



TUGAS AKHIR -EE 184801

**PENGATURAN TEKANAN PADA *RIG* 38-714
MENGUNAKAN KONTROLER PI BERBASIS *NEURAL*
NETWORK DENGAN SKEMA MRAC**

Elka Gumilar
NRP 0711174500047

Dosen Pembimbing
Ir. Ali Fatoni MT.
Eka Iskandar, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



FINAL PROJECT -EE 184801

***PRESSURE CONTROL FOR RIG 38-714 USING PI
CONTROLLER BASED NEURAL NETWORK WITH
MRAC SCHEME***

Elka Gumilar
NRP 07111745000047

Supervisor
Ir. Ali Fatoni MT.
Eka Iskandar, ST., MT.

***ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019***

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pengaturan Tekanan pada Rig 38-714 Menggunakan Kontroler PI Berbasis *Neural Network* dengan Skema MRAC**" merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan – bahan yang tidak diijinkan 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 2019



Elka Gumilar
NRP 07111745000047

[Halaman ini sengaja dikosongkan]

**PENGATURAN TEKANAN PADA RIG 38-714 MENGGUNAKAN
KONTROLER PI BERBASIS NEURAL NETWORK DENGAN
SKEMA MRAC**

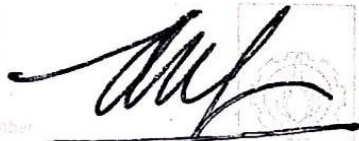
TUGAS AKHIR

**Diajukan Guna 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



Ir. Ali Fatoni MT.
NIP.196206031989031002



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



[Halaman ini sengaja dikosongkan]

PENGATURAN TEKANAN PADA *RIG 38-714* MENGUNAKAN KONTROLER PI BERBASIS *NEURAL NETWORK* DENGAN SKEMA MRAC

Nama : Elka Gumilar
Pembimbing 1 : Ir. Ali Fatoni MT.
Pembimbing 2 : Eka Iskandar, ST., MT.

ABSTRAK

Dalam dunia industri saat ini, sistem pengaturan proses mempunyai peran penting yaitu untuk mencapai kondisi proses seperti yang diinginkan agar diperoleh hasil akhir yang sesuai. Perubahan pada kondisi proses bisa berakibat fatal. Maka dari itu dibutuhkan sebuah aksi pengaturan. Salah satu aksi pengaturan pada sistem pengaturan proses adalah pengaturan *pressure*. Pengaturan yang dilakukan dengan menambahkan kontroler yang memiliki kemampuan untuk beradaptasi dengan kondisi beban maupun masukan yang berubah. Kontroler yang dapat digunakan pada kondisi tersebut adalah kontroler PI berbasis *Neural Network* dengan skema MRAC dimana *Neural Network* digunakan untuk *update* nilai parameter K_p dan K_i . Kontroler tersebut diaplikasikan pada *Pressure Process Rig 38-714* dimana *plant* tersebut merupakan representasi dari sistem pengaturan *pressure* yang ada di industri. Hasil pengujian kontroler adaptif pada *plant* tersebut didapatkan bahwa pada pengujian masukan yang berubah, kontroler mampu beradaptasi sesuai dengan model referensi dengan nilai RMSE yang kecil yaitu 0.0343 dan mampu mencapai nilai *settling time* mendekati waktu yang diinginkan yaitu 50.14 detik. Sedangkan pada pengujian terhadap perubahan beban, kontroler mampu mengembalikan nilai sesuai dengan nilai masukan dalam waktu yang cepat yaitu 9.3 detik ketika katup V5 dibuka dan 9 detik ketika katup V5 ditutup.

Kata kunci : Kontrol PI, *Neural Network*, *Pressure Process Rig 38-714*.

[Halaman ini sengaja dikosongkan]

PRESSURE CONTROL FOR RIG 38-714 USING PI CONTROLLER BASED NEURAL NETWORK WITH MRAC SCHEME

Name : Elka Gumilar
Supervisor 1 : Ir. Ali Fatoni MT.
Supervisor 2 : Eka Iskandar, ST., MT

ABSTRACT

In industry, control process systems have an important role, namely to achieve the desired process conditions so that the appropriate final results are obtained. Changes to process conditions can be dangerous. Therefore it requires a control action. One of the control action in the process control system is the control pressure. Control action made by adding controllers that have the ability to adapt to changing load conditions and inputs. The controller that can be used in these conditions is a Neural Network-based PI controller with an MRAC scheme where the Neural Network is used to update the Kp and Ki parameter values. The controller is applied to the Pressure Process Rig 38-714 where in the plant is a representation of a pressure control system in the industry. The results of the adaptive controller test at the plant found that in testing the changing inputs, the controller was able to adapt according to the reference model with a small RMSE value of 0.0343 and was able to reach the settling time value close to the desired time of 50.14 seconds. While in testing the load changes, the controller is able to return the value according to the input value in a fast time of 9 seconds.

Keywords : *Controller PI, Neural Network, Pressure Process Rig 38-714*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga karya tulis tugas akhir ini dengan judul **“Pengaturan Tekanan pada Rig 38-714 Menggunakan Kontroler PI Berbasis Neural Network dengan Skema MRAC”** dapat selesai tepat pada waktunya. Penulis mengucapkan banyak terima kasih kepada seluruh pihak yang terlibat dan membantu menyelesaikan karya tulis ini. Oleh karena itu, ucapan terima kasih penulis sampaikan secara khusus kepada:

1. Kedua orang tua dan keluarga
2. Bapak Ir. Ali Fatoni MT. dan Bapak Eka Iskandar, S.T, MT. sebagai pembimbing
3. Rekan – rekan Lintas Jalur angkatan 2017, terutama mahasiswa Teknik Sistem Pengaturan dan Lab B105.
4. Hasbi, Jabber, Aan, Fauzi, rekan – rekan Tugas Akhir bimbingan Bapak Ir. Ali Fatoni MT. dan Bapak Eka Iskandar, S.T, MT. yang selalu menyemangati dan menemani
5. Agung, Farish, Khaerul, Enggal, sahabat penghuni VenusP16 dan Q41 yang selalu mendukung
6. Ferly Asri yang selalu memberikan saran dan semangat
7. Pihak lain yang ikut membantu penulis tidak dapat disebutkan namanya satu-persatu.

Semoga karya tulis ini dapat bermanfaat bagi penulis sendiri maupun bagi penelitian selanjutnya.

Surabaya, Juni 2019

Elka Gumilar
Nrp 07111745000047

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN	v
TUGAS AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
BAB 1	1
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Permasalahan.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Metodologi	3
1.6. Sistematika	4
1.7. Relevansi atau Manfaat	4
BAB 2	5
TINJAUAN PUSTAKA.....	5
1.1. <i>Pressure Process Rig</i> 38-714	5
1.2. Sistem Pengaturan Proses [4]	7
1.3. Karakteristik Sistem Orde Pertama [5].....	7
1.4. Kontroler Proporsional Integral [6] [4].....	8

1.5.	Sistem Pengaturan Adaptif	9
1.6.	MRAC [7]	11
1.7.	<i>Neural Network</i> (Jaringan Saraf Tiruan) [8]	11
BAB 3	17
PERANCANGAN SISTEM	17
3.1.	Konfigurasi <i>Plant</i>	17
3.1.1.	Kalibrasi Sensor	18
3.1.2.	Metode Pembebanan	19
3.2.	Identifikasi <i>Plant</i>	19
3.2.1.	Instalasi Perangkat	20
3.2.2.	Konfigurasi Perangkat	23
3.2.3.	Program Identifikasi Pada NI LabVIEW 2018	24
3.2.4.	Identifikasi Sistem	25
3.3.	Validasi Model	28
3.4.	Perancangan Kontroler PI	30
3.5.	Perancangan Kontroler PI dengan <i>Model Reference</i>	31
3.6.	Perancangan Struktur <i>Neural Network</i>	32
BAB 4	39
PENGUJIAN DAN ANALISA	39
4.1.	Simulasi Proses Pembelajaran <i>Neural Network</i>	39
4.2.	Pengujian Kontroler PI berbasis <i>Neural Network</i> dengan skema MRAC tanpa perubahan beban	41
4.3.	Pengujian Kontroler PI Berbasis <i>Neural Network</i> dengan skema MRAC Terhadap Perubahan Beban.....	43
4.4.	Pengujian Kontroler PI Berbasis <i>Neural Network</i> dengan skema MRAC Terhadap Perubahan Masukan	45
BAB 5	49
PENUTUP	49

4.5.	Kesimpulan.....	49
4.6.	Saran.....	49
DAFTAR PUSTAKA		51
LAMPIRAN.....		53
RIWAYAT HIDUP		55

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1	<i>Pressure Process Rig</i> 38-714.....	5
Gambar 2.2	Blok Diagram Umum Orde Pertama.....	8
Gambar 2.3	Diagram Blok Kontroler Proporsional Integral	9
Gambar 2.4	Diagram Masukan Tangga Satuan	9
Gambar 2.5	Diagram Keluaran Kontroler	9
Gambar 2.6	Blok Diagram dari Sistem Pengaturan Adaptif.....	10
Gambar 2.7	Blok Diagram dari MRAC.....	11
Gambar 2.8	Hubungan Antar <i>Neuron</i>	12
Gambar 2.9	Grafik Fungsi Aktivasi Sigmoid	14
Gambar 2.10	Arsitektur Backpropagation	13
Gambar 3.1	<i>Pressure Process Rig</i> 38-714	17
Gambar 3.2	<i>Wiring</i> Kalibrasi Sensor <i>Pressure</i>	18
Gambar 3.3	<i>Valve</i> Manual untuk Pembebanan.....	19
Gambar 3.4	Diagram Instalasi Perangkat pada Sistem Pengaturan <i>Pressure Process Rig</i> 38-714.....	20
Gambar 3.5	Diagram Pengambilan Data <i>Pressure Process</i> <i>Rig</i> 38-714	21
Gambar 3.6	Software ADAM-5000TCP-6000 Utility.....	23
Gambar 3.7	Tampilan <i>Front Panel</i> Identifikasi <i>Plant</i> NI LabVIEW 2018.....	24
Gambar 3.8	Tampilan Program Identifikasi <i>Plant</i> NI LabVIEW 2018.....	25
Gambar 3.9	Skema Identifikasi <i>Plant</i>	25
Gambar 3.10	Respon <i>Plant Open Loop</i> Terhadap Sinyal Uji <i>Step</i>	26
Gambar 3.11	Eror keluaran <i>plant</i> terhadap Masukan	26
Gambar 3.12	Perbandingan Respon <i>Plant</i> dengan Persamaan Model <i>Plant</i>	29
Gambar 3.13	Perbandingan Eror Keluaran <i>Plant</i> terhadap Masukan dan Eror Keluaran Model terhadap Masukan	29
Gambar 3.14	Diagram Blok Sistem Pengaturan <i>Pressure</i> dengan Kontroler PI	30
Gambar 3.15	Diagram Blok Kontroler PI berbasis <i>Neural Network</i> dengan Skema MRAC	32
Gambar 3.16	Struktur <i>Neural Network</i>	33
Gambar 3.17	<i>Neural Network</i> dengan <i>Toolbox</i> Matlab Simulink	34
Gambar 3.18	Perbandingan hasil simulasi dengan variasi jumlah <i>neuron</i> pada <i>hidden layer</i>	34

Gambar 3.19	Respon perbandingan eror hasil simulasi dengan perubahan jumlah <i>neuron</i> pada <i>hidden layer</i>	34
Gambar 4.1	Diagram Blok Proses Pembelajaran <i>Neural Network</i>	39
Gambar 4.2	Simulasi Hasil Pembelajaran <i>Neural Network</i> dengan variasi <i>learning rate</i>	40
Gambar 4.3	Grafik eror Hasil Pembelajaran <i>Neural Network</i> dengan variasi <i>learning rate</i>	40
Gambar 4.4	Respon sistem Kontroler PI berbasis <i>Neural Network</i> dengan skema MRAC tanpa perubahan beban	42
Gambar 4.5	Eror keluaran <i>plant</i> terhadap masukan ketika pengujian tanpa perubahan beban.....	42
Gambar 4.6	Respon Keluaran <i>Plant</i> terhadap perubahan beban ketika katup V5 ditutup	43
Gambar 4.7	Eror keluaran <i>plant</i> terhadap masukan ketika perubahan beban katup V5 ditutup	44
Gambar 4.8	Respon Keluaran <i>Plant</i> terhadap perubahan beban ketika katup V5 dibuka.....	44
Gambar 4.9	Eror keluaran <i>plant</i> terhadap masukan ketika perubahan beban katup V5 ditutup	45
Gambar 4.10	Respon Kontroler PI berbasis <i>Neural Network</i> dengan skema MRAC terhadap Perubahan Masukan	46
Gambar 4.11	Eror keluaran terhadap masukan ketika perubahan masukan.....	46

DAFTAR TABEL

Tabel 3.1 Persamaan Model <i>Plant</i>	28
Tabel 3.2 Nilai RMSE variasi jumlah <i>neuron</i> pada <i>hidden layer</i>	34
Tabel 4.1 Nilai RMSE <i>Learning Rate</i>	41
Tabel 4.2 Spesifikasi Respon Kontroler PI berbasis <i>Neural Network</i> dengan skema MRAC	42
Tabel 4.3 Respon <i>Plant</i> Terhadap Perubahan Beban	45
Tabel 4.4 Nilai RMSE pada Pengujian Perubahan Masukan	47
Tabel 4.5 Karakteristik Respon Transien Keluaran <i>Plant</i> Terhadap Perubahan Masukan Dengan Nilai <i>Learning Rate</i> 0.000001	47
Tabel 4.6 Karakteristik Respon Transien Keluaran <i>Plant</i> Terhadap Perubahan Masukan Dengan Nilai <i>Learning Rate</i> 0.000005	47
Tabel 4.7 Karakteristik Respon Transien Keluaran <i>Plant</i> Terhadap Perubahan Masukan Dengan Nilai <i>Learning Rate</i> 0.00001	47

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Dalam dunia industri saat ini, sistem pengaturan proses mempunyai tujuan yaitu untuk mencapai kondisi proses seperti yang diinginkan agar diperoleh produk akhir yang sesuai. Proses tidak perlu dikendalikan jika tujuan proses tercapai tanpa menggunakan sistem pengaturan. Namun akan sebaliknya jika mencapai tujuan proses memerlukan *monitoring* secara berkala maupun terus menerus. Dalam industri proses perubahan sedikit saja pada kondisi proses bisa berakibat fatal. Maka dari itu dibutuhkan sebuah aksi pengaturan. [1]

Aksi pengaturan pada sistem pengaturan proses meliputi pengaturan *flow*, pengaturan *temperature*, pengaturan *level* dan pengaturan *pressure*. Salah satu aksi pengaturan pada sistem pengaturan proses adalah pengaturan *pressure* (udara bertekanan). Dalam dunia industri saat ini, penggunaan udara bertekanan memiliki fungsi penting. Berbagai proses industri membutuhkan suatu hasil pengukuran udara bertekanan yang akurat, dan hasil pengukuran udara bertekanan tersebut dapat digunakan menjadi suatu sistem otomatis untuk pengaturan pada proses selanjutnya.

Namun dalam pengaplikasiannya, pengaturan udara bertekanan pada sistem mempunyai peranan penting agar sistem dapat berjalan dengan baik dan stabil. Contohnya pada perusahaan minuman, udara bertekanan digunakan untuk proses pencucian botol. Pencuci botol ini menggunakan katup pneumatik untuk mengatur masuk dan keluarnya botol yang dibersihkan. Jika hasil pengukuran tekanan tidak akurat memungkinkan botol yang belum bersih masuk ke proses selanjutnya. Hal ini dapat menyebabkan peningkatan biaya dan masalah *quality control*. [2]

Pengaturan yang dilakukan agar keakuratan terjaga dengan menambahkan kontroler yang memiliki kemampuan untuk beradaptasi dengan kondisi beban maupun masukan yang berubah. Banyak penelitian mengenai pengembangan sistem pengaturan adaptif seperti pada penelitian yang dilakukan oleh Ahmad Faizal. Dalam penelitian [3] membahas tentang kontroler MRAC dengan kombinasi PID yang di implementasikan pada *Pressure Process Rig 38-714*. Respon yang dihasilkan masih menghasilkan osilasi jika hanya menggunakan pengendali MRAC saja. Ketika dikombinasikan dengan PID, respon yang

dihasilkan mampu mengikuti model referensi tanpa osilasi. Untuk mendapatkan nilai gain adaptasi masih menggunakan *trial and error* sehingga hasil yang didapat belum tentu hasil terbaik. Pengujian metode kontrol yang sudah dilakukan belum mencakup pengujian untuk masukan yang berubah-ubah. Pengujian dengan masukan yang berubah perlu dilakukan karena kontroler yang baik harus mampu mengikuti perubahan nilai *setpoint* sesuai dengan spesifikasi yang diinginkan disetiap waktu dalam keadaan perubahan beban disetiap waktunya. Pada tugas akhir yang dikerjakan dengan ditambahkannya sistem pengaturan cerdas *Neural Network* dalam mekanisme adaptasi diharapkan dapat mempermudah perancangan kontroler dengan kondisi variasi beban dan variasi nilai *setpoint* yang akan di implementasikan pada *plant Pressure Process Rig 38-714*.

1.2. Permasalahan

Pada tugas akhir ini, hal yang menjadi fokus permasalahan adalah variasi beban dan perubahan masukan. Kontroler konvensional tidak mampu untuk menjaga spesifikasi respon yang diinginkan ketika terjadi perubahan beban maupun perubahan masukan yang sering terjadi. Perlu kontroler yang dapat meminimalisir gangguan pada saat perubahan beban, mempercepat respon menuju nilai *setpoint* kembali ketika terjadi perubahan beban dan dapat mempertahankan spesifikasi respon yang diinginkan ketika perubahan masukan.

1.3. Batasan Masalah

Fokus permasalahan yang dibahas pada penelitian kali ini memiliki beberapa batasan antara lain:

- a. *Plant* yang digunakan adalah *Pressure Process Rig 38-714* yang berada di Lab B105.
- b. Kontroler yang digunakan adalah kontroler PI berbasis *Neural Network* dengan skema MRAC.

1.4. Tujuan

Tujuan dari tugas akhir ini adalah analisa bagaimana sistem pengaturan adaptif menggunakan skema MRAC dengan kontroler PI berbasis *Neural Network* dapat menyelesaikan permasalahan variasi beban dan perubahan masukan yang akan di implementasikan pada *plant Pressure Process Rig 38-714*.

1.5. Metodologi

Pada penelitian ini terdapat prosedur ilmiah yang dilakukan untuk mendapatkan hasil penelitian dengan beberapa tahapan sebagai berikut:

1. Studi Literatur

Studi literatur dilakukan untuk menunjang penguasaan materi mengenai bahasan yang dikerjakan melalui media cetak berupa buku sumber, paper ataupun jurnal. Hal yang dipelajari meliputi penggunaan *plant Pressure Process Rig 38-714* dan konsep perancangan kontroler menggunakan kontroler PI berbasis *Neural Network* dengan skema MRAC.

2. Identifikasi Sistem

Pada proses identifikasi sistem, *plant* diberikan sinyal uji berupa sinyal *step* guna mendapatkan respon *plant* yang akan diuji dan dianalisa. Dari keluaran tersebut didapatkan fungsi alih sistem *Pressure Process Rig 38-714*.

3. Perancangan Kontroler

Setelah dilakukan identifikasi pada *plant*, dan fungsi alih dari sistem, maka dilakukan perancangan kontroler. Kontroler yang akan digunakan adalah Kontroler PI berbasis *Neural Network* dengan skema MRAC. Diharapkan dengan perancangan kontroler yang sesuai dengan fungsi alih sistem, kontroler dapat berjalan sesuai dengan karakteristik model referensi yang diinginkan.

4. Simulasi & Analisis

Dari hasil pemodelan dan perancangan kontroler dilakukan simulasi yang disertai dengan analisis untuk mengoreksi kesesuaian dengan hasil yang diinginkan. Diperlukan simulasi terlebih dahulu pada struktur *Neural Network* untuk mendapatkan nilai bobot yang sesuai sebelum mengimplementasikannya langsung pada *plant* guna memastikan kontroler yang dirancang aman digunakan pada *plant*.

5. Implementasi

Hasil dari simulasi jika sudah terjadi kesesuaian dan aman untuk diterapkan, maka hasil perancangan diterapkan sebagai kontrol pada sistem *Pressure Process Rig 38-714*.

6. Validasi

Validasi dilakukan untuk mengetahui hasil yang didapat setelah penerapan kontroler pada *plant Pressure Process Rig 38-714* sudah sesuai dengan pemodelan sistem yang sudah dilakukan dengan menggunakan metode validasi RMSE.

7. Penyusunan Laporan TA

Penulisan dan penyusunan laporan tugas akhir, yang terdiri dari bab pendahuluan, dasar teori, perancangan sistem, implementasi dan penutup.

1.6. Sistematika

Dalam penulisan karya tulis tugas akhir ini, sistematika penulisan dibagi menjadi 5 bab dengan uraian pada setiap bab sebagai berikut:

Bab I : Pendahuluan

Pada bab ini akan diberikan penjelasan mengenai latar belakang, permasalahan, tujuan, metodologi, sistematika dan relevansi atau manfaat.

Bab II : Tinjauan Pustaka

Pada bab ini akan diberikan penjelasan mengenai tinjauan pustaka yang dapat membantu menyelesaikan penelitian diantaranya adalah teori sistem pengaturan cerdas, teori sistem pengaturan adaptif, teori penerapan kontroler PID, teori *Neural Network* atau Jaringan Saraf Tiruan dan penjelasan *plant Pressure Process Rig* 38-714.

Bab III : Perancangan Sistem

Pada bab ini akan diberikan penjelasan mengenai langkah perancangan kontroler PI dengan skema MRAC dengan mekanisme adaptasi menggunakan algoritma *Neural Network* yang akan diimplementasikan pada *plant Pressure Process Rig* 38-714.

Bab IV : Pengujian dan Analisa

Pada bab ini akan diberikan penjelasan mengenai hasil yang didapat dari perancangan kontroler yang sudah diimplementasikan pada *plant Pressure Process Rig* 38-714.

Bab V : Penutup

Pada bab ini diberikan penjelasan tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

1.7. Relevansi atau Manfaat

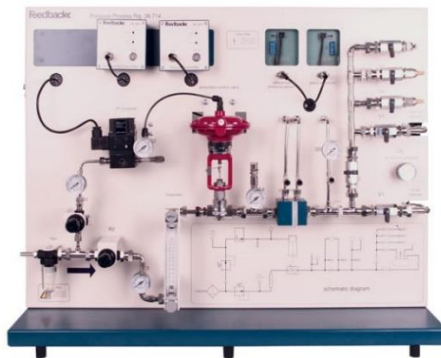
Diharapkan hasil penelitian ini dapat digunakan sebagai bahan pengembangan penelitian selanjutnya pada *Pressure Process Rig* 38-714 khususnya pengembangan dengan sistem pengaturan adaptif skema MRAC dengan kontroler PI berbasis *Neural Network*.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai dasar teori apa saja yang dapat menunjang penelitian yang dilakukan pada tugas akhir ini meliputi definisi *plant Pressure Process Rig 38-714*, sistem pengaturan proses, karakteristik sistem orde pertama, kontrol proposional integral, definisi sistem pengaturan adaptif dan *Neural Network*.

1.1. *Pressure Process Rig 38-714*

Desain dan operasi industri yang sukses membutuhkan pemilihan peralatan instrumentasi dan kontrol yang baik sesuai dengan kemampuannya untuk memerintahkan, memantau hingga melakukan pemeliharaan pada sistem di industri. Teknisi dan para *engineer* harus melatih kemampuannya terutama pada skill penggunaan komputer dan mikrokontroler mengingat kemajuan pesat dalam penggunaannya pada bidang pengendalian proses. Rangkaian yang dirancang ini memungkinkan pengguna dapat memahami secara praktis dari teknologi kontrol proses mulai dari setiap komponen yang digunakan hingga kontrol kompleks yang digunakan pada operasi pengendalian proses nyata. Dalam hal ini, yang digunakan adalah *Pressure Process Rig 38-714* seperti yang terlihat pada Gambar 2.1 yang merupakan bagian dari peralatan Procon yang dirancang untuk memperkenalkan dan menunjukkan prinsip-prinsip pengukuran dan kontrol proses.



Gambar 2.1 *Pressure Process Rig 38-714*

Sistem ini mencakup komponen kontrol pneumatik yang menarik bagi industri proses. Desain ini memungkinkan studi operasi komponen dan koneksi ke perangkat kontrol listrik melalui penggunaan tekanan/ arus transduser. Pada panel depan, *Pressure Process Rig 38-714* menggabungkan diagram skematik dari peralatan yang mengidentifikasi komponen utama dan koneksi yang relevan dalam notasi pneumatik standar. Unit ini terdiri dari *Pneumatic Control Valve*, *Orifice Block* dan *Pressure Tapping*. Katup dioperasikan dengan mengatur arus masukan pada I/P konverter. Untuk pengukuran tekanan langsung menggunakan sensor *pressure* dan sensor *differential pressure* untuk pengukuran tekanan dan aliran. Agar proses pengendalian *pressure* pada *Pressure Process Rig 38-714* ini berjalan dengan baik, perlu diketahui juga informasi dari peralatan pendukung yang menunjang pada proses pengendalian *pressure* diantaranya:

a. *Control Valve*

Control valve terdiri dari aktuator diafragma yang mempunyai posisi batang katup jenis *plug*. Indikator pada batang katup menunjukkan posisi sebenarnya dari katup. Terdapat Indikator untuk menunjukkan sinyal pneumatik yang diterapkan pada *Control valve* oleh I/P konverter. Masukan 4-20 mA ke I/P konverter dapat dihubungkan ke *Interface Process 38-200* atau koneksi *servo valve*. Udara yang mengalir melalui pipa proses melewati *Pneumatic Control Valve* dan *Orifice Block* sebelum dibuang ke atmosfer melalui *Diffuser*. Seperangkat manual valve V1, V2, dan V3 memungkinkan penerima udara yang dipasang di belakang dihubungkan secara seri atau paralel dengan pipa proses untuk mengubah respon sistem. *Air Receiver* menggabungkan *pressure relief valve*. Perubahan langkah dapat diterapkan pada proses dengan pembuangan udara melalui *Diffuser* tambahan dengan membuka dan menutup *valve* V4, V5 dan V7. Rig ini menggabungkan komponen kontrol pneumatik yang sangat menarik bagi industri proses. Desain ini memungkinkan studi operasi komponen dan koneksi perangkat kontrol melalui penggunaan konverter tekanan saat ini.

b. *Process Controller*

Process Controller 38-300 digunakan dengan I/P konverter untuk memberikan sinyal linier. Kedua sensor dilengkapi dengan kondisioner. Unit ini dirancang untuk beroperasi dengan *Interface Process* dan *Process*

Controller untuk mengkonfigurasi rangkaian kontrol *loop* terbuka atau tertutup. Udara terkompresi untuk instrumentasi pneumatik dan proses terhubung ke saluran masuk umum. Udara untuk instrumentasi diatur oleh *Regulator* R1 dan tekanan operasi ditunjukkan pada *Gauge* G1. Udara untuk proses diatur oleh *Regulator* R2 dan tekanan operasi ditunjukkan pada *Gauge* G3. Instrumentasi pneumatik terdiri dari I/P Konverter dan *Pneumatic Control Valve*. I/P Konverter menerima sinyal kontrol 4-20 mA dari *Process Interface* dan mengubahnya menjadi sinyal pneumatik 3-15 psi yang mengoperasikan *control valve*.

1.2. Sistem Pengaturan Proses [4]

Hampir semua proses dalam dunia industri membutuhkan peralatan-peralatan otomatis untuk mengendalikan parameter-parameter proses nya. Otomatisasi tidak saja diperlukan demi kelancaran operasi, keamanan, ekonomi maupun mutu produk, tetapi lebih merupakan kebutuhan pokok. Kelak akan dipelajari bahwa tidak mungkin menjalankan suatu proses industri tanpa bantuan sistem pengendalian, contohnya pengendalian di suatu pengilang minyak tidak mungkin dapat dijalankan tanpa bantuan fungsi sistem pengendalian.

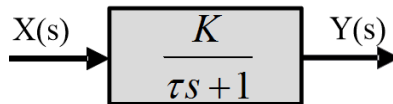
Ada banyak parameter yang harus dikendalikan di dalam suatu proses. Di antaranya, yang paling umum adalah tekanan (*pressure*) di dalam sebuah *vessel* atau pipa aliran (*flow*) di dalam pipa, suhu (*temperature*) di unit proses seperti *heat exchanger*, atau permukaan zat cair (*level*) di sebuah tangki. Ada beberapa parameter lain diluar keempat parameter di atas yang cukup penting dan juga perlu dikendalikan karena kebutuhan spesifik proses, diantaranya: pH di industri petrokimia, *water cut* di ladang minyak mentah, warna produk di suatu fasilitas pencairan gas (NGL), dan sebagainya

Gabungan serta kerja alat-alat pengendalian otomatis itulah yang dinamai sistem pengendalian proses (*process control system*). Sedangkan semua peralatan yang membentuk sistem pengendalian disebut instrumentasi pengendalian proses (*process control instrumentation*). Kedua hal tersebut berhubungan satu sama lain, namun keduanya mempunyai hakikat yang berbeda.

1.3. Karakteristik Sistem Orde Pertama [5]

Karakteristik sistem merupakan ciri – ciri khusus dari respon keluaran sistem. Ciri – ciri khusus ini disebut pula sebagai spesifikasi performansi sistem. Untuk melihat kualitas sistem dapat dilakukan

pengujian dengan mengamati keluaran respon sistem terhadap beberapa masukan sinyal uji. Sistem yang berkenaan dengan tugas akhir ini adalah respon sistem orde pertama. Karakteristik respon waktu untuk sistem orde pertama terbilang sangat sederhana. Parameter sistem orde pertama yang kerap kali jadi dasar analisis adalah *time constant* (τ). Waktu yang dibutuhkan oleh respon sistem untuk mencapai nilai 0,632 dari *set point* disebut waktu tunak atau *time constant* (τ). Adapun diagram blok umum dari sistem orde pertama tertera pada Gambar 2.2.



Gambar 2.2 Blok Diagram Umum Orde Pertama

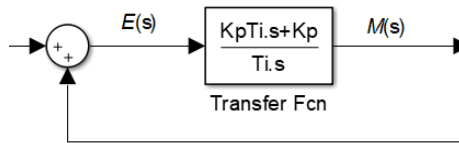
1.4. Kontroler Proporsional Integral [6] [4]

Aksi kontrol dari kontroler proporsional plus integral didefinisikan dengan persamaan (2.1) atau fungsi alih kontroler pada persamaan (2.2). Dimana K_p menyatakan kepekaan proporsional atau penguatan, dan T_i menyatakan waktu integral. Baik K_p maupun T_i dapat diatur. Waktu integral mengatur aksi kontrol integral, sedangkan K_p mempengaruhi baik bagian proporsional maupun bagian integral dari aksi kontrol. Kebalikan dari waktu integral T_i disebut laju *reset*. Laju *reset* adalah banyaknya pengulangan bagian proporsional dari aksi pengontrolan per menit. Laju *reset* diukur dalam bentuk pengulangan per menit. Gambar 2.3 menunjukkan diagram blok kontroler proporsional plus integral. Jika sinyal kesalahan $e(t)$ adalah fungsi tangga satuan, seperti ditunjukkan pada Gambar 2.4, maka keluaran kontroler $m(t)$ menjadi seperti yang ditunjukkan pada Gambar 2.5.

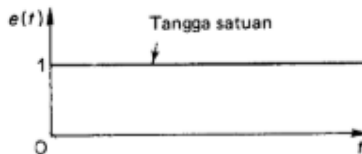
Karena kontroler proporsional plus integral merupakan gabungan dari dua unit kontrol, semua kelebihan serta kekurangan yang ada pada kontroler proporsional dan kontroler integral juga ada padanya. Sifat kontroler proporsional yang selalu meninggalkan *offset* dapat ditutupi oleh kelebihan kontroler integral, sedangkan sifat kontroler integral yang lambat dapat ditutupi oleh kontroler proporsional, sehingga kontroler proporsional plus integral menghasilkan respon yang lebih cepat dari kontroler integral tapi mampu menghilangkan *offset* yang ditinggalkan kontroler proporsional.

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.1)$$

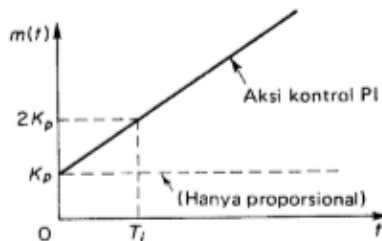
$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2.2)$$



Gambar 2.3 Diagram blok kontroler proporsional plus integral



Gambar 2.4 Diagram masukan tangga satuan

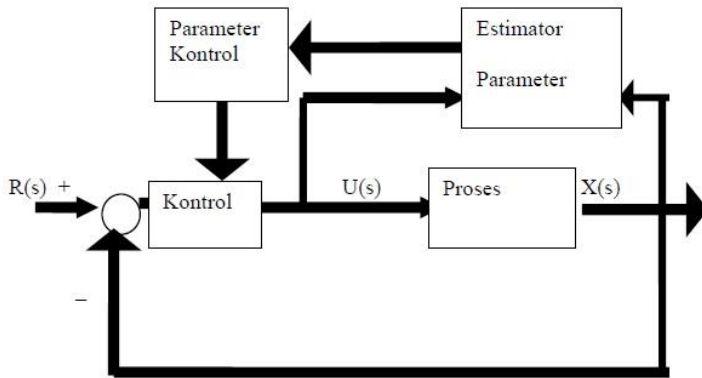


Gambar 2.5 Diagram keluaran kontroler

1.5. Sistem Pengaturan Adaptif

Dalam bahasa sehari-hari, beradaptasi berarti mengubah perilaku agar sesuai dengan keadaan baru. Secara intuitif, pengaturan adaptif merupakan sistem pengaturan yang dapat memodifikasi perilakunya sebagai respon terhadap perubahan dinamika proses dan karakter gangguan. Dalam [7] kontroler adaptif adalah kontroler dengan parameter

yang dapat disesuaikan dan memiliki mekanisme untuk menyesuaikan parameter. Parameter-parameter kontroler tersebut menyesuaikan dengan kondisi perubahan dinamika proses dan karakter gangguan agar sistem dapat berjalan stabil dan dapat mempertahankan kinerja sesuai dengan *setpoint* yang diinginkan. Banyak sistem pengaturan adaptif digunakan pada sistem nonlinier karena pada dasarnya sistem pengaturan adaptif adalah sistem pengaturan yang nonlinier. Sistem pengaturan adaptif juga sangat cocok diterapkan pada sistem yang parameteranya sulit untuk ditentukan.



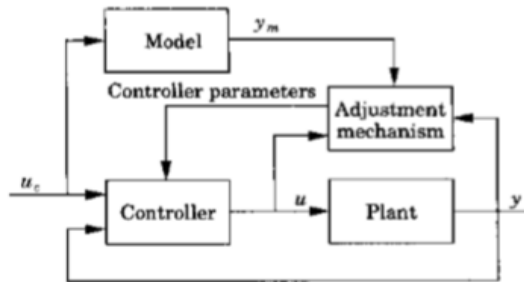
Gambar 2.6 Blok Diagram dari Sistem Pengaturan Adaptif

Pada struktur pengaturan adaptif pada Gambar 2.6 hasil dari estimasi model proses dapat menentukan parameter kontrol. Sehingga dengan menggunakan metode rekursif, model proses dapat diidentifikasi. Biasanya model proses memiliki struktur ARX atau ARMAX. Struktur ARX biasanya menggunakan metode rekursif untuk mengidentifikasi model proses. Algoritma yang biasa digunakan untuk estimasi model adalah metode *least square*. Berdasarkan struktur dan parameter model yang diperoleh melalui langkah estimasi, struktur dan parameter kontrol ditentukan, bergantung metode kontrol yang digunakan. Berbagai pendekatan dapat dilakukan untuk mendapatkan respon yang sesuai. Pada kontroler PID misalnya, parameter kontroler akan tetap selama model proses tidak berubah, tetapi akan berubah sesuai dengan parameter yang diinginkan mengikuti perubahan parameter model proses. Adapun sistem

pengaturan adaptif dimana parameter kontroler diestimasi langsung sesuai dengan keadaan model proses yang di kontrol.

1.6. MRAC [7]

Model Reference Adaptive Control (MRAC) pada awalnya digunakan untuk menyelesaikan masalah di mana spesifikasi diberikan dalam bentuk model referensi yang memberi tahu bagaimana keluaran proses harus merespon sinyal perintah. Diagram blok sistem ditunjukkan pada Gambar 2.7. Kontroler dapat dianggap terdiri dari dua *loop*. *Loop* dalam adalah *loop* umpan balik biasa yang terdiri dari proses dan kontroler. *Loop* luar adalah *loop* untuk menyesuaikan parameter kontroler sedemikian rupa sehingga eror yang merupakan perbedaan antara keluaran proses y dan keluaran model y_m menjadi minimum. MRAC awalnya diperkenalkan untuk kontrol penerbangan. dalam hal ini model referensi menggambarkan respon yang diinginkan dari pesawat terhadap gerakan *joystick*. Masalah utama pada MRAC adalah untuk menentukan mekanisme adaptasi sehingga sistem dapat berjalan dengan stabil dan selisih nilai keluaran model dengan keluaran proses menjadi nol.



Gambar 2.7 Blok Diagram dari MRAC

1.7. Neural Network (Jaringan Saraf Tiruan) [8]

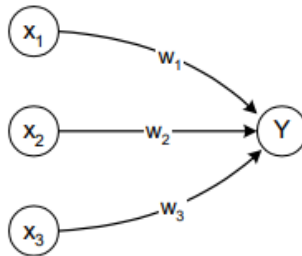
Neural Network atau Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. *Neural Network* dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*). Lalu sinyal dikirimkan diantara *neuron* melalui penghubung. Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah

sinyal. Sehingga untuk menentukan keluaran, setiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada jumlah masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang. *Neural Network* ditentukan oleh 3 hal :

- Pola hubungan antar *neuron* (disebut arsitektur jaringan)
- Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
- Fungsi aktivasi

Seperti yang terlihat pada Gambar 2.8, *neuron Y* menerima masukan dari *neuron* x_1 , x_2 dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Ketiga impuls *neuron* yang ada dijumlahkan seperti pada persamaan (2.4). Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 \quad (2.4)$$

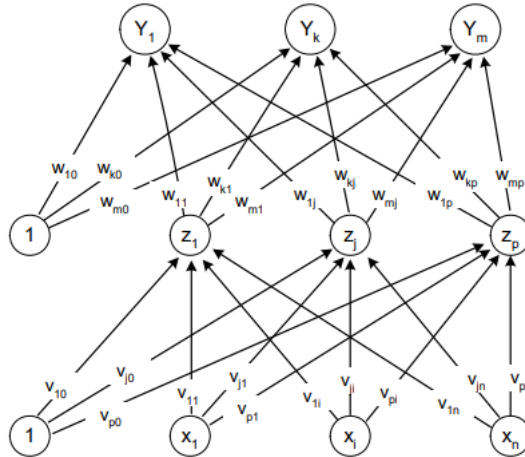


Gambar 2.8 Hubungan antar *neuron*

a. *Backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih *hidden layer*. Gambar 2.9 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah *hidden layer* yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran. v_{ji} merupakan bobot garis dari unit masukan x_i ke unit *hidden layer* z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit

hidden layer z_j). w_{kj} merupakan bobot dari unit *hidden layer* z_j ke unit keluaran Y_k (w_{k0} merupakan bobot dari bias di *hidden layer* ke unit keluaran z_k).



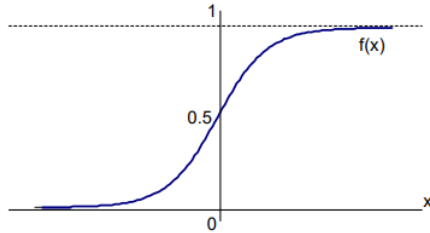
Gambar 2.9 Arsitektur *Backpropagation*

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi *sigmoid biner* yang memiliki range 0 hingga 1 seperti pada persamaan (2.5) dengan turunannya seperti pada persamaan (2.6).

Fungsi sigmoid memiliki nilai maksimum 1 seperti pada Gambar 2.10. Maka untuk pola yang targetnya > 1 , pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi *sigmoid* yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi *sigmoid* hanya pada hidden layer yang bukan unit keluaran. Pada unit keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas : $f(x) = x$.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

$$f'(x) = f(x)(1 - f(x)) \quad (2.6)$$



Gambar 2.10 Grafik Fungsi Aktivasi Sigmoid

b. Pelatihan Standar *Backpropagation*

Pelatihan *Backpropagation* meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari unit masukan hingga unit keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di unit keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

- Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan x_i dipropagasikan ke *hidden layer* menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit *hidden layer* z_j tersebut selanjutnya dipropagasikan maju lagi ke *hidden layer* di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan Y_k . Berikutnya, keluaran jaringan Y_k dibandingkan dengan target yang harus dicapai T_k . Selisih $T_k - Y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

- Fase II : Propagasi mundur
 Berdasarkan kesalahan $T_k - Y_k$, dihitung faktor δ_k ($k=1,2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit Y_k ke semua unit tersembunyi yang terhubung langsung dengan Y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor δ_j di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.
- Fase III : Perubahan Bobot
 Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ *neuron* di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

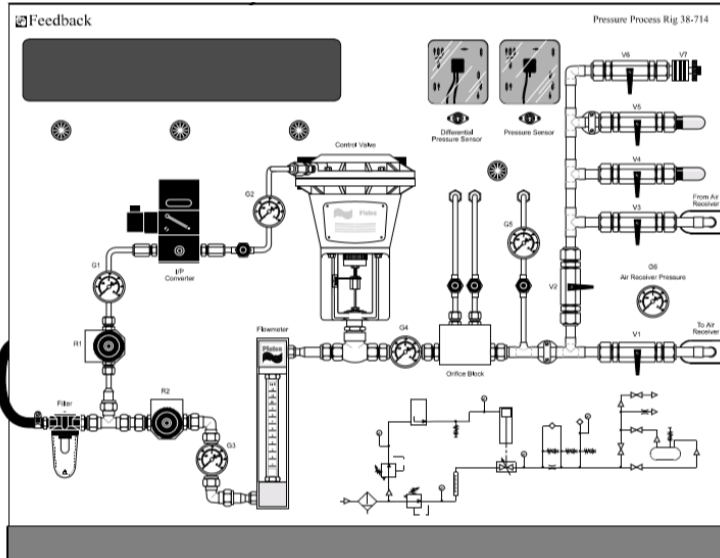
[Halaman ini sengaja dikosongkan]

BAB 3 PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perancangan sistem pengaturan *pressure process* pada rig 38-714 yang terdiri dari beberapa tahapan, yaitu konfigurasi *plant*, proses identifikasi, desain kontroler dan perancangan kontroler.

3.1. Konfigurasi *Plant*

Plant yang digunakan untuk sistem pengaturan proses pada tugas akhir ini adalah *plant Pressure Process Rig 38-714* yang berada di lab B105 Institut Teknologi Sepuluh Nopember Surabaya. Digunakannya *plant rig 38-714* ini karena instrumen yang terdapat pada *rig 38-714* dapat mewakili instrumentasi pada industri nyata. *Plant* yang digunakan dapat dilihat pada Gambar 3.1, yang terdiri dari 6 buah *valve* manual. Tiga *valve* manual sebagai *valve* pembebanan yaitu V4, V5, dan V6 yang dimana pembukaan dari dua *valve* V4 dan V5 ini sebagai kondisi operasi pengendalian *plant*.

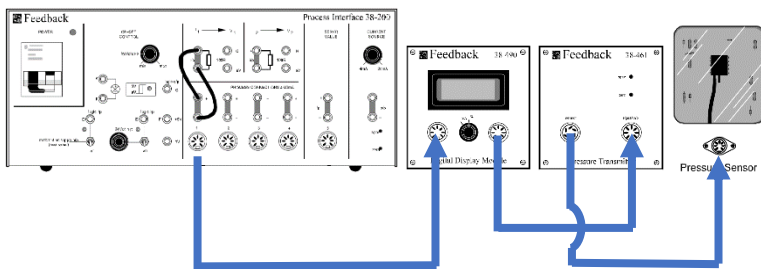


Gambar 3.1 *Pressure Process Rig 38-714*

3.1.1. Kalibrasi Sensor

Dilakukannya kalibrasi sensor pada *pressure process rig* 38-714 bertujuan untuk mengetahui apakah sensor dapat berjalan dengan baik atau tidak. Pada *plant rig* 38-714, sensor *pressure* dapat berjalan dengan baik apabila pada saat *pressure* mencapai nilai minimal atau sama dengan nol psi maka arus yang dikeluarkan dari *pressure trasmitter* harus setara atau mencapai 4mA. Dan ketika *pressure* mencapai nilai maksimal (10psi) maka arus keluaran dari *pressure trasmitter* harus mencapai atau setara dengan 20mA. Adapun langkah untuk proses kalibrasi agar sensor *pressure* mencapai kondisi tersebut antara lain:

- Hubungkan perangkat *process interface* 38-200, *digital display module*, *pressure transmitter* dan *pressure sensor* pada *rig* 38-714 sesuai dengan skema *wiring* pada Gambar 3.2.
- Buang semua udara yang ada dalam sistem dengan membuka katup V2, V4 dan V5.
- Setelah menunggu beberapa saat, sistem akan membuang udara pada sehingga *pressure* bernilai 0 psi. *Digital module display* harus menunjukkan angka 4mA atau 0%.
- Buka katup V1 dan tutup katup V2 untuk memberikan tekanan pada *air receiver* lalu atur R2 agar menghasilkan pembacaan 0.7 bar atau 10 psi pada G3.
- Setelah *air receiver* terisi dengan udara bertekanan, *digital module display* harus terbaca 20mA.
- Jika tidak menunjukkan 20mA atur *span control* pada *pressure transmitter* sehingga menunjukkan 20 mA
- Lakukan pengulangan dari langkah pertama sehingga pada saat keadaan 0% dan 100% sesuai dengan 4 – 20 mA.

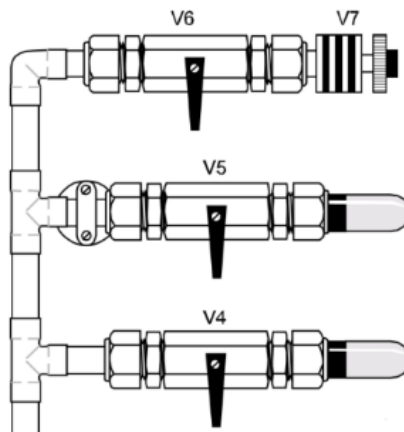


Gambar 3.2 *Wiring* Kalibrasi Sensor *Pressure*

3.1.2. Metode Pembebanan

Metode pembebanan yang dilakukan pada tugas akhir ini dengan mengubah ubah kondisi 2 *valve* manual V5 dan V4 seperti yang terlihat pada Gambar 3.3. Kondisi beban nominal yang dipilih adalah dengan membuka *valve* manual V4 dan V5 serta menutup *valve* manual V6. Kondisi *valve* manual V6 harus berada pada keadaan tertutup, karena jika *valve* manual V6 dibuka dapat menyebabkan sistem tidak dapat mencapai kondisi *setpoint* yang diinginkan. Kondisi perubahan beban pada tugas akhir ini dapat dilakukan dengan 2 kondisi pembebanan, yaitu :

- Kondisi Pembebanan 1 : *Valve* manual V6 dan V5 ditutup, *valve* manual V4 dibuka
- Kondisi Pembebanan 2 : *Valve* manual V6 dan V4 ditutup, *valve* manual V5 dibuka



Gambar 3.3 *Valve* Manual untuk Pembebanan

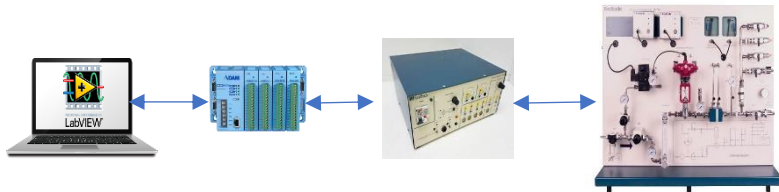
3.2. Identifikasi *Plant*

Untuk mengetahui karakteristik dari *plant pressure process rig* 38-714 diperlukan identifikasi *plant* dengan cara uji *open loop*. Hasil dari identifikasi *plant* ini adalah model matematika dari *pressure process rig* 38-714. Model matematika dari *plant pressure process rig* 38-714 yang diperoleh dapat dilakukan analisis sehingga didapat perilaku dan karakteristik *plant*, kemudian dapat dirancang kontroler yang sesuai

dengan *plant* tersebut. Ada beberapa tahapan identifikasi *plant* yaitu instalasi perangkat, konfigurasi perangkat, perancangan program, dan metode identifikasi.

3.2.1. Instalasi Perangkat

Sebelum melakukan identifikasi, dilakukan instalasi perangkat terlebih dahulu agar perangkat yang digunakan dapat berjalan dengan baik serta pengambilan data *plant* yang benar, sehingga data yang diperoleh sesuai dengan data kondisi *plant*. Diagram instalasi perangkat dalam tugas akhir ini dapat dilihat pada Gambar 3.4.

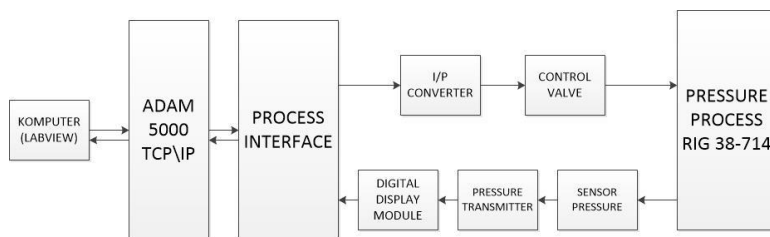


Gambar 3.4 Diagram Instalasi Perangkat pada Sistem Pengaturan *Pressure Process Rig 38-714*

Untuk instalasi perangkat pada sistem pengaturan *Pressure Process Rig 38-714*, komputer yang digunakan sebagai pemberi sinyal uji tidak langsung berinteraksi dengan *plant Pressure Process Rig 38-714*, tetapi harus terhubung dengan 2 perangkat lainnya terlebih dahulu yaitu ADAM 5000 TCP/IP dan *Process Interface 38-200*. ADAM 5000 TCP/IP digunakan sebagai DAQ (*Data Acquisition System*) yang terhubung pada *Process Interface 38-200* yang berfungsi mengirimkan sinyal guna menggerakkan *control valve* dan menerima sinyal pembacaan dari *pressure trasnmmitter* lalu dikonversi menjadi tegangan agar bisa terbaca oleh ADAM 5000 TCP/IP.

Seperti yang terlihat pada Gambar 3.5, nilai *setpoint* yang dikirimkan dari komputer berupa sinyal digital melalui kabel UTP RJ45 yang menghubungkan antara komputer dengan ADAM 5000 TCP/IP sebagai DAQ guna mengolah sinyal digital menjadi sinyal analog. Sinyal analog yang digunakan yaitu 4-20mA. Sinyal analog dengan satuan arus tidak dapat langsung menggerakkan *control valve*, melainkan harus dikonversi menjadi tekanan terlebih dahulu dengan komponen *I/P Converter*. *I/P Converter* mengolah nilai satuan arus menjadi tekanan sehingga dapat berfungsi untuk membuka atau menutup *control valve*.

Untuk proses pengambilan data identifikasi dilakukan dengan pembacaan sensor *pressure* yang terhubung dengan *pressure transmitter*. Kemudian ditampilkan pada *digital display module* dan terhubung dengan *Process Interface*. *Process interface* mengkonversi nilai arus hasil pembacaan dari *pressure transmitter* ke dalam satuan tegangan. Sinyal analog dengan satuan tegangan tersebut dihubungkan dengan ADAM 5000 TCP/IP kemudian dikonversi menjadi sinyal digital yang dapat dibaca oleh komputer.



Gambar 3.5 Diagram Pengambilan Data *Pressure Process Rig 38-714*

Berikut spesifikasi perangkat yang digunakan pada sistem pengaturan *Pressure Process Rig 38-714* :

- Komputer
Spesifikasi komputer yang digunakan pada tugas akhir ini adalah:

<i>Processor</i>	: Intel Celeron B877 1.4GHz
RAM	: DDR3 4GB
<i>Operating System</i>	: Windows 10 32-bit
Resolusi Tampilan	: 1366x768x60
USB Port	: 3 x USB 2.0
<i>Ethernet Port</i>	: Yes
- ADAM – 5000/TCP

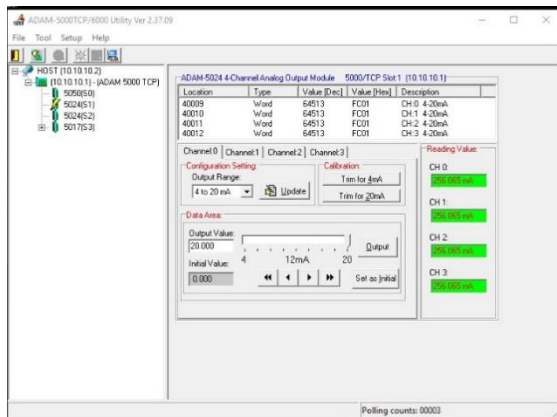
CPU	: ARM 32-bit RISC CPU
<i>Memory</i>	: 4 MB <i>Flash</i> RAM
<i>Operating System</i>	: <i>Real-Time</i> O/S
<i>Timer BIOS</i>	: Yes
<i>I/O Capacity</i>	: 8 <i>slots</i> (ADAM-5000/TCP) 4 <i>slots</i> (ADAM-5000/TCP)
Status Indikator	: <i>Power, CPU, Communication</i>

Ethernet : 10 BASE-T IEEE 802.3
 100 BASE-TX IEEE 802.3u
Wiring : UTP
Koneksi Bus : RJ45 *jack modular*
Protokol Komunikasi : *Modbus/TCP*
Transfer Rate : sampai dengan 100Mbps
Waktu Respon : kurang dari 5ms
Komunikasi : Serial
Sinyal RS-485 : DATA +, DATA –
Mode : *Half Duplex, Multi Drop*
Konektor : *screw terminal*
Kecepatan Transmisi : sampai dengan 115,2 Kbps
Power Unregulated : 10V sampai 30V DC
Proteksi : *Over Voltage* dan *Power Reversal*
Temperatur Operasi : -10 sampai 70°C
Kelembapan : 5 sampai 95%, *non-condensing*
Atmosfir : tidak ada gas korosif

- *Pressure Process Rig* 38-714
 - Masukan Pressure* : tidak melebihi 25 psi
 - Air Filter* : *Yes*
 - Pressure Regulator* : 2 pcs
 - Pressure Gauge* : 6 pcs
 - I/P Converter* : 4-20 mA to 0-25 psi
 - Flowmeter* : *Yes*
 - Control Valve* : 0 – 25 psi
 - Orifice Block* : *Yes*
 - Air Receiver* : *Yes*
 - Manual Valve* : 6 pcs
 - Pressure Sensor* : *Yes*
 - Differential* : *Yes*
 - Pressure Sensor* : *Yes*
 - Digital Display* : *Yes*
 - Module* : *Yes*
 - Pressure* : *Yes*
 - Transmitter* : *Yes*
 - Differential* : *Yes*
 - Pressure* : *Yes*
 - Transmitter* : *Yes*

3.2.2. Konfigurasi Perangkat

Setelah mengetahui spesifikasi dan konfigurasi *plant*, agar proses identifikasi berjalan dengan baik, diperlukan konfigurasi perangkat DAQ (*Data Aquisition*) terlebih dahulu. DAQ yang digunakan adalah ADAM 5000 TCP/IP dengan kabel UTP RJ45 sebagai penghubung antara DAQ dengan komputer. ADAM 5000 TCP/IP memiliki *slot* modul masukan dan *keluaran*. *Slot* modul masukan terdiri dari 8 kanal dengan satuan *volt* dan *millivolt*. *Slot* modul *keluaran* memiliki 4 kanal *keluaran* dengan satuan *volt* dan arus. Pada tugas akhir ini, *slot* masukan yang digunakan adalah masukan dengan satuan *volt* dengan tegangan kerja yang digunakan $\pm 10V$. Modul masukan ini digunakan untuk membaca sinyal keluaran dari *pressure* sensor yang mendapat *signal conditioning* terlebih dahulu pada perangkat *process interface*. Pada modul *keluaran*, satuan yang digunakan berupa arus 4-20mA yang dipakai sebagai sinyal kontrol pada I/P *converter* yang akan dikonversi menjadi *pressure* untuk menggerakkan *control valve* pada *plant*.



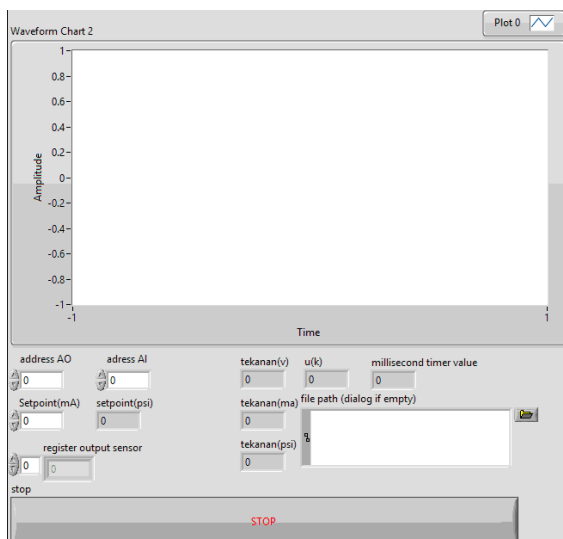
Gambar 3.6 Software ADAM-5000TCP-6000 Utility

Pada konfigurasi perangkat DAQ ini diperlukan juga proses *scaling* dimana proses tersebut untuk menyatakan representasi data masukan dan *keluaran* yang digunakan. Pada modul masukan konfigurasi yang digunakan adalah -10 *volt* untuk nilai paling rendah dan 10 *volt* untuk nilai yang paling tinggi dengan format data 16 bit sehingga data yang diterima

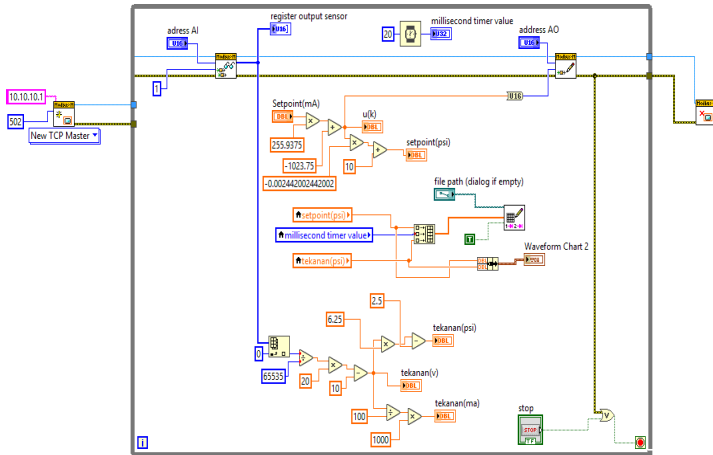
maksimal 65535. Pada modul *keluaran* konfigurasi yang digunakan adalah 4mA untuk nilai paling rendah dan 20mA untuk nilai paling tinggi dengan format data 12 bit sehingga data yang dapat dikirimkan maksimal 4095. Untuk melakukan konfigurasi tersebut digunakan *software* ADAM Utility. Seperti yang terlihat pada Gambar 3.6, pada *software* tersebut dapat dilakukan pemilihan tipe masukan atau *keluaran* yang dipakai serta dapat dilakukan kalibrasi pada perangkat.

3.2.3. Program Identifikasi Pada NI LabVIEW 2018

Pada proses identifikasi *plant* diperlukan suatu *software* pada komputer yang digunakan untuk mengolah data dari hubungan masukan-*keluaran plant*. Pada tugas akhir ini LabVIEW digunakan sebagai *software* untuk memberikan sinyal uji dan menyimpan data masukan-*keluaran* hasil dari identifikasi untuk kemudian data tersebut diolah menggunakan *software* MATLAB sehingga dapat ditentukan model matematika dari *plant Pressure Process Rig 38-714*. Pada Gambar 3.7 dan Gambar 3.8 merupakan tampilan program dan blok diagram program yang digunakan untuk proses identifikasi *plant* pada tugas akhir ini.



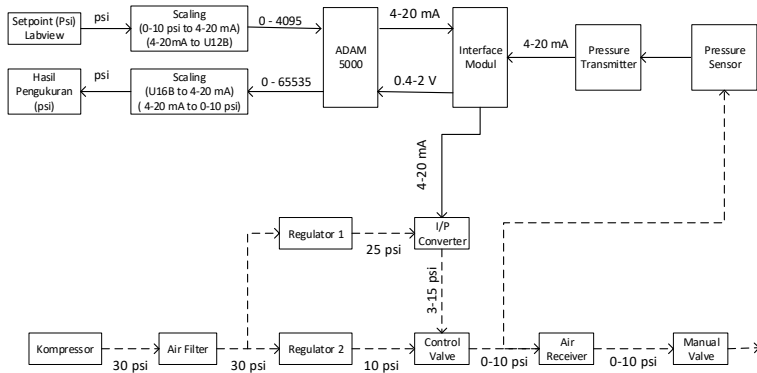
Gambar 3.7 Tampilan *Front Panel* Identifikasi *Plant* NI LabVIEW 2018



Gambar 3.8 Tampilan Program Identifikasi *Plant* NI LabVIEW 2018

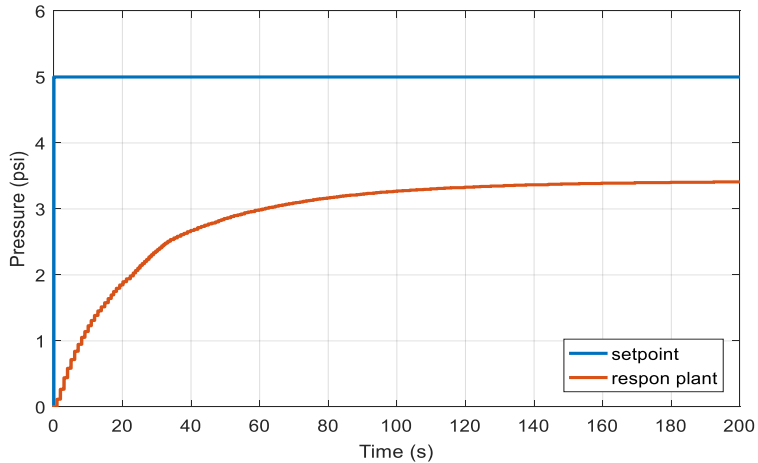
3.2.4. Identifikasi Sistem

Setelah semua perangkat telah dikonfigurasi dengan baik, proses identifikasi dapat dilakukan. Untuk mendapatkan model matematika dari *plant Pressure Process Rig 38-714*, dilakukan proses identifikasi sistem dengan menjalankan *plant* secara *open loop*. Skema identifikasi sistem yang dilakukan pada tugas akhir ini terlihat seperti pada Gambar 3.9. Metode yang digunakan adalah metode identifikasi statis.

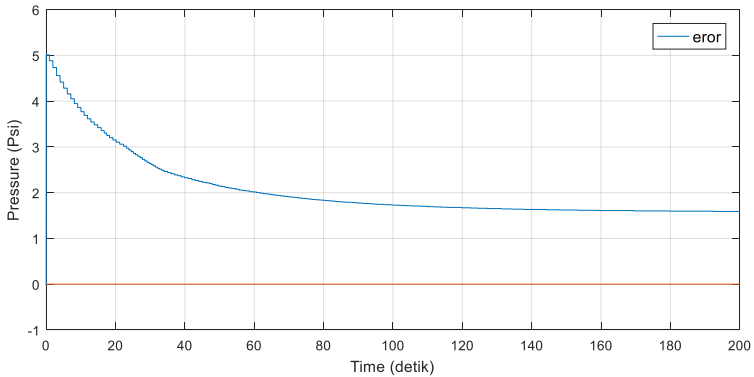


Gambar 3.9 Skema Identifikasi *Plant*

Metode identifikasi statis dilakukan dengan memberikan *set value* berupa sinyal uji *step* pada *plant*. Respon dari *plant* dibaca oleh *pressure* sensor. Hasil pembacaan *pressure* sensor lalu dikumpulkan bersama dengan *set value* yang diberikan pada *plant*. kemudian data tersebut dapat dianalisa dan didapatkan model matematika dari *plant*. Dari *set value* yang diberikan pada keadaan nominal *plant* berupa sinyal uji unit *step*, dapat dilihat respon *plant* pada Gambar 3.10 dan error yang terjadi pada Gambar 3.11.



Gambar 3.10 Respon *Plant Open Loop* Terhadap Sinyal Uji *Step*



Gambar 3.11 Error keluaran *plant* terhadap masukan

Setelah melihat respon sistem hasil identifikasi *plant* secara *open loop*, dapat ditentukan model pendekatan sistem yang akan digunakan. Respon *plant* yang dihasilkan mempunyai kemiripan dengan sistem orde satu. *Delay* pada sistem dapat diabaikan karena nilainya sangat kecil. Maka dari itu model sistem yang digunakan adalah pendekatan sistem orde satu yang dinyatakan dengan fungsi alih pada persamaan (3.1).

$$G(s) = \frac{K}{\tau s + 1} \quad (3.1)$$

Dari persamaan (3.1) dapat diketahui bahwa parameter K merupakan *gain overall* dari sistem dan τ merupakan *time constan* dari sistem. Untuk mencari parameter K dilakukan perhitungan dengan persamaan (3.2), dimana Y_{ss} merupakan respon keluaran sistem dalam keadaan *steady state* dan X_{ss} respon masukan sistem dalam keadaan *steady state*.

$$K = \frac{Y_{ss}}{X_{ss}} \quad (3.2)$$

Melihat dari respon keluaran *plant* pada Gambar 3.10, keluaran sistem dalam keadaan *steady state* pada nilai 3.391 Psi sedangkan kondisi masukan dalam keadaan *steady state* pada nilai 5 Psi. Dengan demikian nilai *gain overall* dapat dihitung menggunakan persamaan (3.2).

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{3,391}{5} = 0.6782$$

Untuk mendapatkan nilai *time constan* dapat dicari ketika respon keluaran sistem mencapai 63.2% dari keadaan *steady state* seperti pada persamaan (3.3). nilai *steady state* pada keluaran sistem adalah 3.391, sehingga 63.2% dari respon keluaran *steady state* adalah 2.14 psi. untuk mencapai nilai 2.14 psi dibutuhkan waktu selama 25.52 detik. Dengan demikian fungsi alih yang didapat dinyatakan pada persamaan (3.4).

$$\tau = 63.2\% \times Y_{ss} \quad (3.3)$$

$$G(s) = \frac{0,6782}{25,52s + 1} \quad (3.4)$$

3.3. Validasi Model

Setelah mendapatkan fungsi alih dari *plant*, dilakukan validasi model agar diketahui kebenaran nilai dari model yang didapat. Validasi dilakukan dengan menggunakan kriteria RMSE (*Root Mean Square Error*) yang dinyatakan pada persamaan (3.5). Semakin kecil nilai RMSE maka akan semakin baik model yang dibuat. Pada tugas akhir ini dilakukan 6 percobaan identifikasi *plant* seperti yang terlihat pada Tabel 3.1.

Tabel 3.1 Persamaan Model *Plant*

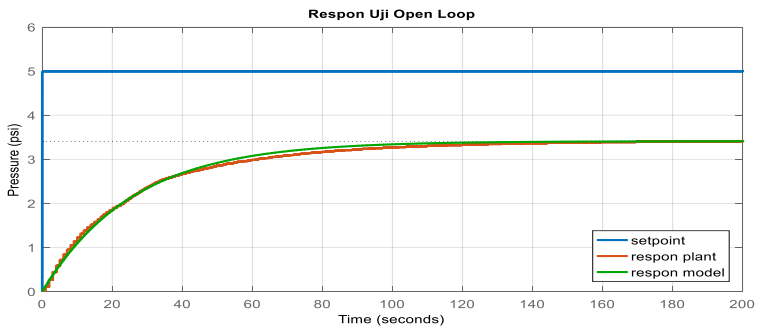
No.	Persamaan Model Plant	RMSE
1	$G(s) = \frac{0,6782}{25,52s + 1}$	0.0765
2	$G(s) = \frac{0,682}{25,77s + 1}$	0.0561
3	$G(s) = \frac{0,682}{27,87s + 1}$	0.0656
4	$G(s) = \frac{0,6232}{27,06s + 1}$	0.6392
5	$G(s) = \frac{0,6232}{26,98s + 1}$	0.6423
6	$G(s) = \frac{0,626}{23,46s + 1}$	0,5890

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2} \quad (3.5)$$

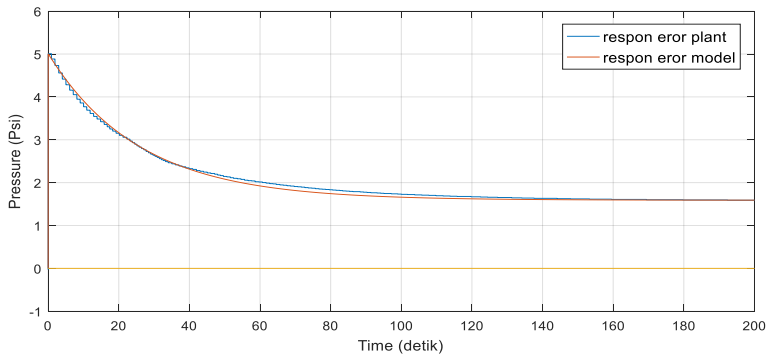
Mengacu pada Tabel 3.1, persamaan model *plant* yang digunakan untuk desain kontroler pada tugas akhir ini adalah persamaan model dengan nilai RMSE terkecil. Maka model yang digunakan untuk desain kontroler adalah model *plant* pada persamaan (3.6).

$$G(s) = \frac{0.682}{25,77s + 1} \quad (3.6)$$

Setelah mendapatkan persamaan model yang diinginkan, dilakukan pengujian model terlebih dahulu sebelum melakukan desain kontroler. Untuk menguji model tersebut maka dilakukan perbandingan grafik antara persamaan model *plant* dengan respon *plant*. Seperti yang terlihat pada Gambar 3.12 dapat disimpulkan bahwa persamaan model dapat menyerupai respon *plant* dengan perbandingan nilai error pada Gambar 3.13. Dengan demikian bahwa identifikasi yang dilakukan sudah mendekati *plant* yang sebenarnya.



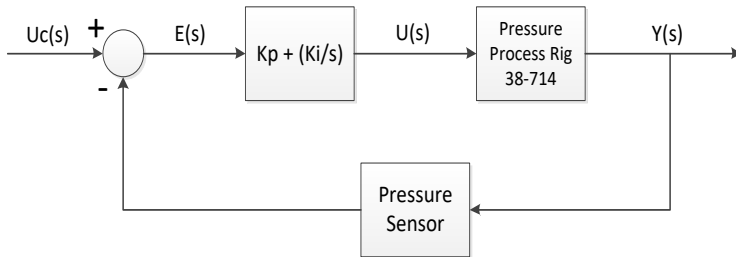
Gambar 3.12 Perbandingan Respon *Plant* dengan Persamaan Model *Plant*



Gambar 3.13 Perbandingan error keluaran *plant* terhadap masukan dan error keluaran model terhadap masukan

3.4. Perancangan Kontroler PI

Mengacu pada persamaan model *plant* yang didapatkan melalui identifikasi *plant* yang dinyatakan dengan fungsi alih pada persamaan (3.6), masih terdapatnya E_{ss} sekitar 31,8% dengan $\tau = 25.77$ detik. Untuk itu diperlukan sebuah kontroler untuk mempercepat τ dan meminimalkan E_{ss} . Kontroler yang dapat mempercepat respon dan menghilangkan E_{ss} adalah kontroler PI yang dinyatakan dalam persamaan (3.7). Untuk merancang kontroler PI digunakan metode analitik dengan menentukan spesifikasi respon yang diinginkan. Diharapkan respon dapat lebih cepat dengan $E_{ss} = 0\%$. Diagram blok sistem pengaturan proses tekanan dengan kontroler PI dapat ditunjukkan pada Gambar 3.12.



Gambar 3.14 Diagram Blok Sistem Pengaturan *Pressure* dengan Kontroler PI

$$\frac{U(s)}{E(s)} = Kp + \frac{Ki}{s} \quad (3.7)$$

Dari persamaan (3.6) diperoleh nilai *gain overall* = 0.682, dengan $\tau = 25.77$ dan *settling time* = 77.31 detik. Dengan nilai *gain overall* yang diinginkan = 1 dan *settling time* yang diinginkan = 50 detik maka didapatkan nilai KP dengan mengetahui nilai τ^* pada persamaan (3.8).

$$t_s(\pm 5\%) = 3 \tau^* \quad (3.8)$$

$$50 = 3 \tau^*$$

$$\tau^* = 16,66$$

Setelah mendapatkan nilai τ^* lalu mencari nilai K_p dengan persamaan (3.9), dimana $\tau_i = \tau$.

$$K_p = \tau_i / \tau^* K = 2,26 \quad (3.9)$$

Setelah mendapatkan nilai K_p maka nilai K_i dapat dicari dengan persamaan (3.10).

$$K_i = \frac{K_p}{\tau_i} = 0,087 \quad (3.10)$$

Sehingga persamaan kontroler seperti yang dinyatakan pada persamaan (3.11)

$$\frac{U(s)}{E(s)} = 2,26 + \frac{0,087}{s} \quad (3.11)$$

Dilakukan diskritisasi dengan metode *backward euler* seperti yang terlihat pada persamaan (3.12), sehingga persamaan PI dapat dinyatakan pada persamaan (3.13), dimana T_s merupakan *time sampling*.

$$\frac{U(z)}{E(z)} = K_p + \frac{K_i T_s z}{z - 1} \quad (3.12)$$

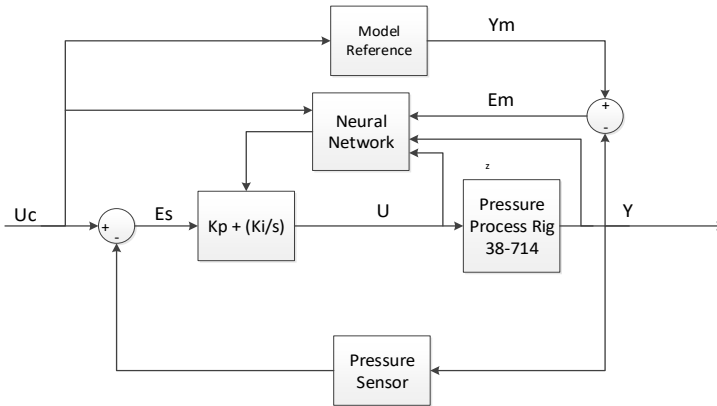
$$U(k) = U(k - 1) + K_p E(k) - K_p E(k - 1) + K_i T_s E(k) \quad (3.13)$$

3.5. Perancangan Kontroler PI dengan *Model Reference*

Agar kontroler dapat berjalan baik ketika ada gangguan berupa perubahan beban maupun perubahan masukan, maka dibutuhkan skema pengaturan adaptif. Sistem pengaturan adaptif dirancang agar kontroler dapat melakukan adaptasi ketika perubahan parameter terjadi. Pada tugas akhir ini skema adaptif yang digunakan adalah MRAC dengan *Neural Network* sebagai algoritma untuk mendapatkan nilai K_p dan K_i . Diagram blok skema pengaturan adaptif yang digunakan pada tugas akhir ini dapat dilihat pada Gambar 3.13.

Perancangan diawali dengan menentukan model referensi yang akan digunakan. Sebelumnya telah diketahui persamaan kontroler PI (3.11) dan persamaan plant (3.6) yang akan digunakan, sehingga setelah

persamaan kontroler dan persamaan *plant* dimasukkan, maka akan menjadi persamaan (3.14). Dengan memasukkan nilai K_p dan K_i maka diperoleh persamaan (3.15). Persamaan (3.15) adalah persamaan yang akan digunakan untuk model referensi dari perancangan kontroler pada tugas akhir ini.



Gambar 3.15 Diagram Blok Kontroler PI berbasis *Neural Network* dengan Skema MRAC

$$\frac{Y(s)}{Uc(s)} = \frac{0,682 Kp s + 0,682 Ki}{25,77 s^2 + (0,682 Kp + 1)s + 0,682 Ki} \quad (3.14)$$

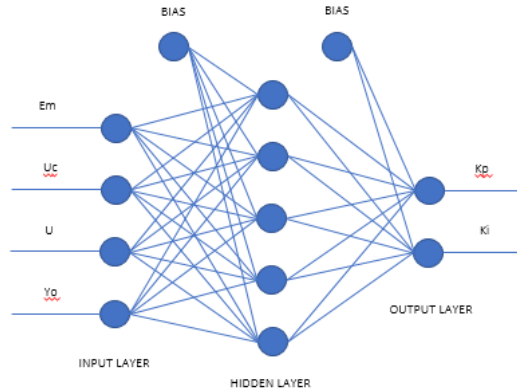
$$\frac{Y(s)}{Uc(s)} = \frac{1,54 s + 0,059}{25,77 s^2 + 2,54s + 0,059} \quad (3.15)$$

3.6. Perancangan Struktur *Neural Network*

Perancangan Struktur *Neural Network* untuk dapat melakukan *tuning* pada kontroler PI dapat dilakukan dengan langkah sebagai berikut:

- Menentukan algoritma pembelajaran yang akan digunakan pada *Neural Network*. Algoritma pembelajaran yang digunakan pada perancangan kontroler ini adalah *backpropagation*.
- Menentukan struktur dari *Neural Network*, seperti jumlah *layer* yang digunakan dan jumlah *neuron* pada setiap *layer*

- c. Melakukan inialisasi nilai bobot yang digunakan pada *Neural Network* pada bagian antara masukan *layer* menuju *hidden layer* dan *hidden layer* menuju keluaran *layer*

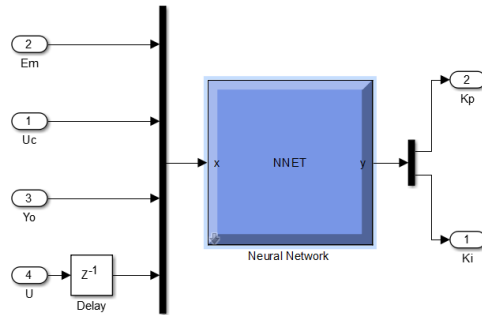


Gambar 3.16 Struktur *Neural Network*

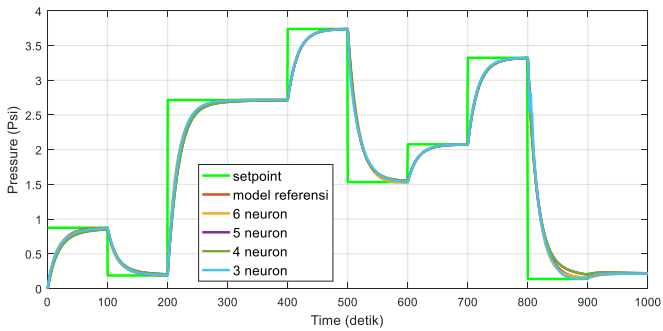
Struktur *Neural Network* pada tugas akhir ini seperti pada Gambar 3.16 terdiri dari 3 *layer* yaitu *layer* masukan, *hidden layer* dan *layer* keluaran. Pada *layer* masukan terdapat 4 *neuron* sebagai masukan yaitu Em sebagai eror model, Y sebagai keluaran *plant*, U sebagai sinyal kontrol dan Uc sebagai *setpoint*. Sementara pada *layer* keluaran terdapat 2 *neuron* untuk nilai Kp dan Ki. Untuk mencari struktur terbaik, dilakukan pengujian untuk menentukan jumlah *neuron* pada *hidden layer* dengan diagram blok seperti pada Gambar 3.17 dengan hasil pada Gambar 3.18 dimana hasil terbaik didapat pada struktur dengan 5 *neuron* pada *hidden layer* dengan nilai RMSE terkecil yang dapat dilihat pada Tabel 3.2. Pada setiap *neuron* terdapat bias dimana bias berfungsi sebagai nilai untuk memperbaiki keluaran agar sesuai dengan nilai yang diinginkan.

Tabel 3.2 Nilai RMSE perbandingan jumlah *neuron* pada *hidden layer*

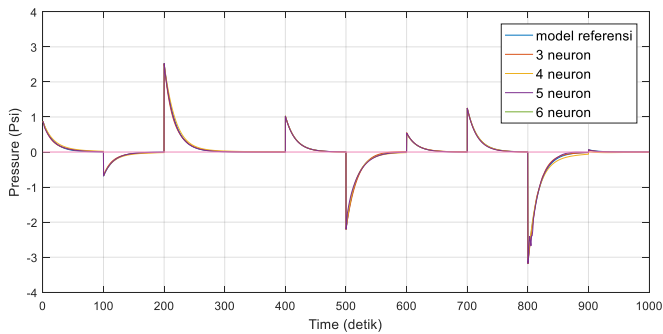
Jumlah <i>Neuron</i>	RMSE
3	0.0284
4	0.0282
5	0.0273
6	0.0283



Gambar 3.17 Neural Network dengan Toolbox Matlab Simulink



Gambar 3.18 Perbandingan hasil simulasi dengan variasi jumlah neuron pada hidden layer



Gambar 3.19 Respon perbandingan error hasil simulasi dengan perubahan jumlah neuron pada hidden layer

Pada tugas akhir ini terdapat dua algoritma yang digunakan pada struktur *Neural Network* yaitu algoritma *feedforward* dan algoritma *backpropagation*. Pada algoritma *feedforward* dilakukan perhitungan maju dari *layer* masukan menuju *layer* keluaran guna mendapatkan nilai keluaran. Sedangkan pada algoritma *backpropagation* digunakan untuk merevisi bobot tersebut agar sesuai dengan nilai parameter K_p atau K_i yang diinginkan. Pada algoritma *feed forward* ini M merupakan jumlah variabel masukan pada *layer* masukan, pada struktur *Neural Network* ini digunakan empat variabel masukan yaitu E_m yaitu selisih antara nilai model referensi dan keluaran dari *plant*, Y yaitu keluaran dari *plant*, *setpoint* yang diberikan pada *plant* dan nilai sinyal kontrol. Masukan pada *hidden layer* dapat dituliskan seperti pada Persamaan (3.16) dan keluaran dari *hidden layer* seperti pada Persamaan (3.17).

$$z_i = \sum_{i=0}^M w_i x_i \quad (i = 1, 2, \dots, M) \quad (3.16)$$

$$O_h = f(z_i) \quad (3.17)$$

$$f(z_i) = \lambda z_i \quad (3.18)$$

Dimana f merupakan fungsi aktivasi pada *hidden layer*. Fungsi aktivasi yang digunakan adalah fungsi aktivasi linier seperti pada persamaan (3.19). w_i merupakan nilai bobot pada jaringan antara *layer* masukan menuju *hidden layer* dan λ merupakan faktor penguat. Untuk masukan dan keluaran pada *layer* keluaran dapat dituliskan dalam persamaan matematis seperti pada Persamaan (3.19) dan (3.20)

$$z_o = \sum_{i=0}^1 w_o O_h \quad (3.19)$$

$$O_k = f(z_o) \quad (3.20)$$

Keluaran dari *layer* keluaran ini merupakan nilai parameter kontroler PI yaitu K_p atau K_i . Kemudian dilakukan mekanisme *backpropagation* untuk merevisi nilai bobot pada *Neural Network*. Pada algoritma ini fungsi eror atau indeks performa pada keluaran sistem

didefinisikan seperti pada Persamaan (3.21) dimana eror yang digunakan adalah selisih antara keluaran model referensi dengan keluaran *plant*.

$$E = \frac{1}{2} em^2 \quad (3.21)$$

Perubahan bobot pada *hidden layer* menuju keluaran *layer* seperti pada Persamaan (3.22)

$$w_o = w_o + \eta \frac{\partial E}{\partial w_o} \quad (3.22)$$

Dimana η merupakan *learning rate* yang merupakan parameter laju pembelajaran pada algoritma *Neural Network* yang mempengaruhi kecepatan algoritma *Neural Network* mendapatkan nilai solusi minimum. Algoritma pembelajaran untuk merevisi nilai bobot dapat dituliskan seperti pada Persamaan (3.23) hingga persamaan (3.29). Sehingga persamaan untuk mendapatkan nilai bobot baru pada *neural network* untuk mendapatkan nilai K_i dapat dilihat pada persamaan (3.30) dan untuk *neural network* untuk mendapatkan nilai K_p pada persamaan (3.31).

$$\frac{\partial E}{\partial w_o} = \frac{\partial E}{\partial em} \cdot \frac{\partial em}{\partial y} \cdot \frac{\partial y}{\partial u(n)} \cdot \frac{\partial u(n)}{\partial O_k} \cdot \frac{\partial O_k}{\partial z_o} \cdot \frac{\partial z_o}{\partial w_o} \quad (3.23)$$

$$\frac{\partial E}{\partial em} = \frac{\partial \left(\frac{1}{2} em^2 \right)}{\partial em} = em \quad (3.24)$$

$$\frac{\partial em}{\partial y} = \frac{\partial (ym - y)}{\partial y} = -1 \quad (3.25)$$

$$\frac{\partial u(n)}{\partial O_{kp}} = e(n) - e(n-1) \quad (3.26)$$

$$\frac{\partial u(n)}{\partial O_{ki}} = e(n) \quad (3.27)$$

$$\frac{\partial O_{kp/ki}}{\partial z_o} = \frac{\partial f(z_o)}{\partial z_o} = f'(z_o) = \lambda \quad (3.28)$$

$$\frac{\partial z}{\partial w_o} = \frac{\partial (w_o \cdot O_{kp/ki})}{\partial w_o} = O_{kp/ki} \quad (3.29)$$

$$w_o = w_o + \eta \cdot em \cdot \frac{\partial y_o}{\partial u(n)} \cdot e(n) \cdot \lambda \cdot O_h \quad (3.30)$$

$$w_o = w_o + \eta \cdot em \cdot \frac{\partial y_o}{\partial u(n)} \cdot (e(n) - e(n-1)) \cdot \lambda \cdot O_h \quad (3.31)$$

[Halaman ini sengaja dikosongkan]

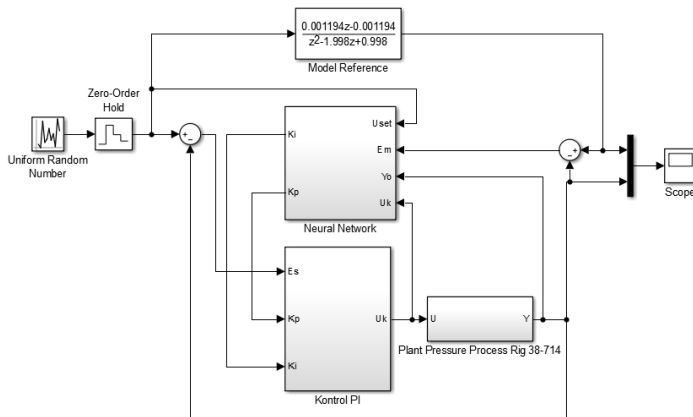
BAB 4

PENGUJIAN DAN ANALISA

Setelah melakukan perancangan kontroler seperti yang terdapat pada bab 3, perlu dilakukan pengujian diantaranya dengan pengujian kontroler terhadap perubahan masukan, dan ketahanan kontroler terhadap perubahan beban.

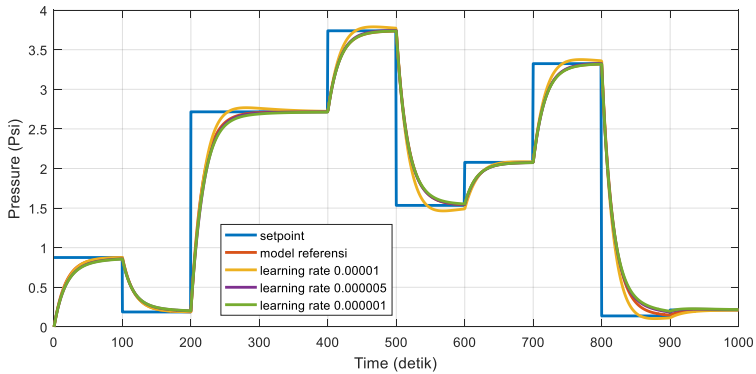
4.1. Simulasi Proses Pembelajaran *Neural Network*

Sebelum melakukan implementasi menggunakan kontroler PI berbasis *Neural Network* dengan skema MRAC dilakukan proses pembelajaran terlebih dahulu pada struktur *Neural Network* yang akan dibuat. Struktur *Neural Network* yang digunakan untuk proses pembelajaran adalah *backpropagation*. Seperti yang dijelaskan pada perancangan *Neural Network* pada bab 3, algoritma *feedforward* digunakan untuk mendapatkan nilai keluaran sedangkan algoritma *backpropagation* digunakan untuk merevisi bobot. Bobot yang telah direvisi dianggap sebagai bobot terbaik yang akan diimplementasikan pada *plant*. Proses pembelajaran *Neural Network* mencari nilai bobot terbaik dilakukan secara simulasi dengan bantuan *software* MATLAB Simulink. Seperti yang terlihat pada Gambar 4.1 yang merupakan diagram blok pada *software* MATLAB Simulink yang digunakan untuk melakukan proses pembelajaran pada *Neural Network*.

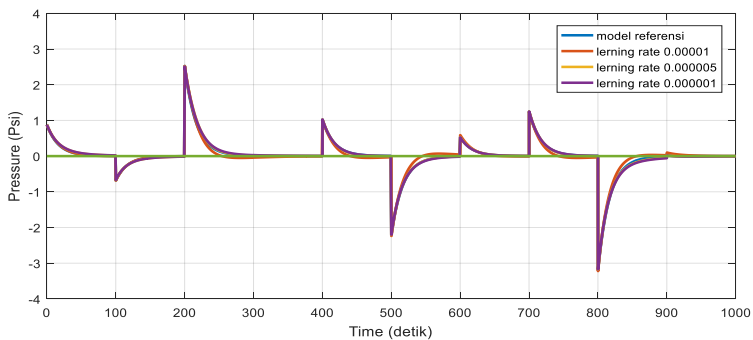


Gambar 4.1 Diagram Blok Proses Pembelajaran *Neural Network*

Proses pembelajaran diawali dengan masukan *random* dengan nilai antara 0 hingga 4. Setelah diberikan nilai secara *random* maka dapat dilihat respon keluaran sistem dapat mengikuti model referensi seperti yang terlihat pada Gambar 4.2. Pada Gambar 4.2 dan Gambar 4.3 dapat dilihat bahwa ketika respon keluaran sudah sesuai dengan model referensi dan bobot sudah berhenti merevisi nilai, maka proses pembelajaran dapat dihentikan sehingga bobot terakhir dianggap sebagai bobot terbaik yang akan diimplementasikan pada *plant*. *Learning rate* yang digunakan pada tugas akhir ini adalah 0.00001, 0.000005, dan 0.000001. Dilakukan beberapa variasi *learning rate* guna mendapatkan bobot yang paling baik, dilihat dari nilai RMSE yang paling kecil.



Gambar 4.2 Simulasi Hasil Pembelajaran *Neural Network* dengan variasi *learning rate*



Gambar 4.3 Grafik error Hasil Pembelajaran *Neural Network* dengan variasi *learning rate*

Dari data hasil simulasi yang didapatkan akan dilakukan perbandingan respon keluaran *plant* dengan respon keluaran model referensi. Dengan metode validasi RMSE dimana semakin kecil nilai RMSE maka respon *plant* akan semakin sesuai dengan model referensi. Tabel 4.1 merupakan hasil validasi dari setiap percobaan *learning rate* yang dilakukan.

Tabel 4.1 Nilai RMSE *Learning Rate*

No.	<i>Learning Rate</i>	RMSE
1	0.00001	0.2966
2	0.000005	0.1837
3	0.000001	0.1512

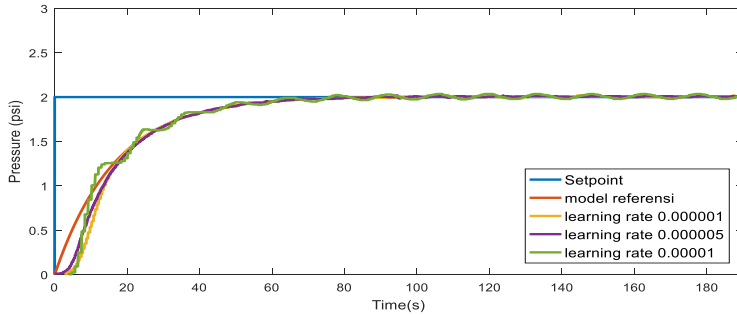
Pada Tabel 4.1 nilai *learning rate* 0.000001 merupakan *learning rate* terbaik dengan nilai RMSE terkecil sehingga dapat dikatakan paling menyerupai dengan model referensi. Semakin besar nilai *learning rate* maka semakin cepat respon keluaran *plant*.

4.2. Pengujian Kontroler PI berbasis *Neural Network* dengan skema MRAC tanpa perubahan beban

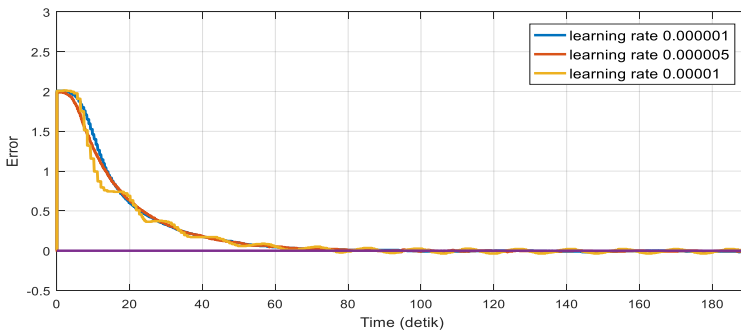
Setelah mendapat nilai bobot terbaik hasil pembelajaran *Neural Network*, maka dilakukan pengujian langsung pada *plant*. Pengujian dilakukan hanya dengan menggunakan algoritma *feedforward* saja tanpa menggunakan algoritma *backpropagation*. Pengujian ini dilakukan untuk mengetahui respon sistem hasil pembelajaran *Neural Network* yang dilakukan apakah dapat mengikuti model referensi atau tidak.

Seperti yang terlihat pada Gambar 4.4 dan Gambar 4.5, respon *plant* mampu mengikuti model referensi dengan nilai RMSE yang kecil pada setiap nilai *learning rate* seperti yang terlihat pada Tabel 4.2 meskipun terdapat *delay* yang dikarenakan kontroler sedang beradaptasi sesuai dengan model referensi. Spesifikasi respon transien *plant* dapat dilihat pada Tabel 4.2. Mengacu pada Tabel 4.2, Respon *plant* hasil implementasi memiliki *error steady state* terkecil dengan nilai 0.5% pada *learning rate* 0.000001 dan 0.000005. Sementara itu untuk respon *plant* hasil implementasi yang memiliki waktu *settling time* paling mendekati waktu yang diinginkan terdapat pada *learning rate* 0.000001 dengan

waktu 50.14 detik sehingga pada pengujian kali ini respon keluaran terbaik didapat pada nilai *learning rate* 0.000005.



Gambar 4.4 Respon sistem Kontroler PI berbasis *Neural Network* dengan skema MRAC tanpa perubahan beban



Gambar 4.5 Error keluaran *plant* terhadap masukan ketika pengujian tanpa perubahan beban

Tabel 4.2 Spesifikasi Respon Kontroler PI berbasis *Neural Network* dengan skema MRAC

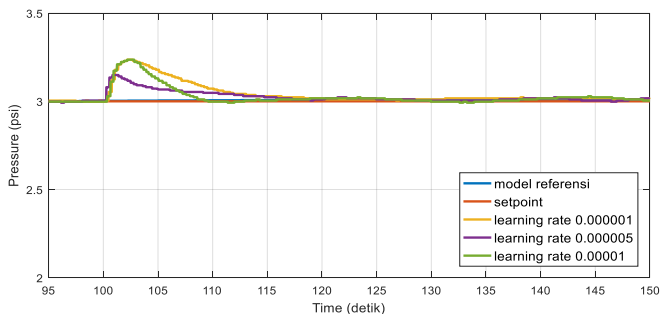
Spesifikasi Respon Transien	<i>Learning Rate</i> 0.000001	<i>Learning Rate</i> 0.000005	<i>Learning Rate</i> 0.00001
<i>Setling Time</i> (5%)	50.14 detik	50.38 detik	47.4 detik
<i>Delay Time</i>	14.08	13.58	11.2
<i>Error Steady State</i>	0.5%	0.5%	1.75%

4.3. Pengujian Kontroler PI Berbasis *Neural Network* dengan skema MRAC Terhadap Perubahan Beban

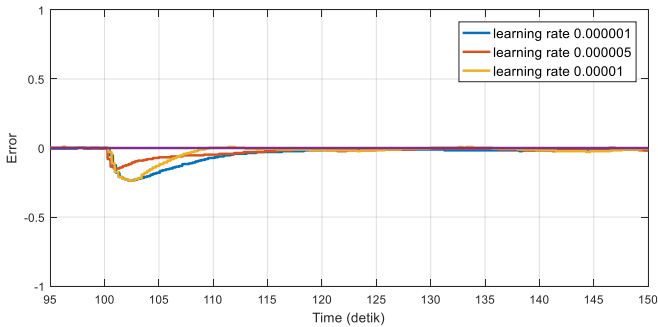
Pengujian selanjutnya yaitu pengujian kontroler PI berbasis *Neural Network* dengan skema MRAC terhadap perubahan beban. Perubahan beban pada tugas akhir ini yaitu dengan mengatur bukaan manual *valve* V5. Tujuan dari pengujian dengan perubahan beban ini adalah untuk melihat bagaimana respon sistem dengan menggunakan kontroler PI berbasis *Neural Network* dengan skema MRAC dapat mempertahankan nilai sesuai dengan nilai referensi ketika terjadi perubahan beban pada keluaran sistem. Semakin cepat respon mencapai nilai *steady state* kembali maka kontroler dapat dikatakan sebagai kontroler yang baik. Pada pengujian ini, dilakukan pengaturan awal *plant* dengan konfigurasi sebagai berikut:

- Jumlah data yang diambil : 10000 data
- Katup terbuka : V1, V3, V4, dan V5
- Katup tertutup : V2 dan V6
- *Time Sampling* : 0.02 detik

Pengujian dilakukan ketika respon sistem sudah mencapai nilai *steady state*, perubahan beban dilakukan dengan menutup manual *valve* V5 terlebih dahulu. Setelah respon mencapai nilai *steady state* kembali, manual *valve* V5 kembali dibuka. Hasil implementasi respon kontroler PI berbasis *Neural Network* dengan skema MRAC terhadap perubahan beban pada *Pressure Process Rig* 38-714 dapat dilihat pada Gambar 4.6 dan Gambar 4.7.

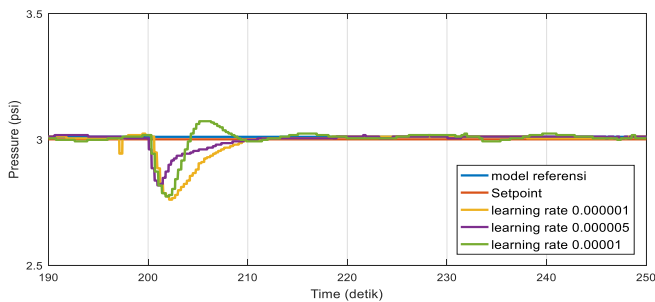


Gambar 4.6 Respon Keluaran *Plant* terhadap perubahan beban ketika katup V5 ditutup

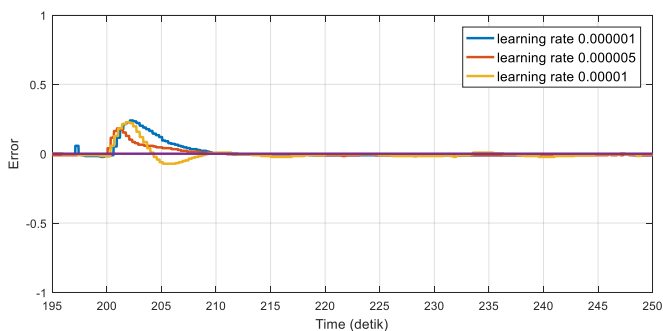


Gambar 4.7 Error keluaran *plant* terhadap masukan ketika perubahan beban katup V5 ditutup

Pada Gambar 4.6 dan Gambar 4.7, dapat dilihat bahwa ketika manual *valve* V5 ditutup pada detik ke 100 *plant* mengalami kenaikan *pressure*. Meskipun mengalami kenaikan *pressure* pada saat manual *valve* V5 ditutup, kontroler PI berbasis *Neural Network* dengan skema MRAC mampu mengembalikan nilai sesuai dengan *setpoint*. Gambar 4.7 merupakan grafik nilai eror ketika katup V5 ditutup. Pada Gambar 4.8 dan Gambar 4.9 saat manual *valve* V5 dibuka kembali pada detik ke 200, respon *plant* mengalami penurunan tekanan, namun kontroler dapat menaikkan nilai *pressure* sesuai dengan *setpoint*. Hal ini menunjukkan bahwa kontroler PI berbasis *Neural Network* dengan skema MRAC dapat dengan baik mengatasi perubahan beban dengan waktu yang cepat untuk mencapai nilai *steady state* nya kembali. Waktu tercepat didapat dengan nilai *learning rate* 0.00001 yang dapat dilihat pada Tabel 4.3.



Gambar 4.8 Respon Keluaran *Plant* terhadap perubahan beban ketika katup V5 dibuka



Gambar 4.9 Error keluaran *plant* terhadap masukan ketika perubahan beban katup V5 ditutup

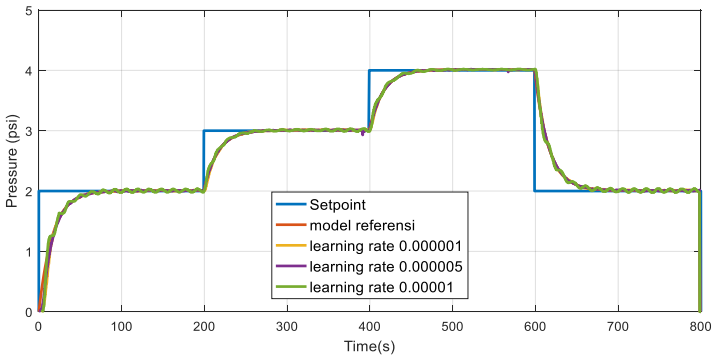
Tabel 4.3 Respon *Plant* Terhadap Perubahan Beban

<i>Recovery Time</i>	<i>Learning Rate</i> 0.000001	<i>Learning Rate</i> 0.000005	<i>Learning Rate</i> 0.00001
Manual Valve V5 Ditutup	27 detik	19 detik	9 detik
Manual Valve V5 Dibuka	11.5 detik	11.3 detik	9.3 detik
<i>Error Steady State</i>	0.56%	0.76%	0.76%

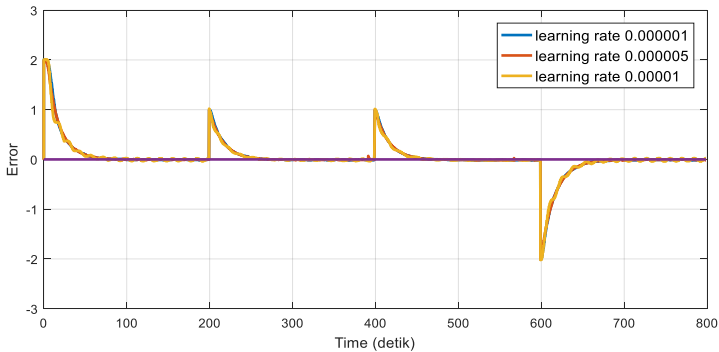
4.4. Pengujian Kontroler PI Berbasis *Neural Network* dengan skema MRAC Terhadap Perubahan Masukan

Pengujian selanjutnya yaitu pengujian kontroler PI berbasis *Neural Network* dengan skema MRAC terhadap perubahan masukan. Perubahan masukan dalam tugas akhir ini yaitu memberikan nilai *setpoint* yang berbeda setiap 200 detik, dimana respon sistem diharapkan dapat mengikuti perubahan *setpoint* yang diberikan sesuai dengan model referensi. Pengujian diberikan dengan nilai *setpoint* awal sebesar 2 psi. Setelah 200 detik, nilai *setpoint* naik menjadi 3 psi. Lalu setelah 200 detik berikutnya naik menjadi 4 psi. lalu 200 detik kemudian nilai *setpoint* turun menjadi 2 psi. Gambar 4.10 dan Gambar 4.11 merupakan respon sistem dengan kontroler PI berbasis *Neural Network* dengan skema MRAC terhadap perubahan *setpoint*. Dapat dilihat pada Gambar 4.10, respon sistem dengan menggunakan kontroler PI berbasis *Neural Network* dengan skema MRAC dapat menyesuaikan dengan masukan

yang berubah dengan karakteristik respon transien yang hampir mendekati seperti respon yang diinginkan yang dapat terlihat pada Tabel 4.5 hingga pada tabel 4.7. Terdapat *delay* pada respon *plant* ketika menuju masukan 2 psi dikarenakan kontroler masih beradaptasi, ketika masukan berubah pada nilai 3 psi, 4 psi dan menuju 2 psi kembali, respon keluaran *plant* sudah bisa mengikuti model referensi yang diinginkan. Respon *plant* juga dianggap sudah dapat mengikuti model referensi dengan nilai RMSE yang kecil ditunjukkan pada Tabel 4.4. Respon terbaik dengan nilai RMSE dan karakteristik yang hampir mendekati dengan respon yang diinginkan didapat dengan *learning rate* 0.000005.



Gambar 4.10 Respon Kontroler PI berbasis *Neural Network* dengan skema MRAC terhadap Perubahan Masukan



Gambar 4.11 Error keluaran terhadap masukan ketika perubahan masukan

Tabel 4.4 Nilai RMSE pada Pengujian Perubahan Masukan

No.	<i>Learning Rate</i>	RMSE
1	0.000001	0.0502
2	0.000005	0.0343
3	0.00001	0.0475

Tabel 4.5 Karakteristik Respon Transien Keluaran *Plant* Terhadap Perubahan Masukan Dengan Nilai *Learning Rate* 0.000001

No.	<i>Setpoint</i>	<i>Setling Time</i>	<i>Delay Time</i>	<i>Error Steady State</i>
1	2 psi	50.14 detik	14.08 detik	0.5%
2	3 psi	47.5 detik	10.9 detik	0.36%
3	4 psi	46.1 detik	10.7 detik	0.425%
4	2 psi	50 detik	11.1 detik	0.9%

Tabel 4.6 Karakteristik Respon Transien Keluaran *Plant* Terhadap Perubahan Masukan Dengan Nilai *Learning Rate* 0.000005

No.	<i>Setpoint</i>	<i>Setling Time</i>	<i>Delay Time</i>	<i>Error Steady State</i>
1	2 psi	50.38 detik	13.58 detik	0.5%
2	3 psi	47.5 detik	10.5 detik	0.36%
3	4 psi	47.5 detik	10.9 detik	0.425%
4	2 psi	50 detik	11 detik	0.5%

Tabel 4.7 Karakteristik Respon Transien Keluaran *Plant* Terhadap Perubahan Masukan Dengan Nilai *Learning Rate* 0.00001

No.	<i>Setpoint</i>	<i>Setling Time</i>	<i>Delay Time</i>	<i>Error Steady State</i>
1	2 psi	47.4 detik	11.2 detik	1.75%
2	3 psi	48.9 detik	11 detik	0.76%
3	4 psi	40.1 detik	9.5 detik	0.425%
4	2 psi	50.2 detik	9.1 detik	2.1%

[Halaman ini sengaja dikosongkan]

BAB 5

PENUTUP

4.5. Kesimpulan

Berdasarkan hasil pengujian dan analisa dapat dikatakan bahwa pada pengujian perubahan masukan sistem pengaturan tekanan *plant Pressure Process Rig 38-714* didapat kontroler PI berbasis *Neural Network* dengan skema MRAC menggunakan *learning rate* 0.000005 memberikan nilai RMSE yang relatif kecil yaitu sekitar 0.0343 sehingga dapat dikatakan bahwa kontroler dapat dengan baik mengatasi permasalahan perubahan masukan. Kontroler PI berbasis *Neural Network* dengan skema MRAC dapat mendekati spesifikasi respon yang diinginkan terhadap model referensi dengan nilai *settling time* 50.14 detik pada nilai *learning rate* 0.000001.

Pada pengujian perubahan beban, sistem pengaturan tekanan *plant Pressure Process Rig 38-714* didapat kontroler PI berbasis *Neural Network* dengan skema MRAC memberikan hasil yang baik pada nilai *learning rate* 0.00001 dengan nilai *recovery time* yang cepat dengan waktu 9 detik ketika manual *valve V5* ditutup dan 9.3 detik ketika manual *valve V5* dibuka kembali. Pada sistem pengaturan tekanan pada *Pressure Process Rig 38-714* dengan menggunakan kontroler PI berbasis *Neural Network* dengan skema MRAC didapat respon sistem dengan nilai *error steady state* yang relatif kecil yaitu dibawah 1 %.

4.6. Saran

Saran untuk penelitian selanjutnya adalah perlunya melakukan banyak percobaan pada proses pembelajaran *Neural Network* agar benar-benar mendapatkan nilai bobot terbaik untuk mendapatkan respon sistem yang diinginkan. Diharapkan penelitian ini dapat dijadikan bahan pembelajaran untuk penelitian selanjutnya.

[Halaman ini sengaja dikosongkan]

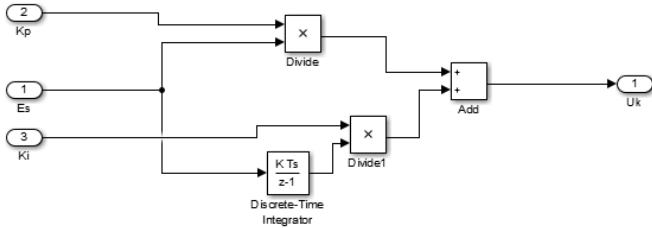
DAFTAR PUSTAKA

- [1] Heriyanto, Pengendalian Proses, Bandung: Politeknik Negeri Bandung, 2010.
- [2] T. Mushiri, A. Mahachi dan C. Mbohwa, "A Model Reference Adaptive Control (MRAC) System for the Pneumatic Valve of the Bottle Washer in Beverages Using Simulink"," *International Conference on Sustainable Materials Processing Manufacturing*, 2017.
- [3] A. Faizal, N. Gemeli, D. Mursyitah, E. Ismaredah dan M. Sadli, "Pengendalian Tekanan Pada Modul Training *Pressure Process Rig* 38-714 Menggunakan Pengendali MRAC Kombinasi PID," *Seminar Teknologi Informasi, Komunikasi dan Industri (SNTIKI-10)*, 2018.
- [4] F. Gunterus, Falsafah Dasar : Sistem Pengendalian Proses, Jakarta: Elex Media Komputindo, 1994.
- [5] N. Gamayanti, Diktat Kuliah Dasar Sistem Pengaturan, Surabaya: Jurusan Teknik Elektro ITS, 2011.
- [6] K. Ogata, Modern Control Engineering, Inc. Englewood Cliffs New Jersey: Prentice Hall, 1970.
- [7] K. J. Astrom dan B. Wittenmark, Adaptive Control Second Edition, New York: Dover Publications, Inc., 2008.
- [8] J. S. Jek, Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan MATLAB, Yogyakarta: ANDI, 2005.

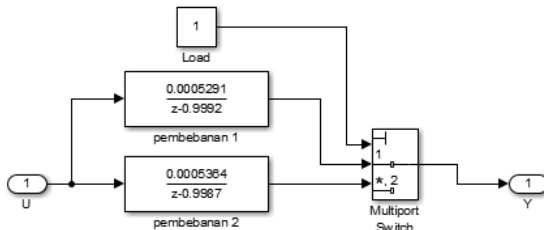
[Halaman ini sengaja dikosongkan]

LAMPIRAN

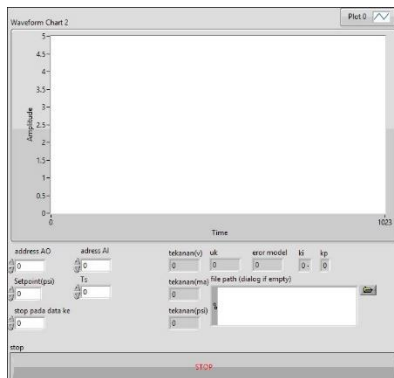
Lampiran 1. *Subsystem* Matlab Simulink Kontroler PI untuk proses pembelajaran.



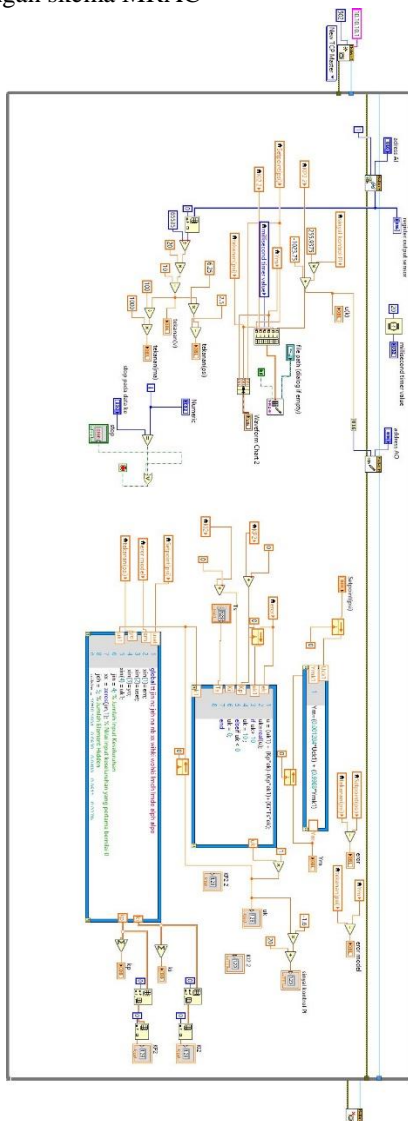
Lampiran 2. *Subsystem* Matlab Simulink *Plant* untuk proses pembelajaran.



Lampiran 3. *Front Panel* Labview kontroler PI berbasis *Neural Network* dengan skema MRAC



Lampiran 4. Program Labview kontroler PI berbasis *Neural Network* dengan skema MRAC



RIWAYAT HIDUP



Elka Gumilar, lahir di Bandung pada tanggal 28 Juli 1995. Putra pertama dari pasangan ayahanda Dedy Kusyadi dan ibunda Ening Komalaningsih. Setelah menempuh pendidikan formal di SDN Cimareme II, SMPN 3 Cimahi dan SMKN 1 Cimahi, penulis melanjutkan melanjutkan studi Diploma 3 jurusan Teknik Otomasi Manufaktur dan Mekatronika di Politeknik Manufaktur Negeri Bandung dan lulus tahun 2016. Kemudian melanjutkan kuliah Lintas Jalur Sarjana di Institut Teknologi Sepuluh Nopember Surabaya dengan mengambil Jurusan Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan.

[Halaman ini sengaja dikosongkan]