



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE184801

**DESAIN KONTROLER *ADAPTIVE PID MRAC* UNTUK
KONTROL KECEPATAN PADA *STEAM TURBINE*
DENGAN MENGGUNAKAN *PID CONTROLLER PLC*
*SIEMENS S7 300***

Mochammad Oza Alfiansyah
NRP 07111540000128

Dosen Pembimbing
Ir. Ali Fatoni, MT.

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



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**DESAIN KONTROLER *ADAPTIVE PID MRAC* UNTUK
KONTROL KECEPATAN PADA *STEAM TURBINE*
DENGAN MENGGUNAKAN *PID CONTROLLER PLC*
*SIEMENS S7 300***

Mochammad Oza Alfiansyah
NRP 07111540000128

Dosen Pembimbing
Ir. Ali Fatoni, MT.

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



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EE 184801

***ADAPTIVE PID MRAC CONTROLLER DESIGN FOR
SPEED CONTROL ON STEAM TURBINE USING PID
CONTROLLER PLC SIEMENS S7 300***

Mochammad Oza Alfiansyah
NRP 07111540000128

Supervisor
Ir. Ali Fatoni, MT.

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

PERNYATAAN KEASLIAN TUGAS AKHIR

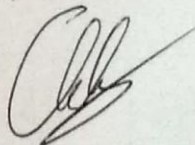
Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

“Desain Kontroler Adaptive PID MRAC untuk Kontrol Kecepatan pada Steam Turbine dengan Menggunakan PID Controller PLC Siemens S7 300”

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, Juni 2019



Mochammad Oza Alfiansyah

NRP 07111540000128

Halaman ini sengaja dikosongkan

**DESAIN KONTROLER ADAPTIVE PID MRAC
UNTUK KONTROL KECEPATAN PADA STEAM
TURBINE DENGAN MENGGUNAKAN PID
CONTROLLER PLC SIEMENS S7 300**

TUGAS AKHIR

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

Menyetujui:

Dosen Pembimbing


Ir. Ali Fafoni, MT.
NIP. 196206031989031002



DESAIN KONTROLER ADAPTIVE PID MRAC UNTUK KONTROL KECEPATAN PADA STEAM TURBINE DENGAN MENGGUNAKAN PID CONTROLLER PLC SIEMENS S7 300

Nama : Mochammad Oza Alfiansyah
NRP : 0711154000128
Dosen Pembimbing I : Ir. Ali Fatoni, MT.

ABSTRAK

Steam turbine tandem compound merupakan salah satu komponen utama yang ada dalam PLTU. Terdapat beberapa elemen penting yang perlu dikontrol pada *steam turbine* agar dapat bekerja sesuai dengan seharusnya, salah satunya adalah kecepatan putar turbin. Dalam kondisi tanpa kontroler, respon kecepatan yang dihasilkan oleh *steam turbine* memiliki bentuk respon integrator yang cenderung menuju nilai tak terhingga. Pada tugas akhir ini, dirancang sebuah kontroler adaptif berupa *PID MRAC* dengan skema *basic MIT Rule* dan *normalized MIT Rule* yang digunakan untuk mengontrol kecepatan *steam turbine* agar tetap mengikuti *set point* yang ditentukan walaupun terdapat beberapa gangguan seperti pembebanan kecepatan pada turbin dan juga perubahan kondisi turbin yang mengakibatkan parameter sistem berubah. Pada kontroler *adaptive PID MRAC*, terdapat bagian *adjustment mechanism* yang akan melakukan penyesuaian besar parameter kontroler tiap waktu agar respon dari *steam turbine* tetap sama dengan respon referensi yang sudah ditentukan. Parameter kontrol ini akan dikirimkan pada kontroler PID yang terdapat pada *PLC Siemens S7 300* yang berperan sebagai kontroler utama pada skema *adaptive PID MRAC* sedangkan untuk *adjustment mechanism* disusun dalam *software* pemrograman *LabVIEW*. Hasil pengujian skema *basic* dan *normalized MIT Rule* pada kondisi tanpa beban secara berturut-turut memberikan nilai *RMSE* 0.2552 dan 0.2521 sedangkan pada kondisi berbeban nilai *RMSE* yang dihasilkan 0.2747 dan 0.2586.

Kata Kunci: *Adaptive PID MRAC, basic MIT Rule, normalized MIT Rule*, parameter kontroler

Halaman ini sengaja dikosongkan

ADAPTIVE PID MRAC CONTROLLER DESIGN FOR SPEED CONTROL ON STEAM TURBINE USING PID CONTROLLER PLC SIEMENS S7 300

Nama : Mochammad Oza Alfiansyah
NRP : 07111540000128
Dosen Pembimbing I : Ir. Ali Fatoni, MT.

ABSTRACT

Steam turbine tandem compound is one kind of main component that needed in steam power plant. There have some important elements that must be controlled in steam turbine so it can work like it must be, especially the rotation speed of steam turbine. When the controller is not applied in the steam turbine plant, the speed response generate by steam turbine has the form of integrator's response which tend towards an infinite value. In this final project, an adaptive controller in the form of PID MRAC with basic MIT Rule and normalized MIT Rule are designed that used to control the speed of the steam turbine so it can keep up with the specific set point value although there are some disturbances such as load that applied on the steam turbine and also changing turbine conditions which result in system parameters changing. In the adaptive PID MRAC controller, there is an adjustment mechanism that will adjust the value of controller parameters, so the steam turbine response remains the same as the specified model reference response. This controller parameters will be sent to the PID controller that contained in the PLC Siemens S7 300 which acts as the main controller in the adaptive PID MRAC control scheme while for adjustment mechanism is compiled in LabVIEW programming software. The result of basic and normalized MIT Rule scheme test in no-load condition respectively give the RMSE values 0.2552 and 0.2521 while in different conditions the RMSE values generated are 0.2747 and 0.2586.

Keyword : *Adaptive PID MRAC, basic MIT Rule, normalized MIT Rule, controller parameters*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul “**Desain Kontroler Adaptive PID MRAC untuk Kontrol Kecepatan pada Steam Turbine dengan Menggunakan PID Controller PLC Siemens S7 300**” guna memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan Departemen Teknik Elektro Fakultas Teknologi Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis juga mengucapkan terimakasih kepada pihak-pihak yang berjasa dalam pengerjaan tugas akhir ini.

Ucapan terimakasih juga tak lupa diberikan kepada seluruh pihak-pihak yang membantu untuk terus menyemangati penulis sehingga penulis dapat menyelesaikan laporan penelitian ini. Khususnya dalam menyelesaikan buku Tugas Akhir ini, Ayah dan Ibu yang selalu melimpahkan doa-doa dan dukungan moril, Bapak Ir. Ali Fatoni, MT, selaku dosen pembimbing yang dengan sabar membimbing penulis walaupun penulis sering melakukan banyak kesalahan. Dosen Bidang Studi Teknik Sistem Pengaturan, Dosen Laboratorium Teknik Pengaturan AJ-104, dan juga Dosen Departemen Teknik Elektro atas pendidikan dan ilmu yang telah diberikan. Kawan-kawan seperjuangan ITS SMANBA 2015, UKM Robotika ITS, Tim Robot VI-ROSE, keluarga besar Laboratorium Teknik Pengaturan AJ-104, dan keluarga besar Teknik Sistem Pengaturan Laboratorium B-405, B105, dan B106 yang banyak memberikan semangat dan support kepada penulis. Terlebih kepada kawan seperjuangan laboratorium Teknik Pengaturan, Ahmad Athfi Noor L selaku mentor penulis dan Mughny Indra Darmawan yang telah memberikan banyak support dan pengalaman berharga. Seluruh sahabat dan teman-teman yang tidak dapat penulis sebutkan satu persatu.

Laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Semoga buku laporan Tugas Akhir ini dapat memberikan manfaat bagi pembaca sebagai acuan penelitian selanjutnya.

Surabaya, 20 Mei 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	vii
LEMBAR PENGESAHAN	x
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	2
1.2. Perumusan Masalah	4
1.3. Batasan Masalah	4
1.4 Tujuan	5
1.5 Metodologi	5
1.6 Sistematika Penulisan	5
1.7 Relevansi	7
BAB 2 TEORI PENUNJANG	9
2.1 Steam Turbine Generator[5]	9
2.1.1 Pengantar	9
2.1.2 Prinsip Kerja	9
2.1.3 Tipe <i>Steam Turbine</i>	14
2.1.4 Model Matematis <i>Steam Turbine Tandem Compound</i> [10]	17
2.1.5 Parameter Sistem Menggunakan <i>Data Heat Balance Steam Turbine Tandem Compound 200-MW</i> [12]	23
2.2 Model-Reference Adaptive Systems[2]	34
2.2.1 Pengenalan	34
2.2.2 <i>MIT Rule</i> [2]	35
2.2.3 <i>Normalized MIT Rule</i> [2]	37
2.3 Kontroler PID (Proportional-Integral-Derivative)[7]	39
2.3.1 Pengenalan	39
2.3.2 Prinsip Umpan Balik	40
2.3.3 Struktur Kontroler PID	41

2.4 Software Pemrograman <i>LabVIEW</i> [4].....	46
2.4.1 Pengenalan.....	46
2.4.2 <i>Control System Analysis, Design, and Simulation</i> [4].....	49
2.5 <i>Programmable Logic Controller</i> [8]	49
2.5.1 Pengenalan.....	49
2.5.2 PLC Siemens S7-300[13]	50
BAB 3 PERANCANGAN SISTEM	55
3.1 Perancangan <i>Plant Steam Turbine Tandem Compound</i> Pengaturan Kecepatan	55
3.1.1 Menentukan <i>Transfer Function Plant</i> Secara Keseluruhan.....	55
3.1.2 Penentuan Parameter Plant	58
3.1.3 Penyederhanaan Bentuk <i>Transfer Function Plant</i> Menjadi Bentuk <i>Plant Orde-2</i>	59
3.2 Penentuan <i>Adjusment Mechanism</i> pada kontroler <i>PID MRAC</i>	63
3.2.1 General MIT Rule.....	64
3.2.2 <i>Normalized MIT Rule (Modified MIT Rule)</i>	69
3.3 Penentuan <i>Model Reference</i> Untuk Kontroler <i>Adaptive PID</i> <i>MRAC</i>	70
3.4 Penyusunan Diagram Blok Sistem Secara Keseluruhan	75
3.4.1 Diagram blok <i>Adaptive PID MRAC</i> dengan <i>basic MIT</i> <i>Rule</i>	76
3.4.2 Diagram blok <i>Adaptive PID MRAC</i> dengan <i>normalized</i> <i>MIT Rule</i>	77
3.5 Perancangan Skema Kontrol <i>Adaptive PID MRAC</i> Secara Keseluruhan dan Integrasi antara <i>LabVIEW</i> dengan <i>PLC</i> <i>Siemens S7 300</i>	79
3.5.1 Perancangan Blok Simulasi pada <i>LabVIEW</i>	81
3.5.2 Perancangan Blok Simulasi <i>LabVIEW</i> dan Implementasi pada <i>PLC Siemens S7 300</i>	84
BAB 4 PENGUJIAN DAN ANALISIS	101
4.1 Pengujian Menggunakan <i>Gain</i> dan <i>Input</i> yang Berbeda-beda dengan Parameter Sistem yang Konstan	101

4.2 Pengujian Menggunakan <i>Input Sinyal Step</i> Bertingkat dengan Parameter Sistem Berubah-ubah dan dalam Kondisi Berbeban	126
4.3 Pengujian Menggunakan <i>PLC Siemens S7 300</i> dengan <i>Input Sinyal Step</i> Bertingkat Kondisi Parameter Sistem Berubah-ubah	135
BAB 5 KESIMPULAN	149
1.1. Kesimpulan	149
1.2. Saran.....	150
DAFTAR PUSTAKA	151
LAMPIRAN	153
Lampiran 1. <i>Adaptive PID MRAC LabVIEW Basic MIT Rule VI</i> untuk Simulasi tanpa <i>PLC</i>	153
Lampiran 2. <i>Adaptive PID MRAC LabVIEW Normalized MIT Rule VI</i> untuk Simulasi tanpa <i>PLC</i>	154
Lampiran 3. <i>Adjustment Mechanism VI Basic MIT Rule</i> untuk Simulasi dengan <i>PLC Siemens S7 300</i>	155
Lampiran 4. <i>Adjustment Mechanism VI Normalized MIT Rule</i> untuk Simulasi dengan <i>PLC Siemens S7 300</i>	156
Lampiran 5. <i>Steam Turbine Virtual Plant VI</i>	157
Lampiran 6. Perancangan dan Integrasi Sistem Secara Keseluruhan.....	158
Lampiran 7. <i>HMI Skema Kontrol Adaptive PID MRAC</i>	160
Lampiran 8. <i>HMI Virtual Plant Steam Turbine Tandem Compound</i> dengan <i>Single Re-Heater</i>	161
RIWAYAT HIDUP PENULIS	163

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Struktur sederhana <i>impulse turbine stages</i> [5]	10
Gambar 2. 2 Struktur <i>velocity compounded</i> dan <i>pressure compounded turbine</i> [5]	12
Gambar 2. 3 Struktur sederhana <i>reaction turbine stages</i> [5].....	13
Gambar 2. 4 Penampang <i>magnetic circuit</i> pada <i>generator</i> [5].....	14
Gambar 2. 5 Tipe-tipe <i>steam turbine</i> [5].....	15
Gambar 2. 6 Struktur <i>tandem compound steam turbine</i> [5]	16
Gambar 2. 7 Struktur <i>cross compound steam turbine</i> [5]	17
Gambar 2. 8 <i>Free body diagram</i> pada <i>unit steam turbine</i> [10]	17
Gambar 2. 9 Ilustrasi <i>continuity equation</i> pada fluida[6].....	18
Gambar 2. 10 Kofigurasi <i>steam turbine</i> tipe <i>tandem compound</i> [10]	21
Gambar 2. 11 Diagram blok <i>steam turbine tandem compound</i> [10]	22
Gambar 2. 12 Diagram <i>steam vessel</i> [12]	24
Gambar 2. 13 <i>Heat balance map</i> pada <i>steam turbine tandem compound 200-MW</i> [12].....	27
Gambar 2. 14 Konfigurasi saluran pipa <i>reheater</i> [12]	30
Gambar 2. 15 Konfigurasi saluran pipa <i>crossover</i> [12]	31
Gambar 2. 16 Diagram blok dari <i>Model-Reference Adaptive System (MRAS)</i> [2].....	34
Gambar 2. 17 Respon sistem dengan <i>input</i> sinyal persegi amplitudo (a) 0.1, (b) 1, (c) 3.5 dan $\gamma = 0.1$ [2].....	37
Gambar 2. 18 Respon sistem dengan <i>input</i> sinyal persegi amplitudo (a) 0.1, (b) 1, (c) 3.5, $\alpha = 0.001$, dan $\gamma = 0.1$ [2]	39
Gambar 2. 19 Diagram blok sistem dengan umpan balik <i>negative</i> [7] ..	41
Gambar 2. 20 Simulasi <i>closed-loop system</i> menggunakan kontroler proporsional[7].....	42
Gambar 2. 21 Implementasi aksi integral sebagai <i>feedback</i> positif disekitar <i>lag</i> [7]	43
Gambar 2. 22 Simulasi <i>closed-loop system</i> menggunakan kontroler integral, pada diagram yang pertama menunjukkan respon dari sistem sedangkan untuk diagram kedua menunjukkan sinyal kontrol[7]	44
Gambar 2. 23 Interpretasi aksi derivatif sebagai kontrol prediksi menggunakan ekstrapolasi linier[7]	45

Gambar 2. 24 Simulasi <i>closed-loop system</i> menggunakan kontroler derivatif, pada diagram yang pertama menunjukkan respon dari sistem sedangkan untuk diagram kedua menunjukkan sinyal kontrol[7].....	45
Gambar 2. 25 Tampilan <i>Front Panel window</i> pada <i>LabVIEW</i> [4].....	47
Gambar 2. 26 Tampilan <i>Block Diagram window</i> pada <i>LabVIEW</i> [4].....	47
Gambar 2. 27 Tampilan <i>Controls Palette</i> pada <i>Front Panel LabVIEW</i> [4].....	48
Gambar 2. 28 Tampilan <i>Functions Palette</i> pada <i>Block Diagram LabVIEW</i> [4].....	48
Gambar 2. 29 Diagram konsep aplikasi <i>PLC</i> [8].....	50
Gambar 2. 30 Elemen penyusun <i>CPU</i> tipe 31_C[13].....	50
Gambar 2. 31 <i>Wiring Diagram</i> modul SM 334 AI 4/AO 2 x 12	53
Gambar 3. 1 Diagram blok pengaturan kecepatan <i>steam turbine</i> [11]....	56
Gambar 3. 2 Diagram blok <i>steam turbine</i> secara keseluruhan.....	57
Gambar 3. 3 Grafik <i>pole-zero plot</i> pada <i>plant steam turbine</i>	62
Gambar 3. 4 Grafik <i>plot uji step response</i> pada <i>transfer function</i> baru	63
Gambar 3. 5 Grafik <i>plot uji step response</i> pada <i>steam turbine tandem compound</i>	71
Gambar 3. 6 Grafik <i>plot uji Ziegler-Nichols</i>	72
Gambar 3. 7 Grafik <i>plot uji step response</i> pada <i>plant steam turbine orde-2</i> dengan kontroler <i>PID</i> hasil <i>tunning Zigler-Nichols</i>	73
Gambar 3. 8 Grafik <i>plot uji step response</i> pada <i>plant steam turbine orde-2</i> dengan kontroler <i>PID</i> hasil <i>tunning trial error</i>	74
Gambar 3. 9 Diagram blok sistem yang digunakan sebagai <i>model reference</i>	75
Gambar 3. 10 Diagram blok skema <i>adaptive PID MRAC</i> secara umum.....	80
Gambar 3. 11 Diagram blok <i>model reference</i> pada <i>adaptive PID MRAC LabVIEW VI</i>	81
Gambar 3. 12 Diagram blok <i>steam turbine tandem compound</i> pada <i>adaptive PID MRAC LabVIEW VI</i>	81
Gambar 3. 13 Diagram blok <i>adjustment mechanism</i> pada <i>adaptive PID MRAC LabVIEW VI</i>	82
Gambar 3. 14 Diagram blok <i>model reference</i> pada <i>normalized adaptive PID MRAC LabVIEW VI</i>	83
Gambar 3. 15 Diagram blok <i>steam turbine tandem compound</i> pada <i>normalized adaptive PID MRAC LabVIEW VI</i>	83

Gambar 3. 16 Diagram blok <i>adjustment mechanism</i> pada <i>normalized adaptive PID MRAC LabVIEW VI</i>	83
Gambar 3. 17 Diagram blok <i>adjustment mechanism, model reference,</i> dan variabel yang di- <i>share</i> pada <i>adaptive PID MRAC adjustment mechanism VI</i>	84
Gambar 3. 18 Diagram blok <i>adjustment mechanism, model reference,</i> dan variabel yang di- <i>share</i> pada <i>normalized adaptive PID MRAC adjustment mechanism VI</i>	85
Gambar 3. 19 Diagram blok persamaan beda $H_{SpeedRelay}$	88
Gambar 3. 20 Diagram blok persamaan beda $H_{ServoMotor}$	88
Gambar 3. 21 Diagram blok persamaan beda $H_{HighPressure}$	89
Gambar 3. 22 Diagram blok persamaan beda $H_{MediumPressure}$	90
Gambar 3. 23 Diagram blok persamaan beda $H_{LowPressure}$	91
Gambar 3. 24 Diagram blok persamaan beda H_{Load}	91
Gambar 3. 25 Variable yang di- <i>share</i> menggunakan fitur <i>shared variable</i>	92
Gambar 3. 26 Halaman <i>project</i> pada <i>SIMATIC Manager</i>	93
Gambar 3. 27 Tampilan <i>setting</i> waktu <i>cycle</i> pada <i>cyclic interrupt</i>	96
Gambar 3. 28 Tampilan <i>IBH OPC Editor</i> beserta variabel yang dihubungkan.....	98
Gambar 3. 29 Konfigurasi variabel K_p pada <i>IBH OPC Server</i>	99
Gambar 3. 30 Konfigurasi variabel T_i pada <i>IBH OPC Server</i>	99
Gambar 3. 31 Konfigurasi variabel T_d pada <i>IBH OPC Server</i>	100
Gambar 3. 32 Konfigurasi variabel M_{an} pada <i>IBH OPC Server</i>	100
Gambar 4. 1 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 1	102
Gambar 4. 2 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 1	103
Gambar 4. 3 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 1	104
Gambar 4. 4 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 1	105
Gambar 4. 5 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3	106
Gambar 4. 6 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3	107

Gambar 4. 7 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3.....	109
Gambar 4. 8 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3	109
Gambar 4. 9 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5.....	111
Gambar 4. 10 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5	112
Gambar 4. 11 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5	113
Gambar 4. 12 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5	114
Gambar 4. 13 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3.....	115
Gambar 4. 14 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3	116
Gambar 4. 15 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3.....	117
Gambar 4. 16 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3	118
Gambar 4. 17 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5.....	119
Gambar 4. 18 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5	120
Gambar 4. 19 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5.....	121
Gambar 4. 20 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5	122
Gambar 4. 21 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5.....	123
Gambar 4. 22 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5	124
Gambar 4. 23 Variasi beban kecepatan yang diberikan pada pengujian kontroler <i>PID MRAC</i> dengan <i>basic MIT Rule</i>	127

Gambar 4. 24 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5 ketika diberi pembebanan dan perubahan parameter	128
Gambar 4. 25 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 5 ketika diberi pembebanan dan perubahan parameter	130
Gambar 4. 26 Variasi beban kecepatan yang diberikan pada pengujian kontroler <i>PID MRAC</i> dengan <i>normalized MIT Rule</i>	131
Gambar 4. 27 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5 ketika diberi pembebanan dan perubahan parameter	132
Gambar 4. 28 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 5 ketika diberi pembebanan dan perubahan parameter	134
Gambar 4. 29 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3 ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	137
Gambar 4. 30 Grafik perubahan parameter kontroler skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3 ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	138
Gambar 4. 31 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3 ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	139
Gambar 4. 32 Grafik perubahan parameter kontroler skema <i>normalized MIT Rule</i> dengan <i>adaptation gain</i> = 3 ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	140
Gambar 4. 33 Variasi beban kecepatan yang diberikan pada pengujian kontroler <i>PID MRAC</i> dengan <i>basic MIT Rule</i> ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	141
Gambar 4. 34 Variasi beban kecepatan yang diberikan pada pengujian kontroler <i>PID MRAC</i> dengan <i>normalized MIT Rule</i> ketika diimplementasikan pada <i>PLC Siemens S7 300</i>	142
Gambar 4. 35 Respon <i>plant</i> dan <i>error</i> antara <i>model reference</i> dan <i>plant</i> skema <i>basic MIT Rule</i> dengan <i>adaptation gain</i> = 3 ketika diimplementasikan pada <i>PLC Siemens S7 300</i> dalam kondisi berbeban	143

Gambar 4. 36 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300* dalam kondisi berbeban 145

Gambar 4. 37 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300* dalam kondisi berbeban 146

Gambar 4. 38 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300* dalam kondisi berbeban 147

DAFTAR TABEL

Tabel 2. 1 Bagian-bagian <i>CPU 31_C series</i>	51
Tabel 2. 2 Daftar status pada <i>LED indicators PLC Siemens</i> seri <i>CPU 31_C</i>	51
Tabel 3. 1 Parameter <i>steam turbine tandem compound</i>	58
Tabel 3. 2 Parameter pada <i>governor</i> dan <i>load</i> menggunakan data <i>heat balance</i>	58
Tabel 3. 3 Parameter <i>turbine sections</i> menggunakan data <i>heat balance</i>	58
Tabel 4. 1 Parameter yang digunakan pada pengujian tanpa beban	101
Tabel 4. 2 Parameter yang digunakan pada pengujian berbeban	126

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Latar belakang dari dilakukannya penelitian adalah untuk mencoba menerapkan metode kontrol adaptif yang berupa *PID Model Reference Adaptive Controller* pada *virtual plant steam turbine* pengaturan kecepatan. Pada penelitian, akan dilakukan desain adjustment mechanism menggunakan skema MIT Rule dengan bahasa pemrograman *LabVIEW* guna melakukan tuning pada parameter kontroler PID (K_p , τ_i , dan τ_d) agar mendapatkan respons output yang sesuai dengan model ideal yang digunakan walaupun terdapat gangguan maupun terdapat perubahan nilai parameter pada *steam turbine*. Selain itu, pada penelitian ini akan dilakukan implementasi parameter kontroler yang sudah diperoleh dengan metode adaptif kedalam *function block PLC Siemens S7-300* guna melihat apakah skema kontroler *Adaptive PID MRAC* yang didesain dengan menggunakan bahasa pemrograman *LabVIEW* dapat dihubungkan dengan kontroler *PID* pada *PLC* sehingga dapat melakukan kontrol pada kecepatan *virtual plant steam turbine* sesuai dengan simulasi yang dilakukan. Pada penelitian ini akan digunakan *steam turbine* bertipe *multistage tandem compound* dengan *single reheater*. Untuk parameter-parameter *plant* yang digunakan dibatasi pada penggunaan parameter dari penelitian pihak lain dimana akan terdapat dua jenis pendekatan yaitu dengan pendekatan menggunakan hukum fisis fluida pada *steam turbine* dan juga pendekatan dengan menggunakan data *Heat Balance* pada *steam turbine*. Dengan menggunakan skema *Adaptive PID MRAC* ini, diharapkan hasil respons output dari *plant steam turbine* tetap sesuai dengan respons model yang ada walaupun terdapat perubahan nilai parameter-parameter pada *plant* dikarenakan faktor-faktor teknis maupun faktor yang lain. Penelitian ini akan diawali dengan studi literatur, perancangan sistem, analisa respons dari sistem dan desain kontroler, pengujian dan analisa metode kontrol yang sudah diterapkan, kemudian diakhiri dengan kesimpulan.

1.1 Latar Belakang

Pada proses konversi energi potensial pada uap pada PLTU menjadi energi listrik yang siap didistribusikan memerlukan berbagai macam proses kontrol untuk menjaga agar energi listrik yang dihasilkan dapat memiliki spesifikasi sesuai dengan yang dibutuhkan. Salah satu hal yang perlu diterapkan metode kontrol adalah putaran dari *steam turbine* yang terhubung pada generator. Hal ini karena putaran turbin yang memutar rotor dari generator akan memiliki peranan langsung dalam penentuan frekuensi tegangan yang dihasilkan dimana frekuensi tegangan akan sebanding dengan kecepatan putar dari rotor[1]. Apabila putaran yang terjadi pada turbin tidak konstan, maka frekuensi tegangan yang dibangkitkan pada generator juga tidak akan konstan. Hal ini tidak sesuai dengan yang diinginkan dimana frekuensi tegangan yang dihasilkan oleh suatu generator idealnya adalah konstan dan sesuai dengan spesifikasi tertentu. Di Indonesia sendiri tegangan AC standard yang banyak digunakan memiliki frekuensi sebesar 50 Hz. Oleh karena itu, perlu diterapkannya kontroler yang digunakan untuk mengatur kecepatan putar pada *steam turbine* agar dapat menghasilkan frekuensi tegangan yang konstan.

Hal yang biasa dikontrol agar *steam turbine* dapat memiliki putaran sesuai dengan yang diinginkan adalah aliran *steam* yang mengalir masuk menuju turbin yang berasal dari *boiler*. Aliran dari *steam* tersebut dikontrol menggunakan *control valve* yang ada pada saluran masuk *steam turbine*. Pada mulanya, aliran uap banyak dikontrol menggunakan peralatan mekanik murni seperti *mechanical governor* dimana pada prinsipnya menggunakan putaran mesin yang dalam hal ini adalah *steam turbine* sebagai input yang akan mengatur intensitas aliran uap yang mengalir. *Mechanical governor* menggunakan prinsip gaya sentrifugal guna mengatur lebar buka tutup jalur yang dilewati oleh *steam*. Tetapi, dengan menggunakan *mechanical governor* seperti ini, cukup sulit untuk mendapatkan respons perubahan kecepatan seperti yang dikehendaki dikarenakan pada *mechanical governor* ini kontrol kecepatan lebih banyak diatur pada kondisi disekitar nilai yang diinginkan dimana ketika kecepatan *turbine* masih dibawah *setpoint* maka *governor* akan membuka *valve*, sedangkan bila kecepatan yang terukur sudah melebihi *setpoint*, maka gaya sentrifugal pada *governor* akan membuat *valve* menutup. Sedangkan untuk kondisi *transient* sulit untuk diatur karena bukaan *valve* yang diatur oleh *mechanical governor* dirasa kurang proporsional terdapat perbedaan kecepatan yang ada. Selain itu,

mechanical governor dinilai memiliki susunan komponen yang lebih kompleks dan banyak jika dibandingkan dengan sistem yang menggunakan sinyal elektrik. Oleh karena itu, pada *governor* sekarang biasanya menggunakan mikroprosesor yang dapat diprogram untuk mengontrol *control valve* agar memberikan bukaan *valve* yang dapat menghasilkan output putaran turbin sesuai dengan spesifikasi yang diinginkan. Kontroler yang biasa dipakai adalah kontroler PID karena kontroler ini memiliki konstruksi yang cukup sederhana dan juga memiliki kehandalan yang sudah teruji di dunia industri. Namun penentuan nilai parameter – parameter kontroler PID yang pas dinilai cukup sulit untuk diterapkan pada *plant* asli yang cenderung rumit dan dinamik. Oleh karena itu, perlu adanya metode khusus guna memudahkan kita untuk mengatur nilai parameter kontroler PID agar dapat memberikan respon sesuai dengan spesifikasi yang diinginkan.

Kontrol adaptif merupakan metode yang dipilih dalam tugas akhir ini untuk menyelesaikan persoalan kontrol kecepatan putaran *steam turbine*. Penggunaan metode kontrol adaptif ini digunakan untuk mengatasi perubahan parameter-parameter pada *plant steam turbine* yang diakibatkan oleh faktor-faktor tertentu. Apabila tanpa menggunakan metode adaptif, dikhawatirkan respons dari *plant* akan mengalami perubahan dikarenakan parameter kontroler yang digunakan sebelumnya di-*tuning* dengan parameter *plant* sebelum mengalami perubahan[2], Sedangkan dengan metode *Adaptive PID MRAC*, respons *output* dari *plant* akan terus dibandingkan dengan *model reference* yang ada dan perbedaan antara kedua nilai ini akan digunakan untuk menyesuaikan kembali nilai parameter kontroler PID agar dapat menghasilkan sinyal kontrol yang mengatur bukaan *valve* sehingga mendapatkan respons kecepatan *steam turbine* yang sesuai dengan kecepatan referensi model yang sudah ditetapkan. Proses penyusunan *Model Reference Adaptive Controller* ini akan dilakukan menggunakan *LabVIEW Programming Software* menggunakan skema *MIT Rule*. Parameter PID yang telah didapatkan nantinya akan dimasukkan kedalam *Function Block PID Controller* pada *PLC (Programmable Logic Controller) Siemens S7-300* yang kemudian akan dihubungkan ke *plant*. Input yang digunakan sebagai *reference* pada sistem berupa data kecepatan putar turbin yang diinginkan sedangkan output yang diperoleh dari *plant* dan akan dijadikan sebagai *feedback system* berupa output kecepatan putar *steam turbine*. Dengan data error berupa perbedaan kecepatan putar *steam turbine* dengan referensi kecepatan yang

diinginkan, nantinya diharapkan perbedaan dari kedua kecepatan ini sama dengan nol yang berarti putaran pada *steam turbine* sudah relatif konstan dan memiliki kecepatan putar yang sesuai dengan spesifikasi yang diinginkan.

1.2. Perumusan Masalah

Kecepatan putar yang tidak konstan pada *steam turbine* merupakan hal yang harus dihindari karena dapat menyebabkan frekuensi tegangan yang dihasilkan dapat tidak sesuai dengan spesifikasi yang sudah ditetapkan. Perubahan kecepatan putar ini bisa disebabkan oleh pengaruh *disturbance* maupun perubahan parameter-parameter *plant*. Oleh karena itu, diperlukan perancangan *adjustment mechanism* yang sesuai pada kontroler *Adaptive PID MRAC* sehingga kecepatan pada *steam turbine* dapat tetap terjaga konstan walaupun mengalami beberapa gangguan dan parameter-parameter pada *steam turbine*. Selain itu perlu adanya suatu protokol khusus guna menghubungkan kontroler yang dalam hal ini adalah *PID Controller Function Block PLC Siemens S7-300* dengan *adjustment mechanism* untuk skema adaptif *MRAC* pada *software* pemrograman *LabVIEW*.

1.3. Batasan Masalah

Pada penelitian ini jenis *plant* yang dipakai adalah *steam turbine tandem compound* dengan single reheater dan menggunakan nilai parameter sistem yang didapat dari dua sumber penelitian pihak lain yaitu pada [10] dan [12]. Selain itu, pada penelitian ini digunakan *PLC Siemens S7 300* yang mempunyai *CPU* berkode *312-5BD01-0AB0* dan modul *analog I/O* berjenis *SM 334 AI 4/AO 2 x 12 bit* yang digunakan sebagai kontroler pada skema *Adaptive PID MRAC*. Untuk penyusunan *adjustment mechanism PID MRAC* digunakan laptop yang sudah terinstall *LabVIEW 2016* yang dilengkapi dengan *Control & Design Simulation Module* dan untuk pertukaran data antara *PLC* dan *LabVIEW* digunakan *IBH OPC Server* yang memanfaatkan komunikasi *serial* melalui *USB* yang terhubung ke *PLC* menggunakan *USB MPI Adapter* dan *Arduino UNO Board* dimana *board Arduino* hanya digunakan sebagai *I/O interface*, bukan sebagai kontroler. Untuk sinyal yang digunakan sebagai input pada pengujian dibatasi antara 1-5 dikarenakan *Arduino UNO* hanya mampu mengolah sinyal analog 1-5 volt.

1.4 Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah mendesain kontroler *adaptif PID MRAC* untuk kontrol kecepatan putar steam turbine agar dapat stabil pada nilai yang diinginkan dan memiliki spesifikasi respon yang cenderung tetap walaupun terdapat perubahan nilai parameter dari plant. Selain itu, tugas akhir ini juga menunjukkan bagaimana cara mengimplementasikan kontroler *adaptive PID MRAC* pada *LabVIEW* dengan *hardware PLC Siemens S7-300* dan melihat kesesuaiannya. Diharapkan juga tugas akhir ini dapat dijadikan acuan untuk penelitian selanjutnya.

1.5 Metodologi

Mekanisme untuk penyusunan *adjustment mechanism* pada kontroler *adaptive PID MRAC* ini akan menggunakan metode *gradient MIT Rule*. Kemudian digunakan *software* pemrograman *LabVIEW* untuk menyusun *diagram block adjustment mechanism* dan juga *virtual plant* dari *Steam Turbine Tandem Compound* yang digunakan. Untuk koneksi antara *PLC* dan *LabVIEW* menggunakan dua media yaitu *OPC (OLE for Process Control) IBH Softec* dan *Arduino Uno*. Sedangkan untuk *transfer variable* antara *virtual plant Steam Turbine VI (Virtual Instrument)* dengan *adjustment mechanism VI* memanfaatkan fitur *shared variable* pada *LabVIEW*. Untuk tahap pengujian akan dilakukan dengan beberapa kondisi yang secara teori dapat diatasi dengan menggunakan kontrol adaptif *MRAC*.

Kemudian pada bagian akhir akan dilakukan penarikan kesimpulan untuk menunjukkan hubungan antara tujuan penelitian dengan hasil yang telah didapat. Dengan hasil tersebut, maka diharapkan dapat digunakan sebagai bahan untuk penelitian lebih lanjut mengenai tema yang serupa.

1.6 Sistematika Penulisan

Tahap terakhir dari sebuah penelitian adalah penulisan laporan. Pada penulisan laporan/buku tugas akhir ini disusun menjadi 5 bab yaitu:

BAB 1 : PENDAHULUAN

Dalam bab ini, akan dijelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan,

metodologi, dan sistematika penulisan yang digunakan untuk menyusun buku tugas akhir ini.

BAB 2 : TEORI PENUNJANG

Dalam bab ini, akan dijelaskan mengenai berbagai teori penunjang yang didapatkan dari sumber-sumber yang relevan untuk membantu menyelesaikan penelitian dalam tugas akhir ini. Teori-teori tersebut akan meliputi pengetahuan umum tentang *steam turbine*, skema kontroler *Model Reference Adaptive System*, kontroler *Proportional-Integral-Derivative*, dan studi literatur yang lain.

BAB 3 : PERANCANGAN SISTEM

Didalam bab ini, akan dijelaskan mengenai perancangan keseluruhan sistem yang akan digunakan yang dimulai dari menentukan model *transfer function* dari *steam turbine tandem compound* yang digunakan, kemudian dilanjutkan dengan melakukan perhitungan untuk mendapatkan *adjustment mechanism* yang digunakan pada kontroler *adaptive PID MRAC*, selanjutnya juga akan dibahas mengenai perancangan koneksi dan komunikasi antara PC yang digunakan dengan *PLC Siemens S7-300* menggunakan *OPC IBH Softec* dan *Arduino UNO*.

BAB 4 : PENGUJIAN DAN ANALISIS

Didalam bab ini, akan dijelaskan mengenai hasil simulasi metode *adaptive PID MRAC* pada *steam turbine* yang dilakukan dengan menggunakan *software* pemrograman *LabVIEW*. Selain itu juga akan ditunjukkan hasil yang didapatkan setelah metode pengaturan yang dibuat diimplementasikan kedalam *PLC Siemens S7-300*. Kedua pengujian ini juga akan dilakukan dengan beberapa skenario pengujian untuk melihat apakah metode pengaturan *adaptive PID MRAC* yang disusun sudah dapat memberikan hasil sesuai dengan yang diinginkan.

BAB 5 : PENUTUP

Didalam bab ini, akan dituliskan mengenai kesimpulan yang dapat diambil setelah melakukan berbagai macam pengujian dan akan diberikan

beberapa saran untuk melakukan pengembangan terhadap penelitian tugas akhir ini.

1.7 Relevansi

Dengan dirancangnya skema kontrol *adaptive PID MRAC* didalam *software* pemrograman *LabVIEW* yang kemudian diterapkan kedalam *function block PID* pada PLC Siemens S7-300 diharapkan dapat dijadikan sebai referensi pada dunia industri agar PLC yang ada dapat dimanfaatkan tidak hanya sebagai kontroler berbasis logika, tetapi juga sebagai kontroler dalam sistem pengaturan proses yang memiliki kemampuan adaptasi di dunia industri.

Halaman ini sengaja dikosongkan

BAB 2

TEORI PENUNJANG

2.1 Steam Turbine Generator[5]

2.1.1 Pengantar

Steam turbine dan *generator* merupakan perangkat utama yang digunakan untuk konversi energi pada *steam power plant*. Fungsi utama dari *steam turbine generator* adalah untuk mengubah energi *thermal* dari uap yang dihasilkan oleh *steam generator* menjadi energi listrik. Untuk melakukan tugas ini, terdapat dua bagian utama yaitu *steam turbine* untuk mengubah energi *thermal* menjadi energi putar mekanis dan bagian *generator* yang digunakan untuk mengubah energi rotasi mekanis yang dihasilkan oleh turbin menjadi energi listrik. Biasanya, poros pada *steam turbine* langsung terkopel dengan poros pada *generator* tanpa melalui *gearbox*.

Produsen *steam turbine* utama ukuran *medium to large* di *United States* adalah *General Electric* dan *Westinghouse*. Kemudian untuk di sektor Eropa, produsen utama *steam turbine* yang juga melakukan ekspor ke *United States* adalah *ASEA Brown Boveri*, *Siemens*, *MAN*, dan *GEC Alsthom*. Selanjutnya untuk di sektor Asia terutama Jepang, produsen utama dari *steam turbine* adalah *Hitachi*, *Mitsubishi*, *Toshiba*, dan *Fuji Electric*.

2.1.2 Prinsip Kerja

Prinsip kerja dari *steam turbine generator* melibatkan ekspansi dari *steam* yang masuk menuju beberapa *stages* didalam *steam turbine*, sehingga menyebabkan *rotor* pada turbin memutar *rotor* pada *generator*. Dikarenakan kumparan *rotor* pada *generator* mengandung medan magnet, maka putaran tersebut akan membangkitkan energi listrik pada sisi kumparan *stator* di *generator*.

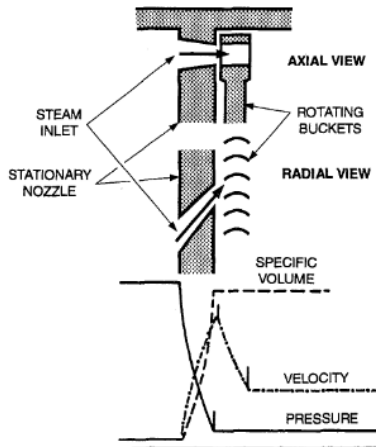
2.1.2.1 Tipe stage pada steam turbine

Energi *thermal* pada *steam* diubah menjadi energi mekanik dengan cara mengekspansi uap yang masuk kedalam turbin. Ekspansi uap pada turbin dapat terjadi pada dua tipe *stage* yaitu *impulse stage* dan *reaction stage*. *Impulse stage* dapat dimisalkan seperti *water wheel* yang akan berputar apabila aliran air mendorong pedal pada *water wheel* tersebut.

Sedangkan untuk *reaction stage* dapat dimisalkan seperti alat penyiram air yang berputar (*rotating sprinkler*) dimana tekanan air yang keluar dari kedua jalur keluar yang berlawanan arah pada *sprinkler* menyebabkan lengan pada penyiram berputar.

- *Impulse Stages*

Didalam impulse stage terdapat *stationary nozzle* dengan bilah yang dapat berputar. Uap yang dilewatkan dan diekspansi melalui *nozzle* akan menyebabkan kecepatan uap meningkat dan tekanan pada uap menurun. Kemudian uap yang keluar dari *nozzle* tersebut akan menabrak bilah-bilah rotor (*rotating buckets*) dan akan menimbulkan gaya pada bilah rotor tersebut sehingga bilah tersebut dapat berputar. Setiap penabrakan ini akan menimbulkan penurunan kecepatan uap yang melaluinya. *Impulse stage* dapat dikelompokkan lagi menjadi *velocity compound stages* atau *pressure compound stages*. Gambar 2.1 menunjukkan penampang sederhana dari turbin bertipe *impulse*.

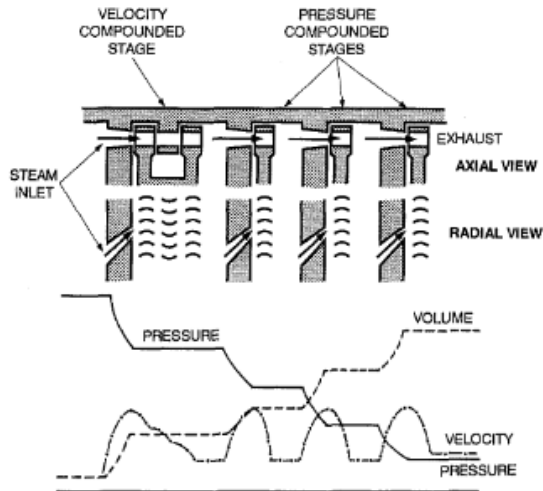


Gambar 2. 1 Struktur sederhana *impulse turbine stages*[5]

Velocity compound stages meliputi nozzle yang stasioner dan beberapa bilah yang berputar (*rotating blades/rotating buckets*) dan bilah yang diam (*stationary blades/stationary buckets*). Didalam nozzle terdapat pressure drop yang tinggi yang menghasilkan peningkatan pada

kecepatan uap yang dihasilkan. Pada set pertama rotating buckets, akan terjadi penurunan kecepatan pada uap akibat dari gaya yang diberikan pada bilah rotor. Uap kemudian akan lanjut mengalir menuju set stationary buckets sehingga arah aliran uap akan kembali seperti ketika sebelum memasuki set pertama dari rotating buckets. Selanjutnya, uap akan kembali dilewatkan menuju set kedua dari rotating buckets untuk memberikan gaya yang digunakan untuk memutar poros pada turbin. Kecepatan uap yang melalui beberapa set dari bilah rotor ini akan melalui kenaikan dan penurunan secara bergantian. Didalam velocity compound stage dapat terdiri dari stationary nozzles dan beberapa set rotating buckets dan stationary buckets. Akan tetapi, biasanya hanya terdapat dua set rotating bucket rows dan satu set stationary bucket row. Tipe velocity compound stage biasanya digunakan sebagai stage pertama pada turbin dikarenakan kemampuannya yang tahan terhadap reduksi tekanan yang tinggi dan efisiensi yang dihasilkan dalam mereduksi tekanan uap secara cepat dan meminimalkan persyaratan untuk high pressure casing. Velocity compound stage ini sering juga diistilahkan sebagai Curtis Stage.

Pressure compound stage melibatkan beberapa set *nozzle* yang memiliki *pressure drop* yang kecil pada setiap set dan memiliki disipasi kecepatan yang lengkap disetiap baris *rotating buckets*. Sama seperti *velocity compound stage*, *pressure compound stage* juga sering diistilahkan sebagai *Rateau impulse stages*. *Pressure compound stage* ini biasa digunakan pada semua *stages* yang ada didalam turbin kecuali untuk turbin yang memiliki desain lama pada *reheat* dan *non-reheat turbines*. Pada desain yang lama, digunakan *Curtis stage* sebagai *stage* awal. Gambar 2.2 menunjukkan struktur turbin bertipe *velocity compounded* dan *pressure compounded*.



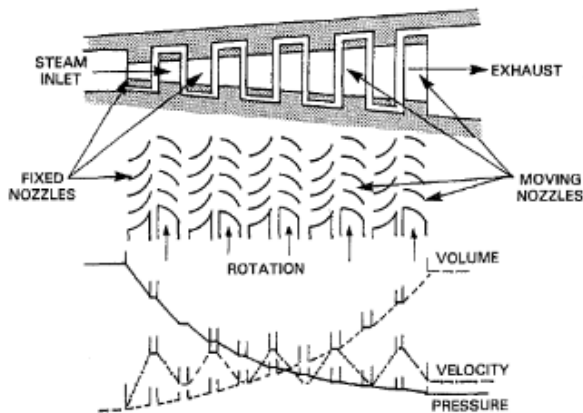
Gambar 2. 2 Struktur *velocity compounded* dan *pressure compounded turbine*[5]

o Reaction Stages

Reaction stages yang ideal terdiri dari rotating nozzles dengan stationary buckets. Akan tetapi, kurang praktis apabila uap yang ada langsung dialirkan untuk memutar nozzles. Pada praktiknya, reaction stage terdiri dari rotating dan stationary nozzles. Uap yang masuk menuju turbin akan diekspansi dan dialirkan melalui stationary nozzles dan mengalami peningkatan kecepatan aliran. Selanjutnya, uap akan memasuki rotating nozzles dan akan mengalami ekspansi lebih lanjut. Gaya dari kecepatan yang didapatkan dari ekspansi mula-mula dan ekspansi lanjutan pada rotating blades akan diberikan ke rotating nozzles.

Ekspansi uap yang terjadi didalam stationary nozzles pada reaction turbine merupakan aksi impuls. Sehingga, reaction stage pada pengaplikasian turbin secara aktual merupakan kombinasi dari prinsip kerja impulse dan reaction. Reaction turbine diklasifikasikan sebagai percent reaction dengan mengacu pada jumlah energi yang dikonversikan pada rotating nozzles. Umumnya, didalam reaction stage terjadi penurunan tekanan sebesar 50% pada rotating blades dan 50% sisanya terjadi pada stationary nozzles.

Terdapat banyak perbedaan yang terdapat pada *impulse turbine* dan *reaction turbine*. Diantara perbedaan-perbedaan tersebut, terdapat tiga perbedaan utama yang berhubungan dengan perilaku alami pada proses ekspansi uap. Perbedaan tersebut adalah jumlah *stages* pada turbin, desain dari *buckets* pada turbin, dan persyaratan untuk melakukan penyegelan (*sealing*) pada tiap-tiap *stage*. Untuk lebih jelasnya, dapat dilihat penampang turbin bertipe *reaction stages* pada Gambar 2.3.



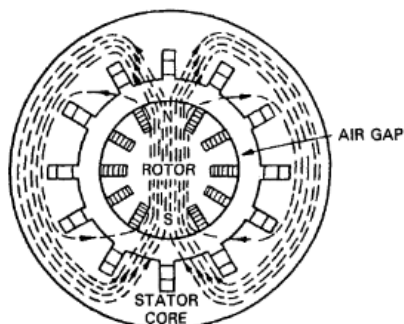
Gambar 2. 3 Struktur sederhana *reaction turbine stages*[5]

2.1.2.2 Proses produksi energi listrik

Energi putar mekanis dikonversi menjadi energi listrik didalam generator menggunakan putaran dari medan magnet yang terdapat pada kumparan rotor.

Putaran yang terjadi didalam turbin akan turut memutar rotor yang terdapat didalam generator sehingga akan menimbulkan energi listrik didalam kumparan stator pada generator. Rotor pada generator mengandung beberapa slot yang digunakan sebagai tempat untuk membentuk kumparan rotor yang biasa juga disebut field windings. Ketika arus DC dialirkan melewati lilitan tersebut, maka medan magnet akan timbul pada rotor. Medan magnet ini kemudian akan diputar oleh poros turbin. Rotor didalam generator dikelilingi oleh kumparan stator. Medan magnet rotor yang melewati kumparan stator ini akan menyebabkan elektron pada kumparan stator bergerak sehingga

menimbulkan arus listrik. Ketika kutub utara dari medan magnet rotor melewati kumparan stator, maka arus akan mengalir menuju arah tertentu, apabila kutub selatan medan magnet rotor juga melewati kumparan stator pada sisi yang sama, maka arus listrik akan mengalir menuju arah yang berlawanan dengan arah arus yang ditimbulkan oleh kutub utara medan magnet rotor. Arus listrik ini disebut sebagai arus AC (Alternating Current) dan merupakan jenis arus yang paling banyak dihasilkan oleh kebanyakan pembangkit listrik.

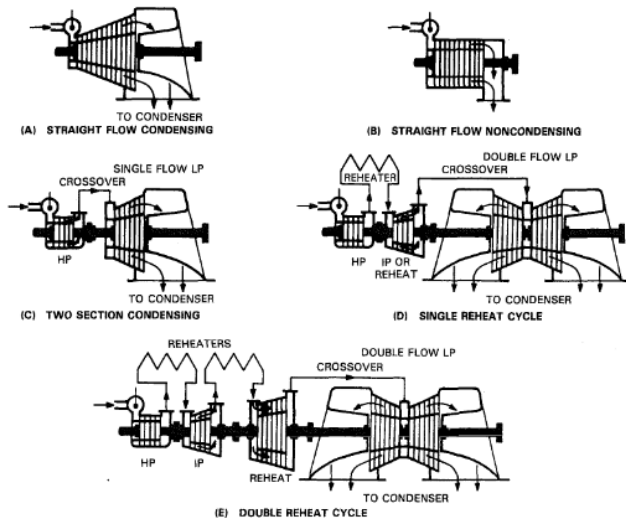


Gambar 2. 4 Penampang *magnetic circuit* pada *generator*[5]

Pada stator terdiri dari tiga grup kumparan yang terpisah sebesar 120 derajat untuk setiap kumparannya. Kumparan-kumparan ini biasa dikenal sebagai fasa (*phase*). Energi listrik akan dihasilkan pada setiap fasa di *stator*. Dikarenakan setiap fasa pada *stator* terpisah sejauh 120 derajat, maka tegangan dan arus yang dibangkitkan pada tiap-tiap fasa juga akan memiliki perbedaan fasa sebesar 120 derajat. Jenis seperti ini biasa disebut dengan arus atau tegangan tiga fasa (R-S-T).

2.1.3 Tipe *Steam Turbine*

Steam turbine dapat dibagi menjadi banyak jenis bergantung pada variasi desain yang digunakan. Berikut merupakan beberapa desain *steam turbine* yang biasa digunakan :



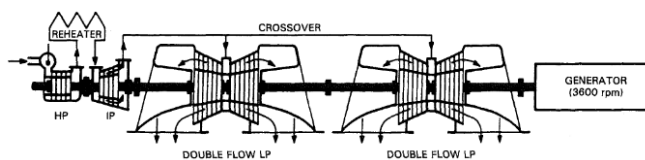
Gambar 2. 5 Tipe-tipe *steam turbine*[5]

Untuk unit yang kecil tanpa menggunakan *reheat*, *steam turbine* biasa terdiri dari *single turbine* dengan aliran uap masuk menuju turbin, mengalami ekspansi didalamnya, dan dikeluarkan menuju kondenser atau *process line*. Turbin seperti ini berjenis *straight-flow turbine* yang ditunjukkan pada Gambar 2.5 (A) dan (B). Kemudian untuk unit besar tanpa *reheat*, uap akan terekspansi pada *section* awal kemudian akan dikeluarkan menuju turbin yang lain. Kemudian uap yang keluar dari turbin terakhir ini akan diteruskan menuju kondenser atau menuju *process line*. Pada konfigurasi seperti ini, turbin yang pertama didesain sebagai *high pressure turbine* dan yang kedua sebagai *low pressure turbine* seperti yang ditunjukkan oleh Gambar 2.5 (C).

Pada *single reheat cycle*, uap dari *boiler* akan mengalir menuju *high pressure turbine* yang kemudian akan diekspansi dan dikembalikan menuju *boiler* untuk dipanaskan kembali pada *reheater*. Uap yang sudah dipanaskan kembali dari *boiler* akan mengalir menuju *intermediate pressure turbine* yang kemudian uap akan diekspansi dan dikeluarkan menuju *low pressure turbine* melalui *crossover line*. Uap yang dikeluarkan dari *LP turbine* akan dialirkan menuju kondenser. Sehingga, pada *single reheat cycle* memiliki *high, intermediate, low pressure turbine sections* seperti yang ditunjukkan oleh Gambar 2.5 (D).

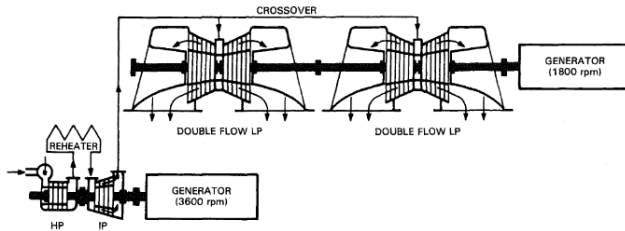
Pada *double reheat cycle*, uap dari *boiler* akan dialirkan melalui *high pressure turbine*, setelah itu uap akan diekspansi dan dikembalikan menuju *boiler* untuk dipanaskan kembali. Uap yang sudah dipanaskan kembali akan dialirkan menuju sisi *intermediate pressure turbine*, kemudian uap tersebut kembali diekspansi dan dialirkan menuju boiler untuk pemanasan ulang. Selanjutnya uap yang sudah dipanaskan kembali oleh *reheater* akan dialirkan menuju *reheater turbine* kemudian uap akan diekspansi dan dikeluarkan menuju *crossover line*. *Crossover line* ini akan mengalirkan uap menuju *low pressure turbine*, kemudian uap yang sudah terekspansi semua akan dikeluarkan menuju kondenser. Konfigurasi ini hampir sama dengan *single reheat cycle* dimana gambar dari konfigurasi *double reheat cycle* dapat dilihat oada Gambar 2.5 (E).

Secara keseluruhan susunan dari *steam turbine generator* pada *steam power plant* terbagi menjadi *tandem compound* atau *cross compound*. Pada konfigurasi secara *tandem compound*, seluruh poros turbin dan generator terkopel menjadi satu. Seluruh bagian turbin menggerakkan generator yang sama dan berjalan dikecepatan yang sama pula.



Gambar 2. 6 Struktur *tandem compound steam turbine*[5]

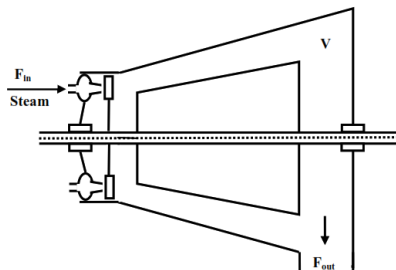
Pada konfigurasi *cross compound*, terdapat dua buah *generator*. Susunan seperti ini digunakan untuk meningkatkan efisiensi dari turbin. Pada *cross compound turbine* biasanya terdiri dari *high pressure* dan *intermediate pressure turbine* yang menjalankan *generator* berkecepatan 3600 rpm. Kemudian uap yang dikeluarkan oleh *intermediate turbine* akan diteruskan menuju *low pressure turbine* yang terhubung dengan *generator* kedua yang beroperasi pada kecepatan 1800 rpm. *Low pressure turbine* beroperasi pada kecepatan yang lebih rendah memungkinkan penggunaan bilah pada *last-stage turbine* yang lebih panjang sehingga menyebabkan *losses* pada proses pembuangan berkurang.



Gambar 2. 7 Struktur *cross compound steam turbine*[5]

2.1.4 Model Matematis *Steam Turbine Tandem Compound*[10]

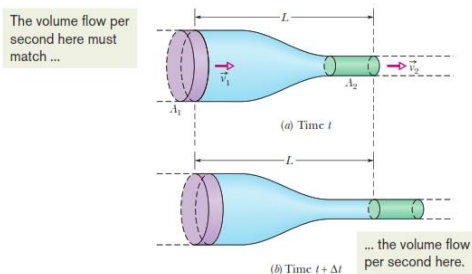
Steam turbine bertipe *tandem compound* memiliki 3 bagian utama turbin yang tersusun didalam satu poros dan tersambung menuju poros generator. Bagian-bagian ini dibagi menjadi tiga, yaitu *high pressure turbine*, *medium pressure turbine*, dan *low pressure turbine*. Setiap bagian ini tersusun dari satu set bilah *rotor* dan bilah *stator* yang memiliki susunan berbeda-beda. Bilah-bilah ini disusun sedemikian rupa agar uap yang masuk kedalam turbin dapat dimanfaatkan seefisien mungkin. Model *free body diagram* yang terdapat pada salah satu bagian dari turbin adalah sebagai berikut :



Gambar 2. 8 *Free body diagram* pada unit *steam turbine*[10]

Permodelan matematis dari *steam turbine* secara teoritis sebenarnya sangat kompleks, sehingga perlu adanya asumsi dan pengabaian beberapa variabel yang ada. Untuk mendapatkan persamaan *input-output* dari gambar 1 diatas, dapat digunakan persamaan kontinuitas (*continuity equation*) pada fluida dimana pada persamaan ini

menyatakan bahwa volume fluida yang masuk kedalam suatu permukaan akan sama dengan yang keluar disisi yang lain[6].



Gambar 2. 9 Ilustrasi *continuity equation* pada fluida[6]

Jika L merupakan jarak yang harus ditempuh fluida untuk mencapai bagian akhir dengan kecepatan v , maka volume fluida (ΔV) yang melewati saluran dapat dituliskan dalam persamaan (2.1).

$$\Delta V = \Delta x = Av\Delta t \quad (2.1)$$

Apabila persamaan diatas diaplikasikan pada sisi kiri dan kanan saluran, maka didapatkan persamaan (2.2).

$$\Delta V = A_1v_1\Delta t = A_2v_2\Delta t \quad (2.2)$$

Dan dikarenakan waktu yang ditempuh fluida didalam saluran adalah sama, maka persamaan (2.2) dapat disederhanakan menjadi

$$A_1v_1 = A_2v_2 \quad (2.3)$$

Hubungan antara kecepatan fluida dan luas penampang yang dilewati fluida pada saluran disebut juga sebagai *continuity equation*.

Didalam turbin, terdapat perbedaan kecepatan aliran yang terdapat pada saluran *input* dan *output*. Selisih antara perbedaan *flow input* dan *flow output* ini massa dari uap yang terdapat didalam turbin pada setiap waktu. Sesuai dengan *continuity equation*, idealnya debit aliran masuk sama dengan debit aliran keluar, sehingga

$$F_{in}(t) = F_{out}(t)$$

Karena terdapat perbedaan aliran, maka terdapat drop aliran didalam turbin, sehingga

$$F_{in}(t) - F_{out}(t) = F_{drop}(t)$$

$$F_{drop}(t) = \frac{dW}{dt} = V \frac{d\rho}{dt} \quad (2.4)$$

Dimana : $F = \text{steam mass flow rate (kg/s)}$
 $W = \text{weight of steam in turbine (kg)}$
 $V = \text{volume of turbine (m}^3\text{)}$
 $\rho = \text{density of steam (kg/m}^3\text{)}$

Kemudian dengan mengacu pada Hukum Pascal, terlihat bahwa

$$P = \frac{F}{A} \text{ dan } F \sim \text{massa} \quad (2.5)$$

Dimana : $P = \text{pressure (Pa)}$
 $F = \text{force (N)}$
 $A = \text{surface area (m}^2\text{)}$

Dan $\text{Flowrate}(F) \sim \text{massa}$

Maka, $P \sim \text{Flowrate}(F)$

$$\frac{P}{P_0} = \frac{F_{out}}{F_0}$$

$$F_{out} = P \cdot \frac{F_0}{P_0} \quad (2.6)$$

Dimana : $P = \text{pressure of steam in the turbine (kPa)}$
 $F_{out} = \text{flow steam out (kg/s)}$
 $F_0 = \text{rated flow out of turbine (kg/s)}$
 $P_0 = \text{rated pressure (kPa)}$

Dengan asumsi bahwa temperature didalam turbin konstan, maka dengan menggunakan *differential chain rule* didapatkan persamaan (2.7)

$$\frac{d\rho}{dt} = \frac{d\rho}{dP} \frac{dP}{dt} \quad (2.7)$$

Dengan melakukan substitusi pada persamaan (2.4) dan (2.7), didapatkan model matematis dari salah satu bagian turbine sebagai berikut :

$$F_{in}(t) - F_{out}(t) = V \frac{d\rho}{dt} = V \frac{d\rho}{dP} \frac{dP}{dt} = V \frac{d\rho}{dP} \frac{d \left(F_{out} \frac{P_0}{F_0} \right)}{dt} = V \frac{P_0}{F_0} \frac{d\rho}{dP} \frac{dF_{out}}{dt} = T_T \frac{dF_{out}}{dt}$$

$$T_T \frac{dF_{out}}{dt} + F_{out}(t) = F_{in}(t) \quad (2.8)$$

Dengan menggunakan transformasi Laplace, ditemukan hubungan *input-output* pada turbin sesuai dengan persamaan (2.9)

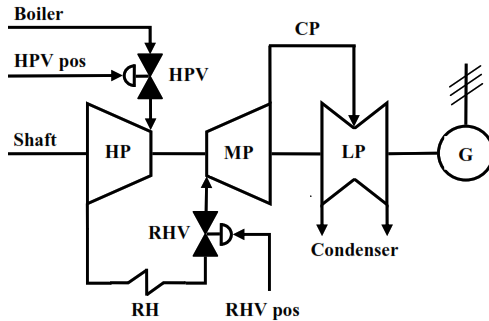
$$H_T(s) = \frac{F_{out}(s)}{F_{in}(s)} = \frac{1}{T_T s + 1} \quad (2.9)$$

Dengan, $T_T = V \frac{P_0}{F_0} \frac{d\rho}{dP}$ merupakan *time constant* turbin tiap *section*

Kemudian, setelah mendapatkan fungsi transfer F_{in} dan F_{out} pada turbin, selanjutnya akan dibentuk persamaan yang menyatakan hubungan antara torsi (T_m) yang dihasilkan dengan input berupa *flow* yang keluar dari turbin (F_{out}). Hubungan antara torsi yang dihasilkan oleh turbine dengan *flow* yang masuk adalah proporsional sehingga dapat dituliskan persamaan torsinya sebagai berikut :

$$T_m(t) = k \cdot F_{out}(t) \quad (2.10)$$

Agar lebih jelas dalam mengamati proses yang terjadi didalam turbin, berikut akan saya sertakan gambar dari struktur *steam turbine tandem compound* secara keseluruhan



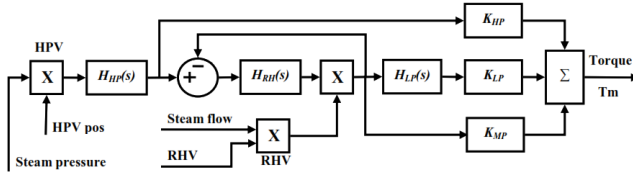
Gambar 2. 10 Kofigurasi *steam turbine* tipe *tandem compound*[10]

Dari gambar tersebut, terlihat bahwa turbin tersusun dari tiga bagian yang dikofigurasi seri, sehingga aliran uap yang terjadi adalah sebagai berikut :

Boiler→HP Valve→HP Section→Re-Heater(Boiler)→RH Valve→MP Section→Cross Over Pipe→LP Section→Condenser

Dari aliran tersebut, terlihat bahwa *flow* utama dari boiler akan pertama kali masuk ke turbin menuju *HP section* dan menghasilkan F_{out1} , kemudian F_{out1} ini akan dialirkan menuju *re-heater* untuk dipanaskan kembali dan dikirim menuju *MP section* dan menghasilkan F_{out2} . Dan uap yang keluar dari sisi MP akan langsung dimasukkan kedalam *LP section* dan menghasilkan F_{out3} . Aliran $F_{out} - F_{out}$ ini nantinya akan proporsional terhadap torsi yang dihasilkan oleh turbin yang digunakan untuk memutar *generator*.

Kemudian, dilakukan penyusunan *block diagram steam turbine* yang mengandung tiga bagian turbin agar didapatkan fungsi transfer dari *turbine tandem compound* secara keseluruhan. *Block diagram* ini mengandung 3 bagian dari *transfer function* yang mewakili tiap-tiap *section* yang ada pada turbin. Kemudian output dari tiap-tiap *section* ini akan dikalikan dengan suatu konstanta dan dijumlahkan masing-masing hasil kalinya sehingga menjadi suatu besaran yang menyatakan torsi turbin (T_m).



Gambar 2. 11 Diagram blok *steam turbine tandem compound*[10]

Terlihat bahwa turbin memiliki 3 *transfer function* utama dimana ketiga *transfer function* tersebut memiliki bentuk persamaan yang sesuai dengan *transfer function* H_T . Input yang masuk menuju turbin adalah *steam flow* yang diatur oleh HPV. Kemudian *flow* dari *steam* ini akan lanjut mengalir sampai *LP section*. Dengan anggapan bahwa bukaan *valve* pada RHV adalah 100% (*full open*), maka akan didapatkan *transfer function* dari tiap-tiap *section* pada turbin sebagai berikut :

$$H_{HP} = \frac{1}{T_{HP}S + 1} \quad (2.11)$$

$$H_{RH} = \frac{1}{T_{RH}S}$$

$$H_{MP} = \frac{H_{RH}}{1 + H_{RH}} = \frac{1}{T_{RH}S + 1} = \frac{1}{T_{MP}S + 1} \quad (2.12)$$

$$H_{LP} = \frac{1}{T_{LP}S + 1} \quad (2.13)$$

K_{HP} , K_{MP} , dan K_{LP} merupakan *power fraction* pada turbin dimana besar dari jumlahan nilai ketiga konstantan tersebut sama dengan 1. Respon pada perubahan *steam flow* pada *input HP turbine* bergantung pada *time constant HP turbine* dimana besarnya memiliki *range* 0.2 – 0.3 sekon. Sedangkan untuk respon pada sisi *MP turbine* bergantung pada *time constant* pada RH yang sudah tergabung dengan *time constant* pada MP dan untuk selanjutnya akan dinotasikan sebagai T_{MP} . Dikarenakan proses yang terjadi pada re-heater cukup Panjang jaraknya, maka *time constant* yang ada pada *MP section* juga cukup besar rangenya yakni antara 5 – 10 sekon. Kemudian untuk respon pada sisi

LP, *time constant* yang dimiliki cukup besar jika dibandingkan dengan sisi HP dikarenakan pada sisi LP ini juga terdapat saluran pipa *crossover* sehingga *time constant* pada LP memiliki nilai sekitar 0.5 sekon.

Dengan menggunakan nilai nominal dari $T_{MP} = 7.5$ sekon dan $T_{HP} = 0.25$ sekon. Maka didapatkan bentuk *transfer function* dari masing-masing section adalah

$$H_{HP} = \frac{1}{0.25s + 1} \quad (2.14)$$

$$H_{MP} = \frac{1}{7.5s + 1} \quad (2.15)$$

$$H_{LP} = \frac{1}{0.5s + 1} \quad (2.16)$$

Sehingga bentuk *transfer function* dari turbin *tandem compound* dengan 3 section adalah sebagai berikut :

$$H_{Turbin}(s) = \frac{K_{HP}}{0.25s + 1} + \frac{K_{MP}}{(0.25s + 1)(7.5s + 1)} + \frac{K_{LP}}{(0.25s + 1)(7.5s + 1)(0.5s + 1)}$$

Dengan memisalkan nilai *power fraction* $K_{HP} = 0.3$, $K_{MP} = 0.3$, dan $K_{LP} = 0.4$, maka didapatkan hasil *transfer function turbine* secara keseluruhan adalah

$$H_{Turbin}(s) = \frac{0.3}{0.25s + 1} + \frac{0.3}{(0.25s + 1)(7.5s + 1)} + \frac{0.4}{(0.25s + 1)(7.5s + 1)(0.5s + 1)}$$

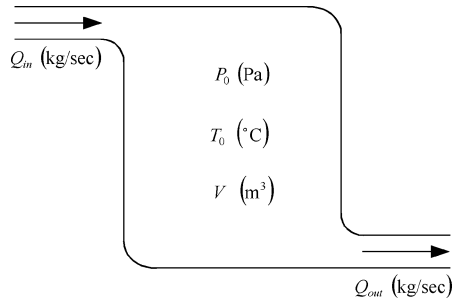
$$H_{Turbin}(s) = \frac{Tm(s)}{Fin(s)} = \frac{1.125s^2 + 2.55s + 1}{0.9375s^3 + 5.75s^2 + 8.25s + 1} \quad (2.17)$$

2.1.5 Parameter Sistem Menggunakan Data Heat Balance Steam Turbine Tandem Compound 200-MW[12]

Steam chest dan *inlet piping* pada *turbine cylinder, reheater*, dan *crossover piping* terdapat *delay* yang ditimbulkan antara perpindahan *valve* dan perubahan *steam flow*. Tujuan utama dari permodelan *steam system* untuk pembelajaran proses dinamis adalah dengan menemukan *delay-delay* yang ada. Selain itu, hal yang juga perlu diperhatikan dalam memodelkan turbin adalah konfigurasi yang ada didalam turbin tersebut,

dikarenakan konfigurasi tersebut akan menentukan persamaan permodelan yang akan terbentuk. Disini konfigurasi turbin yang digunakan adalah *tandem compound* dengan *single reheater* dimana tiap-tiap *section High Pressure, Medium Pressure, dan Low Pressure* akan dikopel kedalam satu poros turbin secara seri.

Untuk memodelkan turbin menggunakan data *heat balance*, pada awalnya akan diasumsikan bahwa *steam vessel* yang ada pada tiap-tiap *section* memiliki bentuk seperti gambar berikut



Gambar 2. 12 Diagram *steam vessel*[12]

Dengan menggunakan *continuity equation*, akan didapatkan persamaan untuk menentukan *time constant* dari *steam* pada saat melalui *vessel* tiap-tiap *section*. Persamaan tersebut dapat dituliskan sebagai persamaan (2.18) dan (2.19).

$$T_v = V_{vessel} \frac{P_0}{Q_0} \times K_{vessel} \quad (2.18)$$

$$K_{vessel} = \frac{d\rho}{dP} = \frac{\partial}{\partial P} \left(\frac{1}{v_{steam}} \right) \Big|_{T_0} \quad (2.19)$$

Dimana T_v merupakan *time constant vessel* pada tiap-tiap *section* dalam detik, P_0 merupakan *steady state pressure* atau *rated pressure* pada *vessel* dengan satuan Pa, Q_0 merupakan *rated flow out* pada *vessel* dengan satuan kg/s, V_{vessel} merupakan volume dari tiap-tiap *vessel* dengan satuan m^3 , K_{vessel} (s^2/m^2) merupakan *density change* terhadap perubahan tekanan pada kondisi *temperature* yang konstan, dan v_{steam} merupakan *specific volume* dari *steam* (m^3/kg) dengan asumsi *temperature* pada *vessel* dalam kondisi yang konstan.

Untuk melakukan perhitungan K_{vessel} , perlu menggunakan tabel *thermodynamic data of steam*. Dengan mengasumsikan P_2 dan P_1 terikat pada interval *pressure* yang paling kecil dan dapat dicari didalam tabel *thermodynamic data* dengan *steam volume* v_2 dan v_1 yang juga dapat ditemukan didalam tabel tersebut yang nilainya berkorespondensi dengan data *pressure* P_2 dan P_1 . Kemudian, pada kondisi perubahan yang dibatasi sangat singkat, maka persamaan K_{vessel} dapat dituliskan kembali menjadi persamaan (2.20).

$$K_{vessel} = \frac{d\rho}{dP} = \frac{\partial}{\partial P} \left(\frac{1}{v_{steam}} \right) \Bigg|_{T_0} = \frac{\frac{1}{v_2} - \frac{1}{v_1}}{P_2 - P_1} \Bigg|_{T_{vessel}} \quad (2.20)$$

Perlu diingat juga bahwa tabel *thermodynamic data*, pada volume yang spesifik dapat ditemukan nilai *pressure* yang berbeda-beda ditiap kondisi temperature yang berbeda. Apabila nilai temperature pada *vessel* yang dicari tidak ada, maka dapat dilakukan interpolasi secara linier untuk menemukan nilai-nilai tersebut. Setelah K_{vessel} dari tiap-tiap *section* diketahui, maka kita dapat mencari parameter *time constant* T_v pada *steam turbine*.

Kemudian, untuk mencari parameter *power fraction* pada tiap-tiap *section*, diasumsikan bahwa *rated output power* pada tiap *section* memiliki nilai sama dengan satu jika dijumlahkan semua karena konfigurasi turbin yang berupa *tandem compound* sehingga *power fraction* akan merepresentasikan kontribusi dari tiap-tiap *section* yang ada.

$$F_{HP} + F_{MP} + F_{LP} = 1 \quad (2.21)$$

Dan perlu diketahui bahwa *power fraction* pada tiap-tiap *section* turbin sebanding dengan total *power* pada *section* turbin tersebut ($F_{HP} \sim P_{HP}$) sehingga dapat dibentuk persamaan baru yang akan membantu dalam proses menemukan nilai parameter *power fraction*. Persamaan tersebut dapat dituliskan sebagai persamaan (2.22) dan (2.23).

$$\frac{F_{LP}}{F_{HP}} = \frac{P_{LP}}{P_{HP}} = a_1 \quad (2.22)$$

$$\frac{F_{MP}}{F_{HP}} = \frac{P_{MP}}{P_{HP}} = a_2 \quad (2.23)$$

Dengan P_{HP} , P_{MP} , dan P_{LP} merupakan *HP turbine power*, *MP turbine power*, dan *LP turbine power*. Untuk menghitung *power* tiap-tiap *section* turbin tersebut, perlu diketahui *steam enthalpy* dan juga *mass flow* pada tiap *section* yang mana data tersebut sudah tersedia didalam *heat balance map*. *Power* dalam tiap turbin dapat disederhanakan menjadi persamaan (2.24).

$$P_X = \sum_k Q_k (h_{ink} - h_{outk}) \quad X = HP, MP, LP \quad (2.24)$$

Dimana P_X merupakan *power* dari turbin X (kJ/s), k merupakan *turbine stages* antar *draining points*, Q_k merupakan *mass flow* pada *stage k* (kg/s), h_{ink} merupakan *inlet steam enthalpy* pada *stage k* (kJ/kg), dan h_{outk} merupakan *outlet steam enthalpy* pada *stage k* (kJ/kg). Dengan menyederhanakan persamaan (2.21), (2.22), dan (2.23) akan diperoleh persamaan baru untuk mencari nilai *power fraction* pada tiap *section*.

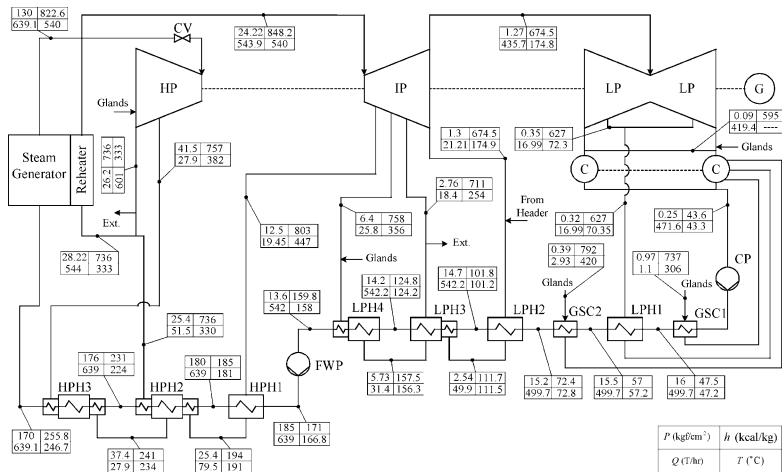
$$F_{HP} = \frac{1}{1 + a_1 + a_2} \quad (2.25)$$

$$F_{MP} = \frac{a_2}{1 + a_1 + a_2} \quad (2.26)$$

$$F_{LP} = \frac{a_1}{1 + a_1 + a_2} \quad (2.27)$$

Dari persamaan-persamaan tersebut dapat ditentukan parameter *power fraction* untuk memodelkan *plant steam turbine*.

Berikut merupakan data *heat balance* yang akan dipakai untuk keperluan mendapatkan model *steam turbine tandem compound* dengan *single reheater*



Gambar 2. 13 Heat balance map pada steam turbine tandem compound 200-MW[12]

Data tersebut merupakan data *heat balance map* dari *steam turbine 200-MW tandem compound* dengan *single reheater*. Didalamnya merepresentasikan data *heat balance* dari hampir semua bagian didalam unit *steam turbine*. Terdapat 4 data *thermodynamic* yang disajikan didalamnya mulai dari *pressure* (kgf/cm²), *enthalpy* (kcal/kg), *mass flow* (T/h), dan *temperature* (°C). dikarenakan memiliki beberapa perbedaan satuan dengan persamaan yang dipakai, maka perlu adanya konversi satuan guna mendapatkan hasil perhitungan yang tepat. Konversi-konversi tersebut dapat ditulis sebagai berikut $1 \text{ kgf/cm}^2 = 98.07 \text{ kPa}$, $1 \text{ T/h} = 0.278 \text{ kg/s}$, dan $1 \text{ kcal/kg} = 4.186 \text{ kJ/kg}$.

Untuk yang pertama, akan dicari parameter *time constant* pada sisi *High Pressure*. Data *heat balance* yang dipakai merupakan data pada *steam chest* antara *control valve* dan sisi *HP turbine*. Dari gambar, dapat kita peroleh *steam chest pressure* $P_{SC} = 130 \frac{\text{kgf}}{\text{cm}^2} = 130 \times 98.07 = 12749.1 \text{ kPa}$ dan juga *mass flow* $Q_{SC} = 639.1 \frac{\text{T}}{\text{h}} = 639.1 \times 0.278 = 177.67 \frac{\text{kg}}{\text{s}}$. Dari *heat balance map* juga dapat diketahui bahwa *temperature* dari *steam* didalam *steam chest* adalah $540 \text{ }^\circ\text{C}$, sehingga dari data tersebut dapat diketahui nilai volume dan *pressure* pada suhu

tertentu melalui tabel *thermodynamic*. Hasil yang dapat diambil adalah sebagai berikut

$$P_2 = 15000 \text{ kPa}, v_2 = 0.0162 \frac{\text{m}^3}{\text{kg}}$$

$$P_1 = 10000 \text{ kPa}, v_1 = 0.03507 \frac{\text{m}^3}{\text{kg}}$$

Kemudian dari data-data tersebut, bisa dicari nilai dari *density change* (K_{vessel}) pada *section steam chest* menggunakan persamaan (2.20)

$$K_{SC} = \frac{1}{15000} - \frac{1}{10000} \Bigg|_{540^{\circ}\text{C}} = 6.64 \times 10^{-6} \frac{\text{s}^2}{\text{m}^2}$$

untuk volume dari *steam chest* antara *control valve* dan sisi *HP turbine* dapat dimisalkan sebagai tabung dengan jari-jari $r_{sc} = 42 \text{ cm}$ dan tinggi tabung $h_{sc} = 150 \text{ cm}$. Sehingga dapat dihitung volume dari tabung *steam chest* adalah

$$V_{SC} = \pi \times r_{sc}^2 \times h_{sc} = \pi \times (42 \times 10^{-2})^2 \times 1.5 = 0.83 \text{ m}^3$$

Sehingga dapat diketahui nilai dari *time constant* pada *steam chest* (T_{SC}) yang kemudian akan disebut sebagai *time constant HP turbine* (T_{HP}) adalah

$$T_{HP} = T_{SC} = V_{SC} \frac{P_{SC}}{Q_{SC}} \times K_{SC} = 0.83 \times \frac{12749.1}{177.67} \times 6.64 \times 10^{-6} = 0.396 \text{ second}$$

Selanjutnya untuk yang kedua adalah mencari *time constant* pada sisi *reheater* yang berada antara sisi *HP turbine* dan *MP turbine*. Dari data *heat balance map*, dapat dilihat bahwa nilai dari *steam pressure* didalam saluran *reheater* menuju *MP turbine* adalah $P_{RH} = 24.22 \frac{\text{kgf}}{\text{cm}^2} = 24.22 \times 98.07 = 2375.25 \text{ kPa}$ dan *mass flow* didalam saluran tersebut $Q_{RH} = 543.9 \frac{\text{T}}{\text{h}} = 543.9 \times 0.278 = 151.2 \frac{\text{kg}}{\text{s}}$. Kemudian, dapat dilihat juga didalam *heat balance map* bahwa steam

temperature didalam *reheater* memiliki nilai diantara 333 °C hingga 540 °C sehingga dapat diambil pendekatan bahwa nilai temperature rata-rata didalam *reheater* adalah

$$T_{av} = \frac{333 + 540}{2} = 436.5 \text{ } ^\circ\text{C}$$

Dengan menggunakan temperature tersebut, dapat diambil nilai *pressure* dan volume dari *steam* pada suhu tertentu menggunakan tabel *thermodynamic*.

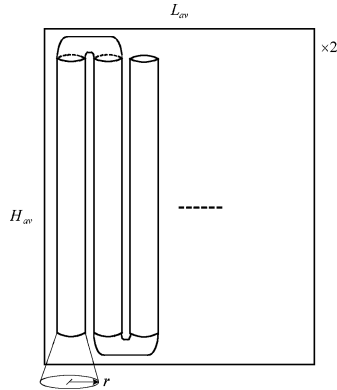
$$P_2 = 2500 \text{ kPa}, v_2 = 0.12743 \frac{\text{m}^3}{\text{g}}$$

$$P_1 = 2000 \text{ kPa}, v_1 = 0.16020 \frac{\text{m}^3}{\text{kg}}$$

Kemudian dengan menggunakan persamaan (2.20) tentang *density change* (K_{vessel}) diperoleh bahwa

$$K_{RH} = \frac{\frac{1}{0.12743} - \frac{1}{0.16020}}{2500 - 2000} \Bigg|_{436.5^\circ\text{C}} = 3.21 \times 10^{-6} \frac{\text{s}^2}{\text{m}^2}$$

Sehingga untuk mencari nilai *time constant* disisi *reheater*, kita perlu mengetahui volume dari saluran pipa yang terdapat didalam *reheater*. Saluran ini terdiri dari 2 set pipa yang di-*mounting* pada *heating wall* didalam *furnace*. Secara sederhana, pipa-pipa dari *reheater* data digambarkan seperti berikut



Gambar 2. 14 Konfigurasi saluran pipa *reheater*[12]

Dari data tersebut diketahui bahwa pipa-pipa disusun sedemikian rupa dengan lebar total untuk satu set pipa mencapai 8 meter ($L_{av} = 8 \text{ m}$) sedangkan untuk tinggi rata-rata masing-masing pipa adalah 12 meter ($H_{av} = 12 \text{ m}$) dan jari-jari dari masing-masing pipa sekitar 45 sentimeter ($r = 45 \text{ cm}$). Untuk satu set pipa tersebut, volume dari saluran pipa uap dapat dihitung sebagai berikut

$$V_{RH \ 1 \ sisi} = \pi \times r^2 \times H_{av} \times \text{jumlah saluran pipa dalam 1 sisi}$$

$$V_{RH \ 1 \ sisi} = \pi \times r^2 \times H_{av} \times \frac{L_{av}}{2r} = \pi \times (45 \times 10^{-2})^2 \times 12 \times \frac{8}{2 \times 45 \times 10^{-2}} = 67.86 \text{ m}^3$$

Sehingga apabila didalam *furnace* terdapat 2 set pipa yang menyusun *reheater*, maka volume total yang terbentuk adalah

$$V_{RH \ 2 \ sisi} = 2 \times V_{RH \ 1 \ sisi} = 2 \times 67.86 = 135.72 \text{ m}^3$$

Dari data-data tersebut, dapat diperoleh nilai dari *time constant reheater* (T_{RH}) yang nantinya akan disebut sebagai *time constant MP turbine* (T_{MP}).

$$T_{MP} = T_{RH} = V_{RH \ 2 \ sisi} \frac{P_{RH}}{Q_{RH}} \times K_{RH} = 135.72 \times \frac{2375.25}{151.2} \times 3.21 \times 10^{-6} = 6.84 \text{ second}$$

Kemudian untuk *section* yang terakhir yaitu mencari nilai *time constant* pada *crossover pipe* yang menghubungkan *MP turbine* dan *HP turbine*. Dari data *heat balance map*, dapat diperoleh bahwa nilai *steam pressure* yang terdapat pada pipa *crossover* adalah $P_{CO} = 1.27 \frac{kgf}{cm^2} = 1.27 \times 98.07 = 124.55 \text{ kPa}$ dan untuk *mass flow* yang dimiliki oleh *steam* didalam pipa *crossover* adalah $Q_{CO} = 435.7 \frac{T}{h} = 435.7 \times 0.278 = 121.12 \frac{kg}{s}$. Dari *heat balance map*, juga dapat diperoleh nilai *temperature* dari *steam* yang terdapat didalam *pipa crossover* sebesar $T_{CO} = 174.8 \text{ }^\circ\text{C}$. Menggunakan nilai *temperature* tersebut, dapat diketahui nilai dari *pressure* dan *volume* dari *steam* pada suhu tertentu menggunakan tabel *thermodynamic*.

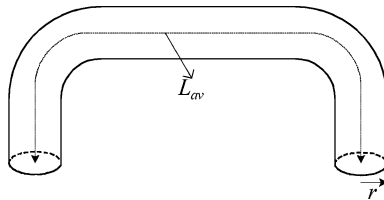
$$P_2 = 200 \text{ kPa}, v_2 = 1.01951 \frac{m^3}{kg}$$

$$P_1 = 100 \text{ kPa}, v_1 = 2.05337 \frac{m^3}{kg}$$

Dengan menggunakan persamaan (2.20) tentang *density change* (K_{vessel}) diperoleh bahwa

$$K_{CO} = \frac{\frac{1}{200} - \frac{1}{100}}{174.8^\circ\text{C}} = 4.94 \times 10^{-6} \frac{s^2}{m^2}$$

Pada sisi *crossover*, digunakan kontruksi pipa memanjang seperti pada gambar berikut



Gambar 2. 15 Konfigurasi saluran pipa *crossover*[12]

Pipa tersebut memiliki jari-jari sebesar 62 sentimeter ($r = 62 \text{ cm}$) dan panjang pipa sekitar 8.5 meter ($L_{av} = 8.5 \text{ m}$). Menggunakan data tersebut, dapat diperoleh besar volume dari saluran pipa *crossover* adalah

$$V_{CO} = \pi \times r_{CO}^2 \times L_{av} = \pi \times (62 \times 10^{-2})^2 \times 8.5 = 10.26 \text{ m}^3$$

Dengan menggunakan semua data yang sudah berhasil ditemukan, maka dapat diperoleh nilai *time constant* pada sisi *crossover* (T_{CO}) yang selanjutnya akan disebut sebagai *time constant LP turbine* (T_{LH}) adalah

$$T_{LP} = T_{CO} = V_{CO} \frac{P_{CO}}{Q_{CO}} \times K_{CO} = 10.26 \times \frac{124.55}{121.12} \times 4.94 \times 10^{-6} = 0.052 \text{ second}$$

Setelah mengetahui nilai dari *time constant* pada seluruh *section turbine*, kemudian dilanjutkan dengan menentukan *power fraction* (F) pada tiap-tiap *section* turbin. Untuk mencari *power fraction* tersebut, diperlukan perhitungan untuk mencari tiap-tiap *turbine's thermodynamic work*. Dengan menggunakan data *heat balance*, maka *thermodynamic work* pada turbin dapat diestimasi nilainya sebagai berikut

➔ P_{HP} : didalam sisi HP turbine, terdapat 2 stage saluran pipa keluar sehingga nilai dari $k = 2$

$$\begin{aligned} P_{HP} &= \sum_{k=1}^2 Q_k (h_{ink} - h_{outk}) \\ &= (639.1 \times (822.6 - 757) + (639.1 - 27.9) \times (757 - \\ &736.1)) \times 0.278 \frac{\text{kg}}{\text{s}} \times 4.186 \frac{\text{kJ}}{\text{kg}} \\ &= 63653 \frac{\text{kJ}}{\text{s}} \end{aligned}$$

➔ P_{MP} : didalam sisi MP turbine, terdapat 4 stage saluran pipa keluar sehingga nilai dari $k = 4$

$$\begin{aligned} P_{MP} &= \sum_{k=1}^4 Q_k (h_{ink} - h_{outk}) \\ &= (543.9 \times (848.2 - 803) + (543.9 - 19.45) \times (803 - \\ &758) \times (543.9 - 19.45 - 25.8) \times (758 - 711) \times (543.9 - \\ &19.45 - 25.8 - 18.4) \times (711 - 674.5)) \times 0.278 \frac{\text{kg}}{\text{s}} \times 4.186 \frac{\text{kJ}}{\text{kg}} \end{aligned}$$

$$= 103745 \frac{kJ}{s}$$

→ P_{LP} : didalam sisi LP turbine, terdapat 2 stage saluran pipa keluar sehingga nilai dari $k = 2$

$$\begin{aligned} P_{LP} &= \sum_{k=1}^2 Q_k (h_{ink} - h_{outk}) \\ &= (435.7 \times (674.5 - 627) + (435.7 - 16.99) \times (627 - \\ &595)) \times 0.278 \frac{kg}{s} \times 4.186 \frac{kJ}{kg} \\ &= 39676 \frac{kJ}{s} \end{aligned}$$

Sehingga, dari data-data tersebut dapat diperoleh nilai dari *power fraction* tiap-tiap *section* pada turbin adalah sebesar

$$a_1 = \frac{P_{LP}}{P_{HP}} = \frac{39676}{63653} = 0.62$$

$$a_2 = \frac{P_{MP}}{P_{HP}} = \frac{103745}{63653} = 1.63$$

$$F_{HP} = \frac{1}{1 + a_1 + a_2} = 0.31$$

$$F_{MP} = \frac{a_2}{1 + a_1 + a_2} = 0.5$$

$$F_{LP} = \frac{a_1}{1 + a_1 + a_2} = 0.19$$

Sehingga parameter-parameter dari sistem *steam turbine* yang diturunkan dengan menggunakan data *heat balance steam turbine* 200-MW dapat diringkas menjadi

→ *Power Fraction*

$$K_{HP} = F_{HP} = 0.31$$

$$K_{MP} = F_{MP} = 0.5$$

$$K_{LP} = F_{LP} = 0.19$$

→ *Time Constant*

$$T_{HP} = T_{SC} = 0.396 \text{ second}$$

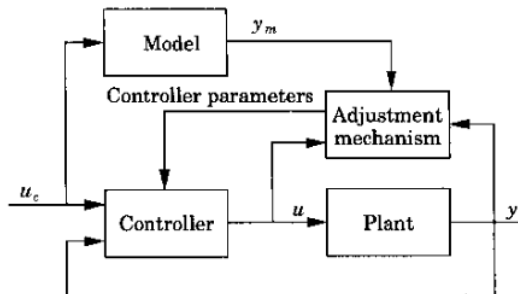
$$T_{MP} = T_{RH} = 6.84 \text{ second}$$

$$T_{LP} = T_{CO} = 0.052 \text{ second}$$

2.2 Model-Reference Adaptive Systems[2]

2.2.1 Pengenalan

Model-Reference Adaptive System (MRAS) merupakan salah satu jenis kontroler adaptif yang cukup penting. Metode ini sering juga dianggap sebagai *adaptive servo system* dimana performansi dan karakter sistem yang diinginkan dinyatakan sebagai *reference model*, yang memberikan respons yang diinginkan kepada sinyal kontrol. Ini merupakan cara yang sesuai untuk memberikan spesifikasi yang cocok pada permasalahan *servo control*. Diagram blok dari *Model-Reference Adaptive System* dapat ditunjukkan melalui gambar berikut



Gambar 2. 16 Diagram blok dari *Model-Reference Adaptive System (MRAS)*[2]

Pada gambar tersebut, terlihat bahwa sistem memiliki dua *loop*, yaitu *feedback loop* normal (*inner loop*) yang terdiri dari *plant* dan *controller* dan juga *feedback loop* lainnya (*outer loop*) yang berfungsi untuk mengubah nilai parameter pada kontroler. Perubahan nilai parameter ini disebabkan oleh *feedback* dari *error* antara *output* dari referensi model dengan *output* dari *plant*. Mekanisme untuk mengatur nilai parameter dari kontroler pada *Model-Reference Adaptive System*

dapat diperoleh menggunakan dua cara, yaitu dengan metode *gradient* (seperti *MIT Rule*) dan *stability theory*.

Pada *MRAS*, respon sistem yang diinginkan akan dinyatakan sebagai model, dan untuk parameter pada controller akan diatur berdasarkan *error* yang timbul dari *output* pada *closed-loop system* dan *model reference*. *Model-Reference Adaptive System* pada awalnya digunakan untuk sistem yang *deterministic continuous-time*.

2.2.2 MIT Rule[2]

MIT Rule merupakan salah satu jenis pendekatan yang digunakan pada *Model-Reference Adaptive System*. Nama tersebut diperoleh dari tempat terciptanya metode tersebut, yaitu *Massachusetts Institute of Technology (MIT)* tepatnya di *Instrumentation Laboratory* yang sekarang lebih dikenal sebagai *Draper Laboratory*.

Untuk menunjukkan lebih lanjut mengenai *MIT Rule*, kita akan meninjau suatu sistem *closed-loop* yang memiliki satu parameter yang dapat diatur (*adjustable parameter*) θ . Respon sistem yang diinginkan akan didefinisikan sebagai respon dari referensi model dimana *output*-nya adalah y_m . Dimisalkan juga e merupakan *error* yang terjadi antara *output* dari referensi model y_m dengan output dari sistem *closed-loop* y . Satu kemungkinan yang bisa dilakukan adalah mengatur parameter kontroler yang ada sedemikian rupa hingga *lost function* (J) yang ada dapat menjadi seminimal mungkin.

$$J(\theta) = \frac{1}{2} e^2 \quad (2.28)$$

Untuk membuat J seminimal mungkin, dapat dilakukan dengan mengubah parameter kontroler searah dengan *gradient* negatif dari J , sehingga

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2.29)$$

Hasil inilah yang melandasi terciptanya *MIT Rule*. Turunan parsial dari *error* yang ada terhadap parameter kontroler ($\partial e / \partial \theta$) biasa disebut dengan *sensitivity derivative* dari sistem, dimana turunan ini menggambarkan bagaimana *error* dipengaruhi oleh *adjustable parameter*. Jika diasumsikan bahwa perubahan parameter yang ada lebih

lambat dari pada *variable* lain yang ada didalam sistem, maka turunan $\partial e / \partial \theta$ dapat diketahui dengan memisalkan bahwa parameter kontroler θ adalah konstan.

Terdapat banyak alternatif bentuk *loss function* yang dapat digunakan selain bentuk umum yang sudah dijelaskan sebelumnya. Apabila *loss function* yang dipakai adalah sebagai berikut

$$J(\theta) = |e| \quad (2.30)$$

maka metode *gradient* yang dapat diperoleh adalah

$$\frac{d\theta}{dt} = -\gamma \frac{\partial e}{\partial \theta} \text{sign } e \quad (2.31)$$

Metode *MRAS* yang pertama kali diimplementasikan menggunakan formulasi *gradient* diatas. Banyak juga kemungkinan bentuk yang bisa diterapkan, sebagai contoh adalah

$$\frac{d\theta}{dt} = -\gamma \text{sign} \left(\frac{\partial e}{\partial \theta} \right) \text{sign } e \quad (2.32)$$

Bentuk seperti ini sering disebut sebagai *sign-sign algorithm*. Bentuk diskrit dari algoritma ini sudah diterapkan pada dunia telekomunikasi yang membutuhkan implementasi yang sederhana dan perhitungan yang cepat.

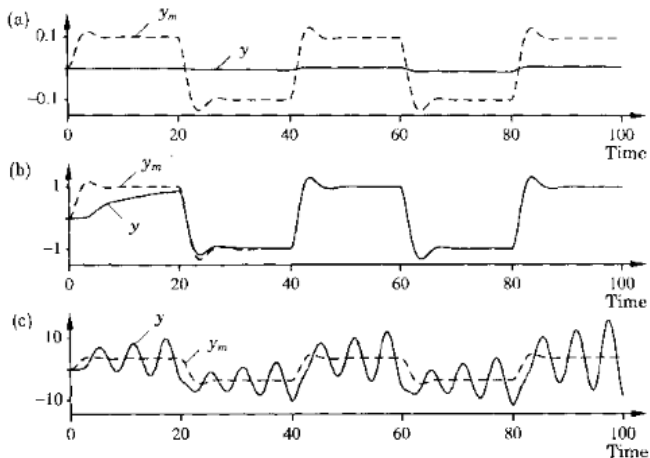
Didalam penggunaan *MRAS* dengan metode *MIT Rule*, terdapat beberapa hal penting yang harus diperhatikan agar proses adaptasi dapat berjalan sesuai dengan yang seharusnya. Salah satu hal tersebut adalah penentuan *adaptation gain* (γ). *Adaptation gain* (γ) akan menentukan kecepatan konvergensi dari perubahan parameter kontroler. Karena pada dasarnya, didalam *MRAS* ini, parameter kontroler akan terus diubah oleh *adjustment mechanism* sampai mencapai suatu nilai tertentu (*convergence*) pada saat *loss function J* sudah minimal. Sehingga sangat penting untuk menentukan nilai yang mungkin digunakan pada *adaptation gain* ini. Secara intuitif, kita dapat menduga bahwa kecepatan perubahan parameter kontroler akan cenderung lambat apabila nilai dari *adaptation gain* kecil. Sehingga dapat dilihat bahwa kecepatan konvergensi parameter kontroler akan sebanding dengan peningkatan nilai *adaptation gain* yang digunakan. Akan tetapi,

terkadang untuk nilai *adaptation gain* yang terlalu besar, pola perubahan nilai parameter kontroler akan sulit untuk dilihat dan diprediksi.

2.2.3 Normalized MIT Rule[2]

Dari beberapa percobaan yang sudah ada, dapat diketahui bahwa sistem dapat menjadi tidak stabil apabila nilai dari *adaptation gain* terlalu besar atau diberi nilai *input* yang relatif besar secara tiba-tiba. Berikut merupakan contoh respon dari sistem yang mengalami kecenderungan untuk menjadi tidak stabil ketika diberi sinyal input yang relatif besar. Sistem yang diuji memiliki bentuk fungsi transfer sebagai berikut

$$G(s) = \frac{1}{s^2 + s + 1} \quad (2.33)$$



Gambar 2. 17 Respon sistem dengan *input* sinyal persegi amplitudo (a) 0.1, (b) 1, (c) 3.5 dan $\gamma = 0.1$ [2]

Dari gambar tersebut terlihat bahwa kecepatan konvergensi output suatu sistem bergantung pada besar dari sinyal *input* yang diberikan. Terlihat juga bahwa sistem akan menjadi tidak stabil saat diberi *input* dengan amplitudo sebesar 3.5. Contoh tersebut memperlihatkan bahwa pemilihan *adaptation gain* sangat krusial dan nilai yang dipilih juga bergantung pada level dari sinyal. Oleh karena itu, penting dilakukan modifikasi algoritma yang sudah ada sehingga ketergantungan dengan

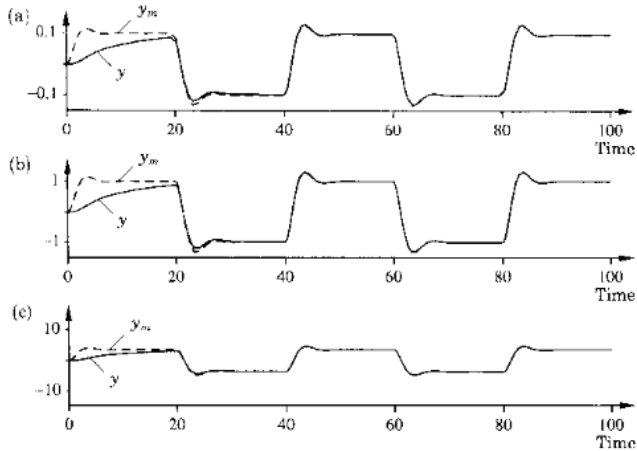
level sinyal dapat diperkecil. Untuk mencapai kondisi tersebut, *MIT Rule* dapat ditulis ulang menjadi bentuk persamaan (2.34).

$$\frac{d\theta}{dt} = \gamma\varphi e \quad (2.34)$$

dimana nilai dari φ dapat diketahui dengan mengacu pada persamaan (2.29) *MIT Rule* yang sudah diperoleh sebelumnya yaitu $\varphi = -\partial e / \partial \theta$. Dari sini, akan dikenalkan model persamaan dari *adjustment mechanism* yang telah dimodifikasi sebagai berikut

$$\frac{d\theta}{dt} = \frac{\gamma\varphi e}{\alpha + \varphi^T \varphi} \quad (2.35)$$

dimana parameter $\alpha > 0$ dipakai untuk menghindari kesulitan dalam perhitungan ketika nilai φ kecil. Perhatikan bahwa penulisan persamaan diatas sudah dibuat sedemikian rupa sehingga dapat juga digunakan ketika θ direpresentasikan dalam bentuk vektor, sehingga φ juga merupakan vektor yang memiliki dimensi yang sama. Bentuk persamaan *adaptation rule* yang baru tersebut selanjutnya akan dikenal sebagai *normalized MIT rule*. Keunggulan dari *MIT Rule* yang ternormalisasi ini adalah sensitivitas terhadap level dari sinyal *input* lebih kecil bila dibandingkan dengan *MIT Rule* yang biasa. Sehingga performansi dari sistem dapat meningkat, peningkatan respon sistem yang menggunakan *normalized MIT rule* dapat dilihat pada grafik berikut



Gambar 2. 18 Respon sistem dengan *input* sinyal persegi amplitudo (a) 0.1, (b) 1, (c) 3.5, $\alpha = 0.001$, dan $\gamma = 0.1$ [2]

Dari gambar tersebut, terlihat bahwa respon dari sistem akan cenderung lebih stabil walaupun level dari sinyal *input* ditingkatkan.

2.3 Kontroler PID (*Proportional-Integral-Derivative*)[7]

2.3.1 Pengenalan

Kontroler PID pada dasarnya memiliki beberapa fungsi penting yang diantaranya adalah sebagai penyedia umpan-balik (*feedback*), memiliki kemampuan untuk mengeliminasi *offset* pada saat *steady state* dengan menggunakan aksi integral, dan juga dapat mengantisipasi kondisi yang akan datang dengan menggunakan aksi derivatif. Kontroler PID sudah cukup banyak digunakan pada berbagai masalah pengontrolan, terutama ketika dinamika pada proses cukup jinak dan spesifikasi performansi yang diinginkan tidak terlalu tinggi. Kontroler PID sudah banyak ditemukan di dunia industri. Kontroler ini memiliki cukup banyak variasi bentuknya. Kontroler PID senduru merupakan salah satu bahan penting yang digunakan pada *distributed control system*. Kontroler ini juga banyak ditanamkan untuk sistem kontrol yang memiliki tujuan spesial (*special-purposed*). Didalam kontrol proses, lebih dari 95% skema *loop* kontrol yang ada merupakan tipe PID, dimana kebanyakan memiliki konfigurasi PI.

Kontroler PID juga sering dikombinasikan dengan *logic, sequential machines, selectors*, dan beberapa blok fungsi yang sederhana untuk membangun suatu sistem otomasi yang rumit yang digunakan pada dunia pembangkit energi, transportasi, dan manufaktur. Banyak juga strategi kontrol lanjutan yang disusun dengan kontroler PID, seperti *model predictive control*. Kontroler PID juga banyak digunakan pada level terendah dimana kontroler *multivariable* memberikan *setpoint* kepada kontroler pada level yang rendah.

Kontroler ini sudah berhasil bertahan walaupun media yang digunakan untuk mengimplementasikan kontroler PID sudah banyak berubah, yang dimulai dari pemakaian untuk kontrol pneumatik secara mekanik hingga pengontrolan menggunakan mikroprosesor dengan *electronic tubes, transistors, dan integrated circuit*. Secara praktiknya, semua kontroler PID saat ini sudah diimplementasikan didalam mikroprosesor. Ini dapat memberikan kesempatan kepada pengguna untuk bisa menggunakan fitur tambahan seperti *auto-tuning, gain scheduling, dan continuous adaptation*. Dengan menggunakan *auto-tuning*, maka parameter pada kontroler dapat detuning secara otomatis sesuai dengan spesifikasi yang ditentukan. Sedangkan untuk *adaptation scheme*, parameter pada kontroler akan terus mengalami pembaharuan sesuai dengan kondisi respon *realtime* dari sistem. Tuning dan adaptasi pada kontroler ini dapat dilakukan dengan berbagai cara.

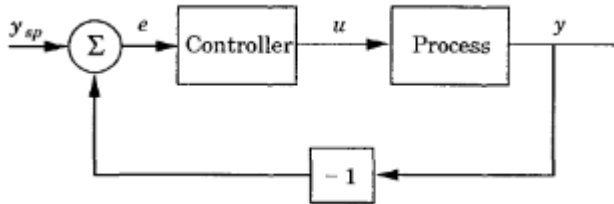
2.3.2 Prinsip Umpan Balik

Ide dalam penerapan umpan balik (*feedback*) cukup sederhana tetapi sangat *powerful*. Prinsip ini juga memiliki pengaruh yang cukup mendalam pada dunia teknologi. Penerapan prinsip umpan balik ini menghasilkan terobosan yang sangat besar didunia kontrol, komunikasi, dan instrumentasi. Kesederhanaan tersebut dapat diasumsikan bahwa umpan balik akan membuat suatu mekanisme yang sedemikian rupa hingga *process variable* pada suatu sistem akan meningkat ketika *manipulated variable* yang ada juga meningkat. Prinsip dari umpan balik ini dapat dinyatakan sebagai berikut:

“Peningkatan nilai *manipulated variable* terjadi ketika *process variable* memiliki nilai lebih kecil dibandingkan *setpoint* dan akan meurun ketika *process variable* memiliki nilai yang lebih besar daripada *setpoint*.”

Konfigurasi umpan balik seperti ini dikenal sebagai *negative feedback* karena *manipulated variable* memiliki arah yang berlawanan

dengan *process variable*. Prinsip umpan balik ini dapat digambarkan sesuai dengan diagram blok berikut



Gambar 2. 19 Diagram blok sistem dengan umpan balik *negative*[7]

Didalam diagram blok tersebut, proses dan kontroler direpresentasikan sebagai suatu *box* dengan arah panah yang menggambarkan *input* dan *output*. Terlihat juga bahwa terdapat suatu symbol khusus yang merepresentasikan penjumlahan sinyal. Didalam diagram blok tersebut ditunjukkan bahwa proses dan kontroler disusun dalam suatu skema *closed-feedback loop*. Kehadiran dari tanda *sign-reversing block* menandakan bahwa umpan balik berjenis negatif.

Alasan penggunaan skema umpan balik ini adalah untuk membuat nilai dari *process variable* sama dengan nilai *setpoint* yang sudah ditetapkan walaupun terdapat gangguan pada sistem.

2.3.3 Struktur Kontroler PID

Struktur algoritma dari kontroler PID secara umum dapat dituliskan sesuai dengan persamaan (2.36).

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.36)$$

dimana u merupakan *control variable* dan e merupakan sinyal *error* dimana ($e = y_{sp} - y$). *Control variable* merupakan jumlahan dari tiga aspek, yaitu *P-term* yang proporsional terhadap *error*, *I-term* yang proporsional terhadap integral dari *error*, dan *D-term* yang proporsional terhadap turunan dari *error*. Parameter-parameter kontroler yang ada pada kontroler PID adalah *proportional gain* K , *integral time* T_i , dan *derivative time* T_d .

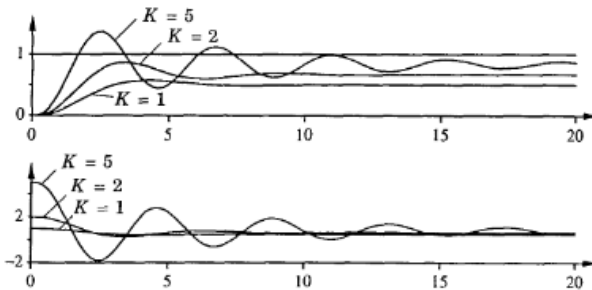
- Aksi Proporsional

Dalam sebuah kontroler proporsional yang murni, persamaan dari kontroler dapat dituliskan sebagai persamaan (2.37).

$$u(t) = Ke(t) + u_b \quad (2.37)$$

Aksi kontrol yang ada memiliki hubungan proporsional terhadap *control error*. Variabel u_b merupakan *bias*. Ketika *error* pada kontroler besarnya sama dengan nol, maka nilai dari *control variable* yang ada sama dengan *bias* $u(t) = u_b$. *Bias* ini biasanya bernilai tertentu $(u_{mac} + u_{min})/2$, tetapi terkadang bisa juga diatur secara manual sehingga *stationary control* yang ada bernilai nol ketika diberikan nilai *setpoint* tertentu.

Contoh dari penerapan kontroler proporsional dapat diilustrasikan kedalam gambar berikut



Gambar 2. 20 Simulasi *closed-loop system* menggunakan kontroler proporsional[7]

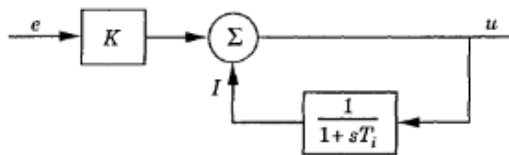
Gambar tersebut menunjukkan perilaku dari sistem ketika diberikan sinyal *input* berupa *unit step*. Untuk fungsi transfer dari plant yaitu $G(s) = 1/(s + 3)^3$. Dengan *gain* kontroler $K = 1$, didapatkan bahwa *error* yang ada sebesar 50%. Apabila dinaikkan nilai *gain* kontrolernya, maka *error* yang ada juga akan semakin mengecil. Terlihat juga bahwa respon dari *plant* akan semakin berosilasi seiring dengan peningkatan *gain* kontroler yang ada.

- Aksi Integral

Fungsi utama yang dimiliki komponen integral adalah untuk memastikan bahwa nilai *process variable* sama dengan nilai *setpoint* pada saat *steadystate*. Dengan hanya menggunakan kontroler proporsional, normalnya akan terdapat *error* pada saat *steady*. Dengan menggunakan aksi integral, sinyal *error* positif yang bernilai kecil akan terus meningkatkan nilai dari sinyal kontrol, dan sinyal *error* negatif yang bernilai kecil juga akan terus menurunkan nilai dari sinyal kontrol tidak peduli seberapa kecil nilai *error* tersebut. Pernyataan ini menunjukkan bahwa secara teori *steady-state error* yang ada pada kontroler dengan aksi integral akan sama dengan nol. Jika diasumsikan bahwa sistem dalam kondisi *steady state* dengan nilai sinyal kontrol yang konstan (u_0) dan nilai *error* yang konstan (e_0), maka persamaan sinyal kontrol yang ada dapat dituliskan sebagai persamaan (2.38).

$$u_0 = K(e_0 + \frac{e_0}{T_i} t) \quad (2.38)$$

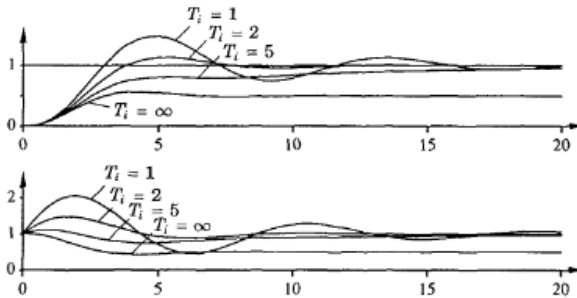
selama nilai $e_0 \neq 0$, ini jelas terlihat bertentangan dengan asumsi yang menyatakan bahwa sinyal kontrol bernilai konstan. Kontroler yang memiliki aksi integral akan selalu memberikan nilai *error steady state* sama dengan nol. Aksi integral juga dapat diilustrasikan sebagai suatu perangkat yang secara otomatis mereset nilai *bias* u_b yang terdapat pada kontroler proporsional. Pernyataan ini dapat diilustrasikan dengan diagram blok berikut



Gambar 2. 21 Implementasi aksi integral sebagai *feedback* positif disekitar *lag*[7]

Pada gambar tersebut ditunjukkan bahwa kontroler proporsional memiliki reset yang di-*adjust* secara otomatis. Proses *adjustment* ini dibuat dengan mengumpukan balikan sinyal yang sudah difilter nilainya dari *ouput* menuju *summing point* pada kontroler. Ini merupakan cikal

bakal dari kontroler dengan aksi integral, atau yang bisa disebut juga sebagai “*automatic reset*”. Contoh dari penerapan kontroler integral dalam sebuah sistem dapat ditunjukkan pada gambar berikut



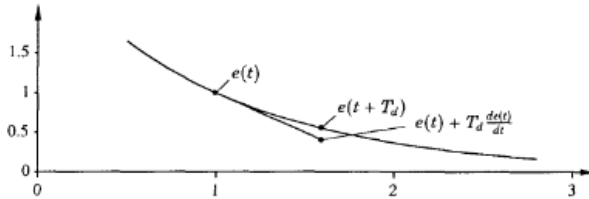
Gambar 2. 22 Simulasi *closed-loop system* menggunakan kontroler integral, pada diagram yang pertama menunjukkan respon dari sistem sedangkan untuk diagram kedua menunjukkan sinyal kontrol[7]

Simulasi diatas merupakan contoh penerapan kontroler PI (*Proportional-Integral*) pada sistem yang memiliki fungsi transfer $G(s) = 1/(s + 3)^3$. Nilai *gain* proporsional konstan untk setiap pengujian $K = 1$. Nilai *integration time* $T_i = \infty$ merepresentasikan kontroler proporsional murni yang memberikan respon sistem yang memiliki nilai *error steady state* 50%. Nilai *error steady state* akan menghilang ketika T_i memiliki nilai tertentu. Ketika nilai *integration time* besar, respon dari sistem akan secara lambat menuju ke *setpoint*. Sedangkan ketika *integration time* diperkecil akan membuat respon lebih cepat menuju *setpoint* tetapi diikuti dengan terjadinya osilasi.

- Aksi Derivatif

Tujuan utama dari aksi derivatif pada suatu kontroler adalah untuk meningkatkan stabilitas dari sistem *closed-loop*. Kondisi ketidakstabilan yang dimaksud dapat dijelaskan sebagai berikut, dikarenakan sifat dinamis yang dimiliki suatu sistem, maka akan memerlukan beberapa waktu agar perubahan yang terdapat pada *control variable* dapat diketahui didalam *output* dari suatu roses. Karena itu, kontroler akan terlambat dalam melakukan koreksi *error*. Aksi dari suatu kontroler yang memiliki kombinasi aksi proporsional dan derivatif dapat diinterpretasikan sebagai kontrol yang proporsional terhadap *process*

output yang diprediksi dimana prediksi ini dibuat dengan melakukan ekstrapolasi *error* yang ada menggunakan *tangent* dari kurva *error*.

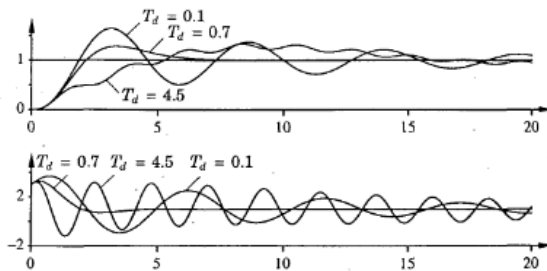


Gambar 2. 23 Interpretasi aksi derivatif sebagai kontrol prediksi menggunakan ekstrapolasi linier[7]

Struktur umum dari kontroler PD (*Proportional-Derivative*) dapat dituliskan sebagai persamaan (2.39).

$$u(t) = K(e(t) + T_d \frac{de(t)}{dt}) \quad (2.39)$$

Sinyal kontrol yang ada akan proporsional terhadap hasil estimasi terhadap *error* pada waktu T_d kedepan, dimana estimasi tersebut didapat dari ekstrapolasi linier. Contoh dari penerapan kontroler PID dapat dilihat pada grafik respon berikut



Gambar 2. 24 Simulasi *closed-loop system* menggunakan kontroler derivatif, pada diagram yang pertama menunjukkan respon dari sistem sedangkan untuk diagram kedua menunjukkan sinyal kontrol[7]

Simulasi diatas merupakan contoh dari penerapan kontroler PID pada sistem yang memiliki fungsi transfer $G(s) = 1/(s + 3)^3$. Nilai dari *gain proportional* dan *integration time* dibuat konstan yaitu $K = 1$ dan $T_i = 2$, kemudian nilai *derivative time* T_d diubah-ubah. Ketika $T_d = 0$, maka kontroler akan berbentuk PI murni. *Output* dari sistem *closed-loop* mengalami osilasi dengan menggunakan parameter yang dipilih tersebut. *Initially damping* akan meningkat seiring meningkatnya *derivative time*, dan akan kembali turun ketika *derivative time* bernilai terlalu tinggi.

2.4 Software Pemrograman *LabVIEW*[4]

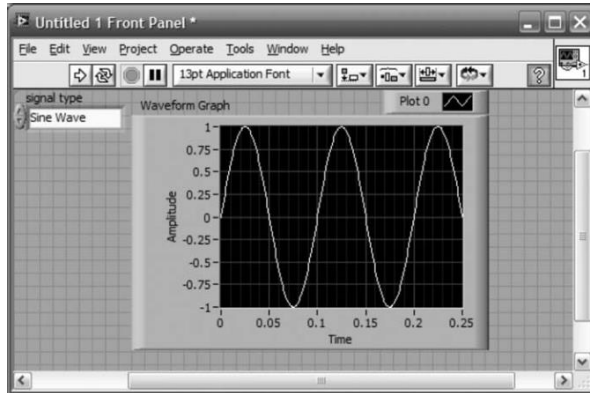
2.4.1 Pengenalan

LabVIEW merupakan suatu *software* pemrograman yang biasa digunakan sebagai alternatif *MATLAB*. *LabVIEW* merupakan *software* pemrograman yang berbasis grafik yang menghasilkan *file* pemrograman berupa *Virtual Instrument (VI)*. *VI* merupakan sebuah representasi ulang seluruh *hardware* dan *instrument* yang digunakan dalam bentuk gambar di layar computer, *instrument* yang dapat digunakan contohnya seperti osiloskop dan *waveform generator*. *VI* sendiri terdiri dari berbagai macam *controls* dan *indicators*, dimana kedua hal ini merepresentasikan *input* dan *output* didalam program yang sudah dibuat. *LabVIEW* sendiri terdiri dari dua bagian halaman utama, yaitu *Front Panel* yang merupakan replika dari *hardware* yang digunakan dan terdiri dari *control & indicator*, dan *Block Diagram* yang tersusun dari kode-kode pemrograman untuk menghubungkan *input* dan *output* pada *Front Panel*.

Didalam *Front Panel Window*, terdapat *Controls palette* yang mengandung ikon-ikon yang memiliki angka dan merepresentasikan bagian *controls* dan *indicators*. Ikon tersebut dapat di-*drag* kedalam *Front Panel window* untuk membuat bagian *control* maupun *indicator*. Kode program yang digunakan untuk menghubungkan bagian kontrol dan indikator disusun dalam *Block Diagram* dengan cara yang sama yaitu *drag & drop*.

Salah satu contoh bentuk tampilan dari bagian *Front Panel* pada *LabVIEW* dapat dilihat pada Gambar 2.25, didalam gambar tersebut terdapat *signal generator* yang berperan sebagai *control input* dan juga bagian *waveform* yang berperan sebagai *output indicator*. Selanjutnya,

juga terdapat contoh Gambar 2.26 yang menampilkan *code block* yang terdapat dalam bagian *Block Diagram* pada *LabVIEW*.

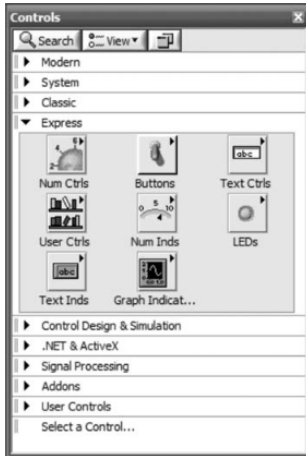


Gambar 2. 25 Tampilan *Front Panel window* pada *LabVIEW*[4]

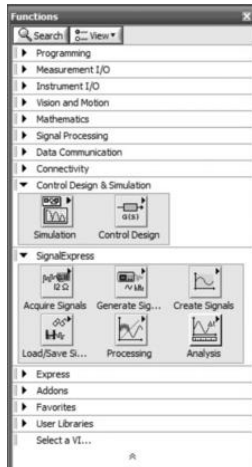


Gambar 2. 26 Tampilan *Block Diagram window* pada *LabVIEW*[4]

Kemudian, untuk gambar dari *palette window* pada *LabVIEW* dapat dilihat pada Gambar 2.27 dan Gambar 2.28.



Gambar 2. 27 Tampilan *Controls Palette* pada *Front Panel LabVIEW*[4]



Gambar 2. 28 Tampilan *Functions Palette* pada *Block Diagram LabVIEW*[4]

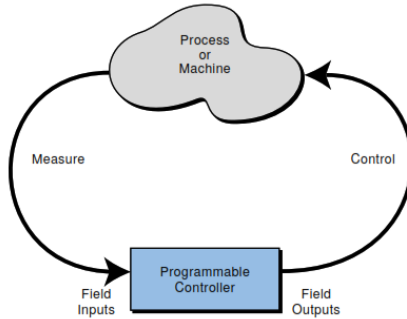
2.4.2 Control System Analysis, Design, and Simulation[4]

Seperti yang dijelaskan pada bagian awal bahwa *LabVIEW* bisa digunakan sebagai alternatif dari *software* pemrograman *MATLAB* yang meliputi analisa, desain, simulasi, *build*, dan *deploy* sistem kontrol. Sebagai pelengkap, maka diperlukan adanya modul tambahan berupa *LabVIEW Control Design and Simulation Module*. Analisa dan desain pada *LabVIEW* dapat dianggap sama seperti menulis kode pada *MATLAB*, sedangkan untuk proses simulasinya sama dengan simulasi menggunakan *Simulink*. Analisa dan desain, serta simulasi pada *LabVIEW* biasanya dimulai dengan *Block Diagram window*, dimana ikon yang merepresentasikan *code blocks* akan dihubungkan satu sama lain. Parameter-parameter yang digunakan oleh kode-kode tersebut juga dapat dengan mudah dipilih dan diganti menggunakan *VI controls* pada *Front Panel window* yang dibentuk bersamaan dengan *drag & drop* ikon pada bagian *Block Diagram*. Sehingga hasil apapun yang diperoleh, akan ditampilkan pada *VI indicators* pada *Front Panel window*.

2.5 Programmable Logic Controller[8]

2.5.1 Pengenalan

Programmable Logic Controller (PLC) merupakan jenis komputer yang masuk kedalam *solid-state members*, yang menggunakan *integrated circuit* dari *electromechanical devices* untuk mengimplementasikan fungsi kontrol. *PLC* memiliki kemampuan untuk menyimpan beberapa instruksi, seperti *sequencing*, *timing*, *counting*, *arithmetic*, *data manipulation*, dan *communication*, untuk mengontrol *industrial machines and processes*. Gambar berikut akan menunjukkan konsep dari pengaplikasian *PLC*.

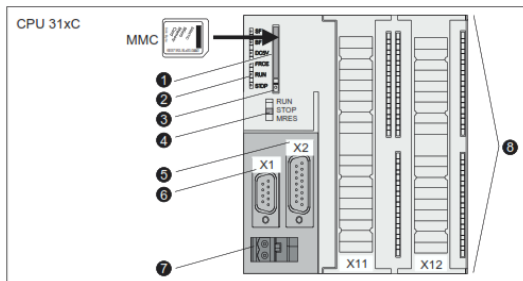


Gambar 2. 29 Diagram konsep aplikasi PLC[8]

Programmable Controller memiliki banyak definisi. Walaupun begitu, PLC dapat diartikan secara sederhana sebagai suatu komputer industri yang dirancang secara khusus arsitektur *central unit* dan rangkain *interface* yang terhubung dengan *field devices (input/output connections to the real world)*.

2.5.2 PLC Siemens S7-300[13]

Tipe PLC yang digunakan kali ini adalah PLC Siemens S7-300 Series yang merupakan salah satu PLC Compact keluaran vendor Siemens. PLC yang dipakai memiliki seri CPU berkode 312-5BD01-0AB0. PLC tersebut memiliki CPU betipe 312C. Didalam CPU tersebut, juga sudah terpasang *digital input output module* yang terdiri dari 10 *digital inputs* dan 6 *digital outputs*. Berikut merupakan tampilan dari elemen dan arsitektur CPU 31_C Series.



Gambar 2. 30 Elemen penyusun CPU tipe 31_C[13]

untuk keterangan tiap-tiap bagian tersebut dapat dilihat pada tabel berikut

Tabel 2. 1 Bagian-bagian *CPU 31_C series*

The diagram illustrates under number	The following operor control / display element of your CPU
(1)	Slot of the Micro Memory Card (MMC)
(2)	Status and error displays
(3)	MMC ejector
(4)	Mode selector switch
(5)	Interface 2 X2 (PtP or DP)
(6)	Interface 1 X1 (MPI)
(7)	Power supply connection
(8)	Connection for the integrated inputs and ouputs

Didalam *PLC Siemens* seri *31_C*, juga disertakan beberapa *LED indicators* yang memudahkan pengguna dalam melihat status dari *CPU*. Berikut merupakan *list* dari beberapa arti *LED indicators* pada *CPU 31_C*.

Tabel 2. 2 Daftar status pada *LED indicators PLC Siemens* seri *CPU 31_C*

LED name	With the LED color	The following CPU states is indicated:
SF	Red	Hardware or software error
BF (only for CPU with DP interface)	Red	Bus error
5 VDC	Green	5V power for CPU and S7-300 bus is OK
FRCE	Yellow	Force job is active
RUN mode	Green	CPU in RUN

		The LED blinks at startup with 2 Hz, in pause with 0.5 Hz
STOP	Yellow	CPU in STOP and PAUSE or startup The LED blinks for reset request with 0.5 Hz, during the reset with 2 Hz

PLC Siemens S7 312C merupakan PLC yang berjenis *compact*, sehingga memiliki *compact CPU* dengan *MPI (Multipoint Interface)*. Selain itu didalamnya juga sudah terdapat 10 *Digital Input* dan 6 *Digital Output* yang memiliki pencacah berkecepatan tinggi dengan frekuensi cacah 10 kHz. *CPU* ini bekerja menggunakan tegangan DC 24 Volt dengan batas bawah yang diperbolehkan sebesar 19.2 VDC sedangkan untuk batas atasnya 28.8 VDC. Pada *CPU 312C*, juga sudah terintegrasi dengan *work memory* sebesar 64 KB dan membutuhkan *load memory* tambahan berjenis *Micro Memory Card (MMC)* yang dapat dipasang pada slot yang tersedia dengan kapasitas maksimal 8 MB. Untuk *processing time* yang diperlukan oleh *CPU 312C* ini dapat diuraikan sebagai berikut

- untuk operasi berjenis *data bit* ≈ 0.1 μs
- untuk operasi berjenis *data word* ≈ 0.24 μs
- untuk operasi aritmetik bertipe *fixed point* ≈ 0.32 μs
- untuk operasi aritmetik bertipe *floating point* ≈ 1.1 μs

Pada *CPU* seri *312C*, konsumsi arus pada saat *open circuit* biasanya berkisar 90 mA sedangkan pada saat kondisi *rated value* bias mencapai 570 mA.

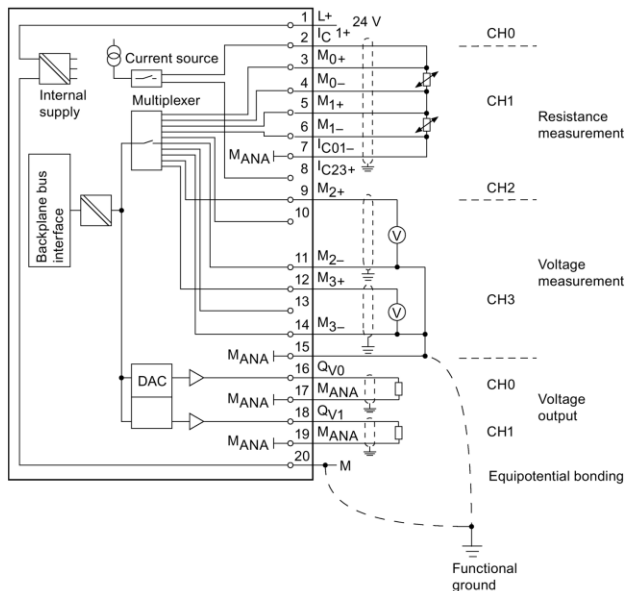
- *I/O Module* pada *PLC Siemens S7-300*
 - *Digital I/O Module*[13]

Pada *PLC Siemens* yang menggunakan *CPU* bertipe *312C*, telah diintegrasikan dengan *digital I/O module* sebanyak 10 *digital input channels* dan 6 *digital output channels*. Masing-masing *channel* tersebut memiliki tegangan kerja 24 VDC. Untuk pengalaman tiap-tiap *channel* secara *default* adalah

I124.0 sampai I125.1 untuk *digital input channels* dan Q124.0 sampai Q124.5 untuk *digital output channels*. Pengalamatan ini juga dapat diubah alamat *bit*-nya sesuai dengan keinginan pengguna menggunakan *setting Hardware Configuration* pada aplikasi *Step-7*.

- *Analog I/O Module*[14]

Pada kali ini, modul input dan output yang dipakai adalah modul berjenis SM 334 AI 4/AO 2 x 12 bit. Modul ini memiliki 4 input pada 2 grup dan 2 output pada 1 grup. Modul analog ini mampu mencacah sinyal analog menjadi bentuk sinyal digital dengan resolusi 12 bits + sign. Sinyal yang dapat diukur oleh modul bertipe SM 334 AI 4/AO 2 x 12 bit ada 3 jenis, yaitu Voltage, Resistance, dan Temperature. Wiring diagram dari modul SM 334 AI 4/AO 2 x 12 bit ini dapat dilihat secara jelas pada Gambar 2.31.



Gambar 2. 31 *Wiring Diagram* modul SM 334 AI 4/AO 2 x 12 Bit[14]

Halaman ini sengaja dikosongkan

BAB 3

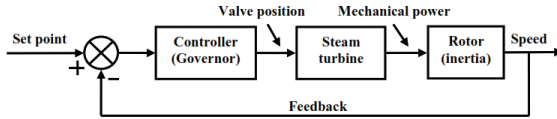
PERANCANGAN SISTEM

Pada perancangan sistem kali ini, akan dibahas mengenai pembentukan fungsi transfer *plant* secara keseluruhan, kemudian juga dibahas mengenai penyederhanaan fungsi transfer dari *plant* yang kemudian akan digunakan untuk merancang *adjustment mechanism* dan merancang referensi model yang digunakan pada kontroler *adaptive PID MRAC*. Selain itu juga akan dibahas mengenai penyusunan fungsi transfer dari *plant steam turbine tandem compound* secara keseluruhan menggunakan *software pemrograman LabVIEW dan Control Design and Simulation Module*. Kemudian juga akan dibuat tampilan *Human Machine Interface (HMI)* untuk mengatur parameter-parameter sistem yang digunakan pada *plant* beserta fungsi-fungsi lain seperti pemilihan jenis *input* yang akan digunakan pada sistem serta plot hasil respons yang didapatkan dari *plant*. Didalam bab ini juga akan dijelaskan perancangan *virtual plant steam turbine tandem compound* pada *LabVIEW* beserta proses diskritisasi *plant* tersebut. Dan diakhiri dengan pendesainan skema kontrol *adaptive PID MRAC* secara penuh yang termasuk desain pada *LabVIEW, IBH OPC Server, dan Step 7 PLC Siemens S7 300*.

3.1 Perancangan *Plant Steam Turbine Tandem Compound* Pengaturan Kecepatan

3.1.1 Menentukan *Transfer Function Plant* Secara Keseluruhan

Sistem kontrol pada *steam turbine* merupakan salah satu contoh sistem pengaturan *closed-loop*. Dimana bentuk sederhananya adalah membuat putaran *rotor* pada turbin berputar dengan kecepatan konstan. Kontroler akan membaca kecepatan putaran *shaft rotor*, kemudian akan dibandingkan kecepatan aktual dengan *set point* yang diinginkan. Apabila terdapat perbedaan nilai diantara keduanya, maka kontroler berupa *governor* akan mengirimkan sinyal kontrol untuk mengoperasikan aktuator *steam valve*, sehingga kecepatan aktual yang terukur akan diatur sampai mendekati kecepatan *set point*.



Gambar 3. 1 Diagram blok pengaturan kecepatan *steam turbine*[11]

Dari gambar diagram blok diatas, terlihat terdapat tiga komponen penyusun skema kontrol kecepatan *steam turbine* secara *closed loop*. Pada bagian *governor* akan terdapat *speed relay* dan *servomotor* sebagai aktuator, kemudian *steam turbine* akan dibagi kedalam tiga *section*, dan yang terakhir ada pada *rotor* yang akan terdapat proses konversi torsi menjadi kecepatan. Seperti dijelaskan sebelumnya, setiap bagian didalam turbin dapat dituliskan sebagai fungsi transfer $H_{HP}(s)$, $H_{MP}(s)$, dan $H_{LP}(s)$ yang bergantung pada suatu *time constant* $T_{HP}(s)$, $T_{MP}(s)$, dan $T_{LP}(s)$. Sehingga, bentuk sederhana dari *steam turbine plant* dapat dituliskan sebagai fungsi transfer sebagai persamaan (3.1).

$$H_{turbin}(s) = \frac{T_m(s)}{F_{in}(s)} = \frac{K_{HP}}{T_{HP}s + 1} + \frac{K_{MP}}{(T_{HP}s + 1)(T_{MP}s + 1)} + \frac{K_{LP}}{(T_{HP}s + 1)(T_{MP}s + 1)(T_{LP}s + 1)} \quad (3.1)$$

kemudian untuk fungsi transfer yang mengubah torsi pada turbin menjadi kecepatan putar dapat didefinisikan sebagai fungsi transfer beban (load) yang dapat diperoleh dari hukum Newton kedua untuk benda putar.

$$\begin{aligned} \tau &= I\alpha \\ \tau &= I \frac{d\omega}{dt} \\ \frac{1}{I} \tau dt &= d\omega \\ \frac{1}{I} \int \tau dt &= \omega \end{aligned} \quad (3.2)$$

dengan melakukan transformasi Laplace pada persamaan (3.2), maka dapat diperoleh persamaan fungsi transfer dari $H_L(s)$.

$$\frac{1}{Is} \tau = \omega$$

$$\frac{\omega}{\tau} = \frac{1}{I_s}$$

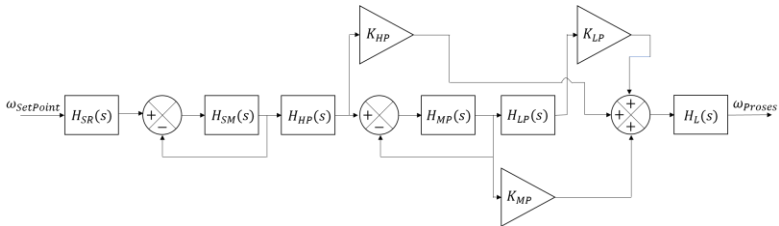
$$H_L(s) = \frac{\omega(s)}{T_m(s)} = \frac{1}{T_L s} \quad (3.3)$$

Pada penelitian kali ini, digunakan sistem kontrol yang menggunakan aktuator bertipe mechanical-hydraulic dimana pada aktuator tersebut akan terdiri dari speed relay yang terhubung dengan servomotor yang dapat dituliskan dengan persamaan fungsi transfer sebagai persamaan (3.4) dan (3.5).

$$H_{SR} = \frac{1}{T_{SR} s + 1} \quad (3.4)$$

$$H_{SM} = \frac{1}{T_{SM} s} \quad (3.5)$$

Sehingga akan didapatkan diagram blok pengaturan kecepatan pada steam turbine bertipe tandem compound secara keseluruhan sesuai dengan gambar berikut:



Gambar 3. 2 Diagram blok *steam turbine* secara keseluruhan

Untuk fungsi transfer secara keseluruhan, dapat dituliskan dengan persamaan (3.6).

$$H_{Total}(s) = \frac{1}{T_{SR} s + 1} \cdot \frac{1}{T_{SM} s + 1} \cdot \left(\frac{K_{HP}}{T_{HP} s + 1} + \frac{K_{MP}}{(T_{HP} s + 1)(T_{MP} s + 1)} + \frac{K_{LP}}{(T_{HP} s + 1)(T_{MP} s + 1)(T_{LP} s + 1)} \right) \cdot \frac{1}{T_L s} \quad (3.6)$$

3.1.2 Penentuan Parameter Plant

Didalam pengaturan kecepatan *steam turbine*, parameter-parameter *plant* yang perlu didefinisikan adalah $K_{HP}(s)$, $K_{MP}(s)$, $K_{LP}(s)$, $T_{HP}(s)$, $T_{MP}(s)$, $T_{LP}(s)$, $T_{SR}(s)$, $T_{SM}(s)$, dan $T_L(s)$. Parameter yang dipakai pada penelitian ini akan dibuat pada dua kondisi yang berbeda yaitu kondisi ideal dan penurunan berdasarkan data *heat balance* pada turbin. Untuk kondisi ideal (nominal), parameter-parameter turbin dapat dituliskan pada tabel berikut :

Tabel 3. 1 Parameter *steam turbine tandem compound*

Parameter	HP Turbin	MP Turbin	LP Turbin
Fraksi Daya	$K_{HP}=0.3$	$K_{MP}=0.3$	$K_{LP}=0.4$
Konstanta Waktu (sekon)	$T_{HP}=0.25$	$T_{MP}=7.5$	$T_{LP}=0.5$

sedangkan untuk konstanta waktu dari servomotor, speed relay, dan load secara berurutan adalah $T_{SM}(s)=0.3 \text{ sec}$, $T_{SR}(s)=0.1 \text{ sec}$, dan $T_L(s)=12 \text{ sec}$. [11]

Kemudian untuk parameter kedua menggunakan data *heat balance* pada *steam turbine* berkapasitas 200-MW [12] yang secara ringkas parameter-parameter tersebut dapat dituliskan dalam tabel dibawah

Tabel 3. 2 Parameter pada *governor* dan *load* menggunakan data *heat balance*

T_{SR}	0.1 sec
T_{SM}	0.25 sec
T_L	12 sec

Tabel 3. 3 Parameter *turbine sections* menggunakan data *heat balance*

K_{HP}	0.31
K_{MP}	0.5
K_{LP}	0.19
T_{HP}	0.396 sec
T_{MP}	6.84 sec
T_{LP}	0.052 sec

Sehingga, secara keseluruhan fungsi transfer dari *plant* pengaturan kecepatan *steam turbine* dalam kondisi nominal dapat dituliskan sebagai berikut :

$$H_{TotIdeal}(s) = \frac{1.125s^2 + 2.25s + 1}{0.3375s^6 + 6.57s^5 + 41.82s^4 + 108.96s^3 + 103.8s^2 + 12s}$$

Sedangkan apabila menggunakan parameter dari *heat balance data*, akan didapatkan fungsi transfer dari sistem sebagai berikut :

$$H_{TotHeatBal}(s) = \frac{0.1103s^2 + 2.163s + 1}{0.04225s^6 + 1.517s^5 + 16.83s^4 + 67.93s^3 + 91.66s^2 + 12s}$$

3.1.3 Penyederhanaan Bentuk *Transfer Function Plant* Menjadi Bentuk *Plant Orde-2*

Untuk menyusun kontroler *adaptive PID MRAC*, akan digunakan bentuk *transfer function plant* yang lebih sederhana untuk memudahkan dalam proses penurunan *adjustment mechanism* tanpa mengubah bentuk respon sistem yang asli. Cara yang diambil adalah dengan memisahkan persamaan *plant* dengan parameter kondisi ideal $H_{TotIdeal}(s)$ kedalam bentuk-bentuk dasarnya menggunakan *partial fraction*, kemudian akan dicari dua *pole* yang paling dominan yang membentuk *transfer function plant* secara keseluruhan sehingga menghasilkan bentuk respon dan karakteristiknya tidak akan jauh berbeda dengan respon *plant* secara keseluruhan. Langkah awalnya yaitu memisahkan kembali bentuk pecahan *transfer function plant* secara keseluruhan menjadi 6 pecahan-pecahan asli yang membentuk *transfer function* tersebut, hal ini dikarenakan pada dasarnya $H_{TotIdeal}(s)$ terbentuk oleh kombinasi dari penjumlahan masing-masing proses yang ada didalam turbin yang meliputi proses pada *servomotor*, *speed relay*, *HP turbine*, *MP turbine*, *LP turbine*, dan kemudian diubah menjadi energi mekanik dalam proses *load*. Oleh karena itu, *transfer function plant* secara keseluruhan dapat dibentuk ulang menjadi persamaan (3.7).

$$H_{Totalideal}(s) = \frac{1.125s^2 + 2.25s + 1}{(0.1s + 1)(0.3s + 1)(0.25s + 1)(7.5s + 1)(0.5s + 1)(12s)}$$

$$H_{Totalideal}(s) = \frac{3.3167s^2 + 7.5156s + 2.9408}{(s + 10)(s + 3.33)(s + 4)(s + 0.133)(s + 2)(s)}$$

$$H_{Totalideal}(s) = \frac{A}{T_{SR} \cdot s + 1} + \frac{B}{T_{SM} \cdot s + 1} + \frac{C}{T_{HP} \cdot s + 1} + \frac{D}{T_{MP} \cdot s + 1} + \frac{E}{T_{LP} \cdot s + 1} + \frac{F}{T_L \cdot s}$$

$$H_{Totalideal}(s) = \frac{A}{0.1s + 1} + \frac{B}{0.3s + 1} + \frac{C}{0.25s + 1} + \frac{D}{7.5s + 1} + \frac{E}{0.5s + 1} + \frac{F}{12s} \quad (3.7)$$

$$H_{Totalideal}(s) = \frac{10A}{s + 10} + \frac{3.33B}{s + 3.33} + \frac{4C}{s + 4} + \frac{0.133D}{s + 0.133} + \frac{2E}{s + 2} + \frac{0.083F}{s}$$

untuk mencari nilai-nilai parameter A, B, C, D, E, dan F dapat dilakukan dengan *partial fraction expansion* untuk persamaan yang mengandung pole yang berbeda-beda. Dimana dengan menggunakan metode tersebut, koefisien tiap-tiap *fraction* dapat dicari menggunakan bentuk

$$a_k = \left[\frac{(s + p_k)B(s)}{A(s)} \right]_{s=-p_k} \quad (3.8)$$

maka,

$$\begin{aligned} A &= \frac{1}{10} \left[\frac{(s + 10)(3.3167s^2 + 7.5156s + 2.9408)}{(s + 10)(s + 3.33)(s + 4)(s + 0.133)(s + 2)(s)} \right]_{s=-10} \\ &= \frac{1}{10} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s + 3.33)(s + 4)(s + 0.133)(s + 2)(s)} \right]_{s=-10} \\ &= -0.0008 \end{aligned}$$

$$\begin{aligned} B &= \frac{1}{3.33} \left[\frac{(s + 3.33)(3.3167s^2 + 7.5156s + 2.9408)}{(s + 10)(s + 3.33)(s + 4)(s + 0.133)(s + 2)(s)} \right]_{s=-3.33} \\ &= \frac{1}{3.33} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s + 10)(s + 4)(s + 0.133)(s + 2)(s)} \right]_{s=-3.33} \\ &= -0.0843 \end{aligned}$$

$$\begin{aligned}
C &= \frac{1}{4} \left[\frac{(s+4)(3.3167s^2 + 7.5156s + 2.9408)}{(s+10)(s+3.33)(s+4)(s+0.133)(s+2)(s)} \right]_{s=-4} \\
&= \frac{1}{4} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s+3.33)(s+10)(s+0.133)(s+2)(s)} \right]_{s=-4} \\
&= 0.0598
\end{aligned}$$

$$\begin{aligned}
D &= \frac{1}{0.133} \left[\frac{(s+0.133)(3.3167s^2 + 7.5156s + 2.9408)}{(s+10)(s+3.33)(s+4)(s+0.133)(s+2)(s)} \right]_{s=-0.133} \\
&= \frac{1}{0.133} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s+3.33)(s+4)(s+10)(s+2)(s)} \right]_{s=-0.133} \\
&= -0.5266
\end{aligned}$$

$$\begin{aligned}
E &= \frac{1}{2} \left[\frac{(s+2)(3.3167s^2 + 7.5156s + 2.9408)}{(s+10)(s+3.33)(s+4)(s+0.133)(s+2)(s)} \right]_{s=-2} \\
&= \frac{1}{2} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s+3.33)(s+4)(s+0.133)(s+10)(s)} \right]_{s=-2} \\
&= 0.0186
\end{aligned}$$

$$\begin{aligned}
F &= \frac{1}{0.083} \left[\frac{(s)(3.3167s^2 + 7.5156s + 2.9408)}{(s+10)(s+3.33)(s+4)(s+0.133)(s+2)(s)} \right]_{s=0} \\
&= \frac{1}{0.083} \left[\frac{(3.3167s^2 + 7.5156s + 2.9408)}{(s+3.33)(s+4)(s+0.133)(s+2)(s+10)} \right]_{s=0} = 1
\end{aligned}$$

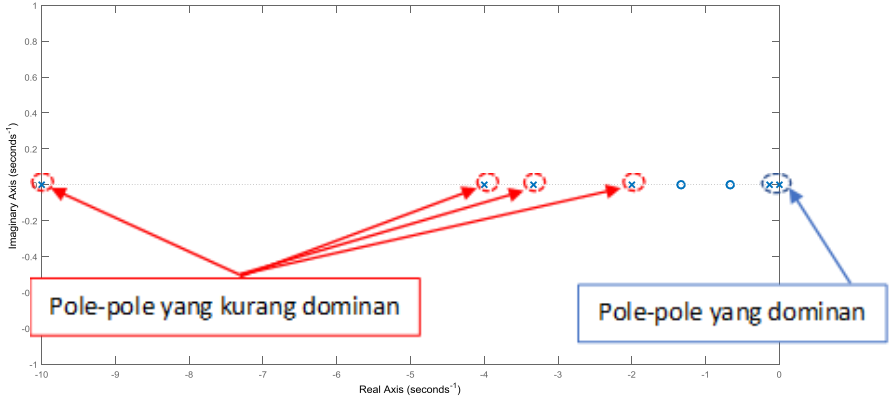
Sehingga, persamaan (3.7) dapat ditulis kembali menjadi

$$H_{rotideal}(s) = -\frac{0.0008}{0.1s+1} - \frac{0.0843}{0.3s+1} + \frac{0.0598}{0.25s+1} - \frac{0.5266}{7.5s+1} + \frac{0.0186}{0.5s+1} + \frac{1}{12s} \quad (3.9)$$

Kemudian, dari keenam pecahan parsial tersebut, kemudian dapat dipilih *pole-pole* yang paling dominan dimana *pole* ini memiliki sifat yang lebih besar 10 kali dibandingkan dengan *pole* yang lain dan letaknya paling dekat dengan sumbu- $y = 0$. *Pole-pole* dominan inilah yang paling menentukan respon yang terbentuk ketika suatu plant diberikan sinyal uji. Dari persamaan diatas, dapat diperoleh bahwa *pole-pole* yang terdapat dalam sistem tersebut adalah sebagai berikut

$$\begin{aligned}
\text{Pole pada plant Steam Turbine : } s_1 = -10, s_2 = -3.33, s_3 = -4, \\
s_4 = -0.13, s_5 = -2, s_6 = 0
\end{aligned}$$

Dari daftar *pole-pole* tersebut, dapat dilihat bahwa *pole* yang paling dominan terletak pada $s_4 = -0.13$ dan $s_6 = 0$ atau secara grafik dapat ditunjukkan pada gambar



Gambar 3.3 Grafik *pole-zero plot* pada *plant steam turbine*

Setelah diketahui nilai dari *pole* yang dominan, maka selanjutnya dilakukan pembentukan *transfer function* baru yang terdiri dari pecahan-pecahan yang mengandung *pole* dominan.

$$H_{Baru}(s) = -\frac{0.5266}{7.5s + 1} + \frac{1}{12s} \quad (3.9)$$

pada pecahan yang mengandung *pole* di nol, dapat digeser sedikit dengan nilai yang cukup kecil sehingga tidak mengubah bentuk respon secara signifikan.

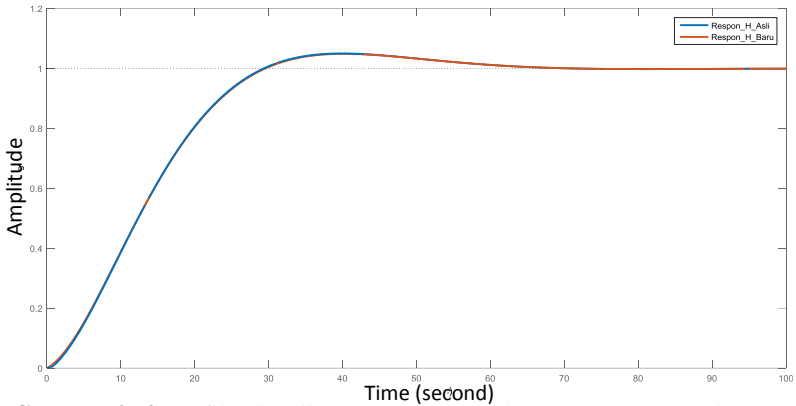
$$H_{Baru}(s) = -\frac{0.5266}{7.5s + 1} + \frac{1}{12s + 0.0001}$$

$$H_{Baru}(s) = \frac{1.1808s + 0.99995}{90s^2 + 12.00075s + 0.0001}$$

$$H_{Baru}(s) = \frac{0.01312s + 0.01111}{s^2 + 0.13334s + 1.11 \times 10^{-6}} = \frac{a_1s + a_2}{s^2 + b_1s + b_2} \quad (3.10)$$

Kemudian dilakukan pengujian *transfer function* dari *plant* secara *closed loop* menggunakan *input* berupa sinyal *unit step* dan kontroler

proporsional dengan *gain* 1. Hasilnya dapat ditunjukkan pada Gambar 3.4



Gambar 3. 4 Grafik *plot* uji *step response* pada *transfer function* baru

Dari gambar tersebut, respon kedua *transfer function* menunjukkan bentuk yang sama sehingga *transfer function* $H_{Baru}(s)$ dapat digunakan sebagai perwakilan $H_{TotalIdeal}(s)$ untuk menemukan *adjustment mechanism* pada kontroler *adaptive PID MRAC*.

3.2 Penentuan *Adjustment Mechanism* pada kontroler *PID MRAC*

Penyusunan mekanisme menggunakan aturan MIT (*MIT Rule*) dimana pada metode ini, akan dicari nilai parameter yang paling sesuai sehingga menghasilkan *Loss Function* (J) yang minimum. *Loss Function* yang dipakai memiliki bentuk fungsi kuadrat terhadap *error* dimana *error* sendiri adalah perbedaan antara respon dari *plant* dengan respon model yang diinginkan.

$$J(\theta) = \frac{1}{2} e^2, \quad e = y_p - y_m \quad (3.11)$$

Agar J menjadi minimum, maka perubahan parameter yang ada haruslah memiliki arah *gradient* yang negatif sehingga *error* yang ada juga ikut menjadi minimum.

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$$

3.2.1 General MIT Rule

Pada kali ini, parameter-parameter yang diubah adalah Kp , τ_i , dan τ_d . Sehingga perubahan parameter-parameter tiap waktunya adalah

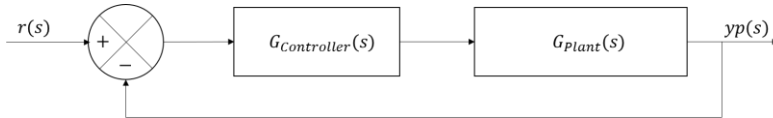
$$\frac{dKp}{dt} = -\gamma \frac{\partial J}{\partial Kp} = -\gamma \frac{\partial J}{\partial e} \frac{\partial e}{\partial yp} \frac{\partial yp}{\partial Kp} \quad (3.13)$$

$$\frac{d\tau_i}{dt} = -\gamma \frac{\partial J}{\partial \tau_i} = -\gamma \frac{\partial J}{\partial e} \frac{\partial e}{\partial yp} \frac{\partial yp}{\partial \tau_i} \quad (3.14)$$

$$\frac{d\tau_d}{dt} = -\gamma \frac{\partial J}{\partial \tau_d} = -\gamma \frac{\partial J}{\partial e} \frac{\partial e}{\partial yp} \frac{\partial yp}{\partial \tau_d} \quad (3.15)$$

Dimana : $\frac{\partial J}{\partial e} = e$, $\frac{\partial e}{\partial yp} = 1$

Dengan menggunakan model fungsi transfer dari *plant* yang sudah disederhanakan, maka dapat dibentuk diagram blok sistem lengkap dengan kontroler PID konvensional untuk proses penurunan *adjustment mechanism*



Dengan menggunakan $G_{Controller}(s) = Kp \left(1 + \frac{1}{\tau_i s} + \tau_d s \right)$ dan $G_{Plant}(s) = \frac{a_1 s + a_2}{s^2 + b_1 s + b_2}$, maka dapat diperoleh bentuk closed loop dari diagram blok tersebut adalah sebagai berikut

$$G_{Closedloop}(s) = \frac{Kp \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \cdot \frac{a_1 s + a_2}{s^2 + b_1 s + b_2}}{1 + \left(Kp \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \cdot \frac{a_1 s + a_2}{s^2 + b_1 s + b_2} \right)}$$

$$G_{closedloop}(s) = \frac{\frac{Kp\tau_i s + Kp + Kp\tau_i\tau_d s^2}{\tau_i s} \cdot \frac{a_1 s + a_2}{s^2 + b_1 s + b_2}}{1 + \left(\frac{Kp\tau_i s + Kp + Kp\tau_i\tau_d s^2}{\tau_i s} \cdot \frac{a_1 s + a_2}{s^2 + b_1 s + b_2} \right)}$$

$$G_{closedloop}(s) = \frac{Kp\tau_i a_1 s^2 + Kp\tau_i a_2 s + a_1 Kp s + a_2 Kp + a_1 Kp\tau_i\tau_d s^3 + a_2 Kp\tau_i\tau_d s^2}{\tau_i s^3 + b_1 \tau_i s^2 + b_2 \tau_i s + Kp\tau_i a_1 s^2 + Kp\tau_i a_2 s + a_1 Kp s + a_2 Kp + a_1 Kp\tau_i\tau_d s^3 + a_2 Kp\tau_i\tau_d s^2}$$

$$G_{closedloop}(s) = \frac{(Kp\tau_i a_1 + Kp\tau_i\tau_d a_2) s^2 + (Kp\tau_i a_2 + a_1 Kp) s + a_2 Kp + a_1 Kp\tau_i\tau_d s^3}{(a_1 Kp\tau_i\tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i\tau_d) s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp) s + Kp a_2}$$

dimana a_1 , a_2 , b_1 , dan b_2 merupakan parameter dari plant yang sudah disederhanakan. Diinginkan agar respon dari sistem tetap disetiap saat walaupun terjadi perubahan parameter sistem, maka dipilihlah referensi model yang dipakai adalah bentuk closed loop dari fungsi transfer dari plant dan PID controller yang dipakai, sehingga $G_{closedloop}(s) = G_{Model}(s)$ dengan nilai parameter-parameter yang ada dibuat konstan sesuai dengan spesifikasi respon yang ingin dihasilkan. Sedangkan untuk menurunkan adjustment mechanism yang akan digunakan, maka parameter kontroler yang ada (Kp , τ_i , dan τ_d) perlu dibuat menjadi variable yang dapat berubah-ubah sepanjang waktu. Hal ini sesuai dengan prinsip MRAC dimana adjustment mechanism akan melakukan penyesuaian terhadap parameter kontroler PID yang dipakai setiap waktu agar mendapatkan respon model sesuai dengan yang diinginkan.

Menggunakan aturan turunan pecahan dengan aturan Quotient Rule yang menyatakan bahwa

$$\left(\frac{u}{v}\right)'(x) = \frac{u'(x)v(x) - u(x)v'(x)}{v^2(x)} \quad (3.16)$$

dan yp merupakan

$$G_{closedloop}(s) = \frac{yp(s)}{r(s)} = \frac{(Kp\tau_i a_1 + Kp\tau_i\tau_d a_2) s^2 + (Kp\tau_i a_2 + a_1 Kp) s + a_2 Kp + a_1 Kp\tau_i\tau_d s^3}{(a_1 Kp\tau_i\tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i\tau_d) s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp) s + Kp a_2}$$

$$yp(s) = \frac{(Kp\tau_i a_1 + Kp\tau_i \tau_d a_2)s^2 + (Kp\tau_i a_2 + a_1 Kp)s + a_2 Kp + a_1 Kp\tau_i \tau_d s^3}{(a_1 Kp\tau_i \tau_d + \tau_i)s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i \tau_d)s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp)s + Kpa_2} \cdot r(s)$$

Maka dapat diperoleh turunan yp terhadap masing-masing parameter kontroler adalah sebagai berikut

$$\frac{\partial yp}{\partial Kp} = \frac{\partial \frac{((Kp\tau_i a_1 + Kp\tau_i \tau_d a_2)s^2 + (Kp\tau_i a_2 + a_1 Kp)s + a_2 Kp + a_1 Kp\tau_i \tau_d s^3) \cdot r}{(a_1 Kp\tau_i \tau_d + \tau_i)s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i \tau_d)s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp)s + Kpa_2}}{\partial Kp}$$

$$\frac{\partial yp}{\partial \tau_i} = \frac{\partial \frac{((Kp\tau_i a_1 + Kp\tau_i \tau_d a_2)s^2 + (Kp\tau_i a_2 + a_1 Kp)s + a_2 Kp + a_1 Kp\tau_i \tau_d s^3) \cdot r}{(a_1 Kp\tau_i \tau_d + \tau_i)s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i \tau_d)s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp)s + Kpa_2}}{\partial \tau_i}$$

$$\frac{\partial yp}{\partial \tau_d} = \frac{\partial \frac{((Kp\tau_i a_1 + Kp\tau_i \tau_d a_2)s^2 + (Kp\tau_i a_2 + a_1 Kp)s + a_2 Kp + a_1 Kp\tau_i \tau_d s^3) \cdot r}{(a_1 Kp\tau_i \tau_d + \tau_i)s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i \tau_d)s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp)s + Kpa_2}}{\partial \tau_d}$$

Dengan menyesuaikan bentuk persamaan seperti yang ada pada persamaan (3.16), maka dapat diperoleh bahwa

$$\frac{\partial yp}{\partial Kp} = \frac{u'(Kp)v(Kp) - u(Kp)v'(Kp)}{v^2(Kp)}$$

$$u = ((Kp\tau_i a_1 + Kp\tau_i \tau_d a_2)s^2 + (Kp\tau_i a_2 + a_1 Kp)s + a_2 Kp + a_1 Kp\tau_i \tau_d s^3) \cdot r(s)$$

$$v = (a_1 Kp\tau_i \tau_d + \tau_i)s^3 + (b_1 \tau_i + Kp\tau_i a_1 + a_2 Kp\tau_i \tau_d)s^2 + (b_2 \tau_i + Kp\tau_i a_2 + a_1 Kp)s + Kpa_2$$

Sehingga,

$$\frac{\partial yp}{\partial Kp} = \frac{((\tau_i a_1 + \tau_i \tau_d a_2)s^2 + (\tau_i a_2 + a_1)s + a_2 + a_1 \tau_i \tau_d s^3) \cdot r \cdot v - ((a_1 \tau_i \tau_d)s^3 + (\tau_i a_1 + a_2 \tau_i \tau_d)s^2 + (\tau_i a_2 + a_1)s + a_2) \cdot u}{v^2}$$

$$\begin{aligned}
&= \frac{\left(((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3) \cdot r \cdot v \right)}{v^2} - \\
&\frac{\left(((a_1 \tau_i \tau_d) s^3 + (\tau_i a_1 + a_2 \tau_i \tau_d) s^2 + (\tau_i a_2 + a_1) s + a_2) \cdot u \right)}{v^2} \\
&= \frac{\left(((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3) \cdot r \right)}{v} - \\
&\frac{\left(((a_1 \tau_i \tau_d) s^3 + (\tau_i a_1 + a_2 \tau_i \tau_d) s^2 + (\tau_i a_2 + a_1) s + a_2) \cdot y p \right)}{v} \\
\frac{\partial y p}{\partial K p} &= \frac{\left(((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3) \cdot (r - y p) \right)}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2}
\end{aligned}$$

Dengan cara yang sama, maka

$$\frac{\partial y p}{\partial \tau_i} = \frac{u'(\tau_i) v(\tau_i) - u(\tau_i) v'(\tau_i)}{v^2(\tau_i)}$$

$$u = \left((K p \tau_i a_1 + K p \tau_i \tau_d a_2) s^2 + (K p \tau_i a_2 + a_1 K p) s + a_2 K p + a_1 K p \tau_i \tau_d s^3 \right) \cdot r(s)$$

$$v = (a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2$$

Sehingga,

$$\begin{aligned}
\frac{\partial y p}{\partial \tau_i} &= \\
&\frac{\left(((K p a_1 + K p \tau_d a_2) s^2 + (K p a_2) s + a_1 K p \tau_d s^3) \cdot r \cdot v - ((a_1 K p \tau_d + 1) s^3 + (b_1 + K p a_1 + a_2 K p \tau_d) s^2 + (b_2 + K p a_2) s) \cdot u \right)}{v^2} \\
&= \frac{\left(((K p a_1 + K p \tau_d a_2) s^2 + (K p a_2) s + a_1 K p \tau_d s^3) \cdot r \cdot v \right)}{v^2} - \\
&\frac{\left(((a_1 K p \tau_d + 1) s^3 + (b_1 + K p a_1 + a_2 K p \tau_d) s^2 + (b_2 + K p a_2) s) \cdot u \right)}{v^2} \\
&= \frac{\left(((K p a_1 + K p \tau_d a_2) s^2 + (K p a_2) s + a_1 K p \tau_d s^3) \cdot r \right)}{v} - \\
&\frac{\left(((a_1 K p \tau_d + 1) s^3 + (b_1 + K p a_1 + a_2 K p \tau_d) s^2 + (b_2 + K p a_2) s) \cdot y p \right)}{v}
\end{aligned}$$

$$\frac{\partial yp}{\partial \tau_i} = \frac{(a_1 K p \tau_d s^3 + (a_1 K p + K p \tau_d) s^2 + K p a_2 s) \cdot (r - yp) - (s^3 + b_1 s^2 + b_2 s) \cdot yp}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2}$$

Dan,

$$\frac{\partial yp}{\partial \tau_d} = \frac{u'(\tau_d) v(\tau_d) - u(\tau_d) v'(\tau_d)}{v^2(\tau_d)}$$

$$u = ((K p \tau_i a_1 + K p \tau_i \tau_d a_2) s^2 + (K p \tau_i a_2 + a_1 K p) s + a_2 K p + a_1 K p \tau_i \tau_d s^3) \cdot r(s)$$

$$v = (a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2$$

Sehingga,

$$\frac{\partial yp}{\partial \tau_d} = \frac{\left(((K p \tau_i a_2) s^2 + a_1 K p \tau_i s^3) \cdot r \cdot v - ((a_1 K p \tau_i) s^3 + (a_2 K p \tau_i) s^2) \cdot u \right)}{v^2}$$

$$= \frac{\left((K p \tau_i a_2) s^2 + a_1 K p \tau_i s^3 \right) \cdot r \cdot v}{v^2} - \frac{\left((a_1 K p \tau_i) s^3 + (a_2 K p \tau_i) s^2 \right) \cdot u}{v^2}$$

$$= \frac{\left((K p \tau_i a_2) s^2 + a_1 K p \tau_i s^3 \right) \cdot r}{v} - \frac{\left((a_1 K p \tau_i) s^3 + (a_2 K p \tau_i) s^2 \right) \cdot yp}{v}$$

$$\frac{\partial yp}{\partial \tau_d} = \frac{(a_1 K p \tau_i) s^3 + (a_2 K p \tau_i) s^2 \cdot (r - yp)}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2}$$

Sehingga, parameter-parameter kontroler yang digunakan untuk kontroler adaptive PID MRAC dapat dituliskan kembali menjadi seperti berikut:

$$K p = \int_0^t -\gamma e \frac{\left((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3 \right) \cdot (r - yp)}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2} dt$$

$$\tau_i = \int_0^t -\gamma e \frac{(a_1 K p \tau_d s^3 + (a_1 K p + K p \tau_d) s^2 + K p a_2 s) \cdot (r - yp) - (s^3 + b_1 s^2 + b_2 s) \cdot yp}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2} dt$$

$$= \int_0^{\tau_d} -\gamma e^{\frac{((a_1 K p \tau_i) s^3 + (a_2 K p \tau_i) s^2) \cdot (r - \gamma p)}{(a_1 K p \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + K p \tau_i a_1 + a_2 K p \tau_i \tau_d) s^2 + (b_2 \tau_i + K p \tau_i a_2 + a_1 K p) s + K p a_2}} dt$$

3.2.2 Normalized MIT Rule (Modified MIT Rule)

Bentuk persamaan dari skema ini adalah sebagai berikut:

$$\frac{d\theta}{dt} = \frac{\gamma e \varphi}{\alpha + \varphi^T \varphi} \quad (3.17)$$

Dengan, φ merupakan $-\frac{\partial e}{\partial \theta}$ dan $\alpha > 0$ untuk menghindari *difficulty of zero division* ketika φ bernilai kecil sehingga pada penelitian kali ini dipilih nilai α sama dengan satu.

Apabila dibandingkan, skema *adjustment mechanism* yang terdapat pada *MIT Rule* standard dengan *Normalized MIT Rule* memiliki sedikit perbedaan, pada *Normalized MIT Rule* terdapat pembagian skema *MIT Rule* yang lama ($\gamma e \varphi$) dengan $\alpha + \varphi^T \varphi$. Penjumlahan ini dilakukan agar perubahan nilai parameter yang ada tidak terlalu cepat (tidak terlalu sensitif) sehingga mengurangi ketidakstabilan yang dapat terjadi ketika terjadi perubahan input atau *adaptation gain* yang cukup besar.

Bentuk skema *adjustment mechanism* yang terdapat pada *normalized MIT Rule* dapat dituliskan dengan persamaan-persamaan berikut :

$$\frac{dKp}{dt} = \frac{-\gamma e \frac{\partial \gamma p}{\partial Kp}}{\alpha + \left(\frac{\partial \gamma p}{\partial Kp}\right)^2} \quad (3.18)$$

$$\frac{d\tau_i}{dt} = \frac{-\gamma e \frac{\partial \gamma p}{\partial \tau_i}}{\alpha + \left(\frac{\partial \gamma p}{\partial \tau_i}\right)^2} \quad (3.19)$$

$$\frac{d\tau_d}{dt} = \frac{-\gamma e \frac{\partial \gamma p}{\partial \tau_d}}{\alpha + \left(\frac{\partial \gamma p}{\partial \tau_d}\right)^2} \quad (3.20)$$

Sehingga, parameter-parameter kontroler yang digunakan untuk kontroler PID MRAC dapat dituliskan kembali menjadi seperti berikut :

$$Kp = \int_0^t \frac{-\gamma e \frac{((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3) \cdot (r - yp)}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2}}{\alpha + \left(\frac{((\tau_i a_1 + \tau_i \tau_d a_2) s^2 + (\tau_i a_2 + a_1) s + a_2 + a_1 \tau_i \tau_d s^3) \cdot (r - yp)}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2} \right)^2} dt$$

$$\tau_i = \int_0^t \frac{-\gamma e \frac{(a_1 Kp \tau_d s^3 + (a_1 Kp + Kp \tau_d) s^2 + Kp a_2 s) \cdot (r - yp) - (s^3 + b_1 s^2 + b_2 s) \cdot yp}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2}}{\alpha + \left(\frac{(a_1 Kp \tau_d s^3 + (a_1 Kp + Kp \tau_d) s^2 + Kp a_2 s) \cdot (r - yp) - (s^3 + b_1 s^2 + b_2 s) \cdot yp}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2} \right)^2} dt$$

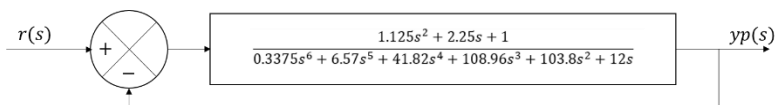
$$\tau_d = \int_0^t \frac{-\gamma e \frac{((a_1 Kp \tau_i) s^3 + (a_2 Kp \tau_i) s^2) \cdot (r - yp)}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2}}{\alpha + \left(\frac{(a_1 Kp \tau_i) s^3 + (a_2 Kp \tau_i) s^2) \cdot (r - yp)}{(a_1 Kp \tau_i \tau_d + \tau_i) s^3 + (b_1 \tau_i + Kp \tau_i a_1 + a_2 Kp \tau_i \tau_d) s^2 + (b_2 \tau_i + Kp \tau_i a_2 + a_1 Kp) s + Kp a_2} \right)^2} dt$$

3.3 Penentuan Model Reference Untuk Kontroler Adaptive PID MRAC

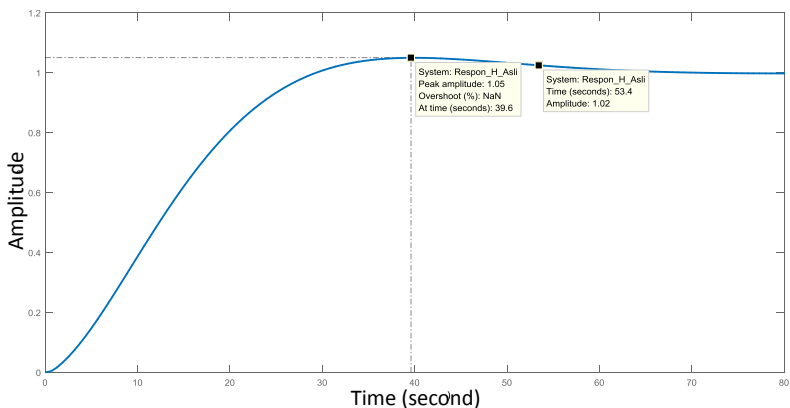
Penentuan referensi model yang digunakan pada penelitian ini yaitu dengan menggunakan bentuk *closed loop* dari sistem secara keseluruhan dimana bentuk tersebut tersusun atas kontroler PID yang sudah ditetapkan parameter kontrol yang digunakan seperti yang dijelaskan sebelumnya dan fungsi transfer dari plant pengaturan kecepatan *steam turbine* secara keseluruhan yang telah direduksi jumlah *pole* sistemnya.

Sebelum mencari bentuk *model reference* yang akan digunakan, maka terlebih dahulu dicari nilai parameter kontrol yang ingin dipakai dan dapat menghasilkan spesifikasi respon sesuai yang dikehendaki. Langkah awal yang perlu dilakukan adalah melakukan uji *closed loop steam turbine* dengan *input unit step* untuk mengetahui respon dari sistem ketika tidak diberikan kontroler.

Bentuk diagram blok *closed loop* dari *transfer function steam turbine* yang akan diuji dengan sinyal *unit step* adalah sebagai berikut :



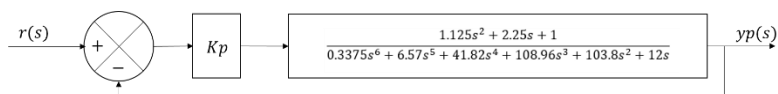
Dengan menggunakan sinyal uji *unit step*, maka didapatkan hasil respon dari *transfer function steam turbine tandem compound* menggunakan parameter sistem ideal adalah sebagai berikut



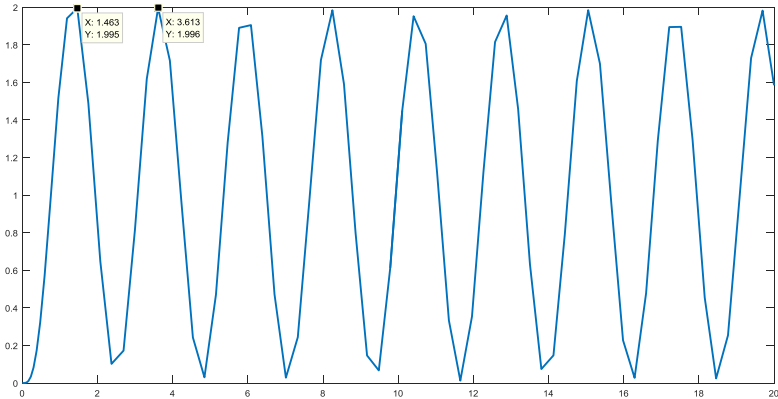
Gambar 3. 5 Grafik *plot uji step response* pada *steam turbine tandem compound*

Dari Gambar 3.5, didapatkan bahwa respon dari sistem memiliki *overshoot* sebesar 0.05 (5 %) dan respon juga menunjukkan *settling time* $t_s(\pm 2\%)$ yang dibutuhkan plant tanpa kontroler adalah sebesar 53.4 detik. *Overshoot* yang ditimbulkan oleh *plant* tanpa diberi kontroler masih dirasa terlalu tinggi. Untuk itu, perlu dicari nilai parameter kontrol PID yang dapat mengurangi *overshoot* dan memberikan bentuk sinyal respon yang sesuai.

Metode awal yang digunakan untuk mencari nilai parameter PID adalah dengan *tuning Ziegler-Nichols* dengan metode *closed loop*. Pada metode ini dilakukan penyusunan diagram blok sistem menggunakan konfigurasi *closed loop* dan sebuah kontroler P (*proportional*) yang dapat diubah-ubah dengan *input* sinyal uji *unit step*.



Kemudian, dilanjutkan dengan mengatur nilai K_p dari nol sampai suatu nilai tertentu yang menghasilkan grafik respon yang berosilasi permanen. Nilai *gain* yang menghasilkan respon seperti ini biasa disebut dengan *gain critical* (K_{cr}). Didapati ketika nilai *gain* K_p sama dengan 193.8164, hasil respon menunjukkan osilasi yang permanen dengan periode osilasi P_{cr} sama dengan 2.15 detik. Untuk lebih jelasnya dapat dilihat hasil respon pada Gambar 3.6

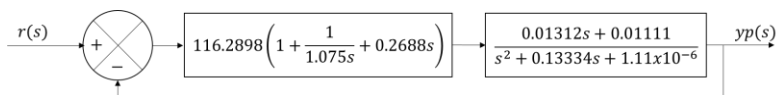


Gambar 3. 6 Grafik plot uji Ziegler-Nichols

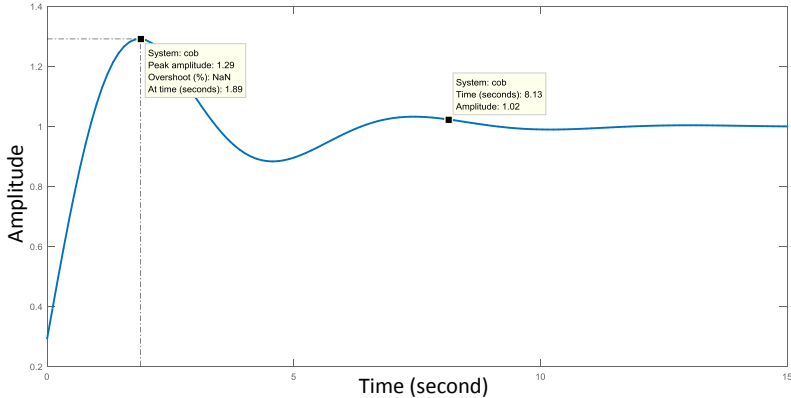
Sehingga, menggunakan tabel *tuning Ziegler-Nichols* dengan metode *closed loop*, dapat diketahui parameter kontroler PID sebagai berikut

- $K_p = 0.6 \times K_{cr} = 0.6 \times 193.8164 = 116.2898$
- $\tau_i = 0.5 \times P_{cr} = 0.5 \times 2.15 = 1.075 \text{ sekon}$
- $\tau_d = 0.125 \times P_{cr} = 0.125 \times 2.15 = 0.2688 \text{ sekon}$

Dengan menggunakan parameter kontrol tersebut, kemudian dilakukan pengujian respon *plant* hasil penyederhanaan orde 2 menggunakan sinyal uji *unit step* dengan diagram blok sebagai berikut



Dari pengujian menggunakan aplikasi MATLAB, didapatkan hasil respon sistem terhadap sinyal uji *unit step* sesuai dengan Gambar 3.7.

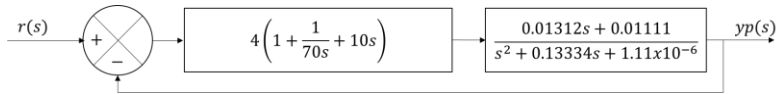


Gambar 3. 7 Grafik *plot* uji *step response* pada *plant steam turbine* orde-2 dengan kontroler *PID* hasil *tunning Zigler-Nichols*

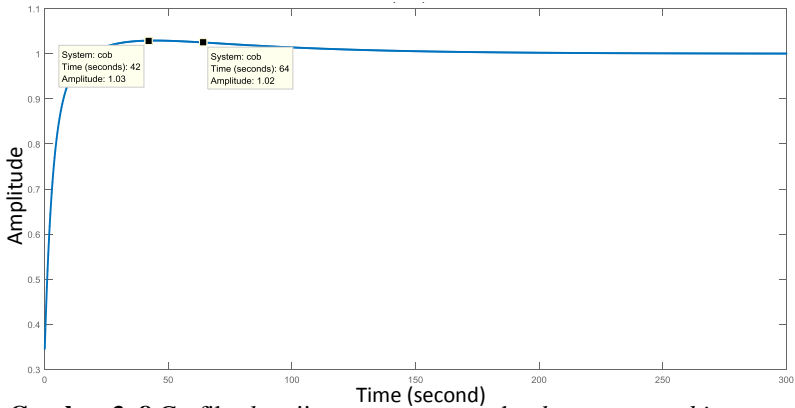
Dari gambar diatas, didapatkan bahwa respon dari sistem dengan kontroler *PID* memiliki *overshoot* sebesar 0.29 (29 %) dan respon juga menunjukkan *settling time* $t_s(\pm 2\%)$ sebesar 8.13 sekon. Nilai *settling time* ini terbilang sudah cukup cepat dibandingkan ketika tidak menggunakan kontroler *PID*, akan tetapi *overshoot* yang ditimbulkan lebih besar jika dibandingkan dengan respon sistem *closed loop* tanpa kontroler *PID*. Karena *overshoot* yang ditimbulkan cukup besar, maka dilakukan *tunning* ulang dengan metode *trial error* untuk menemukan parameter *PID* yang sesuai. Dengan metode *trial error*, didapatkan nilai parameter kontroler *PID* sebagai berikut

- $K_p = 4$
- $\tau_i = 70$ sekon
- $\tau_d = 10$ sekon

Dengan menggunakan parameter kontrol tersebut, kemudian dilakukan pengujian respon *plant* hasil penyederhanaan orde 2 menggunakan sinyal uji *unit step* dengan diagram blok sebagai berikut



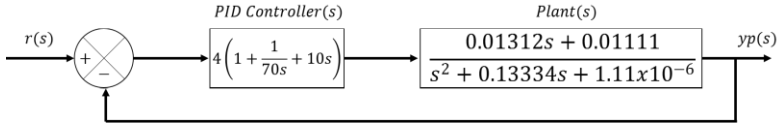
Dari pengujian menggunakan aplikasi MATLAB, didapatkan hasil respon sistem terhadap sinyal uji *unit step* sesuai dengan Gambar 3.8.



Gambar 3. 8 Grafik plot uji *step response* pada *plant steam turbine* orde-2 dengan kontroler *PID* hasil *tunning trial error*

Dari gambar diatas, didapatkan bahwa respon dari sistem dengan kontroler *PID* memiliki *overshoot* sebesar 0.03 (3 %) dan respon juga menunjukkan *settling time* $t_s(\pm 2\%)$ sebesar 64 sekon. Dengan nilai *overshoot* yang sudah lebih kecil jika dibandingkan dengan respon *closed loop plant* tanpa kontroler maupun ketika dengan kontroler *PID* parameter hasil *tunning Ziegler-Nichols* dan *settling time* yang dinilai cukup, maka parameter yang diperoleh menggunakan metode *trial error* akan digunakan sebagai parameter kontroler penyusun *model reference*.

Apabila digambarkan pada diagram blok, maka bentuk *closed loop* dari kontroler *PID* beserta *plant steam turbine* yang sudah disederhanakan kedalam bentuk orde dua memiliki bentuk seperti Gambar 3.9.



Gambar 3. 9 Diagram blok sistem yang digunakan sebagai *model reference*

Dari diagram blok yang tersusun diatas, kemudian dilakukan pembentukan *transfer function closed loop* secara keseluruhan. *Transfer function* yang terbentuk ini kemudian dijadikan sebagai *model reference* pada skema kontrol *adaptive PID MRAC* sehingga respon yang dihasilkan oleh *plant steam turbine tandem compound* dapat sesuai dengan respon yang dihasilkan oleh *model reference* yang sejatinya merupakan respon dari *plant* yang sudah diberikan kontroler PID konvensional sehingga menghasilkan bentuk yang sesuai. Fungsi transfer yang terbentuk dari diagram blok *closed loop* pada Gambar 3.9 dapat dituliskan sesuai persamaan (3.21).

$$H_{Model}(s) = \frac{36.74s^3 + 34.78s^2 + 3.163s + 0.04444}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \quad (3.21)$$

3.4 Penyusunan Diagram Blok Sistem Secara Keseluruhan

Secara umum, mekanisme adaptif dengan skema *MRAC* akan memiliki dua buah loop yaitu inner loop yang terdiri dari *plant* dan kontroler sedangkan yang kedua adalah outer loop yang terdiri atas *adjustment mechanism* untuk mengatur parameter kontroler agar menghasilkan respon sistem yang sesuai dengan referensi model. Pada skema kontrol *adaptive PID MRAC*, akan dibuat dua bentuk diagram blok berdasarkan penyusunan *adjustment mechanism* yang dipakai.

Kedua jenis *adjustment mechanism* yang dipakai adalah *General MIT Rule* dan *Normalized MIT Rule* dimana yang membedakan konfigurasi antar dua jenis metode penyusunan ini adalah blok *adjustment mechanism* yang terbentuk, sedangkan untuk *transfer function* dari *plant steam turbine tandem compound* dan *model reference* tetap sama. Dengan menggunakan parameter sistem yang sudah disederhanakan dan juga parameter kontrol yang sudah ditetapkan, maka diagram blok keseluruhan dari skema *adaptive PID MRAC* dapat

dibentuk. Secara ringkas parameter-parameter tersebut dapat dituliskan sebagai berikut

- Parameter *plant* :
 - $a_1 = 0.01312$
 - $a_2 = 0.01111$
 - $b_1 = 0.13334$
 - $b_2 = 1.11 \times 10^{-6}$

- Parameter kontroler :
 - $Kp = 4$
 - $\tau_i = 70$
 - $\tau_d = 10$

Berikut merupakan konfigurasi dari masing-masing diagram blok yang disusun berdasarkan cara penentuan *adjustment mechanism* dengan parameter sesuai dengan yang sudah ditentukan

3.4.1 Diagram blok *Adaptive PID MRAC* dengan *basic MIT Rule*

Adjustment mechanism yang sudah diturunkan dan disusun menggunakan parameter yang ditentukan dapat dituliskan kembali sebagai berikut

$$\frac{dKp}{dt} = -\gamma \cdot e \cdot \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)$$

$$\frac{d\tau_i}{dt} = -\gamma \cdot e \cdot \left(\frac{(0.5248s^3 + 40.05s^2 + 0.0444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11 \times 10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right)$$

$$\frac{d\tau_d}{dt} = -\gamma \cdot e \cdot \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)$$

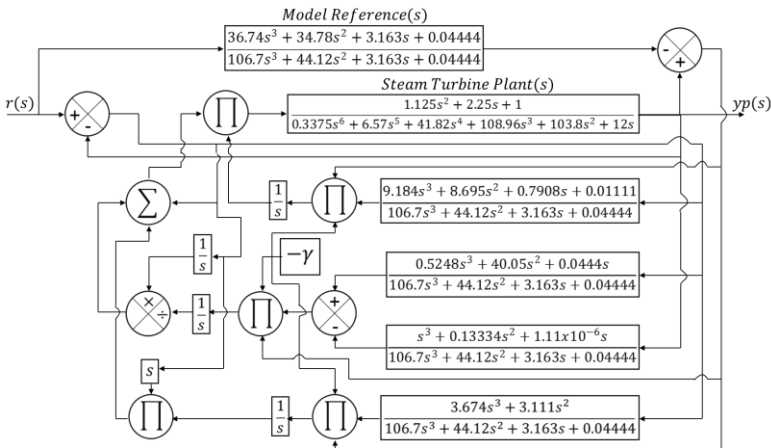
Dari bentuk tersebut, maka parameter-parameter kontroler hasil pada skema *adaptive PID MRAC* dengan *basic MIT Rule* dapat dituliskan kembali sebagai berikut

$$K_p = \int_0^t -\gamma \cdot e \cdot \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right) dt$$

$$\tau_i = \int_0^t -\gamma \cdot e \cdot \left(\frac{(0.5248s^3 + 40.05s^2 + 0.04444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11 \times 10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right) dt$$

$$\tau_d = \int_0^t -\gamma \cdot e \cdot \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right) dt$$

Sehingga diagram blok untuk skema *adaptive PID MRAC* dengan *basic MIT Rule* dapat disusun kembali sebagai berikut



3.4.2 Diagram blok Adaptive PID MRAC dengan normalized MIT Rule

Pada praktiknya, penyusunan *adaptive PID MRAC* dengan *basic MIT Rule* dan *normalized MIT Rule* memiliki bentuk dan langkah yang hampir sama, yang membedakan hanya pada bentuk *adjustment mechanism* yang digunakan. *Adjustment mechanism* yang sudah

diturunkan dan disusun menggunakan parameter yang ditentukan dapat dituliskan kembali sebagai berikut

$$\frac{dKp}{dt} = \frac{-\gamma \cdot e \cdot \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)}{\alpha + \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)^2}$$

$$\begin{aligned} \frac{d\tau_i}{dt} &= \frac{-\gamma \cdot e \cdot \left(\frac{(0.5248s^3 + 40.05s^2 + 0.0444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11x10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right)}{\alpha + \left(\frac{(0.5248s^3 + 40.05s^2 + 0.0444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11x10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right)^2} \end{aligned}$$

$$\frac{d\tau_d}{dt} = \frac{-\gamma \cdot e \cdot \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)}{\alpha + \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)^2}$$

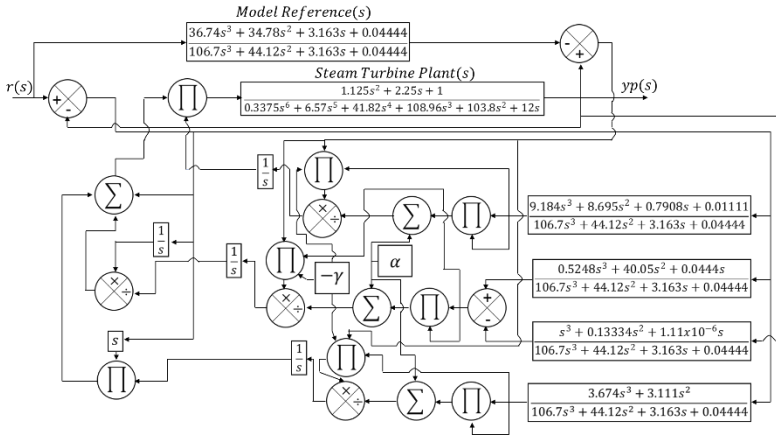
Dari bentuk tersebut, maka parameter-parameter kontroler hasil pada skema *adaptive PID MRAC* dengan *normalized MIT Rule* dapat dituliskan kembali sebagai berikut

$$Kp = \int_0^t \frac{-\gamma \cdot e \cdot \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)}{\alpha + \left(\frac{9.184s^3 + 8.695s^2 + 0.7908s + 0.01111}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)^2} dt$$

$$\tau_i = \int_0^t \frac{-\gamma \cdot e \cdot \left(\frac{(0.5248s^3 + 40.05s^2 + 0.0444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11x10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right)}{\alpha + \left(\frac{(0.5248s^3 + 40.05s^2 + 0.0444s) \cdot (r - yp) - (s^3 + 0.13334s^2 + 1.11x10^{-6}s) \cdot yp}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \right)^2} dt$$

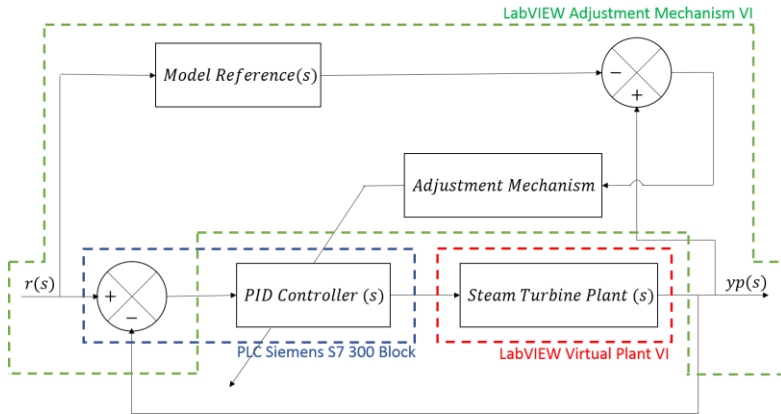
$$\tau_d = \int_0^t \frac{-\gamma \cdot e \cdot \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)}{\alpha + \left(\frac{3.674s^3 + 3.111s^2}{106.7s^3 + 44.12s^2 + 3.163s + 0.04444} \cdot (r - yp) \right)^2} dt$$

Sehingga diagram blok untuk skema *adaptive PID MRAC* dengan *normalized MIT Rule* dapat disusun kembali sebagai berikut



3.5 Perancangan Skema Kontrol *adaptive PID MRAC* Secara Keseluruhan dan Integrasi antara *LabVIEW* dengan *PLC Siemens S7 300*

Untuk merancang skema kontrol *adaptive PID MRAC* dengan menggunakan *LabVIEW* dan *PLC Siemens S7 300*, diperlukan peninjauan kembali diagram blok dari skema *adaptive PID MRAC* secara umum. Gambar 3.10 menyajikan diagram blok dari skema kontrol tersebut beserta dengan pembagian masing-masing komponen pada *PLC* dan *LabVIEW*.



Gambar 3. 10 Diagram blok skema *adaptive PID MRAC* secara umum

Dari gambar tersebut, dapat terlihat bahwa terdapat bagian dari diagram blok yang disusun secara terpisah. Bagian-bagian tersebut adalah *PID Controller* yang akan disusun dalam *PLC Siemens S7 300* menggunakan aplikasi *Step 7*, dan dua bagian yang lain adalah *Steam Turbine Virtual Plant* dan *Adjustment Mechanism* akan disusun kedalam dua halaman *Virtual Instrument (VI)* yang berbeda menggunakan *LabVIEW*.

Untuk menghubungkan antara *Virtual Instrument Steam Turbine Virtual Plant* dengan *Virtual Instrument Adjustment Mechanism*, digunakan fasilitas *shared variable* pada *LabVIEW* sehingga kedua *variable* pada *VI* yang berbeda dapat terhubung satu sama lain. Kemudian untuk menghubungkan dan transfer data antara *LabVIEW* dengan aplikasi *Step 7* pada *PLC Siemens S7 300*, digunakan dua media, yaitu *OPC Servers* yang menggunakan aplikasi *IBH OPC Softech* dan juga *Arduino UNO Board*. *OPC Server* digunakan untuk proses transfer data *online* secara terus menerus antara *adjustment mechanism VI* dengan *Step 7 Siemens S7 300*, data yang ditransfer berupa parameter kontrol K_p , τ_i , dan τ_d yang sudah di-adjust pada *adjustment mechanism VI*. Sedangkan untuk nilai *Set Point*, Sinyal Kontrol, dan Sinyal Output Plant dikirimkan dari *PLC* ke *Labview* atau sebaliknya menggunakan *Arduino UNO Board*.

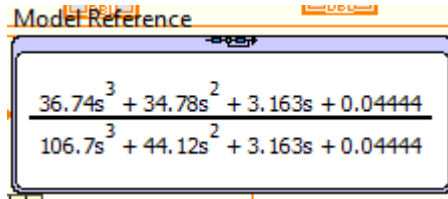
Dari Gambar 3.10, terlihat bahwa terdapat dua halaman *VI* yang berbeda guna menyusun *adjustment mechanism* dan juga *virtual plant*. Skema tersebut digunakan ketika simulasi dan implementasi

menggunakan PLC, akan tetapi apabila hanya simulasi tanpa menggunakan PLC, maka cukup menggunakan satu LabVIEW VI.

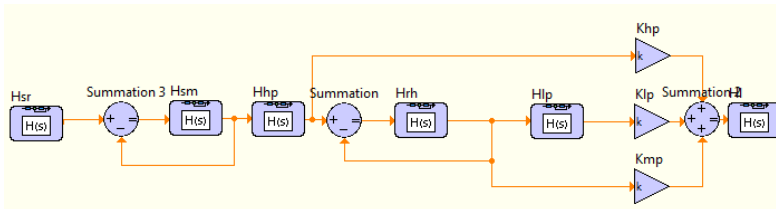
3.5.1 Perancangan Blok Simulasi pada LabVIEW

- *Adaptive PID MRAC LabVIEW VI*

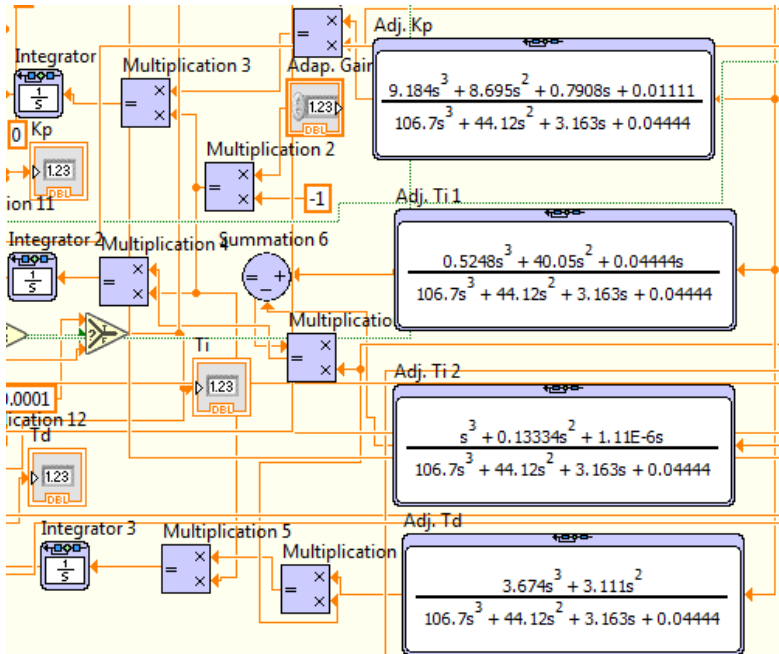
Pada penyusunan diagram blok dari skema kontrol *adaptive PID MRAC* untuk keperluan simulasi tanpa *PLC*, ketiga komponen penting pada skema tersebut yang meliputi kontroler *PID*, *plant steam turbine tandem compound*, dan *adjustment mechanism* akan disatukan dalam satu diagram blok.



Gambar 3. 11 Diagram blok *model reference* pada *adaptive PID MRAC LabVIEW VI*



Gambar 3. 12 Diagram blok *steam turbine tandem compound* pada *adaptive PID MRAC LabVIEW VI*



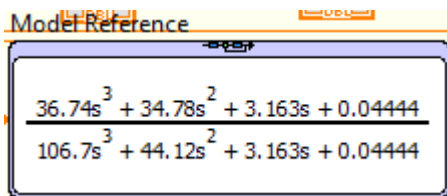
Gambar 3. 13 Diagram blok *adjustment mechanism* pada *adaptive PID MRAC LabVIEW VI*

Didalam penyusunan diagram blok *adaptive PID MRAC*, terdapat tiga sinyal yang digunakan sebagai *input* yang terdiri dari *step*, *sinus*, dan *pulse* dimana pemilihan sinyal dapat dilakukan melalui HMI yang juga dibuat bersamaan dengan pembuatan diagram blok *adaptive PID MRAC*. Selain itu, parameter dari *plant* yang meliputi *time constant* dan *gain overall* dapat juga diubah-ubah melalui HMI yang sudah dibuat. Tujuan dapat diubah-ubahnya parameter ini adalah untuk menguji pengaruh perubahan parameter dari *plant* terhadap respon yang dihasilkan ketika menggunakan konsep *adaptive PID MRAC*

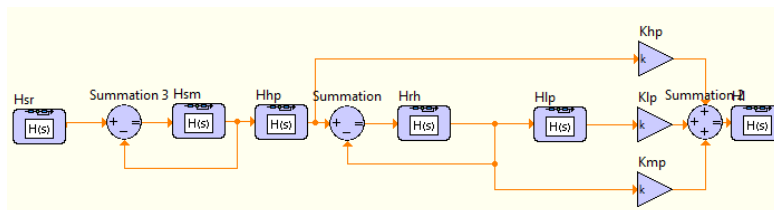
- *Normalized Adaptive PID MRAC LabVIEW VI*

Sama seperti dengan diagram blok *adaptive PID MRAC* biasa, diagram blok versi ternormalisasi ini juga tersusun dari seluruh bagian-bagian yang ada pada diagram blok dengan konsep *adaptive PID*

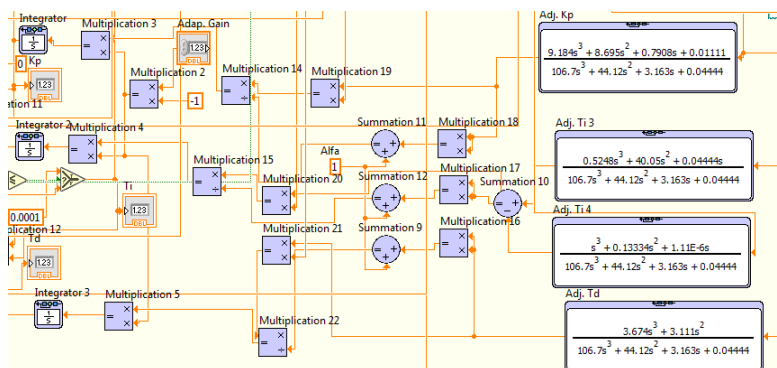
MRAC. Hanya saja yang membedakan pada diagram block adaptive PID MRAC versi ternormalisasi ini adalah penyusunan adjustment mechanism yang mengikuti aturan MIT ternormalisasi.



Gambar 3. 14 Diagram blok *model reference* pada *normalized adaptive PID MRAC LabVIEW VI*



Gambar 3. 15 Diagram blok *steam turbine tandem compound* pada *normalized adaptive PID MRAC LabVIEW VI*



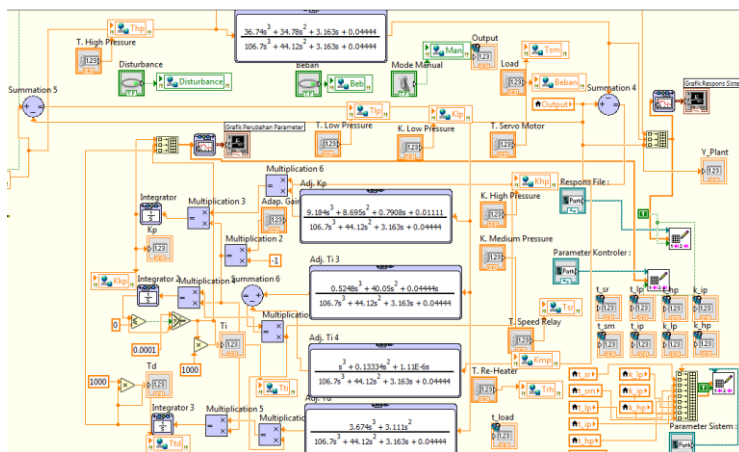
Gambar 3. 16 Diagram blok *adjustment mechanism* pada *normalized adaptive PID MRAC LabVIEW VI*

Pada diagram blok tersebut juga digunakan untuk simulasi tanpa PLC sehingga kontroler dan plant sudah disusun menjadi satu dengan *adjustment mechanism*. Selain itu juga, sama seperti diagram blok *adaptive PID MRAC* versi biasa, pada versi ternormalisasi ini juga terdapat tiga input yang dapat dipilih yang masing-masing terdiri atas *step*, *sinus*, dan *pulse*. Parameter dari *plant* juga dapat diubah-ubah dengan tujuan yang sama seperti yang ada pada skema *adaptive PID MRAC* yang biasa yaitu untuk menguji pengaruh perubahan parameter dari *plant* terhadap respons yang dihasilkan ketika menggunakan konsep *normalized adaptive PID MRAC*.

3.5.2 Perancangan Blok Simulasi LabVIEW dan Implementasi pada PLC Siemens S7 300

- Adaptive PID MRAC Adjustment Mechanism VI

Pada penyusunan diagram blok dari skema kontrol *adaptive PID MRAC* untuk keperluan simulasi dan implementasi pada *PLC Siemens S7 300* perlu dibentuk suatu VI khusus dimana komponen yang ada didalamnya hanya digunakan sebagai penyusun *adjustment mechanism* dari skema *adaptive PID MRAC*, sedangkan untuk diagram blok kontroler akan digunakan *PLC Siemens S7 300* dan untuk *plant* akan disusun kembali pada VI yang berbeda.

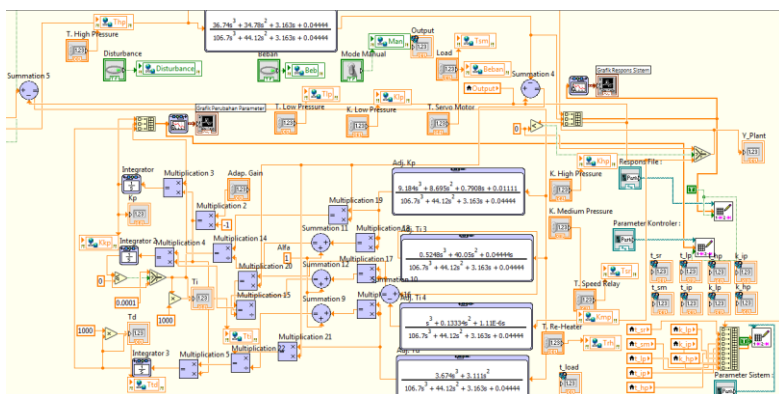


Gambar 3. 17 Diagram blok *adjustment mechanism*, *model reference*, dan variabel yang di-share pada *adaptive PID MRAC adjustment mechanism VI*

Pada diagram blok ini, terdapat tiga variasi sinyal input yang dapat digunakan yaitu *step*, *sinus*, dan *pulse*. Selain itu, dikarenakan VI ini harus terhubung dengan VI untuk *plant* dan kontroler pada *PLC Siemens S7 300*, maka diperlukan penggunaan fitur *data binding* dan *shared variable*. *Data binding* akan melakukan pengikatan (*binding*) data-data yang akan dikomunikasikan dengan PLC melalui media OPC yang diwakili oleh *IBH OPC Server*. Sedangkan fitur *shared variable* akan digunakan untuk membagikan variable antar VI yang meliputi *adjustment mechanism VI* dan *steam turbine virtual plant VI*.

- *Normalized Adaptive PID MRAC Adjustment Mechanism VI*

Sama seperti penyusunan skema kontrol *adaptive PID MRAC* biasa, pada *normalized PID MRAC* ini struktur LabVIEW VI *adjustment mechanism* hanya terdiri dari *model reference*, *adjustment mechanism*, dan beberapa operator saja, sedangkan untuk komponen kontroler terdapat pada *PLC Siemens S7 300* dan untuk *virtual plant steam turbine* akan disusun kembali pada suatu VI tersendiri pada LabVIEW yang nantinya akan dihubungkan dengan VI *adjustment mechanism* menggunakan *shared variable* LabVIEW, dan parameter kontrol yang sudah di-*adjust* pada *adjustment mechanism VI* akan dikirimkan pada PLC melalui *IBH OPC Server*.



Gambar 3. 18 Diagram blok *adjustment mechanism*, *model reference*, dan variabel yang di-*share* pada *normalized adaptive PID MRAC adjustment mechanism VI*

- *Steam Turbine Tandem Compound Virtual Plant VI*

Selain penyusunan dua *adjustment mechanism VI*, dibuat juga suatu VI tersendiri sebagai pengganti *plant* asli secara *virtual* yang nantinya akan digunakan sebagai objek yang akan dikontrol dimana dalam hal ini adalah kecepatan putar dari *steam turbine*. *Plant* ini dibuat dari *transfer function* keseluruhan *steam turbine tandem compound* dan telah mengalami diskritisasi menggunakan metode Tustin dengan *sampling time* sebesar 100 ms.

Input dari *plant* ini merupakan sinyal kontrol dari kontroler *PID MRAC* dan *output*-nya berupa respon dari *plant* tersebut yang kemudian dibandingkan dengan *model reference* yang ada untuk melihat kesesuaian respon yang dihasilkan. Berikut merupakan *transfer function* dari *plant steam turbine tandem compound* dengan parameter ideal secara keseluruhan

$$H_{T_{otideal}}(s) = \frac{1.125s^2 + 2.25s + 1}{0.3375s^6 + 6.57s^5 + 41.82s^4 + 108.96s^3 + 103.8s^2 + 12s}$$

Untuk mendapatkan bentuk diskrit dari *plant* yang ada, digunakan metode transformasi bilinear (*Tustin method*). *Tustin method* ini merupakan salah satu bentuk yang dapat digunakan untuk mendapatkan bentuk sinyal waktu diskrit dari sinyal waktu kontinyu yang dimiliki. Pada transformasi bilinear dengan menggunakan metode Tustin, diperlukan bentuk *transfer function plant* dalam domain S yang ingin diubah menjadi persamaan diskrit. Pada metode Tustin ini, variable z didekati dengan persamaan (3.22).

$$z = e^{sT_s} \approx \frac{1 + sT_s/2}{1 - sT_s/2} \quad (3.22)$$

Sehingga, apabila ingin dilakukan perubahan dari domain frekuensi s kedalam domain frekuensi z, dilakukan transformasi dengan mengganti variable s menjadi

$$s' = \frac{2}{T_s} \cdot \frac{z - 1}{z + 1} \quad (3.23)$$

Setelah dilakukan perubahan domain *plant* kedalam domain frekuensi z , kemudian dilanjutkan dengan melakukan transformasi *invers* z sehingga didapatkan persamaan sistem didalam waktu diskrit.

Pada *plant steam turbine*, terdapat 3 *transfer function* yang menyatakan *section* pada turbin. Selain itu, juga terdapat *transfer function* yang menyatakan *servo motor* ($H_{ServoMotor}$) dan *speed relay* ($H_{SpeedRelay}$) pada governor yang digunakan untuk mengatur *steam flow* yang masuk menuju turbin. Kemudian juga terdapat *transfer function load* (H_{Load}) yang menyatakan hubungan antara *steam flow* yang keluar dari masing-masing *section* pada turbin dengan torsi yang dihasilkan kemudian akan menghasilkan kecepatan putaran turbin.

Dengan menggunakan *sampling time* (T_s) sebesar 100 ms, maka akan didapatkan persamaan bentuk diskrit hasil tranformasi dengan metode Tustin dari masing-masing *transfer function* sebagai berikut :

$H_{SpeedRelay}$:

$$H_{SR}(s) = \frac{1}{T_{SR}s + 1}$$

$$H_{SR}(z) = \frac{1}{T_{SR} \left(\frac{2}{T_s} \cdot \frac{z-1}{z+1} \right) + 1} = \frac{1}{20T_{SR} \left(\frac{z-1}{z+1} \right) + 1} = \frac{z+1}{(20T_{SR}+1)z - 20T_{SR} + 1} \cdot \frac{z^{-1}}{z^{-1}}$$

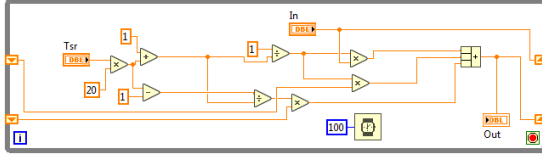
$$= \frac{1+z^{-1}}{(20T_{SR}+1) + (-20T_{SR}+1)z^{-1}}$$

$$H_{SR}(z) = \frac{Y_{SR}(z)}{X_{SR}(z)} = \frac{1+z^{-1}}{(20T_{SR}+1) + (-20T_{SR}+1)z^{-1}} = \frac{(1+z^{-1}) / (20T_{SR}+1)}{1 + \frac{(-20T_{SR}+1)}{(20T_{SR}+1)}z^{-1}} \quad (3.24)$$

Transformasi Z Invers :

$$y_{SR}[n] = \frac{(20T_{SR}-1)}{(20T_{SR}+1)}y_{SR}[n-1] + \frac{1}{(20T_{SR}+1)}x_{SR}[n] + \frac{1}{(20T_{SR}+1)}x_{SR}[n-1] \quad (3.25)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer $H_{SpeedRelay}$ dapat ditunjukkan seperti Gambar 3.19.



Gambar 3. 19 Diagram blok persamaan beda $H_{SpeedRelay}$

$H_{ServoMotor}$:

$$H_{SM}(s) = \frac{1}{T_{SM}s}$$

$$H_{SM}(z) = \frac{1}{T_{SM} \left(\frac{2}{T_s} \cdot \frac{z-1}{z+1} \right)} = \frac{1}{20T_{SM} \left(\frac{z-1}{z+1} \right)} = \frac{z+1}{20T_{SM}z - 20T_{SM}} \cdot \frac{z^{-1}}{z^{-1}}$$

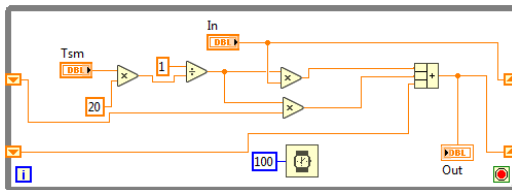
$$= \frac{1+z^{-1}}{20T_{SM} - 20T_{SM}z^{-1}}$$

$$H_{SM}(z) = \frac{Y_{SM}(z)}{X_{SM}(z)} = \frac{1+z^{-1}}{20T_{SM} - 20T_{SM}z^{-1}} = \frac{(1+z^{-1})/20T_{SM}}{1-z^{-1}} \quad (3.26)$$

Transformasi Z Invers :

$$y_{SM}[n] = y_{SM}[n-1] + \frac{1}{20T_{SM}}x_{SM}[n] + \frac{1}{20T_{SM}}x_{SM}[n-1] \quad (3.27)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer $H_{ServoMotor}$ dapat ditunjukkan seperti Gambar 3.20.



Gambar 3. 20 Diagram blok persamaan beda $H_{ServoMotor}$

$H_{HighPressure}$:

$$H_{HP}(s) = \frac{1}{T_{HP}s + 1}$$

$$H_{HP}(z) = \frac{1}{T_{HP} \left(\frac{2}{T_s} \cdot \frac{z-1}{z+1} \right) + 1} = \frac{1}{20T_{HP} \left(\frac{z-1}{z+1} \right) + 1} = \frac{z+1}{(20T_{HP}+1)z - 20T_{HP}+1} \cdot \frac{z^{-1}}{z^{-1}}$$

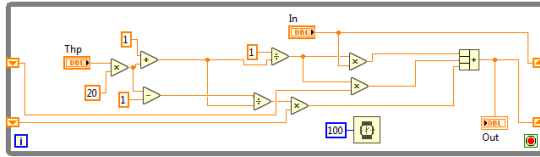
$$= \frac{1+z^{-1}}{(20T_{HP}+1) + (-20T_{HP}+1)z^{-1}}$$

$$H_{HP}(z) = \frac{Y_{HP}(z)}{X_{HP}(z)} = \frac{1+z^{-1}}{(20T_{HP}+1) + (-20T_{HP}+1)z^{-1}} = \frac{(1+z^{-1}) / (20T_{HP}+1)}{1 + \frac{(-20T_{HP}+1)}{(20T_{HP}+1)} z^{-1}} \quad (3.28)$$

Transformasi *Z Invers* :

$$y_{HP}[n] = \frac{(20T_{HP}-1)}{(20T_{HP}+1)} y_{HP}[n-1] + \frac{1}{(20T_{HP}+1)} x_{HP}[n] + \frac{1}{(20T_{HP}+1)} x_{HP}[n-1] \quad (3.29)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer $H_{HighPressure}$ dapat ditunjukkan seperti Gambar 3.21.



Gambar 3. 21 Diagram blok persamaan beda $H_{HighPressure}$

$H_{MediumPressure}$:

$$H_{MP}(s) = \frac{1}{T_{MP}s + 1}$$

$$H_{MP}(z) = \frac{1}{T_{MP} \left(\frac{2}{T_s} \cdot \frac{z-1}{z+1} \right) + 1} = \frac{1}{20T_{MP} \left(\frac{z-1}{z+1} \right) + 1} = \frac{z+1}{(20T_{MP}+1)z - 20T_{MP}+1} \cdot \frac{z^{-1}}{z^{-1}}$$

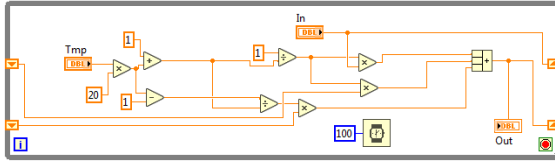
$$= \frac{1+z^{-1}}{(20T_{MP}+1) + (-20T_{MP}+1)z^{-1}}$$

$$H_{MP}(z) = \frac{Y_{MP}(z)}{X_{MP}(z)} = \frac{1+z^{-1}}{(20T_{MP}+1) + (-20T_{MP}+1)z^{-1}} = \frac{(1+z^{-1}) / (20T_{MP}+1)}{1 + \frac{(-20T_{MP}+1)}{(20T_{MP}+1)} z^{-1}} \quad (3.30)$$

Transformasi *Z Invers* :

$$y_{MP}[n] = \frac{(20T_{MP} - 1)}{(20T_{MP} + 1)}y_{MP}[n - 1] + \frac{1}{(20T_{MP} + 1)}x_{MP}[n] + \frac{1}{(20T_{MP} + 1)}x_{MP}[n - 1] \quad (3.31)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer $H_{MediumPressure}$ dapat ditunjukkan seperti Gambar 3.22.



Gambar 3. 22 Diagram blok persamaan beda $H_{MediumPressure}$

$H_{LowPressure}$:

$$H_{LP}(s) = \frac{1}{T_{LP}s + 1}$$

$$H_{LP}(z) = \frac{1}{T_{LP} \left(\frac{2}{T_s} \frac{z-1}{z+1} \right) + 1} = \frac{1}{20T_{LP} \left(\frac{z-1}{z+1} \right) + 1} = \frac{z+1}{(20T_{LP}+1)z - 20T_{LP} + 1} \cdot \frac{z^{-1}}{z^{-1}}$$

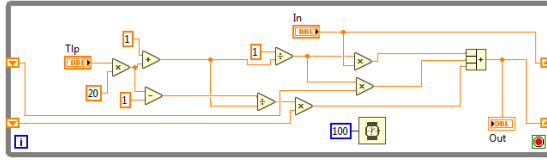
$$= \frac{1+z^{-1}}{(20T_{LP}+1) + (-20T_{LP}+1)z^{-1}}$$

$$H_{LP}(z) = \frac{Y_{LP}(z)}{X_{LP}(z)} = \frac{1+z^{-1}}{(20T_{LP}+1) + (-20T_{LP}+1)z^{-1}} = \frac{(1+z^{-1})/(20T_{LP}+1)}{1 + \frac{(-20T_{LP}+1)}{(20T_{LP}+1)}z^{-1}} \quad (3.32)$$

Transformasi *Z Invers* :

$$y_{LP}[n] = \frac{(20T_{LP} - 1)}{(20T_{LP} + 1)}y_{LP}[n - 1] + \frac{1}{(20T_{LP} + 1)}x_{LP}[n] + \frac{1}{(20T_{LP} + 1)}x_{LP}[n - 1] \quad (3.33)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer $H_{LowPressure}$ dapat ditunjukkan seperti Gambar 3.23.



Gambar 3. 23 Diagram blok persamaan beda $H_{LowPressure}$

H_{Load} :

$$H_L(s) = \frac{1}{T_L s}$$

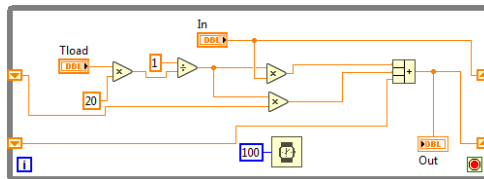
$$H_L(z) = \frac{1}{T_L \left(\frac{2}{T_s} \cdot \frac{z-1}{z+1} \right)} = \frac{1}{20T_L \left(\frac{z-1}{z+1} \right)} = \frac{z+1}{20T_L z - 20T_L} \cdot \frac{z^{-1}}{z^{-1}} = \frac{1+z^{-1}}{20T_L - 20T_L z^{-1}}$$

$$H_L(z) = \frac{Y_L(z)}{X_L(z)} = \frac{1+z^{-1}}{20T_L - 20T_L z^{-1}} = \frac{(1+z^{-1})/20T_L}{1-z^{-1}} \quad (3.34)$$

Transformasi Z Invers :

$$y_L[n] = y_L[n-1] + \frac{1}{20T_L} x_L[n] + \frac{1}{20T_L} x_L[n-1] \quad (3.35)$$

Apabila digambarkan kedalam diagram blok *LabVIEW*, maka persamaan beda dari fungsi transfer H_{Load} dapat ditunjukkan seperti Gambar 3.24.

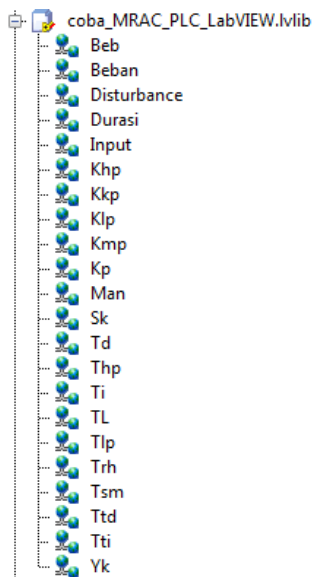


Gambar 3. 24 Diagram blok persamaan beda H_{Load}

Setelah didapatkan bentuk diskrit dari semua *transfer function* tersebut, maka dapat dilakukan penyusunan diagram blok sistem pada *software* pemrograman *LabVIEW*.

Diagram blok *steam turbine virtual plant* merupakan diagram blok yang berisikan *transfer function* diskrit *steam turbine* untuk pengaturan

kecepatan. Didalamnya tersusun atas gabungan dari persamaan beda tiap-tiap komponen penyusun *transfer function* utama. Untuk menghubungkan dengan *adjustment mechanism VI*, digunakan *shared variable* sedangkan untuk mengirimkan informasi pada *PLC Siemens S7 300*, pada VI ini menggunakan blok LINX yang berfungsi untuk membuat Arduino yang dalam hal ini bertipe UNO berfungsi sebagai *I/O interface* sehingga dapat mengirim dan menerima data dengan PLC dalam bentuk sinyal tegangan 0-5 volt. Sinyal yang dikomunikasikan menggunakan Arduino UNO ini ada tiga, yaitu *Set Point*, *Output*, dan Sinyal Kontrol. Sedangkan untuk daftar variabel yang dikomunikasikan antar VI menggunakan *shared variable* dapat dilihat dalam Gambar 3.25.

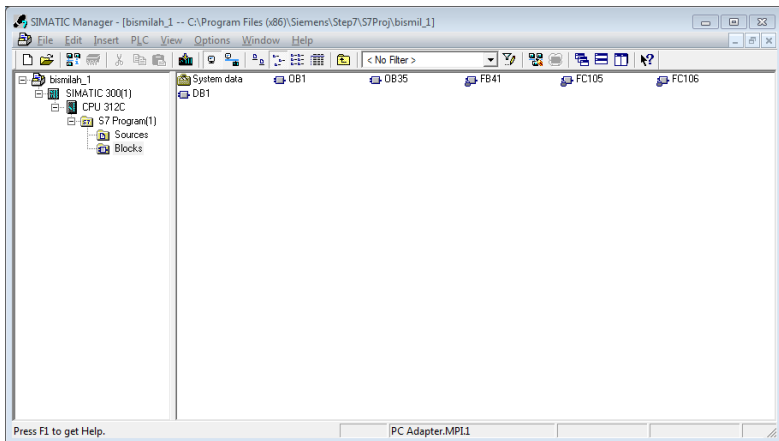


Gambar 3. 25 Variable yang di-share menggunakan fitur *shared variable*

Variabel-variabel tersebut yang terhubung antara *steam turbine virtual plant VI* dengan *adjustment mechanism VI* baik pada *adaptive PID MRAC* maupun *normalized adaptive PID MRAC*.

- Step 7 PLC Siemens S7 300

Step 7 merupakan aplikasi dari *Siemens* yang digunakan untuk merancang pemrograman *ladder diagram* pada *PLC Siemens* yang dalam hal ini digunakan *PLC Siemen S7 300 Series*. Agar bisa melakukan pemrograman *PLC* menggunakan *Step 7*, maka harus dilakukan konfigurasi terlebih dahulu terkait *PLC* dan *I/O* yang digunakan menggunakan aplikasi *SIMATIC Manager Siemens*. Untuk percobaan kali ini digunakan *PLC* bertipe *S7 312C 6ES7 312-5BD01-0AB0* yang didalamnya sudah terintegrasi dengan *Digital Input* 10 dan *Digital Output* 6, selain itu digunakan juga *I/O analog* tambahan yang berjenis *SM 334 6ES7 334-0KE80-0AB0* dimana modul ini memiliki 4 *channel analog input* dan 2 *channel analog output* yang beresolusi 12 *bits*. Setelah dilakukan konfigurasi *hardware*, maka selanjutnya akan terbentuk sebuah *project* yang akan menampilkan *block program* apa saja yang digunakan. Berikut merupakan tampilan dari *project* yang akan digunakan sebagai kontroler *PID* dalam *skema adaptive PID MRAC*.



Gambar 3. 26 Halaman *project* pada *SIMATIC Manager*

Terlihat pada Gambar 3.26, *project adaptive PID MRAC* ini menggunakan 2 *Organization Block (OB)*, 1 *Function Block (FB)*, 2 *Function (FC)*, dan 1 *Data Block*. *OB 1* merupakan main program sweep dimana didalamnya akan terdapat fungsi untuk melakukan pembacaan dan penyekalaan data analog input maupun output (*scaling*

dan *unscaling*). Selain itu, didalamnya juga disusun diagram *ladder* untuk memiliki aktivasi mode PID pada *function block PID Siemens 300*.

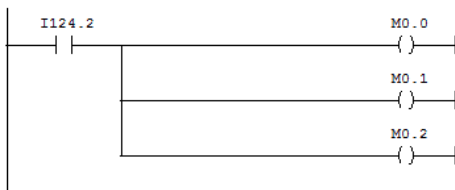
Berikut merupakan gambar susunan *program* yang ada pada OB 1 yang didalamnya terbagi menjadi 4 *network*

- *Network 1* berisikan pemilihan mode kontrol PID

OB1 : "Main Program Sweep (Cycle)"

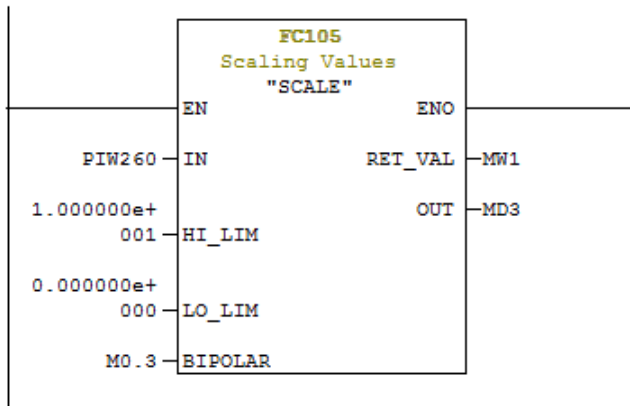
Comment:

Network 1: Aktivasi mode PID



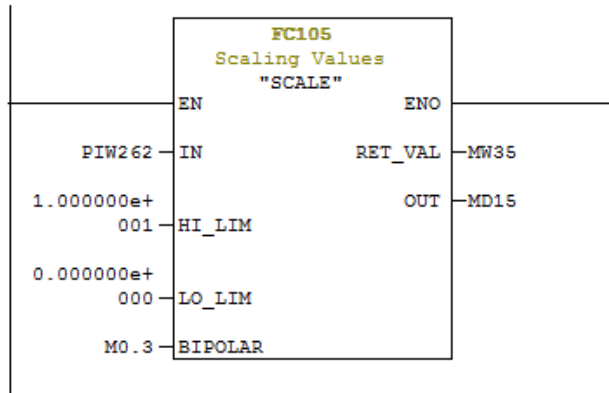
- *Network 2* berisikan fungsi *scaling* nilai PV

Network 2: Penskalaan PV



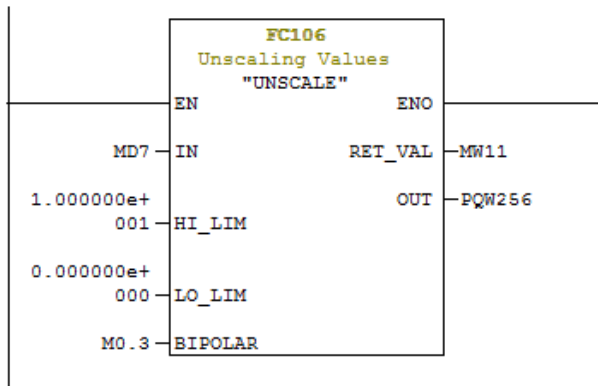
- *Network 3* berisikan fungsi *scaling* nilai SP

□ Network 3 : Penskalaan SP



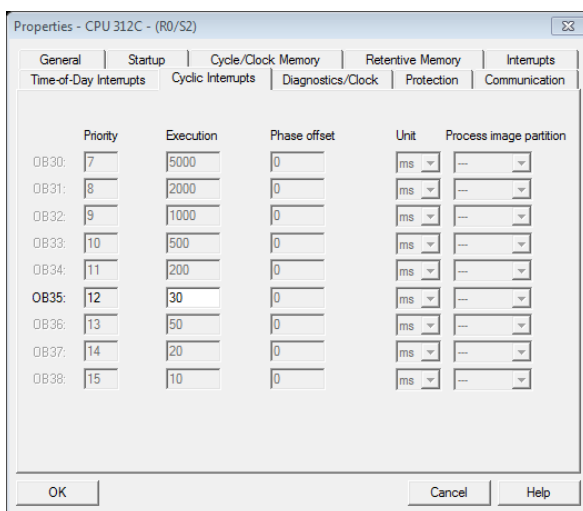
- Network 4 berisikan fungsi *scalling* nilai Sinyal Kontrol

□ Network 4 : Penskalaan Sinyal Kontrol



Selain OB 1, digunakan juga OB 35 yang merupakan blok dengan fungsi *cyclic interrupt* dimana fungsi didalamnya akan melakukan *interrupt* untuk pembacaan ulang tiap data yang ada pada durasi *cycle* yang sudah ditentukan. Dalam hal ini, *cyclic interrupt* yang digunakan sudah diatur dengan durasi 30 ms. Pemilihan durasi ini mempertimbangkan *sampling time steam turbine virtual plant* pada LabVIEW yang sudah diatur sebesar 100 ms. Sehingga sesuai aturan

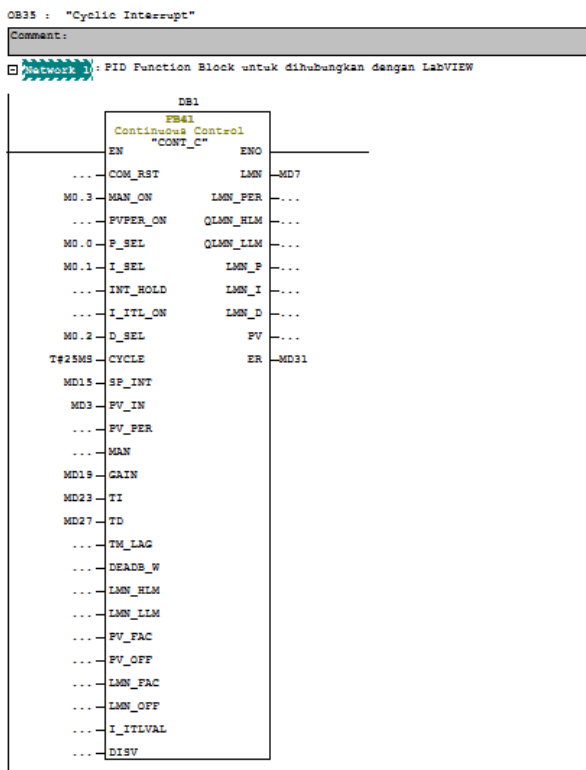
Nyquist agar tidak mengalami *aliasing*, maka frekuensi penyampling harus lebih besar atau sama dengan 2 kali frekuensi yang asli. Disini terlihat bahwa frekuensi dari *steam turbine virtual plant* adalah $f = \frac{1}{T} = \frac{1}{0.1} = 10 \text{ Hz}$, sehingga diperlukan setting frekuensi *sampling* setidaknya 20 Hz keatas. Frekuensi yang dipilih disini adalah $f = \frac{1}{T} = \frac{1}{0.030} = 33.3 \text{ Hz}$ atau 3 kali dari frekuensi asli sehingga sudah memenuhi aturan *Nyquist* tersebut. Tampilan untuk melakukan *setting* waktu pada *cyclic interrupt PLC Siemens S7 300* dapat dilihat melalui Gambar 3.27.



Gambar 3. 27 Tampilan *setting* waktu *cycle* pada *cyclic interrupt*

Didalam OB 35 ini terdapat *PID Function Block* berjenis *Continuous Control (CONT_C)*, pada *function block* ini terdapat 10 parameter utama yang akan di-*setting*. Parameter tersebut adalah *MAN_ON* untuk aktifasi mode manual, *P_SEL* untuk aktifasi mode *Proportional*, *I_SEL* untuk aktifasi mode *Integral*, *D_SEL* untuk aktifasi mode *Derivative*, *CYCLE* untuk menentukan *cycle time* dari *PID controller*, *SP_INT* untuk memberikan nilai *Set Point*, *PV_IN* untuk memberikan nilai *Process Variable*, *GAIN* untuk mengatur nilai K_p , *TI* untuk mengatur nilai τ_i , dan *TD* untuk mengatur nilai τ_d . Sedangkan untuk output dari *function block* berupa sinyal kontrol akan terhubung pada parameter *LMN*.

Berikut merupakan konfigurasi dari *function block PID controller*



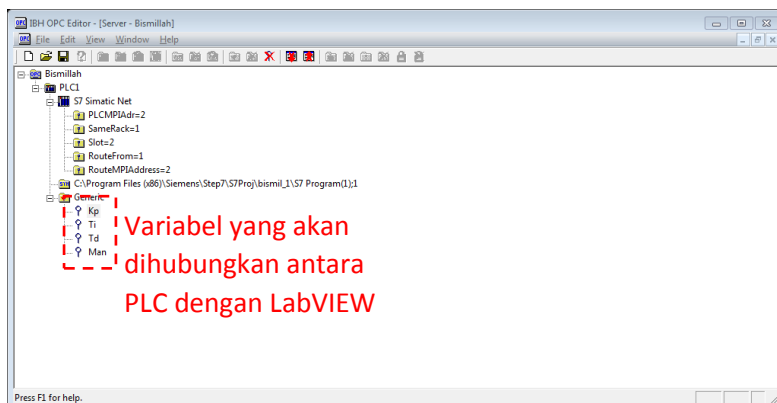
Untuk parameter GAIN, TI, TD, dan MAN_ON terhubung dengan *adjustment mechanism VI* melalui *IBH OPC Server*, sedangkan untuk parameter SP_INT, PV_IN, dan LMN terhubung dengan *steam turbine virtual plant VI* melalui *Arduino UNO*, dan parameter yang lain akan disetting langsung pada pemrograman *ladder Step 7*.

- IBH OPC Server

OPC sendiri merupakan singkatan dari *OLE (Object Linking and Embedding) for Process Control* dan merupakan suatu *protocol standard* yang digunakan untuk menghubungkan program yang ada

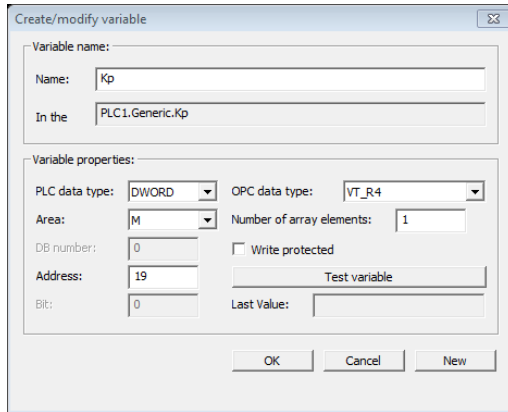
pada *computer* dengan perangkat-perangkat industri seperti PLC. *IBH OPC Server* ini merupakan salah satu jenis OPC yang dapat digunakan untuk menghubungkan *PLC Siemens Step 7* atau *Step 5* dengan berbagai program *Windows*.

Pada skema kontrol *adaptive PID MRAC*, OPC ini digunakan untuk menghubungkan parameter kontrol PID yang sudah di-*adjust* pada *adjustment mechanism VI* kedalam *PID function block* pada *Step 7*. Selain itu, OPC ini juga digunakan sebagai penghubung untuk pemilihan mode *Auto* atau *Manual* yang akan diset melalui *adjustment mechanism VI*. Berikut merupakan tampilan dari *IBH OPC Server* beserta variabel-variabel yang akan dihubungkan.



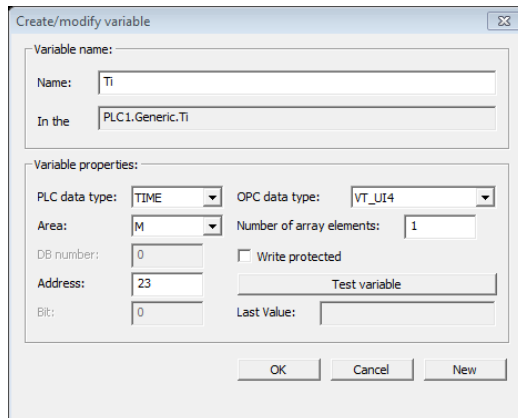
Gambar 3. 28 Tampilan *IBH OPC Editor* beserta variable yang dihubungkan

Kp: variabel ini menghubungkan konstanta proporsional pada *adjustment mechanism VI* dengan *Step 7*. Tipe data yang digunakan adalah *DWORD* dan diletakkan pada memori beralamat 19 (*MD19*).



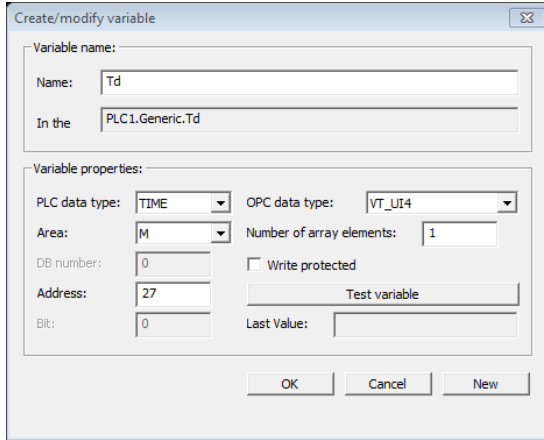
Gambar 3. 29 Konfigurasi variabel Kp pada *IBH OPC Server*

Ti: variabel ini menghubungkan konstanta integral pada *adjustment mechanism* VI dengan Step 7. Tipe data yang digunakan adalah TIME/DWORD dan diletakkan pada memori beralamat 23 (MD23).



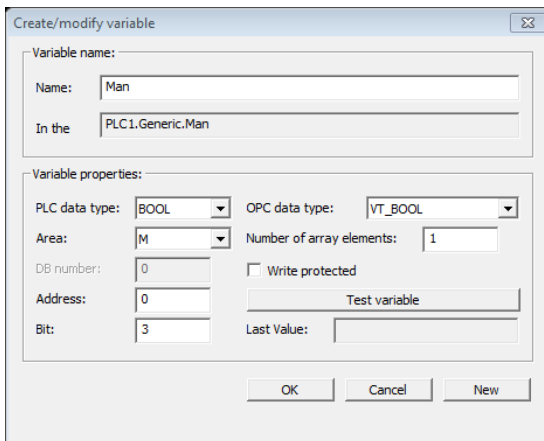
Gambar 3. 30 Konfigurasi variabel Ti pada *IBH OPC Server*

Td: variabel ini menghubungkan konstanta derivatif pada *adjustment mechanism* VI dengan Step 7. Tipe data yang digunakan adalah TIME/DWORD dan diletakkan pada memori beralamat 27 (MD27).



Gambar 3. 31 Konfigurasi variabel Td pada *IBH OPC Server*

Man: variabel ini menghubungkan pemilihan mode Auto atau Manual pada *adjustment mechanism* VI dengan Step 7. Tipe data yang digunakan adalah BOOLEAN dan diletakkan pada memori beralamat 0.3 (M0.3).



Gambar 3. 32 Konfigurasi variabel Man pada *IBH OPC Server*

BAB 4 PENGUJIAN DAN ANALISIS

Pada penelitian kali ini, sistem akan disimulasikan menggunakan *software* pemrograman *LabVIEW* dan juga akan diimplementasikan kedalam *steam turbine tandem compound virtual plant* pada *LabVIEW* dengan menggunakan *PLC Siemens S7 300*. Untuk pengujiannya akan dilakukan dengan beberapa skenario dan kondisi.

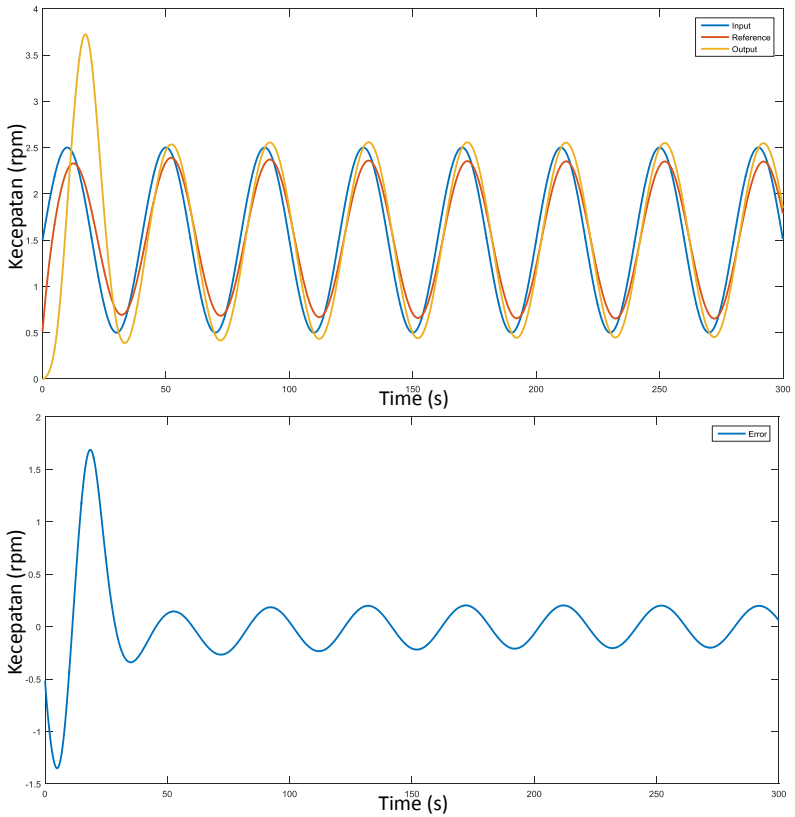
4.1 Pengujian Menggunakan *Gain* dan *Input* yang Berbeda-beda dengan Parameter Sistem yang Konstan

Untuk sesi kali ini, dilakukan pengujian skema kontrol *adaptive PID MRAC basic MIT Rule* dan *normalized MIT Rule* menggunakan *software LabVIEW*. Pengujian ini dilakukan menggunakan tiga jenis sinyal uji yaitu sinyal sinus, *pulse*, dan *step* dengan tujuan agar dapat mengetahui perubahan respon sistem dan parameter kontroler terhadap input yang berubah-ubah tiap waktunya. Dengan menggunakan nilai parameter sistem konstan dimana yang dipilih adalah pada saat kondisi ideal sesuai dengan Tabel 4.1 dan nilai adaptation gain pada tiap skema diubah-ubah antara 1, 3, dan 5, hasil yang diperoleh untuk *plant* yang menggunakan *adaptation gain* sebesar 1 dengan *input* sinyal sinus dapat ditunjukkan sebagai pada Gambar 4.1 dan Gambar 4.2.

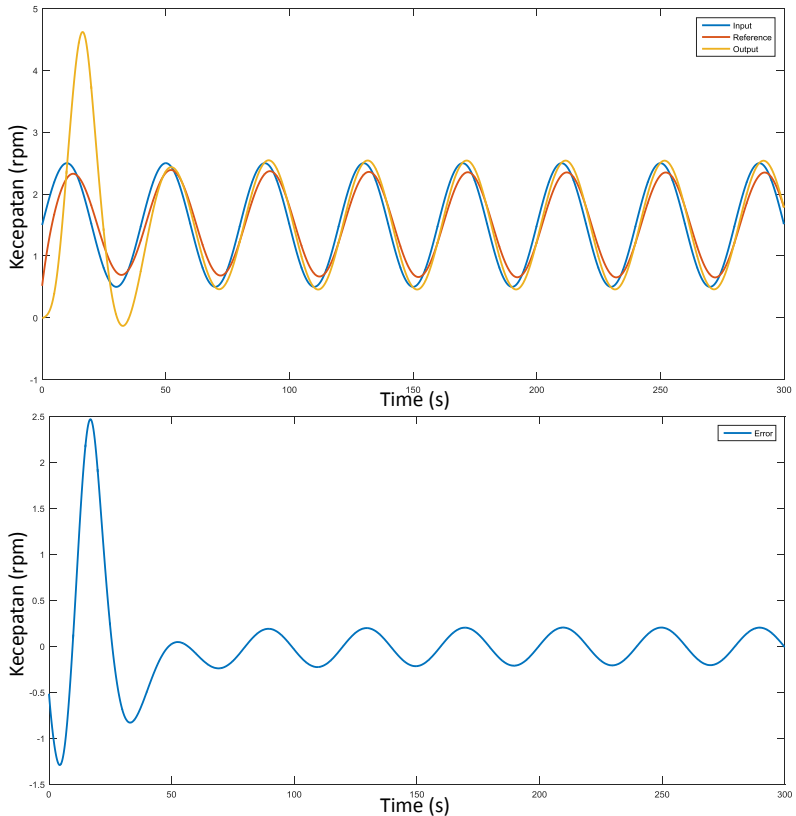
Tabel 4. 1 Parameter yang digunakan pada pengujian tanpa beban

Parameter Sistem	K_{HP}	K_{MP}	K_{LP}	T_{HP}	T_{MP}	T_{LP}	T_{SM}	T_{SR}	T_L
Ideal	0.3	0.3	0.4	0.25	7.5	0.5	0.3	0.1	12

- Input Sinyal Sinus dengan *Adaptation Gain* = 1



Gambar 4. 1 Respon *plant* dan *error* antara model *reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 1

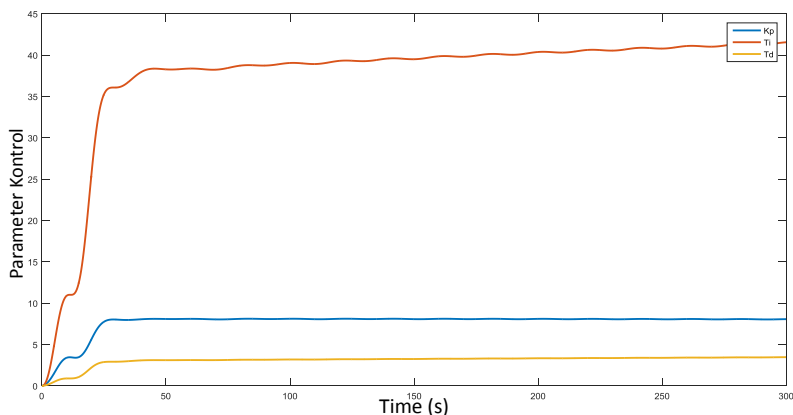


Gambar 4. 2 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 1

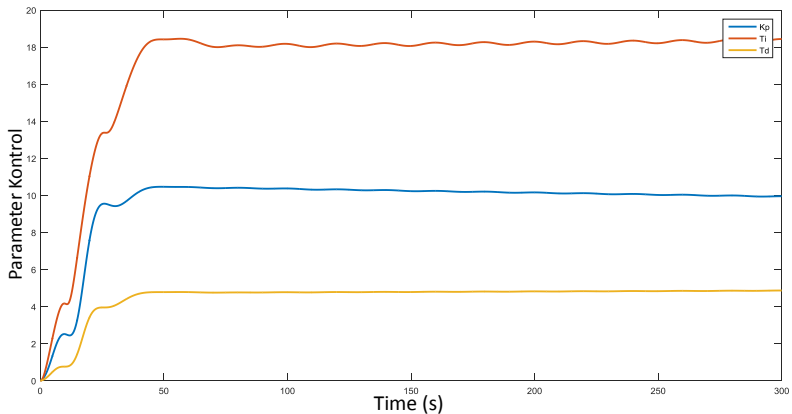
Gambar 4.1 dan 4.2 diatas secara berturut-turut menunjukkan grafik respon dari *plant steam turbine tandem compound* dengan skema kontrol *adaptive PID MRAC basic* dan *normalized* ketika diberi *adaptation gain* sebesar 1 dan input sinyal sinus dengan periode 40 detik. Pada skema *adaptive PID MRAC* dengan *basic MIT Rule*, diperoleh bahwa *peak* awal ketika *start* memiliki nilai sebesar 3.723, sedangkan pada *normalized MIT Rule* nilai *peak* yang dimiliki pada saat *start* lebih besar dari *basic MIT Rule* yakni sekitar 4.621.

Perbedaan ini dikarenakan pada *adaptive PID MRAC* dengan *basic MIT Rule*, nilai dari parameter kontrol yang di-adjust pada *adjustment mechanism* lebih besar dan lebih cepat berubah dibandingkan ketika menggunakan *normalized MIT Rule*, sehingga ketika awal *starting*, parameter kontrol pada *basic MIT Rule* mengalami perubahan nilai parameter yang cukup besar dan dengan nilai parameter kontrol ini respon sistem juga dapat dengan segera mengikuti bentuk respon dari *model reference*. Akan tetapi pada *normalized MIT Rule*, nilai parameter kontroler mengalami perubahan yang cenderung lebih kecil dikarenakan terdapat mekanisme normalisasi pada *adjustment mechanism*-nya sehingga respon yang dihasilkan oleh sistem juga cenderung akan lebih lambat dalam menyesuaikan dengan respon *model reference* pada awalnya.

Untuk mengamati lebih jelas mengenai perubahan parameter kontrol pada kedua skema kontrol *adaptive PID MRAC* diatas, dapat dilihat pada Gambar 4.3 dan 4.4.



Gambar 4. 3 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 1



Gambar 4. 4 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 1

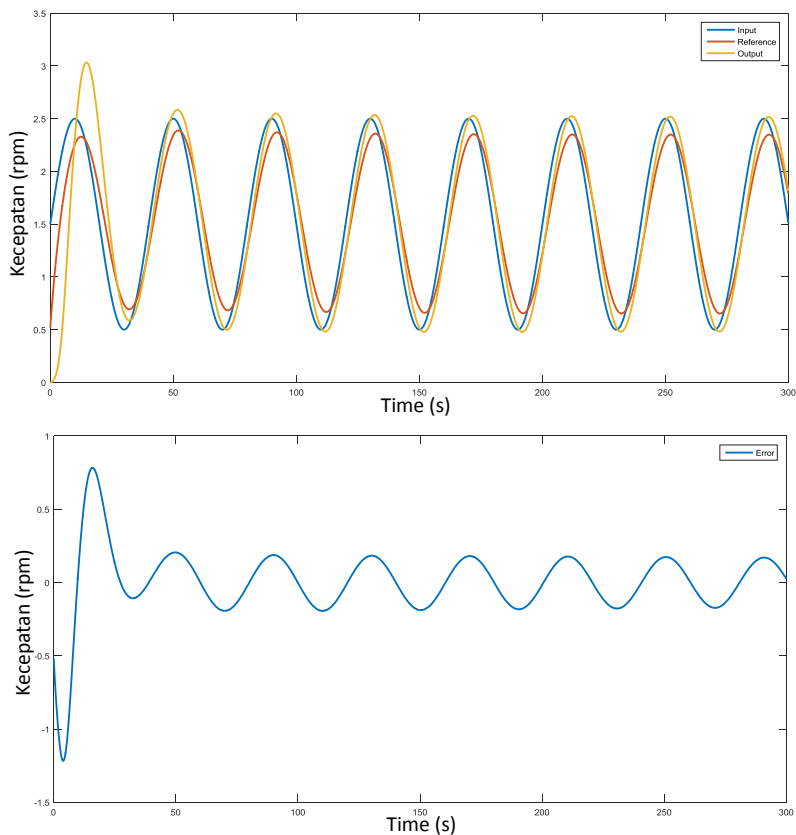
Dari grafik yang ditunjukkan oleh Gambar 4.3 dan 4.4, terlihat bahwa semakin lama nilai dari masing-masing parameter kontrol akan mengalami perubahan yang semakin lambat, hingga ketika nilai respon dari plant dan *model reference* sama atau mendekati, maka nilai dari parameter kontrol ini akan berhenti mengalami perubahan (konvergen).

Walaupun nilai awal dari *output plant* dengan skema *PID MRAC* dengan *basic MIT Rule* lebih kecil nilai *overshoot*-nya jika dibandingkan dengan skema yang menggunakan *normalized MIT Rule*, ketika nilai parameter cenderung tidak berubah nilai output dari *normalized MIT Rule*-lah yang lebih baik dikarenakan *error* yang dimiliki lebih kecil jika dibandingkan dengan skema *basic MIT Rule*. Untuk membuktikannya diambil *sample* nilai dari kedua grafik respon sistem ketika parameter sudah cenderung memiliki perubahan yang kecil. Nilai yang diambil yaitu nilai *amplitude* positif pada detik 52.3, 92.3, 132, 172.2, 211.9, 251.9, dan 292.2. Dari *sample* waktu tersebut, dapat diperoleh bahwa rata-rata dari nilai *overshoot*-nya sebesar $Mean_{OvershootSampleBasic} = 0.19$ untuk skema dengan *basic MIT Rule* dan $Mean_{OvershootSampleNormalized} = 0.164$ untuk skema dengan *normalized MIT Rule*.

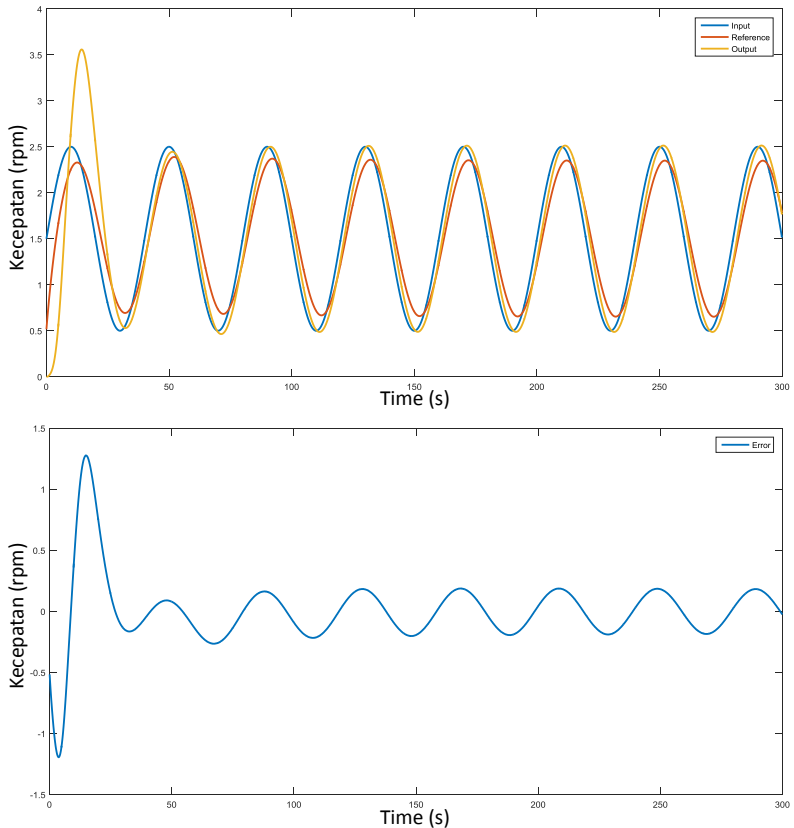
Dari perhitungan tersebut, terlihat bahwa pada kondisi *adaptation gain* = 1 dan input sinyal sinus, ketika parameter kontroler sudah berada pada kondisi yang hampir konvergen, skema *PID MRAC* dengan

normalized MIT Rule memberikan nilai *error* yang sedikit lebih kecil jika dibandingkan dengan *basic MIT Rule*. Hal ini dikarenakan sensitivitas *normalized MIT Rule* terhadap level dari sinyal input lebih kecil bila dibandingkan dengan *MIT Rule* yang biasa sehingga *error* yang ditimbulkan cenderung tidak terlalu besar dan lama-kelamaan akan mendekati nol. Selanjutnya, respon *plant* dengan *input* yang sama dan *adaptation gain* = 3 ditunjukkan pada Gambar 4.5 dan Gambar 4.6.

- Input Sinyal Sinus dengan *Adaptation Gain* = 3



Gambar 4. 5 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 3



Gambar 4. 6 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 3

Ketika sistem diberikan nilai *adaptation gain* yang lebih tinggi dimana kali ini diberikan nilai *gain* sama dengan 3, maka respon dari *plant* memiliki bentuk dan karakteristik yang tidak jauh berbeda dengan sistem ketika menggunakan *adaptation gain* = 1. Yang membedakan adalah pada saat *adaptation gain* dinaikkan, maka kecepatan dan sensitifitas adaptasi pada kedua skema *adjustment mechanism* akan meningkat.

Hal ini dapat dilihat pada kedua respon Gambar 4.5 dan 4.6 dimana pada saat menggunakan *adaptation gain* = 1, ketika menggunakan

skema *PID MRAC* dengan *basic MIT Rule*, *peak* awal pada respon timbul pada saat 17.3 sekon sedangkan pada saat *adaptation gain* dinaikkan menjadi 3 maka *peak* awal timbul pada saat 14.8 sekon. Begitu juga pada skema *PID MRAC* dengan *normalized MIT Rule*, pada saat *adaptation gain* bernilai 1, *peak* awal timbul pada saat 16.4 sekon sedangkan ketika nilai *adaptation gain* dinaikkan, maka *peak* awal mengalami perubahan waktu timbul menjadi 14.4 sekon. Perubahan waktu ini menunjukkan bahwa ketika nilai *adaptation gain* pada skema *adaptive PID MRAC* dinaikkan, maka respon dari *plant* akan semakin cepat beradaptasi dengan respon *model reference*.

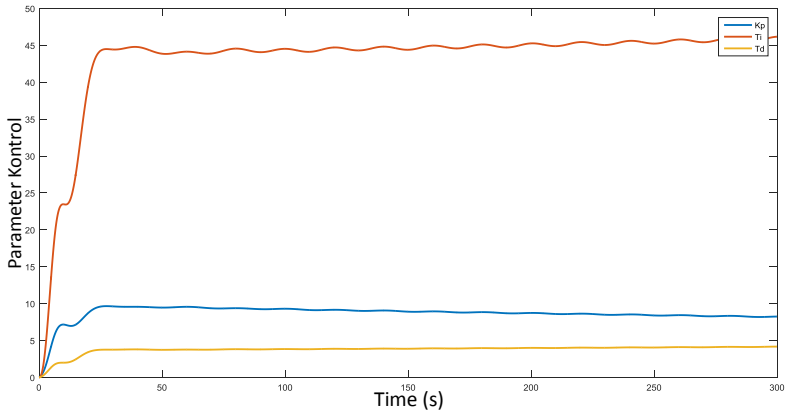
Tidak hanya perubahan waktu, nilai *maximum peak* awal juga ikut mengecil ketika *adaptation gain* dinaikkan menjadi 3. Pada saat nilai *adaptation gain* diperbesar menjadi sama dengan 3, maka nilai *maximum peak* awal untuk skema *PID MRAC* dengan *basic MIT Rule* turun menjadi 3.033 sedangkan untuk skema *PID MRAC* dengan *normalized MIT Rule* juga turun menjadi 3.558. Hal ini akan berdampak baik pada peningkatan performansi sistem dimana perbedaan *peak* yang ditimbulkan antara *model reference* dengan *plant* menjadi lebih kecil sehingga *error* keseluruhan juga turut mengecil.

Selain itu, meningkatnya nilai *adaptation gain* juga dapat memperbaiki *overshoot* yang dapat terjadi. Hal ini dapat dibuktikan dengan cara yang sama dengan pada saat pembuktian rata-rata *overshoot* pada *sample* waktu tertentu saat nilai *adaptation gain* = 1. Pada kali ini, diambil tujuh *sample* nilai respon pada waktu respon mencapai nilai *amplitude*-nya masing-masing, dimana hal ini terjadi pada detik 51.4, 91.5, 131.5, 171.5, 211.5, 251.5, dan 291.5 pada saat menggunakan skema *PID MRAC* dengan *normalized MIT Rule* dan pada detik 51.8, 91.8, 131.8, 171.8, 211.8, 251.8, dan 291.8 pada saat menggunakan skema *PID MRAC* dengan *basic MIT Rule*. Dari *sample* waktu tersebut, dapat diperoleh bahwa rata-rata dari nilai *overshoot*-nya sebesar $Mean_{OvershootSampleBasic} = 0.176$ untuk skema dengan *basic MIT Rule* dan $Mean_{OvershootSampleNormalized} = 0.141$ untuk skema dengan *normalized MIT Rule*.

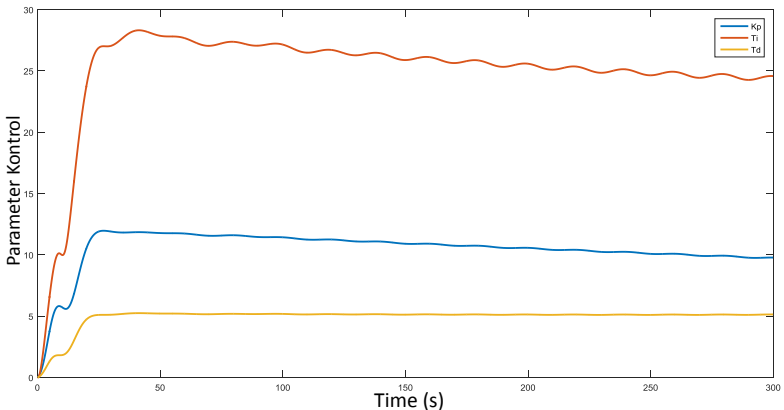
Dari perhitungan tersebut, terlihat bahwa pada kondisi *adaptation gain* = 3 dan input sinyal sinus, ketika parameter kontroler sudah berada pada kondisi yang hampir konvergen, skema *PID MRAC* dengan *normalized MIT Rule* memberikan nilai *error* yang sedikit lebih kecil jika dibandingkan dengan *basic MIT Rule* dan juga jika dibandingkan

dengan skema *normalized MIT Rule* pada saat menggunakan *adaptation gain* sebesar 1.

Selain grafik respon yang berubah, parameter kontroler *PID MRAC* juga ikut mengalami perubahan ketika *adjustment mechanism* diubah dari 1 menjadi 3. Perubahan tersebut dapat dilihat pada Gambar 4.7 dan 4.8.



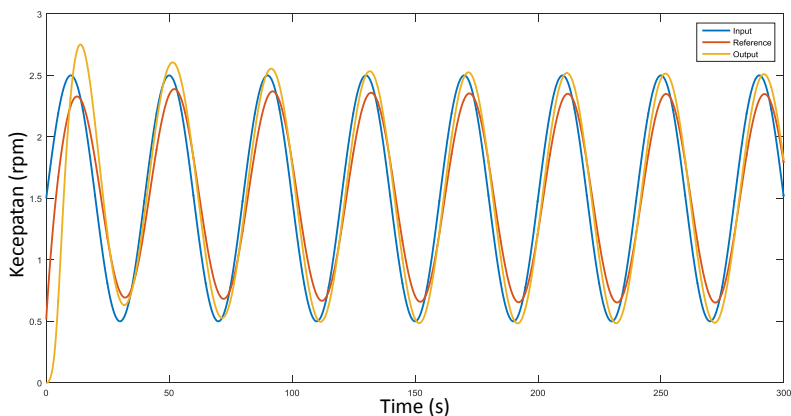
Gambar 4. 7 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 3

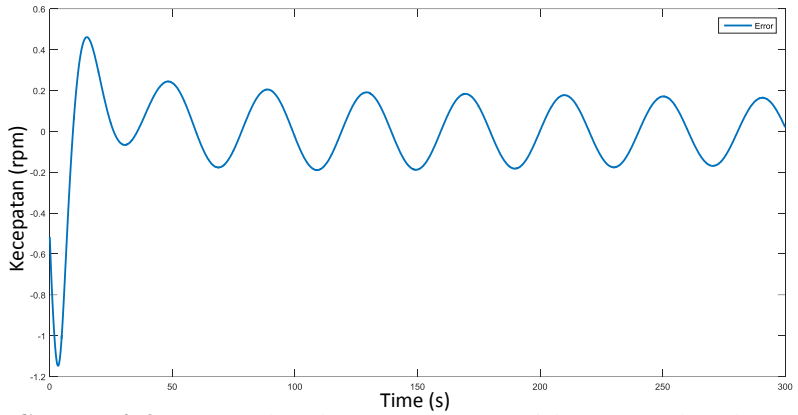


Gambar 4. 8 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 3

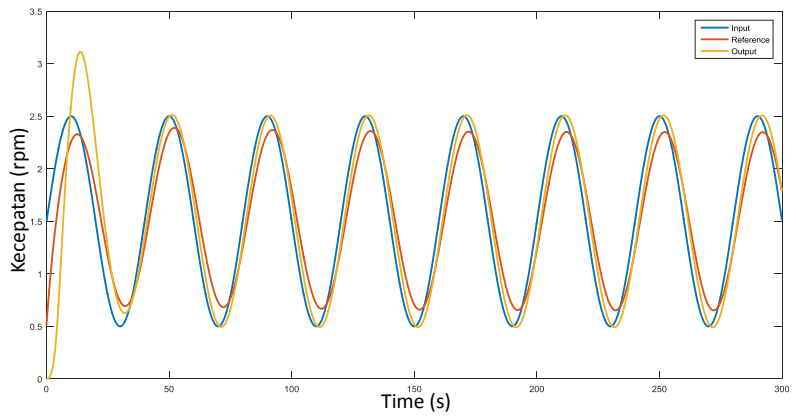
Dari kedua gambar perubahan parameter kontrol pada saat menggunakan *adaptation gain* sebesar 3, terlihat bahwa parameter kontroler pada kedua skema lebih cepat memulai untuk menuju konvergen daripada saat menggunakan *adaptation gain* sebesar 1. Hal ini menunjukkan bahwa ketika *adaptation gain* diperbesar, maka waktu yang diperlukan oleh *adjustment mechanism* untuk mengatur parameter kontroler agar menghasilkan respon yang sesuai dengan *model reference* juga mengalami percepatan sehingga memungkinkan untuk respon pada *plant* lebih cepat dalam melakukan proses adaptasi. Selanjutnya, respon *plant* dengan *input* yang sama dan *adaptation gain* dinaikkan menjadi 5 ditunjukkan pada Gambar 4.9 dan Gambar 4.10.

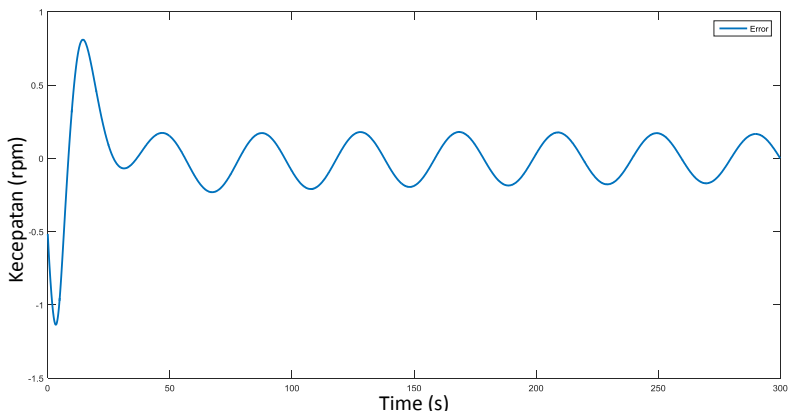
- Input Sinyal Sinus dengan *Adaptation Gain* = 5





Gambar 4. 9 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 5





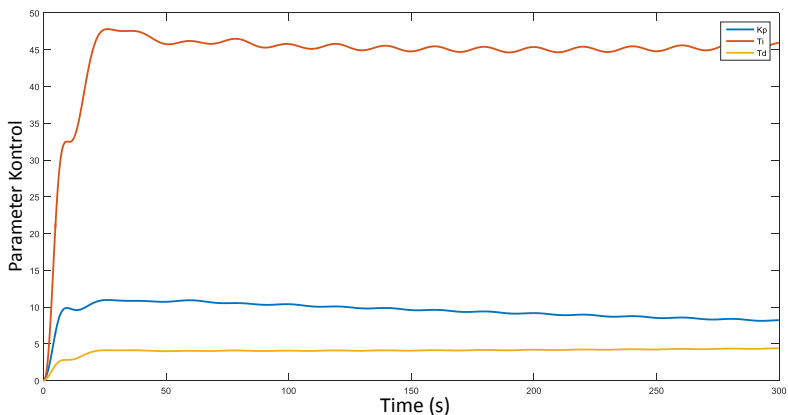
Gambar 4. 10 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 5

Sama seperti dua pengujian sebelumnya, terlihat pada Gambar 4.9 dan 4.10 menunjukkan bahwa dengan menaikkan nilai *adaptation gain*, maka hasil respon dari *plant* juga akan semakin mengikuti nilai respon *model reference*. Secara sederhana, dapat dilakukan pengambilan *sample overshoot* seperti pengujian sebelumnya untuk mengetahui nilai *overshoot* rata-rata yang ditimbulkan ketika *adaptation gain* dinaikkan menjadi 5. Dengan menggunakan nilai *sample* pada detik 51.6, 91.6, 131.6, 171.6, 211.6, 251.6, dan 291.7 untuk skema *basic MIT Rule* dan nilai *sample* pada detik 51.4, 91.4, 131.4, 171.4, 211.4, 251.4, dan 291.7 untuk skema *normalized MIT Rule*, dapat diperoleh bahwa rata-rata dari nilai *overshoot*-nya sebesar $Mean_{OvershootSampleBasic} = 0.1776$ untuk skema dengan *basic MIT Rule* dan $Mean_{OvershootSampleNormalized} = 0.148$ untuk skema dengan *normalized MIT Rule*.

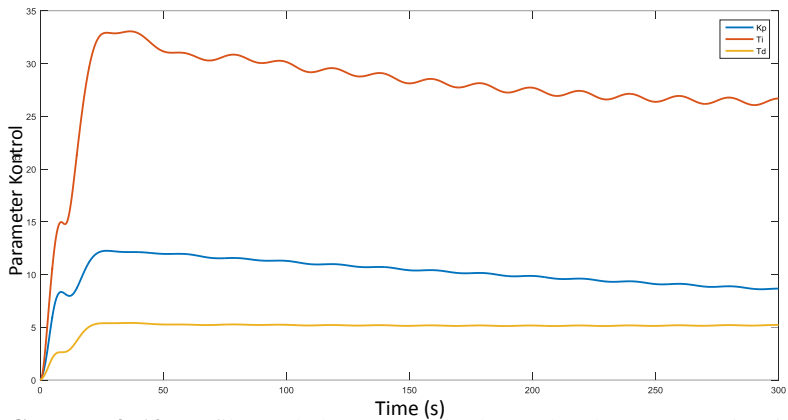
Pada percobaan ketiga ketika menggunakan *adaptation gain* sebesar 5, nilai rata-rata dari *overshoot* pada *sample* yang diambil cenderung tidak mengalami perubahan yang signifikan jika dibandingkan dengan nilai rata-rata *overshoot* ketika menggunakan *adaptation gain* sebesar 3. Bahkan pada saat menggunakan *adaptation gain* sebesar 5, nilai *error* rata-rata *overshoot* sedikit mengalami peningkatan jika dibandingkan dengan *adaptation gain* = 3 yaitu sebesar 0.0016 untuk skema *PID MRAC* dengan *basic MIT Rule* dan 0.007 untuk skema *PID MRAC* dengan *normalized MIT Rule*. Dari sini dapat terlihat bahwa,

meningkatkan nilai *adaptation gain* tidak selalu dapat memperbaiki performansi respon, jika nilai *adaptation gain* terus diperbesar dapat mengakibatkan sensitivitas *adjustment mechanism* dalam melakukan penyesuaian parameter meningkat sehingga dapat menghasilkan parameter kontrol yang terlalu *sensitive* dan berpengaruh pada respon *plant* yang dihasilkan. Bahkan ada kalanya ketika pengaturan *adaptation gain* melebihi suatu nilai tertentu dapat mengakibatkan respon dari *plant* menjadi tak terkendali (*unstable*) dikarenakan sensitivitas yang ada pada *adjustment mechanism* menjadi terlalu besar.

Sama seperti kondisi ketika *adaptation gain* diubah besarnya dari 1 menjadi 3, pada saat perubahan dilakukan kembali tentu terdapat perubahan bentuk parameter kontrol yang diatur pada *adjustment mechanism* pada tiap-tiap skema kontrol. Perubahan tersebut dapat dilihat pada Gambar 4.11 dan 4.12.



Gambar 4. 11 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 5

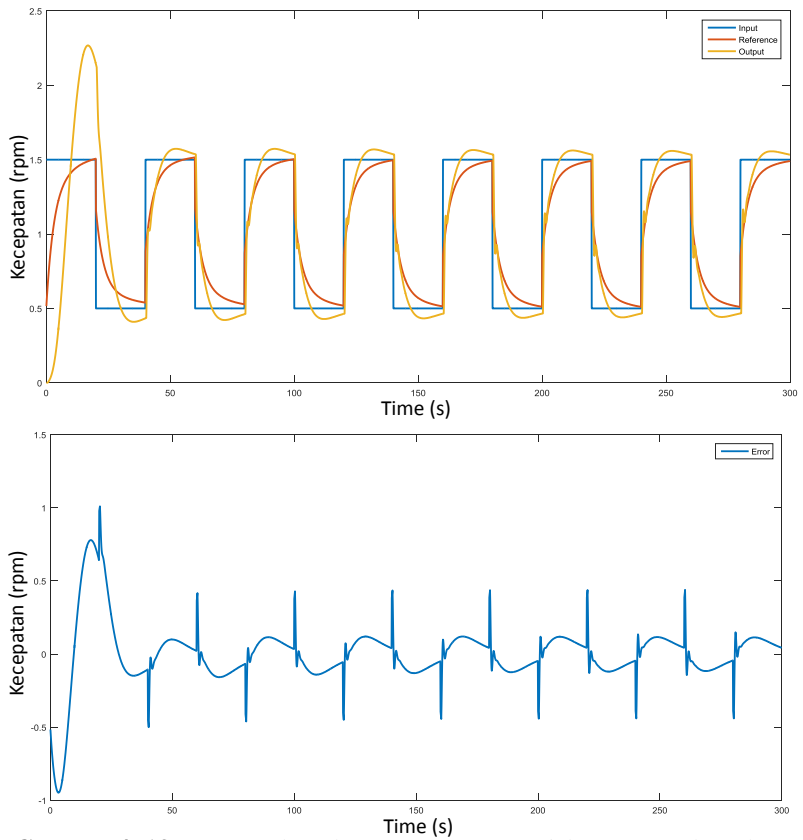


Gambar 4.12 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 5

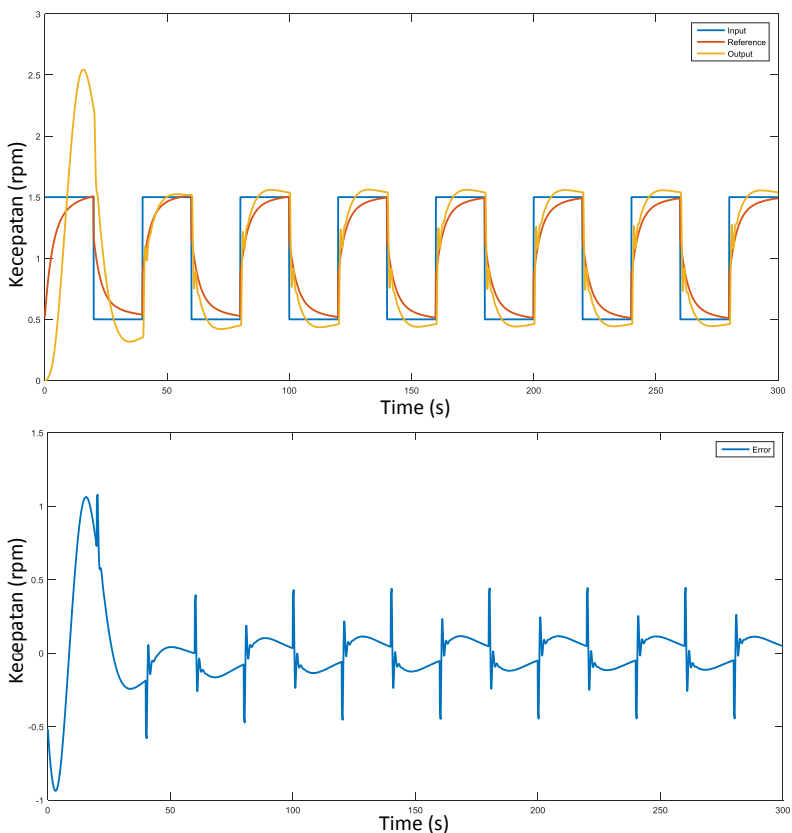
Dari dua gambar perubahan parameter diatas, terlihat bahwa parameter kontroler pada kedua skema lebih cepat menuju konvergen daripada saat menggunakan *adaptation gain* sebesar 1 dan 3. Pada saat *adaptation gain* di-set 1, parameter kontrol pada skema *PID MRAC* dengan *basic MIT Rule* mulai menuju konvergen pada detik ke-44 sedangkan pada skema *PID MRAC* dengan *normalized MIT Rule* mulai menuju konvergen pada detik ke-50. Ketika *adaptation gain* di-set 3, parameter kontrol pada skema *PID MRAC* dengan *basic MIT Rule* mulai menuju konvergen pada detik ke-33 sedangkan pada skema *PID MRAC* dengan *normalized MIT Rule* mulai menuju konvergen pada detik ke-42. Dan pada saat *adaptation gain* di-set menjadi 5, maka waktu konvergen parameter kontrol pada *basic MIT Rule* mengalami kemajuan waktu mulai menjadi 32 detik dan untuk *normalized MIT Rule* juga mengalami kemajuan waktu mulai menjadi 33 detik.

Setelah dilakukan pengujian dengan menggunakan sinyal sinus, selanjutnya dilakukan pengujian menggunakan sinyal *pulse* dengan tanpa beban dan nilai *adaptation gain* yang diubah-ubah dari antara 3 dan 5. Pengujian menggunakan sinyal *pulse* ini bertujuan untuk mengamati respon dari *plant* ketika diberi input yang memiliki perubahan yang cenderung cepat. Hasilnya, pada *plant* yang menggunakan *adaptation gain* = 3 dan *input* sinyal *pulse* dapat dilihat pada Gambar 4.13 dan Gambar 4.14.

- Input Sinyal *Pulse* dengan *Adaptation Gain* = 3



Gambar 4. 13 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 3

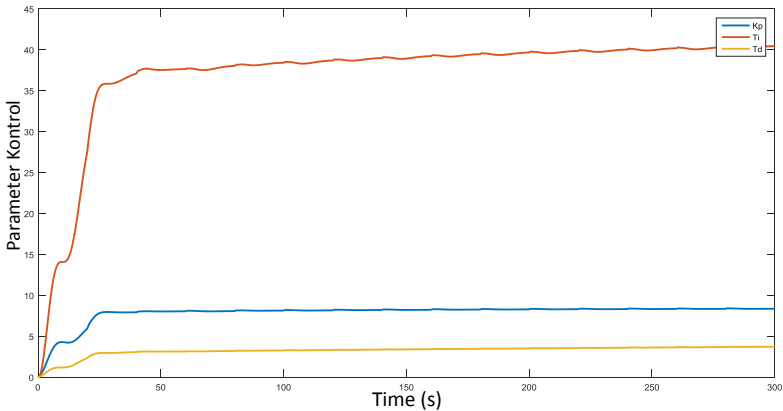


Gambar 4. 14 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 3

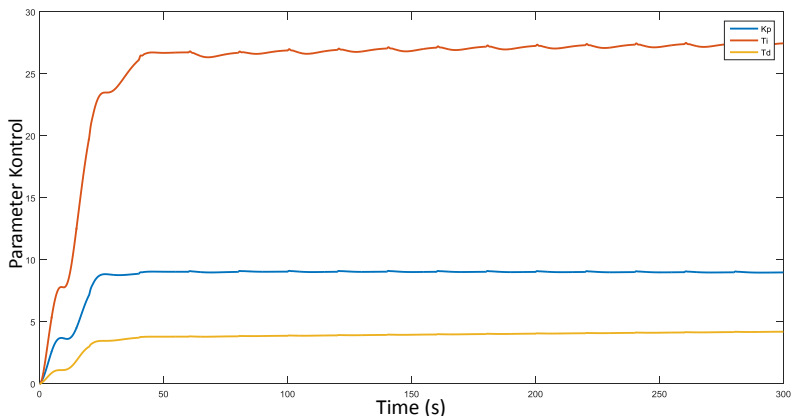
Dengan mengamati kedua grafik respon tersebut, terlihat bahwa respon dari *plant* pada masing-masing skema kontrol mulai mengikuti bentuk respon dari model pada gelombang pulse yang kedua (detik ke-40). Sama seperti pengujian dengan sinyal sinus, pada start awal skema *PID MRAC* dengan *basic MIT Rule* memiliki nilai *peak* yang lebih kecil jika dibandingkan dengan *normalized MIT Rule* yaitu 2.268 pada *basic MIT Rule* dan 2.543 pada *normalized MIT Rule*. Untuk membandingkan kedua respon dari masing-masing skema diatas, dapat dilakukan dengan melakukan perhitungan nilai rata-rata dari *maximum peak* yang

dihasilkan pada saat respon *plant* sudah mulai sama dengan respon model reference (pada detik ke-40), diperoleh bahwa rata-rata dari nilai *maximum peak*-nya sebesar $Mean_{MaxPeakSampleBasic} = 1.565$ untuk skema dengan *basic MIT Rule* dan $Mean_{MaxPeakSampleNormalized} = 1.553$ untuk skema dengan *normalized MIT Rule*.

Apabila diperhatikan dari rata-rata *sample* nilai yang diambil pada masing-masing *peak* pada kedua skema, pada *normalized PID MRAC* memiliki nilai rata-rata *peak* yang lebih baik jika dibandingkan dengan *basic MIT Rule* untuk kondisi *adaptation gain* = 3. Hal ini dikarenakan sifat dari *adjustment mechanism* pada *normalized MIT Rule* yang membuat sensitivitas sistem dalam merespon perubahan *input* lebih kecil jika dibandingkan dengan *basic MIT Rule*, sehingga walaupun terjadi perubahan dengan *input plant* dengan cepat, *normalized MIT Rule* akan berusaha meresponnya dengan lebih tenang sehingga tidak terlalu melonjak tinggi nilai responnya jika dibandingkan dengan *model reference* yang ada. Akan tetapi jika diperhatikan pada kondisi *rise time*, skema *PID MRAC* dengan *normalized MIT Rule* ini sudah menghasilkan respon *plant* yang memiliki osilasi yang sedikit lebih besar jika dibandingkan dengan *basic MIT Rule*. Dalam batasan *adaptation gain* tertentu, osilasi ini bisa menjadikan *plant* masuk kedalam zona respon *unstable*. Kemudian untuk dapat mengamati perbedaan antara kedua skema secara lebih jelas, dapat diamati melalui grafik perubahan parameter kontrol yang ada pada Gambar 4.15 dan 4.16.



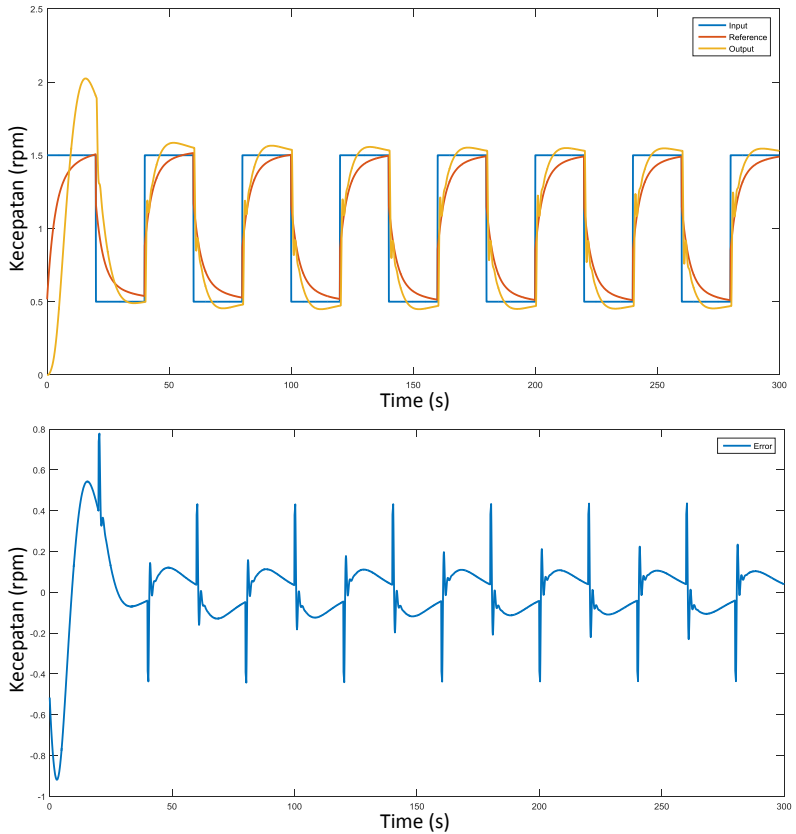
Gambar 4. 15 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 3



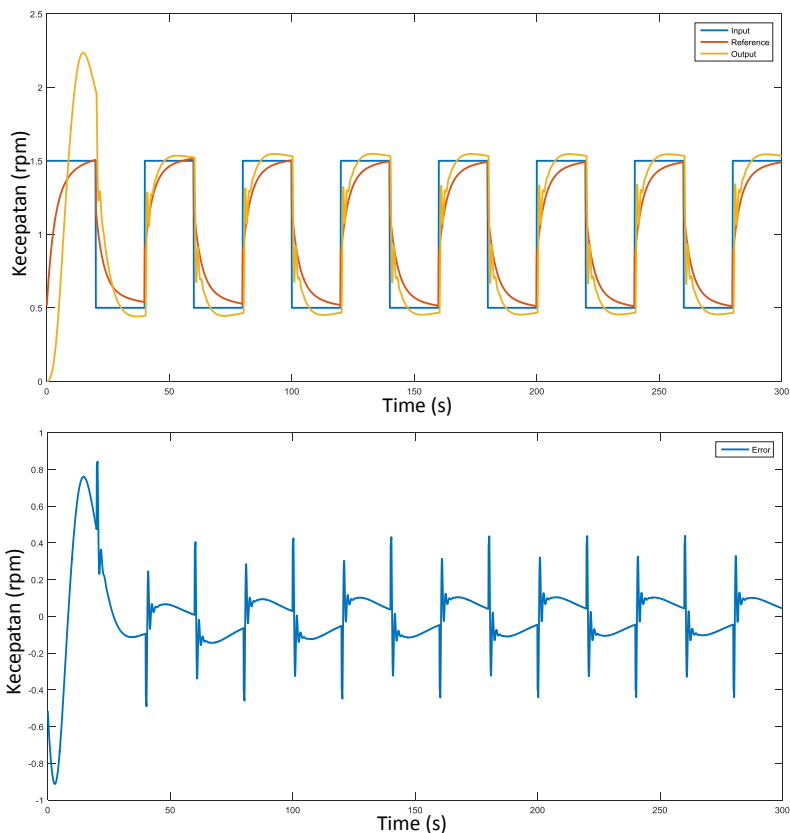
Gambar 4. 16 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 3

Dari kedua gambar tersebut, terlihat bahwa tiap-tiap parameter kontroler pada kedua skema mulai konvergen pada detik ke-30, sehingga respon pada *plant* juga mulai menunjukkan kemiripan dengan bentuk respon *model reference* mulai pada detik tersebut. Selain itu, terlihat juga bahwa perubahan parameter pada skema yang menggunakan *normalized MIT Rule* memiliki nilai yang cenderung lebih kecil jika dibandingkan dengan *basic MIT Rule*, hal ini sesuai dengan sifat *adjustment mechanism* yang dimiliki oleh *normalized MIT Rule* sehingga sensitivitas *plant* terhadap perubahan *input* menjadi lebih kecil. Selanjutnya, respon *plant* dengan *input* yang sama dan *adaptation gain* = 5 ditunjukkan pada Gambar 4.17 dan Gambar 4.18.

- Input Sinyal *Pulse* dengan *Adaptation Gain* = 5



Gambar 4. 17 Respon *plant* dan *error* antara model *reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 5

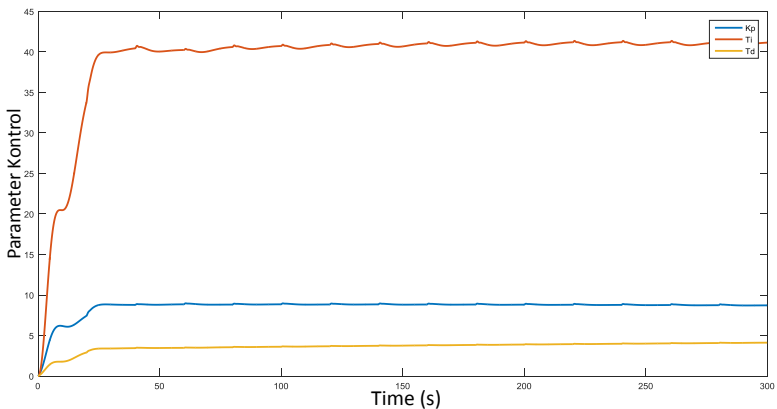


Gambar 4. 18 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 5

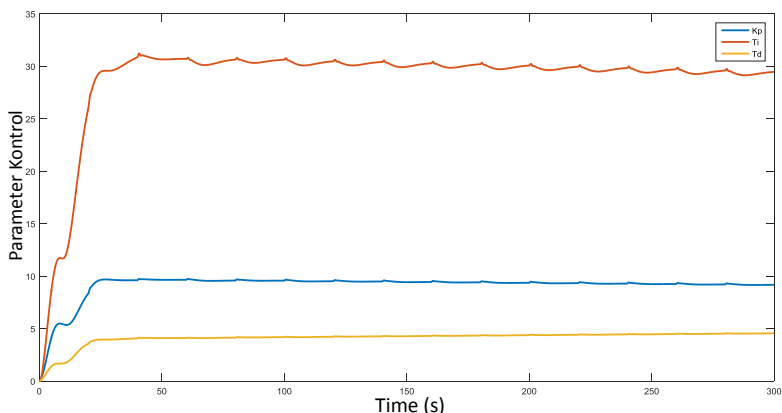
Pada gambar tersebut, terlihat bahwa respon dari *plant* sudah memiliki nilai yang lebih mirip jika dibandingkan dengan pengujian sebelumnya. Hal ini dapat dilihat dari penurunan nilai *peak* yang terjadi pada awal *start* pada masing-masing skema dimana pada *basic MIT Rule* nilai *peak*-nya menjadi 2.024 pada 15.9 detik dan *normalized MIT Rule* menjadi 2.235 pada 14.8. Selain itu, peningkatan juga dapat diamati dengan melakukan perhitungan nilai rata-rata dari *maximum peak* yang dihasilkan pada saat respon *plant* sudah mulai sama dengan respon *model reference* (pada detik ke-40), diperoleh bahwa rata-rata dari nilai

maximum peak-nya sebesar $Mean_{MaxPeakSampleBasic} = 1.557$ untuk skema dengan *basic MIT Rule* dan $Mean_{MaxPeakSampleNormalized} = 1.545$ untuk skema dengan *normalized MIT Rule*.

Pada pengujian kali ini, rata-rata dari *peak* yang diambil pada *sample* tertentu untuk *normalized MIT Rule* memiliki nilai yang lebih baik jika dibandingkan dengan *basic MIT Rule*. Tetapi, walaupun begitu, osilasi yang ditimbulkan pada saat *rise time* yang terdapat pada skema *normalized MIT Rule* terlihat lebih tinggi jika dibandingkan dengan yang ada pada *basic MIT Rule*. Kemudian untuk dapat mengamati perbedaan antara kedua skema secara lebih jelas pada saat *adaptation gain* dinaikkan nilainya, dapat diamati melalui grafik perubahan parameter kontrol yang ada pada Gambar 4.19 dan 4.20.



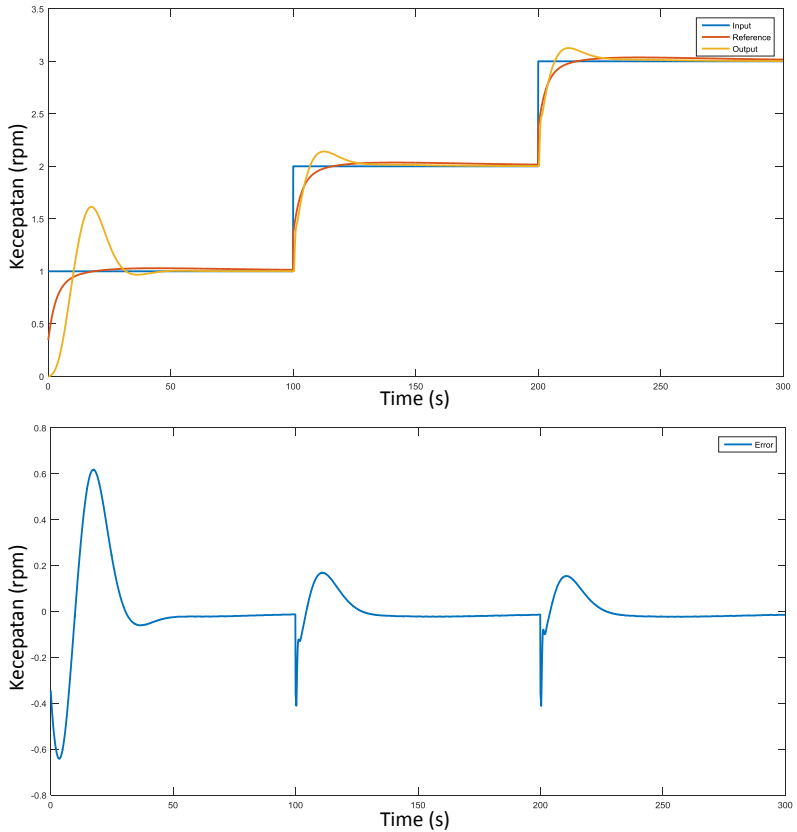
Gambar 4. 19 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 5



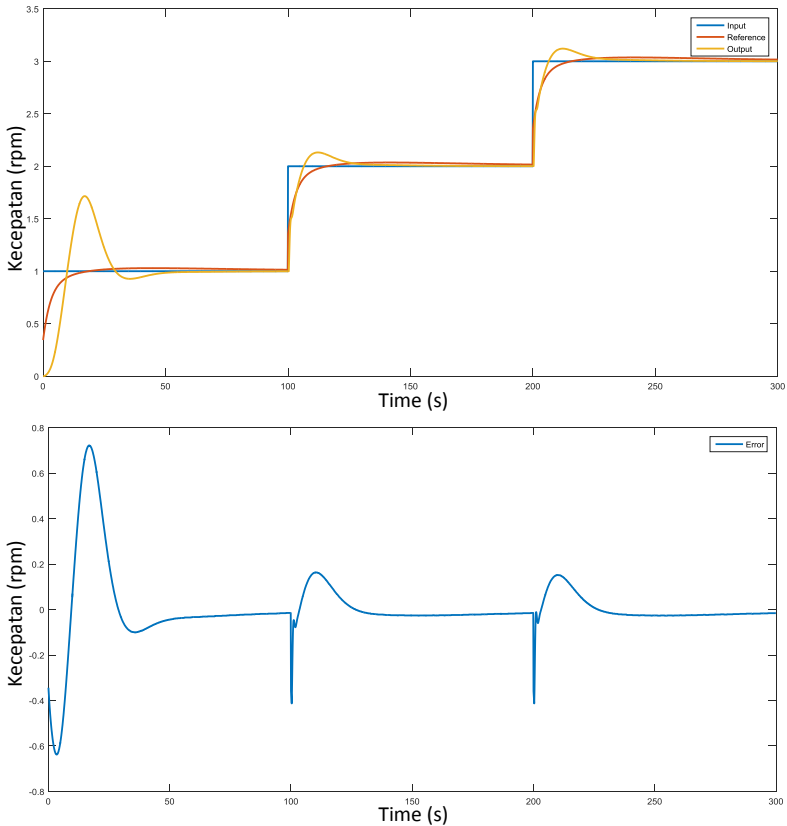
Gambar 4.20 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 5

Dari gambar kedua grafik perubahan parameter diatas, nilai masing-masing parameter kontroler yang diatur pada *adjustment mechanism* mengalami peningkatan ketika menggunakan *adaptation gain* sebesar 5. Hal inilah yang membuat respon dari *plant* lebih cepat menyamai nilai respon dari *model reference* yang ada.

Untuk pengujian berikutnya dilakukan dengan menggunakan input sinyal *step* dan ini merupakan pengujian yang terakhir untuk kondisi *plant* tanpa beban dimana pada pengujian ini akan diamati respon dari *plant* pada kondisi *steady state*. *Input* yang diberikan akan divariasikan antara 1, 2, dan 3 sehingga proses adaptasi juga akan terlihat pada tiap perubahan *input* tersebut sedangkan untuk nilai *adaptation gain* yang digunakan akan ditetapkan sebesar 5. Hasilnya, pada pengujian menggunakan *adaptation gain* = 5 dapat dilihat pada Gambar 4.21 dan Gambar 4.22.



Gambar 4. 21 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 5



Gambar 4. 22 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 5

Pada dua gambar pengujian tersebut, terlihat pada respon *plant* yang menggunakan *PID MRAC* dengan *basic MIT Rule* memiliki nilai *overshoot* awal yang lebih rendah jika dibandingkan dengan *PID MRAC* dengan *normalized MIT Rule*. Pada *basic MIT Rule*, *overshoot* awal bernilai 0.614 sedangkan pada *normalized MIT Rule* nilai *overshoot* cukup tinggi yaitu diangka 0.715.

Perbedaan ini sama seperti pengujian-pengujian sebelumnya dimana pada *PID MRAC* dengan *basic MIT Rule*, nilai dari parameter kontrol yang di-*adjust* pada *adjustment mechanism* lebih besar dan lebih cepat

berubah dibandingkan ketika menggunakan *normalized MIT Rule* sehingga respon pada *basic MIT Rule* lebih rendah *overshoot* awalnya jika dibandingkan dengan *normalized MIT Rule*. Kemudian ketika *gain* pada input step dinaikkan menjadi 2 dan 3 kali, nilai *overshoot* pada kedua skema mulai menunjukkan kesamaan walaupun untuk pengujian kali ini nilai *overshoot* yang ada pada *basic MIT Rule* sedikit lebih besar jika dibandingkan dengan *normalized MIT Rule*.

Pada *basic MIT Rule*, nilai *overshoot* yang dimiliki setelah *gain input step* dinaikkan secara berturut-turut adalah 0.141 dan 0.126, sedangkan pada *normalized MIT Rule* pada pengujian yang sama memiliki *overshoot* sebesar 0.132 dan 0.121. Kemudian untuk *settling time* awal yang dicapai pada *basic MIT Rule* sebesar $T_{sBas1}(\pm 2\%) = 41.1$ sekon dimana nilai ini cenderung lebih cepat jika dibandingkan dengan *settling time* awal *normalized MIT Rule* pada pengujian yang sama yakni sekitar $T_{sNorm1}(\pm 2\%) = 47.9$ sekon. Berbeda halnya ketika *input step* dinaikkan menjadi 2 kali, *settling time* pada skema *normalized MIT Rule* cenderung lebih cepat daripada *basic MIT Rule* dimana pada saat *input step* dinaikkan menjadi 2 kali, respon *plant* pada skema *basic MIT Rule* memerlukan waktu settle sebesar $T_{sBas2}(\pm 2\%) = 130.6$ sekon sedangkan pada *normalized MIT Rule* pada kondisi yang sama hanya memerlukan $T_{sNorm2}(\pm 2\%) = 129.3$ sekon. Ketika *input step* dinaikkan menjadi 3 kali, respon *plant* pada *basic MIT Rule* memerlukan waktu settle sebesar $T_{sBas3}(\pm 2\%) = 211.9$ sekon sedangkan pada *normalized MIT Rule* memerlukan waktu sebesar $T_{sNorm3}(\pm 2\%) = 212$ sekon yang artinya sedikit lebih lama dibandingkan skema *basic MIT Rule*.

Dari sini terlihat pengujian menggunakan *input sinyal step* dengan perbesaran 1, 2, dan 3 kali dengan *adaptation gain* sebesar 5 menunjukkan bahwa nilai *overshoot* pada skema PID MRAC dengan *normalized MIT Rule* cenderung lebih rendah jika dibandingkan dengan skema *basic MIT Rule* ketika parameter kontrol sudah mulai memasuki kondisi konvergen, dan nilai *settling time* yang dimiliki skema *normalized MIT Rule* cenderung lebih cepat jika dibandingkan dengan *basic MIT Rule* kecuali pada saat *gain* dinaikkan menjadi 3 kali yang membuat *settling time* pada *basic MIT Rule* sedikit lebih cepat 0.1 sekon dibandingkan dengan *normalized MIT Rule*.

Pada pengujian ini juga didapatkan nilai *RMSE (Root Mean Square Error)* antara respon *plant* yang menggunakan skema kontrol *adaptive PID MRAC* dengan respon *model reference* yang digunakan. Dengan menggunakan sinyal uji berupa sinyal *step* bertingkat dan *adaptation*

gain sebesar 5 dalam kondisi tanpa beban, didapatkan nilai *RMSE* untuk skema yang menggunakan *basic MIT Rule* dan *normalized MIT Rule* secara berturut-turut 0.1466 dan 0.1569. Dari sini terlihat, walaupun nilai *overshoot* rata-rata yang dihasilkan oleh *plant* yang menggunakan skema *normalized MIT Rule* lebih kecil dari pada skema *basic MIT Rule* pada saat parameter kontroler sudah mulai konvergen, nilai *RMSE* pada *basic MIT Rule* masih lebih kecil jika dibandingkan dengan skema *normalized MIT Rule*. Hal ini karena nilai *RMSE* ini didapatkan berdasarkan data respon *plant* dan *model reference* pada keseluruhan waktu pengujian, sehingga dikarenakan pada waktu awal *start* nilai *overshoot plant* ketika menggunakan skema *normalized MIT Rule* terlampaui cukup tinggi dibandingkan dengan *basic MIT Rule* maka nilai *RMSE* pada *normalized MIT Rule* pun juga terpengaruh dan menjadi lebih tinggi.

4.2 Pengujian Menggunakan *Input Sinyal Step Bertingkat* dengan Parameter Sistem Berubah-ubah dan dalam Kondisi Berbeban

Setelah sebelumnya dilakukan pengujian dengan menggunakan simulasi pada *LabVIEW* untuk skema kontrol *adaptive PID MRAC* dengan *basic MIT Rule* dan *normalized MIT Rule* pada *plant steam turbine tandem compound* dengan kondisi tanpa beban dan parameter sistem yang konstan, pada kali ini akan dilakukan pengujian kembali kedua skema kontrol tersebut dengan kondisi dibebani kecepatan *output*-nya. Selain itu pada pengujian ini juga digunakan parameter sistem yang dapat berubah diwaktu simulasi. Untuk parameter sistem yang digunakan dapat dilihat pada Tabel 4.2 dimana parameter tersebut merupakan parameter sistem yang didapat pada kondisi ideal dan parameter yang didapat menggunakan data *heat balance map* pada *steam turbine tandem compound 200-MW*.

Untuk parameter sistem pada kondisi ideal akan digunakan pada rentang waktu 1/3 awal dan 1/3 akhir dari total waktu pengujian (pada detik ke 0 – 166.7 dan 333.4 – 500) sedangkan untuk parameter sistem yang diperoleh dari penurunan menggunakan data *heat balance map* akan digunakan pada rentang waktu 166.8 – 333.3 sekon.

Tabel 4. 2 Parameter yang digunakan pada pengujian berbeban

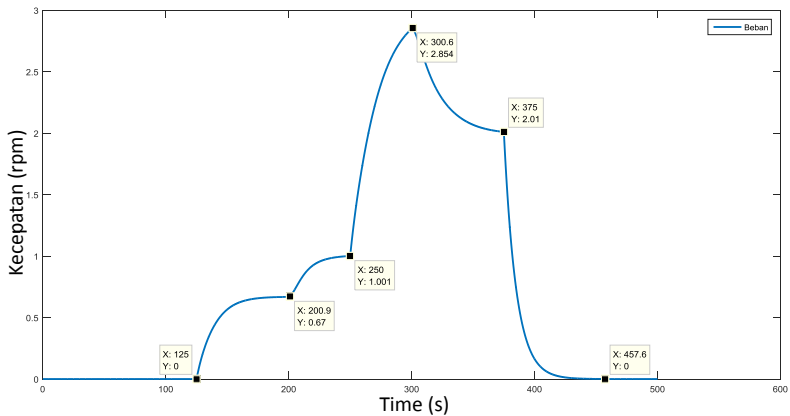
Parameter Sistem	K_{HP}	K_{MP}	K_{LP}	T_{HP}	T_{MP}	T_{LP}	T_{SM}	T_{SR}	T_L
Ideal	0.3	0.3	0.4	0.25	7.5	0.5	0.3	0.1	12

Heat Balance Map	0.31	0.5	0.19	0.396	6.84	0.052	0.25	0.1	12
------------------	------	-----	------	-------	------	-------	------	-----	----

Pengujian dilakukan dengan *input* berupa sinyal *step* bertingkat dan dilakukan dengan *adaptation gain* sebesar 5. Untuk pembebanan pada pengujian ini dilakukan pada *output plant* sehingga kecepatan dari turbin akan berkurang pada saat dibebani.

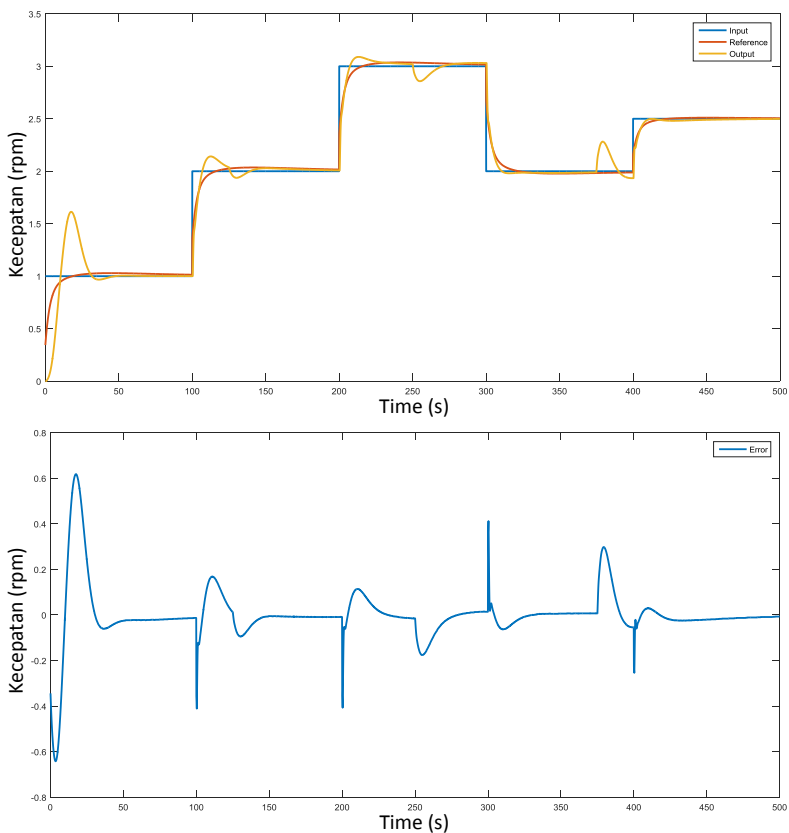
- Skema kontrol *Adaptive PID MRAC* dengan *basic MIT Rule*

Pada pengujian menggunakan skema *adaptive PID MRAC* dengan *basic MIT Rule*, digunakan beban variatif naik dan turun. Beban ini diperoleh dari prosentase langsung kecepatan putar turbin sehingga beban juga akan ikut bervariasi tergantung dengan putaran turbin. Variasi beban yang diberikan pada pengujian kali ini dapat ditunjukkan pada Gambar 4.23.



Gambar 4. 23 Variasi beban kecepatan yang diberikan pada pengujian kontroler *PID MRAC* dengan *basic MIT Rule*

Dengan menggunakan variasi beban dan perubahan parameter yang sudah ditentukan, didapatkan respon dari *plant* yang dikontrol menggunakan skema *adaptive PID MRAC* dengan *basic MIT Rule* seperti pada Gambar 4.24.



Gambar 4. 24 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 5 ketika diberi pembebanan dan perubahan parameter

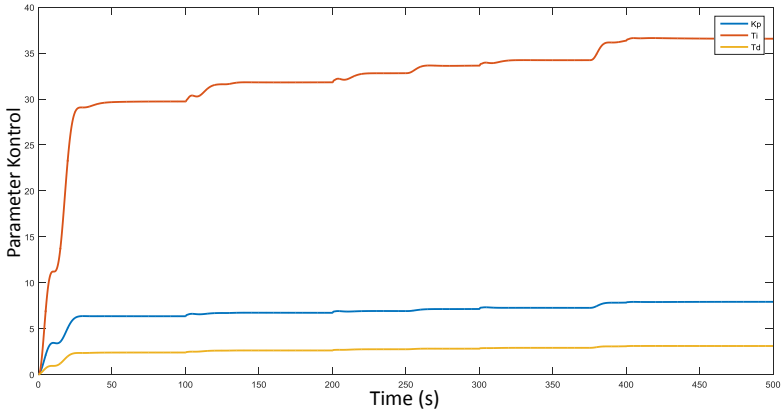
Pada gambar tersebut, terlihat bahwa pada saat awal dibebani pada detik ke-125 hingga detik ke-200.9 dan pada rentang nilai ini juga terjadi perubahan parameter yang dimulai pada detik ke-166.7, respon mulai menunjukkan penurunan nilai kecepatan, akan tetapi sinyal kontrol yang diberikan oleh kontroler *PID MRAC* berusaha agar respon tidak turun terlalu jauh sehingga pada detik ke-130, respon dari *plant* sudah mulai menunjukkan kenaikan dan mulai untuk menuju kondisi normal hingga detik ke-149. Pada saat awal pembebanan ini, *magnitude* dari beban

awal bernilai sekitar 0.67 pada detik ke-200.9, tetapi respon pada plant hanya mengalami penurunan magnitudo sekitar 0.063 dari *set point* dan berhasil kembali menuju bentuk aslinya dengan *recovery time* sebesar 24 sekon, kondisi ini dapat dipertahankan sampai detik ke-200.9. Kemudian pada detik ke-200.9 sampai detik ke-250, *magnitude* dari respon *plant* dapat tetap terjaga dengan baik dan tidak ada penyimpangan yang signifikan terhadap bentuk respon ketika kondisi tanpa beban walaupun *magnitude* dari beban pada rentang detik ini dinaikkan menjadi 1.001.

Beban puncak diberikan pada *range* waktu 250 sekon sampai 300.6 sekon, dimana *magnitude* puncak berada pada detik ke-300.6 dengan nilai 2.854. Pada pembebanan ini, respon *plant* kembali menunjukkan penurunan nilai, akan tetapi kontroler tetap berusaha melakukan aksi kontrol sehingga penurunan yang terjadi tidak terlalu jauh dari *set point*, yaitu hanya sekitar 0.142 dan kembali pada kondisi normal pada detik ke-277 sehingga pada pembebanan ini memiliki *recovery time* sebesar 27 sekon. Kondisi normal ini dapat dipertahankan sampai proses penurunan beban mulai berlangsung, dimana beban mulai mengalami penurunan awal pada *range* waktu 300.6 sekon sampai 375 sekon dan pada rentang ini juga terjadi perubahan parameter kembali. Penurunan awal ini diturunkan hingga *magnitude* beban memiliki nilai sebesar 2.01 atau diturunkan sebesar 0.844 dari kondisi beban puncak. Pada penurunan ini, respon *plant* tidak menunjukkan adanya kenaikan kecepatan yang signifikan hingga detik ke-375. Dan kemudian beban diturunkan sampai menuju nol pada *range* waktu 375 sekon sampai 457.6 sekon. Hasilnya respon dari *plant* kembali menunjukkan penyimpangan kecepatan, akan tetapi kontroler berhasil meredam agar penyimpangan yang terjadi tidak terlalu besar dimana simpangan yang terjadi hanya sekitar 0.282 pada detik ke-379.5 dan kemudian kembali turun dan menyimpang hingga 0.066 pada detik ke-400.3 hingga pada akhirnya respon kembali mengalami penurunan menjadi 2.48 pada detik ke-427 ketika nilai *set point* dinaikkan menjadi 2.5. Dan ketika beban sudah dilepas semua pada detik ke-457.6 respon kembali menuju bentuk aslinya dan mempertahankan hingga pengujian berakhir.

Pada pengujian ini, pengaruh dari perubahan parameter berhasil ditangani oleh skema kontrol *PID MRAC* dengan *basic MIT Rule* dan tidak memberikan pengaruh yang signifikan. Penyimpangan-penyimpangan yang relatif kecil ini berhasil dihasilkan karena kontroler *PID MRAC* tetap melakukan pengaturan besar parameter kontroler

dalam *adjustment mechanism* ketika terdapat perubahan respon *plant* akibat penambahan beban maupun perubahan parameter sistem. Grafik perubahan parameter kontroler tiap waktu dapat dilihat pada Gambar 4.25.



Gambar 4. 25 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 5 ketika diberi pembebanan dan perubahan parameter

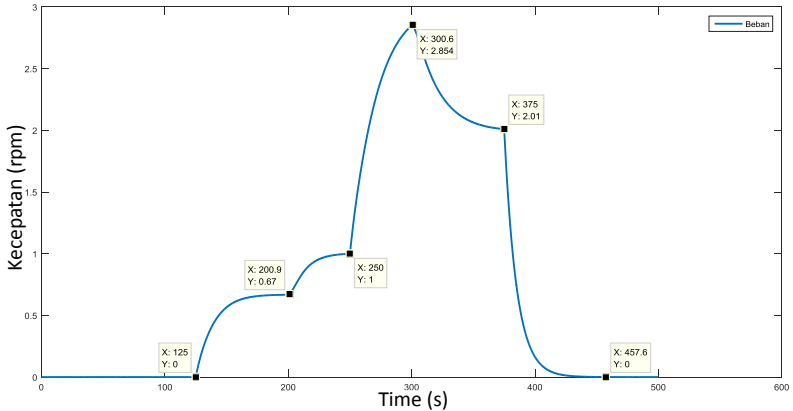
Dari grafik tersebut, terlihat bahwa parameter kontroler akan mulai konvergen pada detik ke-40, tetapi parameter-parameter tersebut akan kembali mengalami perubahan nilai tiap terjadi perubahan-perubahan respon *plant* baik karena perubahan parameter maupun pembebanan.

Dengan perubahan parameter kontroler ini, maka *output* dari *plant* dapat dijaga bentuk responnya agar tetap seperti respon yang terdapat pada *model reference* dan tetap tidak menyimpang jauh dari *set point*.

- Skema kontrol *Adaptive PID MRAC* dengan *normalized MIT Rule*

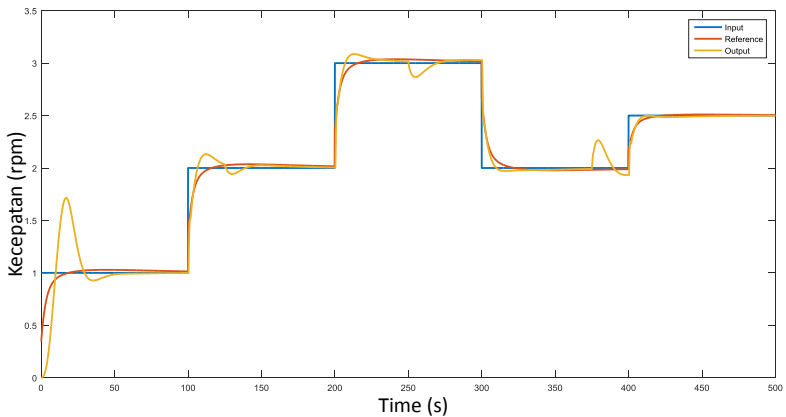
Sama seperti pengujian pada skema *basic MIT Rule*, pada pengujian *adaptive PID MRAC* dengan *normalized MIT Rule* ini juga dilakukan dengan beban variatif naik dan turun. Beban yang digunakan pada pengujian ini juga sama polanya dan diambil dari prosentase kecepatan turbin sehingga diharapkan perubahan beban lebih bervariasi. Grafik dari beban yang digunakan untuk membebani *plant steam turbine*

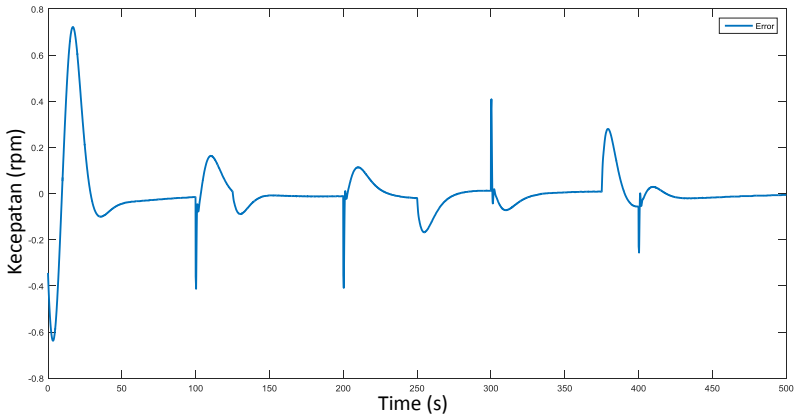
tandem compound dengan skema *normalized MIT Rule* dapat dilihat pada Gambar 4.26.



Gambar 4. 26 Variasi beban kecepatan yang diberikan pada pengujian kontroler *PID MRAC* dengan *normalized MIT Rule*

Dengan menerapkan skenario pengujian berupa pembebanan dan perubahan parameter sistem pada *plant*, maka dapat diperoleh hasil respon dari *plant* yang dikontrol menggunakan kontroler *adaptive PID MRAC* dengan *normalized MIT Rule* dan diberi *input* sinyal *step* yang berubah-ubah seperti pada Gambar 4.27.





Gambar 4. 27 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 5 ketika diberi pembebanan dan perubahan parameter

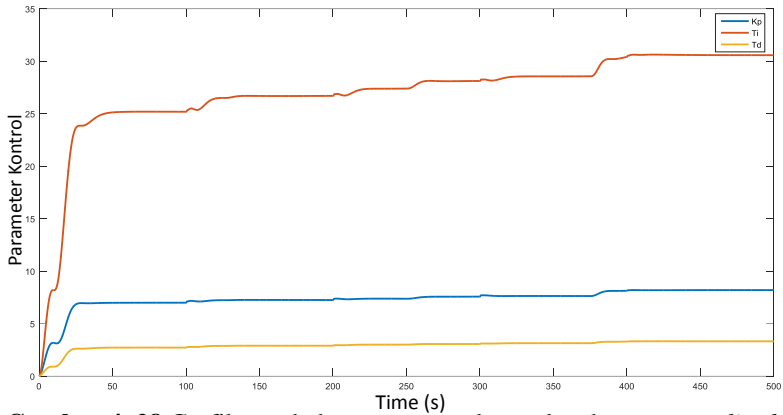
Saat pembebanan awal yang dimulai pada detik ke-125, respon *plant* mulai terlihat mengalami penurunan nilai, akan tetapi kontroler *PID MRAC* dengan *normalized MIT Rule* menjaga agar penurunan ini tidak terlalu jauh dan dapat segera kembali menuju ke nilai aslinya. Hasilnya pada awal pembebanan ini, nilai respon *plant* hanya turun menjadi 1.943 pada detik ke-130 atau sekitar 0.057. Apabila dibandingkan dengan skema yang menggunakan *basic MIT Rule*, maka penurunan yang dihasilkan oleh *normalized MIT Rule* sedikit lebih kecil dikarenakan pada *basic MIT Rule* untuk kondisi yang sama memiliki penurunan nilai sebesar 0.063. Kemudian ketika beban menuju nilai 1 pada detik ke-250, *plant* tidak mengalami penurunan nilai yang signifikan dan berhasil mempertahankan bentuk respon agar sesuai dengan *reference* sampai dengan beban dinaikkan kembali.

Pada saat beban dinaikkan menuju puncak yaitu dengan nilai sekitar 2.854 pada detik ke-300.6, respon kembali mengalami penurunan nilai. Hanya saja, pada skema dengan *normalized MIT Rule* ini, nilai penurunan respon sedikit lebih baik jika dibandingkan dengan skema *basic MIT Rule*. Pada skema *normalized MIT Rule*, respon mengalami penurunan menuju nilai 2.866 pada detik ke-255 atau sekitar 0.134 dari *set point*. Penyimpangan nilai ini sedikit lebih kecil dari skema *basic MIT Rule* yang memiliki nilai simpangan penurunan sebesar 0.142 dari *set point*. Kemudian beban mulai diturunkan pada detik ke-300.6 dengan

penurunan awal dilakukan sampai beban memiliki nilai 2.01 pada detik ke-375. Hasilnya sama seperti ketika menggunakan PID MRAC dengan *basic MIT Rule*, nilai respon berhasil ditahan sehingga penyimpangan nilai yang diakibatkan oleh perubahan beban tidak begitu berpengaruh. Baru ketika beban diturunkan hingga menuju nol, penyimpangan respon kembali terjadi. Penyimpangan terbesar terjadi pada detik ke-379.5 dengan nilai *magnitude* sebesar 2.264 dan dilanjutkan dengan penyimpangan turun menuju nilai 1.932 pada detik ke-400.3. Walaupun begitu, nilai ini tetap lebih rendah jika dibandingkan dengan penyimpangan yang dialami oleh *plant* ketika menggunakan skema dengan *basic MIT Rule* yang memiliki nilai simpangan sampai 2.282 pada detik ke-379.5 dan nilai simpangan sebesar 1.934 pada detik ke-400.3. Kondisi respon kemudian dapat dipertahankan untuk tidak menyimpang lebih jauh lagi hingga beban berhasil dilepas sepenuhnya pada detik ke-457.6.

Untuk *recovery time* yang diberikan pada setiap pembebanan untuk skema dengan *normalized MIT Rule* ini memiliki nilai yang sama dengan pada skema yang menggunakan *basic MIT Rule*.

Pada pengujian ini, terlihat bahwa respon dari *plant* berhasil ditahan oleh skema kontrol PID MRAC dengan *normalized MIT Rule* agar tidak mengalami penyimpangan yang terlalu besar dari *set point* walupun terdapat pembebanan pada *plant* dan perubahan parameter sistem. Selain itu, penyimpangan respon yang dihasilkan oleh *plant* dengan skema kontrol PID MRAC *normalized MIT Rule* memiliki nilai yang cenderung lebih kecil jika dibandingkan dengan skema yang menggunakan *basic MIT Rule*, hal ini disebabkan oleh *adjustment mechanism* pada *normalized MIT Rule* berhasil mengatur perubahan nilai parameter kontrol menjadi tidak terlalu besar sehingga respon *plant* tidak terlalu *sensitive* dalam mengikuti perubahan yang disebabkan oleh pembebanan maupun perubahan parameter sistem. Grafik perubahan parameter kontroler tiap waktu dapat dilihat pada Gambar 4.28.



Gambar 4.28 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 5 ketika diberi pembebanan dan perubahan parameter

Apabila diperhatikan grafik perubahan parameter kontroler antara kedua skema, maka nilai parameter T_i pada skema *normalized MIT Rule* cenderung lebih kecil daripada skema *basic MIT Rule* dan kondisi sebaliknya terjadi pada parameter kontrol K_p dan T_d dimana untuk kedua parameter ini nilai pada skema *normalized MIT Rule* lebih besar daripada *basic MIT Rule*. Hal inilah yang mengakibatkan respon dari *plant* yang menggunakan *PID MRAC* dengan *normalized MIT Rule* memiliki *error* yang relatif lebih kecil dibanding skema yang menggunakan *basic MIT Rule* ketika sudah mencapai waktu yang cukup lama.

Untuk perbandingan performansi dari skema *adaptive PID MRAC* pada kondisi berbeban dan parameter sistem yang berubah-ubah, dilakukan dengan mengamati *RMSE* yang dihasilkan antara respon *plant* dengan respon *model reference* pada saat pengujian menggunakan *input* sinyal *step* bertingkat dan *adaptation gain* = 5 dalam kondisi berbeban. Nilai *RMSE* yang dihasilkan oleh skema *adaptive PID MRAC basic MIT Rule* dan *normalized MIT Rule* secara berturut-turut adalah 0.1225 dan 0.1291. Sama seperti pada saat pengujian tanpa beban dengan *input* sinyal *step*, pada kondisi berbeban ini nilai *RMSE* pada skema *basic MIT Rule* terlihat lebih kecil jika dibandingkan dengan skema *normalized MIT Rule* walaupun pada kondisi parameter kontroler yang sudah mulai konvergen. Hal ini dikarenakan respon awal yang diberikan oleh *plant*

dengan skema *normalized MIT Rule* memiliki nilai *overshoot* yang terpaut cukup tinggi dibandingkan dengan *basic MIT Rule* sehingga nilai *RMSE* keseluruhan pada plant dengan skema *normalized MIT Rule* juga cukup tinggi.

4.3 Pengujian Menggunakan PLC Siemens S7 300 dengan Input Sinyal Step Bertingkat Kondisi Parameter Sistem Berubah-ubah

Pada pengujian selanjutnya, dilakukan penerapan skema *adaptive PID MRAC* kedalam *hardware PLC Siemens S7 300* dan *steam turbine tandem compound virtual plant VI*. Seperti yang dijelaskan pada Bab 3, pertukaran data yang berupa parameter sistem, *variable-variable* pada proses, dan beberapa *setting mode* antara *adjustment mechanism LabVIEW VI* dan *steam turbine tandem compound virtual plant LabVIEW VI* dilakukan menggunakan *shared variable LabVIEW*.

Sedangkan untuk transfer data parameter kontrol yang telah diatur pada *adjustment mechanism VI* menuju kontroler *PID* pada *PLC Siemens S7 300* dilakukan melalui *software IBH OPC Server* yang memanfaatkan komunikasi *serial* melalui *USB MPI Adapter*. Untuk menyalurkan nilai *set point*, *process variable*, dan *manipulated variable* antara kontroler *PID* pada *PLC Siemens S7 300* digunakan *Arduino UNO board* sebagai *I/O interface*.

Pengujian kali ini dilakukan dengan menggunakan *adaptation gain* sebesar 3 dan dengan parameter sistem yang berubah-ubah antara parameter sistem dengan kondisi ideal dan parameter sistem yang diperoleh dari data *heat balance map* sesuai dengan Tabel 4.2. *Input* yang digunakan untuk pengujian kali ini berupa sinyal *step* bertingkat dengan variasi *gain* pada *input* antara 1, 2, dan 3.

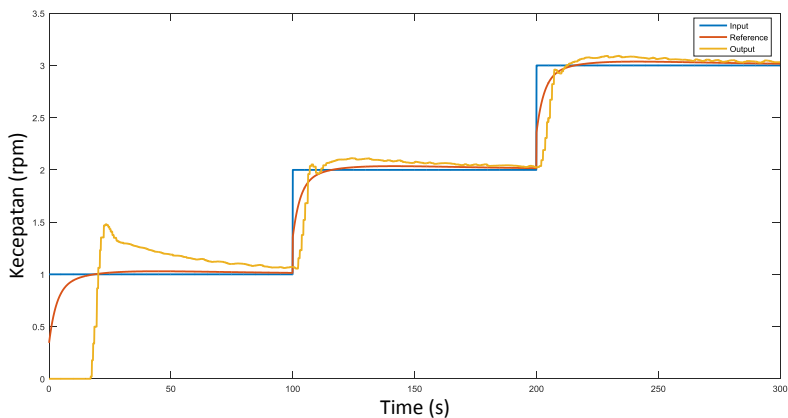
Parameter sistem mengalami perubahan parameter sebanyak 4 kali dalam pengujian yang dijalankan selama 300 sekon. Pada saat awal dimulainya pengujian, parameter yang digunakan oleh sistem merupakan parameter hasil penurunan yang ideal. Ketika melewati detik ke-51.1, parameter sistem berubah menjadi nilai yang didapatkan dari *data heat balance map*. Parameter sistem akan kembali menjadi parameter ideal ketika mencapai detik ke-121.3 dan mengalami perubahan nilai pada detik ke-262.3.

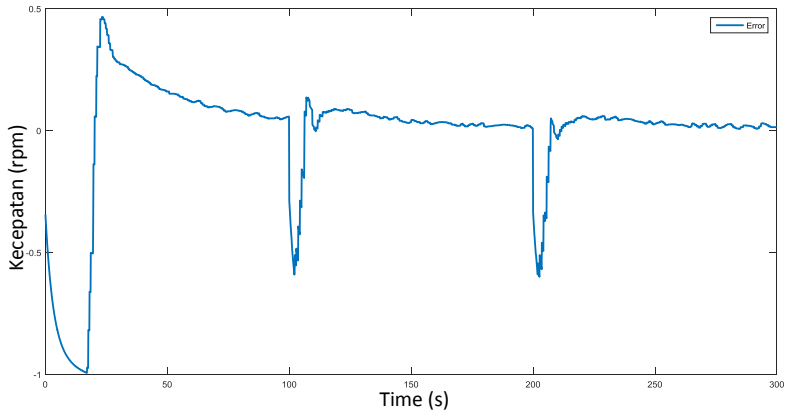
Dengan menggunakan pola perubahan parameter seperti yang sudah ditentukan, maka dapat diperoleh respon *plant* hasil pengujian

menggunakan skema *adaptive PID MRAC basic MIT Rule* dan *normalized MIT Rule* pada *PLC Siemens S7 300* sebagai berikut :

- Pengujian skema kontrol *Adaptive PID MRAC* dengan kondisi tanpa beban

Pada percobaan awal, dilakukan pengujian respon kecepatan *steam turbine tandem compound* yang dikontrol menggunakan skema *adaptive PID MRAC* dengan *basic MIT Rule* maupun *normalized MIT Rule* dengan *input* berupa sinyal *step* yang dibuat berubah-ubah *gain*-nya. Dengan kondisi *plant* yang tidak diberikan pembebanan, diperoleh respon dari *plant steam turbine tandem compound* sesuai dengan grafik yang ditunjukkan oleh Gambar 4.29.



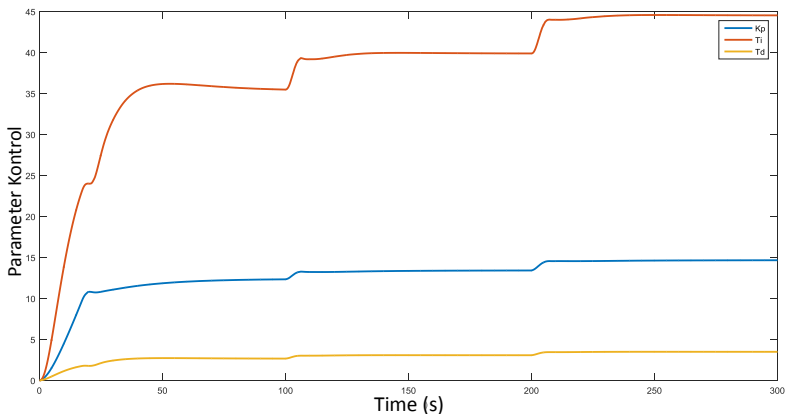


Gambar 4. 29 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300*

Pada grafik tersebut, terlihat bahwa respon awal dari *plant* sempat mengalami *overshoot* yang cukup tinggi, yaitu sekitar 1.479 pada detik ke-23.5. Nilai *overshoot* ini dikarenakan pada saat awal *start*, parameter kontrol masih melakukan proses adaptasi awal dengan *initial condition* sama dengan nol, sehingga respon awal dari *plant* akan cenderung menyimpang dari respon yang dihasilkan oleh *model reference*. Akan tetapi setelah melewati 50 detik, parameter dari kontroler yang diatur pada *adjustment mechanism VI* untuk skema *basic MIT Rule* mulai memiliki nilai yang konvergen sehingga *plant* mulai menghasilkan respon yang sesuai dengan model yang ada. Kondisi ini akan berlanjut sampai detik ke-300 dimana respon *plant* akan semakin mirip dengan respon *model reference* ketika *input step* yang diberikan pada respon diubah-ubah nilai *gain*-nya sehingga parameter kontroler yang ada pada *adjustment mechanism VI* juga akan terus menyesuaikan nilai agar mendapatkan respon yang mirip dengan respon model yang digunakan.

Kemiripan ini secara sederhana dapat dilihat pada *overshoot* yang dihasilkan oleh *plant* setiap perubahan nilai *input step* yang diberikan. Pada perubahan kedua, *overshoot* yang dihasilkan oleh *plant* lebih rendah jika dibandingkan dengan *overshoot* pertama dimana besarnya *overshoot* kedua ini adalah 0.112 pada 124 detik. Kemudian pada saat *gain input* diubah kembali, maka *overshoot* kembali mengalami penurunan menjadi 0.092 pada waktu 233.5 detik. Perubahan parameter

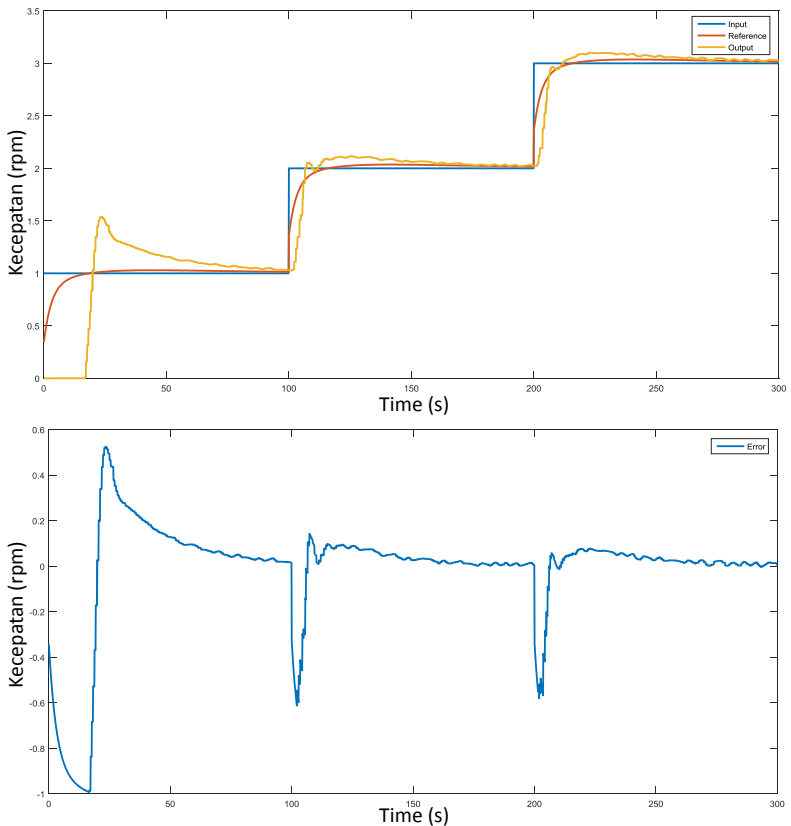
yang mempengaruhi perubahan bentuk respon yang dihasilkan oleh *plant* dengan skema kontrol *PID MRAC basic MIT Rule* ini dapat dilihat pada grafik dalam Gambar 4.30.



Gambar 4. 30 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300*

Dari grafik perubahan parameter kontrol diatas, terlihat bahwa nilai dari tiap-tiap parameter akan cenderung mengalami perubahan ketika terjadi perubahan *gain* pada *input step* yang diberikan pada *plant*. Sedangkan pada saat parameter sistem pada *steam turbine tandem compound* diubah, perubahan yang terjadi pada parameter kontroler cukup kecil dan bahkan cenderung tidak mengalami perubahan. Setelah mengalami perubahan, maka nilai dari tiap parameter kontrol akan *konvergen* kembali. Perubahan inilah yang menyebabkan respon yang dihasilkan oleh *plant* menjadi lebih mirip dengan respon yang dihasilkan oleh *model reference* selama pengujian berlangsung.

Kemudian untuk percobaan selanjutnya, dilakukan pengujian respon kecepatan *steam turbine tandem compound* yang dikontrol dengan skema *adaptive PID MRAC* dengan *normalized MIT Rule* menggunakan *PLC Siemens S7 300*. Dengan menggunakan *input* yang sama seperti pengujian dengan skema *basic MIT Rule*, diperoleh grafik respon yang dihasilkan oleh *plant* seperti yang ditunjukkan oleh Gambar 4.31.

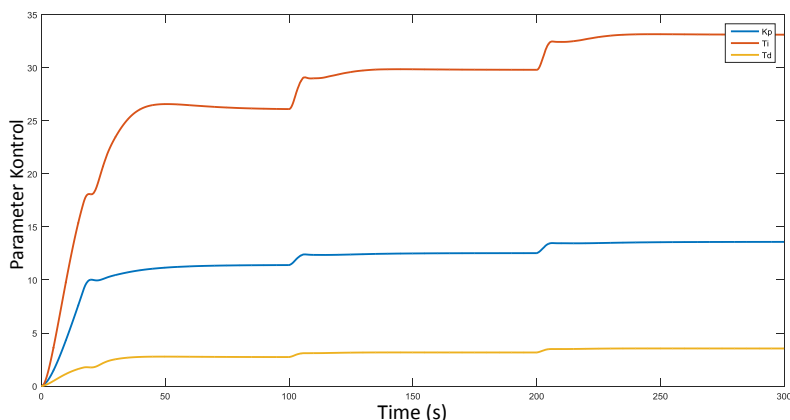


Gambar 4. 31 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300*

Dari gambar tersebut, terlihat bahwa secara garis besar skema kontrol *PID MRAC basic MIT Rule* dan *normalized MIT Rule* menghasilkan bentuk respon *plant* yang hampir sama. Hanya saja untuk pengujian tanpa beban pada *PLC*, nilai *overshoot* yang dimiliki oleh *plant* dengan kontroler *PID MRAC basic MIT Rule* cenderung sedikit lebih kecil jika dibandingkan dengan *normalized MIT Rule*, dimana pada *basic MIT Rule* nilai *overshoot* yang terjadi antara *output plant* dengan *set point* secara berturut-turut adalah 0.479, 0.112, dan 0.092 sedangkan pada

normalized MIT Rule pada kondisi pengujian yang sama memiliki nilai *overshoot* 0.538, 0.116, dan 0.1. Ini mengartikan bahwa respon yang dihasilkan dengan kontroler *adaptive PID MRAC basic MIT Rule* memiliki performansi yang sedikit lebih baik untuk pengujian tanpa beban pada *PLC Siemens S7 300* jika dilihat melalui perbandingan nilai *overshoot* yang dihasilkan antara kedua skema kontrol.

Perbedaan ini dikarenakan nilai parameter kontroler yang diatur oleh *adjustment mechanism VI* yang ada pada *PID MRAC normalized MIT Rule* berbeda dengan nilai parameter kontroler *PID MRAC basic MIT Rule*. Untuk mengetahui secara jelas nilai parameter kontrol yang dihasilkan oleh *adjustment mechanism VI* pada skema kontrol *PID MRAC* dengan *normalized MIT Rule*, dapat diamati pada Gambar 4.32.



Gambar 4. 32 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300*

Sama seperti skema *PID MRAC* dengan *basic MIT Rule*, pada skema *normalized MIT Rule* ini nilai tiap-tiap parameter kontroler akan cenderung mengalami perubahan ketika terdapat perubahan nilai *gain input* yang diberikan, sedangkan ketika parameter sistem *steam turbine* diubah, nilai parameter kontroler cenderung mengalami perubahan yang kecil dan bahkan hampir tidak berubah.

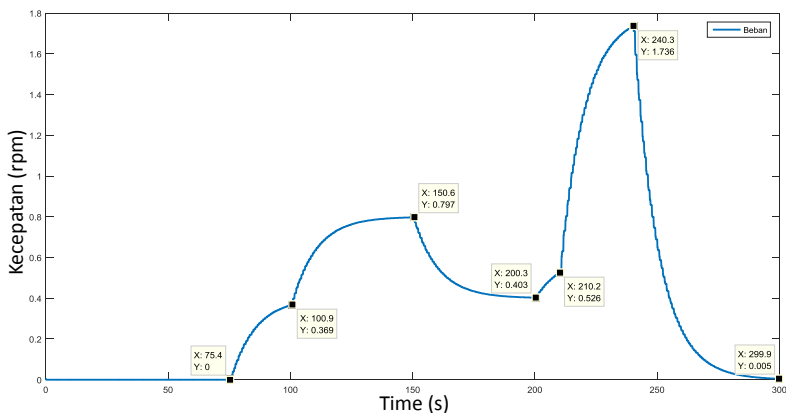
Perubahan nilai parameter kontroler inilah yang menjaga agar respon yang dimiliki oleh *plant* tetap memiliki karakteristik yang mirip dengan respon yang dihasilkan oleh *model reference* walaupun terdapat

gangguan pada sistem yang dalam hal ini berupa perubahan parameter dari sistem.

Pada kondisi pengujian tanpa beban dan parameter sistem yang berubah-ubah, nilai *RMSE* yang didapatkan antara respon *plant* dengan respon *model reference* yang menggunakan skema kontrol *adaptive PID MRAC* dengan *basic MIT Rule* dan *normalized MIT Rule* secara berturut-turut adalah 0.2552 dan 0.2521. Dari nilai *RMSE* ini terlihat bahwa pada pengujian skema kontrol *adaptive PID MRAC* pada PLC *Siemens S7 300* dalam kondisi berbeban, performansi *plant* yang dikontrol menggunakan skema *normalized MIT Rule* cenderung lebih baik dibandingkan dengan skema *basic MIT Rule*.

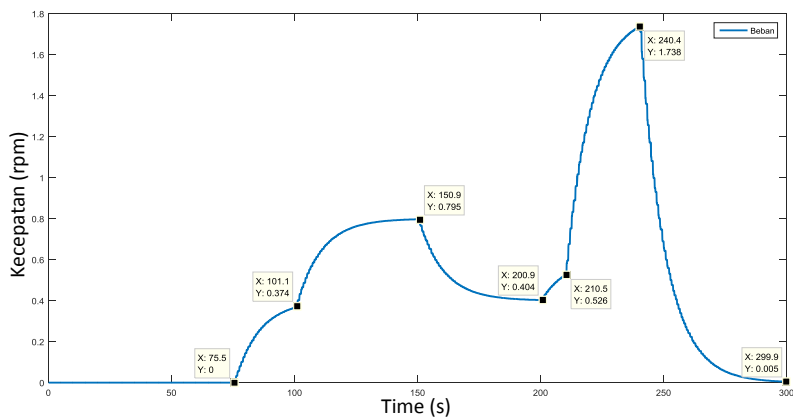
- Pengujian skema kontrol *Adaptive PID MRAC* dengan kondisi berbeban

Untuk pengujian selanjutnya, dilakukan dengan menambahkan pembebanan pada *output* kecepatan *steam turbine tandem compound*. Beban yang digunakan merupakan beban variasi naik dan turun yang diambil langsung dari prosentase *input* yang digunakan disetiap skema kontrol. Pada skema kontrol *adaptive PID MRAC* dengan *basic MIT Rule*, diberikan variasi pembebanan kecepatan yang sesuai dengan Gambar 4.33.



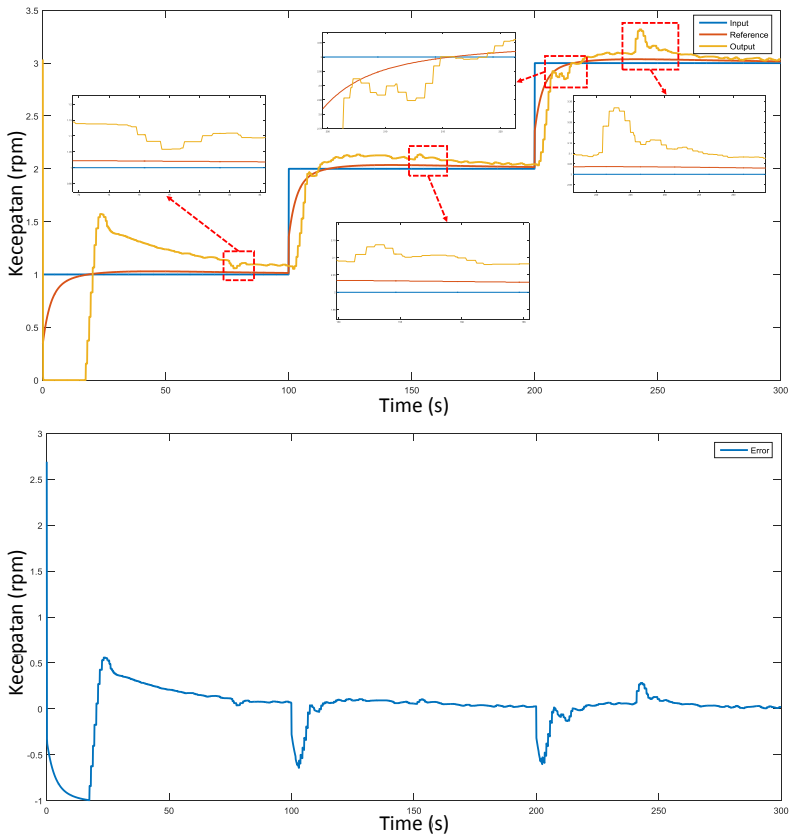
Gambar 4. 33 Variasi beban kecepatan yang diberikan pada pengujian kontroler *PID MRAC* dengan *basic MIT Rule* ketika diimplementasikan pada PLC *Siemens S7 300*

Untuk pembebanan yang diterapkan pada skema kontrol *adaptive PID MRAC normalized MIT Rule* memiliki bentuk yang sama dengan beban yang diberikan pada skema *basic MIT Rule*. Hanya saja yang membedakan adalah waktu mulai beban diterapkan. Untuk lebih jelasnya dapat dilihat grafik pembebanan pada skema *adaptive PID MRAC* dengan *normalized MIT Rule* sesuai pada Gambar 4.34.



Gambar 4. 34 Variasi beban kecepatan yang diberikan pada pengujian kontroler *PID MRAC* dengan *normalized MIT Rule* ketika diimplementasikan pada *PLC Siemens S7 300*

Dengan diberikan gangguan berupa variasi beban sesuai dengan grafik respon beban yang ada, dan juga diterapkannya parameter sistem sesuai dengan Tabel 4.2 yang dapat berubah ketika proses pengujian berlangsung, maka grafik respon yang diberikan oleh *plant* pun memiliki bentuk yang berbeda dengan pengujian yang dilakukan hanya dengan menggunakan gangguan berupa perubahan parameter sistem tanpa adanya pembebanan. Berikut merupakan penjelasan terkait respon *plant* yang dihasilkan oleh tiap-tiap skema kontrol ketika diberi variasi pembebanan dan perubahan parameter sistem. Pada percobaan awal ketika menggunakan skema *basic MIT Rule* dan *adaptation gain* sebesar 3, dapat dilihat pada Gambar 4.35.



Gambar 4. 35 Respon *plant* dan *error* antara *model reference* dan *plant* skema *basic MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300* dalam kondisi berbeban

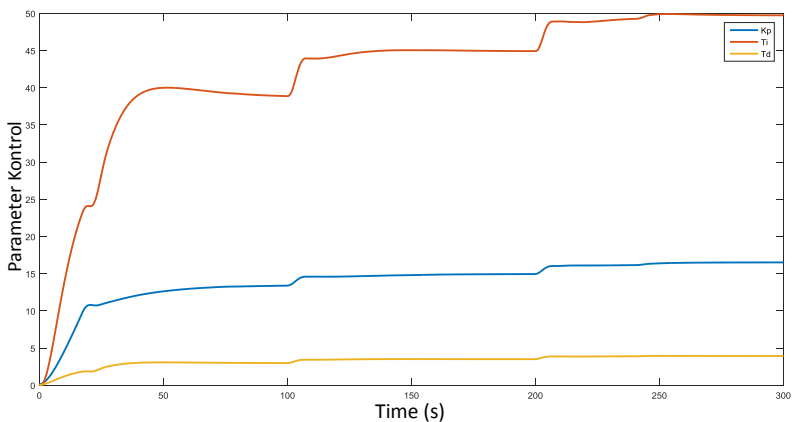
Gambar 4.35 menunjukkan respon *plant* dengan skema kontrol *adaptive PID MRAC basic MIT Rule* ketika diberikan pembebanan kecepatan. Terlihat pada saat diberikan pembebanan awal pada waktu 75.4 detik, respon pada *plant* menunjukkan sedikit penurunan kecepatan hingga menuju 1.058 pada waktu 77.5. Setelah itu, respon kembali menunjukkan kenaikan dan menuju normal kembali saat mencapai 82 detik dengan nilai sebesar 1.109. Kemudian, ketika pembebanan kedua mulai diberikan pada rentang waktu 100.9 detik hingga 150.6 detik,

kontroler berhasil melakukan aksi kontrol yang baik sehingga respon yang dihasilkan oleh *plant* memiliki bentuk yang tidak jauh berbeda dengan respon yang ada pada saat tanpa beban.

Kenaikan kecepatan baru mulai terlihat ketika beban diturunkan hingga detik ke-200.3, kenaikan respon terjadi hingga menuju puncak pada detik ke-153.4 dengan nilai 2.137 dan kemudian segera menuju normal kembali pada detik ke-163. Ketika beban mulai dinaikkan sampai menuju nilai puncak pada rentang waktu 200.3 detik sampai 240.3 detik, respon kembali menunjukkan penurunan nilai pada detik ke-212.4.

Akan tetapi, dengan adanya skema kontrol yang diterapkan maka penurunan yang ada tidak terlalu besar nilainya yakni menjadi bernilai 2.848 dan dapat segera dikembalikan menuju normal pada 218.5 detik. Kondisi ini dapat dipertahankan hingga proses penurunan beban dimulai pada rentang waktu 240.3 detik hingga waktu pengujian selesai. Pada rentang waktu ini, respon *plant* mengalami peningkatan menuju nilai 3.319 pada waktu 240.3 detik dan kembali menuju normal pada detik ke-267.

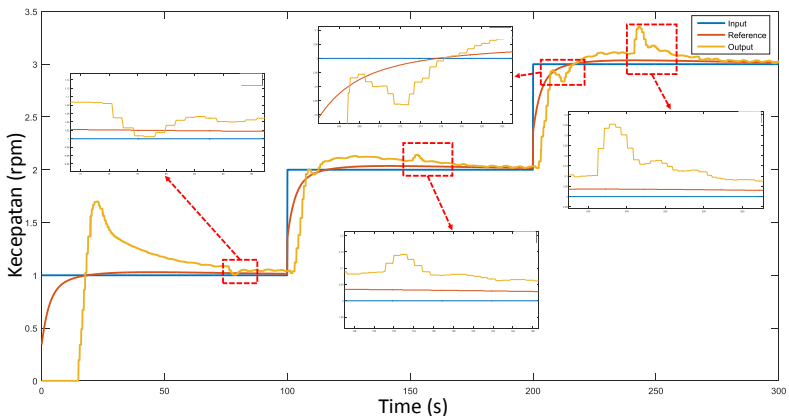
Kondisi ini dapat dicapai berkat kontroler yang selalu melakukan aksi kontrol dan *adjustment mechanism* yang ada pada skema *basic MIT Rule* selalu berusaha menyesuaikan besar parameter kontroler *PID* sesuai dengan *error* yang terjadi antara respon model dan *plant*. Grafik perubahan parameter dari kontroler tiap waktunya ketika proses pengujian berlangsung dengan skema *basic MIT Rule* dapat dilihat pada Gambar 4.36.

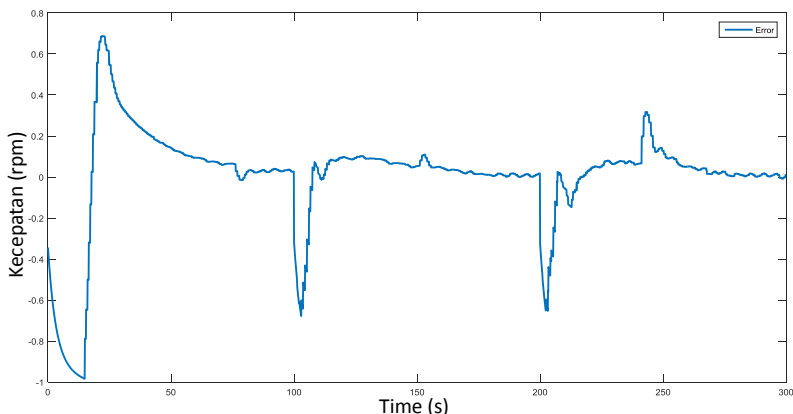


Gambar 4. 36 Grafik perubahan parameter kontroler skema *basic MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada PLC *Siemens S7 300* dalam kondisi berbeban

Dari grafik tersebut, terlihat bahwa parameter kontroler yang diatur menggunakan *adjustment mechanism basic MIT Rule* akan cenderung mengalami perubahan nilai setiap terjadi perubahan *input* maupun pembebanan yang diterapkan. Hal ini karena pada saat terjadi perubahan input maupun pembebanan, maka akan timbul selisih antara respon *model reference* dengan *plant* yang kemudian digunakan oleh *adjustment mechanism* untuk menyesuaikan kembali nilai parameter yang ada.

Kemudian untuk percobaan selanjutnya, dilakukan pengujian respon kecepatan *steam turbine tandem compound* yang dikontrol dengan skema *adaptive PID MRAC* dengan *normalized MIT Rule* menggunakan PLC *Siemens S7 300*. Dengan menggunakan *input* yang sama seperti pengujian awal, dan grafik pembebanan yang sesuai dengan Gambar 4.34, diperoleh grafik respon yang dihasilkan oleh *plant* seperti yang ditunjukkan oleh Gambar 4.37.

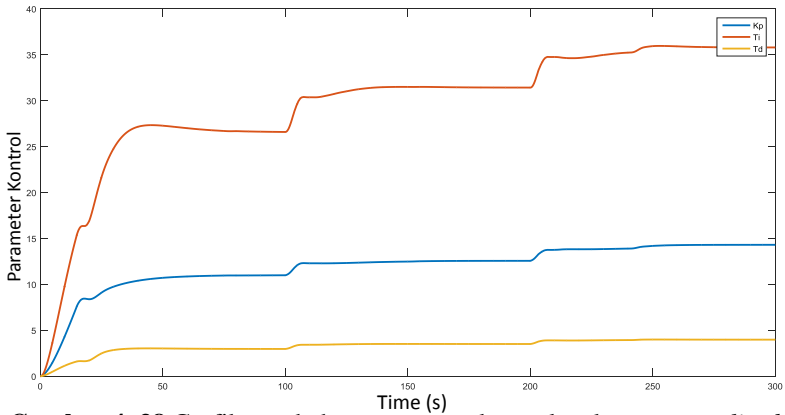




Gambar 4. 37 Respon *plant* dan *error* antara *model reference* dan *plant* skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada *PLC Siemens S7 300* dalam kondisi berbeban

Dari grafik tersebut, terlihat bahwa nilai respon yang dihasilkan dengan skema *normalized MIT Rule* memiliki bentuk yang hampir sama dengan respon yang dihasilkan oleh skema *basic MIT Rule* pada kondisi yang sama. Yang membedakan hanyalah pada skema *normalized MIT Rule*, respon yang dihasilkan cenderung lebih mirip dengan respon *model reference* yang digunakan pada saat pengujian.

Begitu juga dengan perubahan parameter kontrol yang ada pada *normalized MIT Rule*, bentuk grafik yang terbentuk juga memiliki pola yang sama dengan perubahan parameter kontrol yang ada pada *basic MIT Rule* dan dapat diamati bentuknya pada Gambar 4.38.



Gambar 4. 38 Grafik perubahan parameter kontroler skema *normalized MIT Rule* dengan *adaptation gain* = 3 ketika diimplementasikan pada PLC Siemens S7 300 dalam kondisi berbeban

Hanya saja, untuk nilai konvergen pada tiap-tiap parameter yang dihasilkan oleh skema *normalized MIT Rule* cenderung berbeda dengan nilai parameter yang dihasilkan oleh skema *basic MIT Rule*, sehingga hasil dari respon *plant* antara kedua skema ini pada saat pengujian juga memiliki bentuk yang sedikit berbeda.

Untuk pengujian skema kontrol *adaptive PID MRAC* pada kondisi berbeban dan parameter berubah-ubah, didapatkan nilai *RMSE* antara respon *plant* yang dikontrol dengan skema *basic MIT Rule* dan *normalized MIT Rule* secara berturut-turut adalah 0.2747 dan 0.2586. Dikarenakan nilai *RMSE* yang dihasilkan antara respon *plant* yang dikontrol dengan skema yang menggunakan *normalized MIT Rule* lebih kecil dari skema yang menggunakan *basic MIT Rule*, maka untuk pengujian dalam kondisi berbeban dan parameter sistem yang berubah-ubah, *plant* yang dikontrol menggunakan kontroler *adaptive PID MRAC* dengan *normalized MIT Rule* memiliki performansi yang lebih baik dibandingkan dengan skema yang menggunakan *basic MIT Rule*.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN

Bab ini menjelaskan hasil dari pengujian dan simulasi yang telah dilaksanakan yang dinyatakan berupa kesimpulan dan saran. Selama proses penelitian terdapat kendala yang dihadapi. Kendala tersebut telah penulis rangkum menjadi saran sebagai acuan penyempurnaan penelitian lebih lanjut.

1.1. Kesimpulan

Dari penelitian yang telah dilakukan, maka terdapat beberapa kesimpulan yang dapat diambil.

1. Pada pengujian skema kontrol *adaptive PID MRAC* tanpa beban, nilai *overshoot* yang dihasilkan *plant* dengan skema *normalized MIT Rule* cenderung lebih kecil jika dibandingkan dengan *basic MIT Rule*.
2. Pada saat nilai *adaptation gain adaptive PID MRAC* dinaikkan, maka parameter kontroler akan cenderung mengalami konvergensi yang lebih cepat dan *error* yang terjadi antar respon *model* dan *plant* cenderung mengecil akan tetapi apabila nilai *adaptation gain* terlalu besar dapat mengakibatkan sistem osilasi dan menuju kondisi tidak stabil.
3. Pada pengujian skema *adaptive PID MRAC* menggunakan PLC *Siemens S7 300* dengan kondisi tanpa beban, performansi sistem yang menggunakan kontroler dengan skema *normalized MIT Rule* lebih baik dalam melakukan aksi kontrol daripada *basic MIT Rule* dengan nilai *RMSE* 0.2521 untuk *normalized MIT Rule* dan 0.2552 untuk *basic MIT Rule*
4. Pada pengujian skema *adaptive PID MRAC* menggunakan PLC *Siemens S7 300* dengan kondisi berbeban, performansi sistem yang menggunakan kontroler dengan skema *normalized MIT Rule* lebih baik dalam melakukan aksi kontrol daripada *basic MIT Rule* dengan nilai *RMSE* 0.2586 untuk *normalized MIT Rule* dan 0.2747 untuk *basic MIT Rule*
5. Disetiap kondisi pengujian, kontroler *adaptive PID MRAC* dengan skema *normalized MIT Rule* cenderung memberikan *error* yang lebih kecil antara respon *plant* dengan respon *model*

reference dibandingkan dengan skema yang menggunakan *basic MIT Rule*. Akan tetapi pada kondisi tertentu respon yang menggunakan skema *normalized MIT Rule* dapat mencapai kondisi osilasi yang lebih cepat jika dibandingkan dengan skema *basic MIT Rule*.

1.2. **Saran**

Adapun saran dari penelitian ini adalah sebagai berikut :

1. Dalam melakukan pengujian skema kontrol menggunakan *PLC*, dapat digunakan *I/O interface* yang memiliki *sampling time* dan resolusi pencacahan yang lebih baik agar sinyal *digital* yang diproses dan digunakan dalam perhitungan tidak berbeda dengan sinyal *analog* asli yang dicacah.
2. Untuk implementasi pada *real plant*, dapat digunakan *embedded PC* seperti *LabVIEW MyRIO* sebagai tempat penyusunan *adjustment mechanism adaptive PID MRAC*.

DAFTAR PUSTAKA

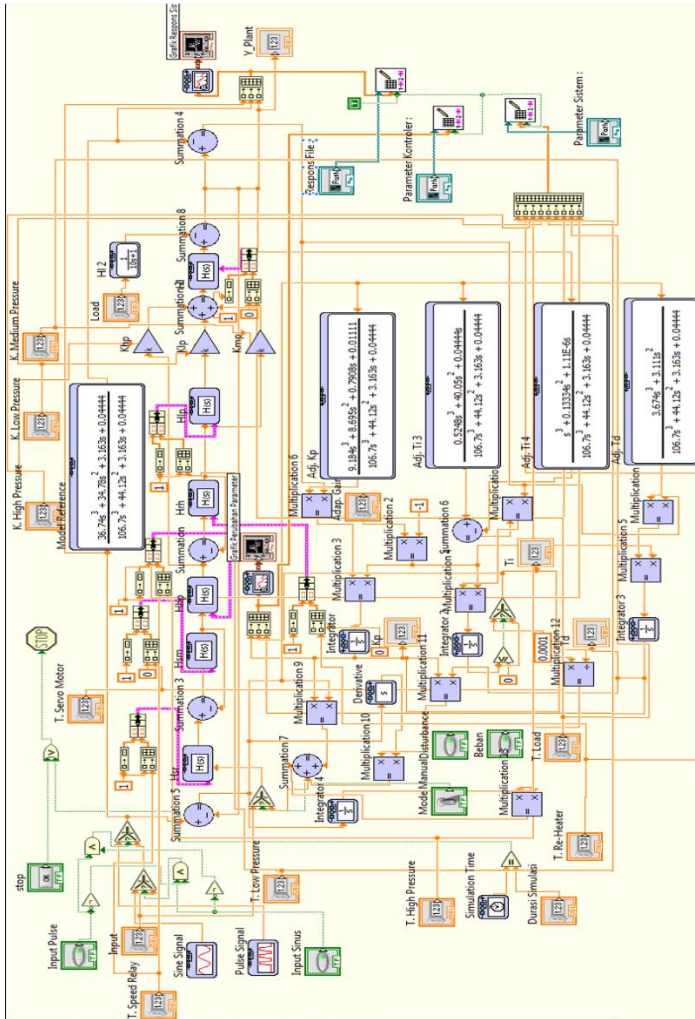
- [1] Chapman. Stephen J, “Electric Machinery Fundamental: Fifth Edition”, Americas, New York, McGraw-Hill, 2012.
- [2] Astrom. Karl J, Wittenmark. Bjorn, “Adaptive Control: Second Edition”, Americas, New York, Mineloa, Dover Publications Inc, 2008.
- [3] Ogata. Katsuhiko, “Modern Control Engineering: Fifth Edition”, Americas, New Jersey, Pearson Prentice Hall, 2010.
- [4] Nise. Norman S, “Control Systems Engineering Sixth Edition”, Americas, Hoboken, River street, John Wiley & Sons Inc, 2017.
- [5] Black, Veatch, “Power Plant Engineering”, Americas, New York, Spring street, Springer, 2005.
- [6] Halliday, Resnick, “Fundamentals of Physics Ninth Edition”, Americas, Alexander Drive, Intrument Society of America, 1995.
- [7] Astrom. Karl J, Hagglund. Tore, “PID Controllers: Theory, Design and Tuning Second Edition”, Americas, New York, Mineloa, Dover Publications Inc, 2008.
- [8] Byan. LA, Bryan. EA, “Programmable Controllers Theory and Implementation Second Edition”, USA, Georgia, Atlanta, Industrial Text Company, 1997.
- [9] Kochummen. Sherin A, Jaffar. Dr. N E, Prof. Nasar A, “Model Reference Adaptive Controller Designs of Steam Turbine Speed Based on MIT Rule”, International Conference on Control, Communication & Computing India, Trivandrum, India, November 9-21 2015.
- [10] Dulau, Mircea. Bica, Dorin, “Mathematical modelling and simulation of the behaviour of the steam turbine”, The 7th

International Conference Interdisciplinarity in Engineering,
University of Tîrgu-Mureş, Romania, October 10-11 2013.

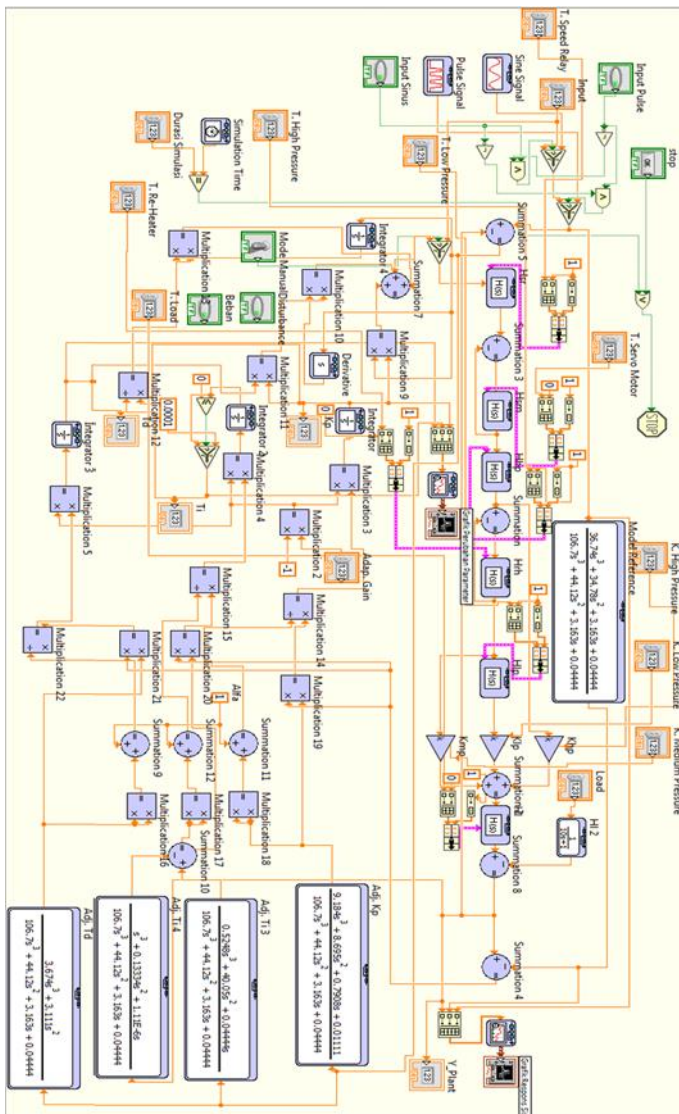
- [11] Dulau, Mircea. Bica, Dorin, “Simulation of speed steam turbine control system”, The 7th International Conference Interdisciplinarity in Engineering, University of Tîrgu-Mureş, Romania, October 10-11 2013.
- [12] Vahidi, Behrooz. Bank Tavakoli, Mohammad Reza. Gawlik, Wolfgang, “Determining Parameters of Turbine’s Model Using Heat Balance Data of Steam Power Unit for Educational Purposes”, IEEE TRANSACTIONS ON POWER SYSTEMS, November 2007.
- [13] SIMATIC Reference Manual, " S7-300 Automation System CPU Specifications : CPU 31xC and CPU 31x", SIEMENS, 2003.
- [14] SIMATIC Manual, " S7-300 Module Data", SIEMENS, 2017.

LAMPIRAN

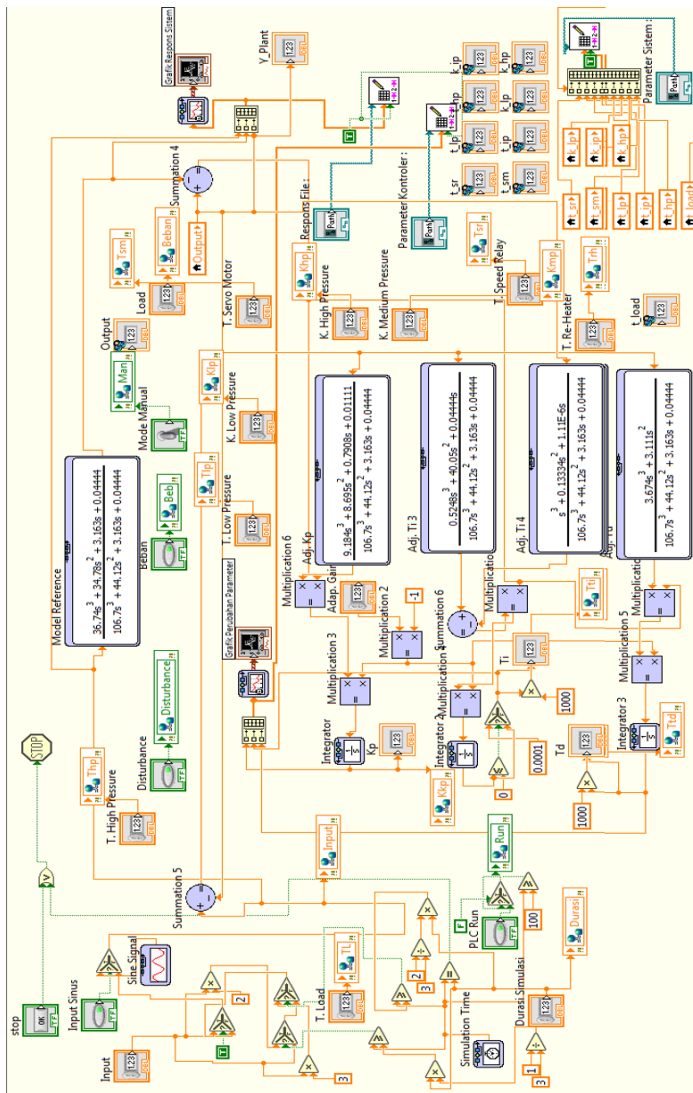
Lampiran 1. Adaptive PID MRAC LabVIEW Basic MIT Rule VI untuk Simulasi tanpa PLC



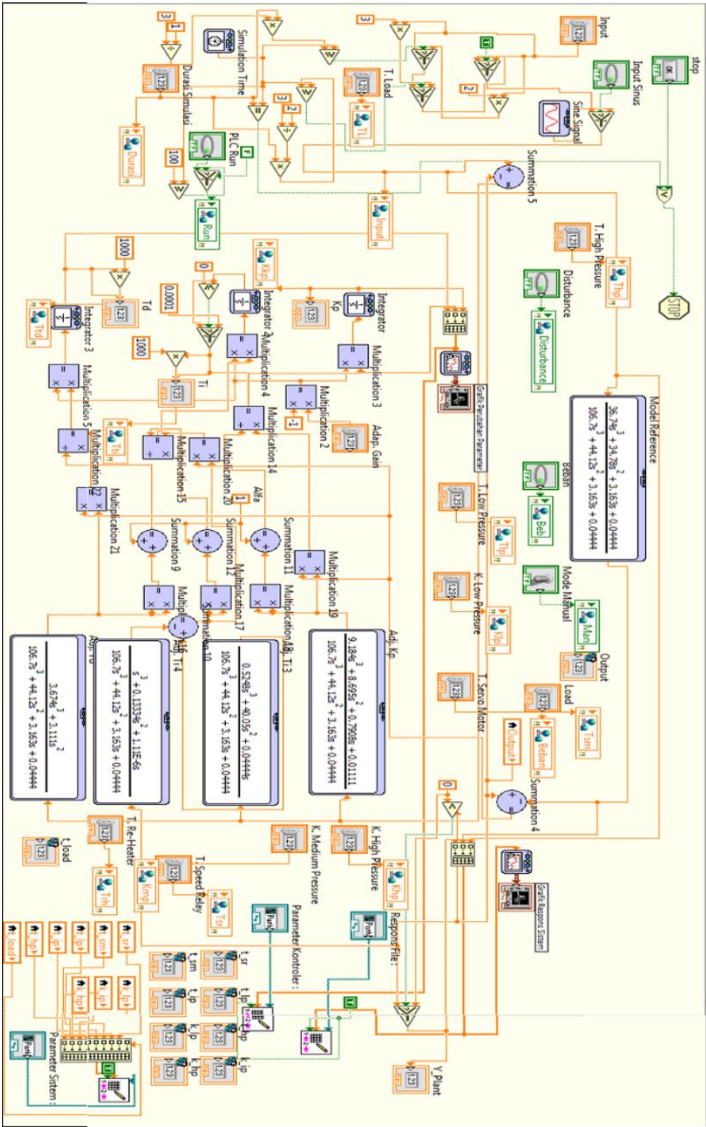
Lampiran 2. Adaptive PID MRAC LabVIEW Normalized MIT Rule VI untuk Simulasi tanpa PLC



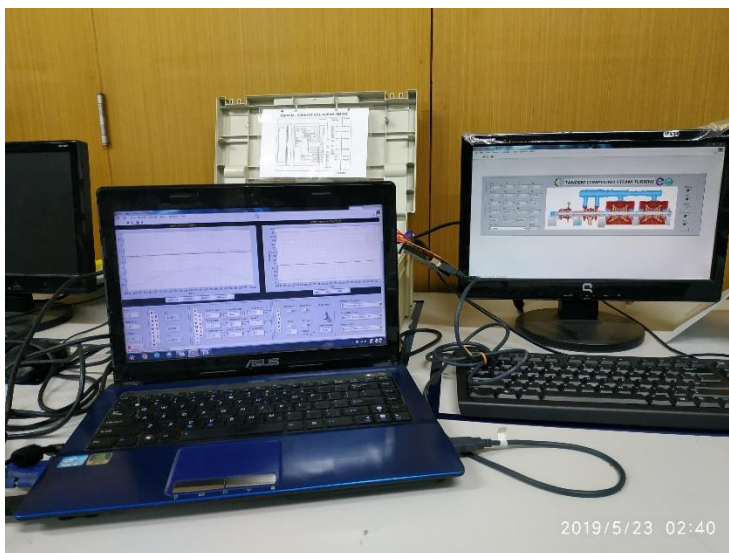
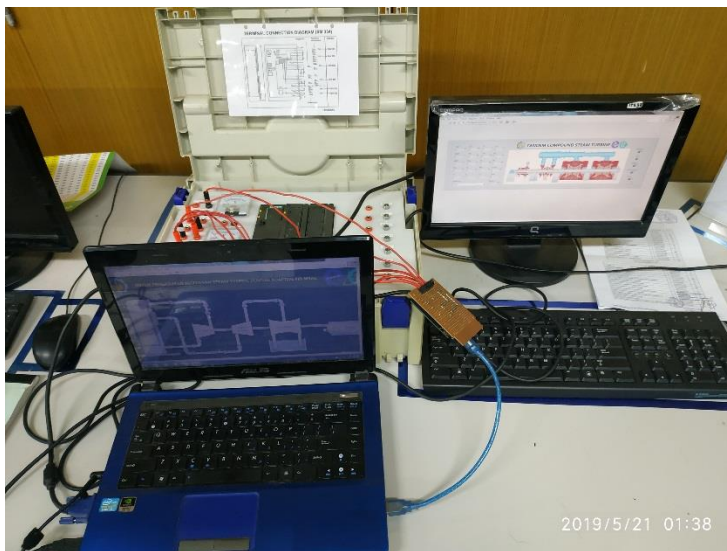
Lampiran 3. Adjustment Mechanism VI Basic MIT Rule untuk Simulasi dengan PLC Siemens S7 300

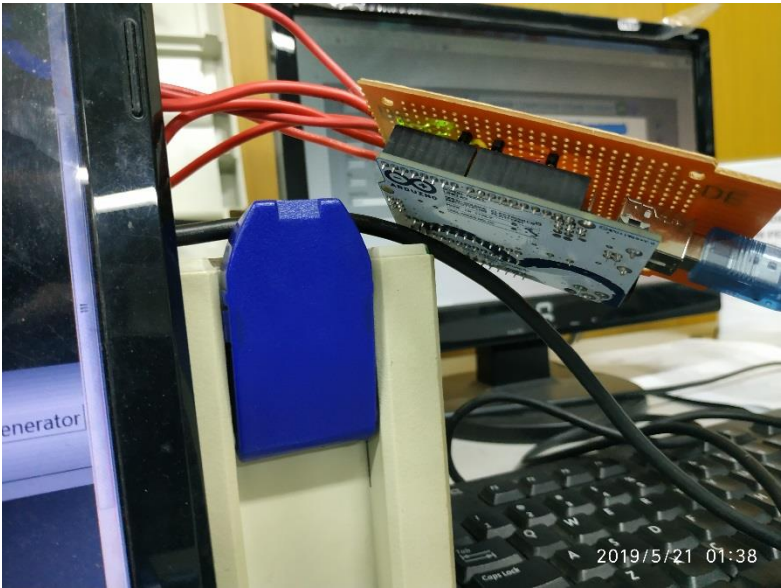
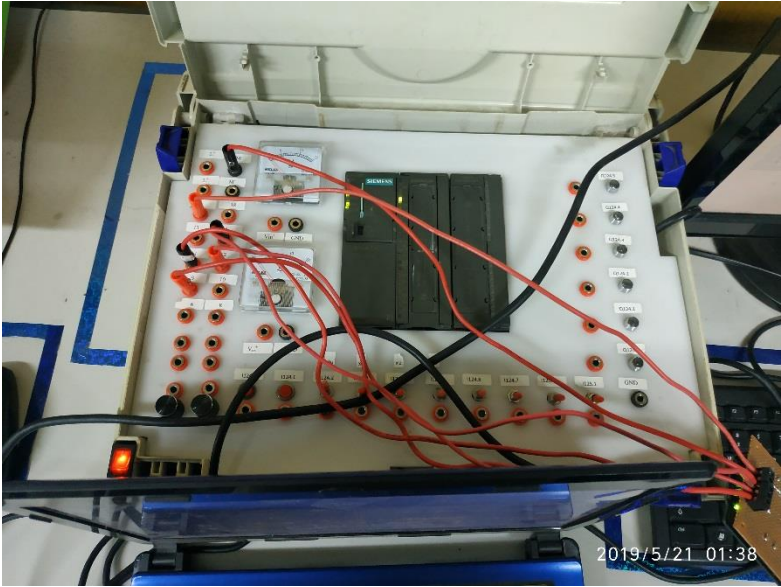


Lampiran 4. Adjustment Mechanism VI Normalized MIT Rule untuk Simulasi dengan PLC Siemens S7 300



Lampiran 6. Perancangan dan Integrasi Sistem Secara Keseluruhan





Lampiran 7. HMI Skema Kontrol Adaptive PID MRAC

Main Control Menu

Grafik Respons Sistem

Input

Adip Gain

V_Pert

STOP

P A K

R O

A N T

M E R

T O

E L

Kp

Ti

Td

P A S

R I

A S

M T

E T

F M

R

T. Low Pressure

T. Re-Heater

T. High Pressure

K. Low Pressure

K. Medium Pressure

K. High Pressure

T. Servo Motor

T. Load

T. Speed Relay

Input Model Plant

O T

H E

R S

Pan. Berubah

Beban (% x Kcc. Pulas)

Input Pulas

Feduresi (Hz)

Ti. Out 2

Duras Simulasi

Input Sinus

Mode Manual

Parameter Kontrol:

Parameter Sistem:

Respons File:

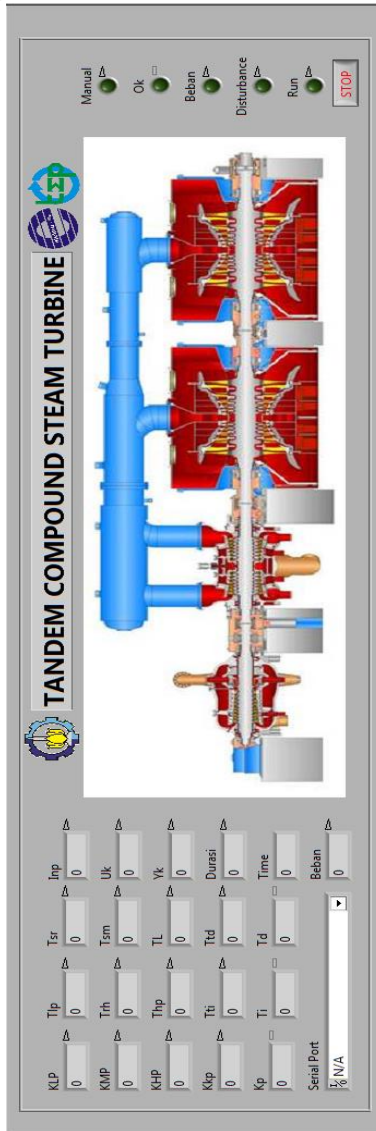
Beban:

Kp Ti Td

Grafik Respons Sistem

Grafik Perubahan Parameter

Lampiran 8. HMI Virtual Plant Steam Turbine Tandem Compound dengan Single Re-Heater



Halaman ini sengaja dikosongkan

RIWAYAT HIDUP PENULIS



Penulis, dengan nama lengkap Mochammad Oza Alfiansyah merupakan mahasiswa tahun angkatan 2015 Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember yang lahir di Kediri pada tanggal 2 April 1997 sebagai putra bungsu dari 3 bersaudara. Riwayat pendidikan penulis adalah SDN Pogar 1 Bangil, SMPN 1 Bangil, dan SMAN 1 Bangil. Penulis pernah menjadi anggota ekskul Robotika semasa SMA dan menjadi anggota tim Line Tracer Mikro hingga pernah menorehkan prestasi bersama timnya diajang kompetisi LTM tingkat SMA di IARC ITS 2014. Semasa kuliah, penulis pernah aktif menjadi anggota tim robot Seni ITS VI-ROSE pada tahun 2016-2017 hingga pada akhirnya penulis memilih untuk aktif di kegiatan Lab Teknik Pengaturan Ruang AJ-104 Teknik Elektro ITS. Penulis ingin menjadi bermanfaat untuk lingkungan masyarakat, ingin membantu perekonomian kedua orangtua, dan ingin menjunjung tinggi derajat kedua orangtua. Penulis ingin memiliki peran aktif di dunia industri dan ingin menghasilkan suatu karya yang bermanfaat bagi perindustrian di Indonesia. Penulis memiliki sifat yang baik, ramah, tetapi terkadang suka tidak peduli pada suatu hal yang tidak penting, bila ingin menghubungi silakan di ozalfiansyah@gmail.com