



**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA  
MENGUNAKAN METODE *BEHAVIOR BASED* DAN KONTROL  
PID PADA MANUVER ROBOT PEMADAM API**

**SKRIPSI**

Oleh

**Drajad Dwi Kurniawan**

**NIM 141910201033**

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

## PERSEMBAHAN

Alhamdulillah, puji syukur atas kehadiran Allah SWT karena atas limpahan rahmat, karunia, hidayah, dan kasih sayang-Nya saya dapat menyelesaikan penelitian ini dengan baik. Akhirnya saya persembahkan penelitian ini kepada :

1. Allah SWT yang Maha Segalanya.
2. Junjungan kita Nabi Muhammad SAW sebagai penutan kita dalam segala hal.
3. Orang Tua, yang telah memberikan kesempatan, dukungan serta semangat dalam kehidupan ini.
4. Dosen Teknik Elektro Universitas Jember yang telah memberikan ilmu yang sangat bermanfaat.
5. Dulur Ketek UJ, dulur yang memberikan pelajaran lain dalam hidup, menumbuhkan kesadaran sosial, serta terimakasih telah merangkul saya dengan cara yang berbeda.
6. Keluarga Besar UKM Robotika Teknik, terimakasih telah mendidik dan mengajarkan tentang kesetiaan, loyalitas serta tanggung jawab.

**MOTTO**

*Aal izz Well (Punchuk Wangdu)*

*Permudahlah, jangan mempersulit. Gembirakanlah, jangan menakut-nakuti*  
(Mutafaq 'Ilaih)

*Never forget who was there for you when no one else*

(Firdausin Nuzula)

*Hiduplah dengan caramu sendiri atau orang lain akan menghidupkanmu*  
*dengan cara mereka*

(Drajad Kurniawan)

**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Drajad Dwi Kurniawan

NIM : 141910201033

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode *Behavior Based* Kontrol PID Pada Manuver Robot Pemadam Api" 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 dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 17 July 2018

Yang menyatakan,

Drajad Dwi Kurniawan

NIM 141910201033

**SKRIPSI**

**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA  
MENGUNAKAN METODE *BEHAVIOR BASED* DAN KONTROL  
PID PADA MANUVER ROBOT PEMADAM API**

Oleh

Drajad Dwi Kurniawan

NIM 141910201033

Pembimbing :

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

Dosen Pembimbing Anggota : Widya Cahyadi, S.T., M.T

**PENGESAHAN**

Skripsi berjudul **“Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode *Behavior Based* Kontrol PID Pada Manuver Robot Pemadam Api”** karya Drajad Dwi Kurniawan telah diuji dan disahkan pada :

Hari :

Tanggal :

Tempat : Fakultas Teknik Universitas Jember

Ketua,

Anggota I,

Sumardi S.T.,M.T

NIP 196701131998021001

Widya Cahyadi S.T.,M.T

NIP 198511102014041001

Anggota II,

Anggota III,

Ike Fibriani S.T.,M.T

NIP 198002072015042001

Guido Dias Kalandro S.T.,M.Eng

NIP 760015734

Mengesahkan

Dekan

Dr. Ir. Entin Hidayah M.U.M

NIP 196612151995032001

## RINGKASAN

**Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode *Behavior Based* Kontrol PID Pada Manuver Robot Pemadam Api;** Drajad Dwi Kurniawan; 141910201033; 2018; 86 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (kecerdasan buatan). Salah satu pekerjaan manusia yang dapat dilakukan oleh robot adalah melakukan pemadaman kebakaran. Pada penelitian ini robot akan menggunakan satu buah sensor flame yang akan digunakan untuk mendeteksi api dan lima buah sensor Ultrasonik HC-SR04 yang akan digunakan untuk mendeteksi jarak agar robot dapat bernavigasi dengan baik tanpa menabrak dinding lintasan. Pada robot ini juga akan menerapkan metode *behavior based control* dan kontrol PID yang diharapkan akan dapat membantu performa robot dalam menyelesaikan misi.

Sistem kontrol *Behavior Based Robot* adalah sebuah sistem kontrol yang dirancang dengan memberikan perilaku kepada robot. Sistem kontrol ini tidak terpaku kepada lingkungan yang statis sehingga robot yang dirancang menggunakan metode *behavior based* akan mampu beradaptasi terhadap lingkungannya. Pada perancangan sistem kontrol *behavior based* ini robot akan diberikan empat buah perilaku yaitu menyisir dinding, mengenali posisi start, mendeteksi api dan memadamkan api, dimana masing-masing perilaku akan dipegang oleh satu buah sensor yang akan memberikan aksi kepada aktuator. Sedangkan pada penelitian ini kontrol PID digunakan untuk memperhalus pergerakan robot dalam bermanuver didalam ruangan sehingga akan mempercepat waktu robot dalam menyelesaikan misi yang diberikan.



Pada penelitian ini robot sudah mampu menyelesaikan misi yang diberikan dengan baik, dimana dari semua perilaku yang diberikan, robot sudah dapat menerapkannya dengan baik. Dimana dari keempat perilaku yang diberikan persentase keberhasilan secara berturut-turut adalah 100% pada perilaku menyisir dinding, 80% pada perilaku mengenali posisi strat, 70% pada perilaku mendeteksi api dan 80% pada perilaku memadamkan api. Dan kontrol PID yang digunakan memiliki respon yang cukup baik untuk membantu pergerakan robot saat menggunakan nilai  $K_p$  18,  $K_i$  7 dan  $K_d$  8 dengan nilai *rise time* 18 detik dan *setting time* 72 detik.

Pada pengujian keseluruhannya dari lima kali percobaan dengan empat kondisi yang berbeda dapat diketahui persentase keberhasilan robot secara berturut-turut adalah pada pengujian pertama 100% dengan rata-rata waktu 72,8 detik, pada pengujian kedua 60% dengan rata-rata waktu 81,6 detik, pada pengujian ketiga 80% dengan rata-rata waktu 83,25 detik, dan pada pengujian yang terakhir 80% dengan rata-rata waktu 86,75 detik. Sesuai dengan hasil pengujian yang dilakukan diketahui bahwa intensitas cahaya yang diterima oleh sensor flame akan sangat mempengaruhi hasil pembacaan nilai ADC nya sehingga pengujian dilakukan pada waktu malam hari saat intensitas cahaya rendah agar robot dapat bekerja dengan maksimal.



## PRAKATA

Puji syukur kehadirat Allah SWT yang Maha Kuasa, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul “*Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode Behavior Based Kontrol PID Pada Manuver Robot Pemadam Api*”. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak sehingga skripsi ini dapat diselesaikan.

Untuk itu penulis mengucapkan terimakasih kepada.

1. Allah SWT yang telah melimpahkan rezeki, rahmat, hidayah dan karunia serta kasih sayang-Nya sehingga penulis mampu menyelesaikan skripsi ini.
2. Nabi besar Muhammad SAW, yang telah menjadi suri tauladan seluruh umat.
3. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku Dekan Fakultas Teknik Universitas Jember
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember.
5. Bapak Sumardi ST., MT dan Widya Cahyadi, S.T.,M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini.
6. Bapak Ike Fibriani, S.T., M.T. dan Guido Dias Kalandro, S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini.
7. Keluarga besar Civitas Akademia Jurusan Teknik Elektro Universitas Jember. Serta semua pihak yang telah mendukung dalam penyelesaian skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 18 July 2018

Penulis

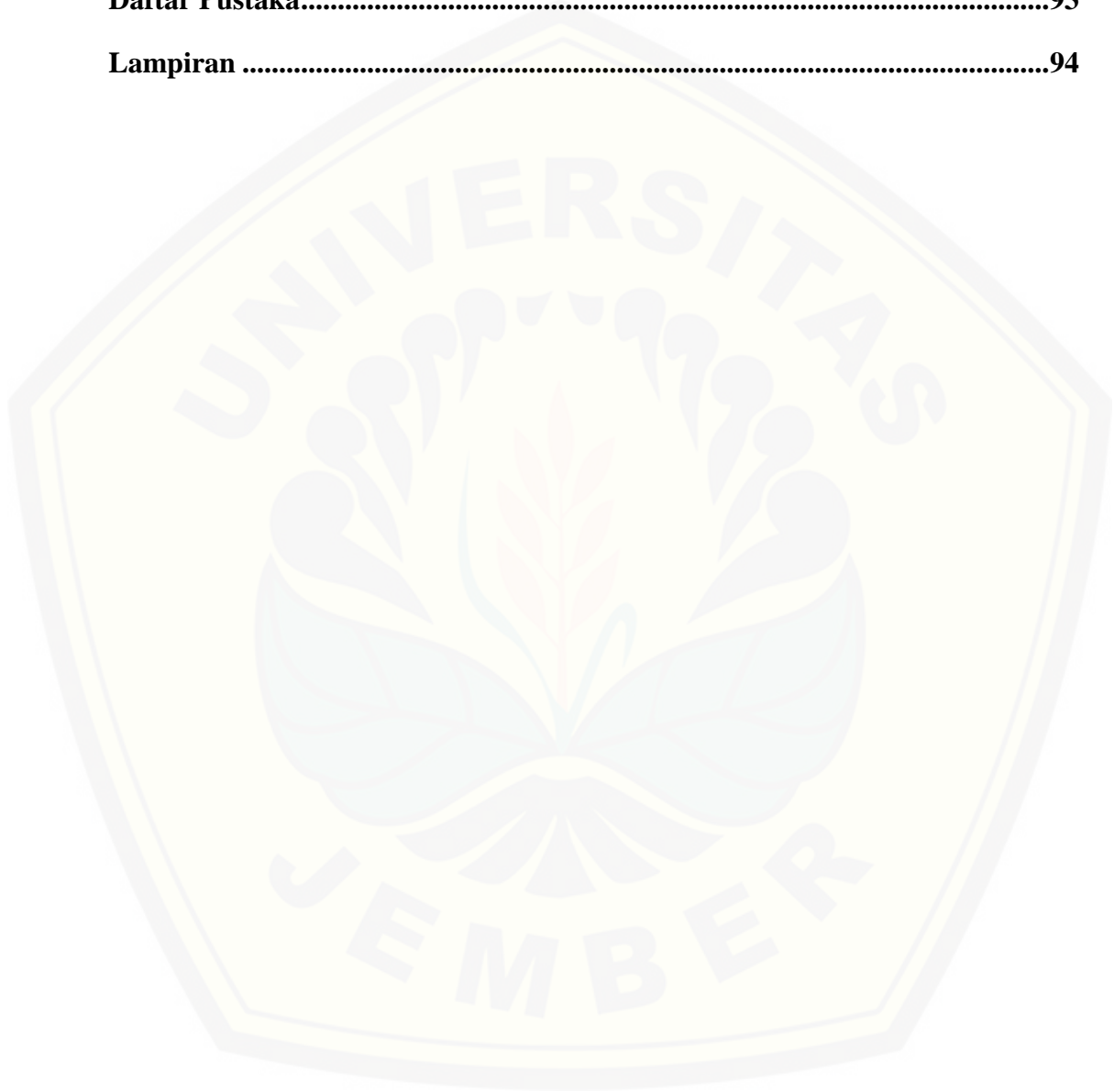
DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL .....	xvi
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang Masalah.....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>3</b>
<b>1.3 Batasan Masalah.....</b>	<b>3</b>
<b>1.4 Tujuan Penelitian .....</b>	<b>3</b>
<b>1.5 Manfaat Penelitian .....</b>	<b>4</b>
<b>BAB 2. TINJAUAN PUSTAKA.....</b>	<b>5</b>
<b>2.1. Penelitian Terkait .....</b>	<b>5</b>
<b>2.2. Arduino Mega .....</b>	<b>5</b>
<b>2.3. Sensor Ultrasonik HC-SR04.....</b>	<b>7</b>
<b>2.4. Sensor Photodiode .....</b>	<b>9</b>
<b>2.5. Sensor Api Flame.....</b>	<b>10</b>
<b>2.6. Motor DC.....</b>	<b>11</b>
<b>2.7. Driver Motor H-Bridge Mosfet .....</b>	<b>12</b>
<b>2.8. Behaviour Based .....</b>	<b>13</b>
2.8.1 Koordinasi <i>Behavior Based</i> .....	13
2.8.2 Arsitektur <i>Subsumption</i> .....	14
<b>2.9. Kontrol PID.....</b>	<b>15</b>
2.9.1 Kontrol <i>Proportional</i> (P) .....	15
2.9.2 Kontrol <i>Integrative</i> (I).....	16
2.9.3 Kontrol <i>Derivatif</i> ve .....	17
2.9.4 Aksi kontroler <i>Proportional + Integrative + Derivative</i> .....	18

<b>BAB 3. METODOLOGI PENELITIAN .....</b>	<b>20</b>
<b>3.1. Tempat dan Waktu Penelitian .....</b>	<b>20</b>
3.1.1 Tempat Penelitian.....	20
3.1.2 Waktu Penelitian .....	20
<b>3.2. Alat dan Bahan .....</b>	<b>21</b>
3.2.1. Perangkat Lunak ( <i>Software</i> ) .....	21
3.2.2. Perangkat Keras ( <i>Hardware</i> ) .....	21
<b>3.3. Tahap Penelitian .....</b>	<b>22</b>
<b>3.4. Blok Sistem.....</b>	<b>23</b>
<b>3.5. Perancangan Desain Alat.....</b>	<b>24</b>
3.5.1. Perancangan Desain Robot .....	24
3.5.2. Perancangan Desain Sensor Ultrasonik dengan Arduino Mega ...	26
3.5.3. Perancangan Desain Sensor Photodiode dengan Arduino Mega ..	27
3.5.4. Perancangan Desain Sensor Flame dengan Arduino Mega .....	28
3.5.5. Perancangan Desain Driver Motor H-Bridge Mosfet .....	31
3.5.6. Perancangan Sistem Kontrol <i>Behaviour Based</i> .....	32
3.5. Perancangan Sistem Kontrol PID.....	34
<b>3.6 Diagram Alir (<i>Flowchart</i>).....</b>	<b>37</b>
3.6.1. Diagram Alir Gerak Robot.....	37
3.6.2. Diagram Alir Sistem Robot secara keseluruhan .....	39
<b>3.7. Pengujian Data.....</b>	<b>40</b>
3.7.1 Pengujian sensor Ultrasonik HC-SR04.....	40
3.7.2 Pengujian sensor Flame .....	40
3.7.3 Pengujian sensor Photodiode .....	41
3.7.4 Pengujian sistem kontrol <i>Behavior Based Robot</i> .....	41
3.7.5 Pengujian sistem kontrol PID .....	42
3.7.6 Pengujian robot secara keseluruhan .....	42
<b>3.8 Hasil perancangan desain robot.....</b>	<b>43</b>

<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>	<b>46</b>
<b>4.1 Pengujian Sensor Ultrasonik HC-SR04.....</b>	<b>46</b>
<b>4.2 Pengujian Sensor Photodiode.....</b>	<b>53</b>
<b>4.3 Pengujian Sensor Flame.....</b>	<b>60</b>
<b>4.4 Pengujian Kontrol PID .....</b>	<b>64</b>
<b>4.4.1 Metode <i>Trial and Error</i> .....</b>	<b>65</b>
4.4.1.1 Pengujian parameter Proporsional (Kp).....	65
4.4.1.2 Pengujian parameter proporsional (Kp) dan derivative (PD) .....	67
4.4.1.3 Pengujian parameter proporsional (Kp), integrative (Ki) dan derivative(Kd) .....	70
4.4.1.4 Perbandingan Parameter Kontrol PID.....	72
<b>4.4.2 Metode Zieger-Nichole.....</b>	<b>73</b>
<b>4.4.3 Perbandingan metode <i>Trial and Error</i> dengan         <i>Zieger-Nichole</i> .....</b>	<b>75</b>
<b>4.5 Pengujian <i>Behavior Based Robot</i> .....</b>	<b>77</b>
4.5.1 Pengujian <i>behavior</i> menyisir dinding .....	80
4.5.2 Pengujian <i>behavior</i> mengenali posisi <i>start (home)</i> .....	81
4.5.3 Pengujian <i>behavior</i> mendeteksi api .....	82
4.5.4 Pengujian <i>behavior</i> memadamkan api .....	84
<b>4.6 Pengujian Keseluruhan.....</b>	<b>85</b>
4.6.1 Pengujian Pertama.....	86
4.6.2 Pengujian Kedua .....	87
4.6.3 Pengujian Ketiga .....	88
4.6.4 Pengujian Keempat .....	89

<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>91</b>
<b>5.1 Kesimpulan.....</b>	<b>91</b>
<b>5.2 Saran.....</b>	<b>92</b>
<b>Daftar Pustaka.....</b>	<b>93</b>
<b>Lampiran .....</b>	<b>94</b>



**DAFTAR GAMBAR**

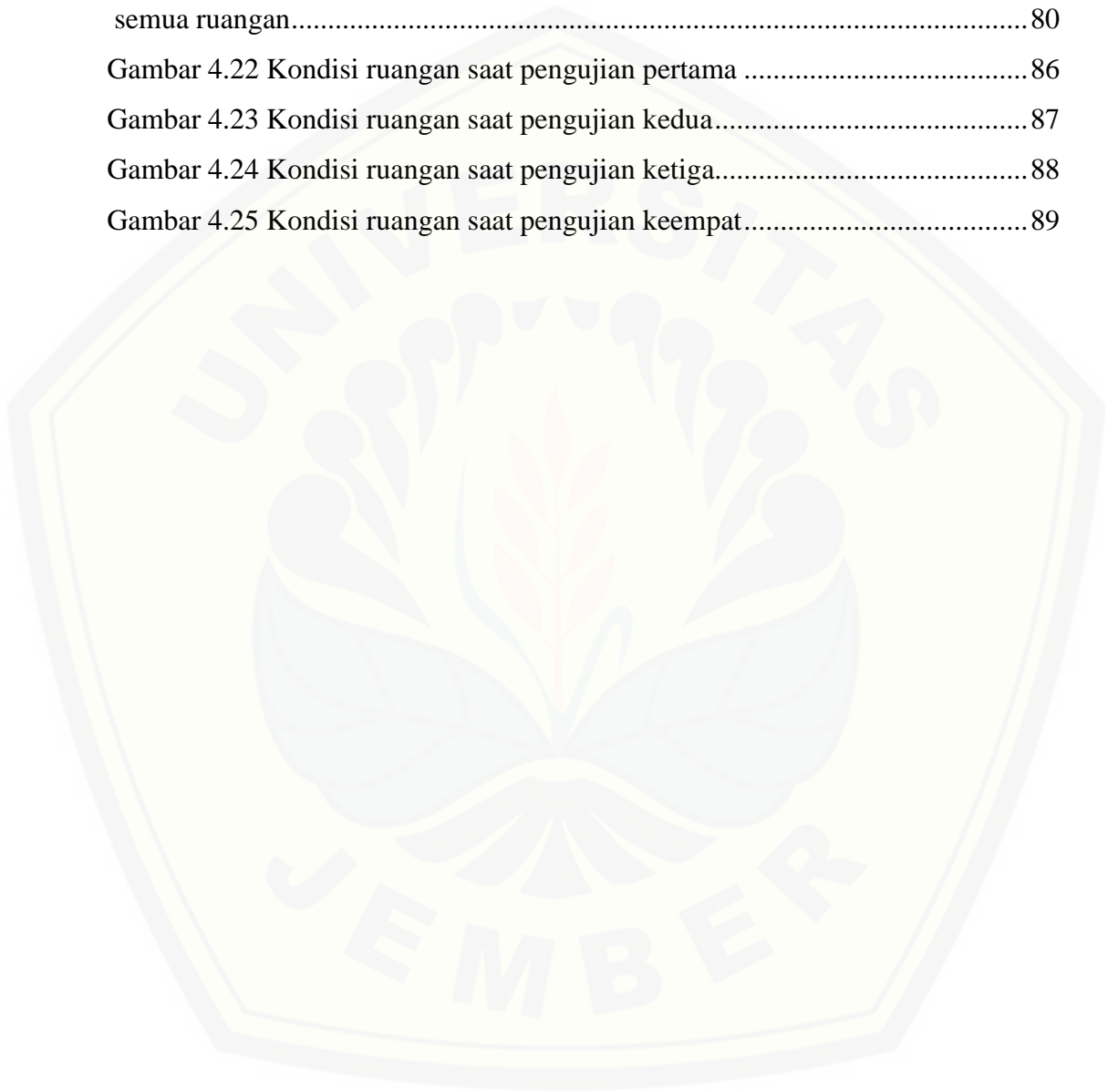
Gambar 2.1 Bentuk Fisik Arduino Mega 25605.....	6
Gambar 2.2 Bentuk Fisik Sensor Ultrasonic HC-SR04.....	7
Gambar 2.3 Cara Kerja Sensor Ultrasonic Dengan Transmister dan Reciver .....	8
Gambar 2.4 Simbol dan Bentuk Fisik Photodioda.....	10
Gambar 2.5 Bentuk Fisik Sensor Flame .....	11
Gambar 2.6 Bentuk Fisik Gearbox Motor DC .....	12
Gambar 2.7 Rangkaian Driver Motor <i>H-Bridge</i> Mosfet.....	13
Gambar 2.8 Koordinasi Kompetitif .....	14
Gambar 2.9 Koordinasi Kooperatif.....	14
Gambar 2.10 Arsitektur Subsumtion behavior based robot.....	15
Gambar 2.11 Diagram blok sistem kontrol PID .....	15
Gambar 2.12 Blok diagram Kontrol <i>proportional</i> .....	16
Gambar 2.13 Blok diagram Kontrol <i>Integrative</i> .....	17
Gambar 2.14 Blok diagram Kontrol <i>Derifative</i> .....	18
Gambar 3.1 Tahap Penelitian.....	23
Gambar 3.2 Blok diagram robot beroda pemadam api .....	23
Gambar 3.3 Desain robot tampak depan .....	26
Gambar 3.4 Desain robot tampak samping .....	26
Gambar 3.5 perancangan desain sensor Ultrasonik HC-SR04 dengan Arduino..	27
Gambar 3.6 Perancangan sensor photodiode dengan Arduino .....	28
Gambar 3.7 Skema rangkaian photodiode menggunakan <i>software eagle</i> .....	28
Gambar 3.8 Perancangan desain sensor flame lima <i>chanel</i> dengan Arduino .....	30
Gambar 3.9 Desain penempatan sensor flame dengan servo dan kipas DC .....	30
Gambar 3.10 Rangkaian <i>driver</i> motor H-Bridge mosfet menggunakan <i>software Eagle</i> .....	32
Gambar 3.11 Perancangan sistem kontrol <i>behavior based robot</i> .....	32
Gambar 3.12 Perancangan sistem koordinator <i>kompetitif behavior based robot</i> .....	33
Gambar 3.13 Perancangan diagram blok kontrol PID .....	34
Gambar 3.14 Diagram Alir gerakan robot .....	37



Gambar 3.15 Diagram Alir Kerja Sistem Keseluruhan .....	39
Gambar 3.16 hasil perancangan desain robot tampak depan .....	43
Gambar 3.17 hasil perancangan desain robot tampak samping .....	43
Gambar 3.18 hasil perancangan desain robot tampak atas .....	44
Gambar 4.1 Grafik pengujian pengaruh jarak terhadap waktu pada sensor HC-SR04.....	49
Gambar 4.2 grafik perbandingan pembacaan sensor HC-SR04 dengan alat ukur .....	49
Gambar 4.3 Grafik pengujian sensor photodiode terhadap warna putih.....	56
Gambar 4.4 Grafik rata-rata pembacaan warna putih oleh kelima sensor .....	56
Gambar 4.5 Grafik pengujian pembacaan sensor photodiode pada warna Hitam.....	59
Gambar 4.6 Grafik rata-rata pembacaan warna hitam oleh kelima sensor .....	59
Gambar 4.7 Grafik respon kontrol proposional saat nilai Kp 7 .....	65
Gambar 4.8 Grafik respon kontrol proposional saat nilai Kp 15 .....	66
Gambar 4.9 Grafik respon kontrol proposional saat nilai Kp 18.....	66
Gambar 4.10 Grafik respon kontrol proposional dan derivative saat nilai Kp 18 dan Kd 7 .....	68
Gambar 4.11 Grafik respon kontrol proposional dan derivative saat nilai Kp 18 dan Kd 15 .....	68
Gambar 4.12 Grafik respon kontrol proposional dan derivative saat nilai Kp 18 dan Kd 19 .....	69
Gambar 4.13 Grafik respon kontrol proposional, integrative dan derivative saat nilai Kp 18, Ki 5 dan Kd 19.....	70
Gambar 4.14 Grafik respon kontrol proposional, integrative dan derivative saat nilai Kp 18, Ki 8 dan Kd 19.....	71
Gambar 4.15 Grafik respon kontrol proposional, integrative dan derivative saat nilai Kp 18, Ki 14 dan Kd 19.....	71
Gambar 4.16 Grafik perbandingan parameter kontrol PID.....	72
Gambar 4.17 Gelombang osilasi metode <i>Zieger-Nichole</i> .....	74
Gambar 4.18 Grafik kontro PID menggunakan metode <i>Zieger-Nichole</i> .....	75



Gambar 4.19 Grafik Perbandingan metode <i>Trial and Error</i> dengan <i>Zieger-Nichole</i> .....	75
Gambar 4.20 Ilustrasi cara kerja robot dalam menjalankan misinya.....	78
Gambar 4.21 Ilustrasi cara kerja robot jika tidak terdapat api pada semua ruangan.....	80
Gambar 4.22 Kondisi ruangan saat pengujian pertama .....	86
Gambar 4.23 Kondisi ruangan saat pengujian kedua.....	87
Gambar 4.24 Kondisi ruangan saat pengujian ketiga.....	88
Gambar 4.25 Kondisi ruangan saat pengujian keempat.....	89



**DAFTAR TABEL**

Tabel 2.1 Spesifikasi Arduino Mega 2560.....	6
Tabel 2.2 Spesifikasi Sensor HC-SR04 .....	9
Table 2.3 Tanggapan sistem kontrol PID terhadap perubahan parameter .....	19
Tabel 4.1 Pengujian pengaruh jarak terhadap waktu pada sensor HC-SR04.....	48
Tabel 4.2 Pengujian pengaruh sudut terhadap jarak pada sensor HC-SR04.....	51
Tabel 4.3 pengujian sensor photodiode terhadap warna putih.....	54
Tabel 4.4 pengujian pembacaan sensor photodiode pada warna hitam .....	57
Tabel 4.5 pengujian pembacaan sensor flame pada malam hari .....	61
Tabel 4.6 pengujian pembacaan sensor flame pada siang hari .....	63
Tabel 4.7 pengujian parameter proporsional (Kp) .....	65
Tabel 4.8 pengujian parameter kontrol proporsional (Kp) dan derivative (Kd) .....	67
Tabel 4.9 pengujian parameter kontrol proporsional (Kp) integrative (Ki) dan derivative (Kd).....	70
Tabel 4.10 Perbandingan kontrol PID <i>trial and error</i> dan <i>Ziegler-Nichols</i> .....	76
Tabel 4.11 Pengujian metode <i>behavior based</i> .....	78
Tabel 4.12 Pengujian <i>behavior</i> menyisir dinding .....	81
Tabel 4.13 Pengujian <i>behavior</i> mengenali posisi <i>start (home)</i> .....	82
Tabel 4.14 Pengujian <i>behavior</i> mendeteksi api .....	83
Tabel 4.15 Pengujian <i>behavior</i> memadamkan api .....	84
Tabel 4.16 Pengujian robot pertama .....	86
Tabel 4.17 Pengujian robot kedua.....	87
Tabel 4.18 Pengujian robot ketiga .....	88
Tabel 4.19 Pengujian robot keempat.....	89

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan dunia robotika memiliki unsur yang sedikit berbeda dengan ilmu-ilmu dasar atau terapan lainnya. Ilmu dasar biasanya berkembang dari suatu asas atau hipotesa yang kemudian diteliti secara metodis, sedangkan ilmu robotika lebih sering berkembang melalui pendekatan praktis. Kemudian melalui pendekatan atau asumsi dari hasil pengamatan perilaku makhluk hidup atau peralatan bergerak lainnya dikembangkanlah penelitian secara teoritis.

Robotika sendiri merupakan bukti perkembangan teknologi dari waktu ke waktu. Wujud robot bukan hanya menyerupai manusia atau binatang namun juga dapat bekerja layaknya bentuk yang dituranya. Kemampuan robot untuk dapat menyelesaikan misinya sangat dipengaruhi oleh kemampuan robot dalam bernavigasi sehingga dapat menunjang kinerja optimum dari robot. Robot pemadam api adalah robot cerdas yang dapat berjalan mencari target titik api dan dapat menghindari serta mengenali simpangan pada suatu arena atau track.

Dalam banyak pengaplikasiannya, sering kali diperlukan respon yang cepat dari robot dalam menyelesaikan misi. *Behavior-based* merupakan suatu konsep sistem kendali dimana robot dapat beradaptasi pada lingkungannya berdasarkan tingkah laku yang diberikan. Karena sistem navigasi robot pemadam api secara otonom, maka robot harus memiliki kemampuan untuk mengenali garis *start*, menghindari halangan, mengenali simpangan serta mendeteksi letak titik api secara adaptif terhadap lingkungannya. Arsitektur *Behavior-based* merupakan suatu sistem kendali yang tidak berbasis model, karena memiliki struktur *behavior* yang bekerja bersama secara paralel.

Dalam *Behavior-based* sistem kendali robot dipisahkan berdasarkan tugas yang ingin dicapai yang disebut dengan *behavior*. Semakin banyak tugas sistemnya maka akan semakin kompleks, sehingga dapat menimbulkan konflik antar *behavior*. Oleh karena itu dikembangkan metode koordinasi antar *behavior*.

Terdapat dua pendekatan mekanisme koordinasi, yaitu *Competitive* dan *Cooperative*. Pada metode *Competitive* hanya satu *behavior* yang diijinkan memberikan sinyal kendali. Sedangkan koordinasi *Cooperative* menggabungkan semua keluran *behavior* yang ada.

Dalam penelitian kali ini akan dikembangkan suatu metode koordinasi *competitive* untuk mengkoordinasikan struktur kendali *Behavior-based control* pada robot pemadam api. Serta control PID untuk memuluskan pergerakan robot saat menelusuri ruangan agar pergerakan robot mampu bermanuver dengan baik, halus, responsive dan cepat dengan kecermatan tinggi sehingga pergerakan robot lebih stabil sesuai dengan medan dinamis yang dihadapi.

Robot sendiri diciptakan untuk memudahkan pekerjaan manusia dalam menyelesaikan masalah. Salah satu pekerjaan manusia yang dapat dikerjakan oleh robot adalah mendeteksi serta memadamkan api. Pekerjaan seperti ini membutuhkan respon yang cepat karna kebakaran dapat dihindari apabila api belum menyebar. Masalah seperti ini akan dapat dihindari apabila sumber api dapat ditemukan dan dipadamkan dalam waktu yang singkat, karena apabila api sudah menyebar maka resiko bagi tim pemadam kebakaran untuk memadamkan api akan jauh lebih besar.

Berdasarkan permasalahan itulah, pada penelitian ini akan dikembangkan robot pemadam api dengan kemampuan bernavigasi dengan menggunakan metode *Behavior-Based* serta control PID. Oleh sebab itu pada penelitian ini akan dirancang suatu pemodelan system robot yang berjudul “Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode *Behavior-Based* dan Kontrol PID pada Manuver Robot Pemadam Api”.

## 1.2 Rumusan masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana membuat kontrol PID pada robot beroda pemadam api dengan gerak mengikuti dinding agar bisa bergerak sesuai dengan yang diinginkan?
2. Dapatkah metode *behavior based* mengontrol sistem navigasi robot beroda pemadam api sehingga robot tidak menabrak dinding saat bergerak?
3. Bagaimana membuat mekanisme koordinasi antar *behavior*?

## 1.3 Batasan Masalah

Ruang lingkup kegiatan ini berisi tentang batasan – batasan masalah dalam pembuatan alat, dimana isi dari batasan – batasan tersebut adalah sebagai berikut :

1. Menggunakan robot mobile beroda yang menggunakan tipe *differensial steering*.
2. Menggunakan sensor Flame untuk mendeteksi sumber api.
3. Menggunakan sensor Ultrasonik HC-SR04 sebanyak lima buah yang akan dipasang pada bagian kanan, kiri dan depan pada *body* robot untuk mendeteksi jarak.
4. Menggunakan sensor Phothodiada untuk mendeteksi garis yang berada di *home*.
5. Menggunakan Arduino Mega 2560 sebagai mikrokontroler.

## 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian adalah sebagai berikut :

1. Rancang bangun robot navigasi pemadam api yang dapat mencari dan memadamkan sumber api.
2. Robot dapat bergerak dengan teratur dengan adanya sistem navigasi dengan menggunakan metode *Behaviour-Based*.

3. Mampu mengontrol gerak robot dengan sistem *tracking* susur kanan / susur kiri dengan menerapkan kontrol PID pada robot.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah menambah pengetahuan penulis mengenai sistem navigasi robot menggunakan metode *Behaviour-Based* yang dapat beradaptasi pada lingkungannya berdasarkan perilaku yang diberikan. Serta digunakan sebagai bahan praktisi dan pengamatan terutama dalam bidang navigasi robot.





## BAB 2. TINJAUAN PUSTAKA

Pada tinjauan pustaka ini, dibuat sebagai bahan dasar acuan penulis untuk melakukan sebuah penelitian tugas akhir yang akan menjelaskan mengenai komponen yang digunakan. Selain itu tinjauan pustaka juga sebagai memperluas wawasan dan mengembangkan ilmu pengetahuan yang telah ada sehingga bidang yang diteliti akan mengalami perkembangan yang lebih baik lagi.

### 2.1 Penelitian Terkait

Pada bagian ini berisikan tinjauan penelitian yang akan membahas mengenai robot *mobile* pemadam api yang menggunakan motor sebagai penggerakannya. Robot adalah sebuah alat mekanik yang dapat bekerja secara terus-menerus untuk membantu pekerjaan manusia, yang dalam menjalankan tugasnya dapat dikontrol langsung oleh manusia ataupun bekerja secara otomatis, sesuai program yang telah ditanamkan pada chip mikrokontrolnya.

Robot beroda pemadam api adalah robot yang dapat membantu manusia untuk mendeteksi serta memadamkan api secara dini sehingga tidak akan terjadi kebakaran apabila terlambat menyadari ada sumber api yang tidak diketahui. Robot ini didesain menggunakan motor DC sebagai penggerakannya karena membutuhkan kecepatan serta akurasi dalam bernavigasi untuk segera menemukan sumber api.

### 2.2 Arduino Mega 2560

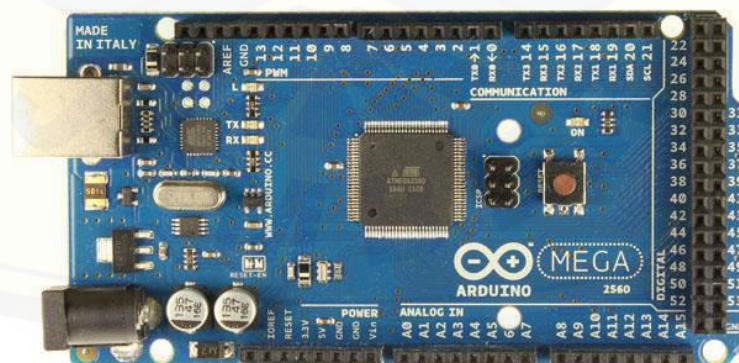
Arduino mega 2560 merupakan mikrokontroler yang berbasis Arduino dengan menggunakan chip Atmega2560. Modul ini memiliki 54 buah pin I/O yang dibagi menjadi 15 pin PWM, 16 pin analog *input*, 4 pin UART (serial port hardware). Modul ini juga dilengkapi dengan oscillator 16 Mhz, sebuah koneksi USB yang digunakan untuk mengupload program, sebuah jack DC yang dapat digunakan sebagai catu daya, serta tombol *push button* yang digunakan sebagai tombol reset. Adapun spesifikasi dari Arduino mega 2560 sendiri adalah sebagai berikut:



Tabel 2.1 Spesifikasi Arduino Mega 2560

Chip mikrokontroler	ATmega2560
Tegangan operasi	5V
Tegangan input	7V – 12V
Tegangan input (limit, via jack DC)	6V – 20V
Digital I/O pin	54 buah, 6 diantara merupakan pin PWM
Analog Input pin	16 buah
Arus DC per pin I/O	20 Ma
Arus DC pin 3.3V	50 Ma
Memori flash	256 KB, 8KB telah digunakan untuk bootloader
SRAM	8 KB
EEPROM	4 KB
Clock speed	16 Mhz
Dimensi	101.5 x 53.4 mm
Berat	37 g

Sumber : (yuhardiasyah, 2016)



Gambar 2.1 Bentuk Fisik Arduino Mega 2560

Sumber : (Arduino.com)

### 2.3 Sensor Ultrasonic HC-SR04

Sensor ultrasonic HC-SR04 merupakan sensor yang digunakan untuk mengukur jarak. Sensor ini dapat bekerja dengan range antara 2cm sampai 4 meter. Sensor ini memiliki 4 buah pin yaitu pin VCC, *Triger*, *Echo* dan GND. Pin VCC digunakan sebagai tegangan positif sebesar 5V, *Triger* berfungsi untuk memancarkan gelombang ultrasonic, *Echo* berfungsi untuk menangkap sinyal yang dipancarkan oleh *Triger*, dan GND digunakan untuk *ground*-nya.

Gelombang ultrasonic merupakan gelombang bunyi yang memiliki frekuensi sangat tinggi yaitu 20.000 HZ. Gelombang ini tidak dapat didengar oleh manusia namun dapat didengar oleh anjing, kelelawar dan lumba – lumba. Gelombang ultrasonic dapat merambat melalui zat padat, cair maupun gas.



Gambar 2.2 Bentuk Fisik Sensor Ultrasonic HC-SR04

Sumber : (depokinstrumen.com)

Cara menggunakan sensor ini yaitu dengan memberikan tegangan positif pada pin VCC sebesar 5V dan tegangan negative pada pin GND. Pin Triger akan bekerja dengan memancarkan gelombang ultrasonic dengan frekuensi sebesar 40kHz yang dihasilkan oleh suatu alat yang disebut *Piezoelektrik*. Sinyal ini akan merambat sebagai gelombang bunyi dengan kecepatan 340 m/s. Jika gelombang membentur suatu objek maka gelombang akan dipantulkan oleh objek tersebut, dan sinyal pantulan itulah yang akan diterima oleh pin *Echo*. Kecepatan rambat gelombang dan selisih waktu saat mengirim dan menerima inilah yang akan dikonversikan menjadi jarak dengan menggunakan rumus persamaan 2.1 (depokinstrumen.com)

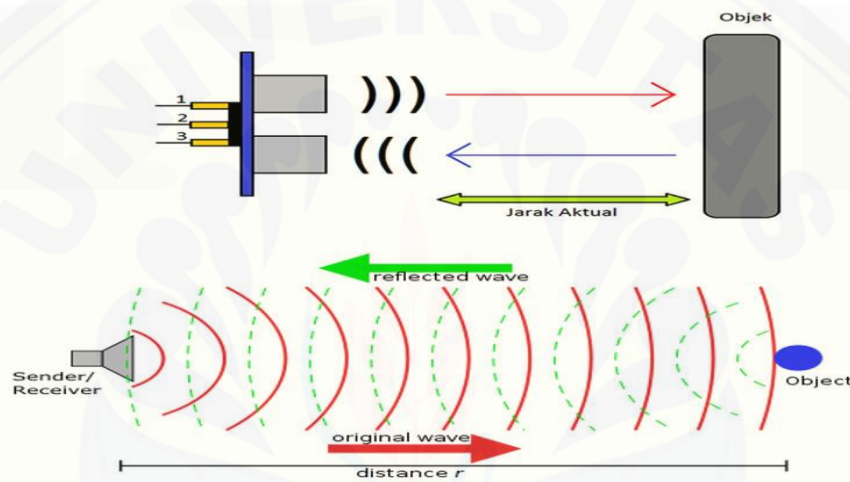
$$S = v \cdot t \quad (2.1)$$

$$S = 340 \cdot t/2$$

Dengan :

S = Jarak sensor ultrasonik dengan benda

t = Selisih antara waktu pemancaran gelombang dan waktu ketika gelombang diterima.



Gambar 2.3 Cara Kerja Sensor Ultrasonic Dengan Transmister dan Reciver

Sumber : (elangskrafti.com)

Jika dibandingkan dengan sensor SRF05 dan SRF02, sensor HC-SR04 jauh lebih stabil dalam pengukuran jarak. Sensor HC-SR04 tidak dapat mendeteksi benda jika jaraknya kurang dari 2 cm (*blind spot*). Berikut adalah spesifikasi dari sensor HC-SR04:

Tabel 2.2 Spesifikasi Sensor HC-SR04

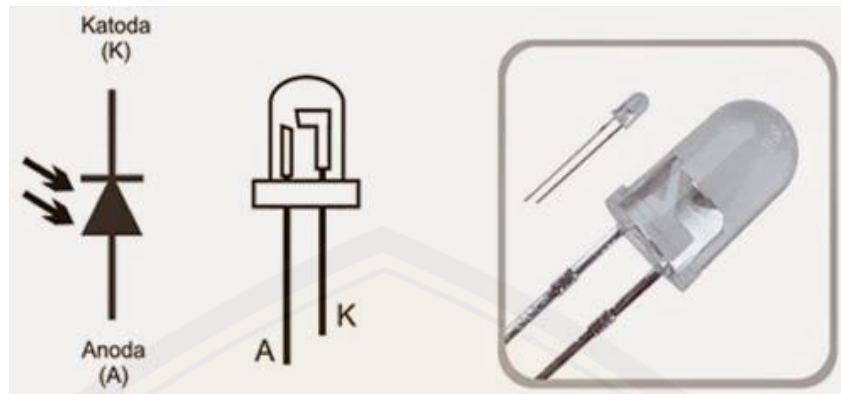
Tegangan	5V DC
Arus Statis	< 2 Ma
Level Output	5V – 0V
Sudut Sensor	< 15°
Jarak yang dapat dideteksi	2cm – 450 cm (4,5 meter)
Tingkat keakuratan	Hampir 0.3 cm (3 mm)
Frekuensi	40 kHz
Resolusi	1 cm

Sumber : (digi – bytes.com)

#### 2.4 Sensor Photodiode

Photodiode merupakan sensor yang peka terhadap cahaya, sensor ini akan mengalami perubahan resistansi saat menerima cahaya dan akan menghasilkan arus listrik secara forward sebagai mana dioda pada umumnya. Secara teori jika semakin gelap cahaya yang diterima maka nilai resistansinya akan semakin besar, yang mengakibatkan arus yang mengalir ke kaki photodiode semakin kecil, dan sebaliknya jika intensitas cahaya yang diterima semakin terang maka nilai resistansinya akan semakin kecil, sehingga arus yang dialirkan pada kaki photodiode akan semakin besar.

Photodiode dibuat dari bahan semikonduktor yang umumnya menggunakan bahan silicon (Si) atau gallium arsenide (GaAs). Karakteristik panjang gelombang dari material ini dalam menyerap cahaya yaitu anatar 2.500 Å – 11.000 Å untuk silicon, dan 8.000 Å – 20.000 Å untuk GaAs



Gambar 2.4 Simbol dan Bentuk Fisik Photodioda

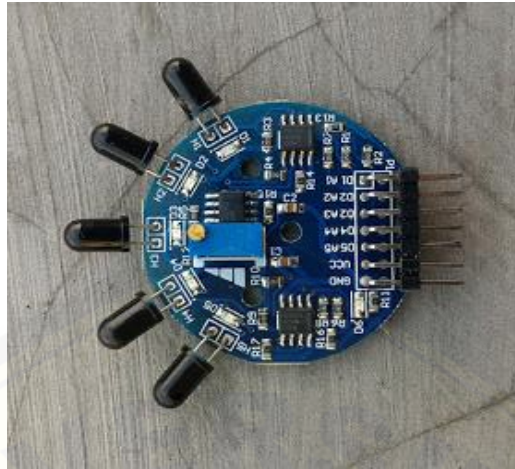
Sumber : (Margino, 2015)

Prinsip kerja dari photodiode yaitu, pada saat photon (satu satuan energi dalam cahaya) dari sebuah sumber cahaya diserap, maka hal tersebut akan mengakibatkan terjadinya pergeseran foton sehingga akan menghasilkan sepasang pembawa muatan tunggal yaitu electron dan hole pada kedua sisi, dimana electron akan menuju ke sumber positif (+) sedangkan hole akan menuju ke sumber negatif (-), sehingga dari pergeseran foton inilah arus akan dapat mengalir didalam rangkaian.

## 2.5 Sensor Api Flame

Sensor flame digunakan untuk mendeteksi api jarak dekat. Sensor ini memiliki tujuh buah pin yaitu VCC, GND, lima pin Analog *Output* dan digital *Output*. Pin analog *output* sensor dapat memperkirakan letak dari sumber api meskipun pembacaannya masih belum akurat, sedangkan jika menggunakan pin analog *output* hanya akan dapat mengetahui ada api atau tidak namun tidak dapat mengetahui letak sumber api. Sensor ini bekerja dengan cara mendeteksi pancaran sinar infrared dengan rentang panjang gelombang 760 nm sampai dengan 1100 nm. Jarak pembacaan dari sensor flame bisa mencapai 3 *feet* atau sekitar 91 cm.





Gambar 2.5 Bentuk Fisik Sensor Flame

Sumber : (Elektronik Nusantara, 2016)

## 2.6 Motor DC Gear Box

Motor DC merupakan salah satu jenis motor listrik yang memerlukan *suplay* tegangan arus searah (*direct curren*) yang akan diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut *stator* (bagian yang tidak bergerak) sedangkan bagian jangkar pada motor DC disebut *rotor* (bagian yang bergerak). Keuntungan pertama dari penggunaan motor DC sendiri adalah dalam hal pengendalian kecepatannya karena tidak mempengaruhi kualitas pasokan daya. Kecepatan motor ini dapat diatur dengan menggunakan nilai *PWM* yang ada pada pin Arduino.

Pada robot ini nantinya akan menggunakan dua buah motor DC yang akan berfungsi sebagai alat penggerak pada roda kanan dan roda kiri. Motor yang digunakan harus memiliki spesifikasi yang sama agar robot dapat berjalan dengan baik. Type motor yang akan digunakan pada robot ini adalah GA25Y370 dengan spesifikasi kecepatan sekitar 200 RPM, Torque sekitar 4 kg serta membutuhkan tegangan input sebesar 12V.



Gambar 2.6 Bentuk Fisik Gearbox Motor DC

Sumber : (Dagu Hitech Electronic)

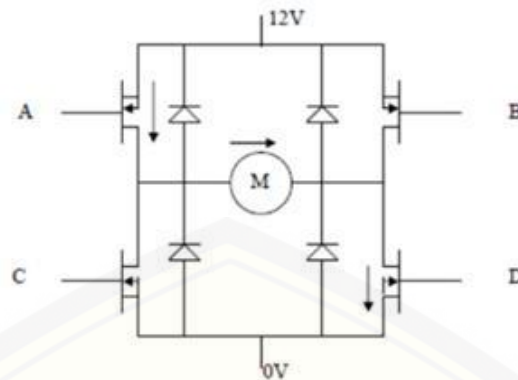
### **2.7 Driver Motor *H-Bridge* Mosfet**

*Driver* motor adalah sebuah rangkaian yang digunakan untuk mengatur kecepatan putaran motor serta arah putaran motor. Ada beberapa jenis *driver* motor yang sering diaplikasikan pada robot yang diantaranya adalah *H-Bridge* transistor, *H-Bridge* Mosfet dan IC driver motor seperti L293 dan L298.

Salah satu *driver* motor yang sering digunakan adalah *H-Bridge* Mosfet, rangkain ini terdiri dari dua buah mosfet kanal P dan dua buah kanal N yang akan bekerja dengan mengatur hidup matinya motor DC yang akan dikendalikan. Transistor jenis mosfet dipilih karena mosfet dapat dilalui arus yang relative lebih besar jika dibandingkan dengan transistor jenis lain.

*Driver* motor *H-Bridge* Mosfet adalah driver motor yang dapat dapat bekerja dengan arus yang cukup besar yaitu lebih dari satu amper dan tegangan yang juga cukup besar. *Driver* motor ini dapat mengubah arah putaran serta kecepatan putaran motor dengan menggunakan metode PWM. Prinsip kerja dari *driver* motor *H-Bridge* Mosfet dapat dilihat dari rangkaian berikut:





Gambar 2.7 Rangkaian Driver Motor *H-Bridge* Mosfet

Sumber : (Fahmizal, 2011)

Pada Gambar 2.7 bagian atas rangkaian akan dihubungkan dengan catu daya positif, sedangkan bagian bawah rangkaian akan dihubungkan dengan *ground*. Huruf “M” pada Gambar 2.7 menyatakan motor DC yang akan dikendalikan. Pada saat mosfet A dan mosfet D *on* sedangkan mosfet B dan mosfet C *off*, maka arus akan mengalir dari tegangan 12V ke mosfet A kemudian akan dialirkan ke motor dan menuju mosfet D yang terhubung dengan *ground*, sehingga motor akan berputar searah jarum jam.

## 2.8 Behaviour Based

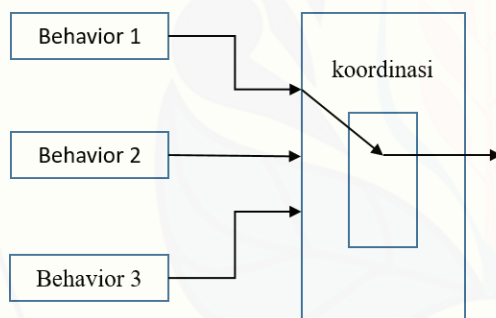
Pada banyak pengaplikasian robotika, seperti sistem *autonomos* robot (robot mandiri) pada lingkungan yang tidak terstruktur sangat sulit untuk mendapatkan model matematis yang tepat dari interaksi robot dengan lingkungannya. Sehingga untuk mewujudkan tujuan tersebut maka pertama-tama diperlukan sistem kendali robot yang tidak berbasiskan model yang dikenal dengan metode *Behavior Based Robot*.

### 2.8.1 Koordinasi *Behavior Based Robot*

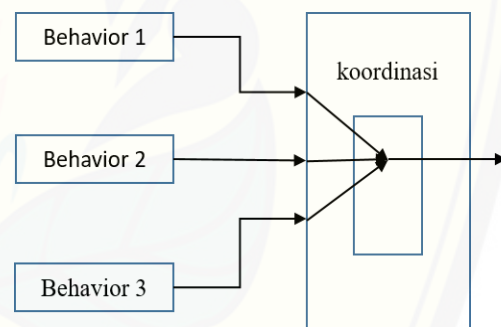
Pada pendekatan ini setiap perilaku diuraikan menjadi beberapa modul yang masing-masing bertanggung jawab untuk melakukan satu perilaku yang telah ditanamkan pada robot. Setiap perilaku memegang kendali lengkap mulai dari sensing sampai dengan aksi. Kumpulan dari beberapa *behavior* yang terdapat dalam

satu modul dapat menimbulkan konflik yang tidak dapat diselesaikan sehingga robot akan *error*.

Oleh karena itu untuk menyelesaikan masalah ini maka diperlukan formulasi koordinasi yang efektif dari setiap perilaku, sehingga hanya ada satu perilaku pada setiap aksi robot. Koordinasi yang digunakan pada *Behavior Based Robot* sendiri dibagi menjadi dua metode, yaitu koordinasi kompetitif dan koordinasi kooperatif. Koordinasi kompetitif sendiri hanya mengizinkan satu perilaku untuk menentukan aksi robot yang akan digunakan, sedangkan koordinasi kooperatif adalah kebalikan dari koordinasi kompetitif dimana koordinasi menjumlahkan semua perilaku yang kemudian akan dikalikan dengan suatu factor penguatan, namun kelemahan dari koordinasi ini adalah kesalahan kontrol yang sering muncul ketika terdapat perubahan lingkungan



Gambar 2.8 Koordinasi Kompetitif

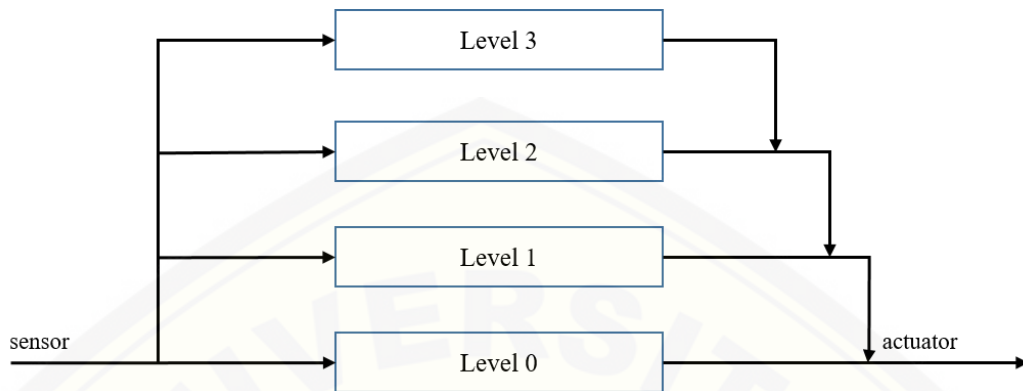


Gambar 2.9 Koordinasi Kooperatif

### 2.8.2 Arsitektur *Subsumption*

Arsitektur yang digunakan pada metode *Behavior Based Robot* adalah *subsumption* yang diusulkan oleh Rodney Brooks. Dalam membangun robotnya Rodney Brooks menguraikan permasalahan sistem kontrol berdasarkan manifestasi luar yang diinginkan oleh sistem kontrol robot, tidak berdasarkan operasi internal dari sistem kontrol robot. Arsitektur ini membuat beberapa tingkatan dari setiap perilaku yang akan disusun menjadi beberapa level sehingga pada level yang tertinggi dianggap memegang kendali penuh dari setiap *behavior*. Arsitektur ini

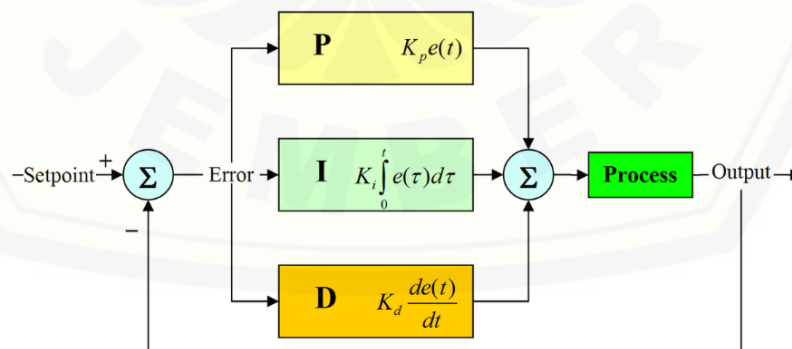
memiliki kelebihan yang mudah ditata sehingga aksi yang dihasilkan adalah pemilihan dari satu *behavior*.



Gambar 2.10 Arsitektur Subsumtion behavior based robot

## 2.9 Kontroler PID

Kontrol PID adalah adalah kontrol umpan balik yang dapat meminimalisir kesalahan atau nilai error pada suatu sistem. Kontroler PID terdiri dari tiga konstanta parameter, yaitu *Proportional* (P), *Integrivel* (I) dan *Derivatif* (D). Ketiga konstanta tersebut sangat berpengaruh terhadap respon output suatu sistem, setiap kekurangan dan kelebihan dari masing-masing konstanta dapat saling menutupi dengan menggabungkan masing-masing konstanta tergantung dari keperluan suatu sistem tersebut.

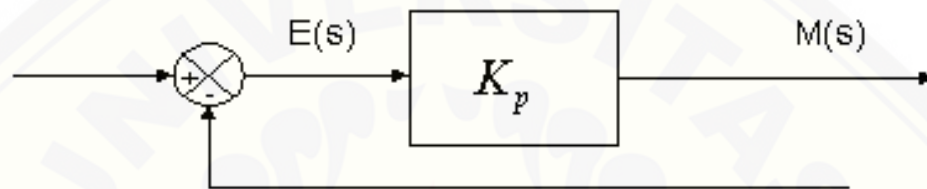


Gambar 2.11 Diagram blok sistem kontrol PID

Sumber : (Asharudin Achzab, 2014)

### 2.9.1 Kontrol *Proportional* (P)

Karakteristik dari aksi kontrol *proportional* adalah untuk memperbaiki respon transien atau mengurangi waktu naik. Penambahan kontrol *proportional* memiliki konsekuensi nilai *overshoot* yang cukup besar jika nilai yang digunakan tidak tepat. Kenaikan nilai *overshoot* sebanding dengan kenaikan nilai parameter  $K_p$  karena output dari kontrol *proportional* adalah hasil perkalian antara konstanta *proportional* dengan nilai *error*nya.



Gambar 2.12 Blok diagram Kontrol *proportional*

Sumber : (Elektronika Indonesia)

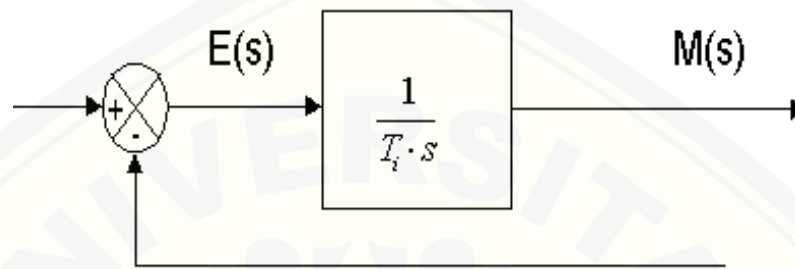
Kontrol *proportional* jika diterapkan kedalam suatu sistem dapat diketahui dengan memperhatikan ciri-ciri sebagai berikut :

- 1) Jika nilai *proportional* ( $K_p$ ) yang digunakan kecil maka kontrol *proportional* hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan mengakibatkan respon sistem yang lambat.
- 2) Jika nilai *Proportional* ( $K_p$ ) yang digunakan dinaikkan maka kontrol *proportional* akan mampu mengoreksi kesalahan yang lebih besar sehingga respon sistem akan semakin cepat.
- 3) Jika nilai *proportional* ( $K_p$ ) yang digunakan dinaikkan sampai melebihi nilai *set point*, maka sistem akan bekerja secara tidak stabil atau Sistem akan mengalami osilasi (Pakpaham, 1988).

### 2.9.2 Kontrol *Integrative* (I)

Karakteristik dari kontrol *integrative* (I) adalah menghilangkan nilai *error steady state*, namun kontrol integral tidak akan mampu menjamin *output* dari sistem

akan menuju nilai sesuai yang kita inginkan kalau sebuah plant tidak memiliki unsur *integrator*. Keluaran kontrol *integrative* sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahannya (Rusli, 1997). Jika sinyal kesalahan tidak mengalami perubahan maka keluaran akan menjaga keadaan seperti sebelum terjadi perubahan masukan.



Gambar 2.13 Blok diagram Kontrol *Integrative*

Sumber : (Elektronika Indonesia)

Kontrol *Integrative* jika diterapkan kedalam suatu sistem dapat diketahui dengan memperhatikan ciri-ciri sebagai berikut :

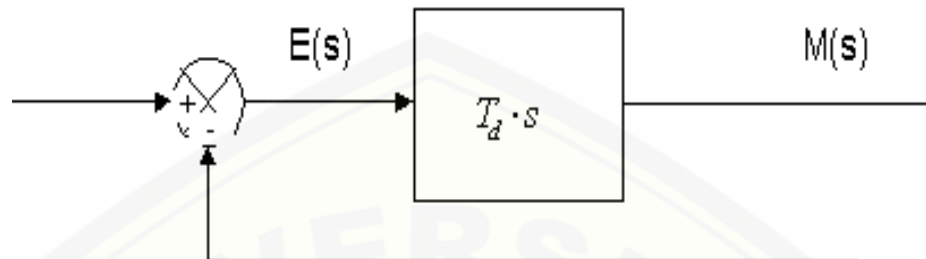
- 1) Ketika sinyal kesalahan berharga nol, maka keluaran kontroler akan bertahan pada nilai sebelumnya.
- 2) Ketika sinyal kesalahan tidak sama berharga nol, maka keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai  $K_i$  (Johnson, 1993).
- 3) Konstanta *integrative*  $K_i$  yang berharga besar akan mempercepat hilangnya osilasi, namun jika semakin besar nilai konstanta  $K_i$  akan mengakibatkan peningkatan osilasi dari sinyal keluaran kontroler (Guterus, 1994).

### 2.9.3 Kontrol *Derivative*

Karakteristik dari Kontrol *Derivative* adalah memberikan efek redaman kepada nilai osilasi. Perubahan nilai yang mendadak pada kontroler akan mengakibatkan perubahan yang sangat besar dan cepat kepada suatu sistem kontrol namun jika ketika masukannya tidak mengalami perubahan maka keluaran kontroler juga tidak mengalami perubahan. Kontroler derivative umumnya dipakai



untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Oleh karena itu kontroler derifative tidak pernah digunakan tanpa ada tambahan kontroler lain (Sutrisno, 1990)



Gambar 2.14 Blok diagram Kontrol *Derivative*

Sumber : (Elektronika Indonesia)

Kontrol *Derivative* jika diterapkan kedalam suatu sistem dapat diketahui dengan memperhatikan ciri-ciri sebagai berikut :

- 1) Kontroler *Derivative* tidak dapat menghasilkan keluaran apabila tidak ada perubahan pada masukannya.
- 2) Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan tergantung pada nilai  $T_d$  dan laju perubahan sinyal kesalahan (Powel, 1994)
- 3) Kontroler *derivative* dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif dan cenderung meningkatkan stabilitas sistem (Ogata, 1997)

#### 2.9.4 Aksi kontrol *Proportional + Integrative + Derivative*

Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan memiliki tujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset serta menghasilkan perubahan awal yang besar (Guterus, 1994). Dalam implementasinya masing-masing konstanta dapat digabungkan secara paralel atau bekerja sendiri. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan sesuai dengan yang diinginkan.



Table 2.3 Tanggapan sistem kontrol PID terhadap perubahan parameter

Respon loop tertutup	Waktu Naik	Overshoot	Waktu Turun	error
Kp	Menurun	Meningkat	Perubahan kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Hilang
Kd	Perubahan kecil	Menurun	Menurun	Perubahan kecil

Sumber : (Muhamad Ali, 2004)

Seperti yang dijelaskan pada Gambar 2.12, Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi dari ketiga parameter P,I dan D, penyetelan konstanta Kp, Ti dan Td akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibandingkan dengan yang lainnya. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994)

Untuk merancang sistem kontrol PID, kebanyakan dilakukan dengan cara coba-coba (*trial and error*). Hal ini karena parameter Kp, Ki dan Kd tidak independent, sehingga untuk mendapatkan aksi kontrol yang baik diperlukan langkah coba-coba dengan menggunakan kombinasi antara P,I dan D samapai ditemukan nilai Kp, Ki dan Kd yang seperti yang diinginkan.

### **BAB 3 METODOLOGI PENELITIAN**

Pembahasan pada bab metode penelitian ini akan dijelaskan mengenai beberapa hal pokok yang di antara lain adalah tempat dan waktu penelitian, alat dan bahan yang digunakan, tahap penelitian, blok diagram sistem, perancangan desain alat, diagram alir (*flowchart*) serta langkah-langkah dalam pengambilan data. Dalam penelitian ini akan membuat desain rancang bangun robot beroda pemadam api menggunakan metode *behavior based* dan kontrol PID pada manuver robot pemadam api yang bertujuan untuk membant manusia dalam mendeteksi adanya sumber api yang tidak diketahui sehingga dapat mencegah terjadinya kebakaran.

#### **3.1 Tempat dan Waktu Penelitian**

##### **3.1.1 Tempat Penelitian**

Adapun tempat dan penelitian uji robot beroda pemadam api menggunakan metode *behavior based* dan kontrol PID pada manuver robot beroda pemadam api akan dilakukan pada dua tempat yaitu untuk pengerjaan alat dan pengambilan data akan dilakukan di Laboratorium Elektronika dan Terapan Fakultas Teknik Universitas Jember yang beralamat di Jln Slamet Riyadi no. 62 Patrang, Jember.

##### **3.1.2 Waktu Penelitian**

Waktu penelitian tentang robot beroda pemadam api menggunakan metode *behavior based* dan kontrol PID pada manuver robot beroda pemadam api akan dilaksanakan dalam jangka waktu selama 5 bulan, yang akan dimulai pada bulan Februari 2018 sampai dengan bulan Juni 2018.

### 3.2 Alat dan bahan

#### 3.2.1 Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan sebagai pendukung pada penelitian ini diantara lain adalah :

- 1) *Software* Arduino yang digunakan untuk melakukan pemograman.
- 2) *Software* Eagle yang digunakan untuk membuat desain rangkaian yang digunakan
- 3) *Software* Microsoft excel 2016 yang digunakan untuk mengolah data yang didapat dari hasil pengujian
- 4) *Software* sketchUP yang digunakan untuk membuat desain bodi robot.

#### 3.2.2 Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) yang digunakan untuk membuat robot pada penelitian ini antara lain adalah :

- 1) Arduino Mega 2560 yang digunakan sebagai mikrokontrol pada robot.
- 2) Sensor Ultrasonik HC-SR04 yang digunakan sebagai sensor jarak untuk membantu navigasi robot.
- 3) Sensor Flame yang digunakan untuk pendeteksi sumber api yang berasal dari lilin.
- 4) Sensor Photodiode yang digunakan untuk mendeteksi perbedaan warna saat robot memasuki ruangan.
- 5) Kipas DC yang digunakan untuk memadamkan sumber api yang berasal dari lilin.
- 6) LCD yang digunakan untuk membantu mengetahui hasil pembacaan sensor.
- 7) Motor Servo yang digunakan untuk memutar sensor flame untuk mendeteksi sumber api pada setiap ruangan.
- 8) Motor DC + Roda yang digunakan sebagai penggerak robot yang diletakkan pada sisi kanan dan kiri bodi robot.

- 9) Free weel yang digunakan sebagai roda bebas untuk membantu pergerakan robot
- 10) Driver Motor yang digunakan untuk mengontrol arah putaran serta kecepatan dari motor DC.
- 11) Akrilik yang digunakan untuk membuat bodi robot secara keseluruhan
- 12) Jumper yang digunakan untuk menghubungkan semua rangkaian.

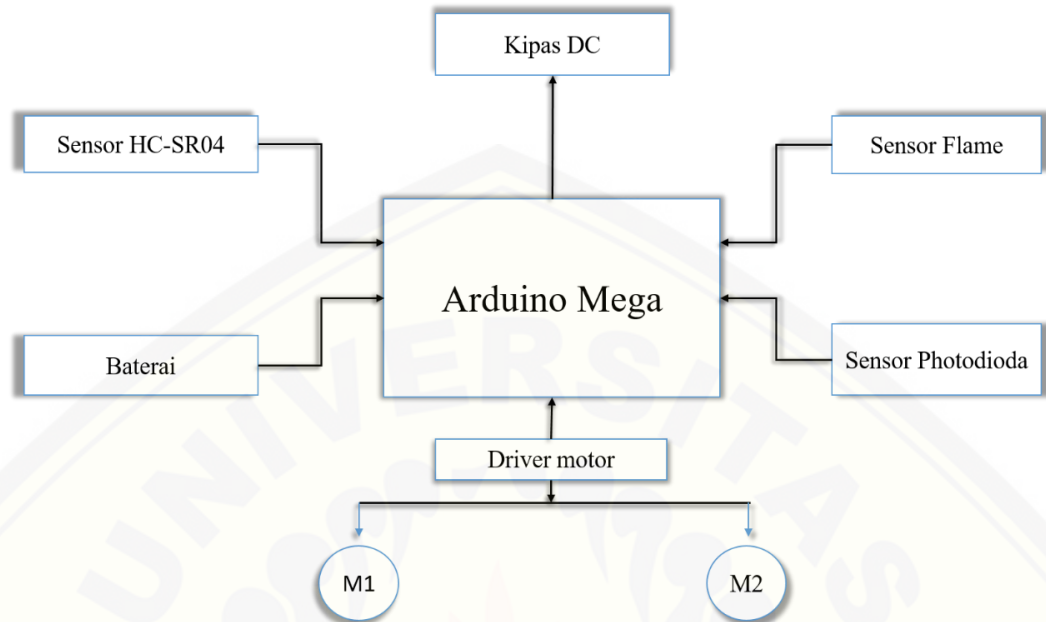
### 3.3 Tahap Penelitian

Pada bab tahap penelitian ini akan menjelaskan mengenai proses pemilihan komponen yang akan digunakan untuk membuat robot, kemudian pengujian komponen yang akan digunakan, perancangan alat penelitian sampai dengan proses pengambilan data dan penyusunan laporan hasil penelitian. Berikut adalah tahapan penelitian yang akan dilakukan :



Gambar 3.1 Tahap Penelitian

### 3.4 Blok Sistem



Gambar 3.2 Blok diagram robot beroda pemadam api

Gambar 3.2 merupakan sebuah blok diagram dari rancang bangun alat yang dibuat, yang menggunakan Arduino Mega sebagai mikrokontrolnya dengan empat buah *input* yang berupa sensor HC-SRO4, sensor flame, sensor photodiode dan baterai dan memiliki dua buah *output* yang berupa kipas DC dan driver motot. Adapun penjelasan dari komponen-komponen yang digunakan adalah sebagai berikut :

- 1) Arduino Mega 2560 digunakan sebagai mikrokontrol dari robot untuk mengatur semua aksi yang akan dilakukan oleh robot yang mendapat *input* dari pembacaan masing-masing sensor.
- 2) Sensor Ultrasonik HC-SRO4 digunakan untuk membaca jarak antar dinding dengan robot yang akan digunakan sebagai masukan dari mikrokontrol untuk memberikan aksi kepada jalannya robot .
- 3) Sensor Photodiode digunakan sebagai pendeteksi garis guna mengetahui robot telah memasuki suatu ruangan yang akan menjadi *input* dari

mikrokontrol untuk memberikan aksi kepada robot melakukan *scan* untuk mendeteksi apakah ada sumber api atau tidak.

- 4) Sensor Flame digunakan untuk mendeteksi keberadaan titik api pada suatu ruangan yang akan menjadi *input* dari mikrokontrol untuk memberikan aksi bergerak mendekati titik api dan memadamkannya.
- 5) Baterai digunakan sebagai *supply* dari Arduino mega, sensor HC-SR04, sensor photodiode, sensor flame, kipas DC, motor DC serta driver motor.
- 6) Kipas DC digunakan sebagai *output* dari mikrokontrol yang mendapat *input* dari sensor flame yang berupa perubahan nilai ADC saat sensor mendeteksi api untuk memberikan aksi memadamkan api.
- 7) *Driver* motor digunakan sebagai *input* dari mikrokontrol, yang mendapat *input* berupa pembacaan jarak dari sensor HC-SR04 yang akan memberikan aksi untuk mengatur arah putaran serta kecepatan putaran dari motor DC.
- 8) Motor DC digunakan sebagai *output* dari *driver* motor guna mengatur pergerakan dari robot yang akan diletakkan pada sisi kanan dan kiri bodi robot.

### 3.5 Perancangan Desain Alat

Pada tahap perancangan desain alat ini akan dibagi menjadi tujuh bagian, yaitu perancangan desain robot, perancangan sensor HC-SR04, perancangan sensor photodiode, perancangan sensor flame, perancangan driver motor, perancangan sistem kontrol behavior based serta perancangan desain kontrol PID. Berikut adalah penjelasan dari perancangan desain alat yang akan dilakukan :

#### 3.5.1 Perancangan desain robot

Pada perancangan desain robot ini nantinya akan menggunakan tiga buah jenis sensor, satu mikrokontrol, satu baterai, serta tiga buah aktuator. Penempatan dari setiap komponen sendiri akan sangat mempengaruhi aksi dan pergerakan yang akan dilakukan oleh robot, oleh karena itu kepresisian dalam penempatan sensor akan sangat mempengaruhi keakuratan sensor dalam mendeteksi suatu objek.

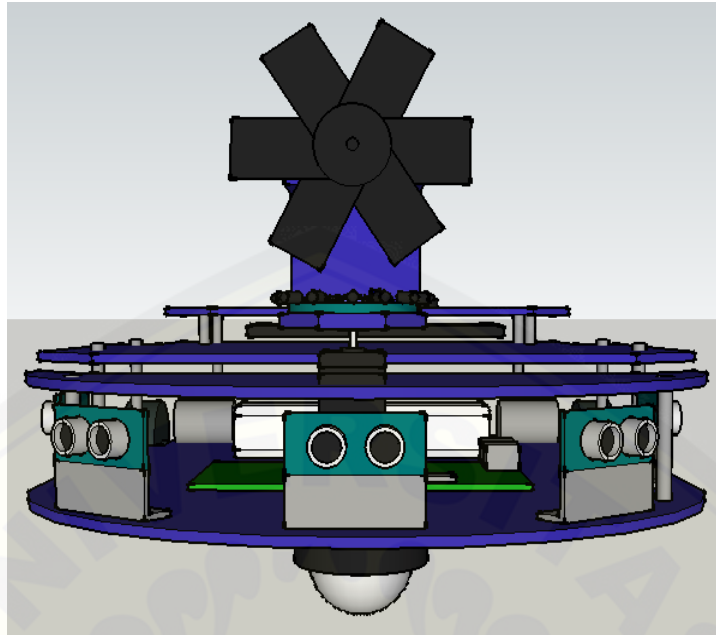


Pada robot ini nantinya akan menggunakan lima buah sensor Ultrasonik HC-SR04 yang akan diletakkan secara melingkar pada bagian depan serta samping kiri dan kanan bodi robot. Sensor ini yang nantinya akan bertanggung jawab untuk mengatur pergerakan dari robot sehingga sensor harus berada pada penempatan yang tepat agar robot tidak sampai menabrak lintasan saat bergerak.

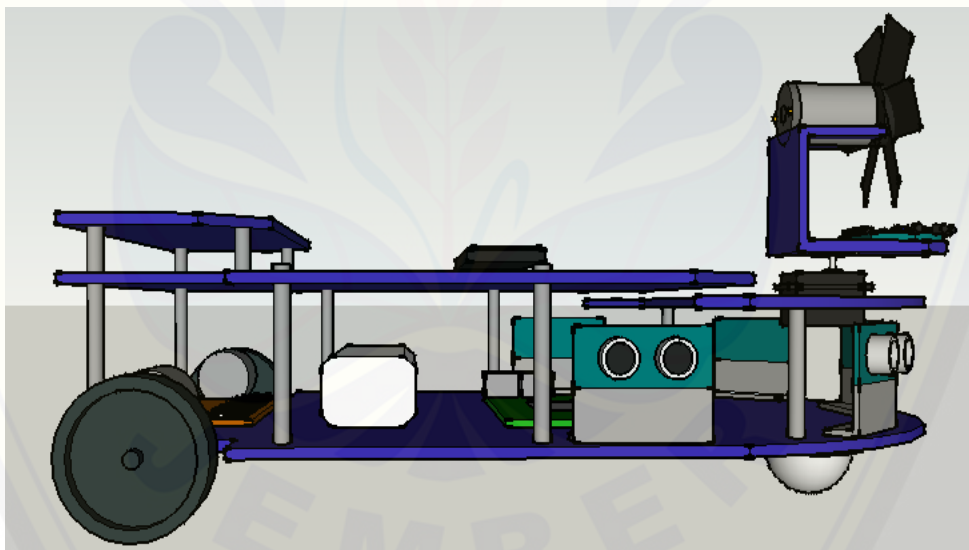
Untuk mendeteksi api satu buah sensor flame akan digabungkan dengan satu buah servo serta sebuah kipas DC yang nantinya akan diletakkan pada bagian depan bodi robot yang akan bertanggung jawab untuk melakukan scan serta mendeteksi api secara bersamaan. Sedangkan untuk mendeteksi garis akan menggunakan lima buah sensor photodiode yang sebelumnya telah digabungkan menjadi sebuah rangkaian yang akan diletakkan pada bagian bawah bodi robot.

Pada bagian belakang robot akan dipasangkan dua buah motor DC yang diletakkan pada bagian kiri dan kanan bodi robot. Penempatan dari motor DC sendiri juga harus presisi karena pergerakan robot juga dipengaruhi oleh spesifikasi serta tata letak dari motor DC sendiri. Jika penempatannya tidak tepat maka robot tidak akan dapat berjalan lurus atau akan cenderung berbelok ke arah kanan atau ke kiri. Pada bagian akan ditambahkan satu buah roda bebas yang nantinya akan membantu pergerakan dari robot agar robot tidak harus menyusur lintasan saat bergerak.

Pada bagian atas dipasang sebuah LCD (*Liquid Crystal Display*) yang akan digunakan sebagai *interface* untuk mengetahui hasil pembacaan dari sensor. Dalam membuat desain robot sendiri beban tidak boleh ditumpukan pada satu titik karena akan membuat gerak robot menjadi lambat dan bodi robot akan sangat rawan patah jika robot menabrak dinding arena.



Gambar 3.3 Desain robot tampak depan

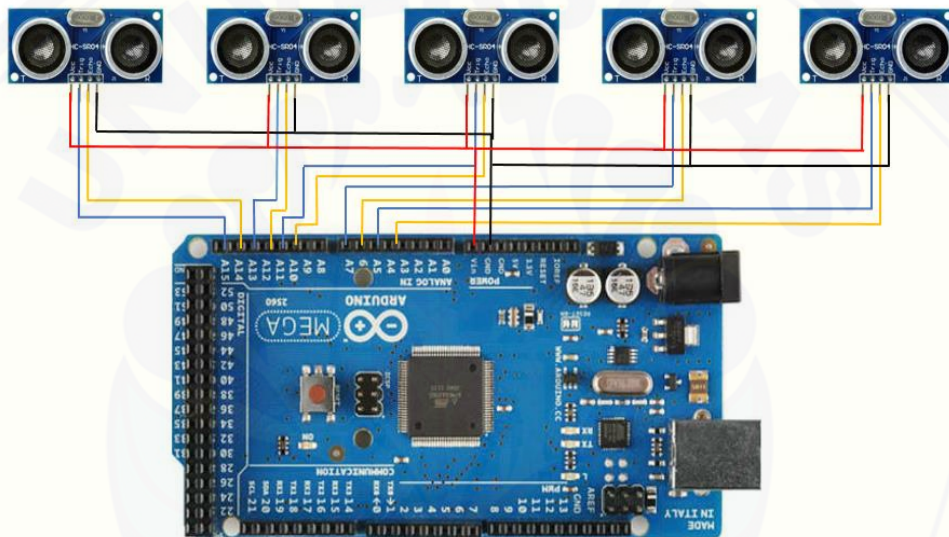


Gambar 3.4 Desain robot tampak samping

### 3.5.2 Perancangan desain sensor HC-SR04

Sensor Ultrasonik HC-SR04 adalah sensor yang digunakan untuk membaca jarak robot terhadap objek. Sensor ini memiliki empat buah pin yang antara lain adalah *VCC*, *GND*, *Echo* serta *Triger*. Sensor ini bekerja dengan cara menembakkan sebuah gelombang ultrasonik yang nantinya pantulan dari gelombang inilah yang akan ditangkap dan kemudian dikonversikan menjadi jarak.

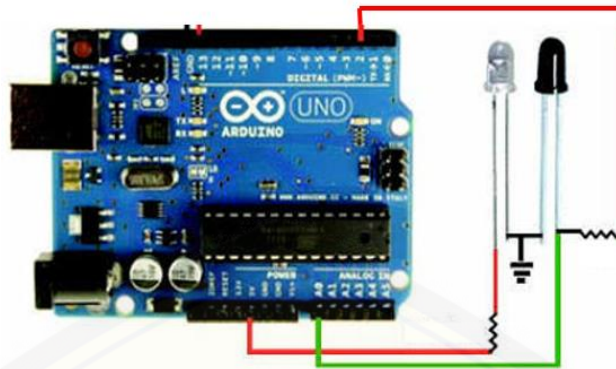
Pada perancangan desain sensor Ultrasonic HC-SR04 ini nantinya akan menggunakan lima buah sensor HC-SR04 yang akan digunakan untuk membantu pergerakan dari robot agar tidak menabrak dinding lintasan saat bergerak. Pada penempatannya pin *Echo* dan *Triger* dari sensor HC-SR04 akan ditancapkan kedalam pin analog ataupun digital pada Arduino sedangkan pin *VCC* dari sensor HC-SR04 akan ditancapkan ke pin *Vin* dari Arduino dan pin *GND* dari sensor HC-SR04 akan ditancapkan ke pin *GND* dari Arduino Berikut adalah desain perancangan lima sensor HC-SR04 kepada Arduino mega :



Gambar 3.5 perancangan desain sensor Ultrasonik HC-SR04 dengan Arduino

### 3.5.3 Perancangan desain sensor photodioda

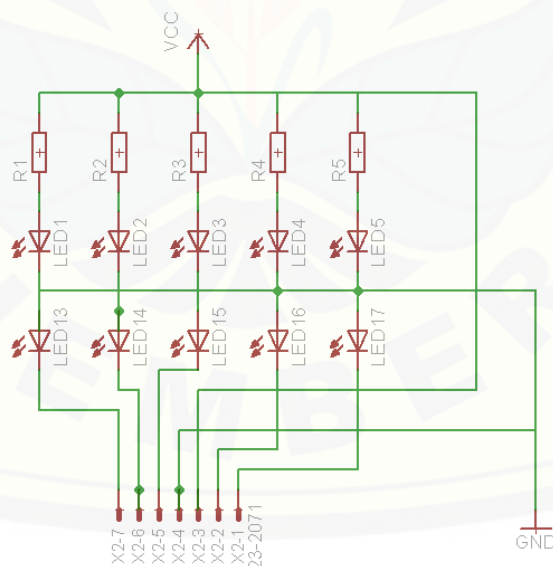
Sensor photodioda adalah sensor yang digunakan untuk mendeteksi garis putih yang berada pada setiap pintu masuk ruangan. Sensor ini bekerja dengan membaca perubahan cahaya yang dipengaruhi dari perubahan warna sehingga akan mempengaruhi nilai resistansinya yang kemudian akan dikonversi menjadi nilai ADC. Pada perancangannya sensor photodioda harus diseri dengan led yang berwarna putih dan resistor dengan nilai  $330\Omega$  pada kaki *negative* led serta resistor dengan nilai  $10K\Omega$  pada kaki *negative* dari sensor photodioda agar perubahan nilai ADC yang dihasilkan lebih besar.



Gambar 3.6 Perancangan sensor photodiode dengan Arduino

Sumber : (Startrobotics)

Perancangan pada robot sensor photodiode yang akan digunakan berjumlah lima buah sehingga harus digabungkan dengan membuat suatu rangkaian yang kompleks yang berisikan lima buah sensor photodiode, lima buah led berwarna putih, lima resistor dengan nilai  $330\Omega$  serta lima resistor dengan nilai  $10K\Omega$ . Berikut adalah skema dan *layout* rangkain dari sensor photodiode yang dibuat menggunakan software eagle :



Gambar 3.7 Skema rangkaian photodiode menggunakan *software eagle*

#### 3.5.4 Perancangan desain sensor Flame

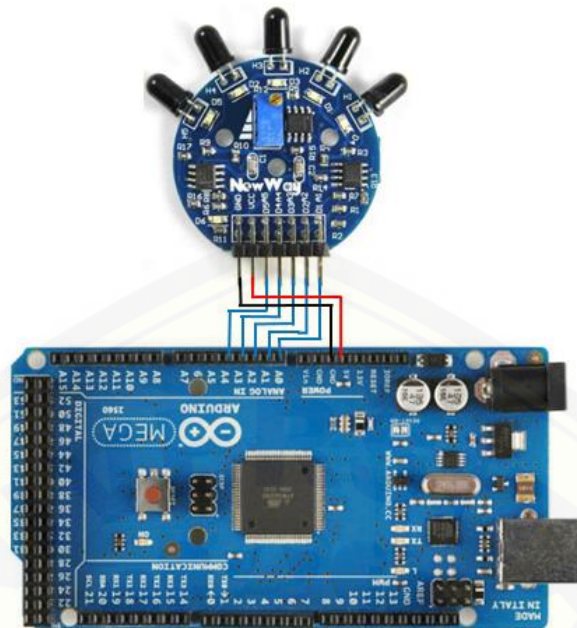
Sensor flame adalah sensor yang digunakan untuk mendeteksi keberadaan titik api pada suatu ruangan. Sensor ini bekerja dengan cara mendeteksi pancaran

sinar *infrared* dengan panjang rentang spectrum antara 760nm sampai dengan 1100 nm. Sensor flame dapat mendeteksi api sampai jarak 90cm, saat mendeteksi api maka nilai ADC dari sensor flame akan berubah, perubahan nilai ADC inilah yang akan menjadi acuan pada robot untuk mengetahui keberadaan titik api pada suatu ruangan.

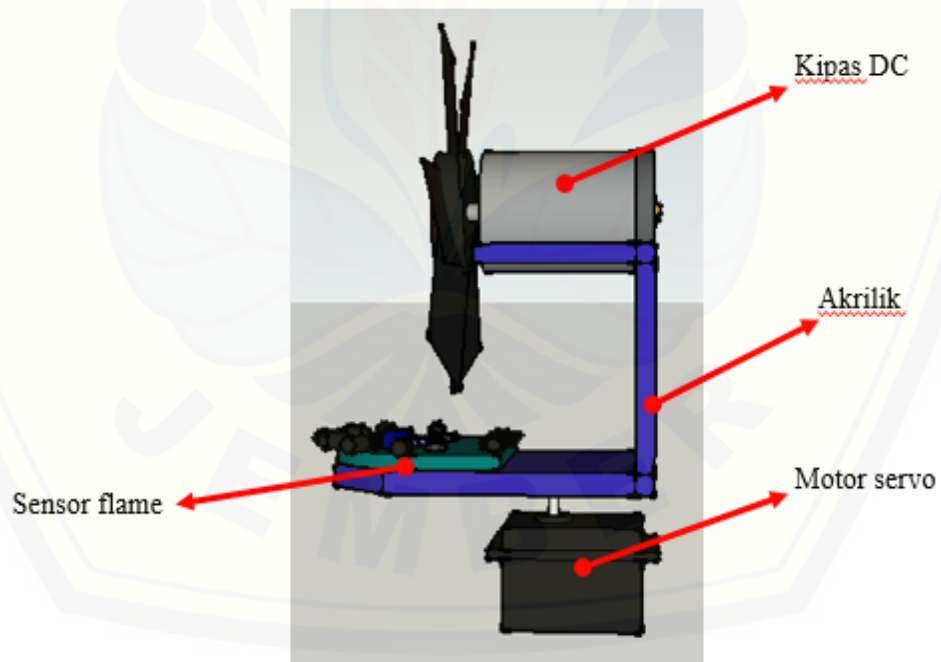
Sensor flame yang digunakan menggunakan lima buah sensor *infrarret* yang dirangkai sedemikian rupa sampai menjadi sebuah modul. Sensor flame yang digunakan memiliki empat belas pin yang antara lain adalah dua buah pin VCC, dua buah pin GND, lima buah pin *output* analog serta lima buah pin *output* digital. Pada perancangan pada robot pin *output* yang digunakan adalah pin analog karena dengan menggunakan pin analog peneliti dapat mengetahui perubahan nilai ADC yang terdeteksi oleh sensor flame sedangkan jika menggunakan pin *output* digital hanya bisa bekerja secara HIGH dan LOW saja atau ada atau tidak keberadaan titik api.

Sensor flame ini nantinya akan digabungkan dengan sebuah motor servo dan kipas DC yang akan diletakkan pada bagian depan bodi robot. Motor servo ini digunakan untuk membantu memutar sensor flame agar dapat melakukan scan pada ruangan untuk mendeteksi api sedangkan kipas DC digunakan untuk memadamkan api saat sensor flame mendeteksi keberadaan titik api. Pada perancangan robot pemadam api sensor flame juga memegang kendali yang cukup penting bagi sistem kendali robot, karena selama melakukan observasi ruangan jika sensor flame tidak dapat mendeteksi keberadaan titik api maka robot hanya akan berputar-putar mengelilingi ruangan saja. Oleh karena itu penempatan dan keakurasian sensor flame dalam mendeteksi api juga harus sangat diperhatikan agar robot dapat bekerja secara maksimal.





Gambar 3.8 Perancangan desain sensor flame lima *chanel* dengan Arduino



Gambar 3.9 Desain penempatan sensor flame dengan motor servo dan kipas DC

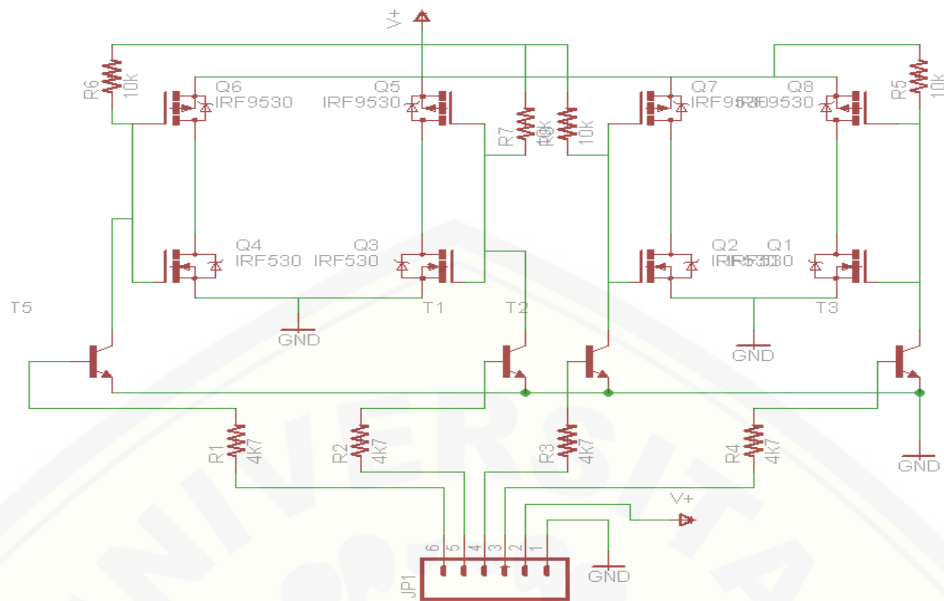


### 3.5.5 Perancangan desain *Driver* Motor H-Bridge Mosfet

*Driver* motor adalah rangkaian yang digunakan untuk mengatur arah putaran serta kecepatan putaran dari motor DC. Rangkaian ini sangat diperlukan dalam membuat robot beroda karna tanpa adanya *driver* motor maka robot tidak akan mampu melakukan manuver belok ke kiri dan ke kanan ataupun berjalan mundur. *Driver* motor sendiri memiliki banyak tipe, misalnya *driver* motor H-Bridge transistor, *driver* motor H-Bridge mosfet ataupun rangkaian *driver* motor yang menggunakan IC seperti L293 dan L298. Masing-masing jenis *driver* motor diatas dapat diaplikasikan pada robot beroda tergantung dari jenis motor DC yang digunakan dan besarnya arus yang dilewati.

Pada perancangan robot ini *driver* motor yang digunakan adalah H-Bridge mosfet karena rangkaian ini mampu dilewati arus yang cukup besar jika dibandingkan dengan transistor jenis lain yaitu lebih dari 1 amper, serta memiliki daya disipasi yang kecil sehingga dapat menghemat pemakaian daya. Rangkaian ini terdiri dari delapan buah mosfet dengan tipe IRF 9540 sebagai kanal P dan IRF 540 sebagai kanal N. Pada rangkaian *driver* motor H-Bridge mosfet ini satu buah motor DC akan diatur oleh dua buah mosfet IRF 9540 dan duan buah mosfet IRF 540.

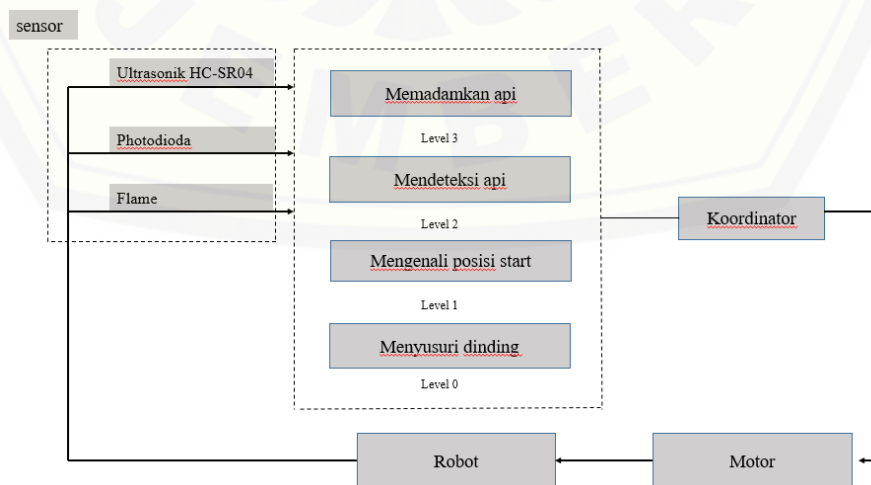
Pada pengaplikasiannya *driver* motor sangat rawan mengalami kerusakan apabila ada kesalahan dalam pemograman, kesalahan dalam pemasangan sumber tegangan yang berasal dari baterai ataupun terjadi *short* antar komponen sehingga mosfet akan terbakar dan jebol, oleh karena itu rangkaian *driver* motor harus dilindungi dari rangkaian lain ataupun pada rangkainnya dilapisi menggunakan selotip agar tidak terjadi *short* apabila tertumpuk oleh rangkaian lainnya. Berikut adalah rangkaian driver motor H-Bridge mosfet yang dibuat menggunakan *software* eagle :



Gambar 3.10 Rangkaian *driver* motor H-Bridge mosfet menggunakan *software eagle*

### 3.5.6 Perancangan sistem kontrol *Behavior Based Robot*

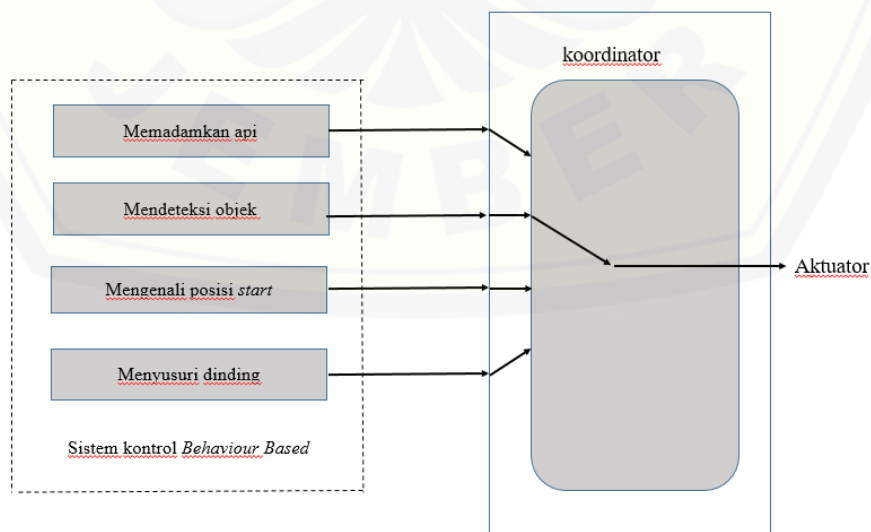
Sistem kontrol *Behavior Based Robot* adalah sebuah sistem kontrol yang dirancang dengan memberikan perilaku kepada robot. Sistem kontrol ini tidak terpaku kepada lingkungan yang statis sehingga robot yang dirancang menggunakan metode *behavior based* akan mampu beradaptasi terhadap lingkungannya. Pada perancangan sistem kontrol *behavior based* ini robot akan diberikan empat buah perilaku yang masing-masing perilaku akan dipegang oleh satu buah sensor yang akan memberikan aksi kepada aktuator.



Gambar 3.11 Perancangan sistem kontrol *behavior based robot*

Pada perancangan sistem kontrol *behavior based* diatas perilaku yang diutamakan adalah menyusuri dinding karena robot harus mampu menjelajahi arena tanpa harus menabrak dinding lintasan untuk mencari letak sumber api pada suatu ruangan secara cepat. Kemudian robot juga harus mampu mengenali posisi start untuk mencari jalan tercepat untuk kembali ke *home* setelah robot mampu memadamkan api. Perilaku ketiga yang harus ditanamkan kepada robot adalah mendeteksi objek yang berupa api kemudian perilaku selanjutnya adalah robot harus memadamkan sumber api yang berasal dari lilin. Dari keempat perilaku diatas aksi-aksi yang akan dilakukan oleh robot tergantung dari sensor mana yang akan bekerja.

Untuk dapat menghubungkan ke empat perilaku yang ditanamkan kepada robot maka akan ditambahkan satu bagian koordinator yang berfungsi untuk mengatur siapa yang akan memegang kendali dari akturator serta menentukan gerakan dari robot. Metode koordinator ini juga digunakan untuk meminimalisir kesalahan yang mungkin akan terjadi akibat terlalu banyaknya perilaku yang diberikan kepada robot. Pada sistem kontrol *behavior based* robot ini metode koordinator yang akan digunakan adalah *kompetitif* karena hanya akan mengizinkan satu buah perilaku saja yang akan memegang kendali dari aktuator dan mengatur pergerakan dari robot.

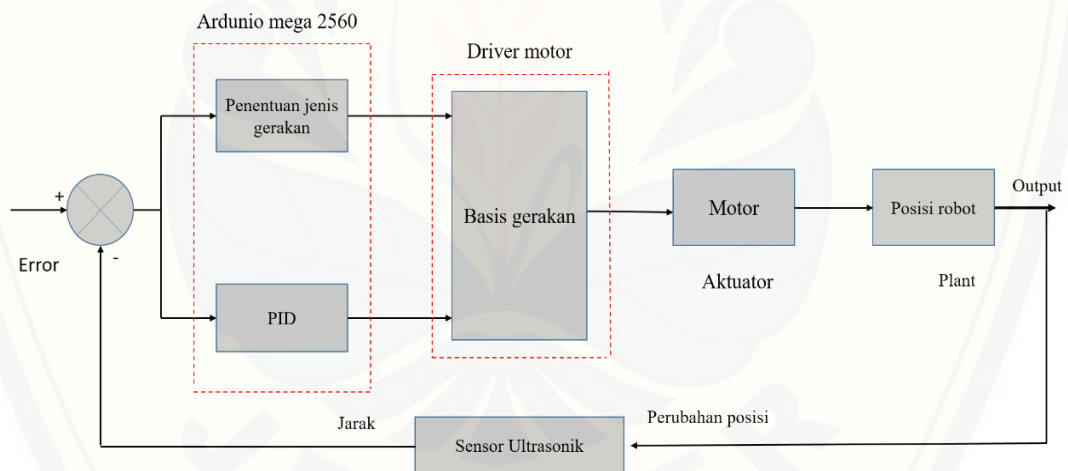


Gambar 3.12 Perancangan sistem koordinator *kompetitif behavior based robot*

### 5.5.7 Perancangan Sistem Kontrol PID

Pada penelitian ini sistem kontrol yang digunakan untuk manuver robot menggunakan Kontrol PID. Sistem kontrol PID dipilih karena merupakan sebuah sistem kontrol umpan balik yang mampu meminimalisir kesalahan sehingga akan mampu membuat navigasi robot menjadi lebih aman. Kontrol PID sendiri memiliki tiga parameter yang masing-masing memiliki karakteristik yang sangat berpengaruh terhadap respon output suatu sistem.

Kontrol *Proporsional* (P) digunakan untuk mempercepat respon suatu sistem, kontrol *Integrative* (I) digunakan untuk menghilangkan *offset* dan kontrol *Derivative* (D) digunakan untuk melakukan peredaman terhadap nilai osilasi. Ketiga parameter ini saling melengkapi satu sama lain sehingga kelemahan-kelemahan pada setiap parameter dapat dibantu dengan parameter lainnya.



Gambar 3.13 Perancangan diagram blok kontrol PID

Komponen dari diagram blok kontrol PID diatas adalah :

- 1) Set poin : 18 cm dari dinding
- 2) Kontroler : Arduino mega
- 3) Aktuator : 2 motor DC
- 4) *Feedback* : sensor HC-SR04
- 5) Plant : posisi robot
- 6) *Output* : perubahan posisi (jarak)

Pada Gambar 3.13 diatas adalah gambar rancangan kontrol PID yang akan diterapkan pada robot beroda pemadam api. Dimana set point yang digunakan sebagai acuan adalah pembacaan jarak robot dari dinding, *outputnya* adalah perubahan posisi dari robot dan *feedbacknya* adalah pembacaan sensor HC-SR04. Set point yang berupa pembacaan jarak antara robot dengan dinding akan dibandingkan terlebih dahulu dengan pembacaan jarak robot saat ini, dari perbandingan tersebut maka akan menghasilkan nilai *error* dimana dari nilai *error* tersebut yang akan menentukan jenis gerakan robot.

Pada setiap gerakan ini nilai PWM pada motor sudah ditentukan agar gerakan pada robot sesuai dengan yang diinginkan. Pergerakan robot tentunya juga akan mempengaruhi posisi robot sehingga posisi robot terhadap dinding juga berubah, perubahan inilah yang akan dibaca oleh sensor HC-SR04 yang nantinya akan menjadi *feedback* untuk dibandingkan dengan nilai set point guna menentukan besarnya nilai *error* yang akan menjadi masukan untuk perhitungan nilai PID.

Untuk dapat mengimplementasikan kontrol PID digital kedalam Arduino mega maka kontrol PID analog harus dikonversi dahulu kedalam bentuk digital. Berikut adalah bentuk matematis dari kontrol PID :

$$u(t) = K_P e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (3.1)$$

Dengan  $K_i = \frac{1}{\tau_i}$  dan  $K_d = \tau_d$ , bentuk integral dan diferensial dalam ditulis dalam bentuk diskrit seperti persamaan berikut :

$$\int_0^t e(t)dt \approx \sum_0^k e(k) \quad (3.2)$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \quad (3.3)$$

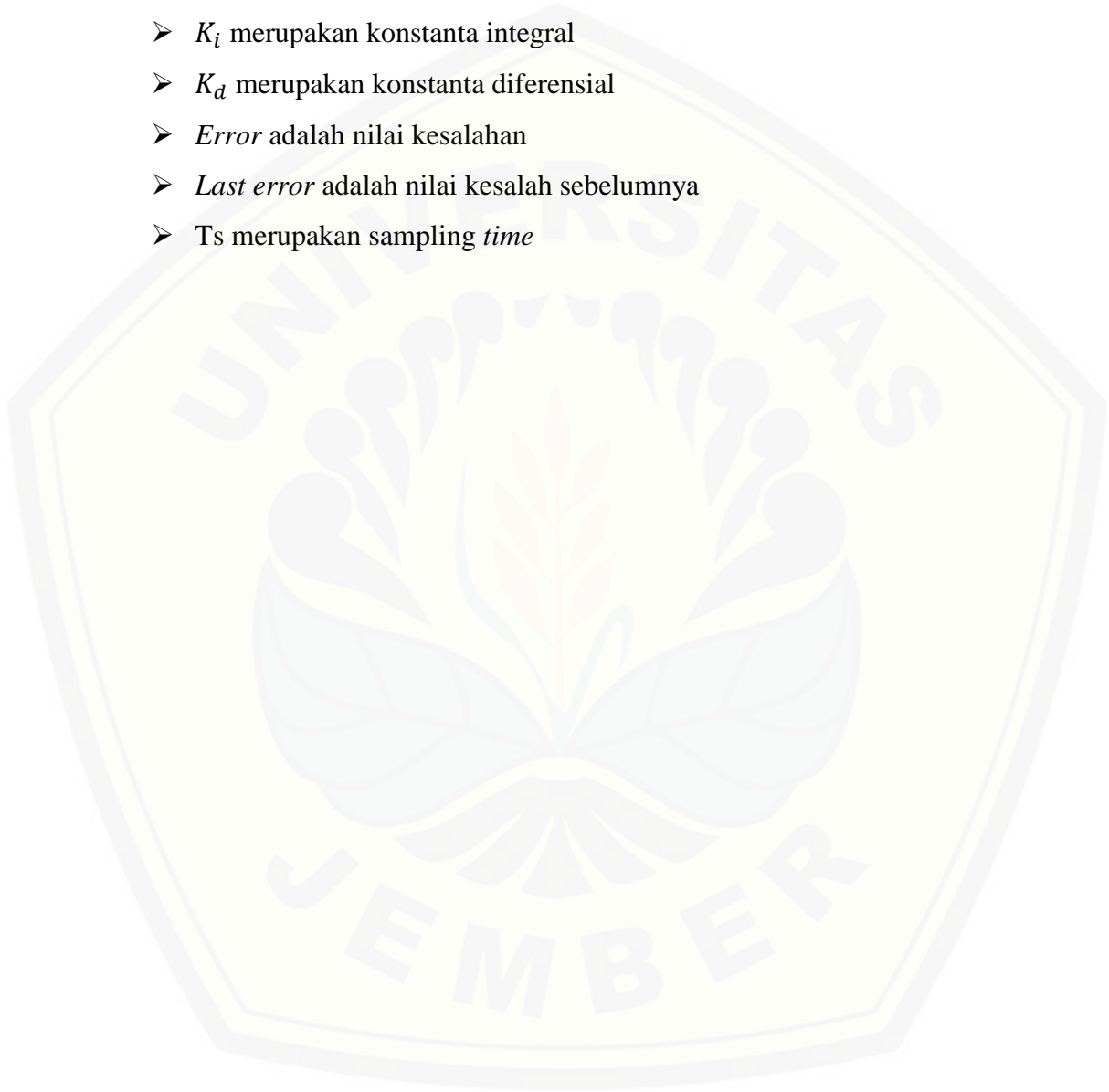
Sehingga diperoleh dalam bentuk persamaan kontrol PID diskrit sebagai berikut :

$$u(k) = K_P e_k + K_i T \sum_0^k e_k + \frac{1}{T} K_d (e_k - e_{k-1}) \quad (3.4)$$

atau :  $u(k) = K_p \times \text{error} + K_i \times (\text{error} + \text{last error}) \times T_s + \frac{K_d}{T_s} \times (\text{error} - \text{last error})$

Dimana :

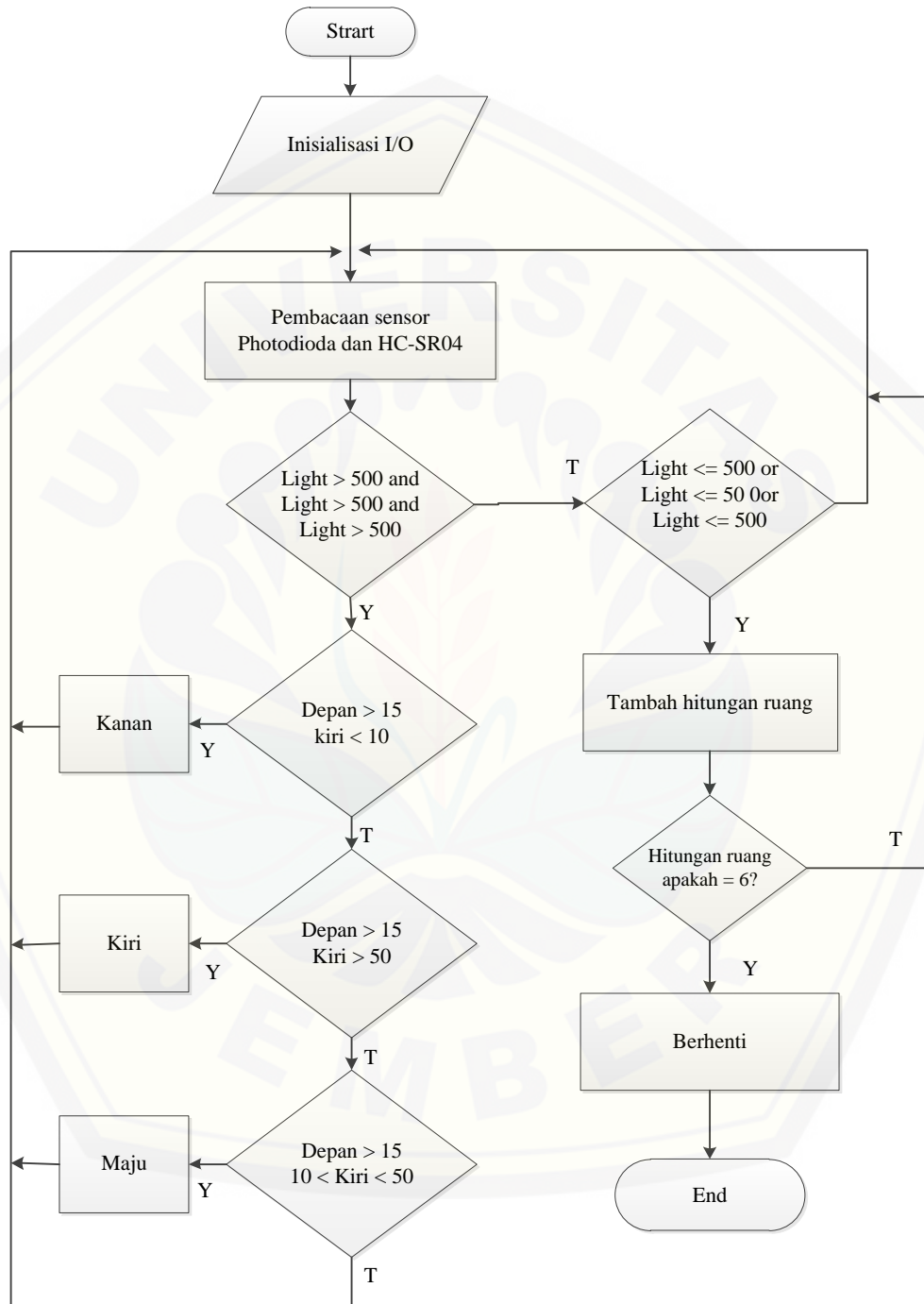
- $K_p$  merupakan konstanta proposional
- $K_i$  merupakan konstanta integral
- $K_d$  merupakan konstanta diferensial
- *Error* adalah nilai kesalahan
- *Last error* adalah nilai kesalah sebelumnya
- $T_s$  merupakan *sampling time*





3.6 Diagram Alir (Flowchart)

6.6.1 Diagram alir gerakan robot



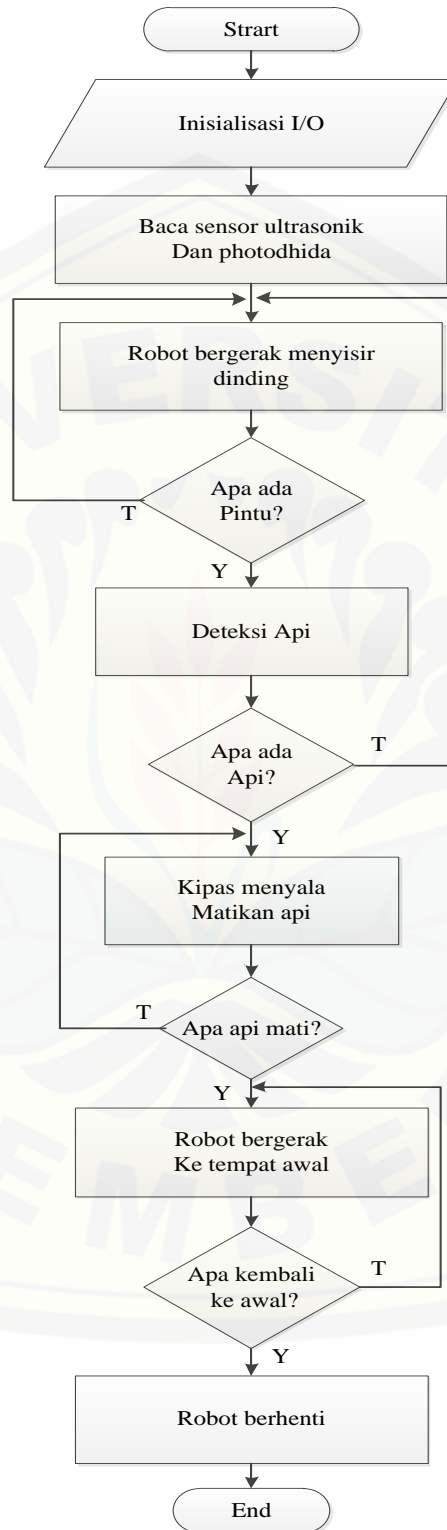
Gambar 3.14 Diagram Alir gerakan robot

Pada gambar 3.14 diatas adalah gambar diagram alir agar robot dapat bergerak menelusuri ruangan dengan baik maka digunakan teknik *wall following* yaitu robot berjalan dengan menggukan dinding sebagai acuan. Dinding yang digunakan sebagai acuan adalah dinding sebelah kanan robot. Dalam mendeteksi dinding robot juga disertai dengan pendeteksian lantai lintasan. Jika kelima sensor photodiode pada robot masih membaca nilai adc diatas 500 maka robot berjalan mengikuti dinding.

Robot akan berjalan maju dengan menjaga jaraknya antara 10 sampai 50 cm dari dinding kiri dan sensor depan mendeteksi lebih dari 15 cm dari dinding. Robot akan berbelok ke kanan apabila sensor depan mendeteksi jarak lebih dari 15 cm dari dinding dan sensor kiri mendeteksi jarak kurang 10 cm dari dinding kiri. Robot akan berbelok ke kiri apabila sensor depan membaca jarak lebih dari 15 cm dan sensor kiri membaca jarak lebih dari 50 cm dari dinding.

Jika selama melakukan pergerakan salah satu sensor photodiode mambaca nilai ADC lebih dari 500 maka robot akan terus bergerak menyisiri dinding namun jika salah satu sensor photodiode membaca nilai ADC kurang dari 500 maka robot akan berhenti selama 3 detik untuk melakukan scan ruangan, dan robot akan menghitung setiap ruangan yang ada sebagai cara untuk kembali ke *home*. Jika robot sudah membaca setiap ruangan sebanyak 6 kali maka robot akan berhenti di *home*.

## 6.6.2 Flowchart jalannya sistem robot secara keseluruhan



Gambar 3.15 Diagram Alir Kerja Sistem Keseluruhan

Gambar 3.15 diatas adalah gambar diagram alir dari sistem kerja robot secara keseluruhan, dimana pada saat pertama kali diaktifkan robot akan mengenali *home* melalui pembacaan dari sensor photodiode, kemudian robot akan mulai berjalan dengan cara menyisir dinding kiri menggunakan sensor ultrasonic. Pada saat bertemu ruangan robot akan mendeteksi ada sumber api apa tidak, jika tidak ada maka robot akan kembali berjalan menyisir dinding tapi jika terdapat sumber api pada ruangan tersebut maka, robot akan mendekati api dan kipas meyal untuk memadamkan api. Saat api sudah padam maka robot akan kembali berjalan menyisir dinding untuk mencari *home*.

### 3.7 Pengujian Data

Pada bab pengujian data ini akan menjelaskan mengenai beberapa hal yang harus diuji guna mengetahui performa dari kerja robot yang diantaranya adalah :

#### 3.7.1 Pengujian sensor ultrasonic HC-SR04

Pengujian pada sensor ultrasonic HC-SR04 ini dilakukan berdasarkan kemampuan keakurasian sensor dalam mendeteksi suatu objek untuk membantu robot dalam bernavigasi secara aman. Pengujian ini nantinya akan dilakukan dengan cara membandingkan hasil pembacaan sensor terhadap objek dengan pengukuran langsung menggunakan penggaris.

Pada pengujiannya akan dilakukan sebanyak 20 *sample* data dengan kelipatan 5 yang akan dilakukan mulai dari jarak 5cm – 100cm. Hasil pembacaan sensor ini akan dibandingkan dengan pengukuran jarak menggunakan penggaris sehingga akan didapatkan nilai *error* persen. Nilai error persen ini didapatkan dari rumus, harga mutlak dari hasil pengukuran penggaris dikurangi dengan hasil pembacaan sensor dibagi dengan nilai pengukuran penggaris dikalikan 100%.

#### 3.7.2 Pengujian sensor Flame

Pengujian pada sensor flame ini dilakukan berdasarkan kemampuan keakurasian robot dalam mendeteksi keberadaan letak titik api pada suatu ruangan. Pengujian ini nantinya akan dilakukan dengan cara mengamati kemampuan sensor

dalam mendeteksi api dari jarak tertentu dengan melihat perubahan nilai ADC sensor.

Pada pengujian ini akan dilakukan sebanyak 16 *sample* data dengan kelipatan 5 yang akan dimulai dari jarak 5cm – 80cm. pada pengujiannya robot akan diletakkan pada posisi yang berbeda-beda sehingga dari kelima sensor infrared yang terdapat pada satu modul sensor flame akan menunjukkan pembacaan nilai ADC yang berbeda. Dari perbedaan inilah yang akan dilihat sensor pada posisi mana yang dapat bekerja secara akurat dalam mendeteksi api saat robot berada pada posisi tertentu kemudian dicari nilai rata-ratanya dengan cara menjumlahkan semua hasil pembacaan nilai ADC sensor kemudian dibagi 5.

### 3.7.3 Pengujian sensor Photodiode

Pengujian pada sensor photodiode ini dilakukan berdasarkan kemampuan robot dalam mendeteksi warna yang berada pada lantai lintasan. Pada lintasan sendiri terdapat dua warna yaitu putih dan hitam. Pembacaan sensor ini sangat dibutuhkan ketelitian agar robot dapat bekerja sesuai dengan yang diinginkan.

Pengujian ini nantinya akan dilakukan dengan cara mengamati perubahan nilai ADC pada kelima sensor yang akan dilakukan selama 60 detik pada masing-masing warna untuk mengetahui kemampuan kelima sensor photodiode dalam membedakan warna putih dan hitam, kemudian dicari nilai rata-ratanya dengan cara menjumlahkan semua hasil pembacaan nilai ADC sensor kemudian dibagi 5.

### 3.7.4 Pengujian sistem kontrol *Behavior Based Robot*

Pengujian pada sistem kontrol *behavior based robot* ini dilakukan untuk mengamati setiap perilaku yang diberikan kepada robot apakah sudah sesuai dengan yang diinginkan atau masih kurang. Perilaku yang diberikan sendiri ada empat yaitu menyusuri dinding, mendeteksi garis, mengenali objek serta mampu memadamkan api.

Pengujian pada perilaku menyisir dinding ini, nantinya akan dilakukan dengan cara mengamati posisi robot ketika sensor ultrasonic HC-SR04 membaca jarak tertentu. Kemudian perilaku mendeteksi garis akan dilakukan pengujian dengan cara mengamati kondisi robot setelah memasuki ruangan. Sedangkan untuk perilaku mendeteksi objek serta memadamkan api akan dilakukan pengujian dengan cara mengamati aksi robot saat mendeteksi api.

### 3.7.5 Pengujian sistem kontrol PID

Pengujian pada sistem kontrol PID ini dilakukan untuk mengamati jalannya robot selama menyusuri lorong. Pada sistem kontrol PID sendiri terdapat empat parameter yaitu proportional (P), integrative (I) dan derifative (D) yang masing-masing memiliki peran penting dalam menyusun suatu sistem kontrol PID.

Pengujian pada sistem kontrol PID ini nantinya akan dilakukan den dua tahap yaitu mencoba-coba merubah nilai parameter dari masing-masing konstanta untuk mendapatkan nilai mana yang tepat digunakan pada navigasi robot beroda. Untuk pengujian yang kedua akan dilakukan dengan cara membandingkan nilai set point yang telah ditentukan dengan nilai error yang dihasilkan dari pembacaan sensor Ultrasonik HC-SR04.

### 3.7.6 Pengujian robot secara keseluruhan

Pada pengujian robot secara keseluruhan ini akan dilakukan dengan mengamati serta menghitung waktu yang dibutuhkan oleh robot dalam menyelesaikan misi. Pengujian ini bertujuan untuk mengetahui persentase keberhasilan robot dalam menyelesaikan misi yang diberikan. Pada pengujian ini alat bantu yang digunakan berupa *stopwatch* yang akan digunakan untuk mengetahui berapa lama waktu yang dibutuhkan oleh robot dari awal berjalan sampai kembali ke *home*.

Pengujian ini akan dilakukan sebanyak 4 kali pengujian, dimana pada setiap pengujian robot akan melakukan percobaan menyelesaikan misi sebanyak 5 kali. Pengujian ini dimulai dari robot bergerak dari titik *start* sampai mendeteksi



objek dan memadamkan api kemudian kembali lagi ke *home*. Pada setiap pengujiannya letak titik api dan posisi home akan berbeda-beda.

### 3.8 Hasil Perancangan Desain Robot

Pada bab hasil perancangan desain robot ini akan menjelaskan hal yang sudah dicapai dari perancangan desain robot sebelumnya.



Gambar 3.16 hasil perancangan desain robot tampak depan



Gambar 3.17 hasil perancangan desain robot tampak samping



Gambar 3.18 hasil perancangan desain robot tampak atas

Hasil dari perancangan robot ini dapat dilihat pada Gambar 3.16, Gambar 3.17 dan Gambar 3.18 yang memperlihatkan bentuk robot yang diambil dari beberapa sudut pandang yaitu tampak depan, tampak samping, serta tampak atas. Robot ini memiliki ukuran panjang 25 cm, lebar 18 cm dan tinggi 19 cm. Bodi robot ini seluruhnya terbuat dari bahan akrilik dengan ketebalan 3mm. robot ini terdiri dari motor DC dan roda, sensor ultrasonic HC-SR04, sensor flame, rangkaian sensor photodiode, kipas DC, motor servo, driver motor, baterai *Li-po*, LCD dan mikrokontroler Arduino mega.

Pada bagian atas robot dapat dilihat terdapat sebuah LCD yang digunakan untuk mengamati hasil pembacaan sensor ultrasonic. Pada bagian belakang robot terdapat saklar DC yang digunakan untuk mengatur hidup dan matinya robot, dua buah motor DC yang digunakan sebagai penggerak robot serta satu buah baterai *Li-po* yang digunakan sebagai *supply* dari robot agar dapat bergerak.

Pada bagian samping robot terdapat dua buah sensor ultrasonic HC-SR04 yang berfungsi untuk membaca jarak antara robot dan dinding yang digunakan untuk membantu navigasi robot. Sedangkan pada bagian depan robot terdapat satu buah sensor ultrasonic HC-SR04 yang digunakan untuk mendeteksi objek yang

berada pada bagian depan robot, satu buah sensor flame yang digunakan untuk mendeteksi keberadaan letak sumber api serta satu buah kipas DC yang digunakan untuk memadamkan api.



## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah melakukan penelitian ini, didapatkan beberapa kesimpulan dari hasil percobaan sebagai berikut :

1. Dengan menggunakan metode *behavior based* persentase keberhasilan robot dari keempat percobaan dalam menyelesaikan misinya berturut-turut adalah pada pengujian pertama 100% dengan rata-rata waktu 72.8 detik, pada pengujian kedua 60% dengan rata-rata waktu 81.6 detik, pada pengujian ketiga 80% dengan rata-rata waktu 83.25 detik dan pada pengujian keempat 80% dengan rata-rata waktu 86.75 detik
2. Kontrol PID yang digunakan dengan menggunakan nilai  $K_p$  18 respon yang dihasilkan cukup cepat dengan nilai rise time 17.6 detik, kemudian pada saat ditambahkan nilai  $K_d$  7 respon yang dihasilkan memiliki nilai osilasi yang kecil dan memiliki nilai rise time 16.6 detik serta dapat mencapai steady state pada waktu 74.4 detik, kemudian pada saat ditambahkan nilai  $K_i$  8 respon yang dihasilkan sudah dapat mencapai steady state dengan rise time 15.2 detik dan setting time 67.2 detik.
3. Koordinasi setiap perilaku yang ditanamkan kepada robot dapat bekerja dengan baik karena selama menjalankan misi robot tidak pernah mengalami kesalahan dalam pengambilan keputusan untuk setiap keadaan. Hal ini dapat dibuktikan pada setiap pengujian *behavior* robot dapat menyelesaikan misi dengan persentase rata-rata keberhasilan 80%

## 5.2 Saran

Dikarenakan pentingnya pengembangan dan penelitian lanjutan yang terkait dengan penelitian ini, penulis memberikan beberapa saran antara lain sebagai berikut :

1. Pada robot beroda pemadam api ini masih perlu diperbaiki pada sensor yang digunakan untuk mendeteksi api, karena sensor yang digunakan masih memiliki spek yang kurang baik sehingga harus sangat diperhatikan penempatan jarak antara lilin dan pintu masuk ruangan, intensitas cahaya dari luarpun juga harus diperhatikan karena bisa mempengaruhi pembacaan dari sensor. Sehingga sangat disarankan untuk menggunakan sensor yang lebih baik seperti sensor UV Tron.



**DAFTAR PUSTAKA**

- Fahmizal. 2011. *Driver Motor DC pada Robot Beroda dengan Konfigurasi H-Bridge MOSFET*. <https://fahmizaleeits.wordpress.com/tag/mosfet-kanal-p/>. (Diakses Tanggal 03 Maret 2016).
- Famosastudio. 2013. *Arduino Mega 2560*. <http://blog.famosastudio.com/2013/09/produk/arduino-mega-2560/531>. (Diakses Tanggal 03 Maret 2016).
- Trianes, Agustian, dan Andik Yulianto. 2015. Implementasi Behavior Based Control dan PID Pada Robot Vacuum Cleaner. 1-6.
- Iqbal, Muhammad Rully, Dikairono, Budi dan Sardjono Tri Arief.. 2012. Implementasi Sistem Navigasi Behavior Based Robotic dan Kontrol Fuzzy Pada Manuver Robot Cerdas Pemadam Api. 1-8.
- Sanjaya, Mada. 2016 Panduan Praktis Membuat Robot Cerdas Menggunakan Arduino dan Matlab.
- Rusli, Mohammad. 1997. Sistem Kontrol Kedua, Malang. Teknik Elektro Universitas Brawijaya.
- Rameli, Mochammad, Rusdhianto E, Djoko Susilo.1996. Sistem Pengaturan
- Gunterus, Frans. 1994. Sistem Pengendali Proses
- Johnson, Curtis. 1988. Process Control Instrumentation Technology.
- Ogata, Katsuhiko. 1991. Teknik Kontrol Automatik. Terjemahan : Ir. Edi Laksono



# LAMPIRAN

## **Listing Program Arduino Mega 2560**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include <Servo.h>
LiquidCrystal_I2C lcd(0x3F,16,2);
Servo swerve;

const int m1a=7,m1b=8,m2a=9,m2b=10;
const int trig[5] = {25,29,33,37,51};
const int echo[5] = {23,27,31,35,53};
const int pinS[5] = {A15,A14,A13,A12,A11};
const int pinKalibrasi = 19;
const int pinApi[5] = {A10,A7,A6,A5,A4};
const int pinServo = 47;
const int pinKipas = 49;

int sBawah[5] = {500,500,500,500,500};
int ref[5] = {400,400,400,400,400};
int logSens[5] = {0,0,0,0,0};
int sApi[5] = {500,500,500,500,500};
int refApi = 600;
int kodeApi = 0;
int kodeSiapScan = 0;
```

```
int      logApi[5]= {0,0,0,0,0};

long     jarak[5],jarakLama[5];

int      kecepatan = 200, kp = 18, ki = 8, kd = 15 ;

int      kodeBehavior = 0;

int      ruang = 0;

void setup() {

    pinMode(pinKipas,OUTPUT);digitalWrite(pinKipas,LOW);

    Serial.begin(9600);

    Serial.println("SERIAL SIAP");

    //=====LCD=====

    lcd.init();lcd.init();

    lcd.backlight();

    Serial.println("LCD SIAP");

    //=====

    for(int i = 0; i<=4; i++){

        pinMode(trig[i],OUTPUT);

        pinMode(echo[i],INPUT_PULLUP);

    }

    pinMode(m1a,OUTPUT);pinMode(m1b,OUTPUT);

    pinMode(m2a,OUTPUT);pinMode(m2b,OUTPUT);

    pinMode(19,INPUT_PULLUP);

    Serial.println("PIN SIAP");

    if(digitalRead(pinKalibrasi)==LOW){kalibrasi();}
```

```
else                                     {ambilRef(); }
}

void loop() {
  //while(1)padamkanApi();
  switch(kodeBehavior){
    case 0 : cariRuang(); break;
    case 1 : cariApi(); break;
    case 2 : padamkanApi(); break;
    case 3 : pulang(); break;
    default : break;
  }
}

//-----SENSOR-----
void ambilJarak(int x){
  long duration, jar;
  digitalWrite(trig[x], HIGH);
  delayMicroseconds(x);
  digitalWrite(trig[x], LOW);
  duration = pulseIn(echo[x], HIGH, 5000);
  jar = duration/29/2;
  if(jar == 0) jar = 101;
  //if(jar == jarakLama[x])
  jarak[x] = jar;
}
```



```
//jarakLama[x] = jar;

Serial.println(String(jarak[x])+" ");

delay(10);

}

void ambilBawah() {

    for(int i = 0; i<5; i++){

        sBawah[i] = analogRead(pinS[i]);

        logSens[i] = 1-(sBawah[i] / ref[i]);

        lcd.setCursor(i+5,1);lcd.write((1-logSens[i])*210+45);

    }

    Serial.println(String(sBawah[0])+" "+String(sBawah[1])+" "+

        String(sBawah[2])+" "+String(sBawah[3])+" "+

        String(sBawah[4]));

}

void kalibrasi() {

    Serial.println("KALIBRASI MULAI");

    int batasAtas[5] = {0,0,0,0,0};

    int batasBawah[5] = {1023,1023,1023,1023,1023};

    lcd.setCursor(0,0);lcd.println("KALIBRASI... ");

    lcd.setCursor(0,1);lcd.println("PROSES %");

    for(int j=0; j<400; j++){

        for(int i=0; i<5; i++){

            sBawah[i] = analogRead(pinS[i]);

            if(sBawah[i] > batasAtas[i] )batasAtas[i] = sBawah[i];
```

```
        if(sBawah[i] < batasBawah[i])batasBawah[i] = sBawah[i];
    }
    delay(10);
    lcd.setCursor(12,1); lcd.print(j/4);
    Serial.println(j);
}
for(int i =0; i<5; i++){
    ref[i] = (batasAtas[i] - batasBawah[i])/2 + batasBawah[i];
    EEPROM.write(i,ref[i]/4);
}
Serial.println("KALIBRASI SELESAI");
lcd.setCursor(0,0);lcd.print("SELESAI... ");
delay(1000);
Serial.println( "REFERENSI : " + String(ref[0])+" "+
                String(ref[1]) +" "+ String(ref[2])+" "+
                String(ref[3]) +" "+ String(ref[4]));
lcd.setCursor(0,0);lcd.print(" ");
lcd.setCursor(0,1);lcd.print(" ");
lcd.setCursor(0,1);lcd.print(ref[0]);
lcd.setCursor(0,0);lcd.print(ref[1]);
lcd.setCursor(6,0);lcd.print(ref[2]);
lcd.setCursor(13,0);lcd.print(ref[3]);
lcd.setCursor(13,1);lcd.print(ref[4]);
delay(2000);lcd.clear();
```

```
}  
  
void ambilRef(){  
  
    Serial.println("TIDAK KALIBRASI");  
  
    for(int i = 0; i<5; i++){  
  
        ref[i] = EEPROM.read(i)*4;  
  
    }  
  
}  
  
//-----BEHAVIOR-----  
  
void susurKanan(){  
  
    ambilJarak(4);  
  
    ambilJarak(3);  
  
    ambilJarak(2);  
  
    lcd.setCursor(0,1);lcd.print("R. ") ;lcd.setCursor(2,1);  
    lcd.print(ruang);  
  
  
    lcd.setCursor(13,1);lcd.print("    ") ;lcd.setCursor(13,1);  
    lcd.print(jarak[4]);  
  
    lcd.setCursor(13,0);lcd.print("    ") ;lcd.setCursor(13,0);  
    lcd.print(jarak[3]);  
  
    lcd.setCursor(6,0); lcd.print("    ") ;lcd.setCursor(6,0);  
    lcd.print(jarak[2]);  
  
  
    if      (jarak[2] < 15)                {motor(-150,150);  
    delay(400);}  
  
    else if (jarak[4] < 10 || jarak[3] < 15) {motor(100,175);  
    }  
}
```

```
    else if (jarak[4] > 50)                {motor(200,70);
}
    else if (jarak[4] > 15)                {motor(200,150);
}
    else                                    {motor(200,200);
}
}
}

void susurKiri(){
    ambilJarak(0);
    ambilJarak(1);
    ambilJarak(2);

    lcd.setCursor(13,1);lcd.print("R. ");lcd.setCursor(15,1);
    lcd.print(ruang);

    lcd.setCursor(0,1);lcd.print("    ");lcd.setCursor(0,1);
    lcd.print(jarak[0]);

    lcd.setCursor(0,0);lcd.print("    ");lcd.setCursor(0,0);
    lcd.print(jarak[1]);

    lcd.setCursor(6,0);lcd.print("    ");lcd.setCursor(6,0);
    lcd.print(jarak[2]);

    if      (jarak[2] < 15)                {motor(150,-150);
delay(500);}
    else if (jarak[0] < 10 || jarak[1] < 15) {motor(175,100);
}
    else if (jarak[0] > 50)                {motor(70,200);
}
    else if (jarak[0] > 15)                {motor(150,200);
}
```

```
    else                                     {motor(200,200);
}
}

void susurKananPID(){
    ambilJarak(4);
    ambilJarak(3);
    ambilJarak(2);
    lcd.setCursor(12,1);lcd.print("    ") ;lcd.setCursor(13,1);
    lcd.print(jarak[4]);
    lcd.setCursor(12,0);lcd.print("    ") ;lcd.setCursor(13,0);
    lcd.print(jarak[3]);
    lcd.setCursor(6,0); lcd.print("    ") ;lcd.setCursor(6,0);
    lcd.print(jarak[2]);
    if      (jarak[2] < 15) {motor(-150,150); delay(500);}
    else if (jarak[3] < 15) {motor(100,175); }
    else if (jarak[4] > 50) {motor(200,70); }
    else if (jarak[4] > 20) {motor(200,150); }
    else                                     {int error = 10-jarak[4];
    pid(error,0);lcd.setCursor(12,1);lcd.write(255);}
}

void susurKiriPID(){
    ambilJarak(0);
    ambilJarak(1);
    ambilJarak(2);
    lcd.setCursor(0,1);lcd.print("    ") ;lcd.setCursor(0,1);
    lcd.print(jarak[0]);
```



```
    lcd.setCursor(0,0);lcd.print("    ");lcd.setCursor(0,0);
    lcd.print(jarak[1]);

    lcd.setCursor(6,0);lcd.print("    ");lcd.setCursor(6,0);
    lcd.print(jarak[2]);

    if      (jarak[2] < 15) {motor(150,-150); delay(500);}
    else if (jarak[1] < 15) {motor(175,150);   }
    else if (jarak[0] > 50) {motor(70,200);   }
    else if (jarak[0] > 20) {motor(150,200);   }

    else      {int error = 10-jarak[0];
    pid(0,error);lcd.setCursor(3,1);lcd.write(255);}
}

void scanApiJauh(){
    swerve.attach(pinServo);

    kodeApi = 0;

    for( int s = 0; s<= 180; s+= 2){

        swerve.write(s);

        for(int i = 0; i< 5; i++){

            sApi[i] = analogRead(pinApi[i]);

            logApi[i] = sApi[i] / refApi;

            lcd.setCursor(i+5,0); lcd.write(logApi[i]*210+45);

            if(logApi[i] >= 1){kodeApi = 1; break;}

        }

        if(kodeApi >= 1)break;

        Serial.println(String(sApi[0])+" "+String(sApi[1])+"
        "+String(sApi[2])+" "+String(sApi[3])+" "+String(sApi[4]));
```

```
    delay(20);
}
if(kodeApi >= 1)goto akhirApi;

for( int s = 180; s>= 0; s-= 2){
    swerve.write(s);
    for(int i = 0; i< 5; i++){
        sApi[i] = analogRead(pinApi[i]);
        logApi[i] = sApi[i] / refApi;
        lcd.setCursor(i+5,0); lcd.write(logApi[i]*210+45);
        if(logApi[i] >= 1){kodeApi = 1; break;}
    }
    if(kodeApi >= 1)break;

    Serial.println(String(sApi[0])+" "+String(sApi[1])+"
"+String(sApi[2])+" "+String(sApi[3])+" "+String(sApi[4]));

    delay(20);
}
akhirApi :
swerve.write(90);delay(500);
swerve.detach();
}

void cariApi(){
    swerve.attach(pinServo);
    swerve.write(90);
    telusurRuang();
```

```
ambilBawah();  
ambilJarak(2);  
if(jarak[2] <= 15){  
    kodeApi = 0;  
    for(int i = 2; i<= 2; i++){  
        sApi[i] = analogRead(pinApi[i]);  
        logApi[i] = sApi[i] / (refApi+150);  
        lcd.setCursor(i+5,0); lcd.write(logApi[i]*210+45);  
        if(logApi[i] >= 1){kodeApi = 1; break;}  
    }  
    if(kodeApi == 1){  
        kodeBehavior = 2;  
    }  
}  
if(logSens[2] ){  
    motor(0,0);  
    motor(-100,-130); delay(250);  
    motor(130,-160); delay(550);  
}  
lcd.clear();  
}  
  
void telusurRuang(){  
    ambilJarak(0);  
    ambilJarak(1);
```

```
    ambilJarak(2);

    lcd.setCursor(0,1);lcd.print("    ");lcd.setCursor(0,1);
    lcd.print(jarak[0]);

    lcd.setCursor(0,0);lcd.print("    ");lcd.setCursor(0,0);
    lcd.print(jarak[1]);

    lcd.setCursor(6,0);lcd.print("    ");lcd.setCursor(6,0);
    lcd.print(jarak[2]);

    if      (jarak[2] < 15)                {motor(100,-100);
delay(700);}

    else if (jarak[0] < 10 || jarak[1] < 15) {motor(125,50);
}

    else if (jarak[0] > 50)                {motor(50,150);
}

    else if (jarak[0] > 15)                {motor(50,130);
}

    else                                    {motor(100,100);
}
}

void telusurLorong(){
    susurKanan();
}

void cariRuang(){
    telusurLorong();

    ambilBawah();

    if((logSens[0] && logSens[1]) || logSens[2] || (logSens[3]
&& logSens[4]) ){

        if(kodeSiapScan==0){
```

```
        kodeSiapScan = 1;
        motor(100,100);delay(500);
    }
else{
    motor(0,0);
    lcd.clear();
    lcd.setCursor(0,1); lcd.print("      STOP      ");
    scanApiJauh();
    if(kodeApi == 1){
        kodeApi = 0;

while(logSens[0]||logSens[1]||logSens[2]||logSens[3]||logSens[
4]){

        motor(100,100);delay(500);
        ambilBawah();
    }
    kodeBehavior = 1;
}
else{
    motor(-180,-150); delay(300);
    motor(30,150); delay(300);
    motor(-180,180); delay(300);
}
}
}
lcd.clear();
```

```
        ruang++;
    }
}

void padamkanApi() {
    swerve.attach(pinServo);
    swerve.write(0);
    motor(0,0);
    delay(250);
    int sudut1 = 10, sudut2=170, maxAdc = 0;
    for(int i=10; i<=170; i+=2) {
        swerve.write(i);
        int adc = analogRead(pinApi[i]);
        if(adc>maxAdc) {maxAdc = adc; sudut1 =
i; Serial.println(sudut1);}
        delay(20);
    }
    for(int i=170; i>=10; i-=2) {
        swerve.write(i);
        int adc = analogRead(pinApi[i]);
        if(adc>maxAdc) {maxAdc = adc; sudut2 =
i; Serial.println(sudut2);}
        delay(20);
    }

    int sudut = (max(sudut1, sudut2) -
min(sudut1, sudut2)) / 2 + min(sudut1, sudut2);

    delay(500);
}
```



```
Serial.println(sudut);

swerve.write(sudut);

lcd.setCursor(0,0);lcd.print(sudut1);
lcd.setCursor(0,1);lcd.print(sudut2);
lcd.setCursor(10,1);lcd.print(sudut);

delay(1000);

lcd.setCursor(0,0);
lcd.print("  MATIKAN  API  ");
digitalWrite(pinKipas,HIGH);
delay(3000);
digitalWrite(pinKipas,LOW);
delay(1000);
digitalWrite(pinKipas,HIGH);
delay(3000);
digitalWrite(pinKipas,LOW);
delay(1000);
digitalWrite(pinKipas,HIGH);
delay(3000);
digitalWrite(pinKipas,LOW);
motor(-100,-150);
delay(1000);

kodeBehavior = 3;

}

void pulang(){
```

```
    ruang = 8 - (2*(ruang-1));  
    int ruangAwal = ruang;  
    while(ruang > 0){  
        telusurLorong();  
        ambilBawah();  
        if((logSens[0] && logSens[1]) || logSens[2] || (logSens[3]  
&& logSens[4]) ){  
            if(ruang==ruangAwal||ruang==1){  
                motor(100,100);  
                delay(500);  
            }  
            else{  
                motor(-180,-150); delay(300);  
                motor(30,150); delay(300);  
                motor(-180,180); delay(300);  
                ruang--;  
            }  
            ruang--;  
        }  
        lcd.clear();  
    }  
    for(int aa = 0; aa <= 10; aa++){  
        susurKanan();  
    }  
    motor(0,0);while(1);
```

```
}  
  
void pid(int e1, int e2){  
    int pwmKi = kecepatan - e1*kp + e2*kp;  
    int pwmKa = kecepatan + e1*kp - e2*kp;  
    motor(pwmKi,pwmKa);  
}  
  
void motor(int ki, int ka){  
    if (ki<-255) ki=-255;  
    else if (ki>255) ki=255;  
    if (ka<-255) ka=-255;  
    else if (ka>255) ka=255;  
  
    if(ki>=0) {digitalWrite(m1a,LOW);analogWrite(m1b,ki); }  
    else {digitalWrite(m1b,LOW);analogWrite(m1a,0-ki);}  
    if(ka>=0) {digitalWrite(m2a,LOW);analogWrite(m2b,ka); }  
    else {digitalWrite(m2b,LOW);analogWrite(m2a,0-ka);}  
}
```