



TUGAS AKHIR TF 141581

PERANCANGAN *ACTIVE FAULT TOLERANCE CONTROL* DENGAN KESALAHAN SENSOR DAN AKTUATOR PADA PENGENDALIAN TEMPERATUR *GASOLINE SPLITTER COLUMN (11C8) FUEL OIL COMPLEX I* PT.PERTAMINA (PERSERO) RU IV CILACAP

**ARINA VIDYA ABSHARI
NRP 2411 100 103**

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

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



FINAL PROJECT TF 141581

**DESIGN OF ACTIVE FAULT TOLERANCE
CONTROL BASED ON FAULT AT SENSOR AND
ACTUATOR IN TEMPERATURE CONTROL
SYSTEM OF GASOLINE SPLITTER COLUMN
(11C8) FUEL OIL COMPLEX I PT. PERTAMINA
(PERSERO) RU IV CILACAP**

**ARINA VIDYA ABSHARI
NRP 2411 100 103**

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

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

LEMBAR PENGESAHAN

PERANCANGAN *ACTIVE FAULT TOLERANCE CONTROL* DENGAN KESALAHAN SENSOR DAN AKTUATOR PADA PENGENDALIAN TEMPERATUR *GASOLINE SPLITTER COLUMN (11C8) FUEL OIL COMPLEX I PT.PERTAMINA (PERSERO)* RU IV CILACAP

Oleh:

Arina Vidya Abshari
NRP. 2411 100 103

Surabaya, Januari 2016
Mengetahui/Menyetujui

Pembimbing


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

Ketua Jurusan
Teknik Fisika FTI-ITS


Arus Muhamad Hatta, ST, MSi, Ph.D
NIP. 19780902 200312 1 002

**PERANCANGAN ACTIVE FAULT TOLERANCE
CONTROL DENGAN KESALAHAN SENSOR DAN
AKTUATOR PADA PENGENDALIAN TEMPERATUR
GASOLINE SPLITTER COLUMN (11C8) FUEL OIL
COMPLEX I PT.PERTAMINA (PERSERO)
RU IV CILACAP**

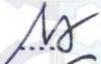
TUGAS AKHIR

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 :

ARINA VIDYA ABSHARI
NRP. 2411 100 103

Disetujui oleh Tim Penguji Tugas Akhir:

- | | | |
|---------------------------------|---|---------------|
| 1. Dr. Bambang Lelono W.,ST,MT |  | Pembimbing |
| 2. Dr. Ir. Purwadi Agus D, MSc |  | Ketua Penguji |
| 3. Dr. Ir. AuliaSiti Aisjah, MT |  | Penguji I |
| 4. Dr. Ridho Hantoro, ST, MT |  | Penguji II |
| 5. Bagus Tris Atmaja, ST, MT |  | Penguji III |

SURABAYA
Januari 2016

PERANCANGAN *ACTIVE FAULT TOLERANCE CONTROL* DENGAN KESALAHAN SENSOR DAN AKTUATOR PADA PENGENDALIAN TEMPERATUR *GASOLINE SPLITTER COLUMN (11C8) FUEL OIL COMPLEX I PT. PERTAMINA (PERSERO)* RU IV CILACAP

Nama Mahasiswa : Arina Vidya Abshari
NRP : 2411 100 103
Jurusan : Teknik Fisika
Dosen Pembimbing : Dr. Bambang Lelono W.,ST, MT

Abstrak

Salah satu permasalahan yang sering terjadi pada hampir semua sistem pengendalian termasuk pengendalian *temperature* adalah kesalahan pada komponen kontrol. Sensor temperatur yang digunakan untuk mengukur besarnya suhu pada *tray 18 Gasoline Splitter Column* di PT.Pertamina (Persero) RU IV Cilacap mengalami kesalahan pembacaan dengan *error* sebesar 0,05% jika dilihat dari data hasil kalibrasi sensor. Serta kesalahan bukaan *control valve* yang mengatur besarnya laju aliran *reflux*. Pada Tugas Akhir ini dilakukan perancangan sistem kontrol yaitu *active fault tolerance control (AFTC)* yang mampu mempertahankan performansi sistem ketika terjadi kesalahan sensor maupun aktuator dengan melakukan rekonfigurasi sinyal kontrol. Langkah awal yang dilakukan adalah menemukan fungsi transfer dari *plant* melalui pendekatan FOPDT dengan bantuan *software HYSYS*. Selanjutnya merancang kontrol PID dan merancang *observer* untuk sensor serta aktuator. Langkah terakhir adalah melakukan rekonfigurasi sinyal kontrol untuk sensor serta aktuator. Hasil simulasi menunjukkan bahwa sistem dengan algoritma AFTC dapat mengatasi kesalahan sensor seperti kesalahan bias, sensitivitas dan *noise* serta kesalahan aktuator

seperti kebocoran dan *lost of effectiveness*. Kedua uji tersebut menghasilkan performansi yang lebih baik dibandingkan dengan sistem pengendali PID tanpa AFTC. Hal ini dapat dilihat dari beberapa parameter seperti *maximum overshoot*, *maximum undershoot*, *error steady state* dan *settling time*. Salah satu bukti untuk uji kesalahan sensor terlihat pada hasil uji bias sebesar -0,75%, pada PID tanpa AFTC sistem tersebut menghasilkan *maximum overshoot* yang lebih besar yaitu 6,91% dan *settling time* yang lebih lama yaitu 9579 detik. Sedangkan pada sistem PID dengan AFTC *maximum overshoot* hanya sebesar 0,06% dengan *time settling* 7109 detik.

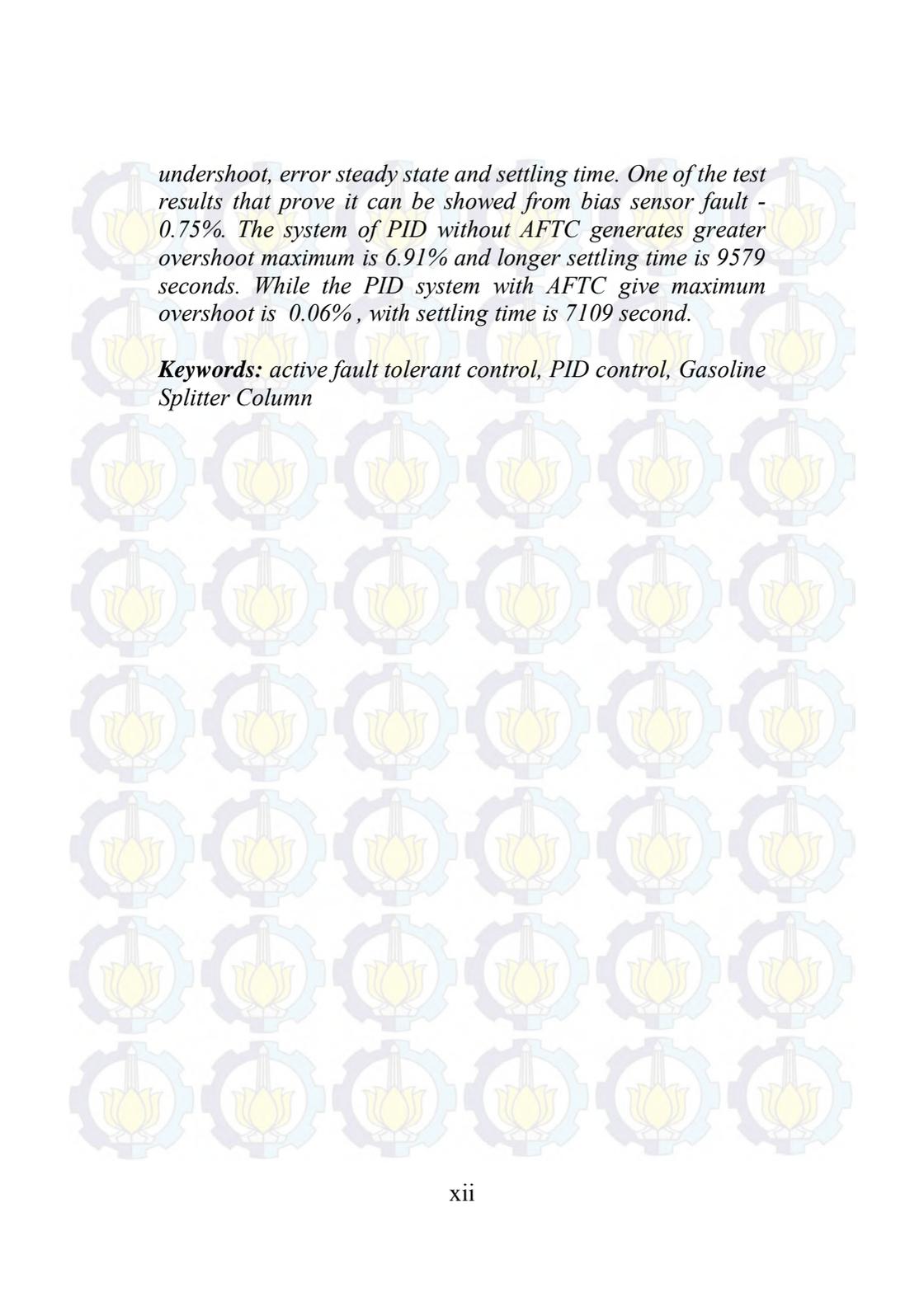
Kata kunci: *active fault tolerant control*, kontrol PID, Gasoline Splitter Column

**DESIGN OF ACTIVE FAULT TOLERANCE CONTROL
BASED ON FAULT AT SENSOR AND ACTUATOR IN
TEMPERATURE CONTROL SYSTEM OF GASOLINE
SPLITTER COLUMN (11C8) FUEL OIL COMPLEX I
PT.PERTAMINA (PERSERO) RU IV CILACAP**

Name : Arina Vidya Abshari
Student Number : 2411 100 103
Department : Engineering Physics
Supervisor : Dr. Bambang Lelono W., ST, MT

Abstract

One of the problem that happened in all control system, including temperature control system is fault on control components. Sensor which is used to measure the temperature of 18th tray Gasoline Splitter Column in PT.Pertamina (Persero) RU IV Cilacap goes wrong reading and has an error of 0,05% when seen from the datasheet of calibration sensor. And also fault of control valve that caused by wrong opening percentage of flow valve that control flow rate of the reflux. This final project is design the control system which call active fault tolerant control (AFTC) who is able to improve their performance when there is a fault of sensor and actuator. They do reconfiguration control to fix it. The first step is to find the transfer function of plant with FOPDT approxiamate that is helped by HYSIS software. The second step is to design the PID control and design the observer for sensor and actuator. The final step is to reconfigure the control system of sensor and actuator. The simulation results show that AFTC algorithm can cover the fault that caused by sensor example bias, sensitivity and noise. It's also can cover the fault that caused by actuator like leak of control valve and lost of effectiveness. Based on the experiment PID with AFTC give better performance than control system without AFTC. It is showed by several parameters , there are maximum overshoot, maximum



undershoot, error steady state and settling time. One of the test results that prove it can be showed from bias sensor fault - 0.75%. The system of PID without AFTC generates greater overshoot maximum is 6.91% and longer settling time is 9579 seconds. While the PID system with AFTC give maximum overshoot is 0.06% , with settling time is 7109 second.

Keywords: active fault tolerant control, PID control, Gasoline Splitter Column

KATA PENGANTAR

Puji syukur atas segala limpahan rahmat, kesehatan, keselamatan, dan ilmu yang Allah SWT berikan kepada penulis hingga mampu menyelesaikan laporan tugas akhir dengan judul:

PERANCANGAN *ACTIVE FAULT TOLERANCE CONTROL* DENGAN KESALAHAN SENSOR DAN AKTUATOR PADA PENGENDALIAN TEMPERATUR GASOLINE SPLITTER COLUMN (11C8) 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 mengucapkan terima kasih kepada:

1. Bapak Agus Muhamad Hatta, ST, Msi, Ph.D selaku Ketua Jurusan Teknik Fisika ITS
2. Bapak Dr. Bambang Lelono W.,ST,MT dan Ibu Katherin Indriawati ST, MT, selaku dosen pembimbing yang dengan sabar membimbing, memberikan saran dan kritiknya.
3. Bapak Ahmad Furchan, S.T, selaku *Engineer* bagian *Maintenance Plan and Scheduling* di PT.Pertamina (Persero) RU IV Cilacap yang telah membimbing dan memfasilitasi selama pengambilan data lapangan.
4. Orang tua saya, Ir.Noer Siswadi dan Ninik Ratna M., serta adik saya Hanum dan Hilmi yang selalu berdo'a untuk kebaikan serta kelancaran Tugas Akhir ini.
5. Teman-teman Teknik Fisika Angkatan 2011, terutama Khusnul, Ria, Arum, Vany, Vani, Riza, Damas dan Dicky, serta *partner* terbaik saya Yusuf yang senantiasa memberikan semangat dan motivasinya.

Demikian laporan tugas akhir ini dibuat dengan sebaik-baiknya. Semoga laporan ini bermanfaat bagi semua pihak, khususnya untuk kemajuan industri di Indonesia.

Surabaya, Januari 2016
Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	ix
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat	4
1.6 Sistematika Laporan	4
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Crude Distillation</i>	7
2.2 <i>Splitter Column</i>	7
2.3 <i>Gasoline Splitter Column</i> Unit FOC I	9
2.4 Sistem Pengendalian <i>Gasoline Splitter Column</i>	10
2.5 Pemodelan Dinamik <i>Gasoline Splitter Column</i>	10
2.6 FOPDT (<i>First Order Plus Dead Time</i>)	15
2.7 Pengendali PID dan Metode <i>Trial and Error</i>	16
2.8 <i>Fault Tolerant Control</i>	17
2.9 <i>Observer State</i>	20
2.10 <i>Linear Matrix Inequality</i> (LMI)	21
BAB III METODOLOGI PENELITIAN	25
3.1 Pengambilan Data <i>Plant</i>	28
3.2 Pemodelan <i>Gasoline Splitter</i>	29
3.3 Pemodelan Matematis Aktuator	33
3.4 Pemodelan Matematis Sensor	35
3.5 Validasi Pemodelan	35

3.6	Perancangan Kontrol <i>Proportional Integral Derivative</i> (PID)	39
3.7	Perancangan <i>Active Fault Tolerant Control</i>	40
3.7.1	Perancangan <i>Observer</i> untuk Sensor	40
3.7.2	Rekonfigurasi Sinyal Kontrol untuk Sensor	46
3.7.3	Perancangan <i>Observer</i> untuk Aktuator	47
3.7.2	Rekonfigurasi Sinyal Kontrol untuk Aktuator	52
3.8	Uji Performansi	53
BAB IV ANALISA DATA DAN PEMBAHASAN		55
4.1	Uji Kesalahan Sensor	55
4.1.1	Uji Kesalahan Bias	55
4.1.2	Uji Kesalahan Sensitivitas	58
4.1.3	Uji Kesalahan <i>Noise</i>	65
4.2	Uji Kesalahan Aktuator	66
4.2.1	Uji Kebocoran	66
4.2.2	Uji <i>Lost of Effectiveness</i>	73
4.1.3	Uji Kesalahan Kesalahan Sensor dan Aktuator Bersamaan	79
BAB V KESIMPULAN DAN SARAN		83
5.1	Kesimpulan	83
5.2	Saran	84
DAFTAR PUSTAKA		85
LAMPIRAN A		87
HMI <i>Gasoline Splitter Column</i> (11C8) FOCI PT.Pertamina (Persero) RU IV Cilacap		87
LAMPIRAN B		89
PFD <i>Gasoline Splitter Column</i> (11C8) FOCI PT.Pertamina (Persero) RU IV Cilacap		89
LAMPIRAN C		91

C.1 Trend Temperatur <i>Gasoline Splitter Column</i> pada Tanggal (21 Januari – 22 Januari 2015) di PT.Pertamina (Persero) RU IV Cilacap	91
C.2 Data Pengukuran Temperatur <i>Gasoline Splitter Column</i> pada Tanggal (21 Januari – 22 Januari 2015) di PT.Pertamina (Persero) RU IV Cilacap	91
C.3 Trend Buka-an Valve <i>Gasoline Splitter Column</i> pada Tanggal (21 Januari – 22 Januari 2015) di PT.Pertamina (Persero) RU IV Cilacap	94
C.4 Data Variasi Buka-an Valve <i>Gasoline Splitter Column</i> pada Tanggal (21 Januari – 22 Januari 2015) di PT.Pertamina (Persero) RU IV Cilacap	94
LAMPIRAN D	97
D.1 Simulink Matlab <i>General</i> dengan Kesalahan pada Sensor	97
D.2 Simulink Matlab <i>Fault Output of Sensor</i>	97
D.3 Simulink Matlab <i>Observer Sensor</i>	98
D.4 Simulink Matlab <i>General</i> dengan Kesalahan pada Sensor	98
D.5 Simulink Matlab <i>Fault Output of Aktuator</i>	99
D.6 Simulink Matlab <i>Observer Sensor</i>	99
D.7 Simulink Matlab <i>General</i> dengan Kesalahan pada Sensor dan Aktuator	100
BIODATA PENULIS	101

DAFTAR GAMBAR

Gambar 2.1	Skema <i>Splitter Column</i>	7
Gambar 2.2	Aliran <i>liquid</i> dan <i>vapour</i>	9
Gambar 2.3	Proses <i>Gasoline Splitter Column</i>	10
Gambar 2.4	<i>Rectifying Section</i>	11
Gambar 2.5	<i>Feed Section</i>	12
Gambar 2.6	<i>General Section</i>	12
Gambar 2.7	<i>Stripping Section</i>	14
Gambar 2.8	FOPDT <i>Model Parameter</i>	16
Gambar 2.9	Kontrol PID	17
Gambar 2.10	Skema Sistem FTC	18
Gambar 2.11	Struktur Umum AFTCS	19
Gambar 2.12	Observer State	21
Gambar 3.1	<i>Flowchart</i> Pengerjaan Tugas Akhir	27
Gambar 3.2	<i>Tray 18 Gasoline Splitter Column</i>	29
Gambar 3.3	Tampilan Simulasi <i>Plant</i> pada HYSYS 2006	31
Gambar 3.4	Grafik Respon <i>Open Loop</i> Hasil Simulasi pada <i>Software HYSYS</i>	32
Gambar 3.5	Blok Simulink pada Simulasi MATLAB <i>Closed Loop</i>	39
Gambar 3.6	Grafik Respon Simulasi Sistem Pengendalian <i>Closed Loop Gasoline Splitter Column</i>	40
Gambar 3.7	Blok Simulink Struktur <i>Observer</i> untuk Sensor	45
Gambar 3.8	Blok Diagram Rekonfigurasi Sinyal Kontrol dari <i>Obeserver</i> Sensor	46
Gambar 3.9	Blok Simulink Simulasi sistem AFTC dengan Kesalahan Sensor	47
Gambar 3.10	Blok Simulink Struktur <i>Observer</i> untuk Aktuator	51
Gambar 3.11	Blok Diagram Rekonfigurasi Sinyal Kontrol dari <i>Obeserver</i> Aktuator	53

Gambar 3.12	Blok Simulink Simulasi sistem AFTC dengan Kesalahan Aktuator	53
Gambar 4.1	Respon Sistem dengan Kesalahan Bias +0,75	56
Gambar 4.2	Respon Sistem dengan Kesalahan Bias -0,75%	57
Gambar 4.3	Respon Sistem dengan Kesalahan Sensitivitas 90%	59
Gambar 4.4	Respon Sistem dengan Kesalahan Sensitivitas 80%	60
Gambar 4.5	Respon Sistem dengan Kesalahan Sensitivitas 70%	61
Gambar 4.6	Respon Sistem dengan Kesalahan Sensitivitas 60%	63
Gambar 4.7	Respon Sistem dengan Pemberian <i>Noise</i>	65
Gambar 4.8	Perbesaran Respon Sistem dengan Pemberian <i>Noise</i>	65
Gambar 4.9	Respon Sistem dengan Kesalahan Kebocoran 30%	67
Gambar 4.10	Perbesaran Respon Sistem dengan Kesalahan Kebocoran 30%	67
Gambar 4.11	Respon Sistem dengan Kesalahan Kebocoran 50%	69
Gambar 4.12	Perbesaran Respon Sistem dengan Kesalahan Kebocoran 50%	69
Gambar 4.13	Respon Sistem dengan Kesalahan Kebocoran 70%	71
Gambar 4.14	Perbesaran Respon Sistem dengan Kesalahan Kebocoran 70%	71
Gambar 4.15	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 50%	73
Gambar 4.16	Perbesaran Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 50%	74
Gambar 4.17	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 70%	75

Gambar 4.18	Perbesaran Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 70%	76
Gambar 4.19	Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 90%	77
Gambar 4.20	Perbesaran Respon Sistem dengan Kesalahan <i>Lost of Effectiveness</i> 90%	78
Gambar 4.21	Hasil Uji Kesalahan Sensor dan Aktuator secara Bersamaan	80
Gambar 4.22	Perbesaran Hasil Uji Kesalahan Sensor dan Aktuator secara Bersamaan	80

DAFTAR TABEL

Tabel 3.1	Data Pemodelan <i>Plant</i>	28
Tabel 3.2	Data Pengukuran di PT.Pertamina (Persero) RU IV Cilacap dengan OP 44,7%	36
Tabel 3.3	Data Pengukuran di PT.Pertamina (Persero) RU IV Cilacap dengan OP 47,36%	36
Tabel 3.4	Data Pengukuran di PT.Pertamina (Persero) RU IV Cilacap dengan OP 52,6%	37
Tabel 3.5	Data Pengukuran di PT.Pertamina (Persero) RU IV Cilacap dengan OP 57,8%	37
Tabel 3.6	Data Pengukuran di PT.Pertamina (Persero) RU IV Cilacap dengan OP 63,1%	38
Tabel 3.7	Data Pengukuran dari Simulasi Sistem pada <i>HYSYS</i>	38
Tabel 3.8	Perhitungan <i>Error</i> Hasil Pengukuran Simulasi <i>HYSYS</i> dengan Hasil Pengukuran Perusahaan	38
Tabel 4.1	Parameter Performansi dengan Kesalahan Bias +0,75%	56
Tabel 4.2	Parameter Performansi dengan Kesalahan Bias -0,75%	57
Tabel 4.3	Parameter Performansi dengan Kesalahan Sensitivitas 90%	59
Tabel 4.4	Parameter Performansi dengan Kesalahan Sensitivitas 80%	60
Tabel 4.5	Parameter Performansi dengan Kesalahan Sensitivitas 70%	62
Tabel 4.6	Parameter Performansi dengan Kesalahan Sensitivitas 60%	63
Tabel 4.7	Parameter Performansi dengan Kesalahan Kebocoran 30%	68
Tabel 4.8	Parameter Performansi dengan Kesalahan Kebocoran 50%	70

Tabel 4.9	Parameter Performansi dengan Kesalahan Kebocoran 70%	72
Tabel 4.10	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 50%	74
Tabel 4.11	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 70%	76
Tabel 4.12	Parameter Performansi dengan Kesalahan <i>Lost of Effectiveness</i> 90%	78

BAB I PENDAHULUAN

1.1 Latar Belakang

PT. Pertamina (Persero) RU IV Cilacap merupakan salah satu unit pengolahan yang didirikan untuk menghasilkan produk BBM dan Non-BBM guna memenuhi kebutuhan dalam negeri. Untuk mendapatkan produk tersebut, diperlukan serangkaian proses pengolahan *crude oil*. *Crude Distilling Unit* (Unit 1100) merupakan unit yang bertugas untuk memisahkan *crude oil* berdasarkan titik didih masing-masing fraksinya [1]. Salah satu *plant* yang ada di unit ini adalah *Gasoline Splitter Column* (11C8). *Gasoline Splitter Column* berfungsi untuk memisahkan *naphtha* yang berasal dari *Stabilizer Colum* (11C7) menjadi *light naphtha* sebagai *top product* dan *heavy naphtha* sebagai *bottom product*. Temperatur merupakan salah satu aspek yang mempengaruhi kualitas produk. Temperatur pada *tray* 18 kolom 11C8 dijaga dalam rentang 85 °C – 100 °C dengan mengatur aliran *reflux* dari *overhead accumulator*. PT. Pertamina (Persero) RU IV Cilacap telah menggunakan *PID controller* sebagai sistem pengendalian, namun sistem pengendalian tersebut belum toleran terhadap kemungkinan terjadinya kesalahan komponen kendali [4].

Salah satu permasalahan yang sering terjadi pada hampir semua sistem pengendalian termasuk pengendalian temperatur adalah kesalahan pada komponen kendali. Dalam kasus ini kesalahan prosentase bukaan *control valve* mengakibatkan aliran *reflux* yang berasal dari *overhead accumulator* tidak sesuai dengan yang diinginkan, sehingga *temperature* dalam kolom tidak sesuai dengan *set point*. Namun terkadang kesalahan *control valve* dan sensor mungkin terjadi secara bersamaan. Ketika terjadi malfungsi dari komponen-komponen kendali tersebut, performansi sistem tidak dapat maksimal. Data kesalahan dari kedua komponen tersebut didapatkan dari data proses pada tanggal 21 Januari 2015 – 22 Januari 2015 (Lampiran C.1). Permasalahan pengendalian temperatur pada *gasoline splitter*

column ini berpotensi menghasilkan produk *light naphtha* dengan kualitas yang kurang baik. Oleh karena itu diperlukan suatu sistem pengendalian yang mampu menjaga performansi walaupun terjadi kesalahan komponen kendali.

FTC (Fault tolerant control) adalah sistem pengendalian yang mempunyai kemampuan untuk mengakomodasi kesalahan komponen pada sistem secara otomatis. Sistem pengendalian ini mampu memelihara kestabilan dan performansi sistem walaupun ada kesalahan pada komponen dari sistem tersebut. Jenis *FTC* ada 2 yakni *Active Fault Tolerance Control (AFTC)* dan *Passive Fault Tolerance Control (PFTC)*. *AFTC* dibagi ke dalam empat buah sub sistem, diantaranya adalah rekonfigurasi kontroler, skema *FDD*, mekanisme rekonfigurasi kontroler, dan perintah dari alat pengatur pada sistem. *FDD* dan kontroler yang telah terekonfigurasi merupakan fitur utama yang membedakan *AFTCS* dari *PFTCS* [12]. Dari penjelasan kondisi permasalahan tersebut, maka akan dilaksanakan penelitian Tugas Akhir yang berkaitan dengan perancangan sistem *active fault tolerance control* pada *plant gasoline splitter column* di unit *1100 fuel oil complex I PT. Pertamina (Persero) RU IV Cilacap*.

1.2 Perumusan Masalah

Berdasarkan tahapan untuk merancang *Active Fault Tolerance Control (AFTC)*, maka permasalahan yang muncul dalam penelitian tugas akhir ini adalah sebagai berikut :

- Bagaimana merancang *observer* dengan kesalahan sensor dan aktuator pada sistem pengendalian temperatur *gasoline splitter column* unit *fuel oil complex I PT. Pertamina (Persero) RU IV Cilacap* ?
- Bagaimana merekonfigurasi sinyal kontrol dengan kesalahan pada sensor dan aktuator pada sistem pengendalian temperatur *gasoline splitter column* unit *fuel oil complex I PT. Pertamina (Persero) RU IV Cilacap* ?

1.3 Batasan Masalah

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

- Pemodelan *plant* yang ditinjau hanya proses *rectifying* dan *stripping* pada *gasoline splitter column* di unit FOC I PT. Pertamina (Persero) RU IV Cilacap,
- Pemodelan *plant* dilakukan dengan pendekatan *FOPDT* berdasarkan grafik respon yang disimulasikan melalui *software Aspen HYSYS*.
- Kesalahan sensor yang dianalisis hanya kesalahan nilai keluaran *output* sensor yang disebabkan oleh kesalahan pembacaan nilai temperatur *tray* 18 pada *gasoline splitter column*,
- Kesalahan aktuator yang dianalisis hanya kesalahan jumlah aliran *reflux* yang berasal dari *overhead accumulator* yang disebabkan oleh kesalahan sinyal *input* yang masuk ke *control valve*,
- Besaran yang dikontrol adalah nilai temperatur pada *tray* 18 (°C) dan besaran yang dimanipulasi adalah laju aliran *reflux* yang berasal dari *overhead acumulator* (kg/s),
- Sampel data proses diambil dari tanggal 1 Januari 2015 - 3 Maret 2015,
- Sistem kendali yang digunakan pada sistem pengendalian temperatur di *gasoline splitter column* menggunakan algoritma PID.

1.4 Tujuan Penelitian

Pada sistem pengendalian temperatur pada *gasoline splitter column* unit FOC I di PT. Pertamina (Persero) RU IV Cilacap terdapat kesalahan pada sensor dan aktuator sehingga diajukan suatu algoritma kontrol untuk memperbaiki unjuk kerja sistem pengendalian temperatur tersebut. Sebagai tujuan dari tugas akhir ini adalah merancang sistem *Active Fault Tolerant Control (AFTC)* dengan kesalahan pada sensor dan aktuator pada *gasoline splitter column* unit FOC I di PT. Pertamina (Persero) RU IV

Cilacap. Untuk menguji unjuk kerja dilakukan perbandingan secara simulasi antara sistem *AFTC* dengan sistem menggunakan kontrol konvensional (tanpa mekanisme rekonfigurasi).

1.5 Manfaat

Manfaat yang didapat dari penelitian tugas akhir ini diantaranya :

- Perancangan *Active Fault Tolerance Control* (*AFTC*) ini dapat dijadikan sebagai sebuah pengetahuan baru tentang perancangan suatu sistem kendali yang tetap handal meskipun terjadi suatu kesalahan pada sensor ataupun aktuator.

1.6 Sistematika Laporan

Sistematika laporan yang digunakan dalam penyusunan laporan penelitian tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang pengambilan topik dan judul penelitian, perumusan masalah yang harus diselesaikan, adanya batasan masalah, tujuan penelitian yang ingin dicapai, manfaat dan sistematika laporan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi mengenai teori – teori yang mendukung pengerjaan tugas akhir ini, antara lain berisi tentang pengertian, gambaran umum, dan proses dari *gasoline splitter column*, sistem kontrol pada *gasoline splitter column*, sistem kendali PID, dan teori mengenai *Active Fault Tolerant Control (AFTC)* serta perancangannya.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tahap – tahap penyelesaian masalah yang telah dirujelaskan pada sub bab I sebelumnya, terdiri dari pemodelan *plant* dengan menggunakan pendekatan *FOPDT*, langkah – perancangan kendali PID, dan perancangan *observer* sensor dan aktuator serta pengujian dengan simulasi pada program MATLAB.

BAB IV ANALISIS DATA DAN PEMBAHASAN

Bab ini berisi hasil uji yang dilakukan terhadap sistem pengendalian temperatur pada *gasoline splitter column*. Uji yang dilakukan berupa uji kesalahan bias, uji sensitivitas, dan uji *noise* untuk sensor serta uji *lost of effectiveness* dan uji kebocoran untuk aktuator. Dalam bab ini dilakukan pula perbandingan parameter uji performansi antara sistem pengendalian temperatur dengan algoritma PID dengan *AFTC* dan sistem pengendalian level dengan algoritma PID saja (tanpa *AFTC*).

BAB V KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan dari tugas akhir yang mengacu pada permasalahan sebelumnya dan diharapkan dapat menjawab permasalahan sebagai solusi. Saran dari penulis juga dicantumkan untuk memperbaiki kekurangan dari hasil perancangan.



Halaman ini sengaja dikosongkan

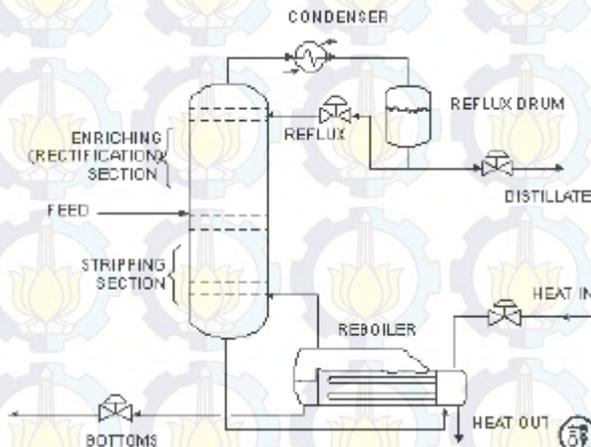
BAB II TINJAUAN PUSTAKA

2.1 *Crude Distillation*

Penyulingan minyak mentah merupakan proses penting dan menjadi *final* pada setiap industri pengolahan minyak. Bahan baku minyak mentah tersebut dipisahkan sesuai dengan karakteristik titik didihnya [3]. Proses pemisahan awal terjadi di *crude distillation column* yang terdiri atas 33 *tray*. *Crude distillation column* memisahkan minyak mentah menjadi lima fraksi, yaitu produk atas (yang terdiri dari *naphtha*, *light ends* dan *light tops*), *kerosene*, *LGO*, *HGO* dan *residue* sebagai produk bawah. Produk *naphtha* ini digunakan sebagai umpan unit *Naphtha Hydrotreater* (NHT) untuk diolah pada *gasoline splitter column* guna mendapatkan produk *light naphtha* dan *heavy naphtha* [4].

2.2 *Splitter Column*

Splitter column digunakan untuk mempertajam pemisahan komponen. Skema *splitter column* secara umum sama seperti *continuous distillation column*, terlihat pada gambar 2.1.

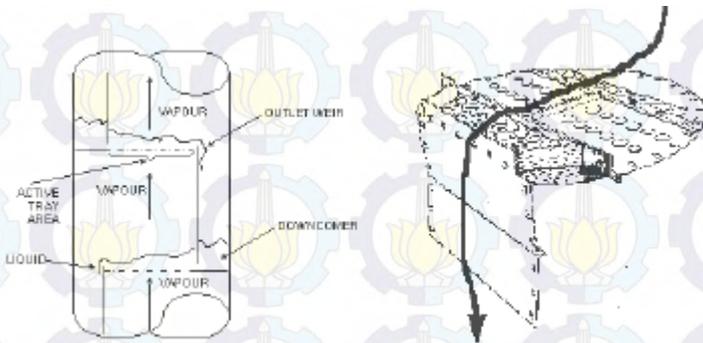


Gambar 2.1 Skema *Splitter Column* [6]

Terdapat dua prinsip metode pemisahan yang biasa dipraktikkan dengan destilasi. Pertama adalah didasarkan pada produksi uap yang dihasilkan dari pemanasan campuran larutan sampai mendidih, kemudian uap yang terbentuk dipisahkan dan dikondensasi tanpa mengembalikan sebagian cairan ke dalam proses (destilasi *non-reflux*). Sedangkan metode yang kedua adalah dengan mengembalikan uap yang telah dikondensasi sebagian atau seluruhnya ke dalam proses untuk dikontakkan kembali dengan uap di dalam kolom (destilasi dengan *reflux* atau rektifikasi) [5].

Umpan yang masuk ke dalam kolom destilasi dapat dalam bentuk *saturated liquid* atau dapat juga dalam bentuk dua fasa. Sebagian umpan yang masuk ke dalam kolom akan teruapkan dan sebagian lagi masih dalam fasa cair. Aliran uap akan naik ke bagian *rectifying* dan aliran cair akan turun ke bagian *stripping*. Pada tiap *tray* aliran uap dan aliran cair akan bercampur sampai mencapai *equilibrium*, sehingga uap dan cairan yang keluar dari tiap *tray* akan berada pada kondisi *equilibrium*. Konsentrasi komponen yang memiliki titik didih paling rendah akan meningkat pada aliran uap yang menuju ke bagian *top* kolom. Produk akhir dari uap di bagian *overhead* akan dikondensasikan dalam *condenser*, sebagian dikeluarkan sebagai *distillate product* dan sebagian lagi akan dikembalikan pada *top* kolom dalam fasa cair [5].

Cairan yang meninggalkan *tray* pada bagian *bottom* akan masuk ke dalam *reboiler*, sebagian akan menjadi uap dan sisanya tetap sebagai cairan. Cairan yang keluar dari *reboiler* mengandung lebih banyak komponen yang memiliki titik didih tinggi. Sedangkan uap dari *reboiler* akan dikembalikan kembali pada bagian *tray* paling bawah untuk dikontakkan kembali dengan cairan yang turun dari *tray* di atasnya [5].

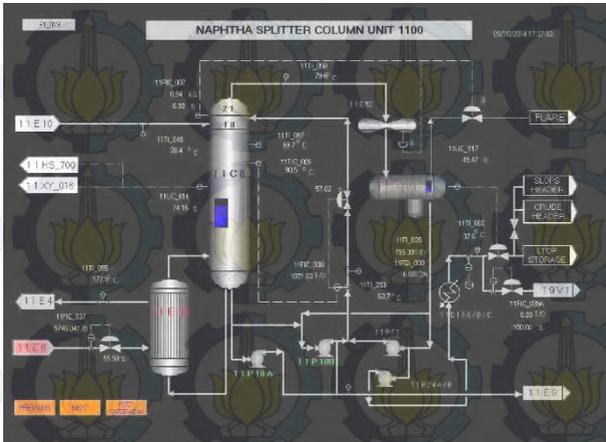


Gambar 2.2 Aliran *liquid* dan *vapour* [6]

Setiap *tray* mempunyai dua sisi bersebelahan yang disebut *downcomers*. *Liquid* jatuh melalui *downcomer* akibat gaya gravitasi, dari satu *tray* ke *tray* lain di bawahnya. Sedangkan *vapour* menuju ke bagian atas kolom melalui *hole* pada setiap *tray* [6].

2.3 Gasoline Splitter Column Unit FOC I

Gasoline splitter column merupakan *splitter column* yang berfungsi untuk memisahkan *naphtha* menjadi *light naphtha* dan *heavy naphtha*. *Gasoline splitter column* ini terdiri atas 21 *tray*. *Feed* berupa *naphtha* yang berasal dari *stabilizer column*, dengan *flow rate* 105.500 kg/h, suhu 116 °C dan tekanan 2 kg/cm² masuk ke kolom pada *tray* 10. Produk bawah dipanaskan kembali oleh *reboiler* 11E13 untuk menjaga suhu operasi kolom bagian bawah pada rentang 136 – 139 °C. Fraksi ringan bergerak menuju *condenser* 11E52, lalu masuk ke *accumulator* 11V3 untuk di-*reflux* sebagian ke *gasoline splitter column* 11C8 pada *tray* 21. Sebagian yang lain dikirim ke *storage* sebagai *top product* untuk *mogas blending* [4].



Gambar 2.3 Proses *Gasoline Splitter Column* [7]

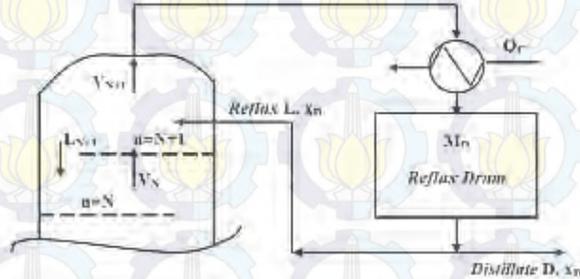
2.4 Sistem Pengendalian Temperatur *Gasoline Splitter Column*

Pada *gasoline splitter column*, terdapat tiga jenis sistem pengendalian yaitu sistem pengendalian level pada *hold up liquid* yang ada di dasar *splitter*, sistem pengendalian temperatur pada *tray 18* dan sistem pengendalian *pressure* pada bagian atas kolom. Dalam tugas akhir ini, loop pengendalian yang menjadi objek studi ialah loop pengendalian temperatur yang dijaga dengan mengendalikan laju aliran *reflux*. Sistem pengendalian temperatur pada *gasoline splitter column* menggunakan kontrol *proportional integral derivative* (PID) yang terintegrasi dalam *distributed control system* (DCS). Di *control room*, *human machine interface* (HMI) akan memberikan *set point* temperatur *tray 18* pada *gasoline splitter column* yang merupakan hasil pengukuran dari sensor *termocouple*. *Set point* dijaga dalam rentang 85 – 100°C [8].

2.5 Pemodelan Dinamik *Gasoline Splitter Column*

Gasoline splitter column memiliki karakteristik yang sama dengan kolom distilasi, oleh karena itu model matematis yang akan digunakan untuk mendapatkan model dinamik dari kolom *splitter* ini adalah model matematis dari kolom distilasi. Dimana

konfigurasi kolom ini dibagi menjadi 4 bagian yaitu, bagian *rectifying (top)*, *feed*, *general*, dan *stripping (bottom)*. Di dalam *gasoline splitter column* terjadi perpindahan *massa*, *energy* dan komponen dari setiap *tray*. Sehingga untuk memodelkan dinamika proses pada *gasoline splitter column* menggunakan hukum kesetimbangan *massa*, *energy* dan komponen [9].



Gambar 2.4 Rectifying Section [9]

Persamaan kesetimbangan massa :

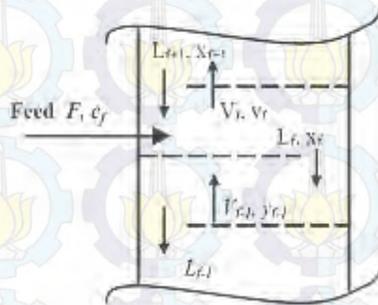
$$\frac{d(M_{N+1})}{dt} = L + V_N - L_{N+1} - V_{N+1} \quad (2.1)$$

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 :

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



Gambar 2.5 Feed Section [9]

Persamaan kesetimbangan massa :

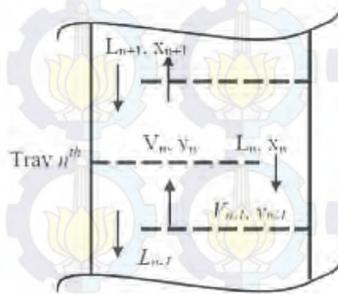
$$\frac{d(M_f)}{dt} = F + L_{f+1} + V_{f-1} - L_f - V_f \quad (2.4)$$

Persamaan kesetimbangan komponen :

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

Persamaan kesetimbangan energi :

$$\frac{d(M_f h_f)}{dt} = h_f F + h_{n+1} L_{n+1} + H_{n-1} V_{n-1} - h_n L_n - H_n V_n \quad (2.6)$$



Gambar 2.6 General Section [9]

Persamaan kesetimbangan massa :

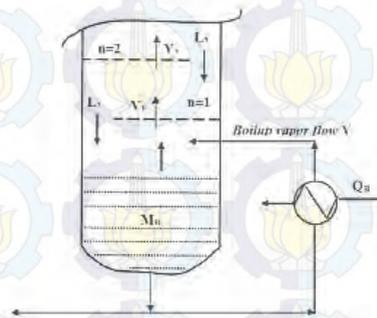
$$\frac{d(M_n)}{dt} = L_{n+1} + V_{n-1} - L_n - V_n \quad (2.7)$$

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 :

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



Gambar 2.7 Stripping Section [9]

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,

n : tray ke - n

V : laju vapor (kg/s)

L : laju liquid (kg/s)

x : konsentrasi liquid untuk fraksi ringan (M)

y : konsentrasi vapor untuk fraksi ringan (M)

h : entalpi liquid (kJoule/kg)

- H : entalpi *vapor* (kJoule/kg)
 B : laju aliran keluar *liquid* dari bawah (kg/s)

2.6 FOPDT (First Order Plus Dead Time)

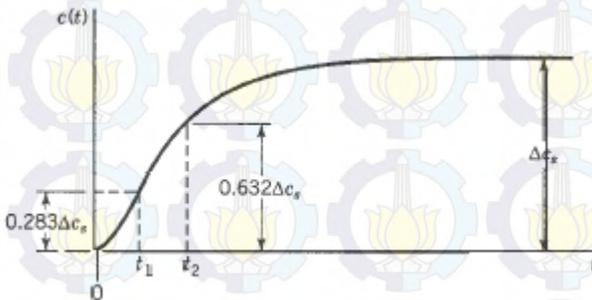
FOPDT (*first order plus dead time*) adalah sebuah sistem orde satu yang memiliki *dead time*. FOPDT telah banyak digunakan untuk memodelkan sistem dalam rangka penyederhanaan dari sistem yang berorde tinggi menjadi sistem yang berorde satu yang memiliki *dead time* [10]. Berikut ini merupakan persamaan FOPDT .

$$\frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta_p s}}{\tau s + 1} \quad (2.13)$$

dimana,

- K : *gain steady plant*
 τ : $1,5((63\% * K) - (28\% * K))$

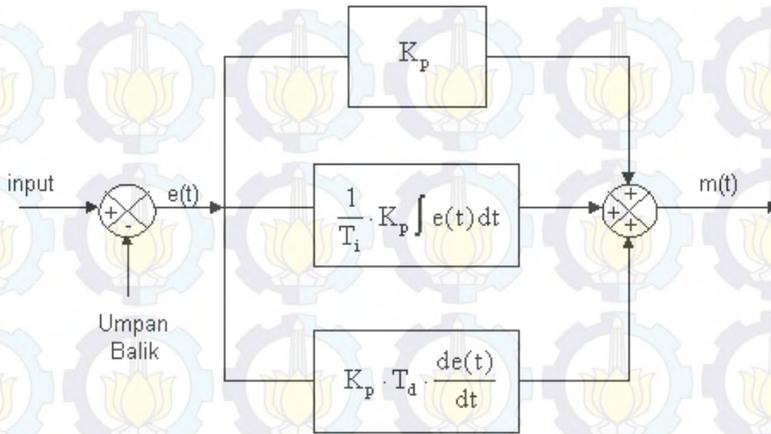
Fungsi transfer tersebut mengandung 3 parameter yaitu *gain*, *time constant*, dan *dead time*. *Gain* menggambarkan besarnya perubahan *output* per perubahan *input*, *time constant* menggambarkan kecepatan proses dalam merespon *input* dan *dead time* merupakan *interval* waktu dari *input* mulai diberikan hingga *variabel output* mulai mengalami perubahan [10].



Gambar 2.8 FOPDT Model Parameter [10]

2.7 Pengendali PID dan Metode *Trial and Error*

Pengendali PID merupakan suatu algoritma kontrol untuk menghitung besarnya koreksi yang diperlukan suatu alat pengendali untuk mengendalikan sebuah proses. Kontrol *proportional* digunakan sebagai penguat atau pembesar sinyal sehingga *process variable* dapat mencapai *set point*. Namun penambahan *gain proportional* akan meninggalkan *offset*. Oleh karena itu, kekurangan dari kontrol *proportional* yang selalu meninggalkan *offset* diperbaiki oleh kontrol *integral* untuk menghilangkan *offset* tersebut. Kemampuan kontrol *integral* untuk menghilangkan *offset* tidak disertai dengan kemampuan respon secara cepat, sehingga menjadi kian nyata kalau kontrol tersebut digunakan untuk elemen proses temperatur. Upaya memperbaiki respon didapatkan dengan menggunakan kontrol *derivative* [11].



Gambar 2.9 Blok Diagram Kontrol PID [11]

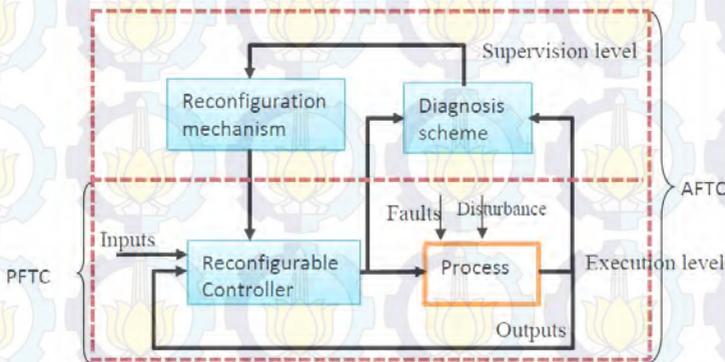
Suatu sistem pengendalian terdapat proses *tuning* atau penyetelan alat agar didapatkan sistem dengan hasil respon yang stabil. Berbagai macam metode *tuning* telah ditemukan. Salah satunya adalah metode *trial and error*. Metode ini secara harfiah dapat disamakan dengan dengan istilah coba-coba, tetapi kata *trial and error* disini mengandung pengertian coba-coba yang didukung dengan pengetahuan yang pasti. Misalnya, seseorang memang mencoba-coba menurunkan kontrol *proportional* untuk mempercepat reaksi, tetapi orang tersebut harus tau benar bahwa kontrol *proportional* cukup diturunkan dari 125% menjadi 100%, tidak dari 125% menjadi 25% [11]. Artinya, metode *trial and error* dapat dikatakan sebagai metode coba-coba yang didukung dengan pengetahuan dari seseorang mengenai akibat yang akan terjadi jika parameter diturunkan secara drastis.

2.8 Fault Tolerant Control

Kini peningkatan performa dan keamanan sangat dibutuhkan dalam suatu sistem kontrol. Pada industri proses, manufaktur, dan

lainnya terkadang terjadi beberapa kerusakan *minor* pada aktuator, sensor ataupun komponen-komponen lain penyusun sistem tersebut. Untuk mengatasi kerusakan *minor* tersebut, dilakukan desain kontrol yang mampu menoleransi kerusakan yang terjadi dengan tetap menjaga stabilitas sistem yang diinginkan. Kontrol sistem ini sering disebut dengan *fault tolerance control system* (FTCS) [12].

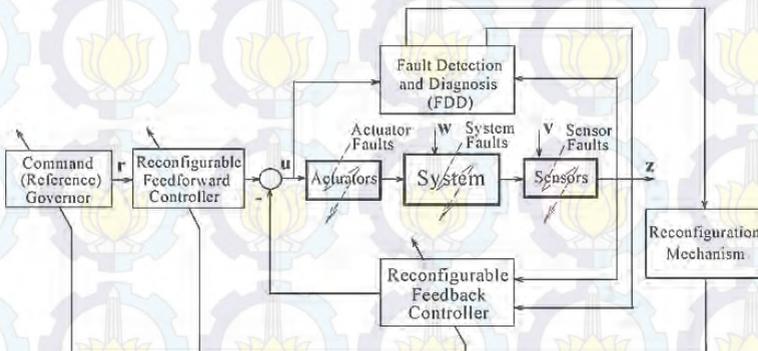
FTCS dapat diklasifikasikan pada 2 tipe yaitu *passive fault tolerance control system* (PFTCS) dan *active fault tolerance control system* (AFTCS). PFTCS didesain untuk menjadi sistem kontrol yang *robust* dari kesalahan suatu komponen. Sedangkan AFTCS bereaksi terhadap kesalahan yang terjadi pada suatu komponen dengan merekonfigurasi aksi kontrol sehingga kestabilan dan kinerja pada sistem dapat dijaga. Dengan demikian, tujuan utama dari FTCS adalah merancang kontroler dengan struktur yang cocok untuk mencapai kestabilan dan kinerja yang diinginkan. Tidak hanya ketika kontrol komponen berfungsi secara normal tetapi ketika terjadi kesalahan pada suatu komponen tersebut [13].



Gambar 2.10 Skema Sistem FTC [14]

ATFCS dapat juga disebut *fault detection and identification* (FDI). Perbedaan AFTC dan PFTC terletak pada perancangan 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, dan mekanisme rekonfigurasi yang sebisa mungkin mampu memulihkan performansi sistem setelah terjadi kesalahan walaupun terdapat ketidakpastian dan *time delay* pada FDI [13].



Gambar 2.11 Struktur Umum AFTCS [13]

Secara umum AFTC terdiri atas empat sub sistem, yaitu [12]:

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

Rekonfigurasi kontrol harus dirancang secara otomatis untuk menjaga stabilitas dan performansi yang diinginkan. Selain itu, untuk memastikan bahwa sistem tersebut *closed loop*, maka rekonfigurasi kontrol harus dirancang secara *feedforward*. Hal ini dilakukan untuk menghindari atau mempertimbangkan akan terjadinya penurunan performansi setelah terjadinya kesalahan. Selain itu, ditambahkan *command governor* untuk menyesuaikan aksi kontrol secara *online*. Pada intinya, perancangan sistem yang penting dalam AFTC terdiri atas dua sub sistem yaitu *fault detection and identification* (FDI) dan mekanisme *reconfiguration*

controller. FDI berfungsi sebagai pendeteksi jenis kesalahan dan besarnya kesalahan tersebut, sedangkan rekonfigurasi sinyal kontrol merupakan mekanisme perubahan aksi kontrol untuk menghilangkan kesalahan yang terjadi [13].

2.9 Observer State

Observer adalah suatu algoritma yang bertujuan untuk mengestimasi *state* sistem berdasarkan model matematis sistem. Sedangkan *descriptor observer*, mendeskripsikan sistem dalam bentuk *state space* dengan menggunakan parameter – parameter sebagai berikut [15].

$$\dot{x} = Ax + Bu \quad (2.14)$$

$$y = Cx \quad (2.15)$$

Model matematis dari *descriptor observer* sama seperti model matematis pada sistem atau *plant* yang telah diubah ke dalam bentuk persamaan *state space* dengan ditambahkan dengan *gain observer* dan menjadi persamaan sebagai berikut [15].

$$\dot{\bar{x}} = A\bar{x} + Bu + K(y - \bar{y}) \quad (2.16)$$

$$\bar{y} = C\bar{x} \quad (2.17)$$

Dengan memasukkan y di persamaan 2.15 dan y bar di persamaan 2.17 ke dalam persamaan 2.16, terbentuklah persamaan seperti berikut,

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

$$\dot{\bar{x}} = (A - KC)\bar{x} + Bu + KCx \quad (2.19)$$

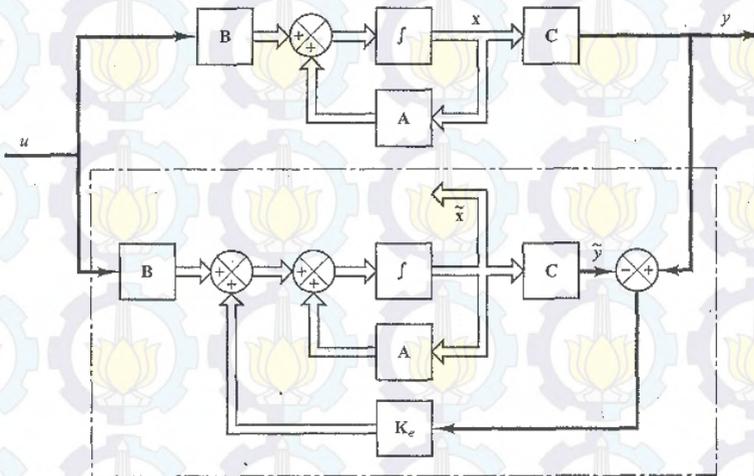
Jika model matematis telah sempurna, maka x estimasi akan sama dengan x sistem. Akan tetapi, seringkali terjadi kesalahan pada model, sehingga akan menimbulkan *error* atau perbedaan antara x sistem dan x estimasi yang dapat dirumuskan sebagai berikut. Untuk menentukan persamaan *observer error*, persamaan 2.19 dikurangi dengan persamaan 2.14 sebagai berikut.

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

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

$$e = x - \bar{x} \quad (2.22)$$

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



Gambar 2.12 *Observer State* [2]

Gambar 2.12 merupakan *representasi* dari persamaan (2.14) , (2.15) , (2.16) dan (2.17).

Dimana :

A = matriks pada *state*

B = matriks pada kontrol *input*

C = matriks pada hasil pengukuran

x = *state* sistem

\tilde{x} = *x estimate*

u = kontrol *input*

2.10 Linear Matrix Inequality (LMI)

Terdapat berbagai metode untuk menguji kestabilan suatu sistem. Salah satu metode yang digunakan untuk menguji kestabilan sistem adalah kestabilan Lyapunov. Kestabilan Lyapunov tersebut dapat digunakan untuk menguji kestabilan sistem linier maupun nonlinier. 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].

Menurut sejarahnya, *Linear Matrix Inequality* (LMI) muncul pada tahun 1890 saat Lyapunov menunjukkan persamaan diferensial berikut ini [18].

$$\frac{d}{dt} x(t) = Ax(t) \quad (2.24)$$

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

$$A^T P + PA < 0 \quad (2.25)$$

$$P = P^T > 0 \quad (2.26)$$

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,

$$\dot{x} = f(x, u) \quad (2.27)$$

Maka sistem akan 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 $\dot{V}(x) < 0$.

Banyak permasalahan dalam sistem dan teori kontrol yang dapat dituangkan kembali ke dalam permasalahan *convex*. Yang termasuk didalamnya adalah *Linear Matrix Inequality* (LMI) yang nantinya dalam penulisan selanjutnya lebih dikenal dalam istilah

bahasa Indonesia sebagai “Pertidaksamaan Matriks Linier”. Dalam banyak kasus, pertidaksamaan memiliki bentuk yang simultan dengan Lyapunov atau pertidaksamaan aljabar Riccati seperti yang telah dijelaskan diatas.

Robust Control Toolbox 3 User Guide menjelaskan bahwa *Linear Matrix Inequality* (LMI) dan teknik LMI muncul sebagai *tool* desain yang sangat kuat dalam jangkauan *control engineering* untuk mengidentifikasi sistem dan desain struktural.

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, pertidaksamaan matriks linier (LMI) memiliki bentuk persamaan sebagai berikut.

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.28)$$

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.



Halaman ini sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai prosedur penelitian yang dilakukan guna mencapai tujuan penelitian Tugas Akhir ini. Berikut beberapa tahapan dalam perancangan *Active Fault Tolerance Control* (AFTC) pada *Gasoline Splitter Column* unit FOC I di PT. Pertamina (Persero) RU IV Cilacap.

- a. Pengambilan Data Spesifikasi *Gasoline Splitter Column*, Sensor dan Aktuator.
Pengambilan data dilakukan di PT.Pertamina (Persero) RU IV Cilacap. Data yang didapatkan meliputi *Spesification Sheet of Gasoline Splitter*, *Spesification Sheet of Temperature Transmitter*, *Spesification Sheet of Flow Valve*, *Process Flow Diagram of 11C8*, *track record DCS* dan Data Analisa Laboratorium FOC I.
- b. Pemodelan *Gasoline Splitter Column*, Sensor dan Aktuator.
Pembuatan model *Gasoline Splitter Column* dilakukan melalui pendekatan FOPDT (*first order plus dead time*) dengan mengambil respon *input* dan *output plant* yang didapatkan dari *software* Aspen HYSYS 2006.
- c. Validasi Model
Validasi model *plant* dilakukan dengan tujuan untuk memastikan model yang dibuat telah merepresentasikan *plant* yang sesungguhnya. Validasi dilakukan dengan perbandingan antara data proses pada *plant* pada saat kondisi normal dan mantap yaitu 86 °C dengan HYSYS pada kondisi *dynamic mode*.
- d. Perancangan kontrol PID
Perancangan kontrol PID bertujuan untuk menentukan nilai parameter *proportional*, *integral* dan *derivative* 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)

Tahap pertama dalam perancangan AFTC adalah merancang *observer* untuk aktuator dan juga sensor. Perancangan *observer* sensor dan *observer* aktuator dilakukan dengan 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. Tahap selanjutnya adalah melakukan rekonfigurasi kontrol, dimana perancangan ini bertujuan untuk mengembalikan stabilitas kontrol yang tidak stabil akibat kesalahan pada sensor dan aktuator dengan mengubah perhitungan sinyal kontrolnya.

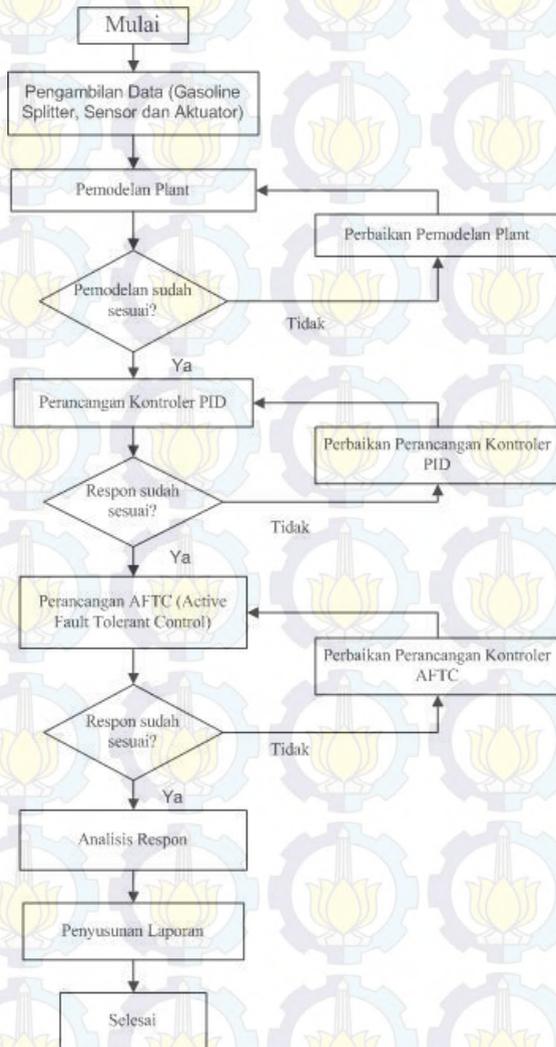
f. Analisis respon

Analisis respon dilakukan berdasarkan uji performansi sistem dengan memberikan kesalahan pada sensor dan aktuator. Kesalahan bias, sensitivitas dan *noise* diberikan untuk uji performansi ketika terjadi kesalahan sensor. Sedangkan *lost of effectiveness* dan kebocoran diberikan untuk uji performansi ketika terjadi kesalahan aktuator. Setiap pemberian jenis uji menghasilkan grafik respon sistem yang berbeda. Performansi dari grafik respon tersebut selanjutnya dibandingkan antara sistem yang menggunakan kontrol PID ditambahkan dengan algoritma AFTC dengan performansi sistem yang hanya menggunakan kontrol PID saja.

g. Penyusunan laporan Tugas Akhir

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

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



Gambar 3.1 Flowchart Pengerjaan Tugas Akhir

3.1 Pengambilan Data *Plant*

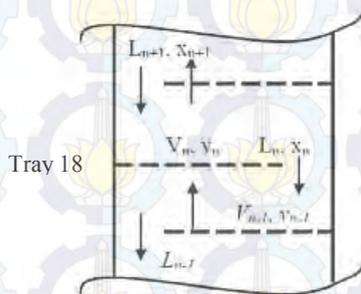
Data – data yang dibutuhkan untuk pemodelan *plant Gasoline Splitter Column* berupa spesifikasi *input plant* (*flow rate*, tekanan, temperatur serta komposisi *feed*), dimensi *gasoline splitter column*, *condenser* dan *reboiler*. Data tersebut diambil dari unit *process engineering* (PE) PT. Pertamina (Persero) RU IV Cilacap yang dituliskan pada lembar *process flow diagram* (PFD) *plant* dan spesifikasi *plant*. Rincian data yang dibutuhkan dalam pemodelan terdapat pada **Tabel 3.1** di bawah ini.

Tabel 3.1 Data Pemodelan *Plant* [17, 18]

Notasi	Keterangan	Nilai dan Satuan
$D_{splitter}$	Diameter <i>Splitter</i>	2,75 m
$H_{splitter}$	Tinggi <i>Splitter</i>	19,54 m
T_{feed}	Temperatur <i>feed</i>	116 °C
P_{feed}	Tekanan <i>feed</i>	2 kg/cm ²
D_{cond}	Diameter <i>condenser</i>	2,5 e ⁻² m
V_{cond}	Volume total <i>condenser</i>	6,426 e ⁻³ m
$D_{reboiler}$	Diameter <i>reboiler</i>	2,5 e ⁻² m
$V_{reboiler}$	Volume total <i>reboiler</i>	9,419 e ⁻⁷ m
T_{top}	Temperatur atas kolom	68 °C
T_{bottom}	Temperatur dasar kolom	136 °C
$T_{refluks}$	Temperatur <i>refluks</i>	57 °C
P_{drop}	<i>Pressure drop condenser</i>	0,3 kg/cm ²
P_{cond}	Tekanan <i>condenser</i>	1,1 kg/cm ²
$P_{reboiler}$	Tekanan <i>reboiler</i>	1,4 kg/cm ²

3.2 Pemodelan Gasoline Splitter

Gasoline Splitter Column memiliki konfigurasi yang sama dengan *Distillation Column*, sehingga dalam menurunkan model matematis dapat menggunakan persamaan untuk *Distillation Column* [9]. Temperatur yang dikendalikan pada *plant* ini berada pada tray 18 dengan mengatur besarnya laju aliran *reflux*. Oleh karena itu, pemodelan difokuskan hanya pada tray tersebut. Model dari tray 18 diadaptasi dari *general section of distillation column*, seperti Gambar 2.6. Berikut ini merupakan *control volume* untuk tray 18.



Gambar 3.2 Tray 18 Gasoline Splitter Column [9]

Perubahan temperatur pada tray 18 (T_{18}) diperoleh dengan melakukan substitusi antara pers. (2.7) dengan pers. (2.9). Sehingga terbentuk persamaan baru seperti berikut,

$$Mn \frac{d(h_n)}{dt} + hn \frac{d(M_n)}{dt} = L_{n+1} + V_{n-1} - L_n - V_n \quad (3.1)$$

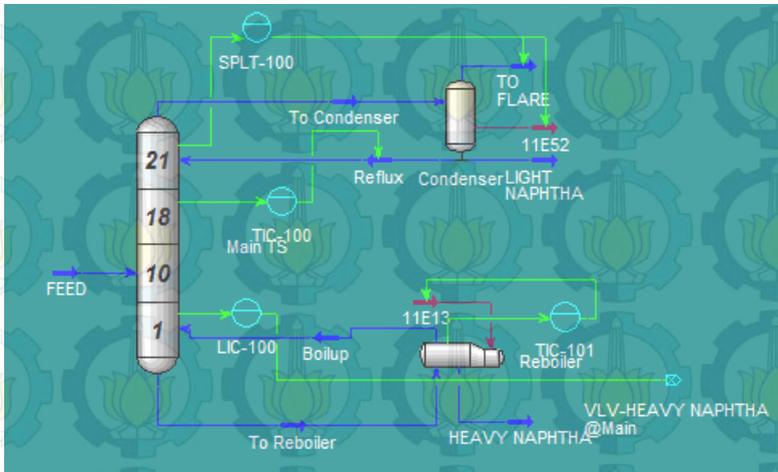
$$Mn \frac{d(h_n)}{dt} = (h_{n+1} - h_n)L_{n+1} + (H_{n-1} - h_n)V_{n-1} - (H_n - h_n)V_n \quad (3.2)$$

$$Mn C_p \frac{d(T_{18})}{dt} = (h_{n+1} - h_n)L_{n+1} + (H_{n-1} - h_n)V_{n-1} - (H_n - h_n)V_n$$

$$\frac{d(T_{18})}{dt} = \frac{(h_{n+1} - h_n)L_{n+1} + (H_{n-1} - h_n)V_{n-1} - (H_n - h_n)V_n}{MnCp} \quad (3.3)$$

Namun dalam pelaksanaannya terdapat beberapa parameter yang tidak diketahui, sehingga menimbulkan banyak asumsi. Asumsi-asumsi tersebut menimbulkan peluang ketidakcocokan yang besar saat proses validasi model, sehingga pemodelan *Gasoline Splitter Column* dilakukan dengan pendekatan FOPDT. Persamaan FOPDT tersebut dibangun dari grafik respon *input output* yang menunjukkan kinerja *plant*. Persamaan FOPDT telah banyak digunakan untuk merepresentasikan *plant* yang berorde tinggi dan fungsi transfer FOPDT dianggap dapat mewakili *real plant* [10]. Tujuan dari pemodelan ini adalah untuk mendapatkan persamaan perubahan temperatur sebagai variabel yang dikendalikan terhadap laju aliran *refluks* dari *condenser* sebagai variabel yang dimanipulasi.

Grafik respon *input output* didapatkan dari simulasi *plant* menggunakan *software* HYSYS saat kondisi *dynamic mode*. *Gasoline Splitter Column* disimulasikan menggunakan *software* HYSYS 2006. Unit operasi tersebut memiliki satu buah *input* dan dua buah *output*, *condenser*, *reboiler* serta tiga *loop* pengendalian yaitu sistem pengendalian level pada *hold up liquid* yang ada di dasar *splitter*, sistem pengendalian temperatur pada *tray* 18 dan sistem pengendalian *pressure* pada bagian atas kolom.



Gambar 3.3 Tampilan Simulasi *Plant* pada HYSYS 2006

Pada FOPDT terdapat tiga variabel karakteristik, yaitu *process gain*, *process time constant*, dan *dead time*. Ketiganya merepresentasikan perilaku dari unit operasi *Gasoline Splitter Column*.

Prosedur yang dilakukan untuk pemodelan unit operasi *Gasoline Splitter Column* adalah sebagai berikut.

- Pengambilan data *open loop* yang dilakukan dengan mengubah mode kontrol pengendali pada HYSYS dari mode auto menjadi mode manual. Mengubah *manipulated variable* (OP) pada kontroler dengan memberikan masukan *step* sebesar 20%. Pengambilan data dihentikan ketika unit proses telah stabil di keadaan baru.
- Data yang dihasilkan dari respon *open loop* tersebut digunakan untuk mendapatkan *process gain*, *process time constant*, dan *dead time*. Persamaan – persamaan yang digunakan adalah sebagai berikut.

$$K = \frac{\Delta}{\delta} \quad (3.4)$$

$$t_{28\%} = \theta + \frac{\tau}{3} \quad (3.5)$$

$$t_{63\%} = \theta + \tau \quad (3.6)$$

$$\tau = 1,5(t_{63\%} - t_{28\%}) \quad (3.7)$$

$$\theta = t_{63\%} - \tau \quad (3.8)$$

dimana,

K : gain steady state

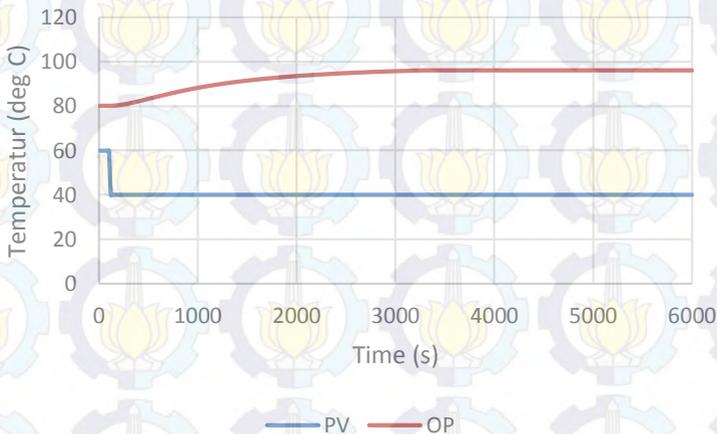
Δ : perubahan steady state pada keluaran proses

\mathcal{D} : perubahan step pada masukan

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

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

$$FOPDT = G_m(s) = K \frac{e^{-\theta s}}{\tau s + 1} \quad (3.9)$$



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

Berdasarkan data grafik pada Gambar 3.3 didapatkan hasil fungsi transfer *plant* melalui persamaan (3.6), dimana,

$$K_p = \frac{\Delta}{\delta} = \frac{15,94}{20} = 0,79 \quad (3.10)$$

$$\tau_p = 1,5(t_{63\%} - t_{28\%}) \quad (3.11)$$

$$\tau_p = 1,5(1272,4 - 623,955) = 972,67 \quad (3.12)$$

Dengan demikian, persamaan (3.6) dapat dituliskan sebagai berikut.

$$\frac{T(s)}{\dot{m}_{refluks}(s)} = \frac{0,79}{972,68s + 1} \quad (3.13)$$

3.3 Pemodelan Matematis Aktuator

Aktuator dari sistem pengendalian temperatur pada *Gasoline Splitter Column* ini adalah *control valve* dengan karakteristik *normally close* atau *failure to open* yang berfungsi untuk mengatur besarnya laju aliran *refluks*. *Input* dari *control valve* 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 *refluks* yang berfungsi mengendalikan temperatur pada *tray* 18. *Gain* aktuator didapatkan dari persamaan berikut.

$$\frac{m_b(s)}{U(s)} = \frac{K_v}{\tau_s + 1} \quad (3.14)$$

Keterangan :

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

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

K_V : *gain total control valve*

τ : *time constant control valve* (s)

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

$$K_V = \frac{\text{Laju Aliran Refluks}}{\text{Bukaan Valve}} \quad (3.15)$$

Sesuai data proses dimana ketika *valve* terbuka sebesar 57%, aliran *refluks* yang kembali ke *tray 18* sebesar 11,55 kg/s. Maka akan didapatkan nilai *gain control valve* seperti berikut ini.

$$K_V = \frac{11,55}{57} = 0,202 \quad (3.16)$$

Sedangkan untuk memperoleh nilai *time constant* dinyatakan dengan persamaan,

$$\tau = T_V(\Delta V + R_V) \quad (3.17)$$

Keterangan :

τ : *time constant control valve* (detik)

T_V : waktu *stroke* penuh (1,5 detik)

ΔV : (kuantitas maks. – kuantitas min.)/kuantitas maks.

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

$$\Delta V = \frac{Quan_{maks} - Quan_{min}}{Quan_{maks}} = \frac{15,32 - 0}{15,32} = \frac{15,32}{15,32} = 1 \quad (3.18)$$

$$\tau = T_v(\Delta V + R_v) = 1,5(1 + 0,3) = 1,95 \quad (3.19)$$

Maka berdasarkan persamaan 3.14 fungsi transfer *control valve* adalah sebagai berikut.

$$\frac{m_b(s)}{U(s)} = \frac{K_v}{\tau s + 1} = \frac{0,202}{1,95s + 1} \quad (3.20)$$

3.4 Pemodelan Matematis Sensor

Sensor dari sistem pengendalian temperatur pada *Gasoline Splitter Column* ini adalah *Termocouple* sebagai sensor untuk mengukur temperatur kolom pada *tray* 18. Pemodelan pada sensor menggunakan nilai *num* dan *denum* sebesar 1 agar nilai keluaran dari sensor menuju unit pengendali merepresentasikan nilai temperatur yang terukur.

3.5 Validasi Pemodelan

Model matematis *plant* didapatkan melalui pendekatan FOPDT. Data yang digunakan untuk membentuk fungsi transfer tersebut berasal dari respon simulasi HYSYS pada mode manual. Oleh karena itu model dapat dikatakan valid ketika data yang didapat dari HYSYS sesuai dengan data proses perusahaan. Dimana pengujian dilakukan dengan cara memberikan 5 variasi bukaan *control valve* sebagai *input* dan didapat output berupa pengukuran temperatur *tray* 18. Hasil simulasi tersebut kemudian dibandingkan dengan hasil pengukuran *real plant*. Tabel di bawah ini menunjukkan perubahan *process variable* sistem yang berubah ketika *input* diberikan ke sistem pada simulasi HYSYS.

Tabel 3.2 Data Pengukuran di PT.Pertamina
(Persero) RU IV Cilacap dengan OP 44,7%

Waktu, Tanggal	OP (%)	PV - Real (deg C)
08/01/2015 18:12	44,737	98,013
10/01/2015 15:52	44,737	97,751
11/01/2015 8:46	44,737	97,828
18/01/2015 1:18	44,737	96,060
19/01/2015 13:27	44,737	99,607
23/01/2015 10:19	44,737	96,584
27/01/2015 18:30	44,737	96,369
03/02/2015 11:00	44,737	97,616
05/02/2015 8:00	44,737	97,024
08/02/2015 0:50	44,737	96,875
Rata-rata PV		97,373

Tabel 3.3 Data Pengukuran di PT.Pertamina
(Persero) RU IV Cilacap dengan OP 47,36%

Waktu, Tanggal	OP (%)	PV - Real (deg C)
06/01/2015 13:28	47,368	96,641
07/01/2015 8:00	47,368	97,441
08/01/2015 13:13	47,368	97,820
09/01/2015 23:27	47,368	96,820
13/01/2015 19:38	47,368	95,052
16/01/2015 20:25	47,368	98,760
17/01/2015 8:52	47,368	97,076
18/01/2015 4:05	47,368	96,007
19/01/2015 0:21	47,368	96,235
07/02/2015 10:10	47,368	97,561
Rata-rata PV		96,941

Tabel 3.4 Data Pengukuran di PT.Pertamina
(Persero) RU IV Cilacap dengan OP 52,6%

Waktu, Tanggal	OP (%)	PV - Real (deg C)
13/01/2015 16:49	52,632	94,393
26/01/2015 6:08	52,632	94,730
31/01/2015 5:10	52,632	93,119
02/02/2015 10:30	52,632	94,196
05/02/2015 9:19	52,632	91,774
11/02/2015 7:02	52,632	93,052
21/02/2015 9:07	52,632	92,913
05/03/2015 8:09	52,632	90,273
08/03/2015 19:37	52,632	90,709
10/03/2015 1:03	52,632	89,726
Rata-rata PV		92,795

Tabel 3.5 Data Pengukuran di PT.Pertamina
(Persero) RU IV Cilacap dengan OP 57,8%

Waktu, Tanggal	OP (%)	PV - Real (deg C)
03/01/2015 9:16	57,895	88,034
04/01/2015 13:15	57,895	88,377
05/01/2015 0:50	57,895	83,728
14/01/2015 8:53	57,895	86,098
16/01/2015 2:30	57,895	88,819
22/01/2015 9:52	57,895	88,842
26/01/2015 7:43	57,895	85,638
11/02/2015 7:08	57,895	84,002
13/02/2015 20:02	57,895	87,740
20/02/2015 14:01	57,895	88,352
Rata-rata PV		86,963

Tabel 3.6 Data Pengukuran di PT.Pertamina
(Persero) RU IV Cilacap dengan OP 63,1%

Waktu, Tanggal	OP (%)	PV - Real (deg C)
04/01/2015 14:46	63,15789	85,197
13/02/2015 16:12	63,15789	81,496
13/03/2015 23:04	63,15789	87,918
Rata-rata PV		84,870

Tabel 3.7 Data Pengukuran dari Simulasi Sistem pada
HYSYS

Input OP (%)	Temperature (°C)
44,7	95,2
47,3	94,1
52,6	91,9
57,8	87,7
63,1	86,1

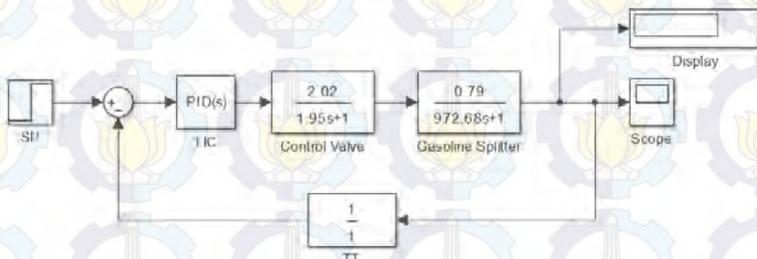
Tabel 3.8 Perhitungan *Error* Hasil Pengukuran Simulasi
HYSYS dan Hasil Pengukuran Perusahaan

OP (%)	PV - Real (°C)	PV - HYSYS (°C)	Error
44,7	97,373	95,2	-2,23142
47,3	96,941	94,1	-2,93097
52,6	92,795	91,96	-0,90028
57,8	86,963	87,7	0,847453
63,1	84,870	86,12	1,472311
<i>Error Rata-Rata</i>			-0,74858

Berdasarkan perbandingan kedua data tersebut model dapat dikatakan valid karena data yang digunakan dalam membentuk fungsi transfer mendekati data *process variable* perusahaan.

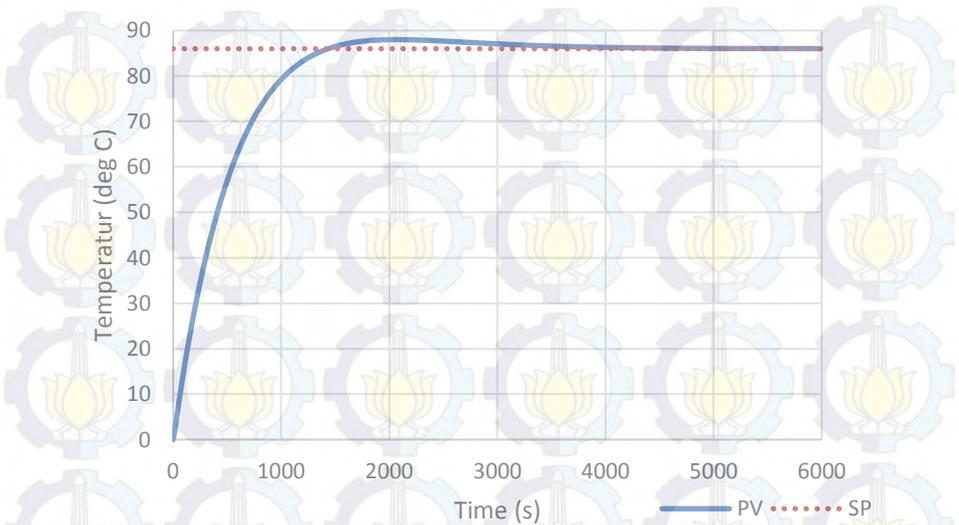
3.6 Perancangan Kontrol *Proportional Integral Derivative* (PID)

Kontrol PID 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, I dan D didapatkan dari hasil simulasi diagram blok sistem pengendalian tersebut.



Gambar 3.5 Blok Simulink pada Simulasi MATLAB *Closed Loop*

Tuning dilakukan menggunakan metode *trial and error*, sehingga didapatkan nilai gain *proportional* sebesar 1,2 , gain *integral* sebesar 0,0016 dan gain *derivative* sebesar 4,6. Hasil grafik menunjukkan bahwa *maximum overshoot* sebesar 2,29%, *settling time* sebesar 4561 detik, dan *error steady state* sebesar 0,00073%. Untuk hasil *tuning trial and error* didapatkan grafik respon sistem pengendalian *closed loop* sebagai berikut.



Gambar 3.6 Grafik Respon Simulasi Sistem Pengendalian *Closed Loop Gasoline Splitter Column*

3.7 Perancangan *Active Fault Tolerant Control (AFTC)*

Terdapat dua tahapan dalam perancangan AFTC, yaitu merancang *observer* dan merekonfigurasi sinyal kontrol.

3.7.1 Perancangan *Observer* untuk Sensor

Perancangan *observer* digunakan untuk mengestimasi kesalahan pembacaan pada sensor temperatur. Perancangan dilakukan dengan mengubah fungsi transfer hasil dari perkalian fungsi transfer *plant*, aktuator, dan sensor ke dalam persamaan ruang keadaan (*state space*).

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

$$Y = CX \quad (3.22)$$

Matriks dari persamaan ruang keadaan (*state space*) didapat dengan *function* 'tf2ss' pada program Matlab.

$$\begin{aligned} \text{num} &= [1.595] \\ \text{den} &= [1896.726 \quad 974.63 \quad 1] \end{aligned}$$

sehingga didapat matriks persamaan ruang keadaan sebagai berikut.

$$A = \begin{bmatrix} -0.5138 & -0.000527 \\ 1 & 0 \end{bmatrix}$$

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

$$C = [0 \quad 0.00084]$$

$$D = [0]$$

A merupakan matriks pada *state*, B merupakan matriks pada kontrol *input*, C merupakan matriks pada hasil pengukuran dan D merupakan matriks transmisi langsung. Sedangkan \dot{X} merupakan integral *state vector* dan Y merupakan *output vector*. Dengan adanya kesalahan pada sensor, maka persamaan ruang keadaan akan menjadi persamaan (3.23) dan (3.24)

$$\dot{X} = AX + BU + E_S f_S + D_S d \quad (3.23)$$

$$\dot{Z} = CX + F f_S + D_\omega \omega - Z \quad (3.24)$$

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

Dengan asumsi bahwa,

$$F = D_\omega$$

$$Y = Z$$

F merupakan matriks dari kesalahan sensor, f_S merupakan *vector* dari kesalahan sinyal yang dihasilkan oleh

sensor, D_ω merupakan matriks dari *noise* dan D_S merupakan matriks dari *load*. Sedangkan \dot{Z} merupakan integral *error vector*. Selanjutnya dilakukan estimasi kesalahan *observer* dari hasil pengukuran. Estimasi kesalahan tersebut didapatkan melalui pendekatan persamaan (3.26), (3.27), dan (3.28) berikut ini.

$$\dot{\hat{X}}_S = A_S \hat{X}_S + B_S U + L(Y - \hat{Y}) \quad (3.26)$$

$$Y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \\ f_S \end{bmatrix} \quad (3.27)$$

$$f_S = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{X}_S \quad (3.28)$$

$\dot{\hat{X}}_S$ merupakan *state estimate vector*, \hat{Y} merupakan *output estimate vector* dan L merupakan *gain* untuk *observer*. Berdasarkan persamaan (3.26), (3.27), dan (3.28), maka ditambahkan matriks tambahan untuk persamaan estimasi kesalahan sebagai berikut.

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

Persamaan estimasi kesalahan dapat disederhanakan menjadi persamaan (3.29).

$$\dot{X}_S = A_S X_S + B_S U + D_S d_D \quad (3.29)$$

Untuk mengestimasi kesalahan pada hasil pengukuran dilakukan pendefinisian matriks – matriks berikut.

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

$$A_S = \begin{bmatrix} -0,5138 & -0,000527 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0,00084 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

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

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

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

$$Y = [C \ 0] \begin{bmatrix} X \\ f_a \end{bmatrix} + D_\omega \cdot e_s \quad (3.30)$$

$$Y = C_S \cdot X_S + D_\omega \cdot e_s \quad (3.31)$$

Error didapatkan melalui persamaan sebagai berikut.

$$\dot{e}_s = \dot{X}_a - \hat{X}_a \quad (3.32)$$

$$\dot{e}_s = A_s e_x + D_s d_D - L(C_s - e_s) \quad (3.33)$$

$$\dot{e}_s = (A_s - L_s C_s) e_s + D_s d_D \quad (3.34)$$

Setelah matriks persamaan ruang keadaan didapatkan, matriks digunakan untuk mencari *gain observer* (L) menggunakan metode *Linear Matrix Inequality* (LMI) melalui analisa kestabilan Lyapunov. Melalui analisa kestabilan Lyapunov metode kedua, maka penyelesaian dalam LMI dapat ditulis pada persamaan (3.30).

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

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

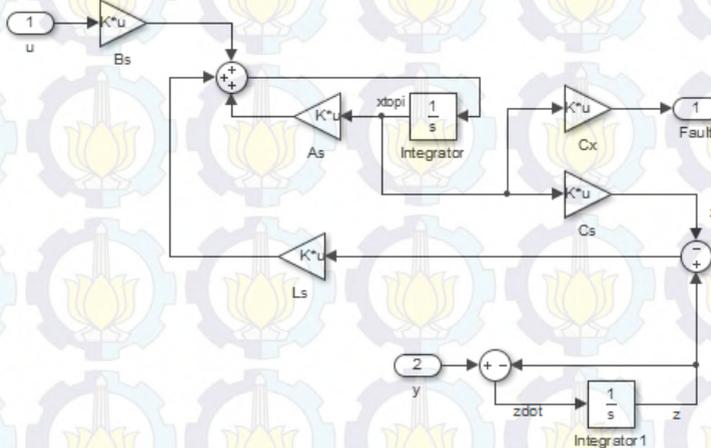
Selanjutnya untuk melakukan iterasi nilai L , digunakan *function* LMI Solver dengan metode *Minimization*, sehingga *command* yang digunakan adalah sebagai berikut.

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

Maka didapatkan matriks L sebagai berikut.

$$L_s = \begin{bmatrix} 0.000556 \\ 0.000216 \\ 0.6702 \\ 1.93 \end{bmatrix}$$

Jika persamaan estimasi kesalahan dibuat dalam program *simulink* MATLAB, maka akan tampak seperti Gambar 3.7.

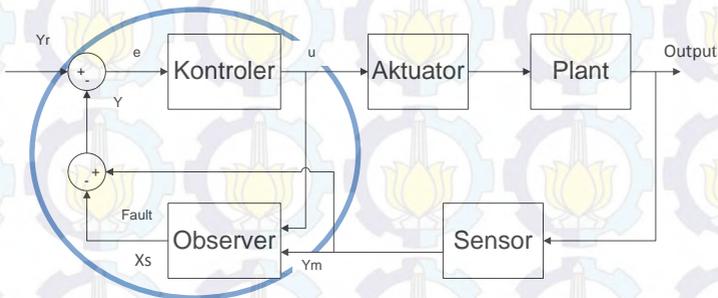


Gambar 3.7 Blok Simulink Struktur *Observer* untuk Sensor

Gambar 3.7 merupakan *representasi* dari persamaan (3.26) Dimana u merupakan sinyal kontrol dari *controller* yang masuk ke *observer* dan y merupakan hasil pengukuran sensor yang telah dipengaruhi oleh kesalahan pembacaan. Output dari *observer* tersebut ialah besarnya kesalahan pembacaan sensor.

3.7.2 Rekonfigurasi Sinyal Kontrol untuk Sensor

Rekonfigurasi sinyal kontrol merupakan tahap akhir pada 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 kontrol akibat kesalahan sensor ditunjukkan oleh lingkaran biru sebagai berikut.



Gambar 3.8 Blok Diagram Rekonfigurasi Sinyal Kontrol dari *Observer* Sensor

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

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

dimana,

Y : nilai sebenarnya ($^{\circ}\text{C}$)

Y_m : nilai hasil pengukuran ($^{\circ}\text{C}$)

X_S : nilai *fault* ($^{\circ}\text{C}$)

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

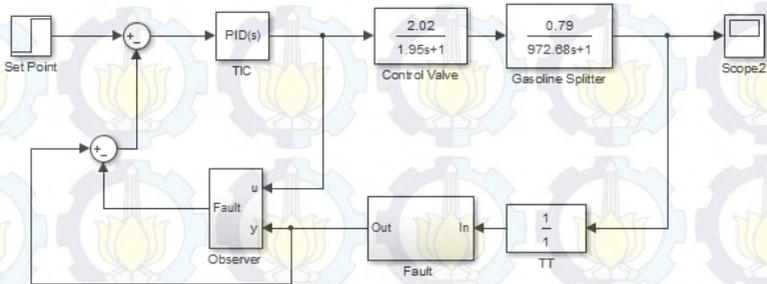
$$e = Y_r - Y \quad (3.37)$$

dimana,

e : nilai *error*

Y_r : nilai *setpoint* ($^{\circ}\text{C}$)

Dengan demikian dalam aplikasi simulink software Matlab (2013a), skema blok diagram simulasi rekonfigurasi kontrol atau sistem dengan AFTC untuk kesalahan pada sensor akan tampak pada Gambar 3.9.



Gambar 3.9 Blok Simulink Simulasi sistem AFTC dengan Kesalahan Sensor

3.7.3 Perancangan *Observer* untuk Aktuator

Perancangan *observer* digunakan untuk mengestimasi kesalahan pada aktuator. Perancangan dilakukan dengan mengubah fungsi transfer hasil dari perkalian fungsi transfer *plant*, aktuator, dan sensor ke dalam persamaan ruang keadaan

(*state space*). Matriks dari persamaan ruang keadaan (*state space*) didapat dengan *function* 'tf2ss' pada program Matlab.

$$\text{num} = [1.595]$$

$$\text{den} = [1896.726 \quad 974.63 \quad 1]$$

sehingga didapat matriks persamaan ruang keadaan sebagai berikut.

$$A = \begin{bmatrix} -0.5138 & -0.000527 \\ 1 & 0 \end{bmatrix}$$

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

$$C = [0 \quad 0.00084]$$

$$D = [0]$$

A merupakan matriks pada *state*, B merupakan matriks pada kontrol *input*, C merupakan matriks pada hasil pengukuran dan D merupakan matriks transmisi langsung. Sedangkan \dot{X} merupakan integral *state vector* dan Y merupakan *output vector*. Dengan adanya kesalahan pada aktuator, maka persamaan ruang keadaan akan menjadi persamaan (3.38) dan (3.39).

$$\dot{X} = AX + BU + E.f_a + D_d.d \quad (3.38)$$

$$Y = CX + F.f_s + D_\omega.\omega \quad (3.39)$$

Dengan asumsi bahwa,

$$F = D_\omega$$

$$E = B$$

E merupakan matriks dari kesalahan aktuator dan f_a merupakan *vector* dari kesalahan sinyal yang dihasilkan oleh aktuator. Maka pada kesalahan aktuator, persamaan (3.39) akan menjadi persamaan (3.40)

$$\begin{aligned} Y &= CX + D_\omega(f_s + \omega) \\ Y &= CX + D_\omega e_s \end{aligned} \quad (3.40)$$

Setelah matriks persamaan ruang keadaan didapatkan, matriks digunakan untuk merancang *observer* sebagai *fault estimate* atau elemen yang mengestimasi kesalahan yang terjadi. *Observer* dirancang melalui pendekatan persamaan (3.41), (3.42), dan persamaan (3.43).

$$\dot{\hat{X}}_a = A_a \hat{X}_a + B_a U + L(Y - \hat{Y}) \quad (3.41)$$

$$\hat{Y} = C_a \hat{X}_a \quad (3.42)$$

$$f_a = [0 \quad 1] \hat{X}_a \quad (3.43)$$

\hat{X}_a merupakan *state estimate vector*, \hat{Y} merupakan *output estimate vector* dan L merupakan *gain* untuk *observer*. Berdasarkan persamaan (3.41), (3.42), dan (3.43), maka ditambahkan matriks tambahan untuk persamaan estimasi kesalahan sebagai berikut.

$$\begin{bmatrix} \dot{X} \\ \dot{f}_a \end{bmatrix} = \begin{bmatrix} A & E \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ f_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U + \begin{bmatrix} D_d & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} d \\ f_a \end{bmatrix} \quad (3.44)$$

$$\dot{\hat{X}}_a = A_a \hat{X}_a + B U + D_a \cdot d_D \quad (3.45)$$

$$Y = [C \quad 0] \begin{bmatrix} X \\ f_a \end{bmatrix} + D_\omega e_s \quad (3.46)$$

$$Y = C_a \cdot X_a + D_\omega \cdot e_s \quad (3.47)$$

Dengan demikian diketahui bahwa,

$$A_a = \begin{bmatrix} -0.5138 & -0.000527 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B_a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C_a = [0 \quad 0.00084 \quad 0]$$

$$D_d = B \quad D_\omega = [1]$$

Maka persamaan (3.41), (3.42), dan (3.43), sebagai pendekatan estimasi *fault* pada *observer* dapat dituliskan sebagai berikut.

$$\dot{\hat{X}}_a = \begin{bmatrix} -0.5138 & -0.000527 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{X}_a + \begin{bmatrix} 1 \\ 0 \end{bmatrix} U + L(Y - \hat{Y}) \quad (3.48)$$

$$\hat{Y} = [0 \quad 0.00084] \hat{X}_a \quad (3.49)$$

$$f_a = [0 \quad 1] \hat{X}_a \quad (3.50)$$

Dengan demikian *observer* sebagai estimator kesalahan akan memiliki keluaran yaitu hasil pembacaan kesalahan (f_a).

Selanjutnya persamaan (3.41), (3.42), dan (3.43), dibuat dalam program *simulink* di MATLAB untuk dilakukan simulasi. Gambar blok diagram program *observer* akan tampak pada Gambar 3.10 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;

```

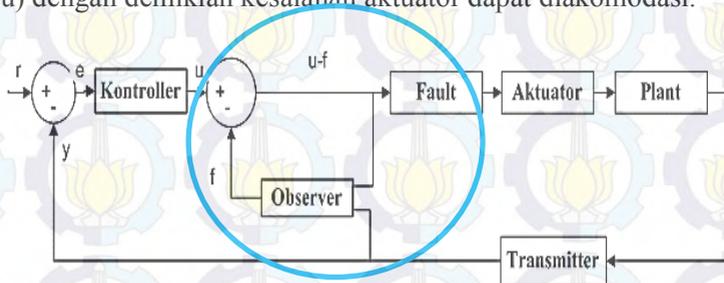
Maka didapatkan matriks L sebagai berikut.

$$L = \begin{bmatrix} 4.1260 \\ 141.7367 \\ 2.2083 \end{bmatrix}$$

3.8.4 Rekonfigurasi Sinyal Kontrol untuk Aktuator

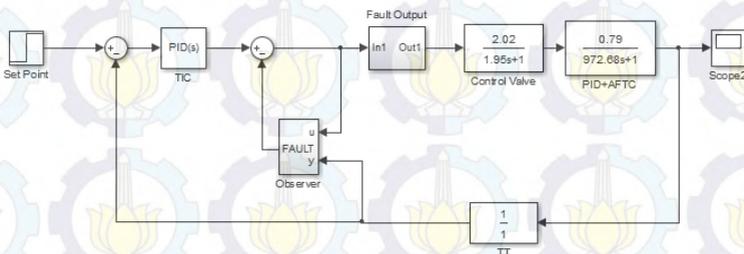
Rekonfigurasi kontrol digunakan untuk mengakomodir kesalahan sehingga *plant* tetap beroperasi. Kesalahan yang terjadi pada aktuator menyebabkan sinyal kontrol (u) yang seharusnya masuk ke aktuator berubah menjadi kesalahan (*fault*) yang didefinisikan dengan ($fault + u$). Agar sinyal yang masuk ke aktuator adalah sinyal kontrol (u) maka hasil estimasi

kesalahan dari *observer* akan dikurangkan setelah sinyal kontrol (u) dengan demikian kesalahan aktuator dapat diakomodasi.



Gambar 3.11 Blok Diagram Rekonfigurasi Sinyal Kontrol dari *Observer* Aktuator

Dengan demikian dalam aplikasi simulink software Matlab (2013a), skema blok diagram simulasi rekonfigurasi kontrol atau sistem dengan AFTC untuk kesalahan pada aktuator akan tampak pada Gambar 3.12.

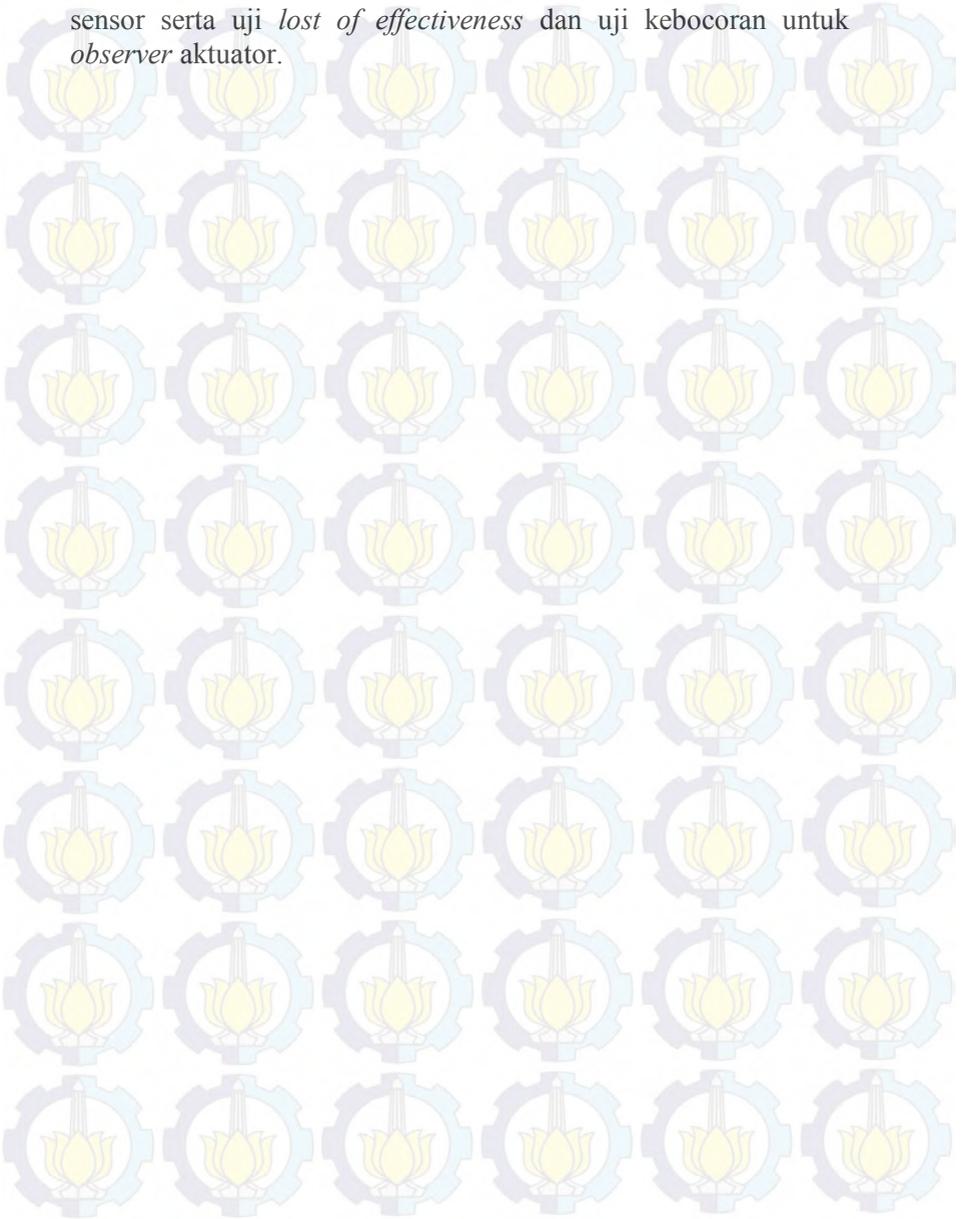


Gambar 3.12 Blok Simulink Simulasi sistem AFTC dengan Kesalahan Aktuator

3.8 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.



BAB IV

ANALISIS DATA DAN PEMBAHASAN

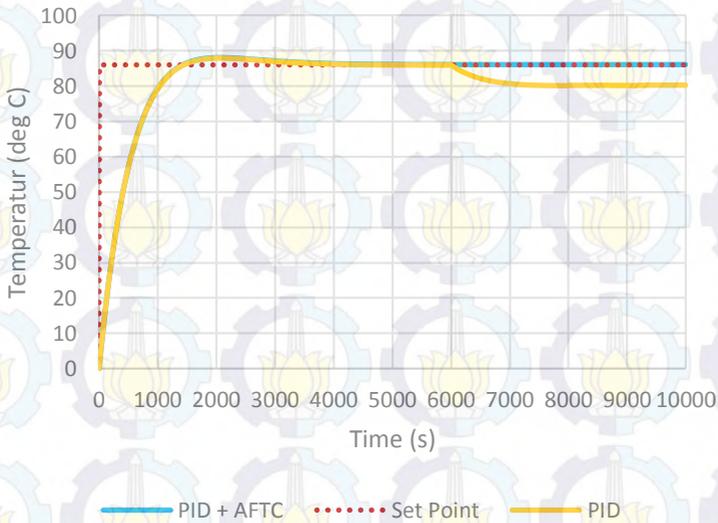
Uji performansi bertujuan untuk mengetahui algoritma AFTC (*Active Fault Tolerant Control*) yang telah dirancang dapat bekerja dengan baik atau tidak jika diberikan kesalahan. Bentuk uji kesalahan yang diberikan terhadap sistem terdiri dari kesalahan pada sensor dan pada aktuator.

4.1 Uji Kesalahan Sensor

Untuk menguji performansi dari sistem pengendalian temperatur dengan kesalahan pada sensor, maka diberikan uji kesalahan berupa kesalahan bias, sensitivitas, dan *noise* pada detik ke-6000.

4.1.1 Uji Kesalahan Bias

Untuk menguji performansi dari sistem pengendalian temperatur maka diberikan kesalahan pada sensor temperatur berupa kesalahan bias. Kesalahan bias ialah kesalahan yang diakibatkan oleh penyimpangan nilai pembacaan dari pengukuran yang sebenarnya. Nilai untuk uji kesalahan bias didapatkan dari batas toleransi sensor. Untuk sensor *termocouple type J* toleransi penyimpangan nilai sebesar $\pm 0,75\%$ dari *span* sensor. *Span* sensor tersebut ialah 750°C , oleh karena itu dalam pengujian kesalahan bias nilai yang diberikan pada blok *fault* sebesar $+5,7^{\circ}\text{C}$ dan $-5,7^{\circ}\text{C}$. Hasil respon sistem dengan kesalahan bias $+0,75\%$ dapat dilihat pada gambar 4.1 berikut.

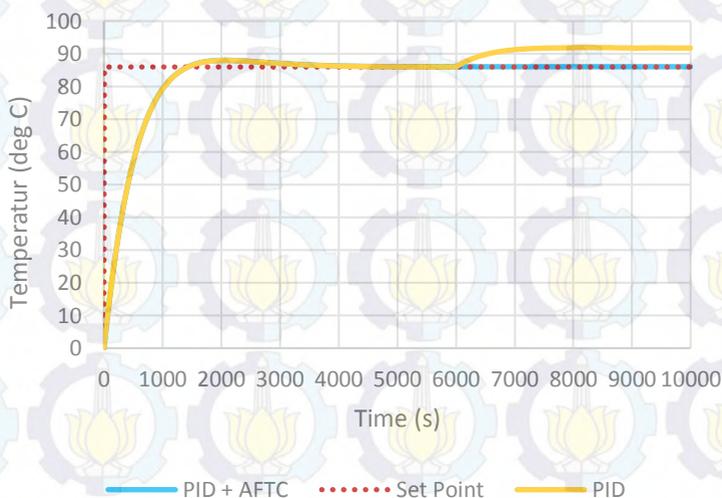


Gambar 4.1 Respon Sistem dengan Kesalahan Bias $+0,75\%$

Tabel 4.1 Parameter Performansi dengan Kesalahan Bias $+0,75\%$

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,05%
2	<i>Maximum Undershoot</i>	6,78%	0%
3	<i>Error Steady State</i>	6,61%	0,05%
4	<i>Settling Time</i>	9319 detik	7059 detik

Hasil respon sistem dengan kesalahan bias $-0,75\%$ dapat dilihat pada gambar 4.2 berikut.



Gambar 4.2 Respon Sistem dengan Kesalahan Bias $-0,75\%$

Tabel 4.2 Parameter Performansi dengan Kesalahan Bias $-0,75\%$

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	6,91%	0,06%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	6,69%	0,05%
4	<i>Settling Time</i>	9579 detik	7109 detik

Gambar 4.1 dan 4.2 menunjukkan bahwa respon sistem pengendalian temperatur dengan algoritma *active fault tolerant control* dapat mengakomodasi kesalahan bias sebesar $\pm 0,75\%$ yang diberikan pada detik ke -6000 untuk kembali mendekati

titik nominalnya yaitu 86°C. Hal ini disebabkan karena adanya rekonfigurasi kontrol yang bekerja untuk mengakomodasi kesalahan tersebut, sedangkan respon sistem pengendalian tanpa *algoritma AFTC* mengalami penurunan dan kenaikan yang tidak sesuai dengan *set point*. Berdasarkan analisis uji kesalahan bias, sistem pengendalian temperatur yang menggunakan *algoritma active fault tolerant control* memiliki performansi yang lebih baik dibandingkan dengan tanpa menggunakan *algoritma active fault tolerant control*. Hal tersebut dapat dilihat dari beberapa parameter seperti *Maximum Overshoot* dan *Error Steady State* dari kedua sistem pengendalian. Pada tabel 4.2 terlihat bahwa *maximum overshoot* PID lebih besar jika dibandingkan PID dengan tambahan AFTC yaitu 6,69% dan waktu yang diperlukan untuk mencapai *steady* lebih lama yaitu 9579 detik.

4.1.2 Uji Kesalahan Sensitivitas

Pemberian uji kesalahan selain kesalahan bias yaitu dengan uji kesalahan sensitivitas. Kesalahan sensitivitas yang diberikan sebesar 90%, 80%, 70%, dan 60%. Kesalahan sensitivitas merupakan kesalahan pembacaan sensor yang diakibatkan hilangnya sinyal yang diterima oleh sensor tersebut. Pemberian kesalahan dilakukan dengan perhitungan dengan persamaan (4.1).

$$U_r = \alpha U_o - \beta \quad (4.1)$$

Keterangan:

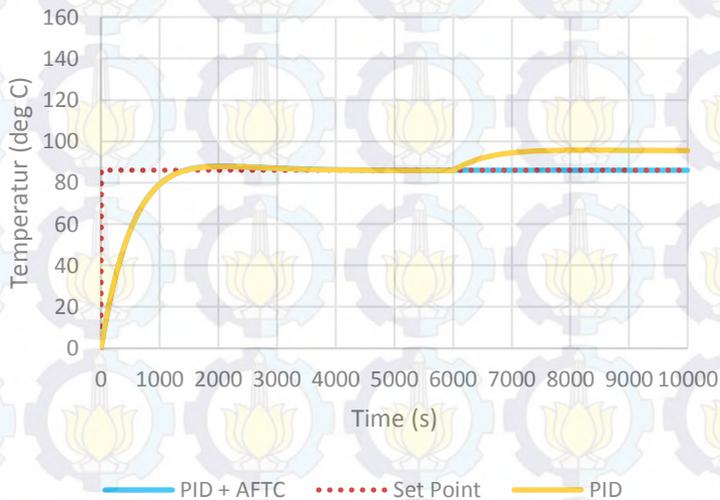
U_r = sinyal kontrol *input residual* (sinyal kontrol *input* yang masuk pada sensor setelah diberi pembebanan berupa *gain* kesalahan)

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

α = *gain* kesalahan sensitivitas

β = *gain* kesalahan Bias

Hasil respon sistem pengendalian temperatur dengan kesalahan sensitivitas 90% ditunjukkan pada gambar 4.3 berikut.



Gambar 4.3 Respon Sistem dengan Kesalahan Sensitivitas 90%

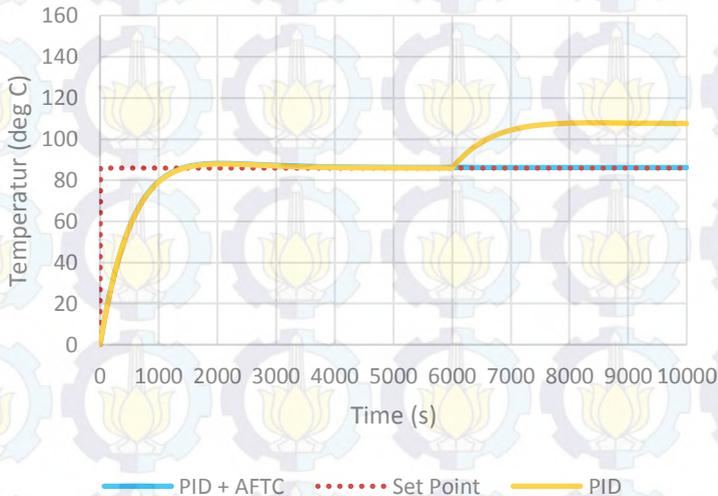
Tabel 4.3 Parameter Performansi dengan Kesalahan Sensitivitas 90%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	11,49%	0,06%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	11,2%	0,05%
4	<i>Settling Time</i>	9579 detik	7215 detik

Pada Gambar 4.3 dan Tabel 4.3 menunjukkan respon sistem ketika diberi kesalahan sensitivitas sebesar 90%, dimana kesalahan sensitivitas 90% berarti sensor kehilangan sinyal sebesar 10% dari sinyal *output*. *Maximum overshoot* pada

pengendali PID dengan AFTC lebih kecil dari pengendali PID tanpa AFTC yaitu 0,06%. Sedangkan waktu untuk mencapai steady (*settling time*) lebih cepat pada saat sistem pengendali PID yang dilengkapi dengan AFTC yaitu pada detik ke-7250. Hal ini membuktikan bahwa pengendali PID yang dilengkapi dengan AFTC dapat meningkatkan performansi dari suatu sistem.

Hasil respon sistem pengendalian temperatur dengan kesalahan sensitivitas 80% ditunjukkan pada gambar 4.4 berikut.



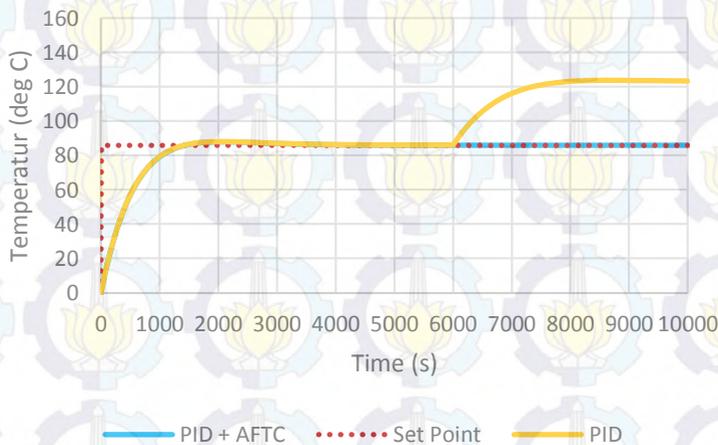
Gambar 4.4 Respon Sistem dengan Kesalahan Sensitivitas 80%

Tabel 4.4 Parameter Performansi dengan Kesalahan Sensitivitas 80%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	25,65%	0,08%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	25,2%	0,05%
4	<i>Settling Time</i>	9612 detik	7302 detik

Pada Gambar 4.4 dan Tabel 4.4 menunjukkan respon sistem ketika diberi kesalahan sensitivitas 80%. *Maximum overshoot* pada pengendali PID dengan AFTC menunjukkan nilai 0,08% dengan *settling time* 7302 detik. Sedangkan pengendali PID tanpa AFTC menunjukkan nilai *maximum overshoot* sebesar 25,65% dengan *setling time* 9612 detik. Pengendali PID dengan AFTC menunjukkan respon yang lebih baik karena memiliki nilai *maximum overshoot* yang kecil dan waktu yang diperlukan repon agar steady relatif lebih cepat dibanding sistem pengendali PID tanpa AFTC. Hal ini membuktikan bahwa pengendali PID yang dilengkapi dengan AFTC dapat meningkatkan performansi dari suatu sistem.

Hasil respon sistem pengendalian temperatur dengan kesalahan sensitivitas 70% ditunjukkan pada gambar 4.5 berikut.



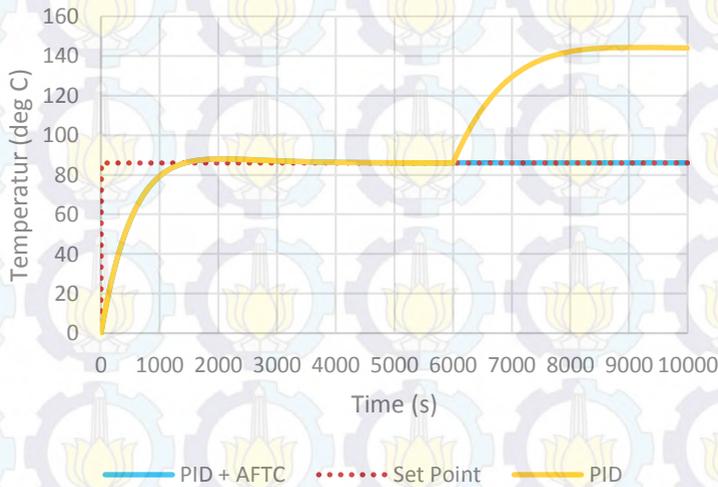
Gambar 4.5 Respon Sistem dengan Kesalahan Sensitivitas 70%

Tabel 4.5 Parameter Performansi dengan Kesalahan Sensitivitas 70%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	43,8%	0,1%
2	<i>Maximum Undershoot</i>	0%	0%
3	<i>Error Steady State</i>	43,27%	0,05%
4	<i>Settling Time</i>	9709 detik	7412 detik

Pada Gambar 4.5 dan Tabel 4.5 menunjukkan respon sistem ketika diberi kesalahan sensitivitas sebesar 70%. *Maximum overshoot* pada pengendali PID yang dilengkapi dengan AFTC menunjukkan nilai 0,1% dengan *settling time* 7412 detik. Sedangkan pengendali PID tanpa AFTC menunjukkan nilai *maximum overshoot* sebesar 43,8% dengan *settling time* 9709 detik. Pengendali PID dengan AFTC menunjukkan respon yang lebih baik karena memiliki nilai *maximum overshoot* dan *settling time* yang lebih kecil. Hal ini membuktikan bahwa pengendali PID yang dilengkapi dengan AFTC dapat meningkatkan performansi dari suatu sistem.

Hasil respon sistem pengendalian temperatur dengan kesalahan sensitivitas 60% ditunjukkan pada gambar 4.6 berikut.



Gambar 4.6 Respon Sistem dengan Kesalahan Sensitivitas 60%

Tabel 4.6 Parameter Performansi dengan Kesalahan Sensitivitas 60%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	67,82%	0,12 %
2	<i>Maximum Undershoot</i>	0 %	0%
3	<i>Error Steady State</i>	67,47%	0,05%
4	<i>Settling Time</i>	9712detik	7503 detik

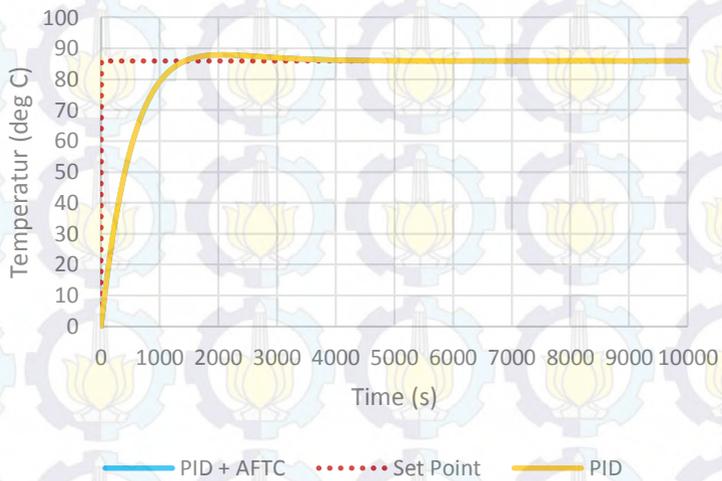
Pada Gambar 4.6 dan Tabel 4.6 menunjukkan respon sistem ketika diberi kesalahan sensitivitas sebesar 60%. *Maximum overshoot* pada pengendali PID dengan AFTC lebih kecil dari pengendali PID tanpa AFTC yaitu 0,12%. Sedangkan waktu untuk mencapai *steady* (*settling time*) lebih cepat pada saat

sistem pengendali PID yang dilengkapi dengan AFTC yaitu pada detik ke-7503. Hal ini membuktikan bahwa pengendali PID yang dilengkapi dengan AFTC dapat meningkatkan performansi dari suatu sistem.

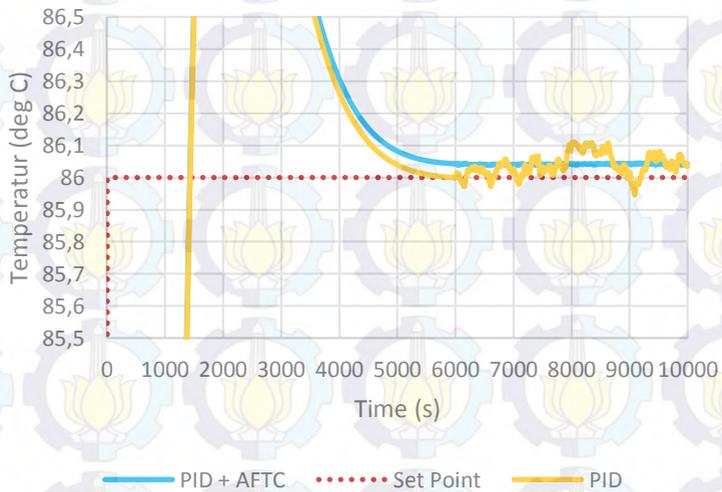
Secara keseluruhan, hasil respon sistem kontrol temperatur dengan algoritma *active fault tolerant control* menunjukkan bahwa walaupun terjadi kesalahan berupa ketidaksensitivitasan sensor dalam bekerja, temperatur tetap pada kondisi nominal atau *setpointnya*. Hal ini dikarenakan adanya rekonfigurasi kontrol yang diberikan oleh algoritma *active fault tolerant control*. Kesalahan aksi kontrol terjadi akibat ketidaksempurnaan sinyal yang diterima oleh sensor temperatur. *Maximum Oershoot* terbesar terjadi pada uji kesalahan sensitivitas 60%. Hal tersebut dikarenakan terdapat sinyal yang hilang sebesar 40%. Namun pada kondisi tersebut sistem pengendali yang ditambahkan algoritma *AFTC* masih mampu mengatasinya.

4.1.3 Uji Kesalahan Noise

Uji *noise* ini dilakukan dengan menambahkan gangguan atau *noise* pada hasil sistem pengukuran yaitu gangguan bertipe *Gaussian noise* dengan *mean* sebesar 0 dan *variance* sebesar 21,096%, sehingga sinyal *output* dari sistem mengandung *noise*. Nilai tersebut didapatkan dari data pengukuran sensor temperatur pada saat sensor tersebut masih bekerja dalam keadaan optimal atau sebelum dilakukan *repairing* atau *replacement*, sehingga didapatkan nilai standar deviasinya yaitu sebesar 4,593, nilai ini kemudian dikuadratkan yang merupakan nilai *variance* sebagai gangguan yang diberikan pada sensor temperatur. Hasil respon sistem pengendalian temperatur dapat dilihat pada gambar 4.7 berikut.



Gambar 4.7 Respon Sistem dengan Pemberian *Noise*



Gambar 4.8 Perbesaran Respon Sistem dengan Pemberian *Noise*

Ripple yang disebabkan oleh *noise* mengakibatkan sistem menjadi tidak stabil. Namun terlihat pada Gambar 4.7 dan gambar 4.8, respon sistem pengendalian temperatur dengan algoritma *active fault tolerant control* menunjukkan bahwa saat diberikan uji *noise* dengan *variance* sebesar 21,096%, algoritma *active fault tolerant control* tetap bisa mengakomodasi kesalahan berupa *noise* tersebut untuk kembali menuju pada keadaan normalnya yaitu sebesar 86 °C, sedangkan pada respon sistem pengendalian temperatur yang tidak menggunakan algoritma *active fault tolerant control*, tidak mencapai kondisi mantapnya yaitu sebesar 85,95°C.

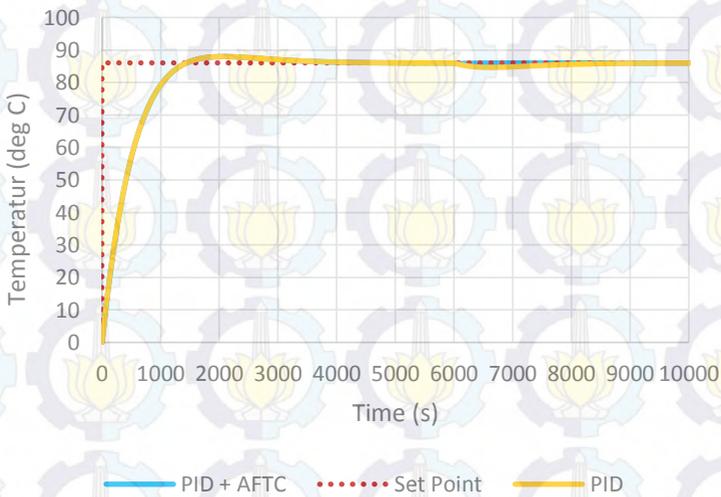
4.2 Uji Kesalahan Aktuator

Untuk menguji performansi dari sistem pengendalian temperatur dengan kesalahan pada aktuator, maka diberikan uji kesalahan berupa *lost of effectiveness* dan kebocoran pada detik ke-6000.

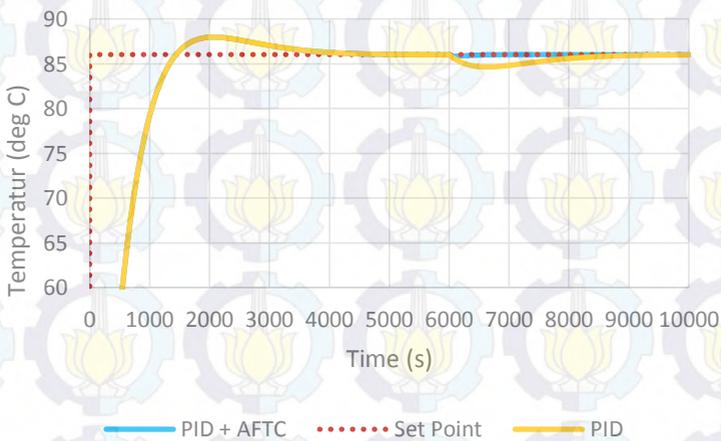
4.2.1 Uji Kebocoran

Kebocoran merupakan kesalahan pada sistem akibat aliran *refluks* yang kembali ke *top Gasoline Splitter Column* tidak terumpan secara penuh. Untuk menguji performansi dari sistem pengendalian temperatur maka diberikan kesalahan kebocoran sebesar 30%, 50% dan 70%, dari nilai laju aliran *refluks* yang disalurkan dari *control valve*, yaitu 11,55 kg/s.

Hasil respon sistem pengendalian temperatur dengan kesalahan kebocoran 30% ditunjukkan pada gambar 4.9 berikut.



Gambar 4.9 Respon Sistem dengan Kesalahan Kebocoran 30%



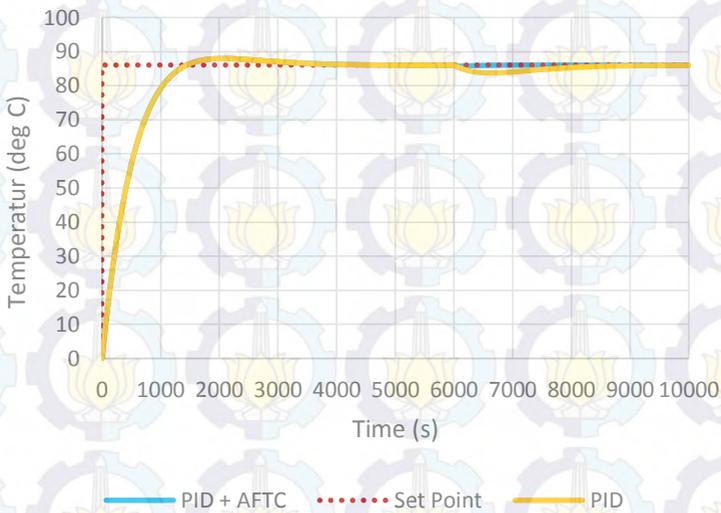
Gambar 4.10 Perbesaran Respon Sistem dengan Kesalahan Kebocoran 30%

Tabel 4.7 Parameter Performansi dengan Kesalahan Kebocoran 30%

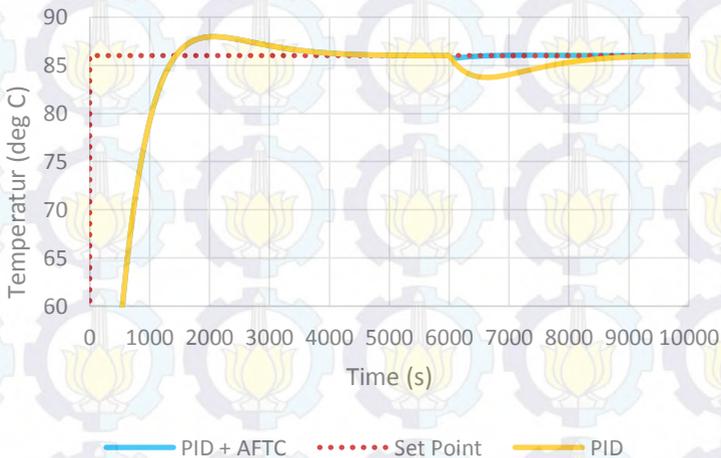
No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,03%
2	<i>Maximum Undershoot</i>	1,57%	0,15%
3	<i>Error Steady State</i>	0,01%	0,001%
4	<i>Settling Time</i>	8913 detik	6673 detik

Berdasarkan hasil dari perbesaran pada Gambar 4.10 dan data pada Tabel 4.7. diketahui bahwa sistem mengalami osilasi kembali ketika diberikan kesalahan kebocoran pada detik ke-6000. Terdapat perbedaan respon antara sistem pengendali dengan AFTC dan sistem pengendali tanpa AFTC. Pada sistem pengendali tanpa AFTC *maximum undershoot* yang dihasilkan lebih besar yaitu 1,57% dengan *settling time* sebesar 8913 detik. Sedangkan pada sistem dengan AFTC *maximum undershoot* yang dihasilkan hanya sebesar 0,03% dengan *settling time* 6673 detik. Hal ini menunjukkan sistem tanpa AFTC memerlukan waktu lebih lama untuk kembali pada kondisi *steady* ketika terdapat kesalahan kebocoran daripada sistem dengan AFTC.

Hasil respon sistem pengendalian temperatur dengan kesalahan kebocoran 50% ditunjukkan pada gambar 4.11 berikut.



Gambar 4.11 Respon Sistem dengan Kesalahan Kebocoran 50%



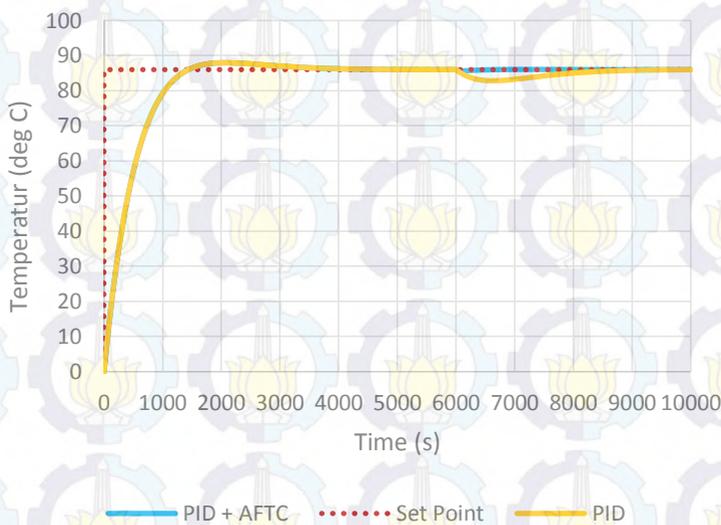
Gambar 4.12 Perbesaran Respon Sistem dengan Kesalahan Kebocoran 50%

Tabel 4.8 Parameter Performansi dengan Kesalahan Kebocoran 50%

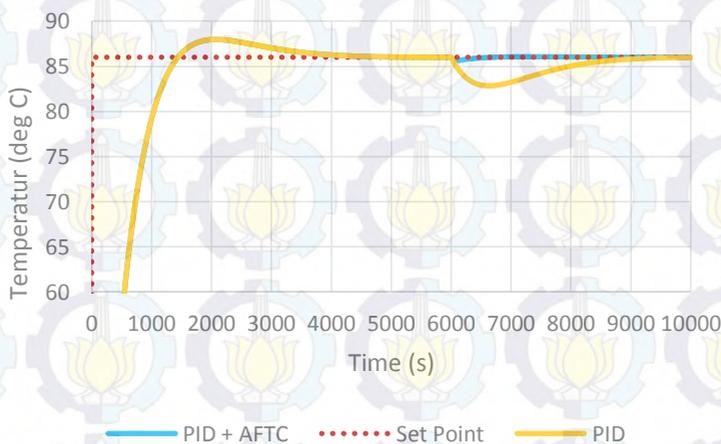
No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,05%
2	<i>Maximum Undershoot</i>	2,62%	0,31%
3	<i>Error Steady State</i>	0,01%	0,001%
4	<i>Settling Time</i>	9187 detik	6667 detik

Berdasarkan hasil dari perbesaran pada Gambar 4.12 dan data pada Tabel 4.8. diketahui bahwa sistem mengalami osilasi kembali ketika diberikan kesalahan kebocoran pada detik ke-6000. Terdapat perbedaan respon antara sistem pengendali dengan AFTC dan sistem pengendali tanpa AFTC. Pada sistem pengendali tanpa AFTC *maximum undershoot* yang dihasilkan lebih besar yaitu 2,62% dengan *settling time* sebesar 9187 detik. Sedangkan pada sistem dengan AFTC *maximum undershoot* yang dihasilkan hanya sebesar 0,31% dengan *settling time* 6667 detik. Hal ini menunjukkan sistem tanpa AFTC memerlukan waktu lebih lama untuk kembali pada kondisi *steady* ketika terdapat kesalahan kebocoran dari pada sistem dengan AFTC.

Hasil respon sistem pengendalian temperatur dengan kesalahan kebocoran 70% ditunjukkan pada gambar 4.13 berikut.



Gambar 4.13 Respon Sistem dengan Kesalahan Kebocoran 70%



Gambar 4.14 Perbesaran Respon Sistem dengan Kesalahan Kebocoran 70%

Tabel 4.9 Parameter Performansi dengan Kesalahan Kebocoran 70%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,08%
2	<i>Maximum Undershoot</i>	3,67%	0,45%
3	<i>Error Steady State</i>	1,12%	0,01%
4	<i>Settling Time</i>	9355 detik	6663 detik

Berdasarkan hasil dari perbesaran pada Gambar 4.14 dan data pada Tabel 4.9. diketahui bahwa sistem mengalami osilasi kembali ketika diberikan kesalahan kebocoran pada detik ke-4000. Terdapat perbedaan respon antara sistem pengendali dengan AFTC dan sistem pengendali tanpa AFTC. Pada sistem pengendali tanpa AFTC *maximum undershoot* yang dihasilkan lebih besar yaitu 3,67% dengan *settling time* sebesar 9355 detik. Sedangkan pada sistem dengan AFTC *maximum undershoot* yang dihasilkan hanya sebesar 0,45% dengan *settling time* 6663 detik. Hal ini menunjukkan sistem tanpa AFTC memerlukan waktu lebih lama untuk kembali pada kondisi *steady* ketika terdapat kesalahan kebocoran dari pada sistem dengan AFTC.

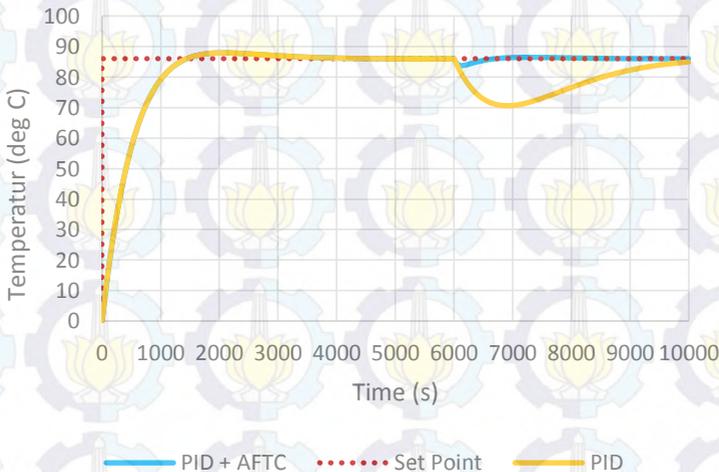
Dari selisih parameter performansi dapat diketahui bahwa sistem dengan algoritma AFTC dapat mengakomodasi kesalahan kebocoran yang diberikan pada detik ke 6000 untuk kembali menuju titik *setpoint*-nya, sistem tersebut menghasilkan parameter performansi yang lebih baik jika dibandingkan dengan sistem pengendalian temperatur dengan algoritma PID saja (tanpa AFTC). *Error steady state* dari kedua pengendali tersebut masih dalam batas normal, namun tetap lebih baik pengendali PID dengan tambahan algoritma AFTC. Tetapi jika dibandingkan dari *time settling* waktu yang dibutuhkan sistem untuk kembali ke *set point* jauh lebih cepat. Semakin kecil nilai parameter performansi maka keandalan dari sistem juga akan semakin baik. Jika nilai keandalan dari sistem baik maka masing-

masing komponen dari sistem akan memiliki *life time* yang lebih lama.

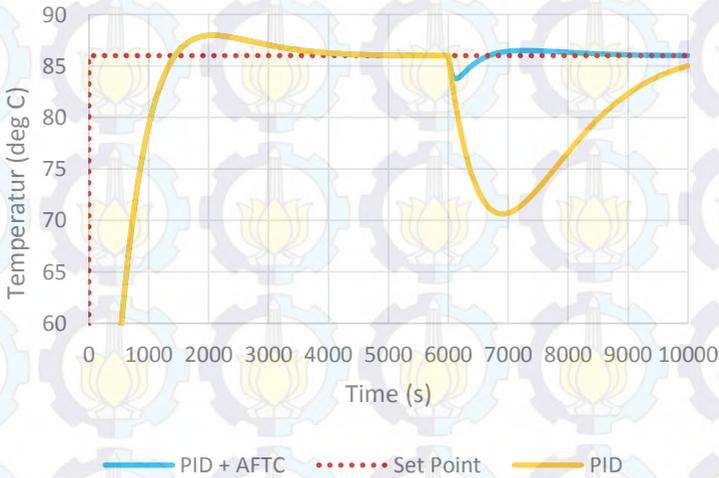
4.2.2 Uji *Lost of Effectiveness*

Kesalahan *lost of effectiveness* adalah kesalahan pada *control valve* karena bukaan valve yang terhambat oleh karat. Hal tersebut menyebabkan berkurangnya sinyal kontrol yang diberikan untuk *control valve*. Untuk menguji performansi dari sistem maka diberikan kesalahan berupa kesalahan *lost of effectiveness* sebesar 90%, 70% dan 50% dari sinyal kontrol yang masuk pada aktuator.

Hasil respon sistem pengendalian temperatur dengan kesalahan *lost of effectiveness* 50% ditunjukkan pada gambar 4.15 berikut.



Gambar 4.15 Respon Sistem dengan Kesalahan *Lost of Effectiveness* 50%



Gambar 4.16 Perbesaran Respon Sistem dengan Kesalahan *Lost of Effectiveness* 50%

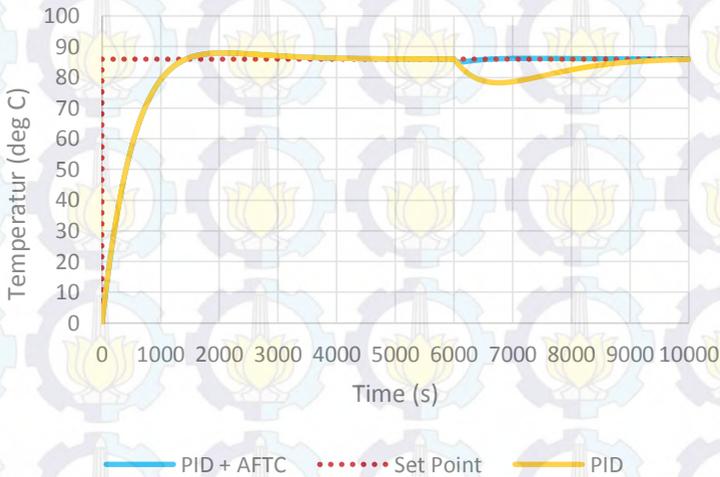
Tabel 4.10 Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 50%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,5%
2	<i>Maximum Undershoot</i>	17,87%	2,58%
3	<i>Error Steady State</i>	1,16%	0,01%
4	<i>Settling Time</i>	9995 detik	8903 detik

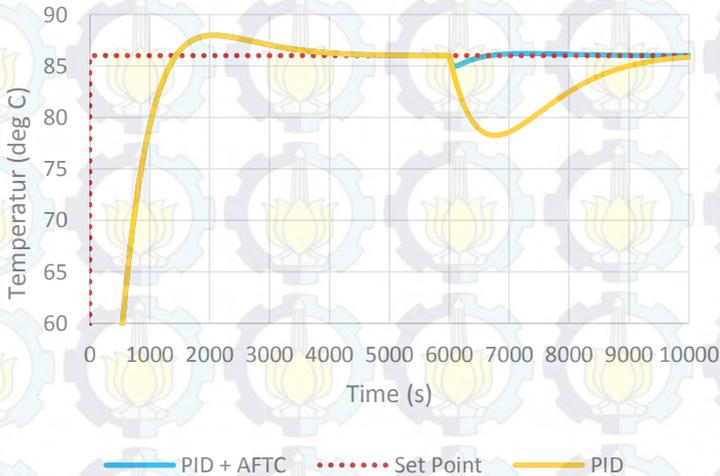
Pada Gambar 4.16 dan Tabel 4.10 menunjukkan respon sistem ketika diberikan kesalahan *lost of effectiveness* sebesar 50% pada detik ke-6000. Respon dibandingkan antara sistem dengan pengendali PID yang dilengkapi AFTC dan sistem dengan pengendali PID tanpa AFTC. Hasil menunjukkan respon dengan AFTC memiliki nilai *maximum undershoot* yang lebih kecil yaitu 2,58%. Selain itu, pengendali AFTC mampu mengolah informasi untuk meminimalisir *error* lebih cepat

dibandingkan dengan pengendali PID saja. Hal tersebut dapat dilihat dari *settling time* antara kedua sistem pengendalian.

Hasil respon sistem pengendalian temperatur dengan kesalahan *lost of effectiveness* 70% ditunjukkan pada gambar 4.17 berikut.



Gambar 4.17 Respon Sistem dengan Kesalahan *Lost of Effectiveness* 70%



Gambar 4.18 Perbesaran Respon Sistem dengan Kesalahan *Lost of Effectiveness* 70%

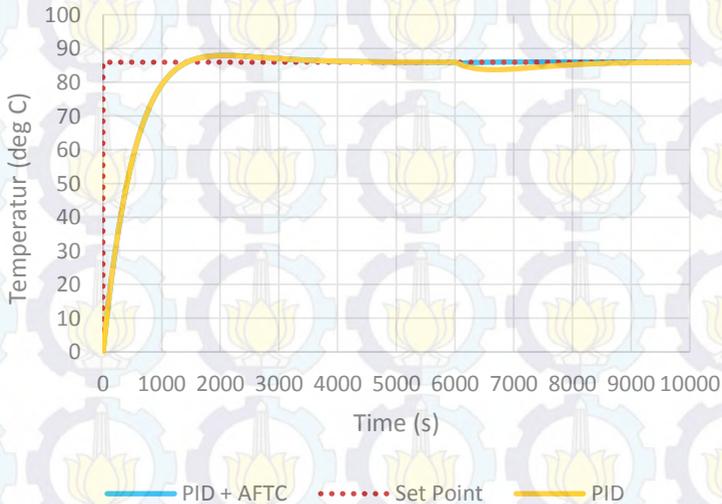
Tabel 4.11 Parameter Performansi dengan Kesalahan *Lost of Effectiveness* 70%

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,2%
2	<i>Maximum Undershoot</i>	9%	1,2%
3	<i>Error Steady State</i>	0,18%	0,007%
4	<i>Settling Time</i>	9881 detik	8277 detik

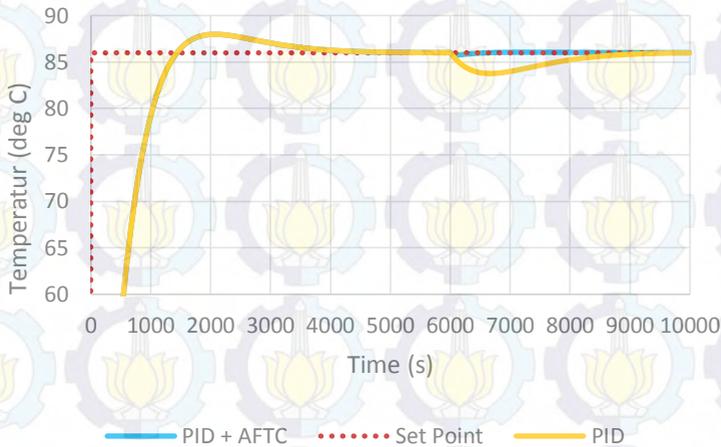
Pada Gambar 4.18 dan Tabel 4.11 menunjukkan respon sistem ketika diberikan kesalahan *lost of effectiveness* sebesar 70% pada detik ke-6000. Respon dibandingkan antara sistem dengan pengendali PID yang dilengkapi AFTC dan sistem dengan pengendali PID tanpa AFTC. Hasil menunjukkan respon dengan AFTC memiliki nilai *maximum undershoot* yang lebih kecil yaitu 1,2%. Selain itu, pengendali AFTC mampu mengolah

informasi untuk meminimalisir *error* lebih cepat dibandingkan dengan pengendali PID saja. Hal tersebut dapat dilihat dari *settling time* antara kedua sistem pengendalian.

Hasil respon sistem pengendalian temperatur dengan kesalahan *lost of effectiveness* 90% ditunjukkan pada gambar 4.19 berikut.



Gambar 4.19 Respon Sistem dengan Kesalahan *Lost of Effectiveness* 90%



Gambar 4.20 Perbesaran Respon Sistem dengan Kesalahan *Lost of Effectiveness 90%*

Tabel 4.12 Parameter Performansi dengan Kesalahan *Lost of Effectiveness 90%*

No	Parameter	PID tanpa AFTC	PID dengan AFTC
1	<i>Maximum Overshoot</i>	0%	0,05%
2	<i>Maximum Undershoot</i>	2,6%	0,3%
3	<i>Error Steady State</i>	0,02%	0,001%
4	<i>Settling Time</i>	9293 detik	6685 detik

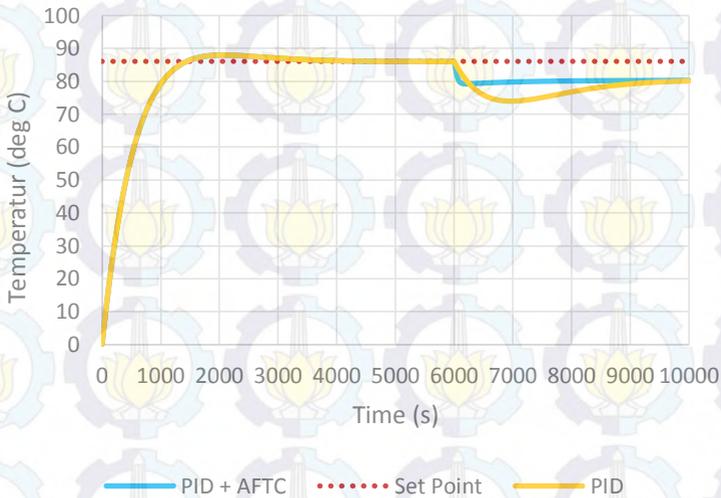
Pada Gambar 4.20 dan Tabel 4.12 menunjukkan respon sistem ketika diberikan kesalahan *lost of effectiveness* sebesar 90% pada detik ke-6000. Respon dibandingkan antara sistem dengan pengendali PID yang dilengkapi AFTC dan sistem dengan pengendali PID tanpa AFTC. Hasil menunjukkan respon dengan AFTC memiliki nilai *maximum undershoot* yang lebih kecil yaitu 0,3%. Selain itu, pengendali AFTC mampu mengolah informasi untuk meminimalisir *error* lebih cepat dibandingkan dengan

pengendali PID saja. Hal tersebut dapat dilihat dari *settling time* antara kedua sistem pengendalian.

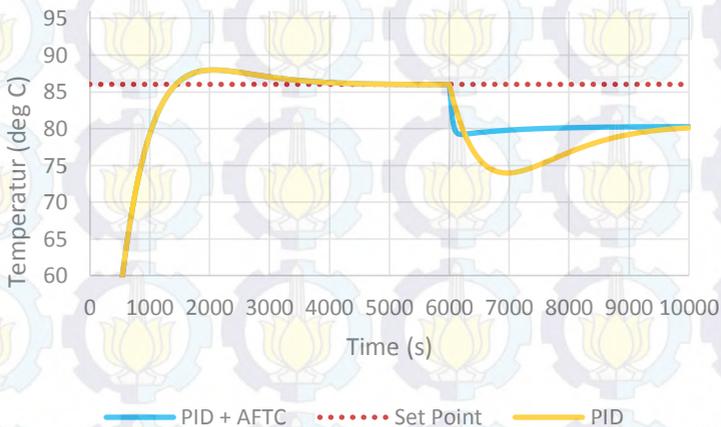
Secara keseluruhan, hasil respon sistem kontrol temperatur dengan algoritma *active fault tolerant control* menunjukkan bahwa walaupun terjadi kesalahan berupa aktuator dalam bekerja, temperatur tetap pada kondisi nominal atau *setpointnya*.. Kesalahan aksi kontrol terjadi akibat ketidaksempurnaan sinyal yang diterima oleh aktuator. *Maximum overshoot* terbesar terjadi pada uji kesalahan *lost of effectiveness* sebesar 50%. Hal ini dikarenakan terdapat sinyal yang hilang sebesar 50%. Namun pada kondisi tersebut sistem pengendali yang ditambahkan algoritma *AFTC* masih mampu mengatasinya dan *maximum overshoot* yang dihasilkan lebih kecil dibandingkan sistem dengan pengendali PID saja.

4.3 Uji kesalahan Sensor dan Aktuator Bersamaan

Untuk menguji performansi dari sistem pengendalian temperature dengan kesalahan pada sensor dan aktuator, maka uji kedua yang dilakukan adalah uji kesalahan sensor dan aktuator secara bersamaan. Kesalahan sensor dan aktuator diberikan secara bersamaan pada detik ke – 6000. Jenis kesalahan yang diberikan adalah acak baik untuk kesalahan sensor maupun juga kesalahan aktuator. Salah satunya ialah kesalahan bias +0,75% untuk sensor dan kesalahan *lost of effectiveness* sebesar 70% untuk aktuator.



Gambar 4.21 Hasil Uji Kesalahan Sensor dan Aktuator secara Bersamaan



Gambar 4.22 Perbesaran Hasil Uji Kesalahan Sensor dan Aktuator secara Bersamaan

Namun hasil untuk uji secara bersamaan tidak berhasil, dikarenakan saat rekonfigurasi dari kesalahan sensor belum selesai, *observer* aktuator sudah bekerja. Hal tersebut mengakibatkan sistem menjadi tidak stabil. Sehingga *observer* tidak bekerja sebagaimana mestinya.





Halaman ini sengaja dikosongkan

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan beberapa kesimpulan dari tugas akhir mengenai perancangan *Active Fault Tolerant Control* (AFTC) pada *Gasoline Splitter Column* di PT. Pertamina (Persero) RU IV Cilacap dengan kesalahan pada sensor dan aktuator adalah :

1. Pada algoritma AFTC telah dirancang *observer* untuk dapat mengestimasi kesalahan dan dirancang rekonfigurasi sinyal kontrol untuk mengakomodasi kesalahan yang terjadi.
2. Sistem dengan algoritma AFTC dapat mengakomodasi atau mengatasi kesalahan sensor seperti kesalahan bias sebesar +0,75% dan -0,75%, sensitivitas sebesar 90%, 80% dan 70% dan kesalahan berupa *noise*. Kedua uji tersebut menghasilkan performansi yang lebih baik dibandingkan dengan sistem pengendali PID tanpa AFTC. Hal ini dapat dilihat dari beberapa parameter seperti *maximum overshoot*, *error steady state* dan *settling time*. Salah satu buktinya terlihat pada hasil uji bias -0,75%, untuk PID tanpa AFTC sistem tersebut menghasilkan *maximum overshoot* yang lebih besar yaitu 6,91% dan *settling time* yang lebih lama yaitu 9579 detik. Sedangkan pada sistem PID dengan AFTC *maximum overshoot* hanya sebesar 0,06% dengan *time settling* 7109 detik.
3. Sistem dengan algoritma AFTC dapat mengakomodasi atau mengatasi kesalahan aktuator seperti kebocoran sebesar 30%, 50% dan 70% serta *lost of effectiveness* sebesar 90%, 70% dan 50%. Kedua uji tersebut juga menghasilkan performansi yang lebih baik dibandingkan dengan sistem pengendali PID tanpa AFTC. Hal ini dapat dilihat dari beberapa parameter seperti *maximum undershoot*, *error steady state* dan *settling time*. Salah satu buktinya terlihat pada hasil uji kebocoran 50%. Pada sistem pengendali tanpa AFTC *maximum undershoot* yang dihasilkan lebih besar

yaitu 2,62% dengan *settling time* sebesar 9187 detik. Sedangkan pada sistem dengan AFTC *maximum undershoot* yang dihasilkan hanya sebesar 0,31% dengan *settling time* 6667 detik.

4. Ketika kesalahan sensor dan aktuator diberikan pada waktu yang bersamaan, AFTC tidak mampu mengakomodasi kedua kesalahan tersebut. Hal ini dikarenakan saat rekonfigurasi dari kesalahan sensor belum selesai, *observer* aktuator sudah bekerja.

5.2 Saran

Dalam proses tugas akhir ini terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya, yakni pengembangan algoritma AFTC dapat dilakukan secara *real time* pada *miniplant* dengan berbagai macam strategi kontrol.

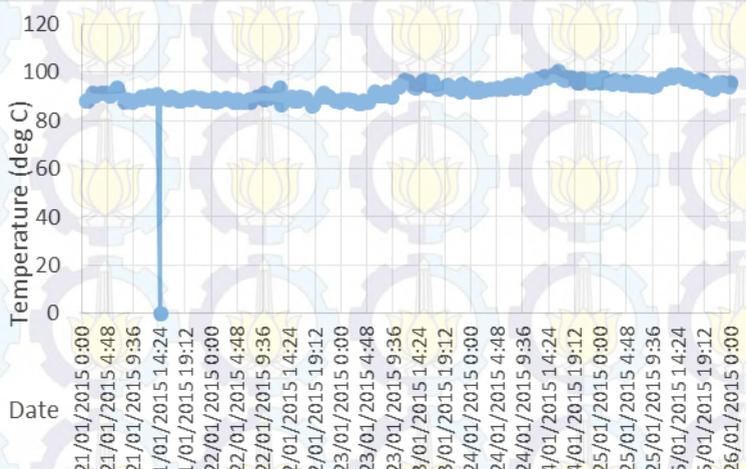
DAFTAR PUSTAKA

- [1] PT. Pertamina (Persero) RU IV Cilacap. “Sekilas Pengenalan Kilang (*Refinery Unit*) RU IV Cilacap”. Cilacap, Jawa Tengah. 7 Mei 2012.
- [2] K. Ogata, *Modern Control Engineering* – 4th ed., Prentice Hall, 2006.
- [3] Math Pro Inc. “*An Introduction to Petroleum Refining and The Production of Ultra Low Sulfur Gasoline and Diesel Fuel*”. Bethesda, Maryland. 24 Oktober 2011.
- [4] 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.
- [5] S. Rici Adi. “Pengendalian Proses Variabel Jamak Kolom Destilasi Menggunakan *Model Predictive Control* pada UNISIM R 390.1” Teknik Kimia : Universitas Indonesia. 2012.
- [6] K. Leily Nurul. A.F. Ramdja, L. Nicky. “Tinjauan Teoritis Perancangan Kolom Distilasi untuk Pra-Rencana Pabrik Skala Industri”. Teknik Kimia, Fakultas Teknik, Universitas Sriwijaya. 16 Desember 2009.
- [7] HMI FOC I PT.Pertamina (Persero) RU IV Cilacap
- [8] Data Proses FOC I PT.Pertamina (Persero) RU IV Cilacap
- [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] Smith, C.A. “*Principles and Practice of Automatic Process Control – Second Edition*”. 1997.
- [11] F. Gunterus, Falsafah Dasar: Sistem Pengendalian Proses, Elex Media Komputindo Jakarta, 1997.
- [12] Youmin Zhang, Jin Jiang, “Bibliographical review on reconfigurable fault-tolerant control system,” *Annual Reviews in Control*, vol. 32, issue 2, pp. 229-252, December 2008.

- [13] 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 December 2013.
- [14] H. Noura, D. Theilliol, J.C. Ponsart, A. Chamseddine, *Fault-tolerance Control Systems: Design and Practical Applications*, Springer Verlag London, 2009.
- [15] Z. Gao, H. Wang, "Descriptor observer approaches for multivariable systems with measurement noises and application in fault detection and diagnosis," *Systems Control Letters*, vol. 55, pp. 304-313, 2006.
- [16] S. Boyd, L. El Ghaoui, Eric Feron, V. Balakhirhnan. "*Linear Matrix Inequalities in System and Control Theory*". Siam. 1994. Pp. 7 – 8.
- [17] PT. Pertamina (Persero) RU IV Cilacap. "*Data/Requisition Sheet for Pressure Vessel (Columns, Reactor, Accumulators, etc) – Equipment No. 11C8*".
- [18] PT. Pertamina (Persero) RU IV Cilacap. "*Fluor Daniel Engineers & Constructors, LTD. "Process Flow Scheme Fractionating Section, (DN 429400 – 4 – 11 – 004)"*". Cilacap, Jawa Tengah. Desember, 2000.

LAMPIRAN C

C.1 Trend Temperatur *Gasoline Splitter Colum* pada Tanggal (21 Januari –22 Januari 2015) di PT. Pertamina (persero) RU IV Cilacap



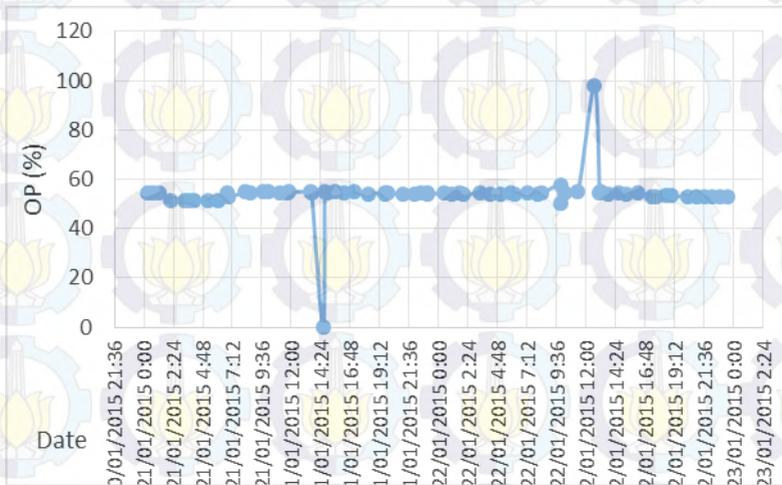
C2. Data Pengukuran Temperatur *Gasoline Splitter Colum* pada Tanggal (21 Januari – 22 Januari 2015) di PT. Pertamina (persero) RU IV Cilacap

Root.FOC1.11TIC_009.PV:Value	21/01/2015 0:53	88,22805
Root.FOC1.11TIC_009.PV:Value	21/01/2015 1:08	88,21548
Root.FOC1.11TIC_009.PV:Value	21/01/2015 2:06	91,2508
Root.FOC1.11TIC_009.PV:Value	21/01/2015 2:34	90,13362
Root.FOC1.11TIC_009.PV:Value	21/01/2015 3:42	91,15057
Root.FOC1.11TIC_009.PV:Value	21/01/2015 4:47	91,34023
Root.FOC1.11TIC_009.PV:Value	21/01/2015 5:14	90,43417
Root.FOC1.11TIC_009.PV:Value	21/01/2015 5:27	90,53505
Root.FOC1.11TIC_009.PV:Value	21/01/2015 6:43	93,29541

Root.FOC1.11TIC_009.PV:Value	21/01/2015 6:49	91,88956
Root.FOC1.11TIC_009.PV:Value	21/01/2015 7:57	87,5285
Root.FOC1.11TIC_009.PV:Value	21/01/2015 8:37	88,8976
Root.FOC1.11TIC_009.PV:Value	21/01/2015 9:35	87,66155
Root.FOC1.11TIC_009.PV:Value	21/01/2015 9:50	88,04781
Root.FOC1.11TIC_009.PV:Value	21/01/2015 11:03	89,78864
Root.FOC1.11TIC_009.PV:Value	21/01/2015 11:12	88,93221
Root.FOC1.11TIC_009.PV:Value	21/01/2015 12:28	90,31055
Root.FOC1.11TIC_009.PV:Value	21/01/2015 12:59	89,39354
Root.FOC1.11TIC_009.PV:Value	21/01/2015 13:57	90,60573
Root.FOC1.11TIC_009.PV:Value	21/01/2015 14:02	90,0461
Root.FOC1.11TIC_009.PV:Value	21/01/2015 14:35	0
Root.FOC1.11TIC_009.PV:Value	21/01/2015 14:40	89,031
Root.FOC1.11TIC_009.PV:Value	21/01/2015 16:02	88,85823
Root.FOC1.11TIC_009.PV:Value	21/01/2015 16:42	89,77362
Root.FOC1.11TIC_009.PV:Value	21/01/2015 17:01	89,1767
Root.FOC1.11TIC_009.PV:Value	21/01/2015 18:00	88,01224
Root.FOC1.11TIC_009.PV:Value	21/01/2015 18:26	88,36553
Root.FOC1.11TIC_009.PV:Value	21/01/2015 19:44	89,37934
Root.FOC1.11TIC_009.PV:Value	21/01/2015 20:12	88,82383
Root.FOC1.11TIC_009.PV:Value	21/01/2015 20:38	89,6426
Root.FOC1.11TIC_009.PV:Value	21/01/2015 22:01	89,43938
Root.FOC1.11TIC_009.PV:Value	21/01/2015 22:40	88,35455
Root.FOC1.11TIC_009.PV:Value	21/01/2015 22:49	88,76846
Root.FOC1.11TIC_009.PV:Value	21/01/2015 23:34	87,97037
Root.FOC1.11TIC_009.PV:Value	22/01/2015 0:45	89,20009
Root.FOC1.11TIC_009.PV:Value	22/01/2015 0:55	87,88847
Root.FOC1.11TIC_009.PV:Value	22/01/2015 1:41	87,92545
Root.FOC1.11TIC_009.PV:Value	22/01/2015 2:43	89,1872
Root.FOC1.11TIC_009.PV:Value	22/01/2015 3:17	88,96674

Root.FOC1.11TIC_009.PV:Value	22/01/2015 4:15	87,7744
Root.FOC1.11TIC_009.PV:Value	22/01/2015 4:50	87,89549
Root.FOC1.11TIC_009.PV:Value	22/01/2015 4:58	88,97366
Root.FOC1.11TIC_009.PV:Value	22/01/2015 6:11	87,48887
Root.FOC1.11TIC_009.PV:Value	22/01/2015 7:09	88,57339
Root.FOC1.11TIC_009.PV:Value	22/01/2015 7:21	87,86389
Root.FOC1.11TIC_009.PV:Value	22/01/2015 8:35	89,95949
Root.FOC1.11TIC_009.PV:Value	22/01/2015 9:51	91,17665
Root.FOC1.11TIC_009.PV:Value	22/01/2015 9:55	88,84166
Root.FOC1.11TIC_009.PV:Value	22/01/2015 10:44	89,26199
Root.FOC1.11TIC_009.PV:Value	22/01/2015 11:23	90,36897
Root.FOC1.11TIC_009.PV:Value	22/01/2015 12:41	93,70157
Root.FOC1.11TIC_009.PV:Value	22/01/2015 12:57	86,38644
Root.FOC1.11TIC_009.PV:Value	22/01/2015 13:25	89,32194
Root.FOC1.11TIC_009.PV:Value	22/01/2015 13:49	88,0525
Root.FOC1.11TIC_009.PV:Value	22/01/2015 14:53	89,54921
Root.FOC1.11TIC_009.PV:Value	22/01/2015 15:16	88,38702
Root.FOC1.11TIC_009.PV:Value	22/01/2015 16:06	88,44536
Root.FOC1.11TIC_009.PV:Value	22/01/2015 16:34	89,68096
Root.FOC1.11TIC_009.PV:Value	22/01/2015 17:25	89,48888
Root.FOC1.11TIC_009.PV:Value	22/01/2015 18:38	85,91183
Root.FOC1.11TIC_009.PV:Value	22/01/2015 18:53	86,76847
Root.FOC1.11TIC_009.PV:Value	22/01/2015 20:17	90,0418
Root.FOC1.11TIC_009.PV:Value	22/01/2015 20:54	91,14367
Root.FOC1.11TIC_009.PV:Value	22/01/2015 21:43	89,55842
Root.FOC1.11TIC_009.PV:Value	22/01/2015 21:52	89,81952
Root.FOC1.11TIC_009.PV:Value	22/01/2015 22:54	88,25969

C.3 Trend Buka-an Valve *Gasoline Splitter Colum* pada Tanggal (21 Januari –22 Januari 2015) di PT. Pertamina (persero) RU IV Cilacap



C4. Data Variasi Buka-an Valve *Gasoline Splitter Colum* pada Tanggal (21 Januari – 22 Januari 2015) di PT. Pertamina (persero) RU IV Cilacap

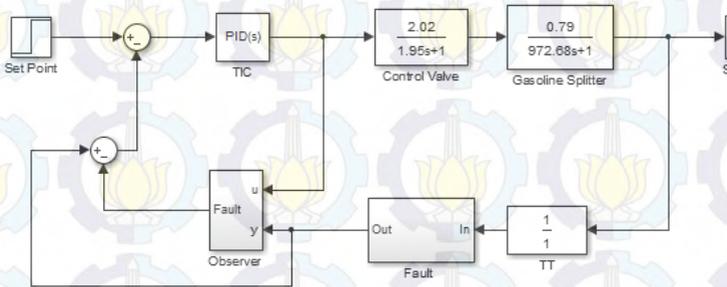
Root.FOC1.11FIC_038.OP:Value	21/01/2015 0:17	54,74394
Root.FOC1.11FIC_038.OP:Value	21/01/2015 0:42	54,48822
Root.FOC1.11FIC_038.OP:Value	21/01/2015 1:17	54,72377
Root.FOC1.11FIC_038.OP:Value	21/01/2015 2:12	51,19704
Root.FOC1.11FIC_038.OP:Value	21/01/2015 3:22	51,45131
Root.FOC1.11FIC_038.OP:Value	21/01/2015 3:40	51,20845
Root.FOC1.11FIC_038.OP:Value	21/01/2015 4:04	51,2001
Root.FOC1.11FIC_038.OP:Value	21/01/2015 5:11	51,4772
Root.FOC1.11FIC_038.OP:Value	21/01/2015 6:02	51,18886
Root.FOC1.11FIC_038.OP:Value	21/01/2015 6:48	54,50635
Root.FOC1.11FIC_038.OP:Value	21/01/2015 6:55	52,81609

Root.FOC1.11FIC_038.OP:Value	21/01/2015 8:16	54,92417
Root.FOC1.11FIC_038.OP:Value	21/01/2015 8:40	54,66332
Root.FOC1.11FIC_038.OP:Value	21/01/2015 9:40	54,93273
Root.FOC1.11FIC_038.OP:Value	21/01/2015 10:09	54,9737
Root.FOC1.11FIC_038.OP:Value	21/01/2015 11:05	54,61997
Root.FOC1.11FIC_038.OP:Value	21/01/2015 11:43	54,61617
Root.FOC1.11FIC_038.OP:Value	21/01/2015 11:48	54,91481
Root.FOC1.11FIC_038.OP:Value	21/01/2015 13:34	54,934
Root.FOC1.11FIC_038.OP:Value	21/01/2015 13:39	54,64294
Root.FOC1.11FIC_038.OP:Value	21/01/2015 14:35	0
Root.FOC1.11FIC_038.OP:Value	21/01/2015 14:44	54,96296
Root.FOC1.11FIC_038.OP:Value	21/01/2015 14:56	54,68578
Root.FOC1.11FIC_038.OP:Value	21/01/2015 15:30	54,91494
Root.FOC1.11FIC_038.OP:Value	21/01/2015 16:15	54,74026
Root.FOC1.11FIC_038.OP:Value	21/01/2015 17:08	54,88595
Root.FOC1.11FIC_038.OP:Value	21/01/2015 18:17	54,10463
Root.FOC1.11FIC_038.OP:Value	21/01/2015 19:39	54,31355
Root.FOC1.11FIC_038.OP:Value	21/01/2015 19:40	54,06134
Root.FOC1.11FIC_038.OP:Value	21/01/2015 19:53	54,29602
Root.FOC1.11FIC_038.OP:Value	21/01/2015 21:06	54,0004
Root.FOC1.11FIC_038.OP:Value	21/01/2015 22:04	53,99063
Root.FOC1.11FIC_038.OP:Value	21/01/2015 22:34	54,26969
Root.FOC1.11FIC_038.OP:Value	21/01/2015 23:04	54,29197
Root.FOC1.11FIC_038.OP:Value	21/01/2015 23:05	54,04893
Root.FOC1.11FIC_038.OP:Value	22/01/2015 0:28	54,30437
Root.FOC1.11FIC_038.OP:Value	22/01/2015 1:05	53,99929
Root.FOC1.11FIC_038.OP:Value	22/01/2015 1:39	54,28545
Root.FOC1.11FIC_038.OP:Value	22/01/2015 2:02	53,96952
Root.FOC1.11FIC_038.OP:Value	22/01/2015 3:20	54,26513
Root.FOC1.11FIC_038.OP:Value	22/01/2015 4:09	53,92916

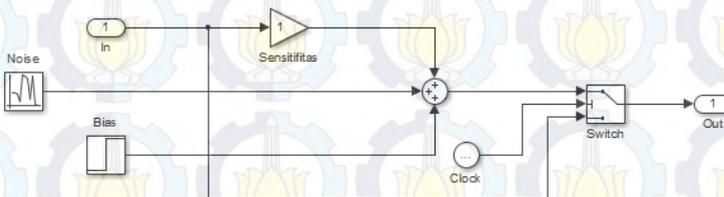
Root.FOC1.11FIC_038.OP:Value	22/01/2015 5:02	53,97335
Root.FOC1.11FIC_038.OP:Value	22/01/2015 5:53	54,27967
Root.FOC1.11FIC_038.OP:Value	22/01/2015 6:13	54,03464
Root.FOC1.11FIC_038.OP:Value	22/01/2015 7:10	54,33089
Root.FOC1.11FIC_038.OP:Value	22/01/2015 8:04	53,97502
Root.FOC1.11FIC_038.OP:Value	22/01/2015 8:23	54,29496
Root.FOC1.11FIC_038.OP:Value	22/01/2015 9:54	57,52142
Root.FOC1.11FIC_038.OP:Value	22/01/2015 9:55	50,12683
Root.FOC1.11FIC_038.OP:Value	22/01/2015 10:17	54,63745
Root.FOC1.11FIC_038.OP:Value	22/01/2015 11:19	54,98713
Root.FOC1.11FIC_038.OP:Value	22/01/2015 12:40	98,08468
Root.FOC1.11FIC_038.OP:Value	22/01/2015 13:05	54,40647
Root.FOC1.11FIC_038.OP:Value	22/01/2015 13:07	54,80239
Root.FOC1.11FIC_038.OP:Value	22/01/2015 13:48	53,99164
Root.FOC1.11FIC_038.OP:Value	22/01/2015 14:37	54,34721
Root.FOC1.11FIC_038.OP:Value	22/01/2015 15:17	54,08393
Root.FOC1.11FIC_038.OP:Value	22/01/2015 16:14	54,26894
Root.FOC1.11FIC_038.OP:Value	22/01/2015 17:21	53,07487
Root.FOC1.11FIC_038.OP:Value	22/01/2015 17:38	52,83453
Root.FOC1.11FIC_038.OP:Value	22/01/2015 18:29	53,39249
Root.FOC1.11FIC_038.OP:Value	22/01/2015 18:56	53,28059
Root.FOC1.11FIC_038.OP:Value	22/01/2015 20:15	52,89998
Root.FOC1.11FIC_038.OP:Value	22/01/2015 21:00	52,77533
Root.FOC1.11FIC_038.OP:Value	22/01/2015 21:36	53,05099
Root.FOC1.11FIC_038.OP:Value	22/01/2015 22:15	52,86724
Root.FOC1.11FIC_038.OP:Value	22/01/2015 22:50	53,08714
Root.FOC1.11FIC_038.OP:Value	22/01/2015 23:27	53,13828

LAMPIRAN D HASIL PERANCANGAN

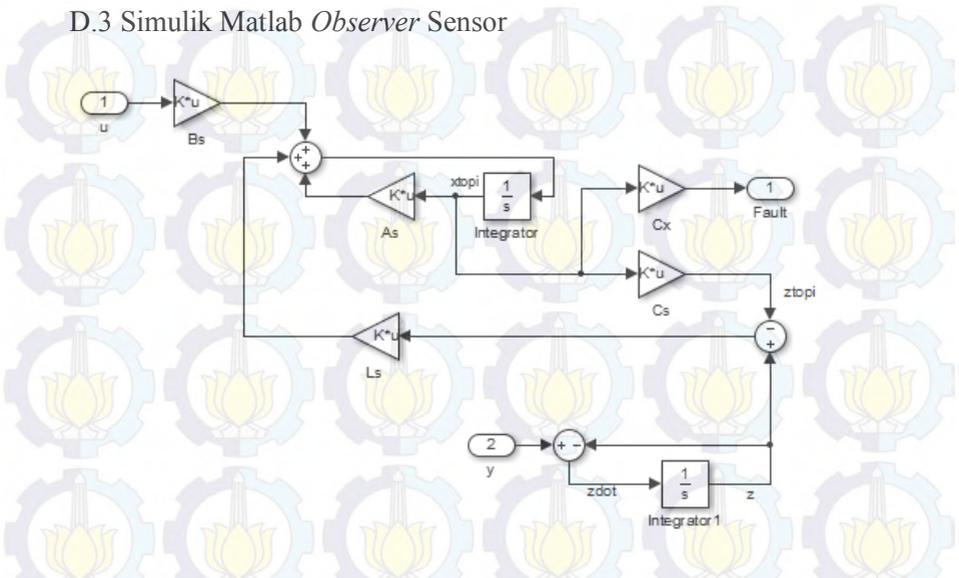
D.1 Simulink Matlab *General* dengan Kesalahan pada Sensor



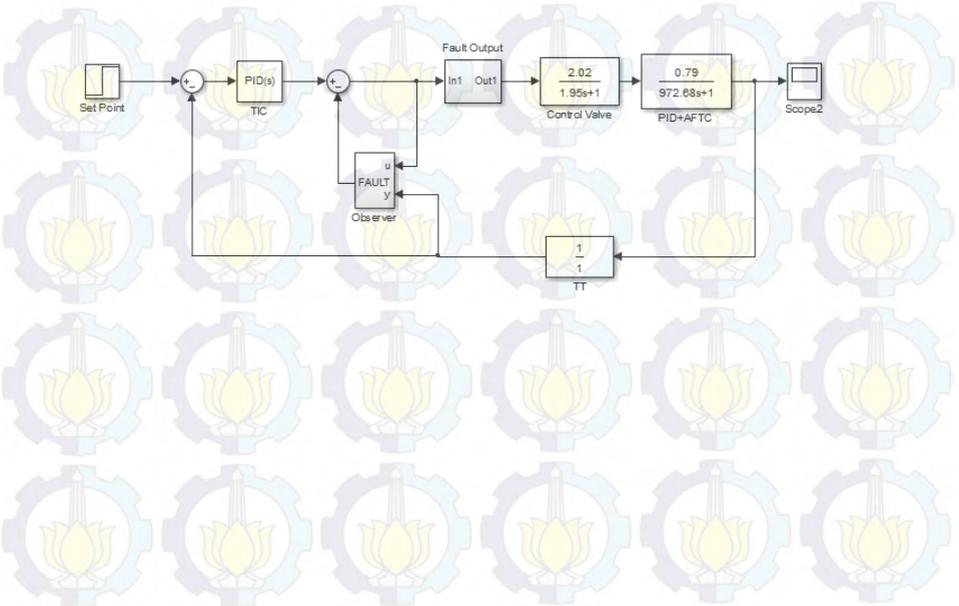
D.2 Simulink Matlab *Fault Output of Sensor*



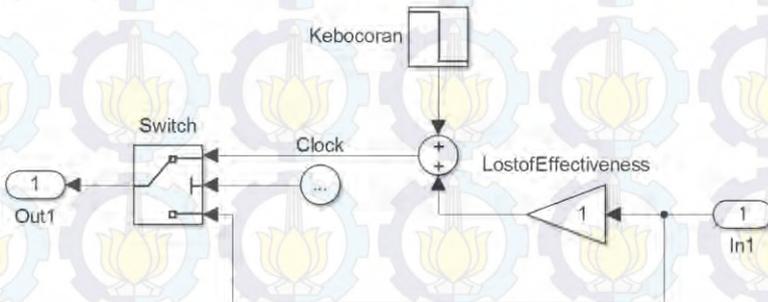
D.3 Simulink Matlab *Observer Sensor*



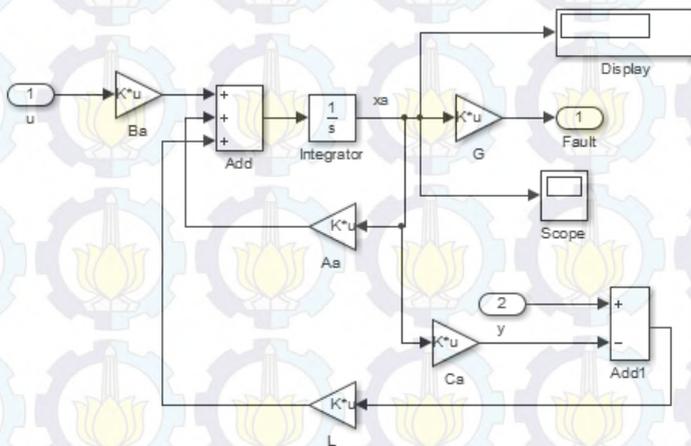
D.4 Simulink Matlab *General* dengan Kesalahan pada Aktuator



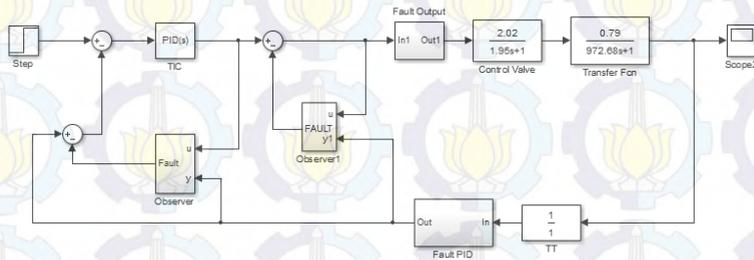
D.5 Simulink Matlab *Fault Output of Actuator*



D.6 Simulink Matlab *Observer* Aktuator



D.7 Simulink Matlab *General* dengan Kesalahan pada Sensor dan Aktuator



BIODATA PENULIS



Arina Vidya Abshari merupakan nama lengkap penulis dengan nama panggilannya, Arin. Penulis dilahirkan di kota Cilacap, Jawa Tengah pada tanggal 2 November 1993 sebagai anak pertama dari tiga bersaudara pasangan Noer Siswadi dan Ninik Ratna Makmurliana. Riwayat pendidikan penulis adalah SD Patra Mandiri, Cilacap tahun (2000–2006), SMP Negeri I Cilacap tahun (2006–2009), SMA Negeri I Cilacap tahun (2009–2011). 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: vidyaa211@gmail.com.