



**RANCANG BANGUN *DC TO DC BOOST CONVERTER* PADA
ROBOT KRSBI BERODA JR – EVO BERBASIS *FUZZY LOGIC***

SKRIPSI

Oleh

Adittyia Aprillia Arganata

NIM 151910201114

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

2019



**RANCANG BANGUN *DC TO DC BOOST CONVERTER* PADA
ROBOT KRSBI BERODA JR – EVO BERBASIS *FUZZY LOGIC***

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi syarat – syarat
Untuk menyelesaikan Program Studi Teknik Elektro (S1)
Dan mencapai gelar Sarjana Teknik

oleh

Adittyaprillia Arganta

NIM 151910201114

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2019

PERSEMBAHAN

Alhamdulillah, puji syukur kepada Allah SWT. Berkat rahmat, taufik dan hidayah-Nya saya bisa menyelesaikan tugas akhir ini. Sholawat serta salam kuberikan kepada junjungan Nabi Muhammad SAW yang sebagai suri taulandanku. Dengan ketulusan hati kupersembahkan laporan tugas akhir ini kepada orang yang senantiasa membantu dan membimbingku, yaitu :

1. Keluargaku, Bapak Agus Purwanto, Ibu Lilis Suryaningsih dan Adik Reza Sandra Dewi yang tersayang. Merekalah yang senantiasa mendukungku, menyematiku, mengasihku dan menyayangiku dari lahir sampai sekarang. Terima kasih atas semua yang telah diberikan kepadaku.
2. Semua guru, dan dosenku yang telah membimbing dan memberikan ilmu yang tak dapat kuhitung berapa banyaknya barokah dan doanya.
3. Almamater Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.
4. Sahabatku semua terkhusus Teknik Elektro Unej UJ 15 di Universitas Jember.
5. Teman seperjuangan dan keluarga besar Tim Robotika Unej yang telah bersama sama berjuang untuk mengharumkan almamater Universitas Jember dikancah regional dan nasional.
6. Keluarga besar Laboratorium Sistem Tenaga yang telah menemani berproses di Teknik Elektro Universitas Jember.
7. Teman seperjuangan Abdul Kharis Ismail, S.T. yang telah berjuang bersama sama dalam proses penyelesaian studi di Universitas Jember
8. Kepada Saudari Dinda Widayanti Mochtar, S.E. yang selalu memberikan suport.
9. Kepada Abah Zein BMW yang selalu memberi secangkir inspirasi dikala proses penyelesaian tugas akhir mulai menemui jalan buntu.

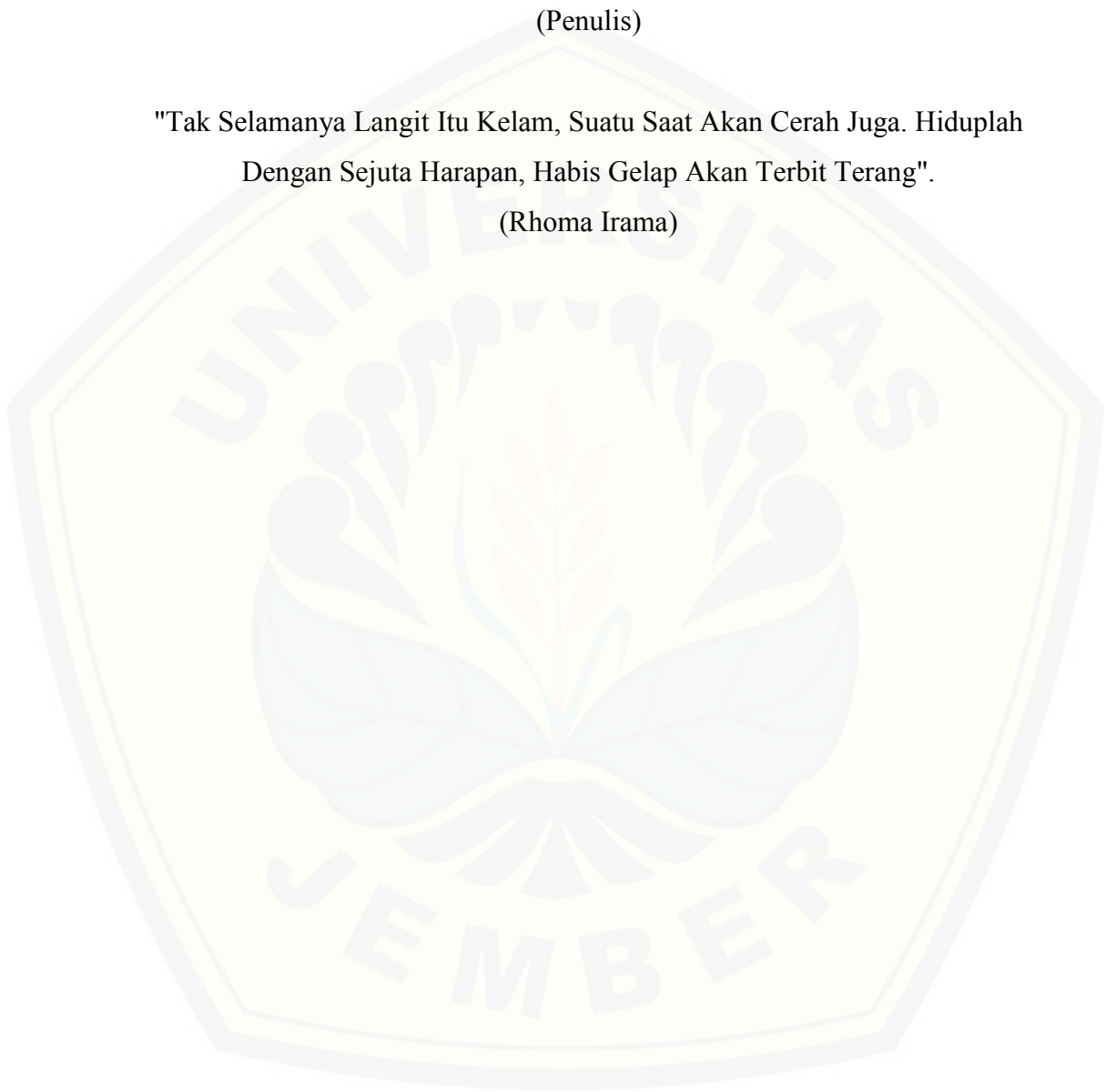
MOTTO

“Jangan Salahkan Waktu Yang Cepat Berlalu, Tapi Salahkan Dirimu Yang Telat
Melakukan Sesuatu”.

(Penulis)

"Tak Selamanya Langit Itu Kelam, Suatu Saat Akan Cerah Juga. Hiduplah
Dengan Sejuta Harapan, Habis Gelap Akan Terbit Terang".

(Rhoma Irama)



PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Adittyta Aprillia Arganta

NIM : 151910201114

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ” Rancang Bangun *Dc To Dc Boost Converter* Pada Robot Krsbi Beroda Jr – Evo Berbasis *Fuzzy Logic*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 31 Mei 2018

Yang menyatakan,

Adittyta Aprillia Arganta

NIM 151910201114

SKRIPSI

**RANCANG BANGUN *DC TO DC BOOST CONVERTER* PADA
ROBOT KRSBI BERODA JR – EVO BERBASIS *FUZZY LOGIC***

Oleh

Aditya Aprillia Arganata

NIM 151910201114

Pembimbing :

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko, S.T., M.T.

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

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Implementasi robot selama ini telah dapat membantu manusia diberbagai bidang terutama dibidang industri. AI (*Artificial Intelligence*) memegang peran penting dalam dunia robotika karena memungkinkan robot dapat bergerak secara otomatis hanya dengan perintah sederhana. Untuk mengembangkan dunia robotika lebih jauh, para peneliti mencoba membuat robot untuk melakukan berbagai macam olahraga dan permainan yang dibuat manusia. (Tony, 2010)

Mobile robot otonom adalah suatu jenis robot cerdas yang mempunyai kemampuan untuk membuat keputusan sendiri, memiliki sistem kendali dan catu daya yang terintegrasi dan mempunyai kemampuan navigasi yaitu sejumlah operasi yang memungkinkan robot mencapai tujuan yang di harapkan. Robot jenis ini memiliki berbagai macam penggerak salah satunya adalah motor. Karena robot *mobile* memiliki karakteristik yang sama dengan konsep dasar permainan sepak bola, atas dasar itulah robot jenis ini di pilih sebagai konsep robot sepak bola. (Gunawan Wibisono, 2011)

Robocup Middle Size League merupakan sebuah kompetisi robot tingkat dunia yang diadakan setiap tahun. Pertandingan ini di mulai pertama kalinya pada tahun 1997. Satu tim di wakili oleh lima robot yang dapat bergerak secara *autonomous* tanpa ada campur tangan manusia selama pertandingan, kecuali untuk mengambil robot jika robot mengalami gangguan. (Robin Soetns, 2015)

Robot sepak bola beroda dalam kontes robot indonesia memiliki regulasi yang mengatur tentang berapa tegangan baterai yang diperbolehkan saat perlombaan. Khususnya untuk menjalankan motor dc tipe PG 45 di butuhkan tegangan 24 V. Akan tetapi jika keseluruhan tegangan di salurkan kepada motor dc maka ada beberapa komponen elektronika seperti sistem penendang yang menggunakan *solenoid* tidak dapat berkerja secara maksimal yang mengakibatkan jarak tendang yang berkurang. Selain itu rangkaian sistem minimum *microcontroller* akan mengalami kegagalan dalam menjalankan operasi atau perintah yang seharusnya.

DC to DC Converter tipe *Boost converter* merupakan suatu rangkaian elektronika daya yang berfungsi menstabilkan tegangan dengan menaikkan tegangan dimana tegangan keluaran lebih tinggi dari tegangan masukan tanpa harus menghilangkan daya yang relatif besar sehingga dapat mengatasi kekurangan tegangan. Pengendalian dilakukan dengan memberikan sinyal/tegangan yang mengatur waktu ON dan waktu OFF *switch*. nilai tegangan berdasarkan nilai *duty cycle* yang merupakan satuan dari PWM (*Pulse With Modulation*). (Ahmad Fathurachman, 2014)

Dalam kenyataannya, tegangan masuk pada *boost converter* tidak sesuai dengan yang di harapkan atau fluktuatif. Atas dasar itu tegangan keluaran harus dikontrol agar sesuai dengan nilai awal yang telah ditetapkan. Pengaturan nilai PWM dengan suatu metode dapat mempermudah untuk mendapatkan nilai tegangan keluaran yang konstan pada *bost converter*.

Kontrol PWM dengan metode *fuzzy logic* dapat diterapkan untuk mengatur nilai tegangan keluaran pada *boost converter* bila terjadi penurunan tegangan dikarenakan meningkatnya arus pada beban.

DC to DC Converter tipe *Boost converter* dengan metode *fuzzy logic* ini akan diimplementasikan pada motor dc yang di gunakan pada robot KRSBI BERODA. Karena adanya regulasi yang ditetapkan oleh pihak penyelenggara tentang penggunaan tegangan pada baterai maksimal 12 V dengan tolenransi sekitar 10 % jadi, sekitar 13,2 V dan motor dc yang digunakan pada robot JR-EVO membutuhkan tegangan 24 V. Maka dari itu *Boost converter* dengan metode *fuzzy logic* di terapkan untuk menaikkan tegangan baterai dari teganan nominal baterai LiPo 11,1 V ke tegangan 24 V agar dapat menjalankan motor dc pada robot. Selain itu, untuk kedepanya semua sistem pada robot yang dilombakan harus hasil kreatifitas dan pengembangan dari tim atau mahasiswa itu sendiri. Jadi, untuk kedepanya semua robot tidak lagi menggunakan alat kendali yang di produksi oleh pabrikan.

Dari permasalahan tersebut, penulis ingin melakukan penelitian dengan judul “ Rancang Bangun *Dc To Dc Boost Converter* Pada Robot KRSBI Beroda

JR–EVO Berbasis *Fuzzy Logic*” Dimana metode *fuzzy logic* di gunakan untuk mengontrol tegangan keluaran agar tetap sesuai dengan nilai yang diinginkan.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan diatas dapat ditarik beberapa permasalahan yaitu :

1. Bagaimana merancang *dc to dc boost converter* pada robot krsbi beroda JR-EVO berbasis *fuzzy logic*?
2. Bagaimana penerapan metode *fuzzy logic* pada *boost converter* ?
3. Bagaimana performa *boost converter* dengan metode *fuzzy* dan *boost converter* tanpa metode *fuzzy* pada Robot KRSBI Beroda ?

1.3 Batasan Masalah

Untuk memfokuskan tujuan penelitian dan meluasnya masalah, maka penulis memberi batasan masalah rencana penelitian ini. Adapun yang menjadi batasan penelitian ini adalah sebagai berikut, yaitu :

1. Motor yang di gunakan adalah motor dc dengan tipe PG 45.
2. Penelitian dilakukan menggunakan mikrokontroler Arduino UNO R3.
3. Pengendalian dilakukan pada rangkaian kontrol motor robot.
4. *DC to DC converter* yang di bahas hanya jenis *boost converter*.
5. Menggunakan baterai *lipo 3 cell* 11,1 Volt 5400 Mah.
6. Toleransi pada tegangan *setpoint* maksimal 10%

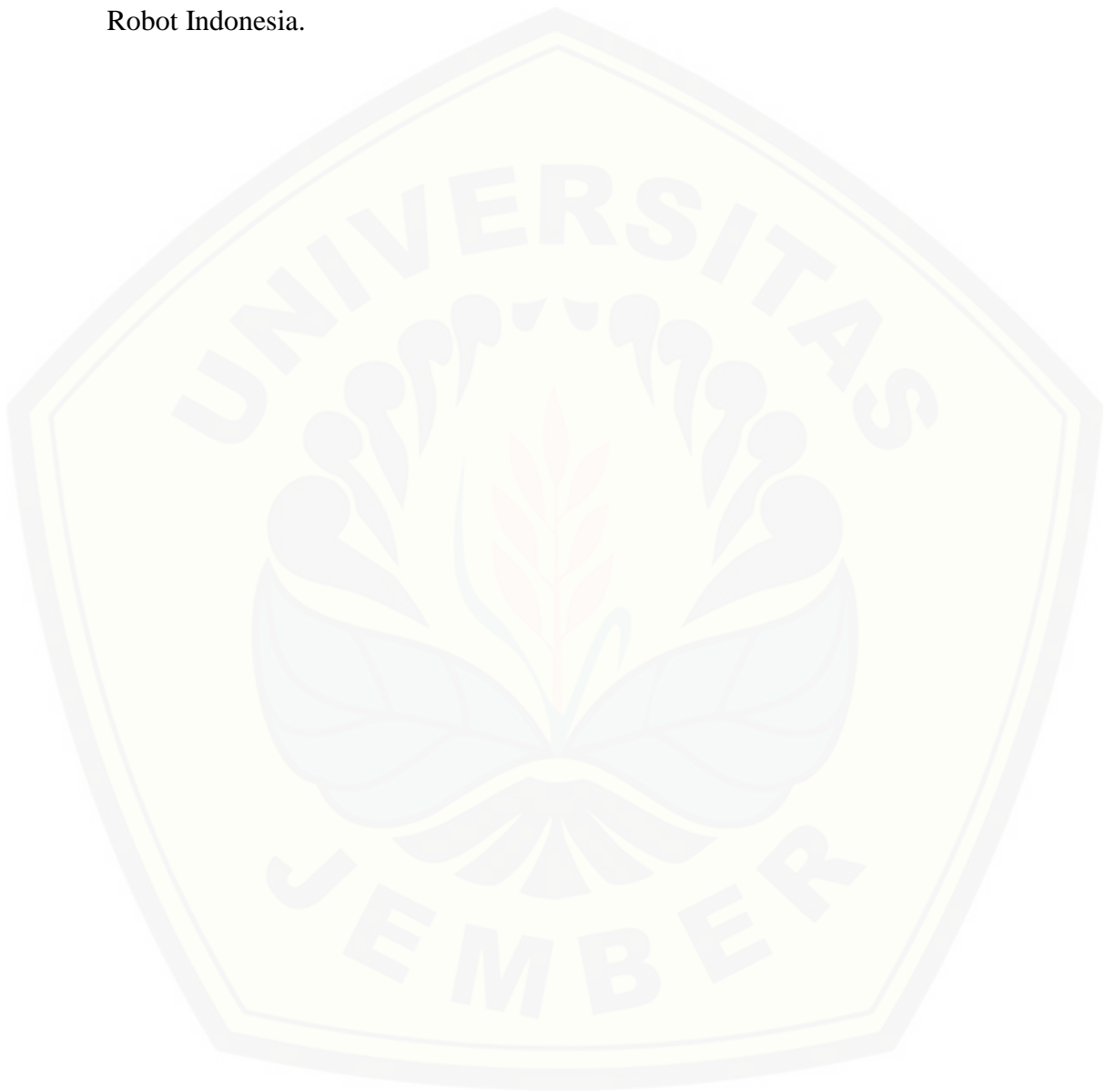
1.4 Tujuan Penelitian

Dengan meninjau latar belakang pada permasalahan yang telah diuraikan, maka dapat dirumuskan tujuan utama dari penelitian ini diantaranya :

1. Merancang *dc to dc boost converter* pada robot krsbi beroda JR–EVO berbasis *fuzzy logic*.
2. Membuat kontrol dengan metode *fuzzy logic* pada rangkaian *boost converter*.
3. Mengetahui Performa *boost converter* dengan metode *fuzzy* dan *boost converter* tanpa metode *fuzzy* pada Robot KRSBI Beroda.

1.5 Manfaat Penelitian

Dalam menyelesaikan tugas akhir ini di harapkan dapat memberi manfaat yang baik yaitu dapat mengetahui perancangan *boost converter* dengan metode *fuzzy logic* yang dapat mengoptimalkan kinerja robot KRSBI Beroda dalam Kontes Robot Indonesia.



BAB 2. TINJAUAN PUSTAKA

2.1 Gambaran Umum Robot KRSBI Beroda

Robot KRSBI Beroda memiliki dua sampai tiga pemain dalam satu tim. Dimana masing masing robot memiliki peranan tersendiri, seperti kiper, penyerang, dan pemain bertahan. Robot KRSBI Beroda memiliki mekanisme penggerak berupa motor DC yang jumlahnya di sesuaikan dengan kebutuhan setiap tim. Semua robot harus berkomunikasi dengan *base station* tim. *Base station* kedua tim harus terhubung dengan *referee box* sebagai pemandu dalam pertandingan. Semua robot bergerak dengan sistem semi otomatis maupun *full* otomatis. Secara sederhana, robot sepak bola beroda bergerak mengikuti arah dan mengejar bola. Ketika bola telah telah didapat, kemudian bola tersebut akan di arahkan dan ditendang ke gawang lawan sehingga menambah satu poin bagi tim yang berhasil mencetak gol.



Gambar 2.1 Robot JR-EVO Universitas jember KRI 2018.

(Sumber : Penulis)

2.1.1 Bagian Bagian Pada Robot KRSBI Beroda

Pada robot KRSBI Beroda memiliki beberapa bagian yang penting yang mendukung robot agar dapat menjalankan permainan dan dapat berkomunikasi dengan robot yang lain dalam satu tim sesuai dengan aturan yang telah di tetapkan. Beberapa bagian yang penting ini memiliki hubungan antara satu dengan bagian yang lain diantaranya yaitu :

1. Kamera *omnidirectional vision*

Omnidirectional vision merupakan sebuah kamera yang didesain secara khusus untuk robot *mobile* berjenis KRSBI Beroda. Dimana pada kamera ini memiliki sudut pandang yang luas yaitu 360 derajat yang diletakan pada bidang horizontal. Agar kamera dapat melihat objek bola dengan sudut pandang yang luas maka tangkapan objek pada kamera akan dipantulkan pada cermin terlebih dahulu. Kamera jenis ini menjadi bagian penting pada robot karena memiliki fungsi sebagai pencari objek bola pada lapangan. Agar dapat mengetahui arah objek maka data dari pembacaan objek pada kamera ini akan diolah dengan metode pengolahan citra. Sehingga menghasilkan keluaran data arah bola berupa sudut. dengan data yang diperoleh akan diproses oleh perintah tertentu pada program yang mengharuskan robot harus melakukan tindakan tertentu misalnya mendekati objek bola.



Gambar 2.2 Hasil Tangkapan Gambar Dari Omnidirectional Vision Pada Robot JR-EVO Universitas jember.

(Sumber : Penulis)

2. Motor DC PG 45

Secara umum motor DC adalah suatu mesin listrik yang memiliki fungsi mengubah energi listrik DC dan menjadikan energi gerak atau energi mekanik, dimana putaran rotor tersebut berupa energi gerak. Pengubahan energi listrik menjadi mekanik dipengaruhi oleh induksi magnet yang ada pada motor. Untuk

meningkatkan kekuatan putaran motor, maka motor DC di *couple* dengan *gearbox* yang memiliki rasio tertentu sehingga torsi motor semaik besar. *Gearbox* sendiri memiliki fungsi sebagai pemindah tenaga dari motor dc ke bagian motor lainnya seperti roda. Motor jenis ini dapat berkerja menggunakan arus berjenis DC (*Direct Current*) dengan masukan tegangan yang bervariasi. Tegangan juga mempengaruhi kecepatan putar motor. Semakin besar tegangan masukan maka semaikin cepat juga putaran motor. Yang perlu diperhatikan adalah kemampuan tegangan maksimal yang ada pada motor tersebut. Jika tegangan yang dimasukan lebih besar dari kapasitas maksimal tegangan motor, maka akan mengakibatkan kerusakan pada *brush* yang ada pada motor tersebut. Pada robot KRSBI Beroda menggunakan motor berjenis PG 45. Motor ini memiliki tegangan kerja maksimal 24 VDC. Dikarenakan memiliki kehandalan mulai dari bentuknya yang relatif kecil dan ringan, maka dari itu motor PG 45 banyak digunakan robot untuk kepentingan lomba ataupun keperluan industri.

Details	Geared motor specification	
	No Load Current (A)	< 1.500
No Load Speed (r. p. m)	468 ± 10%	
Rated Load Torque (kgf. cm)	15	
Rated Current (A)	< 6.500	
Rated Load Speed (r. p. m)	398 ± 10%	
Stall Current (A)		
Stall Torque (kgf. cm)		
Rotation Direction	CCW	

Gambar 2.3 Spesifikasi Motor DC PG 45.

(Sumber: Manufaktur Robot Indonesia)

3. *Driver* Motor DC

Driver motor pada umumnya merupakan bagian yang memiliki fungsi untuk menggerakkan motor DC. Perubahan arah ataupun kecepatan dari motor DC bergantung pada nilai tegangan yang diberikan dari *driver* tersebut. *Driver* motor juga memiliki peran sebagai perantara anatara program dan motor DC itu sendiri. Program pada umumnya memiliki perintah yang menginstruksikan kecepatan dan

arah motor. Dilihat dari arah putarnya ada dua jenis arah putar motor yaitu CW (*clock wise*) atau motor berputar searah jarum jam dan CCW (*counter clock wise*) atau motor berputar berlawanan jarum jam. Pada umumnya *driver* motor memiliki bagian yang dapat digunakan untuk mengontrol kecepatan dan arah putar motor DC, diantaranya yaitu :

4. Baterai

Baterai merupakan suatu perangkat yang digunakan untuk mengubah energi kimia menjadi energi listrik. Perubahan ini dilakukan dengan memanfaatkan prinsip transfer elektron dari satu material ke material lainnya melalui sirkuit elektrik. Baterai saat ini banyak digunakan sebagai sumber suplai tenaga listrik bagi perangkat-perangkat elektronik seperti robot. Dalam sebuah baterai terdapat sebuah proses perubahan dari kimia menjadi energi listrik (proses pengosongan baterai) dan sebaliknya yaitu proses perubahan dari energi listrik menjadi kimia (proses pengisian baterai). Penggunaan baterai pada robot umumnya menggunakan baterai jenis Li-Po (*Lithium polimer*). Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. (Oetomo, 2017)



Gambar 2.4 Baterai Li-Po Yang Digunakan Pada Robot KRSBI Beroda

(Sumber: Penulis)

Baterai jenis Li-Po memiliki rating 3,7V per sel nya. Keuntungannya adalah tegangan pada baterai yang tinggi dapat dicapai dengan menggunakan jumlah sel yang sedikit. Baterai Li-Po memiliki beberapa istilah yang umum digunakan untuk

menggambarkan spesifikasi dari baterai tersebut. Pada robot krsbi beroda menggunakan baterai jenis Li-Po dengan spesifikasi 11,1 V / 5400 mAh / 30C / 3S1P dengan artian sebagai berikut :

1. V (*Voltage*)

Dimana menunjukkan tegangan pada keluaran baterai. Dalam batereai Li-Po yang digunakan menunjukkan nilai 11,1V dengan artian 3 sel x 3,7 volt (3S).

2. mAh (*mili Ampere hours*)

Dimana menunjukkan kemampuan baterai dalam mensuplay arus dalam satu jam.

3. C (*discharge rating*)

Dimana menunjukkan nilai dari arus maksimal yang dapat dikeluarkan oleh baterai secara terus menerus. Semakin tinggi nilai nya maka yang dapat dikeluarkan juga semakin besar.

4. 3S1P (2 Sel 1 Pararel)

Dimana menunjukkan jumlah sel yang ada pada baterai dengan konfigurasi susunan sel (Pararel) dalam satu kesatuan baterai.

Dari spesifikasi diatas sehingga baterai dapat memberi tegangan 11,1V dan arus maksimal yang dapat dikeluarkan oleh baterai dengan kapasitas 5400 mAh serta memiliki *discharge rating* 30 C adalah 162 A.

2.2 Perangkat Pengendali

Pada penelitian yang akan dilakukan terdapat beberapa perangkat pengendali yang memiliki fungsi sebagai pengolah hasil pembacaan sensor, pengendali keluaran sesuai dengan masukan sensor dan tempat dimana proses dimana kontrol *Fuzzy Logic* berkerja. Dianata perangkat pengendali tersebut adalah :

2.2.1 Sensor Tegangan

Sensor tegangan adalah modul pembagi tegangan yang digunakan untuk mengubah tegangan yang akan diukur untuk dapat diumpangkan ke pin kontroler analog *microcontroller*.

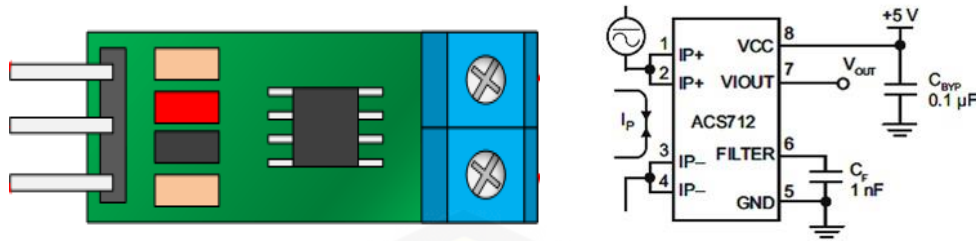


Gambar 2.5 Sensor Tengan Dengan Konsep Pembagi Tegangan

(Sumber : Penulis)

2.2.2 Sensor Arus ACS 712

Sensor Arus ACS712 adalah sebuah modul sensor arus yang dapat digunakan untuk mengukur arus AC atau DC. Modul sensor arus ACS712 telah dilengkapi dengan rangkaian penguat operasional, sehingga sensitivitas untuk pengukuran arusnya meningkat dan dapat mengukur perubahan arus yang relatif kecil. Sensor arus tipe ACS 712 memiliki kemampuan pengukuran arus maksimal sampai 30 A. Sensor jenis ini juga sering digunakan di bidang industri, komersial, maupun komunikasi. Contoh aplikasinya antara lain untuk sensor kontrol motor, deteksi dan manajemen penggunaan daya, sensor proteksi terhadap arus lebih, dan lain sebagainya.



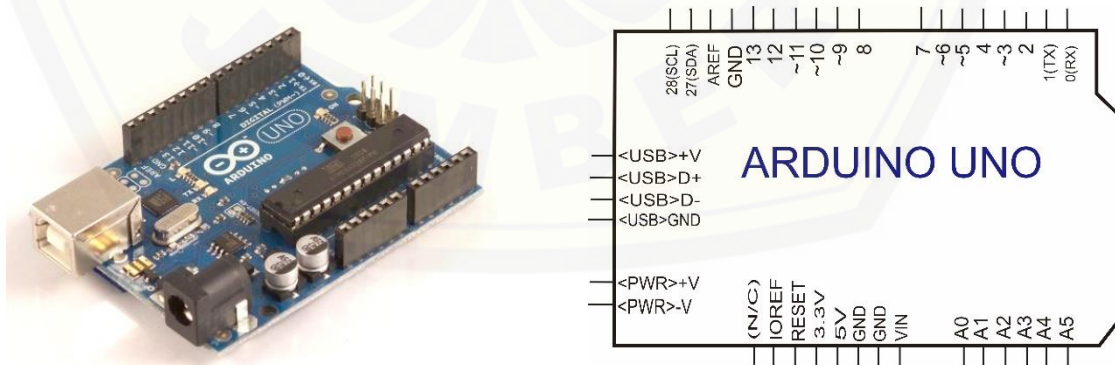
Gambar 2.6 Modul Sensor Arus Dan Skematik Rangkaian Dari ACS712

(Sumber : Henry's Bench)

2.2.3 Arduino UNO R3

Arduino Uno adalah modul mikrokontroler berbasis ATmega 328. Modul ini memiliki empat belas pin *input-output* (14 I/O) dimana 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 UNO 2015).



(a)

(b)

Gambar 2.7 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.

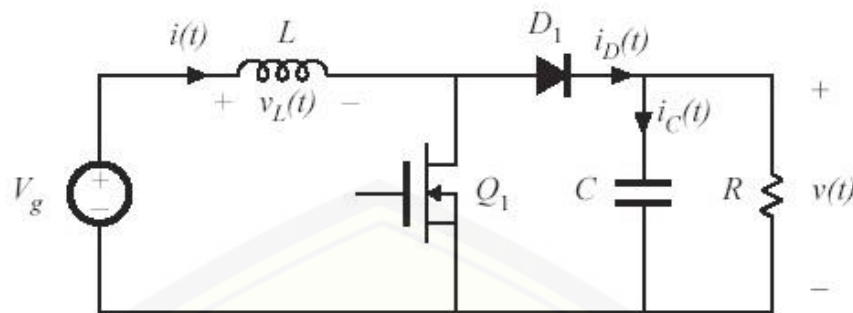
Tabel 2.1 Spesifikasi Arduino UNO-R3

Mikrokontroler	Atmega 328
Tegangan Operasi	5 V
Tegangan Input	7-12 V
Tegangan Input (Batas)	6-20 V
I/O Digital	14 (dengan tersedia 6 buah PWM output)
PWM digital I/O	6 buah
Input Analog	6 buah
Arus DC per I/O pin	40 Ma
Arus DC untuk pin 3.3	50 Ma
Flash Memory	32 KB (ATmega328P) Dengan 0.5 KB digunakan oleh Bootloader
SRAM	2 Kb (ATmega328p)
EEPROM	1 Kb (ATmega328p)
Kecepatan Clock	16 MHz

Diterjemahkan dari: (Arduino UNO)

2.2.4 Boost Converter

Boost Converter merupakan salah satu dari jenis converter DC-DC yang digunakan untuk menaikkan tegangan masukan. Naiknya tegangan keluaran dari boost converter ini ditentukan oleh adanya pengaturan switching pada MOSFET yang diatur berdasarkan besar kecilnya *duty cycle*. Pada pengaturan *duty cycle* ini menggunakan pemberian sinyal PWM (*Pulse Width Modulation*) yang di kontrol dengan bantuan dari sebuah mikrokontroler.



Gambar 2.8 Rangkaian Skematik *Boost Converter*

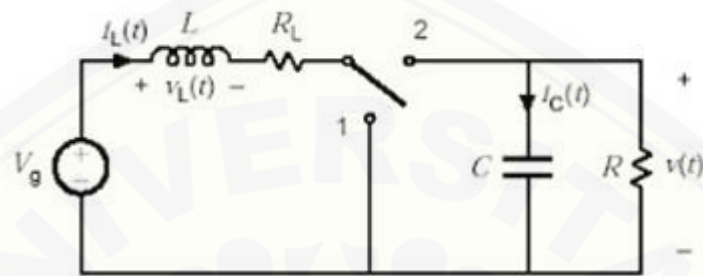
(Sumber : Aziz, 2018.)

Pada rangkaian *boost converter* terdapat komponen induktor, fungsi induktor tersebut adalah untuk menyimpan energi dalam bentuk arus, energi ini nantinya akan disalurkan ke beban. Tegangan pada beban adalah hasil dari tegangan masukan ditambah dengan energi yang tersimpan pada induktor, sehingga tegangan keluaran *boost converter* menjadi lebih besar dari pada tegangan masukannya (Umarella, 2012).

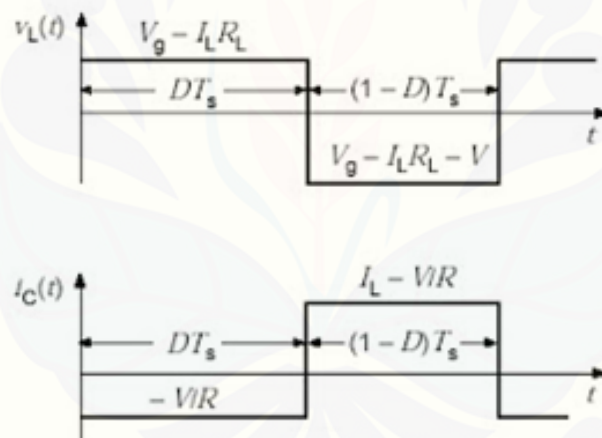
Skema *boost converter* dapat dilihat pada gambar 2.8, pada komponen utama *boost converter* terdiri atas MOSFET, dioda, induktor, dan kapasitor. Pada saat MOSFET dalam kondisi tertutup, maka arus akan mengalir ke induktor sehingga menyebabkan energi yang tersimpan di induktor naik. Kemudian saklar MOSFET terbuka, maka arus pada induktor akan mengalir menuju beban melewati dioda sehingga energi yang tersimpan di induktor akan turun. Rasio antara tegangan keluaran dan tegangan masukan konverter sebanding dengan rasio antara periode pensaklaran dan waktu pembukaan saklar, dan untuk keunggulan dari konverter *boost* adalah mampu menghasilkan arus masukan yang kontinyu.

2.2.4.1 Prinsip Kerja *Boost Converter*

MOSFET yang digunakan pada rangkaian DC *Chopper Tipe Boost* adalah bertindak sebagai saklar yang dapat membuka atau menutup rangkaian sehingga arus dapat dikendalikan sesuai dengan *duty cycle* yang diinginkan. Berikut adalah skema secara umum dari DC *Chopper Tipe Boost*.



(a)

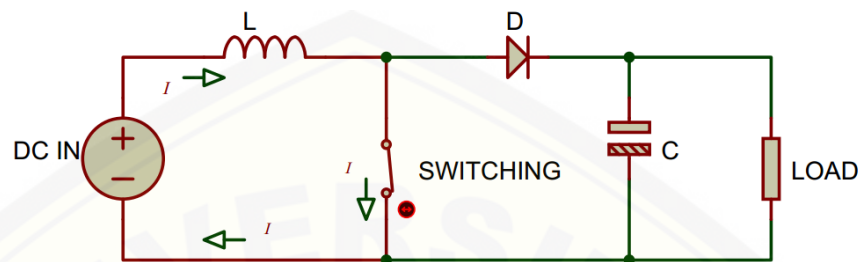


(b)

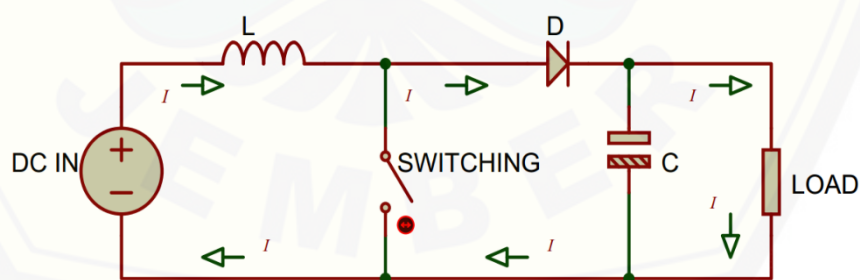
Gambar 2.9 (a) Skema *Boost Converter* (b) gelombang keluaran *boost converter*

Prinsip kerja dari *boost converter* dapat dibagi menjadi 2 kondisi yaitu :

1. Ketika MOSFET *on* (tertutup) sedangkan dioda *off*, arus mengalir searah jarum jam dari sumber menuju ke induktor (terjadi pengisian arus pada induktor). Polaritas induktor pada sisi kiri lebih positif dibandingkan sisi kanannya.

Gambar 2.10 MOSFET *Boost Converter* ON

2. Ketika MOSFET *off* (terbuka) dan dioda *on*, arus yang disimpan di induktor akan berkurang karena impedansi yang lebih tinggi. Berkurangnya arus pada induktor menyebabkan induktor tersebut melawannya dengan membalik polaritasnya (lebih negatif pada sisi kiri). Sehingga, arus yang mengalir pada dioda dan pada beban adalah penjumlahan antara arus pada sumber dan arus pada induktor (seri). Disaat yang bersamaan kapasitor juga akan melakukan penyimpanan energi dalam bentuk tegangan. Itulah sebabnya *boost converter* memiliki keluaran yang lebih tinggi dibandingkan dengan masukannya.

Gambar 2.11 MOSFET *Boost Converter* OFF

2.2.4.2 Parameter perancangan *boost converter*

Empat parameter yang diperlukan untuk menghitung daya pada *boost converter* diantaranya yaitu :

1. Rentang Tegangan Input: $V_{in(min)}$ dan $V_{in(max)}$
2. Nominal Tegangan Output: V_{out}
3. Output Arus Max : I_{out}
4. Rangkaian yang terintegrasi yang digunakan untuk membangun boost converter. Hal ini diperlukan, karena beberapa parameter untuk perhitungan harus diambil dari data sheet dari komponen yang digunakan dalam perancangan.

Langkah-langkah Perancangan *Boost Converter*

1. Penentuan *Duty Cycle*

$$D = 1 - \frac{V_{in(min)} \times \eta}{V_{out}} \quad (2.1)$$

Keterangan :

- D : *Duty Cycle*
 $V_{in(min)}$: Tegangan masukan minimum
 V_{out} : Tegangan Keluaran
 η : Efisiensi dari boost converter

2. Menghitung Estimasi Arus Ripple Induktor

$$\Delta I_L = 0.2 \text{ s. d } 0.4 \times I_L \quad (2.2)$$

Keterangan :

- ΔI_L : Arus ripple induktor
 I_L : Arus Induktor

3. Menghitung Arus Induktor

$$I_L = \frac{I_o}{1 - D_{max}} \quad (2.3)$$

Keterangan :

- I_o : Arus keluaran Maksimum
 D_{max} : *Duty Cycle* Maksimum

4. Menentukan Besarnya Induktansi

$$L = \frac{V_{IN} \times (V_{OUT} - V_{IN})}{\Delta I_L \times f_s \times V_{OUT}} \quad (2.4)$$

Keterangan :

f_s : Frekuensi *switching*

5. Menentukan Tegangan Ripple Kapasitor

$$\Delta V_C = 0,2 \text{ sd } 0,4 \times V_O \quad (2.5)$$

6. Menentukan Besarnya Kapasitansi

$$C_{OUT(min)} = \frac{I_O(V_O \times V_{in})}{f_s V_O \Delta V_C} \quad (2.6)$$

Keterangan :

$C_{OUT(min)}$: kapasitansi minimum

ΔV_C : ripple tegangan keluaran

7. Menentukan Arus *switching* MOSFET

$$I_{SMmax} = \frac{I_{Omax}}{1 - D_{min}} + \frac{V_O D_{max}(1 - D_{min})}{2 f_s L} \quad (2.7)$$

Keterangan :

I_{SMmax} : Arus *switching* MOSFET

I_{Omax} : Arus Keluaran Maksimal

D_{max} : *Duty Cycle Maksimal*

8. Menentukan besarnya resistansi yang digunakan sebagai beban

$$R = \frac{V_o}{I_o} \quad (2.8)$$

Keterangan :

R : Resistansi

2.2.5 LCD (*Liquid Crystal Display*)

Liquid Cristal Display adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari *back-lit* LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.

LCD terdiri atas beberapa lapisan diantaranya yaitu lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan seven-segment dan lapisan elektroda pada bagian kaca belakang. Pada saat elektroda diaktifkan dengan memberikan medan listrik (tegangan), maka molekul organik yang panjang dan silindris akan menyesuaikan diri dengan elektroda dari segmen. Lapisan sandwich memiliki polarizer cahaya vertikal depan dan polarizer cahaya horisontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.



Gambar 2.12 LCD (*Liquid Crystal Display*)

(Sumber : Aziz, 2018.)

Konfigurasi pin untuk LCD diantaranya yaitu :

1. Pin data merupakan jalur untuk memberikan data karakter yang akan ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika low menunjukkan yang masuk adalah perintah, sedangkan logika high menunjukkan data.
3. Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika low tulis data, sedangkan high baca data.
4. Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
9. Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

2.3 Kontrol Fuzzy Logic

2.3.1 Pengertian Fuzzy Logic Controller

Dalam bahasa inggris, *fuzzy* mempunyai arti kabur atau tidak jelas. Jadi, *fuzzy logic* adalah logika yang kabur, atau mengandung unsur ketidak pastian. 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.

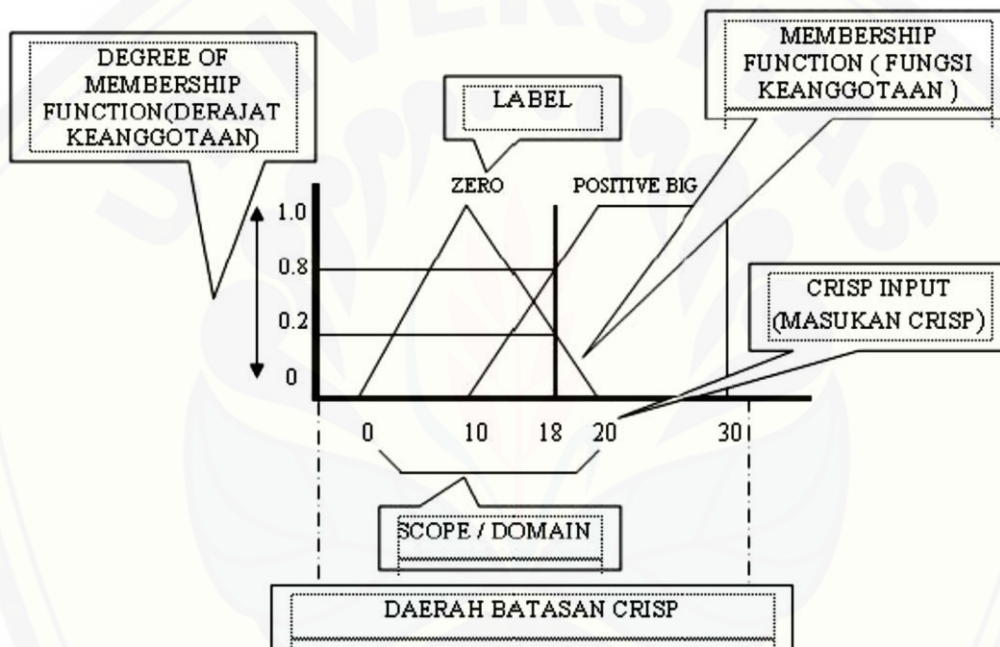
Kontroler logika fuzzy dikategorikan dalam kontrol cerdas (*intelligent control*). Unit logika fuzzy memiliki kemampuan menyelesaikan masalah perilaku sistem yang kompleks, yang tidak dimiliki oleh kontroler konvensional. Secara umum kontroler *fuzzy logika* memiliki kemampuan sebagai berikut:

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 (Chak, 1998).



Gambar 2.13 Konsep Dasar Logika Fuzzy

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 (Sudradjat, 2007).

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:

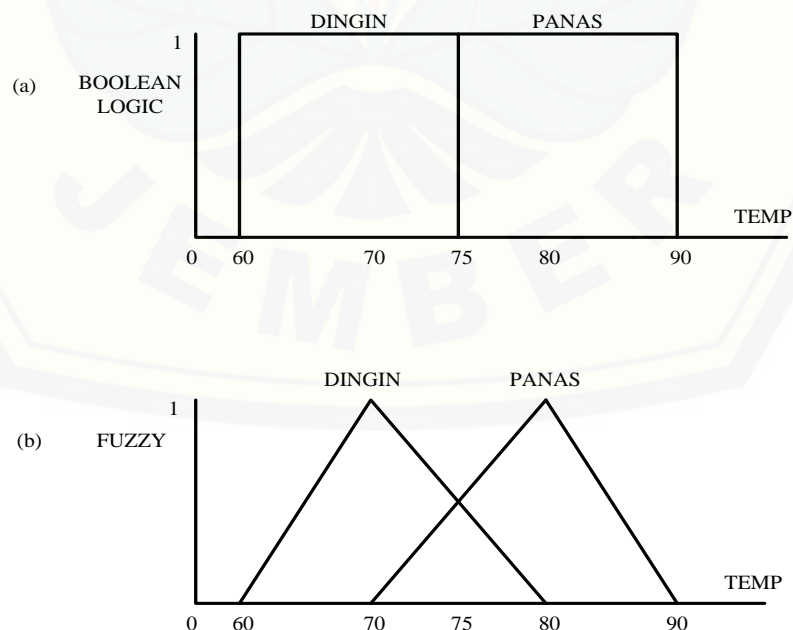
1. Konsep *fuzzy logic* mudah dimengerti
2. *Fuzzy logic* sangat fleksibel
3. *Fuzzy logic* memiliki toleransi terhadap data yang kurang tepat
4. *Fuzzy logic* mampu memodelkan fungsi nonlinier yang kompleks
5. *Fuzzy logic* didasari pada ahasa alami

Fuzzy logic saat ini banyak diterapkan dalam berbagai bidang diantaranya:

1. *Fuzzy rule Based Systems*
2. *Fuzzy Nonlinear Simulations*
3. *Fuzzy Decision Making*
4. *Fuzzy Classification*
5. *Fuzzy Pattern ecognition*
6. *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 2.14 Perbedaan Boolean Logic (a) dengan Fuzzy Logic (b).

2.3.2 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 S, Purnomo H, 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 = $[30 \ 40]$

2.3.3 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 (Wang, 1997 dari Wulandari, F., 2005). 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

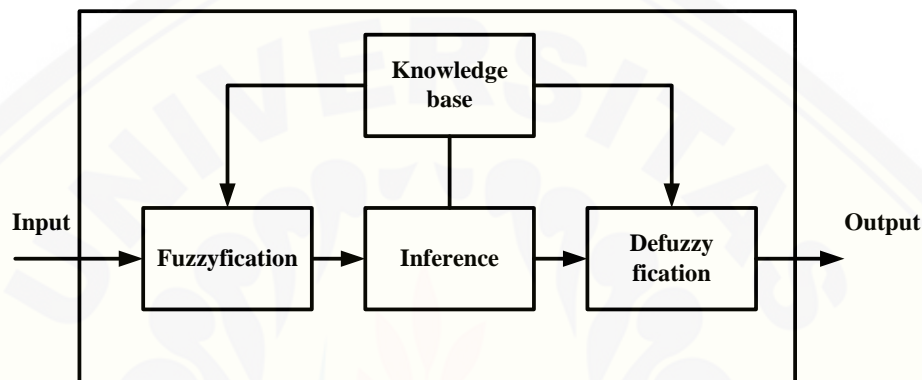
THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator fuzzy, seperti:

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

Dengan • adalah operator (misal: OR, AND atau NOT). Min (*minimum*).

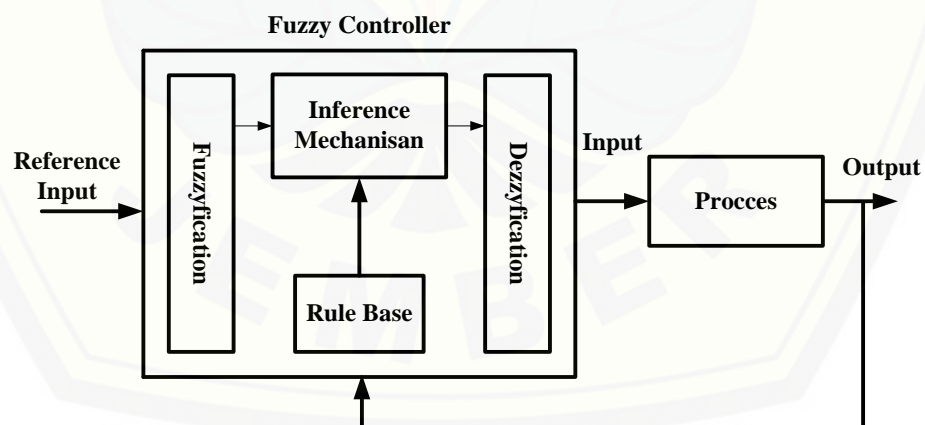
Fungsi ini akan memotong *output* himpunan fuzzy.

2.3.4 Tahap Pemodelan Dalam Fuzzy Logic



Gambar 2.15 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 2.16 Blok diagram sistem fuzzy 2

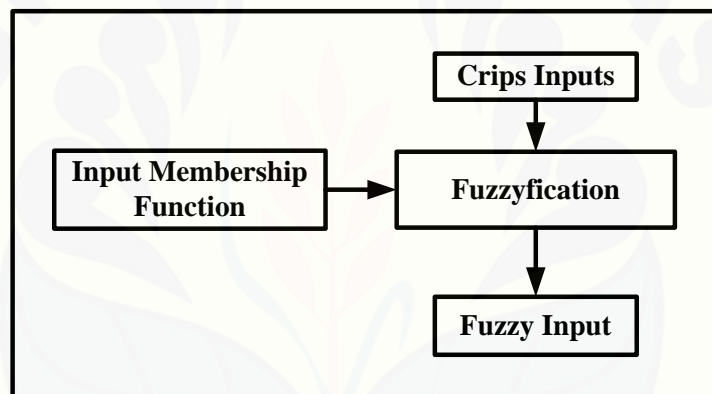
2.3.5 Fuzzification

Fuzzifikasi 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 2.17 *Fuzzification*

2.3.6 Rule Evaluation

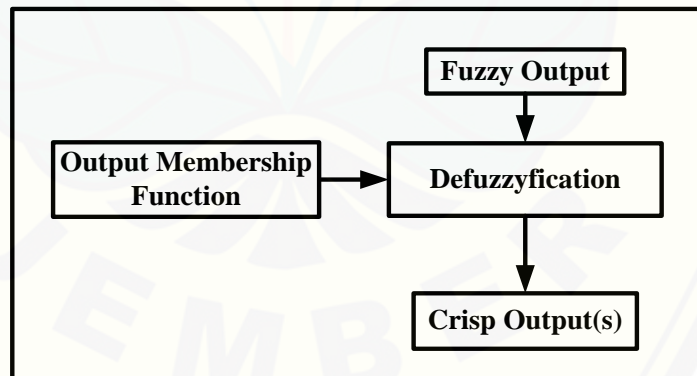
Prosedur fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel *numerik*) menjadi variabel fuzzy (variabel *linguistik*). Nilai *error* dan *delta error* yang dikuantisasi sebelumnya diolah oleh kontroler logika fuzzy, kemudian diubah terlebih dahulu ke dalam variabel fuzzy. Melalui *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

2.3.7 Defuzzification

Defuzzifikasi dapat didefinisikan sebagai proses perubahan 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 2.18 *Defuzzification*

BAB 3. METODELOGI PENELITIAN

3.1 Metode Penelitian

Pada proses penelitian tugas akhir ini ada beberapa tahapan yang akan dilakukan, berikut adalah langkah-langkah dalam penelitian ini yaitu :

a. Studi Literatur

Tahap awal pelaksanaan penelitian ini adalah dengan mencari berbagai macam literatur yang sesuai. Literatur dapat diperoleh dari hasil penelitian sebelumnya melalui buku, internet maupun *e-book* untuk mengetahui karakteristik komponen sistem, prinsip kerja serta teori yang menunjang lainnya. Diharapkan dengan literatur yang telah didapat dapat memberikan arahan untuk mengurangi kesalahan dalam penelitian.

b. Perumusan Masalah

Tahapan kedua adalah dengan melakukan perumusan masalah tentang apa saja yang ada pada robot KRSBI Beroda.

c. Perancangan DC-DC *Boost Converter* Dengan Metode *Fuzzy Logic*

Tahapan yang ketiga adalah proses perancangan DC-DC *Boost Converter* dengan metode yang digunakan adalah *Fuzzy Logic*. Dimana pada tahap ini dilakukan perancangan beberapa komponen pengendali seperti sensor arus dan sensor tegangan, serta komponen penyusun DC-DC *boost converter* lainnya.

d. Pembuatan DC-DC *Boost Converter* Dengan Metode *Fuzzy Logic*

Pada tahap ini, yaitu pembuatan *DC converter* dengan tipe *Boost Converter* dengan memasang semua komponen seperti kapasitor, dioda, Mosfet, dan induktor sesuai dengan *layout* yang telah dibuat. Pembuatan programan pada arduino untuk mendapatkan metode *fuzzy* dengan *set point* yang ditentukan agar tegangan yang akan masuk dari baterai Li-Po menjadi sesuai dengan tegangan yang di butuhkan pada motor DC PG 45 dengan *set Point* 24 Volt.

e. 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 dengan memberi tegangan masukan dari baterai Li-Po 11,1 Volt ke tanganan

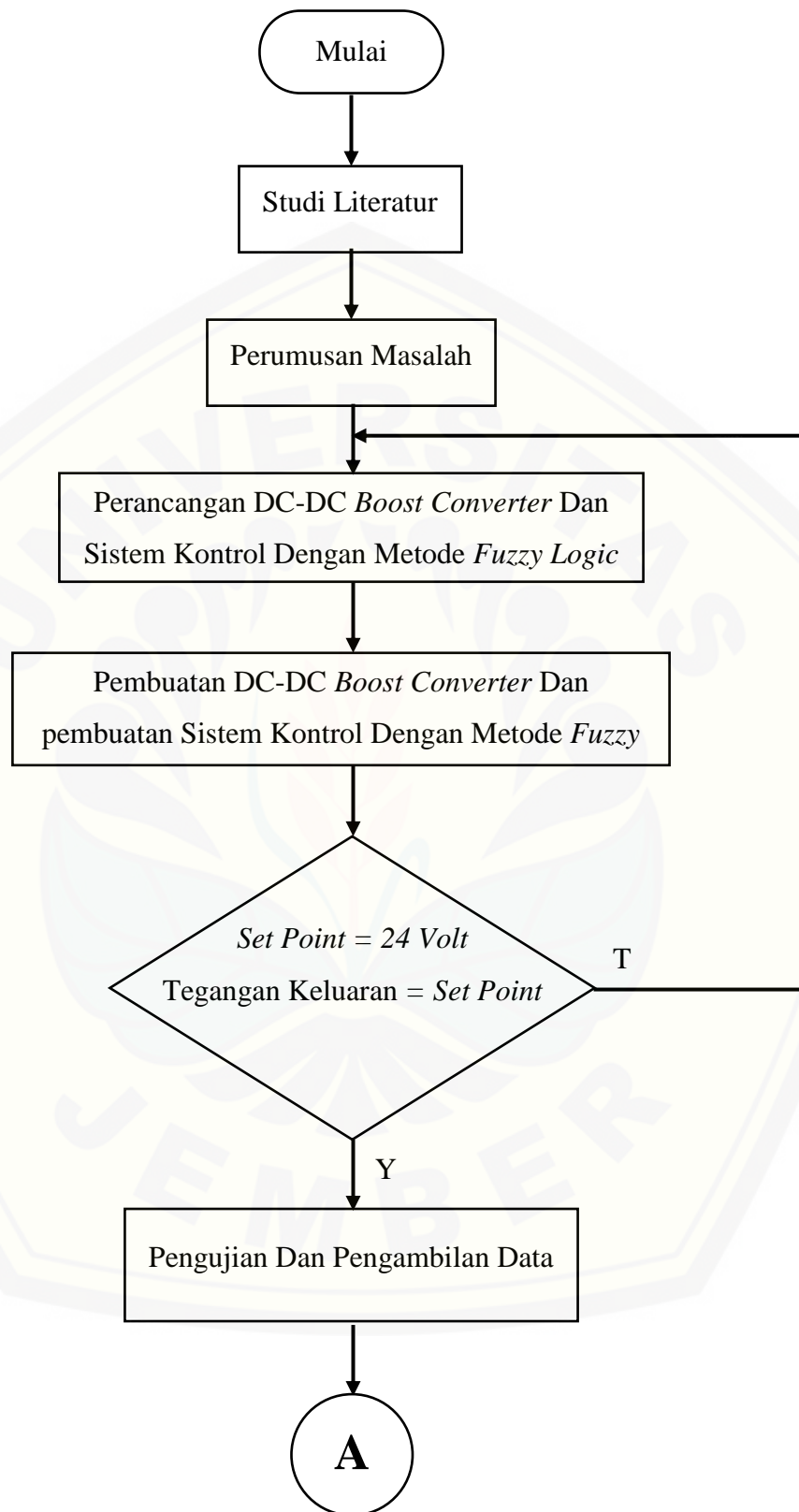
yang di butuhkan motor yaitu 24 volt. Pengujian dan kalibrasi sensor juga di lakukan agar mendapatkan pembacaan sensor akurat sesuai dengan alat ukur. Setelah melakukan pengujian pada keseluruhan sistem dan memastikan bekerja dengan baik dan hasilnya memenuhi target, maka yang dilakukan selanjutnya adalah pengambilan data yang diperlukan untuk kemudian dianalisa dari data yang telah didapatkan.

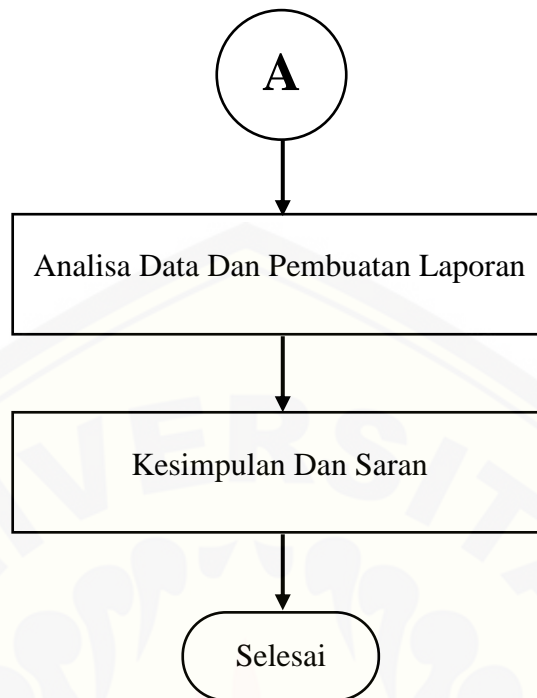
f. Analisa dan Pembuatan Laporan

Pada tahap ini, Langkah selanjutnya adalah melakukan analisis terhadap data yang telah didapat. Analisis tersebut berfungsi untuk mengetahui tegangan keluaran dari DC-DC *boost converter* setelah menggunakan sistem kontrol dengan metode *Fuzzy Logic*. Setelah melakukan analisa pada data, selanjutnya membuatnya menjadi sebuah bentuk laporan hasil dari penelitian yang telah dilakukan.

g. 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.

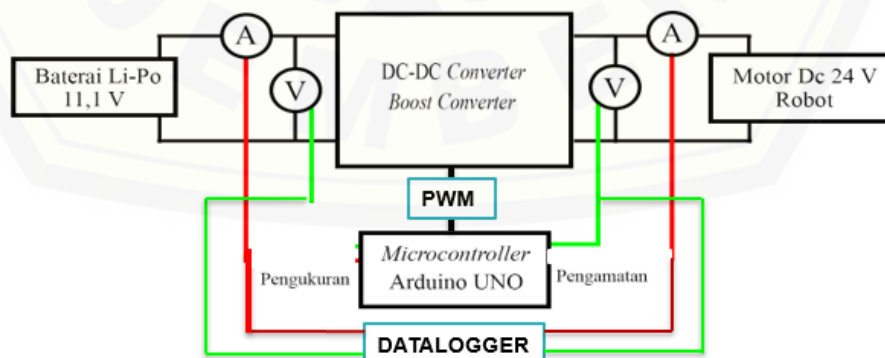




Gambar 3.1 *Flowchart* Penelitian
(Sumber : Penulis)

3.2 Diagram Blok Sistem

Untuk mendapatkan data yang diinginkan dan sesuai dengan tujuan awal yang sesuai dengan topik yang diambil, maka sistem akan di pisahkan berdasarkan fungsinya. Berikut ini adalah blok diagram dari rancang bangun *Dc-DC Boost Converter* Pada Robot KRSBI Beroda JR-EVO Berbasis *Fuzzy Logic* seperti pada gambar 7.2.



Gambar 3.2 Gambar Diagram Blok Sistem
(Sumber : Penulis)

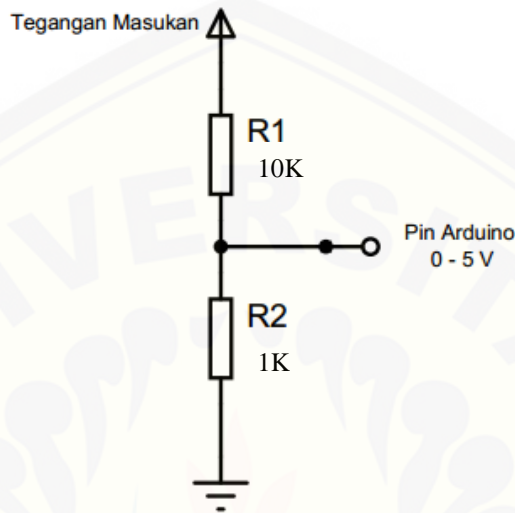
Pada dasarnya prinsip kerja dari DC-DC *Boost Converter* Pada Robot KRSBI Beroda JR-EVO Berbasis *Fuzzy Logic* sama seperti rangkaian DC-DC *boost converter* pada umumnya dan yang membedakan adalah pengaturan tegangan output DC-DC *boost converter* dapat dilakukan secara otomatis menyesuaikan dengan beban dan tegangan masukan pada DC-DC *boost converter* sehingga tegangan keluaran tetap pada *set point* yaitu 24 Volt dengan toleransi 10%. Selain itu DC-DC *boost converter* memiliki kapasitas arus yang dihasilkan bisa mencapai 7 Ampere. Untuk tegangan masukan pada DC-DC *boost converter* menggunakan baterai Li-Po 3 sel dengan tegangan keluaran sebesar 11,1 Volt. Sedangkan untuk menggerakkan motor pada robot KRSBI Beroda JR-EVO membutuhkan Tegangan 24 V . Maka dari itu diperlukan sebuah Dc-Dc Converter *boost converter* untuk menaikkan tegangannya agar dapat menggerakkan motor pada robot. Pengaturan tegangan keluaran pada DC-DC *boost converter* adalah dengan mengatur besarnya PWM, sedangkan PWM dipengaruhi oleh nilai *duty cycle* yang diatur oleh sebuah *microcontroller* berupa arduino uno R3. Didalam program yang telah di buat pada Arduino, terdapat kontrol yang mengatasi apabila terjadi drop tegangan sehingga mempertahankan tegangan keluaran agar selalu berada pada set poin. Sensor tegangan digunakan dengan tujuan untuk mengetahui tegangan keluaran DC-DC *boost converter* dan juga digunakan sebagai masukan dari Arduino dimana berupa nilai ADC. Sebagai tambahan maka tegangan keluaran dari DC-DC *boost converter* akan di kontrol dengan metode *Fuzzy Logic* untuk mendapatkan kestabilan tegangan yang akan masuk kedalam rangkain *driver* motor.

3.3 Perancangan Komponen Pengendali (*Hardware*)

3.3.1 Sensor Tegangan

Sensor tegangan yang digunakan mengadopsi konsep dari rangkaian pembagi tegangan. Pada Perancangannya dilakukan dengan cara menyusun resistor yang digunakan secara seri. Dikarenakan pada pin arduino hanya mampu membaca tegangan maksimal 5 V sedangkan tegangan yang diukur adalah 24 V, maka tegangan 24 V tersebut harus dibagi agar bisa mendekati 5V. Rangkaian pembagi tegangan ini menggunakan resistor 10 k Ω dan resistor 1 K Ω . Dari pembagian

tegangan tersebut didapatkan sensor tegangan yang dapat membaca tegangan dari 0 – 55 V dan perbandingan tegangan untuk diberikan ke arduino uno R3 0 – 5 V. Skema rangkaian sesnsor tegangan dapat dilihat pada gambar 7.3.



Gambar 3.3 Rangkaian Sensor Tegangan
(Sumber : Penulis)

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

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

dimana, $V_{out} = 5$; $V_{in} = 55$ volt; $R_1 = 100 \text{ k}\Omega$

$$5 = \frac{55 \times R_2}{(10k + R_2)}$$

$$5 \times (10k + R_2) = 55 \times R_2$$

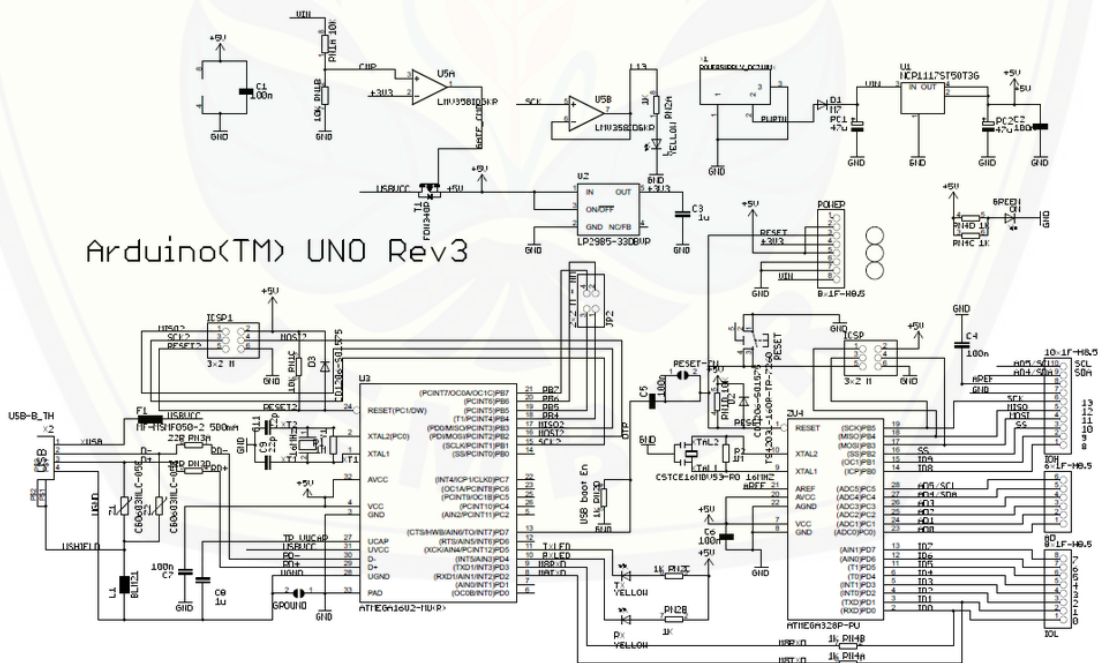
$$50k + 5R_2 = 55R_2$$

$$R_2 = \frac{55k}{55}$$

$$R_2 = 1 \text{ K}\Omega$$

3.3.2 Arduino UNO R3

Arduino UNO R3 Sebagai kontroler yang digunakan dalam *plant*. Pada tahaan penjelasan tentang arduino akan menjadi lebih spesifik, terutama dalam penggunaannya yaitu sebagai *embedded system*. Arduino yang digunakan yaitu dengan tipe arduino uno R3 dimana *embedded system* ini yang menerima *feedback* tegangan dari *output* tegangan DC-DC *boost converter*. Pembacaan nilai *output* tegangan DC-DC *boost converter* 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 *boost 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). 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 proses pengecekan ulang, seperti penerimaan pembacaan ADC oleh sensor.



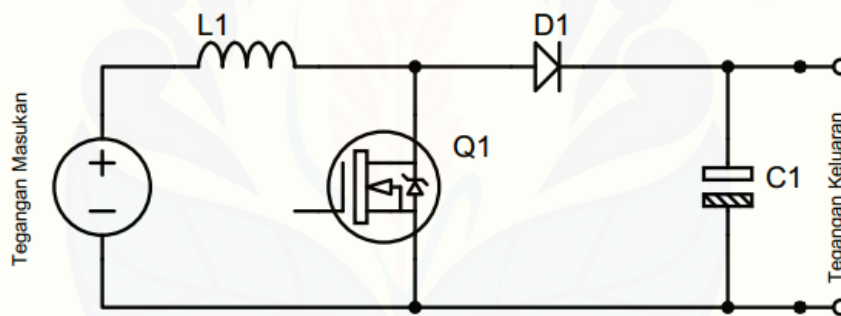
Gambar 3.4 Skematik Arduino Uno R3
(Sumber : arduino.com)

3.3.3 Perancangan DC-DC Boost Converter

Untuk merancang sebuah DC-DC *boost converter* dibutuhkan perhitungan untuk mencari nilai dari *duty cycle*, *filter*, maupun bebannya. Maka digunakan langkah-langkah seperti yang tertera dibawah ini,sesuai dengan parameter yang di dapat :

Tabel 3.2 Parameter DC-DC *Converter Boost Converter*

Tegangan Masukan	11,1 – 12,6 V
Tegangan Keluaran	24 V
Arus Keluaran	0,1 – 5 A
Frekuensi	62 KHz



Gambar 3.5 Desain DC – DC *Boost Converter*

Sebelum mendesain DC-DC *boost converter*, terlebih dahulu menentukan spesifikasi dari *converter* yang akan digunakan. spesifikasi *converter* yang digunakan sesuai dengan tabel 3.2. kemudian langkah selanjutnya yaitu melakukan perancangan ukuran masing masing komponen.

Berdasarkan spesifikasi yang ditentukan, untuk besar tegangan sampel diambil tegangan sebesar 11,8 V. langkah selanjutnya yaitu merancang komponen untuk membuat DC-DC *boost converter*, berikut ini Langkah – langkah menentukan nilai masing-masing komponen melalui perhitungan:

1. Perhitungan *duty cycle*

Menentukan nilai dari *duty cycle* maksimal dan minimal pada sistem *switching* dengan estimasi efisiensi Dc-Dc *boost converter* adalah 85 %.

$$D_{min} = 1 - \frac{V_{IN(max)} \times \eta}{V_{OUT}} \quad (3.2)$$

$$D_{min} = 1 - \frac{12,6 \times 0,85}{24} = 0,44625 = 45 \%$$

$$D_{max} = 1 - \frac{V_{IN(min)} \times \eta}{V_{OUT}} \quad (3.3)$$

$$D_{max} = 1 - \frac{11,1 \times 0,85}{24} = 0,393125 = 39 \%$$

2. Perhitungan Induktor

Setelah *duty cycle* ditemukan selanjutnya mencari besarnya induktansi, besarnya induktansi dapat diperoleh dengan menggunakan persamaan dibawah :

a. Arus Induktor

Untuk menentukan besarnya induktor yang akan digunakan, perlu diketahui terlebih dahulu berapa besar arus yang mengalir pada induktor dengan persamaan dibawah ini .

$$I_L = \frac{I_O}{1 - D_{max}} \quad (3.4)$$

$$I_L = \frac{7}{1 - 0,606875} = 17,80604134 \text{ A}$$

b. Arus Ripple Induktor

Untuk membuat Dc-Dc *boost converter* yang memiliki respon terhadap beban yang baik, maka ditentukan riak arus induktor adalah 3 % dari arus induktor.

$$\Delta i_L = 0,03 \times I_L \quad (3.5)$$

$$\begin{aligned} \Delta i_L &= 0,03 \times 17,80604134 \\ &= 0,53418124 \text{ A} \end{aligned}$$

c. Nilai Induktansi

Setelah mengetahui berapa besar arus yang mengalir pada induktor sehingga diperoleh arus ripple yang telah ditentukan, kemudian menghitung nilai induktor yang digunakan untuk membuat Dc-Dc *boost converter* dengan persamaan di bawah ini :

$$\begin{aligned}
 L &= \frac{V_{In}(V_{Out}-V_{In})}{\Delta i_L \times f_s \times V_{Out}} \\
 &= \frac{11,1(24 - 11,1)}{0,53418124 \times 30000 \times 24} \\
 &= 0,000372298 \text{ H} \\
 &= 372,2987352 \mu\text{H}
 \end{aligned} \tag{3.5}$$

Setelah nilai dari induktor, arus ripple dan nilai induktansi pada induktor telah ditemukan, langkah selanjutnya yaitu menghitung jumlah lilitan, pada penelitian ini jenis induktor yang digunakan digunakan yaitu torroidal core inductor atau inti yang berbentuk O Ring (bentuk donat) dengan tipe T-184-1 buatan micromental dengan nilai AL = 500uH/100 turn. Untuk ukuran enamel yang digunakan dalam penelitian sebesar 10 AWG atau sebesar 2,5882 mm (berdasarkan datasheet).

$$N = 100 \sqrt{\frac{\text{Nilai } L \text{ (uH)}}{AL \left(\frac{\text{uH}}{100} \text{turn}\right)}} \tag{3.6}$$

$$N = 100 \sqrt{\frac{372,2987352}{500}}$$

$$N = 86 \text{ lilitan}$$

3. Pemilihan MOSFET

a. Arus *switching* MOSFET

$$\begin{aligned}
 I_{SMmax} &= \frac{I_{Omax}}{1-D_{min}} + \frac{V_O D_{max}(1-D_{min})}{2f_s L} \\
 &= \frac{7}{1-0,55375} + \frac{24 \times 0,606875(1-0,55375)}{2 \times 30000 \times 0,000372298} = 15,97724354 \text{ A}
 \end{aligned} \tag{3.7}$$

b. Arus rms

$$I_{Srms} = \frac{\sqrt{D_{max}} I_{Omax}}{1-D_{max}} = \frac{\sqrt{0,606875} \times 7}{1-0,606875} = 13,87129477 \text{ A} \tag{3.8}$$

Dengan data seperti yang ditunjukkan dalam perhitungan yang tertera diatas dan menyesuaikan dengan tipe MOSFET yang ada di pasaran maka digunakan MOSFET N-Chanel dengan seri IRF 540 dengan spesifikasi dasar MOSFET sebagai berikut VDS = 100 Volt, ID = 28 Ampere, tegangan pada Gate (VGS)

mosfet dapat bervariasi ± 20 Volt dengan arus pada Gate (IGSS) sebesar 100 Nano Ampere.

4. Perhitungan Kapasitor

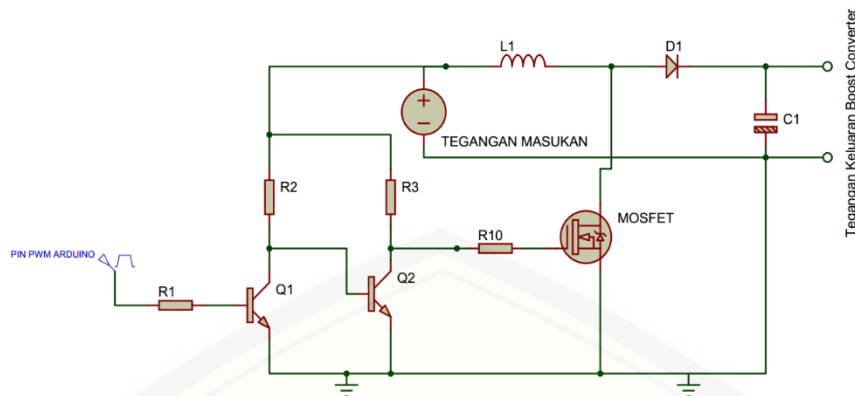
a. Tegangan *ripple*

Sebelum menentukan nilai kapasitansi kapasitor terlebih dahulu menentukan nilai dari tegangan *ripple* sebesar 3%.

$$\begin{aligned}\Delta V_C &= 3\% \times V_O \\ &= 0.03 \times 24 \\ &= 0.72 \text{ V}\end{aligned}\tag{3.9}$$

Sehingga didapatkan nilai dari kapasitor minimum yang akan digunakan pada *boost converter*

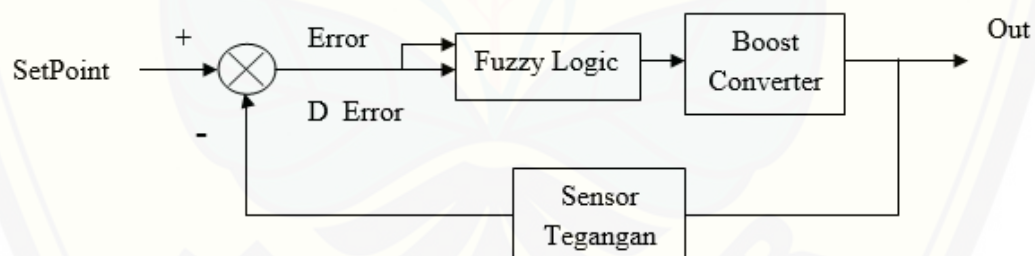
$$\begin{aligned}C_{out} &= \frac{I_O(V_O - V_{in})}{f_s V_O \Delta V_C} \\ &= \frac{7(24 - 11,1)}{30000 \times 24 \times 0,72} \\ &= 0.000174189 \text{ F} \\ &= 174,2 \mu\text{F}\end{aligned}\tag{3.9}$$



Gambar 3.6 Rangkaian Dc-Dc *boost converter* Dengan Driver Mosfet
(Sumber : Penulis)

3.4 Perancangan Kendali *Fuzzy*

Perancangan kontrol *fuzzy* untuk Dc-Dc *boost converter* dirancang secara sistematis sehingga logika *fuzzy* ini dapat diimplementasikan secara langsung ke dalam sebuah *embedded system* yang berupa *microcontroller* arduino UNO R3. Ada beberapa tahapan yang akan dilakukan untuk merancang kontrol *fuzzy*, yaitu dimulai dari tahap *fuzzyfikasi*, *inferent* dan *defuzzifikasi*.



Gambar 3.7 Diagram Blok Sistem Kontrol Dc-Dc *boost converter*
(Sumber : Penulis)

Gambar diatas merupakan diagram blok sistem DC-DC *boost converter* yang menggunakan kontrol dengan metode *fuzzy*. *Mikrokontroller* akan membaca nilai adc tegangan dari beban yang berupa motor penggerak robot dan kemudian akan dihitung nilai *error* dan *delta error* dari nilai *volt reff* yang sudah ditentukan. Output dari kontrol *fuzzy* berupa PWM yang masuk ke DC-DC *boost converter*. Sehingga memungkinkan ketika robot sudah berjalan sesuai dengan plan maka otomatis tegangan dari baterai akan drop, namun karena adanya kontrol dengan

metode fuzzy yang memungkinkan tegangan keluaran DC-DC *boost converter* tetap sesuai dengan *set point* yang telah di tentukan yaitu 24 V.

Perancangan pengendali *Fuzzy* meliputi beberapa tahap yaitu fuzzifikasi, penentuan *rule base*, *defuzzifikasi*. Pada pengujian respon sistem keluaran pengendali *Fuzzy Inference System* berupa penambahan atau pengurangan nilai PWM yang mana akan mempengaruhi nilai *duty cycle* pada rangkaian. *Duty cycle* sendiri merupakan proporsi waktu dimana sistem ketika beroperasi. Pengaturan nilai *duty cycle* mempengaruhi besar kecilnya tegangan keluaran yang dihasilkan oleh *boost converter*.

Sensor kecepatan sudut pada diagram pengendalian ini memiliki nilai kecepatan sudut realita yang terukur pada saat itu. Sedangkan nilai *set point* adalah nilai kecepatan sudut yang diinginkan. Sinyal keluaran dari sensor yang berupa nilai kecepatan sudut, diubah menjadi nilai *error* sebagai salah satu masukan pengendali *Fuzzy Inference System* dengan melakukan pengurangan terhadap nilai *set point*.

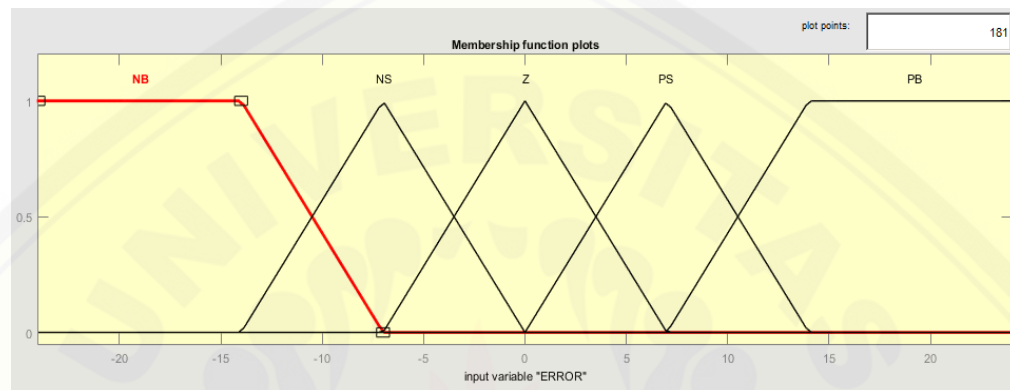
3.4.1 Penentuan Masukan dan Keluaran *fuzzy*

Pada penelitian ini metode *fuzzy* yang digunakan adalah metode Sugeno, dimana keluaran sistem merupakan persamaan linier. Langkah pertama dalam perancangan pengendali *Fuzzy* yaitu dilakukan penentuan parameter masukan yang dapat mempengaruhi performansi sistem. Masukan dari pengendali *Fuzzy* adalah *error* yang didapat dari selisih antara nilai *set point* dengan tegangan keluaran *boost converter* yang terukur, kemudian *delta error* yang diperoleh dari selisih antara *error* pembacaan tegangan keluaran sekarang dan pembacaan *error* yang sebelumnya. Dimana keluaran *Fuzzy* adalah penambahan atau pengurangan PWM yang masuk ke pengaturan *switching* pada *boost converter*. Semakin besar nilai PWM maka tegangan yang dihasilkan *boost converter* semakin besar, sebaliknya jika semakin kecil nilai PWM maka tegangan yang di hasilkan *boost converter* semakin kecil.

Keanggotaan pada *membership function error* dibagi menjadi 5 bagian. Adapun variable linguistik yang digunakan dalam *membership function error* adalah

1. NB : *Negative Big*

2. NS : *Negative Small*
3. Z : *Zero*
4. PS : *Positive Small*
5. PB : *Positive Big*



Gambar 3.8 *Membership Function Error*

(Sumber : Penulis)

Sedangkan yang dimaksud untuk *delta error* disini adalah *error* sekarang diurangi *error* sebelumnya. Keanggotaan pada *membership function delta error* dibagi menjadi 5 bagian. Adapun variable linguistik yang digunakan dalam *membership function error* adalah

1. NB : *Negative Big*
2. NS : *Negative Small*
3. Z : *Zero*
4. PS : *Positive Small*
5. PB : *Positive Big*

4.5 Pengujian Beban Resistif *Open Loop* dan *Close Loop fuzzy logic*

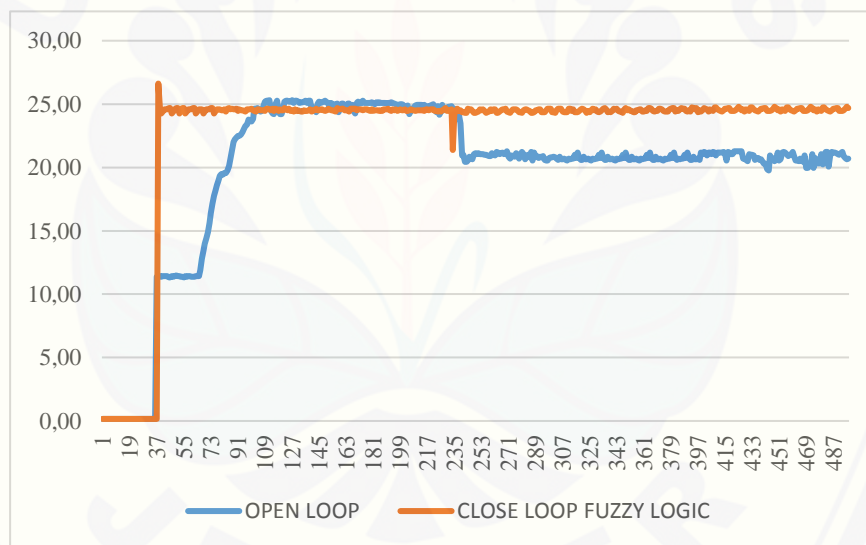
Pada pengujian ini *boost converter* di uji dengan beban resistif menggunakan variasi ukuran 39 *ohm*, 100 *ohm*, dan 240 *ohm* dengan memberi tegangan masukan dari *power supply* sebesar 11,8 V dimana tegangan tersebut merupakan representasi dari tegangan rata-rata pada baterai LiPo yang digunakan pada robot. Adapun langkah pengujian pada kondisi *open loop* yang dilakukan adalah dengan mengatur besarnya *dutycycle* sampai menuju ke *setpoint* yang diatur menggunakan potensiometer dengan ukuran 100K *ohm*. Setelah *boost converter* mencapai *setpoint* yaitu 24 V, kemudian *boost converter* dihubungkan secara langsung dengan beban resistif dengan ukuran 39 *ohm*, 100 *ohm*, dan 240 *ohm*. dari pengujian *open loop* nantinya akan didapatkan data *drop* tegangan pada setiap beban yang diberikan. Kemudian pada pengujian *close loop* dengan metode *fuzzy logic* ketika *boost converter* diaktifkan dan tanpa memutar potensiometer, *dutycycle* akan menyesuaikan dengan tegangan *output* yang diinginkan atau yang dapat disebut dengan *setpoint*. Dari pengujian ini akan didapatkan data respon sistem atau *recovery time* terhadap *drop* tegangan yang disebabkan pemberian beban. Sehingga berdasarkan percobaan pengujian *boost converter open loop* di dapatkan data pengujian seperti pada tabel 4.9

Tabel 4.9 Pengujian Pemberian Beban Resistif pada *Boost Converter*

No	Beban (Ohm)	Tegangan Keluaran (volt)	<i>Recovery Time</i>
1	0	24.63	-
2	240	20.18	0,1 s
3	100	19.94	0,2 s
4	39	18.92	0,2 s

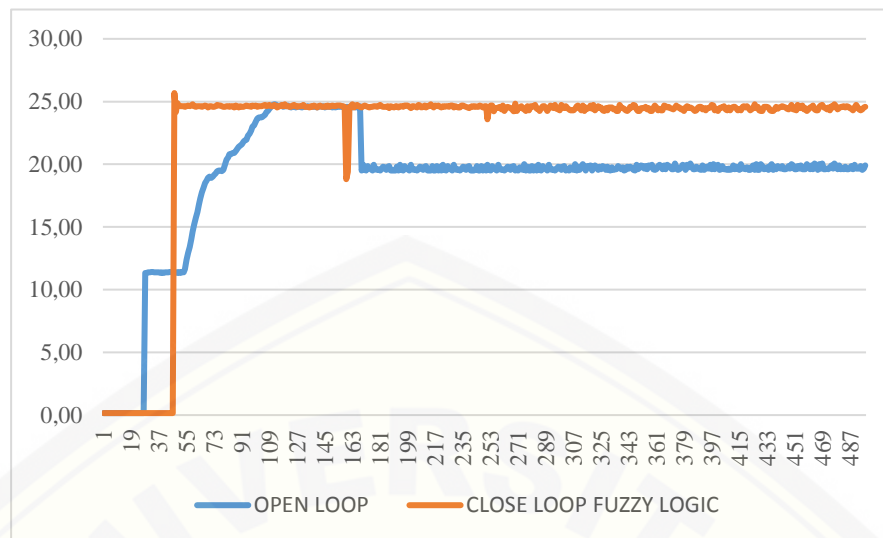
Berdasarkan data pengujian yang dilakukan maka didapatkan grafik hasil perbandingan besarnya *drop* tegangan untuk masing-masing pembebanan yang dilakukan ketika kondisi *open loop* dengan *Recovery Time* pada kondisi *close loop* metode *fuzzy logic*. seperti yang terdapat pada tabel 4.9 dimana diketahui bahwa pada saat terjadi *drop* tegangan maka tegangan tidak akan kembali pada tegangan *set point* yaitu 24 V. *Recovery time* merupakan waktu tempuh tegangan *setpoint*

mengalami *drop* dan kembali menuju ke *setpoint* pada percobaan *close loop*. Pada pengujian ini *drop* tegangan yang paling besar terjadi pada saat *boost converter* diberikan beban sebesar 39 Ohm dimana terjadi *drop* sebesar 5,71 Volt sehingga tegangan keluaran *boost converter* menjadi 18,92 Volt, untuk *drop* tegangan yang terjadi ketika beban 100 Ohm yaitu 4,69 V sehingga menghasilkan tegangan 19,94 V. Sedangkan *drop* tegangan terkecil terjadi ketika pada beban 240 Ohm dimana mengalami *drop* tegangan 4,45V dimana menghasilkan tegangan 20,18 V. Pada pengujian *close loop*, dan pada *boost converter* diterapkan metode *fuzzy logic* sebagai pengatur tegangan keluaran secara *automatis* yang menyesuaikan dan menstabilkan tegangan keluaran pada *setpoint* yaitu 24 V meskipun tegangan masuknya diubah ataupun diberikan beban, tegangan keluarannya akan tetap pada *setpoint*.



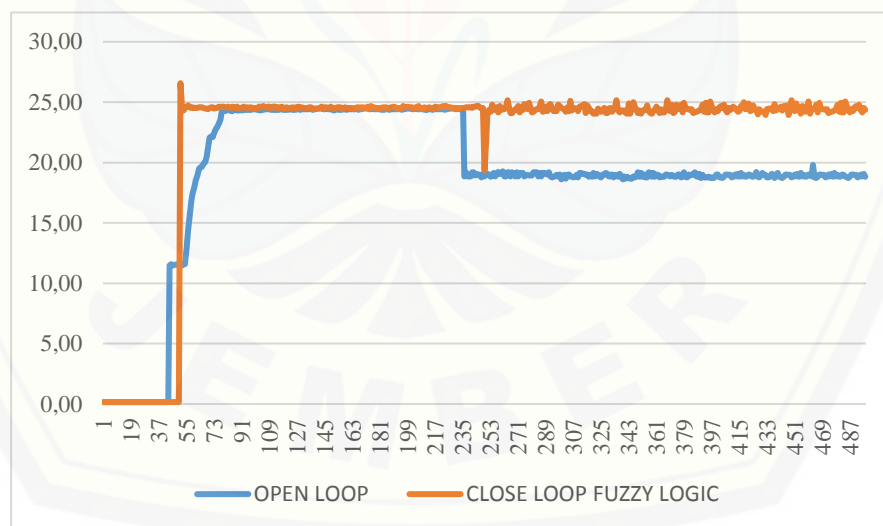
Gambar 4.11 Grafik Perbandingan Antara *Boost Converter Open Loop* Dengan *Close Loop Fuzzy Logic* Pada Beban 240 Ohm

Pada pengujian dengan beban 240 Ohm, terlihat pada grafik *close loop fuzzy logic* yang mengalami *drop* tegangan namun dapat kembali lagi ke *setpoint* dengan *recovery time* 0,1 detik. Berbeda dengan grafik *open loop* yang tidak dapat kembali lagi ke *setpoint*.



Gambar 4.12 Grafik Perbandingan Antara *Boost Converter Open Loop* Dengan *Close Loop Fuzzy Logic* Pada Beban 100 Ohm

Pada pengujian dengan beban 100 Ohm, terlihat pada grafik *close loop fuzzy logic* yang mengalami *drop* tegangan namun dapat kembali lagi ke *setpoint* dengan *recovery time* 0,2 detik. Berbeda dengan grafik *open loop* yang tidak dapat kembali lagi ke *setpoint*.

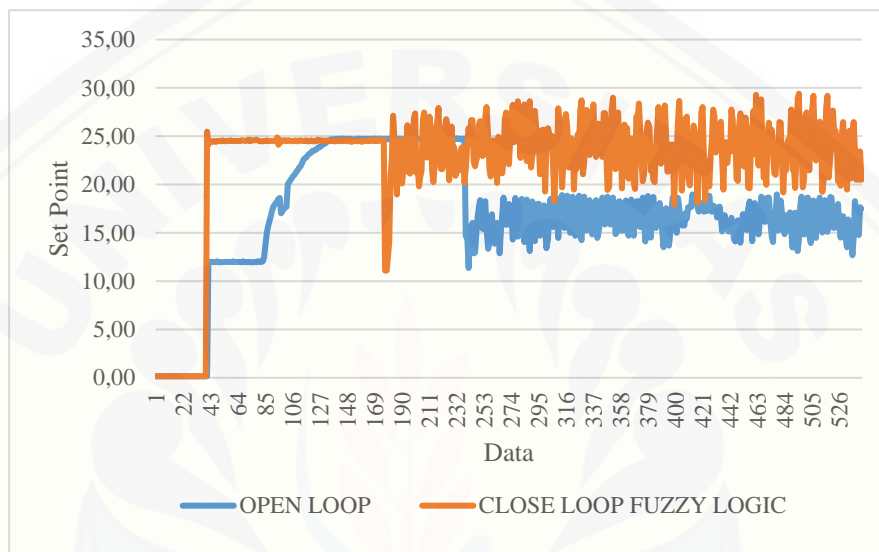


Gambar 4.13 Grafik Perbandingan Antara *Boost Converter Open Loop* Dengan *Close Loop Fuzzy Logic* Pada Beban 39 Ohm

Pada pengujian dengan beban 39 Ohm, terlihat pada grafik *close loop fuzzy logic* yang mengalami *drop* tegangan namun dapat kembali lagi ke *setpoint* dengan *recovery time* 0,2 detik. Berbeda dengan grafik *open loop* yang tidak dapat kembali lagi ke *setpoint*.

4.6 Pengujian Statis Pada Motor PG 45 *Open Loop* dan *Close Loop* Dengan Metode *Fuzzy Logic*

Pada pengujian ini menggunakan beban motor bertipe PG45 yang digunakan pada robot. Dimana menggunakan tegangan masukan dari baterai LiPo. Pada pengujian ini motor dalam keadaan statis dalam artian motor diuji tanpa menggunakan beban *body* robot dan tidak dalam kondisi berjalan.



Gambar 4.14 Grafik Perbandingan Antara *Boost Converter Open Loop* Dengan *Close Loop Fuzzy Logic* Pada Motor PG45 Dengan Kondisi Statis

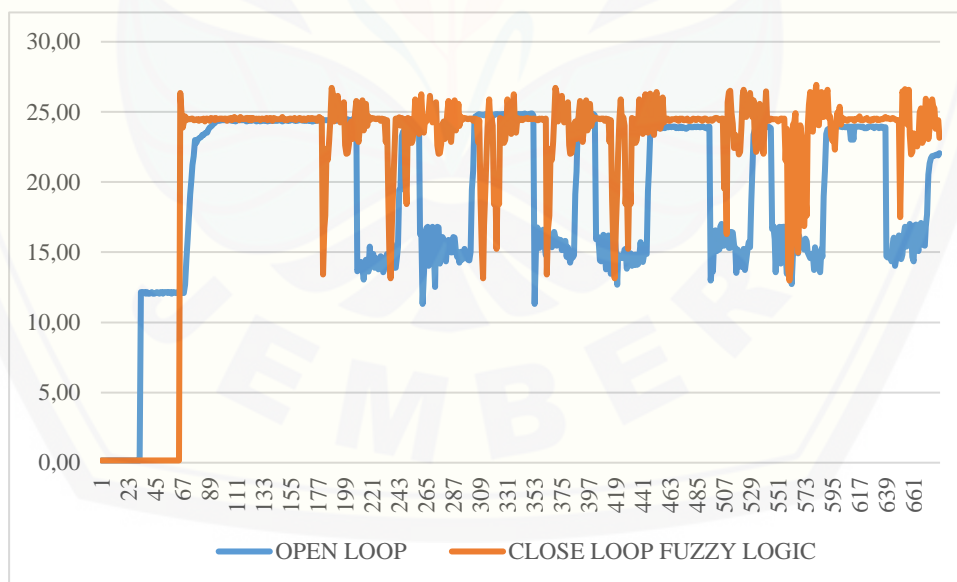
Pada pengujian statis pada motor PG45 *open loop* dimana *setpoint* diatur secara manual menggunakan potensiometer sampai ke tegangan 24 V kemudian *boost converter* dihubungkan ke beban motor dan tegangan pada *boost converter* mengalami penurunan atau *drop* tegangan. Dikarenakan menggunakan beban motor, maka *drop* tegangan juga berubah-ubah dikarenakan resistansi pada motor tidak selalu sama. Lain halnya dengan beban resistor yang nilainya selalu tetap. Dimana *drop* tegangan tertinggi mencapai tegangan 13,22 sehingga tegangan keluaran menghasilkan 11,37 V dan *drop* tegangan terendah mencapai tegangan 5,55 V dan menjadikan tegangan keluaran menjadi 18,97.

Pada pengujian *close loop* dimana tegangan pada *setpoint* diatur 24 V sesuai dengan tegangan pada motor PG45. Saat *boost converter* dihubungkan dengan

motor, tegangan awal sempat mengalami *overshoot* sebesar 25,47 pada *boost converter* dan mengalami *drop* tegangan dengan *recovery time* selama 0,8 detik kemudian tegangan kembali lagi ke *setpoint*. Sama halnya dengan pengujian *open loop*, dikarenakan resistansi pada motor berubah ubah dan mengakibatkan tegangan menjadi fluktuatif namun masih dalam toleransi *setpoint* yaitu 10%.

4.7 Pengujian Dinamis Pada Robot JR-EVO *Open Loop* dan *Close Loop* Dengan Metode *Fuzzy Logic*

Pada percobaan ini *boost converter* diterapkan ke robot dan diuji secara dinamis dengan cara menjalankan robot. *Boost converter* diberikan tegangan masukan dari baterai LiPo kemudian tegangan keluaran pada *boost converter* di hubungkan pada motor robot melalui *driver* motor. Dimana pada *driver* motor memiliki dua jenis *supply* yaitu *supply* untuk motor dan *supply* relay yang berfungsi untuk mengubah arah putar motor. Pada pengujian ini didapatkan data perbandingan tegangan keluaran *boost converter open loop* dan *close loop* dengan metode *fuzzy logic*.



Gambar 4.15 Grafik Perbandingan Antara *Boost Converter Open Loop* Dengan *Close Loop Fuzzy Logic* Pada Robot JR-EVO Dengan Kondisi Dinamis

Dari grafik pengujian sistem pengendali *open loop* dapat dilihat respon sistem tanpa menggunakan kontrol *fuzzy* dan saat di kontrol menggunakan *fuzzy*. Dari grafik tersebut tegangan pada boost converter mengalami *drop* dikarenakan beban dalam kondisi berputar atau berjalan. Dan akan kembali lagi ke *setpoint* ketika motor berhenti berputar. Dimana jika tegangan keluaran *boost converter* mengalami *drop* mencapai 11,32 V akan mengakibatkan kecepatan motor berkurang sampai 296,5 RPM. Hal itu disebabkan tegangan yang masuk ke motor tidak lagi tegangan *setpoint* yaitu 24V melainkan tegangan yang sudah *drop*. Berbeda pada grafik *close loop* dengan metode *fuzzy logic* dimana tegangan keluaran mengalami *overshoot* sampai 26,36 V dan mengalami osilasi selama 0,1 detik dan kembali ke *setpoint*. Ketika robot mulai berjalan dan motor mulai berputar, *boost converter* sempat mengalami *drop* tegangan mencapai 11,08 V sehingga *boost converter* menghasilkan tegangan 13,41 V dalam sesaat dan respon sistem mengembalikan lagi tegangan yang *drop* kembali ke *setpoint* dengan *recovery time* selama 0,7 detik dan kembali lagi ke *setpoint*. dalam waktu seketika ketika motor kembali berjalan maka *boost converter* mengalami *overshoot* sampai 25,89V menyebabkan arus yang mengalir pada motor cukup besar kemudian normal kembali ketika motor sudah berputar stabil. Dari hasil perbandingan grafik yang diperoleh dapat diketahui jika *boost converter* dengan metode *fuzzy logic* dapat berkerja dengan menaikkan dari tegangan baterai LiPo 12 V menjadi tegangan 24 V dengan toleransi 10 %. Sehingga dapat menjalankan motor pada robot yang berkerja dengan tegangan 24 V dengan kecepatan motor 515,3 RPM.

BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang berjudul “Rancang Bangun *Dc To Dc Boost Converter* Pada Robot Krsbi Beroda Jr – Evo Berbasis *Fuzzy Logic*” dapat disimpulkan sebagai berikut :

1. pengujian *boost converter* dengan beban resistif, semakin kecil nilai hambatan yang diberikan, maka semakin besar *drop* tegangan yang akan terjadi, pada saat beban 39Ω terjadi *drop* tegangan sebesar 5,71 Volt sedangkan saat diberikan beban 240Ω terjadi *drop* tegangan sebesar 4,45 Volt.
2. Pada pengujian statis dengan beban motor PG45, dengan kondisi *open loop* pada keadaan *setpoint* ketika motor mulai dijalankan terjadi *drop* tegangan sampai 13,22 V sehingga kecepatan motor turun menjadi 296,5 RPM. Pada kondisi *close loop* dengan metode *fuzzy logic* tegangan yang *drop* dapat kembali ke *setpoint* dengan *recovery time* 0,8 detik dan kecepatan motor menjadi 515,3 RPM.
3. Pada pengujian dengan robot terjadi *over shoot* sesaat sampai tegangan 26,36 V ketika sistem mulai di aktifkan dan saat sistem kembali ke *setpoint*. Dimana pada saat robot mulai berjalan dan motor berputar terjadi *drop* tegangan sesaat sebesar 11,08 V dan kembali lagi ke *setpoint* dengan *recovery time* selama 0,7 detik. Sehingga *boost converter* dengan metode *fuzzy logic* mampu menjaga tegangan tetap pada *setpoint* yaitu 24 V dengan toleransi 10%.

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. kontrol untuk kestabilan tegangan keluaran dapat digantikan dengan sistem kontrol yang lain seperti *hybrid PID fuzzy* ataupun *ANFIS* sebagai perbandingan.
2. Dalam penelitian selanjutnya dimungkinkan untuk menggunakan objek robot lain sebagai beban serta konfigurasi mekanisme pemasangan roda dan motor pada robot yang berbeda seperti *four omni wheel* dan menggunakan roda *mecanum*.

DAFTAR PUSTAKA

- Arduino UNO, R3. 2015. "A000066 Arduino | Mouser." *MOUSER ELECTRONICS*. Diakses 3 september 2018.
- Autodesk.Inc. 2015. *id/Arduino-Voltage-Sensor-0-25V/*. Diakses Agustus 2018, dari instructables: <http://www.instructables.com/id/Arduino-Voltage-Sensor-0-25V/>
- Aziz, M.L. 2018, *Sistem Kontrol Fuzzy Logic Untuk Tegangan Keluaran Boost Converter Pada Pengisian Baterai Sepeda Listrik Menggunakan Solar Cell*. Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember, Jember.
- Bench, H. 2015. "ACS712 Current Sensor User Manual". Diakses Agustus 2018. <http://henrysbench.cdnfat.com/henrys-bench/arduino-current-measurements/acs712-current-sensor-user-manual/>
- Fathurachman, A ., Najmurrokhman, A ., Kusnandar. 2014, *Perancangan Boost Converter Untuk sistem Pembangkit Listrik Tenaga Surya* . Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jendral Achmad Yani, Bandung.
- Halder, T.2017. *An improved design guideline of the PWM flyback converter for the practical engineer*.IEEE.221-226
- Himawan, H. M., Setyawati, O., & Suyono, H. 2016. *Pemodelan Fuzzy Logic Control untuk pengendali PWM pada Buck Converter*. *JNTETI, Vol.5, No. 1*.
- Kusumadewi, S., & Hartati, S. 2006. *NEURO FUZZY: Aplikasi Logika Fuzzy untuk Pendukung Keputusan Edisi 2*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S., & Purnomo, H. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan edisi 2*. Yogyakarta: Graha Ilmu.
- Manufactur, R.I. 2017, "Datasheet Pg45 Series 775 Motor Dc Carbon Brushed". www.Brontoseno.com. Diakses 13 September 2018.
- N.F Nik Ismail., I. Musirin., R. Bahrom.2010. *An improved design guideline of the PWM flyback converter for the practical engineer*. International Conference On Power And Energy, Kuala lumpur, Malaysia.

- Oetomo., Halim, L . 2017. *Perancangan Dan Implementasi Sistem Charging & Monitoring Baterai Lithium*. Universitas Katolik Parahiyangan.
- Soetens, R., Molengraft, R.V.D., Cunha, B. 2011, *Robocup MSL History, Accomplishments, Current Status And Challenges Ahead* . Eindhoven University Of Techology, Netherlands.
- Suswitno., Rahayu, Y., Amri, R., Hamdani, E. 2017, *Perancangan Konverter DC ke DC Untuk Menstabilkan Tegangan Keluaran Panel Solar Cell Menggunakan Teknologi Boost Converter* . Jurusan Teknik Elektro, Fakultas Teknik, Universitas Riau, Riau.
- Tony. 2010, *Analisis Dan Perancangan Teknologi Pada Robot Sepak Bola* . Fakultas Teknologi Informasi, Universitas Trumanegara , Jakarta Barat.
- Ugale, Chetan P., R.B. Dhumale, V.V. Dixit. 2015. *DC-DC Converter Using Fuzzy Logic Controller*. 02. International Research Journal of Engineering and Technology (IRJET).
- Wibisono, G. 2011, *Perancangan Kontroler Fuzzy Untuk Tracking Control Robot Soccer* . Institut Teknologi Sepuluh November, Surabaya.



LAMPIRAN

A. Lampiran *Listing* Program

```
#include "fis_header.h"

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 25;
//=====
int pinpwm = 5;
int pwm = 0;
int analogInputo = A0;
int analogInput = A1;
int val = 0;
float vouto = 0.0;
float vino = 0.0;
int Errorr = 0.0;
int Setpoint = 0.0;
float R1o = 10000;
float R2o = 1018;
int value = 0;
int valueo = 0;
//=====
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
int Error = A5;
int Deltaerror = A4;
void setup()
```

```
{
    // Pin mode for Input: ERROR
    pinMode(Error , INPUT);
    // Pin mode for Input: DELTA-ERROR
    pinMode(Deltaerror , INPUT);

    // initialize the Analog pins for output.
    // Pin mode for Output: INDECPWM
    pinMode(pinpwm , OUTPUT);
    TCCR0B = TCCR0B & B11111000 | B00000001;
    Serial.begin(9600);
}
void loop()
{
    valueo = analogRead(analogInputo);
    vouto = (valueo * 5.0) / 1024.0;
    vino = vouto / (R2o/(R1o+R2o));
    Errorr = vino;
    // Read Input: ERROR
    g_fisInput[0] = analogRead(Error);
    // Read Input: DELTA-ERROR
    g_fisInput[1] = analogRead(Deltaerror);

    g_fisOutput[0] = 0;

    fuzzy();
}
//*****
*****
// Fuzzy Inference System
```



```
//*****  
*****  
void fuzy (){  
    Setpoint = 24;  
    if(Errorr < Setpoint){  
        pwm = pwm+10;  
        pwm = constrain(pwm, 1, 210);  
    }  
  
    if(Errorr > Setpoint){  
        pwm = pwm-10;  
        pwm = constrain(pwm, 1, 210);  
    }  
  
    analogWrite (pinpwm,pwm);  
}  
//*****  
*****  
// Support functions for Fuzzy Inference System  
//*****  
*****  
// Trapezoidal Member Function  
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)  
{  
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];  
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ?  
((d - x) / (d - c)) : 0)));  
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ?  
((x - a) / (b - a)) : 0)));  
    return (FIS_TYPE) min(t1, t2);  
}
```

```
// Triangular Member Function
FIS_TYPE fis_trmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size,
    _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];
```

```
ret = array[0];
for (i = 1; i < size; i++)
{
    ret = (*pfnOp)(ret, array[i]);
}

return ret;
}

//*****
//*****
// Data for Fuzzy Inference System
//*****
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trapmf, fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 5, 5 };

// Count of member function for each Output
int fis_gOMFCount[] = { 5 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMF10Coeff1[] = { -10.42, -10, -5.833, -2.917 };
FIS_TYPE fis_gMF10Coeff2[] = { -5.833, -2.917, 0 };
FIS_TYPE fis_gMF10Coeff3[] = { -2.917, 0, 2.917 };
FIS_TYPE fis_gMF10Coeff4[] = { 0, 2.917, 5.833 };
```

```
FIS_TYPE fis_gMFI0Coeff5[] = { 2.917, 5.833, 10, 10.42 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,
fis_gMFI0Coeff2, fis_gMFI0Coeff3, fis_gMFI0Coeff4,
fis_gMFI0Coeff5 };
FIS_TYPE fis_gMFI1Coeff1[] = { -10.42, -10, -5.833, -2.917 };
FIS_TYPE fis_gMFI1Coeff2[] = { -5.833, -2.917, 0 };
FIS_TYPE fis_gMFI1Coeff3[] = { -2.917, 0, 2.917 };
FIS_TYPE fis_gMFI1Coeff4[] = { 0, 2.917, 5.833 };
FIS_TYPE fis_gMFI1Coeff5[] = { 2.917, 5.833, 10, 10.42 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,
fis_gMFI1Coeff2, fis_gMFI1Coeff3, fis_gMFI1Coeff4,
fis_gMFI1Coeff5 };
FIS_TYPE** fis_gMFI0Coeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff
};

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -15, -10, -6, -3 };
FIS_TYPE fis_gMFO0Coeff2[] = { -6, -3, 0 };
FIS_TYPE fis_gMFO0Coeff3[] = { -3, 0, 3 };
FIS_TYPE fis_gMFO0Coeff4[] = { 0, 3, 6 };
FIS_TYPE fis_gMFO0Coeff5[] = { 3, 6, 10, 15 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1,
fis_gMFO0Coeff2, fis_gMFO0Coeff3, fis_gMFO0Coeff4,
fis_gMFO0Coeff5 };
FIS_TYPE** fis_gMFO0Coeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 1, 1, 1, 0 };
int fis_gMFI1[] = { 0, 1, 1, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1};

// Output membership function set
```

```
int fis_gMFO0[] = { 0, 1, 1, 1, 0 };
int* fis_gMFO[] = { fis_gMFO0};

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 1, 2 };
int fis_gRI2[] = { 1, 3 };
int fis_gRI3[] = { 1, 4 };
int fis_gRI4[] = { 1, 5 };
int fis_gRI5[] = { 2, 1 };
int fis_gRI6[] = { 2, 2 };
int fis_gRI7[] = { 2, 3 };
int fis_gRI8[] = { 2, 4 };
int fis_gRI9[] = { 2, 5 };
int fis_gRI10[] = { 3, 1 };
int fis_gRI11[] = { 3, 2 };
int fis_gRI12[] = { 3, 3 };
int fis_gRI13[] = { 3, 4 };
int fis_gRI14[] = { 3, 5 };
int fis_gRI15[] = { 4, 1 };
int fis_gRI16[] = { 4, 2 };
int fis_gRI17[] = { 4, 3 };
int fis_gRI18[] = { 4, 4 };
int fis_gRI19[] = { 4, 5 };
```



```
int fis_gRI20[] = { 5, 1 };
int fis_gRI21[] = { 5, 2 };
int fis_gRI22[] = { 5, 3 };
int fis_gRI23[] = { 5, 4 };
int fis_gRI24[] = { 5, 5 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3,
fis_gRI4, fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8, fis_gRI9,
fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14,
fis_gRI15, fis_gRI16, fis_gRI17, fis_gRI18, fis_gRI19,
fis_gRI20, fis_gRI21, fis_gRI22, fis_gRI23, fis_gRI24 };

// Rule Outputs
int fis_gRO0[] = { 1 };
int fis_gRO1[] = { 1 };
int fis_gRO2[] = { 2 };
int fis_gRO3[] = { 2 };
int fis_gRO4[] = { 3 };
int fis_gRO5[] = { 1 };
int fis_gRO6[] = { 2 };
int fis_gRO7[] = { 2 };
int fis_gRO8[] = { 3 };
int fis_gRO9[] = { 4 };
int fis_gRO10[] = { 2 };
int fis_gRO11[] = { 2 };
int fis_gRO12[] = { 3 };
int fis_gRO13[] = { 4 };
int fis_gRO14[] = { 4 };
int fis_gRO15[] = { 2 };
int fis_gRO16[] = { 3 };
int fis_gRO17[] = { 4 };
int fis_gRO18[] = { 4 };
int fis_gRO19[] = { 5 };
```

```
int fis_gRO20[] = { 3 };
int fis_gRO21[] = { 4 };
int fis_gRO22[] = { 4 };
int fis_gRO23[] = { 5 };
int fis_gRO24[] = { 5 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3,
fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9,
fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13, fis_gRO14,
fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19,
fis_gRO20, fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24 };

// Input range Min
FIS_TYPE fis_gIMin[] = { -10, -10 };

// Input range Max
FIS_TYPE fis_gIMax[] = { 10, 10 };

// Output range Min
FIS_TYPE fis_gOMin[] = { -10 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 10 };

//*****
*****
// Data dependent support functions for Fuzzy Inference System
//*****
*****

FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int
o)
{
    FIS_TYPE mfOut;
```

```
int r;

for (r = 0; r < fis_gcR; ++r)
{
    int index = fis_gRO[r][o];
    if (index > 0)
    {
        index = index - 1;
        mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
    }
    else if (index < 0)
    {
        index = -index - 1;
        mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
    }
    else
    {
        mfOut = 0;
    }

    fuzzyRuleSet[0][r] = fis_min(mfOut,
fuzzyRuleSet[1][r]);
}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR,
fis_max);
}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
```

```
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) /
(FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF
outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }

    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2)
: (momentum / area));
}

void fuzzy (){
    Setpoint = 24;
    if(Errorr < Setpoint){
        pwm = pwm+1;
        pwm = constrain(pwm, 1, 210);
    }

    if(Errorr > Setpoint){
        pwm = pwm-1;
        pwm = constrain(pwm, 1, 210);
    }

    analogWrite (pinpwm,pwm);
}
```

```
Serial.println(vino);
}
//*****
*****
// Fuzzy Inference System
//*****
*****
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0,
fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i],
fis_gMFICoeff[i][j]);
        }
    }
}
```



```
int index = 0;
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
```

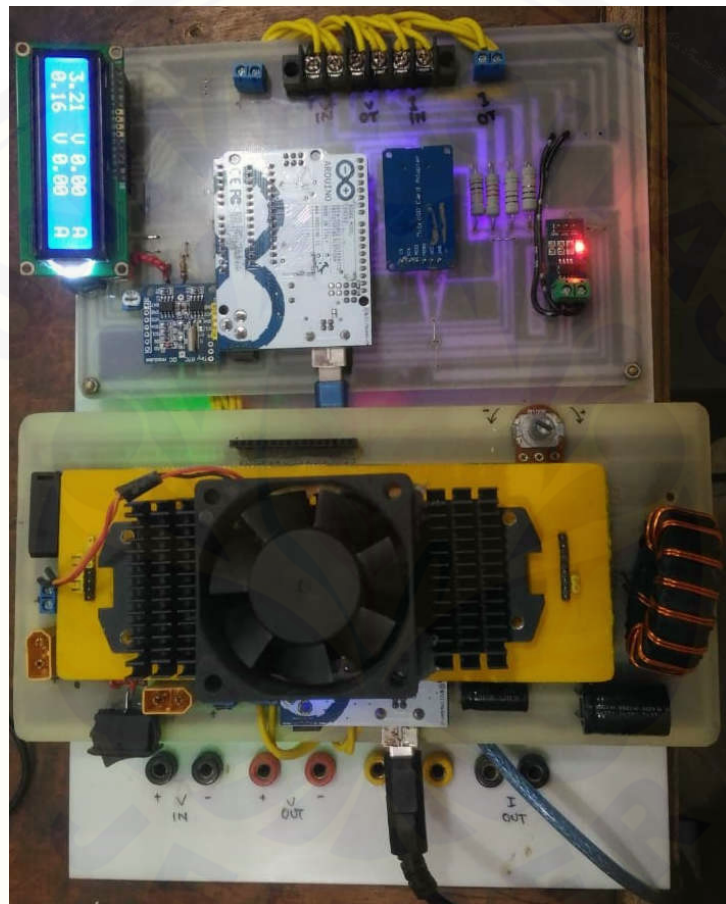
```
        fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
    }
}

fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
sW += fuzzyFires[r];
}

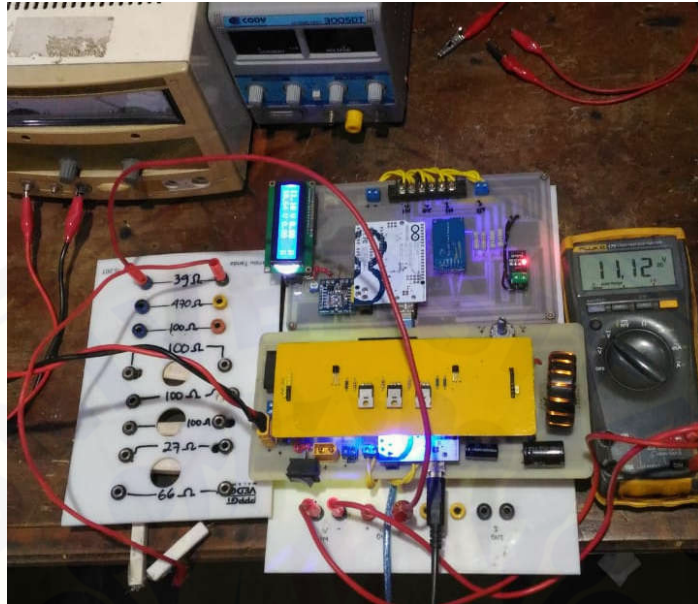
if (sW == 0)
{
    for (o = 0; o < fis_gc0; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) /
2);
    }
}
else
{
    for (o = 0; o < fis_gc0; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet,
o);
    }
}
}
```

B. Lampiran Gambar

B.1 Gambar Alat



B.2 Gambar Pengujian Beban Resistif



B.3 Gambar Pengujian pada robot JR-EVO

