

**IMPLEMENTASI KONTROL LOGIKA FUZZY PADA  
SISTEM KESETIMBANGAN ROBOT BERODA DUA**

**SKRIPSI**

**KONSENTRASI TEKNIK KONTROL**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh:

**SHANTY PUSPITASARI**

**NIM. 105060307111023-63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK ELEKTRO**

**MALANG**

**2015**

## LEMBAR PERSETUJUAN

### IMPLEMENTASI KONTROL LOGIKA FUZZY PADA SISTEM KESETIMBANGAN ROBOT BERODA DUA

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan  
memperoleh Gelar Sarjana Teknik



DISUSUN OLEH :  
**SHANTY PUSPITASARI**  
NIM. 105060307111023 - 63

Telah diperiksa dan disetujui oleh :

Pembimbing 1

Pembimbing 2

Goegoes Dwi Nusantoro, S.T., M.T.  
NIP. 197110132 00604 1 001

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

## LEMBAR PENGESAHAN

### IMPLEMENTASI KONTROL LOGIKA FUZZY PADA SISTEM KESETIMBANGAN ROBOT BERODA DUA

#### SKRIPSI JURUSAN TEKNIK ELEKTRO

Disusun oleh:

**SHANTY PUSPITASARI**  
NIM. 105060307111023 – 63

Skrripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 13 Januari 2015

#### MAJELIS PENGUJI

Dr. Ir. Erni Yudaningsyach, M.T.NIP.  
19650913 199002 2 001

Eka Maulana, S.T., M.T., M.Eng  
NIP. 841130 06 1 1 0280

Ir. Purwanto, M.T  
NIP. 19540424 198601 1 001

Mengetahui  
Ketua Jurusan Teknik Elektro

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

## Kata Pengantar

Puji syukur kepada Tuhan YME yang telah memberikan segala nikmat dan rahmat-Nya sehingga penyusunan skripsi ini dapat diselesaikan. Karena hanya dengan pertolongan-Nya semata penulis mampu melewati segala kendala yang ada selama penyusunan skripsi ini.

Skripsi berjudul “Implementasi Kontrol Logika Fuzzy Pada Sistem Keseimbangan Robot Beroda Dua” ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Terelesaiannya skripsi ini tentunya tidak lepas juga dari bantuanberbagaipihak.Oleh sebab itu, dengan segala kerendahan hati penulis menyampaikan terima kasih kepada:

1. Ayah, Ibu dan Kakak selaku keluarga yang senantiasa memberikan atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini.
2. Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Bapak Hadi Suyono, ST., M.T., Ph.D selaku Sekertaris Jurusan Teknik Elektro Universitas Brawijaya
3. Bapak Ir. Purwanto, MT. selaku Ketua Kelompok Dosen Keahlian Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya atas segala bimbingan, nasehat, pengarahan, motivasi, saran, dan masukan yang telah diberikan
4. Bapak Goegoes Dwi Nusantoro, ST., MT. selaku Dosen Pembimbing 1 atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi dan bantuan yang telah diberikan. dan Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Dosen Pembimbing 2 atas segala bimbingan, nasehat, pengarahan, motivasi, saran, dan masukan yang telah diberikan
5. Bapak dan Ibu dosen beserta staff dan karyawan Jurusan Teknik Elektro, baik secara langsung maupun tidak langsung telah membant umenyelesaikan skripsi ini.
6. Sahabat tercinta saya Intan Febriana, Ira Ayu, Reta, Aya, Safti, Nilfa, Nana, dan teman-teman Magnet 2010 lainnya yang senangtiasa saling mendukung dan memberi motivasi.
7. Seluruh teman-teman, senior dan junior serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu per satu, terima kasih banyak atas bantuan dan dukungannya.
8. Devisi Line Tracer Comunity (LTC) yang telah bersedia memberikan bantuannya

9. Semua pihak yang tidak dapat penulis sebutkan satu persatu

Penulis menyadari bahwa skripsi ini masih belum sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, Januari 2015

Penulis



## DAFTAR ISI

	halaman
<b>PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>ABSTRAK .....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan.....	2
1.5. Sistematika Penulisan .....	2
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1. Penelitian Terdahulu.....	4
2.2. Robot Beroda Dua .....	5
2.3. Robogames .....	7
2.4. Kontroler.....	7
2.4.1 Kontrol Logika Fuzzy .....	7
2.4.2 Fungsi Keanggotaan .....	8
2.4.3 Kendali Logika Fuzzy.....	9
2.4.4 Fuzzifikasi.....	9
2.4.5 Kaidah Atur Fuzzy.....	10
2.4.6 Defuzzifikasi.....	11
2.5. Motor DC.....	12
2.5.1 Prinsip Kerja .....	13
2.5.2 Pengaturan Motor DC .....	14
2.5.3 PWM ( <i>Pulse Width Modulation</i> ) .....	14
2.5.4 Driver H-bridge.....	15
2.6 Arduino Mega 2560.....	18
2.6.1. Daya.....	20
2.6.2. Memory.....	20

2.6.3. Input dan Output .....	19
2.6.4. Komunikasi .....	23
2.7 Pemrograman .....	22
2.8. Sensor Gyroscope .....	22

### **BAB III METODOLOGI PENELITIAN**

3.1. Studi Literatur .....	24
3.2. Penentuan spesifikasi Alat .....	24
3.3 Perancangan Dan Pembuatan Alat .....	25
3.3.1 Perancangan dan Pembuatan Perangkat Keras .....	25
3.3.2 Perancangan dan Penyusunan Perangkat Lunak .....	25
3.4 Pengujian dan Analisis .....	25

### **BAB IV PERANCANGAN DAN PEMBUATAN ALAT**

4.1. Perancangan Sistem .....	26
4.2. Perancangan Perangkat Keras .....	27
4.2.1 Perancangan Mekanik Robot .....	27
4.2.2 Desain Sistem Elektronik .....	28
4.3 Perancangan Arduino Mega 2560 .....	32
4.4 Pin Arduino .....	24
4.5 Perancangan Perangkat Lunak ( <i>Software</i> ) .....	34
4.5.1. Perancangan Kontrol Logika Fuzzy .....	35
4.5.1.1. Menentukan Variabel Masukan dan Variabel Keluaran .....	35
4.5.1.2. Fungsi Keanggotaan Masukan .....	36
4.5.1.3. Fungsi Keanggotaan Keluaran .....	38
4.4.1.4 Metode Inferensi .....	40
4.4.1.5 Defuzzifikasi .....	41
4.5.2. Perancangan Algoritma Perangkat Lunak .....	42

### **BAB V PENGUJIAN DAN ANALISIS**

5.1 Pengujian Aktuator .....	43
5.2 Pengujian <i>Sensor</i> .....	46
5.3 Pengujian Keseluruhan .....	48

### **BAB VI KESIMPULAN DAN SARAN**

6.1. Kesimpulan.....	55
6.2. Saran.....	55

**DAFTAR PUSTAKA**

**LAMPIRAN**



## DAFTAR GAMBAR

Gambar 2.1 Mekanik Robot Penelitian Dahulu .....	2
Gambar 2.2 Mekanik Robot Penelitian Sekarang.....	2
Gambar 2.3 Sketsa Robot Beroda Dua .....	6
Gambar 2.4 Pengendali Kontrol Logika Fuzzy .....	8
Gambar 2.5 Fungsi Kenggotaan Triangular.....	9
Gambar 2.6 Inferensi Metode MAX-MIN.....	11
Gambar 2.7 Metode MAX Membership Principle .....	12
Gambar 2.8 Pernacangan Motor DC.....	13
Gambar 2.9 Proses Kerja Motor DC.....	13
Gambar 2.10 Sinyal PWM Secara Umum .....	15
Gambar 2.11 Rangkaian H-Bridge Driver .....	16
Gambar 2.12 Rangkaian H-Bridge Driver saat $A=0$ $B=1$ .....	17
Gambar 2.13 Board Arduino.....	19
Gambar 2.14 Analogi Axis Gyroscope.....	22
Gambar 4.1 Diagram Blog Sistem .....	26
Gambar 4.2 Rancangan Desain Mekanik Robot.....	27
Gambar 4.3 Prespektif Desain Mekanik Robot .....	28
Gambar 4.4 Desain Sistem Robot.....	28
Gambar 4.5 Sensor IMU Analog Board .....	29
Gambar 4.6 Rangkaian H-Bridge .....	30
Gambar 4.7 Driver Motor .....	30
Gambar 4.8 Rangkain Elektrik Driver .....	31
Gambar 4.9 Arduino Mega 2560 .....	33
Gambar 4.10 Komunikasi Arduino.....	35
Gambar 4.11 Diagram Perancangan KLF.....	35
Gambar 4.12 Fungsi Kenggotaan Masukan Gyro.....	37
Gambar 4.13 Fungsi Keanggotan Masukan Derror .....	37
Gambar 4.14 Fungsi Keanggotaan Keluaran PWM .....	38
Gambar 4.15 Refrensi Rule.....	41
Gambar 5.1 Diagram Blok Pengujian Driver .....	44
Gambar 5.2 Grafik Duty Cycle dengan PWM.....	45
Gambar 5.3 Grafik Tegangan dengan Putaran Motor.....	46

Gambar 5.4 Diagram Blok Pengujian Sensor .....47

Gambar 5.5 Proses Pengambilan Data .....49

Gambar 5.6 Diagram Blok Secara Keseluruhan .....49

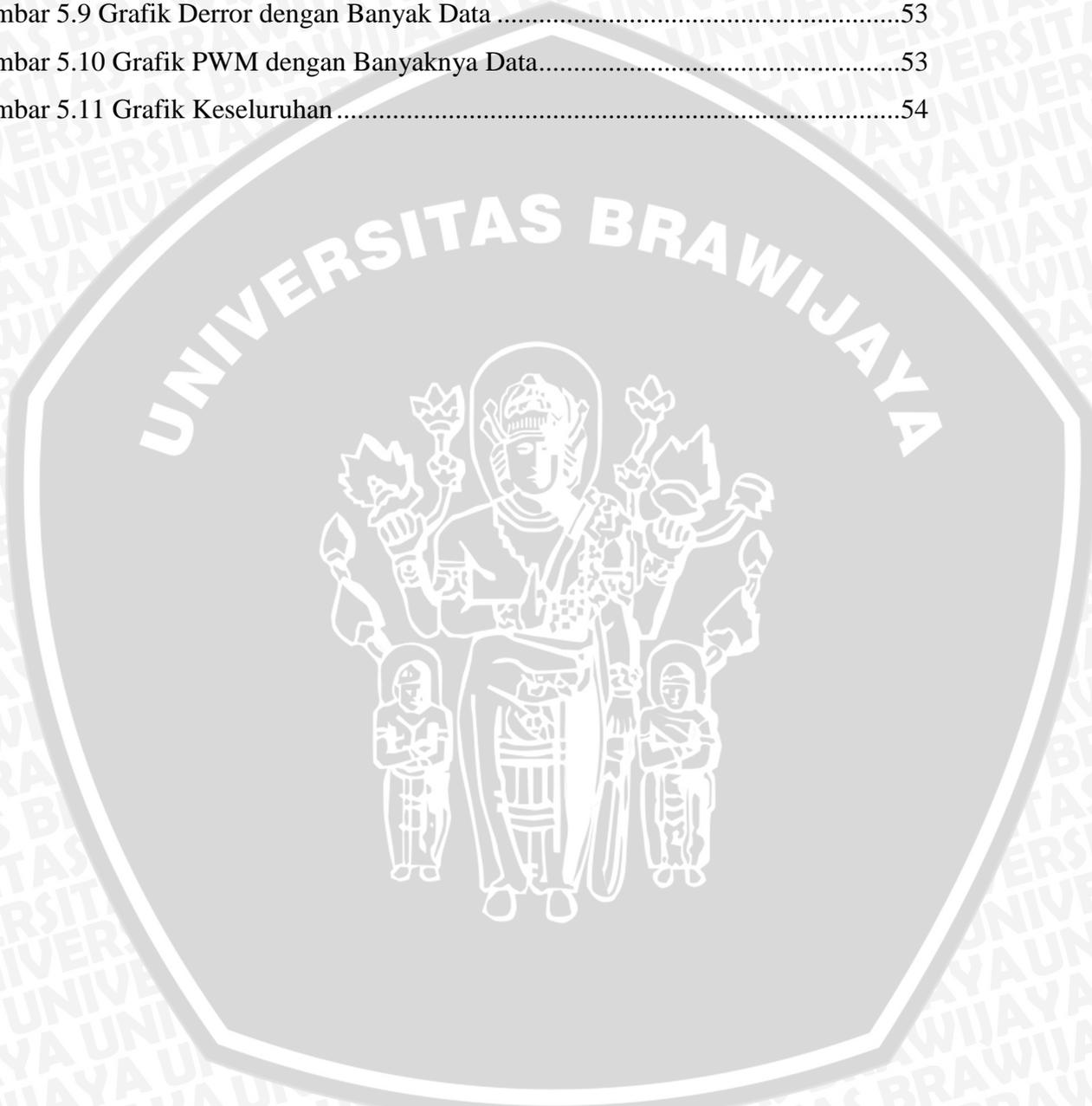
Gambar 5.7 Grafik Sensor dengan Banyaknya Data .....52

Gambar 5.8 Grafik Error dengan Banyaknya Data .....53

Gambar 5.9 Grafik Derror dengan Banyak Data .....53

Gambar 5.10 Grafik PWM dengan Banyaknya Data .....53

Gambar 5.11 Grafik Keseluruhan .....54



**DAFTAR TABEL**

Tabel 4.1 Kebenaran Driver.....32

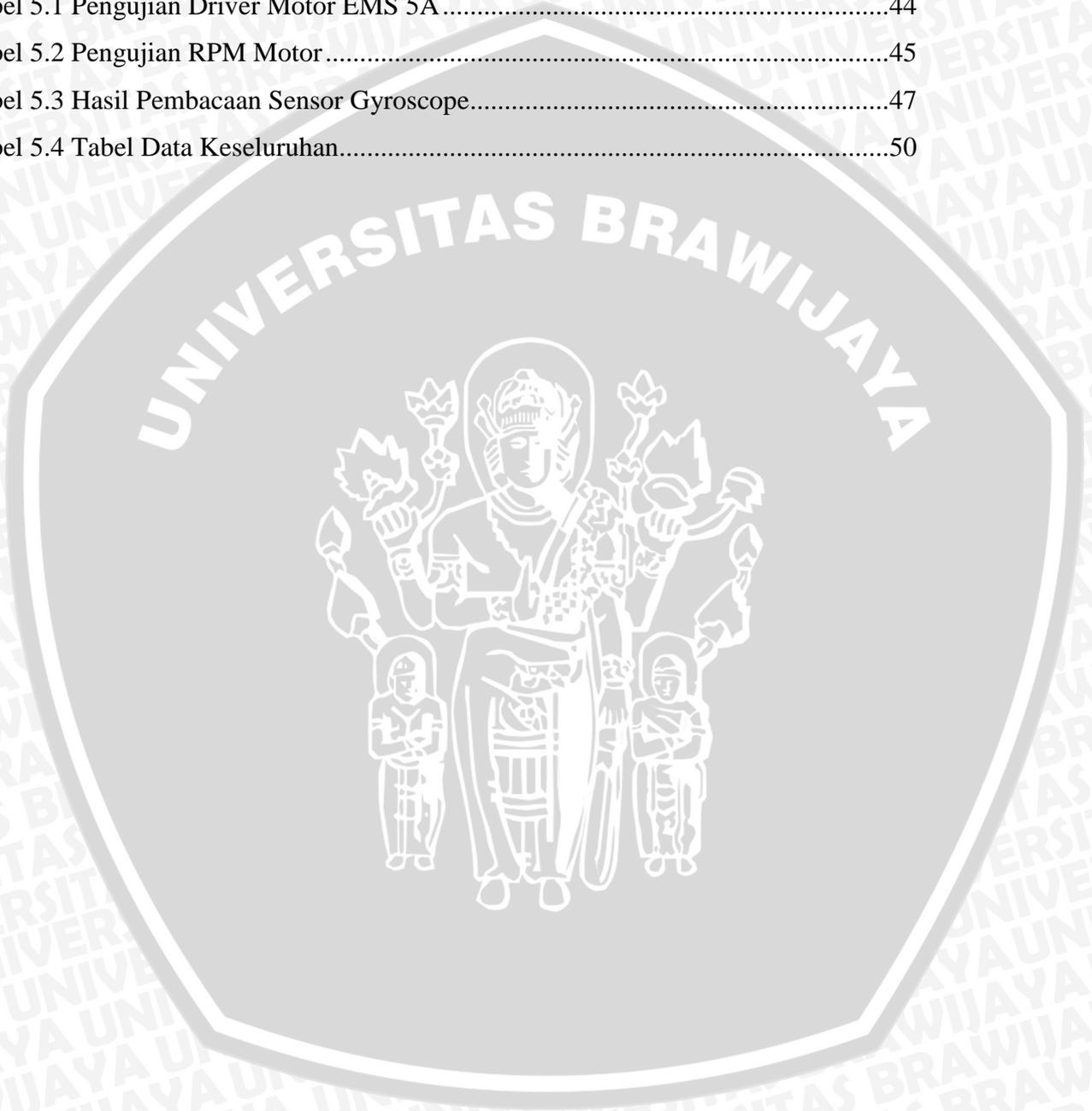
Tabel 4.2 Kaidah aturan Fuzzy .....39

Tabel 5.1 Pengujian Driver Motor EMS 5A.....44

Tabel 5.2 Pengujian RPM Motor .....45

Tabel 5.3 Hasil Pembacaan Sensor Gyroscope.....47

Tabel 5.4 Tabel Data Keseluruhan.....50



## ABSTRAK

**Shanty Puspitasari** , Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2015, Implementasi kontrol Logika Fuzzy Pada Sistem Keseimbangan Robot Beroda Dua, Dosen Pembimbing: Goegoes Dwi Nusantoro dan M. Aziz Muslim

Teknologi robotika telah mampu meningkatkan kualitas maupun kuantitas berbagai industri. Teknologi robotika juga telah menjangkau sisi hiburan dan pendidikan bagi manusia. Sekarang ini sudah ada alat transportasi yang menggunakan dua roda yang di kendalikan oleh satu orang (segway). Pada skripsi ini juga akan membuat alat yang sama dalam hal prinsip kerjanya, namun dengan ukuran yang berbeda. Alat ini juga nantinya dibuat agar dapat menyeimbangkan robot agar tidak terjatuh. Kedua roda robot di hubungkan dengan motor DC. Salah satu cara menambah tingkat kecerdasan sebuah robot adalah dengan menambah sensor dan metode kontrol. Robot beroda dua merupakan suatu robot *mobile* yang memiliki sebuah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Penelitian skripsi ini adalah untuk mendesain dan membangun robot beroda dua yang mampu menyeimbangkan dirinya yang tegak lurus terhadap permukaan bumi di daerah bidang datar. Pada penelitian skripsi ini digunakan Arduino Mega 2560, sensor *gyroscope*. Dengan mengatur keseimbangan Sensor yang digunakan untuk mendeteksi kemiringan robot terhadap bumi serta menggunakan metode kontrol logika *fuzzy* sebagai pengendali. Kontrol logika *fuzzy* digunakan untuk menentukan dan merancang kontroler yang diharapkan untuk membuat robot beroda dua menjadi seimbang atau berada dalam keadaan tegak.

Kata Kunci: robot roda dua, logika *fuzzy*, sensor *gyroscope*, arduino mega 2560

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi robotika telah membuat kualitas kehidupan manusia semakin tinggi. Saat ini perkembangan teknologi robotika telah mampu meningkatkan kualitas maupun kuantitas berbagai industri. Teknologi robotika juga telah menjangkau sisi hiburan dan pendidikan bagi manusia. Salah satu cara menambah tingkat kecerdasan sebuah robot adalah dengan menambah sensor, metode control bahkan memberikan kecerdasan buatan pada robot tersebut. Salah satunya adalah robot beroda dua. (R. Muhammad Syafruddi, 2013)

Robot beroda dua merupakan suatu robot *mobile* yang memiliki sebuah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan handal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar. Bahkan sekarang ini konsep robot beroda dua telah digunakan sebagai alat transportasi yang bernama *segway*. (Pitowarno, 2006)

Penelitian skripsi ini adalah untuk mendesain dan membangun robot beroda dua yang mampu menyeimbangkan dirinya yang tegak lurus terhadap permukaan bumi di daerah bidang datar. Pada penelitian skripsi ini digunakan Pengendali FLC merupakan penerapan konsep logika *Fuzzy* pada suatu sistem pengendalian yang bekerja secara *heuristic* dan termasuk salah satu bagian dari kecerdasan buatan. Metode ini terbukti efektif dalam menghadapi sistem nonlinier yang kompleks dan memiliki ketidak pastian yang sulit di modelkan (Rizan, 2008). Pengendali FLC merupakan pengendali yang mudah dan sederhana untuk dirancang, tetapi juga memiliki kekurangan atau kelemahan, diantaranya adalah keterbatasan analisis maupun sintesis

### 1.2 Rumusan Masalah

Dari masalah yang telah disebutkan di atas, maka dalam skripsi ini rumusan masalah ditekankan pada:

1. Bagaimana merancang kontroler *Fuzzy* untuk mempertahankan keseimbangan pada robot beroda dua?
2. Bagaimana mengaplikasikan kontroler *Fuzzy* pada robot beroda dua dengan menggunakan Arduino Mega dan menggunakan software Arduino?

### 1.3 Batasan Masalah

Dalam perancangan untuk skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

Plant yang digunakan adalah robot beroda dua dengan spesifikasi sebagai berikut:

1. Motor yang digunakan adalah jenis motor EMG 30
2. Driver motor yang digunakan EMS 5A H-Bridge keluaran innovative electronics.
3. Sensor yang digunakan adalah IMU Digital Combo Board - ITG3200/ADXL345 6DOF
4. Metode kontrol yang digunakan adalah metode Kontrol fuzzy.
5. Pusat pengontrolan adalah pada *Board Arduino Mega 2560*.
6. Arena yang digunakan memiliki permukaan datar (kemiringan  $0^0$ ).
7. Robot beroda dua hanya mempertahankan keseimbangan ketika diberi gangguan berupa kemiringan sudut.

#### **1.4 Tujuan**

Tujuan yang ingin dicapai dalam penelitian skripsi ini adalah mempertahankan posisi robot beroda dua dalam kondisi seimbang dan tegak lurus terhadap permukaan bumi di bidang datar dengan menggunakan metode kontrol logika fuzzy.

#### **1.5 Sistematika Penulisan**

Sistematika penulisan dalam laporan skripsi ini adalah:

##### **BAB I Pendahuluan**

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

##### **BAB II Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

##### **BAB III Metodologi Penelitian**

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian dan analisis data.

##### **BAB IV Perancangan dan Pembuatan Alat**

Perancangan dan perealisasi alat yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja, dan realisasi alat.

##### **BAB V Pengujian dan Analisis**

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil

pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

### **BAB VI Kesimpulan dan Saran**

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian di masa akan datang



## BAB II

### TINJUAN PUSTAKA

Kendaraan dua roda yang mempunyai keseimbangan ini mempunyai prinsip kerja berdasarkan inverted pendulum dan sangat cocok digunakan sebagai sarana transportasi pribadi. Beberapa definisi lain menyatakan bahwa robot adalah benda mirip manusia yang memiliki kemampuan untuk mengerjakan suatu pekerjaan dan bertingkah laku seperti manusia. Definisi lain yang lebih kompleks menyatakan bahwa robot adalah suatu manipulator multifungsi yang dapat diprogram untuk memindahkan alat-alat atau perlengkapan melalui bermacam-macam program untuk menyelesaikan suatu pekerjaan tertentu.

Kontrol gerak maju dan mundurnya robot beroda dua ini berdasarkan sudut derajat. Metode kontrol yang baik diperlukan agar robot tetap dalam posisi seimbang saat bergerak maju dan mundur dengan digunakan pendekatan *Kontrol Logika Fuzzy* sebagai kontrol pergerakan kendaraan ini. Metode ini merupakan metode kontrol yang cukup handal untuk diaplikasikan pada sistem yang mempunyai masukan lebih dari satu seperti pada kendaraan ini. Masukan sistem pada kendaraan ini adalah kemiringan kendaraan yang diperoleh dari sensor *gyroskop*.

#### 2.1 Penelitian Terdahulu

Penelitian *robot beroda dua* pernah dilakukan sebelumnya. Pada penelitian Muhammad Miftahur Rokhmah, 2012, dengan judul *Implementasi Sistem Keseimbangan Robot Beroda Dua Dengan Menggunakan Kontroler Proporsional Integral Diferensial*. Dari kontroler tersebut dapat membuat robot beroda dua berdiri seimbang dan tegak lurus.

Sistem yang dijadikan obyek penelitian adalah sistem *Robot Beroda Dua (Mobile Robot)* yang dapat mempertahankan kesetimbangan pada posisi ekuilibrium dengan atau tanpa gangguan. Kontrol yang digunakan adalah kontrol PID yang mampu mempertahankan posisi kesetimbangan pada roda kanan dan kiri.

Hampir 90% sistem kendali yang digunakan di lapangan saat ini adalah pengendali PID. Akan tetapi, pengendali ini memiliki beberapa kelemahan, karena pada dasarnya pengendali ini tidak dirancang untuk sistem non linier dengan banyak memiliki ketidakpastian (*uncertainties*). Dalam sistem pengendalian modern dibutuhkan pengendali yang bekerja dengan baik pada kondisi yang memiliki banyak ketidakpastian. Ketidakpastian adalah sebuah sistem dapat berupa gangguan eksternal, ketidakpastian model, ketidakpastian parameter, ataupun error yang muncul

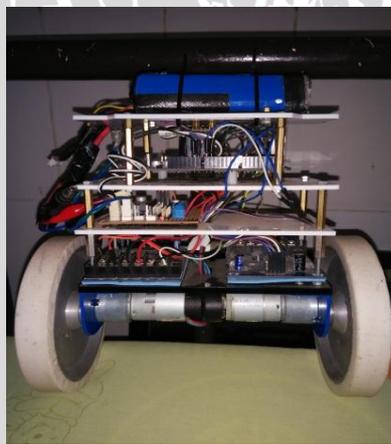
pada saat linierisasi. Ketidakpasian – ketidakpastian ini dapat mempengaruhi kestabilan sistem jika tidak diantisipasi oleh sistem pengendali, sehingga untuk menjaga kestabilan sistem diperlukan sistem pengendalian yang bersifat robust terhadap segala ketidakpastian sistem.

Pada perkembangan selanjutnya mekanik Robot beroda dua ini dibuat ulang oleh Goegoes Dwi Nusantoro. Mekanik yang terdiri dari bahan acrylic serta terdapat dua roda yang digerakan oleh dua motor kanan dan motor kiri.



**Gambar 2.1** Mekanik Robot Beroda Dua pada penelitian

Sumber: M. Miftahur.2012



**Gambar 2.2** Mekanik pada penelitian terbaru

Sumber: Goegoes Dwi Nusantoro,2013

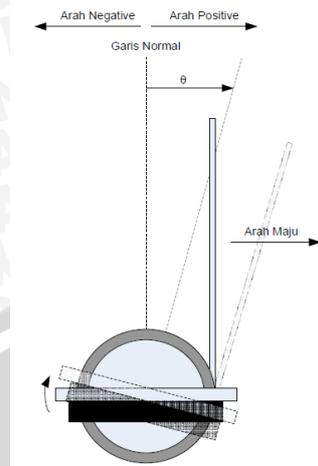
## 2.2 Robot Beroda Dua

Kata “robot” pertama kali digunakan oleh penulis Cekoslovakia bernama Karel Capek, dalam dramanya yang berjudul “*RUR: Rossum’s Universal Robot*”. Kata “robot” dalam bahasa Cekoslovakia berarti pekerja. Sejak saat itu kata robot banyak digunakan untuk

menyebut benda yang memiliki kemampuan untuk menyelesaikan suatu pekerjaan tertentu. Beberapa definisi lain menyatakan bahwa robot adalah benda mirip manusia yang memiliki kemampuan untuk mengerjakan suatu pekerjaan dan bertingkah laku seperti manusia. Definisi lain yang lebih kompleks menyatakan bahwa robot adalah suatu manipulator multifungsi yang dapat diprogram untuk memindahkan alat-alat atau perlengkapan melalui bermacam-macam program untuk menyelesaikan suatu pekerjaan tertentu. Robot didesain untuk membantu pekerjaan manusia dengan beberapa kelebihan dibandingkan pekerja manusia. Pekerjaan yang diselesaikan robot memiliki hasil yang lebih akurat dan waktu yang dibutuhkan lebih cepat, dengan biaya yang lebih murah. Kekurangan yang dimiliki robot adalah tidak fleksibel. (Fahmi Muhamadi, 2010)

Robot didesain untuk menyelesaikan satu atau beberapa pekerjaan yang spesifik, sedangkan manusia dapat melakukan banyak pekerjaan tanpa harus diprogram. Seiring dengan perkembangan zaman, robot kini memiliki program dan fungsi yang lebih kompleks. Robot dapat dikelompokkan menjadi beberapa jenis sesuai dengan spesifikasinya, misalkan fungsi dan alat gerak. Sesuai fungsinya, ada beberapa jenis robot, diantaranya robot penjelajah, robot industri, robot pemadam api, robot medis, robot nano, robot perang, dan robot kompetisi. Jenis robot sesuai alat geraknya adalah robot beroda, dan robot berkaki. Salah satu kategori dalam robot beroda adalah robot beroda dua

Robot beroda dua adalah robot yang mempunyai dua buah penggerak yang dihubungkan ke roda yang terdapat disebelah kanan dan kiri robot. Cara kerja secara general untuk dapat menyeimbangkan robot ini adalah ketika robot akan terjatuh kesisi depan, maka untuk menstabilkan/menyeimbangkannya harus mengirimkan sinyal kontrol yang sesuai ke motor DC sehingga motor DC berputar ke arah depan dan mampu mengikuti pergerakan robot dan dapat kembali ke posisi semula, begitu juga sebaliknya jika robot akan terjatuh kearah belakang untuk menstabilkan/menyeimbangkannya harus mengirimkan sinyal kontrol yang sesuai ke motor DC sehingga motor DC berputar ke arah belakang dan mampu mengikuti pergerakan robot sehingga dapat kembali ke posisi semula. Robot beroda dua ini pada dasarnya seperti pendulum terbalik yang sistemnya tidak stabil. sehingga diperlukan sebuah kontroler yang baik dan efisien sehingga mampu membuat robot tersebut stabil.



Gambar 2.3 Sketsa Robot Beroda Dua

## 2.3 ROBOGAMES

*ROBOGAMES* adalah suatu perlombaan robot tingkat internasional yang diselenggarakan di San Mateo County Fairgrounds, Amerika Serikat. Perlombaan ini diikuti oleh 37 negara, diantaranya adalah Indonesia dan terdapat 50 kategori perlombaan yang dilombakan, diantaranya adalah *Balancer Race*. *Balancer Race* merupakan kategori perlombaan yang mempunyai tujuan menyelesaikan lintasan perlombaan dalam waktu terpendek dan dalam keadaan seimbang. Sebuah robot *balancing* didefinisikan sebagai sebuah robot beroda dua atau satu yang dapat menjaga keseimbangannya tanpa dukungan eksternal. Dalam kategori *Balancer Race* terdapat dua jenis robot yaitu otomatis dan dikendalikan oleh *remote controlled*

## 2.4 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata, K., 1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input* dan *output*

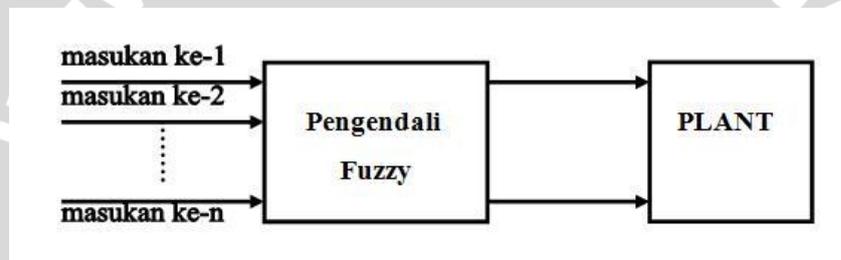
### 2.4.1 Kontrol Logika Fuzzy

Kontrol Logika *Fuzzy* adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan *fuzzy* yang mempresentasikan mekanisme pengambilan

keputusan. Aturan yang dibuat digunakan untuk memetakan variabel input ke variabel output dengan pernyataan If-Then. Kotroler ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzziikasi. Lalu dengan *fuzzy, inference engine* yang akan menentukan hasil keluaran *fuzzy*. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses defuzzifikasi.

Pengendali *fuzzy* secara umum merupakan pengendali kalang tertutup. Struktur dasar pengendali logika fuzzy terdiri dari unit *fuzzifikasi*, mekanisme penentuan keputusan dan basis aturan (*inference system*), dan unit *defuzzifikasi*.

Dalam sistem pengaturan dengan logika *fuzzy* dilibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpankan satu atau lebih keluaran ke plant atau blok lain sebagaimana ditunjukkan dalam Gambar 2.4



Sumber: DonaldCoughanowr,1991

Komponen utama penyusun *Fuzzy Logic Controller* adalah unit fuzzifikasi, *fuzzy inference*, basis pengetahuan dan unit defuzzifikasi.

Basis pengetahuan terdiri dari dua jenis (Yan : 94)

- Basis data : Mendefinisikan parameter *fuzzy* sebagai bagian dari himpunan *fuzzy* dengan menentukan batas-batas fungsi keanggotaan pada semesta pembicaraan untuk tiap-tiap variabel.
- Basis aturan : Memetakan nilai masukan *fuzzy* menjadi nilai keluaran *fuzzy*.

#### 2.4.2 Fungsi Keanggotaan

Fungsi keanggotaan menotasikan nilai kebenaran anggota-anggota himpunan *fuzzy*. Interval nilai yang digunakan untuk menentukan f

ungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan *crisp* ke semesta himpunan *fuzzy*.

Suatu himpunan *fuzzy*  $A$  dalam semesta pembicaraan  $U$  dinyatakan dengan fungsi keanggotaan,  $\mu_A$  yang harganya berada dalam interval  $[0,1]$  (Kuswadi, 2000:27). Secara matematika hal ini dinyatakan dengan :

$$\mu_A : U \rightarrow [0,1] \quad (2.1)$$

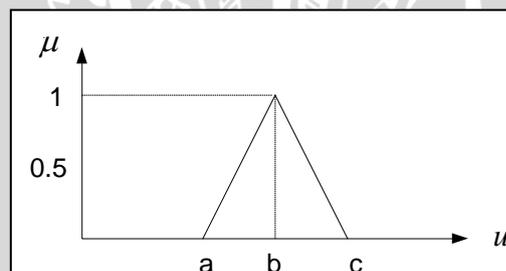
Berikut ini jenis fungsi keanggotaan yang akan digunakan yaitu fungsi keanggotaan  $T$  (*triangular*).

### 1. Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular sebagai berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2.2)$$

Fungsi keanggotaan bentuk Triangular ditunjukkan dalam Gambar 2.5



Gambar 2.5 Fungsi Keanggotaan Bentuk Triangular ( $T$ )

Sumber : Jun Yan, 1994 : 19

Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan *fuzzy* mempunyai nilai proporsional terhadap nol maupun satu.

### 2.4.3 Kendali Logika Fuzzy

Kendali logika *fuzzy* adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan *fuzzy* yang mempresentasikan mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel input ke variabel output dengan pernyataan *If - Then*.

Kontroler ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzzifikasi. Lalu dengan aturan *fuzzy*, *inference engine* yang akan menentukan hasil keluaran *fuzzy*.

Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses defuzzifikasi.

#### 2.4.4 Fuzzifikasi

Fuzzifikasi 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).

$$x = \text{fuzzifier}(x_0)$$

dengan:

$x_0$  = nilai *crisp* variabel masukan

$x$  = himpunan *fuzzy* variabel yang terdefinisi

*fuzzifier* = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan *fuzzy*

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Jun Yan, menggunakan :

1. Himpunan *fuzzy* dengan distribusi simetris.
2. Gunakan himpunan *fuzzy* dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan *fuzzy* agar saling menumpuk.

Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium

#### 2.4.5 Kaidah Atur *Fuzzy* (*Fuzzy Rule*)

*Fuzzy rule* adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat.

Dalam pengendali berbasis *fuzzy*, aturan pengaturan *fuzzy* berbentuk aturan “ IF – THEN”. Untuk sebuah sistem MISO (*Multi Input Single Output*) basis aturan pengaturan *fuzzy* berbentuk seperti berikut ini.

Rule 1 IF  $X_1$  is  $A_{11}$  AND ... AND  $x_m$  is  $A_{1m}$  THEN  $Y$  is  $B_1$

Rule 2 IF  $X_1$  is  $A_{21}$  AND ... AND  $x_m$  is  $A_{2m}$  THEN  $Y$  is  $B_2$

⋮  
⋮  
⋮

Rule n IF  $X_n$  is  $A_{n1}$  AND ... AND  $x_m$  is  $A_{nm}$  THEN  $Y$  is  $B_n$

Dengan  $X_j$  merupakan variabel masukan sistem,  $A_{ij}$  merupakan *fuzzy set* untuk  $X_j$ ,  $Y$  merupakan variabel keluaran sistem,  $B_i$  merupakan *fuzzy set* untuk  $Y$ , AND adalah operator *fuzzy*.

#### 2.4.5 Metode Inferensi

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran.

Ada beberapa cara untuk mengidentifikasi aturan mana yang akan dipakai dengan menggunakan nilai keanggotaan masukan. Menurut Jun Yan, diantara bermacam-macam metode inferensi *fuzzy* ada dua metode yang paling sering digunakan pada kendali logika *fuzzy*, salah satunya yaitu :

##### Metode Inferensi MAX – MIN

Pada metode Max – Min aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan minimum adalah

$$\mu_{C_i} = \bigcup_1^n \alpha_i \wedge \mu_{C_i} \quad (2.3)$$

dengan  $\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0)$

Sebagai contoh, terdapat dua basis kaidah atur *fuzzy*, yaitu :

$R_1$  : Jika x adalah  $A_1$  dan y adalah  $B_1$  maka z adalah  $C_1$

$R_2$  : Jika x adalah  $A_2$  dan y adalah  $B_2$  maka z adalah  $C_2$

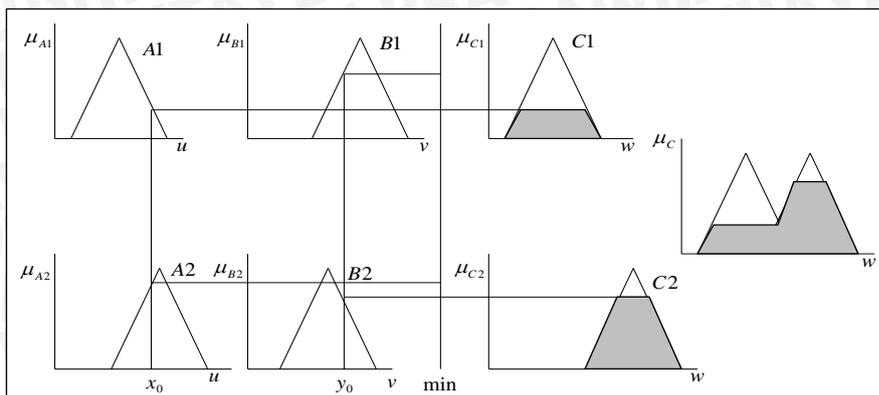
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan

$$\mu_{C_i}(W) = \mu_{C_i1} \vee \mu_{C_i2} = [\alpha_1 \wedge \mu_{C_i1}(w)] \vee [\alpha_2 \wedge \mu_{C_i2}(w)] \quad (2.4)$$

$$\text{dimana } \alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (2.5)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \quad (2.6)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.6



Gambar 2.6 Inferensi Fuzzy dengan Metode MAX-MIN

Sumber : Jun Yan, 1994 : 55

### 2.4.6 Defuzzifikasi

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data *fuzzy* yang dihasilkan dari proses inferensi (Yan, 1994 : 55). Proses defuzzifikasi dinyatakan sebagai berikut :

$$y_0 = \text{defuzzifier}(y) \tag{2.7}$$

dengan:

$y$  : aksi kontrol *fuzzy*.

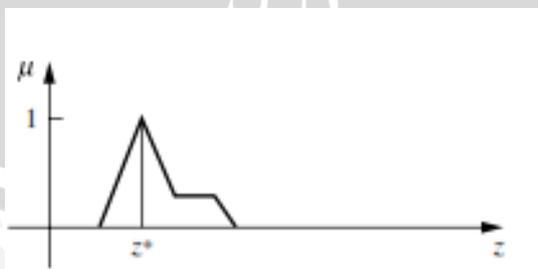
$y_0$  : aksi kontrol *crisp*.

*defuzzifier* : operator *defuzzifikasi*

Pada penelitian ini digunakan metode *max membership principle* (Ross, Timothy J. 1997). Metode ini dikenal juga sebagai metode tinggi, skema interbatas pada *output* fungsinya memuncak. Metode ini didefinisikan sebagai berikut:

$$\mu_C(z^*) \geq \mu_C(z) \text{ untuk semua nilai } z \in Z \tag{2.8}$$

dengan  $z^*$  ialah nilai terdefuzzifikasi dan bisa dilihat pada Gambar 2.12.



Gambar 2.7 Metode Max Membership Principle

Sumber: Timothy J. Ross, 1997

### 2.5 Motor DC

### 2.5.1. Prinsip Kerja Motor DC

Prinsip kerja motor DC *brushed* sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri. Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B.I.L \text{ (Newton)} \dots \dots \dots (1)$$

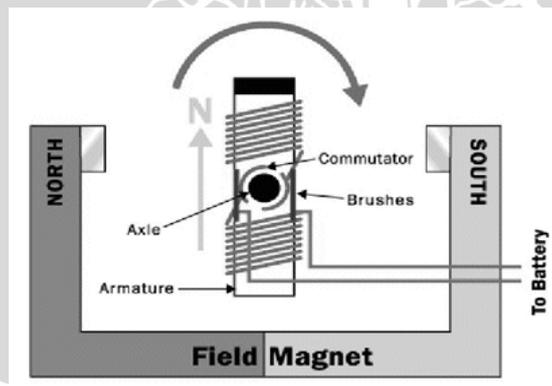
Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)

Dalam hal ini B adalah kerapatan *fluks* magnet ( $Wb/m^2$ ), I adalah arus yang mengalir (A) dan L adalah panjang kawat (m). Gambar penampang melintang dari motor DC ditunjukkan dalam Gambar 2.8



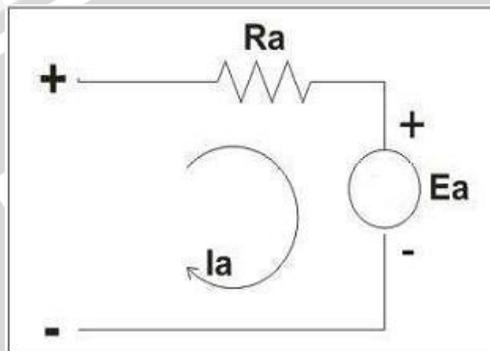
Gambar 2.8 Penampang melintang motor DC

Sumber: [www.electronics-scheme.com](http://www.electronics-scheme.com)

Persamaan merupakan prinsip dasar sebuah motor, di mana terjadinya proses perubahan energy listrik ( $I$ ) menjadi energi mekanik ( $F$ ). Jika motor mempunyai jari-jari sebesar  $r$ , maka akan menimbulkan torsi sebesar :

$$\tau = F.r = B.I.L.r \dots (Nm) \dots \dots \dots (2)$$

Pada saat dibangkitkan, konduktor akan bergerak di dalam medan magnet dan akan menimbulkan gaya gerak listrik (ggl) yang merupakan reaksi lawan terhadap tegangan penyebabnya. Proses konversi energi listrik menjadi energi mekanik dapat berlangsung jika tegangan sumber lebih besar dari gaya gerak listrik lawan (Soemarwanto,1997). Gambar 2.9 menunjukkan proses kerja motor DC.



Gambar 2.9 Proses Kerja Motor DC  
 Sumber: [www.mikron123.com](http://www.mikron123.com)

Motor dapat berputar jika tegangan masukan motor lebih besar dari ggl yang timbul. Hubungan antara tegangan sumber dan ggl lawan seperti ditunjukkan dalam Gambar 2.6 dapat dirumuskan sebagai berikut:

$$E_a = V_{in} - I_a \cdot R_a \dots (V) \dots \dots \dots (3)$$

Dalam hal ini  $E_a$  adalah tegangan pada jangkar,  $V_{in}$  adalah tegangan masukan,  $I_a$  adalah arus jangkar dan  $R_a$  adalah tahanan jangkar, sedangkan induksi yang timbul adalah:

$$E_a = Cn \Phi \dots (V) \dots \dots \dots (4)$$

Dengan  $C$  adalah konstanta,  $n$  adalah kecepatan motor, dan  $\Phi$  adalah fluks magnetik yang besarnya sebanding dengan arus penguatan torsi. Torsi pada motor juga sebanding dengan fluks magnetik dan arus. Hal ini ditunjukkan dalam Persamaan berikut :

$$\tau = C\Phi I_a \dots (Nm) \dots \dots \dots (5)$$

Jika diketahui kecepatan sudut  $\omega$  adalah

$$\omega = \frac{2\pi \cdot n}{60} \dots (rad/s)$$

**2.5.2. Pengaturan Motor DC**

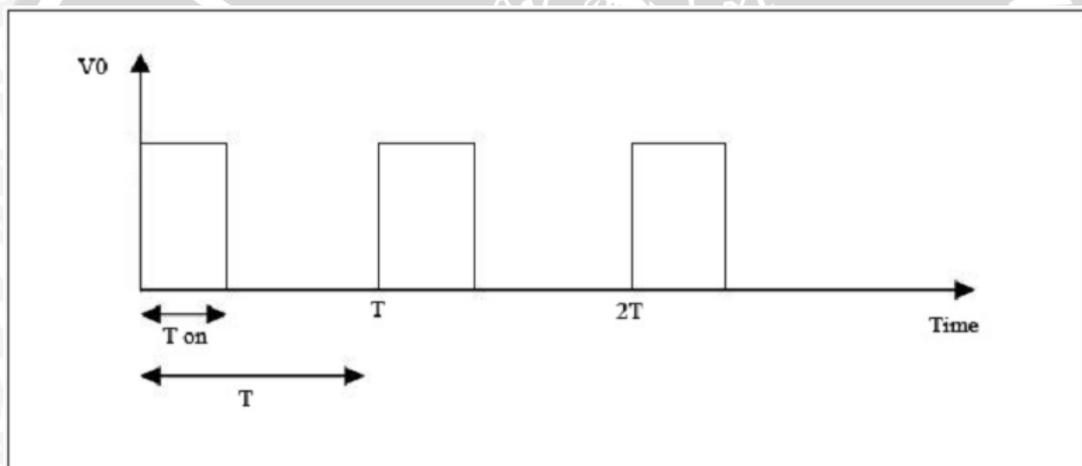
Apabila Persamaan 2.9 disubstitusikan ke dalam Persamaan 2.10 akan didapatkan rumus kecepatan motor ( $n$ ) sebagai berikut :

$$n = \frac{V_{in} - I_a R_a}{C\Phi} \dots (\text{rpm})$$

Dari Persamaan terlihat bahwa kecepatan motor sebanding dengan tegangan masukan ( $V_{in}$ ) (Zuhal, 1993). Jadi apabila tegangan masukan besar maka kecepatan motor akan cepat, demikian pula sebaliknya, jika tegangan masukan kecil maka kecepatan motor akan lambat.

### 2.5.3 PWM (Pulse Width Modulation)

PWM (Pulse Width Modulation) digunakan untuk mengatur kecepatan dari motor DC. Di mana kecepatan motor DC tergantung pada besarnya *Duty cycle* yang diberikan pada motor DC tersebut. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *Duty cycle* bervariasi dari 0%-100%. Dengan mengatur *Duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (Pulse width Modulation) secara umum dapat dilihat dalam gambar 2.7.



Gambar 2.10. Sinyal PWM Secara Umum  
Sumber: [www.electronics-scheme.com](http://www.electronics-scheme.com)

$$\text{Dutycycle} = \frac{T_{on}}{T} \times 100\% \dots (\%)$$

(1.8)

Dengan :

$T_{on}$  = Periode logika tinggi

$T$  = Periode keseluruhan

$$V_{dc} = \text{Dutycycle} \times V_{cc} \dots (V)$$

(1.9)

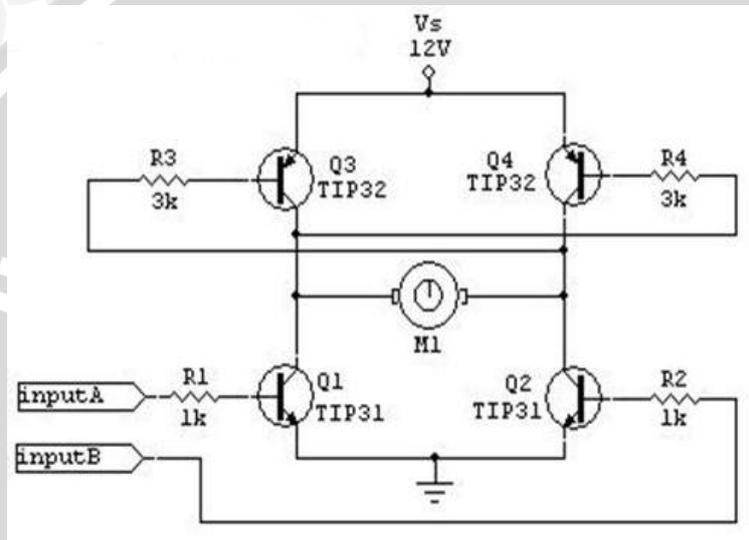
Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut :

$$F_{0n} = \frac{f_{clk} I/O}{N \cdot 256} \dots (Hz)$$

(1.10)

#### 2.5.4 Driver Motor H-Bridge

Rangkaian driver motor DC ini disebut dengan h-bridge dikarenakan konfigurasi/susunan transistornya seperti membentuk huruf H. Transistor – transistor ini digunakan sebagai switching sehingga nantinya motor dapat berputar searah jarum jam (*clockwise*) dan berlawanan arah jarum jam (*counterclockwise*).



**Gambar 2.11.** Rangkaian H-Bridge Driver  
Sumber : <http://elektronikaunej.blogspot.com>

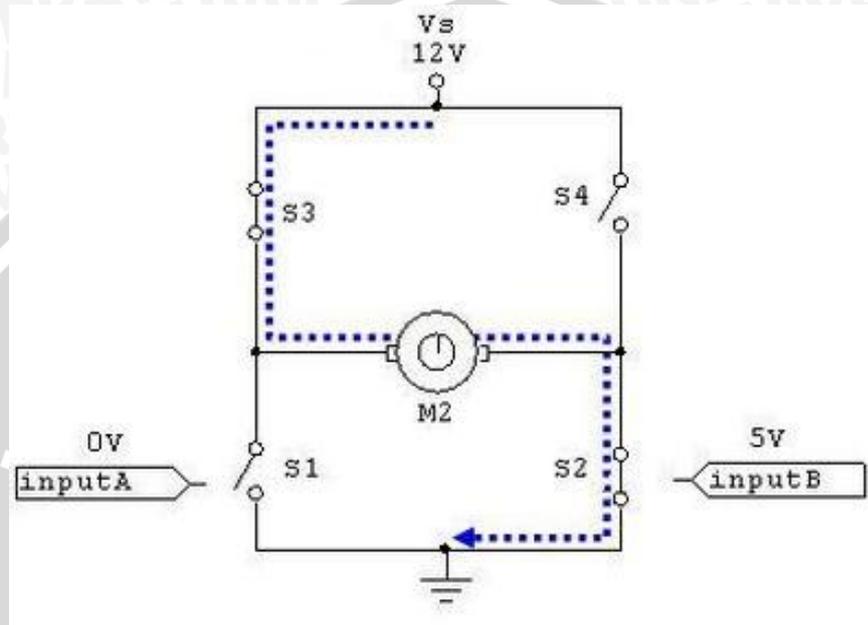
Cara kerja rangkaian h-bridge sebagai berikut :

1. A = B = '0'

Karena input A dan B mempunyai logika yg sama '0' (0V), maka kedua transistor TIP31 ( $Q_1$  &  $Q_2$ ) tidak akan mendapat picuan pada basisnya sehingga transistor bersifat cut-off atau transistor bersifat seperti saklar yg terbuka. Dari rangkaian diatas terlihat pula bahwa kedua TIP 32 ( $Q_3$  &  $Q_4$ ) bergantung pada TIP31 dimana basis kedua TIP32 terhubung pada kolektor TIP 31. Jadi, apabila tidak ada arus yg mengalir pada kolektor TIP 31 maka basis TIP 32 juga tidak akan terpicu akibatnya motor tidak akan berputar atau berhenti.

2. A = '0'; B = '1'

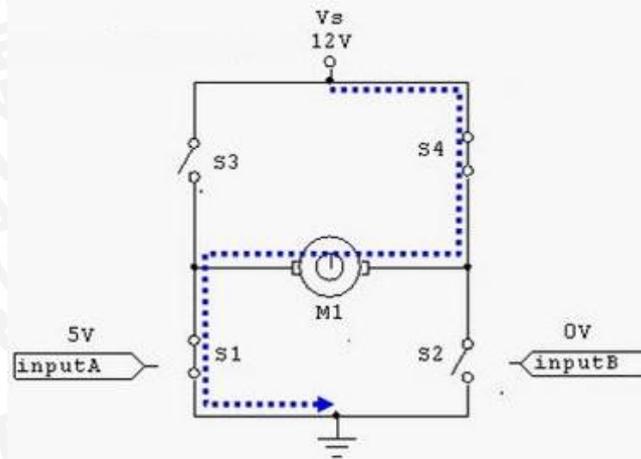
Saat input A diberi logika '0' (0V) dan input B diberi logika '1' (5V) maka  $Q_2$  akan saturasi sedangkan  $Q_1$  tetap cut-off. Karena  $Q_2$  bersifat saturasi atau seperti saklar yang tertutup maka basis  $Q_3$  akan mendapat picuan sehingga  $Q_3$  juga bersifat saturasi. Akibatnya arus akan mengalir dengan urutan seperti berikut :  $V_s - Q_3 - \text{motor} - Q_1 - \text{ground}$ , sehingga motor akan berputar searah jarum jam.



**Gambar 2.11.** Rangkaian H-Bridge saat A = '0'; B = '1'  
Sumber : <http://elektronikaunej.blogspot.com>

### 3. A = 1; B = 0

Saat input A diberi logika '1' (5V) dan input B diberi logika '0' (0V) maka  $Q_1$  akan saturasi sedangkan  $Q_2$  cut-off. Akibatnya  $Q_4$  juga akan menjadi saturasi karena basis  $Q_4$  mendapat picuan dari  $Q_1$ . Sehingga arus akan mengalir dengan urutan seperti berikut :  $V_s - Q_4 - \text{motor} - Q_1 - \text{ground}$  dan motor akan berputar berlawanan arah jarum jam.



**Gambar 2.12.** Rangkaian H-Bridge saat A = '1'; B = '0'  
 Sumber : <http://elektronikaunej.blogspot.com>

#### 4. A = B = '1'

Jika kedua input diberi logika '1' secara bersamaan maka akan mengakibatkan semua transistor dalam kondisi saturasi. Secara logika motor tidak akan berputar karena tidak ada beda potensial pada ujung – ujung konektornya. Namun hal ini akan menyebabkan timbulnya panas yang berlebihan pada semua transistor sehingga dapat menyebabkan kerusakan. Oleh karena itu hal ini harus dihindari

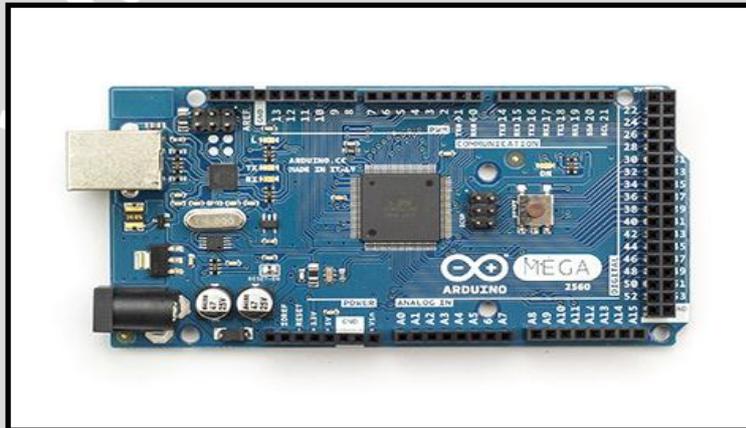
## 2.6 Arduino Mega 2560

Arduino Mega merupakan sebuah mikrokontroler berdasarkan ATmega2560. Memiliki 54 digital input / output pin (yang 14 dapat digunakan sebagai output PWM), inputan analog 16, 4 UART (*port serial hardware*), osilator kristal 16MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino juga dapat didukung melalui koneksi USB dari komputer, dari baterai 9V, atau dari catu daya. Arduino dapat dikontrol dari komputer atau diprogram oleh komputer dan kemudian terputus dan memungkinkan untuk bekerja secara independen. Arduino Mega telah kompatibel dengan sebagian shield yang dirancang untuk Arduino Duemilanove atau Diecimila. Berikut adalah karakteristik dari Arduino Mega 2560:

Microcontroller	Atmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Gambar board Arduino Mega dapat ditunjukkan pada gambar di bawah ini.



**Gambar 2.13.** Arduino Mega  
Sumber : <http://arduino.cc>

Perangkat lunak untuk pemrograman Arduino mudah dan juga tersedia secara bebas untuk Windows, Mac, dan komputer LINUX.

### 2.6.1 Daya

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Eksternal (non-USB) daya dapat berasal baik dari adaptor AC-DC atau baterai. Papan arduino dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika daya kurang dari 7V, namun, 5V pin dapat menyediakan daya kurang dari lima volt dan papan arduino mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Rentang yang dianjurkan adalah 7 sampai 12 volt. Pin listrik adalah sebagai berikut:

1. VIN. Tegangan input ke papan Arduino ketika menggunakan sumber daya eksternal (sebagai lawan 5 volt dari koneksi USB atau regulasi daya lain). Anda dapat menyediakan tegangan melalui pin ini, atau jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
2. 5V. Merupakan pin output 5V diatur dari regulator di papan arduino. Papan arduino dapat diaktifkan dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN dari papan arduino (7-12V). Menyediakan tegangan melalui 5V atau 3.3V pin melewati regulator, dan dapat merusak papan arduino. Kami tidak menyarankan itu.
3. 3V3. Sebuah pasokan 3,3 volt dihasilkan oleh regulator didalam papan arduino. Arus maksimumnya adalah 50 mA.
4. GND. Pin tanah.
5. IOREF. Pin ini pada papan Arduino memberikan tegangan referensi seperti mikrokontroler. Sebuah perisai dikonfigurasi dengan benar dapat membaca pin tegangan IOREF dan pilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

### 2.6.2 Memory

ATmega 2560 memiliki 256KB dari flash memory untuk menyimpan kode (yang 8KB digunakan untuk *bootloader*), 8KB SRAM dan 4KB EEPROM (yang dapat dibaca dan ditulis dengan EEPROM *library*).

### 2.6.3 Input dan Output

Masing-masing dari 54 digital pin pada Mega dapat digunakan sebagai input atau output, menggunakan fungsi *pin Mode ()*, *digital Write ()*, dan *digital Read()*. Fungsi tersebut beroperasi pada 5volt. Setiap pin dapat memberikan atau menerima maksimum 40mA dan memiliki resisto *pull-up* internal (terputus secara *default*) dari 20-50k Ohms. Selain itu, beberapa pin memiliki fungsi khusus:

1. Serial: 0 (RX) dan 1(TX); Serial1: 19 (RX) dan 18 (TX); Serial2: 17 (RX) dan 16 (TX); Serial3: 15(RX) dan 14(TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pins 0 dan 1 juga terhubung ke pin yang sesuai dari ATmega 16 U2 USB -to-TTL Serial keping.
2. Eksternal Interupsi: 2(interrupt 0), 3(mengganggu 1), 18(mengganggu 5), 19(mengganggu 4), 20 (mengganggu 3), dan 21 (mengganggu 2). Pin ini dapat dikonfigurasi untuk memicu interrupt pada nilai yang rendah, naik atau jatuh, atau perubahan nilai. Lihat *tattach Interrupt ()* fungsi untuk rincian.

3. PWM: pin 2 sampai 13 dan 44 sampai 46. Menyediakan 8-bit PWM output dengan fungsi *analog Write()*.
4. SPI: 50(MISO), 51(MOSI), 52(SCK), 53(SS). Pin ini mendukung komunikasi SPI menggunakan *SPI library*. Pin SPI juga pecah pada header ICSP, yang secara fisik kompatibel dengan Uno, Duemilanov dan Diecimila.
5. LED: 13 Ada *built-in* LED terhubung ke pin digital 13. Ketika pin memiliki nilai TINGGI, LED menyala, ketika pin memiliki nilai RENDAH, maka mati.
6. TWI: 20(SDA) dan 21(SCL). Dukungan komunikasi TWI menggunakan *Wire Library*. Perhatikan bahwa pin ini tidak dilokasi yang sama dengan pin TWI pada Duemilanove atau Diecimila.

Arduino Mega 2560 memiliki 16 inputan analog, yang masing-masing menyediakan 10bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari tanah sampai 5 volt, meskipun memungkinkan untuk mengubah batas atas menggunakan pin AREF dan fungsi *analog Reference ()*. Ada beberapa pin lainnya di papan arduino sebagai berikut :

1. AREF. Referensi tegangan untuk input analog. Digunakan dengan *analog Reference()*.
2. Reset. Digunakan untuk me-reset mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset untuk melindungi dari sesuatu yang menghalangi papan arduino.

#### 2.6.4 Komunikasi

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega 2560 menyediakan empat UART hardware untuk TTL (5V) komunikasi serial. Sebuah ATmega 16U2 (ATmega 8U2 pada revisi 1 dan revisi 2 papan arduino) pada salah satu papan USB dan menyediakan *portcom* virtual untuk perangkat lunak pada computer (Windows akan memerlukan file *inf*, tapi OSX dan Linux akan mengakui papan arduino sebagai *port COM* secara otomatis). Perangkat lunak Arduino termasuk *monitorserial* yang memungkinkan data tekstual sederhana yang akan dikirim ke dan dari papan. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui *Chip* ATmega 8U2/ATmega 16U2 dan koneksi USB ke computer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software Serial library* memungkinkan untuk komunikasi serial pada salah satu pin digital pada Mega 2560.

ATmega 2560 juga mendukung TWI dan SPI komunikasi. Perangkat lunak Arduino termasuk *Wire library* untuk menyederhanakan penggunaan bus TWI; lihat dokumentasi untuk rincian. Untuk komunikasi SPI, menggunakan *SPI library*.

## 2.7 Pemograman

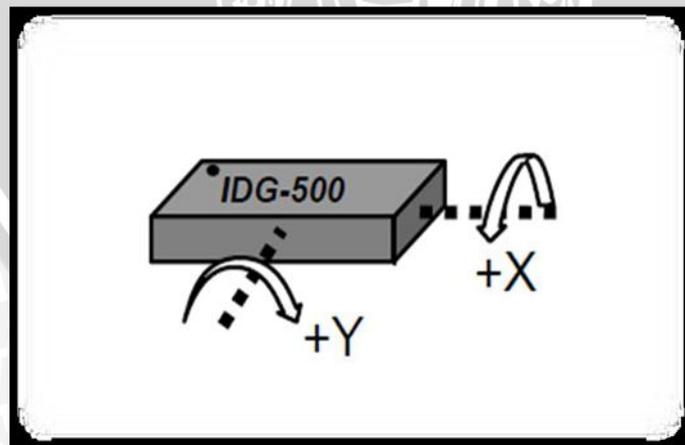
Arduino mega dapat diprogram dengan software Arduino. ATmega2560 pada Arduino mega datang *preburned* dengan *bootloader* yang memungkinkan Anda dapat meng-*upload* kode baru tanpa menggunakan programmer hardware eksternal. Komunikasi ini menggunakan protokol STK500 asli (referensi, file header C). Anda juga dapat memotong *bootloader* dan program mikrokontroler melalui ICSP (In-Circuit Serial Programming) Header menggunakan Arduino ISP atau serupa.

ATmega 16U2 (atau 8U2 dalam papan Rev1 dan rev2) sumber kode *firmware* tersedia di repositori iArduino. ATmega 16U2/8U2 diisikan dengan *bootloader* DFU, yang dapat diaktifkan oleh:

- Pada papan Rev 1: menghubungkan jumper solder di bagian belakang papan (dekat peta Italia) dan kemudian reset 8U2.
- Pada papan Rev2: ada resistor yang menarik 8U2 / 16U2 HWB baris ke ground, sehingga lebih mudah dimasukkan ke dalam mode DFU. Kemudian dapat menggunakan FLIP software Atmel (Windows) atau programme rDFU (Mac OSX dan Linux) untuk memuat firm ware baru. Dapat juga menggunakan header ISP dengan program merk sterna (Timpa *bootloader* DFU).

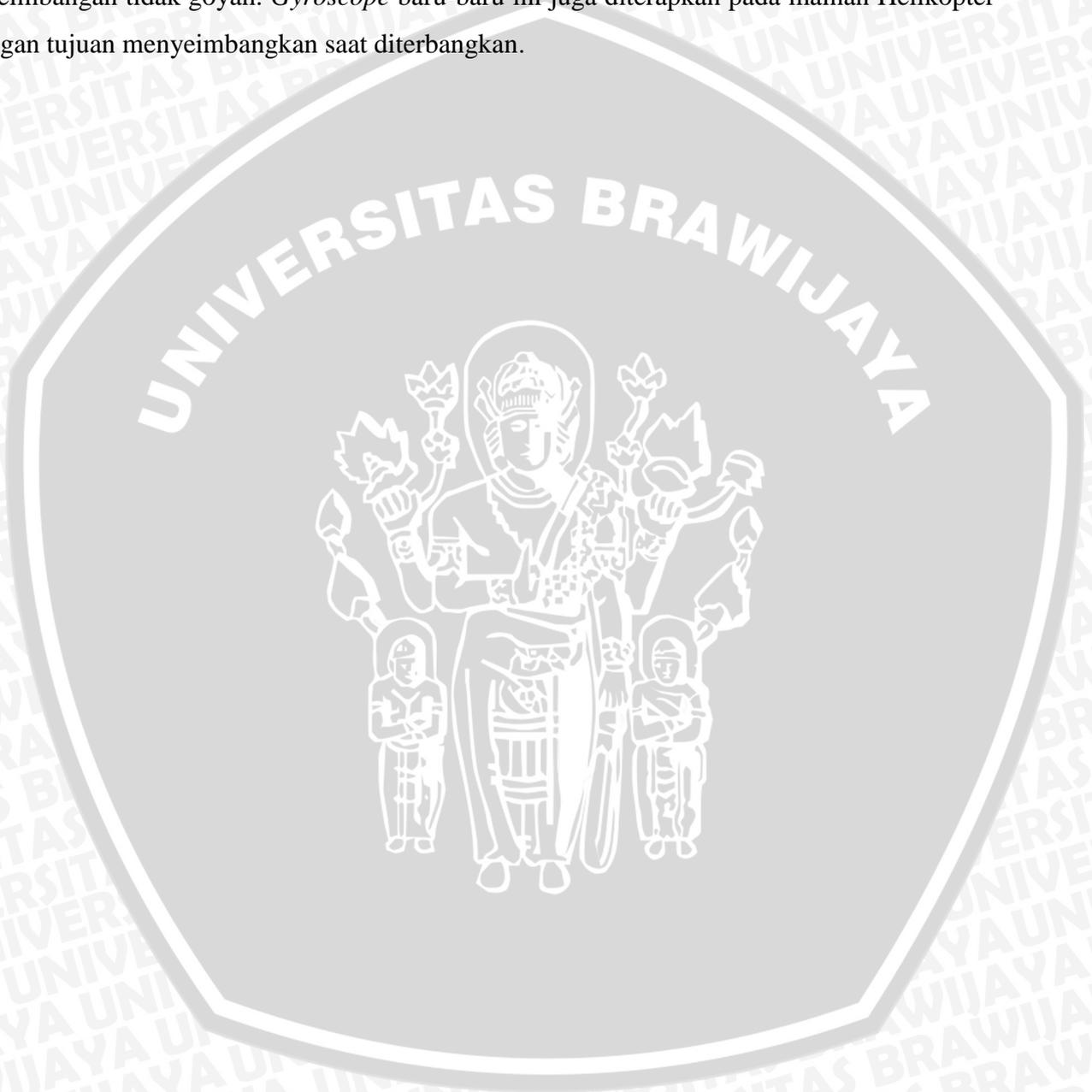
## 2.8 Sensor Gyroscope

*Gyroscope* adalah suatu alat yang berupa sensor gyro untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu. Berdasarkan keterangan tersebut, maka pergerakan *gyroscope* digambarkan seperti pada gambar 2.14



Gambar 2.14 Analogi Axis Gyroscope

*Gyroscope* memiliki *output* yang peka terhadap kecepatan sudut dari arah sumbu  $x$  yang nantinya akan menjadi sudut  $\phi$  (*roll*), dari sumbu  $y$  nantinya menjadi sudut  $\theta$  (*pitch*), dan sumbu  $z$  nantinya menjadi sudut  $\psi$  (*yaw*). Penggunaan *gyroscope* dalam kehidupan sehari-hari belum banyak. *Gyroscope* lazimnya digunakan pada Pesawat terbang, Kapal, Helikopter, dll. Hal tersebut untuk mengurangi getaran yang ditimbulkan mesin agar keseimbangan tidak goyah. *Gyroscope* baru-baru ini juga diterapkan pada mainan Helikopter dengan tujuan menyeimbangkan saat diterbangkan.



## BAB III

### METODOLOGI PENELITIAN

Pada penulisan skripsi ini akan dirancang kontroller Logika Fuzzy yang berfungsi sebagai pengendali robot beroda dua. Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan metode untuk menyelesaikan masalah tersebut.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

#### 3.1 Studi Literatur

Studi literatur dilakukan untuk mempelajari teori penunjang sistem yang dibutuhkan dalam perencanaan dan pembuatan alat. Teori yang diperlukan antara lain berkaitan dengan robot beroda dua, motor DC permanen, *H-bridge*, PWM (*Pulse Width Modulation*), kontrol logika Fuzzy, Arduino Mega, dan Sensor *gyroscope*

#### 3.2. Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut :

- 1) Robot berbahan dasar mika dan plat aluminium.
- 2) Robot beroda dua menggunakan sistem penggerak roda yang terletak disisi kiri dan kanan badan robot serta digerakkan oleh dua buah motor DC.
- 3) Robot beroda dua menggunakan *Kontroler Logika Fuzzy*, serta mampu melakukan gerak dasar yaitu maju, mundur, dan berotasi terhadap sumbu as roda (hingga  $20^{\circ}$ ) dan mampu mempertahankan keseimbangan agar robot tidak terjatuh.
- 4) Arena yang digunakan dalam melakukan pengujian memiliki permukaan datar (kemiringan  $0^{\circ}$  terhadap bidang datar).
- 5) Sistem keseimbangan robot beroda dua menggunakan modul sensor yang terdiri dari sensor *gyroscope* dengan tipe IDG-500
- 6) Sistem keseimbangan robot beroda dua menggunakan modul Arduino Mega 2560 sebagai board utama yang berfungsi untuk memproses input dari sensor menuju aktuator.

- 7) Sistem keseimbangan robot beroda dua menggunakan *Driver Motor H-Bridge 5* Asebagai modul pengendali motor DC yang dapat digunakan mengatur dua buah motor DC sekaligus.
- 8) Menggunakan catu daya rangkaian elektronik sebesar 5 V DC dan catu daya motor sebesar 12 V DC.

### **3.3. Perancangan Dan Pembuatan Alat**

#### **3.3.1. Perancangan Dan Pembuatan Perangkat Keras (*Hardware*)**

- 1). Pembuatan blok diagram lengkap sistem
- 2). Pembuatan mekanik robot
- 3). Pembuatan dan perhitungan komponen yang akan digunakan
- 4). Merakit perangkat keras masing-masing blok.

#### **3.3.2. Perancangan dan Penyusunan Perangkat Lunak**

Setelah mengetahui seperti apa perangkat keras yang dirancang, maka dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan kedalam pada Arduino Mega 2560

### **3.4 PENGUJIAN ALAT**

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) yang dilakukan baik per blok maupun keseluruhan sistem.

#### **3.4.1 PENGUJIAN TIAP BLOK**

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

#### **3.4.2 PENGUJIAN KESELURUHAN SISTEM**

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

### **3.5 PENGAMBILAN KEPUTUSAN**

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya

## BAB IV

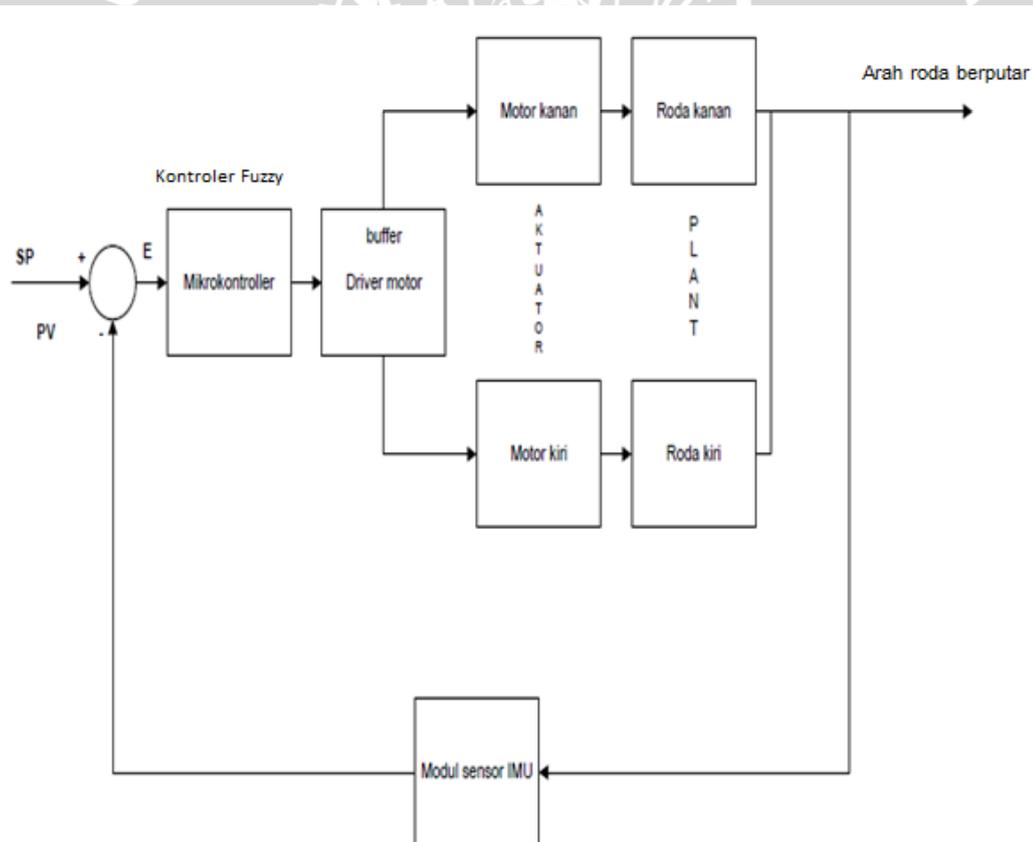
### PERANCANGAN DAN PEMBUATAN ALAT

Perancangan robot beroda dua ini dilakukan secara bertahap sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Perancangan sistem.
- Perancangan perangkat keras terdiri dari dua bagian yaitu sistem mekanik robot dan desain sistem elektronik.
- Perancangan sistem kontroler Logia Fuzzy.
- Perancangan perangkat lunak.

#### 4.1 Perancangan Sistem

Blok diagram sistem yang dirancang dibagi menjadi tiga bagian, yaitu blok mikrokontroler utama, blok sensor, dan blok pengendali motor DC seperti ditunjukkan dalam Gambar 4.1.



**Gambar 4 1.** Diagram Blok Sistem Keseimbangan Robot Beroda Dua

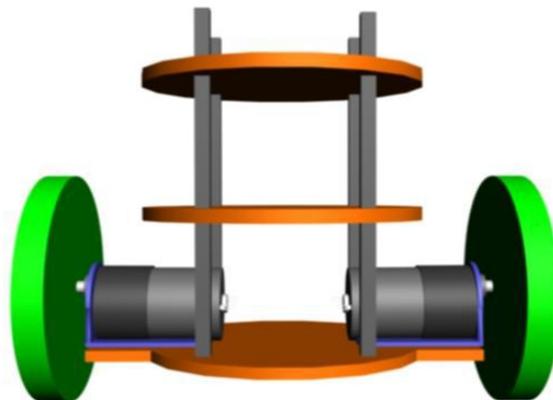
Keterangan berdasarkan dari diagram blok pada gambar 4.1 sebagai berikut :

1. *Input* berupa sudut dan apabila pertama kali robot beroda dua dinyalakan maka sudut tersebut merupakan SP (set point).
2. Mikrokontroler kemudian mengolah *input* dan menghasilkan SK (sinyal kontrol) yang kemudian digunakan untuk mengatur aktuator (motor DC).
3. Karena keluaran dari mikrokontroler tidak bisa langsung digunakan untuk mengatur aktuator, maka ditambahkan rangkaian elektrik yang berfungsi sebagai penguatan SK(Sinyal Kontrol) yaitu driver motor DC.
4. Sinyal kontrol yang telah diperkuat oleh driver motor DC digunakan untuk mengatur aktuator (motor DC) dan diteruskan pada plant (roda).
5. Keluaran sudut (*present value*) kemudian *difeedback* pada modul sensor IMU.
6. Hasil pembacaan sensor kemudian dikurangkan dengan *input* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi

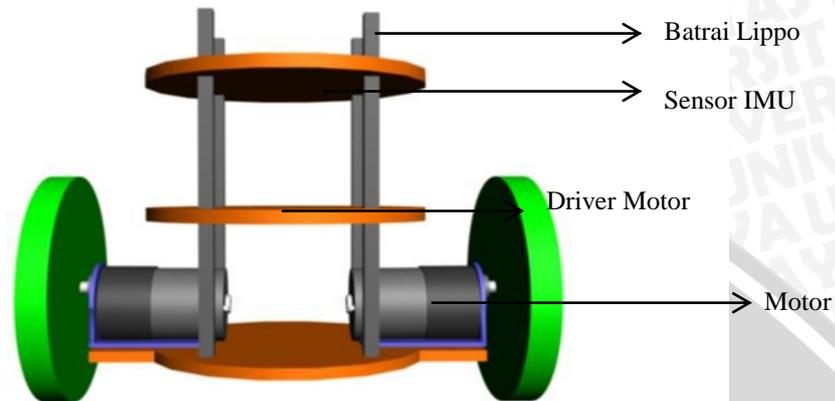
## 4.2 Perancangan Perangkat Keras

### 4.2.1 Perancangan Mekanik Robot

Sistem mekanik yang baik, mendukung pergerakan robot menjadi lebih baik, oleh karena itu perancangan mekanik dalam hal ini bodi dan rangka robot haruslah proporsional dengan panjang dan lebar serta tinggi dari robot. Gambar perspektif dan rancangan ukuran robot beroda dua ditunjukkan dalam Gambar 4.2, Gambar 4.3, dan Gambar 4.4.



**Gambar 4. 2.** Rancang Desain Mekanik Robot Beroda Dua

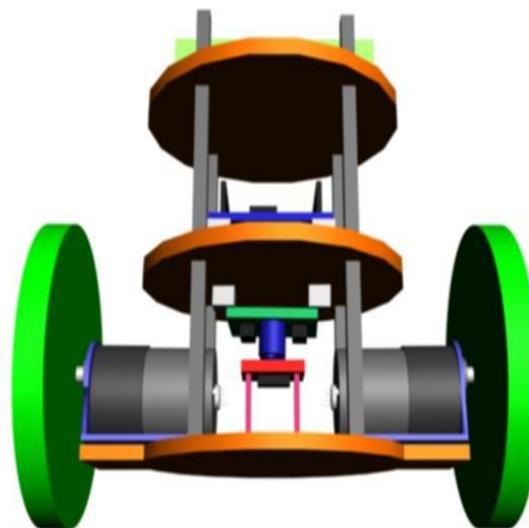


**Gambar 4.3.** Perspektif Desain Mekanik Robot Beroda Dua

Badan robot terbuat dari bahan mika *acrylic* dengan ketebalan 3 mm, ke dua buah roda berbahan nilon dengan tebal 20 mm dan berdiameter 86 mm.

#### 4.2.2 Desain Sistem Elektronik

Diagram blok sistem elektronik, terdiri dari bagian catu daya (baterai), masukan, bagian kendali, bagian keluaran. Pada bagian masukan berupa sensor *gyroscope*. Pada bagian kendali menggunakan Arduino Mega 2560. Pada bagian keluaran berupa *driver* motor sebagai penggerak aktuator robot. Gambar 4.4 menunjukkan penempatan dari rangkaian elektronik pada robot beroda dua



**Gambar 4.4** desain Sistem Robot Beroda Dua

### 1) Catu Daya Sistem

Menggunakan satu jenis catu daya, yaitu catu daya 5 V untuk rangkaian mikrokontroler Arduino Mega 2560 . Sumber catu daya yang dipakai adalah satu buah baterai lipo (*lithium polimer*) 12 V.

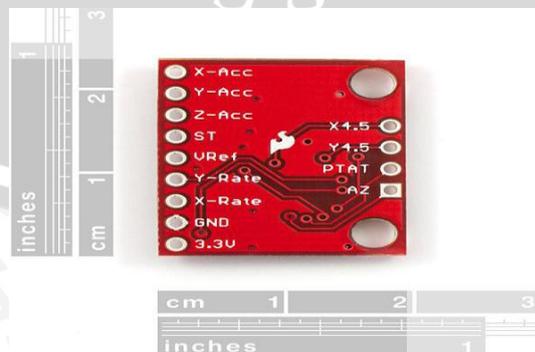
Arduino 2560 dapat bekerja jika diberi catu tegangan antara 6 V hingga 20 V sesuai dengan datasheet Arduino Mega 2560.

### 2) Modul Sensor IMU

Modul sensor IMU berfungsi sebagai sensor pengukur posisi dan sudut robot beroda dua dari posisi tegak lurus (kemiringan  $0^0$ ) terhadap permukaan bumi. Modul sensor IMU ini terdiri dari sensor Gyroscope dan sensor Accelerometer.

Gyroscope dapat membaca kecepatan sudut (angular rate) yang dinamis. Setelah melakukan komputasi menggunakan integral data dari waktu ke waktu, maka perpindahan sudut atau sudut kemiringan dapat dihitung. Tetapi sudut ini akan menjadi tidak akurat dalam jangka panjang karena efek bias yang dihasilkan oleh gyroscope.

Keluaran dari modul sensor IMU ini digunakan sebagai masukan dari kontroler Fuzzy. Sensor ini mempunyai 13 buah kaki atau pin yakni pin VCC, pin GND, pin X-Gyro, pin Y-Gyro, pin VREF, pin ST-Acc, pin Z-Acc, pin Y-Acc, pin X-Acc, pin PTAT, pin AZ, pin Y4.5 dan pin X4.5. Namun dari ketiga belas pin tersebut dalam pembuatan robot beroda ini digunakan hanya 5 buah pin. Yaitu pin VCC, pin GND, pin Y-Gyro, pin Z-Acc dan pin X-acc. Penampakan dari modul sensor gyroscope + accelerometer ditunjukkan dalam Gambar 4.5.



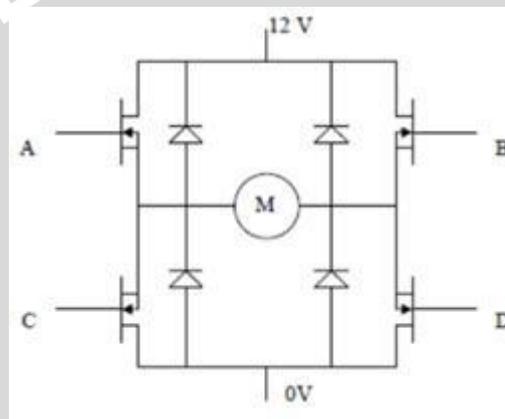
**Gambar 4.5.** IMU Analog Combo Board - 5 Degrees of Freedom IDG500/ADXL345

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

Dalam perancangan ini, digunakan sebuah modul sensor IMU yang diletakkan di bagian tengah bawah sejajar dengan sumbu as motor dan dihubungkan dengan mikrokontroler utama Arduino Mega 2560 sebagai mikrokontroler pengatur data keluaran dari sensor tersebut.

## 2) Modul Pengendali Motor DC

Modul pengendali motor DC ini digunakan untuk mengendalikan arah dan putaran motor DC *brushed* yang menjadi penggerak pada robot. Modul ini dihubungkan dengan rangkaian mikrokontroler Arduino (mikrokontroler) utama. Modul pengendali menggunakan *H-bridge* yaitu sebuah perangkat keras berupa rangkaian yang berfungsi untuk menggerakkan motor. Rangkaian ini diberi nama *H-bridge* karena bentuk rangkaiannya yang menyerupai huruf H seperti ditunjukkan pada Gambar 4.7



Gambar 4.6. Rangkaian *H-bridge*

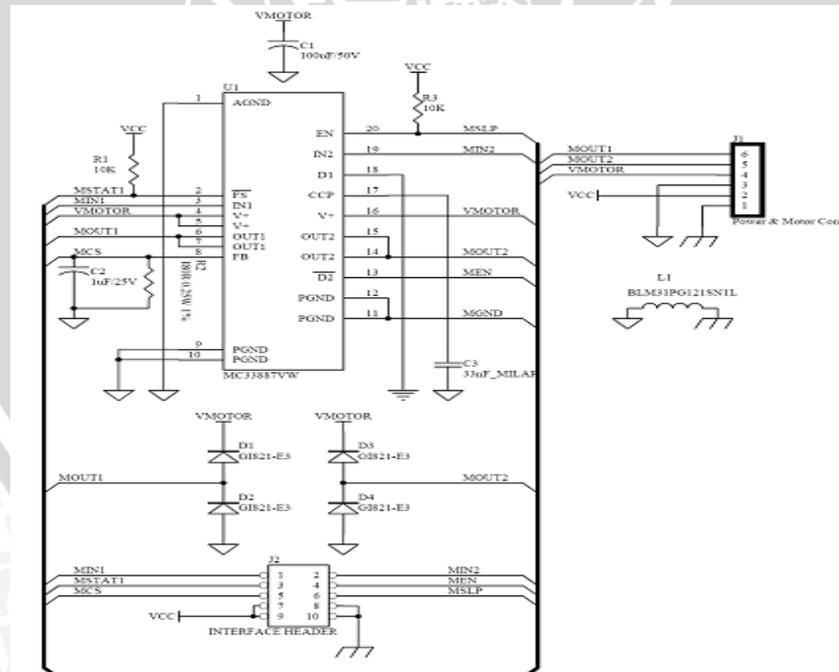
Driver motor DC tipe H-Bridge menggunakan power driver berupa transistor. Rangkaian driver motor DC H-Bridge transistor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (pulse Width Modulation) maupun metode sinyal logika dasar TTL (High) dan (Low). Untuk pengendalian motor DC dengan metode PWM maka dengan rangkaian driver motor DC ini kecepatan putaran motor DC dapat dikendalikan dengan baik.

Apabila menggunakan metode logika TTL 0 dan 1 maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum. Rangkaian driver motor DC H-Bridge ini menggunakan rangkaian jembatan transistor 4 unit dengan proteksi impuls tegangan induksi motor berupa dioda yang dipasang paralel dengan masing-masing transistor secara reversibel yang digunakan sebagai saklar



Gambar 4.7 Modul Driver Motor

Modul pengendali motor DC dengan IC VN2SP30 yang digunakan adalah modul yang dirancang oleh Pololu Robotics and Electronics dengan tipe Dual VN2SP30 Motor Driver Carrier MD03A. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC *brushed* sesuai instruksi kendali dari mikrokontroler pengguna. Dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini dapat men-drive beban-beban induktif seperti misalnya motor DC, motor *stepper*, koil relay, selenoida, dan beban-beban lainnya. Gambar rangkaian dari modul pengendali motor DC *brushed* ditunjukkan pada gambar 4.8



Gambar 4.8 Rangkaian Elektrik Driver

Sumber : Datasheet EMS 5A H-Bridge

Dengan konfigurasi *driver* dalam Gambar 4.7, maka akan didapatkan keadaan-keadaan seperti ditunjukkan dalam Tabel 4.1. Sebuah modul EMS 5A hanya dapat dihubungkan pada satu buah motor. Pin MOUT1 dan MOUT2 dihubungkan pada motor. Pin MEN merupakan pin *enable* yang digunakan sebagai pengaktif sinyal masukan pada *driver*. Pin MIN1 dan MIN2 merupakan pin yang digunakan untuk menentukan arah putaran motor. Apabila MIN1 berlogika *high* dan MIN2 berlogika *low* maka motor akan berputar searah dengan arah jarum jam. Apabila MIN1 berlogika *low* dan MIN2 berlogika *high* maka motor akan berputar berlawanan arah dengan arah jarum jam. Pin MSTAT1 merupakan pin *output digital* yang melaporkan kondisi *fault* pada modul. Pin MSTAT1 akan berlogika *low* jika ada *fault* pada modul atau *output*. Pin MSLP merupakan pin *input* yang mengatur kerja dari modul. Modul akan bekerja pada mode *full-operation* pada saat pin ini diberi logika *high* dan akan bekerja pada mode *sleep* pada saat pin ini diberi logika *low*.

Tabel 4.1 Tabel Kebenaran Driver EMS 5A H-Bridge

Status kerja modul H-Bridge	Input				Status Flag	Output	
	MSLP	MEN	MIN1	MIN2	MSTAT1	MOUT 1	MOUT 2
<i>Forward</i>	H	H	H	L	H	V MOT	MGND
<i>Reverse</i>	H	H	L	H	H	MGND	V MOT
<i>Freewheeling Low</i>	H	H	L	L	H	MGND	MGND
<i>Freewheeling High</i>	H	H	H	H	H	V MOT	V MOT
<i>Free Running Stop</i>	H	L	X	X	L	Z	Z
MIN1 tidak terhubung	H	H	Z	X	H	V MOT	X
MIN2 tidak terhubung	H	H	X	Z	H	X	V MOT
MEN tidak terhubung	H	Z	X	X	L	Z	Z
<i>Undervoltage</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Overtemperature</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Short Circuit</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Mode Sleep</i>	L	X	X	X	H	Z	Z

Keterangan :

H = High

L = Low

X = don't care

Z = High Impedance (tri-state)

#### 4.4.3 Perancangan Arduino Mega 2560

Arduino Mega merupakan sebuah mikrokontroler berdasarkan nATmega 2560. Memiliki 54 digital input / output pin (yang 14 dapat digunakan sebagai output PWM), input analog 16, 4 UART (*port serial hardware*), osilator Kristal 16MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino juga dapat didukung melalui koneksi USB dari komputer, dari baterai 9V, atau dari catu daya. Arduino dapat dikontrol dari komputer atau diprogram oleh komputer dan kemudian terputus dan memungkinkan untuk bekerja secara independen. Arduino Mega telah kompatibel dengan sebagian shield yang dirancang untuk Arduino Duemilanove atau Diecimila. Berikut adalah karakteristik dari Arduino Mega 2560:

Microcontroller	Atmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



**Gambar 4.9** Arduino Mega 2560  
Sumber: <http://arduino.cc>

rangkaian eksternal. Dalam perancangan ini, *pin* yang digunakan adalah sebagai berikut :

Motor Kanan

Pin 22 : Arah Putar Motor 1

Pin 23 : Arah Putar Motor 2

Pin 23 : Kecepatan Motor

Pin A0 : MCS (driver)

Pin 5 dan 12 : Encoder

Pin 7 dan 9 : Vcc

Pin 8 dan 10 : GND

Motor Kiri

Pin 25 : Arah Putar Motor 1

Pin 24 : Arah Putar Motor 2

Pin 9 : Kecepatan Motor

Pin A1 : MCS (driver)

Pin 26 : MSLP (driver)

Pin 2 dan 3 : Encoder

Selanjutnya, Arduino Mega2560 dihubungkan dengan komputer dengan komunikasi serial. Dihubungkan disini bertujuan agar dapat dilakukan pemrograman di komputer, selanjutnya program tersebut dapat di tanamkan pada Arduino Mega2560. Selain itu dapat digunakan sebagai catu untuk Arduino Mega2560. Gambar dari Arduino Mega2560 ditunjukkan pada gambar 4.10



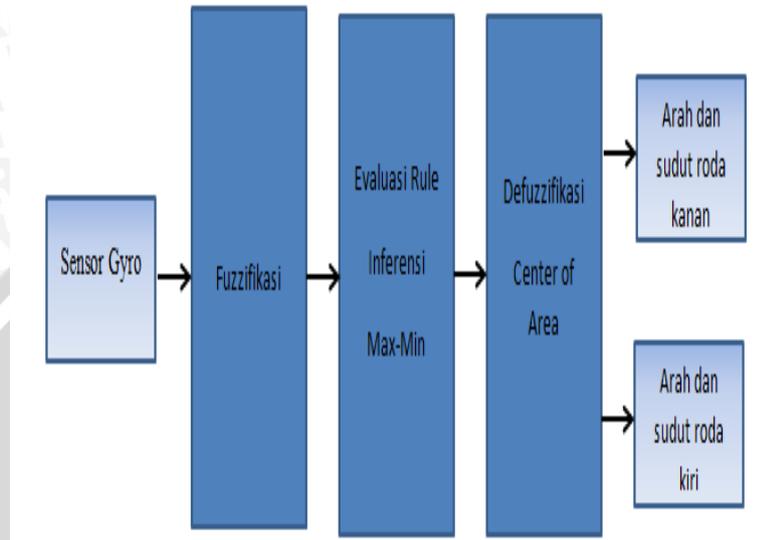
Gambar 4.10 Komunikasi Arduino

#### 4.5 Perancangan Perangkat Lunak (Software)

#### 4.5.1 Perancangan Kontrol Logika Fuzzy

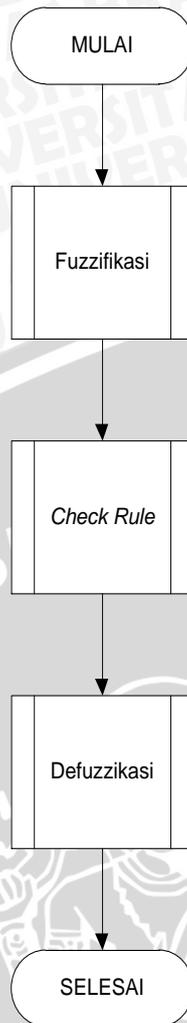
##### 4.5.1.1 Menentukan Variabel Masukan dan Variabel Keluaran

Variabel masukan untuk kontrol logika *fuzzy* terdapat dua yaitu hasil keluaran sensor *gyroscope* dan perubahan kemiringan ( $de/dt$ ) sedangkan keluaran berupa % *duty cycle* untuk PWM motor DC. Berikut gambar 4.10 menunjukkan blok diagram dari perancangan kontrol logika Fuzzy



Gambar 4.11 Diagram perancangan Kontrol Logika Fuzzy

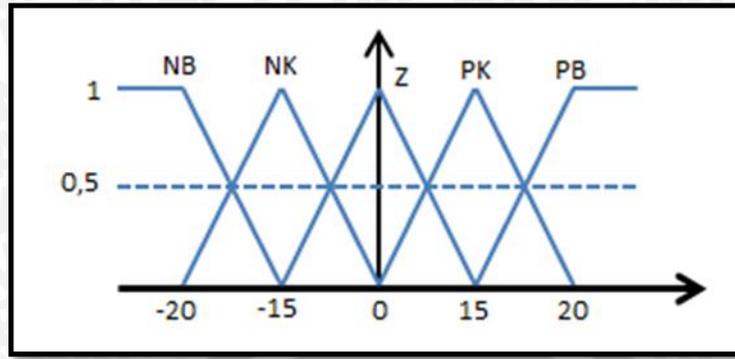




Gambar 4.12 Flowchart subrutin fuzzy

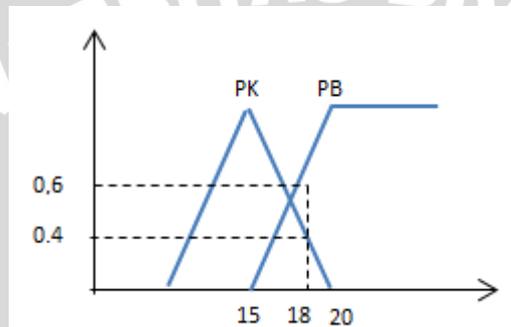
#### 4.5.1.2 Fungsi Keanggotaan Masukan

Pada fungsi keanggotaan *input* variabel sensor gyroscope terdiri dari 5 kurva *membership function* (fungsi keanggotaan), yaitu *negatif besar (NB)*, *negatif kecil (NK)*, *Zero (Z)*, *Positif kecil (PK)*, *Positif Besar (PB)*. Fungsi keanggotaan berjumlah 5 dengan tujuan agar menghasilkan aturan *fuzzy* yang banyak sehingga semakin banyak pula kondisi dan aksi kontrol yang akan dipilih. Penentuan parameter nilai batas dari tiap-tiap kurva sendiri menggunakan salah satu metode yaitu *intuition*. Metode ini nampak sederhana, itu hanyaberasal dari kapasitas manusia untuk mengembangkan fungsi keanggotaan melalui kecerdasan bawaan mereka sendiri dan pemahaman (Timothy J. Ross, 1995: 92). Fungsi keanggotaan masukan pada warna dasar merah dapat ditunjukkan pada Gambar 4.9



Gambar 4.12 Fungsi Keanggotaan Masukan sensor Gyroscope

Misalnya, data dari sensor gyro adalah masuk inputan dari fuzzy yang menjadi fungsi keanggotaan, maka derajat keanggotaan ( $\mu_G$ ) didapat dengan cara sebagai berikut

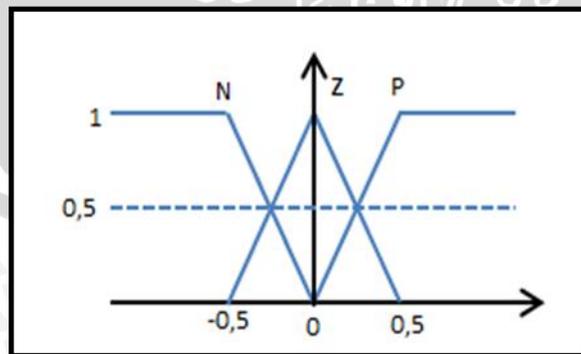


Gambar 4.13 Fungsi Keanggotaan Sensor Gyroscope ( $\mu_G$ ) pada Masukan 18

Dengan Perhitungan  $\mu_G Z = 0$ ;  $\mu_G PK = \frac{20-18}{20-15} = 0,4$  ;  $\mu_G PB = \frac{18-15}{20-15} = 0,6$

Sehingga nilai Keanggotaan  $\mu_G Z = 0$  ;  $PK = 0,4$  ; dan  $PB = 0,6$

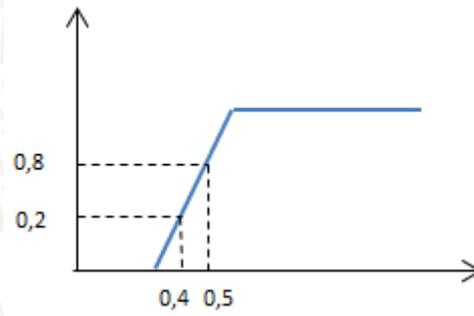
fungsi keanggotaan segitiga selanjutnya yang digunakan dalam perancangan fungsi keanggotaan *Derror* sebagai masukan fuzzy. Variabel masukan *Derror* yang dibagi dalam 3 fungsi keanggotaan yaitu *Negatif (N)*, *Zero (Z)*, *Positif (P)*



Gambar 4.13 Fungsi Keanggotaan Masukan Derror

Misalnya, data dari sensor gyro adalah masuk inputan dari fuzzy yang menjadi fungsi keanggotaan, maka derajat keanggotaan ( $\mu_G$ ) didapat dengan cara sebagai berikut





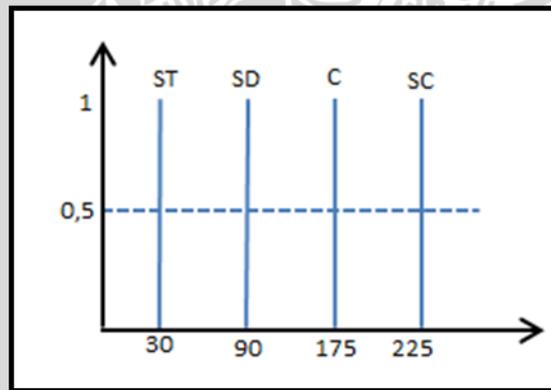
Gambar 4.14 Fungsi Keanggotaan Derror ( $\mu_{De}$ ) pada Masukan 0,4

Dengan perhitungan  $Z=0; \mu_{De} P = \frac{0,5-0,4}{0,5-0} = 0,2$  ;  $\mu_G P = \frac{0,4-0}{0,5-0} = 0,8$

Sehingga nilai Keanggotaan  $\mu_G Z=0$  ;  $P= 0,2$  ; dan  $P= 0,8$

#### 4.5.1.2 Fungsi Keanggotaan Keluaran Logika Fuzzy

Fungsi keanggotaan merupakan nilai PWM yang dikeluarkan oleh Arduino Mega berupa hasil perhitungan dengan metode defuzzifikasi *weighted average* yang telah dibulatkan dalam program. Fungsi keanggotaan keluaran pada output variable PWM didapatkan kondisi *stop (ST)*, *sedang (SD)*, *cepat (C)*, *sangat cepat (SC)*.



Gambar 4.15 Fungsi Keanggotaan Keluaran PWM

#### 4.5.1.3 Menentukan Atur Fuzzy

Aturan *fuzzy (fuzzyrule)* digunakan sebagai penentu keluaran dari fuzzifikasi yang akan diolah dalam proses defuzzifikasi. Jumlah aturan *fuzzy* yang digunakan sebanyak 15. Aturan *fuzzy* diperoleh dari 5 fungsi keanggotaan masukan *gyro* dan 3 *delta error*. Aturan *fuzzy* yang digunakan ditunjukkan dalam Tabel 4.2

Tabel 4.2 Aturan Fuzzy pada Sistem

<b>D</b> \ <b>G</b>	<b>Z</b>	<b>PK</b>	<b>NK</b>	<b>PB</b>	<b>NB</b>
<b>Z</b>	ST	SD	SD	C	C
<b>N</b>	SD	SD	C	C	SC
<b>P</b>	SD	C	SD	SC	SC

**Keterangan:**

1. If (Gyro is zero) and (Derror is Zero) then (PWM is stop) (1)
2. If (Gyro is pos\_kecil) and (Derror is Zero) then (PWM is sedang) (1)
3. If (Gyro is neg\_kecil) and (Derror is Zero) then (PWM is sedang) (1)
4. If (Gyro is pos\_kecil) and (Derror is pos) then (PWM is cepat) (1)
5. If (Gyro is pos\_kecil) and (Derror is Neg) then (PWM is sedang) (1)
6. If (Gyro is neg\_kecil) and (Derror is Neg) then (PWM is cepat) (1)
7. If (Gyro is neg\_kecil) and (Derror is pos) then (PWM is sedang) (1)
8. If (Gyro is pos\_besar) and (Derror is pos) then (PWM is scepat) (1)
9. If (Gyro is pos\_besar) and (Derror is Neg) then (PWM is cepat) (1)
10. If (Gyro is neg\_besar) and (Derror is Neg) then (PWM is scepat) (1)
11. If (Gyro is neg\_besar) and (Derror is pos) then (PWM is cepat) (1)

**4.5.1.4 Metode Referensi**

Metode referensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran.

Ada beberapa cara untuk mengidentifikasi aturan mana yang akan dipakai dengan menggunakan nilai keanggotaan masukan. Menurut Jun Yan, diantara bermacam-macam

metode inferensi *fuzzy* ada dua metode yang paling sering digunakan pada kendali logika *fuzzy*, salah satunya yaitu :

Metode Inferensi MAX – MIN

Pada metode Max – Min aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan minimum adalah

$$\mu_{C'} = \bigcup_1^n \alpha_i \wedge \mu_{Ci} \quad (2.3)$$

dengan  $\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0)$

Sebagai contoh , terdapat dua basis kaidah atur *fuzzy*, yaitu :

R<sub>1</sub> : Jika x adalah A<sub>1</sub> dan y adalah B<sub>1</sub> maka z adalah C<sub>1</sub>

R<sub>2</sub> : Jika x adalah A<sub>2</sub> dan y adalah B<sub>2</sub> maka z adalah C<sub>2</sub>

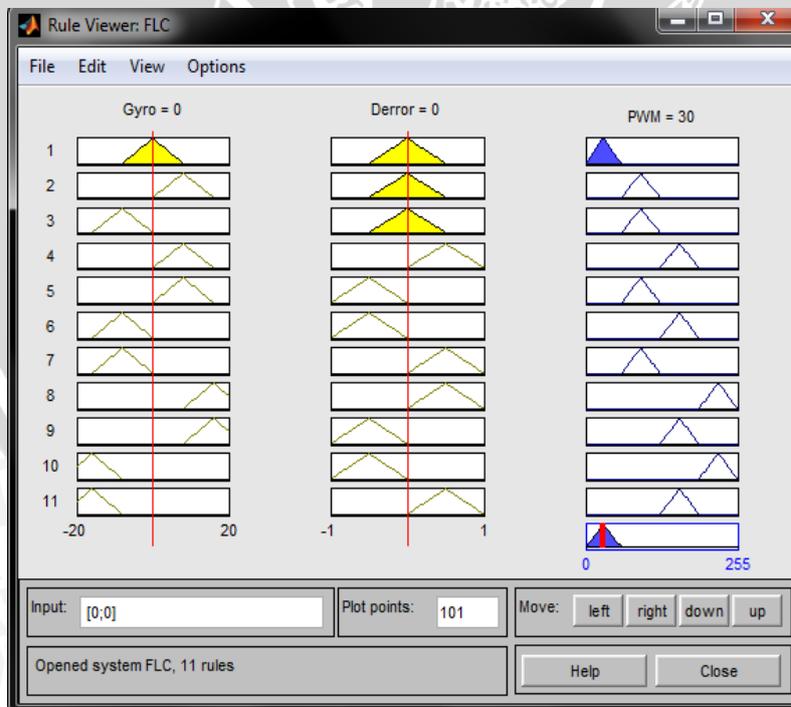
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan

$$\mu_{C'}(W) = \mu_{c_1} \vee \mu_{c_2} = [\alpha_1 \wedge \mu_{c_1}(w)] \vee [\alpha_2 \wedge \mu_{c_2}(w)] \quad (2.4)$$

dimana  $\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$  (2.5)

$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$  (2.6)

Lebih jelas metode ini dideskripsikan dalam Gambar 4.15



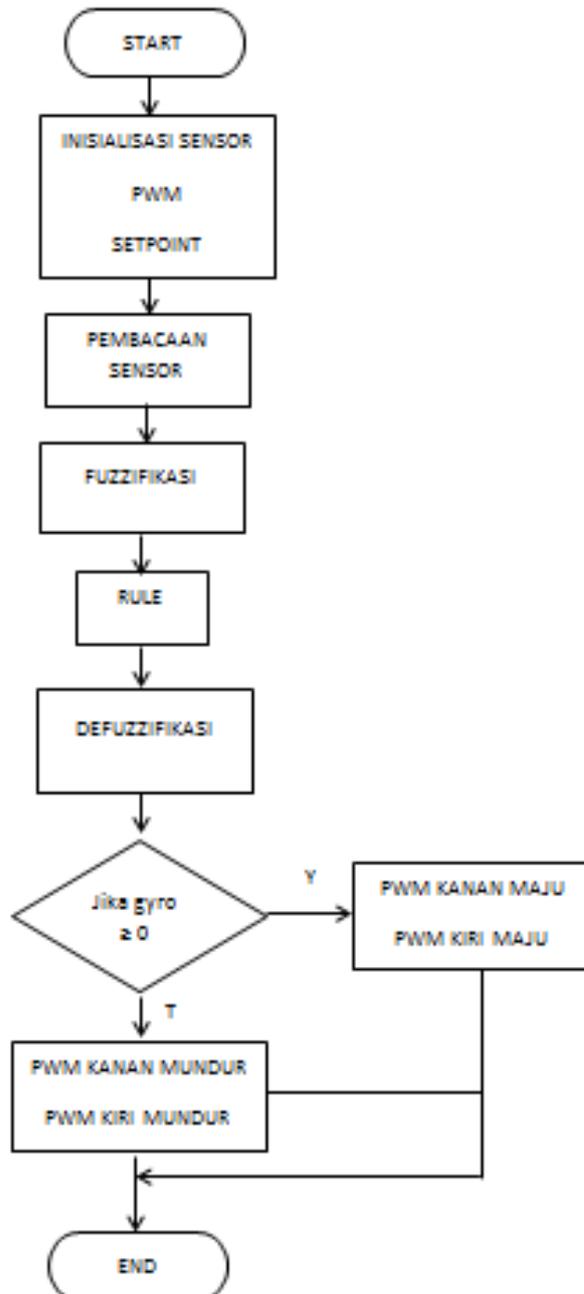
Gambar 4.15 Inferensi Fuzzy Rule

#### 4.5.1.5 Defuzzifikasi

*Defuzzifikasi* adalah metode mengubah *outputfuzzy* menjadi *output* dalam bentuk *crisp*. Hasil *defuzzifikasi* ini yang akan digunakan untuk mengatur besarnya kecepatan pada motor DC. Metode *defuzzifikasi* yang digunakan adalah  $\alpha$ -*predikat* yang diperoleh dari fungsi naik dan fungsi turun, dengan rumusan sebagai berikut :

#### 4.6 Perancangan Algoritma Perangkat Lunak

Perancangan perangkat lunak sistem akan dijelaskan dengan diagram alir sebagai berikut:



**Gambar 4.16** Diagram alir Pengendali kesetimbangan *Kontrol Logika Fuzzy*

## BAB V

### PENGUJIAN DAN ANALISIS

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian pada sistem ini dibagi menjadi tiga bagian, yaitu:

- Pengujian aktuator
- Pengujian sensor
- Pengujian keseluruhan sistem

#### 5.1 Pengujian Aktuator

##### a. Tujuan

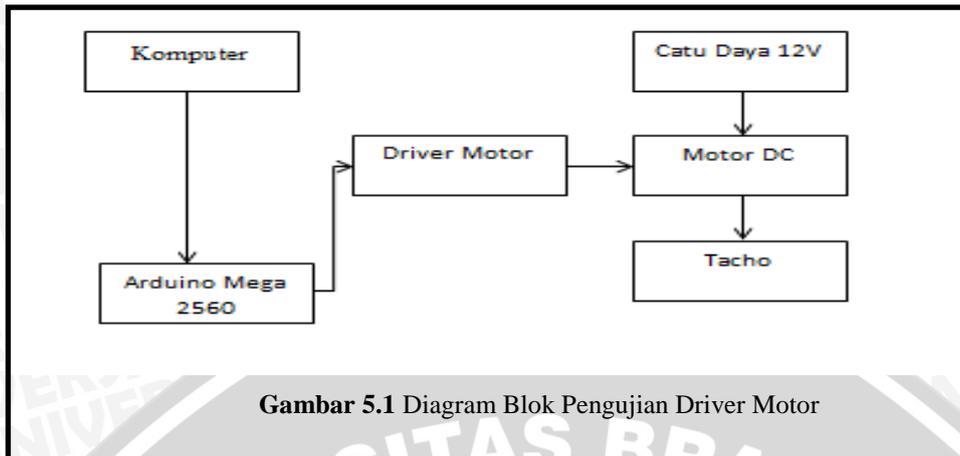
Mengetahui *output* dari aktuator yaitu motor DC apabila diberi *input* yang berbeda-beda.

##### b. Peralatan yang digunakan

- Arduino Mega 2560
- Motor DC
- Komputer
- *Software Arduino 1.0.1.*
- Tachometer Digital Fuji Kyoto Japan
- Catu daya 12 Volt

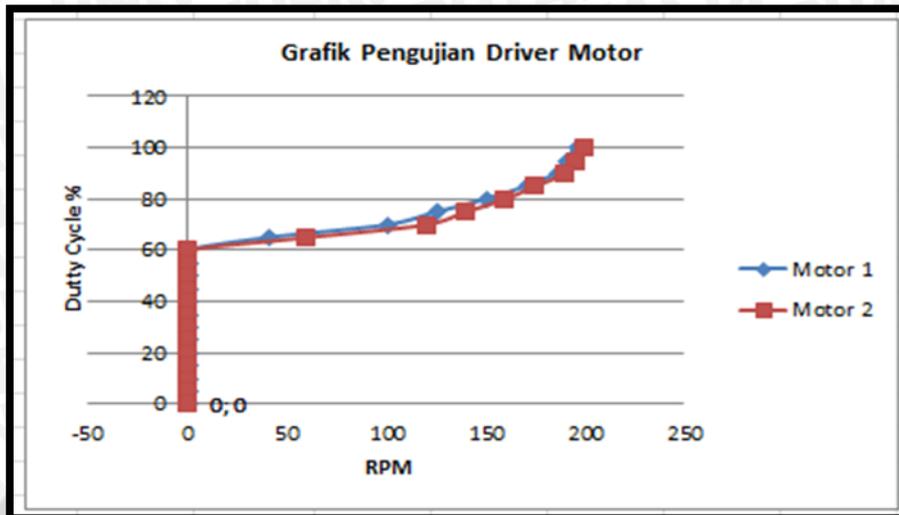
##### c. Langkah Pengujian

1. Merangkai alat seperti pada Gambar
2. Membuat program untuk menentukan arah dan kecepatan motor pada *software Arduino 1.0.1.*
3. Tulis program yang telah dibuat ke dalam mikrokontroler Arduino Mega 2560 menggunakan *software Arduino 1.0.1.*
4. Mengukur tegangan pada *pinPWM* dari mikrokontroler ATmega328 yang digunakan sebagai *pinenable driver* motor
5. Mengukur kecepatan pada motor menggunakan tachometer.
6. Menggambar data yang didapatkan pada grafik.



Dan data hasil pengujian driver motor dapat ditunjukkan dalam tabel 5.1 dan gambar 5.2

Duty Cycle %	PWM	RPM	
		Mtr 1	Mtr 2
100	255	195	200
95	242,25	190	195
90	229,25	185	190
85	216,75	170	175
80	204	150	160
75	191,25	125	140
70	178,5	100	120
65	165,75	40	60
60	153	0	0
55	140,25	0	0
50	127,5	0	0
45	114,75	0	0
40	102	0	0
35	89,25	0	0
30	76,5	0	0
25	63,5	0	0
20	51	0	0
15	38,25	0	0
10	25,5	0	0
5	12,75	0	0
0	0	0	0

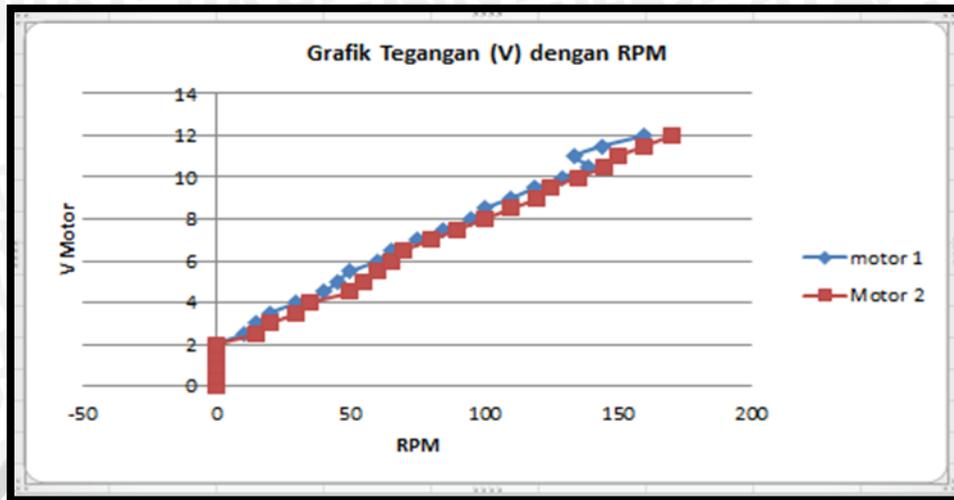


Gambar 5.2 Grafik Hubungan Antara Duty Cycle pada RPM

Tabel 5.2 Pengujian Tenggangan (V) dengan Putaran Motor

Vmotor	RPM	
	Mtr 1	Mtr 2
0	0	0
0,5	0	0
1	0	0
1,5	0	0
2	0	0
2,5	10	15
3	15	20
3,5	20	30
4	30	35
4,5	40	50
5	45	55
5,5	50	60
6	60	65
6,5	65	70
7	75	80
7,5	85	90
8	95	100
8,5	100	110
9	110	120
9,5	119	125
10	129	135
10,5	139	145

11	134	150
11,5	144	160
12	160	170



**Gambar 5.3** Grafik pengujian tegangan dengan Putaran Motor

Dari hasil pengujian aktuator yaitu motor DC berdasarkan Tabel 5.1 dan Gambar 5.2 didapatkan karakteristik dari motor DC semakin besar *duty cycle* (%) yang digunakan maka akan semakin besar pula kecepatan putaran yang akan dihasilkan oleh motor DC. Namun, besarnya kecepatan putaran tiap motor berbeda meskipun dengan jenis dan tipe motor yang sama. Hal ini dikarenakan untuk motor 2 terjadi kelonggaran dalam *gearbox* motor tersebut

## 5.2 Pengujian Sensor

Dalam melakukan pengujian sensor, pengujian dengan hasil yang diinginkan yaitu pengujian pengujian *gyroscope*. *gyroscope* dengan tipe IDG 500.

### 5.2.1 Pengujian Gyroscope

#### a. Tujuan

Mengetahui tingkat keakuratan dari *Gyroscope* dalam membaca perubahan kecepatan sudut yang dinamis dari robot beroda dua.

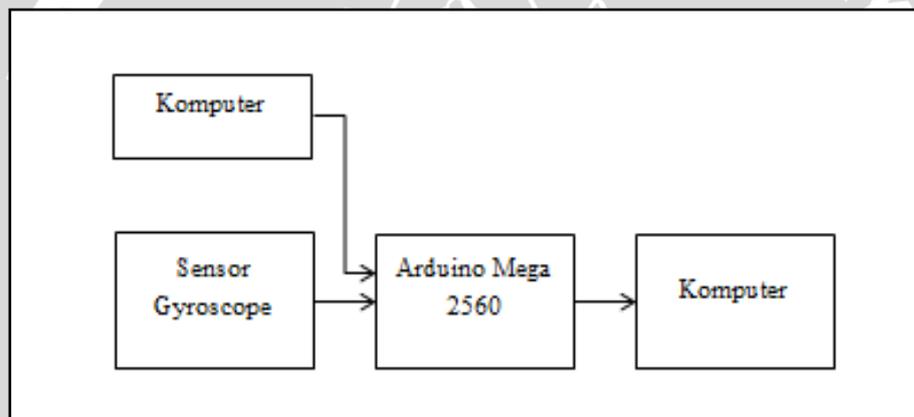
#### b. Peralatan yang digunakan

- Sistem mekanik robot beroda dua yang terhubung dengan *gyroscope* + *accelerometer* dan mikrokontroler ATmega328
- Catu daya 5 volt.
- Catu daya 3.3 volt.

- Komputer.

### c. Langkah Pengujian

1. Memasang alat seperti pada Gambar 5.4.
2. Mengubah posisi kemiringan dari robot beroda dua dari posisi tegak lurus (00) lalu didiamkan sehingga terjatuh dengan sendirinya. Untuk jatuh kearah kanan atau searah dengan jarum jam dibutuhkan sedikit dorongan dari arah kiri robot. Begitu pula sebaliknya.
3. Melihat besaran sudut melalui komputer.
4. Menggambar data keluaran *gyroscope* pada grafik untuk melihat kelinieran dari data modul sensor tersebut.



Gambar 5.4 Diagram Blok Pengujian Sensor

Tabel 5.3 Hasil Pembacaan sensor Gyro

Kemiringan		Nilai Pembacaan	
Ke Depan	Ke Belakang	Ke Depan (-)	Ke Belakang
10	10	10,1	10,2
20	20	19,9	20
30	30	30	30,1
40	40	41	40,2

Dalam tabel 5.3 didapatkan 2 respon utama dari pengujian sensor *gyroscope*. Yaitu dalam keadaan kedepan, berotasi searah jarum jam dan berotasi berlawanan dari jarum jam dengan



### b. Peralatan yang digunakan

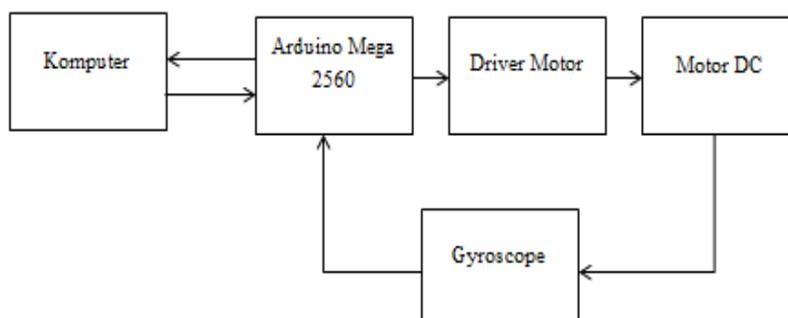
- Plant robot beroda dua lengkap dengan *gyroscope* motor DC
- Arduino Mega 2560
- *Driver* motor EMS 5A H-Bridge.
- Kabel serial
- Program *Microsoft Excel*
- *Software Arduino 1.0.1*

### c. Langkah Pengujian

1. Memasang alat yang dibutuhkan seperti pada Gambar 5.5.
2. Mengaktifkan semua catu daya.
3. Membuat program untuk pengendalian robot beroda dua dengan kontroler PID pada *software Arduino 1.0.1*.
4. Melakukan *make a project* pada *software Arduino 1.0.1*.
5. Melakukan *verify* pada *project* yang telah dibuat,
6. Mengamati kinerja dari robot beroda dua dalam menjaga keseimbangan.



Gambar 5.6 proses pengambilan data



**Gambar 5.7** Blok diagram Pengujian Sistem Secara Keseluruhan

Gambar 5.6 menunjukkan diagram blok pengujian keseluruhan sistem keseimbangan robot beroda dua. Mula-mula robot beroda dua diberdirikan posisi tegak lurus terhadap permukaan bidang datar ( $0^0$ ) kemudian dilepaskan untuk melihat kemampuan sistem penyeimbang dalam menjaga keseimbangan robot beroda dua. Parameter yang diamati pada skripsi ini adalah lama waktu sistem dapat bertahan pada keadaan setimbang sebelum *collapse* atau terjatuh. tabel 5.4 menunjukkan pengamatan data yang terjadi dalam pengujian keseluruhan sistem

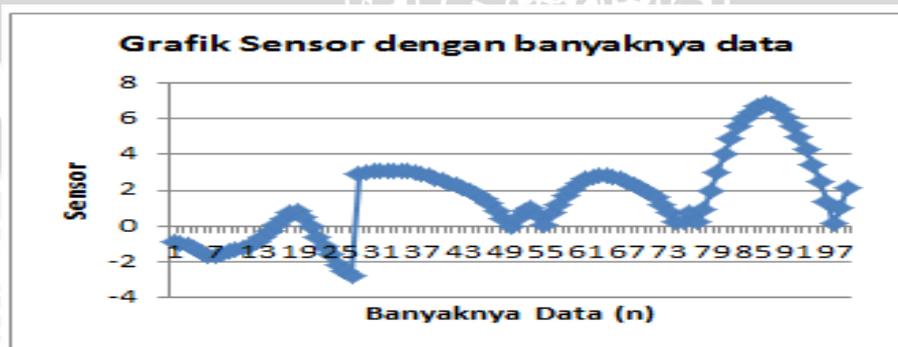
**Tabel 5.4** Data Keseluruhan

Data	Erorr	Derror	pwm
-0.8427664756	0.8427664756	0.0149695873	18.1017570495
-0.9225677490	0.9225677490	0.0798012638	19.6720123291
-1.069867730	1.0698677301	0.1473000049	59.4686584472
-1.2526580095	1.2526580095	0.1827902793	55.4935531616
-1.4738007545	1.4738007545	0.2211427688	54.9339637756
-1.6784448623	1.6784448623	0.2046440839	54.7571716308
-1.6119129657	1.6119129657	-0.0665318965	54.8897476196
-1.5423452854	1.5423452854	-0.0695676803	35.4395751953
-1.4245561599	1.4245561599	-0.1177891492	33.8727989196
-1.3161753416	1.3161753416	-0.1083807945	66.4603881835
-1.2418388128	1.2418388128	-0.0743365287	66.2730255126
-1.1261473894	1.1261473894	-0.1156914234	67.1762313842
-0.8937982559	0.8937982559	-0.2323491096	62.2486534118
-0.5795273303	0.5795273303	-0.3142709732	50.8993568420
-0.2458393335	0.2458393335	-0.3336879968	36.5856475830
0.1016589283	-0.1016589283	-0.3474982738	29.6620254516
0.4477136135	-0.4477136135	-0.3460546731	54.6928482055
0.7647689819	-0.7647689819	-0.3170553445	69.3523178100
0.8696477890	-0.8696477890	-0.1048788428	69.8415603637
0.5029178619	-0.5029178619	0.3667299270	61.9643669128
-0.0230762600	0.0230762600	0.5259941101	17.1574687957
-0.6110679626	0.6110679626	0.5879917144	NA
-1.1099303960	1.1099303960	0.4988624572	NA
-1.6047995090	1.6047995090	0.4948691368	30.1421928405
-2.1345157623	2.1345157623	0.5297162532	30.6413593292
-2.5377855300	2.5377855300	0.4032697677	NA
-2.7546489238	2.7546489238	0.2168633937	42.0912818908
2.9262380599	2.9262380599	0.1715891361	60.3675231933
3.0365314483	3.0365314483	0.1102933883	52.2109565734
3.0787377357	3.0787377357	0.0422062873	52.6828727722
3.1242411136	3.1242411136	0.0455033779	53.1838531494

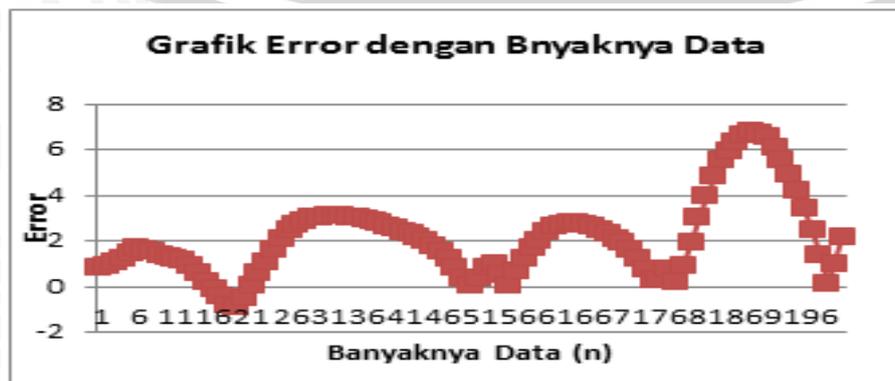
3.1084487438	3.1084487438	0.0157923698	53.0108947753
3.1006808280	3.1006808280	0.0077679157	53.2049293518
3.0777037143	3.0777037143	0.0229771137	52.7989540100
3.0673649311	3.0673649311	0.0103387832	53.0146598815
3.0114312171	3.0114312171	0.0559337139	52.1177864074
2.9062919616	2.9062919616	0.1051392555	52.3770561218
2.8051130294	2.8051130294	0.1011788845	53.6988677978
2.6824364662	2.6824364662	0.1226766109	52.2147674560
2.5674829483	2.5674829483	0.1149535179	52.3001670837
2.4349627494	2.4349627494	0.1325201988	50.3765678405
2.3062427043	2.3062427043	0.1287200450	50.1265525817
2.1601114273	2.1601114273	0.1461312770	48.1879386901
1.9929624557	1.9929624557	0.1671489477	69.8733291625
1.8280794143	1.8280794143	0.1648830175	68.5199584960
1.5768090248	1.5768090248	0.2512704133	66.8689575195
1.2401555776	1.2401555776	0.3366534709	62.5125541687
0.8393712043	0.8393712043	0.4007843971	58.1563110351
0.4038486480	0.4038486480	0.4355225563	48.9372253417
0.0229191398	0.0229191398	0.4267677783,	14.1006088256
0.4103312015	0.4103312015	0.3874120521	68.6692810058
0.8245066642	0.8245066642	0.4141754627	79.1238784790
1.0436226129	1.0436226129	0.2191159725	93.9721908569
0.5693277835	0.5693277835	0.4742948532	57.7956047058
0.0674233531	0.0674233531	0.6367511272	20.1786422729
0.7223554611	0.7223554611	0.6549321174	NA
1.2260717153	1.2260717153	0.5037162303	NA
1.6818772315	1.6818772315	0.4558055400	NA
2.0441055297	2.0441055297	0.3622282743	35.5243072509
2.3950490951	2.3950490951	0.3509435653	47.2214660644
2.6271791458	2.6271791458	0.2321300506	48.6320533752
2.7598364353	2.7598364353	0.1326572895	59.7327232360
2.8307830810	2.8307830810	0.0709466934	49.8108215332
2.7955098152	2.7955098152	0.0352733135	49.3827362060
2.7316641807	2.7316641807	0.0638456344	49.6492462158
2.6380057334	2.6380057334	0.0964685431	49.7771415710
2.4988207817	2.4988207817	0.1391849517	49.7840309143
2.3313241004	2.3313241004	0.1674966812	50.8737907409
2.1364364624	2.1364364624	0.1948876380	70.0735549926
1.9146765708	1.9146765708	0.2217599153	68.6288757324
1.5975607872	1.5975607872	0.3171157836	66.2431106567
1.2238426208	1.2238426208	0.3737181425	64.3561401367
0.7950736045	0.7950736045	0.4287690162	60.6855049133
0.2806361007	0.2806361007	0.5144375801	46.2213401794
0.2614865064	0.2614865064	0.5421226024	187.8754272460
0.7699822425	0.7699822425	0.5084957122	183.7438964843

0.2522596359	0.2522596359	0.5177226066	187.9503936767
0.8951238632	0.8951238632	1.1473834514	NA
1.9317152023	1.9317152023	1.0365914106	NA
3.0159637928	3.0159637928	1.0842485427	NA
4.0153326988	4.0153326988	0.9993688583	NA
4.8968477249	4.8968477249	0.8815150260	NA
5.5407834053	5.5407834053	0.6439356803	NA
5.9862108230	5'.9862108230	0.4454274177	NA
6.3342909812	6.3342909812	0.3480801582	36.8215713500
6.6569628715	6.6569628715	0.3226718902	69.1361312866
6.8537678718	6.8537678718	0.1968050003	73.9456558227
6.7772579193	6.7772579193	0.0765099525	89.6344146728
6.5592193603	6.5592193603	0.2180385589	67.9564437866
6.0894613265	6.0894613265	0.4697580337	83.9011611938
5.5941495895	5.5941495895	0.4953117370	72.7978057861
4.9666786193	4.9666786193	0.6274709701	83.0606765747
4.2662348747	4.2662348747	0.7004437446	90.3368377685
3.4252474308	34.252474308	0.8409873962	97.1698608398
2.4891388416	2.4891388416	0.9361085891	104.7757492065
1.3904656219	1.3904656219	1.0986732292	68.3058853149
0.1228900074	0.1228900074	1.2675756216	NA
1.0323839187	1.0323839187	1.1552739143	NA
2.1767680644	2.1767680644	1.1443841457	NA

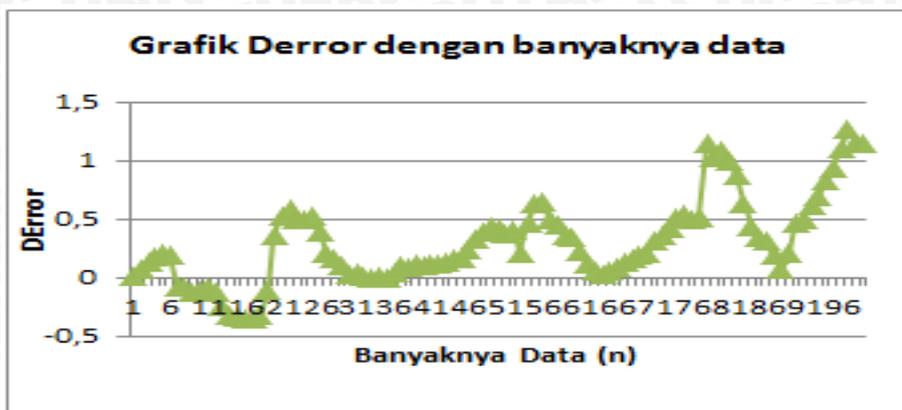
Grafik 5.7 sampai 5.11 merupakan hasil dari data keseluruhan dalam penelitian yang dilakukan dengan menggunakan 100 data dari hasil pembacaan sensor, error, Derror, PWM



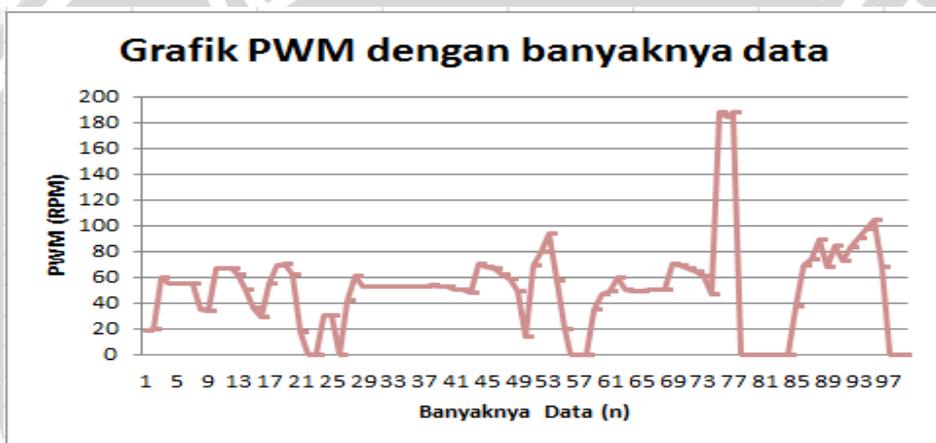
Gambar 5.7 Grafik Sensor dengan banyaknya data



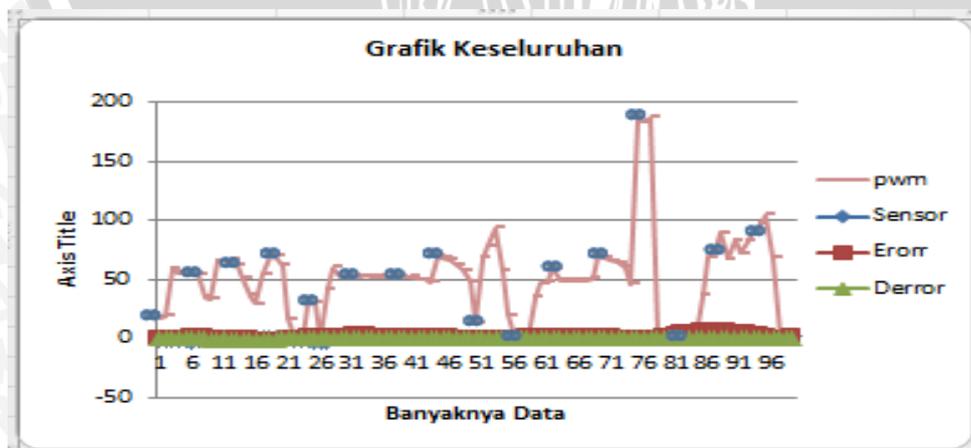
Gambar 5.8 Grafik error dengan Banyaknya data



Gambar 5.9 Grafik Derror dengan banyaknya data



Gambar 5.10 Grafik dengan banyaknya data



Gambar 5.11 Grafik Keseluruhan Data

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut :

1. Sensor *Gyroscope* digunakan untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat
2. Perlu perlakuan khusus terhadap penempatan modul sensor yaitu berada tepat pada sumbu as roda dan berada pada titik tengah robot beroda dua. Karena apabila penempatannya tidak sesuai maka pengambilan data akan mempunyai error yang lebih besar
3. Sistem robot beroda dua dapat stabil yaitu mampu mempertahankan posisi berdiri dan tanpa terjatuh dalam range sekitar  $-20^0$  sampai  $20^0$  dari posisi tegak yaitu  $0^0$
4. Dari beberapa hasil pengujian logika fuzzy untuk pembacaan alat terdapat 4 kondisi yang didapatkan. Yaitu, Stop, Sedang, Cepat, Sangat Cepat
5. Pada robot beroda dua ini untuk mempertahankan keseimbangan dibutuhkan waktu sekitar 5 detik untuk posisi seimbang. Dikarenakan peletakan mekanik seperti batrai dan perkabelan dari alat yang digunakan dapat mempengaruhi hasil dari kerja robot.

#### 6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Penyempurnaan konstruksi mekanik terutama pada sistem mekanik dari *gearbox* motor DC agar lebih simetris antara satu dengan yang lainnya.
2. Menambahkan data pengujian yang digunakan untuk mendapatkan rata-rata waktu pengontrolan pada pengujian keseluruhan.
3. Dilakukan metode pengontrolan selain menggunakan kontrol logika fuzzy
4. Dilakukan metode pengambilan data menggunakan *wireless*. Karena apabila menggunakan kabel maka respon data dari robot beroda dua terjadi lebih besar error yang disebabkan oleh beratnya kabel. Hal ini membebani pada salah satu sisi sehingga pada saat pengambilan data digunakan perangkat tambahan yaitu pembeban yang ditempatkan pada sisi berlawananya.

5. Dibutuhkan pemograman lebih teliti mengenai penentuan set point, karena dalam menentukan set point dapat berpengaruh pada waktu kestabilan robot beroda dua



## Daftar Pustaka

- Braunl, Thomas. 2006. *Embedded Robotics Third Edition*. Jerman: Springer.
- Bretscher, Otto. 2005. *Linear Algebra with Applications*. New Jersey: Pearson Education.
- Rizan, 2008. *Pengendalian PID*
- Kozłowski, Krzysztof. 2006. *Robot Motion and Control*. London: Springer-Verlag.
- Laksana, Andra. 2012. *Balancing Robot Beroda Dua Menggunakan Metode Kendali Proporsional Integral*. Semarang: Skripsi Jurusan Teknik Elektro FT-UNDIP.
- Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 1*, Diterjemahkan Oleh Ir. Edi Leksono, Erlangga, Jakarta, 1994.
- Pitowarno, Endra. 2006. *Robotika: Desain, Kontrol dan Kecerdasan Buatan*. Yogyakarta: Penerbit ANDI.
- Setiawan, Iwan, *Kontrol PID untuk Proses Industri*, Elex Media Komputindo, Jakarta, 2008.
- Spong, Mark W. 2005. *Robot Modelling and Control First Edition*. New York: John Wiley and Sons, Inc.
- Zoel Fachri, Akhmad Hendriawan, Ardik Wijayanto. *Perencanaan Balancing Robot Dengan Dua Roda*. Politeknik Elektronika Surabaya ; 2011.
- Sulistiono, A. 2010. *Pengenalan : Pulse Width Modulation*.  
<http://www.aristy.cz.cc/2010/02/pulse-width-modulation-pwm-pengenalan.html>.

## LAMPIRAN

---

---

### LISTING PROGRAM



// By Shanty Puspitasari - Elektro UB –IMPLEMENTASI KONTROL LOGIKA FUZZY PADA SISTEM KESETIMBANGAN ROBOT BERODA DUA

```
#include <FreeSixIMU.h>
```

```
#include <Wire.h>
```

```
#include <EMS_5A_HBridge.h>
```

```
#define kaMIN1 22 //Pin Header 1 Arah putar Motor
```

```
#define kaMIN2 23 //Pin Header 2 Arah putar Motor
```

```
#define kaMEN 8 //Pin Header 4 Kecepatan Motor
```

```
#define kaMCS A0
```

```
#define kaMSLP 27 //Pen Header 6 MODE Modul "1" Full "0" Sleep
```

```
#define kaEncA 5
```

```
#define kaEncB 12
```

```
//7,9 VCC
```

```
//8,10 GND
```

```
#define kiMIN1 25 //Pin Header 1 Arah putar Motor
```

```
#define kiMIN2 24 //Pin Header 2 Arah putar Motor
```

```
#define kiMEN 9 //Pin Header 4 Kecepatan Motor
```

```
#define kiMCS A1
```

```
#define kiMSLP 26 //Pen Header 6 MODE Modul "1" Full "0" Sleep
```

```
#define kiEncA 2
```

```
#define kiEncB 3
```

```
EMS_5A MtrKa, MtrKi;
```

```
FreeSixIMU ADLX = FreeSixIMU();
```

```
float ArusKa, ArusKi, Sudut_Gy[3], Sudut_Acc[3];
```

```
//char str[30];
```

```
double uE[5];
```

```
double uDE[3];
```

```
double uZ[11];
```

```
double alp[11];
```

```
double err, derr, z, pwm, ern, setp;
```

```
int i;
```

```
const byte naik = 1;
```

```
const byte turun = 2;
```

```
const byte nul = 0;
```

```
const byte minn = 1;
```

```
void setup(){
```

```
  Serial.begin(115200);
```

```
  Wire.begin();
```

```
  ADLX.init();
```

```
  MtrKi.init(kiMIN1,kiMIN2,kiMEN,kiMSLP,kiMCS);
```

```
  MtrKa.init(kaMIN1,kaMIN2,kaMEN,kaMSLP,kaMCS);
```

```

pinMode(kaEncA,INPUT);
pinMode(kaEncB,INPUT);
derr=0;
ernn=0;
setp=0;
}

void loop(){
  ADLX.getEuler(Sudut_Gy);
  err= setp - Sudut_Gy[1];
  fuzzification(err, derr);
  Rule();
  // for (int t=1; t<=11;t++)
  //   {Serial.print("Test-");
  //     Serial.print(t,DEC);
  //     Serial.print(":");
  //     Serial.println(uZ[t]);}

  Defuzz();
  derr=err-ernn;
  ernn=err;

  if (Sudut_Gy[1]>0)
  {
    MtrKa.pwm_turn(MAJU,int(pwm)+5);
    MtrKi.pwm_turn(MAJU,int(pwm));
  }
  else
  {
    // MtrKa.pwm_turn(MUNDUR,int(pwm)+5);
    // MtrKi.pwm_turn(MUNDUR,int(pwm));

    MtrKa.pwm_turn(MUNDUR,int(pwm)+5);
    MtrKi.pwm_turn(MUNDUR,int(pwm));
  }

  //MtrKa.pwm_turn(MUNDUR,50);
  //MtrKi.pwm_turn(MUNDUR,40);

  // Serial.println(Sudut_Gy[1]);

  // delay(500);
}

void cekp(float cp)

```

```

{
  for (int g=1; g<=3; g++)
  {
    delay(1000);
    Serial.println(cp);
  }
}
void Defuzz()
{
  float alptot,ztot;
  float ztemp[11];
  alptot=0;
  for (int j=0; j<=10; j++)
  {
    alptot=alptot+alp[j];
    ztemp[j]=alp[j]*uZ[j];
  }
  ztot=0;
  for (int j=0; j<=10; j++) ztot=ztot+ztemp[j];
  pwm=ztot/alptot;
}
void Rule()
{
  for (int h=0; h<=10; h++)
  {
    uZ[h]=0.0;
    alp[h]=0.0;
  }
  i=1;

  if (uE[3] != 0 && uDE[2] != 0)
  {
    alp[i]=min(uE[3], uDE[2]);
    uZ[i] = nmin(uE[3], uDE[2], 0, 60);
    i++;
  }

  if (uE[4] != 0 && uDE[2] != 0)
  {
    alp[i]=min(uE[4], uDE[2]);
    uZ[i] = nmin(uE[4], uDE[2], 60, 125);
    i++;
  }

  if (uE[2] != 0 && uDE[2] != 0)
  {
    alp[i]=min(uE[2], uDE[2]);
    uZ[i] = nmin(uE[2], uDE[2], 60, 125);
    i++;
  }
}

```



```

if (uE[4] != 0 && uDE[3] != 0)
{
  alp[i]=min(uE[4], uDE[3]);
  uZ[i] = nmin(uE[4], uDE[3], 125, 190);
  i++;
}

if (uE[4] != 0 && uDE[1] != 0)
{
  alp[i]=min(uE[4], uDE[1]);
  uZ[i] = nmin(uE[4], uDE[1], 60, 125);
  i++;
}

if (uE[2] != 0 && uDE[1] != 0)
{
  alp[i]=min(uE[2], uDE[1]);
  uZ[i] = nmin(uE[2], uDE[1], 125, 190);
  i++;
}

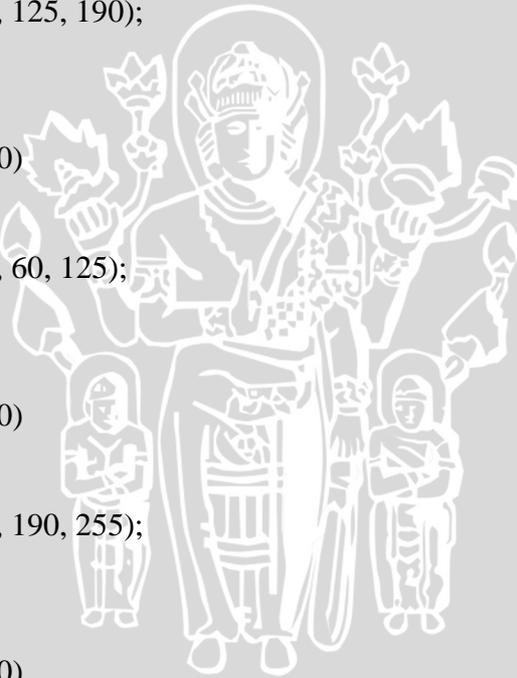
if (uE[2] != 0 && uDE[3] != 0)
{
  alp[i]=min(uE[2], uDE[3]);
  uZ[i] = nmin(uE[2], uDE[3], 60, 125);
  i++;
}

if (uE[5] != 0 && uDE[3] != 0)
{
  alp[i]=min(uE[5], uDE[3]);
  uZ[i] = nmin(uE[5], uDE[3], 190, 255);
  i++;
}

if (uE[5] != 0 && uDE[1] != 0)
{
  alp[i]=min(uE[5], uDE[1]);
  uZ[i] = nmin(uE[5], uDE[1], 125, 190);
  i++;
}

if (uE[1] != 0 && uDE[1] != 0)
{
  alp[i]=min(uE[1], uDE[1]);
  uZ[i] = nmin(uE[1], uDE[1], 190, 255);
  i++;
}

```



```

if (uE[1] != 0 && uDE[3] != 0)
{
    alp[i]=min(uE[1], uDE[3]);
    uZ[i] = nmin(uE[1], uDE[3], 125, 190);
    i++;
}
}

```

```

float nmin(float valE, float valDE, float a, float b)
{
    float retmin;
    if (valE < valDE)
        retmin = b - (valE * (b - a));
    else
        retmin = (valDE * (b - a)) + a;
    return retmin;
}

```

```

float fungsi(byte kondisi, float a, float b, float x)
{
    float retfungsi;
    if (kondisi == naik)
    {
        retfungsi=(x - a)/(b - a);
    }
    if (kondisi == turun)
    {
        retfungsi=(b - x)/(b - a);
    }
    return retfungsi;
}

```

```

void fuzzyfication(float eval, float deval)
{
    for (int f=1; f<=5; f++)
    {
        uE[f]=0.0;
        if (f <= 3)
        {
            uDE[f]=0.0;
        }
    }
}

```

```

if (eval <= -16 && eval >= -24)
    uE[1]=fungsi(naik,-24,-16,eval);
else
    if (eval <= -8 && eval >= -16)
    {
        uE[1]=fungsi(turun,-8,-16,eval);
        uE[2]=fungsi(naik,-8,-16,eval);
    }
}

```



```

}
else
if (eval <= 0 && eval >= -8)
{
    uE[2]=fungsi(turun,-8,0,eval);
    uE[3]=fungsi(naik,-8,0,eval);
}
else
if (eval <= 8 && eval >= 0)
{
    uE[3]=fungsi(turun,0,8,eval);
    uE[4]=fungsi(naik,0,8,eval);
}
else
if (eval <= 16 && eval >= 8)
{
    uE[4]=fungsi(turun,8,16,eval);
    uE[5]=fungsi(naik,8,16,eval);
}
else
if (eval <= 24 && eval >= 16)
    uE[5]=fungsi(turun,16,24,eval);

if (deval <= -0.5 && deval >= -1)
    uDE[1]=fungsi(naik,-1,-0.5,deval);
else
if (deval <= 0 && deval >= -0.5)
{
    uDE[1]=fungsi(turun,-0.5,0,deval);
    uDE[2]=fungsi(naik,-0.5,0,deval);
}
else
if (deval <= 0.5 && deval >= 0)
{
    uDE[2]=fungsi(turun,0,0.5,deval);
    uDE[3]=fungsi(naik,0,0.5,deval);
}
else
if (deval <= 1 && deval >= 0.5)
    uDE[3]=fungsi(turun,0.5,1,deval);
}

```

