



**RANCANG BANGUN SISTEM PENGENDALIAN KECEPATAN MOTOR
PENDORONG ROBOT HOVERCRAFT DENGAN KEMUDI OTOMATIS
BERBASIS ARDUINO**

SKRIPSI

Oleh

Arief Fahmy Ubaidillah

NIM 131910201047

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2018**



**RANCANG BANGUN SISTEM PENGENDALIAN KECEPATAN MOTOR
PENDORONG ROBOT HOVERCRAFT DENGAN KEMUDI OTOMATIS
BERBASIS ARDUINO**

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

Oleh

Arief Fahmy Ubaidillah

NIM 131910201047

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2018**

PERSEMBAHAN

Dengan ini saya persembahkan skripsi kepada:

1. Allah SWT yang Maha Pengasih lagi Maha Penyayang.
2. Kedua orang tua tercinta, Bapak Ibrahim Sholih dan Ibu Siti Muallifah serta adikku Sakirna Syifa Risdiana atas kasih sayang, pengorbanan, dan kesabaran yang tiada tara serta doa yang selalu menyertai.
3. Guru – guru mulai TK Wahid Hasyim Jatimulyo, Misri Riadlotul Uqul jatimulyo, MTS Wahid Hasyim Jatimulyo, SMK Negeri 2 Jember dan dosen-dosen Teknik Elektro Universitas Jember. Terima kasih untuk ilmu dan pengalaman yang telah diajarkan selama ini.
4. Penghuni Kontraan KALIURANG dan teman-teman kost-Assofwah, yang selalu membantu dan menyemangati saya setiap hari.
5. Teman-teman KKN 113 yang selalu memberi semangat setiap hari.
6. Rekan-rekan satu DPU dan DPA yang selalu menemani saya berjuang mulai awal pengerjaan skripsi hingga selesai.
7. Keluarga Intel 2013 yang selalu membantu, menyemangati dan selalu mendampingi saya selama pengerjaan skripsi ini.
8. Almamater Teknik Elektro Universitas Jember.

MOTTO

“Allah tidak hendak menyulitkan kamu, tetapi Dia hendak membersihkan kamu dan menyempurnakan nikmat-Nya bagimu, supaya kamu bersyukur.”

(QS. Al Maidah: 6)

“Barangsiapa yang mempermudah urusan orang yang mengalami kesulitan, maka Allah akan mempermudah urusannya di dunia dan di akhirat”

(HR. Ibnu Majah).

“Siapa yang beriman kepada Allah dan Hari Akhir, maka janganlah menyakiti tetangganya. Siapa yang beriman kepada Allah dan Hari Akhir, maka muliakanlah tamunya. Dan siapa yang beriman kepada Allah dan Hari Akhir, maka bicaralah yang baik atau diamlah”.

(HR Bukhari, Muslim, dan Ahmad)

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

nama : Arief Fahmy Ubaidillah

NIM : 131910201047

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ” Rancang Bangun Sistem Pengendalian Kecepatan Motor Pendorong Robot Hovercraft Dengan Kemudi Otomatis Berbasis Arduino” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 23 November 2018

Yang menyatakan,

Arief Fahmy Ubaidillah

NIM 131910201047

SKRIPSI

**RANCANG BANGUN SISTEM PENGENDALIAN KECEPATAN MOTOR
PENDORONG ROBOT HOVERCRAFT DENGAN KEMUDI OTOMATIS
BERBASIS ARDUINO**

Oleh

Arief Fahmy Ubaidillah

NIM 131910201047

Pembimbing

Dosen Pembimbing Utama : Sumardi, S.T., M.T.

Dosen Pembimbing Anggota : Ali Rizal Chaidir, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Rancang Bangun Sistem Pengendalian Kecepatan Motor Pendorong Robot Hovercraft Dengan Kemudi Otomatis Berbasis Arduino” karya Arief Fahmy Ubaidillah telah diuji dan disahkan pada :

Hari : Jum’at

Tanggal : 23 November 2018

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Anggota I,

Sumardi, M.T.
NIP 196701131998021001

Ali Rizal Chaidir, S.T., M.T.
NIP 760015754

Anggota II,

Anggota III,

Khairul Anam, S.T., M.T., Ph.D.
NIP 197804052005011002

Dodi Setiabudi, S.T., M.T.
NIP 198405312008121004

Mengesahkan
Dekan,

Dr. Ir. Entin Hidayah, M.UM.
NIP 196612151995032001

RINGKASAN

Hovercraft merupakan sebuah kendaraan yang memanfaatkan bantalan udara sebagai landasan, hovercraft bergerak dengan memanfaatkan gaya angkat dan gaya dorong yang dihasilkan oleh baling-baling yang digerakkan oleh mesin dengan kekuatan yang besar. Pengaplikasian dari kendaraan hovercraft yaitu digunakan sebagai kendaraan penjelajah milik militer.

Penelitian ini bertujuan agar dapat membuat robot Hovercraft yang mampu bergerak secara otomatis dengan pengendali fuzzy logic, serta dapat mengetahui proses pembuatan program dengan pengendali fuzzy logic sebagai pengendali kecepatan motor pendorong robot Hovercraft. Manfaat dari penelitian ini yaitu peneliti mampu membuat robot hovercraft yang mampu bergerak secara otomatis dan membuat program dengan pengendali fuzzy logic.

Robot Hovercraft pada penelitian ini menggunakan mikrokontroler Arduino dan sensor HC-SR04 sebagai input, motor servo sebagai pengendali arah berbelok robot Hovercraft dan 2 motor brushless dc sebagai motor pendorong dan motor hover. Fuzzy logic digunakan sebagai pengendali kecepatan motor pendorong pada robot Hovercraft memiliki 2 fungsi keanggotaan yaitu sensor depan, dan fungsi keanggotaan sensor kanan dan kiri dan memiliki 3 variabel pada masing2 variabel yaitu dekat, sedang, jauh, dari 3 variabel ini nantinya hasil jarak yang terbaca akan di proses sesuai rule base dan menghasilkan output PWM dan menghasilkan kecepatan pada motor brushless dc.

Hasil pengujian pada penelitian ini yaitu mengetahui respon robot terhadap lintasan dengan lintasan yang berbeda-beda, lintasan yang digunakan yaitu lintasan darat dan lintasan air, pada saat pengujian di lintasan darat respon robot terhadap lintasan yaitu dalam 10 kali percobaan keberhasilan robot menelusuri lintasan sebesar 60% dengan rata-rata waktu keberhasilan 38 detik dan didapatkan nilai standart deviasi sebesar 2.250, dan pengujian di lintasan air respon robot terhadap lintasan yaitu dengan pengujian 10 kali percobaan keberhasilan robot menelusuri lintasan sebesar 50% dengan rata-rata waktu keberhasilan 24 detik, dan didapatkan nilai standart deviasi sebesar 3.209. semakin kecil nilai standart

deviasi maka konsistensi keberhasilan waktu tempuh robot semakin bagus hasilnya, dapat dikatakan tingkat keberhasilannya semakin tinggi.



SUMMARY

Hovercraft is a vehicle that utilizes air bearings as a foundation, moving hovercraft utilizing the lift force and thrust produced by propellers driven by engines with great force. The application of the hovercraft vehicle is used as a military explorer vehicle. This study aims to be able to make Hovercraft robots capable of moving automatically with fuzzy logic controllers, and can find out the process of making programs with fuzzy logic controllers as Hovercraft robot driving motor speed controllers. The benefit of this research is that researchers are able to make hovercraft robots that are able to move automatically and create programs with fuzzy logic controllers.

The Hovercraft robot in this study used an Arduino microcontroller and HC-SR04 sensor as input, a servo motor as a Hovercraft robotic direction controller and 2 dc brushless motors as a driving motor and a hover motor. Fuzzy logic is used as a driving motor speed controller on Hovercraft robots having 2 membership functions, namely front sensor, and right and left sensor membership functions and has 3 variables in each variable, namely near, medium, far, from these 3 variables, the results of the readable distance process according to the rule base and produce the PWM output and generate speed on the dc brushless motor.

The test results in this study are to know the response of robots to trajectories with different trajectories, the trajectory used is land trajectory and water trajectory, when testing on the land trajectory the robot responds to the trajectory which is in 10 attempts the success of the robot traces the track by 60% with an average success time of 38 seconds and the standard deviation value of 2.250 was obtained, and testing on the robot's water trajectory on the trajectory, namely by testing 10 experiments the success of the robot traced the track by 50% with an average success time of 24 seconds, and obtained a value standard deviation of 3.209. the smaller the standard deviation value, the more successful the travel time of the robot, the better the results, it can be said that the success rate is getting higher.

PRAKATA

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Pengendalian Kecepatan Motor Pendorong Robot Hovercraft Dengan Kemudi Otomatis Berbasis Arduino”. Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan program studi strata satu (S1) Jurusan Teknik Elektro Fakultas Teknik Universitas Jember. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Ibu Dr.Ir. Entin Hidayah M.U.M selaku dekan Fakultas Teknik Universitas Jember.
2. Bapak Sumardi, S.T., M.T selaku dosen pembimbing utama dan Bapak Ali Rizal Chaidir, S.T., M.T., selaku dosen pembimbing anggota yang telah rela meluangkan waktu, pikiran serta motivasi dalam penyusunan skripsi ini.
3. Bapak Khairul Anam, S.T., M.T., P.hd., selaku dosen penguji utama dan Bapak Dodi Setiabudi, S.T., M.T., selaku dosen penguji anggota yang telah memberikan kritik dan saran yang membangun sehingga sangat membantu terhadap penyempurnaan skripsi ini.
4. Bapak Catur Suko Sarwono, S.T., selaku dosen pembimbing akademik yang telah membimbing dan menanamkan rasa disiplin dan tanggung jawab dengan apa yang dilakukan selama penulis menjadi mahasiswa.
5. Bapak Widya Cahyadi, S.T., M.T., selaku Komisi Bimbingan S1 yang telah membantu penulisan skripsi secara administratif.
6. Kedua orang tua tercinta, Bapak Ibrahim Sholeh dan Ibu Siti Mualifah serta adikku Syakirna Syifa Risdiana atas kasih sayang, pengorbanan, dan kesabaran yang tiada tara serta doa yang selalu menyertai.
7. Kepada teman-teman seperjuangan penghuni kontraan kaliurang, fajar, ekky, yang selalu saling mendukung satu sama lain dalam penyusunan skripsi.

8. Rekan satu perjuangan penelitian, Edi Trikurniawan, Dimas Suryo, Fajar Gunawan, Rizman Pebrian, Faiq, Wahyu, yang menjadi teman dalam mengerjakan alat penelitian dan penyusunan skripsi.
9. Teman-teman kontrakan kaliurang yang selama 4 tahun telah saling membantu satu sama lain.
10. Rekan-rekan Fakultas Teknik Universitas Jember khususnya rekan-rekan Teknik Elektro Angkatan 2013 yang tidak dapat disebutkan satu per satu, selama ini telah memberikan pengalaman hidup selama penulis menjadi keluarga Fakultas Teknik Universitas Jember.
11. Serta seluruh pihak yang telah membantu dalam mengerjakan skripsi ini yang tidak bisa disebutkan namanya satu persatu.

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

Jember, 23 November 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN.....	vii
RINGKASAN	viii
SUMMARY	x
PRAKATA.....	xiii
DAFTAR ISI.....	xv
DAFTAR TABEL	xviii
DAFTAR GAMBAR.....	xix
DAFTAR LAMPIRAN	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Hovercraft	4
2.1.1 Pengertian <i>Hovercraft</i>	4
2.1.2 Komponen Utama Pada <i>Hovercraft</i>	4
A. <i>Hull</i>	4
B. <i>Skirt</i>	4
C. Mekanisme <i>Hovercraft</i>	5

2.2 Arduino MEGA 2560	11
2.2.1 Ringkasan Spesifikasi	12
2.2.2 Sumber Daya	12
2.2.3 Memori	12
2.2.4 <i>Input dan Output</i>	12
2.2.5 Komunikasi	14
2.3 Motor <i>Brushless</i> DC (BLDC)	14
2.3.1 Bagian - Bagian Motor <i>Brushless</i> DC.....	16
2.4 <i>Electronic Speed Controller</i> (ESC).....	20
2.5 Baling – Baling (<i>Propeller</i>).....	21
2.6 Motor Servo	22
2.7 Sensor HC-SR04	23
2.7.1 Pengertian Sensor HC-SR04	23
2.7.2 Prinsip Kerja Sensor Jarak Ultrasonik	25
2.7.3 Bentuk Fisik HC-SR04.....	25
2.7.4 Timing Diagram HC-SR04	26
2.8 Pengendali Logika <i>Fuzzy</i> / <i>Fuzzy Logic Controller</i>	27
2.8.1 Pengertian Fuzzy Logic	27
2.8.2 Konsep Fuzzy Logic	28
2.8.3 Himpunan Fuzzy	28
2.8.4 Fungsi Keanggotaan	30
2.8.5 Sistem Inferensi Fuzzy	33
2.8.6 Penegasan (defuzzifikasi)	33
BAB 3. METODOLOGI PENELITIAN	36
3.1 Tempat dan Waktu Penelitian	36
3.2 Perencanaan Jadwal Penelitian	36
3.3 Perancangan Mekanik	37
3.4 Perancangan Elektronik	39
3.4.1 Diagram Blok Perancangan Elektronik.....	39
3.4.2 Rangkaian Elektronik.....	41

3.5 Perancangan Software	44
3.5.1 Diagram Blok Sistem Pengendali Jarak dan Kecepatan Motor Pendorong Robot <i>Hovercraft</i>	44
3.5.2 Perancangan Kontrol <i>Fuzzy Logic</i>	45
3.5.3 Diagram Alir	49
3.7 Pengujian	51
A. Pengujian Hardware	51
B. Pengujian Kontrol Sistem	52
1. Kontrol Kecepatan	52
2. Kontrol Posisi Robot pada Dinding Penghalang	52
C. Pengujian Keseluruhan	52
BAB 4. HASIL DAN PEMBAHASAN	54
4.1 Pengujian <i>Hardware</i>	55
4.1.1 Pengujian Sensor Jarak <i>Ultrasonic</i> HC-SR04	55
4.1.2 Pengujian Motor DC Brushless <i>Hover</i>	59
4.1.3 Pengujian Sudut Motor Servo	61
4.2 Pengujian Kontrol Sistem	63
4.2.1 Pengujian <i>Control Fuzzy Logic</i>	63
4.2.2 Pengujian RPM Motor Pendorong tanpa <i>Fuzzy Logic</i>	64
4.2.3 Pengujian RPM Motor Pendorong dengan <i>Fuzzy Logic</i>	68
4.3 Pengujian Sistem Keseluruhan	72
4.3.1 Pengujian pada Lintasan Darat.....	72
4.3.2 Pengujian pada Lintasa Air	78
BAB 5. PENUTUP.....	82
5.1 Kesimpulan	82
5.2 Saran.....	83
DAFTAR PUSTAKA	84
LAMPIRAN.....	85

DAFTAR TABEL

	Halaman
2.1 Data Sheet Ultrasonik HC-SR04.....	24
3.1 Tabel Penjadwalan Proses Pengerjaan Alat	36
3.2 <i>Fuzzy Rule</i>	48
4.1 Hasil Pengujian Sensor <i>Ultrasonic</i> HC-SR04.....	56
4.2 Hasil Pengujian RPM dan Gaya Angkat Robot	59
4.3 Hasil Pengujian Sudut Motor Servo.....	61
4.4 Hasil Perbandingan Pengujian <i>Output Fuzzy Logic</i> dengan Perhitungan.....	63
4.5 Hasil Pengujian RPM Motor DC <i>Brushless</i> Pendorong Tanpa <i>Fuzzy Logic</i>	64
4.6 Hasil Pengujian RPM Motor DC <i>Brushless</i> Pendorong dengan <i>Fuzzy Logic</i>	68
4.7 Hasil Pengujian Pada Lintasa Darat	74
4.8 Hasil Pengujian Kecepatan Motor dc <i>Brushless</i> Pendorong pada Lintasan Darat.....	76
4.9 Hasil Pengujian Pada Lintasa Air	78
4.10 Hasil Pengujian Kecepatan Motor DC <i>Brushless</i> Pendorong Pada Lintasan Air	80

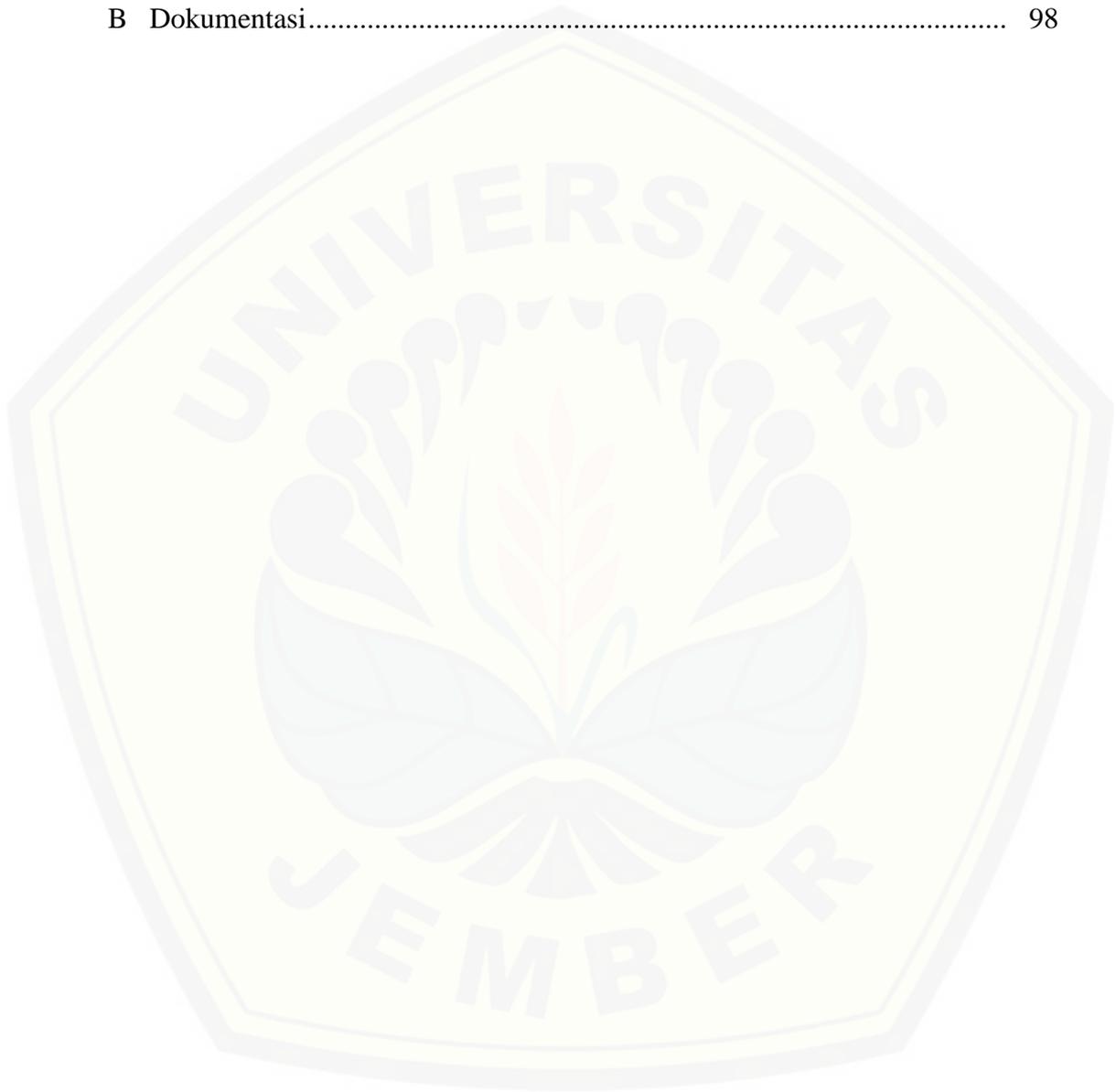
DAFTAR GAMBAR

	Halaman
2.1 Mekanisme <i>Hover</i>	5
2.2 Gaya Angkat Pada <i>Hover</i>	6
2.3 Gaya Dorong Pada <i>Hover</i>	8
2.4 Arduino Mega2560	11
2.5 Motor <i>Brushless</i> DC.....	15
2.6 Motor <i>Brushless</i> DC (a) 1 fasa (b) 2 fasa.....	16
2.7 Motor <i>Brushless</i> DC 3 Fasa	16
2.8 Penggerak Elektronik Daya Untuk Motor <i>Brushless</i> DC	17
2.9 Blok Diagram Sistem Kontroler Dan <i>Driver</i> Pada Motor <i>Brushless</i> Dc ...	17
2.10 Gambar Stator Pada Motor <i>Brushless</i> DC	18
2.11 Komponen Pada Motor <i>Brushless</i> DC	20
2.12 <i>ESC (Electronic Speed Controller)</i>	20
2.13 <i>Propeller</i> Dengan Berbagai Bilah.....	22
2.14 Motor Servo	22
2.15 Cara Kerja Sensor HC-SR04.....	24
2.16 Konfigurasi Pin Dan Tampilan Sensor Ultrasonik Hc-Sr04.....	26
2.17 <i>Timing Diagram</i> Pengoperasian Sensor Ultrasonik Hc-Sr04	26
2.18 Representasi Linear Naik	31
2.19 Representasi Linear Turun	31
2.20 Kurva Segitiga.....	32
2.21 Representasi Kurva Trapezium	33
2.22 Tahapan Sistem Infransi <i>Fuzzy</i>	35
3.1 Desain Robot <i>Hovercraft</i> Tampak Samping	37
3.2 Desain Robot <i>Hovercraft</i> Tampak Atas	37
3.3 Diagram Blok Perancangan Elektronik.....	39
3.4 Rangkaian Kontrol Sensor Jarak.....	41
3.5 Rangkaian Kontrol Motor DC <i>Brushless</i>	43

3.6	Rangkaian Kontrol Motor Servo	43
3.7	Diagram Blok Pengendali Fuzzy	44
3.8	Fungsi Keanggotaan Sensor Depan	45
3.9	Fungsi Keanggotaan Sensor Kanan dan Kiri	46
3.10	<i>Output</i> Jarak dari Pengolahan <i>Fuzzy Logic</i>	48
3.11	Diagram Alir Navigasi Robot <i>Hovercraft</i>	51
3.12	Diagram Alir Motor Pendorong Robot <i>Hovercraft</i> dengan <i>Fuzzy</i>	52
4.1	Robot <i>Hovercraft</i> Dengan Kendali Kemudi Otomatis	54
4.2	Grafik Selisih Massa Robot Ketika <i>Hover</i>	60
4.3	Grafik Rpm Motor Pendorong Pada Pengujian Ke 1	65
4.4	Grafik Rpm Motor Pendorong Pada Pengujian Ke 2.....	65
4.5	Grafik Rpm Motor Pendorong Pada Pengujian Ke 3.....	66
4.6	Grafik Rpm Motor Pendorong Pada Pengujian Ke 4.....	66
4.7	Grafik Rpm Motor Pendorong Pada Pengujian Ke 5.....	66
4.8	Grafik Hasil Pengujian Rata-rata RPM Motor DC <i>Brushless</i> Pendorong ...	67
4.9	Grafik Rpm Motor Pendorong Pada Pengujian Ke 1	69
4.10	Grafik Rpm Motor Pendorong Pada Pengujian Ke 2.....	69
4.11	Grafik Rpm Motor Pendorong Pada Pengujian Ke 3.....	70
4.12	Grafik Rpm Motor Pendorong Pada Pengujian Ke 4.....	70
4.13	Grafik Rpm Motor Pendorong Pada Pengujian Ke 5.....	70
4.14	Grafik Hasil Pengujian Rata-rata RPM Motor DC <i>Brushless</i> Pendorong ...	71
4.15	Posisi Robot Pada Dinding Lintasan.....	73
4.16	Grafik Pengujian Kecepatan Motor DC <i>Brushless</i> Pendorong pada Lintasan Darat.....	77
4.17	Grafik Pengujian Kecepatan Motor DC <i>Brushless</i> Pendorong pada Lintasan Air	81

DAFTAR LAMPIRAN

	Halaman
A Program	85
B Dokumentasi.....	98



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Dunia robot yang semakin maju perkembangannya terutama dikalangan mahasiswa teknik elektronika dan sistem kendali. Pada jaman yang serba robot ini, robot banyak digunakan untuk membantu pekerjaan yang berat, berbahaya, penjelajah luar angkasa, dan pencari tambang, tidak hanya itu robot juga dapat di fungsikan sebagai penyelamatan dari bencana, bidang hiburan, robot cerdas dan sebagai alat bantu rumah tangga, pengembangan robot cerdas di dunia ini sangat cepat perkembangannya dari robot pemilah barang, kemudi otomatis, pencari titik api, pencari benda, hingga robot penjelajah labirin. Perkembangan robot dengan kendali otomatis banyak diciptakan mulai dari kendaraan darat, kendaraan udara, dan kendaraan di air. Buruknya cuaca yang tidak menentu akan mempengaruhi kinerja dari robot tersebut. Hal ini mengakibatkan para peneliti robot dituntut untuk mengembangkan konstruksi robot yang mampu memperbaiki kinerja robot dengan kendali otomatis.

Hovercraft merupakan kendaraan yang memanfaatkan bantalan udara sebagai landasan. Dengan penggerak yang dapat menghasilkan gaya angkat dan gaya dorong dengan kecepatan tertentu sehingga *hovercraft* dapat bergerak. *Hovercraft* disebut sebagai kendaraan amfibi yang yang mampu bergerak di darat maupun bergerak di air dan dapat bergerak dibidang yang berlumpur. Keunggulan lain konstruksi ini adalah memiliki gesekan yang kecil, dan tidak membutuhkan landasan luncur dan gaya yang rendah. Salah satu aplikasi dari kendaraan ini adalah sebagai kendaraan penjelajah milik militer.

Dalam pengembangan konstruksi *Hovercraft* sebagai robot kendali otomatis mengikuti lintasan yang akan *dibuat* adalah robot *Hovercraft* yang ditanamkan sensor jarak. Posisi sensor mewakili *setpoint* untuk mengetahui jarak robot pada dinding tepi, pemanfaatan sensor jarak untuk menghindari terjadinya benturan robot dengan dinding dan juga digunakan sebagai penentuan arah belok robot dengan pemanfaatan sensor jarak diharapkan robot dapat berbelok dengan lancar

tanpa menabrak dinding pembatas dan mengurangi kecepatan motor pendorong saat berbelok ke kanan atau ke kiri sehingga *didapat error* yang kecil.

Disini saya ingin membuat robot *Hovercraft* dengan kendali otomatis dengan menggunakan metode *fuzzy logic*, pada metode ini berfungsi sebagai pengontrol sirip pada *hover board* dan kecepatan motor dengan memanfaatkan sensor jarak sebagai *setpoint* dan di kontrol dengan menggunakan mikrokontroler Arduino sebagai pengendali *input* dari sensor jarak dan *output* yaitu motor *Brushless* dan servo sebagai kemudi. Pada tugas akhir ini saya membuat penelitian tentang “ **Rancang Bangun Sistem Pengendalian Kecepatan Motor Pendorong Robot *Hovercraft* Dengan Kemudi Otomatis Berbasis Arduino** “ sehingga penulis akan mengetahui bagaimana pengaruh metode *fuzzy logic* yang akan *diterapkan* pada robot *Hovercraft*.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan di atas, maka pokok permasalahan yang terdapat dalam proyek akhir ini sebagai berikut :

1. Bagaimana membuat robot *Hovercraft* agar dapat bergerak dengan otomatis.
2. Bagaimana cara mengontrol kecepatan motor pendorong dan kemudi otomatis robot *Hovercraft* dengan metode *fuzzy logic*.

1.3 Batasan Masalah

Untuk memperjelas, menyederhanakan dan menghindari meluasnya masalah maka diberi batasan-batasan sebagai berikut:

1. Sensor hcsr04 digunakan untuk mengukur jarak robot terhadap dinding pembatas.
2. Robot *Hovercraft* diuji pada lintasan air dan lintasan darat, permukaan air yang tenang dan permukaan lantai yang datar, dengan dibuatkan dinding pembatas.
3. *Robot Hovercraft* dibuat hanya berupa *prototype*.
4. *Robot Hovercraft* ini hanya di kontrol menggunakan mikro kontroler Arduino.

5. *Robot Hovercraft* ini hanya di kontrol menggunakan metode *fuzzy logic*.
6. *Fuzzy logic* digunakan untuk mengontrol kecepatan motor *Brushless DC* pendorong.

1.4 Tujuan Penelitian

Tujuan yang diharapkan dari pembuatan robot ini yaitu :

1. Dapat membuat robot *Hovercraft* agar dapat bergerak secara otomatis.
2. Dapat mengontrol kecepatan motor pendorong dan kemudi otomatis robot *Hovercraft* dengan metode *fuzzy logic*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dengan pembuatan penelitian ini yaitu :

1. Dapat membuat program robot *Hovercraft* yang mampu bergerak secara otomatis dengan metode *fuzzy*.
2. Mengetahui proses pembuatan program dengan metode *fuzzy logic*.
3. Dapat membuat robot *Hovercraft* yang mampu bergerak secara otomatis.

BAB 2. TINJAUAN PUSTAKA

2.1 *Hovercraft*

2.1.1 Pengertian *Hovercraft*

Hovercraft dapat didefinisikan sebagai kendaraan yang berjalan di atas bantalan udara (*air cushion*). Bantalan udara tersebut di timbulkan dengan cara menyemburkan udara ke ruang bawah *Hovercraft* melalui sekat yang lentur sehingga tekanan udara di dalam *planum chamber* lebih tinggi dari pada tekanan udara di luar sehingga timbul gaya angkat. Untuk menggerakkan *Hovercraft* di perlukan gaya dorong yang diperoleh dari *propeller* seperti pada pesawat udara. *Hovercraft* bergerak dengan terbang melayang puluhan centimeter di atas permukaan yang dilaluinya. *Hovercraft* mempertahankan ketinggiannya dengan bantalan kipas yang meniup ke bawah melalui lambung dan bidang permukaan. *Hovercraft* dikatakan kendaraan serbaguna karena selama volume udara dalam bantalan itu tetap karena adanya dorongan udara oleh kipas kendaraan akan tetap pada ketinggian, *Hovercraft* meluncur di atas bantalan udara sehingga dapat dipakai di darat maupun di air. (Yuliawan, 2010).

2.1.2 Komponen Utama Pada *Hovercraft*

A. *Hull*

Hull adalah badan *Hovercraft* yang dapat dibuat dari *marine aluminium* atau *fiberglass*, serta dibuat kedap air. Rongga didalam *hull* diisi dengan *polyurethane foam* yang membuat *Hovercraft* tetap mengapung jika terjadi kebocoran pada *hull*. (Yuliawan, 2010).

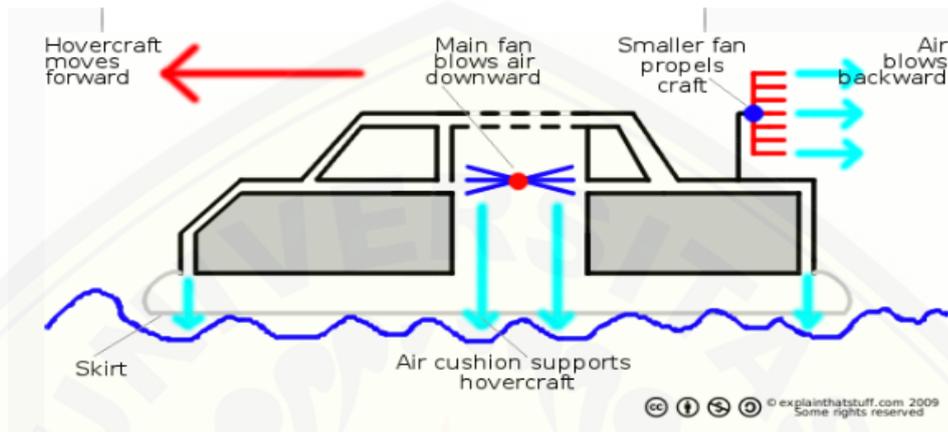
B. *Skirt*

Skirt merupakan bagian dari *Hovercraft* yang berfungsi untuk menahan udara agar tidak mudah keluar. *Skirt* terbuat dari tekstil yang dilapisi karet untuk menjaga agar udara tetap berada di dalam ruang dibawah *hull*. Pada *Hovercraft* sistem bantalan udara memegang peranan penting, karena pada sistem bantalan udara inilah *Hovercraft* dapat dibedakan dengan kapal konvensional. Sistem bantalan udara menyebabkan badan *Hovercraft* terangkat keatas permukaan air,

sehingga seolah-olah mengambang. Sistem yang digunakan untuk menimbulkan bantalan udara terbagi atas tiga kategori utama. (Yuliawan, 2010).

C. Mekanisme *Hovercraft*

Mekanisme *hover* menggunakan sistem *hover* seperti gambar dibawah ini



Gambar 2.1 Mekanisme *Hover* (Prihatama, 2011).

Secara teori prinsip utamanya *Hovercraft* yaitu tentang prinsip gaya angkat, yaitu dengan menerapkan rumus gaya angkat.

$$F = P \times A \dots\dots\dots(1)$$

Dimana :

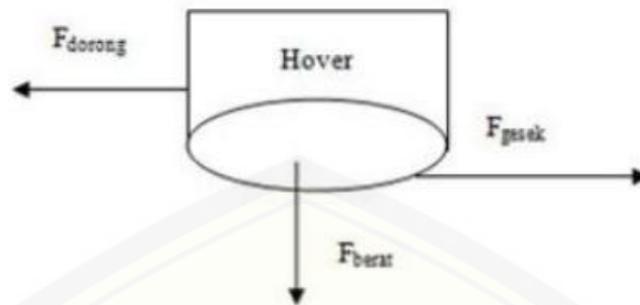
P : Tekanan (N/m² atau dn/cm²)

F : Gaya (N atau dn)

A: Luas alas/penanpang (m² atau cm²)

1. Sistem Angkat dan Daya Angkat

Kendaraan yang berjalan di atas bantalan udara. Bantalan udara ditimbulkan dengan cara memasukkan udara ke ruang bawah plastik bantalan udara melalui rongga dari putaran kipas, sehingga tekanan udara di dalam plastik bantalan udara lebih tinggi dari pada tekanan udara luar, maka dengan sendirinya *hover* akan terangkat dan melayang dari permukaan air atau tanah sehingga kendaraan *hover* dapat melakukan perjalanan tanpa gesekan. (Prihatama, 2011).



Gambar 2.2 Gaya Angkat Pada *Hover* (Prihatama, 2011).

a. Daya untuk Sistem *Lifter* (N_t)

Daya yang diperlukan untuk mengangkat *Hovercraft* merupakan hasil kali tekanan *hover* dengan debit udara yang bekerja dibawah *Hovercraft*. Berdasarkan L.yun dan A.Bliauit secara teoritis untuk menentukan daya pada sistem *lifter* (N_t) menggunakan rumus sebagai berikut:

$$N_t = (H \times Q) / (\eta_f \times \eta_t) \dots \dots \dots (2)$$

Dimana :

H = tekanan total fan (N/m^2)

Q = volume udara angkat (m^3 /s)

F_h = efisiensi fan

M_h = efisiensi transmisi

(Sumber : L.Yun dan A.Bliault, 2000)

b. Debit Udara Angkat (Q)

Debit udara angkat merupakan volume udara yang keluar melalui celah hover tiap satuan waktu sehingga dapat memenuhi kebutuhan debit udara sistem *lift* pada *proses flying*. Untuk menentukan besar debit udara angkat adalah sebagai berikut.

$$Q = Q' \times S_c \times v(2 \times P_c/p_a) \dots \dots \dots (3)$$

Dimana:

Q' = koefisien aliran udara angkat *Hovercraft* berdasarkan statistical method nilainya 0.015 – 0.030

W = berat total Hovercraft (N)

Sc = *cushion pressure* (m^2)

Pc = tekanan oleh berat total Hovercraft terhadap luasan bantalan tekan (N/m^2)

$\rho_a = 1.2257 \text{ kg/m}^3$

Debit (Q) merupakan beban aliran yang harus dikeluarkan oleh *fan* yang digunakan untuk sistem *lift*. (Sumber : *L.Yun dan A.Bliault, 2000*)

c. Tekanan Total *fan* (H)

Berdasarkan *L.Yun dan A.Bliault* untuk menentukan nilai tekanan total *fan* (H) maka terlebih dahulu harus dipilih jenis *fan* yang akan digunakan. Tekanan total dari *fan* adalah jumlah tekanan total yang dihasilkan oleh *fan* untuk menghasilkan gaya angkat, dapat dihitung dengan persamaan sebagai berikut. (Sumber : *L.Yun dan A.Bliault, 2000*)

$$H = H' \times \rho_a \times \mu^2 \dots \dots \dots (4)$$

Dimana:

H' = koefisien tekanan total *fan*

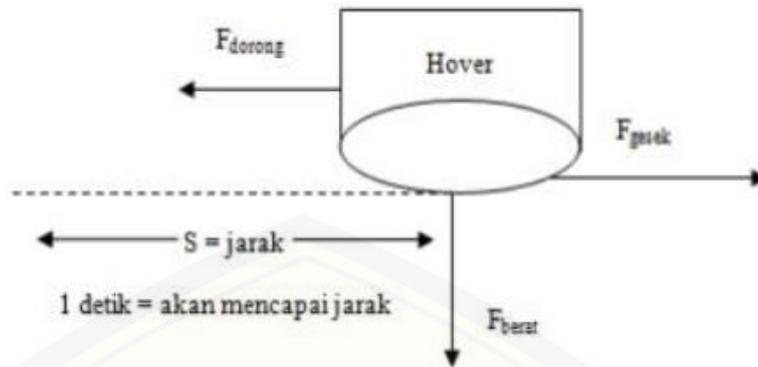
U_2 = *circular velocity of the fan impeller* (m/s), biasanya untuk *airfoil blade* $80 <$

$U_2 < 110 \text{ m/s}$

$\rho_a = 1.2257 \text{ kg/m}^3$

2. Sistem Dorong dan Gaya Dorong

Untuk menggerakkan kendaraan *hover* digunakan gaya dorong yang diperoleh dari *propeller* seperti pada pesawat udara, dikarenakan *hover* membutuhkan tenaga dorong yang lebih besar dari kendaraan lainnya. Sistem dorong biasanya menggunakan tenaga gerakan kipas. Untuk menggerakkan *hover* maju, mundur maupun berbelok, merupakan fungsi utama dari sistem dorong.



Gambar 2.3 Gaya Dorong Pada *Hover* (Prihatama, 2011).

Berdasarkan (Victor L. Streeter, 1979). Secara teoritis perhitungan untuk mencari gaya dorong *fan* aksial (N), yaitu dengan menggunakan rumus sebagai berikut:

1. Kecepatan sudu jalan *fan* aksial:

$$u_1 = \frac{\pi \cdot D_{h1} \cdot N_1}{60} \dots\dots\dots(5)$$

Dimana :

U_1 = Kecepatan keliling *fan* dalam (m/s)

D_{h1} = Diameter sudu jalan (m)

N_1 = Putaran mesin (rpm)

2. Kecepatan relatif pada fan.

$$w_1 = \frac{u_1}{\sin \beta_1} \dots\dots\dots(6)$$

Dimana:

w_1 = Kecepatan relatif (m/s)

U_1 = Kecepatan keliling *fan* (m/s)

$\sin \beta_1$ = Sudut kemiringan sudu

3. Kecepatan *absolut* pada *fan*.

$$C = w_1 \cdot \sin \beta_1 \dots\dots\dots(7)$$

Dimana :

w_1 = Kecepatan relatif (m/s)

C = Kecepatan *absolut* (m/s)

$\sin \beta_1$ = Sudut kemiringan sudu

4. Luas penampang sudu *fan* aksial.

$$A = \frac{1}{4} \pi D^2 \dots\dots\dots(8)$$

Dimana:

A = Luas penampang total kipas (m^2)

D = Diameter kipas

5. Menentukan laju aliran massa yang terjadi pada *fan* aksial sbb :

$$\dot{m} = \rho \cdot A \cdot C \dots\dots\dots(9)$$

Dimana :

ρ = Massa jenis fluida yang tetap/*stagnan* (Kg/m^3)

A = Luas penampang kipas (m^2)

C = Kecepatan *absolut* (m/s)

6. Debit *air propeler* (Q) (Victor L. Streeter, 1979)

$$Q = \dot{m} / \rho \dots\dots\dots(10)$$

Dimana :

\dot{m} = Laju aliran massa (Kg/s)

ρ = Massa jenis udara masuk (Kg/m^3)

Q = Debit udara (m^3/s)

7. Menentukan kecepatan udara keluar (C_2) (Victor L. Streeter, 1979)

$$C_2 = \frac{Q}{A_2} \dots\dots\dots(11)$$

Dimana :

C_2 = Kecepatan udara keluar (m/s)

A_2 = Luas penampang *cover outlet fan* aksial (m²)

Q = Debit udara (m³/s)

8. Menentukan kecepatan udara (C_1)

$$C_1 = \frac{Q}{A_1} \dots\dots\dots(12)$$

Dimana :

C_1 = Kecepatan udara masuk(m/s)

A_1 = Luas penampang *cover inlete fan* aksial (m²)

Q = Debit udara (m³/s)

9. Perhitungan gaya dorong *fan* aksial (N) (Victor L. Streeter, 1979)

$$F_{fan} = \dot{m} \times C_2 \dots\dots\dots(13)$$

Dimana :

F_{fan} = Gaya dorong (N).

\dot{m} = Laju aliran massa (kg/s).

C_2 = Kecepatan udara keluar (m/s).

10. Perhitungan daya *fan* aksial (P)

$$P = F_{fan} \cdot C_2 \dots\dots\dots(14)$$

Dimana :

P = Daya *fan* (Kw)

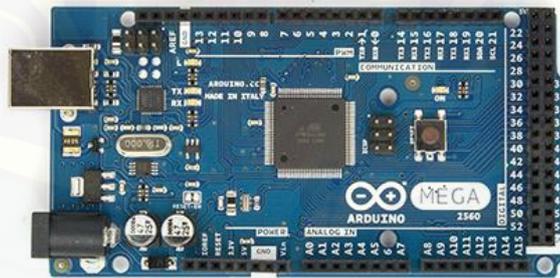
F_{fan} = Gaya dorong *hovercraft* (N)

C_2 = Kecepatan udara keluar (m/s)

2.2 Arduino MEGA 2560

Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560 (datasheet ATmega2560). Arduino Mega2560 memiliki 54 pin digital *input/output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (port serial *hardware*), 16 MHz kristal osilator, koneksi USB, jack power, *header* ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. (Dede Hendriono, 2014)

Arduino Mega2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega2560 adalah versi terbaru yang menggantikan versi Arduino Mega.



Gambar 2.4 Arduino Mega2560 (Yulias, 2013)

2.2.1 Ringkasan Spesifikasi

Dibawah ini adalah spesifikasi sederhana dari Arduino Mega2560:

Mikrokontroler	: ATmega2560
Tegangan Operasi	: 5V
<i>Input Voltage</i> (disarankan)	: 7-12V
<i>Input Voltage</i> (limit)	: 6-20V
Pin Digital I/O	: 54 (yang 15 pin digunakan sebagai <i>output</i> PWM)
<i>Pins Input</i> Analog	: 16

Arus DC per pin I/O	: 40 mA
Arus DC untuk pin 3.3V	: 50 mA
Flash Memory	: 256 KB (8 KB digunakan untuk <i>bootloader</i>)
SRAM	: 8 KB
EEPROM	: 4 KB
Clock Speed	: 16 Hz

2.2.2 Sumber Daya

Sumber daya pada Arduino Mega dapat menggunakan koneksi USB atau dengan catu daya eksternal. Catu daya eksternal dapat menggunakan power supply dengan menghubungkan pada jack sumber tegangan pada Arduino. Jika sumber daya dari baterai, dapat langsung di hubungkan pada pin GND dan pin Vin pada Arduino Mega.

2.2.3 Memori

Arduino ATmega2560 memiliki 256 KB *flash* memori untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

2.2.4 Input dan Output

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara *default*) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- **Serial 0** : 0 (RX) dan 1 (TX); **Serial 1** : 19 (RX) dan 18 (TX); **Serial 2** : 17 (RX) dan 16 (TX); **Serial 3** : 15 (RX) dan 14 (TX). Digunakan untuk menerima

(RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.

- **Eksternal Interupsi** : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
- **SPI** : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan *header* ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.
- **LED** : Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diatur bernilai LOW, maka LED padam (OFF).
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi `analogReference()`.

Ada beberapa pin lainnya yang tersedia, antara lain:

- **AREF** : Referensi tegangan untuk *input* analog. Digunakan dengan fungsi `analogReference()`.
- **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada *shield* yang menghalangi papan utama Arduino.

2.2.5 Komunikasi

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino ATmega328 menyediakan 4 *hardware* komunikasi serial UART TTL (5 Volt). Sebuah chip ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM Port *Virtual* (pada *Device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi *Windows* masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan *Linux* akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

2.3 Motor *Brushless* DC (BLDC)

Motor *Brushless* DC merupakan salah satu jenis motor sinkron magnet permanen yang di suplai oleh sumber listrik DC pada kontrolnya, dan membutuhkan sumber listrik AC tiga fasa untuk menggerakkan bagian rotor motornya. Motor *Brushless* membutuhkan sumber listrik AC 3 fasa karena motor ini memiliki sinkron magnet permanen memiliki 3 buah koil pada stator, hubungan antara koil dan belitan stator *trapezoidal* akan memberikan *electro motive back trapezoidal* (gaya gerak listrik balik *trapezoidal*) yaitu tegangan balik yang dihasilkan oleh belitan motor *Brushless* DC yang akan menggerakkan rotor. Pergerakan dari rotor ini disebabkan oleh medan magnet pada stator yang setiap suplainya hanya dua fasa sementara satu suplai lainnya tidak tersuplai. Sehingga mengakibatkan motor ini seperti motor DC biasa, karena arus yang mengalir ke stator mirip dengan motor DC biasa meskipun sebenarnya dialiri dengan arus tiga fasa.

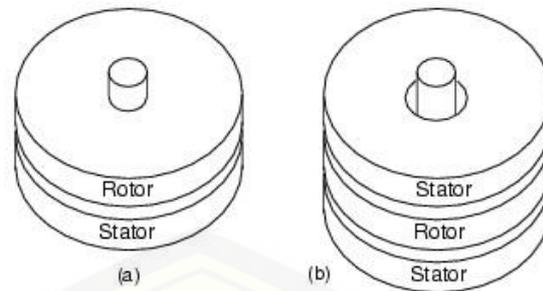
Sistem dari motor *Brushless* DC menggunakan komutasi elektrik atau yang sering di sebut *electronically commutated* motor atau dapat di sebut sebagai *switch electronic*. Komutasi elektronik biasanya menggunakan kombinasi transistor atau biasanya menggunakan MOSFET atau IGBT yang membutuhkan sinyal atau pulsa penyalan, dengan menggerakkan koil pada waktu yang tepat sehingga dapat menggerakkan motor.



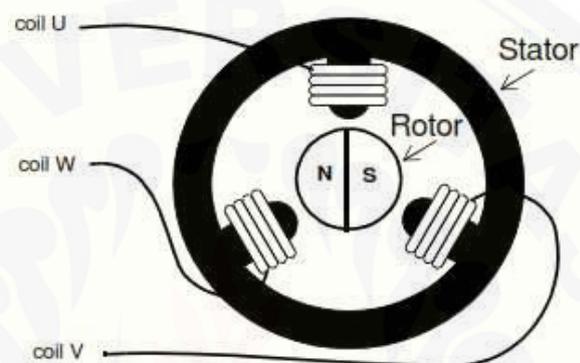
Gambar 2.5 Motor *Brushless* DC (Rusdianto, 2012)

Motor BLDC tidak menggunakan sikat-sikat untuk komutasinya dan memiliki kecepatan yang sinkron antara medan putar stator dan rotornya. Rotor motor BLDC adalah magnet permanen sehingga tidak diperlukan kumparan penguat dan tidak ada arus beban yang mengalirinya. Metode pengaturan kecepatan pada motor BLDC ada 2 macam, yaitu pengaturan tegangan catu dan pengaturan percepatan sudut fasa (*phase advance angle*). Pada pengaturan percepatan sudut fasa, kecepatan optimal dapat diperoleh ketika arus dan ggl bersamaan dalam waktu atau dengan kata lain sefase (Munawar dkk, 2005).

Motor DC *Brushless* memiliki 3 macam jenis motor berdasarkan banyaknya fasa, yaitu motor *Brushless* 1 fasa, 2 fasa, 3 fasa. Pada motor DC *Brushless* memiliki jumlah belitan yang sama pada stator nya, dan yang sering digunakan adalah motor DC *Brushless* 3 fasa.



Gambar 2.6 Motor *Brushless* DC (a) 1 fasa (b) 2 fasa (Mamzulil, 2012)



Gambar 2.7 Motor *Brushless* DC 3 Fasa (Mamzulil, 2012)

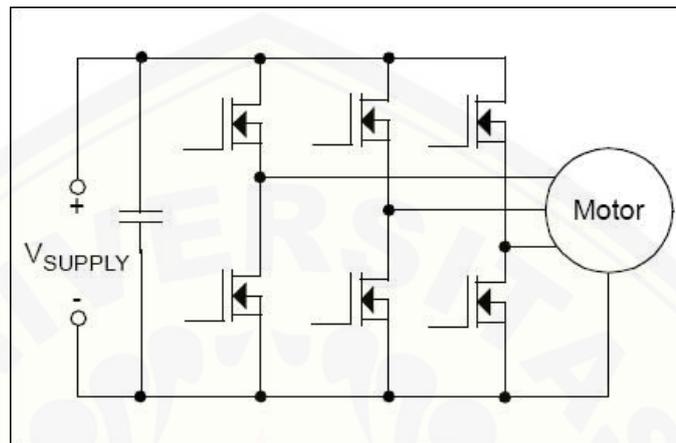
2.3.1 Bagian - Bagian Motor *Brushless* DC

Adapun bagian – bagian terpenting dari motor *Brushless* DC sebagai bahan pelengkap dari motor *Brushless* DC.

1. Kontroler, *Driver* dan *Inverter*

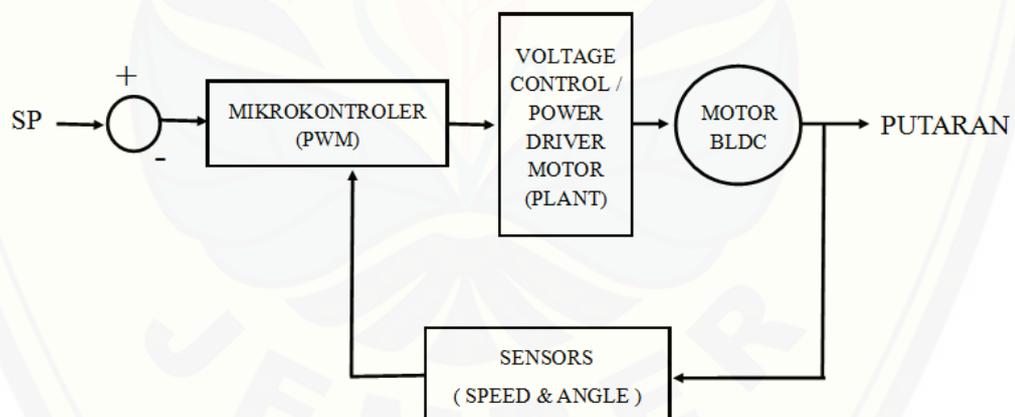
Kontroler, *driver* dan *inverter* merupakan bagian terpenting pada motor *Brushless* DC yang berfungsi sebagai pengendali atau penggerak putaran pada motor tersebut. Kontroler, *driver* dan *inverter* merupakan rangkaian elektronik yang di rangkai sedemikian rupa hingga menjadi satu kesatuan sistem dalam mengendalikan motor *Brushless*. Pengendali utama dari kontroler ini yaitu mikrokontroler sebagai peendali utama, maka digunakan mikrokontroler yang sudah terprogram. Sedangkan untuk *driver* komponen utama nya yaitu menggunakan rangkaian dari MOSFET, sebagai penguat atau *switching* agar arus yang mengalir mampu nggerakkan motor tersebut.

Kontroler berfungsi sebagai pengatur kecepatan pada motor brushles dengan mengatur kerja dari *driver* dan *inverter*. Inverter berfungsi sebagai pengubah tegangan DC menjadi tegangan ac dan di rubah lagi ke DC untuk mensuplai *driver*.



Gambar 2.8 Penggerak Elektronik Daya Untuk Motor *Brushless* DC

(Rambabu, 2007)



Gambar 2.9 Blok Diagram Sistem Kontroler Dan *Driver* Pada Motor

Brushless DC (Rambabu, 2007)

Berdasarkan kemampuan *control power suplay*, dapat dipilih ranting tegangan sesuai dengan kebutuhan, jika motor yang digunakan memiliki ranting tegangan sebesar 48V maka atau di bawah dari itu, biasanya digunakan pada bidang otomotif atau pada robotik. Dan jika pada

motor tersebut memiliki ranting tegangan sebesar 100V maka itu merupakan motor yang dipakai pada bidang industri.

Dalam perancangan dan pembuatan driver dan *inverter* diperlukan pemilihan tipe dari MOSFET (*metal oxider semikonduktor field. Effect transistor*) yang tepat sesuai dengan kebutuhan motor yang akan digunakan. Selain menggunakan MOSFET juga dibutuhkan optocoupler yang berfungsi sebagai penghubung rangkaian daya ke rangkaian kontrol, dan juga berfungsi sebagai pengamanan mikrokontroler dan MOSFET jika terjadi *over current* atau arus yang mengalir melebihi batas maximum pada rangkaian.

2. Stator

Stator merupakan bagian dalam motor yang diam atau bersifat statis pada motor, yang berfungsi sebagai medan putar motor untuk memberikan gaya elektromagnetik pada rotor sehingga motor dapat berputar.



Gambar 2.10 Gambar Stator Pada Motor *Brushless* DC (Dirjis, 2012)

Pada motor *Brushless* DC, stator terdiri dari 12 belitan (elektromagnetik) yang bekerja secara elektromagnetik, dimana stator pada motor *Brushless* DC yang terhubung dengan tiga buah kabel yang masing2 memiliki fasa yang di hubungkan pada kontrol, sedangkan pada motor DC konvensional hanya memiliki dua buah stator yang terdiri dari kutub magnet permanen.

3. Rotor

Rotor merupakan bagian penting yang berfungsi sebagai bagian untuk menggerakkan motor atau membuat motor berputar. Perputaran tersebut terjadi dikarenakan adanya gaya elektromagnetik yang dihasilkan oleh stator. Untuk mengetahui putaran pada motor dapat menggunakan persamaan berikut ini:

$$n = \left(\frac{V - I_a \times R_a}{K \times \Phi} \right) \text{ (Rpm)} \dots\dots\dots(15)$$

dimana :

n = putaran motor (Rpm)

V = tegangan kerja motor (Volt)

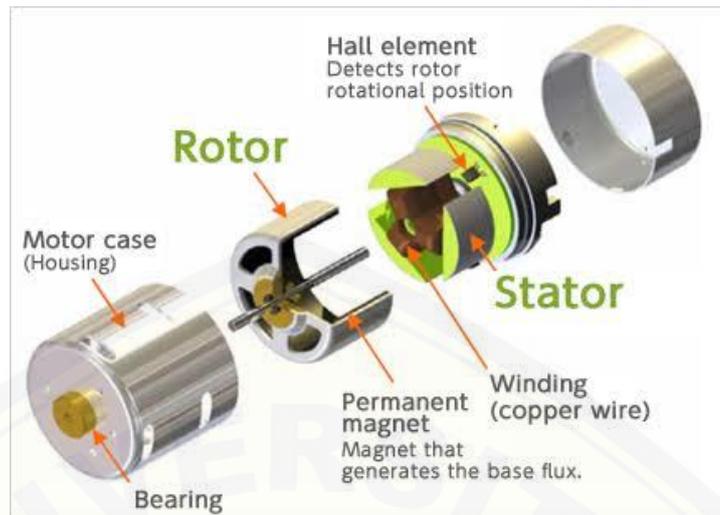
I_a = arus jangkar (A)

R_a = tahanan jangkar (Ω)

K = konstanta motor

Φ = fluks medan magnet (Wb)

Rotor pada motor *Brushless* DC berbeda dengan rotor pada motor DC konvensional. Pada motor DC konvensional rotornya tersusun dari 1 buah elektromagnet yang berbeda diantara *brushes* (sikat) yang terhubung pada 2 buah elektroda yang terangkai ke suplai DC. Sedangkan pada motor *Brushless* DC bagian rotornya tersusun dari 2 hingga 8 pasang kutub magnet permanen berbentuk persegi panjang yang saling direkatkan menggunakan semacam “*epoxy*” dan tidak memiliki sikat.



Gambar 2.11 Komponen Pada Motor *Brushless* DC (Nickned, 2013)

2.4 *Electronic Speed Controller (Esc)*

ESC (*Elektronik Speed Control*) yang berfungsi sebagai pengatur kecepatan motor, selain itu juga berfungsi untuk menaikkan jumlah arus yang diperlukan oleh motor. ESC dapat dikatakan juga sebagai *Drive* motor dengan mengeluarkan pulsa untuk *Brushless* motor yang berasal dari mikrokontroler.



Gambar 2.12 *ESC (Electronic Speed Controller)* (solehudin, 2013)

2.5 Baling – Baling (*Propeller*)

Baling-baling adalah alat yang mengubah gerak putar menjadi daya dorong. Daya dorong inilah yang dimanfaatkan oleh *Hovercraft* sebagai penghasil daya dorong utama. Bilah-bilah dari *propeller* berperan sebagai sayap yang berputar menghasilkan sebuah perbedaan tekanan antara permukaan depan dan belakang bilah tersebut. Ada beberapa parameter penting yang dimiliki *propeller* pada *Hovercraft*. Parameter-parameter ini bisa dijadikan pedoman untuk memilih baling-baling sesuai kebutuhan:

1. Diameter dan *pitch*

Semua baling-baling RC yang tersedia memiliki 2 buah ukuran, yaitu diameter dan *pitch*. Diameter dihitung berdasarkan diameter lingkaran yang dibentuk saat baling-baling berputar. Sedangkan *pitch* merupakan jarak yang ditempuh oleh baling-baling jika diputar 1 putaran penuh. Semakin panjang diameter dan *pitch* baling-baling semakin banyak pula udara yang disapu dan semakin besar pula daya dorong yang dihasilkan. Tapi diameter dan *pitch* dari baling-baling ini harus disesuaikan dengan motor dan sumber daya yang digunakan. Satuan dari diameter dan *pitch* dari baling-baling adalah inch. Baling-baling dengan ukuran 10x4.5 memiliki diameter 10 inch dan *pitch* 4.5 inch.

2. Jumlah bilah

Umumnya jumlah bilah pada baling-baling *Hovercraft* adalah 2 bilah. Tetapi ada yang menggunakan 3 bilah dan 4 bilah. Semakin banyak bilah pada baling-baling menyebabkan semakin banyak udara yang disapu sehingga menghasilkan daya dorong yang lebih besar. Semakin banyak bilah juga menuntut motor dengan torsi yang lebih besar. Biasanya penambahan jumlah bilah bertujuan untuk memperkecil diameter baling-baling, tentunya untuk menghasilkan performa yang sama (dengan motor yang sama) *pitch*-nya harus dikurangi.



Gambar 2.13 *Propeller* Dengan Berbagai Bilah (Sumber:

http://www.troybuiltmodels.com/ns/images/items/biela/biela_2blade.jpg.)

3. Arah putar

Dengan arah gaya dorong yang sama, baling-baling *Hovercraft* memiliki dua jenis arah putaran; searah jarum jam (*CW, clockwise*) dan berkebalikan arah jarum jam (*CCW, counter clockwise*). Arah putaran baling-baling ini menentukan gerakan dari robot *Hovercraft*.

2.6 Motor Servo

Pengembangan dari DC Motor dan sudah memiliki *Gear*, tapi bedanya Motor Servo bisa diatur kecepatannya. Di dalamnya terdapat Potensio Meter dan *Driver Motor*, sehingga bisa diatur dengan derajat. Motor servo bisa digunakan untuk membuat robot berkaki, robot lengan, dan lain-lain. Batas maksimal pergerakan putar kanan hingga 180 derajat dan putar kiri hingga 180 derajat.



Gambar 2.14 Motor Servo (Gonzalez, 2016)

Motor servo adalah sebuah perangkat atau aktuatur putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat di *set-up* atau di atur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas putaran poros motor servo.

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Wide Modulation / PWM*) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya. Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya. (Trikueni Dermanto, 2014)

2.7 Sensor HC-SR04

2.7.1 Pengertian sensor HC-SR04

Sensor *Ultrasonic* adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan jarak suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonic karena sensor ini menggunakan gelombang ultrasonic.

HC-SR04 memiliki kinerja yang baik dalam mendeteksi jarak, dengan tingkat akurasi yang tinggi serta deteksi yang stabil. Sensor ini memiliki fungsi

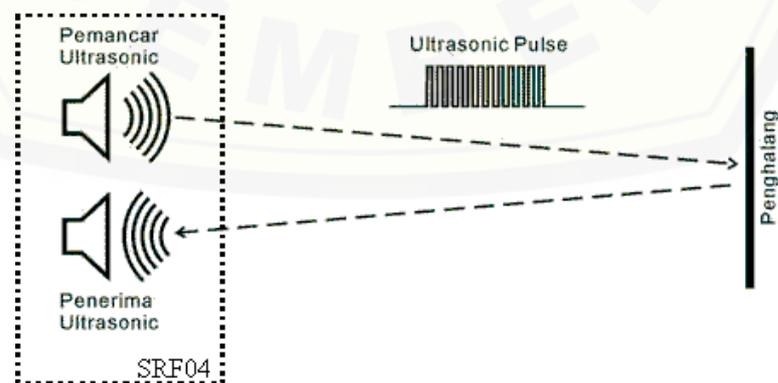
sebagai pengirim, penerima, dan pengontrol gelombang *Ultrasonic* dan juga dapat mengukur jarak benda dari 2cm – 4m dengan akurasi 3mm. sensor ini memiliki 4 pin yaitu *Vcc*, *Gnd*, *Trigger*, dan *echo*. Untuk pin *Trigger* keluarannya sinyal dari sensor dan pin *Echo* untuk menangkap sinyal pantul dari benda. (Hari Santoso, 2015)

Tabel 2.1 Data Sheet *Ultrasonic* HC SR04

Voltage DC	5 V
Working Currensnt	15 mA
Working Frequency	40 Hz
Max Range	4 m
Min Range	2 cm
Measuring Angle	15 degree
Dimension	45 x 20 x 15 mm
Trigger <i>Input</i> Signal 10uS TTL pulse	
Echo <i>Output</i> Signal <i>Input</i> TTL lever signal and the range in proportion	

Berikut adalah pin dari HC-SR04 :

1. VCC : *Input supply 5V*
2. Trig : *Input* untuk memberikan pulsa *trigger*
3. Echo : *Output* untuk pulsa *Echo*
4. GND : *Input supply Ground*



Gambar 2.15 Cara Kerja Sensor HC-SR04 (Priswanto, 2016)

2.7.2 Prinsip kerja sensor jarak *Ultrasonic*

Prinsip pengukuran jarak menggunakan sensor *Ultrasonic* HC-SR04 adalah, ketika pulsa *trigger* diberikan pada sensor, *transmitter* akan mulai memancarkan gelombang *Ultrasonic*, pada saat yang sama sensor akan menghasilkan *output* TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah *receiver* menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan *output* TTL transisi turun. Jika waktu pengukuran adalah t dan kecepatan suara adalah 340 m/s, maka jarak antara sensor dengan objek dapat dihitung dengan menggunakan Persamaan 2.

$$s = t \times \frac{340 \text{ m/s}}{2} \dots\dots\dots (16)$$

Dimana :

s = Jarak antara sensor dengan objek (m)

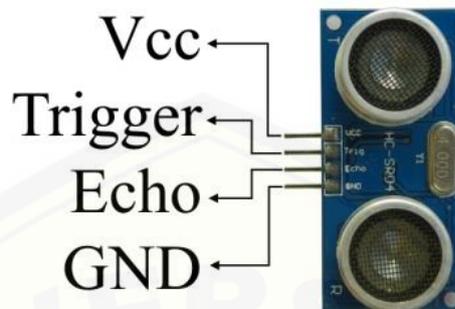
t = Waktu tempuh gelombang *Ultrasonic* dari *transmitter* ke *receiver* (s)

SRF-04 hanya menggunakan 2 port I/O untuk berhubungan dengan mikrokontroler, sehingga sangat ideal untuk aplikasi-aplikasi robotika, SRF-04 dapat mengukur jarak mulai 3 cm sampai 3 m, dan dapat mengukur benda dengan diameter 3 cm pada jarak kurang dari 2 meter.

2.7.3 Bentuk Fisik HC-SR04

Pulsa *Ultrasonic* yang dikirim oleh SRF04 adalah sinyal *Ultrasonic* dengan frekuensi 40 KHz sebanyak 8 periode setiap kali pengiriman. Ketika pulsa mengenai benda penghalang, maka pulsa ini akan dipantulkan kembali dan diterima kembali oleh penerima *Ultrasonic*. Dengan mengukur selang waktu antara saat pulsa dikirim dan pulsa pantul diterima, maka jarak benda penghalang bisa dihitung. Apabila PI (*triger pulse input*) diberi logika 1 (high) selama minimal 10 uS maka SRF04 akan memancarkan sinyal *Ultrasonic*, setelah itu pin PO (*echo pulse output*) akan berlogika *high* selama 100 uS – 18 mS. (tergantung

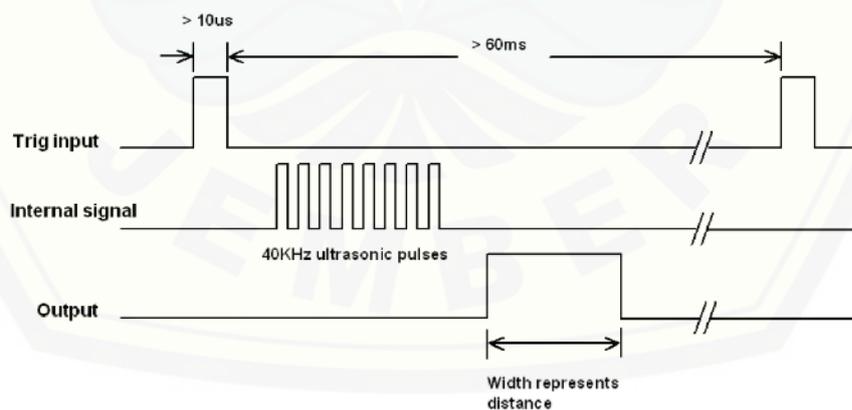
jarak sensor dan penghalang) dan apabila tidak ada penghalang maka PO akan berlogika 1 selama kurang lebih 38 mS.



Gambar 2.16 Konfigurasi Pin Dan Tampilan Sensor *Ultrasonic* Hc-Sr04
(France, 2013)

2.7.4 Timing Diagram HC-SR04

Misalkan lama *Echo Pulse* adalah T , maka untuk mengetahui jaraknya dapat diketahui dengan cara membagi T dengan 58 ($T/58$) untuk satuan centimeter dan dibagi dengan 148 ($T/148$) untuk satuan inchi. Misalkan panjang *Echo Pulse* adalah 5800 mikro detik maka jarak benda adalah 1 meter ($5800/58 = 100 \text{ cm} = 1 \text{ meter}$).



Gambar 2.17 *Timing Diagram* Pengoperasian Sensor *Ultrasonic* Hc-Sr04
(France, 2013)

2.8 Pengendali Logika *Fuzzy* / *Fuzzy logic Controller*

2.8.1 Pengertian *Fuzzy logic*

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*. Titik awal dari konsep modern mengenai ketidakpastian adalah paper yang dibuat oleh Lofti A Zadeh (1972), dimana Zadeh memperkenalkan teori yang memiliki obyek-obyek dari *himpunan fuzzy* yang memiliki batasan yang tidak presisi dan keanggotaan dalam himpunan *fuzzy*, dan bukan dalam bentuk logika benar (*true*) atau salah (*false*), tapi dinyatakan dalam derajat (*degree*). Konsep seperti ini disebut dengan *Fuzziness* dan teorinya dinamakan *Fuzzy Set Theory*. *Fuzziness* dapat didefinisikan sebagai logika kabur berkenaan dengan semantik dari suatu kejadian, fenomena atau pernyataan itu sendiri. Seringkali ditemui dalam pernyataan yang dibuat oleh seseorang, evaluasi dan suatu pengambilan keputusan.

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

Logika *fuzzy* adalah peningkatan dari logika *Boolean* yang berhadapan dengan konsep kebenaran sebagian. Dimana logika klasik (*crisp*) menyatakan bahwa segala hal dapat diekspresikan dalam istilah *binary* (0 atau 1, hitam atau putih, ya atau tidak). Logika *fuzzy* menggantikan kebenaran *Boolean* dengan tingkat kebenaran. Logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk *linguistic*, konsep tidak pasti seperti “sedikit”, “lumayan”, dan “sangat”. Logika ini diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Barkeley pada tahun 1965. Logika *fuzzy* telah digunakan pada bidang-bidang seperti taksonomi, topologi, linguistik, teori automata, teori pengendalian, psikologi, *pattern recognition*, pengobatan, hukum, *decision analysis*, *system theory and information retrieval*. Pendekatan *fuzzy* memiliki kelebihan pada hasil yang terkait dengan sifat kognitif manusia, khususnya pada situasi yang melibatkan pembentukan konsep,

pengenalan pola, dan pengambilan keputusan dalam lingkungan yang tidak pasti atau tidak jelas.

Ada beberapa alasan mengapa orang menggunakan logika *fuzzy* (Kusumadewi 2003) antara lain:

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.8.3 Konsep *Fuzzy logic*

Konsep penggunaan *Fuzzy logic* antara lain:

1. *Fuzzy logic* umumnya diterapkan pada masalah-masalah yang mengandung unsur ketidakpastian (*uncertainty*), ketidaktepatan (*imprecise*), *noisy*, dan sebagainya.
2. *Fuzzy logic* menjembatani bahasa mesin yang presisi dengan bahasa manusia yang menekankan pada makna atau arti (*significance*).
3. *Fuzzy logic* dikembangkan berdasarkan cara berfikir manusia

2.8.4 Himpunan *Fuzzy*

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

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

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

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan *fuzzy* USIA adalah 0,9 maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda. Himpunan *fuzzy* memiliki 2 atribut, yaitu:

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

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Aplikasi logika *fuzzy* untuk pendukung keputusan, Sri Kusumadewi, Hari Purnomo, Edisi kedua, Graha Ilmu, 2010), yaitu:

- a. Variable *fuzzy* Variable *fuzzy* merupakan variabel yang hendak dibahas dalam suatu system *fuzzy*. Contoh: umur, temperature, permintaan, dsb.
- b. Himpunan *Fuzzy* Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.
- c. Semesta Pembicaraan Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasibatas atasnya.

Contoh:

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

d. Domain

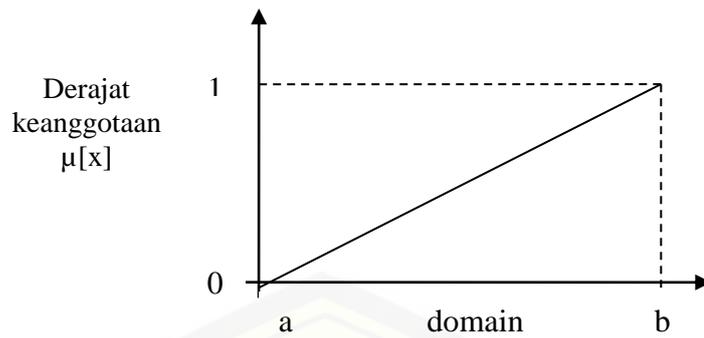
Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain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

2.8.5 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi *fuzzy* dalam U , *Then* A dapat dinyatakan sebagai pasangan terurut (Wang, 1997 yang dirujuk Wulandari, F. 2005). Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang *linear*. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol(0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi (Kusumadewi S, Purnomo H, 2010). Seperti terlihat pada gambar 2.16

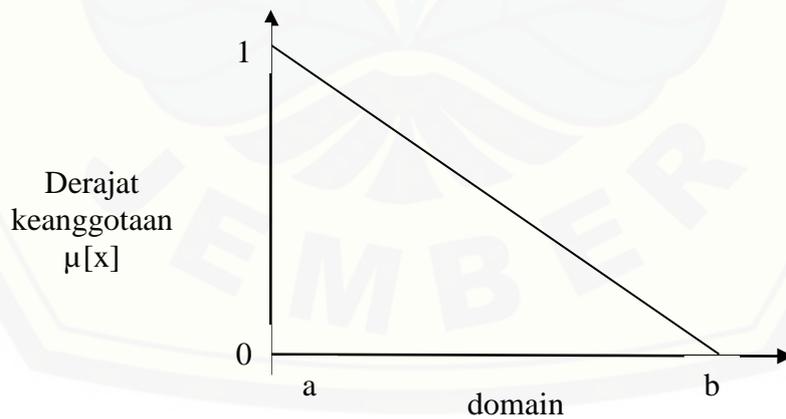


Gambar 2.18 Representasi Linear Naik (Kusumadewi S, Purnomo H, 2010)

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a < x < b \\ 1; & x \geq b \end{cases} \dots\dots\dots(17)$$

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih pendek. Seperti terlihat pada gambar 2.17.



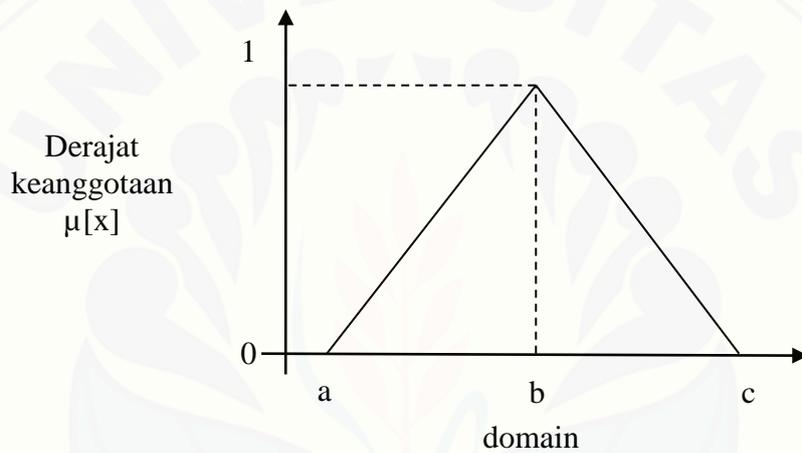
Gambar 2.19 Representasi Linear Turun (Kusumadewi S, Purnomo H, 2010)

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \geq b \\ (b-x) / (b-a) & a < x < b \dots\dots\dots (18) \\ 1; & x \leq a \end{cases}$$

b. Representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*).
Seperti terlihat pada gambar 2.18.



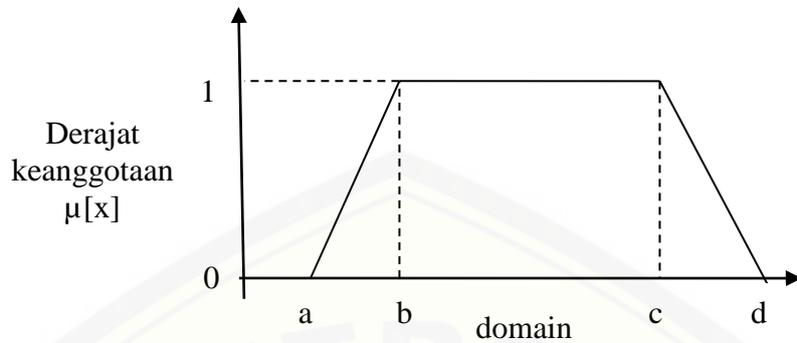
Gambar 2.20 Kurva Segitiga (Kusumadewi S, Purnomo H, 2010)

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \geq c \text{ atau } x \leq a \\ (x-a) / (b-a) & a < x < b \dots\dots\dots (19) \\ (c-x) / (c-b) & b < x < c \end{cases}$$

c. Representase kurva trapezium (Kusumadewi S, Purnomo H, 2010)

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Seperti terlihat pada gambar 2.19.



Gambar 2.21 Representasi Kurva Trapezium (Kusumadewi S, Purnomo H, 2010)

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \geq d \text{ atau } x \leq a \\ \frac{(x-a)}{(b-a)}; & a < x < b \\ \frac{(d-x)}{(d-c)}; & c < x < d \\ 1; & b \leq x \leq c \end{cases} \dots\dots\dots(20)$$

2.8.6 Sistem Inferensi Fuzzy

Salah satu aplikasi logika fuzzy yang telah berkembang amat luas dewasa ini adalah sistem inferensi fuzzy (*Fuzzy Inference System / FIS*), merupakan suatu kerangka komputasi yang di dasarkan pada teori himpunan fuzzy, aturan fuzzy berbentuk IF – THEN, dan penalaran fuzzy. (Kusumadewi S, Hartati S, 2010).

System inferansi fuzzy menggunakan metode SUGENO, memiliki karakteristik yaitu konsekuen tidak merupakan himpunan fuzzy, namun merupakan suatu persamaan linier dengan variabel-variabel sesuai dengan variabel-variabel inputnya. (Kusumadewi S, Hartati S, 2010).

2.8.7 Penegasan (defuzzifikasi)

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari suatu komposisi aturan – aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada himpunan fuzzy tersebut. Sehingga jika diberikan

suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*.

Fuzzifikasi merupakan fase pertama dari perhitungan *fuzzy*, yaitu mengubah masukan - masukan yang nilai kebenarannya bersifat pasti ke dalam bentuk *fuzzy input* yang berupa tingkat keanggotaan / tingkat kebenaran. Fuzzifikasi adalah proses perubahan suatu nilai *crisp* ke dalam variabel *fuzzy* yang berupa variabel linguistik yang nantinya akan dikelompokkan menjadi himpunan *fuzzy*. Dengan demikian, tahap ini mengambil nilai-nilai *crisp* dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, antara lain:

- a. Metode Centroid (Composite Moment)
- b. Metode Largest of Maximum (LOM)
- c. Metode Mean of Maksimum (MOM)
- d. Metode Bisektor

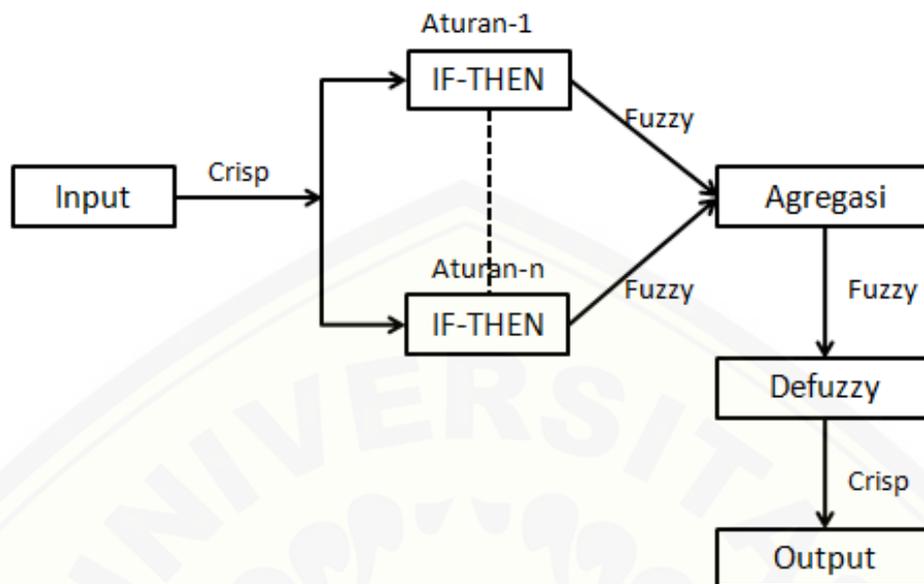
Pada tahap defuzzifikasi penulis menggunakan salah satu metode defuzzifikasi yaitu Metode Bisektor. Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

$$\mu_{(d)} = \frac{1}{2} \sum_{i=1}^n U_{A_i}(d_i) \dots \dots \dots (21)$$

Keterangan:

- d = nilai hasil penegasan (defuzzifikasi),
 d_i = nilai keluaran pada aturan ke-i,
 $\mu_{A_i}(d_i)$ = derajat keanggotaan nilai keluaran pada aturan ke-i,
 n = banyak aturan yang digunakan

Secara garis besar sistem inferensi *fuzzy* dapat dilihat pada gambar 2.20.



Gambar 2. 22 Tahapan Sistem Infransi *Fuzzy* (Kusumadewi, 2006)

BAB 3. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Adapun tempat dan waktu penelitian, pengujian dan analisis dilakukan secara umum di lakukan di :

Tempat : - Lab. Elektronika Terapan Fakultas Teknik Universitas Jember

Alamat : - Jl. Slamet Riyadi No. 62, Patrang, Kabupaten Jember

Waktu : Mei 2018 – Agustus 2018

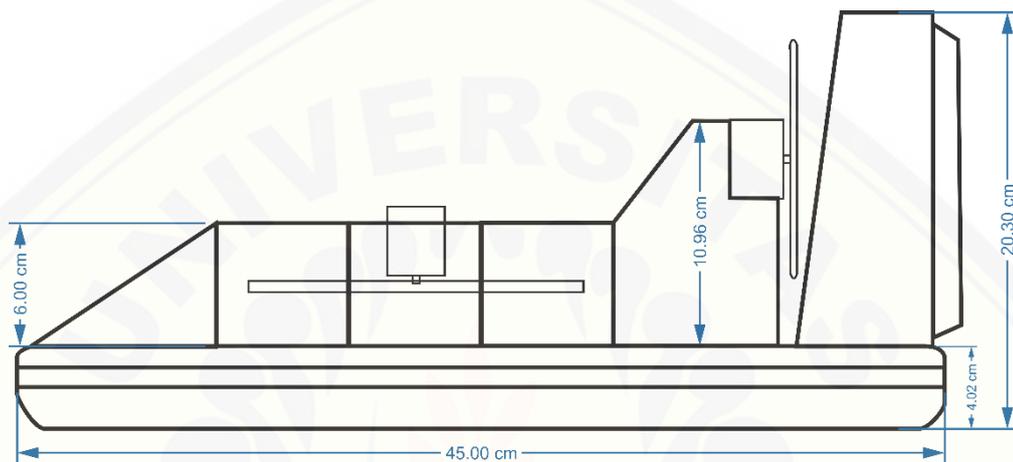
3.2 Rencana Jadwal Pelaksanaan Penelitian

Tabel 3.1 Tabel Penjadwalan Proses Pengerjaan Alat

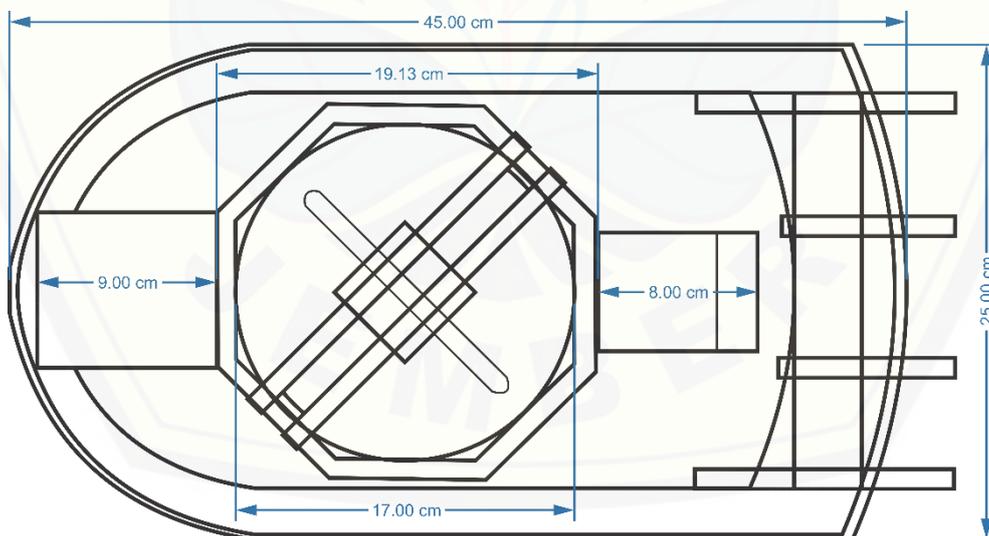
No	Kegiatan	Bulan															
		Mei				Juni				Juli				Agustus			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Persiapan	■	■														
2	Studi Literatur		■	■	■												
3	Pembuatan Desain Robot <i>Hovercraft</i>				■	■	■										
4	Pembelian Alat dan Bahan		■	■	■	■											
5	Pembuatan/Perakitan Robot <i>Hovercraft</i>					■	■	■	■	■	■	■	■				
6	Melakukan Pengujian Alat											■	■	■	■		
7	Analisa Data													■	■	■	■
8	Penulisan Laporan dan Kesimpulan													■	■	■	■

3.3 Perancangan Mekanik

Perancangan dan pembuatan robot *Hovercraft* dengan kemudi otomatis ini dibuat meliputi sensor HC-SR04, mikrokontroler, ESC (*electronic speed controller*), motor *Brushless DC*, servo, dengan bentuk robot seperti pada gambar dibawah ini:



Gambar 3.1 Desain Robot *Hovercraft* Tampak Samping



Gambar 3.2 desain robot *Hovercraft* tampak atas

Keterangan:

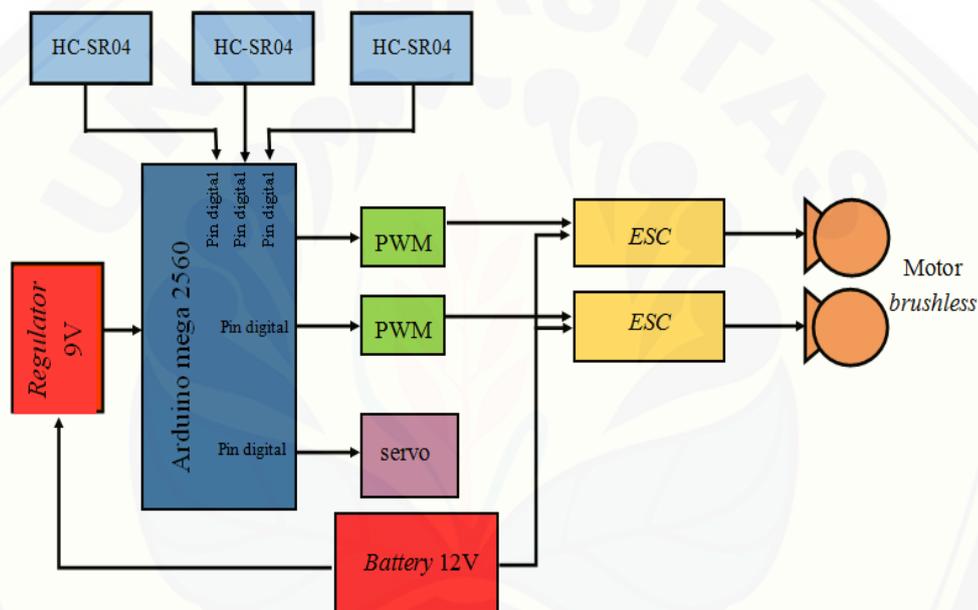
1. Baterai lipo 11.1 Volt
2. Arduino MEGA 2560
3. Sensor HC-SR04
4. ESC (*electronic speed controller*)
5. Motor *Brushless* DC
6. Baling-baling (*propeller*)
7. Motor servo
8. Sirip kemudi
9. Bantalan udara

Dari desain yang telah dibuat, sensor HC-SR04 berfungsi sebagai sensor pendeteksi jarak antara dinding dengan robot, yang di kontrol menggunakan mikrokontroler Arduino mega 2560, robot ini di gerakkan menggunakan motor *Brushless* DC dengan *electronic speed controller* sebagai *driver* motor *Brushless* dengan tegangan tiga fasa DC yang dilengkapi baling-baling sebagai pendorong udara supaya robot dapat mengambang dan bergerak maju, sebagai pengendali kemudi pada robot ini menggunakan servo yang dilengkapi sirip untuk membelokkan angin agar robot dapat belok sesuai arah sirip. Pada robot ini di lengkapi dengan bantalan udara (*skirt*) adalah ciri utama sebuah *Hovercraft*, yaitu sebuah bagian yang membuat kendaraan ini mengapung/mengambang dan fungsinya menahan udara agar tidak keluar, *skirt* sendiri terbuat dari bahan kain (tekstil) yang dilapisi karet,

3.4 Perancangan Elektronik

Kendaraan *Hovercraft* dapat bergerak dikarenakan memiliki mesin yang dapat menggerakkan kipas sehingga terjadi gaya angkat dan gaya dorong. Pada penelitian ini agar robot *Hovercraft* dapat bergerak maka dibuatkan perancangan elektronik untuk mengontrol pergerakan dari robot. Perancangan elektronik pada robot *Hovercraft* dapat dilihat pada gambar 3.3.

3.4.1 Diagram Blok Perancangan Elektronik



Gambar 3.3 Diagram Blok Perancangan Elektronik

Dari gambar 3.3 merupakan perancangan elektronika yang akan digunakan pada robot *Hovercraft*, dari perancangan elektronik ini diharapkan robot dapat bergerak sesuai yang di inginkan yaitu robot dapat menghindari dinding penghalang dan mampu menelusuri lintasan dengan lancar tanpa menabrak dinding lintasan.

Komponen yang digunakan pada robot *Hovercraft* memiliki fungsi yang dapat mendukung lancarnya pergerakan robot *hovercrat*, komponen-komponen tersebut yaitu:

1. Arduino MEGA 2560

Arduino MEGA 2560 merupakan sebuah papan mikrokontroler yang digunakan pengolahan data yang di terima dari sebuah input yaitu sensor ultrasonic dan hasil pengolahan akan di kirim kepada output yaitu motor *Brushless* DC dan motor servo. pada mikrokontrol ini nantinya akan mengolah data-data dengan menggunakan metode fuzzy logic.

2. Sensor HC-SR04

Sensor HC-SR04 ini berfungsi sebagai pengendali pergerakan robot, setelah data yang di hasilkan oleh sensor ini nantinya akan di olah oleh Arduino dan data hasil olahan ini akan di kirimkan pada motor servo untuk menentukan arah belok robot, dan hasil olahan yang menggunakan metode fuzzy nantinya akan dikirimkan pada motor DC *Brushless* pendorong untuk mengatur kecepatan motor tersebut.

3. ESC (*electronic speed controller*)

ESC merupakan sebuah modul elektronik yang berfungsi sebagai driver dari motor DC *Brushless*. ESC bekerja saat diberi nilai pwm yang di hasilkan oleh pengolahan data dari Arduino. ESC juga berfungsi sebagai penguat dari motor DC *Brushless*, motor *Brushless* ini dapat digunakan hanya menggunakan ESC karena pada rangkaian ESC ini memiliki output 3 fasa yang dibutuhkan oleh motor DC *Brushless*.

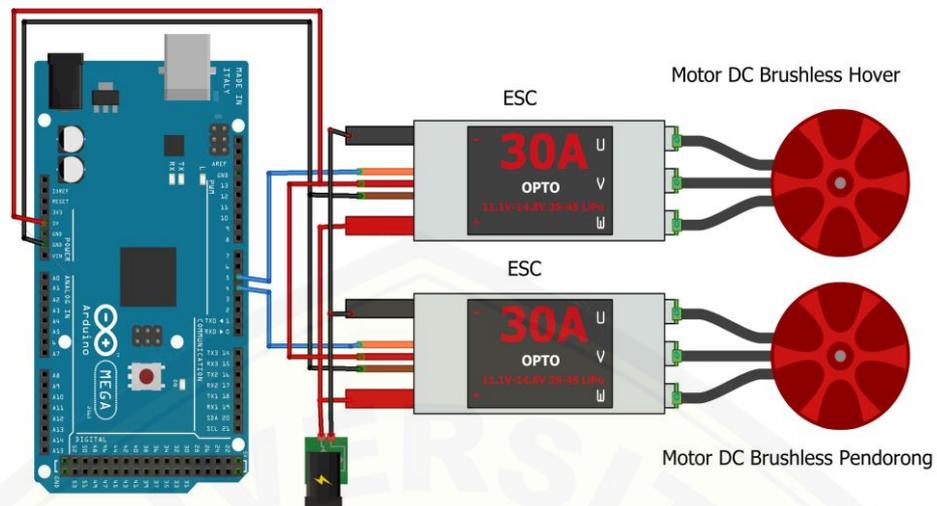
4. Motor DC *Brushless*

Motor DC *Brushless* pada robot *Hovercraft* berfungsi sebagai penggerak *propeller* (baling-baling), motor ini sangat dibutuhkan pada robot *Hovercraft* karena motor ini mempunyai RPM yang tinggi, sehingga menghasilkan hembusan angin yang kuat untuk mendorong robot *Hovercraft* dan juga dapat menghasilkan gaya angkat yang tinggi dan membuat robot dapat bergerak dengan ringan.

Pada gambar 3.4 yaitu gambar rangkaian sensor jarak, pada rangkaian ini menggunakan sensor jarak sebanyak 3 yaitu sebagai sensor jarak depan, sensor jarak kanan, sensor jarak kiri, ketiga sensor jarak ini digunakan sebagai *input* untuk pengendalian motor servo dan pengendalian pada motor DC *Brushless*. Pada sensor jarak ultrasonic HC-SR04 memiliki 4 pin yaitu pin VCC, pin *echo*, pin *trigger* dan pin GND. Pin *echo* dan pin *trigger* dihubungkan pada pin analog Arduino A0, A1, A2, A3, A4, A5 dan pin VCC dihubungkan pada tegangan 5 Volt dan pin GND dihubungkan pada GND pada Arduino. Sensor HC-SR04 membutuhkan power supply sebesar 5 Volt agar dapat bekerja dan arus yang dibutuhkan yaitu sebesar 2 mA, angle yang terbaca oleh sensor ini yaitu 30 derajat serta pembacaan jarak pada sensor yaitu minimal pembacaan 2 cm dan maksimal pembacaan yaitu 500cm dan resolusi pembacaan sensor HC-SR04 yaitu 0.3 cm. Pemilihan pin analog karena saat pembacaan jarak pada sensor memiliki kecepatan pembacaan yang cepat sehingga respon robot menjadi lebih baik.

2. Rangkaian Kontrol Motor DC *Brushless*

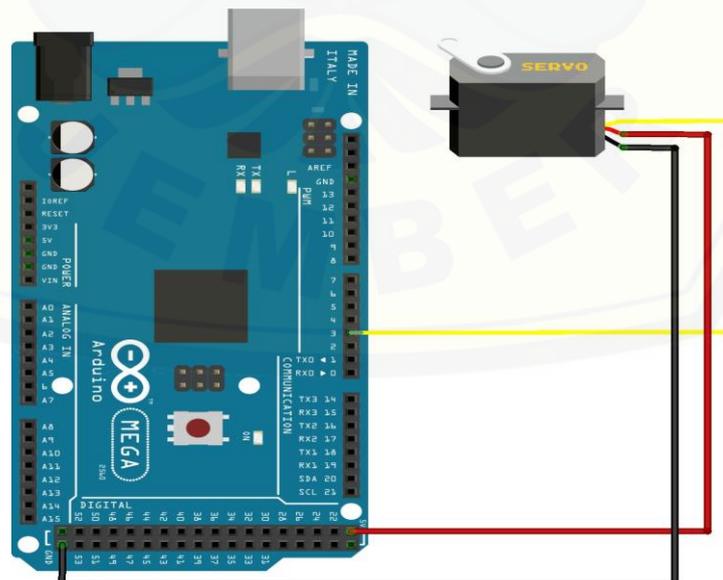
ESC merupakan sebuah rangkaian elektronik yang digunakan untuk mengontrol motor DC *Brushless*. ESC bekerja dengan menerima sinyal PWM yang dikirimkan dari Arduino, sinyal PWM tersebut akan diproses oleh MCU Atmega 8 dan hasil pengolahan tersebut digunakan untuk mengaktifkan *driver per phase output* dari ESC ini memiliki 3 output yaitu U, V, W, sehingga cocok digunakan untuk driver dari motor DC *Brushless*. Penggunaan motor DC *Brushless* pada robot *Hovercraft* dikarenakan pada motor *Brushless* memiliki speed yang tinggi sehingga sangat berguna untuk mendukung robot bergerak. Rangkaian kontrol untuk motor DC *Brushless* seperti pada gambar dibawah ini.



Gambar 3.5 Rangkaian Kontrol Motor DC *Brushless*

Motor DC *Brushless* dikontrol menggunakan Arduino mega 2560, pin yang digunakan yaitu pin digital 4 dan pin digital 5, pin ini merupakan pin PWM sehingga sangat cocok digunakan untuk kontrol motor DC *Brushless*. Pada robot *Hovercraft* digunakan motor DC *Brushless* 2 buah yang akan digunakan sebagai pendorong pada robot *Hovercraft* dan digunakan sebagai *hover/gaya* angkat robot.

3. Rangkaian Kontrol Motor Servo



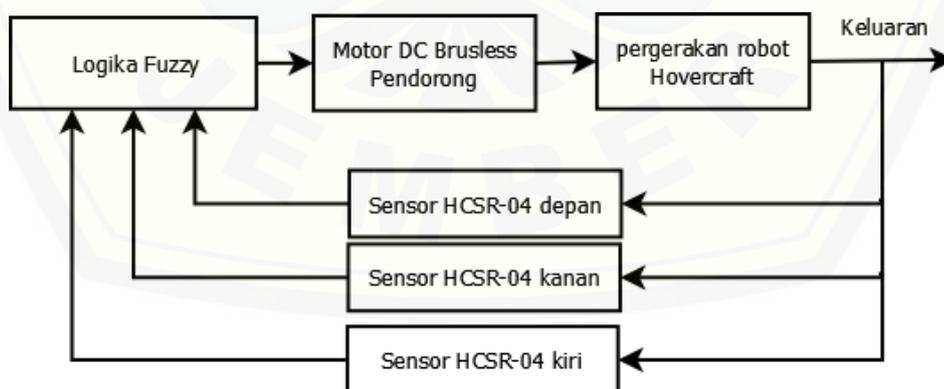
Gambar 3.6 Rangkaian Kontrol Motor Servo

Gambar 3.6 merupakan gambar rangkaian kontrol motor servo, pada rangkaian ini digunakan motor servo sebagai pengendali sirip arah angin untuk menentukan beloknya robot. Pada motor servo ini memiliki 3 pin yaitu pin VCC, pin data, pin GND, pin VCC dihubungkan pada pin 5 Volt dan pin GND dihubungkan pada pin GND, untuk pin data dihubungkan pada pin 3 digital Arduino. Motor servo yang digunakan adalah motor servo MG 996 R dengan berat motor sebesar 55 g yang dapat berputar 180 derajat, yang membutuhkan tegangan 4.8 – 7.2 Volt dan dapat mengangkat beban sebesar 8.5 – 10 kg. dengan spesifikasi tersebut motor servo ini sudah memenuhi kriteria dalam pembuatan robot *Hovercraft*.

3.5 Perancangan *Software*

Perancangan *Software* pada penelitian ini yaitu menggunakan metode logika *fuzzy logic* robot *Hovercraft*, blok diagram aplikasi pengontrolan secara umum dapat dilihat pada gambar 3.7 tentang diagram blok dasar pengendali kecepatan motor pendorong pada robot *Hovercraft*.

3.5.1 Diagram Blok Sistem Pengendali Jarak Dan Kecepatan Motor Pendorong Robot *Hovercraft*



Gambar 3.7 Diagram Blok Pengendali Fuzzy

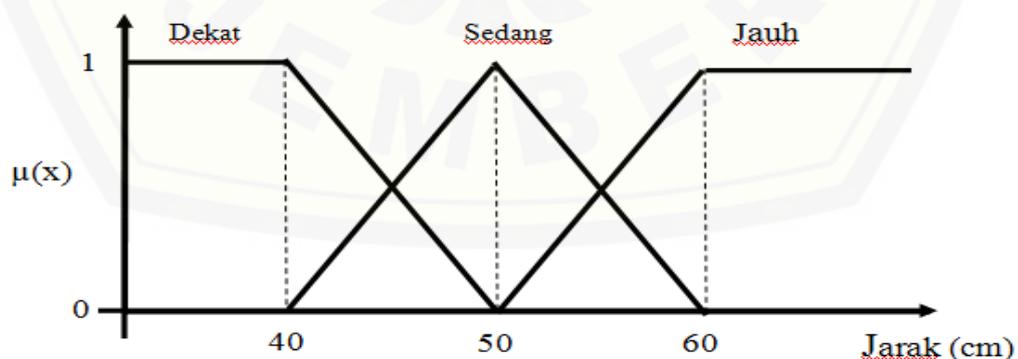
Penjelasan dari gambar 3.7 adalah kontrol *fuzzy* diaplikasikan untuk mengontrol jarak dan kecepatan agar keluaran dari sistem *control* menghasilkan

nilai kecepatan yang konstan sesuai *set point*, masukan dari kontrol *fuzzy* adalah nilai *set point* dari sensor jarak dan kecepatan yang *didapatkan* dari *feedback* keluaran sistem yang di kurangi dengan nilai *set point*, pada gambar 3.5 terdapat kontrol *fuzzy* yang digunakan untuk mengontrol kecepatan motor pendorong yang nantinya akan di *feedback* ke pada sensor jarak.

3.5.2 Perancangan Kontrol *Fuzzy Logic*

Pada sistem pengendalian kecepatan motor pendorong pada robot *Hovercraft* dengan kemudi otomatis terdapat tiga sensor *Ultrasonic HC-SR04* sebagai pengendali kecepatan, untuk metode *fuzzy* yang diterapkan yaitu menggunakan metode sugeno, pada metode ini *didapat* drajat keanggotaan yaitu pada sensor depan terdiri dari DEKAT, SEDANG, JAUH sedangkan pada sensor kanan dan sensor kiri terdiri dari DEKAT, SEDANG dan JAUH, dari drajat keanggotaan tersebut akan didapatkan suatu kondisi yang dilakukan oleh robot, Pada metode logika *fuzzy* yang di fungsikan untuk menentukan kondisi motor pendorong, PELAN (P), SEDANG (S), CEPAT (C) kondisi untuk motor pendorong, dan kondisi untuk motor servo agar robot dapat serong KANAN, serong KIRI.

Fungsi keanggotaan masukan dari sensor HC-SR04 bagian depan, kanan dan kiri diuraikan secara detail pada gambar 3.8 dan gambar 3.9 dibawah ini.



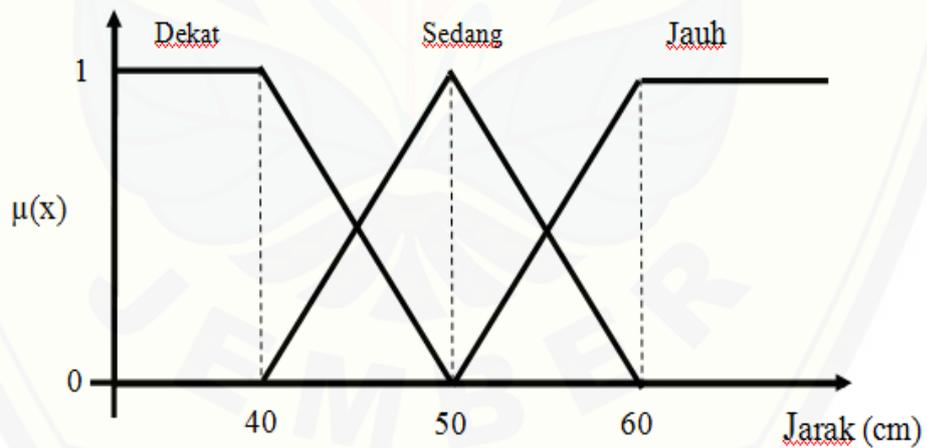
Gambar 3.8 Fungsi Keanggotaan Sensor Depan

fungsi keanggotaan untuk setiap himpunan pada variabel jarak dapat di berikan sebagai berikut:

$$\mu_{dekat} \begin{cases} 1; & jarak \leq 40 \\ \left(\frac{60 - jarak}{60 - 40}\right); & 40 \leq jarak \leq 60 \\ 0; & jarak \geq 60 \end{cases} \dots\dots\dots(23)$$

$$\mu_{sedang} \begin{cases} 0; & jarak \leq 40 \text{ atau } jarak \geq 60 \\ \left(\frac{jarak - 40}{50 - 40}\right); & 40 \leq jarak \leq 50 \\ \left(\frac{60 - jarak}{60 - 50}\right); & 50 \leq jarak \leq 60 \end{cases} \dots\dots\dots(24)$$

$$\mu_{jauh} \begin{cases} 0; & jarak \leq 50 \\ \left(\frac{jarak - 50}{60 - 50}\right); & 50 \leq jarak \leq 60 \\ 1; & jarak \geq 60 \end{cases} \dots\dots\dots(25)$$



Gambar 3.9 Fungsi Keanggotaan Sensor Kanan dan Kiri

fungsi keanggotaan untuk setiap himpunan pada variabel jarak dapat di berikan sebagai berikut:

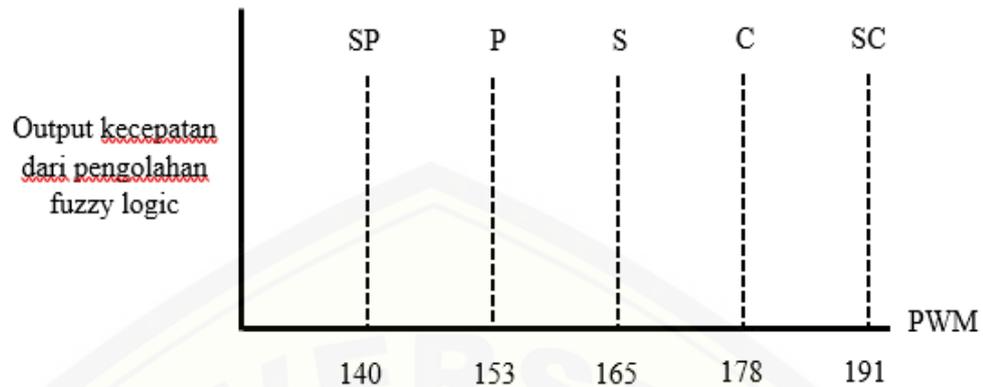
$$\mu_{dekat} \begin{cases} 1; & \text{jarak} \leq 40 \\ \left(\frac{60 - \text{jarak}}{60 - 40}\right); & 40 \leq \text{jarak} \leq 60 \\ 0; & \text{jarak} \geq 60 \end{cases} \dots\dots\dots(26)$$

$$\mu_{sedang} \begin{cases} 0; & \text{jarak} \leq 40 \text{ atau } \text{jarak} \geq 60 \\ \left(\frac{\text{jarak} - 40}{50 - 40}\right); & 40 \leq \text{jarak} \leq 50 \\ \left(\frac{60 - \text{jarak}}{60 - 50}\right); & 50 \leq \text{jarak} \leq 60 \end{cases} \dots\dots\dots(27)$$

$$\mu_{jauh} \begin{cases} 0; & \text{jarak} \leq 50 \\ \left(\frac{\text{jarak} - 50}{60 - 50}\right); & 50 \leq \text{jarak} \leq 60 \\ 1; & \text{jarak} \geq 60 \end{cases} \dots\dots\dots(28)$$

Gambar diatas menunjukkan dua himpunan *fuzzy*, derajat keanggotaan yang menentukan tingkat kesesuaian setiap anggota dengan fungsi keanggotaan sudah di tentukan dalam himpunan *fuzzy* di atas. Aturan dari variabel *linguistic* di atas berbasis jarak dengan menggunakan satuan *centimeter*(Cm).

Dari pengolahan logika *fuzzy* diatas, setelah masukan di proses akan menghasilkan keluaran pada fungsi keanggotaan. Hasil dari pengolahan tersebut berupa nilai kecepatan yang akan di jalankan pada robot *Hovercraft* sebagai kendali kecepatan pada motor pendorong untuk kendali arah robot *Hovercraft* (navigasi). Pada gambar 3.10 akan menggambarkan detail dari keluaran yang merupakan kecepatan dari motor pendorong pada robot *Hovercraft*. Terdapat lima variabel yaitu SP (sangat pelan), P (pelan), S (sedang), C(cepat), dan SC (sangat cepat), dari ke lima variabel ini memiliki nilai PWM yang berbeda-beda seperti pada gambar 3.10 di bawah ini.



Gambar 3.10 *Output Kecepatan Dari Pengolahan Fuzzy logic*

Dari hasil pengolahan masukan dengan proses logika *fuzzy* diatas akan *didapat* aturan, dalam proses tersebut sebagai basis aturan untuk kendali kecepatan motor. Berikut adalah aturan proses logika *fuzzy* yang *diterapkan* pada robot *Hovercraft* sebagai kendali kecepatan motor pendorong.

Tabel 3.2 Fuzzy Rule

Rule	Depan	Kanan	Kiri	Motor
1	Dekat	Dekat	Dekat	SPelan
2	Dekat	Dekat	Sedang	Pelan
3	Dekat	Dekat	Jauh	Sedang
4	Dekat	Sedang	Dekat	Pelan
5	Dekat	Sedang	Sedang	Pelan
6	Dekat	Sedang	Jauh	Sedang
7	Dekat	Jauh	Dekat	Pelan
8	Dekat	Jauh	Sedang	Sedang
9	Dekat	Jauh	Jauh	Sedang
10	Sedang	Dekat	Dekat	Sedang
11	Sedang	Dekat	Sedang	Pelan
12	Sedang	Dekat	Jauh	Sedang

Rule	Depan	Kanan	Kiri	Motor
13	Sedang	Sedang	Dekat	Pelan
14	Sedang	Sedang	Sedang	Pelan
15	Sedang	Sedang	Jauh	Sedang
16	Sedang	Jauh	Dekat	Sedang
17	Sedang	Jauh	Sedang	Sedang
18	Sedang	Jauh	Jauh	Sedang
19	Jauh	Dekat	Dekat	SCepat
20	Jauh	Dekat	Sedang	Sedang
21	Jauh	Dekat	Jauh	Sedang
22	Jauh	Sedang	Dekat	Sedang
23	Jauh	Sedang	Sedang	Sedang
24	Jauh	Sedang	Jauh	Sedang
25	Jauh	Jauh	Dekat	Sedang
26	Jauh	Jauh	Sedang	Sedang
27	Jauh	Jauh	Jauh	SCepat

Aturan dasar (*rule based*) pada kontrol logika fuzzy merupakan suatu bentuk aturan relasi/implikasi “ Jika-Maka” atau “ *If – Then*” seperti pada pernyataan berikut:

IF depan = dekat & kanan = dekat & kiri = dekat THEN motor = SPelan

IF depan = dekat & kanan = dekat & kiri = sedang THEN motor = Pelan

IF depan = dekat & kanan = dekat & kiri = jauh THEN motor = sedang

IF depan = dekat & kanan = sedang & kiri = dekat THEN motor = Pelan

IF depan = dekat & kanan = sedang & kiri = sedang THEN motor = Pelan

IF depan = dekat & kanan = sedang & kiri = jauh THEN motor = sedang

IF depan = dekat & kanan = jauh & kiri = dekat THEN motor = Pelan

IF depan = dekat & kanan = jauh & kiri = sedang THEN motor = sedang

IF depan = dekat & kanan = jauh & kiri = jauh THEN motor = sedang

IF depan = sedang & kanan = dekat & kiri = dekat THEN motor = sedang

IF depan = sedang & kanan = dekat & kiri = sedang THEN motor = Pelan
IF depan = sedang & kanan = dekat & kiri = jauh THEN motor = sedang
IF depan = sedang & kanan = sedang & kiri = dekat THEN motor = pelan
IF depan = sedang & kanan = sedang & kiri = sedang THEN motor = pelan
IF depan = sedang & kanan = sedang & kiri = jauh THEN motor = sedang
IF depan = sedang & kanan = jauh & kiri = dekat THEN motor = sedang
IF depan = sedang & kanan = jauh & kiri = sedang THEN motor = sedang
IF depan = sedang & kanan = jauh & kiri = jauh THEN motor = sedang
IF depan = jauh & kanan = dekat & kiri = dekat THEN motor = SCepat
IF depan = jauh & kanan = dekat & kiri = sedang THEN motor = sedang
IF depan = jauh & kanan = dekat & kiri = jauh THEN motor = sedang
IF depan = jauh & kanan = sedang & kiri = dekat THEN motor = sedang
IF depan = jauh & kanan = sedang & kiri = sedang THEN motor = sedang
IF depan = jauh & kanan = sedang & kiri = jauh THEN motor = sedang
IF depan = jauh & kanan = jauh & kiri = dekat THEN motor = sedang
IF depan = jauh & kanan = jauh & kiri = sedang THEN motor = sedang
IF depan = jauh & kanan = jauh & kiri = jauh THEN motor = SCepat

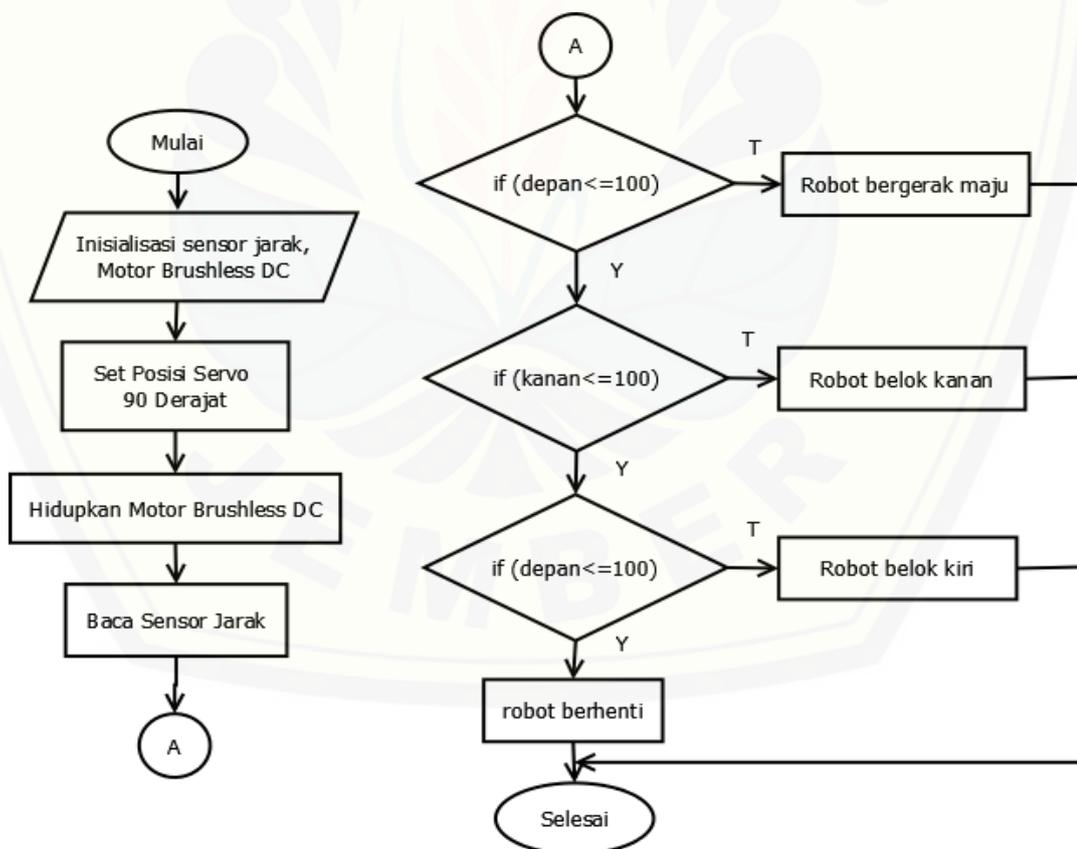
3.5.3 Diagram Alir

Penjelasan pada gambar 3.11 yaitu tentang alur dari sistem kontrol pada robot *Hovercraft* dimana pada diagram alir dijelaskan, pertama mulai kemudian inialisasi *input* atau *output* membaca 3 sensor *Ultrasonic* depan, kanan dan kiri. Setelah 3 sensor *Ultrasonic* terdeteksi maka sensor mengambil data untuk di olah, pada sensor depan memeriksa apakah ada halangan di depan jika tidak data diolah dan memerintah robot untuk bergerak maju, jika sensor depan mendeteksi adanya halangan maka dilanjutkan pada sensor kanan untuk memeriksa apakah ada halangan dibagian kanan, jika tidak ada halangan maka data diolah dan memerintahkan robot untuk bergerak serong ke kanan, jika sensor mendeteksi adanya halangan dibagian kanan maka sensor kiri bekerja memeriksa apakah ada halangan dibagian kiri, jika tidak ada halangan dibagian kiri maka data diolah dan memerintahkan robot untuk bergerak serong ke kiri dan jika sensor kiri membaca

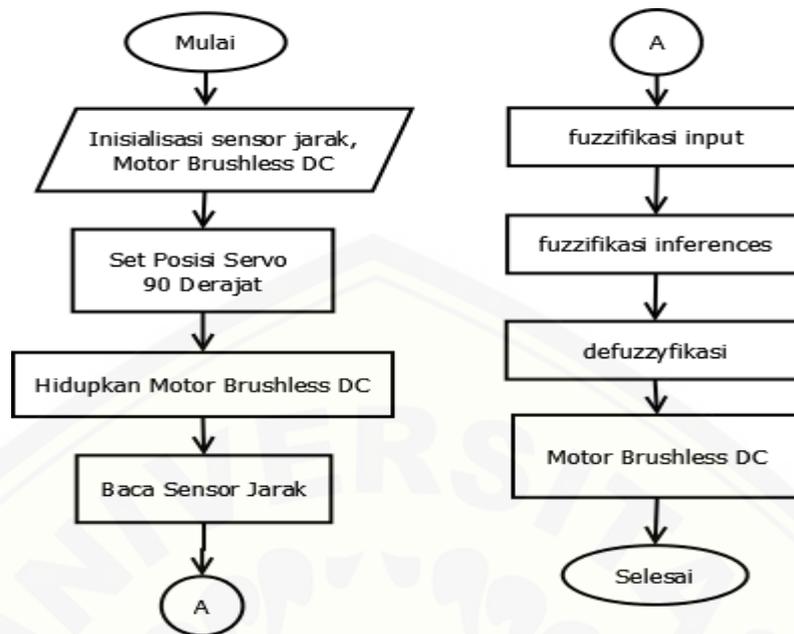
adanya halangan dibagian kiri, atau semua sensor *Ultrasonic* membaca adanya halangan maka data akan dioleh dan memerintahkan robot untuk berhenti.

Pada bagian pengendali kecepatan motor dikontrol menggunakan *fuzzy logic controller* (FLC), menggunakan *setpoint* nilai jarak pada sensor HCSR-04 nilai dari sensor diolah oleh fuzzifikasi *input* hasil dari nilai fuzzifikasi *input* diolah lagi menggunakan fuzzyfikasi inference dan hasil nilai tersebut diolah lagi menggunakan defuzzifikasi dan menghasilkan *output* kecepatan motor *Brushless DC*.

Dalam nilai batas jarak pada sensor terhadap penghalang yaitu sebesar untuk sensor depan sebesar 50 cm batas dekat dan 130 cm batas jauh, untuk sensor kanan dan kiri diberi batasan sebesar 35 cm batas dekat dan 120 cm batas jauh. Sehingga robot akan bergerak tidak sampai menabrak penghalang.



Gambar 3.11 Diagram Alir Navigasi Robot *Hovercraft*



Gambar 3.12 Diagram Alir Motor Pendorong Robot *Hovercraft* dengan *Fuzzy*

3.6 Pengujian

A. pengujian *hardware*

Dalam tahap pengujian *hardware* semua sensor dan *actuator* dikalibrasi terlebih dahulu agar nantinya pada saat pembuatan sistem tidak terjadi kendala yang dapat memperlambat proses pembuatan. Pengujian *hardware* meliputi pengujian sensor HCSR-04, pengujian massa dan kecepatan motor *Brushless hover*, pengujian sudut motor servo, terakhir pengujian motor *Brushless* pendorong. Pengujian terhadap sensor *Ultrasonic* sebagai sensor ukur jarak antara robot dengan dinding, dengan parameter yaitu sebesar 50 cm untuk sensor kanan dan 50 cm untuk sensor kiri dan 150 cm untuk sensor depan. Hasil dari kalibrasi tersebut untuk menentukan posisi titik tengah robot diantara dinding dengan mengatur sudut dari motor servo sebagai pengatur arah angin, sehingga robot tidak sampai menabrak dinding. Selanjutnya menkalibrasi kecepatan motor pendorong robot *Hovercraft* yang dipengaruhi oleh sensor *Ultrasonic* bagian depan, hasil dari pengkalibrasian ini akan menentukan kecepatan laju robot saat akan berbelok ke kanan atau ke kiri.

B. pengujian kontrol sistem

Pengujian sistem dilakukan untuk mengetahui karakteristik *software* pada sistem pengendalian. Pengujian terdiri dari pengujian menggunakan metode *fuzzy logic*.

1. Kontrol kecepatan

Dalam pengujian Kontrol kecepatan setelah melakukan kalibrasi *hardware* maka akan didapatkan nilai kecepatan berupa nilai RPM, sehingga data yang didapat nantinya akan digunakan untuk mendapatkan pendekatan nilai PWM dengan RPM sebenarnya, sehingga dari hasil yang didapat dapat digunakan untuk memproses fuzzyfikasi untuk mengontrol posisi robot dengan mengendalikan motor servo agar robot dapat menghindari dari penghalang.

2. Kontrol posisi robot pada dinding penghalang

Dalam pengujian posisi robot, sensor jarak dan motor servo berperan untuk mengatur jarak antara robot dengan dinding penghalang agar robot selalu berada ditengah-tengah antara dinding penghalang. Hasil dari nilai sensor jarak dapat diproses untuk mengatur sudut motor servo agar robot tetap berada di tengah-tengah diantara dinding penghalang.

C. Pengujian keseluruhan

Pengujian tahap terakhir yaitu mencoba sistem secara keseluruhan, bahwa dalam tahap ini sistem dapat diketahui apakah kontrol sesuai dengan apa yang diharapkan dengan kata lain kecepatan dan jarak sudah dikontrol sesuai dengan *set point*, dalam tahap pengujian keseluruhan dilakukan pengambilan data dari sistem setelah pengujian kecepatan dan jarak telah terpenuhi hasilnya. Dalam pengujian keseluruhan dapat dilihat apakah sistem berjalan seimbang sesuai yang diinginkan atau ada kendala dibagian tertentu, dengan didatakannya data dari hasil pengujian kita dapat menyimpulkan bahwa sistem dalam keadaan normal atau tidak. Pengujian dilakukan pada lintasan yang berbeda yaitu pada lintasan dengan permukaan darat dan lintasan dengan permukaan air yang tenang.

BAB 5. PENUTUP

Dari hasil pengujian pada pengendalian kecepatan motor pendorong robot *Hovercraft* dengan kemudi otomatis berbasis Arduino, maka dapat disimpulkan dan saran yang dapat digunakan untuk pengujian selanjutnya diantaranya yaitu sebagai berikut.

5.1 Kesimpulan

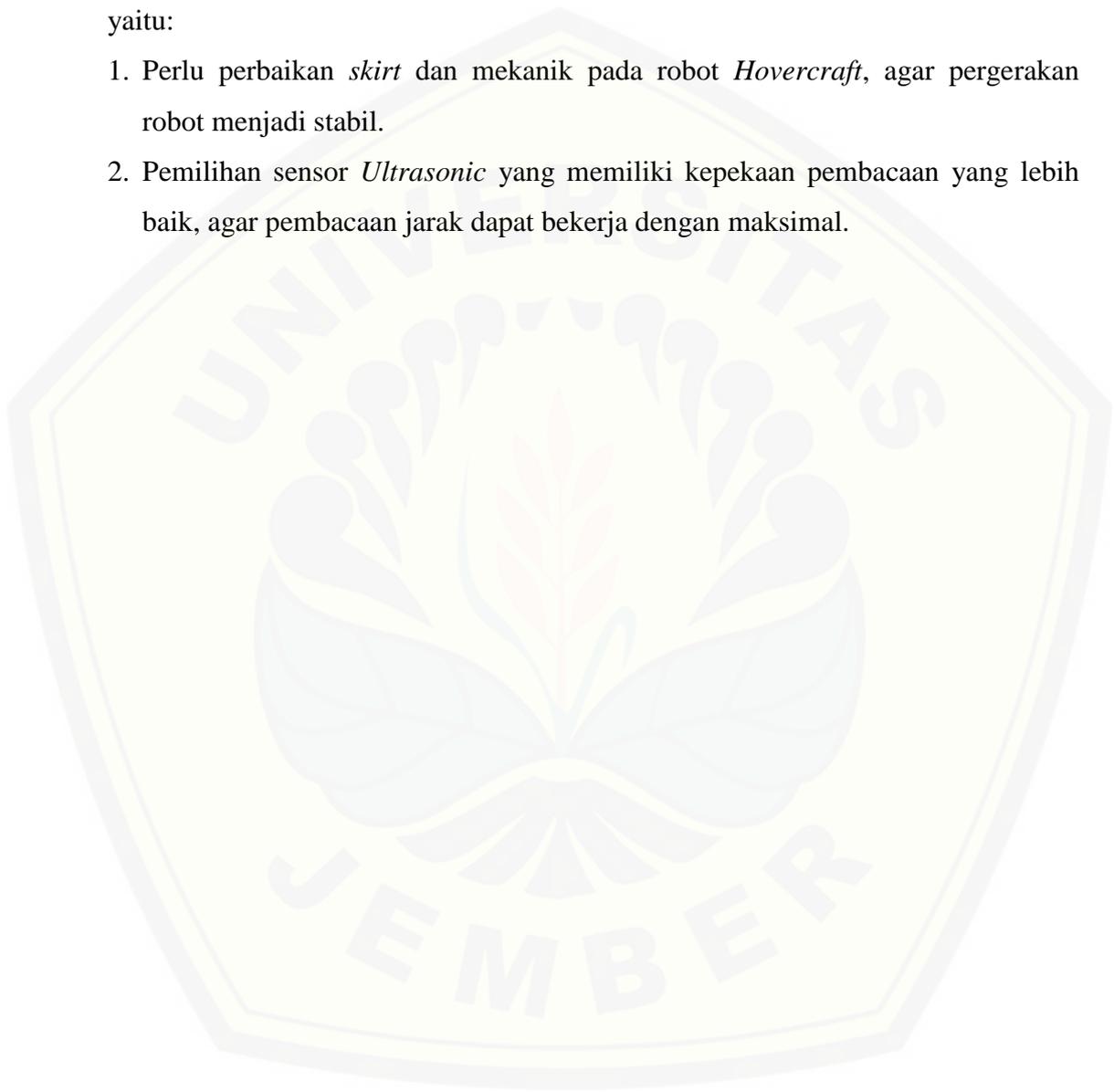
Dari percobaan yang sudah dilakukan pada pengendalian kecepatan motor pendorong robot *Hovercraft* dengan kendali otomatis dapat disimpulkan bahwa:

1. Pada pengujian kecepatan motor *Hover*, semakin besar nilai kecepatan motor *Hover* maka semakin besar gaya angkat pada robot *Hovercraft* hasil ditunjukkan pada tabel 4.2. dari grafik tersebut menunjukkan nilai koefisien kolerasi sebesar 0.946 menunjukkan adanya hubungan linier yang sangat baik antara gaya angkat dengan RPM, maka dapat dikatakan bahwa hubungan antara gaya angkat dengan RPM yaitu sebesar 90%.
2. Dari hasil yang telah didapat bahwa jika menggunakan *fuzzy logic* kecepatan motor pendorong saat berbelok ke kanan atau ke kiri hampir sama, seperti pada tabel 4.6 sehingga pergerakan robot saat berbelok memiliki tingkat keberhasilan yang tinggi, dan jika tidak menggunakan *fuzzy logic* kecepatan motor pendorong saat berbelok ke kanan atau ke kiri nilai kecepatan beda jauh seperti pada tabel 4.5, sehingga pergerakan robot saat berbelok seringkali menabrak dinding.
3. Dari hasil pengujian robot pada lintasan darat dan lintasan air dengan 10 kali pengujian per lintasan, pada lintasan darat keberhasilan robot menelusuri lintasan 60% , dan pada lintasan air keberhasilan robot melnelusuri lintsan 50%, dengan kecepatan rata-rata motor pendorng sebesar 7270.35 RPM pada lintasan darat dan 6429.5 RPM pada lintasan air.
4. Dari hasil pengujian pada lintasan darat dan air, didapatkan nilai *standart deviasi* sebesar 2.250 pada lintasan darat, dan 3.209 pada lintasan air. Semakin kecil nilai *standart deviasi*-nya maka konsistensi waktu tempuh keberhasilan robot semakin besar.

5.2 Saran

Dari hasil penelitian yang telah dilakukan ada beberapa saran untuk dilanjutkannya penelitian tentang rancang bangun sistem pengendalian kecepatan motor pendorong robot *Hovercraft* dengan kemudi otomatis berbasis Arduino yaitu:

1. Perlu perbaikan *skirt* dan mekanik pada robot *Hovercraft*, agar pergerakan robot menjadi stabil.
2. Pemilihan sensor *Ultrasonic* yang memiliki kepekaan pembacaan yang lebih baik, agar pembacaan jarak dapat bekerja dengan maksimal.



DAFTAR PUSTAKA

- Dharmawan, Ade, Skripsi Pengendalian Kecepatan *Brushless* DC Dengan Metode PWM Sinusoidal Menggunakan ATmega16, Universitas Indonesia, Depok, 2009
- Febry Yadi Z, Robot Cerdas Pengangkut Box Dengan Sensor *Ultrasonic* Sebagai Navigasi Berbasis Mikrocontroller, Tugas Akhir (2009).
- Kusumadewi, S., dan Purnomo, H., “Aplikasi Logika Fuzzy untuk Pendukung Keputusan”, Graha Ilmu, Yogyakarta, 2010.
- Munawar, I. and Yahuarsyah, H., *Pemodelan Dan Analisis Sistem Pensaklaran Motor Arus Searah Tanpa Sikat*, 2005. Desertasi S3 Jurusan Teknik Elektro ITB. Bandung.
- Michael Karpelson, John P. Whitney, Gu-Yeon Wei, and Robert J. Wood. Energetics of flapping-wing robotic insects: Towards autonomous hovering flight. In International Conference on Intelligent Robots and Systems, 2010.
- Pitowarno, E., “Robotika : Desain Kontrol dan Kecerdasan Buatan”, Andi Offset, Yogyakarta, 2006.
- Yuliawan, Hendra Dwi, “Perancangan Sistem Thruster Dan Lifter Tipe Terpisah Untuk *Hovercraft* Militer Dengan Payload 15 Ton”, ITS: Surabaya, 2010.
- Prihatama, Erlan Fajar, “Rancang Bangun Pengendali Jarak Jauh Arah Dan Laju Kendaraan *Hover* Dengan 2 Tenaga Dorong”, ITS: Surabaya, 2011.
- Santoso, Hari. 2015. *Panduan Praktis Arduino untuk Pemula*
- L. Steeter Victor, “*Fluid Mechanics*”, McGraw-Hill, United States of America, 1979.
- Yun, Liang, “*Theory And Design Of Air Chusion Craft. Arnold*”, A Member Of The Hodder Headline. Group : London, 2000.

LAMPIRAN

A. Program Arduino

```
#include <Servo.h>
#include <NewPing.h>
Servo myservo;
Servo ESC1;
Servo ESC2;
#define TRIGGER_PINL A0 // Arduino pin tied to trigger pin on ping
sensor.
#define ECHO_PINL A3 // Arduino pin tied to echo pin on ping sensor.
#define MAX_DISTANCE 500 // Maximum distance we want to ping for (in
centimeters). Maximum sensor distance is rated at 400-500cm.
#define TRIGGER_PINF A4 // Arduino pin tied to trigger pin on ping
sensor.
#define ECHO_PINF A5 // Arduino pin tied to echo pin on ping sensor.
#define TRIGGER_PINR A1 // Arduino pin tied to trigger pin on ping
sensor.
#define ECHO_PINR A2 // Arduino pin tied to echo pin on ping sensor.
float jarak1, jarak2, jarak3;
float oldLeftSensor, oldRightSensor, leftSensor, rightSensor, frontSensor,
oldFrontSensor, lSensor, rSensor, fSensor;
int maksampling = 50;
int makdepan = 50;
int inmsg = 0;
float speed;
float speed2;
float jarak;
float defuz, pwm;
//Rule Base
float depan [3];
```

```
float kanan [3];
float kiri [3];
float rule [3][3][3];
float rule000, rule001, rule002, rule010, rule011, rule012, rule020, rule021,
rule022;
float rule100, rule101, rule102, rule110, rule111, rule112, rule120, rule121,
rule122;
float rule200, rule201, rule202, rule210, rule211, rule212, rule220, rule221,
rule222;
float value=0;
float rev=0;
int rpm;
int oldtime=0;
int time;
int a = 100;
int kode=0;
void isr() //interrupt service routine
{
  rev++;
}
void arm1(){
  setkecepatan(0); //Sets speed variable
  delay(1000);
}
void arm2(){
  setkecepatan2(0); //Sets speed variable
  delay(1000);
}
void setkecepatan (int speed){
  int angle = map(speed , 0, 100, 0, 180); //Sets servo positions to different
speeds
```

```
ESC1.write(angle);
}
void setkecepatan2 (int speed2){
  int angle2 = map(speed2 , 0, 100, 0, 180); //Sets servo positions to different
speeds
  ESC2.write(angle2);
}
NewPing sonarLeft(TRIGGER_PINL, ECHO_PINL, MAX_DISTANCE); //
NewPing setup of pins and maximum distance.
NewPing sonarRight(TRIGGER_PINR, ECHO_PINR, MAX_DISTANCE);
NewPing sonarFront(TRIGGER_PINF, ECHO_PINF, MAX_DISTANCE);
unsigned int pingSpeed = 30; // How frequently are we going to send out a
ping (in milliseconds). 50ms would be 20 times a second.
unsigned long pingTimer; // Holds the next ping time.
void setup(){
  Serial.begin(9600);
  attachInterrupt(0,isr,RISING);
  lcd.begin(16,2);
  myservo.attach(3);
  myservo.write(100);
  delay(50);
  ESC1.attach(4);//Adds ESC to certain pin.
  ESC2.attach(5);//Adds ESC to certain pin.
  arm1();
  arm2();
  for(speed = 0,speed2=0; speed <= 60,speed2 <= 60; speed += 5, speed2 +=
5) { //Cycles speed up to 70% power for 1 second
  setkecepatan (speed);//Creates variable for speed to be used in in for loop
  setkecepatan2 (speed2);
  Serial.print("pwm: ");
  Serial.print(speed);
```

```
Serial.print(" pwm2: ");
Serial.println(speed2);
delay(1000);
}
speed=60;
speed2=56;
setkecepatan (speed);
setkecepatan2 (speed2);
Serial.print("pwm: ");
Serial.print(speed);
Serial.print(" pwm2: ");
Serial.println(speed2);
}
void loop(){
  sensorjarak();
  if(jarak1>=110 && jarak3<=130){
    if(kode==0)
    {
      speed = 45;
      speed2 = 50;
      a=100;
      setkecepatan(speed);
      setkecepatan2(speed2);
      myservo.write(a);
      delay(500);
      sensorjarak();
      if(jarak1<60&&jarak3<=120)goto xy;
      a = 80;
      myservo.write(a);
      for(int x = 0; x <= 15; x+=2){
        speed = 45 + x;
```

```
    setkecepatan(speed);
    speed2 = 45 + x;
    setkecepatan2(speed2);
    delay(100);
  }
}
sensorjarak();
if(jarak1<60&&jarak3<=120)goto xy;
speed = 60;
setkecepatan(speed);
speed2 = 57;
setkecepatan2(speed2);
for(int k = 0; k< 5; k++){
  delay(500);
}
xy: kode=1;
}
else if(jarak1<=35){
  kode=0;
  a= 125;
  myservo.write(a);
  if(speed < 60)speed++;
  if(speed2<55)speed2++;
  setkecepatan(speed);
  setkecepatan2(speed2);
}
else if(jarak2<=30){
  a= 90;
  myservo.write(a);
  if(speed < 60)speed+=2;
  if(speed2<55)speed2+=2;
```

```
    setkecepatan(speed);
    setkecepatan2(speed2);
}
else{
    if(kode==1){
        speed = 45;
        speed2 = 50;
        a=100;
        setkecepatan(speed);
        setkecepatan2(speed2);
        myservo.write(a);
        delay(500);
    }
    if(speed < 60)speed+=2;
    if(speed2<55)speed2+=2;
    a=100;
    setkecepatan(speed);
    setkecepatan2(speed2);
    myservo.write(a);
    delay(200);
    kode=0;
}
Fuzzydepan();
Fuzzykanan();
Fuzzykiri();
    Serial.print("kanan: ");
    Serial.print(jarak1);
    Serial.print(" # kiri: ");
    Serial.print(jarak2);
    Serial.print(" # depan: ");
    Serial.print(jarak3);
```

```
Serial.print(" # speed: ");
Serial.println(speed2);
// sensorrpm();
}
void sensorjarak(){
  lSensor = sonarLeft.ping_cm(); //ping in cm
  rSensor = sonarRight.ping_cm();
  fSensor = sonarFront.ping_cm();
  jarak2 = (lSensor + oldLeftSensor) / 2; //average distance between old &
new readings to make the change smoother
  jarak1 = (rSensor + oldRightSensor) / 2;
  jarak3 = (fSensor + oldFrontSensor) / 2;
  oldLeftSensor = jarak2; // save old readings for movment
  oldRightSensor = jarak1;
  oldFrontSensor = jarak3;
}
void Fuzzydepan(){
  // untuk kondisi dekat
  if (jarak3 <= 30)
  { depan [0] = 1;}
  else if (jarak3 > 30 && jarak3 <= 40)
  { depan [0] = (40 - jarak3)/(40 - 30); }
  else
  { depan [0] = 0;}
  // untuk kondisi sedang
  if (jarak3 <= 30)
  { depan [1] = 0;}
  else if (jarak3 > 30 && jarak3 <= 40)
  { depan [1] = (jarak3 -30)/(40-30);}
  else if (jarak3 > 40 && jarak3 <= 50)
  { depan [1] = (50-jarak3)/(50 - 40);}
```

```
else
{ depan [1] = 0;}

// untuk kondisi jauh
if (jarak3 <= 50)
{ depan [2] = 0;}
else if (jarak3 > 50 && jarak3 <= 50)
{ depan [2] = (jarak3-50)/(50-40);}
else if (jarak3 > 50)
{ depan [2] = 1;}
}

void Fuzzykanan(){
// untuk kondisi dekat
if (jarak1 <= 30)
{ kanan [0] = 1;}
else if (jarak1 > 30 && jarak1 <= 40)
{ kanan [0] = (40 - jarak1)/(40 - 30); }
else
{ kanan [0] = 0;}

// untuk kondisi sedang
if (jarak1 <= 30)
{ kanan [1] = 0;}
else if (jarak1 > 30 && jarak1 <= 40)
{ kanan [1] = (jarak1 - 30)/(40 - 30);}
else if (jarak1 > 30 && jarak1 <= 35)
{ kanan [1] = (50 - jarak1)/(50 - 40);}
else
{ kanan [1] = 0;}
```

```
// untuk kondisi jauh
if (jarak1 <= 40)
{ kanan [2] = 0;}
else if (jarak1 > 40 && jarak1 <= 50)
{ kanan [2] = (jarak1-40)/(50-40);}
else if (jarak1 > 50)
{ kanan [2] = 1;}
}
void Fuzzykiri(){
// untuk kondisi dekat
if (jarak2 <= 30)
{ kiri [0] = 1;}
else if (jarak2 > 30 && jarak2 <= 40)
{ kiri [0] = (40 - jarak2)/(40 - 30); }
else
{ kiri [0] = 0;}
// untuk kondisi sedang
if (jarak2 <= 30)
{ kiri [1] = 0;}
else if (jarak2 > 30 && jarak2 <= 30)
{ kiri [1] = (jarak2 - 30)/(40 - 30);}
else if (jarak2 > 30 && jarak2 <= 50)
{ kiri [1] = (50 - jarak2)/(50 - 40);}
else
{ kiri [1] = 0;}
// untuk kondisi jauh
if (jarak2 <= 40)
{ kiri [2] = 0;}
else if (jarak2 > 40 && jarak2 <= 50)
{ kiri [2] = (jarak2 - 40)/(50 - 40);}
else if (jarak2 > 50)
```

```
{ kiri [2] = 1;}
}

void RuleEva (){
int i, j, k;
for ( i=0; i<=2; i=i+1)
{
for ( j=0; j<=2; j=j+1)
{
for ( k=0; k<=2; k=k+1)
{
jarak = min(depan[i], kanan[j]);
jarak = min(depan[i], kiri[k]);
// jarak = min(kanan[j], kiri[k]);
rule [i][j][k] = jarak;
}
}}
rule000 = rule [0][0][0]; // (dekat,dekat,dekat = Lambat)
rule001 = rule [0][0][1]; // (dekat,dekat,sedang = Lambat)
rule002 = rule [0][0][2]; // (dekat,dekat,jauh = Lambat)
rule010 = rule [0][1][0]; // (dekat,sedang,dekat = lambat)
rule011 = rule [0][1][1]; // (dekat,sedang,sedang = lambat)
rule012 = rule [0][1][2]; // (dekat,sedang,jauh = lambat)
rule020 = rule [0][2][0];
rule021 = rule [0][2][1];
rule022 = rule [0][2][2];
rule100 = rule [1][0][0];
rule101 = rule [1][0][1];
rule102 = rule [1][0][2];
rule110 = rule [1][1][0];
rule111 = rule [1][1][1];
```

```
rule112 = rule [1][1][2];
rule120 = rule [1][2][0];
rule121 = rule [1][2][1];
rule122 = rule [1][2][2];
rule200 = rule [2][0][0];
rule201 = rule [2][0][1];
rule202 = rule [2][0][2];
rule210 = rule [2][1][0];
rule211 = rule [2][1][1];
rule212 = rule [2][1][2];
rule220 = rule [2][2][0];
rule221 = rule [2][2][1];
rule222 = rule [2][2][2];
}
void Defuzzy () {
    // metode sugeno (weighted average)
    float slambat = 53;
    float lambat = 55;
    float sedang = 58;
    float cepat = 60;
    float scepat = 63;
    RuleEva();
    pwm = (rule000 * slambat) + (rule001 * lambat)+ (rule002 *sedang)+
(rule010 * lambat)+ (rule011 * lambat)+ (rule012 * sedang) + (rule020 *
lambat)+ (rule021 * sedang)+ (rule022 * sedang)+ (rule100 * sedang) +
(rule101 * lambat)+ (rule102 *sedang)+ (rule110 * lambat)+ (rule111 *
lambat)+ (rule112 * sedang) + (rule120 * sedang)+ (rule121 * sedang)+
(rule122 * sedang)+ (rule200 * scepat) + (rule201 * cepat)+ (rule202
*sedang)+ (rule210 * sedang)+ (rule211 * sedang)+ (rule212 * sedang) +
(rule220 * sedang)+ (rule221 * sedang)+ (rule222 * scepat) ;
    defuz = 0;
```

```
int i, j, k;
for ( i=0; i<=2; i=i+1)
{
  for ( j=0; j<=2; j=j+1)
  {
    for (k=0; k<=2; k=k+1)
    {
      defuz = defuz + rule [i][j][k];
    }
  }
  pwm = pwm / defuz;
}
/* void sensorrpm(){
delay(100);
detachInterrupt(0);      //detaches the interrupt
time=millis()-oldtime;   //finds the time
rpm=(rev/time)*60000;    //calculates rpm
oldtime=millis();        //saves the current time
rev=0;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("__SPEED MOTOR__");
lcd.setCursor(0,1);
lcd.print(  rpm);
lcd.print(" RPM");
lcd.print(" ");
attachInterrupt(0,isr,RISING);
}*/
```

B. Dokumentasi Penelitian

