



TUGAS AKHIR - TE141599

**SWARM ROBOT MENGGUNAKAN SISTEM KOORDINASI
SUARA UNTUK Mencari SUMBER GAS**

**Putu Agus Antara Adiputra
NRP 2212100057**

**Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.**

**JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT - TE141599

**SWARM ROBOT USING SOUND COORDINATION
SYSTEM FOR FINDING GAS RESOURCES**

**Putu Agus Antara Adiputra
NRP 2212100057**

**Supervisor
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.**

**ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul ***Swarm Robot Menggunakan Sistem Koordinasi Suara untuk Mencari Sumber Gas*** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 19 Januari 2017

Putu Agus Antara Adiputra
NRP. 2212100057

**SWARM ROBOT MENGGUNAKAN SISTEM KOORDINASI
SUARA UNTUK Mencari SUMBER GAS**

TUGAS AKHIR

**Diajukan Untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Teknik Elektronika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Muhammad Rivai, ST., MT.

NIP: 196904261994031003

Fajar Budiman, ST., M.Sc.

NIP: 198607072014041001



***SWARM ROBOT* MENGGUNAKAN SISTEM KOORDINASI SUARA UNTUK Mencari SUMBER GAS**

Nama : Putu Agus Antara Adiputra
Pembimbing I : Dr. Muhammad Rivai, ST., MT.
Pembimbing II : Fajar Budiman, ST., M.Sc.

ABSTRAK

Dalam mencari sumber kebocoran gas, salah satu kendala yang menghambat proses pencarian adalah angin. Menggunakan satu buah robot dirasa masih sulit dan membutuhkan waktu yang lama. Maka dari itu diperlukan beberapa robot yang saling bekerja sama membentuk sebuah barisan untuk memudahkan mengikuti aliran gas yang terpengaruh angin. *Swarm robot* mengadaptasi perilaku dari serangga koloni yang mampu berkomunikasi dan bekerja sama melakukan sesuatu, contohnya mencari makanan. Dengan komunikasi dan kerjasama antar beberapa robot dapat mempermudah proses mencari sumber kebocoran gas, sehingga waktu yang diperlukan semakin singkat.

Swarm robot yang dirancang menggunakan Arduino UNO sebagai kontrol utama, modul HC-SR04, sensor TGS2600, *buzzer*, mikrofon kondenser, dan LCD. Dalam mendeteksi gas menerapkan prinsip *stereo nose* yakni dua sensor gas TGS2600 dipasang pada bagian kanan dan kiri depan robot untuk dapat mengetahui arah dari sumber gas. Semakin besar intensitas gas yang diterima maka jumlah sinyal *beep* yang dikirim semakin banyak. Empat buah mikrofon dan rangkaian pengolah sinyal dipasang pada arah berbeda yang bertujuan untuk mencari arah sumber suara dengan memanfaatkan selisih nilai tegangan puncak sinyal. Robot yang membaca intensitas gas tertinggi (*master*) berperan menuntun robot *slave* untuk mendekati ke arah lokasi dengan intensitas gas lebih tinggi.

Hasil pengujian pada tugas akhir ini adalah dua buah *swarm robot* dengan kemampuan mendeteksi gas hingga jarak 150 cm. Sensitivitas sensor gas dipengaruhi oleh jenis objek gas yang dideteksi. Komunikasi suara antar 2 robot memiliki jarak optimum hingga 1,5 meter. Kedepannya agar tujuan utama dari sistem dapat tercapai maka perlu menggunakan lebih dari 2 robot.

Kata kunci: kebocoran gas, komunikasi suara, *stereo nose*, *swarm robot*

Halaman ini sengaja dikosongkan

SWARM ROBOT USING SOUND COORDINATION SYSTEM FOR FINDING GAS RESOURCES

Name : Putu Agus Antara Adiputra
Supervisor : Dr. Muhammad Rivai, ST., MT.
Co-Supervisor : Fajar Budiman, ST., M.Sc.

ABSTRACT

In searching for gas leakage, one obstacle which detains in the gas finding process is wind. It would be difficult and time consuming in using only one robot. Therefore, it is required some robots that are capable to work together to ease the search and efficiently shorten the time. Swarm robots adapt the behavior of insect colonies with small physical form that is able to communicate and work together to do something, for example, search for food. With the communication and cooperation between several robots can simplify the process of searching for the source of the gas leak, so the required time become shorter.

The designed swarm robots use Arduino UNO as the main controller, HC-SR04 module, TGS2600 sensor, buzzer, electret microphone, and LCD. Stereo nose principles is applied in detecting gas, with two similar gas sensor installed in the front right and front left of the robot to know the direction of the gas source. The greater the intensity of the gas amount received the more *beep* are sent. Four electret microphone are mounted in different directions that aim to search for the sound source direction by utilizing the difference in measurement value of the peak voltage. Robots that reads the intensity of the highest acts as a leader (as a master) to guide the other robot (acting as a slave) to move closer into locations with higher gas intensity.

The test results in this project are two swarm robots with maximum range of gas detection as far as 150 cm. Sensor sensitivity affected by the type of detected gas object. Sound communication system between 2 robots has optimum range up to 1.5 meters. For further development need to use more than 2 robots to accomplish the main purpose.

Keywords : gas leakage, sound communication, stereo nose, swarm robot

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas anugerah-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul ***Swarm Robot Menggunakan Sistem Koordinasi Suara untuk Mencari Sumber Gas.***

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Dr.Eng. Ardyono Priyadi, S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
2. Ir. Tasripan, MT. selaku Koordinator Bidang Studi Elektronika.
3. Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing 1 atas bimbingan dan bantuan alat dalam merealisasikan tugas akhir ini.
4. FajarBudiman, ST., M.Sc. selaku dosen pembimbing 2 atas motivasi dan saranyang sangat membantu penulis dalam melaksanakan tugas akhir.
5. Dr. Ir. Hendra Kusuma, M.Eng.Sc., Ir. Harris Pirngadi, MT., Ir. Tasripan, MT. dan Astria Nur Irfansyah, ST., M.Eng., Ph.D. selaku dosen penguji atas materi tambahan yang sangat berguna untuk pengembangan tugas akhir ini.
6. Seluruh dosen bidang studi elektronika.
7. Bapak, Ibu, adik serta seluruh keluarga yang memberikan dukungan material maupun motivasi dan semangat.
8. Gusti Ayu Khrisna Saraswati sebagai sosok perempuan yang selalu menginspirasi dan memotivasi dari awal hingga akhir pembuatan tugas akhir.
9. Teman-teman laboratorium Elektronika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut. Penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak.

Surabaya, 3 Januari 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

| | |
|---|-----|
| ABSTRAK..... | i |
| <i>ABSTRACT</i> | iii |
| KATA PENGANTAR | v |
| DAFTAR ISI..... | vii |
| DAFTAR GAMBAR | ix |
| DAFTAR TABEL..... | xi |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Perumusan Masalah | 2 |
| 1.3. Tujuan | 2 |
| 1.4. Batasan Masalah | 2 |
| 1.5. Metodologi | 2 |
| 1.6. Sistematika Penulisan..... | 3 |
| 1.7. Relevansi dan Manfaat..... | 3 |
| BAB II TINJAUAN PUSTAKA..... | 5 |
| 2.1. <i>Swarm Robot</i> | 5 |
| 2.2. Arduino UNO..... | 7 |
| 2.2.1. Sumber Daya / <i>Power</i> | 8 |
| 2.2.2. Memori..... | 8 |
| 2.2.3. Input dan Output | 8 |
| 2.2.4. Komunikasi | 9 |
| 2.2.5. Perlindungan Arus USB | 9 |
| 2.3. Sensor Gas TGS2600 | 10 |
| 2.4. Mikrofon Kondenser | 12 |
| 2.5. Sensor Ultrasonik..... | 13 |
| 2.6. <i>Buzzer</i> | 14 |
| 2.7. Driver Motor | 16 |
| 2.8. Motor DC..... | 18 |
| 2.9. Multiplexer | 21 |
| 2.10. Rangkaian Osilator..... | 21 |
| 2.11. <i>Liquid Crystal Display (LCD)</i> | 23 |
| BAB III PERANCANGAN SISTEM | 27 |
| 3.1. Perancangan Perangkat Keras | 28 |
| 3.1.1. Perangkat Pendeteksi Gas | 28 |
| 3.1.2. Perangkat Komunikasi | 30 |
| 3.1.3. Perangkat Pendeteksi Jarak | 37 |
| 3.1.4. Perangkat Gerak Robot | 38 |

| | | |
|--|---|----|
| 3.1.5. | Perangkat Tambahan..... | 39 |
| 3.2. | Perancangan Perangkat Lunak..... | 42 |
| 3.2.1. | Proses Komunikasi dan Penentuan <i>Master-Slave</i> | 43 |
| 3.2.2. | Penentuan Gerak | 46 |
| BAB IV PENGUJIAN DAN ANALISIS DATA | | 49 |
| 4.1. | Pengujian Sensor Gas | 49 |
| 4.2. | Pengujian Perangkat Komunikasi..... | 52 |
| 4.3. | Pengujian Sensor Jarak | 58 |
| 4.4. | Pengujian Gerak Robot <i>Master</i> | 60 |
| 4.5. | Pengujian Gerak Robot <i>Slave</i> | 61 |
| 4.6. | Pengujian Gerak Robot Keseluruhan | 62 |
| BAB V KESIMPULAN DAN SARAN | | 65 |
| 5.1. | Kesimpulan..... | 65 |
| 5.2. | Saran | 65 |
| DAFTAR PUSTAKA | | 67 |
| LAMPIRAN | | 69 |
| BIODATA PENULIS | | 85 |

DAFTAR GAMBAR

| | | |
|--------------------|--|----|
| Gambar 2.1 | <i>Swarm robot</i> | 6 |
| Gambar 2.2 | Pinout Arduino UNO | 8 |
| Gambar 2.3 | Potensial penghalang (eVs) tinggi saat tidak mendeteksi gas | 10 |
| Gambar 2.4 | Potensial penghalang (eVs) rendah saat sensor mendeteksi gas | 11 |
| Gambar 2.5 | Rangkaian dasar pengukuran sensor | 11 |
| Gambar 2.6 | Skema dari mikrofon kondenser | 12 |
| Gambar 2.7 | Prinsip kerja sensor ultrasonik | 13 |
| Gambar 2.8 | Timing diagram dari sensor HC-SR04..... | 14 |
| Gambar 2.9 | Buzzer piezokeramik..... | 15 |
| Gambar 2.10 | Cara kerja buzzer piezokeramik..... | 15 |
| Gambar 2.11 | Cara kerja buzzer magnetik..... | 16 |
| Gambar 2.12 | Rangkaian dasar driver motor tipe H-Bridge | 17 |
| Gambar 2.13 | Struktur motor DC | 18 |
| Gambar 2.14 | Pengaturan lebar pulsa (<i>duty cycle</i>) untuk mengontrol posisi servo..... | 20 |
| Gambar 2.15 | Pinout multiplexer CD4051 | 21 |
| Gambar 2.16 | Pinout dari IC 555 | 22 |
| Gambar 2.17 | Rangkaian osilator dengan <i>duty cycle</i> 50%..... | 23 |
| Gambar 2.18 | Diagram blok LCD..... | 25 |
| Gambar 3.1 | Diagram blok <i>swarm robot</i> | 27 |
| Gambar 3.2 | Ilustrasi <i>swarm robot</i> yang dirancang | 28 |
| Gambar 3.3 | Rangkaian pengukuran sensor TGS2600 | 29 |
| Gambar 3.4 | Rangkaian kalibrasi sensor TGS2600 kiri dan kanan .. | 30 |
| Gambar 3.5 | Diagram blok sistem komunikasi suara | 31 |
| Gambar 3.6 | Skematik rangkaian osilator dengan <i>duty cycle</i> 50% ... | 32 |
| Gambar 3.7 | Timing diagram antara output pin 2 Arduino dengan output rangkaian osilator..... | 33 |
| Gambar 3.8 | Skematik rangkaian penerima | 34 |
| Gambar 3.9 | Efek dari nilai resistor dan kapasitor terhadap bentuk selubung | 36 |
| Gambar 3.10 | Tata letak pemasangan perangkat penerima pada robot..... | 36 |
| Gambar 3.11 | Skema antara Arduino, motor servo dan HC-SR04 | 37 |
| Gambar 3.12 | Skema dari perangkat gerak | 39 |

| | | |
|--------------------|--|----|
| Gambar 3.13 | Rangkaian skematik LCD I2C menggunakan PCF8574 | 40 |
| Gambar 3.14 | Rangkaian skematik multiplekser CD4051..... | 41 |
| Gambar 3.15 | Diagram alir pemrograman <i>swarm robot</i> | 42 |
| Gambar 3.16 | Pemetaan penggunaan pin Arduino UNO..... | 43 |
| Gambar 3.17 | Diagram alir proses komunikasi dan penentuan <i>master-slave</i> | 44 |
| Gambar 3.20 | Diagram alir gerak robot <i>master</i> | 46 |
| Gambar 3.21 | Diagram alir gerak robot <i>slave</i> | 47 |
| Gambar 4.1 | Bahan pengujian sensor TGS2600 (Alkohol 96%, spiritus dan bensin) | 49 |
| Gambar 4.2 | Pengujian pembacaan gas terhadap jarak | 51 |
| Gambar 4.3 | Grafik hasil pengujian pengaruh jarak terhadap pembacaan gas | 52 |
| Gambar 4.4 | Bentuk gelombang output dari osilator..... | 53 |
| Gambar 4.5 | Bentuk sinyal output yang telah dikontrol Arduino..... | 53 |
| Gambar 4.6 | Perbandingan antara sinyal input dan output pada frekuensi cutoff BPF..... | 55 |
| Gambar 4.7 | Bentuk sinyal diterima pada jarak 25 cm..... | 56 |
| Gambar 4.8 | Bentuk sinyal yang diterima pada jarak 100 cm..... | 57 |
| Gambar 4.9 | Bentuk sinyal yang diterima pada jarak 175 cm..... | 57 |
| Gambar 4.10 | Pengujian terhadap sudut pembacaan 20 derajat | 58 |
| Gambar 4.11 | Pengujian terhadap sudut pembacaan 30 derajat | 59 |
| Gambar 4.12 | Sensor gas kanan robot mendapat intensitas lebih besar, robot berbelok ke kanan | 60 |
| Gambar 4.13 | Robot berhasil mendekati sumber gas | 60 |
| Gambar 4.14 | Robot <i>slave</i> berada pada range komunikasi dan berusaha mendekati robot <i>master</i> | 61 |
| Gambar 4.15 | Robot <i>slave</i> semakin mendekati robot <i>master</i> | 62 |
| Gambar 4.16 | Posisi awal pengujian robot, robot A (lampu hijau) dan robot B (lampu biru) | 63 |
| Gambar 4.17 | Robot B diberikan aliran gas dan berperan sebagai <i>master</i> | 63 |
| Gambar 4.18 | Robot A diberikan aliran gas, terjadi pergantian mode dari <i>slave</i> ke <i>master</i> | 64 |

DAFTAR TABEL

| | | |
|------------------|---|----|
| Tabel 2.1 | Spesifikasi Arduino UNO | 7 |
| Tabel 2.2 | Spesifikasi sensor HC-SR04 | 13 |
| Tabel 3.1 | Pengelompokan intensitas gas berdasarkan pembacaan nilai ADC | 44 |
| Tabel 4.1 | Pengujian respon sensor TGS2600..... | 50 |
| Tabel 4.2 | Pengujian pengaruh jarak terhadap pembacaan gas | 51 |
| Tabel 4.3 | Respon frekuensi BPF terhadap tegangan..... | 54 |
| Tabel 4.4 | Pengaruh jarak terhadap pembacaan sinyal..... | 56 |
| Tabel 4.5 | Pengaruh sudut pada pembacaan jarak..... | 58 |

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kebocoran gas merupakan awal dari terjadinya bencana mematikan bagi lingkungan sekitarnya. Banyak kendala yang dihadapi ketika mendeteksi sumber kebocoran gas salah satunya adalah pengaruh angin. Perlu waktu yang lama untuk menentukan kebocoran pipa gas yang panjangnya dapat mencapai ratusan kilometer dengan mengandalkan pengelihatian maupun penciuman manusia. Disamping itu gas yang bocor dapat berbahaya apabila dihirup oleh manusia dan juga mudah terbakar apabila terkena percikan api.

Berkembangnya teknologi memungkinkan robot untuk membantu pekerjaan manusia. Keunggulan utama dari penggunaan robot ialah dari segi kepraktisan dan untuk meminimalisir resiko yang terjadi akibat kebocoran gas itu sendiri karena robot dapat dikontrol dari jauh ataupun bergerak sendiri menjelajahi medan. Selain itu robot yang dirancang khusus mampu melalui area yang mungkin sulit untuk dilalui oleh manusia. Namun, satu robot mencakup area yang luasnya sebanding dengan panjang pipa gas membutuhkan waktu yang lama hingga menemukan lokasi sumber kebocoran gas. Dan terkadang pendeteksian gas terhambat oleh aliran angin sehingga robot sulit untuk membaca arah kebocoran gas. Maka dari itu perlu adanya robot pembantu lain untuk mencakup sebagian dari luas area sehingga kerja masing-masing robot lebih mudah dan lebih efisien terhadap waktu.

Swarm intelligence adalah suatu cabang kecerdasan buatan yang memungkinkan robot-robot berperilaku seperti satu koloni serangga yang saling bekerja sama untuk melakukan suatu pekerjaan, misalkan untuk mencari makan atau mengangkut makanan yang lebih besar [1]. Penggunaan konsep *swarm intelligence* bertujuan untuk membuat prototipe 2 buah *swarm robot* yang dapat membentuk sebuah barisan untuk menanggulangi pendeteksian gas akibat pengaruh angin. Contoh ilustrasi kasus, dua buah *swarm robot* (robot A dan B) sedang mencari sumber gas dan robot A memperoleh intensitas gas yang lebih besar. Robot B akan mengikuti robot A yang mendeteksi gas lebih kuat. Tiba-tiba arah angin berubah dan pembacaan gas terkuat terjadi pada robot B, sehingga robot A akan bergerak mendekati robot B. Dengan begitu arah aliran gas akan tetap terdeteksi hingga robot dapat mencapai sumber gas.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana *swarm robot* mendeteksi sumber gas.
2. Bagaimana *swarm robot* dapat berkomunikasi dan mengetahui posisi robot lainnya.

1.3. Tujuan

Tujuan yang ingin dicapai dalam perancangan ini adalah:

1. *Swarm robot* dapat mendeteksi lokasi dari sumber gas menggunakan prinsip *stereo nose*.
2. *Swarm robot* berkomunikasi dan dapat mengetahui posisi robot lainnya dengan menggunakan gelombang suara.

1.4. Batasan Masalah

Dalam penulisan tugas akhir ini digunakan metodologi sebagai berikut:

1. *Swarm robot* yang dibuat sebanyak dua robot.
2. Sensor gas hanya mampu mendeteksi gas yang mudah terbakar.
3. Mikrofon yang digunakan adalah mikrofon kapasitor (*condenser*).
4. Arena pengujian merupakan arena tanpa halangan.

1.5. Metodologi

Dalam penulisan tugas akhir ini menggunakan metodologi sebagai berikut :

1. Studi Literatur
Studi literatur merupakan proses pengumpulan teori-teori penunjang dari tugas akhir ini yang dapat berupa artikel, jurnal keilmiah, forum diskusi, atau media lainnya kebenarannya dapat dipertanggungjawabkan.
2. Perancangan Perangkat Lunak
Pada tahap perancangan perangkat lunak merupakan pembahasan mengenai algoritma pemrograman perilaku dari *swarm robot* dalam mencari sumber gas dan berkomunikasi dengan robot lainnya.
3. Perancangan Perangkat Keras
Perancangan perangkat keras merencanakan susunan modul maupun komponen pendukung dari *swarm robot* seperti sensor gas, Arduino

UNO, sensor suara, sensor jarak, *platform* robot, maupun perangkat tambahan lainnya.

4. Perancangan Sistem

Perancangan sistem merupakan proses implementasi dari perancangan perangkat lunak dan perangkat keras. Hal yang perlu diperhatikan dalam proses ini adalah kesesuaian antara pemasangan perangkat keras dengan perangkat lunak sehingga sistem dapat bekerja sesuai dengan yang direncanakan.

5. Pengujian Sistem

Melakukan pengujian mengenai proses mendeteksi gas dan mekanisme penentuan mode *master* dan *slave* dari masing-masing robot melalui komunikasi suara.

6. Penulisan Laporan Tugas Akhir

Tahap penulisan laporan dimulai ketika melakukan proses pengujian sistem hingga hasil akhir dari pengujian.

1.6. Sistematika Penulisan

Laporan tugas akhir ini disusun dengan sistematika sebagai berikut:

- BABI Pendahuluan akan membahas latar belakang, tujuan, masalah, metode penelitian, sistematika, pembahasan, dan relevansi.
- BAB II Teori penunjang dan literatur yang berkaitan bagi tugas akhir ini.
- BAB III Perencanaan alat baik perangkat keras maupun perangkat lunak.
- BAB IV Berisikan hasil pengujian dan pengembangan alat.
- BAB V Penutup berisi kesimpulan yang diperoleh dari tugas akhir ini, serta saran-saran untuk pengembangan lebih lanjut

1.7. Relevansi dan Manfaat

Hasil yang diharapkan dari tugas akhir ini diharapkan sebagai konsep alternatif dalam pemanfaatan robot sebagai alat pencari kebocoran gas di bidang industri.

Halaman ini sengaja dikosongkan

BAB II TINJAUAN PUSTAKA

2.1. *Swarm Robot*

Swarm robot merupakan robot berkelompok (berkoloni) yang terdiri lebih dari satu robot sederhana dan beroperasi secara bersama-sama dengan sepenuhnya mengontrol dirinya sesuai dengan interaksi yang dilakukan terhadap lingkungannya [1]. Konsep dasar dari *swarm robot* diadaptasi dari tingkah laku serangga berkoloni yang saling bekerja sama dalam melakukan suatu hal. Seperti semut-semut contohnya, memakai jejak-jejak feromon, senyawa kimia yang dihasilkan dan dikeluarkan oleh semut sebagai penanda rute yang telah dilewati untuk menemukan makanan. Jejak yang paling sering dilewati akan menyebabkan suatu akumulasi feromon yang menarik semut-semut baru sementara lintasan yang jarang dilewati tidak mengalami akumulasi feromon dan akhirnya menguap.

Kriteria dari sistem *swarm robot* adalah sebagai berikut :

- Kemandirian – setiap individu robot yang tergabung dalam *swarm robot* haruslah robot yang mandiri (*autonomous*) yang dapat berinteraksi dan merespon terhadap lingkungan.
- Jumlah yang banyak – *swarm robot* dibentuk oleh beberapa robot homogen sehingga perilaku saling bekerja sama dapat terjadi.
- Kemampuan terbatas – *swarm robot* harus memiliki kemampuan relatif terbatas untuk menjalankan sebuah tugas secara individu.
- Skalabilitas dan konsistensi – menambah beberapa unit *swarm robot* akan meningkatkan performa secara keseluruhan, sedangkan mengurangi beberapa unit tidak akan mengakibatkan penurunan performa secara signifikan.
- Koordinasi terdistribusi – *swarm robot* hanya memiliki kemampuan komunikasi jarak dekat dan penginderaan yang terbatas. Koordinasi antar robot bergantung pada informasi yang diterima dan pengambilan keputusan masing-masing robot [3].

Swarm robot dikembangkan untuk melengkapi kekurangan kelemahan dari robot tunggal konvensional, dan keuntungannya adalah efisiensi, ketahanan, dan fleksibilitas ketika robot melakukan tugasnya [4]. Dalam melakukan tugasnya *swarm robot* perlu melakukan

komunikasi antar robot untuk dapat saling berkoordinasi dan memberi informasi terkait lingkungan yang dihadapi robot.

Aplikasi penerapan *swarm robot* disesuaikan dengan tugas yang diperintahkan, seperti:

- a. Mencari/berburu. Penerapannya dapat berbeda-beda bergantung pada kemampuan sekelompok robot, seperti eksplorasi secara kolektif, pencarian jalur terpendek, dan pembagian tugas secara efisien [5]. Beberapa contoh aplikasinya yakni pembersihan limbah beracun, *search and rescue* (SAR) dan mengumpulkan sampel lapangan.
- b. Tugas berbahaya., contohnya seperti mencari/mendeteksi ranjau akan dengan mudah diselesaikan oleh *swarm robot*.
- c. Penjelajahan dan pemetaan. Untuk memeriksa suatu objek daerah biasanya membutuhkan waktu yang lama, dan disinilah dapat memanfaatkan *swarm robot*. Secara individu *swarm robot* memiliki penginderaan yang terbatas, namun kemampuan secara kolektif sebagai sekelompok *swarm robot* dapat digunakan untuk membuat sebuah pengetahuan secara umum (contoh membentuk sebuah peta suatu daerah). Beberapa aplikasi dimasa depan yang dapat dicapai diantaranya menggunakan *swarm robot* untuk menjelajah planet lain atau penggunaan nanorobot pada pembuluh darah manusia dan artri untuk kebutuhan medis (contoh untuk menyembuhkan kanker) [6].



Gambar 2.1 Swarm robot [2]

Tantangan dalam mendesain sebuah *swarm robot* adalah membuat model program yang mudah digunakan dan efektif dengan kesederhanaan dan keterbatasan perangkat yang digunakan. Tidak seperti sistem multi-robot konvensional, *swarm robot* menekankan pada penggunaan secara luas dari robot yang jumlahnya sangat banyak [7]. Hal tersebut membutuhkan sebuah sistem *swarm robot* dengan kemandirian adaptasi masing-masing robot untuk mengatasi jumlah robot yang bervariasi tanpa mengganggu performanya secara signifikan.

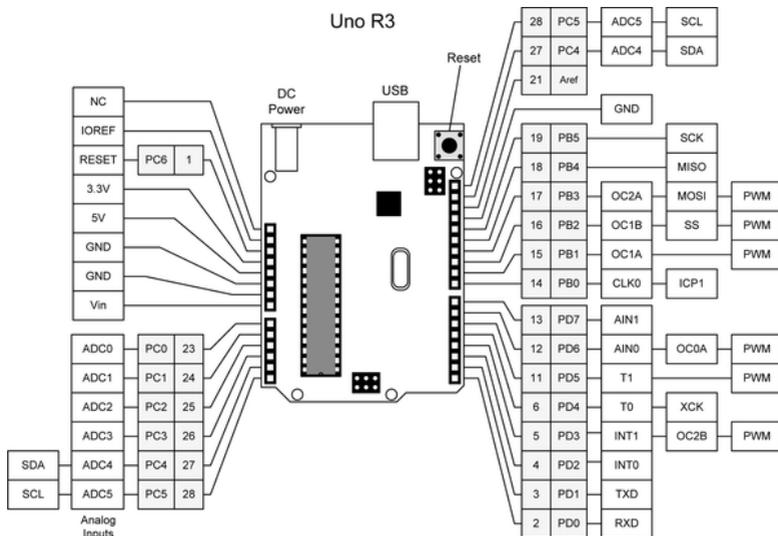
2.2. Arduino UNO

Arduino adalah sebuah *platform* (kombinasi antara arsitektur perangkat keras dan perangkat lunak) yang diciptakan untuk menyederhanakan proses rangkaian dan pemrograman mikrokontroler sehingga menjadi lebih mudah untuk diaplikasikan secara langsung. Arduino ini disusun pada sebuah software yang diberi nama Arduino IDE. Perangkat lunak inilah yang paling utama, membantu menjembatani antara bahasa mesin yang begitu rumit menjadi bahasa dan logika yang lebih mudah dimengerti manusia. Pada dasarnya perangkat keras arduino adalah mikrokontroler ATmega328. Perangkat arduino dirancang untuk mempermudah pengguna melakukan pemrograman dan koneksi Atmega328 dengan komponen lainnya.

Tabel 2.1 Spesifikasi Arduino UNO [8]

| | |
|--------------------------------|---------------------------------------|
| Mikrokontroler | ATmega328 |
| Tegangan pengoperasian | 5V |
| Tegangan input yang disarankan | 7-12V |
| Batas tegangan input | 6-20V |
| Jumlah pin I/O digital | 14 (6 diantaranya menyediakan output) |
| Jumlah pin input analog | 6 |
| Arus DC tiap pin I/O | 20 mA |
| Arus DC untuk pin 3.3V | 50 mA |
| EEPROM | 1 KB (ATmega328P) |
| Kecepatan <i>clock</i> | 16 MHz |

Dalam tugas akhir ini kontroler dari *swarm robot* menggunakan Arduino UNO R3 karena disamping kemudahan dalam penggunaannya juga spesifikasi yang ditawarkan (seperti jumlah pin I/O) sudah mencukupi kebutuhan untuk sensor-sensor dan aktuator dari masing-masing robot. Berikut adalah spesifikasi dari Arduino UNO R3.



Gambar 2.2 Pinout Arduino UNO [9]

2.2.1. Sumber Daya / Power

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. *Board* dapat beroperasi pada suplai eksternal dari 6 sampai 20 volt. Rentang yang disarankan adalah 7 sampai 12 volt karena jika diberi tegangan kurang dari 6 volt akan tidak stabil dan jika menggunakan lebih dari 12V, regulator tegangan menjadi panas dan dapat merusak *board*.

2.2.2. Memori

ATmega328 sebagai mikrokontroler bawaan dari Arduino memiliki memori 32 KB (dengan 0,5 KB digunakan untuk *bootloader*). ATmega328 juga memiliki 2 KB dari SRAM dan 1 KB EEPROM.

2.2.3. Input dan Output

Masing-masing dari 14 pin digital Uno dapat digunakan sebagai input atau output, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()* yang beroperasi pada tegangan 5V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal (terputus secara default) dari 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi spesial:

- Serial : pin 0 (Rx) dan 1 (Tx) Digunakan untuk menerima (Rx) dan mengirimkan (Tx) data serial TTL. Pin ini terhubung dengan pin ATmega8U2 USB-to-Serial TTL.
- *Interrupt* Eksternal : Pin 2 dan 3 dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah (low value), *rising* atau *falling edge*, atau perubahan nilai.
- PWM : Pin 3, 5, 6, 9, 10, dan 11 Menyediakan 8-bit PWM dengan fungsi *analogWrite()*.
- SPI (*Serial Peripheral Interface*) : pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) mendukung komunikasi SPI dengan menggunakan *library* SPI.
- LED : pin 13. Built-in LED terhubung ke pin digital 13. LED akan menyala ketika diberi nilai HIGH

Arduino Uno memiliki 6 input analog, berlabel A0 sampai A5, yang masing-masing menyediakan resolusi 10 bit (1024 nilai yang berbeda). Biasanya rentang pengukuran dari pin analog antara 0-1023 (0 adalah 0V dan 1023 untuk 5V), perubahan tegangan maksimal dapat menggunakan pin AREF dan fungsi *analogReference()*.

2.2.4. Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (Rx) dan 1 (Tx). Pada ATmega16U2 saluran komunikasi serial melalui USB dan muncul sebagai *COM PORT virtual* untuk perangkat lunak pada komputer. *Firmware* 16U2 menggunakan standar driver USB COM, dan tidak perlu *driver* eksternal. Selain itu ATmega328 juga mendukung komunikasi berbasis I2C (*Inter-Integrated Circuit*) dan komunikasi SPI.

2.2.5. Perlindungan Arus USB

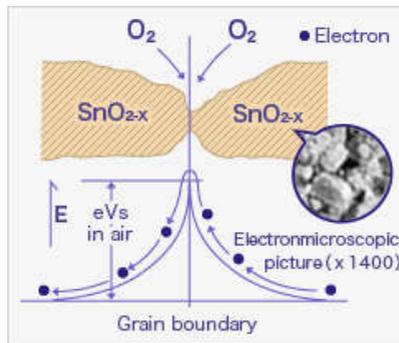
Arduino Uno memiliki *polyfuse reset* yang melindungi port USB komputer pengguna dari arus pendek atau berlebih walaupun kebanyakan komputer telah menyediakan perlindungan internal sendiri. Jika lebih dari 500 mA, sekering otomatis bekerja.

2.3. Sensor Gas TGS2600

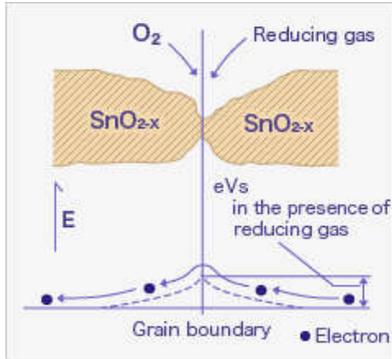
TGS2600 adalah sensor gas jenis semikonduktor. Secara umum prinsip kerja dari sensor gas semikonduktor adalah sebagai berikut. Pada keadaan paling ekstrim dimana kadar oksigen 0%, ketika material sensor metal oksida (biasanya tin dioxide [SnO_{2-x}]) dipanaskan dengan suhu misalkan 400°C , elektron bebas mengalir melalui bagian persambungan dari kristal tin dioxide. Pada udara bersih (kira-kira 21% O_2), oksigen terhisap ke permukaan metal oksida.

Dengan afinitas elektronnya yang tinggi, oksigen terserap menarik elektron bebas di dalam metal oksida, membentuk sebuah potensial penghalang (eVs pada udara) pada titik persambungan. Potensial penghalang ini mencegah terjadinya aliran elektron, sehingga resistansi sensor tinggi pada udara bersih[10].

Ketika sensor dihadapkan dengan gas yang mudah terbakar atau “*reducing gas*” (contohnya karbon monoksida), reaksi oksidasi gas tersebut dengan oksigen terserap terjadi di permukaan *tin dioxide*. Sebagai hasilnya, kadar dari oksigen terserap pada permukaan *tin dioxide* berkurang, dan tinggi dari potensial penghalang juga berkurang. Elektron sangat mudah mengalir melewati potensial penghalang yang berkurang tingginya, sehingga resistansi sensor berkurang. Konsentrasi gas di udara bisa dideteksi dengan mengukur perubahan resistansi dari sensor gas semikonduktor. Reaksi kimia dari gas-gas dan oksigen terhisap pada permukaan *tin dioxide* bervariasi bergantung pada reaktifitas material sensor dan temperatur kerja dari sensor.

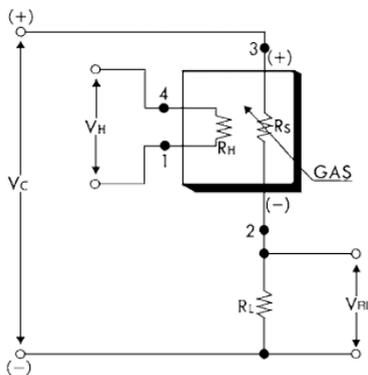


Gambar 2.3 Potensial penghalang (eVs) tinggi saat tidak mendeteksi gas [10]



Gambar 2.4 Potensial penghalang (eVs) rendah saat sensor mendeteksi gas [10]

TGS2600 membutuhkan dua tegangan input, yakni tegangan pemanas (V_H) dan tegangan sirkuit (V_C). Tegangan pemanas (V_H) diberikan untuk menjaga elemen sensor pada temperatur tertentu yang optimal untuk pendeteksian gas. Tegangan sirkuit (V_C) diberikan untuk mengukur tegangan yang melewati resistor beban (R_L) yang terhubung secara seri dengan sensor. Karena sensor memiliki polaritas, maka tegangan yang diperlukan merupakan DC, dan tegangan ini juga bisa diterapkan sebagai tegangan pemanas V_H untuk memenuhi kebutuhan sensor. Besarnya nilai R_L dipilih untuk optimalisasi pengukuran sensor (minimal 0,45 kOhm) dan juga menjaga konsumsi daya sensor sebesar 15 mW. Konsumsi daya terbesar terjadi ketika R_S sama dengan R_L .



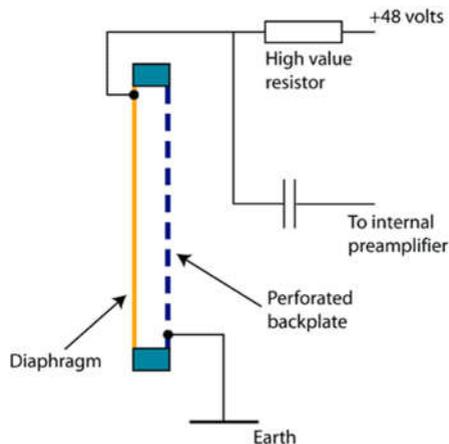
Gambar 2.5 Rangkaian dasar pengukuran sensor [11]

2.4. Mikrofon Kondenser

Mikrofon kapasitor (kondenser) memiliki karakteristik yakni diafragma dari mikrofon membentuk satu sisi plat dari kapasitor. Plat lainnya merupakan *backplate* yang paralel dan berjarak sangat dekat dengan diafragma. Dalam kapasitor, $Q = C \times V$, lebih sesuai jika direpresentasikan menjadi $V = Q/C$. Q merupakan muatan listrik. Ketika kapasitor diisi, jumlah elektron pada satu sisi plat tidak seimbang dengan plat lainnya. C merepresentasikan kapasitansi, yang merupakan kemampuan kapasitor untuk menyimpan muatan. Dan V merupakan tegangan. Karena $V = Q/C$, kapasitor dengan nilai kapasitansi rendah mampu menghitung tegangan yang lebih besar daripada kapasitor dengan kapasitansi tinggi, untuk nilai muatan yang sama.

Diafragma dan *backplate* dari mikrofon umumnya dapat terisi hingga 48 V. Hal ini dilakukan dengan menggunakan resistor bernilai tinggi sehingga sinyal suara tidak terpengaruh. Kapasitansi dari setiap kapasitor diatur oleh jarak antara kedua plat. Semakin dekat jaraknya, semakin tinggi kapasitansinya[12]. Karena $V = Q/C$, jika kapasitansi berubah, maka tegangannya pun akan berubah. Komponen AC dari perubahan tegangan ini adalah sinyal yang dihasilkan oleh mikrofon terhadap respon suara.

Mikrofon kapasitor hanya mampu menghasilkan arus yang sangat lemah dari diafragmanya, maka dari itu mikrofon kapasitor harus memiliki penguat internal dekat dengan diafragma.



Gambar 2.6 Skema dari mikrofon kondenser [12]

2.5. Sensor Ultrasonik

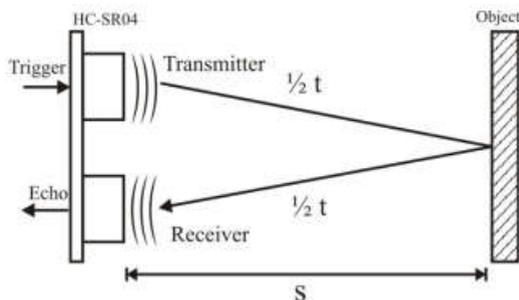
Sensor ultrasonik adalah komponen elektronik yang mampu mengubah energi listrik menjadi energi mekanik berupa gelombang ultrasonik. Sensor ini terdiri dari rangkaian pemancar ultrasonik yang dinamakan *transmitter* dan penerima ultrasonik yang disebut *receiver*. Sensor ultrasonik yang paling sering digunakan adalah HC-SR04. Transmitter memancarkan gelombang ultrasonik dengan frekuensi 40 kHz kemudian menghantam sebuah objek dan pantulan gelombang tersebut ditangkap oleh *receiver*, seperti terlihat pada gambar 2.7 berikut.

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah, ketika pulsa trigger diberikan pada sensor, transmitter akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan output TTL rising edge menandakan sensor mulai menghitung waktu pengukuran, setelah receiver menerima gelombang pantulan maka pengukuran waktu akan dihentikan dengan menghasilkan output TTL falling edge. Jika waktu pengukuran adalah t dan kecepatan suara adalah 344 m/s, maka jarak antara sensor dengan objek dapat dihitung dengan menggunakan rumus berikut.

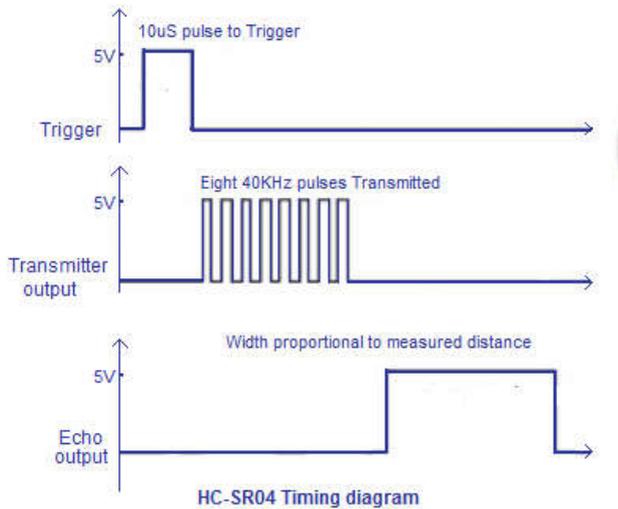
$$s = t \times 344/2 \quad (2.1)$$

Tabel 2.2 Spesifikasi sensor HC-SR04 [14]

| | |
|-----------------|---------------|
| Tegangan Suplai | 5 VDC |
| Arus Kerja | 15 mA |
| Sudut Efektif | <15° |
| Rentang Jarak | 2 cm – 400 cm |
| Resolusi | 0,3 cm |



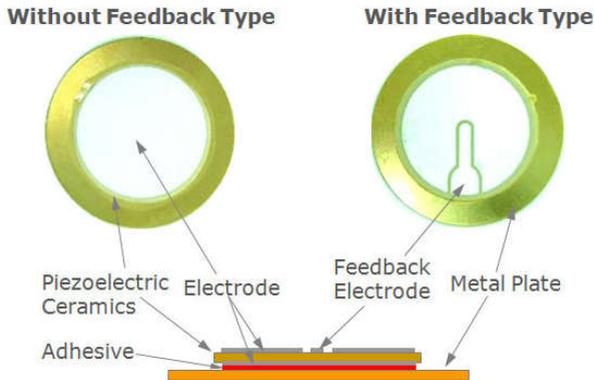
Gambar 2.7 Prinsip kerja sensor ultrasonik [13]



Gambar 2.8 *Timing diagram* dari sensor HC-SR04 [15]

2.6. Buzzer

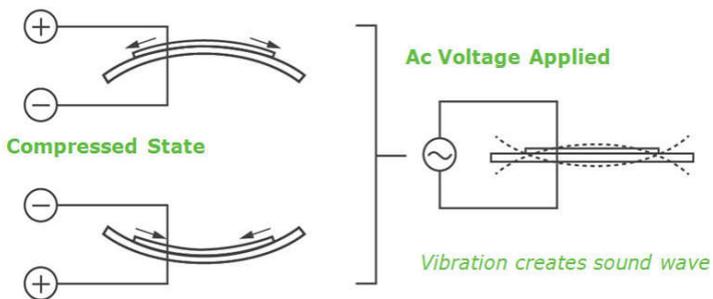
Secara umum *buzzer* dibagi menjadi dua jenis yakni *piezo buzzer* dan *magnetic buzzer* [16]. Inti dari semua jenis *piezo buzzer* ada pada elemen piezoelektriknya. Elemen piezoelektrik tersusun dari sebuah piezokeramik dan sebuah plat logam sejajar. Kedua sisi dari piezoelektrik keramik berisi elektroda sebagai konduksi listrik. Material piezo menunjukkan sebuah fenomena yang diketahui sebagai efek piezoelektrik dan efek reverse piezoelektrik. Apabila mendapat tekanan atau gaya akan menyebabkan material menghasilkan medan listrik, begitu pula sebaliknya.



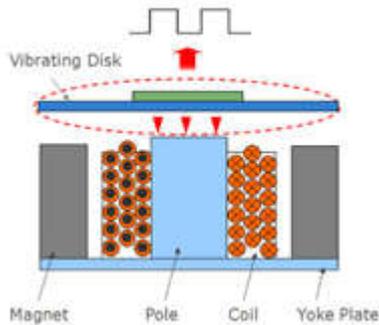
Gambar 2.9 Buzzer piezokeramik [16]

Ketika tegangan bolak-balik diberikan kepada elemen piezokeramik, elemen akan mengendur dan menyusut membentuk sebuah lengkungan. Karakteristik dari material piezoelektrik ini membantu plat keramik bergetar dengan cepat untuk menghasilkan gelombang suara.

Pada magnetic *buzzer*, kepingan yang bergetar di dalam *buzzer* ditarik menuju kutub oleh medan magnet. Saat ada sinyal yang beresilasi melewati gulungan, akan menghasilkan medan magnet yang berubah-ubah dimana menggetarkan kepingan pada frekuensi yang sama dengan sinyal drive.



Gambar 2.10 Cara kerja *buzzer* piezokeramik [16]

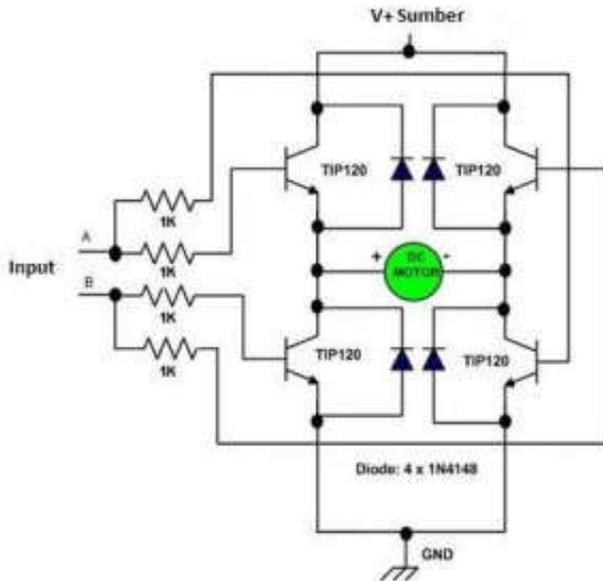


Gambar 2.11 Cara kerja *buzzer* magnetik [16]

Buzzer juga dapat dikategorikan menjadi dua jenis yakni *buzzer* aktif dan pasif. Perbedaan dari kedua jenis itu adalah, *buzzer* aktif sudah mempunyai osilator sendiri di dalam komponennya sehingga ia akan bersuara ketika dihubungkan dengan tegangan DC. Akan tetapi *buzzer* pasif tidak memiliki osilator sendiri, akibatnya ia tidak akan berbunyi bila hanya dihubungkan dengan sumber DC. *Buzzer* pasif membutuhkan sinyal gelombang kotak dengan frekuensi diantara 2 kHz sampai 5 kHz untuk mengaktifkannya. Dengan kata lain, *buzzer* aktif hanya dapat menghasilkan satu jenis frekuensi saja, bergantung pada besar osilatornya, sedangkan *buzzer* pasif dapat diaplikasikan untuk berbagai frekuensi dengan cara memberi sinyal gelombang kotak yang bervariasi.

2.7. Driver Motor

Driver motor DC yang dibahas adalah driver motor DC tipe *H-Bridge* menggunakan power driver berupa transistor. Rangkaian driver motor DC *H-Bridge* transistor ini dapat mengendalikan arah putaran motor DC dalam dua arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*High*) dan (*Low*). Apabila menggunakan metode logika TTL 0 dan 1 maka driver hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum. Rangkaian driver motor DC *H-Bridge* menggunakan rangkaian jembatan transistor 4 unit dengan proteksi impuls tegangan induksi motor DC berupa dioda yang dipasang paralel dengan masing-masing transistor secara reverse bias. Rangkaian driver motor DC secara rinci dapat dilihat pada gambar 2.12 di bawah ini.



Gambar 2.12 Rangkaian dasar driver motor tipe *H-Bridge* [17]

Driver motor DC dengan metode logika TTL (0 dan 1) hanya dapat mengendalikan arah putar motor DC dalam dua arah tanpa pengendalian kecepatan putaran (kecepatan maksimum). Untuk mengendalikan motor DC dalam dua arah dengan rangkaian driver motor DC *H-Bridge* diatas konfigurasi kontrol pada jalur input adalah dengan memberikan input berupa logika TTL ke jalur input A dan B. Untuk mengendalikan arah putar searah jarum jam adalah dengan memberikan logika 1 pada jalur input A dan logika 0 pada jalur input B. Untuk mengendalikan arah putar berlawanan arah jarum jam adalah dengan memberikan logika 1 pada jalur input B dan logika 0 pada jalur input A.

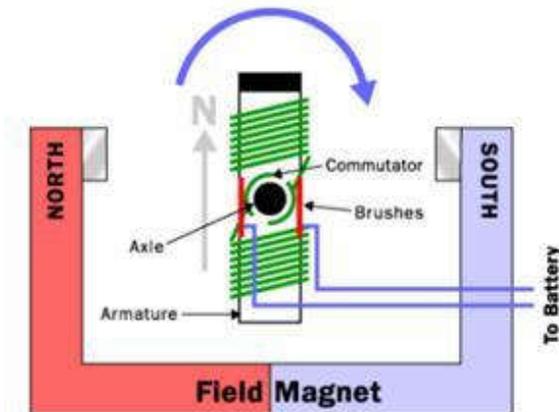
Driver motor DC dengan metode PWM dapat mengendalikan arah putaran motor DC dan kecepatan motor DC menggunakan pulsa PWM yang diberikan ke jalur input A dan B. Untuk mengendalikan arah putar motor DC searah jarum jam dengan kecepatan dikendalikan pulsa PWM maka jalur input B selalu diberikan logika 0 dan jalur input A diberikan pulsa PWM. Untuk mengendalikan arah putar motor DC berlawanan arah jarum jam dengan kecepatan dikendalikan pulsa PWM maka jalur input A selalu diberikan logika 0 dan jalur input B diberikan pulsa PWM. Kecepatan putaran motor DC dikendalikan oleh persentasi duty cycle

pulsa PWM yang diberikan ke jalur input rangkaian driver motor DC *H-Bridge* transistor.

2.8. Motor DC

Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC . Motor DC menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*)[4]. Motor DC bekerja berdasarkan prinsip gaya Lorentz, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus. Mekanisme ini diperlihatkan pada gambar 2.13 berikut.

Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar atau rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak yang disebut komutator.



Gambar 2.13 Struktur motor DC [18]

Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. Kecepatan putar motor DC (N) dirumuskan dengan persamaan,

$$N = (V_{TM} - I_a R_a) / K\phi \quad (2.2)$$

V_{TM} : Tegangan terminal
 I_a : Arus jangkar motor
 R_a : Hambatan jangkar motor
 K : Konstanta motor
 Φ : Fluks magnet yang terbentuk

Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor V_{TM} . Metode lain yang biasa digunakan untuk mengendalikan kecepatan motor DC adalah dengan teknik modulasi lebar pulsa atau PWM.

2.3 Motor Servo

Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotornya akan diumpankan kembali ke rangkaian kontrol dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo.

Keunggulan dari penggunaan motor servo adalah :

- Tidak bergetar dan tidak beresonansi saat beroperasi.
- Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
- Penggunaan arus listrik sebanding dengan beban yang diberikan.
- Resolusi dan akurasi dapat diubah dengan hanya mengganti *encoder* yang dipakai.
- Tidak berisik saat beroperasi dengan kecepatan tinggi.

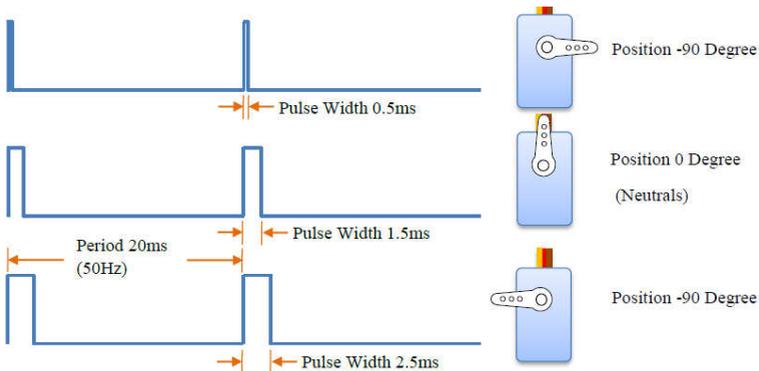
Motor servo biasanya dimanfaatkan pada pembuatan robot, salah satunya sebagai penggerak kaki robot. Motor servo dipilih sebagai penggerak pada kaki robot karena motor servo memiliki tenaga atau torsi yang besar, sehingga dapat menggerakkan kaki robot dengan beban yang

cukup berat. Pada umumnya motor servo yang digunakan sebagai penggerak pada robot adalah motor servo 180°.

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil. Berdasarkan jenis rotasinya, umumnya terdapat dua jenis motor servo yang dan terdapat di pasaran, yaitu motor servo rotasi 180 dan servo rotasi *continuous*.

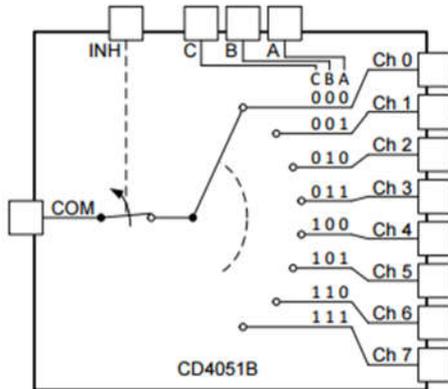
Motor servo standard (servo rotasi 180) adalah jenis yang paling umum, dimana putaran poros outputnya terbatas hanya 90 kearah kanan dan 90 kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180 . Sedangkan motor servo rotasi *continuous* merupakan jenis motor servo yang perputaran porosnya tanpa batasan atau dapat berputar terus, baik ke arah kanan maupun kiri.

Untuk mengendalikan motor servo perlu diberikan sumber tegangan dan sinyal kontrol. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan. Sedangkan untuk mengendalikan putaran motor servo dilakukan dengan memberi pulsa kontrol berfrekuensi 50 Hz (periode 20 ms) dan *duty cycle* yang berbeda. Dimana untuk menggerakkan motor servo sebesar 90° diperlukan pulsa dengan *duty cycle* pulsa positif 1,5 ms dan unjtk bergerak sebesar 180° diperlukan lebar pulsa 2 ms.



Gambar 2.14 Pengaturan lebar pulsa (*duty cycle*) untuk mengontrol posisi servo [19]

2.9. Multiplexer

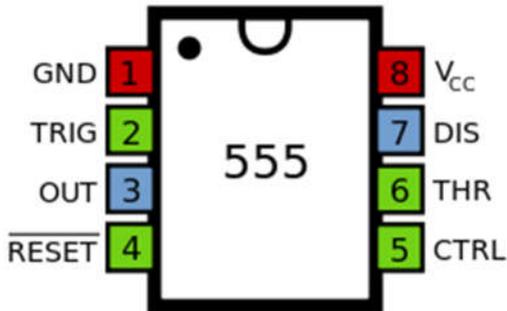


Gambar 2.15 Pinout multiplexer CD4051 [20]

Multiplexer (MUX) adalah rangkaian logika yang dapat memilih input yang akan diteruskan ke bagian output. Pemilihan input dipilih berdasarkan sinyal yang diberikan di bagian kontrol (*select*). CD4051 adalah MUX/DEMUX sinyal analog yang dikontrol secara digital. Komponen ini dapat menampung 8 kanal input dan 1 kanal output dengan input kontrol biner A, B, dan C. Saat input kontrol diberi 000 maka yang aktif adalah kanal 0, begitu seterusnya hingga input kontrol 111 maka yang menyala adalah kanal 7.

2.10. Rangkaian Osilator

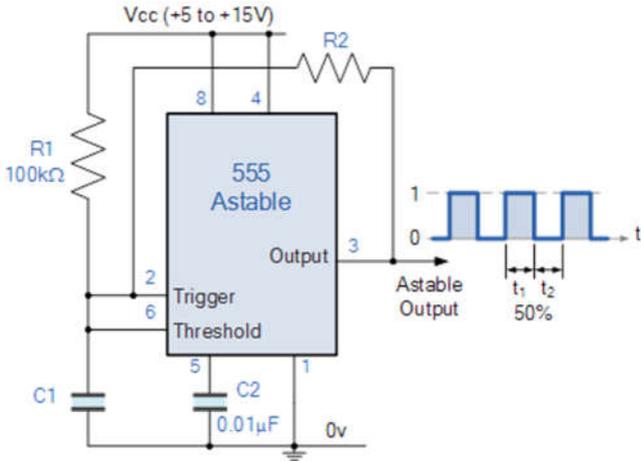
IC 555 umumnya disebut IC *timer* yang dapat digunakan sebagai *timer* ataupun osilator. Osilator adalah rangkaian yang dapat menghasilkan gelombang elektrik konstan, seperti sinusoidal atau gelombang kotak. IC 555 dapat diaplikasikan dalam tiga mode, mode pertama yakni *monostable* untuk menghasilkan pulsa *One-Shot*, mode *astable* untuk menghasilkan pulsa yang kontinu (*timer*), dan *bistable* untuk beroperasi layaknya *Schmitt trigger*.



Gambar 2.16 Pinout dari IC 555 [21]

Pin 1 (GND) berfungsi sebagai tegangan referensi atau 0 V. pin 2 (TRIG) membuat pin 3 (OUT) *high* dan interval waktu dimulai ketika input pada pin 2 lebih rendah dari $1/2$ nilai tegangan CTRL (biasanya $1/3 V_{CC}$, CTRL akan secara otomatis bernilai $2/3 V_{CC}$ ketika tidak terhubung). Sederhananya output akan berlogika 1 selama pin TRIG tetap pada tegangan rendah. Output dari *timer* sepenuhnya bergantung dari amplitudo dari tegangan eksternal *trigger* yang masuk ke pin ini. Pin 3 (OUT) adalah tempat keluaran sinyal sekitar 1,7 V dibawah V_{CC} atau GND. Pin 4 (RESET) adalah pin yang aktif ketika berlogika 0, selama pin 4 diberi tegangan 0 V maka output akan tetap bernilai 0. Pin 5 (CTRL) memungkinkan akses kontrol ke pembagi tegangan internal (umumnya $2/3 V_{CC}$). Pin 6 (THR) mengatur akhir dari interval output *high* ketika tegangan dari THR (*threshold*) lebih besar dari CTRL ($2/3 V_{CC}$ jika CTRL tidak terhubung). Pin 7 (DIS) berupa output *open collector* yang akan *men-discharge* kapasitor pada interval tertentu. Dan pin 8 (V_{CC}) merupakan tempat suplai tegangan diberikan, biasanya antara 5 V-15 V.

Untuk membuat gelombang kotak dengan *duty cycle* 50% dan frekuensi yang dapat diubah-ubah menggunakan rangkaian osilator seperti pada gambar 2.17 berikut.



Gambar 2.17 Rangkaian osilator dengan *duty cycle* 50% [22]

Rangkaian osilator dengan IC 555 dapat menghasilkan *duty cycle* 50% ketika kapasitor pewaktu, C1 melakukan proses *charging* dan *discharging* melalui resistor yang sama. Ketika output dari osilator 555 berlogika 1, kapasitor akan mengisi melalui resistor R2 dan ketika output berlogika 0, kapasitor akan dikosongkan melalui R2 juga. Resistor R1 digunakan untuk memastikan kapasitor terisi penuh senilai dengan tegangan suplai. Berikut adalah hubungan antara frekuensi output dengan nilai RC.

$$f = \frac{1}{0,693 \times (2R2) \times C1} \text{ Hz} \quad (2.3)$$

Perlu dicatat bahwa nilai dari resistor R1 harus cukup tinggi untuk memastikan hal tersebut mengganggu proses pengisian dari kapasitor untuk menghasilkan *duty cycle* 50%. Dan juga mengubah nilai kapasitor pewaktu C akan merubah frekuensi osilasi dari rangkaian *astable*.

2.11. *Liquid Crystal Display (LCD)*

LCD adalah salah satu jenis perangkat penampil elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja bukan dengan menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-*

lit. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.

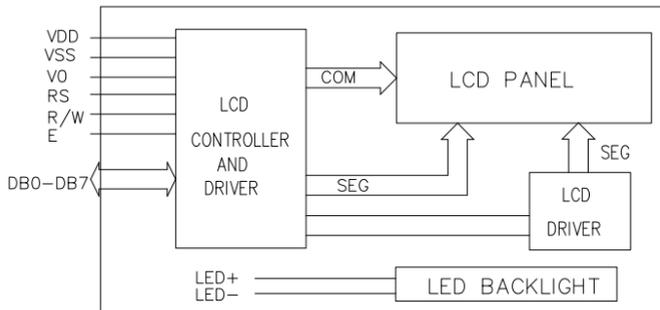
LCD tersusun dari lapisan campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan seven-segment dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan LCD memiliki polarisator cahaya vertikal depan dan polarisator cahaya horizontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler ini dilengkapi dengan memori dan register. Memori yang digunakan mikrokontroler internal LCD adalah :

- DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada.
- CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
- CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

Kontrol register yang terdapat dalam suatu LCD adalah sebagai berikut :

- Register perintah yaitu register yang berisi perintah- perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.
- Register data yaitu register untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.



Gambar 2.18 Diagram blok LCD [23]

Pin input dan kontrol dalam suatu LCD diantaranya :

- Pin data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
- Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika low menunjukkan yang masuk adalah perintah, sedangkan logika high menunjukkan data.
- Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika low tulis data, sedangkan high baca data.
- Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
- Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

Penulisan data ke register perintah dilakukan dengan tujuan mengatur tampilan LCD, inialisasi dan mengatur *address counter* maupun *address data*. Kondisi RS berlogika 0 menunjukkan akses data ke register perintah. RW berlogika 0 menunjukkan proses penulisan data akan dilakukan. *nibble* tinggi (bit 7 sampai bit 4) terlebih dahulu dikirimkan dengan diawali pulsa logika 1 pada *clock* E. Kemudian *nibble* rendah (bit 3 sampai bit 0) dikirimkan dengan diawali kembali pulsa logika 1 pada *clock* E. Untuk mode 8 bit, proses penulisan dapat langsung

dilakukan secara 8 bit (bit 7 sampai dengan bit 0) dan diawali sebuah pulsa logika 1 pada *clock E*.

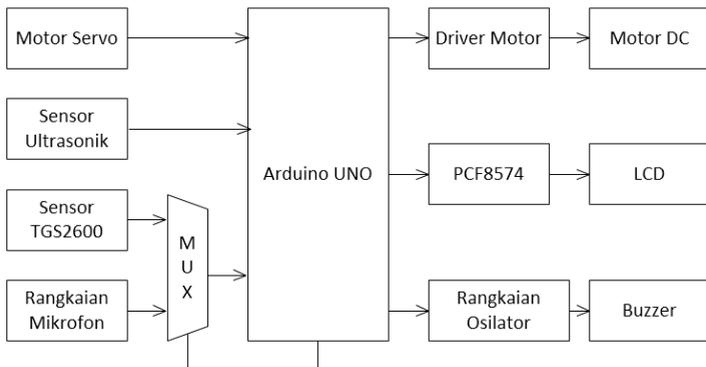
Proses pembacaan data pada register perintah biasa digunakan untuk melihat status *busy* dari LCD atau membaca *address counter*. RS diatur pada logika 0 untuk akses ke register perintah, R/W diatur pada logika 1 yang menunjukkan proses pembacaan data. 4 bit *nibble* tinggi dibaca dengan diawali pulsa logika 1 pada *clock E* dan kemudian 4 bit *nibble* rendah dibaca dengan diawali pulsa logika 1 pada *clock E*. Untuk mode 8 bit, pembacaan 8 bit (*nibble* tinggi dan rendah) dilakukan sekaligus dengan diawali sebuah pulsa logika 1 pada *clock E*.

Penulisan data pada register data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke register data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Data *nibble* tinggi dikirim dengan diawali pulsa logika 1 pada sinyal *Clock E* dan kemudian diikuti *nibble* rendah yang juga diawali pulsa logika 1 pada sinyal *clock E*.

Pembacaan data dari register data dilakukan untuk membaca kembali data yang tampil pada LCD. Proses dilakukan dengan mengatur RS pada logika 1 yang menunjukkan adanya akses ke register data. Kondisi R/W diatur pada logika 1 yang menunjukkan adanya proses pembacaan data. Data *nibble* tinggi dibaca dengan diawali adanya pulsa logika 1 pada *clock E* dan dilanjutkan dengan data *nibble* rendah yang juga diawali dengan pulsa logika 1 pada *clock E*.

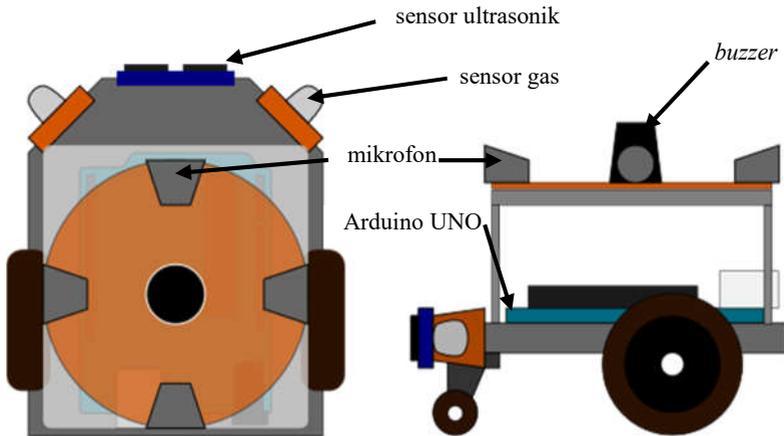
BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan perancangan sistem mulai dari perancangan perangkat keras hingga perancangan perangkat lunak. *Swarm robot* yang dirancang bertujuan untuk mencari sumber kebocoran gas. Gas yang mampu dideteksi merupakan jenis gas mudah terbakar, seperti karbon monoksida, metana, dan lain-lain. Karena kebocoran gas yang dideteksi menyebar melalui udara, arah angin dapat sewaktu-waktu berubah dan menghambat proses pencarian sumber gas. Dengan memanfaatkan jumlah *swarm robot* yang digunakan, dapat meminimalisir efek dari lingkungan sekitar karena ketika salah satu robot kehilangan penginderaan karena pengaruh arah angin, robot lainnya dapat membantu mendeteksi. Untuk dapat berkoordinasi satu sama lain diperlukan sistem komunikasi yang memungkinkan kedua robot saling bertukar informasi. Salah satu metode komunikasi yang bisa diterapkan adalah menggunakan suara, karena melalui suara masing-masing robot dapat saling mengetahui arah robot lainnya. Gambar 3.1 adalah diagram blok dari keseluruhan sistem yang dirancang.



Gambar 3.1 Diagram blok *swarm robot*

3.1. Perancangan Perangkat Keras

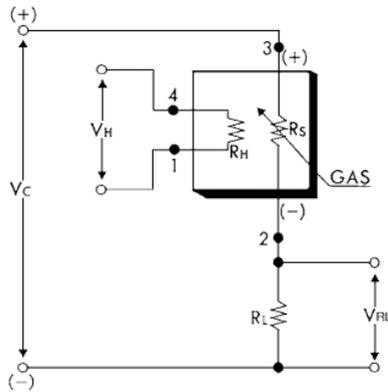


Gambar 3.2 Ilustrasi *swarm robot* yang dirancang

Dalam satu buah robot terdiri dari dua buah sensor pendeteksi gas TGS2600, perangkat komunikasi yang terdiri dari buzzer sebagai pengirim informasi dan mikrofon sebagai penerima, perangkat pendeteksi jarak yakni sensor ultrasonik HC-SR04, perangkat untuk bergerak yang terdiri dari dirver motor dan motor DC, serta perangkat tambahan lainnya yang semuanya dikendalikan oleh sebuah perangkat kontrol Arduino UNO.

3.1.1. Perangkat Pendeteksi Gas

Perangkat pendeteksi gas terdiri dari dua buah sensor TGS2600 yang diletakkan pada bagian depan kiri dan kanan robot menggunakan prinsip *stereo nose*. Tujuannya adalah untuk dapat mengetahui arah dari datangnya gas. Gas mengalir sesuai dengan arah aliran angin. Apabila gas mengalir dari arah kiri robot maka intensitas yang diterima pada sensor kiri lebih besar dari sensor di sebelah kanan dan robot akan bergerak ke arah datangnya gas. Begitu pula ketika gas mengalir dari arah kanan robot, sensor sebelah kanan akan mendapat intensitas yang lebih besar daripada sensor di sebelah kiri dan robot akan bergerak ke kanan. Akan tetapi, jika gas mengalir dari arah depan, pembacaan kedua sensor gas akan sama dan jika pun berbeda tidak menunjukkan nilai yang jauh berbeda.



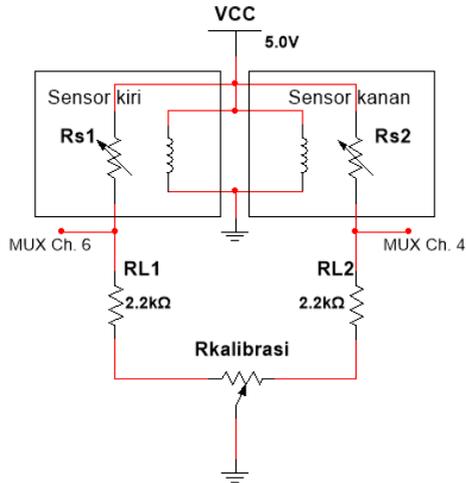
Gambar 3.3 Rangkaian pengukuran sensor TGS2600

Output dari rangkaian pengukuran merupakan sebuah tegangan DC, yang diukur oleh Arduino sebagai representasi dari intensitas gas yang diterima sensor. Dalam keadaan aktif dan kondisi udara normal (tidak mendeteksi gas kontaminan), berdasarkan *datasheet* TGS2600 sensor memiliki resistansi R_S sebesar 10 kOhm hingga 90 kOhm. Nilai R_L yang dipilih adalah 10 kOhm, diatas nilai R_L yang diijinkan yakni 0,45 kOhm.

Dari rangkaian yang ditunjukkan gambar 3.2 di atas, dengan nilai V_C dan V_H adalah 5 V, tegangan output (V_{RL}) dapat diturunkan dengan rumus pembagi tegangan.

$$V_{RL} = \frac{R_L}{R_S + R_L} \times V_C \quad (3.1)$$

Hubungan antara V_{RL} dengan R_S berbanding terbalik. Dalam kondisi normal, nilai R_S besar (10 kOhm - 90 kOhm) sehingga nilai V_{RL} semakin kecil. Namun ketika sensor mendeteksi gas, nilai R_S menurun sehingga V_{RL} menjadi semakin besar. Semakin besar intensitas gas yang dibaca maka semakin kecil nilai R_S . Dengan begitu, besar kecilnya tegangan output V_{RL} dapat merepresentasikan pendeteksian gas oleh robot.

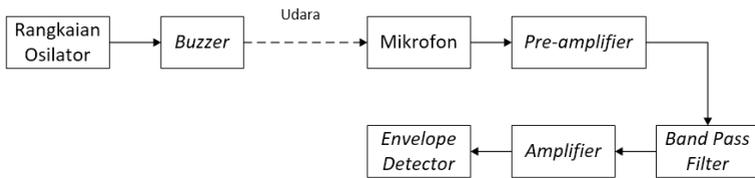


Gambar 3.4 Rangkaian kalibrasi sensor TGS2600 kiri dan kanan

Dalam penerapannya, walaupun dipasang dengan nilai R_L yang sama, adanya pengaruh dari nilai toleransi R_L mempengaruhi nilai pembacaan tegangan dari kedua sensor. Hal ini menyebabkan nilai antara sensor sebelah kanan dan kiri mengalami perbedaan beberapa mV. Nilai tegangan yang berbeda dapat mempengaruhi pembacaan arah sumber gas. Maka dari itu perlu ditambahkan sebuah potensiometer untuk mengkalibrasi kedua nilai tegangan sensor. Rangkaian untuk kalibrasi kedua sensor TGS2600 disajikan pada gambar 3.4 diatas.

3.1.2. Perangkat Komunikasi

Sistem komunikasi antar *swarm robot* menggunakan media suara dengan frekuensi 5 kHz. Alasan pemilihan frekuensi 5 kHz adalah untuk mencegah terjadinya interferensi antara suara komunikasi robot dengan suara manusia (500 Hz – 2 kHz). Bentuk informasi yang dikirim berupa *beep* dalam rentang waktu beberapa detik. Dalam sistem komunikasi tersebut terdiri dari perangkat pengirim sinyal dan penerima sinyal. Ketika bekerja, perangkat penerima maupun pengirim bekerja secara bergantian, ketika robot A sedang mengirim informasi, robot B akan menerima dan sebaliknya, ketika robot B sebagai pengirim maka robot A akan menerima.

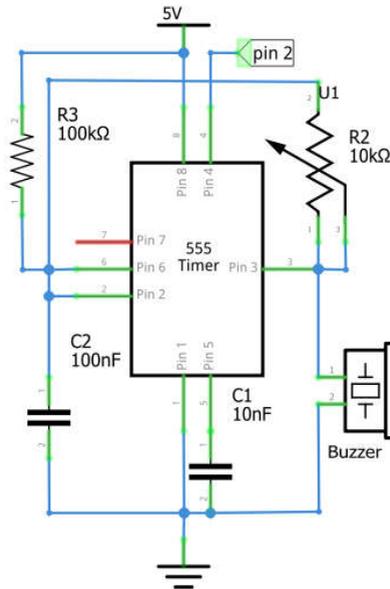


Gambar 3.5 Diagram blok sistem komunikasi suara

Rangkaian komunikasi berupa suara sangat rentan terhadap gangguan-gangguan seperti suara bising, *noise*, maupun gangguan lainnya yang dapat menyebabkan perangkat komunikasi terganggu. Untuk meminimalisir efek dari gangguan tersebut perlu adanya sebuah *filter* yang berfungsi melemahkan semua jenis sinyal yang berada diluar rentang frekuensi *filter*. Dalam perangkat komunikasi ini digunakan *Band Pass Filter* sebagai penyaring sinyal yang masuk dan juga memastikan hanya sinyal komunikasi 5 kHz yang dapat lewat.

3.1.2.1 *Perangkat Pengirim*

Perangkat pengirim terdiri dari rangkaian osilator dan transduser *buzzer*. *Buzzer* yang digunakan adalah jenis *buzzer* pasif, oleh karena itu dibutuhkan sebuah pembangkit sinyal gelombang kotak sebagai sinyal input yang membuat *buzzer* bergetar dan menghasilkan frekuensi suara yang diinginkan. Rangkaian osilator berfungsi untuk menghasilkan sinyal gelombang kotak berfrekuensi 5 kHz. Rangkaian osilator yang dibuat menggunakan komponen utama IC 555. Gambar 3.6 menunjukkan rangkaian osilator menggunakan IC 555.



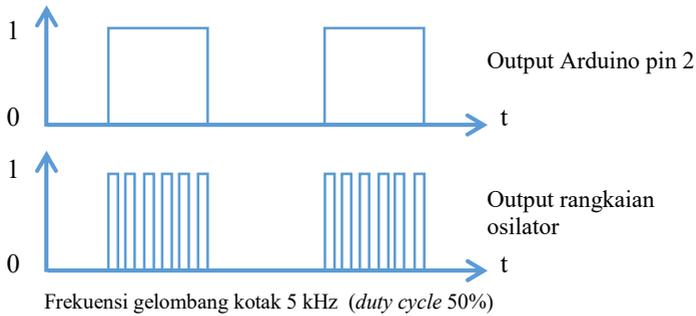
Gambar 3.6 Skematik rangkaian osilator dengan *duty cycle* 50%

Rangkaian pada gambar 3.6 menghasilkan gelombang kotak dengan *duty cycle* 50%, karena proses pengisian dan pengosongan kapasitor C1 melalui resistor yang sama (R2). Ketika output dari osilator berlogika 1, kapasitor terisi melalui R2 dan ketika output berlogika 0, kapasitor juga membuang arus melalui R2. Kemudian *buzzer* akan mengubah sinyal listrik gelombang kotak dari rangkaian osilator menjadi kedalam bentuk gelombang suara dengan frekuensi yang sama.

Untuk mendapatkan output berupa gelombang osilasi 5 kHz dapat dicari dengan rumus berikut.

$$f = \frac{1}{0,693 \times (2R2) \times C1} \text{ Hz} \quad (3.2)$$

Jika C1 sebesar 10 nF, maka nilai dari resistor R2 berdasarkan perhitungan dari rumus diatas adalah 14,4 kOhm.

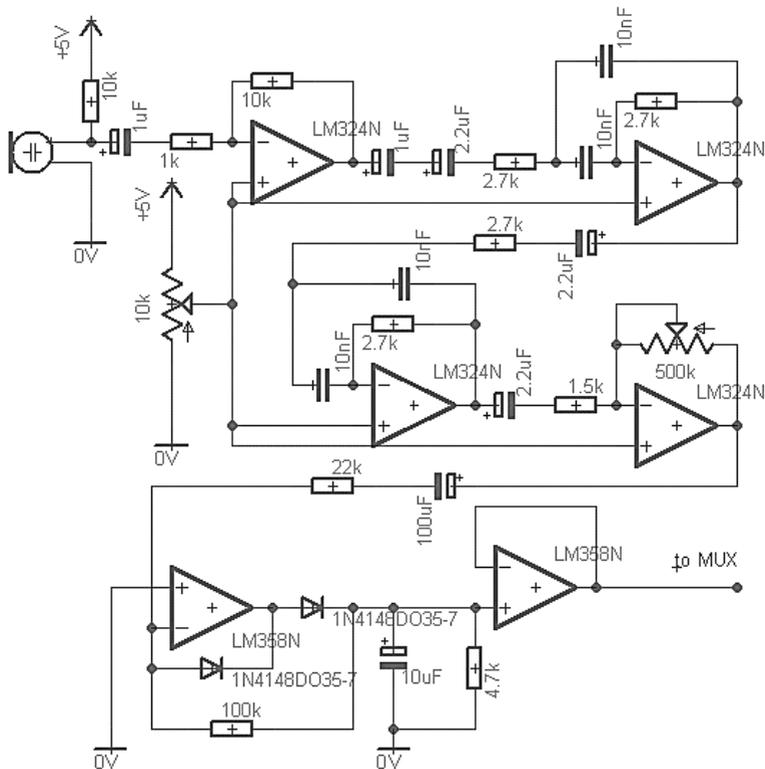


Gambar 3.7 *Timing diagram* antara output pin 2 Arduino dengan output rangkaian osilator

Pin RESET (pin 4) dari IC 555 terhubung dengan pin 2 Arduino untuk mengontrol waktu aktif dan reset rangkaian osilator. Pin RESET aktif ketika diberi logika 0, jadi selama pin ini diberi logika 0 oleh Arduino keluaran dari osilator akan tetap 0 V. Akan tetapi ketika diberi logika 1, maka pin output akan menghasilkan gelombang kotak. Sehingga suara *beep* itu sendiri diperoleh dengan mengkombinasikan antara waktu aktif dan reset dalam rentang waktu yang ditentukan.

3.1.2.2 *Perangkat Penerima*

Perangkat penerima merupakan rangkaian pendeteksi suara dengan komponen utama mikrofon kondenser yang dilengkapi dengan *Band-Pass Filter* (BPF), rangkaian penguat, dan *envelope detector* untuk dapat menghitung level dari suara yang ditangkap dan juga jumlah *beep* yang didengar. Output yang diharapkan dari rangkaian penerima ini adalah berupa sinyal *envelope* yang menyerupai gelombang kotak untuk dapat menghitung jumlah *beep* dan juga mengukur level *peak* dari sinyal. Berikut adalah rangkaian skematik dari sistem.



Gambar 3.8 Skematik rangkaian penerima

Cara kerja dari rangkaian pada gambar 3.8 adalah, ketika mikrofon menerima sebuah getaran berupa suara, getaran tersebut akan dikonversi menjadi sebuah gelombang elektrik yang sangat lemah. Kemudian gelombang elektrik masuk ke rangkaian preamplifier untuk dikuatkan sebesar -10 kali. Rangkaian preamplifier sangat sensitif terhadap sinyal yang lemah dan memiliki penguatan yang sangat besar sehingga mikrofon akan menjadi sangat peka terhadap suara yang masuk.

Jenis IC op-amp yang digunakan merupakan *single supply* (tegangan input V^+ dan *ground*) dan sinyal yang diolah merupakan sinyal AC, oleh karena itu input *non-inverting* (+) dari op-amp diberi tegangan bias DC agar sinyal yang diproses tidak terpotong pada tegangan

negatifnya. Nilai bias 1,75 V dipilih berdasarkan pengaruh karakteristik dari op-amp yang digunakan adalah mampu menghasilkan output tegangan maksimum sebesar 3,5 V ($V_{cc} - 1,5$ V). Apabila penguatannya melebihi dari batas tegangan output tegangan akan mengakibatkan sinyal terpotong pada nilai saturasi op-amp.

Tahap berikutnya merupakan BPF orde 4 dimana rangkaian ini berfungsi untuk memilah sinyal dengan rentang frekuensi tertentu yang dapat lewat. BPF orde 4 dibentuk dari dua buah BPF orde 2 yang disusun secara seri. Karakteristik dari *filter* BPF di atas adalah sebagai berikut.

$$f_r = \frac{1}{2\pi\sqrt{R_1R_2C_1C_2}} \quad (3.3)$$

$$\text{Penguatan maksimal } (A_v) = \frac{R_2}{2R_1} \quad (3.4)$$

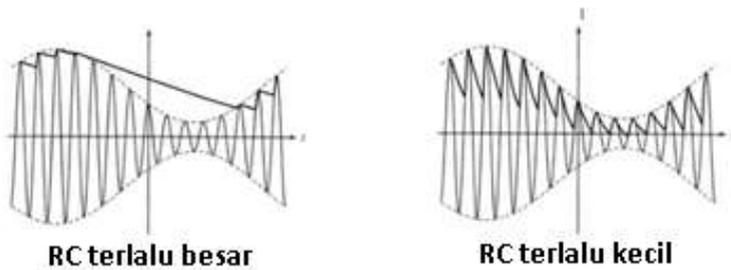
Untuk mendapatkan frekuensi resonan f_r sebesar 5 kHz, dengan nilai $C_1 = C_2 = 10$ nF maka nilai dari resistor yang dibutuhkan :

$$R_2 = R_1 = \frac{1}{2\pi \times 5000 \times 10 \times 10^{-9}} \approx 3,1 \text{ k}\Omega$$

Karena nilai resistor 3,1 kOhm tidak ada dipasaran, dipilih nilai yang mendekati yakni 2,7 kOhm sehingga nilai f_r bergeser menjadi 5,8 kHz. Alasannya adalah untuk semakin menjauhi frekuensi 500 Hz – 2 kHz. Dengan begiitu total penguatan dari BPF orde 4 adalah $(-0,5) \times (-0,5) = 0,25$ kali.

Sinyal yang telah diseleksi oleh BPF kemudian dikuatkan oleh rangkaian penguat. Rangkaian penguat berupa *inverting amplifier* dengan $R_i = 1,5$ kOhm dan R_f berupa resistor variabel 500 kOhm yang nilainya dapat diubah-ubah. Besar penguatan dari rangkaian penguat apabila resistor variabel diatur pada nilai 300 kOhm adalah -200 kali.

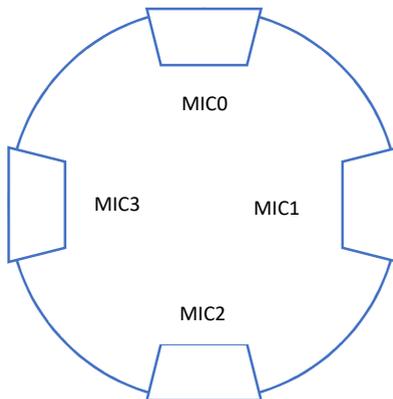
Sinyal yang telah dikuatkan masuk ke rangkaian *envelope detector* yang bertujuan untuk memberi selubung ke tiap deret gelombang atau *beep* untuk memudahkan proses Arduino dalam menghitung jumlah *beep* yang masuk. Rangkaian *envelope detector* terdiri dari rangkaian penyearah setengah gelombang, kapasitor dan resistor. Prinsip kerjanya hampir sama dengan rangkaian detektor puncak, dimana gelombang sinyal disearahkan oleh dioda D1 dan D2 untuk memperoleh gelombang positif dan mengisi kapasitor C4. Kapasitor akan terus mengisi dan perlahan tegangan kapasitor akan sama dengan tegangan puncak dari sinyal.



Gambar 3.9 Efek dari nilai resistor dan kapasitor terhadap bentuk selubung

Saat rangkaian tidak menerima sinyal, kapasitor C4 akan membuang muatan menuju ke resistor R10 hingga tegangan kapasitor sama dengan nol. Nilai resistor dan kapasitor sangat mempengaruhi bentuk selubung yang diperoleh. Apabila kapasitor atau resistor terlalu kecil, selubung yang dihasilkan membentuk *ripple*, namun sebaliknya ketika nilai resistor terlalu besar, proses *discharging* dari kapasitor akan lambat dan mempengaruhi pembacaan pada Arduino.

Perangkat penerima dipasang pada empat arah (depan, kanan, belakang dan kiri) sehingga dengan menghitung besar amplitudo yang ditangkap masing-masing penerima, dapat diperoleh informasi arah datangnya suara.

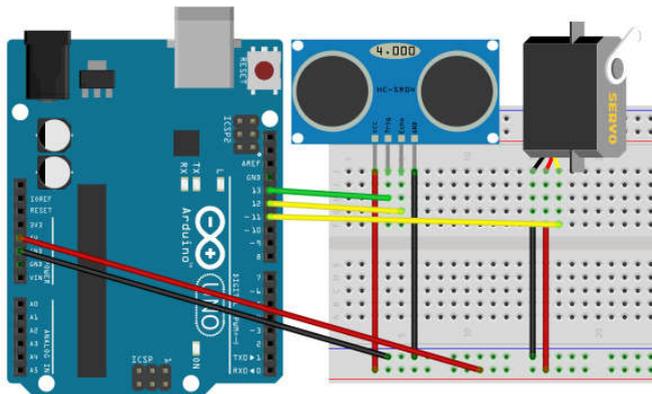


Gambar 3.10 Tata letak pemasangan perangkat penerima pada robot

MIC0, MIC1, MIC2, dan MIC3 merupakan perangkat penerima. Dengan susunan seperti pada gambar 3.10, memungkinkan memperoleh informasi 8 arah yang berbeda. Apabila tegangan puncak yang diterima pada MIC0 dan MIC1 lebih besar maka sumber suara berada pada arah kanan depan. Jika tegangan puncak pada MIC0 dan MIC3 yang lebih besar, maka sumber berada pada arah kiri depan robot, dan begitu seterusnya.

3.1.3. Perangkat Pendeteksi Jarak

Perangkat pendeteksi jarak yang digunakan adalah sensor ultrasonik HC-SR04 yang dibantu oleh motor servo untuk bisa mendeteksi arah kanan dan kiri robot. Sensor ultrasonik dapat mengukur jarak dengan cara menghitung waktu dari gelombang ultrasonik 40 kHz dikirim oleh *transmitter* hingga gelombang dipantulkan dan diterima kembali oleh *reciever*. Namun ada batasan yang dimiliki oleh sensor HC-SR04 dimana sudut efektif pengukuran maksimum adalah 30° dari posisi tegak lurus dengan objek. Apabila lebih dari sudut tersebut maka pembacaan sensor akan salah. Sensor HC-SR04 dipasang pada motor servo untuk membaca jarak kiri dan kanan robot ketika diperlukan. Dengan memberikan sinyal PWM pada servo maka servo akan berputar ke arah sudut tertentu berdasarkan lebar pulsa yang diberikan oleh Arduino. Berikut skema perangkat yang dirancang.



Gambar 3.11 Skema antara Arduino, motor servo dan HC-SR04

Arduino dan perangkat pendeteksi jarak (HC-SR04 dan motor servo) terhubung antara pin-pin yakni.

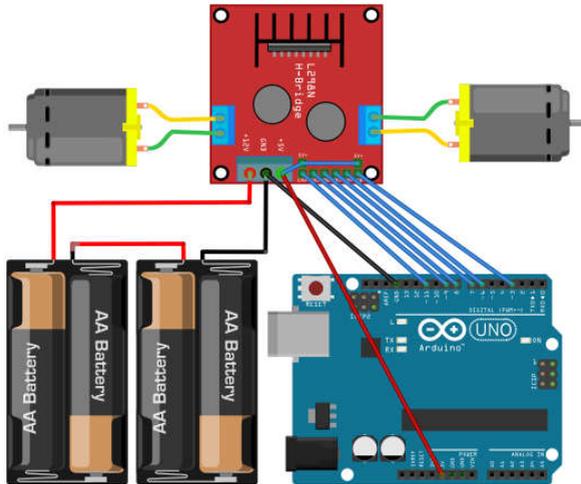
- Pin *Trigger* HC-SR04 → pin 13
- Pin *Echo* HC-SR04 → pin 12
- Pin PWM motor servo → pin 10

3.1.4. Perangkat Gerak Robot

Robot beregerak menggunakan dua buah motor DC dan sebuah driver motor untuk menentukan arah putaran masing-masing motor. Driver motor yang digunakan adalah IC L298N, mampu mengatur dua buah motor DC. L298N memiliki 6 pin input dan 4 pin output. Input 1 dan input 2 berfungsi untuk mengatur arah putaran dari motor 1, sedangkan input 3 dan input 4 mengatur motor 2. Sebagai contoh, ketika input 1 diberi logika 1 dan input 2 diberi logika 0, maka motor 1 akan berputar searah jarum jam, namun apabila diberi logika sebaliknya arah putaran motor 1 juga akan berubah menjadi berlawanan arah jarum jam.

Input EN A dan EN B berfungsi untuk mengatur kecepatan putaran motor 1 dan motor 2 dengan cara memberikan sinyal PWM. Semakin besar *duty cycle* yang diberikan maka semakin cepat pula putaran motor DC, sebaliknya apabila semakin rendah *duty cycle* semakin lambat pula kecepatan putaran motor. Empat pin output terhubung langsung dengan motor DC. Untuk menggerakkan kedua motor DC, driver motor perlu diberi tegangan suplai lain karena perangkat gerak inilah yang mengkonsumsi daya lebih banyak dibandingkan dengan Arduino. Pada *swarm robot* ini tegangan suplai yang digunakan adalah empat buah baterai AA 1,5 V disusun secara seri sehingga menghasilkan tegangan 6 V. Arduino sebagai pusat kontrol dengan driver motor terhubung dengan pin-pin berikut.

- Pin IN1 driver → pin 9
- Pin IN2 driver → pin 8
- Pin IN3 driver → pin 7
- Pin IN4 driver → pin 6
- Pin EN A driver → pin 11
- Pin EN B driver → pin 3



Gambar 3.12 Skema dari perangkat gerak

3.1.5. Perangkat Tambahan

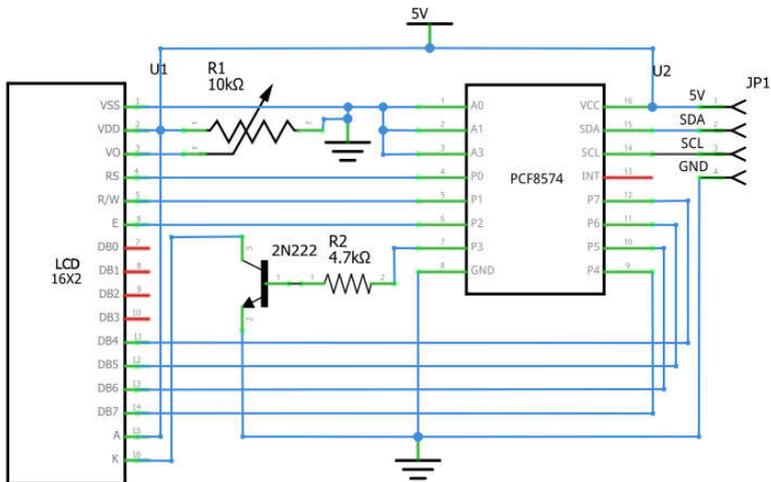
Untuk memudahkan dalam proses pengujian *swarm robot*, masing-masing robot dilengkapi dengan *Liquid Crystal Display* (LCD) yang berfungsi untuk menampilkan data-data penting, angka, ataupun perintah yang sedang dilaksanakan oleh robot. Sehingga perilaku robot dapat dipantau terus tanpa harus menghubungkan ke laptop terlebih dahulu. Apabila terjadi kesalahan (*error*) atau permasalahan ketika pengujian, akan lebih mudah ditelusuri dengan memeriksa data yang ditampilkan LCD.

LCD 16x2 dengan mode 4-bit data membutuhkan 7 buah pin untuk dapat bekerja. Namun Arduino UNO tidak memiliki cukup pin lagi untuk memuat LCD. 4 buah perangkat penerima memerlukan 4 pin analog, 2 buah sensor TGS2600 membutuhkan 2 pin analog, sensor HC-SR04 membutuhkan 2 pin, motor servo membutuhkan 1 pin PWM, perangkat pengirim memerlukan 1 pin, dan driver motor membutuhkan 2 pin PWM dan 4 pin digital. Hanya tersisa 2 pin digital yang belum terpakai dan tentunya tidak cukup untuk LCD.

Alternatif yang diambil untuk menyasiasi penggunaan pin Arduino adalah dengan menambah perangkat tambahan yakni moduli LCD *backpack* I2C (*Inter-Integrated Circuit*) dan sebuah multiplexer. Modul LCD *backpack* I2C ini memungkinkan LCD melakukan komunikasi data

dengan Arduino hanya dengan 2 buah pin, *Serial Data* (SDA) dan *Serial Clock* (SCL). Pin SDA dan SCL yang sudah tersedia pada Arduino UNO ada pada pin A4 dan A5. Akan tetapi, penggunaan pin A4 dan A5 akan membuat ketersediaan pin analog berkurang. Disinilah peranan komponen multiplexer yang berfungsi memilah beberapa jalur dan hanya melewatkan satu jalur dalam satu waktu. Apabila ingin membaca sinyal pada jalur satu, multiplexer akan mengontrol jalur yang dilewatkan ke output adalah jalur satu, begitu seterusnya. Dengan begitu semua sensor akan dapat mengirim informasi ke Arduino.

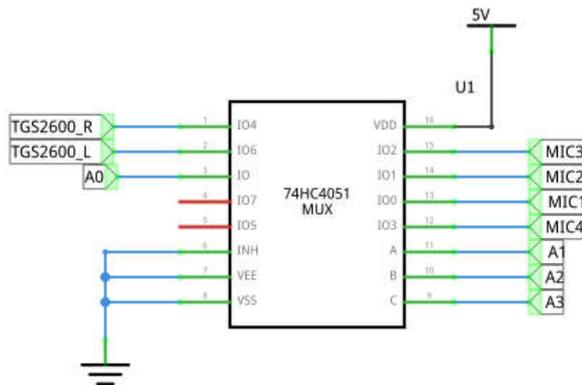
Modul LCD *backpack* I2C menggunakan komponen PCF8574 sebagai I/O *expander* 8-bit, yang memungkinkan Arduino mendapat 8-bit I/O tambahan. Kelebihannya adalah mampu berkomunikasi dengan LCD hanya dengan menggunakan 2 jalur, yang sebelumnya harus menyediakan setidaknya 6 pin untuk berkomunikasi dengan LCD secara langsung. Di sisi lain, frekuensi *clock* dari komunikasi I2C sebesar 100 kHz membuat kecepatan akses menuju pin I/O tidak secepat mengakses pin I/O lainnya pada Arduino. Namun hal ini tidak akan begitu mempengaruhi performa karena PCF8574 digunakan hanya untuk menampilkan ke LCD saja. Berikut ini merupakan gambaran skematik rangkaian dari modul LCD *backpack* yang digunakan.



Gambar 3.13 Rangkaian skematik LCD I2C menggunakan PCF8574

Perangkat tambahan lainnya adalah multiplexer CD4051. Sesuai dengan *datasheet*, komponen CD4051 dapat digunakan baik sebagai multiplexer maupun demultiplexer dengan 8 jalur atau kanal. Apabila CD4051 digunakan sebagai multiplexer, 8 jalur dikontrol oleh 3 buah pin *switch* A, B dan C. Apabila C, B dan A diberi input 000 maka jalur yang dilewatkan pada pin OUT adalah jalur 0, begitu pula ketika diberi 111 maka jalur yang dilewatkan pada pin OUT adalah jalur 7. Perlu dicatat bahwa multiplexer hanya dapat melewatkan satu jalur dalam satu waktu, namun proses *switching* dari satu jalur ke jalur lainnya sangat cepat yakni selama 360 ns. Dengan begitu keempat perangkat penerima maupun dua sensor TGS2600 akan mampu dibaca dengan memilih jalur yang diaktifkan. Berikut adalah pengalamatan dari multiplexer CD4051.

- Jalur 0 → perangkat penerima depan (MIC1)
- Jalur 1 → perangkat penerima kanan (MIC2)
- Jalur 2 → perangkat penerima belakang (MIC3)
- Jalur 3 → perangkat penerima kiri (MIC4)
- Jalur 4 → sensor TGS2600 kanan (TGS2600_R)
- Jalur 6 → sensor TGS2600 kiri (TGS2600_L)
- OUT → Arduino pin A0
- A → Arduino pin A1
- B → Arduino pin A2
- C → Arduino pin A3



Gambar 3.14 Rangkaian skematik multiplexer CD4051

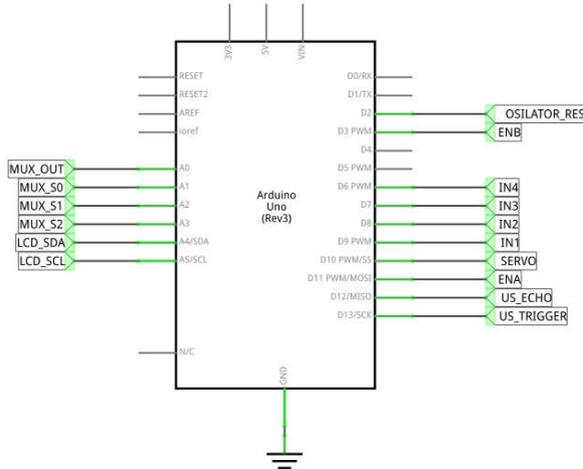
3.2. Perancangan Perangkat Lunak

Perancangan perangkat lunak yang dibahas pada sub bab ini adalah pemrograman Arduino dan algoritma gerak maupun sistem komunikasi dari masing-masing *swarm robot*. Pemrograman satu robot dengan robot lainnya tidak jauh berbeda karena masing-masing robot memiliki perangkat yang sama dan cara kerja yang sama. Algoritma pemrograman dari robot mengikuti diagram alir (*flow chart*) dari gambar berikut.



Gambar 3.15 Diagram alir pemrograman *swarm robot*

Sebelum masuk kedalam pembahasan lebih jauh mengenai pemrograman *swarm robot*, berikut adalah gambar pemetaan penggunaan pin dari Arduino UNO untuk menyesuaikan antara perangkat lunak yang akan dirancang dan perangkat keras yang telah dibahas sebelumnya.



Gambar 3.16 Pemetaan penggunaan pin Arduino UNO

Seperti yang telah dijelaskan pada sub bab sebelumnya, *swarm robot* terdiri dari perangkat pendeteksi gas, perangkat komunikasi, perangkat pendeteksi jarak, perangkat gerak, dan tambahan lainnya. Pada intinya *swarm robot* memanfaatkan kebebasan gerak menyesuaikan dengan lingkungan dan juga komunikasi dengan robot lainnya untuk dapat secara efektif dan cepat mendeteksi arah sumber gas. Tantangannya adalah bagaimana membangun robot yang responsif terhadap lingkungan sekitar dan juga metode komunikasi menggunakan suara antar robot agar robot dapat secara kontinu bertukar informasi.

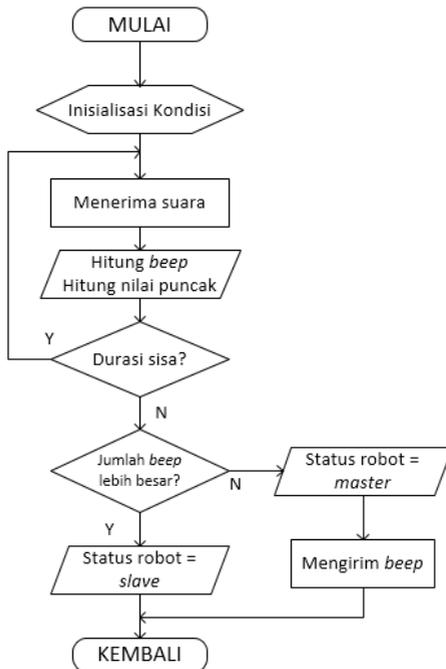
3.2.1. Proses Komunikasi dan Penentuan *Master-Slave*

Komunikasi antar robot dilakukan dengan mengirim dan menerima informasi secara bergantian. Apabila robot A memulai komunikasi dengan mengirim sinyal berupa *beep*, maka robot B akan mengaktifkan perangkat penerima untuk mendengar jumlah *beep* dan nilai *peak* atau puncak dari sinyal yang diterima. Berikutnya giliran dari robot B mengirim sinyal dan robot A mengaktifkan perangkat

penerimanya. Lamanya sinyal suara yang dikirim adalah 1,2 detik berapapun jumlah *beep* yang dikirim. Ketika mengirim 4 *beep* dan 2 *beep* akan terdengar perbedaan durasi dari masing-masing *beep*, dimana 4 *beep* akan terdengar lebih cepat (150 ms) dibandingkan dibandingkan dengan 2 *beep* (300 ms). Untuk mencegah adanya sinyal yang terpotong maka perangkat penerima akan aktif atau mendengar lebih lama dari durasi sinyal yang dikirim, dalam kasus ini ditentukan selama 2 detik.

Tabel 3.1 Pengelompokan intensitas gas berdasarkan pembacaan nilai ADC

| Kelompok | Keterangan | Jumlah <i>beep</i> | Rentang Nilai ADC |
|----------|------------------|--------------------|-------------------|
| 1 | Tidak mendeteksi | 0 | < 160 |
| 2 | Sangat lemah | 1 | 161 – 230 |
| 3 | Lemah | 2 | 231 – 300 |
| 4 | Sedang | 3 | 301 – 370 |
| 5 | Kuat | 4 | 371 – 440 |
| 6 | Sangat kuat | 5 | > 440 |



Gambar 3.17 Diagram alir proses komunikasi dan penentuan *master-slave*

Untuk mengakses perangkat penerima, harus didahului dengan memberi input pada S2, S1, dan S0 multiplexer. Perangkat penerima MIC0 (arah depan) menempati jalur 0 (000), MIC1 (arah kanan) ada pada jalur 1 (001), MIC2 (arah belakang) pada jalur 2 (010), dan yang terakhir MIC3 (arah kiri) berada pada jalur 3 (011). Perangkat penerima yang dipasang sangat sensitif terhadap suara dan *noise* lainnya, terkadang jika hanya mencari nilai maksimum dari sinyal akan menghasilkan kesimpulan yang salah. Maka dari itu untuk meminimalisir kesalahan pembacaan, nilai pembacaan dari masing-masing penerima yang melebihi *threshold* akan di-*sampling* beberapa kali dan dirata-ratakan. Ketika sinyal yang diterima pada saat ini melebihi nilai *threshold*, maka nilai tersebut akan dijumlah dengan nilai sebelum-sebelumnya. Jumlah nilai puncak dibagi dengan jumlah sampel yang diambil akan menghasilkan sebuah nilai rata-rata puncak dari sinyal.

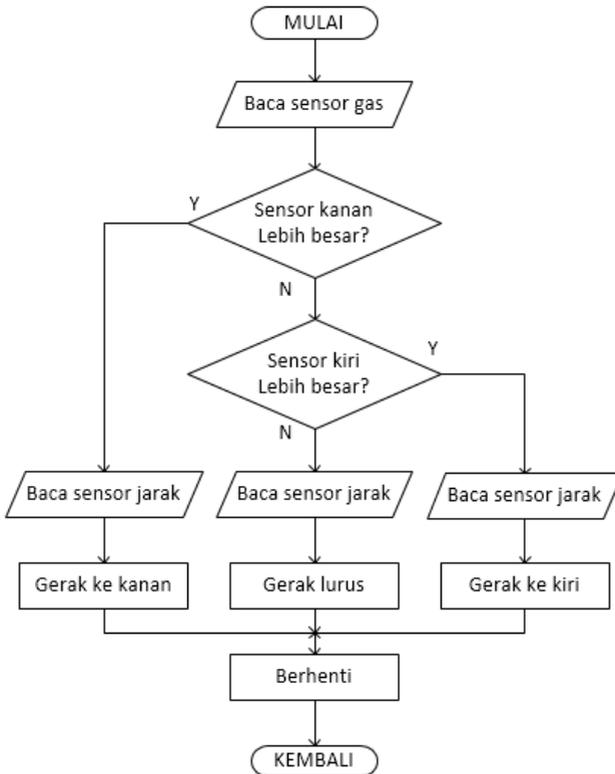
Jumlah *beep* dihitung dengan metode *state change* yakni menghitung jumlah pergantian keadaan sinyal dari logika 0 ke logika 1 dalam 1 periode pengiriman. Logika 0 dan 1 dari sinyal ditentukan melalui membandingkan dengan sebuah nilai *threshold* atau batas yang telah ditentukan. Dikatakan berlogika 0 ketika tegangan sinyal berada dibawah nilai batas dan logika 1 apabila tegangan sinyal lebih besar dari nilai batas. Jika nilai sinyal sekarang lebih besar dari nilai batas (logika 1) dan nilai sinyal sebelumnya dibawah nilai batas (logika 0) maka terjadi perubahan keadaan dan *counter* akan menghitung 1. *Counter* hanya akan menghitung ketika terjadi perubahan keadaan dari logika 0 ke logika 1, jumlah *counter* inilah yang merupakan nilai *beep* yang diterima robot.

Nilai puncak dari sinyal diperlukan untuk menarik informasi mengenai posisi robot lainnya. 4 perangkat penerima dipasang sesuai arah mata angin dan proses pencarian posisi berdasarkan 2 nilai puncak terbesar yang berbeda. Metode penentuan arahnya adalah dengan membandingkan 2 nilai puncak dari arah yang berlawanan (kanan dan kiri, depan dan belakang) sehingga dari 2 perbandingan tersebut diperoleh 2 nilai puncak terbesar. Misalkan nilai puncak arah depan dan kanan lebih besar, maka robot akan menyimpulkan posisi dari robot lain berada di arah serong kanan.

.Setelah proses komunikasi, sekarang robot telah memiliki informasi berupa intensitas gas yang ditangkap robot lain, yang diwakili dengan jumlah *beep* dan nilai puncak dari sinyal. Selanjutnya jumlah *beep* yang diterima oleh robot lain dibandingkan dengan jumlah *beep* yang dikirim untuk menentukan robot *master* dan *slave*. Apabila jumlah *beep*

yang diterima dari robot A lebih besar dari jumlah *beep* yang dikirim robot B maka kesimpulannya robot A mendeteksi gas lebih kuat dibandingkan dengan robot B, sehingga robot A akan berperilaku sebagai *master* dan robot B sebagai *slave*. Apabila jumlah *beep* yang diterima sama, maka robot yang sebelumnya sebagai *master* akan tetap sebagai *master* dan yang sebelumnya sebagai *slave* juga akan tetap menjadi *slave* hingga ia mendapat intensitas yang lebih besar daripada *master*.

3.2.2. Penentuan Gerak

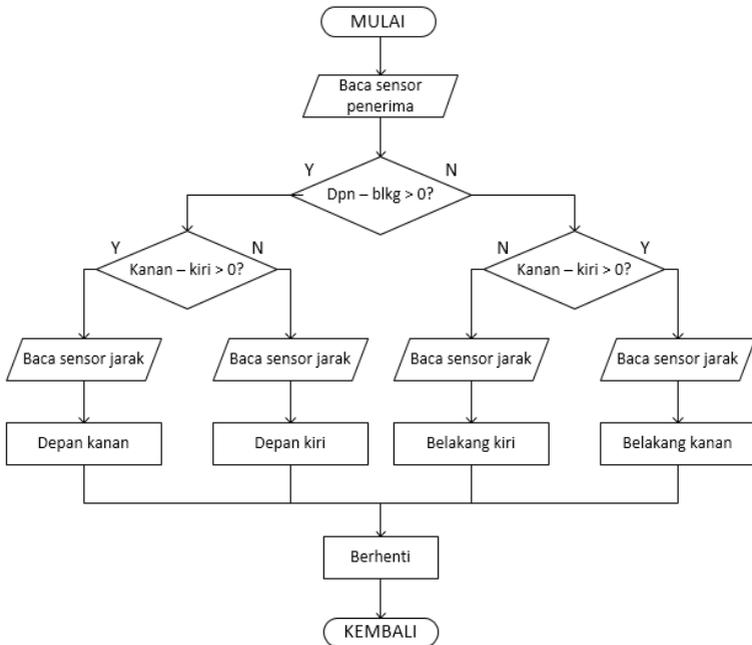


Gambar 3.18 Diagram alir gerak robot *master*

Kedua robot memiliki mode *master* dan *slave*. Setiap mode memiliki algoritma geraknya masing-masing karena fungsi yang

dijalankan berbeda. Mode *master* berfungsi sebagai pemimpin yang memandu robot lainnya untuk bergerak ke tempat yang pembacaan intensitas gasnya lebih tinggi. Maka dari itu gerak robot *master* lebih mengacu pada pembacaan sensor gas.

Berbeda halnya dengan mode *slave*, mode ini berfungsi sebagai pengikut dari robot *master*. Oleh karena itu gerak robot *slave* mengacu pada informasi yang diterima pada perangkat penerima komunikasi. Informasi yang diolah berupa jumlah *beep* dan nilai puncak sinyal. Nilai puncak sinyal digunakan sebagai referensi arah datangnya sumber suara. Pada mode *slave*, robot cenderung lebih pasif, karena menunggu robot lainnya untuk mengirimkan sinyal.



Gambar 3.19 Diagram alir gerak robot *slave*

Halaman ini sengaja dikosongkan

BAB IV PENGUJIAN DAN ANALISIS DATA

4.1. Pengujian Sensor Gas

Pengujian sensor gas TGS2600 menggunakan tiga jenis gas yang mudah terbakar yakni spiritus, alkohol 96% dan bensin. Spiritus tergolong kedalam metanol atau metil alkohol dengan rumus kimia CH_3OH yang memiliki ciri fisik berbentuk cairan pada keadaan atmosfer, mudah menguap, mudah terbakar, dan memiliki bau yang khas. Alkohol 96% ($\text{C}_2\text{H}_5\text{OH}$), sebenarnya merupakan etanol atau etil alkohol yang memiliki ciri tidak berwarna dan mudah menguap. Angka 96% menunjukkan tingkat kemurnian dari alkohol yakni 96% merupakan senyawa alkohol murni dan 4% sisanya adalah campuran umumnya berupa air dan senyawa lainnya. Bahan mudah terbakar lainnya adalah bensin, yang merupakan jenis bahan bakar yang sangat umum yang digunakan pada kendaraan bermotor memiliki ciri berwarna kekuningan dan mudah menguap. Tujuan dari penggunaan tiga jenis gas mudah terbakar untuk membandingkan respon sensor terhadap masing-masing jenis gas. Respon sensor dipengaruhi oleh sensitivitas sensor terhadap gas yang diuji.



Gambar 4.1 Bahan pengujian sensor TGS2600 (Alkohol 96%, spiritus dan bensin)

Pengujian pertama yakni mengamati respon sensor TGS2600 terhadap masing-masing bahan. Caranya adalah dengan mendekatkan masing-masing bahan pengujian ke sensor hingga mencapai nilai saturasi, lalu bahan dijauhkan dan sensor dibiarkan hingga kembali ke nilai awal. Waktu yang ditempuh dari keadaan saturasi ke keadaan normal sebelum mendeteksi gas dicatat. Tujuannya untuk mengetahui respon sensor dari keadaan mendeteksi gas ke keadaan semula, sebelum mendeteksi. Pengujian respon sensor ini dilakukan sebanyak dua kali dan data dari pengujian disajikan kedalam tabel sebagai berikut.

Tabel 4.1 Pengujian respon sensor TGS2600

| Bahan | Kondisi | Pengujian I | Pengujian II |
|----------------|--------------------------|--------------------|---------------------|
| Spiritus | Nilai awal | 0,78 V | 0,59 V |
| | Nilai saturasi | 4,34 V | 4,41 V |
| | Waktu ($t_{recovery}$) | 145 s | 141 s |
| Alkohol 96% | Nilai awal | 0,78 V | 0,53 V |
| | Nilai saturasi | 4,54 V | 4,55 V |
| | Waktu ($t_{recovery}$) | 106 s | 145 s |
| Bensin | Nilai awal | 0,78 V | 0,53 V |
| | Nilai saturasi | 4,27 V | 4,66 V |
| | Waktu ($t_{recovery}$) | 91 s | 134 s |

Dari tabel 4.1, disimpulkan bahwa waktu yang diperlukan sensor dari keadaan saturasi ke kondisi awal sebelum mendeteksi gas membutuhkan waktu hingga 145 detik.

Pengujian kedua yakni menguji pengaruh jarak terhadap pembacaan sensor TGS2600. Pengujian ini tetap menggunakan tiga bahan dan dibantu dengan kipas sebagai alat bantu untuk mensimulasikan arah gerak angin terhadap penyebaran gas. Kipas diarahkan menuju sensor TGS2600 dan di depannya diletakkan bahan penguji (spiritus, alkohol 96% dan bensin) sehingga sensor akan membaca intensitas yang diterima dan pengaruhnya terhadap jarak pembacaan.

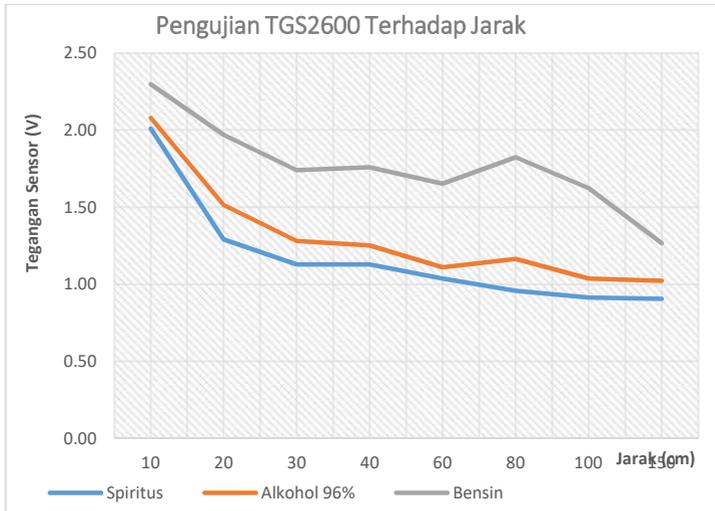


Gambar 4.2 Pengujian pembacaan gas terhadap jarak

Tabel 4.2 Pengujian pengaruh jarak terhadap pembacaan gas

| Jarak | Spiritus | Alkohol 96% | Bensin |
|--------------|-----------------|--------------------|---------------|
| 10 cm | 2.01 V | 2.08 V | 2.30 V |
| 20 cm | 1.29 V | 1.52 V | 1.97 V |
| 30 cm | 1.13 V | 1.28 V | 1.74 V |
| 40 cm | 1.13 V | 1.25 V | 1.76 V |
| 60 cm | 1.04 V | 1.11 V | 1.65 V |
| 80 cm | 0.96 V | 1.16 V | 1.82 V |
| 100 cm | 0.91 V | 1.04 V | 1.62 V |
| 150 cm | 0.90 V | 1.02 V | 1.27 V |

Pada kolom bahan pengujian merupakan nilai tegangan dari rangkaian pengukuran sensor. Semakin jauh jarak bahan pengujian terhadap sensor maka pembacaannya akan semakin berkurang. Selain itu dapat diamati bahwa sensitivitas terhadap ketiga jenis bahan pengujian berbeda-beda. Hal ini dapat dilihat dengan membandingkan nilai tegangan pembacaan masing-masing bahan pada jarak yang sama. Pada jarak 40 cm contohnya, nilai tegangan spiritus, alkohol dan bensin berturut-turut adalah 1,13 V, 1,25 V, dan 1,76 V. Untuk lebih lengkapnya dapat dilihat pada gambar 4.2 berikut.

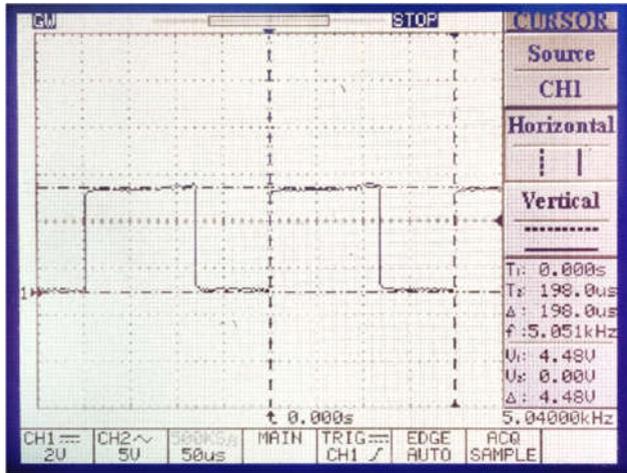


Gambar 4.3 Grafik hasil pengujian pengaruh jarak terhadap pembacaan gas

Dari gambar grafik tersebut dapat disimpulkan bahwa semakin dekat jarak antara sensor dengan sumber gas maka tegangan pembacaan sensor akan semakin besar, sebaliknya semakin jauh jarak sensor dan sumber gas maka semakin kecil pula tegangan pembacaan sensor.

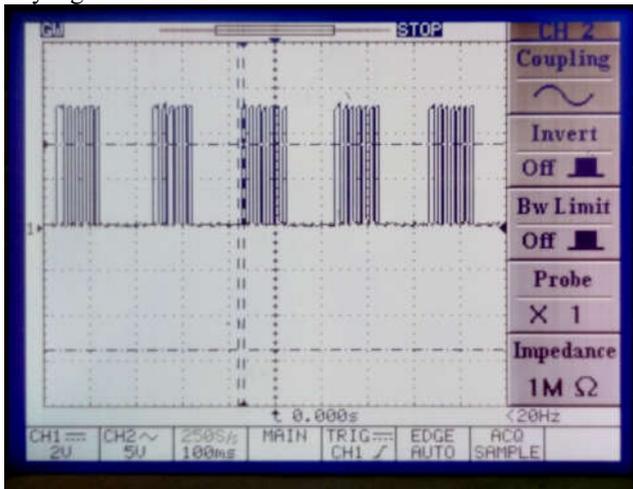
4.2. Pengujian Perangkat Komunikasi

Perangkat komunikasi terdiri dari perangkat pengirim dan penerima. Perangkat pengirim merupakan perangkat yang berfungsi mengirimkan sinyal berupa suara, dalam aplikasi ini frekuensi suara yang digunakan adalah 5 kHz. Frekuensi 5 kHz dihasilkan dari rangkaian osilator menggunakan IC 555 dan jumlah *beep* didapat dengan mengatur waktu nyala dan matinya rangkaian osilator melalui Arduino. Berikut adalah bentuk gelombang yang dihasilkan oleh rangkaian osilator.



Gambar 4.4 Bentuk gelombang output dari osilator

Gelombang kotak yang dihasilkan sudah mendekati 5 kHz dengan mengatur potensiometer R2 sebesar 1,4 kOhm. Kemudian ketika osilator dikontrol oleh Arduino, gelombang yang terlihat membentuk sebuah rangkaian sinyal *beep*. Gambar 4.4 di bawah merupakan bentuk sinyal keluaran yang telah diontrol.



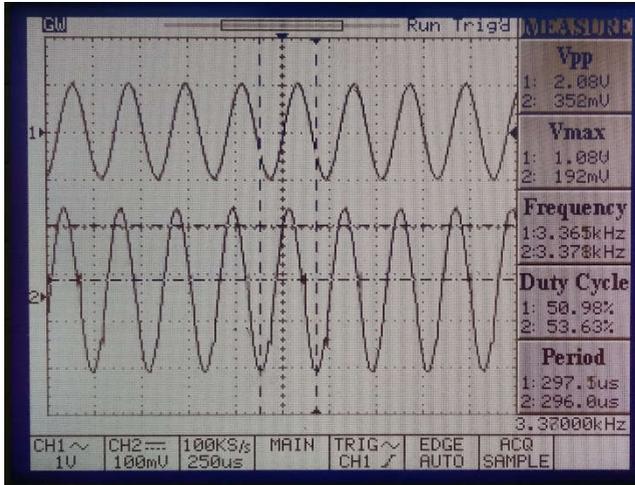
Gambar 4.5 Bentuk sinyal output yang telah dikontrol Arduino

Gambar 4.5 merupakan bentuk sinyal yang telah dikontrol oleh Arduino. Ketika Arduino memberi input logika 1 pada pin RESET IC 555, output dari osilator tetap berupa gelombang kotak 5 kHz. Gelombang kotak ini akan membuat membran pada *buzzer* bergetar sehingga menghasilkan suara dengan frekuensi yang sama, 5 kHz. Namun ketika Arduino memberi input logika 0, IC 555 akan *me-reset* karena pin ini merupakan *active low*, sehingga output dari rangkaian osilator adalah 0 V. Dengan memberi logika 0 dan 1 secara bergantian sudah dapat menghasilkan satu buah sinyal *beep*. Dengan kata lain apabila ingin mendapat sinyal dengan jumlah *beep* 5 maka pemberian logika 0 dan 1 dilakukan berulang-ulang sebanyak lima kali dalam satu kali periode pengiriman sinyal.

Berikut ini data tabel pengujian respon frekuensi dari *filter* dengan input sinyal sinusoidal menggunakan *function generator*.

Tabel 4.3 Respon frekuensi BPF terhadap tegangan

| Frekuensi (Hz) | V _{pp} input (V) | V _{pp} output (mV) |
|----------------|---------------------------|-----------------------------|
| 500 | 2,08 | 20 |
| 1000 | 2,08 | 59,2 |
| 1500 | 2,08 | 122 |
| 2000 | 2,04 | 178 |
| 2500 | 2,04 | 248 |
| 3000 | 2,04 | 312 |
| 3500* | 2,04 | 360 |
| 4000 | 2,04 | 396 |
| 4500 | 2,04 | 432 |
| 5000 | 2,04 | 448 |
| 5500 | 2,04 | 456 |
| 6000 | 2,04 | 456 |
| 6500 | 2,04 | 448 |
| 7000 | 2,04 | 432 |
| 7500 | 2,00 | 424 |
| 8000 | 2,00 | 472 |
| 8500 | 2,00 | 472 |
| 9000 | 2,00 | 456 |
| 9500 | 2,00 | 440 |
| 10000* | 2,00 | 340 |



Gambar 4.6 Perbandingan antara sinyal input dan output pada frekuensi *cutoff* BPF

Salah satu peran penting dari perangkat penerima adalah mem-*filter* sinyal yang masuk untuk meminimalisir *noise* dan gangguan lainnya yang menyebabkan kesalahan dalam proses komunikasi. Pada bab perancangan sebelumnya dijelaskan jenis filter yang digunakan adalah BPF orde 4 pada frekuensi 5,8 kHz. Penguatan total dari realisasi rangkaian BPF adalah 1/4 kali, sehingga apabila diberi sinyal 2 Vpp maka output dari filter pada rentang frekuensi BPF akan sama dengan 500 mV. Untuk mencari frekuensi *cutoff* dari tabel 4.3 dapat dicari tegangan *cutoff* outputnya.

$$V_{cutoff} = 0,707 \times 500 = 353,5 \text{ mV}$$

Pada tabel 4.3, nilai frekuensi yang ditandai dengan (*) merupakan frekuensi yang terdekat dengan frekuensi *cutoff*. Sehingga poin penting dari tabel tersebut adalah realisasi rangkaian BPF memiliki rentang frekuensi dari 3,5 kHz – 10 kHz.

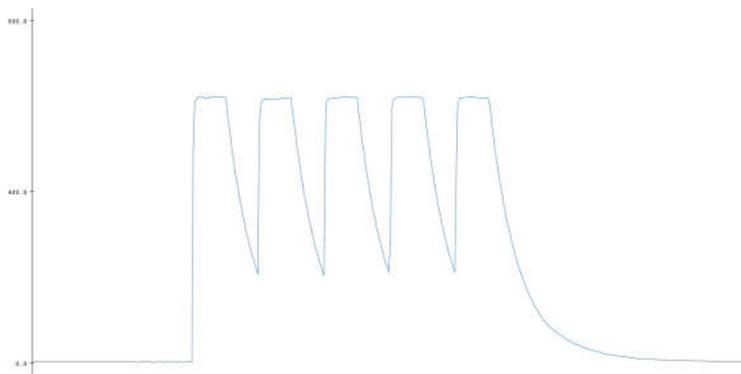
Kemudian pengujian berikutnya adalah dari rangkaian penerima keseluruhan. Setelah rangkaian di-*filter* dan dikuatkan, sinyal dibawa menuju rangkaian *envelope detector* untuk memberi selubung pada sinyal sehingga memudahkan Arduino dalam menghitung nilai puncak dan jumlah *beep*. Pengujian dilakukan dengan mengamati *plot* sinyal yang diterima arduino ketika perangkat penerima diberi sinyal 5 *beep*. Tabel

4.4 berikut menyajikan pengaruh jarak terhadap nilai puncak dan juga pembacaan jumlah *beep*.

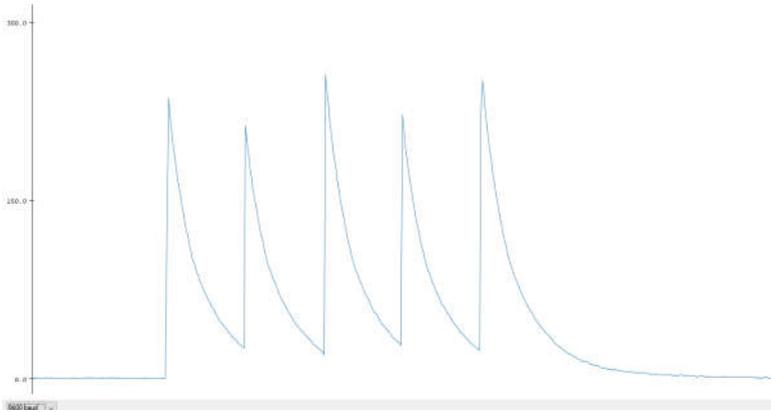
Tabel 4.4 Pengaruh jarak terhadap pembacaan sinyal

| Jarak (cm) | Tx | Rx | |
|------------|-------------|------------------|-------------|
| | <i>Beep</i> | Nilai puncak (V) | <i>Beep</i> |
| 25 | 5 | 2,44 | 5 |
| 50 | 5 | 1,47 | 5 |
| 75 | 5 | 1,30 | 5 |
| 100 | 5 | 0,99 | 5 |
| 125 | 5 | 0,94 | 5 |
| 150 | 5 | 0,88 | 5 |
| 175 | 5 | 0,63 | 8 |
| 200 | 5 | 0,38 | 10 |

Dari grafik 4.4 diatas diperoleh kesimpulan dimana komunikasi optimum antar dua robot berada pada rentang jarak hingga 150 cm. Pada jarak demikian informasi berupa *beep* antara yang dikirim dengan yang diterima sama. Semakin jauh jarak komunikasi maka sinyal yang diterima akan semakin lemah. Diluar daripada jarak 150 cm, perangkat penerima tidak dapat membaca sinyal *beep* secara benar. Pada tabel ditunjukkan dengan jumlah pembacaan *beep* yang berbeda. Visualisasi sinyal hasil pengujian dapat dilihat pada gambar 4.7, 4.8 dan 4.9 berikut.

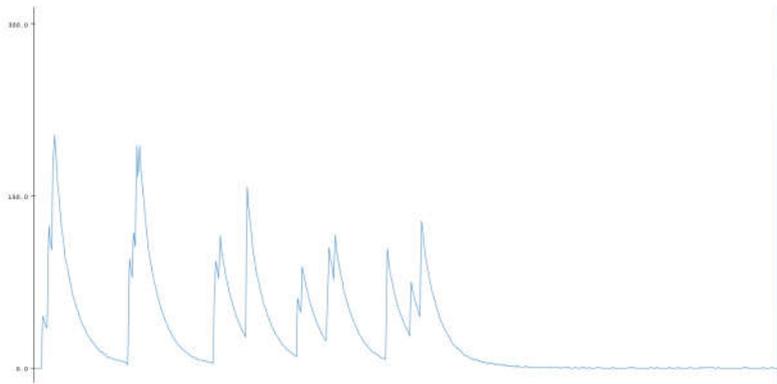


Gambar 4.7 Bentuk sinyal diterima pada jarak 25 cm



Gambar 4.8 Bentuk sinyal yang diterima pada jarak 100 cm

Dari gambar 4.7 dan 4.8 terlihat jelas pengaruh sinyal ketika dikirim pada jarak yang berbeda. Pada jarak 30 cm bentuk sinyal diterima mendekati gelombang kotak dengan nilai puncak sebesar 498 pembacaan ADC (2,44 V). Sedangkan pada saat berjarak 100 cm bentuk sinyal lebih lancip dengan nilai puncak sebesar 193 pembacaan ADC (0,99 V). Kedua gambar tersebut juga memiliki hubungan dengan data pada tabel 4.4. Pada jarak 175 cm dan 200 cm menunjukkan kesalahan pembacaan oleh Arduino. Berikut adalah bentuk sinyal yang diperoleh Arduino pada jarak 175 cm..



Gambar 4.9 Bentuk sinyal yang diterima pada jarak 175 cm

Kesalahan pembacaan ini dipengaruhi oleh *losses* yang dialami sinyal sebelum ditangkap oleh perangkat penerima. Ketika jarak antara pengirim dan penerima lebih dekat, *losses* yang dialami sinyal saat di udara lebih kecil sehingga bentuk sinyal yang diterima lebih mendekati dengan sinyal yang dikirim. Sedangkan ketika jaraknya lebih jauh maka *losses* yang dialami juga lebih besar sehingga amplitudo sinyal yang diterima juga lebih kecil.

4.3. Pengujian Sensor Jarak

Pengujian sensor jarak HC-SR04 dilakukan dengan menguji keseuaian pembacaan sensor dengan nilai pembacaan langsung melalui penggaris. Dan juga dalam pengujian ini menguji pembacaan jarak dari beberapa sudut yang telah ditentukan. Tujuan dari pengujian ini adalah untuk mengetahui karakteristik dari sensor HC-SR04 yang kegunaannya pada *swarm robot* ini adalah untuk mendeteksi jarak agar terhindar dari tabrakan atau benturan dengan tembok atau objek lainnya. Hasil pengujian disajikan pada tabel 4.5 berikut.

Tabel 4.5 Pengaruh sudut pada pembacaan jarak

| Sudut pembacaan | 10 cm | 20 cm | 30 cm |
|------------------------|--------------|--------------|--------------|
| Tegak lurus | 10 cm | 20 cm | 30 cm |
| 10 derajat | 10 cm | 20 cm | 31 cm |
| 20 derajat | 9 cm | 22 cm | 31 cm |
| 30 derajat | 56 cm | 0 cm | 0 cm |



Gambar 4.10 Pengujian terhadap sudut pembacaan 20 derajat



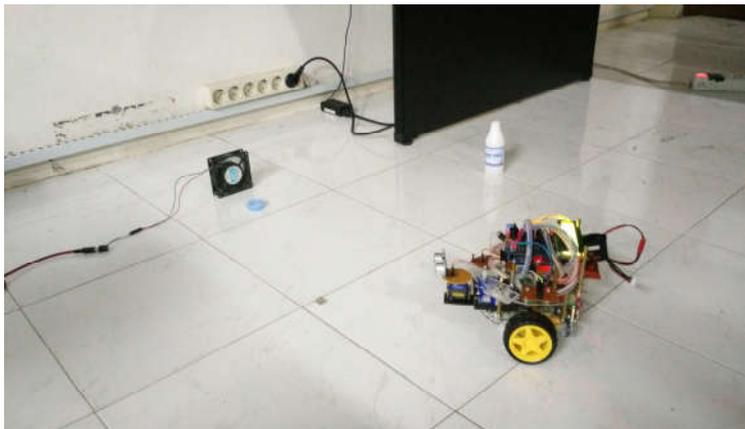
Gambar 4.11 Pengujian terhadap sudut pembacaan 30 derajat

Hasilnya dapat dilihat pada tabel 4.5, untuk pembacaan sensor dengan sudut tegak lurus dengan objek di depannya nilai pembacaan sudah sesuai dengan pengukuran jarak aslinya menggunakan penggaris. Kemudian diuji dengan menyimpang 10° dan 20° dari sumbu tegak lurus, hasilnya pembacaan sensor masih sesuai dan mendekati dengan pengukuran aslinya. Namun ketika sudut simpangan diubah menjadi 30° , dalam jarak lebih besar dari 10 cm menunjukkan *error* yang sangat besar, atau sensor tidak dapat membaca jarak sehingga pembacaannya menjadi 0. Hal ini dikarenakan pada sudut lebih besar dari sama dengan 30° gelombang ultrasonik yang dipancarkan oleh *transmitter* memantul ke arah lain sehingga *receiver* tidak menerima pantulan gelombang ultrasonik. Hal ini juga telah disebutkan pada *datasheet* HC-SR04 yakni sudut pengukuran maksimum sebesar 30° .

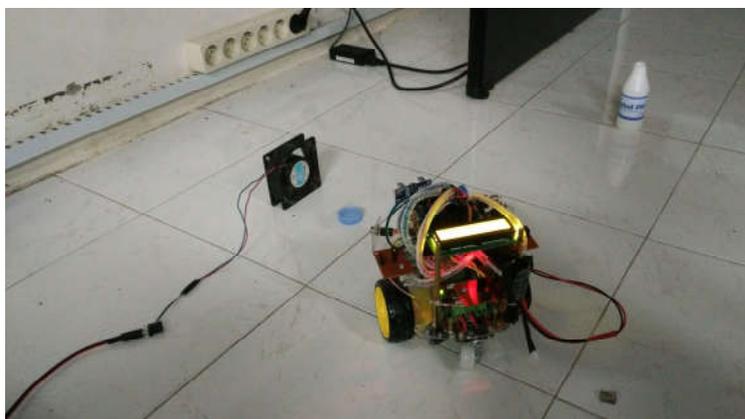
4.4. Pengujian Gerak Robot *Master*

Pengujian gerak robot *master* merupakan pengujian ketika robot mencari sumber gas. Pengujian dilakukan dengan memberikan gas yang dibantu dengan kipas ke suatu arah, selanjutnya robot diletakkan secara

acak dalam radius sekitar 2 m dari sumber gas. Robot akan bergerak dan berusaha mendekati sumber gas.



Gambar 4.12 Sensor gas kanan robot mendapat intensitas lebih besar, robot berbelok ke kanan



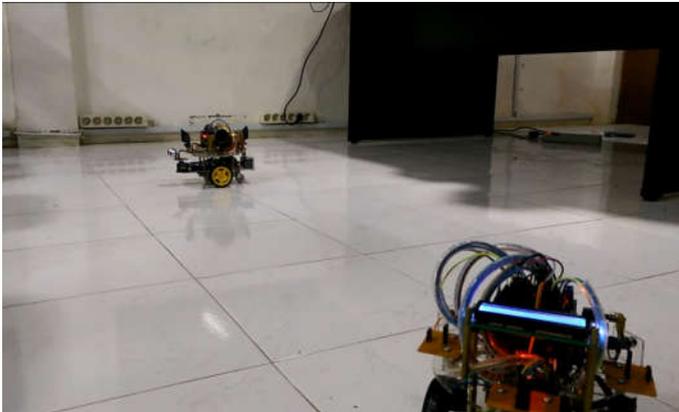
Gambar 4.13 Robot berhasil mendekati sumber gas

Dari pengujian gerak master yang dilakukan masih mengalami kendala *error* dalam eksekusi gerak dikarenakan letak kedua sensor pada robot cukup dekat sehingga ketika menentukan arah kanan, kiri atau lurus masih terjadi kesalahan. Tapi secara keseluruhan robot sudah dapat

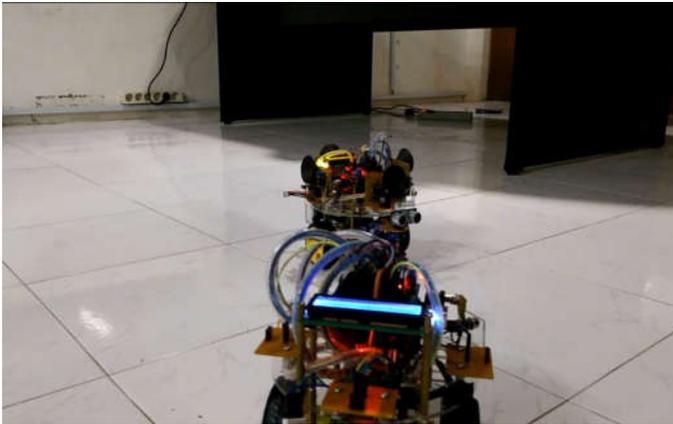
mengetahui arah, membandingkan kedua sensor dan bergerak mendekati sumber gas.

4.5. Pengujian Gerak Robot *Slave*

Pengujian gerak *slave* ini bertujuan untuk menguji gerak robot mendekati robot lainnya ketika berperan sebagai *slave*. Cara pengujiannya adalah dengan menempatkan satu robot *master* yang statis pada tempatnya, lalu robot *slave* akan berusaha mendekati robot *master*. Robot *master* hanya berperan sebagai pemberi sinyal berupa suara *beep* dan selanjutnya tugas robot *slave* untuk membaca arah datangnya suara dan bergerak mendekat. Hasil pengujian dapat dilihat pada gambar berikut.



Gambar 4.14 Robot *slave* berada pada *range* komunikasi dan berusaha mendekati robot *master*



Gambar 4.15 Robot *slave* semakin mendekati robot *master*

Hasil pengujian gerak robot *slave* ini adalah secara umum robot sudah berhasil bergerak perlahan mendekati robot *master*, dan pembacaan arahnya sudah cukup akurat. Namun terkadang masih terjadi kesalahan dimana seharusnya robot bergerak ke kanan, tetapi berkebalikan. Hal ini dikarenakan adanya pantulan suara dari benda-benda sekitar yang mengganggu navigasi dari robot seperti tembok dan meja yang ada pada gambar. Selain itu robot juga masih merespon terhadap suara-suara hentakan atau suara sekitar yang tidak termasuk dalam sinyal karena volume suara yang didengar terlalu besar sehingga tidak cukup dilemahkan oleh *filter*. Namun disamping itu tujuan dari sistem komunikasi suara dengan meminimalisir dampak suara gangguan dari manusia sudah cukup berhasil.

4.6. Pengujian Gerak Robot Keseluruhan

Pengujian keseluruhan dilakukan dengan mengkombinasikan antara pengujian gerak robot *master* dan gerak robot *slave*. Awalnya robot bergerak sebagai robot *master*. Lalu robot A diberikan gas alkohol 96% dibantu dengan kipas untuk mengalirkan gas. Karena robot A menerima intensitas gas yang lebih besar maka robot A berperan sebagai *master* dan robot B berperan sebagai *slave*. Kemudian arah aliran gas diganti menjadi ke arah robot B sehingga saat itu robot B beralih dari *slave* menjadi *master* sedangkan robot A berganti menjadi *slave*.

Hasil dari pengujian hingga saat ini masih belum sempurna, dimana ketika robot A diberikan aliran gas maka robot A berperan sebagai

master dan robot B berperan sebagai *slave*. Robot A sudah berhasil bergerak mendekati arah sumber gas dan robot B sudah berhasil bergerak mendekati robot A. Masih terjadi *error* ketika robot B berusaha mendekati robot A. Terkadang robot B bergerak berlawanan arah yang menyebabkan robot B bergerak menjauhi robot A. Hal ini dikarenakan adanya pantulan suara dari lingkungan sekitar sehingga Arduino salah dalam membaca nilai puncak sinyal dan membuat robot menghasilkan kesimpulan yang salah.

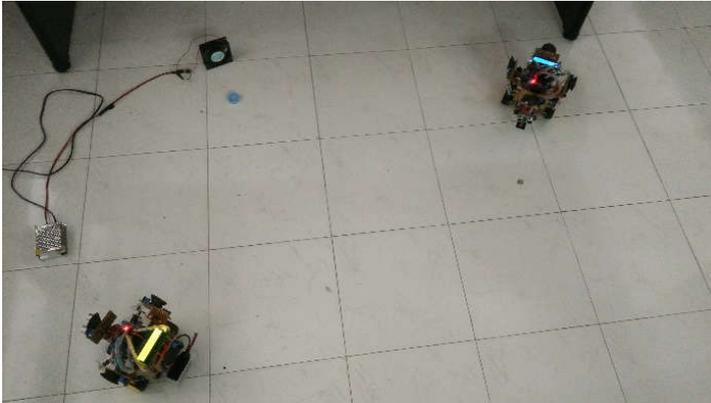


Gambar 4.16 Posisi awal pengujian robot, robot A (lampu hijau) dan robot B (lampu biru)



Gambar 4.17 Robot B diberikan aliran gas dan berperan sebagai *master*

Kemudian ketika selanjutnya aliran gas dipindah ke robot B, maka robot B berperan sebagai *master* dan robot A berganti sebagai *slave*. Sama seperti sebelumnya, masih terjadi *error* dimana robot A terkadang bergerak menjauhi robot B, namun secara keseluruhan robot sudah berhasil mendekati robot.



Gambar 4.18 Robot A diberikan aliran gas, terjadi pergantian mode dari *slave* ke *master*

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang bisa diambil dari tugas akhir ini adalah diperoleh 2 buah *swarm robot* dengan sistem komunikasi suara menggunakan Arduino UNO, sensor HC-SR04, sensor TGS2600, motor servo, mikrofon kondenser, dan *buzzer*. *Swarm robot* yang dirancang memiliki jarak optimum komunikasi antar robot sejauh 150 cm. Robot dapat mendeteksi adanya gas hingga rentang jarak maksimum 150 cm. Disamping itu, sensitivitas sensor juga dipengaruhi oleh jenis gas yang dideteksi. Dari 3 bahan yang diuji, sensitivitas tertinggi ke terendah secara berurutan yakni bensin, alkohol 96% dan spiritus. Sensor pendeteksi jarak mampu mendeteksi dengan benar hingga kemiringan 20 derajat dengan error yang masi bisa ditoleransi sekitar 1cm – 2 cm. Pada pengujian keseluruhan sudah sesuai dengan kriteria perancangan dimana ketika robot A diberi aliran gas, robot A mendapat intensitas gas lebih besar dan menjadi *master* bergerak mengikuti arah datangnya gas sedangkan robot B sebagai *slave* sudah berhasil bergerak mengikuti robot A. Sebaliknya ketika aliran gas diberikan ke robot B, maka robot B yang awalnya sebagai *slave* mendapat intensitas yang lebih tinggi sehingga berganti menjadi *master*, bergerak mengikuti arah aliran gas. Robot A yang sebelumnya sebagai *master* berganti menjadi *slave* dan sudah berhasil bergerak mendekati robot B. Namun masih sering terjadi *error* dalam pengujian dimana ketika robot *slave* mengikuti robot *master* terkadang robot mengarah ke arah yang berlawanan sehingga robot *slave* tidak langsung mendekati robot *master*.

5.2. Saran

Saran yang dapat diambil agar tujuan utama dari sistem yang dibuat pada tugas akhir ini tercapai maka dalam perancangan perlu menggunakan lebih dari 2 robot untuk dapat membandingkan performa masing-masing sistem.

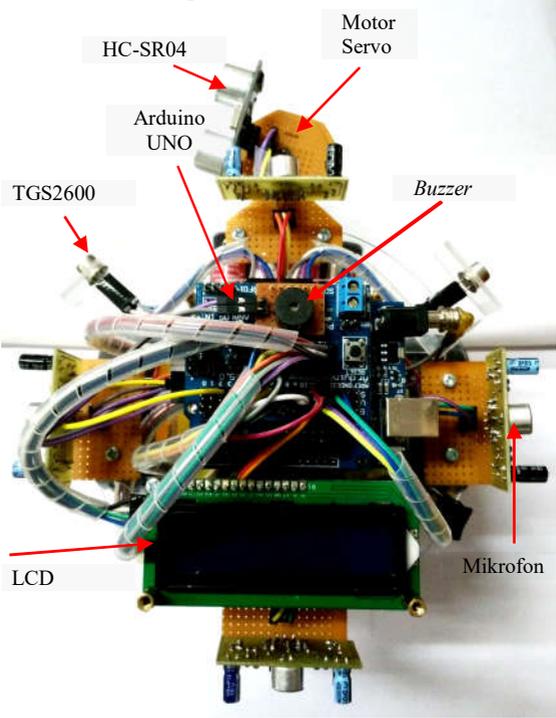
Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Sun Kim. J, Hwan Tak. M, dan Hoon Joo. Y. 2012. "Behaviour Control for the Swarm Robots with Network Connectivity," International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), vol. 9, pp. 337.
- [2] Moss, Richard. 2014. "Low-cost Autonomous Robots Replicate Swarming Behaviour". < <http://newatlas.com/colias-swarm-robot/33897/>>. (Diakses 1 Desember 2016).
- [3] Jovtic. A, Andina. D. 2007. "Swarm Intelligence and Its Applications in Swarm Robotics". International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, vol. 6, pp. 43.
- [4] Parker. L.E, 2008. "Multiple Mobile Robot Systems," Springer Handbook of Robotics, Springer, Ed. By B. Siciliano and O. Khatib, pp. 921-936.
- [5] Campo. A, dan Dorigo. M. 2007. "Efficient Multi Foraging in Swarm Robotics". Lecture Notes in Artificial Intelligence, vol. 4648, pp. 696-705.
- [6] Jovtic. A, Andina. D. 2007. "Swarm Intelligence and Its Applications in Swarm Robotics". International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, vol. 6, pp. 44.
- [7] Sungju. H, Seongsoo. H, dan Joonghyun. L. 2013. "Energy-efficient Distributed Programming Model for Swarm Robot". International Conference on Control, Automation and Systems (ICCAS), vol. 13, pp. 300.
- [8] Arduino. "Arduino/Genuino UNO". <<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. (Diakses 30 Agustus 2016).
- [9] Safari. "Arduino Technical Details". <<https://www.safaribooksonline.com/library/view/arduino-a-technical/9781491934319/ch04.html>>. (Diakses 1 Desember 2017).
- [10] Figaro. 1999. "MOS Type Operating Principle". <<http://www.figaro.co.jp/en/technicalinfo/principle/mos-type.html>>. (Diakses 30 Agustus 2016).
- [11] Figaro. 2008. "TGS 2600 – for The Detection of Air Contaminants". TGS2600 Datasheet, pp. 2.
- [12] David. M. 2006. "The Principle of Operation of Capacitor (Condenser) Microphone," <<http://www.audiomasterclass.com/the->

- principle-of-operation-of-a-capacitor-condenser-microphone>, (Diakses 30 Agustus 2016).
- [13] Gunawan, Ngurah. 2015. “Aplikasi Sensor Kompas dan Sensor Jarak Pada Kacamata Bagi Kaum Tuna Netra Berbasis Mikrokontroler AT89S52”. (Skripsi). Denpasar.
- [14] ElecFreaks. 2011. “Ultrasonic Ranging Module HC-SR04”. HC-SR04 Datasheet, pp.1.
- [15] Praveen. 2015. “Ultrasonic Water Level Controller Using 8051”. <<http://www.circuitstoday.com/ultrasonic-water-level-controller-using-8051>>. (Diakses 1 Desember 2016).
- [16] CUI, “Piezo and Magnetic Buzzers,” <<http://www.cui.com/product-spotlight/piezo-and-magnetic-buzzers>>, (Diakses 30 Agustus 2016).
- [17] Elektronika Dasar. 2012. “Driver Motor DC H-BRIDGE Transistor”. <<http://elektronika-dasar.web.id/driver-motor-dc-h-bridge-transistor/>>. (Diakses 30 Agustus 2016).
- [18] Fahmizal. 2011. “Driver Motor DC Pada Robot Beroda dengan Konfigurasi H-BRIDGE MOSFET”. <<https://fahmizaleeits.wordpress.com/tag/driver-motor-dc-pada-robot-beroda-dengan-konfigurasi-h-bridge-mosfet/>>. (Diakses 30 Agustus 2016).
- [19] Saadi, Muhammad. 2014. “How to Drive Stepper Motors and RC Servo Motors”. <<http://aimagin.com/blog/how-to-drive-stepper-motors-and-rc-servo-motors/>>. (Diakses 1 Desember 2016).
- [20] Marcell. 2015. “Experimenting with The CD4051 (De)Multiplexer”. <<http://www.programelectronics.com/2015/10/18/experimenting-with-the-cd4051-demultiplexer/>>. (Diakses 1 Desember 2016).
- [21] Robomart. “NE555 Precision Timer IC”. <<https://www.robomart.com/ne555-precision-timer-ic>>. (Diakses 1 Desember 2016).
- [22] Electronics Tutorial, “555 Oscillator Tutorial,” <http://www.electronics-tutorials.ws/waveforms/555_oscillator.html>, (Diakses 1 Desember 2016).
- [23] Longtech Optics. 2013. “Specifications of LCD Module”. LCD 16x2 Datasheet, pp. 4.

Gambar realisasi robot (tampak atas)



Gambar realisasi 2 buah *swarm robot*



Kode program Arduino untuk driver motor:

```
void forward()
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enA, SPEED);
  analogWrite(enB, SPEED);
}
```

```
void backward()
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enA, SPEED);
  analogWrite(enB, SPEED);
}
```

```
void stop1()
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

```
void turnleft()
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enA, SPEED);
  analogWrite(enB, SPEED);
}
```

```

void turnright()
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enA, SPEED);
  analogWrite(enB, SPEED);
}

```

Kode program Arduino untuk mendeteksi jarak:

```

void distanceCallback()
{
  Serial.println(String(millis()) + " : Distance");
  servo1.write(87);
  front = sonar.ping_cm();
  Serial.println("Jarak depan:" + String(front));
  if(front >= 3 && front <= 12)
  {
    stop1();
    servo1.write(177);
    myDelay(500);
    left = sonar.ping_cm();
    Serial.println("Jarak kiri:" + String(left));
    servo1.write(5);
    myDelay(1000);
    right = sonar.ping_cm();
    Serial.println("Jarak kanan:" + String(right));
    servo1.write(87);
    myDelay(500);
  }
  if(left <= 12 && right <= 12)
  {
    backward();
    distance.delay(3 * DELAY);
    turnleft();
    distance.delay(3 * DELAY);
  }
  else if(left < right)

```

```

{
  turnright();
  distance.delay(2 * DELAY);
}
else if(left > right)
{
  turnleft();
  distance.delay(2 * DELAY);
}
}

```

Kode program Arduino untuk mengirim *beep*:

```

void beepCallback()
{
  Serial.println(String(millis()) + " : Beep");
  beep_start = millis();
  beep_elapsed = 0;
  lcd.clear();
  lcd.home();
  lcd.print("Transmitting...");
  if(intens >= 0 && intens >= rsignal)
  {
    if(intens == 0)
    {
      myDelay(T_BEEP);
    }
    else
    {
      ton = T_BEEP / (2 * intens);
      while(beep_elapsed < T_BEEP)
      {
        digitalWrite(buzzer, HIGH);
        myDelay(ton);
        digitalWrite(buzzer, LOW);
        myDelay(ton);
        beep_current = millis();
        beep_elapsed = beep_current - beep_start;
      }
    }
  }
}

```

```

    role = "master";           // Master
}
else
{
    role = "slave";           // Slave
    myDelay(T_BEEP);
}
}
}

```

Kode program Arduino untuk menerima sinyal:

```

void commCallback()
{
    Serial.println(String(millis()) + " : Communication");
    for(int m = 0; m < 4; m++)
    {
        total_peak[m] = 0;
        peak[m] = 0;
        samples[m] = 0;
    }
    previous_value = 0;
    counter = 0;
    comm_start = millis();
    comm_elapsed = 0;
    lcd.clear();
    lcd.home();
    lcd.print("Receiving...");
    while(comm_elapsed < T_COMM)
    {
        for(int i = 0; i < 4; i++)
        {
            for(int j = 0; j < 3; j++)
            {
                bagi = i / pow(2, j);
                s[j] = bagi % 2;
            }
            digitalWrite(S0, s[0]);
            digitalWrite(S1, s[1]);
            digitalWrite(S2, s[2]);
            mic[i] = analogRead(A0);
        }
    }
}

```

```

if(mic[i] >= threshold)
{
    samples[i]++;
    total_peak[i] += mic[i];
    peak[i] = total_peak[i] / samples[i];
}
if(mic[0]>mic[2] && mic[1]>mic[3])
{
    current_value = mic[1];
}
else if(mic[0]>mic[2] && mic[1]==mic[3]) {
    current_value = mic[0];
}
else if(mic[0]>mic[2] && mic[1]<mic[3]) {
    current_value = mic[3];
}
else if(mic[0]==mic[2] && mic[1]>mic[3]) {
    current_value = mic[1];
}
else if(mic[0]==mic[2] && mic[1]<mic[3]) {
    current_value = mic[3];
}
else if(mic[0]<mic[2] && mic[1]>mic[3]) {
    current_value = mic[1];
}
else if(mic[0]<mic[2] && mic[1]==mic[3]) {
    current_value = mic[2];
}
else if(mic[0]<mic[2] && mic[1]<mic[3]) {
    current_value = mic[3];
}
//=====pulse counting=====//
if(current_value > threshold && previous_value < threshold)
{
    counter++;
}
previous_value = current_value;
}
comm_current = millis();

```

```

    comm_elapsed = comm_current - comm_start;
};
rsignal = counter;
//=====location decision=====//
if(peak[0]>peak[2] && peak[1]>peak[3])
{
    direct = loc[1];
}
else if(peak[0]>peak[2] && peak[1]==peak[3])
{
    direct = loc[0];
}
else if(peak[0]>peak[2] && peak[1]<peak[3])
{
    direct = loc[7];
}
else if(peak[0]==peak[2] && peak[1]>peak[3])
{
    direct = loc[2];
}
else if(peak[0]==peak[2] && peak[1]<peak[3])
{
    direct = loc[6];
}
else if(peak[0]<peak[2] && peak[1]>peak[3])
{
    direct = loc[3];
}
else if(peak[0]<peak[2] && peak[1]==peak[3])
{
    direct = loc[4];
}
else if(peak[0]<peak[2] && peak[1]<peak[3])
{
    direct = loc[5];
}
else
{
    direct = loc[0];
}

```

```
}  
}
```

Kode program Arduino untuk menghitung intensitas gas:

```
void gasCallback()  
{  
  Serial.println(String(millis()) + " : Gas Read");  
  //TGS RIGHT channel 4 (100)  
  digitalWrite(S0, LOW);  
  digitalWrite(S1, LOW);  
  digitalWrite(S2, HIGH);  
  TGS_right = analogRead(A0);  
  //TGS LEFT channel 6 (110)  
  digitalWrite(S0, LOW);  
  digitalWrite(S1, HIGH);  
  digitalWrite(S2, HIGH);  
  TGS_left = analogRead(A0);  
  gas_max = max(TGS_left, TGS_right);  
  if(gas_max > 350)  
  {  
    intens = 5;  
  }  
  else if(gas_max > 320 && gas_max <= 350)  
  {  
    intens = 4;  
  }  
  else if(gas_max > 290 && gas_max <= 320)  
  {  
    intens = 3;  
  }  
  else if(gas_max > 260 && gas_max <= 290)  
  {  
    intens = 2;  
  }  
  else if(gas_max > 230 && gas_max <= 260)  
  {  
    intens = 1;  
  }  
  else
```

```

{
  intens = 0;
}
}

```

Kode program Arduino untuk menampilkan ke LCD:

```

void lcdCallback()
{
  Serial.println(String(millis()) + " : LCD Display");
  lcd.clear();
  lcd.home();
  lcd.print("L:");
  lcd.print(TGS_left);
  lcd.setCursor(0, 1);
  lcd.print("R:");
  lcd.print(TGS_right);
  lcd.setCursor(6, 0);
  lcd.print(role);
  lcd.setCursor(6, 1);
  lcd.print("A:");
  lcd.print(counter);
  lcd.setCursor(10, 1);
  lcd.print("B:");
  lcd.print(intens);
}

```

Kode program Arduino untuk mulai inisiasi kedua robot:

```

void syncCallback()
{
  while(condition == false)
  {
    for(int i = 0; i <= 3; i++)
    {
      for(int j = 0; j < 3; j++)
      {
        bagi = i / pow(2, j);
        s[j] = bagi % 2;
      }
      digitalWrite(S0, s[0]);
    }
  }
}

```

```

digitalWrite(S1, s[1]);
digitalWrite(S2, s[2]);
mic[i] = analogRead(A0);
}
if(mic[0]>mic[2] && mic[1]>mic[3])
{
    current_value = mic[1];
}
else if(mic[0]>mic[2] && mic[1]==mic[3])
{
    current_value = mic[0];
}
else if(mic[0]>mic[2] && mic[1]<mic[3])
{
    current_value = mic[3];
}
else if(mic[0]==mic[2] && mic[1]>mic[3])
{
    current_value = mic[1];
}
else if(mic[0]==mic[2] && mic[1]<mic[3])
{
    current_value = mic[3];
}
else if(mic[0]<mic[2] && mic[1]>mic[3])
{
    current_value = mic[1];
}
else if(mic[0]<mic[2] && mic[1]==mic[3])
{
    current_value = mic[2];
}
else if(mic[0]<mic[2] && mic[1]<mic[3])
{
    current_value = mic[3];
}
if(current_value < threshold && previous_value > threshold)
{
    condition = true;
}

```

```

    }
    previous_value = current_value;
    lcd.clear();
    lcd.home();
    lcd.print("Waiting..");
    myDelay(200);
  }
}

```

Kode program Arduino untuk penentuan *master* dan *slave*:

```

void master_slaveCallback()
{
  if(role == "master")
  {
    master_move();
  }
  else if(role == "slave")
  {
    slave_move();
  }
}

```

Kode program Arduino untuk gerak robot *slave*:

```

void slave_move()
{
  Serial.println(String(millis()) + " : Moving Slave...");
  Serial.println(direct);
  if(direct == loc[1])
  {
    myDelay(0);
    turnright();
    myDelay(150);
    stop1();
  }
  if(direct == loc[2])
  {
    myDelay(0);
    turnright();
    myDelay(300);
  }
}

```

```
    stop1();
}
if(direct == loc[3])
{
    myDelay(0);
    turnright();
    myDelay(450);
    stop1();
}
if(direct == loc[4])
{
    myDelay(0);
    turnright();
    myDelay(600);
    stop1();
}
if(direct == loc[5])
{
    myDelay(0);
    turnleft();
    myDelay(450);
    stop1();
}
if(direct == loc[6])
{
    myDelay(0);
    turnleft();
    myDelay(300);
    stop1();
}
if(direct == loc[7])
{
    myDelay(0);
    turnleft();
    myDelay(150);
    stop1();
}
forward();
myDelay(1000);
```

```

    stop1();
}

```

Kode program Arduino untuk gerak robot *master*:

```

void master_move()
{
    Serial.println(String(millis()) + " : Moving Master...");
    signed int sensor_left = TGS_left - prev_left;
    signed int sensor_right = TGS_right - prev_right;
    if(sensor_left - sensor_right > 2)
    {
        turnleft();
        myDelay(2 * DELAY);
        stop1();
    }
    else if(sensor_right - sensor_left > 2)
    {
        turnright();
        myDelay(2 * DELAY);
        stop1();
    }
    else
    {
        forward();
    }
    forward();
    myDelay(1000);
    stop1();
    prev_left = TGS_left;
    prev_right = TGS_right;
}

```

Kode program utama robot:

```

void setup()
{
    servo1.attach(10);
    Serial.begin(9600);
    pinMode(buzzer, OUTPUT);
    pinMode(S0, OUTPUT);
}

```

```
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
lcd.begin (16,2);
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);
robot.addTask(sync);
robot.addTask(comm);
robot.addTask(beep);
robot.addTask(master_slave);
robot.addTask(distance);
robot.addTask(gas);
robot.addTask(lcd_display);
robot.enableAll(true);
}
void loop()
{
  robot.execute();
}
```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Putu Agus Antara Adiputra lahir di Klungkung, Bali tanggal 6 September 1994 yang merupakan anak pertama dari dua bersaudara. Penulis menyelesaikan pendidikan dasar di SDPN Tulangampian Denpasar dan dilanjutkan dengan pendidikan menengah di SMPN 2 Denpasar dan SMAN 4 Denpasar. Karena latar belakang ayah yang dulunya sering memperbaiki dan membongkar alat-alat elektronika membuat penulis memutuskan untuk melanjutkan pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun 2012 dengan mengambil bidang studi Elektronika. Selama kuliah, penulis berkontribusi dalam kegiatan kerohanian mahasiswa Hindu TPKH-ITS dan juga sebagai asisten laboratorium Elektronika Dasar B202.

Email: antara.adiputra@gmail.com

Halaman ini sengaja dikosongkan