



TUGAS AKHIR - EE 184801

**KONTROL PERGERAKAN *OMNI-DIRECTIONAL WHEELS*  
*ROBOT* BERDASAR CITRA OBYEK**

Moch. Sudrajat Dwi Rizki  
NRP 0711124000044

Dosen Pembimbing  
Dr. Muhammad Rivai, ST., MT.

DEPATERMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019





**TUGAS AKHIR - EE 184801**

**KONTROL PERGERAKAN *OMNI-DIRECTIONAL WHEELS*  
*ROBOT* BERDASAR CITRA OBYEK**

Moch. Sudrajat Dwi Rizki  
NRP 0711124000044

Dosen Pembimbing  
Dr. Muhammad Rivai, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019





**FINAL PROJECT - EE 184801**

***MOVEMENT CONTROL OF OMNI-DIRECTIONAL WHEELS ROBOT BASED  
ON OBJECT IMAGE***

Moch. Sudrajat Dwi Rizki  
NRP 0711124000044

Supervisor  
Dr. Muhammad Rivai, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2019



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Kontrol Pergerakan *Omni-Directional Wheels Robot* berdasar Citra Obyek**" 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, Januari 2019



Moch. Sudrajat Dwi Rizki  
NRP. 0711124000044

.....*Halaman ini sengaja dikosongkan*.....



**KONTROL PERGERAKAN *OMNI-DIRECTIONAL WHEELS*  
ROBOT BERDASAR CITRA OBYEK**

**TUGAS AKHIR**

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

**Pada  
Bidang Studi Elektronika  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing**



**Dr. Muhammad Rivai, S.T., M.T.**  
**NIP. 196904261994031003**

**SURABAYA  
JANUARI, 2019**



.....*Halaman ini sengaja dikosongkan*.....

## **KONTROL PERGERAKAN *OMNI-DIRECTIONAL WHEELS* ROBOT BERDASAR CITRA OBYEK**

Nama : Moch. Sudrajat Dwi Rizki  
Pembimbing : Dr. Muhammad Rivai, S.T., M.T.

### **ABSTRAK**

Kebakaran merupakan salah satu kecelakaan yang dapat menyebabkan kerugian besar. Kebakaran dapat disebabkan oleh kebocoran gas yang mudah terbakar. Gas yang mudah terbakar dapat ditemukan di media seperti botol sehingga proses pendeteksian gas yang mudah terbakar dapat dilakukan dengan menggunakan kamera untuk menangkap gambar objek di mana ada gas yang mudah terbakar. Dalam penelitian ini, kontrol roda omni-directional robot berdasarkan gambar objek dilakukan pada platform segitiga dengan tiga roda omni yang dilengkapi dengan sensor gas Figaro TGS2600, kamera, dan Raspberry Pi 3 Model B. Gambar yang diambil oleh kamera adalah digunakan untuk deteksi objek dengan metode RGB dan HSV yang dinormalisasi. Sistem ini mampu mendeteksi gambar objek di lingkungan dengan intensitas cahaya di kisaran 0,6 - 57,9 lux. Sistem tidak dapat mendeteksi gambar objek di lingkungan dengan intensitas cahaya di kisaran 0-0,2 lux. Ukuran tinggi gambar botol (piksel) berkurang dengan meningkatnya jarak dari robot. Sensor gas mampu mendeteksi alkohol gas dalam kisaran 10-50cm. Dalam pengujian keseluruhan dengan posisi objek pada sudut 90, jarak 3 meter, dan intensitas cahaya 60,9 lux, tingkat keberhasilan gerakan robot menuju objek dan berhenti di dekat objek adalah 100%. Tingkat keberhasilan robot untuk mendeteksi gas alkohol adalah 80%.

Kata kunci: *omni-directional wheels robot, normalized RGB, HSV, Raspberry Pi 3 Model B*

.....*Halaman ini sengaja dikosongkan*.....

## ***MOVEMENT CONTROL OF OMNI-DIRECTIONAL WHEELS ROBOT BASED ON OBJECT IMAGE***

*Name* : Moch. Sudrajat Dwi Rizki  
*Supervisor* : Dr. Muhammad Rivai, S.T., M.T.

### ***ABSTRACT***

*Fire is one of the accidents that can cause large losses. Fires can be caused by leaks of flammable gas. Flammable gas can be found in media such as bottles so that the process of detecting flammable gases can be carried out using a camera to capture image of object where combustible gases are present. In this study, control of the robot's omni-directional wheels based on object image was carried out on a triangular platform with three omni wheels equipped with Figaro TGS2600 gas sensor, camera, and Raspberry Pi 3 Model B. The image captured by the camera was used for object detection with the Normalized RGB and HSV method. The system is able to detect object images in the environment with light intensity in the range 0.6 - 57.9 lux. The system cannot detect object images in the environment with light intensity in the range 0-0.2 lux. The height size of the bottle image (pixel) decreases with increasing distance from the robot. The gas sensor is able to detect gas alcohol in the range of 10-50cm. In the overall test with the position of the object at an angle of 90, a distance of 3 meters, and light intensity of 60.9 lux, the success rate of the robot movement goes to the object and stops near the object is 100%. The success rate of the robot to detect alcohol gas is 80%.*

*Keywords: omni-directional wheels robot, normalized RGB, HSV, Raspberry Pi 3*

.....*Halaman ini sengaja dikosongkan*.....

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga penelitian dengan judul **“KONTROL PERGERAKAN *OMNI-DIRECTIONAL WHEELS ROBOT* BERDASAR CITRA OBYEK”** dapat terselesaikan dengan baik. Penelitian ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa dalam menyelesaikan penelitian ini banyak mengalami kendala, namun berkat bantuan, bimbingan dan kerjasama dari berbagai pihak kendala tersebut dapat diatasi. Oleh karena itu, penulis ingin menyampaikan banyak terima kasih dan rasa hormat setinggi-tingginya kepada:

1. Bapak Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing tugas akhir atas bimbingan dan motivasi selama ini.
2. Bapak Dr. Ir. Djoko Purwanto, M.Eng., Dr.Eng. Ir. Totok Mujiono M.Kom., Ronny Mardiyanto, ST., MT., Ph.D., dan Ir. Haris Pirngadi, MT. selaku dosen penguji sidang tugas akhir yang memberikan masukan dan pengarahan yang berguna bagi penulis dalam menyelesaikan buku tugas akhir ini dengan baik.
3. Bapak Dr. Eng Ardyono Priyadi, S.T., M.Eng. selaku Ketua Departemen Teknik Elektro Fakultas Teknologi Elektro-ITS.
4. Keluarga penulis yang selalu mendoakan penulis agar diberi kelancaran dalam mengerjakan tugas akhir ini.
5. Teman-teman Laboratorium Elektronika Industri B402 yang selalu membantu proses pengerjaan tugas akhir ini.

Penulis menyadari bahwa pembuatan dan penyusunan penelitian ini masih jauh dari kesempurnaan, untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun sebagai bentuk perbaikan untuk penelitian ini kedepannya. Penulis berharap semoga proyek akhir ini bisa bermanfaat bagi penulis khususnya dan bagi pembaca pada umumnya.

Surabaya, Januari 2019

Moch. Sudrajat Dwi Rizki

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	i
TUGAS AKHIR.....	iii
ABSTRAK.....	v
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Penulisan .....	4
1.7 Relevansi.....	4
TINJAUAN PUSTAKA .....	5
2.1 Omni-Directional Wheels Robot .....	5
2.2 Sensor Gas Figaro TGS 2600.....	7
2.3 Kamera.....	8
2.4 Normalisasi RGB .....	9
2.5 Ruang Warna <i>HSV (Hue, Saturation, and Value)</i> .....	10
2.6 Raspberry Pi 3 Model B.....	11
2.7 ADC ADS1115 .....	12
2.8 Motor DC.....	13
2.9 <i>H-Bridge MOSFET Drivers</i> .....	14
2.10 Kontroller Proporsional .....	14
2.11 OpenCV .....	15
2.12 Image Morphology.....	15
2.13 Find Contours.....	16
2.14 Motor Servo DC.....	17
PERANCANGAN SISTEM .....	19
3.1 Diagram Blok Sistem .....	19
3.2 Perancangan Mekanik .....	20
3.3 Perancangan Elektronik.....	21
3.4 Perancangan Perangkat Lunak .....	25
PENGUJIAN DAN ANALISIS .....	27

4.1 Pengujian Deteksi Citra Obyek .....	29
4.2 Pengujian Sensor Gas Figaro TGS 2600 .....	37
4.3 Pengujian Gerakan Robot .....	39
4.4 Pengujian Keseluruhan Sistem .....	41
PENUTUP .....	43
5.1 Kesimpulan .....	43
5.2 Saran .....	43
DAFTAR PUSTAKA .....	45
LAMPIRAN A .....	47
LAMPIRAN B .....	49
LAMPIRAN C .....	51
LAMPIRAN D .....	52
BIODATA PENULIS .....	63

## DAFTAR GAMBAR

Gambar 2.1 <i>Non-Holonomic Mobile Robot dan Holonomic Mobile Robot</i> .....	5
Gambar 2.2 Gambaran Roda Omni .....	5
Gambar 2.3 Robot omni dengan platform segitiga .....	6
Gambar 2.4 Model level penghalang saat tidak ada gas yang terdeteksi (kiri) dan saat ada gas yang terdeteksi (kanan) .....	7
Gambar 2.5 Struktur sensor <i>TGS</i> .....	8
Gambar 2.6 Rangkaian Pengukuran Dasar Sensor Gas .....	8
Gambar 2.7 Ruang Warna RGB .....	9
Gambar 2.8 Penjelasan HSV .....	11
Gambar 2.9 Board Raspberry Pi 3 Model B .....	12
Gambar 2.10 ADC ADS1115 .....	12
Gambar 2.11 Motor DC dengan sikat karbon .....	13
Gambar 2.12 Rangkaian H-Bridge MOSFET Drivers .....	14
Gambar 2.13 Dilasi : mengambil nilai maksimum di bawah kernel B..	16
Gambar 2.14 Erosi : mengambil nilai minimum di bawah kernel B .....	16
Gambar 2.15 (Atas) Citra tes untuk cvFindContour: (bawah) kontur jenis kontur (garis putus) juga <i>hole</i> (garis titik).....	17
Gambar 2.16 Servo SG-90.....	18
Gambar 2.17 Lebar pulsa sinyal servo .....	18
Gambar 3.1 Diagram Blok Sistem .....	20
Gambar 3.2 Desain Robot Omni .....	21
Gambar 3.3 Rangkaian Catu Daya .....	21
Gambar 3.8 Diagram Alir ( <i>flowchart</i> ) Sistem.....	26
Gambar 3.9 Diagram Blok Sistem Pengaturan .....	26
Gambar 4.1 Realisasi desain robot <i>omni-directional wheels</i> . .....	27
Gambar 4.2 Realisasi rangkaian catu daya.....	27
Gambar 4.3 Realisasi rangkaian <i>H-Bridge MOSFET-Driver</i> .....	28
Gambar 4.4 Realisasi rangkaian pengukuran dasar TGS 2600 .....	28
Gambar 4.5 Kalibrasi dengan <i>trackbar</i> .....	29
Gambar 4.6 Urutan Pengolahan Citra .....	30
Gambar 4.7 Citra awal .....	30
Gambar 4.8 Citra obyek setelah normalisasi RGB.....	31
Gambar 4.9 Citra hasil konversi RGB ke HSV .....	31
Gambar 4.10 Citra setelah <i>threshold</i> .....	32
Gambar 4.11 Citra setelah erosi .....	32
Gambar 4.12 Citra setelah dilasi .....	33

Gambar 4.13 Citra setelah proses erosi dan dilasi .....	33
Gambar 4.14 Kotak untuk tanda deteksi obyek .....	34
Gambar 4.15 Citra obyek berhasil dideteksi .....	35
Gambar 4.16 Citra obyek gagal dideteksi .....	35
Gambar 4.17 <i>Blob</i> tidak terdeteksi saat citra obyek gagal dideteksi .....	36
Gambar 4.18 Pengukuran tinggi botol terhadap jarak .....	37
Gambar 4.19 Pengujian TGS 2600 .....	37
Gambar 4.20 Respon sensor gas 1 (a), gas 2 (b), dan gas 3 (c) pada jarak 10 cm. ....	39
Gambar 4.21 Pengujian arah gerakan robot.....	40

## DAFTAR TABEL

Tabel 3.1 Tabel Kebenaran Gerakan Motor.....	22
Tabel 3.2 Penggunaan pin Raspberry Pi 3 untuk penelitian ini.....	24
Tabel 3.3 Spesifikasi kamera Logitech C170.....	25
Tabel 3.4 Spesifikasi servo SG-90.....	25
Tabel 4.1 Pengujian pada intensitas cahaya berbeda .....	34
Tabel 4.3 Pengukuran tinggi botol dengan jarak berbeda .....	36
Tabel 4.4 Pengujian kecepatan motor DC RS-545SH-2860 .....	39
Tabel 4.5 Pengujian arah gerakan robot.....	40
Tabel 4.6 Pengujian keseluruhan dengan jarak 3 meter, sudut 90, dan 60,9 lux pada <i>background</i> tidak hijau .....	41

.....*Halaman ini sengaja dikosongkan*.....

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kebakaran merupakan salah satu kecelakaan yang dapat menimbulkan kerugian yang besar. Proses tersulutnya api dapat dijelaskan dengan teori segitiga api. Teori segitiga api menjelaskan bahwa api akan tercipta saat terpenuhinya tiga komponen yaitu oksigen, panas, dan bahan bakar. Bahan bakar dapat berupa bahan bakar minyak (BBM) dan bahan bakar gas (BBG). Sebagian besar, kebakaran oleh ledakan disebabkan karena gas yang bocor terperangkap di ruangan dan terakumulasi hingga menyebabkan ledakan hingga kebakaran yang besar. Keberadaan gas bocor dapat dideteksi oleh sensor gas tertentu.

Kebakaran oleh kebocoran gas yang mudah terbakar dapat dicegah secara dini dengan mengetahui keberadaan suatu gas mudah terbakar pada suatu tempat. Gas berbahaya yang mudah terbakar dapat ditemukan pada media seperti botol sehingga proses investigasi gas berbahaya dapat dilakukan tidak hanya bergantung pada sensor gas saja, namun juga dapat menggunakan kamera sebagai pengenalan citra media atau obyek yang dicurigai mengandung gas berbahaya.

Pada penelitian ini, dilakukan kontrol pergerakan *Omni-Directional Wheels Robot* berdasar citra pada *platform* segitiga dengan tiga roda omni yang dilengkapi dengan sensor gas *Figaro TGS2600*, kamera, dan *Raspberry Pi 3 Model B* sebagai *processing unit*. Citra yang ditangkap kamera digunakan untuk deteksi obyek dengan metode *Normalized RGB* dan *HSV*. Motor servo *continuous* membantu kamera berputar  $360^\circ$ . Obyek yang mencurigakan didefinisikan sebagai botol. Robot bergerak pada bidang datar horizontal tanpa halangan. Navigasi robot didasarkan pada sudut lokasi citra botol yang tertangkap. Sudut didapatkan dari sensor kompas *GY-26*. Kecepatan robot diatur oleh *Proportional Controller*. Robot akan menyalakan tanda bahaya jika gas alkohol terdeteksi.

## 1.2 Perumusan Masalah

Permasalahan yang menjadi dasar dilakukannya penelitian ini dapat dirumuskan dalam poin-poin berikut:

1. Pendeteksian citra obyek oleh robot.
2. Penentuan lokasi obyek oleh robot.
3. Pengaturan pergerakan robot menuju obyek

## 1.3 Tujuan Penelitian

Penelitian ini bertujuan sebagai berikut:

1. Penggunaan metode *Normalized RGB (Red, Green, and Blue)* dan *HSV (Hue, Saturation, and Value)*.
2. Penggunaan servo SG-90 dan kamera Logitech C170 untuk penentuan lokasi obyek.
3. Penggunaan invers kinematic untuk pergerakan robot menuju obyek.

## 1.4 Batasan Masalah

Berdasarkan masalah yang telah dirumuskan maka hasil yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Robot bergerak menuju obyek tanpa halangan.
2. Obyek yang dideteksi pada penelitian ini adalah botol berwarna hijau.
3. Gas yang dideteksi pada penelitian ini adalah gas alkohol.

## 1.5 Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

### 1. Studi Literature

Studi literatur berisi serangkaian kegiatan pencarian dan pengkajian sumber-sumber yang relevan dan terpercaya yang menjadi acuan dalam penulisan tugas akhir ini. Literatur yang digunakan dititik-beratkan pada kinematika robot omni, *image processing*, dan kontrol pergerakan robot omni.

### 2. Pembuatan Desain Badan dan Elektronik Robot

Perencanaan yang di maksud bertujuan untuk membuat desain dasar robot dan serta catu daya untuk robot omni. Dasar pembuatan desain badan robot didasarkan pada beda sudut yaitu 120 derajat untuk robot



omni tiga roda. Untuk elektronik robot, desain didasarkan pada kebutuhan catu daya yang dibutuhkan robot yang meliputi tiga buah motor dc RS-545SH-2860, Raspberry Pi 3 Model B, tiga sensor gas TGS 2600, monitor 4.5 inch, tiga kipas dc 12v, MOSFET *Drivers* dan servo SG90.

### **3. Implementasi Desain dan Elektronik Robot**

Implementasi desain badan robot direalisasikan pada bahan *acrylic* bening untuk badan utama robot dan *acrylic* putih untuk penutup elektronik robot. Implementasi elektronik robot direalisasikan pada *pcb* (*printed circuit board*) yang meliputi regulator tegangan 12 volt ke 5 volt, MOSFET *Drivers*, GPIO Raspberry Pi 3 Model B, serta catu daya untuk sensor gas TGS 2600 dan kipas dc 12v.

### **4. Pengujian Deteksi Citra Obyek oleh Robot**

Pengujian deteksi citra obyek dilakukan untuk mengetahui parameter (kondisi) terbaik untuk sistem. Parameter terbaik didapatkan dari pengujian berdasarkan kondisi pencahayaan (Lux) yang berbeda dan pengujian berdasarkan kondisi lingkungan (*background*) yang berbeda.

### **5. Pengujian Kinematika Robot Omni**

Pergerakan robot omni ditentukan oleh kinematika robot omni baik secara *forward* maupun *invers*. Pegujian awal yang dilakukan meliputi pengujian respon kecepatan tiap-tiap motor dc untuk beberapa *duty cycle*. Pengujian kombinasi kecepatan tiap-tiap motor dc untuk beberapa sudut yang diujikan.

### **6. Tahap Pengujian Sistem Keseluruhan**

Pengujian ini dimaksudkan untuk memastikan bahwa kinerja masing-masing komponen dari hasil perealisasiian sistem dapat berfungsi sesuai dengan yang diharapkan. Pada tahap ini, sistem akan diuji pada kondisi pencahayaan (Lux) dan lingkungan (*background*) yang beragam.

### **7. Evaluasi**

Tahap evaluasi dan penyempurnaan kinerja sistem dilakukan setelah pengujian dilakukan sebelumnya. Pada tahap ini akan dinilai sistem kerja dari alat, baik dari metode, kondisi pencahayaan dan lingkungan yang paling sesuai.

### **8. Penyusunan Laporan Akhir**

Penyusunan laporan dilakukan setelah semua tahap terselesaikan sehingga hasil yang diperoleh dari pembuatan alat dapat dijelaskan secara rinci sesuai dengan data yang diperoleh.

### **9. Penulisan Makalah Jurnal POMITS**

Penulisan jurnal dilakukan sebagai sarana publikasi penelitian. Hal ini juga dilakukan agar penelitian mendapatkan legalitas dan pengakuan resmi dari dunia pendidikan.

## 1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi Lima Bab dengan sistematika penulisan sebagai berikut:

- Bab I : Pendahuluan  
Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.
- Bab II : Tinjauan Pustaka  
Bab ini menjelaskan tentang teori penunjang yang terkait maupun yang dibutuhkan dalam pengerjakaan tugas akhir ini. Dasar teori yang menunjang meliputi *omni-directional wheels robot*, sensor gas TGS 2600, *Normalized RGB, HSV (Hue, Saturation, and Value)*, Raspberry Pi 3 Model B, kamera Logitech C170, ADC ADS1115, *MOSFET Drivers*, motor DC, dan motor servo SG90,.
- Bab III : Perancangan Sistem  
Bab ini menjelaskan tentang perencanaan sistem yang meliputi diagram blok, perancangan mekanik dan elektronik pada robot.
- Bab IV : Pengujian dan Analisis  
Bab ini menjelaskan tentang hasil uji coba alat beserta analisisnya.
- Bab V : Penutup  
Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

## 1.7 Relevansi

*Omni-directional wheels robot* merupakan salah satu jenis *mobile robot* yang banyak dikembangkan. Di masa depan, besar kemungkinan penerapan *mobile robot* dengan tipe *omni-directional wheels* pada bidang yang lebih luas.

## BAB II TINJAUAN PUSTAKA

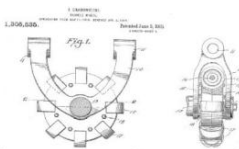
### 2.1 *Omni-Directional Wheels Robot*

*Omni-Directional Wheels Robot* adalah sebuah rancang bangun *holonomic mobile robot* yang menggunakan roda omni (*omni wheels*) sehingga bisa bergerak ke segala arah di sudut manapun tanpa memutar badan terlebih dahulu. Beberapa *omni-directional wheels robot* menggunakan platform segitiga dan persegi empat. Pada platform segitiga, robot omni menggunakan tiga buah roda omni yang terpisah sejauh  $120^\circ$  dan pada platform persegi empat, robot omni menggunakan empat buah roda omni. Perbedaan *holonomic* dan *non-holonomic mobile robot* sebagai berikut [1]:



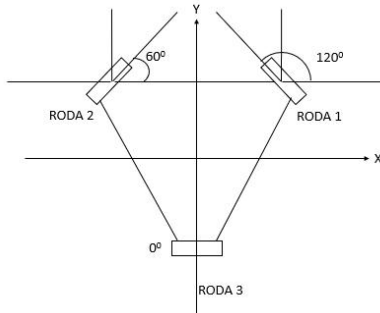
**Gambar 2.1** *Non-Holonomic Mobile Robot dan Holonomic Mobile Robot* [2]

Dua hal mendasar dalam perancangan robot omni yaitu roda omni dan kinematika gerak roda omni.



**Gambar 2.2** Gambaran Roda Omni [3]

Kinematika gerak roda omni didapatkan dengan menentukan komponen kecepatan dengan acuan sumbu  $x$  dan  $y$  robot. Untuk menentukan sumbu  $x$  dan  $y$  robot, maka ditentukan satu posisi roda sebagai acuan sumbu  $x$  dan  $y$  robot. Gambar 2.3 menjelaskan sumbu  $x$  dan  $y$  robot.



**Gambar 2.3** Robot omni dengan platform segitiga

Dari gambar 2.3, dapat diturunkan kecepatan tiap-tiap roda terhadap sumbu x dan y robot sehingga bisa didapatkan  $V_x$  dan  $V_y$  robot. Penurunan kecepatan tiap roda dapat dijelaskan sebagai berikut:

Untuk roda 1, didapatkan komponen kecepatan sebagai:

$$V_{x1} = V_1 \cos(120 + \alpha) \quad (2.1)$$

$$V_{y1} = V_1 \sin(120 + \alpha) \quad (2.2)$$

$$\omega_1 = R\dot{\theta} \quad (2.3)$$

$$V_1 = V_1 \cos(120 + \alpha) + V_1 \sin(120 + \alpha) + R\dot{\theta} \quad (2.4)$$

Untuk roda 2, didapatkan komponen kecepatan sebagai:

$$V_{x2} = V_2 \cos(60 + \alpha) \quad (2.5)$$

$$V_{y2} = V_2 \sin(60 + \alpha) \quad (2.6)$$

$$\omega_2 = R\dot{\theta} \quad (2.7)$$

$$V_2 = V_2 \cos(120 + \alpha) + V_2 \sin(120 + \alpha) + R\dot{\theta} \quad (2.8)$$

Untuk roda 3, didapatkan komponen kecepatan sebagai:

$$V_{x3} = V_3 \cos(0 + \alpha) \quad (2.9)$$

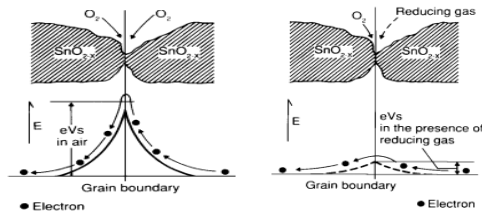
$$V_{y3} = V_3 \sin(0 + \alpha) \quad (2.10)$$

$$\omega_3 = R\dot{\theta} \quad (2.11)$$

$$V_3 = V_3 \cos(120 + \alpha) + V_3 \sin(120 + \alpha) + R\dot{\theta} \quad (2.12)$$

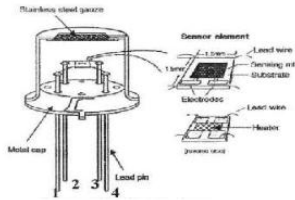
## 2.2 Sensor Gas Figaro TGS 2600

Sensor gas semikonduktor terdiri atas elemen sensor, dasar sensor, dan tudung sensor. Sebuah elemen sensor dibuat menggunakan timah (IV) oksida  $S_nO_2$ , wolfram (VI) oksida  $WO_3$ , dan lain-lain. Pada suatu keadaan dimana kristal oksida logam seperti  $S_nO_2$  dipanaskan pada suhu tinggi tertentu di udara, oksigen akan teradsorpsi pada permukaan kristal dengan muatan negative. Electron yang menjadi donor pada permukaan kristal, akan ditransfer ke oksigen teradsorpsi sehingga membentuk suatu lapisan ruang bermuatan positif. Hasil dari kondisi tersebut menyebabkan potensial permukaan terbentuk dan akan menghambat aliran electron. Proses yang terjadi di dalam sensor adalah arus listrik mengalir melalui bagian-bagian penghubung (batas butir) kristal-kristal mikro  $S_nO_2$ . Di batas-batas antar butir, oksigen yang teradsorpsi membentuk penghalang potensial yang menghambat muatan bebas bergerak sehingga terbentuk sebuah tahanan listrik sensor disebabkan oleh penghalang potensial. Ketika model penghalang potensial antar butir mendeteksi gas, kerapatan oksigen teradsorpsi bermuatan negative pada permukaan semikonduktor sensor menjadi berkurang, sehingga ketinggian penghalang pada batas antar butir berkurang. Sehingga menyebabkan berkurangnya tahanan butir dalam lingkungan gas. Ilustrasi dapat dilihat pada gambar 2.6.



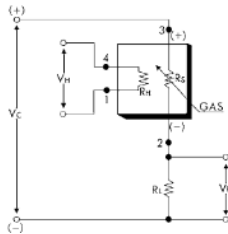
**Gambar 2.4** Model level penghalang saat tidak ada gas yang terdeteksi (kiri) dan saat ada gas yang terdeteksi (kanan) [4].

Sensor TGS 2600 memiliki struktur yang dijelaskan pada gambar 2.5. Gas yang terdeteksi adalah gas yang masuk melalui pori-pori kecil pada sensor TGS 2600. Gas terdeteksi ditandai dengan konduktivitas sensor yang meningkat. Untuk mengetahui perubahan tersebut, maka dibutuhkan suatu rangkaian dasar pengukuran untuk sensor gas TGS 2600.



**Gambar 2.5** Struktur sensor TGS [2]

Sensor TGS memiliki dua bagian utama. Bagian pertama yaitu tin oxide ( $S_nO_2$ ) sebagai sensor material. Material ini terhubung pada pin 2 dan 3 pada sensor. Bagian kedua yaitu heater untuk memanaskan material. Heater ini terhubung dengan pin 1 dan 4. Gambar menunjukkan struktur dari sensor gas TGS secara umum. TGS 2600 berfungsi sebagai *Alcohol Tester* untuk mendeteksi besarnya kontaminasi dalam udara. Rangkaian pengukuran dasar untuk sensor dijelaskan pada gambar 2.6.



**Gambar 2.6** Rangkaian Pengukuran Dasar Sensor Gas [4]

Perubahan konduktivitas sensor akibat adanya gas yang terdeteksi dapat diketahui dengan mengukur perubahan tegangan pada resistor yang terpasang seri dengan material sensor.

### 2.3 Kamera

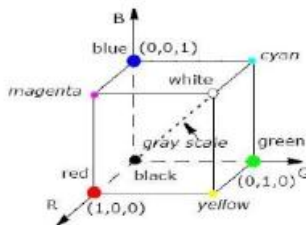
Kamera berfungsi sebagai penangkap citra atau sensor vision. Sensor vision merupakan sensor yang kompleks digunakan dalam robotika. Tahapan pengolahan sensor vision ini dinamakan pengolahan citra digital. Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang ditangkap oleh kamera yang dikuantisasi dalam bentuk diskrit disebut sebagai citra digital (*digital image*), dimana citra digital

ini memiliki 2 jenis yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra [5].

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.13)$$

## 2.4 Normalisasi RGB

Sebuah *Color Space* adalah sebuah metode yang bisa kita spesifikasikan dan visualkan warnanya ke dalam representasi data. *Color space* adalah penjabaran warna menjadi komponen-komponen terpisah yang membangun warna itu sendiri. Ruang warna RGB menjabarkan suatu warna tertentu menjadi kombinasi antara *Red*, *Green*, dan *Blue*, sedangkan ruang warna HSV menyatakan warna tertentu kedalam *Hue*, *Saturation*, dan *Value*. Semua representasi warna tersebut digunakan pada kondisi tertentu tergantung kebutuhan. Pada umumnya, ruang warna direpresentasikan sebagai data representasi 3-D seperti gambar 8.



**Gambar 2.7** Ruang Warna RGB [6]

Seperti digambarkan diatas, pada ruang warna RGB, warna red, green, dan blue dipetakan pada sistem koordinat kartesian 3-D yang membentuk kubus 3-D. Titik-titik kubus adalah warna primer (red, green, dan blue) dan warna sekunder (cyan, yellow, dan magenta). Origin dari koordinat adalah black dimana komponen warna red, green, dan blue semuanya bernilai 0. Pojok kebalikan kubus secara diagonal

adalah white, dimana nilai komponen RGB bernilai maksimum.

Cahaya yang sampai pada mata manusia adalah sebuah fungsi dari pantulan permukaan, warna cerah, dan *lighting geometry*. Warna warna yang kita terima bergantung pada hampir sebagian besar pada pantulan permukaan. Ketergantungan karena *lighting geometry* dan warna cerah dihilangkan melalui sebuah langkah normalisasi citra [7]. Normalisasi RGB dilakukan memisahkan komponen *achromatic* dan *chromatic* dari citra lalu membagi masing-masing dengan jumlah totalnya sesuai formula berikut

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} / (R + G + B) \quad (2.14)$$

Keterangan:

r, g, b = nilai setelah normalisasi RGB

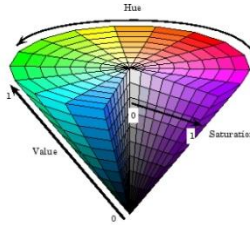
R, G, B = nilai sebelum normalisasi RGB

## 2.5 Ruang Warna *HSV (Hue, Saturation, and Value)*

Warna adalah hasil visual dari spektrum cahaya baik itu ditransmisikan lewat media transparan, diserap, dan dipantulkan oleh sebuah permukaan. Warna adalah panjang gelombang cahaya yang dapat diterima dan diproses oleh mata manusia dengan nilai antara 400nm sampai dengan 700nm. Ruang warna (*Color Space*) model matematis abstrak yang menggambarkan agar suatu warna dapat direpresentasikan sebagai baris angka yang biasanya terdiri dari nilai-nilai tiga atau empat buah warna seperti RGB, CMYK, dan HSV.

Ruang warna HSV terdiri dari tiga bagian yaitu *Hue*, *Saturation (Chroma)*, dan *Value*. *Hue* adalah istilah yang digunakan untuk menyatakan warna dasar sebenarnya seperti biru, hijau, dan merah. *Hue* dinyatakan dalam derajat (*degree*) bervariasi antara 0 sampai 360. *Saturation* adalah nilai warna putih ke warna murni. *Value* menunjukkan kecerahan dari warna. *Saturation* dan *Value* bervariasi antara 0 sampai 1. Berikut adalah ilustrasi dan formula dari HSV.





**Gambar 2.8** Penjelasan HSV [8]

Warna didefinisikan oleh (R, G, B) pada ruang warna RGB. Untuk menentukan nilai HSV dari RGB, digunakan formula berikut :

$$H = \begin{cases} \frac{G-B}{Max-Min} \times 60, & \text{if } R = Max \\ 120 + \frac{B-R}{Max-Min} \times 60, & \text{if } G = Max \\ 240 + \frac{R-G}{Max-Min} \times 60, & \text{if } B = Max \end{cases} \quad (2.15)$$

$$S = \frac{Max-Min}{Max}; V = Max \quad (2.16)$$

Max merupakan nilai maksimum dari RGB. Min merupakan nilai minimum dari RGB.

## 2.6 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B adalah sebuah *SBC* (*Single Board Computer*) berukuran kartu kredit (85 x 56 x 17)mm yang dikembangkan oleh Raspberry Pi Foundation di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer di sekolah-sekolah. Raspberry Pi 3 Model B resmi diluncurkan pada 29 Februari 2016. Model terbaru ini dilengkapi dengan Wi-Fi 802.11n dan Bluetooth versi 4.1 sehingga memudahkan pengguna dalam mengakses jaringan internet tanpa perlu menambahkan modul atau menggunakan *slot* RJ-45. Raspberry Pi 3 menggunakan ARM Cortex-A53 dari Broadcom dengan spesifikasi 64-bit Quad-Core dan berkecepatan 1,2 GHz. Untuk RAM, model ini masih menggunakan RAM 1 GB. Dari segi desain dan struktur *slot*, model ini tidak terlalu berubah. Raspberry Pi dilengkapi empat *slot* USB, sebuah *slot* RJ-45, dan GPIO 40 Pin. Selain itu, model ini juga dilengkapi dengan audio output 3.5 mm jack, video output

HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)), 15-pin Camera Serial Interface, Display Serial Interface, dan Memory Card Slot.



**Gambar 2.9** Board Raspberry Pi 3 Model B

## 2.7 ADC ADS1115



**Gambar 2.10** ADC ADS1115

Gambar 2.10 adalah salah satu contoh bentuk fisik ADC ADS1115 yang ada di pasaran. ADS1115 merupakan *analog-to-digital converters* (ADCs) presisi dengan resolusi 16 bit yang ditawarkan dalam kemasan *ultra-small, leadless*, atau kemasan MSOP-10. ADS1115 didesain dengan pertimbangan presisi, daya, dan mudahnya implementasi. ADS1115 memiliki fitur *oscillator* dan referensi *onboard*. Data dikirim melalui *I<sup>2</sup>C-compatible serial interface*; empat *I<sup>2</sup>C slave address* bisa dipilih. ADS1115 bekerja dari satu daya tunggal yang memiliki *range* dari 2.0v sampai 5.5v.

ADS1115 mampu melakukan konversi hingga kecepatan 860 *sample per second* (SPS). PGA *onboard* tersedia pada ADS1115 yang menyediakan *input range* dari satu daya hingga serendah  $\pm 256\text{mV}$  sehingga memungkinkan baik sinyal kecil maupun besar terukur dengan resolusi tinggi. ADS1115 juga memiliki fitur *input multiplexer* (MUX) yang menyediakan dua *differential* atau empat *single-ended inputs*.

## 2.8 Motor DC

Motor DC memiliki dua bagian dasar yaitu rotor dan stator. Stator atau bagian stationer adalah bagian yang diam pada motor. Stator berfungsi menghasilkan medan magnet yang dibangkitkan dari sebuah koil (elektromagnet) atau magnet permanen. Bagian stator terdiri dari badan motor yang memiliki magnet pada keseluruhan permukaan badan. Untuk motor berukuran kecil, magnet tersebut adalah magnet permanen sedangkan untuk motor berukuran besar, maka menggunakan koil atau kumparan yang dililitkan pada lempeng-lempeng besi. Rotor adalah bagian berputar pada motor DC. Rotor berupa koil yang dialiri oleh arus listrik. Kumparan motor akan berfungsi jika memiliki:

- a. Kumparan jangkar berfungsi sebagai pengimbas GGL pada konduktor yang terletak pada alur-alur jangkar.
- b. Kumparan medan berfungsi sebagai penghasil medan magnet.
- c. Celah udara yang memungkinkan berputarnya jangkar dalam medan magnet.

Motor DC dibagi menjadi dua jenis jika ditinjau dari cara pemberian daya ke rotor yaitu motor DC dengan sikat (*brushed*) dan motor DC tanpa sikat (*brushless*). Sikat (*brush*) pada motor DC *brushed* berguna untuk memberi suplai arus kepada *rotating armature* atau rotor melalui komutator, sementara motor DC *brushless* tidak perlu sikat karena menggunakan rotor dengan magnet permanen. Pada motor DC *brushed*, medan magnet dihasilkan dengan melewatkan arus melalui komutator dan sikat yang terletak di dalam rotor. Sikat terbuat dari bahan karbon. Motor DC RS-545SH 2860 adalah salah satu contoh motor DC *brushed* yang ada di pasaran.

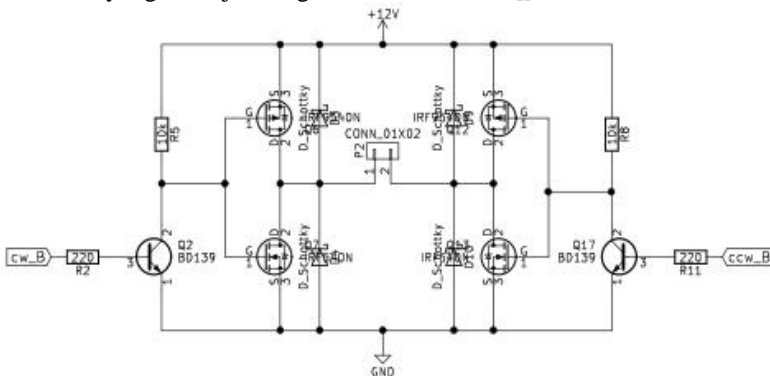


**Gambar 2.11** Motor DC dengan sikat karbon

Pada penelitian ini, motor DC RS-545SH 2860 digunakan sebagai penggerak utama robot omni dengan platform segitiga. Pergerakan robot omni memanfaatkan kombinasi kecepatan dari tiga motor DC RS-545SH 2860.

## 2.9 H-Bridge MOSFET Drivers

*H-bridge MOSFET drivers* merupakan sebuah perangkat yang sering digunakan untuk men-*drive* beban induktif pada banyak aplikasi kontrol. Khususnya pada motor DC, *H-bridge driver* dapat digunakan untuk mengatur arah dan kecepatan motor DC[]. Maka dari itu, *driver* ini menjadi komponen *driving* penting untuk kontrol motor. Struktur dasar dari *H-bridge* dibangun oleh 4 *power transistor* seperti BJT dan MOSFET yang bekerja sebagai saklar elektronik[.]



**Gambar 2.12** Rangkaian H-Bridge MOSFET Drivers

Rangkaian *driver* dijelaskan pada gambar 2.11. Transistor BJT memiliki fungsi saklar elektronik yang dikontrol oleh sinyal dari pin mikrokontroler. Saat pin cw “1”, maka BJT Q2 akan aktif sehingga P-channel MOSFET kiri atas dan N-Channel MOSFET kanan bawah akan aktif sehingga arus dapat melalui jalur tersebut untuk menggerakkan motor dengan arah searah jarum jam. Saat pin ccw “1”, maka BJT Q17 akan aktif sehingga P-channel MOSFET kanan atas dan N-Channel MOSFET kiri bawah akan aktif sehingga arus dapat melalui jalur tersebut untuk menggerakkan motor dengan arah berlawanan arah jarum jam.

## 2.10 Kontroler Proporsional

Motor DC digunakan secara ekstensive pada bidang robotika dan peralatan listrik di industry. Maka dari itu, pengaturan kecepatan motor DC sangat penting dan telah dipelajari sejak beberapa awal decade century terakhir. Pada umumnya, sistem motor DC memiliki

karakteristik tidak tentu dan nonlinear sehingga membutuhkan controller [9]. Salah satu tipe controller yaitu controller proportional. Controller proportional memiliki rumus sebagai berikut

$$u(t) = K_p \cdot e(t) \quad (2.17)$$

Keterangan :

$u(t)$  = sinyal kontrol

$K_p$  = konstanta proportional

$e(t)$  = sinyal error

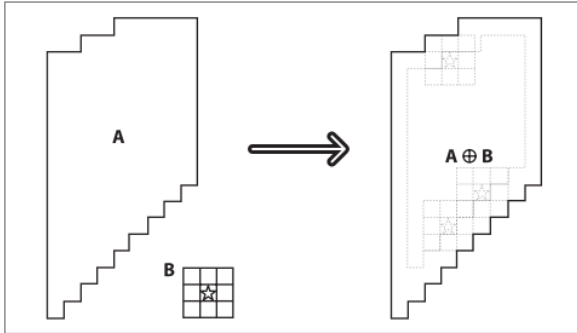
## 2.11 OpenCV

OpenCV (*Open source Computer Vision*) merupakan sebuah pustaka (*library*) fungsi programming untuk pengolahan citra. Pustaka OpenCV dapat digunakan dengan beberapa bahasa pemrograman seperti C++ dan Python untuk melakukan pengambilan, pengolahan, serta menampilkan citra baik berupa citra diam (*still image*) maupun citra bergerak (*moving image*). Pustaka OpenCV bersifat *open source* sehingga memungkinkan penggunaan baik untuk keperluan akademik maupun komersil. Pustaka OpenCV dapat berjalan pada beberapa *Operating System* seperti Windows, Linux, Android, dan MacOS.

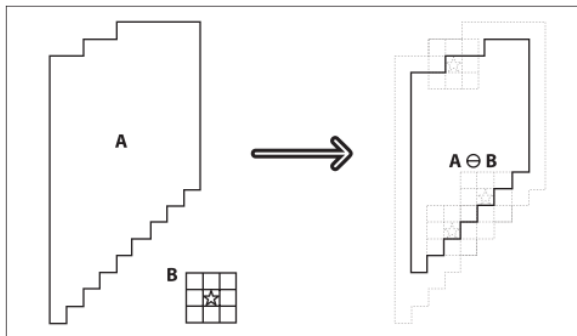
## 2.12 Image Morphology

Dilasi adalah sebuah konvolusi citra A dengan suatu kernel B. Kernel dapat berbentuk apapun dan memiliki *anchor point* tunggal yang ditentukan. Pada umumnya, kernel berupa kotak solid kecil atau lonjong dengan *anchor point* di tengahnya. Kernel bisa dianggap sebagai template atau mask dan efek kernel untuk dilasi adalah *local maximum operator*. Ketika kernel B di-*scan* terhadap suatu citra, kita menghitung nilai piksel maksimal yang dilewati oleh kernel B dan mengganti piksel citra di bawah *anchor point* dengan nilai maksimal tersebut. Ini menyebabkan bagian putih pada citra bertambah seperti pada gambar 2.13. Pertambahan ini adalah asal muasal istilah operator dilasi [10].

Erosi adalah operasi kebalikan dilasi. Aksi operator erosi adalah menghitung *local minimum* dari kernel. Erosi menghasilkan citra baru dari citra asli dengan algoritma: saat kernel B di-*scan* terhadap citra, kita menghitung nilai piksel minimal yang dilewati kernel B dan mengganti piksel citra di bawah *anchor point* dengan nilai minimal tersebut [10]. Proses erosi dijelaskan pada gambar 2.14.



**Gambar 2.13** Dilasi : mengambil nilai maksimum di bawah kernel B

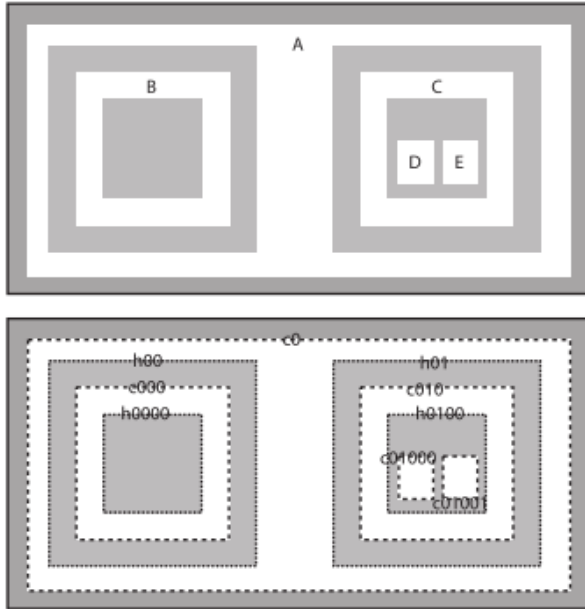


**Gambar 2.14** Erosi : mengambil nilai minimum di bawah kernel B

Secara umum, dilasi menambah region dan erosi mengurangi region. Dilasi akan cenderung menghaluskan cekungan, sementara erosi menghaluskan tonjolan. Tentunya hasil akhir bergantung pada tipe kernel yang digunakan [10].

### 2.13 Find Contours

Sebuah kontur adalah sebuah daftar titik-titik yang merepresentasikan sebuah kurva dalam suatu citra. Kontur direpresentasikan oleh OpenCv berupa *sequences* yang tiap *sequence* berisi informasi lokasi titik selanjutnya pada kurva. Fungsi `cvFindContours()` menghitung kontur-kontur dari citra biner yang bisa berasal dari `cvCanny()` yang memiliki piksel-piksel *edge* atau citra yang dihasilkan dari fungsi `cvThreshold()` atau `cvAdaptiveThreshold()` dimana *edge* adalah batas antara daerah positif dan negative [10].



**Gambar 2.15** (Atas) Citra tes untuk dilakukan `cvFindContours`: (bawah) kontur berjenis kontur (garis putus) juga *hole* (garis titik).

Gambar bagian atas menunjukkan bagian citra dengan daerah putih dengan label A hingga E pada background gelap. Gambar bagian bawah menunjukkan gambar yang sama dengan kontur-kontur yang akan dihitung oleh `cvFindContours()`. Kontur-kontur diberi label `cX` dan `hX`, dimana `c` untuk kontur, `h` untuk *hole*, dan `X` adalah angka. Beberapa kontur berupa garis putus-putus yang merepresentasikan batas luar daerah putih.

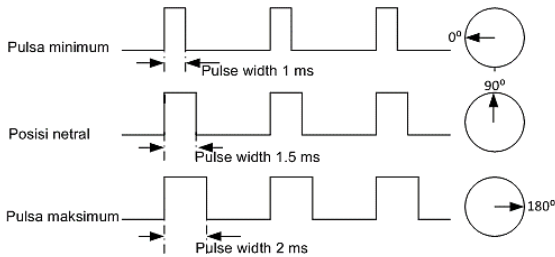
## 2.14 Motor Servo DC

Motor servo merupakan jenis motor DC, yang membedakan motor servo dengan motor DC biasa adalah motor servo dapat diatur perputaran sudutnya. Ada 2 tipe motor servo, yaitu motor servo standart dan motor servo continous. Motor servo standart dapat berputar  $180^\circ$  sedangkan motor servo continous adalah motor servo dengan putaran sampai  $360^\circ$ . Motor servo mempunyai 3 pin *input*, yaitu `Vcc`, `Gnd`, dan

Data. Vcc dan Gnd merupakan sumber tegangan untuk motor servo, sedangkan data merupakan data dari mikrokontroler yang berupa Pulse Width Modulation (PWM). PWM merupakan teknik modulasi yang merubah panjang sinyal high atau low dalam frekuensi yang tetap. Bentuk dari motor servo dapat dilihat pada Gambar 2.16.



**Gambar 2.16** Servo SG-90



**Gambar 2.17** Lebar pulsa sinyal servo

Motor servo memiliki panjang pulsa 20mS, untuk mendapatkan sudut  $0^\circ$  maka pulsa high dapat diset selama 1mS, dan untuk mendapat sudut  $180^\circ$  maka lama waktu high diatur selama 2 mS dan waktu low adalah sisa periode 20 mS tersebut.



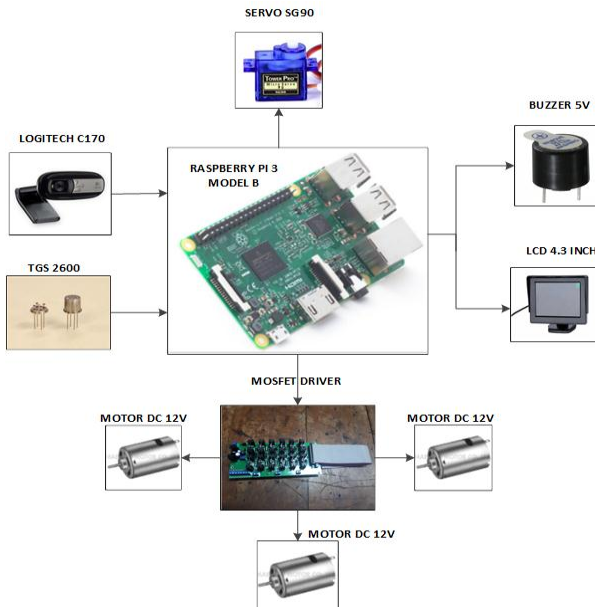
## **BAB III**

### **PERANCANGAN SISTEM**

Perancangan sistem *omni-directional wheels robot* terdiri dari beberapa bagian yang meliputi diagram blok sistem, perancangan mekanik, perancangan elektronik, dan perancangan perangkat lunak. Perancangan mekanik meliputi pembuatan badan dasar robot dari bahan *acrylic* serta aluminium. Perancangan elektronik meliputi pembuatan rangkaian catu daya robot, MOSFET *drivers*, GPIO Raspberry Pi 3 Model B, serta rangkaian pengukuran dasar sensor gas TGS 2600. Perancangan perangkat lunak meliputi pembuatan program untuk deteksi obyek, pembacaan data sensor gas TGS 2600, serta program untuk kontrol pergerakan robot menuju obyek.

#### **3.1 Diagram Blok Sistem**

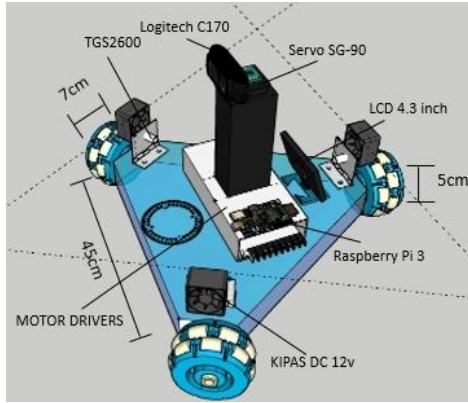
Sistem robot omni dapat digambarkan pada sebuah diagram blok. Diagram blok sistem membantu menjelaskan hubungan satu sama lain antar bagian-bagian penyusun robot omni. Kamera Logitech C170 berfungsi untuk menangkap citra di sekitar robot yang selanjutnya diproses oleh Raspberry Pi 3 Model B untuk mendapatkan citra obyek yang dimaksud. Raspberry Pi 3 model B+ mengirim perintah ke servo SG90 untuk berputar secara *clockwise* dan *counterclockwise*. Dengan bantuan servo SG90, kamera akan mampu menangkap citra secara 180 derajat di sekitar robot. Servo SG90 akan berhenti saat obyek terdeteksi. Pergerakan robot menuju obyek yang terdeteksi memanfaatkan kombinasi kecepatan dari tiga buah motor dc 12v. Robot membutuhkan *driver* motor untuk menggerakkan motor. Rangkaian *driver* motor yang digunakan yaitu MOSFET *driver* dengan tiga *H-Bridge*. Saat robot berhenti di dekat obyek yang terdeteksi, pembacaan sensor gas TGS 2600 akan berfungsi untuk mendeteksi gas alkohol sekitar obyek. Buzzer akan berbunyi jika ada gas alkohol yang terdeteksi oleh sensor gas TGS 2600. Monitor LCD 4.5 inch berfungsi untuk menampilkan *videocapture* dari webcam Logitech C170. Hubungan antar bagian-bagian penyusun robot dapat dijelaskan dengan diagram blok pada gambar 3.1.



**Gambar 3.1** Diagram Blok Sistem

### 3.2 Perancangan Mekanik

Perancangan mekanik utama yang dibuat pada penelitian ini merupakan badan robot dengan ukuran 45 cm x 7 cm x 5 cm. Badan dasar robot diwujudkan dari bahan *acrylic* dan aluminium. Perancangan desain mekanik didasarkan pada perbedaan sudut antar tiga roda omni sebesar 120 derajat. Perbedaan sudut antar masing-masing roda berguna untuk kinematika robot sehingga menjadi dasar pertimbangan dalam desain robot omni dengan tiga roda. Selain dasar perbedaan sudut tiap roda, badan robot juga harus mampu menampung bagian-bagian penyusun robot lainnya seperti catu daya, MOSFET *drivers*, Raspberry Pi 3 model B+, LCD 4.5 inch, kamera Logitech C170, rangkaian pengukuran dasar sensor gas TGS 2600 serta rangkaian *buzzer* (alarm).



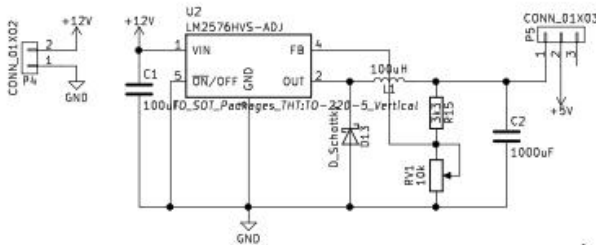
**Gambar 3.2** Desain Robot Omni

### 3.3 Perancangan Elektronik

Perangkat elektronik robot meliputi catu daya, *MOSFET drivers*, penentuan pin GPIO Raspberry Pi 3 model B+, rangkaian pengukuran dasar TGS 2600 dan rangkaian *buzzer* (alarm).

#### 3.3.1 Catu Daya

Daya yang digunakan oleh robot berasal dari baterai LiPo 11.1v 2200mAH. Catu daya berfungsi untuk menyediakan tegangan yang dibutuhkan bagian-bagian robot untuk bekerja. Secara garis besar, bagian-bagian robot membutuhkan tegangan 12v dan 5v. Bagian robot yang membutuhkan tegangan 12v yaitu *MOSFET drivers*, tiga motor dc RS-545SH-2680, tiga kipas dc 12v, dan LCD 4.5 inch. Bagian robot yang membutuhkan tegangan 5v yaitu Raspberry Pi 3 model B+, tiga sensor gas TGS 2600, buzzer 5v, dan ADS 1115.



**Gambar 3.3** Rangkaian Catu Daya

IC Regulator LM2576-5V digunakan untuk menurunkan tegangan 12v dari baterai LiPo menjadi 5v. Pin input regulator harus di-*bypass* dengan kapasitor *electrolytic* 100uF untuk menjaga kestabilan. Kaki kapasitor harus berukuran pendek dan ditempatkan di dekat regulator. Kapasitor output dibutuhkan untuk mem-*filter* tegangan output dan dibutuhkan untuk kestabilan *loop*. Nilai kapasitor output yang digunakan yaitu 1000uF. Dioda yang digunakan yaitu 1N5822 karena dioda ini memiliki *rating* maksimal  $V_{RRM}$  40V.

### 3.3.2 H-Bridge MOSFET Drivers

Motor yang digunakan untuk menggerakkan robot adalah motor DC 12v. Untuk mengatur arah dan kecepatan motor DC 12v ini, Raspberry Pi 3 model B sebagai *processing unit* memerlukan sebuah *driver* motor. Rangkaian *driver* motor yang digunakan pada tugas akhir ini yaitu *H-Bridge MOSFET driver*. Dengan memanfaatkan BJT sebagai switch, arah gerakan motor (*clockwise* atau *counter-clockwise*) dapat diatur. Kecepatan motor dapat diatur dengan memanfaatkan PWM yang dihasilkan oleh Raspberry Pi. Jenis FET yang digunakan pada rangkaian *driver* juga harus memiliki *rating* yang sesuai dengan kebutuhan motor DC 12v. Jenis MOSFET yang digunakan yaitu IRF540 dan IRF9540.

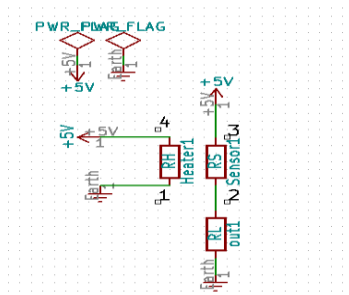
Terdapat tiga rangkaian *driver* motor seperti pada gambar 2.15 pada penelitian ini. Rangkaian *driver* dengan pin *cw\_A* dan *ccw\_A* untuk motor 1, pin *cw\_B* dan *ccw\_B* untuk motor 2, serta *cw\_C* dan *ccw\_C* untuk motor 3. Arah gerakan motor dapat ditentukan dengan mengatur *logic input* dari pin *cw* dan *ccw* sehingga didapatkan tabel 3.1 untuk gerakan tiap motor pada robot omni.

**Tabel 3.1** Tabel Kebenaran Gerakan Motor

Input		Output	Keterangan
Pin cw	Pin ccw	Motor	
1	0	Vmot	Bergerak <i>clockwise</i>
0	1	-Vmot	Bergerak <i>counter-clockwise</i>
0	0	Gnd	Stop
1	1	X	<i>Don't Care</i>

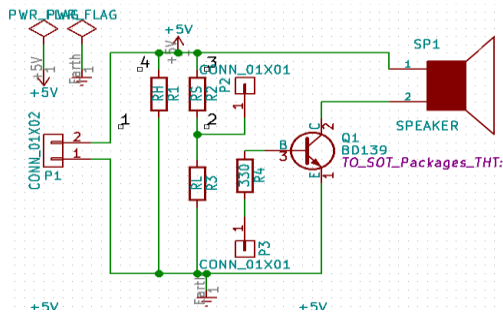
### 3.3.3 Rangkaian Pengukuran Dasar TGS 2600 dan Rangkaian Buzzer

Sensor gas TGS 2600 membutuhkan dua input tegangan yaitu tegangan *heater* (Vh) dan tegangan *circuit* (Vc). Tegangan *heater* digunakan untuk *integrated heater* untuk menjaga *sensing element* dari sensor pada temperature tertentu yang optimal untuk *sensing*. Tegangan *circuit* (Vc) digunakan agar dapat dilakukan pengukuran tegangan pada resistor beban (RL) yang disusun seri dengan sensor.



Gambar 3.4 Rangkaian Pengukuran Dasar TGS 2600

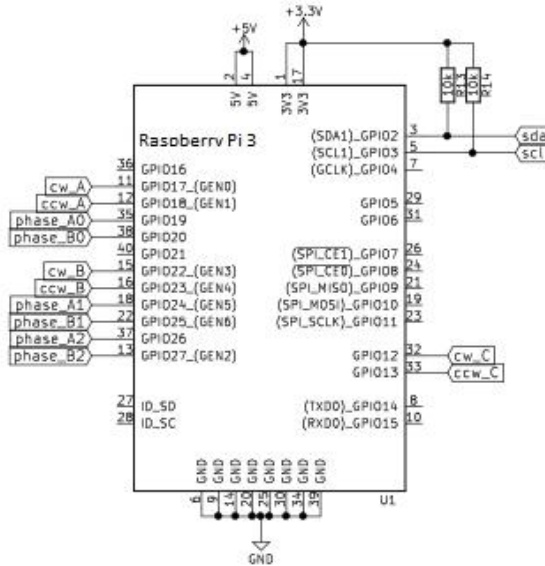
Rangkaian pengukuran dasar TGS 2600 berjumlah sebanyak tiga untuk masing-masing sensor gas di tiap tempat berdekatan dengan roda. Rangkaian buzzer (alarm) diperlukan robot untuk memberi tanda berupa bunyi jika gas alkohol terdeteksi.



Gambar 3.5 Rangkaian Buzzer

### 3.3.4 GPIO Raspberry Pi 3 Model B+

Raspberry Pi yang digunakan di penelitian ini mendapat daya melalui *GPIO*(*General Purpose Input Output* ). Raspberry Pi membutuhkan tegangan 5v dan arus 2A untuk bekerja. Tegangan 5v untuk Raspberry Pi berasal dari output regulator LM2576.



Gambar 3.6 GPIO Raspberry Pi 3

Pin dari Raspberry Pi 3 model B yang digunakan, dijelaskan pada tabel 3.2.

Tabel 3.2 Penggunaan pin Raspberry Pi 3 untuk penelitian ini

Pin Raspberry Pi	Keterangan
GPIO12	Pin untuk motor 1 arah cw
GPIO13	Pin untuk motor 1 arah ccw
GPIO17	Pin untuk motor 3 arah cw
GPIO18	Pin untuk motor 3 arah ccw
GPIO22	Pin untuk motor 2 arah cw
GPIO23	Pin untuk motor 2 arah ccw
GPIO24	Pin untuk <i>buzzer</i>
GPIO26	Pin untuk sinyal servo SG-90

### 3.3.5 Kamera Logitech C170

Kamera yang digunakan untuk menangkap citra pada penelitian ini yaitu kamera Logitech C170.

**Tabel 3.3** Spesifikasi kamera Logitech C170

Spesifikasi	Keterangan
Diagonal Field of View	58 <sup>0</sup>
Focal Length	2.3mm
Video Capture (4:3 SD)	320x240,640x480,1024x768
Frame rate(30)	640x480@30
Weight	75g

### 3.3.6 Servo SG-90

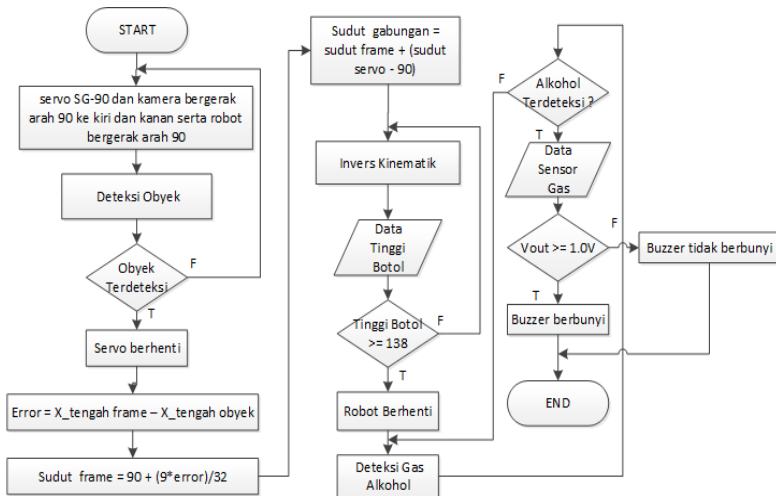
Untuk membantu robot proses deteksi botol hijau, kamera digerakkan dengan servo SG-90. Servo ini mampu bergerak 90 ke kanan dan ke kiri total 180 derajat.

**Tabel 3.4** Spesifikasi servo SG-90

Spesifikasi	Keterangan
Operating Voltage	5V
Rotation	0 -180 degree
Torque	2.5kg/cm
Gear type	Plastic

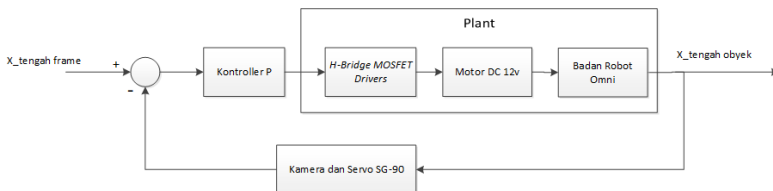
## 3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk pembuatan program untuk deteksi obyek, pembacaan data sensor gas, dan kontrol pergerakan robot menuju obyek. Pembuatan program untuk deteksi obyek memanfaatkan *library OpenCV*. Pustaka *OpenCV* digunakan dalam proses-proses pengolahan citra yang ditangkap kamera hingga didapatkan informasi dari citra obyek yang dimaksud. Informasi ini berupa jumlah *blob* (gumpalan pixel) untuk penentuan obyek terdeteksi atau tidak, posisi obyek tepat di tengah *frame* untuk pengereman servo sehingga bisa didapatkan data sudut, dan tinggi botol (pixel) untuk penentuan posisi berhenti robot dari obyek.



**Gambar 3.4** Diagram Alir (flowchart ) Sistem

Pergerakan robot menuju obyek diatur oleh sebuah kontroller. Tipe kontroller yang digunakan untuk mengatur pergerakan robot yaitu proportional. Kontrol pergerakan robot menuju obyek dapat digambarkan pada diagram blok sistem pengaturan pada gambar 3.9. Nilai Kp yang digunakan sama dengan 1.

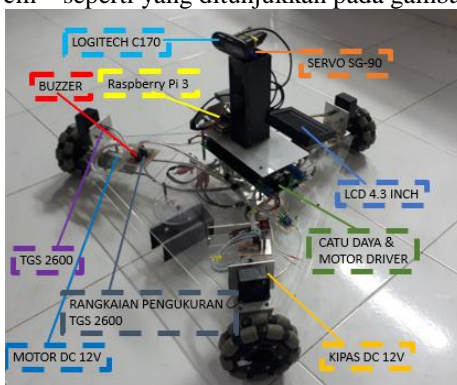


**Gambar 3.5** Diagram Blok Sistem Pengaturan



## BAB IV PENGUJIAN DAN ANALISIS

Bab ini membahas tentang pengujian dari sistem yang telah dirancang beserta analisis data yang diperoleh saat pengujian. Pengujian-pengujian yang dilakukan meliputi pengujian deteksi obyek oleh robot, pengujian sensor gas Figaro TGS2600, dan pengujian arah gerakan robot. Berdasarkan desain pada bab tiga, dilakukan implementasi desain robot *omni-directional wheels* dengan ukuran 45 cm x 7 cm x 5 cm seperti yang ditunjukkan pada gambar 4.1.



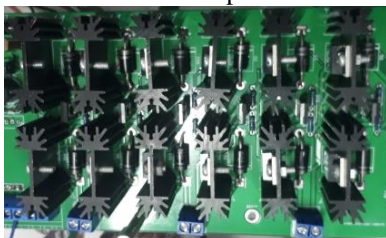
**Gambar 4.1** Realisasi desain robot *omni-directional wheels*.

Pada gambar 4.1 dapat ditinjau realisasi sistem secara keseluruhan robot *omni-directional wheels*. Robot telah dilengkapi dengan kamera Logitech C170, servo SG90, LCD 4.5inch, baterai LiPo Turnigy 11.1v 2200mAH, rangkaian catu daya, rangkaian *H-Bridge MOSFET-Driver*, Raspberry Pi 3 Model B+, tiga motor DC RS-545SH-2860 dengan roda omni masing-masing, tiga kipas dc 12v serta sensor gas TGS2600.



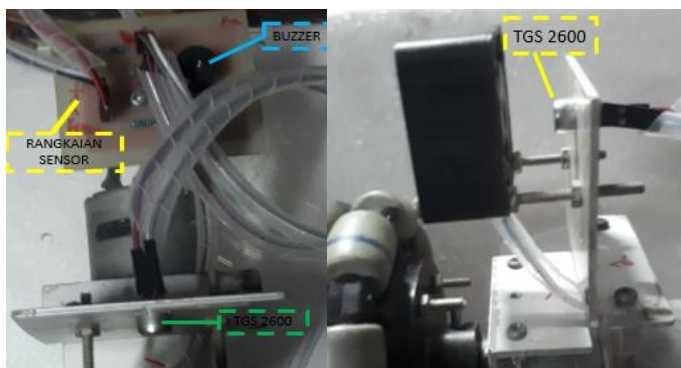
**Gambar 4.2** Realisasi rangkaian catu daya

Pada tugas akhir ini, terdapat tiga rangkaian *H-Bridge MOSFET Drivers* masing-masing untuk tiap motor DC. Tiap motor DC disambung melalui terminal screw dua pin.



**Gambar 4.3** Realisasi rangkaian *H-Bridge MOSFET-Driver*

Sensor gas TGS 2600 membutuhkan rangkaian pengukuran dasar untuk dapat mengetahui perubahan tegangan saat gas alkohol ada. Tegangan yang diukur yaitu tegangan pada  $R_L$ . Robot akan menyalakan *buzzer* jika gas alkohol terdeteksi.



**Gambar 4.4** Realisasi rangkaian pengukuran dasar TGS 2600

Pengujian kontrol pergerakan robot *omni-directional wheels* dilakukan di laboratorium elektronika industri B402 Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

- Pengujian dibagi menjadi yaitu:
1. Pengujian Deteksi Citra Obyek.

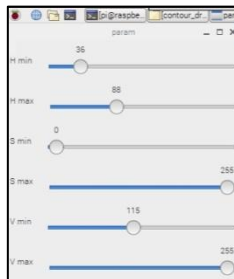
2. Pengujian Sensor Gas TGS 2600.
3. Pengujian Gerakan Robot

#### 4.1 Pengujian Deteksi Citra Obyek

Pengujian deteksi citra obyek merupakan pengujian yang penting dalam penelitian ini. Pengujian deteksi citra obyek meliputi kalibrasi manual, pengolahan citra obyek, pengujian pada intensitas cahaya berbeda, pengujian pada background hijau, dan pengukuran tinggi botol.

##### 4.1.1 Kalibrasi Manual

Kalibrasi manual perlu dilakukan sebelum proses utama pengolahan citra obyek. Kalibrasi manual bertujuan untuk mendapatkan nilai atas (*upper*) dan bawah (*lower*) yang akan digunakan pada proses utama pengolahan citra obyek sebagai batasan *threshold* sehingga didapatkan citra obyek sesuai yang dikehendaki. Proses kalibrasi manual dapat dilakukan dengan memanfaatkan *trackbar* dari *library opencv* sehingga nilai *Hue*, *Saturation*, dan *Value* dapat digeser sesuai keinginan. Berikut adalah *trackbar* untuk penentuan nilai atas dan nilai bawah dari citra obyek.



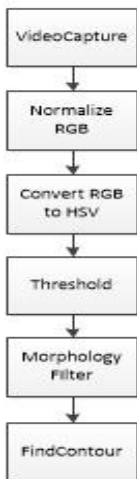
**Gambar 4.5** Kalibrasi dengan *trackbar*

Gambar 4.5 merupakan *window* yang muncul untuk menentukan nilai parameter *Hue*, *Saturation*, dan *Value*. Jendela ini memiliki enam *trackbar* yang digunakan untuk menentukan tiga nilai bawah HSV dan tiga nilai atas HSV.

##### 4.1.2 Pengolahan Citra Obyek

Citra yang ditangkap kamera Logitech C170 akan melalui beberapa proses pengolahan citra sehingga didapatkan citra obyek yang dikehendaki. Citra obyek hasil beberapa proses juga berguna untuk penentuan jarak dimana robot berhenti dari posisi obyek. Berikut adalah

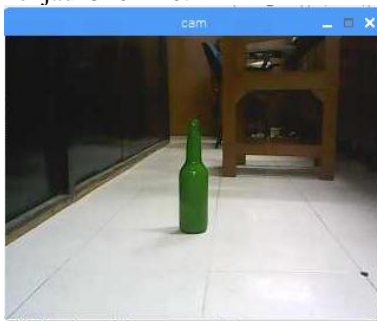
proses-proses yang dilakukan.



**Gambar 4.6** Urutan Pengolahan Citra

#### 4.1.2.1 Video Capture

Langkah pertama yang dilakukan yaitu kamera mengambil citra di area sekitar robot. Citra yang diambil oleh kamera Logitech C170 berukuran 640x480. Untuk mempermudah proses pengolahan, maka ukuran citra diperkecil menjadi 320x240.



**Gambar 4.7** Citra awal

#### 4.1.2.2 Normalisasi RGB

Bagian gelap dan bagian terang akan muncul pada citra obyek saat obyek terkena cahaya. Untuk mengurangi bagian gelap dan bagian terang dari citra obyek, maka perlu dilakukan normalisasi RGB sehingga citra obyek menjadi seperti gambar berikut.

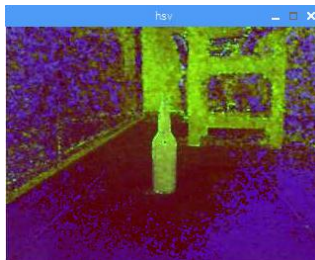


**Gambar 4.8** Citra obyek setelah normalisasi RGB

Normalisasi RGB dilakukan dengan memisahkan komponen R, G, dan B lalu membagi tiga komponen tersebut dengan nilai penjumlahan R, G, dan B sehingga didapatkan citra dengan nilai R, G, dan B yang baru.

#### 4.1.2.3 Konversi RGB ke HSV

Proses konversi RGB ke HSV mengubah citra dari ruang warna RGB ke ruang warna HSV. Proses konversi diperlukan sebelum dilakukan *thresholding* dengan nilai kalibrasi bawah dan atas untuk citra obyek.



**Gambar 4.9** Citra hasil konversi RGB ke HSV

#### 4.1.2.4 Threshold

Proses *Threshold* (pembatasan) yaitu menerapkan nilai bawah

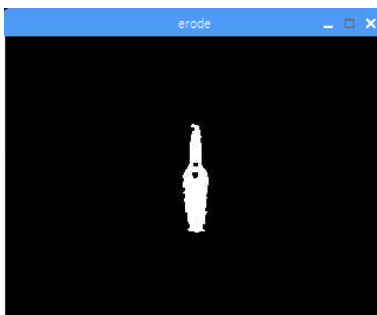
(*lower*) dan nilai atas (*upper*) yang didapatkan dari kalibrasi sebagai nilai batasan sehingga didapatkan gumpalan (*blob*) citra obyek.



**Gambar 4.10** Citra setelah *threshold*

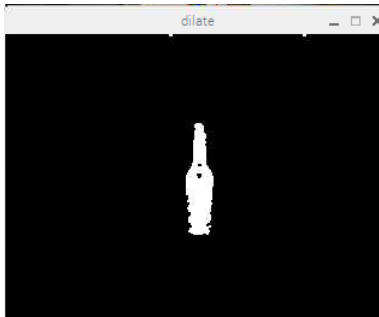
#### 4.1.2.5 Morphology Filter

Citra hasil proses *threshold* masih memiliki *noise* (titik putih) pada *background* dan *hole* (lubang) pada obyek (*foreground*). Untuk mengatasi kedua hal tersebut, dapat dilakukan transformasi morfologi yaitu erosi dan dilasi.



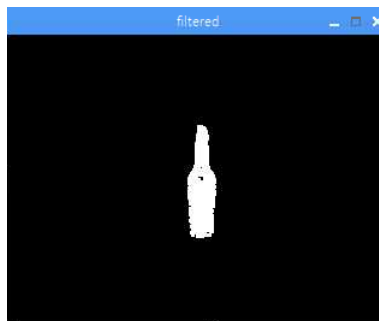
**Gambar 4.11** Citra setelah erosi

Erosi membantu menghilangkan titik-titik putih (*noise*) yang berada pada *background*. Sementara dilasi membantu mengisi lubang / bagian hitam yang ada pada obyek (*foreground*).



**Gambar 4.12** Citra setelah dilasi

Proses erosi dan dilasi dilakukan beberapa kali hingga didapatkan hasil yang diinginkan. Berikut adalah hasil setelah beberapa kali erosi dan dilasi.



**Gambar 4.13** Citra setelah proses erosi dan dilasi

#### **4.1.2.6** *Find Contours*

Proses *Find Contour* dilakukan setelah citra melalui proses transformasi morfologi beberapa kali. Proses *Find Contours* menghitung kontur dari citra binary. Kontur-kontur ini dapat digambarkan dengan beberapa bentuk seperti lingkaran dan kotak. Pada tugas akhir ini, bentuk yang digunakan yaitu kotak.



**Gambar 4.14** Kotak untuk tanda deteksi obyek

#### 4.1.3 Pengujian pada Intensitas Cahaya berbeda

Pengujian deteksi obyek pertama dilakukan pada kondisi lux (pencahayaannya) lingkungan yang berbeda. Obyek diuji pada jarak (cm) yang sama yaitu 54cm. Pengujian dilakukan untuk mengetahui kinerja sistem saat di tempat dengan lux yang berbeda. Tabel 4.1 menunjukkan hasil pengujian sistem deteksi obyek pada lux berbeda.

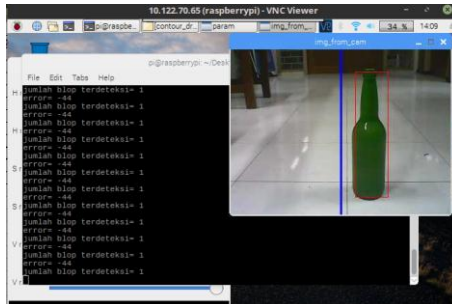
**Tabel 4.1** Pengujian pada intensitas cahaya berbeda

No	Lux	Obyek
1	50.3	Terdeteksi
2	25	Terdeteksi
3	1.4	Terdeteksi
4	0.6	Terdeteksi
5	0.2	Tidak terdeteksi
6	0	Tidak terdeteksi
7	13	Terdeteksi
8	57.9	Terdeteksi
9	36.7	Terdeteksi
10	35.1	Terdeteksi

Pengujian dilakukan di lab B402 dengan intensitas cahaya rentang 0 - 57,9 lux. Tabel 4.1 menunjukkan bahwa obyek terdeteksi pada intensitas cahaya rentang 0,6 – 57,9 lux. Obyek berhasil dideteksi oleh sistem karena citra yang diproses pada range 0,6 – 57,9 dapat dihasilkan *blob* (gumpalan) obyek sehingga dapat dilakukan *FindContour* yang memungkinkan untuk membuat kotak sebagai tanda

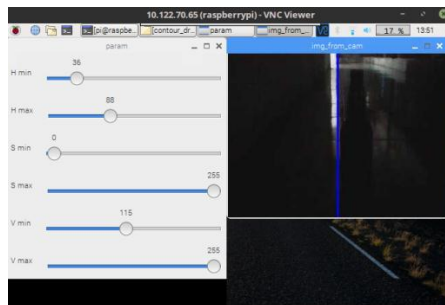


deteksi.



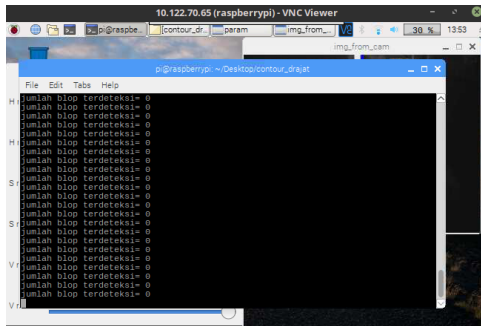
**Gambar 4.15** Citra objek berhasil dideteksi

Tabel 4.1 juga menunjukkan bahwa sistem gagal melakukan deteksi objek pada intensitas cahaya dengan rentang 0 hingga 0.2 lux. Citra yang diproses pada *range* 0 hingga 0.2 tidak dapat dihasilkan *blob* (gumpalan) sehingga tidak memungkinkan untuk dilakukan *Find Contours*.



**Gambar 4.16** Citra objek gagal dideteksi

Proses *Find Contours* bergantung pada *blob* yang dihasilkan sehingga dapat dimunculkan gambar kotak merah sebagai tanda deteksi objek. Sebaliknya, jika *blob* tidak berhasil dideteksi, maka objek juga gagal dideteksi.



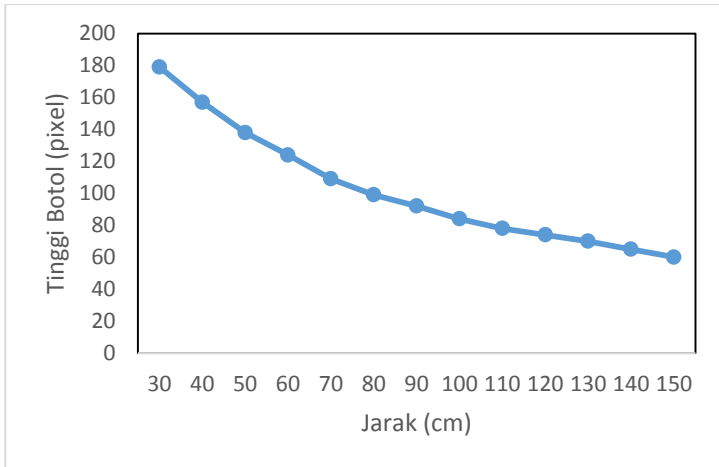
Gambar 4.17 Blob tidak terdeteksi saat citra obyek gagal dideteksi

#### 4.1.4 Pengukuran Tinggi Botol

Dalam kontrol pergerakan robot menuju obyek, robot akan berhenti pada jarak tertentu dari obyek (botol). Jarak dimana robot berhenti dapat ditentukan dari tinggi botol yang berubah-ubah pada citra yang ditangkap kamera. Hasil pengukuran dijelaskan pada table 4.3.

Tabel 4.2 Pengukuran tinggi botol dengan jarak berbeda

Pengukuran Tinggi Botol	
Jarak (cm)	Pixel
30	179
40	157
50	138
60	124
70	109
80	99
90	92
100	84
110	78
120	74
130	70
140	65
150	60



**Gambar 4.18** Pengukuran tinggi botol terhadap jarak

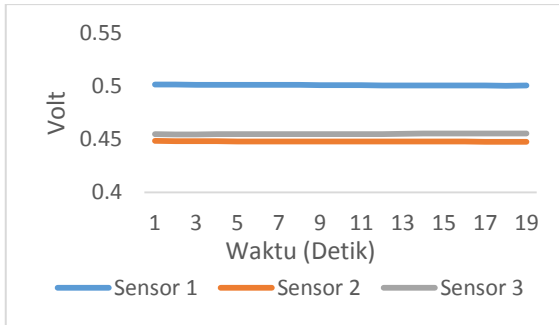
Gambar 4.18 menjelaskan grafik hubungan antara tinggi botol dan jarak. Pola yang terjadi dari jarak 30 cm dan selebihnya dimana semakin jauh jarak botol dari robot, maka ukuran botol semakin kecil. Dari pengukuran tinggi botol ini, juga dapat ditentukan jarak berhenti robot terhadap botol berdasarkan ukuran tinggi botol yang telah didapatkan.

## 4.2 Pengujian Sensor Gas Figaro TGS 2600

Sensor gas TGS 2600 dibutuhkan oleh robot untuk mendeteksi gas alkohol saat robot berhenti di dekat obyek. Pengujian sensor gas TGS 2600 berdasarkan jarak dilakukan dengan tujuan untuk mendapatkan respon tiap sensor gas TGS 2600 pada tiap sudut robot.

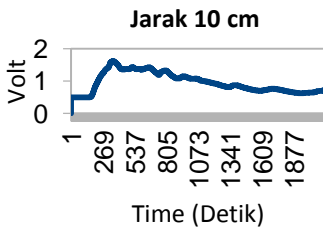


**Gambar 4.19** Pengujian TGS 2600

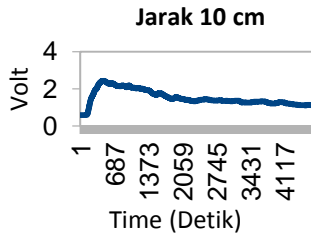


**Gambar 4. 20** Tegangan sensor yang terukur saat tidak ada gas alkohol atau saat udara bersih.

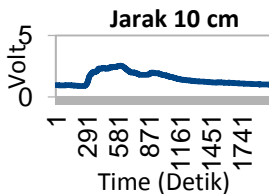
Prosedur pengujian sensor gas TGS 2600 dilakukan dengan bantuan kipas dc 12v untuk tiap sensor gas. Gas yang diujikan yaitu gas alkohol. Pengujian dilakukan dengan menempatkan botol pada jarak 10cm, 20cm, 30cm, 40cm, dan 50cm dari sensor gas. Gas alkohol yang diujikan tiap pengujian sebesar 0.5mL dan disiramkan ke bagian luar botol dengan alat suntik. Alkohol cair disiramkan pada bagian luar botol sebagai ilustrasi saat ada kebocoran atau tumpahan alkohol dari botol. Didapatkan lima belas kurva untuk menjelaskan respon sensor gas TGS 2600 untuk masing-masing pengujian jarak.



(a)



(b)



(c)

**Gambar 4.21** Respon sensor gas 1 (a), gas 2 (b), dan gas 3 (c) pada jarak 10 cm.

Gambar 4.20 menggambarkan respon sensor gas 1, 2, dan 3 saat mendeteksi gas alkohol dengan jarak 10 cm dari sumber gas. Kenaikan nilai tegangan terjadi karena adanya gas alkohol yang terdeteksi oleh sensor gas. Pada lampiran C, terdapat respon masing-masing sensor gas untuk masing-masing jarak pengujian. Fluktuasi yang terjadi pada grafik dikarenakan masuknya uap gas alkohol yang baru sehingga terdeteksi lagi oleh sensor gas. Fluktuasi yang terjadi juga bisa bernilai lebih besar atau lebih kecil bergantung pada konsentrasi uap gas alkohol yang terdeteksi. Nilai pembacaan volt saat terdeteksi gas alkohol kemudian dijadikan sebagai acuan untuk menyalakan *buzzer*.

### 4.3 Pengujian Gerakan Robot

Robot digerakkan dengan tiga roda omni pada penelitian ini. Dilakukan pengukuran kecepatan tiap motor yang digunakan yaitu motor DC RS-545SH-2860 serta pengujian arah gerakan robot.

#### 4.3.1 Pengukuran Kecepatan Motor DC RS-545SH-2860

Pengukuran kecepatan motor DC RS-545SH-2860 dilakukan dengan memberi input tegangan dengan *duty cycle* yang berbeda. Tujuan pengukuran ini yaitu untuk mengetahui kecepatan (rpm) yang dihasilkan tiap motor DC karena arah pergerakan robot ditentukan oleh kombinasi kecepatan motor DC. Prosedur pengukuran dilakukan dengan *Digital Photo Tachometer* DT-2234C. Hasil pengukuran kecepatan motor ditunjukkan pada table 4.4.

**Tabel 4.3** Pengujian kecepatan motor DC RS-545SH-2860

Duty Cycle (%)	Kecepatan Motor DC (RPM)		
	Motor 1	Motor 2	Motor 3
10	5.4	5.5	6
20	9.2	9.7	9.5
30	11	11.6	13.5
40	16.7	17.3	17.6
50	20.6	21.2	21.9
60	24.6	25	25.2
70	28.7	29	29.1
80	32.6	32.9	33.2
90	36.6	37.1	37.4
100	41.5	41.9	41.9

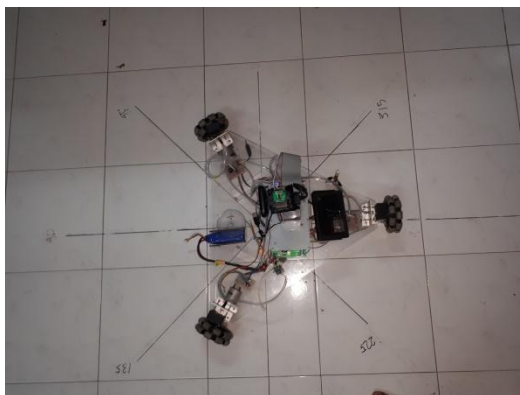
Dari pengukuran kecepatan tiga motor DC yang telah dilakukan, didapatkan hasil yang menunjukkan bahwa saat diberi input *duty cycle* yang sama, tiga motor DC memiliki respon kecepatan yang berbeda.

### 4.3.2 Pengujian Arah Gerakan Robot

Pengujian arah gerakan robot dilakukan untuk mengetahui error dari gerakan robot omni. Pengujian dilakukan dengan mengirim perintah robot omni untuk bergerak pada sudut-sudut yang diujikan. Prosedur pengukuran error dilakukan dengan busur sebagai alat ukur simpangan yang terjadi. Penanda berupa spidol ditempelkan searah sudut uji pada badan robot untuk mengetahui simpangan arah gerakan robot. Gambar 4.21 menjelaskan pengujian arah gerakan robot. Hasil pengukuran dengan busur ada di lampiran D.

**Tabel 4.4** Pengujian arah gerakan robot

Sudut Uji	Pengukuran Busur	Error (%)
0	0	0
45	45	0
90	88	2.22
135	133	1.48
180	180	0



**Gambar 4.21** Pengujian arah gerakan robot

Error muncul pada pengujian dengan sudut 90 derajat dan 135. Pada pengujian sudut 90, error yang muncul sebesar 2,22%.. Sementara

pada sudut 135, error sebesar 1,48%. Error pada sudut 90 muncul bisa diakibatkan karena kecepatan salah satu motor lebih cepat responnya saat diberi *duty cycle* yang sama. Hal yang sama juga berlaku pada sudut 135.

#### 4.4 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dengan tujuan untuk mengetahui tingkat keberhasilan sistem. Sistem dinyatakan berhasil apabila berhasil mendeteksi citra botol hijau, bergerak menuju botol hijau, berhenti di dekat botol hijau, dan membunyikan *buzzer* saat terdeteksi gas alkohol.

Prosedur pengujian keseluruhan sistem yaitu dengan meletakkan botol hijau yang sudah disiram dengan 2 mL alkohol cair pada bagian luar botol pada sudut tertentu dari posisi robot. Hasil pengujian dicatat pada tabel pengujian keseluruhan berikut.

**Tabel 4.5** Pengujian keseluruhan dengan jarak 3 meter, sudut 90, dan 60,9 lux pada *background* tidak hijau

NO	Obyek Terdeteksi		Pengereman		Gas Terdeteksi		Berhasil
	Ya	Tidak	Ya	Tidak	Ya	Tidak	
1	✓		✓			✓	Tidak
2	✓		✓			✓	Tidak
3	✓		✓		✓		Ya
4	✓		✓		✓		Ya
5	✓		✓		✓		Ya
6	✓		✓		✓		Ya
7	✓		✓		✓		Ya
8	✓		✓		✓		Ya
9	✓		✓		✓		Ya
10	✓		✓		✓		Ya

Robot dinyatakan berhasil apabila mampu mendeteksi obyek, berhenti di dekat obyek, dan menyalakan *buzzer* (bunyi) saat terdeteksi gas alkohol. Tabel 4.6 menjelaskan hasil yang didapat saat pengujian keseluruhan dengan jarak 3 meter, sudut 90, dan 60,9 lux pada *background* tidak hijau. Dari sepuluh kali percobaan yang dilakukan, tingkat keberhasilan robot menuju obyek dan melakukan pengereman yaitu 100% dan tingkat keberhasilan robot mendeteksi gas yaitu 80%.





## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan percobaan yang telah dilakukan pada penelitian ini didapat beberapa kesimpulan yang didasarkan pada pengujian sistem. Pada pengujian deteksi citra obyek parameter lux, sistem mampu mendeteksi citra obyek pada lingkungan dengan intensitas cahaya 0,6 – 57,9 lux. Sistem tidak bisa mendeteksi citra obyek pada lingkungan dengan intensitas cahaya 0-0.2 lux.

Dari pengujian tinggi citra botol terhadap jarak dapat ditarik kesimpulan bahwa, ukuran tinggi citra botol (pixel) semakin mengecil seiring bertambahnya jarak botol dari robot. Pada pengujian sensor gas TGS 2600 terhadap jarak, kenaikan nilai tegangan terjadi karena adanya gas alkohol yang terdeteksi oleh sensor gas. Sensor gas mampu mendeteksi gas alkohol pada rentang 10-50cm. Pengujian arah gerakan robot memberikan informasi error terjadi untuk arah pergerakan robot pada sudut 90 dan 135. Error diukur dari simpangan sudut yang terjadi saat robot bergerak menurut sudut yang diujikan. Simpangan ini muncul karena respon motor DC yang berbeda saat diberi input nilai duty cycle yang sama.

Pengujian keseluruhan sistem menunjukkan tingkat keberhasilan robot menuju obyek sebesar 100% pada sudut 90 saat diuji dengan *background* tidak hijau.

#### **5.2 Saran**

Saran untuk pengembangan penelitian ini yaitu metode untuk deteksi obyek 360 derajat seperti penggunaan kamera tipe omnidirectional yang memiliki *field of view* 360 derajat dan lensa *fish eye* agar dapat mendeteksi obyek di segala sudut sehingga mampu memaksimalkan kelebihan robot omni-directional.



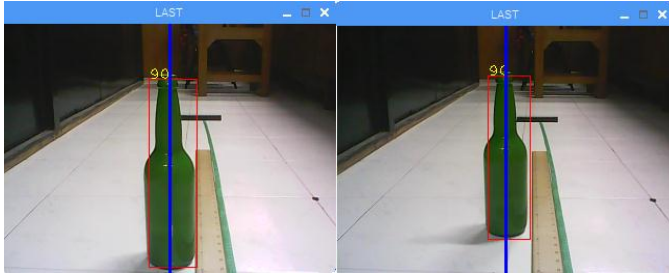
## DAFTAR PUSTAKA

- [1] D. A. Kalimuddin, "Rancang Bangun Omni-Directional Wheels Robot Sebagai Gas Tracker Berbasis Mikrokontroler STM32F4-Discovery," dalam *Tugas Akhir*, Surabaya, 2016.
- [2] Harianto, M. Rivai dan D. Purwanto, "Implementation of Electronic Nose in Omni-directional Robot," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 3, pp. 399-406, June 2013.
- [3] I. R. Syam dan W. Erlangga, "Rancang Bangun OMNI WHEELS ROBOT dengan Roda Penggerak Independent," Makassar, 2012.
- [4] F. Engineering, March 2005. [Online]. Available: [http://www.figarosensor.com/common\(1104\).pdf](http://www.figarosensor.com/common(1104).pdf).
- [5] W. S. Pambudi, "Rancang Bangun 3 Wheels Omni-Directional Mobile Robot Menggunakan Sensor Position Sensitive Device (PSD) Serta Sensor Vision Dengan Metode Kendali Fuzzy Logic Controller Untuk Menghindari Halangan," Batam, 2011.
- [6] Hadha Afrisal, "Portable Smart Sorting and Grading Machine for Fruits Using Computer Vision," *International Conference on Computer, Control, Informatics and Its Applications*, pp. 71-75, 2013.
- [7] S. C. A. G. Martin Loesdau, "Chromatic Indices in the Normalized RGB Color Space," dalam *Research Gate*, Punauia Tahiti, 2017.
- [8] J. S. Wibowo, "Deteksi dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV," *Jurnal Teknologi Informasi DINAMIK*, vol. 16, pp. 118-123, Juli 2011.
- [9] P. a. S. R.K.Munje, "Performance Comparison of PI/PID Controllers for DC Motor," dalam *Annual IEEE India Conference*, Panchavati, 2014.
- [10] G. B. & A. Kaehler, *Learning OpenCv*, Sebastopol: O'reilly Media, 2008.
- [11] F. Hadi, "Rancang Bangun Robot Pencari Lokasi Gas Menggunakan Prinsip Stereo Nose," Surabaya, 2009.



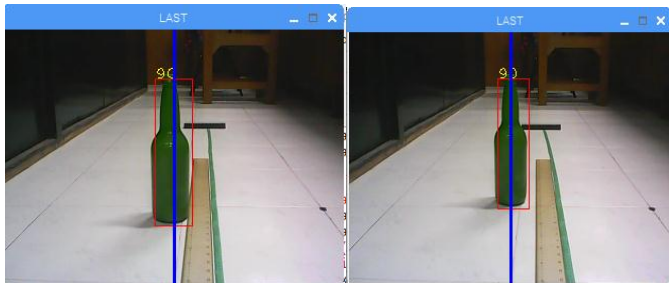
## LAMPIRAN A

### Dokumentasi Pengujian Tinggi Botol



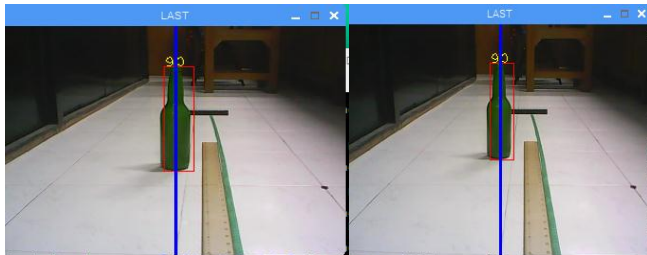
Jarak 30cm

Jarak 40cm



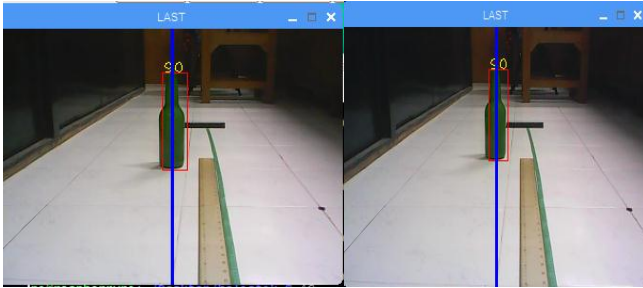
Jarak 50cm

Jarak 60cm



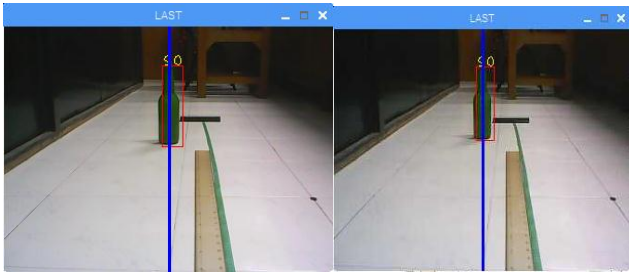
Jarak 70cm

Jarak 80cm



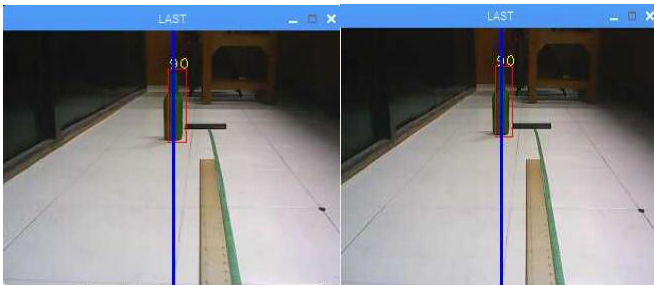
Jarak 90cm

Jarak 100cm



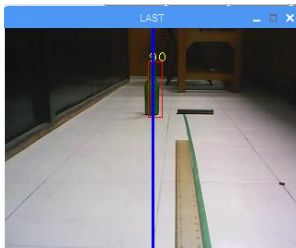
Jarak 110cm

Jarak 120cm



Jarak 130cm

Jarak 140cm

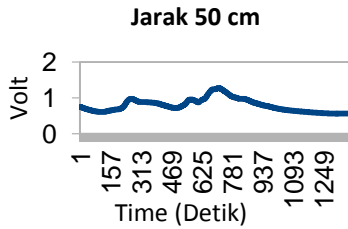
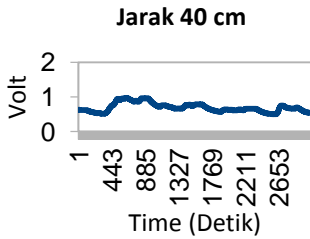
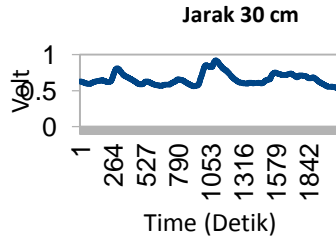
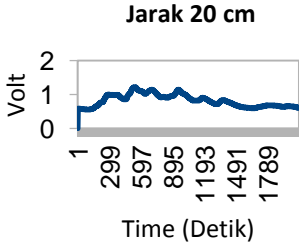


Jarak 150cm

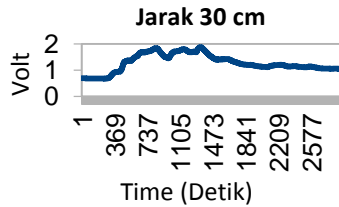
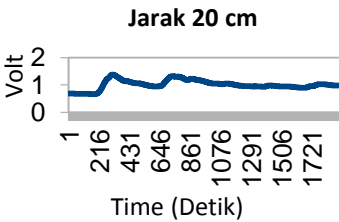
## LAMPIRAN B

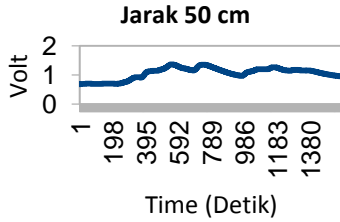
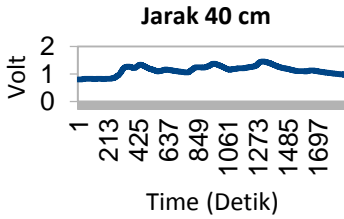
### Dokumentasi Pengujian Sensor Gas TGS 2600

Respon sensor gas 1:

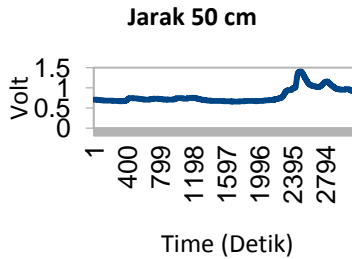
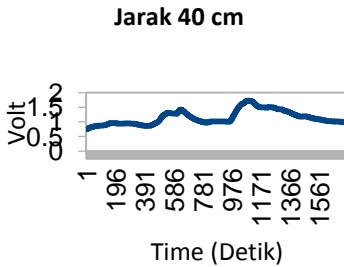
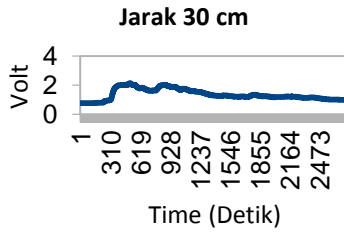
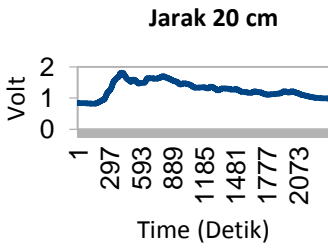


Respon sensor gas 2:





Respon sensor gas 3:





**LAMPIRAN C**  
Dokumentasi Pengujian Arah Gerakan Robot



Sudut 90

Sudut 270



Sudut 180

Sudut 0 (360)



Sudut 225

Sudut 315



Sudut 45

Sudut 135

## LAMPIRAN D

Listing Code Deteksi Obyek:

```
getStructuringElement(MORPH_RECT,cv::Size(2,2));
int h_min,h_max,s_min,s_max,v_min,v_max;
int main(int argc,char** argv){
    cv::VideoCapture cap;
    if(argc==1){
        cap.open(0);
    }
    if(!cap.isOpened()){
        std::cerr << "tidak dapat menampilkan gambar dari kamera"
        <<std::endl;
        return -1;
    }
    cv::Mat gambar,hsv,buffer;
    cv::namedWindow("param",CV_WINDOW_NORMAL);
    h_min = 36, h_max = 88, s_min = 0, s_max = 255, v_min = 115,
    v_max = 255;
    while(1){
        cap>> gambar;
        if(gambar.empty())break;

        cv::createTrackbar("H min", "param",&h_min,255);
        cv::createTrackbar("H max", "param",&h_max,255);
        cv::createTrackbar("S min", "param",&s_min,255);
        cv::createTrackbar("S max", "param",&s_max,255);
        cv::createTrackbar("V min", "param",&v_min,255);
        cv::createTrackbar("V max", "param",&v_max,255);

        //flip image horizontally
        //cv::flip(gambar,gambar,1);

        cv::resize(gambar,gambar,cv::Size(320,240),1,1,cv::INTER_
        CUBIC);
        buffer= gambar.clone();
        //cv::imshow("cam",gambar);
        //RGB normalization
        for(int x=0;x<gambar.cols;x++){
```

```

    for(int y=0;y<gambar.rows;y++){
        Vec3b color= gambar.at<Vec3b>(Point(x,y));
        int total= color[0]+color[1]+color[2];
        if(total==0)total=1;

        color[0]= color[0]*255/total;
        color[1]= color[1]*255/total;
        color[2]= color[2]*255/total;
        gambar.at<Vec3b>(Point(x,y))=color;
    }
}
//cv::imshow("rgbnorm",gambar);
cv::erode(gambar,gambar,element);

//rgb to hsv conversion
cv::cvtColor(gambar,hsv,cv::COLOR_BGR2HSV);
//cv::imshow("hsv",hsv);
//threshold

cv::inRange(hsv,cv::Scalar(h_min,s_min,v_min),cv::Scalar(h_max,s_max,v_max),hsv);
//cv::imshow("thresh",hsv);
//morphology filter
cv::erode(hsv,hsv,element);
//cv::imshow("erode",hsv);
cv::dilate(hsv,hsv,element);
//cv::imshow("dilate",hsv);
cv::erode(hsv,hsv,element);
cv::dilate(hsv,hsv,element);
cv::dilate(hsv,hsv,element);
//cv::imshow("filtered",hsv);
//find contour
vector<vector<Point> > contours;

cv::findContours(hsv,contours,cv::RETR_LIST,cv::CHAIN_APPROX_NONE);
Mat mask= Mat::zeros(hsv.rows,hsv.cols,CV_8UC1);

//hitung contour

```

```

int blop_detected=0;
int height_of_bottle,galat;
char buff[33];
vector<int> sortIdx(contours.size());
vector<float> areas(contours.size());
vector<Rect> boundRect(contours.size());
vector<vector<Point>> contours_poly(contours.size());

for(int n=0;n<(int)contours.size();n++){
    areas[n]= contourArea(contours[n],false);
    if(areas[n]> 400){
        blop_detected++;
        cv::Point top_left,bottom_right;
        int x_center;

        //mendeteksi dan memberikan ciri pada objek yang
terdeteksi

        approxPolyDP(contours[n],contours_poly[n],3,true);
        boundRect[n]= boundingRect(contours_poly[n]);

        rectangle(buffer,boundRect[n].tl(),boundRect[n].br(),cv::Scalar(0,0,255),1);

        //mencari koordinat tengah dari objek yang terdeteksi
        top_left= boundRect[n].tl();
        bottom_right= boundRect[n].br();
        x_center= (top_left.x+bottom_right.x)/2;
        height_of_bottle= abs(bottom_right.y-top_left.y);
        galat= (gambar.cols/2)-x_center;
        galat= (9*galat/32)+90;

        //memberikan label ke objek yang terdeteksi
        sprintf(buff,"%d",galat);

        cv::putText(buffer,buff,boundRect[n].tl(),cv::FONT_HERSHEY_
EY_PLAIN,1,cv::Scalar(0,255,255),1,8);
    }
}

```

```
cv::line(buffer,cv::Point(buffer.cols/2,0),cv::Point(buffer.cols/2,buffer.rows),cv::Scalar(255,0,0),2,LINE_8);
```

```
    //write to file
        int findBlop= blop_detected;
    if(findBlop== 0){
        galat=90;
        height_of_bottle= 0;
    }

        ofstream mytextfile("sudut.csv");
    if(mytextfile.is_open()){
        mytextfile<< galat;
        mytextfile.close();
    }

    ofstream mytextfile1("bottleSize.csv");
    if(mytextfile1.is_open()){
        mytextfile1<< height_of_bottle;
        mytextfile1.close();
    }

    ofstream mytextfile2("blop.csv");
    if(mytextfile2.is_open()){
        mytextfile2<< findBlop;
        mytextfile2.close();
    }

    //showing images
    cv::imshow("LAST",buffer);

    //release memory
    buffer.release();
    hsv.release();
    gambar.release();

    if(cv::waitKey(33) >= 0)break;
}
return 0;
```

```
}
```

Listing code Gerakan Robot:

```
#include <wiringPi.h>
#include <softPwm.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <ads1115.h>
#include <time.h>

//pin GPIO raspi yang digunakan
#define CW_3      17 //pin GPIO17
#define CCW_3     18 //pin GPIO18
#define CW_2     22 //pin GPIO22
#define CCW_2     23 //pin GPIO23
#define CW_1     12 //pin GPIO12
#define CCW_1     13 //pin GPIO13
#define BUZZER    24 //pin GPIO24
#define SIMPANGAN 40 //simpangan awal start
#define SERVO     26 //pin buat servo

//parameter-parameter lain
#define AMBANG    1 //dalam volt
#define ITERASI   10
#define ITERASI1  2300
#define RADIUS    0.2 //dalam m
#define ALPHA     0 //sudut orientasi
#define RAD       57.3 //jari-jari roda

//nomor motor
#define MOTOR_1   0
#define MOTOR_2   1
#define MOTOR_3   2

#define MY_BASE   2222
#define FULL_SCALE 32767
```

```

#define VREF      3.3

const int sudut[3]= { 120,60,0};
char buff[1024];
int dev_gy26,dev_ads1115;
float kanal0,kanal1,kanal2,kanal3;
int gelo_state= 1;
int max_speed;

int sudut_servo(int beta){
    int temp= 7+((18*beta)/180);
return temp;
}

int  pwm_motor(int  kec_translasi,int  nomor_motor,int  arah,int
kec_putar){
    int kec_x= kec_translasi*cos(arah/RAD);
    int kec_y= kec_translasi*sin(arah/RAD);
    int temp=
(kec_x*cos((sudut[nomor_motor]+ALPHA)/RAD))+kec_y*sin((sudut[
nomor_motor]+ALPHA)/RAD))+(RADIUS*kec_putar);
    return temp;
}

void stop_all(int sizeofbottle,int low_threshold){
    if(sizeofbottle> low_threshold){
        softPwmWrite(CW_1,0);
        softPwmWrite(CCW_1,0);
        softPwmWrite(CW_2,0);
        softPwmWrite(CCW_2,0);
        softPwmWrite(CW_3,0);
        softPwmWrite(CCW_3,0);
    }
}

int main(int argc,char* argv[]){
    fprintf(stderr,"program inverse kinematics motor omniwheel\n");

    if(2 != argc){

```

```

        fprintf(stderr,"parameter yang dimasukkan kurang!!\n");
        exit(1);
    }
    //init orientasi robot
    int dir= atoi(argv[1]);
    int sizeBottle,sumBlot;

    wiringPiSetupGpio();

    //pin mode untuk channel motor 3 sebagai output PWM
    pinMode(CW_3,OUTPUT);
    softPwmCreate(CW_3,0,100);
    pinMode(CCW_3,OUTPUT);
    softPwmCreate(CCW_3,0,100);

    //pin mode untuk channel motor 2 sebagai output PWM
    pinMode(CW_2,OUTPUT);
    softPwmCreate(CW_2,0,100);
    pinMode(CCW_2,OUTPUT);
    softPwmCreate(CCW_2,0,100);

    //pin mode untuk channel motor 1 sebagai output PWM
    pinMode(CW_1,OUTPUT);
    softPwmCreate(CW_1,0,100);
    pinMode(CCW_1,OUTPUT);
    softPwmCreate(CCW_1,0,100);

    if(dir ==0){
        softPwmWrite(CW_1,0);
        softPwmWrite(CCW_1,0);
        softPwmWrite(CW_2,0);
        softPwmWrite(CCW_2,0);
        softPwmWrite(CW_3,0);
        softPwmWrite(CCW_3,0);
        exit(1);
    }
    dev_ads1115= ads1115Setup(MY_BASE,0x49);
    //pin mode untuk servo sebagai output PWM
    int alpa= 90;

```



```

pinMode(SERVO,OUTPUT);
softPwmCreate(SERVO,0,200); //frekuensi PWM 50Hz
int i;
i=0;
while(i< ITERASI1){
    softPwmWrite(SERVO,sudut_servo(alpa)); //inisialisasi
sudut servo
    i++;
    delay(1);
}
//pin mode output untuk BUZZER
i=0;
pinMode(BUZZER,OUTPUT);
while(i< ITERASI){
    digitalWrite(BUZZER,HIGH);
    delay(50);
    digitalWrite(BUZZER,LOW);
    delay(50);
    i++;
}
int count=0;
printf("|DATA ADC KANAL 0|;|DATA ADC KANAL 1|;|DATA
ADC KANAL 2|\n");

while(1){
    FILE *ptr_file;
    ptr_file= fopen("sudut.csv", "r");
    if(ptr_file!=NULL){
        char tmp[33];
        while(fgets(buff,1024,ptr_file)){
            dir= atoi(buff);
        }
        fclose(ptr_file);
    }
    ptr_file= fopen("bottleSize.csv", "r");
    if(ptr_file!=NULL){
        char tmp[33];
        while(fgets(buff,1024,ptr_file)){
            sizeBottle= atoi(buff);
        }
    }
}

```

```

        }
        fclose(ptr_file);
    }
    ptr_file= fopen("blop.csv", "r");
    if(ptr_file!=NULL){
        char tmp[33];
        while(fgets(buff,1024,ptr_file)){
            sumBlop= atoi(buff);
        }
        fclose(ptr_file);
    }
    if(sumBlop== 0){
        max_speed= 30;
        //bagian gelo-gelo
        if(gelo_state){
            softPwmWrite(SERVO,sudut_servo(alpa));
            if(count> ITERASI1){
                printf("sudut= %d; count= %d\n",alpa,count);
                alpa+=2;
                count=0;
            }
            if(alpa> 175){
                gelo_state=0;
            }
        }
        else{
            softPwmWrite(SERVO,sudut_servo(alpa));
            if(count> ITERASI1){
                printf("sudut= %d; count= %d\n",alpa,count);
                alpa-=2;
                count=0;
            }
            if(alpa< 5){
                gelo_state=1;
            }
        }
        dir=dir;
    }
    else{

```

```

        if(sizeBottle<138){
            softPwmWrite(SERVO,sudut_servo(alpa));
            max_speed= 100;
            dir= dir+(alpa-90);
        }
        else{
            stop_all(sizeBottle,138);
            max_speed= 0;
            //ads1115 section
            kanal0=(analogRead(MY_BASE+0)*VREF)/FULL_SCALE;
            kanal1=(analogRead(MY_BASE+1)*VREF)/FULL_SCALE;
            kanal2=(analogRead(MY_BASE+2)*VREF)/FULL_SCALE;

            if((kanal0>=AMBANG)|(kanal1>=AMBANG)|(kanal2>=
            AMBANG)){
                int i= 0;
                while(i< ITERASI){
                    digitalWrite(BUZZER,HIGH);
                    delay(50);
                    digitalWrite(BUZZER,LOW);
                    delay(50);
                    i++;
                }
                printf("%f;%f;%f\n",kanal0,kanal1,kanal2);
            }
        }
        int kec_1,kec_2,kec_3;
        kec_1= pwm_motor(max_speed,MOTOR_1,dir,0);
        kec_2= -1*pwm_motor(max_speed,MOTOR_2,dir,0);
        kec_3= pwm_motor(max_speed,MOTOR_3,dir,0);

        //untuk motor 1
        if(kec_1>0){
            softPwmWrite(CW_1,abs(kec_1));
            softPwmWrite(CCW_1,abs(0));
        }
        else{
            softPwmWrite(CCW_1,abs(kec_1));

```

```

        softPwmWrite(CW_1,abs(0));
    }
    //untuk motor 2
    if(kec_2>0){
        softPwmWrite(CW_2,abs(kec_2));
        softPwmWrite(CCW_2,abs(0));
    }
    else{
        softPwmWrite(CCW_2,abs(kec_2));
        softPwmWrite(CW_2,abs(0));
    }

    //untuk motor 3
    if(kec_3>0){
        softPwmWrite(CW_3,abs(kec_3));
        softPwmWrite(CCW_3,abs(0));
    }
    else{
        softPwmWrite(CCW_3,abs(kec_3));
        softPwmWrite(CW_3,abs(0));
    }
    count++;
}
}

```

## **BIODATA PENULIS**



Penulis dilahirkan di Surabaya pada tanggal 10 Mei 1993. Sebagai anak kedua dari dua bersaudara, penulis memulai kegiatan pendidikan formal di SDI Raden Paku Surabaya, yang kemudian dilanjutkan di MTs Unggulan Amanatul Ummah Surabaya, MBI Amanatul Ummah Mojokerto, MA NU TBS Kudus dan pada tahun 2012 penulis diterima sebagai mahasiswa di Departemen Teknik Elektro ITS. Selama menjalani perkuliahan, penulis aktif dalam kepanitiaan mulai dari tingkat jurusan hingga tingkat nasional. Selain itu, penulis juga turut berpartisipasi sebagai asisten praktikum di bidang studi elektronika.

Email : sudrajatdwi@gmail.com  
Hp/WA : 085867693400/081232565172