



TUGAS AKHIR TF 141581

**PERANCANGAN *ACTIVE FAULT TOLERANT CONTROL* AKIBAT KESALAHAN SENSOR DAN AKTUATOR PADA SISTEM PENGENDALIAN LEVEL *LIGHT GAS OIL (LGO) STRIPPER UNIT FUEL OIL COMPLEX I* PT. PERTAMINA (PERSERO) RU IV CILACAP**

**RIA MARSELLINA  
NRP 2411 100 070**

**Dosen Pembimbing  
Dr. Bambang Lelono Widjiantoro, ST, MT**

**JURUSAN TEKNIK FISIKA  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016**



**FINAL PROJECT TF 141581**

**DESIGN OF ACTIVE FAULT TOLERANT CONTROL  
DUE TO SENSOR AND ACTUATOR FAULTS IN  
LEVEL CONTROL SYSTEM LIGHT GAS OIL (LGO)  
STRIPPER UNIT FUEL OIL COMPLEX I  
PT. PERTAMINA (PERSERO) RU IV CILACAP**

**RIA MARSELLINA  
NRP 2411 100 070**

**Supervisor  
Dr. Bambang Lelono Widjiantoro, ST, MT**

**ENGINEERING PHYSICS DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2016**

**LEMBAR PENGESAHAN**

**PERANCANGAN *ACTIVE FAULT TOLERANT CONTROL* AKIBAT KESALAHAN SENSOR DAN AKTUATOR PADA SISTEM PENGENDALIAN LEVEL *LIGHT GAS OIL (LGO) STRIPPER UNIT FUEL OIL COMPLEX I* PT. PERTAMINA (PERSERO)  
RU IV CILACAP**

Oleh :

**Ria Marsellina**  
**NRP. 2411 100 070**

**Surabaya, Januari 2016**  
**Mengetahui/Menyetujui**

**Pembimbing**

  
**Dr. Bambang Lelono Widjiantoro, ST, MT**  
**NIP. 19690507 199512 1 001**

**Ketua Jurusan**  
**Teknik Fisika FTI – ITS**

  
**Agus Muhammad Hatta, ST, MSi, Ph.D**  
**NIP. 19780902 200312 1 002**

**PERANCANGAN ACTIVE FAULT TOLERANT  
CONTROL AKIBAT KESALAHAN SENSOR DAN  
AKTUATOR PADA SISTEM PENGENDALIAN LEVEL  
LIGHT GAS OIL (LGO) STRIPPER UNIT FUEL OIL  
COMPLEX I PT. PERTAMINA (PERSERO)  
RU IV CILACAP**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada  
Bidang Studi Rekayasa Instrumentasi dan Kontrol  
Program Studi S – 1 Jurusan Teknik Fisika  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Oleh :

**RIA MARSELLINA**  
**NRP. 2411 100 070**

Disetujui oleh Tim Penguji Tugas Akhir :

- |   |                                    |       |               |
|---|------------------------------------|-------|---------------|
| 1 | Dr. Bambang Lelono W., ST, MT      | ..... | Pembimbing    |
| 2 | Ir. Ya'umar, MT                    | ..... | Ketua Penguji |
| 3 | Dr. Ir. Aulia Siti Aisjah, MT      | ..... | Penguji I     |
| 4 | Dr.-Ing. Doty Dewi Risanti, ST, MT | ..... | Penguji II    |
| 5 | Arief Abdurrahman, ST, MT          | ..... | Penguji III   |

**SURABAYA**  
**Januari 2016**

**PERANCANGAN ACTIVE FAULT TOLERANT  
CONTROL AKIBAT KESALAHAN SENSOR DAN  
AKTUATOR PADA SISTEM PENGENDALIAN LEVEL  
LIGHT GAS OIL (LGO) STRIPPER UNIT FUEL OIL  
COMPLEX I PT. PERTAMINA (PERSERO)  
RU IV CILACAP**

**Nama Mahasiswa : Ria Marsellina**  
**NRP : 2411 100 070**  
**Jurusan : Teknik Fisika**  
**Dosen Pembimbing : Dr. Bambang Lelono W., ST, MT**

**Abstrak**

*Performansi sistem pengendali tidak lepas dari kinerja instrumennya seperti sensor dan aktuator. Seringkali, sensor dan aktuator mengalami kesalahan operasi. Kesalahan dapat diperbaiki dengan melakukan perancangan active fault tolerant control (AFTC). Pada penelitian ini, dilakukan perancangan AFTC. Pertama, dilakukan pemodelan sistem terdiri dari pemodelan plant, aktuator, dan sensor dilanjutkan merancang kontroler konvensional, kemudian merancang observer sensor dan aktuator dengan gain observer sensor dan aktuator didapatkan dari perhitungan numerik linear matrix inequality (LMI). Terakhir merekonfigurasi sinyal pengendali. Dilakukan tiga jenis pengujian kesalahan untuk melihat performansi rancangan AFTC, yaitu uji sensor, uji aktuator, dan uji sensor – aktuator. Jenis uji kesalahan yang diberikan berupa lost of effectiveness dan kebocoran untuk aktuator serta bias, noise, dan sensitivitas untuk sensor. AFTC memiliki performansi jauh lebih baik dalam mengatasi kesalahan dibandingkan dengan pengendalian konvensional. Terbukti pada uji sensitivitas 90% untuk sistem pengendalian konvensional memiliki nilai parameter maximum overshoot 11,49%, maximum undershoot 0%, error steady state 11,11*

*% , dan settling time 5232 detik, sedangkan sistem pengendali dengan algoritma AFTC memiliki nilai maximum overshoot 0,06%, maximum undershoot 0%, error steady state 0 %, dan settling time 4548 detik. Namun untuk uji sensor – aktuator, kedua observer tidak dapat bekerja karena proses rekonfigurasi sinyal pengendali yang berjalan pada waktu yang sama.*

***Kata kunci*** : *active fault tolerant control, linear matrix inequality, sensor, aktuator, observer.*

**DESIGN OF ACTIVE FAULT TOLERANT CONTROL  
DUE TO SENSOR AND ACTUATOR FAULTS IN LEVEL  
CONTROL SYSTEM LIGHT GAS OIL (LGO) STRIPPER  
UNIT FUEL OIL COMPLEX I PT. PERTAMINA  
(PERSERO) RU IV CILACAP**

**Name** : Ria Marsellina  
**Student Number** : 2411 100 070  
**Department** : Engineering Physics  
**Supervisor** : Dr. Bambang Lelono W., ST, MT

**Abstract**

*Performance of control systems can not be separated from the performance of instruments such as sensor and actuator. Often, sensor and actuator experienced fault operation. Fault can be repaired by designing active fault tolerant control (AFTC). In this study, carried out the design of AFTC. First, the modeling system consists of plant modelling, actuator modelling, and sensor modelling, then continued by designing a conventional controller, then design the observer sensor and actuator which gain observer sensor and actuator is obtained from a numerical calculation of linear matrix inequality (LMI). The last, reconfigure the signal controller. There are three types of fault test to see the performance of design AFTC, the test of sensor, test of actuator, and test of sensor - actuator. The type of fault test is given in the form of lost of effectiveness and leakage to actuator, bias, noise, and sensitivity to sensor. AFTC has much better performance in dealing with fault compared with conventional control. Proven in test sensitivity of 90% for the conventional control system has a parameter value 11.49% for maximum overshoot, 0% for maximum undershoot, 11.11% for steady state error, and 5232 seconds for settling time, while the control system with algorithms AFTC has 0.06% for maximum overshoot, 0% for maximum undershoot,*

0% for error steady state, and 4548 seconds for settling time. But, to test the sensor - actuator, both of observer can not work, due to the reconfiguration of the control signal that runs at the same time.

**Keywords** : active fault tolerant control, linear matrix inequality, sensor, aktuator, observer.

## KATA PENGANTAR

Puji syukur atas segala limpahan rahmat, rezeki, kesehatan, keselamatan, dan ilmu yang Tuhan Yang Maha Esa berikan kepada penulis hingga mampu menyelesaikan laporan tugas akhir dengan judul :

**PERANCANGAN *ACTIVE FAULT TOLERANT CONTROL* AKIBAT KESALAHAN SENSOR DAN AKTUATOR PADA SISTEM PENGENDALIAN LEVEL *LIGHT GAS OIL (LGO) STRIPPER UNIT FUEL OIL COMPLEX I PT. PERTAMINA (PERSERO)*  
RU IV CILACAP**

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

1. Bapak Agus Muhammad Hatta, ST, M.Si, Ph.D selaku Ketua Jurusan Teknik Fisika ITS
2. Bapak Dr. Bambang Lelono Widjiantoro, ST. MT., dan Ibu Katherin Indriawati ST. MT., serta Bapak Totok Ruki Biyanto ST. MT., selaku dosen pembimbing yang dengan sabar membimbing, memberikan saran dan kritiknya.
3. Bapak Ahmad Furchan, selaku Insp. Inst Area di bagian MPS (*Maintenance Planning and Support*) yang sekaligus sebagai pembimbing lapangan yang telah membimbing dan memfasilitasi selama pengambilan data lapangan.
4. Bapak Noer Siswadi dan Ibu Nanik M. selaku orang tua dari Arina yang sudah banyak membantu dalam hal moril maupun materiil bagi kelancaran pengerjaan tugas akhir ini.
5. Orang tua saya, Fransiskus Subagyo dan Lidya Pudji Andayani, serta kakek dan nenek saya yang selalu berdoa untuk kebaikan saya.

6. Seseorang yang selalu mendampingi, mendukung, mengingatkan, dan mendoakan saya hingga saat ini, Jeffry Omega Prima.
7. Senior yang senantiasa menanyakan kabar, peduli, dan perhatian, serta mendukung dan mendoakan selalu, Nur Ulfa Hidayatullah.
8. Teman – teman Teknik Fisika Angkatan 2011, terutama Arina, Arum, Riza Aris, Nisa, Aulia, Firda, Damas, Dicky, Nur Kholish, Vany, Okky, Gigih, Khusnul, dan sahabat yang senantiasa memberikan semangat dan motivasinya.
9. Kerabat jauh saya, Mbah Kolim sekeluarga dan Mbah Basuki sekeluarga yang turut mempermudah segala sesuatunya ketika berada di cilacap.
10. Kerabat lain yang tidak dapat disebutkan satu persatu yang turut menyumbangkan semangat dan doanya bagi kebaikan saya.

Demikian laporan tugas akhir ini dibuat dengan sebaik – baiknya. Semoga laporan ini bermanfaat bagi semua pihak, khususnya untuk kemajuan industri *oil and gas* di Indonesia.

Surabaya, Januari 2016

Penulis

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b>	v
<b>ABSTRAK</b>	ix
<b>KATA PENGANTAR</b>	xiii
<b>DAFTAR ISI</b>	xv
<b>DAFTAR GAMBAR</b>	xix
<b>DAFTAR TABEL</b>	xxiii
<b>DAFTAR SINGKATAN</b>	xxv
<b>BAB I PENDAHULUAN</b>	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat	4
1.6 Sistematika Laporan	5
<b>BAB II TINJAUAN PUSTAKA</b>	7
2.1 Proses <i>Stripping</i>	7
2.2 Kolom <i>Stripper</i>	8
2.3 Kolom <i>Steam Stripper</i>	10
2.4 Model Matematis Kolom <i>Stripper</i>	11
2.5 <i>Light Gas Oil</i> (LGO)	17
2.6 <i>Light Gas Oil</i> (LGO) <i>Stripper</i>	18
2.7 Sistem Pengendalian Level LGO <i>Stripper</i>	20
2.8 Pengendali <i>Proportional Integral Derivative</i> (PID)	20
2.9 Perancangan Pengendali PI	22
2.10 Sensor Level	22
2.11 <i>Control Valve</i>	24
2.12 <i>Software</i> HYSYS	25
2.13 FOPDT ( <i>First Order Plus Dead Time</i> )	26
2.14 <i>Fault Tolerant Control</i>	27
2.15 <i>Observer State</i>	31

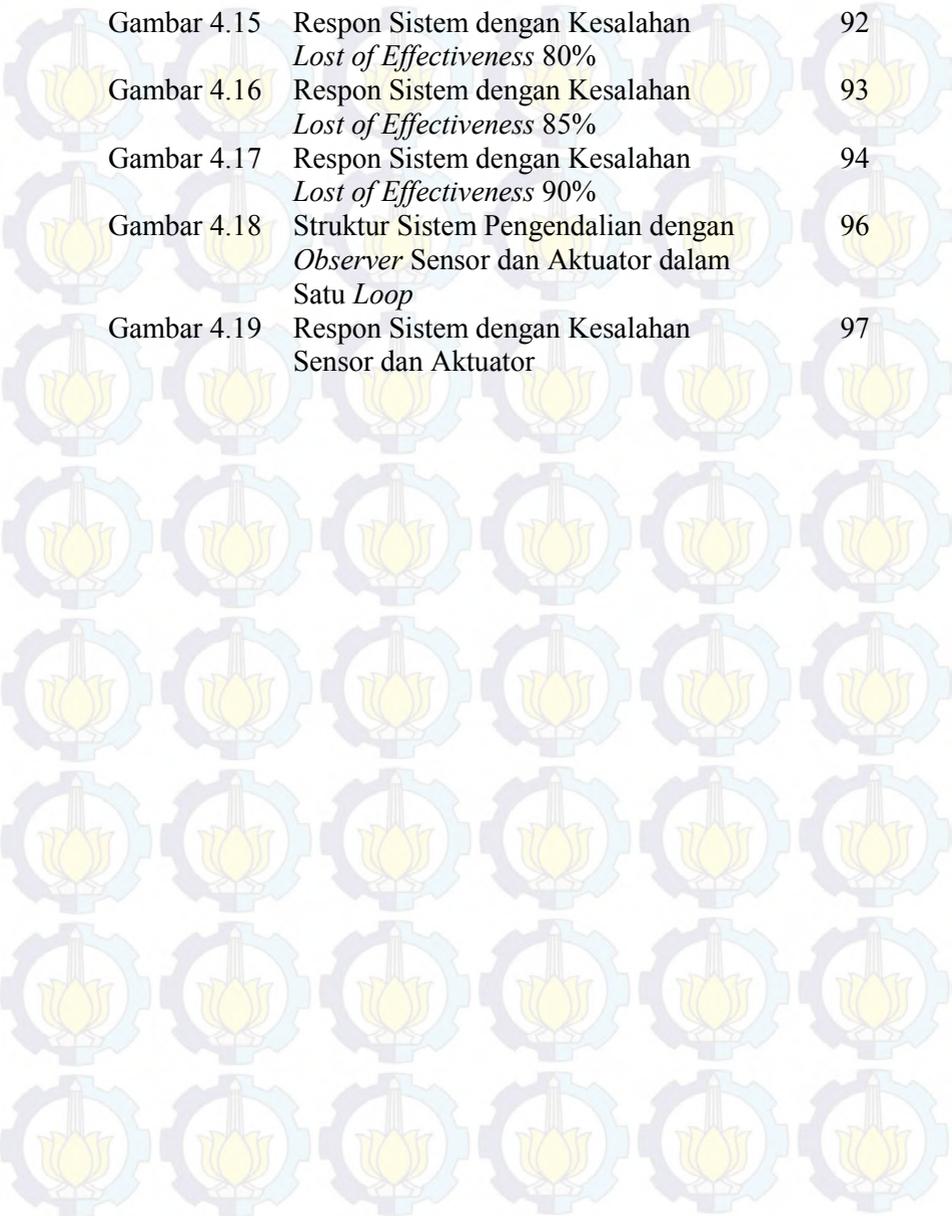
2.16	<i>Linear Matrix Inequality (LMI)</i>	32
<b>BAB III METODOLOGI PENELITIAN</b>		37
3.1	Prosedur Umum Penelitian	37
3.2	Pengambilan Data LGO <i>Stripper</i>	40
3.3	Pemodelan LGO <i>Stripper</i>	42
3.4	Pemodelan Matematis Aktuator	48
3.5	Pemodelan Matematis Sensor	49
3.6	Validasi Pemodelan	50
3.7	Perancangan Pengendali <i>Proportional Integral (PI)</i>	57
3.8	Perancangan <i>Active Fault Tolerant Control</i>	58
3.8.1	Perancangan <i>Observer</i> untuk Sensor	58
3.8.2	Rekonfigurasi Sinyal Pengendali untuk Sensor	65
3.8.3	Perancangan <i>Observer</i> untuk Aktuator	66
3.8.4	Rekonfigurasi Sinyal Pengendali untuk Aktuator	72
3.9	Uji Performansi	73
<b>BAB IV ANALISIS DATA DAN PEMBAHASAN</b>		75
4.1	Uji Kesalahan Sensor	75
4.1.1	Uji Kesalahan Bias	75
4.1.2	Uji Kesalahan Sensitivitas	78
4.1.3	Uji <i>Noise</i>	83
4.2	Uji Kesalahan Aktuator	84
4.2.1	Uji Kebocoran	84
4.2.2	Uji <i>Lost of Effectiveness</i>	89
4.3	Uji Kesalahan Sensor dan Aktuator	95
<b>BAB V KESIMPULAN DAN SARAN</b>		99
5.1	Kesimpulan	99
5.2	Saran	99
<b>DAFTAR PUSTAKA</b>		101

<b>LAMPIRAN A</b>	105
<i>Process Flow Diagram Plant Light Gas Oil Stripper</i>	105
11C4 Unit Fuel Oil Complex I PT. Pertamina (Persero) RU IV Cilacap	105
<b>LAMPIRAN B</b>	106
<i>Process Flow Distributed Control System Unit Control</i>	106
<i>Room Plant Light Gas Oil Stripper</i> 11C4 Unit Fuel Oil Complex I PT. Pertamina (Persero) RU IV Cilacap	106
<b>LAMPIRAN C</b>	107
C.1 Grafik Level LGO Stripper 11C4 Unit FOC I PT. Pertamina (Persero) RU IV Cilacap 30 Januari hingga 3 Februari 2015	107
C.2 Data Pengukuran Level LGO Stripper 11C4 Unit FOC I PT. Pertamina (Persero) RU IV Cilacap 30 Januari hingga 3 Februari 2015	108
C.3 Data Pengambilan Parameter FOPDT dari Simulasi HYSYS dengan <i>Opening Valve</i> 20% dan 30%	115
<b>LAMPIRAN D</b>	117
D.1 Hasil Perancangan <i>Simulink</i> MATLAB <i>General</i> <i>Observer</i> Sensor	117
D.2 Hasil Perancangan <i>Simulink</i> MATLAB <i>General</i> <i>Observer</i> Aktuator	118
D.3 Hasil Perancangan <i>Simulink</i> MATLAB <i>Observer</i> Sensor	119
D.4 Hasil Perancangan <i>Simulink</i> MATLAB <i>Observer</i> Aktuator	119
D.5 Hasil Perancangan <i>Simulink</i> MATLAB Kesalahan Sensor	120
D.6 Hasil Perancangan <i>Simulink</i> MATLAB Kesalahan Aktuator	120
<b>BIODATA PENULIS</b>	121

## DAFTAR GAMBAR

Gambar 2.1	Diagram Menara <i>Stripper</i>	8
Gambar 2.2	<i>Tray Stripper</i> (kiri) dan Menara <i>Stripper</i> (kanan)	9
Gambar 2.3	Kolom <i>Steam Stripper</i>	10
Gambar 2.4	Proses <i>Rectifying</i> pada Kolom <i>Stripper</i>	12
Gambar 2.5	Proses <i>Feed</i> pada Kolom <i>Stripper</i>	13
Gambar 2.6	Proses <i>General</i> pada Kolom <i>Stripper</i>	15
Gambar 2.7	Proses <i>Stripping</i> pada Kolom <i>Stripper</i>	16
Gambar 2.8	Proses LGO Proses <i>Stripper</i>	19
Gambar 2.9	Diagram Blok PID	22
Gambar 2.10	Bentuk Umum <i>Control Valve</i>	25
Gambar 2.11	Tampilan Muka Aspen HYSYS	26
Gambar 2.12	Skema Sistem FTC	28
Gambar 2.13	Struktur Umum AFTCS	29
Gambar 2.14	Klasifikasi Metode FDI	30
Gambar 2.15	Rekonfigurasi Pengendali pada AFTC	31
Gambar 2.16	Blok Diagram <i>Observer</i>	32
Gambar 3.1	Diagram Alir Pengerjaan Tugas Akhir	39
Gambar 3.2	<i>Process Flow Diagram</i> LGO <i>Stripper</i>	42
Gambar 3.3	<i>Process Flow Sheet Diagram</i> Kolom LGO <i>Stripper</i> pada HYSYS	44
Gambar 3.4	Grafik Hasil Simulasi <i>Open Loop</i> pada <i>Software</i> HYSYS	46
Gambar 3.5	Grafik Perbandingan antara Hasil Simulasi <i>Software</i> HYSYS dengan <i>Real Plant</i>	55
Gambar 3.6	Skema Blok Diagram Simulasi MATLAB <i>Open Loop</i>	56
Gambar 3.7	Grafik Respon <i>Open Loop</i> Sistem	56
Gambar 3.8	Skema Blok Diagram Simulasi MATLAB <i>Closed Loop</i>	57
Gambar 3.9	Grafik Respon Simulasi Sistem Pengendalian <i>Closed Loop</i> LGO <i>Stripper</i>	58

Gambar 3.10	Blok Diagram <i>Fault</i> Sensor	60
Gambar 3.11	Struktur <i>Observer</i> untuk Sensor	64
Gambar 3.12	Skema Rekonfigurasi Sinyal Pengendali dari <i>Observer</i> Sensor	65
Gambar 3.13	Blok Diagram <i>Fault</i> Aktuator	68
Gambar 3.14	Struktur <i>Observer</i> untuk Aktuator	72
Gambar 3.15	Skema Rekonfigurasi Sinyal Pengendali dari <i>Observer</i> Aktuator	73
Gambar 4.1	Respon Sistem dengan Kesalahan Bias +1,5%	76
Gambar 4.2	Respon Sistem dengan Kesalahan Bias -1,5%	77
Gambar 4.3	Respon Sistem dengan Kesalahan Sensitivitas 90%	78
Gambar 4.4	Respon Sistem dengan Kesalahan Sensitivitas 85%	79
Gambar 4.5	Respon Sistem dengan Kesalahan Sensitivitas 80%	80
Gambar 4.6	Respon Sistem dengan Kesalahan Sensitivitas 75%	81
Gambar 4.7	Respon Sistem dengan Pemberian <i>Noise</i>	83
Gambar 4.8	Perbesaran Respon Sistem dengan Pemberian <i>Noise</i>	84
Gambar 4.9	Respon Sistem dengan Kesalahan Kebocoran 10%	85
Gambar 4.10	Respon Sistem dengan Kesalahan Kebocoran 15%	86
Gambar 4.11	Respon Sistem dengan Kesalahan Kebocoran 20%	87
Gambar 4.12	Respon Sistem dengan Kesalahan Kebocoran 30%	88
Gambar 4.13	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 70%	90
Gambar 4.14	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 75%	91



Gambar 4.15	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 80%	92
Gambar 4.16	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 85%	93
Gambar 4.17	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 90%	94
Gambar 4.18	Struktur Sistem Pengendalian dengan <i>Observer</i> Sensor dan Aktuator dalam Satu <i>Loop</i>	96
Gambar 4.19	Respon Sistem dengan Kesalahan Sensor dan Aktuator	97

## DAFTAR TABEL

Tabel 3.1	Rincian Data <i>Plant</i>	40
Tabel 3.2	Tabel Data Proses Perusahaan dengan OP 22,9%	50
Tabel 3.3	Tabel Data Proses Perusahaan dengan OP 20,9%	50
Tabel 3.4	Tabel Data Proses Perusahaan dengan OP 22,36%	51
Tabel 3.5	Tabel Data Proses Perusahaan dengan OP 23%	51
Tabel 3.6	Tabel Data Proses Perusahaan dengan OP 23,3%	51
Tabel 3.7	Tabel Data Proses Perusahaan dengan OP 24,6%	51
Tabel 3.8	Tabel Data Proses Perusahaan dengan OP 24,1%	52
Tabel 3.9	Tabel Data Proses Perusahaan dengan OP 24,3%	52
Tabel 3.10	Tabel Data Proses Perusahaan dengan OP 23,7%	52
Tabel 3.11	Tabel Data Proses Perusahaan dengan OP 23,4%	52
Tabel 3.12	Tabel Data Proses Perusahaan dengan OP 23,2%	52
Tabel 3.13	Tabel Data Proses Perusahaan dengan OP 23,9%	52
Tabel 3.14	Tabel Data Proses Perusahaan dengan OP 23,1%	53
Tabel 3.15	Tabel Data Proses Perusahaan dengan OP 23,6%	53
Tabel 3.16	Tabel Data Proses Perusahaan dengan OP 23,8%	53
Tabel 3.17	Tabel Data Simulasi HYSYS	53

Tabel 3.18	Tabel Perhitungan Nilai <i>Error</i> dari <i>Process Value</i> Asli dengan Simulasi	54
Tabel 4.1	Parameter Performansi dengan Kesalahan Bias +1,5%	76
Tabel 4.2	Parameter Performansi dengan Kesalahan Bias -1,5%	77
Tabel 4.3	Parameter Performansi dengan Kesalahan Sensitivitas 90%	79
Tabel 4.4	Parameter Performansi dengan Kesalahan Sensitivitas 85%	80
Tabel 4.5	Parameter Performansi dengan Kesalahan Sensitivitas 80%	81
Tabel 4.6	Parameter Performansi dengan Kesalahan Sensitivitas 75%	82
Tabel 4.7	Parameter Performansi dengan Kesalahan Kebocoran 10%	85
Tabel 4.8	Parameter Performansi dengan Kesalahan Kebocoran 15%	86
Tabel 4.9	Parameter Performansi dengan Kesalahan Kebocoran 20%	87
Tabel 4.10	Parameter Performansi dengan Kesalahan Kebocoran 30%	88
Tabel 4.11	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 70%	90
Tabel 4.12	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 75%	91
Tabel 4.13	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 80%	92
Tabel 4.14	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 85%	93
Tabel 4.15	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 90%	94

## DAFTAR SINGKATAN

AFTC	: Active Fault Tolerant Control
ASTM	: American Standard Testing and Material
ATC	: Air to Close
ATO	: Air to Open
CDU	: Crude Distillation Unit
FDI	: Fault Detection and Identification
FOC	: Fuel Oil Complex
FOPDT	: First Order Plus Dead Time
FTCS	: Fault Tolerant Control System
HART	: Highway Addressable Remote Transducer
HAZOP	: Hazard and Operability
HYSYS	: Hypothetical System
LGO	: Light Gas Oil
LMI	: Linear Matrix Inequality
LOC	: Lube Oil Complex
MATLAB	: Matrix Laboratory
MIMO	: Multi Input Multi Output
PFTC	: Passive Fault Tolerant Control
PI	: Proportional Integral
RC	: Reconfigurable Controller
RU	: Refinery Unit
SRK	: Soave Redlich – Kwong



*Halaman ini sengaja dikosongkan*

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

PT. Pertamina (Persero) *Refinery Unit IV* Cilacap memiliki arah kerja mengolah *crude oil* guna mendapatkan produk yang lebih bernilai. Secara umum, proses pengolahan minyak di *refinery* dibagi menjadi dua, yaitu *primary process* dan *secondary process*. Pada *primary process*, pengolahan *crude oil* dilakukan secara fisis, yaitu menggunakan teknik penyulingan (distilasi). Proses distilasi terdapat pada unit *crude distillation*, yang merupakan unit terpenting dalam pemisahan *crude oil* menjadi beberapa jenis fraksi. Salah satu jenis produk minyak yang dihasilkan adalah *light gas oil* [1].

*Light gas oil* atau disingkat LGO masih tergolong jenuh karena kandungan minyak yang masih bercampur dengan fraksi ringan yang tidak diharapkan seperti CO<sub>2</sub>, garam, dan lain – lain. Oleh karena itu, dalam satu kesatuan *Crude Distillation Unit* terdapat instrumen tambahan berupa *stripper*. *Stripper* merupakan instrumen pendukung pada proses pengolahan *crude oil* yang bertujuan untuk memperbaiki kualitas produk yang dihasilkan, dengan kata lain melakukan pemurnian terhadap jenis produk tersebut. LGO yang diumpun ke dalam *stripper* tersebut, fraksi ringannya akan dipisahkan menggunakan aliran *low pressure steam* sehingga menghasilkan gas LGO sebagai *top product* dan LGO sebagai *bottom product*. LGO sebagai *bottom product* belum murni seutuhnya (masih bercampur dengan cairan hasil pengembunan *steam*) sehingga dibutuhkan kolom *drier* yang digunakan untuk menarik *steam*. *Stripper* di PT. Pertamina (Persero) RU IV Cilacap, khususnya di unit FOC I – CDU I dilengkapi dengan kolom *drier* di bagian bawahnya. Pengoperasian *stripper* dan *drier* harus memperhatikan semua variabel proses dan kondisi operasi agar didapatkan hasil dengan mutu dan jumlah yang optimal sesuai dengan spesifikasi [2].

Oleh karena pentingnya peranan *stripper* dalam proses pengolahan *crude oil* khususnya di *Crude Distillation Unit*, maka

peranan sistem pengendalian pada *stripper* menjadi penting. Proses pengendalian yang terpenting adalah menjaga kestabilan level LGO pada *stripper*. Level LGO yang dikendalikan di dalam *stripper* diharapkan tidak melebihi 45% [3].

Setiap instrumen yang terdapat di perusahaan, tidak akan pernah luput dari permasalahan. LGO *stripper* juga tidak luput dari permasalahan. Berdasarkan sampel grafik proses PT. Pertamina (Persero) RU IV Cilacap (Lampiran C.1) yang diambil pada tanggal 30 Januari 2015 hingga 3 Februari 2015, terjadi fluktuasi level yang terlalu tinggi pada tanggal 30 Januari 2015 pukul 19.50 s.d 21.10 dan terlalu rendah pada tanggal 2 Februari 2015 pukul 13.36. Berdasarkan data OP, pada waktu tertentu, terdapat nilai OP yang menunjukkan nilai nol. Secara keseluruhan dapat dilihat bahwa pada grafik bahwa nilai OP, PV, dan SP memiliki ketidaksempurnaan. Nilai fluktuasi level serta beberapa kali OP yang bernilai nol menunjukkan bahwa terdapat kesalahan skala minor yang terjadi pada sensor dan aktuator. Hal ini dapat disebabkan oleh kesalahan pembacaan sensor level yang mengakibatkan *hunting level* atau kesalahan aktuator yang mengakibatkan bukaan *valve* tidak seperti yang diharapkan. Hal ini didasarkan pada data kalibrasi waktu sesudahnya. Permasalahan ini juga dapat menyebabkan terhambatnya kinerja sistem dalam proses pemurnian *light gas oil* karena diperlukan waktu untuk melakukan *replacing sensor* atau aktuator. Kedua dampak tersebut secara tidak langsung akan menyebabkan kerugian materi pada perusahaan. Oleh karena itu, perlu adanya sistem pengendalian yang mampu menjaga performansi dan keandalan sensor level dan aktuator pada LGO *stripper*.

Sistem pengendalian yang mampu menoleransi kesalahan secara otomatis agar dapat menjaga performansi dan keandalan suatu komponen disebut dengan *fault – tolerant control system* (FTCS). FTCS dapat dibagi menjadi dua tipe yaitu *passive fault tolerant control* (PFTC) dan *active fault tolerant control* (AFTC). PFTC didesain untuk menjadi sinyal pengendali yang *robust* terhadap kesalahan komponen sistem yang telah diestimasi batas toleransi kesalahannya, sedangkan AFTC bekerja dengan

merekonfigurasi ulang aksi pengendalinya ketika terjadi kesalahan komponen sistem [4].

Berdasarkan rujukan penelitian yang telah dilakukan oleh Katherin Indriawati, dkk yang berhasil merancang *active fault – tolerance control* untuk sistem yang linier dengan kesalahan pada aktuator dan sensor, maka pada tugas akhir ini dilakukan perancangan sistem pengendalian yang mampu mengatasi kesalahan pada sensor dan aktuator pada sistem nonlinier dengan judul perancangan *active fault tolerance control* akibat kesalahan sensor dan aktuator pada sistem pengendalian level *light gas oil (LGO) stripper* unit *fuel oil complex* I PT. Pertamina (Persero) RU IV Cilacap.

## 1.2 Perumusan Masalah

Permasalahan yang muncul dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana merancang sistem *observer* akibat kesalahan sensor dan aktuator pada sistem pengendalian level *light gas oil (LGO) stripper* unit *fuel oil complex* I PT. Pertamina (Persero) RU IV Cilacap ?
2. Bagaimana merekonfigurasi sinyal pengendali akibat kesalahan sensor dan aktuator pada sistem pengendalian level *light gas oil (LGO) stripper* unit *fuel oil complex* I PT. Pertamina (Persero) RU IV Cilacap ?
3. Bagaimana hasil perbandingan performansi sistem pengendalian yang menggunakan algoritma PI dengan sistem pengendalian yang menggunakan algoritma PI dan AFTC ?

## 1.3 Batasan Masalah

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

1. Pemodelan *plant* ditinjau pada proses *stripping* di LGO *stripper*.

2. Kesalahan sensor dianalisis pada kesalahan nilai keluaran *output* sensor yang disebabkan oleh kesalahan pembacaan nilai ketinggian *light gas oil* di LGO *stripper*.
3. Kesalahan aktuator dianalisis pada kesalahan presentase bukaan *control valve* yang tidak sesuai dengan besar sinyal elektrik yang diberikan.
4. Besaran yang dikendalikan nilainya adalah nilai level (meter) dan besaran yang dimanipulasi adalah laju aliran keluar dari LGO *heat exchanger* (kg/h) yang masuk ke dalam LGO *drier*.
5. Sampel data proses terdiri dari variabel SP, PV, dan OP yang diambil mulai tanggal 30 Januari 2015 hingga 3 Februari 2015.
6. Algoritma PI digunakan pada sistem pengendalian level LGO *Stripper*.
7. Pemodelan *plant* didapatkan dari *software* HYSYS untuk membantu mendapatkan respon *plant* secara *real*.

#### 1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah sebagai berikut :

1. Merancang sistem *observer* akibat kesalahan sensor dan aktuator pada sistem pengendalian level *light gas oil* (LGO) *stripper* unit *fuel oil complex* I PT. Pertamina (Persero) RU IV Cilacap.
2. Merekonfigurasi sinyal pengendali akibat kesalahan sensor dan aktuator pada sistem pengendalian level *light gas oil* (LGO) *stripper* unit *fuel oil complex* I PT. Pertamina (Persero) RU IV Cilacap.
3. Membandingkan performansi sistem pengendalian yang menggunakan algoritma PI dengan sistem pengendalian yang menggunakan algoritma PI dan AFTC.

#### 1.5 Manfaat

Manfaat tugas akhir ini adalah sebagai berikut :

1. Perancangan *Active Fault Tolerance Control* (AFTC) dapat dijadikan sebagai metode baru pada perancangan suatu

sistem kendali agar menjadi handal dan memiliki dampak kerusakan yang kecil meskipun terjadi suatu kesalahan pada sensor ataupun aktuator.

2. Perancangan AFTC dapat dijadikan sebagai acuan penggunaan *software* untuk sistem kendali, yaitu sebagai pengendali nilai level di LGO *stripper* unit FOC I di PT. Pertamina (Persero) RU IV Cilacap.
3. Penggunaan *software* HYSYS dapat dijadikan sebagai metode baru dalam ranah ilmu pengendali.

## 1.6 Sistematika Laporan

Sistematika laporan tugas akhir diuraikan sebagai berikut :

### BAB I PENDAHULUAN

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat, dan sistematika laporan.

### BAB II DASAR TEORI

Bab ini berisi mengenai teori – teori pada tugas akhir, seperti pengertian, gambaran umum, dan proses *light gas oil* (LGO) pada LGO *stripper*, sistem kendali pada LGO *stripper*, sistem kendali PI, serta *active fault tolerant control* dan perancangannya.

### BAB III METODOLOGI PENELITIAN

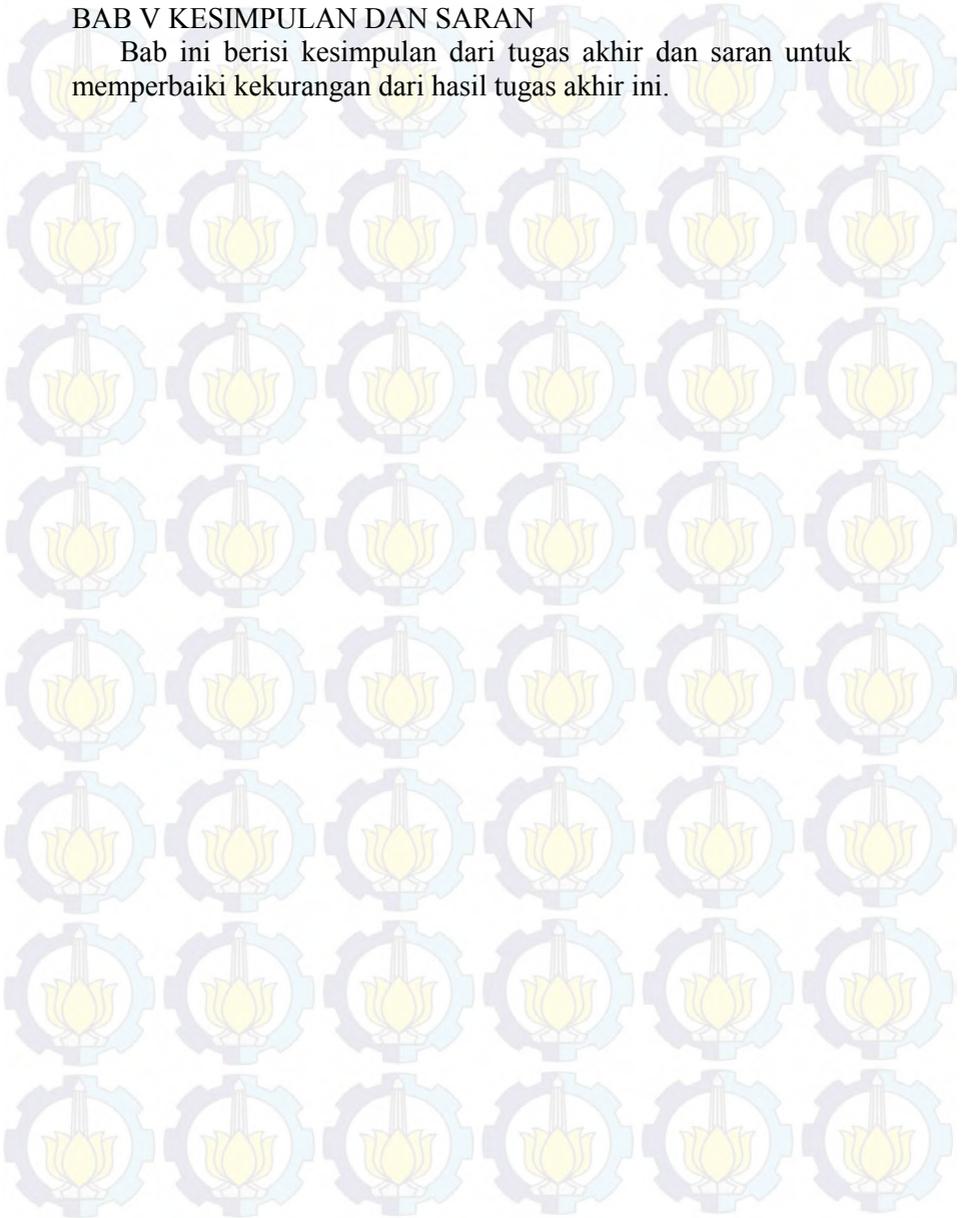
Bab ini terdiri dari pemodelan sistem menggunakan *software* HYSYS untuk mendapatkan grafik hingga bisa dihitung persamaan FOPDT, langkah – langkah perancangan kendali PI, dan langkah – langkah perancangan sistem *observer* sensor dan aktuator serta pengujian dengan simulasi pada program MATLAB.

### BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi mengenai hasil evaluasi uji *lost of effectiveness* dan uji kebocoran untuk aktuator, uji kesalahan bias, uji sensitivitas, dan uji *noise* untuk sensor, serta perbandingan parameter uji performansi anatara sistem pengendalian level dengan algoritma PI dengan AFTC dan sistem pengendalian level dengan algoritma PI tanpa AFTC dan pembahasannya.

## BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari tugas akhir dan saran untuk memperbaiki kekurangan dari hasil tugas akhir ini.

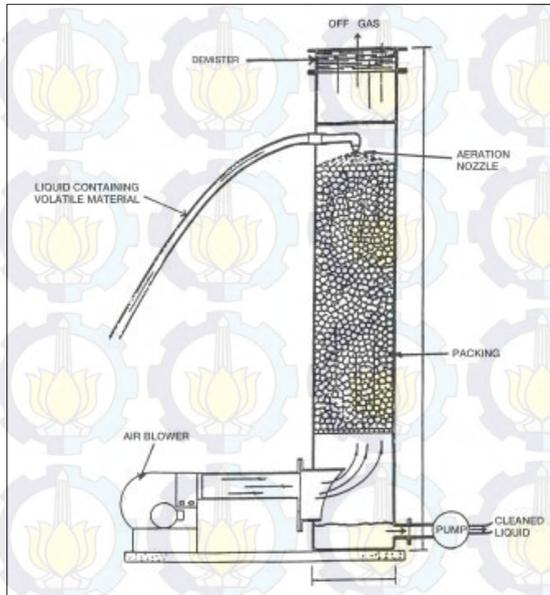


## BAB II TINJAUAN PUSTAKA

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 *Stripping*

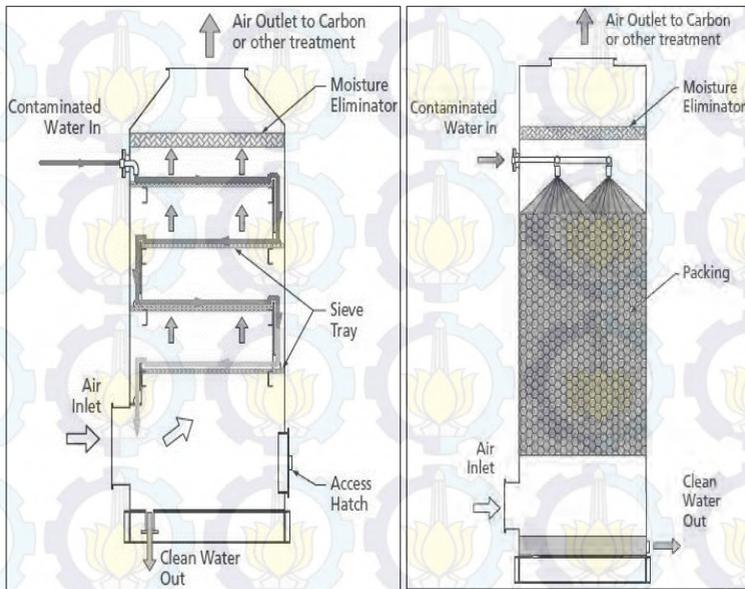
Menurut Ju Chang Huang dan Chii Shang, proses *stripping* adalah suatu proses dimana cairan, (air atau air limbah), dibawa ke dalam kontak dengan gas, (udara), sehingga beberapa zat yang tidak diinginkan terkandung dalam fase cair dapat dilepaskan dan dibawa pergi oleh gas. Gambar 2.1 merupakan gambar menara *stripper* yang menggunakan udara sebagai media *stripping* [6].



**Gambar 2.1** Diagram Menara *Stripper* [6]

## 2.2 Kolom *Stripper*

Skema kolom *stripper* pada gambar 2.2 didesain dengan membedakan tingkat penguapan di antara campuran material organik dengan cairan. Komponen yang lebih mudah menguap akan berpindah dari cairan yang jenuh ke aliran gas. Aliran gas tersebut keluar melalui bagian atas kolom, sedangkan komponen yang mengandung tingkat penguapan paling rendah, akan keluar dalam wujud cairan melalui bagian bawah kolom [7].



**Gambar 2.2** Tray Stripper (kiri) dan Menara Stripper (kanan) [7]

Ketika cairan yang jatuh masuk ke dalam kolom *stripper*, cairan tersebut akan mengalir turun dan mengalami aliran yang berlawanan dengan aliran gas. Aliran gas yang bergerak ke atas meninggalkan kolom *stripper* sebagian akan terdistribusi ke atmosfer dan sebagian lagi akan dikumpulkan untuk dimurnikan kembali guna keperluan lainnya. Kolom *stripper* terdiri dari beberapa *tray* yang terintegrasi menjadi satu kesatuan, bergantung pada penggunaannya.

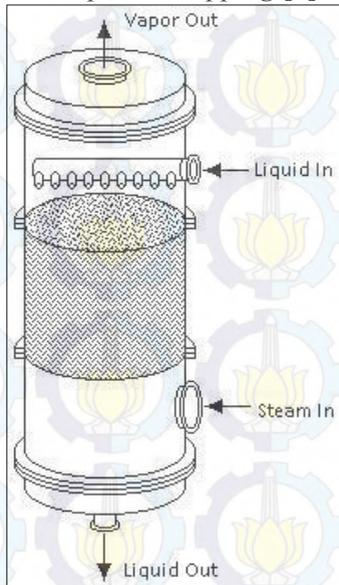
Terdapat beberapa tipe *tray* seperti *sieve*, *valve*, dan *bubble cup*. Bentuk dari *tray* mempengaruhi perluasan area terhadap kontak antara aliran uap dan cairan [7].

### 2.3 Kolom *Steam Stripper*

Proses yang terjadi pada kolom *steam stripper* memiliki kesamaan dengan kolom *stripper* yang telah dijelaskan pada subbab sebelumnya. Kolom *steam stripper* menggunakan *steam* sebagai perantara dalam proses *stripping*nya sebagai pengganti udara yang biasa digunakan.

Penggunaan *steam* memiliki kelebihan dapat memisahkan material organik berfraksi berat dan ringan yang terlarut dalam cairan. Suhu di dalam kolom *steam stripper* ini didesain sangat dekat dengan titik didih dari cairan.

Pada kolom *steam stripper*, umpan cairan masuk melalui bagian atas kolom dan *steam* masuk melalui bagian bawah kolom. Ketika cairan masuk ke dalam kolom *steam stripper*, cairan tersebut akan mengalir ke bawah dan mengalami kontak dengan aliran *steam* yang berlawanan. Pada gambar 2.3 di bawah ini memperlihatkan gambaran kolom *steam stripper*. Kolom *steam stripper* ini dapat meningkatkan jumlah terjadinya kontak antara dua aliran tersebut dalam proses *stripping* [7].



**Gambar 2.3** Kolom *Steam Stripper* [7]

Tekanan operasi pada kolom *steam stripper* dapat mempengaruhi efisiensi dan keandalan dari kolom. Kolom *steam stripper* beroperasi pada keadaan vakum yang membutuhkan biaya besar tetapi memiliki efisiensi yang tinggi. Pada tekanan operasi yang rendah, tingkat penguapan lebih tinggi dan suhu operasi lebih rendah. Kolom beroperasi pada tekanan yang mendekati tekanan atmosfer sehingga dapat menghasilkan tingkat penguapan yang tinggi dan suhu yang rendah tanpa biaya untuk sistem vakum. Pada tekanan operasi yang tinggi, tingkat kelarutan dari pelarut meningkat dan akan mengalami pemisahan.

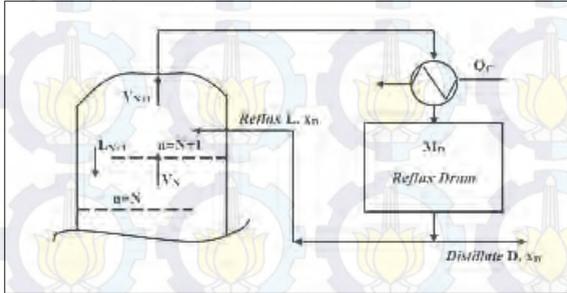
Proses pemanasan sebelum umpan cairan memasuki kolom sangat penting, karena dapat mengurangi jumlah panas yang harus ditambahkan ke dalam kolom untuk mencapai proses pemisahan yang dikehendaki. Umpan mengalami pemanasan dengan menggunakan *heat exchanger* atau sejenisnya [8].

#### **2.4 Model Matematis Kolom Stripper**

Suatu sistem kendali memerlukan peninjauan sebuah model fisis sistem untuk merancang desainnya. Model fisis harus dapat menggambarkan karakteristik dinamis sistem tersebut secara memadai. Model matematis diturunkan dari hukum fisis sistem yang bersangkutan, misal dinamika sistem mekanis dimodelkan dengan menggunakan hukum Newton. Selain itu, sistem modern yang kompleks dan memiliki akurasi yang tinggi (dikarakteristikan dengan MIMO, *time – varying, non – linear, robust*) harus menggunakan pendekatan *state space* yang bersifat domain waktu.

Kolom *stripper* memiliki karakteristik yang sama dengan kolom distilasi, karena model matematis yang digunakan pada kolom *stripper* adalah model matematis dari kolom distilasi. Pemodelan kolom distilasi yang dikemukakan oleh Vu Trieu dan John Pumwa dalam bukunya yang berjudul “*Modelling and Control Simulation for a Condensate Distillation Column*”, dibagi menjadi empat bagian, yaitu bagian *rectifying (top)*, bagian *feed*,

bagian *general*, dan bagian *stripping (bottom)*. Adapun untuk setiap bagian tersebut, terbagi ke dalam tiga buah persamaan, yaitu persamaan kesetimbangan massa, kesetimbangan komponen, dan kesetimbangan energi [9].



**Gambar 2.4** Proses *Rectifying* pada Kolom *Stripper* [9]

Persamaan (2.1), (2.2), dan (2.3) merepresentasikan persamaan yang berlaku untuk bagian *rectifying* pada kolom *stripper*. Ketiga persamaan tersebut didasarkan pada gambar 2.4.

Persamaan kesetimbangan massa :

$$\frac{d(M_{N+1})}{dt} = L + V_N - L_{N+1} - V_{N+1} \quad (2.1)$$

Persamaan kesetimbangan komponen :

$$\frac{d(M_{N+1}x_{N+1})}{dt} = Lx_D + V_Ny_N - L_{N+1}x_{N+1} - V_{N+1}y_{N+1} \quad (2.2)$$

Persamaan kesetimbangan energi :

$$\frac{d(M_{N+1}h_{N+1})}{dt} = h_D L + H_N V_N - h_{N+1} L_{N+1} - H_{N+1} V_{N+1} \quad (2.3)$$

dimana,

- $M_{N+1}$  : total massa yang tertahan pada *tray* N+1
- $L$  : Laju aliran massa cairan *refluks* (kg/s)
- $V_N$  : Laju aliran massa uap (kg/s)

$L_{N+1}$  : Laju aliran massa cairan *refluks* yang turun dari *tray* N+1 (kg/s)

$V_{N+1}$  : Laju aliran massa uap yang naik dari *tray* N+1 (kg/s)

$x_{N+1}$  : Konsentrasi cairan pada *tray* N+1 (M)

$x_D$  : Konsentrasi cairan distilat (M)

$y_N$  : Konsentrasi uap pada *tray* N (M)

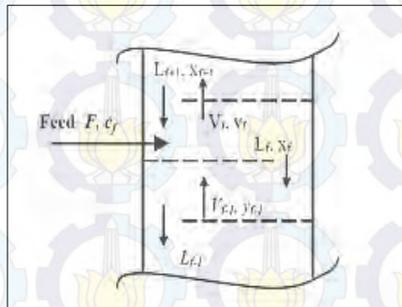
$y_{N+1}$  : Konsentrasi uap pada *tray* N+1 (M)

$h_{N+1}$  : Entalpi cairan pada *tray* N+1 (M)

$h_D$  : Entalpi cairan distilat (Joule/kg)

$H_N$  : Entalpi uap pada *tray* N (Joule/kg)

$H_{N+1}$  : Entalpi uap pada pada *tray* N+1 (Joule/kg)



**Gambar 2.5** Proses *Feed* pada Kolom *Stripper* [9]

Persamaan (2.4), (2.5), dan (2.6) merepresentasikan persamaan yang berlaku untuk bagian *feed* pada kolom *stripper*. Ketiga persamaan tersebut didasarkan pada gambar 2.5.

Persamaan kesetimbangan massa :

$$\frac{d(M_f)}{dt} = F + L_{f+1} + V_{f-1} - L_f - V_f \quad (2.4)$$

Persamaan kesetimbangan komponen :

$$\frac{d(M_f x_f)}{dt} = F x_f + L_{f+1} x_{f+1} + V_{f-1} y_{f-1} - L_f x_f - V_f y_f \quad (2.5)$$

Persamaan kesetimbangan energi :

$$\frac{d(M_f h_f)}{dt} = h_f F + h_{f+1} L_{f+1} + H_{f-1} V_{f-1} - h_f L_f - H_f V_f \quad (2.6)$$

dimana,

$M_f$  : total massa yang tertahan pada tray umpan

$F$  : Laju aliran massa cairan umpan masuk (kg/s)

$V_{f-1}$  : Laju aliran massa uap yang naik dari tray f-1 (kg/s)

$L_{f+1}$  : Laju aliran massa cairan yang turun dari tray f+1 (kg/s)

$L_f$  : Laju aliran massa cairan yang turun dari tray f (kg/s)

$V_f$  : Laju aliran massa uap yang naik dari tray f (kg/s)

$x_f$  : Konsentrasi cairan umpan masuk (M)

$x_{f+1}$  : Konsentrasi cairan pada tray f+1 (M)

$y_f$  : Konsentrasi uap pada tray f (M)

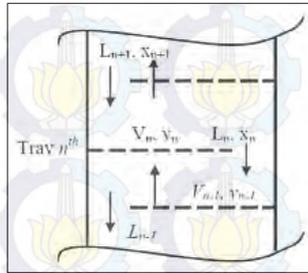
$y_{f-1}$  : Konsentrasi uap pada tray f-1 (M)

$h_f$  : Entalpi cairan umpan masuk (Joule/kg)

$h_{f+1}$  : Entalpi cairan pada tray f+1 (Joule/kg)

$H_f$  : Entalpi uap pada tray f (Joule/kg)

$H_{f-1}$  : Entalpi uap pada tray f-1 (Joule/kg)



**Gambar 2.6** Proses *General* pada Kolom *Stripper* [9]

Persamaan (2.7), (2.8), dan (2.9) merepresentasikan persamaan yang berlaku untuk bagian *general* pada kolom *stripper*. Ketiga persamaan tersebut didasarkan pada gambar 2.6.

Persamaan kesetimbangan massa :

$$\frac{d(M_n)}{dt} = L_{n+1} + V_{n-1} - L_n - V_n \quad (2.7)$$

Persamaan kesetimbangan komponen :

$$\frac{d(M_n x_n)}{dt} = L_{n+1} x_{n+1} + V_{n-1} y_{n-1} - L_n x_n - V_n y_n \quad (2.8)$$

Persamaan kesetimbangan energi :

$$\frac{d(M_n h_n)}{dt} = h_{n+1} L_{n+1} + H_{n-1} V_{n-1} - h_n L_n - H_n V_n \quad (2.9)$$

dimana,

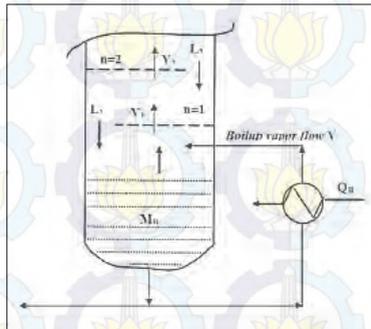
$M_n$  : total massa yang tertahan pada *tray* n

$L_n$  : Laju aliran massa cairan yang turun dari *tray* n (kg/s)

$V_n$  : Laju aliran massa uap yang naik dari *tray* n (kg/s)

$L_{n+1}$  : Laju aliran massa cairan yang turun dari *tray* n+1 (kg/s)

- $V_{n-1}$  : Laju aliran massa uap yang naik dari *tray* n-1 (kg/s)  
 $x_n$  : Konsentrasi cairan pada komponen ringan pada *tray* n (M)  
 $x_{n+1}$  : Konsentrasi cairan pada *tray* n+1 (M)  
 $y_n$  : Konsentrasi uap pada *tray* n (M)  
 $y_{n-1}$  : Konsentrasi uap pada *tray* n-1 (M)  
 $h_n$  : Entalpi cairan pada *tray* n (Joule/kg)  
 $h_{n+1}$  : Entalpi cairan pada *tray* n+1 (Joule/kg)  
 $H_n$  : Entalpi uap pada *tray* n (Joule/kg)  
 $H_{n-1}$  : Entalpi uap pada pada *tray* n-1 (Joule/kg)



**Gambar 2.7** Proses *Stripping* pada Kolom *Stripper* [9]

Persamaan (2.10), (2.11), dan (2.12) merepresentasikan persamaan yang berlaku untuk bagian *stripping* pada kolom *stripper*. Ketiga persamaan tersebut didasarkan pada gambar 2.7.

Persamaan kesetimbangan massa :

$$\frac{d(M_B)}{dt} = L_2 - V_B - B \quad (2.10)$$

Persamaan kesetimbangan komponen :

$$\frac{d(M_B x_B)}{dt} = L_2 x_2 - V_B y_B - B x_B \quad (2.11)$$

Persamaan kesetimbangan energi :

$$\frac{d(M_B h_B)}{dt} = h_2 L_2 + Q_B - h_B B - H_B V_B \quad (2.12)$$

dimana,

$M_B$  : total massa yang tertahan pada ruang level

$L_2$  : Laju aliran massa cairan yang turun dari *tray* ke – 2 (kg/s)

$V_B$  : Laju aliran massa uap yang naik dari ruang level (kg/s)

$B$  : Laju aliran cairan bawah yang keluar dari *stripper* (kg/s)

$x_B$  : Konsentrasi cairan bawah yang keluar dari *stripper* (M)

$x_2$  : Konsentrasi cairan pada *tray* ke – 2 (M)

$y_B$  : Konsentrasi uap bagian bawah (M)

$h_B$  : Entalpi cairan bawah yang keluar dari *stripper* (Joule/kg)

$h_2$  : Entalpi cairan pada *tray* ke – 2 (Joule/kg)

$H_B$  : Entalpi uap bagian bawah (Joule/kg)

$Q_B$  : Kalor yang diberikan oleh *reboiler* (Joule/kg)

## 2.5 Light Gas Oil (LGO)

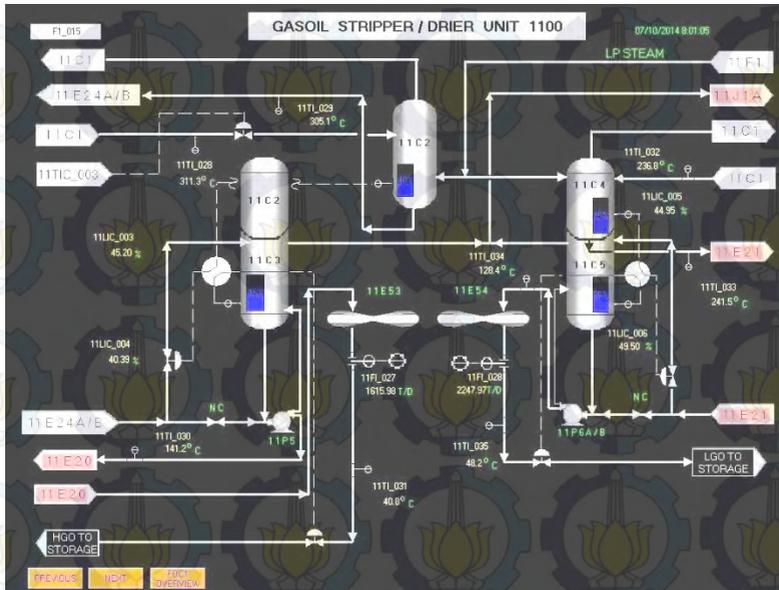
LGO berwujud cairan yang terdiri dari beberapa komponen, seperti *light vacuum gas oil*, *biphenyl*, *saturated hydrocarbons and naphthenes*, *aromatic hydrocarbons*, dan *polynuclear aromatic hydrocarbons*. LGO memiliki karakteristik transparan dengan *flash point* sebesar 112°C – 121°C (dengan metode *closed cup ASTM D – 93*), *specific gravity* sebesar 0,88 – 0,90 pada suhu

15<sup>0</sup>C, kisaran titik beku sebesar 1<sup>0</sup>C – 7<sup>0</sup>C, kisaran titik didih sebesar 230<sup>0</sup>C – 413<sup>0</sup>C, tekanan uap sebesar 0,1 – 0,4 psia pada suhu 38<sup>0</sup>C dan rata – rata densitas sebesar 7,411 lbs/gal pada suhu 15<sup>0</sup>C [10].

## 2.6 *Light Gas Oil (LGO) Stripper*

LGO *stripper* memiliki struktur kolom yang tergabung dengan *drier* yang berada di bawahnya. Umpan LGO *stripper* ini berupa cairan LGO yang berasal dari *tray* ke – 20 *crude splitter column* 11C1. Pada LGO *stripper*, fraksi ringan dipisahkan dari LGO dengan bantuan umpan *low pressure steam*. *Steam* dan LGO saling mengalami kontak satu sama lain dengan tujuan untuk memisahkan fraksi ringan yang tidak diharapkan terkandung dalam LGO [11].

Pada *tray* paling atas, sebagian *steam* yang mengalami kontak dengan LGO ikut bercampur dengan aliran *top product* LGO *stripper* yang berwujud gas dan dikembalikan ke *tray* ke – 24 *crude splitter column* 11C1. Pada *tray* ke – 24 *crude splitter column* 11C1, gas fraksi ringan dan *steam* yang bercampur dipisahkan dengan *ejector steam* menggunakan prinsip sistem vakum. Sebagian *steam* lainnya yang berada di LGO *stripper* mengembun dan bercampur dengan LGO yang berada di bawah. Oleh karena itu, diperlukan *drier* untuk menghilangkan kondensat *steam* yang tercampur dalam LGO akibat proses *stripping* sebelumnya sehingga dihasilkan LGO murni [11].



**Gambar 2.8** Proses LGO Stripper [11]

Proses LGO *stripper* ditunjukkan pada gambar 2.8 di atas. Aliran LGO bercampur dengan kondensat *steam* yang mengalir melalui bagian bawah LGO *stripper* dan menuju ke LGO *heat exchanger* 11E21 pada suhu  $273^{\circ}\text{C}$  bertujuan untuk menukar panas LGO dengan *crude* yang dialirkan oleh *crude feed pumps*. Selanjutnya, LGO yang telah ditukar panasnya, mengalir kembali masuk ke dalam LGO *drier* 11C5. Pada LGO *drier* terjadi proses pengeringan dengan sistem vakum. *Steam* yang masih terkandung di dalam LGO ditarik oleh *ejector steam*. LGO yang telah hilang kandungan *steam*nya dipompa ke *light gas oil air cooler* untuk didinginkan pada suhu  $135,3^{\circ}\text{C}$ , kemudian dialirkan ke *storage*, sedangkan, untuk *steam* yang dipisahkan terangkut ke atas bersama dengan uap air menuju ke *vapour vacuumset* [11].

## 2.7 Sistem Pengendalian Level LGO Stripper

Pada LGO *stripper*, terdapat sistem pengendalian level pada *hold up* cairan yang ada di dasar *stripper*. Level LGO ini dijaga agar tetap konstan sesuai *set point*. *Piping and instrument diagram*, menunjukkan bahwa sistem pengendalian level LGO *stripper* termasuk ke dalam jenis *single loop control*, dimana terdiri dari satu *transmitter*, satu *controller*, dan satu *actuator*. Sistem pengendalian level pada LGO *stripper* menggunakan pengendali *proportional integral* (PI) yang terintegrasi dalam *distributed control system* (DCS). Pada *control room*, *human machine interface* (HMI) memberikan *set point* level LGO *stripper* hasil pengukuran dari sensor *displacer*. Agar proses *stripping* sempurna, maka dilakukan aksi pengendali tidak langsung (*indirect control*) dengan mengatur laju aliran LGO yang berasal dari LGO *heat exchanger* menuju ke *drier* untuk menjaga ketinggian cairan yang terdapat di dasar *stripper* pada *set point* 45%. Jika level yang terukur melebihi *set point*, maka laju aliran *light gas oil* ke *drier* akan dibesarkan, dan sebaliknya [11].

## 2.8 Pengendali *Proportional Integral Derivative* (PID)

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*) [12].

- **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 ( $K_p$ ) [12].

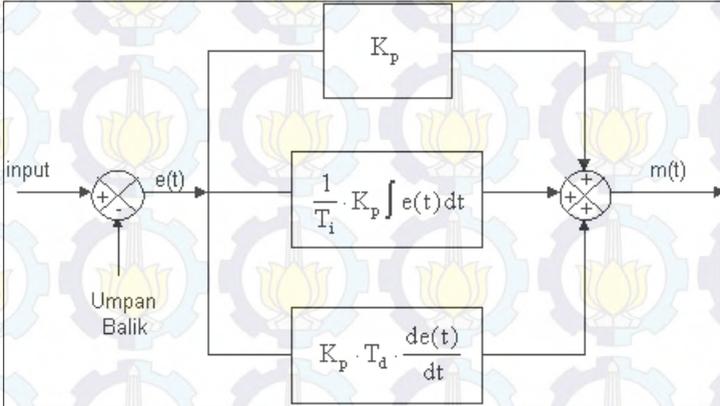
- **Pengendali Integral**

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 [12].

- **Pengendali Differential**

Keluaran pengendali *differential* memiliki sifat seperti suatu operasi derivatif. Perubahan yang mendadak pada masukan pengendali, mengakibatkan perubahan yang sangat besar dan cepat. Setiap kekurangan dan kelebihan dari pengendali P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi pengendali PID. Elemen pengendali P, I dan D secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Gambar 2.9 merupakan diagram blok pengendali PID yang dipasang secara paralel. Karakteristik pengendali PID sangat dipengaruhi oleh

kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta  $K_p$ ,  $T_i$ , dan  $T_d$  mengakibatkan penonjolan sifat dari setiap elemen [12].



**Gambar 2.9** Diagram Blok PID [12]

## 2.9 Perancangan Pengendali PI

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.

## 2.10 Sensor Level

Setiap alat instrumen yang digunakan untuk mengukur dan menunjukkan ketinggian permukaan cairan disebut sebagai alat ukur level, baik pada tangki tertutup maupun terbuka. Tujuan dari pengukuran ketinggian permukaan cairan ini adalah untuk mencegah kerusakan *equipment* akibat kekosongan level serta kerugian akibat cairan terbuang, mengendalikan jalannya proses, dan mendapatkan kualitas produk yang diinginkan [13].

Dalam pengukuran ketinggian permukaan cairan pada umumnya dilakukan dengan dua metode, yaitu metode pengukuran langsung dan metode mekanik. Metode mekanik memanfaatkan gaya pada cairan yang menghasikan gerak mekanik. Pergerakan mekanik ini kemudian dikalibrasi kedalam bentuk skala.

Dilihat dari lokasi keterpasangan dan elemen utamanya, alat ukur level dapat diklasifikasikan sebagai berikut.

- Gelas Penduga (*Level Glass*)

Gelas penduga dapat menunjukkan ketinggian permukaan cairan dalam suatu bejana secara tertutup. Prinsip kerja yang digunakan gelas penduga adalah prinsip bejana berhubungan. Terdapat dua jenis gelas penduga, yaitu ujung terbuka dan ujung tertutup. Gelas penduga ujung terbuka digunakan pada tangki yang tidak bertekanan, yang tingginya melebihi 1,5 meter seperti tangki penampungan minyak diesel, sedangkan gelas penduga ujung tertutup digunakan pada *vessel* yang bertekanan seperti *boiler* [13].

- Pemberat atau Pita

Cara termudah untuk mengukur ketinggian permukaan cairan dalam tangki adalah dengan menggunakan sebuah pita pengukur yang diberi bobot pemberat. Bobot pemberat diturunkan kedalam tangki dan tinggi permukaan cairan dapat dilihat secara langsung pada pita pengukuran. Sistem pengukuran ini sering dilakukan pada tangki yang mengandung cairan lengket atau memberi bekas pada pengukuran [13].

- Penggeser (*Displacer*)

*Displacer* memiliki prinsip kerja menghasilkan nilai gerak yang menggantikan nilai volume cairan yang menghasilkan gerak tersebut [13].

- Metode Gelembung

Meteran alat ukur pada metode gelembung umumnya adalah *pressure gauge* atau tabung *Bourdon* yang telah dikalibrasikan sebelumnya dalam skala persen. Alat ukur jenis ini digunakan

pada tangki air yang tidak bertekanan (statis). Metode gelembung ini menggunakan udara yang bertekanan konstan, maksimum 50 Psi. Udara dimasukkan kedalam tabung yang terbenam (tegak) pada cairan yang diukur, semakin tinggi permukaan cairan semakin besar tekanan udara yang dibutuhkan mengatasi tekanan statis yang diberikan oleh cairan [13].

- Metode Beda Tekanan

Sama seperti *displacer*, diafragma dan penghembus dihubungkan dengan *transmitter*, baik pneumatik maupun elektrik. Kemudian, tekanan sinyal pneumatik atau sinyal elektrik diteruskan ke meteran penunjuk yang telah dikalibrasi sebelumnya [13].

### 2.11 Control Valve

*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 :

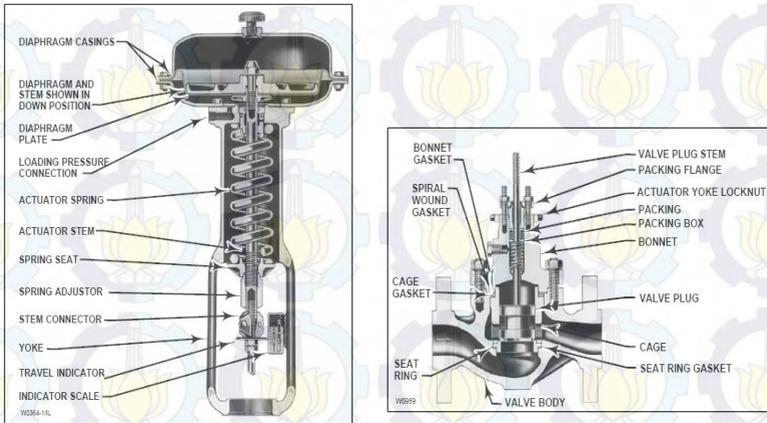
- *Air to Open* (ATO)

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* [14].



**Gambar 2.10** Bentuk Umum *Control Valve* [14]

Ada beberapa jenis tipe *control valve*, seperti *globe valve*, *ball valve*, *butterfly valve*, dan *gate valve*. Gambar 2.10 di atas menggambarkan bentuk umum *control valve* [14].

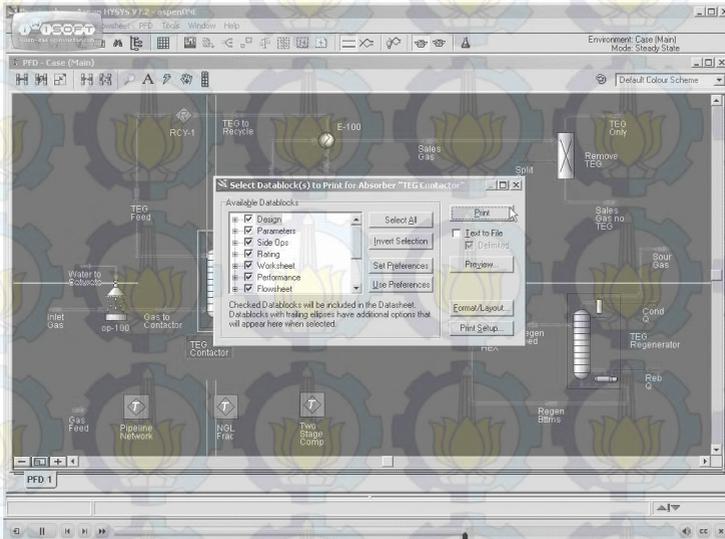
## 2.12 Software HYSYS

Hysys merupakan *software process engineering* yang digunakan untuk mensimulasikan suatu unit proses atau multi unit proses yang terintegrasi, *intuitive*, *iterative*, *open*, dan *extensible*. Simulator Hysys bermanfaat untuk aplikasi di industri seperti perancangan suatu industri, memonitor kemampuan dari industri yang telah *exist*, melacak permasalahan proses yang terjadi di industry, dan dapat meningkatkan kemungkinan kapasitas produksi dari *plant*. Area penggunaan dari simulator Hysys adalah sebagai berikut :

- *Conceptual analysis*,
- *Process design*,
- *Project design*,
- *Operability and safety*,
- *Automation*,

- *Asset utilization.*

Gambar berikut ini merupakan tampilan muka dari Aspen HYSYS.



**Gambar 2.11** Tampilan Muka Aspen HYSYS [15]

### 2.13 FOPDT(*First Order Plus Dead Time*)

FOPDT adalah sebuah sistem orde satu yang memiliki *dead time*. *Dead time* sendiri merupakan waktu antara *input* yang diberikan sampai menghasilkan respon pada keluaran. FOPDT telah banyak digunakan untuk memodelkan sistem dalam rangka penyederhanaan dari sistem yang berorde tinggi menjadi sistem yang berorde satu yang memiliki *dead time*. Bentuk model seperti ini dianggap telah cukup untuk menggambarkan kedinamisan dari sistem yang berorde tinggi. Persamaan FOPDT tampak dalam persamaan (2.13) dan (2.14) berikut ini [16].

$$\frac{Y(s)}{U(s)} = \frac{K_p e^{-\theta_p s}}{s}, \quad (2.13)$$

$$\frac{dy(t)}{dt} = K_p u(t - \theta_p) \quad (2.14)$$

dimana,  $K_p$  adalah *gain integrator* dan  $\theta_p$  adalah *dead time*. Dari fungsi transfer tersebut, dapat disimpulkan bahwa sistem yang berorde tinggi dapat digantikan dengan sistem berorde satu ditambah dengan *dead time*. Jadi terdapat tiga buah parameter yang perlu diketahui untuk mengidentifikasi sistem berorde tinggi yaitu *dead time*, *time constant*, dan *gain*. Terdapat beberapa metode untuk mendapatkan ketiga parameter tersebut, diantaranya adalah dengan menggunakan metode *open loop* atau *closed loop* [16].

Metode yang digunakan saat ini adalah *two points based method* yang termasuk ke dalam metode *open loop*. Pada *two points based method*, yang pertama kali harus dilakukan adalah mencari nilai waktu yang dibutuhkan respon untuk mencapai dua titik yaitu 63% dan 28% dari nilai *steady state* [16].

Bentuk persamaan secara sederhananya dapat dilihat pada persamaan berikut ini.

$$G(s) = \frac{K}{\tau s + 1} e^{-\theta s} \quad (2.15)$$

dimana,

$K$  : *gain steady plant*

$\tau$  :  $1,5((63\% * K) - (28\% * K))$

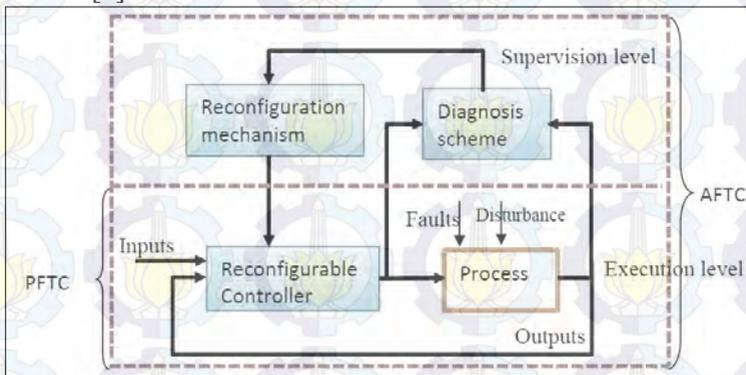
#### 2.14 Fault Tolerant Control

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) [4].

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.

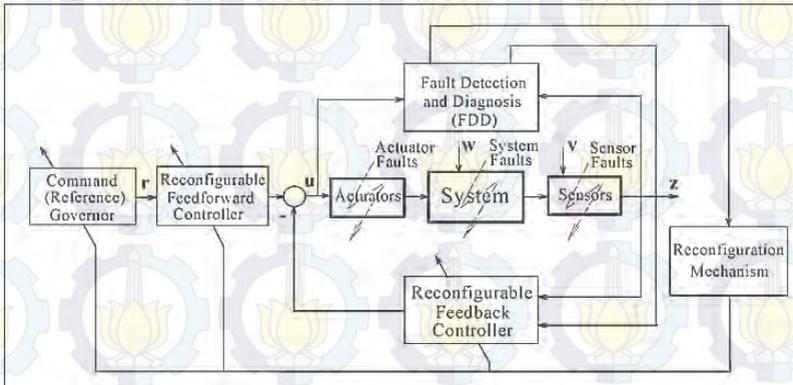
Gambar 2.12 merupakan gambaran skema FTCS. Tujuan utama dari FTCS adalah merancang kontroler dengan struktur yang cocok untuk mencapai kestabilan dan kinerja yang diinginkan. Tidak hanya ketika pengendalian komponen berfungsi secara normal tetapi ketika terjadi kesalahan pada suatu komponen tersebut [4].



**Gambar 2.12** Skema Sistem FTC [17]

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 [4].



**Gambar 2.13** Struktur Umum AFTCS [4]

Pada gambar 2.13 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*.

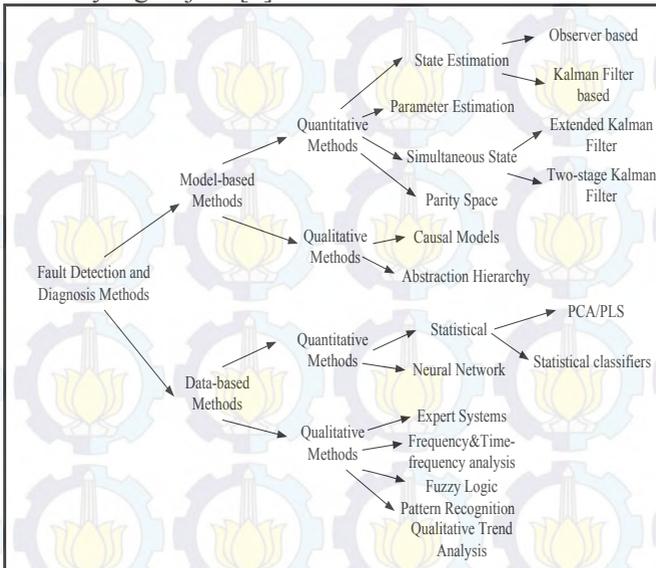
Rekonfigurasi pengendali harus dirancang secara otomatis untuk menjaga stabilitas dan performansi yang diharapkan. Untuk memastikan bahwa sistem tersebut *closed loop*, maka rekonfigurasi pengendali harus dirancang secara *feed forward*. Hal ini dilakukan untuk menghindari terjadinya penurunan performansi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

Pada intinya, perancangan sistem yang penting dalam AFTC terdiri atas dua sub sistem yaitu FDI dan mekanisme RC. FDI berfungsi sebagai pendeteksi jenis kesalahan dan besarnya kesalahan tersebut, sedangkan mekanisme RC merupakan

pengubahan aksi pengendali untuk menghilangkan kesalahan yang terjadi [4].

- **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 [4].



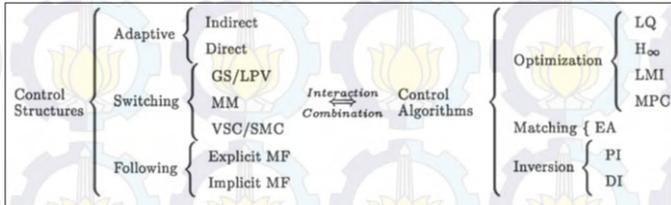
**Gambar 2.14** Klasifikasi Metode FDI [4]

Pada gambar 2.14 di atas metode FDI diklasifikasikan menjadi dua bagian yaitu *model based methods* dan *data based methods*. Kedua metode tersebut dibagi kedalam pendekatan kuantitatif dan kualitatif.

- **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 2.15 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* [5].



**Gambar 2.15** Rekonfigurasi Pengendali pada AFTC [5]

### 2.15 Observer State

*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 [17].

$$\dot{x} = Ax + Bu \quad (2.16)$$

$$y = Cx \quad (2.17)$$

*Descriptor observer* sama seperti model matematis pada sistem atau *plant* yang telah diubah ke dalam bentuk persamaan *state space* dengan adanya penambahan *gain observer* dan menjadi persamaan sebagai berikut [17].

$$\dot{\bar{x}} = A\bar{x} + Bu + K(y - \bar{y}) \quad (2.18)$$

$$\bar{y} = C\bar{x} \quad (2.19)$$

Persamaan *descriptor system* yang disubstitusikan pada persamaan *descriptor observer* menghasilkan persamaan (2.20) berikut ini. Persamaan ini didapatkan dengan memasukkan  $y$  di persamaan (2.17) dan  $\bar{y}$  di persamaan (2.19) ke dalam persamaan (2.18), sehingga akan menjadi,

$$\dot{\bar{x}} = A\bar{x} + Bu + K(Cx - C\bar{x}) \quad (2.20)$$

atau

$$\dot{\bar{x}} = (A - KC)\bar{x} + Bu + KCx \quad (2.21)$$

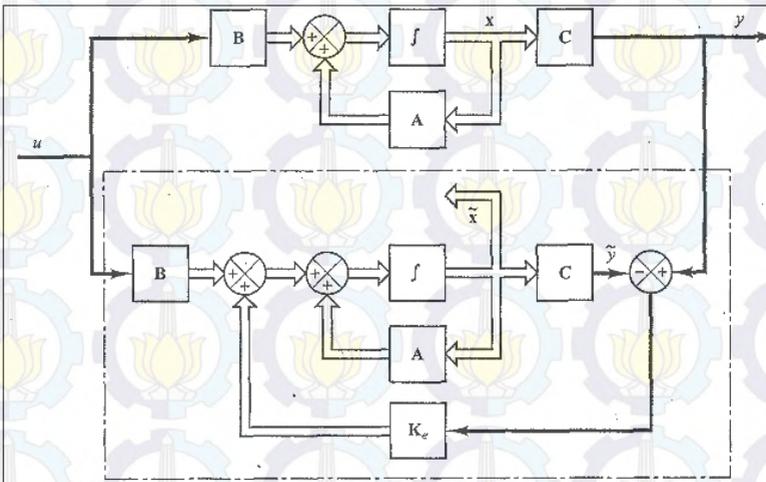
Jika model matematis telah sempurna, maka  $x$  estimasi akan sama dengan  $x$  sistem. Akan tetapi, seringkali terjadi kesalahan pada model, sehingga akan menimbulkan *error* atau perbedaan antara  $x$  sistem dan  $x$  estimasi yang dapat dirumuskan sebagai berikut. Untuk menentukan persamaan *observer error*, persamaan (2.16) dikurangi dengan persamaan (2.20) sebagai berikut. Pada gambar 2.16 terlihat susunan blok diagram *observer*.

$$\dot{x} - \dot{\bar{x}} = Ax - A\bar{x} - K(Cx - C\bar{x}) \quad (2.22)$$

$$\dot{x} - \dot{\bar{x}} = (A - KC)(x - \bar{x}) \quad (2.23)$$

$$e = x - \bar{x} \quad (2.24)$$

$$\dot{e} = (A - KC)e \quad (2.25)$$



Gambar 2.16 Blok Diagram *Observer* [4]

### 2.16 Linear Matrix Inequality (LMI)

Menurut sejarahnya, LMI muncul pada tahun 1890 saat Lyapunov menunjukkan persamaan diferensial berikut ini [18].

$$\frac{d}{dt} x(t) = Ax(t) \quad (2.26)$$

Persamaan diferensial stabil jika dan hanya jika terdapat sebuah solusi untuk pertidaksamaan matriks yang linier dengan matriks P yang diketahui.

$$A^T P + PA < 0 \quad (2.27)$$

$$P = P^T > 0 \quad (2.28)$$

Terdapat berbagai metode untuk menguji kestabilan suatu sistem. Salah satu metode yang digunakan untuk menguji kestabilan sistem linier maupun non linier adalah kestabilan Lyapunov. Berdasarkan analisa kestabilan Lyapunov, kestabilan didasarkan pada energi yang disimpan oleh sistem. Sistem dapat dikatakan stabil jika energi yang disimpan berkurang terhadap penambahan waktu, sehingga energi akan mencapai pada nilai minimal pada titik setimbang sistem [18].

Fungsi Lyapunov adalah fungsi energi buatan yang tergantung pada vektor state sistem dan dapat dinyatakan dengan  $V(x)$ . Dalam metode kedua analisa kestabilan Lyapunov,  $V(x)$  dan turunan pertamanya dapat memberikan informasi kestabilan sistem. Jika diasumsikan suatu sistem dengan persamaan seperti berikut ini,

$$\dot{x} = f(x, u) \quad (2.29)$$

Sistem stabil pada titik kesetimbangan di origin jika terdapat fungsi skalar  $V(x)$  yang kontinu dan turunan pertama yang memenuhi kondisi definite positif yang ditunjukkan dengan  $V(x) > 0$  dan definite negatif yang ditunjukkan dengan  $\dot{V}(x) < 0$ .

Ada tiga buah faktor yang membuat teknik LMI menarik, diantaranya adalah sebagai berikut :

- spesifikasi desain yang beraneka ragam dan fungsi kendala dapat diekspresikan sebagai PML,
- permasalahan yang diformulasikan ke dalam LMI dapat dipecahkan secara tepat dengan algoritma optimasi *convex*,
- banyak permasalahan yang memiliki banyak fungsi kendala atau tujuan kurang solusi analitisnya dalam permasalahan

matriks, dan dengan menggunakan LMI ini, hal tersebut dapat dengan mudah dikerjakan.

Inilah yang membuat LMI digunakan sebagai dasar dalam mendesain metode alternatif analitis klasik. Menurut Boyd, LMI memiliki bentuk persamaan sebagai berikut.

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.30)$$

dimana  $x \in R^m$  merupakan variabel dan matriks simetris.  $F_i = F_i^T \in R^{n \times n}$  dengan  $i=0, \dots, m$ . Tanda pertidaksamaan memiliki arti definit positif, sehingga untuk semua nilai *eigen value* adalah positif.

Telah dibahas bahwa LMI yang *non strict*, memiliki bentuk sebagai berikut.

$$F(x) \geq 0 \quad (2.31)$$

LMI pada persamaan (2.30) adalah sebuah *convex* dengan fungsi kendala  $x$ . Kelipatan LMI  $F^{(1)}(x) \geq 0, \dots, F^{(p)}(x) > 0$  dapat dituliskan sebagai LMI tunggal  $\text{diag}(F^{(1)}(x) > 0, \dots, F^{(p)}(x) > 0)$ . Karena itu tidak ada perbedaan antara satu set LMI dan LMI tunggal. LMI  $F^{(1)}(x) \geq 0, \dots, F^{(p)}(x) > 0$  akan berarti  $\text{diag}(F^{(1)}(x) > 0, \dots, F^{(p)}(x) > 0)$  saat matriks  $F_i$  diagonal LMI  $F(x) \geq 0$  hanya merupakan satu set dari pertidaksamaan linier.

Pertidaksamaan non linier diubah ke dalam bentuk LMI dengan menggunakan komplemen Schur. Ide dasarnya adalah sebagai berikut.

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \quad (2.32)$$

dimana,  $Q(x) = Q(x)^T$ ,  $R(x) = R(x)^T$  dan  $S(x)$  tergantung *affline* pada  $x$ , persamaannya menjadi seperti persamaan (2.33) di bawah ini.

$$R(x) > 0, Q(x) - S(x)R(x)^{-1}S(x)^T > 0 \quad (2.33)$$

Pertidaksamaan non linier pada persamaan (2.33) dapat direpresentasikan sebagai LMI.



*Halaman ini sengaja dikosongkan*

## BAB III METODOLOGI PENELITIAN

### 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 *Active Fault Tolerance Control* (AFTC).

#### a. Pengambilan data *plant*

Pengambilan data *plant* meliputi pengambilan data spesifikasi sensor (sensor *displacer*) dan aktuator (*control valve*), spesifikasi *plant* LGO *stripper*, *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 *displacer* dan *control valve* yang dikaitkan dengan data proses dari DCS untuk menunjukkan *trend* saat terjadi kesalahan pada sensor *displacer* dan *control valve* serta pada saat kondisi normal.

#### b. Pemodelan *plant*, sensor, dan aktuator

Pembuatan model *plant* LGO *stripper* menggunakan *software* Aspen HYSYS 7.1 dengan metode FOPDT (*first order plus dead time*). Pemodelan dilakukan untuk mengidentifikasi perilaku dari unit operasi LGO *stripper* tersebut, sehingga fungsi transfer yang diperoleh benar – benar merepresentasikan *real plant*, sedangkan untuk pembuatan model sensor dan 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 Active Fault Tolerant Control (AFTC)**

Perancangan AFTC dilakukan dengan merancang *observer* bagi sensor dan juga aktuator. Perancangan *observer* sensor dan *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 pada sensor dan aktuator dengan mengubah perhitungan sinyal pengendalinya.

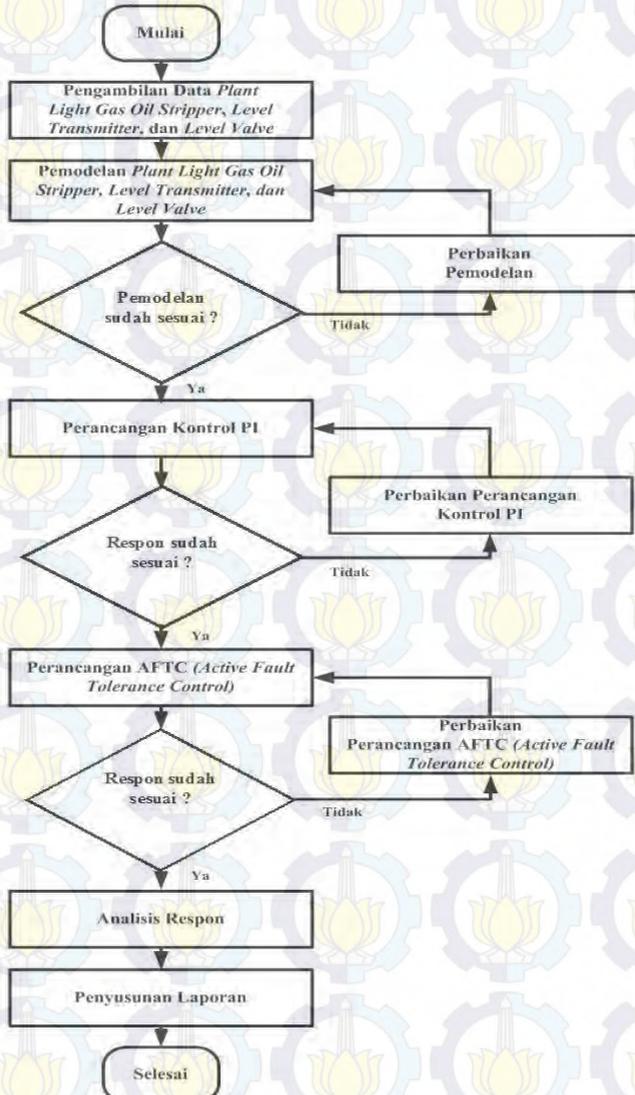
**f. Analisis respon**

Analisis respon didasarkan pada pemberian uji performansi sistem dengan memberikan kesalahan pada sensor dan aktuator berupa kesalahan bias, sensitivitas, *noise*, *lost of effectiveness* dan kebocoran. Setiap pemberian jenis uji tersebut, dilakukan perbandingan antara performansi sistem yang menggunakan pengendali PI dengan algoritma AFTC 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 LGO Stripper

Data – data yang dibutuhkan untuk pemodelan *plant* LGO *stripper* terdapat pada tabel 3.1 di bawah ini. Data tersebut diambil dari unit *process engineering* (PE) PT. Pertamina (Persero) RU IV Cilacap yang dituliskan pada lembar *process flow diagram* (PFD) *plant* dan spesifikasi *plant* yang tampak pada tabel di bawah ini.

**Tabel 3.1** Rincian Data *Plant* [19, 20]

Notasi	Keterangan	Nilai dan Satuan
$V_{total}$	Volume total	9,6 m <sup>3</sup>
$V_{liquid}$	Volume normal <i>liquid</i>	2,1 m <sup>3</sup>
$V_{control}$	Volume level yang dipengendali	3,2 m <sup>3</sup>
$D$	Diameter	1,4 m
$A$	Luas melintang kolom	1,5386 m <sup>2</sup>
$T$	Tinggi	5,92 m
$P$	Tekanan operasi	1,42 kg/cm <sup>2</sup>
$W_{vapour}$	Berat molekul uap	98 wt%
$L_{tray}$	Jarak per <i>tray</i>	0,6 m
$T_{vapour}$	Temperatur atas (Temperatur uap)	247°C
$T_{LGO}$	Temperatur dasar (Temperatur LGO keluar)	237°C
$T_{refluks}$	Temperatur LGO masuk	253°C
$T_{HEout}$	Temperatur LGO setelah keluar dari <i>heat exchanger</i>	135°C
$T_{crude}$	Temperatur <i>crude</i> pendingin	30°C
$T_{steam}$	Temperatur <i>steam</i>	350°C

$P_{vapour}$	Tekanan atas (Tekanan uap)	1,5 kg/cm <sup>2</sup>
$P_{LGO}$	Tekanan dasar (Tekanan LGO keluar)	1,6 kg/cm <sup>2</sup>
$P_{refluks}$	Tekanan LGO masuk	1,42 kg/cm <sup>2</sup>
$P_{HEout}$	Tekanan LGO setelah keluar dari <i>heat exchanger</i>	1,6 kg/cm <sup>2</sup>
$P_{crude}$	Tekanan <i>crude</i> pendingin	30,6 kg/cm <sup>2</sup>
$P_{steam}$	Tekanan <i>steam</i>	2,5 kg/cm <sup>2</sup>
$\dot{m}_{vapour}$	<i>Mass flowrate</i> uap	3,852 kg/s
$\dot{m}_{LGO}$	<i>Mass flowrate</i> LGO keluar	15,740 kg/s
$\dot{m}_{refluks}$	<i>Mass flowrate</i> LGO masuk	19,317 kg/s
$\dot{m}_{HEout}$	<i>Mass flowrate</i> LGO setelah keluar dari <i>heat exchanger</i>	15,63 kg/s
$\dot{m}_{crude}$	<i>Mass flowrate crude</i> pendingin	186,64 kg/s
$\dot{m}_{steam}$	<i>Mass flowrate steam</i>	0,239 kg/s
$\rho_{vapour}$	Massa jenis uap	6,61 kg/m <sup>3</sup>
$\rho_{LGO}$	Massa jenis LGO keluar	668 kg/m <sup>3</sup>
$\rho_{refluks}$	Massa jenis LGO masuk	644 kg/m <sup>3</sup>
$\rho_{HEout}$	Massa jenis LGO setelah keluar dari <i>heat exchanger</i>	745 kg/m <sup>3</sup>
$\rho_{crude}$	Massa jenis <i>crude</i> pendingin	852 kg/m <sup>3</sup>
$\rho_{steam}$	Massa jenis <i>steam</i>	0,85 kg/m <sup>3</sup>
$C_{LGO}$	<i>Specific heat</i> LGO	2,3 kJ/kgK
$h_{steam}$	Entalpi <i>steam</i>	3171,05 kJ/kg
$C_{steam}$	<i>Specific heat steam</i>	2,06131 kJ/kgK



Pada LGO *stripper* terdapat sistem pengendalian level seperti tampak pada gambar di atas. Kolom LGO *stripper* di atas jika diasumsikan bahwa pemodelan matematisnya mengacu pada pemodelan kolom distilasi dikarenakan prinsip kerjanya yang sama, maka dapat digunakan keempat buah persamaan kesetimbangan massa di bawah ini untuk masing – masing jenis *tray*. Hanya digunakan hukum kesetimbangan massa dikarenakan variabel yang dikendalikan adalah level sehingga hanya memerlukan fokus terhadap perpindahan massanya saja [13].

a. *Mass balance* untuk *Tray Rectifying* (atas)

$$\frac{d(M_{N+1})}{dt} = L + V_N - L_{N+1} - V_{N+1} \quad (3.1)$$

b. *Mass balance* untuk *Tray Feed*

$$\frac{d(M_f)}{dt} = F + L_{f+1} + V_{f-1} - L_f - V_f \quad (3.2)$$

c. *Mass balance* untuk *Tray Umum*

$$\frac{d(M_N)}{dt} = L_{N+1} - L_N + V_{N-1} - V_N \quad (3.3)$$

d. *Mass balance* untuk *Level Hold Up* (bawah)

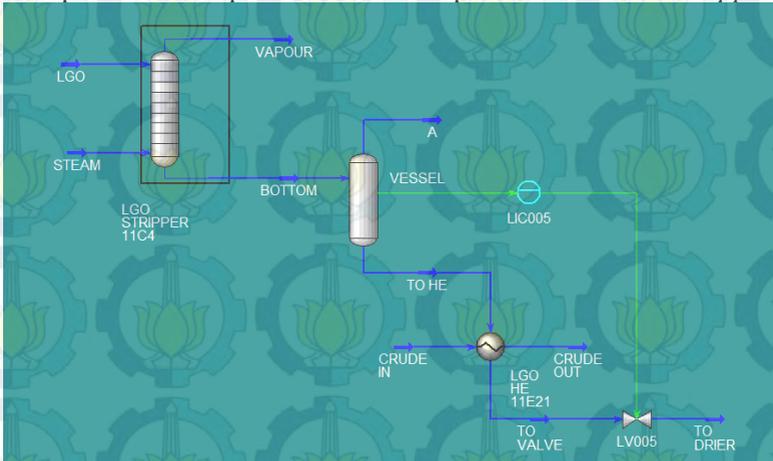
$$\frac{d(M_B)}{dt} = L_1 - V_B - B \quad (3.4)$$

Penurunan model matematis untuk kolom LGO *stripper* berdasarkan hukum kesetimbangan massa masing – masing *tray* di atas banyak menggunakan asumsi – asumsi sehingga akan menimbulkan peluang besar untuk ketidakcocokan saat proses validasi model, sehingga untuk memodelkan *plant* LGO *stripper* digunakanlah persamaan FOPDT, dimana persamaan tersebut hanya dapat dibangun dari grafik respon yang menunjukkan kinerja *plant* saat proses dinamik. Persamaan FOPDT telah banyak digunakan untuk merepresentasikan *plant* yang berorde tinggi. Fungsi transfer FOPDT dianggap telah dapat mewakili *real plant*.

Persamaan FOPDT didapatkan dari grafik simulasi *plant* menggunakan *software* HYSYS dengan tujuan untuk mengetahui dinamika *plant* secara *real*. Kolom LGO *stripper* disimulasikan menggunakan *software* HYSYS 7.3. Kolom LGO *stripper* disimulasikan sebagai *absorber* dan didapat dalam *file* HYSYS (\*.hsc) serta menggunakan model termodinamika *Soave Redlich – Kwong* (SRK) [21].

Pada HYSYS kolom LGO *stripper* berbentuk *process flow sheet diagram*, dan berjalan dalam kondisi dinamik. Unit operasi tersebut memiliki dua buah *input* dan dua buah *output*, sebuah *heat exchanger*, serta sebuah *loop* pengendalian yaitu LIC-005. Tampilan dari *process flow sheet diagram* kolom LGO *stripper* pada HYSYS tampak pada gambar 3.3 di bawah ini.

Pemodelan proses unit operasi kolom LGO *stripper* dilakukan dengan menggunakan metode *First Order Plus Dead Time* (FOPDT). Pemodelan tersebut dilakukan untuk mengidentifikasi perilaku dari unit operasi kolom LGO *stripper*. Pada FOPDT terdapat tiga variabel karakteristik, yaitu *process gain*, *process time constant*, dan *dead time*. Ketiganya merepresentasikan perilaku dari unit operasi kolom LGO *stripper*.



**Gambar 3.3** *Process Flow Sheet Diagram* Kolom LGO *Stripper* pada HYSYS

Prosedur yang dilakukan untuk pemodelan unit operasi kolom LGO *stripper* adalah sebagai berikut.

Pengambilan data *open loop* yang dilakukan dengan mengubah mode kontroler *loop* pengendali pada HYSYS dari mode auto menjadi mode manual. Mengubah *manipulated variable* (OP) pada kontroler dengan memberikan masukan *step* sebesar sekian % dari besarnya *range process variable* (PV). Pengambilan data dihentikan ketika unit proses telah stabil di keadaan baru.

Perhitungan data yang dihasilkan dari respon *open loop* untuk mendapatkan *process gain*, *process time constant*, dan *dead time*. Persamaan – persamaan yang digunakan adalah sebagai berikut.

$$K = \frac{\Delta}{\delta} \quad (3.5)$$

$$t_{28\%} = \theta + \frac{\tau}{3} \quad (3.6)$$

$$t_{63\%} = \theta + \tau \quad (3.7)$$

$$\tau = 1,5(t_{63\%} - t_{28\%}) \quad (3.8)$$

$$\theta = t_{63\%} - \tau \quad (3.9)$$

dimana,

$K$  : *gain steady state*

$\Delta$  : perubahan *steady state* pada keluaran proses

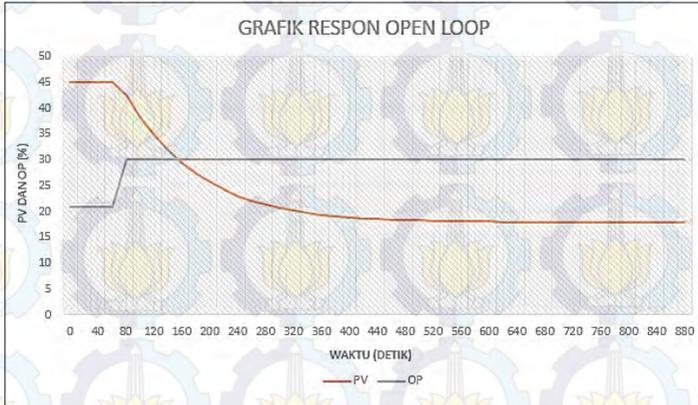
$\delta$  : perubahan *step* pada masukan

$t_{28\%}$  : nilai PV ketika mencapai 28% perubahan

$t_{63\%}$  : nilai PV ketika mencapai 63% perubahan

Dari beberapa persamaan yang telah disebutkan di atas, maka persamaan FOPDT akan didapatkan seperti persamaan (3.10) di bawah ini.

$$FOPDT = G_m(s) = K \frac{e^{-\theta s}}{\tau s + 1} \quad (3.10)$$



**Gambar 3.4** Grafik Hasil Simulasi *Open Loop* pada *Software HYSYS*

Berdasarkan grafik pada Gambar 3.4 dan tabel data pada lampiran C.3, dapat dicari nilai – nilai parameter untuk mendapatkan persamaan FOPDT seperti pada persamaan (3.11) di bawah ini.

$$\frac{H(s)}{\dot{m}_{HEout}(s)} = \frac{Kp}{\tau ps + 1} \quad (3.11)$$

Dimana nilai – nilai parameter tersebut diantaranya adalah *span*, waktu saat *process variable* mencapai 63%, waktu saat *process variable* mencapai 28%, *time constant*, selisih perubahan masukan, dan *gain*.

Beberapa poin di bawah ini akan menjelaskan masing – masing nilai parameter untuk membentuk sebuah persamaan FOPDT yang didasarkan pada tabel data (lampiran C.3).

- *Span*, didapatkan dari pengurangan antara nilai *process variable* awal dengan nilai *process variable* akhir.

$$\Delta = PV_{awal} - PV_{akhir} = 44,9999 - 17,8969 = 27,1030$$

- Waktu saat *process variable* mencapai 63%, didapat melalui rumus berikut ini.

$$t_{63\%} = PV_{awal} - (0,63x\Delta) = 44,9999 - (0,63x27,1030)$$

$$t_{63\%} = 44,9999 - (0,63x27,1030)$$

$$t_{63\%} = 27,925$$

Dari nilai 27,925 yang telah didapat, kemudian dilakukan interpolasi sedemikian rupa dengan variabel waktu hingga didapatkan nilai waktu 63% adalah pada detik ke – 174.

- Waktu saat *process variable* mencapai 28%, didapat melalui rumus berikut ini.

$$t_{63\%} = PV_{awal} - (0,28x\Delta) = 44,9999 - (0,28x27,1030)$$

$$t_{63\%} = 44,9999 - (0,28x27,1030)$$

$$t_{63\%} = 37,4110$$

Dari nilai 37,4110 yang telah didapat, kemudian dilakukan interpolasi sedemikian rupa dengan variabel waktu hingga didapatkan nilai waktu 28% adalah pada detik ke – 105.

- *Time constant*, didapat melalui rumus berikut ini.

$$tp = 1,5(t_{63\%} - t_{28\%})$$

$$tp = 103,5062$$

- Selisih perubahan masukan, didapat melalui rumus berikut ini.

$$\delta = OP_{akhir} - OP_{awal} = 30 - 20,95 = 9,05$$

- *Gain*, didapat melalui rumus berikut ini.

$$Kp = \frac{\Delta}{\delta}$$

$$Kp = \frac{27,103}{9,05} = 2,995$$

Dengan demikian, persamaan (3.11) dapat dituliskan sebagai berikut.

$$\frac{H(s)}{\dot{m}_{HEout}(s)} = \frac{2,995}{103,51s + 1} \quad (3.12)$$

### 3.4 Pemodelan Matematis Aktuator

Aktuator dari sistem pengendalian level pada LGO *Stripper* ini adalah *control valve* dengan karakteristik *normally open* atau *failure to close* yang berfungsi untuk mengatur besar laju aliran dari *light gas oil* yang keluar dari *plant*. *Input* dari *control valve* ini berupa sinyal arus yang memiliki rentang 4 – 20 mA dan dikonversikan ke dalam sinyal pneumatik dalam rupa bukaan *control valve* 0 – 100% dan menghasilkan *ouput* berupa besar laju aliran *light gas oil*. *Gain* aktuator didapatkan dari persamaan berikut.

$$\frac{m_b(s)}{U(s)} = \frac{K_v}{\tau s + 1} \quad (3.13)$$

Keterangan :

$m_b(s)$  : laju aliran yang termanipulasi (kg/s)

$U(s)$  : sinyal masukan ke *control valve* (mA) yang dikonversi dalam rupa bukaan *control valve* 0 – 100%

$K_v$  : *gain total control valve*

$\tau$  : *time constant control valve* (s)

Untuk menghitung *gain control valve* digunakan persamaan di bawah ini.

$$K_v = \frac{\text{lajualiran}}{\text{valveopening}} \quad (3.14)$$

Sesuai data proses dimana dengan bukaan *control valve* sebesar 20,95%, serta berdasarkan spesifikasi *control valve* dimana nilai laju aliran yang dihasilkan adalah 15,52 kg/s, maka akan didapatkan nilai *gain control valve* seperti berikut ini.

$$K_v = \frac{15,52}{20,95} = 0,74081 \quad (3.15)$$

sedangkan untuk memperoleh nilai *time constant* efektif *control valve* adalah berdasarkan hubungan waktu *stroke*, perfeksional terhadap posisi *control valve*, dan perbandingan konstanta waktu *inferent* terhadap waktu *stroke* yang dinyatakan dengan persamaan,

$$\tau = T_v (\Delta V + R_v) \quad (3.16)$$

Keterangan :

$\tau$  : *time constant control valve* (detik)

$T_v$  : waktu *stroke* penuh (1,5 detik)

$\Delta V$  : (kuantitas maks. – kuantitas min.)/kuantitas maks.

$R_v$  : perbandingan konstanta waktu *inferent* terhadap waktu *stroke* (0,03 untuk *diaphragm* dan 0,3 untuk piston)

$$\Delta V = \frac{Quan_{maks} - Quan_{min}}{Quan_{maks}} = \frac{22,326 - 0}{22,326} = \frac{22,326}{22,326} = 1$$

$$\tau = T_v (\Delta V + R_v) = 1,5(1 + 0,3) = 1,95$$

Sehingga fungsi transfer dari aktuator (*control valve*) didapatkan pada persamaan (3.21) berikut ini.

$$\frac{m_b(s)}{U(s)} = \frac{K_v}{\tau s + 1} = \frac{0,74081}{1,95s + 1} \quad (3.17)$$

### 3.5 Pemodelan Matematis Sensor

Sensor dari sistem pengendalian level pada LGO *Stripper* ini adalah *Displacer Level Transmitter*, yaitu suatu alat yang prinsip kerjanya didesain untuk menghasilkan nilai gerak dimana nilai gerak dari *displacer* tersebut menggantikan (*displacement*) nilai volume cairan yang menghasilkan gerak tersebut.

Pemodelan pada sensor menggunakan nilai num dan denum sebesar 1 agar nilai keluaran dari sensor menuju unit pengendali merepresentasikan nilai level yang terukur.

### 3.6 Validasi Pemodelan

Validasi pemodelan dilakukan guna memberikan kepastian terhadap pemodelan sistem yang telah didapatkan. Validasi dilakukan terhadap dari *plant*, sensor, dan aktuator. Validasi pemodelan pada tugas akhir ini dilakukan dengan dua buah jenis validasi, validasi yang pertama adalah validasi simulasi *software* HYSYS dengan data proses, dan validasi yang kedua adalah validasi *open loop* menggunakan *simulink* MATLAB.

Tabel 3.2 hingga tabel 3.16 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 level dalam satuan (%).

**Tabel 3.2** Tabel Data Proses Perusahaan dengan OP 22,9%

Waktu, Tanggal	OP (%)	PV (%)
2/3/2015 7:36	22.99	35.71
2/2/2015 12:24	22.91	34.64
Rata – rata		35,18

**Tabel 3.3** Tabel Data Proses Perusahaan dengan OP 20,9%

Waktu, Tanggal	OP (%)	PV (%)
1/30/2015 10:14	20.92	44.29
1/30/2015 10:17	20.93	43.58
2/2/2015 4:34	20.93	44.55
2/2/2015 1:29	20.95	43.36
Rata – rata		43,95

**Tabel 3.4** Tabel Data Proses Perusahaan dengan OP 22,36%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/3/2015 3:54	22.36	37.24
1/31/2015 7:11	22.36	39.76
2/1/2015 5:36	22.38	37.74
Rata – rata		38,25

**Tabel 3.5** Tabel Data Proses Perusahaan dengan OP 23%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/3/2015 2:45	23.00	34.44
2/3/2015 4:44	23.00	36.29
2/2/2015 11:59	23.09	35.72
2/2/2015 14:56	23.00	36.36
Rata – rata		35,70

**Tabel 3.6** Tabel Data Proses Perusahaan dengan OP 23,3%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 13:00	23.34	33.90
2/2/2015 9:10	23.35	34.77
2/2/2015 14:59	23.35	35.26
2/2/2015 13:00	23.36	33.90
2/2/2015 10:51	23.36	33.98
2/3/2015 2:05	23.36	34.81
Rata – rata		34,44

**Tabel 3.7** Tabel Data Proses Perusahaan dengan OP 24,6%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 16:54	24.63	32.29
2/3/2015 9:21	24.63	32.35
Rata – rata		32,32

**Tabel 3.8** Tabel Data Proses Perusahaan dengan OP 24,1%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 22:11	24.19	32.46

**Tabel 3.9** Tabel Data Proses Perusahaan dengan OP 24,3%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/3/2015 0:57	24.35	32.66

**Tabel 3.10** Tabel Data Proses Perusahaan dengan OP 23,7%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 14:38	23.73	33.41
2/3/2015 3:47	23.71	35.89
Rata – rata		34,65

**Tabel 3.11** Tabel Data Proses Perusahaan dengan OP 23,4%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 18:42	23.44	34.86
2/3/2015 2:20	23.47	33.63
Rata – rata		34,24

**Tabel 3.12** Tabel Data Proses Perusahaan dengan OP 23,2%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/2/2015 23:05	23.24	34.18
2/2/2015 21:24	23.23	33.68
2/3/2015 6:26	23.23	33.69
Rata – rata		33,85

**Tabel 3.13** Tabel Data Proses Perusahaan dengan OP 23,9%

<b>Waktu, Tanggal</b>	<b>OP (%)</b>	<b>PV (%)</b>
2/3/2015 7:34	23.97	33.71

**Tabel 3.14** Tabel Data Proses Perusahaan dengan OP 23,1%

Waktu, Tanggal	OP (%)	PV (%)
2/2/2015 8:20	23.10	36.45
2/2/2015 8:08	23.11	34.82
2/3/2015 6:09	23.19	34.42
2/2/2015 14:34	23.15	35.74
Rata – rata		35,36

**Tabel 3.15** Tabel Data Proses Perusahaan dengan OP 23,6%

Waktu, Tanggal	OP (%)	PV (%)
2/2/2015 18:48	23.66	35.34
2/3/2015 8:50	23.68	33.60
2/3/2015 4:34	23.62	35.12
2/3/2015 0:59	23.61	36.10
Rata – rata		35,04

**Tabel 3.16** Tabel Data Proses Perusahaan dengan OP 23,8%

Waktu, Tanggal	OP (%)	PV (%)
2/2/2015 10:14	23.86	35.14
2/3/2015 7:50	23.87	35.18
Rata – rata		35,16

Tabel 3.17 berikut ini merupakan tabel data hasil simulasi menggunakan *software* HYSYS.

**Tabel 3.17** Tabel Data Simulasi HYSYS

OP (%)	PV (%)
22.9	36.76
20.9	45.28
22.3	39.11
23	36.38
23.3	35.28

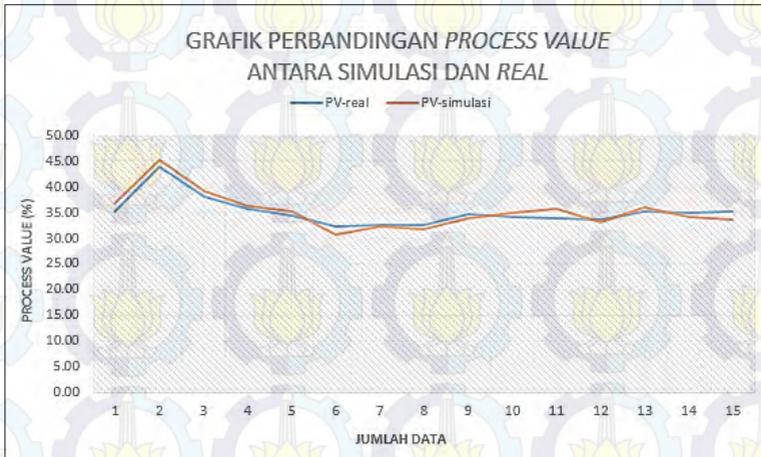
24.63	30.81
24.19	32.22
24.35	31.7
23.7	33.87
23.4	34.92
23.2	35.64
23.9	33.18
23.1	36.01
23.6	34.21
23.8	33.52

Dari beberapa buah tabel yang telah dijelaskan sebelumnya, Tabel 3.18 berikut ini merupakan ringkasan tabel data perbandingan antara *process variable* asli dengan *process variable* hasil simulasi. Di dalam tabel ini pula diberikan hasil nilai perhitungan *error* antara keduanya.

**Tabel 3.18** Tabel Perhitungan Nilai *Error* dari *Process Value* Asli dengan Simulasi

<b>OP (%)</b>	<b>PV ASLI (%)</b>	<b>PV SIMULASI (%)</b>	<b>ERROR (%)</b>	
22,9	35.18	36.76	4.49	4.49
20,9	43.95	45.28	3.03	3.03
22,36	38.25	39.11	2.25	2.25
23	35.7	36.38	1.90	1.90
23,3	34.44	35.28	2.44	2.44
24,6	32.32	30.81	-4.67	4.67
24,1	32.46	32.22	-0.74	0.74
24,3	32.66	31.7	-2.94	2.94
23,7	34.65	33.87	-2.25	2.25
23,4	34.24	34.92	1.99	1.99
23,2	33.85	35.64	5.29	5.29

23,9	33.71	33.18	-1.57	1.57
23,1	35.36	36.01	1.84	1.84
23,6	35.04	34.21	-2.37	2.37
23,8	35.16	33.52	-4.66	4.66
<b>Presentase Error Rata – rata</b>				2.83

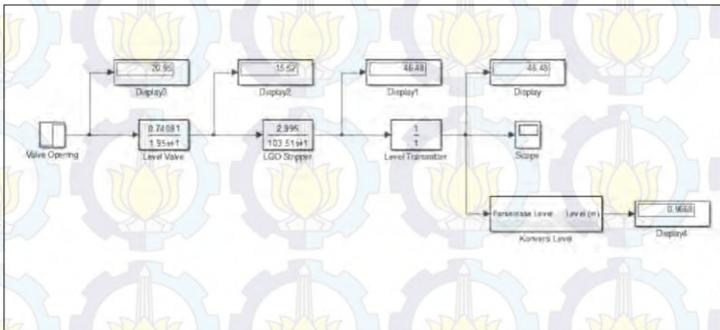


**Gambar 3.5** Grafik Perbandingan antara Hasil Simulasi *Software* HYSYS dengan *Real Plant*

Dari tabel 3.18 dan Gambar 3.5 di atas, dapat ditentukan bahwa data hasil simulasi *software* HYSYS dengan data proses perusahaan valid karena memiliki nilai *error* sebesar 2,83%.

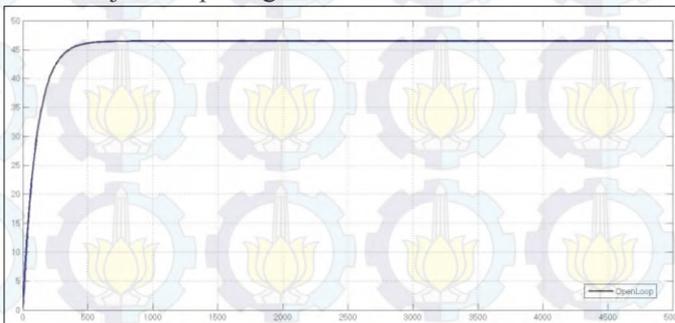
Validasi pemodelan yang kedua dilakukan dengan menggunakan data proses kendali level pada LGO *stripper*, dimana pengujian dilakukan dengan cara memberikan nilai – nilai standar proses berupa bukaan *control valve* sebesar 20,95% sebagai *input* dan didapat laju aliran *light gas oil* pada data proses sebesar 15,52 kg/s diharapkan dapat menghasilkan level dasar kolom pada *light gas oil stripper* sebesar 45% atau jika dikonversikan ke dalam satuan meter adalah sebesar 0,936 meter.

Gambar di bawah ini merupakan skema dari sistem pengendalian *open loop*. Sistem pengendalian *open loop* adalah sebuah sistem pengendalian dimana komponen sistemnya terdiri dari bukaan valve, aktuator, *plant*, dan sensor saja tanpa dipasang komponen pengendali dengan tujuan untuk mengetahui karakteristik sistem terlebih dahulu.



**Gambar 3.6** Skema Blok Diagram Simulasi MATLAB *Open Loop*

Hasil respon sistem pengendalian *open loop* tanpa pengendali dengan menggunakan *simulink* software MATLAB R2013a ditunjukkan pada gambar berikut ini.



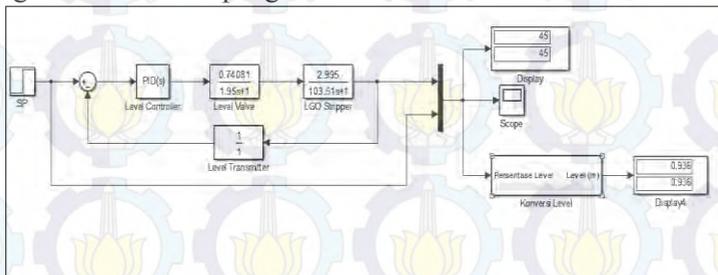
**Gambar 3.7** Grafik Respon *Open Loop* Sistem

Hasil respon diatas menunjukkan bahwa pemodelan sistem LGO *stripper* adalah berorde satu karena pemodelannya yang

menggunakan persamaan FOPDT dan dapat dikatakan valid karena hasil level yang sebesar 46,48% hampir sesuai dengan data level *light gas oil stripper* pada saat kondisi normal dan mantap yaitu sebesar 45%. Terdapat sedikit perbedaan nilai, dikarenakan belum adanya komponen pengendali pada sistem *open loop*.

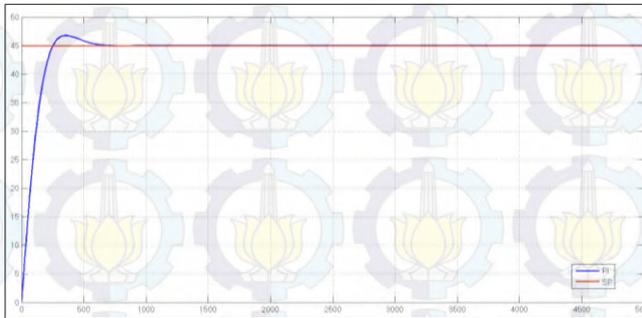
### 3.7 Perancangan Pengendali *Proportional Integral* (PI)

Pengendali PI dirancang dengan menggunakan *simulink software* MATLAB R2013a dengan metode *trial and error* dalam bentuk diagram blok sistem pengendalian *closed loop* dibawah ini. Nilai parameter P dan I didapatkan dari hasil simulasi diagram blok sistem pengendalian tersebut.



**Gambar 3.8** Skema Blok Diagram Simulasi MATLAB *Closed Loop*

Dengan menggunakan metode *trial and error* didapatkan nilai *gain proportional* sebesar 0,30276, *gain integral* sebesar 0,0053, dan *gain derivative* sebesar 0. Hasil grafik menunjukkan bahwa *maximum overshoot* sebesar 3,899%, *settling time* sebesar 1127 detik, dan *error steady state* sebesar 0,0005%. Untuk hasil *tuning trial and error* didapatkan grafik respon sistem pengendalian *closed loop* sebagai berikut.



**Gambar 3.9** Grafik Respon Simulasi Sistem Pengendalian *Closed Loop LGO Stripper*

### 3.8 Perancangan *Active Fault Tolerant Control (AFTC)*

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

#### 3.8.1 Perancangan *Observer* untuk Sensor

Perancangan *observer* digunakan untuk mengestimasi kesalahan pembacaan pada sensor level. Perancangan dilakukan dengan mengubah fungsi transfer sistem ke dalam persamaan ruang keadaan (*state space*). Untuk mendapatkan matriks *state space* maka terlebih dahulu didefinisikan nilai *num* dan *denum* yang didapatkan dari perkalian fungsi transfer *plant* (pers. (3.11)), fungsi transfer aktuator (pers. (3.17)), dan fungsi transfer sensor sebesar 1.

$$FT_{sistem} = \frac{2,2187}{201,845s^2 + 105,46s + 1} \quad (3.18)$$

Dari fungsi transfer sistem di atas, maka dapat didefinisikan nilai *num* dan *denum* sebagai berikut.

$$num = [2.2187]$$

$$denum = [201.845 \quad 105.46 \quad 1]$$

Dengan menggunakan fungsi ‘*tf2ss*’ pada *command window* MATLAB, maka didapatkan matriks persamaan ruang

keadaan sebagai berikut. Nilai denum yang berorde dua menyebabkan matriks yang terbentuk adalah  $2 \times 2$  untuk matriks A,  $2 \times 1$  untuk matriks B,  $1 \times 2$  untuk matriks C dan nol untuk matriks D karena *disturbance* atau *noise* diabaikan. Matriks A merupakan matriks sistem (matriks pada *state*), matriks B merupakan matriks *input*, matriks C merupakan matriks *output* (matriks hasil pengukuran), dan matriks D merupakan matriks transmisi langsung. Berikut ini adalah matriks A, B, C, dan D yang didapatkan dari fungsi transfer sistem yang telah dijabarkan pada persamaan (3.18).

$$A = \begin{bmatrix} -0.5225 & -0.0050 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C = [0 \quad 0.0110]$$

$$D = [0]$$

Persamaan ruang keadaan secara umum dituliskan sebagai berikut.

$$\dot{X} = AX + BU \quad (3.19)$$

$$Y = CX + DU \quad (3.20)$$

Dengan adanya kesalahan pada sensor, maka persamaan ruang keadaan diatas akan menjadi persamaan (3.21) dan (3.23).

$$\dot{X} = AX + BU + \Gamma w \quad (3.21)$$

Nilai  $\Gamma w$  dapat diuraikan lagi menjadi persamaan (3.22).

$$\dot{X} = AX + BU + Ef_a + D_a d \quad (3.22)$$

$$Y = CX + DU + \beta V \quad (3.23)$$

Nilai  $\beta V$  dapat diuraikan lagi menjadi persamaan (3.24).

$$Y = CX + DU + Ff_s \quad (3.24)$$

dimana,

$$\dot{Z} = Y - Z \quad (3.25)$$

sehingga,

$$\dot{Z} = CX + Ff_s + DU - Z \quad (3.26)$$

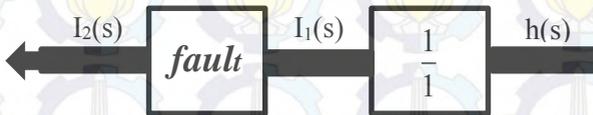
Keterangan :

$F$  : matriks kesalahan sensor

$f_s$  : vektor dari kesalahan sinyal yang dihasilkan oleh sensor

$\dot{Z}$  : *integral error vector*

Adapun *fault* sensor didefinisikan sebagai kesalahan pembacaan sensor terhadap variabel level. Penggambaran *fault* sensor tersebut dapat dilihat pada blok diagram berikut ini.



**Gambar 3.10** Blok Diagram *Fault* Sensor

$$U_r = \alpha U_o - \beta \quad (3.27)$$

$$\frac{1}{1} = \frac{I_1(s)}{h(s)} \quad (3.28)$$

$$I_1(s) = h(s) \quad (3.29)$$

$$I_2(s) = h(s) \cdot U_r = h(s) \cdot (\alpha U_o - \beta) \quad (3.30)$$

dimana,

$I_1(s)$  : *Output* sensor benar

$h(s)$  : pembacaan level dari sistem

$I_2(s)$  : *Output* sensor yang dipengaruhi oleh kesalahan

$U_r$  : sinyal kontrol *input residual* (sinyal kontrol *input* yang masuk pada sensor setelah diberi pembebanan berupa *gain* kesalahan)

- $U_o$  : sinyal kontrol *input* (sinyal kontrol *input* sebelum diberi pembebanan berupa *gain* kesalahan)  
 $\alpha$  : *gain* kesalahan *sensitivitas*  
 $\beta$  : *gain* kesalahan bias

Dari gambar dapat ditunjukkan bahwa blok *fault* diletakkan setelah sensor, dan *observer* dibutuhkan untuk mengestimasi nilai *fault* yang diberikan. Untuk model dari kesalahan sensor, karena fungsi transfer sensor adalah sebesar 1, maka persamaan yang menggambarkan model *fault* adalah pada persamaan 3.30. *Observer* memiliki dua masukan, dengan tujuan untuk membandingkan antara sinyal kontrol yang benar dengan hasil sensor yang telah mengalami kesalahan.

Estimasi kesalahan oleh *observer* terhadap hasil pengukuran dilakukan melalui pendekatan persamaan (3.31), (3.32), dan (3.33) berikut ini.

$$\dot{\hat{X}}_s = A_s \hat{X}_s + B_s U + L(Y - \hat{Y}) \quad (3.31)$$

dimana,

- $\hat{X}_s$  : *state* yang telah diestimasi karena kesalahan sensor  
 $A_s$  : matriks A yang telah diestimasi karena kesalahan sensor  
 $\hat{X}_s$  : vektor dari *state* yang telah diestimasi karena kesalahan sensor  
 $B_s$  : matriks B telah diestimasi karena kesalahan sensor  
 $L$  : matriks *gain observer*  
 $Y - \hat{Y}$  : *error*

$$Y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \\ f_s \end{bmatrix} \quad (3.32)$$

$$f_s = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{X}_s \quad (3.33)$$

Berdasarkan persamaan (3.31), (3.32), dan (3.33), maka ditambahkan matriks tambahan untuk persamaan estimasi kesalahan sebagai berikut.

$$\begin{bmatrix} \dot{X} \\ \dot{Z} \\ f_s \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ C & -I & F \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \\ f_s \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} U + \begin{bmatrix} D_d & 0 & 0 \\ 0 & D_w & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} d \\ w \\ f_s \end{bmatrix}$$

Persamaan estimasi kesalahan dapat disederhanakan menjadi persamaan (3.34).

$$\dot{X}_s = A_s X_s + B_s U + D_s d_D \quad (3.34)$$

Untuk mengestimasi kesalahan pada hasil pengukuran dilakukan pendefinisian matriks seperti berikut ini.

$$A_s = \begin{bmatrix} A_1 & A_2 & 0 & 0 \\ A_3 & A_4 & 0 & 0 \\ C_1 & C_2 & -1 & F \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_s = \begin{bmatrix} -0.5225 & -0.0050 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.0110 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_s = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

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

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ f_a \end{bmatrix} + D_\omega \cdot e_s$$

$$Y = C_s \cdot X_s + D_\omega \cdot e_s$$

Setelah matriks persamaan ruang keadaan didapatkan, matriks digunakan untuk mencari *gain observer* L menggunakan metode LMI melalui analisa kestabilan Lyapunov. Melalui analisa kestabilan Lyapunov metode kedua, maka penyelesaian dalam LMI dapat ditulis pada persamaan (3.35).

$$LMI = \begin{bmatrix} e^T & V^T \end{bmatrix} \begin{bmatrix} P(A_s - LC_s) + (A - LC)^T P + I_m I_m^T & PD_s \\ (D_s)^T P_s & -\gamma^2 I_{d+m} \end{bmatrix} \begin{bmatrix} e \\ V \end{bmatrix} < 0 \quad (3.35)$$

Selanjutnya nilai gain L didapatkan melalui *iterasi function* LMI dengan program MATLAB (R2013a). Iterasi dilakukan dengan mendefinisikan LMI *constraint* dengan *command* sebagai berikut.

```
setlmis([])
P = lmivar(1, [4 1]);
Y = lmivar(2, [4 1]);
gamma = lmivar(1, [1 0]);

lmitem([1 1 1 P], 1, As, 's')
lmitem([1 1 1 Y], -1, Cs, 's')
lmitem([1 1 1 0], 1)
lmitem([1 1 2 P], 1, Ds)
lmitem([1 2 2 gamma], -1, 1)
lmitem([-2 1 1 P], 1, 1)
```

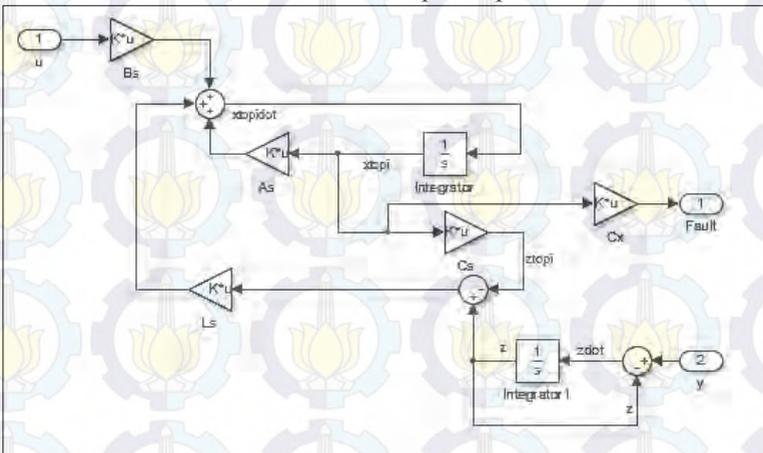
Selanjutnya untuk melakukan iterasi nilai  $L$ , digunakan *function* LMI Solver dengan metode *Minimization*, sehingga *command* yang digunakan adalah sebagai berikut.

```
LMIs = getlmis
[tmin, xfeas] = feasp(LMIs)
Yopt = dec2mat(LMIs, xfeas, Y);
Popt = dec2mat(LMIs, xfeas, P);
Ls = inv(Popt) * Yopt;
```

Maka didapatkan matriks *gain observer* sensor sebagai berikut.

$$L_s = \begin{bmatrix} 0.0051 \\ 0.0099 \\ 0.6920 \\ 1.8811 \end{bmatrix}$$

Jika persamaan estimasi kesalahan dibuat dalam program *simulink* MATLAB, maka akan tampak seperti Gambar 3.11.

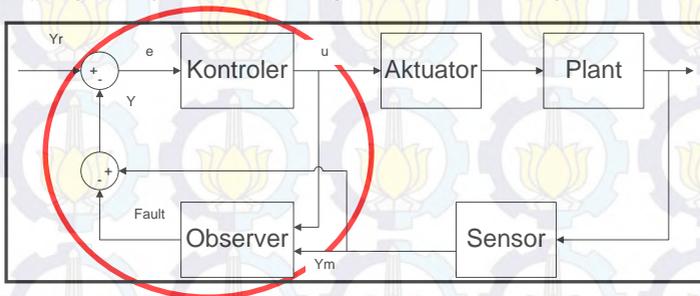


**Gambar 3.11** Struktur *Observer* untuk Sensor

Gambar 3.11 merepresentasikan persamaan (3.31), dimana  $u$  merupakan sinyal yang berasal dari kontroler yang masuk ke dalam *observer* dan  $y$  merupakan sinyal hasil pengukuran sensor yang mengalami kesalahan. Dengan persamaan yang digambarkan sedemikian rupa dalam *simulink* MATLAB, *observer* mengeluarkan besarnya nilai *fault* ( $f_s$ ) sebagai hasil akhir dari perancangannya.

### 3.8.2 Rekonfigurasi Sinyal Pengendali untuk Sensor

Rekonfigurasi sinyal pengendali merupakan tahap akhir perancangan AFTC yang bertujuan untuk mengembalikan kestabilan sistem pada saat terjadi kesalahan sensor. Perancangan skema AFTC dibentuk dari *modified PD descriptor observer* dan rekonfigurasi sinyal pengendali akibat kesalahan sensor adalah seperti yang ditunjukkan oleh lingkaran merah sebagai berikut.



**Gambar 3.12** Skema Rekonfigurasi Sinyal Pengendali dari *Observer* Sensor

Rekonfigurasi pengendali ini dilakukan dengan menambahkan sinyal pengendali berupa  $Y$ . Nilai  $Y$  ini didapatkan dari perhitungan sebagai berikut.

$$Y = Y_u - X_s \quad (3.36)$$

dimana,

$Y$  : nilai sebenarnya

$Y_u$  : nilai hasil pengukuran

$X_s$  : nilai *fault*

Dengan adanya penambahan sinyal pengendali tersebut, maka didapatkan nilai *error* dengan perumusan yang baru sebagai berikut.

$$e = Y_r - Y \quad (3.37)$$

dimana,

$e$  : nilai *error*

$Y_r$  : nilai *setpoint*

### 3.8.3 Perancangan *Observer* untuk Aktuator

Perancangan *observer* digunakan untuk mengestimasi kesalahan pembacaan pada sensor level. Perancangan dilakukan dengan mengubah fungsi transfer sistem ke dalam persamaan ruang keadaan (*state space*). Untuk mendapatkan matriks *state space* maka terlebih dahulu didefinisikan nilai *num* dan *denum* yang didapatkan dari perkalian fungsi transfer *plant* (pers. (3.11)), fungsi transfer aktuator (pers. (3.17)), dan fungsi transfer sensor sebesar 1.

$$FT_{sistem} = \frac{2,2187}{201,845s^2 + 105,46s + 1} \quad (3.38)$$

Dari fungsi transfer sistem di atas, maka dapat didefinisikan nilai *num* dan *denum* sebagai berikut.

$$num = [2.2187]$$

$$denum = [201.845 \quad 105.46 \quad 1]$$

Dengan menggunakan fungsi 'tf2ss' pada *command window* MATLAB, maka didapatkan matriks persamaan ruang keadaan sebagai berikut. Nilai *denum* yang berorde dua menyebabkan matriks yang terbentuk adalah 2x2 untuk matriks A, 2x1 untuk matriks B, 1x2 untuk matriks C dan nol untuk matriks D karena *disturbance* atau *noise* diabaikan. Matriks A merupakan matriks sistem (matriks pada *state*), matriks B merupakan matriks *input*, matriks C merupakan matriks *output*

(matriks hasil pengukuran), dan matriks D merupakan matriks transmisi langsung. Berikut ini adalah matriks A, B, C, dan D yang didapatkan dari fungsi transfer sistem yang telah dijabarkan pada persamaan (3.31).

$$A = \begin{bmatrix} -0.5225 & -0.0050 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [0 \quad 0.0110]$$

$$D = [0]$$

Persamaan ruang keadaan secara umum dituliskan sebagai berikut.

$$\dot{X} = AX + BU \quad (3.39)$$

$$Y = CX + DU \quad (3.40)$$

Dengan adanya kesalahan pada sensor, maka persamaan ruang keadaan diatas akan menjadi persamaan (3.41) dan (3.43).

$$\dot{X} = AX + BU + \Gamma w \quad (3.41)$$

Nilai  $\Gamma w$  dapat diuraikan lagi menjadi persamaan (3.42).

$$\dot{X} = AX + BU + Ef_a + D_d d \quad (3.42)$$

$$Y = CX + DU + \beta V \quad (3.43)$$

Nilai  $\beta V$  dapat diuraikan lagi menjadi persamaan (3.44).

$$Y = CX + DU + Ff_s \quad (3.44)$$

Dengan asumsi bahwa,

$$F = D$$

$$E = B$$

Keterangan :

$F$  : matriks kesalahan aktuator

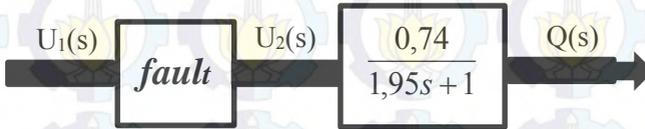
$f_a$  : vektor dari kesalahan sinyal yang dihasilkan oleh aktuator

Maka pada kesalahan aktuator, persamaan (3.44) akan menjadi persamaan (3.45)

$$Y = CX + D(f_s + \omega)$$

$$Y = CX + D.e_s \quad (3.45)$$

Adapun *fault* aktuator didefinisikan sebagai kesalahan bukaan *valve* terhadap sinyal elektrik yang diberikan. Penggambaran *fault* aktuator tersebut dapat dilihat pada blok diagram berikut ini.



**Gambar 3.13** Blok Diagram *Fault* Aktuator

$$U_r = \alpha U_o - \beta = U_2(s) \quad (3.46)$$

$$\frac{0,74}{1,95s + 1} = \frac{Q(s)}{U_2(s)} \quad (3.47)$$

$$Q(s) = \frac{0,74}{1,95s + 1} \cdot U_2(s) \quad (3.48)$$

$$Q(s) = \frac{0,74}{1,95s + 1} \cdot (\alpha U_o - \beta) \quad (3.49)$$

dimana,

$Q(s)$  : laju aliran aktuator

$U_r$  : sinyal kontrol *input residual* (sinyal kontrol *input* yang masuk pada sensor setelah diberi pembebanan berupa *gain* kesalahan)

$U_o$  : sinyal kontrol *input* (sinyal kontrol *input* sebelum diberi pembebanan berupa *gain* kesalahan)

$\alpha$  : gain kesalahan *lost of effectiveness*  
 $\beta$  : gain kesalahan kebocoran

Dari gambar dapat ditunjukkan bahwa blok *fault* diletakkan sebelum aktuator, dan *observer* dibutuhkan untuk mengestimasi nilai *fault* yang diberikan. Untuk model dari kesalahan aktuator, terdapat pada persamaan 3.49. *Observer* memiliki dua masukan, dengan tujuan untuk membandingkan antara sinyal kontrol yang benar dengan hasil sensor yang telah mengalami kesalahan.

Estimasi yang dilakukan oleh *observer* terhadap kesalahan yang terjadi dilakukan melalui pendekatan persamaan (3.50), (3.51), dan (3.52) berikut ini.

$$\dot{\hat{X}}_a = A_a \hat{X}_a + B_a U + L(Y - \hat{Y}) \quad (3.50)$$

dimana,

$\hat{X}_s$  : state yang telah diestimasi karena kesalahan aktuator

$A_a$  : matriks A yang telah diestimasi karena kesalahan aktuator

$\hat{X}_a$  : vektor dari state yang telah diestimasi karena kesalahan aktuator

$B_a$  : matriks B telah diestimasi karena kesalahan aktuator

$L$  : matriks gain *observer*

$Y - \hat{Y}$  : error

$$\hat{Y} = C_a \hat{X}_a \quad (3.51)$$

$$f_a = [0 \quad 1] \hat{X}_a \quad (3.52)$$

Berdasarkan persamaan (3.50), (3.51), dan (3.52), maka ditambahkan matriks tambahan untuk persamaan estimasi kesalahan sebagai berikut.

$$\begin{bmatrix} \dot{X} \\ \dot{f}_a \end{bmatrix} = \begin{bmatrix} A & E \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ f_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U + \begin{bmatrix} D_d & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} d \\ \dot{f}_a \end{bmatrix}$$

Persamaan estimasi kesalahan dapat disederhanakan menjadi persamaan (3.53).

$$\dot{X}_a = A_a \cdot X_a + BU + D_a \cdot d_D \quad (3.53)$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ f_a \end{bmatrix} + D_{\omega} \cdot e_s \quad (3.54)$$

$$Y = C_a \cdot X_a + D_{\omega} \cdot e_s \quad (3.55)$$

Untuk mengestimasi kesalahan pada hasil pengukuran dilakukan pendefinisian matriks seperti berikut ini.

$$A_a = \begin{bmatrix} -0.5225 & -0.0050 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C_a = [0 \quad 0.0110 \quad 0]$$

$$D_d = B$$

$$D_{\omega} = [1]$$

Maka persamaan (3.50), (3.51), dan (3.52), sebagai pendekatan estimasi *fault* pada *observer* dapat dituliskan sebagai berikut.

$$\dot{\hat{X}}_a = \begin{bmatrix} -0.5225 & -0.0050 \\ 1 & 0 \end{bmatrix} \hat{X}_a + \begin{bmatrix} 1 \\ 0 \end{bmatrix} U + L(Y - \hat{Y}) \quad (3.56)$$

$$\hat{Y} = [0 \quad 0.0110] \hat{X}_a \quad (3.57)$$

$$f_a = [0 \quad 1] \hat{X}_a \quad (3.58)$$

Setelah matriks persamaan ruang keadaan didapatkan, matriks digunakan untuk mencari *gain observer* L menggunakan metode LMI melalui analisa kestabilan Lyapunov. Melalui analisa kestabilan Lyapunov metode kedua, maka penyelesaian dalam LMI dapat ditulis pada persamaan (3.59).

$$LMI = \begin{bmatrix} e^T & V^T \end{bmatrix} \begin{bmatrix} P(A_s - LC_s) + (A - LC)^T P + I_m I_m^T & PD_s \\ (D_s)^T P_s & -\gamma^2 I_{d+m} \end{bmatrix} \begin{bmatrix} e \\ V \end{bmatrix} < 0 \quad (3.59)$$

Selanjutnya nilai gain L didapatkan melalui *iterasi function* LMI dengan program MATLAB (R2013a). Iterasi dilakukan dengan mendefinisikan LMI *constraint* dengan *command* sebagai berikut.

```
setlmis([])
P = lmivar(1, [3 1]);
Y = lmivar(2, [3 1]);
gamma = lmivar(1, [1 0]);
lmiterm([1 1 1 P], 1, Aa, 's')
lmiterm([1 1 1 Y], -1, Ca, 's')
lmiterm([1 1 1 0], 1)
lmiterm([1 1 2 Y], 1, D2)
lmiterm([1 1 2 P], -1, D1)
lmiterm([1 2 2 gamma], -1, 1)
lmiterm([-2 1 1 P], 1, 1)
```

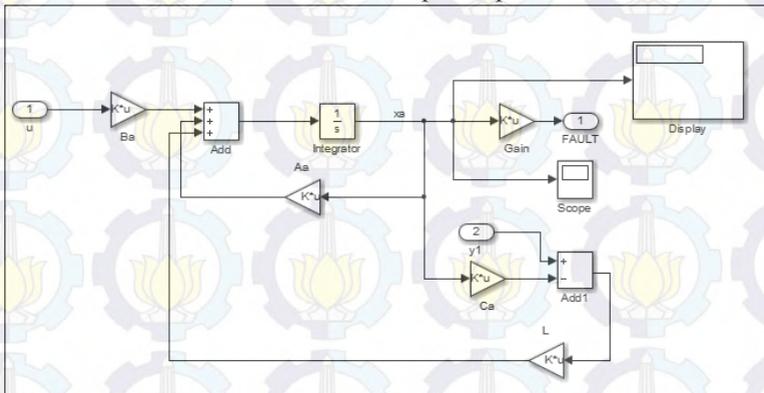
Selanjutnya untuk melakukan iterasi nilai L, digunakan *function LMI Solver* dengan metode *Minimization*, sehingga *command* yang digunakan adalah sebagai berikut.

```
LMIs = getlmis
[tmin, xfeas] = feasp(LMIs)
Yopt = dec2mat(LMIs, xfeas, Y);
Popt = dec2mat(LMIs, xfeas, P);
L = inv(Popt) * Yopt;
```

Maka didapatkan matriks *gain observer* sebagai berikut.

$$L_a = \begin{bmatrix} 4.7068 \\ 46.6223 \\ 2.8190 \end{bmatrix}$$

Jika persamaan estimasi kesalahan dibuat dalam program *simulink* MATLAB, maka akan tampak seperti Gambar 3.14.



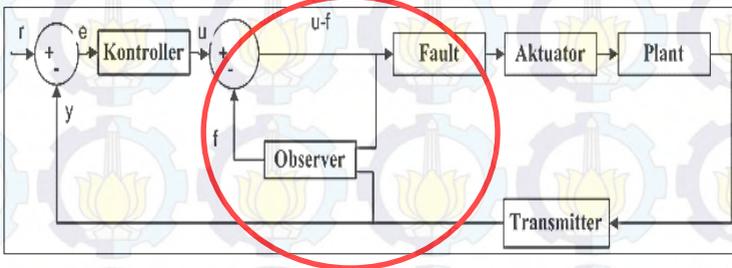
**Gambar 3.14** Struktur *Observer* untuk Aktuator

Gambar 3.14 merepresentasikan persamaan (3.50), dimana  $u$  merupakan sinyal yang berasal dari kontroler yang masuk ke dalam *observer* dan  $y$  merupakan sinyal hasil pengukuran sensor yang mengalami kesalahan. Dengan persamaan yang digambarkan sedemikian rupa dalam *simulink* MATLAB, *observer* mengeluarkan besarnya nilai *fault* ( $f_a$ ) sebagai hasil akhir dari perancangannya.

### 3.8.4 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. Perancangan skema AFTC dibentuk dari *modified PD descriptor observer* dan rekonfigurasi sinyal pengendali akibat kesalahan aktuator adalah seperti yang ditunjukkan oleh lingkaran merah sebagai berikut.



**Gambar 3.15** Skema Rekonfigurasi Sinyal Pengendali dari *Observer* Aktuator

### 3.9 Uji Performansi

Uji performansi dilakukan untuk menguji apakah algoritma AFTC 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 kesalahan bias, uji sensitivitas, dan uji terhadap gangguan (*noise*) untuk *observer* sensor serta uji *lost of effectiveness* dan uji kebocoran untuk *observer* aktuator.



*Halaman ini sengaja dikosongkan*

## BAB IV ANALISIS DATA DAN PEMBAHASAN

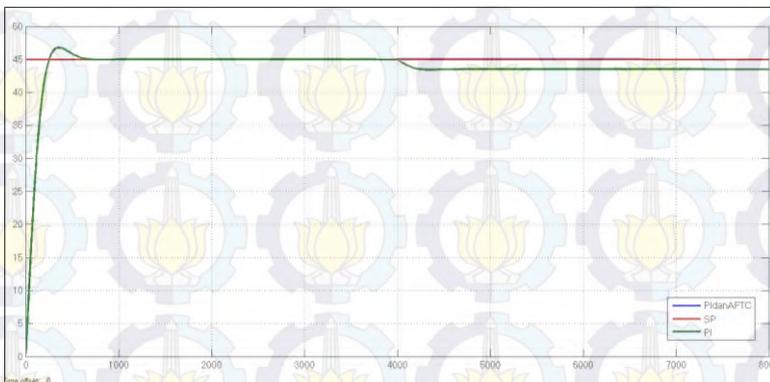
Uji performansi bertujuan untuk mengetahui algoritma AFTC (*Active Fault Tolerant Control*) yang telah dirancang dapat bekerja dengan baik atau tidak jika diberikan kesalahan. Karena perancangan AFTC pada tugas akhir ini berdasarkan pada kesalahan sensor dan aktuator, maka bentuk uji kesalahan yang diberikan terhadap sistem terdiri dari jenis kesalahan pada sensor, pada aktuator, serta pada sensor dan aktuator secara bersamaan. 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 AFTC sama dengan sistem pengendalian PI tanpa algoritma AFTC. Uji kesalahan sensor yang diberikan berupa kesalahan bias, sensitivitas, dan *noise*, sedangkan uji kesalahan aktuator yang diberikan berupa *lost of effectiveness* dan kesalahan kebocoran. Uji kesalahan tersebut diberikan pada satu keadaan saja yakni pada detik ke – 4000.

### 4.1 Uji Kesalahan Sensor

Uji performansi sistem pengendalian level akibat kesalahan sensor dan aktuator yang pertama adalah uji kesalahan sensor yang terdiri dari uji bias, sensitivitas, dan *noise*.

#### 4.1.1 Uji Kesalahan Bias

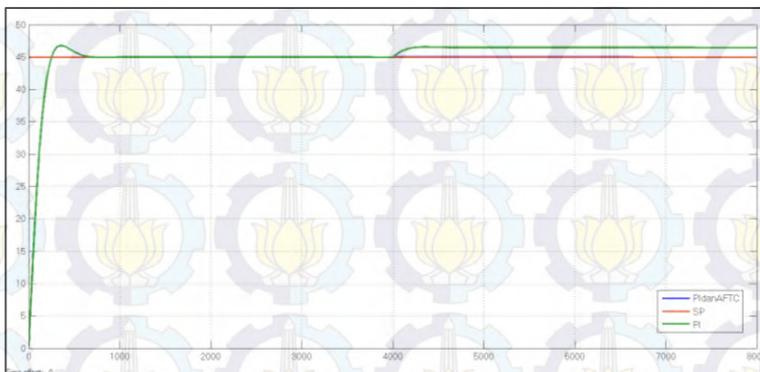
Bias merupakan penyimpangan nilai dari nilai tetapan yang telah ditentukan. Nilai bias yang diberikan pada uji ini adalah  $\pm 1,5\%$  dari *set point* level. Hasil respon sistem dengan kesalahan bias  $+1,5\%$  dapat dilihat pada gambar 4.1 berikut dan respon sistem dengan kesalahan bias  $-1,5\%$  dapat dilihat pada gambar 4.2.



**Gambar 4.1** Respon Sistem dengan Kesalahan Bias +1,5%

**Tabel 4.1** Parameter Performansi dengan Kesalahan Bias +1,5%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0%
2	<i>Maximum Undershoot</i>	3,46%	0,02%
3	<i>Error Steady State</i>	3,3%	0%
4	<i>Settling Time</i>	5500 detik	4302 detik



**Gambar 4.2** Respon Sistem dengan Kesalahan Bias  $-1,5\%$

**Tabel 4.2** Parameter Performansi dengan Kesalahan Bias  $-1,5\%$

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	3,46%	0,02%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	3,33%	0%
4	<i>Settling Time</i>	5500 detik	4302 detik

Pada gambar 4.1 dan 4.2 serta penjabaran nilai parameter yang tampak pada tabel 4.1 dan 4.2, terlihat bahwa respon sistem pengendalian dengan algoritma AFTC mengatasi kesalahan bias sebesar  $\pm 1,5\%$  lebih baik daripada sistem pengendalian tanpa algoritma AFTC.

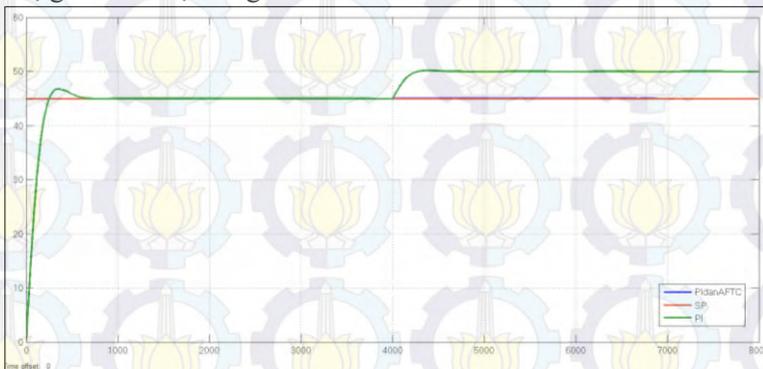
Dari segi parameter, baik *maximum overshoot*, *maximum undershoot*, *error steady state*, maupun *settling time*, sistem pengendalian dengan algoritma AFTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma AFTC. Semakin kecil nilai parameter tersebut, maka semakin baik

performansinya. Hal ini disebabkan oleh adanya rekonfigurasi pengendali atau aksi pengendali yang bekerja untuk mengatasi kesalahan tersebut. Maka dari itu, sistem pengendalian yang menggunakan algoritma AFTC memiliki performansi yang lebih baik dibandingkan dengan sistem pengendalian tanpa menggunakan algoritma AFTC.

#### 4.1.2 Uji Kesalahan Sensitivitas

Sensitivitas berkaitan dengan seberapa peka sebuah sensor menanggapi *input* berupa variabel fisis untuk dikeluarkan dalam rupa nilai yang dapat diolah. Semakin besar nilai sensitivitas sensor, maka semakin baik kinerja sensor tersebut. Nilai sensitivitas yang diberikan pada uji ini adalah 90%, 85%, 80%, dan 75%.

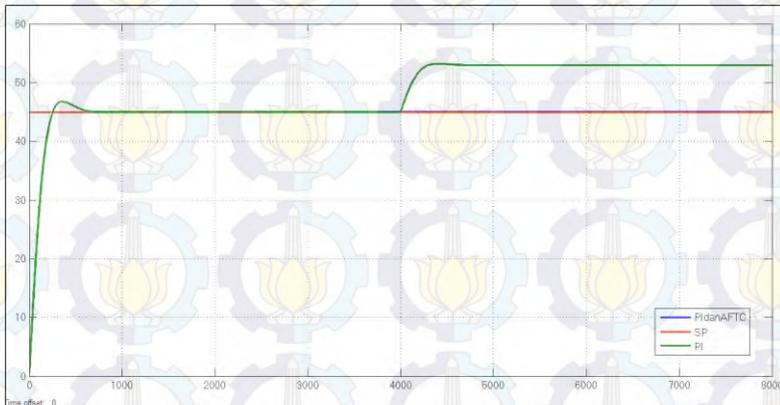
Hasil respon sistem pengendalian level dengan kesalahan sensitivitas 90% ditunjukkan pada gambar 4.3. Selanjutnya, respon sistem pengendalian level dengan kesalahan sensitivitas 85%, 80%, dan 75% masing – masing ditunjukkan pada gambar 4.4, gambar 4.5, dan gambar 4.6.



**Gambar 4.3** Respon Sistem dengan Kesalahan Sensitivitas 90%

**Tabel 4.3** Parameter Performansi dengan Kesalahan Sensitivitas 90%

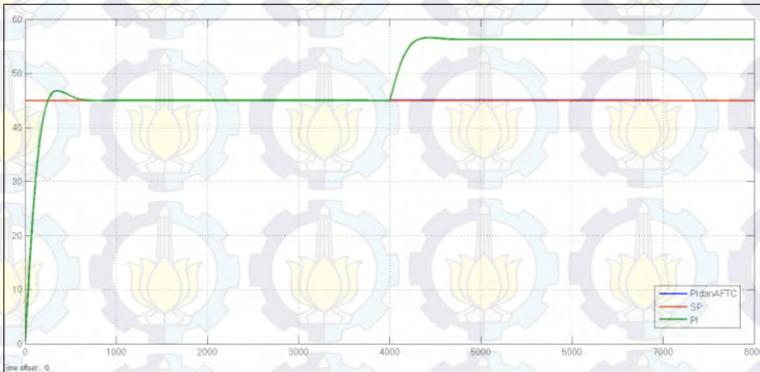
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	11,49%	0,06%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	11,11%	0%
4	<i>Settling Time</i>	5232 detik	4548 detik



**Gambar 4.4** Respon Sistem dengan Kesalahan Sensitivitas 85%

**Tabel 4.4** Parameter Performansi dengan Kesalahan Sensitivitas 85%

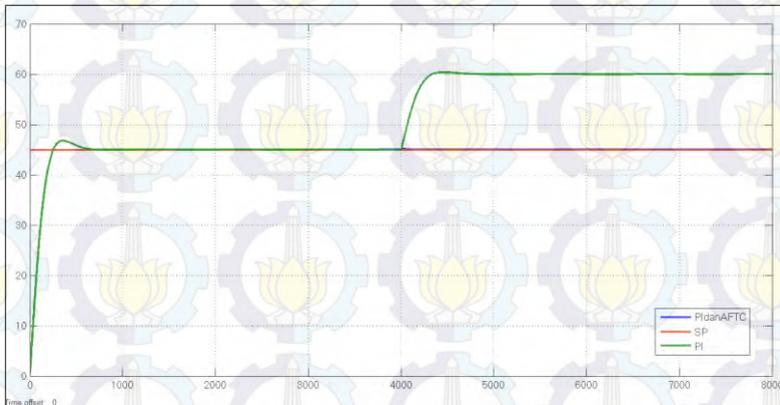
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	18,20%	0,09%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	17,65%	0%
4	<i>Settling Time</i>	6049 detik	5669 detik



**Gambar 4.5** Respon Sistem dengan Kesalahan Sensitivitas 80%

**Tabel 4.5** Parameter Performansi dengan Kesalahan Sensitivitas 80%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	25,70%	0,12%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	25%	0%
4	<i>Settling Time</i>	6061 detik	6010 detik



**Gambar 4.6** Respon Sistem dengan Kesalahan Sensitivitas 75%

**Tabel 4.6** Parameter Performansi dengan Kesalahan Sensitivitas 75%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	34,17%	0,15%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	33,33%	0%
4	<i>Settling Time</i>	6079 detik	6045 detik

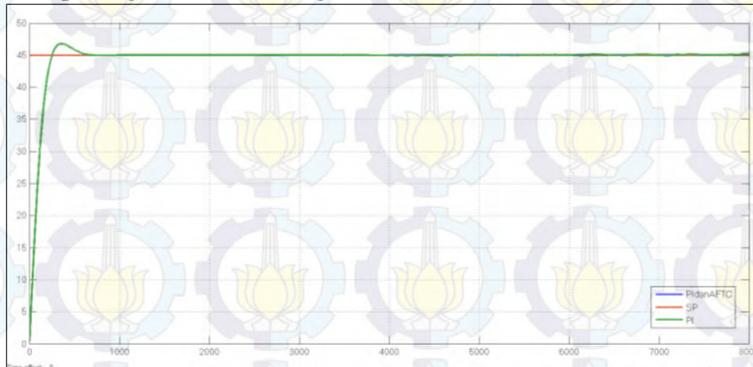
Pada gambar 4.3, gambar 4.4, gambar 4.5, dan gambar 4.6 serta penjabaran nilai parameter yang tampak pada tabel 4.3, tabel 4.4, tabel 4.5, dan tabel 4.6, terlihat bahwa respon sistem pengendalian dengan algoritma AFTC mengatasi kesalahan sensitivitas sebesar 90% hingga 75% lebih baik daripada sistem pengendalian tanpa algoritma AFTC.

Dari segi parameter, baik *maximum overshoot*, *maximum undershoot*, *error steady state*, maupun *settling time*, sistem pengendalian dengan algoritma AFTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma AFTC. Semakin kecil nilai parameter tersebut, maka semakin baik performansinya. Hal ini disebabkan oleh adanya rekonfigurasi pengendali atau aksi pengendali yang bekerja untuk mengatasi kesalahan tersebut. Maka dari itu, sistem pengendalian yang menggunakan algoritma AFTC memiliki performansi yang lebih baik dibandingkan dengan sistem pengendalian tanpa menggunakan algoritma AFTC.

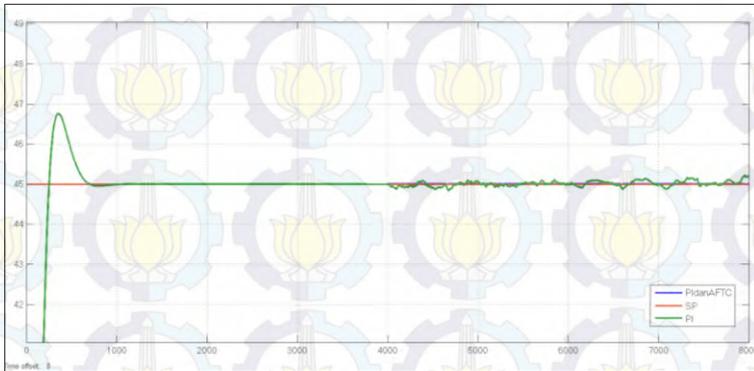
Uji sensitivitas dilakukan hingga nilai sensitivitas mencapai 75% karena dalam batas nilai sensitivitas tersebut, *maximum overshoot* yang dihasilkan oleh sistem pengendalian tanpa algoritma AFTC telah melebihi 50% dari *process variable*.

### 4.1.3 Uji Kesalahan *Noise*

*Noise* yang diberikan pada uji ini dilakukan dengan menambahkan gangguan pada hasil pengukuran yang bertipe *Gaussian noise* dengan *mean* sebesar 0 dan *variance* sebesar 15,773422%, sehingga sinyal *output* dari sistem mengandung *noise*. Nilai tersebut didapatkan dari data pengukuran sensor level pada saat sensor tersebut masih bekerja dalam keadaan optimal atau sebelum dilakukan *repairing* atau *replacement*, sehingga nilai standar deviasinya yaitu sebesar 3,9715767, kemudian dikuadratkan untuk mendapatkan nilai *variance* sebagai gangguan. Hasil respon sistem pengendalian levelnya dapat dilihat pada gambar 4.7 dan gambar 4.8.



**Gambar 4.7** Respon Sistem dengan Pemberian *Noise*



**Gambar 4.8** Perbesaran Respon Sistem dengan Pemberian *Noise*

Pada gambar 4.7 dan gambar 4.8, terlihat bahwa respon sistem pengendalian dengan algoritma AFTC mengatasi masalah *noise* lebih baik daripada sistem pengendalian tanpa algoritma AFTC. Hal ini disebabkan karena adanya rekonfigurasi pengendali atau aksi pengendali yang bekerja untuk mengatasi masalah tersebut, sedangkan respon sistem pengendalian tanpa adanya algoritma AFTC mengalami penurunan dan kenaikan yang berarti (*ripple*) yang menyebabkannya terus bergerak naik dan turun sehingga tidak sesuai dengan *set point*.

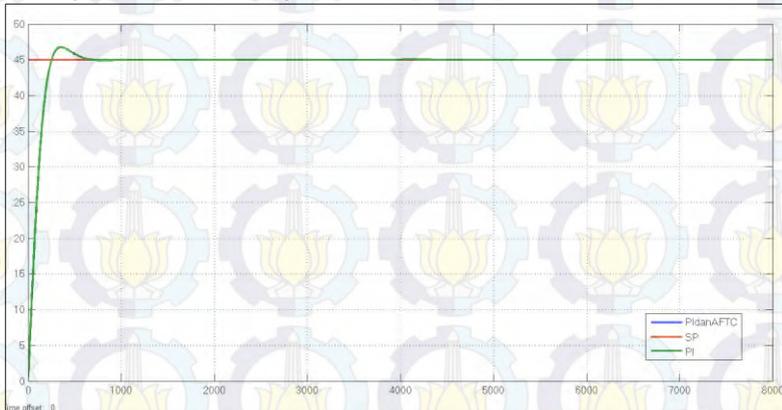
## 4.2 Uji Kesalahan Aktuator

Uji performansi sistem pengendalian level akibat kesalahan sensor dan aktuator yang kedua adalah uji kesalahan aktuator, yang terdiri dari uji kebocoran dan uji *lost of effectiveness*.

### 4.2.1 Uji Kebocoran

Kebocoran merupakan kesalahan pada sistem akibat fluida yang akan masuk ke dalam LGO *drier* tidak terumpan secara penuh. Nilai kebocoran yang diberikan pada uji ini adalah 10%, 15%, 20%, dan 30% dari nilai laju aliran LGO yang disalurkan dari *control valve* hingga masuk ke LGO *drier*, yaitu 15,52 kg/s. Hasil respon sistem pengendalian level dengan kesalahan

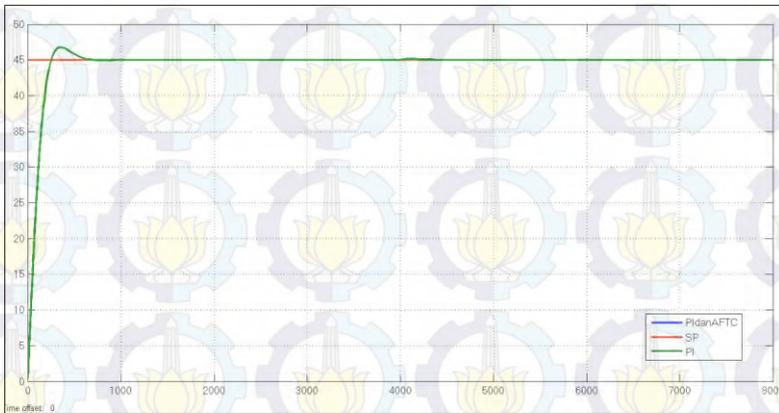
kebocoran 10% ditunjukkan pada gambar 4.9. Selanjutnya, respon sistem pengendalian level dengan kesalahan kebocoran 15%, 20%, dan 30% masing – masing ditunjukkan pada gambar 4.10, gambar 4.11, dan gambar 4.12.



**Gambar 4.9** Respon Sistem dengan Kesalahan Kebocoran 10%

**Tabel 4.7** Parameter Performansi dengan Kesalahan Kebocoran 10%

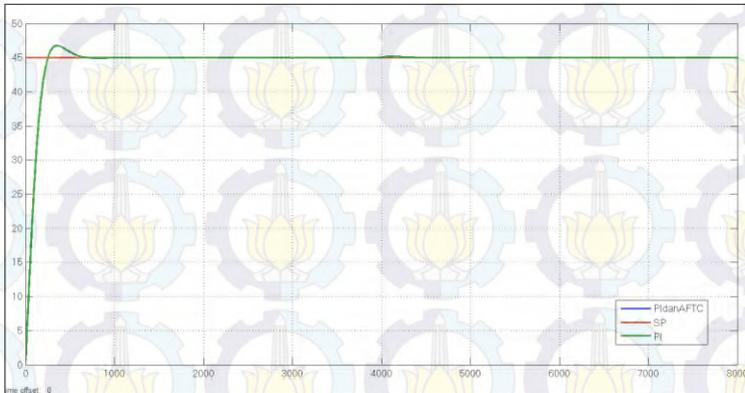
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,20%	0,04%
2	<i>Maximum Undershoot</i>	0,00975%	0,00975%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	6715 detik	4887 detik



**Gambar 4.10** Respon Sistem dengan Kesalahan Kebocoran 15%

**Tabel 4.8** Parameter Performansi dengan Kesalahan Kebocoran 15%

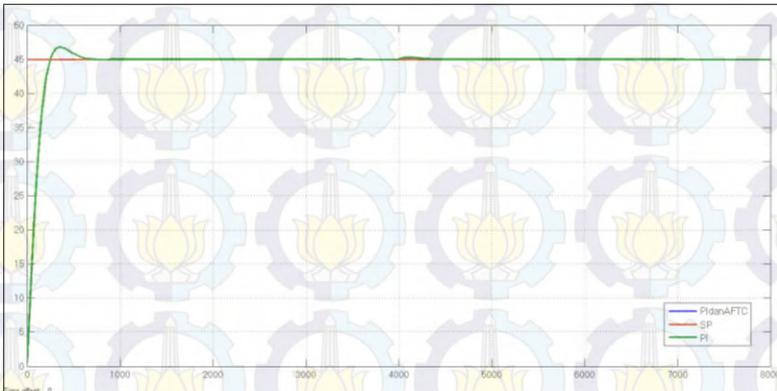
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,30%	0,06%
2	<i>Maximum Undershoot</i>	0,015%	0,015%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	6741 detik	4889 detik



**Gambar 4.11** Respon Sistem dengan Kesalahan Kebocoran 20%

**Tabel 4.9** Parameter Performansi dengan Kesalahan Kebocoran 20%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,40%	0,08%
2	<i>Maximum Undershoot</i>	0,02%	0,02%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	6767 detik	4889 detik



**Gambar 4.12** Respon Sistem dengan Kesalahan Kebocoran 30%

**Tabel 4.10** Parameter Performansi dengan Kesalahan Kebocoran 30%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,59%	0,12%
2	<i>Maximum Undershoot</i>	0,02%	0,03%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	6770 detik	5088 detik

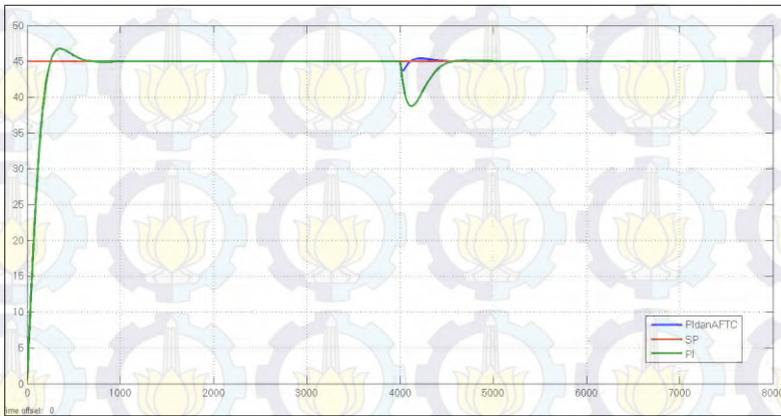
Pada gambar 4.9, gambar 4.10, gambar 4.11, dan gambar 4.12 serta penjabaran nilai parameter yang tampak pada tabel 4.7, tabel 4.8, tabel 4.9, dan tabel 4.10, terlihat bahwa respon sistem pengendalian dengan algoritma AFTC mengatasi kesalahan nilai kebocoran 10% hingga 30% lebih baik daripada sistem pengendalian tanpa algoritma AFTC. Sistem pengendalian dengan atau tanpa algoritma AFTC 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 AFTC. Dengan algoritma AFTC, sistem pengendalian tersebut dapat kembali menuju keadaan normalnya dengan cepat setelah dikenai kesalahan jika dibandingkan dengan sistem pengendalian yang tanpa algoritma AFTC.

Dari segi parameter, *maximum overshoot*, *error steady state*, maupun *settling time*, sistem pengendalian dengan algoritma AFTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma AFTC. Namun, tidak dengan nilai *maximum undershoot*, yang terlihat pada tabel 4.7, tabel 4.8, tabel 4.9, dan tabel 4.10 bahwa nilai *maximum undershoot* untuk sistem pengendalian tanpa algoritma AFTC lebih kecil dibandingkan dengan sistem pengendalian dengan algoritma AFTC. Sistem pengendalian dengan algoritma AFTC 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 AFTC lebih baik jika dibandingkan dengan sistem pengendalian tanpa algoritma AFTC.

#### **4.2.2 Uji *Lost of Effectiveness***

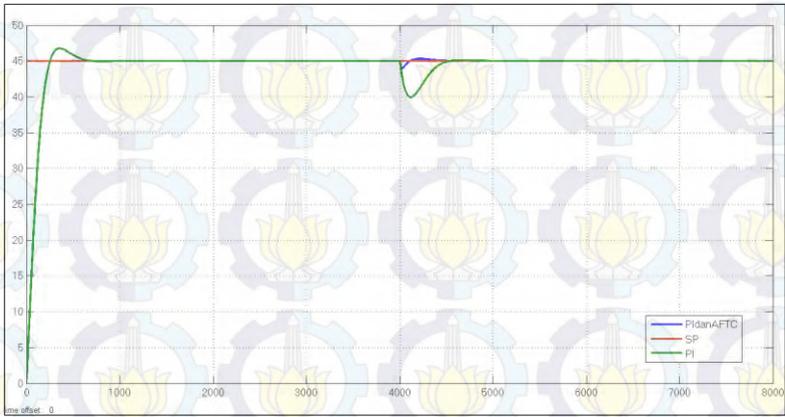
*Lost of effectiveness* adalah kesalahan pada *control valve* ketika bukaan valve terhambat oleh karat yang ditimbulkan dari material yang mengakibatkan berkurangnya sinyal pengendali yang diberikan untuk *control valve*. Nilai *lost of effectiveness* yang diberikan pada uji ini adalah 70%, 75%, 80%, 85%, dan 90% dari sinyal pengendali yang masuk pada aktuator. Hasil respon sistem pengendalian level dengan kesalahan *lost of effectiveness* 70% ditunjukkan pada gambar 4.13. Selanjutnya, respon sistem pengendalian level dengan kesalahan *lost of effectiveness* 75%, 80%, 85%, dan 90% masing – masing ditunjukkan pada gambar 4.14, gambar 4.15, gambar 4.16, dan gambar 4.17.



**Gambar 4.13** Respon Sistem dengan Kesalahan *Lost of Effectiveness* 70%

**Tabel 4.11** Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 70%

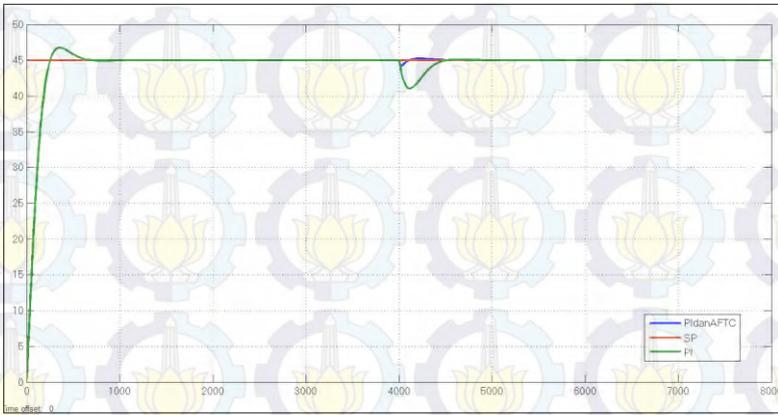
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,89%	0,89%
2	<i>Maximum Undershoot</i>	13,93%	3%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	7163 detik	5715 detik



**Gambar 4.14** Respon Sistem dengan Kesalahan *Lost of Effectiveness* 75%

**Tabel 4.12** Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 75%

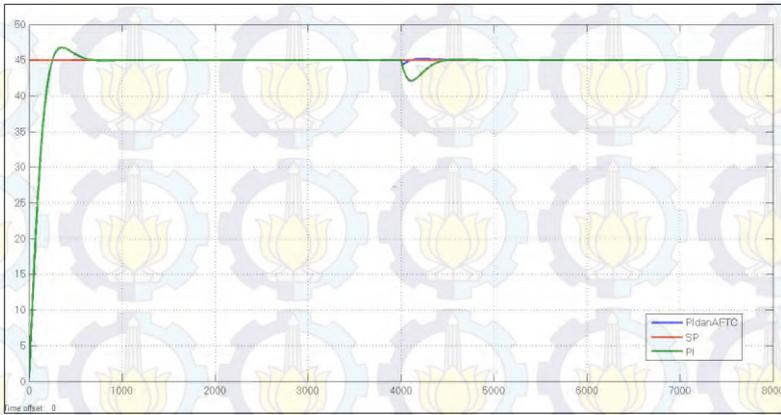
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,68%	0,68%
2	<i>Maximum Undershoot</i>	11,30%	2,39%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	7165 detik	5625 detik



**Gambar 4.15** Respon Sistem dengan Kesalahan *Lost of Effectiveness* 80%

**Tabel 4.13** Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 80%

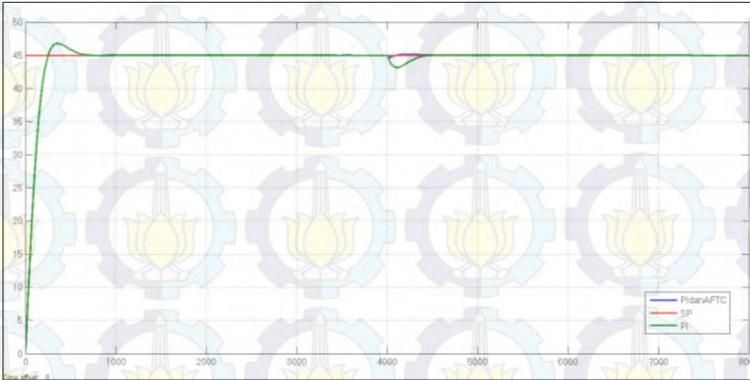
No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,51%	0,51%
2	<i>Maximum Undershoot</i>	8,81%	1,83%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	7155 detik	5549 detik



**Gambar 4.16** Respon Sistem dengan Kesalahan *Lost of Effectiveness* 85%

**Tabel 4.14** Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 85%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,36%	0,36%
2	<i>Maximum Undershoot</i>	6,45%	1,31%
3	<i>Error Steady State</i>	0%	0%
4	<i>Settling Time</i>	7147 detik	5483 detik



**Gambar 4.17** Respon Sistem dengan Kesalahan *Lost of Effectiveness* 90%

**Tabel 4.15** Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 90%

No	Parameter	PI tanpa AFTC	PI dengan AFTC
1	<i>Maximum Overshoot</i>	0,11%	0%
2	<i>Maximum Undershoot</i>	4,2%	0,9%
3	<i>Error Steady State</i>	0%	0%
4	<i>Setling Time</i>	6372 detik	5102 detik

Pada gambar 4.13, gambar 4.14, gambar 4.15, gambar 4.16, dan gambar 4.17 serta penjabaran nilai parameter yang tampak pada tabel 4.11, tabel 4.12, tabel 4.13, tabel 4.14, dan tabel 4.15 terlihat bahwa respon sistem pengendalian dengan algoritma AFTC mengatasi kesalahan nilai *lost of effectiveness* 70% hingga 90% lebih baik daripada sistem pengendalian tanpa algoritma AFTC. Sistem pengendalian dengan ataupun tanpa algoritma AFTC dapat mengembalikan nilai *process variable* menuju ke titik *set point*, namun dengan perbedaan waktu. Perbedaan waktu inilah

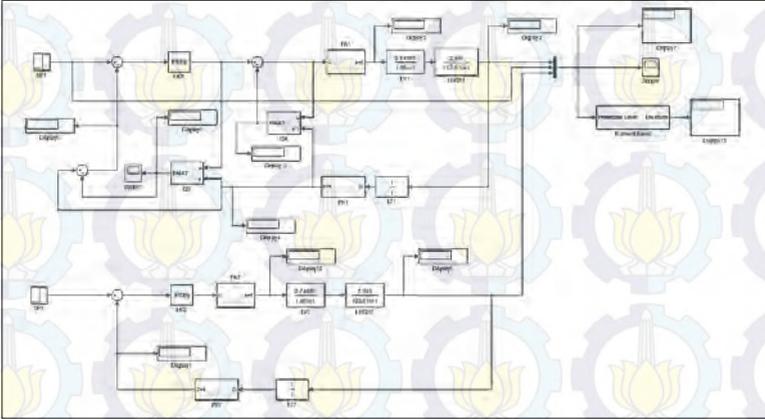
yang menunjukkan keunggulan sistem pengendalian dengan algoritma AFTC. Dengan algoritma AFTC, sistem pengendalian tersebut dapat kembali menuju keadaan normalnya dengan cepat setelah dikenai kesalahan jika dibandingkan dengan sistem pengendalian yang tanpa algoritma AFTC.

Dari segi parameter, baik *maximum undershoot*, *error steady state*, maupun *settling time*, sistem pengendalian dengan algoritma AFTC memiliki nilai lebih kecil daripada sistem pengendalian tanpa algoritma AFTC. Namun, tidak dengan nilai *maximum overshoot*, yang terlihat pada tabel 4.11, tabel 4.12, tabel 4.13, tabel 4.14, dan tabel 4.15 bahwa nilai *maximum overshoot* untuk sistem pengendalian tanpa algoritma AFTC lebih kecil dibandingkan dengan sistem pengendalian dengan algoritma AFTC. Sistem pengendalian dengan algoritma AFTC 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 AFTC, sistem pengendalian dengan algoritma AFTC tetap memiliki performansi yang baik dalam ranah *safety*. Hal ini karena kerja sistem pengendalian yang baik adalah ditentukan oleh besarnya *error* yang dimiliki.

### 4.3 Uji kesalahan Sensor dan Aktuator

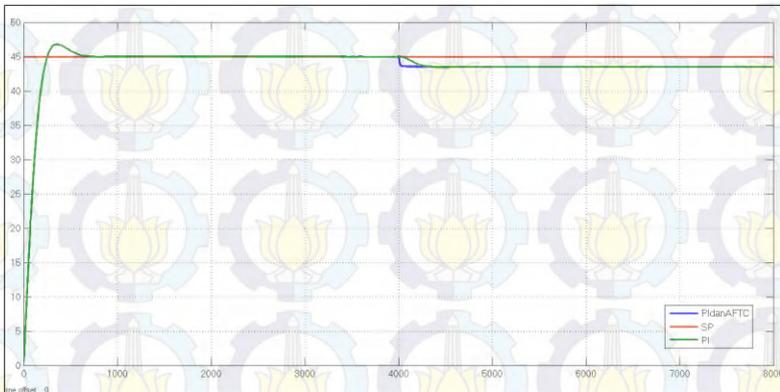
Uji performansi sistem pengendalian level akibat kesalahan pada sensor dan aktuator yang ketiga adalah uji kesalahan sensor dan aktuator secara bersamaan. Kesalahan sensor dan aktuator diberikan secara bersamaan pada detik ke - 4000. Jenis kesalahan yang diberikan adalah acak, baik untuk kesalahan sensor maupun juga kesalahan aktuator. Pada laporan tugas akhir ini, uji kesalahan sensor dan aktuator dilakukan satu kali dengan pemberian nilai bias sebesar +1,5% untuk kesalahan sensor dan pemberian nilai kebocoran 50% untuk kesalahan aktuator.

Gambar 4.18 merupakan gambaran *simulink* MATLAB secara *general* untuk uji sensor dan aktuator secara bersamaan.



**Gambar 4.18** Struktur Sistem Pengendalian dengan *Observer* Sensor dan Aktuator dalam Satu *Loop*

Setelah dilakukan uji sensor dan aktuator secara bersamaan, perancangan sistem pengendalian AFTC diketahui tidak bekerja secara maksimal karena *input* dari *observer* aktuator yang merupakan hasil *output* sensor yang telah mengalami kesalahan, sehingga apabila kesalahan sensor dan aktuator terjadi dalam waktu yang sama, proses rekonfigurasi harus berjalan bersamaan, dimana seharusnya proses rekonfigurasi kesalahan sensor harus berjalan terlebih dahulu baru dilanjutkan dengan proses rekonfigurasi aktuator. Gambar 4.19 merupakan respon sistem pengendalian level dengan kesalahan sensor dan aktuator.



**Gambar 4.19** Respon Sistem dengan Kesalahan Sensor dan Aktuator



*Halaman ini sengaja dikosongkan*

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

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

- Perancangan sistem *observer* akibat kesalahan sensor dan aktuator telah berhasil dilakukan. Parameter keberhasilannya adalah ketika nilai keluaran *observer* sesuai dengan nilai uji kesalahan yang diberikan.
- Pengujian sensor dan aktuator memberikan hasil bahwa sistem pengendalian level dengan algoritma AFTC menghasilkan performansi lebih baik dibandingkan dengan sistem pengendalian level tanpa algoritma AFTC. Buktinya, pada uji sensitivitas 90%, sistem pengendalian dengan algoritma AFTC memiliki nilai *maximum overshoot* sebesar 0,06%, *maximum undershoot* sebesar 0%, *error steady state* sebesar 0 %, dan *settling time* sebesar 4548 detik dan pada uji kebocoran 50% memiliki nilai *maximum overshoot* sebesar 0,21%, *maximum undershoot* sebesar 0,05%, *error steady state* sebesar 0 %, dan *settling time* sebesar 5145 detik.
- Pengujian terhadap sensor dan aktuator memberikan hasil bahwa sistem pengendalian tidak dapat bekerja karena proses rekonfigurasi kontrolnya yang berlangsung secara bersamaan.

### 5.2 Saran

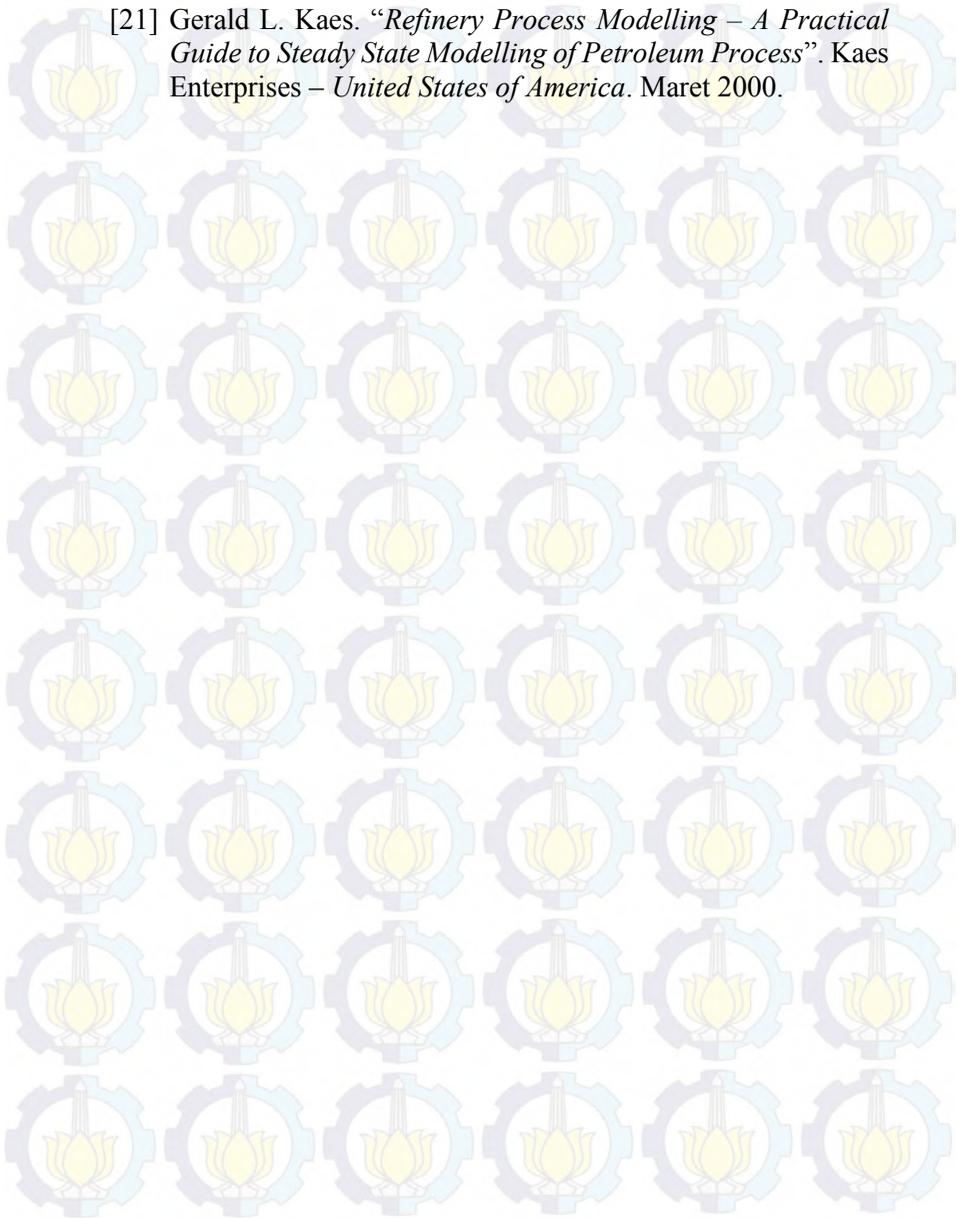
Pada tugas akhir ini, terdapat saran yang dapat dilakukan untuk peningkatan kualitas tugas akhir selanjutnya, yaitu perancangan *observer* tunggal sehingga perancangan AFTC menjadi lebih efisien dan bekerja optimal untuk mengatasi dua kesalahan komponen yaitu sensor dan aktuator dalam satu *loop* pengendalian.

## DAFTAR PUSTAKA

- [1] PT. Pertamina (Persero) RU IV Cilacap. “Sekilas Pengenalan Kilang (*Refinery Unit*) RU IV Cilacap”. Cilacap, Jawa Tengah. 7 Mei 2012.
- [2] PT. Pertamina (Persero) RU IV Cilacap. “*Fluor Daniel Engineers & Constructors, LTD. “Pertamina Cilacap Debottlenecking Project, Java, Indonesia – Description of the Process Flow”*”. Cilacap, Jawa Tengah. Juni, 1997.
- [3] PT. Pertamina (Persero) RU IV Cilacap. “*Scheme of Gasoil Stripper/Drier unit 1100”*”. 7 Oktober 2014.
- [4] 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.
- [5] 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.
- [6] Ju Chang Huang, Chii Shang. “*Air Stripping”*”. Springer Journal No. 978 – 1 – 58829 – 361 – 9. *Advanced Physicochemical Treatment Processes*. 2006.
- [7] R. W. Ellerbe. “*Steam Distillation/Stripping”*”. *Handbook of Separation Techniques for Chemical Engineers*. 2nd ed. Ed. Philip A. Schweitzer. New York: McGraw – Hill, Inc., 1988.
- [8] Timothy M. Z. “*Designing Steam Stripping Columns for Wastewater”*”. *Chemical Engineering* May 2008: 52 – 56.
- [9] Vu Trieu M., John P. “*Modelling and Control Simulation for a Condensate Distillation Column”*”. *Papua New Guinea University of Technology (UNITECH), Lae, Papua, New Guinea*. 2012.
- [10] *Houston Refining Company. “Light Gas Oil Material Safety Data Sheet.”* Houston, Texas. MSDS No. AP0883 *revision date* 6 Januari 2011.

- [11] PT. Pertamina (Persero) RU IV Cilacap. "Orientasi Khusus PT. Pertamina (Persero) RU IV Cilacap – Deskripsi Proses *Plant Light Gas Oil Stripper* 11C4".
- [12] F. Gunterus. "Falsafah Dasar : Sistem Pengendalian Proses, Elex Media Komputindo". Jakarta. 1997.
- [13] Universitas Sumatera Utara. "Pengenalan Alat Ukur Level". Sumatera Utara. 11 Februari 2015.
- [14] F, Agus Handrian. "Perancangan Sistem Pengendalian Temperatur pada *Reboiler Metanol Recovery* menggunakan *Fuzzy Gain Scheduling* PID di PT. Eterindo Nusa Graha Gresik." Institut Teknologi Sepuluh Nopember Surabaya. 2011.
- [15] AspenTech. *Official Website of AspenTech Company*. (www.aspentech.com). Aspen Technology. 1994 – 2005.
- [16] J. E. Arbogast, D. J. Cooper. R. C. Rice. "*Graphical Technique for Modelling Integrating (Non – Self – Regulating) Processes without Steady State Process Data*". Department of Chemical, Materials, dan Biomolecular Engineering, University of Connecticut, Storrs, Connecticut, USA. 2007.
- [17] H. Noura, D. Theillioli, J. Christophe P., A. Chamseddine. "*Fault – Tolerant Control Systems, Design and Practical Application*". AIC (Advances Industrial Control), Springer. New York. 2009.
- [18] S. Boyd, L. El Ghaoui, Eric Feron, V. Balakrishnan. "*Linear Matrix Inequalities in System and Control Theory*". Siam. 1994. Pp. 7 – 8.
- [19] PT. Pertamina (Persero) RU IV Cilacap. "*Data/Requisition Sheet for Pressure Vessel (Columns, Reactor, Accumulators, etc) – Equipment No. 11C4 dan 11C5*".
- [20] PT. Pertamina (Persero) RU IV Cilacap. "*Fluor Daniel Engineers & Constructors, LTD. "Process Flow Scheme Fractionating Section, (DN 429400 – 4 – 11 – 002)"*". Cilacap, Jawa Tengah. Desember, 2000.

[21] Gerald L. Kaes. “*Refinery Process Modelling – A Practical Guide to Steady State Modelling of Petroleum Process*”. Kaes Enterprises – *United States of America*. Maret 2000.

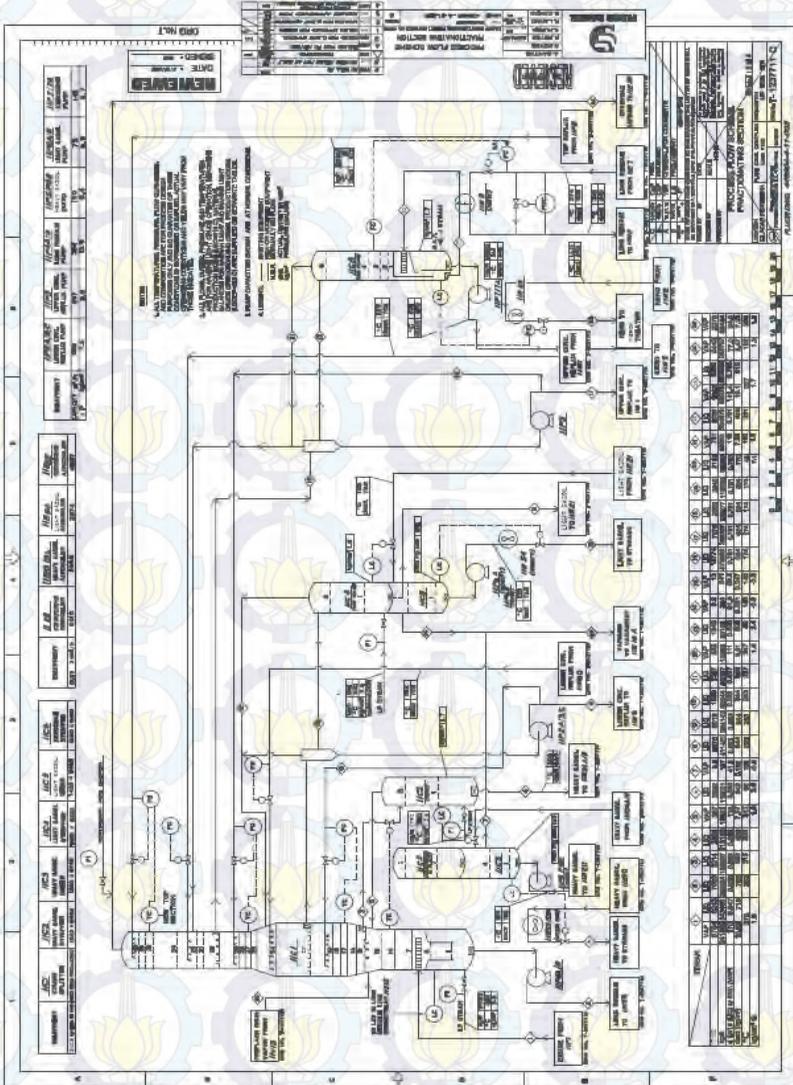




*Halaman ini sengaja dikosongkan*

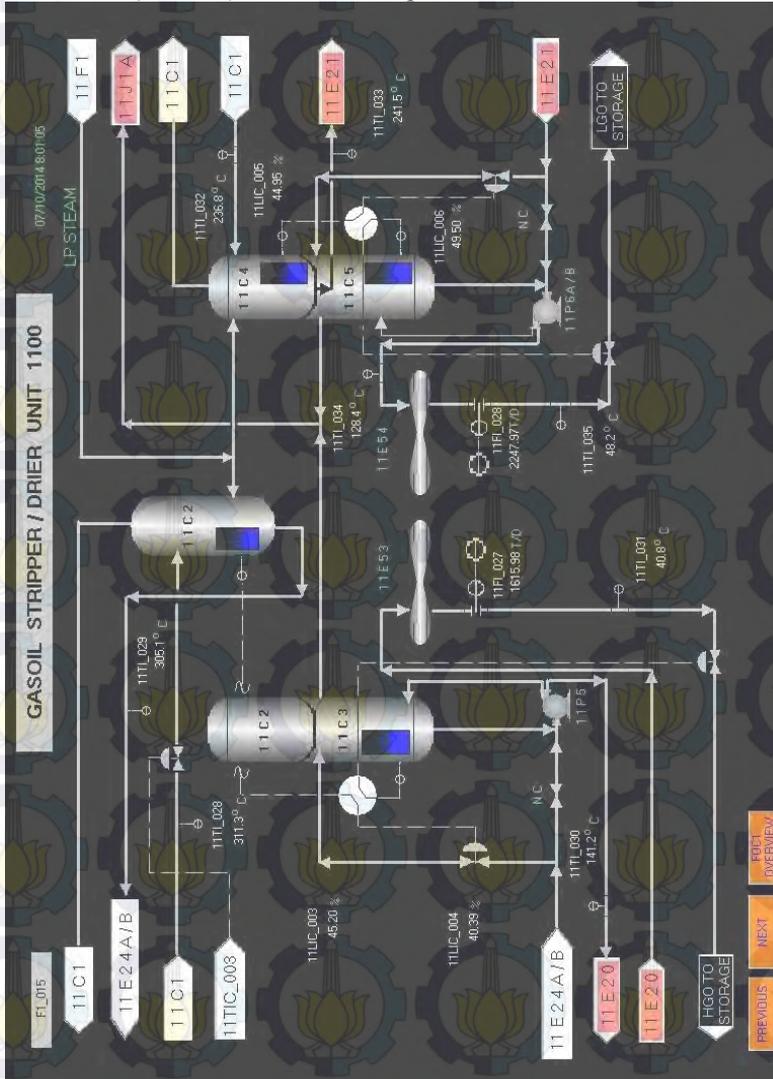
# LAMPIRAN A

## A.1 Process Flow Diagram Plant Light Gas Oil Stripper 11C4 Unit Fuel Oil Complex I PT. Pertamina (Persero) RU IV Cilacap



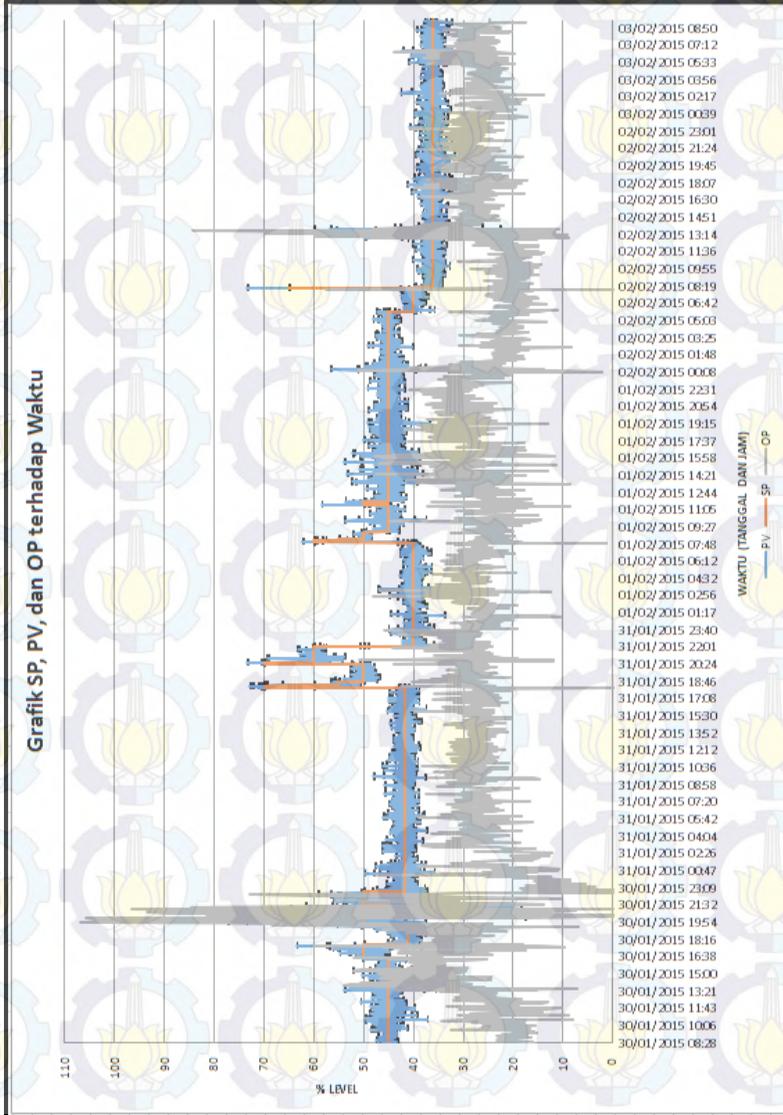
## LAMPIRAN B

**B.1** *Process Flow Distributed Control System Unit Control Room  
Plant Light Gas Oil Stripper 11C4 Unit Fuel Oil Complex I PT.  
Pertamina (Persero) RU IV Cilacap*



## LAMPIRAN C

### C.1 Grafik Level LGO Stripper 11C4 Unit FOC I PT. Pertamina (Persero) RU IV Cilacap 30 Januari hingga 3 Februari 2015



**C.2** Data Pengukuran Level LGO *Stripper* 11C4 Unit FOC I PT. Pertamina (Persero) RU IV Cilacap 30 Januari hingga 3 Februari 2015

<b>Hari, Tanggal</b>	<b>PV (%)</b>	<b>SP (%)</b>	<b>OP</b>
1/30/2015 8:28	43.24798	45.00000	15.95843
1/30/2015 8:30	44.68901	45.00000	16.0173
1/30/2015 9:00	47.10081	45.00000	20.02659
1/30/2015 9:32	43.67236	45.00000	15.81608
1/30/2015 10:00	44.03285	45.00000	23.2258
1/30/2015 10:30	49.60324	45.00000	30.04692
1/30/2015 11:02	40.44145	45.00000	31.20593
1/30/2015 11:31	44.65125	45.00000	21.19592
1/30/2015 12:00	43.03118	45.00000	18.8827
1/30/2015 12:30	45.50963	45.00000	18.20729
1/30/2015 13:03	43.21824	45.00000	25.188
1/30/2015 13:30	45.55249	45.00000	7.189885
1/30/2015 14:00	41.21606	45.00000	38.22827
1/30/2015 14:31	45.46351	45.00000	30.0794
1/30/2015 15:00	42.44369	45.00000	31.65604
1/30/2015 15:31	47.64016	45.00000	38.70798
1/30/2015 16:01	45.70566	45.00000	37.48136
1/30/2015 16:30	47.61932	50.00000	49.91796
1/30/2015 17:01	51.75171	50.00000	42.94632
1/30/2015 17:31	44.34857	50.00000	9.769672
1/30/2015 18:00	39.13464	41.25656	28.1342
1/30/2015 18:30	39.7168	41.25656	28.35814
1/30/2015 19:00	36.60535	41.25656	28.94897
1/30/2015 17:31	44.34857	50.00000	9.769672
1/30/2015 20:01	46.03369	50.00000	58.59475
1/30/2015 20:33	30.95215	50.00000	18.99977

<b>1/30/2015 20:34</b>	31.72854	50.00000	0
<b>1/30/2015 20:36</b>	35.22246	50.00000	7.999029
<b>1/30/2015 21:02</b>	60.51544	50.00000	53.39283
<b>1/30/2015 21:30</b>	41.4445	45.00000	24.78723
<b>1/30/2015 22:02</b>	53.12965	50.00000	54.70282
<b>1/30/2015 22:32</b>	59.0525	50.00000	55.9293
<b>1/30/2015 22:52</b>	47.34979	41.84365	10.91193
<b>1/30/2015 22:54</b>	40.63461	41.84365	0
<b>1/30/2015 22:56</b>	39.36018	41.84365	1.957283
<b>1/30/2015 23:00</b>	44.99392	41.84365	13.24582
<b>1/30/2015 23:31</b>	39.44188	41.84365	11.82164
<b>1/31/2015 0:02</b>	41.61416	41.84365	15.05675
<b>1/31/2015 0:30</b>	42.1037	41.84365	17.36653
<b>1/31/2015 1:02</b>	42.40252	41.84365	17.85707
<b>1/31/2015 1:31</b>	43.10671	41.84365	27.41411
<b>1/31/2015 2:01</b>	40.5732	41.84365	16.19378
<b>1/31/2015 2:33</b>	41.74411	41.84365	23.18513
<b>1/31/2015 3:00</b>	42.5717	41.84365	29.07384
<b>1/31/2015 3:30</b>	42.05341	41.84365	19.43202
<b>1/31/2015 4:01</b>	39.91307	41.84365	23.68871
<b>1/31/2015 4:31</b>	43.65301	41.84365	18.17614
<b>1/31/2015 5:01</b>	42.14008	41.84365	27.15887
<b>1/31/2015 5:30</b>	43.26144	41.84365	24.11183
<b>1/31/2015 6:02</b>	37.88979	41.84365	26.8034
<b>1/31/2015 6:30</b>	40.47596	41.84365	27.3222
<b>1/31/2015 7:00</b>	40.25667	41.84365	31.06695
<b>1/31/2015 7:30</b>	42.91673	41.84365	27.50696
<b>1/31/2015 8:00</b>	43.07964	41.84365	37.41692
<b>1/31/2015 8:32</b>	40.48857	41.84365	25.32587
<b>1/31/2015 9:02</b>	40.88175	41.84365	25.95864

<b>1/31/2015 9:31</b>	42.94236	41.84365	14.66456
<b>1/31/2015 10:02</b>	41.29675	41.84365	25.22938
<b>1/31/2015 10:30</b>	41.30342	41.84365	21.01677
<b>1/31/2015 11:00</b>	39.95121	41.84365	25.19771
<b>1/31/2015 11:30</b>	44.02603	41.84365	28.42429
<b>1/31/2015 12:00</b>	41.70798	41.84365	29.25936
<b>1/31/2015 12:30</b>	42.16138	41.84365	30.19102
<b>1/31/2015 13:00</b>	42.29789	41.84365	28.18286
<b>1/31/2015 13:30</b>	42.99622	41.84365	23.53839
<b>1/31/2015 14:00</b>	40.63652	41.84365	27.14445
<b>1/31/2015 14:30</b>	42.55594	41.84365	26.12584
<b>1/31/2015 15:01</b>	43.59238	41.84365	29.73313
<b>1/31/2015 15:09</b>	42.61323	41.84365	26.41407
<b>1/31/2015 15:30</b>	42.96981	41.84365	24.72638
<b>1/31/2015 16:01</b>	44.85217	41.84365	23.95378
<b>1/31/2015 16:30</b>	40.92489	41.84365	26.86663
<b>1/31/2015 17:01</b>	40.90324	41.84365	32.89227
<b>1/31/2015 17:30</b>	42.25483	41.84365	23.0695
<b>1/31/2015 18:01</b>	39.69872	41.84365	24.00541
<b>1/31/2015 18:31</b>	56.13207	55.00000	30.90648
<b>1/31/2015 19:01</b>	51.62096	50.00000	28.37958
<b>1/31/2015 19:30</b>	50.58674	50.00000	27.08075
<b>1/31/2015 20:02</b>	50.89206	50.00000	25.19404
<b>1/31/2015 20:30</b>	70.45886	70.00000	25.7911
<b>1/31/2015 21:00</b>	56.18816	60.00000	22.32997
<b>1/31/2015 21:32</b>	60.32642	60.00000	26.82865
<b>1/31/2015 21:34</b>	59.93019	60.00000	30.55413
<b>1/31/2015 22:01</b>	50.73293	50.00000	27.62129
<b>1/31/2015 22:30</b>	40.12352	40.00000	29.95759
<b>1/31/2015 23:00</b>	40.28636	40.00000	30.42506

<b>1/31/2015 23:33</b>	38.36852	40.00000	45.78001
<b>1/31/2015 23:41</b>	43.30826	40.00000	25.47322
<b>2/1/2015 0:01</b>	40.20308	40.00000	26.86871
<b>2/1/2015 0:31</b>	38.75057	40.00000	24.78163
<b>2/1/2015 1:00</b>	33.75463	40.00000	28.35727
<b>2/1/2015 1:30</b>	40.09604	40.00000	24.81909
<b>2/1/2015 2:00</b>	38.45107	40.00000	19.13735
<b>2/1/2015 2:32</b>	38.93129	40.00000	25.95257
<b>2/1/2015 3:00</b>	34.94457	40.00000	24.50707
<b>2/1/2015 3:30</b>	42.88539	40.00000	23.96712
<b>2/1/2015 4:00</b>	41.02032	40.00000	29.19805
<b>2/1/2015 4:31</b>	41.22681	40.00000	26.37575
<b>2/1/2015 5:02</b>	42.11469	40.00000	26.21188
<b>2/1/2015 5:30</b>	38.11515	40.00000	33.38078
<b>2/1/2015 6:02</b>	41.62602	40.00000	31.70201
<b>2/1/2015 6:34</b>	38.38735	40.00000	24.63828
<b>2/1/2015 7:03</b>	42.59632	40.00000	30.05468
<b>2/1/2015 7:30</b>	40.22195	40.00000	27.34162
<b>2/1/2015 8:02</b>	58.45321	60.00000	28.95899
<b>2/1/2015 8:32</b>	48.42328	50.00000	20.06712
<b>2/1/2015 9:00</b>	45.5505	45.00000	21.56391
<b>2/1/2015 9:32</b>	46.05555	45.00000	23.01022
<b>2/1/2015 10:01</b>	53.75226	45.00000	15.59228
<b>2/1/2015 10:34</b>	44.16527	45.00000	19.9747
<b>2/1/2015 11:00</b>	46.99426	45.00000	19.60699
<b>2/1/2015 11:30</b>	48.92345	50.00000	29.78406
<b>2/1/2015 12:01</b>	45.65853	45.00000	26.28323
<b>2/1/2015 12:32</b>	42.80043	45.00000	19.61331
<b>2/1/2015 13:01</b>	46.80104	45.00000	30.87142
<b>2/1/2015 13:31</b>	42.77602	45.00000	10.3195

<b>2/1/2015 14:00</b>	47.06852	45.00000	20.65663
<b>2/1/2015 14:30</b>	47.61426	45.00000	32.27729
<b>2/1/2015 15:00</b>	43.05	45.00000	29.58174
<b>2/1/2015 15:30</b>	50.4854	45.00000	43.58522
<b>2/1/2015 16:00</b>	47.35051	45.00000	42.05299
<b>2/1/2015 16:30</b>	39.06511	45.00000	30.39443
<b>2/1/2015 17:00</b>	46.87194	45.00000	20.63587
<b>2/1/2015 17:32</b>	47.88435	45.00000	35.91001
<b>2/1/2015 18:01</b>	47.36078	45.00000	31.20516
<b>2/1/2015 18:32</b>	47.2523	45.00000	25.42239
<b>2/1/2015 19:01</b>	44.14742	45.00000	34.9183
<b>2/1/2015 19:31</b>	41.70757	45.00000	34.89612
<b>2/1/2015 20:00</b>	45.01928	45.00000	31.0188
<b>2/1/2015 20:30</b>	46.97844	45.00000	24.50681
<b>2/1/2015 21:01</b>	47.96803	45.00000	34.7498
<b>2/1/2015 21:32</b>	45.21941	45.00000	31.40902
<b>2/1/2015 22:01</b>	44.4911	45.00000	38.09869
<b>2/1/2015 22:31</b>	43.68209	45.00000	33.57202
<b>2/1/2015 23:00</b>	47.09224	45.00000	27.73507
<b>2/1/2015 23:30</b>	46.45401	45.00000	25.37203
<b>2/2/2015 0:04</b>	45.25809	45.00000	26.60676
<b>2/2/2015 0:31</b>	37.73884	45.00000	25.22724
<b>2/2/2015 1:01</b>	44.34435	45.00000	29.15293
<b>2/2/2015 1:32</b>	44.06337	45.00000	22.98127
<b>2/2/2015 2:00</b>	44.20438	45.00000	23.28588
<b>2/2/2015 2:30</b>	43.22291	45.00000	18.45351
<b>2/2/2015 3:02</b>	46.56021	45.00000	24.39434
<b>2/2/2015 3:31</b>	47.24715	45.00000	25.98096
<b>2/2/2015 4:00</b>	45.22331	45.00000	20.53991
<b>2/2/2015 4:31</b>	43.92321	45.00000	21.27354

<b>2/2/2015 5:00</b>	42.70095	45.00000	22.96657
<b>2/2/2015 5:30</b>	43.62788	45.00000	20.33573
<b>2/2/2015 6:00</b>	35.87655	40.00000	14.73239
<b>2/2/2015 6:32</b>	38.23222	40.00000	19.18682
<b>2/2/2015 7:00</b>	40.74046	40.00000	23.86378
<b>2/2/2015 7:32</b>	41.15954	40.00000	23.01082
<b>2/2/2015 7:55</b>	41.06127	40.00000	16.84602
<b>2/2/2015 7:58</b>	39.5404	40.00000	0
<b>2/2/2015 8:00</b>	39.6856	40.00000	57.48761
<b>2/2/2015 8:32</b>	37.7775	36.20171	20.89103
<b>2/2/2015 9:00</b>	34.54225	36.20171	21.98141
<b>2/2/2015 9:30</b>	36.54822	36.20171	21.82916
<b>2/2/2015 10:01</b>	34.94922	36.20171	18.88443
<b>2/2/2015 10:32</b>	35.18045	36.20171	20.72875
<b>2/2/2015 11:00</b>	35.75583	36.20171	22.38625
<b>2/2/2015 11:30</b>	38.4497	36.20171	13.84327
<b>2/2/2015 12:00</b>	36.92346	36.20171	22.73111
<b>2/2/2015 12:30</b>	35.98035	36.20171	24.90004
<b>2/2/2015 13:00</b>	33.9024	36.20171	23.33827
<b>2/2/2015 13:30</b>	55.44417	36.20171	84.48648
<b>2/2/2015 14:01</b>	39.1716	36.20171	36.41708
<b>2/2/2015 14:34</b>	35.74438	36.20171	23.15257
<b>2/2/2015 15:03</b>	36.28294	36.20171	29.98001
<b>2/2/2015 15:31</b>	34.70438	36.20171	29.22527
<b>2/2/2015 16:01</b>	36.21132	36.20171	28.80656
<b>2/2/2015 16:30</b>	36.01049	36.20171	29.86621
<b>2/2/2015 17:00</b>	35.39931	36.20171	20.71203
<b>2/2/2015 17:30</b>	34.74272	36.20171	29.73665
<b>2/2/2015 18:01</b>	33.27121	36.20171	22.57742
<b>2/2/2015 18:31</b>	39.94167	36.20171	28.5298

<b>2/2/2015 19:02</b>	36.1789	36.20171	28.50307
<b>2/2/2015 19:30</b>	39.46936	36.20171	35.46059
<b>2/2/2015 20:00</b>	34.77886	36.20171	24.02111
<b>2/2/2015 20:30</b>	38.298	36.20171	28.57116
<b>2/2/2015 21:00</b>	37.71056	36.20171	25.98121
<b>2/2/2015 21:33</b>	37.62995	36.20171	26.15363
<b>2/2/2015 22:02</b>	34.69677	36.20171	29.04674
<b>2/2/2015 22:31</b>	36.28546	36.20171	29.62433
<b>2/2/2015 23:00</b>	37.65977	36.20171	23.50656
<b>2/2/2015 23:31</b>	33.9374	36.20171	28.33876
<b>2/3/2015 0:01</b>	36.37252	36.20171	32.21952
<b>2/3/2015 0:31</b>	36.33774	36.20171	26.84769
<b>2/3/2015 1:01</b>	36.44149	36.20171	19.74565
<b>2/3/2015 1:32</b>	35.76744	36.20171	18.61351
<b>2/3/2015 2:31</b>	34.61625	36.20171	33.48749
<b>2/3/2015 3:00</b>	38.29894	36.20171	31.53861
<b>2/3/2015 3:32</b>	37.91594	36.20171	28.57655
<b>2/3/2015 4:03</b>	34.25987	36.20171	29.82421
<b>2/3/2015 4:32</b>	37.15272	36.20171	22.51632
<b>2/3/2015 5:00</b>	35.62849	36.20171	27.35303
<b>2/3/2015 5:31</b>	41.0016	36.20171	24.87702
<b>2/3/2015 6:01</b>	36.14386	36.20171	30.58259
<b>2/3/2015 6:30</b>	34.1315	36.20171	37.48332
<b>2/3/2015 7:01</b>	36.00073	36.20171	25.88638
<b>2/3/2015 7:30</b>	37.85346	36.20171	22.76487
<b>2/3/2015 8:00</b>	35.12028	36.20171	26.91603
<b>2/3/2015 8:30</b>	35.11517	36.20171	31.25181
<b>2/3/2015 9:00</b>	38.00706	36.20171	28.08931

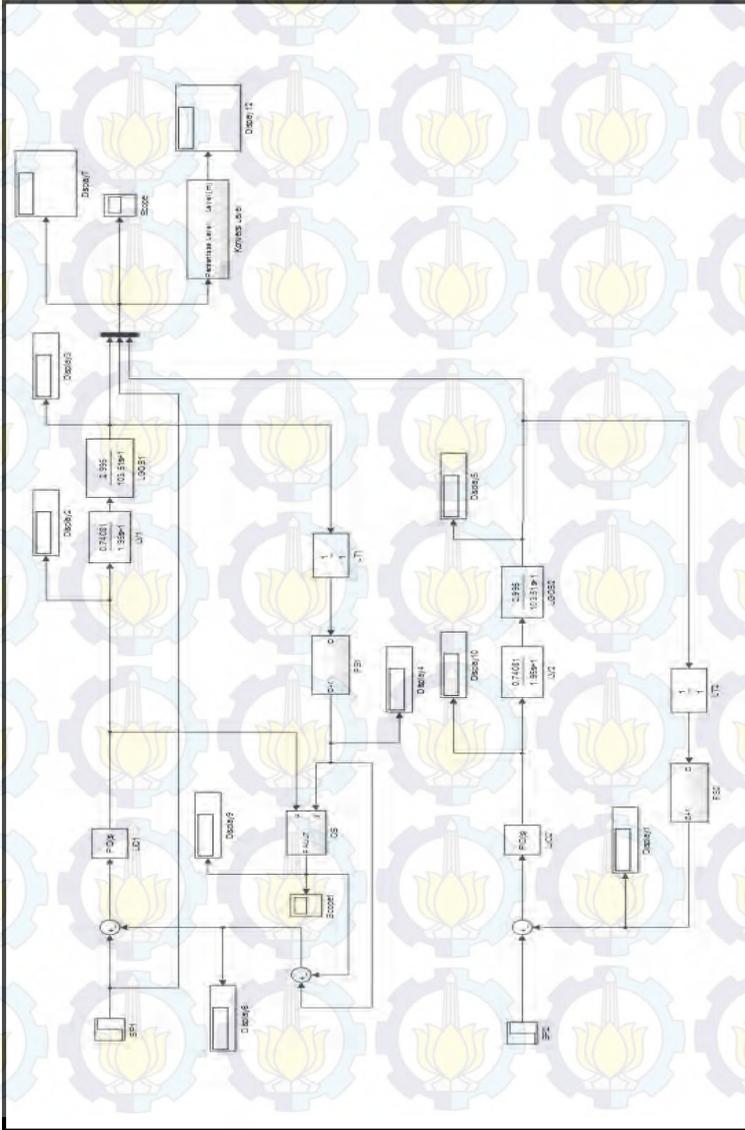
**C.3** Data Pengambilan Parameter FOPDT dari Simulasi HYSYS dengan *Opening Valve* 20% dan 30%.

<b>Detik</b>	<b>SP</b>	<b>PV</b>	<b>OP</b>
<b>0</b>	44.99994	44.9999414	20.95
<b>20</b>	44.99994	44.9999414	20.95
<b>40</b>	44.99994	44.9999414	20.95
<b>60</b>	44.99994	44.9999414	20.95
<b>80</b>	42.41096	42.41096314	30
<b>100</b>	38.3356	38.33560367	30
<b>120</b>	34.83139	34.83138777	30
<b>140</b>	31.87876	31.87875784	30
<b>160</b>	29.40138	29.40137692	30
<b>180</b>	27.33377	27.33377128	30
<b>200</b>	25.61658	25.61658017	30
<b>220</b>	24.19664	24.19663827	30
<b>240</b>	23.02702	23.0270177	30
<b>260</b>	22.0671	22.06710249	30
<b>280</b>	21.28494	21.28494271	30
<b>300</b>	20.64586	20.64586436	30
<b>320</b>	20.12479	20.12478788	30
<b>340</b>	19.70053	19.70053393	30
<b>360</b>	19.35565	19.35565052	30
<b>380</b>	19.07567	19.07566543	30
<b>400</b>	18.84863	18.84862679	30
<b>420</b>	18.6647	18.66470206	30
<b>440</b>	18.51583	18.51582702	30
<b>460</b>	18.39541	18.39541035	30
<b>480</b>	18.29807	18.29807291	30

<b>500</b>	18.21943	18.21943194	30
<b>520</b>	18.15593	18.15592702	30
<b>540</b>	18.10467	18.10466561	30
<b>560</b>	18.0633	18.06330243	30
<b>580</b>	18.02994	18.02993709	30
<b>600</b>	18.00303	18.00303035	30
<b>620</b>	17.98134	17.98133798	30
<b>640</b>	17.96385	17.9638534	30
<b>660</b>	17.94976	17.94976343	30
<b>680</b>	17.93841	17.93841457	30
<b>700</b>	17.92928	17.92927574	30
<b>720</b>	17.92191	17.9219121	30
<b>740</b>	17.91601	17.91601438	30
<b>760</b>	17.91127	17.91127018	30
<b>780</b>	17.90744	17.90744147	30
<b>800</b>	17.90435	17.90435177	30
<b>820</b>	17.90186	17.90185855	30
<b>840</b>	17.89985	17.89984597	30
<b>860</b>	17.89822	17.89822137	30
<b>880</b>	17.89691	17.8969096	30

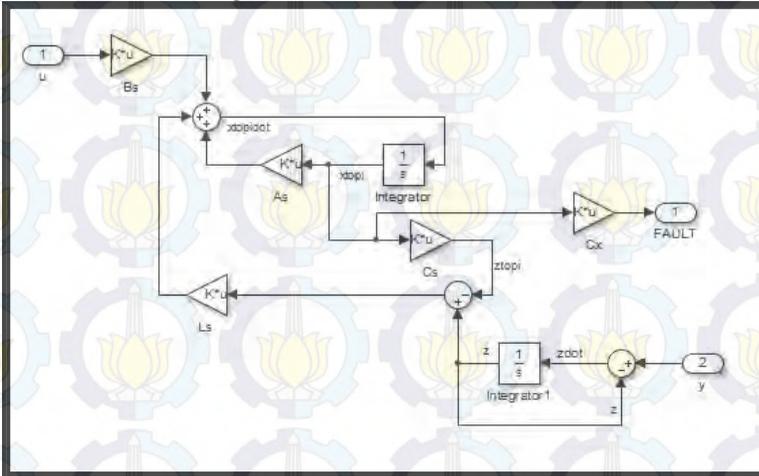
## LAMPIRAN D

### D.1 Hasil Perancangan *Simulink* MATLAB *General Observer* Sensor

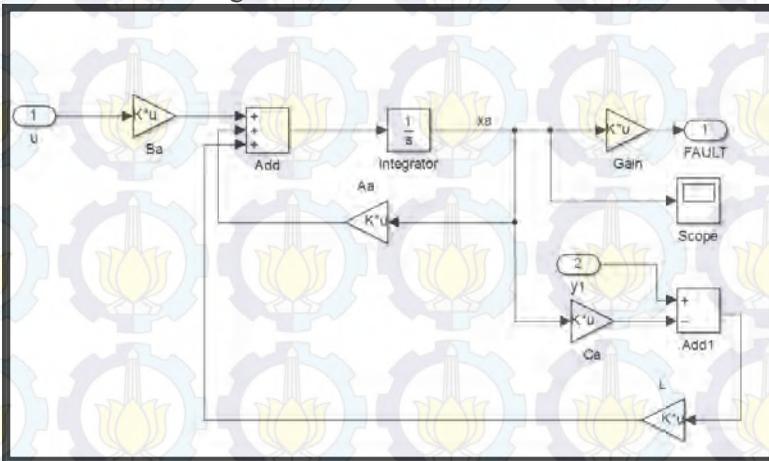




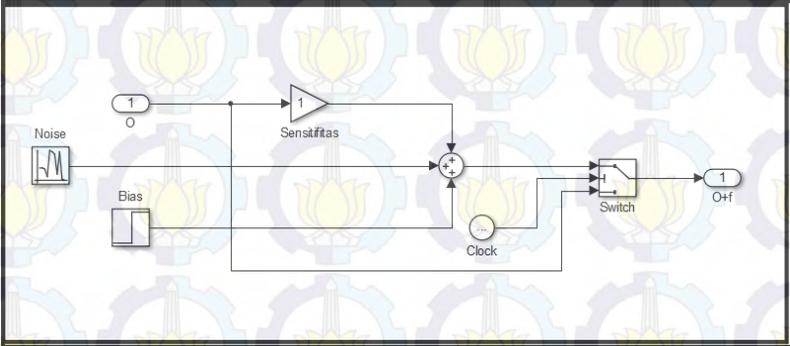
### D.3 Hasil Perancangan *Simulink* MATLAB *Observer* Sensor



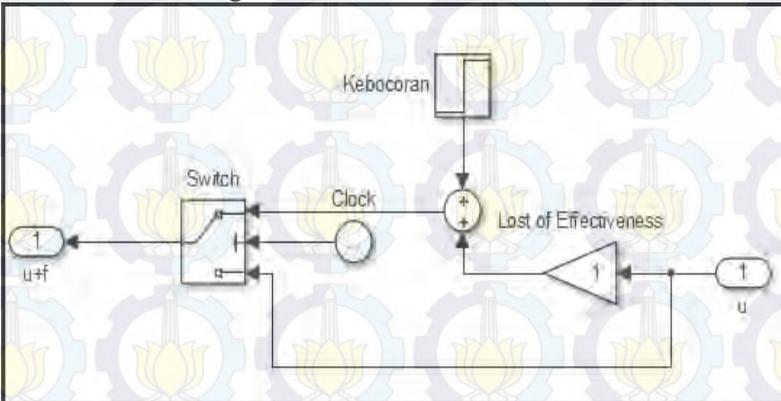
### D.4 Hasil Perancangan *Simulink* MATLAB *Observer* Aktuator



### D.5 Hasil Perancangan *Simulink* MATLAB Kesalahan Sensor



### D.6 Hasil Perancangan *Simulink* MATLAB Kesalahan Aktuator



## BIODATA PENULIS



Brigita Ria Marsellina merupakan nama lengkap penulis dengan nama panggilannya, Ria. Penulis dilahirkan di kota Surabaya, Jawa Timur pada tanggal 6 Maret 1993 sebagai anak tunggal dari pasangan Fransiskus Subagiyo dan Lidya Pudji Andayani. Riwayat pendidikan penulis adalah SD Negeri Dr. Soetomo VII Surabaya

tahun (1999 – 2005), SMP Negeri 6 Surabaya tahun (2005 – 2008), SMA Negeri 6 Surabaya tahun (2008 – 2011). Penulis diterima sebagai mahasiswa S1 Teknik Fisika ITS pada tahun 2011, kemudian fokus pada bidang minat rekayasa instrumentasi dan kontrol untuk menyelesaikan tugas akhirnya.

Penulis dapat dihubungi melalui email :  
[ria.marsellina@ymail.com](mailto:ria.marsellina@ymail.com).