



TUGAS AKHIR - TE 141599

**ANALISA PERBANDINGAN ESTIMASI PARAMETER
RECURSIVE LEAST SQUARE (RLS) DAN PENDEKATAN
JARINGAN SYARAF TIRUAN PADA *SELF-TUNNING
REGULATOR (STR)* UNTUK PENGATURAN TEKANAN
*PROCESS RIG 38-714***

Febry Angga Gunawan
NRP 07111340000104

Dosen Pembimbing
Dr. Ir. Mohammad Rameli
Eka Iskandar, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE 141599

**ANALISA PERBANDINGAN ESTIMASI PARAMETER
RECURSIVE LEAST SQUARE (RLS) DAN PENDEKATAN
JARINGAN SYARAF TIRUAN PADA *SELF-TUNNING
REGULATOR (STR)* UNTUK PENGATURAN TEKANAN
*PROCESS RIG 38-714***

Febry Angga Gunawan
NRP 07111340000104

Dosen Pembimbing
Dr. Ir. Mohammad Rameli
Eka Iskandar, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

Comparative Study of Recursive Least Square (RLS) and Artificial Neural Network Approach for Parameter Estimation in Self-Tunning Regulator on Process Rig 38-714 Pressure Control

Febry Angga Gunawan
NRP 07111340000104

Supervisor
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

**“ANALISA PERBANDINGAN ESTIMASI PARAMETER
RECURCIVE LEAST SQUARE (RLS) DAN PENDEKATAN
JARINGAN SYARAF TIRUAN PADA *SELF-TUNNING
REGULATOR* UNTUK PENGATURAN TEKANAN PROCESS
RIG 38-714”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2018



Febry Angga Gunawan
NRP. 07111340000104

[Halaman ini sengaja dikosongkan]

**ANALISA PERBANDINGAN ESTIMASI PARAMETER
RECURCIVE LEAST SQUARE (RLS) DAN PENDEKATAN
JARINGAN SYARAF TIRUAN PADA SELF-TUNNING
REGULATOR UNTUK PENGATURAN TEKANAN PROCESS RIG
38-714**

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada

Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Ir. Mohammad Rameli
NIP. 19541227 1981031002

Eka Iskandar, ST., MT.
NIP. 19800528 2008121001



[Halaman ini sengaja dikosongkan]

**ANALISA PERBANDINGAN ESTIMASI PARAMETER
RECURCIVE LEAST SQUARE (RLS) DAN PENDEKATAN
JARINGAN SYARAF TIRUAN PADA SELF-TUNNING
REGULATOR UNTUK PENGATURAN TEKANAN PROCESS
RIG 38-714**

Febry Angga Gunawan
07111340000104

Pembimbing : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRAK

Pada sistem pengaturan proses sering terjadi proses perubahan beban. Perubahan tersebut dapat menimbulkan perubahan dinamika dari sistem. Di industri kontroler yang umum digunakan adalah PI atau PID konvensional. Namun, karena proses perubahan beban kontroler itu kurang bisa memenuhi spesifikasi. Pengaturan adaptif adalah salah satu metode pengaturan yang mana kontroler dapat memberikan respon memodifikasi perilakunya karena perubahan dinamika dari proses dan karakteristik dari gangguan. *Self-Tunning Regulator* (STR) adalah salah satu skema pengaturan adaptif. Estimasi parameter adalah salah satu bagian dari STR. Pada penelitian ini implementasi STR dengan estimasi parameter *Recursive Least Square* (STR RLS) dan pendekatan jaringan syaraf tiruan (STR NN) dilakukan pada sistem pengaturan tekanan di *Process Rig 38-741*. Hasil pengujian menunjukkan pada kondisi beban nominal STR NN dengan *learning rate* = 25 memiliki performa yang paling mendekati dengan hasil desain dengan nilai *overshoot* kecil dan *settling time* paling cepat dan sesuai. Pada pengujian dengan kondisi adanya perubahan beban STR NN dengan *learning rate* = 20 menunjukkan performa terbaik terhadap semua kriteria yang dipakai. Sementara untuk pengujian beban nominal terhadap variasi *set-point* STR NN dengan *learning rate* = 10 menunjukkan performa terbaik pada semua kriteria yang dipakai.

Kata kunci : *Self-Tunning Regulator, Pressure Process Rig 38-714, Recursive Least Square, Artificial Neural Network*

[Halaman ini sengaja dikosongkan]

***Comparative Study of Recursive Least Square (RLS) and
Artificial Neural Network Approach for Parameter
Estimation in Self-Tunning Regulator (STR) on Process
Rig 38-714 Pressure Control***

Febry Angga Gunawan
07111340000104

Supervisor : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRACT

In the process control system often occurs the process of changing the load. Such changes may lead to changes in the dynamics of the system. In most industries the controller used is a conventional PI or PID. However, because the process of changing the controller is less able to meet the specifications. Adaptive control is one of the control methods that the controller can respond modifies its behavior due to changes in the dynamics of the process and the characteristics of the disturbance. Self-Tunning Regulator (STR) is one of the adaptatif control schemes. The parameter estimation is one part of STR. In this research, the implementation of STR using parameter estimation of Recursive Least Square (STR RLS) and artificial neural network (STR NN) are applied to pressure control system in Process Rig 38-741. The test results show that for nominal load condition STR NN with learning rate = 25 has the best performance close to the design result with small overshoot value and the fastest and most appropriate settling time. Implementation for condition of the load change was existing, STR NN with learning rate = 20 shows the best performance against all the criteria used. While for implementation for the nominal load condition against set-point variations STR NN with learning rate = 10 shows the best performance on all criteria used.

Keywords : *Self-Tunning Regulator, Pressure Process Rig 38-714
Recursive Least Square, Artificial Neural Network.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “**ANALISA PERBANDINGAN METODE ESTIMASI PARAMETER PADA SELF-TUNNING REGULATOR UNTUK PENGATURAN TEKANAN PROCESS RIG 38-714**”. Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada :

1. Orang tua tercinta serta keluarga dan kerabat yang senantiasa memberikan doa serta dukungan
2. Bapak Mochamad Rameli selaku dosen pembimbing yang selalu memberikan pengarahan, saran dan motivasi dalam kelancaran tugas akhir ini
3. Bapak Eka Iskandar selaku dosen pembimbing yang telah banyak memberikan ilmu dan bimbingan selama penulis mengerjakan tugas akhir ini
4. Seluruh dosen, staf dan karyawan di Teknik Elektro Institut Teknologi Sepuluh Nopember
5. Ervi Ritya Zulvima atas semua bantuan dan dukungan yang telah diberikan
6. Seluruh teman-teman mahasiswa Teknik Elektro
7. Rekan-rekan asisten laboratorium Teknik Sistem Pengaturan
8. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu

Penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Kritik dan saran untuk perbaikan tugas akhir ini sangat diperlukan. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak.

Surabaya, Juni 2018

Febry Angga Gunawan
NRP 07111340000104

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN	viii
ABSTRAK.....	x
<i>ABSTRACT</i>	xii
KATA PENGANTAR	xiv
DAFTAR ISI	xvi
DAFTAR GAMBAR.....	xx
DAFTAR TABEL	xxiv
1 BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika	3
1.7 Relevansi	4
2 BAB 2 TINJAUAN PUSTAKA	5
2.1 Sistem Pengaturan Proses.....	5
2.2 Sistem Pengaturan Tekanan	5
2.3 <i>Pressure Process Rig 38-714</i> [6].....	6
2.3.1 Komponen-Komponen Pressure Process Rig 38-714	6
2.3.2 Cara Kerja Peralatan pada <i>Pressure Process Rig 38-714</i>	17
2.4 <i>Self-Tunning Regulator (STR)</i> [2].....	18
2.5 Kontroler PID [7]	19
2.5.1 Aksi Proporsional	20
2.5.2 Aksi Integral	20
2.5.3 Aksi Derivatif	20
2.6 <i>Data-Acquisition System (DAS)</i> [8]	21
2.6.1 DAS <i>Hardware</i>	22
2.6.2 DAS <i>Software</i>	23
2.7 Jaringan Syaraf Tiruan [9].....	24
2.7.1 Jaringan Syaraf Biologi.....	24
2.7.2 Konsep Dasar Model Jaringan Syaraf Tiruan	25
2.7.3 Arsitektur Jaringan Syaraf Tiruan.....	26

2.7.4	Fungsi Aktivasi	29
2.7.5	Proses <i>Learning</i>	31
2.7.6	Algoritma <i>Back Propagation</i>	31
2.8	Kriteria Kesalahan	33
2.8.1	Root Mean Square Error (RMSE)	33
2.8.2	<i>Integral of Time Multiplied by Square Error</i> (ITSE)	33
2.8.3	<i>Integral of Time Multiplied by Absolut Value of Error</i> (ITAE) 34	
2.8.4	<i>Integral of Square Time Multiplied by Square Error</i> (ISTSE)	34
2.9	LabVIEW [10].....	34
3	BAB 3 PERANCANGAN SISTEM	37
3.1	Arsitektur Sistem	37
3.1.1	Komputer	37
3.1.2	ADAM-5000/TCP [11]	37
3.1.3	<i>Plant Pressure Process Rig 38-714</i>	39
3.1.4	Jaringan Ethernet.....	39
3.1.5	LabVIEW	39
3.2	Perancangan Perangkat Akuisisi Data [11]	40
3.3	Desain I/V <i>Converter</i>	42
3.4	Identifikasi Sistem <i>Open Loop</i>	43
3.4.1	Identifikasi Statis Sistem <i>Open Loop</i>	44
3.4.2	Formulasi Model Sistem	44
3.4.3	Validasi Model	45
3.5	Penentuan Frekuensi Sampling.....	46
3.6	Desain Estimator Parameter Sistem.....	46
3.6.1	<i>Recursive Least Square</i> (RLS) <i>Model Estimation</i> [2]	46
3.6.2	Estimasi Parameter Model dengan Pendekatan Jaringan Syaraf Tiruan [13]	49
3.7	Desain <i>Self-Tunning Regulator</i> [2].....	53
4	BAB 4 PENGUJIAN DAN ANALISA.....	59
4.1	Pengujian Beban Nominal	59
4.2	Pengujian dengan Perubahan Beban.....	67
4.3	Pengujian dengan Variasi Set Point.....	73
5	BAB 5 KESIMPULAN DAN SARAN	79
5.1	Kesimpulan.....	79

5.2	Saran.....	79
DAFTAR PUSTAKA	81
LAMPIRAN	83
RIWAYAT HIDUP	101

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Presure Process Rig 38-714	6
Gambar 2.2 Bagian-bagian pressure process rig 38-714	7
Gambar 2.3 <i>Compressor</i> pada <i>Pressure Process Rig</i> 38-714.....	8
Gambar 2.4 Prinsip kerja I/P converter	9
Gambar 2.5 I/P converter.....	9
Gambar 2.6 <i>Control valve</i>	10
Gambar 2.7 <i>Control valve Pressure Process Rig</i> 38-714	11
Gambar 2.8 <i>Pressure sensor</i>	11
Gambar 2.9 Prinsip kerja sensor tekanan.....	12
Gambar 2.10 Prinsip kerja <i>differential pressure sensor</i>	13
Gambar 2.11 <i>Differential pressure sensor Pressure Process Rig</i> 38-714	13
Gambar 2.12 <i>Air receiver Pressure Process Rig</i> 38-714.....	14
Gambar 2.13 <i>Manual valve</i>	14
Gambar 2.14 <i>Gauge Pressure Process Rig</i> 38-714	15
Gambar 2.15 Regulator <i>Pressure Process Rig</i> 38-714	16
Gambar 2.16 Prinsip kerja <i>flowmeter</i>	17
Gambar 2.17 Diagram fisik peralatan.....	18
Gambar 2.18 <i>Blok diagram</i> sistem pengaturan tekanan <i>Pressure Process Rig</i> 38-714	18
Gambar 2.19 Diagram blok <i>self tuning regulator</i>	19
Gambar 2.20 Sebuah ADC dapat terhubung dengan bus pada komputer bila memiliki <i>tri-state output. Address decoding</i> dibutuhkan supaya ADC dapat dioperasikan melalui perangkat lunak komputer. [8].....	21
Gambar 2.21 Rangkaian khusus dari <i>data-acquisition board</i> untuk digunakan pada <i>slot ekspansi PC</i> [8]	22
Gambar 2.22 Sebuah multiplexer analog bekerja sebagai sebuah <i>switch</i> multiposisi untuk memilih input ke ADC [8]	23
Gambar 2.23 Sekuen dari operasi DAS <i>software</i> [8]	24
Gambar 2.24 Jaringan syaraf biologi [9]	25
Gambar 2.25 Model jaringan syaraf tiruan [9]	26
Gambar 2.26 <i>Feedforward</i> dengan neuron <i>single-layer</i> [9].....	27
Gambar 2.27 Jaringan <i>multilayer</i> dengan satu <i>hidden-layer</i> [9].....	28
Gambar 2.28 Jaringan <i>recurrent</i> tanpa neuron tersembunyi [9]	29
Gambar 2.29 Fungsi <i>threshold</i> [9]	30

Gambar 2.30 Fungsi sigmoid [9]	30
Gambar 2.31 Contoh <i>front panel</i> [10]	35
Gambar 2.32 Contoh sebuah <i>block diagram</i> dan <i>front panel</i> yang terhubung [10].....	36
Gambar 3.1 Arsitektur sistem pengaturan tekanan <i>Process Rig</i> 38-741	37
Gambar 3.2 P&ID <i>Pressure Process Rig</i> 38-741.....	39
Gambar 3.3 Pengalamatan kanal I/O ADAM-5000/TCP [11]	41
Gambar 3.4 <i>Software ADAM-5000TCP-6000 Utility</i>	41
Gambar 3.5 Rangkaian I/V converter	42
Gambar 3.6 Skema konversi sinyal arus ke tegangan	42
Gambar 3.7 Grafik fungsi tegangan terhadap arus desain <i>converter I/V</i>	43
Gambar 3.8 Skema identifikasi sistem	43
Gambar 3.9 Grafik respon <i>open loop Pressure Process Rig</i> 38-714.....	44
Gambar 3.10 Perbandingan respon <i>open loop plant</i> dengan model yang didapat.....	45
Gambar 3.11 Bentuk umum arsitektur <i>artificial neural network</i> yang digunakan.....	50
Gambar 3.12 Arsitektur <i>artificial neural network</i> yang dipakai	52
Gambar 3.13 Kontroler PI.....	55
Gambar 4.1 Respon sistem kondisi beban nominal	60
Gambar 4.2 Respon nilai Kp STR RLS pada kondisi beban nominal....	60
Gambar 4.3 Respon nilai Kp STR NN pada kondisi beban nominal	61
Gambar 4.4 Respon nilai Kp pada kondisi beban nominal	61
Gambar 4.5 Respon Ti STR RLS pada kondisi beban nominal	62
Gambar 4.6 Respon nilai Ti STR NN pada kondisi beban nominal.....	62
Gambar 4.7 Respon nilai Ti pada kondisi beban nominal.....	63
Gambar 4.8 Respon sistem kondisi adanya perubahan beban.....	67
Gambar 4.9 Respon nilai Kp STR RLS pada kondisi perubahan beban	68
Gambar 4.10 Respon Kp STR NN pada kondisi perubahan beban.....	68
Gambar 4.11 Respon nilai Kp pada kondisi perubahan beban	69
Gambar 4.12 Respon nilai Ti STR RLS pada kondisi perubahan beban	69
Gambar 4.13 Respon Ti STR NN pada kondisi perubahan beban	70
Gambar 4.14 Respon nilaiTi pada kondisi perubahan beban	70
Gambar 4.15 Respon sistem kondisi variasi <i>set point</i>	74
Gambar 4.16 Respon nilai Kp STR RLS pada kondisi variasi set point	74

Gambar 4.17 Respon nilai Kp STR NN pada kondisi variasi set point	74
Gambar 4.18 Respon nilai Kp pada kondisi variasi set point	75
Gambar 4.19 Respon nilai Ti STR RLS pada kondisi variasi set point	75
Gambar 4.20 Respon nilai Ti STR NN pada kondisi variasi set point...	76
Gambar 4.21 Respon Nilai Ti pada kondisi variasi set point.....	76

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 ADAM-5000 AI/AO <i>scalling</i> [11]	40
Tabel 4.1 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal.....	63
Tabel 4.2 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal	63
Tabel 4.3 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal dengan normalisasi.....	65
Tabel 4.4 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal dengan normalisasi	66
Tabel 4.5 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal dengan pembobotan	66
Tabel 4.6 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal dengan pembobotan	66
Tabel 4.7 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursice least square</i> pada kondisi adanya perubahan beban	70
Tabel 4.8 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban	71
Tabel 4.9 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursice least square</i> pada kondisi adanya perubahan beban dengan normalisasi	72
Tabel 4.10 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban dengan normalisasi	72
Tabel 4.11 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursice least square</i> pada kondisi adanya perubahan beban dengan pembobotan	72

Tabel 4.12 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban dengan pembobotan	73
Tabel 4.13 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal terhadap variasi <i>set point</i>	76
Tabel 4.14 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi <i>set point</i>	77
Tabel 4.15 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal terhadap variasi <i>set point</i> dengan normalisasi.....	77
Tabel 4.16 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi <i>set point</i> dengan normalisasi	77
Tabel 4.17 Performa sistem hasil pengujian STR dengan estimasi parameter <i>recursive least square</i> pada kondisi beban nominal terhadap variasi <i>set point</i> dengan pembobotan	78
Tabel 4.18 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi <i>set point</i> dengan pembobotan.....	78

[Halaman ini sengaja dikosongkan]

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Bab ini akan memberikan pengenalan pembaca tentang apa yang diteliti, untuk apa suatu penelitian dilakukan. Hal tersebut tersebut akan diuraikan pada bab ini yang meliputi latar belakang, perumusan masalah, tujuan, metodologi, sistematika dan relevansi.

1.1 Latar Belakang

Globalisasi membawa perubahan yang cepat dan memberikan dampak pada ekonomi nasional dan internasional. Dampak jelasnya adalah semakin ketatnya persaingan ekonomi, terutama pada bidang industry [1]. Persaingan industri yang semakin ketat menuntut industri agar meningkatkan produksi dan menurunkan biaya produksi tetapi tetap menjaga kualitas atau bahkan meningkatkan kualitasnya. Hal tersebut menimbulkan permasalahan baru di dunia industri yaitu penanganan yang lebih baik untuk meningkatkan kualitas namun dengan biaya produksi yang dapat diturunkan. Perubahan yang cepat di bidang ilmu pengetahuan telah menjadi sebuah faktor penentu terbentuknya persaingan dan modernisasi industri. Ilmu pengetahuan dan teknologi yang tepat diperlukan untuk bertahan atau bahkan memimpin dalam persaingan industri tersebut.

Sistem pengaturan proses tak bisa dipisahkan dari kata industri karena sistem pengaturan proses memiliki peranan cukup banyak pada bidang industri. Mulai dari industri bidang energi, makanan dan minuman serta migas. Pada sistem pengaturan proses terdapat beberapa variable proses yang biasanya diatur yaitu, temperatur, volume atau ketinggian, dan aliran serta tekanan. Variabel tersebut biasanya saling terhubung satu sama lain sehingga perubahan salah satu variabel akan mempengaruhi variabel yang lain juga. Pada sistem pengaturan proses sering terjadi proses perubahan beban. Perubahan tersebut dapat menimbulkan perubahan dinamika dari sistem.

Pengaturan adaptif adalah salah salah metode pengaturan yang mana kontroler dapat memberikan respon memodifikasi perilakunya karena perubahan dinamika dari proses dan karakteristik dari gangguan [2]. Pengaturan adaptif adalah sebuah kontroler dengan parameter yang dapat diatur dan sebuah mekanisme pengaturan parameter. Pengaturan adaptif dapat digunakan pada *plant* yang memiliki variasi pada dinamika proses. Variasi dinamika proses dapat disebabkan oleh beberapa faktor yaitu

aktuator yang nonlinier, variasi dari *flow* dan kecepatan, variasi di dalam karakteristik gangguan, dan juga variasi di dalam karakteristik beban serta banyak faktor variasi lain dan juga biasanya campuran dari beberapa fenomena yang berbeda. Dengan demikian pengaturan adaptif dapat digunakan digunakan pada *plant* yang mengalami variasi karakteristik beban untuk mencapai spesifikasi yang lebih baik.

1.2 Perumusan Masalah

A Sebagian proses yang ada di industri memiliki sifat dasar yang nonlinier, dimana saat operasi sifat atau dinamikanya berubah terhadap keadaan beban dan waktu [3]. Pada sistem pengaturan proses masalah yang sering terjadi adalah terjadinya perubahan beban atau dinamika proses yang berubah karena beberapa faktor contohnya munculnya gangguan atau usia peralatan. Perubahan beban menimbulkan perubahan karakteristik dari *plant*. Munculnya perubahan tersebut akan memberikan kendala yang harus diperhatikan bila ingin mencapai performansi yang baik serta desain kontroler yang sederhana kurang cukup untuk mengatasinya. *Self-tunning regulator* (STR) adalah salah satu jenis pengaturan adaptif. STR dapat melakukan penalaan secara otomatis ketika terjadi perubahan dinamika dari *plant*. Salah satu bagian dari STR adalah estimasi parameter model *plant*. Untuk mencapai peforma STR yang baik perlu didesain estimasi parameter yang tepat.

1.3 Batasan Masalah

Pada tugas akhir ini fokus terhadap analisa perbandingan metode estimasi parameter pada sistem pengaturan tekanan di dalam kontroler *self-tunning regulator* yang mengalami perubahan karakteristik beban. *Plant* yang digunakan adalah pada sistem PROCON *Process Rig* 38-714 produksi Feedback Inc. Kriteria minimum *root mean square error* (RMSE), *integral time square error* (ITSE), *integral time absolut error* (ITAE) serta *integral square time multiplied by square error* (ISTSE) dipakai sebagai kriteria pembanding antar metode estimasi parameter yang dipakai. Metode estimasi parameter yang akan diuji adalah *recursive least square* (RLS) dan pendekatan jaringan syaraf tiruan. Implementasi perancangan metode kontrol dilakukan di *software* LabVIEW produksi National Instrument.

1.4 Tujuan

Tujuan dari penelitian ini adalah analisis perbandingan estimasi parameter di STR pada kondisi beban nominal, adanya perubahan beban

dan variasi *set point* untuk PROCON *Pressure Process Rig* 38-714 sehingga dapat diketahui karakteristiknya..

1.5 Metodologi

Penelitian dilakukan dengan beberapa tahapan yaitu, studi literatur, identifikasi sensor dan aktuator pada PROCON *Process Rig* 38-714, identifikasi sistem, perancangan kontroler, simulasi, implementasi, analisa hasil implementasi dan simulasi, serta penyusunan buku tugas akhir. Pada tahap studi literatur ditempuh dengan cara mempelajari beberapa referensi dari buku, *paper* dan jurnal mengenai topik yang akan dibahas hingga sistem yang nantinya akan dibahas dapat dipahami secara detail dan jelas. Materi yang dipelajari berupa sistem pengaturan tekanan, *self-tunning regulator*, identifikasi sistem dan *artificial neural network*. Identifikasi sensor dan aktuator ditempuh melalui pengecekan komponen-komponen dari plant untuk mengetahui karakteristik masing-masing komponen. Pada tahap identifikasi sistem dilakukan pengambilan data sistem *open loop* untuk menentukan struktur dari model dan parameter-parameter yang dibutuhkan. Setelah itu dilakukan perancangan metode kontrol dari informasi yang didapat dari studi literatur dan identifikasi sistem. Setelah perancangan metode kontrol berhasil ditempuh baru dilakukan simulasi untuk mengetahui karakteristik respon sistem dan mengetahui apakah sudah aman untuk diimplementasikan pada *plant*. Ketika simulasi hasil perancangan telah mendapatkan hasil yang sesuai kriteria yang telah ditetapkan dan dirasa aman untuk diimplementasikan pada *plant* baru dapat dilakukan implementasi hasil perancangan. Data yang dikumpulkan dari hasil simulasi dan implementasi dianalisa untuk mengetahui performa dari sistem setelah dilakukan perancangan.

1.6 Sistematika

Penulis membagi laporan penelitian ini menjadi lima bab. Hal ini dilakukan untuk mempermudah pembaca guna memahami isi laporan dan menghindari kesulahan penafsiran terhadap laporan yang disusun pada buku tugas akhir ini. Penulisan laporan penelitian ini dapat dijelaskan dengan sistematika sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, permasalahan, batasan masalah, tujuan penelitian, metodologi, sistematika penulisan dan relevansi dari penelitian yang dilakukan.

BAB 2 TINJAUAN PUSTAKA

Bab ini mengulas tentang konsep dan teori serta informasi yang menunjang penelitian ini diantaranya tentang sistem pengaturan tekanan, Process Rig 38-741, kontroler *self-tunning regulator*, estimasi parameter sistem, dan *artificial neural network*.

BAB 3 PERANCANGAN SISTEM

Pada bagian ini dijelaskan mengenai sistem yang akan dirancang, proses identifikasi sistem, formulasi estimasi parameter dan desain kontroler *self-tunning PI*.

BAB 4 PENGUJIAN DAN ANALISIS

Bab ini membahas tentang hasil implementasi sistem menggunakan *Self-Tunning Regulator*, analisis performa sistem terhadap perubahan beban, dan variasi *set point* serta pada keadaan nominal sistem.

BAB 5 KESIMPULAN

Bab 5 membahas tentang kesimpulan dan saran berkaitan dengan penelitian yang telah dilakukan.

1.7 Relevansi

Pengaturan tekanan adalah hal yang umum di industri. Penerapan metode kontrol *self-tunning PI* dapat diaplikasikan untuk menjaga tekanan dari sistem meskipun terjadi perubahan beban atau perubahan dari dinamika sistem akibat adanya gangguan. Hasil yang diperoleh pada penelitian ini bisa menjadi referensi untuk pembelajaran pengaturan tekanan di bidang pendidikan maupun di industri.

BAB 2

TINJAUAN PUSTAKA

Pemahaman teori sistem yang akan dibangun nantinya diperlukan sebagai landasan maupun dasar bagi penulis dalam melakukan sebuah penelitian. Teori-teori tersebut dikaji terlebih dahulu untuk menunjang serta memperkuat penelitian penulis. Bab ini mengulas teori yang mencakup *plant*, komponen-komponen sistem, kontroler dan metode yang digunakan untuk mengatasi masalah pada tugas akhir ini.

2.1 Sistem Pengaturan Proses

Sistem pengaturan proses adalah cabang dari sistem pengaturan yang berurusan dengan operasi *plant* di industri seperti petrokimia, makanan, besi, kaca, kertas dan energi. Biasanya sistem pengaturan proses yang diterapkan pada industri skala besar menangani hal-hal yang secara langsung mempengaruhi produksi produk jadi. Hal tersebut melibatkan manipulasi dari variabel proses yang bermacam-macam untuk memenuhi sistem pengaturan yang otomatis. Jumlah dari variabel tergantung pada produk yang akan diproduksi. Biasanya lebih dari satu variabel yang dikontrol. Beberapa variabel dikontrol biasanya berubah terhadap waktu yaitu, tekanan, temperature, *flow*, *level*, dan massa. Ketika salah satu variabel dirubah seringkali akan mempengaruhi variabel yang lain [4].

Pengaturan proses menjadi area yang sangat penting di industri, karena hal itu menentukan seberapa baik sebuah *plant* beroperasi. Dengan jenis pengaturan yang tepat, sebuah *plant* seharusnya beroperasi secara efisien dan optimal sesuai dengan keadaan yang diperbolehkan.

Dalam waktu singkat, industri telah membuat transisi dalam hal produksi: dari teknik produksi yang sebagian besar beroperasi secara manual sampai pada prosedur otomatis yang membutuhkan sedikit usaha manusia. Melalui transisi ini ada sebuah perubahan yang drastis untuk menghasilkan kualitas produk yang lebih baik dan produksi yang lebih ekonomis.

2.2 Sistem Pengaturan Tekanan

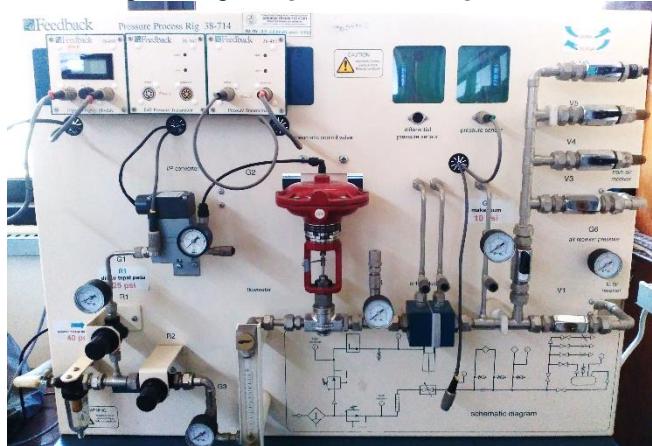
Hampir semua produk yang diproduksi di industri adalah hasil proses yang melibatkan tekanan. Sistem yang merespon ke perubahan terhadap tekanan adalah salah satu divisi terbesar sistem pengaturan proses kini. Sistem tekanan didesain untuk melakukan sebuah fungsi kerja khusus yang menghasilkan tekanan yang ditransfer melalui sebuah jaringan pipa atau tabung, cairan, gas, uap, dan udara secara umum digunakan di

industri untuk mengatur laju dari energi atau daya. Secara spesifik, proses ini melibatkan peralatan-peralatan seperti hidrolik, pneumatik, sistem uap atau penguapan, dan distribusi cairan. [5].

2.3 Pressure Process Rig 38-714 [6]

Keberhasilan desain dan operasi dari *plant* di industri memerlukan pilihan yang optimal dari peralatan instrumentasi dan kontrol bersama dengan kemampuan untuk mempersiapkan, memantau, dan menjaga peralatan seperti itu. Teknisi dan insinyur butuh pelatihan pada kemampuan-kemampuan ini secara khusus, karena cepatnya perkembangan penggunaan elektronika dan mikrokomputer di dalam pengaturan proses.

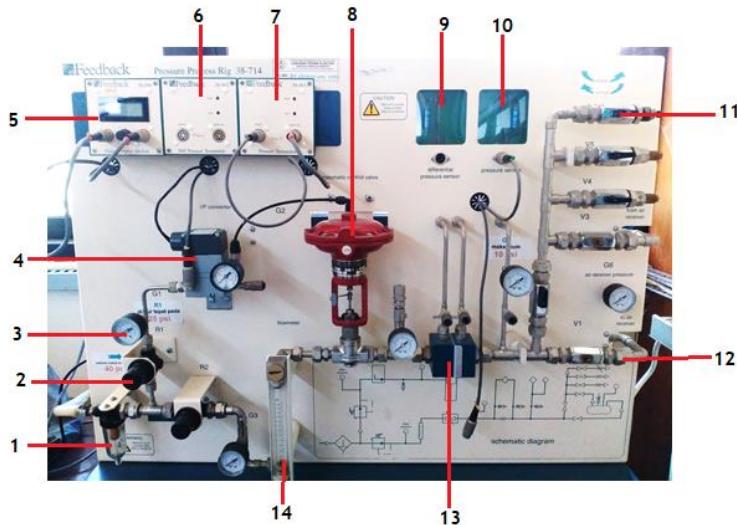
Pressure Process Rig 38-714 merupakan peralatan laboratorium yang berfungsi untuk memperkenalkan dan menunjukkan prinsip dari proses pengukuran dan kontrol tekanan. Pada unit 38-741 terdiri dari *pneumatic control valve*, *orifice block*, *flowmeter* dan *pressure tapping*. *Penumatic control valve* beroperasi melalui *I/P converter*, dan sinyal sensor akan masuk pada bagian *signal conditioning*.



Gambar 2.1 Presuure Process Rig 38-714

2.3.1 Komponen-Komponen Pressure Process Rig 38-714

Bagian-bagian pada pressure process rig 38-741 ditampilkan pada Gambar 2.2.



Gambar 2.2 Bagian-bagian pressure process rig 38-714

Berikut merupakan nama bagian-bagian dari pressure process rig 38-741:

1. Filter
2. *Regulator*
3. *Gauge*
4. I/P converter
5. Modul display
6. Differential pressure transmitter
7. Pressure transmitter
8. Pneumatic control valve
9. Differential pressure sensor
10. Pressure sensor
11. Manual valve
12. Air receiver
13. Orifice block
14. Flowmeter

2.3.1.1 Compressor

Compresor adalah suatu alat yang berfungsi untuk menaikkan tekanan suatu gas dengan cara menurunkan volumenya. Sesuai dengan hukum Boyle yaitu “Apabila suhu gas yang berada dalam ruang tertutup dijaga konstan, maka P (tekanan) berbanding terbalik dengan V (volume).

$$P_1 V_1 = P_2 V_2 \quad (2.1)$$

Alat ini dirancang agar dapat menyimpan dan mendistribusikan suatu gas bertekanan. Gambar 2.3 menunjukkan *compressor* yang digunakan pada



Pressure Process Rig 38-714.

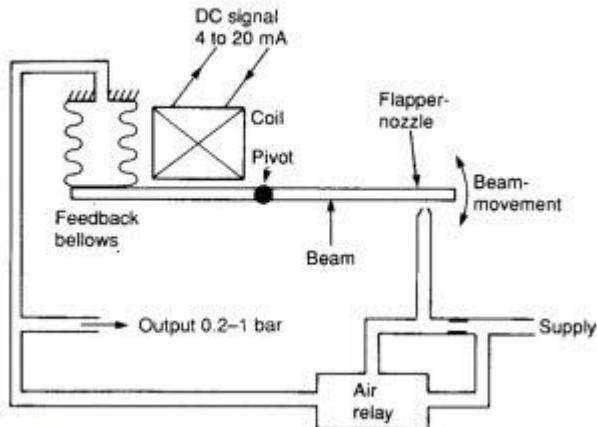
2.3.1.2 Filter

Filter digunakan untuk memisahkan antara udara dengan air, agar udara yang digunakan pada proses merupakan udara yang kering dan bersih. Filter udara dirancang untuk menyaring kotoran dengan ukuran tertentu sehingga pori-pori kertas *filter* harus diganti secara periodik

2.3.1.3 I/P Converter

I/P converter adalah sebuah alat instrumentasi yang berfungsi untuk mengubah suatu bentuk sinyal arus listrik (4-20mA) menjadi sinyal tekanan pneumatik (3-15 psi). Prinsip kerja instrumen ini menggunakan kumparan yang digerakkan secara listrik dengan diberi sinyal input arus listrik sebesar 4-20 mA DC ke posisi *asembli pneumatic flapper nozzle*. Posisi *flapper* yang berhubungan dengan nozzle akan menambah atau menurunkan tekanan *output*. Jika *flapper* lebih dekat ke nozzle, maka

tekanan output bertambah. Jika flapper lebih jauh dari nozzle, maka tekanan outputnya juga akan berkurang.



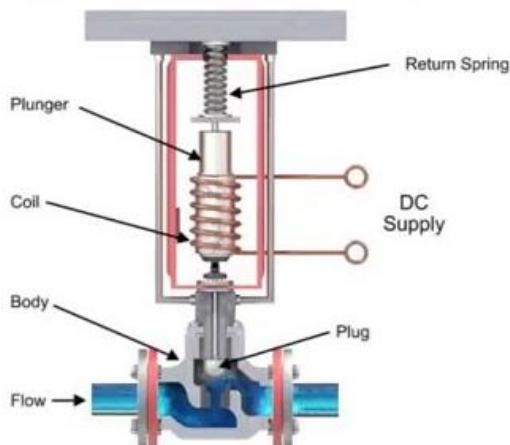
Gambar 2.4 Prinsip kerja I/P converter



Gambar 2.5 I/P converter

2.3.1.4 Pneumatic Control Valve

Control valve adalah rangkaian *valve* dan aktuator yang berfungsi sebagai pengontrol aliran atau tekanan fluida. *Control valve* menerima masukan udara bertekanan 3-15 psi yang berasal dari I/P converter. Besar tekanan yang masuk mempengaruhi besar bukaan *valve*.



Gambar 2.6 Control valve

Gambar 2.6 menampilkan bagian-bagian dari *control valve*. Pada badan *valve* terdapat lubang yang berfungsi untuk mengalirnya udara. *Plug* akan naik atau turun sesuai dengan besar udara yang mengalir atau berdasarkan arus yang diberikan pada kumparan. Sementara pada aktuator berfungsi menggerakkan batang *valve/stem* untuk membuka atau menutup aliran dengan menggunakan sinyal I/P 4-20 mA yang telah diubah menjadi sinyal 3-15 psi. Pada saat tidak ada tekanan udara yang mengalir atau pada saat keadaan 3 psi, *control valve* berada pada keadaan *normally open*. Sedangkan pada saat tekanan yang diberikan yaitu 15 psi maka *control valve* berada pada keadaan *normally close*.

Aliran listrik yang diberikan pada *coil* akan mempengaruhi *spring*. Pada saat aliran listrik yang diberikan adalah 4 mA maka *spring* akan naik ke atas sehingga menyebabkan *valve* berada pada posisi *normally open*.



Gambar 2.7 Control valve Pressure Process Rig 38-714

2.3.1.5 *Display modul*

Modul yang digunakan untuk menampilkan hasil pengukuran *pressure* dan *differential pressure transmitter*. Pada display modul, data akan berubah menjadi data digital. Terdapat rangkaian ADC (Analog to Digital Converter) agar data dapat ditampilkan pada layar display. Data analog akan diubah kedalam data biner sehingga pada layar display akan berubah menjadi angka dalam tujuh digit. Modul LCD yang digunakan adalah 16x2 dengan antar muka parallel 4 bit.

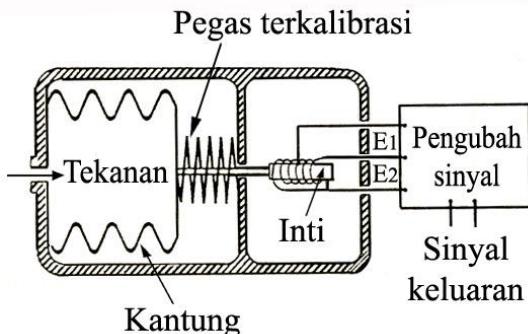
2.3.1.6 *Pressure Sensor*

Sensor tekanan adalah sensor untuk mengukur tekanan suatu zat.



Gambar 2.8 Pressure sensor

Prinsip kerja sensor tekanan yaitu perubahan tekanan pada kantung menyebabkan perubahan posisi inti kumparan sehingga mengakibatkan perubahan induksi magnetic pada kumparan. Kumparan yang digunakan adalah kumparan CT (*Center Tap*), dengan demikian apabila inti mengalami pergeseran maka induktifitas salah satu kumparan bertambah sementara induktansi pada kumparan lain berkurang. Kemudian pengubah sinyal berfungsi untuk mengubah induktansi magnetik yang timbul pada kumparan menjadi tegangan yang sebanding.

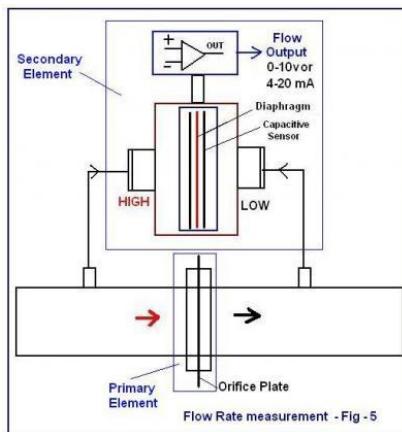


Gambar 2.9 Prinsip kerja sensor tekanan

2.3.1.7 Differential Pressure Sensor

Dua sensor tekanan yang dipasang berdampingan pada jarak tertentu (relativ dekat) dalam satu saluran yang diberi sekat dengan *orifice plate*. Perbedaan tekanan yang terukur adalah informasi untuk mengetahui aliran udara.

Pada differential *pressure sensor* terdapat *orifice plate* dimana udara akan mengalir pada lubang yang terdapat pada *orifice plate* sehingga terdapat perbedaan tekanan aliran udara. Perbedaan tekanan udara tersebut akan menggeser diafragma. Apabila diafragma mendekati *capacitive sensor low* maka *differential pressure* yang terukur adalah kecil dan sebaliknya. Prinsip kerjanya sama dengan hukum Bernauli.



Gambar 2.10 Prinsip kerja *differential pressure sensor*



Gambar 2.11 *Differential pressure sensor Pressure Process Rig 38-714*

2.3.1.8 Pressure Transmitter

Pressure transmitter adalah peralatan yang digunakan dalam pengukuran tekanan. Alat ini memiliki *range* pengukuran tekanan antara 3-15 psi atau sekitar 4-20 mA. Alat ini akan mengirimkan data dari *pressure sensor* untuk dikirimkan menuju *display modul* hingga akhirnya data akan ditampilkan pada komputer.

2.3.1.9 Air Receiver

Air receiver merupakan wadah penampung udara yang berfungsi untuk menampung dan mendistribusikan udara yang didapatkan dari susunan buka tutup katup manual. Jika penampung udara dipergunakan (katup 1 dan katup 3 buka), maka respon tekanan akan menjadi lebih lambat.



Gambar 2.12 Air receiver Pressure Process Rig 38-714

Untuk mendesain air receiver digunakan persamaan berikut:

$$V = \frac{tQ \times Pa}{(P_1 - P_2)} \quad (2.2)$$

Dimana:

t = waktu

Q = Aliran udara

P_1 = Maksimum operating pressure

P_2 = Minimum operating pressure

V = Volume yang diterima

Pa = Tekanan atmosfer (14,7 psi)

2.3.1.10 Manual Valve

Manual valve digunakan untuk membuka dan menutupnya beban yang harus dilakukan secara manual dengan cara diputar tuasnya.



Gambar 2.13 Manual valve

2.3.1.11 Gauge

Gauge digunakan sebagai indikator tekanan yang sedang terjadi. Prinsip kerja dari gauge adalah terdapat tabung bourdon yang berbentuk C atau spiral atau berbentuk *coil*. Saat terdapat tekanan, tekanan menyebabkan tabung bourdon mengembangkan dirinya yang menyebabkan tip bergerak. Pergerakan dari tip diubah menyebabkan jarum bergerak secara rotasi untuk mengindikasikan tekanan yang terukur.



Gambar 2.14 Gauge Pressure Process Rig 38-714

2.3.1.12 Orifice Block

Orifice adalah salah satu alat yang digunakan untuk mengukur laju aliran volume atau massa fluida di dalam saluran yang tertutup (pipa) berdasarkan prinsip beda tekanan. Alat ini berupa plat tipis dengan gagang yang diapit diantara flens pipa. *Orifice block* bekerja menggunakan prinsip dari hukum Bernauli.

$$P + \rho gh + \frac{1}{2} \rho v^2 = \text{konstan} \quad (2.3)$$

P = Tekanan (Pascal)

ρ = massa jenis fluida (kg/m^3)

g = percepatan gravitasi ($9,8 \text{ m/s}^2$)

h = ketinggian (m)

v = kecepatan (m/s)

Hukum Bernauli menyatakan bahwa tekanan dari fluida yang bergerak seperti udara berkurang ketika fluida tersebut bergerak lebih cepat.

2.3.1.13 *Regulator*

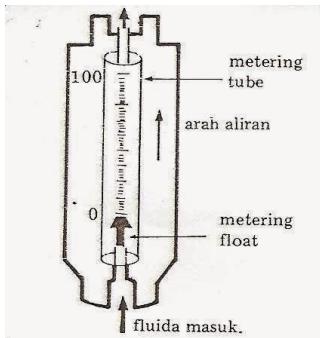
Pengatur manual bukaan saluran udara untuk membatasi tekanan udara yang melewati saluran tertentu. *Pressure regulator* merupakan pengatur manual katup saluran udara untuk membatasi tekanan udara yang melewati saluran tersebut. Dalam sistem ini, terdapat dua *pressure regulator*. Regulator pertama diatur sebesar 25 psi untuk menggerakkan *pneumatic control valve* sedangkan regulator kedua sebesar 10 psi untuk keperluan pengaturan tekanan dan kecepatan alir.



Gambar 2.15 Regulator *Pressure Process Rig 38-714*

2.3.1.14 *Flowmeter*

Flowmeter adalah alat ukur yang digunakan untuk mengukur laju aliran atau jumlah fluida yang bergerak mengalir dalam suatu pipa tertutup. Jenis flowmeter yang digunakan adalah *tube glass flowmeter*. *Tube glass flowmeter* memiliki kemampuan untuk mengukur tekanan paling besar 10 bar

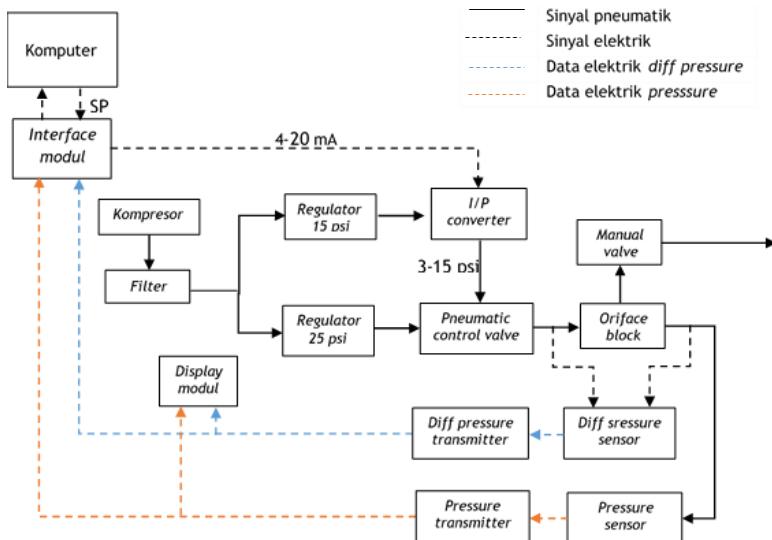


Gambar 2.16 Prinsip kerja flowmeter

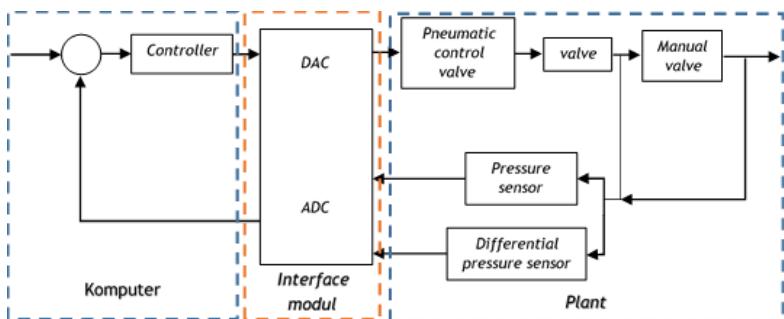
Fluida mengalir ke atas melalui tabung gelas berisi float yang dapat bergerak dengan bebas. Untuk menunjukkan besarnya aliran fluida *metering float* naik ke atas, seperti terlihat pada Gambar 2.16. Kepala metering float menunjukkan angka 0, membuktikan bahwa valve yang terletak di bawah meteran ini belum terbuka. Rotameter bila dipasang pada pipa-pipa berukuran $\frac{1}{4}$ - 3 inci standar. Pembacaan tinggi *float* pada *scale line* sebanding dengan perubahan besarnya aliran yang terjadi .

2.3.2 Cara Kerja Peralatan pada *Pressure Process Rig* 38-714

Compresor bekerja untuk menghasilkan udara bertekanan, udara bertekanan yang masuk ke *plant* dibagi menjadi dua yaitu untuk kontrol katup *pneumatik* (diatur oleh *regulator* tekanan 1) dan untuk proses (diatur oleh *regulator* tekanan 2). Udara pada kontrol katup *pneumatik* digunakan sebagai tenaga untuk menggerakan kontrol katup *pneumatik* melalui proses pengubah arus ke *pneumatik* (*I/P*). Sedangkan, udara pada proses dikendalikan/diatur tekanannya oleh kontrol katup *pneumatik*. Udara dapat dibuang langsung ke tekanan atmosfer atau ditampung ke penampung udara terlebih dahulu, penggunaan penampung udara dapat disusun secara seri, pararel atau tidak digunakan, hal ini tergantung dari susunan katup manual.



Gambar 2.17 Diagram fisik peralatan



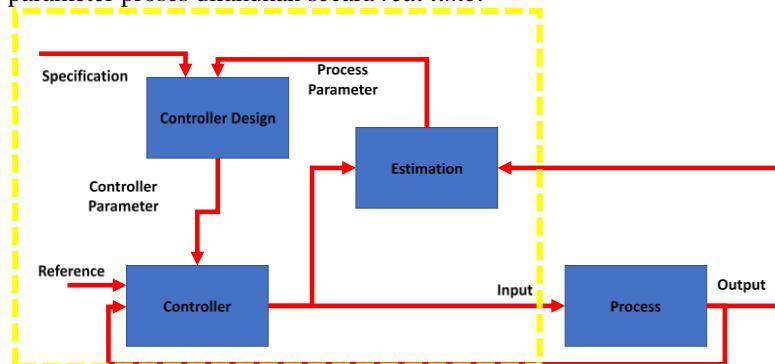
Gambar 2.18 Blok diagram sistem pengaturan tekanan Pressure Process Rig 38-714

2.4 Self-Tunning Regulator (STR) [2]

Pengembangan sebuah sistem pengaturan melibatkan banyak kegiatan seperti pemodelan, desain kontroler, implementasi, dan validasi. *Self-tunning regulator* bertujuan mengotomasi beberapa kegiatan tersebut. Alasan utama untuk menggunakan *self-tunning regulator* adalah proses atau lingkungan berubah secara kontinyu. Hal tersebut

memberikan kendala tambahan dalam menganalisa sistem tersebut. Untuk mengatasi permasalahan tersebut dapat diasumsikan bahwa proses konstan tetapi terdapat parameter yang tidak diketahui.

STR merupakan salah satu dari skema sistem pengaturan adaptif. Secara umum kontroler STR dapat dipandang memiliki dua loop. Loop dalam mengandung proses dan kontroler umpan balik biasa. Parameter dari kontroler diatur oleh *loop luar* yang terdiri dari estimasi parameter secara rekursif dan mekanisme *tuning* parameter. Diagram blok dari *self-tunning regulator* diilustrasikan pada Gambar 2.19. Pada STR estimasi parameter proses dilakukan secara *real time*.



Gambar 2.19 Diagram blok *self tuning regulator*

2.5 Kontroler PID [7]

Kontroler PID atau modifikasinya adalah salah satu kontroler yang banyak digunakan di industri. Alasan utamanya yaitu memiliki struktur yang sederhana mudah diaplikasikan. Algoritma kontroler PID dapat dideskripsikan dengan persamaan:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.4)$$

dimana u adalah sinyal kontrol dan e adalah sinyal eror ($e = y_{sp} - y$). Sinyal kontrol merupakan jumlahan dari tiga aksi yaitu: P (yang mana proporsional terhadap eror), I (yang mana proporsional terhadap integral dari eror), dan D (yang mana proporsional terhadap derivative dari eror). Parameter kontroler adalah gain proporsional (K_p), waktu integral (T_i), dan waktu derivatif (T_d).

2.5.1 Aksi Proporsional

Pada kasus kontrol proporsional murni, persamaan kontroler dapat dideskripsikan dengan persamaan

$$u(t) = K_p e(t) + u_b \quad (2.5)$$

Aksi kontrol secara sederhana proporsional terhadap eror. Variabel u_b adalah bias atau reset. Ketika sinyal eror adalah nol variabel kontrol mengambil nilai $u(t) = u_b$. Bias seringkali ditetapkan dengan $(u_{max} - u_{min})/2$, tetapi kadang-kadang dapat diatur secara manual sehingga sinyal kontrol adalah nol pada *set point* yang diberikan. Semakin besar nilai K_p dapat menyebabkan peningkatan kecepatan respon dan peningkatan *overshot* pada sistem.

2.5.2 Aksi Integral

Fungsi utama dari aksi integral adalah untuk memastikan keluaran dari proses sesuai dengan *set point* pada saat *steady state*. Dengan control proporsional, secara normal ada eror pada saat *steady state*. Dengan aksi integral, sebuah eror positif yang kecil akan selalu menuntun kepada sebuah aksi peningkatan sinyal kontrol, dan sebuah eror negatif akan memberi penurunan sinyal kontrol tidak peduli seberapa pun kecil yang ada. Penyataan sederhana berikut menunjukkan bahwa eror *steady state* akan selalu menjadi nol dengan aksi integral namun aksi integral juga dapat memperlambat sistem.

τ_i merupakan komponen dalam aksi integral yang menyatakan integral time atau waktu integral. Aksi integral dapat dijelaskan pada Persamaan (2.6).

$$u(t) = \frac{1}{\tau_i} \int_0^t e(t) dt \quad (2.6)$$

Persamaan (2.6) menunjukkan bahwa aksi integral nilainya proporsional terhadap integral dari eror.

2.5.3 Aksi Derivatif

Tujuan dari aksi derivative adalah untuk meningkatkan stabilitas *closed-loop*. Mekanisme ketidakstabilan dapat dideskripsikan sebagai berikut. Karena dinamika proses, hal ini akan memerlukan beberapa waktu sebelum sebuah perubahan pada variabel kontrol tampak di keluaran proses. Dengan demikian, sistem pengaturan akan terlambat di

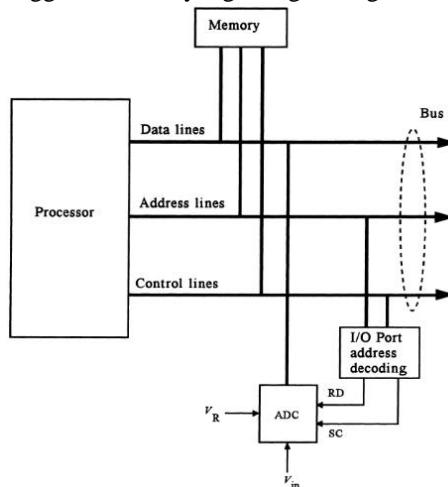
dalam mengkoreksi sebuah eror. Aksi derivative dapat diformulasikan pada Persamaan (2.7). Aksi derivatif memberikan aksi yang nilainya

$$u(t) = T_d \frac{de(t)}{dt} \quad (2.7)$$

proporsional terhadap derivative dari eror. Aksi derivatif tidak dapat berdiri sendiri karena kontroler ini tidak akan menghasilkan aksi bila terjadi kesalahan yang konstan sehingga harus dirancang bersama aksi yang lain, bisa berpasangan dengan aksi P atau I.

2.6 Data-Acquisition System (DAS) [8]

Mikroprosesor berbasis personal computer (PC) telah digunakan secara luas untuk mengimplementasikan *direct digital control*. Komputer desktop yang familiar didesain seperti sistem yang ditampilkan pada Gambar 2.20 , menggunakan bus yang mengandung *data line*, *address*



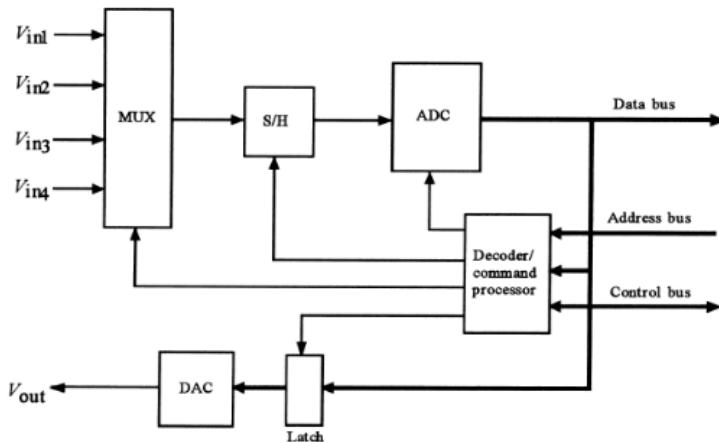
Gambar 2.20 Sebuah ADC dapat terhubung dengan bus pada komputer bila memiliki *tri-state output*. *Address decoding* dibutuhkan supaya ADC dapat dioperasikan melalui perangkat lunak komputer. [8]

-line, dan control line. Semua komunikasi dengan prosesor melalui saluran bus ini. Personal komputer juga menghubungkan saluran bus pada sejumlah *socket printed circuit board* (PCB). *Socket* ini disebut dengan slot ekspansi. Banyak jenis khusus dari peralatan *peripheral* seperti fax, *board* modem, dan *board* koneksi jaringan yang didesain pada PCB dapat ditancapkan ke slot ekspansi.

Sebuah PCB khusus disebut *data-acquisition system* (DAS) telah dikembangkan untuk tujuan menyediakan input dan output data analog. Hal ini digunakan ketika PC digunakan untuk sistem pengaturan.

2.6.1 DAS Hardware

Fitur perangkat keras dari sistem akuisisi data digambarkan pada Gambar 2.21. Walaupun ada variasi antar pabrik, sistem yang ditunjukkan pada gambar dan dideskripsikan di dalam sini mendemonstrasikan fitur utama dari DAS.



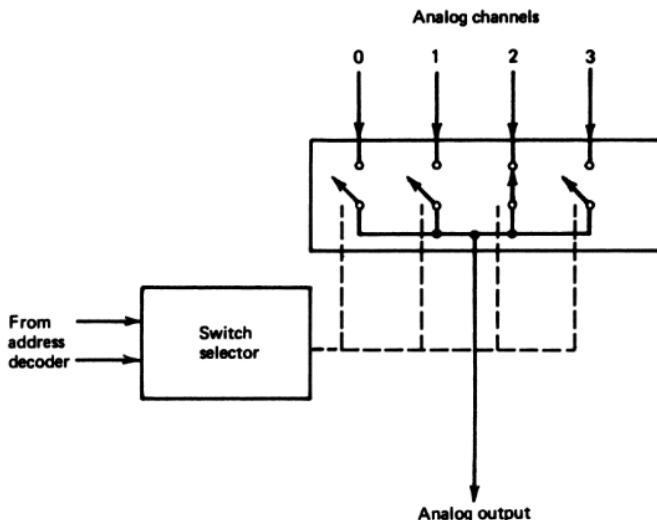
Gambar 2.21 Rangkaian khusus dari *data-acquisition board* untuk digunakan pada slot ekspansi PC [8]

Di dalam DAS terdapat *analog to digital converter* (ADC) dan sistem *sampler and holder* (S/H). DAS secara khusus memiliki kecepatan komputasi yang tinggi, dan sebuah S/H yang cepat. Kapanpun DAS diminta untuk memperoleh sebuah sampel data, S/H secara otomatis S/H terlibat dalam proses. Waktu konversi ADC merupakan bagian utama dari waktu akuisisi pencuplikan data, tetapi waktu akuisisi S/H harus juga dipertimbangkan untuk waktu pelewatan maksimum.

Analog multiplexer (MUX) membuat DAS dapat memilih data dari sejumlah sumber yang berbeda. MUX memiliki sejumlah *channel input*, masing-masing *channel* terhubung dengan sumber tegangan yang berbeda. MUX bekerja seperti kumpulan *switch* yang digambarkan pada Gambar 2.22 yang disusun dengan sebuah cara agar tiap *input channel*

dapat dipilih dari rangkaian S/H dari ADC. Pada beberapa kasus, DAS dapat diprogram untuk mengambil sampel dari *channel* secara sekuensial.

Salah komponen lain dari DAS adalah *address decoder/command processor*. Komputer dapat memilih masukan sebuah sampel dari *channel* yang ada dengan mengirim sebuah pilihan yang tepat pada *address line* dan *control line* dari bus komputer. Pengkodean itu untuk menginisiasi sekuen yang tepat perintah pada MUX, ADC, dan S/H. Fitur umum lainnya adalah kemampuan memprogram DAS untuk mengambil sejumlah sampel dari sebuah *channel* dengan waktu yang spesifik diantara tiap pencuplikan. Pada kasus ini komputer diberi notifikasi oleh intrupsi ketika sampel telah siap untuk *input*.



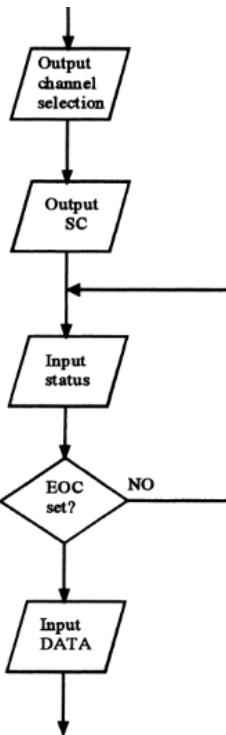
Gambar 2.22 Sebuah multiplexer analog bekerja sebagai sebuah *switch* multiposisi untuk memilih input ke ADC [8]

Untuk tujuan *output*, DAS seringkali dilengkapi dengan *latch* dan DAC. *Address decoder/command processor* digunakan untuk mengunci data yang ditulis ke DAS, yang mana kemudian dikonversi ke sebuah sinyal analog yang cocok oleh DAC.

2.6.2 DAS Software

Proses memilih sebuah *channel* dan menginisiasi sebuah *input* data dari *channel* melibatkan beberapa penghubung diantara komputer dan

DAS. Penghubung ini difasilitasi dengan bantuan *software* yang dieksekusi oleh komputer. *Software* tersebut dapat dibuat sendiri oleh pengguna atau seringkali disediakan oleh produsen DAS dengan bentuk program di dalam CD/DVD. Gambar 2.23 adalah sebuah diagram alir dari sekuen dasar pada operasi yang harus terjadi ketika sebuah sampel diperlukan dari DAS.



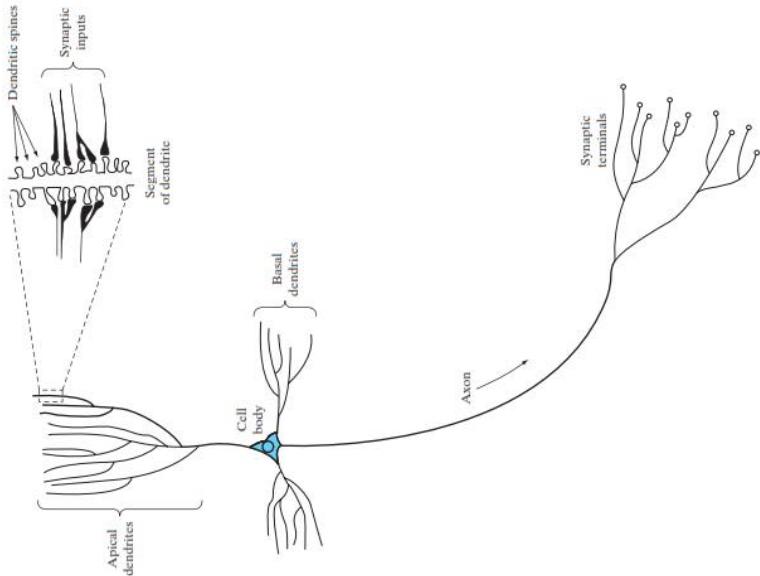
Gambar 2.23 Sekuen dari operasi DAS *software* [8]

2.7 Jaringan Syaraf Tiruan [9]

2.7.1 Jaringan Syaraf Biologi

Otot manusia memiliki konfigurasi yang sangat kompleks dan diberi kemampuan yang sangat luar biasa. Gambar 2.24 melukiskan bahwa dendrit bertugas untuk mengumpulkan informasi dan mengirimkannya ke badan sel. Badan sel akan mengolah informasi yang masuk tersebut menghasilkan output. Output dari badan sel akan dikirim ke sel syaraf lain

melalui axon. Axon terhubung pada sel-sel syaraf yang lain. Sinapsis adalah yang menyambungkan antara axon dan dendrit dari sel syaraf lain. Kekuatan sinapsis menentukan seberapa besar atau banyak informasi yang diteruskan dari axon menuju dendrit.



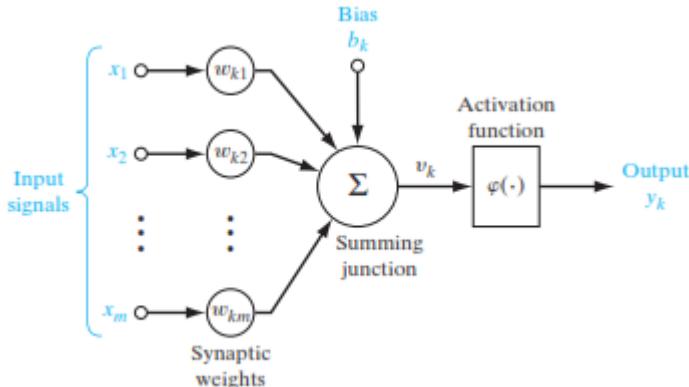
Gambar 2.24 Jaringan syaraf biologi [9]

2.7.2 Konsep Dasar Model Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah sistem pemrosesan informasi yang memiliki karakter menyerupai jaringan syaraf biologi. Jaringan syaraf tiruan dibentuk dari model-model dari neuron atau *node* yang terhubung membentuk sebuah jaringan. Model-model dari neuron yang digunakan merupakan versi yang lebih sederhana dari aksi dari neuron asli. Informasi yang tersimpan di jaringan seringkali memiliki kekuatan koneksi atau bobot yang berbeda yang berasosiasi dengan sinapsis pada model jaringan syaraf tiruan.

Bila dimisalkan sebuah neuron memiliki n input. Masing-masing jaringan dapat mengirimkan sebuah nilai yang riil. Fungsi aktivasi yang dihitung di dalam neuron dapat dipilih dari beberapa fungsi aktivasi yang

ada. Setiap jaringan memiliki bobot saluran sendiri-sendiri, kemudian informasi yang masuk ke dalam jaringan akan dikalikan dengan bobot tersebut. Informasi yang masuk akan diolah di dalam neuron dengan fungsi aktivasi yang telah dipilih. Kemudian *output* fungsi dievaluasi untuk memperbaiki nilai bobot tiap jaringan.



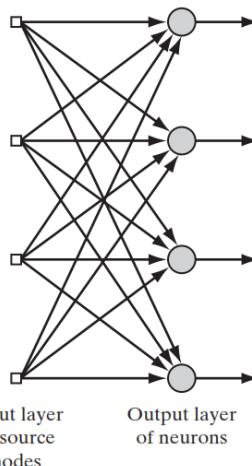
Gambar 2.25 Model jaringan syaraf tiruan [9]

2.7.3 Arsitektur Jaringan Syaraf Tiruan

Secara umum jaringan syaraf tiruan dibagi menjadi tiga struktur yang biasa digunakan yaitu jaringan *single-layer*, *multilayer*, dan *recurrent*.

2.7.3.1 Single Layer Network

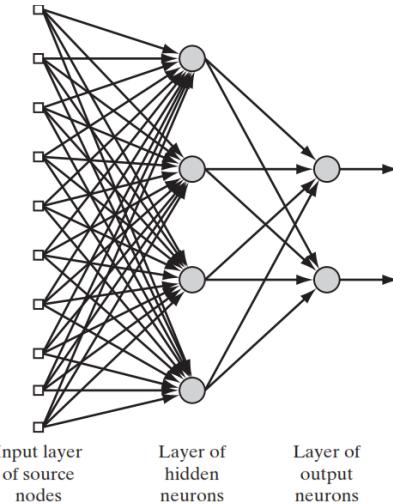
Sebuah jaringan syaraf tersusun di dalam bentuk lapisan-lapisan. Bentuk paling sederhana dari jaringan syaraf berlapis adalah ketika sebuah lapisan *input* dari *node* sumber diproyeksikan langsung ke dalam lapisan *output* dari neuron, tetapi tidak berlaku sebaliknya. Dengan kata lain jaringan ini termasuk tipe *feedforward*. Pada Gambar 2.26 untuk kasus 4 *node* pada lapisan *input* dan *output*. Jaringan seperti itu disebut jaringan *single-layer* dengan tanda *single-layer* mengacu pada lapisan *output* dari node komputasi (neuron). Lapisan input dari *node* sumber tidak dihitung dikarenakan tidak terjadi proses komputasi pada lapisan tersebut.



Gambar 2.26 *Feedforward* dengan neuron *single-layer* [9]

2.7.3.2 Multilayer Network

Jenis arsitektur yang kedua berbeda dengan jaringan *single-layer* dikarenakan adanya satu atau lebih lapisan yang tersembunyi atau *hidden-layer*, yang mana *node* komputasinya saling berhubungan dan disebut dengan neuron tersembunyi atau *hidden neuron* atau juga bisa disebut unit tersembunyi. Istilah tersembunyi mengacu pada fakta bahwa bagian ini pada jaringan syaraf tidak dapat dilihat secara langsung baik dari jaringan *input* maupun *output*. Fungsi dari neuron pada lapisan tersembunyi adalah memberi pengaruh diantara *input* eksternal dan jaringan *output*. Dengan menambah satu atau lebih lapisan tersembunyi, jaringan dapat mengekstak orde statistik yang lebih tinggi dari input. Gambar 2.27 merupakan sebuah contoh dari jaringan multi layer yang terdiri satu *hidden-layer*.

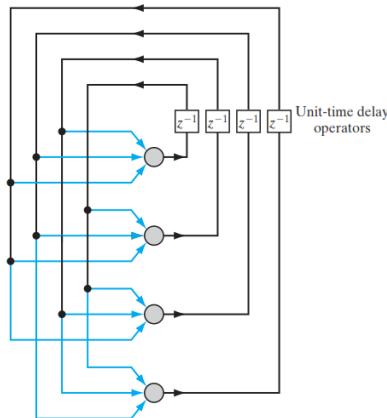


Gambar 2.27 Jaringan *multilayer* dengan satu *hidden-layer* [9]

2.7.3.3 Recurrent Network

Sebuah jaringan *recurrent* memiliki ciri tersendiri yang berbeda dengan jaringan *feedforward* yaitu jaringan tersebut memiliki paling tidak satu *feedback loop*. Sebagai contoh, sebuah jaringan *recurrent* dapat terdiri dari neuron *single-layer* dengan tiap neuron memberi umpan balik sinyal *output* terhadap semua *input* pada neuron yang dilukiskan pada Gambar 2.28.

Adanya *feedback loop* pada struktur jaringan *recurrent* memberikan dampak yang amat besar pada kemampuan belajar dan performanya. Selain itu *feedback loop* melibatkan penggunaan *unit time delay*, yang mana hasil dari perilaku dinamika yang nonlinier bisa diasumsikan dengan jaringan yang mengandung unit nonlinier.



Gambar 2.28 Jaringan *recurrent* tanpa neuron tersembunyi [9]

2.7.4 Fungsi Aktivasi

Pada jaringan syaraf tiruan fungsi aktivasi menentukan *output* dari sebuah *neuron*. Terdapat dua tipe dasar fungsi aktivasi yaitu *threshold* dan *sigmoid*.

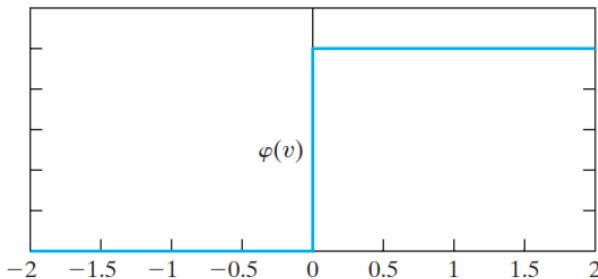
2.7.4.1 Fungsi Threshold

Tipe fungsi aktivasi ini dideskripsikan pada Gambar 2.29 dan dapat dinyatakan pada Persamaan (2.8). Pada istilah teknik fungsi aktivasi *threshold* secara umum disebut sebagai *heaviside function*.

$$\varphi(v) = \begin{cases} \text{if } v \geq 0 \\ \text{if } v < 0 \end{cases} \quad (2.8)$$

Dimana v adalah jumlah total input yang sudah diberi bobot dengan bias yang dapat dinyatakan dengan Persamaan (2.9).

$$v = \sum_{j=1}^m w_{kj}x_j + b_k \quad (2.9)$$



Gambar 2.29 Fungsi *threshold* [9]

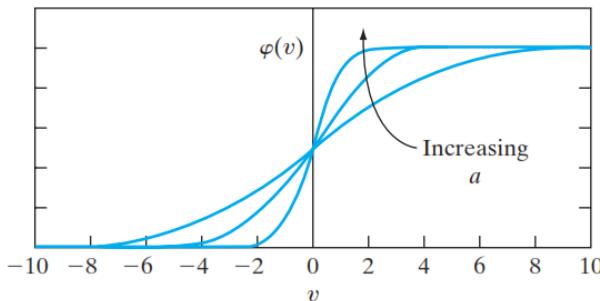
Pada komputasi jaringan syaraf tiruan, neuron yang menggunakan fungsi itu disebut sebagai model McCulloch-Pitts (1943). Pada model tersebut, output dari sebuah neuron adalah nonnegatif. Statemen tersebut mendeskripsikan sifat *all-or-none* dari model McCulloch-Pitts.

2.7.4.2 Fungsi Sigmoid

Fungsi sigmoid yang mana grafiknya berbentuk “S” adalah fungsi yang umum biasa digunakan untuk membentuk jaringan syaraf tiruan. Fungsi sigmoid dapat dinyatakan dengan Persamaan (2.10).

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (2.10)$$

Dimana a adalah parameter *slope* dari fungsi sigmoid. Dengan memvariasi parameter a bisa didapatkan fungsi dengan slope yang berbeda yang ditunjukkan pada Gambar 2.30.



Gambar 2.30 Fungsi sigmoid [9]

2.7.5 Proses Learning

Jaringan syaraf tiruan memiliki cara yang mirip dengan cara kita belajar pada lingkungan sekitar. Dalam arti luas, proses pembelajaran pada jaringan syaraf tiruan dapat dikategorikan menjadi dua yaitu *learning with teacher* dan *learning without teacher*.

2.7.6 Algoritma Back Propagation

Algoritma ini digunakan untuk mencari nilai error minimum. Untuk mencapai eror minimum tersebut dilakukan perubahan bobot yang mana perubahan tersebut memenuhi kriteria $E(n)$ minimum. Dimana $E(n)$ dinyatakan pada Persamaan (2.12)

$$E_j(n) = \frac{1}{2} e_j^2(n) \quad (2.11)$$

$$E(n) = \sum_{j \in C} E_j(n) \quad (2.12)$$

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.13)$$

$$e_j(n) = d_j(n) - y_{net_j}(n) \quad (2.14)$$

$$y_{net_j}(n) = \varphi_j(v_j(n)) \quad (2.15)$$

$$v_j(n) = \sum_{i=1}^m w_{ij}(n) x_i(n) \quad (2.16)$$

dimana $e_j(n)$ menyatakan sinyal eror, $d_j(n)$ adalah keluaran yang diinginkan, dan y_{net} adalah keluaran dari neuron j. Untuk mendapatkan bobot $(w_{ij}(n))$ yang tepat dimana memenuhi kriteria $E(n)$ minimum maka dapat diformulasikan dengan menggunakan aturan rantai sebagai berikut

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_{net_j}(n)} \frac{\partial y_{net_j}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ij}(n)} \quad (2.17)$$

Bila Persamaan (2.11) dirurunkan terhadap $e_j(n)$ maka didapatkan Persamaan (2.18).

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (2.18)$$

Setelah itu turunkan Persamaan (2.14) terhadap $y_{net_j}(n)$ maka didapatkan Persamaan (2.19)

$$\frac{\partial e_j(n)}{\partial y_{net_j}(n)} = -1 \quad (2.19)$$

Kemudian turunkan Persamaan (2.15) terhadap $v_j(n)$ maka akan diperoleh Persamaan (2.20).

$$\frac{\partial y_{net_j}}{\partial v_j(n)} = \varphi'(v(n)) \quad (2.20)$$

Akhirnya dengan menurunkan Persamaan (2.16) terhadap $w_{ij}(n)$ maka akan diperoleh Persamaan (2.21).

$$\frac{\partial v_j(n)}{\partial w_{ij}(n)} = x_i(n) \quad (2.21)$$

Subtitusi Persamaan (2.18)-(2.21) pada Persamaan (2.17) dihasilkan Persamaan (2.22).

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -e_j(n)\varphi'(v(n))x_i(n) \quad (2.22)$$

Koreksi bobot yang diterapkan pada w_{ij} didefinisikan sebagai $\Delta w_{ij}(n)$ yang nilainya sebanding dengan

$$\Delta w_{ij}(n) = -\eta \frac{\partial E(n)}{\partial w_{ij}(n)} \quad (2.23)$$

dimana η adalah *learning rate* pada algoritma *back propagation*.

Persamaan (2.23) dapat dinyatakan juga sebagai

$$\Delta w_{ij}(n) = -\eta \delta_j(n) x_i(n) \quad (2.24)$$

yang mana propagasi eror $\delta(n)$ didefinisikan dengan Persamaan (2.25).

$$\delta_j(n) = \frac{\partial E(n)}{\partial v_j(n)} \quad (2.25)$$

$\delta_j(n)$ dapat diformulasikan dengan aturan rantai sebagai Persamaan (2.26).

$$\delta_j(n) = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial x_i(n)} \frac{\partial x_i(n)}{\partial v_j(n)} \quad (2.26)$$

$$\delta_j(n) = e_j(n)\varphi'(v(n)) \quad (2.27)$$

Dengan demikian dapat diformulasikan mekanisme pembaruan bobot yang dinyatakan pada Persamaan (2.28).

$$w_{ij}(n) = w_{ij}(n - 1) + \Delta w_{ij}(n) \quad (2.28)$$

Pada algoritma *back propagation* secara garis besar memiliki dua tahapan yaitu *feedforward* dan *back propagation*. Langkah-langkah algoritma ini adalah sebagai berikut:

- i. Inisialisasi bobot
- ii. Perhitungan feedforward
- iii. *Back propagation* pada lapisan *output*
- iv. *Back propagation* pada lapisan tersembunyi
- v. Revisi bobot

2.8 Kriteria Kesalahan

2.8.1 Root Mean Square Error (RMSE)

Root mean square error (RMSE) adalah suatu tolak ukur kesalahan yang berdasar pada kuadratis simpangan antara nilai sebenarnya dan nilai estimasi. Persamaan *root mean square* ditunjukkan pada Persamaan (2.29)

$$RMSE = \sqrt{\frac{\sum_{m=1}^N e_m^2}{N}} \quad (2.29)$$

dimana nilai e_m sebanding dengan selisih *output* sebenarnya $y(n)$ dan *output* model $\hat{y}(n)$ yang dinyatakan pada Persamaan (2.30). N adalah banyaknya data.

$$e_m = y(n) - \hat{y}(n) \quad (2.30)$$

2.8.2 Integral of Time Multiplied by Square Error (ITSE)

Integral of time multiplied by square error (ITSE) dapat menjadi tolak ukur dari suatu kesalahan. ITSE bersdasarkan pada integral perkalian antara waktu dan error. ITSE dapat dinyatakan dengan Persamaan (2.31).

$$ITSE = \int_0^N te^2(t)dt \quad (2.31)$$

ITSE memasukan komponen waktu terhadap tolak ukurnya. Komponen waktu ini nantinya akan dapat menunjukan seberapa cepat suatu sistem dapat menghilangkan nilai kesalahannya atau menunjukan kualitas masa transien dari sistem.

2.8.3 Integral of Time Multiplied by Absolut Value of Error (ITAE)

Integral of time multiplied by absolut value of error (ITAE) merupakan kriteria pengukur kesalahan yang lain. Hampir menyerupai ITSE, ITAE juga memasukan komponen waktu namun ITAE memakai nilai absolut dari error yang menjadi nilai tolak ukur. Persamaan (2.32) memformulasikan ITAE.

$$ITAE = \int_0^N t|e(t)|dt \quad (2.32)$$

Pada ITAE nilai kesalahan lebih menjadi perhatian ketimbang kriteria yang lain karena digunakannya nilai absolut dari error.

2.8.4 Integral of Square Time Multiplied by Square Error (ISTSE)

Integral of time multiplied by square error (ISTSE) adalah satu indeks pengukur kesalahan. ISTSE dapat diformulasikan dengan Persamaan (2.33).

$$ISTSE = \int_0^N t^2 e^2(t)dt \quad (2.33)$$

Berbeda dengan ITAE dan ITSE, ISTSE memasukan komponen kuadrat dari waktu hal itu nantinya akan memberikan bobot waktu yang lebih besar disbanding jenis kriteria yang lain. Dengan demikian ISTSE lebih cenderung mengukur kecepatan transien dari suatu sistem daripada kriteria yang lain.

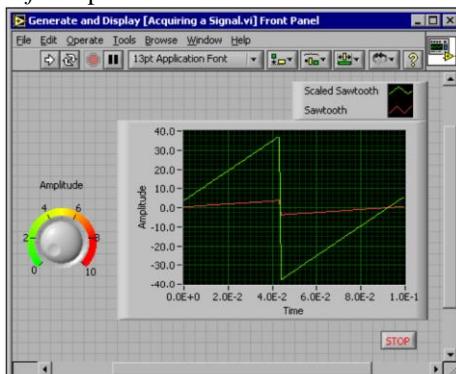
2.9 LabVIEW [10]

LabVIEW adalah sebuah bahasa pemrograman yang menggunakan ikon yang menggantikan barisan dari teks untuk membuat aplikasi. Sangat berbeda dengan pemrograman berbasis teks, dimana instruksi menentukan eksekusi program, LabVIEW menggunakan pemrograman aliran data, dimana aliran dari data menentukan eksekusinya.

Pada LabVIEW dibangun *user interface* dengan kumpulan *tools* dan *object*. *User interface* disebut sebagai *front panel*. Kemudian dapat ditambahkan kode menggunakan representasi gambar dari fungsi untuk mengontrol obyek pada *front panel*. Pada bagian *block diagram* mengandung kode tersebut. Pada beberapa cara, *block diagram* menyerupai sebuah *flowchart*.

Program LabVIEW disebut *virtual instrument*, atau VI karena tampilan dan operasinya meniru instrument fisik, seperti osiloskop dan multimeter. Semua VI menggunakan fungsi yang memanipulasi *input* dari *user interface* atau sumber lain dan menampilkan informasi tersebut atau memindahnya ke *file* lain atau komputer lain.

Front panel adalah *user interface* dari VI. Gambar 2.31 menampilkan contoh dari sebuah *front panel*.

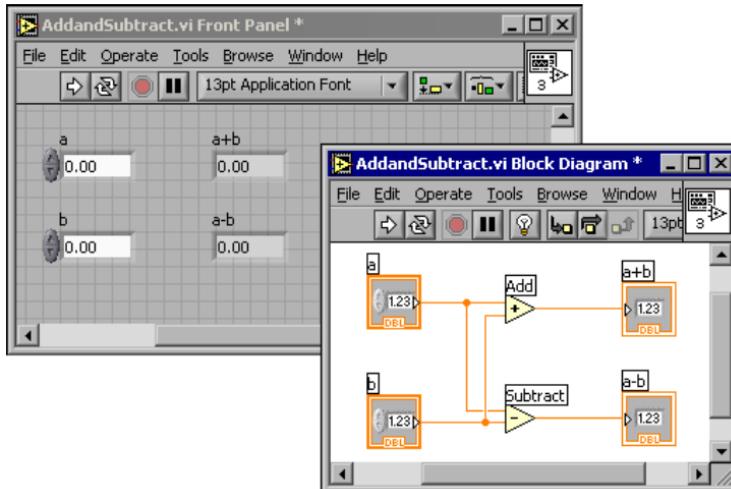


Gambar 2.31 Contoh *front panel* [10]

Front panel dibagun dengan beberapa ‘indicator’ dan ‘control’, yang mana adalah input interaktif dan terminal output dari VI secara berurutan. ‘Control’ adalah knob, push button, dial, dan peralatan input lain. ‘Indicator’ adalah grafik, LED dan tampilan yang lain. ‘Control’ mensimulasikan peralatan masukan instrument dan mensuplai data pada block diagram dari VI. Indicator mensimulasikan peralatan keluaran instrumen dan menampilkan data yang diperoleh block diagram atau yang dihasilkan.

Setelah *front panel* berhasil dibangun, kemudian ditambahkan kode menggunakan representasi gambar dari fungsi untuk mengatur *front panel*. *Block diagram* mengandung *source code* yang berupa gambar. Obyek pada *front panel* tampak sebagai terminal pada *block diagram*. VI

pada Gambar 2.32 menunjukkan beberapa obyek dasar *block diagram*, terminal, fungsi dan wire.



Gambar 2.32 Contoh sebuah *block diagram* dan *front panel* yang terhubung [10]

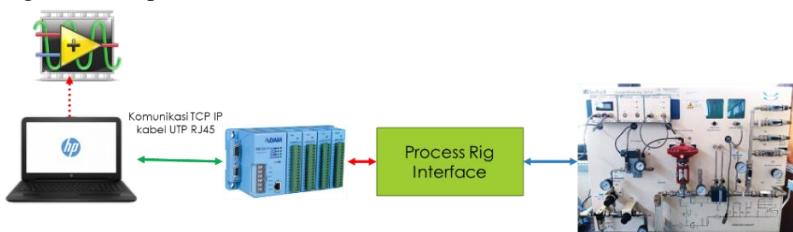
Terminal merepresentasikan tipe data ‘control’ dan ‘indicator’. Terminal adalah *port* masuk dan keluar tempat bertukar informasi diantara *front panel* dan *block diagram*. Misalkan pada Gambar 2.32, data yang dimasukan pada ‘control’ di *front panel* memasuki *block diagram* melalui terminal ‘control’. Kemudian data memasuki fungsi ‘Add’ dan ‘Subtract’. Ketika ‘Add’ dan ‘Subtract’ telah menyelesaikan perhitungan internal, mereka akan memproduksi nilai data yang baru. Data kemudian mengalir ke terminal pada ‘indicator’, dimana data tersebut keluar dari *block diagram* dan masuk kembali ke *front panel*, dan nampak di ‘indicator’ *front panel*.

BAB 3

PERANCANGAN SISTEM

3.1 Arsitektur Sistem

Arsitektur dari sistem yang akan dibangun terdiri beberapa komponen yaitu komputer, *plant*, perangkat akuisisi data, dan jaringan komunikasi. Kontroler dari sistem berada di dalam komputer. Skema yang digunakan adalah *direct digital control* (DDC). Ilustrasi arsitektur system digambarkan pada Gambar 3.1



Gambar 3.1 Arsitektur sistem pengaturan tekanan *Process Rig 38-741*

3.1.1 Komputer

Komputer disini berfungsi sebagai kontroler. Komputer yang digunakan dilengkapi dengan *software* LabVIEW yang berfungsi untuk implementasi metode kontrol dan penghubung antara perangkat akuisisi data dan komputer.

Spesifikasi komputer yang dipakai adalah sebagai berikut:

Processor	: Intel® Core™ i3 M380 2,53 GHz
RAM	: DDR3 4GB
VGA	: Intel® 1,68 GB
<i>Operating System</i>	: Windows 10 64-bit
Resolusi Tampilan	: 1366×768×60
USB Port	: 2×USB 2.0 dan 1×USB 3.0
Ethernet Port	: Yes

3.1.2 ADAM-5000/TCP [11]

Tidak diragukan lagi koneksi ethernet menjadi tren besar untuk aplikasi industri. Jarak komunikasi yang lebih jauh, kecepatan komunikasi yang lebih cepat, dan keuntungan yang lebih besar menarik orang-orang ke dalam pengembangan sistem mereka berbasis skenario jaringan ini. Orang-orang biasanya harus menyiapkan sebuah server pertukaran data

diantara sistem informasi dan sistem kontrol sebagai jembatan komunikasi. Secara jelas, hal itu memakan banyak waktu dan dana. Untuk memenuhi kebutuhan pengguna Advantech meluncur sebuah sistem sistem DA&C, seri ADAM-5000/TCP, solusi Ethernet I/O bagi orang-orang yang mengembangkan arsitektur otomasi berbasis ethernet mereka. Hal itu dapat diaplikasikan ke berbagai bidang seperti lalu lintas, bangunan, telekomunikasi, pengolahan air dan lain sebagainya.

Spesifikasi Teknis dari ADAM5000/TCP:

System

CPU: ARM 32-bit RISC CPU

Memory: 4 MB Flash RAM

Operating System: Real-time O/S

Timer BIOS: Yes

I/O Capacity: 8 slots (ADAM-5000/TCP)

4 slot (ADAM-5000L/TCP)

Status Indicator: Power (3,3V, 5V), CPU, Communication (*Link, Collide*, 10/100 Mbps, Tx, Rx)

CPU Power Consumption: 5,0 W

Reset Push Button: Yes

Komunikasi Ethernet

Ethernet: 10 BASE-T IEEE 802.3
100 BASE-TX IEEE 802.3u

Wiring: UTP, kategori 5 atau lebih

Koneksi Bus: RJ45 jack modular

Protokol Komunikasi: Modbus/TCP

Data Transfer Rate: sampai dengan 100Mbps

Waktu respon: kurang dari 5 ms

Komunikasi Serial

Sinyal RS-485: DATA+, DATA-

Mode: *Half duplex, multi-drop*

Konektor: *screw terminal*

Kecepatan transmisi: sampai dengan 115,2 Kbps

Jarak Transmisi maksimum: 4000 kaki (1220m)

Power

Unregulated 10V sampai 30 V DC

Proteksi: *Over-voltage* dan *power reversal*

Lingkungan

Temperatur Operasi: -10 sampai 70°C (14 sampai 158°F)

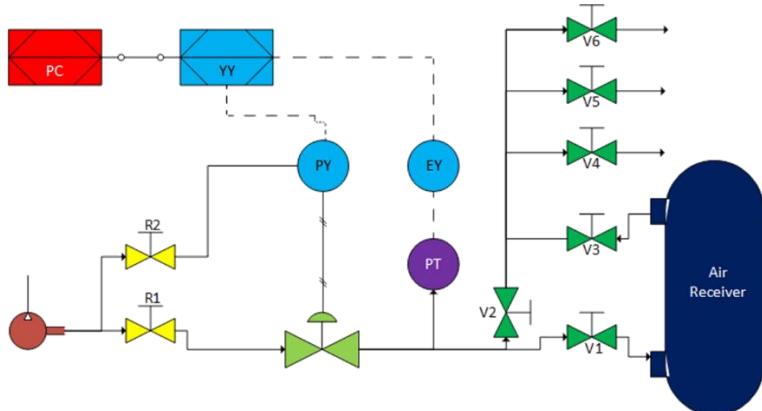
Temperatur Penyimpanan: -25 sampai 85°C (-13 sampai 185°F)

Kelembapan: 5 sampai 95%, *non-condensing*

Atmosfir: tidak ada gas korosif

3.1.3 Plant Pressure Process Rig 38-714

Pada penelitian ini *plant* yang akan dikendalikan adalah *Pressure Process Rig-38-714*. Variabel yang akan dikontrol adalah tekanan. Untuk mengatur tekanan pada *plant* dilakukan dengan memanipulasi *control valve* sebagai aktuator sistem sehingga sinyal kontrol akan diberikan pada *control valve* melalui I/P converter.



Gambar 3.2 P&ID Pressure Process Rig 38-741

Keterangan:

PT = *Pressure Transmitter (Pressure Sensor)*

I/P = Pengonversi arus (I) ke bentuk *pressure* (P)

I/V = Pengonversi dari dan ke bentuk tegangan (V) dan arus (I)

V1-V6 = *Manual Valve*

3.1.4 Jaringan Ethernet

Kabel UTP RJ-45 digunakan pada sistem ini. Panjang kabel adalah 1.5 meter. Jaringan ethernet dipakai sebagai media komunikasi data antara komputer dan perangkat akuisisi data.

3.1.5 LabVIEW

Untuk menghubungkan antara perangkat keras akuisisi data dan komputer digunakan LabVIEW yang dapat berfungsi sebagai penghubung kedua perangkat tersebut. LabVIEW juga dipakai untuk mengimplementasikan metode kontrol yang dirancang. Pada LabVIEW yang dipakai adalah versi 32bit. Modul DSC dan Real Time harus sudah

ter-install pada LabVIEW supaya dapat menggunakan NI I/O server agar perangkat akuisisi dapat terhubung dengan LabVIEW.

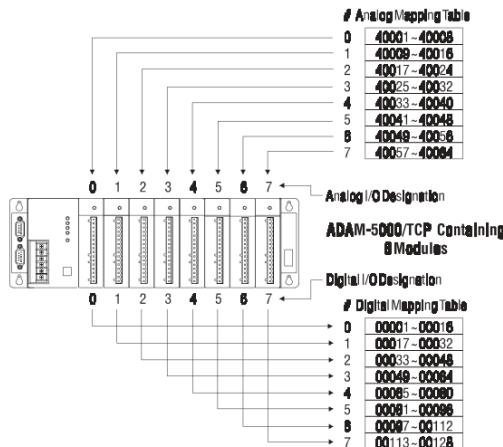
3.2 Perancangan Perangkat Akuisisi Data [11]

Untuk mengolah data hasil pengukuran diperlukan sebuah penghubung ke komputer. Perangkat akuisisi data yang digunakan adalah ADAM-5000/TCP. Perangkat tersebut dilengkapi dengan satu slot modul input dan satu slot modul output. Modul *input* adalah ADAM-5017 yang terdiri dari 8 kanal *input* berupa tegangan dengan satuan milivolt (± 150 mV dan ± 500 mV) dan volt (± 1 , ± 5 V, dan ± 10 V). Modul *output* adalah ADAM-5024 yang dilengkapi dengan 4 kanal *output* yang dapat berupa tengangan (0-10V) atau arus (0-20mA dan 4-20mA). Satu kanal input dengan operasi 0-10V dipakai untuk membaca keluaran *pressure sensor* yang telah mendapat *signal conditioning* pada *process rig interface*. Satu kanal output dengan operasi 4-20mA dipakai untuk memberi sinyal kontrol pada *plant*. Kanal nomor 0 pada modul input ADAM-5017 dan ADAM-5024 digunakan untuk menjalankan operasi seperti yang telah dijelaskan sebelumnya.

Karena adanya proses akuisisi terjadi juga proses *scaling* pada ADAM-5000/TCP untuk menyatakan representasi data *input* dan *output*. Tabel 3.1 menunjukkan representasi data penyekalaan yang dilakukan pada perangkat akuisisi. Pengalamatan data kanal pada modul *input* dan *output* yang terpasang bergantung pada tipe modul yang dipakai. ADAM-5000L/TCP memiliki nomor slot 0 sampai dengan 3 yang berasal dari slot paling kiri, sementara ADAM-5000/TCP memiliki nomor slot 0-7 yang juga berasal dari slot paling kiri. Gambar 3.3 menunjukkan pengalamatan kanal dengan nomor slot dan tipe slot yang dipakai.

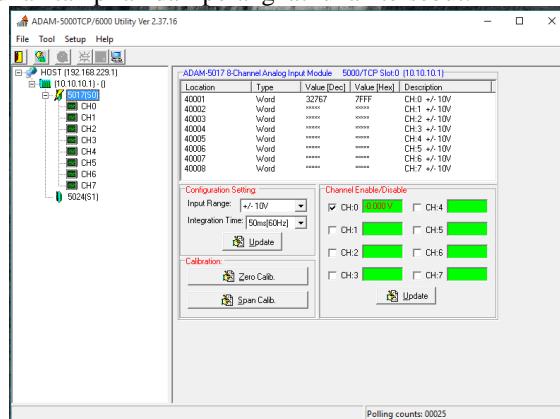
Tabel 3.1 ADAM-5000 AI/AO scaling [11]

Modul	Tipe	Range Low	Range High	Scale Low	Scale High	Format Data
5017 AI	mV	-150	150	0	65535	U16B
	mV	-500	500	0	65535	U16B
	V	-1	1	0	65535	U16B
	V	-5	5	0	65535	U16B
	V	-10	10	0	65535	U16B
	mA	-20	20	0	65535	U16B
5024 AO	V	0	10	0	4095	U12B
	mA	4	20	0	4095	U12B
	mA	0	20	0	4095	U12B



Gambar 3.3 Pengalamatan kanal I/O ADAM-5000/TCP [11]

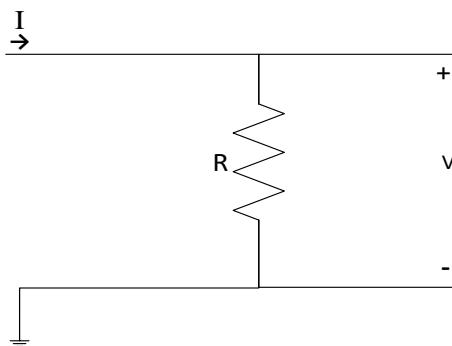
Untuk melakukan penyetelan awal pada perangkat akuisisi ADAM-5000TCP-6000 Utility dapat digunakan. Penyetelan yang dapat dilakukan adalah pemilihan tipe I/O yang dipakai, kanal mana yang dipakai, dan pemantauan kondisi perangkat, serta dapat dilakukan kalibrasi. Gambar 3.4 menunjukkan tampilan dari perangkat lunak tersebut.



Gambar 3.4 Software ADAM-5000TCP-6000 Utility

3.3 Desain I/V Converter

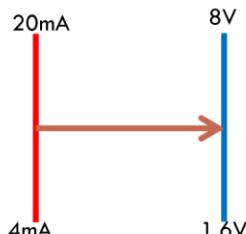
Output dari pressure sensor yang berupa arus tidak dapat diterima langsung oleh *analog input channel* pada modul input ADAM-5017 yang masukannya berupa tegangan. Agar bisa diterima maka sinyal arus tersebut harus diubah menjadi tegangan. Rangkaian listrik dengan komponen pasif digunakan untuk merubah arus menjadi tegangan. Arus dari *pressure sensor* diberikan beban resitif sehingga menghasilkan tegangan. Gambar 3.5 menunjukkan rangkaian yang digunakan.



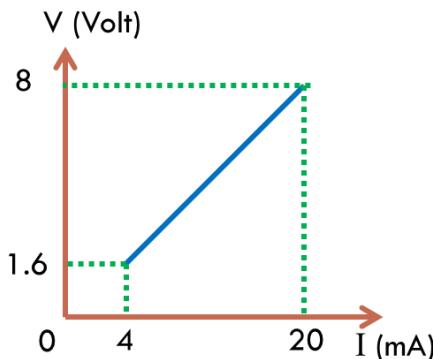
Gambar 3.5 Rangkaian I/V converter

Arus yang masuk berupa 4-20 mA DC akan dirubah menjadi tegangan 1,6-8 V DC.

$$V = I R \quad (3.1)$$



Gambar 3.6 Skema konversi sinyal arus ke tegangan



Gambar 3.7 Grafik fungsi tegangan terhadap arus desain *converter* I/V

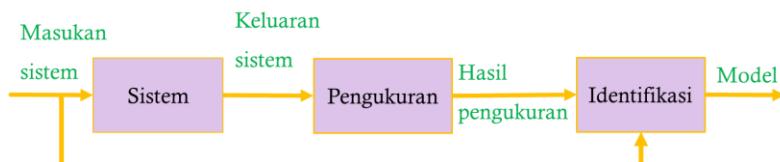
Desain grafik fungsi dari tegangan terhadap arus yang diinginkan dilukiskan pada Gambar 3.7. Melalui grafik tersebut dapat diturunkan persamaan tegangan terhadap arus yang dinotasikan pada Persamaan (3.2).

$$V = 400 I \quad (3.2)$$

Dengan demikian maka pada rangkaian *converter* digunakan beban resistif sebesar 400Ω .

3.4 Identifikasi Sistem *Open Loop*

Pemodelan adalah tahapan yang penting di dalam siklus desain sistem pengaturan. Untuk mendapatkan model dari *plant Pressure Process Rig* 38-714 dilakukan identifikasi sistem yang merupakan salah satu cara untuk mendapatkan model dari sistem. Gambar 3.8 menunjukkan skema identifikasi sistem yang dilakukan. Metode yang digunakan adalah identifikasi statis.



Gambar 3.8 Skema identifikasi sistem

3.4.1 Identifikasi Statis Sistem *Open Loop*

Identifikasi statis adalah salah satu metode identifikasi, dimana dilakukan dengan memberikan masukan atau *set value* pada *plant* berupa sinyal konstan. Sinyal yang dapat digunakan adalah sinyal *unit step*. Sistem yang akan diidentifikasi adalah sistem *open loop* dari *plant Pressure Process Rig 38-714*. Data respon *plant* nantinya dikumpulkan dan kemudian dianalisa utnuk mendapatkan model dari *plant*. Identifikasi dilakukan dengan keadaan *plant* sebagai berikut:

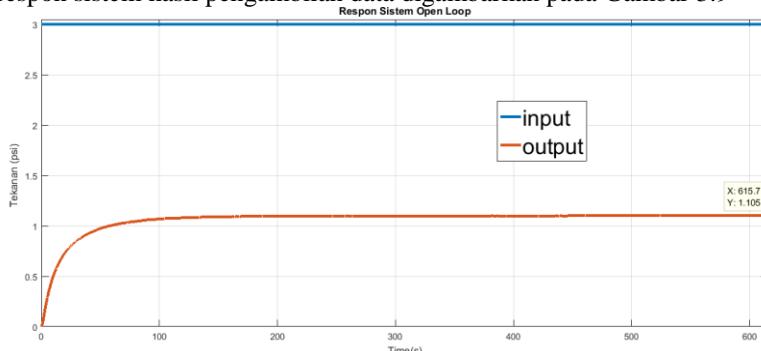
1. Masukan *plant* : 15,2 mA (3 psi)
2. Jumlah data yang diambil : 30000 sampel
3. Kondisi nominal :
 - Katup terbuka : V1, V3, V4, dan V5
 - Katup tertutup : V2 dan V6
4. *Air receiver* : Digunakan

3.4.2 Formulasi Model Sistem

Model dari sistem didekati dengan model sistem orde 1 karena setelah dilihat respon dari sistem menunjukkan kemiripan sistem orde satu. *Delay* pada pada sistem tidak dimodelkan karena nilainya yang sangat kecil kecil sehingga dapat diabaikan. Secara umum sistem orde satu dapat dinyatakan dengan fungsi alih yang ditulis pada Persamaan (3.3) [12]. Grafik

$$G(s) = \frac{K}{\tau s + 1} \quad (3.3)$$

respon sistem hasil pengambilan data digambarkan pada Gambar 3.9



Gambar 3.9 Grafik respon *open loop Pressure Process Rig 38-714*

Proses pencarian parameter-parameter dari sistem tersebut dilakukan dengan beberapa tahapan dibawah. Pertama dilakukan perhitungan *gain overall* (K) yang dapat dicari dengan Persamaan (3.4).

$$K = \frac{Y_{ss}}{X_{ss}} \quad (3.4)$$

dimana Y_{ss} adalah respon sistem pada kondisi *steady state* dan X_{ss} adalah masukan sistem pada kondisi *steady state*. Respon *output* sistem mengalami kondisi *steady state* ketika bernilai 1.105 psi sedangkan masukan sistem mengalami *steady state* ketika 3 psi. Dengan demikian nilai *gain overall* didapatkan

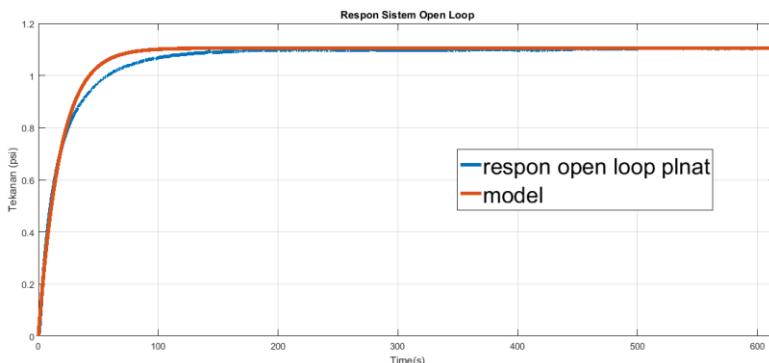
$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1.105}{3} = 0,36833$$

Setelah mendapatkan nilai *gain overall* dari model sistem adalah dilakukan pencarian nilai *time response* (τ) dari sistem. Nilai τ dapat dicari melalui nilai waktu ketika respon mencapai 63.2% dari keadaan *steady state*. Nilai steady state dari sistem adalah 1.105 sehingga 63.2% dari nilai tersebut adalah 0.69836 psi. Waktu ketika respon mencapai nilai tersebut adalah 18.12 detik. Dengan demikian fungsi alih dari *plant* dapat dinyatakan pada Persamaan (3.5).

$$G(s) = \frac{0.36833}{18.12s + 1} \quad (3.5)$$

3.4.3 Validasi Model

Gambar 3.10 menunjukkan perbandingan antara model sistem yang didapat pada pemodelan pada bagian 3.4.2. Bila dipakai kriteria *mean*



Gambar 3.10 Perbandingan respon *open loop plant* dengan model yang didapat

square error (MSE) untuk membandingkan model dan respon *open loop* dari *plant* didapatkan nilai $3,5435 \times 10^{-4}$.

3.5 Penentuan Frekuensi Sampling

Penentuan frekuensi sampling pada sistem pengaturan digital adalah permasalahan yang penting. Frekuensi sampling mempengaruhi banyak sifat pada sebuah sistem, seperti noise pengukuran dan sensitivitas pada dinamika yang belum termodelkan. Dengan demikian pemilihan frekuensi sampling menjadi permasalahan dalam tahap desain sistem pengaturan.

Salah satu aturan untuk menentukan yang berguna untuk metode desain adalah memilih interval sampling h dipilih sedemikian rupa sehingga memenuhi

$$\omega_o h \approx 0,2 - 0,6 \quad (3.6)$$

dimana ω_o adalah frekuensi natural pole dominan dari sistem *closed-loop* [2].

3.6 Desain Estimator Parameter Sistem

Salah satu bagian dari *self-tunning regulator* adalah estimator parameter model sistem. Pada bagian ini akan dibahas desain estimator parameter dengan dua metode yaitu *recursive least square* dan *artificial neural network*.

3.6.1 Recursive Least Square (RLS) Model Estimation [2]

Sebuah model matematik dapat dinyatakan dalam bentuk

$$y(k) = \varphi_1(k)\theta_1 + \varphi_2(k)\theta_2 + \dots + \varphi_n(k)\theta_n = \boldsymbol{\varphi}^T(k)\boldsymbol{\theta} \quad (3.7)$$

dimana y adalah *output* dari sistem, θ adalah parameter dari model matematik yang tidak diketahui, dan φ adalah variable regresi yang terdiri dari data *input* dan *output* sistem.

$$\boldsymbol{\varphi}^T(k) = [\varphi_1(i) \quad \varphi_2(i) \quad \dots \quad \varphi_n(k)] \quad (3.8)$$

$$\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]^T \quad (3.9)$$

Bila dinyatakan error *output* sebenarnya dibanding estimasi *output* dinyatakan dengan

$$e_k = y(k) - \hat{y}(k) \quad (3.10)$$

$$y(k) = \hat{y}(k) + e_i \quad (3.11)$$

$$E = [e_1 \quad e_2 \quad \dots \quad e_n]^T \quad (3.12)$$

Least square adalah permasalahan perhitungan parameter model sistem dimana parameter tersebut memenuhi kriteria jumlahan dari eror estimasi minimal yang dinyatakan pada Persamaan (3.13).

$$\sum_{n=1}^N e_n^2 = E^T E \quad (3.13)$$

$$E^T E = Y - \hat{Y} \quad (3.14)$$

$$E^T E = (Y - \phi\theta)^T (Y - \phi\theta) \quad (3.15)$$

$$E^T E = Y^T Y - \phi^T Y \theta - Y^T \phi \theta + \phi^T \phi \theta^2 \quad (3.16)$$

Matriks $\phi^T Y$ besifat simeteri sehingga Persamaan (3.16) dapat dinyatakan sebagai Persamaan (3.17).

$$E^T E = Y^T Y - 2\phi^T Y \theta + \phi^T \phi \theta^2 \quad (3.17)$$

Nilai parameter model (θ) yang meminimalkan kriteria Persamaan (3.13) dapat dicari dengan menurunkan persamaan fungsi kriteria tersebut terhadap variabel parameter model sistem (θ) dengan nilai turunan sama dengan nol.

$$\frac{\partial(E^T E)}{\partial \theta} = 0 \quad (3.18)$$

$$\frac{\partial(E^T E)}{\partial \theta} = \frac{\partial(Y^T Y - 2\phi^T Y \theta + \phi^T \phi \theta^2)}{\partial \theta} = 0 \quad (3.19)$$

$$-2\phi^T Y + 2\phi^T \phi \theta = 0 \quad (3.20)$$

$$2\phi^T Y = 2\phi^T \phi \theta \quad (3.21)$$

$$\theta = (\phi^T \phi)^{-1} \phi^T Y \quad (3.22)$$

Proses *least square* di atas dilakukan secara *batch* atau semua data dikumpulkan terlebih dahulu untuk mencari parameter model sistem. Pada *self tuning regulator* estimasi parameter dilakukan secara rekursif atau terus menerus setiap pengambilan sampel data. Dengan demikian harus dilakukan modifikasi pada estimasi model secara *least square* agar dapat bekerja secara rekursif.

Bila dinyatakan sebuah matrik untuk menyatakan perkalian matriks regresor transpose terhadap matrik regresor yang dinyatakan pada Persamaan (3.23).

$$P^{-1}(k) = \phi^T(t)\phi(k) = \sum_{i=1}^k \varphi(i)\varphi^T(i) \quad (3.23)$$

$$P^{-1}(k) = \sum_{i=1}^{k-1} \varphi(i)\varphi^T(i) + \varphi(k)\varphi^T(k) \quad (3.24)$$

$$P^{-1}(k) = P^{-1}(k-1) + \varphi(k)\varphi^T(k) \quad (3.25)$$

Persamaan *least square* yang didapat pada Persamaan (3.22) dapat dinyatakan sebagai persamaan (3.26).

$$\theta(k) = \left(\sum_{i=1}^k \varphi(i)\varphi^T(i) \right)^{-1} \left(\sum_{i=1}^k \varphi(i)y(i) \right) \quad (3.26)$$

$$\theta(k) = P(k) \left(\sum_{i=1}^k \varphi(i)y(i) \right) \quad (3.27)$$

$$\theta(k) = P(k) \left(\sum_{i=1}^{k-1} \varphi(i)y(i) + \varphi(k)y(k) \right) \quad (3.28)$$

$$\sum_{i=1}^{k-1} \varphi(i)y(i) = \emptyset^T(k-1)Y(k-1) \quad (3.29)$$

$$\emptyset^T(k-1)Y(k-1) = \emptyset^T(k-1)\emptyset(k-1)\theta(k-1) \quad (3.30)$$

$$\sum_{i=1}^{k-1} \varphi(i)y(i) = P^{-1}(k-1)\theta(k-1) \quad (3.31)$$

$$\sum_{i=1}^{k-1} \varphi(i)y(i) = (P^{-1}(k) - \varphi(k)\varphi^T(k))\theta(k-1) \quad (3.32)$$

$$\sum_{i=1}^{k-1} \varphi(i)y(i) = P^{-1}(k)\theta(k-1) - \varphi(k)\varphi^T(k)\theta(k-1) \quad (3.33)$$

$$\theta(k) = P(k)(P^{-1}(k)\theta(k-1) - \varphi(k)\varphi^T(k)\theta(k-1) + \varphi(k)y(k)) \quad (3.34)$$

$$\theta(k) = \theta(k-1) - P(k)(\varphi(k)\varphi^T(k)\theta(k-1) + \varphi(k)y(k)) \quad (3.35)$$

Persamaan (3.35) telah menunjukkan bahwa estimasi parameter dapat dilakukan secara rekursif.

RLS menghitung parameter model sistem pada tiap siklus. Ketika jumlah masukan pada RLS terlalu banyak hal tersebut akan membuat waktu kalkulasi matriks P menjadi lebih lama. Algoritma proyeksi Kaczmarz's adalah solusi sederhana untuk menyelesaikan permasalahan kalkulasi matrik P yang menjadi lebih lama. Pada algoritma ini nilai pemilihan nilai θ dengan meminimalkan $\|\theta(k) - \theta(k-1)\|$ dengan batasan $y(k) = \varphi^T(k)\theta(k)$. Dilakukan penambaan pengali Lagrange

(α) pada batasan tersebut. Dengan demikian maka dapat dilakukan peminimalan fungsi yang dinyatakan pada Persamaan (3.36).

$$V = \frac{1}{2}(\theta(k) - \theta(k-1))^T(\theta(k) - \theta(k-1)) + \alpha(y(k) - \varphi^T(k)\theta(k)). \quad (3.36)$$

Bila Persamaan 3.6.30 diturunkan terhadap θ dan α didapatkan

$$\theta(k) - \theta(k-1) - \alpha\varphi(k) = 0 \quad (3.37)$$

$$y(k) - \varphi^T(k)\theta(k) = 0 \quad (3.38)$$

Kedua persamaan tersebut dapat dicari solusinya yang mana didapatkan

$$\begin{aligned} \theta(k) &= \theta(k-1) + \\ &\frac{\varphi(k)}{\varphi^T(k)\varphi(k)}(y(k) - \varphi^T(k)\theta(k-1)) \end{aligned} \quad (3.39)$$

Penambahan variabel *forgetting factor* sangat berguna bila diaplikasikan pada proses dengan parameter yang *time varying* karena itu dilakukan modifikasi pada Persamaan (3.39) dengan menambahkan γ dan α sehingga terbentuk Persamaan (3.40).

$$\begin{aligned} \theta(k) &= \theta(k-1) + \\ &\frac{\gamma\varphi(k)}{\alpha + \varphi^T(k)\varphi(k)}(y(k) - \varphi^T(k)\theta(k-1)) \end{aligned} \quad (3.40)$$

Nilai α harus dipilih bilangan positif ($\alpha > 0$) untuk menghindari pembagi nol pada Persamaan (3.40), dan nilai harus γ dipilih antara 0 dan 2 ($0 < \gamma < 2$).

3.6.2 Estimasi Parameter Model dengan Pendekatan

Jaringan Syaraf Tiruan [13]

Bentuk umum struktur model ARX dapat dinyatakan pada Persamaan 3.6.35.

$$y(k) = a_1y(k-1) + \dots + a_ny(k-n) + b_0x(k) + \dots + b_mx(k-m) + e(k) \quad (3.41)$$

$$y(k) = \sum_{i=0}^m b_i x(k-i) + \sum_{j=1}^n a_j y(k-j) + e(k) \quad (3.42)$$

Dimana $x(k)$ dan $y(k)$ menyatakan nilai *input* dan *output* pada saat k sementara b_i dan a_j adalah parameter dari model. Persamaan (3.42) dapat dinyatakan dalam bentuk vektor Persamaan (3.43)

$$y(k) = \theta^T \emptyset(k) \quad (3.43)$$

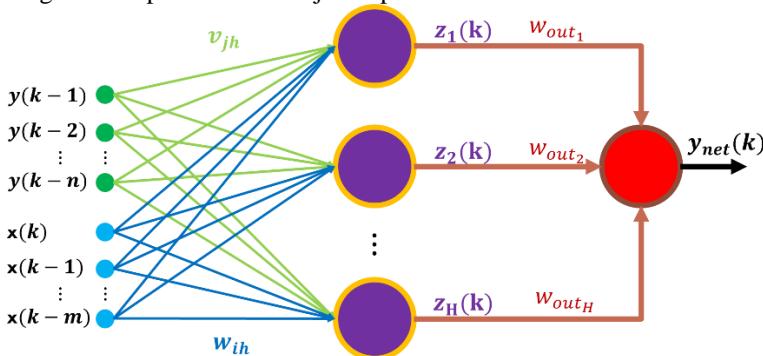
dimana θ^T dan $\emptyset(k)$ dinyatakan sebagai Persamaan (3.44) dan

Persamaan (3.45)

$$\theta^T = [a_1, \dots, a_n, b_0, \dots, b_m] \quad (3.44)$$

$$\emptyset(k) = [y(k-1), \dots, y(k-n), x(k), \dots, x(k-m)] \quad (3.45)$$

Jaringan syaraf tiruan atau *artificial neural network* dapat digunakan untuk mengestimasi parameter dari model sistem ARX. Salah satu arsitektur *artificial neural network* yang dapat digunakan untuk mengestimasi parameter ditunjukkan pada Gambar 3.11.



Gambar 3.11 Bentuk umum arsitektur *artificial neural network* yang digunakan

Keluaran dari *artificial neural network* dinyatakan pada Persamaan (3.46) sementara keluaran dari neuron *hidden layer* dinyatakan pada Persamaan (3.47).

$$y_{net}(k) = g\left(\sum_{h=1}^H z_h(k) w_{out_h}\right) \quad (3.46)$$

$$z_h(k) = f\left(\sum_{i=0}^m w_{ih} x(k-i) + \sum_{j=1}^n v_{jh} y(k-j)\right) \quad (3.47)$$

Fungsi aktivasi yang digunakan adalah fungsi tangen hiperbolik yang nilainya berkisar antara -1 sampai 1. Dengan menggunakan fungsi tangen hiperbolik maka Persamaan (3.47) dapat dinyatakan sebagai Persamaan (3.48)

$$z_h(k) = \frac{1 - e^{-\alpha net_h(k)}}{1 + e^{-\alpha net_h(k)}} \quad (3.48)$$

$$net_h(k) = \sum_{i=0}^m w_{ih} x(k-i) + \sum_{j=1}^n v_{jh} y(k-j) \quad (3.49)$$

Nilai α dipilih sama dengan 1 agar mempermudah perhitungan pendekatan fungsi tangen hiperbolik nantinya. Taylor *expansion* dipakai

untuk mencari pendekatan fungsi tangen hiperbolik. Bentuk umum Taylor expansion dinyatakan pada Persamaan (3.50) [14].

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x - a)^k \quad (3.50)$$

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x - a) &= f(a) + f'(a)(x - a) + \\ &\quad \frac{f''(a)}{2!} (x - a) + \dots \\ &\quad + \frac{f^{(k)}(a)}{k!} (x - a)^k \end{aligned} \quad (3.51)$$

Pendekatan fungsi tangen hiperbolik dilakukan pada titik 0 yang masih berada di dalam daerah kodomain fungsi tangen hiperbolik. Persamaan (3.43) bila didekati dengan Taylor expansion pada titik 0 maka akan diperoleh Persamaan (3.52).

$$\begin{aligned} z_h(k) &= \frac{1 - e^0}{1 + e^0} + \frac{2e^0}{(1 + e^0)^2} (net_h(k) - 0) \\ &\quad + \frac{-2(1 - e^0)}{(1 - e^0)^3} (net_h(k) - 0)^2 + \dots \end{aligned} \quad (3.52)$$

Agar terhindar dari nonlinieritas hanya diambil dua deret pertama saja.

$$z_h(k) = \frac{2e^0}{(1 + e^0)^2} (net_h(k)) \quad (3.53)$$

$$z_h(k) = \frac{1}{2} (\sum_{i=0}^m w_{ih} x(k-i) + \sum_{j=1}^n v_{jh} y(k-j)) \quad (3.54)$$

Sementara bila persamaan *output* atau Persamaan (3.46) didekati dengan Taylor expansion akan diperoleh Persamaan (3.55).

$$y_{net}(k) = \frac{1}{2} \sum_{h=1}^H (w_{outh} z_h(k)) \quad (3.55)$$

Subtitusi persamaan (3.54) pada Persamaan (3.55) maka diperoleh Persamaan (3.57).

$$y_{net}(k) = \frac{1}{2} \sum_{h=1}^H \left(\frac{w_{outh}}{2} \left(\begin{array}{l} \sum_{i=0}^m w_{ih} x(k-i) \\ + \sum_{j=1}^n v_{jh} y(k-j) \end{array} \right) \right) \quad (3.56)$$

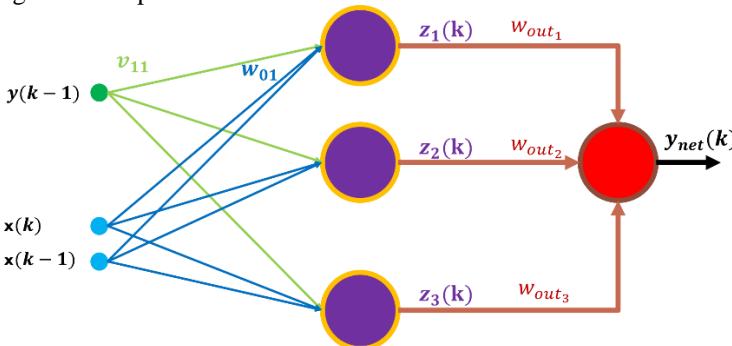
$$y_{net}(k) = \frac{1}{4} \sum_{h=1}^H \left(w_{out_h} \left(\sum_{i=0}^m w_{ih} x(k-i) \right) \right) + \frac{1}{4} \sum_{h=1}^H \left(w_{out_h} \left(\sum_{j=1}^n v_{jh} y(k-j) \right) \right) \quad (3.57)$$

Bentuk persamaan (3.57) akan menyerupai bentuk umum persamaan ARX yaitu persamaan (3.42) jika ditambahkan komponen eror estimasi ($e(k)$). Bila dibandingkan kedua persamaan tersebut maka nilai parameter-parameter dapat dikalkulasikan sebagai berikut :

$$\hat{a}_j = \frac{1}{4} \left(\sum_{h=1}^H (w_{out_h} v_{jh}) \right) \quad (3.58)$$

$$\hat{b}_l = \frac{1}{4} \left(\sum_{h=1}^H (w_{out_h} w_{ih}) \right) \quad (3.59)$$

Model dari sistem yang akan diestimasi adalah struktur model ARX yang bisa dinyatakan dengan Persamaan (3.72). Arsitektur dari *artificial neural network* yang digunakan untuk mengestimasi parameter ARX(1,1) digambarkan pada Gambar 3.12.



Gambar 3.12 Arsitektur *artificial neural network* yang dipakai

Dengan menggunakan Persamaan (3.58) dan (3.59) nilai parameter model dapat dikalkulasikan sebagai berikut:

$$\widehat{a_1} = \frac{1}{4} \left(\sum_{h=1}^H (w_{out_h} v_{1h}) \right) \quad (3.60)$$

$$\widehat{b_0} = \frac{1}{4} \left(\sum_{h=1}^H (w_{out_h} w_{0h}) \right) \quad (3.61)$$

$$\widehat{b_1} = \frac{1}{4} \left(\sum_{h=1}^H (w_{out_h} w_{1h}) \right) \quad (3.62)$$

3.7 Desain Self-Tunning Regulator [2]

Setelah dilakukan identifikasi dan diketahui respon sistem meyerupai sistem orde satu maka fungsi alih *plant* dapat dinyatakan pada Persamaan (3.63) [12]. Dimana $Y(s)$ adalah *output* dan $U(s)$ adalah *input*. Model tersebut

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \quad (3.63)$$

akan dinyatakan dengan model diskrit karena hasil estimasi parameter adalah model diskrit. Persamaan (3.63) dapat diubah ke model diskrit dengan menggunakan Transformasi Bilinier yang mana didefinisikan pada Persamaan (3.64) [15].

$$s = \frac{2}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (3.64)$$

Subtitusi Persamaan (3.64) pada Persamaan (3.63) maka akan didapatkan Persamaan (3.65).

$$\frac{Y(z)}{U(z)} = \frac{K}{\frac{2\tau}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) + 1} \quad (3.65)$$

$$Y(z) \left(\frac{2\tau}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) + 1 \right) = KU(z) \quad (3.66)$$

$$Y(z) + \frac{2\tau}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) Y(z) = KU(z) \quad (3.67)$$

$$(1 + z^{-1})Y(Z) + \frac{2\tau}{T_S}(1 - z^{-1})Y(Z) = K(1 - z^{-1})U(z) \quad (3.68)$$

$$\left(1 + \frac{2\tau}{T_S}\right)Y(z) + \left(1 - \frac{2\tau}{T_S}\right)z^{-1}Y(z) = KU(z) - Kz^{-1}U(z) \quad (3.69)$$

$$Y(z) = -\frac{\left(1 - \frac{2\tau}{T_S}\right)}{\left(1 + \frac{2\tau}{T_S}\right)}z^{-1}Y(z) + \frac{K}{\left(1 + \frac{2\tau}{T_S}\right)}U(z) - \frac{K}{\left(1 + \frac{2\tau}{T_S}\right)}z^{-1}U(z) \quad (3.70)$$

$$y(k) = -\frac{\left(1 - \frac{2\tau}{T_S}\right)}{\left(1 + \frac{2\tau}{T_S}\right)}y(k-1) + \frac{K}{\left(1 + \frac{2\tau}{T_S}\right)}u(k) - \frac{K}{\left(1 + \frac{2\tau}{T_S}\right)}u(k-1) \quad (3.71)$$

Secara umum bentuk model dengan struktur ARX(1,1) dapat dinyatakan dengan Persamaan (3.72).

$$y(k) = -a_1y(k-1) + b_0u(k) + b_1u(k-1) + e(k) \quad (3.72)$$

Persamaan (3.72) memiliki struktur yang sama dengan model *plant* yang telah diubah ke bentuk diskrit yaitu Persamaan (3.71) bila ditambahkan komponen *noise* $e(k)$. Dengan demikian dapat diformulasikan pada persamaan-persamaan berikut

$$a_1 = -\frac{\left(1 - \frac{2\tau}{T_S}\right)}{\left(1 + \frac{2\tau}{T_S}\right)} \quad (3.73)$$

$$b_0 = \frac{K}{\left(1 + \frac{2\tau}{T_S}\right)} \quad (3.74)$$

$$b_1 = -\frac{K}{\left(1 + \frac{2\tau}{T_S}\right)} \quad (3.75)$$

Persamaan (3.73) dapat digunakan untuk menurunkan nilai *time constant* (τ) yang dinyatakan pada Persamaan (3.79).

$$a_1 \left(1 + \frac{2\tau}{T_S}\right) = -\left(1 - \frac{2\tau}{T_S}\right) \quad (3.76)$$

$$a_1 + \frac{2a_1\tau}{T_S} = -1 + \frac{2\tau}{T_S} \quad (3.77)$$

$$a_1 + 1 = \frac{2}{T_S}(1 - a_1)\tau \quad (3.78)$$

$$\tau = \frac{a_1 + 1}{\frac{2}{T_S}(1 - a_1)} \quad (3.79)$$

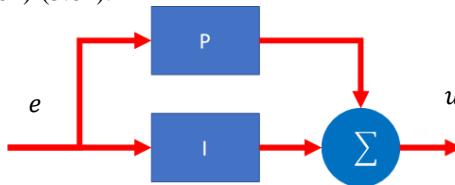
Persamaan (3.74) dapat digunakan untuk menentukan nilai *gain overall* dari *plant* yang dinyatakan pada Persamaan (3.80).

$$K = b_0 \left(1 + \frac{2\tau}{T_S}\right) \quad (3.80)$$

Subtitusikan Persamaan (3.79) pada Persamaan (3.80) maka akan didapatkan Persamaan (3.81).

$$K = b_0 \left(1 + \frac{2}{T_S} \left(\frac{a_1 + 1}{\frac{2}{T_S} (1 - a_1)} \right) \right) \quad (3.81)$$

Konfigurasi kontroler yang dipakai pada desain *self-tunning regulator* adalah kontroler PI yang dinyatakan dengan fungsi alih pada pada Persamaan (3.82). Dimana $U(s)$ menyatakan sinyal kontrol dan $E(s)$ menyatakan sinyal eror. Persamaan kontroler dapat dinyatakan pada Persamaan (3.82)-(3.84).



Gambar 3.13 Kontroler PI

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} \right) \quad (3.82)$$

$$\frac{U(s)}{E(s)} = \frac{K_p \tau_i s}{\tau_i s} + \frac{K_p}{\tau_i s} \quad (3.83)$$

$$\frac{U(s)}{E(s)} = \frac{K_p \tau_i s + K_p}{\tau_i s} \quad (3.84)$$

Persamaan (3.84) diubah ke bentuk persamaan diskrit dengan menggunakan Transformasi Bilinier dimana hasilnya dinyatakan pada Persamaan (3.90).

$$\frac{U(z)}{E(z)} = \frac{K_p \tau_i \frac{2}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) + K_p}{\tau_i \frac{2}{T_S} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)} \quad (3.85)$$

$$\frac{U(z)}{E(z)} = \frac{K_p \tau_i \frac{2}{T_S} (1 - z^{-1}) + K_p (1 + z^{-1})}{\tau_i \frac{2}{T_S} (1 - z^{-1})} \quad (3.86)$$

$$\left(\tau_i \frac{2}{T_S} (1 - z^{-1}) \right) U(z) = \left(\begin{matrix} K_p \tau_i \frac{2}{T_S} (1 - z^{-1}) \\ K_p (1 + z^{-1}) \end{matrix} \right) E(z) \quad (3.87)$$

$$\begin{pmatrix} \tau_i \frac{2}{T_S} U(z) - \\ \tau_i \frac{2}{T_S} z^{-1} U(z) \end{pmatrix} = \begin{pmatrix} K_p \tau_i \frac{2}{T_S} E(z) - K_p \tau_i \frac{2}{T_S} z^{-1} E(z) \\ + K_p E(z) + K_p z^{-1} E(z) \end{pmatrix} \quad (3.88)$$

$$U(z) = \begin{pmatrix} \frac{\tau_i \frac{2}{T_S} z^{-1}}{\tau_i \frac{2}{T_S}} U(z) + \frac{(K_p \tau_i \frac{2}{T_S} + K_p)}{K_p \tau_i} E(z) \\ + \frac{(K_p - K_p \tau_i \frac{2}{T_S}) z^{-1}}{K_p \tau_i} E(z) \end{pmatrix} \quad (3.89)$$

$$u(k) = \begin{pmatrix} u(k-1) + \frac{(K_p \tau_i \frac{2}{T_S} + K_p)}{K_p \tau_i} e(k) \\ + \frac{(K_p - K_p \tau_i \frac{2}{T_S})}{K_p \tau_i} e(k-1) \end{pmatrix} \quad (3.90)$$

Setelah konfigurasi kontroler telah diformulasikan pada Persamaan (3.90) dapat dilakukan perhitungan parameter kontroler PI. Pendekatan analitik digunakan untuk mencari parameter-parameter kontroler yang mana jika diinginkan eror *steady state* adalah nol maka nilai parameter τ_i adalah sama dengan nilai *time constant* dari sistem *open loop* yang diformulasikan pada Persamaan (3.91).

$$\tau_i = \tau \quad (3.91)$$

Dimana nilai konstanta waktu sistem *open loop* dinyatakan pada Persamaan (3.79), dengan demikian maka akan didapatkan persamaan τ_i dalam fungsi parameter model sistem yang mana dinyatakan di Persamaan (3.92).

$$\tau_i = \frac{a_1 + 1}{\frac{2}{T_S}(1 - a_1)} \quad (3.92)$$

Nilai konstanta proporsional kontroler dapat dicari dengan Persamaan (3.93).

$$K_p = \frac{\tau_i}{\tau^* K} \quad (3.93)$$

Persamaan (3.93) dapat diubah dengan melakukan substitusi Persamaan (3.81) dan (3.92) pada persamaan tersebut untuk mendapatkan hubungan antara nilai konstanta proporsional terhadap nilai parameter model sistem.

$$K_p = \frac{\left(\frac{a_1 + 1}{\frac{2}{Ts}(1 - a_1)} \right)}{\tau^* \left(b_0 \left(1 + \frac{2}{Ts} \left(\frac{a_1 + 1}{\frac{2}{Ts}(1 - a_1)} \right) \right) \right)} \quad (3.94)$$

Dimana τ^* spesifikasi *time constant* respon sistem *closed loop* yang dikehendaki. Nilai spesifikasi τ^* yang dikehendaki pada perancangan ini adalah 50 detik.

[Halaman ini sengaja dikosongkan]

BAB 4

PENGUJIAN DAN ANALISA

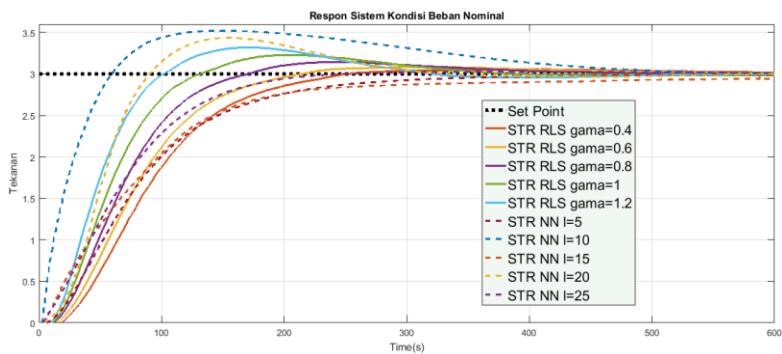
Pada bagian ini akan dibahas mengenai hasil pengujian sistem yang telah dirancang sebelumnya pada Bab 3 untuk mengetahui performa dari *plant* apabila diterapkan metode *self-tunning regulator* dengan estimasi parameter *recursive least square* (STR RLS) dan *self-tunning regulator* dengan estimasi parameter pendekatan jaringan syaraf tiruan (STR NN). Pengujian dibagi menjadi tiga bagian yaitu pengujian pada kondisi beban nominal dan pengujian dengan adanya perubahan beban serta pengujian terhadap variasi nilai *set point*. Nilai RMSE, ITSE, dan ITAE serta ISTSE dari sistem dipakai sebagai kriteria pembanding, kemudian tak lupa melihat performansi lain sistem yang telah dirancang seperti *settling time* (t_s) dan persentase *overshoot* serta persentase *peak* dan *recovery time*.

4.1 Pengujian Beban Nominal

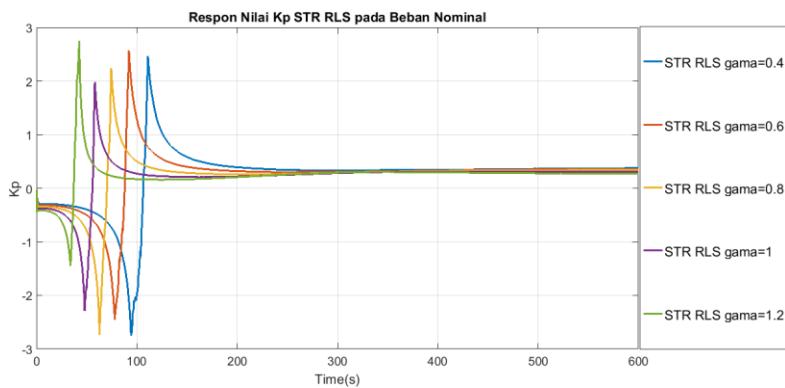
Pengujian dilakukan untuk mengetahui peforma sistem ketika tidak terjadi perubahan beban. Kondisi beban nominal sistem adalah sebagai berikut:

- a) Set point : 15,2 mA (3 psi)
- b) Jumlah data yang diambil : 2000 sampel
- c) Kondisi nominal :
 - i. Katup terbuka : V1, V3, V4, dan V5
 - ii. Katup tertutup : V2 dan V6
- d) Air receiver : Digunakan
- e) Periode sampling : 300 ms
- f) Initial condition output : 0 psi

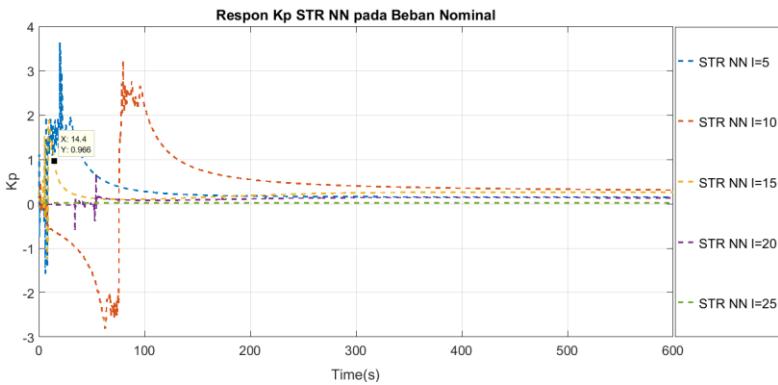
Respon sistem hasil implementasi terhadap metode yang diuji pada keadaan beban nominal ditunjukan pada Gambar 4.1. Tabel 4.1 dan Tabel 4.2 menunjukkan performa dari sistem hasil desain pada kondisi beban nominal. Gambar 4.2-Gambar 4.7 menggambarkan respon nilai parameter *gain propotional* K_p dan T_i pada kondisi beban nominal. Tabel 4.3-Tabel 4.6 menampilkan tabel performa sistem yang telah dinormalisasi dan diberikan bobot.



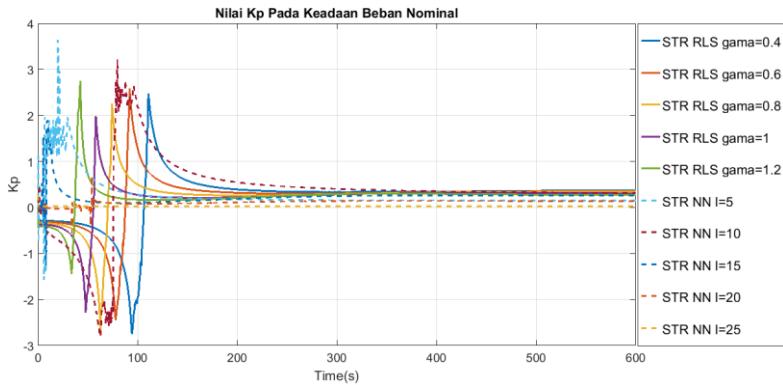
Gambar 4.1 Respon sistem kondisi beban nominal



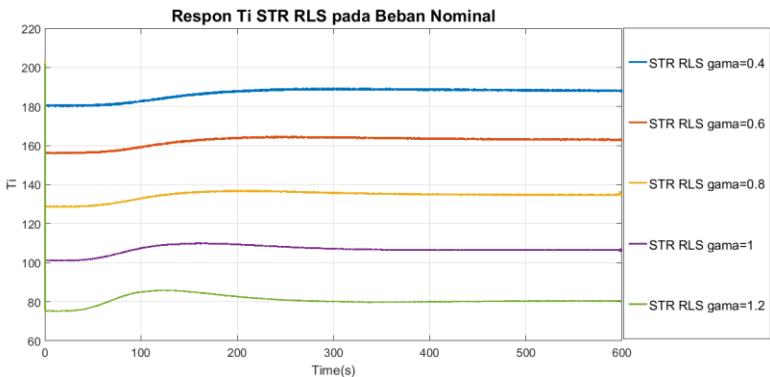
Gambar 4.2 Respon nilai K_p STR RLS pada kondisi beban nominal



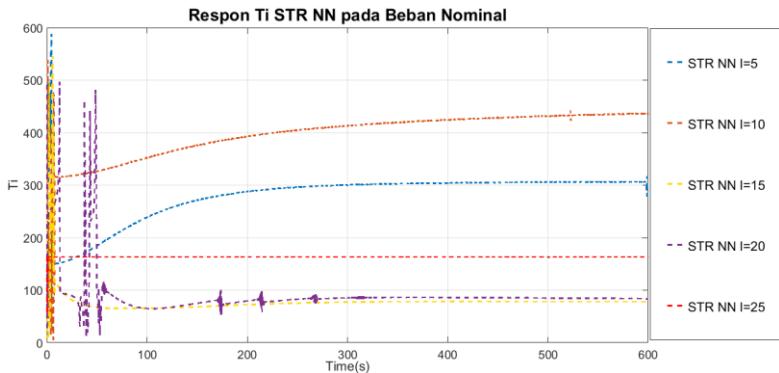
Gambar 4.3 Respon nilai K_p STR NN pada kondisi beban nominal



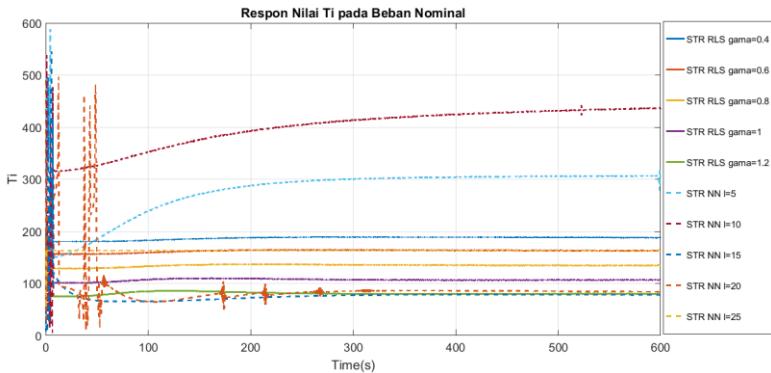
Gambar 4.4 Respon nilai K_p pada kondisi beban nominal



Gambar 4.5 Respon Ti STR RLS pada kondisi beban nominal



Gambar 4.6 Respon nilai Ti STR NN pada kondisi beban nominal



Gambar 4.7 Respon nilai Ti pada kondisi beban nominal

Tabel 4.1 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
RMSE	0,4915	0,4398	0,3869	0,3496	0,3170
ITSE	$2,4329 \times 10^4$	$1,8972 \times 10^4$	$1,4222 \times 10^4$	$1,1024 \times 10^4$	$0,8505 \times 10^4$
ITAE	$2,2559 \times 10^4$	$1,9350 \times 10^4$	$1,6009 \times 10^4$	$1,5062 \times 10^4$	$1,4673 \times 10^4$
ISTSE	$1,6779 \times 10^6$	$1,1999 \times 10^6$	$0,8395 \times 10^6$	$0,6857 \times 10^6$	$0,6024 \times 10^6$
%overshot	1,9000	2,8667	4,8333	7,6000	10,7000
ts($\pm 2.5\%$)	222,9s	355,5s	340,2s	319,2s	285,6s

Tabel 4.2 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal

STR NN	$L=5$	$L=10$	$L=15$	$L=20$	$L=25$
RMSE	0,4675	0,3376	0,4693	0,3464	0,3643
ITSE	$2,0289 \times 10^4$	$1,1630 \times 10^4$	$1,8248 \times 10^4$	$1,0572 \times 10^4$	$1,2867 \times 10^4$
ITAE	$2,2145 \times 10^4$	$3,1680 \times 10^4$	$2,9760 \times 10^4$	$1,6745 \times 10^4$	$1,4037 \times 10^4$
ISTSE	$1,4992 \times 10^6$	$2,3769 \times 10^6$	$1,7444 \times 10^6$	$0,8345 \times 10^6$	$0,7506 \times 10^6$
%overshot	0,0000	17,3667	0,0000	14,5333	0,9000

ts($\pm 2\%$)	297,6s	452,1s	476,1s	284,7s	192,6s
-----------------	--------	--------	--------	--------	--------

Dari semua kriteria yang dipakai %overshot dan time settling (ts) memiliki prioritas yang lebih tinggi dari kreiteria yang lain. Hal ini dipilih karena %overshot mempengaruhi keamanan dari sistem, sedang ts merupakan menyatakan seberapa dekat respon dengan spesifikasi desain yang dirancang. RMSE, ITSE, ITAE, dan ISTSE dipakai sebagai kriteria pendukung untuk melakukan pertimbangannya. Semua kriteria yang ada diberikan akan bobot. Sebelum itu Tabel 4.1 dan Tabel 4.2 pada baris *time settling* diubah ke bentuk absolut simpangan dari nilai desain yang mana dinyatakan pada Persamaan (4.1). Kemudian kedua table digabungkan kemudian dinormalisasi menggunakan aturan normalisasi *min max* yang dinyatakan pada Persamaan (4.2) [16], setelah diberikan bobot pada data hasil normalisasi. Aturan pemberian bobot dinyatakan pada Persamaan (4.3). Dimana x_N menyatakan data hasil normalisasi, x_{old} menyatakan data sebelum dinormalisasi, x_{max} nilai data terbesar pada baris tabel tersebut, $x_{w_{RMSE}}$ adalah data yang telah diberi bobot kriteria RMSE, $x_{w_{ITSE}}$ adalah data yang telah diberi bobot kriteria ITSE, $x_{w_{ITAE}}$ merupakan data yang telah diberi bobot kriteria ITAE, $x_{w_{ISTSE}}$ merupakan data yang telah diberi bobot kriteria, $x_{w_{simpangan\;t_s}}$ menyatakan data yang telah diberi bobot kriteria ISTSE, dan adalah data yang telah diberi bobot kriteria simpangan t_s , serta $x_{w_{overshoot}}$ adalah data yang telah diberi bobot kriteria overshoot.

$$\text{simpangan } t_s = |t_s \text{ respon} - t_s \text{ desain}| \quad (4.1)$$

$$x_N = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \quad (4.2)$$

$$x_{w_{RMSE}} = 0,1 \times x_N$$

$$x_{w_{ITSE}} = 0,1 \times x_N$$

$$x_{w_{ITAE}} = 0,1 \times x_N$$

$$x_{w_{ISTSE}} = 0,1 \times x_N$$

$$x_{w_{overshoot}} = 0,3 \times x_N$$

$$x_{w_{simpangan\;t_s}} = 0,3 \times x_N$$

$$(4.3)$$

Pada kondisi beban nominal *self-tunning regulator* dengan estimasi parameter *recursice least square* menunjukkan bahwa STR dengan *forgetting factor* = 1.2 memiliki nilai RMSE, ITSE, ITAE, ISTSE yang paling kecil, sedangkan pada *self-tunning regulator* dengan estimasi parameter pendekatan jaringan syaraf tiruan menunjukkan, nilai RMSE

terkecil dimiliki oleh STR NN dengan *learning rate* = 10, nilai ITSE paling kecil dimiliki oleh STR NN dengan *learning rate* = 20, serta nilai ITAE dan ISTSE terkecil dimiliki oleh STR NN dengan *learning rate* = 25. Secara keseluruhan pada pengujian STR pada kondisi beban nominal STR RLS dengan *forgetting factor* = 1,2 menunjukkan nilai RMSE ITSE dan ISTSE paling kecil, sementara STR dengan pendekatan jaringan syaraf tiruan dengan *learning rate* = 25 memeliki nilai ITAE terendah.

Nilai *overshoot* terendah ditunjukan pada STR NN dengan *learning rate* 5, 15 dan 25 dengan nilai persentase *overshoot* secara berturut adalah 0,0000, 0,0000 dan 0,9000. Overshoot yang terbesar terjadi pada STR NN dengan *learning rate* 10 dan 20 serta STR RLS dengan *forgetting factor* 1,2 dengan nilai secara berturut-turut adalah 17,3667%, dan 10,7000%. Bila dikaji pada STR RLS saja *overshoot* terendah dimiliki oleh STR RLS dengan *forgetting factor* 0,4. Pada STR RLS *overshoot* semakin besar sebanding terhadap nilai besarnya *forgetting factor*.

Spesifikasi desain nilai konstanta waktu (τ^*) pada subbab 3.7 adalah 50 detik. Dengan demikian apabila dipakai kriteria ($\pm 2\%$) *settling time* (ts) hasil desain adalah $ts^* = 4\tau^* = 200$ detik, nilai ts hasil pengujian yang paling mendekati hasil desain adalah STR NN dengan *learning rate* = 25 yang mana nilainya 192,6 detik dan STR RLS dengan *forgetting factor* = 0,4 yang mana nilainya 222,9 detik. Secara keseluruhan nilai t_s paling terendah dimiliki oleh STR NN dengan *learning rate* = 25 sedangkan jika pada STR RLS nilai ts paling rendah dimiliki oleh STR RLS dengan *forgetting factor* = 0,4.

Tabel 4.3 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal dengan normalisasi

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
$x_{N_{RMSE}}$	1	0,703725	0,400573	0,186819	0
$x_{N_{ITSE}}$	1	0,661464	0,361287	0,159189	0
$x_{N_{ITAE}}$	0,48302	0,301139	0,111772	0,058097	0,036048
$x_{N_{ISTSE}}$	0,60608	0,336715	0,133615	0,046943	0
$x_{N_{overshoot}}$	0,10940	0,165069	0,278308	0,437619	0,616122
$x_{N_{simpangan\;ts}}$	0,10687	0,574603	0,520635	0,446561	0,328042

Tabel 4.4 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal dengan normalisasi

STR NN	L=5	L=10	L=15	L=20	L=25
x_{NRMSE}	0,862464	0,118052	0,872779	0,168481	0,27106
x_{NITSE}	0,744692	0,197485	0,61571	0,130624	0,275657
x_{WITAE}	0,459559	1	0,891175	0,153489	0
x_{WISTSE}	0,505382	1	0,643562	0,130797	0,083516
$x_{Wovershoot}$	0	1	0	0,836849	0,051823
$x_{wsimpangan t_s}$	0,37037	0,915344	1	0,324868	0

Tabel 4.5 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal dengan pembobotan

STR RLS	g=0,4	g=0,6	g=0,8	g=1	g=1,2
x_{wRMSE}	0,1	0,070372	0,040057	0,018682	0
x_{wITSE}	0,1	0,066146	0,036129	0,015919	0
x_{WITAE}	0,048302	0,030114	0,011177	0,00581	0,003605
x_{WISTSE}	0,060609	0,033671	0,013362	0,004694	0
$x_{wovershoot}$	0,032821	0,049521	0,083493	0,131286	0,184836
$x_{wsimpangan t_s}$	0,032063	0,172381	0,15619	0,133968	0,098413
Bobot total	0,373796	0,422206	0,340408	0,310359	0,286854

Tabel 4.6 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal dengan pembobotan

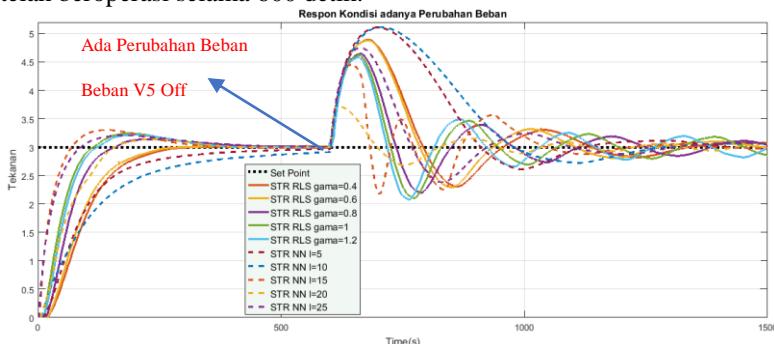
STR NN	L=5	L=10	L=15	L=20	L=25
x_{wRMSE}	0,086246	0,011805	0,087278	0,016848	0,027106
x_{wITSE}	0,074469	0,019748	0,061571	0,013062	0,027566
x_{WITAE}	0,045956	0,1	0,089117	0,015349	0
x_{WISTSE}	0,050538	0,1	0,064356	0,01308	0,008352
$x_{wovershoot}$	0	0,3	0	0,251055	0,015547
$x_{wsimpangan t_s}$	0,111111	0,274603	0,3	0,09746	0
Bobot total	0,368321	0,806157	0,602323	0,406854	0,07857

4.2 Pengujian dengan Perubahan Beban

Permasalahan yang disoroti pada bagian ini adalah mengetahui performa dari sistem ketika terjadi perubahan beban. Kondisi mula-mula dari sistem adalah sebagai berikut:

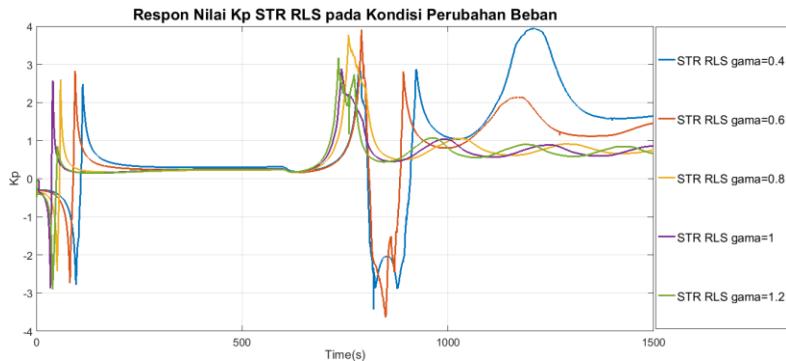
- a) Set point : 15,2 mA (3 psi)
- b) Jumlah data yang diambil : 5000 sampel
- c) Kondisi nominal :
 - i. Katup terbuka : V1, V3, V4, dan V5
 - ii. Katup tertutup : V2 dan V6
- d) Air receiver : Digunakan
- e) Periode sampling : 300 ms
- f) Initial condition output : 0 psi

Perubahan beban dilakukan dengan cara menonaktifkan salah satu satu *manual valve*. *Manual valve* V5 dinonaktifkan atau ditutup ketika sistem telah beroperasi selama 600 detik.

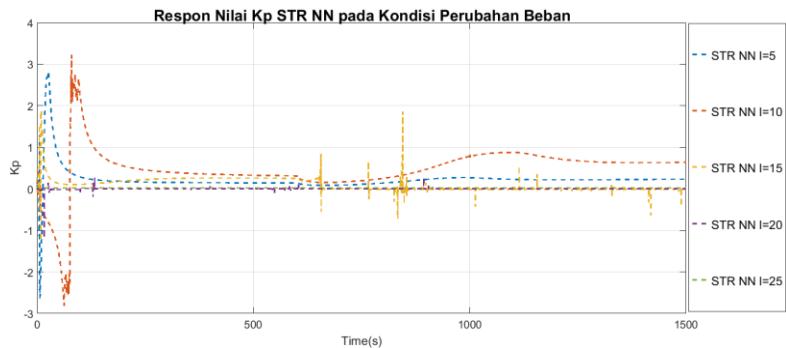


Gambar 4.8 Respon sistem kondisi adanya perubahan beban

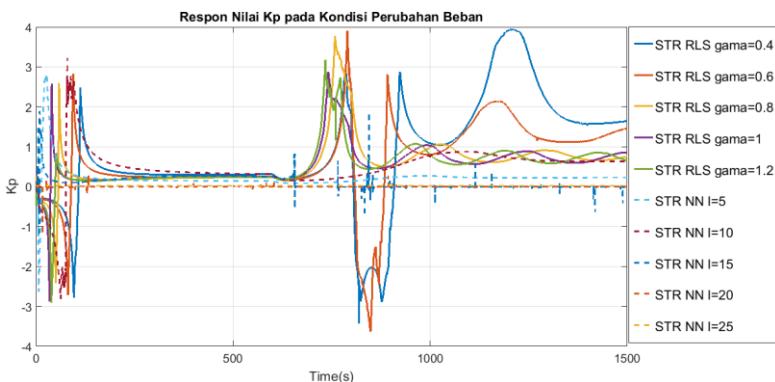
Respon sistem hasil implementasi terhadap metode yang diuji pada keadaan adanya perubahan beban ditunjukkan pada Gambar 4.8. Tabel 4.7 dan Tabel 4.8 menunjukkan performa sistem dari pengujian yang dilakukan pada keadaan adanya perubahan beban. Gambar 4.9-Gambar 4.14 menggambarkan dari respon nilai K_p dan T_i yang diperoleh. Tabel 4.9-Tabel 4.12 menyajikan tabel performa sistem yang telah dinormalisasi serta diberikan bobot kriteria.



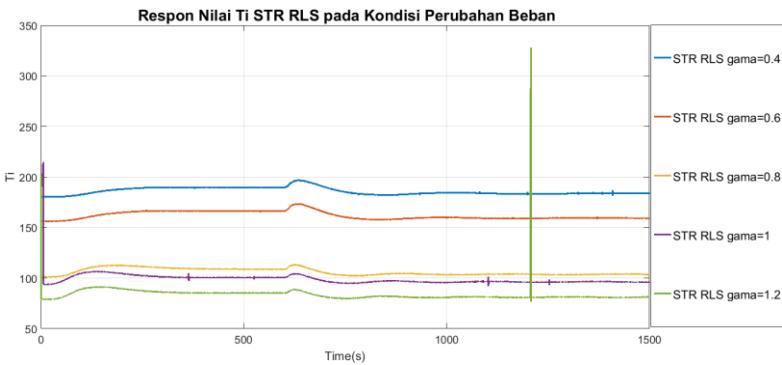
Gambar 4.9 Respon nilai K_p STR RLS pada kondisi perubahan beban



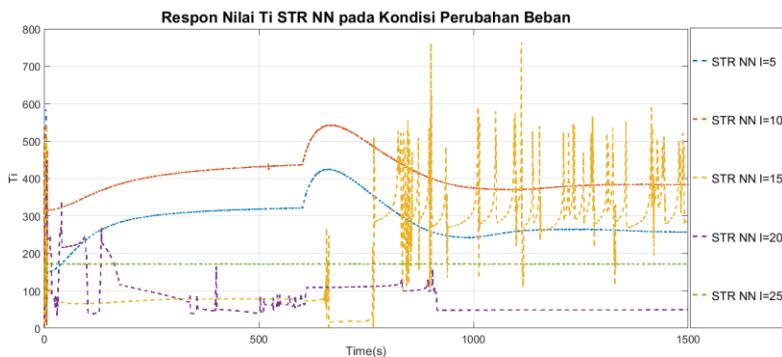
Gambar 4.10 Respon K_p STR NN pada kondisi perubahan beban



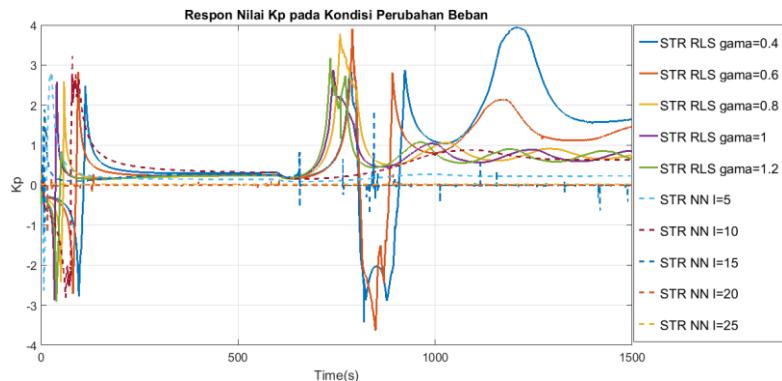
Gambar 4.11 Respon nilai K_P pada kondisi perubahan beban



Gambar 4.12 Respon nilai T_i STR RLS pada kondisi perubahan beban



Gambar 4.13 Respon Ti STR NN pada kondisi perubahan beban



Gambar 4.14 Respon nilai Ti pada kondisi perubahan beban

Tabel 4.7 Performa sistem hasil pengujian STR dengan estimasi parameter *recursice least square* pada kondisi adanya perubahan beban

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
RMSE	0,4401	0,4178	0,3363	0,3133	0,3163
ITSE	$3,0450 \times 10^5$	$2,8968 \times 10^5$	$1,8187 \times 10^5$	$1,7260 \times 10^5$	$1,6498 \times 10^5$
ITAE	$3,1020 \times 10^5$	$2,9710 \times 10^5$	$2,5068 \times 10^5$	$2,5650 \times 10^5$	$2,5501 \times 10^5$
ISTSE	$2,0299 \times 10^8$	$1,9408 \times 10^8$	$1,2369 \times 10^8$	$1,2267 \times 10^8$	$1,1721 \times 10^8$
% Peak	63,0333	62,7667	54,9333	54,1667	52,9333

Recovery time $(\pm 5\%)$	600,5s	617s	721s	811s	865s
------------------------------	--------	------	------	------	------

Tabel 4.8 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban

STR NN	L=5	L=10	L=15	L=20	L=25
RMSE	0,5341	0,6172	0,2371	0,2043	0,2794
ITSE	$5,3761 \times 10^5$	$6,2660 \times 10^5$	$1,2528 \times 10^5$	$0,3281 \times 10^5$	$2,0671 \times 10^5$
ITAE	$4,1076 \times 10^5$	$4,7990 \times 10^5$	$1,9943 \times 10^5$	$1,0000 \times 10^5$	$2,4580 \times 10^5$
ISTSE	$3,7918 \times 10^8$	$4,4754 \times 10^8$	$0,9044 \times 10^8$	$0,1686 \times 10^8$	$1,4395 \times 10^8$
% Peak	70,2000	70,4333	48,5667	23,7000	5,.3667
Recovery time $(\pm 5\%)$	471s	589s	407,5s	283,8s	449,5s

Pada pengujian ini STR NN dengan *learning rate* = 20 memiliki nilai RMSE, ITAE, dan ITAE serta ISTSE terendah. Apabila hanya dikaji pada STR RLS nilai RMSE terendah dimiliki oleh STR RLS dengan *forgetting factor* = 1 serta nilai ITSE, ITAE, dan ISTSE terendah dimiliki oleh STR RLS dengan nilai *forgetting factor* = 1.2.

Untuk kondisi adanya perubahan beban nilai % peak terkecil dimiliki oleh STR NN dengan *learning rate* = 20. Jika hanya diperhatikan pada STR RLS nilai peak terkecil dimiliki oleh STR dengan *forgetting factor* = 1.2. Pada STR RLS untuk pengujian kondisi adanya perubahan beban nilai % peak semakin kecil dengan semakin besarnya nilai *forgetting factor*.

Nilai *recovery time* yang paling cepat diperoleh pada pengujian STR NN dengan *learning rate* = 20. Jika hanya dikaji pada STR RLS, nilai *recovery time* terendah dimiliki oleh STR RLS dengan *forgetting factor* = 0.4. Pada STR RLS semakin kecil nilai *forgetting factor* maka semakin cepat juga nilai *recovery time* dari sistem.

Prioritas utama pembobotan pada kriteria yang dipakai adalah % peak dan recovery time karena itu diberikan bobot yang lebih besar dari pada

yang. Pembobotan $\%peak$ dan $recovery\ time$ dinyatakan pada Persamaan (4.4).

$$\begin{aligned}x_{w_{peak}} &= 0.3 \times x_N \\x_{w_{recovery\ time}} &= 0.3 \times x_N\end{aligned}\quad (4.4)$$

Tabel 4.9 Performa sistem hasil pengujian STR dengan estimasi parameter *recursice least square* pada kondisi adanya perubahan beban dengan normalisasi

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
x_{NRMSE}	0,571083	0,517074	0,31969	0,263986	0,271252
x_{NITSE}	0,457552	0,432594	0,251032	0,23542	0,222587
x_{NITAE}	0,553304	0,518821	0,396631	0,411951	0,408028
x_{NISTSE}	0,432177	0,411489	0,24805	0,245681	0,233004
x_{Npeak}	0,841655	0,83595	0,668331	0,651927	0,625535
$x_{N_{recovery\ time}}$	0,544907	0,573297	0,752237	0,907089	1

Tabel 4.10 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban dengan normalisasi

STR NN	$L=5$	$L=10$	$L=15$	$L=20$	$L=25$
x_{NRMSE}	0,798741	1	0,079438	0	0,181884
x_{NITSE}	0,850132	1	0,155728	0	0,292864
x_{NITAE}	0,818005	1	0,261727	0	0,383785
x_{NISTSE}	0,841274	1	0,170846	0	0,295091
x_{Npeak}	0,995008	1	0,532098	0	0,741799
$x_{N_{recovery\ time}}$	0,322092	0,52512	0,212836	0	0,2851

Tabel 4.11 Performa sistem hasil pengujian STR dengan estimasi parameter *recursice least square* pada kondisi adanya perubahan beban dengan pembobotan

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
x_{wRMSE}	0,057108	0,051707	0,031969	0,026399	0,027125
x_{wITSE}	0,045755	0,043259	0,025103	0,023542	0,022259
x_{wITAE}	0,05533	0,051882	0,039663	0,041195	0,040803
x_{wISTSE}	0,043218	0,041149	0,024805	0,024568	0,0233
x_{wpeak}	0,252496	0,250785	0,200499	0,195578	0,18766

$x_{w\text{recovery time}}$	0,163472	0,171989	0,225671	0,272127	0,3
Bobot total	0,057108	0,051707	0,031969	0,026399	0,027125

Tabel 4.12 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi adanya perubahan beban dengan pembobotan

STR NN	L=5	L=10	L=15	L=20	L=25
x_{wRMSE}	0,079874	0,1	0,007944	0	0,018188
x_{wITSE}	0,085013	0,1	0,015573	0	0,029286
x_{wITAE}	0,0818	0,1	0,026173	0	0,038379
x_{wISTSE}	0,084127	0,1	0,017085	0	0,029509
$x_{wovershoot}$	0,298502	0,3	0,159629	0	0,22254
$x_{w\text{recovery time}}$	0,096628	0,157536	0,063851	0	0,08553
Bobot total	0,079874	0,1	0,007944	0	0,018188

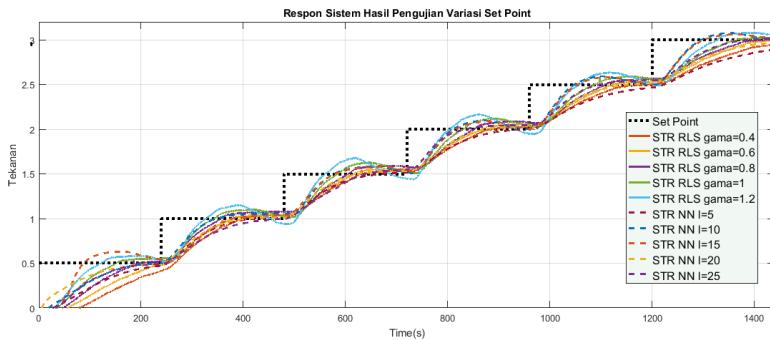
4.3 Pengujian dengan Variasi Set Point

Permasalahan yang diperhatikan pada bagian ini adalah mengetahui performa dari sistem ketika terjadi variasi *set point*. Kondisi awal dari sistem adalah sebagai berikut:

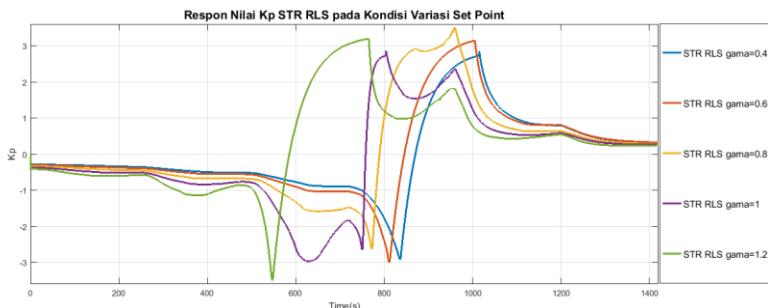
- a) *Set point* : 20 mA (0 psi)
- b) Jumlah data yang diambil : 4800 sampel
- c) Kondisi nominal :
 - i. Katup terbuka : V1, V3, V4, dan V5
 - ii. Katup tertutup : V2 dan V6
- d) *Air receiver* : Digunakan
- e) Periode *sampling* : 300 ms
- f) *Initial condition output* : 0 psi

Perubahan nilai *set point* yang akan diuji adalah pada nilai 0-3 psi dengan selang 0,5 psi. Perubahan *set point* dilakukan setiap 240 detik sekali.

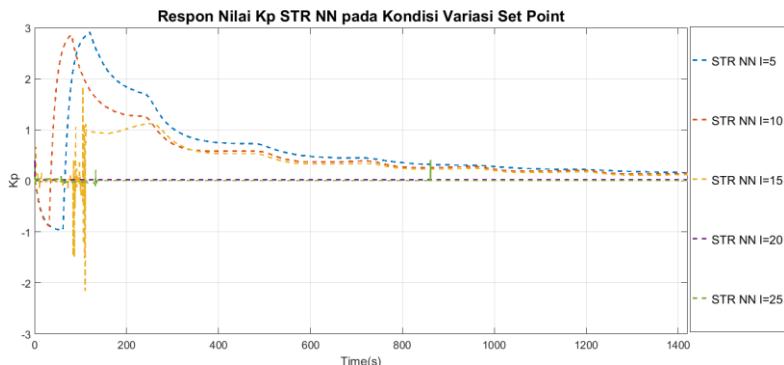
Respon sistem hasil implementasi terhadap metode yang diuji terhadap variasi *set-point* ditunjukkan pada Gambar 4.15. Tabel 4.13 dan Tabel 4.14 menunjukkan nilai RMSE dari tiap pengujian yang dilakukan terhadap variasi *set point*. Respon nilai parameter PI ditampilkan di Gambar 4.16-Gambar 4.21. Tabel 4.15-Tabel 4.18 menampilkan tabel performa sistem yang telah dinormalisasi serta diberikan bobot kriteria.



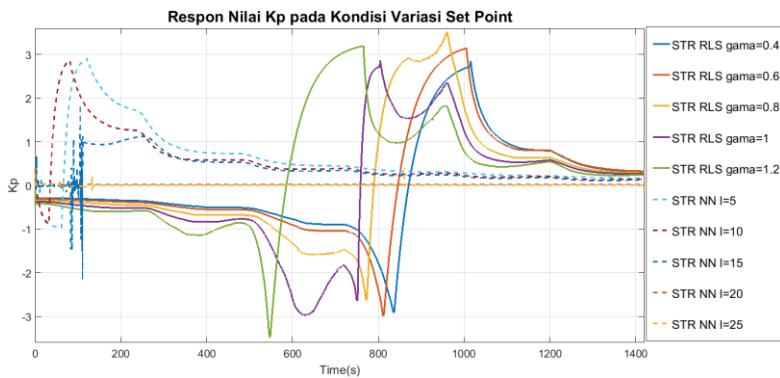
Gambar 4.15 Respon sistem kondisi variasi *set point*



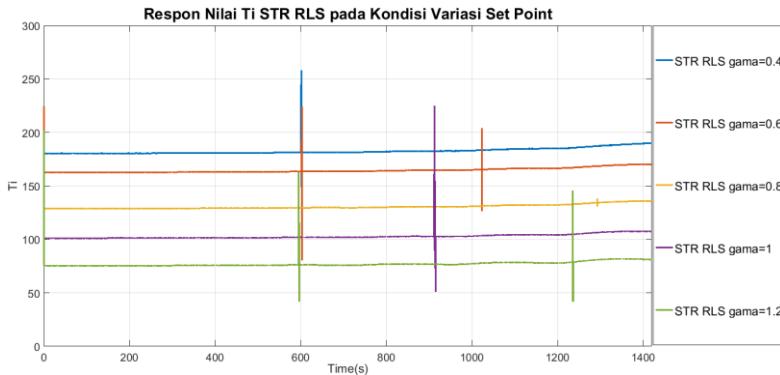
Gambar 4.16 Respon nilai Kp STR RLS pada kondisi variasi set point



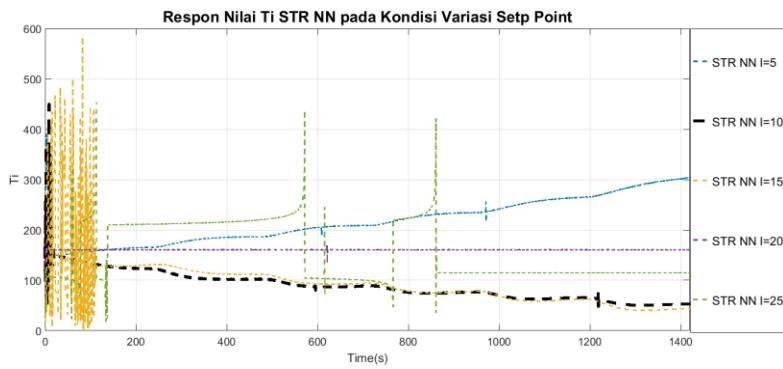
Gambar 4.17 Respon nilai Kp STR NN pada kondisi variasi set point



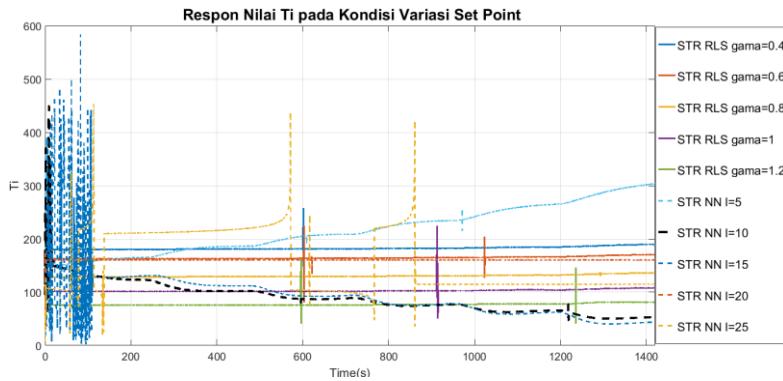
Gambar 4.18 Respon nilai K_p pada kondisi variasi set point



Gambar 4.19 Respon nilai Ti STR RLS pada kondisi variasi set point



Gambar 4.20 Respon nilai Ti STR NN pada kondisi variasi set point



Gambar 4.21 Respon Nilai Ti pada kondisi variasi set point

Tabel 4.13 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal terhadap variasi *set point*

STR RLS	$g=0,4$	$g=0,6$	$g=0,8$	$g=1$	$g=1,2$
RMSE	0,2199	0,1927	0,1797	0,1699	0,1673
ITSE	$6,399 \times 10^4$	$5,2842 \times 10^4$	$4,7868 \times 10^4$	$4,7504 \times 10^4$	$5,0267 \times 10^4$
ITAE	$1,9583 \times 10^5$	$1,7162 \times 10^5$	$1,6430 \times 10^5$	$1,5928 \times 10^5$	$1,6083 \times 10^5$
ISTSE	$5,9480 \times 10^8$	$4,8557 \times 10^8$	$4,3892 \times 10^8$	$4,3279 \times 10^8$	$4,5903 \times 10^8$

Tabel 4.14 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi *set point*

STR NN	L=5	L=10	L=15	L=20	L=25
RMSE	0,2031	0,1434	0,1487	0,1549	0,1602
ITSE	6,6964 $\times 10^4$	3,6656 $\times 10^4$	3,8044 $\times 10^4$	4,7108 $\times 10^4$	4,1377 $\times 10^4$
ITAE	2,0751 $\times 10^5$	1,2685 $\times 10^5$	1,3056 $\times 10^5$	1,5569 $\times 10^5$	1,3890 $\times 10^5$
ISTSE	6,6705 $\times 10^8$	3,2802 $\times 10^8$	3,3558 $\times 10^8$	4,4667 $\times 10^8$	3,6476 $\times 10^8$

Ketika pengujian variasi *set point* pada kondisi beban nominal STR NN dengan *learning rate* = 10 menunjukkan peforma terbaik dengan nilai RMSE, ITSE, dan ITAE serta ISTSE terendah. Sementara jika hanya diperhatikan pada STR RLS, nilai RMSE terendah dimiliki oleh STR RLS dengan *forgetting factor* = 1.2 serta nilai ITSE, ITAE, dan ISTSE terendah dimiliki oleh STR RLS dengan *forgetting factor* = 1. Pada STR RLS RMSE semakin kecil dengan semakin bertambahnya nilai *forgetting factor*.

Tabel 4.15 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal terhadap variasi *set point* dengan normalisasi

STR RLS	g=0,4	g=0,6	g=0,8	g=1	g=1,2
$x_{N_{RMSE}}$	1	0,644444	0,47451	0,346405	0,312418
$x_{N_{ITSE}}$	0,901874	0,53405	0,369935	0,357925	0,449089
$x_{N_{ITAE}}$	0,855195	0,555046	0,464295	0,402058	0,421274
$x_{N_{ISTSE}}$	0,786892	0,464708	0,32711	0,309029	0,386426

Tabel 4.16 Perfor,ma sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi *set point* dengan normalisasi

STR NN	L=5	L=10	L=15	L=20	L=25
$x_{N_{RMSE}}$	0,780392	0	0,069281	0,150327	0,219608
$x_{N_{ITSE}}$	1	0	0,045796	0,344859	0,155767
$x_{N_{ITAE}}$	1	0	0,045996	0,35755	0,149393

x_{NISTSE}	1	0	0,022299	0,349969	0,108368
--------------	---	---	----------	----------	----------

Tabel 4.17 Performa sistem hasil pengujian STR dengan estimasi parameter *recursive least square* pada kondisi beban nominal terhadap variasi *set point* dengan pembobotan

STR RLS	g=0,4	g=0,6	g=0,8	g=1	g=1,2
x_{WRMSE}	0,1	0,064444	0,047451	0,034641	0,031242
x_{WITSE}	0,090187	0,053405	0,036994	0,035793	0,044909
x_{WITAE}	0,085519	0,055505	0,046429	0,040206	0,042127
x_{WISTSE}	0,078689	0,046471	0,032711	0,030903	0,038643
Bobot total	0,354396	0,219825	0,163585	0,141542	0,156921

Tabel 4.18 Performa sistem hasil pengujian STR dengan estimasi parameter pendekatan jaringan syaraf tiruan pada kondisi beban nominal terhadap variasi *set point* dengan pembobotan

STR NN	L=5	L=10	L=15	L=20	L=25
x_{WRMSE}	0,078039	0	0,006928	0,015033	0,021961
x_{WITSE}	0,1	0	0,00458	0,034486	0,015577
x_{WITAE}	0,1	0	0,0046	0,035755	0,014939
x_{WISTSE}	0,1	0	0,00223	0,034997	0,010837
Bobot total	0,378039	0	0,018337	0,120271	0,063314

BAB 5

KESIMPULAN DAN SARAN

Bagian ini mengulas hasil penelitian yang telah dilakukan dan kendala yang dihadapi selama proses pengerjaan dimuat dalam kesimpulan dan saran.

5.1 Kesimpulan

Berdasarkan proses penelitian yang telah dilakukan pada tugas akhir ini, dapat disimpulkan beberapa hal, antara lain :

1. Pada keadaan beban nominal STR NN dengan *learning rate* = 25 menunjukkan performa yang paling mendekati hasil desain, dengan nilai *time settling* yang paling cepat dan *overshoot* yang kecil yaitu 0,9 %.
2. Pada kondisi adanya perubahan beban STR NN dengan *learning rate* = 20 menunjukkan performa terbaik pada semua kriteria yang dipakai yaitu RMSE, ITSE, ITAE, dan ISTSE serta dengan *% perturbation peak* = 23,7% dan *recovery time* = 283,8 detik.
3. Pada keadaan beban nominal dengan variasi *set-point* STR NN dengan *learning rate* = 10 menunjukkan performa terbaik untuk semua kriteria yang dipakai yaitu RMSE, ITSE, dan ITAE serta ISTSE.
4. Kedua estimator parameter dapat melakukan penaalaan pada STR

5.2 Saran

Beberapa saran yang perlu diberikan untuk penelitian selanjutnya antara lain :

1. Pada saat implementasi lebih baik dilakukan kalibrasi terlebih dahulu pada sensor dan aktuator
2. Perhitungan nilai *forgetting factor* dan *learning rate* yang lebih akurat sebaiknya dilakukan supaya memberikan hasil yang lebih baik
3. Penentuan frekuensi sampling harus dilakukan secara cermat agar sistem berjalanannya sebagaimana mestinya

[Halaman ini sengaja dikosongkan]

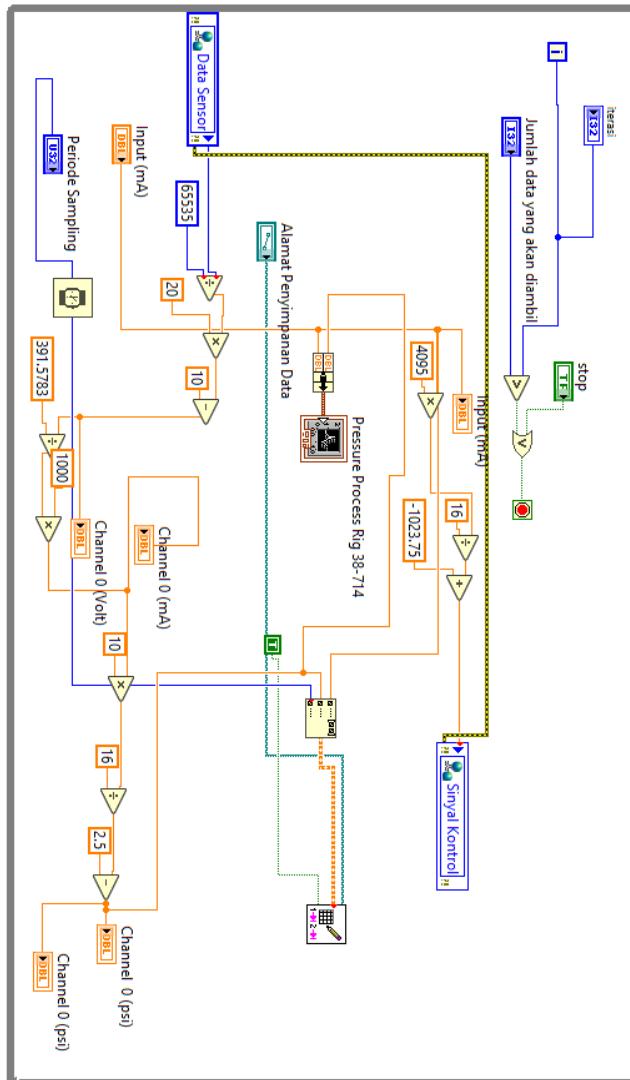
DAFTAR PUSTAKA

- [1] Massarudin, S. Nurjanah and R. B. Furner, “The Deepening of Structure of Industries Having Global Competitiveness : Competititveness In Automotive, Electronics & Electric and Heavy Equipment Industries,” pp. 1-1, 2007.
- [2] K. J. Astroom and B. Wittenmark, Adaptive Control, Second ed., Dover Publication, 1994.
- [3] T. D.A. and D. R.H., “Comparative Analysis of Concentration Control for Nonlinear Continuous Stirred Tank Reactor,” in *International Conference on Advanced in Computing Communication and Informatics(ICACCI)*, Jaipur, India, 2016.
- [4] Feedback Instrument Ltd, Process Control Trainer Level, Flow, & Temperature, Crowborough, pp. 1-1.
- [5] D. R. Patrick and S. W. Fardo, Industrial Process Control System, 2nd ed., CRC Press, 2009.
- [6] Feedback Instrument Ltd, Pressure Control Trainer Instruction Manual, Crowborough: FI Ltd.
- [7] K. J. Astrom and T. Hagglund, Advanced PID Control, United States of America: Instrumentation, Systems, and Automation Society, 2006.
- [8] J. Curtis D, “Data-Acquisition System,” in *Process Control Instrumentation Technology*, 7th ed., New Jersey, Hyderabad: Pearson Prentice Hall, 2016, pp. 153-156.
- [9] S. Haykin, “Introduction,” in *Neural Network and Learning Machine*, 3rd ed., New Jersey, Pearson, 2009, pp. 31-76.
- [10] National Instrument, LabVIEW User Manual, Austin, Texas: National Instrument.
- [11] Advantech, ADAM-5000/TCP Distributed DA&C System Based on Ethernet, Taiwan: Advantech Industrial Automation, 2001, pp. 1524-1530.
- [12] K. Ogata, Modern Control Engineering, 4th ed., Prentice Hall, 2002.

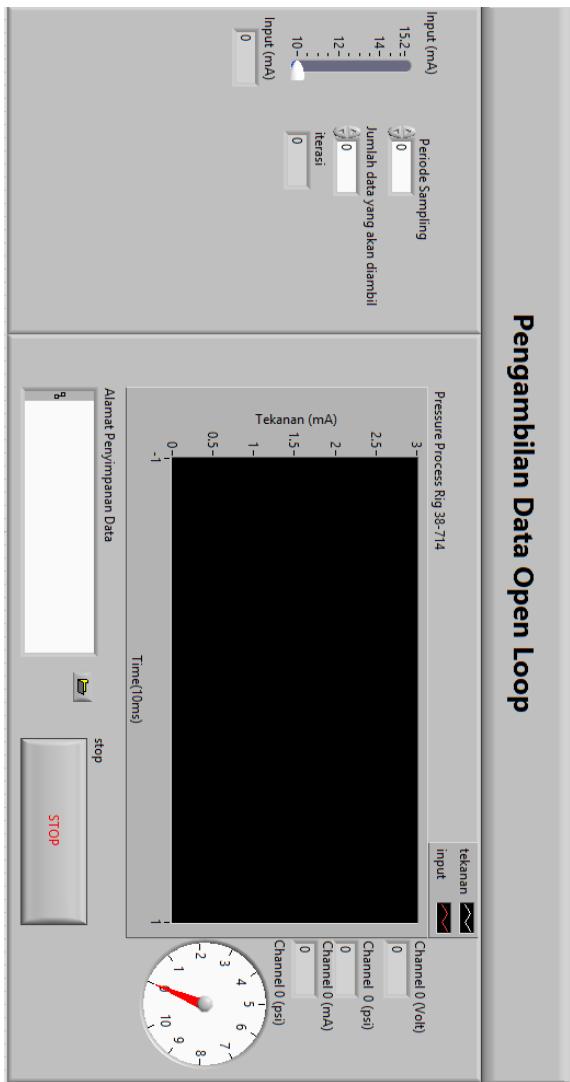
- [13] T. A. Tutunji, “Parametric System Identification Using Neural Networks,” *Applied Soft Computing*, pp. 2-4, 2016.
- [14] C. J. Zarowski, An Introduction to Numerical Analysis for Electrical and Computer Engineers, New Jersey: John Wiley & Sons., 2004, pp. 78-96.
- [15] A. V. Oppenheim, R. W. Schafer and J. R. Buck, Discrete Time Signal Processing, 2nd ed., New Jersey: Prentice Hall, 1999, pp. 450-454.
- [16] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2 ed., British Columbia: Morgan Kaufmann, 2006, p. 71.
- [17] M. B. A. M. J. O. A. Vasickaninova, “Fuzzy Controller Design for Heat Exchanger,” in *19th International Conference on Intelligent Engineering System*, Bratislava, Slovakia, 2015.

LAMPIRAN

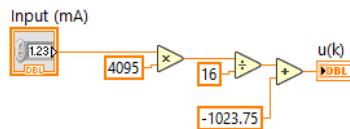
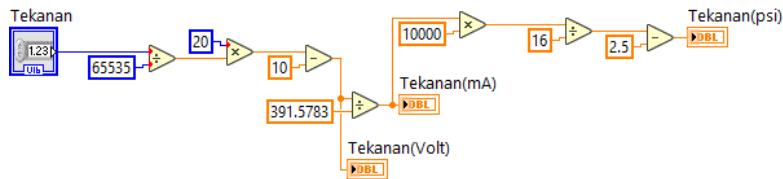
Lampiran 1. Program LabVIEW pengambilan data sistem open loop



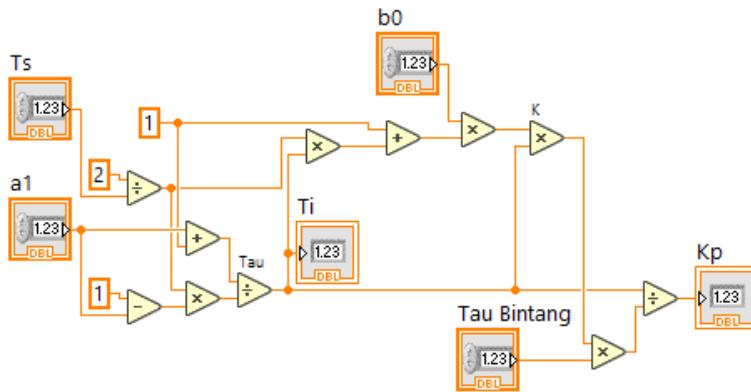
Lampiran 2. Tampilan *Human Machine Interface* (HMI) program pengambilan data sistem *open loop*



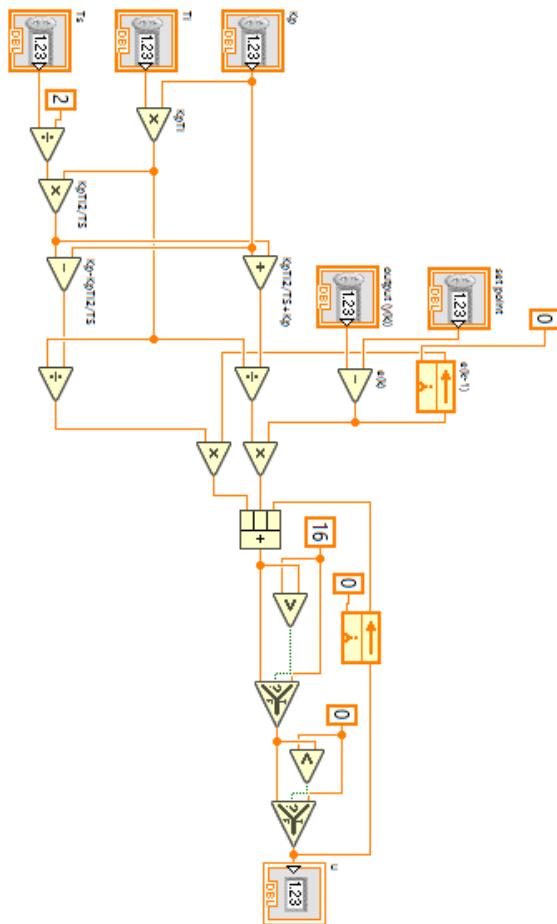
Lampiran 3. Program LabVIEW penyekalaan *input output* sistem pengakuisision data



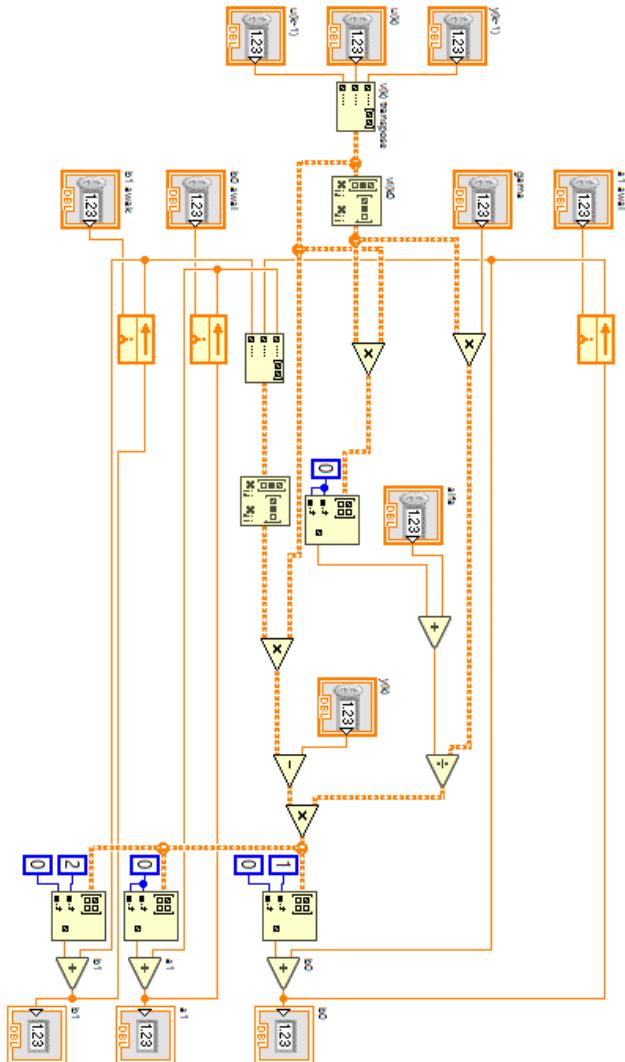
Lampiran 4. Program LabVIEW mekanisme *tunning* parameter PI



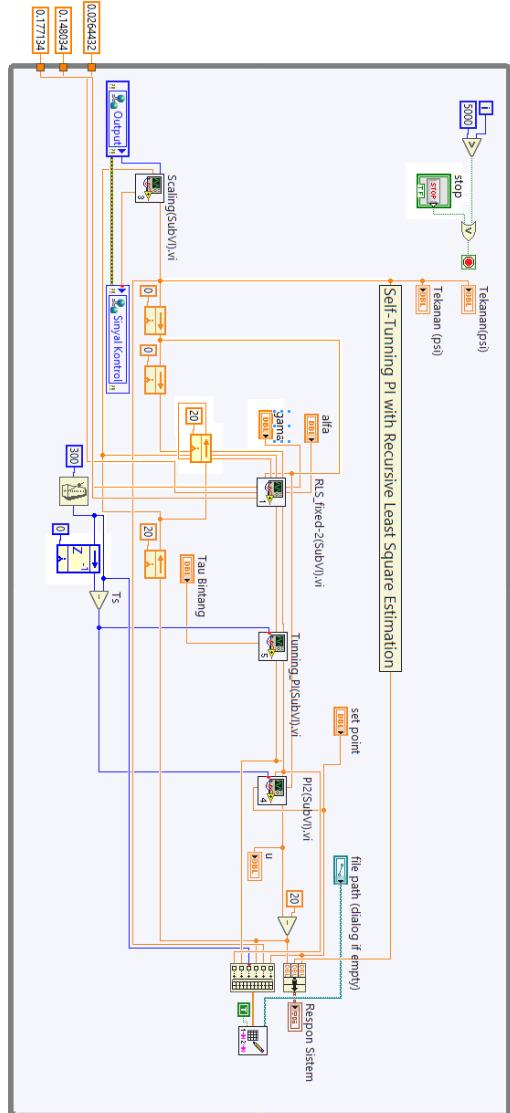
Lampiran 5. Program LabVIEW kontroler PI diskrit



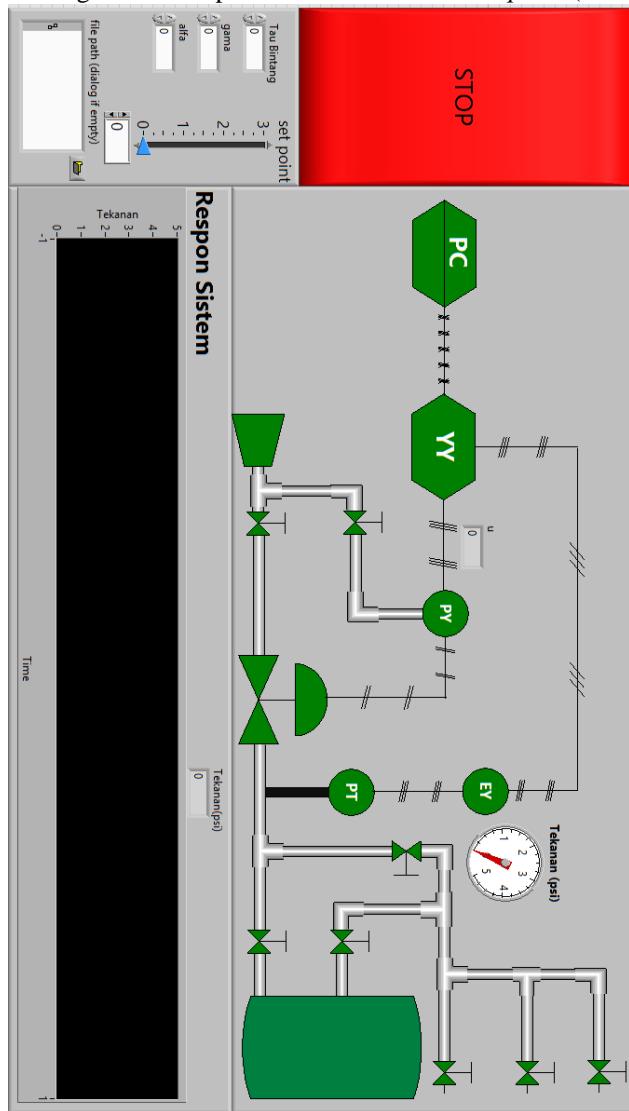
Lampiran 6. Program LabVIEW estimasi parameter *recursive least square*



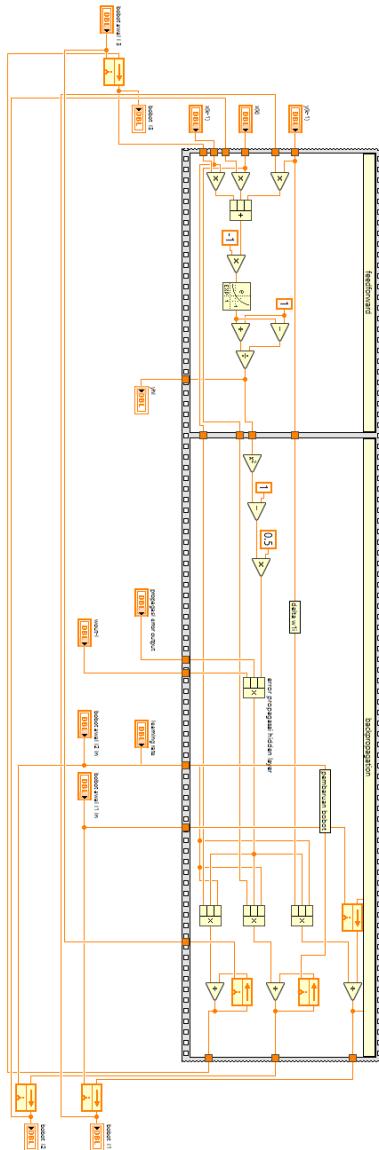
Lampiran 7. Program LabVIEW *self-tunning regulator* dengan estimasi parameter recursive least square (RLS)



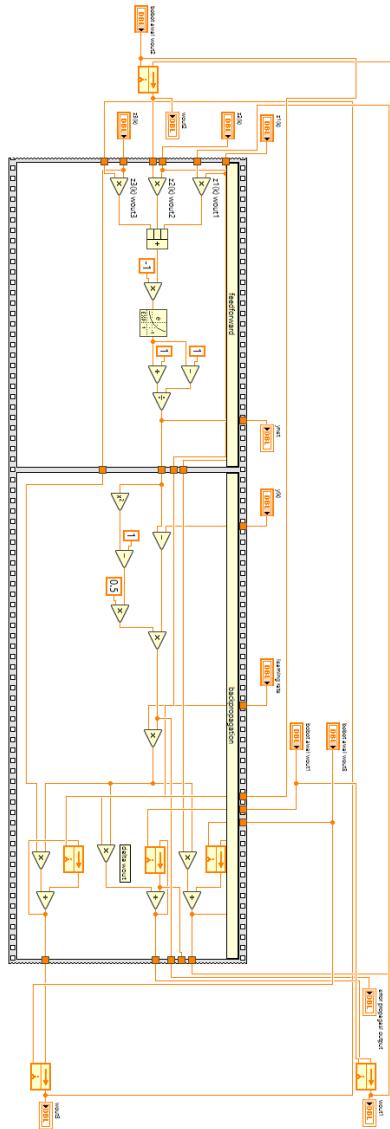
Lampiran 8. Human Machine Interface (HMI) program *self-tunning regulator* dengan estimasi parameter recursive least square (RLS)



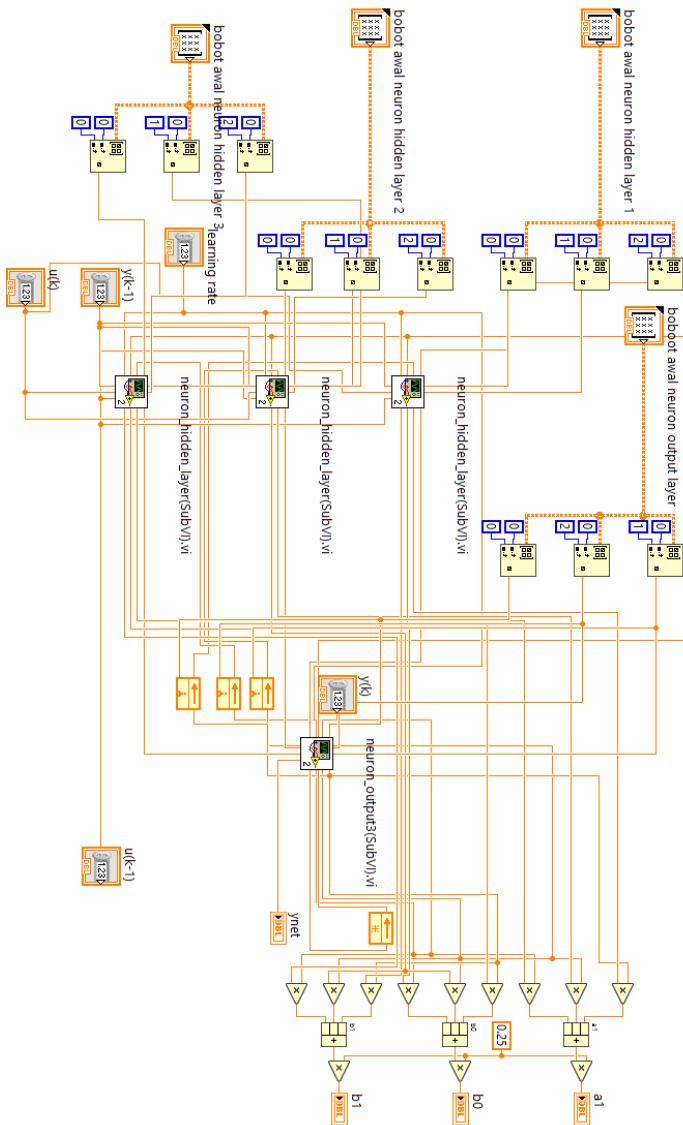
Lampiran 9. Program LabVIEW neuron *hidden-layer*



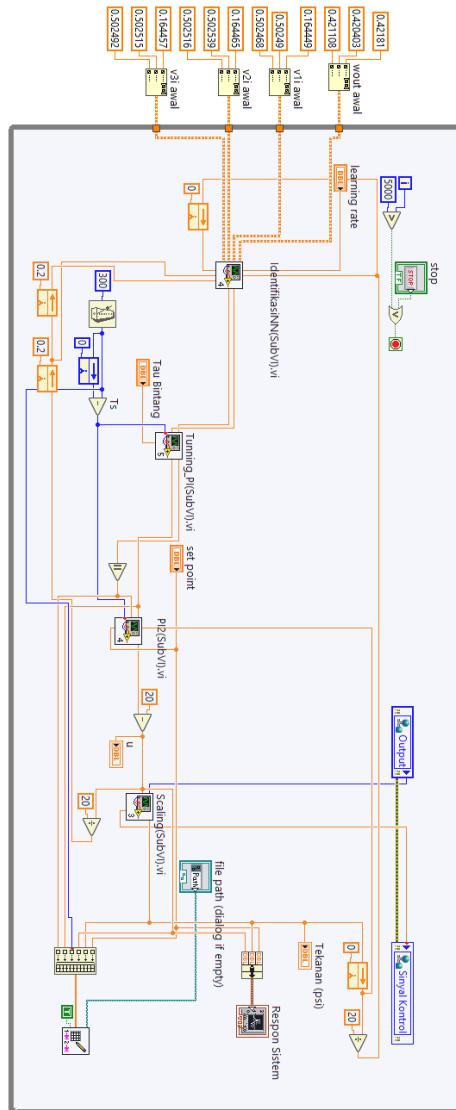
Lampiran 10. Program LabVIEW neuron *output-layer*



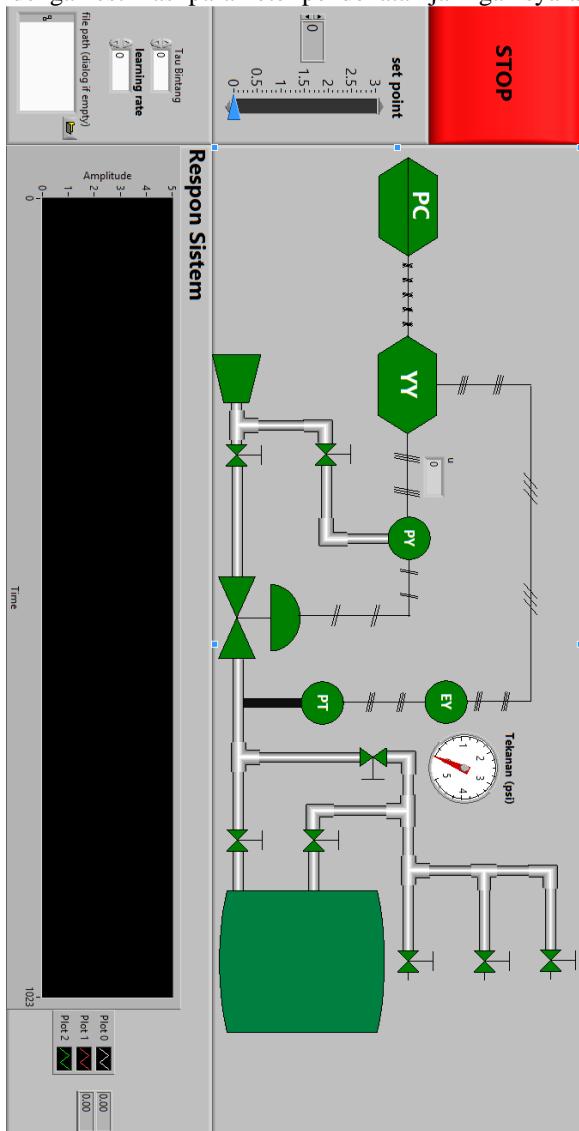
Lampiran 11. Program LabVIEW estimasi parameter pendekatan jaringan syaraf tiruan



Lampiran 12. Program LabVIEW *self-tunning regulator* dengan estimasi parameter pendekatan jaringan syaraf tiruan



Lampiran 13. Human machine interface (HMI) program self-tunning regulator dengan estimasi parameter pendekatan jaringan syaraf tiruan



Lampiran 14. Program MATLAB pengolah nilai RMSE, ITSE, ITAE, dan ISTSE serta perhitungan overshoot.

```
%% penyekalaan waktu
%%gama04 = data STR RLS dengan forgetting factor
=0.4
%%gama06 = data STR RLS dengan forgeetting
factor =0.6
%%gama08 = data STR RLS dengan forgeetting
factor =0.8
%%gama1 = data STR RLS dengan forgetting factor
=1
%%gama12 = data STR RLS dengan forgetting factor
=1.2
%%l5= data STR NN dengan learning rate=5
%%l10= data STR NN dengan learning rate=10
%%l15= data STR NN dengan learning rate=15
%%20= data STR NN dengan learning rate=20
%%l25= data STR NN dengan learning rate=25
%%gamaxx(1,4) dan lxx(1,4) = waktu waktu awal
pengambilan data
gama04(:,4)=gama04(:,4)-gama04(1,4);
%%gamaxx(1,4)= waktu waktu awal pengambilan data
gama04(:,4)=gama04(:,4)/1000;; %% milisekon ke
sekon
gama06(:,4)=gama06(:,4)-gama06(1,4);
gama06(:,4)=gama06(:,4)/1000; %% milisekon ke
sekon
gama08(:,4)=gama08(:,4)-gama08(1,4);
gama08(:,4)=gama08(:,4)/1000; %% milisekon ke
sekon
gama1(:,4)=gama1(:,4)-gama1(1,4);
gama1(:,4)=gama1(:,4)/1000; %% milisekon ke
sekon
gama12(:,4)=gama12(:,4)-gama12(1,4);
gama12(:,4)=gama12(:,4)/1000; %% milisekon ke
sekon
l5(:,4)=l5(:,4)-l5(1,4);
l5(:,4)=l5(:,4)/1000; %% milisekon ke sekon
l10(:,4)=l10(:,4)-l10(1,4);
```

```

110(:,4)=110(:,4)/1000;%% milisekon ke sekon
115(:,4)=115(:,4)-115(1,4);
115(:,4)=115(:,4)/1000;%% milisekon ke sekon
120(:,4)=120(:,4)-120(1,4);
120(:,4)=120(:,4)/1000;%% milisekon ke sekon
125(:,4)=125(:,4)-125(1,4);
125(:,4)=125(:,4)/1000;%% milisekon ke sekon

%gama04(:,1)=set point
%sampling time = 300 ms

%% RMSE
rmse(1,1)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
gama04(:,2)).^2)).^0.5))%gama=0.4
rmse(1,2)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
gama06(:,2)).^2)).^0.5))%gama=0.6
rmse(1,3)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
gama08(:,2)).^2)).^0.5))%gama=0.8
rmse(1,4)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
gama1(:,2)).^2)).^0.5))%gama=1
rmse(1,5)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
gama12(:,2)).^2)).^0.5))%gama=1.2
rmse(1,6)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
15(:,2)).^2)).^0.5))%learning rate=5
rmse(1,7)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
110(:,2)).^2)).^0.5))%learning rate=10
rmse(1,8)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
115(:,2)).^2)).^0.5))%learning rate=15
rmse(1,9)=
(1/length(gama04(:,1)) * (sum(((gama04(:,1)-
120(:,2)).^2)).^0.5))%learning rate=20

```

```

rmse(1,10)=
(1/length(gama04(:,1)))*(sum(((gama04(:,1)-
125(:,2)).^2)).^0.5));%learnig rate=25
(i,2))*0.3);%learning rate=25
End

%%ITSE
ITSE=zeros(1,10);
for i=1:length(gama04(:,1))
    ITSE(1,1)=ITSE(1,1)+%
(gama04(i,4)*((gama04(i,1)-
gama04(i,2))^2)*0.3);%gama=0.4
    ITSE(1,2)=ITSE(1,2)+%
(gama06(i,4)*((gama04(i,1)-
gama06(i,2))^2)*0.3);%gama=0.4
    ITSE(1,3)=ITSE(1,3)+%
(gama08(i,4)*((gama04(i,1)-
gama08(i,2))^2)*0.3);%gama=0.4
    ITSE(1,4)=ITSE(1,4)+%
(gama1(i,4)*((gama04(i,1)-
gama1(i,2))^2)*0.3);%gama=0.4
    ITSE(1,5)=ITSE(1,5)+%
(gama12(i,4)*((gama04(i,1)-
gama12(i,2))^2)*0.3);%gama=0.4
    ITSE(1,6)=ITSE(1,6)+ (15(i,4)*((gama04(i,1)-
15(i,2))^2)*0.3);%learning rate=5
    ITSE(1,7)=ITSE(1,7)+%
(l10(i,4)*((gama04(i,1)-
l10(i,2))^2)*0.3);%learning rate=10
    ITSE(1,8)=ITSE(1,8)+%
(l15(i,4)*((gama04(i,1)-
l15(i,2))^2)*0.3);%learning rate=15
    ITSE(1,9)=ITSE(1,9)+%
(l20(i,4)*((gama04(i,1)-
l20(i,2))^2)*0.3);%learning rate=20
    ITSE(1,10)=ITSE(1,10)+%
(l25(i,4)*((gama04(i,1)-
l25(i,2))^2)*0.3);%learning rate=25
End

```

```

%%ITAE
ITAE=zeros(1,10);
for i=1:length(gama04(:,1))
    ITAE(1,1)=ITAE(1,1) +
        (gama04(i,4)*abs(gama04(i,1)-
        gama04(i,2)) *0.3);%gama=0.4
    ITAE(1,2)=ITAE(1,2) +
        (gama06(i,4)*abs(gama04(i,1)-
        gama06(i,2)) *0.3);%gama=0.6
    ITAE(1,3)=ITAE(1,3) +
        (gama08(i,4)*abs(gama04(i,1)-
        gama08(i,2)) *0.3);%gama=0.8
    ITAE(1,4)=ITAE(1,4) +
        (gama1(i,4)*abs(gama04(i,1)-
        gama1(i,2)) *0.3);%gama=1
    ITAE(1,5)=ITAE(1,5) +
        (gama12(i,4)*abs(gama04(i,1)-
        gama12(i,2)) *0.3);%gama=1.2
    ITAE(1,6)=ITAE(1,6) +
        (15(i,4)*abs(gama04(i,1)-
        15(i,2)) *0.3);%learning rate=5
    ITAE(1,7)=ITAE(1,7) +
        (110(i,4)*abs(gama04(i,1)-
        110(i,2)) *0.3);%learning rate=10
    ITAE(1,8)=ITAE(1,8) +
        (115(i,4)*abs(gama04(i,1)-
        115(i,2)) *0.3);%learning rate=15
    ITAE(1,9)=ITAE(1,9) +
        (120(i,4)*abs(gama04(i,1)-
        120(i,2)) *0.3);%learning rate=20
    ITAE(1,10)=ITAE(1,10) +
        (125(i,4)*abs(gama04(i,1)-
        125(i,2)) *0.3);%learning rate=25
End

%%ISTSE
ISTSE=zeros(1,10);
for i=1:length(gama04(:,1))

```

```

ISTSE(1,1)=ISTSE(1,1) +
((gama04(i,4)^2)*((gama04(i,1)-
gama04(i,2))^2)*0.3);%gama=0.4
ISTSE(1,2)=ISTSE(1,2) +
((gama06(i,4)^2)*((gama04(i,1)-
gama06(i,2))^2)*0.3);%gama=0.4
ISTSE(1,3)=ISTSE(1,3) +
((gama08(i,4)^2)*((gama04(i,1)-
gama08(i,2))^2)*0.3);%gama=0.4
ISTSE(1,4)=ISTSE(1,4) +
((gama1(i,4)^2)*((gama04(i,1)-
gama1(i,2))^2)*0.3);%gama=0.4
ISTSE(1,5)=ISTSE(1,5) +
((gama12(i,4)^2)*((gama04(i,1)-
gama12(i,2))^2)*0.3);%gama=0.4
ISTSE(1,6)=ISTSE(1,6) +
((15(i,4)^2)*((gama04(i,1)-
15(i,2))^2)*0.3);%learning rate=5
ISTSE(1,7)=ISTSE(1,7) +
((110(i,4)^2)*((gama04(i,1)-
110(i,2))^2)*0.3);%learning rate=10
ISTSE(1,8)=ISTSE(1,8) +
((115(i,4)^2)*((gama04(i,1)-
115(i,2))^2)*0.3);%learning rate=15
ISTSE(1,9)=ISTSE(1,9) +
((120(i,4)^2)*((gama04(i,1)-
120(i,2))^2)*0.3);%learning rate=20
ISTSE(1,10)=ISTSE(1,10) +
((125(i,4)^2)*((gama04(i,1)-
125(i,2))^2)*0.3);%learning rate=25
End

%% %Overshoot
ov(1,1)=(max(gama04(:,2))-gama04(1,1))/gama04(1,1)*100;%gama=0.4
ov(1,2)=(max(gama06(:,2))-gama04(1,1))/gama04(1,1)*100;%gama=0.6
ov(1,3)=(max(gama08(:,2))-gama04(1,1))/gama04(1,1)*100;%gama=0.8

```

```
ov(1,4)=(max(gama1(:,2))-  
gama04(1,1))/gama04(1,1)*100;%gama=1  
ov(1,5)=(max(gama12(:,2))-  
gama04(1,1))/gama04(1,1)*100;%gama=1.2  
ov(1,6)=(max(l15(:,2))-  
gama04(1,1))/gama04(1,1)*100;%learning rate=5  
ov(1,7)=(max(l10(:,2))-  
gama04(1,1))/gama04(1,1)*100;%learning rate=10  
ov(1,8)=(max(l15(:,2))-  
gama04(1,1))/gama04(1,1)*100;%learning rate=15  
ov(1,9)=(max(l120(:,2))-  
gama04(1,1))/gama04(1,1)*100;%learning rate=20  
ov(1,10)=(max(l125(:,2))-  
gama04(1,1))/gama04(1,1)*100;%learning rate=25
```

RIWAYAT HIDUP



Penulis bernama Febry Angga Gunawan, lahir di Magetan, 13 Juli 1994. Merupakan anak kedua dari tiga bersaudara. Berasal dari Kota Magetan dan sekarang tinggal di Jl. KH. Ahmad Dahlan 50A Surabaya. Penulis menempuh pendidikan di SD Negeri 3 Magetan pada tahun 2001 - 2007, kemudian penulis melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 1 Magetan tahun 2007 – 2010, setelah itu penulis melanjutkan pendidikan mengengah atas di SMA Negeri 3 Madiun tahun 2010 – 2013, penulis menempuh pendidikan tinggi di Institut

[Halaman ini sengaja dikosongkan]