



**PENGENDALIAN TEGANGAN *OUTPUT* GENERATOR DC  
PENGUATAN TERPISAH MENGGUNAKAN  
KONTROL *HYBRID* PID-FUZZY  
BERBASIS ARDUINO UNO R3**

**SKRIPSI**

**oleh**

**Aditiya Chandra Purnama  
NIM 121910201096**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**



**PENGENDALIAN TEGANGAN *OUTPUT* GENERATOR DC  
PENGUATAN TERPISAH MENGGUNAKAN  
KONTROL *HYBRID* PID-FUZZY  
BERBASIS ARDUINO UNO R3**

**SKRIPSI**

**diajukan guna melengkapi skripsi dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi S1 Teknik Elektro  
dan mencapai gelar Sarjana Teknik**

**oleh**

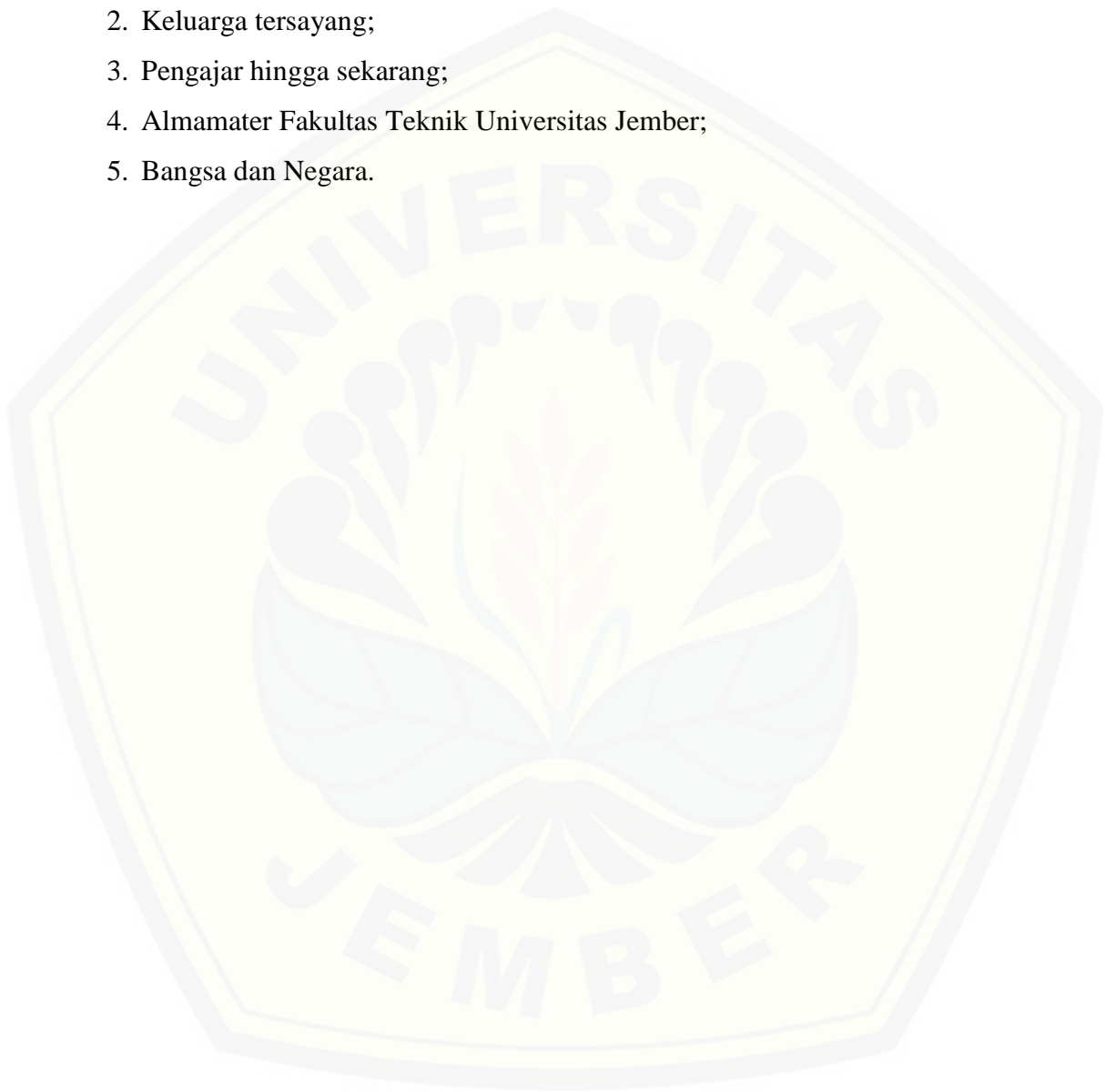
**Aditiya Chandra Purnama  
NIM 121910201096**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

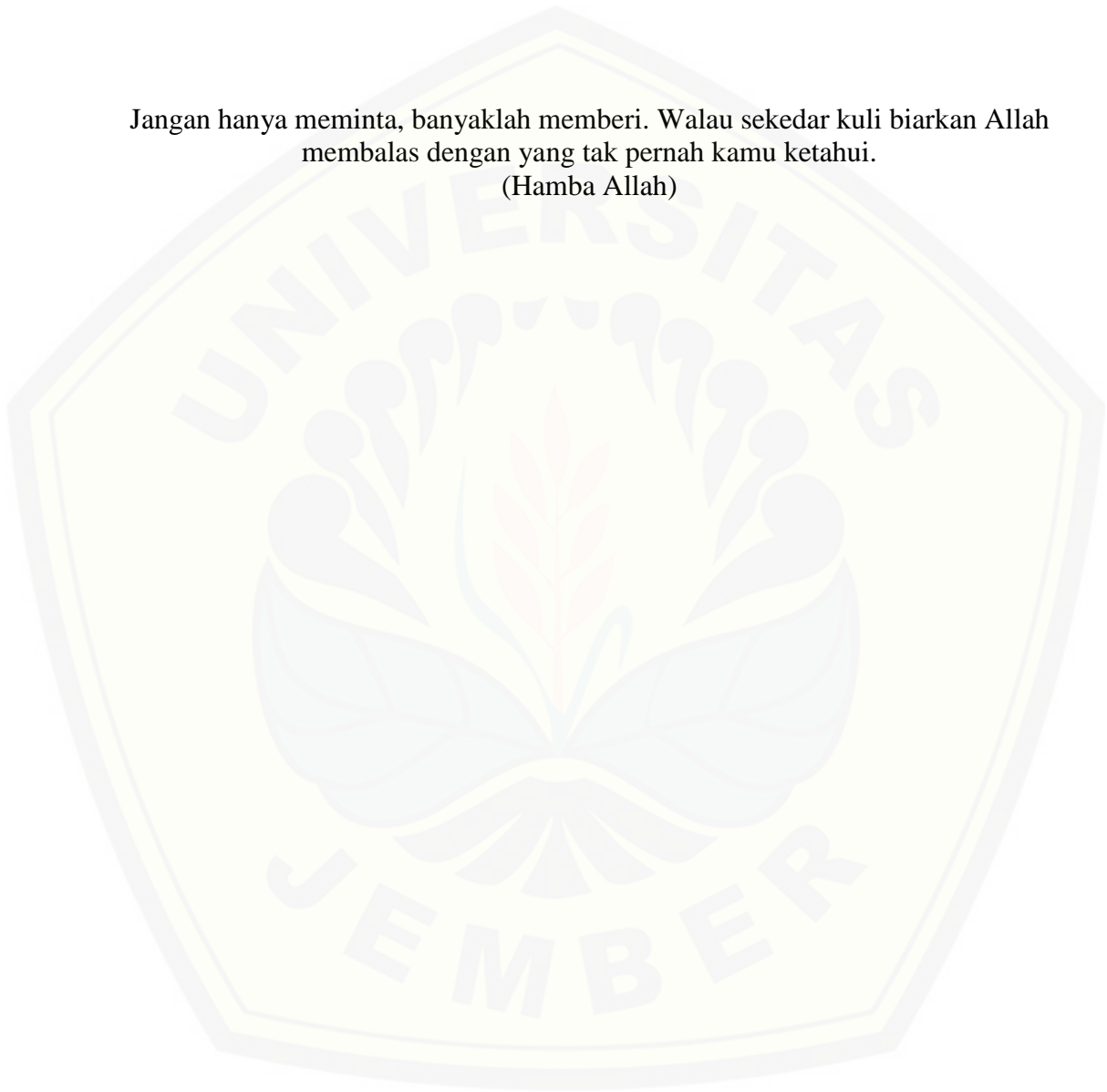
1. Bapak Ibu Tercinta;
2. Keluarga tersayang;
3. Pengajar hingga sekarang;
4. Almamater Fakultas Teknik Universitas Jember;
5. Bangsa dan Negara.



**MOTO**

Prinsip kuat, pemikiran hebat, jiwa anti sambat dan berjuta sahabat. Semua ini ada dalam wadah yang tepat untuk segala nikmat.  
(Hamba Allah)

Jangan hanya meminta, banyaklah memberi. Walau sekedar kuli biarkan Allah membalas dengan yang tak pernah kamu ketahui.  
(Hamba Allah)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Aditiya Chandra Purnama

NIM : 121910201096

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2016

Yang menyatakan,

Aditiya Chandra Purnama  
NIM 121910201096

**SKRIPSI**

**Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah  
Menggunakan Kontrol *Hybrid* PID-Fuzzy  
Berbasis Arduino UNO R3**

oleh

Aditiya Chandra Purnama  
NIM 121910201096

Pembimbing

Dosen Pembimbing Utama : Ir. Widyono Hadi, M.T.

Dosen Pembimbing Anggota : Dr. Triwahju Hardianto, S.T., M.T.

**PENGESAHAN**

Skripsi berjudul “Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3” telah diuji dan disahkan pada:

hari, tanggal : Kamis, Juni 2016

tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Ketua,

Ir. Widyono Hadi, M.T.  
NIP 19610414 198902 1 001

Anggota I,

Suprihadi Prasetyono, S.T., M.T.  
NIP 19700404199601 1 001

Sekretaris,

Dr. Triwahju Hardianto, S.T., M.T.  
NIP 19700826 199702 1 001

Anggota II,

Dedy Kurnia Setiawan, S.T., M.T.  
NIP 19800610 200501 1 003

Mengesahkan  
Dekan Fakultas Teknik,

Dr.Ir. Entin Hidayah, M.UM.  
NIP 19661215 199503 2 001

Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3

**Aditya Chandra Purnama**

*Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember*

**ABSTRAK**

Generator DC merupakan suatu mesin listrik yang mengubah energi mekanis menjadi energi listrik. Pada generator DC memiliki karakteristik ketika diberi beban maka *output* tegangan yang dihasilkan mengalami *drop* tegangan. Tujuan dari penelitian ini adalah menerapkan kontrol *hybrid* PID-Fuzzy untuk didapatkan hasil kestabilan *output* tegangan generator DC sebesar 220V. Kontrol *hybrid* PID-fuzzy adalah gabungan dari kontrol PID dan kontrol fuzzy dimana kontrol PID menjadi kontrol utama sedangkan kontrol *fuzzy* sebagai kontrol untuk membantu mendapatkan hasil yang sesuai dengan yang diinginkan, yaitu kestabilan tegangan *output* generator DC sebesar 220V. Variasi beban yang digunakan adalah sebesar 15, 25, 40, 50 dan 65 watt untuk menganalisa hasil kestabilan generator DC menggunakan kontrol *hybrid* PID-Fuzzy. Hasil yang diperoleh dari penelitian *drop* tegangan terbesar mencapai 208 Volt dengan *recovery time* 2.2 s, *error steady-state* 2.27% dan *error* tegangan *output* terhadap *set point* sebesar 1.05% pada pemberian beban 65 watt.

**Kata kunci:** Generator DC, Arduino UNO-R3, *Hybrid* PID-Fuzzy



*Output Voltage Control for Separately Excited DC Generator Using Hybrid PID-Fuzzy Controller Based Arduino UNO R3*

**Aditya Chandra Purnama**

*Department of Electrical Engineering, Faculty of Engineering, University of Jember*

**ABSTRACT**

*DC Generator is an electrical machine that converts mechanical energy to electrical energy. DC generator has characteristics when given load the output voltage resulting drop. The purpose of this study is to apply hybrid PID-Fuzzy controller to obtain the result of the stability of output voltage is 220 V. Hybrid PID-Fuzzy controller is a combination of PID controller and fuzzy logic controller where PID controller into the main controller while fuzzy logic controller as a control to help get result as expected that stability of the generator output voltage is 220V. Variation load used was 15, 25, 40, 50 and 65 watt to analyze the result of stability of the DC generator using hybrid PID-Fuzzy controller. The result of the study is largest drop voltage reached 208 V with recovery time 2.2 s, a steady state error 2.27% and output voltage error of set point 1.05% when load 65 watt.*

**Keywords :** DC Generator, Arduino Uno R3, Hybrid PID-Fuzzy controller

## RINGKASAN

**Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3**; Aditiya Chandra Purnama, 121910201096; 2016; 67 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Generator DC merupakan suatu mesin listrik yang mengubah energi mekanis menjadi energi listrik. Karakteristik generator DC ketika diberi beban *output* tegangan pada generator mengalami *drop* tegangan. Dalam hal ini dibutuhkan kontrol pada arus eksitasi sehingga dapat mengatur tegangan *output* agar stabil. Kontrol yang akan digunakan dalam penelitian ini adalah menggunakan kontrol *hybrid* PID-Fuzzy dimana kontrol utamanya adalah kontrol PID sedangkan kontrol *fuzzy* digunakan sebagai pembantu untuk meminimalkan *overshoot* atau *undershoot* dan memperbaiki *recovery time*.

Dalam menerapkan sistem kontrol ini digunakan arduino UNO R3 sebagai *embedded system*. Arduino ini digunakan dalam penerapan langsung pada pengontrolan generator DC penguatan terpisah.

Komponen pengendali yang digunakan sebagai penunjang adalah *buck converter*, sensor tegangan dan data *logger*. *Buck converter* sebagai pemberi kontrol dari arus eksitasi yang diperlukan untuk mencapai kestabilan generator DC penguatan terpisah dengan mendapatkan perintah dari arduino. Sensor tegangan digunakan sebagai pembacaan tegangan *output* dari generator DC yang akan dijadikan nilai masukan pada arduino. Data *logger* digunakan sebagai penyimpan data pada hasil penelitian yang dilakukan.

Penelitian ini didapatkan hasil dimana penggunaan kontrol *hybrid* PID-Fuzzy dapat sesuai dengan yang diinginkan yaitu menjaga kestabilan generator DC ketika diberi variasi beban pada tegangan *output* sebesar 220V. Pada hasil analisa *drop* tegangan terbesar yaitu sebesar 208 V dengan *recovery time* 2.2 s dan *error steady state* sebesar 1.01%.

## PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ir. Widyono Hadi., M.T. selaku Dosen Pembimbing Utama, Dr. Triwahju Hardianto, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. H.R.B. Moch. Gozali, S.T., M.T. selaku Dosen Penguji I, Dedy Kurnia Setiawan, S.T., M.T. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang sangat membangun demi penyempurnaan skripsi ini;
3. H.R.B. Moch. Gozali, S.T., M.T. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Andi Setiawan, S.T.,M.T. selaku Dosen Pembimbing luar yang telah membimbing selama penulis menjadi mahasiswa;
5. Bambang Supeno, S.T., M.T. selaku Komisi Bimbingan S1 seminar proposal, Dedy Kurnia Setiawan, S.T., M.T. selaku Komisi Bimbingan S1 seminar hasil dan Mohamad Agung Prawira Negara, S.T., M.T. selaku Komisi Bimbingan S1 sidang skripsi yang telah membantu penulisan skripsi secara administratif;
6. Bunda Indah Wulandari dan Ayahanda Hisbullah yang telah memberikan dukungan moril dan materiil serta kasih sayang yang tak terhingga;
7. Adik-adikku tersayang yang telah memberikan motivasi dan semangat;
8. Keluarga Besar Elektro Universitas Jember yang telah mendukung dengan segenap jiwa , cinta dan rasa;
9. Keluarga Besar SATE UJ yang terus ada dan selalu setia;
10. Keluarga besar TK, SD, SMP dan SMA yang telah mendoakan;

11. Keluarga Besar 65 Watt yang terus mendukung baik mental, fisik , materi dan sebagainya;
12. Keluarga Besar Patrang yang selalu ada dan terus berkarya;
13. Medina Nanda Utami yang senantiasa memberikan support dan dukungannya serta doanya;
14. Semua Pihak yang tidak bisa disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juni 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	ii
HALAMAN PERSEMBAHAN .....	iii
HALAMAN MOTO .....	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBINGAN.....	vi
HALAMAN PEMNGESAHAN .....	vii
ABSTRAK .....	viii
ABSTRACT .....	ix
RINGKASAN .....	x
PRAKATA .....	xii
DAFTAR ISI.....	xiv
DAFTAR TABEL .....	xvii
DAFTAR GAMBAR.....	xviii
DAFTAR LAMPIRAN .....	xxi
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang.....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>2</b>
<b>1.3 Batasan Masalah .....</b>	<b>2</b>
<b>1.4 Tujuan.....</b>	<b>2</b>
<b>1.5 Manfaat.....</b>	<b>3</b>
<b>1.6 Sistematika Penelitian .....</b>	<b>3</b>
<b>BAB 2. TINJAUAN PUSTAKA.....</b>	<b>4</b>
<b>2.1 Generator DC Penguatan Terpisah .....</b>	<b>4</b>
2.1.1 Teori kontrol Tegangan <i>Output</i> Generator DC penguatan terpisah .....	6
<b>2.2 Komponen Pengendali.....</b>	<b>7</b>
2.2.1 Arduino uno R3 .....	7
2.2.2 Buck Converter.....	8

2.2.3 Sensor Tegangan .....	13
<b>2.3 Kontrol PID .....</b>	<b>13</b>
<b>2.4 Kendali Logika Fuzzy .....</b>	<b>15</b>
2.4.1 Fuzzification .....	16
2.4.2 Rule Evaluation .....	17
2.4.3 Defuzzification .....	18
<b>2.5 Hybrid kendali PID-Fuzzy .....</b>	<b>18</b>
<b>2.6 Respon Sistem .....</b>	<b>19</b>
<b>BAB 3. METODOLOGI PENELITIAN .....</b>	<b>20</b>
<b>3.1 Alat dan Bahan .....</b>	<b>20</b>
<b>3.2 Tahapan Penelitian .....</b>	<b>21</b>
<b>3.3 Blok Diagram dan Perancangan Sistem .....</b>	<b>23</b>
<b>3.4 Sensor Tegangan .....</b>	<b>23</b>
<b>3.5 Buck Converter .....</b>	<b>25</b>
<b>3.6 Generator DC Penguatan Terpisah .....</b>	<b>27</b>
<b>3.5 Perancangan Kontrol .....</b>	<b>28</b>
<b>3.7 Perancangan Kontrol PID .....</b>	<b>28</b>
<b>3.8 Perancangan Kontrol Fuzzy .....</b>	<b>29</b>
<b>3.9 Perancangan Program Arduino .....</b>	<b>33</b>
<b>3.10 Aplikasi Monitoring Generator .....</b>	<b>35</b>
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>37</b>
<b>4.1 Penentuan Nilai Parameter PID dengan Metode <i>Manual Tuning</i> .....</b>	<b>37</b>
<b>4.2 Pengujian Komponen Pengendali .....</b>	<b>39</b>
4.2.1 Pengujian Sensor Tegangan .....	39
4.2.2 Pengujian <i>Buck Converter</i> .....	40
<b>4.3 Pengujian Keseluruhan Sistem .....</b>	<b>41</b>
4.3.1 Pengujian Tanpa Kontrol .....	42
4.3.2 Pengujian Dengan Kontrol .....	42
4.3.2.1 Pengujian Kontrol Tanpa Beban .....	43

A. Pengujian Kontrol PID Tanpa Beban.....	43
B. Pengujian Kontrol Fuzzy Tanpa Beban .....	43
C. Pengujian Kontrol <i>Hybrid</i> PID-Fuzzy Tanpa Beban.....	44
4.3.2.2 Pengujian Kontrol Dengan Beban .....	45
A. Pengujian Beban 15 Watt.....	45
B. Pengujian Beban 25 Watt.....	47
C. Pengujian Beban 40 Watt.....	49
D. Pengujian Beban 50 Watt.....	51
C. Pengujian Beban 65 Watt.....	53
<b>4.4 Hubungan Pembebanan Dengan Eksitasi .....</b>	<b>55</b>
<b>4.5 Perbandingan Hasil Pengujian .....</b>	<b>56</b>
4.5.1 Perbandingan Kontrol <i>Hybrid</i> PID-Fuzzy, Kontrol PID dan Kontrol Fuzzy Tanpa Beban .....	56
4.5.2 Perbandingan Kontrol <i>Hybrid</i> PID-Fuzzy, Kontrol PID dan Kontrol Fuzzy Dengan Beban .....	57
4.5.3 Perbandingan Sistem <i>Open Loop</i> dan <i>Close Loop Hybrid</i> PID-Fuzzy.....	59
<b>BAB 5. PENUTUP.....</b>	<b>61</b>
<b>5.1 KESIMPULAN .....</b>	<b>61</b>
<b>5.2 SARAN.....</b>	<b>61</b>
<b>DAFTAR PUSTAKA .....</b>	<b>62</b>
<b>LAMPIRAN.....</b>	<b>63</b>

**DAFTAR TABEL**

	Halaman
2.1 Spesifikasi Arduino UNO-R3 .....	8
2.2 Karakteristik Masing – Masing Kontrol <i>P-I-D</i> .....	15
4.1 <i>Manual Tuning</i> Parameter PID .....	38
4.2 Pengujian Sensor Tegangan .....	39
4.3 Pengujian <i>Buck Converter</i> .....	40
4.4 Daya <i>Input</i> dan Daya <i>Output</i> .....	40
4.5 Pengujian Pemberian Beban Generator <i>Open Loop</i> .....	42
4.6 Pengujian Pelepasan Beban Generator <i>Open Loop</i> .....	42
4.7 Tegangan <i>Output</i> Pemberian Beban 15 Watt.....	46
4.8 Tegangan <i>Output</i> Pelepasan Beban 15 Watt.....	47
4.9 Tegangan <i>Output</i> Pemberian Beban 25 Watt.....	48
4.10 Tegangan <i>Output</i> Pelepasan Beban 25 Watt.....	49
4.11 Tegangan <i>Output</i> Pemberian Beban 40 Watt.....	50
4.12 Tegangan <i>Output</i> Pelepasan Beban 40 Watt.....	51
4.13 Tegangan <i>Output</i> Pemberian Beban 50 Watt.....	52
4.14 Tegangan <i>Output</i> Pelepasan Beban 50 Watt.....	53
4.15 Tegangan <i>Output</i> Pemberian Beban 65 Watt.....	54
4.16 Tegangan <i>Output</i> Pelepasan Beban 65 Watt.....	55
4.17 Hubungan Beban Terhadap Arus Eksitasi .....	56
4.18 Hasil Pengujian Kontrol PID, Kontrol Fuzzy dan Kontrol <i>Hybrid</i> PID -Fuzzy .....	57
4.19 Hasil Pengujian Kontrol PID, Kontrol Fuzzy dan Kontrol <i>Hybrid</i> PID -Fuzzy dengan Pemberian Beban.....	58
4.20 Hasil Pengujian Kontrol PID, Kontrol Fuzzy dan Kontrol <i>Hybrid</i> PID -Fuzzy dengan Pelepasan Beban.....	58
4.21 Perbandingan <i>Drop</i> Tegangan Sistem <i>Open Loop</i> dan <i>Close Loop Hybrid</i> PID-Fuzzy .....	60
4.22 Perbandingan <i>Over</i> Tegangan Sistem <i>Open Loop</i> dan <i>Close Loop Hybrid</i> PID-Fuzzy .....	60



DAFTAR GAMBAR

	Halaman
2.1 Rangkaian Generator DC Penguatan Terpisah .....	4
2.2 Karakteristik $\phi$ - $I_f$ .....	5
2.3 Gambar 2.3. Karakteristik $E_a$ – $I_f$ .....	6
2.4 Modul Kontroler Arduino UNO-R3 .....	7
2.5 Rangkaian <i>Buck Converter</i> .....	9
2.6 (a) Tangkaian Ekuivalen Saat <i>Switch</i> Tertutup (b) Rangkaian Ekuivalen Saat <i>Switch</i> Terbuka (c) Tegangan Induktor pada Satu Periode <i>Switching</i> .....	10
2.7 Grafik Hubungan $V_o$ – $D$ .....	11
2.8 Arus Induktor .....	12
2.9 Sensor Tegangan DCT-Elektronik.....	13
2.10 Model Paralel PID .....	15
2.11 Diagram Blok Kendali Fuzzy.....	16
2.12 Proses Fuzzifikasi.....	17
2.13 <i>Rules Evaluation</i> .....	17
2.14 Defuzzifikasi .....	18
2.15 <i>Hybrid</i> PID-fuzzy .....	19
3.1 <i>Flowchart</i> Penelitian.....	21
3.2 Blok Diagram Sistem.....	23
3.3 Skema Rangkaian Sensor Tegangan.....	24
3.4 Rangkaian Pengujian Sensor Tegangan .....	25
3.5 Rangkaian <i>Buck Converter</i> .....	26
3.6 Generator DC.....	28
3.7 Blok Kendali Fuzzy .....	29
3.8 <i>Membership Function Error</i> .....	30
3.9 <i>Membership Function Delta Error</i> .....	31
3.10 Rule Base .....	32

3.11	Himpunan Defuzzifikasi .....	32
3.12	Perhitungan Titik Berat Luasan .....	33
3.13	<i>Flowchart</i> Program Arduino .....	34
3.14	Aplikasi Monitoring Generator.....	35
3.15	<i>Flowchart</i> Aplikasi Monitoring.....	36
4.1	Grafik Karakteristik Tegangan <i>Output</i> dan <i>Duty Cycle</i> .....	41
4.2	Pengujian Kontrol PID Tanpa Beban .....	43
4.3	Pengujian Kontrol Fuzzy Tanpa Beban.....	44
4.4	Pengujian Kontrol <i>Hybrid</i> PID-Fuzzy Tanpa Beban.....	44
4.5	Pemberian Beban 15 Watt .....	45
4.6	Pelepasan Beban 15 Watt .....	46
4.7	Pemberian Beban 25 Watt .....	47
4.8	Pelepasan Beban 25 Watt .....	48
4.9	Pemberian Beban 40 Watt .....	49
4.10	Pelepasan Beban 40 Watt .....	50
4.11	Pemberian Beban 50 Watt .....	51
4.12	Pelepasan Beban 50 Watt .....	52
4.13	Pemberian Beban 65 Watt .....	53
4.14	Pelepasan Beban 65 Watt .....	54
4.15	Pemberian dan Pelepasan Beban .....	59

DAFTAR LAMPIRAN

	Halaman
A. Listing Program Arduino Uno R3.....	63
B. Listing Program Monitoring C#.....	69
C. Data Pemberian Beban.....	74
C1. Pemberian Beban 15 Watt.....	74
C2. Pemberian Beban 25 Watt.....	76
C3. Pemberian Beban 40 Watt.....	78
C4. Pemberian Beban 50 Watt.....	80
C5. Pemberian Beban 65 Watt.....	82
D. Data Pelepasan Beban.....	84
D1. Pelepasan Beban 15 Watt.....	84
D2. Pelepasan Beban 25 Watt.....	86
D3. Pelepasan Beban 40 Watt.....	88
D4. Pelepasan Beban 50 Watt.....	90
D5. Pelepasan Beban 65 Watt.....	92
E. <i>Buck Converter</i> dan Sensor Tegangan.....	94
F. Generator DC Penguatan Terpisah.....	94
G. Beban Lampu Pijar.....	95
H. Gambar Keseluruhan Sistem.....	95

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Generator DC merupakan suatu mesin listrik yang mengubah energi mekanis menjadi energi listrik. Berdasarkan rangkaian belitan magnet atau penguat eksitasinya terhadap jangkar generator DC dibagi menjadi 3 jenis, yaitu generator penguat terpisah, generator *shunt* dan generator kompon.

Generator DC sudah banyak digunakan dalam kehidupan sehari-hari, mulai dari kendaraan yang sering ditemui seperti mobil hingga digunakan juga di bengkel dan pabrik. Pusat pembangkit listrik juga menggunakan generator DC sebagai penguat pada generator utama.

Generator DC akan mengalami penurunan tegangan *output* bila arus beban semakin besar, sehingga untuk mengatasi hal tersebut dapat dilakukan dengan meningkatkan tegangan induksi pada jangkar generator DC. Peningkatan tegangan induksi jangkar dapat dilakukan dengan cara meningkatkan kecepatan putaran dan meningkatkan flux medan generator DC. Kecepatan putaran generator DC sulit dikendalikan karena membutuhkan pengontrol kecepatan yang terdiri dari kontrol mekanis dan elektris yang cukup rumit, apalagi ketika dalam kondisi kecepatan tinggi. Agar lebih mudah, tegangan induksi pada jangkar dinaikkan dengan meningkatkan flux medan generator DC dengan mengatur arus eksitasi.

Pengaturan arus eksitasi memerlukan kontrol, sehingga tegangan induksi tidak mengalami kenaikan yang terlalu tinggi dan penurunan tegangan yang terlalu rendah. Kontrol *hybrid* PID-fuzzy adalah gabungan dari kontrol PID dan kontrol fuzzy yang dijadikan satu untuk mendapatkan hasil yang maksimal dari keuntungan dua kontrol tersebut. Pada sistem ini kendali utama adalah PID sedangkan kendali logika fuzzy bekerja membantu untuk meminimalkan *overshoot* atau *undershoot* serta *recovery* time pada sistem (Samsul, 2004). Kontrol *hybrid* PID-fuzzy ini digunakan untuk mengontrol arus eksitasi pada medan secara otomatis bila terjadi

penurunan tegangan *output* karena meningkatnya arus beban dengan *feedback* tegangan *output* generator DC.

Kontrol *hybrid* PID-fuzzy digunakan pada generator penguatan terpisah. Arduino UNO R3 digunakan sebagai *embedded system* yang digunakan dalam penerapan langsung pada pengontrolan generator DC penguatan terpisah. Diharapkan adanya kontrol ini mampu menstabilkan tegangan *output* pada generator DC.

## 1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana penerapan kontrol *hybrid* PID-fuzzy untuk mengontrol tegangan *output* generator DC penguatan terpisah.
2. Bagaimana kestabilan tegangan *output* generator DC setelah diterapkan kontrol *hybrid* PID-fuzzy.

## 1.3 Batasan Masalah

Berdasarkan rumusan masalah tersebut, maka pembahasan penelitian ini dibatasi pada hal-hal sebagai berikut :

1. Kontrol pada tegangan *output* hanya mengatur arus eksitasi pada medan di stator generator DC penguatan terpisah.
2. *Prime over* menggunakan mesin DC.
3. Tegangan *output* 220 V.
4. Beban pengujian menggunakan variasi beban lampu pijar 15W, 25W, 40W, 50W dan 65 W.
5. Pembahasan parameter kontrol *hybrid* PID-fuzzy pada *overshoot*, *recovery time*, *drop* tegangan dan *over* tegangan (*peak*) dan *error steady state*.

## 1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Membuat kontrol tegangan *output* generator DC penguatan terpisah menggunakan kontrol *hybrid* PID-fuzzy.

2. Penerapan kontrol *hybrid* PID-fuzzy dapat menjaga kestabilan tegangan *output* generator DC ketika diberikan beban.

### 1.5 Manfaat

Beberapa manfaat dalam penelitian ini adalah sebagai berikut :

1. Mengatasi kekurangan generator DC penguatan terpisah yaitu *drop voltage* ketika diberi beban dan *over voltage* ketika beban dilepaskan.
2. Penggunaan beban pada generator DC penguatan terpisah dapat bervariasi tanpa terjadi penurunan tegangan *output*.

### 1.6 Sistematika Penelitian

Secara garis besar laporan penelitian ini adalah sebagai berikut:

#### a. BAB 1. PENDAHULUAN

Berisi tentang latar belakang penelitian yang berjudul “Pengendalian Tegangan *Output* Generator DC Penguatan Terpisah Menggunakan Kontrol *Hybrid* PID-Fuzzy Berbasis Arduino UNO R3” tentang garis besar tujuan dalam penelitian serta poin-poin yang akan dibahas dalam BAB selanjutnya.

#### b. BAB 2. TINJAUAN PUSTAKA

Berisi penjelasan tentang teori penunjang Generator DC penguatan terpisah, komponen pengendali yang akan digunakan serta kontrol *hybrid* PID-Fuzzy.

#### c. BAB 3. METODOLOGI PENELITIAN

Menjelaskan tentang metode penelitian yang akan dilakukan meliputi alur penelitian, penjelasan mengenai rancangan komponen yang akan digunakan, rancangan kontrol *hybrid* PID-Fuzzy dan perancangan *software* monitoring penelitian.

#### d. BAB 4 HASIL DAN PEMBAHASAN

Berisi hasil penelitian dan analisa hasil penelitian dari pengujian komponen yang telah dirancang, pengujian *tuning manual*, pengujian keseluruhan sistem yang telah dibuat.

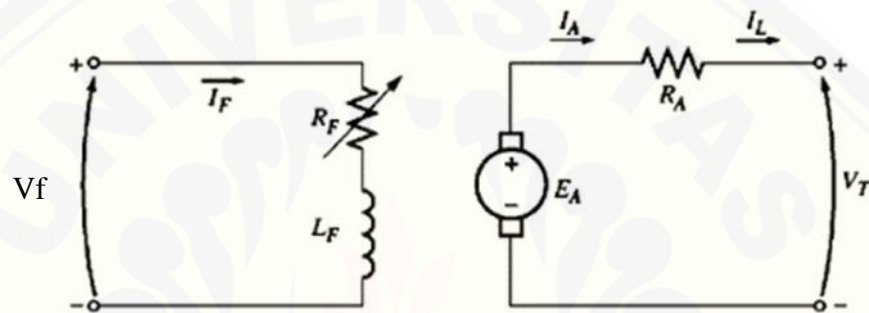
#### e. BAB 5 PENUTUP

Berisi tentang kesimpulan dan saran dari hasil penelitian.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Generator DC Penguatan Terpisah

Pada generator penguatan terpisah, belitan eksitasi ( penguat eksitasi) tidak terhubung menjadi satu dengan rotor. Sehingga untuk membangkitkan fluks pada medan diperlukan arus searah yang dipasangkan pada kumparan medan yang mempunyai tahanan  $R_f$  sehingga menghasilkan fluks pada kedua kutub. Persamaan pada generator penguatan terpisah sebagai berikut.



Gambar 2.1 Rangkaian Generator DC Penguatan Terpisah

(Sumber : Chapman, 2005)

Persamaan :

$$I_f = \frac{V_f}{R_f} \quad (2.1)$$

$$V_T = E_A - I_A R_A \quad (2.2)$$

Keterangan :

$I_f$  = Arus kumparan medan (A)

$V_f$  = Tegangan kumparan medan (V)

$R_f$  = Resistansi kumparan medan (  $\Omega$  )

$E_A$  = Tegangan induksi jangkar (V)

$I_A$  = Arus jangkar (A)

$R_A$  = Resistansi jangkar (  $\Omega$  )

$I_L$  = Arus beban (A)

$V_T$  = Tegangan *output* (V)

Dari persamaan tersebut dapat diketahui bahwa ketika beban semakin tinggi maka nilai tegangan *output* semakin kecil karena ada drop tegangan dari  $I_a R_a$ . Untuk meningkatkan tegangan *output* dapat dilakukan dengan cara meningkatkan tegangan induksi ( $E_a$ ), dengan persamaan sebagai berikut.

$$E = K \omega_r \quad (2.3)$$

Keterangan :

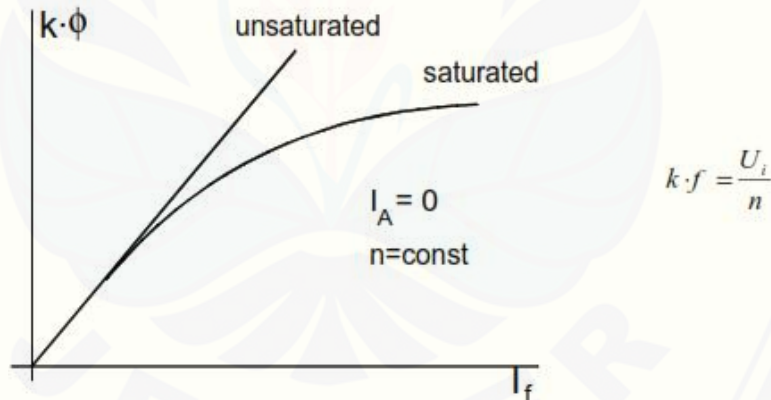
$E_a$  = Tegangan Induksi (V)

$K$  = Konstanta

$\phi$  = Fluks magnet (Wb)

$\omega_r$  = Kecepatan putar (RPM)

Tegangan induksi pada generator dipengaruhi oleh nilai fluks dan kecepatan putar dari generator. Untuk meningkatkan fluks dapat dilakukan dengan cara meningkatkan arus atau tegangan pada medan generator. Karakteristik fluks dengan  $I_f$  dapat dikatakan linier sebelum mencapai saturasi.



Gambar 2.2. Karakteristik  $\phi - I_f$

(Sumber: Hameyer, 2001)

Semakin tinggi nilai  $I_f$  semakin tinggi nilai fluks yang akan mengakibatkan tegangan induksi pada generator akan meningkat, sama halnya dengan kecepatan. Semakin tinggi tinggi kecepatan maka tegangan induksi semakin tinggi begitu pula dengan sebaliknya.



### 2.1.1 Teori Kontrol Tegangan *Output* Generator DC Penguatan Terpisah

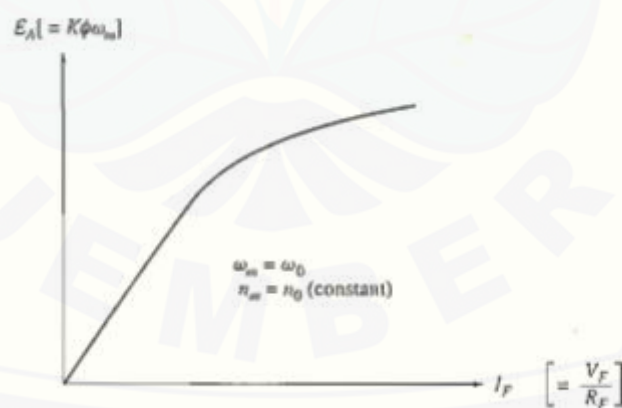
Tegangan *output* pada generator DC penguatan terpisah dapat dilakukan dengan merubah tegangan induksi pada mesin DC. Berdasarkan hukum Kirchoff II tegangan pada rangkaian tertutup akan bernilai sama dengan nol menghasilkan persamaan berikut  $V_T = E_a - I_a R_a$ . Jadi, jika  $E_a$  meningkat maka tegangan *output* akan meningkat dan jika  $E_a$  menurun maka tegangan *output* akan menurun. Pada persamaan  $E_a$  adalah  $E_a = K\phi \omega_r$  sehingga dapat dilakukan dua kemungkinan untuk mengontrol tegangan *output* yang pertama adalah  $\omega_r$  (kecepatan putar) dan  $\phi$  (fluks):

1. Merubah kecepatan putar ( $\omega_r$ ).

Jika  $\omega_r$  dinaikan dari  $E_a = K\phi \omega_r$  akan meningkatkan tegangan *output* karena persamaan tegangan *output*  $V_T = E_a - I_a R_a$ .

2. Merubah fluks ( $\phi$ )

Jika hambatan pada medan ( $R_f$ ) dikurangi maka arus akan naik karena  $I_f = V_f / R_f$ . Dimana jika arus medan meningkat akan meningkatkan fluks dan akan meningkatkan juga tegangan induksi  $E_a = K\phi \omega_r$  dan meningkatkan tegangan *output*  $V_T = E_a - I_a R_a$ .



Gambar 2.3. Karakteristik  $E_a - I_f$

(Sumber : Chapman, 2005)

Kecepatan putar *prime mover* sangat terbatas dan sukar untuk dikendalikan dalam penerapannya sehari-hari. Pada generator DC penguatan terpisah akan lebih

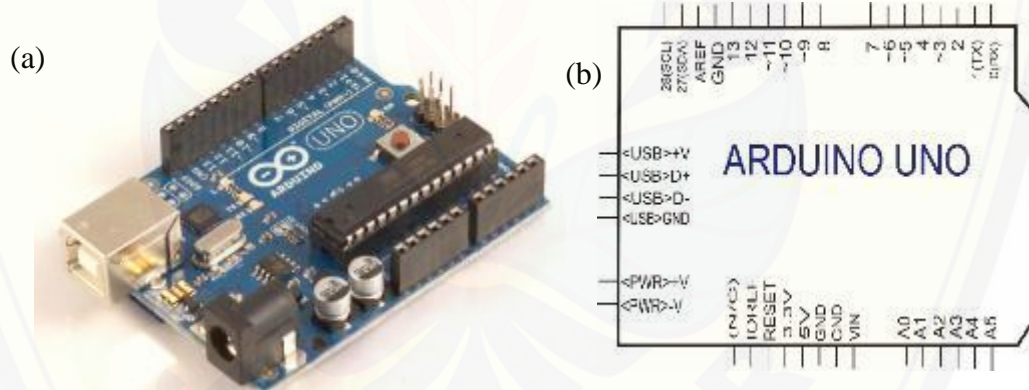
mudah mengontrol tegangan *output* dengan mengendalikan arus  $I_f$  (Chapman, 2005).

## 2.2 Komponen Pengendali

Pada penelitian yang akan dilakukan terdapat komponen pengendali yang digunakan untuk pengolah hasil dari sensor tegangan, dimana sebagai tempat proses kontrol sesuai dengan yang diinginkan. Komponen pengendali tersebut adalah:

### 2.2.1 Arduino Uno R3

Arduino Uno adalah modul mikrokontroler berbasis ATmega 328. Modul ini mempunyai 14 pin *input-output* (14 I/O) 6 diantaranya dapat digunakan sebagai *output pulse width modulation* (PWM), 6 *input* analog, sebuah osilator kristal 16MHz, sebuah koneksi USB, tombol *reset* dan koneksi catu daya. Kontroler ini semua yang dibutuhkan untuk mendukung mikrokontroler.



Gambar 2.4. Modul Kontroler Arduino UNO-R3

(Sumber : [www.arduino.com](http://www.arduino.com))

“Uno” memiliki arti satu dalam bahasa itali dan sebagai tanda versi pertama (Arduino 1.0) yang menjadi versi acuan Arduino selanjutnya. Uno adalah seri terakhir dari seri modul USB Arduino. Uno tidak lagi menggunakan *chip* FTDI USB-to-serial tetapi menggunakan ATmega 8U2 sebagai konverter USB ke serial. Gambar 3.4 (a) adalah Arduino UNO-R3 sedangkan Gambar 2.4 (b) menunjukkan skema dari modul Arduino UNO-R3 yang merupakan penyederhanaan dari gambar modul asli (Arduino UNO 2015).

Masing-masing pin I/O dapat digunakan sebagai *input* maupun *output* dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Masing-masing pin ini beroperasi pada tegangan 5 volt dan dapat menerima arus 40 mA serta sudah memiliki *pull-up* resistor 20-50 k $\Omega$  yang secara *default* tidak tersambung (*disconnected*). Berikut ini adalah tabel spesifikasi dari Arduino Uno R3 dapat kita lihat pada Tabel 2.1.

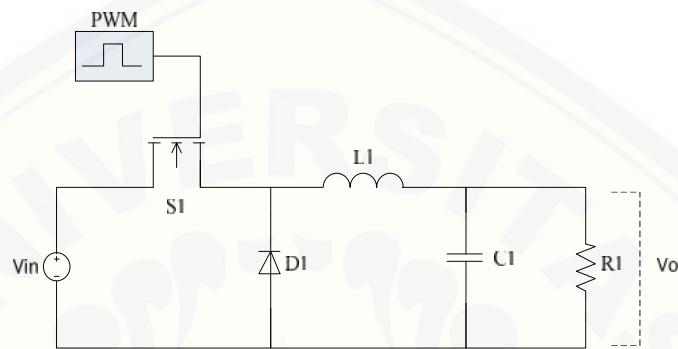
Tabel 2.1. Spesifikasi Arduino UNO-R3

Microcontroller	ATmega328
Tegangan Operasi	5 V
Tegangan Input	7-12 V
Tegangan Input(Batas)	60-20 V
I/O Digital	14 (6 buah menyediakan PWM output)
Input Analog	6 buah
Arus DC per I/O pin	40 Ma
Arus DC untuk pin 3.3	50 Ma
Flash Memory	Bootloader
SRAM	2 Kb
EEPROM	1 Kb
Kecepatan Klok	16 MHz
Diterjemahkan dari: (Arduino UNO 2015)	

### 2.2.2 Buck Converter

*Buck converter* adalah salah satu DC – DC *converter* yang dapat menurunkan tegangan DC berdasarkan *duty-cycle* yang diberikan pada *switching device*. Gambar adalah rangkaian *buck converter*. Tegangan masukan ( $V_{in}$ ) adalah catu daya DC dimana tegangannya yang akan diubah menjadi lebih rendah. *Controllable switch* (S1) merupakan *semiconductor switching device* yang digunakan sebagai saklar pada *buck converter*. Saklar inilah yang nantinya mengatur besar kecilnya *duty cycle* sehingga tegangan pada *buck converter* dapat diatur sedemikian rupa tanpa melewati tegangan masukan. Komponen lainnya pada *buck converter* adalah induktor (L1) yang berfungsi sebagai penyimpan arus. Dioda

(D1) merupakan bahan semikonduktor yang akan aktif dalam keadaan *forward bias*, yaitu tegangan anoda lebih tinggi daripada tegangan katoda. Kapasitor (C1) merupakan komponen yang berfungsi sebagai *low pass filter* dan resistor (R1) merupakan beban sistem dan berfungsi sebagai bagian dari *low pass filter*. Tegangan keluaran ( $V_o$ ) merupakan hasil tegangan keluaran yang diinginkan (Gunawan, 2009).



Gambar 2.5. Rangkaian *Buck Converter*

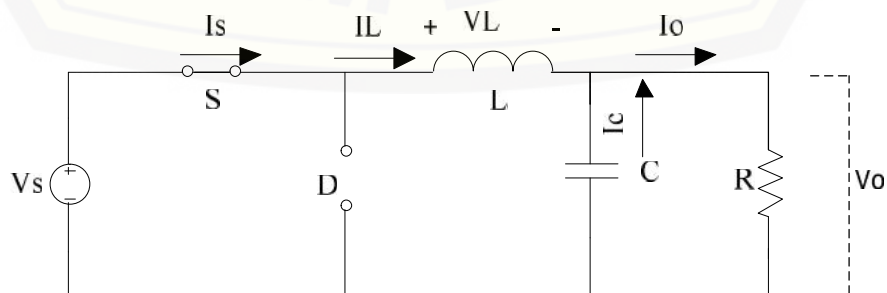
Pada saat *switch* tertutup, rangkaian ekivalen *buck converter* akan tampak seperti gambar . Pada kondisi ini tegangan induktor adalah,

$$V = V_s - V_o = L \frac{d_L}{d} \quad (2.4)$$

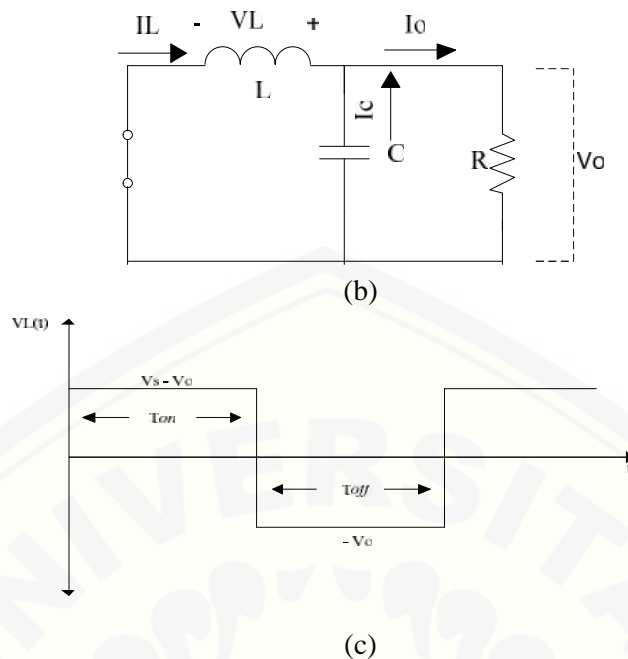
pada saat *switch* terbuka, rangkaian ekivalen *buck converter* akan tampak seperti gambar . Pada posisi ini tegangan induktor adalah,

$$V = -V_o = L \frac{d_L}{d} \quad (2.5)$$

hasil yang didapatkan adalah tegangan induktor seperti ditunjukkan gambar , dimana  $T_{on}$  menunjukkan waktu saat *switch* tertutup dan  $T_{off}$  menunjukkan saat *switch* terbuka.



(a)



Gambar 2.6 (a) Tangkai Ekivalen Saat *Switch* Tertutup (b) Rangkaian Ekivalen Saat *Switch* Terbuka (c) Tegangan Induktor pada Satu Periode *Switching*.

(Sumber : Gunawan, 2009)

pada kondisi tunak tercapai ketika arus yang dilepaskan induktor sama dengan arus yang diserap induktor dalam satu periode. Menurut persamaan induktor,

$$V = L \frac{d_L}{d} \quad (2.6)$$

maka integrasi dalam satu periode dari  $t = t_0$  sampai dengan  $t = T$  didapatkan,

$$I_L(T) - I_L(t_0) = \int_0^T V \, d \quad (2.7)$$

karena ketika kondisi tunak tercapai nilai total perubahan arus adalah nol, maka didapatkan,

$$0 = \int_0^T V \, d \quad (2.8)$$

persamaan 2.5 disebut *inductor Volt-second balance* yang digunakan untuk menurunkan karakteristik tunak dari sistem *buck converter*. Dari gambar 2.6c didapatkan nilai tegangan induktor selama satu periode sehingga,

$$\int_0^T V \, d = (V_s - V_o)D + (-V_o)(1 - D)T \quad (2.9)$$

dengan mensubstitusikan persamaan 2.9 ke persamaan 2.10 didapatkan,

$$(2.10)$$

$$0 = (V_s - V_o)D + (-V_o)(1 - D)T$$

dengan membagi kedua sisi dengan T didapatkan,

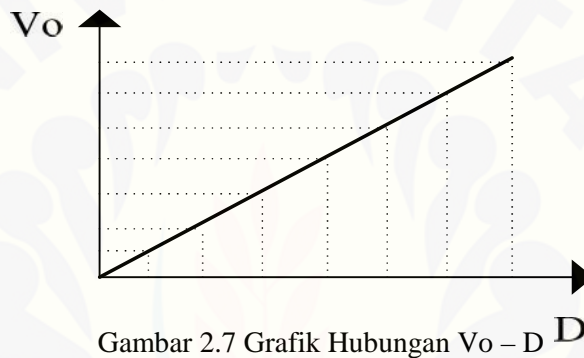
$$0 = (V_s - V_o)D + (-V_o)(1 - D) \quad (2.11)$$

sehingga didapatkan penyelesaian untuk  $V_o$  adalah,

$$V_o = DV_s = M(D)V_s \quad (2.12)$$

dimana  $M(D)$  menunjukkan *gain* dari sistem.

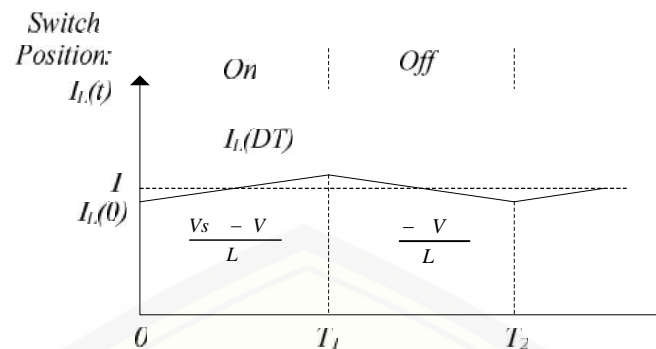
Persamaan ini menunjukkan bahwa semakin besar nilai *duty cycle* maka semakin besar nilai tegangan keluaran. Nilai tegangan keluaran secara ideal dapat bernilai 0 Volt hingga tegangan masukan.



Gambar 2.7 Grafik Hubungan  $V_o - D$

(Sumber : Gunawan, 2009)

Agar sistem bekerja pada daerah yang kontinyu maka arus induktor harus tetap kontinyu dalam satu periode. Gambar menunjukkan arus dari induktor selama satu periode dalam keadaan tunak. Pada *buck converter*, arus rata-rata induktor ( $I_L$ ) dalam kondisi tunak adalah sama dengan arus beban  $I$ . Gambar 2.8 menunjukkan bahwa arus induktor berada pada sekitar arus rata-rata  $I$  dengan nilai arus maksimum sebesar  $I+I_L$  dan arus minimum  $I-I_L$ .



Gambar 2.8 Arus Induktor

(Sumber : Gunawan, 2009)

Untuk mendapatkan nilai arus induktor kontinyu, nilai  $I - I_L$  harus lebih besar daripada nol. Nilai  $I - I_L$  ini menjadi batasan besarnya  $L_{\min}$  yang dibutuhkan, Dari gambar arus pada kondisi *switch* kedua didapatkan,

$$\frac{2\Delta L}{T_s(D-1)} = \frac{-V}{L} \quad (2.13)$$

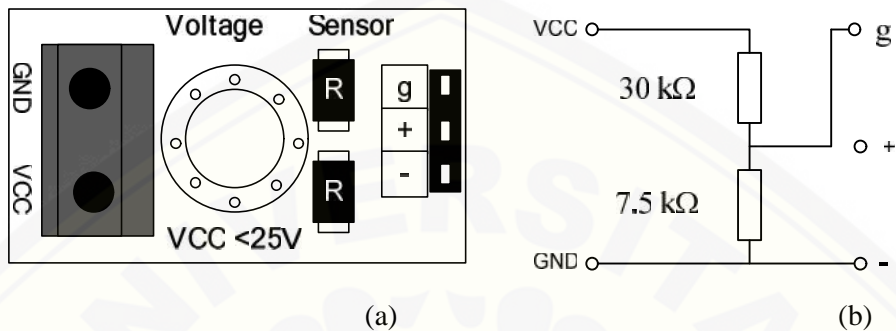
Nilai  $I - I_L$  harus lebih besar daripada nol agar sistem bekerja pada daerah kontinyu. Dalam hal ini dibutuhkan perhitungan nilai frekuensi dengan nilai  $L$  yang telah ditentukan,

$$\begin{aligned} -\Delta L &> 0 \\ \frac{V}{R} - \frac{V}{2L} T_s(1-D) &> 0 \\ \frac{1}{R} &> \frac{T_s(1-D)}{2L} \\ \frac{1}{T_s} &> \frac{(1-D)R}{2L} \\ f &> \frac{(1-D)R}{2L} \end{aligned} \quad (2.14)$$

Tegangan keluaran sistem pada kenyataannya tidak dapat bernilai konstan secara sempurna. Hal ini dikarenakan kapasitor yang digunakan akan terus melakukan pengisian dan pelepasan muatan (Gunawan, 2009).

### 2.2.3 Sensor Tegangan

Sensor tegangan adalah modul pembagi tegangan yang digunakan untuk mengubah tegangan yang akan diukur untuk dapat diumpungkan ke pin kontroler yang sudah memiliki konverter analog ke digital (ADC).



Gambar 2.9 Sensor Tegangan DCT-Elektronik

Gambar 2.9(a) adalah modul sensor pembagi tegangan, sedangkan Gambar 2.9(b) adalah diagram skematik dari sensor tersebut. Kaki VCC dan GND adalah kaki-kaki untuk menempatkan terminal tegangan yang akan diukur sedangkan pin g, + dan - adalah pin untuk dihubungkan dengan kaki ADC kontroler (SurplusTek, 2015).

## 2.3 Kontrol PID

Kontrol PID merupakan kontroler yang memiliki karakteristik adanya umpan balik sehingga sistem instrumentasi sesuai dengan yang diharapkan. Komponen PID terdiri dari *proportional*, *integratif* dan *derivatif*.

### a. *Proportional* ( $K_p$ )

Kontrol  $K_p$  merupakan penguatan dari sinyal *feedback* sehingga berpengaruh dalam menambah atau mengurangi kestabilan sistem, khususnya memperbaiki respon transien yaitu *rise time* dan *settling time* dan mengurangi *error steady state* tetapi tidak dapat menghilangkannya. Hubungan kontrol  $K_p$  dengan sinyal *error* ( $e(t)$ ):



$$u(t) = K_p e(t) \quad (2.15)$$

b. *Integratif* ( $K_i$ )

Kontrol  $K_i$  bertujuan menghilangkan kesalahan dalam keadaan tunak (*offset*) yang biasanya dihasilkan kontrol *propotional*. Hubungan kontrol  $K_i$  dengan sinyal *error* ( $e(t)$ )

$$u(t) = K_i \int_0^t e(t) dt \quad (2.16)$$

c. *Derivatif* ( $K_d$ )

Kontrol  $K_d$  menghasilkan *output* kontroler sebanding dengan laju perubahan sinyal *error*. Kontrol  $K_d$  tidak pernah digunakan sendirian karena hanya aktif pada periode peralihan. Hubungan kontrol  $K_d$  dengan sinyal *error* ( $e(t)$ )

$$u(t) = K_d \frac{d e(t)}{dt} \quad (2.17)$$

Gabungan kontrol PID dapat dirumuskan sebagai berikut:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d e(t)}{dt} \quad (2.18)$$

Dengan :

$u(t)$  = sinyal *output* pengendali PID

$K_p$  = konstanta proporsional

$K_i$  = konstanta integral

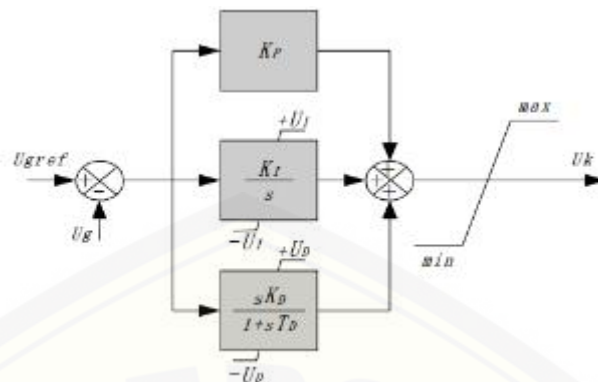
$K_d$  = konstanta derivatif

$e(t)$  = sinyal *error*

Persamaan PID tersebut masih dalam waktu *continuous*, sehingga perlu didiskritkan terlebih dahulu agar bisa dimasukkan ke dalam *embedded system*. Sehingga persamaannya menjadi sebagai berikut :

$$u(k) = u(t) = K_p e(t) + K_i T_c [e(k-1) + e(t)] + K_d \frac{e(k) - e(k-1)}{T} \quad (2.19)$$

$T_c$  = waktu *sampling* atau waktu cuplik (*Sampling time*).



Gambar 2.10 Model Paralel PID

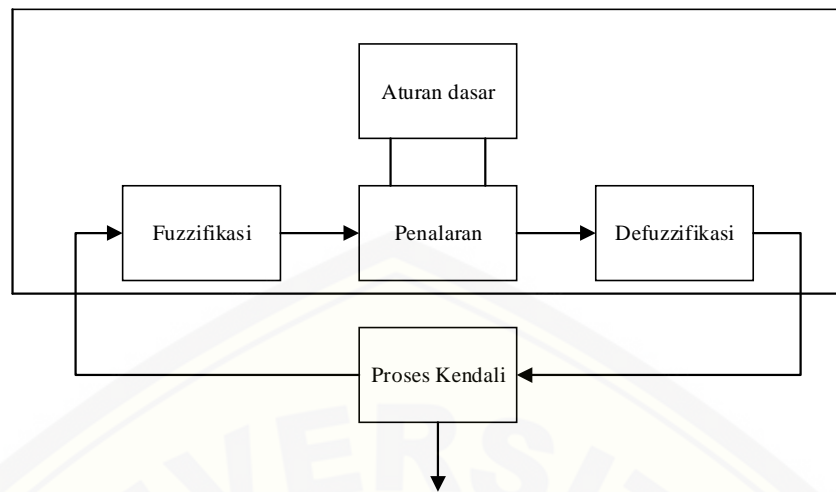
Penentuan parameter kontroler PID sangat penting supaya sistem *close loop* memenuhi performa yang diinginkan yang disebut dengan *tuning*. *Tuning* digunakan untuk menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$ . Cara *tuning* ini dapat dilakukan dengan *trial* dan *error*. Cara ini sangat mudah dilakukan yaitu dengan cara coba-coba memasukkan nilai konstanta pada  $K_p$ ,  $K_i$  dan  $K_d$  hingga diperoleh hasil yang diinginkan dengan mengacu pada karakteristik dari masing – masing  $K_p$ ,  $K_i$  dan  $K_d$ .

Tabel 2.2 Karakteristik masing – masing kontrol *P-I-D*

<i>Respon loop</i> tertutup	<i>Rise time</i>	<i>Overshoot</i>	<i>Settling time</i>	<i>Error steady-state</i>
$K_p$	Mengurangi	Meningkatkan	Mengurangi	Mengurangi
$K_i$	Mengurangi	Meningkatkan	Meningkatkan	Menghilangkan
$K_d$	Perubahan Kecil	Mengurangi	Menurun	Perubahan Kecil

## 2.4 Kendali Logika Fuzzy

Kendali logika fuzzy menggunakan pengetahuan kendali dari operator manusia. Kesulitan yang sering dihadapi dalam perumusan suatu model matematika dapat disebabkan oleh sistem non linier, variasi waktu dari proses kendali, gangguan-gangguan yang tidak dapat diperkirakan sebelumnya, keterbatasan sensor atau kesulitan dalam mendapatkan hasil pengukuran yang akurat. Dalam permasalahan tersebut, operator manusia tetap mampu melaksanakan tugasnya dalam mengendalikan suatu proses dengan baik.



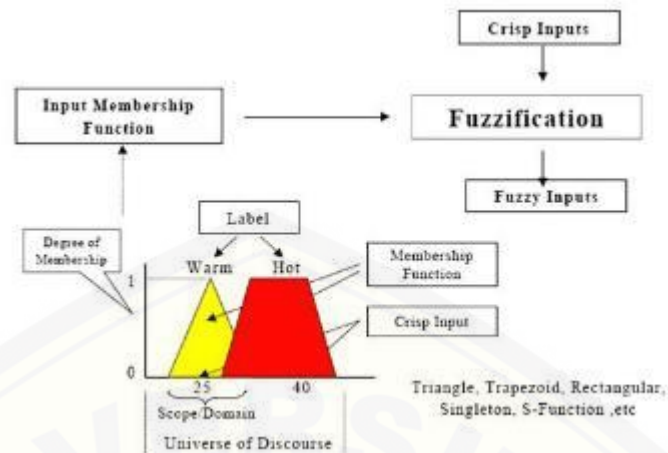
Gambar 2.11 Diagram Blok Kendali Fuzzy

Kemampuan manusia ini dapat digunakan sebagai model alternatif yang akurat dalam proses kendali. Cara untuk menerapkan kemampuan manusia ini adalah dengan menggunakan deskripsi linguistik yang terdiri dari sekumpulan aturan (*If – Then Rules*). Misalnya, jika *error* PB dan *d error* PB maka kecepatan motor akan PB.

Desain kendali logika fuzzy biasanya dilakukan melalui 3 langkah utama yaitu proses fuzzifikasi, proses *rule evaluation* dan proses defuzzifikasi.

#### 2.4.1 Fuzzification

*Fuzzification* adalah langkah pertama dalam kendali logika fuzzy melibatkan transformasi domain yang disebut dengan proses fuzzifikasi (*fuzzification*). Tujuan utama adalah mentransformasikan *crisp input* menjadi *fuzzy input*. Dalam mentransformasikan *crisp input* menjadi *fuzzy input*. Langkah awal adalah penentuan *membership function* untuk setiap *crisp input*. Lalu proses selanjutnya proses fuzzifikasi akan mengambil nilai *crisp input* (*error* dan *derror*) dan membandingkan dengan *membership function* yang telah ada untuk menghasilkan nilai *fuzzy input*.

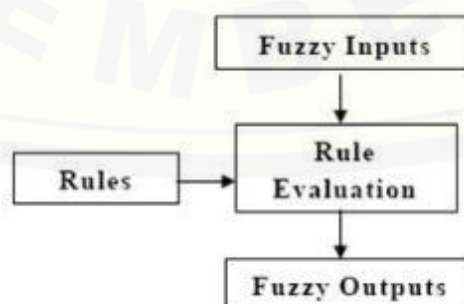


Gambar 2.12 Proses fuzzifikasi  
(Sumber : <https://fahmizaleeits.wordpress.com/category/kuliah-kontrol/fuzzy-logic/>)

#### 2.4.2 Rule Evaluation

*Rule evaluation* merupakan tahap kedua dari proses kendali logika fuzzy. Dengan menggunakan aturan (*rule*) yang telah dibuat untuk menentukan aksi kontrol yang harus dilakukan sesuai dengan nilai *input* yang dihasilkan *fuzzy input*. Pada tahap ini akan dilakukan evaluasi tiap-tiap *rule* dengan *input* yang dihasilkan pada proses *fuzzification*.

Aturan-aturan pada fuzzy pada umumnya merupakan *statement if-then* yang menggambarkan aksi yang harus dilakukan sebagai respon dari berbagai macam *fuzzy input*. Proses yang terjadi pada tahap *rule evaluation* bisa dilihat pada gambar 2.13

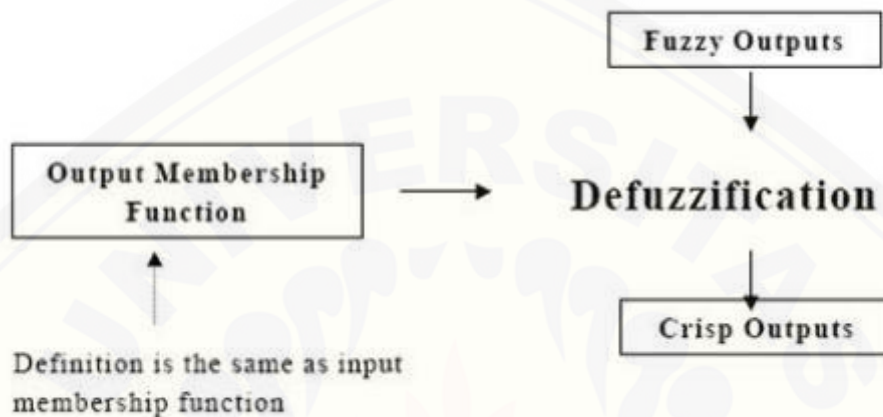


Gambar 2.13 Rule Evaluation

(Sumber : <https://fahmizaleeits.wordpress.com/category/kuliah-kontrol/fuzzy-logic/>)

### 2.4.3 Defuzzification

Proses terakhir pada tahapan kendali logika fuzzy adalah *defuzzification*. Proses ini bertujuan untuk merubah kembali nilai linguistik *output* dari proses inferensi menjadi nilai *crispt* yang akan digunakan pada proses sistem kontrol selanjutnya.

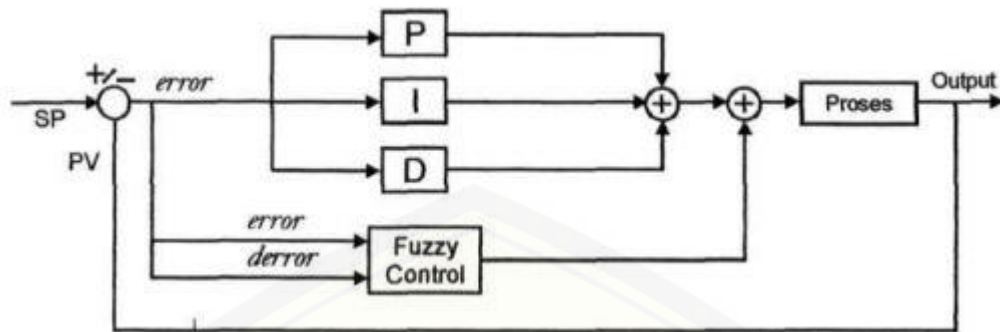


Gambar 2.14 Defuzzifikasi

(Sumber : <https://fahmizaleeits.wordpress.com/category/kuliah-kontrol/fuzzy-logic/>)

### 2.5 Hybrid PID-Fuzzy

Hybrid PID-Fuzzy adalah ide untuk memanfaatkan sisi keuntungan dari kedua kontroler (Priyanka, 2014). *Hybrid* adalah kombinasi atau penggabungan dari dua kategori yang berbeda. Dimana proses pada *hybrid* menggabungkan kinerja dua controller yang berbeda untuk mencapai kinerja yang lebih baik dibandingkan dengan pengendali individu. Pada sistem kontrol *hybrid* PID-fuzzy, kontrol PID adalah kontrol utama sedangkan kontrol fuzzy adalah kontrol yang membantu untuk mengurangi *overshoot* dan *disturbance* yang terjadi pada sistem. Oleh karena itu, kontrol fuzzy akan digunakan dengan beban yang lebih kecil bila dibandingkan dengan kontrol PID sebagai kendali utama.



Gambar 2.15 Hybrid PID-fuzzy

(Sumber : <https://fahmizaleeits.wordpress.com/category/kuliah-kontrol/fuzzy-logic/>)

## 2.6 Respon Sistem

Perubahan perilaku *output* terhadap perubahan sinyal *input*. Dalam hal ini terdapat beberapa respon sistem yaitu:

a. *Overshoot*

Kondisi dimana nilai maksimum respon yang melampaui kondisi nilai *steady-state*.

b. *Recovery time*

Waktu pengembalian respon *output* hingga mencapai hasil yang disesuaikan dengan *set point*.

c. *Peak* atau gangguan

Gangguan yang terjadi ketika sistem diberikan beban atau ketika sistem mengalami pelepasan beban.

d. *Error steady-state*

Kondisi ketika sistem sudah mencapai kondisi stabilnya, sinyal respon akan berhenti dikisaran *set point* dimana terdapat selisih nilai akhir dengan *set point*.

## BAB 3. METODOLOGI PENELITIAN

### 3.1 Alat dan Bahan

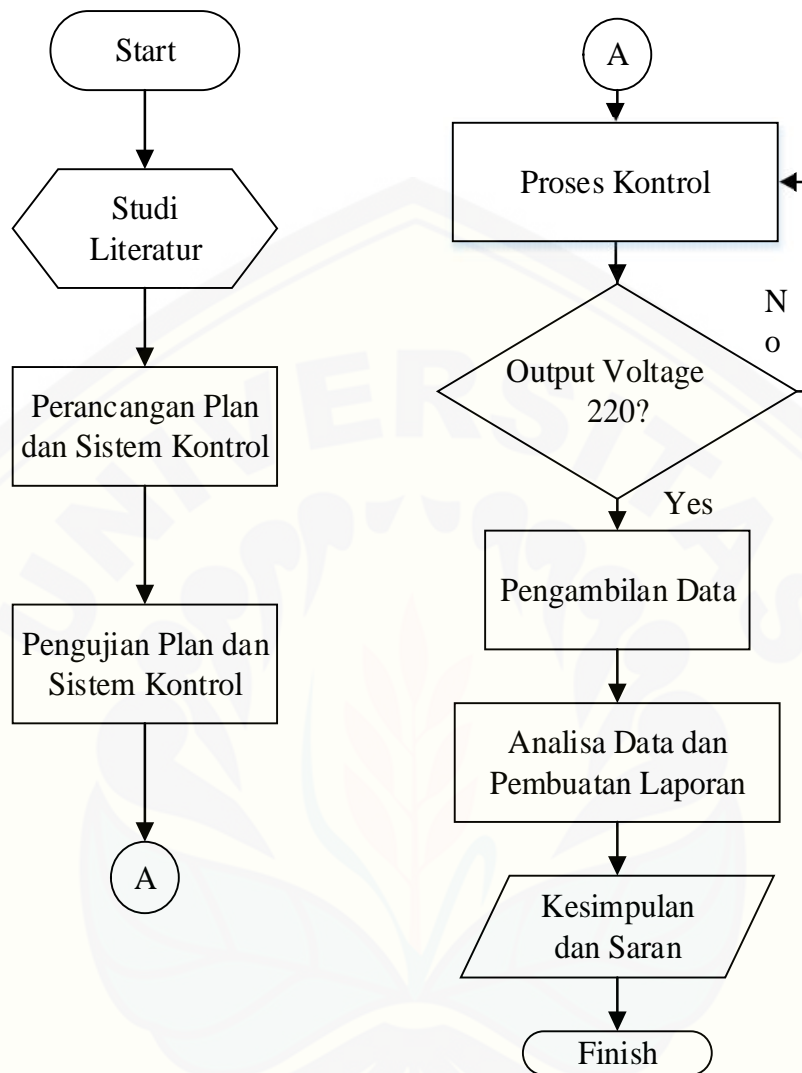
#### 3.1.1 Alat :

- a. Multimeter
- b. Tachometer
- c. Timer
- d. Data *logger*

#### 3.1.2 Bahan:

- a. Generator DC penguatan terpisah
- b. Motor DC sebagai *prime mover*
- c. *Belt*
- d. AVR
- e. Arduino UNO R3
- f. Sensor tegangan
- g. *Buck Converter*
- h. *Rectifier*
- i. MCB
- j. Resistor 100
- k. *Jumper*

### 3.2 Tahapan Penelitian



Gambar 3.1 Flowchart Penelitian

Tahapan penelitian rancang bangun kontrol *hybrid* PID-fuzzy tegangan *output* generator DC penguatan terpisah berbasis ARDUINO UNO R3 adalah sebagai berikut:

1. Studi literatur

Tahap pertama memulai penelitian ini dilakukan dengan mencari literatur yang terkait dengan penelitian ini, dengan tujuan agar perancangan dan pelaksanaan penelitian dapat sesuai dengan yang diinginkan.



2. Perancangan plan dan sistem kontrol

Tahap ketiga adalah proses perancangan plan dan sistem kontrol *hybrid* PID-fuzzy tegangan *output* generator DC penguatan terpisah yang terdiri dari *buck converter*, sensor tegangan, program dan arduino uno R3

3. Pengujian plan dan sistem kontrol

Tahap keempat adalah pengujian plan dan sistem kontrol *hybrid* PID-fuzzy. Hal ini dilakukan untuk mengetahui plan dan sistem yang dirancang berjalan sesuai dengan perancangan.

4. Pemrograman sistem kontrol

Tahap kelima adalah menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  serta menentukan *membership function* dari logika fuzzy yang akan digunakan pada plan agar mencapai target yang diinginkan.

5. Pengujian dan pengambilan data

Tahap keenam adalah pengujian dan pengambilan data dari rancang bangun kontrol *hybrid* PID-fuzzy tegangan *output* generator DC penguatan terpisah dengan variasi beban yang telah ditentukan.

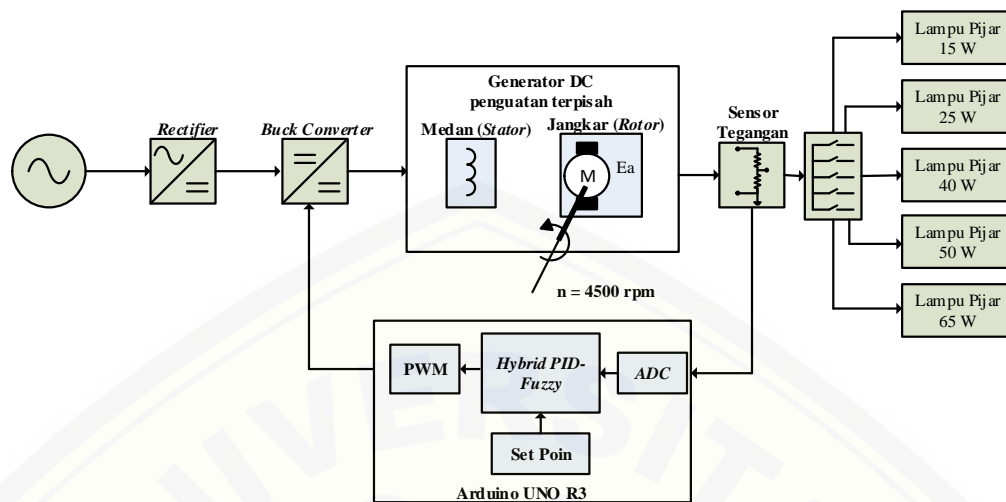
6. Analisa data dan pembuatan laporan

Tahap ketujuh adalah melakukan analisa pada data yang diperoleh dari tahap sebelumnya.

7. Penarikan kesimpulan

Tahap kedelapan adalah tahap terakhir pada penelitian yaitu memberikan kesimpulan dari pengujian dan analisa data yang diperoleh.

### 3.3 Blok Diagram dan Perancangan Sistem



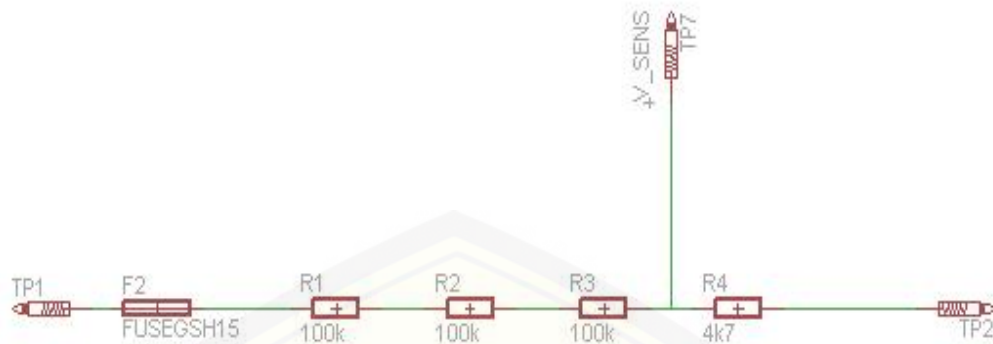
Gambar 3.2 Blok Diagram Sistem

Prinsip kerja pada penelitian ini dapat dilihat pada gambar 3.2. *Buck converter* diatur dan dikendalikan oleh *hybrid PID-fuzzy* sehingga eksitasi pada generator DC penguatan terpisah dikontrol oleh *hybrid PID-fuzzy*. Awal mula *prime mover* memutar generator DC akan menghasilkan tegangan tertentu dengan nilai tegangan yang ditentukan sebagai *set point* dari kontrol dengan sensor tegangan yang memberikan nilai *feedback* pada arduino. *Feedback* tersebut akan diolah pada *hybrid PID-fuzzy* sebagai pemberi nilai masukan pada *buck converter* yang akan memberikan nilai eksitasi pada kumparan medan generator yang optimal sehingga tegangan keluaran sesuai dengan *set point* yang diinginkan.

### 3.4 Sensor Tegangan

Sensor tegangan dirancang dengan menggunakan rangkaian pembagi tegangan. Rangkaian pembagi tegangan yang dirancang menggunakan resistor  $300 \text{ k}$  dan resistor  $4\text{k}7$ . Resistor yang digunakan dirangkai seri agar dapat membaca tegangan  $0 - 324.15 \text{ V}$  dan perbandingan tegangan untuk di olah pada arduino uno R3  $0 - 5 \text{ V}$ .

Pada penelitian yang dilakukan penggunaan resistor  $300 \text{ k}$  digantikan dengan merangkai seri resistor  $100 \text{ k}$  karena resistor  $300 \text{ k}$  sulit dicari di pasaran. Skema rangkaian sensor tegangan dapat dilihat pada gambar 3.3



Gambar 3.3 Skema Rangkaian Sensor Tegangan

Perhitungan dari perancangan sensor tegangan dimana

$$V = \frac{V \times R2}{(R1 + R2)}$$

$$V_{out} = 5; V_{in} = 324.15; R1 = 300 \text{ k}$$

$$5 = \frac{324.15 \times R2}{(300k + R2)}$$

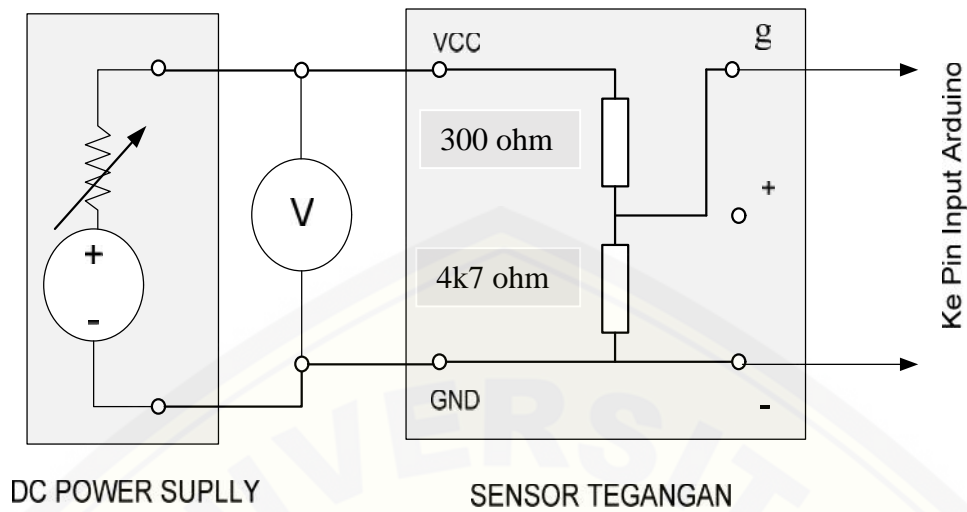
$$5 \times (300k + R2) = 324.15 \times R2$$

$$1500k + 5R2 = 324.15R2$$

$$R2 = \frac{1500k}{319.15}$$

$$R2 = 4k7 \Omega$$

Pengujian sensor tegangan dapat dilihat seperti pada gambar 3.4

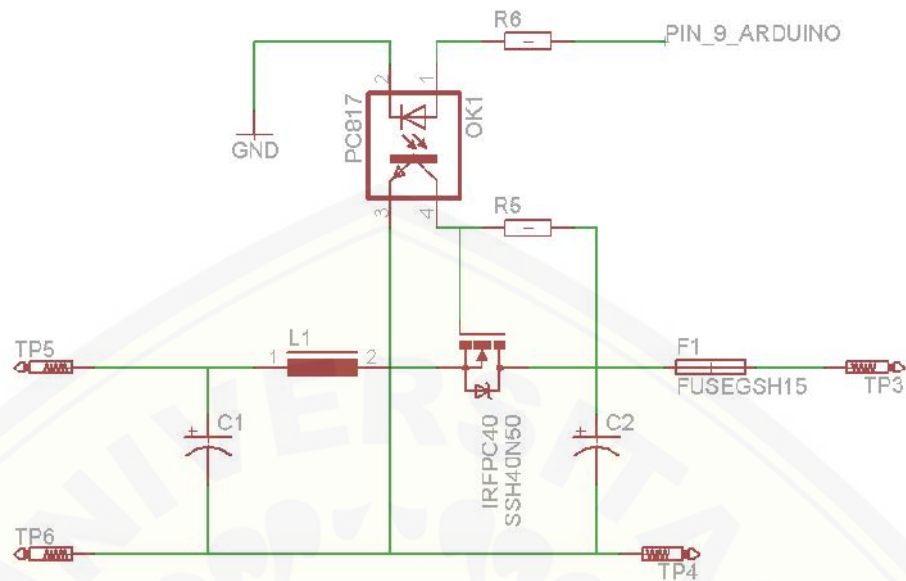


Gambar 3.4 Rangkaian Pengujian Sensor Tegangan

Pengujian menggunakan DC *power supply* sebagai sumber tegangan yang dihubungkan dengan *input* dari sensor tegangan. Kemudian untuk mengetahui hasil pembacaan sensor tegangan digunakan multimeter untuk membandingkan hasil pembacaan dari sensor tegangan.

### 3.5 Buck Converter

*Buck Converter* digunakan sebagai pemberi besarnya eksitasi pada generator sesuai dengan kontrol dari arduino. *Buck converter* telah ditentukan batas maksimum pemberian eksitasi sehingga tidak melebihi eksitasi yang diberikan dengan adanya spesifikasi *buck converter* yang dibuat. Skema rangkaian *buck converter* dapat dilihat pada gambar 3.5



Gambar 3.5 Rangkaian *Buck Converter*

Pada rangkaian *buck converter* terdapat IGBT SSH40N50. IGBT mampu menahan tegangan antara *collector* dan *emitter* hingga 220 V. *Optocoupler* juga ditambahkan pada rangkaian yang berguna untuk memisahkan tegangan yang lebih tinggi dari tegangan yang digunakan pada arduino uno R3 sehingga ketika terjadi kesalahan arduino uno R3 tetap aman. Selain itu juga digunakan sebagai *switching device* untuk IGBT. Resistansi beban pada medan generator DC yang digunakan adalah 914 yang diketahui dengan melakukan pengukuran. Frekuensi yang akan digunakan pada *buck converter* adalah 25Khz. Selain itu terdapat dua kapasitor yang besarnya 470  $\mu\text{F}$ . *Buck converter* dapat bekerja secara kontinu dengan mengetahui perhitungan nilai frekuensi yang digunakan dimana. Dalam menentukan pemilihan induktor (L) dilakukan perhitungan sebagai berikut

$$f > \frac{(1-D)R}{2L}$$

$$L > \frac{(1-D)R}{2f}$$

$$L > \frac{(1-D)914}{2 * 25000}$$

$$L > 0,01828 (1 - D)$$

Sehingga semua nilai *duty cycle* terpenuhi, maka nilai L harus lebih besar dari 0,01828 H. Oleh karena itu, dalam perancangan *buck converter* digunakan nilai L sebesar 22mH.

Setelah itu menentukan nilai kapasitor yang akan digunakan dalam perancangan *buck converter*. Pada perancangan ini diketahui *ripple* tegangan kurang lebih 0,003%. Sehingga

$$0,003 \% = \frac{(1 - D)}{8 L C f^2}$$

Nilai D di asumsikan nol, sehingga

$$0,003 \% = \frac{(1 - 0)}{8 \cdot 22\text{mH} \cdot C \cdot 20\text{Khz}^2}$$

$$C = \frac{1}{8 \cdot 22\text{mH} \cdot 0,003\% \cdot 20\text{Khz}^2}$$

$$C = \frac{1}{8 \cdot 22\text{mH} \cdot 0,003\% \cdot 20\text{Khz}^2}$$

$$C = \frac{1}{2112}$$

$$C = 470 \mu\text{F}$$

### 3.6 Generator DC Penguatan Terpisah

Generator DC penguatan terpisah yang digunakan merupakan tipe *feedback* dengan nomer seri 63-120 yang mempunyai tegangan kerja sebesar 220 V dengan menghasilkan daya 250 Watt. Generator DC penguatan terpisah bekerja pada kecepatan nominal 2000 rpm.

Penggerak mula (*prime mover*) yang digunakan adalah mesin DC seri dengan daya 1 HP atau 0.75 KW, tegangan 220 V dengan kecepatan putar yang digunakan adalah 4500 rpm. Sistem kopel menggunakan *belt* dengan tipe B2340.



Gambar 3.6 Generator DC

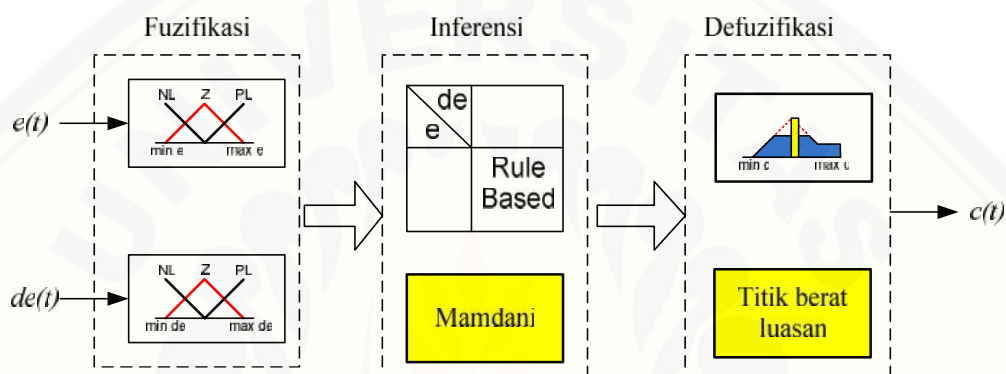
### 3.7 Perancangan Kontrol PID

Perancangan kontrol PID menggunakan *trial and error* yaitu dengan cara *manual tuning*. Langkah dalam melakukan *manual tuning* sebagai berikut.

1. Memberikan *gain* pada *proportional* sehingga *plan* mencapai *set point*. *Proportional* semakin tinggi dapat meningkatkan *overshoot* dan *error steady-state* pada *output*.
2. Memberikan *gain integral* sehingga *error steady-state* dapat dikurangi. *Integral* semakin tinggi dapat meningkatkan *overshoot*.
3. Memberikan *gain derivative* sehingga *overshoot* dapat dikurangi. *Derivative* semakin tinggi dapat membuat sistem menjadi tidak stabil.

### 3.8 Perancangan Kontrol Fuzzy

Pada perancangan fuzzy ini *input* yang akan digunakan adalah 1 parameter yaitu tegangan *output* yang dibaca secara kontinyu oleh sensor tegangan. Dimana nantinya parameter tegangan ini akan digunakan sebagai nilai *error* dan *delta error*. Kemudian parameter *output* yang diinginkan adalah berupa PWM yang akan diberikan kepada *buck converter*. Perancangan kontrol fuzzy melalui tiga tahap yaitu seperti pada gambar 3.7



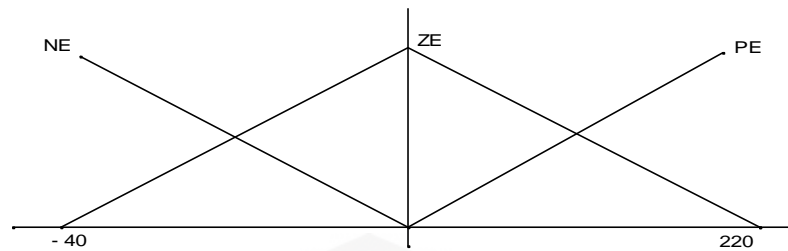
Gambar 3.7 Blok Kendali Fuzzy

#### 1. Fuzzyfikasi

Tahap pertama pada pembuatan fuzzy dimana tahap ini adalah menentukan *membership function*. Pada tahap ini terdapat dua *input*, yaitu *error* dan *delta error*. Dalam sistem pengendalian fuzzy bertujuan untuk mengendalikan tegangan. Oleh karena itu, kendali dilakukan berdasarkan *error* ( $e$ ) tegangan dan perubahan *error* tegangan atau *delta error* tegangan ( $de$ ). Kontroller memperoleh sinyal *input* dari sensor tegangan. Lalu diproses selisih tegangan aktual dengan tegangan referensi menjadi  $e(t)$ . Setelah itu diproses selisih error dengan nilai *error* tegangan sebelumnya,  $e(t-1)$  yang disebut *delta error*  $de(t)$ . Hasil perhitungan *error* dan *delta error* diproses sesuai dengan tahapan blok kendali fuzzy.

Nilai *error* diperoleh dari nilai *setpoint* dikurangi *output* tegangan. *Membership function error* dapat dilihat pada gambar 3.8





Gambar 3.8 Membership Function Error

Pemilihan batas daerah *crisp input* pada *membership function error* didapatkan dari pengujian yang dilakukan pada generator DC penguatan terpisah. Batas *error max* 220 didapatkan dari kondisi dimana ketika keadaan generator DC masih dalam keadaan nol, maka *error* untuk mencapai *set point* adalah sebesar 220. Batas *min error* -40 didapatkan dari kondisi dimana keadaan generator mencapai tegangan *output* 260 Volt ketika dilakukan pengujian sehingga dalam pencapaian *set point* terdapat *error* -40. Perhitungan *membership function error* dalam proses pemrograman pada arduino adalah

a) Jika  $-40 < E < 0$

$$NE_{\_} = \frac{-E}{4}$$

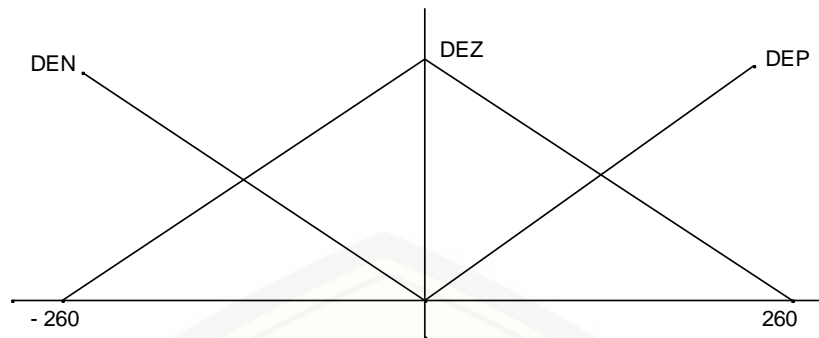
$$ZE = \frac{E+4}{4}$$

b) Jika  $0 < E < 220$

$$ZE_{\_} = \frac{2 - E}{2}$$

$$PE = \frac{-E}{2}$$

Besarnya nilai *delta error* adalah nilai *error* sekarang dikurangi dengan nilai *error* sebelumnya. *Membership function delta error* dapat dilihat pada gambar 3.9



Gambar 3.9 Membership Function Delta Error

Pemilihan batas daerah *crisp input* pada *membership function delta error* didapatkan dari pengujian pada generator DC penguatan terpisah. Batas *delta error max* 260 didapatkan dari kondisi ketika generator mencapai tegangan *output* 260 Volt, maka nilai *error* untuk *mencapai set point* adalah -40. Lalu pada *error* berikutnya adalah generator mencapai tegangan *output* nol sehingga *error* selanjutnya adalah 220. *Delta error* adalah *error* sekarang dikurangi dengan *error* yang lalu, sehingga 260 didapatkan dari 220 dikurangi dengan (-40). Batas *min delta error* adalah -260 didapatkan ketika kondisi generator pada kondisi awal terdapat *error* 220 karena masih dalam keadaan tegangan *output* sama dengan nol. Lalu *error* berikutnya adalah -40 ketika generator mencapai tegangan *output* 260 Volt. Sehingga didapatkan batas *min* -260 dari -40 dikurangi dengan 220. Perhitungan *membership function error* dalam proses pemrograman pada arduino adalah

- a) Jika  $-260 < E < 0$

$$DEN\_ = \frac{-D}{2}$$

$$DEZ\_ = \frac{D + 2}{2}$$

- b) Jika  $-260 < E < 0$

$$DEN\_ = \frac{-D}{2}$$

$$DEZ\_ = \frac{D + 2}{2}$$

- c) Jika  $0 < E < 260$

$$DEZ\_ = \frac{2 - D}{2}$$

$$DEP = \frac{D}{2}$$

2. Inferensi

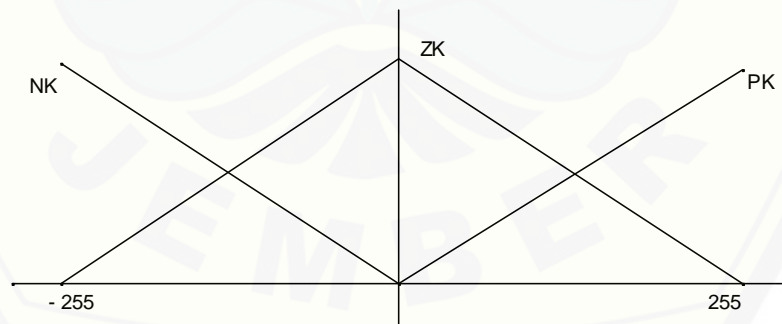
Proses ini adalah memetakan nilai derajat keanggotaan dari masing – masing *error* dan *delta error* pada tabel *rule base*.

	E	PE	ZE	NE
DE				
DEN	U <sub>1</sub> NK	U <sub>2</sub> NK	U <sub>3</sub> PK	
DEZ	U <sub>4</sub> NK	U <sub>5</sub> ZK	U <sub>6</sub> PK	
DEP	U <sub>7</sub> NK	U <sub>7</sub> PK	U <sub>9</sub> PK	

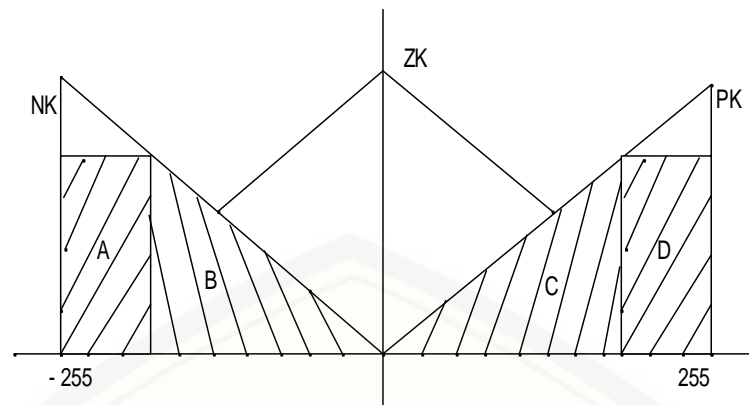
Gambar 3.10 *Rule Base*

3. Defuzzyfikasi

Tahap terakhir pada kontrol fuzzy dimana dilakukan proses pemetaan nilai – nilai U ke himpunan defuzzifikasi yang dinyatakan dengan persamaan – persamaan garis. Defuzzifikasi akan menghasilkan luasan yang dibatasi dengan tinggi masing – masing nilai U dan garis yang dihasilkan dari pemetaan ke *rule base*.



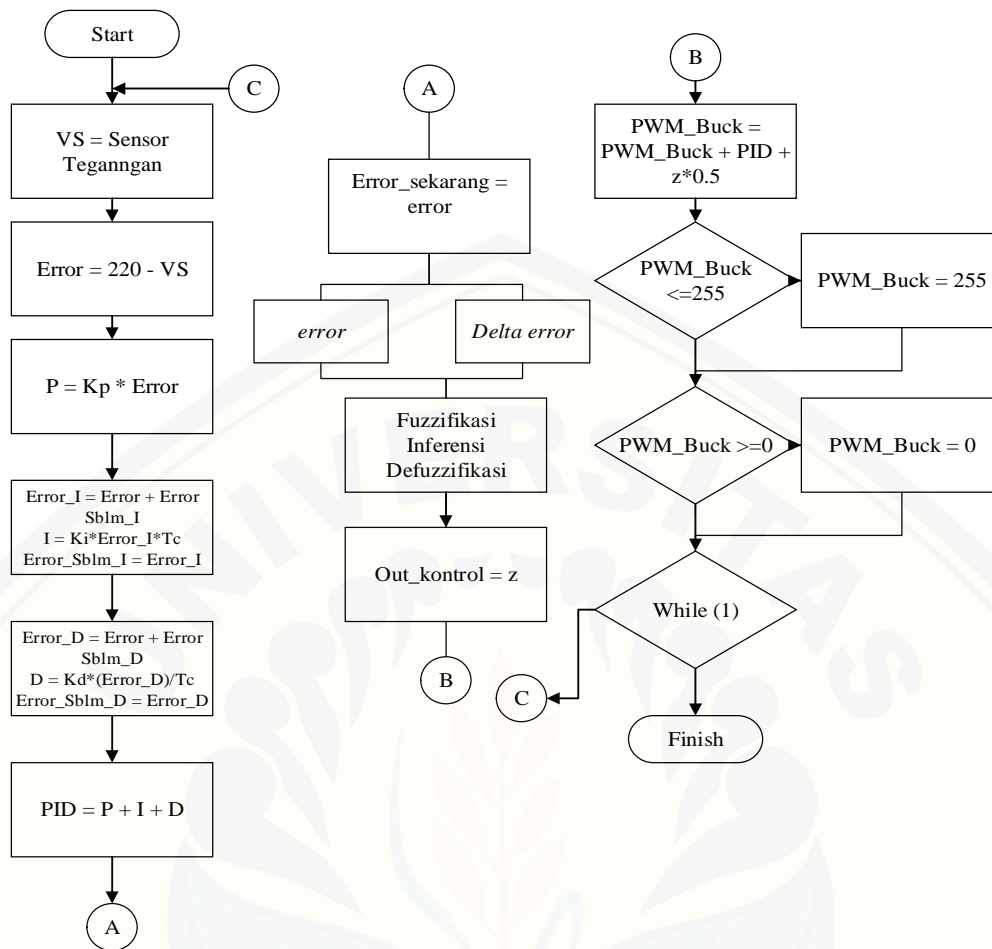
Gambar 3.11 Himpunan Defuzifikasi



Gambar 3.12 Perhitungan Titik Berat Luasan.

### 3.9 Perancangan Program Arduino

Perancangan kontrol *hybrid* PID-fuzzy dilakukan setelah perancangan kontrol PID dan kontrol fuzzy telah dirancang dengan baik sesuai dengan keinginan dalam mencapai *set point*. Kontrol PID dan kontrol fuzzy ditambahkan hasil *output* kontrol sehingga memperoleh hasil yang diinginkan, yaitu memperbaiki kinerja kontrol PID dengan adanya kontrol fuzzy sebagai pembantu dalam memperbaiki *overshoot* dan *disturbance* (gangguan). Berikut adalah *flowchart* arduino kontrol *hybrid* PID-fuzzy dapat dilihat pada gambar 3.13



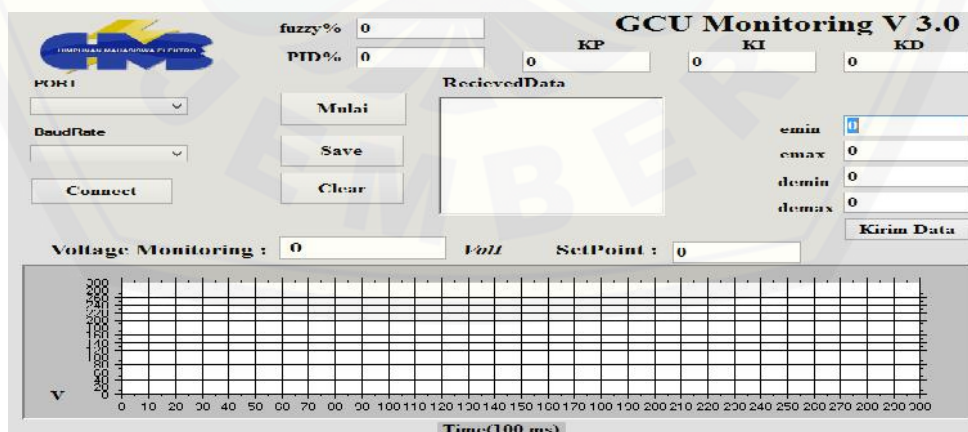
Gambar 3.13 Flowchart Arduino

Pada arduino awal mulanya akan menerima *input* berupa referensi tegangan aktual yang telah dibaca oleh sensor tegangan. Setelah itu hasil pembacaan tadi diterima oleh arduino dan diproses sesuai dengan program yang sudah dibuat. Nilai *error* adalah *set point* dikurangi dengan nilai pembacaan tegangan secara aktual. *Proportional* adalah hasil kali dari  $K_p$  dengan nilai *error*. *Error I* adalah *error* ditambah dengan *error* sebelum *I* dimana *error I* adalah hasil penjumlahan dari *error* sebelumnya secara kontinyu. Sehingga *I* dapat diketahui dengan perkalian dari  $K_i$  *error I* dengan *time sampling*. *Error D* hasil penjumlahan *error* dengan *error* sebelum *D* secara kontinyu. Sehingga nilai *D* dapat diketahui dengan perkalian  $K_d$  dengan *error D* dibagi dengan *time sampling*. Ketika nilai *P*, *I* dan *D* telah diketahui maka ketiganya dijumlahkan. Lalu berlanjut pada proses kontrol fuzzy dimana nilai *error* sekarang sama dengan *error*. Lalu delta *error* adalah *error*

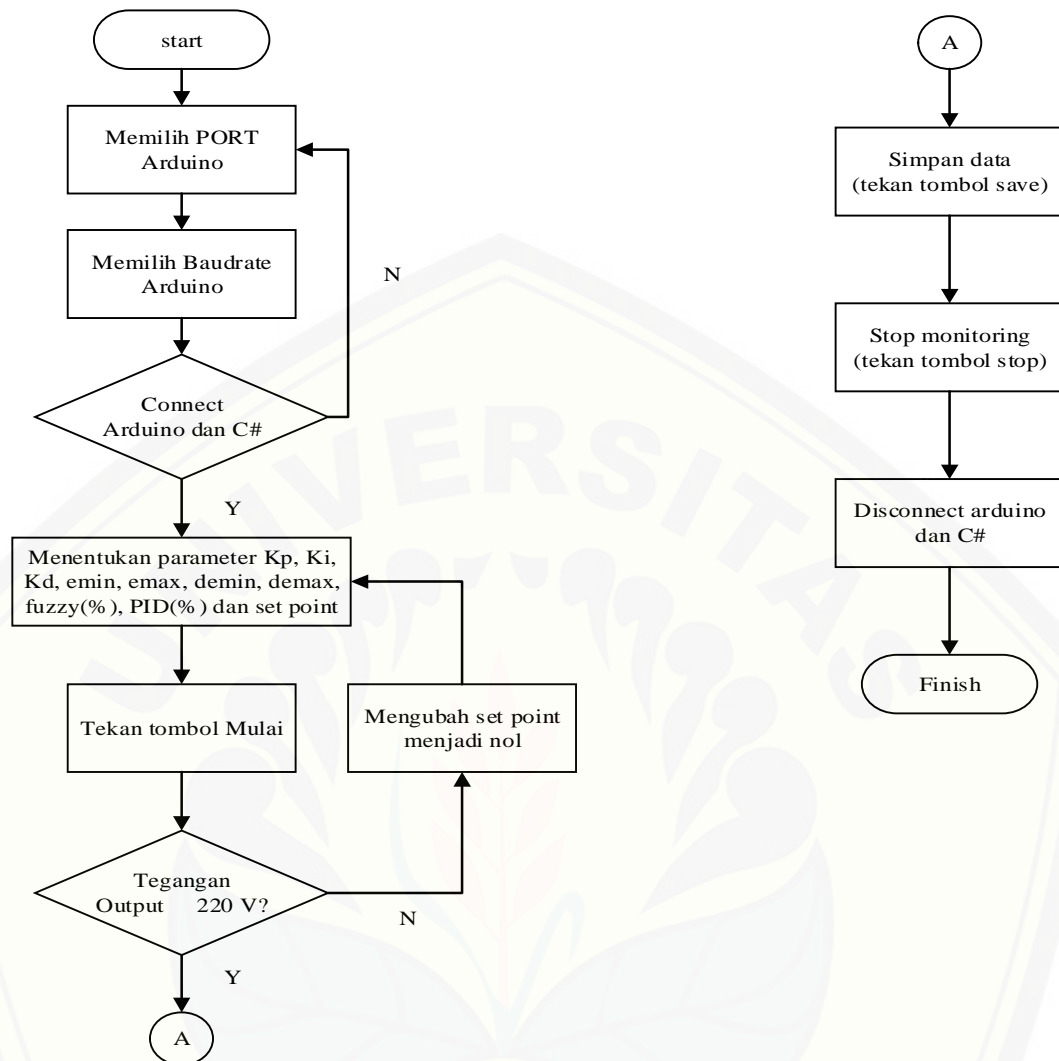
sekarang dikurangi dengan *error* lalu. Dimana nilai *error* lalu didapatkan dari nilai *error* sekarang pada satu tahap sebelumnya secara kontinyu. Dalam tahap selanjutnya masuk kedalam tahap fuzzifikasi, inferensi dan defuzzifikasi dimana secara rinci program dapat dilihat pada lampiran. Penamaan *z* adalah sebagai inisialisasi pada keluaran hasil dari proses fuzzy. Hasil dari keluaran PID dan fuzzy dijumlahkan dengan PWM dari *Buck converter*. Keluaran dari fuzzy yang dihasilkan mempunyai beban yang lebih kecil dari kendali PID (Samsul, 2004). Pada hal ini, fuzzy yang digunakan keluarannya 50%. Hal ini merujuk pada latar belakang bahwa pada sistem ini kendali utama adalah PID sedangkan kendali logika fuzzy bekerja membantu untuk meminimalkan *overshoot* atau *undershoot* serta *recovery time* pada sistem (Samsul, 2004). Selanjutnya, keluaran yang akan diberikan kepada *buck converter* adalah berupa PWM yang akan digunakan untuk menentukan eksitasi yang diperlukan dalam mencapai tegangan *output* sesuai dengan nominal *set point* yaitu 220 Volt.

### 3.10 Aplikasi Monitoring Generator

Aplikasi Monitoring generator pada gambar 3.14 adalah aplikasi yang digunakan untuk memberikan nilai parameter PID, parameter *membership function* dan memantau nilai tegangan *output* pada sistem. Pada aplikasi ini terdapat data *logger* yang digunakan dalam menyimpan data.



Gambar 3.14 Aplikasi Monitoring Generator



Gambar 3.15 Flowchart Aplikasi Monitoring

Mulanya, dalam menjalankan aplikasi monitoring ini adalah menyambungkan arduino pada aplikasi PC yang telah ada aplikasi monitoring. Tahap selanjutnya adalah pemilihan PORT arduino yang digunakan dan pemilihan *baudrate* lalu tekan *connect*. Setelah arduino dan PC telah terkoneksi, tahap selanjutnya adalah memasukkan parameter-parameter yang diperlukan untuk proses pengontrolan dan monitoring lalu tekan mulai. Tegangan *output* akan dapat dipantau pada layar PC pada aplikasi monitoring secara aktual. Hasil dari data yang telah dibaca dapat disimpan pada PC secara keseluruhan. Setelah semua data telah didapatkan, maka proses monitoring dan kontrol dapat dihentikan dengan menekan tombol *stop*. Lalu menghentikan sambungan PC dan arduino dan selesai.

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Setelah melakukan pengujian dan analisa data pada penelitian yang berjudul “Rancang Bangun Kontrol *Hybrid* PID-Fuzzy pada Tegangan *Output* Generator DC Penguatan Terpisah Berbasis Arduino Uno R3” didapatkan kesimpulan sebagai berikut.

1. Kontrol *hybrid* PID-Fuzzy dapat diterapkan dalam mengontrol tegangan *output* generator DC penguatan terpisah agar tetap pada kondisi *set point* ketika diberi beban.
2. Kestabilan tegangan *output* generator DC penguatan terpisah menggunakan kontrol *hybrid* PID-Fuzzy lebih baik dengan nilai *error* tegangan *output* sebesar 0,63% dibandingkan dengan kontrol PID dengan nilai *error* tegangan *output* 1.05% dan kontrol fuzzy 0.71% pada pembebanan 50 Watt
3. Kinerja kontrol *hybrid* PID-Fuzzy lebih baik dengan nilai *peak* lebih kecil yaitu 208 V dibandingkan dengan kontrol PID sebesar 206V serta *recovery time* yang lebih cepat yaitu sebesar 2.2 detik dibandingkan dengan kontrol PID sebesar 3.1 detik dan kontrol fuzzy sebesar 2.7 detik pada pembebanan 65 Watt

### 5.2 Saran

Setelah melaksanakan penelitian, penulis memberikan saran untuk mengembangkan penelitian ini. Saran yang dapat diberikan adalah sebagai berikut.

1. *Buck Converter* dapat diganti dengan *boost converter* atau *buck-boost converter*.
2. Metode ziegler nichols 1 atau ziegler nichols 2 dapat digunakan dalam mencari nilai parameter PID.
3. *Plan* yang digunakan dapat diganti dengan generator AC atau generator dengan jenis penguatan lainnya.
4. Pada kontrol fuzzy menggunakan dua parameter input atau lebih untuk membandingkan dengan 2 input 1 parameter tegangan, seperti parameter tegangan dan kecepatan motor.



**DAFTAR PUSTAKA**

- Arduino UNO, R3. 2015. "A000066 Arduino | Mouser." *MOUSER ELECTRONICS*. Accessed November 3.
- Chapman. (2005). *Electric Machinery Fundamentals*. New York: Library of Congress Cataloging.
- Hameyer, Kay.2001. *Electrical Machines*. Univ. RWTH Aachen. Bundesland NRW . Jerman
- Gunawan. 2009. Rancang Bangun DC-DC Buck Converter dengan PID Diskrit Sebagai Pengendali Tegangan Keluaran. Depok.  
<http://surplustek.ca/dct-rsa-capteur-de-voltage-standard-pour-robot-arduino.html>.  
<https://fahmizaleeits.wordpress.com/category/kuliah-kontrol/fuzzy-logic/>
- Kusumadewi, S., & Hartati, S. (2006). *NEURO FUZZY: Integrasi Sistem Fuzzy & Jaringan Saraf*. Yogyakarta: Graha Ilmu.
- Setiawan, Andi.2013. *Kontrol PID Pengendali Sistem*. Universitas Jember. Indonesia.
- Srivastava, Piryanka., Singh, Shri,S.k. Hybrid Fuzzy-PID controller in buck-boost coverter.IJAERT.2014
- Fattah, Ahmed. 2009. Design and Analysis of Speed Control Using Hybrid PID-Fuzzy Controller for Induction Motors.2015.Master's Theses. Paper 595
- Kurniawan, Andi., Apriwansyah, Andi. Aplikasi Kontroller Hybrid Fuzzy-PID Pada Kendali Kecepatan Putar Motor DC. eLEKTRIKA ISSN 2085-0565, vol 1 no.1 33-48
- Samsul, Bachri. 2004. Sistem Kendali Hybrid PID - Logika Fuzzy Pada Pengaturan Kecepatan Motor DC. MAKARA,TEKNOLOGI. Vol 8 No.1 25 – 34
- Hanchevici, Bogdan,. Monica Patrascu., Dumitrache Ioan. 2012. A Hybrid PID – Fuzzy Control for Linear SISO Systems with Variant Communication Delays. Hindawi Publishing Corporation. Volume 2012, Article ID 217068, 8 pages

## LAMPIRAN

### A. Listing Program Arduino Uno R3

```
String textFromSerial = "";
boolean parsing = false;
String inData = "";
String data[15];
float data_masuk = 0;

float V = 0.00;
float error = 0.00;
float setpoint = 0;

//////////////////// Fuzzy Inialisasi
float persenFuzzy = 0.00;
float delta_error = 0.00;
float error_sekarang;
float error_lalu;
float emin=0;
float emax=0;
float demin=0;
float demax=0;

float U1,U2,U3,U4,U5,U6,U7,U8,U9;
float Z = 0.00;

float NE_, ZE, ZE_, PE;
float DEN_, DEZ, DEZ_, DEP;

float LA,XA,LB,XB;
int WANK1,WANK2,WANK4,WANK7;

float LC,XC,LD,XD;
int WAPK3,WAPK6,WAPK8,WAPK9;

int WAZK;

int out_kontrol;
//////////////////// PID Inialisasi
float persenPID = 0.00;
float KP=0;
float KI=0;
float KD=0;
float error_sblm_I=0;
float error_sblm_D=0;
float Tc=0.1;
int error_I;
int error_D;
float out_P;
float out_I;
float out_D;
```

```
int out_PID;
////////////////////////////////////

float sens =0.00;
int tegangan = 0;

int pwm;
int con;

void akses_tegangan() {
    sens = (float)analogRead(A0)/1023;
    tegangan = sens * 400;
}

//////////////////////////////////// Fuzzy
void fuzzyfikasi(){

    U1 = 0;
    U2 = 0;
    U3 = 0;
    U4 = 0;
    U5 = 0;
    U6 = 0;
    U7 = 0;
    U8 = 0;
    U9 = 0;

    V = tegangan;
    error = setpoint - V;

    if(error < emin){
        error = emin;
    }
    if(error > emax){
        error = emax;
    }

    error_sekarang = error;
    delta_error = error_sekarang - error_lalu;

    if(delta_error < demin){
        delta_error = demin;
    }
    if(delta_error > demax){
        delta_error = demax;
    }

    error_lalu = error_sekarang;

    if(error < 0 && error >= emin){
        NE_ = (-error)/abs (emin);
        ZE = (error + abs (emin))/abs (emin);

        if(delta_error < 0 && delta_error >= demin){
```

```

DEN_ = (-delta_error)/abs (demin);
DEZ = (delta_error + abs (demin))/abs (demin);

U1 = min(NE_,DEN_);
U4 = min(NE_,DEZ);
U2 = min(ZE,DEN_);
U5 = min(ZE,DEZ);
}
if(delta_error >= 0 && delta_error <= demax){
  DEZ_ = (demax - delta_error)/demax;
  DEP = delta_error/demax;

  U4 = min(NE_,DEZ_);
  U7 = min(NE_,DEP);
  U5 = min(ZE,DEZ_);
  U8 = min(ZE,DEP);
}
}

if(error >= 0 && error <= emax){
  ZE_ = (emax - error)/emax;
  PE = error/emax;

  if(delta_error < 0 && delta_error >= demin){
    DEN_ = (-delta_error)/abs (demin);
    DEZ = (delta_error + abs (demin))/abs (demin);

    U2 = min(ZE_,DEN_);
    U5 = min(ZE_,DEZ);
    U3 = min(PE,DEN_);
    U6 = min(PE,DEZ);
  }
  if(delta_error >= 0 && delta_error <= demax){
    DEZ_ = (demax - delta_error)/demax;
    DEP = delta_error/demax;

    U5 = min(ZE_,DEZ_);
    U8 = min(ZE_,DEP);
    U6 = min(PE,DEZ_);
    U9 = min(PE,DEP);
  }
}

LA = ((1 - U1)* 255) * U1;
XA = ((1 - U1)* -255)/2 + (-255*U1);
LB = 0.5*U1*(255*U1);
XB = ((U1 * -255)*2)/3;
WANK1 = (LA*XA + LB*XB)/(LA + LB);

LA = ((1 - U2)* 255)*U2;
XA = ((1 - U2)* -255)/2 + (-255*U2);
LB = 0.5*U2*(255*U2);
XB = ((U2 * -255)*2)/3;
WANK2 = (LA*XA + LB*XB)/(LA + LB);

```

```

LA = ((1 - U4)* 255)*U4;
XA = ((1 - U4)* -255)/2 + (-255*U4);
LB = 0.5*U4*(255*U4);
XB = ((U4 * -255)*2)/3;
WANK4 = (LA*XA + LB*XB)/(LA + LB);

LA = ((1 - U7)* 255)*U7;
XA = ((1 - U7)* -255)/2 + (-255*U7);
LB = 0.5*U7*(255*U7);
XB = ((U7 * -255)*2)/3 ;
WANK7 = (LA*XA + LB*XB)/(LA + LB);

LC = 0.5*U3*(255*U3);
XC = ((255*U3)*2)/3;
LD = ((1 - U3)*255)*U3;
XD = ((1 - U3)*255)/2 + (255*U3);
WAPK3 = (LC*XC + LD*XD)/(LC + LD);

LC = 0.5*U6*(255*U6);
XC = ((255*U6)*2)/3;
LD = ((1 - U6)*255)*U6;
XD = ((1 - U6)*255)/2 + (255*U6);
WAPK6 = (LC*XC + LD*XD)/(LC + LD);

LC = 0.5*U8*(255*U8);
XC = ((255*U8)*2)/3;
LD = ((1 - U8)*255)*U8;
XD = ((1 - U8)*255)/2 + (255*U8);
WAPK8 = (LC*XC + LD*XD)/(LC + LD);

LC = 0.5*U9*(255*U9);
XC = ((255*U9)*2)/3;
LD = ((1 - U9)*255)*U9;
XD = ((1 - U9)*255)/2 + (255*U9);
WAPK9 = (LC*XC + LD*XD)/(LC + LD);

WAZK = 0;

Z = ((WANK1*U1) + (WANK2*U2) + (WAPK3*U3) + (WANK4*U4) + (WAZK*U5) +
(WAPK6*U6) + (WANK7*U7) + (WAPK8*U8) + (WAPK9*U9))/(U1 + U2 + U3 + U4 + U5 +
U6 + U7 + U8 + U9);
out_kontrol = Z*persenFuzzy;
}

//////////////////////////////////// PID
void kontrol_PID() {
// error = setpoint - tegangan;
if(error>=200)
{
error=200;
}
if(error<=(-200))
{
error=-200;
}
}

```

```
out_P = KP*error;

error_I = error + error_sblm_I;
out_I = KI*error_I*Tc;
error_sblm_I = error_I;

error_D = error-error_sblm_D;
out_D = (KD*error_D)/Tc;
error_sblm_D = error_D;

out_PID = out_P + out_I + out_D;
out_PID = out_PID*persenPID;
}
/////////////////////// Kontrol
void kontrol() {
  pwm = pwm + out_kontrol + out_PID;
  if(pwm >= 255)
  {
    pwm = 255;
  }
  if(pwm <= 0)
  {
    pwm=0;
  }
  con = 255 - pwm;
  analogWrite(9, con);
}

void setup(){
  Serial.begin(9600);
  pinMode(9,OUTPUT);
  analogWrite(9,255);
}

void loop(){
  terima_serial();

  if(data_masuk==1){

    akses_tegangan();
    fuzzyfikasi();
    kontrol_PID();
    kontrol();
    kirim_data_ke_PC();

  }else if(data_masuk == 0){

  }

}

void terima_serial(){
  while(Serial.available()>0) {
    char inChar = (char)Serial.read();
```

```
if(inChar == '\n'){
    parsing = true;
}else{
    inData +=inChar;
}
}

if(parsing){
    int j=0;
    data[j] = "";
    for(int i=0;i<inData.length();i++){
        if(inData[i]== ' '){
            j++;
            data[j] = "";
        }else{
            data[j] += inData[i];
        }
    }
}

data_masuk = data[0].toFloat();
emin      = data[1].toFloat();
emax      = data[2].toFloat();
demin     = data[3].toFloat();
demax     = data[4].toFloat();
KP        = data[5].toFloat();
KI        = data[6].toFloat();
KD        = data[7].toFloat();

persenFuzzy = data[8].toFloat();
persenPID   = data[9].toFloat();
setpoint    = data[10].toFloat();

parsing=false;
inData = "";
}
}

void kirim_data_ke_PC(){
    //Serial.print(setpoint);
    //Serial.print("\t");
    Serial.print(tegangan);
    Serial.print("\n");
    //Serial.print(error);
    //Serial.print("\t");
    //Serial.print(delta_error);
    //Serial.print("\t");
    //Serial.print(Z);
    //Serial.print("\t");
    //Serial.print(pwm);
    //Serial.print("\n");
    delay(100);
}
}
```

**B. Listing Program Monitoring C#**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.IO.Ports;
using System.IO;

namespace GCU_Monitoring
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                string[] portNames = System.IO.Ports.SerialPort.GetPortNames();
                for (int i = 0; i <= portNames.Length - 1; i++)
                {
                    comboBox1.Items.Add(portNames[i]);
                }
                comboBox2.Items.Add("110");
                comboBox2.Items.Add("300");
                comboBox2.Items.Add("600");
                comboBox2.Items.Add("1200");
                comboBox2.Items.Add("2400");
                comboBox2.Items.Add("4800");
                comboBox2.Items.Add("9600");
                comboBox2.Items.Add("14400");
                comboBox2.Items.Add("19200");
                comboBox2.Items.Add("38400");
                comboBox2.Items.Add("56000");
                comboBox2.Items.Add("115200");
            }

            private void Form1_FormClosing(object sender, FormClosingEventArgs
e)
            {
                serialPort1.Close();
            }

            private void button1_Click(object sender, EventArgs e)
            {
                if (!serialPort1.IsOpen)
                {
                    if (comboBox1.Text.Length > 0 && comboBox2.Text.Length > 0)
                    {
```



```

        serialPort1.PortName = comboBox1.Text;
        serialPort1.BaudRate = Convert.ToInt32(comboBox2.Text);
        serialPort1.Open();
        button1.Text = "disConnect";
    }
    else
    {
        MessageBox.Show("Please Choice PORT and BaudRate !!!");
    }
}
else
{
    serialPort1.Close();
    button1.Text = "Connect";
}
}

private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    Invoke(new EventHandler(bacaData));
}

string indata;
double detik=0;
private void bacaData(object sender, EventArgs e)
{
    indata = serialPort1.ReadLine();
    string[] data = indata.Split(' ');
    textBox1.Text = data[0];
    richTextBox1.AppendText(data[0] + "\n");
    detik++;
    axMathworks_Strip1.AddXY(0, detik, Convert.ToDouble(data[0]));
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    richTextBox1.AllowDrop = true;
    richTextBox1.ScrollToCaret();
}

private void button2_Click(object sender, EventArgs e)
{
    richTextBox1.SaveFile("D:\\logger\\logger.txt",
RichTextBoxStreamType.PlainText);
}

private void button3_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (Convert.ToInt32(textBox2.Text) <= 220 && textBox2.Text
!= "" && textBox5.Text != "" && textBox4.Text != "" && textBox3.Text !=
"" && textBox6.Text != "" && tbxKP.Text != "" && tbxKI.Text != "" &&
tbxKD.Text != "" && tbxFUZZY.Text != "" && tbxPIDz.Text != "" && textBox2.Text
!= "")
        {

```

```

if (button4.Text == "Stop")
{
    serialPort1.Write("1" + " "); //Mulai
    serialPort1.Write(textBox5.Text + " "); //emin
    serialPort1.Write(textBox4.Text + " "); //emax
    serialPort1.Write(textBox3.Text + " "); //demin
    serialPort1.Write(textBox6.Text + " "); //demax
    serialPort1.Write(tbxKP.Text + " "); //KP
    serialPort1.Write(tbxKI.Text + " "); //Ki
    serialPort1.Write(tbxKD.Text + " "); //Kd
    serialPort1.Write(tbxFUZZY.Text + " "); //fuzzy
    serialPort1.Write(tbxPIDz.Text + " "); //pid persen

    //serialPort1.Write(textBox7.Text + " "); //V
    serialPort1.Write(textBox2.Text + "\n"); //setPoint
}
else if (button4.Text == "Mulai")
{
    MessageBox.Show("Can't Transfer Data because Serial
Stopped !!!");
}
else
{
    MessageBox.Show("SetPoint Over Load > 220 !!! Or No
data transfer KP,KI,KD or setPoint");
}
else
{
    MessageBox.Show("No Connection !!!");
}
}

private void button4_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (button4.Text == "Mulai")

```

```
{

    serialPort1.Write("1" + " "); //Mulai
    serialPort1.Write (textBox5.Text + " "); //emin

    serialPort1.Write(textBox4.Text + " "); //emax

    serialPort1.Write(textBox3.Text + " "); //demin

    serialPort1.Write(textBox6.Text + " "); //demax

    serialPort1.Write(tbxKP.Text + " "); //KP

    serialPort1.Write(tbxKI.Text + " "); //Ki

    serialPort1.Write(tbxKD.Text + " "); //Kd

    serialPort1.Write(tbxFUZZY.Text + " "); //fuzzy persen

    serialPort1.Write(tbxPIDz.Text + " "); //pid persen

    //serialPort1.Write(textBox7.Text + " "); //V
    serialPort1.Write(textBox2.Text + "\n"); //setPoint
    button4.Text = "Stop";
}
else if (button4.Text == "Stop")
{

    serialPort1.Write("0" + " "); //Mulai
    serialPort1.Write(textBox5.Text + " "); //emin

    serialPort1.Write(textBox4.Text + " "); //emax

    serialPort1.Write(textBox3.Text + " "); //demin

    serialPort1.Write(textBox6.Text + " "); //demax

    serialPort1.Write(tbxKP.Text + " "); //KP

    serialPort1.Write(tbxKI.Text + " "); //Ki

    serialPort1.Write(tbxKD.Text + " "); //Kd

    serialPort1.Write(tbxFUZZY.Text + " "); //fuzzy persen

    serialPort1.Write(tbxPIDz.Text + " "); //pid persen

    //serialPort1.Write(textBox7.Text + " "); //V
    serialPort1.Write(textBox2.Text + "\n"); //setPoint
    button4.Text = "Mulai";
}
```

```
    }  
    else  
    {  
        MessageBox.Show("No Connection !!!");  
    }  
}  
  
private void button5_Click(object sender, EventArgs e)  
{  
    richTextBox1.Text = "";  
    textBox1.Text = "0";  
}  
  
private void axMathworks_Strip1_CursorChange(object sender,  
AxMathworks_STRIPLib._DStripEvents_CursorChangeEvent e)  
{  
}  
}  
}
```

### C. Data Pemberian Beban

#### C1. Pemberian Beban 15 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	219	220
2	2	220	218	220
3	3	220	219	219
4	4	220	219	219
5	5	220	219	220
6	6	220	220	220
7	7	220	220	219
8	8	220	220	220
9	9	221	220	220
10	10	220	220	219
11	11	220	220	220
12	12	220	220	219
13	13	216	219	217
14	14	216	215	219
15	15	216	218	217
16	16	219	215	219
17	17	217	218	218
18	18	219	217	218
19	19	218	218	220
20	20	220	219	219
21	21	220	220	220
22	22	218	219	218
23	23	220	219	219
24	24	218	220	220
25	25	220	220	218
26	26	219	218	220
27	27	218	220	218
28	28	221	219	220
29	29	219	219	220
30	30	221	218	219
31	31	218	219	220

32	32	221	220	218
33	33	219	219	220
34	34	220	220	220
35	35	221	218	220
36	36	220	220	221
37	37	218	219	218
38	38	218	220	220
39	39	221	221	218
40	40	218	218	219
41	41	220	220	220
42	42	219	218	219
43	43	220	220	221
44	44	221	219	218
45	45	218	220	220
46	46	221	221	218
47	47	219	218	220
48	48	220	221	221
49	49	219	218	218
50	50	220	219	220
Error Persen		0,43%	0,38%	0,37%

## C2 Pemberian Beban 25 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	218	220
2	2	220	219	220
3	3	220	219	220
4	4	219	220	219
5	5	220	220	219
6	6	220	220	220
7	7	220	220	220
8	8	220	220	220
9	9	220	220	220
10	10	220	220	220
11	11	220	220	219
12	12	220	220	220
13	13	220	219	220
14	14	220	219	220
15	15	215	220	220
16	16	214	213	219
17	17	217	216	214
18	18	219	218	217
19	19	216	215	216
20	20	218	217	218
21	21	217	215	216
22	22	218	217	218
23	23	218	217	219
24	24	218	217	218
25	25	221	219	219
26	26	218	218	220
27	27	220	218	218
28	28	217	220	220
29	29	220	219	218
30	30	217	219	220
31	31	220	218	217
32	32	218	219	220

33	33	220	220	218
34	34	220	219	220
35	35	219	218	219
36	36	220	219	220
37	37	219	218	220
38	38	220	220	219
39	39	218	219	220
40	40	221	218	219
41	41	218	220	219
42	42	220	218	218
43	43	217	220	219
44	44	221	218	220
45	45	218	220	219
46	46	220	219	221
47	47	218	220	218
48	48	220	218	220
49	49	219	220	217
50	50	219	219	219
Error Persen		0,47%	0,43%	0,40%



## C3 Pemberian Beban 40 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	221	220	220
2	2	221	220	219
3	3	220	221	220
4	4	220	220	220
5	5	220	220	220
6	6	220	219	220
7	7	220	219	220
8	8	220	219	219
9	9	220	219	219
10	10	220	219	219
11	11	220	220	219
12	12	220	220	220
13	13	220	213	220
14	14	211	215	220
15	15	214	212	219
16	16	214	215	212
17	17	216	212	216
18	18	217	215	213
19	19	215	218	217
20	20	218	217	216
21	21	216	215	217
22	22	218	218	218
23	23	215	215	216
24	24	218	218	218
25	25	216	216	215
26	26	218	219	218
27	27	216	217	215
28	28	219	219	218
29	29	217	217	217
30	30	219	219	218
31	31	217	218	217
32	32	219	219	218

33	33	217	220	218
34	34	219	220	220
35	35	218	218	221
36	36	220	219	219
37	37	218	219	219
38	38	219	219	220
39	39	220	219	220
40	40	220	218	219
41	41	218	220	219
42	42	220	218	220
43	43	218	220	220
44	44	221	218	218
45	45	218	221	220
46	46	221	218	219
47	47	218	221	220
48	48	221	218	218
49	49	218	221	220
50	50	221	218	218
Error Persen		0,52%	0,53%	0,32%

## C4 Pemberian Beban 50 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	220	220
2	2	220	220	220
3	3	220	220	219
4	4	220	220	219
5	5	220	220	219
6	6	220	220	219
7	7	220	216	219
8	8	212	211	210
9	9	212	214	214
10	10	215	212	214
11	11	214	214	215
12	12	215	213	216
13	13	215	213	214
14	14	216	215	217
15	15	217	213	214
16	16	215	216	217
17	17	217	213	215
18	18	216	216	217
19	19	218	213	215
20	20	215	216	217
21	21	218	214	215
22	22	216	216	217
23	23	218	214	216
24	24	216	216	218
25	25	217	215	216
26	26	217	217	218
27	27	218	214	216
28	28	217	217	218
29	29	217	215	218
30	30	218	217	220
31	31	217	215	218
32	32	218	217	219

33	33	217	216	220
34	34	218	219	218
35	35	217	220	218
36	36	219	218	220
37	37	217	220	217
38	38	226	217	219
39	39	219	218	217
40	40	219	219	219
41	41	218	224	218
42	42	217	216	219
43	43	224	219	218
44	44	218	221	219
45	45	220	220	220
46	46	218	219	218
47	47	225	218	222
48	48	218	219	220
49	49	220	218	218
50	50	218	221	218
Error Persen		1,05%	0,71%	0,63%

## C5 Pemberian Beban 65 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	219	220
2	2	220	219	220
3	3	220	218	220
4	4	220	219	220
5	5	220	219	220
6	6	220	219	221
7	7	220	220	220
8	8	213	220	208
9	9	211	216	212
10	10	206	213	215
11	11	213	210	215
12	12	211	212	215
13	13	213	211	216
14	14	211	212	215
15	15	211	211	216
16	16	213	211	215
17	17	214	211	213
18	18	213	211	216
19	19	216	211	213
20	20	213	211	216
21	21	215	211	214
22	22	214	212	216
23	23	216	211	214
24	24	214	213	216
25	25	215	211	215
26	26	215	213	217
27	27	215	211	215
28	28	215	216	218
29	29	213	218	220
30	30	216	215	220
31	31	213	213	220
32	32	216	222	221

33	33	217	218	215
34	34	217	219	221
35	35	218	218	215
36	36	219	220	217
37	37	218	218	215
38	38	219	215	221
39	39	220	220	215
40	40	219	215	221
41	41	224	219	217
42	42	219	221	221
43	43	218	215	215
44	44	224	221	220
45	45	216	218	218
46	46	224	219	219
47	47	216	218	221
48	48	220	218	215
49	49	222	221	221
50	50	218	215	215
Error Persen		1,12%	1,00%	1,05%

**D. Data Pelepasan Beban****D1 Pelepasan Beban 15 Watt**

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	221	217	218
2	2	218	220	220
3	3	220	218	218
4	4	219	220	220
5	5	220	221	220
6	6	224	219	219
7	7	224	221	221
8	8	224	218	218
9	9	224	220	221
10	10	224	218	220
11	11	222	219	223
12	12	221	220	222
13	13	221	218	221
14	14	221	221	220
15	15	220	217	220
16	16	220	220	220
17	17	221	219	221
18	18	220	224	221
19	19	220	222	221
20	20	220	221	220
21	21	220	220	220
22	22	220	219	220
23	23	220	219	220
24	24	220	220	220
25	25	221	220	220
26	26	221	221	220
27	27	220	220	220
28	28	220	220	220
29	29	221	220	220
30	30	220	220	220
31	31	220	220	220
32	32	220	220	220

33	33	221	220	221
34	34	220	220	220
35	35	220	220	220
36	36	220	221	219
37	37	220	221	219
38	38	220	220	220
39	39	220	220	220
40	40	220	219	220
41	41	220	219	220
42	42	220	218	219
43	43	220	219	219
44	44	220	219	219
45	45	220	219	219
46	46	220	220	220
47	47	220	220	220
48	48	220	220	220
49	49	220	220	220
50	50	220	220	220
Error Persen		0,06%	0,18%	0,12%



## D2 Pelepasan Beban 25 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	221	219	220
2	2	218	222	219
3	3	220	218	221
4	4	219	221	218
5	5	220	218	220
6	6	224	225	218
7	7	224	225	224
8	8	224	225	224
9	9	224	225	222
10	10	224	225	222
11	11	222	224	220
12	12	221	223	221
13	13	221	223	220
14	14	221	222	220
15	15	220	221	220
16	16	220	221	220
17	17	221	221	220
18	18	220	221	220
19	19	220	220	220
20	20	220	220	220
21	21	220	221	219
22	22	220	220	220
23	23	220	220	220
24	24	220	221	220
25	25	221	221	219
26	26	221	220	219
27	27	220	220	219
28	28	220	220	220
29	29	221	221	220
30	30	220	220	220
31	31	220	220	220
32	32	220	220	220

33	33	221	221	220
34	34	220	220	220
35	35	220	220	220
36	36	220	220	220
37	37	220	220	220
38	38	220	220	220
39	39	220	220	220
40	40	220	220	220
41	41	220	220	220
42	42	220	220	220
43	43	220	220	219
44	44	220	220	220
45	45	220	221	220
46	46	220	220	219
47	47	220	220	220
48	48	220	221	220
49	49	220	220	220
50	50	220	220	219
Error Persen		0,10%	0,18%	0,09%

## D3 Pelepasan Beban 40 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	219	221
2	2	221	220	219
3	3	220	219	221
4	4	221	220	218
5	5	220	219	221
6	6	220	220	218
7	7	219	218	220
8	8	220	220	220
9	9	222	226	226
10	10	225	226	226
11	11	227	225	226
12	12	227	224	224
13	13	229	221	222
14	14	228	220	221
15	15	228	220	220
16	16	227	220	220
17	17	227	220	220
18	18	226	221	220
19	19	225	221	221
20	20	224	221	220
21	21	224	221	220
22	22	222	220	219
23	23	221	220	219
24	24	222	219	220
25	25	221	219	220
26	26	220	220	220
27	27	220	220	220
28	28	221	220	220
29	29	221	221	220
30	30	220	221	220
31	31	220	220	219
32	32	221	219	220

33	33	220	219	219
34	34	220	219	220
35	35	220	219	220
36	36	221	219	220
37	37	220	220	221
38	38	220	220	220
39	39	220	220	220
40	40	221	219	220
41	41	220	219	220
42	42	220	220	220
43	43	220	220	220
44	44	221	220	219
45	45	220	221	219
46	46	220	220	219
47	47	220	220	219
48	48	220	220	220
49	49	220	219	220
50	50	220	219	220
Error Persen		0,11%	0,22%	0,13%

## D4 Pelepasan Beban 50 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	220	221	215
2	2	218	219	218
3	3	221	218	217
4	4	218	219	219
5	5	223	220	218
6	6	219	216	218
7	7	221	216	218
8	8	220	215	217
9	9	220	217	219
10	10	219	216	217
11	11	218	220	224
12	12	229	216	226
13	13	228	228	226
14	14	229	225	226
15	15	230	222	225
16	16	230	221	222
17	17	229	222	222
18	18	229	222	220
19	19	228	222	220
20	20	227	221	220
21	21	226	221	220
22	22	225	221	220
23	23	223	221	219
24	24	222	220	220
25	25	222	220	220
26	26	222	221	220
27	27	221	221	220
28	28	220	220	220
29	29	220	221	220
30	30	221	220	220
31	31	221	220	220
32	32	220	220	219

33	33	221	219	219
34	34	221	220	219
35	35	220	220	219
36	36	220	220	219
37	37	221	221	220
38	38	221	221	220
39	39	220	220	220
40	40	220	219	220
41	41	220	219	220
42	42	220	219	219
43	43	220	218	220
44	44	220	219	220
45	45	221	219	221
46	46	220	220	221
47	47	220	220	220
48	48	220	220	220
49	49	220	220	218
50	50	221	220	218
Error Persen		0,16%	0,22%	0,18%

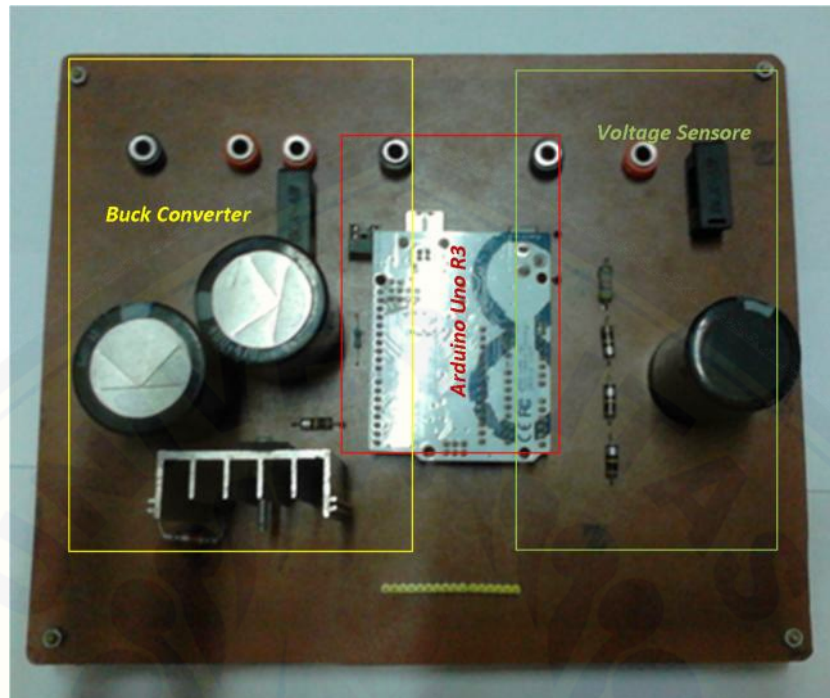
## D5 Pelepasan Beban 65 Watt

No	Waktu(ms)	Tegangan Output		
		PID	FUZZY	HYBRID PID-FUZZY
1	1	213	218	217
2	2	222	220	219
3	3	215	219	222
4	4	220	219	212
5	5	223	220	220
6	6	213	220	217
7	7	223	229	219
8	8	215	223	217
9	9	228	219	219
10	10	226	220	222
11	11	224	222	224
12	12	225	222	224
13	13	226	223	225
14	14	227	223	224
15	15	228	222	224
16	16	228	221	224
17	17	228	220	222
18	18	227	220	221
19	19	227	220	221
20	20	226	221	221
21	21	225	221	220
22	22	224	220	220
23	23	224	221	221
24	24	222	220	220
25	25	221	221	220
26	26	221	220	220
27	27	221	220	221
28	28	221	219	220
29	29	220	219	220
30	30	221	219	220
31	31	221	220	219
32	32	220	220	219

33	33	221	221	220
34	34	221	221	220
35	35	221	220	220
36	36	220	220	220
37	37	220	219	221
38	38	221	218	221
39	39	221	219	219
40	40	220	219	219
41	41	220	219	219
42	42	221	220	219
43	43	221	220	219
44	44	220	221	219
45	45	220	220	220
46	46	220	220	220
47	47	220	219	220
48	48	220	219	220
49	49	220	219	220
50	50	220	219	220
Error Persen		0,19%	0,27%	0,18%



**E. Buck Converter dan Sensor Tegangan**



**F. Generator DC Penguatan Terpisah**



**G. Beban Lampu Pijar**



**H. Gambar Keseluruhan Sistem**

