RANCANG BANGUN ROBOT *LINE FOLLOWER* PEMADAM API MEMANFAATKAN *FLAME SENSOR* DAN BLUETOOTH BERBASIS ARDUINO

BADRUT TAMAM

5215134384

Skripsi ini Ditulis untuk Memenuhi Sebagian Persyaratan dalam Memperoleh Gelar Sarjana

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI JAKARTA

2017

LEMBAR PENGESAHAN

PENGESAHAN DOSEN PEMBIMBING

Nama Dosen	Tanda Tangan	Tanggal
Drs. Jusuf Bintoro, MT NIP.196101081987031003 (Dosen Pembimbing I)	.....	4/8-2017 .....
Drs. Pitoyo Yuliatmojo, MT NIP.196807081994031003 (Dosen Pembimbing II)	.....	10/8-2017 .....

PENGESAHAN PANITIA UJIAN SKRIPSI

Nama Dosen	Tanda Tangan	Tanggal
Dr. Moch.Sukardjo,M.Pd. NIP. 195807201985031003 (Ketua Penguji)	.....	7/8-2017 .....
Drs. Wisnu Djatmiko, MT NIP. 196702141992031001 (Sekretaris)	.....	9/8-2017 .....
Dr. Baso Maruddani, MT NIP.198305022008011006 (Dosen Ahli)	.....	7/8-17 .....

Tanggal Lulus: 14-08-2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi negeri lain.
2. Karya tulis ini adalah murni gagasan, rumusan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Jakarta, 25 Juli 2017

Pembuat Pernyataan

Badrut Tamam

5215134384

KATA PENGANTAR

Puji serta syukur peneliti panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat, hidayah serta Karunia-Nya, sehingga peneliti dapat menyelesaikan skripsi dengan judul "Rancang Bangun robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino". Sholawat serta salam peneliti curahkan kepada suri tauladan akhir zaman nabi Muhammad SAW.

Skripsi ini dibuat untuk memnuhi salah satu syarat dalam menyelesaikan studi untuk meraih gelar sarjana pendidikan di Prodi Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Jaakarta. Pada kesempatan ini peneliti mengucapkan banyak terima kasih kepada semua pihak yang telah membantu peneliti, terutama kepada:

1. Drs. Jusuf Bintoro, MT selaku Dosen Pembimbing I yang telah memberikan arahan dalam menyelesaikan penelitian ini.
2. Drs. Pitoyo Yuliatmojo, MT selaku Ketua Program Studi Pendidikan Teknik Elektronika serta Dosen Pembimbing II yang selalu membimbing danb memberikan pengarahan dalam penelitian penelitian ini.
3. Kedua orang tua dan keluarga peneliti, yang telah banyak memberikan dukungan baik secara moril maupun materiil.
4. Teman-teman S1 Elektronika 2013 dan Teman-teman Tim Robotik FT UNJ.
5. Dan seluruh pihak lain yang telah membantu saya dalam menyelesaikan penyusunan proposal ini, yang tidak bisa saya sebutkan satu-persatu.

Peneliti menyadari bahwa dalam penyusunan Skripsi ini masih jauh dari kesempurnaan, untuk itu peneliti mohon maaf apabila terdapat kekurangan dan kesalahan baik dari isi maupun tulisan. Akhir kata, peneliti berharap semoga Skripsi ini bermanfaat bagi pembaca dan peneliti mendo'akan semoga segala bantuan yang telah diberikan oleh semua pihak mendapatkan balasan rahmat dari Allah SWT.

Jakarta,25 Juli 2017

Peneliti,

Badrut Tamam

ABSTRAK

BADRUT TAMAM (5215134384) "Rancang Bangun robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino". Skripsi. Jakarta: Program Studi Pendidikan Elektronika, Fakultas Teknik, Universitas Negeri Jakarta, Juli 2017. Dosen Pembimbing, Drs. Jusuf Bintoro, MT dan Drs. Pitoyo Yuliatmojo, MT.

Tujuan penelitian ini adalah membuat sistem *monitoring* ruangan yang berfungsi untuk memantau keberadaan api di dalam ruangan, dan membuat robot yang bisa memadamkan api secara otomatis berdasarkan hasil *monitoring* ruangan dengan memanfaatkan *flame sensor* yang digunakan untuk mendeteksi api di dalam ruangan dan memanfaatkan komunikasi bluetooth untuk mengirim informasi dari ruangan ke robot.

Penelitian ini dilakukan dengan menggunakan metode penelitian dan pengembangan (*Research and Development*) yang meliputi perencanaan, analisis kebutuhan perancangan, pengujian, implementasi perangkat keras (*Hardware*) dan implementasi perangkat lunak (*Software*).

Hasil dari penelitian menunjukan bahwa robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino yang telah dirancang sebelumnya berhasil dibuat, dan robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino ini sudah di coba di *track* yang mempunyai permukaan yang rata dengan ukuran 3m x 3m dengan 3 *prototype* ruangan berukuran 90cm x 60cm yang berada di dalam ruangan yang tidak tertembus cahaya matahari dan dengan api yang berasal dari nyala lilin. Pada saat tidak ada api di ruangan robot akan dalam kondisi diam di *start* dan saat ada api di ruangan, maka robot dapat langsung memadamkan api di ruangan yang terdeteksi terdapat titik apinya secara otomatis.

Kata-kata Kunci: Robot *Line Follower,* Robot Pemadam Api, *Flame Sensor,* Bluetooth, Arduino.

ABSTRACT

BADRUT TAMAM (5215134384) ***"Design a Line follower fire extinguisher utilizing Robot use flame sensor and bluetooth Arduino based". Essay. Jakarta: Program study of Electronic Education, Faculty of Engineering, State University Of Jakarta, July 2017. Supervisor,*** **Drs. Jusuf Bintoro, MT** ***and*** **Drs. Pitoyo Yuliatmojo, MT.**

The purpose of this Researh is to Create a monitoring Room system that serves to monitor the presence of fire in the room, and make a robot that can extinguish the fire automatically based on the results of monitoring the room by using a flame sensor used to detect fire in the room and utilize bluetooth communication to send information From room to robot.

This research is conducted by using research and development method which includes planning, design requirement analysis, testing, hardware implementation and software implementation.

The results of the research show that the robot line follower fire extinguisher utilizes flame sensors and bluetooth-based Arduino that has been designed previously successfully made, and the robot line follower fire extinguisher utilizing flame sensor and bluetooth-based Arduino is already on the track that has a flat surface with the size 3m x 3m with 3 prototype room measuring 90cm x 60cm which is indoors indestructible sunlight and with fire coming from a candle flame. When there is no fire in room robot will be in silent condition at start and when there is fire in room, hence robot can directly Extinguish the fire in the detected room there is a fire point automatically.

Keywords: *Line Follower robot, Fire Extinguisher Robot, Flame Sensor, Bluetooth, Arduino.*

DAFTAR ISI

LEMBAR PENGESAHAN ii
LEMBAR PERNYATAAN iii
KATA PENGANTAR iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR GAMBAR xii
DAFTAR TABEL xix
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Identifikasi Masalah 4
1.3. Pembatasan Masalah 5
1.4. Perumusan Masalah 5
1.5. Tujuan Penelitian 6
1.6. Manfaat Penelitian 6
BAB II TINJAUAN PUSTAKA 7
2.1. Landasan Teori 7
2.1.1. Pengertian Robot 7
2.1.2. Klasifikasi Robot 8
2.1.2.1. *Turtle* 8
2.1.2.2. *Vehicle* 9
2.1.2.3. *Rover* 9
2.1.2.4. Robot Manipulator 9
2.1.2.5. Robot *Walker* 10
2.1.2.6. Robot Terbang (*Flying Robot*) 10
2.1.2.7. Robot Humanoid 10
2.1.3. Teori Api 11
2.1.4. Teori Pemadaman Api 12
2.1.5. Pengertian Robot *Line Follower* 12
2.1.6. Pengertian Robot Pemadam Api 14
2.1.7. Pengertian Robot *Line Follower* Pemadam Api 15

2.1.8. Arduino 15
2.1.3.1. Arduino Uno 18
2.1.3.1.1. Kelebihan Arduino Uno 19
2.1.3.1.2. Kekurangan Arduino Uno 20
2.1.3.2. Arduino Mega 2560 21
2.1.3.2.1. Kelebihan Arduino Mega 2560 22
2.1.3.2.2. Kekurangan Arduino Mega 2560 23
2.1.9. Arduino IDE(*Integrated Development Environment*) 24
2.1.10. Sensor 25
2.1.11. Sensor Garis 26
2.1.6.1. Bagian-Bagian Sensor Garis 28
2.1.6.1.1. Resistor 28
2.1.6.1.2. LED (*Light Emitting Diode*) 29
2.1.6.1.3. Photodioda 31
2.1.12. *Flame* Sensor 32
2.1.13. Modul *Flame* Sensor KY-026 33
2.1.14. Bluetooth 35
2.1.9.1. Prinsip Kerja Bluetooth 37
2.1.9.2. Protokol Bluetooth 39
2.1.9.2.1. Protokol Inti 40
2.1.9.2.2. Protokol Pengganti Kabel 40
2.1.9.2.3. Protokol Telefoni 40
2.1.9.2.4. Protokol-Protokol Adopsi 40
2.1.9.3. Sistem Koreksi Error 41
2.1.9.4. Macam-Macam Kelas Bluetooth 44
2.1.9.5. Kelebihan Bluetooth 44
2.1.9.6. Kekurangan Bluetooth 45
2.1.15. Modul Bluetooth HC-05 45
2.1.16. *Push Button* 47
2.1.17. Modul LCD (*Liquid Crystal Display*) 16 x 2 49
2.1.18. Modul I2C (*Inter Integreted Circuit*) 51
2.1.19. Motor DC (*Direct Current*) 54

2.1.20. Motor Servo 56
2.1.21. *Driver Motor* 59
2.1.22. *Driver Motor* IC L298N 60
2.1.23. *Driver Motor Optocoupler* 62
2.1.24. Rangkaian Regulator 7805 63
2.2. Kerangka Berfikir 64
2.2.1. Blok Diagram Sistem 65
2.2.2. *Flow Chart* Sistem 66
2.3. Penelitian Yang Sudah Pernah dilakukan 67
BAB III METODOLOGI PENELITIAN 69
3.1. Tempat dan Waktu Penelitian 69
3.2. Alat dan Bahan Penelitian 69
3.2.1. Alat Penelitian 69
3.2.1.1. Perangkat Keras 69
3.2.1.2. Perangkat Lunak 70
3.2.2. Bahan Penelitian 70
3.3. Diagram Alir Penelitian 71
3.3.1. Analisis Kebutuhan Sistem 73
3.3.2. Perancangan Sistem 74
3.3.3. Tahap Pengujian 75
3.3.4. Implementasi Sistem Perangkat Keras 76
3.3.5. Implementasi Sistem Perangkat Lunak 77
3.4. Rancangan Penelitian 77
3.4.1. Perancangan Perangkat Keras Sistem 77
3.4.1.1. Perancangan *Base* Robot 77
3.4.1.2. Perancangan *Standing* sebagai Tempat Servo 79
3.4.1.3. Perancangan Rangkaian Sensor Garis Robot 82
3.4.1.4. Perancangan *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805 84
3.4.1.5. Perancangan *Shield* Uno dan Rangkaian Regulator 7805 87
3.4.1.6. Perancangan Rangkaian *Driver motor* dengan Menggunakan IC L298N 89

3.4.1.7. Perancangan Rangkaian *Driver Motor* dengan Menggunakan *Optocoupler* 91
3.4.1.8. Perancangan Rangkaian *Push Button* 93
3.4.1.9. Perancangan Integrasi Modul Bluetooth HC-05 dengan Arduino 94
3.4.1.10. Perancangan Jalur(*Track*) dan Ruangan 96
3.4.2. Perancangan Perangkat Lunak Sistem 97
3.5. Teknik dan Prosedur Pengumpulan Data 99
3.6. Teknik Analisa Data 100
3.6.1. Pengujian *Flame Sensor* KY-026 di Dalam Ruangan dan di Robot. 100
3.6.2. Pengujian Komunikasi Serial Bluetooth 102
3.6.3. Pengujian Sensor Garis 104
3.6.4. Pengujian Modul I2C LCD dan Modul LCD 16x2 105
3.6.5. Pengujian *Driver motor* dengan Menggunakan L298N 105
3.6.6. Pengujian Motor Servo sebagai Penggerak Penyemprot 107
3.6.7. Pengujian *Driver motor* dengan Menggunakan *Optocoupler* 108
3.6.8. Tes Pergerakan Robot 108
3.6.9. Pengujian Robot dengan Variasi Tempat Pemadaman Api 109
3.6.10. Uji Kelayakan Pemadaman Api oleh Robot 112
BAB IV HASIL PENELITIAN 116
4.1. Deskripsi Hasil Penelitian 116
4.1.1. Prinsip Kerja Alat 116
4.1.2. Langkah-Langkah Penggunaan Alat 117
4.1.3. Kekurangan Alat 118
4.2. Analisis Hasil Penelitian 119
4.2.1. Hasil Pengujian *Flame Sensor* KY-026 123
4.2.2. Hasil Pengujian Komunikasi Bluetooth HC-05 129
4.2.3. Hasil Pengujian Sensor Garis 131
4.2.4. Hasil Pengujian Modul LCD 16x2 dan I2C LCD 149
4.2.5. Hasil Pengujian *Driver Motor* dengan Menggunakan L298N 150
4.2.6. Hasil Pengujian Motor Servo sebagai Penggerak Penyemprot 151
4.2.7. Hasil Pengujian *Driver Motor* dengan Menggunakan *Optocoupler* 152

4.2.8. Hasil Tes Pergerakan Robot 153
4.2.9. Hasil Pengujian Kelayakan Robot *Line Follower* Pemadam Api dengan Memanfaatkan Modul Bluetooth HC-05 sebagai Pelacak Lokasi Api Berbasis Arduino 153
4.3. Pembahasan 157
4.4. Aplikasi Hasil Penelitian 160
BAB V KESIMPULAN DAN SARAN 161
5.1. Kesimpulan 161
5.2. Saran 162
DAFTAR PUSTAKA 163
LAMPIRAN-LAMPIRAN 167

DAFTAR GAMBAR

Gambar 2.1 Robot 8
Gambar 2.2 Robot Manipulator 9
Gambar 2.3 Robot Terbang (*Flying Robot*) 10
Gambar 2.4 Robot Humanoid 11
Gambar 2.2 Robot *Line Follower* 14
Gambar 2.3 Robot Pemadam Api 15
Gambar 2.7 Macam-Macam *Board* Arduino 17
Gambar 2.8 *Board* Arduino Uno 19
Gambar 2.9 *Board* Arduino Mega 2560 21
Gambar 2.10 Tampilan *Sketch* Arduino IDE 25
Gambar 2.11 Macam-Macam Sensor 26
Gambar 2.12 Rangkaian Sensor Garis 27
Gambar 2.13 (a) Simbol Resistor 29
(b) Bentuk Fisik Resistor 29
Gambar 2.14 (a) Simbol LED 30
(b) Bentuk Fisik LED 30
Gambar 2.15 (a) Simbol Photodioda 32
(b) Bentuk Fisik Photodioda 32
Gambar 2.16 (a) Bentuk Fisik *Flame Sensor* UVtron 33
(b) Bentuk Fisik Modul *Flame Sensor* KY-026 33
Gambar 2.17 Modul *Flame Sensor* KY-026 34
Gambar 2.18 Bluetooth 36
Gambar 2.19 Contoh Skema Pikonet dan *Scatternet* 38
Gambar 2.20 Skema FH-TDD 39
Gambar 2.21 Arsitektur Protokol Bluetooth 39
Gambar 2.22 Contoh Transmisi Ulang 41
Gambar 2.23 Operasi Perangkat Penerima dalam Menentukan bit ARQ 42
Gambar 2.24 Operasi Transmisi-Ulang untuk Paket-Paket dengan CRC 43
Gambar 2.25 Modul *Bluetooth* HC-05 46
Gambar 2.26 (a) Simbol *Push Button* 48

(b) Bentuk Fisik *Push Button* 48
Gambar 2.27 Modul LCD (*Liquid Crystal Display*) 16x2 50
Gambar 2.28 Alur Kontrol I2C 52
Gambar 2.29 I2C LCD 53
Gambar 2.30 Motor DC 55
Gambar 2.31 Motor servo 57
Gambar 2.32 Bagian-Bagian Motor Servo 59
Gambar 2.33 (a) Bentuk Fisik IC L298N 61
(b) Blok diagran IC L298N 61
Gambar 2.34 Skema Rangkaian *Driver* Motor IC L298N 62
Gambar 2.35 Skema Rangkaian *Driver Motor Optocoupler* 63
Gambar 2.36 Bentuk Fisik Regulator 7805 64
Gambar 2.37 Blok Diagram Sistem 65
Gambar 2.38 *Flow Chart* Robot 66
Gambar 2.39 *Flow Chart Monitoring* Ruangan 67
Gambar 3.1 Tahap-tahap Metodologi Penelitian Perancangan alat 72
Gambar 3.2 Rancangan *Base* Robot 78
Gambar 3.3 Integrasi antara Modul LCD 16x2 dengan Modul I2C LCD 79
Gambar 3.4 Integrasi antara Modul I2C LCD dengan Arduino Mega 2560 dan Rangkaian Regulator 79
Gambar 3.5 Rancangan *Standing* Robot 80
Gambar 3.6 Integrasi antara Motor Servo dengan Arduino Mega 2560 dan Rangkaian Regulator 81
Gambar 3.7 Integrasi antara Rangkaian *Driver motor Optocoupler* dengan Arduino Mega dan Sumber Tegangan 81
Gambar 3.8 Integrasi antara *Flame Sensor* dengan Arduino Mega 2560 dan Rangkaian Regulator 82
Gambar 3.9 Skema Rangkaian Sensor Garis Robot 83
Gambar 3.10 *Layout* Sensor Garis Robot 83
Gambar 3.11 Desain Rangkaian Sensor Garis 84
Gambar 3.12 Integrasi Rangkaian Sensor Garis dengan Arduino Mega 2560 dan Rangkaian Regulator 84

Gambar 3.13 Skema Rangkaian *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805 85
Gambar 3.14 *Layout Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805 85
Gambar 3.15 Desain Rangkaian *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805 86
Gambar 3.16 Integrasi antara Rangkaian Regulator dengan Arduino Mega 2560. 87
Gambar 3.17 Skema Rangkaian *Shield* Arduino Uno dan Rangkaian Regulator 7805 88
Gambar 3.18 *Layout Shield* Arduino Uno dan Rangkaian Regulator 7805 88
Gambar 3.19 Integrasi antara Rangkaian Regulator dengan Arduino Uno 89
Gambar 3.20 Skema Rangkaian *Driver* Motor dengan Menggunakan IC L298N 90
Gambar 3.21 *Layout driver* Motor dengan Menggunakan IC L298N 90
Gambar 3.22 Desain Rangkaian *Driver* Motor Dengan Menggunakan IC L298N 90
Gambar 3.23 Integrasi antara rangkaian *Driver motor* dengan Arduino Mega 2560 dan rangkaian regulator 91
Gambar 3.24 Skema Rangkaian *Driver* Motor dengan Menggunakan *Optocoupler* 92
Gambar 3.25 *Layout Driver* Motor dengan Menggunakan *Optocoupler* 92
Gambar 3.26 Skema Rangkaian *Push Button* 93
Gambar 3.27 *Layout* Rangkaian *Push Button* 93
Gambar 3.28 Desain Rangkaian *Push Button* 94
Gambar 3.29 Integrasi antara rangkaian *Push Button* dengan Arduino Mega 2560 94
Gambar 3.30 Integrasi antara Modul Bluetooth HC-05 dengan Arduino Mega 2560 dan Rangkaian Regulator 95
Gambar 3.31 Integrasi antara Modul Bluetooth Hc-05 dengan Arduino Uno dan Rangkaian Regulator 95
Gambar 3.32 Desain Jalur dan Ruangan yang digunakan Untuk Penelitian 96

Gambar 3.33 Integrasi antara *Flame Sensor d*engan Arduino Uno dan Rangkaian Regulator 97
Gambar 3.34 Penomoran Ruangan 110
Gambar 4.1 Rangkaian Sensor Garis 119
Gambar 4.2 Rangkaian *Push Button* 120
Gambar 4.3 Rangkaian *Shield* Arduino Mega 2560 120
Gambar 4.4 Rangkaian *Shield* Arduino Uno 121
Gambar 4.5 Rangkaian *Driver Motor* dengan Menggunakan L298N 121
Gambar 4.6 Rangkaian *Driver Motor* dengan Menggunakan *Optocoupler* 122
Gambar 4.7 *Desain Base* Robot 122
Gambar 4.8 *Desain Standing* Servo Robot 122
Gambar 4.9 *Desain Track*(Jalur) 123
Gambar 4.10 Pembacaan Nilai Analog *Flame Sensor* di Ruang 1 dalam Kondisi Ruangan Tanpa Api 124
Gambar 4.11 Pembacaan Nilai Analog *Flame Sensor* di Ruang 2 dalam Kondisi Ruangan Tanpa Api 125
Gambar 4.12 Pembacaan Nilai Analog *Flame Sensor* di Ruang 3 dalam Kondisi Ruangan Tanpa Api 125
Gambar 4.13 Pembacaan Nilai Analog *Flame Sensor* di Ruang 1 dalam Kondisi Ruangan Ada Api 126
Gambar 4.14 Pembacaan Nilai Analog *Flame Sensor* di Ruang 2 dalam Kondisi Ruangan Ada Api 126
Gambar 4.15 Pembacaan Nilai Analog *Flame Sensor* di Ruang 3 dalam Kondisi Ruangan Ada Api 127
Gambar 4.16 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 1 di Alas Berwarna Putih 132
Gambar 4.17 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 2 di Alas Berwarna Putih 133
Gambar 4.18 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 3 di Alas Berwarna Putih 133
Gambar 4.19 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 4 di Alas Berwarna Putih 133

Gambar 4.20 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 5 di Alas Berwarna Putih 134
Gambar 4.21 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 6 di Alas Berwarna Putih 134
Gambar 4.22 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 7 di Alas Berwarna Putih 134
Gambar 4.23 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 8 di Alas Berwarna Putih 135
Gambar 4.24 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 9 di Alas Berwarna Putih 135
Gambar 4.25 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 10 di Alas Berwarna Putih 135
Gambar 4.26 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 1 di Alas Berwarna Hitam 136
Gambar 4.27 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 2 di Alas Berwarna Hitam 136
Gambar 4.28 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 3 di Alas Berwarna Hitam 136
Gambar 4.29 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 4 di Alas Berwarna Hitam 137
Gambar 4.30 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 5 di Alas Berwarna Hitam 137
Gambar 4.31 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 6 di Alas Berwarna Hitam 137
Gambar 4.32 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 7 di Alas Berwarna Hitam 138
Gambar 4.33 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 8 di Alas Berwarna Hitam 138
Gambar 4.34 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 9 di Alas Berwarna Hitam 138
Gambar 4.35 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 10 di Alas Berwarna Hitam 139

Gambar 4.36 Hasil Pengukuran Nilai ADC pada Sensor Garis 1 di Alas Berwarna Putih 139
Gambar 4.37 Hasil Pengukuran Nilai ADC pada Sensor Garis 2 di Alas Berwarna Putih 140
Gambar 4.38 Hasil Pengukuran Nilai ADC pada Sensor Garis 3 di Alas Berwarna Putih 140
Gambar 4.39 Hasil Pengukuran Nilai ADC pada Sensor Garis 4 di Alas Berwarna Putih 141
Gambar 4.40 Hasil Pengukuran Nilai ADC pada Sensor Garis 5 di Alas Berwarna Putih 141
Gambar 4.41 Hasil Pengukuran Nilai ADC pada Sensor Garis 6 di Alas Berwarna Putih 142
Gambar 4.42 Hasil Pengukuran Nilai ADC pada Sensor Garis 7 di Alas Berwarna Putih 142
Gambar 4.43 Hasil Pengukuran Nilai ADC pada Sensor Garis 8 di Alas Berwarna Putih 143
Gambar 4.44 Hasil Pengukuran Nilai ADC pada Sensor Garis 9 di Alas Berwarna Putih 143
Gambar 4.45 Hasil Pengukuran Nilai ADC pada Sensor Garis 10 di Alas Berwarna Putih 144
Gambar 4.46 Hasil Pengukuran Nilai ADC pada Sensor Garis 1 di Alas Berwarna Hitam..... 144
Gambar 4.47 Hasil Pengukuran Nilai ADC pada Sensor Garis 2 di Alas Berwarna Hitam..... 145
Gambar 4.48 Hasil Pengukuran Nilai ADC pada Sensor Garis 3 di Alas Berwarna Hitam..... 145
Gambar 4.49 Hasil Pengukuran Nilai ADC pada Sensor Garis 4 di Alas Berwarna Hitam..... 146
Gambar 4.50 Hasil Pengukuran Nilai ADC pada Sensor Garis 5 di Alas Berwarna Hitam..... 146
Gambar 4.51 Hasil Pengukuran Nilai ADC pada Sensor Garis 6 di Alas Berwarna Hitam..... 147

Gambar 4.52 Hasil Pengukuran Nilai ADC pada Sensor Garis 7 di Alas Berwarna Hitam..... 147
Gambar 4.53 Hasil Pengukuran Nilai ADC pada Sensor Garis 8 di Alas Berwarna Hitam..... 148
Gambar 4.54 Hasil Pengukuran Nilai ADC pada Sensor Garis 9 di Alas Berwarna Hitam..... 148
Gambar 4.55 Hasil Pengukuran Nilai ADC pada Sensor Garis 10 di Alas Berwarna Hitam..... 149

DAFTAR TABEL

Tabel 2.1 Alokasi Frekuensi untuk Bluetooth di Berbagai Negara 36
Tabel 2.2 fungsi-fungsi pin LCD (*Liquid Crystal Display*) 50
Tabel 3.1 Pin I/O Arduino Mega 2560 yang digunakan 98
Tabel 3.2 Pin I/O Arduino UNO yang digunakan 98
Tabel 3.3 Tabel Pengukuran Tegangan dan Nilai Analog pada *Flame* Sensor di dalam Ruangan 101
Tabel 3.4 Tabel Pengukuran Tegangan dan Nilai Analog pada *Flame Sensor* di Robot 102
Tabel 3.5 Kriteria Pengujian Komunikasi Serial Dua *Bluetooth* 103
Tabel 3.6 Tabel Pengujian dan Pengukuran Sensor Garis 105
Tabel 3.7 Tabel Pengujian Modul I2C LCD dan Modul LCD 16x2 105
Tabel 3.8 Tabel Pengujian Pergerakan Motor Menggunakan *Driver* L298N 106
Tabel 3.9 Tabel Pengujian Tegangan Keluaran Rangkaian *Driver* L298N 107
Tabel 3.10 Tabel Pengujian Pergerakan Motor Servo sebagai Penggerak Penyemprot 107
Tabel 3.11 Tabel Pengujian Data Pergerakan Motor yang digunakan untuk Memadamkan Api 108
Tabel 3.12 Tabel Pengamatan Gerakan Robot dengan Menggerakan Roda Robot 109
Tabel 3.13 Variasi Pemadaman api 111
Tabel 3.14 Pengujian Posisi Api 1 113
Tabel 3.15 Pengujian Posisi Api 2 113
Tabel 3.16 Pengujian Posisi Api 3 113
Tabel 3.17 Pengujian Posisi Api 4 114
Tabel 3.18 Pengujian Posisi Api 5 114
Tabel 3.19 Pengujian Posisi Api 6 114
Tabel 3.20 Pengujian Posisi Api 7 114
Tabel 3.21 Uji Kelayakan *Line Follower* Pemadam Api 115
Tabel 4.1 Hasil Pengukuran Nilai Analog pada *Flame Sensor* di Dalam Ruangan 124

Tabel 4.2 Hasil Pengukuran Nilai Analog pada *Flame Sensor* di Robot 127
Tabel 4.3 Hasil Pengujian Komunikasi Serial Dua Bluetooth 130
Tabel 4.4 Hasil Pengujian dan Pengukuran Sensor Garis 132
Tabel 4.5 Hasil Pengujian Modul I2C LCD dan Modul LCD 16x2 149
Tabel 4.6 Hasil Pengujian Data Pergerakan Motor Menggunakan *Driver* L298N 150
Tabel 4.7 Hasil Pengujian Tegangan Keluaran Rangkaian *Driver* L298N 150
Tabel 4.8 Hasil Pengujian Pergerakan Motor Servo sebagai Penggerak Penyemprot 151
Tabel 4.9 Hasil Pengujian Data Pergerakan Motor yang digunakan untuk Memadamkan Api 152
Tabel 4.10 Hasil Data Pengamatan Gerakan Robot dengan Menggerakan Roda Robot 153
Tabel 4.11 Pengujian Posisi Api 1 154
Tabel 4.12 Pengujian Posisi Api 2 154
Tabel 4.13 Pengujian Posisi Api 3 154
Tabel 4.14 Pengujian Posisi Api 4 155
Tabel 4.15 Pengujian Posisi Api 5 155
Tabel 4.16 Pengujian Posisi Api 6 155
Tabel 4.17 Pengujian Posisi Api 7 156
Tabel 4.18 Uji Kelayakan *Line Follower* Pemadam Api 156
Tabel 4.19 Nilai Error pada Masing-Masing Pin Arduino Mega 2560 158

BAB I

PENDAHULUAN

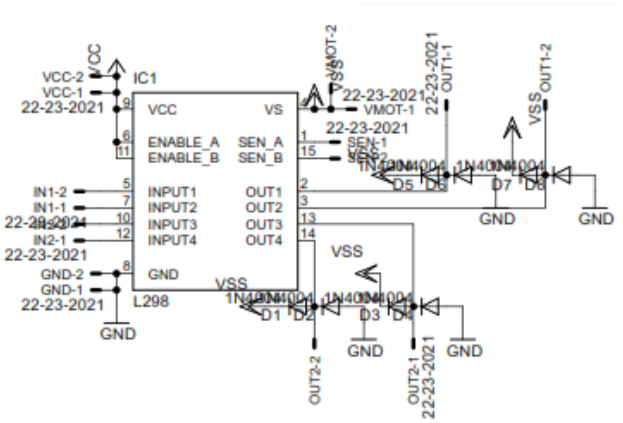
1.1. Latar Belakang

Ilmu pengetahuan dalam bidang IPTEK kini telah mengalami perkembangan yang sangat pesat, hal ini disebabkan karena semakin majunya sumber daya manusia yang ada dan semakin butuhnya manusia akan sistem atau alat yang efektif dan efisien untuk memenuhi kebutuhan kehidupannya dalam sehari-hari.

Salah satu perkembangan ilmu di bidang teknologi yang sedang berkembang dan masih sangat diminati oleh para peneliti saat ini adalah robot. Kata robot berasal dari bahasa Czezch, *robota*, yang berarti pekerja, mulai populer ketika seorang penulis berbangsa Czezch(Ceko), Karl Capek, membuat pertunjukan dari lakon komedi yang ditulisnya pada tahun 1921yang berjudul RUR (*Rossum's Universal Robot*). Pitowarno bercerita tentang mesin yang menyerupai manusia tapi mampu bekerja terus-menerus tanpa lelah. (Pitowarno, 2016:1).

Hampir semua bidang keteknikan dan industri di dunia ini menggunakan robot. Alasannya, robot mampu melakukan suatu pekerjaan tertentu secara terus-menerus dengan tahap-tahap yang telah ditentukan. Berbeda dengan manusia yang memiliki kecenderungan untuk mengalami kejenuhan jika dihadapkan pada pekerjaan yang sama sehingga kemungkinan untuk melakukan kesalahan kerja menjadi besar. Selain itu, robot diharapkan dapat menjangkau daerah yang tidak dapat dijangkau manusia karena resiko yang sangat besar. Misalnya, pengambilan

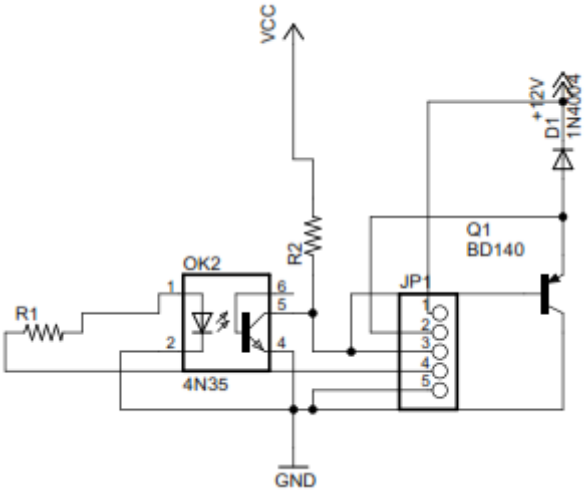
5VDC lalu langsung memberikan tegangan analog pada pin masukan IC, dan cara kedua adalah dengan menggunakan 4 pin input digital dan 2 pin output analog yaitu dengan memberikan tegangan digital pada pin masukan IC lalu memberikan tegangan analog pada pin enable untuk mengatur kecepatan putaran motor. Gambar skema rangkaian driver motor IC L298N dapat dilihat pada Gambar 2.34.



Gambar 2.34 Skema Rangkaian Driver Motor IC L298N

2.1.21.2 Driver Motor Optocoupler

Driver optocoupler adalah sebuah rangkaian elektronika yang menggunakan optocoupler sebagai pengendali pergerakan motor yang akan dikendalikannya, dalam penelitian ini motor yang akan dikendalikan geraknya adalah sebuah motor DC yang berfungsi sebagai pemadam api. Gambar skema rangkaian driver motor optocoupler dapat dilihat pada Gambar 2.35.



Gambar 2.35 Skema Rangkaian Driver motor Optocoupler

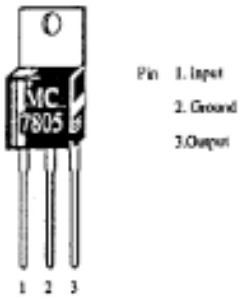
Sesuai dengan Gambar 2.35, untuk mengaktifkan motor kita hanya cukup mengaktifkan optocoupler saja, daya yang dibutuhkan untuk mengaktifkan optocoupler pun tidak terlalu besar, karena kita hanya membutuhkan daya untuk menghidupkan led di dalam optocoupler Dengan menggunakan rangkaian tersebut, peluang kerusakan mikrokontroler yang terjadi akibat kerusakan yang terjadi pada motor DC sehingga daarus balik dari motor DC akan kecil karena mikrokontroler hanya terhubung dengan bagian led optocoupler, sedangkan motor DC terhubung dengan bagian transistor optocoupler, keduanya adalah dua blok yang berbeda sehingga apabila terjadi kerusakan pada motor DC yang memiliki peluang ikut rusak hanyalah transistor dalam optocouplernya saja.

2.1.22. Rangkaian Regulator

Rangkaian catu daya yang stabil dan teregulasi sangat diperlukan pada semua rangkaian elektronika. Banyak sekali rangkaian catu daya yang dilengkapi dengan stabilisator dan regulator tegangan dari rangkaian sederhana sampai rangkaian yang cukup rumit. Stabilizer dan regulator adalah bagian yang

terdiri dari komponen diode zener, transistor, IC atau kombinasi ketiganya. Komponen-komponen tersebut fungsinya adalah sebagai penstabil. Dan mengatur tegangan yang berasal dari rangkaian penyaring. (Anwar, dkk., 2010:3). Sedangkan menurut Firmansyah, dkk.(2016:4) Regulator tegangan adalah bagian power supply yang berfungsi untuk memberikan stabilitas output pada suatu power supply.

Rangkaian regulator adalah rangkaian yang digunakan untuk menyetabilkan tegangan input sehingga dapat mengurangi kedaya dan kekeliruan yang terjadi pada suatu sistem. Gambar regulator 7805 dapat dilihat pada Gambar 2.36.



Gambar 2.36 Bentuk Fisik Regulator 7805

2.2. Kerangka Berfikir

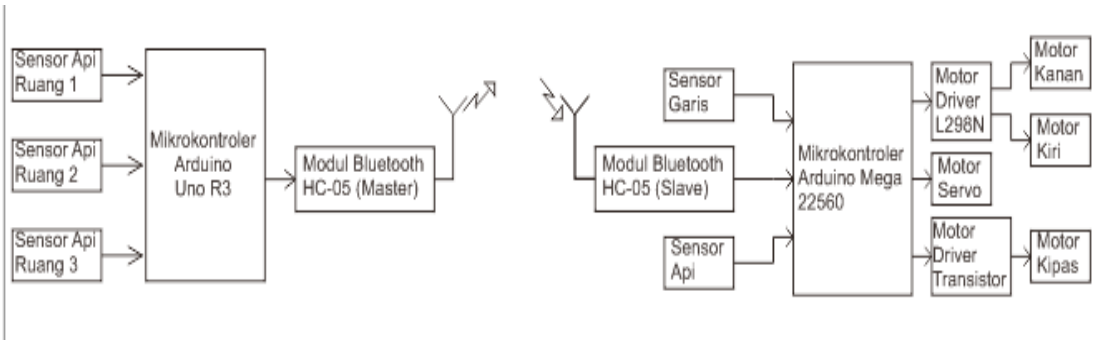
Robot line follower adalah robot yang berfungsi untuk mengikuti garis yang berada dibawahnya, garis tersebut adalah garis yang warnanya kontras dengan warna dasar alasnya

Robot pemadam api adalah sebuah robot yang digunakan untuk mencari lokasi api serta memadamkannya.

Robot line follower pamadam api adalah sebuah robot yang berfungsi untuk mengikuti garis yang berada di bawahnya dan pada garis tertentu robot ini berfungsi untuk memadamkan api.

Diduga, robot robot line follower pamadam api bisa mengikuti garis yang berwarna kontras yang berada di bawahnya, dan pada garis tertentu robot ini akan bertugas untuk memadamkan api yang berasal dari nyala lilin.

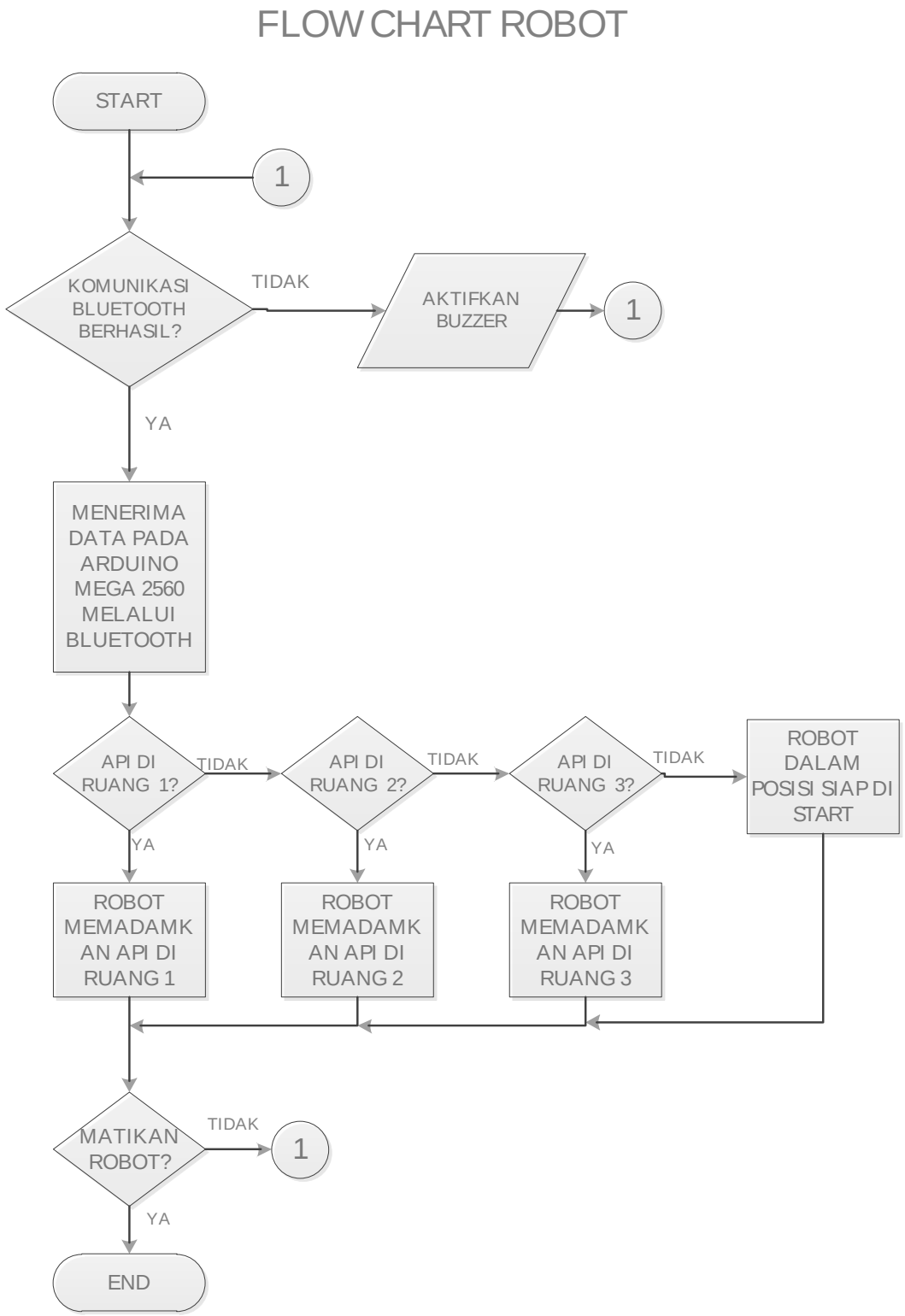
2.2.1. Blok diagram Sistem



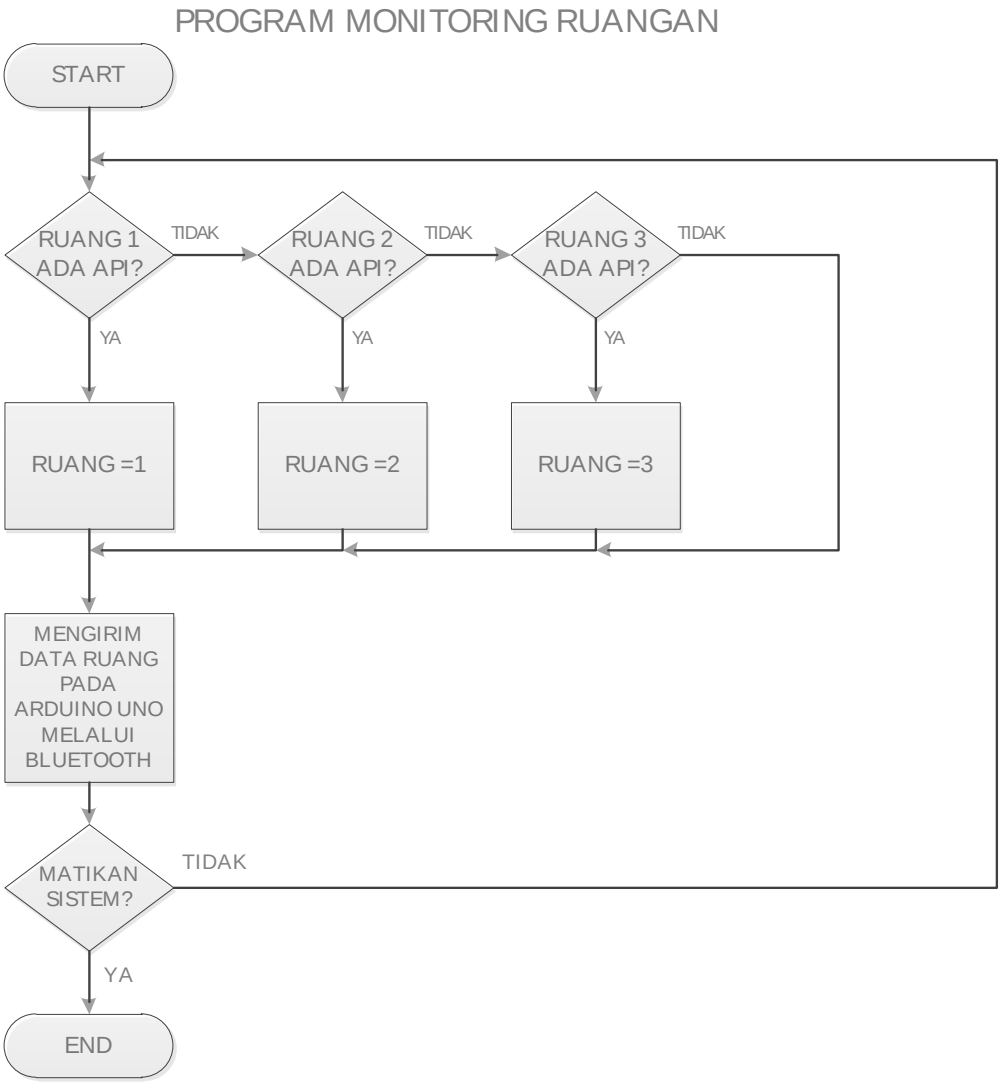
Gambar 2.37 Blok Diagram Sistem

Berdasarkan blok diagram pada Gambar 2.37, sistem menggunakan dua kontroler yang akan berkomunikasi dengan menggunakan bluetooth. Kontroler Arduino Uno berfungsi untuk memonitoring ketiga ruangan, sensor yang digunakan Arduino Uno untuk memonitoring ruangan adalah sensor api dan sensor asap karena yang akan dideteksi di dalam ruangan adalah api yang berasal dari lilin dan mengirimkan data mengenai lokasi api dengan menggunakan bluetooth. Lalu kontroler yang berfungsi sebagai robot line follower pemadam adalah Arduino Mega 2560, robot ini akan menunggu data yang akan diterima menggunakan bluetooth, dan bertugas untuk mengikuti garis yang sudah disediakan dengan menggunakan sensor garis, lalu mendeteksi letak api yang ada di dalam ruangan dengan menggunakan sensor api dan dengan bantuan motor servo untuk meggerakan sensor api dan kemudian memadamkan api dengan menggunakan motor kipas. Robot ini akan bergerak dengan menggunakan dua motor DC yang di kendalikan dengan menggunakan driver motor IC L298N.

2.2.2.Flow Chart Sistem



Gambar 2.38 Flow Chart Robot



Gambar 2.39 Flow Chart Monitoring Ruangan

2.3. Penelitian Yang Sudah Pernah dilakukan

1. Penelitian yang dilakukan Rizki Riyadi dengan judul Prototipe Robot Mobil Pengangkut Barang Otomatis dengan Smartphone Android sebagai Pengendali sebagai syarat kelulusan di Universitas Negeri Jakarta. Penelitian yang dilakukan untuk membuat robot yang bisa memindahkan barang dari satu ruangan ke ruangan yang lainnya, pemindahan barang ini bisa menggunakan metode otomatis yang menggunakan prinsip robot

line follower dan menggunakan metode manual yang dikendalikan dengan menggunakan smartphone (Riyadi,2015)

2. Penelitian yang dilakukan Aziz Akbar Guntama dengan judul Prototype Robot Line Follower Pengangkut Sampah dengan Memanfaatkan Solar Cell Berbasis Mikrokontroler Arduino Mega 2560 sebagai syarat kelulusan di Universitas Negeri Jakarta. Penelitian yang dilakukan untuk membuat robot yang bisa digunakan untuk mengambil sampah dari tempat sampah, lalu memindahkannya ke bak penyimpanan yang berada di dalam robot. Prototype Robot Line Follower Pengangkut Sampah dengan Memanfaatkan Solar Cell Berbasis Mikrokontroler Arduino Mega 2560 menggunakan prinsip line folllower untuk perpindahannya dan menggunakan sensor warna untuk mengenali tempat sampah ya berada di beberapa tempat. (Guntama,2017)

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Tempat penelitian dilakasanakan di Laboratorium Mekatronika dan Robotika Teknik Elektro Universitas Negeri Jakarta dan di Bengkel Mekanik Teknik Elektro Universitas Negeri Jakarta. Waktu penelitian dimulai pada tanggal 5 Desember 2016 sampai dengan bulan Juli 2017. Waktu tersebut cukup efektif untuk melakukan penelitian.

3.2. Alat dan Bahan Penelitian

Alat dan bahan yang digunakan dalam Rancang Bangun Robot *Line Follower* Pemadam Api Memanfaatkan *Flame Sensor* dan Bluetooth Berbasis Arduino adalah sebagai berikut:

3.2.1. Alat Penelitian

3.2.1.1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitiaan ini adalah sebagai berikut:

1. *Personal Computer* (PC) atau *Notebook*
2. Multitester
3. Solder
4. Obeng satu set
5. Bor AC
6. Bor DC
7. Tang jepit
8. Tang potong
9. Modul *flame sensor* KY-026

10. Modul sensor asap MQ-2
11. Modul LCD (*Liquid Crystal Display*) 16x2
12. Arduino Mega 2560
13. Arduino Uno
14. *Driver motor* DC
15. Motor DC
16. Motor Servo

3.2.1.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitiaan ini adalah sebagai berikut:

1. Arduino IDE
2. *Google sketch up*
3. *Corel draw*
4. *Eagle*
5. *Proteus SP*
6. *Microsoft Word* 2010
7. *Microsoft Visio* 2010

3.2.2. Bahan Penelitian

Bahan habis pakai yang digunakan dalam penelitiaan ini adalah sebagai berikut:

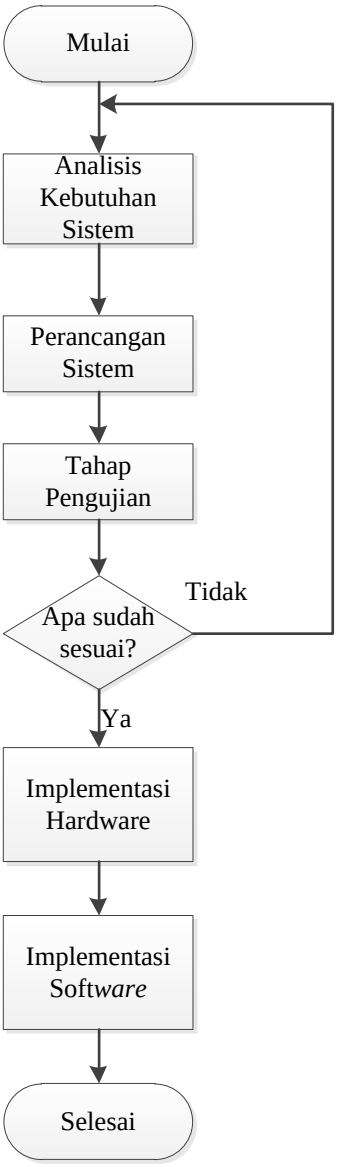
1. akrilik
2. Resistor
3. *R-pack*
4. LED 3mm
5. Photodioda 3mm

6. *Pinhead*
7. *Socket pinhead*
8. Kabel pelangi
9. Kabel *jumper*
10. Modul I2C LCD (*Liquid Crystal Display*) 16x2
11. Saklar
12. *Push button*
13. Modul Bluetooth HC-05
14. Papan
15. IC regulator 7805
16. *Elco* (*Electrolit Condencer*)
17. IC L298N
18. Dioda
19. Timah
20. PCB
21. HCL
22. H_2O_2

3.3. Diagram Alir Penelitian

Diagram alur penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Metode Penelitian dapat diartikan sebagai langkah-langkah yang dilakukan oleh peniliti agar bisa membuat produk atau mengembangkan produk yang sudah ada. Metode yang digunakan dalam merancang Robot *Line Follower* Pemadam Api Memanfaatkan *Flame Sensor* dan Bluetooth Berbasis Arduino menggunakan metode *research and*

development (penelitian dan pembangunan). Tahap-tahap metodologi peneelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahap-tahap Metodologi Penelitian Perancangan alat

3.3.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah tahap penelitian menentukan kebutuhan yang dibutuhkan oleh sistem agar Rancang Bangun Robot *Line Follower* Pemadam Api Memanfaatkan *Flame Sensor* dan Bluetooth Berbasis Arduino dapat dibuat sesuai dengan tujuan penelitian. Untuk memenuhi tujuan tersebut peneliti membagi beberapa hal yang dibutuhkan oleh sistem menjadi seperti berikut:

1. Robot membutuhkan baterai sebagai sumber tegangan 12V dan rangkaian regulator sebagai sumber tegangan 5V.
2. Robot membutuhkan sensor garis yang digunakan untuk membedakan warna garis yang berada dibawahnya.
3. Robot membutuhkan air untuk memadamkan api.
4. Robot membutuhkan sebuah *base* yang bisa digunakan untuk meletakan rangkaian elektronik, air dan motor.
5. Alat ini membutuhkan 4 *flame sensor* yang digunakan untuk mendeteksi keberadaan api di ruangan, dan digunakan untuk memastikan kondisi api melalui robot.
6. Untuk menutupi cahaya matahari agar kinerja *flame sensor* tidak terganggu, alat ini membutuhkan 3 *protoype* ruangan yang berukuran 90 cm x 60 cm.
7. Robot membutuhkan bluetooth yang digunakan untuk menerima informasi dari sistem monitoring ruangan.
8. Robot membutuhkan Arduino Mega 2560 yang digunakan untuk mengendalikan kinerja robot, dan untuk mempermudah pengkabelan maka dibutuhkan *shield* Arduino Mega 2560.

9. Robot membutuhkan 3 motor DC, 2 motor DC digunakan untuk pergerakan robot, dan 1 motor digunakan untuk memompa air dari tempat penampung air agar air bisa disemprotkan melalui penyemprot.
10. Robot membutuhkan *driver motor* yang digunakan untuk mengendalikan pergerakan motor DC yang berfungsi sebagai penggerak dan mengendalikan pergerakan motor DC yang berfungsi sebagai penyemprot.
11. Robot membutuhkan motor servo untuk menggerakan penyemprot air agar air tidak hanya tertuju pada satu titik saja.
12. Robot membutuhkan LCD 16x2 yang digunakan untuk menyampaikan informasi kepada manusia tentang status robot (*interfacing*).

3.3.2. Perancangan Sistem

Perancangan sistem dalam penelitian Rancang Robot *Line Follower* Pemadam Api Memanfaatkan *Flame Sensor* dan Bluetooth Berbasis Arduino terbagi atas beberapa subsistem yaitu *hardware, software* dan *track* Di dalam subsistem tersebut terdapat subsistem-subsistem yang lainnya sesuai dengan blok diagram dan *flowchart* yang terdapat pada Gambar 2.37, Gambar 2.38 dan Gambar 2.39. Ada beberapa kriteria sistem yang terdapat pada alat , kriteria-kriteria tersebut adalah sebagai berikut:

1. Alat ini menggunakan modul bluetooth untuk media berkomunikasi antara robot dengan sistem *monitoring* ruangan.
2. Pada alat ini, terdapat sensor api yang digunakan untuk membaca kondisi ruangan dan keberadaan api. Sensor ini akan mengirimkan sinyal ke kontroler ketika mendeteksi adanya api.
3. Robot bisa berjalan otomatis dengan menggunakan sensor cahaya.

4. Robot mempunyai tempat penyimpanan untuk menyimpan air yang akan digunakan untuk memadamkan api.
5. Robot menggunakan motor DC agar dapat bergerak dan mengalirkan air dari tempat penyimpanan air.

3.3.3. Tahap Pengujian

Pada tahap pengujian dan analisis, penulis melakukan pengujian secara bertahap. Pengujain pada tahap awal yang dilakukan oleh peneliti adalah sebagai berikut:

1. Pengujian tegangan regulator sebagai sumber tegangan. Pengujian ini bertujuan untuk mengetahui tegangan yang dihasilkan oleh rangkaian regulator.
2. Pengujian *flame sensor* sebagai sensor api yang akan digunakan untuk mendeteksi keberadaan api pengujian *flame sensor* bertujuan untuk mengetahui nilai ADC yang tepat untuk mendeteksi api yang berasal dari lilin.
3. Pengujian modul bluetooth HC-05 sebagai media komunikasi. Pengujian modul bluetooth bertujuan untuk mengetahui cara penggunaan modul bluetooth dengan menggunakan Arduino, baik cara mengirimkan data maupun menerima data.
4. Pengujian rangkaian sensor cahaya sebagai sensor pendeteksi garis.pengujian ini bertujuan untuk mengetahui apakah sensor cahaya akan memberikan perubahan nilai ADC ketika berada di atas garis putih dan garis hitam.
5. Pengujian rangkaian *driver motor DC* yang akan digunakan sebagai penggerak robot. Pengujian *driver motor DC* bertujuan untuk mengetahui apakah rangkaian *driver motor* L298N dan rangkaian *driver motor optocoupler* bisa dikendalikan dengan menggunakan Arduino Mega 2560.

6. Pengujian servo sebagai penggerak penyemprot air. Pengujian ini bertujuan untuk mengetahui apakah posisi motor servo bisa dikendalikan oleh Arduino Mega 2560.
7. Pengujian I2C LCD 16x2 dan LCD 16x2 sebagai media *interfacing* robot dengan pengguna. pengujian ini bertujuan untuk mngetahui apakah LCD bisa menampilkan karakter sesuai dengan karakter yang dikendalikan oleh Arduino Mega 2560.

Pengujian tahap akhir adalah pengujian kelayakan robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino untuk mengetahui apakah benar alat ini bisa mendeteksi keberadaan api dan memadamkannya secara otomatis, dan apakah benar robot ini bisa kembali ke tempat awal ketika sudah tidak api yang terdeteksi lagi.

3.3.4. Implementasi Sistem Perangkat Keras

Setelah melakukan pengujian maka selanjutnya adalah implementasi sistem perangkat keras, dimana perangkat keras yang akan dibuat berupa *base* robot, 3 *prorotype* ruangan yang akan menutupi cahaya matahari yang berfungsi untuk meletakan lilin, *line* berwarna hitam sebagai media gerak robot, sensor garis untuk mendeteksi garis yang kontras, *shield* Arduino untuk mempermudah pengkabelan, *driver motor* untuk mengendalikan gerakan motor yeng berfungsi sebagai penggerak robot dan penyemprot air, LCD 16x2 untuk *interfacing* robot, dan mekanik penggerak penyemprot air. Selanjutnya, perangkat keras ini digabungkan menjadi satu agar menjadi satu kesatuan sehingga robot yang dirancang bisa menjadi robot *line follower* pemadam api.

3.3.5. Implementasi Sistem Perangkat Lunak

Perangkat lunak yang akan banyak digunakan adalah Arduino IDE yang digunakan untuk menulis, verifikasi dan meng-*upload* program. Program dibuat untuk dapat mebaca nilai ADC dari *flame sensor*, untuk mengirim dan menerima data dari modul bluetooth, untuk dapat membaca nilai ADC dari sensor cahaya, mengendalikan pergerakan motor DC, mengendalikan pergerakan motor servo untuk menggerakan penyemprot, dan untuk mengendalikan karakter yang ditampilkan oleh LCD 16x2. Perangkat lunak ini akan digabungkan dengan perangkat keras agar menjadi satu sehingga bisa berfungsi sebagai robot *line follower* pemadam api.

3.4. Rancangan Penelitian

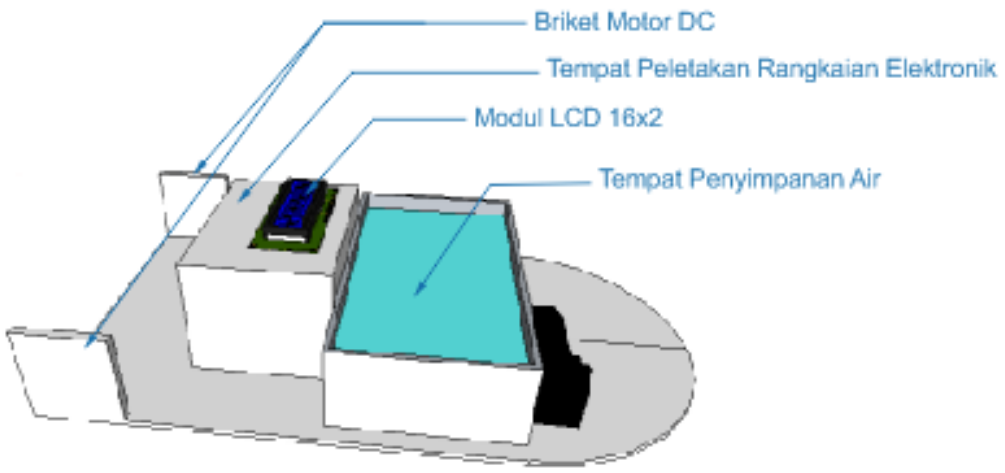
3.4.1. Perancangan Perangkat Keras Sistem

Perancangan perangkat keras menentukan keberhasilan suatu sistem secara fisik. Dalam perancangan perangkat keras dirancang agar sesuai dengan kebutuhan penilitian rancang bangun robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino.

3.4.1.1. Perancangan *Base* Robot

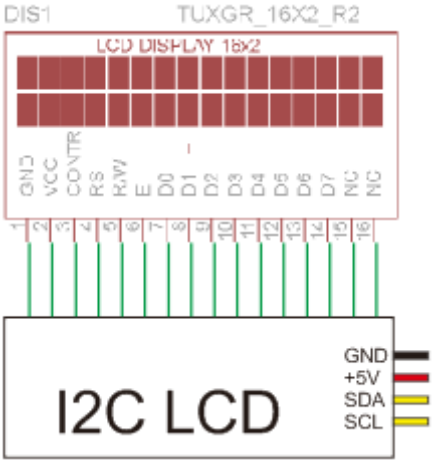
Pada penelitian ini, rancangan *base* robot yang dibuat adalah berbentuk persegi panjang dengan tambahan sebuah setengah lingkaran agar robot tidak tersangkut ketika memasuki ruangan, selain itu, ada tambahan sebuah penyangga untuk tempat meletakan motor DC. Panjang *base* adalah 35 cm dengan lebar 20 cm dan tinggi 5,5 cm. Bahan yang digunakan untuk *base* adalah akrilik berwarna hitamdengan tebal 5. Perancangan *base* bertujuan untuk membuat sebuah sasis robot yang akan digunakan sebagai tempat meletakan kontroler, sensor dan motor

driver. *Base* bagian depan digunakan untuk meletakan sensor garis dan *standing* servo, *base* bagian tengah akan digunakan sebagai tempat meletakan tempat penyimpanan air, *shiled* kontroler ,rangkaian *driver motor*, baterai, dan kontrolernya. sedangkan *base* bagian belakang digunakan untuk tempat meletakan motor DC. Gambar desain *base* robot dapat dilihat pada Gambar 3.2.

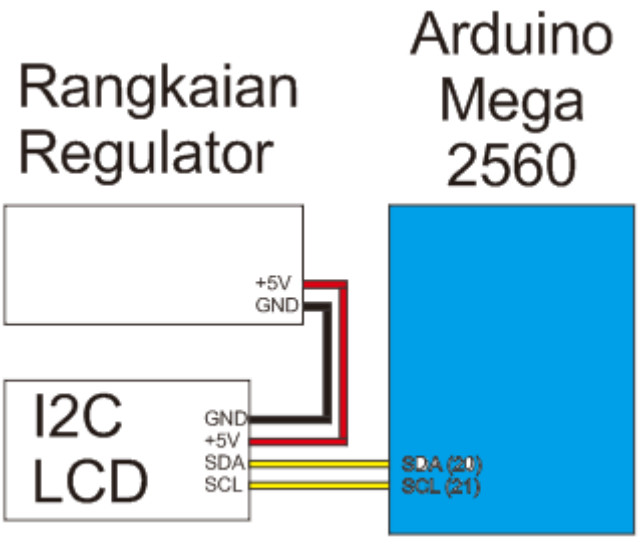


Gambar 3.2 Rancangan *Base* Robot

Pada bagian tengah *base* robot, selain digunakan untuk membuat tempat penyimpanan air, bagian ini juga digunakan untuk meletakan rangkaian-rangkaian elektronik, salah satunya adalah digunakan untuk meletakan modul LCD 16x2. Modul LCD 16x2 ini digunakan untuk melakukan antar muka dengan robot. Pengantar mukaan ini berfungsi untuk melihat status yang sedang dijalankan oleh robot. Dalam penelitian ini, digunakan pula sebuah modul I2C LCD yang akan digunakan untuk mengendalikan LCD 16x2. Penggunaan modul I2C LCD ini bertujuan untuk mengurangi penggunaan pin pada Arduino Mega 2560. Integrasi antara modul LCD 16x2 dengan modul I2C LCD dapat dilihat pada Gambar 3.3. dan integrasi antara modul I2C LCD dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.4.



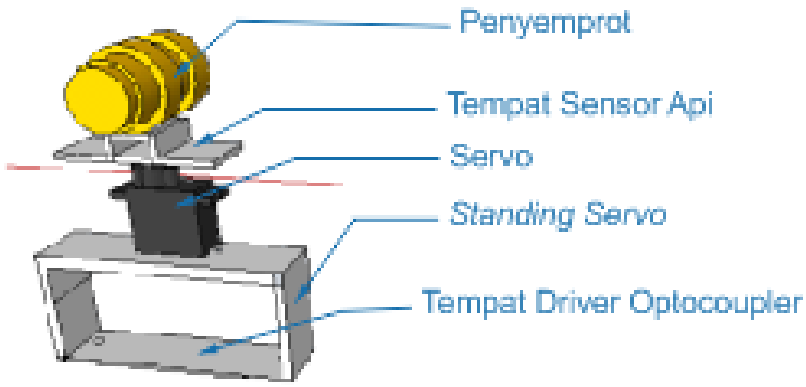
Gambar 3.3 Integrasi antara Modul LCD 16x2 dengan Modul I2C LCD



Gambar 3.4 Integrasi antara Modul I2C LCD dengan Arduino Mega 2560 dan Rangkaian Regulator

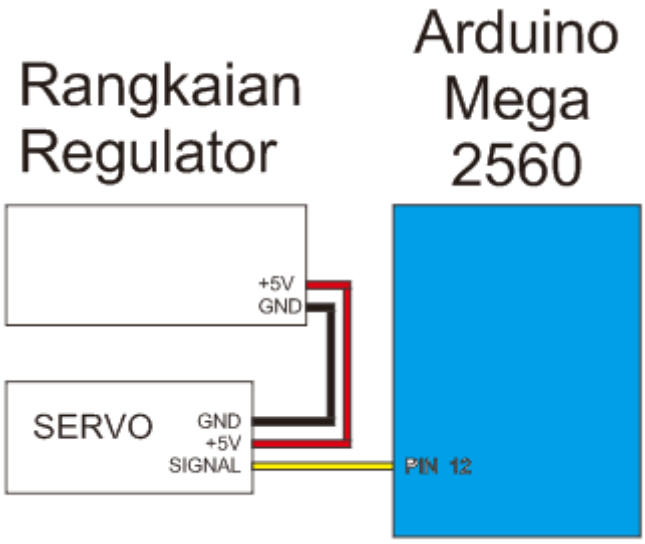
3.4.1.2. Perancangan *Standing* sebagai Tempat Servo

Pada penelitian ini, rancangan *standing* robot yang dibuat adalah berbentuk persegi dengan tambahan sebuah penyangga seperti *base* robot. Panjang standing adalah 8 cm dengan lebar 2 cm dan tinggi sekitar 4 cm. Bahan yang digunakan untuk *standing* adalah akrilik berwarna hitam dengan tebal 5 mm. Gambar desain *standing* dapat dilihat pada Gambar 3.5.

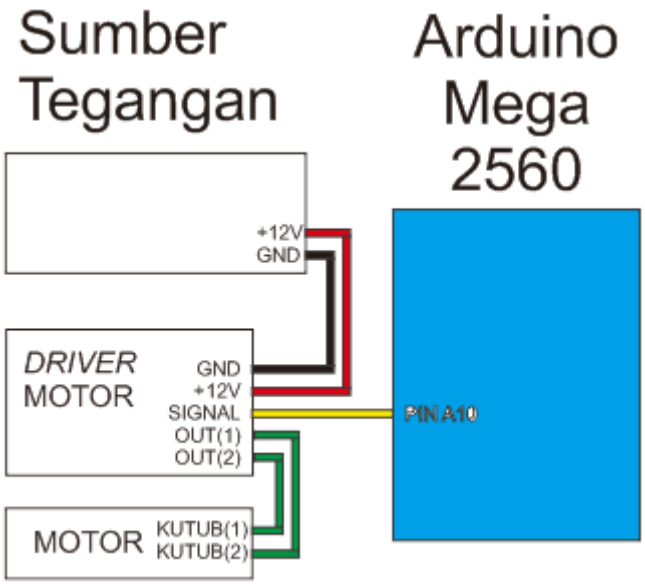


Gambar 3.5 Rancangan *Standing* Robot

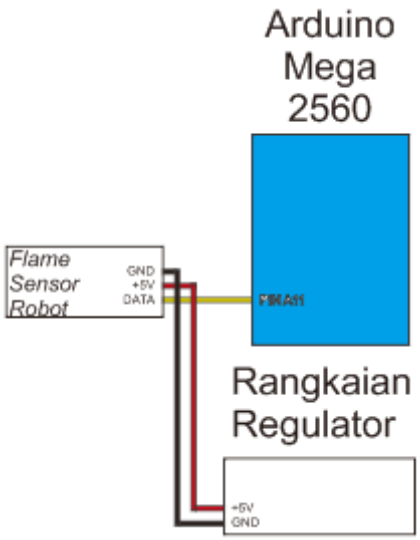
Tujuan pembuatan *standing* ini adalah untuk membuat tempat supaya servo bisa berdiri dengan tegak dan tidak jatuh ketika robot bergerak. Selain sebagai tempat servo, *standing* ini juga digunakan untuk meletakan rangkaian *driver motor* dengan menggunakan *optocoupler*. Di bagian atas juga dogunakan untuk meletakan sensor api dan penyemprot air. Integrasi antara motor servo dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.6, integrasi antara rangkaian *driver motor optocoupler* dengan Arduino Mega dan rangkaian regulator dapat dilihat pada Gambar 3.7, dan integrasi antara *flame sensor* dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.8.



Gambar 3.6 Integrasi antara Motor Servo dengan Arduino Mega 2560 dan Rangkaian Regulator



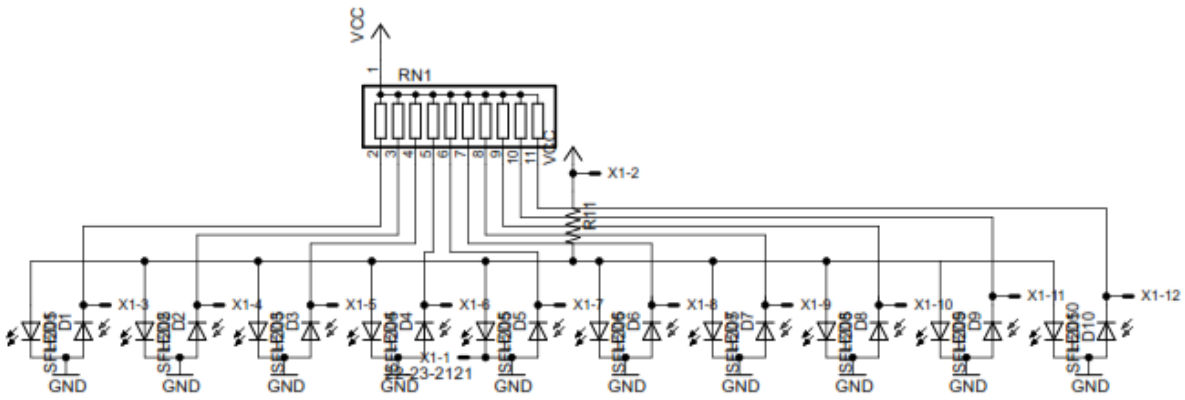
Gambar 3.7 Integrasi antara Rangkaian *Driver motor Optocoupler* dengan Arduino Mega dan Sumber Tegangan



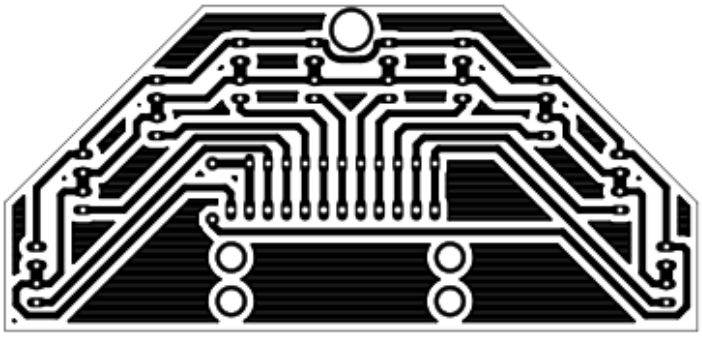
Gambar 3.8 Integrasi antara *Flame Sensor* dengan Arduino Mega 2560 dan Rangkaian Regulator

3.4.1.3. Perancangan Rangkaian Sensor Garis Robot

Pada perangcangan dan pembuatan rangkaian sensor garis yang akan digunakan sebagai bagian *input* robot penulis menggunakan 10 buah sensor cahaya yang terpasang membentuk huruf "V". Dengan jumlah sensor cahaya yang banyak diharapkan akan meminimalisir terjadinya *sort-cut* (robot keluar jalur). Gambar skema rangkaian sensor garis yang digunakan bisa dilihat pada Gambar 3.9. dan Gambar *layout* sensor garis yang digunakan bisa dilihat pada Gambar 3.10.

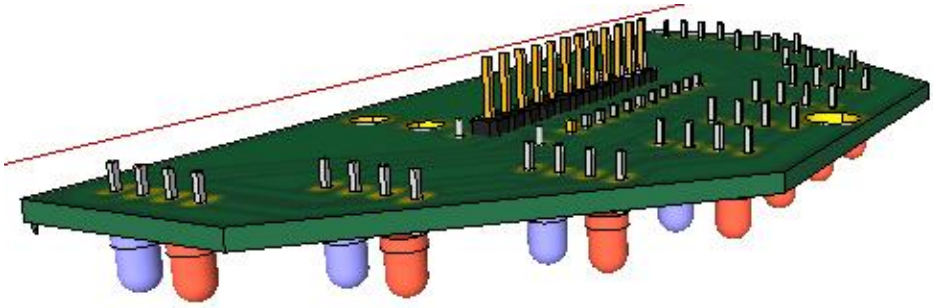


Gambar 3.9 Skema Rangkaian Sensor Garis Robot



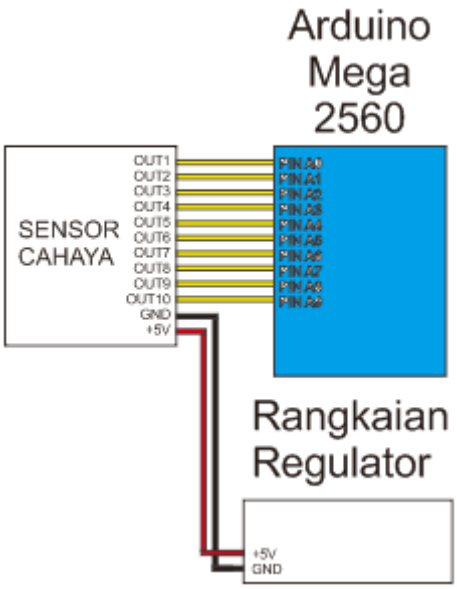
Gambar 3.10 *Layout* Sensor Garis Robot

Jarak antara masing-masing sensor cahaya adalah 1cm, sedangkan jarak antara LED dengan photodioda adalah 5mm. Pemasangan 4 sensor secara sejajar di depan bertujuan agar robot bisa bergerak secara stabil di *track* lurus, pemasangan 4 sensor lainnya secara menurun digunakan untuk pada saat robot menghadapi *track* yang berbelok dan meletakan 2 sensor sisanya di bagian belakang bertujuan agar ketika menghadapi sebuah pertigaan atau perempatan robot tidak berbelok ketika mengenai sensor bagian tengah, maka robot masih bisa berbelok dengan memaanfaatkan kedua sensor tersebut. Bisa dianggap 2 sensor di belakang ini berfungsi sebagai pengaman ketika robot mengalami gangguan. Desain sensor garis dapat dilihat pada Gambar 3.11.



Gambar 3.11 Desain Rangkaian Sensor Garis

Integrasi antara rangkaian sensor garis dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.12.

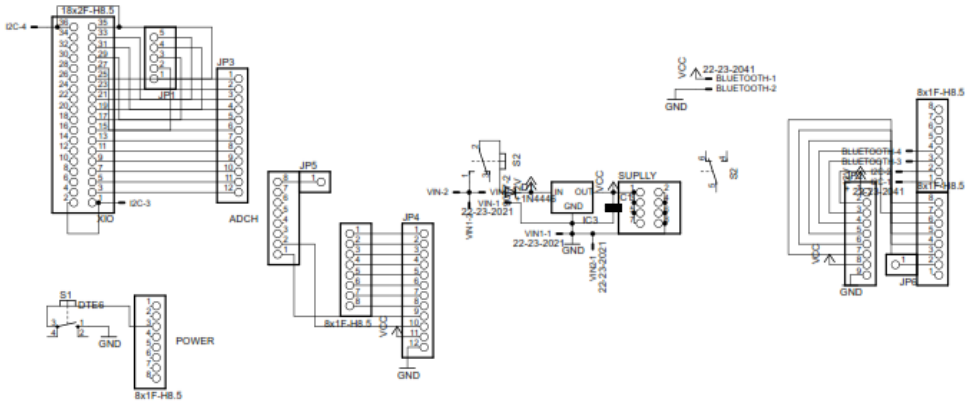


Gambar 3.12 Integrasi Rangkaian Sensor Garis dengan Arduino Mega 2560 dan Rangkaian Regulator

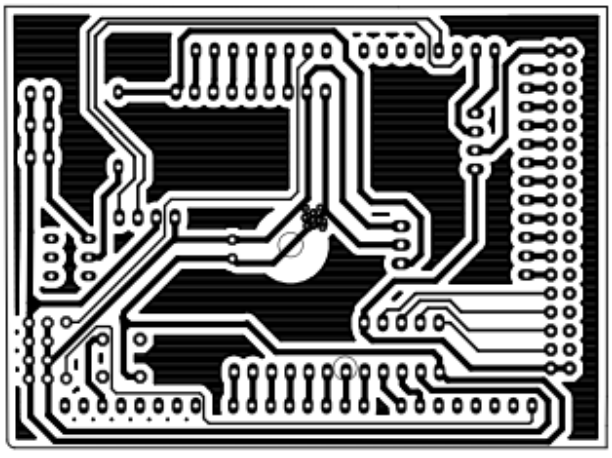
3.4.1.4. Perancangan *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805

Pada perangcangan dan pembuatan rangkaian *Shield* Arduino Mega 2560 dan rangkaian regulator 7805 yang akan digunakan sebagai bagian pengatur atau pengendali gerakan robot. Tujuan pembuatan *shield* ini adalah agar mudah mengubungkan komponen-komponen *input* dan komponen-komponen *output*

dengan rangkaian proses, dan tujuan pembuatan rangkaian regulator adalah sebagai sumber tegangan 5 VDC untuk rangkaian lainnya. Dengan adanya *shield*, maka peneliti sudah memberikan tempat-tempat khusus bagi modul-modul *input* dan modul *output*, sehingga peneliti hanya tinggal mengubungkan modul-modul yang sudah ada dengan Arduino Mega 2560 melalui tempat-tempat khusus yang sudah disediakan. Gambar skema rangkaian *shield* Arduino Mega 2560 yang digunakan bisa dilihat pada Gambar 3.13. dan Gambar *layout shield* Arduino Mega 2560 yang digunakan bisa dilihat pada Gambar 3.14.

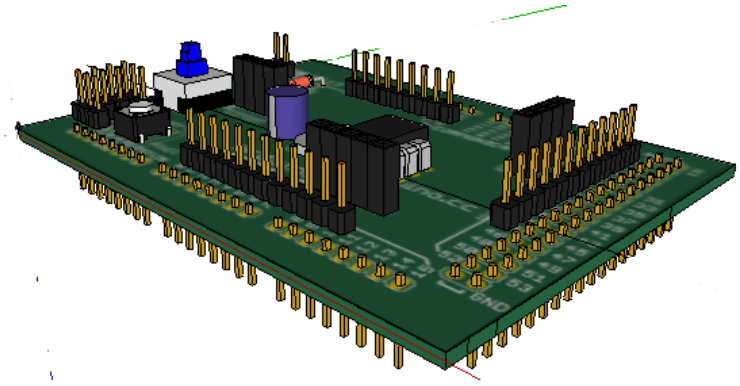


Gambar 3.13 Skema Rangkaian *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805



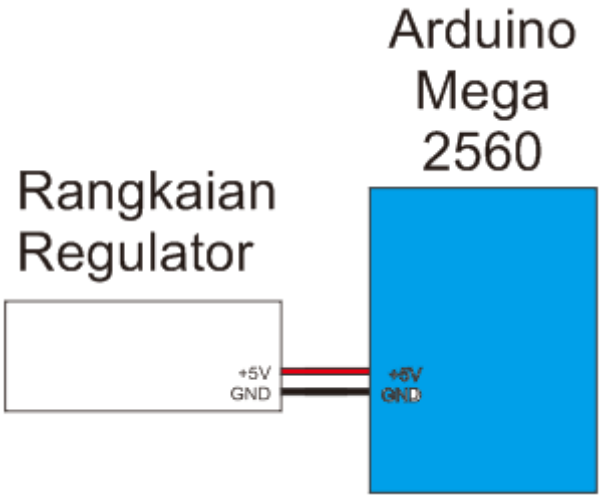
Gambar 3.14 *Layout Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805

Pemasangan beberapa komponen yang berada di bawah rangkaian *push button* seperti rangkaian regulator dan pin I2C LCD (*Liquid Crystal Display*) bertujuan agar memaksimalkan lebar PCB yang sesuai dengan Arduino Mega 2560, dan peletakan komponen yang berada di dalam *shield* bertujuan agar membuat sekecil mungkin rangkaian *shield* yang akan digunakan. Pemasangan pin *output* di belakang yang digunakan untuk rangkaian *driver motor* bertujuan agar mempermudah mengubungkan rangkaian *shield* Arduino Mega 2560 dengan rangkaian *driver motor* yang berada di belakang rangkaian *shield*. Sedangkan pemasangan pin *input* di depan yang digunakan untuk rangkaian sensor garis bertujuan agar dapat mempermudah mengubungkan rangkaian *shield* Arduino Mega 2560 dengan rangkaian sensor garis yang berada di depannya. Desain rangkaian *shield* Arduino Mega 2560 dapat dilihat pada Gambar 3.15.



Gambar 3.15 Desain Rangkaian *Shield* Arduino Mega 2560 dan Rangkaian Regulator 7805

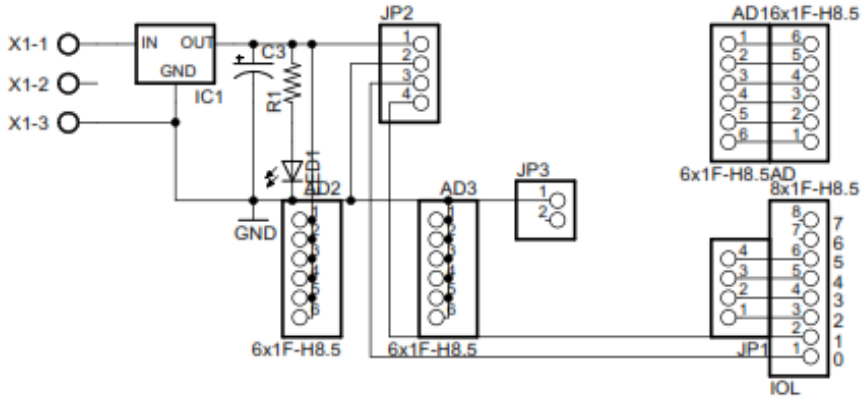
Integrasi antara rangkaian regulator dengan Arduino Mega 2560 dapat dilihat pada Gambar 3.16.



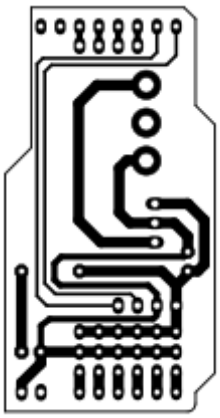
Gambar 3.16 Integrasi antara Rangkaian Regulator dengan Arduino Mega 2560

3.4.1.5. Perancangan *Shield* Uno dan Rangkaian Regulator 7805

Pada perangcangan dan pembuatan rangkaian *Shield* Uno dan rangkaian regulator 7805 yang akan digunakan sebagai bagian pengatur atau pengendali monitoring ruangan. Tujuan pembuatan *shield* ini adalah agar mudah mengubungkan komponen-komponen *input* dan komponen-komponen *output* dengan rangkaian proses, dan tujuan pembuatan rangkaian regulator adalah sebagai sumber tegangan 5 VDC untuk rangkaian lainnya. Dengan adanya *shield*, maka peneliti sudah memberikan tempat-tempat khusus bagi modul-modul *input* dan modul *output*, sehingga peneliti hanya tinggal mengubungkan modul-modul yang sudah ada dengan Arduino Uno melalui tempat-tempat khusus yang sudah disediakan. Gambar skema rangkaian *shield* Arduino Uno yang digunakan bisa dilihat pada Gambar 3.17. dan Gambar *layout shield* Uno yang digunakan bisa dilihat pada Gambar 3.18.



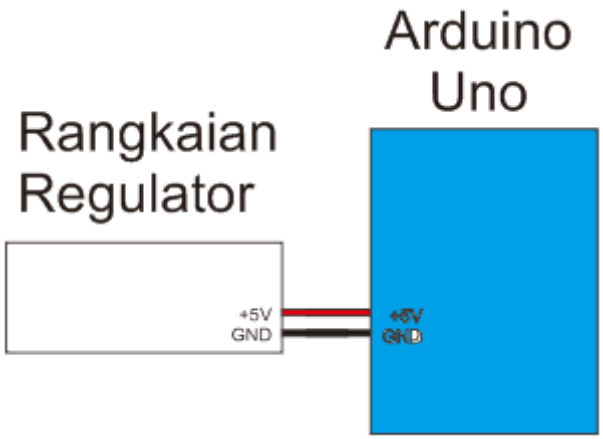
Gambar 3.17 Skema Rangkaian *Shield* Arduino Uno dan Rangkaian Regulator 7805



Gambar 3.18 *Layout Shield* Arduino Uno dan Rangkaian Regulator 7805

Peletakan pin *input* dilakukan berjajar bertujuan agar mempermudah integrasi antara *shield* Arduino Uno dan modul *flame sensor.*

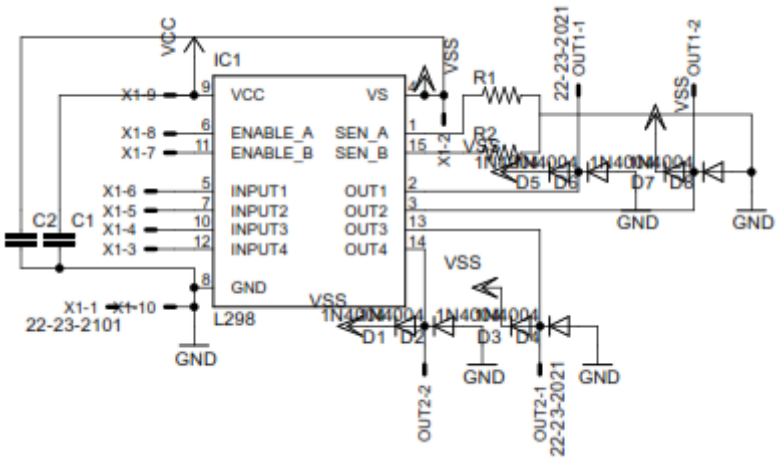
Integrasi antara rangkaian regulator dengan Arduino Uno dapat dilihat pada Gambar 3.19.



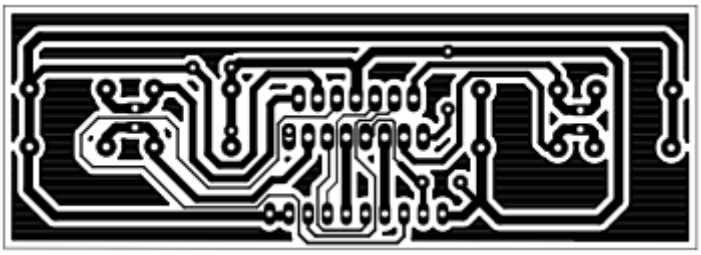
Gambar 3.19 Integrasi antara Rangkaian Regulator dengan Arduino Uno

3.4.1.6. Perancangan Rangkaian *Driver motor* dengan Menggunakan IC L298N

Pada perangcangan dan pembuatan rangkaian *driver motor* dengan menggunakan IC L298N yang akan digunakan sebagai bagian pengatur gerakan motor DC yang akan bertindak sebagai penggerak roda yang berfungsi sebagai penggerak. Pemilihan komponen IC L298N ini dikarenakan kendali motor DC bukan hanya berupa beputar atau tidak, melainkan kecepataan putaran motornya harus dikontrol. Selain itu, IC jenis ini bisa mengalirkan arus 2 *ampere* sehingga sangat sesuai dengan robot yang hanya menggunakan batrai 1 *ampere*. Dengan desain yang digunakan penulis, *driver* ini hanya membutuhkan *input* nilai PWM (*Pulse Width Modulation*) langsung dari pengendali. Hal ini disebabkan oleh pin *enable* IC sudah dihubungkan dengan VCC sehingga akan selalu berlogika *high*. Gambar skema rangkaian *driver motor* dengan menggunakan IC L298N yang digunakan bisa dilihat pada Gambar 3.20. dan Gambar *layout driver motor* dengan menggunakan IC L298N yang digunakan bisa dilihat pada Gambar 3.21.

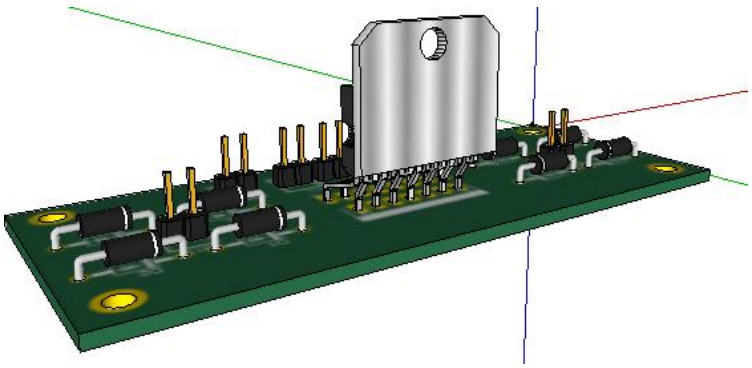


Gambar 3.20 Skema Rangkaian *Driver motor* dengan Menggunakan IC L298N



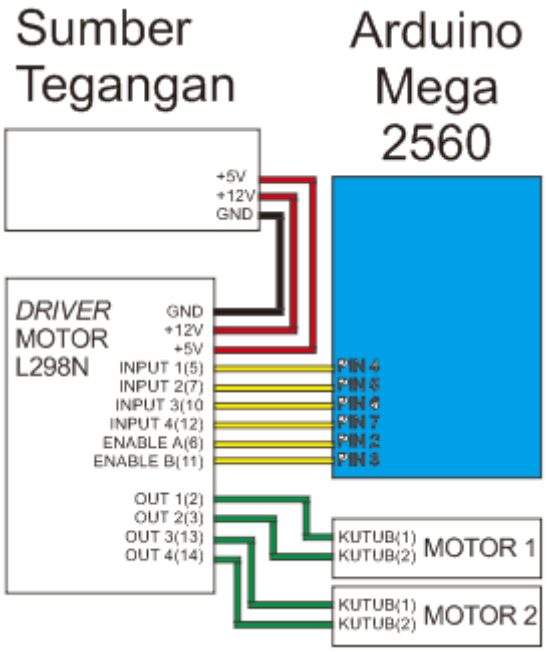
Gambar 3.21 *Layout Driver motor* dengan Menggunakan IC L298N

Pemasangan pin *output* pada sisi kanan dan kiri bertujuan agar mempermudah mengubungkan rangkaian *driver* dengan motor DC yang berada pada sisi kanan dan kiri robot. Sedangkan pemasangan pin *input* pada sisi tengah bertujuan agar dapat mempermudah mengubungkan rangkaian *driver* dengan rangkaian pengendali yang berada di depannya. Desain rangkaian *driver motor* dengan menggunakan IC L298N dapat dilihat pada Gambar 3.22.



Gambar 3.22 Desain Rangkaian *Driver motor* dengan Menggunakan IC L298N

Integrasi antara rangkaian *Driver motor* dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.23.

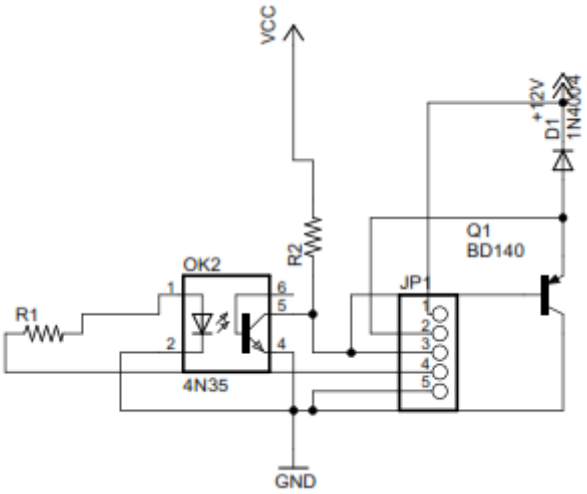


Gambar 3.23 Integrasi antara rangkaian *Driver motor* dengan Arduino Mega 2560 dan rangkaian regulator

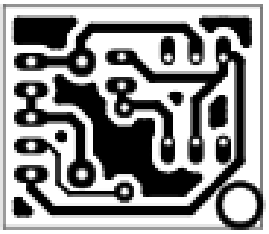
3.4.1.7. Perancangan Rangkaian *Driver Motor* dengan Menggunakan *Optocoupler*

Pada perangcangan dan pembuatan rangkaian *driver motor* dengan menggunakan *optocoupler* yang akan digunakan sebagai bagian pengatur gerakan motor DC yang akan bertindak sebagai pemadam api. Pemilihan komponen *optocoupler* ini dikarenakan kendali motor DC hanya berupa beputar atau tidak. Jika kita ingin mengaktifkan *optocoupler*, maka kita hanya perlu mengaktifkan led yang berada di dalam *optocoupler*nya saja, sehingga tidak membutuhkan arus dan tegangan yang besar untuk mengaktifkan *optocoupler*. Selain itu, keuntungan menggunakan *optocoupler* adalah ketika terjadi kerusakan pada motor DC atau

adanya arus balik dari motor DC maka yang akan rusak adalah transistor yang berada di dalam *optocoupler* saja, sehingga mikrokontroler akan dalam kondisi yang aman. Di dalam rangkaian ini juga terdapat sebuah led yang akan digunakkan sebagai aktif atau tidaknya *optocoupler*, dan ada *pinhead* yang akan digunakan sebagai penghubung dengan rangkaian yang lain. Gambar skema rangkaian *driver motor* dengan menggunakan *optocoupler* yang digunakan bisa dilihat pada Gambar 3.24. dan Gambar *layout driver motor* dengan menggunakan *optocoupler* yang digunakan bisa dilihat pada Gambar 3.25.



Gambar 3.24 Skema Rangkaian *Driver motor* dengan Menggunakan *Optocoupler*

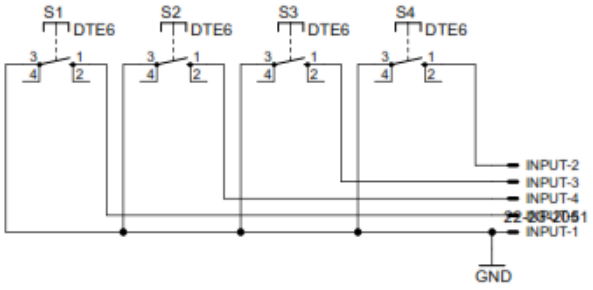


Gambar 3.25 *Layout Driver motor* dengan Menggunakan *Optocoupler*

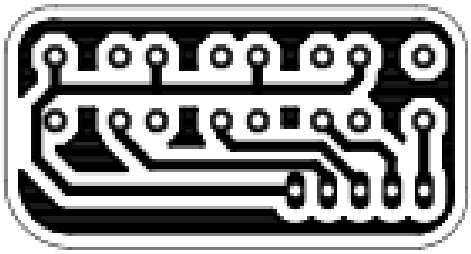
Pemasangan komponen-komponen yang digunakan dilakukan secara merapat yang bertujuan agar papan PCB yang digunakan tidak terlalu besar sehinnga mudah untuk ditempatkan meskipun di tempat yang sempit.

3.4.1.8. Perancangan Rangkaian *Push Button*

Pada perangcangan dan pembuatan rangkaian *push button* akan digunakan sebagai bagian pemberi masukan terhadap rangkaian *shield* Arduino Mega 2560. Peletakan rangkaian yang dilakukan secara terpisah dengan *shield* bertujuan agar rangkaian ini tidak memakan banyak tempat pada *shield*. Gambar skema rangkaian *push button* yang digunakan bisa dilihat pada Gambar 3.26. dan Gambar *layout* rangkaian *push button* yang digunakan bisa dilihat pada Gambar 3.27.



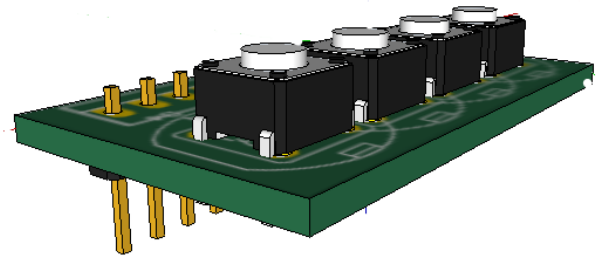
Gambar 3.26 Skema Rangkaian *Push Button*



Gambar 3.27 *Layout* Rangkaian *Push Button*

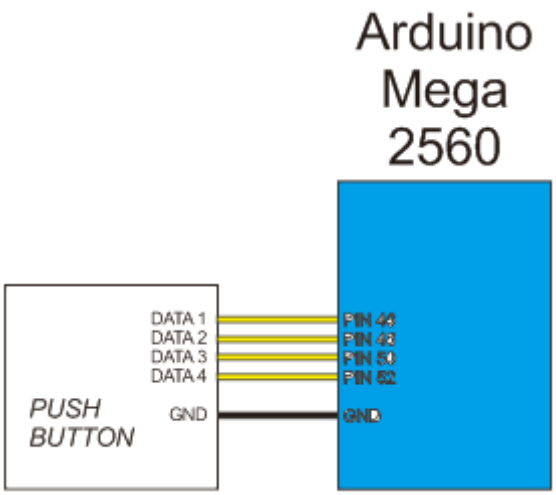
Pemasangan *push button* secara sejajar bertujuan agar peneliti tidak mengalami kesusahan dalam mengatur sebuah parameter di dalam program. Dan pemasangan

pin header secara terbalik bertujuan agar memudahkan menghubungkan rangkaian *push button* dengan rangkaian *shield*. Desain rangkaian *push button* dapat dilihat pada Gambar 3.28.



Gambar 3.28 Desain Rangkaian *Push Button*

Integrasi antara rangkaian *Push Button* dengan Arduino Mega 2560 dapat dilihat pada Gambar 3.29.

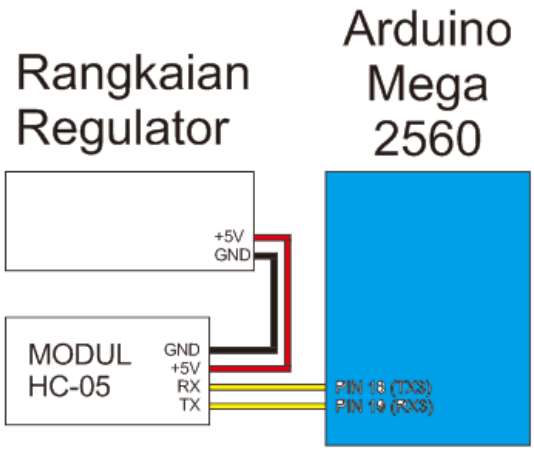


Gambar 3.29 Integrasi antara rangkaian *Push Button* dengan Arduino Mega 2560

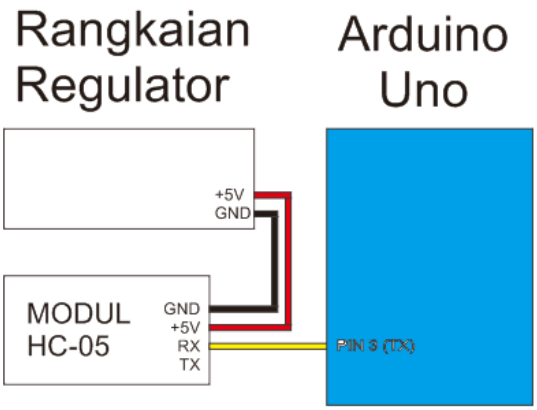
3.4.1.9. Perancangan Integrasi Modul Bluetooth HC-05 dengan Arduino

Pada perancangan modul bluetooth, tipe modul bluetooth yang digunakan adalah modul bluetooth HC-05. Pemilihan modul tersebut karena modul tersebut bisa digunakan dalam dua mode, bisa sebagai mode *master* dan bisa sebagai mode

slave. Mode *master* akan digunakan sebagai pengirim data dari ruangan ke robot dan terintegrasi dengan Arduino Uno. Sedangkan Mode *slave* akan digunakan sebagai pengirim data dari robot ke ruangan dan terintegrasi dengan Arduino Mega 2560. Integrasi antara modul bluetooth HC-05 dengan Arduino Mega 2560 dan rangkaian regulator dapat dilihat pada Gambar 3.30 dan Integrasi antara modul bluetooth HC-05 dengan Arduino Uno dan rangkaian regulator dapat dilihat pada Gambar 3.31.



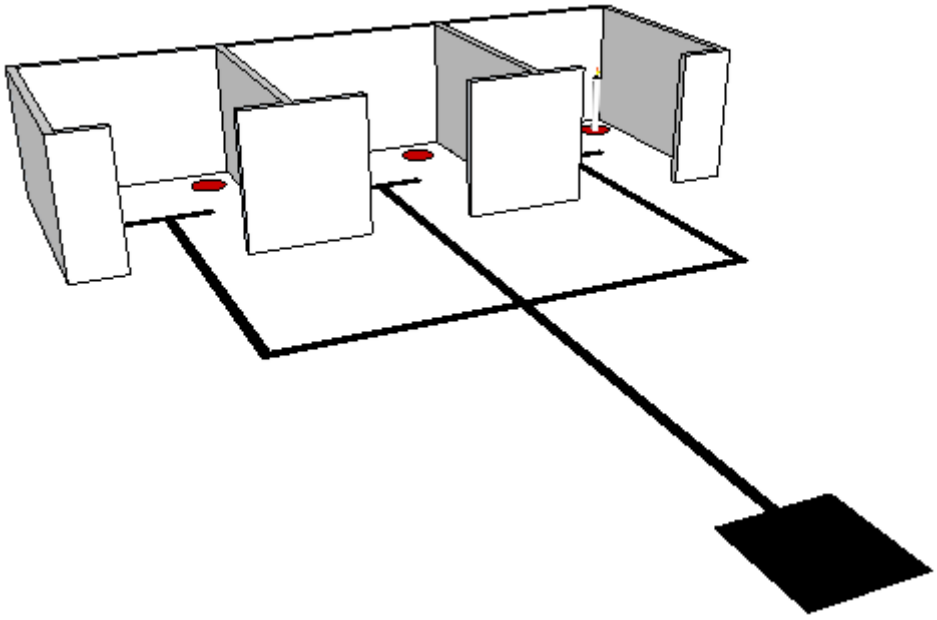
Gambar 3.30 Integrasi antara Modul Bluetooth HC-05 dengan Arduino Mega 2560 dan Rangkaian Regulator



Gambar 3.31 Integrasi antara Modul Bluetooth Hc-05 dengan Arduino Uno dan Rangkaian Regulator

3.4.1.10.Perancangan Jalur(*Track*) dan Ruangan

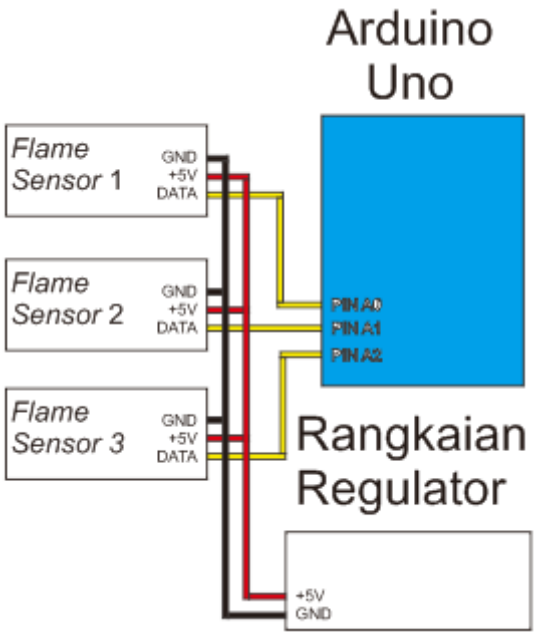
Perancangan jalur yang digunakan pada penelitian ini adalah sebuah garis berwarna hitam dengan lebar garis 2 cm. Bentuk garis yang akan digunakan adalah seperti bentuk garpu dengan titik *start* berbentuk kotak dengan ukuran 20 cm x 20 cm. Dan bentuk garis pada area titik api berupa pertigaan dengan panjang garis akhir 20 cm. Semenntara itu perancangan ruangan yang akan digunakan di dalam penelitian ini adalah ruangan dengan bentuk persegi yang terdapat pada setiap area titik api. Jumlah ruangan yang akan digunakan adalahh tiga ruangan dengan posisi peletakan apinya ada dua tempat di dalam setiap ruangan Gambar desain jalur dan ruangan yang akan digunakan dalam penelitian ini dapat dilihat pada Gambar 3.32.



Gambar 3.32 Desain Jalur dan Ruangan yang digunakan untuk Penelitian

Disetiap ruangan terdapat sebuah *flame sensor* yang akan digunakan untuk memeriksa kondisi ruangan apakah ada api atau tidak. Semua *flame sensor* yang terdapat pada ruangan terhubung dengan Arduino Uno, integrasi antara *flame*

sensor dengan Arduino Uno dan Rangkaian Regulator dapat dilihat pada Gambar 3.33.



Gambar 3.33 Integrasi antara *Flame Sensor d*engan Arduino Uno dan Rangkaian Regulator

3.4.2. Perancangan Perangkat Lunak Sistem

Perancangan perangkat lunak adalah merancang sekumpulan data elektronik berupa program atau instruksi yang akan menjalankan suatu perintah. Perancangan perangkat lunak pada penelitian ini adalah dengan menggunakan *software* Arduino IDE yang akan digunakan untuk memprogram Arduino Mega 2560 dan Arduino Uno. *Software* ini dipilih karena bahasa pemrograman yang digunakan mudah dipahami oleh penneliti dan dengan demikian diharapkan dapat mempermudah untuk merancang dan menuliskan program agar sistem bisa bekerja seperti yang diinginkan. Dalam pembuatan robot *line follower* pemadam api dan sistem *monitoring* ruangan digunakan beberapa *input* dan *output*.

Perancangan perangkat lunak membutuhkan beberapa parameter data yang diperlukan untuk pemrograman, tabel 3.1 dan tabel 3.2 adalah beberapa parameter data yang digunakan pada Arduino Uno dan Arduino Mega 2560.

Tabel 3.1 Pin I/O Arduino Mega 2560 yang digunakan

Pin Arduino Mega 2560	Penggunaan
A0-A9	Masukan sensor garis (sensor 1-senssor 10)
A11	Masukan sensor api pada robot
46	Masukan data *push button* 1
48	Masukan data *push button* 2
50	Masukan data *push button* 3
52	Masukan data *push button* 4
2	Pengatur kecepatan motor kanan
3	Pengatur kecepatan motor kiri
4	Pengatur gerakan motor kanan untuk membuat robot bergerak maju
5	Pengatur gerakan motor kanan untuk membuat robot bergerak mundur
6	Pengatur gerakan motor kiri untuk membuat robot bergerak maju
7	Pengatur gerakan motor kiri untuk membuat robot bergerak mundur
A10	Pengatur gerakan motor DC yang bertugas sebagai pemadam
12	Mengatur sudut servo
SDA (Pin 20)	Serial Data
SCL (Pin 21)	Serial *Clock*
TX 3 (Pin 18)	Mengirim data serial
RX 3 (Pin 19)	Menerima data serial

Tabel 3.2 Pin I/O Arduino Uno yang digunakan

Pin Arduino Uno	Penggunaan
A0	Masukan sensor api pada ruangan 1
A1	Masukan sensor api pada ruangan 2
A2	Masukan sensor api pada ruangan 3
TX (Pin 3)	Mengirim data serial

3.5. Teknik dan Prosedur Pengumpulan Data

Prosedur pengumpulan data yang ditempuh dalam penelitian dalam merancang, membangun dan menguji robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino terdiri dari beberapa tahap, di antaranya adalah:

1. Pembuatan robot dimulai dari tahap pemilihan komponen dan modul yang tepat untuk membuat robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino.
2. Pengujian *flame sensor* KY-026 secara sederhana dengan menggunakan program ADC untuk mengecek apakah *flame sensor* KY-026 dapat digunakan atau tidak.
3. Pengujian modul bluetooth HC-05 secara sederhana dengan menggunakan program serial komunikasi untuk mengecek apakah modul bluetooth HC-05 dapat digunakan atau tidak.
4. Pengujian rangkaian sensor garis secara sederhana dengan menggunakan program ADC untuk mengecek apakah rangkaian sensor garis dapat digunakan atau tidak.
5. Pengujian modul I2C LCD dan modul LCD 16x2 secara sederhana dengan menggunakan program untuk mencetak karakter pada LCD untuk mengecek apakah modul I2C LCD dan modul LCD dapat digunakan atau tidak.
6. Pengujian rangkaian *driver motor* L298N secara sederhana dengan menggunakan program *analog output* dan *digital output* untuk mengecek apakah rangkaian *driver motor* L298N dapat digunakan atau tidak.

7. Pengujian servo secara sederhana dengan menggunakan program *sweep-servo* untuk mengecek apakah rangkaian servo dapat digunakan atau tidak.
8. Pengujian rangkaian *driver motor optocoupler* secara sederhana dengan menggunakan program *digital output* untuk mengecek apakah rangkaian servo dapat digunakan atau tidak.
9. Tahap berikutnya adalah tahap pemasangan, perakitan dan pengkabelan komponen dan modul yang sudah di pilih.
10. Tahap teakhir adalah tahap pengujian rangkaian dan modul yang akan digunakan lalu tahap pengujian kelayakan alat.

3.6. Teknik Analisa Data

Teknik analisis data adalah kriteria pengujian untuk mendapatkan data yang diperlukan pada seluruh subsistem yang ada pada alat. Kriteria pengujian dilaksanakan peneliti untuk menyatakan bahwa sistem atau alat yang telah dibuat dinyatakan berhasil atau gagal. Berikut ini merupakan kriteria pengujian pada robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino.

3.6.1. Pengujian *Flame Sensor* KY-026 di Dalam Ruangan dan di Robot

Pengujian *flame sensor* dilakukan dengan cara menguji masing-masing sensor. *Output* sensor dihubungkan dengan pin ADC (*Analog to Digital Converter*) pada Arduino. Besar tegangan yang digunakan untuk mengaktifakan sensor adalah 5 VDC. Tahap pertama pengujian *flame sensor* adalah dengan meletakan *flame sensor* tidak di depan api, lalu setelah itu ukurlah nilai tegangan keluarannya dengan menggunakan multitester dan catatlah hasilnya pada tabel 3.3 pada kolom "Tegangan Pada Daerah Bebas Api". Setelah itu bukalah serial monitor pada

software Arduino IDE dan catatlah hasil pembacaan nilai analog di setiap *flame sensor* pada tabel 3.3 pada kolom "Nilai Analog Pada Daerah Bebas Api".

Tahap kedua pengujian *flame sensor* adalah dengan meletakan *flame sensor* di depan api, lalu setelah itu ukurlah nilai tegangan keluarannya dengan menggunakan multitester dan catatlah hasilnya pada tabel 3.3 pada kolom "Tegangan Pada Daerah Api". Setelah itu bukalah serial monitor pada *software* Arduino IDE dan catatlah hasil pembacaan nilai analog di setiap sensor pada tabel 3.3 pada kolom "Nilai Analog Pada Daerah Api".

Tahap terakhir pengujain modul *flame sensor* adalah dengan melihat perubahan nilai tegangan dan nilai analog yang terjadi ketika jarak antara *flame sensor* dan api semakin bertambah jauh sebesar 1cm, pengujian ini bertujuan agar peneliti bisa mengetahui data nilai analog yang dibaca oleh *flame sensor* pada jarak-jarak tertentu. Tabel 3.3 merupakan tabel pengukuran *flame sensor* yang ada di dalam ruangan. Tabel 3.4 merupakan tabel pengukuran *flame sensor* yang ada di robot.

Tabel 3.3 Tabel Pengukuran Nilai Analog pada *Flame Sensor* di Dalam Ruangan

No.	Label	Kriteria Pengujian	Nilai Analog		Tampilan	
			Pada Daerah Bebas Api	Pada Daerah Api	Pada Daerah Bebas Api	Pada Daerah Api
1.	*Flame sensor* 1	Nilai analog dan tegangan keluaran sensor saat mendetaksi api lebih kecil dibandingkan dengan saat tidak ada api.	.....	.....	.....	.....
2.	*Flame sensor* 2		.....	.....	.....	.....
3.	*Flame sensor* 3		.....	.....	.....	.....

Tabel 3.4 Tabel Pengukuran Nilai Analog pada *Flame Sensor* di Robot

No.	Jarak	Kriteria Pengujian	Hasil Pengukuran Nilai Analog	Tampilan
1.	1 cm	Nilai analog dan tegangan keluaran yang dihasilkan sensor akan semakin kecil ketika jarak antara api dengan sensor semakin kecil.	.....	.....
2.	5 cm		.....	.....
3.	10 cm		.....	.....
4.	15 cm		.....	.....
5.	20 cm		.....	.....

3.6.2. Pengujian Komunikasi Serial Bluetooth

Pengujian komunikasi serial bluetooth dilakukan dengaan menguji komunikasi serial antara ruangan yang akan dideteksi keberadaan apinya dengan robot *line follower* pemadam api. Hasil pembacaan kondisi ruangan akan ditampilkan pada serial monitor Arduino IDE sebagai indikator keberhasilan komunikasi antara dua Arduino melalui komunikasi serial bluetooth. Tabel pengujian komunikasi serial bluetooth dapat dilihat pada tabel 3.5 berikut ini:

Tabel 3.5 Kriteria Pengujian Komunikasi Serial Dua Bluetooth

No.	Pengujian arah komunikasi	Data yang Akan dikrirm	Kriteria pengujian	Data yang terecetak di serial monitor Arduino IDE	Tampilan
1.	Pengiriman dan peneriamaan data dari *Transimiter bluetooth master* kepada *bluetooth slave*	Data terikirm dari Arduino 2 berupa *string* "0" dan diterima oleh Arduino 1.	Data yang diterima oleh *slave* akan sama dengan data yang dikirim oleh *master*.	.....	.....
2.		Data terikirm dari Arduino 2 berupa *string* "1" dan diterima oleh Arduino 1.		.....	.....
3.		Data terikirm dari Arduino 2 berupa *string* "2" dan diterima oleh Arduino 1.		.....	.....
4.		Data terikirm dari Arduino 2 berupa *string* "3" dan diterima oleh Arduino 1.		.....	.....

3.6.3. Pengujian Sensor Garis

Pengujian sensor garis dilakukan dengan cara menguji masing-masing sensor. *Output* sensor dihubungkan dengan pin ADC (*Analog to Digital Converter*) pada Arduino. Besar tegangan yang digunakan untuk mengaktifkan sensor adalah 5 VDC. Sensor yang terdapat pada ujung kiri adalah sensor dengan label sensor 1, dan sensor yang terdapat pada ujung kanan adalah sensor dengan label sensor 10. Tahap pertama pengujian sensor adalah dengan meletakan sensor garis di atas alas yang berwarna putih, lalu setelah itu ukurlah nilai tegangan keluarannya dengan menggunakan multitester dan catatlah hasilnya pada tabel 3.6 pada kolom "Tegangan Pada Alas Putih". Setelah itu bukalah serial monitor pada *software* Arduino IDE dan catatlah hasil pembacaan nilai analog di setiap sensor pada tabel 3.6 pada kolom "Nilai Analog Pada Alas Putih".

Setalah data pada alas putih sudah didapatkan, maka tahap kedua pengujian sensor adalah dengan meletakan sensor garis di atas alas yang berwarna hitam, lalu setelah itu ukurlah nilai tegangan keluarannya dengan menggunakan multitester dan catatlah hasilnya pada tabel 3.6 pada kolom "Tegangan Pada Alas Hitam". Setelah itu bukalah serial monitor pada *software* Arduino IDE dan catatlah hasil pembacaan nilai analog di setiap sensor pada tabel 3.6 pada kolom "Nilai Analog Pada Alas Hitam". Tabel 3.6 berikut merupakan tabel pengukuran sensor garis.

Tabel 3.6 Tabel Pengujian dan Pengukuran Sensor Garis

No.	Label	Kriteria Pengujian	Tegangan (V)		Nilai Analog	
			Alas Putih	Alas Hitam	Alas Putih	Alas Hitam
1.	Sensor 1	Nilai analog dan tegangan keluaran yang dihasilkan sensor saat berada di atas garis hitam akan lebih besar dibandingkan pada saat berada di garis putih	.....	.....	.....	.....
2.	Sensor 2		.....	.....	.....	.....
3.	Sensor 3		.....	.....	.....	.....
4.	Sensor 4		.....	.....	.....	.....
5.	Sensor 5		.....	.....	.....	.....
6.	Sensor 6		.....	.....	.....	.....
7.	Sensor 7		.....	.....	.....	.....
8.	Sensor 8		.....	.....	.....	.....
9.	Sensor 9		.....	.....	.....	.....
10.	Sensor 10		.....	.....	.....	.....

3.6.4. Pengujian Modul I2C LCD dan Modul LCD 16x2

Pengujian modul I2C LCD dan Modul LCD 16x2 bertujuan untuk mengetahui apakah kedua modul tersebut bisa digunakan untuk menampilkan karakter yang diinginkan melalui kontroler Arduino Mega 2560. Tabel pengujian dapat dimasukan ke dalam tabel 3.7.

Tabel 3.7 Tabel Pengujian Modul I2C LCD dan Modul LCD 16x2

No.	Karakter Yang Ingin ditampilkan Arduino Mega	Kriteria Pengujian	Karakter Yang Tampil Pada LCD
1.	Mencetak Karakter Huruf “ABCDEFGHIJKLMNO”	LCD Menampilkan karakter “ABCDEFGHIJKLMNO”	.....
2.	Mencetak Karakter Angka “123456789”	LCD Menampilkan karakter “123456789”	.....
3.	Mencetak Karakter Pada Memori LCD “0x7E”	LCD Menampilkan karakter “→”	.....

3.6.5. Pengujian *Driver motor* dengan Menggunakan L298N

Pengujian sensor garis dilakukan dengan cara menguji masing-masing *driver*. *Input driver* dihubungkan dengan pin digital dan PWM pada Arduino. Besar

tegangan yang digunakan untuk mengaktifkan IC L298 adalah 5 VDC, dan tegangan yang digunakan untuk menggerakan motor adalah 12VDC. Tahap pertama pengujian *driver* adalah dengan memberikan nilai digital pada pin 5 dan 7 lalu memberikan nilai analog pada pin 6 secara perlahan sampai nilai analog yang diberikan maksimal. Setelah itu ukurlah nilai tegangan keluaran dari pin 5 dan 7 dengan menggunakan multitester dan catatlah hasilnya pada tabel 3.8 pada kolom "Tegangan Pin Digital". Setelah itu ukurlah tegangan yang dihasilkan oleh pin 6 sesuai dengan nilai analog yang diberikan pada tabel 3.8 pada kolom "Tegangan Pin Analog". Ubahlah nilai digital yang sudah diberikan pada pin 5 dan 7 sesuai dengan tabel 3.8 lalu isilah kolom "Gerakan Motor" sesuai dengan kondisi motor pada tabel 3.8 Ubahlah nilai analog pada pin 6 secara perlahan sesuai dengan tabel 3.9 lalu ukurlah nilai teganagnnya dengan menggunakan multimeter dan masukanlah nilai hasil pengukuran pada tabel 3.9 pada kolom "Tegangan Pin Analog", setelah itu ukurlah nilai tegangan keluaran pada motor DC dengan menggunakan multimeter lalu masukanlah data tersebut pada tabel 3.9 pada kolom "Tegangan Keluaran Rangkaian".

Tabel 3.8 Tabel Pengujian Pergerakan Motor Menggunakan *Driver* L298N

No.	Data Pin 4	Data Pin 5	Data Pin 2	Kriteria Pengujian	Tegangan Pin Digital		Tegangan Pin Analog	Gerakan Motor
					Pin 4	Pin 5		
1.	0	0	0	Arah gerakan motor akan berubah ketika pin 5 dan pin 7 diberikan kondisi yang berbeda	.....	.....	.....	.....
2.	0	0	255		.....	.....	.....	.....
3.	0	1	0		.....	.....	.....	.....
4.	0	1	255		.....	.....	.....	.....
5.	1	0	0		.....	.....	.....	.....
6.	1	0	255		.....	.....	.....	.....
7.	1	1	0		.....	.....	.....	.....
8.	1	1	255		.....	.....	.....	.....

Tabel 3.9 Tabel Pengujian Tegangan Keluaran Rangkaian *Driver* L298N

No.	Data Pin 4	Data Pin 5	Data Pin 2	Kriteria Pengujian	Tegangan Pin Analog	Tegangan Keluaran Rangkaian
1.	0	1	0	Tegangan keluaran rangkaian akan semakin besar Ketika nilai analognya semakin besar	.....	.....
2.	0	1	50		.....	.....
3.	0	1	100		.....	.....
4.	0	1	150		.....	.....
5.	0	1	200		.....	.....
6.	0	1	250		.....	.....
7.	0	1	255		.....	.....

3.6.6. Pengujian Motor Servo sebagai Penggerak Penyemprot

Pengujian motor servo bertujuan untuk mengetahui apakah pergerakan motor servo yang akan digunkan bisa dikendalikan oleh Arduino Mega 2560 atau tidak. Tabel pengujian pergerakan motor servo dapat dimasukan ke dalam tabel 3.10.

Tabel 3.10 Tabel Pengujian Pergerakan Motor Servo sebagai Penggerak Penyemprot

No.	Pin 12	Kriteria Pengujian	Posisi Motor Servo	Tampilan
1.	Mengirimkan sinyal untuk membuat servo bergerak 0^0	Motor Servo bergerak menuju sudut 0^0	.....	.....
2.	Mengirimkan sinyal untuk membuat servo bergerak 30^0	Motor Servo bergerak menuju sudut 30^0	.....	.....
3.	Mengirimkan sinyal untuk membuat servo bergerak 60^0	Motor Servo bergerak menuju sudut 60^0	.....	.....
4.	Mengirimkan sinyal untuk membuat servo bergerak 90^0	Motor Servo bergerak menuju sudut 90^0	.....	.....
5.	Mengirimkan sinyal untuk membuat servo bergerak 120^0	Motor Servo bergerak menuju sudut 120^0	.....	.....

No.	Pin 12	Kriteria Pengujian	Posisi Motor Servo	Tampilan
6.	Mengirimkan sinyal untuk membuat servo bergerak 150^0	Motor Servo bergerak menuju sudut 150^0	.....	.....
7.	Mengirimkan sinyal untuk membuat servo bergerak 180^0	Motor Servo bergerak menuju sudut 180^0	.....	.....

3.6.7. Pengujian *Driver motor* dengan Menggunakan *Optocoupler*

Motor untuk memadamkan api menggunakan motor DC dengan operasi tegangannya adalah 12 VDC. Untuk menggerakan motor DC, digunakan sebuah rangkaian *driver optocoupler*. Pemberian sinyal *low* dan *high* pada kaki anoda led *optocoupler* digunakan untuk mengaktifkan dan menonaktifkan *optocoupler*. Tabel pengujian data pergerakan motor yang digunakan untuk memadamkan api dapat dilihat pada Tabel 3.11.

Tabel 3.11 Tabel Pengujian Data Pergerakan Motor yang digunakan untuk Memadamkan Api

No.	Kondisi Kaki Anoda		Kriteria Pengujian	Kondisi Motor	
	Sinyal Masukan	Tegangan		Tegangan Keluaran	Status Motor
1.	*Low*		Motor dalam kondisi diam dan tegangan keluaran 0V	.....	.....
2.	*High*		Motor dalam kondisi bergerak dan tegangan keluaran sekitar 12V	.....	.....

3.6.8. Tes Pergerakan Robot

Pergerakan robot dipengaruhi oleh arah putaran dari masing-masing motr DC yang dipasang pada *base* robot. Untuk mengetes pergerakan robot, maka peneliti menggerakan roda robot secara bergantian, lalu menggerakan kedua roda robot secara bersamaan. Untuk mengatur kecepataan putaran robot dilakukan dengan

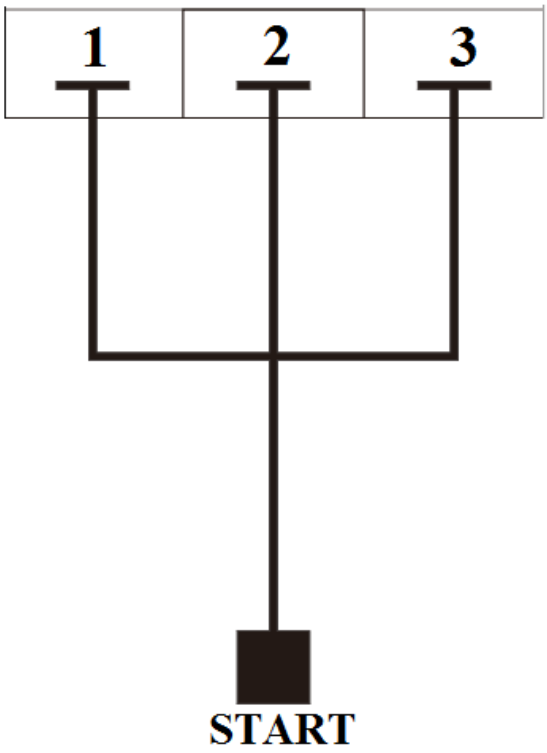
memberikan nilai PWM pada masing-masing motor. Setelah melakukan kedua hal tersebut peneliti mencatat hasilnya pada tabel 3.12.

Tabel 3.12 Tabel Pengamatan Gerakan Robot dengan Menggerakan Roda Robot

No.	Roda Kiri	Roda Kanan	Kriteria Pengujian	Gerakan robot
1.	Diam	Diam	Robot dalam Kondisi diam	.....
2.	Diam	Maju	Robot dalam Kondisi belok ke kiri dan arahnya maju	.....
3.	Diam	Mundur	Robot dalam Kondisi belok ke kanan dan arahnya mundur	.....
4.	Maju	Diam	Robot dalam Kondisi belok ke kanan dan arahnya maju	.....
5.	Mundur	Diam	Robot dalam Kondisi belok ke kiri dan arahnya mundur	.....
6.	Maju	Maju	Robot dalam Kondisi maju	.....
7.	Maju	Mundur	Robot dalam Kondisi putar balik ke arah kanan	.....
8.	Mundur	Maju	Robot dalam Kondisi putar balik ke arah kiri	.....
9.	Mundur	Mundur	Robot dalam Kondisi mundur	.....

3.6.9. Pengujian Robot dengan Variasi Tempat Pemadaman Api

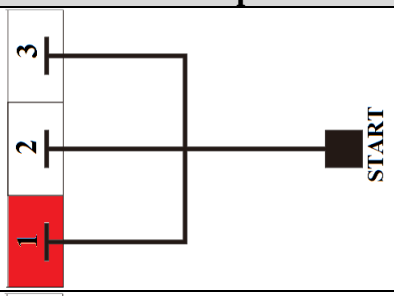
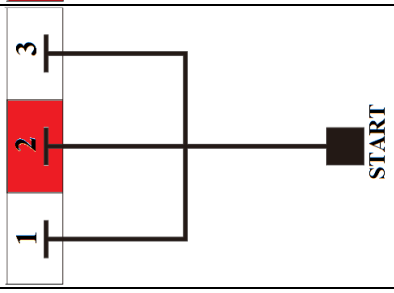
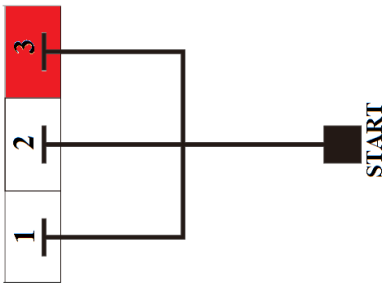
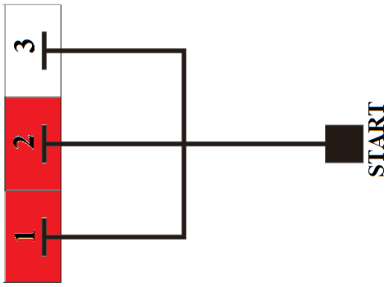
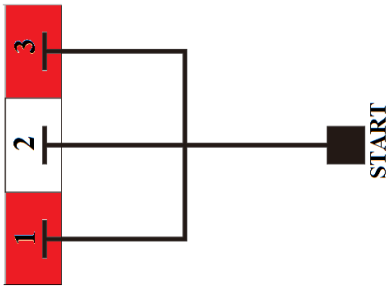
Dari desain jalur*(track*) yang ada pada Gambar 3.32, maka peneliti menentukan penomoran ruangan di mulai dari ruangan kiri sebagai ruangan dengan nomor 1, ruangan tengah sebagai ruangan dengan nomor 2, dan yang terakhir adalah ruangan kanan sebagai ruangan dengan nomor 3. Penomoran ruangan tersebut dapat dilihat pada Gambar 3.34.

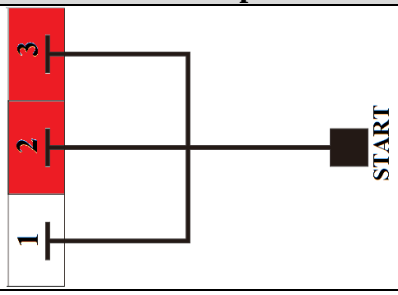
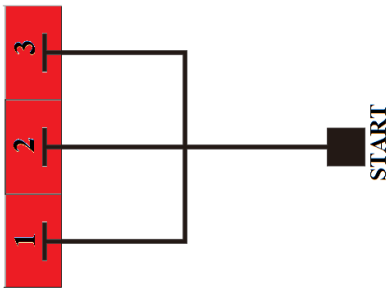


Gambar 3.34 Penomoran Ruangan

Dari ketiga lokasi titik api ini ditemukan banyak variasi dalam memadamkan api, variasi-variasi titik api tersebut dapat dilihat pada Tabel 3.13. pemadaman api pada satu titik api di salah satu ruangan akan dilakukan dengan cara robot langsung menuju ruangan tersebut. Dan pemadaman api pada dua titik api yang ada di dalam ruangan akan dilakukan dengan cara menuju ke ruangan yang titik apinya paling terdekat dengan robot terlebih dahulu. Lalu setelah itu ruangan-ruangan yang lain.

Tabel 3.13 Variasi Pemadaman api

No.	Posisi Api	Sensor Aktif	Keterangan
1.		Sensor api 1 aktif Pin A0 = *High* Pin A1 = *Low* Pin A2 = *Low*	Lokasi api berada di ruang 1
2.		Sensor api 2 aktif Pin A0 = *Low* Pin A1 = *High* Pin A2 = *Low*	Lokasi api berada di ruang 2
3.		Sensor api 3 aktif Pin A0 = *Low* Pin A1 = *Low* Pin A2 = *High*	Lokasi api berada di ruang 3
4.		Sensor api 1 dan 2 aktif Pin A0 = *High* Pin A1 = *High* Pin A2 = *Low*	Lokasi api berada di ruang 1 dan 2, dengan urutan api di ruang 1 lalu di ruang 2
5.		Sensor api 1 dan 3 aktif Pin A0 = *High* Pin A1 = *Low* Pin A2 = *High*	Lokasi api berada di ruang 1 dan 3, dengan urutan api di ruang 3 lalu di ruang 1

No.	Posisi Api	Sensor Aktif	Keterangan
6.		Sensor api 2 dan 3 aktif Pin A0 = *Low* Pin A1 = *High* Pin A2 = *High*	Lokasi api berada di ruang 2 dan 3, dengan urutan api di ruang 2 lalu di ruang 3
7.		Sensor api 1, 2 dan 3 aktif Pin A0 = *High* Pin A1 = *High* Pin A2 = *High*	Lokasi api berada di ruang 1, 2 dan 3, dengan urutan api di ruang 1, di ruang 2 lalu di ruang 3, kemudian dengan urutan api di ruang 2, di ruang 3 lalu di ruang 1, dan kemudian dengan urutan api di ruang 3, di ruang 1 lalu di ruang 2.

3.6.10. Uji Kelayakan Pemadaman Api oleh Robot

Uji kelayakan pemadaman api digunakan untuk mengetahui kelayakan robot *line follower* pemadam api yang diprogram untuk memadamkan api secara otomatis. Uji kelayakan dilakukan dengan cara melakukan simulasi memadamkan api yang dilakukan secara berulang-ulang dengan meletakan api yang berasal dari lilin pada ruangan yang sudah disediakan. Pengujian pemadaman api dilakukan berdasarkan variasi posisi api dimana n di dalam rumus adalah jumlah percobaan yang berhasil dilakukan oleh robot. Selain itu, pengujian ini juga bertujuan untuk mengetahui tingkat efektifitas robot dalam memadamkan api.

Tabel 3.14 Pengujian Posisi Api 1

Posisi Api 1			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.15 Pengujian Posisi Api 2

Posisi Api 2			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.16 Pengujian Posisi Api 3

Posisi Api 3			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.17 Pengujian Posisi Api 4

Posisi Api 4			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.18 Pengujian Posisi Api 5

Posisi Api 5			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.19 Pengujian Posisi Api 6

Posisi Api 6			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Tabel 3.20 Pengujian Posisi Api 7

Posisi Api 7			
No.	**Nomor Pengujian**	**Berhasil atau tidak**	**Waktu(detik)**
1.	Percobaan 1	.....	.....
2.	Percobaan 2	.....	.....
3.	Percobaan 3	.....	.....

Presentase keberhasilan = $\frac{n}{3}$ x 100%

Untuk mengetahui kelayakan robot *line follower* pemadam api, maka berdasarkan presentase keberhasilan dalam memadamkan api, nilai-nilai presentase keberhasilan tersebut dimasukan ke dalam tabel 3.21 untuk kemudian dihitung presentase keberhasilan secara keseluruhan.

Tabel 3.21 Uji Kelayakan *Line Follower* Pemadam Api

No.	Posisi Api	Jumlah Pengujian	Hasil Pengujian		Presentase Keberhasilan (%)
			Berhasil	Gagal	
1.	Posisi 1	3	.....	.....	.....
2.	Posisi 2	3	.....	.....	.....
3.	Posisi 3	3	.....	.....	.....
4.	Posisi 4	3	.....	.....	.....
5.	Posisi 5	3	.....	.....	.....
6.	Posisi 6	3	.....	.....	.....
7.	Posisi 7	3	.....	.....	.....

$$\text{Presentase keberhasilan} = \frac{\text{Hasil Pengujian Berhasil}}{\text{Jumlah Pengujian}} \text{ x } 100\%$$

BAB IV

HASIL PENELITIAN

4.1. Deskripsi Hasil Penelitian

Pada bab ini membahas mengenai hasil pembuatan bagian-bagian alat yang telah dirancang pada bagian perancangan perangkat keras di BAB III dan membahas mengenai pengujian robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino, dimana pada alat ini arduino mega 2560 dan arduino Uno telah diunggah program yang sesuai dengan *flowchart* robot dan *monitoring* ruangan pada BAB III.

Berdasarkan rancangan penelitian yang telah dijelaskan pada BAB III, maka peneliti akan mendeskripsikan hasil penelitian berupa prinsip kerja alat dan langkah-langkah penggunaan alat.

4.1.1. Prinsip Kerja Alat

Pada robot robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino ini memiliki prinsip kerja untuk memadamkan api yang berada di dalam ruangan yang berasal dari lilin. Untuk perpindahannya robot ini menggunakan prinsip *line follower* yang akan mengikuti garis yang sudah ditentukan oleh peneliti. Dalam pengujiannya, robot ini memadamkan api yang berasal dari lilin dengan tinggi lilin 7,5cm-10cm dan berdiameter 1,5cm-3cm. Saat pertama robot diletakan di *track*, robot dalam kondisi diam dan menunggu informasi dari kontroler yang berada di ruangan. Untuk mendeteksi keberadaan api di dalam ruangan sistem ini menggunakan *flame sensor*, setelah mendeteksi

keberadaan api lalu Arduino Uno mengirimkan data tentang lokasi api melalui bluetooth ke Arduino Mega 2560.

Setelah data diterima, robot akan bergerak menuju ke ruangan yang diduga ada apinya. Setelah sampai di ruangan, robot langsung akan memadamkan api dan setelah padam robot akan menunggu informasi tentang keberadaan api di ruangan yang lainnya. Jika tidak ada api lagi di ruangan yang lain maka robot akan kembali lagi ke titik awal(alas kotak dengan warna hitam). Tapi, jika diduga ada api di ruangan lain maka robot akan segera menuju ruangan tersebutuntuk memadamkannya dan kemudian memeriksa kembali informasi tentang keberadaan api di ruangan yang lainnya. Robot ini akan langsung menuju ke ruangan yang diduga memiliki titik api tanpa menggunakan metode *scaning* setiap ruangan. Jika api di dalam dua ruangan atau lebih menyala secara bersamaan, maka kesensitifan *flame sensor* di dalam ruanganlah yang akan menentukan priororitas ruangan yang akan dipadamkan terlebih dahulu.

4.1.2. Langkah-Langkah Penggunaan Alat

Berikut adalah langkah-langkah penggunaan robot robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino:

1. Hubungkan alat dan laptop dengan menggunakan kabel dengan port USB.
2. *Download* program yang sudah dibuat di Arduino IDE ke dalam alat.
3. Letakan lilin di temapt yang sudah disediakan.
4. Letakan robot diatas *track* hitam.
5. Hubungkan power supply dengan sistem.
6. Nyalakan sistem dengan menekan saklar power pada alat.

7. Pastikan kontroler robot dan kontroler ruangan sudah terhubung dengan melihat LCD tidak ada tampilan “Tidak ada konektifitas Reset Arduino!!!”
8. Cek sensitifitas sensor garis melalui tampilan yang ada pada LCD.
9. Jika kurang sensitif, lakukanlah *scaning sensor* dan gerakan sensor robot agar sensor melalui garis putih dan hitam.
10. Setelah mengecek sensor, masuklah dalam mode tunggu.
11. Letakan robot pada *track*.
12. Nyalakan lilin pada ruang berapapun.
13. Robot akan mengikuti garis yang sudah ditentukan menuju ruangan yang diduga ada apinya.
14. Pada garis akhir, robot akan berhenti dan kemudian memadamkan api.
15. Jika sudah berhasil memadamkan semua titik api, robot akan kembali ke tempat awal.
16. Jika terdengar suara alarm dari buzzer, ini menandakan bahwa komunikasi bluetooth yang dillakukan kedua kontroler gagal dilakukan.
17. Diperlukan operator agar robot bisa melakukan *scaning sensor* dan masuk dalam mode tunggu.

4.1.3. Kekurangan Alat

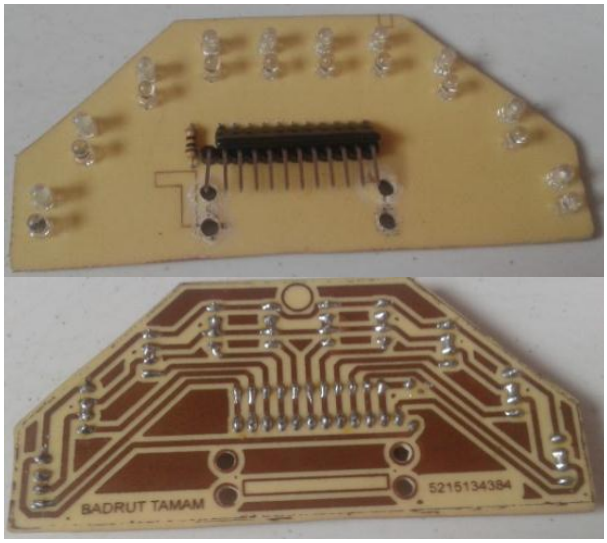
Berikut adalah kekurangan-kekurangan yang terdapat pada Robot *Line Follower* Pemadam Api Memanfaatkan *Flame Sensor* dan Bluetooth Berbasis Arduino:

1. *Track line follower* tidak bisa digunakan di tempat yang permukaannya tidak rata.
2. Ketinggian lilin tidak bisa kurang dari 10 cm dan lebih dari 15 cm.

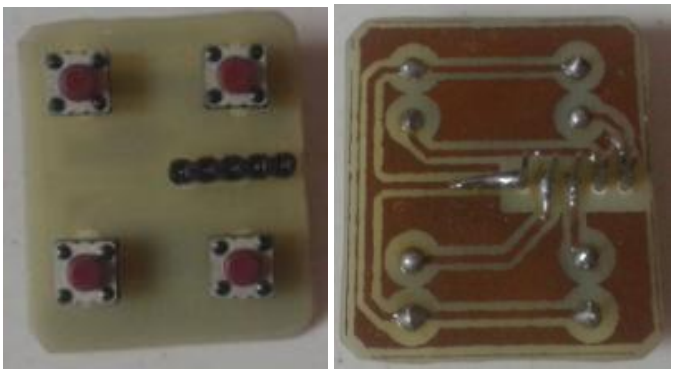
3. Volume air yang dibawa robot tidak bisa kurang dari 300 ml dan lebih dari 500 ml.
4. Robot tidak bisa digunakan diluar ruangan.

4.2. Analisis Hasil Penelitian

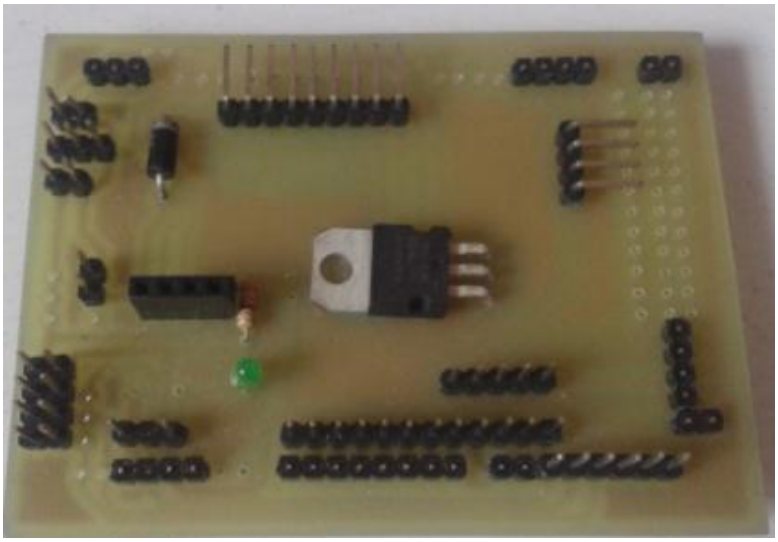
Pada perancangan perangkat keras di BAB III, bagian-bagian yang dibuat diantaranya adalah rangkaian sensor garis yang dapat dilihat pada Gambar 4.1, rangkaian *push button* yang dapat dilihat pada Gambar 4.2, rangkaian *shield* Arduino Mega 2560 yang dapat dilihat pada Gambar 4.3, rangkaian *shield* Arduino Uno yang dapat dilihat pada Gambar 4.4, rangkaian *driver motor* dengan menggunakan L298N yang dapat dilihat pada Gambar 4.5, rangkaian *driver motor* dengan menggunakan *optocoupler* yang dapat dilihat pada Gambar 4.6, *desain base* robot yang dapat dilihat pada Gambar 4.7, *desain standing* servo robot yang dapat dilihat pada Gambar 4.8, dan *desain track*(jalur) yang dapat dilihat pada Gambar 4.9.

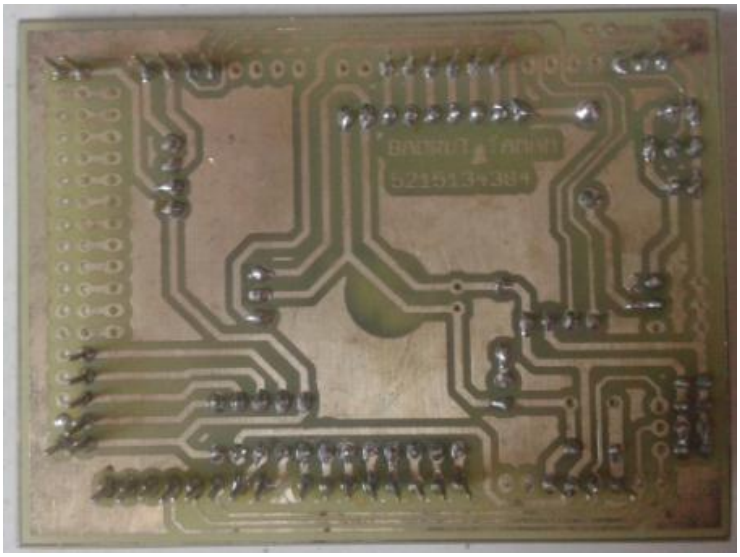


Gambar 4.1 Rangkaian Sensor Garis

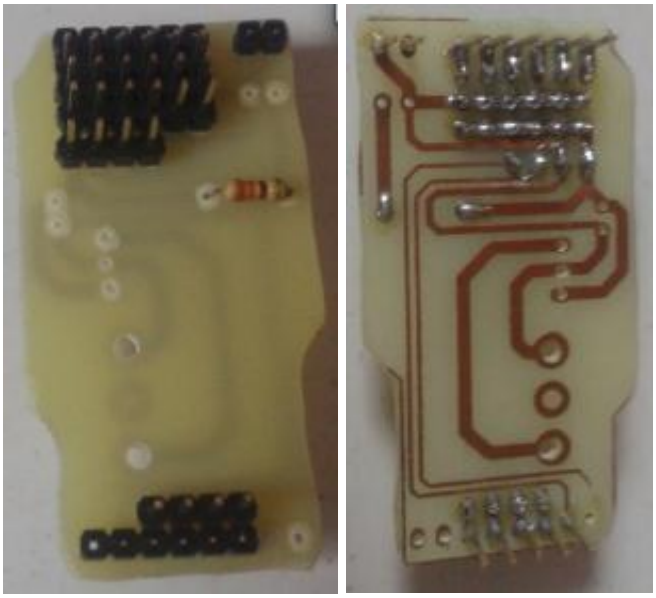


Gambar 4.2 Rangkaian *Push Button*

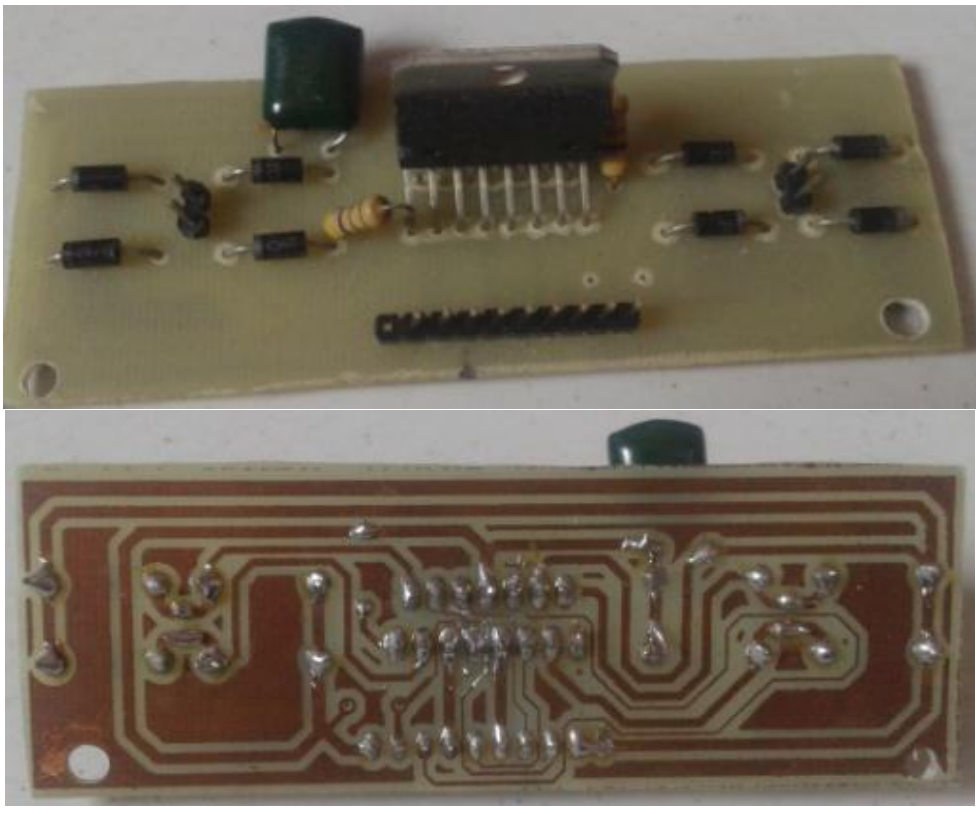




Gambar 4.3 Rangkaian *Shield* Arduino Mega 2560



Gambar 4.4 Rangkaian *Shield* Arduino Uno

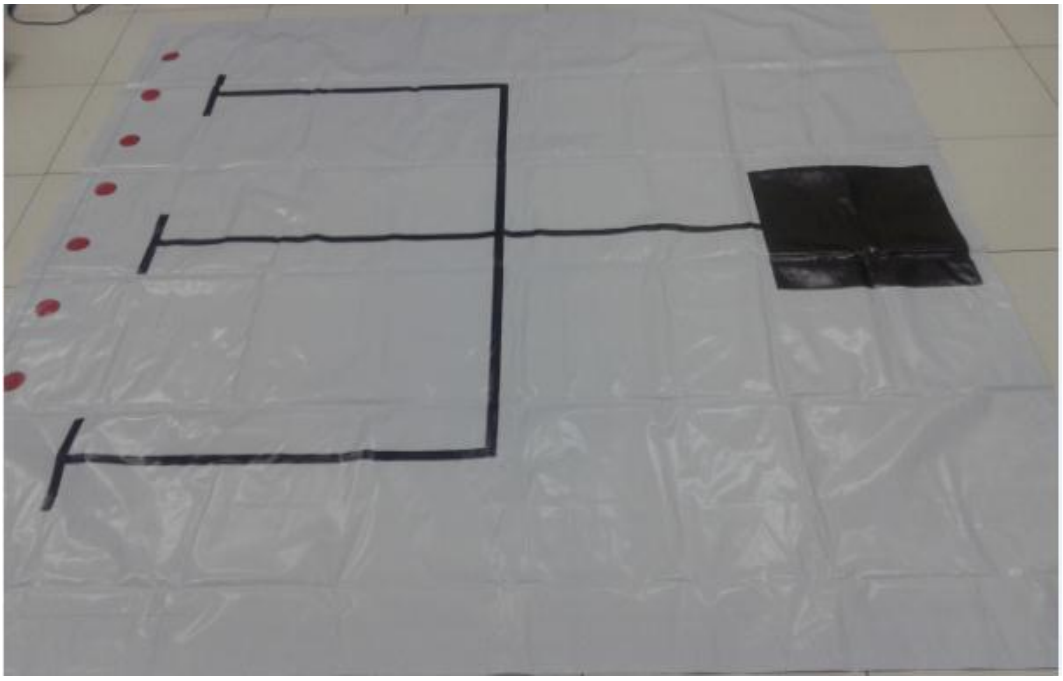


Gambar 4.5 Rangkaian *Driver Motor* dengan Menggunakan L298N

Gambar 4.6 Rangkaian *Driver Motor* dengan Menggunakan *Optocoupler*

Gambar 4.7 *Desain Base* Robot

Gambar 4.8 *Desain Standing* Servo Robot



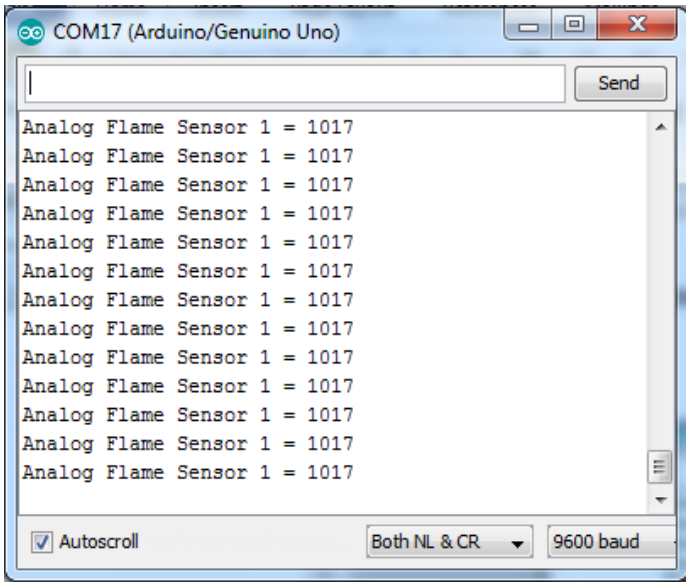
Gambar 4.9 *Desain Track*(Jalur)

4.2.1. Hasil Pengujian *Flame Sensor* KY-026

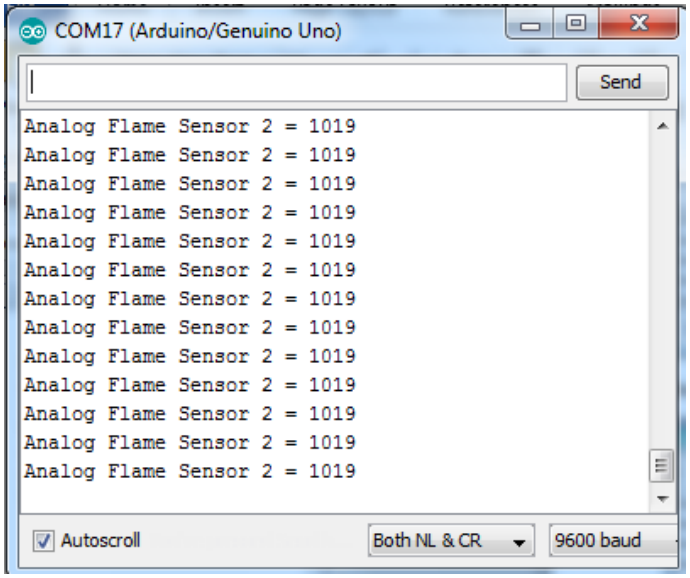
Pengujian *flame sensor* KY-026 bertujuan untuk mengetahui data nilai analog yang dibaca oleh *flame sensor* saat ada api dan saat tidak ada api, serta untuk mengetahui mengetahui data nilai analog yang dibaca oleh *flame sensor* pada jarak-jarak tertentu. Pengujian ini dilakukan dengan dua tahap, tahap pertama adalah dengan cara meletakan *flame sensor* di tempat yang ketinggiannya sejajar dengan api, lalu melihat data nilai analog yang dibaca melalui serial monitor arduino. Tahap kedua adalah dengan cara meletakan *flame sensor* di tempat yang ketinggiannya sejajar dengan api, lalu mengukur jarak antara *flame sensor* dan api dengan menggunakan penggaris sesuai dengan dataa jarak yang ada pada tabel 3.4. setelah itu melihat data nilai analog yang dibaca pada masing-masing jarak melalui serial monitor arduino.

Tabel 4.1 Hasil Pengukuran Nilai Analog pada *Flame Sensor* di Dalam Ruangan

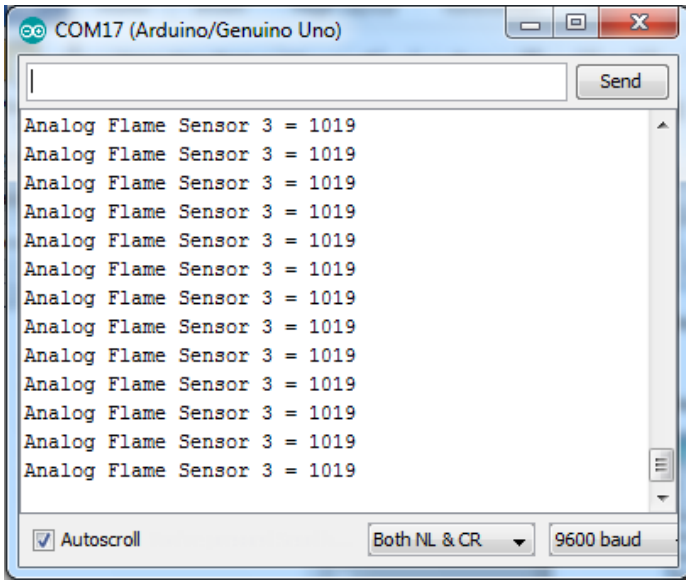
No.	Label	Kriteria Pengujian	Nilai Analog		Tampilan	
			Pada Daerah Bebas Api	Pada Daerah Api	Pada Daerah Bebas Api	Pada Daerah Api
1.	*Flame sensor* 1	Nilai analog dan tegangan keluaran sensor saat mendetaksi api lebih kecil dibandingkan dengan saat tidak ada api.	1017	23	Gambar 4.10	Gambar 4.13
2.	*Flame sensor* 2		1019	44	Gambar 4.11	Gambar 4.14
3.	*Flame sensor* 3		1019	46	Gambar 4.12	Gambar 4.15



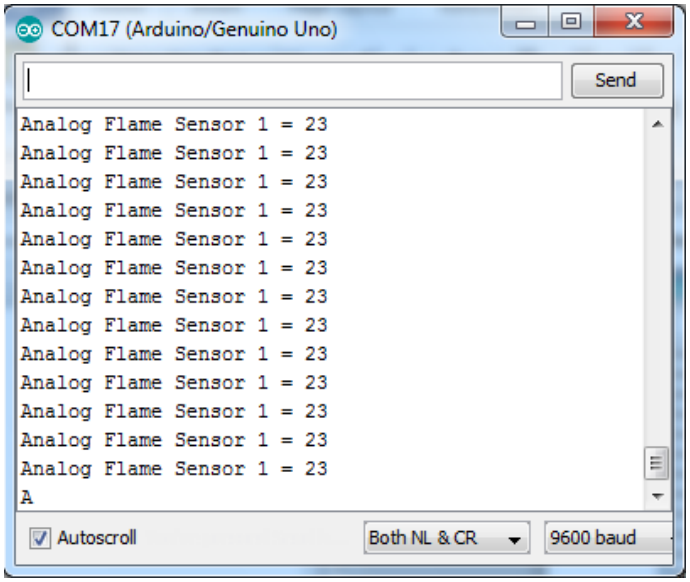
Gambar 4.10 Pembacaan Nilai Analog *Flame Sensor* di Ruang 1 dalam Kondisi Ruangan Tanpa Api



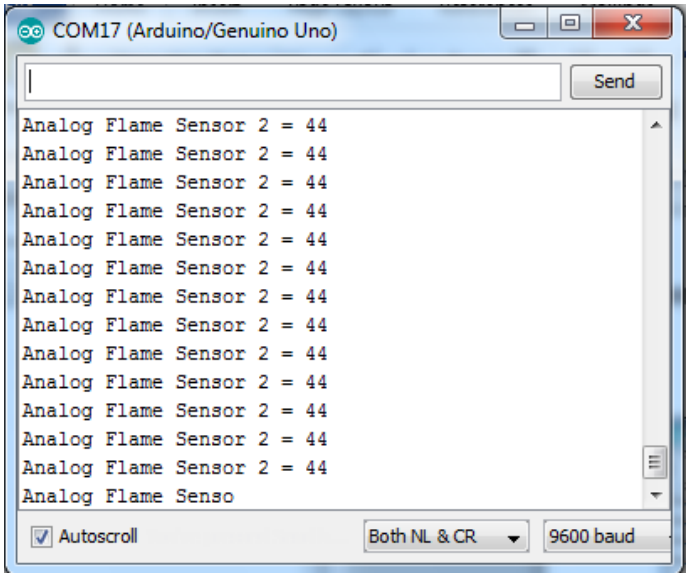
Gambar 4.11 Pembacaan Nilai Analog *Flame Sensor* di Ruang 2 dalam Kondisi Ruangan Tanpa Api



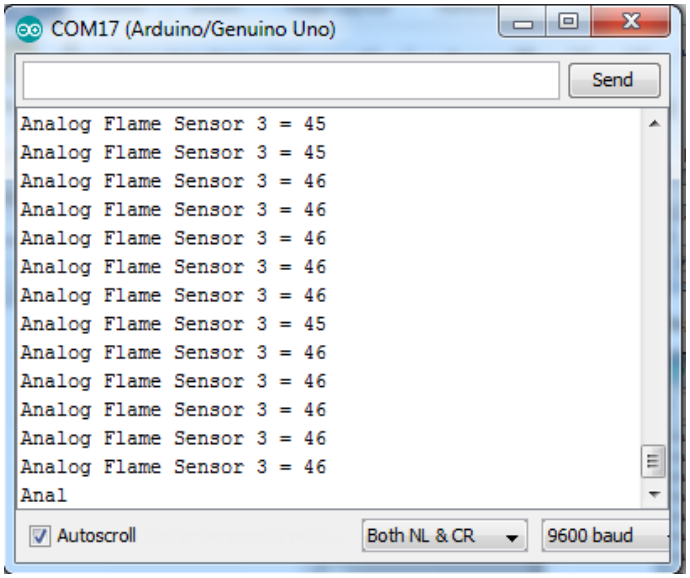
Gambar 4.12 Pembacaan Nilai Analog *Flame Sensor* di Ruang 3 dalam Kondisi Ruangan Tanpa Api



Gambar 4.13 Pembacaan Nilai Analog *Flame Sensor* di Ruang 1 dalam Kondisi Ruangan Ada Api

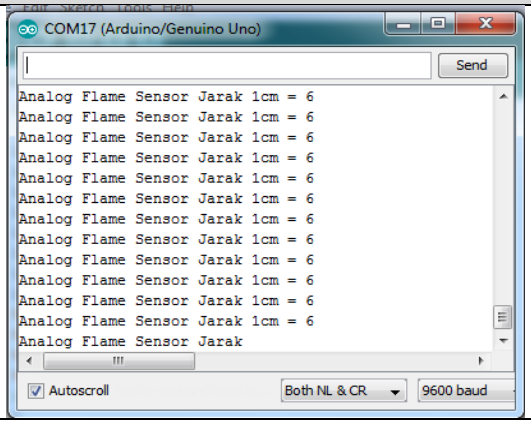


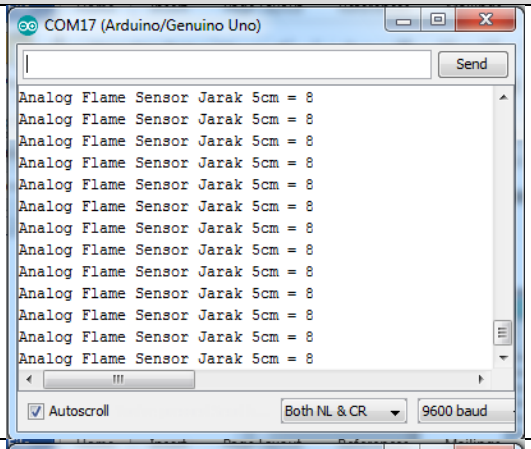
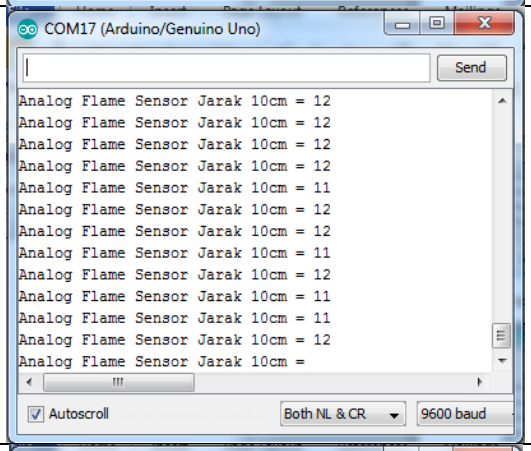
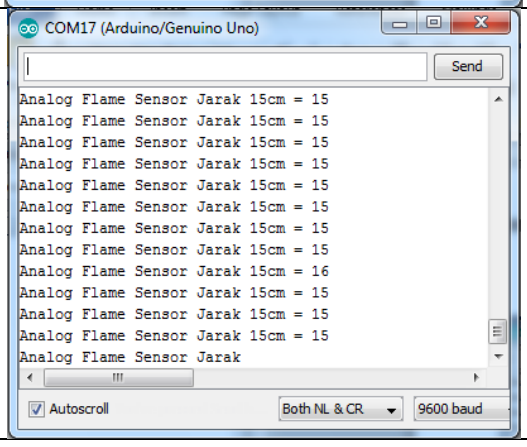
Gambar 4.14 Pembacaan Nilai Analog *Flame Sensor* di Ruang 2 dalam Kondisi Ruangan Ada Api

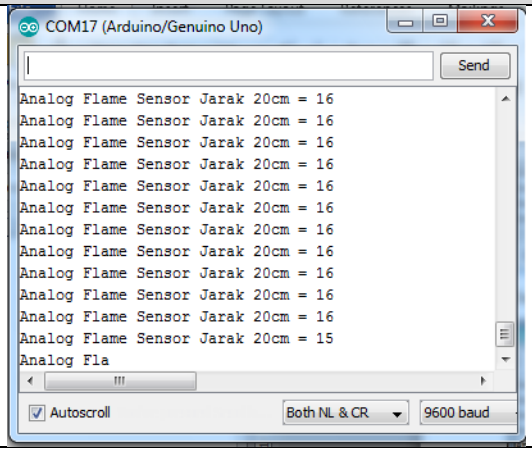


Gambar 4.15 Pembacaan Nilai Analog *Flame Sensor* di Ruang 3 dalam Kondisi Ruangan Ada Api

Tabel 4.2 Hasil Pengukuran Nilai Analog pada *Flame Sensor* di Robot

No.	Jarak	Kriteria Pengujian	Hasil Pengukuran Nilai Analog	Tampilan
1.	1 cm	Nilai analog dan tegangan keluaran yang dihasilkan sensor akan semakin kecil	6	

No.	Jarak	Kriteria Pengujian	Hasil Pengukuran Nilai Analog	Tampilan
2.	5 cm	ketika jarak antara api dengan sensor semakin kecil.	8	
3.	10 cm		12	
4.	15 cm		15	

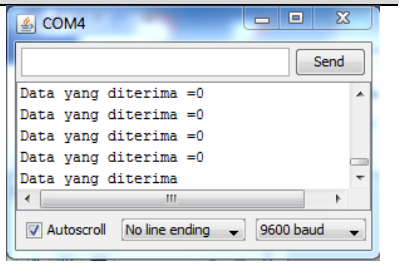
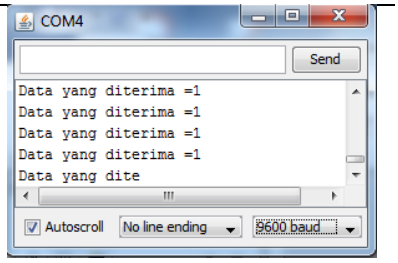
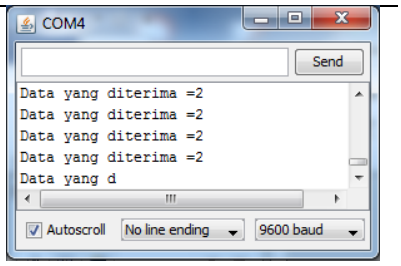
No.	Jarak	Kriteria Pengujian	Hasil Pengukuran Nilai Analog	Tampilan
5.	20 cm		16	

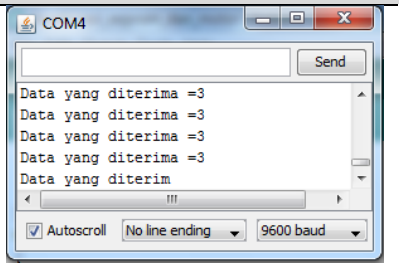
Berdasarkan hasil pengujian pada tabel 4.1 dan tabel 4.2, didapatkan bahwa nilai ADC yang dibaca oleh *flame sensor* KY-026 akan bernilai kecil saat mendeteksi adanya api, dan akan bernilai besar saat tidak mendeteksi adanya api. Nilai ADC yang dibaca oleh *flame sensor* KY-026 akan semakin kecil saat jarak sensor dengan api semakin dekat.

4.2.2. Hasil Pengujian Komunikasi Bluetooth HC-05

Pengujian komunikasi serial bluetooth bertujuan untuk menguji komunikasi antara dua perangkat modul bluetooth HC-05. Selain itu pengujian ini bertujuan untuk mengetahui apakah modul HC-05 bisa dikendalikan menggunakan Arduino atau tidak. Data yang diterima oleh modul bluetooth HC-05 yang betugas sebagai *slave* akan ditampilkan pada serial monitor Arduino *IDE* sebagai indikator keberhasilan komunikasi antara dua Arduino melalui komunikasi serial bluetooth.

Tabel 4.3 Hasil Pengujian Komunikasi Serial Dua Bluetooth

No.	Pengujian arah komunikasi	Data yang Akan dikrirm	Kriteria pengujian	Data yang terecetak di serial monitor Arduino *IDE*	Tampilan
1.	Pengiriman dan peneriamaan data dari *Transimiter* bluetooth *master* kepada bluetooth *slave*	Data terikirm dari Arduino 2 berupa *string* "0" dan diterima oleh Arduino 1.	Data yang diterima oleh *slave* akan sama dengan data yang dikirim oleh *master*.	0	
2.		Data terikirm dari Arduino 2 berupa *string* "1" dan diterima oleh Arduino 1.		1	
3.		Data terikirm dari Arduino 2 berupa *string* "2" dan diterima oleh Arduino 1.		2	

No.	Pengujian arah komunikasi	Data yang Akan dikrirm	Kriteria pengujian	Data yang terecetak di serial monitor Arduino *IDE*	Tampilan
4.		Data terikirm dari Arduino 2 berupa *string* "3" dan diterima oleh Arduino 1.		3	

Berdasarkan hasil pengujian pada tabel 4.3, didapatkan bahwa data yang diterima oleh bluetooth yang berfungsi sebagai *slave* sudah sesuai dengan data yang dikirimkan oleh bluetooth yang berfungsi sebagai *smaster*. Berdasarkan hasil taabel 4.3 tersebut, maka data yang didapatkan sudah sesuai dengan kriteria pengujian dan pengujian dapat dinyatakan berhasil.

4.2.3. Hasil Pengujian Sensor Garis

Pengujian ini bertujuan untuk mengetahui apakah sensor garis dapat memberikan perubahan nilai ADC (*Analog to Digital Converter*) sesuai dengan perubahan cahaya yang diterima dari pantulan cahaya yang berasal dari jalur. Pengujian ini juga berfungsi untuk melihat seberapa besar nilai error yang ada pada ADC (*Analog to Digital Converter*) Arduino Mega 2560. Nilai ADC (*Analog to Digital Converter*) dan tegangan keluaran pada masing-masing sensor garis dapat dilihat pada Gambar 4.16 sampai dengan Gambar 4.55.

Tabel 4.4 Hasil Pengujian dan Pengukuran Sensor Garis

No.	Label	Kriteria Pengujian	Tegangan (V)		Nilai Analog	
			Alas Putih	Alas Hitam	Alas Putih	Alas Hitam
1.	Sensor 1	Nilai analog dan tegangan keluaran yang dihasilkan sensor saat berada di atas garis hitam akan lebih besar dibandingkan pada saat berada di garis putih	209,9mV	3,65V	39	744
2.	Sensor 2		197,1mV	3,083V	37	630
3.	Sensor 3		214,1mV	3,80V	41	775
4.	Sensor 4		199,5mV	3,421V	38	700
5.	Sensor 5		200,9mV	3,370V	38	689
6.	Sensor 6		204,0mV	3,141V	38	646
7.	Sensor 7		197,1mV	3,250V	37	655
8.	Sensor 8		188,1mV	2,061V	36	418
9.	Sensor 9		214,5mV	3,247V	41	666
10.	Sensor 10		214,4mV	2,840V	40	578

Berdasarkan hasil pengujian pada tabel 4.4, didapatkan bahwa nilai ADC yang dibaca oleh sensor garis akan bernilai kecil saat berada diatas *track* berwarna putih, dan akan bernilai besar saat berada diatas *track* berwarna hitam.

Gambar 4.16 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 1 di Alas Berwarna Putih

Gambar 4.17 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 2 di Alas Berwarna Putih

Gambar 4.18 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 3 di Alas Berwarna Putih

Gambar 4.19 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 4 di Alas Berwarna Putih

Gambar 4.20 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 5 di Alas Berwarna Putih

Gambar 4.21 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 6 di Alas Berwarna Putih

Gambar 4.22 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 7 di Alas Berwarna Putih

Gambar 4.23 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 8 di Alas Berwarna Putih

Gambar 4.24 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 9 di Alas Berwarna Putih

Gambar 4.25 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 10 di Alas Berwarna Putih

Gambar 4.26 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 1 di Alas Berwarna Hitam

Gambar 4.27 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 2 di Alas Berwarna Hitam

Gambar 4.28 Hasil Pengukuran Tegangan Keluaran pada Sensor Garis 3 di Alas Berwarna Hitam

1%, hanya ada satu pin yang mempunyai error di atas 1%, sedangkan pin ADC lainnya di bawah 1%.

Pada tabel 4.5 menampilkan data hasil pengujian pada I2C LCD dan modul LCD, yang dilakukan dengan cara mencetak beberapa karakter ke LCD, lalu setelah itu melihat kesesuaian karakter yang tercetak di LCD dengan karakter yang ingin di cetak..

Pada tabel 4.6 dan tabel 4.7 menampilkan data dan hasil pengujian pergerakan motor dengan menggunakan *driver* motor L298N. Pengujian dilakukan untuk memastikan pemberian logika yang benar supaya robot dapat bergerak maju, mundur, belok kanan, belok kiri dan diam sesuai dengan yang diinginkan..

Pada tabel 4.8 menampilkan data dan hasil pengujian pergerakan motor servo, pengujian dilakukan untuk mengetahui nilai derajat servo yang tepat supaya penyemprot bisa mengarah langsung ke lilin yang menyala, baik pada saat lilin berada di kiri, di tengah, atau di kanan ruangan.

Pada tabel 4.11 sampai dengan tabel 4.18 menampilkan hasil data pengujain keberhasilan sistem dalam mendeteksi keberadaan api dan memadamkan api.

Berdasarkan data pada tabel 4.11 sampai tabel 4.13 dapat kita ketahui bahwa waktu yang dibutuhkan untuk memadamkan api di satu ruangan membutuhkan waktu kurang dari 20 detik, bahkan untuk memadamkan api di ruang 2, robot hanya membutuhkan waktu 14 detik. Pada tabel 4.14 sampai tabel 4.16 dapat kita ketahui bahwa waktu yang dibutuhkan untuk memadamkan api di dua ruangan membutuhkan waktu kurang dari 40 detik. Dan berdasarkan data pada tabel 4.17

dapat kita ketahui bahwa waktu yang dibutuhkan untuk memadamkan api di tiga ruangan membutuhkan waktu kurang dari 55 detik.

Berdasarkan data pada tabel 4.18 dapat kita ketahui bahwa robot berhasil menyelesaikan tugasnya untuk memadamkan api sebanyak 20 kali dari 21 percobaan. kegagalan tersebut disebabkan oleh kurangnya jumlah air yang dibawa robot sehingga ketika ingin memadamkan api di ruang terakhir robot tidak bisa menyemprotkan air. Dengan data keberhasilan tersebut, maka presentase keberhasilan robot untuk memadamkan api adalah 95,24%.

4.4. Aplikasi Hasil Penelitian

Berdasarkan perancangan dan hasil pengujian yang sudah dilakukan maka robot *line follower* pemdam api memaanfaatkan *flame sensor* dan bluetooth berbasis Arduino dapat diaplikasikan untuk mengurangi jumlah kebakaran dengan mengurangi presentase kebakaran yang disebabkan oleh api yang berasal dari lilin. Selain itu robot ini dapat juga digunnakan untuk salah satu solusi pemanfaatan teknologi robotik dibidang keamanan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, melalui tahap perancangan, pembuatan dan pengujian robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino dapat disimpulkan bahwa:

1. Perancangan robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino dapat direalisasikan menggunakan R&D.
2. Robot *line follower* pemadam api dapat direalisasikan dengan membuat *base* robot, air, sensor garis, *flame sensor*, Arduino, motor DC, *driver motor* DC, *flame sensor,*Arduino dan modul bluetooth.
3. Rata-rata waktu yang dibutuhkan untuk memadamkan api di 1 ruangan kurang dari 20 detik, untuk 2 ruangan kurang dari 40 detik dan untuk 3 ruangan kurang dari 55 detik.

5.2. Saran

Dalam proses perancangan robot *line follower* pemadam api memanfaatkan *flame sensor* dan bluetooth berbasis Arduino peneliti menyadari bahwa masih banyak kekurangan dan kelemahan, maka dari itu peneliti memberikan saran untuk peneliti lain yang ingin mengembangkan alat ini. Berikut adalah saran-sarannya:

1. Desain robot yang perlu diperbaiki agar mudah dalam meletakan semua perangkat elektronik dan perangkat-perangkat lainnya, serta tidak bertumpu pada satu bagian robot saja.
2. Bahan konstruksi harus lebih kokoh supaya bisa menopang berat robot yang berat dan bertumpu pada satu titik.
3. Pemilihan motor DC yang memiliki torsi yang lebih besar agar robot bisa berjalan walaupun diberikan nilai PWM yang kecil.
4. Jika ingin diimplementasikan ditempat yang luas, maka penggunaan bluetooth sebaiknya dirubah ke wifi agar jangkauan komunikasinya lebih luas.
5. Pastikan jumlah air yang dibawa oleh robot melebihi tinggi *extinguizer*.

DAFTAR PUSTAKA

Akbar, Z., dkk. (2012). *Penentuan Rangking Berdasarkan Waktu Lomba Renang pada Rangkaian Push-Button Berbasiskan Mikrokontroler At89s51*. Depok: Universitas Gunadarma.:3.

Andrianto, H. & Darmawan, A., (2016). *Arduino Belajar Cepat dan Pemrograman.* Bandung: Informatika Bandung.

Anwar, S., dkk. (2010). *Pemakaian Remote Control TV dengan Menggunakan Mikrokontroler At89s51 sebagai Alat Pemutus Dan Penghubung Tegangan Kwh Meter 1 Phasa.* Padang: Politeknik Negeri Padang.:3.

Banzi, M. (2005). *ArduinoBoardMega*. Dipetik November 27, 2016, dari Arduino web site: www.arduino.cc

Banzi, M. (2005). *ArduinoBoardUno*. Dipetik November 27, 2016, dari Arduino web site: www.arduino.cc

Boylestad, R. L. & Nahelsky, L. (2013). *Electronic Devices and Circuit Theory.* USA: Pearson Education, Inc.

Budiharto, W. (2014). *Panduan Praktis Perancangan dan Pemrograman Hasta Karya Robot.* Yogyakarta: CV. ANDI OFFSET.

Chandra, E. A., dkk., (2013). *Robot Line Follower (Line Tracking Robot)*. :3.

Endaryono, P. J., dkk., (2014). *Rancang Bangun Sistem Pembayaran Mandiri pada Wahana Permainan.* Surabaya: STIMIK STIKOM Surabaya.:3.

Fatah, L. A. dan Alimudin, (2009). *Prototipe Sistem Pendeteksi Dini Kebakaran dengan Sms sebagai Media Informasi Berbasis Mikrokontroler*. Bandung: STMIK LPKIA.:2.

Firmansyah, T., dkk., (2016). *Rancang Bangun Low Power Elektric Surgery (Pisau Bedah Listrik) pada Frekuensi 10 KHz* . Banten: Universitas Sultan Ageng Tirtayasa.:4.

[FT] Fakultas Teknik. 2015. *Buku Panduan Penyusunan Skripsi Dan Non Skripsi* Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Ginting, V. b., dkk., (2013). *Sistem Pengendalian Asap Rokok Multikanal dengan Menggunakan PWM Berbasis Mikrokontroler ATMega* 8.:2.

Guntama, A. A. (2017). *Prototype Robot Line Follower Pengangkut Sampah dengan Memanfaatkan Solar Cell Berbasis Mikrokontroler Arduino Mega 2560.*[Skripsi]. Jakarta: Fakultas Tenik, Univeritas Negeri Jakarta.

Hidayat, Q. (2008). *Pengaturan Kecepatan Motor DC dengan Menggunakan Mikrokontroler ATMega 8535*.:2.

ITB, T. P. (2011). *Jurus Kilat Jago Membuat Robot.* Bekasi: Dunia Komputer.

Kadir, A. (2013). *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino.* Yogyakarta: C.V ANDI OFFSET.

Morgalis, M. (2011). *Arduino Cookbook.* Sebastopol: O'Reilly Media, Inc.

Nalwan, A. (2012). *Teknik Rancang Bangun Robot.* Yogyakarta: CV. ANDI OFFSET.

Oktairawan, I., dkk., (2013). *Pembuatan Sistem Otomasi Dispenser Menggunakan*

Mikrokontroler Arduino Mega 2560. :3.

Pitowarno, E. (2006). *Robotika Desain, Kontrol, dan Kecerdasan Buatan.* Yogyakarta: CV. ANDI OFFSET.

Pratama, O. W., dkk., (2014). *Sistem Kendali Gerak Robot Menggunakan PC Berbasis Bluetooth.* :3.

Rachmad, A. (2016). *Rancang Bangun Robot Pemanjat Tiang dan Pemasang Propeller Menggunakan Gripper Elektro Pneumatik sebagai Hybrid Robot pada Kontes Robot Abu Indonesia 2016.* [Skripsi].Jakarta: Fakultas Tenik, Univeritas Negeri Jakarta.

Ramli, S. (2010). *Sistem Manajemen Keselamatan dan Kesehatan Kerja*. Jakarta: DIAN RAKYAT.

Riyadi, R. (2015). *Prototipe Robot Mobil Pengangkut Barang Otomatis dengan Smartphone Android sebagai Pengendali* .[Skripsi]. Jakarta: Fakultas Tenik, Univeritas Negeri Jakarta.

Rudiyanto, H. B. (2009). *Rancang Bangun Robot Pengantar Surat Menggunakan Mikrokontroler At89s51*.: 2.

Sanjaya, M. (2016). *Panduan Praktis Membuat Robot Cerdas Menggunakan Arduino dan Matlab.* Yogyakarta: CV. ANDI OFFSET.

Setiawan, A. (2015). *Robot Pemadam Api dengan Tracking Target Menggunakan Accelerometer Berbasis Mikrokontroler Arduino Due. :*2.

Sofana. (2008). *Membangun Jaringan Komputer.* Bandung: Penerbit Informatika.

Stalings, W. (2007). *Komunikasi dan Jaringan Nirkabel Jilid 2.* Jakarta: Erlangga.

Sulistyo, E. (2014). *Rancang Bangun Robot Pemadam Api Menggunakan Komunikasi I2c.* [Skripsi]. Jakarta: Fakultas Teknik Universitas Negeri Jakarta.

Turang, A. (2015). *Pengembangan Sistem Relay Pengendalian dan Penghematan Pemakaian Lampu Berbasis Mobile.:*4.

LAMPIRAN-LAMPIRAN

PROGRAM ROBOT

```
#include <EEPROM.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <sensor.h>
#include <motor.h>
LiquidCrystal_I2C layar(0x27, 16, 2);
sensor cahaya;
motor motor_kanan;
motor motor_kiri;
Servo servo;
int baca_adc_eeprom[10];
int tulis_adc_eeprom[10];
int error_eeprom[10];
int static error_ke=0;
int motor_pemadam=A10;
int buzzer=A12;
int nilai_buzzer=0;
int jarak_buzzer=1;
int pb[]={46,48,50,52}; //ok,cancel,pilih1,pilih2
int button[4];
int flame=A11;
int flame_sensor;
int gain,adc_ke,kecepatan_home, data_sensor, pwm_kanan,
pwm_kiri,error,last_error,posisi,ruang,d,d1,p,pd,kd;
int delay_belok=500;
const int jarak = 10;
```

```
unsigned long dulu = 0;
boolean menu1=false;
boolean menu_gain=false;
boolean menu_error=false;
boolean menu_konfirmasi1=false;
boolean menu2=false;
boolean menu_scan=false;
boolean menu_mulai=false;
boolean menu_konfirmasi2=false;
boolean menu3=false;
boolean menu_tanya=false;
boolean perempatan=false;
boolean mode_putar_balik=false;
boolean menu4=false;
boolean tanya_scan=false;
boolean menu_tunggu=false;
boolean mode_belok_kanan=false;
boolean pemadam=false;
boolean pertigaan=false;
boolean menu_kd=false;
boolean mode_belok_kiri=false;
boolean koneksi=false;
boolean menu_adc=false;
boolean menu_ubah_error=false;
boolean mode_putar_balik_pulang=false;
boolean mode_mundur_pulang=false;
boolean mode_lurus=false;
boolean menu_awal=false;
```

```
void cek_kondisi()
{
  if(Serial1.available()>0)
  {
    ruang=Serial1.read();
    digitalWrite(buzzer,LOW);
    koneksi=true;
  }
  else
  {
    koneksi =false;
  }
}
void ambil_data()
{
  for(int i=0;i<10;i++)
  {
    baca_adc_eeprom[i]=EEPROM.read(i);
    error_eeprom[i]=EEPROM.read(i+12);
    gain=EEPROM.read(10);
    kecepatan_home=EEPROM.read(11);
    kd=EEPROM.read(23);
  }
}
void setor_data()
{
  for(int i=0;i<10;i++)
  {
```

```
tulis_adc_eeprom[i]=cahaya.nilai_tengah(i)/4;
    EEPROM.update(i,tulis_adc_eeprom[i]);
  EEPROM.update((i+12),error_eeprom[i]);
    EEPROM.update(10,gain);
    EEPROM.update(11,kecepatan_home);
    EEPROM.update(23,kd);
  }
}
void maju()
{
  motor_kanan.maju(190);
  motor_kiri.maju(190);
}
void mundur()
{
  motor_kanan.mundur(150);
  motor_kiri.mundur(150);
}
void mundur_pulang()
{
  motor_kanan.mundur(140);
  motor_kiri.mundur(240);
}
void rem_motor()
{
  motor_kanan.rem();
  motor_kiri.rem();
}
```

```
void belok_kanan()
{
  motor_kanan.rem();
  motor_kiri.maju(230);
}
void belok_kiri()
{
  motor_kanan.maju(230);
  motor_kiri.rem();
}
void putar_balik()
{
  motor_kanan.mundur(180);
  motor_kiri.maju(180);
}
void putar_balik_pulang()
{
  motor_kanan.mundur(150);
  motor_kiri.maju(150);
}
void putar_balikruang()
{
  motor_kanan.mundur(250);
  motor_kiri.maju(250);
}
void lurus()
{
  motor_kanan.maju(200);
```

```
  motor_kiri.maju(200);
}
void putar_balik_ruangan()
{
  maju();
  servo.write(90);
  delay(500);
if(data_sensor==0b0000100000||data_sensor==0b0000010000
)
 putar_balikruang();
 delay(2000);
  menu_mulai=true;
  pertigaan=false;
}
void sudah_pulang()
{
  layar.setCursor(2,0);
  layar.println("Sudah Pulang");
  lurus();
  delay(800);
  putar_balikruang();
  delay(1000);
  layar.clear();
}
void tampilan_sensor()
{
  layar.setCursor(3,0);
  for(int i=0;i<10;i++)
  {
```

```
    if(data_sensor&(0b1000000000>>i))
    {
      layar.write(0xff);
    }
    else
    {
      layar.print("_");
    }
  }
  layar.setCursor(3,1);
  for(int i=0;i<10;i++)
  {
    layar.print(cahaya.digital(i));
  }
}
void kendali()
{
  switch(data_sensor)
  {
    case 0b1000000000  : mode_belok_kiri=true;
menu_mulai=false; break;
    case 0b1110000000  : error=error_eeprom[8]; break;
    case 0b1100000000  : error=error_eeprom[8]; break;
    case 0b0100000000  : error=error_eeprom[7]; break;
    case 0b0111000000  : error=error_eeprom[6]; break;
    case 0b0110000000  : error=error_eeprom[6]; break;
    case 0b0010000000  : error=error_eeprom[5]; break;
    case 0b0011100000  : error=error_eeprom[4]; break;
    case 0b0011000000  : error=error_eeprom[4]; break;
```

```
case 0b0001000000  : error=error_eeprom[3]; break;
case 0b0001100000  : error=error_eeprom[2]; break;
case 0b0000100000  : error=error_eeprom[1]; break;
case 0b0000110000  : error=error_eeprom[0]; break;
case 0b0000010000  : error=-error_eeprom[1]; break;
case 0b0000011000  : error=-error_eeprom[2]; break;
case 0b0000001000  : error=-error_eeprom[3]; break;
case 0b0000001100  : error=-error_eeprom[4]; break;
case 0b0000011100  : error=-error_eeprom[4]; break;
case 0b0000000100  : error=-error_eeprom[5]; break;
case 0b0000000110  : error=-error_eeprom[6]; break;
case 0b0000001110  : error=-error_eeprom[6]; break;
case 0b0000000010  : error=-error_eeprom[7]; break;
case 0b0000000011  : error=-error_eeprom[8]; break;
case 0b0000000111  : error=-error_eeprom[8]; break;
case 0b0000000001  : mode_belok_kanan=true;
menu_mulai=false; break;
case 0b1000000001  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b1000000010  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b0100000001  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b0100000010  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b1100000011  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b1100000001  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
case 0b1100000010  : layar.clear(); pertigaan=true;
menu_mulai=false; break;
```

```
case 0b1000000011  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0100000011  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0010000100  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0100000100  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0010000010  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0001111110  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b0111111000  : layar.clear(); pertigaan=true;
menu_mulai=false; break;

case 0b1100110011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000110001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100110010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100110001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000110010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000110011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100110010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0110110110  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010110100  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010110010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100110100  : layar.clear();
perempatan=true; menu_mulai=false; break;
```

```
case 0b1100010011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000010001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100010010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100010001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000010010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000010011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100010010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0110010110  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010010100  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010010010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100010100  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100100011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000100001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100100010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100100001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000100010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000100011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100100010  : layar.clear();
perempatan=true; menu_mulai=false; break;
```

```
case 0b0110100110  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010100100  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0010100010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100100100  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100011011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000011001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100011010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0100011001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000011010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1000011011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1100011010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1101100011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1001100001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0101100010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b0101100001  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1001100010  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1001100011  : layar.clear();
perempatan=true; menu_mulai=false; break;

case 0b1101100010  : layar.clear();
perempatan=true; menu_mulai=false; break;
```

```
  case 0b1100001011  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1000001001  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b0100001010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b0100001001  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1000001010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1000001011  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1100001010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1101000011  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1001000001  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b0101000010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b0101000001  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1001000010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1001000011  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1101000010  : layar.clear();
  perempatan=true; menu_mulai=false; break;

  case 0b1111111111  : layar.clear(); delay(100);
  sudah_pulang();

  mode_putar_balik_pulang=true;

  menu_mulai=false; break;

 }

p=error*gain;

d1=error-last_error;
```

```
  d=kd*d1;
  pd=p+d;
  pwm_kanan=kecepatan_home+pd;
  pwm_kiri=kecepatan_home-pd;
  if(pwm_kanan>255) pwm_kanan=255;
  if(pwm_kanan<-255) pwm_kanan=-255;
  if(pwm_kiri>255) pwm_kiri=255;
  if(pwm_kiri<-255) pwm_kiri=-255;
  if(pwm_kanan>0) motor_kanan.maju(pwm_kanan);
  if(pwm_kanan<0) motor_kanan.mundur(-(pwm_kanan));
  if(pwm_kiri>0) motor_kiri.maju(pwm_kiri);
  if(pwm_kiri<0) motor_kiri.mundur(-(pwm_kiri));
  last_error=error;
}
void setup()
{
  digitalWrite(motor_pemadam,LOW);
  pinMode(motor_pemadam,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(flame,INPUT);
  digitalWrite(buzzer,LOW);
  Serial1.begin(9600);
  posisi=0;
  cahaya.atur_pin(A0,A1,A2,A3,A4,A5,A6,A7,A8,A9);
  layar.init();
  layar.backlight();
  servo.attach(12);
  for(int i=0;i<4;i++)
```

```
{
  pinMode(pb[i],INPUT_PULLUP);
  button[i]=digitalRead(pb[i]);
}
layar.clear();
while(!tanya_scan)
{
  for(int i=0;i<4;i++)
  {
    pinMode(pb[i],INPUT_PULLUP);
    button[i]=digitalRead(pb[i]);
  }
  ambil_data();
  layar.setCursor(1,0);
  layar.print("Ingin Scaning?");
  layar.setCursor(0,1);
  layar.print("Ya[0]   [1]Tidak");
  if(!button[0])
  {
    delay(150);
    layar.clear();
    menu_awal=false;
    menu_scan=true;
    tanya_scan=true;
  }
  if(!button[1])
  {
    for(int i=0;i<10;i++)
```

```
      {
          baca_adc_eeprom[i]=EEPROM.read(i);
          cahaya.get_nilai_tengah[i]=baca_adc_eeprom[i]
          *4;
      }
      delay(150);
      layar.clear();
      menu_awal=true;
      tanya_scan=true;
    }
  }
}
void loop()
{
  cahaya.atur_pin(A0,A1,A2,A3,A4,A5,A6,A7,A8,A9);
  motor_kanan.atur_pin(6,7,3);
  motor_kiri.atur_pin(4,5,2);
  data_sensor=cahaya.total();
  flame_sensor=analogRead(flame);
  for(int i=0;i<4;i++) button[i]=digitalRead(pb[i]);
  unsigned long sekarang = millis();
  if (sekarang - dulu >= jarak)
  {
    cek_kondisi();
    dulu = sekarang;
    jarak_buzzer++;
    if(jarak_buzzer>50)
    {
      jarak_buzzer=1;
```

```
    nilai_buzzer=1-nilai_buzzer;
  }
}
if(koneksi)
{
  if(ruang!=0 && ruang!=1 && ruang!=2 && ruang!=3)
  {
    layar.clear();
    delay(150);
    layar.setCursor(0,0);
    layar.print("Data yg diterima");
    layar.setCursor(2,1);
    layar.print("Tidak Sesuai");
  }
if(menu_mulai)
  {
    tampilan_sensor();
    kendali();
    layar.setCursor(0,0);
    layar.print("P");
    layar.print(posisi);
    layar.setCursor(14,0);
    layar.print("R");
    layar.print(ruang);
    layar.setCursor(0,1);
    layar.print(pwm_kiri);
    layar.setCursor(13,1);
    layar.print(pwm_kanan);
```

```
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu_tunggu=true;
    menu_mulai=false;
  }
}
else if(menu_tunggu)
{
  layar.setCursor(2,0);
  layar.print("Ruangan Aman");
  layar.setCursor(1,1);
  layar.print("Tidak ada Api");
  if(ruang!=0)
  {
    layar.clear();
    delay(100);
    layar.setCursor(1,0);
    layar.print("Api Terdeteksi");
    maju();
    delay(500);
    layar.clear();
    delay(100);
    menu_mulai=true;
    menu_tunggu=false;
  }
}
```

```
else if(menu1)
{
  layar.setCursor(0,0);
  layar.write(0x7e);
  layar.print("Atur Nilai Gain");
  layar.setCursor(1,1);
  layar.print("Atur Nilai E");
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu_gain=true;
    menu1=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu_awal=true;
    setor_data();
    menu1=false;
  }
  if(!button[2])
  {
    delay(100);
    layar.clear();
    menu4=true;
    menu1=false;
```

```
  }
  if(!button[3])
  {
    delay(100);
    layar.clear();
    menu2=true;
    menu1=false;
  }
}
else if(menu2)
{
  layar.setCursor(1,0);
  layar.print("Atur Nilai Gain");
  layar.setCursor(0,1);
  layar.write(0x7e);
  layar.print("Atur Nilai E");
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu_error=true;
    menu2=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu_awal=true;
```

```
    setor_data();
    menu2=false;
  }
  if(!button[2])
  {
    delay(100);
    layar.clear();
    menu1=true;
    menu2=false;\
  }
  if(!button[3])
  {
    delay(100);
    layar.clear();
    menu3=true;
    menu2=false;
  }
}
else if(menu3)
{
  layar.setCursor(0,0);
  layar.write(0x7e);
  layar.print("Scan Sensor");
  layar.setCursor(1,1);
  layar.print("Cek Nilai ADC");
  if(!button[0])
  {
    delay(100);
```

```
      layar.clear();
      menu_scan=true;
      menu3=false;
    }
    if(!button[1])
    {
      delay(100);
      layar.clear();
      menu_awal=true;
      setor_data();
      menu3=false;
    }
    if(!button[2])
    {
      delay(100);
      layar.clear();
      menu2=true;
      menu3=false;
    }
    if(!button[3])
    {
      delay(100);
      layar.clear();
      menu4=true;
      menu3=false;
    }
  }
  else if(menu4)
```

```
{
  layar.setCursor(1,0);
  layar.print("Scan Sensor");
  layar.setCursor(0,1);
  layar.write(0x7e);
  layar.print("Cek Nilai ADC");
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu_adc=true;
    menu4=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu_awal=true;
    setor_data();
    menu4=false;
  }
  if(!button[2])
  {
    delay(100);
    layar.clear();
    menu3=true;
    menu4=false;
  }
```

```
  if(!button[3])
  {
    delay(100);
    layar.clear();
    menu1=true;
    menu4=false;
  }
}
else if(menu_gain)
{
  layar.setCursor(0,0);
  layar.write(0x7e);
  layar.print("Nilai Gain =");
  layar.print(gain);
  layar.setCursor(1,1);
  layar.print("Nilai Kd = ");
  layar.print(kd);
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu_kd=true;
    menu_gain=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
```

```
    menu1=true;
    menu_gain=false;
  }
  if(gain<0) gain=0;
  if(!button[2])
  {
    delay(100);
    gain-=1;
    layar.clear();
  }
  if(!button[3])
  {
    delay(100);
    gain+=1;
    layar.clear();
  }
}
else if(menu_kd)
{
  layar.setCursor(1,0);
  layar.print("Nilai Gain =");
  layar.print(gain);
  layar.setCursor(0,1);
  layar.write(0x7e);
  layar.print("Nilai Kd = ");
  layar.print(kd);
  if(!button[0])
  {
```

```
    delay(100);
    layar.clear();
    menu_gain=true;
    menu_kd=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu1=true;
    menu_kd=false;
  }
  if(kd<0) kd=0;
  if(!button[2])
  {
    delay(100);
    kd-=1;
    layar.clear();
  }
  if(!button[3])
  {
    delay(100);
    kd+=1;
    layar.clear();
  }
}
```

```
else if(menu_error)
{
  layar.setCursor(0,0);
  layar.write(0x7e);
  layar.print("Nilai E[");
  layar.setCursor(9,0);
  layar.print(error_ke);
  layar.setCursor(11,0);
  layar.print("]=");
  layar.setCursor(13,0);
  layar.print(error_eeprom[error_ke]);
  layar.setCursor(1,1);
  layar.print("Ubah E");
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu_ubah_error=true;
    menu_error=false;
  }
  if(!button[1])
  {
    delay(100);
    layar.clear();
    menu2=true;
    menu_error=false;
  }
  if(error_ke<0) error_ke=9;
```

```
  if(error_ke>9) error_ke=0;
  if(!button[2])
  {
    delay(100);
    error_ke-=1;
    layar.clear();
  }
  if(!button[3])
  {
    delay(100);
    error_ke+=1;
    layar.clear();
  }
}
else if(menu_ubah_error)
{
  layar.setCursor(1,0);
  layar.print("Nilai E[");
  layar.setCursor(9,0);
  layar.print(error_ke);
  layar.setCursor(11,0);
  layar.print("]=");
  layar.setCursor(13,0);
  layar.print(error_eeprom[error_ke]);
  layar.setCursor(0,1);
  layar.write(0x7e);
  layar.print("Ubah E");
```

```
if(!button[0])
{
  delay(100);
  layar.clear();
  menu_error=true;
  menu_ubah_error=false;
}
if(!button[1])
{
  delay(100);
  layar.clear();
  menu2=true;
  menu_error=false;
}
if(error_ke<0) error_ke=9;
if(error_ke>9) error_ke=0;
if(!button[2])
{
  delay(100);
  error_eeprom[error_ke]-=1;
  layar.clear();
}
if(!button[3])
{
  delay(100);
  error_eeprom[error_ke]+=1;
  layar.clear();
}
```

```
}
else if(menu_adc)
{
  layar.setCursor(0,0);
  for(int i=0;i<10;i++)
  {
    if(data_sensor&(0b1000000000>>i))
    {
      layar.write(0xff);
    }
    else
    {
      layar.print("_");
    }
  }
  layar.setCursor(adc_ke,1);
  layar.write(0x5e);
  layar.print(cahaya.adc(adc_ke));
  layar.setCursor(15,1);
  layar.print(cahaya.digital(adc_ke));
  layar.setCursor(11,0);
  layar.print(cahaya.nilai_tengah(adc_ke));
  if(adc_ke<0) adc_ke=0;
  if(adc_ke>9) adc_ke=9;
  if(!button[1])
  {
    delay(100);
    layar.clear();
```

```
    menu4=true;
    menu_adc=false;
  }
  if(!button[2])
  {
    delay(100);
    adc_ke-=1;
    layar.clear();
  }
  if(!button[3])
  {
    delay(100);
    adc_ke+=1;
    layar.clear();
  }
}
else if(menu_scan)
{
  layar.setCursor(1,0);
  layar.println("Proses Scaning");
  cahaya.scan(150,800);
  layar.clear();
  layar.setCursor(1,0);
  layar.println("Scaning Selesai");
  delay(1000);
  layar.clear();
  menu_tanya=true;
  menu_scan=false;
```

```
}
else if(menu_tanya)
{
  layar.setCursor(1,0);
  layar.print("Apa anda yakin");
  layar.setCursor(0,1);
  layar.print("Ingin save data?");
 if(!button[0]||!button[1]||!button[2]||!button[3])
  {
    delay(100);
    layar.clear();
    menu_konfirmasi1=true;
    menu_tanya=false;
  }
}
else if(menu_konfirmasi1)
{
  layar.setCursor(0,0);
  layar.write(0x7e);
  layar.print("Ya");
  layar.setCursor(11,0);
  layar.print("Tidak");
  if(!button[0])
  {
    setor_data();
    delay(100);
    layar.clear();
    menu3=true;
    menu_konfirmasi1=false;
```

```
  }
  if(!button[3])
  {
    delay(100);
    layar.clear();
    menu_konfirmasi2=true;
    menu_konfirmasi1=false;
  }
}
else if(menu_konfirmasi2)
{
  layar.setCursor(1,0);
  layar.print("Ya");
  layar.setCursor(10,0);
  layar.write(0x7e);
  layar.print("Tidak");
  if(!button[0])
  {
    delay(100);
    layar.clear();
    menu3=true;
    menu_konfirmasi2=false;
  }
  if(!button[2])
  {
    delay(100);
    layar.clear();
    menu_konfirmasi1=true;
```

```
    menu_konfirmasi2=false;
  }
}
else if(pertigaan)
{
  rem_motor();
  layar.setCursor(0,0);
  layar.print("Mode Pertigaan");
  layar.setCursor(0,1);
  layar.print("Berhentiii");
  delay(500);
  pemadam=true;
  pertigaan=false;
}
else if(perempatan)
{
  layar.setCursor(0,0);
  layar.print("Mode Peremapatan");
  layar.setCursor(3,1);
  switch(ruang)
  {
   case 0: switch(posisi){
           case 0: layar.println("Putar Balik");
           posisi=0; putar_balik(); delay(1000);
           mode_putar_balik=true; break;
           case 1: layar.println("Belok Kanan");
           posisi=0; belok_kanan(); delay(500);
           mode_belok_kanan=true; break;
           case 2: layar.println("Lurus");
           posisi=0; lurus(); delay(500);
           menu_mulai=true; break;
```

```
      case 3: layar.println("Belok Kiri");
      posisi=0; belok_kiri(); delay(500);
      mode_belok_kiri=true; break;

     }

     break;

case 1: switch(posisi){

      case 0: layar.println("Belok Kiri");
      posisi=1; belok_kiri();
      delay(500);mode_belok_kiri=true; break;

      case 1: layar.println("Putar Balik");
      posisi=1; putar_balik(); delay(1000);
      mode_putar_balik=true; break;

      case 2: layar.println("Belok Kanan");
      posisi=1; belok_kanan(); delay(500);
      mode_belok_kanan=true; break;

      case 3: layar.println("Lurus");
      posisi=1; lurus(); delay(500);
      menu_mulai=true; break;

      }

      break;

case 2: switch(posisi){

      case 0: layar.println("Lurus");
      posisi=2; lurus(); delay(500);
      menu_mulai=true; break;
      case 1: layar.println("Belok Kiri");
      posisi=2; belok_kiri(); delay(500);
      mode_belok_kiri=true; break;

      case 2: layar.println("Putar Balik");
      posisi=2; putar_balik(); delay(1000);
      mode_putar_balik=true; break;

      case 3: layar.println("Belok Kanan");
      posisi=2; belok_kanan(); delay(500);
      mode_belok_kanan=true; break;

      }

      break;

case 3: switch(posisi){
```

```
            case 0: layar.println("Belok Kanan");
            posisi=3; belok_kanan(); delay(500);
            mode_belok_kanan=true; break;

            case 1: layar.println("Lurus");
            posisi=3; lurus(); delay(500);
            menu_mulai=true; break;

            case 2: layar.println("Belok Kiri");
            posisi=3; belok_kiri(); delay(500);
            mode_belok_kiri=true; break;

            case 3: layar.println("Putar Balik");
            posisi=3; putar_balik(); delay(1000);
            mode_putar_balik=true; break;

            }

            break;

      default: layar.println("Tidak Cocok");
      delay(3000); break;

    }

    layar.clear();

    perempatan=false;

}

else if(pemadam)

{

    layar.setCursor(2,0);

    layar.print("Mode Pemadam");

    layar.setCursor(0,1);

    layar.print("Cari Apiii");

    delay(300);

    if(flame_sensor<300)

    {

      digitalWrite(motor_pemadam,HIGH);

      for(int i=0;i<7;i++)

      {
```

```
      servo.write(70);
      delay(200);
      servo.write(110);
      delay(200);
    }
    servo.write(90);
    digitalWrite(motor_pemadam,LOW);
  }
  else
  {
    putar_balik();
    delay(delay_belok);
    mode_putar_balik=true;
    pemadam=false;
  }
}
    else if(menu_awal)
{
  tampilan_sensor();
  motor_kanan.diam();
  motor_kiri.diam();
  layar.setCursor(1,0);
  layar.print(ruang);
  layar.setCursor(13,0);
  layar.print(kecepatan_home);
  if(!button[0])
  {
    delay(100);
```

```
  layar.clear();
  menu1=true;
  ambil_data();
  menu_awal=false;
}
if(!button[1])
{
  delay(100);
  layar.clear();
  maju();
  delay(500);
  menu_tunggu=true;
  menu_awal=false;
}
if(!button[2])
{
  delay(50);
  layar.clear();
  ruang-=1;
  kecepatan_home-=5;
  EEPROM.update(11,kecepatan_home);
}
if(!button[3])
{
  delay(50);
  layar.clear();
  ruang+=1;
  kecepatan_home+=5;
```

```
    EPROM.update(11,kecepatan_home);
  }
  if(kecepatan_home<0) kecepatan_home=0;
  if(kecepatan_home>255) kecepatan_home=255;
  if(ruang<0) ruang=0;
  if(ruang>3) ruang=3;
}
else if(mode_putar_balik)
  {
  if((data_sensor==0b0000001100)||(data_sensor==0b0
  000001000))
  {
    menu_mulai=true;
    mode_putar_balik=false;
  }
  else
  {
    putar_balik();
  }
}
else if(mode_putar_balik_pulang)
  {
  if((data_sensor==0b0001000000)||(data_sensor==0b0
  000110000)||(data_sensor==0b0000100000))
  {
    rem_motor();
    delay(300);
    mode_mundur_pulang=true;
    mode_putar_balik_pulang=false;
  }
```

```
  else
  {
    putar_balik_pulang();
  }
}
else if(mode_belok_kiri)
  {
  if((data_sensor==0b0001100000)||(data_sensor==0b0
  000100000))
  {
    menu_mulai=true;
    mode_belok_kiri=false;
  }
  else
  {
    belok_kiri();
  }
}
else if(mode_belok_kanan)
{
if((data_sensor==0b0000010000)||(data_sensor==0b000
0011000))
  {
    menu_mulai=true;
    mode_belok_kanan=false;
  }
  else
  {
    belok_kanan();
  }
```

```
}
else if(mode_mundur_pulang)
{
  if(data_sensor==0b1111111111)
  {
    rem_motor();
    delay(100);
    menu_tunggu=true;
    mode_mundur_pulang=false;
  }
  else
  {
    mundur_pulang();
  }
}
else if(mode_lurus)
   {
   if((data_sensor==0b0000000001)||(data_sensor==0b
   0000000010)||(data_sensor==0b0000000100)||(data_
   sensor==0b0000001000)||(data_sensor==0b000001000
   0)||(data_sensor==0b0000100000)||(data_sensor==0
   b0001000000)||(data_sensor==0b0010000000)||(data
   _sensor==0b0100000000)||(data_sensor==0b10000000
   00))
  {
    menu_mulai=true;
    mode_lurus=false;
  }
  else
  {
    lurus();
```

```
      }
    }
}

  else
  {
      layar.setCursor(0,0);
      layar.print("Komunikasi Gagal");
      layar.setCursor(0,1);
      layar.print("Cek Wiring&Reset");
      digitalWrite(buzzer,nilai_buzzer);
  }
}
```

PROGRAM MONITORING RUANGAN

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(2,3);
int sensor[3]={A0,A1,A2};
int sekarang=0,selanjutnya1,selanjutnya2;
int batas_sensor[3]={700,700,700};
int kondisi[3];
unsigned long kini,lalu,jarak;
void setup()
{
  for(int i=0;i<3;i++)
  {
    pinMode(sensor[i],INPUT);
  }
  bluetooth.begin(9600);
}
void loop()
{
  kini=millis();
  jarak=kini-lalu;
  if(jarak>=10)
  {
    lalu=kini;
    cek_kondisi();
  }
  for(int i=0;i<3;i++)
  {
```

```
    kondisi[i]=analogRead(sensor[i]);
  }
  if(kondisi[0]<batas_sensor[0])
  {
    if(sekarang==0)    sekarang=1;
    if(sekarang!=0 && selanjutnya1==0 && sekarang!=1)
    selanjutnya1=1;
    if(sekarang!=0 && selanjutnya1!=0&&selanjutnya1!=1
    && sekarang!=1) selanjutnya2=1;
  }
  if(kondisi[1]<batas_sensor[1])
  {
    if(sekarang==0)    sekarang=2;
    if(sekarang!=0 && selanjutnya1==0&& sekarang!=2)
    selanjutnya1=2;
    if(sekarang!=0 && selanjutnya1!=0&&selanjutnya1!=2
    && sekarang!=2) selanjutnya2=2;
  }
  if(kondisi[2]<batas_sensor[2])
  {
    if(sekarang==0)    sekarang=3;
    if(sekarang!=0 && selanjutnya1==0&& sekarang!=3)
    selanjutnya1=3;
    if(sekarang!=0 && selanjutnya1!=0&&selanjutnya1!=3
    && sekarang!=3) selanjutnya2=3;
  }
  bluetooth.write(sekarang);
}
void cek_kondisi()
{
```

```
  if((kondisi[0]>batas_sensor[0])&&sekarang==1)
  sekarang=0;

  if((kondisi[1]>batas_sensor[1])&&sekarang==2)
  sekarang=0;

  if((kondisi[2]>batas_sensor[2])&&sekarang==3)
  sekarang=0;

  if((kondisi[0]>batas_sensor[0])&&selanjutnya1==1)
  selanjutnya1=0;

  if((kondisi[1]>batas_sensor[1])&&selanjutnya1==2)
  selanjutnya1=0;

  if((kondisi[2]>batas_sensor[2])&&selanjutnya1==3)
  selanjutnya1=0;

  if((kondisi[0]>batas_sensor[0])&&selanjutnya2==1)
  selanjutnya2=0;

  if((kondisi[1]>batas_sensor[1])&&selanjutnya2==2)
  selanjutnya2=0;

  if((kondisi[2]>batas_sensor[2])&&selanjutnya2==3)
  selanjutnya2=0;

  if(sekarang==0)

  {

    sekarang=selanjutnya1;

    selanjutnya1=selanjutnya2;

    selanjutnya2=0;

  }

  if(sekarang!=0 && selanjutnya1==0)selanjutnya2=0;

}
```

MOTOR.H

```
#ifndef _motor_h
#define _motor_h
class motor{
  private:
    int digital1,digital2,pwm;
  public:
    motor();
    ~motor();
    int atur_pin(int,int,int);
    void diam();
    int maju(int);
    int mundur(int);
    void rem();
};
#endif
```

MOTOR.CPP

```
#include <Arduino.h>
#include <motor.h>
motor::motor()
{
}
motor::~motor()
{
}
int motor::atur_pin(int pin_digital1,int pin_digital2,
int pin_pwm)
{
  digital1=pin_digital1;
  digital2=pin_digital2;
  pwm=pin_pwm;
  pinMode(digital1,OUTPUT);
  pinMode(digital2,OUTPUT);
  pinMode(pwm,OUTPUT);
}
void motor::diam()
{
  digitalWrite(digital1,LOW);
  digitalWrite(digital2,LOW);
  analogWrite(pwm,0);
}

int motor::maju(int nilai_pwm_maju)
{
  digitalWrite(digital1,HIGH);
```

```
  digitalWrite(digital2,LOW);
  analogWrite(pwm,nilai_pwm_maju);
}

int motor::mundur(int nilai_pwm_mundur)
{
  digitalWrite(digital1,LOW);
  digitalWrite(digital2,HIGH);
  analogWrite(pwm,nilai_pwm_mundur);
}

void motor::rem()
{
  digitalWrite(digital1,HIGH);
  digitalWrite(digital2,HIGH);
  analogWrite(pwm,255);
}
```

SENSOR.H

```
#ifndef _sensor_h
#define _sensor_h
class sensor{
  private:
  int pin[10];
  int get_adc[10];
  int get_adc_low[10];
  int get_adc_high[10];
  int get_digital[10];
  public:
  int get_nilai_tengah[10];
  sensor();
  ~sensor();
  int atur_pin(int,int,int,int,int,int,int,int,int,int);
  int scan(int,int);
  int adc(int);
  int digital(int);
  int nilai_tengah(int);
  int total();
};
#endif
```

SENSOR.CPP

```
#include <Arduino.h>

#include <sensor.h>

sensor::sensor()

{

}

sensor::~sensor()

{

}

int sensor::atur_pin(int pin1,int pin2,
int pin3, int pin4, int pin5, int pin6,
int pin7, int pin8, int pin9, int pin10)

{

 pin[0]=pin1;

 pin[1]=pin2;

 pin[2]=pin3;

 pin[3]=pin4;

 pin[4]=pin5;

 pin[5]=pin6;

 pin[6]=pin7;

 pin[7]=pin8;

 pin[8]=pin9;

 pin[9]=pin10;

 for(int i=0;i<10;i++)

 {

  pinMode(pin[i],INPUT);

  get_adc[i]=analogRead(pin[i]);

  if(get_adc[i]>=get_nilai_tengah[i])

  {
```

```
    get_digital[i]=1;
  }
  else
  {
    get_digital[i]=0;
  }
 }
}
int sensor::scan(int nilai_tengah_sementara,int n)
{

  for(int i=0;i<n;i++)
  {
    Serial.println("Masih Scan");
    for(int a=0;a<10;a++)
    {
      get_adc[a]=analogRead(pin[a]);
      if(get_adc[a]<nilai_tengah_sementara)
      {
        get_adc_low[a]=get_adc[a];
      }
      else if(get_adc[a]>=nilai_tengah_sementara)
      {
        get_adc_high[a]=get_adc[a];
      }
get_nilai_tengah[a]=(get_adc_low[a]+
get_adc_high[a])/2;
    }
  }
}
```

```
int sensor::adc(int adc_ke)
{
  return get_adc[adc_ke];
}
int sensor::digital(int digital_ke)
{
  return get_digital[digital_ke];
}
int sensor::nilai_tengah(int tengah_ke)
{
  return get_nilai_tengah[tengah_ke];
}
int sensor::total()
{
int bit_sensor=0;
for(int i=0;i<11;i++)
{
  if(get_digital[i]>0)
  bit_sensor=bit_sensor|(0b1000000000>>i);
}
return bit_sensor;
}
```

DATA RIWAYAT HIDUP

Badrut Tamam. Lahir di Cirebon, 30 April 1996 anak kelima dari delapan bersaudara dari pasangan Khafidz dan Nihayah. Bertempat tinggal di Desa Kalirahayu, Blok Madrasah Rt. 08 Rw. 02 Kecamatan Losari, Kabupaten Cirebon. Penulis menyelesaikan pendidikan formal dimulai dari Pendidikan Sekolah Dasar di MI Miftahul Huda 01 pada tahun 2000 dan lulus pada tahun 2007. Kemudian melanjutkan ke Pendidikan Sekolah Menengah Pertama di SMP Negeri 1 Losari pada tahun 2007 dan lulus pada tahun 2010. Penulis meneruskan ke Pendidikan Sekolah Menengah Atas di SMA Negeri 1 Losari pada tahun 2010 dan lulus pada tahun 2013. Setelah Tamat SMA penulis melanjutkan pendidikan ke Universitas Negeri Jakarta diterima setelah melaksanakan Ujian SBMPTN pada tahun 2013 untuk Program Studi Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Jakarta.