



**PENGEMBANGAN SISTEM ROBOT BERKAKI ENAM  
(HEXAPOD) MENGGUNAKAN METODE *BEHAVIOR*  
*BASED CONTROL* PADA KONTES ROBOT  
INDONESIA PEMADAM API BERKAKI**

**SKRIPSI**

Oleh

**Ahmad Wahyu Tri Utama**

**NIM 141910201030**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS JEMBER**

**2018**



**PENGEMBANGAN SISTEM ROBOT BERKAKI ENAM (*HEXAPOD*)  
MENGUNAKAN METODE *BEHAVIOR BASED CONTROL*  
PADA KONTES ROBOT INDONESIA  
PEMADAM API BERKAKI**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi syarat – syarat  
untuk menyelesaikan Program Studi Strata 1 Teknik Elektro (S1)  
dan mencapai gelar Sarjana Teknik (ST)

Oleh

**Ahmad Wahyu Tri Utama**

**NIM 141910201030**

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

## PERSEMBAHAN

Dengan puja dan puji syukur kepada Allah SWT. Berkat rahmat dan karunia-Mu akhirnya Karya Tulis Ilmiah yang sederhana ini dapat terselesaikan tepat pada waktunya. Karya Tulis Ilmiah ini ku persembahkan untuk:

1. Ibunda Nurhayati dan ayahanda Abdul Hamim yang telah memberikan kasih sayang, segala dukungan, serta do'a yang tiada henti untuk kesuksesan anaknya. Ucapan terimakasih saja takkan pernah cukup untuk membalas kebaikan beliau, karena itu terimalah persembahan bakti dan cinta ku untuk Ibu Bapakku;
2. Semua sanak saudaraku kakak Eko Indah Astuti dan Ita Dwi Puspita yang selalu setia menyemangati serta memberikan dukungan baik dari segi mental maupun materi terhadap semua aktifitas perkuliahan dari awal sampai saat ini;
3. Semua teman – teman pondok bambu dan saudara – saudara keluarga penangan yang selalu memberi motivasi terutama Tante martini yang selalu sabar memotivasi selama berada di Jember dan bimbingannya;
4. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
5. Semua Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Jember yang senantiasa memberikan ilmunya. Semoga ilmu yang Bapak/Ibu berikan bermanfaat dan barokah untukku dan untuk pribadi masing-masing serta menjadi amalan penolong Bapak/Ibu kelak;
6. Teman - teman asisten laboratorium yang selalu membuat suasana saat mengerjakan menjadi semangat dan keluarga besar ROBOTIKA UNEJ yang selalu menyemangati.
7. Saudara-saudaraku Teknik Elektro 2014 Universitas Jember;

## MOTTO

Sesungguhnya sesudah kesulitan itu ada kemudahan  
(terjemahan Surat Al-Insyirah Ayat 6)\*<sup>1</sup>

Cobalah untuk tidak menjadi orang sukses,  
tapi menjadi orang lebih bernilai. \*\*<sup>2</sup>

Kesuksesan bukan akhir, kegagalan bukan hal yang fatal,  
hal itu adalah keberanian untuk melanjutkan  
apa yang penting. \*\*\*<sup>3</sup>

---

<sup>1</sup> Departemen Agama Republik Indonesia.1998. *Alquran dan Terjemahannya*.Semarang:PT. Kumudasmoro Grafindo.

<sup>2</sup> Irfan Syahputra, “*Kumpulan Motto Albert Einstein*”(<http://irfansyahp.blogspot.co.id/2013/09/kumpulan-motto-albert-einstein.html?m=1>, Diakses pada 19 Maret 2018, 2018)

<sup>3</sup> Antonius Halim, “*Winston Churchill Archive*”(<http://katabijak.com/tag/winston-churchill.html>, Diakses pada 19 Maret 2018, 2018)

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Wahyu Tri Utama

NIM : 141910201030

Menyatakan dengan sesungguhnya bahwa proyek akhir yang berjudul: *“Pengembangan Sistem Robot Berkaki Enam (Hexapod) Menggunakan Metode Behavior Based Control Pada Kontes Robot Indonesia Pemadam Api Berkaki”* adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi mana pun serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 29 Juni 2018

Yang menyatakan,

Ahmad Wahyu Tri Utama  
NIM 141910201030

**SKRIPSI**

PENGEMBANGAN SISTEM ROBOT BERKAKI ENAM (*HEXAPOD*)  
MENGUNAKAN METODE *BEHAVIOR BASED CONTROL* PADA KONTES  
ROBOT INDONESIA PEMADAM API BERKAKI

Oleh

Ahmad Wahyu Tri Utama

NIM 141910201030

Pembimbing :

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

Dosen Pembimbing Anggota : Khairul Anam, S.T., M.T., Ph.D.

## PENGESAHAN

Skripsi berjudul “*Pengembangan Sistem Robot Berkaki Enam (Hexapod) Menggunakan Metode Behavior Based Control Pada Kontes Robot Indonesia Pemadam Api Berkaki*” telah diuji dan disahkan pada:

Hari : Jumat  
Tanggal : 29 Juni 2018  
Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Sumardi, S.T., M.T.  
NIP 19670113 199802 1 001

Anggota II,

Dodi Setiabudi, S.T., M.T.  
NIP 19840531 200812 1 004

Anggota I,

Khairul Anam, ST.,MT., Ph.D.  
NIP 19780405 200501 1 002

Anggota III,

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

Mengesahkan  
Dekan,

Dr. Ir. Entin Hidayah, M.U.M.  
NIP 19661215 199503 2 001

## RINGKASAN

**Pengembangan Sistem Robot Berkaki Enam (Hexapod) Menggunakan Metode Behavior Based Control Pada Kontes Robot Indonesia Pemadam Api Berkaki**, Ahmad Wahyu Tri Utama, 141910201030;2018: 72 halaman; Jurusan Teknik Elektro Universitas Jember.

Robot merupakan seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan ataupun dengan program yang sudah didefinisikan terlebih dahulu seperti menanamkan perilaku terhadap robot. Pengembangan robot juga tidak asing bagi mahasiswa elektro khususnya pada konsentrasi Kontrol dan Instrumentasi, salah satunya robot *Hexapode* pemadam api.

Robot *Hexapode* merupakan robot yang bergerak menggunakan kaki enam sesuai dengan nama dari robot itu sendiri yang biasanya bergerak menggunakan motor servo yang telah disusun sesuai dengan *design*. Pergerakan robot sesuai dengan banyaknya servo yang digunakan seperti pada penelitian ini menggunakan dua servo dengan n-DOF yang digunakan sebanyak 2 DOF. Robot dengan kaki dua DOF pergerakan kakinya dapat bergerak maju – mundur dan naik – turun. Robot *Hexapode* itu sendiri banyak digunakan perguruan tinggi untuk mengikuti pada Kontes Robot Pemadam Api ( KRPAI ) walaupun ada yang menggunakan hanya dengan 4 kaki dari kebanyakan peserta banyak yang menggunakan kaki dengan jumlahnya 6 kaki.

Pergerakan kaki – kaki yang terkadang tersendat – sendat dan tidak lancar bahwasanya sangat mengganggu pergerakan robot itu sendiri untuk berjalan mengitari maupun untuk pergerakan ketika melakukan suatu kondisi perilaku yang diinginkan robot. Penggunaan arduino sebagai mikrokontroler sangatlah membantu terbentuknya robot *Hexapode* ini yang mana robot sebelumnya hanya menggunakan satu mikrokontroler. Pada penelitian ini menggunakan dua mikrokontroler yang saling berhubungan dengan memanfaatkan fitur *binner HIGH* dan *LOW* pada kondisi pergerakan robot dari komunikasi binner ini robot akan mengirimkan data tertentu sesuai dengan kondisi yang diinginkan dan data tidak boleh ada yang



tertukar atau pun tidak dikenali atau tidak diterima. Maka diperlukanya sistem komunikasi data yang tidak mudah tertukar yaitu komunikasi biner karena memiliki dua basis data 0 dan 1 dalam penelitian menggunakan 4 bit.

Pada penelitian ini robot *Hexapode* menggunakan metode *Behavior Based Control* untuk *control high level* dengan ini robot dapat melakukan tugasnya dengan baik untuk tujuan dari memadamkan api seperti robot yang digunakan dalam penelitian ini. Metode *Behavior Based Control* merupakan metode yang terdiri dari perilaku – perilaku yang sudah tertanam yang nantinya perilaku – perilaku itu melakukan koordinasi yang nantinya robot akan melakukan pengambilan keputusan sesuai keadaan yang didapatkan dari Bergeraknya robot itu sendiri pada lintasan. Perilaku robot tertanam pada dua arduino yang mana arduino pertama digunakan untuk perilaku robot melihat kondisi sekitar dan untuk arduino yang kedua digunakan untuk perilaku pergerakan robot dari maju, mundur, dan berbelok.

Dari penelitian yang dilakukan di Laboratorium Elektronika dan Terapan Jurusan Teknik Elektro Universitas Jember dapat diketahui bahwasanya program robot merupakan metode *Behavior Based Control* yang masih sederhana, sehingga masih terdapat kesalahan yang terjadi ketika perpindahan antara *behavior* robot masih kurang halus. Terlihat pada data percobaan memindai boneka yang masih memiliki kesalahan 40 % dari 5 kali percobaan.

## PRAKATA

*Bismillahirrohmanirrohim*

Puji syukur ke hadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga laporan skripsi yang berjudul “*Pengembangan Sistem Robot Berkaki Enam (Hexapod) Menggunakan Metode Behavior Based Control Pada Kontes Robot Indonesia Pemadam Api Berkaki*” dapat terselesaikan dengan baik. Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata Satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Terselesaikannya laporan skripsi ini tidak terlepas dari bantuan berbagai pihak, oleh karena itu disampaikan ucapan terima kasih kepada:

1. Allah Swt yang telah melimpahkan rahmat dan rizki-Nya serta memberi kelancaran dan kemudahan sehingga terselesaikannya skripsi ini.
2. Nabi Muhammad SAW yang telah membawa kita ke peradaban manusia yang lebih baik.
3. Ibu dan Bapak serta saudara-saudaraku tercinta, atas jasa-jasanya, kesabaran, do’a, dan tidak pernah lelah dalam mendidik dan memberi cinta yang tulus dan ikhlas kepada penulis semenjak kecil.
4. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
5. Bapak Dr. Ir. Bambang Sri Kaloko, S.T., M.T. selaku Ketua Jurusan Fakultas Teknik Elektro Universitas Jember.
6. Bapak Dedi Kurnia Setiawan, S.T., M.T. selaku Ketua Prodi S1 Fakultas Teknik Elektro Universitas Jember.
7. Bapak Sumardi, S.T., M.T. selaku dosen pembimbing utama dan Bapak Khairul Anam, ST., MT., Ph.D. selaku dosen pembimbing anggota yang telah meluangkan waktu dan pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan proyek akhir ini.

8. Seluruh Dosen yang ada di Fakultas Teknik khususnya Teknik Elektro beserta karyawan.
9. Keluarga kecil di Jember tante Martini tercinta yang telah banyak memberikan dorongan, semangat, kasih sayang, dan bantuan baik secara moral maupun materi demi lancarnya penyusunan skripsi ini
10. Keluarga besar Teknik Elektro seperjuangan khususnya angkatan 2014 KETEK UJ, terimakasih atas dukungan dan motivasi yang kalian berikan selama menjalani masa kuliah sampai terlaksananya skripsi ini.
11. Semua pihak yang tidak dapat saya sebutkan satu persatu, terima kasih atas dukungan dan motivasi kalian dalam penyusunan skripsi ini.

Penulis menyadari bahwa sebagai manusia biasa tidak terlepas dari keterbatasan, yang biasanya akan mewarnai kadar ilmiah dari skripsi ini. Oleh karena itu penulis selalu terbuka terhadap masukan dan saran dari semua pihak yang sifatnya membangun untuk mendekati kesempurnaan. Tidak lupa penulis menyampaikan permohonan maaf yang sebesar-besarnya jika terdapat kesalahan dan kekeliruan. Akhir kata penulis berharap laporan ini dapat memberikan manfaat bagi pembaca dan dapat menjadi bahan acuan yang bermanfaat di kemudian hari.

Jember, 29 Juni 2018

Penulis

## DAFTAR ISI

PENGEMBANGAN SISTEM ROBOT BERKAKI ENAM ( <i>HEXAPOD</i> ) MENGUNAKAN METODE <i>BEHAVIOR</i>	i
<i>BASED CONTROL</i> PADA KONTES ROBOT INDONESIA PEMADAM API BERKAKI	i
PERSEMBAHAN	ii
MOTTO	iii
PERNYATAAN	iv
SKRIPSI	v
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
<b>BAB I PENDAHULUAN</b>	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
<b>BAB 2 TINJAUAN PUSTAKA</b>	4
2.1 Robot Berkaki.....	4
2.2 Arduino Mega 2560.....	5
2.3 Penggunaan Sensor Ultrasonik PING.....	8
2.4 Motor Servo.....	8
2.5 Sensor Api.....	10
2.6 Pompa Wiper ( <i>Extinguisher</i> ).....	12
2.7 Algoritma penggunaan <i>Behavior based control hexapod robot</i> .....	12
2.8 <i>Invers Kinematics</i> .....	13
<b>BAB 3 METODE PELAKSANAAN DATA</b>	17
3.1 Tempat dan Waktu Penelitian.....	17

3.2	Ruang Lingkup Kegiatan .....	17
3.3	Jenis dan Sumber Data.....	18
3.4	Metode Pengumpulan Data.....	18
3.5	Perancangan Alat .....	19
3.5.1	Perancangan <i>Design</i> Mekanik Robot.....	19
3.5.2	Perancangan Sistem Elektronika Robot.....	20
3.6	Sistem Kendali Robot <i>Behavior Based Control</i> .....	27
3.6.1	<i>Behavior Based Control</i> .....	27
3.6.2	Bagian <i>Behavior Based Control</i> .....	28
3.6.2.1	<i>Behavior</i> Navigasi Robot .....	28
3.6.2.2	<i>Behavior</i> Mendeteksi Garis .....	30
3.6.2.3	<i>Behavior</i> Memindai Api .....	31
3.6.2.4	<i>Behavior</i> Mendeteksi Boneka.....	32
3.6.2.5	<i>Behavior</i> Memadamkan Api.....	33
3.7	<i>Flowchart Behavior</i> Robot Keseluruhan .....	35
3.8	Pengujian .....	37
3.8.1	Pengujian Perangkat keras .....	37
3.8.1.1	Ultrasonik <i>PING</i> .....	37
3.8.1.2	Sensor <i>Sharp</i> .....	37
3.8.1.3	Sensor Api <i>UVTron</i> .....	37
3.8.1.4	Sensor Garis.....	37
3.8.2	Pengujian Invers Kinematik pada robot <i>hexapode</i> .....	38
3.8.3	Pengujian <i>Behavior Based Control</i> .....	38
3.8.4	Pengujian Keseluruhan.....	38
3.9	Hasil Perancangan Alat.....	38
<b>BAB 4 HASIL DAN ANALISA PENGUJIAN</b>		<b>42</b>
4.1	Pengujian Data Sensor Ultrasonik <i>PING</i> .....	42
4.2	Pengujian sensor Inframerah <i>Sharp</i> .....	44
4.3	Pengujian Sensor Api <i>UVTron</i> .....	47
4.3.1	Pengujian Jarak pendeteksian Sensor Api <i>UVTron</i> .....	47
4.3.2	Pengujian Lebar Sudut Pendeteksian Sensor Api <i>UVTron</i> .....	48
4.4	Pengujian Sensor Garis .....	49
4.5	<i>Invers Kinematics</i> .....	51
4.5.1	<i>Inverse Kinematics Jalan</i> .....	53

4.6 Behavior Hexapode Robot.....	55
4.6.1 Behavior Robot Keluar Ruangan .....	55
4.6.2 Behavior Robot Keluar dari Ruang 1.....	56
4.6.3 Behavior Robot Keluar dari Ruang 2.....	57
4.6.4 Behavior Robot Keluar dari Ruang 3.....	58
4.6.5 Behavior Robot Keluar dari Ruang 4.....	59
4.6.6 Behavior Robot Mendeteksi dan Behavior Robot Memadamkan Api..60	
4.6.6.1 Behavior Robot memadamkan Api dari Ruang 1 menuju ruang 3	60
4.6.6.2 Behavior Robot memadamkan Api dari Ruang 2 menuju ruang 4	61
4.6.6.3 Behavior Robot memadamkan Api dari Ruang 3 menuju ruang 1	62
4.6.6.4 Behavior Robot memadamkan Api dari Ruang 4 menuju ruang 2	63
4.6.7 Pengujian Behavior Hexapode Robot .....	64
4.6.8 Pengujian Koordinasi Behavior Hexapode Robot .....	67
4.6.9 Perbandingan Waktu Robot Sebelumnya Dengan Sekarang .....	69
<b>BAB 5 PENUTUP</b>	<b>70</b>
5.1 KESIMPULAN.....	70
5.2 SARAN.....	70
LAMPIRAN	72

## DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Mega 2560.....	6
Tabel 4.1 Hasil Pengujian Data sensor Ultrasonik PING .....	42
Tabel 4.2 Pembagian range jarak pembacaan masing – masing sensor.....	43
Tabel 4.3 Data Pengukuran Jarak sensor Sharp dengan jarak penggaris .....	44
Tabel 4.4 Perilaku Robot hexapode ketika mendeteksi boneka.....	46
Tabel 4.5 Pengujian jarak pendeteksian api sensor UVTron .....	47
Tabel 4.6 Pengujian Sudut pendeteksian api sensor UVTron.....	48
Tabel 4.7 Data Pengujian Sensor Garis Lampu dimatikan .....	49
Tabel 4.8 Data Pengujian Sensor Garis Pada Saat Lampu Lintasan dihidupkan...50	
Tabel 4.9 Data Logika Sensor Mnenemukan Garis .....	50
Tabel 4.10 Pengujian <i>forward kinematics</i> .....	53
<b>Tabel 4.11</b> Pengujian Jarak Langkah Kaki Robot.....	54
Tabel 4.12 Pengujian Behavior Robot pencapaian waktu tiap ruang .....	64
Tabel 4.13 Pengujian Behavior Robot Waktu Pemadaman Api .....	64
Tabel 4.14 Pengujian Behavior Robot Memadamkan Api Sudut Pada Api .....	65
Tabel 4.15 Pengujian Behavior Robot Memindai Boneka.....	65
Tabel 4.16 Pengujian Behavior Robot Mendeteksi Garis .....	66
Tabel 4.17 Pengujian Data koordinasi behavior .....	68
Tabel 4.18 Tabel Perbandingan Robot Sebelumnya dan Sekarang .....	69

## DAFTAR GAMBAR

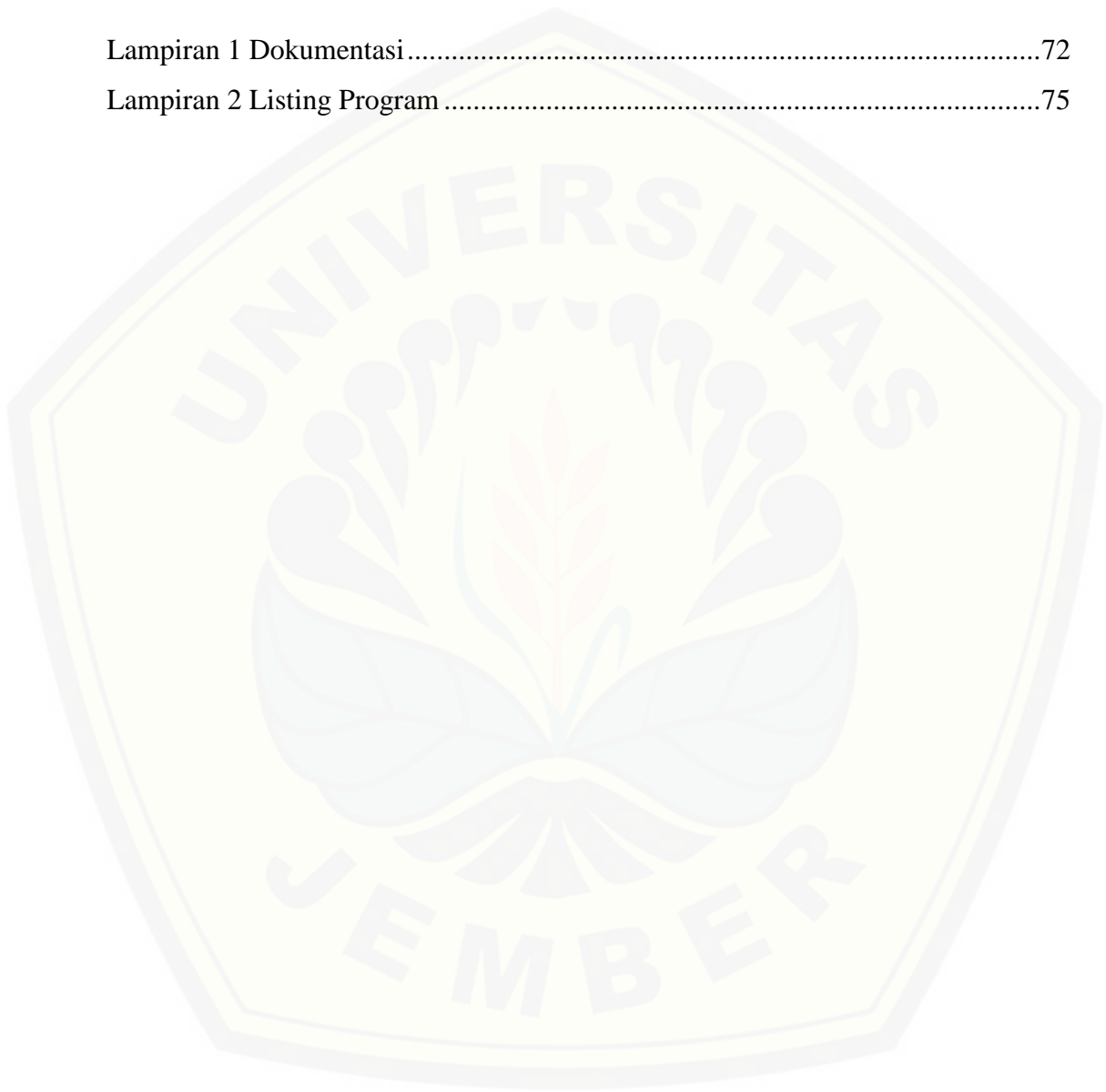
Gambar 2.1 Robot Berkaki .....	4
Gambar 2.2 Arduino Mega 2560 .....	5
Gambar 2.3 Sensor Ultrasonik PING .....	8
Gambar 2.4 Motor Servo.....	9
Gambar 2.5 Pulsa Kendali Motor Servo .....	10
Gambar 2.6 Sensor Api UVTron .....	11
Gambar 2.7 Tabung UV Flame Detector Tabung sensor UV Flame Detector dan sudut respon sensor secara horizontal.....	11
Gambar 2.8 Penyemprot Air Pemadam Api .....	12
Gambar 2.9 Struktur koordinasi <i>Behavior</i> .....	13
Gambar 2.10 Model Kontrol Kinematika Robot.....	14
Gambar 2.11 Grafik Representasi Lengan Robot Berkaki.....	15
Gambar 2.12 Grafik Referensi untuk Mencari sebuah Grafik Referensi untuk Mencari $\theta_1$ .....	16
Gambar 3.1 Mekanik robot pemadam api berkaki.....	19
Gambar 3.2 Design Kaki.....	19
Gambar 3.3 Desain Peletakan Sensor pada Robot .....	20
Gambar 3.4 Blok diagram untuk Sistem Elektronika Robot Berkaki.....	20
Gambar 3.5 Rangkaian Sensor Ultrasonik Pada Robot Berkaki.....	21
Gambar 3.6 Rangkaian Sensor Sharp pada robot berkaki.....	22
Gambar 3.7 Rangkaian Sensor Sharp pada robot berkaki.....	23
Gambar 3.8 Gambar Rangkain Extinguisher .....	24
Gambar 3.9 Komonikasi antar mikrokontroler .....	25
Gambar 3.10 Gambar Rangkain Sensor Garis .....	26
Gambar 3.11 Algoritma <i>behaviour based control</i> robot KRPAI berkaki.....	27
Gambar 3.12 <i>Flowchart Behavior Navigasi</i> Telusur Kanan.....	28
Gambar 3.13 <i>Flowchart Behavior</i> mendeteksi garis .....	30
Gambar 3.14 <i>Flowchart Behavior</i> memindai api.....	31
Gambar 3.15 <i>Flowchart Behavior</i> mendeteksi boneka.....	32



Gambar 3.16 <i>Flowchart Behavior</i> memadamkan api .....	33
Gambar 3.17 <i>Flowchart Behavior</i> Robot Berkaki Pemadam Api .....	35
Gambar 3.18 Robot tampak dari atas .....	39
Gambar 3.19 Robot tampak dari Samping .....	39
Gambar 3.20 Robot tampak Depan .....	40
Gambar 3.21 Arena Robot pemadam api untuk berkaki .....	41
Gambar 4.1 Pembacaan Sensor Sharp .....	45
Gambar 4.2 Pembagian Zona Pembacaan Sensor .....	46
Gambar 4.3 Peletakan Sensor Garis .....	49
Gambar 4.4 Kaki Robot, (1) coxa, (2) femur (3) <i>tibia</i> .....	51
Gambar 4.5 Grafik Representasi untuk mencari sudut jalan kaki robot .....	53
Gambar 4.6 <i>Behavior</i> Robot Keluar Ruangan .....	55
Gambar 4.7 <i>Behavior</i> Robot dari Ruang 1 .....	56
Gambar 4.8 <i>Behavior</i> Robot dari ruang 2 .....	57
Gambar 4.9 <i>Behavior</i> Robot dari ruang 3 .....	58
Gambar 4.10 <i>Behavior</i> Robot dari Ruang 4 .....	59
Gambar 4.11 <i>Behavior</i> Robot dari ruang dari ke ruang 3 .....	60
Gambar 4.12 <i>Behavior</i> Robot dari Ruang 2 menuju Ruang 4 .....	61
Gambar 4.13 Pergerakana Robot dari Ruang 3 menuju Ruang 1 .....	62
Gambar 4.14 <i>Behavior</i> Robot dari Ruang 4 menuju Ruang 2 .....	63
Gambar 4.15 Koordinasi <i>Behavior</i> Hexapode Robot .....	67

**DAFTAR LAMPIRAN**

Lampiran 1 Dokumentasi.....	72
Lampiran 2 Listing Program.....	75



## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan robot yang sangat pesat beberapa tahun terakhir membuat ilmu robotika menyebar kesegala bidang, seperti robot *humanoid*, robot sepak bola, robot berkaki dan semakin banyak robot yang dikembangkan untuk membantu pekerjaan manusia. Robot bukanlah benda yang diam melainkan bergerak sesuai apa yang diprogramkan oleh pembuatnya. Sebagai contoh robot pemadam api berkaki yang bergerak untuk mencari api dan memadamkannya.

Pada kontes robot pemadam api Indonesia biasanya robot yang digunakan pada pemadam api berkaki tempatnya atau arena yang digunakan sudah ditentukan dengan peraturan – peraturan yang dibuat oleh panitia penyelenggara. Dalam aturan, terdapat penghalang atau dengan 4 keadaan yang digunakan dalam perlombaan dengan posisi start di salah satu ruangan yang diacak oleh panitia. Selain itu, penempatan api diacak juga dan diposisikan pada ruangan sesuai hasil undian.

Robot cerdas pemadam api merupakan sebuah robot yang menggunakan *Behaviorkaki* yang mensimulasikan rumah untuk menemukan api dan memadamkan api yang berupa lilin. Pada KRPAI (Kontes Robot Pemadam Api) 2017, level yang digunakan hanya level 2. Level dua ini terdiri dari 4 kemungkinan posisi start dan posisi api yang diacak oleh juri ketika akan melakukan perlombaan (kontes robot indonesia.org).

Dalam pembuatan *robot mobile* ini. Hal yang mendasar yaitu robot dapat mengenali objek – objek disekitarnya. Penggunaan *Behavior Based Control* dapat mendukung *Behavior* Robot yang dapat menghindari halangan dengan baik. (Gitakarma, Nurhayata, Elektronika, Teknik, & Ganesha, 2013).

Beberapa permasalahan yang pernah terjadi dalam perlombaan kontes robot indonesia pemadam api divisi berkaki yaitu algoritma untuk berjalan yang masih sangat lambat yang mampu memakan waktu yang begitu lama untuk berjalan menelusuri ruangan perlombaan. Oleh karena itu diperlukan metode yang tepat

untuk mengatasi ini agar didapat robot dapat berjalan dengan cepat untuk *Behavior* kakinya ketika melakukan perjalanan didalam arena lomba.

Menurut peneliti dari universitas surabaya yang menggunakan robot beroda dimana pemrosesan *sensor* mulai dari pengaktifan *sensor*, pengambilan data dilakukan bila diperlukan saja. Semisal ketika ingin mendeteksi ada tidaknya dinding, maka saat itu baru dilakukan pengaktifan *sensor* dan pengambilan data. Apabila dilakukan rutin pengaktifan semua *sensor* secara bergantian menyebabkan beban kerja mikrokontroler sangat tinggi. Belum lagi jika dilakukan hal tersebut, dapat mengakibatkan gerak robot yang terputah-putah (Hendi Wicaksono, 2013) .

Penggunaan servo yang banyak juga memerlukan kontrol servo yang baik sebagaimana yang dilakukan Sudarmanto “dalam pembuatan modul servo controller ini dibutuhkan dua buah mikrokontroler utama yang dapat mengontrol dua belas motor servo sekaligus dan mikrokontroler bantu yang digunakan untuk mengirim alamat dan data lebar pulsa secara serial ke mikrokontroler utama, juga sebagai masukan perintah pengendalian gerakan robot”(Sudarmanto, 2007).

Penggunaan algoritma juga mempengaruhi *Behavior* Robot agar lebih cerdas. Pada penelitian berikut dari hasil simulasi nampak bahwa *Behavior based hexapod robot* memiliki mobilitas yang baik (mampu melewati halangan setinggi maksimal 10 cm) dan dapat menyelesaikan tugasnya untuk menghindari halangan (Handy Wicaksono, dkk, 2009).

Berdasarkan uraian yang telah disebutkan, dapat dilihat perancangan robot berkaki pemadam api membutuhkan penggunaan *Behavior based control hexapod robot* untuk lebih baik serta dua mikrokontroler untuk proses navigasi agar kinerja robot dapat optimal. Penelitian ini design robot berkaki dengan yang menggunakan arduino mega 2560 yang nantinya robot tidak lagi berjalan dengan terputah – putah yang disertai dengan penggunaan *Behavior based hexapod robot* untuk menambah keandalan pada robot.

## 1.2 Perumusan Masalah

Berdasarkan uraian latar belakang diatas, dapat dirumuskan beberapa rumusan masalah sebagai berikut :

1. Bagaimana mendesign robot berkaki agar *Behavior* kaki – kakinya berjalan dengan lancar.
2. Bagaimanan mendesain *Behavior based hexapod robot*.

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini, antara lain:

1. Merancang robot berkaki enam (*hexapode*).
2. Menerapkan *multiprosesor* untuk permasalahan *Behavior* kaki – kaki agar tidak terpatah – patah dengan *Behavior based robot*.

## 1.4 Manfaat

Manfaat yang di dapatkan dari penelitian ini adalah :

1. Robot KRPAI berkaki ini dapat lebih baik dalam bernavigasi dalam kontes robot berkaki selanjutnya.
2. Penelitian ini dapat digunakan sebagai rujukan oleh peneliti lain untuk melakukan pengembangan atau penelitian selanjutnya.

## 1.5 Batasan Masalah

Ruang lingkup penelitian ini dibatasi pada:

1. Pada penelitian ini hanya menggunakan dua mikrokontroler yang berbasis *arduino*.
2. Lapangan pengujian dilakukan dalam arena yang digunakan dalam perlombaan tahunan kontes robot indonesia.
3. Target penelitian ini robot dapat memadamkan api dengan baik dan tida ada navigasi robot untuk kembali ke *home*.
4. Target pencapaian dari perancangan ini robot dapat berjalan tanpa terpatah – patah.

## BAB 2

### TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa bagian robot yang menjadi dasar dalam pembuatan robot berkaki bagi tim robotika unej untuk peningkatan kinerja robot untuk tahun – tahun berikutnya mengikuti lomba. Hal yang pertama dalam bab ini yaitu bagaimana deskripsi bentuk robot berkaki ini, serta penggunaan algoritma yang akan digunakan untuk navigasi. Kemudian pada bab ini juga dijelaskan komponen – komponen yang digunakan seperti arduino mega 2560, sensor ultrasonik, Motor servo, dan sensor flame pendeteksi api.

#### 2.1 Robot Berkaki

Pada penelitian ini terkait dengan penggunaan algoritma pemrograman pada divisi robotika berkaki yang dapat membantu peningkatan algoritma yang lebih baik dari yang sebelumnya. Serta dalam penelitian ini menggunakan mekanik dari tim robotika unej yang telah ada.



**Gambar 2.1 Robot Berkaki**

(Sumber: <http://masbarron.blogspot.co.id/2011/05/program-robot-berkaki.html>)

Kaki *hexapod* digerakan menggunakan servo. Jumlah servo yang digunakan berbeda-beda mulai dari 3 servo hingga 18 servo. Perbedaan jumlah

servo tentunya membedakan pula algoritme *Behaviorkaki*, peletakan servo, serta *design* kerangka *hexapod*. *Design* kerangka *hexapod* terdiri atas tubuh yang kaku dengan enam kaki yang sesuai, masing-masing kaki memiliki *Behavior* yang bebas (Saranli *et al*, 2001).

Selain dari *design* kerangka robot yang berbeda, algoritme *Behaviorkaki* antara cara berjalan *tripod* dengan cara berjalan lambat atau berjalan lainnya berbeda-beda, hal ini dikarenakan kontrol algoritme berjalan biasanya tidak terpusat, yang berarti bahwa *Behavior* setiap kaki relatif bebas (Thirion dan Thiry, 2002).

## 2.2 Arduino Mega 2560

Arduino Mega 2560 adalah merupakan sebuah mikrokontroler yang dikembangkan oleh pihak arduino. Aplikasinya yang *open source* sehingga memudahkan dalam menggunakan. Arduino Mega 2560 penggunaannya baik bila dipakai untuk pengendalian yang banyak aktuatornya atau sensor – sensor. Arduino Mega 2560 sendiri memiliki chip ATmega 2560 serta banyak pin pada *board* arduino. Pada *board* ini memiliki pin Input/Output yang lumayan banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin *input* analog, 4 pin UART (*serial port hardware*). Adapun spesifikasi dari arduino mega 2560 sendiri dapat dilihat pada Gambar 2.2.



**Gambar 2.2** Arduino Mega 2560

( Sumber : <https://www.arduino.cc/en/Main/arduinoBoardMega> )

Terdapat beberapa pin tegangan yang tersedia pada Arduino antara lain, Pin 5V dimana pin tersebut mengeluarkan tegangan regulator 5V karena regulator tersebut yang mengatur tegangan tersebut. Selanjutnya pin 3V3 ialah pin yang mengeluarkan tegangan menghasilkan 3,3 V dan untuk arus sendiri memiliki batas maksimal 50mA. Kemudian untuk pin VIN ialah *input* tegangan arduino menggunakan daya eksternal, atau jika memasok tegangan melalui *jack power*, dapat diakses melalui tegangan yang ada pada pin ini. Selanjutnya GND atau Ground ialah pin ground atau massa. Sedangkan untuk IOREF berfungsi sebagai pemberi tegangan 10 referensi yang beroperasi pada mikrokontroler. Untuk tegangan referensi itu sendiri dapat memilih antara 5V atau 3,3.

**Tabel 2.1** Spesifikasi Arduino Mega 2560.

Chip mikrokontroller	ATmega2560
Tegangan operasi	5V
Tegangan input (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM output
Analog Input pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori Flash	256 KB, 8 KB telah digunakan untuk bootloader
SRAM	8 KB
EEPROM	4 KB
Clock speed	16 Mhz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g

( Sumber : <https://www.arduino.cc/en/Main/arduinoBoardMega> )



#### ✚ *Input dan Output (I/O)*

*Input* dan *output* pada Arduino Mega 2560 memiliki jumlah pin paling banyak dari semua *board* Arduino. Arduino Mega 2560 memiliki sebanyak 54 buah pin digital yang bekerja pada tegangan 5V, dan setiap pin dapat memberi maupun menerima arus sebesar 20mA, yang dapat difungsikan sebagai *input* maupun *output*, dengan menggunakan sintak pemrograman `pinMode()`, `digitalWrite()`, dan `digitalRead()`.

Pin-pin memiliki fungsi khusus:

- **Serial**, memiliki 4 pin *serial* yang masing-masing terdiri dari 2 pin RX dan TX. *Serial 0* : pin 0 (RX) dan pin 1 (TX). *Serial 1* : pin 19 (RX) dan pin 18 (TX). *Serial 2* : pin 17 (RX) dan pin 16 (TX). *Serial 3* : pin 15 (RX) dan pin 14 (TX). RX sebagai penerima dan TX sebagai pengirim data *serial*. Pin 0 dan pin 1 adalah pin *serial* yang digunakan oleh USB to TTL ATmega16U2.
- **External Interrupts**, yaitu 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*). Dengan demikian Arduino Mega 2560 memiliki jumlah *interrupt* yang cukup banyak yaitu 6 buah. Untuk mengaktifkannya dengan sintak `attachInterrupt()`.
- **PWM**: Pin 2 sampai 13 dan 44 sampai 46, adalah pin yang menyediakan *output* PWM 8 bit dengan menggunakan sintak `analogWrite()`.
- **SPI** : Arduino Mega 2560 mendukung komunikasi SPI yang terdapat pada Pin 50 (MISO), 51 (MOSI), 52 (SCK), dan 53 (SS) dengan menggunakan *SPI Library* untuk mengaktifkannya.
- **LED** : Pada pin 13 terdapat *built-in* LED yang dikendalikan oleh pin 13. Set *HIGH* untuk menyalakan, dan set *LOW* untuk mematikannya.
- **I2C** : Arduino Mega 2560 juga mendukung komunikasi I2C yang terdapat pada Pin 20 (SDA) dan pin 21 (SCL) dengan menggunakan *Wire Library* untuk mengaktifkannya.
- **AREF**: Sebagai tegangan referensi untuk *input* analog.

- **Reset:** Sama dengan tombol *reset*. Pada Arduino Mega terdapat pin yang berfungsi sebagai *reset* dengan dihubungkan ke *LOW* maka Arduino akan melakukan *reset*.

### 2.3 Penggunaan Sensor Ultrasonik PING

Penggunaan sensor ultrasonic pada robot berkaki sebanyak 7 buah sensor ultrasonik yang digunakan. Pada robot sebelumnya kita menggunakan sensor ultrasonic sebanyak 3 buah. Dikarenakan kurang efektif maka penggunaan 7 buah sensor ultrasonik sangat efektif.

Sensor Ping merupakan sensor jarak ultrasonik buatan Parallax. Sensor ini bekerja pada frekuensi 40 KHz dan hanya memiliki 3 jalur pin, yaitu 11 jalur sinyal (SIG), jalur VCC 5V dan jalur ground. Sensor Ping mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama 200 $\mu$ s, kemudian mendeteksi pantulannya. (Budiharto, 2008) Spesifikasi sensor Ping adalah sebagai berikut: a. Kisaran pengukuran 3cm – 3m b. Input trigger-positive TTL pulse, minimal 2 $\mu$ s, tipikal 5 $\mu$ s c. Echo hold off 750 $\mu$ s dari fall of trigger pulse d. Waktu tunda untuk pengukuran selanjutnya 200 $\mu$ s e. Indikator LED untuk aktifitas sensor. (Tri Sutrisno Wahyu, 2010).



**Gambar 2.3 Sensor Ultrasonik PING**

( Sumber : <https://www.parallax.com> )

### 2.4 Motor Servo

Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari motor dc, rangkaian *gear*, potensiometer

dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo, sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo.(Hidayat, Iswanto, & Muhammad, 2015).

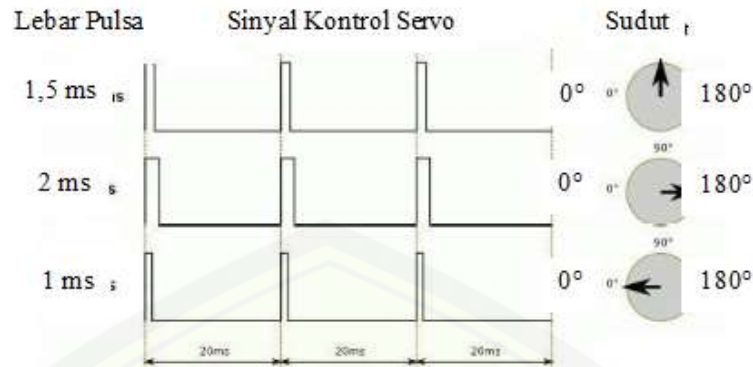
Motor servo pun dari penelitian ini digunakan untuk aktuator untuk menggerakkan kaki – kaki pada robot pemadam api berkaki yang mana setiap kaki ada 2 persendian yang akan digunakan untuk bergerak.



**Gambar 2.4** Motor Servo

( Sumber : <http://www.robotiksistem.com> )

Untuk menggerakkan motor servo ke kanan atau ke kiri, berdasarkan dari nilai *delay* yang kita berikan. Untuk membuat motor servo pada posisi *center* diberikan pulsa sebesar 1.5ms. Dan untuk membuat servo berputar ke kanan diberikan pulsa lebih kecil dari 1.3ms, dan pulsa lebih besar dari 1.7ms untuk berputar ke kiri dengan delay 20ms, seperti pada gambar 2.5.



**Gambar 2.5 Pulsa Kendali Motor Servo**

(Sumber: Sujarwata, 2013)

## 2.5 Sensor Api

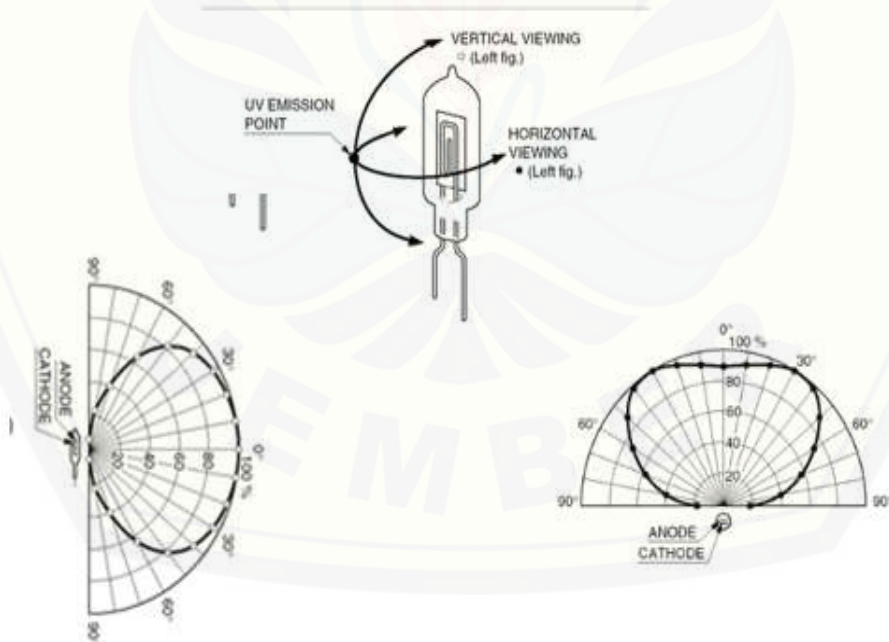
UVTron *Flame Detector* dan rangkaian *driver*-nya dapat mendeteksi api dari lilin atau puntung rokok dalam jarak 5 meter. Sensor ini digunakan sebagai alat untuk mendeteksi sumber api yang beroperasi pada panjang spectral 185 nm hingga 160 nm. Tabung UVTron bekerja ketika pada katoda diberikan sinyal ultraviolet, fotoelektron akan dipancarkan dari katoda oleh efek fotoelektrik dan dipercepat ke anoda menggunakan medan listrik. Ketika tegangan diberikan, medan magnet meningkat, medan listrik semakin besar dan energi kinetik elektron menjadi besar untuk mengionisasi molekul gas di tabung untuk bertubrukan. Elektron yang dibangkitkan oleh ionisasi dipercepat yang membuat elektron mengionisasi molekul lainnya sebelum mencapai anoda. Ion positif yang dipercepat ke katoda dan bertubrukan 12 menyebabkan elektron sekunder lainnya, kejadian ini menyebabkan arus yang cukup besar diantara elektroda dan terjadi pembuangan muatan. Berikut bentuk sensor UVTron beserta *driver*-nya.



**Gambar 2.6** Sensor Api UVTron

( Sumber: <http://klinikrobot.com/product/heat-fire-array-sensor/flame-sensor-uv-tron-w-driving-circuit.html>).

Adapun dari karakteristik sensor UV Tron terdapat jarak jangkauan yang ditangkap ketika mendeteksi api adapun jarak jangkauan dapat dilihat seperti pada Gambar 2.7.



**Gambar 2.7** Tabung UV Flame Detector Tabung sensor UV Flame Detector dan sudut respon sensor secara horizontal.

( Sumber : <http://klinikrobot.com/product/heat-fire-array-sensor/flame-sensor-uv-tron-w-driving-circuit.html> )

## 2.6 Pompa Wiper (*Extinguisher*)

Pada robot berkaki biasanya menggunakan berbagai pompa yang ada dipasaran namun dari berbagai pompa itu kami menggunakan pompa pada gambar 2.8 yang mana pompa tersebut memiliki daya sembur yang cukup kuat dari tegangan 12 volt yang digunakan. Pada biasanya pompa ini sering digunakan pada mobil – mobil untuk membersihkan kotoran pada kaca mobil, dimana penyemprot ini akan menghasilkan hasil semprotan yang dikeluarkan dengan tekanan 3 Pa memungkinkan untuk memadamkan api pada jangkauan lebih dari 20 cm.

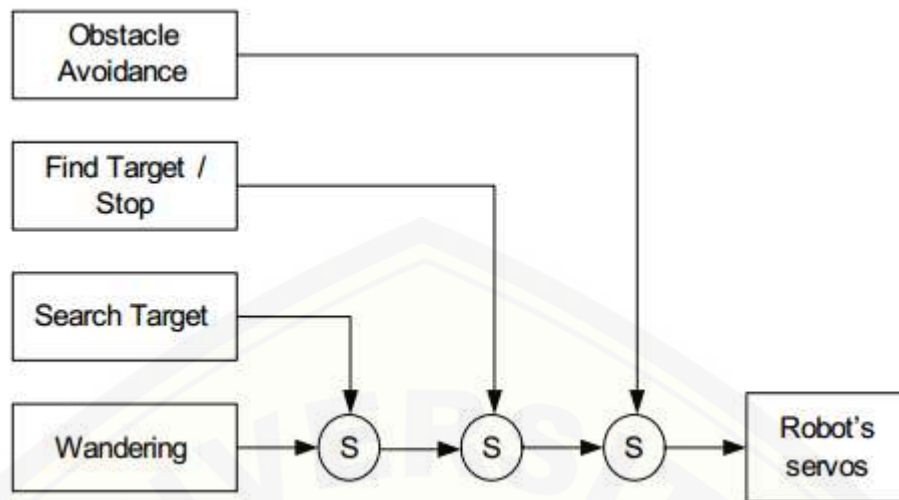


**Gambar 2.8** Penyemprot Air Pemadam Api

(Sumber: muhammad yasfi., 2017)

## 2.7 Algoritma penggunaan *Behavior based control hexapod robot*

Robot berkaki enam (*hexapod*) memiliki kelebihan dibanding robot beroda dalam hal kemampuannya melewati daerah tidak rata. Pada penelitian ini, mobilitas *Behaviorhexapod* akan diuji untuk mengetahui performanya dalam melewati balok dan tangga. Supaya dapat bereaksi dengan cepat, maka arsitektur *Behavior based* akan digunakan pada *hexapod*. Aplikasi navigasi otonom dipilih untuk menunjukkan bahwa arsitektur tersebut berjalan dengan baik. Dari hasil simulasi nampak bahwa *Behavior based hexapod robot* memiliki mobilitas yang baik (mampu melewati halangan setinggi maksimal 10 cm) dan dapat menyelesaikan tugasnya untuk menghindari halangan dan menemukan sumber cahaya. (Handy Wicaksono et al., 2008).



**Gambar 2.9** Struktur koordinasi *Behavior*

(Handy Wicaksono et al., 2008).

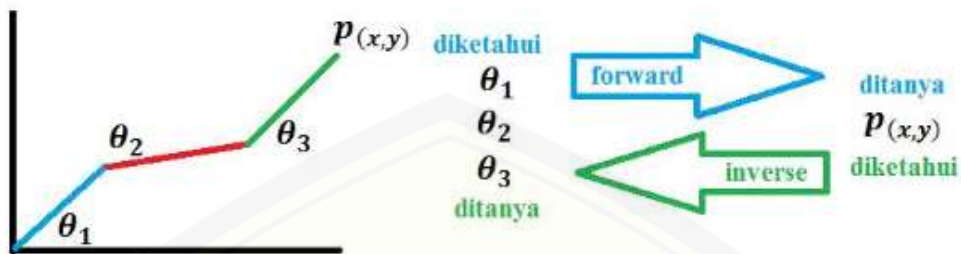
Tampak bahwa *Wandering* merupakan *Behavior* dengan level terendah, sehingga jika ada *Behavior* lain yang aktif, maka *wandering* tidak akan aktif. *Behavior* dengan level prioritas tertinggi ialah *obstacle avoidance*. Hal ini berarti jika *Behavior* tersebut aktif, maka semua *Behavior* lain akan non aktif. Dengan kata lain masing – masing *Behavior* akan aktif secara bergantian.

Dalam buku yang ditulis oleh Brooks, Rodney A., 1991, menjelaskan bahwa *Behavior based robots* sering menampilkan tindakan biologis muncul pada tiap-tiap tahap komputasi, hal ini dapat berdampak pada mengulangi tahap tersebut sehingga robot menjadi bingung. Namun terdapat suatu keuletan dimana program pada tahap tersebut diulang berkali-kali. Dengan begitu munculnya suatu *Artificial Intelligent* pada robot tersebut.

## 2.8 *Invers Kinematics*

*Inverse Kinematics* merupakan kebalikan dari *Forward Kinematics*, jika *Forward Kinematics* adalah metode untuk menentukan orientasi dan posisi *end-effector* dari besarnya sudut sendi dan panjang link lengan. Persamaan *forward kinematics* didapatkan berdasarkan jumlah DOF dan jenis *kinematics chain* dari kaki-kaki robot berkaki. Jadi *inverse kinematics* yaitu menentukan besarnya sudut sendi dari orientasi dan posisi *end-effector* saat panjang link telah ditetapkan.

Metode ini diperlukan untuk mengetahui besarnya sudut pada persendian yang diperlukan agar *end-effector* dapat mencapai posisi yang dikehendaki.



**Gambar 2.10** Model Kontrol Kinematika Robot

(Sumber: Handritour. 2013)

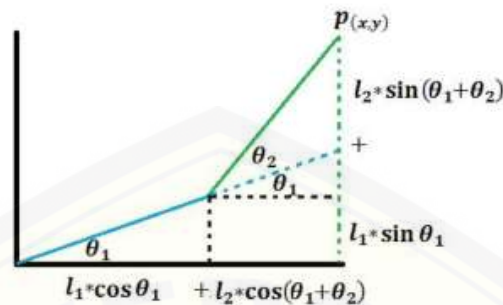
*Inverse Kinematics* lebih banyak diaplikasikan, namun memiliki kerumitan dalam perhitungannya karena beberapa hal, diantaranya:

1. Melibatkan persamaan *non-linier*, Solusi yang dihasilkan bisa banyak dan bahkan menjadi tak hingga.
2. Kemungkinan tidak mendapatkan solusi terjadi ketika posisi *end-effector* berada di luar *wrokspace* atau *configuration space*.
3. Solusi perhitungan akan semakin rumit ketika jumlah link dan sendi semakin banyak namun posisi *end-effector* menjadi semakin akurat.

Ada beberapa metode untuk memecahkan permasalahan *inverse kinematics*, yang metode yang dirasa paling sederhana adalah dengan pendekatan persamaan trigonometri.



Berikut ini adalah gambar grafik representasi sendi gerak lengan robot yang digunakan dalam perhitungan *inverse kinematics* pada robot berkaki:



**Gambar 2.11** Grafik Representasi Lengan Robot Berkaki

Dalam *inverse kinematics* 2 DOF memiliki kesulitan jika diselesaikan dengan hanya menggunakan persamaan trigonometri, tetapi masih dapat diselesaikan dengan cara berikut,

1. Berdasarkan *forward kinematics* 2 DOF

$$Px = l_1 \cdot \cos\theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2) \dots\dots\dots (2.1)$$

$$Py = l_1 \cdot \sin\theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2) \dots\dots\dots (2.2)$$

2. Pertama, harus mencari  $\theta_2$  terlebih dahulu,  $\theta_2$  dapat diperoleh dari peng-kuadrat-an dikedua sisi persamaan baik Px maupun Py,

$$P_x^2 = [l_1 \cdot \cos\theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2)]^2$$

$$P_x^2 = l_1^2 \cdot \cos^2\theta_1 + l_2^2 \cdot \cos^2(\theta_1 + \theta_2) + 2l_1l_2 \cdot \cos\theta_1 \cdot \cos(\theta_1 + \theta_2)$$

$$P_y^2 = [l_1 \cdot \sin\theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2)]^2$$

$$P_y^2 = l_1^2 \cdot \sin^2\theta_1 + l_2^2 \cdot \sin^2(\theta_1 + \theta_2) + 2l_1l_2 \cdot \sin\theta_1 \cdot \sin(\theta_1 + \theta_2)$$

$$P_x^2 + P_y^2 = l_1^2 \cos^2\theta_1 + l_2^2 \cos^2(\theta_1 + \theta_2) + 2l_1l_2 \cos\theta_1 \cdot \cos(\theta_1 + \theta_2) + l_1^2 \sin^2\theta_1 + l_2^2 \sin^2(\theta_1 + \theta_2) + 2l_1l_2 \sin\theta_1 \cdot \sin(\theta_1 + \theta_2)$$

$$P_x^2 + P_y^2 = l_1^2(\sin^2\theta_1 + \cos^2\theta_1) + l_2^2(\sin^2(\theta_1 + \theta_2) + \cos^2(\theta_1 + \theta_2)) + 2l_1l_2(\cos\theta_1 \cdot \cos(\theta_1 + \theta_2) + \sin\theta_1 \cdot \sin(\theta_1 + \theta_2))$$

3. Berdasarkan persamaan  $\cos^2\theta + \sin^2\theta = 1$

$$P_x^2 + P_y^2 = l_1^2 + l_2^2 + 2l_1l_2(\cos\theta_1 \cdot \cos(\theta_1 + \theta_2) + \sin\theta_1 \cdot \sin(\theta_1 + \theta_2))$$

4. Berdasarkan

$$\begin{aligned} \cos(a + b) &= \cos a \cdot \cos b - \sin a \cdot \sin b \ \& \ \sin(a + b) \\ &= \sin a \cdot \cos b + \cos a \cdot \sin b \end{aligned}$$

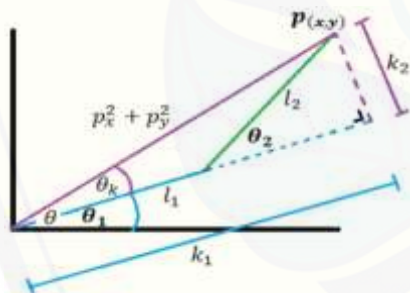
Maka:

$$\begin{aligned}
 P_x^2 + P_y^2 &= l_1^2 + l_2^2 + 2l_1l_2(\cos\theta_1 \cdot \cos(\theta_1 + \theta_2) + \sin\theta_1 \cdot \sin(\theta_1 + \theta_2)) \\
 P_x^2 + P_y^2 &= l_1^2 + l_2^2 + 2l_1l_2(\cos^2\theta_1 \cdot \cos\theta_2 - \cos\theta_1 \cdot \sin\theta_1 \cdot \sin\theta_2 + \sin^2\theta_1 \cdot \cos\theta_2 \\
 &\quad - \sin\theta_1 \cdot \cos\theta_1 \cdot \sin\theta_2) \\
 P_x^2 + P_y^2 &= l_1^2 + l_2^2 + 2l_1l_2(\cos\theta_2[\cos^2\theta_1 + \sin^2\theta_1] + \sin\theta_1 \cdot \cos\theta_1 \cdot \sin\theta_2 \\
 &\quad - \sin\theta_1 \cdot \cos\theta_1 \cdot \sin\theta_2) \\
 P_x^2 + P_y^2 &= l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2 \\
 2l_1l_2\cos\theta_2 &= P_x^2 + P_y^2 - l_1^2 - l_2^2 \\
 \cos\theta_2 &= \frac{P_x^2 + P_y^2 - l_1^2 - l_2^2}{2l_1l_2} \dots\dots\dots(2.3)
 \end{aligned}$$

5. Maka dari cos theta2 dapat diperoleh yaitu:

$$\theta_2 = \arccos\left(\frac{P_x^2 + P_y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \dots\dots\dots(2.4)$$

6. Untuk mencari  $\theta_1$  maka dapat menggunakan referensi dari gambar berikut:



**Gambar 2.12** Grafik Referensi untuk Mencari sebuah Grafik Referensi untuk Mencari  $\theta_1$

$$\begin{aligned}
 k_1 &= l_1 + l_2 \cdot \cos\theta_2 \\
 k_2 &= l_2 \cdot \sin\theta_2 \\
 \theta &= \operatorname{atan}\left(\frac{Py}{Px}\right) \\
 \theta &= \theta_k + \theta_1 \\
 \theta_1 &= \theta - \theta_k \\
 \theta_1 &= \operatorname{atan}\left(\frac{Py}{Px}\right) - \operatorname{atan}\left(\frac{k_2}{k_1}\right) \dots\dots\dots(2.5)
 \end{aligned}$$

## **BAB 3**

### **METODE PELAKSANAAN DATA**

Pada bab ini menjelaskan tentang dimana tempat dan kapan waktu pembuatan, ruang lingkup, jenis dan sumber data, serta metode pengumpulan data.

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

Penelitian yang berjudul tentang “Pengembangan algoritma robot berkaki dengan *implementasi Behavior Based Control* untuk menyelesaikan navigasi pemadam api pada kontes robot pemadam api berkaki”. Pada kegiatan ini pelaksanaan pembuatan alat dan pengujian alat akan dilaksanakan di Laboratorium Elektronika dan Terapan, Fakultas Teknik, Universitas Jember yang beralamat di Jalan. Slamet Riyadi no. 62 Patrang, Jember..

#### **3.2 Ruang Lingkup Kegiatan**

Ruang lingkup berkaitan tentang batasan – batasan masalah pada saat pembuatan alat yang mana batasan – batasan itu antara lain:

- a. Menggunakan robot berkaki *hexapod* dengan 2 DOF
- b. Sensor yang digunakan pada pembuatan alat ini yaitu sensor ultrasonik PING sebanyak 7 buah yang ditempatkan sesuai dengan gambar perancangan alat.
- c. Penggunaan algoritma *Behavior based control* untuk menambah kehandalan pada robot.
- d. Penggunaan sensor UV Tron untuk mendeteksi keadaan api.
- e. Akuisisi data berbasis arduino Mega 2560.

### 3.3 Jenis dan Sumber Data

Adapun jenis dan sumber data merupakan tentang bagaimana pengambilan data yang akan dijelaskan keseluruhannya pada alat yang akan dibuat, sebagai berikut:

- a. Servo digunakan untuk aktuator dari robot berkaki atau sebagai alat gerak untuk berpindah tempat.
- b. Sensor jarak ultrasonik PING, digunakan untuk navigasi robot yang ditempatkan seperti pada gambar perancangan alat.
- c. Sensor *Flame* UV Tron untuk mendeteksi api pada ruangan yang dituju oleh robot.
- d. *Extinguisher* digunakan untuk alat pemadam api yang dapat mengeluarkan cairan.
- e. Arduino Mega 2560 untuk mikrokontroler atau pengontrol *Behavior* Robot .

*Shielded Arduino 2560* digunakan untuk memudahkan pemasangan arduino serta efektif untuk kegunaan dan mempermudah pengerjaan pada saat pembuatan elektrika.

### 3.4 Metode Pengumpulan Data

Dalam pembuatan tugas akhir ini penggunaan metode *behaviour based control* untuk mendapatkan kehandalan pada sistem navigasi robot berkaki yang digunakan pada kontes robot indonesia dari divisi pemadam api.

- a. Pengumpulan studi Literatur.
- b. Melakukan pembuatan elektrikal serta design mekanik yang digunakan dan Melakukan pengujian elektrikal yang telah dibuat.
- c. Pengujian *hardware* yang ada untuk mengetahui fungsi normal atau tidak.
- d. Pengujian hardware dengan program yang dibuat.
- e. Pembuatan software untuk kinerja pada sistem.

- f. Pengujian keseluruhan elektrikal, hardware dan software yang dibuat yang telah dilakukan pengujian satu persatu tadi.
- g. Menganalisa hasil pengujian yang telah dilakukan.

### 3.5 Perancangan Alat

#### 3.5.1 Perancangan *Design* Mekanik Robot

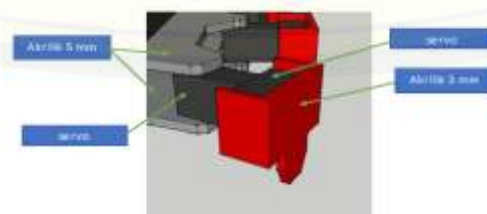
Pada pembuatan alat ini menggunakan sensor ultrasonik untuk sensor jarak yang digunakan robot bernavigasi. Pada mekanik robot menggunakan servo yang digunakan untuk *Behavior* Robot untuk berpindah posisi yang saling berintegrasi dengan sensor ultrasonik untuk melakukan *manuver*. kemudian juga menggunakan sensor *flame* UV Tron untuk mendeteksi keberadaan api pada arena perlombaan. Pada robot berkaki ini penggunaan sensor jarak ultrasonik.



**Gambar 3.1** Mekanik robot pemadam api berkaki

Dari bentuk bentuk robot diatas kita dapat mengetahui design kaki dan design body yang digunakan yakni :

- a. *Design* Kaki Robot Pemadam Api Berkaki



**Gambar 3.2** Design Kaki

b. *Design* Peletakan Sensor Pada Robot

Pada peletakan sensor pada robot berkaki pemadam api sebagai berikut karena dinilai lebih efektif untuk melakukan manuver ketika pada saat pengujian.

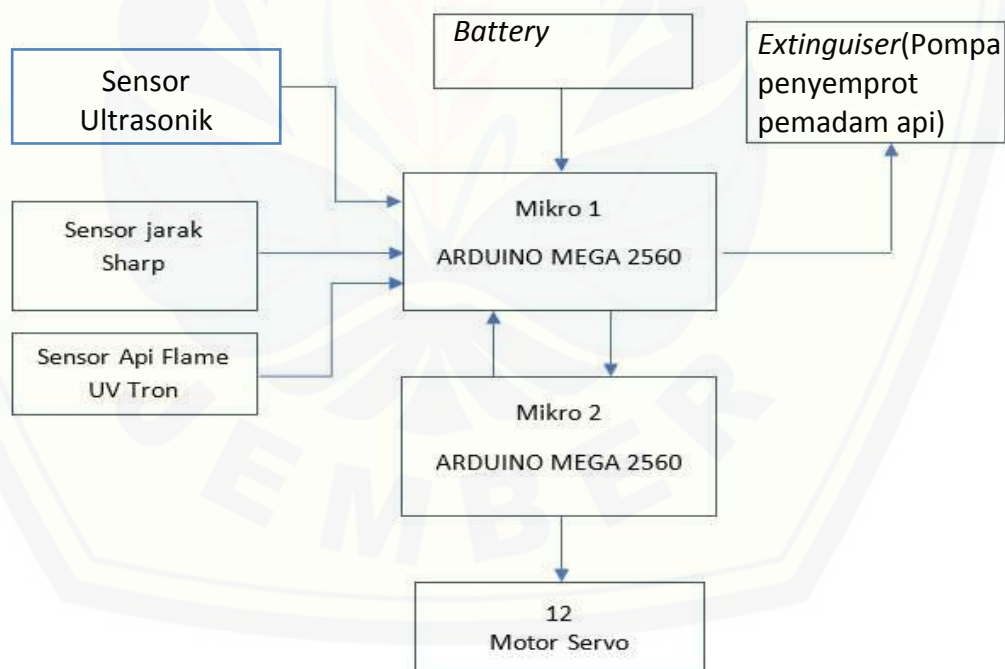


**Gambar 3.3** Desain Peletakan Sensor pada Robot

### 3.5.2 Perancangan Sistem Elektronika Robot

Pada bagian ini akan menjelaskan pada perancangan perangkat keras yang akan digunakan.

a. Blok Diagram



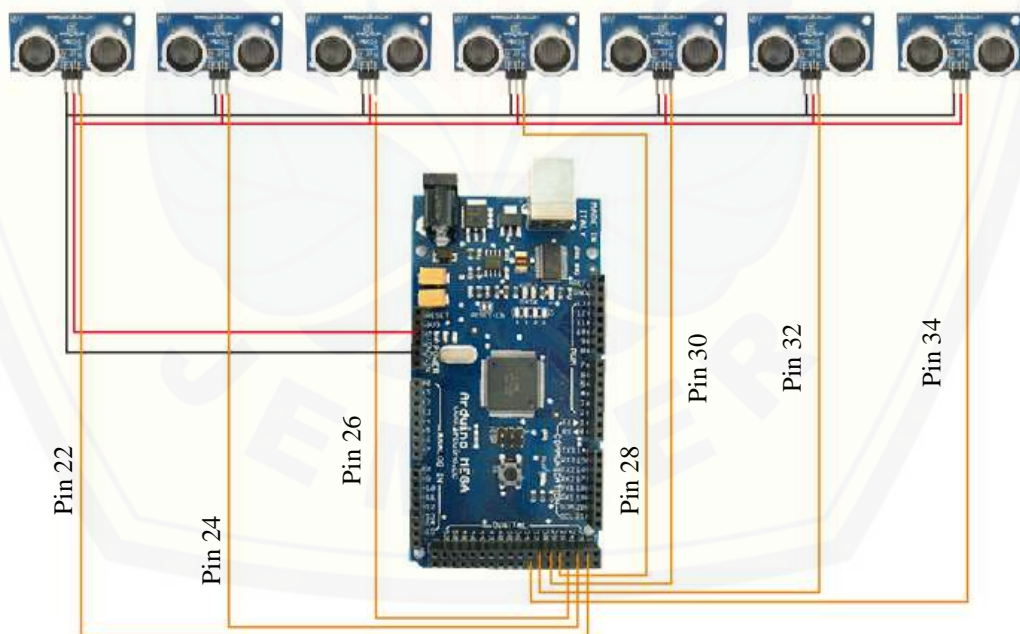
**Gambar 3.4** Blok diagram untuk Sistem Elektronika Robot Berkaki

Pada diagram blok diatas arduino mendapatkan *supply* oleh *batteray* yang mana untuk catu daya yang akan digunakan, kemudian pada sensor terdapat sensor ultrasonik yang sebanyak 7 sesuai perancangan alat yang dijelaskan diatas, serta

penggunaan sensor jarak sharp yang bertujuan untuk mendeteksi keberadaan penghalang yaitu boneka karena sensor ultrasonik tidak dapat mendeteksi keberadaan boneka. Oleh karena itu digunakan sensor jarak sharp tersebut yaitu dengan pengenalan lingkungan dengan infrared beda dengan sensor ultrasonik yang mendeteksi menggunakan gelombang ultrasonik. Serta penggunaan sensor api flame UV Tron untuk mendeteksi api. Lalu pada sensor jarak tersebut menggerakkan *Behaviorservo* yang mana ultrasonik dan sensor sharp saling berintegrasi dengan motor servo. Ketika sensor UV Tron mendeteksi api nantinya api akan dimatikan dengan cairan yang mana cairan tersebut keluar dari exinguiser untuk memadamkan api.

b. Rangkaian Ultrasonik Sensor

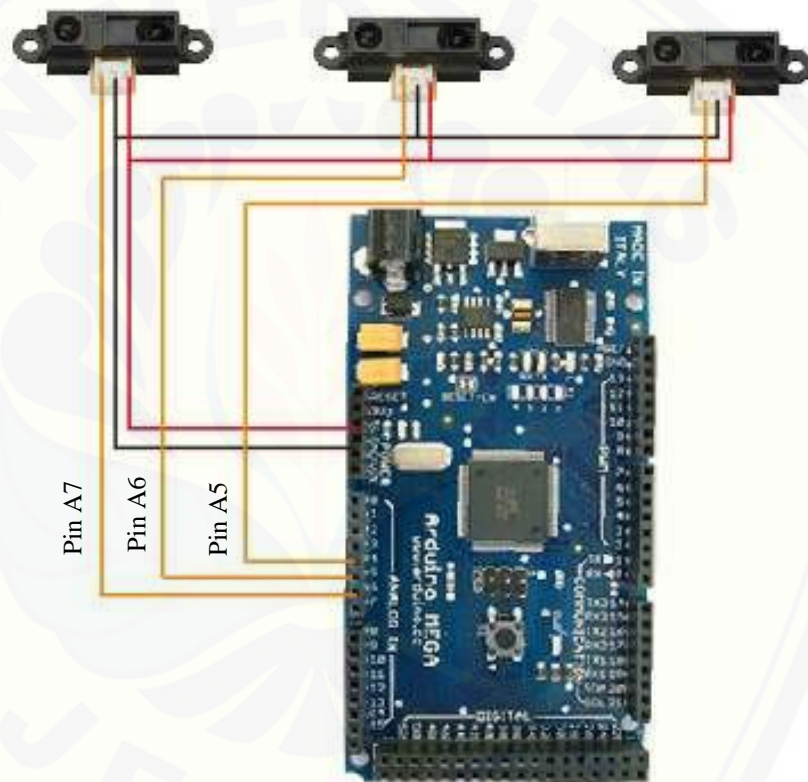
Pada sensor ultrasonik terdapat 3 pin yaitu GND, VCC, INPUT yang masing masing terhubung pada arduino Mega 2560. Pada *Input* terhubung dengan pin pada arduino Mega yaitu pada pin 22, pin 24, pin 26, pin 28, pin 30, pin 32, dan pin 34.



**Gambar 3.5** Rangkaian Sensor Ultrasonik Pada Robot Berkaki

c. Rangkaian Sensor Sharp

Rangkaian sensor sharp pada penggunaan robot berkaki menggunakan tiga sensor yang mana penempatan sensor tersebut yaitu didepan, depan serong kanan sedikit, dan depan serong kiri sedikit. Pada sensor sharp terdapat 3 pin GND, VCC, dan INPUT yang akan terhubung pada *mikrokontroler* arduino Mega yang mana pin yang digunakan yaitu pada masing – masing sensor diantaranya pin A5 untuk sensor kanan, pin A6 digunakan sensor depan, dan A7 digunakan untuk sensor kiri.

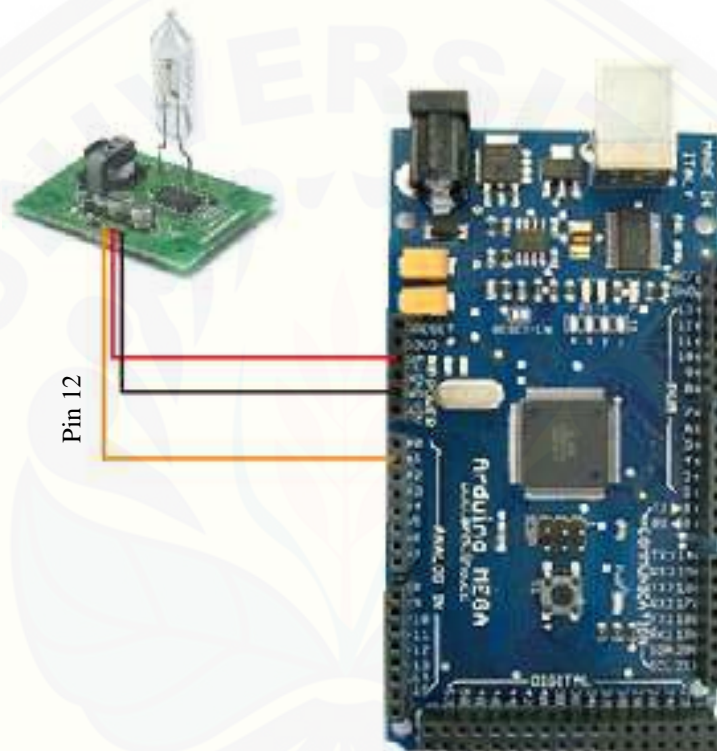


**Gambar 3.6** Rangkaian Sensor Sharp pada robot berkaki



d. Rangkaian Sensor Api UVTron

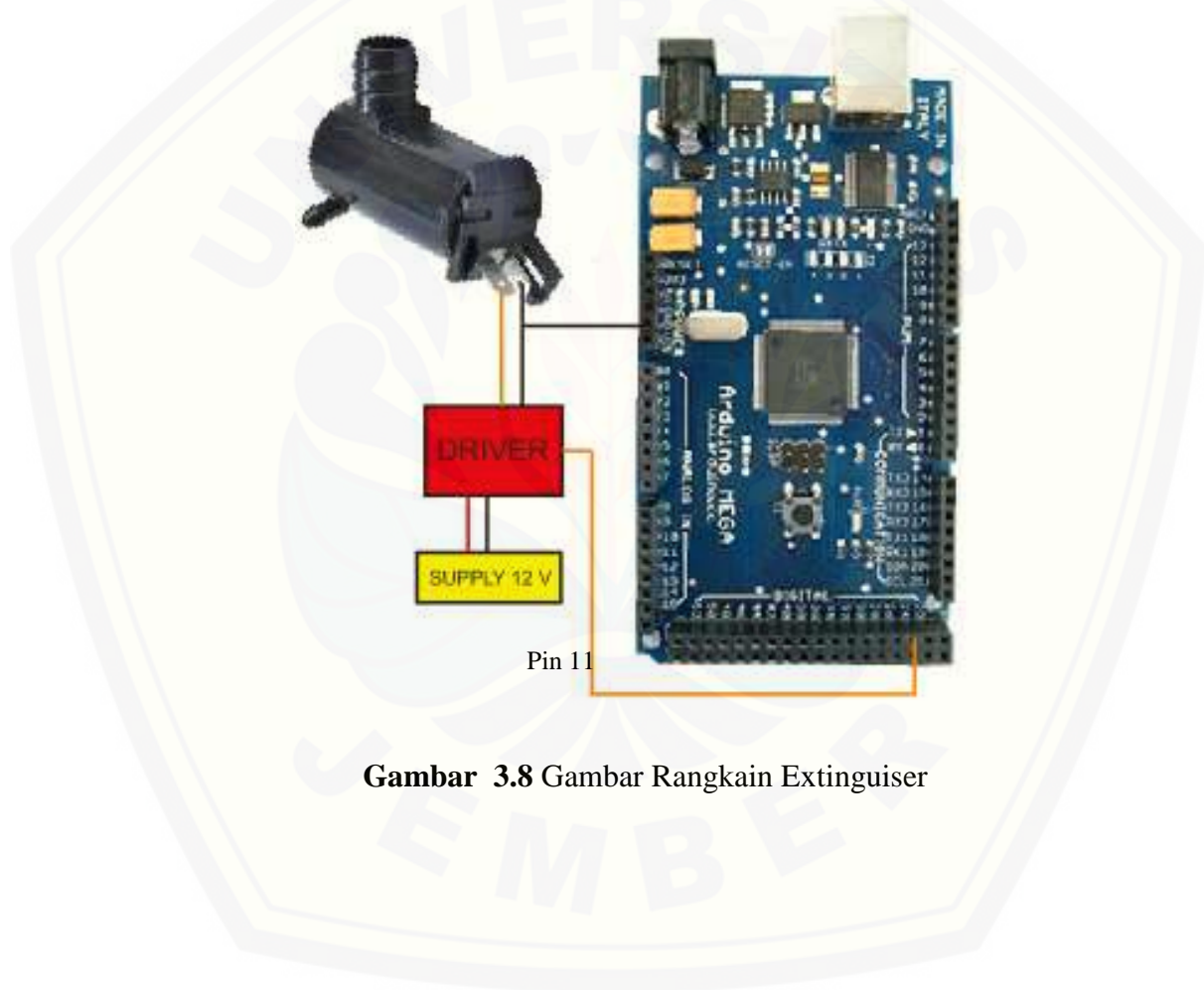
Pada rangkaian untuk sensor sharp sendiri terdapat 4 pin yang terdapat pada modul sensor itu sendiri namun yang digunakan hanya tiga yaitu GND, VCC, dan INPUT yang ketika menemukan api berlogika LOW. Jadi penggunaan sensor ini untuk mendeteksi api pada ruangan yang nantinya akan berlogika LOW ketika menemukan api dan berlogika HIGH ketika tidak menemukan api.



**Gambar 3.7** Rangkaian Sensor Sharp pada robot berkaki

e. Rangkaian *Extinguisher*

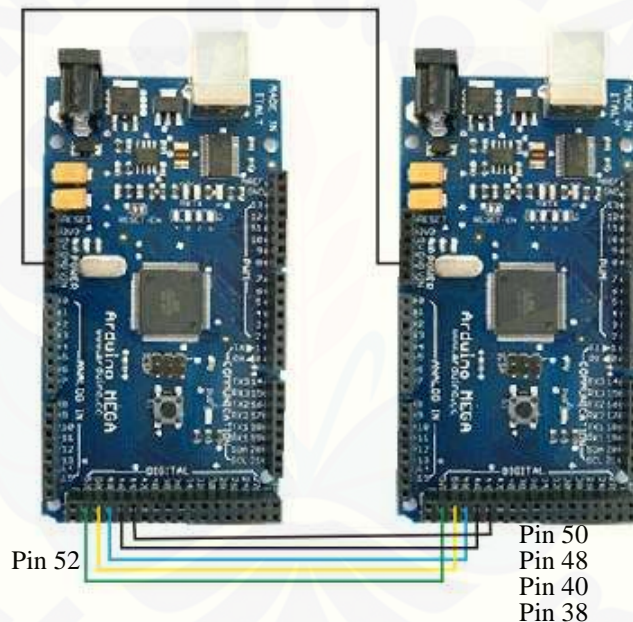
Pada rangkaian *Extinguisher* kita menggunakan *supply* 12 volt yang mana pada rangkaian terhubung dengan *battery* 12 volt lalu terhubung dengan driver motor untuk menggerakkan pompa tersebut. Pada driver motor terdapat pin yang terhubung dengan pin arduino Mega yaitu VCC dan INPUT untuk GND sendiri sudah terhubung dengan GND dari *supply* itu sendiri. Pada pin yang digunakan yaitu pin 24 pada *mikrokontroler* yang bawah.



**Gambar 3.8** Gambar Rangkain Extinguisher

f. Rangkaian Komunikasi antar Mikrokontroler

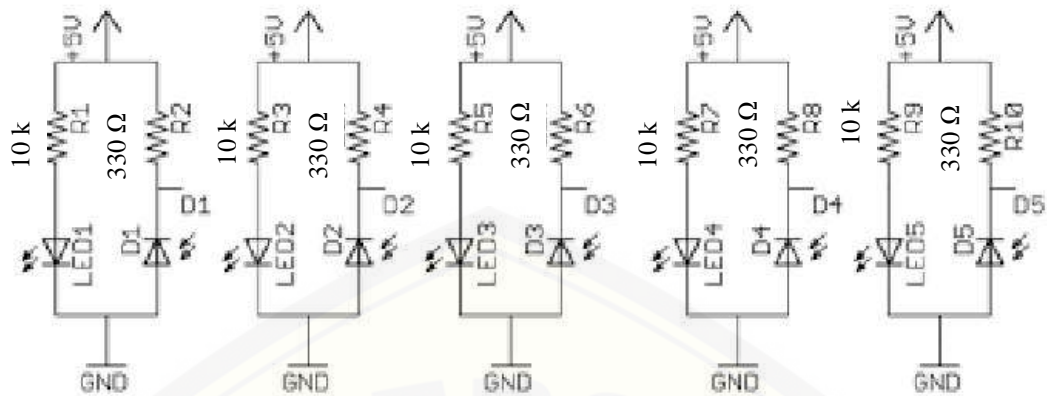
Komunikasi yang digunakan pada penelitian ini yaitu dengan komunikasi pada pin *digital* yang biasanya digunakan untuk *input* dan *output*. Pada komunikasi yang digunakan ini menggunakan 4 pin diantaranya pin 48, pin 50, pin 52, pin 40, pin 38 yang saling terhubung dan masing – masing *ground* yang telah terhubung. Pada ketiga pin tersebut nantinya dibuat sebuah pengkodean biner yang nantinya menghasilkan 16 keadaan yang dapat diterapkan pada *Behavior Robot* untuk berjalanya. Pada setiap keadaan memiliki perintah yang berbeda yang dapat membantu robot bergerak dengan dengan baik tanpa ada data yang tertukar.



**Gambar 3.9** Komonikasi antar mikrokontroler

g. Rangkaian Sensor Garis

Pada rangkaian sensor garis yang digunakan semakin kuat intensitas cahaya yang memaparnya, semakin rendah nilai resistansinya, seperti halnya warna yang terlihat oleh mata kita merupakan efek pantulan cahaya yang mengenai permukaan suatu benda. Warna putih dihasilkan oleh pantulan sempurna dari cahaya terhadap suatu permukaan, sedangkan warna hitam disebabkan tidak adanya pantulan.



**Gambar 3.10** Gambar Rangkain Sensor Garis

Keterangan :

- D1 = Pin yang digunakan A1
- D1 = Pin yang digunakan A2
- D1 = Pin yang digunakan A3
- D1 = Pin yang digunakan A4
- D1 = Pin yang digunakan A5

### 3.6 Sistem Kendali Robot *Behavior Based Control*

#### 3.6.1 *Behavior Based Control*

Pada bagan dibawah ini akan menjelaskan tentang bagaimana robot krpai berkaki melakukan perilaku yang diinginkan untuk perlombaan pada kontes robot indonesia.



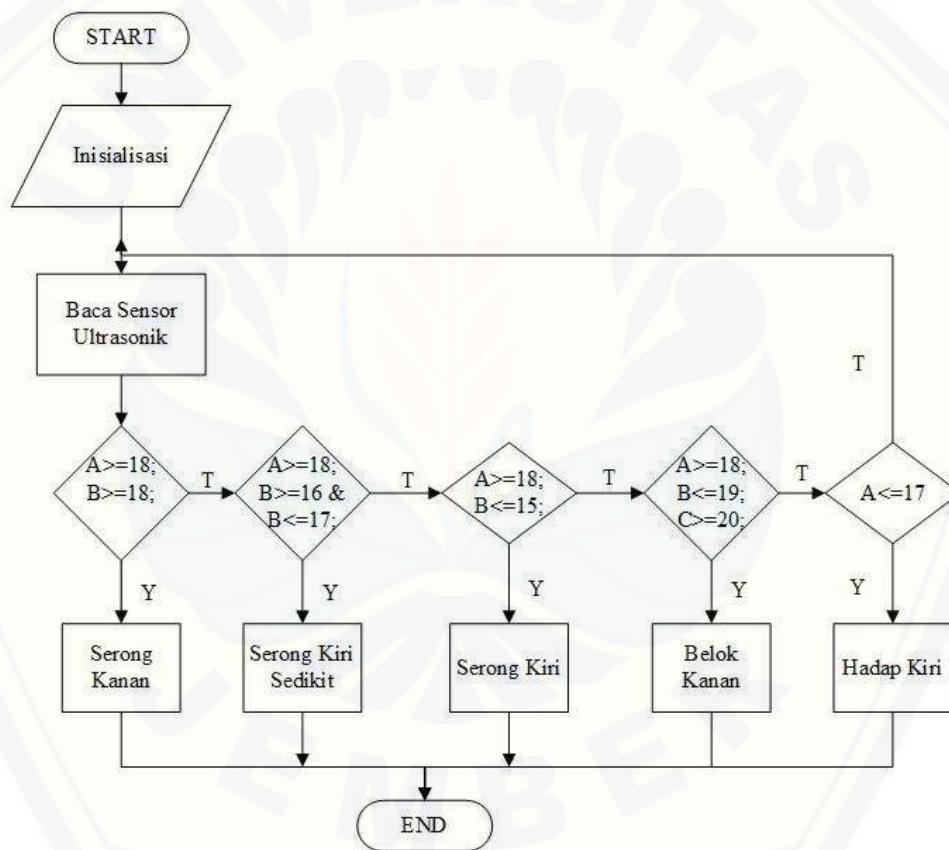
**Gambar 3.11** Algoritma *behaviour based control* robot KRPAl berkaki

Robot sistem kontrol diatas merupakan perilaku yang akan diprogram oleh robot. Perilaku pada robot tersebut dipengarungi oleh sensor yaitu sensor ultrasonik yang berada pada kepala robot yang digunakan untuk mendeteksi variabel untuk mengambil keputusan. Untuk melakukan pemadaman api robot membutuhkan aktuator untuk memutuskan robot bergerak kemana dari input yang digunakan yaitu sensor ultrasonik PING mendeteksi keberadaan benda disekitarnya sebelum menelusuri ruangan dengan menggunakan *Metode Behavior Based Control*. Metode ini menjadikan beberapa perilaku menjadi satu. Perilaku pertama robot dapat mengenali objek disekitar atau jarak – jarak yang disekitarnya. Kemudian robot bergerak setelah menemukan posisi yang sesuai untuk bergerak menelusuri ruangan. Selanjutnya melakukan dan mendeteksi garis agar keluar dari dalam ruangan. Selanjutnya robot bergerak menelusuri arena dengan mencari garis disetiap ruangan untuk mendeteksi api. Pendeteksian api setiap garis yang didapat

diruangan, robot mulai berperilaku pendeteksi api jika tidak ada api yang dideteksi maka robot berperilaku bergerak mundur serta berbalik untuk mencari ruangan yang lain. Selanjutnya robot berperilaku yang sama dengan bantuan koordinator robot berjalan tetap mencari api ketika api dideteksi di ruangan tertentu maka perilaku robot akan menuju api terlebih dahulu sampai robot mendeteksi kurang lebih 30 cm dan robot mulai berperilaku memadamkan api.

### 3.6.2 Bagian *Behavior Based Control*

#### 3.6.2.1 *Behavior Navigasi Robot*



**Gambar 3.12** Flowchart Behavior Navigasi Telusur Kanan.

Keterangan Gambar 3.12 flowchart dari Behavior Navigasi Robot

A= Sensor depan;

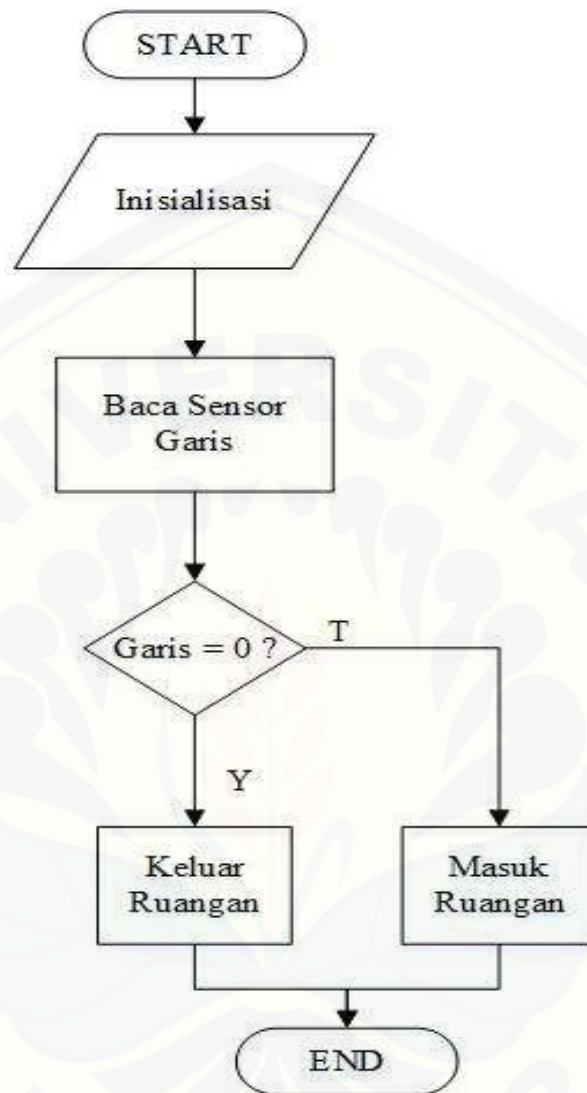
B= Sensor samping kanan;

C= Sensor kanan;

*Behavior* navigasi robot dimulai dari pembacaan sensor ultrasonik dimana robot akan melakukan perilaku :

- Serong kanan apabila nilai dari sensor depan dan sensor samping kanan lebih besar dari atau sama dengan 18.
- Serong Kiri Sedikit apabila nilai dari sensor depan lebih besar dari atau sama dengan 18 dan sensor samping kanan lebih besar dari 16 dan kurang dari 17.
- Serong kiri apabila nilai dari sensor depan lebih besar dari atau sama dengan 18 dan sensor samping kanan kurang dari atau sama dengan 15.
- Belok Kanan apabila nilai dari sensor depan lebih besar dari atau sama dengan 18 dan sensor samping kanan kurang dari atau sama dengan 19 dan sensor kanan lebih besar dari atau sama dengan 20.
- Hadap kiri apabila nilai dari sensor depan kurang dari atau sama dengan 17.

### 3.6.2.2 Behavior Mendeteksi Garis

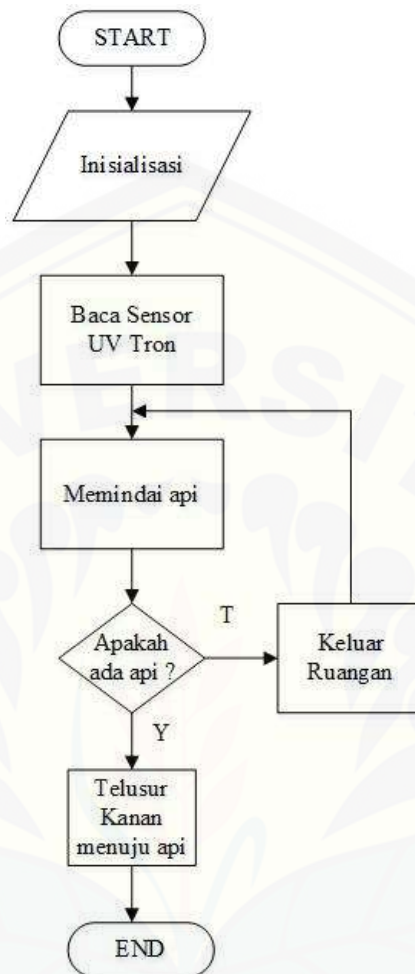


**Gambar 3.13** Flowchart Behavior mendeteksi garis

Gambar 3.13 merupakan *flowchart Behavior* mendeteksi garis pada perilaku ini robot akan melakukan pembacaan sensor garis kemudian akan melihat kondisi apakah sensor dapat mendeteksi garis dengan logika 0, dimana nilai 0 merupakan nilai garis pertama. Ketika sensor mendeteksi garis dengan nilai 0 maka robot akan keluar ruangan. Apabila robot tidak mendeteksi garis dengan nilai 0 maka robot akan masuk ke ruangan.



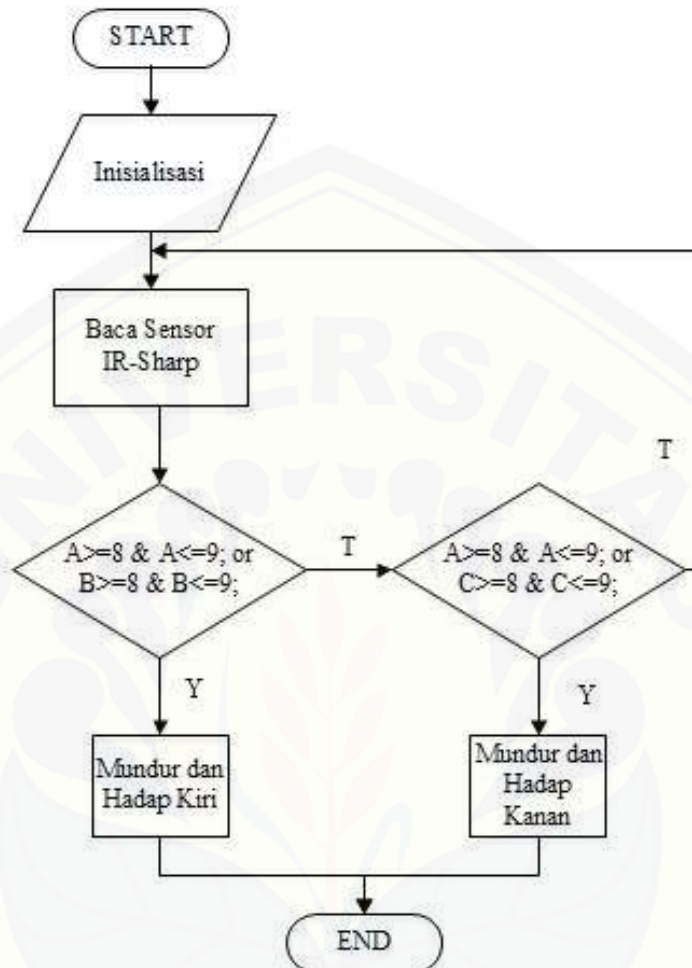
### 3.6.2.3 Behavior Memindai Api



**Gambar 3.14** Flowchart Behavior memindai api

Gambar 3.14 merupakan *Flowchart Behavior* memindai api, dimana robot ketika memindai api akan berhenti dan memindai dengan sensor api yaitu UV Tron yang digunakan pada robot berkaki ini. Setelah robot mendeteksi api maka Sensor api akan membaca api berlogika LOW (0) yang berarti robot menemukan api. Maka robot akan langsung menuju api dan masuk dalam ruangan dengan berperilaku telusur kanan untuk menuju api. Jika robot tidak mendeteksi api maka robot akan keluar ruangan dan kembali memindai api pada ruangan lainnya.

### 3.6.2.4 Behavior Mendeteksi Boneka



**Gambar 3.15** Flowchart Behavior mendeteksi boneka

Keterangan Gambar 3.15 Flowchart Behavior mendeteksi boneka :

A= Sensor depan;

B= Sensor kanan;

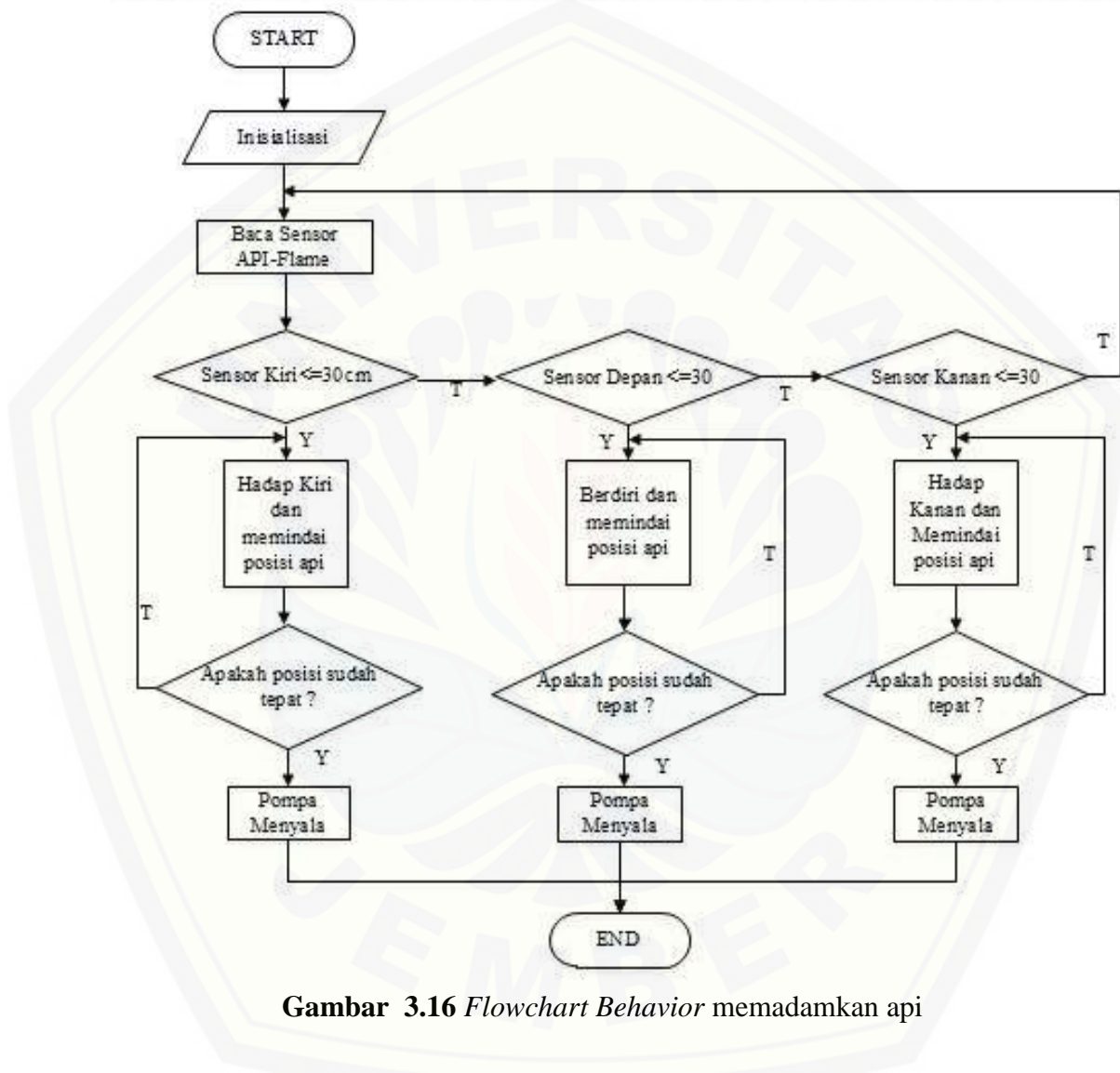
C= Sensor kiri;

Behavior mendeteksi boneka dimulai dari pembacaan sensor IR-Sharp dimana robot akan melakukan perilaku :

- Mundur dan hadap kiri apabila nilai sensor depan lebih besar dari atau sama dengan 8 dan kurang dari atau sama dengan 9 atau nilai sensor kanan lebih besar dari atau sama dengan 8 dan kurang dari atau sama dengan 9.

- Mundur dan Hadap Kanan apabila nilai sensor kiri lebih besar dari atau sama dengan 8 dan kurang dari atau sama dengan 9.

### 3.6.2.5 Behavior Memadamkan Api



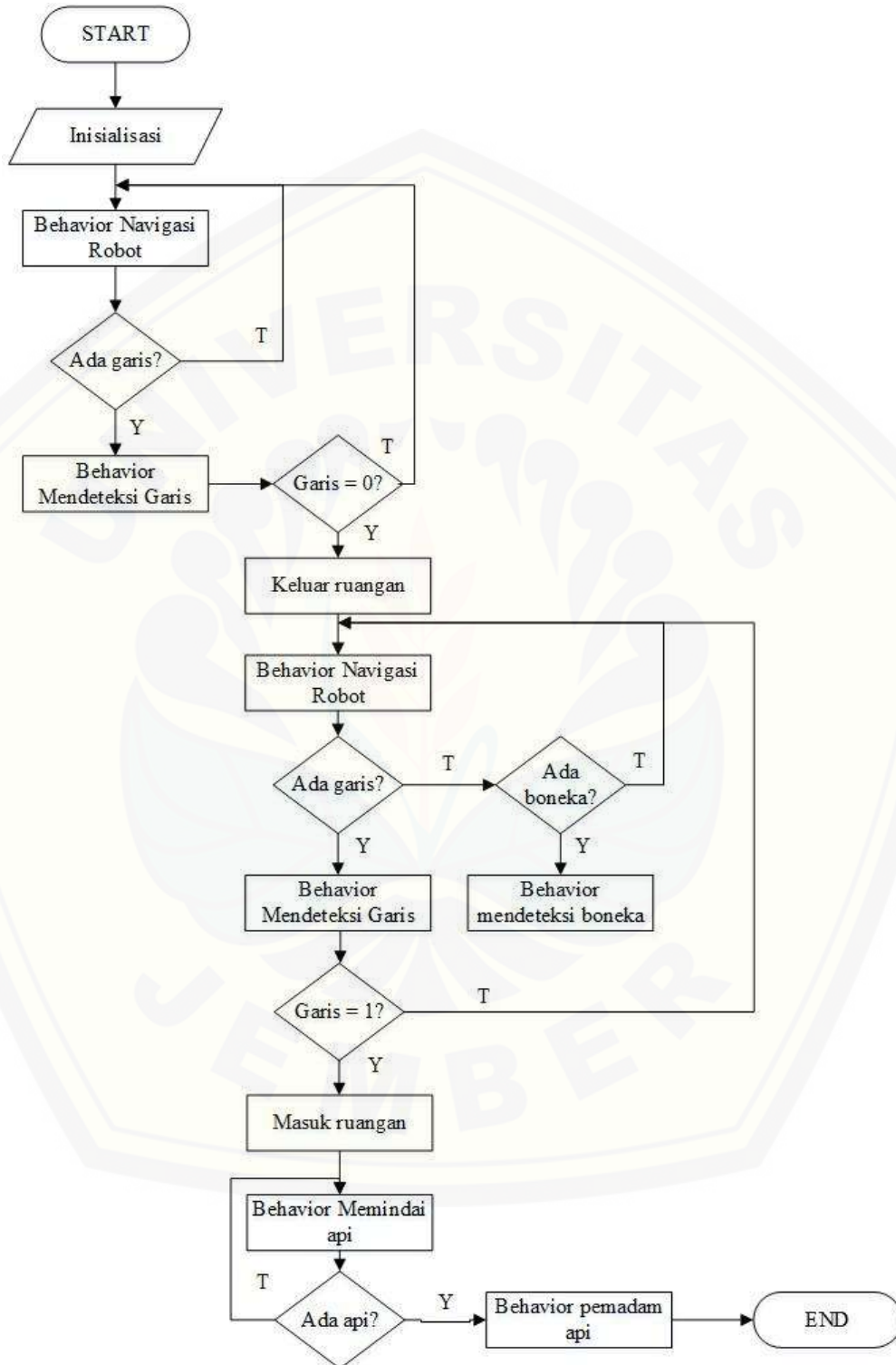
**Gambar 3.16** Flowchart Behavior memadamkan api

Gambar 3.16 Flowchart Behavior memadamkan api dimulai dari pembacaan sensor api-flame kemudian akan berperilaku :

- Ketika sensor kiri mendeteksi api pada jarak kurang dari atau sama dengan 30 maka robot akan berperilaku hadap kiri kemudian akan memindai posisi atau keberadaan api ketika posisi keberadaan api sudah ditemukan maka

pompa air menyala apabila posisi api belum ditemukan maka akan melakukan pemindaian ulang api.

- Ketika sensor depan mendeteksi api pada jarak kurang dari atau sama dengan 30 maka robot akan berperilaku berdiri kemudian akan memindai posisi atau keberadaan api ketika posisi keberadaan api sudah ditemukan maka pompa air menyala apabila posisi api belum ditemukan maka akan melakukan pemindaian ulang api.
- Ketika sensor depan mendeteksi api pada jarak kurang dari atau sama dengan 30 maka robot akan berperilaku hadap kanan kemudian akan memindai posisi atau keberadaan api ketika posisi keberadaan api sudah ditemukan maka pompa air menyala apabila posisi api belum ditemukan maka akan melakukan pemindaian ulang api.

**3.7 Flowchart Behavior Robot Keseluruhan****Gambar 3.17** Flowchart Behavior Robot Berkaki Pemadam Api

Gambar 3.17 merupakan flowchart *Behavior* robot berkaki pemadam api, perilaku yang dilakukan oleh robot sesuai gambar 3.17 adalah sebagai berikut :

- ❖ Pertama robot berada di dalam ruangan ketika dinyalakan robot akan melakukan proses inisialisasi.
- ❖ Setelah proses inisialisasi robot akan melakukan perilaku navigasi atau *Behavior* Robot ke arah kanan, kiri, mundur atau serong untuk melakukan penelusuran ruangan.
- ❖ ketika melakukan penelusuran ruangan robot akan menelusuri ruangan untuk mencari garis, ketika ditemukan garis robot akan berperilaku mendeteksi garis, jika tidak menemukan garis robot akan terus melakukan penelusuran hingga bertemu garis.
- ❖ Ketika robot berperilaku mendeteksi garis, robot akan mendeteksi apakah garis yang dilalui bernilai 0 jika iya robot akan melintasi garis tersebut untuk keluar ruangan, jika tidak robot kembali menelusuri dan mendeteksi garis.
- ❖ Setelah robot keluar dari ruangan, robot akan berperilaku navigasi robot untuk bergerak mencari garis berikutnya, ketika garis dideteksi maka akan dilihat apakah garis itu bernilai 1 jika iya robot akan masuk ke ruangan untuk melakukan pendeteksian api.
- ❖ Jika dalam penelusuran ruangan robot mendeteksi ada boneka maka robot akan secara otomatis menghindari dari boneka dan bergerak mencari garis untuk masuk ke ruangan untuk melakukan penelusuran dan pendeteksian api.
- ❖ Jika robot mendeteksi adanya api maka robot akan melakukan penelusuran posisi api, jika posisi api sudah tepat robot mulai melakukan perilaku pemadaman api dengan pompa air. Jika tidak mendeteksi api robot akan terus berperilaku navigasi robot dan melakukan pendeteksian ke setiap ruangan.

### **3.8 Pengujian**

Pengujian robot berkaki *hexapod* dibagi menjadi beberapa bagian antara lain mencakup pengujian bagian, metode *Behaviour based control*, dan pengujian keseluruhan.

#### **3.8.1 Pengujian Perangkat keras**

Pada tahap ini digunakan untuk pengujian bagian – bagian perangkat keras yang digunakan untuk kinerja pada robot sehingga menunjang kehandalan dari robot itu sendiri.

##### **3.8.1.1 Ultrasonik *PING***

Pada tahap ini pengujian sensor berfungsi untuk mengetahui dari kemampuan sensor jarak tersebut. Serta mengetahui kemampuan yang sudah diketahui seperti keakuratan jarak pendeteksi pada sensor dengan jarak yang sesungguhnya. Serta sudut benda yang dapat dideteksi oleh sensor.

##### **3.8.1.2 Sensor *Sharp***

Pengujian pada sensor ini bertujuan untuk mengetahui robot dapat mengenali boneka dikarenakan sensor jarak *PING* tidak mampu mendeteksi adanya boneka secara maksimal. Pada pengujian didepan sensor akan diletakkan sebuah boneka yang sering digunakan oleh panitia lomba dengan jarak yang telah ditentukan serta posisi dan sudut yang nantinya pengujian telah ditentukan juga.

##### **3.8.1.3 Sensor Api *UVTron***

Pengujian ini ditujukan pada saat robot mendeteksi dirungan dengan jarak yang telah ditentukan yaitu pada jarak sekitar ruangan tersebut. Pada pengujian ini diharapkan robot sekali berjalan pada saat pendeteksi api robot dapat menangkap adanya api ketika diruangan tersebut

##### **3.8.1.4 Sensor Garis**

Pada tahap ini pengujian sensor garis digunakan untuk mengetahui warna putih atau garis yang telah ditentukan pada arena tersebut yang nantinya pengujian

akan dilakukan dengan pencahayaan yang cukup terang dan redup pada arena tersebut.

### **3.8.2 Pengujian Invers Kinematik pada robot *hexapode***

Pada pengujian ini nanti setiap kaki robot dilakukan pengujian apakah jarak sesuai yang dikehendaki semisal 2 cm apakah kaki robot bergerak dengan 2 cm tersebut.. selanjutnya robot dilakukan sekali jalan apakah jarak yang dikehendaki benar – benar sesuai.

### **3.8.3 Pengujian *Behavior Based Control***

Pengujian metode *Behavior based control* membahas tentang setiap perilaku dari metode tersebut. Pengujian berawal dari proses robot bergerak menentukan posisi lalu robot akan keluar ruangan dengan mendeteksi garis serta robot menelusuri ruangan dan robot masuk ruangan dan memindai api yang ada pada ruangan tersebut dan robot bergerak lagi keluar dari ruangan.

### **3.8.4 Pengujian Keseluruhan**

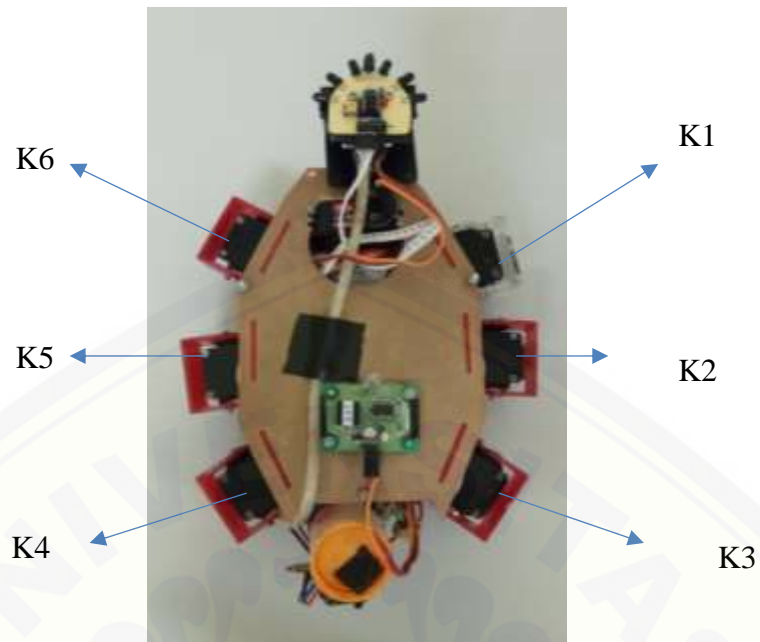
Pengujian keseluruhan meliputi pengujian bagian dan *Behavior based control*. Pada pengujian ini menguji cara kerja robot ketika berada didalam ruangan dan keluar ruangan hingga memadamkan api serta mengetahui *Behavior Robot* untuk mencapai tujuan. Pengujian ini juga mengacu pada parameter *metode behavior based control*.

## **3.9 Hasil Perancangan Alat**

Perancangan alat keseluruhan hampir menyerupai desain yang diinginkan hal ini terlihat pada gambar dibawah ini 3.18 , 3.19, dan 3.20. berikut merupakan hasil dari alat yang telah dibuat :

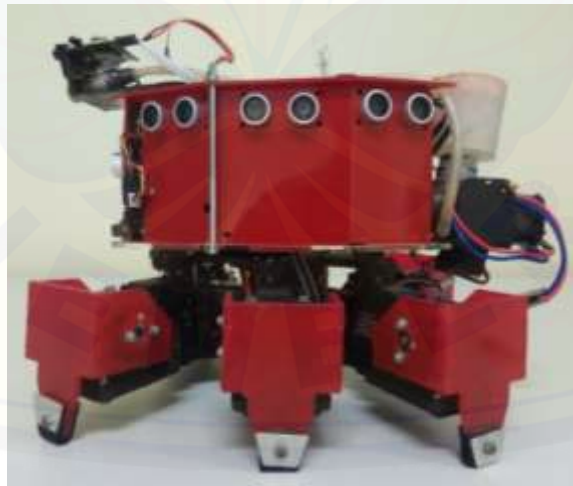
Pada bagian atas ini terlihat pada gambar 3.18 beberapa sensor yaitu terutama sensor api yang digunakan untuk mendeteksi api ketika sudah berada pada ruangan yang dilewati robot yang nantinya robot akan berperilaku memadamkan api jika menemukan titik api.





**Gambar 3.18** Robot tampak dari atas

Pada bagian samping terlihat pada gambar 3.19 beberapa sensor yang digunakan untuk navigasi robot dan mendeteksi halangan yang terdapat disekitarnya. Yang mana sensor tersebut yaitu sensor ultrasonik PING serta paling belakang terlihat tempat air dan pompa untuk memadamkan api.



**Gambar 3.19** Robot tampak dari Samping

Selanjutnya pada gambar terlihat pada gambar 3.20 bagian depan yang mana robot terlihat beberapa sensor navigasi juga namun disini terlihat terdapat dua sensor yang terlihat berbeda yang pertama sensor ultrasonik PING dan yang lain

sensor infrared yang digunakan untuk mendeteksi boneka ketika robot berjalan dan menemukan boneka. Robot akan bergerak sesuai perilaku yang diinginkan yaitu menghindari boneka tersebut.



**Gambar 3.20** Robot tampak Depan

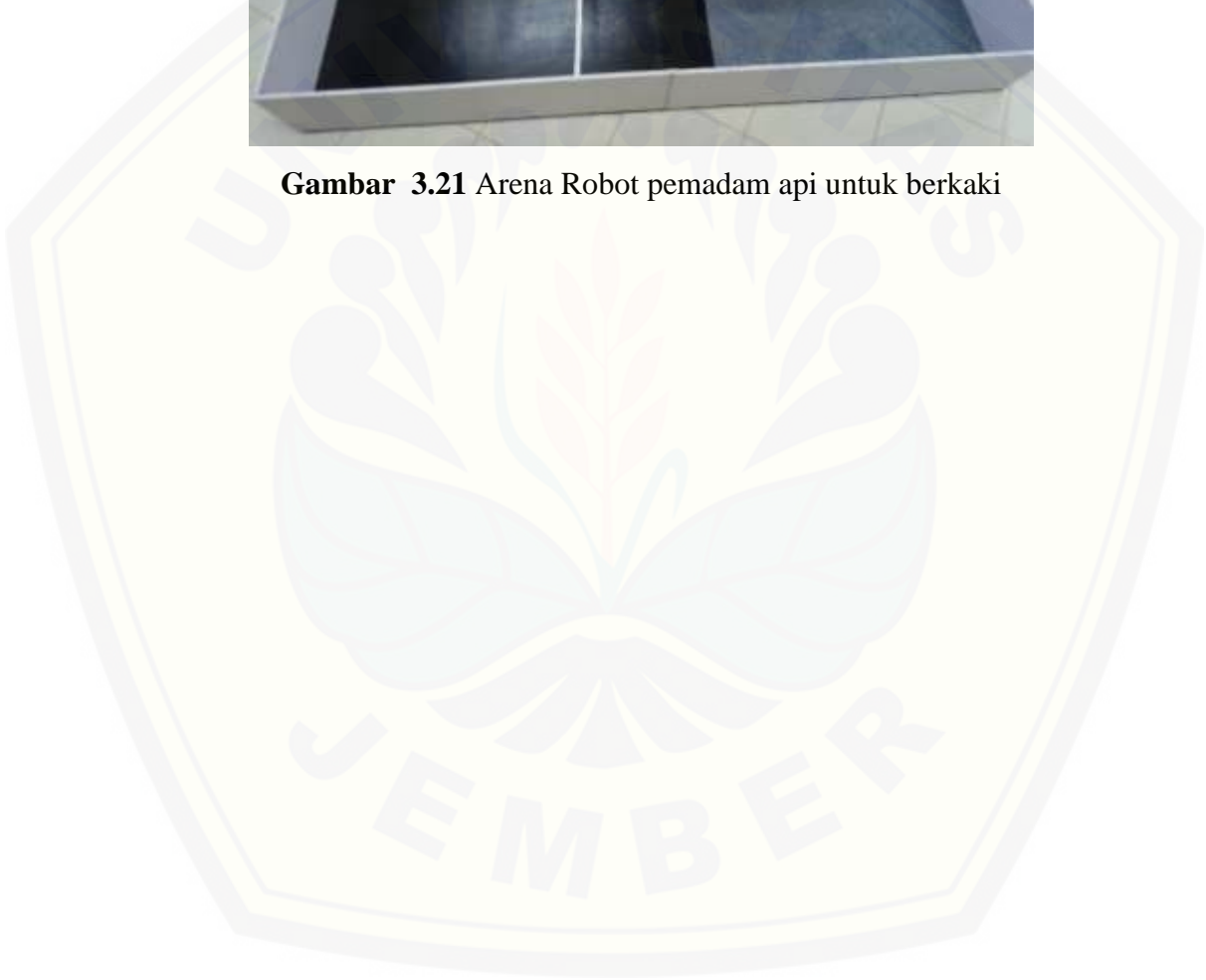
Pada desain awal robot *design* terlihat ada LCD yang digunakan namun pada jadinya robot tanpa menggunakan LCD tersebut dikarenakan beberapa hal yaitu tidak terlalu berfungsi robot menggunakan LCD karena robot sudah bekerja secara otomatis tanpa harus melakukan *tuning* lagi melalui robot sendiri karena dapat dilakukan monitor pada laptop.

Pada *design* juga terlihat kosong dari tampak samping pada robot setelah jadi robot tertutup sehingga terlihat tinggi namun tetap karena hanya pada bagian samping robot tertutupi akrilik agar terlihat rapi untuk elektrikalnya.

Selain robot ada komponen lain yaitu sebuah arena beserta bentuk kondisi yang ada. Lapangan tersebut merupakan suatu hal yang penting karena dengan lapangan tersebut robot dapat melakukan *Behavior* sesuai *rule* yang ada dengan lapangan berjenis kayu yang sifatnya *statis* dan sudah diatur bentuk ruangnya dengan jarak keseluruhan panjang 244 cm dan lebar 244 cm. Setiap pintu dan setiap ruangan berjarak 46 cm dan lorong yang digunakan untuk berjalan atau lintasan tersebut berjarak 46 cm.



**Gambar 3.21** Arena Robot pemadam api untuk berkaki



## BAB 5

### PENUTUP

#### 5.1 KESIMPULAN

1. Metode *behavior based control* merupakan *high control* hingga pada program yang dijalankan robot dapat memiliki pembelajaran yang sesuai dengan perilaku yang ditanamkan pada robot. Dalam penelitian robot dapat bekerja dengan baik dengan tingkat keberhasilan rata – rata 84 % dari metode yang digunakan.
2. Pada percobaan pengujian *behavior* koordinasi robot sebelum memadamkan api robot sengaja diberi halangan berupa boneka robot dapat mendeteksi perilaku keberadaan boneka namun robot juga akan mencari api setelah menghindari boneka tersebut dengan tingkat kesalahan 20 % dari 5 kali percobaan
3. Robot layak mengikuti kontes robot pemadam api karena pergerakan dan waktu dari robot sebelumnya lebih baik dari robot yang sekarang dengan selisih 30 detik dari percobaan perpindahan waktu dari tempat yang sama.

#### 5.2 SARAN

Perlunya diperhatikan pembacaan sensor untuk boneka agar stabil dan baik ketika mendeteksi boneka. Perlu diperhatikan juga tata letak dari sensornya dan perlunya control sensor boneka itu sendiri atau perlunya penggantian sensor yang spesifikasi di atasnya dan juga stabil. Perlunya penambahan Sensor agar pada saat memadamkan api langsung menuju pada api tanpa perlu menyusuri ruangan. Perlunya menambahkan sensor juga untuk mendeteksi ruangan yang dituju agar dapat kembali dalam home.

## DAFTAR PUSTAKA

- Arduino. 2017. Arduino Mega. <http://www.arduino.cc/en/Main/arduinoBoardMega>
- Braunl, Thomas. “*Embedded Robotics: Mobile Robot Design and Applications with Embedded System*”. Australia: Springer-Verlag Berlin Heidelberg.
- Gitakarma, M. S., G. Nurhayata, J. T. Elektronika, F. Teknik, dan U.P. Ganesha. 2013. “*Perancangan behavior-based robot dengan lgoritma fuzzy q-learning (fql) pada Sistem navigasi robot otonom*”. 2(1): 139-150.
- Handy Wicaksono, Prihastono, Khairul Anam, Rusdhianto Effendi, Indra Adji Sulistijono, Son Kuswadi, Achmad Jazidie, dan Mitsuji Sampei. 2008. “*Perancangan sistem navigasi otonom pada behavior based hexapod robot*”. Jurnal Teknik Elektro. 8(2):70-78.
- Hendi Wicaksono, dan Ari Bengnarly Tanjung. 2013. “*Pemrosesan SRF05, CMPS03, TPA81, Sistem Motor Secara MultiProsesor pada Robot KRPAI*”. 1-6.
- Hidayat, L., Iswanto, dan H. Muhammad. 2015. “*Perancangan robot pemadam api divisi senior berkaki*”. Jurnal semesta teknika. 14(2):112-116.SA
- Mulya. 2010. Sensor Jarak Sharp. [http://green-elektronik.blogspot.co.id/p/blog-page\\_10.html](http://green-elektronik.blogspot.co.id/p/blog-page_10.html)
- Parallax. 2017. Sensor Ping. <https://www.parallax.com>
- Prihlasit. 2011. Ultrasonic Sensor for Distance Measurement. <http://ap.urpi.fei.stuba.sk/sensorwiki/index.php/Acrob015>
- Sigit, Riyanto. 2007. “*Robotika Sensor dan Aktuator*”. Yogyakarta: Graha Ilmu Vol.63.
- Tzafestas, Spyros G. 2014. “*Introduction to Mobile Robot Control*”. Greece: School of Electrical and Computer Engineering National Technical University of Athens.

## LAMPIRAN

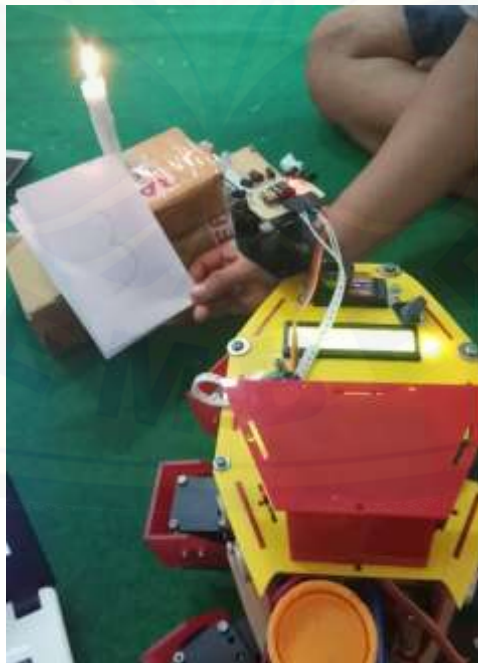
### Lampiran 1 Dokumentasi

1. Pengambilan data sensor garis



Gambar Dokumentasi Pengambilan Sensor Garis

2. Pengambilan data Sudur pembacaan Sensor Api



Gambar Dokumentasi Pengambilan Sudut Sensor Api

3. Pengambilan data sensor sharp



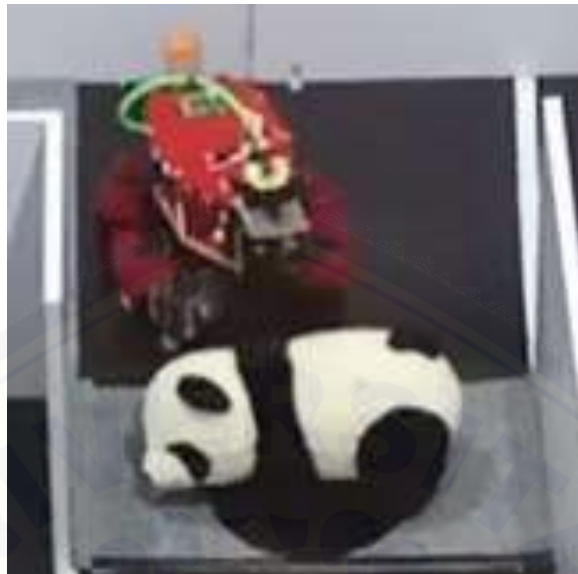
Gambar Dokumentasi Nilai jarak Sensor Sharp

4. Pengujian Keseluruhan



Gambar Dokumentasi Pengujian Keseluruhan

5. Pengujian Boneka



Gambar Dokumentasi Pengujian Terhadap Boneka



## Lampiran 2 Listing Program

### 1. Listing Program Arduino 1

```

#include <Wire.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
//-----sharp-----
#include <SharpIR.h>
#define ir A5 // kanan
#define ir1 A6 // depan
#define ir2 A7 // kiri
#define model 1080
boolean done = false;
SharpIR sharp(ir, 25, 93, model);
SharpIR sharp1(ir1, 25, 93, model);
SharpIR sharp2(ir2, 25, 93, model);
//-----PING-----
const int pin[7] = {22, 24, 26, 28, 30, 32, 34};
int pin52, pin50, pin48, pin40, pin38; // pin komunikasi
int K, L, M, N, O; // Sensor Bawah
int D = 50;
int jarak_sharp, jarak_sharp1, jarak_sharp2, titik_api1
= 0, titik_api2 = 0;
Servo myservoatas;
void setup() {
  Wire.begin();
  Serial.begin(9600);
  myservoatas.attach(44);
//-----sensor Sharp-----
  pinMode(A5, INPUT);
  pinMode(A6, INPUT); // depan
  pinMode(A7, INPUT); // kiri

```

```
//-----sensor garis-----
pinMode(A12, INPUT);
pinMode(A13, INPUT);
pinMode(A14, INPUT);
pinMode(A15, INPUT);
//-----Pin Komunikasi-----
pinMode(52, OUTPUT);
pinMode(50, OUTPUT);
pinMode(48, OUTPUT);
pinMode(40, OUTPUT);
pinMode(38, OUTPUT);
pinMode(12, INPUT_PULLUP);
}

void loop() {
  s_kanan_keluar();
}

int pink(int x) {
  long duration, cm;
  pinMode(pin[x], OUTPUT);
  digitalWrite(pin[x], LOW);
  delayMicroseconds(2);
  digitalWrite(pin[x], HIGH);
  delayMicroseconds(5);
  digitalWrite(pin[x], LOW);

  pinMode(pin[x], INPUT);
  duration = pulseIn(pin[x], HIGH);
  cm = duration / 29 / 2;
  delay(30);
```

```
    return cm;
}

void s_kanan_keluar() {
    int jarak_depan = pink(0);
    int jarak_samping_ka = pink(1);
    int jarak_kanan = pink(2);
    int jarak_samping_ka_belakang = pink(3);
    int jarak_samping_ki = pink(4);
    int jarak_kiri = pink(5);
    int jarak_samping_ki_belakang = pink(6);
    jarak_sharp = sharp.distance();
    jarak_sharp2 = sharp2.distance();
    //Serial.println(String(jarak_depan)+"
"+String(jarak_samping_ka)+"      "+String(jarak_kanan)+"
"+String(jarak_samping_ka_belakang)+"
"+String(jarak_samping_ki)+"      "+String(jarak_kiri)+"
"+String(jarak_samping_ki_belakang));
    K = analogRead(A0); //biru putih
    L = analogRead(A1); //putih
    M = analogRead(A2); //hijau putih
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        kanan();
        kanan();
        delay(1000);
        for (;;) {
```

```
        s_kanan_masuk();
    }
}
if (jarak_depan <= 17 ) { // hadap kiri
    kiri();
    kiri();
    kiri();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        kanan();
        kanan();
        for (;;) {
            s_kanan_masuk();
        }
    }
}
else if (jarak_depan >= 18 && jarak_samping_ka >= 18)
{ // serong kanan
    serong_kanan();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        kanan();
        kanan();
        delay(1000);
        for (;;) {
            s_kanan_masuk();
```

```
    }
  }
  else if ((jarak_sharp >= 8 && jarak_sharp <= 9) ||
(jarak_sharp1 >= 7 && jarak_sharp1 <= 8) || (jarak_depan
>= 17 && jarak_sharp >= 8 && jarak_sharp <= 9 &&
jarak_samping_ka >= 18)) {
    mundur();
    mundur();
    kiri();
    kiri();
  }
}
else if (jarak_depan >= 18 && (jarak_samping_ka >= 16
&& jarak_samping_ka <= 17)) { // serong kiri dikit
  serong_kiri_dikit();
  if (L <= 175 || M <= 175) {
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    kanan();
    kanan();
    delay(1000);
    for (;;) {
      s_kanan_masuk();
    }
  }
}
else if (jarak_depan >= 18 && jarak_samping_ka <= 15)
{ // serong kiri
```

```
serong_kiri();
if (L <= 175 || M <= 175) {
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    kanan();
    kanan();
    delay(1000);
    for (;;) {
        s_kanan_masuk();
    }
}
if (jarak_depan >= 18 && jarak_samping_ka >= 19 &&
jarak_kanan >= 20 ) { //serong kanan nemen
    serong_kanan_nemen();
    serong_kanan_nemen();
    serong_kanan_nemen();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        kanan();
        kanan();
        //hadap_kanan();
        delay(1000);
        for (;;) {
```

```
        s_kanan_masuk();
    }
}
else if ((jarak_sharp >= 8 && jarak_sharp <= 10) ||
(jarak_sharp2 >= 8 && jarak_sharp2 <= 10) || (jarak_depan
>= 17 && jarak_sharp >= 8 && jarak_sharp <= 10 &&
jarak_samping_ka >= 18)) {
    mundur();
    mundur();
    kiri();
    kiri();
}
}
}
void s_kanan_masuk() {
    int jarak_depan = pink(0);
    int jarak_samping_ka = pink(1);
    int jarak_kanan = pink(2);
    int jarak_samping_ka_belakang = pink(3);
    int jarak_samping_ki = pink(4);
    int jarak_kiri = pink(5);
    int jarak_samping_ki_belakang = pink(6);
    jarak_sharp = sharp.distance();
    jarak_sharp2 = sharp2.distance();
    Serial.println(String(jarak_depan) + " " +
String(jarak_samping_ka) + " " + String(jarak_kanan) +
" " + String(jarak_samping_ka_belakang) + " " +
String(jarak_samping_ki) + " " + String(jarak_kiri) + "
" + String(jarak_samping_ki_belakang));
    K = analogRead(A0); //biru putih
```

```
L = analogRead(A1); //putih
M = analogRead(A2); //hijau putih
if (L <= 175 || M <= 175) {
    kanan();
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    lurus();
    berdiri();
    scan_api_garis();
}
if (jarak_depan <= 19 ) { // hadap kiri
    kiri();
    kiri();
    kiri();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        berdiri();
    }
}
```



```
        scan_api_garis();
    }
}
else if (jarak_depan >= 20 && jarak_samping_ka >= 18)
{ // serong kanan
    serong_kanan();
    if (L <= 175 || M <= 175) {
        kanan();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        berdiri();
        scan_api_garis();
    }
    else if ((jarak_sharp >= 9 && jarak_sharp <= 10) ||
(jarak_sharp2 >= 9 && jarak_sharp2 <= 10) || (jarak_depan
>= 17 && jarak_sharp2 >= 9 && jarak_sharp2 <= 10 &&
jarak_samping_ka >= 18)) {
        mundur();
        mundur();
        mundur();
        mundur();
        hadap_kiri();
        hadap_kiri();
        hadap_kiri();
    }
}
```

```
        hadap_kiri();
        hadap_kiri();
    }
}
else if (jarak_depan >= 20 && (jarak_samping_ka >= 16
&& jarak_samping_ka <= 17)) { // serong kiri dikit
    serong_kiri_dikit();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        berdiri();
        scan_api_garis();
    }
}
else if (jarak_depan >= 20 && jarak_samping_ka <= 15)
{ // serong kiri
    serong_kiri();
    if (L <= 175 || M <= 175) {
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
    }
}
```

```
        lurus();
        lurus();
        lurus();
        berdiri();
        scan_api_garis();
    }
}
if (jarak_depan >= 20 && jarak_samping_ka >= 19 &&
jarak_kanan >= 20 ) { //serong kanan nemen
    serong_kanan_nemen();
    serong_kanan_nemen();
    serong_kanan_nemen();
    serong_kanan_nemen();
    serong_kanan_nemen();
    serong_kanan_nemen();
    if (L <= 175 || M <= 175) {
        kanan();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        berdiri();
        scan_api_garis();
    }
    else if ((jarak_sharp >= 8 && jarak_sharp <= 10) ||
(jarak_sharp2 >= 9 && jarak_sharp2 <= 10) || (jarak_depan
```

```
>= 20 && jarak_sharp2 >= 9 && jarak_sharp2 <= 10 &&
jarak_samping_ka >= 18)) {
    mundur();
    mundur();
    mundur();
    mundur();
    mundur();
    delay(100);
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
}
}
else if ((jarak_sharp >= 9 && jarak_sharp <= 10) ||
(jarak_sharp2 >= 9 && jarak_sharp2 <= 10) || (jarak_depan
>= 20 && jarak_sharp2 >= 9 && jarak_sharp2 <= 10 &&
jarak_samping_ka >= 18)) {
    mundur();
    mundur();
    mundur();
    mundur();
    delay(100);
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
    hadap_kiri();
}
```

```
        kiri();
    }
}

void s_kiri() {
    cek_ruang();
    int jarak_depan = pink(0);
    int jarak_samping_ka = pink(1);
    int jarak_kanan = pink(2);
    int jarak_samping_ka_belakang = pink(3);
    int jarak_samping_ki = pink(4);
    int jarak_kiri = pink(5);
    int jarak_samping_ki_belakang = pink(6);
    jarak_sharp2 = sharp2.distance();
    jarak_sharp1 = sharp1.distance();
    //Serial.println(String(jarak_depan)+"
    "+String(jarak_samping_ka)+"      "+String(jarak_kanan)+"
    "+String(jarak_samping_ka_belakang)+"
    "+String(jarak_samping_ki)+"      "+String(jarak_kiri)+"
    "+String(jarak_samping_ki_belakang));
    K = analogRead(A0); //biru putih
    L = analogRead(A1); //putih
    M = analogRead(A2); //hijau putih
    if (L <= 175 || M <= 175) {
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
        digitalWrite(48, HIGH);
    }
    if (jarak_depan <= 17 ) { // hadap kanan
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
    }
}
```

```
digitalWrite(48, LOW);
if (L <= 175 || M <= 175) {
    digitalWrite(52, HIGH);
    digitalWrite(50, LOW);
    digitalWrite(48, HIGH);
}
}
else if (jarak_depan >= 18 && jarak_samping_ki >= 20)
{ // serong kiri
    digitalWrite(52, LOW);
    digitalWrite(50, HIGH);
    digitalWrite(48, LOW);
    if (L <= 175 || M <= 175) {
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
        digitalWrite(48, HIGH);
    }
    else if ((jarak_sharp >= 8 && jarak_sharp <= 9) ||
(jarak_sharp1 >= 7 && jarak_sharp1 <= 8) || (jarak_depan
>= 17 && jarak_sharp >= 8 && jarak_sharp <= 9 &&
jarak_samping_ki >= 18)) {
        digitalWrite(52, HIGH);
        digitalWrite(50, HIGH);
        digitalWrite(48, HIGH);
    }
}
}
else if (jarak_depan >= 18 && (jarak_samping_ki >= 18
&& jarak_samping_ki <= 19)) { // seronkanan dikit
    digitalWrite(52, HIGH);
    digitalWrite(50, HIGH);
    digitalWrite(48, LOW);
```

```
    if (L <= 175 || M <= 175) {
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
        digitalWrite(48, HIGH);
    }
}
else if (jarak_depan >= 18 && jarak_samping_ki <= 17)
{ // serong kanan
    digitalWrite(52, LOW);
    digitalWrite(50, LOW);
    digitalWrite(48, HIGH);
    if (L <= 175 || M <= 175) {
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
        digitalWrite(48, HIGH);
    }
}
    if (jarak_depan >= 18 && jarak_samping_ki >= 21 &&
jarak_kiri >= 21 ) { //serong kiri nemen
    digitalWrite(52, LOW);
    digitalWrite(50, HIGH);
    digitalWrite(48, HIGH);
    if (L <= 175 || M <= 175) {
        digitalWrite(52, HIGH);
        digitalWrite(50, LOW);
        digitalWrite(48, HIGH);
    }
}
    else if ((jarak_sharp >= 8 && jarak_sharp <= 9) ||
(jarak_sharp1 >= 7 && jarak_sharp1 <= 8) || (jarak_depan
>= 17 && jarak_sharp >= 8 && jarak_sharp <= 9 &&
jarak_samping_ki >= 18)) {
```

```
        digitalWrite(52, HIGH);
        digitalWrite(50, HIGH);
        digitalWrite(48, HIGH);
    }
}
else if ((jarak_sharp2 >= 8 && jarak_sharp2 <= 9) ||
(jarak_sharp1 >= 7 && jarak_sharp1 <= 9) || (jarak_depan
>= 17 && jarak_sharp2 >= 8 && jarak_sharp2 <= 9 &&
jarak_samping_ki >= 18)) {
    digitalWrite(52, HIGH);
    digitalWrite(50, HIGH);
    digitalWrite(48, HIGH);
}
}
void scan_api_garis() {
    int xg = 0;
    for (int yg = 0; yg <= 180; yg++) {
        myservoatas.write(yg);
        if (digitalRead(12) == LOW && xg == 0) {
            delay(50);
            titik_apil = yg;
            xg = 1;
        }
        delay(30);
    }
    int sudut1 = titik_apil;
    if (xg == 1) {
        myservoatas.write(sudut1);
        delay(50);
        for (;;) {
            menuju_api();
        }
    }
}
```



```
    }  
  }  
  else {  
    myservoatas.write(80);  
    berdiri();  
    delay(2000);  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    hadap_kiri();  
    for (;;) {  
      s_kanan_keluar();  
    }  
  }  
  titik_apil = 0;  
}  
  
void scan_api_ruang() {  
  int x = 0;  
  myservoatas.write(180);  
  delay(500);  
  for (int y = 180 ; y >= 10; y--) {  
    delay(20);  
    myservoatas.write(y);  
    if (digitalRead(A14) == HIGH && x == 0) {  
      titik_apil = y;  
      x = 1;  
    }  
  }  
}
```

```
    }
    delay(20);
}
for (int y1 = 10; y1 <= 180; y1++) {
    delay(20);
    myservoatas.write(y1);
    if (digitalRead(A14) == HIGH && x == 1) {
        titik_api2 = y1;
        x = 2;
    }
    delay(20);
}
int sudut = (titik_api1 + titik_api2) / 2;
delay(20);
if (x == 2) {
    for (;;) {
        delay(30);
        myservoatas.write(sudut);
        delay(500);
        digitalWrite(11, HIGH);
        delay(5500);
        digitalWrite(11, LOW);
        x = 3;
        break;
        //delay(500);
    }
}
if (x == 3) {
    hadap_kanan();
    delay(1000);
    for (;;) {
```

```
        mundur();
        mundur();
        berdiri();
        delay(100000);
    }
}
digitalWrite(11, LOW);
myservoatas.write(120);
titik_api1 = 0;
titik_api2 = 0;
}

void menuju_api() {
    int jarak_depan = pink(0);
    int jarak_samping_ka = pink(1);
    int jarak_kanan = pink(2);
    int jarak_samping_ka_belakang = pink(3);
    int jarak_samping_ki = pink(4);
    int jarak_kiri = pink(5);
    int jarak_samping_ki_belakang = pink(6);
    jarak_sharp = sharp.distance();
    jarak_sharp1 = sharp1.distance();
    int flame1 = analogRead(A13); // kanan
    int flame2 = analogRead(A12); // kiri
    int flame3 = analogRead(A15); // tengah
    K = analogRead(A0); //biru putih
    L = analogRead(A1); //putih
    M = analogRead(A2); //hijau putih
    myservoatas.write(90);
    if (L <= 175 || M <= 175) {
        mundur();
    }
}
```

```
mundur();
mundur();
mundur();
mundur();
mundur();
mundur();
mundur();
mundur();
mundur();
mundur();
hadap_kiri();
hadap_kiri();
hadap_kiri();
hadap_kiri();
hadap_kiri();
hadap_kiri();
hadap_kiri();
hadap_kiri();
kiri();
}
if (jarak_depan <= 17) {
    kiri();
    kiri();
    kiri();
    if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
        berdiri();
        scan_api_ruang();
    }
}
else if (jarak_depan >= 18 && jarak_samping_ka >= 18)
{
```

```
    serong_kanan();
    if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
        berdiri();
        scan_api_ruang();
    }
}
else if (jarak_depan >= 18 && jarak_samping_ka <= 15)
{
    serong_kiri();
    if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
        berdiri();
        scan_api_ruang();
    }
}
else if (jarak_depan >= 18 && (jarak_samping_ka >= 16
&& jarak_samping_ka <= 17)) {
    serong_kiri_dikit();
    if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
        berdiri();
        scan_api_ruang();
    }
}
if (jarak_depan >= 18 && jarak_samping_ka >= 19 &&
jarak_kanan >= 20) {
    serong_kanan_nemen();
    if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
        berdiri();
        scan_api_ruang();
    }
}
}
```

```
If ((jarak_sharp >= 8 && jarak_sharp <= 9) ||
(jarak_sharp2 >= 7 && jarak_sharp2 <= 9) || (jarak_depan
>= 17 && jarak_sharp >= 8 && jarak_sharp <= 9 &&
jarak_samping_ka >= 18)) {
    mundur();
    kiri();
    kiri();
    kiri();
}
if (flame1 <= 40 || flame2 <= 40 || flame3 <= 40) {
    berdiri();
    scan_api_ruang();
}
}

void berdiri() {
    digitalWrite(52, HIGH);
    digitalWrite(50, LOW);
    digitalWrite(48, LOW);
    digitalWrite(40, LOW);
    digitalWrite(38, LOW);
    delay(100);
}

void lurus() {
    digitalWrite(52, LOW);
    digitalWrite(50, HIGH);
    digitalWrite(48, LOW);
    digitalWrite(40, LOW);
    digitalWrite(38, LOW);
    delay(200);
}
```

```
void serong_kanan() {
    digitalWrite(52, LOW);
    digitalWrite(50, LOW);
    digitalWrite(48, HIGH);
    digitalWrite(40, LOW);
    digitalWrite(38, LOW);
}

void serong_kanan_nemen() {
    digitalWrite(52, LOW);
    digitalWrite(50, LOW);
    digitalWrite(48, LOW);
    digitalWrite(40, HIGH);
    digitalWrite(38, LOW);
}

void serong_kanan_dikit() {
    digitalWrite(52, LOW);
    digitalWrite(50, LOW);
    digitalWrite(48, LOW);
    digitalWrite(40, LOW);
    digitalWrite(38, HIGH);
}

void serong_kiri() {
    digitalWrite(52, HIGH);
    digitalWrite(50, HIGH);
    digitalWrite(48, LOW);
    digitalWrite(40, LOW);
    digitalWrite(38, LOW);
    delay(D);
}

void serong_kiri_nemen() {
    digitalWrite(52, LOW);
```

```
digitalWrite(50, HIGH);
digitalWrite(48, HIGH);
digitalWrite(40, LOW);
digitalWrite(38, LOW);
delay(D);
}
void serong_kiri_dikit() {
  digitalWrite(52, LOW);
  digitalWrite(50, LOW);
  digitalWrite(48, HIGH);
  digitalWrite(40, HIGH);
  digitalWrite(38, LOW);
  delay(D);
}
void kiri() {
  digitalWrite(52, LOW);
  digitalWrite(50, LOW);
  digitalWrite(48, LOW);
  digitalWrite(40, HIGH);
  digitalWrite(38, HIGH);
  delay(200);
}
void kanan() {
  digitalWrite(52, HIGH);
  digitalWrite(50, LOW);
  digitalWrite(48, LOW);
  digitalWrite(40, LOW);
  digitalWrite(38, HIGH);
  delay(100);
}
void mundur() {
```



```
digitalWrite(52, HIGH);
digitalWrite(50, LOW);
digitalWrite(48, HIGH);
digitalWrite(40, LOW);
digitalWrite(38, LOW);
delay(200);
}
void hadap_kiri() {
  digitalWrite(52, HIGH);
  digitalWrite(50, LOW);
  digitalWrite(48, LOW);
  digitalWrite(40, HIGH);
  digitalWrite(38, LOW);
  delay(200);
}
void hadap_kanan() {
  digitalWrite(52, LOW);
  digitalWrite(50, LOW);
  digitalWrite(48, HIGH);
  digitalWrite(40, LOW);
  digitalWrite(38, HIGH);
}
void baca_flame() {
  int flame1 = analogRead(A12); //kiri
  int flame2 = analogRead(A13); //kanan
  int flame3 = analogRead(A15); //tengah 2
  int flame4 = digitalRead(A14); // high low
  Serial.println(String(flame1) + " " + String(flame3)
+ " " + String(flame2) + " " + String(flame4));
  delay(100);
}
```