



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**SIMULASI PENGENDALIAN LEVEL AIR DI DALAM *STEAM DRUM* PADA SISTEM *BOILER* PLTU MENGGUNAKAN SKEMA *ROBUST SELF-TUNING* UNTUK LOGIKA *FUZZY* TIPE *PID***

Mughny Indra Darmawan  
NRP 0711154000131

Dosen Pembimbing  
Ir. Ali Fatoni, MT.

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



**TUGAS AKHIR - EE 184801**

**SIMULASI PENGENDALIAN LEVEL AIR DI DALAM  
*STEAM DRUM* PADA SISTEM *BOILER* PLTU  
MENGUNAKAN SKEMA *ROBUST SELF-TUNING*  
UNTUK LOGIKA *FUZZY* TIPE PID**

Mughny Indra Darmawan  
NRP 0711154000131

Dosen Pembimbing  
Ir. Ali Fatoni, MT.

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



**FINAL PROJECT - EE 184801**

***SIMULATION WATER LEVEL CONTROL IN THE  
STEAM DRUM OF POWERPLANT'S BOILER SYSTEM  
USING A ROBUST SELF-TUNING SCHEME FOR PID-  
TYPE FUZZY***

Mughny Indra Darmawan  
NRP 07111540000131

*Supervisor*  
Ir. Ali Fatoni, MT.

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



**PERNYATAAN KEASLIAN  
TUGAS AKHIR**

## PERNYATAAN KEASLIAN TUGAS AKHIR

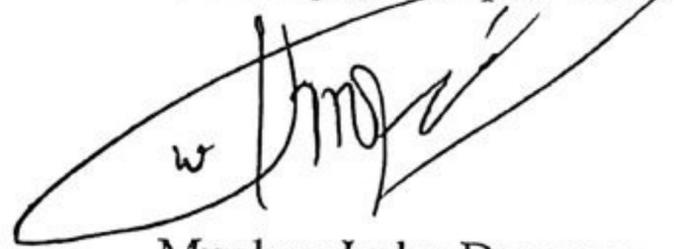
Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

**“Simulasi Pengendalian Level Air di dalam *Steam Drum* pada Sistem *Boiler* PLTU menggunakan Skema *Robust Self-Tuning* untuk Logika *Fuzzy Tipe PID*”**

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

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

Surabaya, 12 April 2019



Mughny Indra Darmawan  
NRP. 071115400001

*Halaman ini sengaja dikosongkan*



**LEMBAR PENGESAHAN**

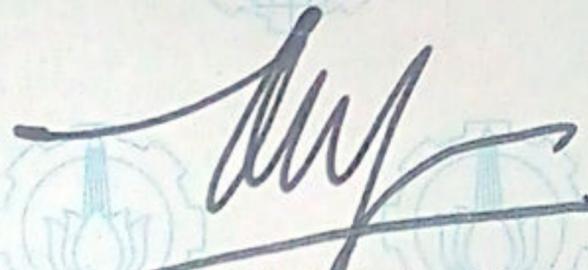
**SIMULASI PENGENDALIAN LEVEL AIR DI  
DALAM *STEAM DRUM* PADA SISTEM *BOILER*  
PLTU MENGGUNAKAN SKEMA *ROBUST SELF-  
TUNING* UNTUK LOGIKA *FUZZY* TIPE *PID***

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Memperoleh Gelar Sarjana Teknik Elektro  
Pada  
Bidang Studi Teknik Sistem Pengaturan  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember

Menyetujui :

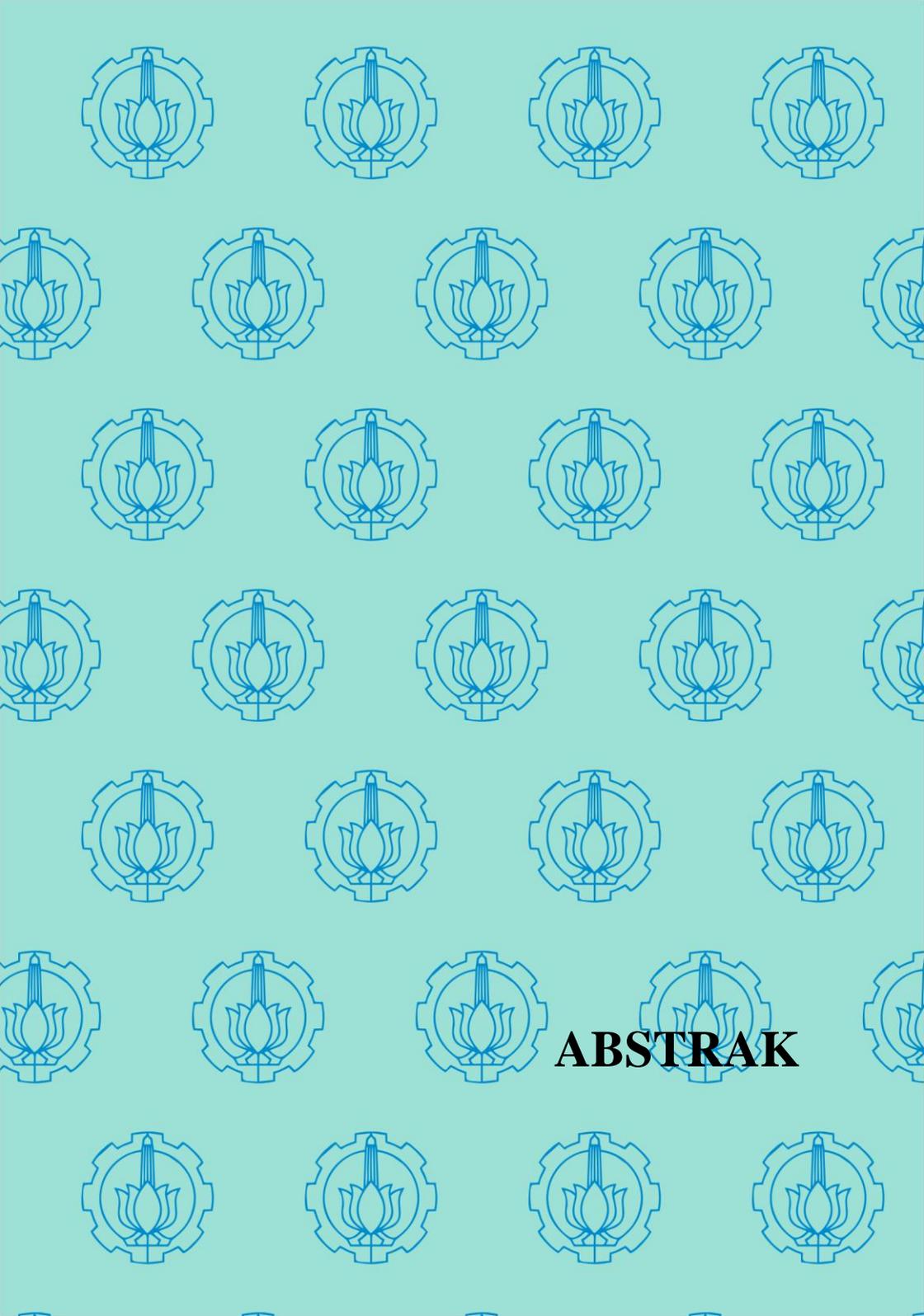
Dosen Pembimbing



Ir. Ali Fatoni, MT.  
NIP. 196206031989031002



*Halaman ini sengaja dikosongkan*



# ABSTRAK

# **SIMULASI PENGENDALIAN LEVEL AIR DI DALAM *STEAM DRUM* PADA SISTEM *BOILER* PLTU MENGGUNAKAN SKEMA *ROBUST SELF- TUNING* UNTUK LOGIKA *FUZZY* TIPE *PID***

Mughny Indra Darmawan  
07111540000131

Dosen Pembimbing : Ir. Ali Faton, MT.  
NIP : 196206031989031002

## **ABSTRAK**

Level air didalam *steam drum* perlu dijaga konstan pada titik tertentu. Oleh karena itu, diperlukan pengendalian untuk laju aliran air *feedwater* yang masuk agar menyesuaikan terhadap gangguan laju aliran uap yang keluar. Besar kecilnya laju aliran uap yang keluar bergantung pada permintaan beban. Jika permintaan beban tinggi dan fluktuasi, maka untuk memperahankan level air di dalam *steam drum* akan semakin sulit. Kontroler logika *fuzzy* tipe *PID* dengan skema *robust self-tuning* akan diimplementasikan pada sistem pengaturan level air dalam *steam drum*. Pada skema *three-element control*, ketika diberikan beban tinggi 700 MW sistem tersebut menghasilkan simpangan eror sebesar 15,69 mm *peak* terhadap *set point*-nya. Nilai tersebut lebih kecil dari pada skema *single-element control* dengan menghasilkan simpangan eror sebesar 18,5 mm terhadap *set point*-nya. Namun, ketika diberikan perubahan *set point* sebesar 40 mm skema *three-element control* menghasilkan respons 16,82 mm *peak* simpangan eror terhadap *set point*-nya. Nilai tersebut jauh lebih besar dari pada respons sistem dengan skema *single-element control* yang hanya menghasilkan simpangan eror 3,91 mm *peak* terhadap *set point*-nya.

**Kata Kunci :** *Fuzzy PID, skema robust self-tuning, steam drum.*

*Halaman ini sengaja dikosongkan*



***ABSTRACT***

***SIMULATION WATER LEVEL CONTROL IN THE  
STEAM DRUM OF POWERPLANT'S BOILER SYSTEM  
USING A ROBUST SELF-TUNING SCHEME FOR PID-  
TYPE FUZZY***

Mughny Indra Darmawan  
07111540000131

*Supervisor* : Ir. Ali Fatoni, MT.  
*ID Number* : 196206031989031002

***ABSTRACT***

*The water level in the steam drum needs to be kept constant at a certain point. Therefore, control is needed for the flow rate of incoming water feedwater to adjust to the disruption of the steam flow rate that comes out. The size of the steam flow rate that comes out depends on the load demand. If the demand for loads is high and fluctuations, then the water level in the steam drum will be more difficult. The PID type fuzzy logic controller with a robust self-tuning scheme will be implemented in the water level regulation system in the steam drum. In the three-element control scheme, when given a high load of 700 MW the system produces an error deviation of 15.69 mm peak against the set point. This value is smaller than the single-element control scheme by producing an error deviation of 18.5 mm against the set point. However, when given a set point change of 40 mm the three-element control scheme produced a response of 16.82 mm peak error error to the set point. This value is far greater than the system response with the single-element control scheme which only produces an error deviation of 3.91 mm peak against the set point.*

***Keywords*** : *Fuzzy PID, robust self-tuning scheme, steam drum.*

*Halaman ini sengaja dikosongkan*



# **KATA PENGANTAR**

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan karunianya, sehingga penulis dapat membuat dan menyelesaikan laporan tugas akhir ini dengan semestinya serta tepat waktu.

Kegiatan tugas akhir ini termasuk salah satu mata kuliah yang wajib ditempuh di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember ini. Penelitian yang penulis lakukan mengambil topik pengaturan level air di dalam *steam drum*. Laporan tugas akhir ini disusun untuk melengkapi dan mendokumentasikan atau membukukan hasil capaian dari penelitian yang telah dilaksanakan.

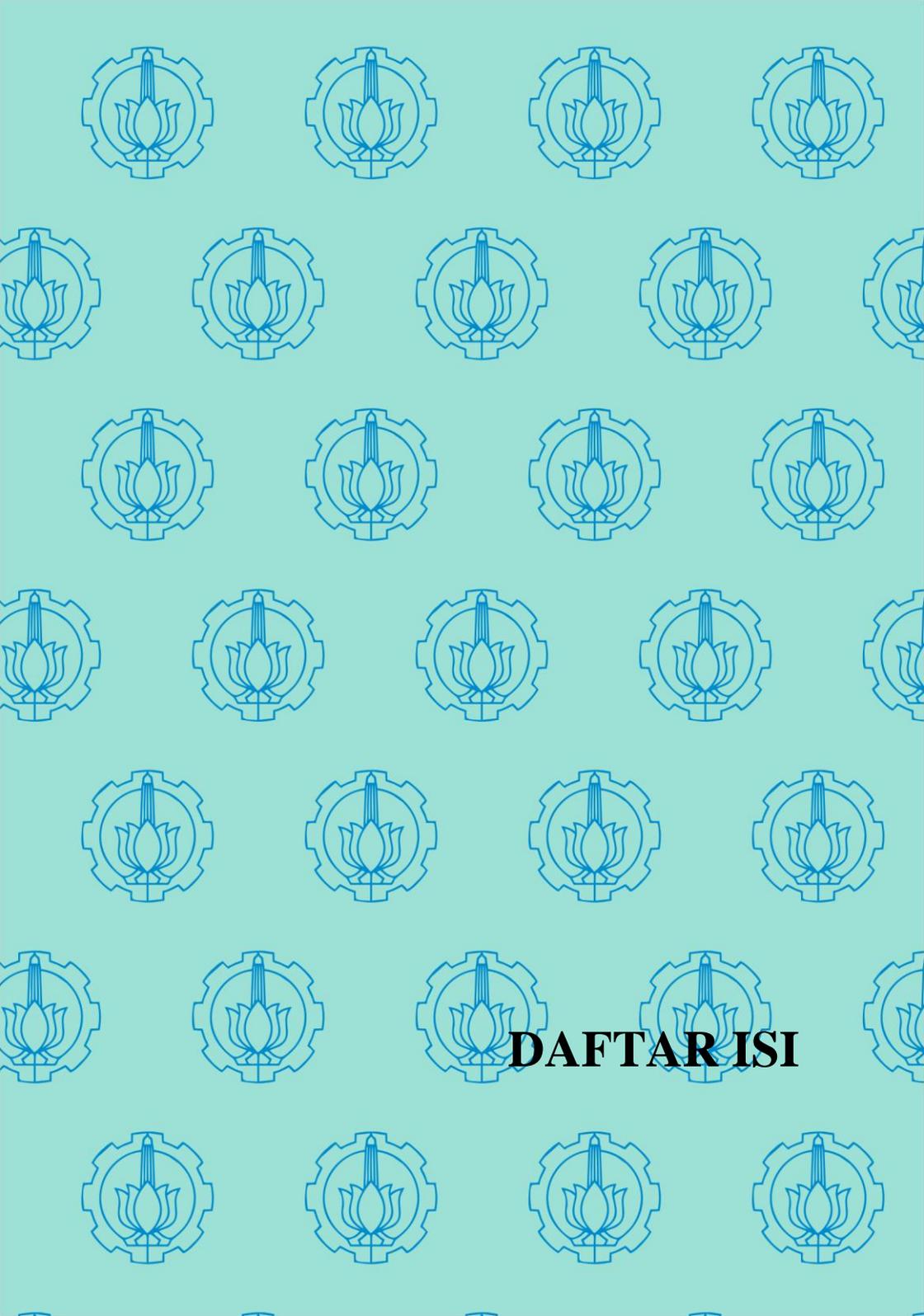
Laporan tugas akhir ini bisa diselesaikan tidak terlepas dari ulur tangan pihak ketiga yang meliputi dosen pembimbing dan pembina laboratorium teknik pengaturan serta rekan-rekan sekalian yang telah senantiasa mendukung selama penelitian yang penulis kerjakan, sehingga penulis dengan tulus mengucapkan kasih yang sebesar-besarnya.

Penulis menyadari bahwa banyak kekurangan dari laporan tugas akhir ini, baik dari segi materi maupun teknis penyajiannya, mengingat masih kurangnya pengetahuan dan pengalaman penulis. Namun pengalaman dan wawasan baru yang penulis banyak dapatkan juga patut untuk penulis syukuri dan diterapkan menjadi ilmu yang manfaat.

Surabaya, 25 Maret 2019

Penulis

*Halaman ini sengaja dikosongkan*



# **DAFTAR ISI**

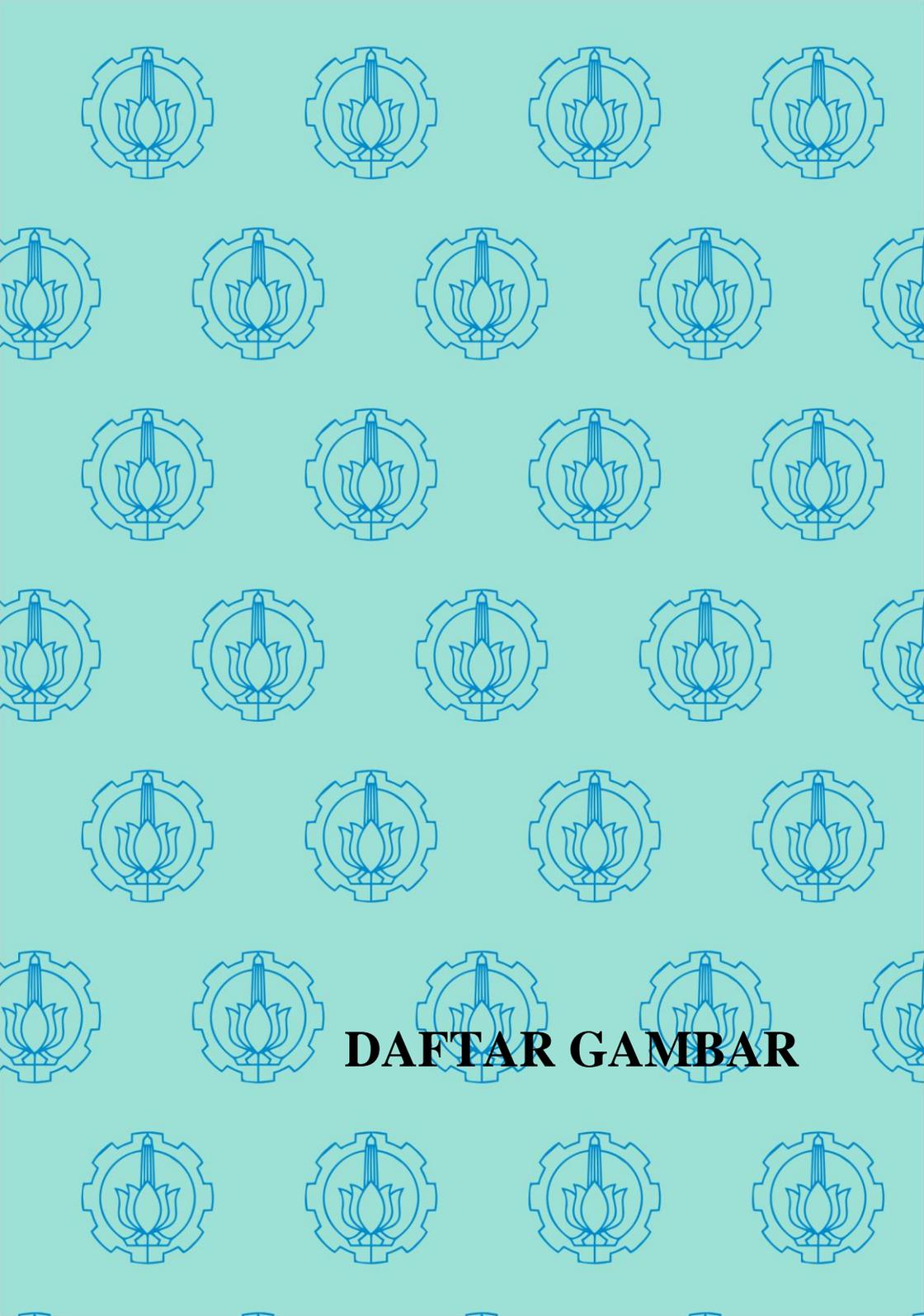
## DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
ABSTRAK.....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL.....	xxi
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	3
1.7 Relevansi.....	4
BAB 2 SISTEM PENGATURAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU.....	5
2.1 Sistem Kerja PLTU .....	5
2.2 <i>Boiler</i> pada Sistem PLTU .....	8
2.3 Permodelan Sistem <i>Boiler</i> .....	10
2.3.1 Permodelan <i>Steam Drum</i> .....	10
2.3.2 Permodelan Aktuator FPT .....	11
2.3.3 Hubungan Beban dengan Laju Aliran Uap .....	11
2.3.4 Permodelan <i>Blowdown Water</i> .....	12
2.4 Sistem Pengendalian pada <i>Steam Drum</i> .....	12
2.4.1 <i>Single-element Control</i> dan <i>Two-element Control</i> .....	12
2.4.2 <i>Three-element Control</i> .....	15
2.5 Perancangan Logika <i>Fuzzy</i> .....	16
2.5.1 <i>Fuzzification</i> .....	17
2.5.2 <i>Fuzzy Inference Rule</i> dan <i>Fuzzy Knowledge Base</i> .....	18
2.5.3 <i>Defuzzification</i> .....	18
2.6 Sistem Kendali dengan Logika <i>Fuzzy</i> .....	19

2.6.1	Kontroler Logika <i>Fuzzy</i> Tipe-PID .....	20
2.6.2	Kontroler Logika <i>Fuzzy</i> Adaptif .....	21
2.7	Pemrograman dengan LabVIEW .....	22
2.7.1	Bagian Utama LabVIEW .....	23
2.7.2	<i>Shared Variable Engine</i> .....	23
2.8	Tuning PID dengan Metode <i>Ziegler-Nichols Closed-loop</i> .....	23
<b>BAB 3 SIMULASI PENGENDALIAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU .....</b>		<b>25</b>
3.1	Perancangan Simulasi Sistem Pengendalian Level <i>Steam Drum</i> .....	25
3.1.1	Permodelan <i>Steam Drum</i> .....	25
3.1.2	Permodelan Feedwater Pump .....	27
3.1.3	Hubungan Beban dan Laju Aliran Uap .....	29
3.1.4	Permodelan Blowdown Rate dan Continuous Blowdown .....	29
3.1.5	Hubungan <i>Steam Flow</i> dengan Tekanan pada <i>Steam Drum</i> .....	29
3.2	Perancangan Kontroler .....	29
3.2.1	Proses Fuzzifikasi .....	30
3.2.2	Penentuan <i>Rule Base</i> .....	32
3.2.3	Inferensi <i>Mamdani</i> .....	33
3.2.4	Proses Defuzzifikasi .....	33
3.2.5	Melakukan <i>Scalling</i> Ulang .....	35
3.2.6	Penentuan Parameter Kontroler .....	35
3.2.7	Skema Kontroler Logika <i>Fuzzy</i> .....	35
3.3	Spesifikasi Sistem .....	37
3.4	Perancangan Sistem Komunikasi antar HMI .....	37
3.5	Indeks Performansi Kriteria dan Tuning Sistem .....	42
3.5.1	Analisa Error pada Sistem dengan Kontroler .....	42
3.5.6	<i>Phase Plane Trajectory</i> .....	43
<b>BAB 4 PENGUJIAN DAN ANALISIS .....</b>		<b>45</b>
4.1	Pengujian Komponen Sistem .....	45
4.1.1	Pengujian aktuator BFPT secara <i>Open-loop</i> .....	45
4.1.2	Pengujian <i>Plant Steam Drum</i> .....	46

4.2	Analisa Kontroler pada <i>Single-element Control</i> .....	46
4.2.1	Pengujian Sistem dengan Kontroler PID Konvensional .....	47
4.3	Pengujian Kontroler Logika <i>Fuzzy</i> .....	50
4.3.1	Kontroler Logika <i>Fuzzy</i> Tipe-PI <i>Robust Self-tuning</i> ....	50
4.3.2	Kontroler Logika <i>Fuzzy</i> Tipe-PD <i>Robust Self-tuning</i> ..	53
4.3.3	Kontroler Logika <i>Fuzzy</i> Tipe-PID <i>Robust Self-tuning</i> .	57
4.3.4	Pengujian Parameter Normalisasi Kontroler <i>Fuzzy</i> Tipe PID.....	61
4.4	Analisa Kontroler pada <i>Three-element Control</i> .....	66
4.4.1	Pengujian Kontroler pada Sistem <i>Inner-loop</i> .....	66
4.4.2	Pengujian Kontroler pada Sistem <i>Outer-loop</i> .....	68
4.5	Pengujian Sistem terhadap <i>Set-point</i> Naik .....	73
4.6	Pengujian Sistem terhadap Perubahan Beban Acak .....	74
BAB 5 KESIMPULAN.....		77
LAMPIRAN.....		81
RIWAYAT PENULIS .....		97

*Halaman ini sengaja dikosongkan*

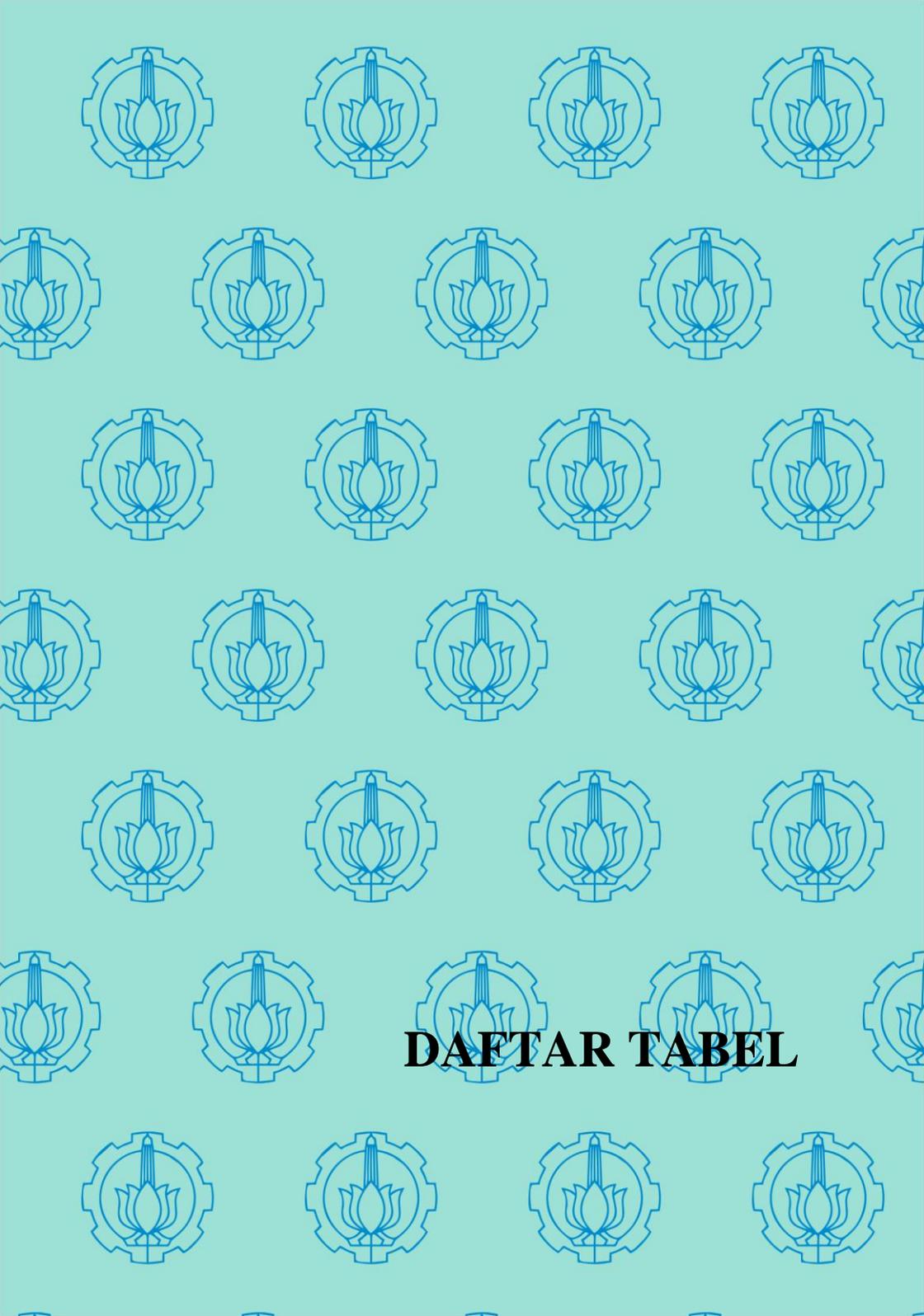


# DAFTAR GAMBAR

## DAFTAR GAMBAR

Gambar 2.1 Siklus Air dan Uap Air pada PLTU .....	6
Gambar 2.3 Gambar model <i>steam drum</i> .....	10
Gambar 2.4 Diagram P&ID <i>single-element control</i> .....	13
Gambar 2.5 Diagram blok <i>single-element control</i> .....	13
Gambar 2.6 Diagram P&ID <i>two-element control</i> .....	14
Gambar 2.7 Diagram blok <i>two-element control</i> .....	15
Gambar 2.8 Diagram P&ID <i>three-element control</i> .....	15
Gambar 2.9 Diagram blok <i>three-element control</i> .....	16
Gambar 2.10 Contoh fungsi keanggotaan segitiga dalam <i>crisp</i> .....	17
Gambar 2.11 <i>Jenis-jenis fungsi keanggotaan dalam diagram crisp</i> ....	18
Gambar 2.12 Rancangan pengendalian logika <i>fuzzy</i> .....	19
Gambar 2.13 Kontroler logika <i>fuzzy</i> tipe PD .....	20
Gambar 2.14 Kontroler logika <i>fuzzy</i> tipe PI.....	20
Gambar 2.15 Kontroler logika <i>fuzzy</i> tipe PID .....	21
Gambar 2.16 <i>Fuzzy PID</i> dengan satu skema <i>fuzzy sets</i> .....	21
Gambar 2.17 <i>Bagian virtual instrument (VI) pada LabVIEW</i> .....	22
Gambar 2.18 Respons sistem <i>closed-loop</i> dengan <i>gain-marginal</i> .....	24
Gambar 3.1 Desain model fisik <i>steam drum</i> .....	25
Gambar 3.2 Hasil regresi aktuator BFPT .....	28
Gambar 3.3 Kontroler <i>fuzzy</i> tipe PID <i>robust-self tuning</i> .....	30
Gambar 3.4 <i>Membership function</i> sinyal error .....	31
Gambar 3.5 <i>Membership function</i> sinyal delta-error .....	31
Gambar 3.6(a) Defuzzifikasi <i>singleton</i> pada <i>Controller</i> .....	34
Gambar 3.6(b) Defuzzifikasi <i>singleton</i> pada <i>Gain-tuning Mechanism</i> .	34
Gambar 3.7 Rancangan konroler logika <i>fuzzy</i> tipe-PD .....	36
Gambar 3.8 Rancangan konroler <i>fuzzy</i> tipe-PI.....	36
Gambar 3.9 Rancangan konroler <i>fuzzy</i> tipe-PID .....	37
Gambar 3.10 Skema pertukaran data dengan <i>shared variabel engine</i> ..	38
Gambar 3.11 <i>Shared Variable Engine (SVE)</i> pada NI-PSP.....	39
Gambar 3.12 Tampilan HMI pada menu <i>Overview</i> .....	40
Gambar 3.13 Tampilan HMI pada menu <i>Steam Drum</i> .....	40
Gambar 3.14 Tampilan HMI pada menu <i>Fuzzy-sets</i> .....	41
Gambar 3.15 Tampilan window <i>project</i> pada Simulator .....	41
Gambar 3.16 Rancangan <i>valve</i> tertutup ketika <i>boiler trip (MFT)</i> .....	42
Gambar 3.14 Diagram <i>phase plane trajectory</i> .....	44
Gambar 4.1 Respons model BFPT terhadap waktu sampling .....	45
Gambar 4.2 Respons laju perubahan level air pada <i>Steam Drum</i> .....	46

Gambar 4.3 Sistem <i>closed-loop</i> dengan <i>disturbance</i> .....	47
Gambar 4.4 Respons <i>Closed-loop</i> Sistem dengan <i>gain marginal</i> .....	48
Gambar 4.5 Respons sistem tanpa gangguan .....	48
Gambar 4.6 Respons input-output aktuator BFPT dengan gangguan ...	49
Gambar 4.7 Respons tanpa gangguan <i>fuzzy-PI</i> .....	50
Gambar 4.8 Respons tanpa gangguan <i>fuzzy-PI robust self-tuning</i> .....	51
Gambar 4.9 Respons tanpa gangguan <i>fuzzy-PI</i> .....	52
Gambar 4.10 Respons dengan gangguan <i>fuzzy-PI</i> .....	53
Gambar 4.10 Respons tanpa gangguan <i>fuzzy-PD</i> .....	54
Gambar 4.11 Respons tanpa gangguan <i>fuzzy-PD robust self-tuning</i> .....	54
Gambar 4.12 Respons sistem dengan parameter PD yang sama .....	55
Gambar 4.13 Respons tanpa gangguan <i>fuzzy-PD</i> .....	56
Gambar 4.14 Respons sistem dengan <i>fuzzy-PD</i> terhadap gangguan .....	57
Gambar 4.15 <i>Fuzzy-PD</i> dengan <i>fuzzy-PID robust self-tuning</i> .....	58
Gambar 4.16 Respons sistem dengan <i>fuzzy-PID</i> terhadap gangguan .....	59
Gambar 4.17 Respons sistem terhadap perubahan nilai PI .....	60
Gambar 4.18 Respons sistem terhadap perubahan $G_e$ tanpa gangguan .....	61
Gambar 4.19 Respons sistem terhadap nilai $G_e$ dengan gangguan .....	63
Gambar 4.20. <i>Phase plane trajectory</i> terhadap perubahan $G_e$ .....	63
Gambar 4.21 Respons sistem terhadap nilai $G_{de}$ dengan gangguan .....	64
Gambar 4.22 <i>Phase plane trajectory</i> terhadap perubahan $G_{de}$ .....	65
Gambar 4.23 Respons BFPT dengan <i>gain marginal</i> .....	67
Gambar 4.24 Respons BFPT dengan kontroler PI .....	67
Gambar 4.25 Respons sistem pada <i>three-element control</i> .....	68
Gambar 4.26 <i>Phase plane trajectory</i> terhadap variasi nilai PD .....	69
Gambar 4.27 Respons sistem dengan variasi nilai PI .....	70
Gambar 4.28 <i>Phase plane trajectory</i> terhadap variasi nilai PI.....	70
Gambar 4.29 Respons sistem terhadap variasi nilai $G_e$ dan $G_{de}$ .....	71
Gambar 4.30 Respons sistem dengan <i>single</i> dan <i>three element control</i> .....	72
Gambar 4.31 Respons sistem terhadap perubahan <i>set-point</i> naik .....	73
Gambar 4.32 Respons sistem terhadap beban acak .....	74

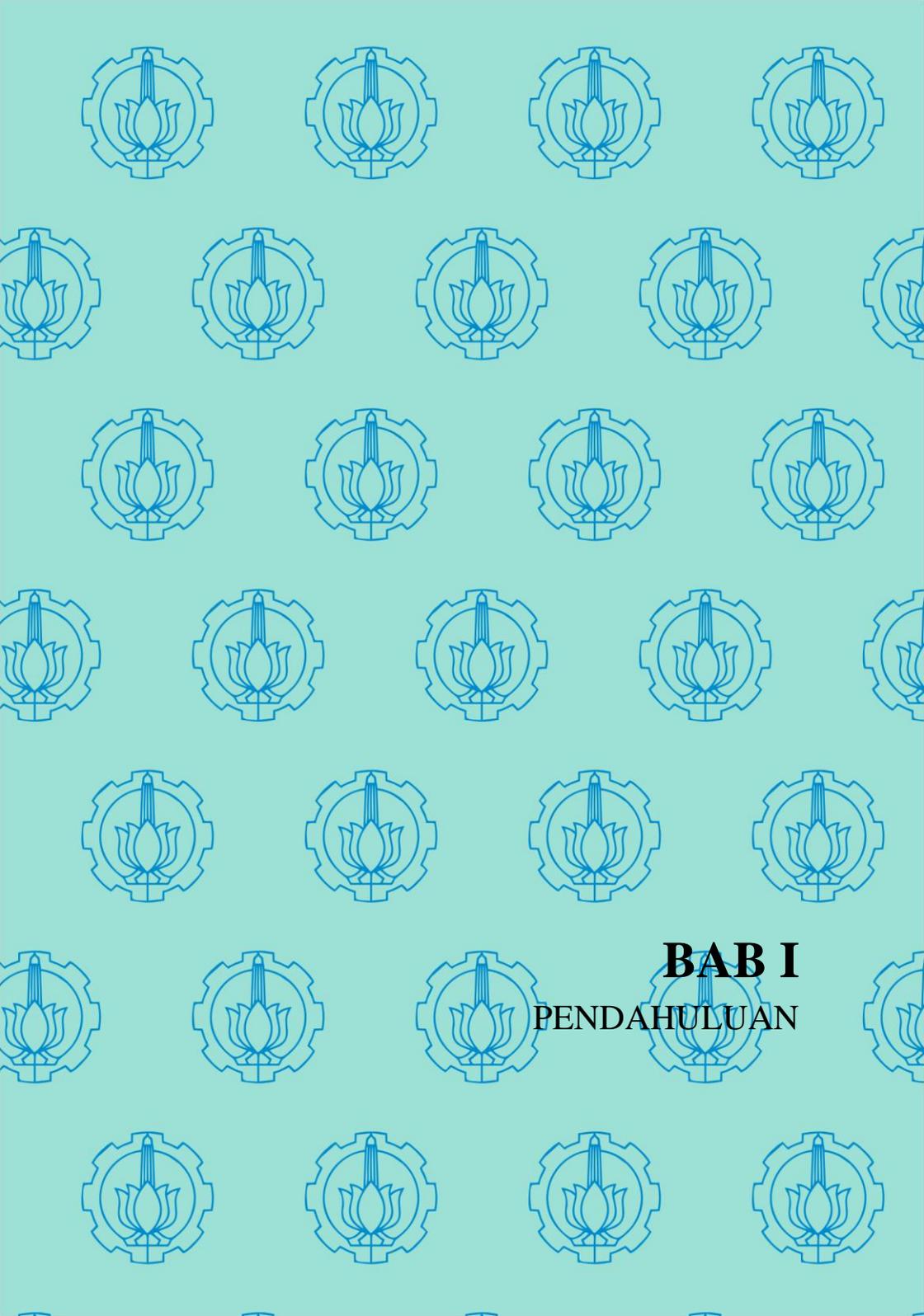


**DAFTAR TABEL**

## DAFTAR TABEL

Tabel 2.1 Tipe inferensi secara umum pada <i>fuzzy</i> .....	18
Tabel 2.2 <i>Tabel tuning ziegler-nichols metode closed-loop</i> .....	24
Tabel 3.1 <i>Rule-base</i> untuk <i>Controller</i> .....	32
Tabel 3.2 <i>Rule-base</i> untuk <i>Gain-tuning Mechanism</i> .....	33
Tabel 4.1 Kriteria respons sistem dengan kontroler PID .....	49
Tabel 4.2 Pengaruh skema <i>robust self-tuning</i> terhadap kriteria sistem .	52
Tabel 4.3 Pengaruh skema <i>robust self-tuning</i> terhadap kriteria sistem .	56
Tabel 4.4 Pengaruh skema <i>robust self-tuning</i> terhadap kriteria sistem .	57
Tabel 4.5 Kriteria sistem dengan PD = 35 dan PI = 0,06.....	60
Tabel 4.6 Kriteria sistem tanpa gangguan terhadap perbedaan Ge .....	62
Tabel 4.7 Kriteria sistem dengan gangguan terhadap perubahan Ge ....	62
Tabel 4.8 Kriteria sistem dengan gangguan terhadap perubahan Gde ..	65
Tabel 4.9 Hasil tuning parameter PID terhadap aktuator BFPT.....	66
Tabel 4.10 Kriteria sistem dengan gangguan terhadap perubahan PD ..	69
Tabel 4.11 Kriteria sistem dengan gangguan terhadap perubahan Gde	70
Tabel 4.12 Kriteria sistem dengan <i>single</i> dan <i>three element control</i> ....	71
Tabel 4.13 Kriteria sistem dengan <i>single</i> dan <i>three element control</i> ....	72
Tabel 4.14(a) Respons sistem pada <i>single</i> dan <i>three element control</i> ...	73
Tabel 4.14(b) Kriteria error sistem pada <i>single</i> dan <i>three element control</i> .....	74

*Halaman ini sengaja dikosongkan*



# **BAB I**

## **PENDAHULUAN**

# BAB 1

## PENDAHULUAN

Hal yang mendasari dilakukannya penelitian terkait metode pengendalian dengan menggunakan logika *fuzzy* ini mengacu pada model dari *plant steam drum* yang stabil marginal. Pengendalian level air didalam *steam drum* harus dijaga konstant untuk menghindari kerusakan komponen serta terkandungnya uap basah yang dihasilkan akibat level air yang terlalu tinggi. Sehingga sistem pada *steam drum* harus tahan terhadap segala macam gangguan (*robust*). Umumnya jenis pengendalian yang diterapkan pada sistem PLTU ini adalah pengendalian *single-element control* (biasa diterapkan pada saat *start-up* dan beban rendah) serta *three-element control* (biasa diterapkan pada beban tinggi dan fluktuasi).

### 1.1 Latar Belakang

PLTU merupakan merupakan suatu pembangkit listrik yang dirancang dengan memanfaatkan energi bahan bakar batu bara untuk menghasilkan uap sehingga dapat memutar turbin generator untuk menghasilkan energi listrik. Proses perubahan air menjadi uap dilakukan oleh sistem *boiler* yang menyediakan panas[1]. Uap yang dihasilkan tersebut akan dikendalikan sehingga putaran turbin pada generator konstan pada nilai  $50 \pm 0,5$  Hz terhadap perubahan beban yang fluktuasi[2].

Di dalam sistem *boiler* tersebut terdapat komponen *steam drum* yang berperan dalam memisahkan campuran antara air dan uap. Level air didalam *steam drum* juga perlu dijaga konstan, sedangkan kebutuhan uap yang keluar dari *steam drum* juga berubah-ubah menyesuaikan beban listriknya. Karena itu diperlukan sistem pengaturan level pada *steam drum* agar sesuai dengan kondisi *normal water level*—nya.

Pengendalian level air pada *steam drum* merupakan salah satu dari beberapa kondisi yang perlu dijaga untuk menghindari *trip*. Oleh karena itu, dalam Tugas Akhir ini, akan dirancang suatu metode sistem kendali untuk mengatur level air pada *steam drum* agar tetap pada kondisi nominalnya terhadap perubahan variabel beban fluktuatif.

Metode pengaturan yang digunakan dalam Tugas Akhir ini adalah pengendalian secara heuristik dengan logika *fuzzy* yang lebih mudah diterapkan untuk sistem yang kompleks. Secara umum pengaturan *steam*

*drum* dapat dilakukan dengan metode *single element control* yang biasanya diterapkan pada beban <30% atau *three element control* yang lebih kompleks dari pada *single element control* untuk beban >30%.

Skema kontroler logika *fuzzy* yang akan diterapkan pada penelitian ini adalah menggunakan skema *robust self-tuning*[3] yang diharapkan lebih tahan terhadap gangguan serta dengan memiliki *adaptation gain* pada integrator akan dapat menghindari *wind-up* sistem akibat saturasi aktuator. Sedangkan untuk tipe *fuzzy control* yang diterapkan pada penelitian ini adalah pengendalian logika *fuzzy* dengan tipe PID[4].

## 1.2 Perumusan Masalah

Konsumen menggunakan energi listrik berbeda setiap waktu, sehingga menimbulkan fluktuasi beban pada pembangkit. Jika beban pembangkit meningkat, maka laju aliran uap untuk memutar turbin pada pembangkit semakin besar. Hal ini menyebabkan air pada *steam drum* berkurang, sehingga level airnya turun. Jika kontroler tidak dapat menjaga level air pada batas yang diizinkan, maka sistem proteksi pembangkit akan bekerja untuk menghentikan operasi *boiler*. Berdasarkan hal tersebut, pembangkit membutuhkan sistem pengaturan untuk mengatur level air pada *steam drum* pada nilai *set-point*.

Level air pada *steam drum* berubah-ubah berdasarkan laju aliran uap yang keluar dari drum. Oleh karena itu, sistem pengaturan level air di dalam *steam drum* membutuhkan kontroler dengan kemampuan meredam gangguan yang baik agar dapat memenuhi spesifikasi yang diinginkan.

Penggunaan listrik oleh konsumen tidak tetap dan akan selalu berubah tiap waktunya, yang menyebabkan nilai perubahan beban listrik yang tidak menentu. Selain itu, beban yang terlalu tinggi juga dapat berdampak negatif bagi sistem boiler. Akibat dari kondisi beban tinggi dan berubah-ubah adalah energi uap yang dibutuhkan oleh *steam turbine* juga akan menjadi berubah-ubah tidak menentu.

## 1.3 Batasan Masalah

Pada penelitian ini dilakukan permodelan sistem dengan pendekatan linier untuk dilakukan *modelling* secara hukum fisis serta dengan proses regresi linier berdasarkan data yang didapatkan di PT. PJB UBJ O&M Paiton. Sistem yang dirancang pada penelitian ini merupakan sistem pengendalian level air pada *steam drum* ketika kondisi operasi atau *running* dengan daya listrik 400MW hingga 700MW. Jenis pengendalian yang diterapkan pada penelitian ini adalah

kontroler logika *fuzzy* tipe PI, PD, dan PID dengan skema *robust self-tuning*.

#### **1.4 Tujuan Penelitian**

Kontroler logika *fuzzy* yang dirancang untuk digunakan dalam sistem pengendalian level air pada *steam drum* sehingga sistem tersebut dapat meredam gangguan dan menjaga nilai *set-point* dari level sesuai dengan spesifikasi yang telah ditentukan.

#### **1.5 Metodologi**

Pada penelitian ini tahap awal akan dilakukan studi literatur untuk mencari referensi yang menunjang wawasan dan pengetahuan terkait metode *fuzzy* PID[4] serta sistem pengaturan level air pada *steam drum*[2]. Literatur dapat berasal dari buku, jurnal, *paper*, atau data teknik perusahaan. Kemudian dilanjutkan dengan perancangan sistem, dimulai dari permodelan *plant*[6,7] dan aktuator BFPT[13] serta menentukan desain kontroler[3] dan spesifikasi[15] yang diharapkan. Setelah dilakukan perancangan kemudian akan dilakukan analisa dan pengujian sistem untuk mendapatkan suatu simpulan dari penelitian yang dilakukan. Rancangan sistem tersebut selanjutnya akan diimplementasikan ke dalam program Labview untuk difungsikan sebagai simulator pengaturan level air pada *steam drum* di PLTU dengan komunikasi menggunakan bantuan *shared variabel engine*[8] melalui jaringan *ethernet*.

#### **1.6 Sistematika Penulisan**

Pada penelitian ini disusun sistematika penulisan sesuai dengan penjelasan sebagai berikut :

## **BAB 1 PENDAHULUAN**

Pada BAB ini dijelaskan tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, serta relevansi dari penelitian yang dilakukan.

## **BAB 2 SISTEM PENGATURAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU**

Pengetahuan tentang *steam drum* dalam penelitian ini mendasari konsep dari perancangan simulator PLTU dan permodelan sistemnya. Selain itu juga menjelaskan terkait skema pengendalian *single element control* dan *three element control*, beserta juga langkah untuk mendapatkan parameter kontrolernya.

## **BAB 3 SIMULASI PENGENDALIAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU**

Perancangan pengendalian sistem yang akan digunakan serta desain tampilan HMI untuk Simulator PLTU, yang selanjutnya rancangan sistem tersebut akan diimplementasikan pada program LabVIEW.

## **BAB 4 PENGUJIAN DAN ANALISIS**

Pada desain sistem yang telah dirancang dalam penelitian ini akan dilakukan pengujian dan analisa terhadap ketahanan serta indeks performansi yang dihasilkan terhadap gangguan yang terdapat pada sistem.

## **BAB 5 KESIMPULAN**

Pada BAB ini akan ditarik suatu kesimpulan terkait hasil yang telah dicapai pada penelitian yang telah dilakukan.

### **1.7 Relevansi**

Pengendalian level air di dalam *steam drum* menjadi masalah umum pada sistem *boiler*, termasuk juga yang terdapat di PLTU untuk menghindari *trip*. Terlebih lagi sistem yang terdapat di dalamnya dapat dikatakan memiliki kompleksitas yang cukup tinggi, dengan demikian pengendalian secara heuristik seperti *fuzzy logic controller* akan menjadi opsi utama untuk diimplementasikan pada sistem tersebut. Pada penelitian ini, metode pengendalian tersebut diterapkan pada simulator PLTU yang dapat menjadi alat bantu pembelajaran untuk memahami serta memberikan gambaran terkait konsep dari sistem *steam drum* yang terdapat pada PLTU.



**BAB II**  
**SISTEM PENGATURAN LEVEL AIR DI**  
**DALAM *STEAM DRUM* PADA PLTU**

## **BAB 2**

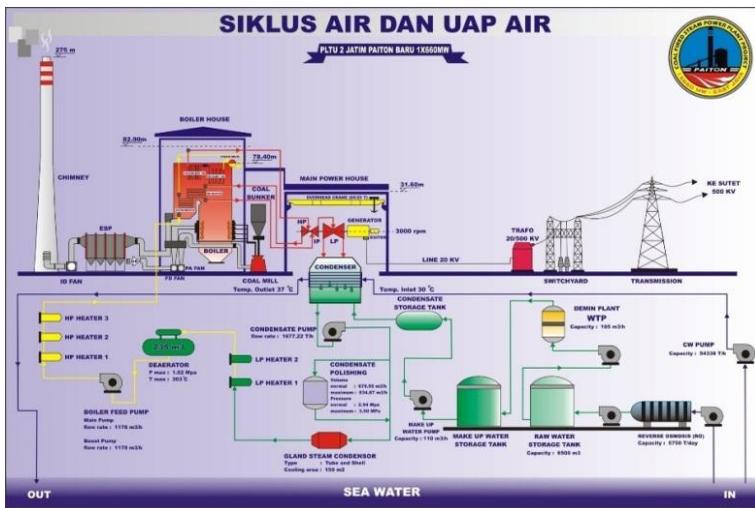
### **SISTEM PENGATURAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU**

PLTU merupakan pembangkit listrik yang memanfaatkan energi dari pembakaran batu bara untuk memanaskan air sehingga menghasilkan uap bertekanan tinggi. Sistem proses pembakaran untuk menghasilkan uap tersebut sering disebut juga sebagai sistem *boiler*. Uap tersebut akan dikendalikan sehingga dapat digunakan untuk memutar turbin dengan kecepatan konstan, dengan demikian karakteristik uap yang keluar dari *boiler* akan berubah-ubah. Sehingga level air di dalam *steam drum* juga perlu dikendalikan agar mampu menyesuaikan perubahan kondisi uap tersebut, salah satunya dengan cara mengendalikan aliran nilai *feedwater*.

Pengendalian level air pada *steam drum* umumnya menggunakan skema *single element control* dan *three element control (cascade control)* yang cukup kompleks, sehingga metode pengendalian secara heuristik menggunakan *fuzzy control* tipe PID menjadi opsi utama untuk diterapkan dalam sistem tersebut.

#### **2.1 Sistem Kerja PLTU**

PLTU merupakan suatu sistem yang kompleks seperti yang dipaparkan pada *Gambar 2.1* dimana sistem tersebut terdiri dari begitu banyak proses hingga sebelum akhirnya mampu untuk menghasilkan uap jenuh bertekanan tinggi. Namun pada dasarnya produksi listrik di PLTU terdiri dari dua sub sistem atau proses utama, diantaranya adalah sistem *boiler* untuk menghasilkan uap serta sistem *steam turbine* yang berperan dalam menghasilkan daya listrik. Sistem pemanasan air pada *boiler* sehingga menjadi uap difungsikan untuk menggerakkan turbin dari hasil uap bertekanan tinggi tersebut. Selanjutnya turbin akan memutar generator dan menghasilkan tenaga listrik. Sedangkan uap pembuangan dari turbin dijadikan air kembali dengan proses kondensasi. Hal ini dilakukan secara terus menerus (berulang) sehingga membentuk siklus tertutup.



**Gambar 2.1** Siklus Air dan Uap Air pada PLTU

Air Laut dimurnikan dengan sistem penyaringan (*Reverse Osmosis*) dan sistem penukaran ion (*Ionic Exchanger*) melalui peralatan *Water Treatment Plant* (WTP). Air murni yang dihasilkan oleh WTP harus memenuhi syarat untuk disalurkan melalui sistem pengisian air kedalam *Boiler* sehingga dapat meningkatkan *lifetime* dari peralatan di sistem PLTU tersebut.

Proses pembakaran awal (*Start Up*) pada PLTU dimulai dengan memanaskan air menjadi uap digunakan bahan bakar minyak HSD (*High Speed Diesel*) yang diproses di *Auxiliary Boiler* (Boiler mini).

*High Speed Diesel* merupakan jenis bahan bakar solar yang digunakan untuk sebuah mesin dengan kecepatan putaran mesin diatas 1000rpm. Pada proses *start-up* selain dilakukan pemanasan air juga ditetapkan level air harus dijaga tetap dalam batas tertentu untuk menghindari *trip* yang mengharuskan untuk mengulang proses *start-up* mulai dari awal, sedangkan untuk proses selanjutnya menggunakan bahan bakar batubara hingga beban maksimal.

Sistem pengisian air ke boiler berasal dari *feedwater*, yang dimana air didalam *feedwater* didapatkan dari hasil pemurnian air dari WTP (*make-up water*) serta juga berasal dari hasil proses kondensasi uap, dimana air hasil kondensasi uap (*water condensate*) yang tertampung di dalam *hot well* dipompa oleh *Condensate Extraction pump* (CEP)

kemudian sebagian air yang kualitasnya masih kurang dialirkan ke *Condensate Polisher Plant* (Unit Pengolahan Air Dengan Penukaran Ion). Setelah itu air akan dilewatkan ke pemanas tekanan rendah (LPH-1 dan LPH-2) selanjutnya ke *Dearator Storage Tank* (DST). Dari DST, air pengisian boiler dipompa dengan *Boiler Feed Pump* (BFP). Pada PLTU umumnya memiliki dua jenis BFP, yaitu *Boiler Feed Pump Motor* (BFP-M) yang biasa digunakan untuk proses *start-up* serta *Boiler Feed Pump Turbine* (BFP-T) yang biasa digunakan ketika sudah operasi. Air dari DST tersebut kemudian dilewatkan ke pemanas tekanan tinggi (HPH-1, HPH-2, HPH-3) kemudian ke *economizer*, *economizer* ini menyerap panas gas yang keluar dari boiler untuk pemanasan awal supaya air yang masuk ke *Steam Drum* memiliki temperatur tinggi. Dari *economizer*, air masuk ke *steam drum*.

Di dalam *steam drum* dihasilkan *saturated steam*. *Saturated steam* kemudian dilewatkan ke *superheater* sehingga dihasilkan uap yang benar-benar kering (*superheating steam*). Uap kering tersebut dialirkan ke turbin tekanan tinggi untuk memutar *High Pressure Turbine* (*HP Turbine*), dan sisa uap dari *HP turbine*. Uap yang keluar dari *HP Turbine* masih memiliki tekanan yang cukup tinggi, namun suhunya sudah terlalu rendah untuk digunakan memutar turbin kembali, sehingga uap tersebut akan dipanaskan lagi di *reheater*. Dari *reheater* uap tersebut dialirkan ke turbin tekanan menengah untuk memutar *Intermediate Pressure Turbine* (*IP Turbine*) dan selanjutnya sisa uap dari *IP Turbine* dialirkan ke turbin tekanan rendah (*LP Turbine*) yang terdiri dari dua buah turbin dengan diameter yang lebih besar (*LP Turbine A* dan *LP Turbine B*). Poros turbin-turbin tersebut saling terhubung (*coupled*) untuk memutar generator sehingga menghasilkan tenaga listrik.

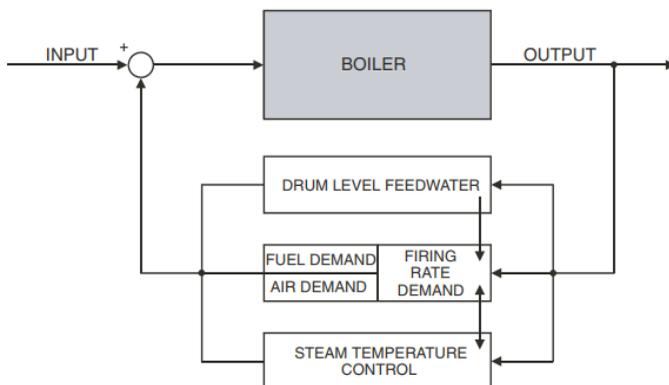
Sistem pengaturan untuk menjaga putaran turbin konstan tersebut dari permintaan beban konsumen yang tinggi serta fluktuatif (berubah-ubah) tersebut umumnya menggunakan *boiler follow*, *turbine follow*, atau *coordinate follow*. *Boiler follow* merupakan sistem kontrol dengan sistem pengaturan *firing rate* pada boiler akan mengikuti atau bergantung pada kondisi tekanan *throttle valve* (*TV*) pada *steam turbine* terhadap perubahan beban. Sedangkan pada *turbine follow*, kondisi dari *throttle valve* akan bergantung pada kondisi tekanan uap yang dihasilkan oleh boiler dengan mengatur *firing rate*-nya terhadap perubahan beban. Kemudian pada *coordinate control* merupakan perpaduan antara boiler

*follow* dan *turbine follow* yang dimana perubahan beban akan mempengaruhi baik *control firing rate* maupun *control boiler*.

Generator PLTU umumnya dapat menghasilkan tegangan listrik hingga sebesar 20 KV dengan daya yang bervariasi (sekitar 660MW), dan kemudian tegangan tersebut dinaikkan oleh *Main Transformer* menjadi 500 KV untuk mengurangi rugi-rugi *resistif* yang berasal dari kabel transmisi.

## 2.2 Boiler pada Sistem PLTU

*Boiler* merupakan suatu bejana tertutup yang di dalamnya menampung air yang telah dipanaskan serta juga uap yang dihasilkan melalui siklus seperti pada *Gambar 2.2*, yang kemudian didalamnya akan dipisahkan oleh *separator*. Untuk Hasil keluaran energi panas dari uap tersebut biasa disebut dengan *main steam* yang memiliki temperatur serta tekanan yang paling tinggi, dan akan digunakan untuk memutar *HP Turbine* pada sistem Pembangkit Listrik Tenaga Uap (PLTU). *Main steam* tersebut akan menyebabkan generator dapat menghasilkan keluaran energi listrik berupa tegangan AC dengan frekuensi tertentu. Frekuensi yang dihasilkan oleh generator juga akan bergantung pada kecepatan dari putaran turbin tersebut. Mengingat sumber listrik yang digunakan sehari-hari adalah sumber tegangan AC dengan frekuensi 50Hz, maka PLTU tersebut juga harus memastikan agar turbin pada generator berputar sedemikian rupa sehingga keluaran frekuensi tegangan AC dijaga konstan di 50Hz dengan mengatur *Governor Valve* (GV), *Intermediate Valve* (IV), serta *Trotle Valve* (TV) pada *steam turbine*.



**Gambar 2.2** Blok diagram sistem boiler

Selain menjaga konstan putaran turbin dengan mengatur *valve* tersebut, sebelumnya juga harus dipastikan kondisi dari *main steam* memutar turbin tersebut memenuhi kebutuhan dari *steam turbine* untuk menjaga konstan kecepatan 3000 rpm tersebut. Selain tekanan uap yang harus dijaga agar putaran turbin pada generator konstan di suatu nilai tertentu, juga harus diperhatikan variabel lainnya seperti temperatur dan kelembaban dari uap. Uap diusahakan tidak mengandung air (*superheated steam*) saat memutar turbin pada generator agar peralatan yang dilewati uap seperti saluran pipa dan turbin tersebut tidak mengalami korosi. Selain itu untuk temperatur uap pada dasarnya bila semakin tinggi maka akan lebih meningkatkan efisiensi, hanya saja peralatan pada industri juga memiliki batas kemampuan temperatur yang mampu ditampungnya.

Untuk menjaga kualitas uap tersebut agar tetap kering salah satunya dilakukan dengan mengatur level air didalam *steam drum* agar tidak terlalu tinggi. Kondisi level air didalam *steam drum* yang terlalu rendah juga perlu dihindari agar tidak terjadi *over pressure* yang juga dapat membahayakan peralatan itu sendiri. Oleh karena itu, level air di dalam *steam drum* perlu dijaga pada titik optimalnya agar tidak terlalu tinggi serta juga tidak terlalu rendah.

Pengendalian level air di dalam *steam drum* dapat melibatkan beberapa variabel seperti *feedwater flow*, *steam flow*, tekanan di dalam *steam drum*, serta juga temperatur. Keempat variabel tersebut sebenarnya saling berkaitan dan saling mempengaruhi satu sama lain, dengan *major system* dari *boiler* yang akan dikontrol tersebut adalah level air. Level air lebih menentukan kondisi uap yang akan dihasilkan, dan untuk pengendalian level air tersebut umumnya juga memerlukan variabel lainnya, diantaranya adalah *steam flow* serta *feedwater flow*.

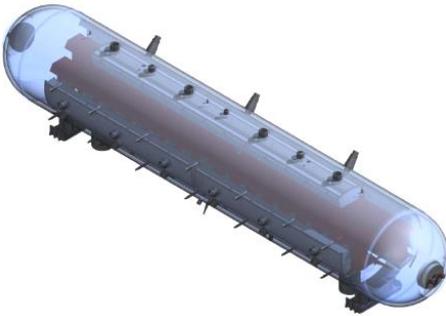
Selain itu, untuk meningkatkan efisiensi dan *lifetime* dari peralatan di dalam *boiler* tersebut harus terbebas dari kotoran atau zat kimia lainnya. Oleh karena itu pada *feedwater* perlu dipastikan sudah mengandung air murni, dan juga untuk mengantisipasi adanya kotoran pada *steam drum* maka perlu disediakan *blowdown water* untuk membuang air yang mengandung kotoran tersebut. Proses *blowdown water* ini umumnya ada yang terus-menerus secara kontinu dan ada juga yang periodik (berkala) sesuai dengan kondisi air yang ada, proses tersebut dijalankan manual oleh *operator* tersebut.

### 2.3 Permodelan Sistem Boiler

Beberapa komponen pada sistem boiler akan dimodelkan baik dengan permodelan analitik melalui hukum fisis, identifikasi data, atau juga berdasarkan referensi. Komponen-komponen yang akan dimodelkan diantaranya adalah *plant steam drum*, aktuator *Boiler Feedwater Pump Turbine* (BFPT), serta beberapa variabel lain seperti grafik hubungan antara beban dengan tekanan dan aliran uap, atau dapat juga berupa *blowdown rate* sebagai gangguan.

#### 2.3.1 Permodelan Steam Drum

Permodelan dilakukan dengan hukum fisis melalui persamaan kesetimbangan massa, dimana jumlah perubahan massa yang terjadi di dalam *steam drum* dinyatakan dalam selisih antara jumlah massa air dari *feedwater* (*fw*) yang masuk dengan jumlah massa *steam* (*st*) atau uap air yang keluar.



**Gambar 2.3** Gambar model *steam drum*

Secara matematis persamaan kesetimbangan massa tersebut dapat dinyatakan dalam persamaan (2.2.1).

$$M_{fw}(t) - M_{st}(t) = \partial M_{total}(t) \dots\dots\dots (2.1.1)$$

$$M_{fw}(t) - M_{st}(t) = \frac{\partial[\rho_{fw}V_{fw}(t) + \rho_{st}V_{st}(t)]}{\partial t} \dots\dots\dots (2.1.2)$$

$$M_{fw}(t) - M_{st}(t) = \rho_{fw} \cdot A \cdot \frac{\partial h(t)}{\partial t} + \rho_{st} \frac{\partial(V_{drum} - V_{fw})}{\partial t} \dots\dots\dots (2.1.3)$$

$$M_{fw}(t) - M_{st}(t) = \rho_{fw} \frac{\partial V_{fw}}{\partial t} + \rho_{st} \frac{\partial V_{drum}}{\partial t} - \rho_{st} \frac{\partial V_{fw}}{\partial t} \dots\dots\dots (2.1.4)$$

$$M_{fw}(t) - M_{st}(t) = A \cdot \frac{\partial h(t)}{\partial t} \cdot (\rho_{fw} - \rho_{st}) + \rho_{st} \frac{\partial V_{drum}}{\partial t} \dots\dots\dots (2.1.5)$$

Volume drum dianggap tetap, sehingga dapat dinyatakan perubahan volume drum terhadap waktu adalah nol, sehingga didapatkan persamaan (2.1.7).

$$M_{fw}(t) - M_{st}(t) = A \cdot \frac{\partial h(t)}{\partial t} \cdot (\rho_{fw} - \rho_{st}) \dots\dots\dots (2.1.6)$$

$$\frac{\partial h(t)}{\partial t} = \frac{M_{fw}(t) - M_{st}(t)}{A \cdot (\rho_{fw} - \rho_{st})} \dots\dots\dots (2.1.7)$$

Fungsi luas permukaan air ( $A$ ) pada *steam drum* dapat dinyatakan melalui persamaan (2.1.9).

$$A_{drum(t)} = A_{bola(t)} + A_{tabung(t)} \dots\dots\dots (2.1.8)$$

$$A_{drum(t)} = \left( \pi(2rh_{(t)} - h_{(t)}^2) \right) + \left( 2l \sqrt{2\pi rh_{(t)} - h_{(t)}^2} \right) \dots\dots (2.1.9)$$

**2.3.2 Permodelan Aktuator FPT**

Proses permodelan untuk aktuator BFPT dapat dilakukan dengan metode *least Square* dengan mencari hubungan antara *input-output* dengan masing-masing bobot parameter tertentu. Untuk sistem yang memiliki *bias* yang dirumuskan sebagai berikut :

$$y(k) = a \cdot y(k - 1) + b \cdot u(k) \dots\dots\dots (2.2.1)$$

$$Y = \varphi \cdot \theta \dots\dots\dots (2.2.2)$$

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix} \dots\dots\dots (2.2.3)$$

$$\varphi = \begin{bmatrix} Y(0) & u(0) \\ Y(1) & u(1) \\ \vdots & \vdots \\ Y(k - 1) & u(k - 1) \end{bmatrix} \dots\dots\dots (2.2.4)$$

Sehingga didapatkan nilai, dengan estimasi *output* didapatkan persamaan (2.2.5) dengan estimasi  $y$  adalah seperti yang terdapat pada pada persamaan (2.2.6).

$$\theta = (\varphi' \cdot \varphi)^{-1} \cdot \varphi' \cdot Y \dots\dots\dots (2.2.5)$$

$$\hat{Y} = \varphi \cdot \theta \dots\dots\dots (2.2.6)$$

**2.3.3 Hubungan Beban dengan Laju Aliran Uap**

Hubungan antara beban dengan laju aliran uap juga dapat dimodelkan dengan menggunakan metode *least square*. Permodelan pada *least square* tidak hanya dapat dilakukan untuk sistem dinamis, untuk memodelkan sistem secara statis yaitu mencari hubungan

persamaan garis lurus dengan *bias* tertentu yang dapat dirumuskan seperti pada persamaan (2.2.1), (2.2.2), (2.2.3) dan (2.2.4).

Parameter *a* pada persamaan (2.2.3) merepresentasikan *bias* yang terdapat pada suatu sistem statis atau dapat dikatakan juga konstanta pada persamaan garis lurus, sedangkan parameter *b* dari persamaan di atas menyatakan gradien atau kemiringan dari persamaan garis lurus yang dihasilkan. Sehingga didapatkan seperti pada persamaan (2.2.5) dan (2.2.6).

### 2.3.4 Permodelan *Blowdown Water*

Berdasarkan dari referensi yang didapatkan didapatkan estimasi nilai untuk rating dari *flow blowdown* yang dihasilkan seperti pada persamaan (2.3.1), dimana *BR* adalah *Blowdown Rate*, *CC* adalah *Concentration Cycle*, dan *BC* adalah *Boiler Capacity*, sehingga didapatkan persamaan (2.3.2)

$$BR = \frac{1}{CC} BC \dots\dots\dots (2.3.1)$$

$$BR = \frac{1}{8,6} \cdot (\text{Flowrate Feedwater})^t / h \dots\dots\dots (2.3.2)$$

Sedangkan untuk *continous blowdown water*, pada umumnya memiliki rating nilai mulai 0,5% sampai dengan 3% dari *steam flow* yang dihasilkan.

## 2.4 Sistem Pengendalian pada *Steam Drum*

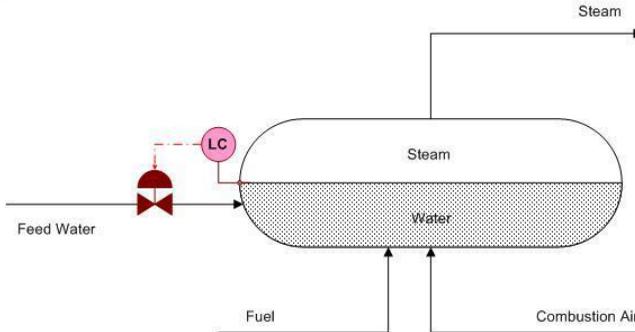
Sistem pengendalian pada *steam drum* secara umum dilakukan dengan tiga metode, diantaranya adalah *single-element control*, *two-element control*, dan *three-element control*. Untuk proses *starting* awal atau ketika beban rendah metode yang paling banyak digunakan adalah *single-element control* atau juga *two-element control* karena alasan sistemnya yang lebih sederhana, sedangkan ketika beban tinggi (*saat operasi*) akan dialihkan ke mode *three-element control*.

Pada dasarnya sesuai dengan namanya, perbedaan dari ketiga metode tersebut adalah jumlah variabel yang akan dikendalikan. Saat ini metode *two-element control* sudah jarang diterapkan dan lebih cenderung untuk menerapkan *three-element control*.

### 2.4.1 *Single-element Control* dan *Two-element Control*

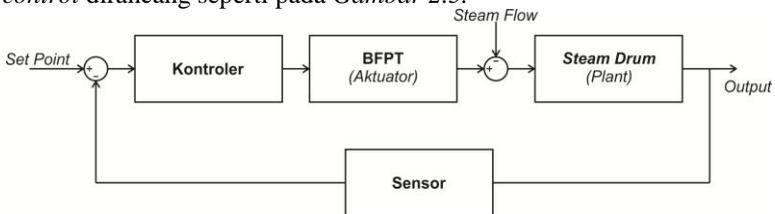
Pada metode ini variabel yang dikendalikan hanya level air di dalam *steam drum* saja, sehingga sistem dengan metode ini lebih

sederhana dan juga tuning parameter kontrolernya pun lebih mudah. Skema *single-element control* direpresentasikan dengan P&ID seperti pada *Gambar 2.4*.



**Gambar 2.4** Diagram P&ID *single-element control*

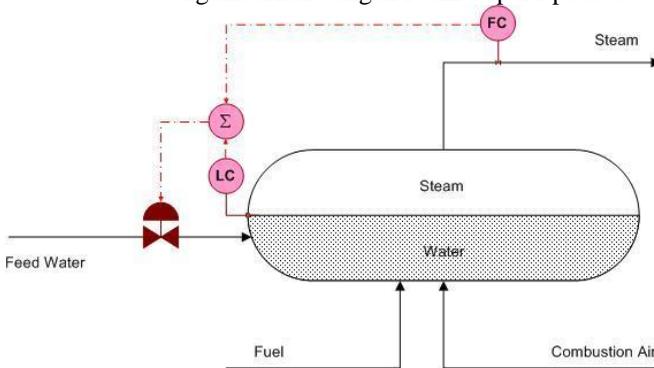
Namun karena hanya satu elemen saja yang dikendalikan metode ini tidak akan mampu menangani perubahan beban sangat cepat (fluktuatif) atau saat beban tinggi. Hal tersebut dikarenakan elemen yang dikendalikan hanyalah level. Pengondisian sinyal kontrol terjadi setelah adanya perubahan sinyal eror, artinya respons kontroler untuk memperbaiki sistem harus menunggu terjadinya perubahan level. Karena itu metode ini dikatakan memiliki respons yang lambat sehingga tidak cocok untuk sistem dengan *disturbance* yang fluktuatif dan tinggi. Diagram blok untuk sistem pengaturan level dengan *single-element control* dirancang seperti pada *Gambar 2.5*.



**Gambar 2.5** Diagram blok *single-element control*

Metode *two-element control* sebenarnya tidak berbeda jauh dengan *single-element control*. Pada metode ini, variabel yang dikendalikan selain level air respons kontroler juga akan bergantung pada cepat atau lambatnya aliran uap (*steam flow*) yang dihasilkan oleh *steam drum*.

Seperti yang telah diketahui bila perubahan laju aliran uap akan mempengaruhi pula pada perubahan level air di dalam *steam drum*. Naiknya aliran uap yang keluar dari steam drum akan dapat menurunkan level air dan juga berlaku sebaliknya, sehingga dapat dikatakan *steam flow* di sini sebagai *disturbance* untuk level air tersebut. Skema *two-element control* digambarkan dengan P&ID seperti pada *Gambar 2.6*.



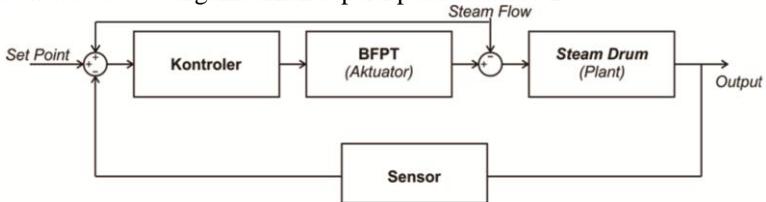
**Gambar 2.6** Diagram P&ID *two-element control*

Sebenarnya hubungan antara tekanan dengan steam flow berbanding lurus dan linier. *Steam flow* yang diterima oleh DCS dapat juga berupa hasil scaling dari nilai yang dihasilkan oleh *steam pressure transmitter*, dalam arti antara *steam flow* dengan *steam pressure* nilai pengukurannya didapatkan dari satu *transmitter* yang sama. Karena steam flow paling erat hubungannya dengan terjadinya perubahan level air di dalam steam drum metode kontrol ini menggunakan variabel tersebut.

Metode ini pada dasarnya muncul untuk menjawab kekurangan dari *single-element control*. Jika di sini *steam flow* yang menyebabkan perubahan level, maka diberikanlah elemen *steam flow* pada kontroler sehingga ketika steam flow berubah (ada *disturbance*), sebelum perubahan steam flow mempengaruhi level air kontroler tersebut merespons untuk memperbaiki sistem tersebut. Dengan demikian dapat dikatakan input *steam flow* tersebut berperan sebagai *disturbance rejection*. Respon kontroler menjadi tidak hanya menunggu terjadinya perubahan level melainkan juga juga perubahan pada *steam flow*.

Namun tetap saja metode ini masih memiliki kekurangan. Meskipun sudah dapat menangani *disturbance* yang tinggi dan

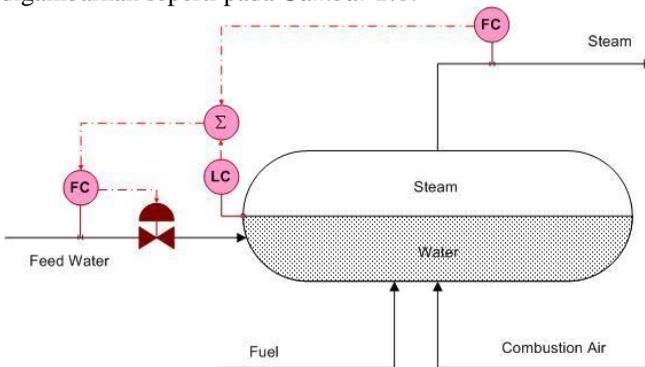
fluktuatif, ternyata dengan kenaikan *steam flow* dan *pressure* juga menyebabkan *pressure drop* terhadap *feedwater* sehingga *feedwater flow* akan turun dan tidak mampu memenuhi suplai kebutuhan dari steam drum. Diagram blok untuk sistem pengaturan level dengan *two-element control* digambarkan seperti pada Gambar 2.7.



**Gambar 2.7** Diagram blok *two-element control*

### 2.4.2 Three-element Control

Masalah utama fluktuasi beban serta *pressure drop* yang timbul pada *single-element control* dan *two-element control* menjadi alasan utama diterapkannya metode ini. *Three-element control* mampu mengatasi masalah tersebut melalui pengendalian 3 variabel yang meliputi level air di dalam *steam drum*, *steam flow*, serta juga dilengkapi lagi dengan *feedwater flow*. Skema *three-element control* dengan P&ID digambarkan seperti pada Gambar 2.8.

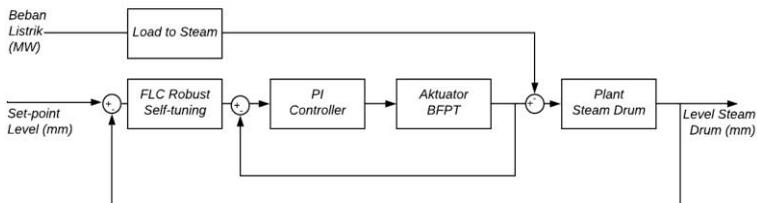


**Gambar 2.8** Diagram P&ID *three-element control*

*Pressure drop* dapat terjadi saat tekanan pada *steam drum* meningkat atau dapat juga ketika sebuah *feedwater* menyuplai banyak *boiler*. Tekanan tinggi pada *steam drum* biasa terjadi ketika kondisi beban sedang fluktuasi dan tinggi. Selain beban tinggi, kapasitas uap di

dalam *steam drum* yang meningkat menjadi jauh lebih banyak dari pada kapasitas airnya juga dapat menyebabkan tekanan di dalam *steam drum* juga akan meningkat. Akibatnya laju aliran air dari *feedwater* juga akan menjadi terhambat dan turun karena kenaikan tekanan di dalam *steam drum* tersebut.

Seperti yang telah diketahui bahwa kondisi level air di dalam *steam drum* bergantung pada kondisi *steam flow* dan *feedwater flow*, selain itu juga respons dari *feedwater flow* lebih cepat dari pada level air pada *steam drum*. Perancangan kontroler *cascade* seperti pada *Gambar 2.9* dilakukan dengan *feedwater* dengan akan dikendalikan oleh *secondary controller* pada *inner loop* serta *steam drum* akan dikendalikan oleh *primary controller* pada *outer loop*. Selain itu pada *secondary controller* juga akan diberikan *feedforward* dari *steam flow* sehingga disini didapat ada tiga elemen yang akan dikendalikan.



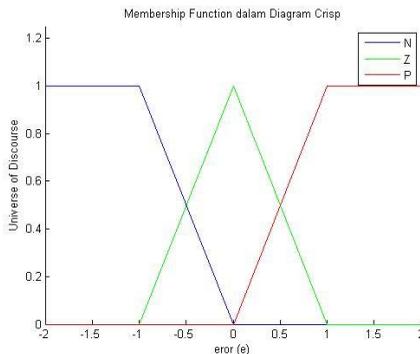
**Gambar 2.9** Diagram blok *three-element control*

## 2.5 Perancangan Logika Fuzzy

Fuzzy merupakan suatu tools atau metode yang menanamkan pengetahuan terstruktur manusia menjadi algoritma yang dapat diterapkan. Logika *fuzzy* merupakan suatu sistem pertimbangan logika (benar atau salah) yang ditujukan untuk menyediakan model yang direpresentasikan dalam suatu nilai kisaran dan bukan nilai pasti. Metode fuzzy banyak diaplikasikan untuk sistem yang rumit, misalkan seperti pengendalian sistem dengan orde tinggi atau sistem non-linier yang rumit untuk dianalisa secara matematis. Perancangan metode fuzzy dapat dilakukan melalui beberapa prosedur.

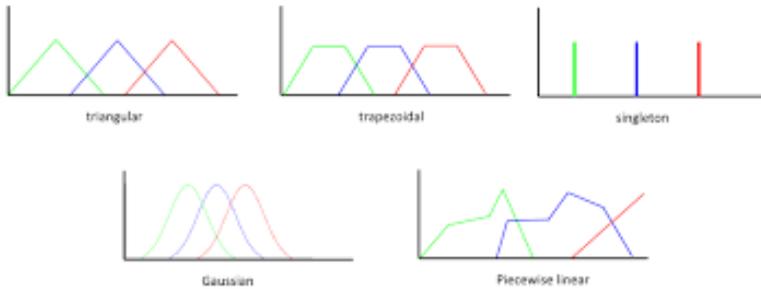
### 2.5.1 Fuzzification

Konsep dasar dari pertimbangan pada logika *fuzzy* adalah dengan menentukan fungsi keanggotaan seperti contoh pada *Gambar 2.10* melalui proses *fuzzifikasi* yang merupakan suatu proses pemetaan dari suatu pengamatan input. Fungsi keanggotaan menyatakan derajat atau tingkat keanggotaan masing-masing kelas atau *sets*. Umumnya penentuan fungsi keanggotaan ini direpresentasikan dalam nilai *crisp* sederhana. Umumnya bentuk atau pola nilai *crisp* dari *membership function* berupa fungsi segitiga, trapesium, *gaussian*, *singleton*, dan lain sebagainya.



**Gambar 2.10** Contoh fungsi keanggotaan segitiga dalam *crisp*

Selain fungsi segitiga, juga terdapat fungsi-fungsi yang lainnya seperti fungsi trapesium, *gaussian*, *singleton*, dan lain sebagainya yang digambarkan pada *Gambar 2.11*. Masing-masing fungsi tersebut memiliki karakteristik, kelebihan, serta kekurangannya tersendiri. Misalkan saja pada fungsi segitiga umum digunakan karena lebih mudah dipahami dan diterapkan, trapesium umumnya dianggap sebagai yang paling cocok diterapkan untuk fungsi tujuan ketahanan (*robustness*), *singleton* yang lebih sederhana diterapkan untuk proses defuzzifikasi, dan lain sebagainya.



**Gambar 2.11** Jenis-jenis fungsi keanggotaan dalam diagram *crisp*

### 2.5.2 Fuzzy Inference Rule dan Fuzzy Knowledge Base

Pada umumnya suatu *fuzzy rule* yang biasa dinyatakan dalam pola 'IF-THEN' biasa juga disebut dengan *fuzzy implication*. *Inference rule* dalam *fuzzy* menyatakan suatu hubungan atau relasi dalam suatu *fuzzy knowledge base*. Proses *reasoning* di dalam *fuzzy* utamanya menggunakan dua jenis *fuzzy inference rules*, diantaranya adalah *Generalized Modus Ponens* (GMP) juga sering disebut sebagai *direct reasoning* serta *Generalized Modus Tollens* yang sering disebut sebagai *indirect reasoning*. Lebih jelasnya dapat dilihat perbedaan antara kedua modus tersebut melalui Tabel 2.1. Kemudian sejumlah dari *fuzzy rule* tersebut dapat disebut dengan *fuzzy knowledge base* yang menggambarkan dasar pengetahuan atau pengalaman yang didapat untuk diimplementasikan ke dalam perancangan *fuzzy*.

**Tabel 2.1** Tipe inferensi secara umum pada *fuzzy*

	<i>Modus Ponens</i>	<i>Modus Tollens</i>
<b>Fact or Premise</b>	$x$ adalah $A$	$y$ bukan $B$
<b>Implication</b>	Jika $x$ adalah $A$ , maka $y$ adalah $B$	Jika $x$ adalah $A$ , maka $y$ adalah $B$
<b>Consequence of Conclusion</b>	$y$ adalah $B$	$x$ bukan $A$

### 2.5.3 Defuzzification

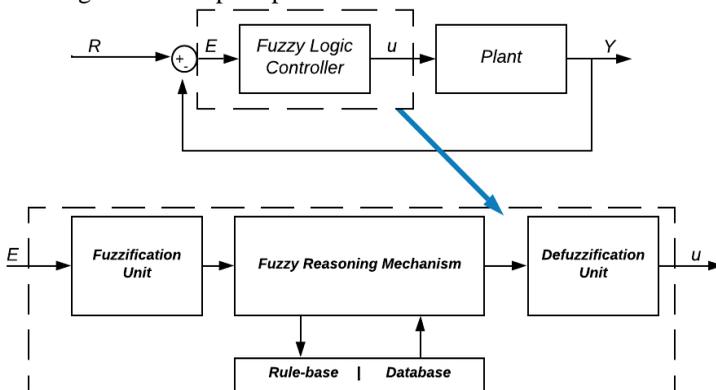
Prosedur untuk memperoleh nilai output dari logika *crisp* yang telah ditentukan dengan *membership function* tersebut disebut dengan proses defuzzifikasi. Hasil dari pengolahan *fuzzy sets* tersebut merepresentasikan output dari proses defuzzifikasi. Seperti halnya pada tahap-tahap sebelumnya dalam proses ini juga dapat dilakukan melalui

beberapa metode yang diantaranya adalah *center of area (COA)*, *first of maxima (FOM)*, *last of maxima (LOM)*, *mean of maxima (MOM)*, dan lain sebagainya.

Fungsi keanggotaan untuk proses defuzzifikasi pada *Mamdani* dapat berupa diagram *crisp* segitiga seperti pada proses fuzzifikasi sebelumnya maupun bentuk fungsi keanggotaan lainnya. Salah satu yang juga paling sering digunakan adalah tiap satu fungsi keanggotaan dalam bentuk *singleton* yang dimana nilai derajat keanggotaan hanya terdapat pada satu titik tertentu saja.

## 2.6 Sistem Kendali dengan Logika Fuzzy

Kontroler logika fuzzy pada umumnya dapat diterapkan pada pengaturan sistem *closed-loop*. Struktur utama pada kontroler logika fuzzy diantaranya adalah *fuzzification unit*, *fuzzy logic reasoning unit*, *knowledge base*, serta *defuzzification unit*. Pada *fuzzy knowledge base* memiliki dua tipe informasi yang utama, diantaranya adalah *database* dan *rule base*. *Database* merepresentasikan fungsi keanggotaan dari *fuzzy sets* yang diterapkan dalam bentuk derajat bilangan atau nilai, sedangkan *rule base* pada dasarnya memetakan nilai-nilai dari input menjadi suatu nilai output. Untuk memetakan nilai-nilai input dan output tersebut maka diperlukan suatu proses fuzzifikasi dan defuzzifikasi untuk menyatakan kedalam nilai-nilai fuzzy yang diterapkan didalamnya. Kesemua proses tersebut direpresentasikan dalam diagram blok seperti pada *Gambar 2.12*.



**Gambar 2.12** Rancangan pengendalian logika fuzzy

### 2.6.1 Kontroler Logika Fuzzy Tipe-PID

Skema pengendalian dengan logika *fuzzy* dapat dirancang dengan banyak cara, salah satu yang umum digunakan adalah kontroler logika *fuzzy* dengan dua input yang diantaranya adalah sinyal error dan sinyal delta-eror seperti tampak pada *Gambar 2.13*. Sinyal output dari pengendalian logika *fuzzy* dengan dua input ini akan merepresentasikan suatu sinyal kontrol dari kontroler logika *fuzzy* dengan tipe-PD tersebut, dimana pengaruh derivatif tersebut berasal dari input sinyal delta-eror tersebut.



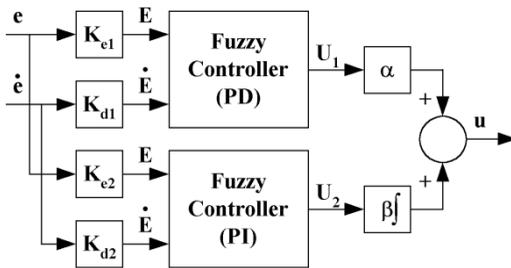
**Gambar 2.13** Kontroler logika *fuzzy* tipe PD

Sedangkan untuk kontroler logika *fuzzy* dengan tipe PI yang digambarkan pada *Gambar 2.14*, selain dengan menggunakan sinyal input untuk error dan integral error, juga dapat dinyatakan dengan rancangan sinyal input error dan delta-eror dengan adanya *incremental* sinyal output. Atau lebih mudahnya dapat direpresentasikan dengan skema kontroler logika *fuzzy* tipe PD yang dimana output atau keluaran sinyal kontrolnya akan diakumulasikan nilainya atau dengan kata lain diberikan sistem integrator.



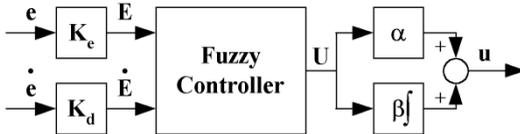
**Gambar 2.14** Kontroler logika *fuzzy* tipe PI

Untuk merancang kotroler logika *fuzzy* tipe PID dapat dilakukan dengan mengambil nilai dari keduanya berdasarkan skema kontroler logika *fuzzy* tipe PI dan PD, sehingga didapatkan skema dengan jumlahan dari kedua nilai tersebut dinyatakan seperti pada *Gambar 2.15*. Dengan demikian pengaruh integral dan derivatifnya dapat diatur dengan merubah masing-masing bobot dari  $\alpha$  untuk derivatif serta  $\beta$  untuk nilai integralnya. Karena skema antar keduanya mirip, maka dapat juga dirancang dengan satu buah skema *fuzzy-sets* yang sama seperti pada *Gambar 2.16*.



**Gambar 2.15** Kontroler logika fuzzy tipe PID

Sebagai perbandingan antara kedua metode tersebut, kontroler logika fuzzy tipe PI lebih umum digunakan daripada kontroler logika fuzzy dengan tipe PD. Kontroler logika fuzzy tipe PD masih cukup sulit untuk menghilangkan error *steady-state* dibandingkan dengan kontroler logika fuzzy tipe PI yang mampu menghasilkan sistem dengan memiliki error *steady-state* sebesar nol. Namun dari segi performa, pada respons *transient* sistem dengan kontroler logika fuzzy tipe PI lebih tidak diunggulkan daripada kontroler logika fuzzy dengan tipe PD terutama untuk pengendalian sistem yang memiliki orde tinggi.



**Gambar 2.16** Fuzzy PID dengan satu skema fuzzy sets

Karena itu untuk mendapatkan pengaruh proporsional, integral, dan derivatif pada kontroler logika fuzzy dengan dua input secara bersamaan dalam satu skema pengendalian maka dapat diterapkan skema kombinasi antara keduanya. Kombinasi dengan mengakumulasikan antara kedua sinyal kontrol dari pengendali logika fuzzy tipe PI dan PD dapat dirancang baik dengan dua *rule-base* yang berbeda ataupun dengan satu buah *rule-base* yang sama.

### 2.6.2 Kontroler Logika Fuzzy Adaptif

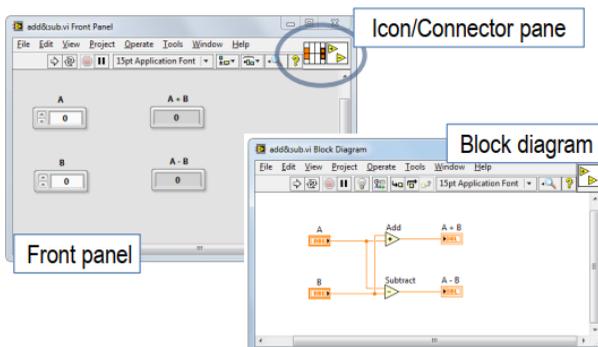
Kontroler logika fuzzy dengan skema adaptif saat ini banyak dibutuhkan untuk berbagai pengendalian pada suatu sistem. Karena itu parameter awal dari pengendali logika fuzzy yang meliputi *fuzzy sets*, *membership function*, *universe of discourse*, serta juga *control rule* perlu

untuk diubah menyesuaikan dengan perubahan karakteristik dari sistem yang dikendalikan tersebut. Pengendalian dengan parameter kontroler yang berubah menyesuaikan lingkungannya ini biasa disebut dengan kontroler logika *fuzzy* adaptif.

Skema adaptasi dalam kontroler logika *fuzzy* juga dapat bermacam-macam mengingat banyaknya pula parameter yang terdapat pada pengendali logika *fuzzy* tersebut. Sehingga proses adaptasi kontroler dapat dilakukan dengan beberapa skema seperti dengan menghasilkan *fuzzy-rule* baru, merubah *fuzzy-rule* yang sudah dirancang, merubah skema *fuzzy-sets* yang telah dirancang, menyesuaikan fungsi keanggotaannya, menyesuaikan *universe of discourse* atau nilai derajat keanggotaan yang dihasilkan, serta dapat juga dengan menyesuaikan resolusi dari sinyal kontrol atau *scalling factors*.

## 2.7 Pemrograman dengan LabVIEW

*Laboratory Visual Instrument Engineering Workbench* atau biasa disebut dengan *LabVIEW* merupakan suatu perangkat lunak system engineering atau rekayasa sistem untuk penerapan yang memerlukan pengujian, pengukuran, dan pengaturan dengan kecepatan tinggi ke hardware dan data yang luas. Dalam arti lain *LabVIEW* merupakan suatu perangkat pemrograman yang berbasis diagram atau gambar serta juga menyediakan interface tampilan (HMI) yang dikembangkan oleh National Instruments. Dalam penerapannya program *LabVIEW* dirancang di dalam suatu elemen yang disebut *Virtual Instrument (VI)*. Sedangkan di dalam VI itu sendiri seperti tampak pada *Gambar 2.17* terdiri dari 3 bagian utama, diantaranya adalah block diagram, front panel, serta icon.



**Gambar 2.17** Bagian virtual instrument (VI) pada LabVIEW

### **2.7.1 Bagian Utama LabVIEW**

*Block diagram* merupakan tempat dimana *programmer* meletakkan program yang akan dirancangnya. Seperti yang telah dijelaskan sebelumnya, program LabVIEW berbasis gambar sehingga *block diagram* tersebut akan menjadi sekumpulan diagram blok yang merepresentasikan suatu program. Oleh karena itu, saat melakukan pemrograman pada LabVIEW akan dilakukan pemanggilan fungsi dan juga wiring. Sedangkan *front panel* digunakan sebagai tampilan dari kontrol, indikator dan lain sebagainya untuk user. Dengan kata lain front panel memuat informasi visual yang dapat menjadi perantara (*interface*) penghubung antara mesin dengan manusia (*users*). Kemudian *icon* merepresentasikan VI secara visual, dan memiliki *connectors* untuk input/output dari program, yang berperan dalam beberapa hal seperti ketika menggunakan SubVI.

### **2.7.2 Shared Variable Engine**

Menggunakan variabel yang dibagikan, Anda dapat berbagi data antar *loop* pada suatu diagram atau antara VIs pada suatu jaringan komunikasi. Berbagai metode berbagi data yang berbeda terdapat pada LabVIEW, seperti UDP / TCP, LabVIEW *queues*, dan FIFO *Real-Time*, Konfigurasi untuk berbagi data dapat dilakukan ketika edit program dengan memilih *properties* dari objek tertentu yang akan digunakan untuk berbagi data.

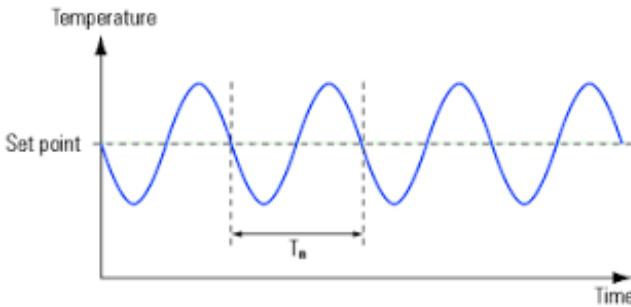
*Shared variabel* dapat dibuat dengan dua tipe yang berbeda, diantaranya adalah *single-process* dan *Networks-published shared variable*. Untuk membuat *shared variable*, dapat dilakukan pada window *project*, didalam diagram pohon (*project tree*) tersebut dapat disisipkan *shared variabel* baru sesuai dengan tipe yang dibutuhkan. Umumnya untuk komunikasi VIs antar komputer melalui jaringan *ethernet* dapat dilakukan dengan menggunakan tipe *Networks-published shared variable*.

### **2.8 Tuning PID dengan Metode Ziegler-Nichols Closed-loop**

Agar sistem dapat beroperasi sesuai dengan spesifikasi yang diinginkan, maka perlu juga untuk menentukan parameter kontroler PID (proporsional, integral, dan derivatif) yang digunakan untuk mengendalikan sistem tersebut. Proses untuk mencari parameter tersebut biasa disebut dengan proses *tuning* kontroler, yang umumnya dapat dilakukan baik dengan metode analitik atau dengan berdasarkan pada

eksperimen seperti halnya pada metode tuning *ziegler-nichols closed-loop* ini.

Metode *tuning ziegler-nichols closed-loop* ini dilakukan mulanya dengan merancang suatu sistem *closed-loop* seperti pada *Gambar 2.18* dengan *gain* yang akan menjadi penentu untuk parameter *tuning* PID dalam metode ini. Nilai *gain* terkecil agar sistem berosilasi secara permanen itulah yang harus dicari nilainya, yang merepresentasikan suatu *gain* yang menyebabkan sistem *closed-loop* tersebut memiliki pole yang terletak di sekitar nol.



**Gambar 2.18** Respons sistem *closed-loop* dengan *gain-marginal*

Setelah ditentukan nilai *gain marginal* yang merupakan *gain* terkecil hingga sistem berosilasi secara permanen dan tidak membesar, kemudian dicari periode osilasinya. Sehingga dari nilai *gain marginal* ( $K_c$ ) dan periode osilasi ( $T_c$ ) dapat diestimasi parameter PID yang optimal berdasarkan *Tabel 2.2*.

**Tabel 2.2** *Tabel tuning ziegler-nichols metode closed-loop*

Jenis Kontroler	Proportional (P)	Integral-time (Ti)	Derivative-time (Td)
<b>P</b>	$0,5 \cdot K_c$	-	-
<b>PI</b>	$0,45 \cdot K_c$	$T_c / 1,2$	-
<b>PID</b>	$0,6 \cdot K_c$	$0,5 \cdot T_c$	$0,25 \cdot T_c$



**BAB III**  
**SIMULASI PENGENDALIAN LEVEL AIR**  
**DI DALAM *STEAM DRUM* PADA PLTU**

### BAB 3

## SIMULASI PENGENDALIAN LEVEL AIR DI DALAM STEAM DRUM PADA PLTU

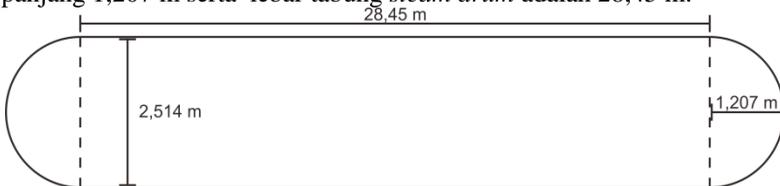
Simulasi pengendalian dalam sistem PLTU yang dirancang dalam penelitian ini merepresentasikan *Human Machine Interface* (HMI) seperti yang biasanya tampak pada *Control Room* sistem PLTU, dengan fokus pada *plant steam drum* sistem boiler. Beberapa HMI akan terhubung dengan jaringan komunikasi *ethernet*.

### 3.1 Perancangan Simulasi Sistem Pengendalian Level Steam Drum

Sistem pengendalian yang digunakan didasarkan pada model pengendalian yang biasa digunakan PLTU pada umumnya, yaitu *singlet element control* dan *three element control*. Pada penelitian ini, ditetapkan sistem tersebut dengan *plant steam drum* serta aktuator berupa *boiler feedwater pump – turbine* (BFPT).

#### 3.1.1 Permodelan Steam Drum

Model dari *steam drum* berdasarkan hasil permodelan secara hukum fisis didapatkan berdasarkan data model dari PT. PJB UBJ O&M Unit 9 Paiton seperti pada *Gambar 3.1*. Dengan didapatkan data model *steam drum* tersebut diperoleh nilai parameter *rating*-nya ketika saat beban operasi broiler adalah jari-jari bola *steam drum* ( $r$ ) dengan panjang 1,207 m serta lebar tabung *steam drum* adalah 28,45 m.



**Gambar 3.1** Desain model fisik *steam drum*

Sehingga berdasarkan penurunan yang didapat pada persamaan (2.1.7) didapatkan fungsi transfer model *steam drum* tersebut dalam domain frekuensi 's' (*Laplace*) dinyatakan seperti pada persamaan (3.1.2).

$$s \cdot H(s) = \frac{M_{fw}(s) - M_{st}(s)}{A_{drum} \cdot (\rho_{feedwater} + \rho_{steam})} \dots \dots \dots (3.1.1)$$

$$\frac{H(s)}{M_{fw}(s) - M_{steam}(s)} = \frac{1}{A \cdot (\rho_{fw} + \rho_{st}) \cdot s} \dots \dots \dots (3.1.2)$$

Kemudian fungsi luas permukaan air (A) pada *steam drum* berdasarkan persamaan (2.1.9) dengan dimasukkan parameter *steam drum* yang didapat dari PT. PJB UBJ O&M Paiton Unit 9 didapatkan suatu persamaan seperti pada persamaan (3.1.4).

$$A_{drum}(t) = A_{bota} + A_{tabung} \dots\dots\dots (3.1.3)$$

$$A_{drum}(t) = (\pi(2rh(t) - h(t)^2)) + (2l\sqrt{2\pi rh(t) - h(t)^2}) \dots\dots (3.1.4)$$

Sehingga dari nilai parameter konstan tersebut didapatkan nilai konstan dari persamaan luas permukaan air di dalam *steam drum* ketika operasi, dapat dinyatakan seperti pada persamaan (3.1.5).

$$A_{drum}(t) = (7,5838h(t) - 3,1416h(t)^2) + (56,9\sqrt{7,5838h(t) - 3,1416h(t)^2}) \dots\dots\dots (3.1.5)$$

Karena level air dijaga konstan pada kisaran area titik tengah dari *steam drum*, maka dapat dikatakan *rating* tinggi dari level air pada *steam drum* adalah 1,207 m, sehingga luas permukaan didapatkan dengan pendekatan seperti persamaan (3.1.6) luas permukaan *steam drum* adalah sebesar 126,3059 m<sup>2</sup>.

$$A_{drum}(t) = (7,5838(1,207) - 3,1416(1,207)^2) + (56,9\sqrt{7,5838(1,207) - 3,1416(1,207)^2}) \dots\dots\dots (3.1.6)$$

Sehingga didapatkan fungsi alih dari *steam drum* adalah seperti tampak pada persamaan (3.1.7).

$$\frac{H(s)}{M_{fw}(s) - M_{st}(s)} = \frac{1}{126,3059 \cdot (\rho_{fw} + \rho_{st}) \cdot s} \dots\dots\dots (3.1.7)$$

Kemudian model tersebut ditransformasikan kedalam domain z dengan menggunakan transformasi *bilinier* standar *tustin* yang dinyatakan dalam persamaan (3.1.8)

$$s = \frac{2(1-z^{-1})}{T_s(1+z^{-1})} \dots\dots\dots (3.1.8)$$

Sehingga didapatkan model *steam drum* dalam bentuk diskrit dengan time sampling (Ts) adalah 0,1 detik adalah seperti pada persamaan (3.1.12).

$$\frac{H(z) \cdot (\rho_{feedwater} + \rho_{steam})}{M_{feedwater}(z) - M_{steam}(z)} = \frac{1}{126,3059 \cdot \left( \frac{2(1-z^{-1})}{T_s(1+z^{-1})} \right)} \dots\dots\dots (3.1.9)$$

$$\frac{H(z) \cdot (\rho_{feedwater} + \rho_{steam})}{M_{feedwater}(z) - M_{steam}(z)} = \frac{T_s \cdot (1+z^{-1})}{2 \cdot 126,3059 \cdot (1-z^{-1})} \dots\dots\dots (3.1.10)$$

$$\frac{H(z) \cdot (\rho_{feedwater} + \rho_{steam})}{M_{feedwater}(z) - M_{steam}(z)} = \frac{0,1 \cdot (1+z^{-1})}{252,6118 \cdot (1-z^{-1})} \dots\dots\dots (3.1.11)$$

$$\frac{H(z) \cdot (\rho_{feedwater} + \rho_{steam})}{M_{feedwater}(z) - M_{steam}(z)} = \frac{0,000396 + 0,000396z^{-1}}{1-z^{-1}} \dots\dots\dots (3.1.12)$$

Dimana nilai dari massa jenis air ( $\rho_{feedwater}$ ) dan massa jenis uap ( $\rho_{steam}$ ) dipengaruhi juga oleh tekanan dan suhu fluida tersebut. Ketika sistem boiler beroperasi nilai tekanan yang dihasilkan di dalam steam drum adalah sekitar +170 bar serta suhu yang dihasilkan adalah sekitar  $\pm 540^\circ\text{C}$ , hal tersebut berdasarkan pengamatan di PT. PJB UBJ O&M Paiton Unit 9.

Dari nilai suhu dan tekanan tersebut, didapat konversi nilai massa jenis air dan uap saat operasi adalah  $\rho_{feedwater} = 572,087977 \text{ kg/m}^3$  dan  $\rho_{steam} = 51.014215 \text{ kg/m}^3$ . Kemudian karena satuan dari feedwater flow dan steam flow adalah  $\text{ton/h}$ , maka harus dirubah ke dalam satuan SI yaitu  $\text{kg/s}$  dengan nilai  $1 \text{ ton/h} = \frac{1000 \text{ kg}}{3600 \text{ s}} = \frac{5}{18} \text{ kg/s}$ , sedangkan pada output level, karena satuan yang digunakan adalah mili meter, maka dilakukan scaling dengan nilai 0,001. Sehingga didapatkan model dari steam drum adalah seperti persamaan (3.1.15).

$$\frac{H(z) \cdot \frac{5 \times 10^{-3}}{18} \cdot (572,09 + 51.01)}{M_{fw}(z) - M_{fw}(z)} = \frac{0,000396 + 0,000396z^{-1}}{1-z^{-1}} \dots\dots\dots (3.1.13)$$

$$\frac{H(z) \cdot \frac{5}{18} \cdot 623,102 \cdot 10^{-3}}{M_{fw}(z) - M_{st}(z)} = \frac{0,000396 + 0,000396z^{-1}}{1-z^{-1}} \dots\dots\dots (3.1.14)$$

$$\frac{H(z)}{M_{fw}(z) - M_{st}(z)} = \frac{(2,2879 + 2,2879z^{-1}) \cdot 10^{-3}}{1-z^{-1}} \dots\dots\dots (3.1.15)$$

### 3.1.2 Permodelan Feedwater Pump

Model dari BFPT didekatkan dengan model persamaan orde pertama dengan input adalah feedwater demand yang merupakan permintaan laju feedwater flow dan output adalah laju feedwater yang dihasilkan .

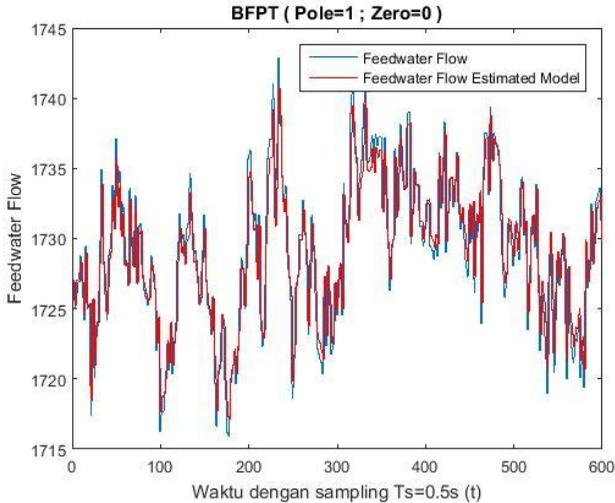
Data didapat dari PT. PJB UBJ O&M Paiton Unit 9 dengan 600 buah *sampel* data dengan waktu *sampling* adalah 0,5 detik. Dengan menggunakan metode regresi *least Square* diperoleh persamaan untuk mendapatkan parameter seperti pada persamaan (3.2.1).

$$\theta = \begin{bmatrix} 0,8625 \\ 0,1698 \end{bmatrix} \dots\dots\dots (3.2.1)$$

Sehingga didapatkan nilai, dengan model dalam domain waktu diskrit adalah seperti pada persamaan (3.2.2).

$$y(k) = 0,8625y(k - 1) + 0,1698u(k) \dots\dots\dots (3.2.2)$$

Model tersebut memiliki RMSE sebesar 2,5445 dengan grafik perbandingan antara sinyal asli dengan sinyal hasil regresi linier sesuai dengan *Gambar 3.2*.



**Gambar 3.2** Hasil regresi aktuator BFPT

Dalam domain frekuensi diskrit dengan di transformasikan ke dalam domain-Z (transformasi-Z) menggunakan *bilinear transformation* standar *tustin* didapatkan fungsi alih dengan ditetapkan waktu *sampling* adalah sebesar 0,1 detik menjadi seperti tampak pada persamaan (3.2.3).

$$\frac{Y(z)}{U(z)} = \frac{-0,0144.z + 0.0216}{0,98832.z - 0,98248} \dots\dots\dots (3.2.3)$$

### 3.1.3 Hubungan Beban dan Laju Aliran Uap

Pada simulator, dari data yang didapat dari PT. PJB UBJ O&M Paiton Unit 9, dengan sejumlah 600 data input beban dan output laju aliran uap yang memiliki waktu *sampling* sebesar 5,22 detik tidak memungkinkan untuk menghasilkan model data yang *valid* dengan menggunakan metode *least square* secara dinamis. Oleh karena itu, data tersebut akan dimodelkan dengan menggunakan persamaan linier garis lurus dengan nilai *bias* serta gradien atau kemiringan tertentu dengan menggunakan proses *regresi least square*.

Berdasarkan persamaan tersebut didapatkan nilai parameter *bias* adalah 406,586 serta parameter kemiringan atau gradien adalah sebesar 1,85. Sehingga persamaan hubungan antara beban dan laju aliran uap dapat dinyatakan dalam persamaan (3.3.1).

$$Steam(k) = 406,586 + 1,85 \cdot Load(k) \dots\dots\dots (3.3.1)$$

### 3.1.4 Permodelan Blowdown Rate dan Continous Blowdown

Berdasarkan dari referensi yang didapatkan didapatkan estimasi nilai untuk *rating* dari *flow blowdown* yang dihasilkan seperti yang dijelaskan pada persamaan (2.3.2), didapatkan nilai *blowdown* yang diperlukan sistem adalah sebesar 254,42t/h. Sedangkan untuk *continous blowdown water*, pada umumnya memiliki *rating* nilai mulai 0,5% sampai dengan 3% dari *steam flow* yang dihasilkan yaitu 2188 t/h. Sehingga rentang nilai untuk *continous blowdown water* adalah sekitar 109,4 - 656,4 t/h. Nilai tersebut diimplementasikan dalam Simulator.

### 3.1.5 Hubungan Steam Flow dengan Tekanan pada Steam Drum

Tekanan yang dihasilkan oleh uap didalam *steam drum* sebenarnya berbanding lurus linier dengan laju aliran uap yang dihasilkan oleh *boiler* tersebut. Oleh karena itu, nilai tekanan di dalam *steam drum* juga dapat dinyatakan dalam persamaan garis lurus dengan *bias* dan gradien tertentu sesuai dengan data yang di dapat dari PT. PJB UBJ O&M Paiton Unit 9, dalam persamaan (3.4.2).

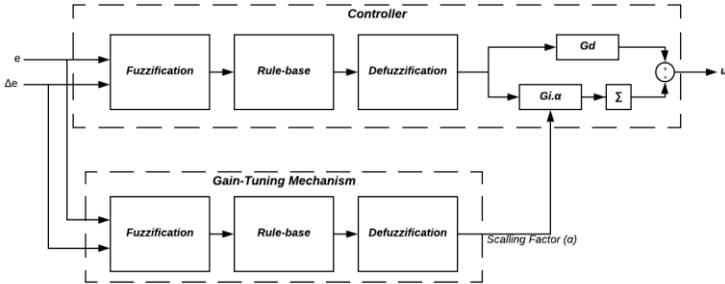
$$P_{steam\_drum} = \frac{178,47}{1790,87} \cdot M_{st} \dots\dots\dots (3.4.1)$$

$$P_{steam\_drum} = 0,0997 \cdot M_{st} \dots\dots\dots (3.4.2)$$

## 3.2 Perancangan Kontroler

Perancangan kontroler logika *fuzzy* tipe PID pada penelitian ini menggunakan skema *robust self-tunning*. Skema tersebut pada dasarnya terdiri dari dua komponen utama, diantaranya adalah *controller* yang

menghasilkan sinyal kontrol serta *gain tuning mechanism* sebagai adaptasi *gain* dalam bentuk *scaling factor* ( $\alpha$ ) yang direpresentasikan ke dalam diagram blok seperti pada *Gambar 3.3*.



**Gambar 3.3** Kontroler fuzzy tipe PID *robust-self tuning*

### 3.2.1 Proses Fuzzifikasi

Diagram *crisp* untuk proses fuzzifikasi pada penelitian ini dirancang dengan menggunakan fungsi keanggotaan segitiga dan *linear gamma function* atau setengah trapesium untuk masing-masing tepi sisi kanan dan kirinya. Rentang nilai fuzzifikasi akan ditetapkan mulai dari -3 sampai dengan 3.

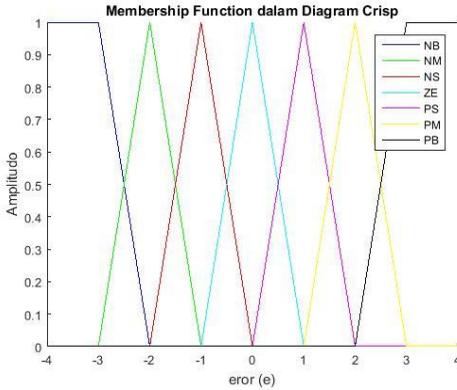
Sebelum dilakukan proses fuzzifikasi akan dilakukan proses normalisasi pada sinyal error dan delta-error sesuai dengan rentang nilai fuzzifikasi mulai dari -3 sampai dengan 3 tersebut. Pada *single-element control* proses normalisasi untuk sinyal error adalah berdasarkan rentang kerja dari level *steam drum*, yaitu -350mm hingga 350mm, sehingga didapatkan nilai penskalaan  $3/700$ . Sedangkan untuk sinyal delta-error ditetapkan nilai normalisasinya adalah  $3/10,0118504$  berdasarkan persamaan (3.5.4) model karakteristik dari *steam drum*.

$$y(k) = y(k - 1) + 2,2879 \cdot 10^{-3}u(k) + 2,2879 \cdot 10^{-3}u(k - 1) \quad (3.5.1)$$

$$y(k) - y(k - 1) = 2,2879 \cdot 10^{-3}u(k) + 2,2879 \cdot 10^{-3}u(k - 1) \quad (3.5.2)$$

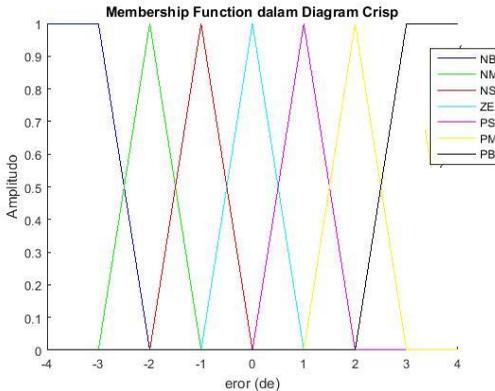
$$\text{delta - error} = 2,2879 \cdot 10^{-3}2188 + 2,2879 \cdot 10^{-3} \cdot 2188 \dots\dots\dots (3.5.3)$$

$$\text{delta - error} = 10,0118504 \dots\dots\dots (3.5.4)$$



**Gambar 3.4** Membership function sinyal error

Sedangkan pada sinyal delta error akan dinyatakan dalam diagram *crisp* dengan tujuh buah fungsi keanggotaan yang dinyatakan dalam notasi *Zero Equals (ZE)*, *Very Small (VS)*, *Small (S)*, *Small Big (SB)*, *Medium Big (MB)*, *Big (B)*, dan *Very Big (VB)* yang direpresentasikan ke dalam diagram *crisp* seperti pada *Gambar 3.5*.



**Gambar 3.5** Membership function sinyal delta-error

Begitu pula pada *Scaling Factor* atau *Gain-tuning Mechanism* untuk sinyal error dan delta error, proses fuzzifikasi akan didefinisikan dengan fungsi keanggotaan yang sama persis dengan proses fuzzifikasi pada *Controller*.

Kemudian masing-masing dari kedua *membership function* tersebut direpresentasikan dalam program Labview melalui *case structure* dengan memanfaatkan fungsi program *if ... else ...*, dengan *state* yang diberikan peran dalam mendefinisikan sinyal eror terhadap kondisi *membership function* serta *case* sebagai nilai *universe of discourse* yang ditentukan. Program tersebut dirancang pada *case structure* Labview pada *Lampiran 14*.

### 3.2.2 Penentuan Rule Base

Setelah proses fuzzifikasi kemudian ditentukan *rule-base* terhadap masing-masing fungsi keanggotaan dari *Controller* dan *Scalling Factor* atau *Gain-Tuning Mechanism* tersebut. Perancangan *rule-base* menyesuaikan dengan jumlah fungsi keanggotaan untuk proses fuzzifikasi dan defuzzifikasi untuk *Mamdani*.

Pada *Controller*, perancangan *rule* ditetapkan dengan tujuh buah *reference* yang diantaranya adalah *Very Negative (VN)*, *Medium Negative (MN)*, *Small Negative (SN)*, *Zero Equals (ZE)*, *Small Positive (SP)*, *Medium Positive (MP)*, serta *Very Positive (VP)*. Masing-masing hubungan fungsi keanggotaan dapat dinyatakan dalam tabel *Mack Vicar* pada *Tabel 3.1*.

**Tabel 3.1** *Rule-base* untuk *Controller*

$\Delta e$ $\backslash$ $e$	NB	NM	NS	ZE	PS	PM	PB
NB	VN	VN	VN	MN	SN	SN	ZE
NM	VN	MN	MN	MN	SN	ZE	SP
NS	VN	MN	SN	SN	ZE	SP	MP
ZE	VN	MN	SN	ZE	SP	MP	VP
PS	MN	SN	ZE	SP	SP	MP	VP
PM	SN	ZE	SP	MP	MP	MP	VP
PB	ZE	SP	SP	MP	VP	VP	VP

Sedangkan pada *Gain-tuning Mechanism* atau *Scalling Factor*, perancangan *rule* ditetapkan dengan tiga buah *reference* yang diantaranya adalah *Zero Equals (ZE)*, *Very Small (VS)*, *Small (S)*, *Small Big (SB)*, *Big (B)*, *Medium Big (MB)* serta *Very Big (VB)*. Masing-masing hubungan fungsi keanggotaan dapat dinyatakan dalam tabel *Mack Vicar* pada *Tabel 3.2*.

**Tabel 3.2** Rule-base untuk *Gain-tuning Mechanism*

$\Delta e \backslash e$	NB	NM	NS	ZE	PS	PM	PB
NB	VB	VB	VB	B	SB	S	ZE
NM	VB	VB	B	B	MB	S	VS
NS	VB	MB	B	VB	VS	S	VS
ZE	S	SB	MB	ZE	MB	SB	S
PS	VS	S	VS	VB	B	MB	VB
PM	VS	S	MB	B	B	VB	VB
PB	ZE	S	SB	B	VB	VB	VB

**3.2.3 Inferensi Mamdani**

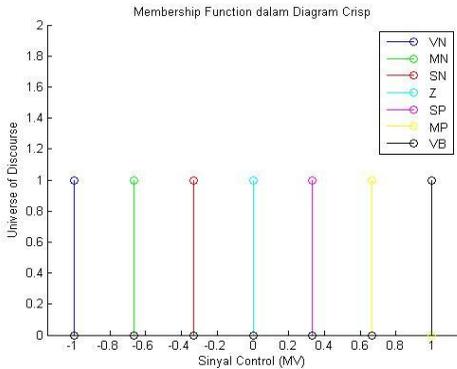
Setelah itu barulah dilakukan inferensi, dimana dalam hal ini dapat dilakukan dengan beberapa metode. Dalam pengaturan level ini, baik Controller maupun *Gain-tuning Mechanism* atau *Scaling Factor* metode inferensi yang digunakan keduanya adalah *minimum of maximum* (MOM), yang dinyatakan dalam persamaan (3.6.1).

$$u_x = \max \left( u_x, \min (mf_{1(i)}, mf_{2(j)}) \right) \dots\dots\dots (3.6.1)$$

Variabel  $u_x$  menyatakan hasil inferensi, mf1 dan mf2 merupakan nilai dari masing-masing fungsi keanggotaan untuk  $i = 1, 2, \dots$  (jumlah fungsi keanggotaan error) serta  $j = 1, 2, \dots$  (jumlah fungsi keanggotaan delta error).

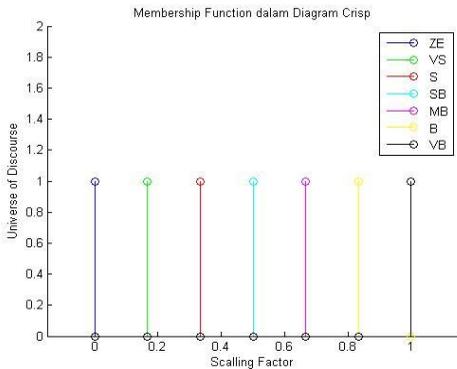
**3.2.4 Proses Defuzzifikasi**

Pada tahap ini nilai-nilai dari *reference* yang ditentukan pada *rule-base* akan dinyatakan dalam diagram *crisp*. Baik *Controller* maupun *Gain-tuning Mechanism* atau *Scaling Factor* keduanya menggunakan fungsi keanggotaan tipe *singleton* yang direpresentasikan ke dalam diagram *crisp* seperti pada *Gambar 3.6(a)* dan *Gambar 3.6(b)*.



**Gambar 3.6(a)** Defuzzifikasi *singleton* pada *Controller*

Pada *Controller* dirancang nilai sinyal kontrol  $u_x$  yang simetri, yaitu ditetapkan pada nilai  $\{-1, -2/3, -1/3, 0, 1/3, 2/3, 1\}$ . Sedangkan pada *Gain-tuning Mechanism* atau *Scaling Factor* dirancang nilai  $\alpha$  yang simetri pula pada nilai  $\{0, 1/6, 2/6, 3/6, 4/6, 5/6, 1\}$ , dengan rentang nilai minimal tidak nol dengan mempertimbangkan agar sinyal kontrol  $u_x$  dari kontroler masih dapat melakukan koreksi error dengan baik.



**Gambar 3.6(b)** Defuzzifikasi *singleton* pada *Gain-tuning Mechanism*

Terdapat beberapa metode untuk proses defuzzifikasi, salah satunya adalah dengan metode *Centre of Area (COA)*, yang dalam *singleton* dapat dinyatakan melalui persamaan (3.7.1).

$$x^* = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \dots\dots\dots (3.7.1)$$

dimana  $x^*$  adalah hasil atau nilai keluaran dari proses defuzzifikasi,  $n$  adalah banyaknya elemen sampel,  $x_i$  adalah elemen sampel serta  $\mu_i$  menyatakan nilai membership function.

### 3.2.5 Melakukan *Scalling* Ulang

Karena proses yang terdapat pada kontroler logika *fuzzy* merupakan nilai hasil dari normalisasi, maka output dari variabel  $x^*$  tersebut perlu diskalakan kembali sesuai dengan interval area kerjanya. Pada *single-element control* adalah 2188t/h sedangkan pada *Three-element control* adalah bernilai sama 2188t/h untuk *inner loop* serta 700mm pada *outer loop*. Kemudian dari beberapa proses tersebut, mulai dari *fuzzy rule*, inferensi, serta defuzzifikasi akan dirancang ke dalam *diagram block* pada program Labview.

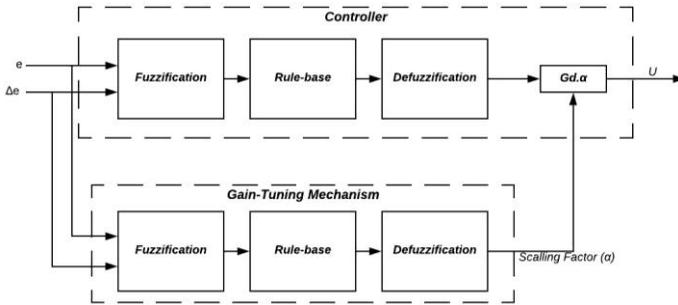
### 3.2.6 Penentuan Parameter Kontroler

Penentuan parameter untuk menyesuaikan daerah kerja dari sinyal eror dan delta-eror, dilakukan dengan meninjau berdasarkan acuan dari *rating* nilainya. Setelah diketahui *rating* nilainya tersebut kemudian akan dibandingkan dengan rentang nilai maksimal yang sudah didapatkan sebelumnya untuk menentukan nilai parameter untuk *gain error* ( $G_e$ ) dan *gain delta-error* ( $G_{de}$ ).

Untuk *single-element control*, penentuan  $G_e$  adalah  $700/40$  atau sama dengan 17,5. Sedangkan nilai untuk parameter  $G_{de}$  karena area kerja aktuator BFPT adalah nilai tengahnya yaitu 1094 maka nilai rentang perubahan delta-eror juga akan diskalakan menjadi dua kalinya, sehingga nilai parameter dari  $G_{de}$  adalah 2.

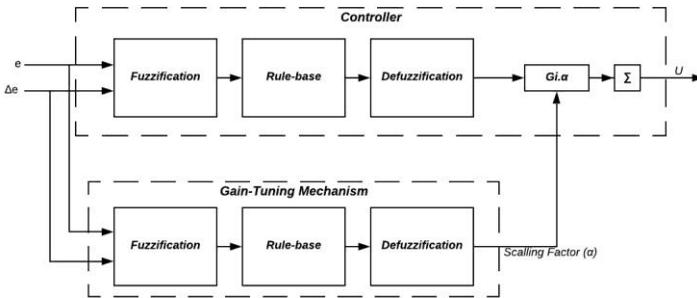
### 3.2.7 Skema Kontroler Logika *Fuzzy*

Pada penelitian ini, akan dipertimbangkan beberapa rancangan metode kontrol logika *fuzzy* yang berbeda namun dirancang dengan skema *fuzzy-sets* yang sama. Rancangan untuk kontroler logika *fuzzy* antara tipe PI yang direpresentasikan dalam *Gambar 3.7* dan PD yang direpresentasikan dalam *Gambar 3.8* sebenarnya cukup mirip.



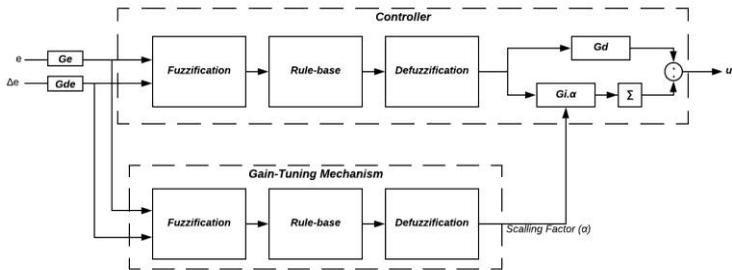
**Gambar 3.7** Rancangan konroler logika *fuzzy* tipe-PD

Kontroler logika *fuzzy* untuk tipe PI merepresentasikan kontroler logika *fuzzy* dengan tipe PD yang dimana sinyal kontrolnya diakumulasi (ditambahkan integrator) melalui persamaan  $u(k)=u(k-1)+\Delta u(k)$  dimana  $u(k)$  adalah nilai output sinyal kontrol dari *fuzzy* tipe-PI saat ini,  $u(k-1)$  adalah nilai output sinyal kontrol dari *fuzzy* tipe-PI pada iterasi sebelumnya (0,1 detik sebelumnya sesuai waktu *sampling*), serta  $\Delta u(k)$  adalah output sinyal kontrol dari *fuzzy* tipe-PD saat ini.



**Gambar 3.8** Rancangan konroler *fuzzy* tipe-PI

Kontroler logika *fuzzy* untuk tipe PID dirancang dengan mengakumulasi nilai sinyal kontrol dari kedua kontroler logika *fuzzy* tipe PI dan PD seperti skema pada *Gambar 2.16*. Namun untuk *gain* parameter kontroler logika *fuzzy* tipe PI akan diintegrasikan dengan *gain-tuning mechanism* dengan dasar untuk menghindari *wind-up* sistem akibat saturasi nilai pada aktuator BFPT.



**Gambar 3.9** Rancangan konroler *fuzzy* tipe-PID

### 3.3 Spesifikasi Sistem

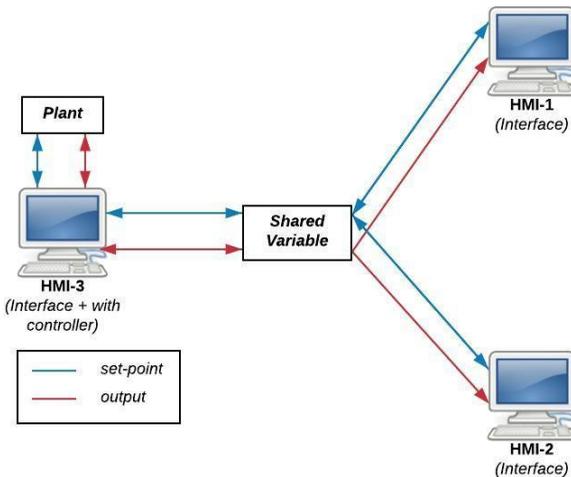
Spesifikasi sistem menggambarkan target atau hasil yang kita inginkan terhadap sistem tersebut. Sehingga perancangan kontroler yang dilakukan dimaksudkan agar sistem tersebut memenuhi kriteria respons yang sesuai dengan spesifikasi yang telah ditentukan. Pada sistem pengaturan level air didalam *steam drum* yang diamati ditetapkan kriteria sistem terhadap perubahan *set-point* (naik atau turun) tanpa beban serta kriteria respons sistem terhadap perubahan gangguan beban listrik yang terjadi mulai dari nilai nominal operasi 400MW hingga nilai maksimal 700MW.

Rentang level air agar sistem *boiler* tidak mengalami MFT (*trip*) adalah level air tidak lebih dari 254 mm dan tidak kurang dari -361 mm. Selain itu dari cepat tidaknya respons *transient* sistem, diharapkan respons sistem menjadi tidak lebih lambat dari nilai *settling-time* respons sistem ketika tanpa kontroler. Selain itu, dari respons *transient* sistem ditetapkan kriteria dengan *settling-time* tidak lebih dari 38,7 detik berdasarkan pada karakteristik respons *itransient* dari BFPT.

### 3.4 Perancangan Sistem Komunikasi antar HMI

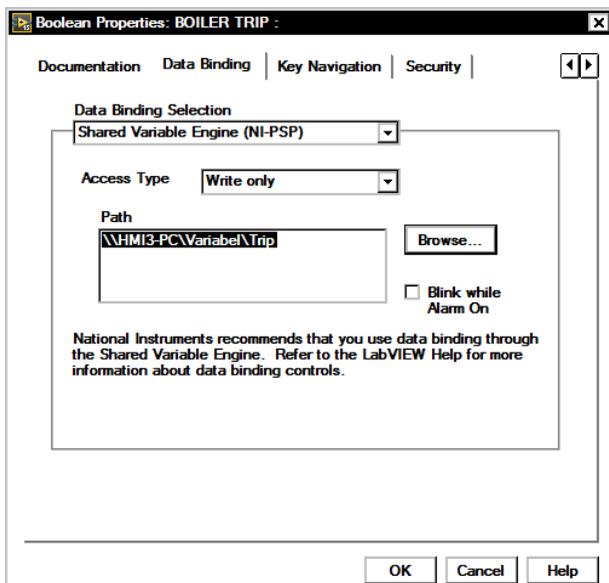
HMI dikenal sebagai penghubung atau *interface* yang menghubungkan komunikasi antara mesin dengan manusia. HMI yang dirancang pada penelitian ini didesain untuk meniru tampilan *interface local control* yang terdapat pada sistem PLTU, dengan fokus kontrol dan monitoring pada *plant steam drum* yang terdapat pada sistem *boiler*. HMI untuk simulasi ini dapat dijadikan sebagai bahan pembelajaran dalam bentuk gambaran tentang proses kerja sistem PLTU beserta pengendalian untuk *singlet element control* dan *three element control*.

Pada *Gambar 3.10* menjelaskan skema pertukaran data dengan menggunakan jaringan *ethernet* (TCP-IP) melalui *shared variable network stack* NI-PSP. Pertukaran data dilakukan secara *real-time* (FIFO) melalui *network-published variables* seperti pada *Gambar 3.11*. Pada desain simulator PLTU terdapat satu HMI utama (HMI-I) yang terhubung langsung dengan *plant* sehingga HMI-I ini berperan juga dalam mengolah sinyal *input-output* untuk pengendalian *singlet element control* dan *three element control*, dengan kata lain kontroler logika *fuzzy* tersebut dirancang pada HMI-I ini. Sedangkan untuk HMI-II dirancang dengan tampilan serta akses yang sama persis dengan HMI-I namun pada HMI-II ini tidak terdapat program atau sistem pengendalian melainkan hanya terhubung dengan memberi atau menerima nilai dari dan ke HMI-I. Dapat digunakan HMI-II lebih dari satu, sehingga dapat menyesuaikan kebutuhan.



**Gambar 3.10** Skema pertukaran data dengan *shared variabel engine*

Pada dasarnya kedua HMI tersebut merepresentasikan prinsip kerja dari struktur distribusi *Client-Server*, dengan HMI-I sebagai *client* dan HMI-II sebagai *server*. Jadi pada HMI-I baik *set-point* maupun outputnya adalah *read/write*, sedangkan pada HMI-II *set-point* adalah *read/write* sedangkan outputnya adalah *read*.



**Gambar 3.11** *Shared Variable Engine (SVE)* pada NI-PSP

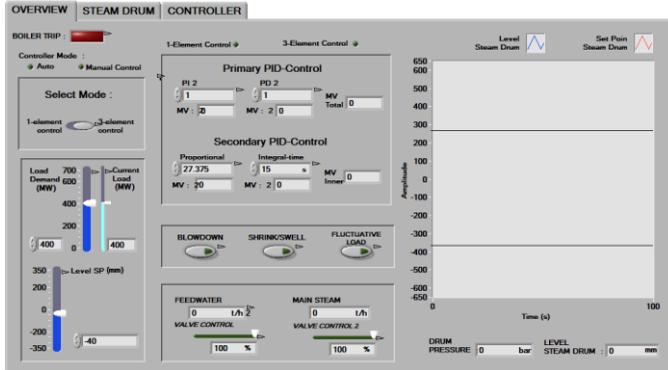
### 3.4 Perancangan HMI pada Simulator PLTU

Perancangan HMI pada penelitian ini didesain dengan mengacu pada *interface* yang diterapkan pada PT. PJB UBJ O&M Paiton. Terdapat tiga buah sub-menu tampilan, diantaranya adalah *Overview* yang menampilkan parameter sistem (input-output) yang diperlukan, *Steam Drum* yang merepresentasikan kondisi *steam drum* secara visual dengan parameter nilai (output) yang didapatkan dari pengukuran sistem, serta *Controller* yang menampilkan input parameter kontrol yang ditetapkan serta output hasil pengolahan sinyal kontrol logika *fuzzy* tersebut. Terdapat 3 menu tampilan utama untuk perancangan HMI pada penelitian ini. Menu-menu tampilan tersebut diantaranya adalah *Overview*, *Steam Drum*, dan *Controller*.

Pada tampilan *Overview* *Gambar 3.12* yang menampilkan informasi secara umum. Informasi tersebut sebagai input berupa mulai dari grafik respon sistem, parameter kontroler, *selector mode* kendali *Gambar 3.10*, *element control* atau *three element control*, permintaan beban, *set point level*, *percent valve* untuk *feedwater* dan *steam*, serta *blowdown*. Sedangkan pada output meliputi kondisi *percent valve*,

tekanan dan level air pada *steam drum*, indikator *boiler trip*, serta grafik respon sistem.

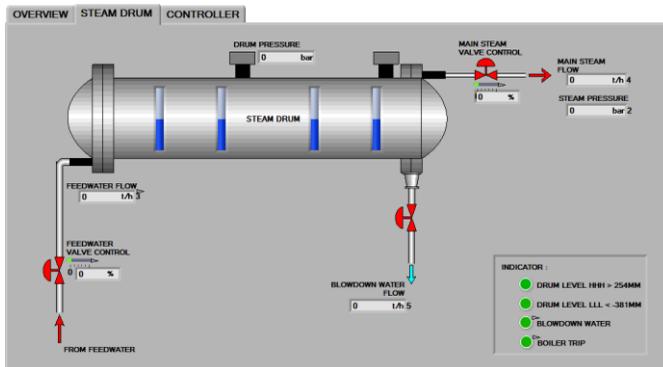
### BOILER CONTROL



**Gambar 3.12** Tampilan HMI pada menu *Overview*

Kemudian pada tampilan *Steam Drum* Gambar 3.13 menampilkan informasi terkait variabel yang terdapat secara khusus pada *steam drum* beserta gambar fisiknya dengan tujuan agar informasi lebih mudah untuk diterima.

### BOILER CONTROL

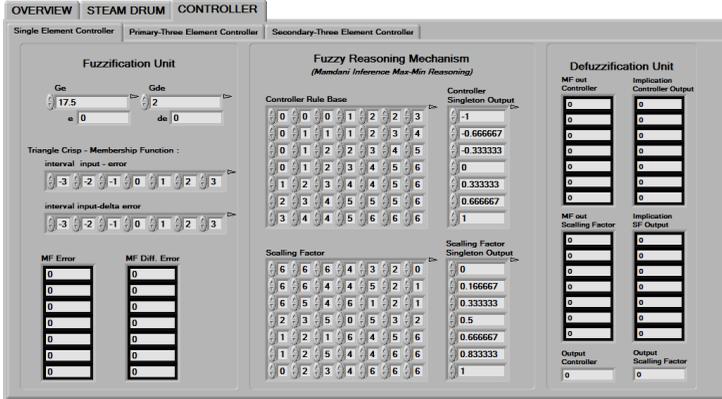


**Gambar 3.13** Tampilan HMI pada menu *Steam Drum*

Menu tampilan yang terakhir adalah *Controller* Gambar 3.14 yang menampilkan parameter kontroler logika *Fuzzy*. Parameter-parameter

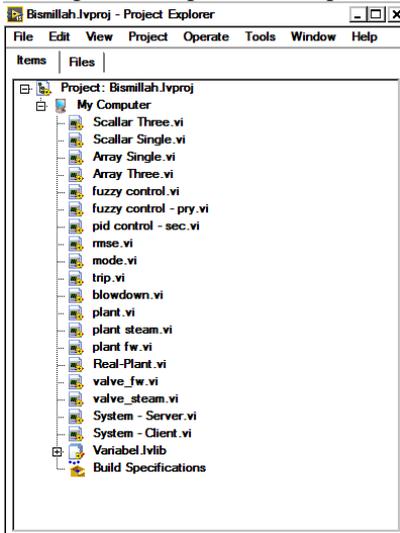
tersebut dapat diubah atau dituning kembali sesuai kebutuhan secara langsung oleh operator dan menyesuaikan dengan kondisi sistem.

### BOILER CONTROL



**Gambar 3.14** Tampilan HMI pada menu *Fuzzy-sets*

Kemudian untuk blok diagram sebagai program pada sistem Simulator PLTU dirancang dengan VI utama seperti pada *Lampiran 13*, dengan beberapa Sub-VI seperti tampak pada *Gambar 3.15*



**Gambar 3.15** Tampilan window *project* pada Simulator

Ketika boiler *trip*, program dirancang agar mode kontrol beralih ke *manual control* serta valve dari pipa *feedwater* dan *steam* menjadi tertutup (0%) untuk menghindari *offset* level yang terlalu besar. Sehingga untuk pengaturan *valve* tertutup ketika terjadi *trip* dirancang seperti pada *Gambar 3.16*, dengan rancangan yang sama pula untuk *valve* pada pipa *steam*.



**Gambar 3.16** Rancangan *valve* tertutup ketika *boiler trip* (MFT)

Selain itu, dirancang pula banya model program yang lain seperti tampak pada *Lampiran 13*, meliputi panel yang otomatis berpindah sesuai mode kontrol yang digunakan, sinkronisasi nilai antara P dan KP, Ti dan Ki, Td dan Kd, serta konfigurasi yang sesuai agar tiap *set-point* pada masing-masing HMI-1, HMI-2, dan HMI-3 bisa saling mengikuti perubahan nilai yang ditetapkan tanpa menimbulkan *crash* dari pertukaran data.

### 3.5 Indeks Performansi Kriteria dan Tuning Sistem

Performansi sistem merupakan suatu performa atau kinerja dari sistem tersebut yang dapat ditinjau mulai dari karakteristik respons transien hingga *steady state*-nya. Dalam penelitian ini, metode untuk menguji performa dari sistem pengaturan level *steam drum* digunakan dua metode yang diantaranya adalah metode *objective function* serta metode *phase plane trajectory*.

#### 3.5.1 Analisa Error pada Sistem dengan Kontroler

Untuk menghasilkan sistem dengan respons yang sebaik mungkin maka perlu untuk memperhitungkan parameter kontroler agar dapat berkerja secara optimal. Penyesuaian nilai parameter kontroler atau biasa juga disebut dengan proses *tuning* dapat ditinjau keoptimalannya melalui penggambaran suatu kriteria atau nilai, yang umumnya berupa indeks performansi sistem. Nilai dari indeks performansi dari sistem dapat diperoleh dari analisa error yang menggambarkan kriteria respons dari sistem. Jenis kriteria yang umum digunakan adalah seperti *Integral Absolute Error (IAE)* pada persamaan (3.8.1) serta *Integral Square Error (ISE)* pada persamaan (3.8.2).

*Integral Absolute Error* sesuai dengan namanya yaitu merupakan suatu kriteria sistem yang dinyatakan dalam jumlahan dari nilai absolut atau mutlak dari eror yang dihasilkan oleh sistem tersebut. Sedangkan pada *Integral Square Error* merupakan jumlahan dari kuadrat sinyal eror yang dihasilkan oleh sistem. Dari kedua karakteristik tersebut antara ISE dan IAE berdasarkan pengalaman penulis memiliki indeks performansi yang cukup berbeda, seperti untuk IAE umumnya lebih menekankan pada seberapa mampu sistem untuk mencapai titik *steady-state*, sedangkan pada ISE lebih menekankan pada seberapa besar simpangan atau *peak* yang dihasilkan oleh sistem terhadap *set-point* (*overshoot* dan *undershoot*).

Kemudian dari kedua kriteria yang dipertimbangkan tersebut akan dinyatakan dengan indeks performansi sistem keseluruhan atau total berdasarkan kedua karakteristik yang berbeda tersebut. Sesuai dengan *experience* terhadap sistem tersebut, dapat ditentukan bobot dari masing –masaing kriteria yang berbeda tersebut untuk dinyatakan dalam satu indeks performansi keseluruhan dari sistem. Persamaan yang digunakan untuk mengidentifikasi titik keoptimalan dari sistem tersebut biasa disebut dengan *objective function* yang umumnya dinotasikan dengan simbol *J* sesuai persamaan (3.8.3).

$$IAE = \int |e(t)| dt \dots\dots\dots (3.8.1)$$

$$ISE = \int e(t)^2 dt \dots\dots\dots (3.8.2)$$

$$J = w_1 \cdot IAE + w_2 \cdot ISE \dots\dots\dots (3.8.3)$$

Ditetapkan nilai bobot untuk IAE adalah  $w_1$  sebesar 0,9 serta nilai bobot untuk ISE adalah  $w_2$  sebesar 0,1 sehingga didapatkan kriteria *J* dengan persamaan seperti pada (3.8.4).

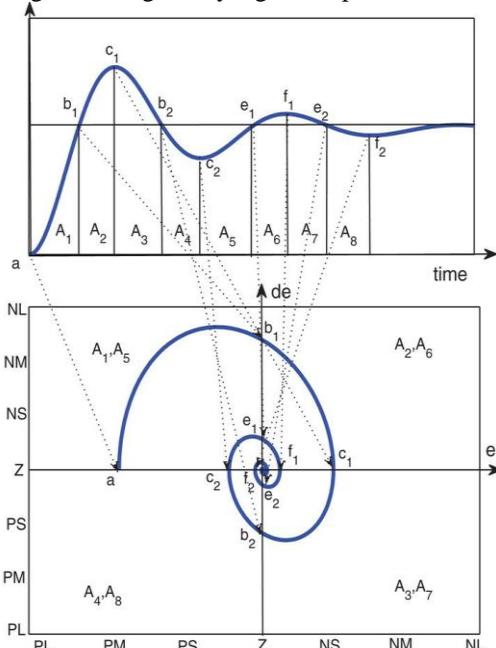
$$J = 0,9 \cdot IAE + 0,1 \cdot ISE \dots\dots\dots (3.8.4)$$

### **3.5.6 Phase Plane Trajectory**

Metode *phase plane trajectory* merupakan suatu teknik untuk merepresentasikan secara grafis pada karakteristik dinamik dari suatu sistem kontrol ke dalam bidang fase. Dalam analisa pengendali logika *fuzzy* tipe PID dimana memiliki skema dengan dua input eror serta delta eror yang nanti akan menjadi titik aksisnya. Sehingga kedua variabel tersebut dapat diambil sampelnya untuk dianalisa ke dalam *phase plane trajectory* seperti digambarkan pada *Gambar 3.14*.

Tujuan dari analisa dengan menggunakan metode *phase plane trajectory* ini adalah untuk mengamati bagaimana sinyal eror dan delta

eror ini dihasilkan dari *fuzzy sets* kontroler yang sudah ditentukan. Dari nilai eror dan delta eror yang digambarkan dalam *phase plane trajectory* akan dapat diperkirakan bagaimana proses *fuzzyfikasi* terjadi sehingga dapat disesuaikan bagaimana sinyal tersebut perlu dinormalisasi atau *scaling* ulang sehingga proses *fuzzyfikasi* dapat bekerja optimal sesuai dengan rentang nilai yang diharapkan.



**Gambar 3.14** Diagram *phase plane trajectory*



**BAB IV**  
**PENGUJIAN DAN ANALISIS**

## BAB 4

### PENGUJIAN DAN ANALISIS

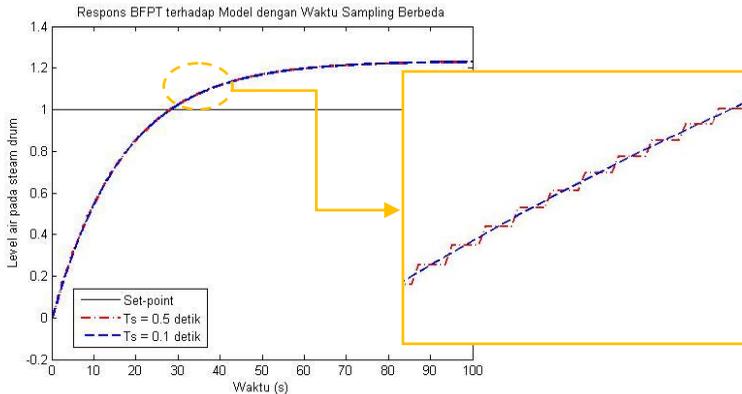
Pengujian dilakukan dengan mensimulasikan sistem simulator yang telah dirancang dengan skema untuk metode pengendalian level *steam drum* pada penelitian ini.

#### 4.1 Pengujian Komponen Sistem

Pengujian sistem dilakukan pada komponen aktuator BFPT serta juga *plant steam drum*. Pada pengujian aktuator BFPT, akan dilakukan pengujian secara *open-loop* dan juga pada pengujian *plant steam drum* dilakukan secara *open-loop*.

##### 4.1.1 Pengujian aktuator BFPT secara *Open-loop*

Pada aktuator BFPT yang merupakan model sistem dinamik orde 1, akan diuji secara *open-loop* dengan menggunakan sinyal *step*. Pertamanya, model sistem hasil transformasi bilinear standar *tustin* dengan waktu *sampling* 0,1 detik akan diuji kecocokannya dengan model aslinya dengan waktu *sampling* 0,5 detik. Hasilnya seperti tampak pada *Gambar 4.1*. Model hasil transformasi tersebut sudah sesuai dan dapat digunakan pada simulator yang ditetapkan memiliki waktu *sampling* 0,1 detik.



**Gambar 4.1** Respons model BFPT terhadap waktu sampling

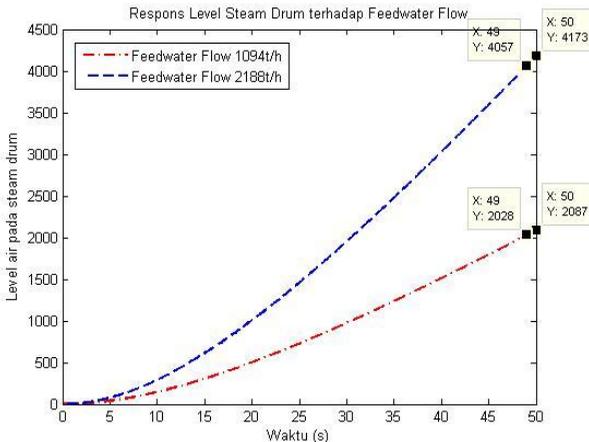
Selanjutnya berdasarkan respons tersebut akan didapatkan karakteristik respons dari BFPT mulai dari respons *transient* hingga respons *steady state*-nya. Berdasarkan hasil dari grafik respons yang

dihasilkan didapatkan beberapa kriteria sistem diantaranya adalah *time-constant* dengan nilai 16,9 detik dan *settling-time* dengan nilai 38,7 detik. Sistem tersebut memiliki *gain overall* sebesar 1,23.

#### 4.1.2 Pengujian *Plant Steam Drum*

Pengujian dilakukan secara *open-loop* dengan memberikan sinyal input laju aliran *feedwater* berupa *unit-step* dengan nilai yang berbeda, yaitu dengan input laju aliran sebesar 1094t/h dan 2188t/h. Sehingga dapat diamati dari *Gambar 4.2* bahwa dengan kedua perbedaan nilai input tersebut menghasilkan rentang perubahan level air pada *steam drum* yang berbeda tiap waktunya.

Berdasarkan grafik respons tersebut didapatkan untuk input laju aliran *feedwater* adalah 1094t/h didapatkan perubahan level air pada *steam drum* adalah sekitar 59 mm per detik. Sedangkan pada input laju aliran *feedwater* adalah 2188t/h memiliki laju perubahan level dua kali lebih cepat, yaitu sekitar 116 mm per detiknya.



**Gambar 4.2** Respons laju perubahan level air pada *Steam Drum*

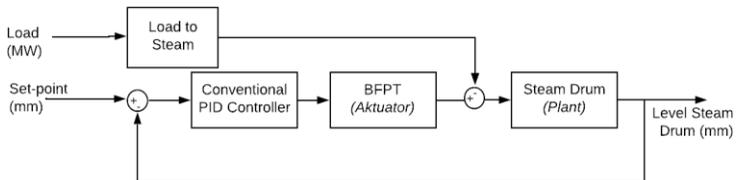
#### 4.2 Analisa Kontroler pada *Single-element Control*

Pengendalian level air didalam *steam drum* pada *single-element control* dilakukan dengan memberikan *set-point* untuk level air pada *steam drum* yang diinginkan adalah sebesar 40 mm sesuai dengan *rating* perubahan level yang terjadi, serta juga diuji dengan diberikan gangguan dengan ditetapkan *set-point* -40 mm. *Disturbance* yang diberikan berupa laju aliran uap yang nilainya bergantung pada beban atau daya listrik

yang ingin dihasilkan. Pola gangguan yang diterapkan adalah dengan nilai beban awal ditetapkan 400MW lalu kemudian selang 50 detik kemudian beban dinaikkan secara spontan menjadi 700MW. Pada pengendalian logika *fuzzy* untuk *single-element control* ini akan dilakukan beberapa pengujian, diantaranya adalah pengujian untuk pengendalian PID konvensional dengan dan tanpa gangguan untuk mengetahui respon sistem terhadap perubahan *set-point* serta ketahanan sistem terhadap gangguan.

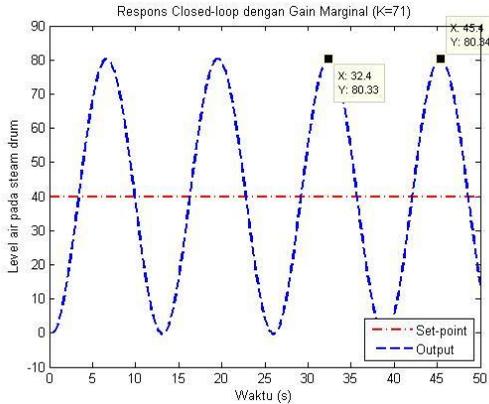
#### 4.2.1 Pengujian Sistem dengan Kontroler PID Konvensional

Mulanya sistem pengaturan level air pada *steam drum* diuji dengan menggunakan kontroler PID konvensional pada umumnya. Karena sistem merupakan tipe-1 sedangkan aktuator BFPT memiliki rentang kerja non-negatif antara 0 sampai 2188 t/h, maka proses tuning untuk metode *ziegler-nichols closed-loop* tidak dapat dilakukan karena selalu naik dan tidak dapat menghasilkan osilasi. Oleh karena itu proses *tuning* dilakukan dengan diberikan *disturbance* sistem seperti pada *Gambar 4.3* yang ditetapkan nilainya sesuai dengan *rating*-nya saat operasi, yaitu dengan beban 400 MW.



**Gambar 4.3** Sistem *closed-loop* dengan *disturbance*

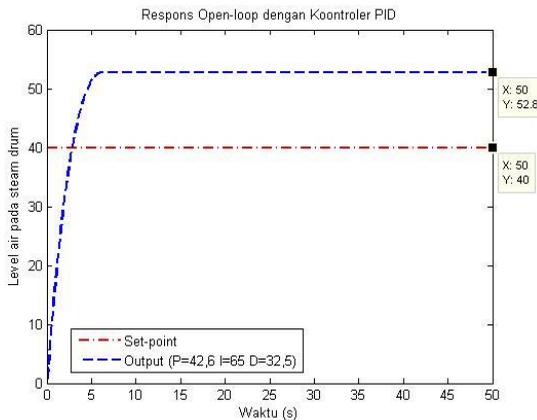
Dengan menggunakan metode seperti pada *Gambar 2.13* didapatkan *gain-marginal* dari sistem adalah sebesar 71 dengan respons seperti tampak pada *Gambar 4.4*, dan menghasilkan periode osilasi permanen yang dihasilkan adalah sekitar 13 detik. Sehingga didapatkan nilai  $K_c$  adalah 71 dan  $T_c$  adalah 13.



**Gambar 4.4** Respons *Closed-loop* Sistem dengan *gain marginal*

Melalui penghitungan berdasarkan pada *Tabel 2.1*, didapatkan parameter kontroler adalah  $P = 42,6$ ,  $\tau_i = 6,5$  detik, dan  $\tau_d = 3,25$  detik. Karena pengaturan yang dilakukan merupakan pengaturan digital dengan waktu *sampling* adalah 0,1 detik, maka didapatkan parameter kontroler dalam waktu diskrit adalah  $P = 42,6$ ,  $\tau_i = 65$ , dan  $\tau_d = 32,5$ .

Pengujian awal dilakukan dengan tanpa gangguan dengan diberikan sinyal *unit-step* yang memiliki amplitudo sebesar 40 mm sebagai *set-point*. Hasilnya didapatkan grafik respons seperti tampak pada *Gambar 4.5* adalah terjadi eror *steady-state* sebesar 12,8 mm.

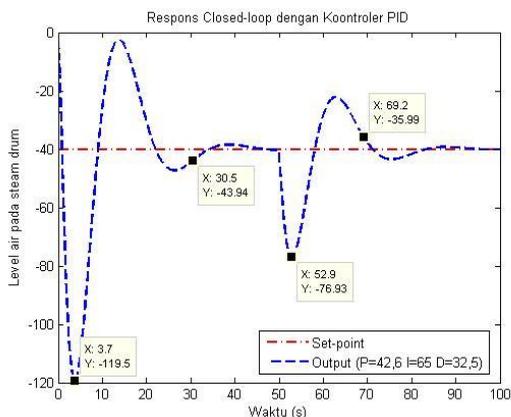


**Gambar 4.5** Respons sistem tanpa gangguan

Selanjutnya dilakukan pengujian dengan gangguan, sehingga didapatkan hasil respons sistem seperti tampak pada *Gambar 4.6*. Sistem tidak mampu memberikan respons yang baik terhadap *set-point* yang turun 40mm dengan beban nominal 400MW. *Undershoot* yang dihasilkan menjadi sangat besar akibat saturasi nilai output dari aktuator BFPT. Karena nilai sinyal kontrol integral akan terus mengakumulasi sinyal eror dengan besar sesuai dengan *gain* integratornya, jika cepat waktu respon terbatas (lambat) akibat saturasi maka sinyal kontrol akan semakin lebih besar. Sehingga nilai *undershoot* yang dihasilkan dari sistem dengan gangguan bergantung pada seberapa besar *wind-up* sistem yang terjadi. Lebih jelasnya dapat dilihat respons sistem dengan kriteria yang dihasilkan seperti pada *Tabel 4.1*.

**Tabel 4.1** Kriteria respons sistem dengan kontroler PID

Kriteria Sistem	Respons Sistem
<i>Perturbation Peak</i> (perubahan <i>set-point</i> dan beban nominal)	79,5 mm
<i>Perturbation Peak</i> (beban maksimal)	36,93 mm
<i>Settling-time</i> (10%) pada beban nominal	30,5 detik
<i>Settling-time</i> (10%) pada beban maksimal	19,2 detik
IAE	11.360,59
ISE	416.614,78
J	51.886,01



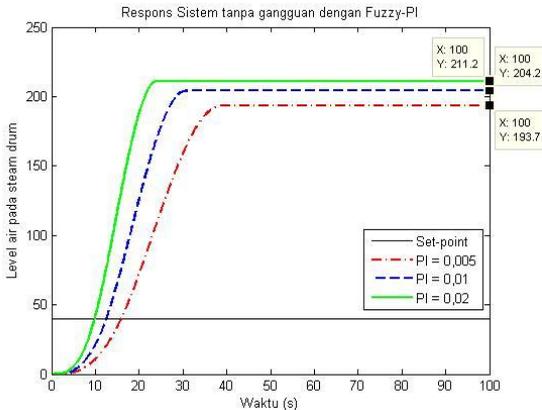
**Gambar 4.6** Respons input-output aktuator BFPT dengan gangguan

### 4.3 Pengujian Kontroler Logika Fuzzy

Metode pengendalian dapat dirancang dengan banyak metode dan parameter yang berbeda-beda, dimana masing-masing memiliki pengaruh terhadap sistem yang berbeda pula. Oleh karena itu perlu dilakukan suatu kajian karakteristik sistem terhadap pengaruh masing-masing metode dan parameter kontrol yang dirancang tersebut. Pengujian kontroler logika fuzzy dimaksudkan dengan menganalisa serta membandingkan respons sistem terhadap beberapa metode kontrol yang berbeda, diantaranya adalah metode kontrol fuzzy-PI, fuzzy-PD, fuzzy-PID, serta fuzzy-PID dengan skema *robust self-tuning* masing-masing terhadap perubahan *set-point* dan juga pengaruh *disturbance*.

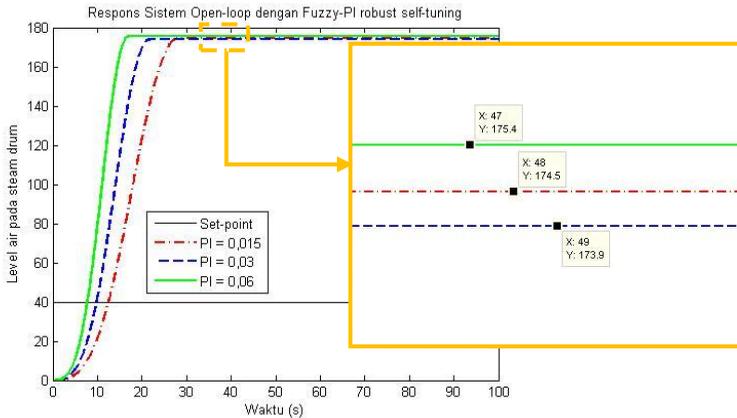
#### 4.3.1 Kontroler Logika Fuzzy Tipe-PI *Robust Self-tuning*

Untuk pengendalian logika fuzzy dengan tipe-PI, sistem diuji dengan skema seperti tampak pada Gambar 3.8. Pertama, sistem diuji tanpa gangguan dengan rentang perubahan *set-point* sebesar sebesar 40 mm. Parameter PI yang digunakan adalah 0,005, 0,01, dan 0,02. Hasil yang didapat adalah sistem menghasilkan error yang sangat besar seperti tampak pada Gambar 4.7, dapat disimpulkan bahwa kontroler logika fuzzy tipe PI belum mampu untuk meredam *wind-up* yang terjadi pada sistem. Ketika nilai PI terlalu kecil, maka respon sistem menjadi lambat sehingga akumulasi sinyal error pada sinyal kontrol juga semakin besar. Sebaliknya, ketika nilai PI terlalu besar maka meskipun respon sistem menjadi lebih cepat namun akumulasi nilai error pada sinyal kontrol juga semakin besar.



Gambar 4.7 Respons tanpa gangguan fuzzy-PI

Kemudian sistem diuji dengan pengendali logika *fuzzy* tipe PI skema *robust self-tuning*. Pengujian dilakukan dengan menggunakan parameter PI yang nilainya tiga kali lebih besar dari pada sebelumnya. Sehingga ditetapkan pengujian dengan nilai PI adalah 0,015, 0,03, dan 0,06. Pada *Gambar 4.8* menjelaskan bahwa dapat bahwa ketiga parameter tersebut hampir menghasilkan nilai *steady-state* yang sama. Dengan demikian dapat dikatakan bahwa skema *robust self-tuning* mampu memberikan pengaruh pada sistem sehingga dapat beradaptasi terhadap kondisi saturasi aktuator. Berdasarkan pada nilai *steady state*-nya, didapatkan bahwa parameter PI dengan nilai 0,03 menghasilkan respons yang paling baik.



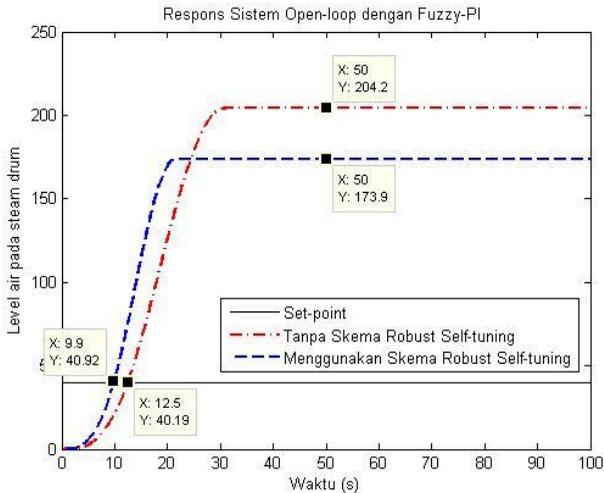
**Gambar 4.8** Respons tanpa gangguan *fuzzy-PI robust self-tuning*

Jika dibandingkan antara respons sistem dengan skema *robust self-tuning* dengan sistem tanpa skema *robust self-tuning* didapatkan respons seperti tampak pada *Gambar 4.9*. Kontroler dengan skema *robust self-tuning* menghasilkan error *steady-state* yang dimana nilainya masih lebih kecil bila dibandingkan dengan kontroler tanpa skema *robust self-tuning* yang menghasilkan error *steady-state* dengan selisih sebesar 75,75% atau 30,3 mm. Meskipun parameter PI pada skema *robust self-tuning* menjadi tiga kali lebih besar, berdasarkan *Tabel 4.2* dapat disimpulkan bahwa skema kontroler tersebut mampu beradaptasi untuk membuat sistem menyesuaikan terhadap *wind-up* sistem yang terjadi. Pada skema *robust self-tuning* respons sistem lebih cepat menuju level air 40 mm sehingga proses akumulasi nilai error juga menjadi berkurang, meskipun begitu

sinyal kontrol yang dihasilkan tetap besar. Namun karena kontroler mampu beradaptasi, setelah 40 mm parameter PI menjadi diperbesar sehingga proses koreksi error juga semakin cepat.

**Tabel 4.2** Pengaruh skema *robust self-tuning* terhadap kriteria sistem

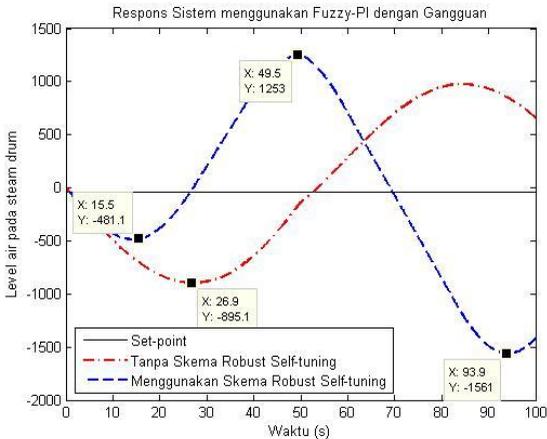
Kriteria Sistem	Tanpa skema <i>Robust-Self-tuning</i>	Dengan skema <i>Robust-Self-tuning</i>
Eror <i>steady-state</i>	164,2 mm	133,9 mm
Waktu untuk mencapai 40 mm	12,5 detik	9,9 detik
Sinyal kontrol (PI) saat mencapai 40 mm	464,3007	695.6142



**Gambar 4.9** Respons tanpa gangguan *fuzzy-PI*

Kemudian sistem diuji dengan gangguan dan tanpa skema *robust self-tuning*. Sistem diuji dengan diberikan *set-point* turun menjadi -40 mm disertai beban nominal, dan kemudian setelah 50 detik sistem diberi beban maksimal. Didapatkan respon sistem yang harusnya diharapkan mampu untuk mengeleminasi eror sistem yang terjadi akibat dari gangguan laju aliran uap yang tinggi, hanya saja bila dengan menggunakan pengendalian *fuzzy* tipe-PI kestabilan sistem menjadi rendah seperti tampak pada *Gambar 4.10*. Dilihat dari grafik respons

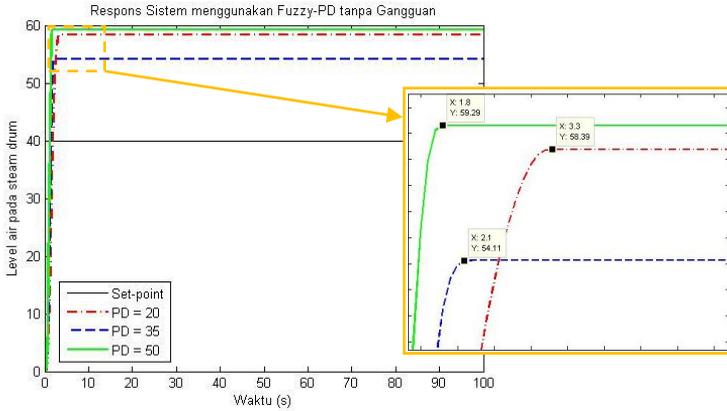
tersebut skema *robust self-tuning* menghasilkan *perturbation peak* 441,1 mm yang lebih besar kecil dibandingkan dengan tanpa skema *robust self-tuning* dengan 855,1 mm *peak*. Hanya saja ketika diberikan gangguan berupa beban maksimal, kontroler dengan skema *robust self-tuning* lebih mampu untuk mempertahankan kestabilannya dari pada skema kontroler tanpa menggunakan *robust self-tuning*.



**Gambar 4.10** Respons dengan gangguan *fuzzy-PI*

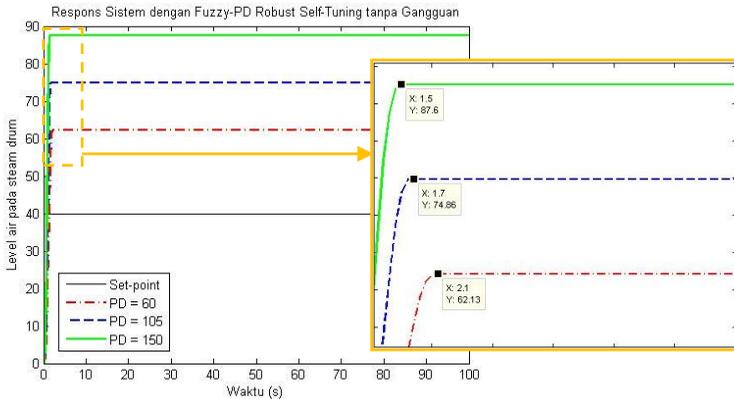
#### 4.3.2 Kontroler Logika *Fuzzy Tipe-PD Robust Self-tuning*

Pada perancangan kontroler logika *fuzzy* tipe-PD seperti pada Gambar 3.7, dilakukan pengujian sistem tanpa gangguan dengan rentang perubahan *set-point* sebesar 40 mm. Dari hasil respons seperti pada Gambar 4.11 didapatkan bahwa dengan menggunakan kontroler logika *fuzzy* tipe-PD menghasilkan error *steady-state* yang jauh lebih kecil bila dibandingkan dengan menggunakan kontroler logika *fuzzy* tipe-PI. Pada Tabel 4.3, didapatkan untuk parameter PD senilai 35 memiliki respons yang lebih optimal dibandingkan dengan dua nilai parameter PD yang lainnya. Kontroler dengan parameter PD sebesar 35 memiliki respons yang lebih cepat bila dibandingkan dengan parameter PD sebesar 20. Selain itu parameter PD dengan nilai 35 juga menghasilkan error *steady-state* yang paling kecil bila dibandingkan dengan dua nilai parameter PD yang lainnya, meskipun respons sistem menjadi lebih lambat bila dibandingkan dengan parameter PD sebesar 50.



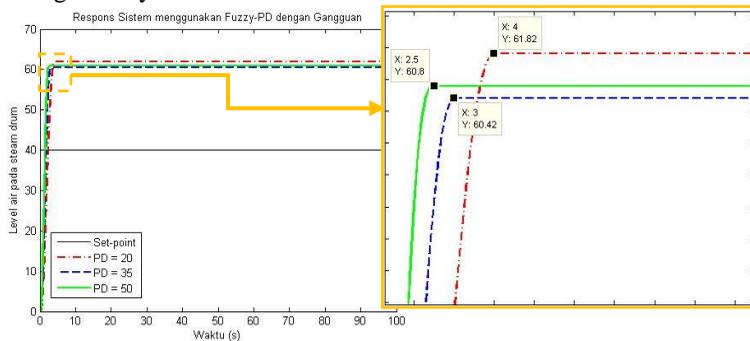
**Gambar 4.10** Respons tanpa gangguan *fuzzy-PD*

Kemudian sistem diuji dengan menggunakan skema *robust self-tuning* dengan ditetapkan parameter PD diperbesar menjadi tiga kalinya. Sehingga dalam pengujian kontroler logika *fuzzy* tipe-PD dengan skema *robust self-tuning* digunakan parameter PD dengan nilai 60, 105, dan 150. Hasilnya didapatkan bahwa parameter PD yang sebelumnya adalah 35 dan diperbesar menjadi 105 tidak menghasilkan respons yang paling optimal diantara yang lainnya. Sehingga dapat dikatakan bahwa proses *tuning* dengan memberikan nilai tiga kali lebih besar pada kontroler logika *fuzzy* tipe-PD *robust self-tuning* tidak berlaku.



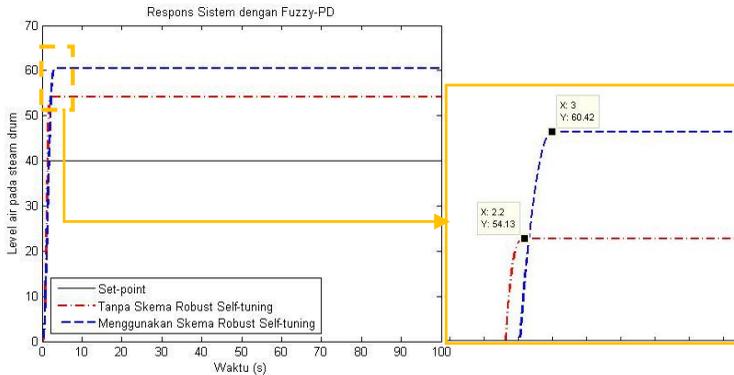
**Gambar 4.11** Respons tanpa gangguan *fuzzy-PD robust self-tuning*

Parameter PD yang digunakan selanjutnya adalah sama seperti sebelum menggunakan skema *robust self-tuning*, yaitu 20, 35, dan 50. Hasil yang didapatkan seperti pada *Gambar 4.12*, parameter PD dengan nilai sebesar 35 menghasilkan respons yang lebih optimal bila dibandingkan dengan dua parameter lain yang diuji. Seperti halnya hasil respons pada *fuzzy-PI* dengan skema *robust self-tuning*, skema tersebut memberikan error *steady-state* yang paling kecil diantara ketiganya, meskipun respons *transient* yang dihasilkan bukan yang paling cepat. Sehingga dapat disimpulkan bahwa skema adaptasi *robust self-tuning* juga dapat diimplementasikan ke dalam kontroler logika *fuzzy* tipe-PD, namun dengan parameter *tuning* PD tidak diubah untuk menjadi lebih besar tiga kalinya.



**Gambar 4.12** Respons sistem dengan parameter PD yang sama

Kemudian jika dibandingkan respons sistem dengan kontroler logika *fuzzy* tipe-PD antara menggunakan skema *robust self-tuning* dan tanpa menggunakan skema *robust self-tuning*, didapatkan respons seperti pada *Gambar 4.13*. Hasil tersebut menunjukkan bahwa skema *robust self-tuning* tidak cukup memberikan pengaruh terhadap kontroler logika *fuzzy* dengan tipe-PD. Terlebih dari itu meskipun tidak signifikan, kontroler logika *fuzzy* tanpa skema *robust self-tuning* sedikit menghasilkan kriteria yang lebih baik seperti tampak pada *Tabel 4.3*. Sehingga dapat disimpulkan berdasarkan pengujian tanpa gangguan ini, kontroler logika *fuzzy* dengan skema *robust self-tuning* tidak mampu untuk memberikan respons sistem dengan kriteria yang lebih baik terhadap adaptasi nilai parameter PD.



**Gambar 4.13** Respons tanpa gangguan *fuzzy*-PD

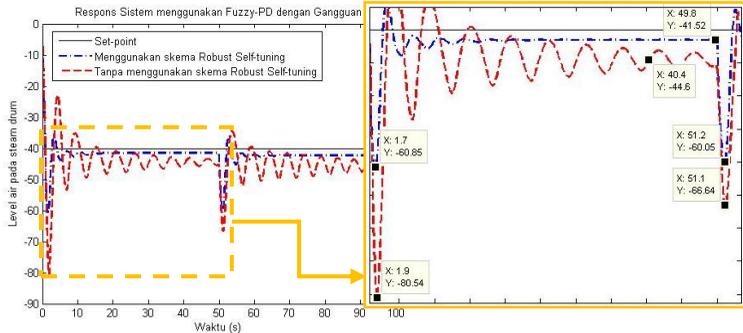
**Tabel 4.3** Pengaruh skema *robust self-tuning* terhadap kriteria sistem

Kriteria Sistem	Tanpa skema <i>Robust-Self-tuning</i>	Dengan skema <i>Robust-Self-tuning</i>
Eror <i>steady-state</i>	14,13 mm	20,42 mm
Waktu untuk mencapai 40 mm	1,45 detik	1,8 detik

Setelah pengujian tanpa gangguan, selanjutnya kontroler logika *fuzzy* tipe- PD diuji pada sistem dengan gangguan. Hasil respons sistem didapatkan bahwa sistem sudah mampu mencapai kestabilannya, hanya saja sistem tersebut masih tidak mampu untuk menghilangkan eror *steady-state* yang dihasilkan seperti tampak pada *Gambar 4.14*. Ketika diberi perubahan *set-point* (turun) -40mm dengan beban nominal, parameter PD yang semakin besar akan menghasilkan nilai *undershoot* serta eror *steady-state* yang semakin kecil. Namun, nilai parameter PD yang semakin besar juga mendegradasi kestabilan dari sistem.

Pada respons sistem dengan kontroler logika *fuzzy* tipe PD dengan skema *robust self-tuning*, sistem mengalami osilasi di sekitar nilai *steady state*-nya. Sedangkan pada sistem dengan kontroler tanpa menggunakan skema *robust self-tuning* dihasilkan sistem dengan hasil respons sistem yang memiliki osilasi lebih kecil. Hal tersebut juga membuktikan bahwa skema *robust self-tuning* tidak mampu menghasilkan kriteria sistem yang lebih baik terhadap pengendalian

logika *fuzzy* tipe-PD. Selain itu dilihat dari *perturbation peak* yang dihasilkan pada Tabel 4.4, respons sistem tanpa skema *robust self-tuning* memiliki *peak (undershoot)* terhadap *steady-state* yang lebih kecil bila dibandingkan dengan sistem menggunakan kontroler dengan menggunakan skema *robust self-tuning*.



**Gambar 4.14** Respons sistem dengan *fuzzy*-PD terhadap gangguan

**Tabel 4.4** Pengaruh skema *robust self-tuning* terhadap kriteria sistem

Kriteria Sistem	Tanpa skema <i>Robust-Self-tuning</i>	Dengan skema <i>Robust-Self-tuning</i>
Eror <i>steady-state</i> (beban nominal)	1,52 mm	4,46 mm
Eror <i>steady-state</i> (beban maksimal)	2,25 mm	5,74 mm
<i>Perturbation peak</i> (perubahan <i>set-point</i> dan beban nominal)	20,85 mm	40,54 mm
<i>Perturbation peak</i> (beban maksimal)	20,05 mm	26,64 mm

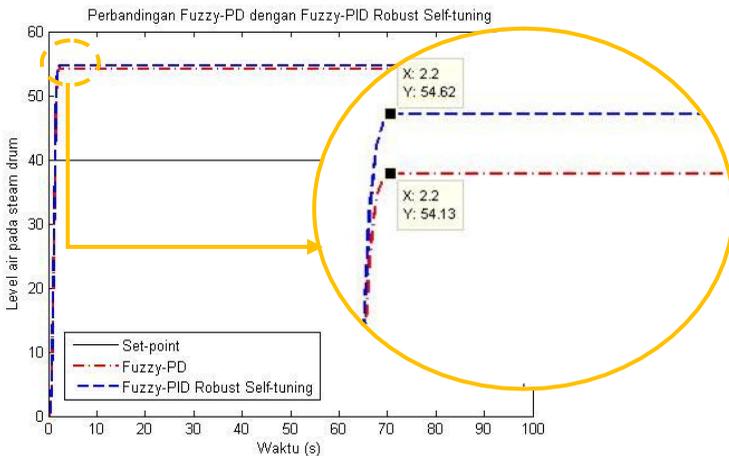
### 4.3.3 Kontroler Logika *Fuzzy* Tipe-PID *Robust Self-tuning*

Para sistem pengendalian logika *fuzzy* PID, seperti halnya pada kontroler logika *fuzzy* tipe PI dan PD sebelumnya dalam tipe PID ini memiliki rancangan *fuzzy-sets* yang sama seperti yang terdapat pada Gambar 3.9. Hanya saja adaptasi *gain* parameter kontroler hanya diterapkan pada parameter kontroler PI saja, sedangkan untuk parameter kontroler PD tidak dilibatkan dengan variabel *scaling factor* dari *gain-tuning mechanism*.

Kontroler logika *fuzzy* ini dirancang berdasarkan jumlahan dari keluaran antara kontroler logika *fuzzy* tipe PI dan PD. Pertamanya,

sistem akan diuji dengan menggunakan nilai parameter kontroler berdasarkan pengujian sebelumnya yang menghasilkan kriteria sistem paling baik terhadap parameter kontroler tersebut. Sehingga pada pengujian ini ditetapkan parameter kontroler dengan nilai PI adalah 0.06 dan PD adalah 35. Pengujian sistem dilakukan dengan memberikan *set-point* yang naik sebesar 40mm tanpa beban, serta kemudian *set-point* turun sebesar 40mm dengan beban nominal, dan kemudian diberikan beban maksimal.

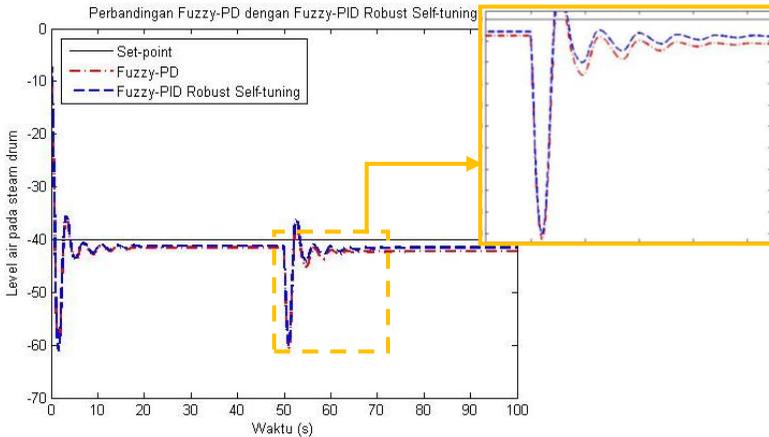
Hasil yang didapatkan adalah seperti tampak pada *Gambar 4.15*, dimana bila dibandingkan antara kontroler logika *fuzzy* tipe-PD dengan kontroler logika *fuzzy* tipe-PID *robust self-tuning* tidak memberikan perbedaan yang cukup signifikan. Cepat respons *transient* sistem terhadap kedua jenis kontroler tersebut sama, yaitu mencapai 40 mm dalam waktu 1,4 detik. Namun dari eror *steady-state* yang dihasilkan kontroler logika *fuzzy* tipe-PID sedikit lebih besar dibandingkan dengan kontroler logika *fuzzy* tipe-PD. Hal tersebut dikarenakan pengaruh dari integrator PI dengan *gain* sebesar 0,06.



**Gambar 4.15** Fuzzy-PD dengan *fuzzy*-PID *robust self-tuning*

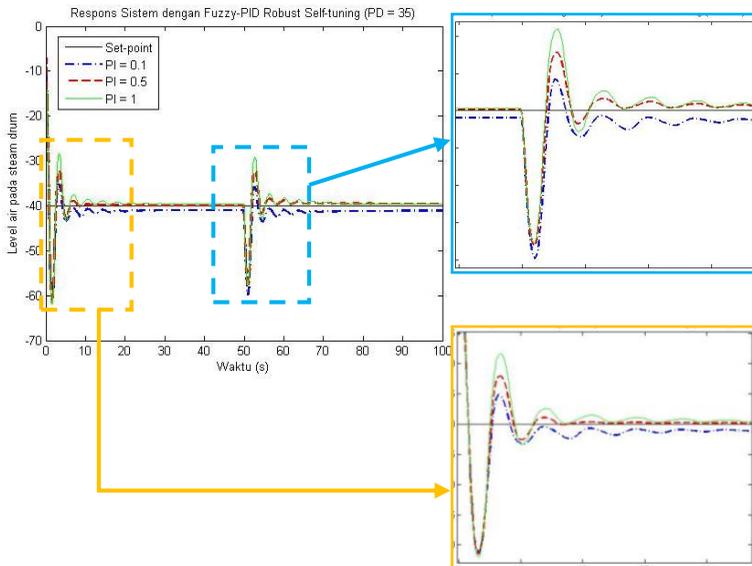
Kemudian berdasarkan respons sistem pada *Gambar 4.16* tersebut, dapat dilihat bahwa respons sistem dengan adanya gangguan juga hampir tidak memiliki perbedaan dengan ketika menggunakan kontroler logika *fuzzy* tipe-PD tanpa skema *robust self-tuning*. Namun dibandingkan terhadap pengujian tanpa beban, meskipun tidak

signifikan kontroler logika *fuzzy* tipe-PID Hal tersebut menunjukkan *robust self-tuning* sedikit menghasilkan respons sistem yang lebih baik dari pada kontroler logika *fuzzy* tipe-PD. Hal tersebut menunjukkan bahwa pengaruh integrator kontroler terlalu kecil terhadap sistem tersebut, dengan demikian maka perlu untuk melakukan *tuning* ulang terhadap parameter PI. Oleh karena itu sistem diuji kembali dengan parameter PI yang lebih besar, diantaranya adalah ditetapkan nilai 0,1, 0,5, dan 1.



**Gambar 4.16** Respons sistem dengan *fuzzy*-PID terhadap gangguan

Didapatkan hasil respons sistem terhadap masing-masing nilai parameter PI yang lebih besar tersebut adalah seperti tampak pada *Gambar 4.17*. Respons sistem untuk ketiga nilai parameter PI tersebut tidak memiliki perbedaan yang signifikan, lebih jelasnya dapat dilihat *Tabel 4.5*. Jika dilihat berdasarkan data karakteristik respons sistem, didapat nilai PI yang semakin tinggi dapat menghasilkan osilasi sistem yang semakin besar Selain itu juga eror *steady-state* yang dihasilkan menjadi lebih kecil seperti pada nilai parameter PI sebesar 0,1 dan 0,5 yang memiliki kriteria respons sistem yang sedikit lebih baik. Namun, jika nilai parameter PI terlalu tinggi maka osilasi, *undershoot*, serta eror *steady-state* yang dihasilkan juga semakin besar. Sehingga didapatkan nilai PI yang mampu menghasilkan respons sistem yang paling baik adalah sebesar 0,5 berdasarkan pada nilai indeks performansi sistem (J) yang dihasilkan pula.



**Gambar 4.17** Respons sistem terhadap perubahan nilai PI

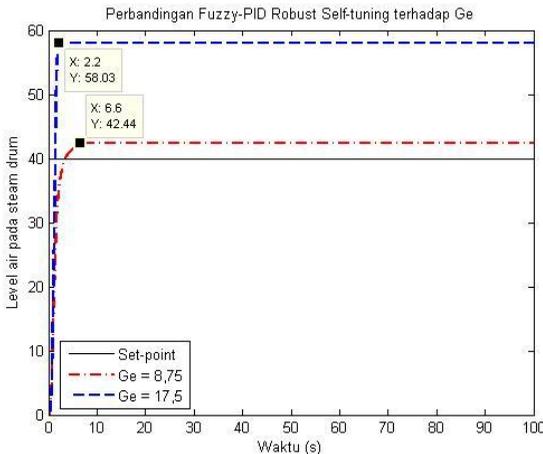
**Tabel 4.5** Kriteria sistem dengan PD = 35 dan PI = 0,06

Kriteria Sistem	PI = 0,1	PI = 0,5	PI = 1
Eror <i>steady-state</i> setelah 50 detik	0,96 mm	0,14 mm	0,33 mm
Eror <i>steady-state</i> setelah 100 detik	1,11 mm	0,44 mm	0,38 mm
<i>Perturbation peak</i> (perubahan <i>set-point</i> dan beban nominal)	19,77 mm	20,66 mm	20,08 mm
<i>Perturbation peak</i> (beban maksimal)	21,14 mm	21,06 mm	21,11 mm
IAE	1.920,8	1.225,9	1.527,7
ISE	15.103,8	13.738,9	15.367,7
J	3.239,1	2.477,2	2.911,7

#### 4.3.4 Pengujian Parameter Normalisasi Kontroler *Fuzzy Tipe PID*

Parameter normalisasi pada kontroler yang dimaksud adalah parameter *gain error* ( $G_e$ ) serta *gain delta-error* ( $G_{de}$ ) sesuai pada yang terdapat pada skema blok diagram kontroler *Gambar 2.9*. Dalam pengujian ini akan ditinjau rentang nilai (pada proses fuzzifikasi) kontroler bekerja serta pengaruh parameter  $G_e$  dan  $G_{de}$  terhadap respons sistem. Berdasarkan grafik respons pada *Gambar 4.18* dapat diamati respons sistem secara keseluruhan terhadap perubahan parameter  $G_e$ . Parameter PI dan PD untuk kontroler logika *fuzzy-PID robust self-tuning* digunakan nilai yang dapat menghasilkan respons sistem yang paling baik berdasarkan pengujian sebelumnya. Sehingga ditetapkan nilai parameter PI sebesar 0,5 dan nilai parameter PD sebesar 35.

Pengujian tanpa beban, untuk nilai  $G_e$  diperkecil menjadi 8,75 dihasilkan respons sistem seperti pada *Gambar 4.17*. Hasil tersebut menunjukkan bahwa parameter  $G_e$  dengan nilai yang dua kali lebih kecil dari sebelumnya memiliki kriteria sistem yang lebih baik seperti tampak pada *Tabel 4.6*. Error *steady-state* yang dihasilkan oleh kontroler dengan  $G_e$  sebesar 8,75 jauh lebih kecil bila dibandingkan dengan nilai  $G_e$  sebesar 17,5. Hal tersebut terjadi karena sinyal kontrol yang dihasilkan pada  $G_e$  sebesar 8,75 lebih kecil, dan dapat disimpulkan *wind-up* sistem yang terjadi juga lebih kecil.



**Gambar 4.18** Respons sistem terhadap perubahan  $G_e$  tanpa gangguan

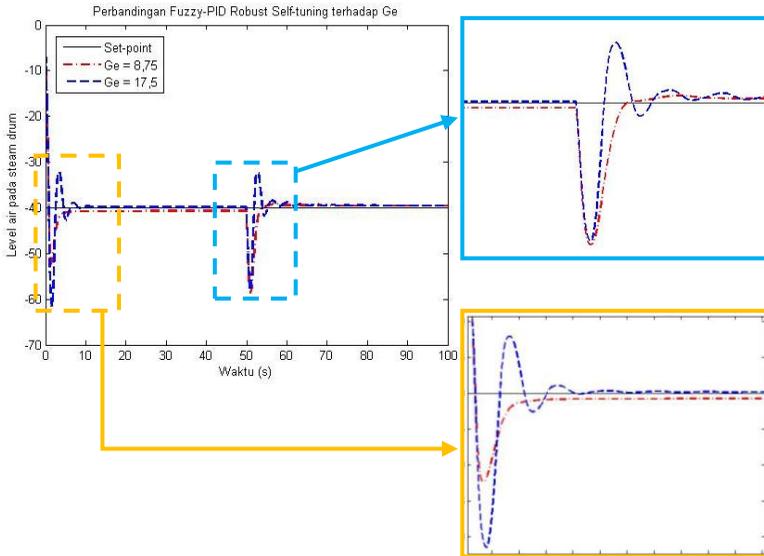
**Tabel 4.6** Kriteria sistem tanpa gangguan terhadap perbedaan Ge

Kriteria Sistem	Ge = 8,75	Ge = 17,5
Eror <i>steady-state</i>	2,44 mm	18,03 mm
Waktu untuk mencapai 40 mm	3,5 detik	1,4 detik
Sinyal kontrol (PI) saat mencapai 40 mm	428,8	1.220,1

Selanjutnya sistem diuji dengan diberikan gangguan dan perubahan *set-point* turun, dengan respons yang dihasilkan adalah seperti tampak pada *Gambar 4.19*. Didapatkan bahwa dari respons tersebut bahwa nilai pada Ge sebesar 8,75, osilasi yang dihasilkan oleh sistem tersebut semakin kecil. Hal tersebut menunjukkan bahwa kestabilan sistem menjadi lebih tinggi. Selain itu berdasarkan pada *Tabel 4.7* menjelaskan bahwa *perturbation-peak* yang dihasilkan Ge sebesar 8,75 terhadap perubahan *set-point* turun 40 mm dengan beban nominal menghasilkan respons yang lebih baik. Namun, dari eror *steady-state* dan *perturbation-peak* pada beban maksimal yang dihasilkan parameter Ge sebesar 17,5 masih sedikit lebih baik meskipun perbedaannya tidak signifikan. Dapat disimpulkan bahwa nilai Ge yang semakin besar dapat memberikan sedikit ketahanan terhadap gangguan dan mempercepat respons sistem, namun osilasi dan *wind-up* sistem semakin besar. Kemudian jika dilihat dari respons keseluruhan dari sistem berdasarkan pada kriteria yang dihasilkan, parameter Ge sebesar 8,75 lebih baik daripada parameter Ge sebesar 17,5.

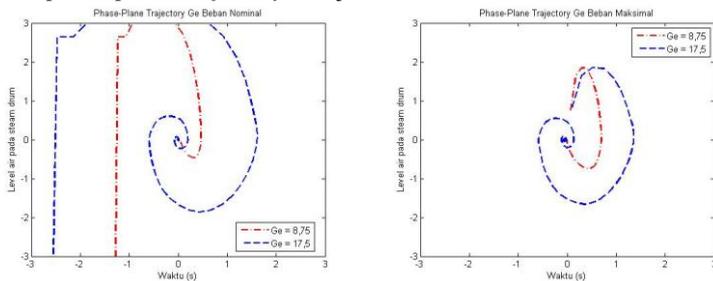
**Tabel 4.7** Kriteria sistem dengan gangguan terhadap perubahan Ge

Kriteria Sistem	Ge = 8,75	Ge = 17,5
Eror <i>steady-state</i> setelah 50 detik	1,13 mm	0,14 mm
Eror <i>steady-state</i> setelah 100 detik	0,46 mm	0,44 mm
<i>Settling-time</i> (beban nominal)	2,45 detik	2,8 detik
<i>Settling-time</i> (beban maksimal)	1,75 detik	2,55 detik
<i>Perturbation peak</i> (perubahan <i>set-point</i> dan beban nominal)	12,34 mm	21,49 mm
<i>Perturbation peak</i> (beban maksimal)	18,50 mm	18,06 mm
IAE	1.268,6	1.225,9
ISE	11.164,4	13.738,8
J	2.258,2	2.477,2



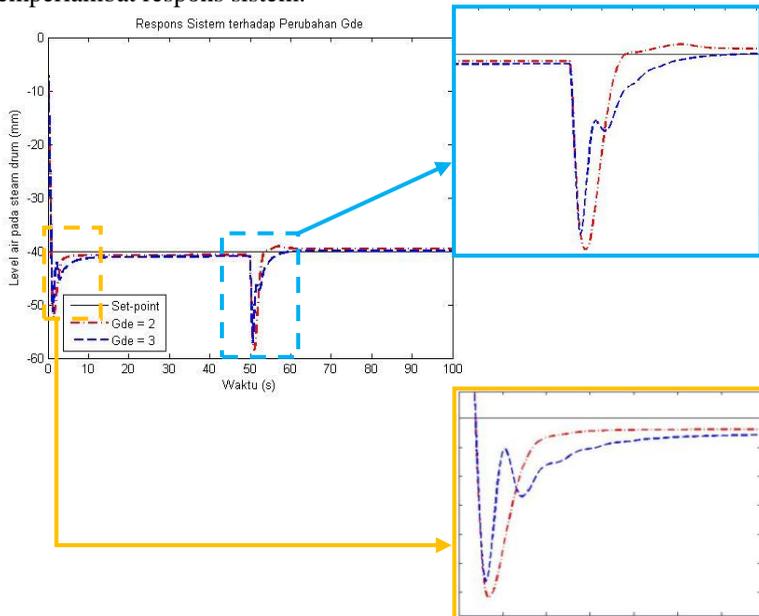
**Gambar 4.19** Respons sistem terhadap nilai  $G_e$  dengan gangguan

Berdasarkan grafik dari *phase plane trajectory* pada Gambar 4.20, didapatkan bahwa semakin besar nilai  $G_e$  maka respons sistem menghasilkan osilasi semakin kecil karena rentang nilai pada proses fuzzifikasi terhadap sinyal error semakin kecil, dalam artian rentan (*band*) proses fuzzifikasi terhadap sinyal error semakin sempit. Akibatnya untuk menuju titik *steady-state* pada koordinat (0,0) pada grafik *phase plane trajectory* menjadi semakin lama.



**Gambar 4.20.** *Phase plane trajectory* terhadap perubahan  $G_e$

Selanjutnya akan dilakukan pengujian sistem dengan meninjau respons dari sistem tersebut terhadap perubahan nilai parameter Gde yang terdapat pada *Gambar 2.9*, sehingga dapat diamati respons keseluruhan sistem pada *Gambar 4.21*. Berdasarkan hasil pengujian tersebut didapatkan seperti pada *Tabel 4.8*, pada respons sistem terhadap *perturbation-peak* yang dihasilkan saat beban maksimal tidak menghasilkan perbedaan nilai yang signifikan terhadap kedua parameter yang diuji tersebut. Sehingga dapat disimpulkan bahwa parameter Gde tidak banyak mempengaruhi ketahanan sistem terhadap *perturbation peak* yang dihasilkan, baik ketika perubahan *set-point* turun dengan beban nominal maupun ketika *steady-state* dengan beban maksimal. Selain itu untuk eror *steady-state* yang dihasilkan pun baik pada beban nominal maupun ketika beban maksimal menghasilkan nilai yang tidak jauh berbeda. Namun dari waktu *transient* yang dihasilkan, Gde yang semakin besar dengan redaman yang tinggi dapat memperlambat respons sistem.

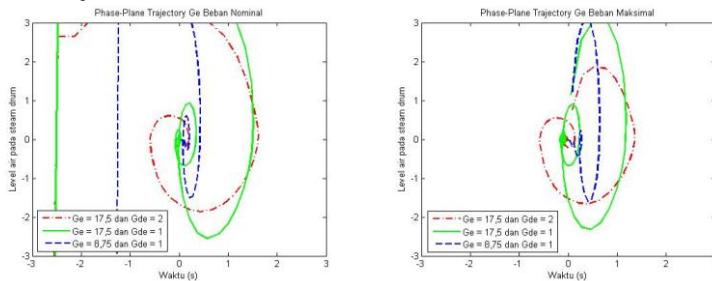


**Gambar 4.21** Respons sistem terhadap nilai Gde dengan gangguan

**Tabel 4.8** Kriteria sistem dengan gangguan terhadap perubahan Gde

Kriteria Sistem	Gde = 2	Gde = 3
Error <i>steady-state</i> setelah 50 detik	0,66 mm	0,92 mm
Error <i>steady-state</i> setelah 100 detik	0,46 mm	0,09 mm
<i>Settling-time</i> (perubahan <i>set-point</i> dan beban nominal)	2,8 detik	3,6 detik
<i>Settling-time</i> (beban maksimal)	2,6 detik	3,2 detik
<i>Perturbation peak</i> (perubahan <i>set-point</i> dan beban nominal)	12,35 mm	11,27 mm
<i>Perturbation peak</i> (beban maksimal)	18,5 mm	17,09 mm
IAE	1.268,6	1.252,6
ISE	11.164,4	9.250,2
J	2.258,2	2.052,4

Pada Gambar 4.22 didapatkan bahwa dengan memperkecil nilai parameter Gde berarti juga memperbesar rentang nilai (*band*) dari proses *fuzzifikasi* terhadap sinyal selisih error semakin besar. Sehingga dalam proses *fuzzifikasi*, sinyal tersebut nilainya lebih cenderung dianggap sebagai *Zero Equals (ZE)* dengan nilai selisih error yang menjadi lebih kecil. Sehingga hal tersebut dapat meningkatkan stabilitas dari sistem tersebut. Nilai Gde sebesar 1 dapat dilihat memiliki respons yang paling baik karena lebih cepat dan sedikit osilasi ketika menuju koordinat (0,0). Namun jika nilai Gde terlalu kecil dapat dilihat bahwa faktor redaman dari kontroler juga menjadi lebih kecil dan osilasi disekitar nilai *steady-state* menjadi semakin besar.



**Gambar 4.22** Phase plane trajectory terhadap perubahan Gde

#### 4.4 Analisa Kontroler pada *Three-element Control*

Pada kontroler *three-element control* akan dirancang suatu model kontrol *cascade* dengan tiga elemen atau variabel yang akan dilibatkan dalam pengendalian seperti pada *Gambar 2.9*. Variabel tersebut diantaranya adalah level *steam drum* sebagai *set-point* dan juga *outer-loop* dalam kontrol *cascade* sedangkan sebagai *inner-loop* adalah laju aliran *feedwater* serta diberikan *disturbance rejection* oleh variabel laju aliran *steam*.

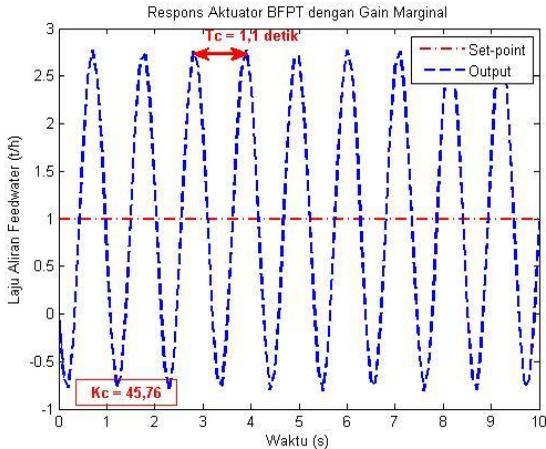
Sehingga pengujian dari rancangan ini dimulai dari pengujian kontrol *inner-loop* pada BFPT, kemudian setelah itu dilanjutkan pengujian keseluruhan sistem dengan *tuning* kontroler pada *outer-loop*. Pada *inner-loop* karena aktuator BFPT cukup sederhana dengan sistem orde 1 maka cukup digunakan kontroler PI, sedangkan untuk kontroler *outer-loop* akan digunakan kontroler logika *fuzzy* PID dengan skema *robust self-tuning* dengan modifikasi rule seperti sebelumnya.

##### 4.4.1 Pengujian Kontroler pada Sistem *Inner-loop*

Pada aktuator BFPT, akan dilakukan metode *tuning ziegler—nichols* dengan *closed-loop* untuk mendapatkan parameter proporsional dan waktu integral yang sesuai. Didapatkan *gain marginal* dari sistem adalah sebesar 45,76 sehingga menghasilkan periode osilasi sebesar 1,1 detik seperti pada *Gambar 4.23*. Melalui penghitungan berdasarkan tabel *ziegler-nichols* metode *closed-loop* seperti pada *Tabel 3.1*, didapatkan nilai parameter proporsional (P), konstanta waktu integral (Ti), dan konstanta waktu derivatif (Td) seperti pada *Tabel 4.9*.

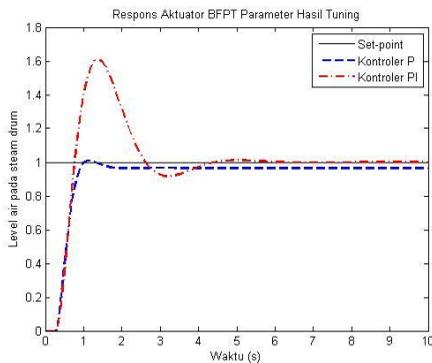
**Tabel 4.9** Hasil tuning parameter PID terhadap aktuator BFPT

Jenis Kontroler	Proportional (P)	Integral-time (Ti)	Derivative-time (Td)
P	22,88	-	-
PI	20,592	0,9167 detik	-
PID	27,456	0,55 detik	0,225 detik



**Gambar 4.23** Respons BFPT dengan *gain marginal*

Dari hasil *tuning* tersebut kemudian di uji pada sistem tersebut, berdasarkan *Gambar 4.24* didapatkan hasil bahwa sistem tersebut sudah memiliki respons *transient* yang cepat, hanya saja *overshoot* cukup besar dengan nilai 63,92% masih dihasilkan oleh kontroler PI sedangkan kontroler P masih menghasilkan eror *steady-state* meskipun nilainya kecil yaitu hanya 3,43%. Disamping itu untuk kontroler PID hasil *tuning* menghasilkan respons sistem yang tidak stabil, sehingga hasil *tuning* tersebut tidak dapat diimplementasikan ke dalam pengaturan *inner-loop closed-loop* dari aktuator BFPT.

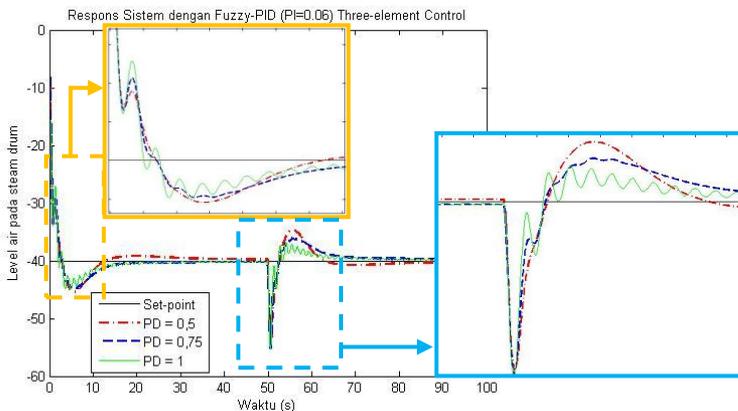


**Gambar 4.24** Respons BFPT dengan kontroler PI

#### 4.4.2 Pengujian Kontroler pada Sistem *Outer-loop*

Setelah didapatkan parameter PID hasil *tuning* untuk pengendalian pada *inner-loop* maka selanjutnya diimplementasikan pada sistem secara keseluruhan untuk *three-element control*. Karena *integrator* sudah didapatkan pada *outer-loop* serta sistem sudah merupakan tipe-1, maka untuk pengendalian *inner-loop* ditetapkan cukup menggunakan kontroler P hasil *tuning* saja.

Mulnaya sistem tersebut diuji dengan nilai parameter pengendalian pada *outer loop* dengan PI sebesar 0,06 dan dicari nilai PD yang paling sesuai. Hasilnya didapatkan respons sistem seperti pada *Gambar 4.25*. Dapat disimpulkan bahwa pada *three-element control* mampu meredam *wind-up* lebih baik dari pada *single-element control* sehingga menghasilkan *undershoot* yang lebih kecil ketika diberikan *set-point* turun dengan beban nominal. Sedangkan *perturbation-peak* yang dihasilkan perbedaannya tidak signifikan, dengan parameter PD sebesar 1 memiliki nilai paling kecil dari yang lainnya. Dari eror *steady-state* yang dihasilkan, ketiganya sudah menghasilkan nilai yang cukup kecil seperti tampak pada *Tabel 4.10*. Secara keseluruhan sistem berdasarkan kriteria respons yang dihasilkan, parameter kontroler PD dengan nilai 1 menghasilkan respons sistem yang paling baik diantara ketiganya. Namun, pada parameter PD sebesar 1 menghasilkan osilasi disekitar nilai *steady state*-nya, sehingga pada nilai parameter tersebut kestabilan sistem terdegradasi.

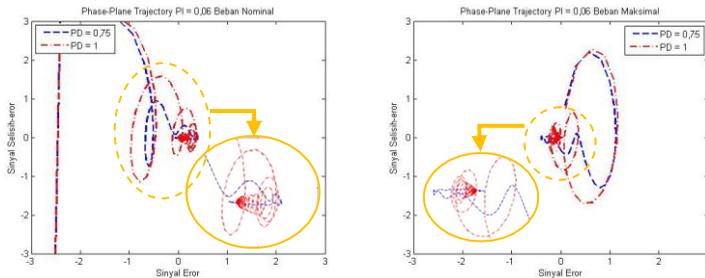


**Gambar 4.25** Respons sistem pada *three-element control*

**Tabel 4.10** Kriteria sistem dengan gangguan terhadap perubahan PD

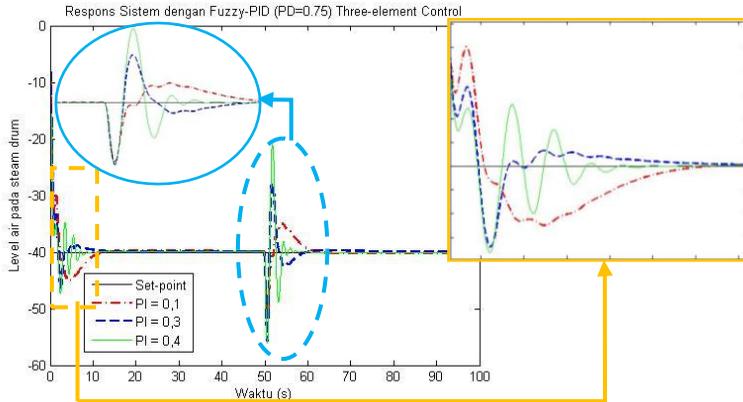
Kriteria Sistem	PD = 0,5	PD = 0,75	PD = 1
IAE	1.387,8	1.305,6	1.057,2
ISE	9.356,0	8.737,7	7.814,5
J	2.184,6	2.048,8	1.732,9

Phase plane trajectory yang dihasilkan seperti tampak pada Gambar 4.26, kontroler dengan PD sebesar 1 menghasilkan osilasi yang lebih besar dari pada PD dengan nilai 0,75. Bahkan di sekitar titik koordinat (0,0) pun pada parameter PD sebesar 1 masih menghasilkan osilasi. Selain itu, bila dilihat dari respons tersebut mulanya titik setimbang atau *ekuilibrium* yang dihasilkan sistem bukan pada koordinat (0,0) meskipun pada akhirnya bergeser menuju titik (0,0). Hal tersebut menunjukkan bahwa faktor integral pada kontroler masih terlalu kecil, sehingga proses koreksi error menjadi lebih lambat.



**Gambar 4.26** Phase plane trajectory terhadap variasi nilai PD

Nilai PI yang terlalu kecil menyebabkan pergeseran titik setimbang atau *ekuilibrium* dari sistem menjadi lebih lambat. Oleh karena itu dilakukan pengujian terhadap kontroler tersebut dengan menetapkan parameter PD sebesar 0,75 serta PI sebesar 0,1, 0,3, dan 0,4. Nilai *perturbation-peak* serta respons waktu *transient* yang dihasilkan oleh sistem tidak menunjukkan perbedaan yang signifikan, hanya saja secara keseluruhan berdasarkan nilai kriteria sistem yang dihasilkan seperti pada Gambar 4.27 dan Tabel 4.11 parameter PI sebesar 0,3 menghasilkan respons yang paling baik.

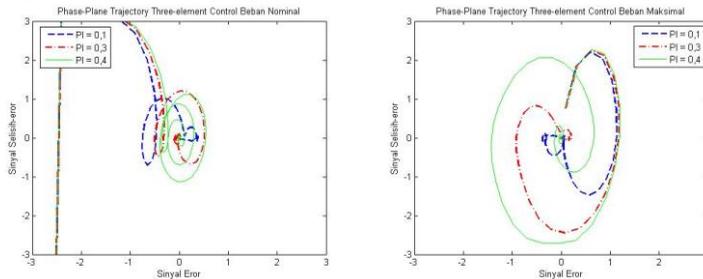


**Gambar 4.27** Respons sistem dengan variasi nilai PI

**Tabel 4.11** Kriteria sistem dengan gangguan terhadap perubahan Gde

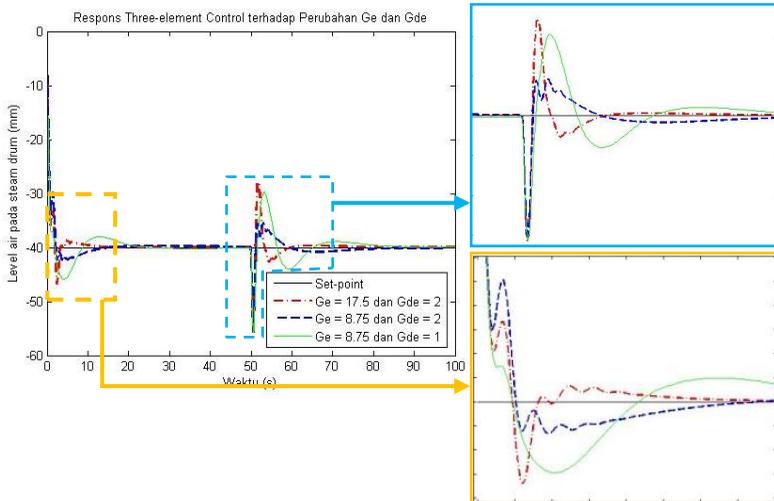
Kriteria Sistem	PI = 0,1	PI = 0,3	PD = 0,4
IAE	989,2806	734,7113	753,1171
ISE	7.964,5	7.529,4	9.521,5
J	1.686,8	1.414,2	1.630,0

Secara *phase plane trajectory* didapatkan hasil respons seperti pada *Gambar 4.28*, dimana ketiganya sudah menghasilkan respons sistem yang mampu menuju titik setimbang pada koordinat (0,0) lebih cepat dari pada sebelumnya. Dapat disimpulkan bahwa nilai PI yang semakin besar dapat lebih mempercepat respons sistem dalam pengoreksian nilai error, namun kestabilan dari sistem menjadi terdegradasi dan menghasilkan osilasi yang lebih besar.



**Gambar 4.28** *Phase plane trajectory* terhadap variasi nilai PI

Seperti pengujian sistem pada *single-element control* sebelumnya, pada *three-element control* setelah ditemukan parameter kontroler yang paling baik selanjutnya akan dilakukan proses *tuning* ulang untuk parameter normalisasi pada kontroler. Hasil pengujian didapatkan seperti tampak pada *Gambar 4.29*. Kemudian berdasarkan pada *Tabel 4.12*, didapatkan bahwa parameter normalisasi kontroler yang memberikan respons sistem yang paling baik adalah dengan nilai  $G_e$  sebesar 17,5 dan  $G_{de}$  sebesar 2.



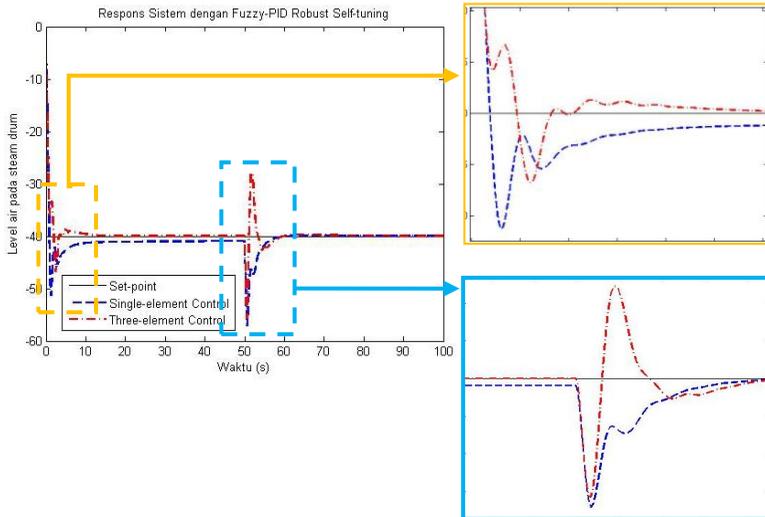
**Gambar 4.29** Respons sistem terhadap variasi nilai  $G_e$  dan  $G_{de}$

**Tabel 4.12** Kriteria sistem dengan *single* dan *three element control*

Kriteria Sistem	$G_e = 17,5$ $G_{de} = 2$	$G_e = 8,75$ $G_{de} = 2$	$G_e = 8,75$ $G_{de} = 1$
IAE	734,7113	899,4748	1.429,6
ISE	7.529,4	7.041,5	10.339,2
J	1.414,2	1.513,7	2.320,6

Berdasarkan pada parameter terbaik dari hasil pengujian yang telah dilakukan, jika dibandingkan respons sistem antara menggunakan *single-element control* dengan *three-element control*, respons sistem dengan menggunakan *three-element control* memiliki respons yang lebih baik berdasarkan pada hampir di semua kriteria respons sistem

yang dihasilkan seperti tampak pada *Gambar 4.30* dan *Tabel 4.13*. Hanya nilai dari *settling-time* sistem yang dihasilkan tidak signifikan karena dihasilkannya *peak-overshoot* pada sistem.



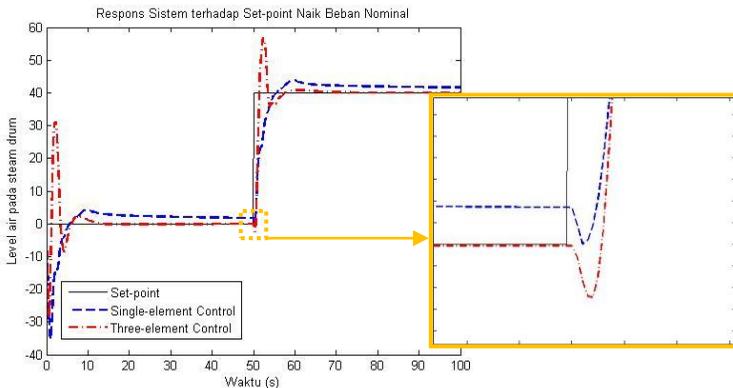
**Gambar 4.30** Respons sistem dengan *single* dan *three element control*

**Tabel 4.13** Kriteria sistem dengan *single* dan *three element control*

Kriteria Sistem	<i>Single-element Control</i>	<i>Three-element Control</i>
Eror <i>steady-state</i> setelah 50 detik	0,66 mm	0,04 mm
Eror <i>steady-state</i> setelah 100 detik	0,46 mm	0,06 mm
<i>Settling-time</i> (perubahan <i>set-point</i> dan beban nominal)	2,8 detik	2,85 detik
<i>Settling-time</i> (beban maksimal)	2,55 detik	2,55 detik
<i>Perturbation peak</i> (perubahan <i>set-point</i> dan beban nominal)	12,34 mm	6,79 mm
<i>Perturbation peak</i> (beban maksimal)	18,5 mm	15,69 mm
<i>IAE</i>	1.252,6	734,7113
<i>ISE</i>	9.250,2	7.529,4
<i>J</i>	2.052,4	1.414,2

#### 4.5 Pengujian Sistem terhadap *Set-point* Naik

Setelah dilakukan pengujian sistem dengan *set-point* turun dengan beban nominal, selanjutnya sistem diuji dengan *set-point* naik 40 mm dengan beban nominal. Kemudian sama seperti sebelumnya setelah 50 detik sistem akan diberikan beban maksimal. Respons sistem yang dihasilkan adalah seperti pada *Gambar 4.31*, yang menunjukkan pada *single-element control* mampu menghasilkan respons yang jauh lebih baik dari pada *three-element control* terhadap perubahan *set-point*. Hal tersebut menunjukkan bahwa *single-element control* digunakan saat proses *starting* yang dimana pada saat itulah pada sistem banyak terjadi perubahan *set-point* naik. Secara keseluruhan kriteria sistem dengan perubahan *set-point* naik dapat dilihat pada *Tabel 4.14*.



**Gambar 4.31** Respons sistem terhadap perubahan *set-point* naik

**Tabel 4.14(a)** Respons sistem pada *single* dan *three element control*

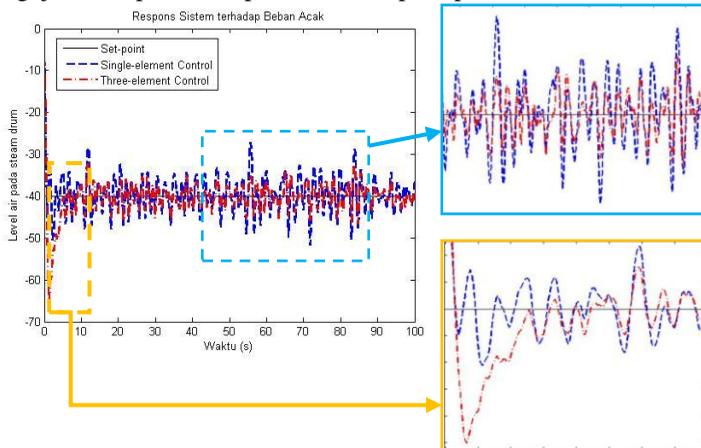
Kriteria Sistem	<i>Single-element Control</i>	<i>Three-element Control</i>
<i>Inverse response peak (initial)</i>	35,27 mm	28,69
<i>Inverse response peak (perubahan set-point)</i>	1,70 mm	2,44 mm
<i>Overshoot (initial)</i>	4,44 mm	31,26 mm
<i>Overshoot (perubahan set-point)</i>	3,91 mm	16,82 mm
<i>Settling-time (initial)</i>	10,45 detik	5,1 detik
<i>Settling-time (perubahan set-point)</i>	3,25 detik	2,75 detik
Eror <i>steady-state</i> (setelah 50 detik)	1,71 mm	0,07 mm
Eror <i>steady-state</i> (setelah 100 detik)	1,59 mm	0,05 mm

**Tabel 4.14(b)** Kriteria eror sistem pada *single* dan *three element control*

Kriteria Sistem	<i>Single-element Control</i>	<i>Three-element Control</i>
<i>IAE</i>	3.418,9	1.713,3
<i>ISE</i>	39.730,0	34.744,0
<i>J</i>	7.050,0	5.016,3

#### 4.6 Pengujian Sistem terhadap Perubahan Beban Acak

Pengujian dilakukan dengan membandingkan respons kedua skema antara *single-element control* dengan *three-element control* terhadap adanya beban atau gangguan acak. Gangguan yang diberikan adalah nilai acak beban dengan rentang mulai dari 400 MW hingga 700 MW untuk diberikan ke kedua skema pengendalian tersebut. Skema kontroler yang digunakan pada *single-element control* adalah kontroler logika *fuzzy* tipe PID dengan skema *robust self-tuning*, begitu juga pada pengendali *outer-loop* pada *three-element control* sedangkan untuk *inner-loop* digunakan kontroler P konvensional. Seperti pada pengujian sebelumnya, sistem diberikan *set-point* turun 40 mm dengan diberikan beban nilai acak, Parameter kontroler yang digunakan diambil dari nilai yang paling baik berdasarkan pada hasil analisa sebelumnya. Hasil dari pengujian didapatkan respons sistem seperti pada *Gambar 4.32*.



**Gambar 4.32** Respons sistem terhadap beban acak

Skema *three-element control* tidak mampu menghasilkan respons yang baik terhadap perubahan *set-point* seperti pada pengujian sebelumnya. Di sisi lain, pada skema *single-element control* tidak mampu mempertahankan nilai *steady state*-nya dengan baik pada kondisi beban yang fluktuasi (acak), sedangkan pada skema *three-element control* sistem tersebut mampu meredam gangguan lebih baik dengan simpangan yang dihasilkan cenderung lebih kecil. Hal tersebut sekaligus membuktikan bahwa *three-element control* memiliki respons yang lebih cepat.

*Halaman ini sengaja dikosongkan*



**BAB V**  
KESIMPULAN

## BAB 5

### KESIMPULAN

Sistem dengan kontroler PID konvensional tidak mampu mengatasi *wind-up* sistem dengan baik dan menghasilkan 79,5 mm *peak* terhadap perubahan *set-point* dan beban nominal. Selain itu, sistem masih belum dapat mempertahankan nilai *steady state*-nya terhadap beban maksimum dengan menghasilkan 36,93 mm *perturbation peak*.

Kontroler logika *fuzzy* tipe-PI tidak dapat diimplementasikan pada *plant steam drum* karena menghasilkan kriteria respons sistem yang tidak sesuai dengan spesifikasi yang telah ditentukan dan menghasilkan 855,1 mm *peak* dengan beban nominal. Namun, skema *robust self-tuning* masih dapat diimplementasikan untuk memperbaiki respons sistem dengan menghasilkan 441,1 mm *peak* dengan beban nominal.

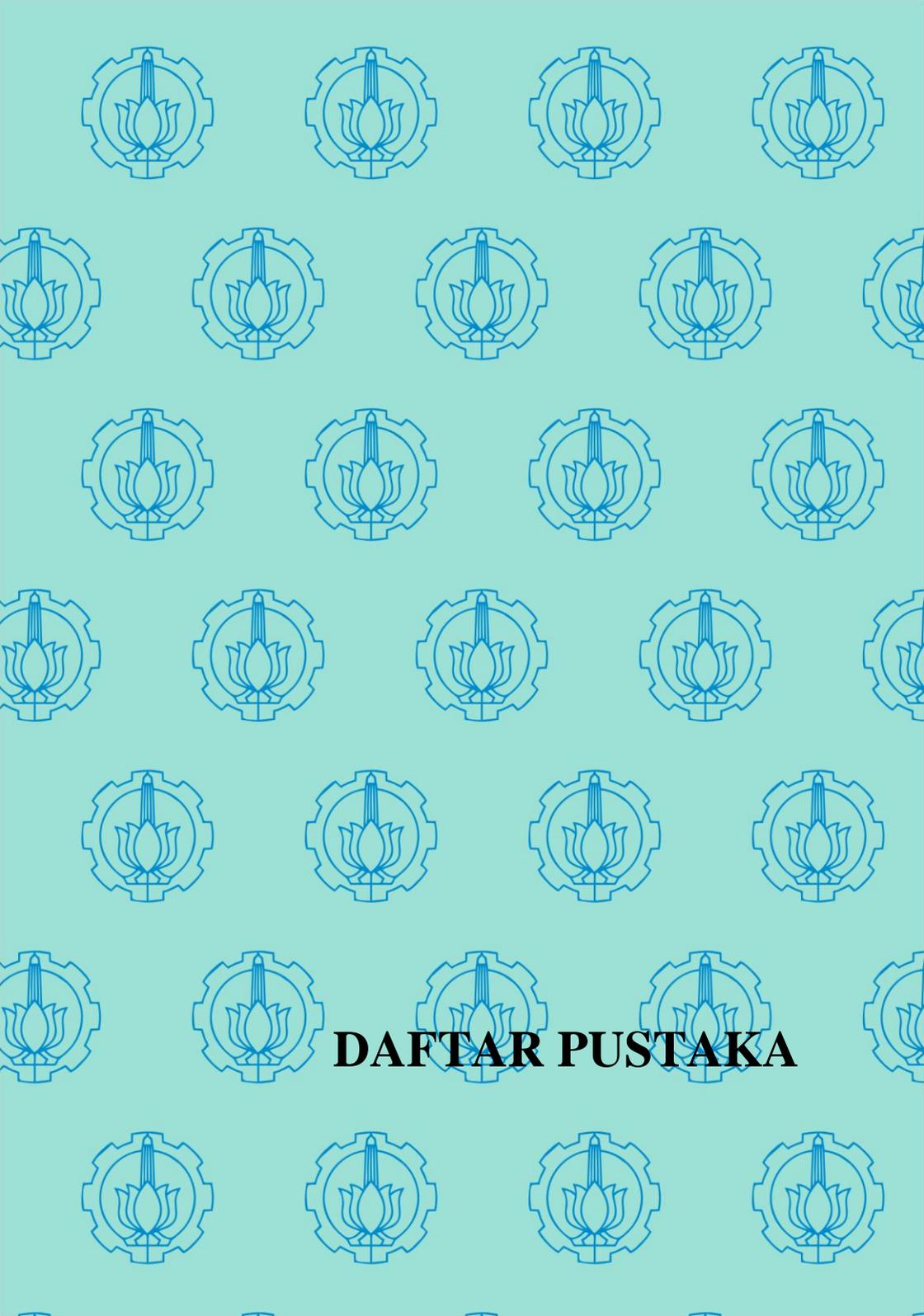
Pada kontroler logika *fuzzy* tipe-PD didapatkan respons sistem yang lebih mampu mengatasi *wind-up* sistem dengan menghasilkan 20,85 terhadap perubahan *set-point*, 20,05 *peak* pada beban maksimal dan masih memiliki eror *steady-state* ( $e_{ss}$ ) sebesar 1,52 mm. Namun, skema *robust self-tuning* tidak dapat menghasilkan respons sistem yang lebih baik dengan menghasilkan 40,54 terhadap perubahan *set-point*, 26,64 mm *peak* pada beban maksimal dan  $e_{ss}$  sebesar 5,74 mm.

Penggunaan kontroler logika *fuzzy* tipe-PID dengan skema *robust self-tuning* mampu melengkapi kekurangan dari kontroler logika *fuzzy* tipe-PI dan tipe-PD dengan menghasilkan 17,09 mm *peak* pada beban maksimal dan eror *steady-state* sebesar 0,09 mm.

Skema pengendalian *three-element control* mampu menghasilkan respons sistem yang lebih tahan terhadap gangguan dari pada sistem ketika menggunakan pengendalian dengan skema *single-element control*. Skema *three-element control* mampu mempertahankan kondisi *steady state*-nya dengan menghasilkan *perturbation-peak* sebesar 15,69 mm terhadap beban maksimal, dimana nilai tersebut lebih kecil dibandingkan dengan *single element control* yang menghasilkan *perturbation peak* sebesar 18,5 mm terhadap beban maksimal.

Untuk proses *starting* pada sistem *boiler*, *single-element control* mampu menghasilkan respons yang lebih baik bila dibandingkan dengan skema *three-element control*. *Single-element control* menghasilkan 3,91 mm *peak* terhadap perubahan *set-point* naik dengan beban nominal, dimana nilai tersebut jauh lebih baik dari pada ketika menggunakan skema *three-element control* yang menghasilkan nilai 16,82 mm.

*Halaman ini sengaja dikosongkan*

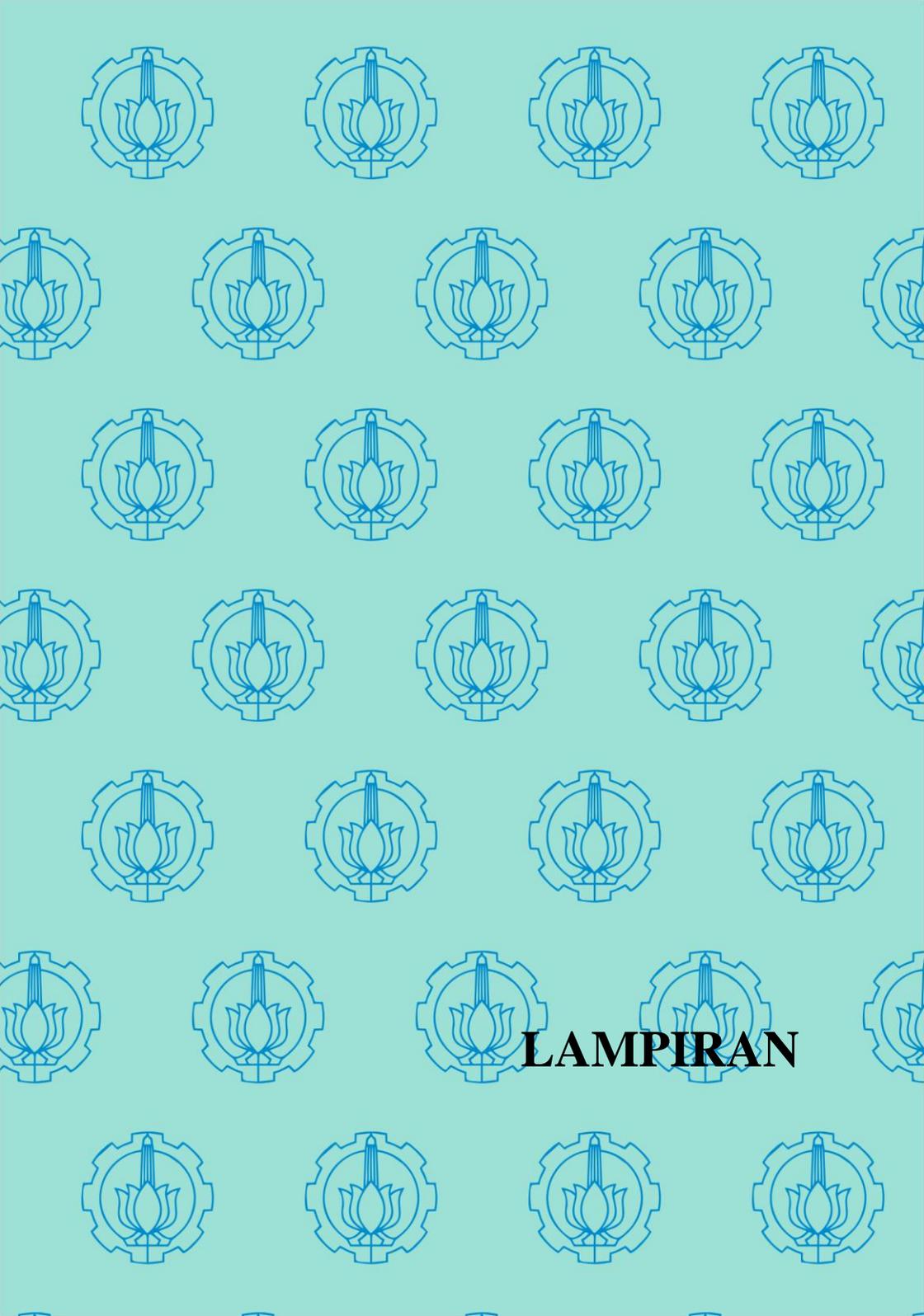


# DAFTAR PUSTAKA

## DAFTAR PUSTAKA

- [1] K. D. P. and N. I. J., "*Power System Engineering*", New Delhi: Tata McGraw-Hill, 2008.
- [2] Gilman. G.F. (Jerry), "*Boiler Control System Engineering: Second Edition*", United States of America (USA): International Society of Automation, 2010.
- [3] Rajani K Mudi, "*A Robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers*", IEEE Transactions on Fuzzy Systems : Vol. 7 - No. 1, February 1999.
- [4] Yesil. Engin. dkk., "*Fuzzy PID Controllers: An Overview*", The 3<sup>rd</sup> Triennial ETAI International Conference on Applied Automatic Systems At: Ohrid. Macedonia, October 2003.
- [5] Yan. Jun, dkk., "*Using Fuzzy Logic: Towards Intelligent Systems*", New York. London: Prentice Hall, 1948.
- [6] Tawfeic. S. R. , "*Boiler Drum-Level Modeling*", pp. 1812 – 1829, 5 May 2013.
- [7] --, "*Saturated Properties Calculator*", Energy Efficiency & Renewable Energy.
- [8] --, "*Using the LabVIEW Shared Variable*", National Instrument, Nov. 12<sup>th</sup> 2018.
- [9] Ogunnaike. Babatunde A. and Ray. W. Harmon, "*Process Dynamics, Modeling, and Control*", Oxford. New York: Oxford University Press, 1994.
- [10] Ogata. Katsuhiko, "*Modern Control Engineering: Fifth Edition*", United States of America (USA): Prentice Hall, 2010 .
- [11] Kevin M. Passino and Stephen Yurkovich, "*Fuzzy Control*", California: Addison Wesley Longman, Inc., 1998.
- [12] Keeman. Vojislav, "*Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*". United States of America: A Bradford Book. MIT Press, 2001.
- [13] Isermann. Rolf and Munchhof. Marco, "*Identification of Dynamic Systems: An Introduction with Applications*", New York. London: Springer, 2011.
- [14] Astrom. Karl J. and Wittenmark, Bjorn, "*Computer Controlled Systems : Theory and Design*", China : Prentice Hall, 1997.

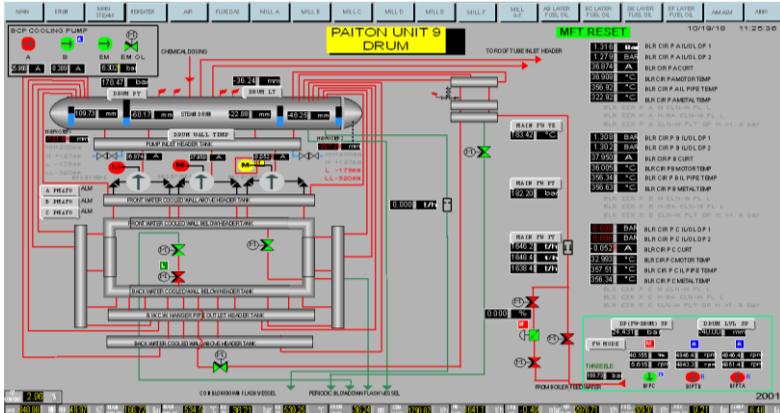
- [15] Chakraborty. Sanjoy Kumar, *“Importance of Three-elements Boiler, Drum Level Control And Its Installation In Power Plant”*. International Journal of Instrumentation and Control Systems (IJICS) Vol.4, No.2, April 2014.



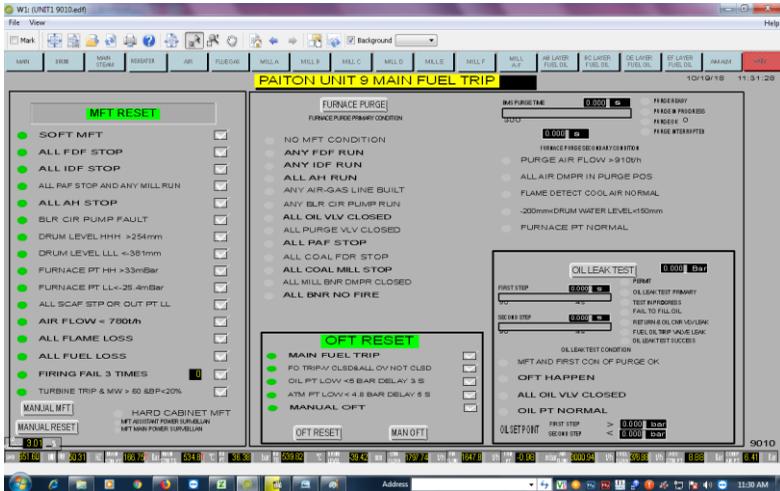
# LAMPIRAN

# LAMPIRAN

Lampiran 1. Tampilan HMI Steam Drum pada control room di PT. PJB UBJ O&M Unit 9 Paiton



Lampiran 2. Tampilan HMI dan kondisi MFT pada control room di PT. PJB UBJ O&M Unit 9 Paiton



### Lampiran 3. Program regresi *least square* pada script Matlab

```
clear m n i j Yd si Xd X2d Y_td teta_t
Y_topi RMSE

clc

Yd = [0 ; fw];
Yd = Yd(1:size(fw,1));
Xd = [0 ; dem];
Xd = Xd(1:size(dem,1));
si = [Yd Xd];

teta = inv(si'*si)*si'*fw
Y_topi = si*teta;
RMSE = sqrt(sum((fw-
Y_topi).^2)/size(fw,1))

plot(fw)
hold;
plot(Y_topi, 'red')
title('BFPT ( Pole=1 ; Zero=0 )')
xlabel('Waktu dengan sampling Ts=0.5s
(t)')
ylabel('Feedwater Flow')
legend('Feedwater Flow', 'Feedwater Flow
Estimated Model')

clear m n i j Yd si Xd X2d Y_td Y_topi
RMSE
```

## Lampiran 4. Kalkulasi nilai massa jenis air dan uap di dalam *steam drum* pada nilai tekanan nominal

**Saturated Properties Calculator** [watch tutorial](#) [view guide](#)  
 Calculates saturated liquid and gas properties for a given pressure or temperature using the IAPWS Industrial Formulation 1997.

Pressure: 11710.2 psig

\* Required

Examples:

Example: Random Pressure

Steam Properties

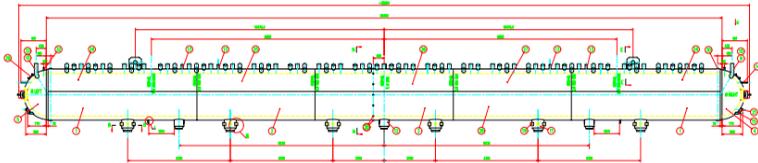
	Saturated Liquid	Evaporation	Saturated Gas	units
Pressure		1,710.2		psig
Temperature		615.2		°F
Specific Enthalpy	639.3	516.3	1,155.6	btu/lbm
Specific Entropy	0.833	0.480	1.314	btu/lbm/R
Specific Volume	0.024	0.207	0.231	ft <sup>3</sup> /lb

Temperature-Entropy Diagram

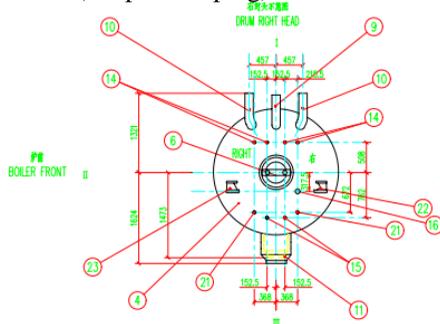
Recent Saturation Properties (max 20) - [\[download\]](#) - [\[clear\]](#)

Pressure	Temperature	Specific Enthalpy			Specific Entropy			Specific Volume		
		Saturated Liquid	Evaporation	Saturated Gas	Saturated Liquid	Evaporation	Saturated Gas	Saturated Liquid	Evaporation	Saturated Gas
psig	°F	btu/lbm			btu/lbm/R			ft <sup>3</sup> /lb		
1,710.2	615.2	639.3	516.3	1,155.6	0.833	0.480	1.314	0.024	0.207	0.231
2,450.0	666.1	726.6	368.6	1,066.2	0.909	0.327	1.237	0.028	0.106	0.134

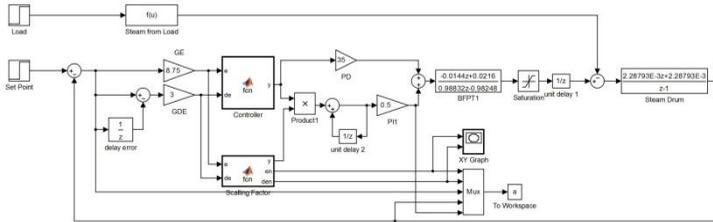
## Lampiran 5. Desain model *steam drum* di PT. PJB UBJ O&M Paiton Unit 9 (tampak depan)



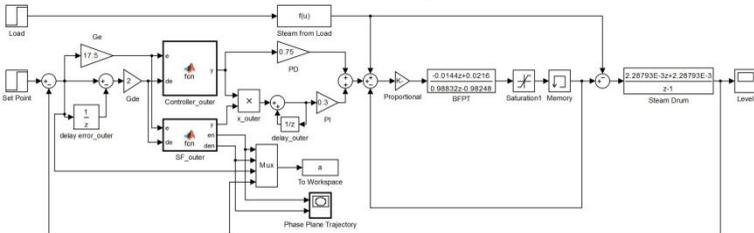
## Lampiran 6. Desain model *steam drum* di PT. PJB UBJ O&M Paiton Unit 9 (tampak samping)



**Lampiran 7.** Diagram blok Simulink untuk skema kontroler logika *fuzzy robust self-tuning* pada *single-element control*



**Lampiran 8.** Diagram blok Simulink untuk skema kontroler logika *fuzzy robust self-tuning* pada *three-element control*



**Lampiran 9.** Desain kontroler logika *fuzzy* dalam bentuk program *script* pada Matlab

```
function y = fcn(e,de)
en=3/700*e;
den=3/10.0118504*de;
mf1=[0 0 0 0 0 0 0];
mf2=[0 0 0 0 0 0 0];
u=[0 0 0 0 0 0 0];

if en<=-3
    mf1(1)=1;
elseif en<=-2
    mf1(1)=-2-en;
    mf1(2)=en+3;
elseif en<=-1
    mf1(2)=-1-en;
    mf1(3)=en+2;
elseif en<=0
    mf1(3)=-en;
```

```

        mf1(4)=en+1;
elseif en<=1
        mf1(4)=1-en;
        mf1(5)=en;
elseif en<=2
        mf1(5)=2-en;
        mf1(6)=en-1;
elseif en<=3
        mf1(6)=3-en;
        mf1(7)=en-2;
else
        mf1(7)=1;
end

if den<=-3
        mf2(1)=1;
elseif den<=-2
        mf2(1)=-2-den;
        mf2(2)=den+3;
elseif den<=-1
        mf2(2)=-1-den;
        mf2(3)=den+2;
elseif den<=0
        mf2(3)=-den;
        mf2(4)=den+1;
elseif den<=1
        mf2(4)=1-den;
        mf2(5)=den;
elseif den<=2
        mf2(5)=2-den;
        mf2(6)=den-1;
elseif den<=3
        mf2(6)=3-den;
        mf2(7)=den-2;
else
        mf2(7)=1;
end

```

```

rule = [ 1 1 1 2 3 3 4 ;
        1 2 2 2 3 4 5 ;
        1 2 3 3 4 5 6 ;
        1 2 3 4 5 6 7 ;
        2 3 4 5 5 6 7 ;
        3 4 5 6 6 6 7 ;
        4 5 5 6 7 7 7 ];

for i=1:7
    for j=1:7
        n = rule(i,j);
        u(n)=max(u(n),min(mf1(i),mf2(j)));
    end
end

c = [-1 -2/3 -1/3 0 1/3 2/3 1];
y = (sum(u.*c))/(sum(u));

y=y*700;

```

**Lampiran 10.** Desain *gain-tuning mechanism* logika fuzzy dalam bentuk program *script* pada Matlab

```

function [y,en,den] = fcn(e,de)
en=3/700*e;
den=3/10.0118504*de;
mf1=[0 0 0 0 0 0 0];
mf2=[0 0 0 0 0 0 0];
u=[0 0 0 0 0 0 0];

if en<=-3
    mf1(1)=1;
elseif en<=-2
    mf1(1)=-2-en;
    mf1(2)=en+3;
elseif en<=-1
    mf1(2)=-1-en;
    mf1(3)=en+2;
elseif en<=0
    mf1(3)=-en;

```

```

        mf1(4)=en+1;
elseif en<=1
        mf1(4)=1-en;
        mf1(5)=en;
elseif en<=2
        mf1(5)=2-en;
        mf1(6)=en-1;
elseif en<=3
        mf1(6)=3-en;
        mf1(7)=en-2;
else
        mf1(7)=1;
end

if den<=-3
        mf2(1)=1;
elseif den<=-2
        mf2(1)=-2-den;
        mf2(2)=den+3;
elseif den<=-1
        mf2(2)=-1-den;
        mf2(3)=den+2;
elseif den<=0
        mf2(3)=-den;
        mf2(4)=den+1;
elseif den<=1
        mf2(4)=1-den;
        mf2(5)=den;
elseif den<=2
        mf2(5)=2-den;
        mf2(6)=den-1;
elseif den<=3
        mf2(6)=3-den;
        mf2(7)=den-2;
else
        mf2(7)=1;
end

```

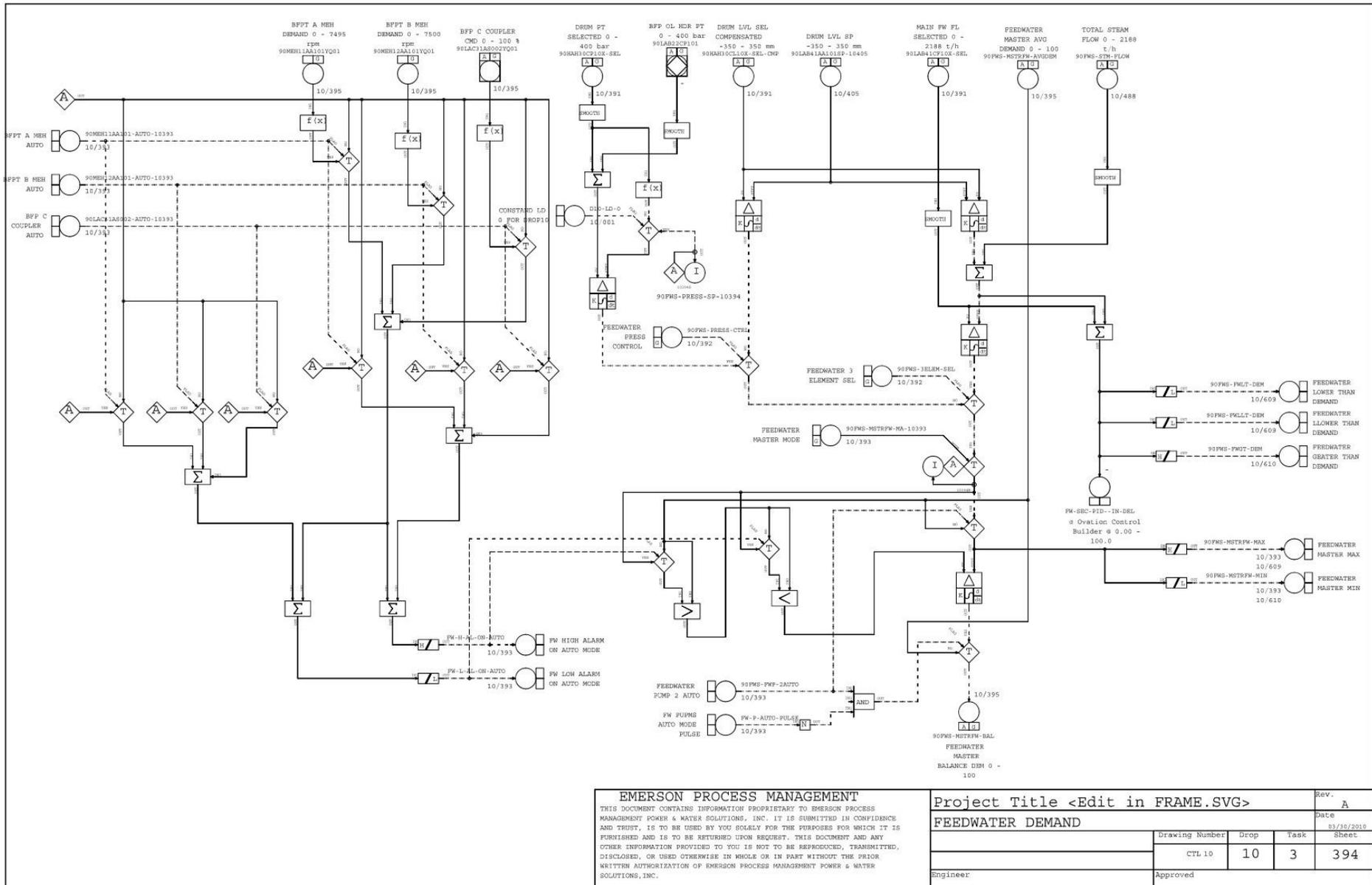
```

rule = [ 7 7 7 5 4 3 1 ;
        7 7 5 5 4 3 2 ;
        7 6 5 7 2 3 2 ;
        3 4 6 1 6 4 3 ;
        2 3 2 7 5 6 7 ;
        2 3 4 5 5 7 7 ;
        1 3 4 5 7 7 7 ];
for i=1:7
    for j=1:7
        n = rule(i,j);
        u(n)=max(u(n),min(mf1(i),mf2(j)));
    end
end

c = [0 1/6 1/3 1/2 2/3 5/6 1];
y = (sum(u.*c))/(sum(u));

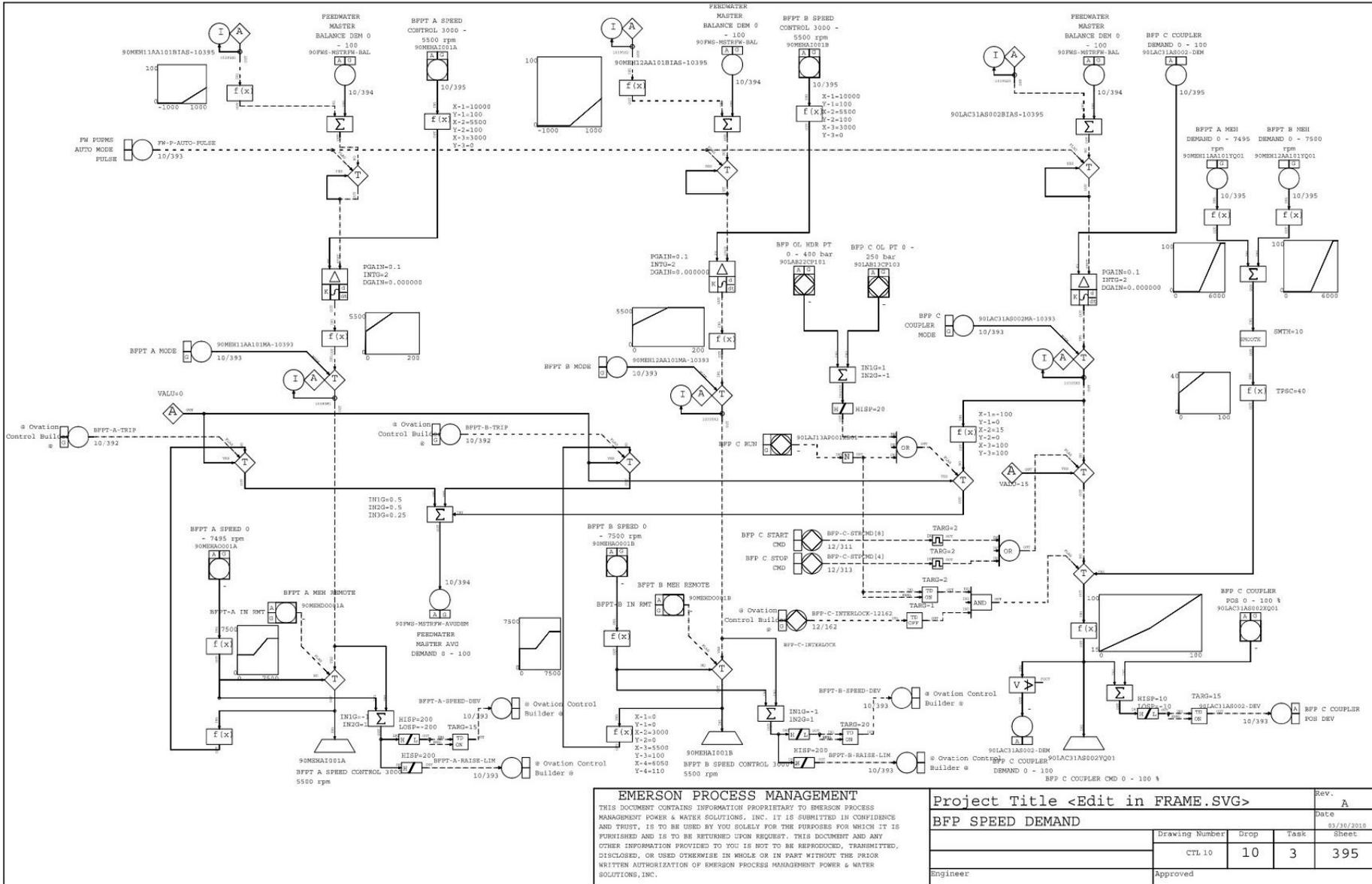
```

**Lampiran 11. SAMA Logic Diagram untuk Feedwater Demand**



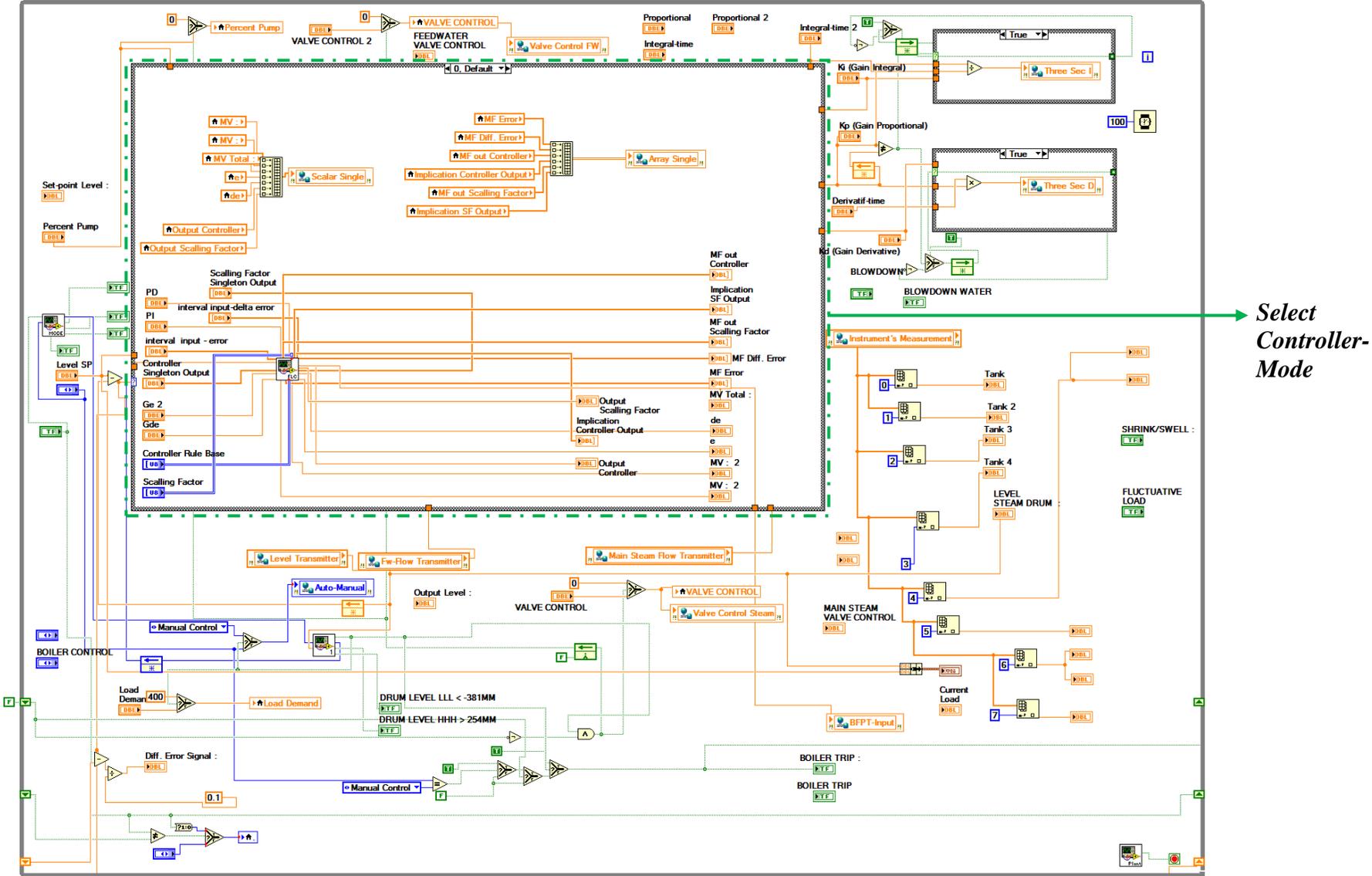
*Halaman ini sengaja dikosongkan*

Lampiran 12. SAMA Logic Diagram untuk BFP Speed Demand



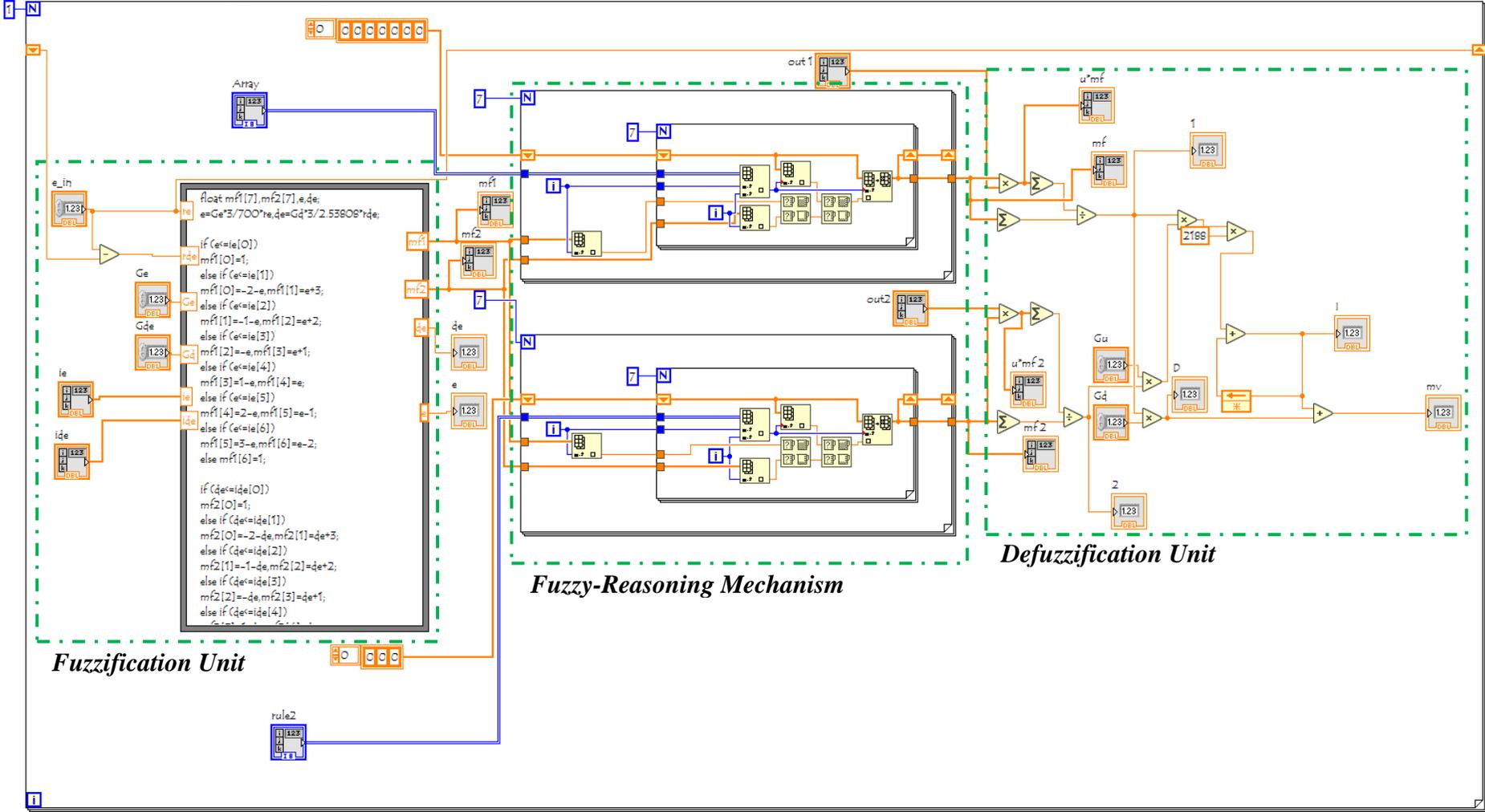
*Halaman ini sengaja dikosongkan*

Lampiran 13. Diagram Blok Utama Simulator PLTU pada Program LabVIEW

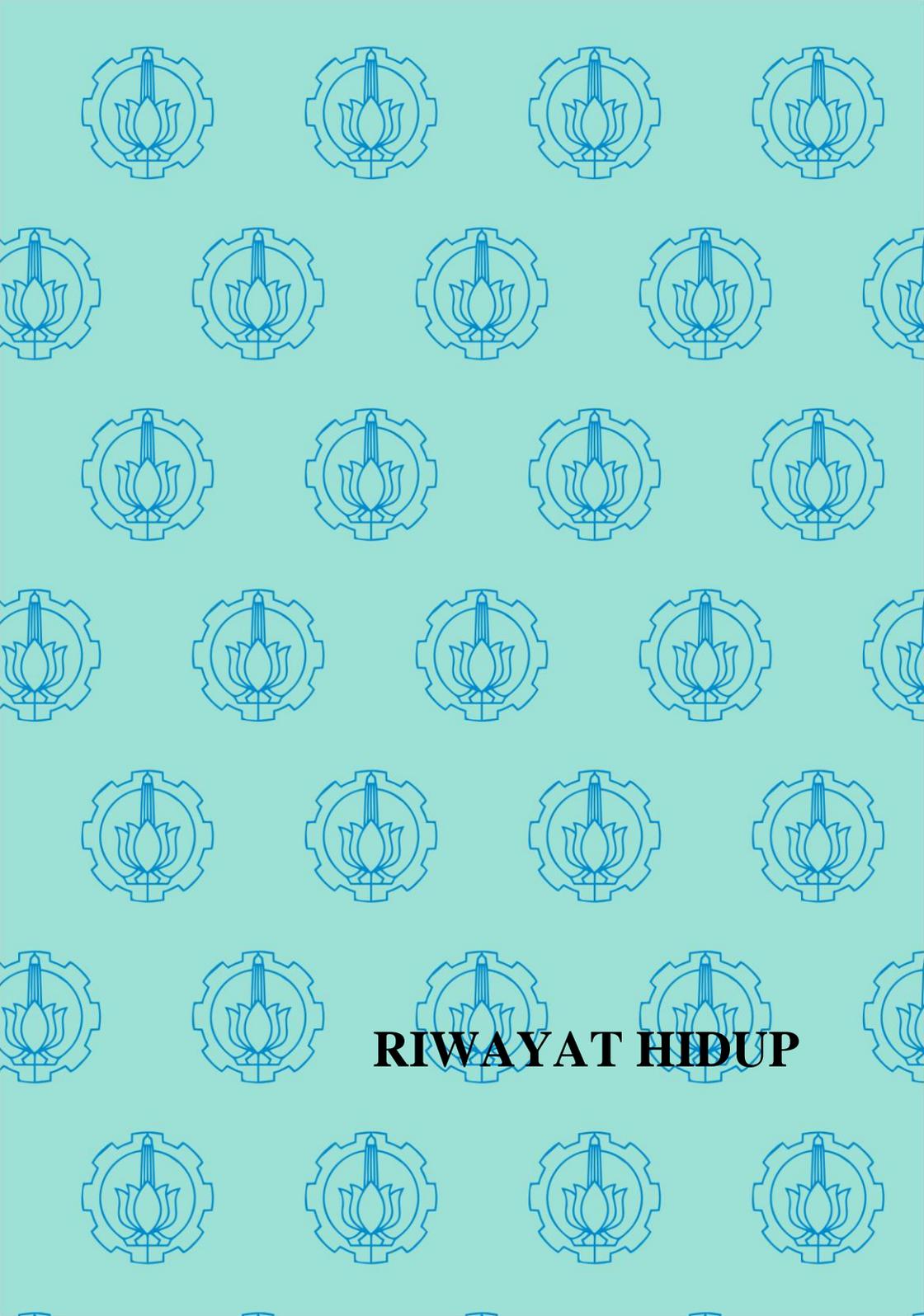


*Halaman ini sengaja dikosongkan*

**Lampiran 14.** Diagram Blok Kontroler Logika Fuzzy tipe-PID Robust Self-Tuning pada Program LabVIEW



*Halaman ini sengaja dikosongkan*



# **RIWAYAT HIDUP**

## RIWAYAT PENULIS



Penulis, dengan nama lengkap Mughny Indra Darmawan merupakan mahasiswa tahun / angkatan 2015 Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember yang lahir di Lumajang, pada tanggal 24 September 1997 sebagai putra bungsu dari 3 bersaudara. Alhamdulillah penulis oleh Allah SWT telah dianugerahi seorang ayah bernama Balok Chafid serta ibu bernama Wiwik Rukminarti yang mampu membawa penulis hingga menempuh jenjang perkuliahan ini dengan segala upaya telah yang dicurahkan. Untuk itu, penulis berharap penuh agar dapat membalas segala anugerah dan upaya yang telah penulis terima. Selebihnya lagi, penulis sangat berharap untuk bisa selalu dapat memberikan ilmu yang bermanfaat bagi diri sendiri, lingkungan, ataupun masyarakat serta senantiasa mendapatkan ridho dari Allah SWT.

*E-mail:* [mughny.id@gmail.com](mailto:mughny.id@gmail.com)