

**PENALAAN STRUKTUR PI-D MENGGUNAKAN *MODEL*
REFERENCE ADAPTIVE CONTROL (MRAC) PADA
PENGENDALIAN TEGANGAN *OUTPUT* GENERATOR DC 73411**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



DIANA RAMADHANI
NIM. 135060301111049

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2017

LEMBAR PENGESAHAN
PENALAAAN STRUKTUR PI-D MENGGUNAKAN *MODEL REFERENCE*
***ADAPTIVE CONTROL (MRAC)* PADA PENGENDALIAN TEGANGAN OUTPUT**
GENERATOR DC 73411

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



DIANA RAMADHANI

NIM. 135060301111049

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 24 Januari 2017

Dosen Pembimbing I

Ir. Purwanto
NIP. 19540424 198601 1 001

Dosen Pembimbing II

Ir. Mochammad Rusli, Dpli.Ing.
NIP. 19630104 198701 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D
NIP. 19741203 200012 1 001

JUDUL SKRIPSI :

Penalaan Struktur PI-D Menggunakan *Model Reference Adaptive Control (MRAC)* pada Pengendalian Tegangan *Output* Generator DC 73411

Nama Mahasiswa : Diana Ramadhani
NIM : 135060301111049
Program Studi : Teknik Elektro
Konsentrasi : Teknik Kontrol

KOMISI PEMBIMBING :

Ketua : Ir.Purwanto, M.T

Anggota : Ir. Mochammad Rusli. Dipl. Ing.

TIM DOSEN PENGUJI :

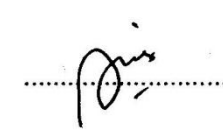
Dosen Penguji 1 : Dr. Ir. Bambang Siswojo, M.T

Dosen Penguji 2 : Rahmadwati, S.T., M.T., Ph.D

Dosen Penguji 3 : Dr. Ir. Erni Yudaningtyas, M.T

Tanggal Ujian : 18 Januari 2017

SK Penguji : No. 83/UN10.F07/SK/2017



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, Januari 2017

Mahasiswa,



Diana Ramadhani

NIM. 135060301111049



UNIVERSITAS BRAWIJAYA

*Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda Tercinta*



PENGANTAR

Bismillahirrohmanirrohim. Alhamdulillah, puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penalaan Struktur PI-D Menggunakan *Model Reference Adaptive Control (MRAC)* pada Pengendalian Tegangan *Output* Generator DC 73411” dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

- Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
- Mama dan Papa tercinta, Bapak Abdul Azis Alkatiri dan Ibu Elly Suhartatik yang selalu memberikan kasih sayang dan do’a yang tak pernah putus.
- Adik tercinta, Silvia Rafidah Alkatiri dan Jihan Safirah Alkatiri yang selalu berbagi cerita, semangat, canda, dan tawa.
- Kakak tercinta, Dedi dan Dodik yang selalu memberikan dukungan.
- Saudara tersayang, yolanda kumala, silviana antika, alya amanda, dan arlin napreliansyah yang tak pernah lelah mendengar cerita suka dan duka.
- Ibu Rossa, Ibu Rahma, Bapak Fery, Bapak Unggul dan Bapak Agus Nelwan yang telah memberikan do’a, dukungan, serta motivasi.
- M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Hadi Suyono, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Ali Mustofa, ST., MT. selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya.
- Dr. Ir. Erni Yudaningtyas, MT. selaku Kepala Laboratorium Sistem Kontrol yang selalu memberikan dukungan dan motivasi.
- Ir. Purwanto, MT. sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan.
- Ir. Mochammad Rusli, Dipl. Ing. sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan.



- Mbak Eka Desiana sebagai pranata laboratorium sistem kontrol yang selalu memberikan dukungan, motivasi, dan berbagi cerita.
- Teman-teman asisten laboratorium sistem kontrol, Mas Rozi, Mas Dimas, Mas Tesu, Mas Emon, Mas Azri, Mas Zai, Mas Mirza, Mbak Suro, Mas Dennis, Mas Andri, Mas Hilmy, Mas Yudha, Mas Faris, Mas Indrut, Mas Avif, Ana, Rony, Yudha N, Zaini, Iqbal, Rifan, Alfin, Alfian, Aldi, Pangky, Vilard, Bobby, Canggih, Iqbal K, dan Yoppy, terimakasih telah memberikan bannyak bantuan dan canda tawa.
- Mbak Zakiyah, Mas Hesa, dan Mas Rozi yang telah memberikan bantuan dan motivasi.
- Teman teman divisi robotika, aeronautika dan awak media, itsna, eki, surya, adit, rony, dan alwin yang banyak memberikan bantuan.
- Teman-teman bermain, Keiko, Ana, Febri, dan Nuke yang selalu memberikan keceriaan.
- Teman-teman otomasi, Rifan, Mukti F, Wicak, Keiko, Fauzan, Kevin, Zaini, Hani, Oliv, Yeilla, Damegi, Reynald, Anwi, Anthony, Izzul, Karil, Gozal, Dhani, Indra dan Yusril.
- Teman-teman Kos Kerto Rahayu 65, Atin, Aji, Maria, Ervi, Diana, Putri, Arnes, Ulfa, dan Indah yang selalu berbagi keceriaan.
- Teman-teman seperjuangan SMA, Atin, Oca, Ardha, Itis, Hani, Janu, Yahya, Yul, Niko, dan Zaid yang saling memberikan motivasi dan semangat.
- Keluarga besar Teknik Kontrol 2013 dan Spectrum 2013 untuk dukungan dan semangat.
- Fakhrrur Rozi untuk kesebaran, bantuan, saran, doa, dan semangat yang tak pernah putus.
- Semua pihak, yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Januari 2017

Penulis

RINGKASAN

Diana Ramadhani, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2017, Penalaan Struktur PI-D Menggunakan *Model Reference Adaptive Control (MRAC)* pada Pengendalian Tegangan *Output* Generator DC 73411, Dosen Pembimbing: Purwanto dan Mochammad Rusli.

Plant motor generator DC 73411 merupakan salah satu panel pelatihan di Laboratorium Sistem Kontrol Universitas Brawijaya Malang. *Plant* motor generator DC 73411 memiliki *steady state gain* sebesar 0,72, faktor redaman sebesar 1,507, dan waktu untuk mencapai *steady state* sebesar 4,12 detik. Pengontrolan dengan struktur PI-D dilakukan agar tegangan *output* generator DC memiliki *error steady state* kurang dari 5%, *settling time* kurang dari 4 detik dan tidak memiliki *overshoot*. Penalaan struktur PI-D dilakukan oleh metode MRAC (*Model Reference Adaptive Control*). Hal ini bertujuan agar sistem lebih tahan terhadap perubahan *setpoint* dan gangguan. *Model Reference Adaptive Control* memiliki ide dasar untuk membuat respon sistem agar menyerupai perilaku respon model referensi.

Kata Kunci: Generator DC, Tegangan *Output*, Struktur PI-D, MRAC.



SUMMARY

Diana Ramadhani, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, January 2016, Tuning of PI-D Structure Using Model Reference Adaptive Control (MRAC) in Control of Output Voltage 73411 DC Generator, Academic Supervisor: Purwanto and Mochammad Rusli.*

73411 DC motor generator plant is one of training panel in Control System Laboratory of Malang Brawijaya University. Steady state gain of 73411 DC motor generator plant is 0.72, damped factor is 1.507, and time to reach steady state is 4.12 seconds. Control with PI-D structure is done so that output voltage of DC generator has error steady state less than 5 %, settling time less than 4 seconds, and does not have overshoot. Tuning of PI-D structure is done by MRAC (Model Reference Adaptive Control) method. It aims to make system more resistant in change of setpoint and disturbance. Model Reference Adaptive Control has basic idea to make system responses in order to have a similar behaviour with reference model responses.

Keywords: DC Generator, Output Voltage, PI-D Structure, MRAC.



DAFTAR ISI

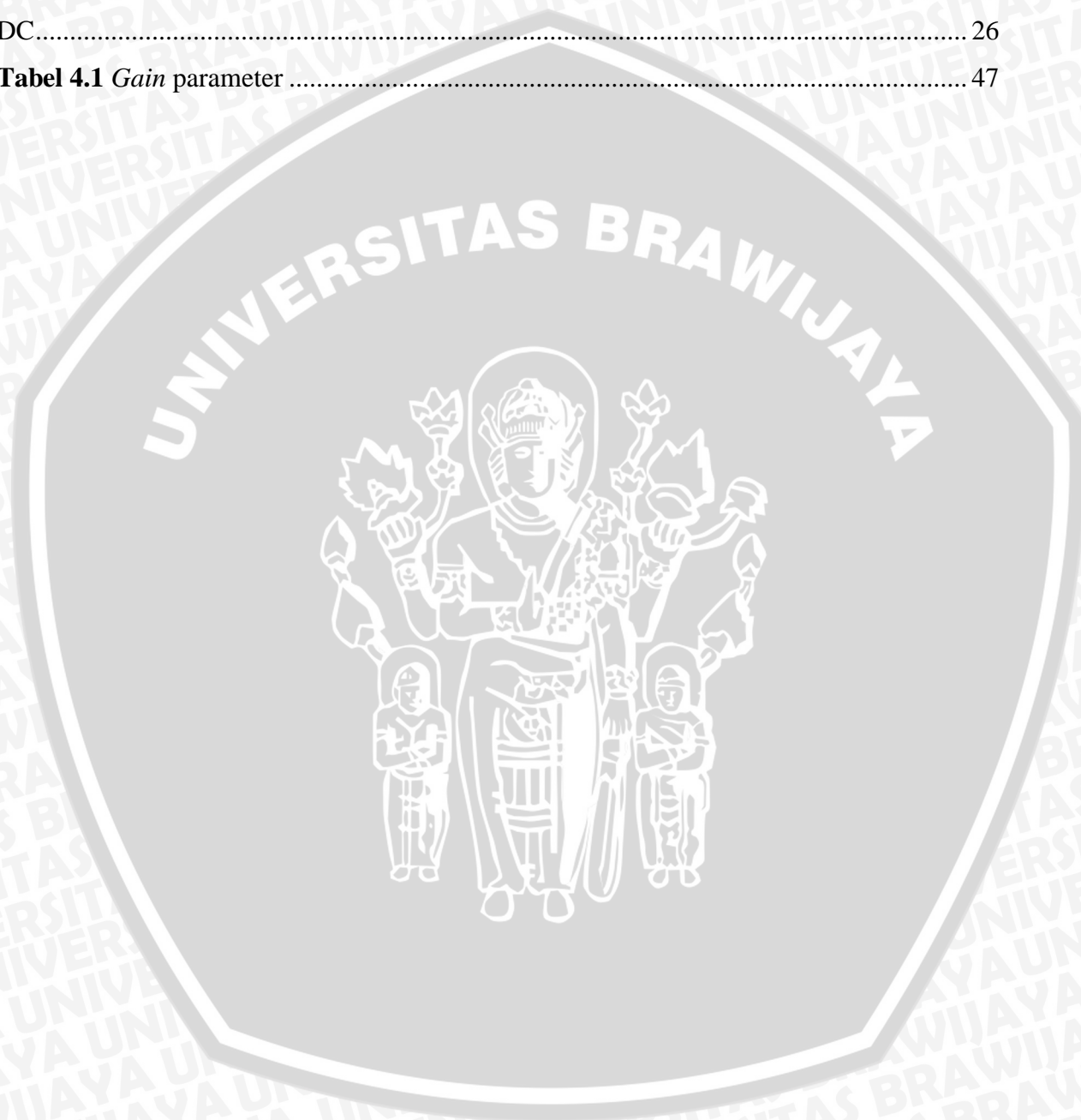
PENGANTAR	i
RINGKASAN	iii
SUMMARY	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Generator <i>Direct Current</i> (DC).....	5
2.2 <i>Plant</i> Motor Generator DC (73411).....	7
2.3 Sensor Tegangan.....	7
2.4 Arduino mega 2560.....	8
2.5 <i>Driver Motor H-Bridge</i> BTS7960.....	9
2.6 Beban Resistif.....	10
2.7 <i>Pulse Width Modulation</i> (PWM).....	11
2.8 Kontroler.....	11
2.9 <i>PID Modified Structure</i>	14
2.10 <i>Model Reference Adaptive Control</i> (MRAC).....	15
2.11 Model Referensi.....	16
2.12 Aturan MIT.....	17



BAB III METODE PENELITIAN.....	19
3.1 Perancangan Blok Diagram Sistem	19
3.2 Spesifikasi Desain.....	20
3.3 Karakterisasi Setiap Blok	21
3.5.1 Karakterisasi <i>Plant</i> (Motor Generator DC 73411).....	21
3.5.2 Karakterisasi <i>Driver Motor H-Bridge</i> BTS7960.....	23
3.5.3 Karakterisasi Sensor Tegangan	25
3.5.4 Penentuan Fungsi Alih Motor Generator DC 73411	27
3.4 Pembuatan Perangkat Keras	29
3.5 Perancangan Algoritma	32
3.5.1 Desain Kontroler PI-D Menggunakan Teknik MRAC.....	32
3.5.2 Penentuan Model Referensi.....	37
3.5.3 Penetapan Parameter Kontroler.....	39
3.5.4 Desain Persamaan Beda	39
3.5.5 <i>Flowchart</i> Program	42
BAB IV HASIL DAN PEMBAHASAN.....	45
4.1 Identifikasi <i>Plant</i>	45
4.2 Simulasi Sistem	47
4.3 Implementasi Sistem.....	52
BAB V KESIMPULAN DAN SARAN.....	59
5.1 Kesimpulan.....	59
5.2 Saran	59
DAFTAR PUSTAKA	61

DAFTAR TABEL

Tabel 2.1 Rangkuman spesifikasi arduino mega 2560	9
Tabel 3.1 Nilai tegangan keluaran terhadap tegangan masukan motor generator DC.....	22
Tabel 3.2 Nilai tegangan keluaran <i>driver</i> terhadap masukan sinyal PWM	24
Tabel 3.3 Nilai tegangan pembacaan sensor terhadap tegangan masukan motor generator DC.....	26
Tabel 4.1 <i>Gain</i> parameter	47



DAFTAR GAMBAR

Gambar 2.1 Konstruksi generator DC	5
Gambar 2.2 <i>Plant</i> motor generator DC.....	7
Gambar 2.3 Sensor tegangan	8
Gambar 2.4 Arduino mega 2560	8
Gambar 2.5 Driver motor H-Bridge BTS7960t.....	9
Gambar 2.6 Beban resistif jenis lampu pijar.....	10
Gambar 2.7 Sinyal PWM.....	11
Gambar 2.8 Diagram blok kontroler proporsional.....	12
Gambar 2.9 Diagram balok kontroler integral.....	12
Gambar 2.10 Diagram blok kontroler diferensial.....	13
Gambar 2.11 Diagram blok kontroler proporsional integral diferensial	14
Gambar 2.12 Struktur PI-D.....	15
Gambar 2.13 Skema <i>Model Reference Adaptive Control</i> (MRAC).....	16
Gambar 3.1 Blok diagram sistem MRAC.....	20
Gambar 3.2 Perubahan tegangan keluaran terhadap tegangan masukan motor generator DC	22
Gambar 3.3 Perubahan tegangan keluaran <i>driver</i> terhadap <i>duty cycle</i>	24
Gambar 3.4 Perubahan tegangan pembacaan sensor terhadap tegangan masukan generator DC	26
Gambar 3.5 Sinyal masukan PRBS dan keluaran motor generator DC 73411	27
Gambar 3.6 <i>System Identification Tool</i>	28
Gambar 3.7 Sinyal <i>output</i> dari estimasi model.....	28
Gambar 3.8 Skema pembuatan perangkat keras.....	29
Gambar 3.9 <i>Power Supply Unit (PSU)</i>	30
Gambar 3.10 Arduino mega 2560	30
Gambar 3.11 Sensor tegangan	30
Gambar 3.12 <i>Driver motor H-Bridge</i> BTS 7960.....	30
Gambar 3.13 Rangkaian beban resistif.....	31
Gambar 3.14 Komputer yang terinstall arduino	31
Gambar 3.15 <i>Plant motor generator DC</i> 73411	31
Gambar 3.16 Diagram blok sistem <i>loop</i> tertutup.....	32
Gambar 3.17 Diagram blok sistem dengan model referensi.....	33

Gambar 3.18	Diagram blok penggunaan teknik MRAC pada penalaan struktur PI-D.....	37
Gambar 3.19	Nilai <i>time constant</i> respon model dengan masukan unit step	38
Gambar 3.20	Nilai <i>settling time</i> respon model dengan masukan unit step	38
Gambar 3.21	Diagram blok sistem dengan desain struktur PI-D menggunakan teknik MRAC	40
Gambar 3.22	<i>Flowchart</i> program utama.....	43
Gambar 3.23	<i>Flowchart</i> sub rutin <i>timer interrupt</i>	44
Gambar 4. 1	Nilai <i>time constant</i> respon <i>plant</i>	45
Gambar 4. 2	Waktu yang dibutuhkan <i>plant</i> mencapai keadaan <i>steady state</i>	46
Gambar 4. 3	Validasi <i>output</i> fungsi alih dengan <i>output</i> motor generator DC 73411	46
Gambar 4.4	Respon simulasi sistem dengan variasi nilai <i>gain</i> parameter.....	48
Gambar 4.5	Perubahan <i>error</i> pada simulasi sistem	48
Gambar 4.6	Perubahan parameter struktur PI-D di Simulasi Sistem.....	49
Gambar 4.7	Perubahan sinyal kontrol di simulasi sistem	49
Gambar 4.8	Respon simulasi sistem dengan <i>setpoint</i> 9 volt.....	50
Gambar 4.9	Respon simulasi sistem dengan <i>setpoint</i> 10 volt.....	50
Gambar 4.10	Respon simulasi sistem dengan <i>setpoint</i> 11 volt.....	51
Gambar 4.11	Respon simulasi sistem dengan perubahan <i>setpoint</i>	51
Gambar 4.12	Respon implementasi sistem dengan <i>setpoint</i> 9 volt.....	52
Gambar 4.13	Parameter struktur PI-D dengan <i>setpoint</i> 9 volt.....	52
Gambar 4.14	Respon implementasi sistem dengan <i>setpoint</i> 10 volt.....	53
Gambar 4.15	Parameter struktur PI-D dengan <i>setpoint</i> 10 volt.....	53
Gambar 4.16	Respon implementasi sistem dengan <i>setpoint</i> 11 volt.....	54
Gambar 4.17	Parameter struktur PI-D dengan <i>setpoint</i> 11 volt	54
Gambar 4.18	Respon implementasi sistem dengan perubahan <i>setpoint</i>	55
Gambar 4.19	Respon implementasi sistem dengan <i>setpoint</i> 9 volt dan diberi gangguan..	55
Gambar 4.20	Parameter struktur PI-D dengan <i>setpoint</i> 9 volt dan diberi gangguan	56
Gambar 4.21	Respon implementasi sistem dengan <i>setpoint</i> 10 volt dan diberi gangguan	56
Gambar 4.22	Parameter struktur PI-D dengan <i>setpoint</i> 10 volt dan diberi gangguan	57
Gambar 4.23	Respon implementasi sistem dengan <i>setpoint</i> 11 volt dan diberi gangguan	57
Gambar 4.24	Parameter Struktur PI-D dengan <i>setpoint</i> 11 volt dan diberi gangguan.....	58





BAB I PENDAHULUAN

1.1 Latar Belakang

Generator DC adalah sebuah perangkat mesin listrik yang mengubah energi mekanis menjadi energi listrik. Prinsip generator dengan sederhana dikatakan bahwa tegangan diinduksikan pada konduktor apabila konduktor digerakkan pada medan magnet sehingga memotong garis gaya. Generator digerakkan oleh beberapa jenis mekanis seperti uap, turbin air, mesin bensin, atau motor listrik (Petruzella, 2001: 312).

Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya memiliki panel pelatihan berupa motor generator DC tipe 73411. Motor Generator DC 73411 terdiri dari generator DC yang dikopel dengan motor DC. *Plant* motor generator DC 734 11 dapat dihubungkan ke beban listrik. Beban listrik ini bisa berupa lampu bohlam (Amalia, 2016: 1).

Pada penelitian sebelumnya (Amalia, 2016) telah dilakukan pengembangan rangkaian dan sistem untuk mengontrol *output* tegangan generator DC. Metode pengontrolan yang digunakan adalah *output feedback control* dengan *pole placement*, pola gangguannya ialah deterministik, dan respon sistem masih memiliki *overshoot*. Oleh karena itu, pada skripsi ini melanjutkan penelitian sebelumnya dengan pola gangguan yang bersifat mendadak, menggunakan metode pengontrolan yang adaptif, serta diharapkan respon sistem tidak memiliki *overshoot*.

Terdapat beberapa klasifikasi pada metode pengontrolan adaptif, salahsatunya adalah *Model Reference Adaptive Control (MRAC)*. Pada teknik MRAC perilaku respon sistem dipaksa untuk menyerupai perilaku respon model referensi. Mekanisme kontrol dilakukan sebagai penalaan parameter untuk mengurangi perbedaan respon sistem dengan respon model referensi. (Bobal, 2005: 12).

Kontroler PID merupakan kontroler yang dapat mengurangi *error steady state*, mempercepat respon transien, mengurangi *overshoot* (Xiong, 2007: 2616). Namun, kontroler PID ideal tidak dapat direalisasikan di laboratorium karena pangkat tertinggi persamaan *Laplace* pembilangnya lebih besar dibandingkan dengan pangkat tertinggi penyebutnya (Rusli, 2015: 115). Salah satu cara yang dapat dilakukan agar kontroler PID dapat digunakan adalah dengan memodifikasi struktur PID menjadi PI-D (Ang dkk, 2007: 561). Dalam skripsi ini akan dibuat suatu desain penalaan struktur PI-D menggunakan teknik *Model Reference Adaptive Control (MRAC)* pada *output* tegangan generator DC 73411.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada poin berikut:

- Bagaimana menentukan fungsi alih *plant* motor generator DC 73411?
- Bagaimana merancang algoritma sistem kontrol tegangan *output* motor generator DC 73411 menggunakan teknik *Model Reference Adaptive Control (MRAC)* pada penalaan struktur PI-D?
- Bagaimana performansi respon sistem setelah diberi kontroler?
- Bagaimana performansi respon sistem jika diberi perubahan nilai *setpoint*?
- Bagaimana performansi respon sistem jika diberi gangguan?

1.3 Batasan Masalah

Mengacu pada rumusan masalah pada skripsi ini, maka penelitian akan dibatasi pada:

- Plant* yang digunakan adalah motor generator DC (73411) di Laboratorium Sistem Kontrol FTUB.
- Algoritma penalaan parameter pada teknik *MRAC* menggunakan pendekatan aturan MIT
- Pola pembebanan dilakukan secara mendadak.
- Pembahasan ditekankan pada penggunaan teknik *MRAC* pada penalaan struktur PI-D.

1.4 Tujuan

Tujuan pemecahan masalah skripsi ini adalah mendesain dan mengaplikasikan teknik *MRAC* pada penalaan kontroler struktur PI-D sebagai kontrol tegangan *output* motor generator DC (73411).

1.5 Manfaat

Manfaat skripsi ini adalah dapat dipahaminya sistem pengontrolan tegangan *output plant* motor generator DC dengan menggunakan teknik *Model Reference Adaptive Control (MRAC)* pada penalaan struktur PI-D.



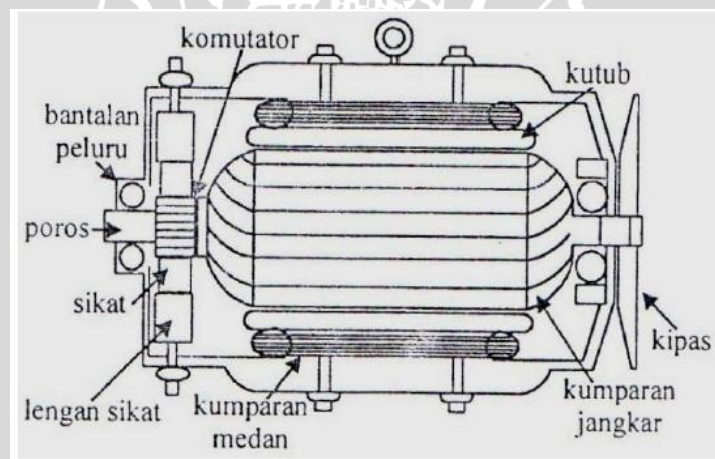


BAB II TINJAUAN PUSTAKA

2.1 Generator *Direct Current* (DC)

Generator *Direct Current* (DC) merupakan sebuah piranti elektronik yang mengubah energi mekanis menjadi energi listrik. Energi mekanis di pergunakan untuk memutar kumparan kawat penghantar di dalam medan magnet. Energi yang dipergunakan bisa berbentuk uap, turbin air, mesin bensin, atau motor listrik.

Generator DC memiliki konstruksi yang terdiri atas dua bagian yaitu bagian yang berputar (rotor) dan bagian yang diam (stator). Contoh dari stator adalah rangka, komponen magnet dan komponen sikat. Sedangkan yang termasuk rotor adalah jangkar, kumparan jangkar dan komutator (Ardian, 2010: 6). Secara umum, konstruksi generator DC dapat dilihat pada Gambar 2.1.



Gambar 2.1 Konstruksi generator DC
Sumber: Ardian, 2010: 7

Prinsip kerja dari suatu generator DC adalah berdasarkan hukum Faraday mengenai induksi elektromagnetik. Hukum Faraday menyebutkan “Apabila suatu konduktor digerakkan dan memotong garis-garis gaya magnetik (fluksi), maka pada konduktor tersebut akan timbul suatu gaya gerak listrik (g.g.l.)” (Soemarwanto, 2010: 79).

Jika kumparan kawat dihubungkan dengan rangkaian listrik tertutup, maka akan timbul pula arus listrik yang mengalir pada rangkaian. Berdasarkan Hukum Induksi Faraday/ Hukum Faraday besarnya g.g.l. yang diinduksikan satu belitan adalah:

$$e = -\frac{d\phi}{dt} 10^{-8} \text{ volt } (\phi \text{ dalam Maxwell}) \quad (2-1)$$

Untuk N belitan maka:

$$e = -N \frac{d\phi}{dt} \times 10^{-8} \text{ volt} \quad (2-2)$$

Apabila kecepatan putara = $\frac{n \text{ putaran}}{\text{menit}}$, maka untuk 1 putaran = $\frac{1}{n}$ menit = $\frac{60}{n}$ detik, konduktor

akan memotong $P \times \phi$ Maxwell, sehingga:

$$e_{rata-rata} = \frac{p \cdot \phi}{60/n} \times 10^{-8} \text{ Volt} = \frac{p \cdot n \cdot \phi}{60} \times 10^{-8} \quad (2-3)$$

Karena ada $\frac{z}{a}$ konduktor pada setiap cabang jangkar paralel, maka:

$$E = \frac{z \cdot p \cdot n \cdot \phi}{a \cdot 60} \times 10^{-8} \text{ Volt} \quad (2-4)$$

$$E = \frac{p \cdot n}{a \cdot 60} \cdot z \cdot \phi \times 10^{-8} \text{ Volt} \quad (2-5)$$

dengan:

P = Jumlah kutub.

a = Jumlah cabang jangkar paralel.

n = Jumlah putaran (putaran/menit).

z = Jumlah konduktor di jangkar.

ϕ = Fluksi tiap kutub (Maxwell).

Di dalam generator DC besar p, a, dan z adalah tertentu/ tetap/ konstan. Maka dari persamaan 2-5, apabila $\frac{p \cdot 1}{a \cdot 60} \cdot z \times 10^{-8} C$ konstan, maka:

$$E = C \cdot n \cdot \phi \text{ Volt} \quad (2-6)$$

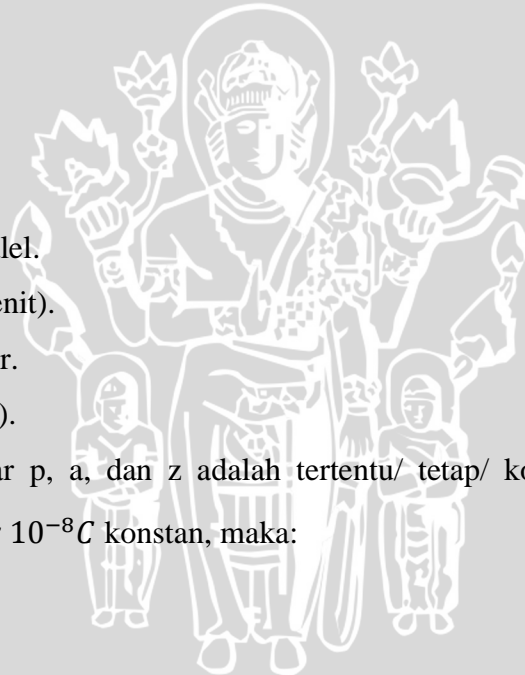
Dengan:

E : Tegangan induksi

C : Konstanta

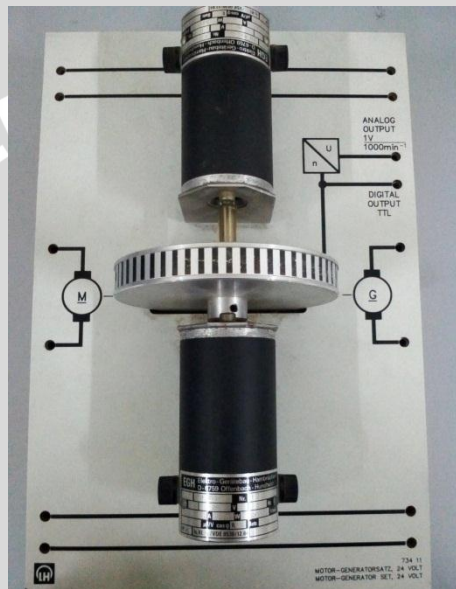
n : Jumlah putaran (putaran/menit).

ϕ : Fluksi tiap kutub (Maxwell).



2.2 Plant Motor Generator DC (73411)

Plant motor generator DC merupakan sebuah panel pelatihan yang dimiliki oleh Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya Malang. *Plant* tersebut terdiri dari generator DC yang dikopel dengan motor DC. Putaran motor DC diubah oleh generator DC menjadi tegangan listrik. Motor DC tersebut mempunyai spesifikasi yaitu catu tegangan sumber maksimal 20 V, arus 0,47 A, daya 3,8 W dan range kecepatan (0–3000) rpm. Bentuk fisik dari *plant* generator DC (73411) dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Plant* motor generator DC

2.3 Sensor Tegangan

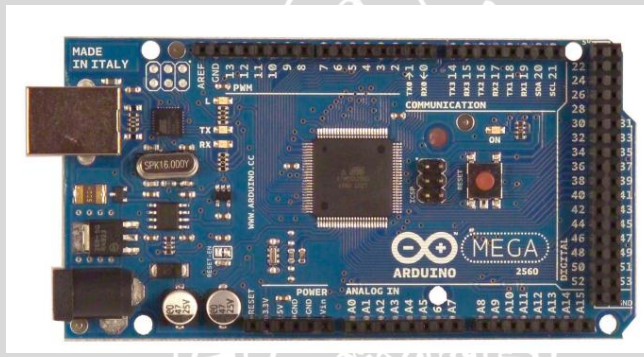
Sensor merupakan sebuah piranti yang berfungsi membaca nilai keluaran sistem. Sensor yang digunakan menggunakan prinsip pembagi tegangan. Piranti tersebut dapat mengurangi tegangan input hingga lima kali dari tegangan aslinya sehingga range tegangan input dari sensor tersebut adalah 0-25 V. Sensor tersebut memiliki resolusi analog sebesar 0.00489V. Dimana, input konektor terminal katoda dihubungkan ke Vcc dan anoda ke Ground (GND). Bentuk fisik dari modul sensor tegangan ada pada Gambar 2.3.



Gambar 2.3 Sensor tegangan
Sumber: <http://instructables.com>

2.4 Arduino mega 2560

Arduino mega 2560 merupakan papan mikrokontroler berdasarkan Atmega328. Papan ini memiliki 54 *pin digital input/output* (14 *pin* dapat digunakan sebagai *output* PWM), 16 *input analog*, 16 MHz osilator kristal, USB koneksi, *jack* listrik, *header* ICSP, dan tombol reset. Bentuk fisik dari arduino mega 2560 dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arduino mega 2560
sumber: <http://microelectronics.com>

Arduino mega 2560 dapat diaktifkan melalui koneksi USB atau dengan tegangan sumber eksternal. Sumber eksternal (non-USB) daya dapat berasal baik dari AC ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2,1 mm konektor daya. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor *power*. Arduino dapat beroperasi dengan tegangan sumber eksternal 6 V sampai 20 V. Namun, jika menggunakan lebih dari 12 V, regulator tegangan dapat panas dan merusak papan. Kisaran yang disarankan adalah 7 V hingga 12 V. Rangkuman spesifikasi dari arduino mega 2560. Rangkuman spesifikasi dari arduino mega 2560 dapat dilihat pada Tabel 2.1.

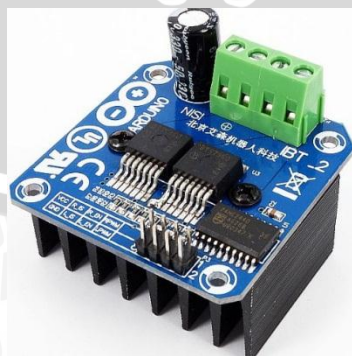
Tabel 2.1 Rangkuman spesifikasi arduino mega 2560
 Sumber: <http://robotshop.com>

<i>Microcontroller</i>	Atmega2560
<i>Operating Voltage</i>	5 V
<i>Input Voltage (recommended)</i>	7-12 V
<i>Input Voltage (limits)</i>	6-20 V
<i>Digital I/O Pins</i>	54 (of which 14 provide PWM output)
<i>Analog Input Pins</i>	16
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3 V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB of which used by bootloader
<i>SRAM</i>	8 KB
<i>Eeprom</i>	4 KB
<i>Clock Speed</i>	16 MHz

2.5 Driver Motor H-Bridge BTS7960

Driver motor berfungsi sebagai pengubah sinyal PWM dari mikrokontroler menjadi tegangan. Dalam aplikasinya, *driver motor* biasanya tersusun dari rangkaian transistor-transistor yang tersusun sedemikian rupa sehingga mampu mengendalikan arah putar dan kecepatan motor berdasarkan arah loop dan tegangan kutub motor.

Driver motor H-Bridge BTS7960 didesain untuk menghasilkan *driver* dua arah gerakan *clockwise* (cw) dan *counterclockwise* (ccw) dengan arus kontinyu sampai dengan 30 A pada tegangan 5 Volt sampai 27 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengontrol. Modul ini mampu *men-drive* (menggerakkan) motor DC yang menjadi penggerak generator. Rangkaian ini dihubungkan ke Arduino mega 2560. Bentuk fisik dari *driver motor H-Bridge BTS7960* adalah seperti pada Gambar 2.5.



Gambar 2.5 Driver motor H-Bridge BTS7960
 Sumber: Driver Motor H-Bridge BTS7960 datasheet

2.6 Beban Resistif

Beban resistif (R) yaitu beban yang terdiri dari komponen tahanan ohm saja (resistance), seperti elemen pemanas (heating element) dan lampu pijar. Beban jenis ini hanya mengkonsumsi beban aktif saja dan mempunyai faktor daya samadengan satu. Sifat beban resistif adalah arus beban resistif sefasa dengan tegangannya (Lanang, 2013: 4). Beban resistif yang dipakai pada penelitian ini adalah berjenis lampu pijar. Bentuk fisik dari beban resistif adalah seperti pada Gambar 2.6.



Gambar 2.6 Beban resistif jenis lampu pijar
Sumber: [http:// creativeelectro.com](http://creativeelectro.com)

Secara matematis nilai resistor (R) adalah:

$$R = \rho \frac{l}{A} \quad (2-7)$$

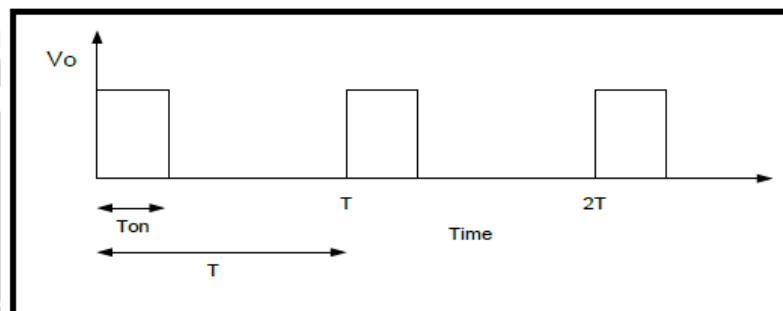
Dengan R = resistor (ohm), ρ = hambatan jenis, l = panjang dari resistor, A = luas penampang. Jika suatu resistor dilewati oleh sebuah arus (I), maka pada kedua ujung resistor tersebut akan timbul beda potensial atau tegangan, yang sesuai dengan Hukum Ohm, yaitu:

$$V_R = I R \quad (2-8)$$

Dengan V_R = tegangan di resistor (Volt), I = arus (Ampere), R = resistor (Ohm). Lampu pijar yang digunakan pada penelitian ini memiliki tegangan maksimal 2,5 volt dan arus maksimal 0,3 A. Lampu pijar yang digunakan disusun secara seri hingga nilai tegangan total beban, minimal sama dengan tegangan keluaran sistem.

2.7 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) adalah salah satu jenis modulasi. Modulasi PWM dilakukan dengan cara mengubah lebar pulsa dari suatu pulsa data. Total 1 periode (T) pulsa dalam PWM adalah tetap, dan data PWM pada umumnya menggunakan perbandingan pulsa positif terhadap total pulsa. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh *output* yang diinginkan. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada *plant*. Sinyal PWM secara umum ditunjukkan pada Gambar 2.7.



Gambar 2.7 Sinyal PWM
Sumber: Ibrahim, 2016: 10

$$\text{Duty Cycle} = \frac{T_{\text{on}}}{T} \times 100\%$$

(2-9)

dimana:

T_{on} = periode logika tinggi

T = periode

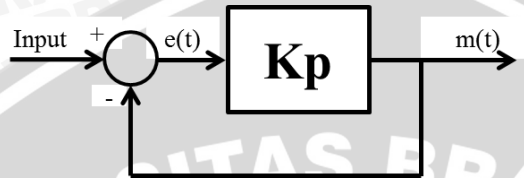
2.8 Kontroler

Kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Hal itu disebabkan oleh tidak dapat dirubahnya komponn penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya. Perubahan perilaku sistem hanya dapat dilakukan melalui penambahan kontroler (Prabarianto, 2016: 14).

Prinsip kerja kontroler adalah membandingkan nilai *output plant* dengan nilai *setpoint*, menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan *error* (Ogata, 2010: 21). Beberapa klasifikasi dari kontroler adalah sebagai berikut:

a. Kontroler Proporsional (P)

Kontroler proporsional memiliki *output* yang besarnya sebanding dengan besarnya sinyal *error*. *Output* kontroler merupakan perkalian antara penguatan proporsional dengan sinyal *error* (Ogata, 2010: 24). Gambar 2.8 menunjukkan diagram blok kontroler proporsional dan persamaan 2-10 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.8 Diagram blok kontroler proporsional

$$m(t) = K_p e(t) \quad (2-10)$$

dimana:

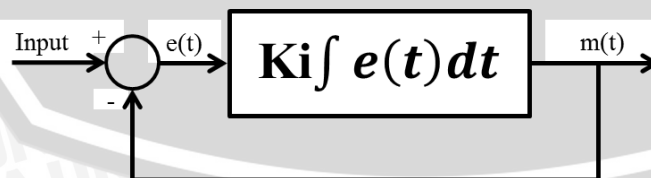
K_p = penguatan proporsional

$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

b. Kontroler Integral (I)

Kontroler integral memiliki karakteristik seperti sebuah operasi integral. *Output* kontroler dipengaruhi oleh perubahan yang sebanding dengan perubahan nilai sinyal *error*. *Output* kontroler merupakan penjumlahan terus menerus dari perubahan sinyal *error* (Ogata, 2010: 24). Gambar 2.9 menunjukkan diagram blok kontroler integral dan persamaan 2-11 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.9 Diagram balok kontroler integral

$$m(t) = K_i \int e(t) dt \quad (2-11)$$

dimana:

K_i = penguatan integral

$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

c. Kontroler Diferensial (D)

Kontroler diferensial memiliki sifat seperti suatu operasi diferensial. Perubahan yang mendadak pada masukan kontroler mengakibatkan perubahan yang sangat besar dan cepat. Kontroler ini tidak akan menghasilkan *output* saat sinyal *error* konstan sehingga tidak akan mempengaruhi keadaan mantap (Ogata, 2010: 25). Gambar 2.10 menunjukkan diagram blok kontroler diferensial dan persamaan 2-12 menunjukkan hubungan antara *output* kontroler dengan sinyal *error*.



Gambar 2.10 Diagram blok kontroler diferensial

$$m(t) = K_d \frac{de(t)}{dt} \quad (2-12)$$

dimana:

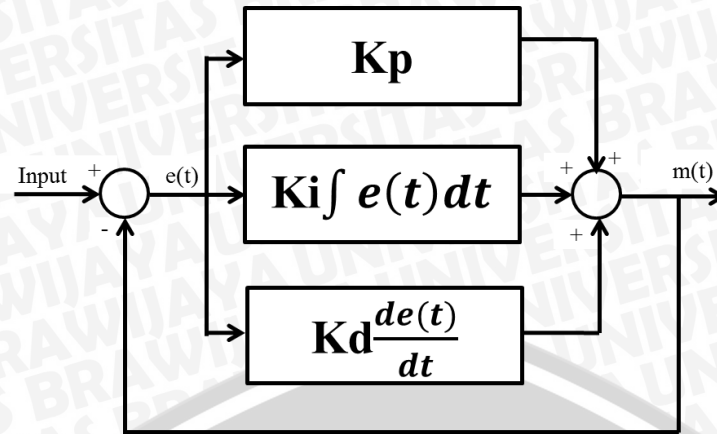
K_d = penguatan derfatif

$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

d. Kontroler Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan derivatif yang terlihat dalam Gambar 2.11 memiliki keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler (Ogata, 2010: 25). Gambar 2.11 menunjukkan diagram blok kontroler Proporsional Integral Diferensial dan persamaan dari kontroler PID dapat dinyatakan sebagai persamaan 2-13.



Gambar 2.11 Diagram blok kontroler proporsional integral diferensial

$$m(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2-13)$$

dimana:

K_p = penguatan proporsional

K_i = penguatan integral

K_d = penguatan derivatif

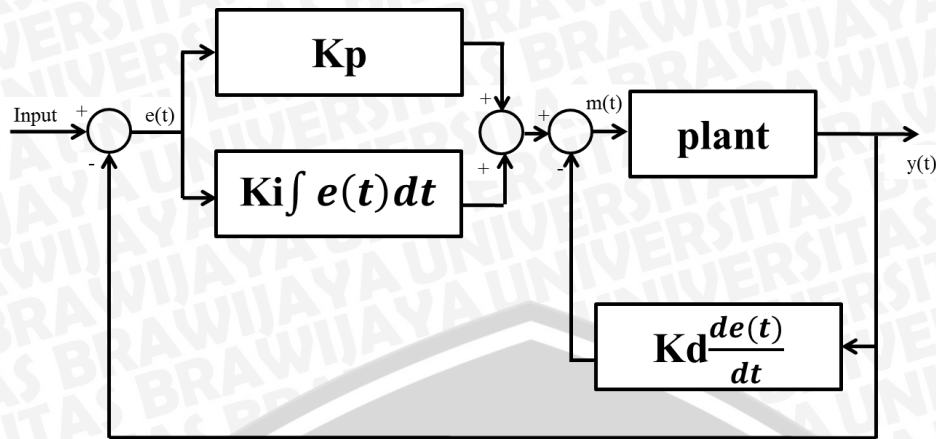
$e(t)$ = sinyal *error*

$m(t)$ = *output* kontroler

2.9 PID Modified Structure

Kontroler PID ideal tidak dapat direalisasikan di laboratorium karena pangkat tertinggi persamaan Laplace pembilangnya lebih besar dibandingkan dengan pangkat tertinggi penyebutnya (Rusli, 2015: 115). Bagian diferensial pada kontroler PID ideal akan mempercepat respon transien melalui kompensasi frekuensi tinggi oleh sebuah diferensiator. Hal yang perlu diperhatikan adalah dengan bertambahnya gain K_d , maka tingkat kestabilan sistem juga akan terpengaruh (Ang dkk, 2007: 560).

Agar kontroler PID dapat direalisasikan, maka terdapat beberapa cara yaitu menambahkan komponen *low pass filter*, menggunakan *PID modified structure*, atau dengan menggunakan *median filter* (Ang dkk, 2007: 561). Dalam penelitian ini digunakan modifikasi struktur PID menjadi struktur PI-D dimana sinyal *error* digantikan dengan variabel proses pada bagian derivatif. Gambar 2.12 menunjukkan diagram blok dari struktur PI-D dan persamaan dari struktur PI-D ditunjukkan oleh persamaan 2-14.



Gambar 2.12 Struktur PI-D

$$m(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{dy(t)}{dt} \quad (2-14)$$

dimana:

K_p = penguatan proporsional

K_i = penguatan integral

K_d = penguatan derifatif

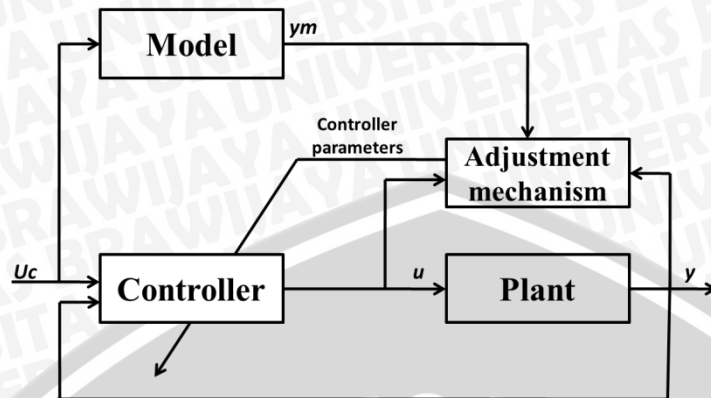
$e(t)$ = sinyal *error*

$y(t)$ = variabel proses

$m(t)$ = *output* kontroler

2.10 Model Reference Adaptive Control (MRAC)

Dasar pemikiran munculnya *Model Reference Adaptive Control* (MRAC) adalah respon sistem yang sesuai dengan keinginan desain dibuat dalam bentuk model $M(s)$. *Output* dari model ini dibandingkan dengan *output* sistem, selisih *output* model dengan *output* sistem digunakan untuk mengatur strategi kontrol (Rozi, 2015: 10). MRAC mempunyai dua lup yaitu lup dalam dan lup luar. Lup dalam terdiri dari *plant* dan kontroler. Lup luar bertujuan untuk menyesuaikan nilai parameter yang digunakan pada lup dalam. Setelah nilai parameter kontroler sudah tepat, maka lup dalam berjalan seperti sistem kontrol pada umumnya. Skema dari *Model Reference Adaptive Control* (MRAC) dapat dilihat pada Gambar 2.13.



Gambar 2.13 Skema Model Reference Adaptive Control (MRAC)
Sumber: Astrom, 1995: 185

Model referensi merupakan sebuah persamaan matematis yang mengandung spesifikasi desain yang diinginkan. Kontroler digunakan untuk menghasilkan sinyal kontrol u yang selanjutnya digunakan sebagai *input plant*. *Plant* menghasilkan sinyal *output* y dan model referensi menghasilkan sinyal *output* y_m . *Output plant* dibandingkan dengan *output* model referensi. Jika terdapat *error* diantara keduanya maka parameter yang ada pada kontroler akan berubah. Perubahan parameter kontroler dilakukan melalui hukum adaptasi yang terjadi pada *adjustment mechanism*. Pada hukum adaptasi, parameter kontroler ini akan diatur sedemikian rupa sehingga parameter kontroler dapat membentuk sinyal kontrol u yang akan menyebabkan *output plant* sesuai dengan model referensi. Apabila *plant* telah mengikuti referensi, maka nilai parameter kontroler sudah tetap seperti yang diharapkan.

2.11 Model Referensi

Model referensi ditentukan melalui derajat relatif sistem. Derajat relatif model referensi sistem harus sama dengan derajat relatif *plant*. Derajat relatif adalah selisih antara derajat polinomial pole dengan derajat polinomial zero. Orde model referensi disesuaikan dengan orde *plant* (Rozi, 2015: 11).

2.12 Aturan MIT

Aturan MIT adalah pendekatan yang sebenarnya pada MRAC. Nama MIT dipakai karena metode tersebut dikembangkan di Laboratorium Instrumentasi di MIT. Untuk mengetengahkan aturan MIT, dimisalkan sebuah sistem loop tertutup di mana kontrolernya mempunyai sebuah parameter θ yang dapat diatur. Respon loop tertutup yang diinginkan ditentukan dengan suatu model yang memiliki *output* y_m . Misalnya e sebagai *error* antara *output* sistem lup tertutup y dan *output* respon mode y_m . Satu kemungkinan untuk mengatur parameter dengan meminimalisasi *loss function* (fungsi kerugian, $J(\theta)$) (Astrom, 1995: 186).

$$J(\theta) = \frac{1}{2} e^2 \quad (2-15)$$

Untuk mendapatkan nilai J yang kecil, merupakan hal yang beralasan untuk mengubah parameter dalam arah negatif gradien dari J , yaitu

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2-16)$$

Turunan parsial $\frac{\partial e}{\partial \theta}$ yang disebut dengan turunan sensitivitas sistem, menunjukkan bagaimana *error* dipengaruhi oleh parameter yang dapat diatur. Jika diasumsikan bahwa parameter berubah lebih lambat daripada variabel yang lain di dalam sistem, sehingga turunan $\frac{\partial e}{\partial \theta}$ dapat dievaluasi di bawah asumsi bahwa θ adalah konstan (Astrom, 1995: 187).

Dengan menggunakan aturan MIT dan mengacu pada Gambar 2.13, respon kontroler merupakan parameter θ yang dikalikan dengan *error* atau selisih antara $y_r(t)$ dan $y(t)$. Hubungan tersebut dapat dilihat pada persamaan 2-17.

$$u(t) = \theta [y_r(t) - y(t)] \quad (2-17)$$

dimana:

$u(t)$ = respon kontroler

θ = parameter kontroler

$y_m(t)$ = respon model referensi

$y(t)$ = respon *plant*

Perubahan parameter kontroler terhadap waktu dinyatakan dalam model pendekatan gradien yang ditunjukkan pada persamaan 2-18 dan 2-19.

$$\frac{d\theta}{dt} = -\gamma \frac{\partial}{\partial \theta} [y(t) - y_m(t)]^2 \quad (2-18)$$

Dengan menurunkan fungsi $[y(t) - y_m(t)]^2$ maka persamaan 2-18 akan menjadi persamaan 2-19.

$$\frac{d\theta}{dt} = -2\gamma[y(t) - y_m(t)] \frac{\partial}{\partial \theta} [y(t) - y_m(t)]$$

(2-19)

dimana:

u(t) = respon kontroler

 θ = parameter kontroler $y_m(t)$ = respon model referensiy(t) = respon *plant* γ = *gain* adaptasi

UNIVERSITAS BRAWIJAYA



BAB III METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Dalam menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan, maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Perancangan blok diagram sistem
2. Spesifikasi desain
3. Karakterisasi setiap blok

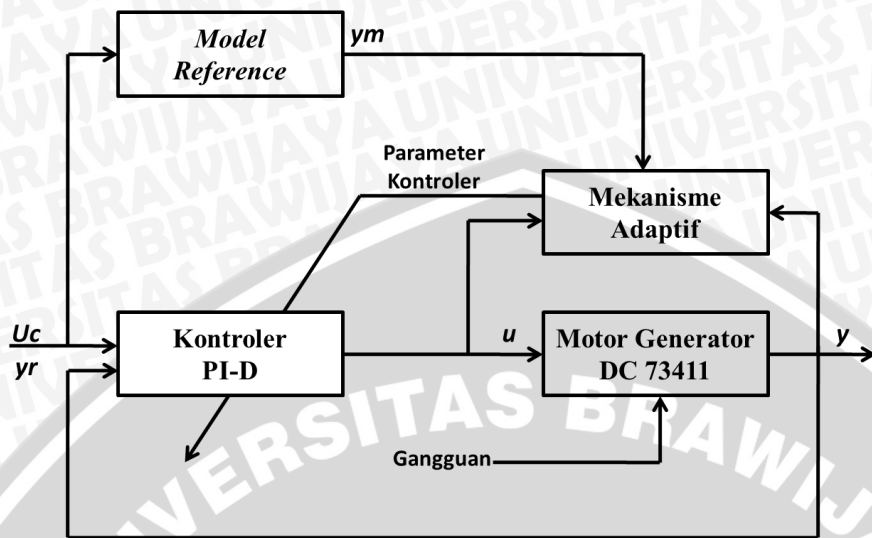
Karakterisasi setiap blok dilakukan untuk mempermudah analisis sistem. Karakterisasi dibagi menjadi beberapa bagian, yaitu:

- a. Karakterisasi *plant* (motor generator DC 73411)
 - b. Karakterisasi *driver motor H-Bridge* BTS7960
 - c. Karakterisasi sensor tegangan
 - d. Penentuan fungsi alih motor generator DC 73411
 - e. Validasi fungsi alih motor generator DC 73411
4. Pembuatan perangkat keras
 5. Perancangan Algoritma

Metode penelitian tersebut dijelaskan sebagai berikut:

3.1 Perancangan Blok Diagram Sistem

Perancangan diagram blok sistem menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan yang didesain. Blok diagram sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok diagram sistem MRAC

Keterangan Gambar:

1. $U_c = y_r$ merupakan *set point* (nilai masukan sistem yang diinginkan pada keluaran sistem).
2. y_m merupakan respon keluaran *model reference*.
3. y merupakan respon keluaran *plant*.
4. u merupakan respon keluaran kontroler PI-D.
5. *Model reference* merupakan persamaan matematis yang akan menjadi model referensi.
6. Mekanisme adaptif merupakan algoritma kontrol adaptif yang akan menghasilkan parameter kontroler yang baru untuk kontroler PI-D.
7. Kontroler PI-D merupakan kontroler yang akan menghasilkan sinyal kontrol untuk *plant*.
8. Motor Generator DC 73411) merupakan *plant* (objek fisik yang dikontrol).
9. Gangguan yang diberikan ke sistem adalah beban resistif berupa lampu bohlam.

3.2 Spesifikasi Desain

Spesifikasi desain pada perancangan pengontrolan tegangan keluaran generator DC adalah sebagai berikut:

1. *Error steady state* kurang dari 5%

Diinginkan *error steady state* kurang dari 5% karena sistem yang baik memiliki *output* dengan batas nilai akhir 5% dari *setpoint*.

2. Rasio peredaman (ξ) sama dengan 1

Diinginkan rasio peredaman (ξ) sama dengan 1 agar *output* tidak berosilasi dan dapat mencapai *setpoint*.

3. *Settling time* kurang dari 4 detik

Diinginkan *settling time* kurang dari 4 detik agar nilai *settling time* respon sistem lebih cepat dari sebelum diberi kontroler.

4. Tidak memiliki *overshoot*

Diinginkan respon sistem yang tidak memiliki *overshoot* agar rangkaian elektrik motor generator dc tidak rusak ketika terjadi lonjakan tegangan.

3.3 Karakterisasi Setiap Blok

3.5.1 Karakterisasi *Plant* (Motor Generator DC 73411)

Karakterisasi motor generator DC dilakukan untuk mengetahui karakter atau *gain* motor generator DC. Hal tersebut diperoleh dengan mengamati tegangan keluaran terhadap tegangan masukan generator DC. Adapun peralatan yang digunakan saat melakukan karakterisasi *plant* adalah sebagai berikut:

1. *Power Supply Unit* (PSU),
2. Motor Generator DC 73411,
3. Multimeter,
4. Perangkat komputer, dan
5. Kabel penghubung.

Adapun langkah langkah saat melakukan karakterisasi *plant* adalah sebagai berikut:

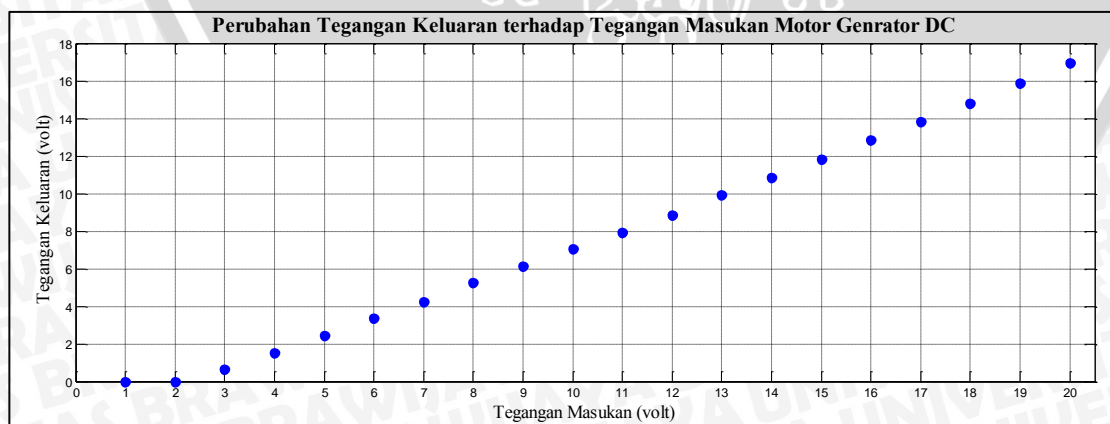
1. Tegangan *output Power Supply Unit* dihubungkan ke masukan generator DC.
2. Tegangan *output Power Supply Unit* diatur dari 0 volt hingga 20 volt dengan kenaikan setiap 1 volt.
3. Tegangan keluaran motor generator DC diukur menggunakan multimeter
4. Tegangan keluaran motor generator DC dicatat dan diamati.

Nilai tegangan keluaran motor generator DC dapat dilihat pada Tabel 3.1.

Tabel 3.1 Nilai tegangan keluaran terhadap tegangan masukan motor generator DC

Tegangan Masukan Motor Generator DC (volt)	Tegangan Keluaran Motor Generator DC (volt)
0	0
1	0
2	0,65
3	1,51
4	2,45
5	3,35
6	4,25
7	5,25
8	6,11
9	7,07
10	7,9
11	8,85
12	9,92
13	10,83
14	11,83
15	12,85
16	13,8
17	14,8
18	15,88
19	16,94
20	17,96

Hubungan tegangan keluaran dengan tegangan masukan motor generator DC pada Tabel 3.1 dapat direpresentasikan menjadi grafik seperti pada Gambar 3.2.



Gambar 3.2 Perubahan tegangan keluaran terhadap tegangan masukan motor generator DC

Dari grafik pada Gambar 3.2 didapatkan *gain* motor generator DC sebagai berikut:

$$m = \frac{y_{19} - y_{14}}{x_{19} - x_{14}} = \frac{16,94 - 11,83}{19 - 14} = \frac{5,11}{5} = 1,022$$

3.5.2 Karakterisasi *Driver Motor H-Bridge* BTS7960

Karakterisasi *driver* dilakukan untuk mengetahui karakteristik, kinerja dan *output* dari *driver motor H-Bridge* BTS7960. Hal ini dilakukan dengan cara membandingkan tegangan *output* efektif *driver* dengan masukan *duty cycle* sinyal PWM yang diberikan oleh arduino mega 2560. Adapun peralatan yang digunakan saat melakukan karakterisasi *plant* adalah sebagai berikut:

1. *Power Supply Unit* (PSU),
2. Perangkat komputer,
3. *Driver motor H-Bridge* BTS7960,
4. Multimeter,
5. Arduino mega 2560, dan
6. Kabel penghubung.

Adapun langkah langkah saat melakukan karakterisasi *plant* adalah sebagai berikut:

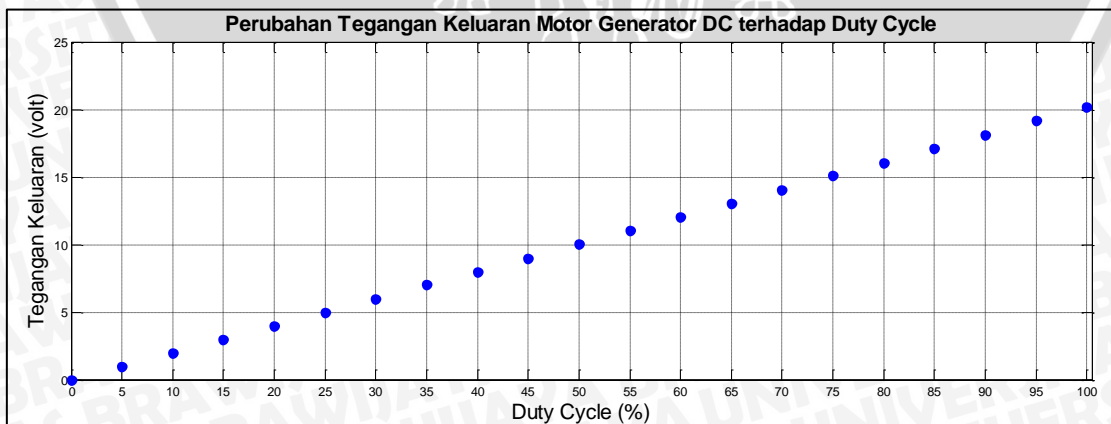
1. Tegangan keluaran *Power Supply Unit* dihubungkan ke masukan tegangan referensi *driver*.
2. Pin keluaran PWM di arduino mega 2560 dihubungkan ke pin masukan tegangan *driver*.
3. Tegangan keluaran *driver* dihubungkan ke multimeter.
4. *Duty cycle* pada sinyal PWM diatur mulai 0% dingga 100% dengan kenaikan setiap 5 % untuk membangkitkan tegangan keluaran *driver*.
5. Nilai tegangan yang terbaca oleh multimeter dicatat dan diamati.

Nilai tegangan keluaran *driver motor H-Bridge* BTS7960 dapat dilihat pada Tabel 3.2.

Tabel 3.2 Nilai tegangan keluaran *driver* terhadap masukan sinyal PWM

Duty Cycle (%)	Tegangan Keluaran Driver dengan multimeter (volt)
0	0
5	0,94
10	1,97
15	2,99
20	3,94
25	4,97
30	5,99
35	7,02
40	7,97
45	9,00
50	10,03
55	11,06
60	12,01
65	13,04
70	14,07
75	15,11
80	16,06
85	17,09
90	18,13
95	19,16
100	20,20

Hubungan tegangan keluaran driver dengan masukan driver yang berupa sinyal PWM pada Tabel 3.2 dapat direpresentasikan menjadi grafik seperti pada Gambar 3.3.



Gambar 3.3 Perubahan tegangan keluaran *driver* terhadap *duty cycle*

Dari grafik pada Gambar 3.3 didapatkan *gain driver motor H-Bridge* BTS7960 sebagai berikut:

$$m = \frac{y_{85} - y_{70}}{x_{85} - x_{70}} = \frac{17,09 - 14,07}{85 - 70} = \frac{2,01}{10} = 0,201 \frac{\text{volt}}{\%}$$

3.5.3 Karakterisasi Sensor Tegangan

Karakterisasi sensor tegangan dilakukan untuk mengetahui tingkat kelinieran dalam pembacaan tegangan keluaran motor generator DC. Adapun peralatan yang digunakan saat melakukan karakterisasi *plant* adalah sebagai berikut:

1. *Power Supply Unit* (PSU),
2. Sensor tegangan,
3. Motor generator DC 73411,
4. Multimeter,
5. Arduino mega 2560,
6. Perangkat komputer, dan
7. Kabel penghubung.

Adapun langkah langkah saat melakukan karakterisasi *plant* adalah sebagai berikut:

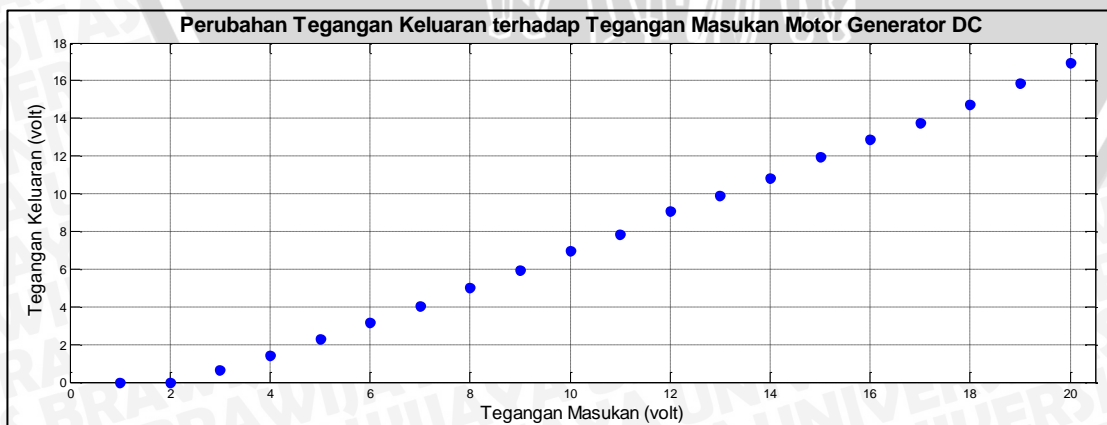
1. Tegangan keluaran *Power Supply Unit* dihubungkan ke masukan motor generator DC.
2. Tegangan keluaran motor generator DC dihubungkan ke sensor tegangan.
3. Sensor tegangan dihubungkan ke pin pembacaan analoag di arduino mega 2560.
4. Tegangan keluaran *Power Supply Unit* diatur dari 0 volt hingga 20 volt dengan kenaikan sebesar 1 volt.
5. Nilai tegangan yang terbaca oleh sensor dicatat dan diamati.

Nilai tegangan keluaran yang telah dibaca sensor dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai tegangan pembacaan sensor terhadap tegangan masukan motor generator DC

Tegangan Masukan Motor Generator DC (volt)	Hasil pembacaan sensor (volt)
0	0
1	0
2	0,57
3	1,38
4	2,24
5	3,16
6	4,04
7	5,02
8	5,97
9	6,95
10	7,94
11	9,20
12	10,01
13	10,95
14	12,03
15	12,97
16	13,85
17	14,94
18	16,02
19	17,13
20	18,14

Hubungan tegangan keluaran yang dibaca sensor dengan tegangan keluaran motor generator DC dapat direpresentasikan menjadi grafik seperti pada Gambar 3.4.



Gambar 3.4 Perubahan tegangan pembacaan sensor terhadap tegangan masukan generator DC

Dari grafik pada Gambar 3.4 didapatkan *gain* sensor tegangan sebagai berikut:

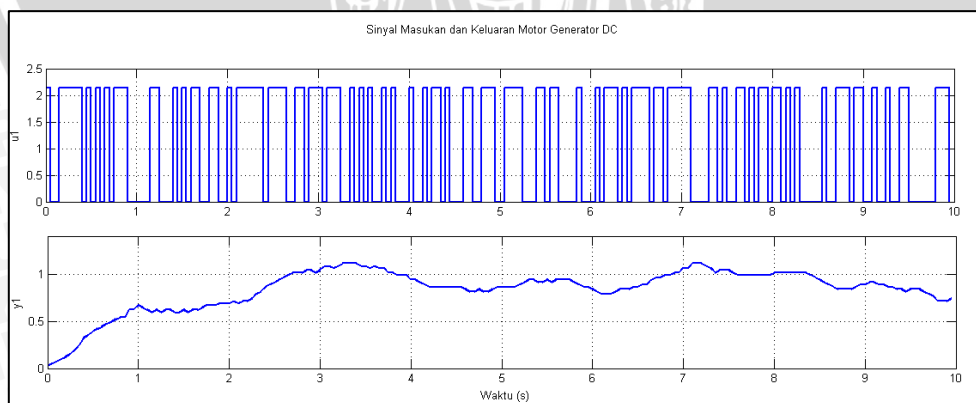
$$m = \frac{y_{10} - y_9}{x_{10} - x_9} = \frac{7,94 - 6,95}{10 - 9} = \frac{0,99}{1} = 0,99$$

3.5.4 Penentuan Fungsi Alih Motor Generator DC 73411

Pengontrolan tegangan keluaran motor generator DC 73411 menggunakan arduino mega 2560 sebagai pengolah dan memberikan data berupa *Pulse Width Modulation* (PWM) agar motor bergerak. Motor generator yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan sensor tegangan. Karakteristik motor generator DC 734 11 pada perancangan ini didapatkan dengan cara memberikan input unit step.

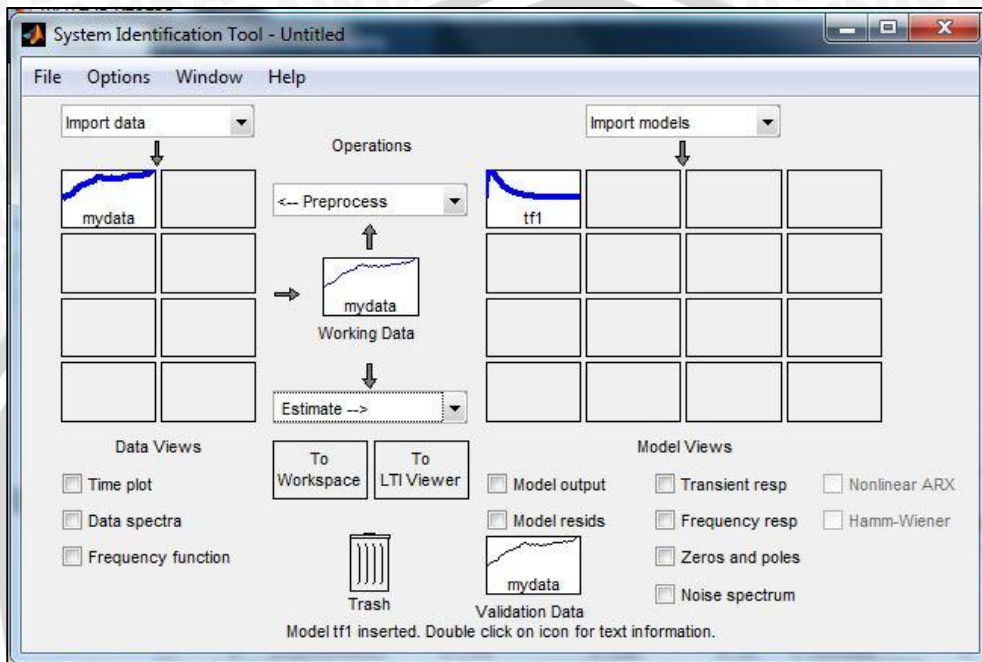
Fungsi alih *plant* motor generator didapatkan dari pemodelan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Langkah yang dilakukan untuk membangkitkan sinyal PRBS hingga mendapatkan fungsi alih adalah sebagai berikut:

1. Nilai linier dicari dari hasil tegangan *output* terhadap duty cycle PWM.
2. Nilai batas atas dan bawah dimasukkan berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai input motor DC.
4. Setelah didapatkan data sinyal PRBS dan data tegangan *output* motor generator 734 11 yang ditunjukkan pada Gambar 3.5, selanjutnya identifikasi dilakukan dengan menggunakan software Matlab.

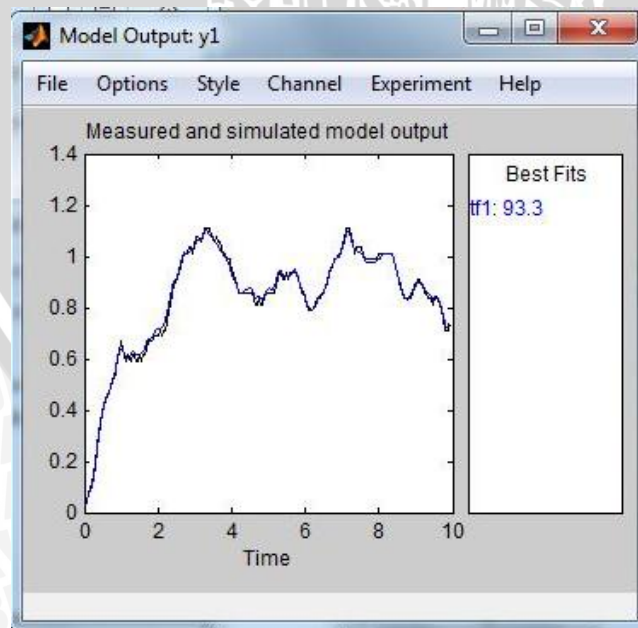


Gambar 3.5 Sinyal masukan PRBS dan keluaran motor generator DC 73411

5. Dengan menggunakan sintaks `ident` pada *command window* pada Matlab, sinyal PRBS dan tangan keluaran generator yang telah disimpan kemudian di *import* pada blok *System Identification Toolbox* yang ditunjukkan pada Gambar 3.6. Setelah dilakukan beberapa estimasi model berdasarkan nilai yang telah di *import*, maka didapatkan fungsi alih *plant* motor generator DC 73411 dengan best fit sebesar 93.33% yang ditunjukkan pada Gambar 3.7.



Gambar 3.6 *System Identification Tool*



Gambar 3.7 Sinyal *output* dari estimasi model

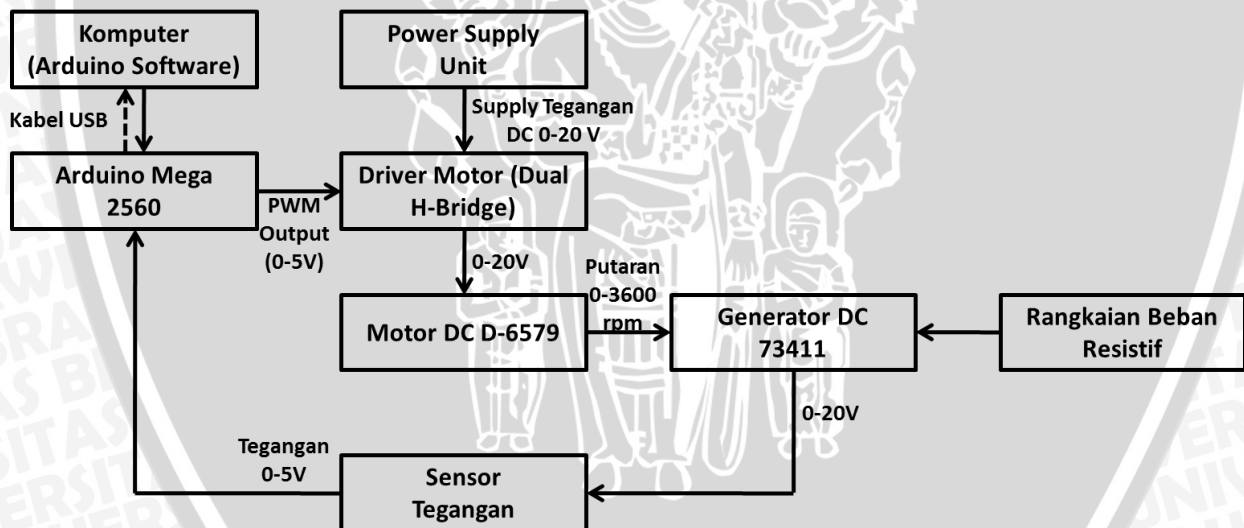
6. Dari hasil identifikasi, didapatkan fungsi alih motor generator DC 73411 seperti pada persamaan 3-1.

$$\frac{Y(s)}{U(s)} = \frac{5,088}{s^2 + 8,316s + 7,057} \quad (3-1)$$

Dari fungsi alih pada yang telah diperoleh, dapat diketahui bahwa *plant* memiliki *steady state gain* sebesar 0,72 dan faktor redaman sebesar 1,507.

3.4 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya. Hal ini dimaksudkan agar pemodelan Generator DC (734 11) dengan pengontrol tegangan *output* menggunakan *Model Reference Adaptive Control* (MRAC) dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Hal yang perlu dilakukan pada pembuatan perangkat keras adalah pembuatan skema perangkat keras dan pemilihan modul elektronik. Skema dari perangkat keras adalah seperti pada Gambar 3.8.



Gambar 3. 8 Skema pembuatan perangkat keras

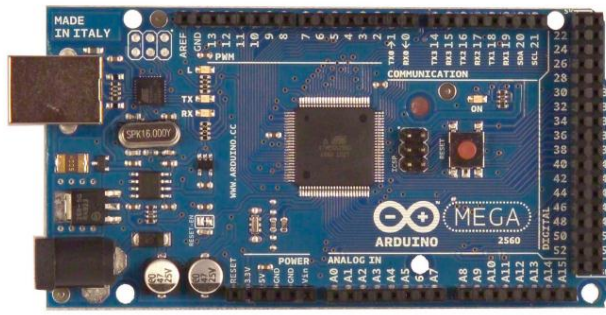
Modul elektronik yang digunakan adalah sebagai berikut:

1. Power Supply Unit sebagai catu daya driver motor



Gambar 3.9 Power Supply Unit (PSU)

2. Mikrokontroler arduino mega 2560 sebagai perangkat kontroler.



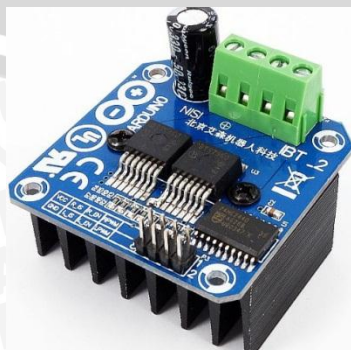
Gambar 3.10 Arduino mega 2560

3. Sensor tegangan digunakan sebagai pembagi tegangan keluaran motor generator DC agar sesuai dengan tegangan masukan arduino.



Gambar 3.11 Sensor tegangan

4. Driver motor H-Bridge BTS 7960



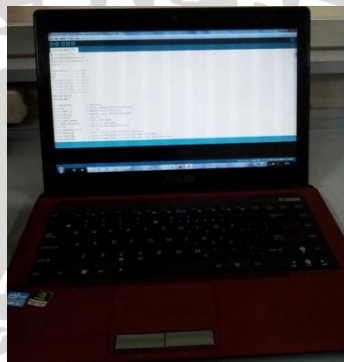
Gambar 3.12 Driver motor H-Bridge BTS 7960

- Rangkaian beban resistif lampu bohlam 2,5 volt



Gambar 3.13 Rangkaian beban resistif

- Komputer atau PC yang sudah terinstall software arduino



Gambar 3.14 Komputer yang terinstall arduino

- Plant motor generator DC 73411



Gambar 3.15 Plant motor generator DC 73411

3.5 Perancangan Algoritma

Terdapat lima tahapan pada proses perancangan algoritma, yaitu:

1. Desain kontroler PI-D menggunakan teknik MRAC,
2. Penentuan model referensi,
3. Penetapan parameter kontroler,
4. Desain persamaan beda, dan
5. Pembuatan *flowchart* program.

Adapun penjelasan kelima tahapan tersebut adalah sebagai berikut:

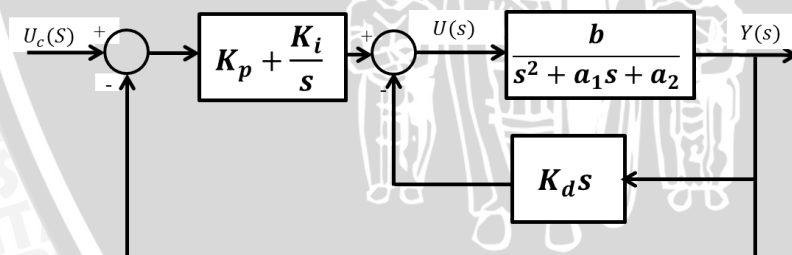
3.5.1 Desain Kontroler PI-D Menggunakan Teknik MRAC

Fungsi alih motor generator DC 73411 merupakan orde dua seperti pada persamaan 3-

1. Untuk menjadikan fungsi alih tersebut menjadi persamaan orde dua yang umum, maka koefisien dan konstanta pada persamaan 3-1 diubah menjadi variabel seperti pada persamaan 3-2.

$$\frac{Y(s)}{U(s)} = \frac{b}{s^2 + a_1s + a_2} \quad (3-2)$$

Kontroler PI-D yang digunakan dapat dilihat pada Gambar 3.16.



Gambar 3.16 Diagram blok sistem *loop* tertutup

Fungsi alih dari diagram blok pada Gambar 3.16 adalah seperti pada persamaan 3-3.

$$\frac{Y(s)}{U_c(s)} = \frac{bK_p s + bK_i}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \quad (3-3)$$

Model referensi ditentukan melalui derajat relatif sistem. Derajat relatif model referensi sistem harus sama dengan derajat relatif *plant*. Derajat relatif adalah selisih antara derajat polinomial pole dengan derajat polinomial zero. Orde model referensi disesuaikan dengan orde *plant*, sehingga model referensi dapat diasumsikan dalam bentuk fungsi alih orde 3 pada Persamaan 3-4.

$$\frac{Y_m(s)}{U_c(s)} = \frac{\beta s + \alpha_1}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} \quad (3-4)$$

Dengan menyamakan persamaan 3-3 dan 3-4 maka didapatkan persamaan 3-5 sampai 3-8.

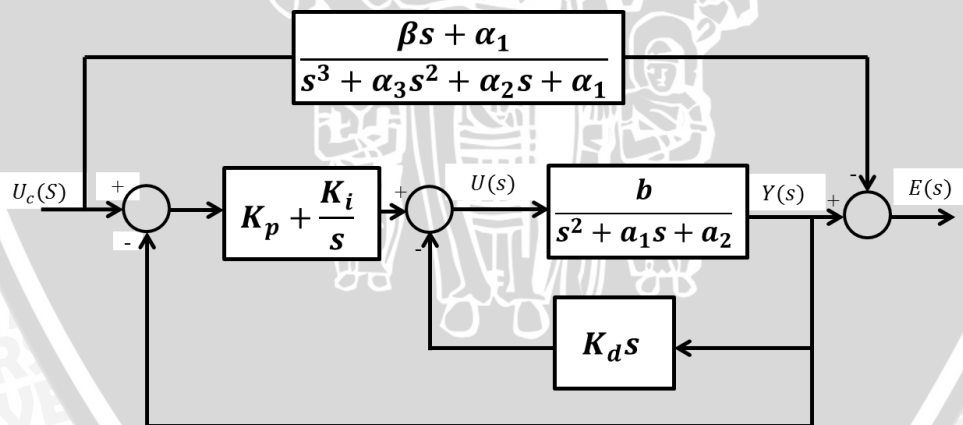
$$\beta = bK_p \quad (3-5)$$

$$\alpha_1 = bK_i \quad (3-6)$$

$$\alpha_2 = a_2 + bK_p \quad (3-7)$$

$$\alpha_3 = a_1 + bK_d \quad (3-8)$$

Diagram blok pengontrolan *plant* dengan penalaan struktur PI-D dengan teknik MRAC dapat dilihat pada Gambar 3.17.



Gambar 3.17 Diagram blok sistem dengan model referensi

Perubahan parameter kontroler terhadap waktu dinyatakan dalam model pendekatan gradien dimana *error* didefinisikan sebagai selisih antara respon *plant* y dan respon model referensi y_m ($e = y - y_m$). Sehingga memungkinkan untuk didapatkan parameter kontroler K_p , K_i dan K_d dengan menerapkan aturan MIT $\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}$ dan θ didapat dari penurunan berikut:

$$\frac{\partial e}{\partial \theta} = \frac{\partial (y - y_m)}{\partial \theta} \quad (3-9)$$

$$\frac{d\theta}{dt} = s\theta = -\gamma e \frac{\partial e}{\partial \theta} \quad (3-10)$$

dari persamaan 3-9 dan 3-10, maka didapatkan persamaan 3-11,

$$\theta = \frac{-\gamma}{s} e \frac{\partial e}{\partial \theta} \quad (3-11)$$

dengan mengganti θ dengan K_p dan K_i maka diperoleh persamaan 3-12 sampai 3-14,

$$\frac{dK_p}{dt} = -\gamma_p \left(\frac{\partial J}{\partial K_p} \right) = -\gamma_p \left(\frac{\partial J}{\partial e} \right) \left(\frac{\partial e}{\partial y} \right) \left(\frac{\partial y}{\partial K_p} \right) \quad (3-12)$$

$$\frac{dK_i}{dt} = -\gamma_i \left(\frac{\partial J}{\partial K_i} \right) = -\gamma_i \left(\frac{\partial J}{\partial e} \right) \left(\frac{\partial e}{\partial y} \right) \left(\frac{\partial y}{\partial K_i} \right) \quad (3-13)$$

$$\frac{dK_d}{dt} = -\gamma_d \left(\frac{\partial J}{\partial K_d} \right) = -\gamma_d \left(\frac{\partial J}{\partial e} \right) \left(\frac{\partial e}{\partial y} \right) \left(\frac{\partial y}{\partial K_d} \right) \quad (3-14)$$

dimana,

$$e = y - y_m \quad (3-15)$$

$$\frac{\partial e}{\partial y} = 1 \quad (3-16)$$

$$\frac{\partial J}{\partial y} = e \quad (3-17)$$

dengan menggunakan hubungan persamaan 3-12 sampai 3-17, maka didapatkan persamaan 3-18 sampai 3-20,

$$\frac{dK_p}{dt} = -\gamma_p \left(\frac{\partial J}{\partial K_p} \right) = -\gamma_p e \left(\frac{\partial y}{\partial K_p} \right) \quad (3-18)$$

$$\frac{dK_i}{dt} = -\gamma_i \left(\frac{\partial J}{\partial K_i} \right) = -\gamma_i e \left(\frac{\partial y}{\partial K_i} \right) \quad (3-19)$$

$$\frac{dK_d}{dt} = -\gamma_d \left(\frac{\partial J}{\partial K_d} \right) = -\gamma_d e \left(\frac{\partial y}{\partial K_d} \right) \quad (3-20)$$

untuk mendapatkan persamaan $\frac{\partial y}{\partial K_p}$, $\frac{\partial y}{\partial K_i}$, $\frac{\partial y}{\partial K_d}$ adalah dengan cara menurunkan persamaan

3-3 yang dituliskan kembali menjadi persamaan 3-21,

$$y(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i) = (bK_p s + bK_i)u_c \quad (3-21)$$

dengan memisalkan ruas kiri pada persamaan 3-21 adalah A dan ruas kanan pada persamaan 3-21 adalah B , maka didapatkan persamaan 3-22,

$$\frac{\partial}{\partial K_p} (y(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)) = \frac{\partial}{\partial K_p} ((bK_p s + bK_i)u_c) \quad (3-22)$$

$$\frac{\partial}{\partial K_p} (A) = \frac{\partial}{\partial K_p} (B)$$

turunan dari A adalah seperti pada persamaan 3-24,

$$\frac{\partial}{\partial K_p} (A) = \left(y \frac{\partial}{\partial K_p} ((s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)) \right) + \left(((s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i) \frac{\partial y}{\partial K_p}) \right) \quad (3-23)$$

$$\frac{\partial}{\partial K_p} (A) = ybs + \left((s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i) \frac{\partial y}{\partial K_p} \right) \quad (3-24)$$

turunan dari B adalah seperti pada persamaan 3-25,

$$\frac{\partial}{\partial K_p} (B) = bsu_c \quad (3-25)$$

dengan mensubstitusikan persamaan 3-24 dan 3-25 kedalam persamaan 3-22, maka didapatkan persamaan 3-26 dan 3-27,

$$ybs + \left((s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i) \frac{\partial y}{\partial K_p} \right) = bsu_c \quad (3-26)$$

$$\frac{\partial y}{\partial K_p} = \frac{bs}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} (u_c - y) \quad (3-27)$$

dengan cara yang sama seperti pada persamaan 3-21 sampai 3-27, maka didapatkan $\frac{\partial y}{\partial K_i}$ dan

$\frac{\partial y}{\partial K_d}$ seperti pada persamaan 3-28 dan 3-29,

$$\frac{\partial y}{\partial K_p} = \frac{b}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} (u_c - y) \quad (3-28)$$

$$\frac{\partial y}{\partial K_p} = \frac{-bs^2}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} (u_c - y) \quad (3-29)$$

dengan mensubstitusikan persamaan 3-27 sampai 3-29 kedalam persamaan 3-18 sampai 3-20, maka didapatkan persamaan 3-30 sampai 3-32.

$$\frac{dK_p}{dt} = -\gamma_p e \frac{bs}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} (u_c - y) \quad (3-30)$$

$$\frac{dK_i}{dt} = \frac{-\gamma_i}{s} e \frac{b}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} (u_c - y) \quad (3-31)$$

$$\frac{dK_d}{dt} = \frac{-\gamma_d}{s} e \frac{-bs^2}{(s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i)} y \quad (3-32)$$

Dari persamaan 3-30 sampai 3-32, *gain parameter* tidak dapat ditemukan ketika nilai α_1 , α_2 dan b tidak diketahui. Jadi rumus yang diturunkan dari aturan MIT ini belum dapat digunakan. Sebagai gantinya, beberapa pendekatan diperlukan. Jika berfikir sistem merupakan model yang sempurna, bandingkan hubungan antara *input-output* dari sistem dengan model referensi, pendekatan yang akan digunakan seperti pada persamaan 3-33 sampai 3-36.

$$\beta = b \quad (3-33)$$

$$\alpha_1 = bK_i \quad (3-34)$$

$$\alpha_2 = (a_2 + bK_p) \quad (3-35)$$

$$\alpha_3 = (a_1 + bK_d) \quad (3-36)$$

dengan mensubstitusikan persamaan 3-33 sampai 3-36 kedalam persamaan 3-30 sampai 3-32, maka didapatkan persamaan 3-37 sampai 3-39,

$$\frac{dK_p}{dt} = -\gamma_p e \frac{\beta s}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} (u_c - y) \quad (3-37)$$

$$\frac{dK_i}{dt} = \frac{-\gamma_i}{s} e \frac{\beta}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} (u_c - y) \quad (3-38)$$

$$\frac{dK_d}{dt} = \frac{-\gamma_d}{s} e \frac{-\beta s^2}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} y \quad (3-39)$$

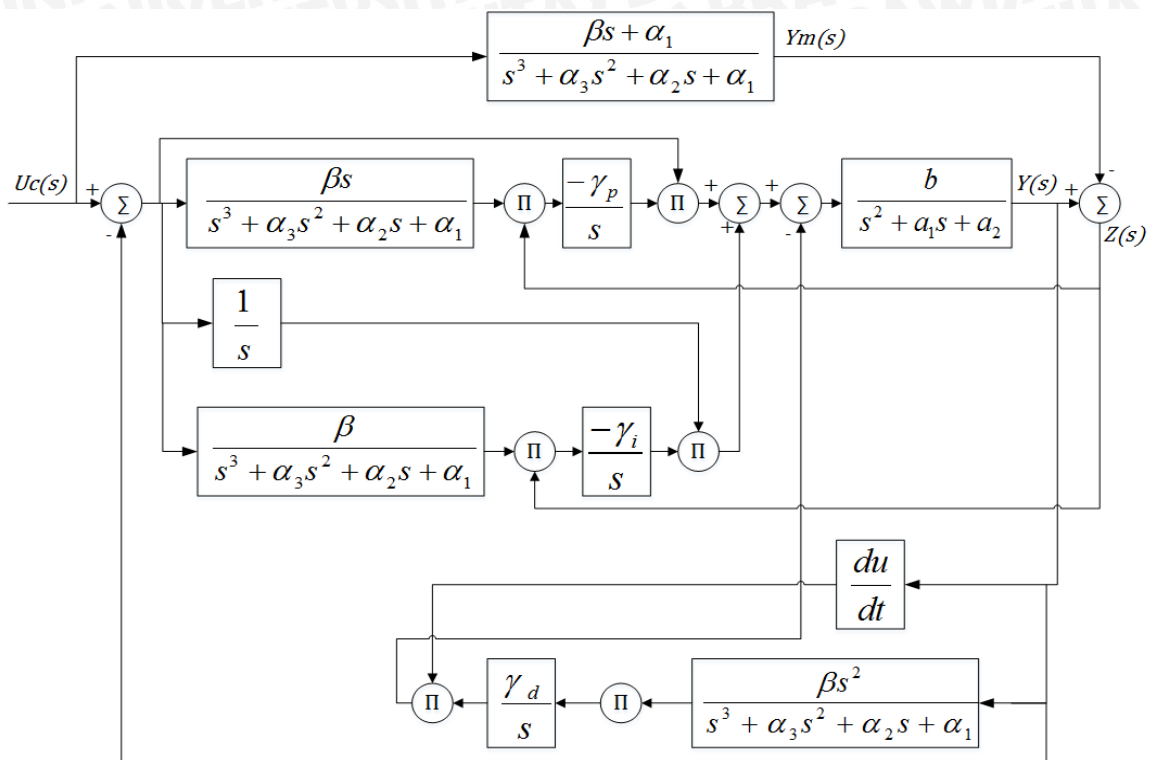
dengan mengintegrasikan kedua ruas pada persamaan 3-37 sampai 3-39, maka didapatkan persamaan 3-40 sampai 3-42,

$$K_p = \frac{-\gamma_p}{s} e \frac{\beta s}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} (u_c - y) \quad (3-40)$$

$$K_i = \frac{-\gamma_i}{s} e \frac{\beta}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} (u_c - y) \quad (3-41)$$

$$K_d = \frac{\gamma_d}{s} e \frac{\beta s^2}{s^3 + \alpha_3 s^2 + \alpha_2 s + \alpha_1} y \quad (3-42)$$

Representasi diagram blok berdasarkan penurunan yang telah diperoleh adalah seperti pada Gambar 3.18.



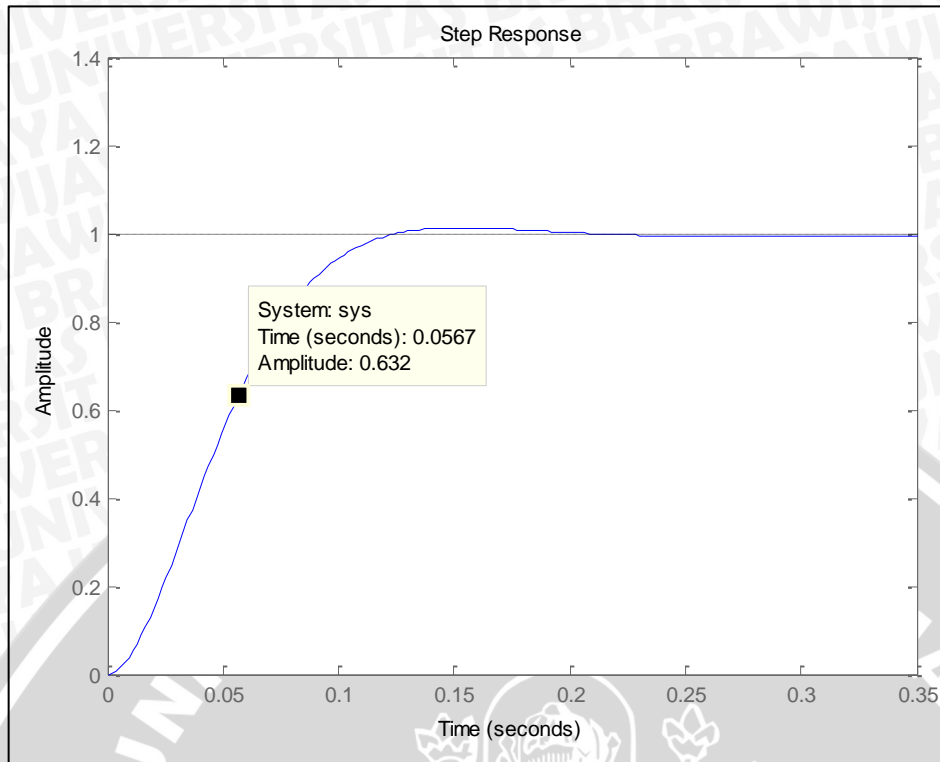
Gambar 3.18 Diagram blok penggunaan teknik MRAC pada penalaan struktur PI-D

3.5.2 Penentuan Model Referensi

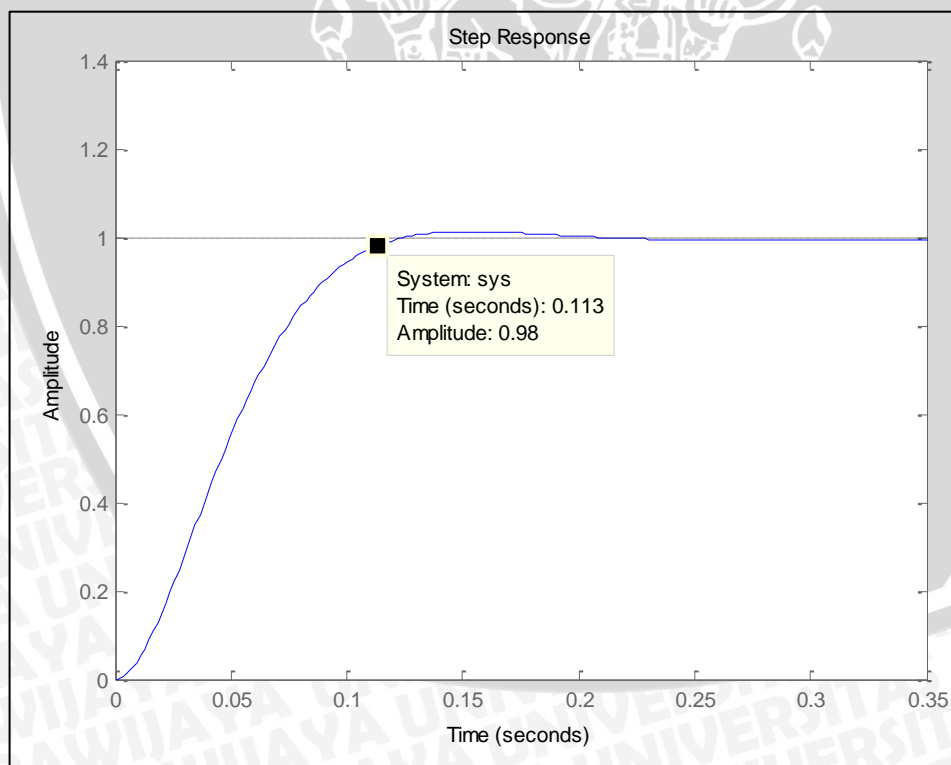
Penentuan model referensi dilakukan dengan cara *try and error* sehingga didapatkan respon yang stabil yang mengikuti karakteristik dari *plant*. Model referensi yang akan ditentukan tidak memiliki *error steady state*, tidak terdapat osilasi dan memiliki *settling time* yang cepat. Penentuan model referensi dapat ditentukan dengan cara melihat hasil respon saat simulasi melalui aplikasi matlab dan pada saat implementasi sistem. Fungsi alih dari model referensi adalah seperti pada persamaan 3-43.

$$\frac{Y_m(s)}{U_c(s)} = \frac{1052,3s+379,5}{s^3+50,79s^2+1079,55s+379,5} \tag{3-43}$$

Dengan memberikan masukan unit step dari program matlab, maka dapat diketahui bahwa respon model referensi memiliki nilai *time constant* sebesar 0,057 detik dan nilai *settling time* sebesar 0,102 detik. Respon transien dari model referensi yang telah diberi masukan unit step adalah seperti pada Gambar 3.19 dan 3.20.



Gambar 3.19 Nilai *time constant* respon model dengan masukan unit step



Gambar 3.20 Nilai *settling time* respon model dengan masukan unit step

3.5.3 Penetapan Parameter Kontroler

Penetapan parameter kontroler terhadap waktu dinyatakan dalam pendekatan gradien dimana *error* didefinisikan sebagai selisih antara keluaran *plant* y dan keluaran model referensi y_m ($e = y - y_m$). Sehingga memungkinkan untuk didapatkan parameter kontroler K_p , K_i , dan K_d dengan menerapkan aturan MIT. Berdasarkan desain kontroler PI-D dengan teknik MRAC yang telah dijelaskan sebelumnya, maka akan didapatkan persamaan 3-44 sampai 3-46.

$$K_p = \frac{-\gamma_p}{s} e \frac{1052,3s}{s^3+50,79s^2+1079,55s+379,5} (u_c - y) \quad (3-44)$$

$$K_i = \frac{-\gamma_i}{s} e \frac{1052,3}{s^3+50,79s^2+1079,55s+379,5} (u_c - y) \quad (3-45)$$

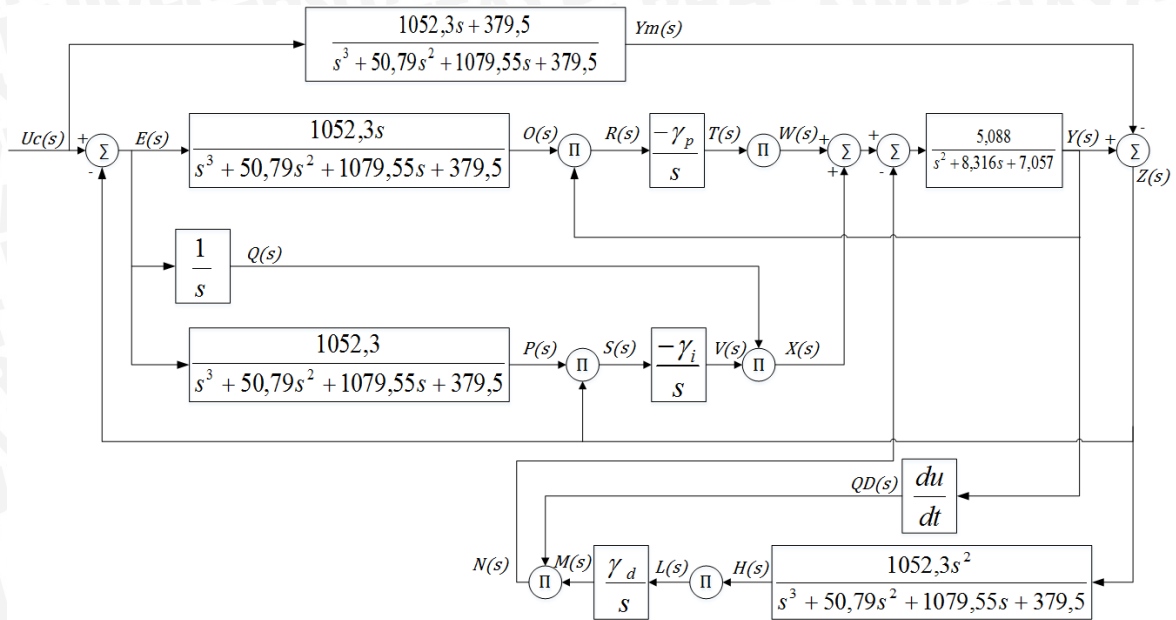
$$K_d = \frac{\gamma_d}{s} e \frac{1052,3s^2}{s^3+50,79s^2+1079,55s+379,5} y \quad (3-46)$$

Pada persamaan 3-44 sampai 3-46, terdapat dua parameter yang harus ditentukan nilainya terlebih dahulu dalam mendesain sistem yang akan dikontrol, yaitu parameter γ_p , γ_i , dan γ_d . Dimana cara menentukan nilai parameter tersebut dilakukan dengan cara *try and error*.

3.5.4 Desain Persamaan Beda

Terdapat banyak cara yang dapat digunakan untuk proses diskritisasi, tiga diantaranya yang banyak digunakan dalam bidang kontrol adalah *backward difference*, *forward difference* (Metode *Euler*), dan *bilinear transformation*. Ketiga metode tersebut hanya merupakan pendekatan, sehingga hasilnya tidak akan persis sama dengan bentuk analog karena dalam bentuk diskrit selalu ada sebagian informasi yang hilang, yaitu informasi yang terjadi antara satu cuplikan dengan cuplikan lainnya walaupun frekuensi cuplik yang digunakan tinggi. Selain itu, formula matematika yang digunakan dalam proses diskritisasi juga diturunkan berdasarkan pendekatan agar lebih mudah digunakan. Pada perancangan ini akan digunakan metode *backward difference* karena seluruh daerah stabil dan sebagian daerah tidak stabil di bidang s di-*mapping* ke daerah stabil z , hal inilah yang menyebabkan metode *backward difference* jauh lebih *robust* dibandingkan dengan kedua metode lainnya (Ibrahim, 2016: 41).

Berdasarkan hasil dari penentuan fungsi alih, model referensi, dan parameter kontroler dapat disusun menjadi sebuah sistem yang direpresentasikan dalam diagram blok seperti dalam Gambar 3.21.



Gambar 3.21 Diagram blok sistem dengan desain struktur PI-D menggunakan teknik MRAC

Model referensi dari sistem dinyatakan dalam bentuk frekuensi (dalam bentuk s) seperti pada persamaan . Dengan menggunakan metode backward difference, ganti operator s dalam persamaan analog bentuk s dengan persamaan 3-47.

$$s = \frac{1-z^{-1}}{T_s} \quad (3-47)$$

Dimana T_s merupakan periode sampling. Dengan mensubstitusikan persamaan 3-47 kedalam blok model referensi, maka didapatkan persamaan $Y_m(z)$ seperti pada persamaan 3-48 dan 3-49.

$$Y_m(z) = \frac{1052,3\left(\frac{1-z^{-1}}{T_s}\right) + 379,5}{\left(\frac{1-z^{-1}}{T_s}\right)^3 + 50,79\left(\frac{1-z^{-1}}{T_s}\right)^2 + 1079,55\left(\frac{1-z^{-1}}{T_s}\right) + 379,5} U_c(z) \times \frac{T_s}{T_s} \quad (3-48)$$

$$Y_m(z) = \frac{1052,3T_s^2(1-z^{-1}) + 379,5T_s^3}{(1-z^{-1})^3 + 50,79T_s(1-z^{-1})^2 + 1079,55T_s^2(1-z^{-1}) + 379,5T_s^3} U_c(z) \quad (3-49)$$

Dengan mengalikan silang kedua ruas, maka didapatkan persamaan 3-50,

$$Y_m(z) \left((379,5T_s^3 + 1079,55T_s^2 + 50,79T_s + 1) - (1079,55T_s^2 + 101,58T_s + 3)z^{-1} + (50,79T_s + 3)z^{-2} - z^{-3} \right) = \left((379,5T_s^3 + 1052,3T_s^2) - 1052,3T_s^2z^{-1} \right) U_c(z) \quad (3-50)$$

persamaan 3-50 disusun berdasarkan variabelnya (Y_m atau U_c) dan berdasarkan jenis operatornya (z^{-n}), sehingga didapatkan persamaan 3-51,

$$Y_m(z)(1291T_s^3 + 583,75T_s^2 + 71,87T_s + 1) = (1079,55T_s^2 + 101,58T_s + 3)Y_m(z)z^{-1} - (50,79T_s + 3)Y_m(z)z^{-2} + Y_m(z)z^{-3} + (379,5T_s^3 + 1052,3T_s^2)U_c(z) - 1052,3T_s^2z^{-1}U_c(z) \quad (3-51)$$

apabila persamaan 3-51 disederhanakan lagi, maka didapatkan persamaan 3-52,

$$Y_m(z) = \frac{A}{G}Y_m(z)z^{-1} - \frac{B}{G}Y_m(z)z^{-2} + \frac{C}{G}Y_m(z)z^{-3} + \frac{D}{G}U_c(z) - \frac{E}{G}z^{-1}U_c(z) \quad (3-52)$$

dalam bentuk time domain, dengan mengganti operator z^0 menjadi waktu saat ini (k), z^{-1} menjadi waktu sesaat sebelumnya atau satu kali waktu tunda ($k - 1$), dan seterusnya maka akan didapatkan persamaan 3-53

$$y_m(k) = \frac{A}{G}y_m(k - 1) - \frac{B}{G}y_m(k - 2) + \frac{C}{G}y_m(k - 3) + \frac{D}{G}u_c(k) - \frac{E}{G}u_c(k - 1) \quad (3-53)$$

dengan menggunakan cara yang sama seperti persamaan 3-47 sampai 3-53, maka didapatkan persamaan model referensi untuk K_p , K_i , dan K_d seperti pada persamaan 3-54 sampai 3-56,

$$o(k) = \frac{A}{G}o(k - 1) - \frac{B}{G}o(k - 2) + \frac{C}{G}o(k - 3) + \frac{E}{G}e(k) - \frac{E}{G}e(k - 1) \quad (3-54)$$

$$p(k) = \frac{A}{G}p(k - 1) - \frac{B}{G}p(k - 2) + \frac{C}{G}p(k - 3) + \frac{F}{G}e(k) \quad (3-55)$$

$$h(k) = \frac{A}{G}h(k - 1) - \frac{B}{G}h(k - 2) + \frac{C}{G}h(k - 3) + \frac{J}{G}y(k) - \frac{2J}{G}y(k - 1) + \frac{J}{G}y(k - 2) \quad (3-56)$$

dimana,

$o(k)$ = Persamaan model referensi K_p ,

$p(k)$ = Persamaan model referensi K_i ,

$h(k)$ = Persamaan model referensi K_d ,

$$A = 1079,55T_s^2 + 101,58T_s + 3$$

$$B = 50,79T_s + 3$$

$$C = 1$$

$$D = 379,5T_s^3 + 1052,3T_s^2$$

$$E = 1052,3T_s^2$$

$$F = 1052,3T_s^3$$

$$J = 1052,3T_s$$

$$G = 379,5T_s^3 + 1079,55T_s^2 + 50,79T_s + 1$$

dengan menggunakan cara yang sama pula seperti persamaan 3-47 sampai 3-53, maka didapatkan persamaan integrator, diferensiator, serta K_p , K_i , dan K_d seperti pada persamaan 3-57 sampai 3-61,

$$q(k) = q(k - 1) + T_s e(k) \quad (3-57)$$

$$qd(k) = \frac{1}{T_s} y(k) - \frac{1}{T_s} y(k - 1) \quad (3-58)$$

$$t(k) = t(k - 1) - \gamma_p T_s r(k) \quad (3-59)$$

$$v(k) = v(k - 1) - \gamma_i T_s s(k) \quad (3-60)$$

$$m(k) = m(k - 1) + \gamma_d T_s l(k) \quad (3-61)$$

dimana,

$q(k)$ = persamaan Integrator,

$qd(k)$ = persamaan diferensiator,

$t(k)$ = persamaan K_p ,

$v(k)$ = persamaan K_i ,

$m(k)$ = persamaan K_d ,

$r(k) = o(k) * z(k)$

$s(k) = p(k) * z(k)$

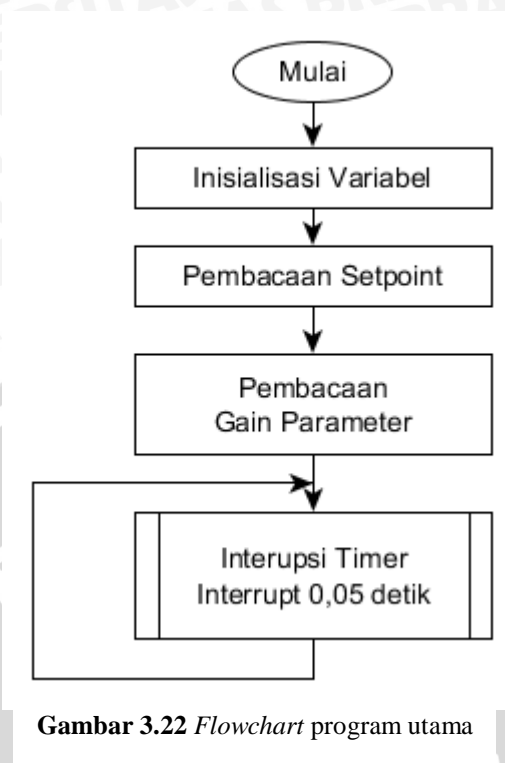
$l(k) = h(k) * z(k)$

$z(k) = y_m(k) - y(k)$

3.5.5 Flowchart Program

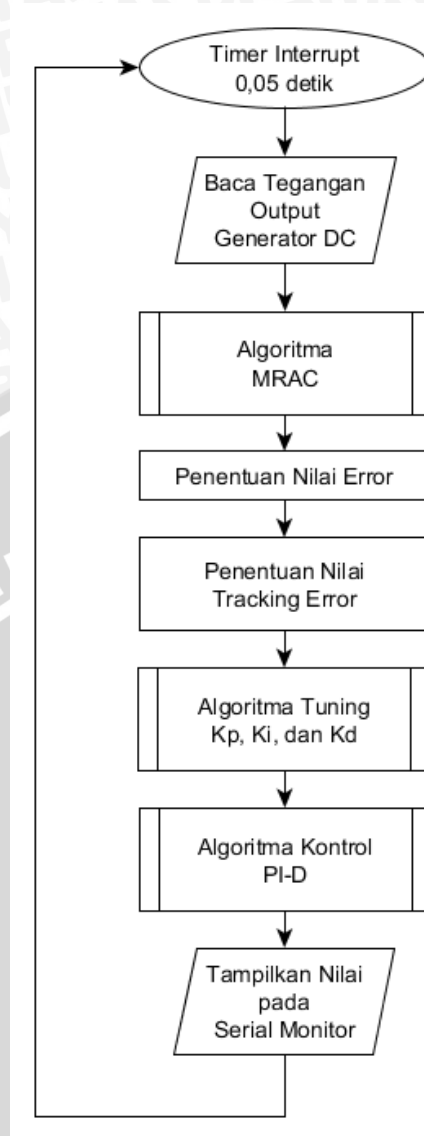
Program diawali dengan inisialisasi masing-masing variabel dan sub-rutin untuk proses kontrol. Pada program utama terdapat sub rutin *timer interrupt* yang dieksekusi setiap 0,05 detik. *Flowchart* dari program utama ditunjukkan pada Gambar 3.22.





Gambar 3.22 Flowchart program utama

Pada flowchart sub-rutin *timer interrupt*, program dieksekusi setiap 0,05 detik sekali dan diawali dengan pembacaan tegangan keluaran *plant*. Setelah itu algoritma kontrol MRAC dijalankan. Kemudian dilakukan perhitungan nilai *error* dengan cara mengurangkan tegangan keluaran *plant* dengan *setpoint*. Selain *error*, ditentukan *tracking error* yaitu selisih antara respon *plant* dan respon model referensi. *Error* dan *tracking error* yang telah didapat digunakan sebagai masukan algoritma penalaan K_p , K_i , dan K_d pada struktur PI-D. Nilai struktur PI-D dikeluarkan sebagai sinyal kontrol yang digunakan untuk mengatur tegangan masukan motor generator DC. Flowchart dari sub-rutin timer interrupt dapat dilihat dalam Gambar 3.23



Gambar 3.23 Flowchart sub rutin timer interrupt

BAB IV HASIL DAN PEMBAHASAN

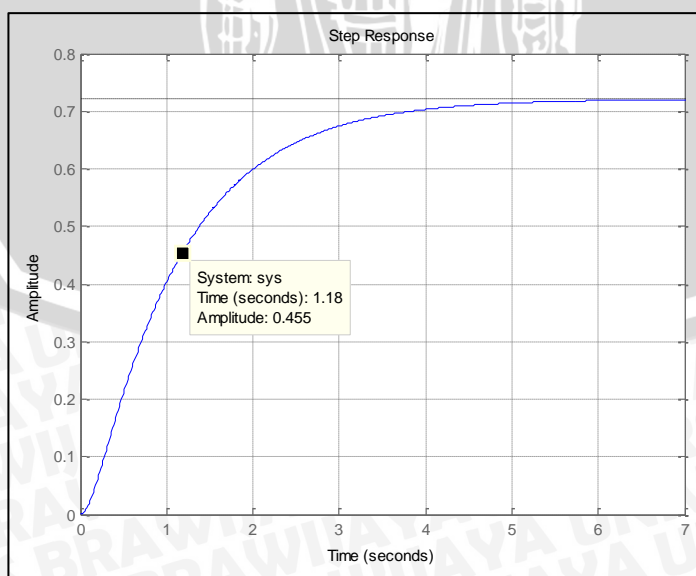
Hasil dan pembahasan diperoleh melalui Identifikasi *plant*, simulasi sistem dan implementasi sistem. Identifikasi *plant* dilakukan untuk mencari fungsi alih *plant* motor generator DC 73411. Simulasi dilakukan untuk mencari gain parameter yang sesuai. Berikut merupakan pembahasan dari hasil yang diperoleh melalui identifikasi *plant*, simulasi sistem dan implementasi sistem:

4.1 Identifikasi *Plant*

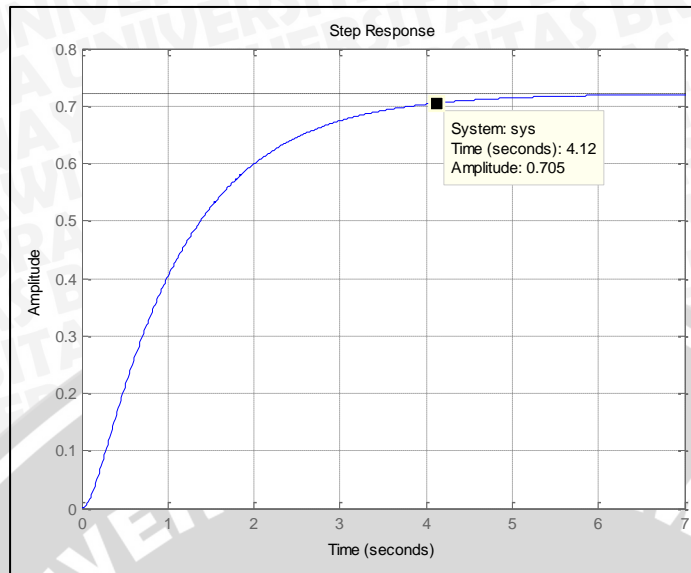
Dari hasil identifikasi, didapatkan fungsi alih motor generator DC 73411 seperti pada persamaan 4-1.

$$\frac{Y(s)}{U(s)} = \frac{5,088}{s^2 + 8,316s + 7,057} \quad (4-1)$$

Dari fungsi alih pada yang telah diperoleh, dapat diketahui bahwa *plant* memiliki *steady state gain* sebesar 0,72 dan faktor redaman sebesar 1,507. Dengan memberikan masukan unit step di program matlab, maka dapat diketahui bahwa *plant* memiliki *time constant* sebesar 1,19 detik dan waktu untuk mencapai *steady state* sebesar 4,12 detik seperti pada gambar 4.1 dan 4.2.

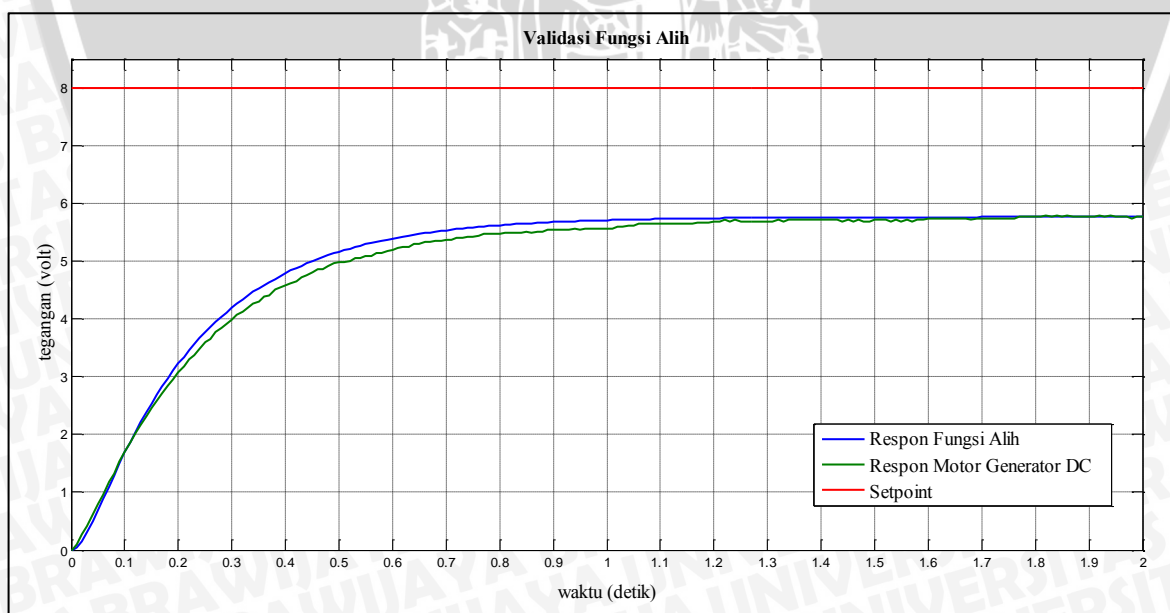


Gambar 4. 1 Nilai *time constant* respon *plant*



Gambar 4. 2 Waktu yang dibutuhkan *plant* mencapai keadaan *steady state*

Proses validasi fungsi alih dilakukan dengan cara membandingkan *output* fungsi alih dan tegangan *output* motor generator DC 734 11 yang didapat dari pembacaan sensor tegangan. Proses validasi dilakukan dengan memberikan masukan step sebesar 8 volt. Grafik dari *output* fungsi alih dan *output* motor generator DC 734 11 ditunjukkan pada gambar 4.3.



Gambar 4. 3 Validasi *output* fungsi alih dengan *output* motor generator DC 73411

Dari Gambar 4.3 dapat dilihat bahwa *output* fungsi alih yang telah didapat dari proses identifikasi hampir menyerupai tegangan *output* motor generator DC 734 11. Jadi fungsi alih yang telah didapatkan dianggap dapat mewakili pemodelan *plant* motor generator DC 734 11.

4.2 Simulasi Sistem

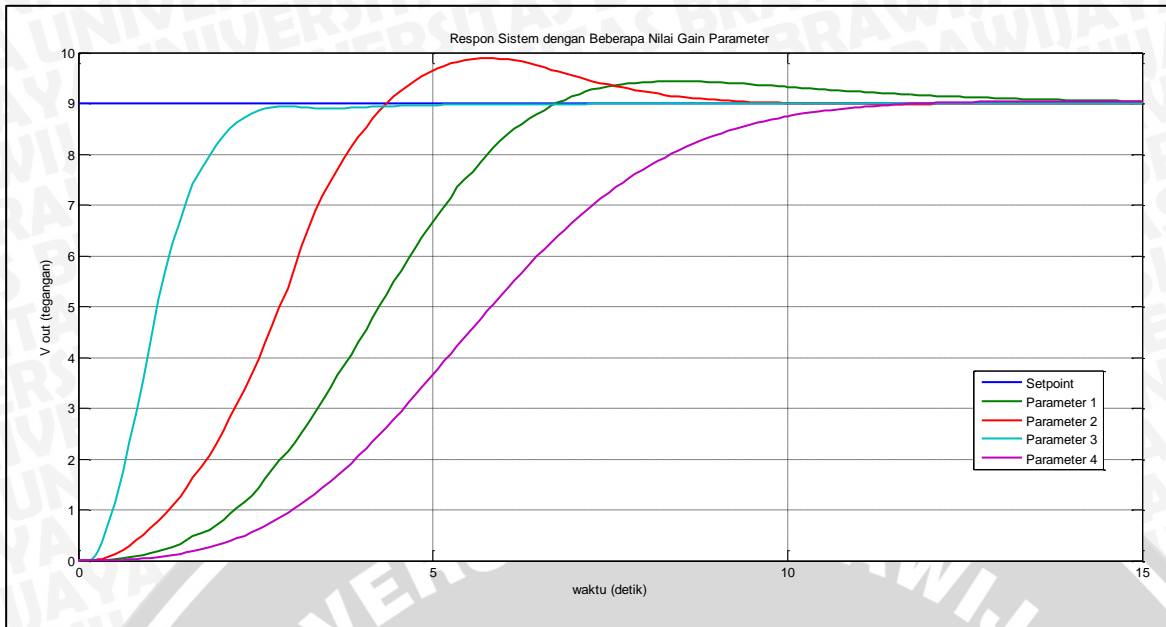
Simulasi dilakukan dengan memberikan masukan berupa sinyal step pada blok diagram simulink yang ditunjukkan lampiran 1.

Penentuan *gain* parameter dilakukan dengan cara *try error* dari nilai yang paling kecil sampai didapatkan respon yang sesuai dengan spesifikasi desain. Terdapat empat nilai *gain* parameter yang disimulasikan yang dituliskan pada Tabel 4.1.

Tabel 4.1 *Gain* parameter

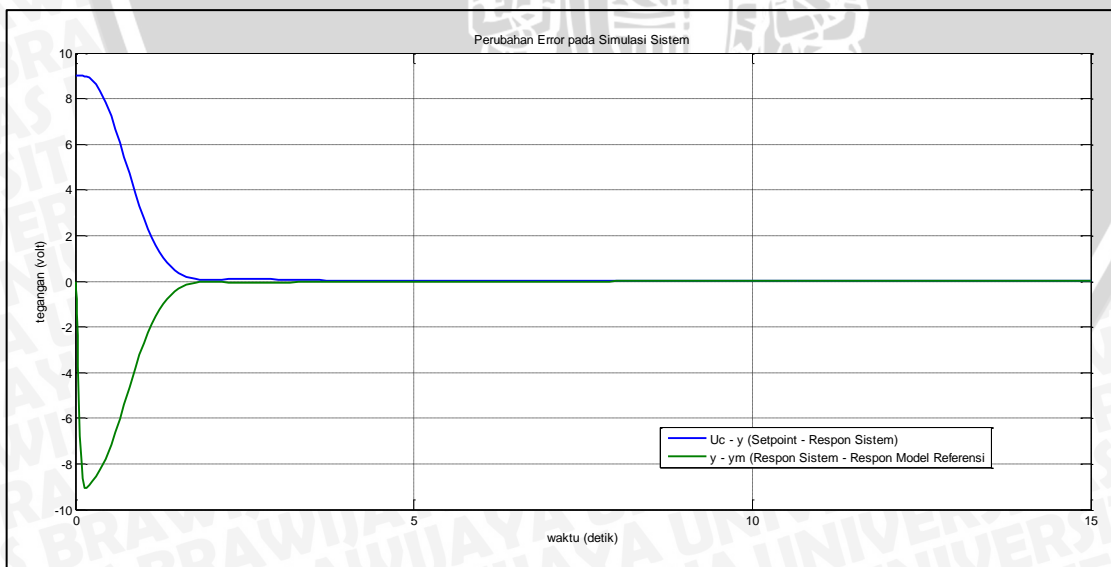
Parameter ke-	γ_p	γ_i	γ_d
1	0,001	0,001	0,001
2	0,005	0,005	0,005
3	0,195	0,07	0,08
4	0,0004	0,0004	0,0004

Respon simulasi sistem dengan menggunakan variasi nilai *gain* parameter ditunjukkan pada Gambar 4.4.



Gambar 4.4 Respon simulasi sistem dengan variasi nilai *gain* parameter

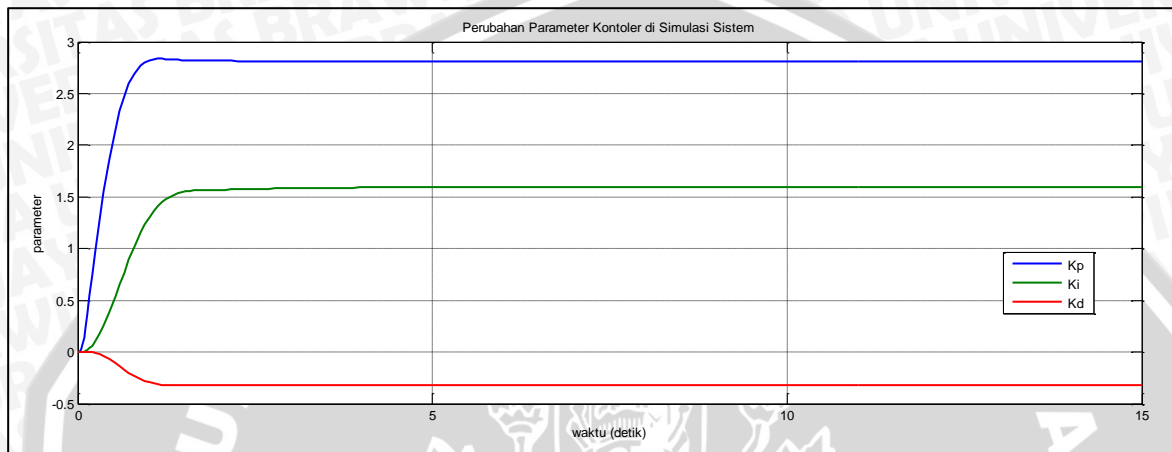
Nilai γ_p , γ_i , dan γ_d ditentukan melalui respon sistem yang memiliki nilai *settling time* paling cepat dan tidak memiliki *overshoot*. Berdasarkan respon simulasi sistem pada Gambar 4.2 diperoleh nilai *gain* parameter yaitu γ_p sebesar 0,195, γ_i sebesar 0,07, dan γ_d sebesar 0,08. Dengan menggunakan gain parameter yang telah diperoleh, maka grafik perubahan *error* ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Perubahan *error* pada simulasi sistem

Dari Gambar 4.5 dapat diketahui bahwa nilai selisih *setpoint* dengan respon sistem menurun sampai pada keadaan *steady state*. Sedangkan nilai selisih respon sistem dengan respon model referensi meningkat hingga pada keadaan *steady state*.

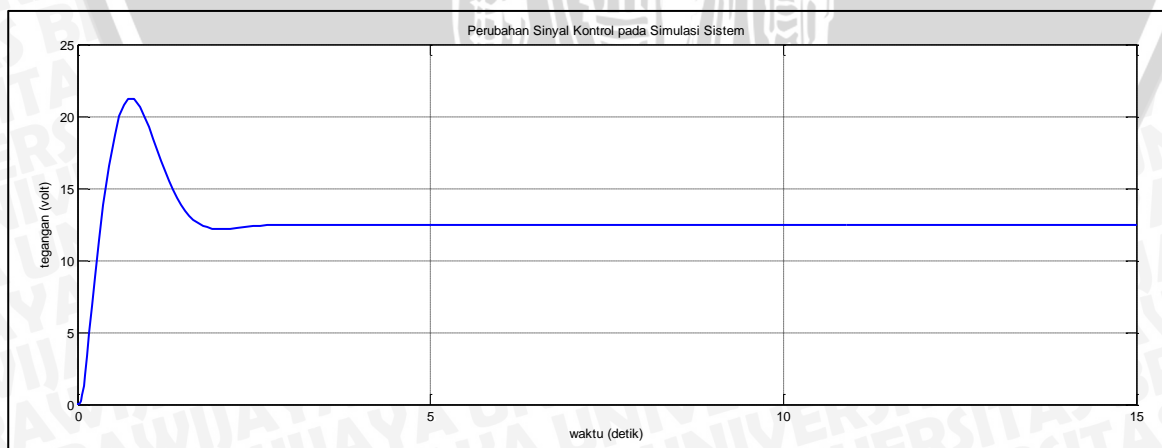
Grafik perubahan parameter struktur PI-D di simulasi sistem ditunjukkan pada Gambar 4.6.



Gambar 4.6 Perubahan parameter struktur PI-D di Simulasi Sistem

Dari Gambar 4.6 dapat diketahui bahwa nilai K_p dan K_i meningkat lalu konstan ketika memasuki *steady state*, sedangkan nilai K_d menurun lalu konstan ketika memasuki *steady state*.

Grafik perubahan sinyal kontrol di simulasi sistem ditunjukkan pada Gambar 4.7.



Gambar 4.7 Perubahan sinyal kontrol di simulasi sistem

Dari Gambar 4.7 dapat diketahui bahwa sinyal kontrol meningkat kemudian menurun hingga keadaan *steady state*. Untuk mengetahui respon simulasi sistem, digunakan tiga nilai *setpoint* yaitu 9 volt, 10 volt, dan 11 volt.

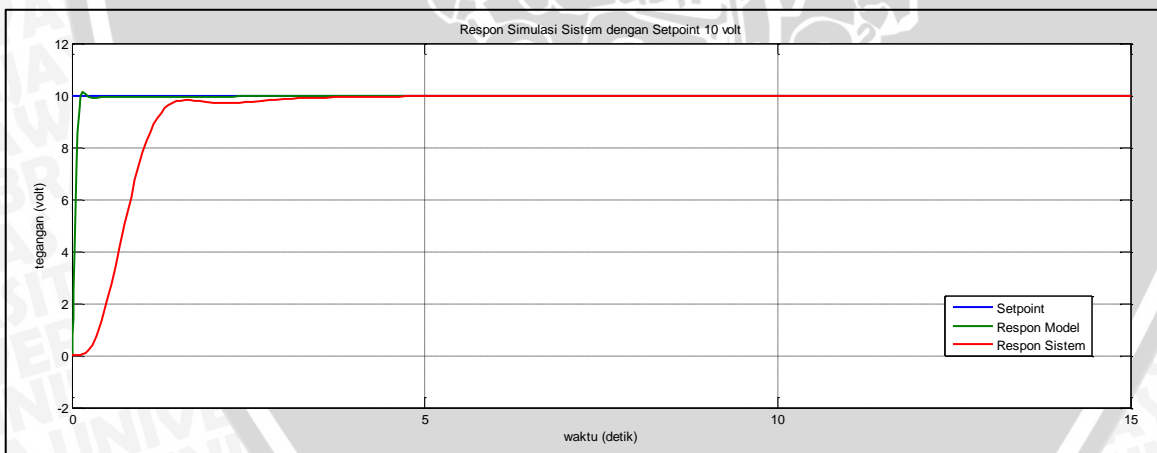
Respon simulasi sistem dengan *setpoint* 9 volt ditunjukkan oleh Gambar 4.8.



Gambar 4.8 Respon simulasi sistem dengan *setpoint* 9 volt

Dari Gambar 4.8 dapat diketahui bahwa respon simulasi sistem dengan *setpoint* 9 volt memiliki nilai *settling time* sebesar 1,46 detik, tidak memiliki *error steady state* dan tidak memiliki *overshoot*.

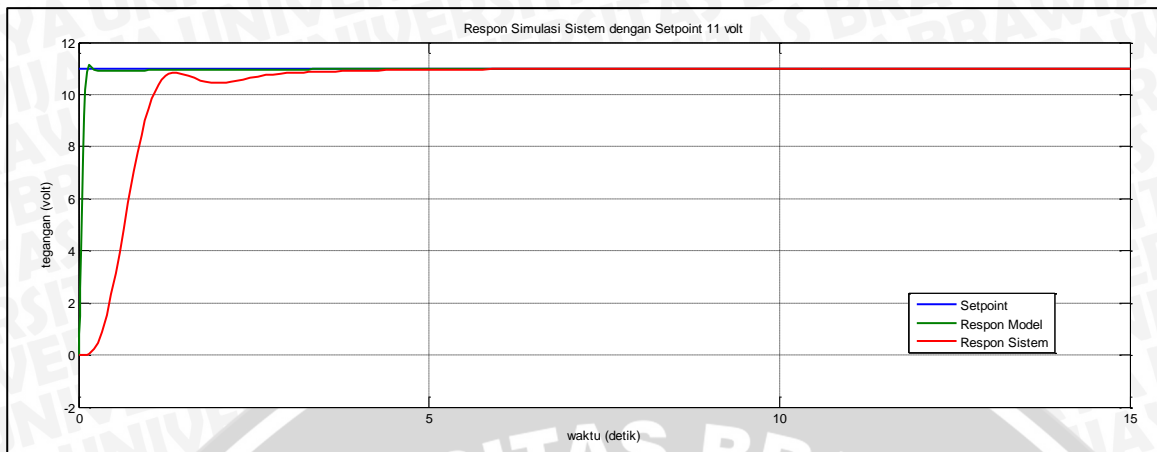
Respon simulasi sistem dengan *setpoint* 10 volt ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Respon simulasi sistem dengan *setpoint* 10 volt

Dari Gambar 4.9 dapat diketahui bahwa respon simulasi sistem dengan *setpoint* 10 volt memiliki nilai *settling time* sebesar 1,31 detik, tidak memiliki *error steady state* dan tidak memiliki *overshoot*.

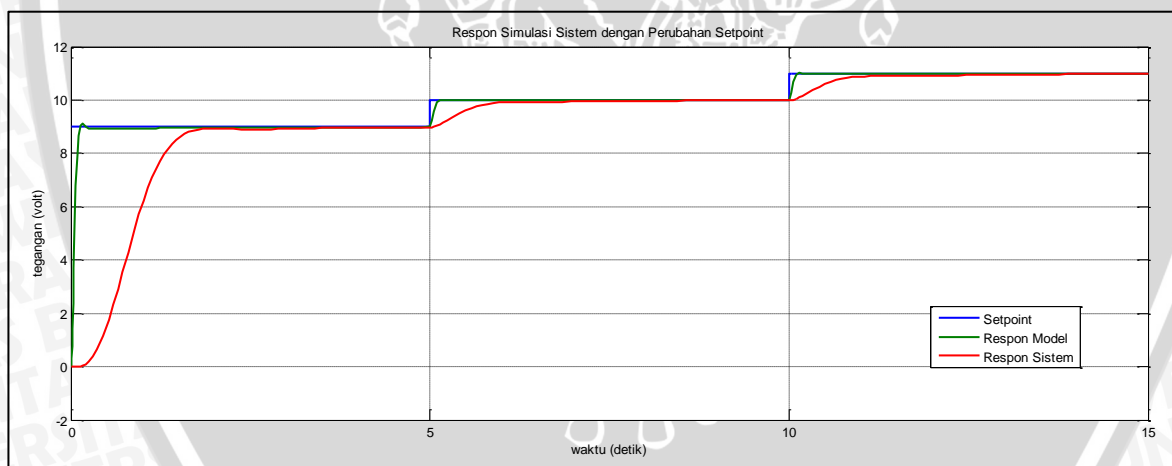
Respon simulasi sistem dengan *setpoint* 11 volt ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Respon simulasi sistem dengan *setpoint* 11 volt

Dari Gambar 4.10 dapat diketahui bahwa respon simulasi sistem dengan *setpoint* 11 volt memiliki nilai *settling time* sebesar 1,18 detik, tidak memiliki *error steady state* dan tidak memiliki *overshoot*.

Respon simulasi sistem dengan perubahan *setpoint* ditunjukkan oleh Gambar 4.11.

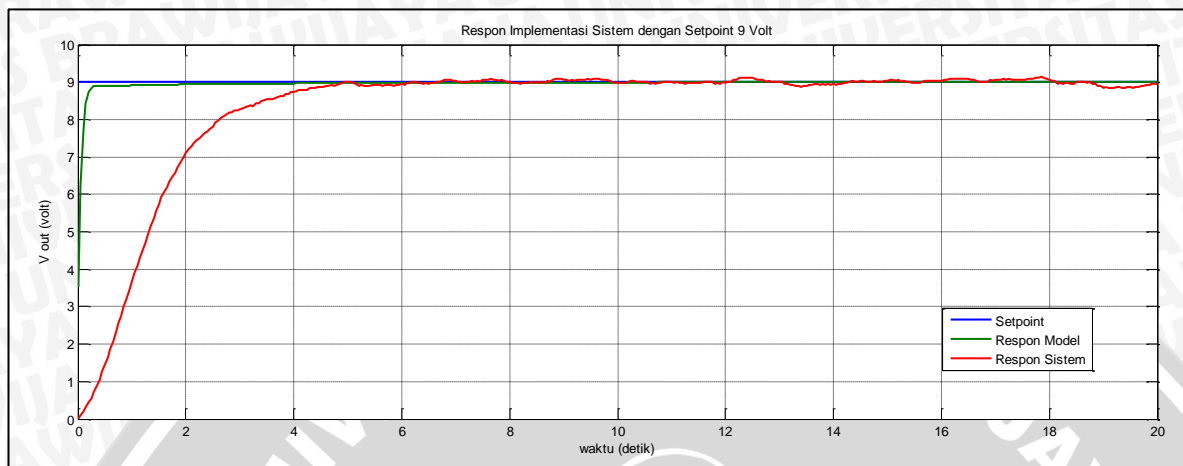


Gambar 4.11 Respon simulasi sistem dengan perubahan *setpoint*

Dari Gambar 4.11 dapat diketahui bahwa respon simulasi sistem dengan perubahan *setpoint* memiliki nilai *settling* sebesar 1,4 detik menuju *setpoint* 9 volt, 0,4 detik menuju *setpoint* 10 volt, 0,4 detik menuju *setpoint* 11 volt, tidak memiliki *error steady state* dan tidak memiliki *overshoot*.

4.3 Implementasi Sistem

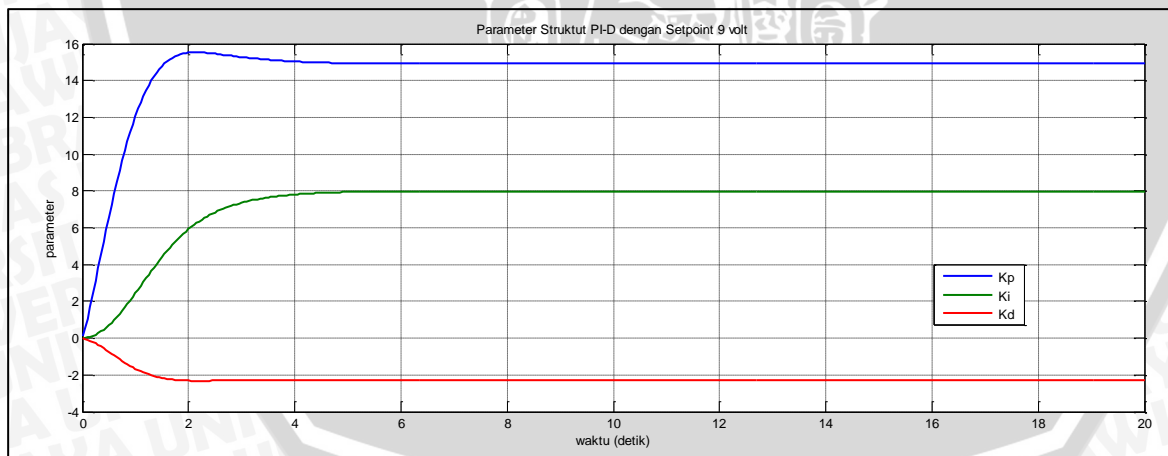
Implementasi sistem menggunakan waktu *sampling* setiap 0,05 detik. *Setpoint* yang digunakan pada implementasi sistem adalah 9 volt, 10 volt, dan 11 volt. Respon implementasi sistem dengan *setpoint* 9 volt ditunjukkan pada Gambar 4.12.



Gambar 4.12 Respon implementasi sistem dengan *setpoint* 9 volt

Dari Gambar 4.12 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 9 volt memiliki nilai *settling time* sebesar 3,55 detik, *error steady state* sebesar 0,7% dan tidak memiliki *overshoot*.

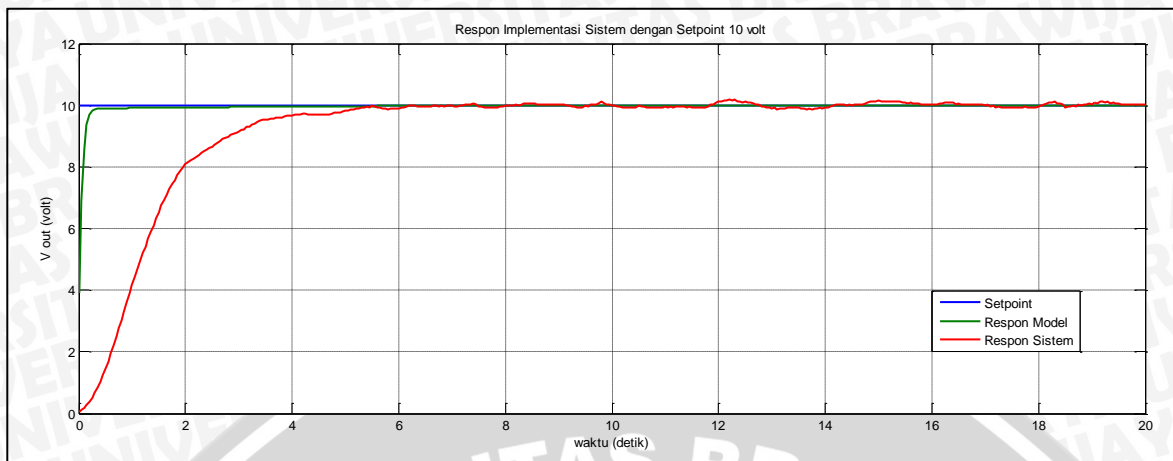
Perubahan parameter struktur PI-D dengan *setpoint* 9 volt ditunjukkan pada Gambar 4.13.



Gambar 4.13 Parameter struktur PI-D dengan *setpoint* 9 volt

Dari Gambar 4.13 dapat diketahui bahwa K_p dan K_i meningkat lalu konstan ketika memasuki *steady state*, sedangkan nilai K_d menurun lalu konstan ketika memasuki *steady state*.

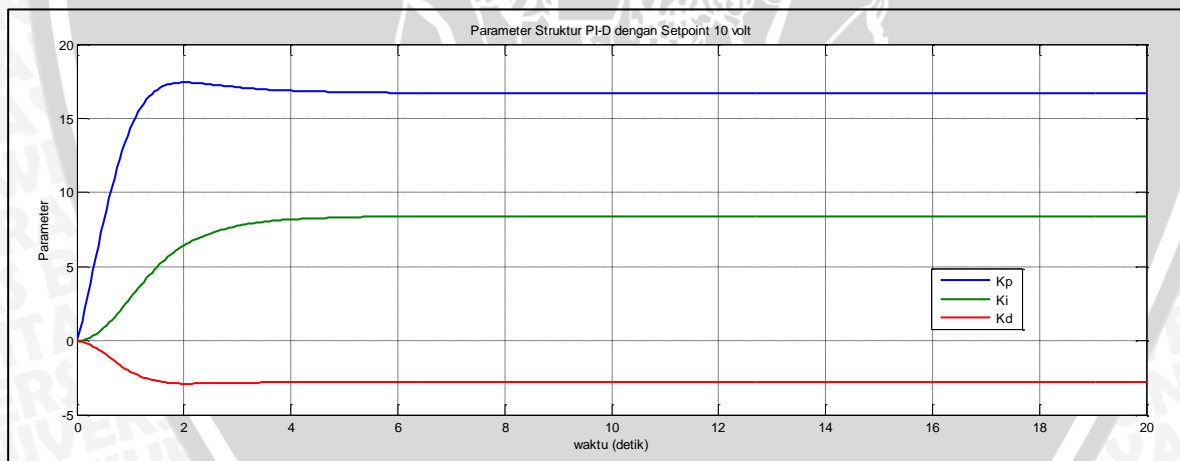
Respon implementasi sistem dengan *setpoint* 10 volt ditunjukkan pada Gambar 4.14.



Gambar 4.14 Respon implementasi sistem dengan *setpoint* 10 volt

Dari Gambar 4.14 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 10 volt memiliki nilai *settling time* sebesar 3,575 detik, *error steady state* sebesar 0,8% dan tidak memiliki *overshoot*.

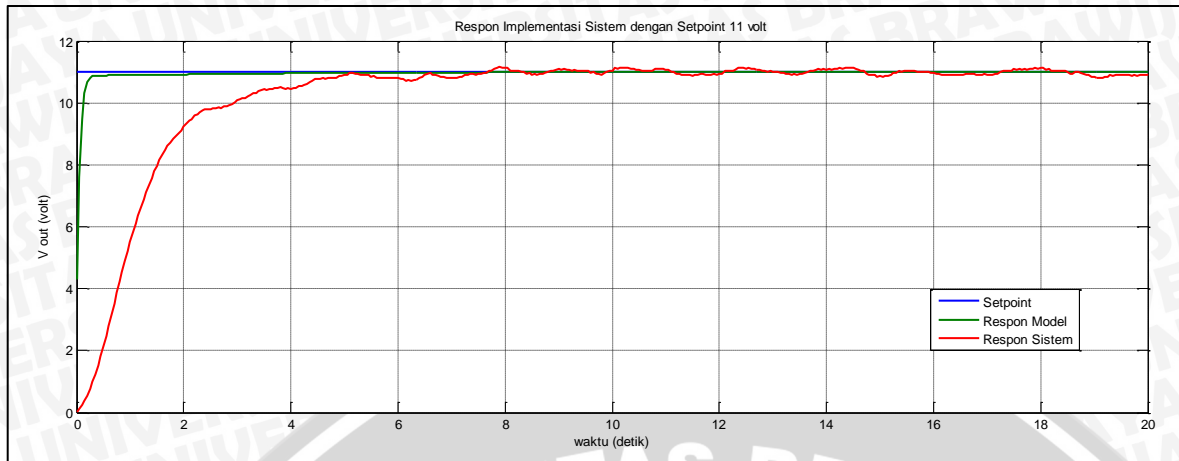
Perubahan parameter struktur PI-D dengan *setpoint* 10 volt ditunjukkan pada Gambar 4.15.



Gambar 4.15 Parameter struktur PI-D dengan *setpoint* 10 volt

Dari Gambar 4.15 dapat diketahui bahwa K_p dan K_i meningkat lalu konstan ketika memasuki *steady state*, sedangkan nilai K_d menurun lalu konstan ketika memasuki *steady state*.

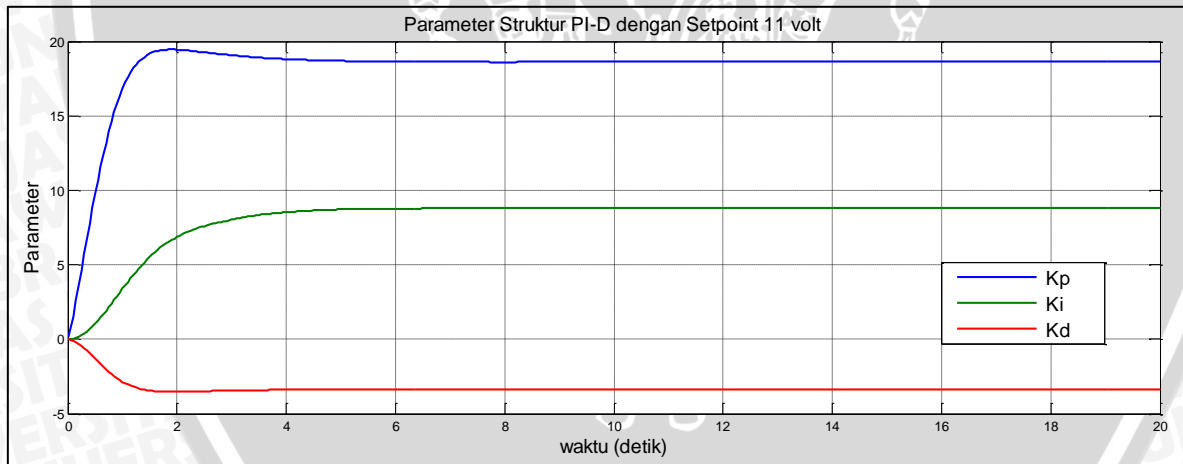
Respon implementasi sistem dengan *setpoint* 11 volt ditunjukkan pada Gambar 4.16.



Gambar 4.16 Respon implementasi sistem dengan *setpoint* 11 volt

Dari Gambar 4.16 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 11 volt memiliki nilai *settling time* sebesar 3,6 detik, *error steady state* sebesar 0,9% dan tidak memiliki *overshoot*.

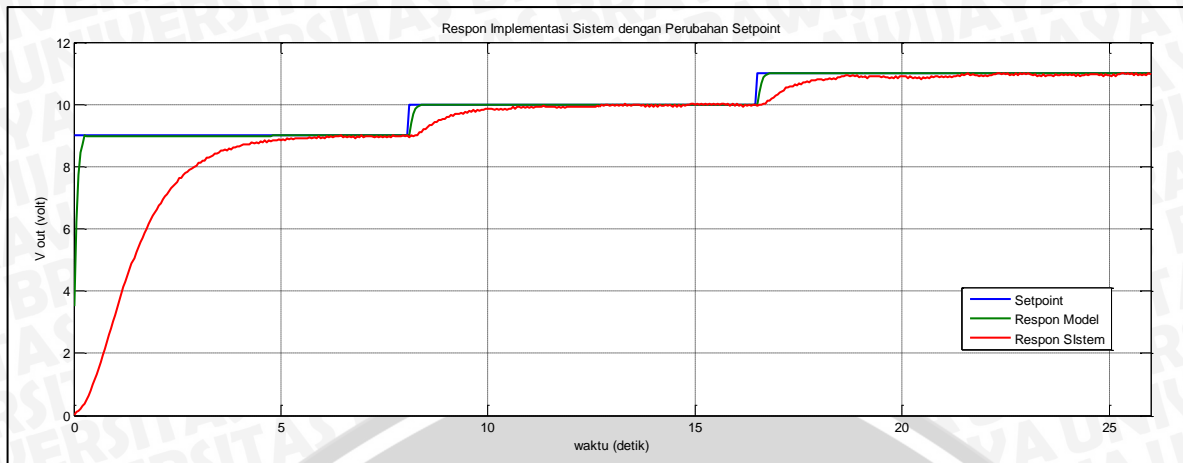
Perubahan parameter struktur PI-D dengan *setpoint* 11 volt ditunjukkan pada Gambar 4.17.



Gambar 4.17 Parameter struktur PI-D dengan *setpoint* 11 volt

Dari Gambar 4.17 dapat diketahui bahwa K_p dan K_i meningkat lalu konstan ketika memasuki *steady state*, sedangkan nilai K_d menurun lalu konstan ketika memasuki *steady state*.

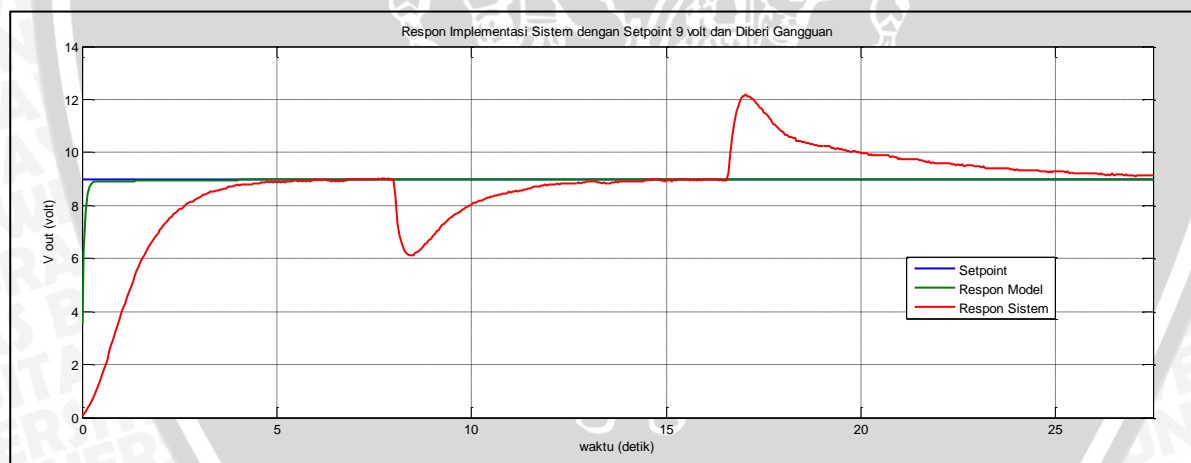
Respon implementasi sistem dengan perubahan *setpoint* ditunjukkan pada Gambar 4.18.



Gambar 4.18 Respon implementasi sistem dengan perubahan *setpoint*

Dari Gambar 4.18 dapat diketahui bahwa respon implementasi sistem dengan perubahan *setpoint* memiliki nilai *settling time* sebesar 3,65 detik untuk menuju *setpoint* 9 volt, 0,75 detik untuk menuju *setpoint* 10 volt, dan 0,65 detik untuk menuju *setpoint* 11 volt.

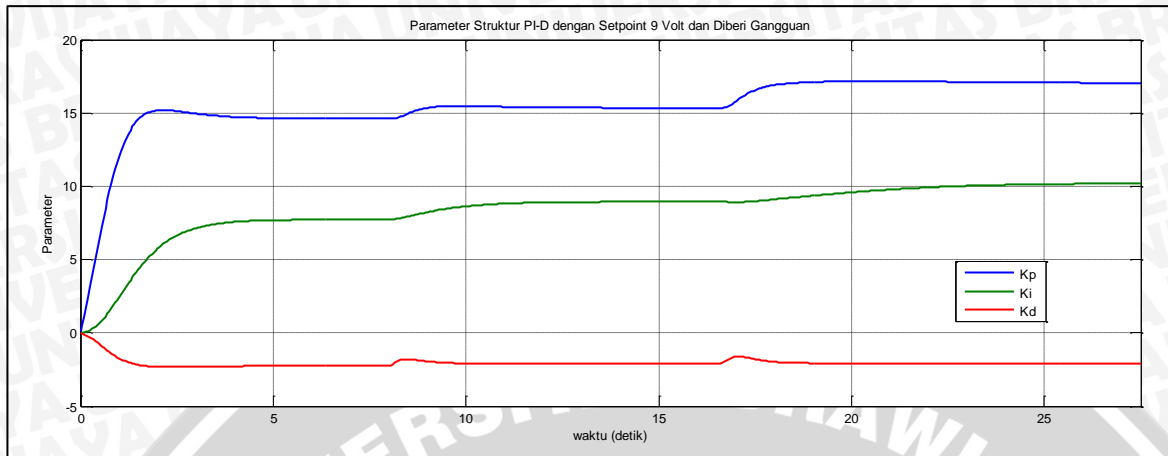
Gangguan yang diberikan untuk sistem dengan *setpoint* 9 volt adalah 4 buah lampu bohlam 2,5 volt yang disusun secara seri. Respon implementasi sistem dengan *setpoint* 9 volt dan diberi gangguan ditunjukkan pada Gambar 4.19.



Gambar 4.19 Respon implementasi sistem dengan *setpoint* 9 volt dan diberi gangguan

Dari Gambar 4.19 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 9 volt memiliki nilai *recovery time* sebesar 2,65 detik ketika gangguan diberikan, dan 6,125 detik ketika gangguan dihilangkan.

Perubahan parameter struktur PI-D dengan *setpoint* 9 volt dan diberi gangguan ditunjukkan pada Gambar 4.20.



Gambar 4.20 Parameter struktur PI-D dengan *setpoint* 9 volt dan diberi gangguan

Dari Gambar 4.20 dapat diketahui bahwa saat sistem diberikan gangguan maka nilai K_p , K_i , dan K_d akan beradaptasi hingga respon sistem mencapai *steady state*.

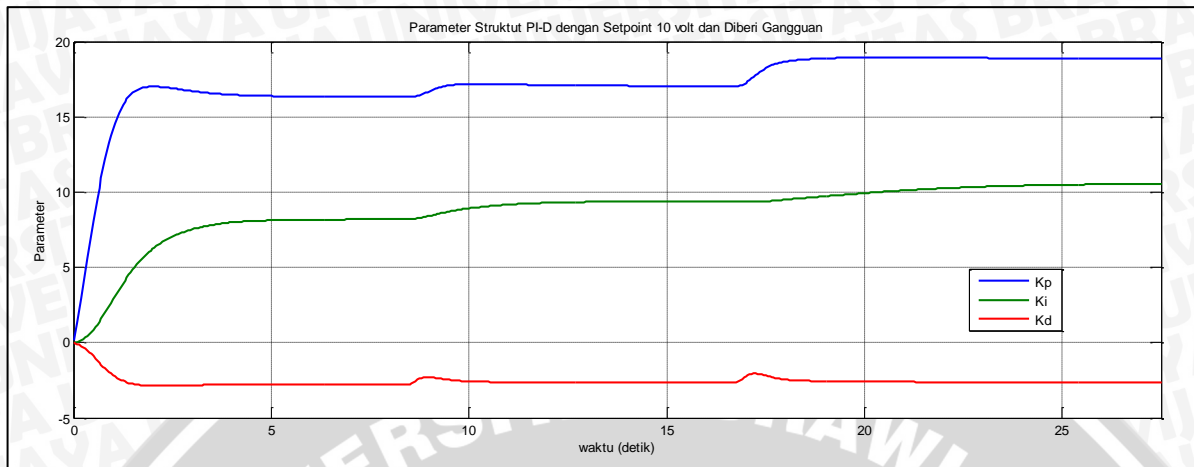
Gangguan yang diberikan untuk sistem dengan *setpoint* 10 volt adalah 5 buah lampu bohlam 2,5 volt yang disusun secara seri. Respon implementasi sistem dengan *setpoint* 10 volt dan diberi gangguan ditunjukkan pada Gambar 4.21.



Gambar 4.21 Respon implementasi sistem dengan *setpoint* 10 volt dan diberi gangguan

Dari Gambar 4.21 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 10 volt memiliki nilai *recovery time* sebesar 2,626 detik ketika gangguan diberikan, dan 5,55 detik ketika gangguan dihilangkan.

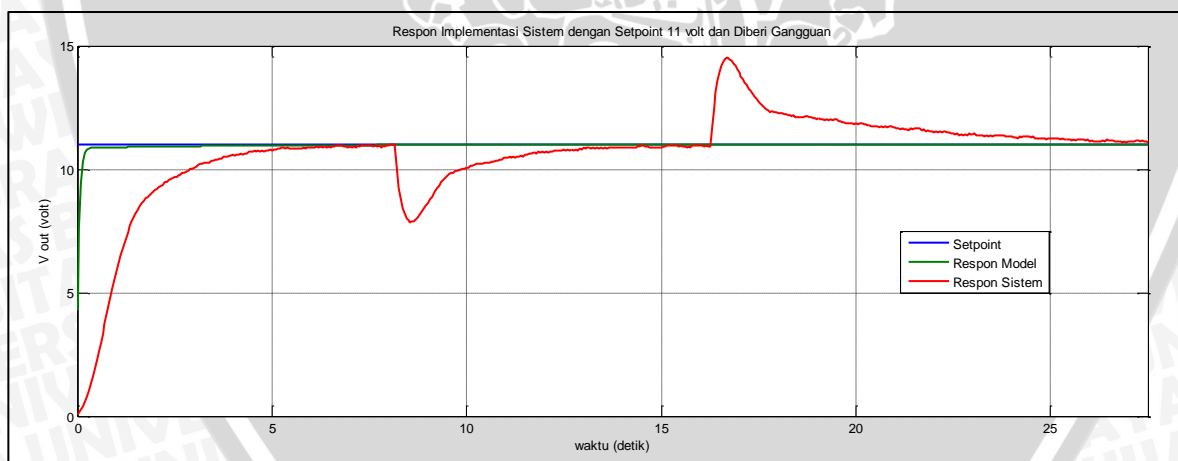
Perubahan parameter struktur PI-D dengan *setpoint* 10 volt dan diberi gangguan ditunjukkan pada Gambar 4.22.



Gambar 4.22 Parameter struktur PI-D dengan *setpoint* 10 volt dan diberi gangguan

Dari Gambar 4.22 dapat diketahui bahwa saat sistem diberikan gangguan maka nilai K_p , K_i , dan K_d akan beradaptasi hingga respon sistem mencapai *steady state*.

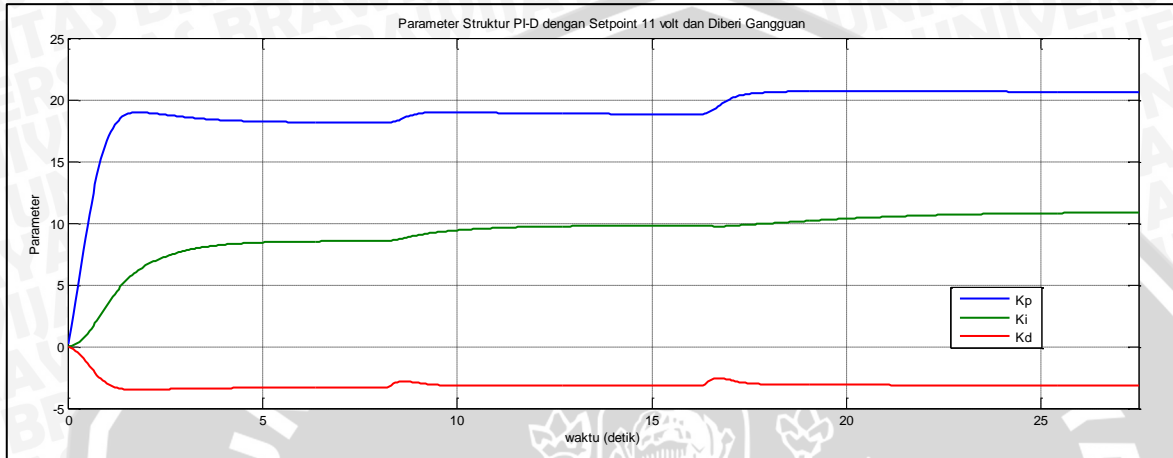
Gangguan yang diberikan untuk sistem dengan *setpoint* 11 volt adalah 5 buah lampu bohlam 2,5 volt yang disusun secara seri. Respon implementasi sistem dengan *setpoint* 11 volt dan diberi gangguan ditunjukkan pada Gambar 4.23.



Gambar 4.23 Respon implementasi sistem dengan *setpoint* 11 volt dan diberi gangguan

Dari Gambar 4.23 dapat diketahui bahwa respon implementasi sistem dengan *setpoint* 10 volt memiliki nilai *recovery time* sebesar 2,375 detik ketika gangguan diberikan, dan 5,1 detik ketika gangguan dihilangkan.

Perubahan parameter struktur PI-D dengan *setpoint* 10 volt dan diberi gangguan ditunjukkan pada Gambar 4.24.



Gambar 4.24 Parameter Struktur PI-D dengan *setpoint* 11 volt dan diberi gangguan

Dari Gambar 4.24 dapat diketahui bahwa saat sistem diberikan gangguan maka nilai K_p , K_i , dan K_d akan beradaptasi hingga respon sistem mencapai *steady state*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Berdasarkan hasil identifikasi *plant*, didapatkan fungsi alih motor generator DC 73411 adalah $\frac{Y(s)}{U(s)} = \frac{5,088}{s^2+8,316s+7,057}$ dan telah divalidasi.
2. Berdasarkan hasil simulasi sistem, didapatkan nilai *gain* parameter yaitu $\gamma_p = 0,195$, $\gamma_i = 0,07$, dan $\gamma_d = 0,08$.
3. Berdasarkan hasil implementasi sistem, respon sistem dengan *setpoint* 9 volt, 10 volt, dan 11 volt tidak memiliki *overshoot*, memiliki nilai *settling time* kurang dari 4 detik dengan nilai masing-masing adalah 3,55 detik, 3,575 detik, dan 3,6 detik. Sedangkan nilai *error steady state* kurang dari 5% dengan nilai masing-masing yaitu 0,7%, 0,8%, dan 0,9%.
4. Berdasarkan hasil implementasi sistem dengan perubahan nilai *setpoint*, respon sistem memiliki nilai *settling time* sebesar 3,65 detik ketika menuju 9 volt, 0,75 detik ketika menuju 10 volt, dan 0,65 detik ketika menuju 11 volt.
5. Berdasarkan hasil implementasi sistem saat diberi gangguan, respon sistem dengan *setpoint* 9 volt, 10 volt, dan 11 volt memiliki nilai *recovery time* masing-masing yaitu 2,65 detik, 2,626 detik, dan 2,375 detik. Sedangkan saat gangguan dihilangkan, respon sistem memiliki nilai *recovery time* masing-masing yaitu 6,12 detik, 5,55 detik, dan 5,1 detik.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan mengimplementasikan motor generator DC sebagai aktuator seperti pemanas pada inkubator atau proses pemasakan keju dan coklat.



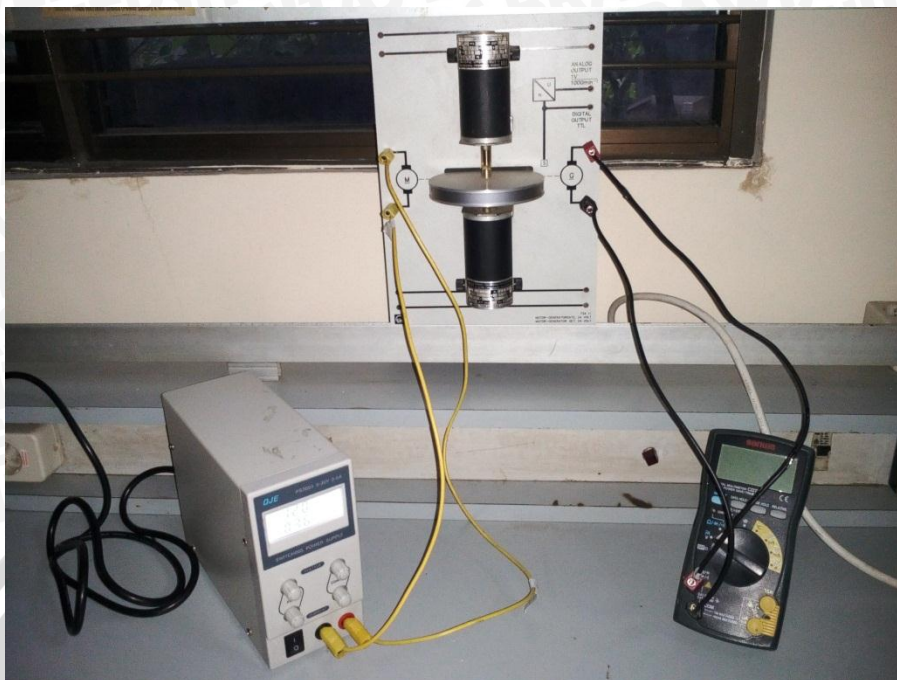
DAFTAR PUSTAKA

- Amaliyah, Zakiyah. 2016. *Desain Kontroler Struktur Output Feedback Control dengan Pole Placement pada Pengontrolan Tegangan Output Generator DC*. Skripsi. Tidak Dipublikasikan. Malang: Universitas Brawijaya.
- Ang, K.H., Chong G.C.Y dan Li,Y. 2007. *PID control system analysis, design, and technology*. IEEE: Transaction on Control Systems Technology.
- Astrom, K.J. dan B. Wittenmark. 1995. *Adaptive Control*. Addison-Wesley Publishing Company, Inc. USA.
- Bobal, V, J., Bohm, J., Fessl dan Machacek. 2005. *Digital Self-Tuning Controller*. London: Springer verlag.
- Ibrahim, Mochammad Hesa. 2016. *Penggunaan Teknik Model Reference Adaptive Control pada Tuning Pengontrolan PI pada Sistem Kontrol Kecepatan Sepeda*. Skripsi. Tidak Dipublikasikan. Malang: Universitas Brawijaya.
- Lanang, Sang., Sardono Sarwito, dan Indra Ranu Kusuma. 2013. *Analisa Pengaruh Beban Induktif dan Resistif pada Generator Induksi pada Pembangkit Listrik Tenaga Gelombang Laut*. Jurnal Teknik Perkapalan ITS Surabaya.
- Niko, Ardian S. 2010. *Analisis Perbandingan Karakteristik Berbeban Generator Arus Searah Penguatan Bebas dengan Generator Arus Searah Penguat Shunt*. Skripsi. Tidak Dipublikasikan. Medan: Universitas Sumatera Utara.
- Petruzella, Frank D. 2001. *Elektronik Industri*. Yogyakarta: Andi Yogyakarta.
- Prabariyanto, Bayu. 2016. *Metode Root Locus Untuk Mencari Parameter PID pada Pengendalian Kecepatan Motor DC D-6759 Menggunakan Arduino Mega 2560*. Skripsi. Tidak Dipublikasikan. Malang: Universitas Brawijaya.
- Rozi, Fakhrrur. 2015. *Tuning kontrol PI dengan menggunakan teknik Model Reference Adaptive Control (MRAC) pada sistem kontrol kecepatan motor DC*. Skripsi. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Rusli, M. 2015. *Dasar Teknik Kontrol*. Malang: Penerbit UB Press.
- Xiong, Ai dan Youngkun Fan. 2007. *Application of a PID Controller using MRAC Techniques for Control of the DC Electromotor Driver*. IEEE: International Conference on Mechatronics and Automation.

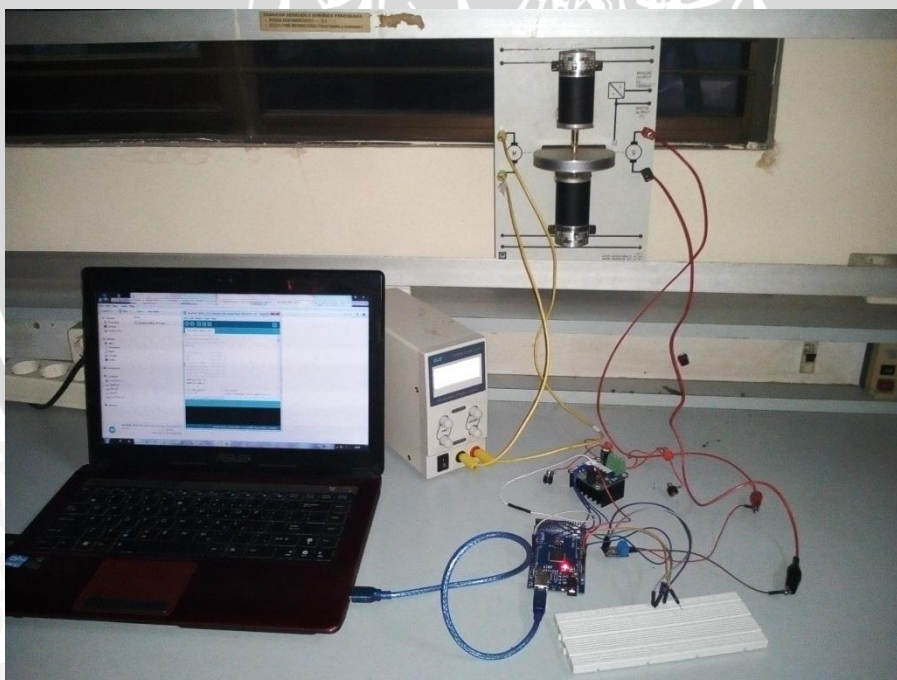


Lampiran 1
Foto Alat

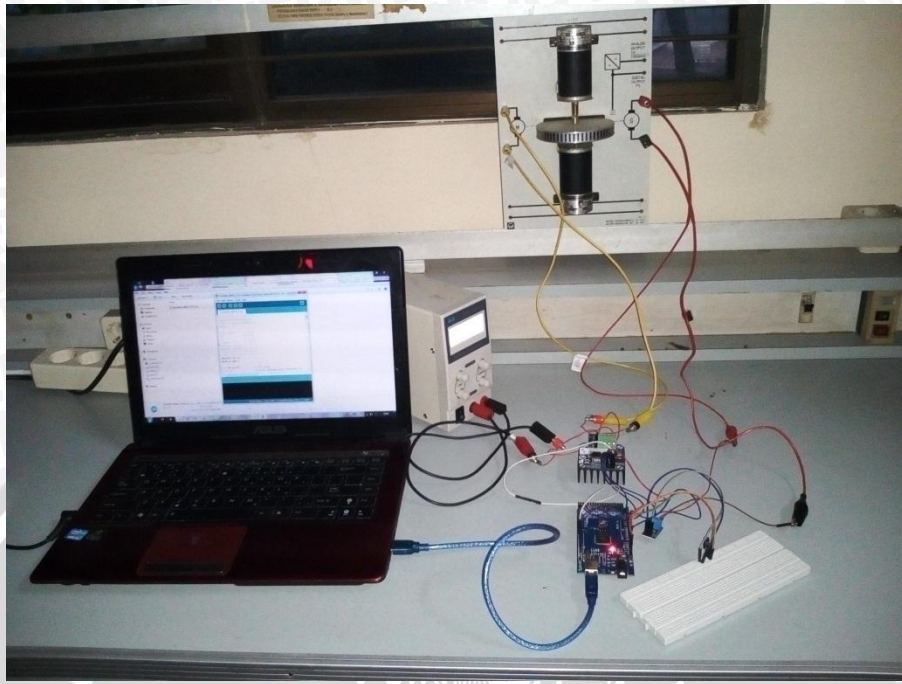




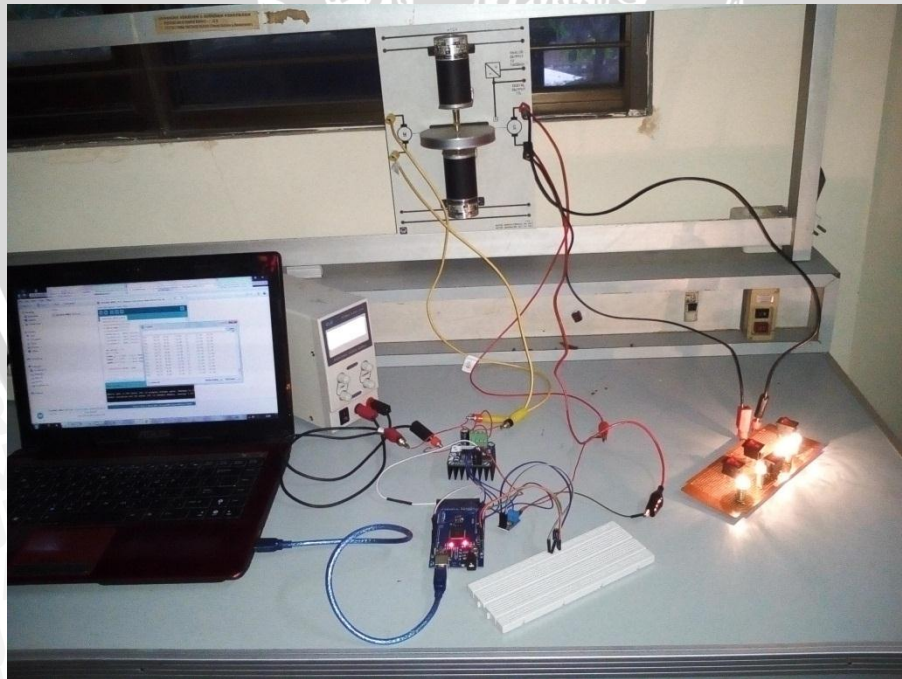
Karakterisasi Motor Generator DC 73411



Karakterisasi Sensor Tegangan dan Driver Motor



Pencarian Fungsi Alih Plant



Pengujian Sistem



Lampiran 2
Diagram Blok



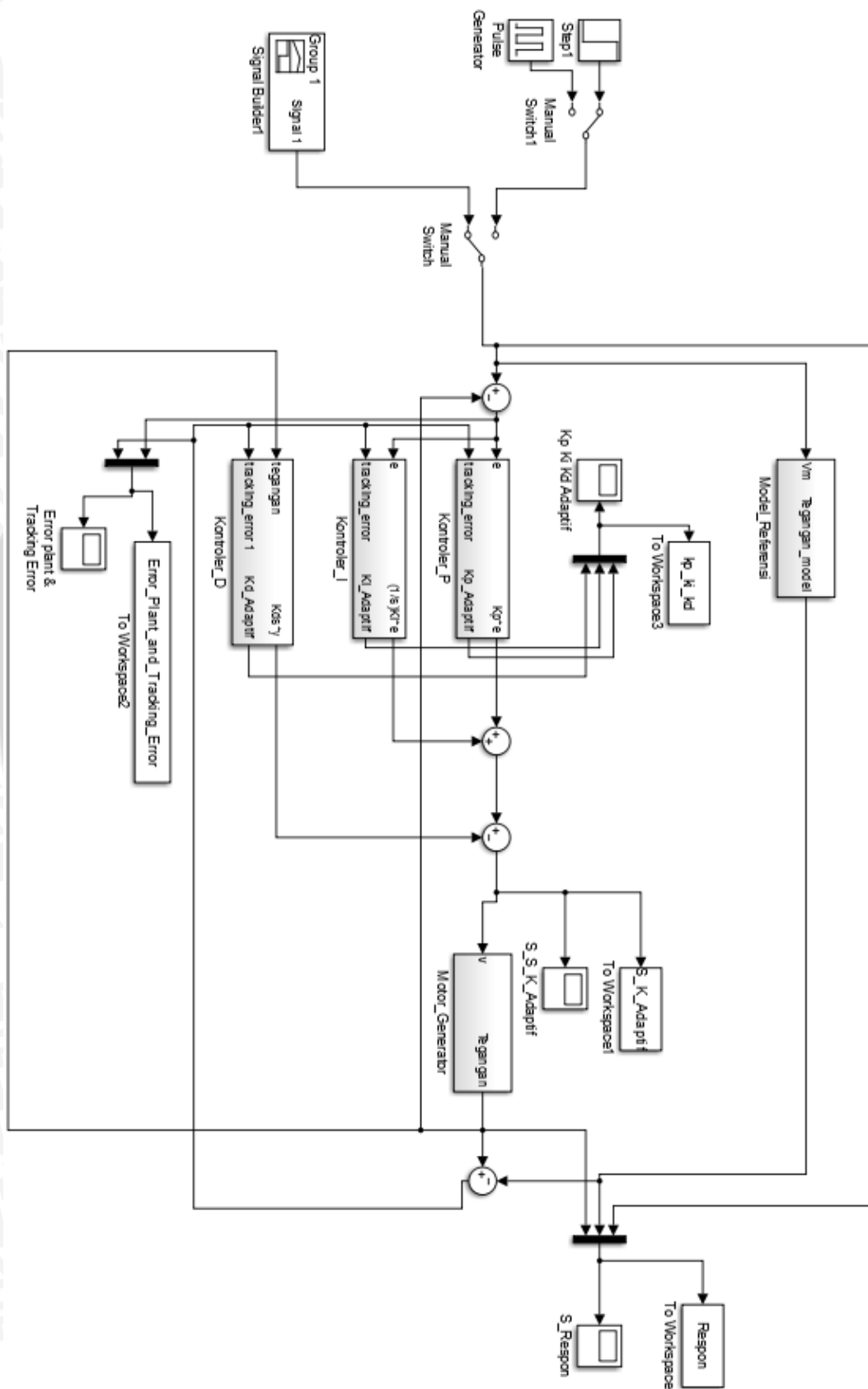


Diagram Blok Simulink Sistem



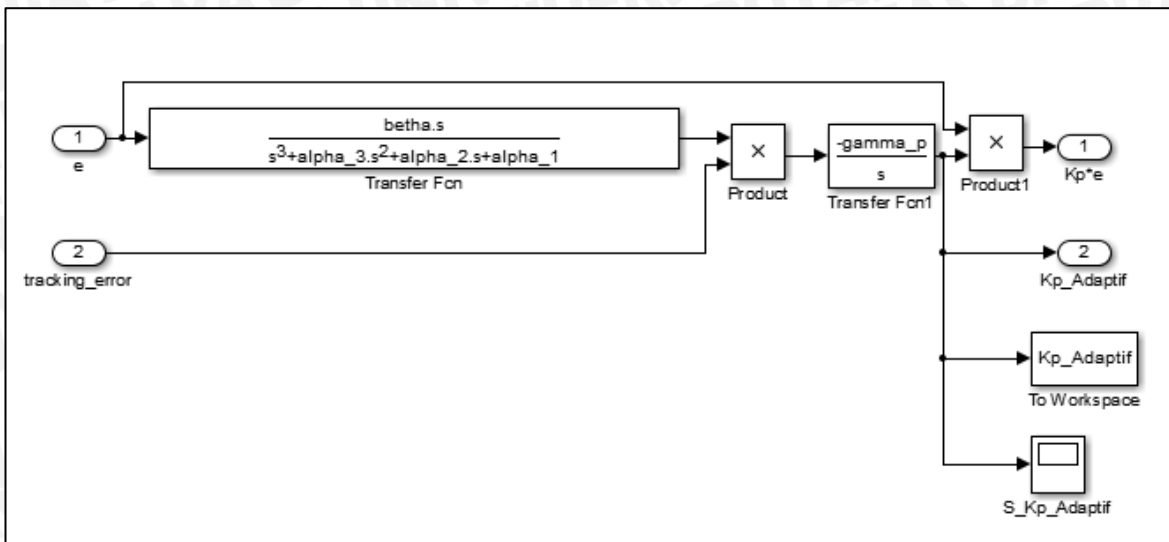


Diagram Blok Simulink Subsystem Kontroler P

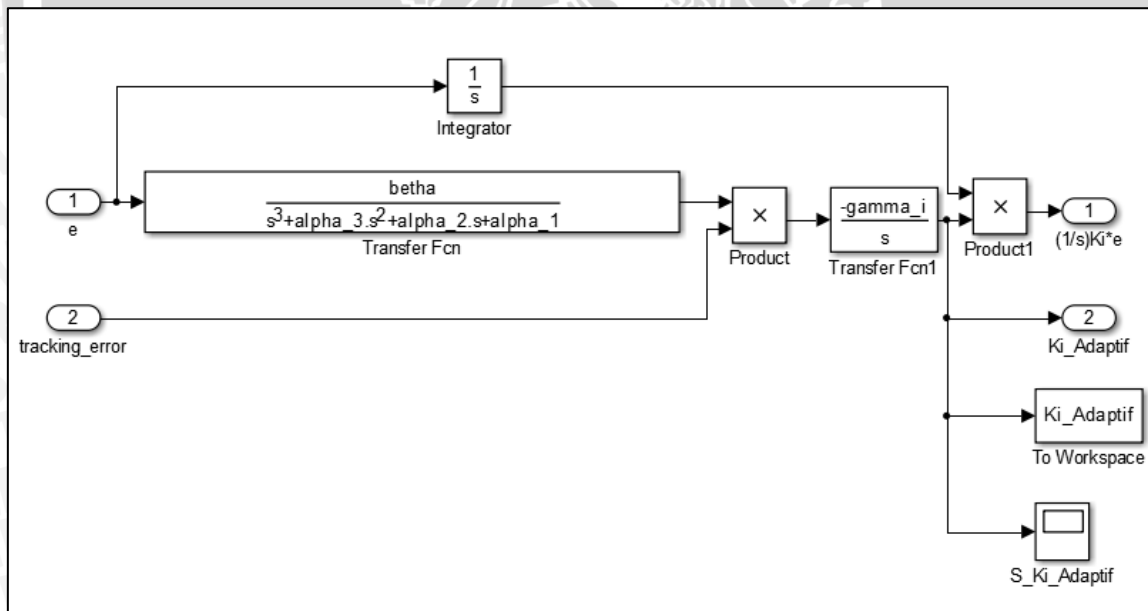


Diagram Blok Simulink Subsystem Kontroler I

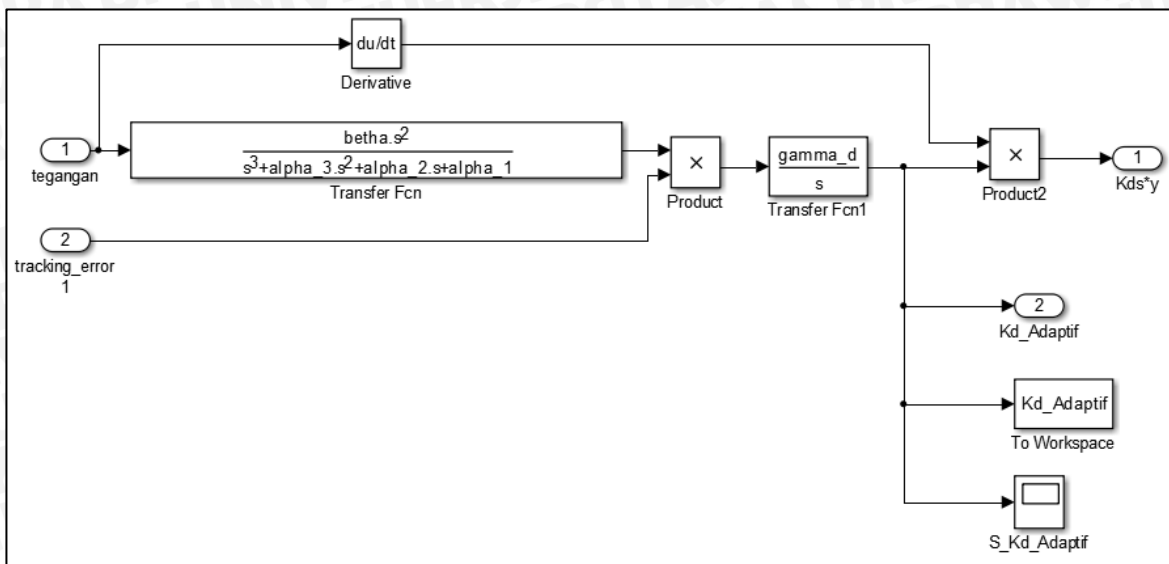


Diagram Blok Subsystem Kontroler D

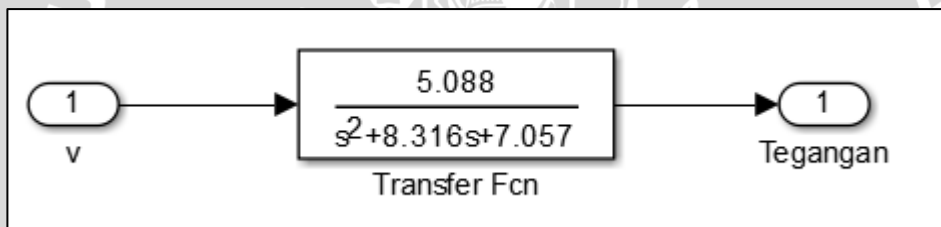


Diagram Blok Simulink Subsystem Motor Generator DC 73411

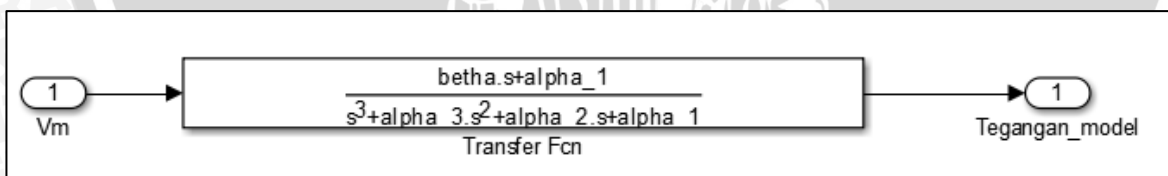


Diagram Blok Simulink Subsystem Model Referensi

Lampiran 3
Listing Program



Listing program M-File Simulasi

```

clear all;
clc;
close all;

%Penentuan Koefisian Model Referensi
betha = 1052.3;
alpha_1 = 379.5;
alpha_2 = 1079.55;
alpha_3 = 50.79;

%parameter1
%gamma_p = 0.001;
%gamma_i = 0.001;
%gamma_d = 0.001;

%parameter2
%gamma_p = 0.005;
%gamma_i = 0.005;
%gamma_d = 0.005;

%parameter3
%gamma_p = 0.07;
%gamma_i = 0.069;
%gamma_d = 0.01;

%parameter 4
%gamma_p = 0.0004;
%gamma_i = 0.0004;
%gamma_d = 0.0004;

%parameter 5
gamma_p = 0.195;
gamma_i = 0.07;
gamma_d = 0.08;

%parameter 6
%gamma_p = 0.069;
%gamma_i = 0.055;
%gamma_d = 0.01;

%Parameter 7
%gamma_p = 0.059;
%gamma_i = 0.01;
%gamma_d = 0.01;

%parameter 8
%gamma_p = 0.069;
%gamma_i = 0.55;
%gamma_d = 0.01;

%parameter 9
%gamma_p = 0.069;
%gamma_i = 0.000055;
%gamma_d = 0.0008;

```



Listing program Sistem Kontrol Adaptive dengan MRAC

```

/*****
Sensor Tegangan    ==> A0
PWM                ==> PIN5
_*****/

#define pwm 5
int set_point;      // Set point
float y;           // Nilai tegangan Motor Generator DC
float last1_y;     //tegangan motor sebelumnya
float last2_y;     //tegangan motor sebelum sebelumnya
float error;       // set_point - y
float last1_error; // error sebelumnya
int last1_set_point; // Set point sebelumnya
float ym;          // Nilai keluaran Model referensi
float tracking_error; // y - ym
float last1_ym;    // Nilai keluaran Model referensi sebelumnya
float last2_ym;    // Nilai keluaran Model referensi sebelumnya sebelumnya
float last3_ym;    // Nilai keluaran Model referensi sebelumnya sebelumnya sebelumnya
float model_p;     // Nilai keluaran Model di P
float last1_model_p; // Nilai keluaran Model di P sebelumnya
float last2_model_p; // Nilai keluaran Model di P sebelumnya sebelumnya
float last3_model_p; // Nilai keluaran Model di P sebelumnya sebelumnya sbelumnya
float model_i;     // Nilai keluaran Model di I
float last1_model_i; // Nilai keluaran Model di I sebelumnya
float last2_model_i; // Nilai keluaran Model di I sebelumnya sebelumnya
float last3_model_i; // Nilai keluaran Model di I sebelumnya sebelumnya sbelumnya
float model_d;     // Nilai keluaran Model di D
float last1_model_d; // Nilai keluaran Model di D sebelumnya
float last2_model_d; // Nilai keluaran Model di D sebelumnya sebelumnya
float last3_model_d; // Nilai keluaran Model di D sebelumnya sebelumnya sbelumnya
long int pulsa;   // pulsa rotary encoder
float gamma_p;    // parameter kendali P
float gamma_i;    // parameter kendali I
float gamma_d;    // parameter kendali D
float Ts;         // Time Sampling Model Referensi
float A, B, C, D, E, F, J, G; // Konstanta Model
float pwmMotor;   // PWM Motor
float T, U;       // Batas atas dan bawah sinyal kontrol
float Prop;       // Var sementara P
float Intg;       // Var sementara I
float Div;        // Var sementara D
double MV;        // Nilai keuaran PI-D
double last1_MV; // Nilai keuaran PI-D sebelumnya
float Kp;         // Nilai konstanta Kp
float last1_Kp;  // Nilai konstanta Kp sebelumnya
float Ki;        // Nilai konstanta Ki
float last1_Ki;  // Nilai konstanta Ki sebelumnya
float Kd;        // Nilai konstanta Kd
float last1_Kd;  // Nilai konstanta Kd sebelumnya
float integrator; // Nilai integrator Kontroler I
float last1_integrator; // Nilai integrator Kontroler I sebelumnya
float diferensiator; // Nilai diferensiator Kontroler D
float last1_diferensiator; // Nilai diferensiator Kontroler D sebelumnya
float R, S, L;   // Hasil perkalian model dengan tracking error
int as,ap;

int cnt=0;
void setup()
{

```

```

pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
pinMode(pwm,OUTPUT);
pinMode(led,OUTPUT);
set_point = 11;
Ts = 0.05;
gamma_p = 0.195; //gain adaptasi P
gamma_i = 0.07; // gain adaptasi I
gamma_d = 0.08; // gain adaptasi D

//Konstanta Model
float a;
float b;
float c;
float d;
a = 952.3;
b = 279.8;
c = 53.79;
d = 979.8;

A = 3+((2*c)*Ts)+(d*Ts*Ts);
B = 3+(c*Ts);
C = 1;
D = (a*Ts*Ts)+(b*Ts*Ts*Ts);
E = (a*Ts*Ts);
F = (a*Ts*Ts*Ts);
J = (a*Ts);
G = 1+(c*Ts)+(d*Ts*Ts)+(b*Ts*Ts*Ts);

//inisialisasi awal
last1_y = 0; last2_y = 0;
last1_ym = 0; last2_ym = 0; last3_ym = 0; last1_set_point = 0;
last1_error = 0; last1_integrator = 0; last1_diferensiator=0;
last1_model_p = 0; last2_model_p = 0; last3_model_p = 0;
last1_model_i = 0; last2_model_i = 0; last3_model_i = 0;
last1_model_d = 0; last2_model_d = 0; last3_model_d = 0;
last1_Kp = 0; last1_Ki = 0; last1_Kd=0;

noInterrupts();
TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;

OCR1A = 3124; // compare match register 16MHz/256/50Hz/50ms
TCCR1B |= (1 << WGM12); // CTC mode
TCCR1B |= (1 << CS12); // 256 prescaler
TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt

TCCR2A = 0b00000010;
TCCR2B = 0b00000111;
TIMSK2 |= (1 << OCIE2A); // enable timer compare interrupt
OCR2A = 156; // compare match register 16MHz/64/25KHz/10ms

TCCR0A = 0b00000010;
TCCR0B = 0b00000101;
TIMSK0 |= (1 << OCIE0A); // enable timer compare interrupt
OCR0A = 156; // compare match register 16MHz/64/25KHz/10ms

```



```

attachInterrupt(0, hitung_pulsa, FALLING);
interrupts(); // enable all interrupts
Serial.begin(9600);
}

ISR(TIMER2_COMPA_vect) // timer compare interrupt service routine
{
  if(as==5)
  {
    as=0;
    sensor();
  }
  as++;
}

ISR(TIMER1_COMPA_vect) // timer compare interrupt service routine
{
  MRAC();
}

ISR(TIMER0_COMPA_vect) // timer compare interrupt service routine
{
  if(ap==5)
  {
    ap=0;
    error = set_point - y;
    tracking_error = y - ym;
    Tuning_Kp();
    Integrator();
    Tuning_Ki();
    Diferensiator();
    Tuning_Kd();
    Kontroler();

    Serial.print(cnt);
    Serial.print("\t");
    Serial.print(Kp);
    Serial.print("\t");
    Serial.print(Ki);
    Serial.print("\t ");
    Serial.print(Kd);
    Serial.print("\t ");
    Serial.print(set_point);
    Serial.print("\t ");
    Serial.print(ym);
    Serial.print("\t ");
    Serial.print(y);
    Serial.print("\n");
    cnt++;
  }
  ap++;
}

void MRAC()
{
  ym = ((A/G)*last1_ym) - ((B/G)*last2_ym) + ((C/G)*last3_ym) + ((D/G)*set_point) -
  ((E/G)*last1_set_point); // Model Referensi
  last1_set_point = set_point;
  last3_ym = last2_ym;
  last2_ym = last1_ym;
}

```



```

last1_ym = ym;
}

void sensor()
{
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  y = sensorValue * (25.22 / 1023.0);
}

void Tuning_Kp()
{
  model_p = ((A/G)*last1_model_p) - ((B/G)*last2_model_p) + ((C/G)*last3_model_p) + ((E/G)*error) -
  ((E/G)*last1_error); //Model K. Proporsional
  last3_model_p = last2_model_p;
  last2_model_p = last1_model_p;
  last1_model_p = model_p;
  last1_error = error;
  R = model_p*tracking_error;
  Kp = last1_Kp + ((-gamma_p)*Ts*R);
  last1_Kp = Kp;
}

void Tuning_Ki()
{
  model_i = ((A/G)*last1_model_i) - ((B/G)*last2_model_i) + ((C/G)*last3_model_i) + ((F/G)*error);
  //Model K. Integral
  last3_model_i = last2_model_i;
  last2_model_i = last1_model_i;
  last1_model_i = model_i;
  S = model_i*tracking_error;
  Ki = last1_Ki + ((-gamma_i)*Ts*S);
  last1_Ki = Ki;
}

void Integrator()
{
  integrator = last1_integrator + (Ts * error); //Integrator 1/s
  last1_integrator = integrator;
}

void Tuning_Kd()
{
  model_d = ((A/G)*last1_model_d) - ((B/G)*last2_model_d) + ((C/G)*last3_model_d) + ((J/G)*y) -
  ((2*J)/G)*last1_y + ((J/G)*last2_y); //Model K. Diferensial
  last3_model_d = last2_model_d;
  last2_model_d = last1_model_d;
  last1_model_d = model_d;
  last2_y = last1_y;
  last1_y = y;
  L = model_d*tracking_error;
  Kd = last1_Kd + ((gamma_d)*Ts*L);
  last1_Kd = Kd;
}

void Diferensiator()
{
  diferensiator = ((1/Ts)*y) - ((1/Ts)*last1_y); //diferensiator 1/s
  //last1_y = y;
}

```

```
void Kontroler()
{
  Prop = Kp * error;
  Intg = Ki * integrator;
  Div = Kd * diferensiator;
  MV = ((Prop + Intg)-Div);

  if (MV<0)MV = 0;
  pwmMotor = MV;

  if (pwmMotor<0)pwmMotor = 0;
  else if (pwmMotor>255)pwmMotor = 255;
  else pwmMotor = pwmMotor;
}

void loop()
{
  analogWrite(pwm,pwmMotor);
}

void hitung_pulsa()
{
  pulsa++;
}
```

UNIVERSITAS BRAWIJAYA





Lampiran 4
Datasheet



