



**RANCANG BANGUN *AC – AC CHOPPER* SEBAGAI *DRIVER*
MOTOR UNIVERSAL BERBASIS *FUZZY LOGIC*
*CONTROLLER***

SKRIPSI

Oleh
Ardhiansyah Fany Ilhami
NIM. 161910201071

**PROGRAM STUDI TEKNIK ELEKTRO STRATA 1
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2021**



**RANCANG BANGUN AC – AC *CHOPPER* SEBAGAI *DRIVER*
MOTOR UNIVERSAL BERBASIS *FUZZY LOGIC*
*CONTROLLER***

SKRIPSI

Diajukan guna melengkapai tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh
Ardhiansyah Fany Ilhami
NIM. 161910201071

**PROGRAM STUDI TEKNIK ELEKTRO STRATA 1
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2021**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, Tuhan semesta alam tempatku kembali.
2. Nabi Muhammad SAW, Nabi junjungan umat Islam yang telah menarik kami dari zaman jahiliyah ke zaman yang lebih baik.
3. Kedua orang tua saya, bapak drh. Mohammad Amir Roeslan dan ibu Fabtimawati.
4. Dosen pembimbing saya, bapak Andi Setiawan, S.T., M.T., dan bapak Dr. Ir. Widjonarko, S.T., M.T.

MOTTO

“Jika kalian bersyukur maka akan aku tambahkan nikmat-Ku untuk kalian.”

(QS. Ibrahim : 7)

“Kegagalan hanya akan terjadi bila kita menyerah.”

(Prof. Dr. Ing. H. Bacharuddin Jusuf Habibie, FEng.)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ardhiansyah Fany Ilhami

NIM : 161910201071

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: **“Rancang Bangun AC – AC Chopper sebagai Driver Motor Universal Berbasis Fuzzy Logic Controller”** adalah benar-benar karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi mana pun, serta bukan karya jiplakan. Saya bertanggung jawab 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 mana pun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 22 Januari 2021

Yang menyatakan,

Ardhiansyah Fany Ilhami

NIM.161910201071

SKRIPSI

**RANCANG BANGUN *AC – AC CHOPPER* SEBAGAI *DRIVER*
MOTOR UNIVERSAL BERBASIS *FUZZY LOGIC*
*CONTROLLER***

Oleh

Ardhiansyah Fany Ilhami

NIM 161910201071

Pembimbing

Dosen Pembimbing Utama : Andi Setiawan, S. T., M. T.

Dosen Pembimbing Anggota : Dr. Ir. Widjonarko, S. T., M. T.

PENGESAHAN

Skripsi berjudul “Rancang Bangun AC – AC Chopper sebagai Driver Motor Universal Berbasis *Fuzzy Logic Controller*” karya Ardhiansyah Fany Ilhami telah diuji dan disahkan pada:

Hari : Jumat

Tanggal : 22 Januari 2021

Tempat : Fakultas Teknik Universitas Jember

Tim Penguji,

Ketua,

Anggota I,

Andi Setiawan, S. T., M. T.
NIP 196910101997021001

Dr. Ir. Widjonarko, S. T., M. T.
NIP 197109081999031001

Anggota II,

Anggota III,

Ir. Widyono Hadi, M.T.
NIP 196104141989021001

Guido Dias Kalandro, S. ST., M.Eng.
NRP 760015734

Mengesahkan
Dekan,

Dr. Ir. Triwahju Hardianto, S.T., M.T.
NIP 197008261997021001

RINGKASAN

Rancang Bangun AC – AC Chopper sebagai Driver Motor Universal Berbasis Fuzzy Logic Controller; Ardhiansyah Fany Ilhami, 161910201071; 2020; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik. Motor listrik dapat ditemukan pada peralatan rumah tangga seperti kipas angin, mesin cuci, pompa air dan penyedot debu. Pada motor listrik tenaga listrik diubah menjadi tenaga mekanik. Motor universal adalah motor arus bolak balik, konstruksi maupun karakteristik motor universal sama dengan motor arus searah. Keuntungan motor universal ini dapat dioperasikan dengan sumber tegangan bolak balik atau dengan tegangan arus searah pada nilai tegangan yang sama. Motor universal yang banyak digunakan memiliki perangkat pengendali yang berfungsi untuk mengendalikan kecepatan yang akan dihasilkan oleh motor. Chopper (pemangkas) merupakan suatu rangkaian yang digunakan untuk mengubah sumber masukan tegangan AC tetap menjadi sumber keluaran tegangan AC yang dapat dikendalikan atau diatur.

Rangkaian *AC Chopper* menjadi salah satu piranti yang dapat mengendalikan kecepatan yang dihasilkan oleh motor. Piranti ini akan mengendalikan kecepatan dengan cara merubah nilai tegangan input yang digunakan. *AC Chopper* yang digunakan berjenis *basic AC Chopper* menggunakan 4 buah diode bertipe *super-fast recovery rectifier diode* dan dikombinasikan dengan MOSFET bertipe IRF740 yang berfungsi sebagai *switching* sinyal PWM yang akan mengendalikan perubahan tegangan *chopping*. Rangkaian *AC Chopper* sebagai *driver* motor universal dikombinasikan dengan *Fuzzy Logic Controller* untuk mengendalikan kecepatan pada kestabilan 1000 RPM, serta menggunakan sensor *Optical Encoder* LM393 sebagai perangkat dengan hasil pembacaan akan dikirim kembali pada sistem fuzzy sebagai *input*. *Fuzzy Logic Controller* yang digunakan menggunakan sistem MISO (*Multiple Input Single Output*) dengan *input* adalah nilai *error* dan *Derror*, dimana nilai *error* diperoleh dari selisih nilai pembacaan

sensor dengan nilai kecepatan yang diinginkan, sedangkan untuk nilai *Derror* diperoleh dari selisih nilai *error* sekarang dengan *error* sebelumnya. Pengujian dilakukan sebanyak lima kali untuk mengetahui *error* dan besaran nilai *chopping* tegangan yang dihasilkan oleh piranti *AC Chopper*. Terdapat juga pengujian tanpa control dengan beban untuk mengetahui efisiensi dari perangkat *driver*, serta dilakukan pengujian saat kondisi tanpa beban dan berbeban menggunakan *Fuzzy Logic Controller*. Beban yang digunakan adalah sebuah *permanent magnet synchron generator*.

Hasil dari pengujian yang telah dilakukan diperoleh nilai *error* terkecil 0,13% dan nilai *error* yang terbesar 81,16% dengan *error* rata – rata dari perangkat *AC Chopper* sebesar 38,7% dan rata – rata nilai pemangkasan tegangan sebesar 17,74 V pada kisaran *dutycycle* 7,8% - 94% saat pengujian pertama dengan 5 jenis nilai tegangan *input* yang berbeda. Nilai efisiensi yang dihasilkan dari pengujian kontrol manual dengan beban *permanent magnet synchron generator* memiliki rata – rata nilai 86,96% dengan *range* nilai *dutycycle* 52% hingga 80%. Pada pengujian sistem keseluruhan dengan nilai tegangan *input* pada *driver* 63,57 V dan nilai *setpoint dutycycle* 50% dilakukan dua jenis pengujian, tanpa beban dan berbeban. Saat pengujian tanpa beban kecepatan motor dapat stabil di rentang 1000 RPM dengan *dutycycle* 7% - 10 % dengan nilai tegangan *output* 44,26 V hingga 44,59 V dengan arus tanpa beban 1,5 mA. Sedangkan saat pengujian kondisi diberi beban *permanent magnet synchron generator*, kecepatan motor dapat stabil di rentang 1000 RPM dengan *dutycycle* 85% hingga 86% dengan nilai tegangan *output* 60,02 V hingga 60,51 V serta memiliki nilai arus 3,02 mA.

Hasil dari perbandingan keda pengujian saat tanpa menggunakan beban dengan menggunakan beban yaitu waktu untuk mencapai nilai 1000 RPM yang dihasilkan motor universal pada saat kondisi tidak menggunakan beban lebih cepat dari pada saat menggunakan beban.

SUMMARY

AC - AC Chopper Design as Universal Motor Driver based on Fuzzy Logic Controller; Ardhiansyah Fany Ilhami, 161910201071; 2020; Department of Electrical Engineering, Faculty of Engineering, University of Jember.

Converting electrical energy into mechanical energy is a electric motor function. Household appliances like a fan, washing machines, water pumps and vacuum cleaners is a example use of the motor. Electric power is converted into mechanical power in electric motors. Universal motor is an alternating current motor, the construction and characteristics of a universal motor are the same as a direct current motor. It is that it can be operated with an alternating voltage source or with a direct current voltage at the same voltage value one of the advantage. Universal motors that are widely used have a controlling device that functions to control the speed to be generated by the motor. Chopper is a circuit used to convert a fixed AC voltage input source into a controllable or regulated AC voltage output source.

Speed of motor universal can be controlled by AC Chopper circuit. It will control the speed by changing the value of the input voltage used. Basic AC Chopper type using 4 diodes of the super-fast recovery rectifier diode and combined with the IRF740 type MOSFET which functions as a PWM signal switching that will control the change in the chopping voltage. Fuzzy Logic Controller is combined in AC Chopper circuit as a universal motor driver to control the speed at a stability of 1000 RPM, and uses the LM393 Optical Encoder sensor that will give a feedback signal to the fuzzy system as input. The method of fuzzy is using the MISO (Multiple Input Single Output) system with the input error and Derror values, where the error value is obtained from the difference between the sensor reading value and the desired speed value, while the Derror value is obtained from the difference between the current error value and the previous error. Five times tested to determine the error and the amount of the voltage chopping value generated by the AC Chopper appliance. There is also controlled manual test with a load to determine the

efficiency of the driver device, as well as testing at no load and loaded using the Fuzzy Logic Controller. The load used is a permanent magnet synchron generator.

In the results of the tests that have been carried out obtained the smallest error value of 0.13% and the largest error value of 81.16% with an average error of the AC Chopper device of 38.7% and an average value of voltage trimming of 17.74 V at dutycycle range of 7.8% - 94% during the first test with 5 different types of input voltage values. With a permanent magnet synchron generator load has an average value of 86.96% with a dutycycle value range of 52% to 80% that was efficiency value resulting from controlled manual test. Whole system test are using input voltage value at the driver 63.57 V and the dutycycle setpoint value of 50%, two types of tests were carried out, no load and no load. Motor speed can be stable in the range of 1000 RPM with a dutycycle of 7% - 10% with an output voltage value of 44.26 V to 44.59 V with a no-load current of 1.5 mA when doing test without load. Meanwhile, when testing the conditions given a permanent magnet synchron generator load, the motor speed can be stable in the range of 1000 RPM with a dutycycle of 85% to 86% with an output voltage value of 60.02 V to 60.51 V and a current value of 3.02 mA.

Results of the comparison of the test periods when without using a load using a load, namely the time to reach the 1000 RPM value produced by a universal motor when conditions do not use a load is faster than when using a load.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun AC-AC Chopper sebagai Driver Motor Universal Berbasis *Fuzzy Logic Controller*”. Skripsi ini disusun guna memenuhi salah satu syarat menuntaskan 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. Allah SWT, atas kuasa-Nya penulis dapat menuntaskan skripsi ini.
2. Nabi Muhammad SAW, Nabi junjungan umat Islam yang telah menarik kami dari zaman jahiliyah ke zaman yang lebih baik.
3. Kedua orang tua, bapak drh. Mohammad Amir Roeslan dan ibu Fabtimawati yang membantu baik secara moral, materil serta doa yang tidak pernah putus.
4. Kakak dan adik kandung penulis, Amfeby Ferdiansyah, S.Kom., Ismed Miftahus Salem, dan Bobby yang mewarnai suasana ketika penulis di rumah.
5. Bapak Andi Setiawan, S.T., M.T. dan bapak Dr. Ir. Widjonarko, S.T., M.T. selaku pembimbing serta bapak Ir. Widyono Hadi, M.T. dan bapak Guido Dias Kalandro, S. ST., M.Eng. selaku penguji yang banyak memberi ilmu berharga.
6. Pengurus HME periode 2017-2019, seluruh panitia JLT 8, PLP dan Asisten Laboratorium Komputer dan Multimedia yang telah memberi penulis banyak pelajaran berharga yang sangat bermanfaat.
7. Dulur INDUKTRO, Kontraan C2/25, dan Pertamina yang banyak membantu dalam berproses sedari maba hingga penyelesaian skripsi.
8. Sahabat saya OHCAM yang selalu memberi support ketika di Kamal.
9. Alif Willis Nurmalia, S.T. yang selalu siap membantu dan mendukung penuh dalam proses penyelesaian skripsi.

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

Jember, 22 Januari 2021.

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA	xii
DAFTAR ISI.....	xiii
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xvi
BAB 1. PENDAHULUAN	18
1.1 Latar Belakang	18
1.2 Rumusan Masalah	20
1.3 Batasan Masalah.....	20
1.4 Tujuan	21
1.5 Manfaat	21
1.6 Sistematika Penulisan	22
BAB 2. TINJAUAN PUSTAKA.....	23
2.1 Penelitian Terkait	23
2.2 Driver Motor Universal	24
2.2.1 Rangkaian AC Chopper	24
2.2.2 Bootstrap MOSFET	25
2.3 Motor Universal.....	26
2.4 Pengaturan Kecepatan Motor Universal	29
2.5 Arduino.....	30
2.5.1 Arduino Uno R3.....	31
2.5.2 IDE Arduino.....	33
2.6 Sensor Optical Encoder LM393.....	34

2.7	<i>Fuzzy Logic Control</i>	35
2.7.1	<i>Rule Evaluation</i>	35
2.7.2	Pengertian <i>Fuzzy Logic</i>	36
2.7.3	Himpunan Fuzzy	38
2.7.4	Fungsi Keanggotaan	40
2.7.5	Tahap Pemodelan dalam <i>Fuzzy Logic</i>	41
2.7.6	Fuzzyfikasi	42
2.7.7	Defuzzyfikasi	43
BAB 3.	METODOLOGI PENELITIAN	44
3.1	Tempat dan Waktu Penelitian	44
3.1.1	Tempat Penelitian	44
3.1.2	Waktu Penelitian	44
3.2	Metode Penelitian	45
3.3	Definisi Variabel Penelitian	45
3.4	Jenis dan Pengumpulan Data	46
3.5	Kerangka Pemecahan Masalah	47
3.6	Alat dan Bahan	48
3.7	Rancangan Penelitian	49
3.8	Blok Diagram dan Perancangan Sistem	49
3.9	Desain Alat	50
3.10	Alur Perancangan <i>Driver</i> Motor Universal	51
3.10.1	Pembuatan Rangkaian <i>AC Chopper</i>	51
3.10.2	Pembuatan Rangkaian <i>Bootstrap driver</i> MOSFET	52
3.10.3	Pemasangan Sensor <i>Optical Encoder</i> LM393	52
3.10.4	Pemodelan Kontrol Fuzzy	53
BAB 4.	ANALISIS HASIL DAN PEMBAHASAN	56
4.1	Pengujian Rangkaian <i>Bootstrap</i> MOSFET	56
4.2	Pengujian Rangkaian <i>AC Chopper</i>	59
4.3	Kalibrasi dan Pengujian Sensor	62
4.4	Desain <i>Fuzzy Logic Controller</i>	65
4.5	Analisis Pengujian <i>AC Chopper</i> Kontrol Manual	69

4.6 Analisis dan Pengujian Sistem Secara Keseluruhan.....	73
4.6.1 Pengujian <i>Driver</i> Motor Universal Tanpa Beban	74
4.6.2 Pengujian <i>Driver</i> Motor Universal Berbeban	77
4.6.3 Analisis dan Pengujian <i>Driver</i> Motor Keseluruhan.....	79
BAB 5. PENUTUP.....	85
5.1 Kesimpulan	85
5.2 Saran.....	85
DAFTAR PUSTAKA	86
LAMPIRAN.....	88

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino UNO	31
Tabel 3.1 Rencana Kegiatan Penelitian	44
Tabel 3.2 Spesifikasi Sensor <i>Optical Encoder LM393</i>	53
Tabel 4.1 Data Pengujian Rangkaian <i>Bootstrap</i> MOSFET	56
Tabel 4.2 Pengujian 1 Rangkaian <i>AC Chopper</i>	60
Tabel 4.3 Pengujian 2 Rangkaian <i>AC Chopper</i>	60
Tabel 4.4 Pengujian 3 Rangkaian <i>AC Chopper</i>	60
Tabel 4.5 Pengujian 4 Rangkaian <i>AC Chopper</i>	60
Tabel 4.6 Pengujian 5 Rangkaian <i>AC Chopper</i>	60
Tabel 4.7 Kalibrasi Sensor <i>Optical Encoder LM393</i>	63
Tabel 4.8 Pengujian <i>AC Chopper</i> Tanpa Beban	69
Tabel 4.9 Pengujian <i>AC Chopper</i> Berbeban	70
Tabel 4.10 Pengujian Efisiensi <i>AC Chopper</i>	71
Tabel 4.11 Pengujian <i>Driver</i> Motor Universal Tanpa Beban	74
Tabel 4.12 Pengujian <i>Driver</i> Motor Universal Berbeban	77
Tabel 4.13 Pengujian dengan kondisi Generator Berbeban	81

DAFTAR GAMBAR

Gambar 2.1 Hasil pengujian penelitian sebelumnya.....	23
Gambar 2.2 <i>Single – phase AC-AC voltage controller</i>	24
Gambar 2.3 Proses Gelombang <i>AC Chopper</i>	25
Gambar 2.5 Rangkaian <i>Bootstrap Input High and Low</i>	26
Gambar 2.6 Konstruksi Motor Universal.....	27
Gambar 2.7 Prinsip Kerja Motor Universal	27
Gambar 2.8 Rangkaian Ekuivalen Motor Universal	27
Gambar 2.9 Karakteristik Motor Universal	28
Gambar 2.10 Karakteristik Pengaturan Kecepatan Terhadap Perubahan Frekuensi	30
Gambar 2.11 Karakteristik Perubahan Tegangan Terhadap Kecepatan Motor Universal	30
Gambar 2.12 Arduino Uno.....	32
Gambar 2.13 Tampilan <i>Software IDE</i> Arduino	33
Gambar 2.14 Cara Kerja Sensor <i>Optical Encoder LM393</i>	34
Gambar 2.15 Sensor <i>Optical Encoder LM393</i>	35
Gambar 2.16 Konsep Dasar Logika Fuzzy	37
Gambar 2.17 Blok diagram Kontrol Fuzzy.....	38
Gambar 2.18 Blok Diagram Sistem Fuzzy 1	41
Gambar 2.19 Blok Diagram Sistem Fuzzy 2	42
Gambar 2.20 <i>Fuzzyfication</i>	43
Gambar 2.21 <i>Defuzzyfication</i>	43
Gambar 3.1 <i>Flowchart</i> Penelitian	46
Gambar 3.2 Blok Diagram Sistem	50
Gambar 3.3 Desain Rangkaian <i>Driver</i> Motor 1 Fasa.....	50
Gambar 3.4 <i>Schematic</i> Rangkaian <i>driver</i> Motor Universal.....	51
Gambar 3.5 <i>Schematic</i> Rangkaian <i>Bootstrap</i> MOSFET.....	52
Gambar 3.6 <i>Schematic</i> Rangkaian Sensor <i>Optical Encoder LM393</i>	53
Tabel 3.7 <i>Rule Base</i>	55
Gambar 4.1 PWM dengan nilai <i>dutycycle</i> 7%	56
Gambar 4.2 PWM dengan nilai <i>dutycycle</i> 39%	57
Gambar 4.3 PWM dengan nilai <i>dutycycle</i> 39%	57
Gambar 4.4 PWM dengan nilai <i>dutycycle</i> 78%	57
Gambar 4.5 PWM dengan nilai <i>dutycycle</i> 94%	58
Gambar 4.6 <i>Bootstrap</i> MOSFET	59
Gambar 4.7 Grafik Pengaruh <i>Dutycycle</i> Terhadap Tegangan <i>Output</i>	61
Gambar 4.8 Pengujian <i>AC Chopper</i>	62
Gambar 4.9 Grafik Perbandingan Pembacaan Sensor dan <i>Tachometer</i>	64

Gambar 4.10 Sensor <i>Optical Encoder</i> LM393	64
Gambar 4.11 FIS Editor untuk Metode Mamdani	65
Gambar 4.12 Fungsi Keanggotaan <i>Input error</i>	66
Gambar 4.13 Fungsi Keanggotaan <i>Input Derror</i>	66
Gambar 4.14 Fungsi Keanggotaan <i>Output Derror</i>	67
Gambar 4.15 <i>Rule Base Fuzzy Logic Controller</i>	68
Gambar 4.16 Grafik Kecepatan Motor Universal Kontrol Manual	72
Gambar 4.17 Grafik Pengaruh <i>Dutycycle</i> terhadap Daya	72
Gambar 4.18 <i>Driver</i> Motor Universal.....	74
Gambar 4.19 Pengujian Sistem Secara Keseluruhan	74
Gambar 4.20 Grafik Perbandingan Motor Universal dengan Kontrol dan Kontrol Manual.....	76
Gambar 4.21 Grafik Perbandingan Motor Universal dengan Kontrol dan Kontrol Manual.....	79
Gambar 4.22 Grafik Perbandingan Kecepatan Motor Universal	79
Gambar 4.23 Grafik Pengujian <i>Driver</i> Motor Universal	84

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Motor listrik adalah alat untuk mengubah energi listrik menjadi energi mekanik. Alat yang berfungsi sebaliknya, mengubah energi mekanik menjadi energi listrik disebut generator atau dinamo. Motor listrik dapat ditemukan pada peralatan rumah tangga seperti kipas angin, mesin cuci, pompa air dan penyedot debu. Pada motor listrik tenaga listrik diubah menjadi tenaga mekanik. Perubahan ini dilakukan dengan mengubah tenaga listrik menjadi magnet yang disebut sebagai elektromagnet. Sebagaimana kita ketahui bahwa kutub-kutub dari magnet yang senama akan tolak-menolak dan kutub-kutub tidak senama, tarik-menarik (Bahariawan, 2018).

Maka kita dapat memperoleh gerakan jika kita menempatkan sebuah magnet pada sebuah poros yang dapat berputar, dan magnet yang lain pada suatu kedudukan yang tetap. Berdasarkan suplaynya motor dibagi menjadi dua yaitu motor suplai AC dan suplai DC (Prihanto, 2019). Motor universal adalah motor arus bolak balik, konstruksi maupun karakteristik motor universal sama dengan motor arus searah. Keuntungan motor universal ini dapat dioperasikan dengan sumber tegangan bolak balik atau dengan tegangan arus searah pada nilai tegangan yang sama (Tomy, 2019). Motor universal banyak digunakan sebagai tenaga penggerak pada peralatan listrik portable seperti gerinda, mixer, blender dimana pengaturan kecepatannya umumnya memiliki jangkah yang terbatas dan tidak halus. Tujuan penelitian ini adalah mengembangkan system pengaturan kecepatan motor universal yang mampu memberikan jangkah pengaturan lebih lebar dan halus (Nurhayata, 2015).

Motor universal yang banyak digunakan memiliki perangkat pengendali yang berfungsi untuk mengendalikan kecepatan yang akan dihasilkan oleh motor. *AC Chopper* (pemangkas) merupakan suatu rangkaian yang digunakan untuk mengubah sumber masukan tegangan AC tetap menjadi sumber keluaran tegangan AC yang dapat dikendalikan atau diatur. Komponen semikonduktor daya yang

digunakan dapat berupa SCR, transistor, dan MOSFET yang beroperasi sebagai sakelar dan pengatur. Ditinjau dari proses pengaturan menurut menyatakan bahwa: chopper dapat dibedakan dalam tiga jenis, yaitu : chopper penurun tegangan (step-down), chopper penaik tegangan (step-up), dan chopper penaik penurun tegangan (Nazaruddin, 2015). Rangkaian *AC Chopper* menjadi salah satu piranti yang dapat mengedalikan kecepatan yang dihasilkan oleh motor. Piranti ini akan mengendalikan kecepatan dengan cara merubah nilai tegangan *input* yang digunakan. (Elsevier, 2007) Pengendali yang baik akan mempengaruhi efisiensi yang dihasilkan dari motor universal satu fasa yang digunakan. (Sudibyo., Teknosia Vol. II). Untuk sebuah kontrol yang baik diperlukan sebuah sensor pada motor yang *outputnya* menjadi umpan balik untuk mengetahui kecepatan aktual yang dihasilkan motor universal tersebut.

Salah satu metode kendali umpan balik yang umum digunakan adalah metode kendali *Fuzzy Logic*. Dalam penelitian ini metode kendali *Fuzzy* dipilih untuk mendapatkan pengaturan kecepatan motor yang baik. Dengan sensor LM393 *Optical Encoder* sebagai umpan balik ke *driver* yang akan membaca kecepatan motor yang dihasilkan dan akan dapat disesuaikan oleh kendali *Fuzzy* dengan memetakan suatu ruang *input* yang telah ditentukan menjadi sebuah *output*.

Pada penelitian sebelumnya oleh P. Sravan Kumar, B. Mahendar, dan M. Shruthi yang berjudul “*Design and Implementation of AC Chopper*”. Pada penelitian ini bertujuan untuk merancang dan mengimplementasikan rangkaian *AC Chopper basic* dan *synchronous* menggunakan *software* MATLAB untuk menghasilkan perpotongan tegangan berdasarkan sudut penyalan seperti halnya dimmer. Menggunakan empat buah dioda rectifier yang terhubung pada sebuah IGBT untuk rangkaian *AC Chopper basic* yang tidak mempengaruhi frekuensi yang dihasilkan. (Kumar, 2014).

Sedangkan pada penelitian Rancang Bangun *AC-AC Chopper* Sebagai *Driver* Motor Universal Berbasis *Fuzzy Logic Controller* menggunakan MOSFET sebagai pemicu sinyal dan mengimplementasikannya untuk sebuah *driver* motor universal serta menambahkan logika fuzzy sebagai umpan balik untuk membuat kecepatan motor tetap stabil.

1.2 Rumusan Masalah

Dari uraian singkat dan latar belakang, maka diperoleh rumusan masalah sebagai berikut :

1. Bagaimana merancang sistem kendali kecepatan motor universal yang berbasis pada piranti *AC Chopper* untuk mengatur tegangan *input* motor yang dikendalikan dengan PWM dengan metode kendali umpan balik *Fuzzy Logic Controller* ?
2. Berapa rata-rata efisiensi *AC Chopper* dengan kendali umpan balik fuzzy pada pengaturan kecepatan yang sesuai dengan kecepatan tertentu ?
3. Bagaimana mengimplementasikan rancangan sistem kendali ini untuk mengatur kecepatan motor universal dengan beban yang ditentukan ?

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, maka batasan masalah pada penelitian ini sebagai berikut :

1. Kendali kecepatan ini akan diaplikasikan untuk pengaturan kecepatan motor universal.
2. Beban motor universal adalah sebuah *permanent magnet synchronous generator* yang bekerja dengan frekuensi 50 Hz.
3. Pengaturan kecepatan menggunakan piranti *AC Chopper* dan sebuah *driver MOSFET* untuk mengatur tegangan *input* pada motor universal.
4. Kendali *Fuzzy Logic* diimplementasikan dengan menggunakan Arduino sebagai mikrokontroler.
5. Pengendalian tegangan *output* pada piranti *AC Chopper* melalui PWM yang terhubung pada Arduino.

1.4 Tujuan

Dari persamaan masalah diatas, tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang sistem kendali kecepatan motor universal yang berbasis pada piranti *AC Chopper* untuk mengatur tegangan *input* motor yang dikendalikan dengan PWM dengan metode kendali umpan balik *Fuzzy*.
2. Memilih variable *Fuzzy* yang sesuai untuk mendapatkan pengendalian kecepatan dengan *range* tertentu.
3. Mengimplementasikan rancangan sistem kendali ini untuk mengatur kecepatan motor universal dengan beban yang sudah ditentukan dengan mikrokontroler Arduino

1.5 Manfaat

Adapun manfaat yang diharapkan dapat tercapai, dalam penelitian ini adalah sebagai berikut :

1. Bagi masyarakat secara umum, hasil kegiatan penelitian ini dapat menjadi alternatif sistem pengendalian kecepatan motor universal satu fasa yang secara umum dapat diterapkan pada banyak sistem yang lain.
2. Untuk dunia penelitian dan pendidikan, hasil kegiatan penelitian ini dapat menjadi referensi untuk penelitian selanjutnya.
3. Untuk mahasiswa sebagai peneliti dan mahasiswa pada umumnya, hasil penelitian ini dapat menjadi acuan pembelajaran dan pengembangan mengenai *driver* motor universal.

1.6 Sistematika Penulisan

Secara garis besar penyusunan skripsi ini adalah sebagai berikut:

a. **BAB 1. PENDAHULUAN**

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penelitian.

b. **BAB 2. TINJAUAN PUSTAKA**

Berisi tentang teori yang berhubungan dengan penelitian.

c. **BAB 3 METODE PENELITIAN**

Menjelaskan tentang metode kajian yang digunakan untuk menyelesaikan skripsi

d. **BAB 4. ANALISIS HASIL DAN PEMBAHASAN**

Berisi hasil penelitian berupa data dan analisis hasil penelitian.

e. **BAB 5. PENUTUP**

Berisi tentang kesimpulan dan saran dari penulis.

BAB 2. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian ini merupakan pengembangan lebih lanjut dari penelitian yang berjudul “*Design and Implementation of AC Chopper*” oleh P. Sravan Kumar, B. Mahendar, dan M. Shruthi pada tahun 2014 . Pada penelitian ini bertujuan untuk merancang dan mengimplementasikan rangkaian *AC Chopper basic* dan *synchronous* menggunakan *software* MATLAB untuk menghasilkan perpotongan tegangan berdasarkan sudut penyalan seperti halnya rangkaian dimmer yang kemudian bisa diterapkan untuk pengaturan tegangan. Menggunakan empat buah dioda rectifier yang terhubung pada sebuah IGBT untuk rangkaian *AC Chopper basic* yang tidak mempengaruhi frekuensi yang dihasilkan. Tegangan yang digunakan pada penelitian ini yaitu 100V AC dengan frekuensi 50Hz.

Firing Angle	Theoretical Output Voltage	Practical Output Voltage	Practical Output Current	Theoretical Output Current	Practical Output Power	Theoretical Output Power
36 deg	84.458 V	80 V	0.090 A	0.0854 A	7.2 W	7.21W
45 deg	77.099 V	70 V	0.085 A	0.0779 A	5.95 W	5.93W
60 deg	64.867 V	55 V	0.060 A	0.0648 A	3.30W	4.20W
75 deg	45.485 V	40 V	0.040 A	0.0454 A	1.60 W	2.04W
89 deg	11.550 V	10 V	0.020 A	0.0115 A	0.20 W	0.12W

Gambar 2.1 Hasil pengujian penelitian sebelumnya (Kumar, 2014)

Pada gambar 2.1 menunjukkan hasil pengujian yang membuktikan bahwa *inverter AC-AC Chopper* dapat diimplementasikan sebagai perangkat untuk mengendalikan tegangan, dimana pada penelitian tersebut menggunakan 100V AC dengan frekuensi 50Hz memiliki hasil *practical output voltage* yang hampir sama dengan *theoretical output voltage* dengan melakukan pengendalian sudut penyalan melalui IGBT yang digunakan.

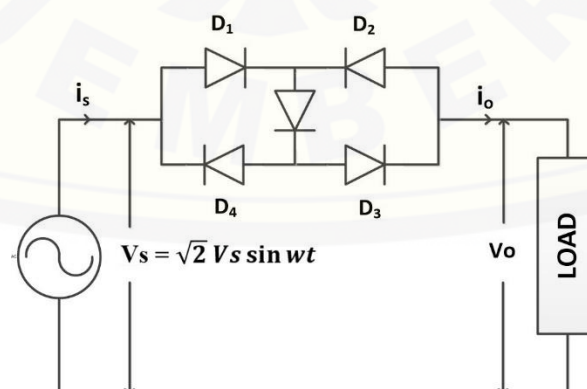
2.2 Driver Motor Universal

Pada sebuah rangkaian *driver* motor universal memiliki beberapa piranti yang terintegrasi menjadi satu rangkaian pengendali (*driver*) motor universal satu fasa. Adapun piranti – piranti tersebut yaitu :

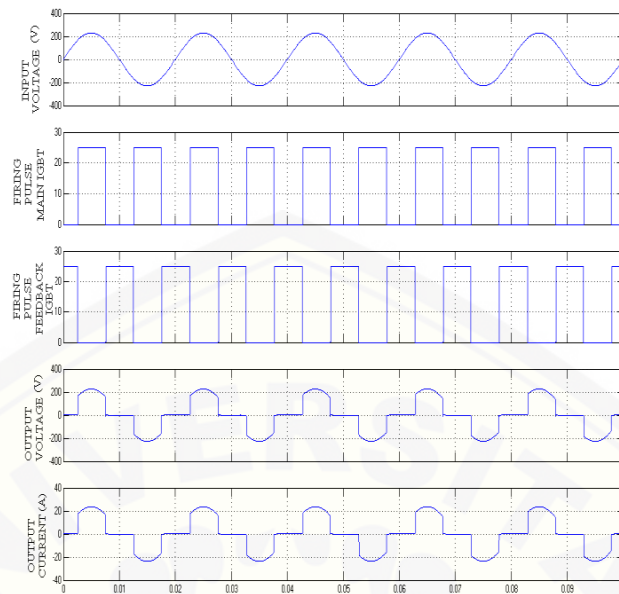
2.2.1 Rangkaian AC Chopper

AC Chopper adalah suatu rangkaian elektronika daya yang memiliki fungsi untuk Mengoversi AC ke AC yang menggunakan variasi dari tegangan untuk melintasi beban dengan frekuensi tetap. Pada phase control rangkaian ini menggunakan dua buah silicon controlled rectifier (SCR) atau TRIAC. Sedangkan untuk on-off control menggunakan switch dengan kendali penuh seperti Gate Turn-off Thyristor, Power Transistor, Insulate Gate Bipolar Transistor, dan MOS-Controlled Thyristor. (Dr. Gamal Sowilam, 2016).

Gambar Rangkaian Single – phase AC-AC voltage controller :



Gambar 2.2 Single – phase AC-AC voltage controller



Gambar 2.3 Proses Gelombang AC Chopper

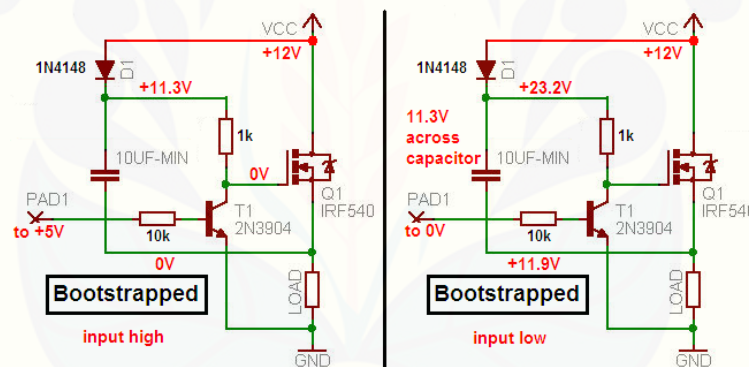
Rangkaian AC Chopper terdiri dari SCR yang terkoneksi dan juga empat dioda dan terhubung pada beban dengan sumber AC 220 volt. Terdapat juga controller AC-AC dengan dua SCR yang terkoneksi Back-to-Back dan beroperasi sebagai full wave dan half wave controller. Dalam metode ini switch AC Chopper digunakan untuk menghubungkan sumber ke rangkaian beban dan dipilih bagian dari setiap siklus tegangan sumber.

Kendali AC Chopper untuk MOSFET merupakan suatu kendali yang dirancang khusus untuk MOSFET, dengan *output* yang berupa pulsa PWM yang di salurkan ke logical gate MOSFET. Rangkaian pengendali ini memiliki dua *output* dengan phase delay 180°. Desain dari rangkaian ini terdiri dari logical gate dan monostable vibrator.

2.2.2 Bootstrap MOSFET

Rangkaian gate *driver* berfungsi sebagai penghubung antara sinyal pengendali tegangan titik G terhadap S pada MOSFET. Tegangan tersebut bernama voltage gate to source (VGS). Secara umum metode yang sering digunakan adalah metode bootstrap.

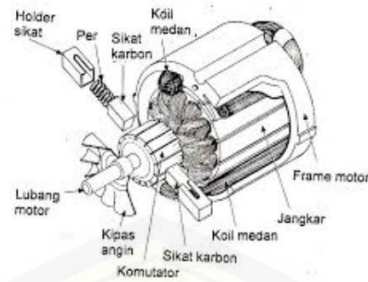
Gate *driver* juga berfungsi untuk menerjemahkan sinyal bertegangan rendah dari sumber sinyal diskrit untuk diumpankan ke MOSFET sebagai saklar daya sehingga dapat beroperasi sesuai parameternya. Ketersediaan tegangan tersebut dipengaruhi oleh proses pengisian kapasitor bootstrap. Kapasitor backup menjaga agar proses pengisian muatan kapasitor bootstrap tidak mengganggu proses pensaklaran MOSFET utama. Sinyal diskrit dari mikroprosesor ataupun sinyal yang masuk dari sumber dapat diproses seutuhnya tanpa terganggu proses pengisian kapasitor seperti pada conventional bootstrap. Dari hasil pengujian tersebut, electromagnetic bootstrap mampu memberikan sinyal kendali ke gate yang sesuai dengan kebutuhan MOSFET yaitu 12-15V dengan referensi tegangan gate terhadap source serta mampu menjaga ketersediaan tegangan yang digunakan untuk melakukan proses pensaklaran pada MOSFET utama. (Pramono, April, 2018)



Gambar 2.4 Rangkaian Bootstrap *Input High and Low*

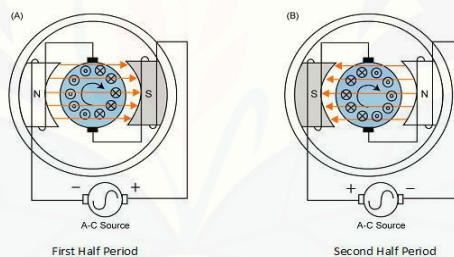
2.3 Motor Universal

Motor universal sesuai dengan namanya adalah salah satu motor listrik yang bekerja pada sumber tegangan listrik arus bolak-balik (AC) dan pada arus searah (DC) karena memiliki konstruksi yang sama dengan motor dc seri (Nurhayata, 2015). Adapun konstruksi dari motor universal seperti tampak pada gambar 2.6.



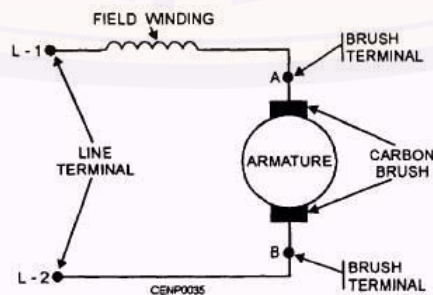
Gambar 2.5 Konstruksi Motor Universal

Prinsip kerja motor universal adalah ketika sumber tegangan bolak-balik membangkitkan medan magnet bolak-balik pada inti besi stator yang berubah arah sesuai dengan polaritas sumber tegangan. Pada kumparan rotor mengalir arus dan membangkitkan medan magnet rotor yang juga berubah sesuai dengan perubahan arah polaritas sumber. Interaksi kedua medan magnet tersebut menyebabkan timbul gaya gerak yang mengakibatkan rotor berputar (Nurhayata, 2015).



Gambar 2.6 Prinsip Kerja Motor Universal

Konstruksi motor universal dapat dijelaskan dalam bentuk rangkaian ekivalen seperti pada gambar 2.8.

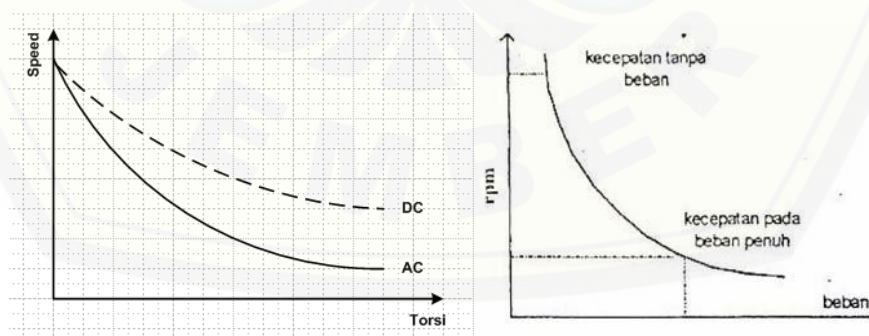


Gambar 2.7 Rangkaian Ekivalen Motor Universal

Berdasarkan rangkaian tersebut maka kecepatan putaran motor universal dipengaruhi oleh besarnya tegangan sumber V_s dan arus jangkar I_a . Besarnya arus jangkar dipengaruhi oleh faktor pembebanan atau torsi beban sehingga jika semakin besar beban maka kecepatannya akan menurun seperti pada gambar 2.7. Pada gambar terlihat bahwa kecepatan motor universal merosot tajam seiring dengan perubahan beban (Nurhayata, 2015).

Motor universal mempunyai karakteristik motor seri karena berputar pada kecepatan rata-rata bila bebannya juga rata-rata, dan apabila bebannya dikurangi maka kecepatannya akan naik. Motor ini mempunyai sifat-sifat yang sama seperti motor DC seri. Pada pembebanan ringan motor berputar dengan cepat dan menghasilkan kopel yang kecil. Tetapi pada keadaan pembebanan yang berat, maka motornya berputar secara perlahan-lahan dengan torsi yang besar. Jadi, motor mengatur kecepatannya sesuai dengan beban yang dihubungkan kepadanya. Motor jenis ini banyak ditemui antara lain pada: blower, dinamo mesin jahit rumah, mesin bor, dan mixer (Cyneand Martin, 1987) (Adam, 2011).

Putaran motor universal biasanya tinggi, apalagi dalam keadaan tanpa beban. Maka dari itu, biasanya motor universal dihubungkan langsung dengan beban sehingga putaran motor yang tinggi bisa berkurang dengan pembebanan tersebut.



Gambar 2.8 Karakteristik Motor Universal

Karakteristik motor universal AC dan DC cukup berbeda karena dua alasan :

1. Pada motor universal dengan sumber tegangan AC, tegangan reaktansi akan timbul pada rangkaian AC ketika sebuah magnet dibentuk oleh arus yang mengalir dalam rangkaian listrik. Tegangan reaktansi ini menyerap sebagian dari tegangan jala-jala dan akan mengurangi tegangan yang diaplikasikan ke jangkar sehingga mengurangi kecepatan motor.
2. Dengan sumber tegangan AC, rangkaian magnetis menjadi cukup jenuh pada puncak gelombang arus, dan nilai rms fluks menjadi lebih kecil dibandingkan dengan sumber tegangan DC. Pada keadaan tanpa beban dengan nilai rms yang sama, torsi cenderung lebih kecil dan kecepatannya lebih tinggi dibandingkan dengan sumber tegangan DC.

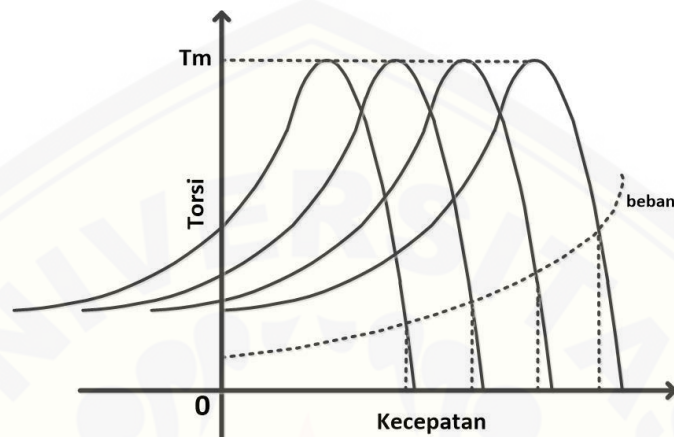
Pengaturan kecepatan motor universal dapat dilakukan dengan mengatur variabel tegangan sumber V_s . Dari persamaan kecepatan motor diatas, dapat diketahui semakin besar tegangan sumber maka kecepatan motor universal akan semakin meningkat demikian pula sebaliknya. Dengan demikian pengaturan kecepatan motor universal hanya dapat dilakukan dengan mengatur tegangan sumber yang menuju ke motor universal (Nurhayata, 2015).

2.4 Pengaturan Kecepatan Motor Universal

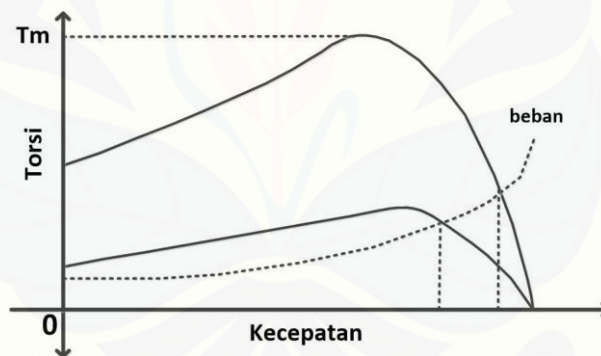
Pengaturan kecepatan motor universal dapat dilakukan dengan:

1. Variasi Frekuensi Stator
Pengaturan putaran motor universal dapat dilakukan dengan mengatur besar frekuensi yang dimiliki oleh motor universal tersebut. Frekuensi berbanding lurus dengan kecepatan motor yang akan dihasilkan. Frekuensi juga berkaitan dengan torka akibat adanya beban yang diperlihatkan pada gambar 2.10.
2. Mengatur Tegangan Terminal
Pengaturan putaran motor universal juga dapat dilakukan dengan mengubah tegangan terminal. Semakin besar nilai tegangan pada motor, maka kecepatan motor akan semakin besar seperti pada gambar 2.11. Semakin besar beban yang diberikan pada motor mengakibatkan tegangan pada motor akan semakin besar

mengikuti batas kecepatan motor. Tegangan pada motor universal berpengaruh karena perubahan frekuensi mengakibatkan perubahan tegangan yang mengikuti.



Gambar 2.9 Karakteristik Pengaturan Kecepatan Terhadap Perubahan Frekuensi



Gambar 2.10 Karakteristik Perubahan Tegangan Terhadap Kecepatan Motor Universal

2.5 Arduino

Arduino merupakan mikrokontroler yang dapat disematkan sebuah program di dalamnya, Arduino memiliki perangkat keras (*hardware*) yang memiliki banyak jenis salah satunya Arduino Uno R3 dan memiliki sebuah perangkat lunak (*software*) untuk membuat program dengan Bahasa C yang bernama Arduino IDE.

2.5.1 Arduino Uno R3

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI *driver* USB-to-serial. (eleraning.me, 2020).

Tabel 2.1 Spesifikasi Arduino UNO

Mikrokontroler	ATmega328
Tegangan Operasi	5 Volt
Input Voltage (disarankan)	7 - 12 Volt
Input Voltage (batas akhir)	6 - 20 Volt
Digital I/O Pin	14 (6 pin sebagai output PWM)
Analog Input Pin	6
Arus DC per pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	32 KB (ATmega328) 0,5 KB untuk bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 Hz

Masing-masing dari 14 pin digital arduino UNO dapat digunakan sebagai masukan atau keluaran menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor pull-up internal (diputus secara default) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukandigital memiliki kegunaan khusus sebagai berikut:

1. Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima(RX) dan mengirim(TX) data secara serial.
2. Interupsi Eksternal: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
3. Pulse-width modulation (PWM): pin 3,5,6,9,10 dan 11, menyediakan keluaran PWM 8-bit dengan menggunakan fungsi analogWrite().
4. Serial Peripheral Interface (SPI): pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan SPI library.
5. LED: pin 13, terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai HIGH maka LED menyala, sebaliknya ketika pin bernilai LOW maka LED akan padam.

Arduino UNO memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara default pin mengukur nilai tegangan dari ground (0V) hingga 5V, walaupun dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi analogReference(). Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi Two Wire Interface (TWI) atau Inter Integrated Circuit (I2C) dengan menggunakan Wire library(Arduino).

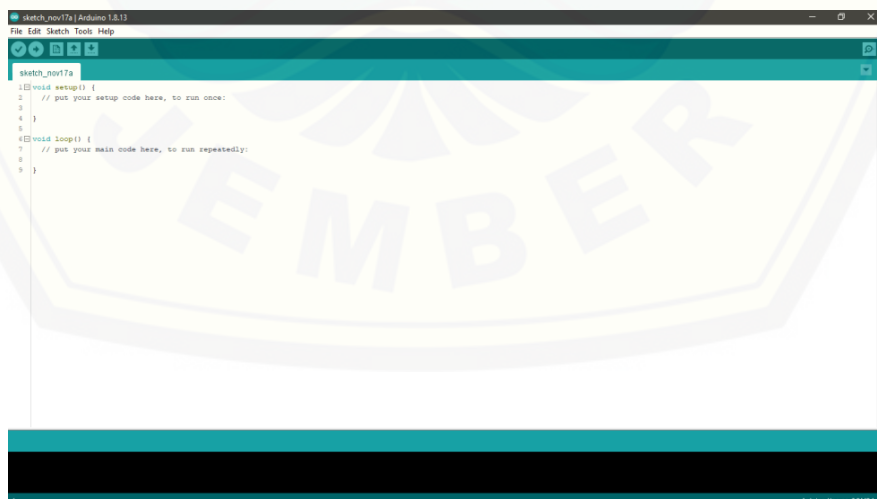


Gambar 2.11 Arduino Uno

2.5.2 IDE Arduino

Software Arduino yang akan digunakan adalah driver dan IDE. IDE diciptakan untuk para pemula bahkan yang tidak memiliki basic Bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui library. IDE Arduino adalah software canggih yang ditulis dengan menggunakan bahasa Java. Software IDE Arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam Bahasa processing. Listing program pada Arduino disebut sketch.
2. *Compiler*, sebuah modul yang berfungsi mengubah bahasa processing (kode program) menjadi kode biner karena kode biner merupakan bahasa program yang dipahami oleh mikrokontroler.
3. *Uploader*, sebuah modul yang berfungsi memasukkan kode biner ke dalam memori mikrokontroler. Struktur perintah pada Arduino secara garis besar terdiri dari dua bagian yaitu void setup dan void loop. Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan sedangkan void loop berisi perintah yang akan dieksekusi berulang-ulang selama Arduino dinyalakan.

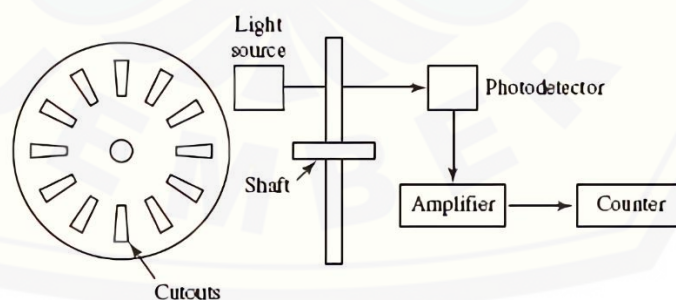


Gambar 2.12 Tampilan *Software* IDE Arduino

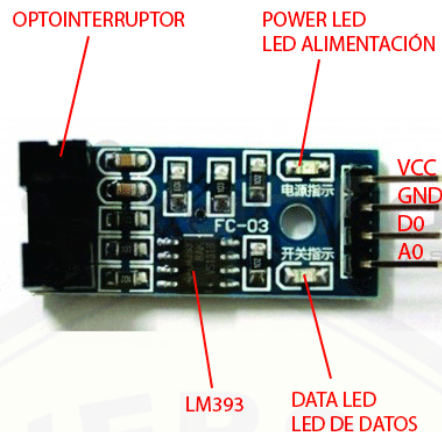
2.6 Sensor *Optical Encoder* LM393

Sensor *Optical Encoder* LM393 adalah piranti elektromekanik yang dapat memonitor gerakan dan posisi. Sensor *Optical Encoder* LM393 umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh Sensor *Optical Encoder* LM393 untuk diteruskan oleh rangkaian kendali. Sensor *Optical Encoder* LM393 umumnya digunakan pada pengendalian robot ataupun motor drive.

Prinsip kerja secara umum dari Sensor *Optical Encoder* LM393, dimana sebuah piringan tipis dan LED yang ditempatkan sedemikian rupa sehingga cahayanya tetap terfokus pada piringan tersebut. Sebuah transistor aktif cahaya ditempatkan pada sisi lain dari piringan sehingga dapat mendeteksi cahaya dari LED. Piringan tersebut ditempatkan pada poros (shaft) yang bergerak. Dimana pergerakan piringan tersebut sesuai dengan pergerakan poros (shaft) sehingga ketika poros (shaft) berputar, maka piringan ikut berputar. Ketika piringan yang di atasnya ditempatkan LED yang memancarkan cahaya yang terfokus terhadap fototransistor akan berada dalam keadaan saturasi yang keluarannya akan berupa pulsa gelombang kotak.



Gambar 2.13 Cara Kerja Sensor *Optical Encoder* LM393



Gambar 2.14 Sensor *Optical Encoder* LM393

2.7 Fuzzy Logic Controller

Fuzzy Logic Controller adalah sebuah sistem yang dapat dirancang untuk mengatur tegangan, kecepatan, hingga daya melalui PWM yang disematkan pada sebuah program pada mikrokontroler Arduino Uno. Terdapat beberapa proses dalam perancangan sebuah kontrol fuzzy antara lain :

2.7.1 Rule Evaluation

Prosedur fuzzifikasi adalah proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi 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* akan didapatkan derajat keanggotaan bagi masing – masing nilai *error* dan *delta error*. (Helmy Mukti Himawan, 2016).

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, antara lain:

1. Model Sugeno
2. Model Mamdani

2.7.2 Pengertian *Fuzzy Logic*

Dalam bahasa Inggris, fuzzy mempunyai arti kabur atau tidak jelas. Jadi, logika fuzzy 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 logika fuzzy mengenal nilai antara benar dan salah. Kebenaran dalam logika fuzzy dapat dinyatakan dalam derajat kebenaran yang nilainya antara 0 sampai 1. (Institut Teknologi Bandung, 2009).

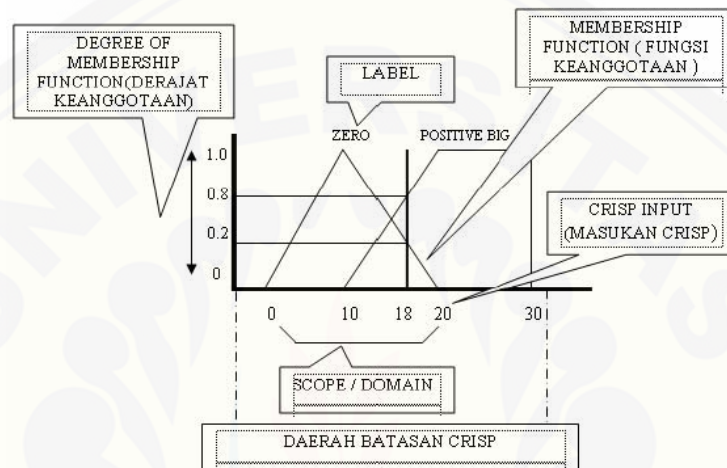
Logika fuzzy diperkenalkan pertama kali pada tahun 1965 oleh Prof. Lutfi A. Zadeh seorang peneliti di Universitas California di Berkeley dalam bidang ilmu komputer. Professor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah logika fuzzy yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika fuzzy terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika fuzzy, keanggotaan elemen berada di interval $[0,1]$.

Logika fuzzy menjadi alternatif dari berbagai sistem yang ada dalam pengambilan keputusan karena logika fuzzy mempunyai kelebihan sebagai berikut:

1. Logika fuzzy memiliki konsep yang sangat sederhana sehingga mudah untuk dimengerti.
2. Logika fuzzy sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian.
3. Logika fuzzy memiliki toleransi terhadap data yang tidak tepat.
4. Logika fuzzy mampu mensistemkan fungsi-fungsi non-linier yang sangat kompleks.
5. Logika fuzzy dapat mengaplikasikan pengalaman atau pengetahuan dari para pakar.
6. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

7. Logika fuzzy didasarkan pada bahasa sehari-hari sehingga mudah dimengerti.

Logika fuzzy memiliki beberapa komponen yang harus dipahami seperti himpunan fuzzy, fungsi keanggotaan, operator pada himpunan fuzzy, inferensi fuzzy dan defuzzifikasi. (UNY, 2011).



Gambar 2.15 Konsep Dasar Logika Fuzzy

Fungsi keanggotaan (membership function), Sudradjat [25] adalah suatu kurva yang menunjukkan pemetaan titik input data kedalam nilai keanggotaanya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. (Sudradjat, 2008).

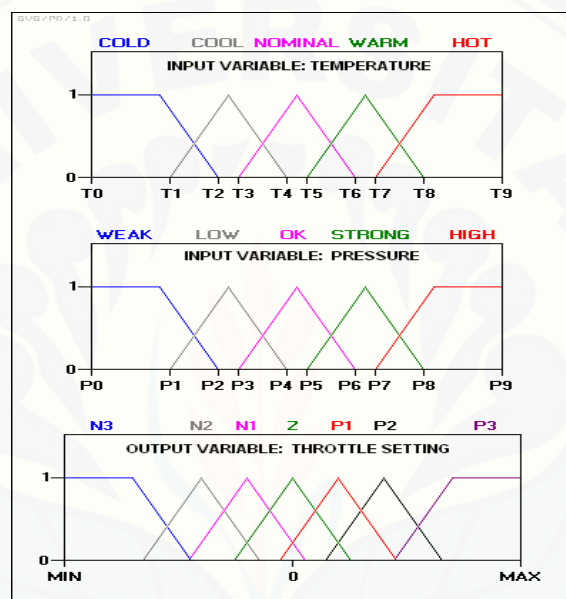
Fuzzy Logic saat ini banyak diterapkan dalam berbagai bidang, diantaranya:

1. Fuzzy rule Based Systems
2. Fuzzy Nonlinear Simulations
3. Fuzzy Decision Making
4. Fuzzy Classification
5. Fuzzy Pattern ecognition
6. Fuzzy Control Systems

(Sri Kusumadewi, Hari Purnomo, 2010)

Fuzzy Logic Control adalah suatu metode pengendali sistem berdasarkan basis pengetahuan yang dimiliki, menggunakan logika *Fuzzy* sebagai cara

pengambilan keputusan. Metode ini dapat menyesuaikan keluaran sistem berdasarkan perubahan parameter masukan akibat perubahan keadaan disekitarnya. Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Logika benar dan salah dari logika Boolean tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Logika *Fuzzy* mempunyai nilai yang kontinu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. (Sutikno, 2018).



Gambar 2.16 Blok diagram Kontrol Fuzzy

2.7.3 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki dua kemungkinan (Sri Kusumadewi, Hari 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.

Himpunan fuzzy yaitu suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Himpunan Fuzzy memiliki dua 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 variabel seperti: 40, 25, 50. Ada beberapa hal yang perlu diketahui dalam memahami system fuzzy yaitu :

a. *Variable Fuzzy*

Variable Fuzzy merupakan variable yang hendak dibahas dalam suatu system *fuzzy*. Contoh : umur, temperature, permintaan, dsb

b. Himpunan *Fuzzy*

Himpunan *Fuzzy* merupakan suatu grup yang mewakili suatu keadaan tertentu dalam suatu variable *fuzzy*.

c. Semesta Pembicaraan

Semesta Pembicaraan yaitu keseluruhan nilai yang diperoleh untuk dioperasikan dalam suatu variabel fuzzy, semesta pembicaraan merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri kekanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contohnya semesta pembicaraan untuk variabel suhu : $[0 \ 40]$.

d. Domain

Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicara 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.

(Sri Kusumadewi, 2004).

2.7.4 Fungsi Keanggotaan

Fungsi Keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam 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. (Wulandari, 2005). Ada beberapa fungsi yang bisa digunakan.

- a. Representasi Linier
- b. Representasi Kurva Trapesium
- c. Representasi Bentuk Bahu
- d. Representasi Kurva-S
- e. Representasi Kurva Bentuk Lonceng (*Bell Curve*)
- f. Koordinat Keanggotaan
- g. Representasi Linier

Operator Dasar Operasi Himpunan *Fuzzy*

3.8.4.1 Operator AND

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

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

3.8.4.2 Operator OR

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

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

3.8.4.3 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:

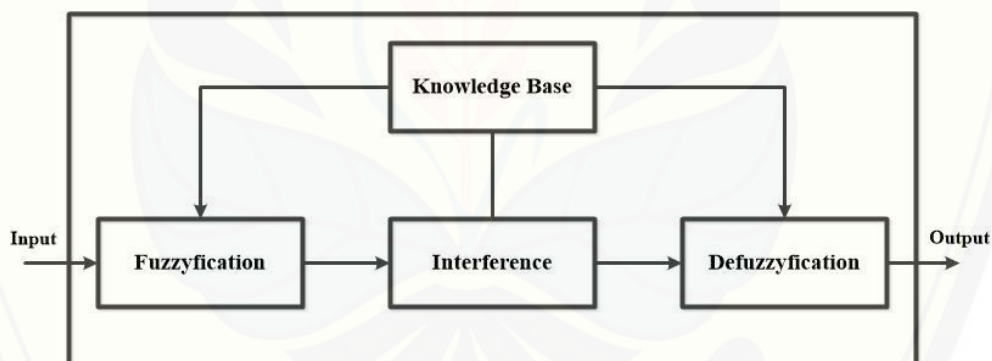
$$IF\ x\ is\ A\ THEN\ y\ is\ B$$

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. Proporsi ini dapat diperluas dengan menggunakan operator fuzzy, seperti :

$$IF(x_1\ is\ A_1) \cdot (x_2\ is\ A_2) \cdot (x_3\ is\ A_3) \dots (x_n\ is\ A_n)\ THEN\ y\ is\ B$$

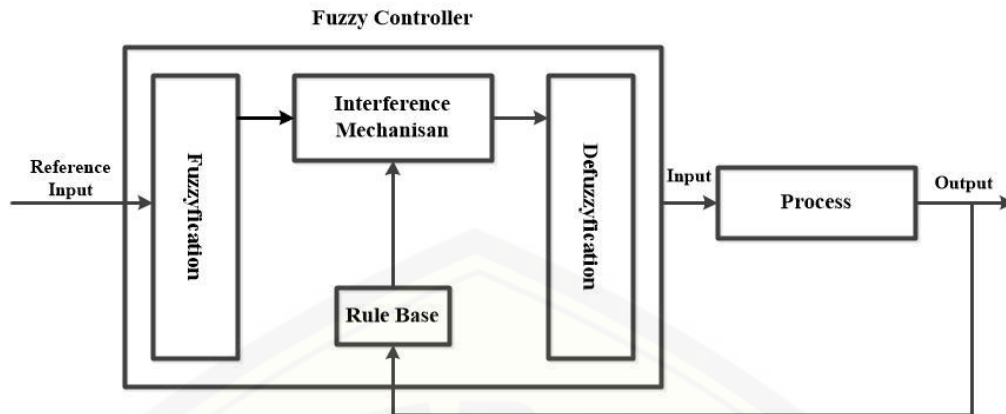
Dengan \cdot adalah operator (misal: OR, AND atau NOT). MIN (*minimum*). Fungsi ini akan memotong *output* himpunan fuzzy.

2.7.5 Tahap Pemodelan dalam Fuzzy Logic



Gambar 2.17 Blok Diagram Sistem Fuzzy 1

Dari blok diagram diatas, bila diterapkan dalam pengendalian suatu proses dapat kita gambarkan seperti blok diagram dibawah ini :



Gambar 2.18 Blok Diagram Sistem Fuzzy 2

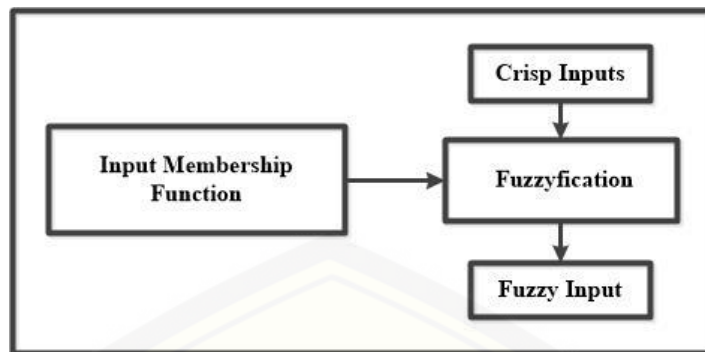
(Harittsa, 2019).

2.7.6 Fuzzyfikasi

Fuzzyfikasi adalah proses memetakan nilai crisp (numerik) ke dalam himpunan fuzzy dan menentukan derajat keanggotaannya di dalam himpunan fuzzy. Hal ini dilakukan karena data diproses berdasarkan teori himpunan fuzzy sehingga data yang bukan dalam bentuk fuzzy harus diubah ke dalam bentuk fuzzy (derajat keanggotaan) (Munir, 2012). Terdapat beberapa jenis fungsi keanggotaan, yaitu :

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

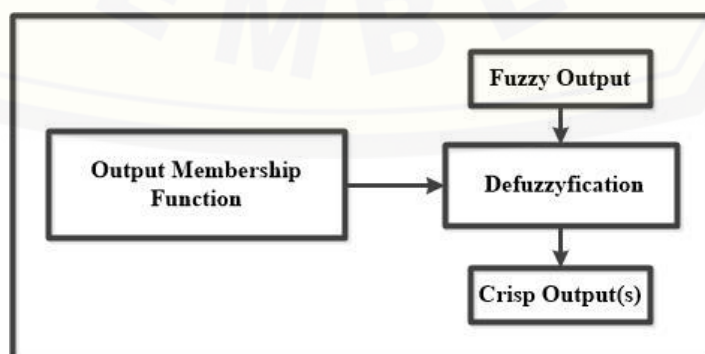
Terdapat komponen lain yang memiliki peranan penting yaitu label. Label didefinisikan dari fungsi keanggotaan, fungsi keanggotaan yang dikumpulkan akan menghasilkan *fuzzy set*. Dalam logika fuzzy terdapat basis pengetahuan yang terdiri dari fakta (*data base*) dan kaidah atur (*rule base*).



Gambar 2.19 Fuzzyfication

2.7.7 Defuzzyfikasi

Defuzzyfikasi 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 nilai *fuzzy output* yang berasal dari *rule evaluation* diambil kemudian dimasukkan ke dalam suatu *membership function output*. Bentuk bangun yang digunakan dalam *membership function* adalah bentuk *singleton* yaitu garis lurus vertikal keatas, 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. (Harittsa, 2019).



Gambar 2.20 Defuzzyfication

BAB 3. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Pada metode pelaksanaan data ini akan menjelaskan tempat dan waktu, ruang lingkup, jenis dan sumber data, serta metode pengumpulan data yang dilakukan dalam proses penelitian :

3.1.1 Tempat Penelitian

Penelitian ini dilakukan Laboratorium Sistem Tenaga, Fakultas Teknik, Universitas Jember yang beralamat di Jln. Slamet Riyadi No. 62 Patrang. Penelitian ini direncanakan akan dilakukan pada bulan Juli 2020 sampai dengan selesai.

3.1.2 Waktu Penelitian

Waktu penelitian ini dimulai pada bulan Juli 2020 sampai selesai, dengan rincian sebagai berikut:

Tabel 3.1 Rencana Kegiatan Penelitian

No	Kegiatan	Bulan Ke-/Minggu											
		Bulan ke-1		Bulan ke- 2-5				Bulan ke- 6					
		3	4	1	2	3	4	1	2	3	4		
1.	Studi Literatur	■	■										
2.	Penelitian		■	■	■	■	■						
3.	Analisis Data					■	■	■	■				
4.	Pembahasan dan Kesimpulan									■	■	■	■

Keterangan :

■ : Kegiatan dilaksanakan

3.2 Metode Penelitian

Penelitian ini menggunakan metode eksperimental menurut Prof. Dr. Sugiyono dalam bukunya Metode Penelitian Pendidikan (2010) dengan *pre-experimental design* dimana pada desain penelitian ini terdapat suatu kelompok diberi *treatment* (perlakuan) dan selanjutnya diobservasi hasilnya (*treatment* adalah sebagai variabel independen dan hasil adalah sebagai variabel dependen) serta menggunakan variabel kontrol. Dalam eksperimen ini subjek disajikan dengan beberapa jenis perlakuan lalu diukur hasilnya. (Perpusku, 2020).

3.3 Definisi Variabel Penelitian

1. Variabel Independent (Variabel Bebas)

Variabel Independent atau disebut juga variable bebas ialah variabel yang mempengaruhi atau yang menjadi penyebab terjadinya perubahan atau timbulnya variabel terikat. Pada penelitian ini yang menjadi variable bebas adalah kontrol *driver*, tegangan *input*, *dutycycle*, dan beban yang digunakan.

2. Variabel Dependent (Variabel Terikat)

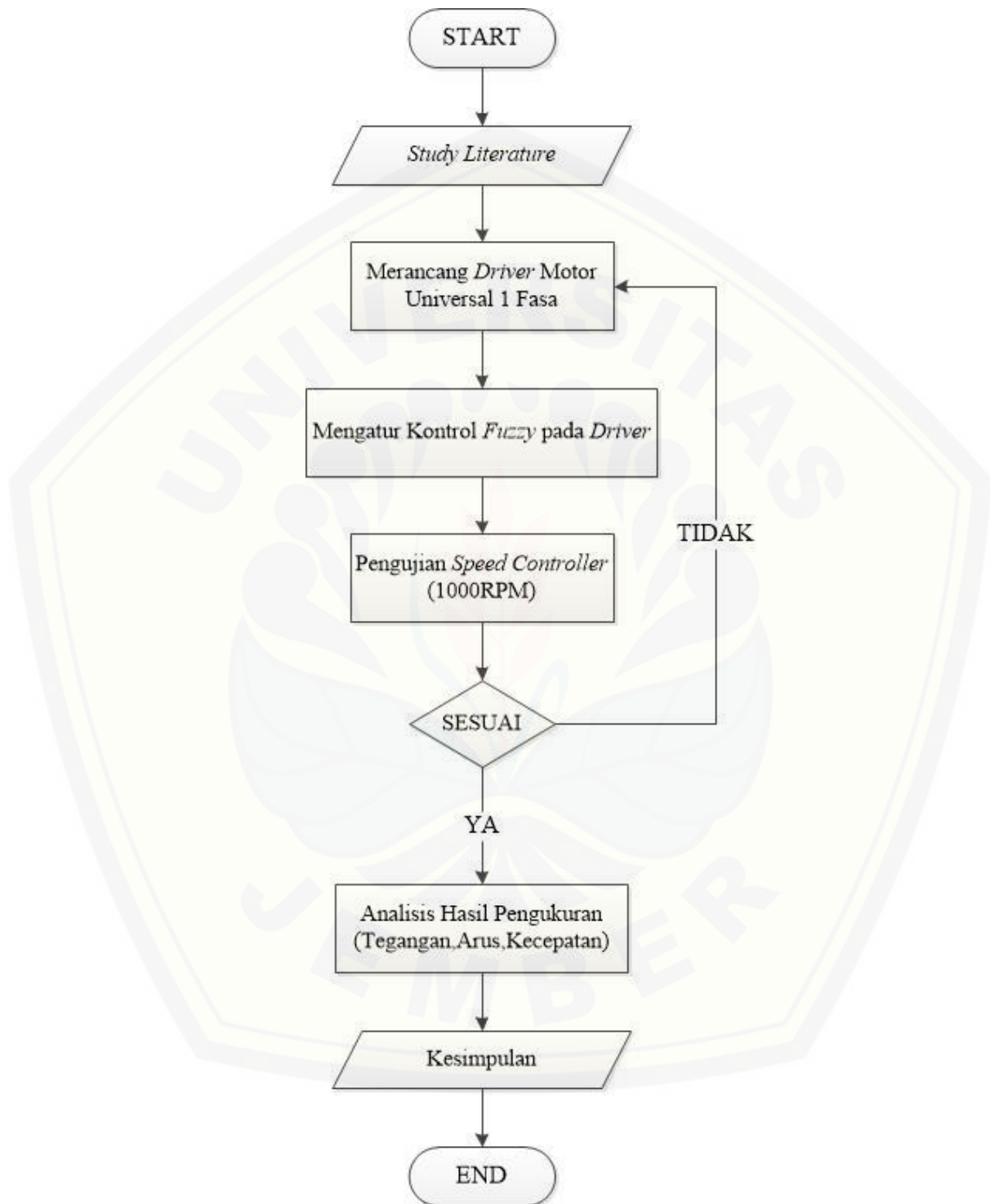
Variabel Dependent atau disebut juga variable terikat ialah variabel yang berubah dipengaruhi variabel bebas. Pada penelitian ini yang menjadi variable terikat adalah tegangan *output driver*, tegangan *output* generator, dan kecepatan aktual motor universal.

3. Variabel Kontrol

Variabel kontrol merupakan variabel yang ditetapkan sebagai kontrol dan pembanding terhadap variabel bebas dan terikat. Pada penelitian ini yang menjadi variabel kontrol adalah kecepatan motor pada 1000 RPM.

(Sugiyono, 2010).

3.4 Jenis dan Pengumpulan Data



Gambar 3.1 Flowchart Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian ini sebagai berikut:

4. Penelusuran Pustaka (*Literature Review*)

Penelusuran pustaka atau dapat disebut juga *literature review* berisi uraian tentang teori, temuan, dan bahan penelitian lain yang diperoleh dari bahan acuan untuk dijadikan landasan kegiatan penelitian. Uraian dalam *literature review* diarahkan untuk menyusun kerangka pemikiran yang jelas tentang pemecahan masalah yang sudah diuraikan pada perumusan masalah. Hal ini merupakan langkah pertama untuk mengumpulkan informasi yang relevan bagi penelitian dimana berguna untuk mendapatkan landasan teori mengenai objek yang akan diteliti, mendapatkan gambaran yang berkenaan dengan apa yang sudah pernah dikerjakan orang lain sebelumnya, dan merupakan langkah awal agar peneliti dapat lebih memahami permasalahan yang sedang diteliti dengan benar sesuai dengan kerangka berpikir ilmiah. Sumber-sumber yang digunakan untuk *literature review* berasal dari buku-buku, internet, Jurnal Internasional, Jurnal nasional dan juga *e-book*.

5. Penelitian Observasi Langsung (*Direct Observation*)

Penelitian ini dilakukan dengan cara pengambilan data dari parameter dan variabel yang ada secara langsung. Penelitian bertempat di Laboratorium Sistem Tenaga yang beralamat di Jln Slamet Riyadi No 62, Patrang, Jember, Jawa Timur. Data yang akan diambil ialah kecepatan motor universal r, tegangan luaran dan arus luaran dari *driver* motor universal satu fasa. Jenis data ini merupakan data primer karena dilakukannya observasi secara langsung dengan terjun ke lapangan.

3.5 Kerangka Pemecahan Masalah

Adapun tahap-tahap dalam penelitian dan penyusunan penelitian ini secara keseluruhan yang digunakan sebagai kerangka dalam pemecahan masalah pada penelitian kali ini dapat dinyatakan dalam diagram alir sebagaimana terlihat dalam Gambar 3.1.

Tahapan-tahapan pada penelitian yang dilakukan pada perancangan dan pengujian *driver* motor universal adalah:

1. Menentukan alat dan bahan yang akan digunakan.
2. Menentukan spesifikasi motor universal yang menggunakan tegangan 220V.
3. Merancang *driver* motor universal.
4. Merancang program *fuzzy logic controller*.
5. Menyematkan variabel bebas, terikat dan kontrol saat pengujian motor.
6. Hasil data yang diperoleh saat pengujian dapat dibahas dalam pembahasan sekaligus dapat menganalisa untuk membuat kesimpulan dari motor tersebut.

3.6 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Alat :

1. Obeng
2. Solder
3. Krisbow KW0600832 *digital multimeter bench type*
4. Extech *Photo Tachometer*
5. *Power Supply*
6. Arduino Uno R3
7. *Permanent Magnet Synchron Generator*
8. TDGC2 *Voltage Regulator of Contact*
9. Motor Universal
10. *Boost Converter DC – DC XL6009 5V – 38V DC*
11. *Fan PC 8cm 12V*
12. Hantek 5000 *series digital oscilloscopescilloscope*

Bahan :

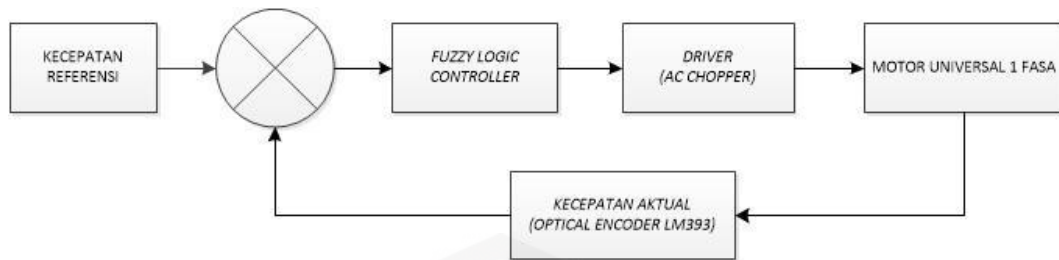
1. Sensor IR *Speed* LM393 *Optical Encoder*
2. Timah
3. Kabel Jumper
4. Akrilik 2 mm dan 2,5 mm
5. Baut M4 2cm
6. Terminal Blok 2 pin dan 3 pin
7. Transistor BC547
8. *Diode* HER307 *Ultra Fast Recovery*
9. MOSFET IRF740 N *Channel* 400V
10. *Spacer* 1 cm, 1,5 cm, dan 2 cm
11. PCB
12. Mur M4

3.7 Rancangan Penelitian

Jenis penelitian yang akan dilakukan yaitu dengan membuat rancang bangun lalu menguji kerja alat. Kemudian melakukan uji coba yang bertujuan mengambil data mengenai kecepatan, arus, dan tegangan pada *driver* motor. Proses yang terjadi yaitu memberikan nilai kecepatan aktual pada *driver* motor, setelah itu rotor akan berputar dan dengan itu dapat dilihat dari nilai yang terbaca oleh *optical encoder* untuk mengetahui kesesuaian kecepatan yang didapat pada motor universal dengan kontrol yang telah diberikan.

3.8 Blok Diagram dan Perancangan Sistem

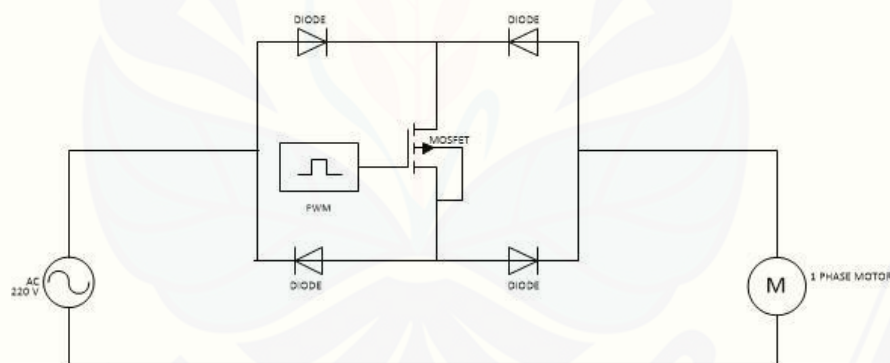
Pada sistem *driver* motor universal satu fasa, sumber tegangan yang digunakan adalah sumber tegangan AC (*Alternating Current*). Pada *driver* motor universal satu fasa menggunakan AC *Chopper* sebagai *driver* yang mengatur tegangan dan kontrol *Fuzzy Logic* sebagai kontroler. Pengujian pada *driver* motor universal menguji variabel berupa tegangan keluar, arus keluar, dan kecepatan.



Gambar 3.2 Blok Diagram Sistem

Pada blok diagram sistem dapat dimulai dari menentukan kecepatan referensi sebagai variabel kontrol yaitu 1000 RPM, *Fuzzy Logic Controller* berfungsi sebagai kontrol untuk mengatur nilai *dutycycle* yang dipengaruhi oleh selisih dari kecepatan referensi dan kecepatan aktual yang diperoleh dari pembacaan sensor *optical encoder LM393*. *Driver* yang telah di kontrol akan mengontrol kecepatan motor universal agar tetap stabil pada kecepatan 1000 RPM.

3.9 Desain Rangkaian AC Chopper

Gambar 3.3 Desain Rangkaian *Driver* Motor 1 Fasa

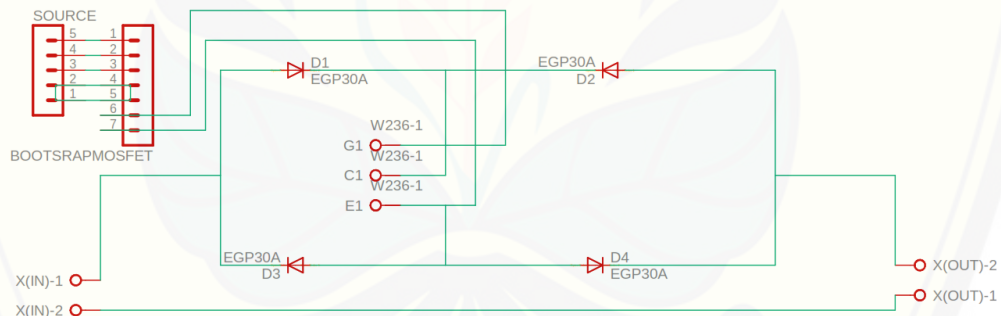
Rangkaian *AC Chopper* ini memiliki sumber tegangan 0V hingga 220VAC yang diteruskan ke rangkaian empat buah *rectifier diode* bertipe *ultra fast recovery* dengan kode HER307 dengan maksimal tegangan 600V dan maksimal arus 3 A serta menggunakan MOSFET dengan kode IRF740 dengan tegangan gate 15V. Tegangan yang dihasilkan dari rangkaian *AC Chopper* akan dipengaruhi oleh *switching* pada MOSFET berdasarkan nilai *dutycycle* yang diberikan. Tegangan *output* akan mempengaruhi kecepatan motor universal.

3.10 Alur Perancangan *Driver* Motor Universal

Penelitian ini mengacu pada sebuah rancang bangun, untuk merancang sebuah *driver* motor universal dengan *fuzzy logic controller* memiliki tahapan – tahapan yang harus dilakukan. Pada penelitian ini memiliki empat tahapan yang memiliki alur sebagai berikut.

3.10.1 Pembuatan Rangkaian *AC Chopper*

Rangkaian *AC Chopper* terdiri dari empat dioda rectifier dengan jenis *Ultra Fast Recovery Diodes* HER307 yang memiliki *maximum voltage* 700 V dan *maximum current* sebesar 3 A dan satu MOSFET IRF730 dengan *maximum VDS* sebesar 400 V dan *VGS* 15 V sebagai switching sinyal PWM yang akan mengatur tegangan *output*. Rangkaian tersebut nantinya akan terhubung langsung dengan sumber AC 220 V dan beban motor universal adalah *Permanent Magnet Synchron Generator*.

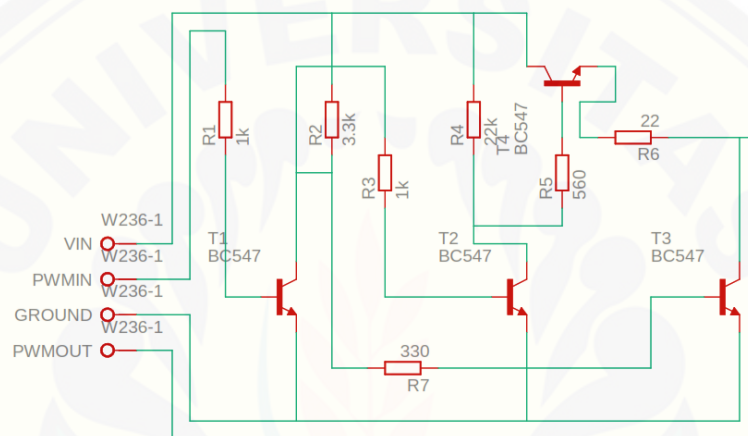


Gambar 3.4 *Schematic* Rangkaian *driver* Motor Universal

Rangkaian *AC Chopper* pada gambar 3.4 yang akan menjadi *driver* motor universal memiliki pengaturan kecepatan yang dipengaruhi oleh tegangan *output* dari *AC Chopper* dan menjadi tegangan *input* motor universal. Tegangan *output* tersebut akan dipengaruhi oleh PWM dengan MOSFET yang telah di tingkatkan tegangannya dengan *Bootstrap* MOSFET.

3.10.2 Pembuatan Rangkaian *Bootstrap driver* MOSFET

Rangkaian *bootstrap* ini tersusun atas tujuh buah resistor dan empat buah transistor tipe BC547, yang terhubung dengan sumber tegangan DC 18V. Rangkaian *bootstrap* berfungsi untuk meningkatkan tegangan sinyal PWM dari arduino yaitu sebesar 5V menjadi 18V dan diteruskan menuju MOSFET pada rangkaian AC *Chopper*. Sinyal PWM dari MOSFET kemudian akan mempengaruhi tegangan *output* dari *driver* motor.



Gambar 3.5 Schematic Rangkaian *Bootstrap* MOSFET

Bootstrap yang digunakan pada gambar 3.5 terdiri atas empat buah transistor bertipe BC547 dan 7 buah resistor dengan besaran R1 1k ohm, R2 3,3 ohm, R3 1k ohm, R4 22k ohm, R5 560 ohm, R6 22 ohm, dan R7 330 ohm. Yang terhubung dengan 4 pinheader dengan 3 pin menjadi *input* (Vin, PWMIn, Ground), dan satu output menuju *gate* pada MOSFET yaitu PWMout yang telah ditingkatkan tegangannya menjadi 18V.

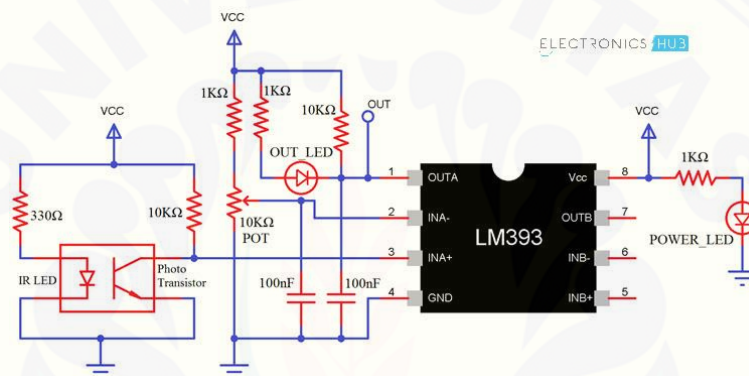
3.10.3 Pemasangan Sensor *Optical Encoder* LM393

Sensor yang digunakan pada penelitian ini berbentuk modul yaitu Sensor *Optical Encoder* LM393. Sensor ini akan menampilkan kecepatan aktual (RPM) dari motor universal pada serial monitor dengan program yang akan dibuat dan disematkan pada arduino. Hasil dari pembacaan sensor ini berfungsi sebagai umpan balik ke *driver* motor dan dikirim kembali menuju kontrol *Fuzzy* pada *driver* sesuai nilai yang telah diatur sebelumnya untuk menghasilkan kecepatan yang diinginkan.

Tabel 3.2 Spesifikasi Sensor *Optical Encoder LM393*

Working Voltage	3.3V – 5V
Dimensions	32 x 14 x 7 mm
I/O port	3(Vcc,Ground,D0)
Censor reading slot	5mm

Dengan *working voltage* 3-5V maka tegangan sensor bersumber dari Arduino Uno, serta ground dan D0 juga terhubung dengan Arduino. Adapun *schematic* dari modul sensor *optical encoder lm393* sebagai berikut :

Gambar 3.6 *Schematic* Rangkaian Sensor *Optical Encoder LM393*

Pada gambar 3.6 rangkaian *schematic* dari modul sensor terdapat sebuah IC LM393 serta sebuah modul yang berisi IR LED dan *Photo Transistor* yang berfungsi untuk membaca kecepatan *encoder* yang berputar. Terdapat juga sebuah *out LED* sebagai akan menyala apabila pancaran IR LED terbaca oleh *Photo Transistor* yang melalui celah *rotary encoder*.

3.10.4 Pemodelan Kontrol Fuzzy

Kontroler yang digunakan adalah *Fuzzy Logic Controller (FLC)*. Kontroler berfungsi untuk menerima masukan dari sinyal kesalahan yang berasal dari selisih antara kecepatan referensi (RPM_{in}) dan hasil kecepatan aktual (RPM) motor saat ini. FLC akan disematkan pada mikrokontroler Arduino UNO dengan tahap fuzzyfikasi, tahap *inferent* dan tahap defuzzyfikasi. Perancangan *Fuzzy Logic Controller* akan menggunakan dua *input* (*error* dan *Derror*) dan satu *output* (*Dutycyle*) sebagai *membership function*.

3.8.4.1 Fuzzifikasi

Tahap Fuzzifikasi merupakan langkah awal yang harus dilakukan untuk membuat *Fuzzy Logic Controller*. Pada tahap ini menentukan keanggotaan *membership function* yang akan disematkan pada dua *input* (*error* dan *Derror*) dan satu *output* (*Dutycyle*). Keanggotaan pada *membership function* dibagi menjadi tiga yaitu, N (*Negative*), Z (*Zero*), dan P (*Positive*).

Keanggotaan pada *membership function* ini memiliki tujuan untuk memproses sinyal PWM yang akan mengatur *ouput* (*Dutycycle*), kendali tersebut dilakukan berdasarkan nilai *error* dan nilai *Derror* sebagai *input*. *Fuzzy Logic Controller* akan membaca sinyal dari sensor yang menerima *input* dari pembacaan sensor yaitu kecepatan aktual motor (RPM).

Dengan algoritma :

$$Error = SP - PV$$

$$Derror = Error(n) - Error(n - 1)$$

SP = Nilai RPM aktual (RPMout)

PV = Nilai RPM yang diinginkan (RPM)

$Error(n)$ = Nilai *error* sekarang

$Error(n-1)$ = Nilai *error* sebelumnya

(Universitas Kristen Petra, 2001).

Fuzzy Logic Controller akan memngolah nilai dari kecepatan aktual (RPMout) dikurangi kecepatan referensi (RPM) dan hasilnya menjadi *error*. Setelah itu *Fuzzy Logic Controller* akan memproses nilai perubahan *error* dan menjadikannya *Derror*.

3.8.4.2 Inference

Tahap *Rule Evaluation* atau tahap *inference* adalah untuk memetakan nilai derajat keanggotaan dari *error* ataupun *Derror* ke dalam tabel yang berisi aturan (*rule base*) yang dipilih dan menggunakan sebuah metode untuk pemilihan nilai dari derajat keanggotaan. Tabel *Rule Base* dapat dilihat pada tabel 3.7.

Tabel 3.7 Rule Base

error/Derror	N	Z	P
N	P	P	N
Z	P	Z	N
P	P	N	N

Keterangan :

N = *Negative*Z = *Zero*P = *Positive*

3.8.4.3 Defuzzifikasi

Tahap defuzzifikasi merupakan langkah terakhir yang dilakukan dalam membuat *Fuzzy Logic Controller*. Tahap defuzzifikasi menghasilkan bilangan pada domain himpunan fuzzy. Defuzzifikasi dapat dilakukan secara normal dengan dua algoritma, yaitu Center of Area (COA) dan Metode Kriteria Maksimum (MCM). Metode defuzzifikasi yang digunakan adalah metode Center of Area.

$$z^* = \frac{\text{momen}}{\text{luas daerah}} = \frac{\int \mu_x(z) \cdot z dz}{\int \mu_x(z) dz}$$

Dengan derajat keanggotaan dari ketiga anggota himpunan bernilai 1 dan menggunakan tipe segitiga, maka nilai z^* dapat diketahui dengan menghitung luas daerah dengan rumus segitiga.

BAB 5. PENUTUP

5.1 Kesimpulan

Dari analisis dan hasil pengujian dari penelitian yang berjudul “ Rancang Bangun *AC-AC Chopper* Sebagai *Driver* Motor Universal Berbasis *Fuzzy Logic Controller* ” diperoleh kesimpulan sebagai berikut :

1. *Driver* motor universal yang dirancang menggunakan *AC Chopper* memiliki pengaturan kecepatan melalui nilai *dutycycle* dari PWM yang memiliki rata-rata pemotongan tegangan 17,74 V, serta memiliki desain fuzzy yang berfungsi sebagai umpan balik yang akan menghasilkan penambahan atau pengurangan *dutycycle* dengan *range* 7% - 94%.
2. *AC Chopper basic* yang dirancang sebagai *driver* motor universal dengan kontrol umpan balik *fuzzy logic controller* memiliki nilai rata – rata efisiensi adalah sebesar 86,96%.
3. Hasil pengendalian kecepatan motor universal saat kondisi berbeban memiliki rentang waktu 3 – 5 detik untuk mencapai 1000 RPM, sedangkan saat kondisi pelepasan beban memiliki nilai *overshoot* hingga 1193,33 RPM namun dapat mencapai 1000 RPM dalam rentang 2 - 3 detik.

5.2 Saran

Diperlukan penelitian lebih lanjut mengenai *AC Chopper* yang difungsikan sebagai motor dalam hal penggunaan jenis MOSFET yang berbeda, algoritma yang digunakan, hingga jenis *AC Chopper* yang lain untuk meningkatkan nilai dari tegangan yang bisa di *chopping*.

DAFTAR PUSTAKA

- Yibo Wang, G. C. (2020). An Improved Bipolar-Type AC-AC Converter Topology Based on Non-Differential Dual-Buck PWM AC Choppers . *IEEE*.
- Said A. Deraz, H. Z. (2017). Current limiting soft starter for three phase induction motor drive system using PWM AC Choppers. *IET Power Electronics*.
- Elsevier. (2007). Single Phase AC-AC Voltage Controller with On/Off Control. In M. H. Rashid, *Power Electronics Handbook: Devices, Circuits and Applications* (pp. 487-488). Elsevier.
- Hafiz Furqan Ahmed, M. S. (2019). A Reliable Single-Phase Bipolar Buck-Boost Direct PWM AC-AC Converter with Continuous Input/ Output Currents. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* .
- Helmy Mukti Himawan, O. S. (2016). Pemodelan Fuzzy Logic Control untuk Pengendali PWM. *JNTETI, Vol. 5, No. 1, Februari*, 1-8.
- Kumar, P. S. (2014). Design and implementation of AC chopper. *Int. J. Emerg. Eng. Res. Technol*, 2(1), 36-41.
- Nurhayata, I. G. (2015). SISTEM PENGATURAN KECEPATAN MOTOR UNIVERSAL SATU FASA DENGAN METODE KONTROL SUDUT FASA BERBASIS MIKROKONTROLER AT89S52. *JPTK, UNDIKSHA, Vol. 12*.
- Sri Kusumadewi, Hari Purnomo. (2010). Aplikasi logika fuzzy untuk pendukung keputusan. In S. Kusumadewi, *Koleksi Buku 2010*. Yogyakarta: Graha Ilmu.
- Adam, A. A. (2011). PERBANDINGAN KONSUMSI DAYA OLEH ALAT PENGATUR KECEPATAN MOTOR UNIVERSAL YANG MENGGUNAKAN METODE MODULASI LEBAR PULSA (PWM) DAN METODE TAHANAN DEPAN . *Jurnal SMARTek, Vol. 9*, 231.
- Bahariawan, A. S. (2018). *Buku Ajar Energi dan Elektrifikasi Pertanian*. DEEPUBLISH.
- Harittsa, A. R. (2019). *Rancang Bangun Konverter SEPIC untuk pengisian baterai 48V berbasis Fuzzy Logic*. Jember: Universitas Jember.
- Institut Teknonolgi Bandung. (2009). LOGIKA FUZZY. *MAKALAH IF2091* , 1-5.

- Munir, R. (2012). *Sistem Inferensi Fuzzy*. Bandung: STEI.
- Nazaruddin, R. (2015). RANCANG BANGUN SISTEM TELEMETRI PENGATUR KECEPATAN PUTAR MOTOR DC DENGAN METODE DC CHOPPER MENGGUNAKAN GUI (GRAPHICAL USER INTERFACE). *repository.upi.edu* .
- Pramono, W. B. (April, 2018). RANCANG BANGUN ELECTROMAGNETIC BOOTSTRAP SEBAGAI GATE DRIVER PADA MOSFET. *PROSIDING SEMINAR NASIONAL ENERGI & TEKNOLOGI (SINERGI)*, 50-59.
- Prihanto, T. A. (2019, February). Retrieved from Academia: https://www.academia.edu/15314402/Motor_universal?auto=download
- Sri Kusumadewi, P. (2004). Aplikasi Logika Fuzzy untuk Pendukung Keputusan.
- Sudibyoo., S. (Teknosia Vol. II). ANALISIS EFISIENSI MOTOR INDUKSI PADA KONDISI TEGANGAN NON RATING DENGAN METODE SEGREGATED LOSS.
- Sudradjat. (2008). DASAR - DASAR FUZZY LOGIC. In U. Padjadjaran, *Modul Kuliah* (pp. 1-63). Bandung.
- Sutikno, I. W. (2018). *Media Neliti*. Retrieved 12 12, 2020, from Media Neliti: <https://media.neliti.com/media/publications/112322-ID-perbandingan-metode-defuzzifikasi-sistem.pdf>
- Tomy, R. O. (2019, September 13). *Motor Listrik*. Retrieved from General TOMY: <http://rizkitomy10.blogs.uny.ac.id/tag/motor-universal/>
- Universitas Kristen Petra. (2001). Aplikasi Kendali Fuzzy Logic untuk Pengaturan Kecepatan Motor Universal. *Jurnal Teknik Elektro Vol.1, No.1*, 33-42.
- UNY. (2011). Logika Fuzzy.
- Prof. Dr. Sugiyono (2010). Metode Penelitian Pendidikan(pendekatan kuantitatif, kualitatif, dan R & D). Bandung: Alfabeta.
- Wulandari, F. (2005). Pembuatan sistem pendukung keputusan berbasis teori fuzzy untuk mengembangkan suatu produk baru. *Jurnal Sain, Teknologi & Industri*, 62-66

LAMPIRAN

1. Program *Software* Arduino IDE

```
#include "Wire.h"
#include "fis_header.h"
#include <LiquidCrystal.h>
#define enA 2 // hubungkan encoderA ke pin 2
#define enB 3 // hubungkan encoderB ke pin 2
//LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
//Fuzzy initialize
// 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 = 9;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
float RPMIn=1000,errorn=0,error,Derror;
//=====//
const int button = 9;
int PWMPin = 3; //PWM Connect to Pin 3
float PWM = 128; //start pwm with 50% duty cycle
int Duty;
long prevMillis_control = 0,
prevMillis_ADC = 0,
prevMillis_display = 0,
interval_control = 100*64, //100 milisecon
interval_ADC = 50*64, //50 milisecon
interval_display = 200*64; //500 milisecon
```

```
uint8_t interval_serial = 0;
// sensorrpm
int pulse = 0;
float rps;
float rpm;
int jumlahPulsa,lastjumlahPulsa;
int interval=500;
unsigned int waktulama, waktusekarang;
void setup()
{
  TCCR2B = TCCR2B & B11111000 | B00000101;
  pinMode(8, OUTPUT);
  digitalWrite(8, HIGH);
  pinMode(PWMPin, OUTPUT);
  Serial.begin(9600);
  //TCCR2B = TCCR2B & 0b11111000 | B00000001; //change the pwm Pin
  //frequency to 244.05 Hz
  Serial.begin(9600);
  pinMode(enA, INPUT_PULLUP);
  pinMode(enB, INPUT_PULLUP);
  attachInterrupt(0, encoder, CHANGE);
}
//pembacaansensorrpm
void RPM(){
  waktusekarang=millis();
  if (waktusekarang - waktulama >= interval){
    rps = (jumlahPulsa/360.00)*20;
    // rpm = (jumlahPulsa/360.00)/0.008333333;
    rpm = (abs(rps*60));
    pulse=0;
    jumlahPulsa=0;
```

```
waktulama=waktusekarang;
}
}
void encoder()
{
if (digitalRead(enA) == digitalRead(enB))
{
pulse++;
}
else
{
pulse--;
}
jumlahPulsa = pulse*2;
}
//initialfuzzy//
void initialfuzzy(){
error=rpm-RPMin;
Derror=error-errorn;
g_fisInput[0] = error;
g_fisInput[1] = Derror;
g_fisOutput[0] = 0;
errorn=error;
}

//*****
// 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[] = { 3, 3 };
// Count of member function for each Output
int fis_gOMFCount[] = { 3 };
// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -1100, -1100, -800, -89 };
FIS_TYPE fis_gMFI0Coeff2[] = { -90, 0, 90 };
FIS_TYPE fis_gMFI0Coeff3[] = { 89, 800, 2000, 2000 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
    fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { -40, -40, -28, -3.9 };
FIS_TYPE fis_gMFI1Coeff2[] = { -4, 0, 4 };
FIS_TYPE fis_gMFI1Coeff3[] = { 3.9, 28, 40, 40 };

```

```
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2,
fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };
// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -0.03, -0.025, -0.01 };
FIS_TYPE fis_gMFO0Coeff2[] = { 0, 0, 0 };
FIS_TYPE fis_gMFO0Coeff3[] = { 0.01, 0.025, 0.03 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };
// Input membership function set
int fis_gMFI0[] = { 0, 1, 0 };
int fis_gMFI1[] = { 0, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };
// Output membership function set
int fis_gMFO0[] = { 1, 1, 1 };
int* fis_gMFO[] = { fis_gMFO0 };
// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };
// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };
// Rule Inputs
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 3, 3 };
int fis_gRI2[] = { 1, 2 };
int fis_gRI3[] = { 3, 2 };
int fis_gRI4[] = { 1, 3 };
int fis_gRI5[] = { 3, 1 };
int fis_gRI6[] = { 2, 1 };
int fis_gRI7[] = { 2, 3 };
int fis_gRI8[] = { 2, 2 };
```



```

int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5,
fis_gRI6, fis_gRI7, fis_gRI8 };
// Rule Outputs
int fis_gRO0[] = { 3 };
int fis_gRO1[] = { 1 };
int fis_gRO2[] = { 3 };
int fis_gRO3[] = { 1 };
int fis_gRO4[] = { 3 };
int fis_gRO5[] = { 1 };
int fis_gRO6[] = { 3 };
int fis_gRO7[] = { 1 };
int fis_gRO8[] = { 2 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4,
fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8 };
// Input range Min
FIS_TYPE fis_gIMin[] = { -1100, -40 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 2000, 40 };
// Output range Min
FIS_TYPE fis_gOMin[] = { -0.03 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 0.03 };
//*****
// 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));
}
//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 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_gcO; ++o)
  {
    g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
  }
}
else
{
  for (o = 0; o < fis_gcO; ++o)
  {
    g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
  }
}
}

void PWMoutput(){
  PWM += (g_fisOutput[0] * 255);
  PWM= constrain (PWM, 20, 240);
}

void disp() {
//=====Serial Monitor=====//
  Serial.print(rpm) ;
  Serial.print(" | ");
  Serial.print(error);
  Serial.print(" | ");
```

```
Serial.print(Derror);
Serial.print(" | ");
Serial.println(Duty);
delay (500);
}
void loop (){
//programfuzzy
RPM();
initialfuzzy();
fis_evaluate();
PWMoutput();
analogWrite(PWMPin, PWM);
Duty = (PWM/255)*100;
if(millis()-prevMillis_display > interval_display){
prevMillis_display = millis();
interval_serial += 1;
if(interval_serial == 10);}
disp();
// delay (500);
}
```