	1,04	9,48	2,93	42,68
40	2,32	148,87	1,19	9,65	2,93	42,68
41	2,32	170,67	1,36	9,65	2,93	43,50
42	1,50	338,41	1,70	9,48	2,93	43,50
43	1,50	262,28	1,96	9,48	2,93	43,50
44	1,50	262,28	2,22	9,48	2,93	43,50
45	1,50	262,28	2,48	9,48	2,93	42,68
46	2,32	148,87	2,63	9,65	2,93	42,68
47	2,32	170,67	2,80	9,65	2,93	43,50
48	1,50	338,41	3,14	9,48	2,93	42,68
49	2,32	148,87	3,29	9,65	2,93	43,50
50	1,50	338,41	3,63	9,48	2,93	42,68
51	2,32	148,87	3,78	9,65	2,93	43,50
52	1,50	338,41	4,11	9,48	2,93	42,68
53	2,32	148,87	4,26	9,65	2,93	42,68
54	2,32	170,67	4,43	9,65	2,93	42,68
55	2,32	170,67	4,61	9,65	2,93	42,68
56	2,32	170,67	4,78	9,65	2,93	42,68
57	2,32	170,67	4,95	9,65	2,93	43,50
58	1,50	338,41	5,28	9,48	2,93	43,50
59	1,50	262,28	5,55	9,48	2,93	42,68
60	2,32	148,87	5,70	9,65	2,93	43,50

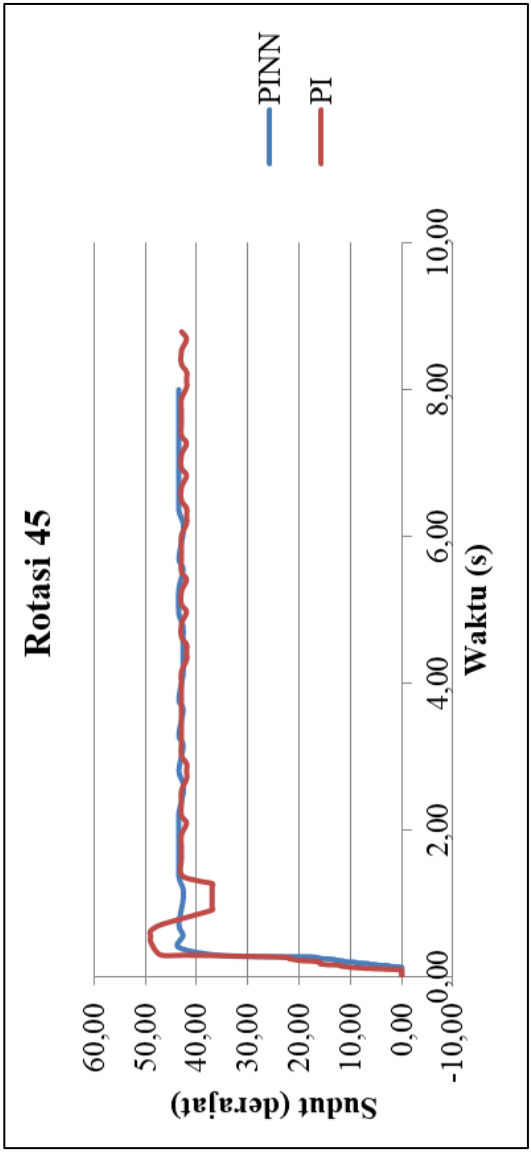
61	1,50	338,41	6,03	9,48	2,93	42,68
62	2,32	148,87	6,18	9,65	2,93	42,68
63	2,32	170,67	6,35	9,65	2,93	43,50
64	1,50	338,41	6,69	9,48	2,93	43,50
65	1,50	262,28	6,95	9,48	2,93	43,50
66	1,50	262,28	7,22	9,48	2,93	43,50
67	1,50	262,28	7,48	9,48	2,93	43,50
68	1,50	262,28	7,74	9,48	2,93	43,50
69	1,50	262,28	8,00	9,48	2,93	43,50
70	1,50	262,28	8,27	9,48	2,93	43,50
71	1,50	262,28	8,53	9,48	2,93	43,50
72	1,50	262,28	8,79	9,48	2,93	43,50
73	1,50	262,28	9,05	9,48	2,93	43,50
74	1,50	262,28	9,32	9,48	2,93	43,50
75	1,50	262,28	9,58	9,48	2,93	43,50
76	1,50	262,28	9,84	9,48	2,93	42,68
77	2,32	148,87	9,99	9,65	2,93	42,68
78	2,32	170,67	10,16	9,65	2,93	43,50
79	1,50	338,41	10,50	9,48	2,93	43,50
80	1,50	262,28	10,76	9,48	2,93	43,50
81	1,50	262,28	11,02	9,48	2,93	43,50
82	1,50	262,28	11,28	9,48	2,93	43,50
83	1,50	262,28	11,55	9,48	2,93	43,50
84	1,50	262,28	11,81	9,48	2,93	42,68
85	2,32	148,87	11,96	9,65	2,93	42,68
86	2,32	170,67	12,13	9,65	2,93	43,50
87	1,50	338,41	12,47	9,48	2,93	43,50
88	1,50	262,28	12,73	9,48	2,93	43,50
89	1,50	262,28	12,99	9,48	2,93	42,68

90	2,32	148,87	13,14	9,65	2,93	42,68
91	2,32	170,67	13,31	9,65	2,93	42,68
92	2,32	170,67	13,48	9,65	2,93	43,50
93	1,50	338,41	13,82	9,48	2,93	43,50
94	1,50	262,28	14,08	9,48	2,93	43,50
95	1,50	262,28	14,34	9,48	2,93	43,50
96	1,50	338,41	14,68	9,48	2,93	43,50
97	1,50	262,28	14,95	9,48	2,93	43,50
98	1,50	262,28	15,21	9,48	2,93	42,68
99	2,32	148,87	15,36	9,65	2,93	42,68
100	2,32	170,67	15,53	9,65	2,93	42,68

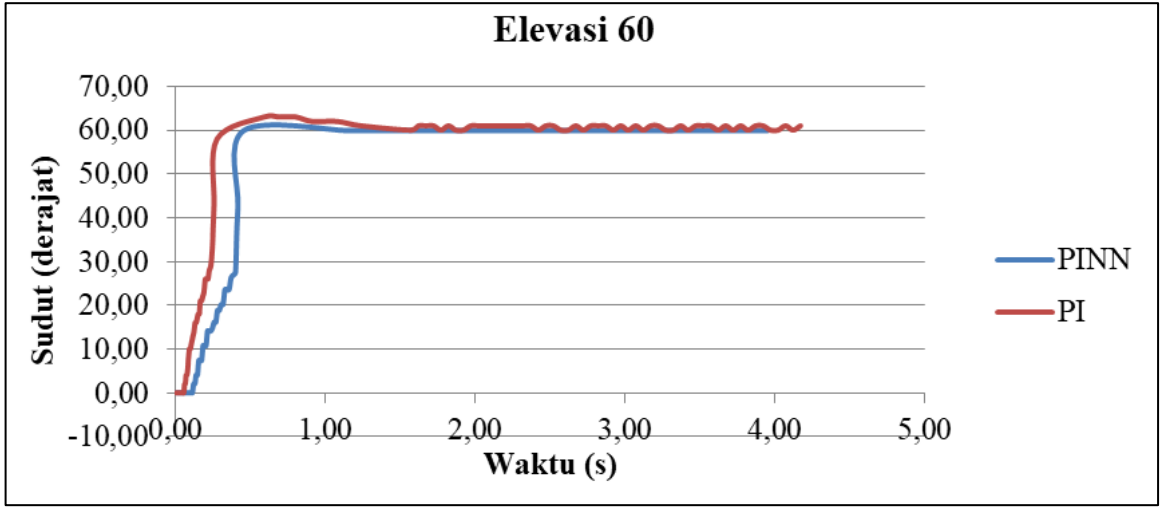
Tabel 5.7 menunjukkan perubahan dari pengulangan data 1 sampai pengulangan data ke 100 untuk percobaan 1 *set point*. Data tersebut digunakan untuk mendapatkan nilai *Settling Time*, *%overshoot, Rise time,* dan *Steady-state Error*. Nilai tersebut didapat dengan teori respon sistem pada bab 2. Tabel 5.5 juga meunjukkan bentuk grafik dari respon yang didapat, lihat gambar 5.5 menunjukkan pebandingan hasil pengendali PI dan PINN.

Pada gambar 5.5 diberikan *set point* untuk sudut rotasi sebesar 45 derajat. Dari gambar 5.5 dapat dilihat perbandingan antara sistem pengendali PI dan PINN. Respon dari sistem PI memiliki nilai o*vershoot* yang lebih besar dari sistem PINN.

Selain gambar 5.5 terdapat juga gambar 5.6 yang diberikan *set point* untuk sudut elevasi sebesar 60 derajat. Dari gambar 5.6 Dapat dilihat perbandingan antara sistem pengendali PI dan PINN. Pada sudut elevasi menunjukkan respon dari sistem PI memiliki nilai o*vershoot* yang lebih besar dari sistem PINN. Maka, dari hasil perbandingan grafik ini bisa dilihat bahwa sistem PINN lebih baik dari sistem PI dengan sudut rotasi 45 derajat dan sudut elevasi 60 derajat.



Gambar 5.5 grafik perbandingan respon sistem untuk sudut rotasi dengan *set point* 45 derajat



Gambar 5.6 Grafik perbandingan respon sistem untuk sudut elevasi dengan *set point* 60

Tabel 5.8 Performa sistem *turret-gun* dengan PINN *controller* untuk sudut rotasi

ROTASI				
Set Point	ST	OS	RT	SSE
15	0,45	0,00	0,42	4,33
30	0,50	0,00	0,45	5,38
45	0,44	0,00	0,14	1,50
60	0,46	6,40	0,15	1,32
75	0,53	12,71	0,15	1,32
90	0,28	12,16	0,16	1,92
105	0,28	0,83	0,16	0,87
Nilai tertinggi	0,53	12,71	0,14	5,38
Nilai terendah	0,28	0,00	0,45	0,87

Tabel 5.8 adalah hasil pengujian *prorotype turret gun* untuk mengetahui performa dari sistem pengendali PINN pada sudut rotasi. Pengujian ini dilakukan dengan pemberian *set point* yang telah ditentukan pada bab sebelumnya, maka didapatkan nilai *%overshoot* (%OS) tertinggi adalah 12,71% pada *set point* 75 derajat dan terendah adalah 0,00% pada *set point* 15, 30, dan 45 derajat. Sedangkan untuk nilai *rise time* (RT) tertinggi adalah 0,45 detik pada *set point* 30 derajat dan terendah 0,14 detik pada *set point* 45 derajat.

Kemudian untuk nilai *settling time* (ST) tertinggi adalah 0,53 detik pada *set point* 75 derajat dan terendah 0,28 detik pada *set point* 90 dan 105 derajat. Untuk nilai *steady-state error* (SSE) tertinggi adalah 5,38 derajat pada *set point* 30 derajat dan terendah 0,87 derajat pada *set point* 105 derajat. Hasil ini nantinya akan menjadi pembanding dengan hasil PI dengan PINN pada pembahasan sebelumnya.

Tabel 5.9 Performa sistem *turret-gun* dengan PINN *controller* untuk sudut elevasi

ELEVASI				
Set Point	ST	OS	RT	SSE
10	0,88	0,00	0,79	2,60
20	0,84	0,00	0,21	4,52
30	0,65	0,00	0,25	3,75
40	0,39	0,00	0,23	0,29
50	1,29	9,04	0,25	2,88
60	0,35	0,00	0,27	0,10
70	0,77	1,43	0,25	0,67
Nilai tertinggi	1,29	9,04	0,79	4,52
Nilai terendah	0,35	0,00	0,21	0,10

Tabel 5.9 adalah hasil pengujian *prorotype turret gun* untuk mengetahui performa dari sistem pengendali PINN pada sudut elevasi. Pengujian ini dilakukan dengan pemberian *set point* yang telah ditentukan pada bab sebelumnya, maka didapatkan nilai *%overshoot* (%OS) tertinggi adalah 9,04% pada *set point* 50 derajat dan terendah adalah 0,00% pada *set point* 10, 20, 30, 40, dan 60 derajat. Sedangkan untuk nilai *rise time* (RT) tertinggi adalah 0,79 detik pada *set point* 10 derajat dan terendah 0,21 detik pada *set point* 20 derajat.

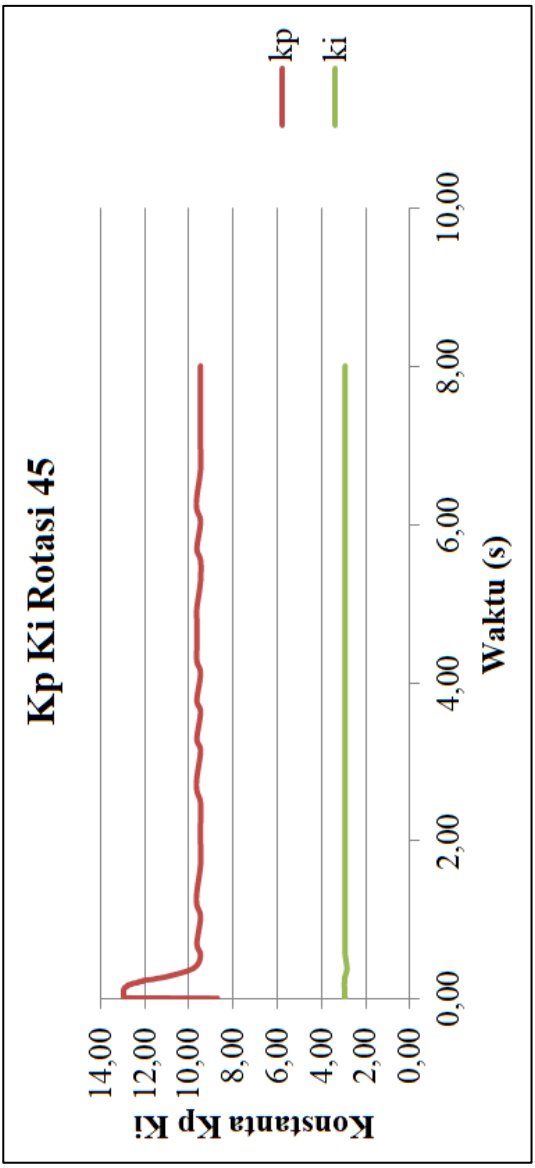
Kemudian untuk nilai *settling time* (ST) tertinggi adalah 1,29 detik pada *set point* 50 derajat dan terendah 0,35 detik pada *set point* 60 derajat. Untuk nilai *steady-state error* (SSE) tertinggi adalah 4,52 derajat pada *set point* 20 derajat dan terendah 0,10 derajat pada *set point* 60 derajat. Hasil ini nantinya akan menjadi pembanding dengan hasil PI dengan PINN pada pembahasan sebelumnya.

5.7 Parameter Gain Proportional-Integral (PI) pada Sistem Pengendali PINN

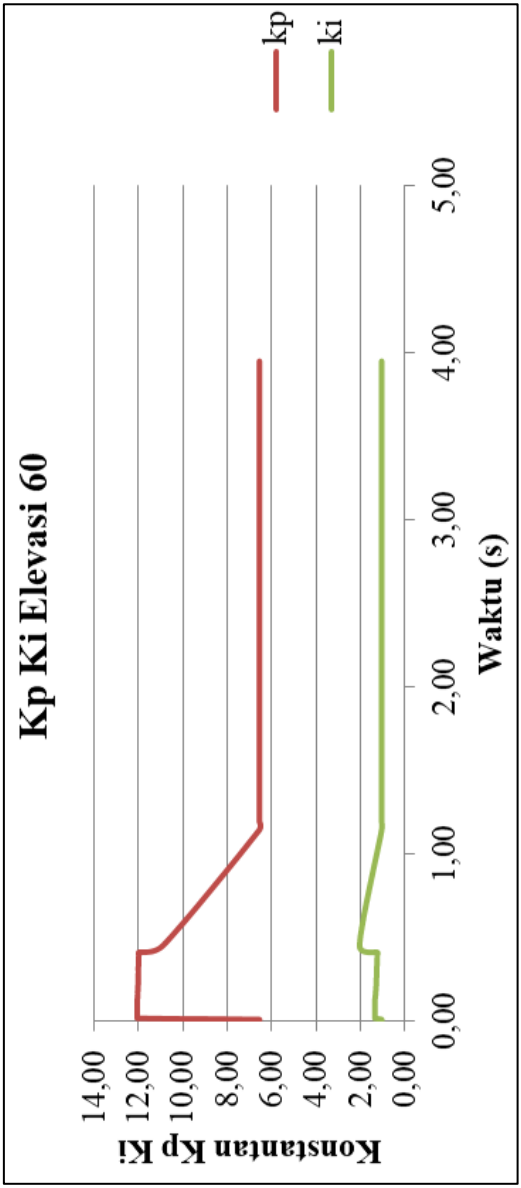
Parameter *proportional* gain (Kp) dan *integral* gain (Ki) pada *controller* yang diterapkan dengan metode *neural network* memiliki nilai kedua parameter yang dapat berubah. Perubahan ini terjadi disetiap waktu dengan mengikuti perubahan dari nilai error yang terjadi. Nilai Kp dan Ki dapat berubah menyesuaikan nilai error yang terjadi. Berikut adalah beberapa contoh perubahan nilai Kp dan Ki.

Pada gambar 5.7 terdapat grafik nilai Kp dan Ki sudut rotasi pada *set point* sebesar 45 derajat. Grafik menunjukkan bahwa perubahan nilai Kp dan Ki yang menyesuaikan perubahan nilai error pada pergerakan prototype. Pada grafik dapat dilihat untuk sudut rotasi meiliki nilai Kp berkisar di angka 8 sampai 13. Sedangkan perubahan nilai Ki untuk sudut rotasi berkisar di angka 2 sampai 3. Pada detik 0,5 sampai 1 Kp dan Ki mulai stabil yang menunjukkan nilai error juga sudah stabil.

Pada gambar 5.8 terdapat grafik nilai Kp dan Ki sudut elevasi pada *set point* sebesar 60 derajat. Grafik menunjukkan bahwa perubahan nilai Kp dan Ki menyesuaikan perubahan nilai error pada pergerakan prototype. Pada grafik dapat dilihat untuk sudut rotasi meiliki nilai Kp berkisar di angka 6 sampai 12. Sedangkan perubahan nilai Ki untuk sudut rotasi berkisar di angka 1 sampai 2. Pada detik 0,5 sampai 1 Kp dan Ki juga mulai stabil yang menunjukkan nilai error juga sudah stabil.



Gambar 5.7 Grafik Kp dan Ki pada sudut rotasi dengan *set point* 45 derajat



Gambar 5.8 Grafik Kp dan Ki pada sudut elevasi dengan *set point* 60 derajat

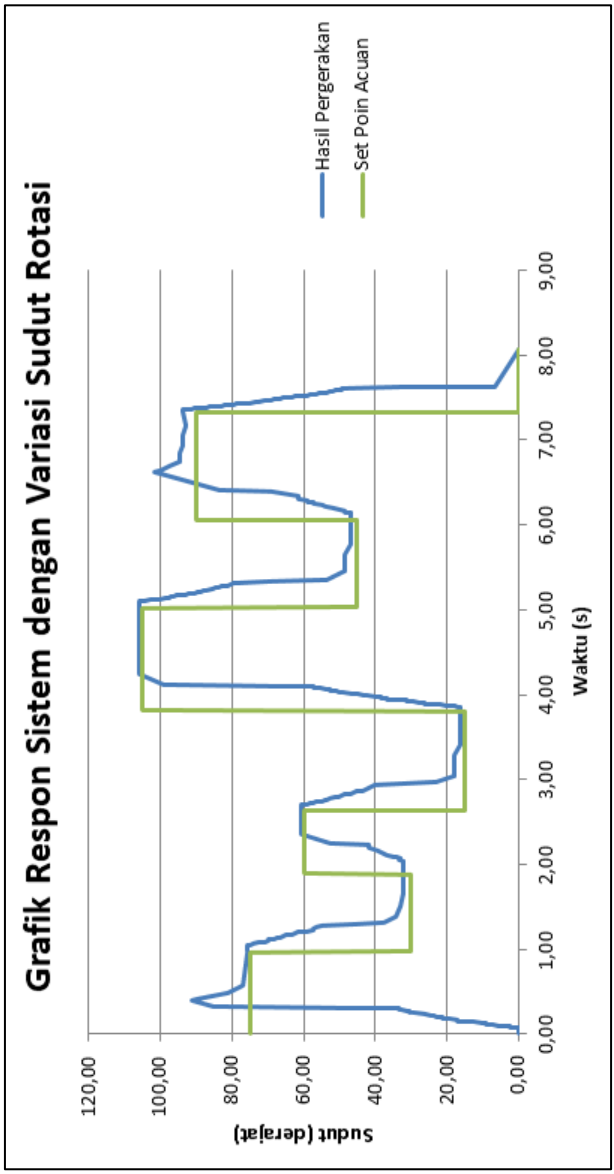
5.8 Simulasi pada Prototype dengan Variasi Set Point

Simulasi pada prototype dengan variasi sudut diperlukan untuk mengetahui apakah hasil dari neural networks dapat mengikuti perubahan sudut pada setiap waktu. Pengujian sistem turret gun dilakukan dengan variasi *set point* secara acak.

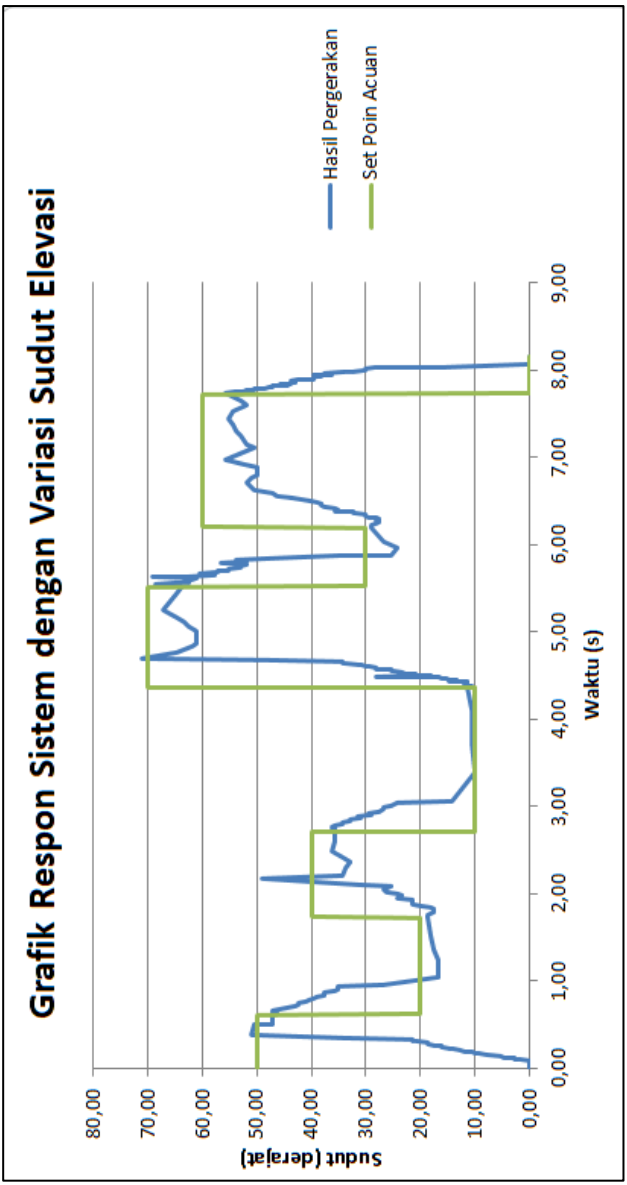
Gambar 5.9 adalah grafik respons dari sistem *turret-gun* dengan berbagai variasi *set point* sudut Rotasi. Sudut rotasi diawali dengan sudut 0 derajat. Kemudian dilanjutkan ke sudut 75 derajat, 30 derajat, 60 derajat, 15 derajat,105 derajat, 45 derajat, 90 derajat, dan terakhir di kembalikan ke sudut 0 derajat.

Sedangkan Gambar 5.10 menunjukkan grafik respon dengan variasi *set point* untuk sudut elevasi. Perubahan *set point* sudut elevasi juga di lakukan dengan diawali sudut 0 derajat, kemudian sudut 50 derajat, 20 derajat, 40 derajat, 10 derajat, 70 derajat, 30 derajat, 60 derajat, dan kembali lagi di berikan sudut 0 derajat.

Hasil dari pemberian *set point* ini memperlihatkan pembacaan potensiometer pada sudut rotasi lebih stabil dari pada pembacaan potensiometer oleh sudut elevasi, pada sudut elevasi beberapa kali terjadi kenaikan nilai secara mendadak di beberap titik. Hal ini bisa terjadi disebabkan potensiometer pada sudut elevasi lebih sensitif dari pada potensiometer pada sudut rotasi.



Gambar 5.9 Grafik respon sistem turret-gun dengan variasi set point sudut rotasi (0,75,30,60,15,105,45,90,0)



Gambar 5.10 Grafik respon sistem turret-gun dengan variasi *set point* sudut elevasi (0,50,20,40,10,70,30,60,0)

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Perancangan sistem kendali yang telah dibangun mampu mengendalikan *prototype turret gun*. Pada sudut rotasi didapatkan nilai *%overshoot* (%OS) tertinggi adalah 12,71% dan terendah adalah 0,00% Sedangkan untuk nilai *rise time* (RT) tertinggi adalah 0,45 detik dan terendah 0,14 detik. Untuk nilai *steady-state error* (SSE) tertinggi adalah 5,38 derajat dan terendah 0,87 derajat. Pada sudut elevasi didapatkan nilai *%overshoot* (%OS) tertinggi adalah 9,04% dan terendah adalah 0,00%. Sedangkan untuk nilai *rise time* (RT) tertinggi adalah 0,79 detik dan terendah 0,21 detik. Untuk nilai *steady-state error* (SSE) tertinggi adalah 4,52 derajat dan terendah 0,10 derajat.

6.2 Saran

Adapun saran yang dapat digunakan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Pembuatan mekanisme alat disesuaikan dengan kemampuan dari komponen alat yang digunakan, sehingga dihasil sistem kontrol yang lebih optimal pada *prototype turret gun*.
2. Pada penelitian selanjutnya, pengujian sistem pengendali pada alat dapat dilakukan dengan memberi input yang berubah-ubah mengikuti perubahan waktu, diharapkan dengan input yang berubah-ubah bisa memberikan hasil pengujian yang lebih baik dari penelitian saat ini.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] Noor, D. T., 2017. **Desain dan Analisa Penentuan Parameter Gain Sistem Pengendalian *Proportional-Integral* dengan Metode *Neural Network* pada Sistem *Turrent Gun*.** Surabaya: ITS.

[2] Lin, L., Peng, X., 2010. **A PID Neural Network Control For Permanent Magnet Synchronous Motor Servo System,** IEEE Xplore Digital Library 1,1:13-18.

[3] Gou-Jen, W., 2001. **Neural-Network-Based Self-Tuning PI Controller for Precise Motion Control of PMAC Motors.** IEEE Transaction on Industrial Electronics,vol. 48 No. 2, 1,1:408-415.

[4] Pitowarno, E., 2006. **Robotika Desain, Kontrol dan Kecerdasan Buatan**. Yogyakarta: Andi Yogyakarta

[5] Nise, N., 2011. **Control Sistems Engineering Sixth Edition**. California State Polytechnic University, Pomona.

[6] Brown, M., Harris, C., 1994. **Neurofuzzy Adaptive Modelling and Control.** Southampton University : Prentice Hall.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Lampiran 1. Program Arduino

```
#include <Servo.h>
#include <stdio.h>
#include <stdlib.h>
Servo servo_rot, servo_elev;

//variabel
float pos_rot, pos_elev,pos_rot1, pos_elev1;
int data;
String dataIn;
String dt[10];
int k;
boolean parsing=false;
float kp, ki,kp1,ki1;
float error, error_i, p, i, pi,error_last;
float error1, error_i1, p1, i1, pi1,error_last1;
float  as,rotasi=0,elevasi=0,prev_rotasi=0,prev_elevasi=0;
float simpan_error=0,simpan_error1=0, ti, ti1;
float sel1,sel2,tet1,tet2,tet11,tet22,min_s,min_s1,P,O;
int h=0;
int z=0;
float M11,M12,M13,M14,M15,M16,M17;
float M21,M22,M23,M24,M25,M26,M27;

//potensio
void read_position(){
  P=(analogRead(A4));
  O=(analogRead(A5));
  pos_rot=(333-analogRead(A4))*-1;  //(130) 462
  pos_elev=(792-analogRead(A5)); //(104) 687
  pos_rot1=(pos_rot/1.2185);
```

```
  if(pos_rot1<0){pos_rot1=0;}
  if(pos_rot1>106){pos_rot1=106;}
  pos_elev1=(pos_elev/1.4857);
  if(pos_elev1<0){pos_elev1=0;}
  if(pos_elev1>71){pos_elev1=71;}
}

void setup(){
  Serial.begin(9600);
  servo_rot.attach(2);
  servo_elev.attach(3);
}

void parsingData(){
  int j=0;
  dt[j]="";
  for(k=1;k<dataIn.length();k++){
    if ((dataIn[k] == '#') || (dataIn[k] == ',')){
      j++;
      dt[j]="";
    }
    else{
      dt[j] = dt[j] + dataIn[k];
    }
  }
  rotasi=dt[0].toInt();
  if(rotasi<0){rotasi=0;}
  else if(rotasi>105){rotasi=105;}
  elevasi=dt[1].toInt();
  if(elevasi<0){elevasi=0;}
  else if(elevasi>70){elevasi=70;}
}

void NN () {
    //Rotasi 13,8
```

```
    M11 = 1/(1+exp(-((simpan_error*0.2098)+(-13.9853))));
    M12 = 1/(1+exp(-((simpan_error*0.2459)+(-6.9377))));
    M13 = 1/(1+exp(-((simpan_error*1.1965)+(0.0097583))));
    M14 = 1/(1+exp(-((simpan_error*0.3)+(6.9004))));
    M15 = 1/(1+exp(-((simpan_error*-0,8741)+(15.0562))));

    M16 =
(M11*1.2106)+(M12*3.2238)+(M13*2.10943)+(M14*4.4342)+(
M15*2.09457)+1.1328;
    M17 = (M11*(-
1.0668))+(M12*0.039197)+(M13*0.021397)+(M14*(0.001137))
+(M15*(2.0445))+0.9163;

    //Elevasi 13,8
    M21 = 1/(1+exp(-((simpan_error1*0.7862)+(-13.0858))));
    M22 = 1/(1+exp(-((simpan_error1*0.5961)+(-5.246))));
    M23 = 1/(1+exp(-((simpan_error1*0.0405)+(-5.101))));
    M24 = 1/(1+exp(-((simpan_error1*0.3932)+(-6.2996))));
    M25 = 1/(1+exp(-((simpan_error1*0.145)+(14.0498))));

    M26 =
(M21*2.035687)+(M22*2.333)+(M23*1.70936)+(M24*3.0569)+
(M25*2.591377)+1.93093;
    M27 = (M21*(-
1.5594))+(M22*2.778)+(M23*2.9177)+(M24*(-
1.0020563))+(M25*(-1.00017516))+1.85618;
}

//Algoritma PID
void pid(){
//rotasi
  kp = M16;
  ki = M17;
  error = simpan_error;
  error_i = error-error_last ;
```

```
  p = kp * error;
  i = ki * error_i;
  pi=abs(p+i);
  error_last = error;

//elevasi
  kp1 = M26;
  ki1 = M27;
  error1 = simpan_error1;
  error_i1 = error1-error_last1 ;
  p1 = kp1 * error1;
  i1 = ki1 * error_i1;
  pi1=abs(p1+i1);
  error_last1 = error1;
}

void loop(){
  prev_rotasi=rotasi;
  prev_elevasi=elevasi;
  if(Serial.available()>0) {
    char inChar = (char)Serial.read();
    dataIn += inChar;
    if (inChar == '\n') {
      parsing = true;
    }
  }
  if(parsing){
    parsingData();
    parsing=false;
    dataIn="";
  }
  for(h=0;h<=z;h++){
  //sel1=rotasi- prev_rotasi;
  //sel2=elevasi- prev_elevasi;
  sel1=rotasi- pos_rot1;
```

```
sel2=elevasi-pos_elev1;
if(sel1<0){sel1=sel1*(-1);}
if(sel2<0){sel2=sel2*(-1);}
min_s=sel1;
min_s1=sel2;;
simpan_error=min_s;
simpan_error1=min_s1;
if(min_s==0){min_s=1;}
if(min_s1==0){min_s1=1;}
z=max(min_s,min_s1);
//
NN();
pid();

  //tet1=prev_rotasi+(((rotasi-prev_rotasi)/x)*h);
  //tet2=prev_elevasi+(((elevasi-prev_elevasi)/x)*h);
  tet1=prev_rotasi+(((rotasi- pos_rot1)/z)*h);
  tet2=prev_elevasi+(((elevasi-pos_elev1)/z)*h);
  tet11=58+tet1;
  tet22=tet2;
  ti=4000/(pi+1);
  if(ti>=1000){ti=5;}
  ti1=4000/(pi1+1);
  if(ti1>=1000){ti1=5;}
  delay(ti);
  servo_rot.write(tet11);
  delay(ti1);
  servo_elev.write(tet22);
  read_position();


  Serial.print(simpan_error);
  Serial.print(", ");
  Serial.print(ti);
  Serial.print(", ");
```

```
    Serial.print(kp);
    Serial.print(", ");
    Serial.print(ki);
    Serial.print("(");
    Serial.print(pos_rot1);
    Serial.print(", ");
    Serial.print(P);
    Serial.print(")| ");
    Serial.print(simpan_error1);
    Serial.print(", ");
    Serial.print(ti1);
    Serial.print(", ");
    Serial.print(kp1);
    Serial.print(", ");
    Serial.print(ki1);
    Serial.print("(");
    Serial.print(pos_elev1);
    Serial.print(", ");
    Serial.print(O);
    Serial.println(")");

  }
}
```

BIODATA PENULIS

Penulis dilahirkan di Jakarta, 17 Desember 1995, merupakan anak ketiga dari 11 bersaudara. Penulis telah menempuh pendidikan tingkat dasar di Sekolah Dasar Islam Terpadu IQRO. Sementara untuk pendidikan menengah, penulis menimba ilmu di SMPIT Al-Hassan dan SMA Negeri 5 Kota Bekasi. Kemudian setelah mengikuti Seleksi Bersama Mahasiswa Perguruan Tinggi Negeri, penulis diterima di Departemen Teknik Mesin FTI-ITS pada tahun 2013 dan terdaftar dengan NRP 02111340000154.

Di Jurusan Teknik Mesin ini, penulis mengambil Bidang Studi Manufaktur. Penulis sempat aktif di beberapa organisasi tingkat jurusan yaitu Himpunan Mahasiswa Mesin ITS sebagai kepala departemen sosial masyarakat pada tahun 2015/2016. Tingkat institut yaitu Badan Eksekutif Mahasiswa ITS sebagai menteri inkubator kajian pada tahun 2016/2018 dan aktif di komunitas pendidikan tingkat regional yaitu komunitas 1000 guru Surabaya pada tahun 2016/2018. Selain itu, selama kuliah penulis juga aktif sebagai Asisten dan Grader Praktikum Pengukuran Teknik di Laboratorium Perancangan dan Pengembangan Produk (Lab P3) pada tahun 2015/2018.

[Halaman ini sengaja dikosongkan]