



TUGAS AKHIR TF 181801

**PERANCANGAN FAULT TOLERANT CONTROL
SYSTEM (FTCS) AKIBAT KESALAHAN AKTUATOR
PADA SISTEM PENGENDALIAN TEKANAN
SCRUBBER DI PT. PERTADAYA GAS SEMARANG**

**Anak Agung Yuniartha
NRP 0231 1745 000037**

**Dosen Pembimbing
Ir. Ya'umar, M.T.
Moh. Kamalul Wafi, S.T.,MSc.DIC**

**DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2019**



FINAL PROJECT TF 181801

***DESIGN OF FAULT TOLERANT CONTROL
SYSTEM (FTCS) DUE TO ACTUATOR ERRORS
ON THE PRESSURE CONTROL SYSTEM
SCRUBBER IN PT. PERTADAYA GAS SEMARANG***

*Anak Agung Yuniartha
NRP 0231 1745 000037*

*Supervisor
Ir. Ya'umar, M.T.
Moh. Kamalul Wafi, S.T.,MSc.DIC*

***ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019***

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini

Nama : Anak Agung Yuniartha
NRP : 02311745000037
Departemen/ Prodi : Teknik Fisika/ S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Perancangan Fault Tolerant Control System (FTCS) Akibat Kesalahan Aktuator Pada Sistem Pengendalian Tekanan Scrubber di PT. Pertadaya Gas Semarang” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 7 Juni 2019
Yang membuat pernyataan,

Anak Agung Yuniartha
NRP. 02311745000037

LEMBAR PENGESAHAN

**PERANCANGAN FAULT TOLERANT CONTROL
SYSTEM (FTCS) AKIBAT KESALAHAN AKTUATOR
PADA SISTEM PENGENDALIAN TEKANAN
SCRUBBER DI PT. PERTADAYA GAS SEMARANG**

Oleh:

Anak Agung Yuniartha
NRP. 0231 1745 000037

Surabaya, Juni 2019
Mengetahui/Menyetujui

Pembimbing I,

Pembimbing II,

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

Moh. Kamalul Wafi, S.T., MSc.DIC

Ketua Departemen
Teknik Fisika FTI – ITS

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

“Halaman ini memang dikosongkan”

LEMBAR PENGESAHAN

PERANCANGAN FAULT TOLERANT CONTROL SYSTEM (FTCS) AKIBAT KESALAHAN AKTUATOR PADA SISTEM PENGENDALIAN TEKANAN SCRUBBER DI PT. PERTADAYA GAS SEMARANG

TUGAS AKHIR

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

Oleh :

Anak Agung Yuniartha
NRP. 0231 1745 00037

Disetujui oleh Tim Penguji Tugas Akhir:

- | | | |
|---|-------|---------------|
| 1. Ir. Ya'umar, M.T | | Pembimbing I |
| 2. Moh. Kamalul Wafi, S.T., MSc. DIC | | Pembimbing II |
| 3. Dr. Bambang L. Widjiantoro, S.T, M.T | | Ketua Penguji |
| 4. Dr. Katherin Indriawati, S.T, M.T | | Penguji I |
| 5. Erna Septyaningrum, S.T, M.T | | Penguji II |

SURABAYA
2019

“Halaman ini memang dikosongkan”

**PERANCANGAN FAULT TOLERANT CONTROL
SYSTEM (FTCS) AKIBAT KESALAHAN AKTUATOR
PADA SISTEM PENGENDALIAN TEKANAN SCRUBBER
DI PT. PERTADAYA GAS SEMARANG**

Nama Mahasiswa : Anak Agung Yuniartha
NRP : 0231174500037
Departemen : Teknik Fisika
Dosen Pembimbing : Ir. Ya'umar, MT
Moh. Kamalul Wafi, S.T.,MSc.DIC

Abstrak

Pengendalian tekanan gas pada *scrubber* sangatlah penting yang mana akan berpengaruh pada laju gas yang keluar dari plan *scrubber*. Performansi sistem pengendali tidak lepas dari kinerja instrumennya, dimana seringkali aktuator mengalami kesalahan operasi. Pada Tugas Akhir ini dilakukan perancangan *fault tolerant control system (FTCS)* yang mampu bekerja ketika terdapat kesalahan pada aktuator dan dapat mempertahankan performansi sistem. Pertama, dilakukan pemodelan sistem terdiri dari pemodelan plant, aktuator, dan sensor. Selanjutnya merancang kontroler konvensional dan observer untuk aktuator. Kemudian merekonfigurasi sinyal pengendali. Dilakukan tiga jenis pengujian kesalahan untuk melihat performansi rancangan FTC, yaitu uji *loss of effectiveness*, uji kebocoran, dan kesalahan keduanya secara bersamaan dimana FTC memiliki performansi jauh lebih baik dalam mengatasi kesalahan dibandingkan dengan pengendalian konvensional. Terbukti pada salah satu uji kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* 6,351% lebih sedikit, *settling time* 28 detik lebih cepat dan *Error Steady State* 0,0287% yang lebih sedikit.

Kata kunci : *Fault tolerant control, scrubber*, aktuator, observer

“Halaman ini sengaja dikosongkan”

**DESIGN OF FAULT TOLERANT CONTROL SYSTEM
(FTCS) DUE TO ACTUATOR ERRORS ON THE PRESSURE
CONTROL SYSTEM SCRUBBER IN PT. PERTADAYA GAS
SEMARANG**

Name : *Anak Agung Yuniartha*
Student Number : *0231174500037*
Department : *Engineering Physics*
Supervisor : *Ir. Ya'umar, MT*
Moh. Kamalul Wafi, S.T.,MSc.DIC

Abstract

Control of gas pressure in the scrubber is very important which will affect the rate of gas coming out of the plan scrubber. Control system performance can not be separated from the performance of the instrument, where often the actuator has an operating error. In this Final Project, a fault tolerant control system (FTCS) is designed that can work when there is an error in the actuator and can maintain system performance. First, system modeling consists of modeling the plant, actuator, and sensor. Then design the conventional controller and observer for the actuator. Then reconfigure the control signal. Three types of error testing were performed to see the performance of the FTC design, namely loss of effectiveness test, leakage test, and both errors simultaneously where FTC has far better performance in overcoming errors compared to conventional controls. Evidenced in one of the 75% leak error tests along with a 50% loss of effectiveness with the FTC algorithm producing a maximum overshoot of 6.351% less, settling time 28 seconds faster and Error Steady State 0.0287% less.

Key Words: *fault tolerant control, scrubber, actuator, observer*

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Alhamdulillahirabbil'aalamiin, rasa syukur atas segala limpahan rahmat, kesehatan, keselamatan, dan ilmu yang Allah SWT berikan kepada penulis hingga mampu menyelesaikan laporan tugas akhir dengan judul : “PERANCANGAN FAULT TOLERANT CONTROL SYSTEM (FTCS) AKIBAT KESALAHAN AKTUATOR PADA SISTEM PENGENDALIAN TEKANAN SCRUBBER DI PT. PERTADAYA GAS SEMARANG”.

Dalam penyusunan laporan tugas akhir ini, tidak terlepas dari semua pihak yang turut membantu baik moril maupun materiil. Untuk itu penulis mengucapkan terima kasih kepada:

1. Bapak Agus Muhamad Hatta, S.T., M.Si., Ph.D. selaku ketua departemen teknik fisika beserta dosen wali penulis yang telah memberikan petunjuk, ilmu, serta bimbingan selama menempuh pendidikan di teknik fisika.
2. Bapak Ir. Yaumar M.T selaku dosen pembimbing yang dengan sabar membimbing, memberikan saran dan kritiknya selama proses pengerjaan tugas akhir.
3. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku kepala Laboratorium Rekayasa Instrumensi yang telah memberikan ilmu, petunjuk, nasihat, serta kemudahan perizinan yang.
4. Kedua orang tua (Bapak Agung Pribadi dan Ibu Yuniati). Terimakasih atas segala cinta, kasih sayang, doa, perhatian, serta dukungan moril dan materiil yang telah diberikan.
5. Teman-teman asisten laboratorium fisika rekayasa yang senantiasa memberikan dukungan dan fasilitas laboraorium.
6. Seluruh teman – teman Departemen Teknik Fisika Lintas Jalur angkatan 2017 terima kasih untuk semuanya
7. Seluruh dosen, karyawan dan civitas akademik Teknik Fisika, terimakasih atas segala bantuan dan kerjasamanya.
8. Semua pihak yang tidak dapat disebutkan satu persatu, terimakasih atas bantuannya.

Penulis sadar bahwa penulisan laporan tugas akhir ini tidak sempurna, namun semoga laporan ini dapat memberikan kontribusi

yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika khususnya, dan civitas akademik ITS pada umumnya. Semoga laporan tugas akhir ini dapat bermanfaat sebagai referensi pengerjaan laporan tugas akhir bagi mahasiswa yang lain.

Surabaya, 13 Mei 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
BAB I. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Laporan.....	4
BAB II. DASAR TEORI	
2.1 Proses <i>Scrubber</i>	5
2.2 Sistem Pengendalian Tekanan pada <i>Scrubber</i>	7
2.3 Pengendali Proportional-Integral (PIC-0105).....	11
2.4 Control Valve (PCV-0105).....	12
2.5 <i>Fault Tolerant Control</i>	12
2.6 <i>Observer</i>	19
2.7 Penentuan Gain Matriks Observer	22
2.8 Skema Rekonfigurasi Kontrol.....	23
BAB III. METODOLOGI PENELITIAN	
3.1 Prosedur Umum Penelitian	25
3.2 Pengambilan Data	28
3.3 Pemodelan.....	29
3.4 Validasi Model.....	36
3.5 Perancangan Kontroler PI.....	39
3.6 Perancangan <i>Fault Tolerant Control (FTC)</i>	40
3.7 Uji Performansi.....	46
BAB IV. ANALISA DATA DAN PEMBAHASAN	
4.1 Uji Kesalahan Kebocoran	47
4.2 Uji Kesalahan <i>Loss of effectiveness</i>	52

4.3 Uji Kesalahan kebocoran dan <i>Loss of effectiveness</i> ..	58
BAB V. KESIMPULAN DAN SARAN	
5.1 Kesimpulan	71
5.2 Saran	71
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	<i>Scrubber</i>	6
Gambar 2.2	Diagram Blok Sistem Pengendalian Tekanan pada <i>Scrubber</i>	8
Gambar 2.3	Bentuk Umum <i>Control Valve</i>	13
Gambar 2.4	Skema Sistem FTC	15
Gambar 2.5	Struktur Umum FTCS.....	16
Gambar 2.6	Klasifikasi Metode FDI.....	17
Gambar 2.7	Rekonfigurasi Pengendali pada AFTC	18
Gambar 2.8	Skema Rekonfigurasi Kontrol.....	23
Gambar 3.1	Diagram Alir Metodologi Penelitian.....	27
Gambar 3.2	Pengendalian <i>pressure</i> pada <i>scrubber</i>	29
Gambar 3.3	Diagram Blok <i>Open Loop</i>	38
Gambar 3.4	Grafik Respon Validasi <i>Open Loop</i>	38
Gambar 3.5	Diagram Blok Perancangan Pengendali PI	39
Gambar 3.6	Grafik Respon Pengendali PI.....	39
Gambar 3.7	Skema Rekonfigurasi Kontrol Aktuator	46
Gambar 4.1	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 15%	48
Gambar 4.2	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 50%	49
Gambar 4.3	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 75%	50
Gambar 4.4	Respon Sistem Pengendalian dengan Kesalahan <i>Loss of effectiveness</i> 15%	53
Gambar 4.5	Respon Sistem Pengendalian dengan Kesalahan <i>Loss of effectiveness</i> 50%	54
Gambar 4.6	Respon Sistem Pengendalian dengan Kesalahan <i>Loss of effectiveness</i> 85%	55
Gambar 4.7	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 15%	58
Gambar 4.8	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 50%	60

Gambar 4.9	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 75%	61
Gambar 4.10	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 50% dan <i>Loss of effectiveness</i> 15%	62
Gambar 4.11	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 50% dan <i>Loss of effectiveness</i> 50%	63
Gambar 4.12	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 50% dan <i>Loss of effectiveness</i> 85%	65
Gambar 4.13	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 15%	66
Gambar 4.14	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 50%	66
Gambar 4.15	Respon Sistem Pengendalian dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 85%	67

DAFTAR TABEL

Tabel 3.1 Data Pemodelan Plant Scrubber Berdasarkan Data Lapangan	28
Tabel 3.2 Komposisi Fluida yang masuk <i>Scrubber V-100</i>	28
Tabel 3.3 Data Proses Perusahaan PT. Pertadaya Gas Semarang	36
Tabel 3.4 Data Perbandingan Antara <i>Real Plant</i> dengan Simulasi	37
Tabel 4.1 Parameter Performansi dengan Kesalahan Kebocoran 15%	48
Tabel 4.2 Parameter Performansi dengan Kesalahan Kebocoran 50%	49
Tabel 4.3 Parameter Performansi dengan Kesalahan Kebocoran 75%	51
Tabel 4.4 Parameter Performansi dengan Perbandingan uji Kebocoran dengan kesalahan 15%, 50% dan 75% ...	52
Tabel 4.5 Parameter Performansi dengan Kesalahan <i>Loss of effectiveness</i> 15%	53
Tabel 4.6 Parameter Performansi dengan Kesalahan <i>Loss of effectiveness</i> 50%	54
Tabel 4.7 Parameter Performansi dengan Kesalahan <i>Loss of effectiveness</i> 85%	56
Tabel 4.8 Parameter Performansi dengan Perbandingan uji Kebocoran dengan kesalahan 15%, 50% dan 75% ...	59
Tabel 4.9 Parameter Performansi dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 15%	60
Tabel 4.10 Parameter Performansi dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 50%	61
Tabel 4.11 Parameter Performansi dengan Kesalahan Kebocoran 15% dan <i>Loss of effectiveness</i> 85%	62
Tabel 4.12 Parameter Performansi dengan Kesalahan Kebocoran 50% dan <i>Loss of effectiveness</i> 15%	63
Tabel 4.13 Parameter Performansi dengan Kesalahan Kebocoran 50% dan <i>Loss of effectiveness</i> 50%	64
Tabel 4.14 Parameter Performansi dengan Kesalahan	

	Kebocoran 50% dan <i>Loss of effectiveness</i> 85%	65
Tabel 4.15	Parameter Performansi dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 15%	67
Tabel 4.16	Parameter Performansi dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 50%	68
Tabel 4.17	Parameter Performansi dengan Kesalahan Kebocoran 75% dan <i>Loss of effectiveness</i> 15%	69

BAB I

PENDAHULUAN

Pada bab pendahuluan ini dibahas beberapa hal secara umum diantaranya adalah alasan diperlukannya rekonfigurasi sistem pengendali dengan adanya kegagalan pada control valve meliputi latar belakang, perumusan masalah, batasan masalah, dan tujuan penelitian dari tugas akhir yang dilakukan.

1.1 Latar Belakang

Seiring dengan kebijakan dalam hal percepatan ketersediaan gas alam untuk pemenuhan permintaan kebutuhan gas dalam negeri. Pemerintah Indonesia telah menunjuk PT Pertamina (Persero) untuk mengembangkan bisnis gas alam di Indonesia Bagian Timur melalui Transportasi Gas/LNG skala Kecil, Terminal, Regasifikasi dan Distribusi. Dalam proses pengolahan gas terdapat beberapa stage yang harus dilalui. Umumnya gas yang akan diproses diperoleh dari gas outlet pada separator. Gas keluaran separator didinginkan di heat exchanger. Kemudian gas tersebut dilewatkan melalui *scrubber* agar menghilangkan fasa *liquid* untuk selanjutnya dibawa menuju kompresor (Huang dkk, 2013). Salah satu gas *scrubber* di Pertamina Gas yang sangat vital adalah V-100 yang bekerja memisahkan fasa gas dan *liquid* dari campuran gas-kondensat hasil pendinginan gas alam. *Scrubber* merupakan bagian dari unit Gas Sweetening yang berfungsi untuk memisahkan fraksi berat yang terikat ke dalam gas. Di dalam *scrubber* fasa *liquid* yang terbawa dalam aliran gas akan dipisahkan sehingga diperoleh dry gas yang bebas *liquid*. Cairan (*liquid*) yang tertinggal di dalam *scrubber* sebagai kondensat dapat dijadikan nilai tambah produk dimana semakin banyak jumlah gas yang dikeringkan maka semakin banyak kondensat yang didapat.

Gangguan berupa kesalahan komponen yang sering terjadi di bidang industri terdapat pada komponen aktuator dan sensor. Aktuator dan sensor merupakan komponen yang paling mudah rusak karena terletak di lapangan dan langsung berhubungan dengan lingkungan. Kesalahan pada komponen menyebabkan

jalannya produksi dari suatu perusahaan tidak dapat berjalan secara lancar yang berdampak pada kerugian bagi perusahaan. Kesalahan tersebut memiliki dampak yang dapat berakibat fatal karena dapat menyebabkan gagalnya sistem untuk bekerja, sehingga menyebabkan keselamatan pengemudi dan masyarakat sekitar terancam. Kesalahan pada komponen dapat diatasi dengan penggunaan sistem pengendalian yang mampu menoleransi kesalahan secara otomatis, sehingga kegagalan sistem untuk bekerja dapat diatasi.

Sistem kendali yang mampu menoleransi gangguan berupa kesalahan pada komponen secara otomatis disebut Fault Tolerant Control System (FTCS). FTCS merupakan suatu perkembangan teknologi untuk kebutuhan keamanan dan perbaikan unjuk kerja. FTCS dapat diklasifikasikan pada 2 tipe yaitu Passive Fault Tolerance Control System (PFTCS) dan Active Fault Tolerance Control System (AFTCS). AFTCS terdiri dari 2 tahapan, yaitu Fault Detection and Identification (FDI) dan Reconfigurable Control (Indriawati dkk, 2013). FDI digunakan untuk memperkirakan kesalahan sedangkan Reconfigurable Control digunakan agar respon sistem menjadi lebih baik, dengan parameter maximum overshoot yang rendah, settling time yang cepat, error steady state yang mendekati nol. Oleh karena itu, pada tugas akhir ini akan dilakukan perancangan sistem Fault Tolerant Control (FTC) akibat kesalahan aktuator pada sistem pengendalian tekanan *scrubber* di PT. Pertadaya Gas Semarang. Penggunaan FTCS dapat menoleransi kesalahan yang terjadi pada plant *scrubber*, sehingga sistem dapat bekerja secara stabil dan performansi tetap terjaga.

1.2 Perumusan Masalah

Permasalahan yang muncul dalam tugas akhir ini adalah sebagai berikut :

- Bagaimana merancang sistem observer akibat kesalahan Aktuator pada sistem pengendalian tekanan *Scrubber* di PT. Pertadaya Gas Semarang ?

- Bagaimana melakukan kompensasi kesalahan akibat kesalahan Aktuator pada sistem pengendalian tekanan *Scrubber* di PT. Pertadaya Gas Semarang?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian tugas akhir ini adalah sebagai berikut :

- Kesalahan Aktuator yang ditinjau adalah bukaan *pressure control valve* yang menyebabkan reduksi nilai laju aliran gas dari *Scrubber*.
- Aktuator yang digunakan adalah pneumatik aktuator
- Pemodelan plant hanya meninjau satu proses di bagian dasar *Scrubber*.
- Kontroller yang terpasang pada plant *Scrubber* yaitu pengendali PI.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam pengerjaan tugas akhir ini adalah dapat merancang sistem FTC pada sistem pengendalian tekanan *Scrubber* di PT. Pertadaya Gas Semarang dengan kesalahan Aktuator.

1.5 Manfaat

Manfaat tugas akhir ini adalah sebagai berikut :

- Perancangan *Fault Tolerance Control* (FTC) dapat dijadikan sebagai metode baru pada perancangan suatu sistem kendali agar menjadi handal dan memiliki dampak kerusakan yang kecil meskipun terjadi suatu kesalahan pada aktuator.
- Perancangan FTC dapat dijadikan sebagai acuan penggunaan *software* untuk sistem kendali, yaitu sebagai pengendali nilai tekanan di *Scrubber* pada PT. Pertadaya Gas Semarang.

“Halaman ini sengaja dikosongkan”

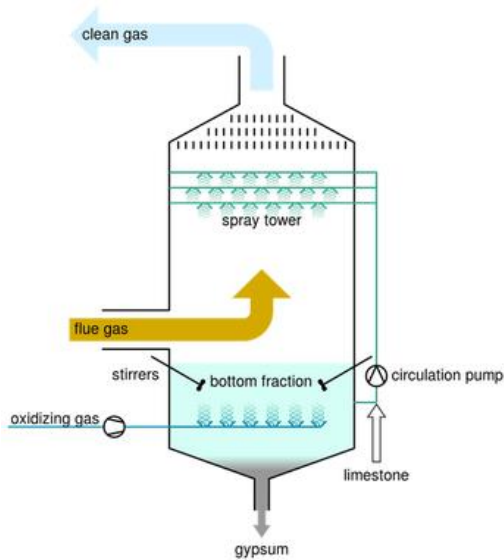
BAB II DASAR TEORI

Bab ini berisi teori dasar yang melandasi alur berpikir dalam menyelesaikan permasalahan dan mencapai tujuan dari penelitian tugas akhir ini. Teori – teori tersebut mencakup hal – hal yang mendukung dan menjadi dasar rujukan dalam tugas akhir. Pengambilan dasar teori berasal dari *text book*, jurnal ilmiah yang dipublikasikan melalui internet, *Instruction Manual Book* yang di akses dari perpustakaan perusahaan tempat penulis mengambil data. Pembuatan model proses dengan mengambil beberapa asumsi/pendekatan yang didasarkan pada berbagai referensi juga disajikan dalam bab ini. Model proses yang telah didapatkan tersebut nantinya digunakan pada bab selanjutnya untuk di program pada *simulink* MATLAB.

2.1 Proses Scrubber

Scrubber atau yang kebocorana dikenal sebagai Two-phase Separator atau Gas-Liquid Separator merupakan peralatan yang bekerja untuk menjernihkan gas dari kondensat yang masih tercampur di dalam gas agar dihasilkan produk sesuai permintaan pembeli. *Scrubber* dapat memiliki bentuk vertikal maupun horizontal yang terpasang di unit-unit pengolahan gas. Khusus untuk vertical *scrubber* digunakan untuk memisahkan *liquid* dari aliran dengan gas-to-*liquid* ratio yang tinggi (NORSOK standart, 2001). Proses pemisahan/separasi dua fasa gas-*liquid* sering dilakukan dalam berbagai macam industri seperti industri nuklir, kimia dan perminyakan dimana proses pemisahannya dapat berdasarkan gaya gravitasi dan sentrifugal (Huang dkk, 2013). Syarat-syarat separasi dapat dilakukan yaitu (Vu dan John, 2012) :

- sebelum memasuki separator harus sudah terbentuk fasa *liquid* dan fasa vapor
- komponen-komponen yang akan dipisahkan harus memiliki volatilitas yang berbeda.



Gambar 2.1 *Scrubber* (Vu dan John, 2012)

Secara umum, *scrubber* adalah suatu bejana separator 2 fasa untuk membersihkan sedikit cairan dari aliran gas. *Scrubber* berfungsi untuk mengambil fasa *liquid* yang terbawa dalam aliran gas agar didapat gas yang bebas *liquid*. Ada banyak metode penjerinahan gas pada *scrubber*, salah satunya adalah dengan menggunakan system *vane pack* dengan cara , menabrakkan inlet gas. *Scrubber* kebocorannya dipasang pada pipa gas setelah separator produksi dan untuk selanjutnya gas dialirkan ke beberapa tempat. *Scrubber* kebocorannya berupa tabung tegak tetapi ada juga yang horizontal, yang digunakan untuk tujuan tertentu. Contoh aplikasi dari *scrubber* adalah gas yang keluar dari gas outlet separator sebelum dialirkan ke flare untuk dibakar atau dialirkan ke alat proses selanjutnya seperti pemakaian *gas engine*, kompor gas ataupun untuk *pensupply* alat alat *instrument*, harus dibebaskan dari kandungan cairan atau dengan kata lain gas

tersebut harus dibersihkan dari polutan dengan dikeringkan terlebih dahulu.

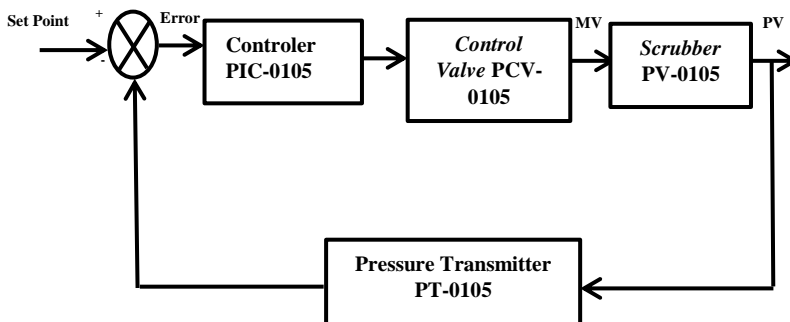
Dalam alur proses produksi gas, *scrubber* PV-0105 merupakan bagian dari unit Gas *Sweetening* yang berfungsi untuk meminimalisir adanya *liquid* atau kondensat yang terikut ke dalam gas. Karena antara gas dan *liquid* mempunyai tekanan yang berbeda, maka proses pemisahan dilakukan dengan cara mengatur tekanan pada *scrubbe*, sehingga antara gas dan *liquid* akan terpisah dengan dibatasi oleh tekanan yang dikehendaki yaitu sebesar 24 barg.

2.2 Sistem pengendalian tekanan pada *Scrubber*

Sistem pengendalian tekanan pada plant *Scrubber* PT Pertamina Gas Semarang yaitu menjaga tekanan sebagai variable proses dengan algoritma kontrol PI. Untuk mengendalikan tekanan gas didalam *scrubber* maka digunakan laju aliran gas dari keluaran *scrubber* bagian atas. Laju aliran oleh bukaan *control valve* akan ditentukan oleh perbandingan informasi yang didapat melalui *pressure transmitter*. Kondisi *pressure* diharuskan stabil dan berada dibatas antara *Low Water Pressure* dan *Normally Water Pressure*. Peningkatan dan penurunan tekanan akan sangat berpengaruh pada produk yang dihasilkan, kualitas produk yang dihasilkan dipengaruhi besarnya tekanan pada plant *Scrubber* dimana produk dihasilkan pada tekanan tertentu apabila tekanan terlalu tinggi maka dapat menyalakan alarm karena dapat mengakibatkan ledakan jika tekanan terlalu rendah maka produk yang dihasilkan akan berkurang.

Dengan adanya penurunan dan peningkatan tekanan pada keluaran *scrubber* maka diberikan suatu pengendalian tekanan pada laju aliran gas yang keluar dimana jika tekanan tinggi maka pengendali tekanan akan mengakomodasi sinyal untuk memperkecil bukaan actuator *control valve* pada aliran yang keluar menuju *station inlet metering* dan berlaku sebaliknya untuk tekanan yang rendah. Tekanan di dalam *scrubber* harus dijaga pada tekanan 24 barg dibagian atas *scrubber* dengan memanipulasi

variable laju dari gas *low pressure*. Sistem pengendalian tekanan pada *scrubber* dapat digambarkan dalam diagram blok berikut :



Gambar 2.2 Diagram blok system pengendalian tekanan pada *Scrubber*

Sistem pengendalian tekanan pada *Scrubber* PV-0105 merupakan system yang berfungsi menstabilkan tekanan didalam *Scrubber* PV-0105 pada nilai yang dikehendaki dengan harapan untuk meminimalisir kandungan *liquid* yang terikut ke dalam gas yang keluar dari *scrubber* PV-0105 serta untuk mencegah terjadinya overpressured di dalam *Scrubber*. Faktor yang mempengaruhi sistem pengendalian tekanan di *Scrubber* PV-0105 adalah besarnya gas yang masuk, semakin banyak gas yang masuk maka akan semakin besar pula tekanan di dalam *Scrubber*.

Hukum kesetimbangan massa :

Massa yang masuk pada *plant* adalah massa fraksi vapor yang merupakan massa campuran antara gas dan *liquid* hasil dari kondensasi pada air cooler. Dengan melihat kondisi proses dimana terjadi pemisahan gas dan *liquid* sebagai kondensat dan dengan melihat kondisi gas sebelum masuk *scrubber* maka dapat ditentukan kesetimbangan massa sebagai berikut

$$\frac{d}{dt} [\rho_g V_g + \rho_q V_1] = q_{in} - q_{o(l)} - q_{o(g)} \quad (2.1)$$

$$\frac{d(\rho_g V_g)}{dt} + \frac{d(\rho_l V_l)}{dt} = q_{in} - q_{out} - q_{o(g)} \quad (2.2)$$

$$V_g \frac{d\rho_g}{dt} + \rho_g \frac{dV_g}{dt} + V_l \frac{d\rho_l}{dt} + \rho_l \frac{dV_l}{dt} = q_{in} - q_{out} - q_{o(g)} \quad (2.3)$$

Keterangan :

ρ_g : Massa jenis Gas dalam *Scrubber* (Kg/m³)

V_g : Volume gas dalam *scrubber* (m³)

ρ_l : Massa jenis *liquid* dalam *Scrubber* (Kg/m³)

V_l : Volume *liquid* dalam *scrubber* (m³)

q_{in} : laju aliran masuk (kg/s)

$q_{o(l)}$: laju aliran *liquid* keluar (kg/s)

$q_{o(g)}$: laju aliran gas keluar (kg/s)

Hukum Kesetimbangan Energi :

Dalam termodinamika perubahan energi terdiri dari tiga komponen yaitu : energi potensial (berkaitan dengan posisi system dalam medan gravitasi bumi), energi kinetik (berkaitan dengan pergerakan system terhadap kerangka koordinat eksternal) dan energi dalam, sehingga perubahan energi total system diberikan dengan persamaan: (Moran,1998)

$$E_2 - E_1 = (EK_2 - EK_1) + (EP_2 - EP_1) + (U_2 - U_1) \quad (2.4)$$

Atau

$$\Delta E = \Delta EK + \Delta EP + \Delta U \quad (2.5)$$

Sehingga persamaan (2.5) dapat dituliskan dalam bentuk,

$$Q - W = \Delta EK + \Delta EP + \Delta U \quad (2.6)$$

Dan kesetimbangan energi yang didasarkan pada laju perubahan waktu adalah :

$$\dot{Q} - \dot{W} = \frac{dEK}{dt} + \frac{dEP}{dt} + \frac{dU}{dt} \quad (2.7)$$

Dengan demikian laju kesetimbangan energi yang menyertai perpindahan massa pada volume control adalah :

$$\frac{dE_{cv}}{dt} = \dot{Q}_{cv} - \dot{W}_{cv} + \sum_i \dot{m}_i \left(h_i + \frac{v_i^2}{w} + gz_i \right) - \sum_e \dot{m}_e \left(h_e + \frac{v_e^2}{2} + gz_e \right) \quad (2.8)$$

2.3 Pressure Transmitter (PT-0105)

Pressure Transmitter (PT-0105) berfungsi untuk mendeteksi besarnya tekanan di dalam *Scrubber* dan mengirimkan sinyal ke *controller* PIC-0105. Di dalam *pressure transmitter* (PT-0105), sensing element yang digunakan adalah diaphragm. Besarnya tekanan yang terdeteksi dapat dilihat pada indicator di *pressure transmitter* yang ada di lapangan. Prinsip kerja *pressure transmitter* (PT-0105) adalah jika sensor tekanan atau *Primary Element* mendeteksi adanya tekanan, maka *transmitter* akan mengirimkan sinyal elektronik sekitar 4-20 mA DC ke *controller* sebagai besaran proses (PV). Data PV ini digunakan untuk indikator di *control room* dan untuk keperluan pengendalian tekanan pada *controller*.

Model matematis *transmitter* PT-0105 dapat didekati dengan menggunakan persamaan,

$$G_T(s) = \frac{K_T}{\tau_T s + 1} \quad (2.9)$$

Dimana K_T : Gain *transmitter* (mA/psig)

τ_T : Time konstan *transmitter* (diasumsikan 0,76 s)

K_T dapat dicari dengan rumus,

$$K_T = \frac{\text{Span Output}}{\text{Span Input}} = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \quad (2.10)$$

2.4 Pengendali Proportional Integral (PIC-0105)

Sistem pengendali adalah kombinasi dari instrumen dan komponen yang dihubungkan oleh sebuah interaksi untuk melakukan kerjasama dengan tujuan tertentu. Sistem pengendalian dapat berupa sistem fisika, biologi, ekonomi, dan sebagainya. Sistem pengendali dapat diaplikasikan pada berbagai bidang, contohnya pada pengendali temperatur, posisi, aliran, tekanan, level, dan lain sebagainya. Pada dasarnya pengendali terbentuk dari empat fungsi dasar, yaitu pengukuran (*measurement*), perbandingan (*comparison*), koreksi (*correction*) dan hasil (*judgement*) (Gunterus,1997).

- **Pengendali Proportional**

Pengendali *proportional* memiliki keluaran yang sebanding dengan besarnya sinyal kesalahan. Secara lebih sederhana dapat dikatakan bahwa keluaran pengendali *proportional* merupakan perkalian antara konstanta *proportional* dengan masukannya. Perubahan pada sinyal masukan menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya. Pengendali *proportional* memiliki dua parameter, yaitu pita *proportional* dan konstanta *proportional*. Daerah kerja pengendali efektif dicerminkan oleh pita *proportional*, sedangkan konstanta *proportional* menunjukkan nilai faktor penguatan terhadap sinyal kesalahan (K_p) (Gunterus,1997).

- **Pengendali Integral**

Pengendali *integral* menghasilkan respon sistem yang memiliki kesalahan keadaan mantap (*steady state*) nol. Jika sebuah *plant* tidak memiliki unsur *integrator*, maka pengendali *proportional* tidak mampu menjamin keluaran sistem memiliki *error steady state* nol. Pengendali *integral* memiliki karakteristik seperti sebuah *integral*. Keluaran pengendali sangat dipengaruhi oleh perubahan sinyal kesalahan. Keluaran pengendali ini merupakan penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti

sebelum terjadinya perubahan masukan. Sinyal keluaran pengendali integral merupakan luas bidang yang dibentuk oleh kurva kesalahan penggerak. Sinyal keluaran akan berharga sama dengan harga sebelumnya ketika sinyal kesalahan berharga nol (Gunterus,1997).

Merancang pengendalian PI dapat dilakukan melalui beberapa cara, diantaranya adalah sebagai berikut.

- Metode respon dengan *quarter – amplitudo decay* (diperkenalkan oleh *Ziegler* dan *Nichols*, pada pertemuan tahunan *The American Society of Chemical Engineers* pada 1 – 5 Desember 1941).
- Metode *trial and error*. Perancang harus mencoba kombinasi pengendali beserta konstantanya untuk mendapatkan hasil terbaik yang paling sederhana.

2.5 Control Valve (PCV-0105)

Final Control Element yang digunakan pada system pengendalian tekanan pada *Scrubber* PV-0105 adalah *control valve* PCV-0105. *Control Valve* disini yang digunakan sebagai aktuator dimana katup atau bukaan dari *control valve* akan bekerja bila mendapat perintah dari *Controller* yang berupa sinyal pneumatik. Pada sistem pengendalian tekanan pada *Scrubber* PV-0105, *control valve* yang digunakan memiliki bentuk *globe valve* dengan karakteristik linear dan merupakan jenis ATC (*Air to Close*) atau FO (*Fail Open*). Prinsip kerja dari *globe valve* sendiri adalah valve yang bekerja dengan prinsip tekanan. *Control valve* bertugas melakukan langkah koreksi terhadap variabel termanipulasi sebagai hasil akhir dari sistem pengendalian. *Control valve* merupakan salah satu elemen pengendali akhir (*final element control*) yang paling umum digunakan. Berdasarkan aksinya *control valve* dibedakan menjadi dua macam, yaitu :

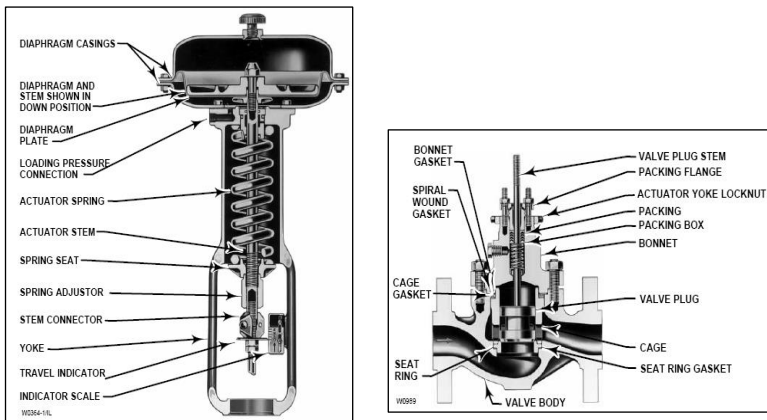
- *Air to Open* (ATO)
ATO adalah kondisi posisi *valve* yang secara normal terbuka (*open*) dan ketika mendapatkan sinyal kendali (4 – 20 mA atau

3 – 15 psi) posisi *valve* menutup secara proporsional dengan sinyal kendali tersebut.

- *Air to Close (ATC)*

ATC adalah kondisi posisi *valve* yang secara normal tertutup (*close*) dan ketika mendapatkan sinyal kendali (4 – 20 mA atau 3 – 15 psi) posisi *valve* membuka secara proporsional dengan sinyal kendali tersebut.

Pemilihan aksi *control valve* dilakukan melalui prosedur HAZOP atau sebelum sistem pengendalian dipasang pada proses operasi. Kesalahan pemilihan aksi *control valve* berakibat fatal terutama ketika berkaitan dengan keselamatan proses maupun *equipment* (Marselina,2016).



Gambar 2.3 Bentuk Umum *Control Valve* (Marselina, 2016)

Ada beberapa jenis tipe *control valve*, seperti *globe valve*, *ball valve*, *butterfly valve*, dan *gate valve*. Gambar 2.3 di atas menggambarkan bentuk umum *control valve* (Marselina,2016).

Model Matematis *Control Valve* dapat didekati dengan :

- Mencari nilai *gain control valve* (G_v)

$$G_v = \frac{\text{Span Output}}{\text{Span Input}} \quad (2.11)$$

- Mencari nilai *gain* I/P ($G_{I/P}$)

$$G_{I/P} = \frac{\text{Span Output}}{\text{Span Input}} \quad (2.12)$$

Sehingga diperoleh *gain* total *control valve* :

$$K_v = G_v \cdot G_{I/P} \quad (2.13)$$

Time constan efektif *control valve* diperoleh berdasarkan hubungan waktu stroke, perfeksional terhadap posisi valve dan perbandingan konstanta waktu *inferent* terhadap waktu *stroke* yang dinyatakan :

$$\tau_v = T_v(\Delta V + R_v) \quad (2.14)$$

Dengan :

τ_v = *time constant control valve* (s)

T_v = waktu stroke penuh (asumsi 0.9 s)

R_v = perbandingan konstanta waktu inderent terhadap waktu stroke ($R_v = 0,03$)

$$\Delta V = \frac{\text{Aliran max} - \text{Aliran min}}{\text{Aliran max}} \quad (2.15)$$

Model matematis *control valve* diperoleh dengan persamaan :

$$G_v = \frac{m_b(s)}{U(s)} = \frac{K_v}{\tau_v s + 1} \quad (2.16)$$

dengan

$m_b(s)$ = laju aliran *input* yang termanipulasi (MMSCFD)

$U(s)$ = sinyal masukan ke *control valve* (mA)

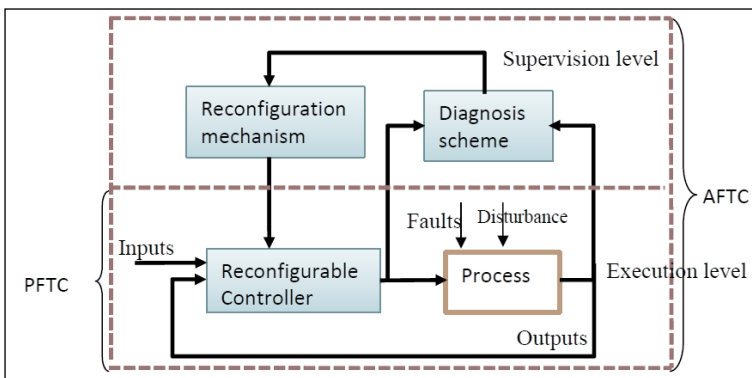
K_v = *gain* total *control valve*

τ_v = *time constant control valve* (s)

2.6 Fault Tolerant Control

Suatu perkembangan mengenai peningkatan performansi dan kebutuhan *safety* sangat dibutuhkan dalam suatu sistem pengendalian. Desain sistem pengendalian umpan balik (*feedback control design*) yang diperuntukkan bagi sistem yang kompleks mungkin dapat menghasilkan kinerja yang kurang stabil, apabila terjadi beberapa kerusakan pada aktuator, sensor, ataupun komponen penyusun sistem tersebut. Telah dilakukan penelitian terkait dengan desain sistem pengendalian yang mampu memberikan toleransi terhadap kerusakan yang terjadi dengan tetap menjaga stabilitas sistem yang diinginkan. *Fault Tolerant Control* adalah sebuah sistem pengendalian modern yang mampu mengakomodasi kegagalan secara otomatis untuk *maintenance* kestabilan dan kinerja saat terjadi kegagalan atau *failure*. Sistem pengendalian ini sering disebut dengan *fault tolerance control system* (FTCS) (Zhang dan Jiang, 2008).

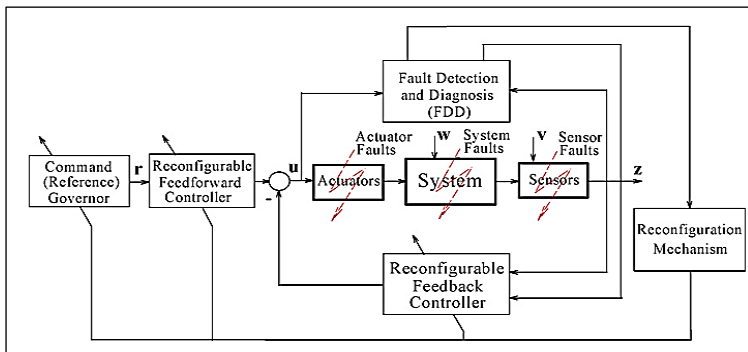
FTCS dapat diklasifikasikan pada dua tipe yaitu *active fault tolerance control system* (AFTCS) dan *passive fault tolerance control system* (PFTCS). AFTCS bereaksi terhadap kesalahan yang terjadi pada suatu komponen dengan merekonfigurasi aksi pengendalian sehingga kestabilan dan kinerja pada sistem dapat dijaga, sedangkan PFTCS didesain untuk menjadi sistem pengendalian yang *robust* dari kesalahan suatu komponen.



Gambar 2.4 Skema Sistem FTC (Noura dkk, 2009)

Gambar 2.4 merupakan gambaran skema FTCS. Tujuan utama dari FTCS adalah merancang kontroler dengan struktur yang cocok untuk mencapai kestabilan dan kinerja yang diinginkan. Tidak hanya ketika pengendalian komponen berfungsi secara normal tetapi ketika terjadi kesalahan pada suatu komponen tersebut (Zhang dan Jiang, 2008).

Perbedaan AFTC dan PFTC terletak pada perancangan *fault detection and identification* (FDI) dan *reconfigurable controller* (RC), sehingga kunci utama pada AFTC adalah merancang kontroler yang dapat direkonfigurasi, skema FDI dengan sensitivitas yang tinggi terhadap kesalahan dan *robust* terhadap ketidakpastian, serta mekanisme rekonfigurasi yang sebisa mungkin mampu memulihkan performansi sistem setelah terjadi kesalahan walaupun terdapat ketidakpastian dan *time delay* pada FDI (Zhang dan Jiang, 2008).



Gambar 2.5 Struktur Umum AFTCS (Zhang dan Jiang, 2008)

Pada gambar 2.5 di atas, Menurut Zhang dan Jiang (2008), Secara umum AFTC terdiri atas empat sub sistem, yaitu :

1. *Reconfigurable controller*, RC;
2. *FDI scheme*;
3. Mekanisme RC; dan
4. *Command reference governor*.

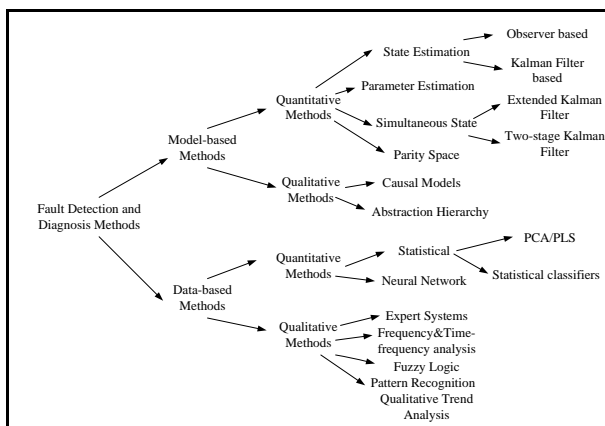
Rekonfigurasi pengendali harus dirancang secara otomatis untuk menjaga stabilitas dan performansi yang diharapkan. Untuk

memastikan bahwa sistem tersebut *closed loop*, maka rekonfigurasi pengendali harus dirancang secara *feed forward*. Hal ini dilakukan untuk menghindari terjadinya penurunan performansi setelah terjadinya kesalahan. Selain itu, ditambahkan dengan *command governor* untuk menyesuaikan aksi pengendali secara *online*.

Pada intinya, perancangan sistem yang penting dalam AFC terdiri atas dua sub sistem yaitu FDI dan mekanisme RC. FDI berfungsi sebagai pendeteksi jenis kesalahan dan besarnya kesalahan tersebut, sedangkan mekanisme RC merupakan perubahan aksi pengendali untuk menghilangkan kesalahan yang terjadi (Zhang dan Jiang, 2008).

- **Klasifikasi FDI**

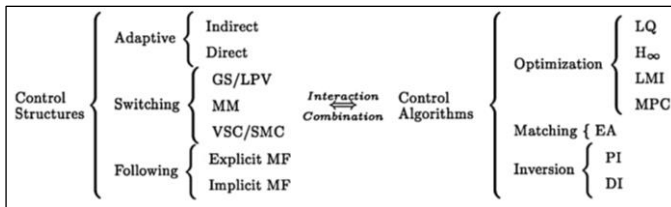
FDI merupakan suatu tahapan untuk mendeteksi *fault* atau kesalahan yang terjadi pada aktuator, sensor, maupun komponen penyusun dalam sistem. Tujuan dari pendeteksian kesalahan ini agar sinyal pengendali yang baru dapat menyampaikan informasi pada kontroler untuk mentoleransi kesalahan yang terjadi (Zhang dan Jiang, 2008). Pada gambar 2.6 di bawah metode FDI diklasifikasikan menjadi dua bagian yaitu *model based methods* dan *data based methods*. Kedua metode tersebut dibagi kedalam pendekatan kuantitatif dan kualitatif.



Gambar 2.6 Klasifikasi Metode FDI (Zhang dan Jiang, 2008)

- **Klasifikasi *Reconfiguration Control***

Merancang sistem kendali yang toleran terhadap kesalahan aktuator dan sensor, maka dilakukan konfigurasi ulang perhitungan sinyal kendali berdasarkan jenis kesalahan yang terjadi. Gambar 2.7 berikut ini merupakan klasifikasi dari RC. Rekonfigurasi sinyal kendali dilakukan untuk mengatasi kesalahan pada aktuator dan sensor yang memerlukan proses diagnosis kesalahan (*fault diagnosis*) untuk mendeteksi dan mengidentifikasi kesalahan yang terjadi, disebut *fault detection and isolation* (Indriawati dkk, 2013).



Gambar 2.7 Rekonfigurasi Pengendali pada AFTC (Indriawati dkk, 2013)

2.7 Observer

Observer adalah suatu algoritma yang bertujuan untuk mengestimasi *state* sistem berdasarkan model sistem, sedangkan *descriptor observer* adalah sistem dalam bentuk *state space* dengan menggunakan parameter sebagai berikut (Noura dkk, 2009).

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.17)$$

$$y(t) = Cx(t) + Du(t) \quad (2.18)$$

Dimana:

- \dot{x} = perubahan state terhadap waktu
- y = output sistem
- A = matriks pada state
- B = matriks pada kontrol input

- C = matriks pada hasil pengukuran
 x = state
 u = kontrol input
 D = measurement output noise

Model matematis *observer* sama seperti model matematis pada sistem sehingga *observer* dapat mengestimasi keadaan sistem dalam bentuk persamaan ruang keadaan. Untuk mengkompensasi kesalahan aktuator dan kesalahan sensor pada waktu yang sama diperlukan modifikasi algoritma *observer* baru. Sehingga jika di dalam sistem atau plant terjadi kesalahan, maka persamaan 2.17 dan 2.18 akan menjadi persamaan berikut (Indriawati, 2015).

$$\dot{x}(t) = Ax(t) + Bu(t) + F_a f_a(t) \quad (2.19)$$

$$y(t) = Cx(t) + Du(t) + F_s f_s(t) \quad (2.20)$$

Dimana:

- F_a = matriks kesalahan aktuator
 F_s = matriks kesalahan sensor
 f_a = vektor kesalahan aktuator
 f_s = vector kesalahan sensor

Penambahan kesalahan $f(t)$ termasuk $f_a(t)$ dan $f_s(t)$ ditunjukkan pada persamaan dibawah.

Observer dibuat dengan kondisi (Indriawati, 2015):

- R_1 Rank $I \geq r + q$
- R_2 Rank $(CF_{ai}) \geq r$
- R_3 Rank $(\tilde{A}_a, \tilde{C}_a)$ adalah *observerable*

Untuk q dan r masing-masing adalah akumulasi pangkat matriks A dan C . \tilde{A}_a dan \tilde{C}_a adalah pelebaran matriks A dan C yang merupakan penurunan dari persamaan (2.19) dan (2.20) dari kedua persamaan tersebut dapat didefinisikan persamaan baru sebagai berikut:

$$\dot{z}(t) = A_z(y - z) \quad (2.21)$$

$$\dot{z}(t) = A_z(Cx(t) - z) \quad (2.22)$$

$$\dot{z}(t) = A_z.Cx(t) - A_z.z(t) \quad (2.23)$$

Untuk A_z merupakan matriks identitas (I) untuk $A_z \in I^q$, sedangkan matriks dari kesalahan aktuator didefinisikan sebagai

$$f = [f_a(t)]$$

Dari persamaan 2.19 dan 2.23 di atas, maka dapat didefinisikan matriks yang akan digunakan untuk mendapatkan matriks yang sudah di ekspansi sebagai berikut.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ A_z.C & -A_z \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} F_a & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_a(t) \\ f_s(t) \end{bmatrix} \quad (2.24)$$

$$\dot{y}(t) = [0 \quad 1] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \quad (2.25)$$

Maka matrik-matrik diatas dapat direpresentasikan dalam bentuk A_a , B_a , C_a , dan E_a sebagai berikut

$$A_a = \begin{bmatrix} A & 0 \\ A_z.C & -A_z \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = [0 \quad 1],$$

$$E_a = \begin{bmatrix} F_a & 0 \\ 0 & A_z.F_s \end{bmatrix}$$

Dengan $x_a = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}$ maka persamaan 2.23 diatas dapat direpresentasikan dalam bentuk persamaan 2.25 berikut

$$\begin{aligned} \dot{X}_a(t) &= A_a x_a(t) + B_a u(t) + E_a f(t) \\ \dot{y}(t) &= C_a x_a(t) \end{aligned} \quad (2.26)$$

Untuk memperoleh nilai dari *gain observer* maka dipakai ekspansi matrik melalui persamaan ruang keadaan baru yang ditunjukkan pada persamaan berikut.

$$\begin{aligned}\hat{X}(t) &= A_a x_a + B_a u + E_a \begin{bmatrix} \hat{f}_a \\ \hat{f}_s \end{bmatrix} + L_X (Y - \hat{Y}) \\ \begin{bmatrix} \hat{f}_a \end{bmatrix} &= L_f (Y - \hat{Y}) \\ \hat{Y} &= C_a X_a\end{aligned}\quad (2.27)$$

Dimana \hat{X} , \hat{f} , \hat{Y} merupakan *state estimate*, *fault estimate* dan *output estimate*. L_X adalah *gain propotional* dari *observer* dan L_f adalah *gain integral* dari *observer*. Sehingga dari persamaan 2.27 diatas akan menjadi persamaan ruang keadaan 2.28 dan 2.29 sebagai berikut.

$$\begin{bmatrix} \hat{X}(t) \\ \hat{f}_a(t) \end{bmatrix} = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{f}_a \end{bmatrix} + \begin{bmatrix} B_a \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_f \end{bmatrix} [C_a \ 0] (Y - \hat{Y}) \quad (2.28)$$

$$\hat{Y}(t) = [C_a \ 0] \begin{bmatrix} \hat{x} \\ \hat{f}_a \end{bmatrix} \quad (2.29)$$

Dari bentuk (2.28) dan (2.29) dapat diubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut.

$$\hat{X}(t) = \tilde{A}_a \tilde{x}(t) + \tilde{B}_a u(t) + \tilde{K}_e (\tilde{y}(t) - \hat{y}(t)) \quad (2.30)$$

$$\hat{Y}(t) = \tilde{C}_a \tilde{x}(t) \quad (2.31)$$

Sehingga matriks yang sudah diekspansi ditunjukkan sebagai berikut.

$$\tilde{A}_a = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix} \quad \tilde{C}_a = [C_a \ 0]$$

Maka *observer* akan ditentukan sebagai:

$$\dot{\tilde{x}}(t) = A \tilde{x} + B u + K_e (Y - C \tilde{x}) \quad (2.32)$$

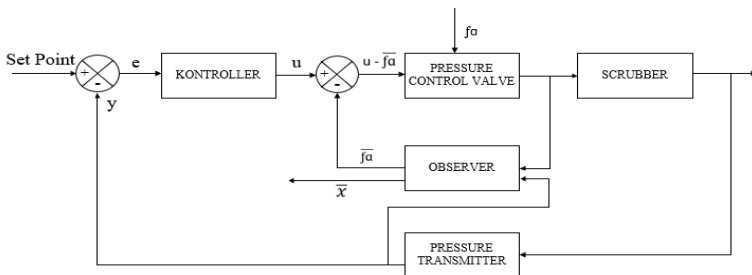
Dimana:

\hat{x} = menunjukkan state estimasi dari $x(t)$

K_e = gain matriks *observer*

2.8 Skema Rekonfigurasi Kontrol

Design reconfiguration dari sinyal kontrol akan bekerja secara otomatis agar dapat mengakomodir kesalahan sehingga *plant* tetap beroperasi. Mekanisme rekonfigurasi dirancang agar mampu memulihkan performansi sistem setelah terjadi kesalahan meskipun terdapat ketidakpastian dan *delay time* pada *fault detection*. Pada Gambar 2.8, menunjukkan bahwa *estimation compensation* dan *fault diagnosis* adalah bagian dari *observer*. *Fault diagnosis* bertujuan untuk mendefinisikan kesalahan dengan mendapatkan informasi dari sinyal hasil pengukuran. Selanjutnya informasi diteruskan menuju *estimation compensation* untuk menestimasi kesalahan dengan mendapat sinyal referensi dari sinyal control. Dengan demikian *estimation compensation* dan *fault diagnosis* akan mendapat input berupa sinyal kontrol (u) dan sinyal hasil pengukuran (y).



Gambar 2.8 Skema Rekonfigurasi Kontrol (Indriawati dkk,2013)

Rekonfigurasi kontrol ini dilakukan dengan menambahkan sinyal kontrol berupa U . Nilai U ini didapatkan dari perhitungan sebagai berikut:

$$U = U_m - f_a \quad (2.33)$$

Dimana:

U = Nilai sebenarnya

U_m = Nilai hasil pengukuran

$Fault$ = Nilai hasil fault

“Halaman ini sengaja dikosongkan”

BAB III

METODOLOGI PENELITIAN

Bab ini berisi metodologi penelitian yang dilakukan dalam menyelesaikan permasalahan dan mencapai tujuan dari penelitian tugas akhir ini.

3.1 Prosedur Umum Penelitian

Sub bab ini menjelaskan mengenai prosedur penelitian tugas akhir yang dilakukan guna mencapai tujuan penelitian. Berikut ini akan disebutkan beberapa tahapan untuk perancangan *Fault Tolerance Control* (FTC).

a. Pengambilan data *plant*

Pengambilan data *plant* meliputi pengambilan data spesifikasi sensor (sensor *pressure transmitter*) dan aktuator (*control valve*), spesifikasi *plant scrubber*, *piping and instrument diagram* dari *plant*, *process flow diagram* dari *plant*, data proses *plant* dari DCS selama rentang waktu tertentu, dan yang terakhir adalah lembar kalibrasi dari sensor *pressure transmitter* dan *control valve* yang dikaitkan dengan data proses dari DCS untuk menunjukkan *trend* saat terjadi kesalahan pada sensor *control valve* serta pada saat kondisi normal.

b. Pemodelan *plant*, sensor, dan aktuator

Pembuatan model *plant scrubber* menggunakan *software* simulink MATLAB R2013a.. Pemodelan dilakukan untuk mengidentifikasi perilaku dari unit operasi *scrubber* tersebut, sehingga fungsi transfer yang diperoleh benar – benar merepresentasikan *real plant*, sedangkan untuk pembuatan model aktuator dilakukan menggunakan persamaan dan data yang ada.

c. Validasi pemodelan

Validasi pemodelan dilakukan untuk memastikan model dapat merepresentasikan sistem *plant* yang sesungguhnya. Validasi pemodelan ada dua macam, yaitu pertama dilakukan dengan membandingkan antara data *real plant* dengan data hasil

simulasi, yang kedua dilakukan menggunakan data proses pada sistem *plant* pada saat kondisi normal dan mantap. Validasi pemodelan ini dilakukan dengan menggunakan fitur *simulink* pada *software* MATLAB R2013a.

d. Perancangan pengendali PI

Perancangan pengendali PI bertujuan untuk menentukan nilai parameter *proportional* dan *integral* yang mampu menghasilkan respon yang baik untuk sistem. Metode yang digunakan untuk mendapatkan nilai parameter – parameter tersebut ialah metode *trial and error*.

e. Perancangan *Fault Tolerant Control* (FTC)

Perancangan FTC dilakukan dengan merancang *observer* bagi aktuator. Perancangan *observer* aktuator yaitu mengubah fungsi transfer *laplace model* menjadi *state space model* sehingga didapatkan matriks A, B, C, D yang digunakan untuk mendapatkan *gain observer* pada *simulink software* MATLAB R2013a. Kemudian, dilakukan rekonfigurasi pengendali, dimana perancangan ini bertujuan untuk mengembalikan stabilitas pengendali yang tidak stabil akibat kesalahan aktuator dengan mengubah perhitungan sinyal pengendalinya.

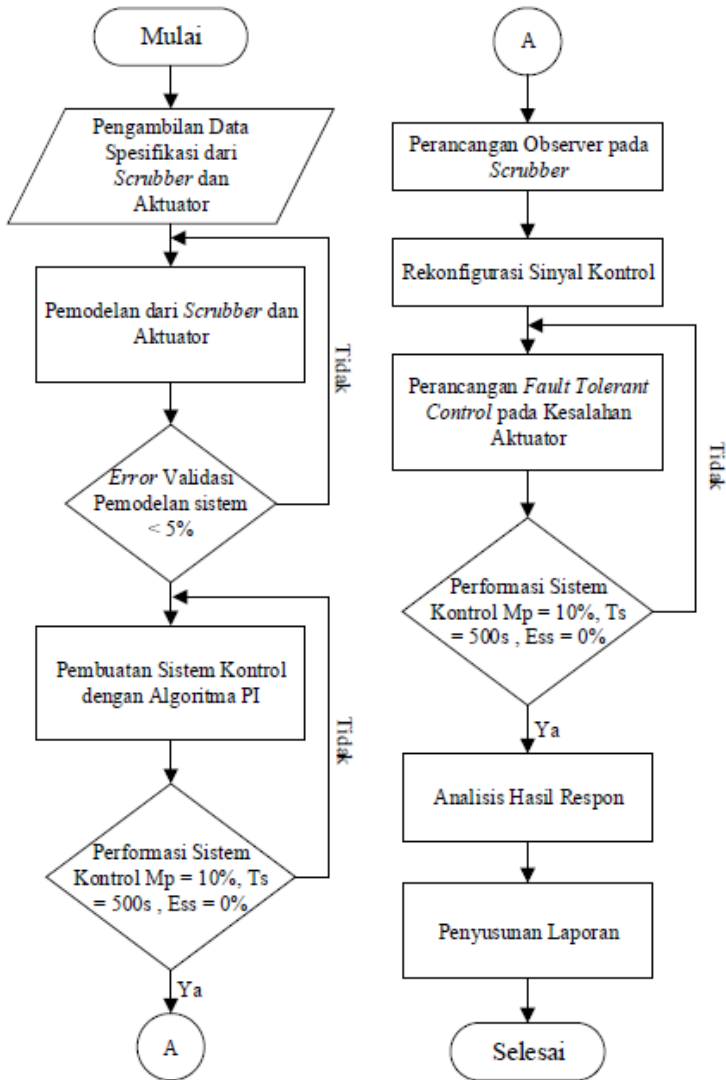
f. Analisis respon

Analisis respon didasarkan pada pemberian uji performansi sistem dengan memberikan kesalahan aktuator berupa kesalahan *loss of effectiveness* dan kebocoran. Setiap pemberian jenis uji tersebut, dilakukan perbandingan antara performansi sistem yang menggunakan pengendali PI dengan algoritma FTC dengan performansi sistem yang hanya menggunakan pengendali PI saja.

g. Penyusunan laporan Tugas Akhir

Penyusunan laporan tugas akhir berupa pembukuan serta dokumentasi dari semua langkah yang telah ditempuh dalam pengerjaan tugas akhir berupa tulisan ilmiah.

Diagram alir dari tugas akhir ditunjukkan pada gambar 3.1 berikut ini.



Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir

3.2 Pengambilan Data *Scrubber*

Data – data yang dibutuhkan untuk pemodelan *plant scrubber* terdapat pada tabel 3.1 di bawah ini. Data tersebut diambil dari PT. Pertadayagas Semarang yang dituliskan pada lembar *process flow diagram* (PFD) *plant* dan spesifikasi *plant* yang tampak pada tabel di bawah ini.

Tabel 3.1. Data Lapangan Pemodelan *Plant Scrubber*

Notasi	Keterangan	Nilai dan Satuan
V_i	Volume <i>Scrubber</i>	2,5 m ³
D	Diameter <i>scrubber</i>	1,07 m
H	Tinggi total <i>scrubber</i>	2,4 m
P_{set}	Tekanan <i>set point</i>	24 bar
γ_L	<i>specific gravity</i> dari cairan	0,726
γ_G	<i>specific gravity</i> dari gas	1,173

Data dari komposisi fluida yang memasuki *Scrubber* dapat dilihat pada grafik berikut :

Tabel 3.2 Komposisi Fluida yang masuk *Scrubber* V-100

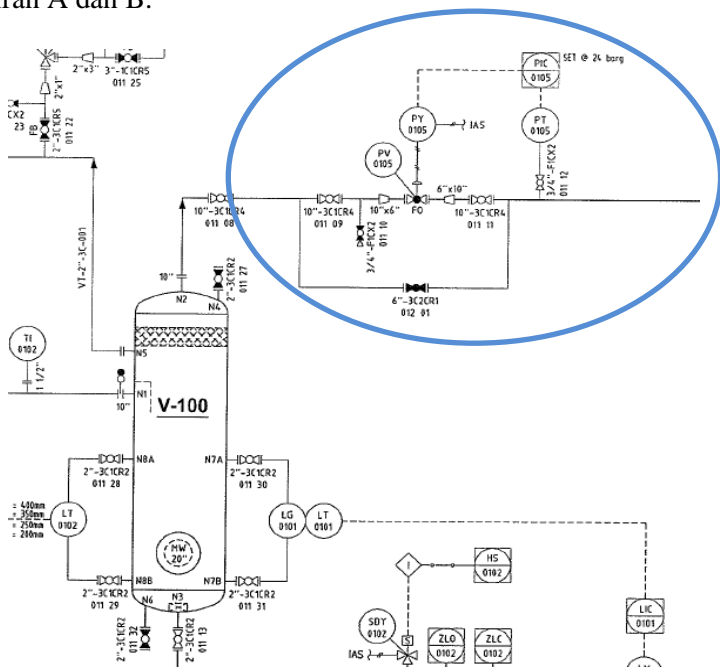
Komposisi gas	Unit	Unit Input
Methane	<i>Comp Mole Frac</i>	0,7331
Ethane	<i>Comp Mole Frac</i>	0,0631
Propane	<i>Comp Mole Frac</i>	0,0396
I – Butane	<i>Comp Mole Frac</i>	0,0087
n – Butane	<i>Comp Mole Frac</i>	0,0068
I – Pentane	<i>Comp Mole Frac</i>	0,0038
n – Pentane	<i>Comp Mole Frac</i>	0,0033
n – Hexane	<i>Comp Mole Frac</i>	0,0042
n – Heptane	<i>Comp Mole Frac</i>	0,0052
n – Octane	<i>Comp Mole Frac</i>	0,0048
n – Nonane	<i>Comp Mole Frac</i>	0,0021
n – Decane	<i>Comp Mole Frac</i>	0,0011

Lanjutan tabel 3.2

n – C11	Comp Mole Frac	0,0004
n – C12	Comp Mole Frac	0,0003
CO ₂	Comp Mole Frac	0,0127
H ₂ O	Comp Mole Frac	0,1105
Nitrogen	Comp Mole Frac	0,0003

3.3 Pemodelan Scrubber

Pemodelan matematis pada tugas akhir ini adalah pemodelan *scrubber* di PT. Pertadayagas Semarang. Cuplikan *Process Flow Diagram* (PFD) dari sistem pengendalian *pressure* pada *scrubber* dapat dilihat pada gambar 3.2 berikut ini, sedangkan untuk gambar lebih detailnya serta gambar proses pada DCS dapat dilihat pada lampiran A dan B.



Gambar 3.2 Pengendalian *pressure* pada *scrubber*

Pemodelan Matematis Plant *Scrubber*

Berdasarkan persamaan pada (2.1) yang disubstitusikan sampai persamaan (2.4) maka pemodelan matematis pada *scrubber* dapat didekati dengan persamaan sebagai berikut.

$$\left[\frac{\text{Energi control Volume}}{\text{Waktu}} \right] = \left[\frac{\text{Energi masuk}}{\text{Waktu}} \right] - \left[\frac{\text{Energi keluar}}{\text{Waktu}} \right] + \left[\frac{\text{Perubahan panas dalam sistem}}{\text{Waktu}} \right]$$

$$\frac{dE}{dt} = E_{in} - E_{out} + Q \quad (3.1)$$

$$\frac{d}{dt} [\rho_g h_g V_g + \rho_l h_l V_l] = q_{in} h_{in} - q_{o(l)} h_{o(l)} - q_{o(g)} h_{o(g)} \quad (3.2)$$

Untuk *liquid* dalam *scrubber*

$$\frac{d}{dt} [\rho_g h_g V_g = \rho_g h_g \dot{m}_{in} - \rho_{o(g)} h_{o(g)} \dot{m}_{o(g)}] \quad (3.3)$$

Untuk vapor atau gas dalam *scrubber*

$$\frac{d}{dt} [\rho_g h_g V_g = \rho_g h_g \dot{m}_{in} - \rho_{o(g)} h_{o(g)} \dot{m}_{o(g)}] \quad (3.4)$$

Berdasarkan kesetimbangan tekanan

$$\frac{dP}{dt} = \left(\frac{\rho h g}{A} \right) (\dot{m}_{in} - \dot{m}_{out}) \quad (3.5)$$

$$\frac{dP}{dt} = \frac{\rho_{in} h_{in} g \dot{m}_{in}}{A} - \frac{\rho_{o(g)} h_{o(g)} \dot{m}_{o(g)}}{A} \quad (3.6)$$

Dengan

$$\dot{m}_{out(g)} = k\sqrt{P} \quad (3.7)$$

Sehingga

$$\frac{dP}{dt} = \frac{\rho_{in} h_{in} g \dot{m}_{in}}{A} - \frac{\rho_{o(g)} h_{o(g)} k \sqrt{P}}{A} \quad (3.8)$$

Linearisasi dengan menggunakan Deret Taylor

$$\begin{aligned} k\sqrt{P} &= k\sqrt{P_o} + \left[\frac{d(k\sqrt{P})}{dP} \right]_{P=P_o} (P - P_o) \\ &\quad + \left[\frac{d^2(k\sqrt{P})}{dP^2} \right]_{P=P_o} \frac{(P-P_o)^2}{2!} + \dots \\ k\sqrt{P} &= k\sqrt{P_o} + \frac{P}{2\sqrt{P_o}} (P - P_o) - \frac{k}{8\sqrt{P_o^3}} (P - P_o)^2 \end{aligned} \quad (3.9)$$

Hilangkan orde dua dan orde tinggi sehingga menjadi :

$$k\sqrt{P} = k\sqrt{P_o} + \frac{k}{2\sqrt{P_o}} (P - P_o) \quad (3.10)$$

Sehingga

$$\frac{dP}{dt} = \frac{\rho_{in} h_{in} g \dot{m}_{in}}{A} - \frac{\rho_{o(g)} h_{o(g)} k \sqrt{P}}{A} \quad (3.11)$$

$$\frac{dP}{dt} + \frac{\rho_{o(g)} h_{o(g)} g}{A} \left[k\sqrt{P_o} + \frac{k}{2\sqrt{P_o}} (P - P_o) \right] = \frac{\rho_{in} h_{in} g \dot{m}_{in}}{A} \quad (3.12)$$

$$\frac{dP}{dt} + \frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_o}} P = \frac{\rho_{in} h_{in} g \dot{m}_{in}}{A} \quad (3.13)$$

Melaplace kan fungsi (3.13) sehingga menjadi :

$$s \cdot P(s) + \frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_o}} \cdot P(s) = \frac{\rho_{in} h_{in} g}{A} \dot{m}_{in}(s) \quad (3.14)$$

$$P(s) \left[s + \frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_o}} \right] = \frac{\rho_{in} h_{in} g}{A} \dot{m}_{in}(s) \quad (3.15)$$

Kemudian persamaan diatas dibagi dengan

$$\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}} \quad (3.16)$$

Sehingga menjadi

$$P(s) \left[\frac{s}{\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}}} + 1 \right] = \frac{\frac{\rho_{in} h_{in} g}{A} \dot{m}_{in}(s)}{\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}}} \quad (3.17)$$

Sehingga

$$\tau = \frac{s}{\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}}} = \frac{A 2\sqrt{P_0}}{\rho_{o(g)} h_{o(g)} g k} \quad (3.18)$$

$$P(s) = [\tau s + 1] = \frac{\frac{\rho_{in} h_{in} g}{A}}{\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}}} \dot{m}_{in}(s) \quad (3.19)$$

$$K = \frac{\frac{\rho_{in} h_{in} g}{A}}{\frac{\rho_{o(g)} h_{o(g)} g}{A} \frac{k}{2\sqrt{P_0}}} \quad (3.20)$$

$$P(s) = \frac{K}{[\tau s + 1]} \dot{m}_{in}(s) \quad (3.21)$$

$$\frac{P(s)}{F(s)} = \frac{K}{[\tau s + 1]} \quad (3.22)$$

Keterangan :

P_{in} : Pressure Input (Psi)

$P_{o(g)}$: Pressure output gas/vapor (Psi)

$P_{o(l)}$: Pressure output *liquid* (Psi)

ρ_g : Massa jenis gas dalam *scrubber* (kg/m³)

h_g : Entalpi gas (J/kg)

V_g : Volume gas dalam *scrubber* (m³)

ρ_l : Massa jenis *liquid* dalam *scrubber* (kg/m³)

V_l : Volume *liquid* dalam *scrubber* (m³)

q_{in} : Laju aliran masuk (kg/s)

h_{in} : Entalpi gas masuk (J/kg)

$q_{o(l)}$: Laju aliran *liquid* keluar (kg/s)

$h_{o(l)}$: Entalpi *liquid* keluar (J/kg)

$q_{o(g)}$: Laju aliran gas keluar (kg/s)

$h_{o(g)}$: Entalpi gas keluar (J/kg)

Dengan memasukkan nilai parameter yang didapat pada tabel 3.1 dan 3.2 untuk τ dan K maka didapat,

$$\tau = \frac{A2\sqrt{P_0}}{\rho_{o(g)}h_{o(g)}gk} \quad (3.23)$$

$$\tau = \frac{0,898 \cdot 2 \cdot \sqrt{348,091}}{4,2 \cdot 4,1 \cdot 9,8 \cdot 1} \quad (3.24)$$

$$\tau = \frac{33,508}{168,756} \quad (3.25)$$

$$\tau = 0,1986 \quad (3.26)$$

Sedangkan untuk nilai K didapatkan,

$$K = \frac{\frac{\rho_{in}h_{in}g}{A}}{\frac{\rho_{o(g)}h_{o(g)}g}{A} \frac{k}{2\sqrt{P_0}}} \quad (3.27)$$

$$K = \frac{\rho_{in}h_{in}g}{A} \times \frac{A}{\rho_{o(g)}h_{o(g)}g} \frac{2\sqrt{P_0}}{k} \quad (3.28)$$

$$K = \frac{\rho_{in}h_{in}2\sqrt{P_0}}{\rho_{o(g)}h_{o(g)}gk} \quad (3.29)$$

$$K = \frac{5,2 \cdot 4,9 \cdot 2 \cdot \sqrt{348,091}}{4,2 \cdot 4,1 \cdot 1} \quad (3.30)$$

$$K = \frac{950,771}{17,22} \quad (3.31)$$

$$K = 55,213 \quad (3.32)$$

Jadi,

$$\frac{P(s)}{F(s)} = \frac{K}{\tau s + 1} \quad (3.33)$$

$$\frac{P(s)}{\dot{m}(s)} = \frac{55,213}{0,199s + 1} \quad (3.34)$$

3.3.2 Pemodelan Matematis Aktuator (PV-0105)

Aktuator yang digunakan dalam sistem ini adalah *Control Valve* yang berfungsi untuk mengatur laju aliran dari *scrubber* agar tekanan *output scrubber* sesuai dengan *setpoint*. Model matematik *control valve* diperoleh dengan persamaan (3.35) berikut :

$$\frac{\dot{m}(s)}{U(s)} = \frac{K}{\tau s + 1} \quad (3.35)$$

Dimana:

$\dot{m}(s)$ = Manipulated variable berupa *flow scrubber* (kg/s)

K = Gain *control valve*

$U(s)$ = sinyal kontrol (mA)

τ = *Time constant*

Untuk menghitung *Gain control valve* dengan menggunakan persamaan (3.38) berikut :

$$K_v = K_{i/p} \times K_{aktuator} \quad (3.36)$$

$$K_{i/p} = \frac{(\text{span Output})}{(\text{Span Input})} \quad (3.37)$$

$$K_{aktuator} = \frac{\dot{m}_b \text{ max} - \dot{m}_b \text{ min}}{\text{Span Output}} \quad (3.38)$$

Adapun dari *data sheet* dari *control valve PV-0105* didapatkan beberapa nilai seperti berikut. Untuk *control valve* dengan tipe *I/P Converter* yang mengubah sinyal input 4 - 20 mA menjadi sinyal *pneumatic* 3 -15 psig. Dalam hal ini, *control valve* memiliki masukan sinyal berupa arus listrik kemudian diubah menjadi tekanan untuk menggerakkan stem *control valve*. Gas yang mengalir pada kolom atas *scrubber* dengan laju aliran maksimum sebesar 16 mmscfd dan laju aliran minimum sebesar 12 mmscfd. Maka perhitungan gain *control valve* adalah sebagai berikut.

$$K_{i/p} = \frac{(15-3)psi}{(20-4)mA} = 0,75 \text{ psi/mA} \quad (3.39)$$

$$K_{aktuator} = \frac{(16-12)mmscfd}{(15-3)psi} = 0,33 \text{ mmscfd/psi} \quad (3.40)$$

$$K_v = 0,75 \frac{psi}{mA} \cdot 0,33 \frac{mmscfd}{psi} = 0,25 \text{ mmscfd/mA} \quad (3.41)$$

Nilai waktu konstan pada *control valve* dapat dilihat pada persamaan (3.42) berikut.

$$\tau = T_v (\Delta V + Rv) \quad (3.42)$$

Dimana ΔV berupa fraksi perubahan *control valve*

$$\Delta V = \frac{\dot{m}_b \text{ max} - \dot{m}_b \text{ min}}{\dot{m}_b \text{ max}} = \frac{(16-12) \text{ mmscfd}}{16 \text{ mmscfd}} = 0,25 \quad (3.43)$$

Sedangkan Rv adalah perbandingan waktu konstan inheren dengan time stroke, bernilai 0.03 untuk aktuator diafragma dan 0.3 untuk aktuator piston, untuk $T_v = \frac{Yc}{Cv}$, Yc merupakan faktor *stroking time control valve* dimana nilainya adalah 0.68 menurut Lee dan Choi 1999 . Sedangkan dari data desain nilai Cv adalah 117, maka nilai waktu konstan aktuator adalah :

$$\tau = \frac{0,68}{117} (0,25 + 0,03) = 0,252 \quad (3.44)$$

Sehingga fungsi transfer *control valve* PV – 0105 adalah seperti berikut.

$$\frac{\dot{m}(S)}{U(S)} = \frac{0,25}{0,252s+1} \quad (3.45)$$

3.3.3 Pemodelan Matematis Sensor (PT-0105)

Untuk plant tekanan, sensor dari plant adalah *Pressure Trasmitter*, yaitu suatu alat yang berfungsi untuk mengukur tekanan pada *scrubber* dengan memanfaatkan perbedaan tekanan yang didapat masing-masing *tube transmitter*. Pemodelan pada sensor menggunakan nilai 1 dikarenakan nilai time konstan dari sensor tidak didapatkan dari perusahaan serta untuk memenuhi nilai keluaran agar sesuai antara nilai pemodelan yang kita lakukan dengan keadaan *real* di lapangan.

3.4 Validasi Model

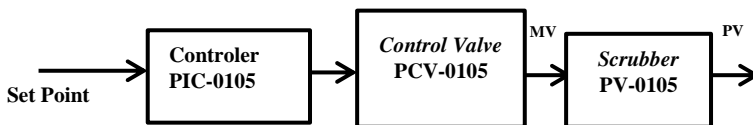
Setelah mendapatkan pemodelan matematis sistem, maka pemodelan tersebut divalidasi menggunakan software MATLAB R2014a untuk mengetahui apakah pemodelan tersebut sudah menggambarkan keadaan plant yang sebenarnya di PT. Pertadaya Gas Semarang. Validasi dilakukan dengan uji open loop fungsi transfer *scrubber* terhadap laju aliran gas yang diberikan. Tabel 3.3 merupakan tabel data proses perusahaan dengan beberapa macam bukaan *valve* yang berbeda. OP menunjukkan bukaan *valve*, sedangkan PV menunjukkan *process variable* yang dalam hal ini adalah tekanan dalam satuan psi.

Untuk membuat validasi pemodelan maka Data real plant yang didapatkan dari perusahaan dibandingkan dengan nilai hasil simulasi. Perbandingan antara *process variable* asli dengan *process variable* hasil simulasi beserta hasil nilai perhitungan *error* antara keduanya yang ditunjukkan pada tabel 3.4 berikut.

Tabel 3.4 Data Perbandingan Antara Real Plant dengan Simulasi

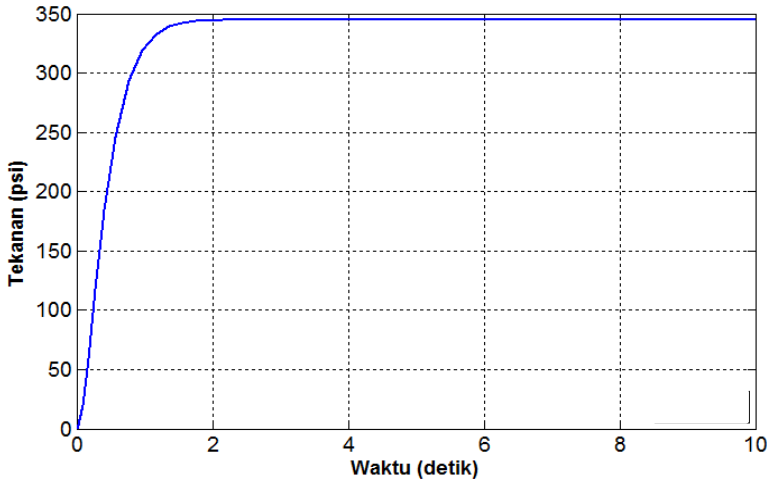
OP (%)	PV real (psi)	Simulasi (psi)	Error (%)
25,33	346,113	349,6	1,01
25,42	347,235	351	1,08
25,5	348,091	352,1	1,15
25,37	346,702	350,2	1,01
25,2	344,805	347,9	0,90
25,35	345,921	349,9	1,15
25,4	347	350,7	1,07
25,28	345,065	349	1,14
25	342,186	345,1	0,85
25,19	344,237	347,7	1,01
25,26	344,989	348,7	1,08
25,54	348,872	352,6	1,07
25,32	345,501	349,5	1,16
25,21	344,52	348	1,01
25,11	343,425	346,6	0,92

Validasi model dilakukan dengan uji open loop fungsi transfer *scrubber* terhadap besarnya bukaan valve yang diberikan menggunakan software Matlab R2014a. Diagram blok validasi model ditunjukkan pada Gambar 3.3 berikut.

**Gambar 3.3** Diagram Blok Open Loop

Grafik respon validasi model ditunjukkan pada Gambar 3.4. Pengujian dilakukan dengan memberikan bukaan valve sebesar 25%. Pada grafik tersebut terlihat bahwa tekanan yang terbaca pada

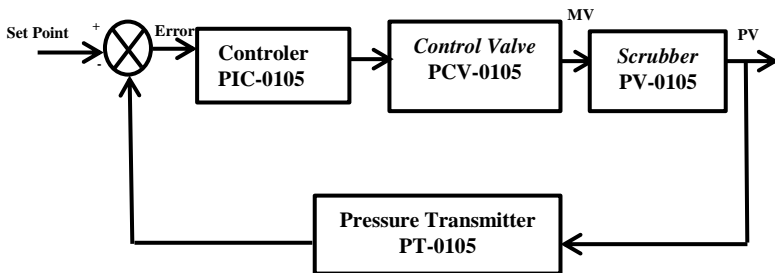
plant adalah 345,1 psi. Nilai tersebut mendekati dengan tekanan *scrubber* pada kondisi real yakni sebesar 342,186 psi dengan error sebesar 0,85 %. Besar nilai error tersebut $\leq 2\%$ sehingga model tersebut dapat dikatakan valid.



Gambar 3.4. Grafik Respon Validasi Model

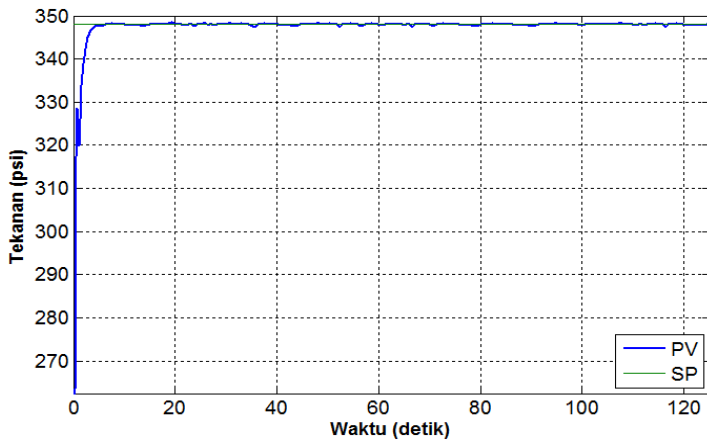
3.5 Perancangan Kontroler PI

Pengendalian tekanan pada *scrubber* menggunakan algoritma kontrol PI. Untuk menentukan *gain* pengendali PI menggunakan metode *trial and error* dimana pertama dilakukan pendekatan dengan *auto tuning* lalu dilakukan *trial and error* agar nilai yang dihasilkan memiliki karakteristik respon yang sesuai. Nilai parameter PI untuk K_p sebesar 0,1396 s dan T_i sebesar 0,32947 s. Setelah itu sistem disimulasikan dengan menggunakan *software* Matlab R2014a dengan skema seperti Gambar 3.5



Gambar 3.5 Skema Diagram Blok Perancangan Pengendali PI

Berdasarkan Gambar 3.5 *setpoint* yang digunakan adalah besaran tekanan yaitu 348 psi. Didapatkan hasil grafik pada gambar 3.6 yaitu *maximum overshoot* sebesar 7,07%, *settling time* sebesar 1,32 detik, dan *error steady state* sebesar 0,0005%.



Gambar 3.6 Grafik Respon Perancangan Kontroler PI

3.6 Perancangan Fault Tolerant Control (FTC)

Perancangan FTC yang dilakukan meliputi dua tahapan, yaitu merancang *observer* dan merekonfigurasi sinyal pengendali.

3.6.1 Perancangan *Observer* untuk Aktuator

Perancangan *observer* digunakan untuk mengestimasi kesalahan pembacaan pada aktuator. Langkah pertama dalam merancang *observer* adalah merubah fungsi transfer sistem ke dalam bentuk matriks ruang keadaan. Berdasarkan persamaan 2.17 dan 2.18 maka fungsi transfer sistem dapat digambarkan menjadi bentuk matriks seperti pada persamaan 3.46 dan 3.47 berikut.

$$\begin{bmatrix} \frac{dP(t)}{dt} \\ \frac{d\dot{m}(t)}{dt} \end{bmatrix} = [A] \begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} + [B][u(t)] \quad (3.46)$$

$$\begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} = [C] \begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} + [D][u(t)] \quad (3.47)$$

Untuk mencari nilai A didapatkan dari fungsi transfer plant dan aktuator seperti pada persamaan (3.34) dan persamaan (3.45) kemudian di *invers laplace* sehingga hasilnya dapat berubah ke domain waktu seperti yang tertera pada persamaan (3.48) dan (3.49) agar bisa didapatkan nilai dari persamaan ruang keadaan.

- Fungsi transfer *plant*

$$\frac{P(s)}{\dot{m}(s)} = \frac{55,213}{0,199s+1}$$

$$P(s)(0,199s + 1) = 55,213 \dot{m}(s)$$

$$0,199 P(s) + P(s) = 55,213 \dot{m}(s)$$

$$\frac{dP(t)}{dt} = \frac{55,213}{0,199} \dot{m}(t) - \frac{1}{0,199} P(t)$$

$$\frac{dP(t)}{dt} = 277,45 \dot{m}(t) - 5,025 P(t) \quad (3.48)$$

- Fungsi transfer aktuator

$$\frac{\dot{m}(s)}{U(s)} = \frac{0,25}{0,252 s + 1}$$

$$\dot{m}(s)(0,252 s + 1) = 0,25 U(s)$$

$$0,252 s \dot{m}(s) + \dot{m}(s) = 0,25 U(s)$$

$$0,252 s \dot{m}(s) = 0,25 U(s) - \dot{m}(s)$$

$$s \dot{m}(s) = 0,992 U(s) - 3,968 \dot{m}(s)$$

$$\frac{d\dot{m}(t)}{dt} = 0,992 u(t) - v g 3,968 \dot{m}(t) \quad (3.49)$$

Sehingga dari persamaan 3.48 dan 3.49 didapatkan hasil matrik ruang keadaan sebagai berikut :

$$\begin{bmatrix} \frac{dP(t)}{dt} \\ \frac{d\dot{m}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -5,025 & 277,45 \\ 0 & -3,968 \end{bmatrix} \begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0,992 \end{bmatrix} [u(t)] \quad (3.50)$$

$$\begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P(t) \\ \dot{m}(t) \end{bmatrix} + [0][u(t)] \quad (3.51)$$

Dari persamaan diatas maka dapat diketahui nilai dari matriks A, B, C, D. Matriks A adalah matriks yang menyatakan state sistem. Berdasarkan penjabaran dari fungsi transfer plant dan aktuator tersebut, maka matriks A bernilai:

$$A = \begin{bmatrix} -5,025 & 277,45 \\ 0 & -3,968 \end{bmatrix}$$

Matriks B merupakan matriks yang menyatakan input kontrol sistem. Untuk mencari nilai matriks B didapatkan dari fungsi transfer aktuator, sehingga matriks B bernilai:

$$B = \begin{bmatrix} 0 \\ 0,992 \end{bmatrix}$$

Matriks C merupakan matriks output sistem dan merupakan matriks identitas sehingga matriks C bernilai:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Matriks D merupakan matriks yang menyatakan *noise* sistem, jika diasumsikan tidak terjadi *noise* maka matriks D bernilai:

$$D = 0$$

Matriks kesalahan terdapat dua macam, yakni kesalahan aktuator (Fa) dan sensor (Fs). Dalam tugas akhir ini hanya dianalisa kesalahan aktuator, sehingga matriks Fa bernilai:

$$Fa = [B]$$

Untuk mengkompensasi kesalahan aktuator maka diperlukan modifikasi algoritma *observer* baru, seperti pada persamaan 2.19 dan 2.20

Kemudian *state* baru $z(t)$ ditambahkan untuk memindahkan kesalahan aktuator dari persamaan keluaran ke persamaan keadaan, persamaan tersebut dapat didefinisikan persamaan baru pada persamaan 2.21, 2.22, dan 2.23.

Untuk Az merupakan matriks identitas (I), $Az \in Iq$. Agar dapat mendefinisikan matriks kesalahan aktuator, maka bentuk dari matriks ruang keadaan dapat diekspansi menjadi seperti pada persamaan 2.24 dan 2.25. Maka dapat dimasukkan nilai matrik sebagai berikut

$$Aa = \begin{bmatrix} A & 0 \\ Az.C & -Az \end{bmatrix}$$

$$= \begin{bmatrix} -5.0250 & 277.4500 & 0 & 0 \\ 0 & -3.9680 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$Ba = \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.9920 \\ 0 \\ 0 \end{bmatrix}$$

$$Ca = \begin{bmatrix} 0 \\ C \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Ea = \begin{bmatrix} F_a \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Untuk memperoleh nilai *gain observer*, maka ditambahkan *state* yang mengandung *gain* agar dapat mengestimasi kesalahan aktuator, sehingga ekspansi matriks ruang keadaan baru yang ditunjukkan pada persamaan 2.27. Sehingga dari persamaan diatas jika diubah kedalam bentuk yang lebih ringkas akan menjadi sebagai berikut.

$$\begin{bmatrix} \hat{X}(t) \\ \hat{f}_a(t) \end{bmatrix} = \begin{bmatrix} Aa & Ea \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{f}_s \end{bmatrix} + \begin{bmatrix} Ba \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_f \end{bmatrix} [Ca \quad 0] (Y - \hat{Y}) \quad (3.61)$$

$$\hat{Y}(t) = [Ca \quad 0] \begin{bmatrix} \hat{x} \\ \hat{f}_a \end{bmatrix} \quad (3.62)$$

Dengan mensubstitusikan nilai matriks Aa , Ca dan Ea , diperoleh nilai matriks \tilde{A}_a dan \tilde{C}_a sebagai berikut.

$$\tilde{A}_a = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -5.0250 & 277.4500 & 0 & 0 & 0 \\ 0 & -3.9680 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{C}_a = [C_a \quad 0] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Dari bentuk (3.61) dan (3.62) dapat diubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut.

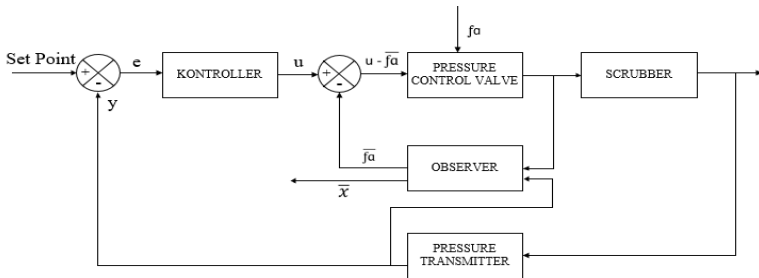
$$\hat{\tilde{x}}(t) = \tilde{A}_a \hat{\tilde{x}}(t) + \tilde{B}_a u(t) + \tilde{K}_e (\tilde{y}(t) - \hat{\tilde{y}}(t)) \quad (3.63)$$

$$\hat{\tilde{y}}(t) = \tilde{C}_a \hat{\tilde{x}}(t) \quad (3.64)$$

Selanjutnya dilakukan simulasi pada Matlab R2014a untuk didapatkan nilai gain observer yang sesuai dengan menggunakan metode LMI dimana listing dari kodingan LMI dilampirkan pada lampiran G dan F. Setelah didapatkan nilai gain observer maka dilakukan penyusunan skema observer menurut matriks ruang keadaan 3.62 dan 3.63 seperti tampak pada lampiran C

3.6.2 Rekonfigurasi Sinyal Pengendali untuk Aktuator

Rekonfigurasi pengendali digunakan untuk mengatasi kesalahan sehingga *plant* tetap beroperasi. Kesalahan yang terjadi pada aktuator menyebabkan sinyal pengendali (u) yang seharusnya masuk ke aktuator berubah menjadi kesalahan (*fault*) yang didefinisikan dengan ($fault + u$). Agar sinyal yang masuk ke aktuator adalah sinyal pengendali (u) maka hasil estimasi kesalahan dari *observer* akan dikurangkan setelah sinyal pengendali (u) dengan demikian kesalahan aktuator dapat diakomodasi.



Gambar 3.7 Skema Rekonfigurasi Sinyal Pengendali dari *Observer* Aktuator

3.7 Uji Performansi

Uji performansi dilakukan untuk menguji apakah algoritma FTC yang telah dibuat dapat berjalan dengan baik atau tidak. Uji ini dilakukan dengan memberikan suatu kesalahan pada sensor dan aktuator. Kesalahan yang diberikan berupa uji *loss of effectiveness* dan uji kebocoran untuk *observer* aktuator.

“Halaman ini sengaja dikosongkan”

BAB IV

ANALISIS DATA DAN PEMBAHASAN

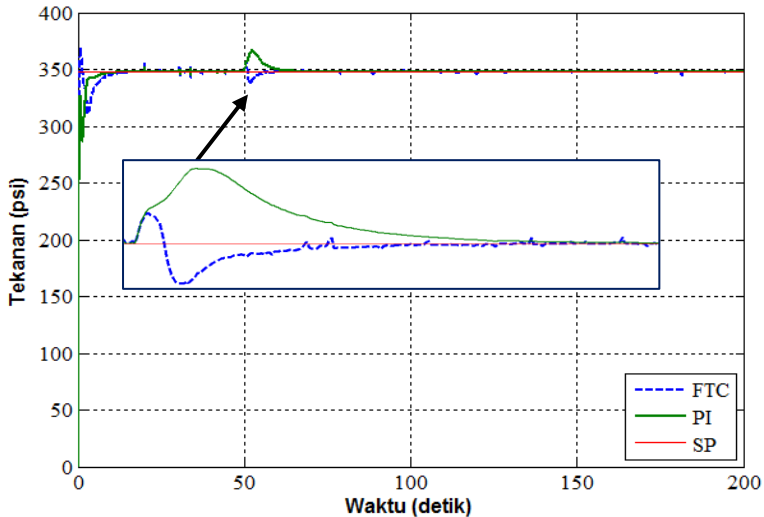
Uji performansi bertujuan untuk mengetahui algoritma FTC (*Fault Tolerant Control*) yang telah dirancang dapat bekerja dengan baik atau tidak jika diberikan kesalahan. Karena perancangan FTC pada tugas akhir ini berdasarkan pada kesalahan 47ctuator, maka bentuk uji kesalahan yang diberikan terhadap sistem terdiri dari jenis kesalahan pada 47ctuator. Parameter yang digunakan untuk menilai performansi sistem pengendalian diantaranya adalah *maximum overshoot*, *maximum undershoot*, *settling time*, dan *error steady state*. Penilaian terhadap uji yang diberikan dilakukan pada satu kondisi saja, yaitu kondisi saat sistem pengendalian mulai diberikan kesalahan, karena pada kondisi *start up* performansi sistem pengendalian PI dengan algoritma FTC sama dengan sistem pengendalian PI tanpa algoritma FTC. Penilaian terhadap uji dilakukan pada kondisi saat dikenai kesalahan di pada detik 50. Uji kesalahan aktuator yang diberikan berupa *loss of effectiveness* dan kesalahan kebocoran. Kesalahan *loss of effectiveness* pada 47ctuator dapat berupa penurunan gain 47ctuator akibat penyumbatan atau valve berkarat.

.1 Uji Kebocoran

Kebocoran merupakan kesalahan pada sistem akibat laju gas yang akan keluar dari *scrubber* tidak terumpan secara penuh. Nilai kebocoran yang diberikan pada uji ini adalah 15%, 50%, dan 75% dari nilai laju aliran gas *scrubber* yang disalurkan dari *control valve* hingga masuk ke proses selanjutnya. Semakin kecil nilai kesalahan berupa kebocoran maka nilai *error steady state* juga semakin kecil dan sebaliknya. Pengujian dilakukan dengan menggunakan *step time* pada *set point* pada detik 0 dan kesalahan pada detik 50. Hasil respon sistem pengendalian tekanan dengan kesalahan kebocoran 15% ditunjukkan pada gambar 4.1. Selanjutnya, respon sistem pengendalian tekanan dengan kesalahan kebocoran 50%, dan 75% masing – masing ditunjukkan pada gambar 4.2, dan gambar 4.3.

4.1.1 Uji Kesalahan kebocoran 15%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.1 berikut.



Gambar 4.1 Respon Sistem dengan Kesalahan Kebocoran 15%

Tabel 4.1 Parameter Performansi dengan Kesalahan Kebocoran 15%

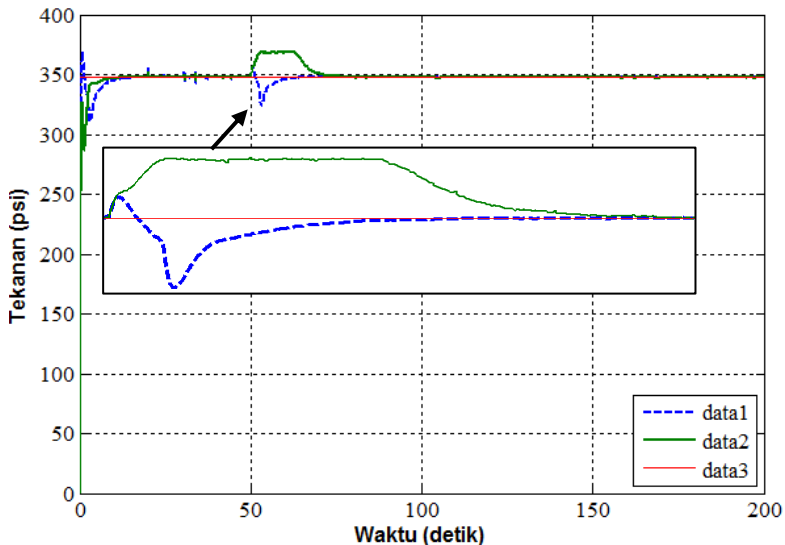
No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	5,25 %	2,127%
2	<i>Settling Time</i>	71 detik	56,1 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.1 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.1, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 15% lebih baik daripada sistem pengendalian

tanpa FTC. Sistem pengendalian dengan algoritma FTC memiliki *maximum undershoot* sebagai kompensasi dari nilai *maximum overshoot* yang kecil. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 15% dengan algoritma FTC menghasilkan *maximum overrrshoot* 3,13 % lebih sedikit, *settling time* 15 detik lebih cepat dan *error steady state* 0%.

4.1.2 Uji Kesalahan kebocoran 50%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.2 berikut.



Gambar 4.2 Respon Sistem dengan Kesalahan Kebocoran 50%

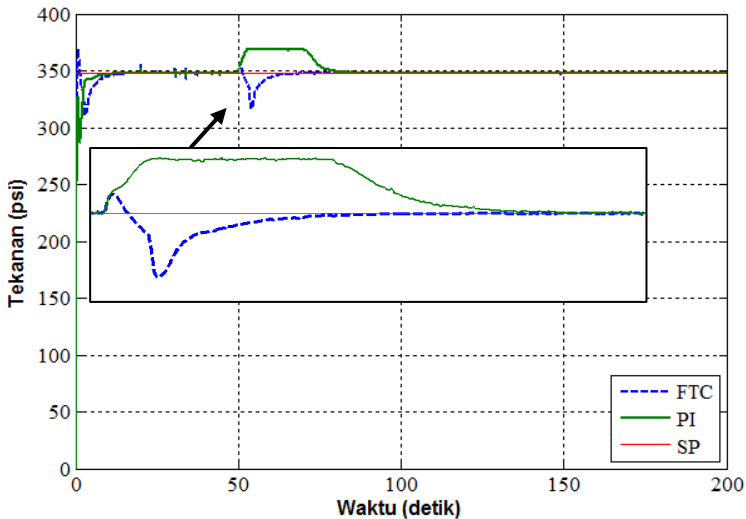
Tabel 4.2 Parameter Performansi dengan Kesalahan Kebocoran 50%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	6,034 %	2,155%
2	<i>Settling Time</i>	77 detik	67 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.2 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.2, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 50% lebih baik daripada sistem pengendalian tanpa FTC. Sistem pengendalian dengan algoritma FTC memiliki *maximum undershoot* sebagai kompensasi dari nilai *maximum overshoot* yang kecil. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 50% dengan algoritma FTC menghasilkan *maximum overrrshoot* 3,879 % lebih sedikit, *settling time* 10 detik lebih cepat dan *error steady state* 0%.

4.1.3 Uji Kesalahan kebocoran 75%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.3 berikut.



Gambar 4.3 Respon Sistem dengan Kesalahan Kebocoran 75%

Tabel 4.3 Parameter Performansi dengan Kesalahan Kebocoran 75%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	5,977 %	2,126%
2	<i>Settling Time</i>	86 detik	68 detik
3	<i>Error Steady State</i>	0,057%	0%

Pada Gambar 4.3 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.3, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 75% lebih baik daripada sistem pengendalian tanpa FTC. Sistem pengendalian dengan algoritma FTC memiliki *maximum undershoot* sebagai kompensasi dari nilai *maximum overshoot* yang kecil. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi

sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 75% dengan algoritma FTC menghasilkan *maximum overshoot* 3,851 % lebih sedikit, *settling time* 18 detik lebih cepat dan *error steady state* 0,057% lebih sedikit.

4.1.4 Pembahasan uji kebocoran

Pada gambar 4.1, gambar 4.2, dan gambar 4.3 serta penjabaran nilai parameter yang tampak pada tabel 4.1, tabel 4.2, dan tabel 4.3, terlihat bahwa respon sistem pengendalian dengan algoritma FTC mengatasi kesalahan nilai kebocoran 15% hingga 75% lebih baik daripada sistem pengendalian tanpa algoritma FTC ditunjukkan pada tabel 4.4 dibawah berupa perbandingan kesalahan kebocoran secara keseluruhan. Sistem pengendalian dengan atau tanpa algoritma FTC dapat mengembalikan nilai *process variable* menuju ke titik *set point*, namun dengan adanya perbedaan waktu. Perbedaan waktu inilah yang menunjukkan keunggulan sistem pengendalian dengan algoritma FTC. Dengan algoritma FTC, sistem pengendalian tersebut dapat kembali menuju keadaan normalnya dengan cepat setelah dikenai kesalahan jika dibandingkan dengan sistem pengendalian yang tanpa algoritma FTC. Dari segi parameter, *maximum overshoot*, *settling time*, maupun *error steady state* sistem pengendalian dengan algoritma FTC memiliki nilai error lebih kecil daripada sistem pengendalian tanpa algoritma FTC.

Tabel 4.4 Perbandingan uji kebocoran dengan kesalahan 15%, 50% dan 75%

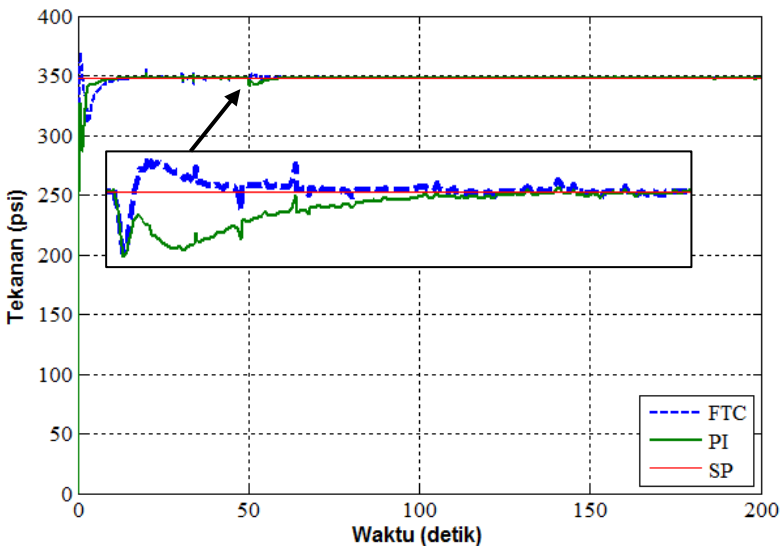
No	Parameter	15%	50%	75%
1	<i>Maximum Overshoot</i>	2,126%	2,155%	2,126%
2	<i>Settling Time</i>	71 detik	77 detik	86 detik
3	<i>Error Steady state</i>	0%	0%	0%

4.2 Uji Loss of Effectiveness

Loss of effectiveness adalah kesalahan pada *control valve* ketika bukaan valve terhambat oleh karat yang ditimbulkan dari material yang mengakibatkan berkurangnya sinyal pengendali yang diberikan untuk *control valve*. Nilai *loss of effectiveness* yang diberikan pada uji ini adalah 15%, 50%, dan 85% dari sinyal pengendali yang masuk pada aktuator. Hasil respon sistem pengendalian tekanan dengan kesalahan *loss of effectiveness* 15% ditunjukkan pada gambar 4.4. Selanjutnya, respon sistem pengendalian level dengan kesalahan *loss of effectiveness* 50%, dan 85% masing – masing ditunjukkan pada gambar 4.5 dan 4.6.

4.2.1 Uji Kesalahan Loss of Effectiveness 15%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.4 berikut.



Gambar 4.4 Respon Sistem dengan Kesalahan *Loss of Effectiveness* 15%

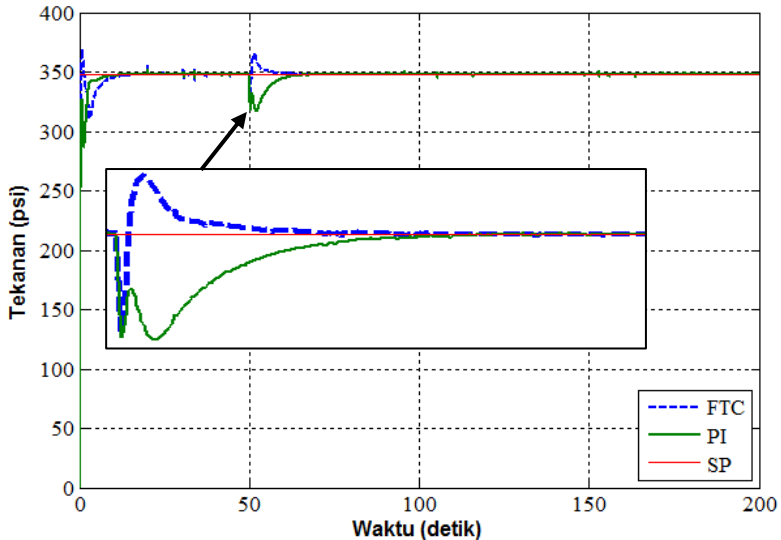
Tabel 4.5 Parameter Performansi dengan Kesalahan
Loss of Effectiveness 15%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	1,867 %	1,724%
2	<i>Settling Time</i>	63 detik	54 detik
3	<i>Error Steady State</i>	0.086%	0.057%

Pada Gambar 4.4 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.5, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan *loss of effectiveness* 15% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 15% dengan algoritma FTC menghasilkan *maximum overshoot* yang lebih tinggi dikarenakan pada kesalahan *loss of effectiveness* terjadi undershoot sehingga untuk mengkompensasi kesalahan maka FTC menghasilkan overshoot. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 15% dengan algoritma FTC menghasilkan *maximum overshoot* 0,143% lebih sedikit, *settling time* 9 detik lebih cepat dan *error steady state* 0,086% lebih sedikit.

4.2.2 Uji Kesalahan *Loss of Effectiveness* 50%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.5 berikut.



Gambar 4.5 Respon Perbesaran Sistem dengan Kesalahan *Loss of Effectiveness* 50%

Tabel 4.6 Parameter Performansi dengan Kesalahan *Loss of Effectiveness* 50%

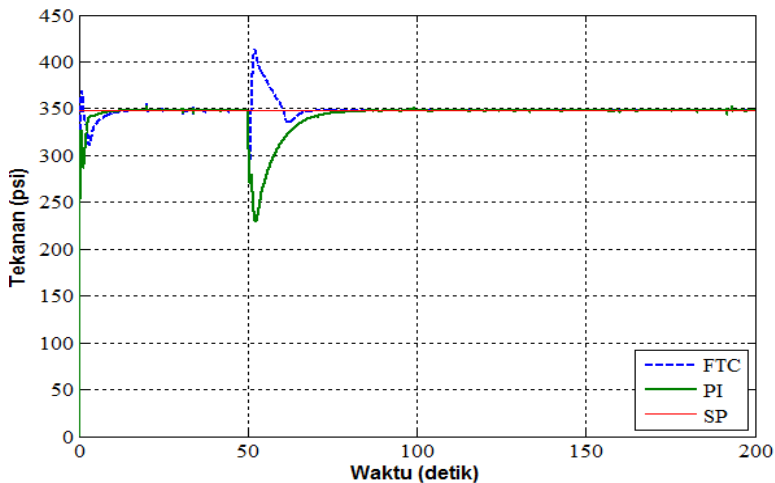
No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	8,62 %	8,045%
2	<i>Settling Time</i>	69 detik	61 detik
3	<i>Error Steady State</i>	0,02873%	0%

Pada Gambar 4.5 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.6, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan *loss of effectiveness* 85% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* yang lebih tinggi dikarenakan pada kesalahan *loss of effectiveness* terjadi

undershoot sehingga untuk mengkompensasi kesalahan maka FTC menghasilkan overshoot. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* 0,575% lebih sedikit, *settling time* 8 detik lebih cepat dan *error steady state* 0,0287% lebih sedikit.

4.2.3 Uji Kesalahan Loss of Effectiveness 85%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.6 berikut.



Gambar 4.6 Respon Sistem dengan Kesalahan *Loss of Effectiveness* 85%

Tabel 4.7 Parameter Performansi dengan Kesalahan
Loss of Effectiveness 85%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	33,90 %	19,54%
2	<i>Settling Time</i>	83 detik	60 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.6 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.7, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan *loss of effectiveness* 85% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 85% dengan algoritma FTC menghasilkan *maximum overshoot* yang lebih tinggi dikarenakan pada kesalahan *loss of effectiveness* terjadi undershoot sehingga untuk mengkompensasi kesalahan maka FTC menghasilkan overshoot. Hal ini tidak mempengaruhi performansi sistem pengendalian secara keseluruhan karena beberapa parameter lain yang masih menunjukkan bahwa performansi sistem pengendalian dengan algoritma FTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* 14,36% lebih sedikit, *settling time* 23 detik lebih cepat dan *error steady state* 0%.

4.2.4 Pembahasan uji kesalahan *Loss of Effectiveness*

Pada gambar 4.4, gambar 4.5, dan gambar 4.6 serta penjabaran nilai parameter yang tampak pada tabel 4.5, tabel 4.6, dan tabel 4.7 terlihat bahwa respon sistem pengendalian dengan algoritma FTC mengatasi kesalahan nilai *loss of effectiveness* 15% hingga 85% lebih baik daripada sistem pengendalian tanpa algoritma FTC dengan ditunjukkan pada tabel nilai hasil perbandingan keseluruhan uji *loss of effectiveness* FTC pada tabel

4.8 dibawah. Hal ini disebabkan oleh adanya rekonfigurasi pengendali atau aksi pengendali yang bekerja untuk mengatasi kesalahan tersebut. Sistem pengendalian dengan atau tanpa algoritma FTC dapat mengembalikan nilai *process variable* menuju ke titik *set point*, namun dengan perbedaan waktu. Perbedaan waktu inilah yang menunjukkan keunggulan sistem pengendalian dengan algoritma FTC. Dengan algoritma FTC, sistem pengendalian tersebut dapat kembali menuju keadaan normalnya dengan cepat setelah dikenai kesalahan jika dibandingkan dengan sistem pengendalian yang tanpa algoritma FTC. Dari segi parameter *rise time*, *settling time*, maupun *peak time* sistem pengendalian dengan algoritma FTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma FTC

Sistem pengendalian dengan algoritma FTC memiliki *maximum overshoot* sebagai kompensasi dari nilai *maximum undershoot* yang kecil. Walaupun memiliki ketimpangan dalam nilai *maximum overshoot*nya, tetapi apabila dibandingkan sistem pengendalian tanpa algoritma FTC, sistem pengendalian dengan algoritma FTC tetap memiliki performansi yang baik dalam ranah *safety*. Hal ini karena kerja sistem pengendalian yang baik adalah ditentukan oleh besarnya *error* yang dimiliki.

Tabel 4.8 Perbandingan uji *loss of effectiveness* dengan kesalahan 15%, 50% dan 75%

No	Parameter	15%	50%	75%
1	<i>Maximum Overshoot</i>	1,724%	8,045%	19,54%
2	<i>Settling Time</i>	54 detik	61 detik	60 detik
3	<i>Error Steady State</i>	0.057%	0%	0%

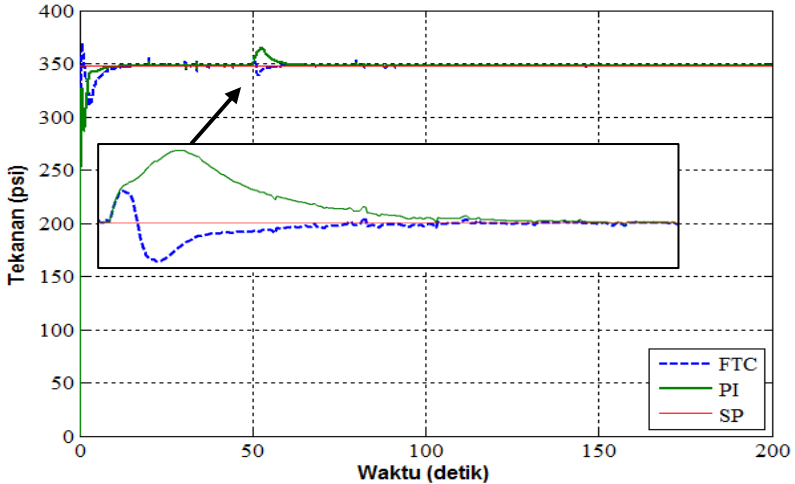
4.3 Uji kesalahan kebocoran dan *loss of effectiveness*

Pada Uji Kesalahan kebocoran dan *loss of effectiveness* dilakukan pengujian secara bersamaan dengan memberikan nilai kesalahan pada kebocoran dan kesalahan pada *loss of*

effectiveness. Uji ini dilakukan untuk mengetahui kinerja sistem ketika terjadi kesalahan secara bersamaan, apakah sistem tetap mampu bekerja dengan baik. Uji ini dilakukan dengan memberikan variasi kesalahan berupa kesalahan kebocoran sebesar 15%, 50%, dan 75% divariasikan dengan kesalahan *loss of effectiveness* 15%, 50%, dan 85%.

4.3.1 Uji Kesalahan kebocoran 15% dan *Loss of effectiveness* 15%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.7 berikut.



Gambar 4.7 Respon sistem dengan kesalahan kebocoran 15% dan *loss of effectiveness* 15%

Pada Gambar 4.7 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.9, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness* 15% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness*

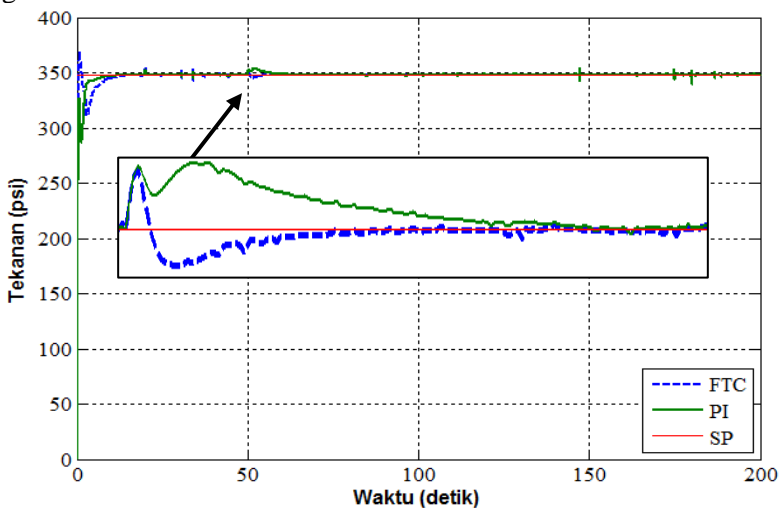
15% dengan algoritma FTC menghasilkan *maximum overshoot* 2,73% lebih sedikit, *settling time* 23 detik lebih cepat dan *Error Steady State* 0%.

Tabel 4.9 Parameter Performansi dengan Kesalahan Kebocoran 15% dan *loss of effectiveness* 15%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	4,885 %	2,155%
2	<i>Settling Time</i>	83 detik	60 detik
3	<i>Error Steady State</i>	0%	0%

4.3.2 Uji Kesalahan kebocoran 15% dan *Loss of effectiveness* 50%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.8 berikut.



Gambar 4.8 Respon sistem dengan kesalahan kebocoran 15% dan *loss of effectiveness* 50%

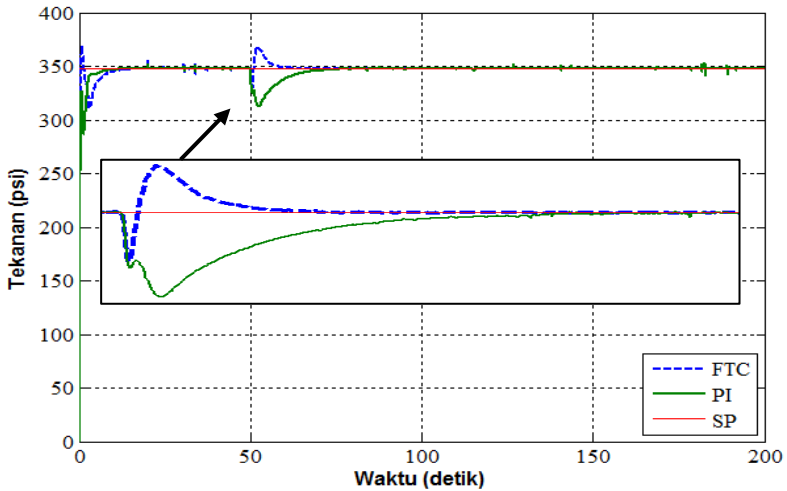
Tabel 4.10 Parameter Performansi dengan Kesalahan Kebocoran 15% dan *loss of effectiveness* 50%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	1,724 %	1,436%
2	<i>Settling Time</i>	63 detik	55 detik
3	<i>Error Steady State</i>	0,057%	0%

Pada Gambar 4.8 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.10, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness* 50% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overrrshoot* 0,288 lebih sedikit, *settling time* 8 detik lebih cepat dan *Error Steady State* 0,057% lebih sedikit.

4.3.3 Uji Kesalahan kebocoran 15% dan *Loss of effectiveness* 85%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.9. Pada Gambar 4.9 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.11, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness* 85% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 15% bersamaan dengan kesalahan *loss of effectiveness* 85% dengan algoritma FTC menghasilkan *maximum overrrshoot* 4,45% lebih sedikit, *settling time* 23 detik lebih cepat dan *Error Steady State* 0%.



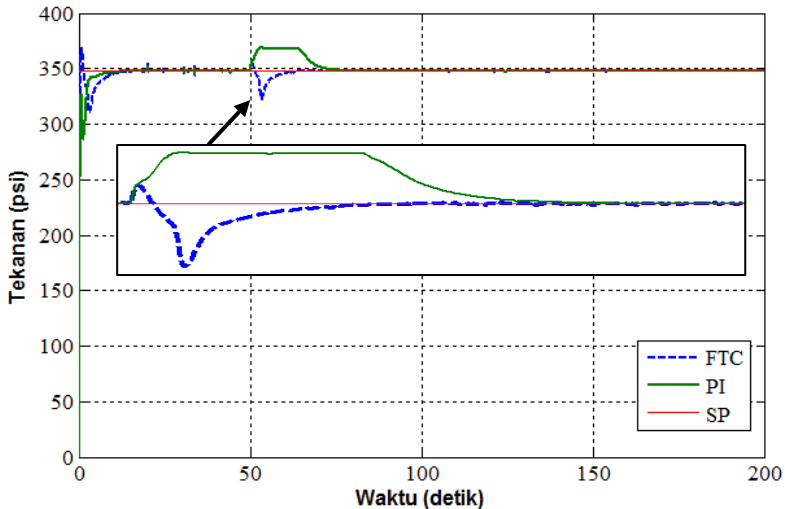
Gambar 4.9 Respon sistem dengan kesalahan kebocoran 15% dan *loss of effectiveness* 85%

Tabel 4.11 Parameter Performansi dengan Kesalahan Kebocoran 15% dan *loss of effectiveness* 85%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Undershoot</i>	10,344%	5,89%
2	<i>Settling Time</i>	75 detik	63 detik
3	<i>Error Steady State</i>	0%	0%

4.3.4 Uji Kesalahan kebocoran 50% dan *Loss of effectiveness* 15%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.10 berikut.



Gambar 4.10 Respon sistem dengan kesalahan kebocoran 50% dan *loss of effectiveness* 15%

Tabel 4.12 Parameter Performansi dengan Kesalahan Kebocoran 50% dan *loss of effectiveness* 15%

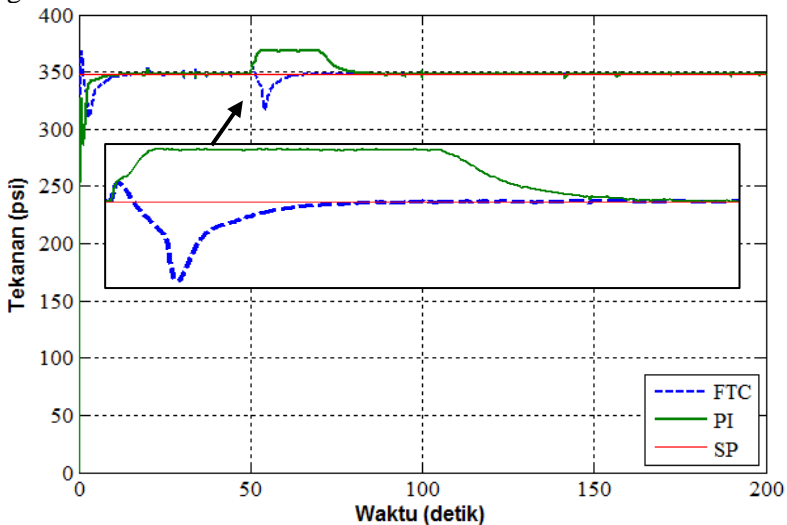
No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	7,183 %	5,74%
2	<i>Settling Time</i>	80 detik	67 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.10 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.12, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 15% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 15% dengan algoritma FTC menghasilkan *maximum*

overrshoot 1,443% lebih sedikit, *settling time* 13 detik lebih cepat dan *Error Steady State* 0%.

4.3.5 Uji Kesalahan kebocoran 50% dan *Loss of effectiveness* 50%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.11 berikut.



Gambar 4.11 Respon sistem dengan kesalahan kebocoran 50% dan *loss of effectiveness* 50%

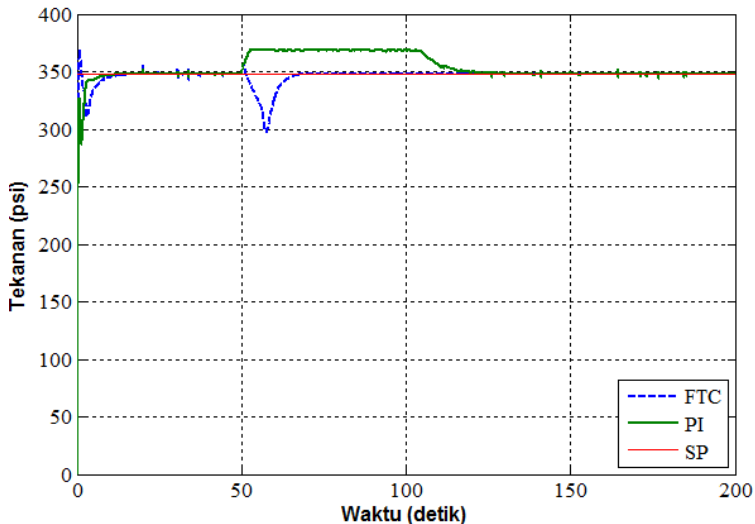
Tabel 4.13 Parameter Performansi dengan Kesalahan Kebocoran 50% dan *loss of effectiveness* 50%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	8,908 %	5,747%
2	<i>Settling Time</i>	86 detik	67 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.11 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.13, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 50% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* 3,161% lebih sedikit, *settling time* 19 detik lebih cepat dan *Error Steady State* 0%.

4.3.6 Uji Kesalahan kebocoran 50% dan *Loss of effectiveness* 85%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.12 berikut.



Gambar 4.12 Respon sistem dengan kesalahan kebocoran 50% dan *loss of effectiveness* 85%

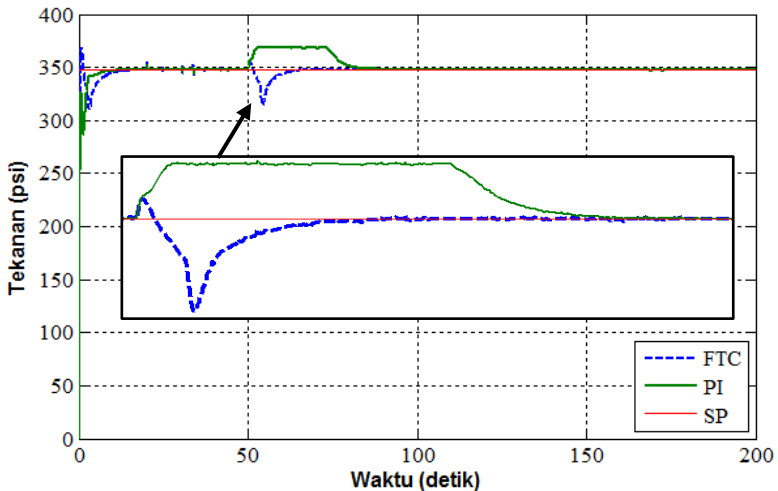
Tabel 4.14 Parameter Performansi dengan Kesalahan Kebocoran 50% dan *loss of effectiveness* 50%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	7,183%	5.04%
2	<i>Settling Time</i>	125 detik	72 detik
3	<i>Error Steady State</i>	0%	0%

Pada Gambar 4.12 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.14, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 85% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 50% bersamaan dengan kesalahan *loss of effectiveness* 85% dengan algoritma FTC menghasilkan *maximum overrrshoot* 2,143% lebih sedikit, *settling time* 53 detik lebih cepat dan *Error Steady State* 0%.

4.3.7 Uji Kesalahan kebocoran 75% dan *Loss of effectiveness* 15%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.13 berikut.



Gambar 4.13 Respon sistem dengan kesalahan kebocoran 75% dan *loss of effectiveness* 15%

Tabel 4.15 Parameter Performansi dengan Kesalahan Kebocoran 75% dan *loss of effectiveness* 15%

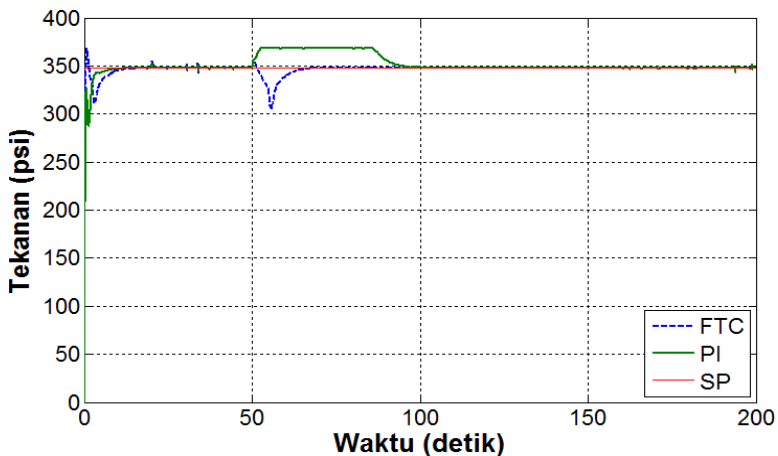
No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	9,77 %	6,1%
2	<i>Settling Time</i>	89 detik	68 detik
3	<i>Error Steady State</i>	0,0287%	0%

Pada Gambar 4.13 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.14, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 15% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 15% dengan algoritma FTC menghasilkan *maximum*

overshoot 3,67% lebih sedikit, *settling time* 21 detik lebih cepat dan *Error Steady State* 0.0287% lebih sedikit.

4.3.8 Uji Kesalahan kebocoran 75% dan *Loss of effectiveness* 50%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.14. Pada Gambar 4.14 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.16, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 50% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 50% dengan algoritma FTC menghasilkan *maximum overshoot* 6,351% lebih sedikit, *settling time* 28 detik lebih cepat dan *Error Steady State* 0,0287%.



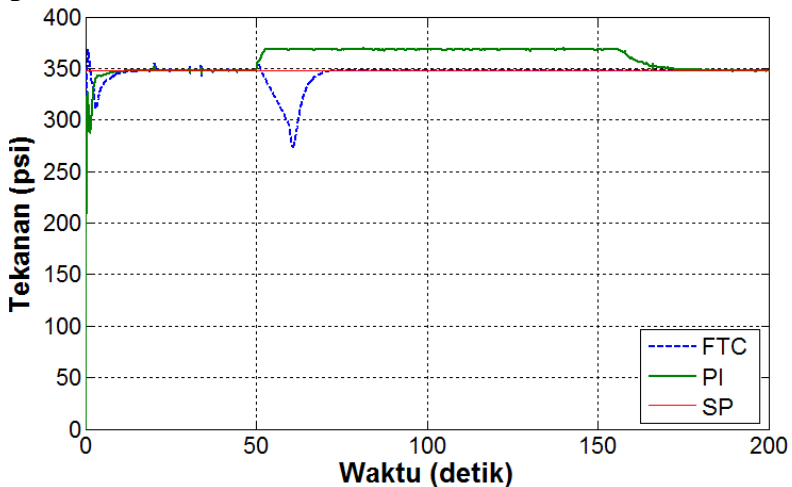
Gambar 4.14 Respon sistem dengan kesalahan kebocoran 75% dan *loss of effectiveness* 50%

Tabel 4.16 Parameter Performansi dengan Kesalahan Kebocoran 75% dan *loss of effectiveness* 50%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	12,3563 %	6,005%
2	<i>Settling Time</i>	97 detik	69 detik
3	<i>Error Steady State</i>	0,0287%	0%

4.3.9 Uji Kesalahan kebocoran 75% dan *Loss of effectiveness* 85%

Hasil uji respon sistem terhadap kesalahan ditinjau pada kondisi saat terjadi kesalahan (*fault*) pada detik ke-50 ditunjukkan pada gambar 4.15 berikut.



Gambar 4.15 Respon sistem dengan kesalahan kebocoran 75% dan *loss of effectiveness* 85%

Pada Gambar 4.15 serta penjabaran nilai parameter performansi sistem yang tampak pada tabel 4.17, terlihat bahwa performansi dalam sistem pengendalian FTC dapat mengatasi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of*

effectiveness 85% lebih baik daripada sistem pengendalian tanpa FTC. Dari segi parameter performansi sistem yaitu saat diberi kesalahan kebocoran 75% bersamaan dengan kesalahan *loss of effectiveness* 85% dengan algoritma FTC menghasilkan *maximum overshoot* 4,22% lebih sedikit, *settling time* 103 detik lebih cepat dan *Error Steady State* 0%.

Tabel 4.17 Parameter Performansi dengan Kesalahan Kebocoran 75% dan *loss of effectiveness* 85%

No	Parameter	PI tanpa FTC	PI dengan FTC
1	<i>Maximum Overshoot</i>	6,32%	2,1%
2	<i>Settling Time</i>	176detik	73 detik
3	<i>Error Steady State</i>	0%	0%

4.3.10 Pembahasan uji Kesalahan kebocoran dengan *loss of effectiveness*

Pada pengujian bersamaan kesalahan kebocoran dengan *loss of effectiveness* dilakukan dengan variasi kesalahan kebocoran 15%, 50%, dan 75% terhadap kesalahan *loss of effectiveness* 15%, 50%, dan 85% diketahui bahwa FTC dapat mengkompensasi kesalahan ketika terjadi secara bersamaan. Hal ini disebabkan oleh adanya rekonfigurasi pengendali atau aksi pengendali yang bekerja untuk mengatasi kesalahan tersebut. Sistem pengendalian dengan ataupun tanpa algoritma FTC dapat mengembalikan nilai *process variable* menuju ke titik *set point*, namun dengan perbedaan waktu. Perbedaan waktu inilah yang menunjukkan keunggulan sistem pengendalian dengan algoritma FTC. Dengan algoritma FTC, sistem pengendalian tersebut dapat kembali menuju keadaan normalnya dengan cepat setelah dikenai kesalahan jika dibandingkan dengan sistem pengendalian yang tanpa algoritma FTC. Dari segi parameter *maximum overshoot*, *settling time*, maupun *error steady state* sistem pengendalian dengan algoritma FTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma FTC.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil analisis data yang telah dilakukan, dapat disimpulkan bahwa :

- Perancangan sistem *observer* akibat kesalahan aktuator telah berhasil dilakukan. Parameter keberhasilannya adalah ketika nilai keluaran *observer* sesuai dengan nilai uji kesalahan yang diberikan.
- Sistem dengan FTC lebih mampu menjaga performansi ketika terjadi kesalahan pada aktuator dibandingkan dengan sistem tanpa FTC. Hal ini ditunjukkan oleh kriteria respon *maximum undershoot* dan *error settling time* yang lebih kecil dengan menggunakan sistem FTC hingga 14,3% lebih kecil dan 103 detik lebih cepat mencapai *set point*.
- Pengujian kesalahan pada Aktuator berupa uji kebocoran, uji *Loss of effectiveness*, serta uji variasi gabungan antara keduanya memberikan hasil bahwa sistem pengendalian tekanan dengan algoritma FTC mampu mengakomodasi kesalahan tersebut.

5.2 Saran

Pada tugas akhir ini, terdapat saran yang dapat dilakukan pada penelitian selanjutnya yaitu dapat mengimplementasikan FTC pada *plant rill* serta dapat menggabungkan kesalahan aktuator beserta sensor secara bersamaan.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- NORSOK Standard P-100, Norwegian Technology Center, Process System Rev.2 November, 2001.
- Huang Shanfang, Wen Yiqian, Wang Dong, “*Performance analysis of gas-liquid cylindrical cyclone (GLCC) separator with an inclined and perforated wall*”. Nuclear Science and Techniques 24. 2013 010603
- Youmin Zhang, Jin Jiang. “*Bibliographical Review on Reconfigurable Fault – Tolerant Control System*”. *Annual Reviews in Control*, vol. 32, issue 2, pp. 229 – 252, Desember 2008.
- K. Indriawati, T. Agustinah, A. Jazidie. “*Reconfigurable Fault – Tolerant Control of Linier System with Actuator and Sensor Faults*”. *IEEE Conference on Control System, Computing and Engineering*, 29 November – 1 Desember 2013.
- F. Gunterus. “*Falsafah Dasar : Sistem Pengendalian Proses*, Elex Media Komputindo”. Jakarta. 1997.
- Indriawati.Katherin, N.Sebe, T. Agustinah, A. Jazidie, “*Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear System with Simultaneous Actuator and Sensor Faults: Application to a DC Series Motor Speed Drive,*” *International Review of Automatic Control (I.R.E.A.CO)*, Vol.8,N. 6 ISSN 1974-6059, 2015.
- Marselina, Ria. (2016). *Perancangan Active Fault Tolerant Control akibat kesalahan sensor dan actuator pada sistem pengendalian level light gas oil (LGO) stripper unit fuel oil complex 1 PT. Pertamina (persero) RU IV Cilacap. Surabaya: Institut Teknologi Sepuluh Nopember.*
- H. Noura, D. Theilliol, J. Christophe P., A. Chamseddine. “*Fault – Tolerant Control Systems, Design and Practical Application*”. AIC (Advances Industrial Control), Springer. New York. 2009.

“Halaman ini sengaja dikosongkan”

BIODATA PENULIS



Anak Agung Yuniartha merupakan nama lengkap penulis dengan nama panggilannya, Artha. Penulis dilahirkan di kota Malang, Jawa Timur pada tanggal 19 Juli 1996 sebagai anak tunggal dari pasangan Agung Pribadi dan Yuniati. Riwayat pendidikan penulis adalah SD Negeri III Donomulyo tahun (2002 – 2008), SMP Negeri 1 Donomulyo tahun (2008 – 2011), SMA Negeri 1 Kepanjen tahun (2011 – 2014), D3 Metrologi dan Instrumentasi tahun (2014-2017). Penulis diterima sebagai mahasiswa S1 Teknik Fisika ITS pada tahun 2017, kemudian fokus pada bidang minat fisika rekayasa untuk menyelesaikan tugas akhirnya. Penulis dapat dihubungi melalui email : yuniarta199@gmail.com.