



**SISTEM KONTROL *FUZZY LOGIC* PADA GENERATOR DC
PENGUATAN TERPISAH BERBASIS
ARDUINO UNO R3**

SKRIPSI

oleh

**Hendro Rosyidi Setiyawan
NIM 121910201065**

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



**SISTEM KONTROL *FUZZY LOGIC* PADA GENERATOR DC
PENGUATAN TERPISAH 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

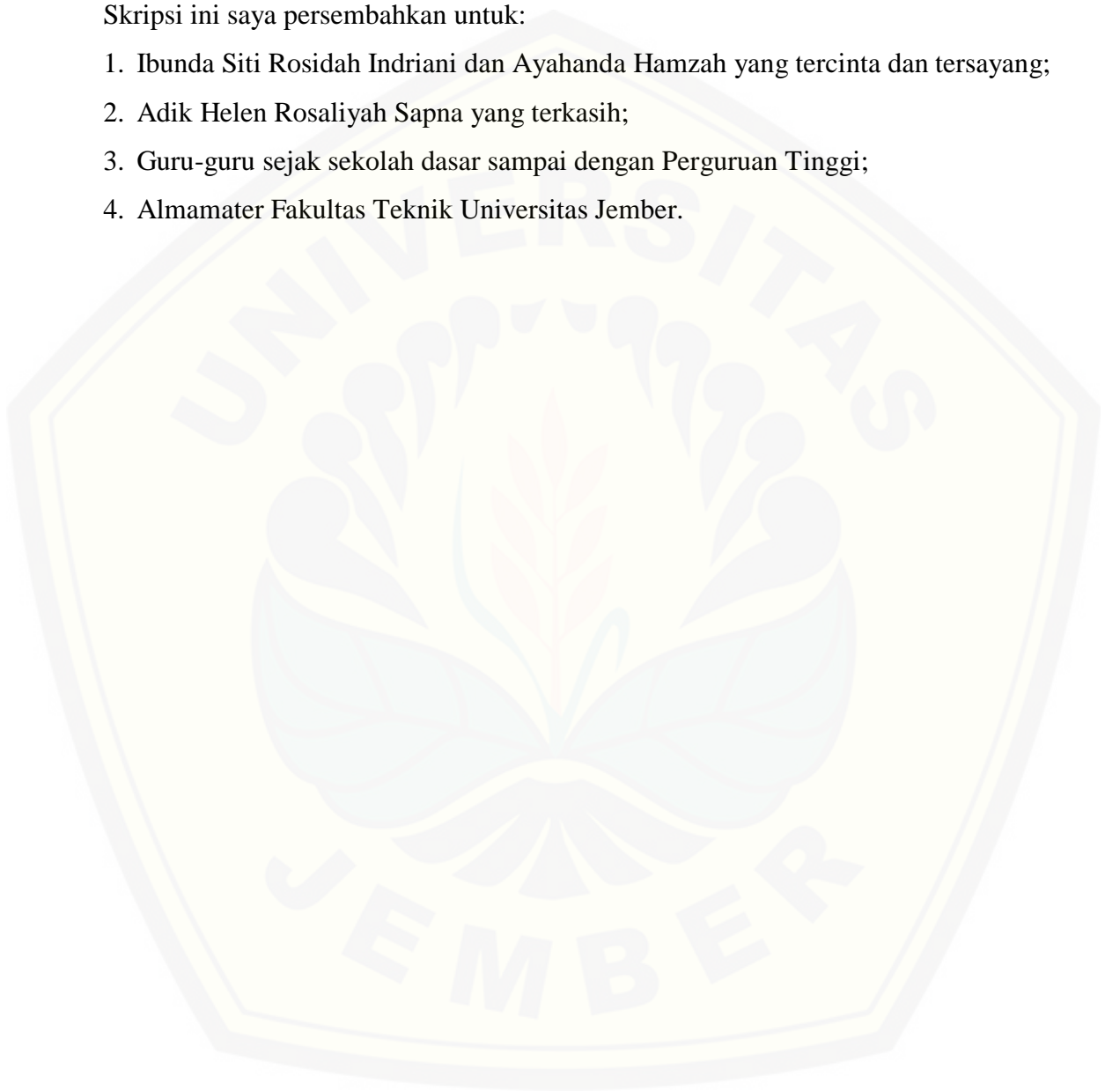
**Hendro Rosyidi Setiyawan
NIM 121910201065**

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

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Siti Rosidah Indriani dan Ayahanda Hamzah yang tercinta dan tersayang;
2. Adik Helen Rosaliyah Sapna yang terkasih;
3. Guru-guru sejak sekolah dasar sampai dengan Perguruan Tinggi;
4. Almamater Fakultas Teknik Universitas Jember.



MOTO

Buatlah perubahan walaupun hanya sedikit.^{*)}

وَإِلَىٰ رَبِّكَ فَارْغَبْ

Dan hanya kepada Tuhanmulah hendaknya kamu berharap.^{**)}

Apa yang kita kerjakan tidak ada yang sia-sia. Hasilnya mungkin tidak sekarang,
waktulah yang menentukan.^{***)}

^{*)} Anonim

^{**)} (Q.S. Alam Nasyrah (94): 8).

^{***)} Anonim.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Hendro Rosyidi Setiyawan

NIM : 121910201065

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Sistem Kontrol *Fuzzy Logic* Pada Generator DC Penguatan Terpisah 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,

Hendro Rosyidi Setiyawan
NIM 121910201065

SKRIPSI

**SISTEM KONTROL *FUZZY LOGIC* PADA GENERATOR DC
PENGUATAN TERPISAH BERBASIS
ARDUINO UNO R3**

oleh

Hendro Rosyidi Setiyawan
NIM 121910201065

Pembimbing

Dosen Pembimbing Utama : Ir. Widyono Hadi, M.T.
Dosen Pembimbing Anggota : Dr. Triwahju Hardianto, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Sistem Kontrol *Fuzzy Logic* Pada Generator DC Penguatan Terpisah Berbasis Arduino UNO R3” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 2 Juni 2016

tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Pembimbing Utama,

Pembimbing Anggota,

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

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

Penguji I,

Penguji II,

Supriyadi Prasetyono, S.T., M.T.
NIP 19700404 199601 1 001

Dr. Ir. Bambang Sujanarko, M.M.
NIP 19631201 199402 1 002

Mengesahkan
Dekan Fakultas Teknik,

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

Sistem Kontrol *Fuzzy Logic* Pada Generator DC Penguatan Terpisah Berbasis Arduino UNO R3

Hendro Rosyidi Setiyawan

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Generator DC merupakan perangkat listrik dinamis yang mengubah energi mekanik menjadi energi listrik, namun generator DC memiliki karakteristik saat diberi beban maka tegangan keluaran generator akan mengalami *drop* tegangan. Tujuan dari penelitian ini adalah menerapkan kontrol *fuzzy logic* sehingga diperoleh kestabilan tegangan keluaran generator DC. Besarnya injeksi yang diberikan pada generator DC diatur melalui *buck converter* yang dikontrol dari mikrokontroler. Tahapan pembuatan program *fuzzy logic* yang dilakukan antara lain *fuzzyfication*, *inference* dan *defuzzyfication*. Pembuatan pemodelan sistem *fuzzy* mengubah hasil dari pemodelan sistem *fuzzy* yang dirancang menjadi program agar dapat di *input* ke arduino UNO R3. Analisa perbandingan yang dilakukan yaitu analisa *drop* tegangan dan *over voltage* pada keadaan *close loop* maupun *open loop fuzzy logic*. Selisih dari *drop* tegangan dan *over voltage* yang terjadi untuk sistem *open loop* jauh lebih besar jika dibandingkan dengan *drop* tegangan dan *over voltage* dengan sistem *close loop fuzzy logic*.

Kata Kunci: Arduino UNO-R3, *Buck Converter*, *Fuzzy Logic*, Generator DC

Fuzzy Logic Control System On The Separately Excited DC Generator Based Arduino UNO R3

Hendro Rosyidi Setiyawan

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

ABSTRACT

DC generator is an dinamic electrical machine which convert mechanical energy into electrical energy, DC generator's output will coming through voltage drop if be given load. This study aimed to apply fuzzy logic control for stability of DC generator's output. The amount of injection given to a DC generator's arranged through controlled buck converter of microcontroller. Steps of programing fuzzy logic were fuzzyfication, inference and defuzzyfication. Fuzzy system would converted the result of fuzzy system plan into program that included to arduino UNO R3. Comparison analyze that has been applied in this study is voltage drop and over voltage's analyze in close loop and open loop fuzzy logic. Difference voltage drop and over voltage between open loop and close loop fuzzy logic were significant.

Keywords: *Arduino UNO-R3, Buck Converter, DC Generator, Fuzzy Logic*

RINGKASAN

Sistem Kontrol *Fuzzy Logic* Pada Generator DC Penguatan Terpisah Berbasis Arduino UNO R3; Hendro Rosyidi Setiyawan, 121910201065; 2016; 71 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Keberlangsungan kehidupan manusia saat ini tidak jauh dari kebutuhan akan energi listrik. Banyak sumber energi yang mampu menghasilkan energi listrik sebagai pemasok kebutuhan listrik sehari-hari. Semakin meningkatnya pertumbuhan ekonomi saat ini memberikan beberapa kendala pada sisi pemasok sumber energi listrik. Salah satunya yaitu beban yang ditanggung sangatlah besar dan kestabilan sistem tidaklah mampu mengatasinya.

Pemakai kontrol diharapkan dapat mengatasi kestabilan sistem pada suatu *plan* yang digunakan. Dengan memanfaatkan perkembangan mikrokontroler yang berupa arduino uno maka mempermudah proses perancangan sistem kontrol yang dibuat. Sedangkan untuk metode kontrol yang digunakan adalah metode *fuzzy logic*. *Fuzzy logic* sendiri telah banyak digunakan dan dikembangkan untuk mengontrol secara otomatis pada suatu *plan* khususnya pada pengontrolan motor listrik dan juga generator.

Pada penelitian ini tegangan keluaran dari generator DC dikondisikan agar tetap stabil pada *setpoint* yang telah ditentukan yaitu 220 Volt walau pada kondisi dibebani dengan variasi yang berbeda-beda hingga sebesar 65 Watt. Dimana untuk generator DC yang digunakan berupa generator DC penguatan terpisah dengan *inject* yang berasal dari *buck converter*. Besarnya *inject* yang diberikan pada generator DC diatur dengan *fuzzy logic* yang diproses dalam arduino sehingga menghasilkan PWM yang akan menentukan *output* dari *buck converter*.

Tegangan keluaran pada generator tetap pada keadaan stabil di *setpoint* yang ditentukan saat dilakukan pengujian pemberian dan pelepasan beban. Respon dari kontrol *fuzzy logic* lebih baik dari sisi respon saat terjadi *over voltage* dan juga *under voltage* serta untuk *rise time* juga lebih baik dari kontrol PID.



PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Kontrol *Fuzzy Logic* Pada Generator DC Penguatan Terpisah 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. RB. Moch Gozali, S.T., M.T. selaku Dosen Penguji I, Dr. Ir. Bambang Sujanarko, M.M. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang sangat membangun demi penyempurnaan skripsi ini;
3. Samsul Bachri Masmachofari, S.T., M.MT. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Mohamad Agung Prawira Negara, S.T., M.T. selaku Komisi Bimbingan S1 yang telah membantu penulisan skripsi secara administratif;
5. Andi Setiawan, S.T., M.T. yang selalu berbagi ilmu selama proses pengerjaan skripsi di Laboratorium;
6. Bunda Siti Rosidah Indriani dan ayahanda Hamzah yang telah memberikan dukungan moril dan materiil serta kasih sayang yang tak terhingga;
7. Adik tersayang Helen Rosaliyah Sapna yang telah memberikan motivasi dan semangat;
8. Saudaraku “ 65 Watt Squad ”, Abdul Fajar dan Aditiya Chandra Purnama yang mulai dari awal hingga akhir pengerjaan skripsi ini selalu bersama-sama baik

suka maupun duka. Terimakasih kalian banyak memberikan pelajaran tentang kehidupan;

9. Keluarga besar SATE UJ 2012 “Saduluran Arek Teknik Elektro Universitas Jember 2012” yang telah banyak mengukirkan cerita baru selama 4 tahun menimbah ilmu di kota rantau. Tetap jaga persaudaraan kita sampai kapanpun. “Dulur saklawase ora dulur sak luluse”;
10. Keluarga Besar Laboratorium Sistem Tenaga Fakultas Teknik Universitas Jember yang selalu memberikan semangat serta rasa kebersamaannya;
11. Tim hore Ayu Prativia Yonenda, S.Kg yang selalu memberi semangat dan keceriaan serta selalu menemani saat suka dan duka. Terimakasih atas bantuannya selama ini dalam proses editing jurnal;
12. Rekan yang selalu ceria Rina Oliph Anggraeni yang terkadang merepotkan dan juga direpotkan. Terimakasih selalu menjadi devisa konsumsi disetiap seminar selama proses penyusunan skripsi;
13. Semua pihak yang tidak dapat disebutkan satu persatu.

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
<i>ABSTRACT</i>	ix
RINGKASAN	x
PRAKATA	xii
DAFTAR ISI.....	xiv
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR.....	xviii
DAFTAR LAMPIRAN.....	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penelitian	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Generator DC Penguatan Terpisah	4
2.2 Sensor Tegangan	8
2.3 Arduino UNO R3	9

2.4 Buck Converter	11
2.5 Fuzzy Logic	15
2.5.1 Pengertian <i>Fuzzy Logic</i>	15
2.5.2 Himpunan Fuzzy	19
2.5.3 Fungsi Keanggotaan	21
2.5.4 Tahap Pemodelan dalam <i>Fuzzy Logic</i>	23
2.5.4.1 <i>Fuzzyfication</i>	23
2.5.4.2 <i>Rule Evaluation</i>	24
2.5.4.3 <i>Defuzzyfication</i>	25
BAB 3. METODOLOGI PENELITIAN	26
3.1 Alat dan Bahan	26
3.2 Tahapan Penelitian	27
3.3 Blok Diagram Sistem	29
3.4 Perancangan Komponen Pengendali (<i>Hardware</i>)	30
3.4.1 Perancangan Sensor Tegangan	30
3.4.2 Perancangan <i>Buck Converter</i>	32
3.4.3 Generator DC Penguatan Terpisah	33
2.4.4 Konfigurasi PIN Arduino UNO R3	34
3.5 Perancangan Sistem Kendali Fuzzy	34
3.5.1 Tahap Fuzzyfikasi	35
3.5.2 Tahap Inferensi	37
3.5.3 Tahap Defuzzyfikasi	38
3.6 Aplikasi Monitoring Generator	40
BAB 4. HASIL DAN PEMBAHASAN	42
4.1 Pengujian Rangkaian Sensor	43
4.2 Pengujian <i>Buck Converter</i>	45
4.3 Sistem Kendali <i>Fuzzy Logic</i>	49
4.3.1 Proses Fuzifikasi	50

4.3.2 Proses Inferensi	52
4.3.3 Proses Defuzifikasi	53
4.4 Pengujian Generator DC Penguatan Terpisah Tanpa Kontrol	55
4.4.1 Pengujian Pemberian Beban Tanpa Kontrol	55
4.4.2 Pengujian Pelepasan Beban Tanpa Kontrol	56
4.5 Pengujian Generator DC Penguatan Terpisah Menggunakan Kontrol <i>Fuzzy Logic</i>	58
4.5.1 Pengujian Pemberian Beban Menggunakan Kontrol <i>Fuzzy Logic</i>	58
4.5.2 Pengujian Pelepasan Beban Menggunakan Kontrol <i>Fuzzy Logic</i>	63
4.6 Perbandingan Sistem <i>Open Loop</i> dan <i>Close Loop Fuzzy Logic</i>	67
BAB 5. PENUTUP	70
5.1 Kesimpulan	70
5.2 Saran	71
DAFTAR PUSTAKA	72
LAMPIRAN	74

DAFTAR TABEL

	Halaman
2.1 Spesifikasi Arduino UNO-R3	11
3.1 Daftar Peralatan yang Digunakan	26
4.1 Pengujian Sensor Tegangan	43
4.2 Pengujian <i>Buck Converter</i>	46
4.3 Daya <i>Input</i> , Daya <i>Output</i> dan Efisiensi	47
4.4 Pengujian Pemberian Beban pada Generator Tanpa Kontrol	55
4.5 Pengujian Pelepasan Beban pada Generator Tanpa Kontrol	56
4.6 Pengujian Pemberian Beban dengan Kontrol Fuzzy.....	58
4.7 Pengujian Pelepasan Beban dengan Kontrol Fuzzy.....	63
4.8 Perbandingan <i>Drop</i> Tegangan Sistem <i>Open Loop</i> dan <i>Close Loop</i>	68
4.9 Perbandingan <i>Over Voltage</i> Sistem <i>Open Loop</i> dan <i>Close Loop</i>	69

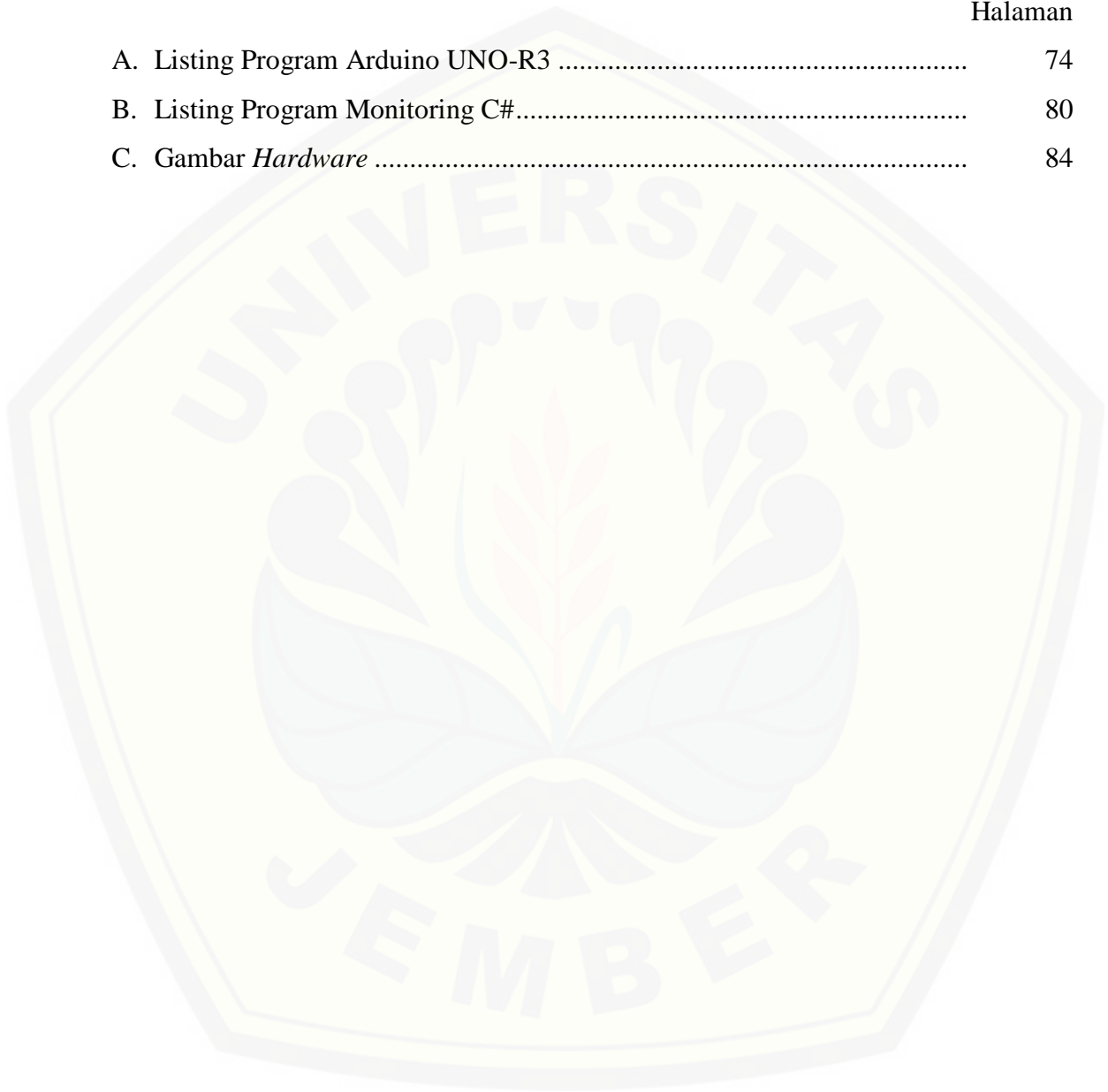
DAFTAR GAMBAR

	Halaman
2.1 Generator DC Penguatan Terpisah	5
2.2 Karakteristik Generator DC Berbeban.....	6
2.3 Karakteristik $\phi - I_f$	7
2.4 Karakteristik Generator DC Penguatan Terpisah	7
2.5 Sensor Tegangan DCT-Elektronik	9
2.6 Modul Kontroler Arduino UNO-R3	10
2.7 Skematik <i>Buck Converter</i>	11
2.8 <i>Output Voltage and Current Waveform</i>	12
2.9 <i>Voltage Step Down Copper with Feedback Controller</i>	14
2.10 Konsep Dasar Logika Fuzzy.....	17
2.11 Perbedaan <i>Booleana Logic</i> dengan <i>Fuzzy Logic</i>	19
2.12 Blok Diagram Sistem Fuzzy 1	23
2.13 Blok Diagram Sistem Fuzzy 2	23
2.14 Blok <i>Fuzzyfication</i>	24
2.15 Blok <i>Defuzzyfication</i>	25
3.1 <i>Flowchart</i> Penelitian.....	27
3.2 Blok Diagram Sistem.....	29
3.3 Skematik Sensor Tegangan	31
3.4 Rangkaian Pengujian Sensor Tegangan.....	31
3.5 Skematik Rangkaian <i>Buck Converter</i>	32
3.6 Generator DC.....	33
3.7 Blok Sistem Kendali Fuzzy	35
3.8 <i>Membership Function Error</i>	35
3.9 <i>Membership Function Delta Error</i>	37
3.10 <i>Rule Base</i>	37

3.11	<i>Membership Function Control</i>	38
3.12	Perhitungan Titik Berat Luasan	39
3.13	<i>Flowchart</i> Program.....	40
3.14	Aplikasi Monitoring Generator	41
4.1	Pengujian Sensor Tegangan dengan Multimeter	44
4.2	Karakteristik <i>Duty Cycle</i> terhadap <i>Output</i> Tegangan.....	47
4.3	Respon Kontrol pada <i>Buck Converter</i>	48
4.4	<i>Membership Function Error</i>	51
4.5	<i>Membership Function Delta Error</i>	52
4.6	<i>Membership Function Control</i>	52
4.7	Grafik Perbandingan Pemberian Beban Sistem <i>Open Loop</i>	55
4.8	Grafik Perbandingan Pelepasan Beban Sistem <i>Open Loop</i>	57
4.9	Perbandingan Respon Kontrol Tiap Pengujian	59
4.10	Pengujian Pemberian Beban 15 Watt	59
4.11	Pengujian Pemberian Beban 25 Watt	60
4.12	Pengujian Pemberian Beban 40 Watt	61
4.13	Pengujian Pemberian Beban 50 Watt	61
4.14	Pengujian Pemberian Beban 65 Watt	62
4.15	Perbandingan Respon Kontrol Tiap Pengujian	64
4.16	Pengujian Pelepasan Beban 15 Watt	64
4.17	Pengujian Pelepasan Beban 25 Watt	65
4.18	Pengujian Pelepasan Beban 40 Watt	65
4.19	Pengujian Pelepasan Beban 50 Watt	66
4.20	Pengujian Pelepasan Beban 65 Watt	66

DAFTAR LAMPIRAN

	Halaman
A. Listing Program Arduino UNO-R3	74
B. Listing Program Monitoring C#.....	80
C. Gambar <i>Hardware</i>	84



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Energi listrik dalam kehidupan sehari-hari telah menjadi kebutuhan yang sangat penting untuk menunjang kelangsungan aktifitas manusia. Penggunaannya juga telah banyak dilakukan pengaturan yang semakin komplek agar kehandalan sistem semakin baik. Untuk menjadikan sumber energi listrik yang handal, banyak dikembangkan sistem kontrol untuk mengatur proses dan juga hasil dari energi listrik. Efisiensi pengontrolan perangkat listrik juga berpengaruh dalam kehandalan *output* yang dihasilkan.

Generator DC merupakan sebuah perangkat listrik dinamis yang mengubah energi mekanik menjadi energi listrik. Generator DC dibedakan berdasarkan dari rangkaian belitan magnet atau penguat eksitasinya terhadap jangkar, salah satunya yaitu generator penguat terpisah. Pada generator DC penguat terpisah, *supply* yang diberikan dipasok dari daya konstan tegangan *supply* terpisah (Chapman, 2005). Besar tegangan listrik yang dihasilkan oleh generator sebanding dengan besar medan magnet di dalamnya, sedangkan besar medan magnet ini sebanding dengan besar arus eksitasi yang dibangkitkan.

Karakteristik generator DC ketika diberi beban maka *output* pada generator akan mengalami *drop* tegangan. Prinsip ini yang menjadi dasar untuk mengontrol arus eksitasi sehingga dapat mengatur tegangan keluaran. Sistem kontrol yang dilakukan untuk mengatur besarnya arus eksitasi ini dengan menggunakan metode *fuzzy logic*. Kontrol ini nantinya akan mengatur *output* yang dihasilkan *buck converter* yang digunakan untuk menginjek arus eksitasi pada generator melalui sinyal PWM. *Fuzzy logic* yang fleksibel dimungkinkan untuk mengatur tegangan keluaran melalui arus eksitasi sehingga saat tegangan keluaran turun atau terjadi *drop* tegangan, maka besarnya arus eksitasi yang dikontrol akan dinaikkan secara

otomatis oleh sistem, begitu juga sebaliknya. Dengan menggunakan metode *fuzzy logic* diharapkan dapat mengatasi *drop* tegangan dan juga *overshoot* pada tegangan keluaran generator saat diberi variasi beban.

1.2 Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah diantaranya :

1. Bagaimana perancangan sistem kontrol *fuzzy logic* untuk kontrol tegangan keluaran pada generator DC penguatan terpisah.
2. Bagaimana kestabilan tegangan keluaran generator DC setelah menggunakan kontrol *fuzzy logic* terhadap variasi beban.

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, maka pembahasan pada skripsi dibatasi pada hal-hal sebagai berikut :

1. Kontrol hanya pada tegangan keluaran generator DC dengan mengatur arus eksitasi medan di stator generator DC penguatan terpisah. Dimana generator DC dirangkai secara *shunt*.
2. Pengaturan arus eksitasi medan di stator generator DC penguatan terpisah dengan menggunakan *buck converter*.
3. Analisa hanya dititik beratkan pada tegangan keluaran pada generator DC bukan pada *output* yang dihasilkan dari *buck converter*.
4. *Prime mover* yang digunakan yaitu mesin DC.
5. Pengujian pada tegangan 220 Volt dengan variasi beban untuk pengujian yaitu mulai dari 15 Watt, 25 Watt, 40 Watt, 50 Watt dan 65 Watt dengan cara pemberian beban dan pelepasan beban.

1.4 Tujuan Penelitian

Penelitian ini dilakukan karena memiliki beberapa tujuan, diantaranya :

1. Merancang sistem kontrol *fuzzy logic* untuk kontrol tegangan keluaran pada generator DC penguatan terpisah berbasis arduino UNO-R3.
2. Menerapkan kontrol *fuzzy logic* sehingga diperoleh kestabilan tegangan keluaran generator DC terhadap variasi beban yang diberikan.

1.5 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat diantaranya :

1. Kestabilan tegangan keluaran pada generator DC penguatan terpisah dapat diatur secara otomatis dengan sistem *close loop*.
2. Penggunaan beban dapat lebih bervariasi dengan tegangan keluaran tetap stabil.

1.6 Sistematika Penelitian

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut:

- a. BAB 1. PENDAHULUAN
Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.
- b. BAB 2. TINJAUAN PUSTAKA
Berisi penjelasan tentang teori yang berhubungan dengan penelitian.
- c. BAB 3. METODOLOGI PENELITIAN
Menjelaskan tentang metode kajian yang digunakan untuk menyelesaikan skripsi.
- d. BAB 4. HASIL DAN PEMBAHASAN
Berisi hasil penelitian dan analisa hasil penelitian.
- e. BAB 5 PENUTUP
Berisi tentang kesimpulan dan saran dari penulis.

BAB 2. TINJAUAN PUSTAKA

2.1 Generator DC Penguatan Terpisah

Generator DC terdiri dua bagian, yaitu stator merupakan bagian dari mesin DC yang diam, dan bagian rotor, yaitu bagian mesin DC yang berputar. Bagian stator terdiri dari: rangka motor, belitan stator, sikat arang, bearing dan terminal box. Sedangkan bagian rotor terdiri dari: komutator, belitan rotor, kipas rotor dan poros rotor. Pembangkitan tegangan induksi oleh sebuah generator diperoleh melalui dua cara ada yang dengan menggunakan cincin-seret dimana menghasilkan tegangan induksi bolak-balik dan juga dengan menggunakan komutator yang menghasilkan tegangan DC.

Berdasarkan cara dimana fluks medan yang dihasilkan, generator DC diklarifikasikan menjadi lima jenis antara lain (Chapman, 2005):

1. *Separately excited generator* / Generator penguatan terpisah
2. *Shunt generator* / Generator Shunt
3. *Series generator* / Generator Series
4. *Cumulatively compounded generator*
5. *Differentially compounded generator*

Pada generator arus searah / generator DC berlaku hubungan-hubungan sebagai berikut

$$E_a = \frac{\phi z n P}{60 a} \text{ Volt} \quad (2.1)$$

Dimana:

E_a = GGL yang dibangkitkan pada jangkar generator

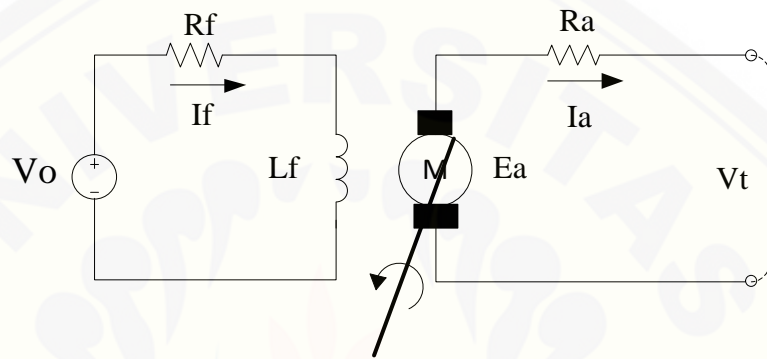
ϕ = Fluks per kutub

z = jumlah penghantar total

n = kecepatan putar

a = jumlah hubungan paralel

Generator DC penguatan terpisah adalah generator yang arus medan dipasok oleh sumber tegangan DC eksternal yang terpisah. Untuk membangkitkan fluks pada medan diperlukan arus searah yang dipasangkan pada kumparan medan yang mempunyai tahanan R_f sehingga menghasilkan fluks pada kedua kutub (Fajar, Hadi, & Kaloko, 2016).



Gambar 2.1 Generator DC Penguatan Terpisah

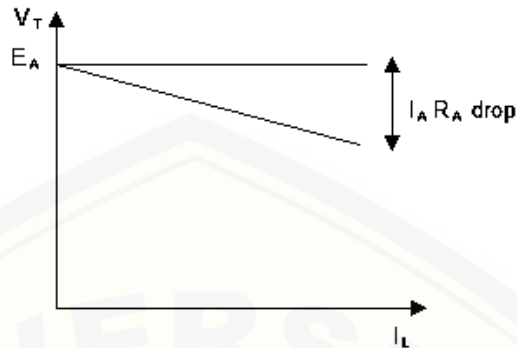
Persamaan :

$$I_f = \frac{V_o}{R_f} \quad (2.2)$$

$$V_T = E_a - I_a R_a \quad (2.3)$$

Keterangan :

- V_T = Tegangan keluaran (V)
- I_f = Arus kumparan medan (A)
- V_o = Tegangan kumparan medan (V)
- R_f = Resistansi kumparan medan (Ω)
- E_a = Tegangan induksi jangkar (V)
- I_a = Arus jangkar (A)
- R_a = Resistansi jangkar (Ω)
- I_L = Arus Beban (A)



Gambar 2.2 Karakteristik Generator DC Berbeban

Dari persamaan (2.3) dapat diketahui bila beban semakin tinggi maka untuk besarnya tegangan keluaran semakin turun, hal ini diakibatkan adanya *drop* tegangan dari $I_a R_a$. Untuk meningkatkan tegangan keluaran yang turun tersebut dapat diatasi dengan meningkatkan tegangan induksi (E_a) dengan persamaan sebagai berikut.

$$E_a = K\phi\omega r \quad (2.4)$$

Keterangan:

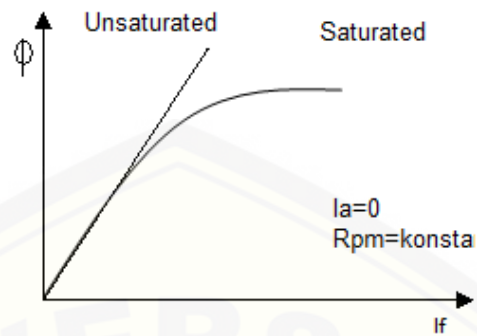
E_a = Tegangan Induksi (V)

K = Konstanta

ϕ = Fluks magnet (Wb)

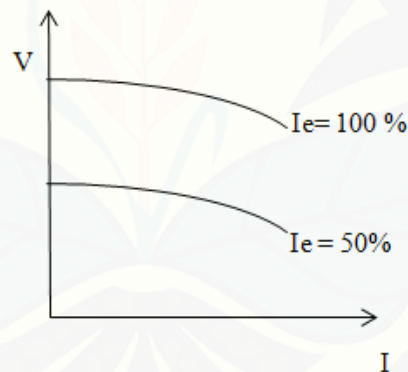
ωr = Kecepatan putar (RPM)

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 2.3 Karakteristik $\phi - I_f$

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.



Gambar 2.4 Karakteristik Generator DC Penguatan Terpisah

Dari gambar grafik karakteristik generator DC penguatan terpisah diatas dapat diketahui bahwa dimana I_e merupakan arus eksitasi dan I adalah arus beban. Saat arus eksitasi penuh ($I_e = 100\%$) dan pada saat arus eksitasi setengah penuh ($I_e = 50\%$) tegangan keluaran generator akan sedikit turun jika arus beban semakin besar. Dengan semakin turunnya tegangan tersebut akibat resistansi jangkar dan reaksi

jangkar, yang menyebabkan turunnya *supply* arus eksitasi ke medan magnet sehingga tegangan induksi menjadi kecil.

Tegangan keluaran pada generator DC penguatan terpisah dapat dilakukan dengan merubah tegangan induksi pada mesin DC. Menurut hukum Kirchoff II tegangan pada rangkaian tertutup akan bernilai sama dengan nol menghasilkan persamaan berikut $V_T = E_a - I_a R_a$ (Hameyer, 2001). Jadi, jika E_a meningkat maka tegangan keluaran akan meningkat dan jika E_a menurun maka tegangan keluaran akan menurun. Pada persamaan E_a adalah $E_a = K\phi\omega_r$ sehingga dapat dilakukan dua kemungkinan untuk mengontrol tegangan keluaran yang pertama adalah ω_r (kecepatan putar) dan ϕ (fluks) :

a. Merubah kecepatan putar (ω_r).

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

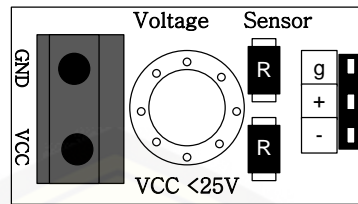
b. Merubah fluks (ϕ)

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

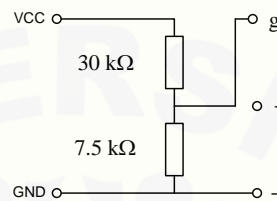
2.2 Sensor Tegangan

Pada penelitian yang dilakukan ini, untuk menstabilkan tegangan keluaran dengan cara mengontrol arus eksitasi digunakan beberapa komponen pengendali dimana memiliki fungsi masing-masing, seperti sebagai *input* data , pengolah data dan juga pengontrol *output* sesuai dengan *input* yang diterima.

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



(a)



(b)

Gambar 2.5 Sensor Tegangan DCT-Elektronik

Gambar 2.4(a) adalah modul sensor pembagi tegangan, sedangkan Gambar 2.4(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 (Autodesk.Inc, 2015).

2.3 Arduino UNO R3

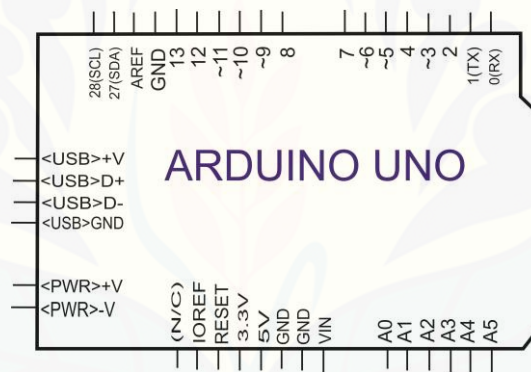
Arduino Uno adalah modul mikrokontroler berbasis ATmega 328. Modul ini memiliki empat belas pin *input-output* (14 I/O) di mana enam diantaranya dapat digunakan sebagai *output pulse width modulation* (PWM), enam *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, tombol *reset* dan koneksi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

“Uno” berarti satu dalam bahasa Itali dan hal ini untuk menandai versi pertama (Arduino 1.0) yang akan menjadi versi acuan bagi Arduino berikutnya. Uno adalah seri terakhir dari seri papan atau modul USB Arduino. Uno tidak lagi menggunakan *chip* FTDI USB-to-serial tetapi menggunakan Atmega 8U2 sebagai konverter USB ke serial. Gambar 2.6 (a) adalah modul Arduino UNO-R3 sedangkan Gambar 2.6 (b)

menunjukkan skema dari modul Arduino UNO-R3 yang merupakan penyederhanaan dari gambar modul aslinya (Arduino LLC, 2015).



(a)



(b)

Gambar 2.6 Modul Kontroler Arduino UNO-R3
(Sumber : www.arduino.com)

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 Ω yang secara *default* tidak tersambung (*disconnected*).

Berikut ini adalah tabel spesifikasi dari Arduino Uno R3 dapat kita lihat pada Tabel 2.1 dibawah ini.

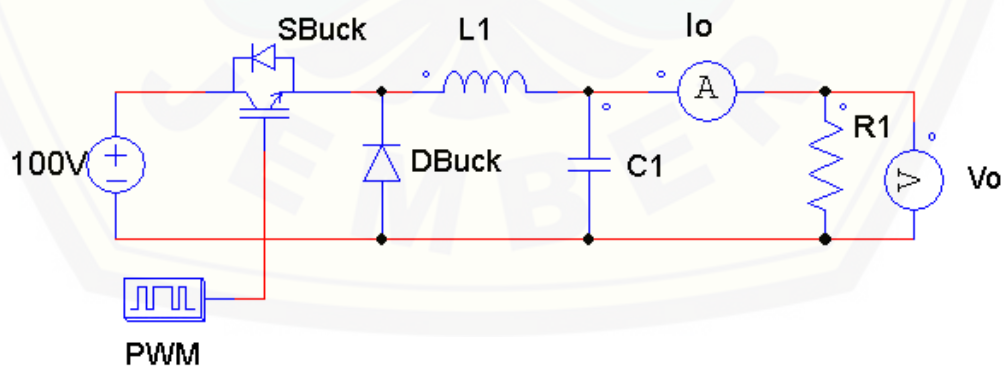
Tabel 2.1 Spesifikasi Arduino UNO-R3

Mikrokontroler	Atmega 328
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 Clock	16 MHz

Diterjemahkan dari: (Arduino UNO 2015)

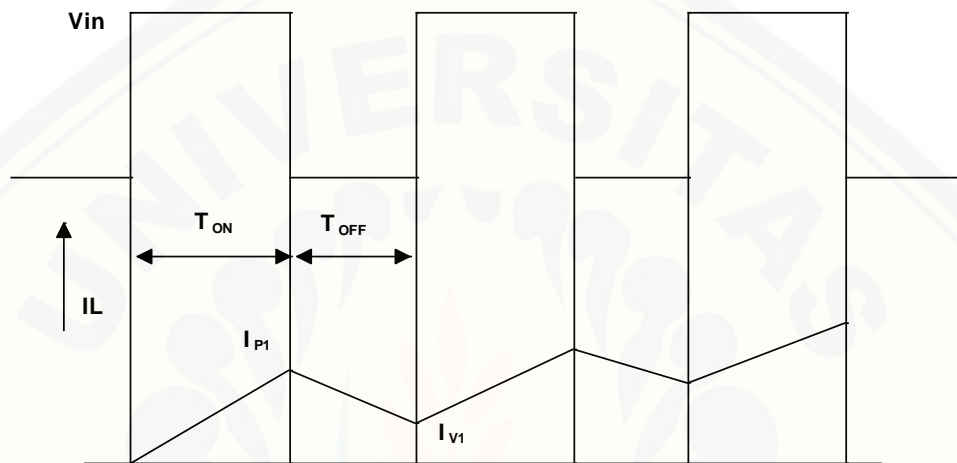
2.4 Buck Converter

Buck converter adalah salah satu DC-DC *converter* yang dapat menurunkan tegangan DC berdasarkan *Duty-Cycle* MOSFET pada rangkaian *buck converter*. Tegangan keluaran pada *buck converter* akan turun bila *duty-cycle* pada *Sbuck* turun.



Gambar 2.7 Skematik *Buck Converter*

Untuk prinsip kerja dari *buck converter* yaitu dimana MOSFET yang mengatur proses *switching* yang tinggi pada saat pemberian sinyal PWM yang dikontrol. Selain MOSFET yang digunakan untuk proses *switching* bisa juga dengan menggunakan IGBT. Pada gambar dibawah ini menunjukkan tegangan keluaran dan juga bentuk gelombang yang dihasilkan.



Gambar 2.8 Output Voltage and Current Waveform

Ketika MOSFET pada keadaan *ON* maka tegangan V_{in} diterapkan secara *reverse* didioda. Karena itu dioda harus tetap pada kondisi *OFF* selama MOSFET tetap pada keadaan *ON*. Keadaan *ON* dari MOSFET selalu menunjukkan keadaan *OFF* dari dioda. Ketika MOSFET pada keadaan *ON*, maka arus I_L meningkat. Kenaikan arus I_L terjadi secara eksponensial dikarenakan induktansi (L). T_{ON} untuk selang waktu MOSFET pada kondisi *ON* dan T_{OFF} untuk keadaan MOSFET pada kondisi *OFF*. Pada saat MOSFET pada kondisi *OFF*, I_L memiliki nilai yang terbatas (I_{P1}) yang merupakan nilai puncak arus keluaran selama selang waktu siklus pertama. Nilai puncak terjadi pada saat MOSFET berubah kondisi *OFF* dimana terdapat induktansi yang bertujuan untuk mencegah penurunan atau *drop* I_L ke nol secara tiba-tiba.

Untuk tegangan keluaran yang dihasilkan dari *buck converter* akan memiliki hubungan dengan tegangan *input* dan juga *duty cycle* dimana D merupakan *duty cycle* hal ini dapat ditunjukkan dengan persamaan rumus dibawah ini:

$$V_{out} = V_{in} \cdot D \quad (2.5)$$

$$D = \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad (2.6)$$

Pada masa sekarang ini *buck converter* banyak yang menggunakan mode *switching* dengan pengaturan menggunakan *pulse width modulation* (PWM) yang memiliki efisiensi yang lebih tinggi dibanding sistem catu daya linier. Secara umum ada dua permasalahan yang sering terjadi pada *buck converter*. Pertama, sulitnya pengendalian arus pada induktor yang memiliki *ripple* yang cukup besar. Kedua adalah transient tegangan keluaran yang muncul pada saat *start up* (Himawan, Setyawati, & Suyono, 2016).

Sinyal *Power Width Modulation* (PWM) merupakan salah satu sinyal kontrol yang paling sering diterapkan pada proses *switch* dalam *switching DC converter*. Sinyal PWM ini sering digunakan pada proses *switching* dikarenakan sinyal PWM merupakan sinyal dengan frekuensi tetap. Frekuensi sangat berhubungan dengan *output ripple*. Untuk mendapatkan tegangan *output ripple* yang jauh lebih kecil dari nilai rata-rata, maka sebagian besar arus *ripple* induktor harus melalui kapasitor. Tegangan *output ripple* dapat ditentukan oleh persamaan berikut:

$$\frac{\Delta V_{out}}{V_{out}} = \frac{(1-D)T_{SW}^2}{8LC} \quad (2.7)$$

Dimana

T_{SW} merupakan periode *switching*

$$f_{SW} = 1 / T_{SW}$$

Tegangan *output ripple* dari DC-DC *converter* dapat diminimalkan dengan penentuan pada besarnya nilai induktor dan juga pada kapasitor yang diberikan, hal ini dapat ditentukan seperti pada perhitungan dibawah ini

$$L = \frac{D(V_{IN} - V_{OUT})}{I_{ripple} \times F_s} \quad (2.8)$$

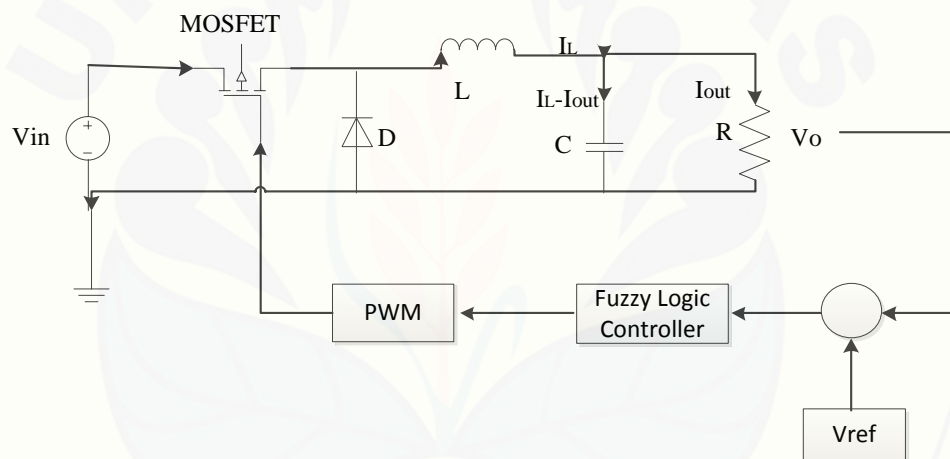
$$C = \frac{I_{ripple}}{8 \times F_s \times \Delta V} \quad (2.9)$$

Dimana V_{IN} = Input tegangan pada *buck converter*

V_{OUT} = Output tegangan

F_s = Frekuensi *switching*

I_{ripple} merupakan arus *ripple* induktor, biasanya 30% dari arus keluaran maksimum. ΔV adalah tegangan *output ripple*, biasanya 1% dari tegangan *output*. Kemudian untuk ESR (*Effective Series Resistance*) merupakan resistansi seri efektif kapasitor dimana pada umumnya 0.03 Ω . Sebuah kapasitor dengan ESR rendah akan membantu untuk meminimalkan *ripple* pada *output* tegangan.



Gambar 2.9 Voltage step down chopper with feed back controller

Sebuah teknik kontrol yang sesuai untuk konverter dc-dc harus mengatasi non-linear intrinsik dan lebar tegangan input serta variasi beban, memastikan stabilitas dalam kondisi operasi apapun sambil memberikan respon yang cepat. Ketika konverter *switching* merupakan kasus sistem struktur variabel, teknik kontrol *sliding mode* bisa menjadi pilihan yang mungkin untuk mengontrol jenis sirkuit (Guldemir, 2011).

Dalam DC-DC *converter*, kerugian dapat diklasifikasikan menjadi dua jenis yaitu kerugian konduksi dan juga pada frekuensi kerugian *switching*. Dari dua jenis kerugian tersebut masing-masing dibagi lagi menjadi:

A. Rugi Konduksi

Selama mode konduksi kontinyu (berarti induktor saat ini tidak akan mencapai ke nol) di mana arus beban relatif besar, kontribusi utama dari rugi daya adalah hilangnya konduksi pada resistensi dari sisi tinggi (R_{on_PFET}) dan sisi rendah (R_{on_NFET}) switch dan resistansi seri dari induktor dan kapasitor (R_L, R_{ESR})

- a) Rugi konduksi saat *switching*
- b) Rugi konduksi pada induktor dan kapasitor

B. Rugi *Switching*

Kerugian-kerugian pada *switching* tergantung pada frekuensi, hal ini dapat dibedakan menjadi dua jenis yaitu:

- a) *Hard Loss (overlap loss)*
- b) *Soft Loss (gate drive loss)*

2.5 Fuzzy Logic

2.5.1 Pengertian *Fuzzy Logic*

Dalam bahasa Inggris, *fuzzy* mempunyai arti kabur atau tidak jelas. Jadi, *fuzzy logic* adalah logika yang kabur, atau mengandung unsur ketidakpastian. Pada logika biasa, yaitu logika tegas, kita hanya mengenal dua nilai, salah atau benar, 0 atau 1. Sedangkan *fuzzy logic* mengenal nilai antara benar dan salah. *Fuzzy logic* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk *linguistik*, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". Kebenaran dalam *fuzzy logic* dapat dinyatakan dalam derajat kebenaran yang nilainya antara 0 sampai 1.

Titik awal dari konsep modern mengenai ketidakpastian adalah paper yang dibuat oleh Lofti A Zadeh (1965), dimana Zadeh memperkenalkan teori yang

memiliki obyek-obyek dari *himpunan fuzzy* yang memiliki batasan yang tidak presisi dan keanggotaan dalam himpunan *fuzzy*, dan bukan dalam bentuk logika benar (*true*) atau salah (*false*), tapi dinyatakan dalam derajat (*degree*). Konsep seperti ini disebut dengan *Fuzziness* dan teorinya dinamakan *Fuzzy Set Theory*. *Fuzziness* dapat didefinisikan sebagai logika kabur berkenaan dengan semantik dari suatu kejadian, fenomena atau pernyataan itu sendiri. Seringkali ditemui dalam pernyataan yang dibuat oleh seseorang, evaluasi dan suatu pengambilan keputusan. Sebagai contoh:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayanan yang diberikan.
3. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.

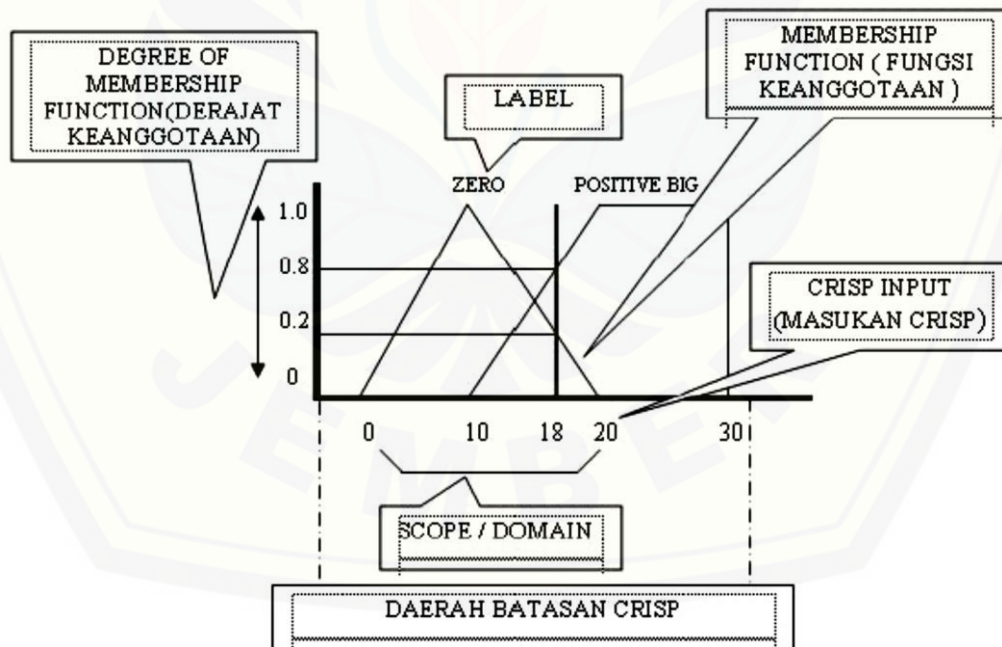
Fuzzy system (sistem kabur) didasari atas konsep himpunan kabur yang memetakan domain input kedalam domain output. Perbedaan mendasar himpunan tegas dengan himpunan kabur adalah nilai keluarannya. Himpunan tegas hanya memiliki dua nilai output yaitu nol atau satu, sedangkan himpunan kabur memiliki banyak nilai keluaran yang dikenal dengan nilai derajat keanggotaannya.

Kontroler logika fuzzy dikategorikan dalam kontrol cerdas (*intelligent control*). Unit logika fuzzy memiliki kemampuan menyelesaikan masalah perilaku sistem yang kompleks, yang tidak dimiliki oleh kontroler konvensional. Secara umum kontroler *fuzzy logika* memiliki kemampuan sebagai berikut:

- 1) Beroperasi tanpa campur tangan manusia secara langsung, tetapi memiliki efektifitas yang sama dengan kontroler manusia.
- 2) Mampu menangani sistem-sistem yang kompleks, non-linier dan tidak stasioner.
- 3) Memenuhi spesifikasi operasional dan kriteria kinerja. Strukturnya sederhana, kokoh dan beroperasi *real time*.

Himpunan fuzzy mempunyai peranan yang penting dalam perkembangan matematika khususnya dalam matematika himpunan. Matematikawan German George Cantor (1845-1918) adalah orang yang pertama kali secara formal mempelajari konsep tentang himpunan. Teori himpunan selalu dipelajari dan di terapkan sepanjang masa, bahkan sampai saat ini matematikawan selalu mengembangkan tentang bahasa matematika (teori himpunan). Banyak penelitian-penelitian yang menggunakan teori himpunan fuzzy dan saat ini banyak literatur-literatur tentang himpunan fuzzy, misalnya yang berkaitan dengan teknik kontrol *fuzzy logic* dan relasi fuzzy.

Pada dasarnya, teori himpunan fuzzy merupakan perluasan dari teori himpunan klasik. Pada teori himpunan klasik (*crisp*), keberadaan suatu elemen pada suatu himpunan A, hanya akan memiliki 2 kemungkinan keanggotaan, yaitu menjadi anggota A atau tidak menjadi anggota A.



Gambar 2.10 Konsep Dasar Logika Fuzzy

Dalam merancang pengendali berdasarkan logika fuzzy (*Fuzzy Logic Controller*), faktor mendasar yang harus diperhatikan adalah penskalaan dari *input/output* aturan dasar kendali fuzzy dan tipe fungsi keanggotaan fuzzy yang digunakan. Suatu tipe fungsi keanggotaan dapat diterjemahkan/dirubah menjadi fungsi keanggotaan dengan tipe yang berlainan.

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. *Fuzzy logic* dapat dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika modern dan metodis baru ditemukan pada tahun 1965, padahal sebenarnya konsep tentang *fuzzy logic* itu sendiri sudah ada sejak lama. Beberapa alasan digunakannya *fuzzy logic* (Kusumadewi & Purnomo, 2010) yaitu:

- 1) Konsep *fuzzy logic* mudah dimengerti
- 2) *Fuzzy logic* sangat fleksibel
- 3) *Fuzzy logic* memiliki toleransi terhadap data yang kurang tepat
- 4) *Fuzzy logic* mampu memodelkan fungsi nonlinier yang kompleks
- 5) *Fuzzy logic* didasari pada ahasa alami

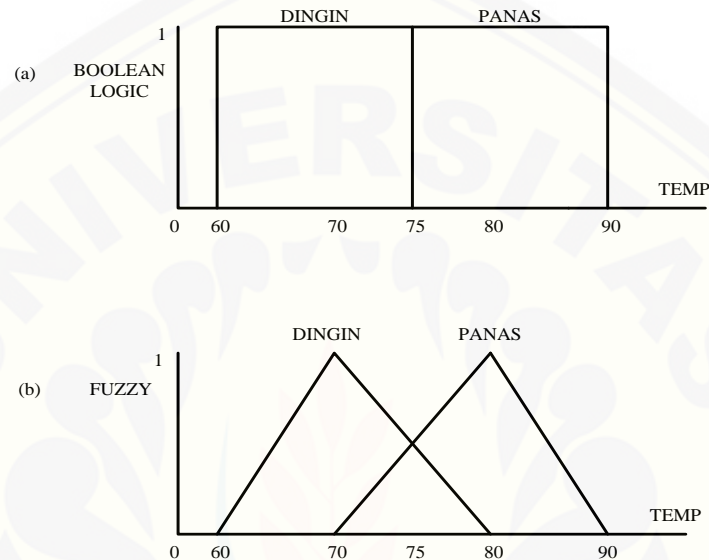
Fuzzy logic saat ini banyak diterapkan dalam berbagai bidang diantaranya:

- a) *Fuzzy rule Based Systems*
- b) *Fuzzy Nonlinear Simulations*
- c) *Fuzzy Decision Making*
- d) *Fuzzy Classification*
- e) *Fuzzy Pattern ecognition*
- f) *Fuzzy Control Systems*

Secara umum, sistem fuzzy sangat cocok untuk penalaran pendekatan terutama untuk sistem yang menangani masalah-masalah yang sulit didefinisikan dengan menggunakan model matematis Misalkan, nilai masukan dan parameter sebuah

sistem bersifat kurang akurat atau kurang jelas, sehingga sulit mendefinisikan model matematikanya.

Perbedaan *Fuzzy Logic* dengan *Boolean Logic* terlihat pada gambar dibawah ini :



Gambar 2.11 Perbedaan *Boolean Logic* dengan *Fuzzy Logic*.

2.5.2 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan (Kusumadewi & Purnomo, 2010) yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan fuzzy dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan fuzzy memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas

mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan fuzzy USIA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan fuzzy memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy yaitu:

variable fuzzy

- a. Variable fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperature, permintaan, dsb.
- b. Himpunan Fuzzy
Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.
- c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh:

- a. Semesta pembicaraan untuk variable mahasiswa: $[0 \ 50]$
- b. Semesta pembicaraan untuk variable dosen: $[0 \ 50]$

d. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan.

Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan fuzzy :

- MUDA = [0 45]
- PABOBAYA = [35 55]
- TUA = [45 +8]
- DINGIN = [0 20]
- SEJUK = [15 25]
- NORMAL = [20 30]
- HANGAT = [25 35]
- PANAS = [30 40]

2.5.3 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi fuzzy dalam U , maka A dapat dinyatakan sebagai pasangan terurut Ada beberapa fungsi yang bisa digunakan.

- a. Representasi Linier
- b. Representasi Kurva Segitiga
- c. Representasi Kurva Trapesium
- d. Representasi Kurva Bentuk Bahu

- e. Representasi Kurva-S
- f. Representasi Kurva Bentuk Lonceng (*Bell Curve*)
- g. Koordinat Keanggotaan

Operator Dasar Operasi Himpunan Fuzzy

a. Operator AND

Diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y])$$

b. Operator OR

Diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y])$$

c. Operator NOT

Diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A[x]$$

Fungsi Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan fuzzy akan berhubungan dengan suatu relasi fuzzy. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

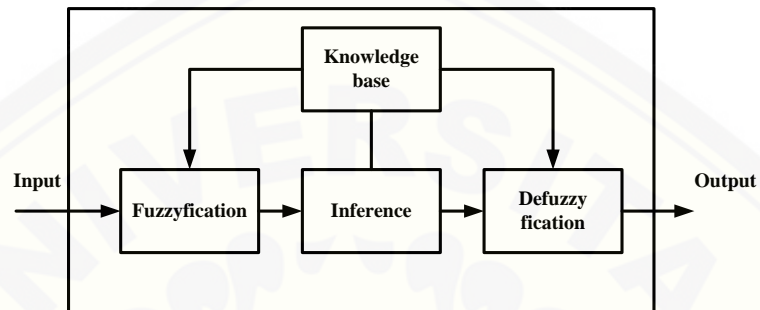
IF x is A THEN y is B

Dengan x dan y adalah skalar, dan A dan B adalah himpunan fuzzy. Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator fuzzy, seperti:

IF (x1 is A1) . (x2 is A2) . (x3 is A3) (Xx is Ax) THEN y is B

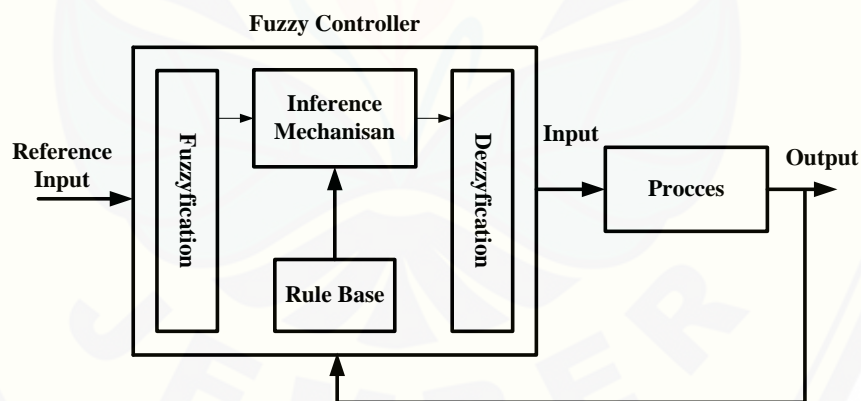
Dengan \bullet adalah operator (misal: OR, AND atau NOT). Min (*minimum*). Fungsi ini akan memotong *output* himpunan fuzzy.

2.5.4 Tahap Pemodelan dalam Fuzzy Logic



Gambar 2.12 Blok Diagram Sistem Fuzzy 1

Dari blok diagram di atas, bila diterapkan dalam pengendalian suatu proses dapat kita gambarkan seperti blok diagram di bawah ini :



Gambar 2.13 Blok Diagram Sistem Fuzzy 2

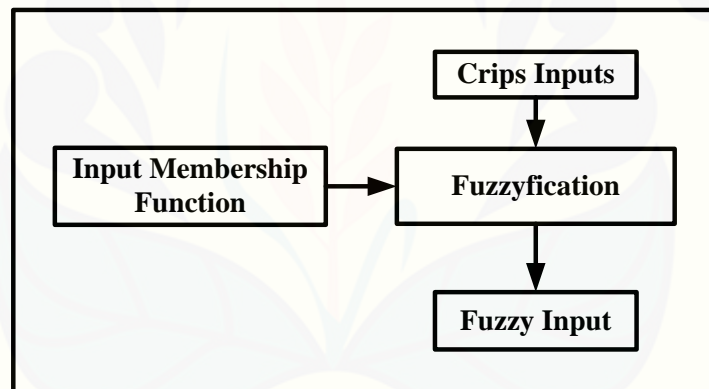
2.5.4.1 Fuzzification

Fuzifikasi atau *Fuzzyfication* adalah proses yang dilakukan untuk mengubah variabel nyata menjadi variabel fuzzy, ini ditujukan agar masukan kontroler fuzzy bisa dipetakan menuju jenis yang sesuai dengan himpunan fuzzy. Pemetaan

dilakukan dengan bantuan model dari fungsi keanggotaan agar dapat diketahui besar masukan tersebut (derajat keanggotaan). Terdapat beberapa jenis penggambaran fungsi keanggotaan, antara lain :

1. Gaussian
2. Segitiga
3. Trapesium
4. Bahu

Komponen lainnya yang memiliki peranan penting adalah Label. Label didefinisikan dari fungsi keanggotaan, fungsi keanggotaan apabila dikumpulkan akan menghasilkan fuzzy set. Dalam logika fuzzy terdapat *basispengetahuan*, dimana terdiri dari fakta (*data base*), dan kaidah atur (*rulebase*).



Gambar 2.14 Block *Fuzzification*

2.5.4.2 Rule Evaluation

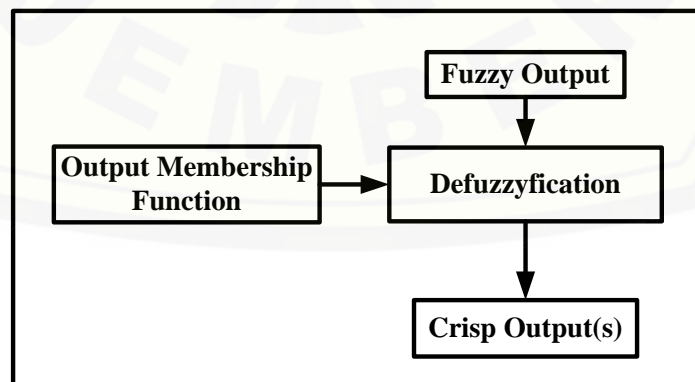
Prosedur fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel *numerik*) menjadi variabel fuzzy (variabel *linguistik*). Nilai *error* dan *delta error* yang dikuantisasi sebelumnya diolah oleh kontroler logika fuzzy, kemudian diubah terlebih dahulu ke dalam variabel fuzzy. Melalui *membership function* (fungsi keanggotaan) yang telah disusun, maka dari nilai *error* dan *delta error* kuantisasi akan didapatkan derajat keanggotaan bagi masing-masing nilai *error* dan *delta error*.

Pada umumnya aturan-aturan fuzzy dinyatakan dalam bentuk “*IF antecedent THEN consequent*” yang merupakan inti dari relasi fuzzy. Terdapat dua model aturan fuzzy yang digunakan secara luas dalam berbagai aplikasi, yaitu:

1. Model Mamdani
2. Model Sugeno

2.5.4.3 Defuzzification

Defuzzifikasi dapat didefinisikan sebagai proses pengubahan besaran fuzzy yang disajikan dalam bentuk himpunan-himpunan fuzzy keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya (*crisp*). Hal ini diperlukan sebab dalam aplikasi nyata yang dibutuhkan adalah nilai tegas (*crisp*). Prosesnya adalah ketika suatu nilai fuzzy *output* yang berasal dari rule evaluation diambil kemudian dimasukkan ke dalam suatu *membership function output*. Bentuk bangun yang digunakan dalam *membership function output* adalah bentuk *singleton* yaitu garis lurus vertikal ke atas, seperti yang ditunjukkan pada gambar di bawah. Besar nilai fuzzy *output* dinyatakan sebagai *degree of membership function output*. Nilai-nilai tersebut dimasukkan ke dalam suatu rumus yang dinamakan COG (*Center Of Gravity*) untuk mendapatkan hasil akhir yang disebut *crisp output*. *Crisp output* adalah suatu nilai analog yang akan kita butuhkan untuk mengolah data pada sistem yang telah dirancang.



Gambar 2.15 Blok Defuzzification

BAB 3. METODOLOGI PENELITIAN

Pada penelitian yang dilakukan ini, untuk memperoleh data dan hasil dari penelitian yang sesuai dengan tujuan, maka dilakukan beberapa tahapan-tahapan pada proses penelitian ini. Kegiatan penelitian sebagaimana dapat dijelaskan dalam susunan seperti berikut.

3.1 Alat dan Bahan

Dalam penelitian ini digunakan peralatan-peralatan sebagai penunjang pengambilan data. Sedangkan untuk penjelasan beberapa komponen lebih lengkapnya sebagaimana dijelaskan dalam Bab 2. Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

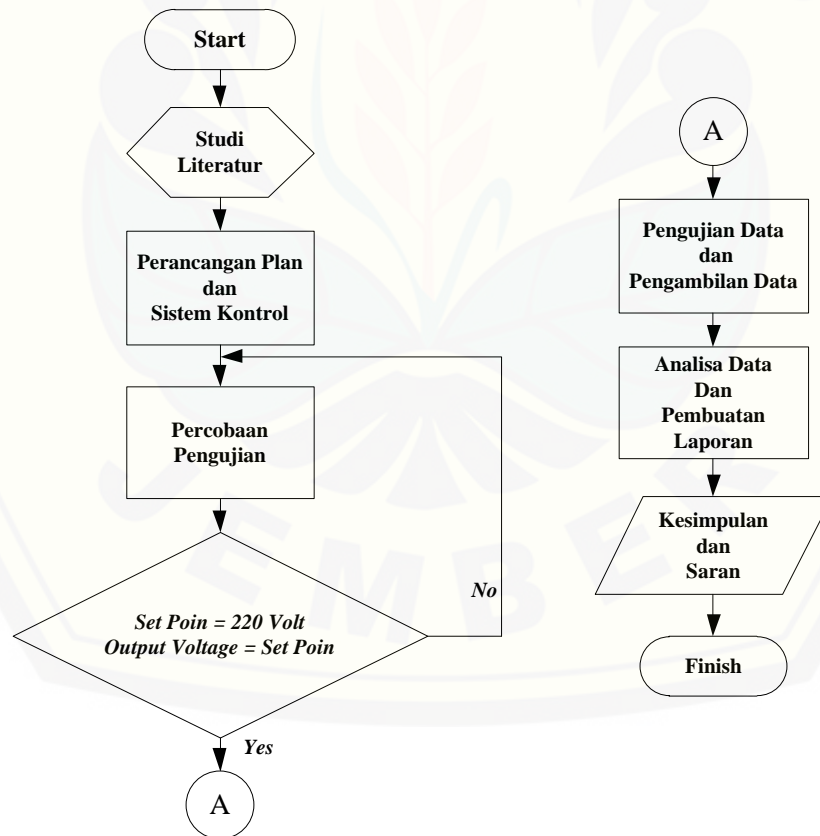
Tabel 3.1. Daftar Peralatan yang Digunakan

No	Nama Alat	Spesifikasi Teknis dan Penjelasan
1.	Digital Multimeter	Sanwa CD751 3200 Count/Bar Graph DC-1000 V AC-750 V MAX(CAT.II) DC-AC 600 V MAX(CAT.III)
2.	DC Power Supply	Protex DF1730B5A Single Output Power Supply 0 ~ 30V, 0 ~ 5A, Output terisolasi untuk pengoperasian seri atau parallel, Tampilan baca LED 3 digit
3.	Arduino Controller	ARDUINO UNO R3. ATmega328-AU microcontroller, Input voltage - 7-12V, 5V Electric current : 500MA, 3.3V Electric current : 50MA, 14 Digital I/O Pins (6 PWM outputs), 8 Analog Inputs, 32k Flash Memory, 16Mhz Clock Speed
4.	<i>Prime Over</i>	Motor DC kompon panjang
5.	Generator DC	Generator DC Penguatan Terpisah
6.	<i>Belt</i>	Bandi Fan Belt B42340 9,5x865La FM-34
7.	Tachometer	Lutron DT 2234C Tachometer. 0.1 – 5 – 999.9 rpm. 1-1000-99,999 rpm.
8.	AVR	Model: TDGC-2/0,5 ; INPUT 110/220 Volt ; OUTPUT 0-250 Volt ; 20-60 Hz ; I _{max} 8A ; CAP. 2KVA
9.	Sensor Tegangan	Terdiri dari 4 buah resistor yang dirangkai secara seri

	3 buah resistor 100 k Ω dan 1 buah resistor 4k7 Ω
10. <i>Buck Converter</i>	Menggunakan IGBT SSH40N50 ,2 buah kapasitor dengan, nilai 470 μ F, Inductor sebesar 22.0 mH, Resistor sebesar 914 Ω
11. <i>Rectifier</i>	Mnggunakan 4 buah dioda Masig-masing dengan tipe V06 6A10
12. Lampu Pijar	Phillip
	1 buah lampu pijar 15 Watt, 2 buah lampu pijar 25 Watt
13. Peralatan lain	<i>Jumper, bus bar, Obeng, Kunci</i> dan lain-lain

3.2 Tahapan Penelitian

Adapun tahap-tahap dalam penelitian dan penyusunan laporan ini secara keseluruhan dapat dinyatakan dalam diagram alir sebagaimana terlihat dalam Gambar 3.1



Gambar 3.1 *Flowchart* Penelitian

Tahapan-tahapan yang dilakukan pada penelitian sistem kontrol *fuzzy logic* pada generator DC penguatan terpisah berbasis arduino uno R3 adalah sebagai berikut :

1. Studi literatur

Tahap pertama memulai penelitian ini dilakukan dengan tahapan mencari sumber literatur yang berkaitan dengan penelitian ini, dengan harapan perancangan dan pelaksanaan penelitian dapat sesuai dengan yang diharapkan.

2. Perancangan plan dan sistem kontrol

Tahapan kedua adalah perancangan plant dan sistem kontrol dimana ditahap ini dilakukan perancangan beberapa komponen pengendali seperti sensor tegangan, *buck converter* dan plan secara keseluruhan.

3. Percobaan pengujian

Tahap ketiga adalah dimana dilakukan percobaan pengujian antara lain sensor tegangan untuk mendapat pembacaan sensor yang baik. Kemudian pengujian *buck converter* serta pengujian pada generator DC yang digunakan.

4. Pembuatan program *fuzzy logic* pada arduino

Tahap keempat adalah pemodelan sistem *fuzzy logic* dimana menentukan rancangan yang akan diimplementasi pada sistem kontrol. Tahapan yang dilakukan anatara lain *fuzzyfication*, *Inference* dan *defuzzyfication*. Pembuatan pemodelan sistem *fuzzy* ini dimana yang mengubah hasil dari pemodelan sistem *fuzzy* yang dirancang menjadi program agar dapat di *input* ke arduino uno R3.

5. Pengujian dan pengambilan data

Tahap kelima adalah pengujian dan pengambilan data dari sistem kontrol yang telah dibuat pada keseluruhan plan, dimana pada pengujian plan tersebut dilakukan pemberian beban yang bervariasi berupa lampu pijar.

6. Analisa data dan pembuatan laporan

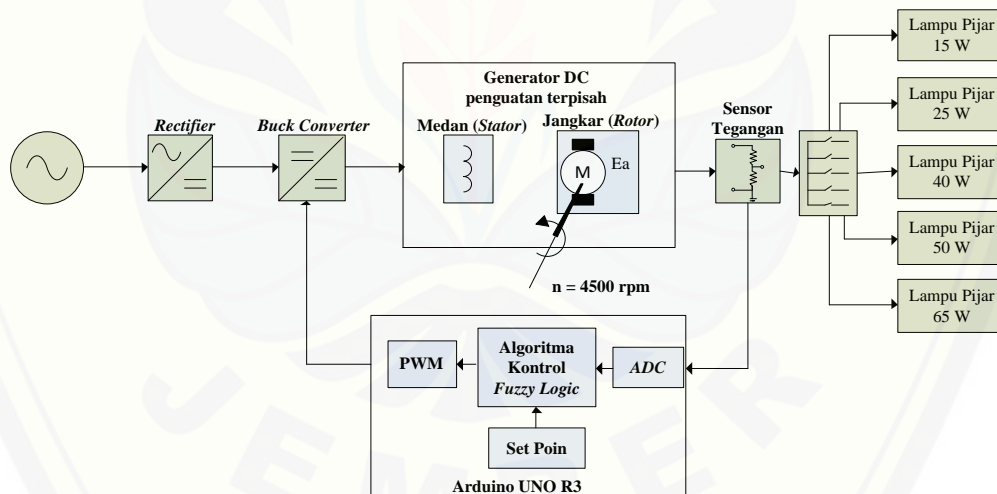
Tahap keenam adalah melakukan analisa pada data yang diperoleh dan membuatnya menjadi sebuah bentuk laporan hasil dari penelitian yang telah dilakukan.

7. Kesimpulan dan saran

Tahap ketujuh merupakan tahapan terakhir pada penelitian ini yaitu memberikan kesimpulan dari pengujian dan analisa data yang telah diperoleh, serta memberikan saran untuk penelitian selanjutnya..

3.3 Blok Diagram Sistem

Pada penelitian kali untuk mendapatkan data yang diinginkan dan sesuai dengan tujuan awal maka sesuai dengan topik yang diambil untuk bentuk dari blok diagram sistem secara keseluruhan dapat dilihat seperti pada gambar 3.2.



Gambar 3.2 Blok Diagram Sistem

Pada dasarnya prinsip kerja dari sistem kontrol *fuzzy logic* pada generator DC penguatan terpisah berbasis arduino uno R3, dapat dilihat dari gambar 3.2 yang merupakan gambaran dari blok diagram sistem secara keseluruhan. Generator yang digunakan dalam *plan* ini yaitu generator DC penguatan terpisah. *Prime mover* yang

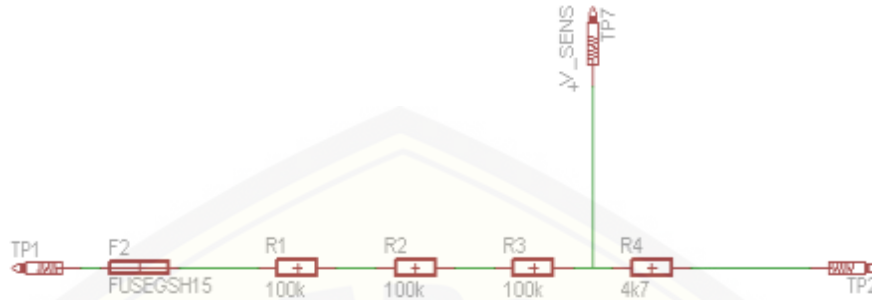
digunakan untuk memutar generator tersebut akan diputar dengan kecepatan sebesar 4500 rpm. Untuk membangkitkan tegangan *output* dari generator tersebut maka diberikan besarnya arus eksitasi yang diatur dari sebuah *buck converter*. *Buck converter* mengatur besarnya eksitasi berdasarkan *duty cycle*. Sedangkan untuk *duty cycle* diatur oleh sebuah kontrol yang berupa arduino uno R3 melalui besarnya nilai PWM. Didalam arduino ini terdapat kontrol berupa *fuzzy logic* yang akan merespon ketika pada generator tersebut dibebani dengan variasi beban dari lampu pijar. Pengujian beban menggunakan lampu pijar dengan variasi 15, 25, 40, 50 dan 65 Watt. Kontrol *fuzzy logic* ini juga yang akan mengatasi *drop* tegangan sehingga mempertahankan *output* tegangan tetap berada pada *set poin*. Sensor tegangan yang dipasang bertujuan untuk mengetahui *output* tegangan dari generator dan juga sebagai *input* dari arduino uno R3 yang berupa pembacaan ADC.

3.4 Perancangan Komponen Pengendali (*Hardware*)

3.4.1 Perancangan Sensor Tegangan

Sensor tegangan dibangun dari rangkaian pembagi tegangan. Perancangan sensor tegangan dilakukan dengan menyusun resistor secara seri. Rangkaian pembagi tegangan ini menggunakan resistor 300 k Ω dan resistor 4k7 Ω . Tujuan dari dirangkainya secara seri agar dapat membaca tegangan 0 - 324.15 V dan perbandingan tegangan untuk diberikan ke arduino uno R3 0 – 5 V.

Pada penelitian kali untuk penggunaan resistor sebesar 300 k Ω digantikan dengan menggunakan tiga buah resistor sebesar 100 k Ω secara seri karena untuk resistor 300 k Ω sukar dicari dipasaran. Untuk skema rangkaian sensor tegangan dapat dilihat pada gambar



Gambar 3.3 Skematik Sensor Tegangan

Untuk perhitungan dari perancangan sensor tegangan ini dapat dilihat seperti pada persamaa 3.1

$$V_{out} = \frac{V_{in} \times R_2}{(R_1 + R_2)} \quad (3.1)$$

dimana, $V_{out} = 5$; $V_{in} = 324.15$; $R_1 = 300 \text{ k}\Omega$

$$5 = \frac{324.15 \times R_2}{(300k + R_2)}$$

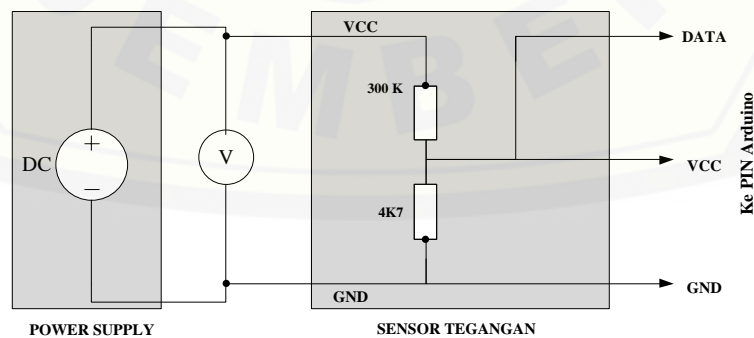
$$5 \times (300k + R_2) = 324.15 \times R_2$$

$$1500k + 5R_2 = 324.15R_2$$

$$R_2 = \frac{1500k}{319.15}$$

$$R_2 = 4k7 \Omega$$

Sedangkan untuk pengujian pada sensor tegangan ini sendiri dilakukan dengan cara seperti yang terlihat pada gambar 3.4.

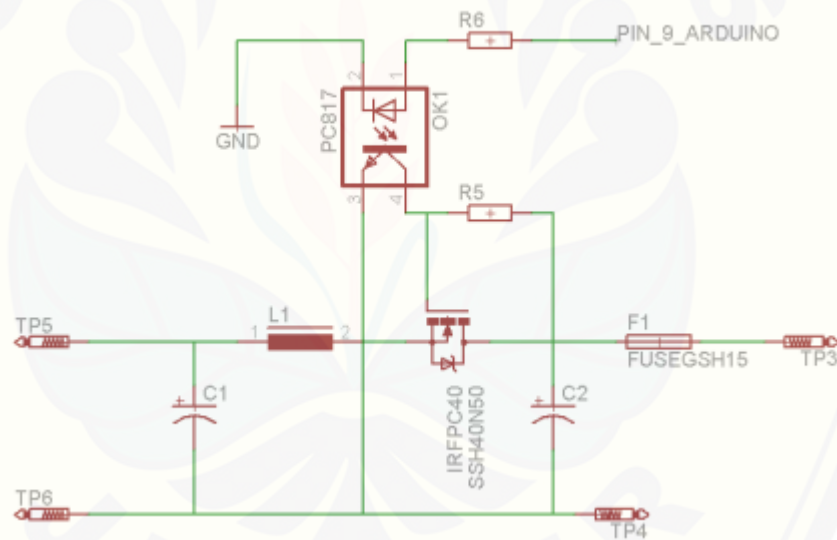


Gambar 3.4 Rangkaian Pengujian Sensor Tegangan

Dalam pengujian ini digunakan DC *power supply* sebagai sumber tegangan pada sisi *input* dari sensor tegangan. Kemudian sebuah digital multimeter untuk melakukan pengukuran tegangan dan nantinya dibandingkan antara pembacaan alat ukur dengan pembacaan dari sensor tegangan pada arduino.

3.4.2 Perancangan *Buck Converter*

Pada penelitian kali untuk mengatur besarnya eksitasi yang diberikan pada generator DC maka digunakan sebuah *buck converter* dimana untuk skematik rangkaian *buck converter* dapat dilihat pada gambar 3.5.



Gambar 3.5 Skematik Rangkaian *Buck Converter*

Pada *buck converter* ini menggunakan IGBT SSH40N50. IGBT tersebut mampu menahan tegangan antara *collector* dan *emitter* hingga 220 V, sedangkan *optocoupler* digunakan untuk memisah tegangan yang lebih tinggi dari tegangan yang digunakan pada komponen arduino uno R3 agar ketika terjadi kesalahan pada tegangan yang lebih tinggi, arduino uno R3 akan tetap aman dan juga berfungsi

sebagai *switching device* untuk IGBT. Nilai C1 dan C2 adalah 470 μF dan nilai induktor yang digunakan adalah 22.0 mH dengan hambatan beban pada medan generator DC yang digunakan adalah 914 Ω . Agar *buck converter* bekerja secara kontinu diperlukan perhitungan nilai frekuensi yang digunakan, dengan persamaan 3.2.

$$f > \frac{(1-D)R}{2L} \quad (3.2)$$

dimana, R = 914 Ω dan L = 22 mH

$$f > \frac{(1-D)914}{2(22 \times 10^{-3})}$$
$$f > 20772.72 (1-D)$$

jadi agar semua kondisi *duty cycle* terpenuhi maka didapatkan nilai frekuensi harus lebih dari 20,772.72 Hz.

3.4.3 Generator DC Penguatan Terpisah

Pada penelitian ini untuk generator DC yang digunakan seperti yang telah dijelaskan sebelumnya yaitu menggunakan generator DC penguatan terpisah dimana generator DC ini merupakan tipe *feedback* dengan nomer seri 63-120 yang memiliki tegangan kerja yaitu sebesar 220 Volt dengan daya yang mampu dihasilkan yaitu sebesar 250 Watt. Generator DC penguatan terpisah ini bekerja pada kecepatan nominal yaitu 2000 rpm dan dengan kecepatan maksimum pada 6000 rpm.



Gambar 3.6 Generator DC

(Sumber : *feedback mechine datasheet*)

Untuk kumparan medan pada stator yang terdapat pada generator DC ini terdiri dari dua jenis kumparan, yaitu kumparan pendek dan kumparan panjang. Sedangkan untuk jenis rangkaian yang dapat digunakan pada generator DC ini yaitu dirangkai secara seri, *shunt* dan *compuond*.

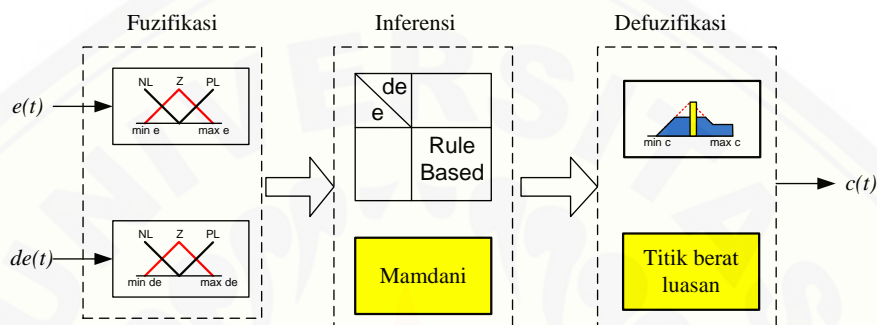
3.4.4 Konfigurasi PIN Arduino UNO R3

Sebagai kontroler yang digunakan dalam *plant*, arduino ini sendiri sebelumnya telah dijelaskan dalam bab 2. Namun pada tahap ini penjelasannya akan lebih spesifik dalam penggunaannya sebagai *embedded system*. Arduino yang digunakan yaitu dengan tipe arduino uno R3 dimana *embedded system* ini yang menerima *feedback* tegangan dari *output* tegangan generator. Pembacaan nilai *output* tegangan generator ini melalui sensor tegangan yang menggunakan proses ADC dan terhubung pada pin arduino yaitu pada pin A0. Untuk ADC sendiri pada arduino terdapat pada pin A0-A5. Sedangkan untuk pin yang mengontrol *output* PWM pada *buck converter* menggunakan pin 9. Pada arduino sendiri untuk pin-pin yang dapat mengontrol PWM yaitu terdapat pada pin 3, 5, 6, 9 dan 10 (Arduino LLC, 2015). Frekuensi PWM yang dapat dihasilkan pada pin PWM ini adalah 31,250 Hz. Pada tahap ini, pengujian *embedded system* dilakukan dengan mencoba menghubungkan pada pin arduino yang diinginkan karena tidak jarang pin pada arduino juga tidak dapat digunakan dengan sehingga dibutuhkan pengecekan ulang, seperti penerimaan pembacaan ADC oleh sensor.

3.5 Perancangan Sistem Kendali Fuzzy

Perancangan kontrol fuzzy pada sistem eksitasi pada generator DC penguatan terpisah ini dirancang dengan perhitungan secara sistematis. Dimana *logika fuzzy* ini akan diimplementasikan secara langsung kedalam sebuah *embedded system* yang berupa arduino uno R3. Tahap pembuatan *fuzzy logic control* ini dilakukan melalui

beberapa tahap yaitu dimulai dari tahap *fuzzyfikasi* dimana terdiri dari pembuatan *membership fuction*, kemudian dilanjutkan ketahap *inferent* dimana pada tahap ini merupakan proses pembuatan *rule base* dan kemudian ketahap *defuzzyfikasi* yaitu proses *output* yang diperoleh dari keseluruhan proses. Tahapan-tahapan *fuzzy logic control* dapat dilihat dengan jelas pada gambar 3.7.

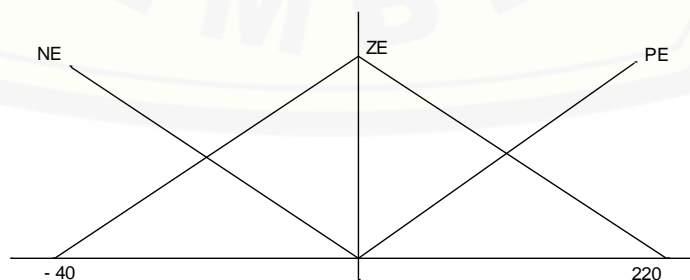


Gambar 3.7 Blok Sistem Kendali Fuzzy

3.5.1 Tahap Fuzzyfikasi

Tahap awal pembuatan fuzzy diawali dengan proses fuzzyfiasi dimana pada tahap ini dilakukan terhadap *error* dan *delta error*, yaitu dengan membuat himpunan keanggotaan untuk *error* dan *delta error*, dan kemudian melakukan perhitungan untuk menentukan derajat keanggotaan masing-masing nilai *error* dan *delta error*.

Pada tahap ini terlebih dahulu ditentukan besarnya *setpoint* yang digunakan untuk membuat *membership fuction* dan juga pada proses perhitungan *error* dan *delta error*. Untuk besarnya nilai *setpoint* yang digunakan yaitu 220, dimana nilai ini merupakan yang menjadi acuan dalam kontrol agar tetap stabil pada nilai tersebut.



Gambar 3.8 Membership Function Error

Dalam sistem pengendalian ini bertujuan untuk mengendalikan tegangan, maka kendali dilakukan berdasarkan *error* (e) tegangan dan perubahan *error* tegangan atau *delta error* tegangan (de). Kontroler mengindra sinyal dari sensor tegangan dan menerima *input* berupa tegangan referensi. Kontroler memproses selisih tegangan aktual dan tegangan referensi menjadi $e(t)$ dan mencatatnya. Kontroler ini kemudian memproses selisih *error* ini dengan nilai *error* tegangan sebelumnya, $e(t-1)$ dan menjadikannya perubahan *error* tegangan $de(t)$. Hasil perhitungan *error* dan *delta error* ini kemudian diproses sesuai dengan tahapan pada blok sistem kendali fuzzy.

Nilai *error* diperoleh dari nilai *setpoint* dikurangi dengan *output* tegangan dari generator yang berasal dari pembacaan sensor tegangan. Untuk perhitungan *membership function error* dalam proses pembuatan program pada arduino ini dapat dilihat seperti dibawah ini

a) Jika $-40 < E < 0$

$$NE_{-} = \frac{-E}{40}$$

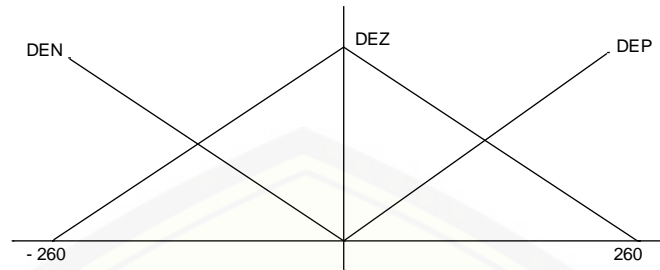
$$ZE = \frac{E+40}{40}$$

b) Jika $0 < E < 220$

$$ZE_{-} = \frac{220-E}{220}$$

$$PE = \frac{-E}{220}$$

Hampir sama pada pembuatan *membership function error*, untuk pembuatan *membership function delta error* dilakukan juga berdasarkan perhitungan sesuai dengan aturan pada pembuatan fuzzy pada umumnya. Besarnya nilai *delta error* dihitung dari besarnya nilai *error* sekarang dikurangi dengan nilai *error* sebelumnya.



Gambar 3.9 Membership Fucntion Delta Error

a) Jika $-260 < E < 0$

$$DEN_ = \frac{-DE}{260}$$

$$DEZ = \frac{DE+260}{260}$$

b) Jika $0 < E < 260$

$$DEZ_ = \frac{260-DE}{260}$$

$$DEP = \frac{DE}{260}$$

3.5.2 Tahap Inferensi

Inferensi dimulai dengan memetakan nilai derajat keanggotaan dari masing-masing *error* dan *delta error* ke sebuah tabel yang berisi basis aturan yang dipilih (rule base) dan menggunakan sebuah metode untuk pemilihan nilai dari derajat keanggotaan. Dengan demikian terlebih dahulu harus dibuat matrik atau tabel *rule base*.

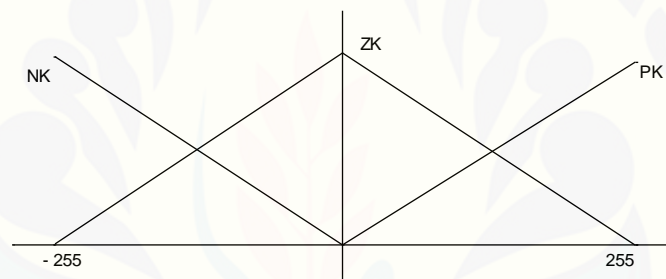
	E			
DE		PE	ZE	NE
DEN	U1	NK	NK	PK
DEZ	U4	NK	ZK	PK
DEP	U7	NK	PK	PK

Gambar 3.10 Rule Base

Setelah langkah ini maka proses inferensi selesai dan untuk selanjutnya adalah langkah defuzifikasi untuk menentukan besarnya nilai PWM yang dikirim ke *buck converter*.

3.5.3 Tahap Defuzifikasi

Defuzifikasi dilakukan dengan memetakan nilai-nilai U ke himpunan defuzifikasi yang dinyatakan dengan persamaan-persamaan garis. Berbeda dengan fuzifikasi, defuzifikasi akan menghasilkan luasan-luasan yang dibatasi dengan tinggi masing-masing nilai U dan garis yang dihasilkan dari pemetaan ke *rule base*.

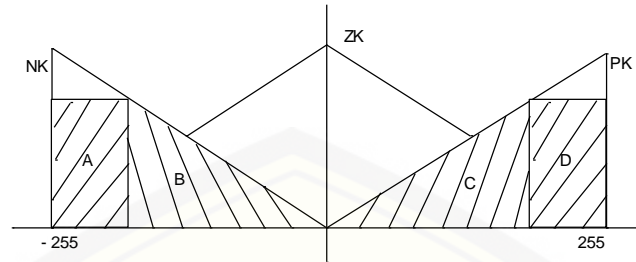


Gambar 3.11 *Membership Function Control*

Penentuan nilai kontrol (c) ditentukan dengan perhitungan titik berat luasan-luasan hasil pemetaan ini, atau diistilahkan dengan *weighted average method*. Secara matematik, titik berat ini dihitung sebagai berikut (Istraniady, Andrian, & Mardiani, 2013):

$$Z = \frac{\sum_{i=1}^9 \alpha_i * \alpha w_i}{\sum_{i=1}^9 \alpha_i} \quad (3.3)$$

$$Z = \frac{WANK1 \times U1 + WANK2 \times U2 + WAPK3 \times U3 + WANK4 \times U4 + WAZK \times U5 + WAPK6 \times U6 + WANK7 \times U7 + WAPK8 \times U8 + WAPK9 \times U9}{U1 + U2 + U3 + U4 + U5 + U6 + U7 + U8 + U9}$$



Gambar 3.12 Perhitungan Titik Berat Luasan

NK / Negatif Kontrol terletak pada bangun A dan B. Untuk perhitungannya seperti dibawah ini

- $LA = ((1 - U) \times 255) \times U$
- $XA = ((1 - U) \times (-255)) / 2 + (-255 \times U)$
- $LB = 0,5 \times U \times (255 \times U)$
- $XB = ((U \times -255) \times 2) / 3$

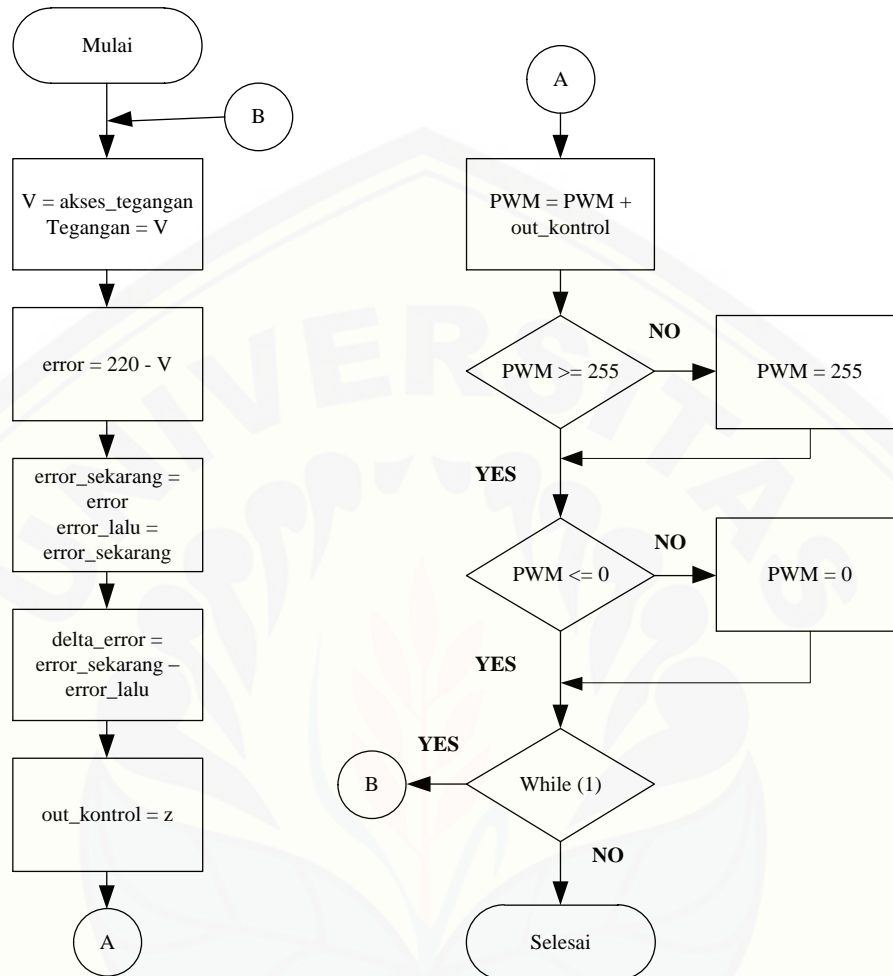
$$\text{Sehingga } WANK = \frac{LA \times XA + LB \times XB}{LA + LB}$$

PK / Positif Kontrol terletak pada bangun C dan D. Untuk perhitungannya seperti dibawah ini

- $LC = 0,5 \times U \times (255 \times U)$
- $XC = ((U \times 255) \times 2) / 3$
- $LD = ((1 - U) \times 255) \times U$
- $XD = ((1 - U) \times (-255)) / 2 + (-255 \times U)$

$$\text{Sehingga } WAPK = \frac{LC \times XC + LD \times XD}{LC + LD}$$

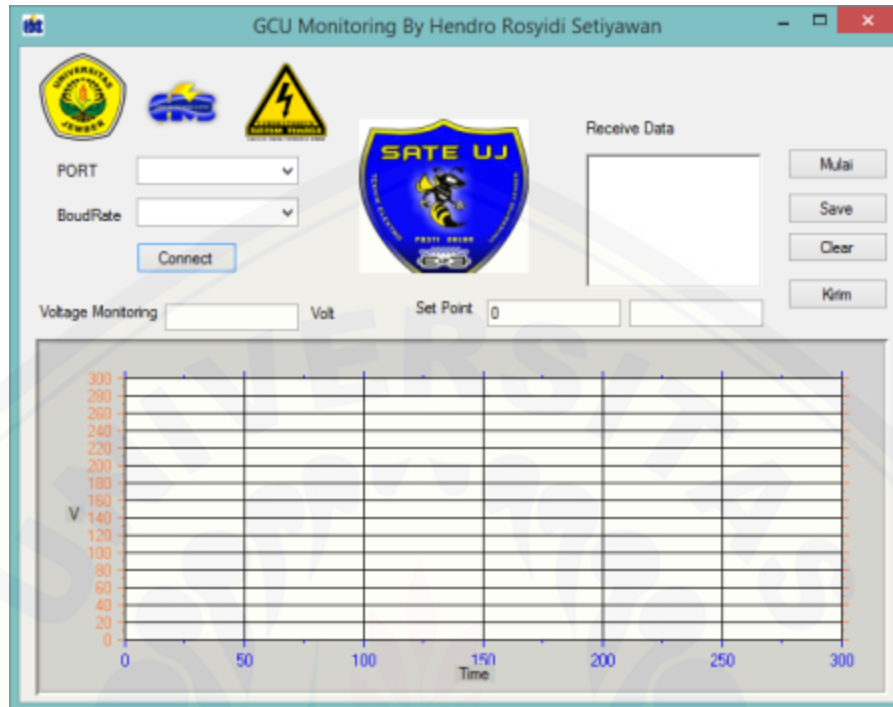
Setelah semua tahap pembuatan fuzzy selesai maka setelah itu dilakukan pengujian untuk pengambilan data. Secara umum alur dari program yang dibuat pada arduino dapat dilihat pada gambar 3,13



Gambar 3.13 Flowchart Program

3.6 Aplikasi Monitoring Generator

Pada penelitian ini aplikasi yang digunakan pada proses monitoring yaitu dengan *software* Microsoft Visual C# 2010 Express. Aplikasi ini terhubung secara serial dengan arduino uno R3 sehingga proses pembacaan tegangan oleh sensor serta respon kontrol *setpoint* yang digunakan dapat dilihat secara langsung dari aplikasi ini.



Gambar 3.14 Aplikasi Monitoring Generator

Didalam aplikasi ini juga terdapat fungsi *logger* dimana untuk pembacaan data tegangan yang nantinya akan diolah dapat tersimpan secara langsung. Tegangan keluaran dari generator juga akan terlihat secara langsung pada grafik yang telah dibuat sehingga untuk kontrolnya dapat terlihat langsung pada saat pengujian.

BAB 5. PENUTUP

Dari hasil-hasil pengujian dan pembahasan yang telah dilakukan dalam Bab 4 dapat diambil beberapa kesimpulan dan saran-saran yang dapat digunakan sebagai dasar untuk penelitian selanjutnya.

5.1 Kesimpulan

Dari pengujian dan analisa data yang telah dilakukan pada penelitian yang berjudul “ Sistem Kontrol *Fuzzy Logic* pada Generator DC Penguatan Terpisah Berbasis Arduino UNO-R3 ”, maka dapat diambil beberapa kesimpulan sebagai berikut.

1. Pada perancangan sistem kontrol *fuzzy logic* untuk kontrol tegangan keluaran pada generator DC penguatan terpisah untuk pengujian sensor tegangan diperoleh persentase *error* pembacaan sensor yaitu 0,37 %. Sedangkan pada pengujian efisiensi tertinggi pada *buck converter* didapatkan sebesar 87 % ketika *duty cycle* sebesar 100 %.
2. Kestabilan tegangan keluaran generator DC ketika menggunakan kontrol *fuzzy logic* saat pengujian pemberian beban tertinggi didapatkan *under voltage* sebesar 211 Volt dengan *recovery-time* sebesar 2,6 s. Sedangkan ketika pengujian pelepasan beban tertinggi didapatkan *over voltage* tertinggi sebesar 228 Volt dengan *recovery-time* sebesar 1,4 s. Dimana untuk pengujian beban tertinggi yaitu sebesar 65 Watt.

5.2 Saran

Setelah melakukan penelitian ini, ada beberapa saran yang dapat digunakan untuk pengembangan penelitian yang selanjutnya. Beberapa saran tersebut antara lain sebagai berikut.

1. Dalam pembuatan *membership fuction* untuk kontrol *fuzzy logic* dapat dibuat kembali dengan pola yang lain sehingga didapatkan respon kontrol lain yang lebih baik.
2. Generator DC penguatan terpisah yang digunakan dapat diganti dengan jenis penguatan lainnya seperti *shunt*, *series* atau *compound*.
3. Penggunaan kontrol dapat diganti dengan *hybrid* PID Fuzzy sehingga dapat diperoleh respon kontrol yang lebih baik.
4. *Boost converter* atau *buck-boost converter* dapat digunakan untuk menggantikan *buck converter*.
5. *Prime over* yang digunakan dapat diganti dengan motor bakar atau yang lain dengan kecepatan yang konstan atau stabil.
6. *Embedded system* yang dipakai dapat diganti dengan *embedded system* yang terbaru.

DAFTAR PUSTAKA

- Arduino LLC. (2015). www.arduino.cc/en/Main/ArduinoBoardUno. Dipetik February 24, 2016, dari [arduino.cc/en](http://www.arduino.cc/en).
- Arissandi, R. (2014). Implementasi Kontrol Logika Fuzzy (KLF) sebagai Pengendali Suhu Sistem Pasturisasi Kuning Telur Cair Berbasis Mikrokontroler Arduino.
- Autodesk.Inc. (2015). [id/Arduino-Voltage-Sensor-0-25V/](http://www.instructables.com/id/Arduino-Voltage-Sensor-0-25V/). Dipetik Maret 11, 2016, dari instructables: <http://www.instructables.com/id/Arduino-Voltage-Sensor-0-25V/>
- Chapman, S. J. (2005). *Electric Machinery Fundamentals*. New York: Library of Congress Cataloging.
- Fajar, A., Hadi, W., & Kaloko, B. S. (2016). Design PID Control Of Output Voltage Separately Excited DC Generator Based Arduino UNO R3.
- Febrianto, V. N. (2014). Aplikasi Kontrol PID Untuk Pengaturan Putaran Motor DC pada Alat Pengepres Adonan Roti (Screw Conveyor). *II*.
- Guldemir, H. (2011, September). Study of Sliding Mode Control of DC-DC Buck Converter. *Scientific Research*.
- Hadiyanto, R., Widjonarko, & Supriyadi Prasetyono. (2015). Prototype Sistem Compressed Air Energy Storage dengan Pengaturan Valve Menggunakan Fuzzy Logic Controller. *BERKALA SAINSTEK*.
- Hameyer, K. (2001). *Electrical Machine*. Bundesland NRW, Jerman.
- Himawan, H. M., Setyawati, O., & Suyono, H. (2016, Februari). Pemodelan Fuzzy Logic Control untuk Pengendali PWM pada Buck Converter. *JNTETI, Vol. 5, No. 1*.
- Istraniady, Andrian, P., & Mardiani. (2013). Analisis Perbandingan Metode Fuzzy Tsukamoto dan Metode Fuzzy Mamdani pada Perbandingan Harga Sepeda Motor Bekas.

- Khadiri, K. E., & Qjidaa, H. (2014, March). Inverting Buck-Boost DC-DC Converter Design Challenges. *International Journal of Embedded Systems and Applications (IJESA)*, IV.
- Kusumadewi, S., & Hartati, S. (2006). *NEURO FUZZY: Integrasi Sistem Fuzzy & Jaringan Saraf*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S., & Purnomo, H. (2010). *Aplikasi Logika Fuzzy untuk Pendukung Keputusan edisi 2*. Yogyakarta: Graha Ilmu.
- Suratno. (2016). Pengaruh Perbedaan Tipe Fungsi Keanggotaan Pada Pengendali Logika Fuzzy Tanggapan Waktu Sistem Orde Dua Secara Umum .

LAMPIRAN

A. Listing Program Arduino UNO-R3

```
String textFromSerial = "";
boolean parsing = false;
String inData = "";
String data[10];
float data_masuk = 0;

float V = 0.00;
float error = 0.00;
float setpoint = 0;
float delta_error = 0.00;
float error_sekarang;
float error_lalu;

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;

float sens =0.00;
int tegangan = 0;

int pwm;
int con;
int out_kontrol;

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

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 < -40){
    error = -40;
}
if(error > 220){
    error = 220;
}

error_sekarang = error;
delta_error = error_sekarang - error_lalu;

if(delta_error < -260){
    delta_error = -260;
}
if(delta_error > 260){
    delta_error = 260;
}

error_lalu = error_sekarang;

if(error < 0 && error >= -40){
    NE_ = (-error)/40;
    ZE = (error + 40)/40;

    if(delta_error < 0 && delta_error >= -260){
        DEN_ = (-delta_error)/260;
        DEZ = (delta_error + 260)/260;

        U1 = min(NE_,DEN_);
        U4 = min(NE_,DEZ);
        U2 = min(ZE,DEN_);
        U5 = min(ZE,DEZ);
    }
}
if(delta_error >= 0 && delta_error <= 260){
    DEZ_ = (260 - delta_error)/260;
    DEP = delta_error/260;

    U4 = min(NE_,DEZ_);
```

```

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

if(error >= 0 && error <= 220){
ZE_ = (220 - error)/220;
PE = error/220;

if(delta_error < 0 && delta_error >= -260){
DEN_ = (-delta_error)/260;
DEZ = (delta_error + 260)/260;

U2 = min(ZE_,DEN_);
U5 = min(ZE_,DEZ);
U3 = min(PE,DEN_);
U6 = min(PE,DEZ);
}
if(delta_error >= 0 && delta_error <= 260){
DEZ_ = (260 - delta_error)/260;
DEP = delta_error/260;

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;

```

```

}

```

```

void kontrol() {
  pwm = pwm + out_kontrol;
  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==1.00){

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

  }else{

  }

}

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();
  setpoint = data[1].toFloat();

  parsing=false;
  inData = "";
}
```

```
}  
  
void kirim_data_ke_PC(){  
  Serial.print(setpoint);  
  Serial.print(" ");  
  Serial.print(tegangan);  
  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 WindowsFormsApplication1
{
    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_FormClosed(object sender, FormClosedEventArgs 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("Plase Choise PORT and BoudRate !!!");
        }
    }
    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[1];
    textBox2.Text = data[0];
    richTextBox1.AppendText(data[1] + "\n");
    detik++;
    axMathworks_Strip2.AddXY(0, detik, Convert.ToDouble(data[1]));
}

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

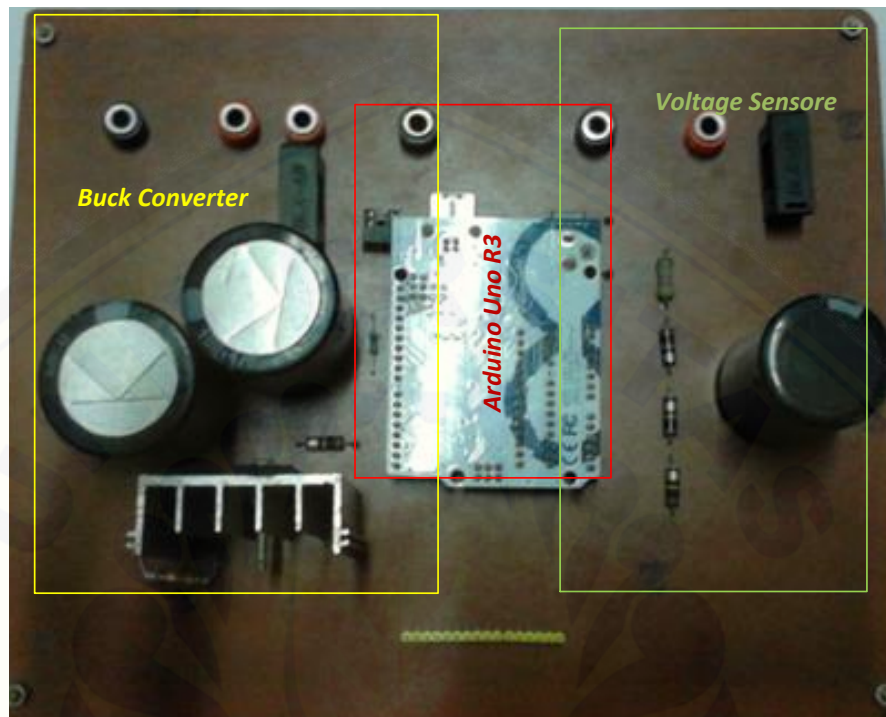
private void button3_Click(object sender, EventArgs e)
{
    richTextBox1.SaveFile("D:\\FileLogger\\data_logger.txt",
RichTextBoxStreamType.PlainText);
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    richTextBox1.Text = "";
    SetPoint.Text = "0";
}
private void KIrim_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (Convert.ToInt32(SetPoint.Text) <= 220 && SetPoint.Text !=
""))
        {
            if (Mulai.Text == "Stop")
            {
                serialPort1.Write("1" + " "); //Mulai
                serialPort1.Write(SetPoint.Text + "\n"); //setPoint
            }
            else if (Mulai.Text == "Mulai")
            {
                MessageBox.Show("Can't Transfer Data Because Serial
Stopped !!!");
            }
        }
        else
        {
            MessageBox.Show("SetPoint Over Load > 220");
        }
    }
    else
    {
        MessageBox.Show("No Connection !!!");
    }
}
private void Mulai_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (Mulai.Text == "Mulai")
        {
            serialPort1.Write("1" + " "); //Mulai
            serialPort1.Write(SetPoint.Text + "\n"); //setPoint
            Mulai.Text = "Stop";
        }
        else if (Mulai.Text == "Stop")
        {
            serialPort1.Write("0" + " "); //Mulai
            serialPort1.Write(SetPoint.Text + "\n"); //setPoint
            Mulai.Text = "Mulai";
        }
    }
    else
    {
```

```
        MessageBox.Show("No Connection !!!");
    }
}
private void axMathworks_Strip2_CursorChange(object sender,
AxMathworks_STRIPLib._DStripEvents_CursorChangeEvent e)
{
}
}
```



C. Gambar *Hardware*



Gambar *Buck Converter* dan *Voltage Sensor*



Gambar *Generator DC Penguatan Terpisah*



Gambar Beban Lampu Pijar



Gambar Keseluruhan Sistem Saat Pengambilan Data



Gambar Pengujian Alat dan Pengambilan Data



Gambar Keseluruhan Sistem Saat Pengambilan Data