



**PENGGUNAAN METODE *FINITE STATE MACHINE* (FSM)  
UNTUK MENGONTROL ROBOT BERKAKI PEMADAM API**

**SKRIPSI**

Oleh:

**Richard Juliano  
NIM 141910201041**

**PROGRAM STUDI STRATA I TEKNIK ELEKTRO**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS JEMBER**

**2018**



**PENGGUNAAN METODE *FINITE STATE MACHINE* (FSM)  
UNTUK MENGONTROL ROBOT BERKAKI PEMADAM API**

**SKRIPSI**

Oleh:

**Richard Juliano  
NIM 141910201041**

**PROGRAM STUDI STRATA I TEKNIK ELEKTRO**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS JEMBER**

**2018**

## PERSEMBAHAN

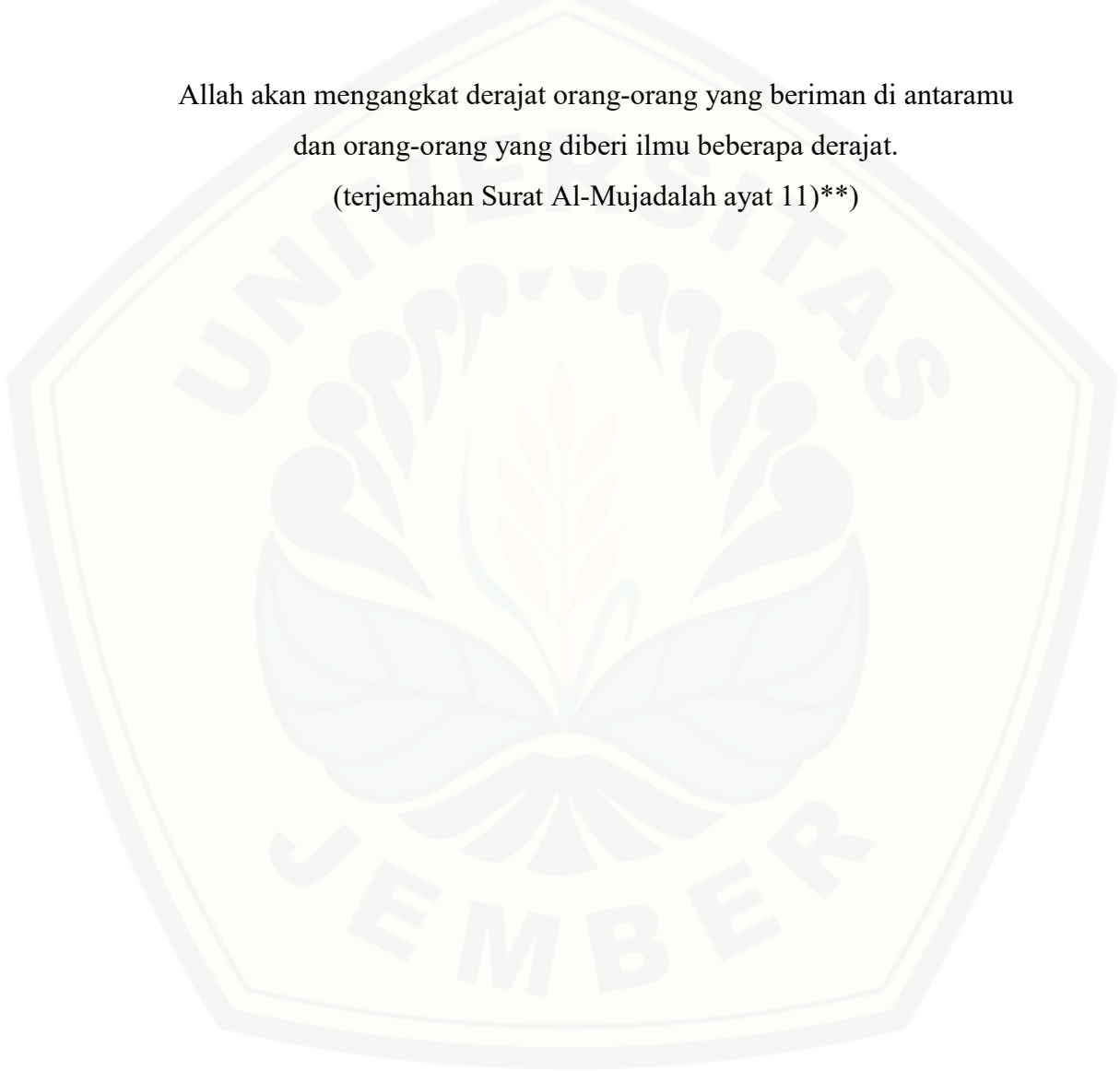
Skripsi ini saya persembahkan untuk:

1. Orang tua saya tercinta, Yulianto S.E dan Endah Yulia Dewi S.Ag yang selalu memberikan kasih sayang, doa tiada henti, serta pengorbanan yang dilakukan setiap waktu;
2. Adikku Shifa Adelia Septi Duana yang memberikan dukungan moral;
3. Para guru-guru sejak taman kanak-kanak hingga perguruan tinggi yang telah memberikan ilmu dan mendidik saya hingga menjadi manusia yang berilmu;
4. Semua teman-teman Pondok Bambu dan saudara – saudaraku penangan yang selalu memberikan motivasi terutama tante Martini;
5. Saudara-saudara KETEK UJ dan Robotika UNEJ yang telah memberikan pengalaman serta kehangatan kekeluargaan;
6. Almamater Fakultas Teknik Universitas Jember.

## MOTTO

Barangsiapa yang berjihad(bersungguh-sungguh), sesungguhnya jihadnya itu adalah untuk dirinya sendiri. (terjemahan surat Al-Ankabut ayat 6)\*)

Allah akan mengangkat derajat orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat.  
(terjemahan Surat Al-Mujadalah ayat 11)\*\*)



---

\*) \*\*) Syaamil Quran. 2010. Hijaz Terjemah Tafsir Per Kata. Bandung: Syaamil Quran

## PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Richard Juliano

NIM :142010101041

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Berkaki 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 atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi. Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 20 Juli 2018  
Yang menyatakan,

Richard Juliano  
NIM 141910201041

**SKRIPSI**

PENGGUNAAN METODE *FINITE STATE MACHINE* (FSM) UNTUK  
MENGONTROL ROBOT BERKAKI PEMADAM API

Oleh  
Richard Juliano  
NIM 141910201041

Pembimbing :

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

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

PENGESAHAN

Skripsi berjudul “Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Berkaki Pemadam Api” karya Richard Juliano telah diuji dan disahkan pada:

Hari, Tanggal : Jumat, 20 Juli 2018

Tempat : Fakultas Teknik Universitas Jember

Tim penguji :

Ketua,

Anggota I,

Widya Cahyadi, S.T., M.T.  
NIP 198511102014041001

Khairul Anam, ST.,MT., Ph.D.  
NIP 197804052005011002

Anggota II,

Anggota III,

Ike Fibriani, S.T., M.T.  
NIP 198002072015042001

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

Mengesahkan  
Dekan,

Dr. Ir. Entin Hidayah, M.U.M.  
NIP 19661215 199503 2001

## RINGKASAN

**Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Berkaki Pemadam Api, Richard Juliano, 141910201041;2018: 103 halaman; Jurusan Teknik Elektro Universitas Jember.**

Kontes Robot Pemadam Api Indonesia adalah sebuah perlombaan antar Universitas se-Indonesia dibawah naungan Kemenristekdikti dengan Robot berkaki pemadam api dituntut untuk menemukan api dan memadamkan api yang berupa lilin. Robot Berkaki ini berjalan menyusuri Arena yang telah disediakan oleh panitia dengan berbagai macam aturan yang dibuat. Permasalahan yang sering terjadi dalam perlombaan Kontes Robot Indonesia Pemadam Api Divisi Berkaki Universitas Jember yaitu pada saat robot berkaki berjalan masih kurang sempurna dan masih terlalu lama untuk menemukan api.

Untuk mengatasi masalah tersebut dibutuhkan sebuah metode, *Finite State Machine* adalah solusi yang dapat dipilih dikarenakan sistem kontrol ini menggambarkan tingkah laku dengan menggunakan tiga hal yaitu: *State* (Keadaan), *Event* (Kejadian), *Action* (aksi). *Policy Learning* adalah salah satu aspek dari *Finite State Machine* yang digunakan pada penelitian kali ini dengan keputusan suatu *state* pindah ke *state* selanjutnya, menggunakan data *feedback* pada sensor. Pada saat robot telah menjalankan *state* dan mendapat keluaran yang diharapkan, maka akan menuju *state* selanjutnya hingga menuju akhir *state*.

Penelitian kali ini bertujuan untuk : (1) Merancang robot berkaki enam supaya berjalan dengan lancar menggunakan motor servo. (2) Menjelaskan kinerja robot menggunakan metode *Finite State Machine*. Robo berkaki ini dirancang menggunakan dua buah mikrokontrol arduino mega 2560 dimana kedua mikrokontrol dibagi tugasnya untuk mikrokontrol 1 yang posisinya berada diatas digunakan sebagai masukan dari beberapa sensor dan mikronkontrol 2 yang posisinya berada dibawah digunakan sebagai keluaran untuk menggerakkan servo.

Pengujian pada robot kali ini mencakup beberapa hal yaitu Pengujian Sensor, Pengujian State dan Pengujian Keseluruhan. Setelah melakukan pengujian



dapat diketahui metode *Finite State Machine* dengan menggunakan aspek *Policy Learning* berhasil berjalan dengan lancar dengan persentase keberhasilan 70% dan dapat diimplementasikan pada Kontes Robot Pemadam Api Berkaki Indonesia.



## PRAKATA

*Bismillahirrohmanirrohim*

Puji syukur ke hadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga laporan skripsi yang berjudul “Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Berkaki Pemadam Api” 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 Moch. Gozali, S.T., M.T. selaku Ketua Prodi S1 Fakultas Teknik Elektro Universitas Jember.
7. Bapak Widya Cahyadi, 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. Aini Mei Tanti, Khana Nurfadhila, Rifaldi Fachrizal Rachman, Tear Norman, Yolanda D.N, Melsa Netika, Ahmad Wahyu Tri Utama, Aden Tia, Putra Suci Bachtiar, Dhamas Agung, Faiq Aprilian, Tigo dan 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 dikemudian hari.

Jember, 20 Juli 2018

Penulis

DAFTAR ISI

LEMBAR JUDUL.....	i
LEMBAR PERSEMBAHAN.....	ii
LEMBAR MOTTO.....	iii
LEMBAR PERNYATAAN.....	iv
SKRIPSI.....	v
LEMBAR PENGESAHAN.....	vi
LEMBAR RINGKASAN.....	vii
LEMBAR PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
<b>BAB 1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
<b>BAB 2. TINJAUAN PUSTAKA.....</b>	<b>4</b>
2.1 Penelitian Terkait.....	4
2.2 Arduino Mega 2560.....	4
2.3 Motor Servo.....	6
2.4 Sensor Ultrasonik PING.....	6
2.5 Sensor Api.....	7
2.6 CMPS03.....	9
2.7 <i>Finite State Machine</i> .....	10
<b>BAB 3. METODOLOGI PELAKSANAAN DATA.....</b>	<b>12</b>
3.1 Waktu dan Tempat Kegiatan.....	12
3.2 Ruang Lingkup Kegiatan.....	12

<b>3.3 Jenis dan Sumber Data</b> .....	13
<b>3.4 Metode Pengumpulan Data</b> .....	13
<b>3.5 Perancangan Alat</b> .....	14
<b>3.5.1 Perancang Desain Alat</b> .....	14
<b>3.5.2 Perancangan Perangkat Keras</b> .....	15
<b>3.5.3 Metode <i>Finite State Machine</i></b> .....	20
<b>3.5.4 Perancangan Sistem Kontrol</b> .....	21
<b>3.5.5 <i>Flowchart</i> Sistem</b> .....	23
<b>3.6 Pengujian</b> .....	28
3.6.1 Pengujian Sensor.....	28
3.6.2 Pengujian <i>State</i> .....	28
3.6.3 Pengujian Keseluruhan .....	30
<b>3.7 Hasil Perancangan Alat</b> .....	33
<b>4.1 Pengujian Sensor</b> .....	37
4.1.2 Sensor Ultrasonik.....	37
4.1.2 Sensor Photodiode.....	39
4.1.3 Sensor UV Tron .....	41
4.1.4 Sensor Kompas .....	43
<b>4.2 Pengujian State</b> .....	43
4.2.1 Pengujian Robot Maju,Pindai dinding,Telusur diruang start, dan Keluar Ruang. .....	43
4.2.2 Pengujian Telusur Lorong.....	46
4.2.3 Pengujian Menuju Api dan Maju Api .....	47
4.2.4 Pengujian Padamkan Api.....	49
4.2.5 Pengujian <i>Home</i> .....	49
4.3 Pengujian Keseluruhan .....	51
4.4 Pengujian Kuantitatif .....	56
5.1 Kesimpulan .....	58
5.2 Saran .....	58
<b>DAFTAR PUSTAKA</b> .....	<b>59</b>
<b>LAMPIRAN</b> .....	<b>60</b>
Lampiran 1 Dokumentasi.....	60

Lampiran 2 Listing Program..... 62



**DAFTAR GAMBAR**

Gambar 2. 1 Robot Berkaki (Sumber: Santoso et al., 2017).....	4
Gambar 2.2 Arduino Mega 2560 (sumber: arduino.cc) .....	5
Gambar 2.3 Motor Servo (sumber: Hidayat et al., 2011).....	6
Gambar 2.4 Sensor Ultrasonik (sumber: Arief, 2011 ) .....	7
Gambar 2.5 Sensor Api UV Tron (Sumber: Suryatini et al., 2013).....	8
Gambar 2.6 Sensor Kompas CMPS03 (sumber: http://tokorobotik.com/CMPS03,2018).....	9
Gambar 2.7 Tiga learning FSM (Sumber: Grollman & Jenkins, 2010).....	10
Gambar 3.1 Mekanik robot berkaki 6 .....	14
Gambar 3.2 Desain Kaki .....	14
Gambar 3.3 Desain Peletakan Sensor Ultrasonik .....	15
Gambar 3.4Blok diagram untuk navigasi robot berkaki .....	16
Gambar 3.5 Rangkaian Ultrasonik.....	17
Gambar 3.6 Rangkaian UV TRON .....	18
Gambar 3.7 Rangkaian Photodiode.....	19
Gambar 3.8 Policy Learning .....	20
Gambar 3.9 Diagram FSM 1 .....	21
Gambar 3.10 Diagram FSM 2.....	21
Gambar 3.11Flowchart Sistem Keseluruhan.....	23
Gambar 3.12Flowchart sistem pindai dinding .....	24
Gambar 3.13Flowchart sistem telusur lorong .....	25
Gambar 4.14 Flowchart sistem padamkan api .....	26
Gambar 3.15 <i>Flowchart</i> sistem <i>Home</i> .....	27
Gambar 3.16 Rencana Pengujian Robot maju hingga keluar ruang .....	29
Gambar 3.17 Rencana Pengujian Telusur lorong .....	29
Gambar 3.18 Rencana Pengujian Robot Maju api .....	30
Gambar 3.19 Skema Pengujian 1 .....	31
Gambar 3.20 Skema Pengujian 2 .....	31
Gambar 3.21 Skema Pengujian 3 .....	32
Gambar 3.22 Skema pengujian 4 .....	32
Gambar 3.23 Robot tampak depan.....	33
Gambar 3.24 Robot tampak samping.....	33
Gambar 3.25 Robot tampak atas .....	34
Gambar 3.26 Lintasan Robot Berkaki Pemadam Api.....	35
Gambar 3.27 Pencahayaan Arena Robot .....	36

Gambar 4.1 Grafik Pembacaan Sensor Ultrasonik .....	38
Gambar 4.2 Grafik Pembacaan sensor photodiode lux 81 .....	39
Gambar 4.3 Grafik Pembacaan sensor photodiode lux 356.....	40
Gambar 4.4 Arah Mata Angin.....	43
Gambar 4.5 Proyeksi Pengujian 1 .....	44
Gambar 4.6 Proyeksi Pengujian 2.....	44
Gambar 4.7 Proyeksi pengujian lorong 1.....	46
Gambar 4.8 Proyeksi Pengujian Lorong 2 .....	46
Gambar 4.9 Letak awal pengujian menuju api dan maju api.....	48
Gambar 4.10 Letak akhir pengujian menuju api dan maju api .....	48
Gambar 4.11 Letak awal pengujian menuju api dan maju api.....	48
Gambar 4.12 Letak akhir pengujian menuju api dan maju api .....	48
Gambar 4.13 Letak awal pengujian menuju api dan maju api.....	48
Gambar 4.14 Letak akhir pengujian menuju api dan maju api .....	48
Gambar 4.15 Letak awal pengujian robot memadamkan.....	49
Gambar 4.16 Letak akhir pengujian robot memadamkan.....	49
Gambar 4.17 Proyeksi pengujian pulang home 3 .....	50
Gambar 4.18 Proyeksi pengujian pulang home 1 .....	50
Gambar 4.19 Proyeksi pengujian pulang home 2 .....	51
Gambar 4.20 Proyeksi pengujian pulang home 3 .....	51
Gambar 4.21 Pengujian Keseluruhan 1.....	52
Gambar 4.22 Pengujian Keseluruhan 2.....	53
Gambar 4.23 Pengujian Keseluruhan 3.....	53
Gambar 4.24 Pengujian Keseluruhan 4.....	54
Gambar A. 1 Dokumentasi Pengambilan Data Sudut Posisi Api. ....	60
Gambar A. 2 Dokumentasi Pengambilan Data Jarak.....	60
Gambar A. 3 NilaiLux Meter .....	61



**DAFTAR TABEL**

Tabel 2.1 Spesifikasi Arduino Mega 2560.....	5
Tabel 2.2 Spesifikasi Flame Sensor UVTRON.....	8
Tabel 3.1 Rencana Kegiatan Tugas Akhir.....	12
Tabel 4.1 Data Pembacaan Sensor PING.....	38
Tabel 4.2 Data pembacaan Photodiode.....	39
Tabel 4.3 Data Pembacaan Jarak Sensor UV Tron.....	41
Tabel 4.4 Data Pembacaan Sudut Sensor UV Tron.....	42
Tabel 4.5 Pembacaan Sudut Sensor kompas.....	43
Tabel 4.6 proyeksi posisi dan waktu robot.....	44
Tabel 4.7 Proyeksi telusur lorong dan waktu.....	46
Tabel 4.8 Proyeksi menuju api dan maju api.....	48
Table 4.9 Proyeksi Robot Padamkan Api.....	49
Tabel 4.10 Proyeksi Robot Home.....	50
Tabel 4.11 Pengujian Keseluruhan.....	55
Tabel 4.12 Data Pengujian Kuantitatif.....	57

**DAFTAR LAMPIRAN**

Lampiran 1 Dokumentasi.....	60
Lampiran 2 Listing Program.....	62



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Robot merupakan hal yang tidak asing ditelinga bahkan sudah banyak yang mampu membuat robot bagi mahasiswa elektro khususnya konsentrasi elektronika. Robot berkembang begitu pesat dengan berbagai macam sistem yang baru. Robot berkaki merupakan salah satu jenis robot yang banyak dikembangkan. Robot ini bergerak menggunakan motor servo sebagai kakinya.

Kontes Robot Pemadam Api Indonesia adalah sebuah perlombaan antar Universitas se-Indonesia dibawah naungan Kemenristekdikti dengan Robot berkaki pemadam api dituntut untuk menemukan api dan memadamkan api yang berupa lilin. Robot Berkaki ini berjalan menyusuri Arena yang telah disediakan oleh panitia dengan berbagai macam aturan yang dibuat. Permasalahan yang sering terjadi dalam perlombaan Kontes Robot Indonesia Pemadam Api Divisi Berkaki Universitas Jember yaitu pada saat robot berkaki berjalan masih kurang sempurna dan masih terlalu lama untuk menemukan api.

Pada penelitian sebelumnya tentang robot berkaki enam mengikuti dinding menggunakan metode kontrol proporsional derivatif, mampu menjejak dinding dengan nilai  $K_p=4$  dan  $K_d=4$  dengan sudut ketajaman tikungan antara  $70^\circ$  - $210^\circ$ . Nilai *error* yang dihasilkan saat pengujian dinding lurus sebesar 1,44 cm. Pemberian nilai  $K_p$  lebih dari 8 dapat menyebabkan robot sulit untuk mempertahankan posisi terhadap referensi. Kontroler diferensial mempengaruhi untuk memperbaiki respon sistem dan disarankan untuk pengembangan menggunakan metode kontrol yang lain (Nashrullah *et al.*, 2013).

*Finite State Machine* adalah sebuah metode sistem kontrol yang menggambarkan tingkah laku dengan menggunakan tiga hal yaitu: *state* (keadaan), *event* (kejadian), *action* (aksi). Pada saat dalam periode waktu yang signifikan sistem berada pada *state* yang aktif. Sistem dapat berpindah menuju *state* lain saat dapat masukan dari perangkat luar atau komponen dalam sistemnya sendiri. Perpindahan ini juga disertai oleh aksi yang dilakukan sistem

ketika ada masukan yang terjadi. Salah satu keuntungan menggunakan metode FSM adalah kemampuannya dalam mendekomposisi aplikasi yang besar dengan hanya menggunakan sejumlah kecil item *state* (Setiawan, 2006).

Pada beberapa penelitian sebelumnya telah digunakan metode *Proportional Derivatif*, *Fuzzy Logic*, dan *Behaviourbased control* untuk mengontrol robot berkaki. Menurut satriyo, penggunaan metode *Finite State Machine* (FSM) untuk mengontrol robot sepakbola beroda didapatkan hasil yang sangat baik dengan tingkah laku yang terstruktur dan terdiri dari beberapa banyak perilaku kecil dibandingkan dengan metode *Behaviour based control* (Satriyo,2014). Oleh karena itu, pada penelitian ini metode *Finite State Machine* digunakan pada robot berkaki pemadam api sehingga dapat membantu membuka wawasan mahasiswa Teknik Elektro Universitas Jember bahwa masih banyak metode yang bisa digunakan pada perancangan sistem kontrol robot.

Dari Proyek Akhir yang ingin saya buat ini, tentang pembuatan alat dengan judul “Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Berkaki Pemadam Api”. Peneliti berharap dengan menggunakan metode *Finite State Machine* pada robot berkaki ini, diharapkan data yang diperoleh dapat dibandingkan dengan metode lain sehingga dapat membantu Tim Robotika Universitas Jember Divisi Kontes Robot Pemadam Api. Secara keseluruhan sistem kontrol ini tersusun dari beberapa komponen yaitu Motor Servo sebagai penggerak kaki, Badan Robot sebagai wadah untuk Arduino Mega 2560 dan Arduino Mega sebagai Mikrokontrolnya.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka dapat diambil beberapa pokok permasalahan yang terdapat dalam proyek akhir sebagai berikut:

1. Bagaimana merancang robot berkaki supaya berjalan dengan lancar ?
2. Bagaimana kinerja robot berkaki dengan metode *Finite State Machine* (FSM)?

## 1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai adalah:

1. Merancang robot berkaki enam supaya berjalan dengan lancar menggunakan motor servo,
2. Menjelaskan kinerja robot berkaki dengan metode *finite state machine*.

## 1.4 Batasan Masalah

Adapun Batasan masalah pada penelitian ini sebagai berikut :

1. Pengujian dilakukan diarena yang digunakan dalam perlombaan Kontes Robot Pemadam Api Indonesia.
2. Menggunakan 2 mikrokontroler Arduino.
3. Robot berkaki berjalan dengan telusur kanan.

## 1.5 Manfaat Penelitian

Dari pembuatan proyek akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

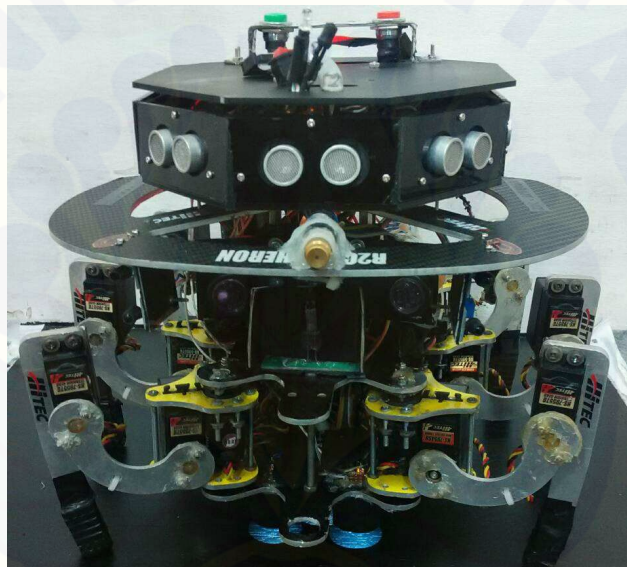
1. Dapat menjadi rujukan untuk menggunakan metode *Finite State Machine* dalam pembuatan robot.
2. Dapat menjadi rujukan oleh peneliti lain untuk melakukan pengembangan pada robot berkaki.
3. Dapat dimanfaatkan pada Kontes Robot Pemadam Api Berkaki Indonesia.

## BAB 2. TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang komponen yang digunakan dan pengertian umum tentang komponen. Juga dijelaskan tentang pendapat yang terkait dengan masalah yang dibahas, dapat dijelaskan dibawah ini antara lain :

### 2.1 Penelitian Terkait

Robot berkaki memiliki bentuk kaki yang mirip seperti laba-laba atau hewan yang lainnya dan bahkan manusia yang dapat melangkah.



Gambar 2. 1 Robot Berkaki (Sumber: Santoso et al., 2017)

Robot *mobile* yang mampu berjalan menggunakan kaki, pada kali ini menggunakan 6 (enam) kaki karena robot lebih dapat stabil daripada menggunakan 4 kaki serta dengan jumlah kaki yang sama pada tiap sisi, maka mempermudah desain rangkaian sehingga tidak memerlukan untuk menghitung berat rangkaian di tiap sisi robot sehingga robot dapat berdiri tegak.

### 2.2 Arduino Mega 2560

Arduino merupakan papan mikrokontroler terpadu yang merupakan pengembangan dari mikrokontroler konvensional seperti AVR. Kelebihan arduino dibandingkan mikrokontroler lain diantaranya adalah memiliki bahasa

pemrograman sendiri yang merupakan modifikasi dari bahasa pemrograman C++. Arduino juga memiliki konverter *USB to Serial* yang tergabung dalam *board* sehingga komunikasi serial menjadi lebih praktis.



Gambar 2.2 Arduino Mega 2560 (sumber: arduino.cc)

Arduino Mega adalah jenis arduino yang menggunakan mikrokontroler AVR ATmega2560 sebagai chip utama. Berikut spesifikasi dari Arduino Mega 2560 :

Tabel 2.1 Spesifikasi Arduino Mega 2560

Chip	ATMega328P
Tegangan Operasi	5 V
Tegangan <i>Input</i> (Rekomendasi)	7-12 V
Tegangan <i>Input</i> (Limit)	6-20 V
Pin I/O Digital	14 (6 merupakan PWM)
Pin <i>Input</i> Analog	6
Arus DC pin I/O	20 mA
Arus DC pin 3.3 V	50 mA
Flash Memory	32 KB, 0,5 KB untuk <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
<i>Clock Speed</i>	16 MHz
Pin LED	13
Panjang	68,6 mm
Lebar	53.4 mm
Berat	25 g

(sumber : arduino.cc)

### 2.3 Motor Servo

Motor Servo adalah sebuah motor menggunakan sistem *closed feedback* yang bekerja memberitahu posisi motor ke rangkaian kontrol yang ada didalam motor servo. Motor ini terdiri dari motor dc, potensiometer yang berfungsi sebagai penentuan batas sudut dari putaran servo, sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo, rangkaian kontrol dan rangkaian *gear*. (Hidayat *et al.*, 2011)



Gambar 2.3 Motor Servo (sumber: Hidayat *et al.*, 2011)

Pengendalian motor servo dilakukan menggunakan PWM (*Pulse Width Modulation*). Motor Servo berputar searah jarum jam jika jumlah pulsa lebih kecil dari 1600 pulsa (1,6 ms pulsa). Sebaliknya jika berputar berlawanan jarum jam ketika pulsa lebih besar dari 1600 pulsa (1,6 ms pulsa).

Motor Servo memiliki 2 jenis yaitu motor servo *Continous* dan motor servo standar. Perbedaanya jika servo standar hanya mampu berputar 180 derajat biasanya digunakan untuk robot lengan dan jika servo *Continous* mampu berputar 360 derajat biasa digunakan untuk *Mobile Robot*.

### 2.4 Sensor Ultrasonik PING

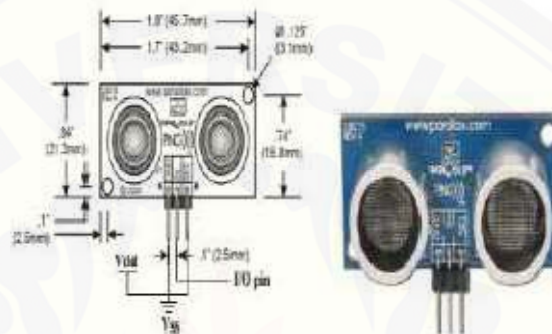
Gelombang Ultrasonik adalah Gelombang Akustik yang mempunyai frekuensi 20 kHz sampai 20 MHz. Frekuensi yang didapat tergantung pada medium yang dilewati bisa dari kerapatan rendah pada fasa gas, cair hingga padat.



Jika gelombang ultasonik melewati sebuah medium, besarnya jarak dapat dihitung sebagai berikut :

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

Dimana S adalah jarak dalam satuan meter, v adalah kecepatan suara, dan t adalah waktu tempuh dalam satuan detik. Saat gelombang ultrasonik melewati suatu penghalang maka sebagian gelombang dipantulkan dan diserap sedangkan bagian yang lain diteruskan. (sumber: Arief, 2011)

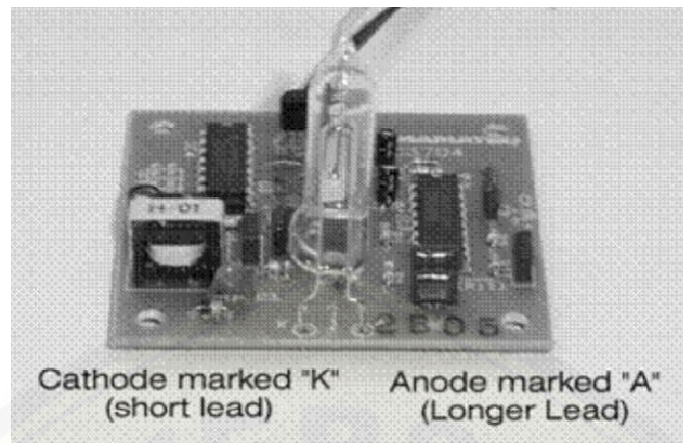


Gambar 2.4 Sensor Ultrasonik (sumber: Arief, 2011 )

Sensor ultrasonik adalah sebuah sensor yang merubah besaran fisis (bunyi) menjadi besaran listrik. Sensor ultrasonik ini terdiri dari *transmitter*, *reciever*, dan *komparator*. Selain itu, gelombang ultrasonik dibangkitkan oleh sebuah kristal tipis bersifat *piezoelektrik* dimana akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. (sumber: Arief, U., et al. 2011 )

## 2.5 Sensor Api

Menggunakan UV Tron *Flame Detector* untuk mendeteksi ultraviolet dengan nyala sebuah lilin yang dapat dideteksi dan masih bagus hingga berjarak 5 meter. Cahaya terlihat tidak mempengaruhi UV Tron. Sensor ini berguna sebagai alat mendeteksi sumber api yang panjang spectral 185 nm hingga 160 nm. Sensor UVTron digunakan sebagai masukan pada mikrokontroler (Sumber: Suryatini et al., 2013)



Gambar 2.5 Sensor Api UV Tron (Sumber: Suryatini et al., 2013)

Cara kerja Tabung UVTron ketika katoda diberi sinyal ultraviolet, foto elektron akan dipancarkan dari katoda oleh efek foto elektrik dan dipercepat menuju anoda melalui medan listrik. Saat tegangan dimasukkan, medan magnet meningkat, medan listrik semakin besar dan energi kinetik elektron menjadi besar sehingga molekul gas ditabung bertubrukan. Ion positif yang dipercepat menuju katoda bertubrukan menyebabkan elektron sekunder lain hal ini membuat arus cukup besar diantara elektroda.

Tabel 2.2 Spesifikasi Flame Sensor UVTRON

Spectral Response	185 to 260 nm
Window Material	UV glass
Weight	Approx 1.5 g
Supply Voltage	400 V
Average Discharge Current	1 Ma
Operating Temperature	-20 to +60 °C
Discharge Starting Voltage (with UV radiation) (DC)	280 V
Recommended Operating Voltage (DC)	325±25 V
Background	10 min <sup>-1</sup>
Sensitivity	5000 min <sup>-1</sup>

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

## 2.6 CMPS03

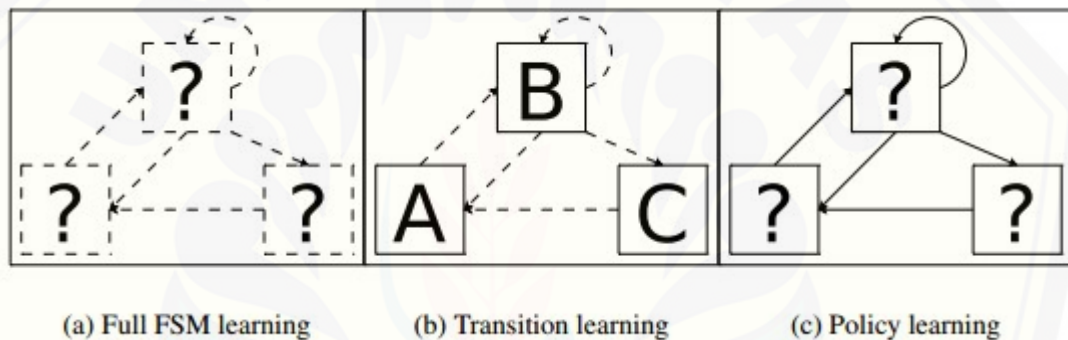
CMPS03 adalah kompas digital yang bertujuan sebagai navigasi robot yang menggunakan sensor medan magnet, dengan arah mata angin dibagi dalam bentuk derajat (0) utara, (90) timur, (180) selatan, dan (270) barat. CMPS03 membutuhkan catu daya 5v pada pada arus nominal 25mA dan memiliki *interface*I2C yang digunakan untuk membaca arah dalam bentuk data serial. Dengan mode 8 bit arah utara ditunjukkan data 255 dengan resolusi 1,40625 derajat/bit sedangkan mode 16 bit utara ditunjukkan data 65535 dengan resolusi 0,0055 derajat/bit.



Gambar 2.6 Sensor Kompas CMPS03 (sumber: <http://tokorobotik.com/CMPS03,2018>)

## 2.7 Finite State Machine

*Finite State Machine* adalah sebuah metode sistem kontrol yang menggambarkan tingkah laku dengan menggunakan tiga hal yaitu: *State* (Keadaan), *Event* (Kejadian), *Action* (aksi). Pada saat dalam periode waktu yang signifikan sistem berada pada *state* yang aktif. Sistem dapat berpindah menuju *state* lain saat dapat masukan dari perangkat luar atau komponen dalam sistemnya sendiri. Perpindahan ini juga disertai oleh aksi yang dilakukan sistem ketika ada masukan yang terjadi. Salah satu keuntungan menggunakan metode FSM (*Finite State Machine*) adalah kemampuannya dalam mendekomposisi aplikasi yang besar dengan hanya menggunakan sejumlah kecil item *state* (Setiawan, I., 2006).



Gambar 2.7 Tiga learning FSM (Sumber: Grollman & Jenkins, 2010)

Dari gambar 2.8 tersebut terdapat beberapa aspek FSM, yaitu :

### a. Full FSM Learning

Mempelajari Beberapa *state* yang diberikan kepada sistem kontrol. Beberapa *state* diletakkan secara berurutan, sehingga *state* awal bisa menuju *state* akhir. Mirip seperti orang tidur hingga berdiri, pada tahap ini pertama kali harus duduk, lalu tangan menopang untuk berdiri, dan yang terakhir mengangkat badan supaya berdiri. Proses tersebut dilakukan secara berurutan, jika tidak berurutan maka tidak akan bisa untuk berdiri.

### b. Transition Learning

Mempelajari kebijakan *state* yang berdiri secara individu. Masing-masing *state* mempunyai program untuk melakukan suatu perilaku tertentu untuk memudahkan dalam memutuskan *state* mana yang akan dijadikan *loop* tertutup.

*c. Policy learning*

Mempelajari hubungan antar *state*. Keputusan *state* pindah ke *states* selanjutnya, menggunakan data *feedback* pada sensor. Pada saat robot sudah menjalankan *state* dan mendapatkan *output* yang diinginkan, maka akan menuju *states* selanjutnya hingga akhir *state*.



### BAB 3. METODOLOGI PELAKSANAAN DATA

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

#### 3.1 Waktu dan Tempat Kegiatan

Pembuatan, penelitian, pengujian, dan analisi robot berkaki dengan metode FSM ini secara umum dilakukan di:

Tempat : Laboratorium Elektronika dan Terapan, Fakultas Teknik, Universitas Jember yang beralamat di Jalan. Slamet Riyadi no. 62 Patrang, Jember.

Waktu : Desember 2017 sampai dengan Juli 2018.

Tabel 3.1 Rencana Kegiatan Tugas Akhir

No	Kegiatan	Bulan ke-		
		I	II	III
1	Studi literatur dan Pembuatan laporan Bab 1 sampai Bab 3	■		
2	Pembuatan rangkaian penyusun sistem dan Konsultasi	■	■	
3	Proses Kalibrasi dan Konsultasi		■	
4	Pengujian alat dan Konsultasi		■	■
5	Menganalisa data hasil pengujian dan Konsultasi			■
6	Pembuatan Laporan			■

Keterangan:

■ : Kegiatan dilaksanakan

#### 3.2 Ruang Lingkup Kegiatan

Ruang lingkup kegiatan ini berisi tentang batasan masalah dalam Tugas Akhir:

- Menggunakan Sensor Ultrasonik PING untuk navigasi robot.

- b. Akusisi data berbasis Arduino Mega 2560.
- c. Tidak didesain untuk mapping lintasan.
- d. Menggunakan Sensor UV Tron untuk mendeteksi keberadaan api.
- e. Penggunaan Metode *Finite State Machine*.

### 3.3 Jenis dan Sumber Data

Jenis dan sumber data menjelaskan tentang keseluruhan alat yang akan dibuat, yaitu:

1. Servo sebagai aktuator dari robot berkaki untuk menggerakkan kaki robot.
2. Sensor Ultrasonik PING, sebagai navigasi robot.
3. Sensor *Flame UVTron*, sebagai pendeteksi api pada ruangan yang akan dituju robot.
4. Exhinguiser, sebagai penyemprot air untuk memadamkan api
5. Arduino Mega 2560 sebagai Mikrokontrol pergerakan robot.
6. *Shield Arduino 2560* sebagai mempermudah pemasangan arduino saat pengerjaan pembuatan elektrik.

### 3.4 Metode Pengumpulan Data

Dalam pembuatan alat ini menggunakan Metode *Finite State Machine*.

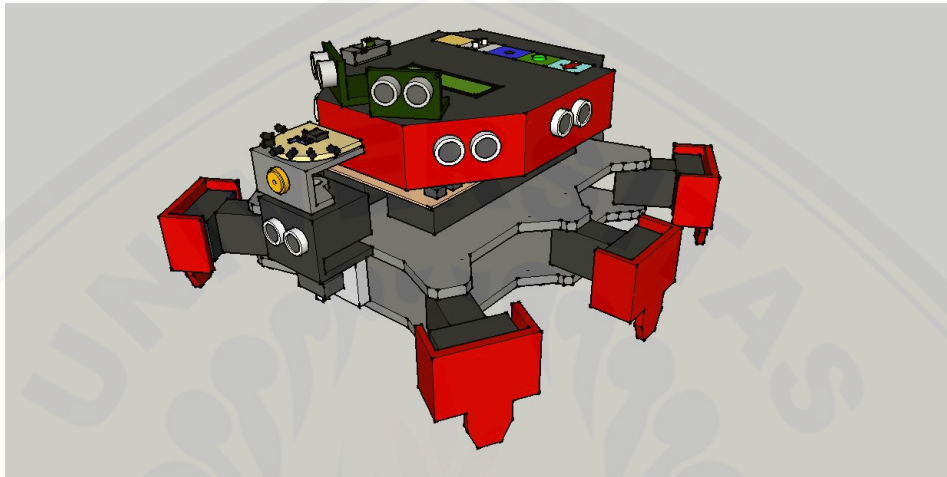
Adapun langkah-langkah penelitian sebagai berikut:

- a. Pengumpulan Studi Literatur
- b. Melakukan perancangan perangkat keras desain mekanik robot dan perangkat lunak.
- c. Melakukan pembuatan rangkaian penyusun sistem.
- d. Melakukan pengujian perangkat keras dan perangkat lunak. Pertama pengujian ini dilakukan secara terpisah dan selanjutnya akan dilakukan pengujian secara bersama.
- e. Menganalisa data yang telah diperoleh saat pengujian.

### 3.5 Perancangan Alat

#### 3.5.1 Perancang Desain Alat

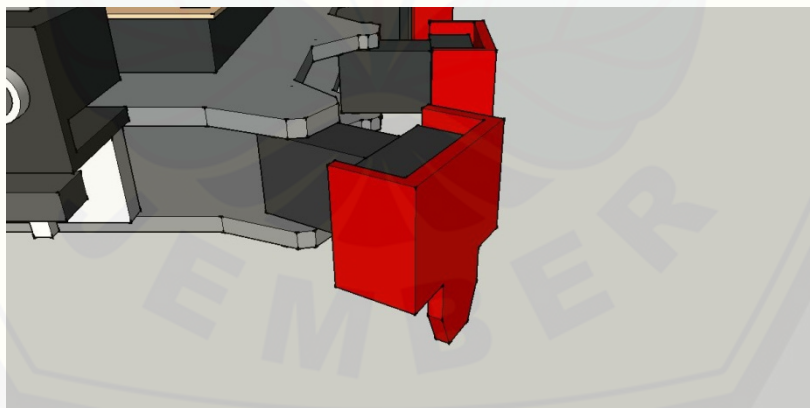
Pada pembuatan alat ini memakai Sensor ultrasonik yang digunakan untuk menghindari halangan yang ada dan sebagai navigasi robot serta menggunakan motor servo sebagai penggerak kaki robot untuk berjalan.



Gambar 3.1 Mekanik robot berkaki 6

Dari Gambar mekanik dapat diketahui desain kaki dan bodi robot yaitu:

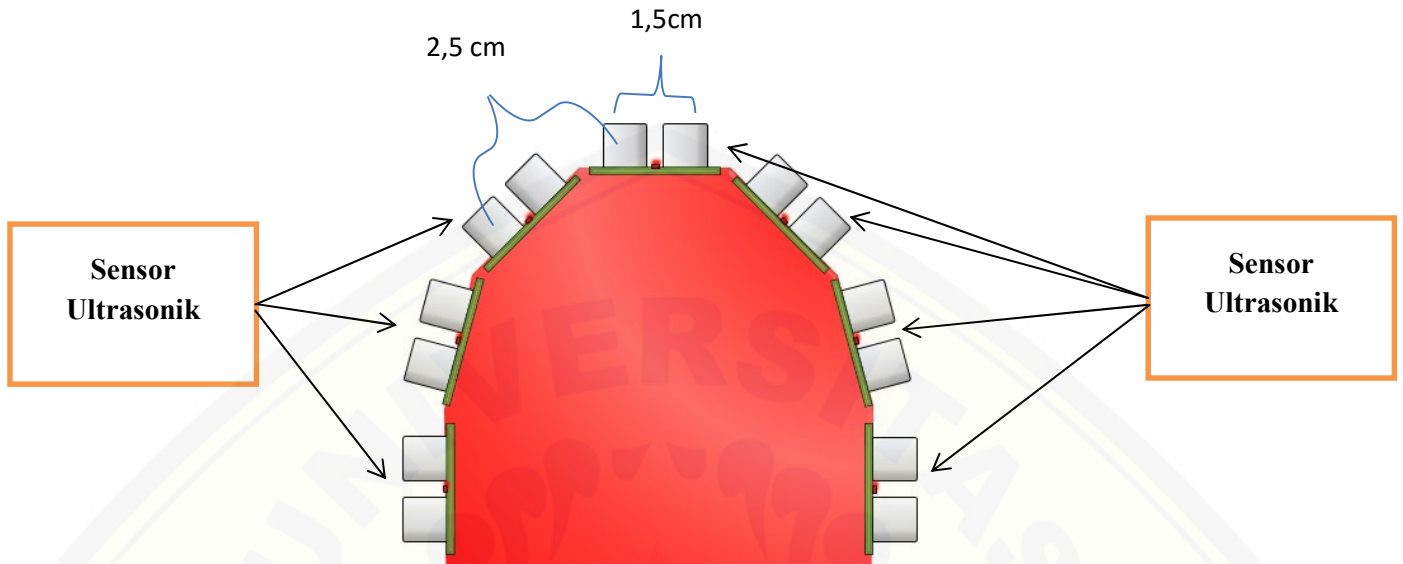
- a. Desain Kaki Robot berkaki.



Gambar 3.2 Desain Kaki



b. Desain Peletakan Sensor Ultrasonik.

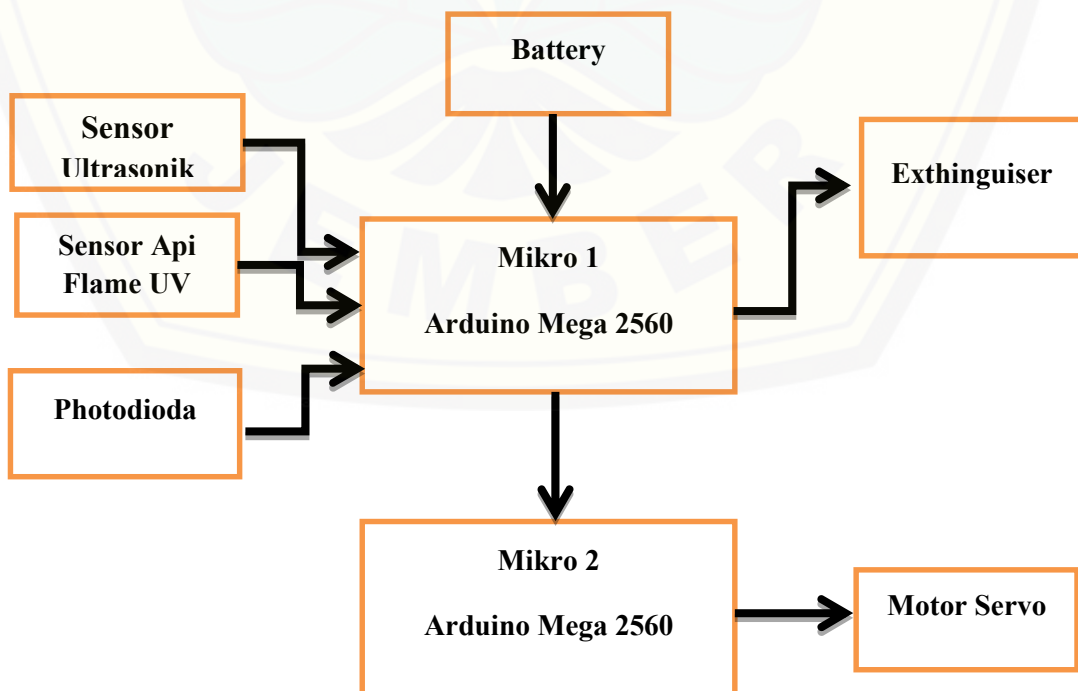


Gambar 3.3 Desain Peletakan Sensor Ultrasonik

3.5.2 Perancangan Perangkat Keras

Dalam pembuatan robot berkaki dengan metode FSM ini diharapkan dapat berjalan dengan yang diharapkan, oleh sebab itu dibuatlah blok diagram.

a. Blok Diagram

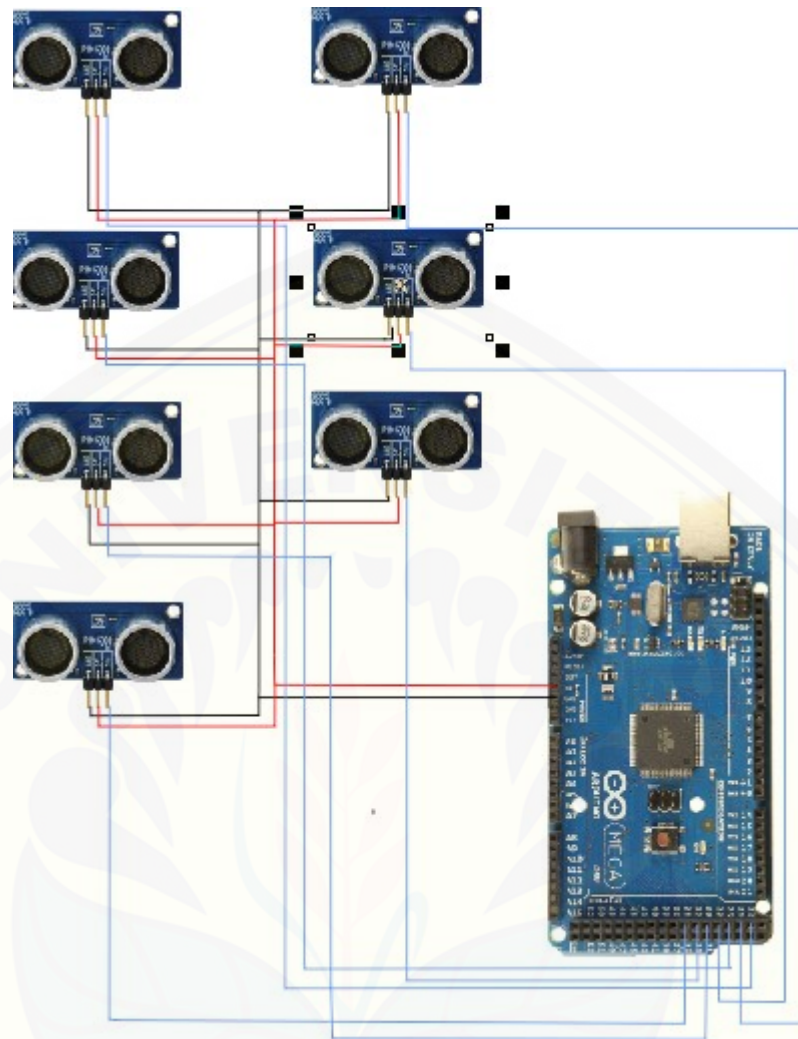


Gambar 3.4 Blok diagram untuk navigasi robot berkaki

Pada diagram blok diatas *supply* didapat dari baterai yang digunakan untuk catu daya, Sensor Ultrasonik menggerakkan pergerakan motor servo.

b. Rangkain Sensor Ultrasonik

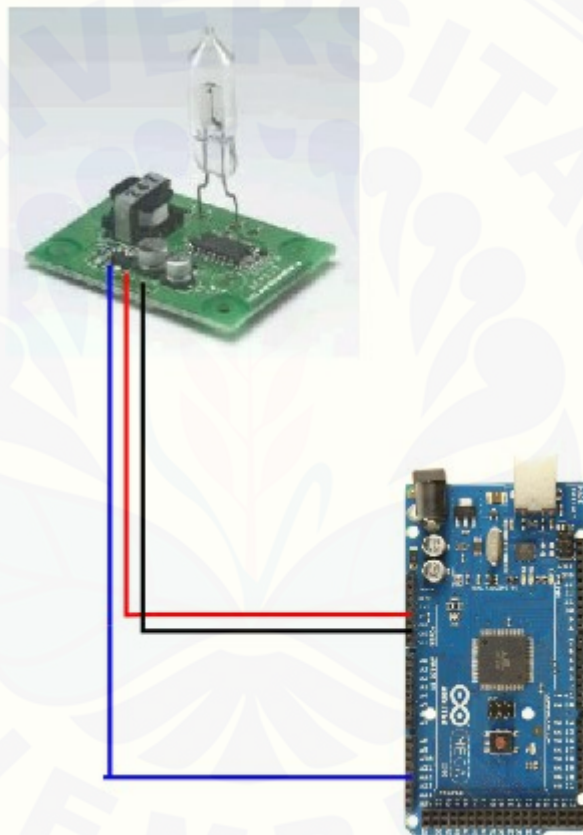
Ada 3 pin yaitu GND, VCC, *INPUT*, disini digunakan 6 buah sensor yang masing-masing input terhubung dengan arduino mega 2560. Sensor 1 terhubung dengan pin 22, Sensor 2 Terhubung dengan pin 24, Sensor 3 Terhubung dengan pin 26, Sensor 4 Terhubung dengan pin 28, Sensor 5 Terhubung dengan pin 30, Sensor 6 Terhubung dengan pin 32, Sensor7 Terhubung dengan pin 32, Untuk GND dihubungkan dengan pin GND sedangkan Untuk VCC dihubungkan dengan pin 5V.



Gambar 3.5 Rangkaian Ultrasonik

c. Rangkaian Sensor UV Tron

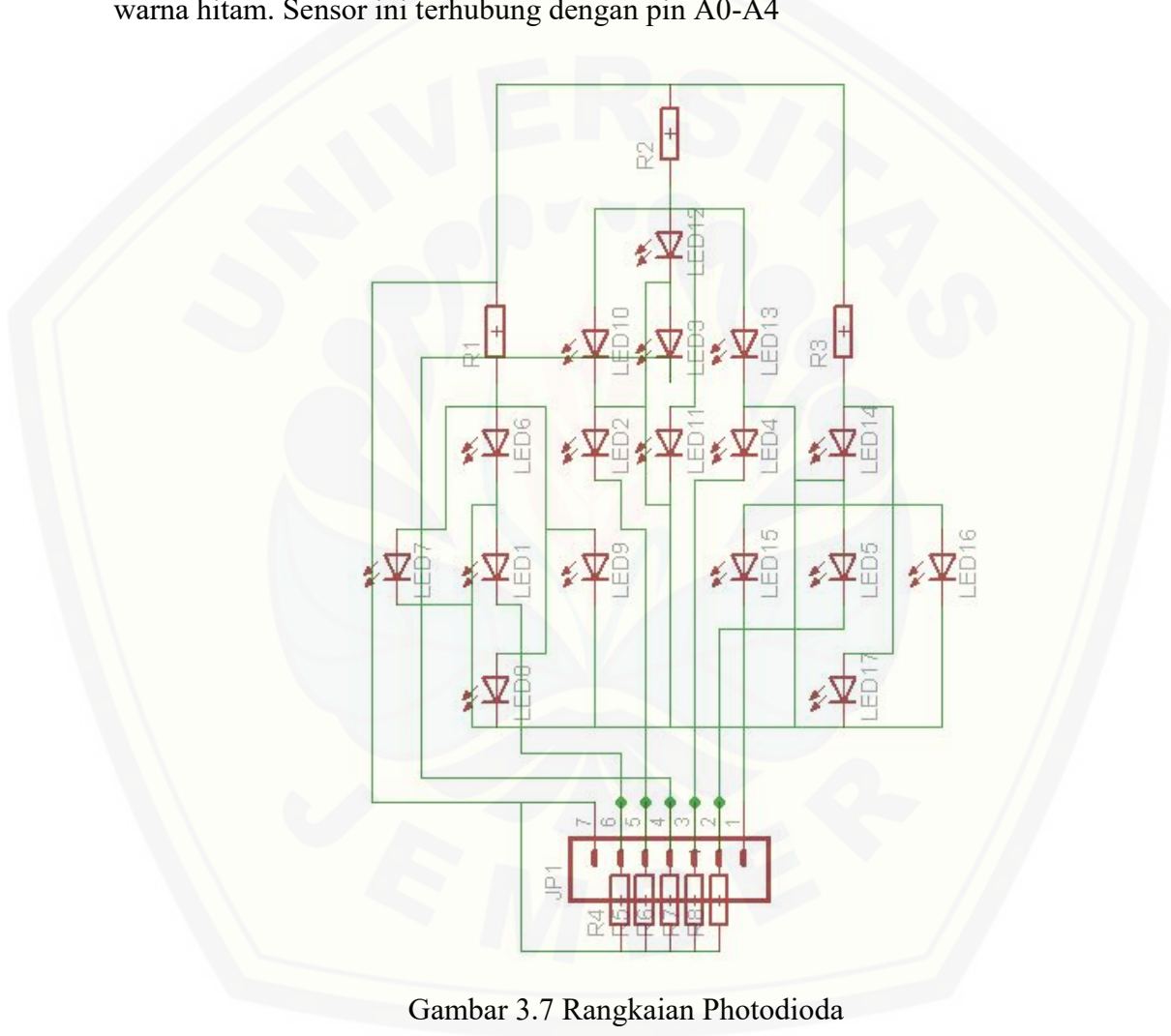
Ada 4 Pin yang digunakan hanya GND, VCC, *INPUT*, untuk menggunakan sensor ini dibutuhkan Driver UV Tron C10423, jadi sensor ini dapat mendeteksi api hingga jangkauan 5 meter jika sensor menemukan api maka Input Berlogika *LOW* jika tidak menemukan api maka berlogika *HIGH*, Sensor ini terhubung dengan pin 12.



Gambar 3.6 Rangkaian UV TRON

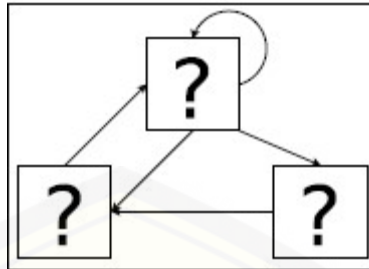
d. Rangkaian Photodioda

Pada rangkaian ini digunakan photodioda untuk mendekteksi garis putih menggunakan 6 buah photo dioda dan 12 buah LED supaya mendapatkan kuantitas cahaya, cahaya yang didapatkan semakin besar maka resistansi semakin kecil. Ini berarti pewarnaan garis putih memantulkan cahaya lebih besar dari pada warna hitam. Sensor ini terhubung dengan pin A0-A4



Gambar 3.7 Rangkaian Photodioda

### 3.5.3 Metode *Finite State Machine*



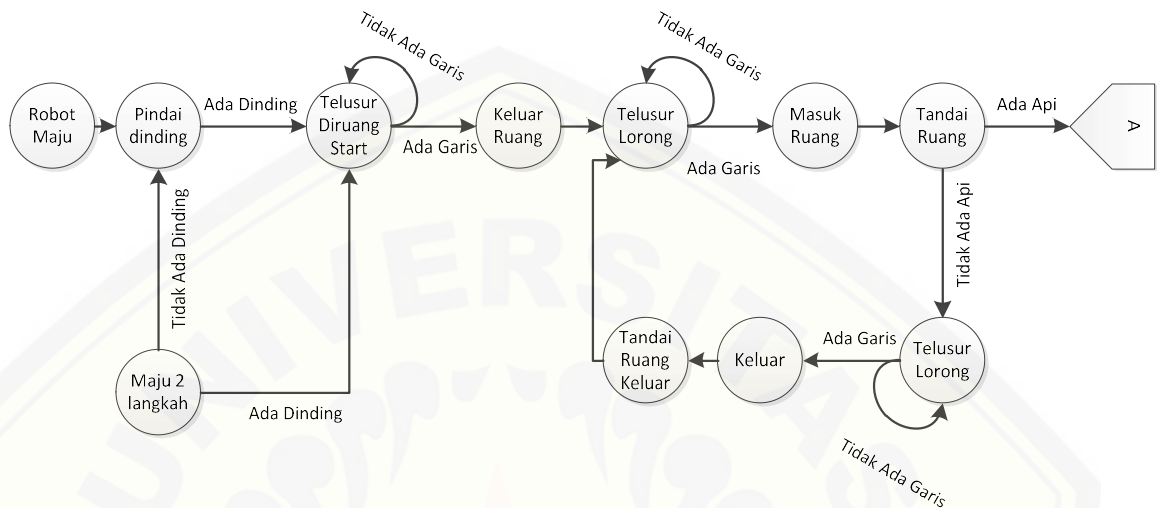
Gambar 3. 8 *Policy Learning*

Penggunaan Metode *Finite State Machine* menggunakan aspek policy learning. Pada tahap ini mempelajari hubungan antar masing-masing *state*. Keputusan suatu *state* pindah ke *state* selanjutnya, megunakan data *feedback* pada sensor. Pada saat robot telah menjalankan *state* dan mendapat keluaran yang diharapkan, maka akan menuju *state* selanjutnya hingga menuju akhir *state*.

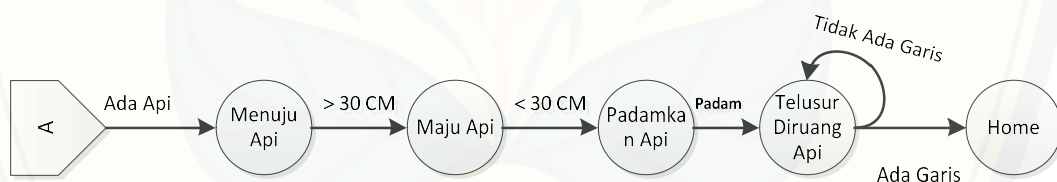
Pada robot berkaki peamadam api dengan menggunakan metode FSM terdapat beberapa perilaku antara lain; memindai dinding, telusur diruang start, telusur lorong, menuju api, padamkan api, dan *Home*. Dari beberapa perilaku tersebut akan menjadi beberapa *state* yang akan membentuk tingkah laku robot.

### 3.5.4 Perancangan Sistem Kontrol

Pada bagian ini akan dijelaskan desain sistem kontrol robot untuk mengetahui perilaku robot.



Gambar 3.9 Diagram FSM 1



Gambar 3.10 Diagram FSM 2

- Robot Maju adalah tindakan mikrokontrol memerintahkan servo untuk menggerakkan robot jalan untuk memindai dinding dan maju keluar garis.
- Pindai Dinding adalah dimana sensor ultrasonik Robot memindai dinding jika dinding di deteksi maka robot maju ke *state* telusur diruang start, jika dinding tidak terdeteksi maka robot maju 2 langkah hingga mendeteksi dinding.
- Telusur Diruang Start adalah tindakan robot untuk menelusuri ruang start hingga menemukan garis.

- d. Maju Dua Langkah adalah tindakan dimana robot tidak menemukan dinding maka robot maju hingga menemukan dinding.
- e. Keluar Ruang adalah tindakan dimana robot mampu menemukan garis sehingga keluar dari *start*.
- f. Telusur Lorong tindakan dimana robot menelusuri lorong hingga menemukan garis.
- g. Masuk Ruang adalah tindakan dimana robot telah mendeteksi adanya garis, untuk memindai tidak atau adanya api pada ruangan menggunakan sensor UV TRON.
- h. Tandai Ruang adalah tindakan dimana robot dapat mengetahui ruang yang dituju.
- i. Keluar adalah tindakan dimana robot telah menandai ruang dan tidak menemukan api dan robot telah menemukan garis untuk menandai ruang bahwa tidak menemukan api.
- j. Menemukan Api adalah tindakan dimana sensor UV TRON mendeteksi adanya Api dalam suatu ruangan.
- k. Menuju Api adalah tindakan robot pada saat menemukan titik api maka robot akan mendekati titik api.
- l. Maju Api adalah tindakan mendekati api dengan jarak kurang dari 30 cm untuk *scan* api dan memadamkan api.
- m. Padamkan Api adalah tindakan dimana sensor *flame* memastikan posisi api yang akan dipadamkan, dan menghidupkan *exthinguiser* untuk menyembrotkan air yang bertujuan untuk memadamkan api hingga api padam.
- n. Telusur diruang Api adalah tindakan dimana robot menelusur diruang api hingga menemukan garis untuk keluar dari ruangan.
- o. *Home* adalah tindakan kembali keposisi start awal robot berjalan.



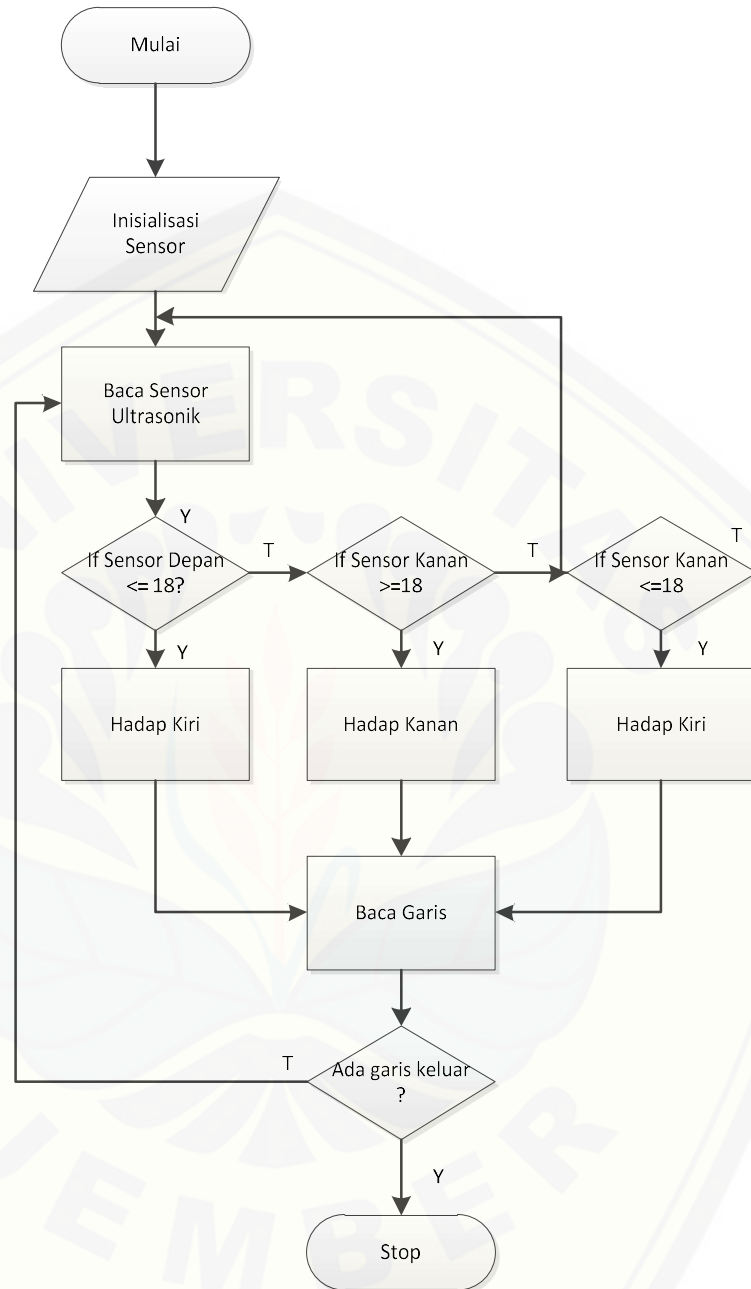
### 3.5.5 Flowchart Sistem

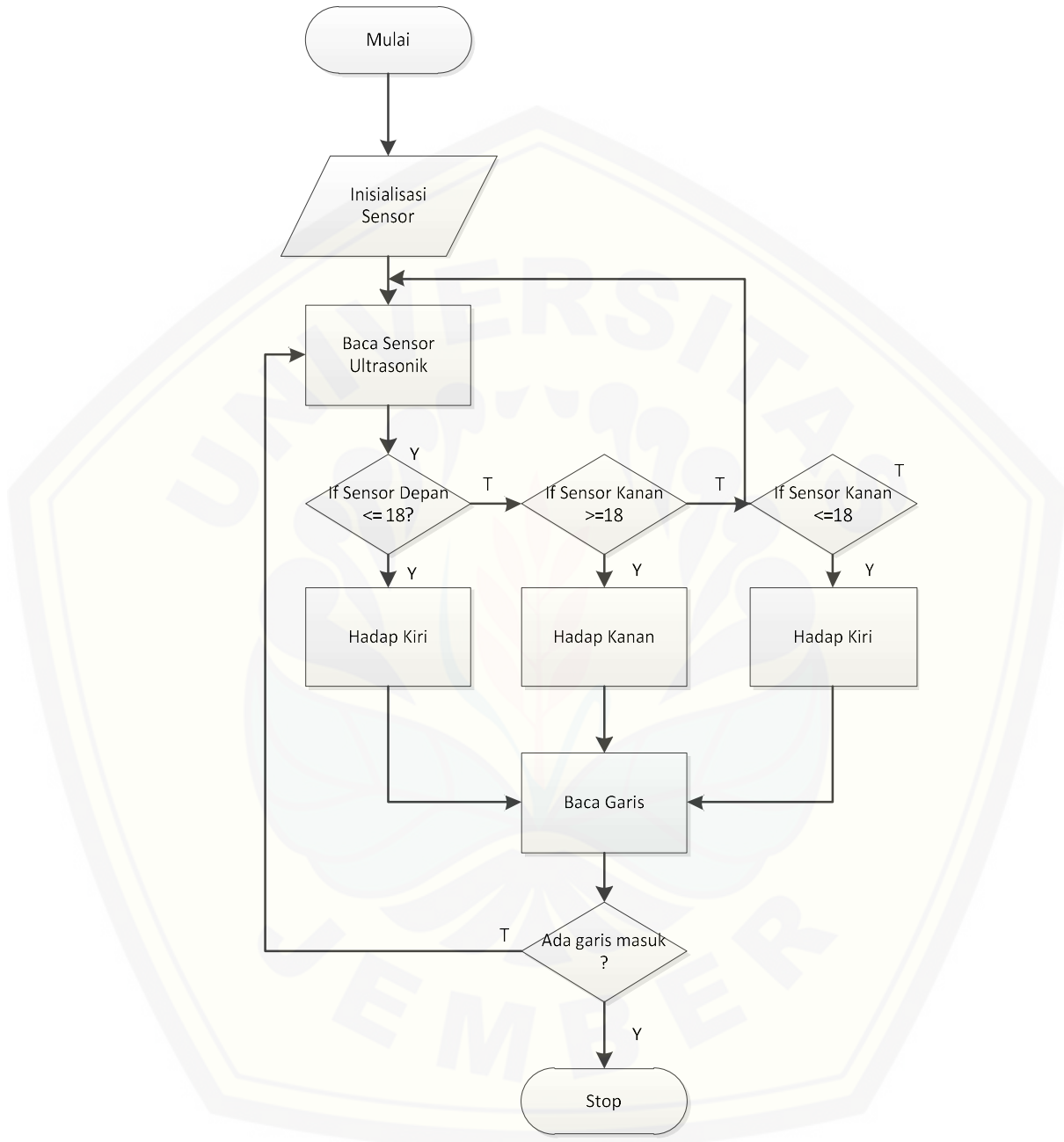
Berikut adalah *flowchart* pada pembuatan Robot berkaki dengan metode FSM:

#### a. *Flowchart* sistem Keseluruhan

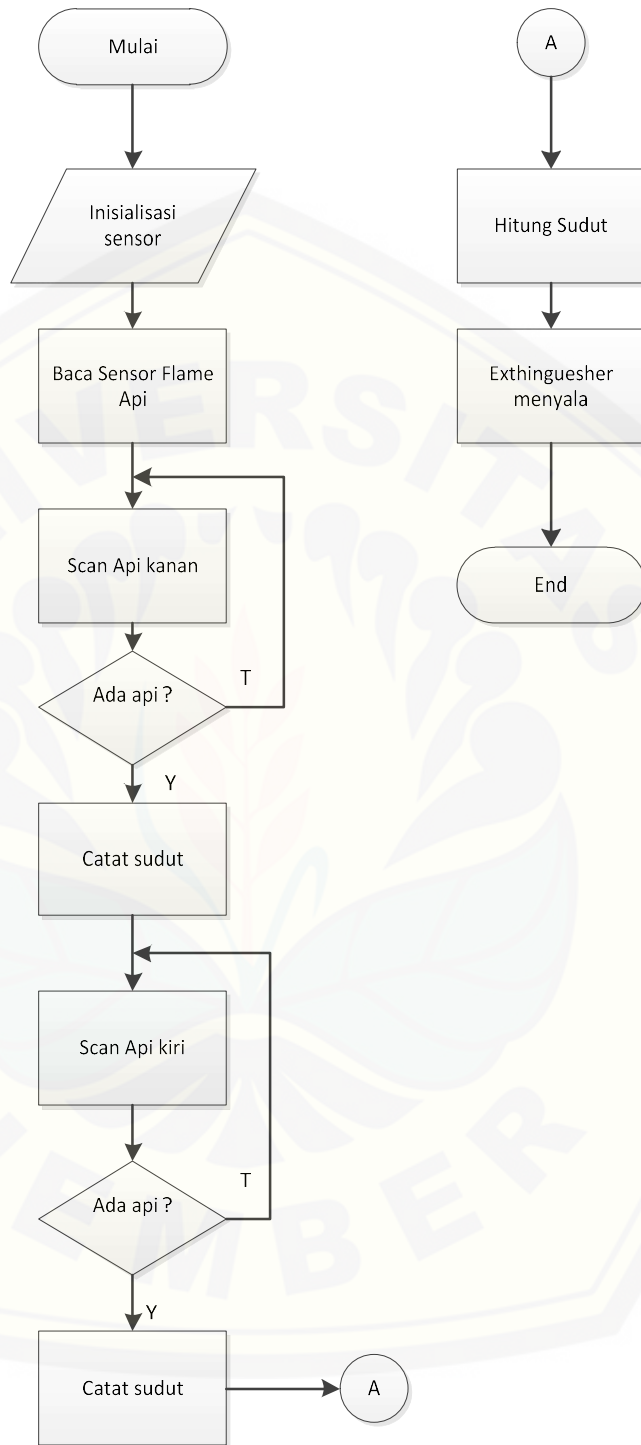


Gambar 3.11 *Flowchart* Sistem Keseluruhan

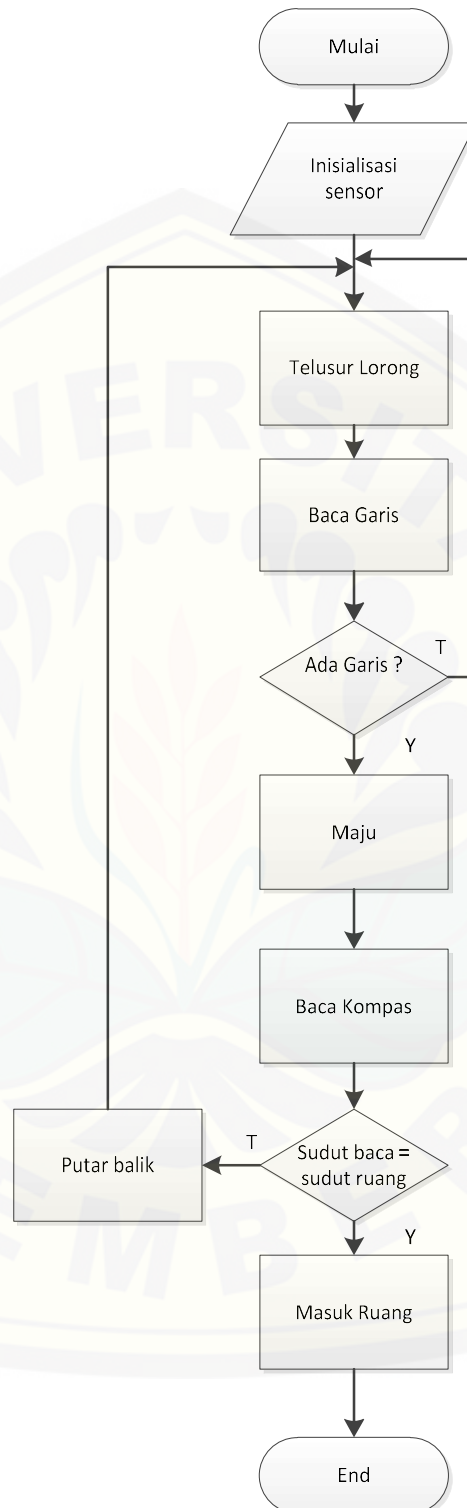
b. *Flowchart* sistem pindai dindingGambar 3.12 *Flowchart* sistem pindai dinding

c. *Flowchart* Sistem Telusur LorongGambar 3.13 *Flowchart* sistem telusur lorong

d. *Flowchart* padamkan api



Gambar 4.14 *Flowchart* sistem padamkan api

e. *Flowchart sistem Home*Gambar 3.15 *Flowchart sistem Home*

### 3.6 Pengujian

Pengujian Robot Berkaki Pemadam Api dibagi menjadi beberapa bagian antara lain mencakupi pengujian Sensor, pengujian metode FSM, dan pengujian keseluruhan. Berikut adalah pengujian tersebut.

#### 3.6.1 Pengujian Sensor

Pengujian ini dibagi beberapa bagian yaitu Sensor Ultrasonik, Sensor Photodiode, Sensor UV Tron,

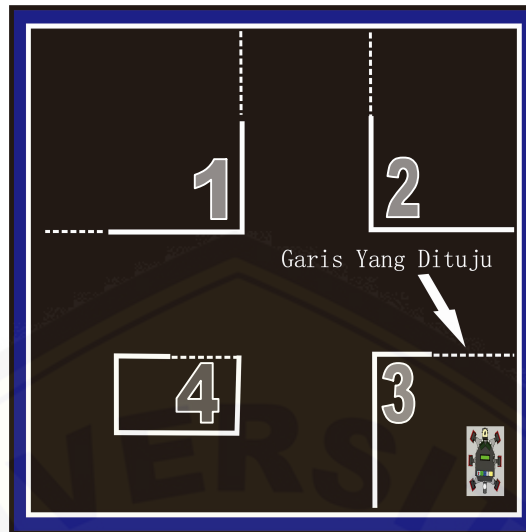
- a. Pada pengujian Sensor Ultrasonik membandingkan hasil pembacaan sensor dengan jarak sebenarnya menggunakan penggaris. Pada pengujiannya didepan sensor diberi penghalang sehingga mampu membaca jarak dengan penghalang, lalu diukur apakah sesuai dengan pembacaan menggunakan penggaris.
- b. Pengujian Sensor Photodiode membandingkan hasil pembacaan sensor photodiode dengan 3 warna berbeda hitam, abu-abu, putih sesuai warna dasar lantai arena. Pada pengujiannya dilakukan dengan memberikan lux yang berbeda.
- c. Pengujian Sensor UV Tron dengan memberikan api untuk dideteksi. Pada pengujiannya api diletakkan dengan jarak yang berbeda-beda untuk mengetahui sejauh mana sensor UV Tron mampu membaca api.
- d. Pengujian Sensor Kompas dilakukan dengan mencari range derajat arah kompas dengan mengarahkan kompas ke arah Barat, Utara, Timur, Selatan.

#### 3.6.2 Pengujian *State*

Pada Pengujian dibahas tentang setiap *state* dari metode, Pengujian berawal dari start hingga robot memadamkan api. Tahap ini Fokus disetiap State, Berikut macam-macam state yang akan diuji.

- a. Pengujian Robot Maju, Pindai dinding, Telusur diruang start, dan Keluar Ruang.

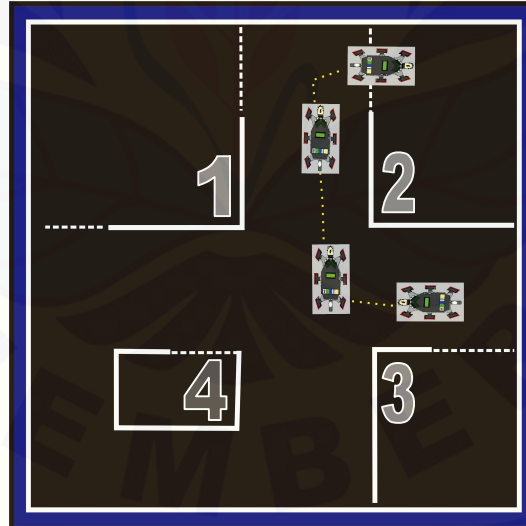
Pengujian Robot Maju akan dihitung seberapa lama robot menemukan garis diruang start hingga mampu keluar dari ruang start.



Gambar 3.16 Rencana Pengujian Robot maju hingga keluar ruang

b. Pengujian telusur Lorong

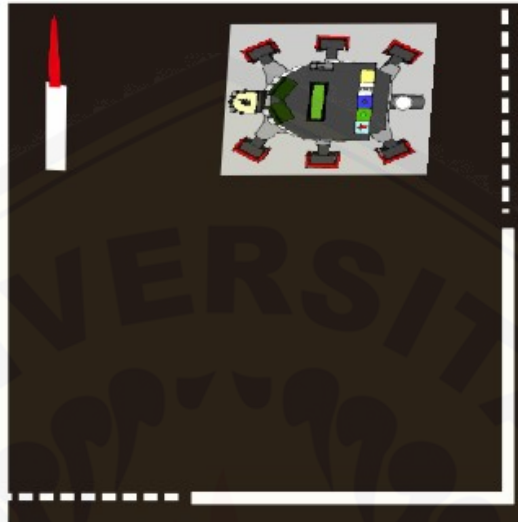
Pengujian ini menguji seberapa lama robot menelusuri lorong hingga masuk keruang untuk mencari api



Gambar 3.17 Rencana Pengujian Telusur lorong

c. Pengujian Menuju api dan Maju Api

Pengujian Menuju api ini menguji seberapa lama robot mampu mendeteksi adanya api dan maju menuju api jika lebih dari 30 cm.



Gambar 3.18 Rencana Pengujian Robot Maju api

d. Pengujian padamkan api

Pengujian padamkan api ini untuk mengetahui *exthinguisher* dapat memadamkan api.

e. Pengujian *Home*

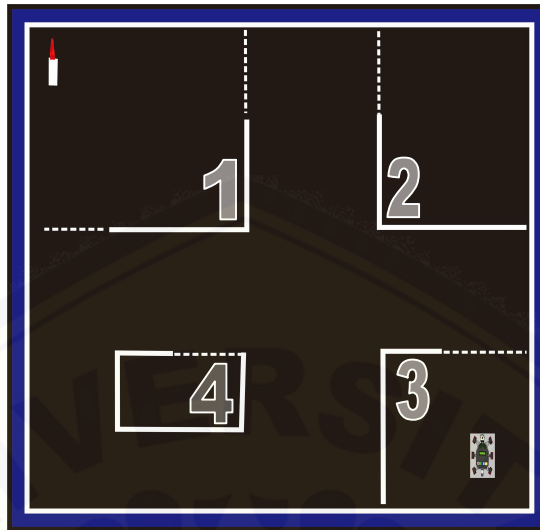
Pengujian *Home* ini menguji seberapa lama robot mampu kembali ke *Home* atau start awal setelah memadamkan api.

### 3.6.3 Pengujian Keseluruhan

Pengujian keseluruhan meliputi pengujian metode FSM. Menguji cara kerja robot ketika berada di arena dan waktu robot dari mulai dinyalakan hingga akhir.

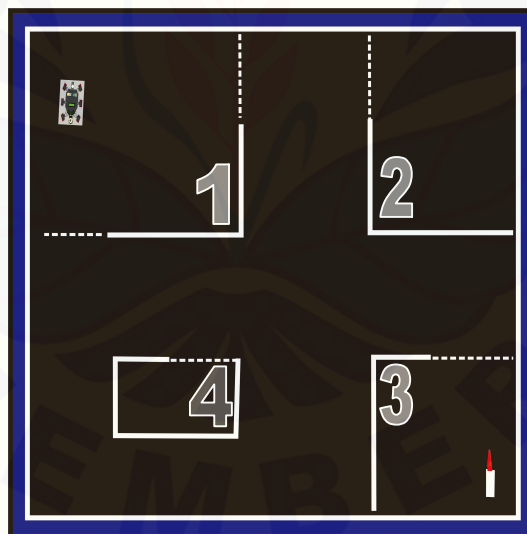


a. Peletakan Start dan posisi api 1



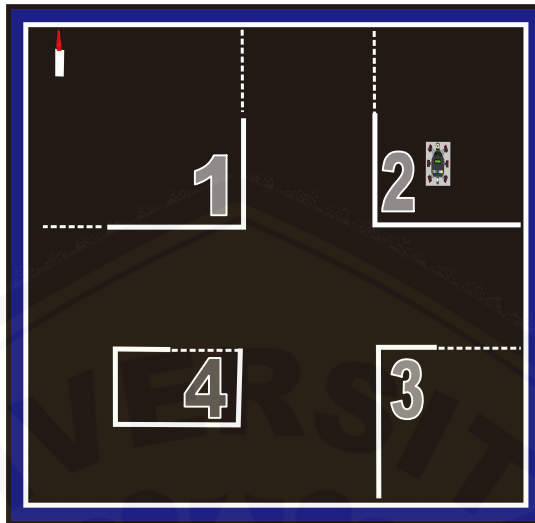
Gambar 3.19 Skema Pengujian 1

b. Peletakan Start dan posisi api 2



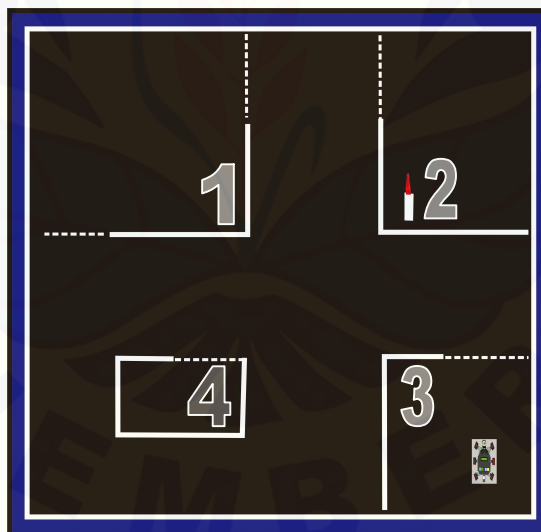
Gambar 3.20 Skema Pengujian 2

c. Peletakan Start dan posisi api 3



Gambar 3.21 Skema Pengujian 3

d. Peletakan Start dan posisi api 4



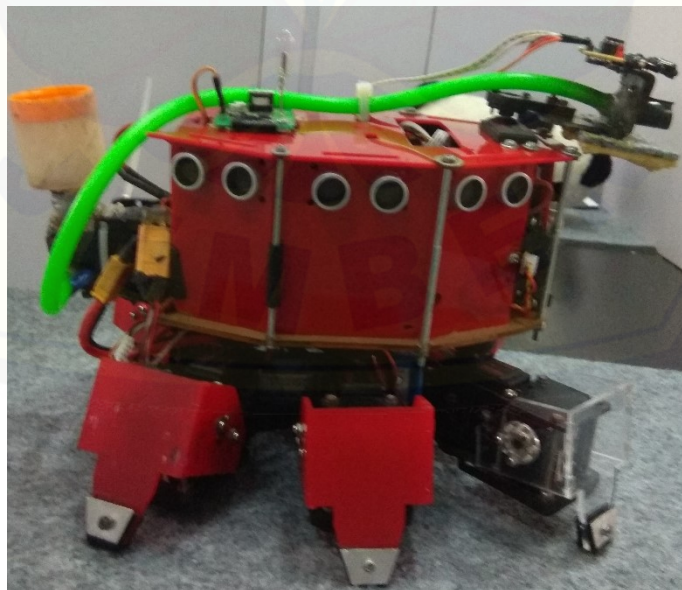
Gambar 3.22 Skema pengujian 4

### 3.7 Hasil Perancangan Alat

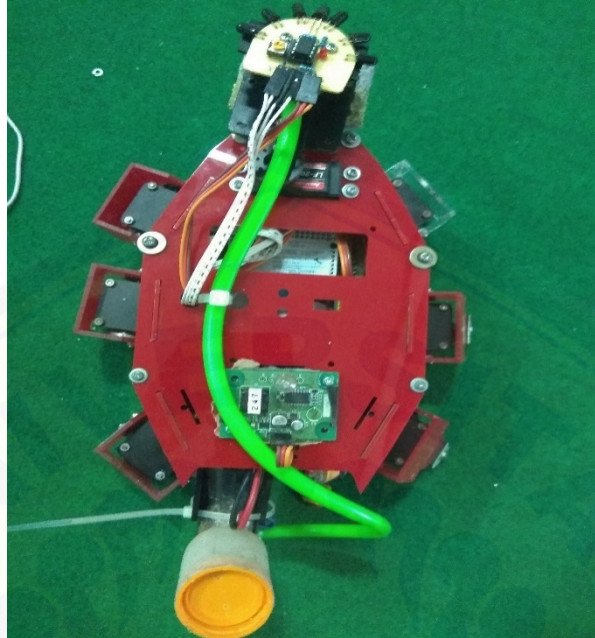
Perancangan alat hampir sama seperti desain pada gambar 9 dan 10, Berikut foto robot yang telah dibuat:



Gambar 3.23 Robot tampak depan



Gambar 3.24 Robot tampak samping



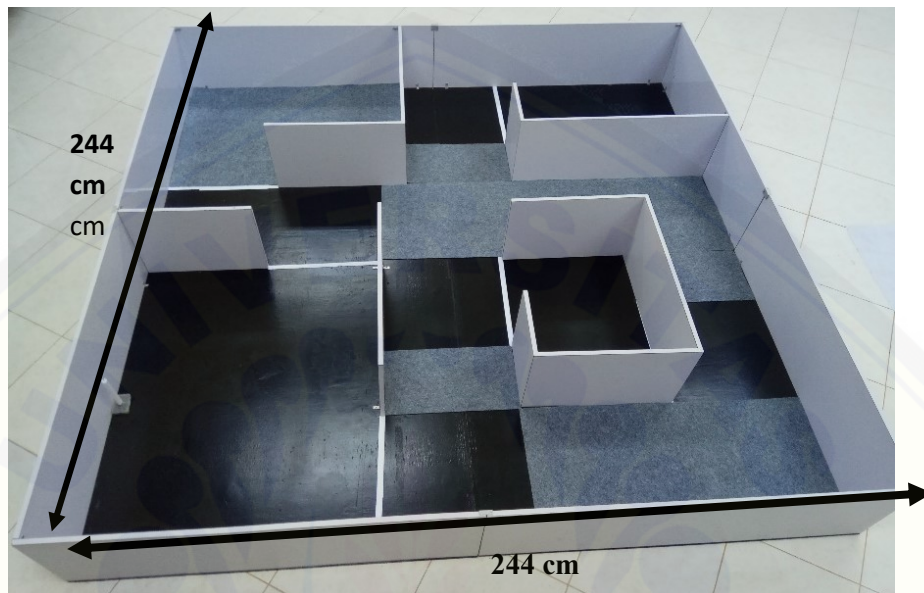
Gambar 3.25 Robot tampak atas

Dari gambar diatas terlihat UV TRON yang dipasang diatas badan dari robot supaya dapat menjangkau api peletakan ditaruh paling atas supaya tidak adanya gangguan pada saat pembacaan sensor, penempatan *exthinguiser* diletakkan dibelakang supaya jika ada kebocoran air tidak merusak rangkaian.

Kerangka dari robot ini menggunakan akrilik 2 milimeter karena akrilik mudah didapatkan serta bahanya yang ringan dan tidak mudah patah, pada gambar 3.22 dan 3.23 terlihat sensor ping yang digunakan untuk navigasi robot dan menghindari halangan. Tinggi dari robot 21 cm disesuaikan dengan lilin api yang digunakan pada saat perlombaan yaitu 17 cm – 20 cm.

Perbedaan dengan desain robot yaitu tata letak sensor ping pada bagian depan awalnya ditaruh diatas bodi robot, dikarenakan pada beberapa kali percobaan robot tidak dapat membaca sempurna maka ditaruh sejajar dengan sensor ping yang berada disamping. Bodi robot yang tertutup berguna supaya rangkaian tidak terlihat dan terlihat rapi sehingga tidak mengganggu pergerakan robot.

Selain itu juga terdapat hal yang penting yaitu arena sesuai standard lomba Kontes robot pemadam api berkaki, Terdapat 4 ruangan pada arena ini, berikut gambar arena robot pemadam api berkaki:



Gambar 3.26 Lintasan Robot Berkaki Pemadam Api

Panjang arena ini 244 cm dan lebar 244 cm, terdapat 4 ruang sebagai labirin untuk meletakkan posisi api, terdapat garis putih untuk mendakan bahwa robot memasuki ruangan yang dibaca oleh sensor photodiode, bahan dari pembuatan arena ini menggunakan kayu pipih.



Gambar 3.27 Pencahayaan Arena Robot

Pengujian penelitian ini dilakukan di ruangan gedung LPM (Lembaga Pemberdayaan Masyarakat) Universitas Jember, Penggunaan cahaya ini berguna supaya pembacaan sensor lebih sempurna dan disesuaikan dengan standard lomba kontes robot pemadam api berkaki. Untuk diketahui intensitas cahaya yang ada pada arena maka perlu diukur dengan *lux* meter, setelah diukur maka intensitas cahaya pada arena adalah 356.0 lux

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Setelah melakukan perencanaan dan perancangan sistem serta pembuatan dan pengujian dapat diambil kesimpulan sebagai berikut:

1. Dengan menggunakan metode *Finite State Machine* (FSM) robot mampu berjalan dengan lancar dapat dilihat pada tabel 4.7 dengan dua kali pengujian lorong, robot tidak menabrak dinding dan selalu berjalan sesuai skenario karena kontrol yang dilakukan lebih detail dan terstruktur.
2. Kinerja robot berkaki pemadam api dengan menggunakan metode *Finite State Machine* (FSM) sangat cocok digunakan karena pemadam api berkaki terdiri dari banyak perilaku kecil sehingga kontrol dapat dilakukan pada setiap kondisi yang berbeda, terbukti dengan 10 kali pengujian dimana keberhasilan yang didapatkan sebesar 70% dengan rata-rata waktu 192,1 detik.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan tentang “Penggunaan Metode Finite State Machine (FSM) Untuk Mengontrol Robot Berkaki Pemadam Api” saran yang dapat peneliti sampaikan agar penelitian selanjutnya dapat memberikan manfaat yang lebih baik lagi dimasa mendatang:

1. Penggantian Servo luar dan dalam pada kaki robot dengan menggunakan servo yang lebih baru.
2. Perlu memodifikasi rangkaian sehingga kabel-kabel terlihat lebih rapi.
3. Perlu pengembangan pada driver *Exthinguisher* karena masih sering terbakar pada kabel dan mosfet 3205.
4. Mendesain ulang kaki supaya lebih kokoh menopang badan robot dikarenakan sempat patah 1 kali.
5. Pengembangan pada *Exthunguisher* supaya air yang keluar lebih banyak sehingga mempermudah memadamkan api.
6. Perlunya menggunakan servo *dynamixel* supaya robot dapat berjalan lebih cepat.

DAFTAR PUSTAKA

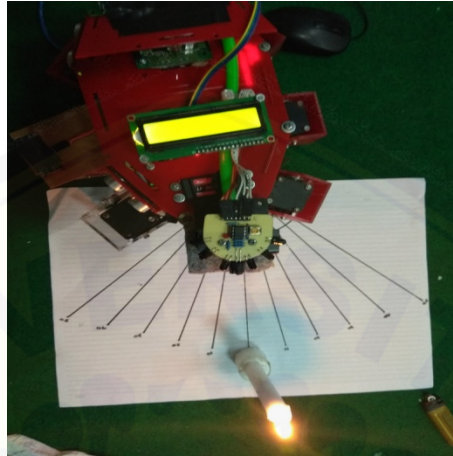
- Setiawan, I. 2006. Perancangan Software Embedded System Berbasis FSM. *Tesis*. Semarang: Fakultas Teknik Universitas Diponegoro.
- Santoso, D. D, Susilo, dan J, Wasesa. 2017. Pengembangan Robot Berkaki Enam yang dapat Mengidentifikasi Ruang pada *Map* Kontes Robot Pemadam Api Indonesia menggunakan Algoritma Pengenalan Karakter Ruang. *Techné Jurnal Ilmiah Elektroteknika*. 16(1): 11-23.
- Hidayat, L. Iswanto, dan H, Muhammad. 2011. Perancangan Robot Pemadam Api Divisi Senior Berkaki. *Jurnal Ilmiah Semesta Teknika*. 14(2): 112-116.
- Arief, U., M. 2011. Pengujian Sensor Ultrasonik PING untuk Pengukuran Level Ketinggian dan Volume Air. *Jurnal Ilmiah "Elektrikal Enjiniring" UNHAS*. Semarang: Fakultas Teknik Universitas Semarang.
- Grollman, D. H., & Jenkins, O. C. (2010). Can We Learn Finite State Machine Robot Controllers from Interactive Demonstration? *Studies in Computational Intelligence*, 407-430.
- Suryatini, F. J, Kustija, dan E, Haritman. 2013. Robot Cerdas Pemadam Api Menggunakan PING *Ultrasonic Range finder* dan UVTron *Flame Detector* Berbasis Mikrokontrol ATMega 128. *Electrans*. 12(1): 29-38.
- Nashrullah, E. A, Triwiyanto, dan B, Setiyono. 2013. Pengendalian Robot Berkaki Enam Penjejak Dinding Menggunakan Metode Kontrol Proporsional Derivatif. *Transient*. 2(2).
- Kusuma, J., W. S, Puspasari. 2013. Penerapan *invers* Kinematik Terhadap Pergerakan Kaki Pada Robot Hexapod. Palembang: Jurusan Teknik Informatika STMIK GI MDP.
- Satriyo, M., B. M, Agung, dan K, Anam 2017. Penggunaan Metode *Finite State Machine* (FSM) Untuk Mengontrol Robot Sepak Bola Beroda. Jember: Jurusan Teknik Elektro Universitas Jember
- Arduino Mega. <http://www.arduino.cc/en/Main/arduinoBoardMega> (Diakses pada 13 desember 2017)
- Hamamatsu. 2017. Flame Sensor UVTRON. [www.hamamatsu.com](http://www.hamamatsu.com) (Diakses pada 16 november 2017)
- <http://tokorobotik.com/CMPS03,2018> (Diakses pada 18 januari 2018)



**LAMPIRAN**

Lampiran 1 Dokumentasi

1. Pengambilan Data Sudut posisi api.



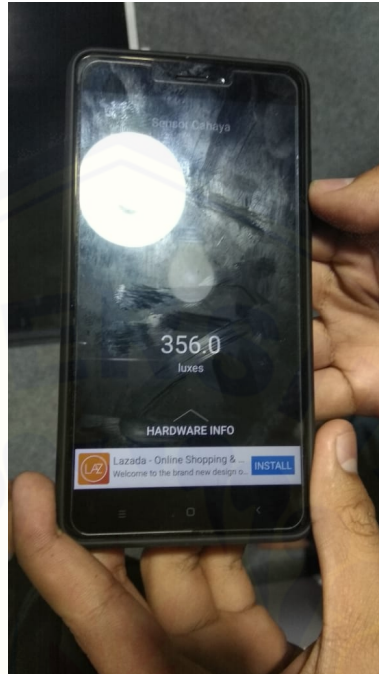
Gambar A. 1 Dokumentasi Pengambilan Data Sudut Posisi Api.

2. Pengambilan Data Jarak.



Gambar A. 2 Dokumentasi Pengambilan Data Jarak.

3. Nilai Lux Meter



Gambar A. 3 Nilai Lux Meter

## Lampiran 2 Listing Program

```
#include <Wire.h>

#include <Servo.h>

#include <LiquidCrystal_I2C.h>

//-----
//-----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_apil = 0,
titik_api2 = 0;

//-----KOMPAS-----
//-----

#define address 0x60

#define kalkom 0 //kalibrasi kompas

char kompas ;

//void cek_ruang();

int arah = 1, rumah = 3 ;int kode = 0;

LiquidCrystal_I2C lcd(0x3F, 16, 2);

Servo myservoatas;

void setup() {

    lcd.begin();

    lcd.backlight();

    lcd.print("RJ 20");

    delay(1000);

    Wire.begin();
```

```
Serial.begin(9600);

Serial.println("SIAP");

myservoatas.attach(10);

//-----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);
pinMode(11, OUTPUT);

lcd.clear();

}

void loop() {
```

```
//cek_ruang();
//pulangKanan();
memindai_dinding() ;
//s_telusur_start();
//scan_api_ruang();
//scan_api_garis();
//menuju_api();
//s_sharp();
//s_telusur_start();
// // baca_flame();
////scan_api_ruang();
// byte highByte;
// byte lowByte;
// Wire.beginTransaction (address);
// Wire.write(2);
// Wire.endTransmission();
//
// Wire.requestFrom(address, 2);
// while(Wire.available() < 2);
// highByte = Wire.read();
// lowByte = Wire.read();
// int bearing=((highByte<<8)+lowByte)/10);
// Serial.print(bearing);
// Serial.print("\xB0\t");// penampilan karakter derajat
// Serial.println(); //enter
// delay(100);
```

```
}

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 memindai_dinding() {
    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);
    if(jarak_samping_ka <= 17) {
```

```
    for(;;){
        s_telusur_start();
    }
}
else if (jarak_samping_ka>=18){
    lurus();
    if(jarak_depan<=17){
        for(;;){
            s_telusur_start();
        }
    }
}
}

void s_telusur_start() {
    kode = 0;
    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);

    //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();
    lcd.clear();
    kode = 1;
}
if (jarak_depan <= 17 ) { // hadap kiri
    kiri();kiri();kiri();
    if (L<=175 || M<=175){
        lurus();lurus();kanan();
        kanan();
        lcd.clear();
        kode = 1;
    }
}
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();
        lcd.clear();
        kode = 1;
    }
}
```



```
    }  
  }  
  
  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();  
  
      lcd.clear();  
  
      kode = 1;  
    }  
  }  
  
  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();  
  
      lcd.clear();  
  
      kode = 1;  
    }  
  }  
  
  }  
  
  else 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);
    lcd.clear();
    kode = 1;
}
}
if(kode==1){
    berdiri();
    cek_ruang();
    if (arah == 2) {
        int jarak_depan = pink(0);
        if(jarak_depan > 50)rumah = 1;
        else          rumah = 4;
    }
    else if (arah == 3)    rumah = 2;
    else {
        int jarak_kanan = pink(2);
        if(jarak_kanan < 50)rumah = 3;
        else          rumah = 4;
    }

    lcd.clear();lcd.setCursor(0,0);lcd.print("Arah
");lcd.print(arah);
```

```
        lcd.setCursor(0,1);lcd.print("Home
");lcd.print(rumah);

    delay(1000);

    while(1){

        s_telusur_lorong();

    }

}

void s_telusur_lorong() {

    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);

    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

    int flame1 = analogRead(A13); // kanan

    int flame2 = analogRead(A12); // kiri

    int flame3 = analogRead(A15); // tengah

    if (flame1 <=40 || flame2<=40 || flame3<=40){
```

```
berdiri();  
scan_api_ruang();  
}  
if (L<=175 || M<=175){  
    kanan();  
    lurus();  
    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){  
        kanan();  
        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();
        lurus();
        lurus();
        lurus();
        lurus();
        lurus();
        berdiri();
        scan_api_garis();
        lcd.clear();
```

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

```
lurus();
lurus();
lurus();
lurus();
lurus();
lurus();
lurus();
lurus();
berdiri();
scan_api_garis();
lcd.clear();
}
}
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();  
    berdiri();  
    scan_api_garis();  
    lcd.clear();  
}  
}  
  
}  
  
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();
    hadap_kiri();
    for(;;){
        s_telusur_start();
    }
}
titik_apil = 0;
}

void menuju_api(){
    int jarak_depan = pink(0);
    int jarak_samping_ka= pink(1);
```



```
        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 <=38 || flame2<=38 || flame3<=38){
            berdiri();
            scan_api_ruang();
        }
    }
    else if (jarak_depan >= 18 && jarak_samping_ka>= 18){
        serong_kanan();
        if (flame1 <=38 || flame2<=38 || flame3<=38){
            berdiri();
            scan_api_ruang();
        }
    }
}
```

```
else if (jarak_depan >=18 && jarak_samping_ka<= 15){
    serong_kiri();
    if (flame1 <=38 || flame2<=38 || flame3<=38){
        berdiri();
        scan_api_ruang();
    }
}

else if (jarak_depan >= 18 && (jarak_samping_ka>= 16 &&
jarak_samping_ka<= 17)){
    serong_kiri_dikit();
    if (flame1 <=38 || flame2<=38 || flame3<=38){
        berdiri();
        scan_api_ruang();
    }
}

if (jarak_depan >=18 && jarak_samping_ka>=19 &&
jarak_kanan>=20){
    serong_kanan_nemen();
    if (flame1 <=38 || flame2<=38 || flame3<=38){
        berdiri();
        scan_api_ruang();
    }
}

if (flame1 <=38 || flame2<=38 || flame3<=38){
    berdiri();
    scan_api_ruang();
}
}
```

```
}  
  
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_api1 = 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();  
        pulangKanan();  
    }  
}  
  
digitalWrite(11, LOW);  
myservoatas.write(120);  
titik_apil = 0;  
titik_api2 = 0;
```

```
}  
  
void keluarApiKanan() {  
  
    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);  
  
    //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));  
  
    bacaBawah();if(kode==1)goto xy;  
  
    if (jarak_depan <= 17 ) { // hadap kiri  
  
        bacaBawah();if(kode==1)goto xx1;  
  
        kiri();  
  
        bacaBawah();if(kode==1)goto xx1;  
  
        kiri();  
  
        bacaBawah();if(kode==1)goto xx1;  
  
        kiri();  
  
        bacaBawah();if(kode==1)goto xx1;  
  
        xx1;;  
  
    }  
  
    else if (jarak_depan >= 18 && jarak_samping_ka>= 18) { //  
serong kanan  
  
        serong_kanan();  
  
    }
```

```
}  
  
else if (jarak_depan >= 18 && (jarak_samping_ka>= 16 &&  
jarak_samping_ka<= 17)) { // serong kiri dikit  
  
    serong_kiri_dikit();  
  
}  
  
else if (jarak_depan >=18 && jarak_samping_ka<= 15) { //  
serong kiri  
  
    serong_kiri();  
  
}  
  
else if (jarak_depan >=18 && jarak_samping_ka>=19 &&  
jarak_kanan>=20 ) { //serong kanan nemen  
  
    serong_kanan_nemen();  
  
    bacaBawah();if(kode==1)goto xx2;  
  
    serong_kanan_nemen();  
  
    bacaBawah();if(kode==1)goto xx2;  
  
    serong_kanan_nemen();  
  
    bacaBawah();if(kode==1)goto xx2;  
  
    xx2:;  
  
}  
  
xy:;  
  
if(kode==1){  
  
    lurus();lurus();lurus();  
  
    lurus();lurus();  
  
    kanan();  
  
    berdiri();  
  
    lcd.setCursor(0,0);lcd.print("Keluar API");  
  
    delay(2000);  
  
    lcd.clear();
```



```
    }  
}  
  
void pulangKanan() {  
    kode = 0;  
    while(1) {  
        keluarApiKanan();  
        if(kode==1) break;  
    }  
    lurus(); lurus(); lurus();  
    lurus(); lurus(); kanan();  
    kode = 0;  
    while(1) {  
        kode = 0;  
        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();
    berdiri();
    kode = 1;
}
if (jarak_depan <= 19 ) { // hadap kiri
    kiri();
    bacaBawah();if(kode==1)goto x1;
    kiri();
    bacaBawah();if(kode==1)goto x1;
    kiri();
    bacaBawah();if(kode==1)goto x1;
    x1:;
}
else if (jarak_depan >= 20 && jarak_samping_ka>= 18) {
// serong kanan
    serong_kanan();
}
else if (jarak_depan >= 20 && (jarak_samping_ka>= 16 &&
jarak_samping_ka<= 17)) { // serong kiri dikit
    serong_kiri_dikit();
}
else if (jarak_depan >=20 && jarak_samping_ka<= 15) { //
serong kiri
    serong_kiri();
}
```

```
    if (jarak_depan >=20  &&  jarak_samping_ka>=19  &&
    jarak_kanan>=20 ) { //serong kanan nemen

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        serong_kanan_nemen();

        bacaBawah(); if(kode==1) goto x2;

        x2::
    }

    if(kode==1) {

        int ruang= 0;

        berdiri();

        cek_ruang();

        if      (arah == 1){

            ruang = 2;

        }

        else if (arah == 2) {

            int jarak_depan = pink(0);

            if(jarak_depan > 50) ruang = 3;

            else                ruang = 4;
```





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

```
}  
  
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);
    delay(200);
}

void sensor_garis(){
    K = analogRead(A0); //biru putih
    L = analogRead(A1); //putih
    M = analogRead(A2); //hijau putih
    N = analogRead(A3);
    O = analogRead(A4);

    Serial.print(K );
    Serial.print("K ");
```

```
Serial.print(L );
Serial.print("L ");
Serial.print(M );
Serial.print("M ");
Serial.print(N );
Serial.print("N ");
Serial.print(O );
Serial.println("O ");
delay(100);
}
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);
}
char cek_ruang(){
  byte highByte;
  byte lowByte;
  Wire.beginTransaction (address);
  Wire.write(2);
  Wire.endTransmission();
```

```
Wire.requestFrom(address, 2);
while(Wire.available() < 2);
highByte = Wire.read();
lowByte = Wire.read();
int bearing=((highByte<<8)+lowByte)/10)-kalkom;
if(bearing<0)
bearing = 360+bearing;

if ((bearing >= 0 && bearing <123)){
Serial.print("Utara\t");
kompas = 'U';
arah = 1;
}
else if (bearing >= 123 && bearing < 153){
Serial.print("Timur\t");
kompas = 'T';
arah = 2;
}
else if (bearing >=153 && bearing <205){
Serial.print("Selatan\t");
kompas = 'S';
arah = 3;
}
else if (bearing >=205 && bearing <360){
Serial.print("Barat\t");
kompas = 'B';
```

```
    arah = 4;
}
Serial.print(bearing);
Serial.print("\xB0\t");// penampilan karakter derajat
Serial.println(); //enter
    delay(100);
return kompas;
}
void bacaBawah(){
    K = analogRead(A0);//biru putih
    L = analogRead(A1); //putih
    M = analogRead(A2);//hijau putih
    if (L<=175 || M<=175){
        kode = 1;
    }
}
```