



TUGAS AKHIR - TF 141581

**PERANCANGAN JARINGAN SYARAF TIRUAN
SEBAGAI ESTIMATOR LOAD DISTURBANCE
TORQUE PADA SISTEM SERVO MODULAR
MS150 DC**

Syahrul Munir
NRP. 02311440000040

Dosen Pembimbing :
Ir. Ya'umar, M.T.

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

“Halaman ini memang dikosongkan”



FINAL PROJECT - TF 141581

**DESIGNING ARTIFICIAL NEURAL NETWORK
AS LOAD DISTURBANCE TORQUE
ESTIMATOR ON MODULAR SERVO SYSTEM
MS150 DC**

Syahrul Munir
NRP. 0231144000040

Supervisors :
Ir. Ya'umar, M.T.

ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya
2018

“Halaman ini memang dikosongkan”

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini

Nama : Syahrul Munir
NRP : 02311440000040
Departemen/ Prodi : Teknik Fisika/ S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Perancangan Jaringan Syaraf Tiruan Sebagai Estimator *Load Disturbance Torque* Pada Sistem Servo Modular MS150 DC” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada tugas akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 23 Juli 2018
Yang membuat pernyataan,



Syahrul Munir
NRP. 02311440000040

“Halaman ini memang dikosongkan”

LEMBAR PENGESAHAN I

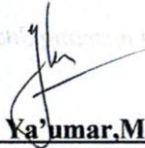
**PERANCANGAN JARINGAN SYARAF TIRUAN
SEBAGAI ESTIMATOR LOAD DISTURBANCE
TORQUE PADA SISTEM SERVO MODULAR
MS150 DC**

Oleh:

Svahrul Munir
NRP.0231144000040

Surabaya, 23 Juli 2018

**Menyetujui,
Dosen Pembimbing**



Ir. Ya'umar, M.T.
NIPN.19540406 198103 1 003

**Mengetahui,
Kepala Departemen
Teknik Fisika FTI-ITS**



Agus Muhamad Hatta, S.T., M.Si., Ph.D.
NIPN. 19780902 200312 1 002

“Halaman ini memang dikosongkan”

LEMBAR PENGESAHAN II

PERANCANGAN JARINGAN SYARAF TIRUAN SEBAGAI ESTIMATOR LOAD DISTURBANCE TORQUE PADA SISTEM SERVO MODULAR MS150 DC

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi
Progam Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember


Oleh:

Syahrul Munir

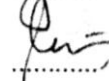
NRP. 0231144000040

Disetujui oleh Tim Penguji Tugas Akhir:


1. Ir. Ya'umar, M.T.


..... (Pembimbing)

2. Dr. Katherin Indriawati, S.T., M.T.


..... (Ketua Penguji)

3. Hendra Cordova, S.T., M.T.


..... (Penguji I)

SURABAYA

23 Juli 2018

“Halaman ini memang dikosongkan”

PERANCANGAN JARINGAN SYARAF TIRUAN SEBAGAI ESTIMATOR LOAD DISTURBANCE TORQUE PADA SISTEM SERVO MODULAR MS150 DC

Nama : Syahrul Munir
NRP : 0231144000040
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Ir.Ya'umar, M.T.

Abstrak

Penggunaan motor DC telah digunakan secara luas baik pada sistem yang membutuhkan sensitivitas kecil hingga sensitivitas yang tinggi. Pada sistem dengan sensitivitas yang tinggi butuh untuk menjaga kepresisian semaksimal mungkin bahkan akibat adanya *load disturbance torque* pada *shaft* motor yang terhubung pada rotor. Dikarenakan memiliki tingkat kesulitan yang tinggi untuk melakukan pengukuran secara langsung, estimasi digunakan untuk mengukur besarnya *load disturbance torque* pada *shaft* motor. Untuk mengatasi permasalahan ini, dilakukan penelitian mengenai perancangan estimasi *load disturbance torque* pada motor DC dengan menggunakan jaringan syaraf tiruan (JST). Dari penelitian ini, telah dirancang model Jaringan Syaraf Tiruan (JST) untuk estimasi *load disturbance torque* dengan jumlah *node* terbaik sebanyak 3 *node* masukan, antara lain kecepatan sudut, tegangan masukan, dan arus, 9 *hidden node*, dan 1 *node* keluaran, yakni *load disturbance torque*. Dari hasil pelatihan JST didapatkan nRMSE sebesar 3,75025% dan dari hasil pengujian didapatkan nRMSE sebesar 4,03775%. Sehingga dapat disimpulkan bahwa JST telah mampu mengestimasi *load disturbance torque* dengan nRMSE dibawah 7%.

Kata Kunci: motor DC, *load disturbance torque*, jaringan syaraf tiruan, Levenberg-Marquardt, RMSE

“Halaman ini memang dikosongkan”

DESIGNING ARTIFICIAL NEURAL NETWORK AS LOAD DISTURBANCE TORQUE ESTIMATOR ON MODULAR SERVO SYSTEM MS150 DC

Name : Syahrul Munir
NRP : 0231144000040
Department : Engineering Physics FTI-ITS
Supervisors : Ir. Ya'umar, M.T.

Abstract

The use of DC motors has been widely used both on systems that require low sensitivity up to high sensitivity. In systems with high sensitivity, it is necessary to maintain the maximum possible precision even due to the load disturbance torque on the motor shaft connected to the rotor. Due to having a high difficulty for direct measurement, estimates are used to measure the amount of load disturbance torque on the motor shaft. To overcome this problem, a study conducted for load disturbance torque estimation on DC motor using artificial neural network (ANN). From this research, Artificial Neural Network (ANN) model has been designed to estimate the load disturbance torque with the best node quantity of 3 input nodes, including angular velocity, input voltage, and current, 9 hidden nodes and 1 output node, ie load disturbance torque. From the results of training ANN obtained nRMSE of 3.75025% and from the test results obtained nRMSE of 4.03775%. So it can be concluded that ANN has been able to estimate the load disturbance torque with nRMSE below 7%.

Keywords: DC motor, load disturbance torque, artificial neural network, Levenberg-Marquardt, RMSE

“Halaman ini memang dikosongkan”

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat serta hidayah-Nya, serta shalawat serta salam kepada Nabi Muhammad SAW, hingga terselesaikannya tugas akhir beserta laporan tugas akhir yang berjudul

PERANCANGAN JARINGAN SYARAF TIRUAN SEBAGAI ESTIMATOR LOAD DISTURBANCE TORQUE PADA SISTEM SERVO MODULAR MS150DC

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian tugas akhir dan laporan Tugas Akhir ini. Penulis mengucapkan terimakasih kepada :

1. Bapak Agus Muhamad Hatta, S.T., M.Si., Ph.D selaku Kepala Departemen Teknik Fisika yang telah memberikan petunjuk, ilmu, serta bimbingan selama menempuh pendidikan di Teknik Fisika.
2. Bapak Ir. Ya'umar, M.T. selaku dosen pembimbing yang telah dengan sabar memberikan petunjuk, ilmu, serta bimbingan yang sangat bermanfaat.
3. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku Kepala Laboratorium Fisika Rekayasa tempat saya mengerjakan tugas akhir yang telah memberikan ilmu, petunjuk, nasihat, serta kemudahan.
4. Bapak Dr. Ir. Totok Soehartanto, DEA. selaku dosen wali yang telah membimbing penulis selama perkuliahan.
5. Orang tua serta seluruh keluarga. Terimakasih atas segala cinta, kasih sayang, doa, perhatian, serta dukungan moril dan materil yang telah diberikan.
6. Seluruh keluarga asisten Laboratorium Simulasi dan Komputasi, terima kasih untuk semuanya.
7. Seluruh teman Tugas Akhir Laboratorium Fisika Rekayasa dan Tugas Akhir Jaringan Syaraf Tiruan , terima kasih untuk semuanya.

8. Seluruh teman-teman Departemen Teknik Fisika angkatan 2014, *Tree Fortress*, terima kasih untuk semuanya.
9. Seluruh dosen, karyawan dan civitas akademik Teknik Fisika, terimakasih atas segala bantuan dan kerjasamanya.
10. Semua pihak yang tidak dapat disebutkan satu persatu, terimakasih atas bantuannya.

Penulis sadar bahwa penulisan laporan tugas akhir ini tidak sempurna, namun semoga laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika khususnya, dan civitas akademik ITS pada umumnya. Semoga laporan tugas akhir ini dapat bermanfaat sebagai referensi pengerjaan laporan tugas akhir bagi mahasiswa yang lain.

Surabaya, 23 Juli 2018

Penulis

DAFTAR ISI

Halaman Judul.....	i
<i>Title Page</i>	iii
PERNYATAAN BEBAS PLAGIARISME.....	v
LEMBAR PENGESAHAN I	vii
LEMBAR PENGESAHAN II	ix
Abstrak.....	xi
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Laporan.....	3
BAB II TEORI PENUNJANG.....	5
2.1 Motor DC.....	5
2.2 Sistem Servo Modular MS150 DC.....	7
2.3 Model Matematik Motor DC.....	8
2.4 Jaringan Syaraf Tiruan	10
2.4.1 Arsitektur Jaringan Syaraf Tiruan	11
2.4.1.1 <i>Single-Layer</i>	11
2.4.1.2 <i>Multi-Layer</i>	13
2.4.2 Fungsi Aktivasi	15
2.4.2.1 Sigmoid Logaritmik	15
2.4.2.2 Sigmoid Tangen	15
2.4.2.3 Linear.....	16
2.4.3 Algoritma Pelatihan Jaringan Syaraf Tiruan	17
2.4.3.1 Algoritma <i>Steepest Descent</i>	17
2.4.3.2 Metode Newton.....	17
2.4.3.3 Metode Gauss-Newton.....	18
2.4.3.4 Algoritma <i>Levenberg-Marquardt</i>	19
2.5 <i>Root Mean Square Error</i> (RMSE)	20

BAB III METODOLOGI PENELITIAN	23
3.1 Perumusan Masalah	23
3.2 Studi Literatur	24
3.3 <i>Set-Up</i> Sistem Servo Modular MS150 DC	24
3.4 Pengambilan Data	25
3.5 Uji Korelasi	27
3.6 Perancangan Model JST dengan Variasi <i>Hidden Node</i>	28
3.7 Pengujian Model JST	31
3.8 Analisa Data	31
3.9 Kesimpulan dan Saran.....	31
BAB IV HASIL DAN PEMBAHASAN	33
4.1 Uji Korelasi	33
4.2 Perancangan Estimator (JST)	34
4.2.1 Pelatihan JST.....	35
4.2.2 Pengujian JST	41
BAB V PENUTUP	43
5.1 Kesimpulan	43
5.2 Saran	43
DAFTAR PUSTAKA.....	45
LAMPIRAN A.....	49
LAMPIRAN B1	51
LAMPIRAN B2	63
LAMPIRAN C	67
LAMPIRAN D	69
LAMPIRAN E	71
LAMPIRAN F.....	73
LAMPIRAN G.....	83
BIODATA PENULIS.....	85

DAFTAR GAMBAR

Gambar 2.1 Motor DC	6
Gambar 2.2 Sistem Servo Modular MS150	7
Gambar 2.3 Rangkaian motor DC.....	9
Gambar 2.4 Arsitektur <i>Single-Layer</i>	12
Gambar 2.5 Arsitektur <i>Multi-Layer</i>	13
Gambar 2.6 Fungsi Sigmoid Logaritmik.....	15
Gambar 2.7 Fungsi Sigmoid Tangen	16
Gambar 2.8 Fungsi Linear	16
Gambar 2.9 Blok diagram pelatihan menggunakan algoritma LM untuk 5 output	19
Gambar 3.1 Diagram Alir Penelitian	23
Gambar 3.2 <i>Set-Up</i> Sistem Servo Modular MS150DC	24
Gambar 3.3 Tampilan GUI LabVIEW Pengambilan Data	26
Gambar 3.4 Hubungan <i>brake scale</i> terhadap <i>load torque</i>	26
Gambar 4.1 Arsitektur JST untuk 9 <i>hidden node</i>	36
Gambar 4.2 Plot Regresi untuk 9 <i>hidden node</i>	39
Gambar 4.3 Performansi Pelatihan untuk 9 <i>hidden node</i>	40
Gambar 4.4 Hasil Pelatihan JST Menggunakan 9 <i>hidden node</i> . 41	
Gambar 4.5 Hasil Pengujian JST	42

“Halaman ini memang dikosongkan”

DAFTAR TABEL

Tabel 3.1 Pedoman Menentukan Tingkat Hubungan Berdasarkan Koefisien Korelasi	28
Tabel 3.2 Parameter Model JST.....	30
Tabel 4.1 Hasil Uji Korelasi Nilai r	33
Tabel 4.2 Nilai RMSE dan nRMSE Untuk Variasi <i>Hidden Node</i>	35
Tabel 4.3 Nilai bobot JST dari <i>input layer</i> ke <i>hidden layer</i>	37
Tabel 4.4 Nilai bobot JST dari <i>hidden layer</i> ke <i>output layer</i>	38

“Halaman ini memang dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejak abad 19 konsep mengenai motor listrik terus berkembang. Penggunaan motor listrik telah digunakan secara luas, antara lain banyak digunakan pada peralatan instrumentasi dan otomasi industri, otomotif, kedirgantaraan, dunia medis dan lain sebagainya[1][2]. Salah satu jenis motor listrik adalah motor DC (*Direct Current*). Motor DC adalah perangkat elektromagnetis yang berfungsi mengubah energi listrik menjadi energi mekanik. Motor DC tersusun dari dua bagian, yaitu rotor (bagian yang berputar) dan stator (bagian yang tidak berputar). Pada rotor terdapat beberapa bagian seperti kumparan jangkar atau angker dinamo (*armature*), komutator (*commutator*), sikat (*brush*), dan *shaft*.

Motor DC digunakan baik pada sistem yang membutuhkan sensitivitas kecil seperti jendela otomatis dan pintu otomatis, hingga pada sistem dengan sensitivitas yang tinggi seperti pada kontrol *steering* dan *throttle*. Pada sistem dengan sensitivitas yang tinggi butuh untuk menjaga kepresisian semaksimal mungkin bahkan akibat adanya *load disturbance torque* pada *shaft* motor yang terhubung pada rotor. Oleh karena itu, diperlukan pengetahuan mengenai adanya gangguan torsi ini untuk menjaga kepresisian motor[3].

Dikarenakan memiliki tingkat kesulitan yang tinggi untuk melakukan pengukuran secara langsung, estimasi digunakan untuk mengukur besarnya *load disturbance torque* pada *shaft* motor. Nilai estimasi ini nantinya dapat bermanfaat untuk meningkatkan *robust* dari sistem terhadap gangguan luar. Beberapa penelitian sebelumnya melakukan penelitian untuk mengestimasi besarnya *load disturbance torque*. Salah satu penelitian sebelumnya mengestimasi besarnya *load disturbance torque* pada motor DC dengan menggunakan *observer*[3]. Penelitian lainnya menggunakan model yang telah dioptimasi dan *fuzzy-PID*[4]. Terdapat pula penelitian lain dengan mengestimasi

torsi menggunakan *observer* dan mendapatkan nilai *normalized Root Mean Square Error* (nRMSE) sebesar 7% dan 8.32%[5]. Oleh karena itu, pada penelitian ini dilakukan penelitian mengenai estimasi *load disturbance torque* pada motor DC dengan menggunakan metode baru, yaitu jaringan syaraf tiruan, untuk mendapatkan nilai nRMSE dibawah 7%.

1.2 Rumusan Masalah

Dari latar belakang diatas, maka rumusan masalah pada penelitian kali ini adalah:

- a. Bagaimana rancangan model Jaringan Syaraf Tiruan (JST) untuk estimasi *load disturbance torque* pada motor DC?
- b. Apakah model Jaringan Syaraf Tiruan (JST) yang telah dirancang mampu menghasilkan nilai nRMSE estimasi dibawah 7%?

1.3 Tujuan

Dari perumusan masalah yang telah diuraikan diatas, maka didapatkan tujuan dalam tugas akhir ini yaitu mampu merancang model Jaringan Syaraf Tiruan (JST) untuk estimasi *load disturbance torque* pada motor DC yang dapat menghasilkan nilai nRMSE estimasi dibawah 7%.

1.4 Batasan Masalah

Adapun batasan masalah pada tugas akhir ini adalah:

- a. Motor DC yang digunakan adalah motor DC pada sistem servo modular MS150 DC di Laboratorium Fisika Rekayasa.
- b. *Load disturbance torque* diberikan dengan menggunakan bantuan magnet permanen yang didekatkan pada *shaft* motor.
- c. Variabel yang diukur adalah arus, tegangan masuk, dan kecepatan sudut.
- d. Kecepatan sudut motor pada saat pengukuran adalah 1000 rpm dengan toleransi $\pm 5\%$.

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

a. **BAB I PENDAHULUAN**

Pada bab I ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika laporan.

b. **BAB II TEORI PENUNJANG**

Pada bab II ini dibahas mengenai teori-teori yang berkaitan dengan penelitian yang akan dilakukan.

c. **BAB III METODOLOGI PENELITIAN**

Pada bab III ini berisi mengenai rancangan dari penelitian yang dilakukan, metode dan langkah-langkah dalam melakukan penelitian.

d. **BAB IV HASIL DAN PEMBAHASAN**

Pada bab IV ini berisi tentang analisis hasil perancangan model jaringan syaraf tiruan yang telah dibuat.

e. **BAB V PENUTUP**

Pada bab V ini diberikan kesimpulan tentang tugas akhir yang telah dilakukan berdasarkan analisis data yang diperoleh, serta diberikan saran sebagai penunjang maupun pengembangan tugas akhir selanjutnya.

“Halaman ini memang dikosongkan”

BAB II

TEORI PENUNJANG

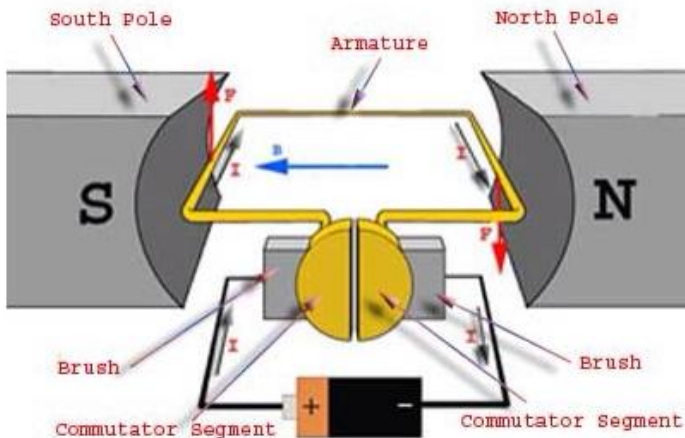
2.1 Motor DC

Motor DC merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan jangkar disebut rotor (bagian yang berputar) dan kumparan medan pada motor DC disebut stator (bagian yang tidak berputar). Saat terjadi putaran pada kumparan jangkar, maka akan timbul tegangan yang berubah-ubah arah pada setiap setengah putaran dan menghasilkan tegangan bolak-balik. Arus searah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator sehingga arus yang berbalik arah dengan kumparan jangkar berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen[6].

Salah satu jenis motor DC yang sering ditemukan yaitu motor DC jenis magnet permanen, dimana medan magnet dihasilkan oleh magnet permanen dan menghasilkan fluks konstan. Motor DC magnet permanen tersusun dari beberapa bagian seperti magnet permanen, kumparan jangkar atau angker dinamo (*armature*), komutator (*commutator*), dan sikat (*brush*) yang ditunjukkan pada Gambar 2.1.

Terdapat dua prinsip dasar yang mendasari kerja motor DC magnet permanen. Pertama, magnet permanen sebagai stator motor tersusun dari dua magnet berlawanan yaitu kutub utara dan kutub selatan yang saling berhadapan dan berfungsi menghasilkan fluks yang nilainya konstan dari kutub utara ke kutub selatan. Komutator yang termasuk dalam bagian dari rotor motor terletak pada kedua ujung kumparan jangkar yang berfungsi mengumpulkan arus induksi dari jangkar dan mengkonversinya menjadi arus searah. Sikat berfungsi menyalurkan arus listrik dari sumber diluar motor ke dalam kumparan jangkar dan kumparan jangkar (*armature*) sebagai rotor motor digambarkan dalam

bentuk sebuah kawat yang memiliki bentuk persegi panjang dan berfungsi untuk merubah energi listrik menjadi energi mekanik dalam bentuk gerak putar. Kedua, sumber tegangan DC pada Gambar 2.1 diilustrasikan dengan baterai yang memiliki kutub positif dan kutub negatif, kemudian baterai dari kutub positif akan mengalirkan arus menuju sikat (*brush*) yang selanjutnya menuju komutator, kemudian arus menuju kumparan jangkar (*armature*), dan kembali lagi ke komutator, lalu sikat karbon dan terakhir menuju ke kutub negatif baterai. Dibagian stator motor, kumparan medan stator menghasilkan fluks dari kutub utara ke kutub selatan. Dengan adanya arus di dalam kumparan jangkar yang terletak didalam medan magnet yang mengandung fluks, maka akan menghasilkan suatu gaya, yang biasa disebut gaya Lorentz (F), selanjutnya gaya tersebut akan menggerakkan kumparan jangkar untuk berputar. Komponen komutator yang selalu ikut berputar dengan rotor dan sikat karbon yang selalu diam, menjadi komponen yang akan menjaga arah arus listrik untuk selalu tetap yakni mengalir dari sisi kiri kawat angker ke kanan. Arah arus listrik yang selalu tetap di setiap setengah putaran rotor inilah yang akan membuat rotor motor listrik selalu berputar selama masih ada arus listrik yang mengalir ke kumparan jangkar[7].



Gambar 2.1 Motor DC[8]

2.2 Sistem Servo Modular MS150 DC

Sistem servo modular MS150 DC merupakan blok rangkaian elektronik yang digunakan untuk pengendalian kecepatan dan posisi dari motor DC.



Gambar 2.2 Sistem Servo Modular MS150[9]

Sistem servo modular MS150 DC secara umum terdiri dari beberapa bagian yaitu :

a. *Power supply*

Sumber tegangan yang dipakai untuk menyuplai rangkaian pengaturan motor DC adalah modul Feedback tipe PS150E. Alat ini menghasilkan tegangan keluaran sebesar 24V DC 2A ke *servo amplifier*. Pada panel depan terdapat dua set *socket* untuk pasokan tegangan referensi.

b. *Motor DC*

Motor DC yang digunakan merupakan motor DC tipe DCM150F yang terdiri dari tiga bagian yaitu motor DC magnet permanen dengan poros yang diperpanjang, *tachogenerator* dengan output diatas unit, dan poros berkecepatan rendah.

c. *Servo amplifier*

Modular *feedback servo amplifier* SA150D terdiri dari rangkaian transistor yang dapat menggerakkan motor DC dengan dua arah putaran. Untuk mencegah motor dari *overloading*, terdapat pengukur arus agar arus yang mengalir tidak lebih dari 2A.

d. *Attenuator*

Attenuator AU150B berfungsi sebagai pemberi hambatan yang terdiri dari dua potensiometer dengan kapasitas masing-masing sebesar 10 K Ω yang dapat digunakan secara bersamaan.

e. *Load Unit*

Load Unit LU150L merupakan magnet permanen yang dapat diatur sudutnya. Terdapat juga piringan alumunium yang dapat dipasang pada *shaft* motor dimana ketika piringan tersebut berputar diantara magnet, maka akan menciptakan efek pengereman.

f. *Tachogenerator*

Tachogenerator GT150X berfungsi untuk mengkonversi besaran mekanik menjadi besaran listrik sehingga putaran motor dapat diukur.

g. *PID Unit*

PID Unit PID150Y merupakan unit kontrol yang dapat diatur sebagai kontroller P, PI, maupun PID dengan mengatur lewat saklar. Terdapat potensiometer yang digunakan untuk mengubah konstanta K_p , T_i , dan T_d dan juga saklar sebagai pengali konstanta.

h. *Op-Amp*

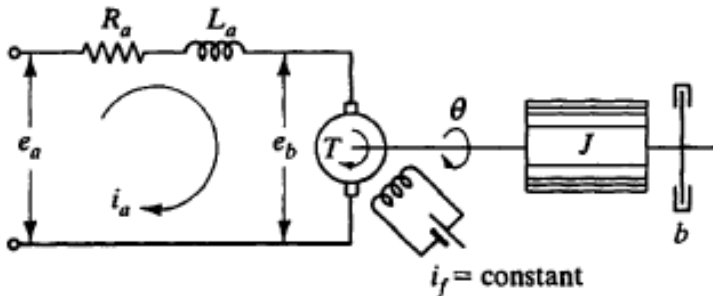
Operational Amplifier OA150A berfungsi sebagai penguat pada pengendalian umpan balik dan dapat menjumlahkan 2 atau 3 sinyal.

2.3 Model Matematik Motor DC

Sistem merupakan suatu rangkaian yang disusun dari bermacam-macam komponen yang saling berhubungan dalam menjalankan suatu aksi. Berdasarkan komponen yang digunakan, suatu sistem dapat berupa sistem mekanik, pneumatik, elektrik atau jenis elektro-mekanik. Motor DC termasuk dalam sistem elektro-mekanik.

Suatu prasyarat dasar pada kebanyakan strategi yang dipergunakan dalam membangun suatu sistem kendali adalah kemampuan dalam memodelkan secara matematika dari sistem yaitu sistem yang akan dikendalikan. Model matematika dari

suatu sistem dapat berupa persamaan differensial, fungsi transfer atau ruang *keadaan* (*state space*). Prinsipnya terdapat dua macam pemodelan matematika yang dapat dipilih, yaitu berdasarkan pada teori pengetahuan dengan menggunakan hukum fisika atau dengan proses eksperimen yaitu melakukan pengukuran. Kebanyakan kasus tidak mungkin untuk membuat model yang sempurna hanya dengan menggunakan pengetahuan fisika saja. Beberapa parameter harus ditentukan dari suatu eksperimen. Pendekatan ini disebut dengan identifikasi sistem. Model matematik untuk *plant* motor DC dapat dijabarkan pada bagian elektrik dan mekanik.



Gambar 2.3 Rangkaian motor DC [10]

Berdasarkan gambar 2.3, tegangan catu (V) untuk sistem motor DC adalah masukan yang dipasang pada armatur (R , L , θ motor). Tegangan tersebut oleh motor diubah menjadi gerak putar yang merupakan keluaran sistem sebagai kecepatan rotasi dari rotor motor. Kecepatan rotasi poros motor dinyatakan sebagai $d\theta/dt$, dengan asumsi rotor dan poros *rigid*. Secara umum, torsi yang dibangkitkan oleh motor DC adalah proporsional terhadap arus motor (armatur) dan kuat medan magnet, dengan asumsi bahwa medan magnet konstan maka torsi motor proporsional terhadap arus motor i dengan faktor konstanta K_t seperti dituliskan pada persamaan berikut [11].

$$T = K_t \cdot i \quad (2.1)$$

Bila kumpulan magnet berputar maka tegangan akan sebanding dengan hasil kali fluks dan kecepatan sudut yang

diinduksikan pada kumparan magnet. Untuk fluks yang konstan, tegangan induksi e_b berbanding lurus dengan kecepatan sudut $d\theta/dt$, atau[11]

$$e_b = K_b \cdot \dot{\theta} \quad (2.2)$$

dimana e_b adalah emf balik dan K_b adalah konstanta emf balik. Dalam unit SI, konstanta torsi motor dan gaya gerak listrik emf balik adalah sama, sehingga $K_t = K_b$; dengan demikian dapat digunakan konstanta K untuk menyatakan keduanya.

Arus jangkar magnet menghasilkan torsi yang bekerja terhadap inersia dan gesekan[11], sehingga

$$J \frac{d\omega}{dt} + b\omega = T = K \cdot i \quad (2.3)$$

Kecepatan jangkar magnet servomotor DC dikendalikan oleh tegangan kumparan magnet e_a . Persamaan diferensial rangkaian kumparan magnet adalah[11],

$$L \frac{di}{dt} + Ri + e_b = e_a \quad (2.4)$$

$$L \frac{di}{dt} + Ri = V - K \cdot \dot{\theta} \quad (2.5)$$

Keterangan:

- e_a = tegangan yang dikenakan pada jangkar, volt
- e_b = ggl - gaya gerak listrik balik, volt
- I = arus kumparan jangkar, ampere
- R = Tahanan kumparan jangkar, ohm
- L = induktansi kumparan jangkar, henry
- V = tegangan masukan servo, volt

2.4 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* (ANN) merupakan suatu sistem pemrosesan informasi yang memiliki karakteristik performansi yang mirip dengan jaringan syaraf biologis. JST telah dikembangkan sebagai generalisasi dari

model matematik untuk kesadaran manusia atau syaraf biologis, dengan asumsi dasar sebagai berikut:

- a. Informasi yang diproses muncul sebagai beberapa elemen sederhana yang disebut neuron atau node.
- b. Sinyal dilewatkan diantara neuron menggunakan link koneksi atau koneksi yang terhubung.
- c. Setiap link koneksi membunyai bobot tertentu, dimana pada jaringan syaraf secara umum, bobot tersebut akan melipatgandakan sinyal yang ditransmisikan antara neuron yang satu dengan yang lainnya.
- d. Setiap neuron mengaplikasikan fungsi aktivasi (biasanya non linear) menuju masukan jaringan (gabungan bobot dari sinyal masukan) untuk menentukan sinyal keluaran dari neuron[12].

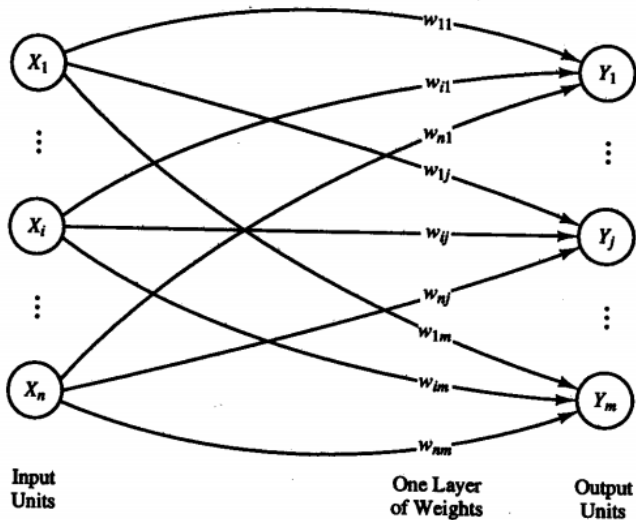
JST dikarakterisasi oleh berbagai hal yaitu, pola dari koneksi antara neuron (*node*) atau disebut juga dengan arsitektur jaringan, metode untuk menentukan bobot pada koneksi yang disebut juga algoritma pelatihan atau pembelajaran, dan fungsi aktivasi. Hal-hal tersebut akan memiliki pola atau nilai yang berbeda pada tiap kasus dalam menggunakan jaringan syaraf tiruan.

2.4.1 Arsitektur Jaringan Syaraf Tiruan

Arsitektur JST menggambarkan bagaimana susunan dari neuron (*node*) dan pola dari hubungan antara neuron tersebut. Beberapa arsitektur yang sering digunakan yaitu *single-layer* dan *multi-layer*.

2.4.1.1 Single-Layer

Arsitektur *single-layer* hanya mempunyai satu lapisan bobot saja. Sehingga, setiap masukan akan dihubungkan langsung dengan keluaran.



Gambar 2.4 Arsitektur *Single-Layer*[12]

Dari gambar 2.4 diatas, dapat dilihat bahwa seluruh unit masukan terhubung ke setiap unit keluaran dengan bobot yang berbeda-beda. Tiap unit masukan tidak terhubung ke unit masukan lain dan unit keluaran juga tidak terhubung ke unit keluaran yang lain.

Node masukan (*input node*) disimbolkan sebagai X_i node keluaran (*hidden node*) disimbolkan sebagai Y_j , dimana nilai i dan j sesuai dengan jumlah node yang digunakan. Persamaan untuk memperoleh besar sinyal masukan ke node keluaran adalah sebagai berikut,

$$y_{in,j} = \sum_{i=1}^n x_i w_{ij} \quad (2.6)$$

dengan:

$y_{in,j}$ =sinyal masukan ke node tersembunyi ($j=1,2,3,\dots,n$)

x_i =sinyal keluaran dari node masukan ($i=1,2,3,\dots,m$)

w_{ij} =nilai bobot dari node masukan menuju node tersembunyi.

Kemudian, sinyal keluaran dari node keluaran dapat diperoleh dengan mengaplikasikan sinyal masukannya pada

fungsi aktivasi. Persamaan berikut dapat digunakan untuk mencari sinyal keluaran node tersembunyi,

$$y_j = f(y_{in,j}) \quad (2.7)$$

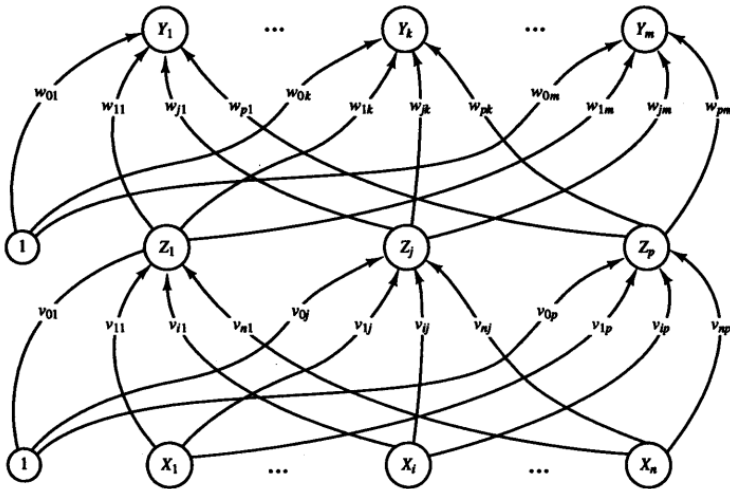
dengan:

y_j = sinyal keluaran dari node tersembunyi ($j=1,2,3,\dots,n$)

2.4.1.2 Multi-Layer

Jaringan *multi-layer* merupakan perluasan dari *single-layer*. Dalam jaringan ini, selain unit input dan output, ada unit-unit lain (sering disebut lapisan tersembunyi). Dimungkinkan pula ada beberapa lapisan tersembunyi. Sama seperti pada unit masukan dan keluaran, unit-unit dalam satu lapisan tidak saling berhubungan.

Sebagai contoh, gambar 2.5 merupakan contoh Jaringan Syaraf Tiruan dengan menggunakan pola atau arsitektur *Multilayer Network* dengan satu lapisan masukan (*input layer*), satu lapisan tersembunyi (*hidden layer*), dan satu lapisan keluaran (*output layer*).



Gambar 2.5 Arsitektur *Multi-Layer*[12]

Node masukan (*input node*) disimbolkan sebagai X_i , node tersembunyi (*hidden node*) disimbolkan sebagai Y_j , dan node keluaran (*output node*) disimbolkan sebagai Z_k , dimana nilai i, j , dan k sesuai dengan jumlah node yang digunakan. Persamaan untuk memperoleh besar sinyal masukan ke node tersembunyi adalah sebagai berikut,

$$y_{in.j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.8)$$

dengan:

$y_{in.j}$ = sinyal masukan ke node tersembunyi ($j=1,2,3,\dots,n$)

v_{0j} = nilai bobot dari bias menuju node tersembunyi

x_i = sinyal keluaran dari node masukan ($i=1,2,3,\dots,m$)

v_{ij} = nilai bobot dari node masukan menuju node tersembunyi.

Persamaan untuk memperoleh besar sinyal masukan ke node keluaran adalah sebagai berikut,

$$z_{in.k} = w_{0k} + \sum_{j=1}^n y_j w_{jk} \quad (2.9)$$

dengan:

$z_{in.k}$ =sinyal masukan ke node keluaran ($k=1,2,3,\dots,p$)

w_{0k} =nilai bobot dari bias menuju node keluaran

y_j =sinyal keluaran dari node tersembunyi ($j=1,2,3,\dots,n$)

w_{jk} =nilai bobot dari node tersembunyi menuju node keluaran.

Kemudian,sinyal keluaran dari node tersembunyi dan node keluaran dapat diperoleh dengan mengaplikasikan sinyal masukan pada fungsi aktivasi. Salah satu fungsi aktivasi yang dapat digunakan yaitu bipolar sigmoid atau disebut juga tangen hiperbolik. Persamaan berikut dapat digunakan untuk mencari sinyal keluaran node tersembunyi,

$$y_j = \tanh\left(\frac{y_{in.j}}{2}\right) \quad (2.10)$$

dengan:

y_j = sinyal keluaran dari node tersembunyi ($j=1,2,3,\dots,n$).

Persamaan mencari sinyal keluaran dari node keluaran adalah sebagai berikut,

$$z_k = \tanh\left(\frac{z_{in.k}}{2}\right) \quad (2.11)$$

dengan:

z_k = sinyal keluaran dari node keluaran ($k=1,2,3,\dots,p$).

2.4.2 Fungsi Aktivasi

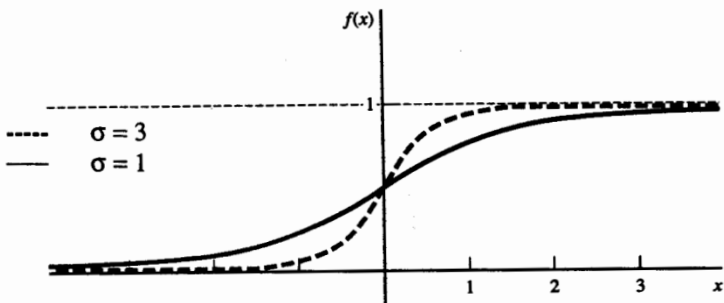
Fungsi aktivasi digunakan untuk menghasilkan nilai keluaran dari setiap node yang dilewati. Terdapat beberapa jenis fungsi aktivasi yang paling sering digunakan pada jaringan syaraf tiruan yaitu sigmoid logaritmik (*logsig*), sigmoid tangen (*tansig*), dan linear (*purelin*). Fungsi aktivasi sigmoid logaritmik dan sigmoid tangen dapat meminimalisir kapasitas komputasi untuk pelatihan, sedangkan fungsi aktivasi linear lebih banyak digunakan pada *output node*[13].

2.4.2.1 Sigmoid Logaritmik

Fungsi sigmoid logaritmik merupakan fungsi aktivasi dengan range keluaran dari 0 hingga 1. Fungsi ini biasa disebut sebagai *binary sigmoid*. Persamaan fungsi ini dan turunannya adalah sebagai berikut

$$f(x) = \frac{1}{1+e^{-\sigma x}} \quad (2.12)$$

$$f'(x) = \sigma f(x)[1 - f(x)] \quad (2.13)$$



Gambar 2.6 Fungsi Sigmoid Logaritmik

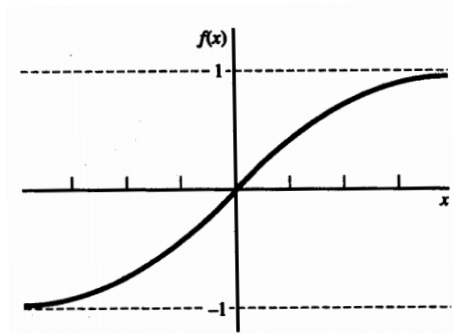
2.4.2.2 Sigmoid Tangen

Fungsi sigmoid tangen merupakan fungsi aktivasi dengan *range* keluaran dari -1 hingga 1. Fungsi ini biasa disebut sebagai

bipolar sigmoid. Persamaan fungsi ini dan turunannya adalah sebagai berikut

$$f(x) = \frac{2}{1+e^{-\sigma x}} - 1 \quad (2.14)$$

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \quad (2.15)$$



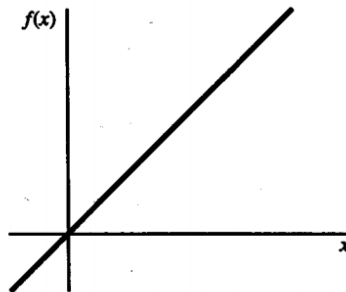
Gambar 2.7 Fungsi Sigmoid Tangen

2.4.2.3 Linear

Fungsi linear merupakan fungsi aktivasi dengan keluaran sama seperti masukannya. Fungsi ini biasa disebut sebagai fungsi identitas. Persamaan fungsi ini dan turunannya adalah sebagai berikut

$$f(x) = x \quad (2.16)$$

$$f'(x) = 1 \quad (2.17)$$



Gambar 2.8 Fungsi Linear

2.4.3 Algoritma Pelatihan Jaringan Syaraf Tiruan

2.4.3.1 Algoritma *Steepest Descent*

Algoritma *steepest descent* merupakan algoritma turunan pertama. Algoritma ini menggunakan turunan pertama dari fungsi *error* total untuk meminimalkan *error*. Gradien g didefinisikan sebagai turunan orde pertama dari fungsi *error* total

$$g = \frac{\partial E(x,w)}{\partial w} = \left[\frac{\partial E}{\partial w_1} \frac{\partial E}{\partial w_2} \frac{\partial E}{\partial w_3} \dots \frac{\partial E}{\partial w_N} \right]^T \quad (2.18)$$

Untuk perubahan bobotnya dapat dihitung dengan persamaan

$$w_{k+1} = w_k + \alpha g_k \quad (2.19)$$

dengan α adalah konstanta pembelajaran.

Proses pelatihan dari algoritma ini bersifat asimtotik konvergen. Saat mendekati nilai yang diinginkan, semua elemen tidak menjadi sangat kecil dan akan mengakibatkan perubahan bobot yang sangat kecil sekali.

2.4.3.2 Metode Newton

Metode Newton menggunakan turunan orde 2 dari fungsi *error* total yang diwakilkan dengan matriks Hessian H .

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \quad (2.20)$$

Untuk perubahan bobotnya dapat dihitung dengan persamaan

$$w_{k+1} = w_k - H_k^{-1} g_k \quad (2.21)$$

Sebagai turunan orde 2, matriks Hessian memberikan evaluasi yang tepat pada perubahan vektor gradien. Dengan membandingkan persamaan (2.19) dan (2.21), dapat diketahui

bahwa ukuran step yang cocok diberikan dengan invers dari matriks Hessian.

2.4.3.3 Metode Gauss-Newton

Jika metode Newton digunakan untuk memperbarui bobot, untuk mendapat nilai matriks Hessian H , maka turunan orde dua dari fungsi *error* total haruslah dihitung dan dapat memperlama waktu yang digunakan. Untuk mempermudah, maka dikenalkan matriks Jacobian

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{1,M}}{\partial w_1} & \frac{\partial e_{1,M}}{\partial w_2} & \dots & \frac{\partial e_{1,M}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{P,1}}{\partial w_1} & \frac{\partial e_{P,1}}{\partial w_2} & \dots & \frac{\partial e_{P,1}}{\partial w_N} \\ \frac{\partial e_{P,2}}{\partial w_1} & \frac{\partial e_{P,2}}{\partial w_2} & \dots & \frac{\partial e_{P,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{P,M}}{\partial w_1} & \frac{\partial e_{P,M}}{\partial w_2} & \dots & \frac{\partial e_{P,M}}{\partial w_N} \end{bmatrix} \quad (2.22)$$

Dikenalkan juga vektor *error* dengan bentuk

$$e = \begin{bmatrix} e_{1,1} \\ e_{1,2} \\ \dots \\ e_{1,M} \\ \dots \\ e_{P,1} \\ e_{P,2} \\ \dots \\ e_{P,M} \end{bmatrix} \quad (2.23)$$

Untuk perubahan bobotnya dapat dihitung dengan persamaan

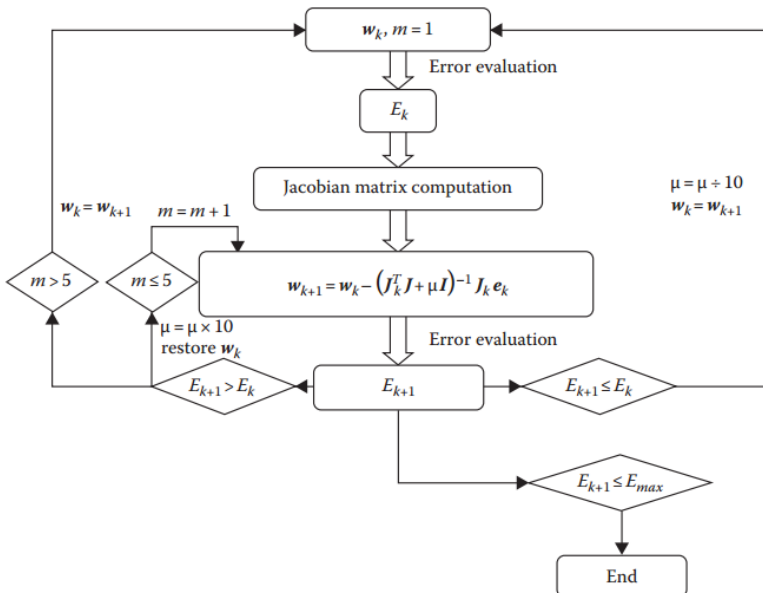
$$w_{k+1} = w_k - (J_k^T J_k)^{-1} J_k e_k \quad (2.24)$$

Kelebihan algoritma ini dibandingkan metode Newton adalah tidak perlu menghitung turunan kedua dari fungsi *error* total

2.4.3.4 Algoritma *Levenberg-Marquardt*

Algoritma *Levenberg-Marquardt* (LM) merupakan pengembangan dari algoritma *Back Propagation* (BP). Algoritma ini merupakan gabungan antara algoritma *steepest descent* dan algoritma Gauss-Newton. Algoritma LM menggabungkan kecepatan dari algoritma Gauss-Newton dan stabilitas dari algoritma EBP[14].

Adapun tahapan dari pelatihan algoritma LM dapat dilihat pada gambar 2 di bawah ini



Gambar 2.9 Blok diagram pelatihan menggunakan algoritma LM untuk 5 output[14]

Adapun dari gambar 2.9, dapat dijelaskan tahapan pelatihan algoritma LM sebagai berikut[14] :

1. Dengan memberikan bobot awal (acak), dihitung nilai SSE melalui persamaan

$$E(x, w) = \frac{1}{2} \sum_{p=1}^p \sum_{m=1}^m e_{p,m}^2 \quad (2.25)$$

dengan :

x = vektor input

w = vektor bobot

p = indeks pola

m = indeks output

$e_{p,m}$ = *error* pelatihan pada *output* m saat diberikan pola p yang didefinisikan sebagai

$$e_{p,m} = d_{p,m} - o_{p,m} \quad (2.26)$$

dengan :

d = output yang diinginkan

o = output yang dihasilkan

2. Menghitung elemen matriks Jacobian
3. Memperbarui bobot dan menghitung kembali SSE.
4. Jika nilai SSE yang diperbarui lebih besar, maka nilai w_k tidak diubah dan koefisien kombinasi dinaikkan dengan faktor 10 atau faktor lainnya. Kemudian kembali ke langkah 2 dan perbarui bobot lagi.
5. Jika nilai SSE yang diperbarui lebih kecil, maka nilai w_k diperbarui dan koefisien kombinasi diturunkan dengan faktor 10 atau faktor yang sesuai dengan faktor langkah 4.
6. Kembali ke langkah 2 dengan bobot baru sampai nilai SSE lebih kecil dari yang diinginkan.

2.5 Root Mean Square Error (RMSE)

Model JST yang dirancang harus dievaluasi performansinya. Metode pengukuran evaluasi performansi model JST dapat menggunakan metode RMSE. Metode RMSE mengukur dengan membandingkan perbedaan nilai estimasi dengan nilai aktual. Dari RMSE, dapat diketahui kualitas model JST apakah layak

atau tidak. Semakin kecil nilai RMSE semakin baik karena *error* yang dihasilkan hanya sedikit. Model yang bagus memiliki nilai RMSE mendekati nol. Persamaan RMSE dapat dilihat pada persamaan di bawah [15]

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.27)$$

dengan:

y_i = data aktual
 \hat{y}_i = data hasil prediksi
 n = jumlah data

Untuk membandingkan nilai RMSE pada data yang berbeda ukuran, maka dapat dilakukan dengan menormalisasi RMSE yang dikenal sebagai *normalized* RMSE (nRMSE). Persamaan untuk menghitung nilai nRMSE yaitu

$$nRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (2.28)$$

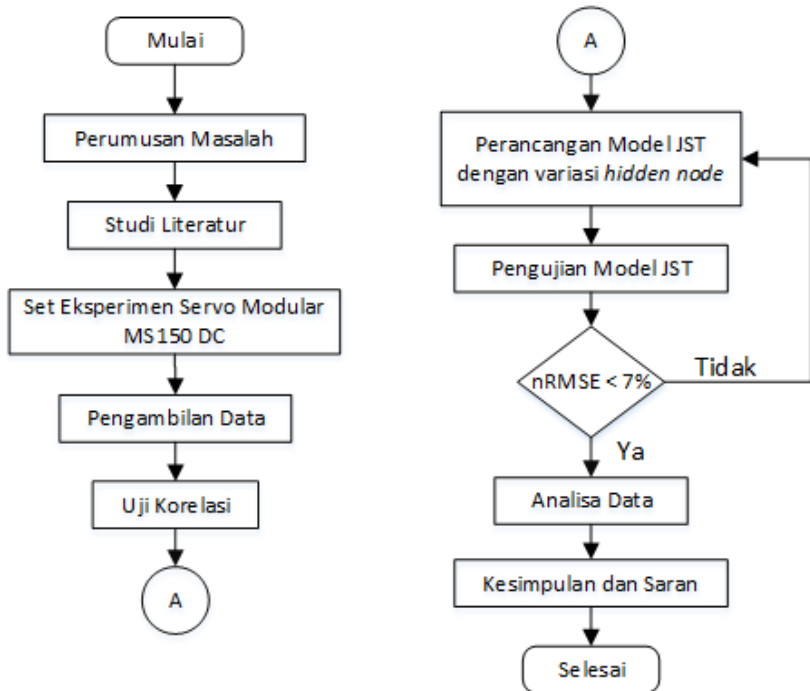
dengan:

y_{max} = data maksimum aktual
 y_{min} = data minimum aktual

“Halaman ini memang dikosongkan”

BAB III METODOLOGI PENELITIAN

Tahapan yang dilakukan dalam Tugas Akhir ini ditampilkan dengan sebuah diagram alir (*flowchart*) pada Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian

3.1 Perumusan Masalah

Motor DC digunakan baik pada sistem yang membutuhkan sensitivitas kecil hingga pada sistem dengan sensitivitas yang tinggi. Pada sistem dengan sensitivitas yang tinggi butuh untuk menjaga kepresisian semaksimal mungkin bahkan akibat adanya *load disturbance torque* pada *shaft* motor yang terhubung pada rotor. Oleh karena itu, diperlukan pengetahuan mengenai adanya gangguan torsi ini untuk menjaga kepresisian motor. Akan tetapi, dikarenakan tingkat kesulitan yang tinggi untuk melakukan

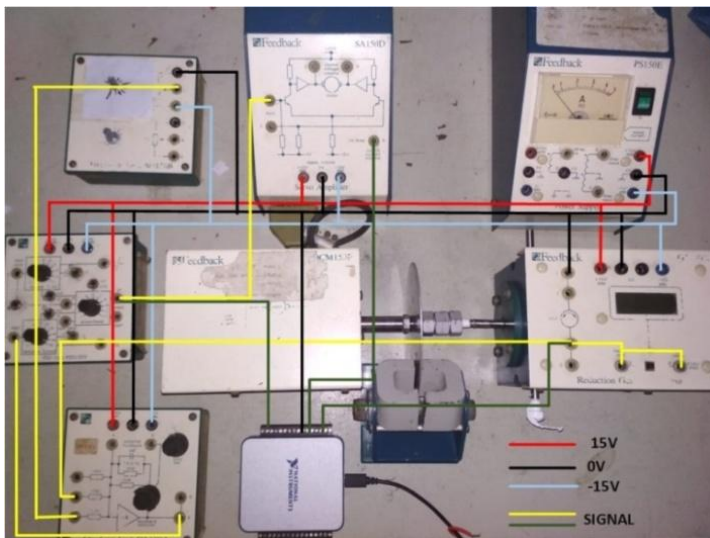
pengukuran secara langsung, estimasi digunakan untuk mengukur besarnya *load disturbance torque* pada *shaft* motor. Nilai estimasi ini nantinya dapat bermanfaat untuk meningkatkan *robust* dari sistem terhadap gangguan luar.

3.2 Studi Literatur

Pada penyusunan tugas akhir ini diperlukan studi literatur untuk membangun pemahaman awal hingga mendalam secara teoritis terhadap materi yang mendukung pada penelitian tugas akhir ini antara lain pemahaman mengenai sistem servo modular MS150 DC, *load disturbance torque*, dan jaringan syaraf tiruan.

3.3 Set-Up Sistem Servo Modular MS150 DC

Pada penelitian ini, digunakan sistem servo modular MS150 DC. Sebelum pengambilan data, dilakukan *set-up* terlebih dahulu dengan menghubungkan tiap instrumen yang diinginkan dengan kabel konektor. Adapun hubungan antar kabel konektor tersebut dapat dilihat pada gambar 3.2 dibawah ini.



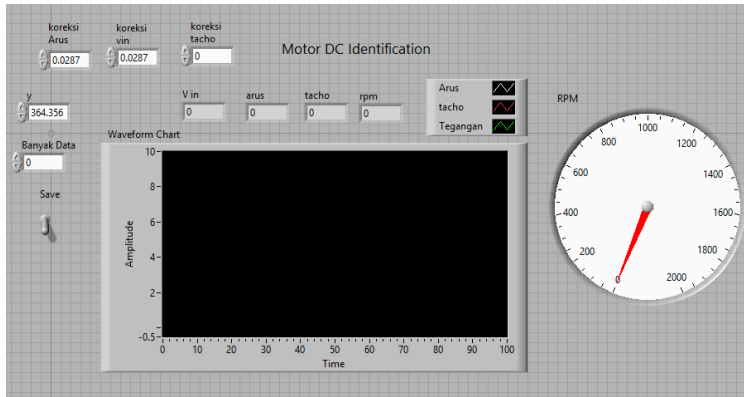
Gambar 3.2 Set-Up Sistem Servo Modular MS150DC

Pengendalian kecepatan motor dapat dilakukan dengan mengatur sinyal masukan tegangan yang masuk ke *servo amplifier*. Sinyal masukan yang diberikan ke motor DC berupa tegangan dengan rentang 0-15V. Kemudian perputaran motor dengan rentang 0-2000 rpm dideteksi oleh *tachogenerator* yang kemudian dikonversi menjadi tegangan 0-5 volt dan digunakan untuk sinyal masukan *op-amp* bersamaan dengan sinyal referensi yang diberikan *attenuator unit*. Keluaran sinyal *error* dari *op-amp* digunakan sebagai sinyal masukan *PID Unit*. Pada *PID Unit* terdapat resistor variabel yang digunakan untuk mengatur besarnya konstanta K_p , T_i , dan T_d . Tegangan keluaran *PID Unit* yang berupa sinyal kendali digunakan sebagai masukan bagi *servoamplifier*. Masukan tersebut akan menggerakkan motor DC, dimana tegangan terminal akan proporsional dengan kecepatan. Hasil pergerakan dari motor DC akan dikopel dengan *tachogenerator* dengan bagian *shaft* diberikan piringan aluminium. Diantara piringan aluminium tersebut, diletakkan *Load Unit* sebagai pemberi beban torsi yang dapat diatur sudutnya.

3.4 Pengambilan Data

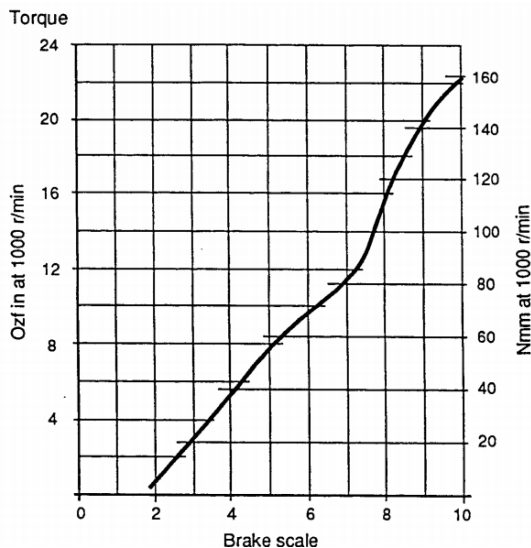
Data-data yang dibutuhkan untuk penelitian ini berupa data kecepatan dalam bentuk tegangan yang dikonversi oleh *tachogenerator*, data arus dalam bentuk tegangan, data tegangan yang masuk ke *Servo Amplifier*, dan data *load disturbance torque*.

Data dari kecepatan motor DC, tegangan input *Servo Amplifier*, dan arus masuk motor DC dibaca dengan bantuan NI USB-6001 dan PC yang sudah tersedia *software* LabVIEW. Tampilan dari program LabVIEW untuk pembacaan data dapat dilihat pada gambar 3.3 dibawah ini



Gambar 3.3 Tampilan GUI LabVIEW Pengambilan Data

Data tersebut akan diambil dengan waktu cuplik 100milidetik. Data akan diambil sebanyak 50 data pada tiap variasi *brake scale* dari *Load Unit*. Untuk menentukan besarnya *load disturbance torque* dari variasi *brake scale* yang diberikan, digunakan grafik hubungan antara *brake scale* dan *load torque* dibawah ini.



Gambar 3.4 Hubungan *brake scale* terhadap *load torque*[9]

Dengan memvariasikan mulai dari *brake scale* 1 hingga 9, maka didapatkanlah 450 set data. Data tersebut kemudian dibagi menjadi 75% sebagai data pelatihan dan 25% sebagai data pengujian yang selengkapnya dapat dilihat pada Lampiran B1 untuk data pelatihan dan Lampiran B2 untuk data pengujian.

3.5 Uji Korelasi

Uji korelasi merupakan suatu pengujian data untuk mengetahui tingkat keeratan hubungan antar variabel secara linier yang dinyatakan dalam bentuk koefisien korelasi (r). Nilai koefisien korelasi terletak di antara -1 hingga +1. Semakin mendekati nilai -1 atau +1, semakin erat hubungan antar variabel. Arti tanda negatif merupakan korelasi antarvariabel berlawanan. Sedangkan tanda positif merupakan korelasi antarvariabel searah.

Metode yang digunakan untuk menghitung koefisien korelasi adalah Pearson Correlation dengan persamaan koefisien korelasi pada persamaan (3.1) dibawah ini [16]

$$r = \frac{\sum(x_i - \bar{x}) \sum(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} \quad (3.1)$$

dengan :

- r = koefisien korelasi variabel x dengan y
- x_i = data variabel x ke- i
- \bar{x} = rata-rata variabel x
- y_i = data variabel y ke- i
- \bar{y} = rata-rata variabel y

Dari nilai koefisien korelasi inilah dapat ditentukan variabel apa saja yang dapat dijadikan sebagai data masukan dalam melakukan estimasi *load disturbance torque*. Semakin nilai koefisien korelasi mendekati nol, maka semakin besar error untuk membuat prediksi. Sebagai pedoman dalam memberikan pernyataan terhadap koefisien korelasi yang dihitung, maka dapat berpedoman pada ketentuan yang tertera pada tabel 3.1 dibawah ini.

Tabel 3.1 Pedoman Menentukan Tingkat Hubungan Berdasarkan Koefisien Korelasi [17]

Koefisien Korelasi	Tingkat Hubungan
0,00 – 0,199	Sangat rendah atau tidak berpengaruh
0,20 – 0,399	Rendah
0,40 – 0,599	Sedang
0,60 – 0,799	Kuat
0,80 – 1,000	Sangat kuat

3.6 Perancangan Model JST dengan Variasi *Hidden Node*

Perancangan model JST dilakukan dengan bantuan MATLAB. Langkah pertama yang dilakukan yaitu melakukan normalisasi data dari data yang telah diambil. Persamaan normalisasi yang digunakan adalah [18]

$$x' = \frac{0.8(x-b)}{(a-b)} + 0.1 \quad (3.2)$$

dengan :

- x' = data hasil normalisasi
- x = data asli
- a = nilai maksimum data asli
- b = nilai minimum data asli

Dari persamaan (3.2), maka diharapkan data hasil normalisasi memiliki rentang antara 0,1 hingga 0,9.

Data pelatihan yang disediakan dimasukkan dalam program MATLAB untuk melatih jaringan syaraf tiruan. Dikarenakan data yang masuk dalam model JST merupakan data normalisasi, maka nilai MSE yang menjadi target dari JST pada MATLAB merupakan nilai MSE dari data yang telah dinormalisasi. Sehingga perlu dihitung nilai MSE yang menjadi target di MATLAB.

Persamaan mengenai hubungan MSE dan RMSE adalah

$$MSE = RMSE^2 \quad (3.3)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.4)$$

Dengan menggunakan persamaan (2.28) dengan target nRMSE sebesar 7%, maka didapatkan target RMSE dengan y_{\max} dan y_{\min} didapatkan dari data *load disturbance torque* yang telah didapat sebesar 142 dan 0 berturut-turut yaitu

$$RMSE = 0.07(142 - 0) \quad (3.5)$$

$$RMSE = 9.94 \quad (3.6)$$

Dengan menggunakan target RMSE bernilai 9,59, dengan memasukkan ke persamaan (3.3), maka didapatkan nilai MSE

$$MSE = 9.94^2 \quad (3.7)$$

$$MSE = 98.8036 \quad (3.8)$$

Nilai MSE diatas merupakan nilai MSE yang sesungguhnya. Untuk mengetahui nilai MSE yang telah dinormalisasi dihitung dengan menggabungkan persamaan (3.4) yang dimodifikasi sehingga menjadi persamaan dibawah ini

$$MSE' = \frac{1}{n} \sum_{i=1}^n (y'_i - \hat{y}'_i)^2 \quad (3.9)$$

dengan :

MSE' = MSE hasil normalisasi

y'_i = data aktual normalisasi

\hat{y}'_i = data prediksi normalisasi

Dengan menggabungkan persamaan (3.9) diatas dengan persamaan (3.2), maka didapatkan persamaan

$$MSE' = \frac{1}{n} \sum_{i=1}^n \left(\left(\frac{0.8(x-b)}{(a-b)} + 0.1 \right) - \left(\frac{0.8(\hat{x}-b)}{(a-b)} + 0.1 \right) \right)^2 \quad (3.10)$$

$$MSE' = \frac{1}{n} \sum_{i=1}^n \left(\frac{0.8(x-b) - 0.8(\hat{x}-b)}{(a-b)} \right)^2 \quad (3.11)$$

$$MSE' = \frac{1}{n} \sum_{i=1}^n \left(\frac{0.8(x-\hat{x})}{(a-b)} \right)^2 \quad (3.12)$$

$$MSE' = \left(\frac{0.8}{a-b}\right)^2 \frac{1}{n} \sum_{i=1}^n (x - \hat{x})^2 \quad (3.13)$$

$$MSE' = \frac{0.64}{(a-b)^2} MSE \quad (3.14)$$

Substitusi nilai MSE dari persamaan (3.8), maka didapatkan nilai MSE normalisasi

$$MSE' = \frac{0.64}{(142-0)^2} \times 98.8036 \quad (3.15)$$

$$MSE' = 0,003136 \quad (3.16)$$

Dari MSE tersebut, maka rancangan arsitektur dibuat dengan parameter seperti pada tabel dibawah ini.

Tabel 3.2 Parameter Model JST

No	Parameter	Keterangan
1	Arsitektur	<i>Multi-layer feedforward</i>
2	Algoritma pelatihan	Levenberg-Marquardt
3	Jumlah <i>hidden layer</i>	1
4	Target MSE	0,003
5	Epoch maksimum	250
7	Koefisien kombinasi	0,001
8	Fungsi aktivasi	tangen sigmoid (<i>input ke hidden</i>), linier (<i>hidden ke output</i>)

Kemudian dilakukan percobaan dengan melakukan variasi pada jumlah *hidden node* pada *hidden layer* yang dimulai dari 1 hingga 20 *hidden node* pada model JST. Nilai keluaran dari JST merupakan nilai yang telah dinormalisasi, sehingga untuk mengetahui nilai RMSE sesungguhnya maka dilakukan denormalisasi melalui persamaan yang menggabungkan persamaan (3.3), dan (3.14).

$$RMSE' = \frac{0.8}{(a-b)} RMSE \quad (3.17)$$

$$RMSE = 177.5 \times RMSE' \quad (3.18)$$

dengan :

$RMSE$ = nilai RMSE sesungguhnya

$RMSE'$ = nilai RMSE akibat normalisasi data

Hasil dari data pelatihan tersebut kemudian dilihat nilai RMSE sesungguhnya yang terbaik dari semua jumlah *hidden node*, sehingga akan diperoleh jumlah *hidden node* yang tepat untuk melakukan estimasi. Nilai RMSE nantinya akan diubah dalam bentuk nRMSE dengan menggunakan persamaan (2.28).

3.7 Pengujian Model JST

Pengujian model JST dilakukan adalah dengan cara melakukan pengujian data yang belum pernah dilatih sebelumnya. Sama seperti saat pelatihan data, data yang akan diuji akan dinormalisasi sebelum masuk dalam model JST. Dari model JST tersebut akan dapat diketahui nilai RMSE. Nilai RMSE tersebut akan didenormalisasi sehingga mendapatkan nilai RMSE sesungguhnya dari data uji tersebut. Dari RMSE tersebut, maka dapat dihitung nilai nRMSE melalui persamaan (2.28). Jika nilai nRMSE lebih besar dari 7%, maka akan dilakukan kembali perancangan model JST dengan mengubah parameter JST.

3.8 Analisa Data

Analisa hasil pengujian dilakukan setelah melakukan pengujian sistem estimator yang telah dirancang. Estimator dapat dikatakan telah berhasil dilakukan atau dapat bekerja dengan baik apabila nilai nRMSE dari estimasi yang dihasilkan bernilai kurang dari 7%. Estimasi *load disturbance torque* yang dapat dikatakan memiliki akurasi yang lebih tinggi dan hasil yang lebih baik apabila nilai nRMSE yang dihasilkan semakin mendekati nilai nol.

3.9 Kesimpulan dan Saran

Bagian kesimpulan menunjukkan apakah masalah yang dirumuskan telah dapat terselesaikan atau tidak. Selain itu, kesimpulan juga merupakan jawaban dari rumusan masalah yang telah dibuat. Saran yang dapat digunakan untuk melakukan penelitian selanjutnya yang berhubungan dengan masalah yang

sama dengan penelitian tugas akhir terkait kemudian dapat ditulis juga pada subbab kesimpulan dan saran.

BAB IV HASIL DAN PEMBAHASAN

4.1 Uji Korelasi

Uji korelasi dilakukan dengan menggunakan *software* IBM SPSS *Statistics* 25. Uji korelasi digunakan mengetahui arah dan hubungan antara variabel independen dan dependen. Uji korelasi digunakan untuk menyeleksi variabel independen yang tidak berpengaruh terhadap variabel dependen. Hal ini nantinya akan digunakan untuk menentukan variabel apa saja yang dapat dijadikan sebagai data masukan dalam melakukan estimasi variabel *load disturbance torque* yang diinginkan.

Variabel independen yang digunakan adalah variabel yang telah diperoleh datanya, yaitu kecepatan sudut, tegangan masuk, dan arus. Data yang digunakan untuk uji korelasi merupakan data gabungan dari Lampiran B1 dan B2. Uji korelasi yang dilakukan menggunakan uji korelasi Pearson. Hasil dari uji korelasi dapat dilihat dari Tabel 4.1 dibawah ini.

Tabel 4.1 Hasil Uji Korelasi Nilai r

	Kecepatan sudut	Tegangan Masuk	Arus	<i>Load disturbance Torque</i>
Kecepatan sudut	1	-	-	-
Tegangan Masuk	-0,99	1	-	-
Arus	-0,962	0,978	1	-
<i>Load disturbance Torque</i>	-0,981	0,991	0,978	1

Tabel 4.1 menunjukkan hasil uji korelasi dari masing-masing variabel. Untuk mengetahui hubungan antara satu variabel independen terhadap *load disturbance torque* dapat dilihat dari nilai r pada baris *load disturbance torque*. Dilihat dari nilai yang tertera, maka dapat disimpulkan bahwa variabel kecepatan sudut, tegangan masukan, dan arus memiliki nilai koefisien korelasi berturut-turut sebesar -0,981, 0,991, dan 0,978 sehingga dapat dikategorikan memiliki hubungan yang sangat kuat terhadap *load disturbance torque* berdasarkan Tabel 3.1. Pada nilai koefisien korelasi kecepatan sudut, terdapat tanda negatif yang berarti bahwa semakin besar nilai *load disturbance torque*, maka kecepatan sudut akan berkurang. Sedangkan pada nilai koefisien korelasi pada tegangan masuk dan arus, tidak terdapat tanda yang berarti bahwa semakin besar nilai *load disturbance torque*, maka tegangan masuk dan arus akan bertambah.

4.2 Perancangan Estimator (JST)

Pada penelitian ini digunakan arsitektur jaringan syaraf tiruan dengan menggunakan satu *hidden layer*, satu *input layer*, dan satu *output layer*. Dengan menggunakan algoritma pelatihan Levenberg-Marquardt, digunakan koefisien kombinasi (μ) sebesar 0,0001 dan target MSE sesuai dengan perhitungan sebelumnya yakni sebesar 0,0003. Fungsi aktivasi tangent sigmoid dipilih sebagai fungsi aktivasi dari *input layer* ke *hidden layer*, dan fungsi aktivasi linear dipilih sebagai aktivasi dari *hidden layer* ke *output layer*.

Variabel yang digunakan pada *input layer* dipilih berdasarkan hasil dari pengujian korelasi sebelumnya, yaitu kecepatan sudut (ω), tegangan masukan (V_{in}), dan arus (I). Data yang digunakan adalah data yang telah diambil, yakni sebanyak 450 set data. Data tersebut dibagi menjadi dua bagian, yaitu sebanyak 342 set data untuk data pelatihan dan 108 set data untuk validasi data.

Pada penelitian ini, dilakukan variasi jumlah *hidden node*. Jumlah *hidden node* tersebut diharapkan mampu meningkatkan performa dan keakuratan dari hasil keluaran JST[19]. Untuk mengetahui berapa jumlah *hidden node* yang akan menghasilkan

nilai terbaik, dilakukan percobaan dengan menggunakan *hidden node* dengan jumlah 1 hingga 20 dengan menggunakan algoritma jaringan syaraf tiruan. Dari percobaan tersebut kemudian dianalisa hasil yang didapat dilihat dari nilai nRMSE yang dihasilkan.

4.2.1 Pelatihan JST

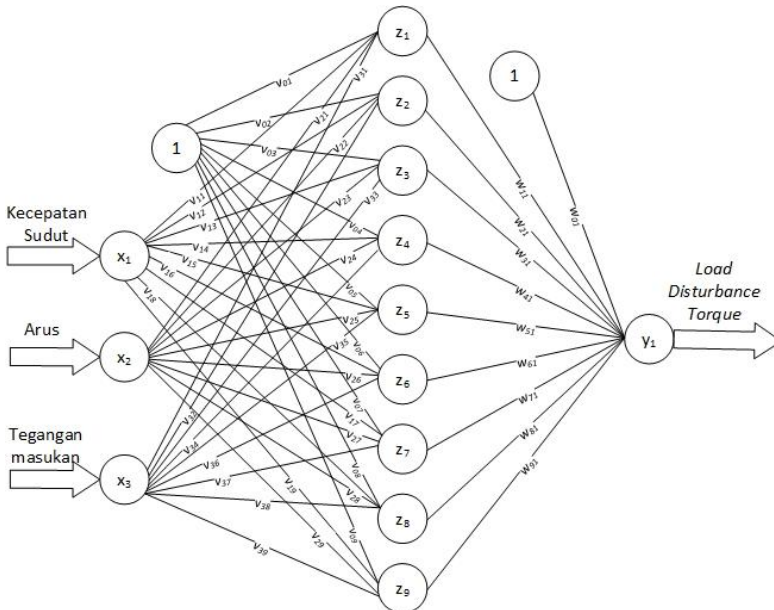
Variabel yang dapat digunakan sebagai variabel masukan pada *input layer* untuk melakukan estimasi *load disturbance torque* berdasarkan hasil pengujian data adalah arus dan tegangan masukan. Data untuk pelatihan dapat dilihat selengkapnya pada Lampiran B1. Kemudian dilakukan pelatihan data untuk estimasi *load disturbance torque* dengan variasi jumlah *hidden node*. Nilai RMSE dan nRMSE dari setiap *hidden node* dapat dilihat pada Tabel 4.2 dibawah ini.

Tabel 4.2 Nilai RMSE dan nRMSE Untuk Variasi *Hidden Node*

Jumlah <i>Hidden Node</i>	RMSE	nRMSE	Epoch
1	7,023853	4,946375	1
2	9,536543	6,715875	2
3	7,03681	4,9555	3
4	6,625365	4,66575	2
5	6,33746	4,463	2
6	8,586385	6,04675	3
7	6,64205	4,6775	3
8	7,958568	5,604625	2
9	5,325355	3,75025	2
10	9,07167	6,3885	2
11	6,559513	4,619375	4
12	8,474383	5,967875	3
13	7,65806	5,393	2
14	6,626785	4,66675	2
15	9,018065	6,35075	3
16	6,836235	4,81425	2
17	8,347825	5,87875	2

Jumlah <i>Hidden Node</i>	RMSE	nRMSE	Epoch
18	7,03042	4,951	2
19	8,03649	5,6595	2
20	5,730588	4,035625	2

Tabel 4.2 menjelaskan hasil estimasi berupa nilai RMSE dan nRMSE pelatihan. Dari 20 variasi tersebut, yang memiliki nilai RMSE dan nRMSE terkecil yaitu sebesar 5,325355 dan 3,75025% berturut-turut pada jumlah *hidden node* yang digunakan sebanyak 9 *hidden node*. Sehingga, dari 20 variasi *hidden node* tersebut, maka dipilihlah arsitektur JST dengan 1 *hidden layer* dengan 9 *hidden node* sebagai estimator *load disturbance torque*. Adapun arsitektur JST yang digunakan ditunjukkan pada gambar dibawah ini.



Gambar 4.1 Arsitektur JST untuk 9 *hidden node*

Berdasarkan Gambar 4.1 diatas, besarnya bobot pada tiap *node* selengkapnya dapat dilihat pada Tabel 4.3 dan Tabel 4.4 dibawah ini.

Tabel 4.3 Nilai bobot JST dari *input layer* ke *hidden layer*

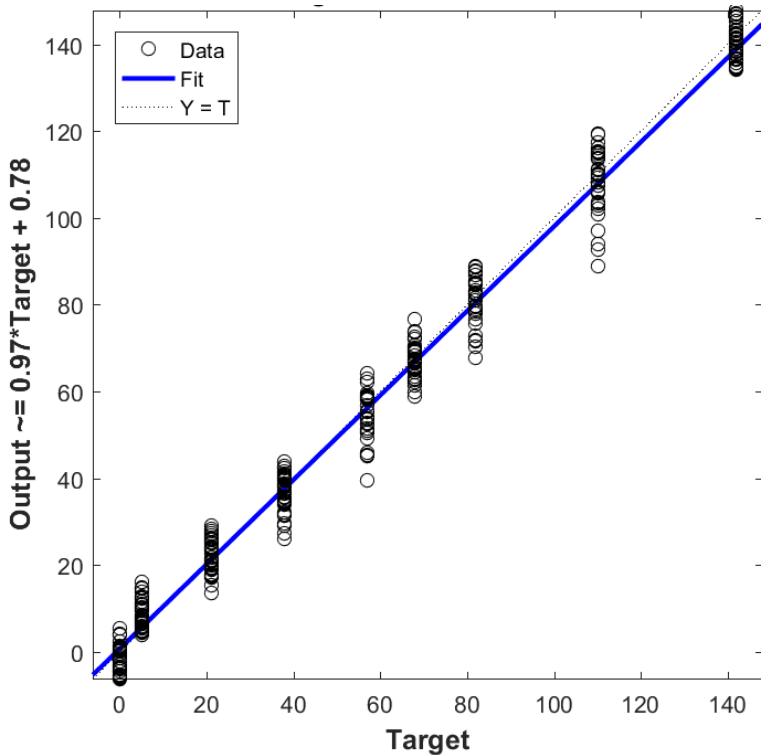
Node Asal	Node Tujuan	Notasi	Nilai
Bias	<i>Hidden Node 1 (z₁)</i>	v_{01}	8,841051
	<i>Hidden Node 2 (z₂)</i>	v_{02}	-8,09998
	<i>Hidden Node 3 (z₃)</i>	v_{03}	-2,48351
	<i>Hidden Node 4 (z₄)</i>	v_{04}	-1,38631
	<i>Hidden Node 5 (z₅)</i>	v_{05}	-4,3012
	<i>Hidden Node 6 (z₆)</i>	v_{06}	3,127364
	<i>Hidden Node 7 (z₇)</i>	v_{07}	-3,73584
	<i>Hidden Node 8 (z₈)</i>	v_{08}	-4,67576
	<i>Hidden Node 9 (z₉)</i>	v_{09}	-2,26752
Input 1 (x ₁)	<i>Hidden Node 1 (z₁)</i>	v_{11}	-5,59727
	<i>Hidden Node 2 (z₂)</i>	v_{12}	5,081097
	<i>Hidden Node 3 (z₃)</i>	v_{13}	1,487832
	<i>Hidden Node 4 (z₄)</i>	v_{14}	3,74803
	<i>Hidden Node 5 (z₅)</i>	v_{15}	0,161185
	<i>Hidden Node 6 (z₆)</i>	v_{16}	-7,174
	<i>Hidden Node 7 (z₇)</i>	v_{17}	6,554548
	<i>Hidden Node 8 (z₈)</i>	v_{18}	-2,74544
	<i>Hidden Node 9 (z₉)</i>	v_{19}	3,692227
Input 2 (x ₂)	<i>Hidden Node 1 (z₁)</i>	v_{21}	-4,16653
	<i>Hidden Node 2 (z₂)</i>	v_{22}	4,581879
	<i>Hidden Node 3 (z₃)</i>	v_{23}	5,568856
	<i>Hidden Node 4 (z₄)</i>	v_{24}	3,152378
	<i>Hidden Node 5 (z₅)</i>	v_{25}	7,174394
	<i>Hidden Node 6 (z₆)</i>	v_{26}	-1,38324
	<i>Hidden Node 7 (z₇)</i>	v_{27}	3,044731
	<i>Hidden Node 8 (z₈)</i>	v_{28}	2,255902
	<i>Hidden Node 9 (z₉)</i>	v_{29}	1,236468

<i>Node Asal</i>	<i>Node Tujuan</i>	<i>Notasi</i>	<i>Nilai</i>
Input 3 (x_3)	<i>Hidden Node 1 (z_1)</i>	v_{31}	-1,85386
	<i>Hidden Node 2 (z_2)</i>	v_{32}	2,999181
	<i>Hidden Node 3 (z_3)</i>	v_{33}	-4,49846
	<i>Hidden Node 4 (z_4)</i>	v_{34}	-5,3532
	<i>Hidden Node 5 (z_5)</i>	v_{35}	1,787429
	<i>Hidden Node 6 (z_6)</i>	v_{36}	0k,61789
	<i>Hidden Node 7 (z_7)</i>	v_{37}	0,686189
	<i>Hidden Node 8 (z_8)</i>	v_{38}	6,507461
	<i>Hidden Node 9 (z_9)</i>	v_{39}	6,264272

Tabel 4.4 Nilai bobot JST dari *hidden layer* ke *output layer*

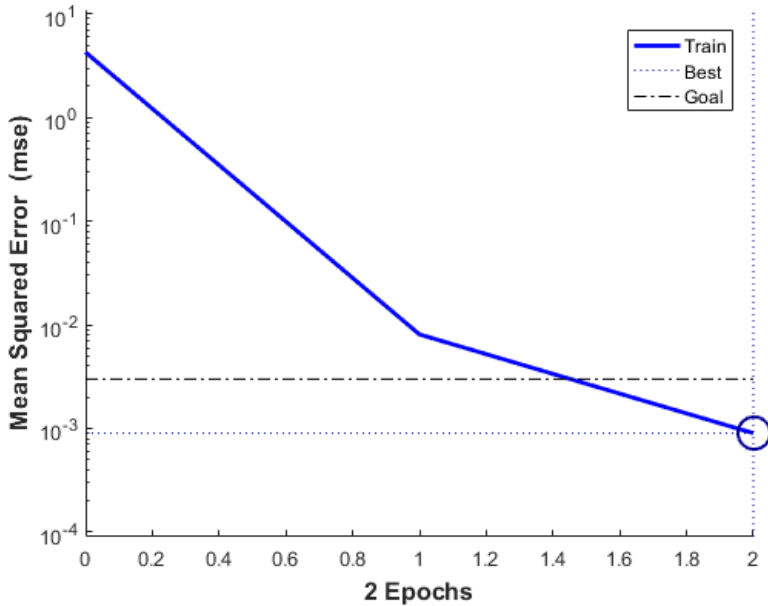
<i>Node Asal</i>	<i>Node Tujuan</i>	<i>Notasi</i>	<i>Nilai</i>
Bias	<i>Output Node</i> (y_1)	w_{01}	-0,2253
<i>Hidden Node 1 (z_1)</i>		w_{11}	1,267289
<i>Hidden Node 2 (z_2)</i>		w_{21}	0,541184
<i>Hidden Node 3 (z_3)</i>		w_{31}	0,507404
<i>Hidden Node 4 (z_4)</i>		w_{41}	-0,20428
<i>Hidden Node 5 (z_5)</i>		w_{51}	0,001927
<i>Hidden Node 6 (z_6)</i>		w_{61}	0,003341
<i>Hidden Node 7 (z_7)</i>		w_{71}	-0,27455
<i>Hidden Node 8 (z_8)</i>		w_{81}	0,041719
<i>Hidden Node 9 (z_9)</i>		w_{91}	0,519277

Dari program MATLAB yang telah dijalankan, didapatkan tiga grafik pada tiap *hidden node*, yaitu plot regresi, plot performansi, dan plot perbandingan hasil JST dengan target. Grafik hasil regresi untuk 9 *hidden node* ditunjukkan pada gambar dibawah ini.



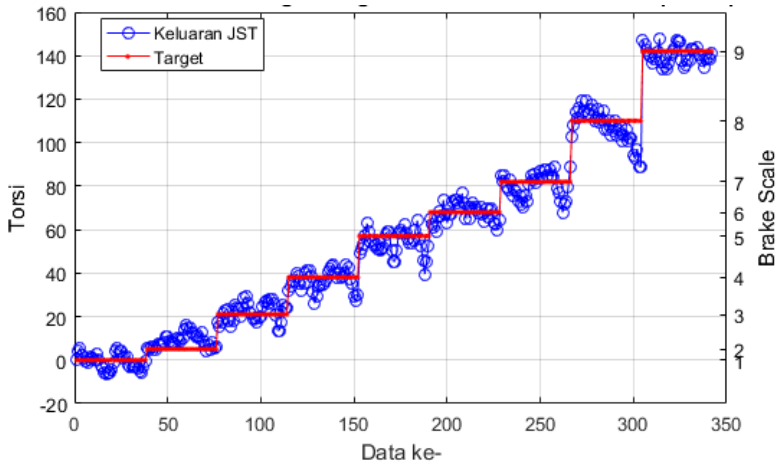
Gambar 4.2 Plot Regresi untuk 9 *hidden node*

Dari Gambar 4.2 diatas, dapat diketahui bahwa dengan 9 *hidden node* memiliki nilai R sebesar 0,99329. Dikarenakan nilai R yang mendekati nilai 1, maka dapat dikatakan model JST dengan 9 *hidden node* yang telah dirancang mampu melakukan estimasi dengan sangat baik. Untuk grafik performansi dari pelatihan dengan 9 *hidden node* ditunjukkan pada gambar dibawah ini.



Gambar 4.3 Performansi Pelatihan untuk 9 *hidden node*

Dari gambar 4.3 diatas, dapat diketahui bahwa pelatihan dengan 9 *hidden node* mampu mencapai *error* yang diinginkan, yakni sebesar 3×10^{-3} sehingga proses pelatihan berhenti pada *epoch* ke-2 dengan *error* $9,0013 \times 10^{-4}$. Perbandingan nilai keluaran JST dengan target dengan jumlah *hidden node* sebanyak 9 *node* tersebut juga dapat dilihat grafik estimasi dibawah ini.



Gambar 4.4 Hasil Pelatihan JST Menggunakan 9 *hidden node*

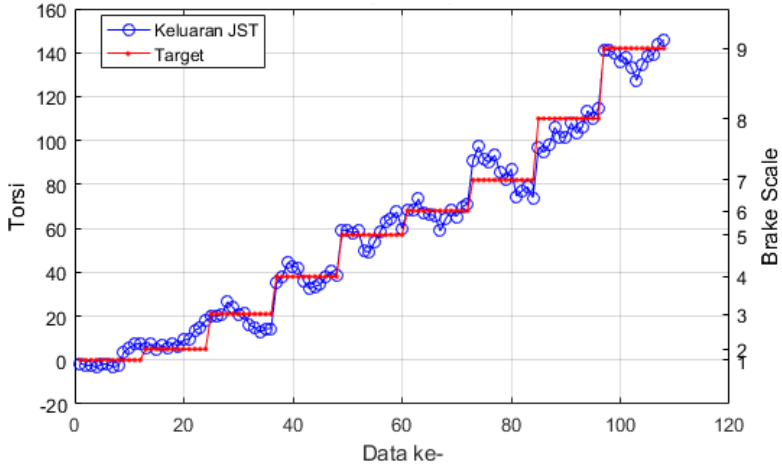
Gambar 4.4 diatas merupakan grafik perbandingan hasil keluaran JST yang ditunjukkan dengan lingkaran biru dan nilai sesungguhnya yang ditunjukkan dengan garis merah. Dari Gambar 4.4 tersebut, dapat diketahui bahwa hampir semua keluaran JST mendekati target yang diinginkan dengan nilai RMSE yang diperoleh sebesar 5,325355 dan nRMSE sebesar 3,75025%.

Dari Gambar 4.4, dapat dilihat bahwa JST mampu mengikuti target dengan baik pada *brake scale* kecil hingga *brake scale* maksimum. Terdapat beberapa data yang menyimpang, seperti pada torsi dengan *brake scale* 7 dan 8 dikarenakan data pelatihan pada *brake scale* tersebut memiliki data masukannya yang nilainya memiliki perbedaan yang cukup jauh sehingga model JST tidak mampu mengikuti target yang diinginkan pada data masukan yang berbeda jauh tersebut.

4.2.2 Pengujian JST

Model JST yang telah dibuat, diuji menggunakan data baru yang belum pernah dilatihkan sebelumnya. Data sebanyak 108 set telah disiapkan sebagai data pengujian JST, selengkapnya dapat dilihat pada Lampiran B2. Data tersebut dimasukkan dalam program MATLAB yang telah tersedia untuk dilakukan estimasi.

Grafik perbandingan antara keluaran JST dengan data pengujian dapat dilihat dari grafik dibawah ini.



Gambar 4.5 Hasil Pengujian JST

Dari grafik pada Gambar 4.5 diatas, dapat diketahui bahwa model JST yang telah dibuat telah mampu mengikuti target keluaran dengan baik. Masih terdapat sebagian kecil data yang menyimpang cukup tinggi, yakni pada torsi dengan *brake scale* diatas 7. Hal ini terjadidikarenakan semakin besar *load disturbance torque* yang diberikan, maka semakin tidak stabil keluaran dari arus dan tegangan masukan, sehingga JST tidak mampu mengikuti dengan baik target yang diinginkan. Selain itu, hali ini dapat terjadi akibat adanya *overfitting*. *Overfitting* terjadi jika data yang digunakan pada pelatihan terlalu akurat, sehingga pada pelatihan mampu menghasilkan *error* yang kecil tetapi saat diberikan data selain data pelatihan tidak bisa menghasilkan *error* yang kecil. Hal ini dibuktikan dari nilai RMSE yang dihasilkandari pengujian lebih besar dari pelatihan, yakni sebesar 5,733605. Sedangkan nilai nRMSE yang dihasilkan yakni sebesar 4,03775%. Dikarenakan nilai nRMSE berada dibawah nilai 7%, maka model JST yang telah dirancang telah mampu mengestimasi nilai *load disturbance torque*.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan kesimpulan dari tugas akhir ini sebagai berikut :

- a. Telah dilakukan perancangan model Jaringan Syaraf Tiruan (JST) untuk estimasi *load disturbance torque* dengan jumlah *node* terbaik sebanyak 3 *node* masukan, antara lain kecepatan sudut, tegangan masukan, dan arus, 9 *hidden node*, dan 1 *node* keluaran, yakni *load disturbance torque*.
- c. Model Jaringan Syaraf Tiruan (JST) yang telah dirancang mampu menghasilkan nilai nRMSE estimasi dibawah 7%, yakni pada pelatihan sebesar 3,75025% dan pengujian sebesar 4,03775%.

5.2 Saran

Dalam proses tugas akhir ini terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya, yaitu diterapkannya filter pada pembacaan data agar data yang dibaca dapat stabil dan digunakan NI-DAQ dengan rentang yang lebih lebar.

“Halaman ini memang dikosongkan”

DAFTAR PUSTAKA

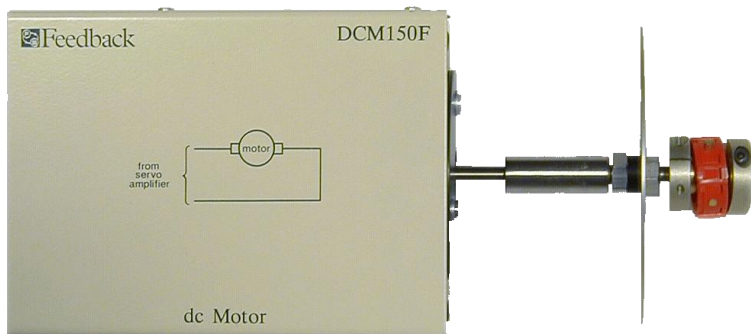
- [1] Yong Liu and Z.Q.:Howe,David Zhu, "Direct Torque Control of Brushless DC Drives with Reduced Torque Ripple," vol. 41, 2005.
- [2] N. Parhizkar, M. Shafiei, and M. Bahrami Koushahi, "Direct Torque Control of Brushless DC Motor Drives with Reduced Starting Current Using Fuzzy Logic Controller," vol. 50, 2014.
- [3] Danny Grignon, Xiang Chen, Narayan Kar, and Huijie Qian, "Estimation of Load Disturbance Torque for DC Motor Drive Systems Under Robustness and Sensitivity Consideration," *IEEE Transactions on Industrial Electronic*, vol. 61, pp. 930-942, 2013.
- [4] D.T Liem and K.K Ahn, "DC Motor Parameters Identification and Sensorless Torque Estimation Using Fuzzy PID," *12th International Conference on Control, Automation, and Systems*, pp. 76-81, October 2012.
- [5] Damien Koenig, Olivier Sename, Pascal Moulairé Kazusa Yamamoto, "Driver Torque Estimation in Electric Power Steering System Using an H inf/H 2 Proportional Integral Observer," *54th IEEE Conference on Decision and Control, CDC 2015*, Dec 2015.
- [6] Sumanto, *Mesin Arus Searah*. Yogyakarta: ANDI OFFSET, 1994.
- [7] Dinar Setyaningrum, "Desain dan Implementasi Model Reference Adaptive Control untuk Pengaturan Tracking Optimal Posisi Motor DC," Surabaya, 2012.
- [8] M. J. Burrige and Z Qu, "An improved nonlinear control design for series DC motors," *Computers and Electrical Engineering*, vol. 29, pp. 730-735, 2003.

- [9] Feedback Instruments Ltd, *Control & Instrumentation*. East Sussex: Feedback Instruments Ltd, 2013.
- [10] Katsuhiko Ogata, *System Dynamics*. United States of America: Pearson Prentice Hall, 1978.
- [11] Katsuhiko Ogata, *Modern Control Engineering 3rd ed.* London: Prentice Hall International, 1997.
- [12] L. Fausett, *Fundamentals of Neural Network Architecture, Algorithms, and Application.*: Prentice Hall, 1994.
- [13] J. C. Montesdeoca-Contreras, J. C. Zambrano-Abad, J. A. Morales-Garcia, and R. S. Ávila-Campoverde, "Virtual Speed Sensor for DC Motor using Back-Propagation Artificial Neural Networks," in *IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Ixtapa, Mexico, 2014.
- [14] Hao Yu and B. M. Wilamowski, *Industrial Electronic Handbook*, 2nd ed.: CRC Press, 2011, vol. V.
- [15] S. Papantoniou and D. Kolokotsa, "Prediction of Outdoor Air Temperature Using Neural Networks; Application in 4 European Cities," *Energy & Buildings*, pp. 1-31, 2015.
- [16] H Zhou, Z Deng, Y Xia, and M Fu, "A New Sampling Method in Particle Filter Based on Pearson Correlation Coefficient," *Neurocomputing*, pp. 208-215, 2016.
- [17] Sugiyono, *Statistika Untuk Penelitian*. Bandung, Jawa Barat, Indonesia: Alfabeta, 2017.
- [18] Jong Jek Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI Offset, 2005.

- [19] Harry Ganda Nugraha and Azhari SN, "Optimasi Bobot Jaringan Syaraf Tiruan Menggunakan Particle Swarm Optimization," in *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, vol. 8, 2014, pp. 25-36.
- [20] Duwi Priyatno, *5 Jam Belajar Olah Data dengan SPSS 17*, Joko Widiyatmoko, Ed. Yogyakarta, Indonesia: C.V ANDI OFFSET, 2009.

“Halaman ini memang dikosongkan”

LAMPIRAN A SPESIFIKASI MOTOR DC



- Jenis Motor : *-Permanent Magnet DC Motor*
-Single armature winding
- Arus maksimal : 2 A
- Kecepatan maksimum: 2000 rpm
- Tegangan kerja : 15 V
- Komponen tambahan : *-Extended shaft*
-Magnetic base

“Halaman ini memang dikosongkan”

LAMPIRAN B1
HASIL PENGAMBILAN DATA PELATIHAN

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque (Nmm)</i>
1	1005.889186	0.962019	0.276752	0
2	1001.632545	0.984592	0.296397	0
3	999.997814	0.991358	0.294970	0
4	1002.131313	0.976980	0.275180	0
5	1004.510296	0.969554	0.250202	0
6	1004.198903	0.963563	0.231711	0
7	1005.828046	0.950007	0.246540	0
8	1004.907562	0.958605	0.279631	0
9	1005.633605	0.956980	0.314196	0
10	1006.835520	0.943430	0.332496	0
11	1008.712322	0.933059	0.345693	0
12	1007.184396	0.940106	0.363279	0
13	1008.359581	0.930431	0.339165	0
14	1009.796185	0.926817	0.307634	0
15	1010.924219	0.919823	0.265277	0
16	1010.654074	0.909226	0.213009	0
17	1008.972550	0.913898	0.187542	0
18	1007.301096	0.920182	0.202543	0
19	1006.787062	0.916332	0.220374	0
20	1007.405646	0.914974	0.252709	0
21	1006.381248	0.928368	0.301222	0
22	1004.761066	0.944214	0.349534	0
23	1005.053148	0.952295	0.374979	0
24	1007.882305	0.952596	0.398211	0
25	1007.229175	0.951475	0.386404	0
26	1007.337579	0.939208	0.342345	0
27	1005.616447	0.940764	0.325010	0
28	1003.700198	0.946585	0.299709	0

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
29	1005.091604	0.938934	0.256044	0
30	1006.47693	0.937382	0.239703	0
31	1005.3929	0.940384	0.231503	0
32	1005.510757	0.941313	0.217569	0
33	1006.18097	0.934972	0.235493	0
34	1006.868445	0.923810	0.248851	0
35	1007.292323	0.903886	0.247803	0
36	1008.366557	0.888870	0.260104	0
37	1007.047087	0.893478	0.288061	0
38	1005.402213	0.902287	0.302341	0
39	1018.889666	1.006131	0.357457	5
40	1020.153359	1.011808	0.385388	5
41	1016.318228	1.038275	0.434552	5
42	1016.540707	1.032709	0.458492	5
43	1014.071785	1.037935	0.474231	5
44	1014.834896	1.024498	0.470645	5
45	1012.047839	1.034343	0.497151	5
46	1012.150668	1.036587	0.480732	5
47	1011.587968	1.050059	0.4536	5
48	1014.099549	1.047931	0.407642	5
49	1012.308036	1.05157	0.361133	5
50	1014.190372	1.035195	0.332654	5
51	1012.851459	1.033368	0.351244	5
52	1011.679971	1.049497	0.400512	5
53	1010.29316	1.053289	0.440201	5
54	1011.253359	1.050471	0.47306	5
55	1009.337652	1.061874	0.488413	5
56	1010.550275	1.067635	0.480545	5
57	1011.317954	1.061214	0.444041	5
58	1013.114404	1.061239	0.402988	5
59	1012.732233	1.05332	0.336176	5

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
60	1013.214157	1.053285	0.318173	5
61	1010.866614	1.0503	0.311074	5
62	1010.63424	1.062395	0.337533	5
63	1009.892411	1.065027	0.359096	5
64	1008.780424	1.079805	0.404557	5
65	1007.612268	1.073907	0.425234	5
66	1008.081303	1.082343	0.474538	5
67	1008.146834	1.06417	0.473236	5
68	1007.876437	1.07471	0.51153	5
69	1010.368278	1.041723	0.489383	5
70	1012.209953	1.038897	0.485814	5
71	1012.98186	1.017168	0.426086	5
72	1011.277643	1.041214	0.440715	5
73	1013.308006	1.019198	0.390222	5
74	1012.363632	1.047421	0.392434	5
75	1013.751389	1.011424	0.339216	5
76	1013.57712	1.016143	0.35067	5
77	1012.059809	1.114858	0.445576	21
78	1012.871568	1.09648	0.423021	21
79	1010.568146	1.107378	0.413328	21
80	1007.063115	1.132327	0.418914	21
81	1007.91677	1.133927	0.410907	21
82	1008.15463	1.151922	0.450149	21
83	1009.279277	1.125605	0.454873	21
84	1008.117012	1.117164	0.487671	21
85	1006.188693	1.127487	0.536242	21
86	1003.257	1.1498	0.595199	21
87	1007.753949	1.143363	0.624057	21
88	1008.671321	1.13893	0.623764	21
89	1009.045737	1.122462	0.600659	21
90	1005.902228	1.1411	0.611895	21

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
91	1002.978451	1.163195	0.623917	21
92	1002.104408	1.162408	0.623997	21
93	1005.090734	1.154608	0.622296	21
94	1004.309442	1.131659	0.573611	21
95	1003.172559	1.127353	0.551102	21
96	1003.692596	1.125591	0.523067	21
97	1003.422217	1.133798	0.518654	21
98	1005.116589	1.128395	0.496048	21
99	1005.089077	1.141638	0.486511	21
100	1004.237097	1.137709	0.448255	21
101	1002.234318	1.162545	0.453396	21
102	1002.371576	1.172945	0.44844	21
103	1000.710364	1.170659	0.435821	21
104	1001.415186	1.156546	0.41375	21
105	998.709025	1.163556	0.408736	21
106	998.449148	1.165774	0.408886	21
107	1001.026508	1.16006	0.420762	21
108	1005.216852	1.142105	0.448707	21
109	1009.007344	1.102951	0.463116	21
110	1009.505521	1.101478	0.518467	21
111	1007.647616	1.111834	0.564606	21
112	1004.973804	1.144812	0.614114	21
113	1005.907924	1.154415	0.600494	21
114	1005.08402	1.170876	0.576708	21
115	999.040473	1.216389	0.55829	38
116	995.84924	1.220108	0.530316	38
117	993.973483	1.223688	0.5487	38
118	996.180813	1.219193	0.597914	38
119	997.542387	1.229307	0.662663	38
120	1001.183438	1.213026	0.690288	38
121	1002.181991	1.201018	0.665992	38

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
122	999.79877	1.202922	0.613355	38
123	996.039893	1.218878	0.584008	38
124	993.750368	1.23939	0.596324	38
125	994.037046	1.243172	0.621989	38
126	996.935634	1.24174	0.664675	38
127	1001.865626	1.222586	0.6828	38
128	1005.265666	1.196148	0.649238	38
129	1005.02596	1.18541	0.585496	38
130	1000.975339	1.20712	0.54897	38
131	996.760252	1.222804	0.5271	38
132	995.118639	1.217773	0.54636	38
133	996.774515	1.20784	0.601762	38
134	999.090494	1.194046	0.645275	38
135	1001.268447	1.187909	0.681943	38
136	999.391632	1.215012	0.693981	38
137	997.501612	1.243157	0.675028	38
138	996.581637	1.250439	0.67701	38
139	993.817192	1.242642	0.665802	38
140	994.652698	1.210099	0.644045	38
141	996.900677	1.183199	0.654082	38
142	997.675634	1.187837	0.681005	38
143	999.687578	1.198658	0.680734	38
144	1002.109723	1.215159	0.687811	38
145	1000.281208	1.241764	0.674625	38
146	994.871733	1.271452	0.641532	38
147	993.313034	1.264504	0.580199	38
148	996.692668	1.244917	0.571223	38
149	999.130593	1.234205	0.610805	38
150	1002.312743	1.195116	0.600957	38
151	1005.716637	1.178	0.620842	38
152	1006.017712	1.181708	0.645689	38

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
153	989.768332	1.257865	0.666928	57
154	989.169235	1.260004	0.684254	57
155	988.446731	1.280988	0.683852	57
156	987.033947	1.313139	0.689185	57
157	984.976657	1.343996	0.723638	57
158	988.716862	1.331919	0.729359	57
159	992.045251	1.311209	0.718798	57
160	994.853579	1.312245	0.739051	57
161	996.239702	1.302841	0.725144	57
162	994.002112	1.305773	0.715668	57
163	991.43643	1.293827	0.692601	57
164	990.02234	1.285173	0.664123	57
165	989.605398	1.289251	0.658648	57
166	989.93744	1.293929	0.668634	57
167	987.573035	1.324318	0.707145	57
168	986.414868	1.321465	0.707435	57
169	985.217466	1.304337	0.701737	57
170	989.624191	1.266858	0.679453	57
171	992.214571	1.246995	0.659838	57
172	993.436508	1.260097	0.667779	57
173	992.809065	1.284484	0.705445	57
174	988.454053	1.32225	0.74074	57
175	990.33353	1.327884	0.761415	57
176	993.232488	1.317846	0.765604	57
177	994.879281	1.324957	0.79303	57
178	997.209378	1.308113	0.77017	57
179	996.079064	1.303388	0.755507	57
180	992.357274	1.296294	0.734103	57
181	990.122787	1.290092	0.717791	57
182	989.064743	1.299468	0.71913	57
183	987.562298	1.320038	0.750861	57

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
184	985.365749	1.343704	0.776705	57
185	990.449195	1.302586	0.737473	57
186	990.909201	1.286703	0.717031	57
187	994.846688	1.259009	0.689992	57
188	997.027322	1.234662	0.651868	57
189	994.391416	1.264224	0.671485	57
190	990.27126	1.298198	0.69104	57
191	986.617842	1.34585	0.800983	68
192	986.317127	1.319372	0.791149	68
193	986.395674	1.323456	0.765172	68
194	987.653734	1.310228	0.755029	68
195	983.260456	1.335112	0.792427	68
196	982.194891	1.359825	0.793094	68
197	983.574339	1.352696	0.78486	68
198	981.096981	1.390743	0.852157	68
199	985.235363	1.358744	0.812091	68
200	985.333527	1.340314	0.753839	68
201	985.054041	1.362689	0.789075	68
202	984.143574	1.363912	0.774034	68
203	981.362303	1.397076	0.767156	68
204	981.734939	1.403546	0.797488	68
205	981.003213	1.398217	0.829739	68
206	982.967618	1.396313	0.812655	68
207	985.372771	1.382893	0.806216	68
208	982.900857	1.412034	0.859986	68
209	986.886776	1.38921	0.825166	68
210	988.846083	1.352149	0.795728	68
211	986.76414	1.368129	0.844188	68
212	985.67018	1.348357	0.79707	68
213	980.424467	1.369609	0.778403	68
214	979.645683	1.380684	0.800861	68

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
215	981.088627	1.370146	0.792369	68
216	983.899378	1.374427	0.801256	68
217	986.746207	1.359955	0.819647	68
218	984.396718	1.372932	0.84676	68
219	987.62242	1.355351	0.822189	68
220	989.845856	1.33182	0.80874	68
221	988.683528	1.355961	0.855813	68
222	988.667977	1.345058	0.82196	68
223	982.979885	1.364354	0.826696	68
224	983.594754	1.361355	0.84711	68
225	986.157418	1.32653	0.801087	68
226	986.733494	1.33202	0.781342	68
227	988.543071	1.315414	0.77529	68
228	983.844029	1.332284	0.777363	68
229	981.171839	1.451651	0.893917	82
230	981.396701	1.439538	0.88754	82
231	979.575491	1.454171	0.906216	82
232	984.710939	1.413128	0.895897	82
233	981.729035	1.416372	0.888553	82
234	979.843704	1.43834	0.91067	82
235	981.163517	1.4019	0.873367	82
236	977.983813	1.416297	0.869166	82
237	981.40532	1.412604	0.891354	82
238	983.415409	1.388831	0.85339	82
239	983.010874	1.404693	0.861724	82
240	989.214517	1.371978	0.842362	82
241	985.554723	1.376817	0.829519	82
242	982.384904	1.402893	0.872385	82
243	982.431912	1.390967	0.841531	82
244	976.788388	1.433361	0.856012	82
245	975.889039	1.455621	0.869297	82

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
246	976.048492	1.442029	0.845937	82
247	975.294306	1.456714	0.893464	82
248	980.600709	1.437248	0.886452	82
249	979.199531	1.44706	0.895089	82
250	978.085476	1.462917	0.918131	82
251	978.06816	1.448707	0.887269	82
252	974.680874	1.464932	0.921381	82
253	975.082198	1.465951	0.922123	82
254	976.025893	1.452893	0.90883	82
255	976.175992	1.467642	0.928303	82
256	982.943564	1.439672	0.910115	82
257	984.409166	1.428242	0.920738	82
258	986.98866	1.426474	0.930354	82
259	988.094563	1.395382	0.889797	82
260	983.354873	1.405727	0.879139	82
261	984.311083	1.389047	0.849064	82
262	985.231729	1.361571	0.820674	82
263	982.722625	1.384019	0.840947	82
264	983.619911	1.386278	0.859846	82
265	977.091701	1.418748	0.873787	82
266	972.340131	1.46281	0.910268	82
267	972.127924	1.538422	1.001832	110
268	968.431159	1.568022	1.044461	110
269	964.718376	1.598696	1.098595	110
270	969.468848	1.565945	1.107591	110
271	968.034996	1.589602	1.133131	110
272	966.052415	1.61513	1.131492	110
273	971.076626	1.575359	1.079975	110
274	969.217077	1.592626	1.048603	110
275	966.251562	1.619897	1.05927	110
276	970.271545	1.583395	1.043021	110

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
277	967.261225	1.599741	1.071311	110
278	964.07037	1.617941	1.103804	110
279	968.255769	1.572594	1.072241	110
280	965.762982	1.585075	1.059915	110
281	963.251032	1.608507	1.095058	110
282	968.244695	1.56909	1.072889	110
283	965.810521	1.581944	1.086569	110
284	963.090422	1.598605	1.127736	110
285	969.494729	1.543616	1.087992	110
286	968.09447	1.548733	1.081832	110
287	965.286494	1.574702	1.107019	110
288	971.423277	1.533901	1.055111	110
289	970.150387	1.548104	1.059326	110
290	966.713428	1.57775	1.060943	110
291	973.307681	1.532471	1.028292	110
292	973.287307	1.541777	1.028334	110
293	970.411516	1.559231	1.040269	110
294	975.396838	1.51741	1.001809	110
295	974.645877	1.525608	1.025703	110
296	971.261091	1.545177	1.064134	110
297	975.627586	1.501439	1.048368	110
298	974.829144	1.508675	1.05655	110
299	971.826164	1.525821	1.050042	110
300	975.981326	1.483836	0.996412	110
301	975.051651	1.489092	0.962023	110
302	971.993903	1.512941	0.976782	110
303	975.693838	1.469029	0.952865	110
304	974.783478	1.466188	0.962365	110
305	951.768349	1.76698	1.239424	142
306	953.097444	1.757215	1.234835	142
307	953.821253	1.74506	1.21808	142

No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
308	953.793228	1.740094	1.22325	142
309	953.507554	1.732603	1.221138	142
310	954.223532	1.716636	1.171026	142
311	954.763426	1.717678	1.119421	142
312	953.390246	1.722351	1.135814	142
313	951.777703	1.728042	1.146821	142
314	950.064281	1.748072	1.137001	142
315	951.20515	1.702212	1.195014	142
316	954.234156	1.702086	1.171681	142
317	955.115954	1.706767	1.142302	142
318	957.193866	1.705167	1.154183	142
319	957.947714	1.713873	1.171455	142
320	957.453811	1.727154	1.150006	142
321	957.632372	1.732104	1.147862	142
322	954.959078	1.745548	1.167933	142
323	951.09956	1.762418	1.184866	142
324	950.417782	1.756543	1.181665	142
325	950.28118	1.752292	1.171125	142
326	953.692424	1.722451	1.192578	142
327	955.7191	1.706998	1.209596	142
328	957.151668	1.709739	1.213995	142
329	960.074746	1.700127	1.230085	142
330	957.565809	1.708188	1.252718	142
331	952.322362	1.7413	1.284311	142
332	952.040917	1.746719	1.263449	142
333	950.589195	1.741836	1.253805	142
334	951.360292	1.735386	1.136231	142
335	951.638539	1.715408	1.139409	142
336	951.730434	1.715976	1.13456	142
337	953.755551	1.701328	1.100824	142
338	954.143373	1.694091	1.124889	142

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque (Nmm)</i>
339	952.456977	1.718799	1.168838	142
340	953.083226	1.728553	1.182951	142
341	953.110828	1.725506	1.184566	142
342	950.948362	1.734037	1.227908	142

LAMPIRAN B2
HASIL PENGAMBILAN DATA PENGUJIAN

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
1	1007.7125	0.897391	0.311598	0
2	1007.51431	0.909672	0.295487	0
3	1005.99625	0.927470	0.272091	0
4	1008.51726	0.929301	0.243128	0
5	1009.55178	0.942391	0.234171	0
6	1007.08251	0.945775	0.218217	0
7	1006.91495	0.936962	0.233127	0
8	1004.43534	0.934583	0.250820	0
9	1001.18546	0.947847	0.302082	0
10	1001.18136	0.933595	0.321232	0
11	1002.37456	0.929404	0.349597	0
12	1002.04344	0.922995	0.346483	0
13	1016.77757	0.989792	0.304733	5
14	1014.91718	1.023629	0.35368	5
15	1017.18808	1.006147	0.359216	5
16	1015.33322	1.042937	0.422584	5
17	1016.33959	1.034688	0.438674	5
18	1013.89297	1.058013	0.481907	5
19	1015.48204	1.04597	0.484534	5
20	1012.49129	1.070074	0.490108	5
21	1011.97013	1.06722	0.438436	5
22	1008.37101	1.08793	0.405495	5
23	1011.3825	1.068684	0.355118	5
24	1009.92895	1.082426	0.348019	5
25	1005.44156	1.147024	0.522303	21
26	1003.69373	1.143265	0.493881	21
27	1002.3848	1.147058	0.492133	21
28	997.491061	1.164897	0.508015	21

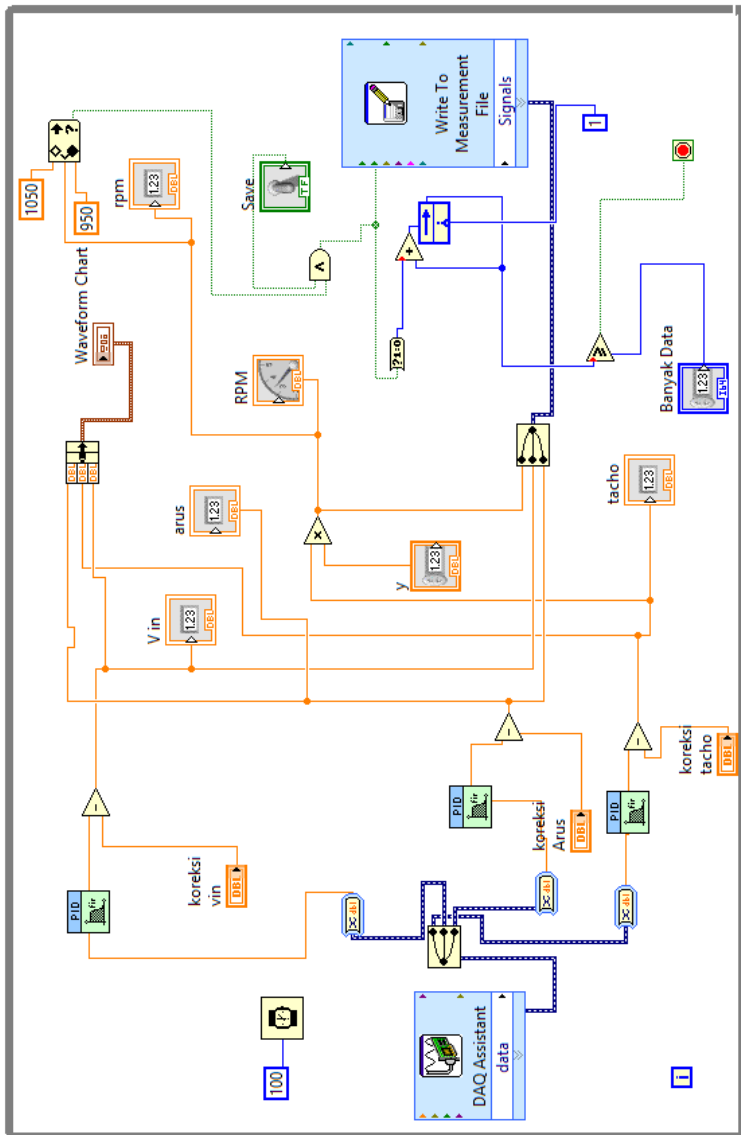
No	ω (rpm)	$V_{in}(V)$	I (A)	<i>Load Disturbance Torque (Nmm)</i>
29	999.340147	1.154585	0.510644	21
30	1001.84366	1.134806	0.51551	21
31	1003.14302	1.126274	0.55122	21
32	1008.19832	1.088039	0.559014	21
33	1012.86038	1.072244	0.589003	21
34	1015.55037	1.058033	0.584089	21
35	1014.9981	1.072954	0.595	21
36	1013.84382	1.082055	0.58356	21
37	1002.74232	1.208048	0.66403	38
38	1003.20048	1.215535	0.684985	38
39	1000.76238	1.239353	0.71914	38
40	998.261778	1.240123	0.69597	38
41	993.757056	1.236491	0.639877	38
42	994.036041	1.215779	0.559988	38
43	994.546901	1.196365	0.519127	38
44	994.489161	1.193737	0.550952	38
45	996.384171	1.195189	0.600553	38
46	997.906936	1.204322	0.65671	38
47	997.777837	1.206128	0.682696	38
48	997.478212	1.209292	0.659554	38
49	986.207218	1.32167	0.713614	57
50	987.813208	1.321772	0.739786	57
51	991.091905	1.317951	0.754263	57
52	995.27929	1.3111	0.778375	57
53	999.332366	1.270956	0.736545	57
54	996.277211	1.278956	0.71838	57
55	990.525017	1.297489	0.71967	57
56	987.117438	1.305463	0.744672	57
57	985.31218	1.328228	0.779705	57
58	985.511736	1.339242	0.798292	57
59	984.717194	1.356773	0.817322	57

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque</i> (Nmm)
60	988.615474	1.314421	0.778523	57
61	981.609835	1.360153	0.776408	68
62	982.453354	1.362462	0.795735	68
63	981.337142	1.391282	0.84732	68
64	985.617777	1.366601	0.795994	68
65	984.977715	1.355423	0.792322	68
66	986.691137	1.348803	0.814965	68
67	989.52251	1.319729	0.757628	68
68	987.820635	1.345163	0.798179	68
69	987.194967	1.35571	0.837479	68
70	985.465052	1.351416	0.788985	68
71	982.693064	1.374922	0.808148	68
72	982.091584	1.378342	0.826984	68
73	971.027735	1.464205	0.910766	82
74	968.634517	1.494443	0.931111	82
75	973.954925	1.484149	0.955446	82
76	975.42871	1.477027	0.948467	82
77	975.560619	1.490255	0.966955	82
78	984.614549	1.427534	0.928306	82
79	985.519527	1.416366	0.906815	82
80	985.849273	1.423065	0.940539	82
81	987.27917	1.378225	0.884235	82
82	981.981634	1.405472	0.895715	82
83	981.885587	1.411529	0.911865	82
84	981.458697	1.392383	0.859527	82
85	971.42969	1.502099	1.018961	110
86	972.660747	1.483047	1.030512	110
87	969.601405	1.499877	1.043201	110
88	966.018571	1.549017	1.06928	110
89	968.334108	1.527475	1.038303	110
90	968.349848	1.531264	1.000212	110

No	ω (rpm)	V_{in} (V)	I (A)	<i>Load Disturbance Torque (Nmm)</i>
91	968.190112	1.568463	1.031341	110
92	970.646943	1.545117	0.986459	110
93	968.998966	1.559393	0.981985	110
94	967.810319	1.593986	1.034882	110
95	969.960499	1.572109	1.042439	110
96	969.661541	1.585443	1.081147	110
97	950.729999	1.730657	1.256174	142
98	950.893482	1.73581	1.235545	142
99	951.161808	1.725256	1.211458	142
100	952.061814	1.701299	1.206233	142
101	952.181266	1.705115	1.146489	142
102	955.582593	1.687886	1.095091	142
103	959.378692	1.669413	1.097647	142
104	958.543817	1.70182	1.116864	142
105	959.362534	1.714143	1.146409	142
106	959.726555	1.714327	1.163423	142
107	954.684988	1.746745	1.172583	142
108	952.574655	1.758521	1.182846	142

LAMPIRAN C

PROGRAM PENGAMBILAN DATA LABVIEW



“Halaman ini memang dikosongkan”

LAMPIRAN D

PROGRAM PELATIHAN JST MATLAB

```
% Proses membaca data latih dari excel
Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:3);
target_latih = Data(:,4);
[m,n] = size(data_latih);
    % Pembuatan JST
    net = newff(minmax(data_latih),[jumlahnode
1],{'tansig','purelin'},'trainlm');

    % Memberikan parameter untuk mempengaruhi proses
pelatihan
    net.performFcn = 'mse';
net.trainParam.goal = 0.003;
    net.trainParam.show = 20;
    net.trainParam.epochs = 250;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;

% Hasil prediksi
hasil_latih = sim(net_keluaran,data_latih);
```

```
hasil_latih = ((hasil_latih-0.1)*(max_data-  
min_data)/0.8)+min_data;
```

```
% Performansi hasil prediksi  
target_latih_asli = xlsread(filename, sheet, xlRange);  
errorbaru=target_latih_asli'-hasil_latih;  
MSE = (1/n)*sum(errorbaru.^2);  
RMSE = sqrt(MSE);
```

```
figure,  
plotregression(target_latih_asli,hasil_latih,'Regression')
```

```
figure,  
plotperform(tr)
```

```
figure,  
plot(hasil_latih,'bo-')  
hold on  
plot(target_latih_asli,'ro-')  
hold off  
grid on
```

LAMPIRAN E

PROGRAM PENGUJIAN JST MATLAB

```
% load jaringan yang sudah dibuat pada proses pelatihan  
load netbest.mat
```

```
% Proses membaca data uji dari excel
```

```
Data = xlsread(filename, sheet, xlRange);  
data_uji = Data(:,1:3)';  
target_uji = Data(:,4)';  
[m,n] = size(data_uji);
```

```
% Hasil prediksi
```

```
hasil_uji = sim(net_keluaran,data_uji);  
nilai_error = hasil_uji-target_uji;
```

```
hasil_uji = ((hasil_uji-0.1)*(max_data-  
min_data)/0.8)+min_data;
```

```
% Performansi hasil prediksi
```

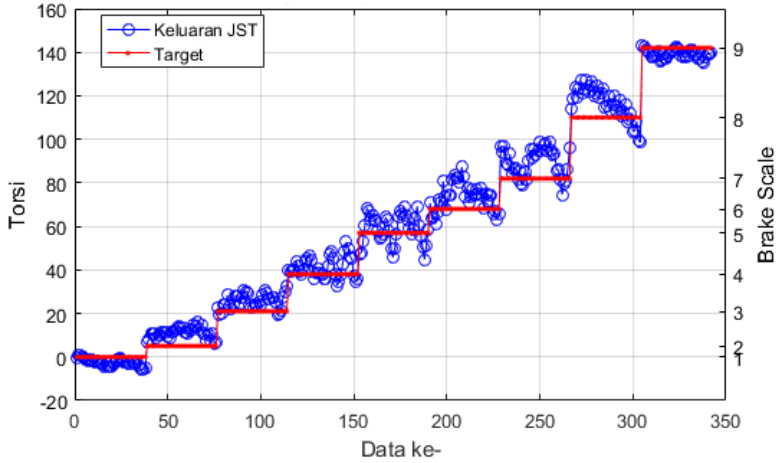
```
target_uji_asli = xlsread(filename, sheet, xlRange);
```

```
errorbaru=target_uji_asli'-hasil_uji;  
MSE = (1/n)*sum(errorbaru.^2);  
RMSE = sqrt(MSE)
```

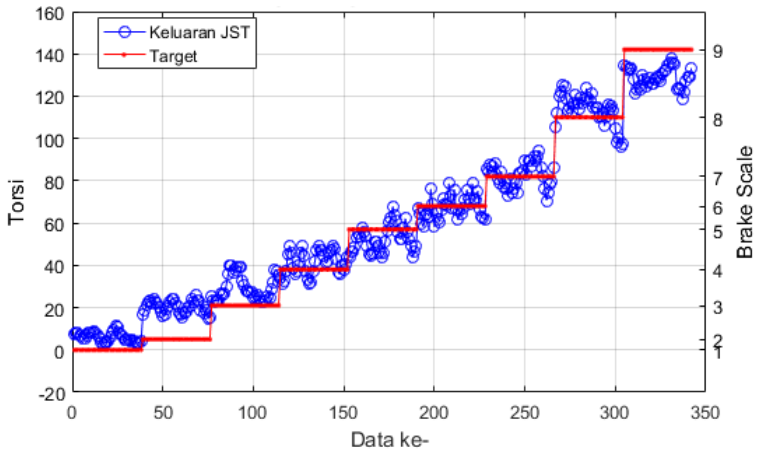
```
figure,  
plot(hasil_uji,'bo-')  
hold on  
plot(target_uji_asli,'ro-')  
hold off  
grid on
```

“Halaman ini memang dikosongkan”

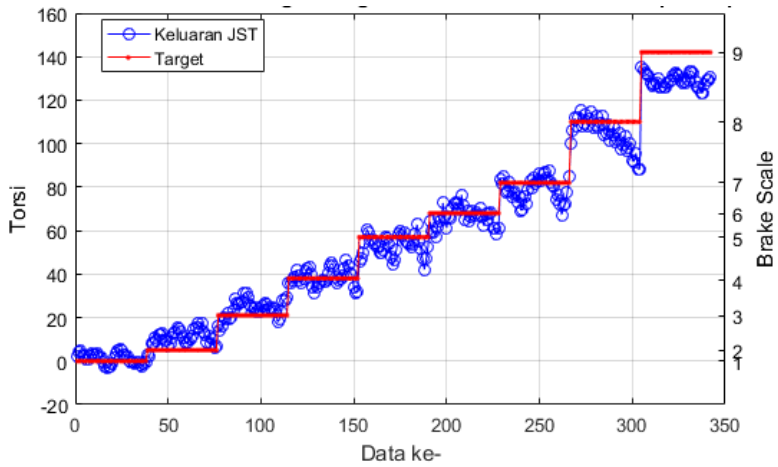
LAMPIRAN F GRAFIK PELATIHAN



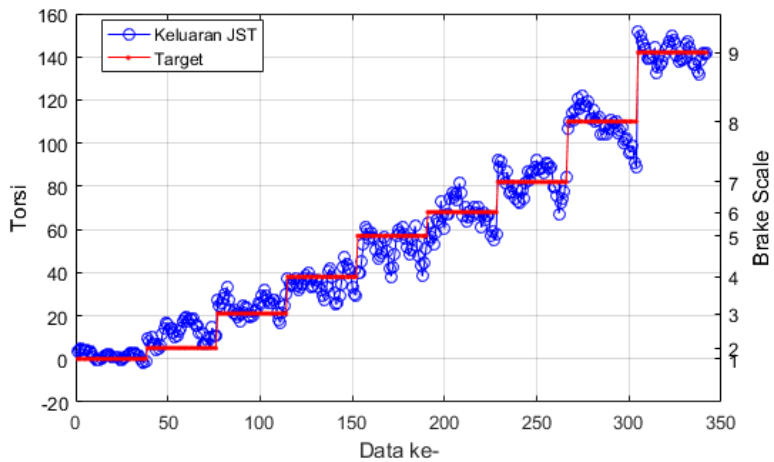
Gambar 1. Hasil Pelatihan JST menggunakan 1 *hidden node*



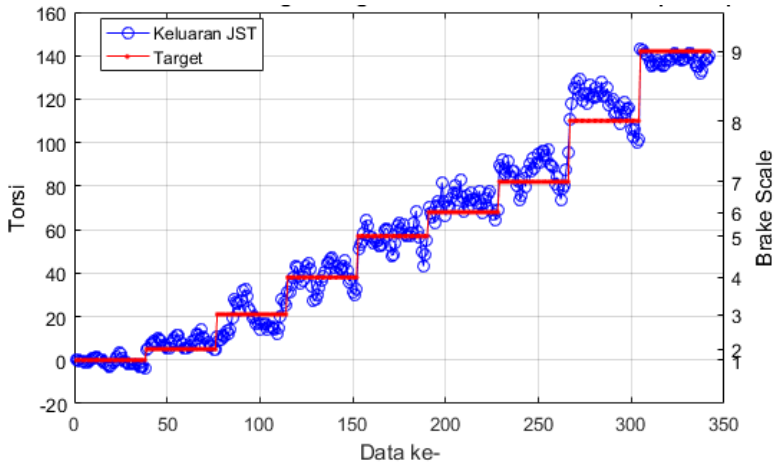
Gambar 2. Hasil Pelatihan JST menggunakan 2 *hidden node*



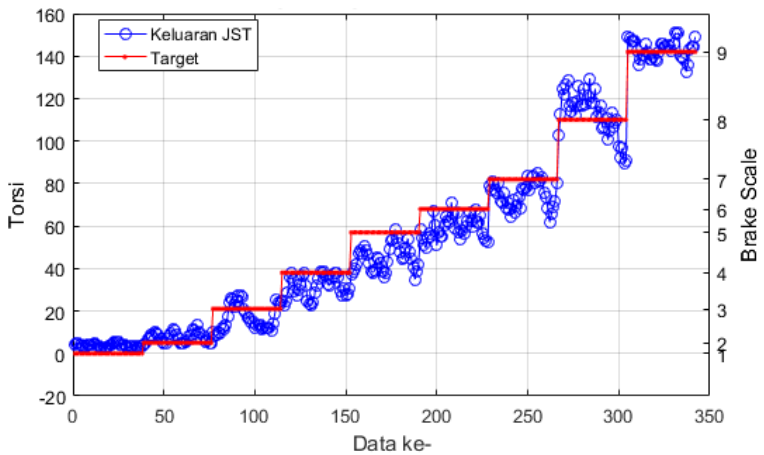
Gambar 3. Hasil Pelatihan JST menggunakan 3 *hidden node*



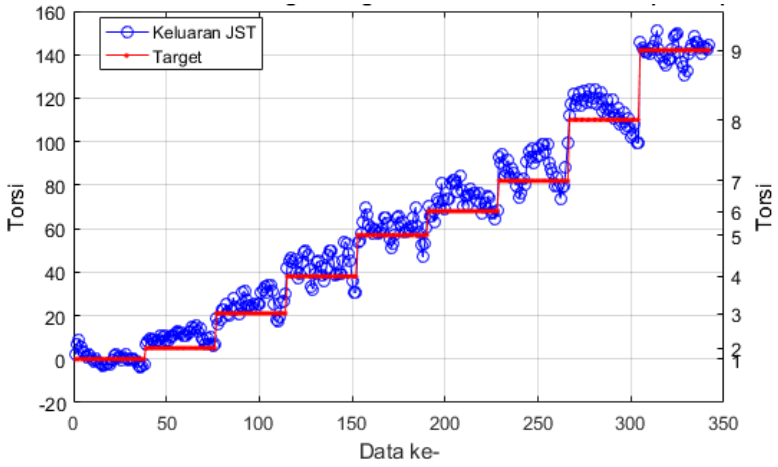
Gambar 4. Hasil Pelatihan JST menggunakan 4 *hidden node*



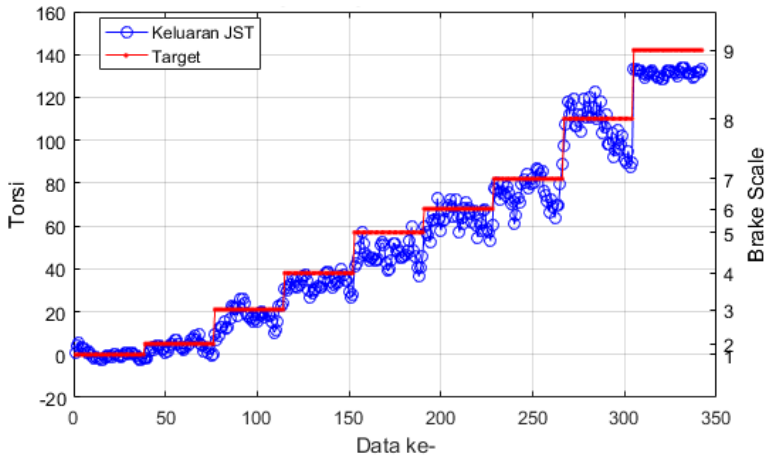
Gambar 5. Hasil Pelatihan JST menggunakan 5 *hidden node*



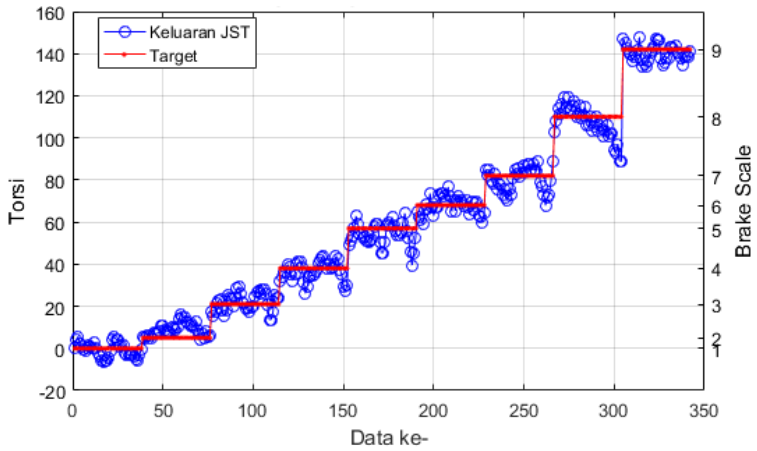
Gambar 6. Hasil Pelatihan JST menggunakan 6 *hidden node*



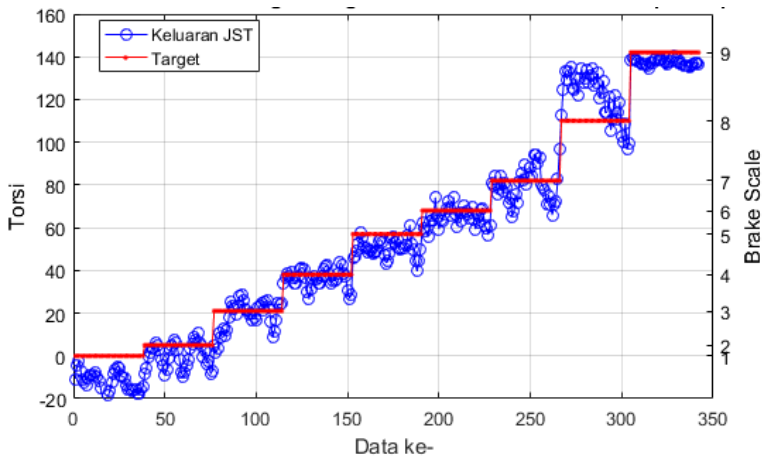
Gambar 7. Hasil Pelatihan JST menggunakan 7 *hidden node*



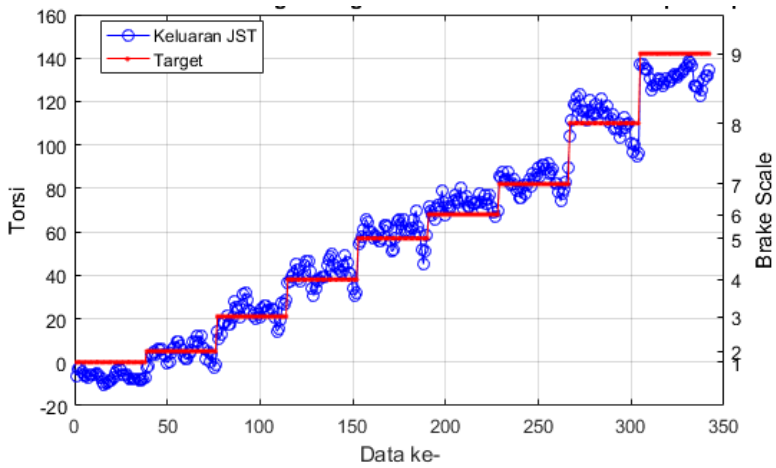
Gambar 8. Hasil Pelatihan JST menggunakan 8 *hidden node*



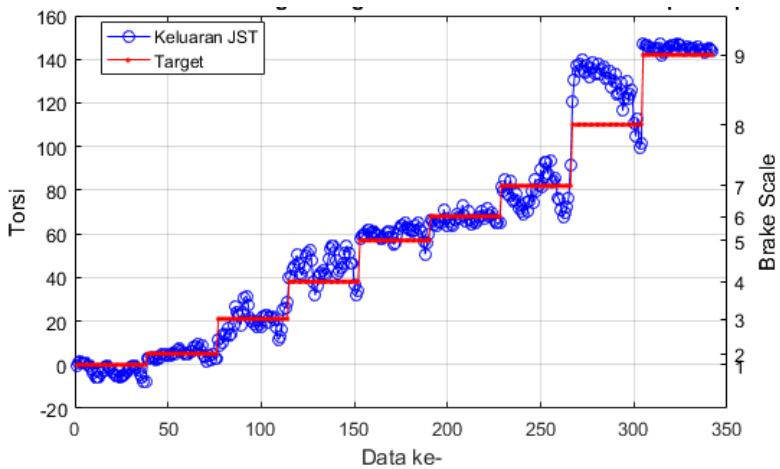
Gambar 9. Hasil Pelatihan JST menggunakan 9 *hidden node*



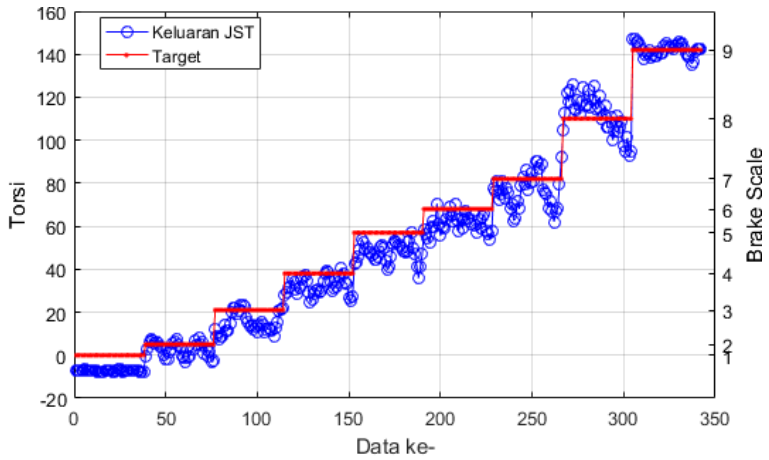
Gambar 10. Hasil Pelatihan JST menggunakan 10 *hidden node*



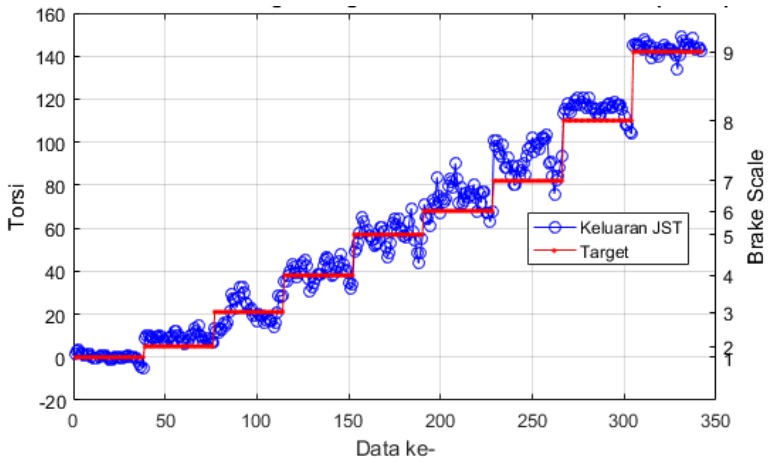
Gambar 11. Hasil Pelatihan JST menggunakan 11 *hidden node*



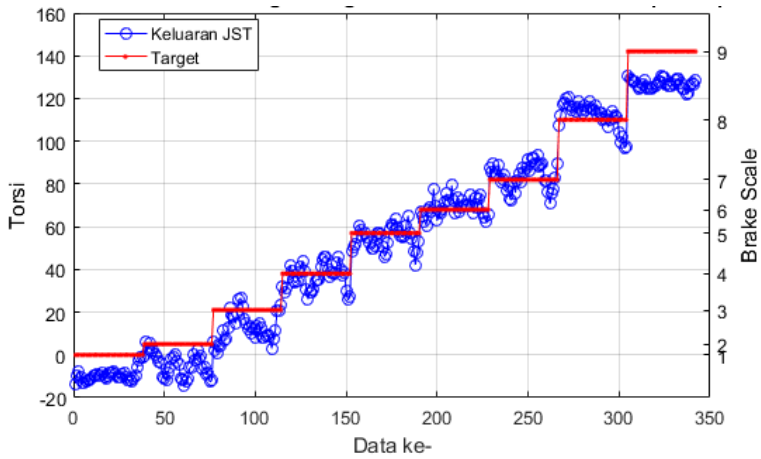
Gambar 12. Hasil Pelatihan JST menggunakan 12 *hidden node*



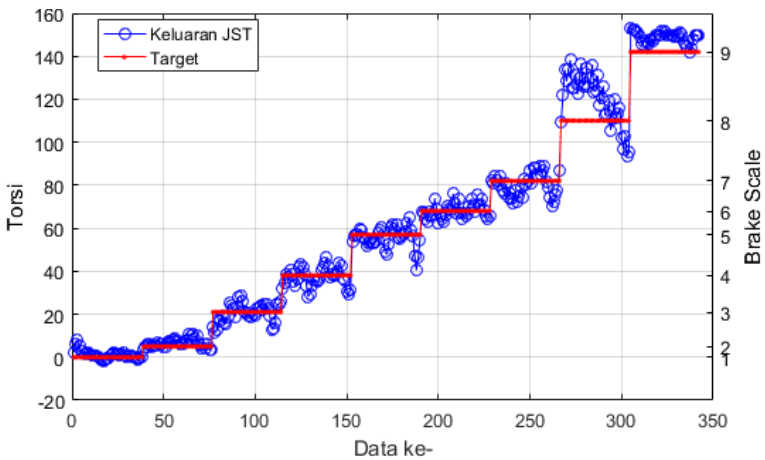
Gambar 13. Hasil Pelatihan JST menggunakan 13 *hidden node*



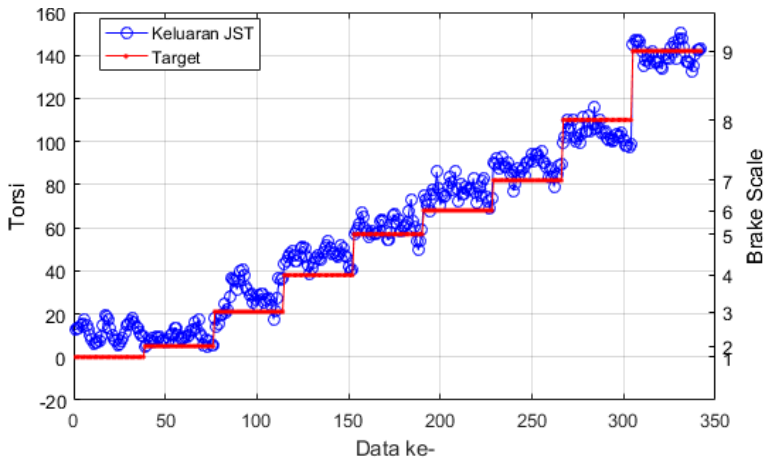
Gambar 14. Hasil Pelatihan JST menggunakan 14 *hidden node*



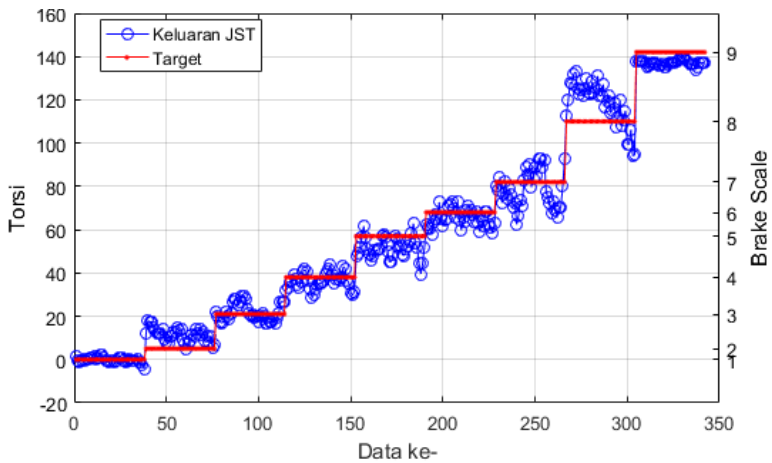
Gambar 15. Hasil Pelatihan JST menggunakan 15 *hidden node*



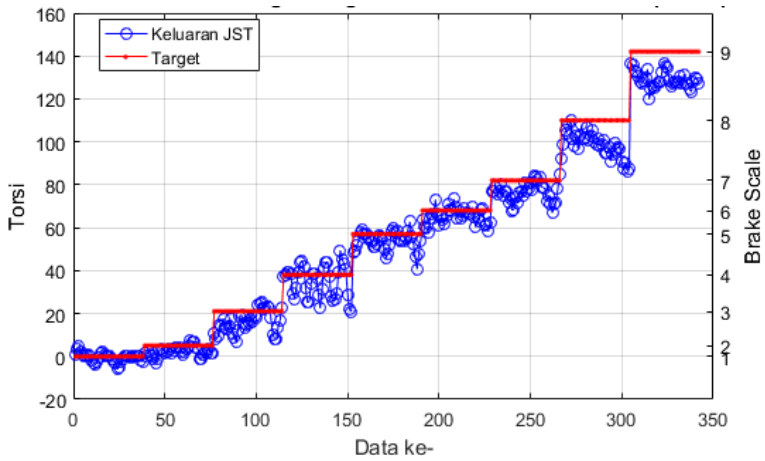
Gambar 16. Hasil Pelatihan JST menggunakan 16 *hidden node*



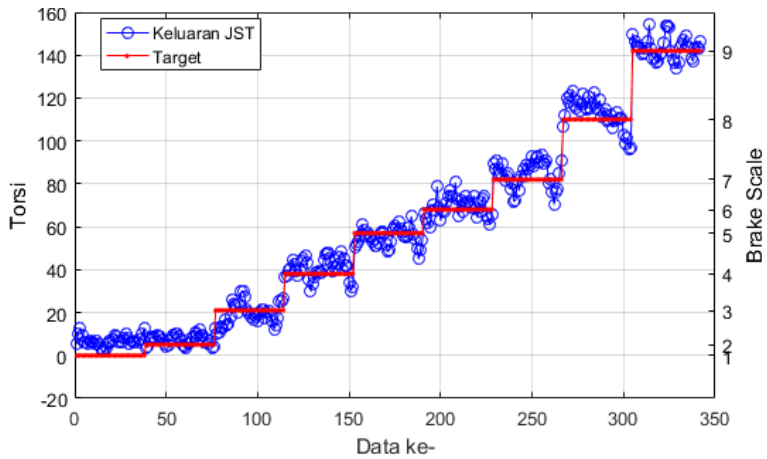
Gambar 17. Hasil Pelatihan JST menggunakan 17 *hidden node*



Gambar 18. Hasil Pelatihan JST menggunakan 18 *hidden node*



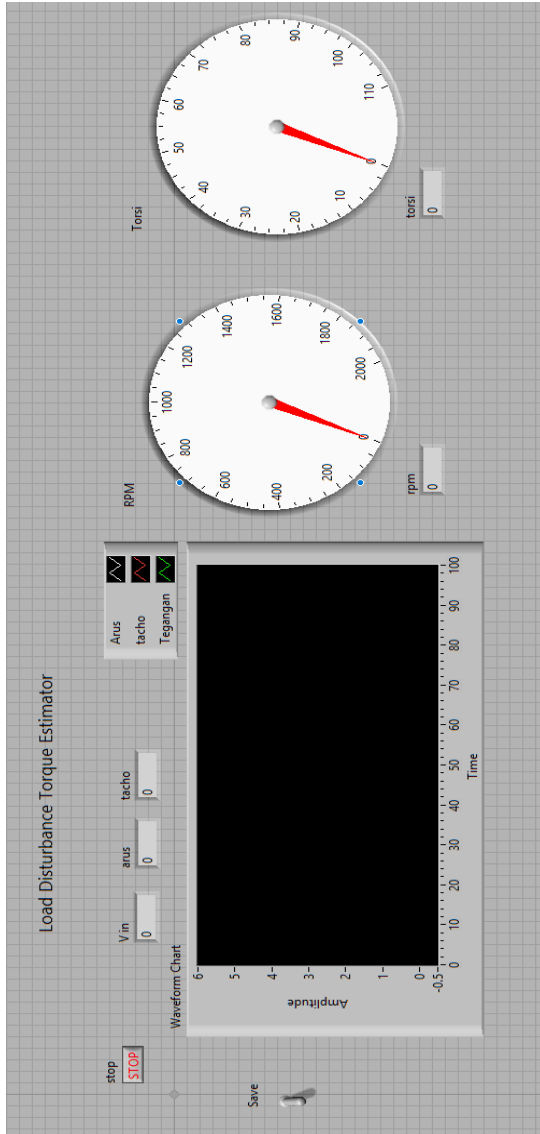
Gambar 19. Hasil Pelatihan JST menggunakan 19 *hidden node*

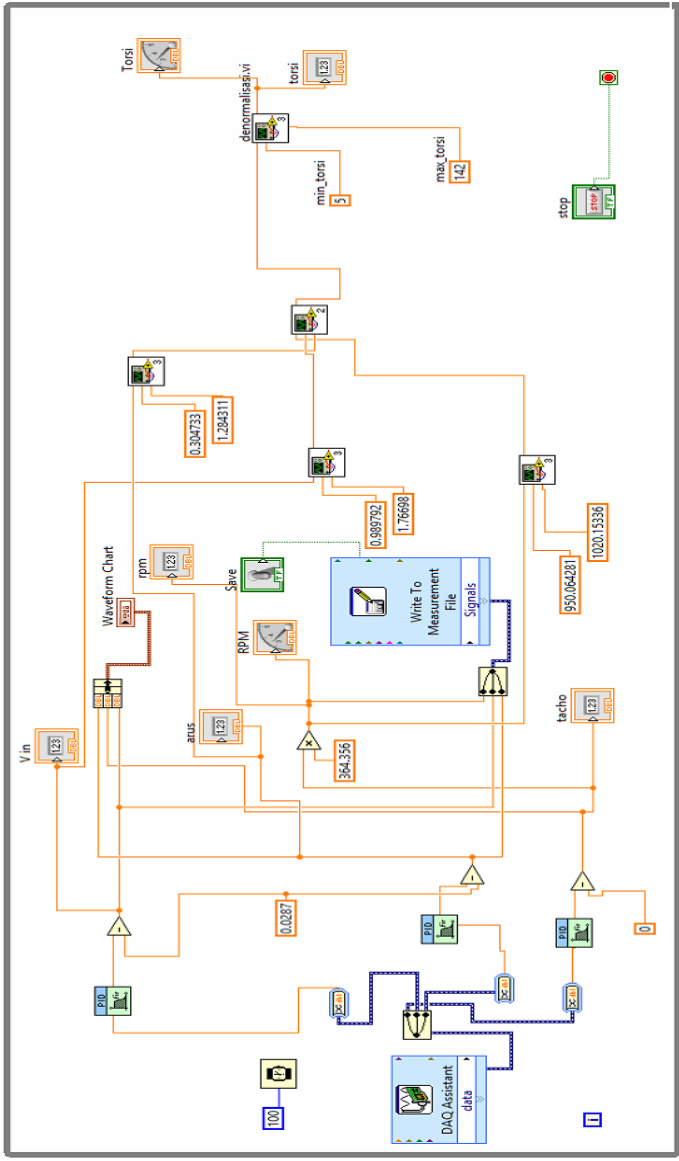


Gambar 20. Hasil Pelatihan JST menggunakan 20 *hidden node*

LAMPIRAN G

PROGRAM ESTIMATOR LOAD DISTURBANCE TORQUE LABVIEW





BIODATA PENULIS



Nama penulis Syahrul Munir dilahirkan di Sidoarjo, 15 Februari 1996 dari ayah bernama H. Ali Thoah (Alm.) dan ibu bernama Ummu Salamah. Saat ini penulis tinggal di Bebekan Masjid Gg II No. 34 RT 05 RW 02 Kecamatan Taman Kabupaten Sidoarjo. Penulis telah menyelesaikan pendidikan di MI Roudlotul Banat pada tahun, pendidikan di SMPN 1 Taman pada tahun, pendidikan di SMAN 15 Surabaya, dan sedang menempuh

pendidikan S1 Teknik Fisika FTI di Institut Teknologi Sepuluh Nopember Surabaya hingga sekarang.

Pada bulan Juli 2018, penulis telah menyelesaikan Tugas Akhir dengan judul **Perancangan Jaringan Syaraf Tiruan Sebagai Estimator Load Disturbance Torque Pada Sistem Servo Modular MS150 DC**. Bagi pembaca yang memiliki kritik, saran, maupun ingin berdiskusi lebih lanjut mengenai Tugas Akhir ini, maka dapat menghubungi penulis melalui email : syahrul.mnr15@gmail.com

“Halaman ini memang dikosongkan”