



TUGAS AKHIR - TE 141599

**PERANCANGAN ROBOT BERODA UNTUK
MENDETEKSI GAS BERBAHAYA MENGGUNAKAN
IMAGE PROCESSING BERBASIS *SINGLE BOARD
COMPUTER***

Hanifar Kahira
NRP 2211100161

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

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

***WHEEL ROBOT DESIGN FOR DANGEROUS GAS
DETECTION USING IMAGE PROCESSING BASED
ON SINGLE BOARD COMPUTER***

Hanifar Kahira
NRP 2211100161

Advisor
Dr. Muhammad Rivai ST., MT.
Rudy Dikairono ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**PERANCANGAN ROBOT BERODA UNTUK
MENDETEKSI GAS BERBAHAYA MENGGUNAKAN
IMAGE PROCESSING BERBASIS SINGLE BOARD
COMPUTER**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I,



Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003

Dosen Pembimbing II,



Rudy Dikairono, ST., MT.
NIP.198103252005011002



PERANCANGAN ROBOT BERODA UNTUK MENDETEKSI GAS BERBAHAYA MENGUNAKAN *IMAGE PROCESSING* BERBASIS *SINGLE BOARD COMPUTER*

Nama : Hanifar Kahira
Pembimbing I : Dr. Muhammad Rivai ST., MT.
Pembimbing II : Rudy Dikairono ST., MT.

ABSTRAK

Gas dapat dideteksi melalui indra penciuman. Akan tetapi hal tersebut dapat membahayakan manusia dikarenakan gas yang di hirup dapat berupa racun. Banyak benda yang mengandung suatu gas berbahaya yang dapat menyebabkan suatu bencana apabila terjadi kebocoran. Oleh karena itu dibutuhkan suatu sistem yang mampu mendeteksi suatu kebocoran gas tanpa merugikan manusia.

Teknologi robot beroda memungkinkan untuk dapat melakukan manuver secara cepat sehingga dapat dimanfaatkan untuk membuat mekanisme kerja robot yang dapat mendeteksi gas berbahaya. Hal ini dimaksudkan untuk mengganti peran manusia dalam mendeteksi bahan berbahaya agar tidak terjadi penambahan korban jiwa. Single Board Computer digunakan karena memiliki kelebihan bersifat open source dan prosesor yang lebih tinggi daripada mikrokontroler. Selain itu robot ini dilengkapi dengan kamera *webcam* sebagai media untuk menangkap obyek sehingga robot dapat berjalan menuju obyek dan mendeteksi kebocoran gas pada benda yang dituju. Sensor gas yang digunakan sensitif terhadap gas LPG, alkohol, Benzine, dan Hexane. Robot ini terbuat dari bahan aluminium dan acrylic yang dapat menahan beban hingga 5 kg.

Pergerakan robot menggunakan kontrol proporsional untuk menuju ke target dengan error rata-rata sebesar 34,806%. Sedangkan robot ini sendiri jika diuji seluruhnya mampu mencapai target dengan kesalahan sebesar 40% dari total pengujian yang dilakukan.

Kata kunci : Gas Berbahaya, Robot Beroda, *Single Board Computer*

WHEEL ROBOT DESIGN FOR DANGEROUS GAS DETECTION USING IMAGE PROCESSING BASED ON SINGLE BOARD COMPUTER

Name : Hanifar Kahira
1st Advisor : Dr. Muhammad Rivai ST., MT.
2nd Advisor : Rudy Dikairono ST., MT.

ABSTRACT

Gas can be detected through the sense of smell. However, it can be dangerous to humans because in inhaled gas can be toxic. Many objects that contain a dangerous gas that can cause a disaster in the event of a leak. Therefore, it needs a system that can detect a gas leak without harming humans.

Wheeled robot technology makes it possible to be able to maneuver quickly so that it can be used to make the robot work mechanism that can detect dangerous gases. It is intended to replace the role of humans in detecting hazardous materials in order to avoid additional casualties. Single Board Computer is used because it has the advantages of open source and processor are higher than the microcontroller. Moreover the robot is equipped with a web camera as a medium for capturing an object so that the robot can walk towards an object and detect gas leaks on the targeted object. Gas sensor used is sensitive to LPG gas, alcohol, Benzine, and Hexane. This robot is made of aluminum and acrylic that can withstand loads up to 5 kg.

The movement of the robot uses proportional control to get to the target with an average error of 34.806%. While this robot itself if tested entirely capable of achieving the target with an error of 40% of the total tests performed.

Keywords : Dangerous Gas, Wheeled Robot, Single Board Computer

KATA PENGANTAR

Alhamdulillah, puji syukur tiada henti penulis haturkan kepada Tuhan semesta alam, Allah SWT, atas seluruh limpahan rahmat, karunia, dan hidayahNya selama ini sehingga penulis mampu menyelesaikan tugas akhir ini.

Dalam mengerjakan tugas akhir ini penulis banyak mendapatkan berbagai macam bantuan dan doa , oleh karena itu penulis mengucapkan terima kasih atas bantuannya dan semoga Allah membalasnya dengan sebaik-baik balasan,

- Bapak, Ibu, adik-adik, serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
- Bapak Dr. Muhammad Rivai ST., MT. selaku dosen pembimbing pertama, yang tanpa bimbingan beliau, mungkin tugas akhir ini tidak akan terselesaikan.
- Bapak Rudy Dikairono ST., MT. selaku dosen pembimbing kedua, yang juga dengan bimbingan beliau, tugas akhir ini dapat terselesaikan dengan baik.
- Tasripan, IR. MT. selaku Koordinator Bidang Studi Elektronika dan dosen wali yang memberikan begitu banyak motivasi dan inspirasi.
- Dr. Tri Arief Sardjono, ST, MT selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
- Seluruh dosen bidang studi elektronika dan seluruh dosen Teknik Elektro ITS.
- Teman-teman seperjuangan asisten laboratorium elektronika.

Penulis sepenuhnya menyadari bahwa tugas akhir ini jauh dari sempurna. Saran dan kritik yang membangun senantiasa penulis terima guna pengembangan lebih lanjut. Penulis berharap agar Tugas Akhir ini dapat memberikan manfaat sebesar-besarnya bagi semua pihak yang ingin mengembangkan sistem lab automation atau manajemen energi.

Surabaya, Juni 2015

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL	
HALAMAN PENGESAHAN	
PERANCANGAN ROBOT BERODA UNTUK MENDETEKSI GAS BERBAHAYA MENGGUNAKAN <i>IMAGE PROCESSING</i> BERBASIS <i>SINGLE BOARD COMPUTER</i>	1
PERANCANGAN ROBOT BERODA UNTUK MENDETEKSI GAS BERBAHAYA MENGGUNAKAN <i>IMAGE PROCESSING</i> BERBASIS <i>SINGLE BOARD COMPUTER</i>	i
ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II	7
2.1. Bahan Berbahaya	7
2.2. Sejarah Robot	11
2.1.1. Mobile Robot	13
2.3. Pengolahan Citra (Image Processing)	14
2.2.1. Deteksi Citra (Image Detection)	15
2.2.2. Smoothing	18

2.2.3. MeanShift	19
2.2.4. CamShift	20
2.4. <i>OpenCV</i>	22
2.5. Kontrol Proporsional	24
2.6. Single Board Computer	25
2.7. Webcam	27
2.8. Mikrokontroler	27
2.9. Ardumoto	32
2.10. Sensor Gas MQ-3	33
BAB III	37
3.1 Perangkat Keras	38
3.1.1. Mekanik Robot	38
3.1.2. Kamera	40
3.1.3. <i>Single Board Computer</i>	40
3.1.4. <i>Interfacing Driver Motor Shield</i> dengan Arduino Uno	41
3.1.5. Sensor Gas	42
3.2 Perancangan <i>Software</i>	45
3.2.1. Perancangan Perangkat Lunak untuk SBC	45
3.2.2. Perancangan Perangkat Lunak Mikrokontroler	49
BAB IV	55
4.1. Pengujian Sensor Gas	55
4.2. Pengujian Driver Motor	56
4.3. Pengujian Deteksi Benda Menggunakan Kamera	58
4.4. Pengujian Kontrol Proporsional	61
4.5. Pengujian Keseluruhan Sistem	62
BAB V	63
5.1 Kesimpulan	63
5.2 Saran	63
DAFTAR PUSTAKA	65
BIODATA PENULIS	89

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Simbol Bahan Berbahaya dan Beracun.....	10
Gambar 2. 2 <i>Mobile Robot</i>	14
Gambar 2. 3 Skema Deteksi Warna HSV.....	16
Gambar 2. 4 Perbedaan HSV dengan HSL.....	18
Gambar 2. 5 Alur Algoritma <i>CamShift</i>	21
Gambar 2. 6 <i>Struktur OpenCV</i>	23
Gambar 2. 7 Diagram Blok Kontroler Proporsional.....	24
Gambar 2. 8 <i>Raspberry Pi</i>	26
Gambar 2. 9 Kamera <i>Webcam</i>	27
Gambar 2. 10 Diagram Blok ATmega 328.....	30
Gambar 2. 11 <i>Arduino Uno</i>	31
Gambar 2. 12 Pin Mapping <i>Arduino Uno</i>	31
Gambar 2. 13 Circuit Diagram IC L298.....	32
Gambar 2. 14 Driver Motor Shield <i>Ardumoto</i>	33
Gambar 2. 15 Sensor Gas MQ3 Tampak Belakang.....	34
Gambar 2. 16 Sensor Gas MQ3 Tampak Depan.....	34
Gambar 2. 17 Sensitivitas Sensor Gas MQ 3.....	35
Gambar 3. 1 Cara Kerja Sistem.....	37
Gambar 3. 2 Diagram Blok Sistem.....	38
Gambar 3. 3 Desain Robot Tampak Miring.....	39
Gambar 3. 4 Desain Robot Tampak Atas.....	39
Gambar 3. 5 Desain Robot Tampak Depan.....	39
Gambar 3. 6 Desain Base Robot.....	40
Gambar 3. 7 Driver Motor <i>Ardumoto</i> yang Dihubungkan <i>Arduino</i> ..	41
Gambar 3. 8 Konfigurasi Pin Pada Driver Motor <i>Ardumoto</i>	42
Gambar 3. 9 Motor DC.....	42
Gambar 3. 10 Sensor Gas MQ-03.....	43
Gambar 3. 11 Bagian Robot dari Atas.....	44
Gambar 3. 12 Bagian Depan Robot.....	45
Gambar 3. 13 Flowchart Deteksi Bola.....	46

Gambar 3. 14 Deteksi Benda	47
Gambar 3. 15 Nilai posisi benda.....	49
Gambar 3. 16 <i>Flowchart</i> Gerakan Robot.....	50
Gambar 4. 1 <i>Pengujian Sensor Gas</i>	55
Gambar 4. 2 Grafik Pengujian Sensor Gas	55
Gambar 4. 3 Pengujian Driver Motor	57
Gambar 4. 4 Pengujian Nilai Hue Dari Obyek	58
Gambar 4. 5 Nilai Histogram Untuk Warna Merah.....	59
Gambar 4. 6 Hasil Histogram Untuk Benda Berwarna Biru.....	60
Gambar 4. 7 Hasil Histogram Untuk Benda Berwarna Hijau.....	60
Gambar 4. 8 Grafik Kontrol Proporsional	61

DAFTAR TABEL

	Halaman
Tabel 4. 1 Hasil Pengujian Sensor Gas Terhadap Jarak Tabel	56
Tabel 4. 2 Hasil Pengujian Driver Motor	57
Tabel 4. 3 Nilai Hue Pada Warna yang Terdeteksi	58
Tabel 4. 4 Tabel Pengujian Sistem.....	62

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Banyak sekali kecelakaan yang terjadi yang diakibatkan oleh kebocoran gas. Disamping itu gas yang mengalami kebocoran merupakan gas yang memiliki senyawa yang dapat membahayakan bagi manusia. Dalam dunia industri banyak sekali benda yang mengandung gas berbahaya yang apabila terjadi kebocoran dapat menyebabkan bencana. Pada tabung LPG apabila terjadi kebocoran, gas yang berada di dalam tabung sangat mudah terbakar sehingga dapat mengakibatkan terjadinya ledakan secara tiba-tiba apabila terkena percikan api. Selain itu banyak gas yang apabila terhirup oleh manusia dapat menyebabkan sakit seperti gas CO. Terhadap kesehatan, gas CO merupakan gas yang berbahaya untuk tubuh karena daya ikat gas CO terhadap Hb adalah 240 kali dari daya ikat CO terhadap O₂. Apabila gas CO darah (HbCO) cukup tinggi, maka akan mulai terjadi gejala antara lain pusing kepala, mual dan sesak nafas, gangguan penglihatan dan konsentrasi menurun tidak sadar, koma dan apabila berlanjut akan dapat menyebabkan kematian. Gas CO yang tinggi di dalam darah dapat berasal dari rokok dan asap dari kendaraan bermotor. Terhadap lingkungan udara dalam ruangan, gas CO dapat pula merupakan gas yang menyebabkan building associated illnesses, dengan keluhan berupa nyeri kepala, mual, dan muntah.

Menurut data Pusat Studi Kebijakan Publik (Puskepi), sejak 2008 hingga Juli 2010, di Indonesia terjadi sebanyak 189 kali kasus ledakan dalam pemakaian tabung gas elpiji rumah tangga. Rinciannya, pada 2008 terjadi 61 kasus, kemudian turun menjadi 50 kasus pada 2009. Tapi kemudian jumlah temuan meningkat tajam hingga pertengahan 2010, mencapai 78 kasus.

Proses pengidentifikasian kebocoran gas biasanya masih bersifat konvensional yaitu dengan cara didekati terlebih dahulu kemudian dicium bau dari gas tersebut. Hal ini tidak efektif dan dapat membahayakan manusia karena dapat memicu terjadinya hal yang tidak diinginkan seperti penyakit pada saluran pernafasan bagi manusia.



Gambar 1. 1 Tabung LPG 3 Kg

Karena membahayakan bagi manusia, perlu dibuat suatu sistem mekanisme kerja robot yang mampu mengenali benda dan mendeteksi kebocoran pada benda tersebut. Robot mampu menggantikan peran manusia dalam melaksanakan tugas-tugas berbahaya bagi manusia sehingga dapat meminimalisir terjadinya suatu bencana. Robot beroda menggunakan penggerak berupa roda yang memungkinkan untuk dapat memaanuver secara cepat sehingga dapat menjangkau tempat-tempat yang sulit dijangkau.

Robot ini dilengkapi dengan sensor gas yang mampu mengenali objek tersebut apakah mengandung gas berbahaya atau tidak. Robot ini juga dilengkapi dengan kamera sehingga dapat membedakan objek yang akan dituju dengan objek lain yang ada. Diharapkan implementasi pada robot ini dapat mampu menuju ke arah sumber bahan peledak dan mengidentifikasi objek tertentu apakah itu berbahaya atau tidak.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah :

1. Bagaimana cara mengolah data dari sensor gas
2. Bagaimana cara mengolah data kamera ke Raspberry Pi
3. Bagaimana cara mengontrol motor menggunakan Machine Vision.
4. Bagaimana cara mengirim data dari Raspberry Pi ke Arduino.

1.3 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Kamera yang digunakan USB Webcam.
2. Single Board Computer yang digunakan Raspberry Pi b+.
3. Mikrokontroler yang digunakan adalah Arduino UNO R3
4. Motor yang digunakan adalah motor DC 12 Volt 38 rpm.
5. Sensor gas yang digunakan adalah MQ-3.
6. Data gambar yang dikirimkan diolah dengan Raspberry Pi b+.
7. Obyek yang dituju tidak terkena halangan atau *obstacle*.
8. Berada di ruangan tertutup dengan cahaya yang cukup.

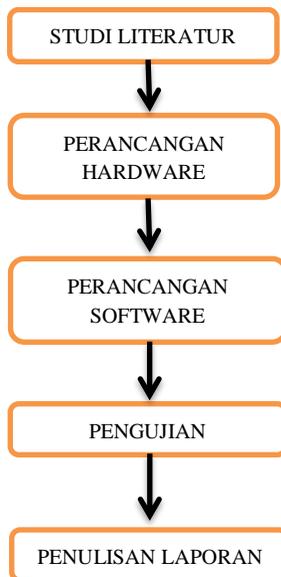
1.4 Tujuan

Penelitian pada tugas akhir ini bertujuan sebagai berikut :

1. Robot mampu menuju target yang telah ditentukan.
2. Robot mampu mendeteksi gas yang bersifat berbahaya dan tidak.

1.5 Metodologi Penelitian

Dalam penyelesaian Tugas Akhir ini digunakan metodologi sebagai berikut :



1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet. Meliputi :

- Pencarian alat dan bahan yang tepat dan sesuai dalam membuat robot.
- Pencarian spesifikasi dari mikrokontroler yang digunakan yaitu Arduino Uno Rev3 dan Single Board Computer Raspberry Pi b+.
- Mempelajari metode *thresholding* untuk menentukan obyek yang dituju dengan warna.
- Penggunaan library *OpenCV* pada proses pengolahan citra.
- Mempelajari bagaimana komunikasi serial antara Raspberry Pi dengan Arduino.

2. Perancangan Hardware

Pada tahap ini dibuat suatu perancangan mobile robot yang meliputi pemilihan motor DC, driver motor, pembuatan *base robot*, dan kamera.

3. Perancangan Software

Pada tahap ini dibuat suatu program untuk dapat mendeteksi obyek dengan metode merubah BGR menjadi HSV (*Hue, Saturation, Value*) dan *threshold*, lalu mengirim data posisi piksel dari frame dari Raspberry Pi ke Arduino melalui komunikasi serial.

4. Pengujian Sistem

Pengujian sistem dilakukan untuk menentukan keandalan dari sistem yang telah dibuat. Pengujian meliputi deteksi benda dan pergerakan robot serta respon sensor gas yang digunakan.

5. Penulisan Laporan Akhir

Tahap penulisan laporan Tugas Akhir dilakukan setelah mendapatkan data – data yang dibutuhkan.

1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

➤ **Bab 1 : Pendahuluan**

Bab ini meliputi latar belakang, perumusan masalah, tujuan, sistematika penulisan, metodologi, dan relevansi.

➤ **Bab 2 : Dasar Teori**

Bab ini menjelaskan tentang berbagai macam teori-teori penunjang dalam pengerjaan tugas akhir ini, diantaranya : teori dasar tentang pengolahan citra HSV, komunikasi serial, kontrol pergerakan robot, driver motor dan Arduino yang digunakan, dan sinyal output sensor gas.

➤ **Bab 3: Perancangan Sistem**

Pada bab ini menjelaskan tentang langkah-langkah dalam merencanakan dan membangun sistem secara keseluruhan baik *software* maupun *hardware* agar didapatkan suatu sistem robot pendeteksi kebocoran gas berbahaya menggunakan image processing.

➤ **Bab 4 : Pengujian dan Analisis**

Bab ini menjelaskan tentang berbagai macam pengujian pada robot dan analisa kesalahan yang dilakukan oleh robot dalam mencapai tujuan akhir yaitu mampu mendeteksi gas berbahaya.

➤ **Bab 5 : Penutup**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangannya.

1.7 Relevansi

Hasil dari tugas akhir ini diharapkan mampu membantu mendeteksi suatu gas yang berbahaya dan menggantikan peran manusia untuk mendeteksi suatu gas yang bersifat berbahaya tanpa menimbulkan korban.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

Tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan teknologi yang telah ada. Sedangkan bagian dasar teori menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem pada tugas akhir ini.

2.1. Bahan Berbahaya

Pengertian B3 atau Bahan Berbahaya dan Beracun menurut OSHA (Occupational Safety and Health of the United State Government) adalah bahan yang karena sifat kimia maupun kondisi fisiknya berpotensi menyebabkan gangguan pada kesehatan manusia, kerusakan properti dan atau lingkungan.

Sedangkan menurut Peraturan Pemerintah Nomor 74 Tahun 2001 tentang Pengelolaan Bahan Berbahaya dan Beracun, B3 didefinisikan sebagai bahan yang karena sifat dan atau konsentrasinya dan atau jumlahnya, baik secara langsung maupun tidak langsung, dapat mencemarkan dan atau merusak lingkungan hidup, dan atau dapat membahayakan lingkungan hidup, kesehatan, kelangsungan hidup manusia serta makhluk hidup lainnya.

Pemerintah Indonesia telah menerbitkan beberapa peraturan terkait pengelolaan Bahan Berbahaya dan Beracun. Peraturan-peraturan tersebut berisikan bagaimana pengelolaan B3 dan tentunya jenis-jenis dan pengelompokkan (penggolongan) Bahan Berbahaya dan Beracun.

Salah satu peraturan yang mengatur pengelolaan B3 adalah Peraturan Pemerintah Nomor 74 Tahun 2001 tentang Pengelolaan Bahan Berbahaya dan Beracun. Dalam PP ini, B3 diklasifikasikan menjadi :

- Mudah meledak (explosive), yaitu bahan yang pada suhu dan tekanan standar (25 0C, 760 mmHg) dapat meledak atau melalui reaksi kimia dan atau fisika dapat menghasilkan gas dengan suhu dan tekanan tinggi yang dengan cepat dapat merusak lingkungan di sekitarnya.
- Pengoksidasi (oxidizing), yaitu bahan yang memiliki waktu pembakaran sama atau lebih pendek dari waktu pembakaran senyawa standar.

- Mangat mudah sekali menyala (*extremely flammable*), yaitu B3 padatan dan cairan yang memiliki titik nyala di bawah 0 derajat C dan titik didih lebih rendah atau sama dengan 35 OC.
- Sangat mudah menyala (*highly flammable*), yaitu bahan yang memiliki titik nyala 0-210C.
- Mudah menyala (*flammable*).
- Amat sangat beracun (*extremely toxic*);
- Sangat beracun (*highly toxic*);
- Beracun (*moderately toxic*), yaitu bahan yang bersifat racun bagi manusia dan akan menyebabkan kematian atau sakit yang serius apabila masuk ke dalam tubuh melalui pernafasan, kulit atau mulut.
- Berbahaya (*harmful*), yaitu bahan baik padatan maupun cairan ataupun gas yang jika terjadi kontak atau melalui inhalasi ataupun oral dapat menyebabkan bahaya terhadap kesehatan sampai tingkat tertentu.
- Korosif (*corrosive*), yaitu bahan yang menyebabkan iritasi pada kulit, menyebabkan proses pengkaratan pada lempeng baja SAE 1020 dengan laju korosi lebih besar dari 6,35 mm/tahun, atau mempunyai pH sama atau kurang dari 2 untuk B3 bersifat asam dan sama atau lebih besar dari 12,5 untuk yang bersifat basa.
- Bersifat iritasi (*irritant*), yaitu bahan padat atau cair yang jika terjadi kontak secara langsung, dan apabila kontak tersebut terus menerus dengan kulit atau selaput lendir dapat menyebabkan peradangan.
- Berbahaya bagi lingkungan (*dangerous to the environment*), yaitu bahaya yang ditimbulkan oleh suatu bahan seperti merusak lapisan ozon (misalnya CFC), persisten di lingkungan (misalnya PCBs), atau bahan tersebut dapat merusak lingkungan.
- Karsinogenik (*carcinogenic*), yaitu bahan yang dapat menyebabkan sel kanker.
- Teratogenik (*teratogenic*), yaitu bahan yang dapat mempengaruhi pembentukan dan pertumbuhan embrio.
- Mutagenik (*mutagenic*), yaitu bahan yang menyebabkan perubahan kromosom (merubah genetika).

Jenis dan klasifikasi Bahan Berbahaya dan Beracun juga diuraikan dalam Keputusan Menteri Kesehatan No. 453/Menkes/Per/XI/1983. Dalam Kepmenkes ini B3 dikelompokkan dalam 4 klasifikasi yaitu :

- Klasifikasi I, meliputi :
 1. Bahan kimia atau sesuatu yang telah terbukti atau diduga keras dapat menimbulkan bahaya yang fatal dan luas, secara langsung atau tidak langsung, karena sangat sulit penanganannya dan pengamanannya;
 2. Bahan kimia atau sesuatu yang baru yang belum dikenal dan patut diduga menimbulkan bahaya.
- Klasifikasi II, meliputi :
 1. Bahan radiasi;
 2. Bahan yang mudah meledak karena gangguan mekanik;
 3. Bahan beracun atau bahan lainnya yang mudah menguap dengan LD50 (rat) kurang dari 500 mg/kg atau yang setara, mudah diabsorpsi kulit atau selaput lendir;
 4. Bahan etilologik/biomedik;
 5. Gas atau cairan beracun atau mudah menyala yang dimampatkan;
 6. Gas atau cairan atau campurannya yang bertitik nyala kurang dari 350C;
 7. Bahan padat yang mempunyai sifat dapat menyala sendiri.
- Klasifikasi III, meliputi :
 1. Bahan yang dapat meledak karena sebab-sebab lain, tetapi tidak mudah meledak karena sebab-sebab seperti bahan klasifikasi II;
 2. Bahan beracun dengan LD50 (rat) kurang dari 500 mg/kg atau setara tetapi tidak mempunyai sifat seperti bahan beracun klasifikasi II;
 3. Bahan atau uapnya yang dapat menimbulkan iritasi atau sensitisasi, luka dan nyeri;
 4. Gas atau cairan atau campurannya dengan bahan padat yang bertitik nyala 350Csampai 600C;
 5. Bahan pengoksidasi organik;
 6. Bahan pengoksidasi kuat;
 7. Bahan atau uapnya yang bersifat karsinogenik, tetragenik dan mutagenik;
 8. Alat atau barang-barang elektronika yang menimbulkan radiasi atau bahaya lainnya.
- Klasifikasi IV, yaitu :
 1. Bahan beracun dengan LD50 (rat) diatas 500 mg/kg atau yang setara;

2. Bahan pengoksidasi sedang;
3. Bahan korosif sedang dan lemah;
4. Bahan yang mudah terbakar.



Gambar 2. 1 Simbol Bahan Berbahaya dan Beracun

Tanpa kita sadari banyak produk di sekitar kita yang seringkali dapat disalahgunakan. Hal ini harus dapat dicegah, termasuk didalamnya adalah mencegah penyalahgunaan inhalan. Inhalan adalah suatu kelompok senyawa yang mudah menguap yang menghasilkan efek toksik yang mirip dengan alkohol. Inhalan terdapat pada pelarut yang mudah menguap, atau aerosol yang biasa digunakan dalam rumah tangga, seperti lem, penghapus cat kuku, pengencer cat, deodoran dan cairan pembersih.

Penyalahgunaan produk-produk yang menghasilkan inhalan tidak kalah bahayanya dibandingkan dengan penyalahgunaan narkotika seperti penggunaan mariyuana, kokain, ganja dan psikotropika yang selama ini telah menjadi perhatian banyak pihak.

Sekarang ini, banyak anak muda yang menghirup uap dari pelarut produk-produk tersebut diatas untuk mendapatkan efek seperti efek dari alkohol, yang dapat dinikmati dengan cepat tanpa menyadari bahwa hal ini dapat menyebabkan dampak serius bagi kesehatan.

Ciri-ciri orang yang menggunakan inhalan antara lain:

- Mata merah, berkaca-kaca atau berair,
- Pengucapan kata- kala yang lambat, bergumam kental dan tidak jelas,
- Terlihat seperti orang mabuk,
- Terdapat noda cat pada tangan atau sekitar mulut,

- Bau bahan kimia di pakaian atau di dalam ruangan,
- Bau mulut yang tidak biasa,
- Sering merasa mual dan / atau kehilangan nafsu makan.

Terdapat 4 kategori inhalan yaitu:

- Pelarut yang mudah menguap, adalah cairan yang menguap pada suhu kamar. Biasanya terkandung dalam produk yang sering digunakan dalam rumah tangga atau industri seperti pengencer cat, cairan yang digunakan untuk mencuci kering (dry cleaning fluids), lem, correction fluids dan spidol.
- Aerosol, adalah produk semprot yang mengandung gas dan cairan. Termasuk didalamnya adalah cat semprot, deodorant dan hair spray.
- Gas, termasuk anestetik seperti eter, kloroform, halotan dan nitrogen oksida yang sering disebut "gas tertawa". Nitrogen oksida adalah gas yang paling sering disalahgunakan dan biasanya terdapat pada kemasan whipped cream dan produk untuk menaikkan tingkat oktan pada mobil balap.
- Nitrit, yang sering dimasukkan sebagai golongan khusus dari inhalan, karena tidak seperti golongan inhalan lain yang kerjanya langsung mempengaruhi sistem saraf pusat, nitrit terutama bekerja dengan memperlebar pembuluh darah dan melemaskan otot. Bila inhalan lain biasanya digunakan untuk mengubah mood, maka nitrit sering digunakan untuk memperbaiki aktifitas seksual. Yang termasuk nitrit adalah sikloheksit, isoamil nitrit dan isobutil nitrit.

2.2. Sejarah Robot

Istilah robot berawal bahasa Cheko “robot” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Pada kamus Webster pengertian robot adalah :

An automatic device that performs function ordinarily ascribed to human beings

(sebuah alat otomatis yang melakukan fungsi berdasarkan kebutuhan manusia).

Dari kamus Oxford diperoleh pengertian robot adalah:

A machine capable of carrying out a complex series of actions automatically, especially one programmed by a computer.

(Sebuah mesin yang mampu melakukan serangkaian tugas rumit secara otomatis, terutama yang diprogram oleh komputer).

Untuk dapat diklasifikasikan sebagai robot, mesin harus memiliki dua macam kemampuan yaitu:

- Bisa mendapatkan informasi dari sekelilingnya.
- Bisa melakukan sesuatu secara fisik seperti bergerak atau memanipulasi objek.

Untuk dapat dikatakan sebagai robot sebuah sistem tidak perlu meniru semua tingkah laku manusia, namun suatu sistem tersebut dapat mengadopsi satu atau dua saja sistem yang ada pada diri manusia saja sudah dapat dikatakan sebagai robot. Sistem yang diadopsi berupa sistem penglihatan (mata), sistem pendengaran (telinga) ataupun sistem gerak.

Sebuah robot dapat saja dibuat untuk berbagai macam aktifitas, namun sebuah robot harus dibuat dengan tujuan untuk kebaikan manusia. Ada hukum robotika yang perlu dipegang sebelum seseorang terjun dalam robotika, antara lain:

- Robot tidak boleh menciderai manusia atau dalam keadaan tanpa aksi mengizinkan manusia mendekat untuk disakiti.
- Robot harus menuruti perintah yang diberikan oleh manusia kecuali jika perintah tersebut bertentangan dengan hukum yang pertama.
- Robot harus melindungi eksistensinya, selama tidak bertentangan dengan hukum pertama dan kedua.

Sesuai dengan definisi robot di atas, maka terdapat banyak kegunaan dari robot. Secara umum kegunaan robot adalah untuk menggantikan kerja manusia yang membutuhkan ketelitian yang tinggi atau mempunyai resiko yang sangat besar atau bahkan mengancam keselamatan manusia. Maka untuk mengurangi resiko kerja tersebut perlu digunakan robot yang menggantikan kerja manusia di bidang tersebut, sehingga resiko kecelakaan kerja dapat dikurangi bahkan dihilangkan.

Ada juga sebagian robot yang sengaja diciptakan untuk menemani manusia di dalam aktifitasnya. Robot-robot ini dapat disebut robot bermain. Robot ini diciptakan untuk membantu manusia yang mengalami kesepian diri sehingga dapat mempunyai teman. Robot-robot yang termasuk jenis ini termasuk antar lain Battle Bots, Robot contest, Robot anjing.

2.1.1. Mobile Robot

Robot mobil (mobile robot) adalah sebuah mesin otomatis yang mampu bergerak pada suatu kondisi tertentu. Robot mobil diklasifikasikan menjadi dua, yaitu menurut lingkungan tempat robot tersebut bekerja dan alat yang digunakan untuk bergerak. Berdasarkan lingkungan tempat robot tersebut bekerja, robot mobil terbagi menjadi empat macam:

- Robot yang bekerja di atas permukaan tanah (land robot),
- Robot udara yang biasa disebut unmanned aerial vehicle (UAV),
- Autonomous underwater vehicles (AUVs),
- Robot yang bekerja pada lingkungan kutub—robot yang berkerja pada kondisi permukaan tanah yang dilapisi es (polar robots).

Sedangkan berdasarkan alat yang digunakan untuk bergerak, robot mobil terbagi menjadi robot berlengan atau berkaki—lengan atau kaki menyerupai manusia (android) ataupun hewan, robot beroda—*Wheeled Mobile Robot* (WMR) (Wikipedia, Mobile Robot, 2006).

Robot mobil telah menjadi sarana yang sering digunakan oleh pihak militer maupun pihak sipil untuk melakukan pengintaian, penjelajahan, dan pengawasan ke tempat-tempat yang berbahaya bagi manusia. Kemampuannya yang dapat dikontrolkan dari jauh atau bahkan dirancang agar dapat bergerak sendiri sesuai dengan lintasan tertentu akan sangat menguntungkan bagi manusia. Beberapa keuntungan yang dapat diperoleh antara lain meminimalisasi resiko, meminimalisasi objektif, dan lain sebagainya (Hartini, 2011). [1]

Wheeled Mobile Robot (WMR) atau robot beroda adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik lain. Mobile robot adalah tipe robot yang paling populer dalam dunia

penelitian robot. Sebutan ini biasa digunakan sebagai kata kunci utama untuk mencari rujukan atau referensi yang berkaitan dengan robotik di internet. Publikasi dengan judul yang berkaitan mobile robot sering menjadi daya tarik, tidak hanya bagi kalangan peneliti, tapi juga bagi kalangan awam. Dari segi manfaat, penelitian tentang berbagai tipe mobile robot diharapkan dapat membantu manusia dalam melakukan otomasi dalam transportasi, platform bergerak untuk robot industri, eksplorasi tanpa awak dan masih banyak lagi. Contoh dari *Wheeled Mobile Robot* dapat dilihat pada gambar 2.1



Gambar 2. 2 *Mobile Robot*

2.3. Pengolahan Citra (Image Processing)

Pengolahan citra adalah salah satu teknologi yang dapat memanipulasi dan mencari informasi-informasi penting pada suatu obyek dua dimensi yang nantinya akan diproses lebih lanjut, supaya dapat bermanfaat untuk bidang tertentu sesuai dengan apa yang diinginkan. Manfaat *Image Processing* dapat digunakan pada dunia industri dan militer. Selain dapat membantu meringankan pekerjaan manusia, *Image Processing* juga menghasilkan akurasi yang tepat. Beberapa metode yang digunakan pada Tugas Akhir ini adalah sebagai berikut :

2.2.1. Deteksi Citra (Image Detection)

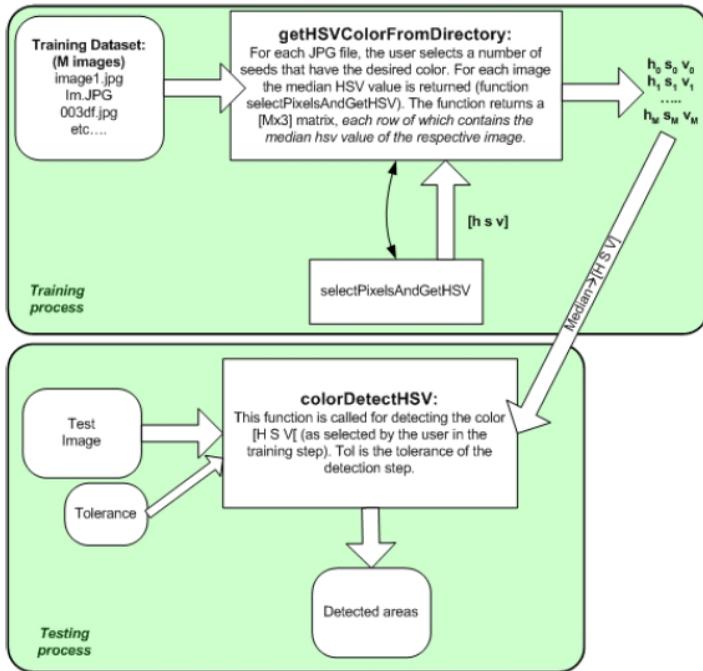
Dalam proses pengenalan obyek atau deteksi obyek diperlukan suatu pemisahan bagian atau segmen tertentu dalam citra yang akurat, proses pemisahan tersebut dikenal sebagai proses segmentasi. Proses

pengenalan segmen merupakan salah satu kunci dalam mendapatkan suatu hasil pengenalan atau deteksi yang akurat. Segmentasi membagi suatu citra menjadi beberapa bagian yang lebih sederhana sehingga dapat dilakukan analisa lebih lanjut. Segmentasi warna merupakan pemisahan segmen dalam suatu citra berdasarkan warna yang terkandung dalam citra. Dalam komputer vision telah dilakukan berbagai macam metode untuk melakukan segmentasi warna seperti metode clustering dan metode indeks.

Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra sesuai fitur yang diinginkan. Segmentasi citra menggunakan metode HSV menggunakan dasar seleksi warna pada model warna HSV dengan toleransi tertentu.

Pada metode segmentasi dengan deteksi warna HSV seperti pada gambar 2.2, dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Citra digital menggunakan model warna RGB sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna RGB ke HSV. Untuk membentuk segmen sesuai dengan warna yang diinginkan maka ditentukan nilai toleransi pada setiap dimensi warna HSV, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses adaptive threshold. Hasil dari proses threshold tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan. Secara garis besar, gambaran proses segmentasi menurut Giannakopoulos (2008).[2]

- Tentukan citra RGB yang menjadi obyek deteksi, nilai warna HSV yang menjadi acuan (hasil proses pelatihan data) dan nilai toleransi HSV yang digunakan.
- Transpose citra RGB ke HSV
- Lakukan filter warna pada citra berdasarkan nilai acuan (T) dan nilai toleransi (tol). Dengan x sebagai warna HSV pada piksel yang ada maka warna yang tidak termasuk dalam rentang $T - tol < x < T + tol$ diberi warna hitam.
- Transpose kembali citra ke RGB, tampilkan hasil filter.



Gambar 2. 3 Skema Deteksi Warna HSV

Citra dari target dapat dipengaruhi oleh sumber cahaya. Disini digunakan metode untuk mengubah RGB dan CYM ke dalam HSV yang dapat secara signifikan mampu mengurangi dampak cahaya. Model HSV terdiri dari tiga komponen penyusun yaitu: hue (H), saturation (S), and value (V). H adalah sudut dari 0 derajat hingga 360 derajat. Saturation menunjukkan kisaran abu-abu di ruang warna. Hal ini berkisar dari 0 hingga 100%. Kadang-kadang nilai dihitung dari 0 hingga 1. Value adalah kecerahan warna dan bervariasi dengan saturasi warna. Hal ini berkisar antara 0 hingga 100%. Nilai RGB ditransfer ke HSV dengan mengikuti persamaan sebagai berikut:

$$H = \cos^{-1} \frac{(R-G) + (R-B)}{\sqrt{(R-B)^2 + (R-B)(G-B)}}, B \leq G$$

$$H = 2\pi - \cos^{-1} \frac{(R-G) + (R-B)}{\sqrt{(R-B)^2 + (R-B)(G-B)}}, B > G \quad (1)$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R + B + G)} \quad (2)$$

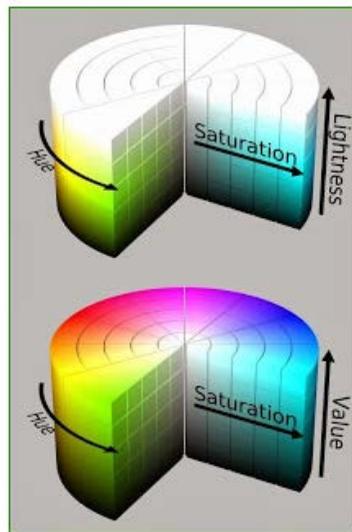
$$V = \frac{\max(R, G, B)}{255} \quad (3)$$

Citra biner adalah mengubah kecerahan gambar menjadi 0 atau 255, 0 mewakili hitam dan 255 mewakili putih. Jadi gambar asli dapat disederhanakan menjadi latar belakang dan depan. Metode ini dapat mengurangi jumlah pengolahan data citra. Dalam gambar biner, bagaimana menemukan threshold yang tepat sangat penting. Ambang batas akan mempengaruhi citra yang diambil pada objek pelacakan. Citra biner adalah gambar dalam pixel per titik dibandingkan dengan nilai ambang batas. Jika nilai pixel lebih kecil dari ambang batas, itu diatur menjadi 0.[3][4]

HSL adalah kepanjangan dari (hue, saturation, and lightness), kadang disebut HSL (hue, saturation, and luminosity/luminance). HSV (hue, saturation, and value), kadang disebut HSB (hue, saturation, and brightness). Ada bentuk lainnya yaitu HSI (h,s,intensity) tapi biasanya digunakan pada komputer vision. Perbedaan antara HSL dan HSV terletak di Lightness dan Value, dimana Lightness akan semakin memperterang tiap hue (warna) hingga ke tingkat warna paling terang (putih), sedangkan Value hanya sampai mempercerah tiap hue (warna) saja dari posisi awal paling bawah berwarna gelap (hitam). Perbedaan HSV dan HSL seperti pada gambar 2.3. Berdasarkan pada persepsi dasar manusia, HSL model tersusun atas tiga karakteristik warna dasar :

- Hue adalah warna yang direfleksikan atau pun ditransmisikan sebuah objek. Nilainya diukur dari lokasi pada roda standar warna, yang diekspresikan dengan nilai derajat sudut di antara 0° dan 360°. Dalam penggunaannya, hue mengidentifikasi nama dari sebuah warna seperti merah, orange (jingga), atau hijau.

- Lightness adalah sebuah ukuran relative skala pencahayaan (lightness) atau kegelapan (darkness) dari sebuah warna, umumnya diukur sebagai persentase dari 0% (black) sampai dengan 100% (white).
- Saturation, sering dikenal dengan chroma, yaitu ukuran atau kemurnian sebuah warna, Saturation merepresentasikan ukuran (kuantitas) dari proporsi keabuan pada hue, ukurannya dalam bentuk persentase dari 0% (gray) sampai dengan 100% (fully saturated). Pada roda standar warna, nilai saturation dari pusat roda (lingkaran) menuju tepian roda akan semakin bertambah.



Gambar 2. 4 Perbedaan HSV dengan HSL

2.2.2. Smoothing

Smoothing (melembutkan), disebut juga *blurring* (mengaburkan), merupakan sebuah proses pencitraan yang sering digunakan untuk mengurangi *noise* kamera. Terdapat banyak metode untuk melakukan smoothing namun yang sering digunakan adalah Gaussian *Blur*. [5]

Filter Gaussian merupakan kelas filter smoothing linear dengan bobot yang dipilih sesuai dengan fungsi Gaussian. Fungsi Gaussian dalam 1 dimensi adalah :

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (2.1)$$

dimana parameter penyebaran Gaussian σ menentukan lebar Gaussian.

Untuk pengolahan gambar 2-dimensi:

$$g[i, j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (2.2)$$

2.2.3. MeanShift

Algoritma mean shift, menggunakan prosedur perulangan sederhana, dari nilai data dan nilai rata-rata, dalam nilai data pixel tetangga. Data warna, tekstur dan gradient merupakan karakteristik distribusi static dari sebuah objek. Pengembangan dari algoritma mean shift, adalah algoritma continuously adaptive mean shift, bekerja dengan cara mencari titik pusat dan ukuran dari objek berwarna dalam suatu gambar dalam sebuah frame.

Algoritma mean shift, bergantung kepada level bawah pendeteksian. Background objek yang terdapat noise, mengakibatkan pembacaan terhadap objek tidak dapat dilakukan dengan baik. Tanpa mendefinisikan objek bergerak, pembagian dan mekanisme prediksi dan perbaikan. Algoritma ini, tidak dapat membedakan antara objek dengan yang bukan objek. Langkah – langkah algoritma *Mean Shift* adalah :

- Langkah 1 : Tentukan ukuran search window .
- Langkah 2 : Tentukan lokasi awal search window.
- Langkah 3 : Hitung daerah mean dalam search window.
- Langkah 4 : Posisikan search window ke tengah daerah mean seperti dihitung pada step (3).
- Langkah 5 : Ulangi langkah (3) dan (4) hingga konvergen (atau hingga pergeseran daerah mean kurang dari threshold / batas yang ditentukan). [6]

2.2.4. CamShift

CamShift merupakan singkatan dari *Continuously Adaptive Mean-Shift*. Algoritma CamShift merupakan pengembangan dari algoritma dasar Mean-Shift yang dilakukan secara berulang untuk dapat melacak pergerakan dari obyek. Algoritma Mean-Shift beroperasi berdasarkan distribusi probabilitas dari citra. Untuk melacak citra berwarna pada video sequence, maka citra berwarna tersebut harus direpresentasikan dalam bentuk distribusi probabilitas citra dengan menggunakan distribusi histogram dari citra tersebut. Distribusi warna dari video sequence selalu berubah tiap waktu, maka algoritma Mean-Shift harus dikembangkan supaya dapat digunakan untuk melacak obyek. Perkembangan dari algoritma ini dinamakan algoritma CamShift.

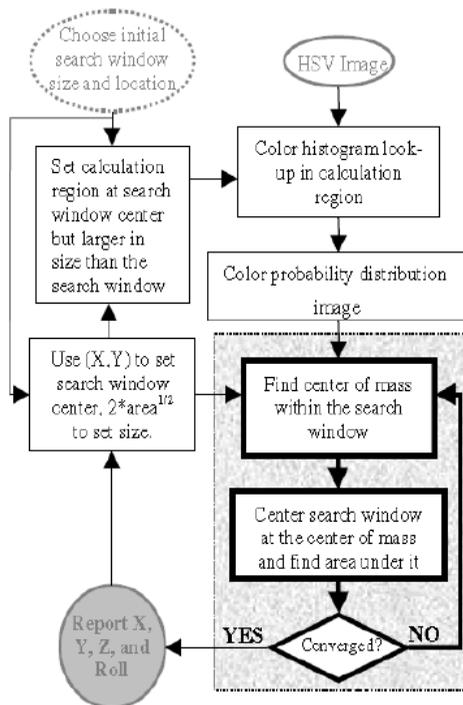
Tahap pertama dari algoritma CamShift adalah dengan melakukan penghitungan terhadap nilai hue dari obyek sampel. Nilai histogram hue sebagai hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai lookup table.

Tahap kedua, dilakukan pemilihan lokasi awal dari Mean-Shift 2D search window. Kemudian dilakukan penghitungan nilai histogram hue yang menjadi nilai probabilitas untuk tiap-tiap pixel pada citra.

Tahap ketiga, dijalankan algoritma Mean-Shift untuk mencari pusat dan besar search window yang baru. Lokasi tengah dan besar daerah yang dihasilkan disimpan. Lalu dilakukan looping ke tahap dua dimana lokasi awal dan search window menggunakan hasil algoritma Mean-Shift pada tahap tiga.

Alur dari algoritma CamShift dapat dilihat pada Gambar 2.4 Tahap pertama adalah melakukan penghitungan terhadap nilai hue dari objek target. Nilai histogram hue hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai lookup table (LUT) distribusi probabilitas citra. Tahap kedua adalah penginisiasian ukuran search window awal yaitu seukuran dengan frame pertama yang datang dari kamera. Tahap kedua atau preprocessing adalah pengekstraksian fitur dari setiap frame yang diterima. Tahap ketiga merupakan inti dari algoritma CAMShift. Tidak sama dengan algoritma MeanShift yang didesain untuk gambar statis, algoritma CAMShift didesain untuk

distribusi gambar yang bersifat dinamis. Hal ini terjadi karena objek yang dideteksi sebagai video bergerak yang akan menyebabkan lokasi dan ukuran search window berubah terhadap waktu. Oleh karena itu algoritma CAMShift akan selalu mengeset ukuran search window pada setiap pengolahan frame dengan kata lain search window akan lebih adaptif. [7]



Gambar 2. 5 Alur Algoritma *CamShift*

Tahap algoritma *CamShift* dapat dijabarkan sebagai berikut:

- Langkah 1 : Tentukan ukuran awal search window.
- Langkah 2 : Tentukan lokasi awal dari search window.

- Langkah 3 : Tentukan daerah kalkulasi (calculation region) pada bagian tengah search window dengan ukuran lebih besar dari search window.
- Langkah 4 : Frame citra video dikonversi ke dalam sistem warna HSV(Hue, Saturation, Value), kemudian langkah selanjutnya adalah membuat histogram dari citra untuk mengetahui distribusi probabilitas warna.
- Langkah 5 : Lakukan algoritma mean shift seperti di atas (satu atau banyak iterasi) dengan input berupa ukuran dan lokasi search window serta citra distribusi probabilitas warna dan simpan zeroth moment.
- Langkah 6 : Set nilai x, y, z yang diperoleh dari langkah (5).
- Langkah 7 : Nilai x dan y dipakai untuk menentukan titik tengah search window, sedangkan $(2 \cdot \text{area} / 2)$ untuk menentukan ukuran search window.
- Langkah 8 : Ulangi langkah (3) untuk setiap pergantian frame citra video. [8]

2.4. *OpenCV*

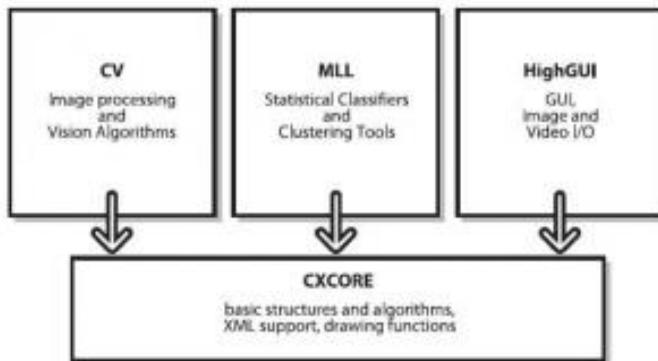
Open CV (Open Computer Vision) adalah sebuah API (Application Programming Interface) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision. OpenCV pertama kali diluncurkan secara resmi pada tahun 1999 oleh Inter Research sebagai lanjutan dari bagian proyek bertajuk aplikasi intensif berbasis CPU, real-time ray tracing dan tembok penampil 3D. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi di Intel Russia, dan juga Tim Pusataka Performansi Intel. Pada awalnya, tujuan utama dari proyek OpenCV ini dideskripsikan sebagai berikut,

- Penelitian penginderaan citra lanjutan tidak hanya melalui kode program terbuka, tetapi juga kode yang telah dioptimasi untuk infrastruktur penginderaan citra.
- Menyebarkan ilmu penginderaan citra dengan menyediakan infrastruktur bersama di mana para pengembang dapat menggunakannya secara bersama-sama, sehingga kode akan tampak lebih mudah dibaca dan ditransfer.

- Membuat aplikasi komersial berbasis penginderaan citra, di mana kode yang telah dioptimasi tersedia secara bebas dengan lisensi yang tersedia secara bebas yang tidak mensyaratkan program itu harus terbuka atau gratis.

Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dll.

OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video. OpenCV juga menyediakan interface ke Integrated Performance Primitives (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi Vision anda jika menggunakan prosesor Intel.



Gambar 2. 6 Struktur OpenCV

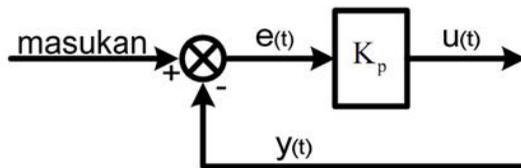
OpenCV sendiri terdiri dari 5 library, yaitu :

- CV : untuk algoritma Image processing dan Vision.

- ML : untuk machine learning library
- Highgui : untuk GUI, Image dan Video I/O.
- CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.

2.5. Kontrol Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya) [Sharon, 1992, 19]. Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.



Gambar 2. 7 Diagram Blok Kontroler Proporsional

Seperti yang terlihat pada gambar diagram 2.6, kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal error), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara input kontroler $u(t)$ dengan sinyal error $e(t)$ terlihat pada persamaan berikut

$$u(t) = K_p e(t) \quad (2.3)$$

Ciri-ciri kontroler proporsional harus diperhatikan ketika kontroler tersebut diterapkan pada suatu sistem. berikut ini adalah ciri-ciri dari kontrol proporsional:

1. Kalau nilai K_p kecil, kontroler proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.

2. Kalau nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan mantabnya.
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosilasi [Pakpahan, 1988, 193].

2.6. Single Board Computer

Single Board Computer adalah komputer papan tunggal yang lengkap dengan mikroprosesor, memori, input/output (I/O) dan fitur lain yang dibutuhkan dari sebuah komputer fungsional. SBC menggunakan memori eksternal seperti kartu SD dan kartu mikro SD.

Raspberry Pi adalah komputer papan tunggal (Single Board Computer/SBC) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan seperti spreadsheet, game, bahkan sebagai media player. Raspberry Pi dikembangkan oleh Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris.

Raspberry Pi memiliki dua model yaitu model A dan model B. Perbedaan model A dan B terletak pada memory yang digunakan. Model A menggunakan 256 MB dan model B menggunakan 512 MB. Selain itu terdapat Raspberry Pi 2 yang memiliki memory yang lebih yaitu 1GB. Pada model B terdapat ethernet port yang tidak terdapat pada model A. Gambar 2.7 merupakan contoh dari Raspberry Pi tipe B+. Berikut adalah spesifikasi dari Raspberry Pi:

- Specifications :
 - Chip Broadcom BCM2835 SoC
 - Core architecture ARM11 CPU 700 MHz Low Power ARM1176JZFS Applications Processor GPU Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
 - Memory 512MB SDRAM
 - Operating System Boots from Micro SD card, running a version of the Linux operating system
 - Dimensions 85 x 56 x 17mm
 - Power Micro USB socket 5V, 2A
- Connectors:
 - Ethernet 10/100 BaseT Ethernet socket
 - Video Output HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
 - Audio Output 3.5mm jack, HDMI
 - USB 4 x USB 2.0 Connector
 - GPIO Connector 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
 - Camera Connector 15-pin MIPI Camera Serial Interface (CSI-2)
 - JTAG Not populated
 - Display Connector Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
 - Memory Card Slot SDIO [9]



Gambar 2. 8 *Raspberry Pi*

2.7. Webcam

Webcam adalah kamera atau video digital yang dihubungkan dengan komputer dan dapat dikoneksikan dengan sebuah web.

Kegunaan webcam:

- Live streaming
- Live audio
- dapat digunakan dengan software lain

Webcam terhubung dengan komputer dengan media kabel USB atau pararell port. Agar webcam dapat digunakan oleh komputer, kita membutuhkan suatu perangkat lunak. Perangkat lunak itu akan menangkap tiap frame pada waktu tertentu, biasa disebut frame rate. Data yang disimpan biasanya berformat avi (video), tapi dapat diubah dalam bentuk JPG (image) atau yang lainnya agar tidak terlalu banyak memakan tempat pada media penyimpanan.



Gambar 2. 9 Kamera *Webcam*

2.8. Mikrokontroler

Mikrokontroler sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang kecil serta dapat diproduksi secara masal (dengan jumlah banyak) membuat harga menjadi murah dibandingkan mikroprosesor. Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi, mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja.

Secara teknis, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagian itu yaitu *RISC* dan *CISC*. *RISC* merupakan kependekan dari *Reduced Instruction Set Computer*. Instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak. Sebaliknya, *CISC* kependekan dari *Complex Instruction Set Computer*. Instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya.

Adapun kelebihan dari mikrokontroler adalah sebagai berikut:

- Penggerak pada mikrokontroler menggunakan bahasa pemrograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem.
- Mikrokontroler tersusun dalam satu chip dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan control sistem sehingga mikrokontroler dapat di katakana computer mini.
- Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan computer.
- Tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.

Salah satu contoh pengenalan mikrokontroler adalah AVR yang merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur *RISC (Reduced Instruction Set Computer)*. AVR mempunyai 32 *register general-purpose, timer / counter* fleksibel dengan *mode compare, interrupt internal* dan *eksternal, serial UART, programmable Watchdog Timer*, dan *mode power saving, ADC dan PWM internal*. AVR juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk deprogram ulang dalam sistem menggunakan hubungan *serial SPI*.

Mikrokontroler AVR menjadi chip utama pada alat pengembangan prototype Arduino. Arduino dikatakan sebagai sebuah platform dari *physical computing* yang bersifat *open source*. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan *Integrated Development*

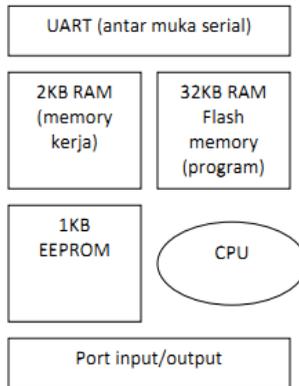
Environment (IDE) yang canggih. IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory mikrokontroler. Ada banyak projek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi. [10]

Arduino dikembangkan oleh sebuah tim yang beranggotakan orang-orang dari berbagai belahan dunia. Anggota inti dari tim ini adalah:

- Massimo Banzi Milano, Italy
- David Cuartielles Malmoe, Sweden
- Tom Igoe New York, US
- Gianluca Martino Torino, Italy
- David A. Mellis Boston, MA, USA

Komponen utama di dalam papan Arduino adalah sebuah mikrokontroler 8 bit dengan merk ATmega yang dibuat oleh perusahaan Atmel Corporation. Berbagai papan Arduino menggunakan tipe ATmega yang berbeda-beda tergantung dari spesifikasinya, sebagai contoh Arduino Uno menggunakan ATmega328 seperti pada gambar 2.9 sedangkan Arduino Mega 2560 yang lebih canggih menggunakan ATmega2560.[11]

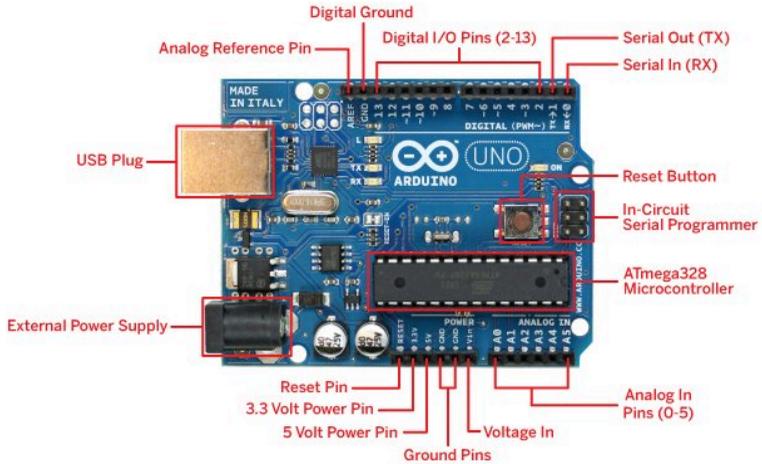
Arduino Uno adalah board mikrokontroler AVR yang menggunakan Atmega328 dari Atmel. Diagram blok dari Atmega 328 seperti pada gambar 2.8. Arduino Uno memiliki 14 pin digital input output (6 dapat digunakan sebagai output PWM), 6 pin input analog, menggunakan kristal 16 MHz, koneksi USB, power supply, ICSP, dan tombol reset. Pin pada Arduino Uno dapat dilihat pada gambar 2.10.



Gambar 2. 10 Diagram Blok ATmega 328

Arduino memiliki kelebihan yaitu bersifat *open source*. Selain itu arduino memiliki bahasa pemrograman tersendiri yang berupa bahasa C. Didalam board arduino terdapat loader yang berupa USB sehingga dapat secara langsung diprogram. Arduino dapat langsung diprogram menggunakan IDE Arduino yang menggunakan Java. IDE Arduino terdiri dari :

- Editor program, sebuah window yang memungkinkan pengguna menulis dan mengeditprogram dalam bahasa Processing.
- Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
- Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.



Gambar 2. 11 Arduino Uno

ATmega168/328 Pin Mapping

Arduino function	ATmega Pin	ATmega Pin	Arduino function	
reset	(PCINT14/RESET) PC6	1	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	GND	GND
GND	GND	8	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	PB3 (MOSI/OC2A/PCINT3)	digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/CP1) PB0	14	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

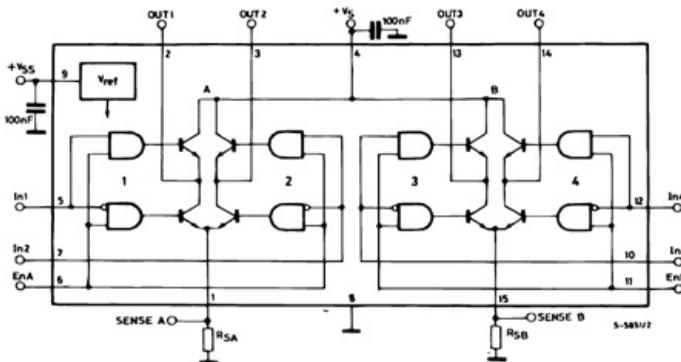
Gambar 2. 12 Pin Mapping Arduino Uno

2.9. Ardumoto

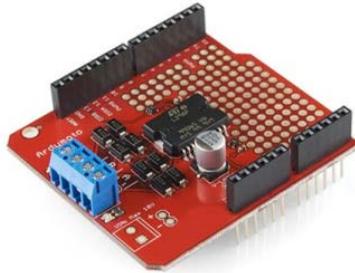
Driver motor ini menggunakan IC L298P. L298 adalah jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan motor DC ataupun Motor stepper. Mampu mengeluarkan output tegangan untuk Motor dc dan motor stepper sebesar 50 volt. IC L298 terdiri dari transistor-transistor logik (TTL) seperti pada gambar 2.11 dengan gerbang nand yang memudahkan dalam menentukan arah putaran suatu motor dc dan motor stepper. Dapat mengendalikan 2 untuk motor dc namun pada hanya dapat mengendalikan 1 motor stepper.

Ardumoto adalah shield arduino driver motor H-Bridge. Dapat dilihat pada gambar 2.12, shield ini berfungsi untuk mengontrol kecepatan motor dengan menggunakan PWM. Pin 3 dan 11 untuk keluaran besaran PWM dari setiap motor sedangkan untuk pin 12 dan 13 untuk menentukan arah dari masing-masing motor.

Ardumoto menggunakan IC L298P yang mampu mengontrol 2 motor DC hingga 2 Ampere dan dapat bergerak secara bidireksional, artinya motor dapat diatur bergerak secara clockwise (CW) atau counter-clockwise (CCW).



Gambar 2. 13 Circuit Diagram IC L298



Gambar 2. 14 Driver Motor Shield Ardumoto

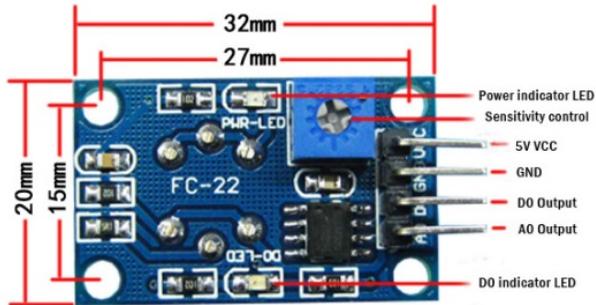
2.10. Sensor Gas MQ-3

Sensor MQ3 memiliki karakteristik sebagai berikut :

- Catu daya heater : 5V AC/DC
- Catu daya rangkaian : 5VDC
- Range pengukuran : 0,05mg/L – 10mg/L
- Cocok untuk aplikasi pengetes kadar alkohol di udara
- Output : analog (perubahan tegangan) dengan tambahan Rload

Gas Sensor MQ-3 merupakan Sensor alkohol yang cocok untuk mendeteksi kadar alkohol secara langsung, misal pada nafas kita. Sensor alkohol MQ-3 memiliki sensitifitas tinggi dan waktu respon yang cepat. Sensor alkohol MQ-3 Rangkaian driver untuk Sensor alkohol MQ-3 ini sangat sederhana, hanya perlu 1 buah variable resistor. Dapat dilihat pada gambar 2.13, pada sensor gas MQ-3 terdapat variabel resistor untuk mengubah sensitivitas dari sensor tersebut. Output dari Sensor alkohol MQ-3 ini berupa tegangan analog yang sebanding dengan kadar alkohol yang diterima. Antarmuka yang diperlukan cukup sederhana, bisa menggunakan ADC yang dapat merespon tegangan 0 volt – 3,3 volt saja. Nilai Resistor yang dipasang harus dibedakan untuk berbagai jenis konsentrasi gas. Jadi perlu dikalibrasi untuk 0,4mg/L (sekitar 200ppm)

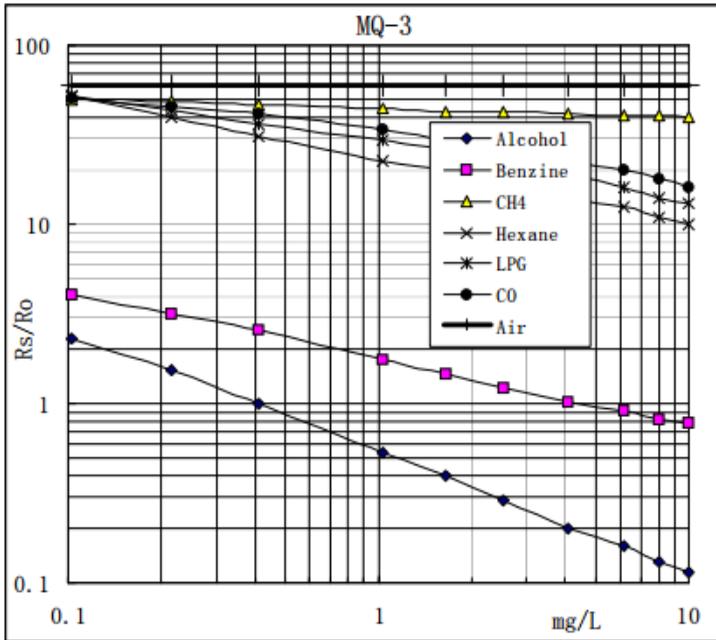
konsentrasi alkohol di udara dan resistansi pada output sekitar 200K (100K sampai 470K). Pada gambar 2.14 adalah sensor gas MQ-03 tampak dari depan. sedangkan pada gambar 2.15 adalah grafik sensitivitas dari sensor tersebut. Dapat dilihat dari gambar 2.15 bahwa sensor gas MQ-03 sensitif terhadap gas benzine, alkohol, CH₄, LPG, dan Hexane.



Gambar 2. 15 Sensor Gas MQ3 Tampak Belakang



Gambar 2. 16 Sensor Gas MQ3 Tampak Depan

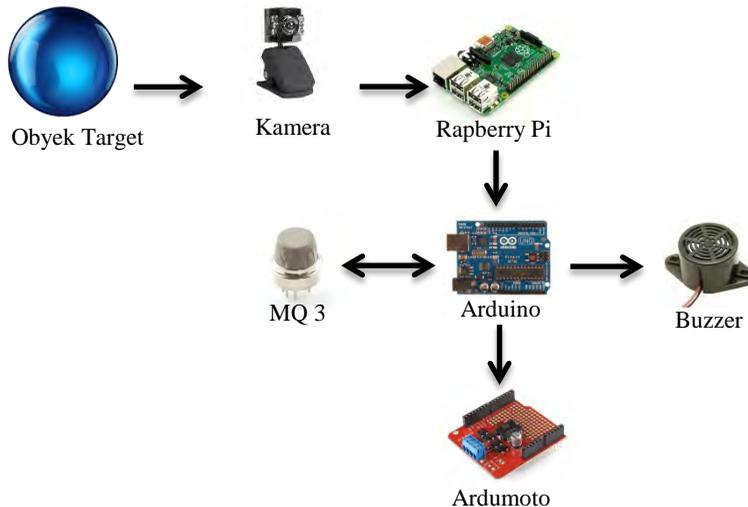


Gambar 2. 17 Sensitivitas Sensor Gas MQ 3

Halaman ini sengaja dikosongkan

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan terkait perancangan sistem secara keseluruhan, hal ini mencakup perancangan *hardware* dan perancangan *software*. Raspberry Pi akan melakukan *Image Processing* dari kamera untuk menentukan obyek yang akan dituju. Data posisi dari target akan dikirimkan dari Raspberry Pi menuju Arduino untuk dapat mengontrol gerakan dari motor. Arduino akan mengontrol driver motor Ardumoto dan sensor gas MQ 3. Alarm akan berbunyi jika sensor gas mendeteksi suatu gas yang berpotensi berbahaya. Ilustrasi cara kerja sistem dapat dilihat pada gambar 3.1



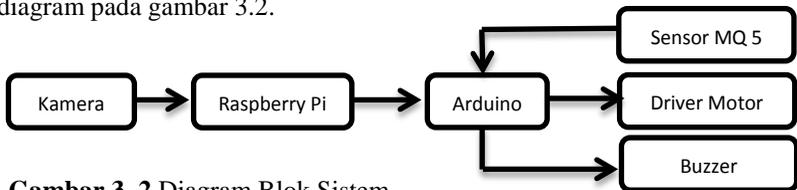
Gambar 3. 1 Cara Kerja Sistem

Berikut ini merupakan cara kerja keseluruhan sistem :

- Kamera akan menangkap data gambar dari target yang akan dituju.
- Penetapan ROI untuk algoritma CamShift.
- Muncul histogram nilai Hue benda.

- Dilakukan thresholding antara nilai Saturation dan Value untuk hasil yang maksimal tanpa noise
- Perhitungan data posisi dan ukuran menjadi gerakan motor.
- Pembacaan sensor gas untuk pendeteksian gas alkohol.

Cara kerja dari sistem secara umum digambarkan pada blok diagram pada gambar 3.2.



Gambar 3. 2 Diagram Blok Sistem

3.1 Perangkat Keras

Perangkat keras yang digunakan pada Tugas Akhir ini adalah pembuatan base mekanik robot, kamera, Raspberry Pi, mikrokontroler Arduino Uno, driver motor Ardumoto, motor DC, LCD, dan sensor gas MQ-02. Pada robot ini menggunakan motor DC 12V 1A yang mempunyai torsi 9,8 kg dan 38 rpm sehingga mampu menahan beban yang berat tetapi memiliki kelemahan kecepatan yang relatif pelan sehingga membutuhkan waktu yang lama untuk dapat sampai ke tujuan.

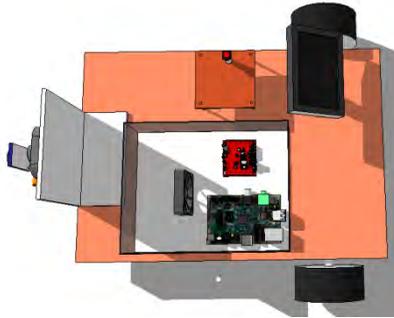
Daya yang digunakan pada robot ini ada 2 yaitu baterai Li-Po 12V 1000mAh untuk menggerakkan motor dan Power Bank 11000mAh untuk Raspberry Pi.

3.1.1. Mekanik Robot

Pada Tugas Akhir ini menggunakan mekanik dengan rangka yang terbuat dari alumunium. Hal ini disebabkan karena alumunium memiliki massa jenis yang ringan tetapi kuat untuk menopang beban. Robot menggunakan dua roda penggerak dan dua roda bebas untuk mendukung manuver robot. Masing-masing roda penggerak menggunakan motor dc 12 volt 38 rpm. Pemilihan dari motor ini dikarenakan motor ini mampu menahan beban yang berat dikarenakan memiliki torsi yang tinggi walaupun kecepatan yang dihasilkan rendah. Berikut desain mekanik robot. Desain dibuat dengan dalam satuan milimeter.



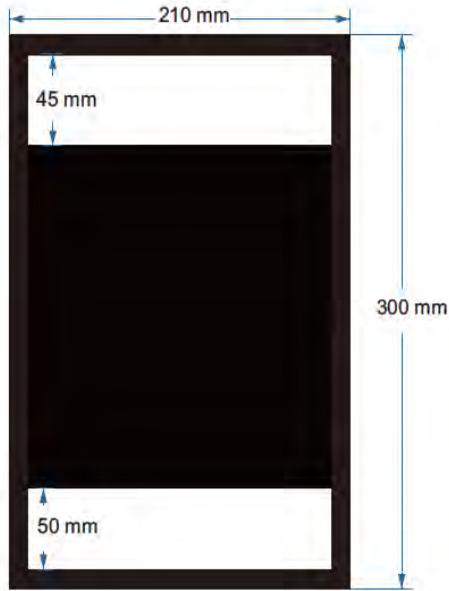
Gambar 3. 3 Desain Robot Tampak Miring



Gambar 3. 4 Desain Robot Tampak Atas



Gambar 3. 5 Desain Robot Tampak Depan



Gambar 3. 6 Desain Base Robot

3.1.2. Kamera

Kamera disini digunakan untuk menangkap gambar dari obyek yang dituju. Kamera yang digunakan pengaturan fokusnya dapat diubah dan tidak memiliki fitur auto fokus. Kamera ini terhubung dengan Raspberry Pi sebagai *processing unit* untuk mengolah data gambar melalui kabel USB. Kamera yang digunakan adalah kamera webcam standar dengan spesifikasi minimum :

- Resolution : 1920 x 1080 pixels
- Frame Rate : 30 fps
- Computer Interface : USB 2.0

3.1.3. Single Board Computer

Pada Tugas Akhir ini menggunakan *Single Board Computer* Raspberry Pi b+ dengan spesifikasi sebagai berikut :

- Soc Broadcom BCM2835
- CPU : 700 MHz ARM1176JZF-S core
- GPU : Broadcom VideoCore IV, OpenGL ES 2.0, 1080P30h.264/MPEG-4 AVC high-profile decoder
- Memory (SDRAM) : 512 MB (Shared with GPU)
- Operating System : Raspbian.
- Size : 85.60 mm x 53.98 mm

3.1.4. *Interfacing Driver Motor Shield dengan Arduino Uno*

Sebagaimana dijelaskan pada bab 2, pada Tugas Akhir ini menggunakan driver motor Ardumoto yang ditumpukkan diatas Arduino Uno sehingga driver Ardumoto mampu mengendalikan putaran dari masing-maing motor kiri dan motor kanan berdasarkan data yang diolah di Arduino. Data yang masuk ke Arduino akan digunakan untuk mengontrol kecepatan motor kiri dan motor kanan dengan menggunakan kontrol Proporsional sehingga kecepatan dari motor dapat diatur.



Gambar 3. 7 Driver Motor Ardumoto yang Dihubungkan Arduino

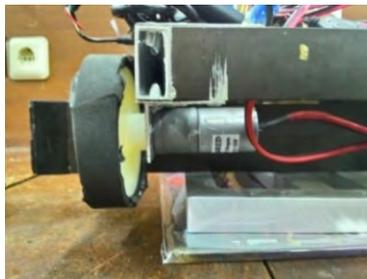
Seperti yang terlihat pada gambar 3.4, driver ini dihubungkan dengan Arduino dengan pin header. Pin ini terhubung dengan pin digital Arduino 3, 11, 12, dan 13. Pin 3 dan 11 untuk mengatur besarnya PWM untuk motor sedangkan pin 12 dan 13 digunakan untuk mengatur arah. Kabel positif dari motor DC dihubungkan dengan pin A1 dan B3 seperti pada gambar 3.5. Sedangkan untuk kabel negatif dihubungkan dengan A2 dan B4. Ini dapat dilakukan sebaliknya karena hanya mengubah polaritas saja.

Daya yang digunakan pada Ardumoto menggunakan baterai 12V 1300mAh sehingga Vin pada Ardumoto dan Arduino tidak diperbolehkan terhubung karena Arduino hanya membutuhkan sumber sebesar 5V.



Gambar 3. 8 Konfigurasi Pin Pada Driver Motor Ardumoto

Motor yang digunakan pada robot ini yaitu motor geared DC motor dengan kecepatan sebesar 38 RPM dan torsi 9.8 kg seperti gambar 3.6. Motor ini membutuhkan supply dari luar dikarenakan arus dan tegangan dari Arduino tidak mencukupi untuk men-supply motor DC. Motor DC yang digunakan membutuhkan supply sebesar 12 V dan arus sebesar 1 A. Berikut adalah contoh gambar motor DC yang digunakan.



Gambar 3. 9 Motor DC

3.1.5. Sensor Gas

Mikrokontroler memiliki kemampuan untuk menerima, mengolah, dan memberikan output signal digital. Sebagai contoh bila

mikrokontroler dengan catu daya 5 volt, mikrokontroler tersebut dapat mengenali tegangan sebesar 0 volt sebagai signal digital low, atau tegangan 5 volt sebagai signal digital high. Sedangkan untuk mengenali nilai tegangan antara 0 volt hingga 5 volt, diperlukan feature khusus, yakni Analog-to-Digital Converter. Analog-to-Digital Converter atau biasa disebut ADC, memungkinkan mikrokontroler untuk mengenali suatu nilai analog melalui suatu pendekatan digital. Saat ini banyak mikrokontroler, termasuk Arduino telah dilengkapi dengan feature ADC yang terintegrasi di dalamnya.

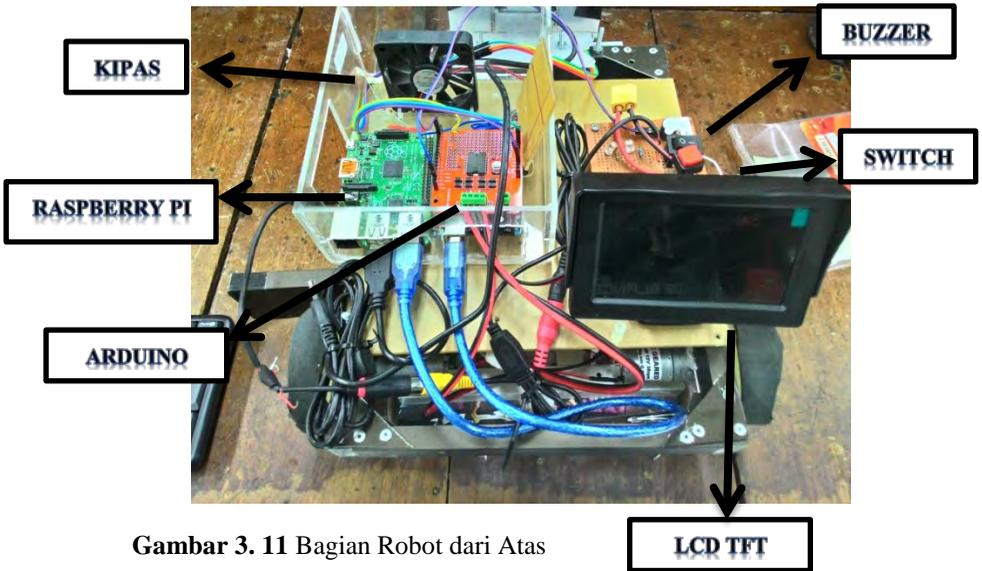
Tapi tidak semua pin Arduino dapat digunakan untuk mengolah signal analog. Pada board Arduino Uno, terdapat enam pin analog, yakni mulai dari A0 hingga A5. Huruf A pada awal nama pin Arduino menandakan pin tersebut dapat digunakan untuk mengolah signal analog. Seberapa tepat nilai signal analog yang dipetakan secara digital, ditentukan oleh seberapa besar resolusi ADC. Semakin besar resolusi ADC, maka semakin mendekati nilai analog dari signal tersebut. Untuk resolusi ADC pada board Arduino Uno ialah 10 bit, yang berarti mampu memetakan hingga 1024 diskrit analog level. Beberapa jenis mikrokontroler lain memiliki resolusi 8 bit, 256 diskrit analog level, bahkan ada yang memiliki resolusi 16 bit, 65536 diskrit analog level.

Pada tugas akhir ini digunakan sensor gas MQ-03 yang mampu mendeteksi jenis gas alkohol, propane, buthane, dan LPG seperti pada gambar 3.7. Keluaran dari sensor ini dapat berupa digital maupun analog. Pembacaan nilai ADC dari sensor ini akan mengaktifkan buzzer apabila pembacaan ADC lebih dari 1V. Sensor ini dihubungkan dengan Arduino pin analog A0 dan buzzer dihubungkan dengan pin digital 2 pada Arduino.



Gambar 3. 10 Sensor Gas MQ-03

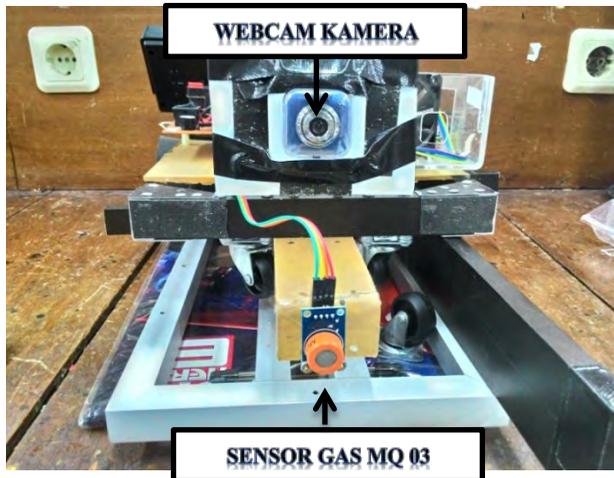
Bentuk keseluruhan robot sebagai berikut.



Gambar 3. 11 Bagian Robot dari Atas

Seperti yang terlihat pada gambar 3.8, pada bagian atas robot terdapat switch yang berfungsi untuk menyalakan LCD TFT dan mengaktifkan Arduimoto *Shield*. Dikarenakan Arduimoto *Shield* tidak mendapatkan daya dari Arduino maka Arduimoto *Shield* menggunakan daya dari luar hanya saja pin pada Arduimoto dan Arduino saling terhubung tetapi tidak pada pin Vin.

Pada bagian depan robot terdapat kamera untuk menangkap gambar dan sensor gas. Sensor gas digunakan di depan karena jarak untuk mendeteksi gas kurang lebih 3cm dari target. Bagian depan robot dapat dilihat pada gambar 3.9. Sensor gas diletakkan lebih maju daripada kamera dikarenakan sensor gas membutuhkan jarak yang pendek untuk dapat mendeteksi kebocoran gas. Selain itu kamera tidak dapat menangkap gambar apabila benda terlalu dekat dengan robot.



Gambar 3. 12 Bagian Depan Robot

3.2 Perancangan Software

Pada perancangan *software* terbagi menjadi dua, yaitu yang pertama bagian program untuk *Single Board Computer* dan mikrokontroller. Program untuk *Single Board Computer* digunakan *image processing* untuk mendeteksi dan mendapatkan data benda yang menjadi target yang dituju. Sedangkan program untuk mikrokontroller digunakan untuk mengendalikan gerakan robot agar dapat bergerak menuju sasaran yang telah ditentukan oleh *Single Board Computer* dan membaca nilai dari sensor gas.

3.2.1. Perancangan Perangkat Lunak untuk SBC

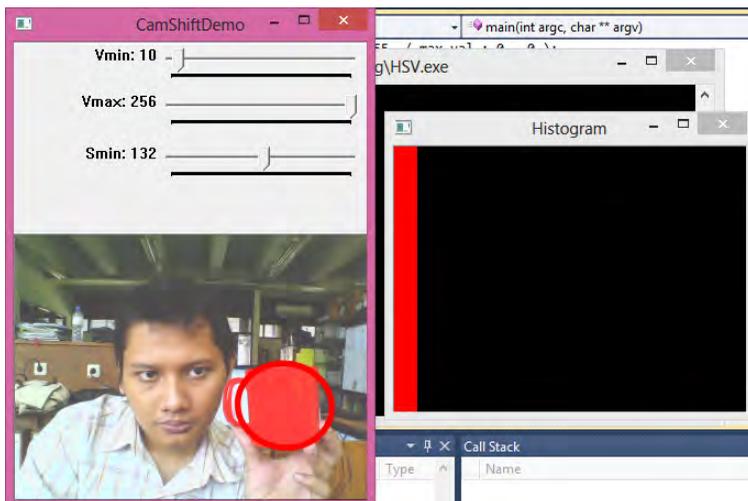
Seperti yang sudah dijelaskan pada sebelumnya, Perancangan program untuk *Single Board Computer* yang akan dirancang adalah *image processing* untuk mendapatkan warna dari target yang dituju dan software yang digunakan adalah *OpenCV*. Target yang akan dituju akan ditentukan menggunakan algoritma CamShift. Berikut adalah diagram alir deteksi benda :



Gambar 3. 13 Flowchart Deteksi Bola

Algoritma Camshift akan mengeluarkan nilai histogram nilai warna dari benda yang dituju. Pertama benda yang dituju ditentukan dulu ROI menggunakan pointer. Setelah itu nilai histogram warna dari target akan keluar dan akan dikalibrasi nilai Saturation dan Value untuk mendapatkan hasil yang maksimal. Setelah itu posisi dari benda akan keluar dan dikirim pada Arduino.

Berikut adalah contoh hasil deteksi benda menggunakan algoritma *CamShift* :



Gambar 3. 14 Deteksi Benda

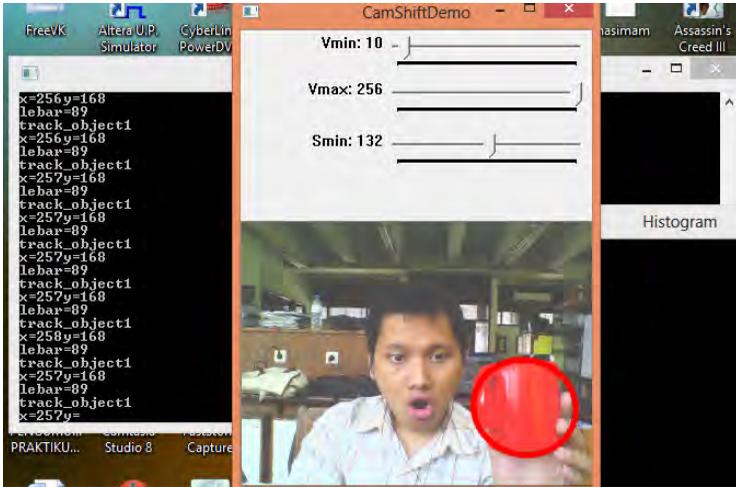
Pada gambar 3.11, setelah Raspberry Pi dapat mendeteksi benda maka akan muncul histogram nilai Hue dari benda yang dituju. Setelah itu menyesuaikan nilai dari Value dan Saturation. Hal ini diperlukan untuk menghilangkan noise pada gambar sehingga tidak terjadi kesalahan pendeteksian benda. Apabila benda yang ditentukan sudah sesuai dengan pendeteksian pada kamera maka posisi dari benda tersebut akan dikirimkan ke mikrokontroler Arduino Uno melalui komunikasi serial yang nantinya digunakan untuk mengontrol pergerakan robot. Kecepatan motor akan diatur berdasarkan besarnya error pada frame. Semakin besar error maka semakin besar pula

kecepatan yang dikeluarkan . Berikut adalah program untuk mengirim data posisi dari kamera

```
outputChars[0] = (x/100)+48;
outputChars[1] = ((x %100) / 10)+48;
outputChars[2] = (x % 10)+48;
WriteFile(hSerial, outputChars, 3 , &btsIO, NULL);

// posisi koordinat Y
// mundur
if (r > 100)
{
    b[0] = 'b';
    WriteFile(hSerial, b, 1, &btsIO, NULL);
}
// MAJU
if (r < 50)
{
    c[0] = 'f';
    WriteFile(hSerial, c, 1, &btsIO, NULL);
}
// STOP
if (r > 50 && r < 100 )
{
    d[0] = 's';
    WriteFile(hSerial, d, 1, &btsIO, NULL);
}
```

Nilai posisi ini nantinya akan dimasukkan pada mikrokontroler Arduino untuk dapat diproses menjadi PWM yang berguna untuk mengontrol kecepatan motor DC dan mengontrol arah gerakan robot. Nilai dari posisi obyek yang ditangkap oleh kamera akan dikirimkan menggunakan komunikasi serial dari Raspberry Pi menuju Arduino menggunakan kabel USB. Jadi power yang digunakan untuk menyalakan Arduino di suplai oleh Raspberry Pi. Berikut adalah gambar 3.12 hasil posisi benda pada Raspberry Pi yang akan dikirim ke mikrokontroler.



Gambar 3. 15 Nilai posisi benda

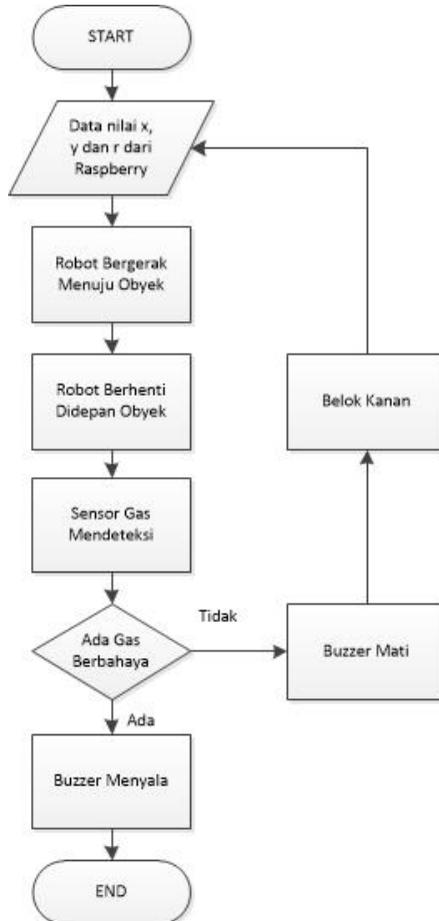
Tahap akhir setelah mendapatkan nilai posisi maka nilai ini akan dikirimkan ke mikrokontroler untuk dioalah menjadi gerakan robot menggunakan kontrol Proporsional.

3.2.2. Perancangan Perangkat Lunak Mikrokontroler

Mikrokontroler Arduino menggunakan bahasa pemrograman C dan dapat diprogram menggunakan Arduino IDE. Seperti yang dijelaskan sebelumnya, Arduino berfungsi untuk menerima data dari Raspberry Pi yang kemudian diolah untuk mengatur pergerakan dari robot. Nilai posisi dari obyek akan dimasukkan sebagai *error* untuk kontrol Proporsional. Kontrol proporsional akan memberikan keluaran seberapa besar PWM yang dibutuhkan untuk dapat mengurangi error. Semakin besar error atau semakin jauh obyek dari titik tengah maka semakin besar pula nilai PWM yang akan dikeluarkan oleh Arduino. Apabila Arduino tidak menerima data dari Raspberry Pi maka Arduino bertahan pada data sebelumnya yang dikirim oleh Raspberry Pi.

Arduino juga menerima data dari sensor gas yang nantinya akan jadi penanda apakah obyek tersebut mengeluarkan gas berbahaya atau tidak. Sensor gas yang digunakan seri MQ yaitu MQ-03 yang sensitif terhadap alkohol, buthanol, dan gas LPG. Sensor gas ini akan mengeluarkan nilai ADC dan besarnya nilai analog yang dikeluarkan

bergantung dari jarak gas tersebut dengan sensor. Semakin jauh gas yang terdeteksi maka nilai ADC yang terbaca pun akan semakin kecil dan bisa hampir tidak terbaca. Tetapi hal ini dapat di setting dengan trimpot yang ada pada sensor gas MQ-03. Berikut ini diagram alir dari gerakan robot :



Gambar 3. 16 Flowchart Gerakan Robot

Awalnya nilai x, y dan r yang diproses oleh Raspberry Pi akan menjadi input oleh Arduino dalam menentukan gerakan robot. Nilai dari posisi benda akan dikirimkan melalui USB dengan komunikasi serial. Komunikasi serial antara Arduino dengan Raspberry Pi biasanya membutuhkan Logic Level Converter dikarenakan digital pin serial Rx dan TX pada Raspberry Pi bekerja dengan 3,3V sedangkan pada Arduino 5V. Tetapi dikarenakan komunikasi serial dilakukan melalui USB maka tidak diperlukan Logic Level Converter. Berikut adalah fungsi program untuk membaca data dari Raspberry Pi.

```
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:

    if (i<=3)
    { if(i==0)
      {
        a[0]= Serial.read();
      }
      else if(i==1)
      {
        a[1]= Serial.read();
      }
      else if(i==2)
      {
        a[2]= Serial.read();
      }
      else if(i==3)
      {
        a[3]= Serial.read();
        i=-1;
      }
      i++;
    }
  }
  nilai = (((int)a[0]-48)*100 + ((int)a[1]-48)*10 + ((int)a[2]-48));
  nilai2 = ((nilai-160.00)/160.00);
}
```

Jadi nilai yang dikirimkan oleh Raspberry Pi akan ditampung terlebih dulu. Hal ini dikarenakan komunikasi serial hanya dapat

mengirimkan data bertipe char saja. Data yang telah ditampung kemudian di konversi menjadi integer agar dapat diolah menjadi nilai PWM untuk mengontrol gerakan robot. Kontrol gerakan pada robot menggunakan kontrol proporsional sehingga apabila posisi robot jauh dari titik tengah maka nilai proporsional yang dikeluarkan maka akan lebih besar sehingga kecepatan motor juga bertambah. Berikut adalah contoh program untuk menghitung besar nilai PWM.

```
void loop() // run over and over
{
    if (nilai2 < (-0.40))
    {
        arah1=false;
        arah2=true;
        kp=((nilai2*(-1.00))*10)+80;
    }
    else if (nilai2 > 0.40)
    {
        arah1=true;
        arah2=false;
        kp = ((nilai2*1.00)*10)+80;
    }
    else if (nilai2 > (-0.40)&& nilai2<(0.40))
    {
        if (a[3]=='f'){
            arah1=false;
            arah2=false;
            kp=90;
        }
        if (a[3]=='b'){
            arah1=true;
            arah2=true;
            kp=90;
        }
    }
}
```

Jadi ketika nilai input dari Raspberry Pi bernilai kurang dari 0 maka itu menandakan bahwa obyek yang terdeteksi berada di sebelah kiri dari frame yang diambil sehingga gerakan robot harus berbelok ke arah kiri

dan arah dari motor kanan harus maju sedangkan motor kiri mundur. Sedangkan apabila input Raspberry Pi bernilai dari 0, menandakan bahwa obyek yang terdeteksi berada di sebelah kanan dari frame yang diambil sehingga robot harus bergerak ke arah kanan dan arah motor kanan harus mundur dan motor kiri maju. Besarnya kecepatan pada tiap motor sama bergantung nilai dari PWM yang dihitung. Apabila sudah berada diantara batas kanan dan batas kiri maka robot akan maju apabila jari-jari lingkaran dirasa kecil. Hal ini menunjukkan bahwa jarak obyek yang dituju masih jauh. Apabila terlalu dekat maka robot akan mundur. Kecepatan maju dan mundur diatur sama untuk setiap motor.

Sensor gas akan bereaksi ketika robot berhenti di depan obyek yang dituju. Ketika robot berhenti sensor gas akan aktif dan akan mendeteksi apakah obyek tersebut mengandung gas yang berbahaya atau tidak. Apabila obyek tersebut mengandung gas berbahaya maka buzzer akan menyala dan apabila tidak maka buzzer akan mati. Sensor gas ini bereaksi terhadap beberapa zat seperti alkohol, butanol, propane, dan gas LPG. Nilai keluaran dari sensor ini ada dua yaitu dapat berupa digital dan analog. Perubahan nilai yang digunakan disini berupa analog sehingga dibutuhkan pembacaan nilai ADC.

Pada Arduino menggunakan sumber tegangan 5 volt. Berikut adalah cara mencari nilai ADC, dengan menerapkan persamaan yang ada.

- Board Arduino Uno memiliki resolusi 10 bit, dengan nilai terbesar 1023
- Tegangan sumber 5 volt dan tegangan terbaca ialah 2,12 volt
- Nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber
- Sehingga diperoleh nilai ADC sebesar 434

Dengan demikian diperoleh nilai ADC sebesar 434 dari tegangan terukur 2,12 volt. Untuk setiap unit ADC tersebut memiliki perbandingan tegangan sebesar 4,9 mV.

Selain dipengaruhi oleh besarnya nilai resolusi ADC, tepat tidaknya pengukuran nilai ADC juga dipengaruhi oleh clock speed ADC tersebut. Untuk board Arduino Uno sendiri clock speed ADC maximum yang disarankan ialah 200 kHz. Nilai clock speed 200 kHz tersebut berdasarkan spesifikasi internal DAC (Digital to Analog Converter) pada rangkaian pengubahnya. Meski demikian, penggunaan clock speed

pada 1 MHz mesalnya, tidak mengurangi kualitas resolusi ADC tersebut. Berikut adalah contoh program pembacaan nilai ADC pada sensor gas MQ-02.

```
digitalWrite(2,LOW);
val = analogRead(A0); // read the input pin
if (val > 205){
    digitalWrite(2,HIGH);
    delay(1000);
}
else {
    digitalWrite(2,LOW);
}
```

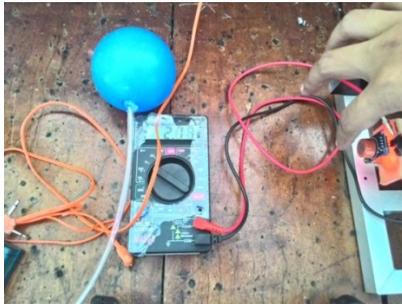
Pada pembacaan nilai ADC sensor gas pada Arduino, nilai masukan dari pin A0 akan dibandingkan apabila lebih dari 1 volt maka buzzer akan menyala dan apabila tidak maka buzzer akan mati. Perbandingan ini didapatkan karena ADC dari Arduino 10 bit analog to digital converter sehingga untuk mengetahui besar nilai 1 volt yaitu dengan cara mengubah menjadi digital. Apabila tegangan maksimum yang dapat di deteksi adalah 5 volt sedangkan ADC pada Arduino 10 bit yaitu 1024 maka untuk 1 volt = $1024/5$ yaitu sebesar 204,8 dibulatkan sehingga sebesar 205. Oleh karena itu apabila nilai pembacaan dari sensor gas lebih besar dari 205 maka buzzer akan menyala.

BAB IV PENGUJIAN DAN ANALISIS

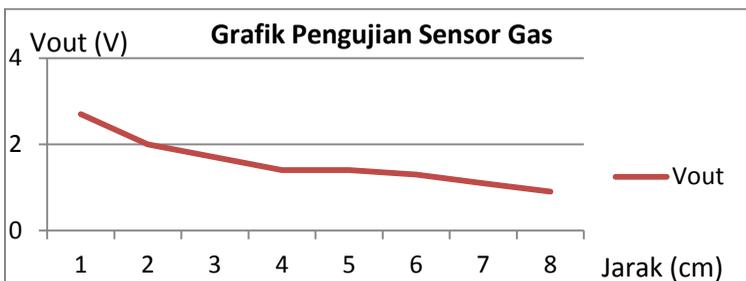
Pada bab ini akan dilakukan pengambilan data dan pengujian sistem. Pengambilan data ini berupa karakterisasi sensor gas dan driver motor, juga menguji kamera untuk mendeteksi benda menggunakan metode HSV. Setelah dilakukan karakterisasi kemudian dilakukan pengujian sistem secara keseluruhan.

4.1. Pengujian Sensor Gas

Pengujian ini bertujuan untuk mengetahui seberapa jauh sensor gas yang digunakan dapat mendeteksi dan sensitivitas dari sensor tersebut.



Gambar 4. 1 *Pengujian Sensor Gas*



Gambar 4. 2 Grafik Pengujian Sensor Gas

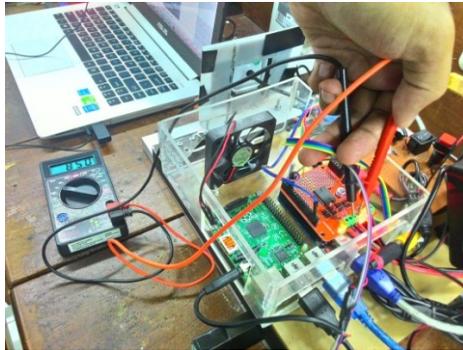
Tabel 4. 1 Hasil Pengujian Sensor Gas Terhadap Jarak Tabel

Jarak	Tegangan
1 cm	2.7 v
2 cm	2 v
3 cm	1.7 v
4 cm	1.4 v
5 cm	1.4 v
6 cm	1.3 v
7 cm	1.1 v
8 cm	0.9 v

Pengujian dilakukan dengan jarak yang berbeda dan menggunakan jenis cairan yang sama yaitu Alkohol 70%. Dari hasil data pada tabel 4.1, didapatkan yaitu jika gas berada pada jarak 1 cm di depan sensor gas tegangan yang terbaca sebesar 2.7 volt. Sedangkan apabila gas berada pada jarak 5 cm didepan gas, tegangan yang terbaca 1.4 volt. Hal ini dapat disimpulkan bahwa semakin jauh gas yang terbaca maka besarnya tegangan yang keluar semakin kecil. Oleh karena itu sensor gas harus tepat diletakkan didepan sumber gas dengan jarak kurang lebih 5 cm.

4.2. Pengujian Driver Motor

Pengujian driver motor dengan cara memberi supply 12 V dan mengetahui seberapa besar tegangan yang dikeluarkan apabila nilai PWM berubah-ubah. Pada pengujian ini nantinya akan diukur seberapa besar tegangan keluaran pada setiap motor sehingga kecepatan pada motor juga dapat diukur. Variabel yang diganti pada setiap motor adalah besarnya PWM yang dikeluarkan. Jika PWM yang disetting pada tiap motor kecil maka kecepatan motor juga akan sebanding dengan PWM. Akan tetapi untuk setiap gerakan pada robot menggunakan nilai PWM yang berbeda. Apabila robot bergerak ke kiri maka PWM pada motor kanan akan diperbesar dan PWM motor kiri akan diperkecil tetapi tidak bernilai 0. Hal ini dilakukan untuk mencegah terjadinya keterlambatan respon pada motor apabila mengganti polaritas secara tiba-tiba.



Gambar 4. 3 Pengujian Driver Motor

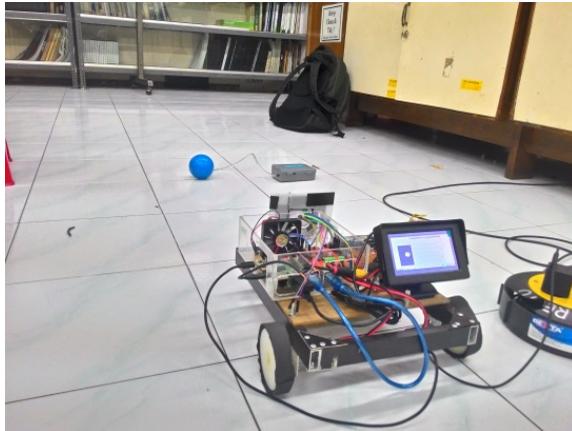
Pada gambar 4.2, hasil yang didapat, yaitu jika PWM bernilai 0, maka tegangan outputnya 0V, sedangkan apabila PWM yang diberikan sebesar 19% dari *duty cycle*, tegangan outputnya sebesar 1.6 V. Pada saat PWM bernilai 39% dari *duty cycle*, tegangan outputnya sebesar 6.3 V, dan apabila PWM yang diberikan sebesar 78% dari *duty cycle*, tegangan outputnya sebesar 9.5 V. Dari data pada tabel 4.2 didapatkan bahwa semakin besar PWM yang digunakan maka tegangan yang digunakan untuk menggerakkan motor juga semakin besar sehingga kecepatan pada motor DC akan semakin cepat. Tetapi pada motor yang digunakan kecepatan maksimum yang dicapai yaitu sebesar 38rpm sehingga apabila PWM mencapai maksimum yaitu 100% *duty cycle* maka kecepatan yang dicapai adalah sebesar 38rpm. Berikut adalah tabel hasil pengujian driver motor L298P:

Tabel 4. 2 Hasil Pengujian Driver Motor

PWM	Output
19% <i>duty cycle</i>	1.6 V
24% <i>duty cycle</i>	2.8 V
27% <i>duty cycle</i>	3.9 V
31% <i>duty cycle</i>	4.8 V
39% <i>duty cycle</i>	5.7 V
58% <i>duty cycle</i>	8.5 V
78% <i>duty cycle</i>	9.5 V

4.3. Pengujian Deteksi Benda Menggunakan Kamera

Pada pengujian ini akan dilakukan pendeteksian benda yang berupa bola tetapi menggunakan warna yang berbeda-beda. Pengujian ini dilakukan untuk mengetahui nilai Threshold HSV untuk tiap masing-masing warna. Gambar 4.3 merupakan pengujian nilai Hue pada benda berwarna biru.



Gambar 4. 4 Pengujian Nilai Hue Dari Obyek

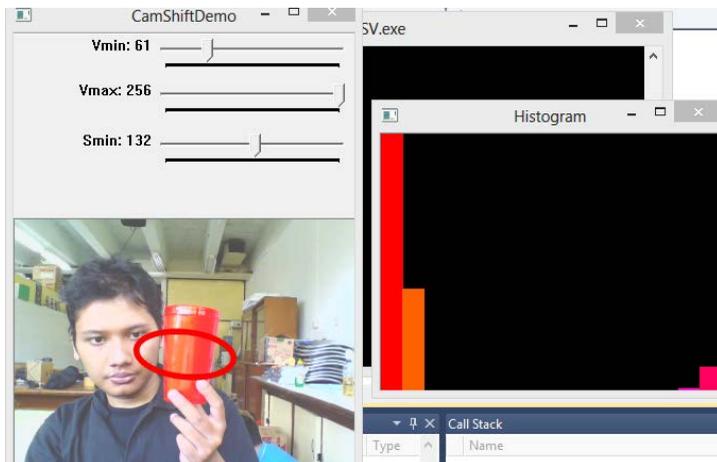
Tabel 4. 3 Nilai Hue Pada Warna yang Terdeteksi

Warna	Nilai Hue
Orange	0 – 22
Kuning	22 – 38
Hijau	38 – 75
Biru	75 – 130
Violet	130 – 160
Merah	160 – 179

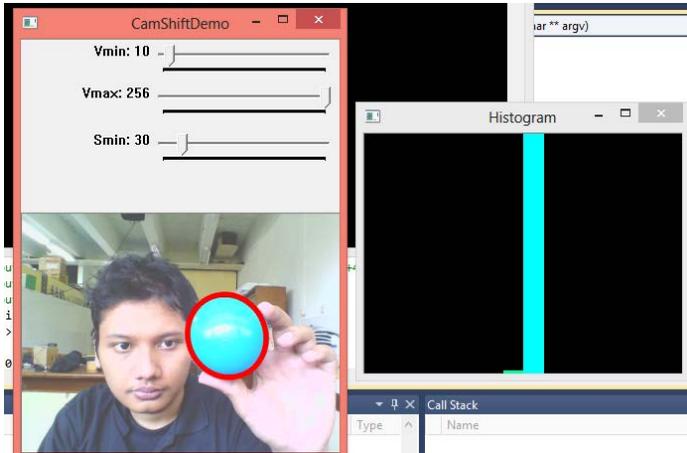
Dari hasil pengujian pada tabel 4.3, untuk mendapatkan nilai warna sesuai dengan HSV maka yang harus diatur adalah besaran nilai *Hue* sedangkan *Saturation* (S) dan *Value* (V) bergantung dengan pencahayaan. Apabila kondisi ketika cahaya cukup maka nilai *Saturation* dan *Value* relatif kecil sedangkan apabila kondisi cahaya

redup nilai Saturation dan Value relatif tinggi untuk mengatur sensitivitas warna terhadap ruangan. Selain itu pemantulan cahaya juga berpengaruh sehingga apabila cahaya yang dipantulkan sangat banyak maka lantai juga akan menimbulkan pantulan pada obyek yang dituju sehingga di kamera akan terlihat dua obyek yang memiliki nilai HSV yang sama dan hal ini dapat mempengaruhi dalam pendeteksian obyek.

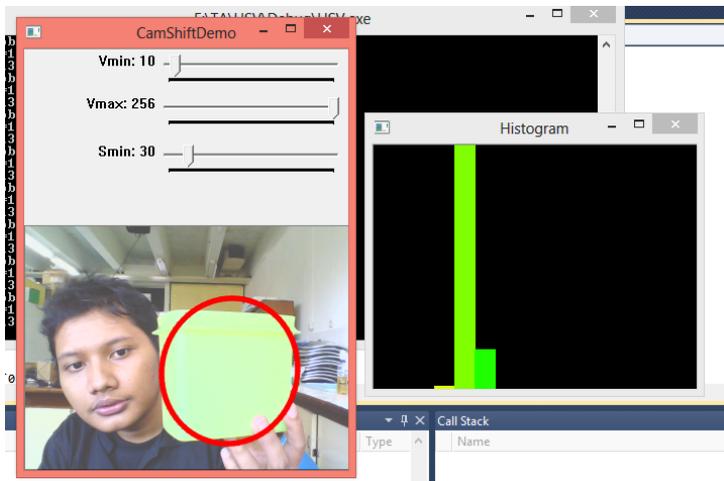
Pengujian deteksi benda menggunakan algoritma *camshift* akan menghasilkan nilai histogram dari *hue* pada objek yang telah diberi ROI (*Region Of Interest*) dengan menggunakan pointer. Hasil histogram untuk setiap warna akan berbeda-beda seperti berikut :



Gambar 4. 5 Nilai Histogram Untuk Warna Merah.



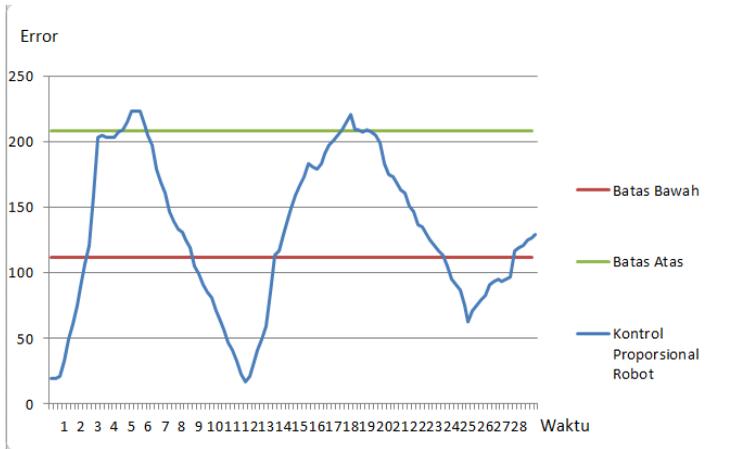
Gambar 4. 6 Hasil Histogram Untuk Benda Berwarna Biru.



Gambar 4. 7 Hasil Histogram Untuk Benda Berwarna Hijau.

4.4. Pengujian Kontrol Proporsional

Pada pengujian ini akan dilakukan kontrol proporsional pada sistem untuk memperkecil error. Error ini merupakan error posisi piksel pada frame yang diambil. Berikut adalah grafik hasilnya



Gambar 4. 8 Grafik Kontrol Proporsional

Pada gambar 4.4, grafik dapat dilihat hasilnya yaitu ketika benda berada di sebelah kiri maka error yang dihasilkan relatif besar dibandingkan dengan apabila benda yang berada sebelah kanan. Apabila benda berada di sebelah kanan maka nilai error piksel diatas 160 dan apabila benda berada di sebelah kiri maka error yang dihasilkan berada dibawah 160. Error dibawah 160 lebih sulit untuk kembali daripada saat error diatas 160. Berarti dapat disimpulkan robot ini sulit untuk kembali saat berbelok ke kiri daripada belok kanan. Hal ini disebabkan mekanik robot tidak sempurna. Disini menggunakan batas atas sebesar 208 dan batas bawah 112. Hal ini dilakukan untuk kompensasi error saat robot mendekati benda. Apabila robot mendekat pada benda maka gambar dari obyek yang di deteksi akan menjadi besar sehingga posisi benda tidak lagi tepat di tengah. Hal ini untuk mengatasi kekeliruan apabila benda dianggap tidak berada di tengah. Pada pengujian ini terdapat error rata-rata yaitu sebesar 34,806%.

4.5. Pengujian Keseluruhan Sistem

Pada pengujian ini akan dilakukan pengujian keseluruhan sistem untuk mengetahui apakah sistem ini mencapai tujuan yang diberikan yaitu menuju pada benda yang ditentukan dengan menggunakan algoritma *CamShift* dan mampu mendeteksi kebocoran gas pada benda tersebut.

Tabel 4. 4 Tabel Pengujian Sistem

Pengujian	Hasil
Pengujian 1	Berhasil
Pengujian 2	Gagal
Pengujian 3	Gagal
Pengujian 4	Berhasil
Pengujian 5	Berhasil

Dari hasil pengujian yang dilakukan maka dapat disimpulkan bahwa robot mampu menuju benda yang dituju dan mendeteksi kebocoran gas alkohol pada benda tersebut. Akan tetapi masih terdapat error yaitu sekitar 40% dari total percobaan yang dilakukan.

BAB V

PENUTUP

5.1 Kesimpulan

Dari beberapa percobaan yang dilakukan dapat ditarik kesimpulan yaitu robot mampu menuju ke obyek yang ditentukan dengan metode konversi dari RGB ke HSV. Robot juga mampu mendeteksi gas alkohol pada benda yang berjarak kurang lebih 3cm. Untuk dapat memaksimalkan Raspberry Pi maka digunakan image processing dengan lebar frame yang kecil sehingga dapat melakukan pemrosesan gambar meskipun tidak maksimal dikarenakan memori yang digunakan pada Raspberry Pi kecil. Penggunaan algoritma *CamShift* memudahkan untuk mendeteksi obyek dengan warna yang berbeda dengan menyesuaikan histogram yang dibentuk saat menentukan ROI (*Region Of Interest*). Kontrol Proporsional yang digunakan pada robot mampu membuat robot mencapai target dengan error rata-rata sebesar 34,806%. Akan tetapi sistem ini masih terdapat error sekitar 40% dari total percobaan yang dilakukan hal ini dikarenakan oleh banyak faktor seperti desain mekanik dan pemrosesan pada Single Board Computer.

5.2 Saran

Saran yang diberikan pada sistem ini untuk mencegah terjadinya error sebaiknya digunakan Single Board Computer yang lebih baik dalam pengolahan citra dan memiliki memori yang relatif besar sehingga proses tidak terjadi delay. Diperlukan suatu metode untuk dapat menghindari halangan yang ada sehingga robot dapat mengenali dan membedakan antara halangan dengan obyek yang dituju. Obyek yang akan dituju disarankan berada pada kondisi cahaya yang cukup sehingga warna terlihat saling mencolok satu sama lain. Mekanik Robot dapat diperbaiki menjadi lebih baik untuk memudahkan pergerakan robot dan melakukan manuver yang lebih baik sehingga robot dapat lebih presisi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1]. _____, <http://digilib.its.ac.id/public/ITS-Undergraduate-17442-Chapter1-575320.pdf>
- [2]. Benedictus Yoga Budi Putranto, Widi Hapsari, Katon Wijana, ”*Segmentasi Warna Citra Dengan Deteksi Warna HSV Untuk Mendeteksi Objek*”, Fakultas Teknik Program Studi Teknik Informatika Universitas Kristen Duta Wacana, Yogyakarta, 2010.
- [3]. Hong-Kui Liu, Jun Zhou, “*Moving Object Detecting And Tracking Method Based On Color Image*”, Proceedings of the 7th World Congress on Intelligent Control and Automation, China, June 2008.
- [4]. Liu Dong, Xi Lin, “*Monocular-Vision-Based Study on Moving Object Detection and Tracking*”, Beijing Union University, 2015.
- [5]. Darius, Dr. Achmad Arifin, ST., M.Eng., Pujiono S.T., M.T., “*Sistem Pengukuran Gerakan Lower Limb Segment Pada Sagittal Plane Dengan Menggunakan Sebuah Web Camera Dan Marker*”, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2015
- [6]. Resmana Lim, Yulia Roy Otniel Pantouw, “*Pelacakan dan Pengenalan Wajah Menggunakan Webcam & Metode Gabor Filter*”, Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra,_____
- [7]. Kurnia Khafidhatur Rafiah, Rita Magdalena,Ir., M.T., Nur Andini, S.T.,MT., “*Deteksi Rambu-Rambu Lalu Lintas Untuk Membantu Pengguna Jalan Raya Dengan Algoritma CamShift*”, Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung,_____
- [8]. Shanty Eka Agustina, Imam Mukhlash, “*Implementasi Metode Scale Invariant Feature Transform (SIFT) Dan Metode Continuously Adaptive Mean-Shift (Camshift) Pada Penjejakan Objek Bergerak*”, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS), Surabaya,_____

- [9]. _____, “*Raspberry Pi Model B+ Specification*”
<https://www.adafruit.com/datasheets/pi-specs.pdf>
- [10]. Feri Djuandi, “Pengenalan Arduino”, Juli 2011.
- [11]. MrLndr, “*FAQ: The Arduino Anatomy*”,
URL:<http://arduinoarts.com/2011/08/the-arduino-uno-anatomy/>,
9 Agustus 2011.

LAMPIRAN

Program OpenCV

```
#ifdef _CH_
#pragma package <opencv>
#endif

#define CV_NO_BACKWARD_COMPATIBILITY

#ifdef _EiC
#include "cv.h"
#include "highgui.h"
#include <stdio.h>
#include <ctype.h>
#endif

#include <iostream>

#include<cvaux.h>
#include<highgui.h>
#include<cxcore.h>

#include <sstream>
#include <string>
#include <cv.h>

#include<stdio.h>
#include<stdlib.h>

// Need to include this for serial port communication
#include <Windows.h>
using namespace std;

IplImage *image = 0, *hsv = 0, *hue = 0, *mask = 0,
*backproject = 0, *histimg = 0;
IplImage *image2 = 0;
CvHistogram *hist = 0;

int backproject_mode = 0;
int select_object = 0;
```

```

int track_object = 0;
int show_hist = 1;
CvPoint origin;
CvRect selection;
CvRect track_window;
CvBox2D track_box;
CvConnectedComp track_comp;
CvPoint2D32f koordinat;
CvSize2D32f radius;
int r;
int x;
int y;
int hdims = 16;
float hranges_arr[] = {0,180};
float* hranges = hranges_arr;
int vmin = 10, vmax = 256, smin = 30;

void on_mouse( int event, int x, int y, int flags,
void* param )
{
    if( !image )
        return;

    if( image->origin )
        y = image->height - y;

    if( select_object )
    {
        selection.x = MIN(x,origin.x);
        selection.y = MIN(y,origin.y);
        selection.width = selection.x + CV_IABS(x -
origin.x);
        selection.height = selection.y + CV_IABS(y -
origin.y);

        selection.x = MAX( selection.x, 0 );
        selection.y = MAX( selection.y, 0 );
        selection.width = MIN( selection.width, image-
>width );
    }
}

```



```

int main( int argc, char** argv )
{
    // Setup serial port connection and needed
    // variables.
    HANDLE hSerial = CreateFile(L"COM2",
    GENERIC_READ | GENERIC_WRITE, 0, 0, OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL, 0);

    if (hSerial !=INVALID_HANDLE_VALUE)
    {
        printf("Port opened! \n");

        DCB dcbSerialParams;
        GetCommState(hSerial,&dcbSerialParams);

        dcbSerialParams.BaudRate = CBR_9600;
        dcbSerialParams.ByteSize = 8;
        dcbSerialParams.Parity = NOPARITY;
        dcbSerialParams.StopBits = ONESTOPBIT;

        SetCommState(hSerial, &dcbSerialParams);
    }
    else
    {
        if (GetLastError() ==
    ERROR_FILE_NOT_FOUND)
        {
            printf("Serial port doesn't exist!
\n");
        }

        printf("Error while setting up serial
port! \n");
    }

    char outputChars[3]=" ";
    char a[5]=" ";
    char b[5]=" ";
    char c[5]=" ";

```

```

char d[5]=" ";
//char e[5]=" ";
DWORD btsIO;

CvSize size640x480 = cvSize(320, 240);
CvCapture* capture = 0;

cvSetCaptureProperty(capture,CV_CAP_PROP_FRAME_W
IDTH,320);

cvSetCaptureProperty(capture,CV_CAP_PROP_FRAME_H
EIGHT,240);

if( argc == 1 || (argc == 2 && strlen(argv[1]) == 1
&& isdigit(argv[1][0])))
    capture = cvCaptureFromCAM( argc == 2 ?
argv[1][0] - '0' : 0 );
else if( argc == 2 )
    capture = cvCaptureFromAVI( argv[1] );

if( !capture )
{
    fprintf(stderr,"Could not initialize
capturing...\n");
    return -1;
}

printf( "Hot keys: \n"
        "\tESC - quit the program\n"
        "\tc - stop the tracking\n"
        "\tb - switch to/from backprojection view\n"
        "\th - show/hide object histogram\n"
        "To initialize tracking, select the object with
mouse\n" );

cvNamedWindow( "Histogram", 1 );
cvNamedWindow( "CamShiftDemo", 1 );
cvSetMouseCallback( "CamShiftDemo", on_mouse, 0 );
cvCreateTrackbar( "Vmin", "CamShiftDemo", &vmin,
256, 0 );

```

```

    cvCreateTrackbar( "Vmax", "CamShiftDemo", &vmax,
256, 0 );
    cvCreateTrackbar( "Smin", "CamShiftDemo", &smin,
256, 0 );

    for(;;)
    {
        IplImage* frame = 0;
        int i, bin_w, c;

        frame = cvQueryFrame( capture );
        if( !frame )
            break;

        if( !image )
        {
            /* allocate all the buffers */
            image =
cvCreateImage(cvSize(320,240),8,3);
            cvResize(frame,image,1);
            hsv = cvCreateImage(
cvGetSize(image), 8, 3 );
            hue = cvCreateImage( cvGetSize(image), 8, 1
);
            mask = cvCreateImage( cvGetSize(image), 8,
1 );
            backproject = cvCreateImage(
cvGetSize(image), 8, 1 );
            hist = cvCreateHist( 1, &hdims,
CV_HIST_ARRAY, &hranges, 1 );
            histimg = cvCreateImage( cvGetSize(image),
8, 3 );
            cvZero( histimg );
        }

        cvResize(frame,image,1);
        cvCvtColor( image, hsv, CV_BGR2HSV );

        if( track_object )
        {

```

```

        cout << "track_object" <<
track_object << endl;
        int _vmin = vmin, _vmax = vmax;

        cvInRangeS( hsv,
cvScalar(0,smin,MIN(_vmin,_vmax),0),
cvScalar(180,256,MAX(_vmin,_vmax),0), mask );
        cvSplit( hsv, hue, 0, 0, 0 );

        if( track_object < 0 )
        {
            float max_val = 0.f;
            cvSetImageROI( hue, selection );
            cvSetImageROI( mask, selection );
            cvCalcHist( &hue, hist, 0, mask );
            cvGetMinMaxHistValue( hist, 0,
&max_val, 0, 0 );
            cvConvertScale( hist->bins, hist->bins,
max_val ? 255. / max_val : 0., 0 );
            cvResetImageROI( hue );
            cvResetImageROI( mask );
            track_window = selection;
            track_object = 1;

            cvZero( histimg );
            bin_w = histimg->width / hdims;
            for( i = 0; i < hdims; i++ )
            {
                int val = cvRound(
cvGetReal1D(hist->bins,i)*histimg->height/255 );
                CvScalar color =
hsv2rgb(i*180.f/hdims);
                cvRectangle( histimg,
cvPoint(i*bin_w,histimg->height),
cvPoint((i+1)*bin_w,histimg->height - val),
                    color, -1, 8, 0 );
            }

```

```

    }

    cvCalcBackProject( &hue, backproject, hist
);
    cvAnd( backproject, mask, backproject, 0 );
    cvCamShift( backproject, track_window,
                cvTermCriteria( CV_TERMCRIT_EPS
| CV_TERMCRIT_ITER, 10, 1 ),
                &track_comp, &track_box );
    track_window = track_comp.rect;
    koordinat = track_box.center;
    radius = track_box.size;
    r=(radius.width + radius.height)/2;
    x= koordinat.x;
    y= koordinat.y;
    //cvPointFrom32f(radius);
    cvPointFrom32f(koordinat);

    cout << "x=" << x << "y=" << y <<
endl;

    cout << "lebar=" << r << endl;

    //p_fltXYRadius1 =
(float*)cvGetSeqElem(p_seqCircles, 1);
    outputChars[0] = (x/100)+48;
    outputChars[1] = ((x%100)/10)+48 ;
    outputChars[2] = (x % 10 ) +48 ;
    //outputChars[0] = (int
(koordinat.x[0])/100)+48;
    //outputChars[1] = ((int
(p_fltXYRadius[0]) %100) / 10)+48;
    //outputChars[2] = (int
(p_fltXYRadius[0]) % 10)+48;
    //outputChars[] = int
(p_fltXYRadius[0]) % 10;
    WriteFile(hSerial, outputChars, 3 ,
&btsIO, NULL);
    if (r > 100)
    {
        b[0] = 'b';
    }

```

```

        WriteFile(hSerial, b, 1,
&btsIO, NULL);
    }

    // Check whether camera should turn
to its right if the circle gets near the left end of
the screen
    if (r < 50)
    {
        b[0] = 'f';
        WriteFile(hSerial, b, 1,
&btsIO, NULL);
    }

    if (r > 50 && r < 100 )
    {
        b[0] = 's';
        WriteFile(hSerial, b, 1,
&btsIO, NULL);
    }

    if( backproject_mode )
        cvCvtColor( backproject, image,
CV_GRAY2BGR );
    if( !image->origin )
        track_box.angle = -track_box.angle;
    cvEllipseBox( image, track_box,
CV_RGB(255,0,0), 3, CV_AA, 0 );
}

    if( select_object && selection.width > 0 &&
selection.height > 0 )
    {
        cvSetImageROI( image, selection );
        cvXorS( image, cvScalarAll(255), image, 0
);
        cvResetImageROI( image );

```

```

    }

    cvShowImage( "CamShiftDemo", image );
    cvShowImage( "Histogram", histimg );

    c = cvWaitKey(10);
    if( (char) c == 27 )
        break;
    switch( (char) c )
    {
    case 'b':
        backproject_mode ^= 1;
        break;
    case 'c':
        track_object = 0;
        cvZero( histimg );
        break;
    case 'h':
        show_hist ^= 1;
        if( !show_hist )
            cvDestroyWindow( "Histogram" );
        else
            cvNamedWindow( "Histogram", 1 );
        break;
    default:
        ;
    }
}

cvReleaseCapture( &capture );
// cvReleaseCapture( &image );
cvDestroyWindow("CamShiftDemo");
cvDestroyWindow("Histogram");

return 0;
}

#ifdef _EiC
main(1, "camshiftdemo.c");
#endif

```

Program Arduino

```
//LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

char a[] = " ";

char c[] = " ";

int i = 0 ;

int b = 0;

int pwm_a = 3; //PWM control for motor outputs 1 and 2 is on digital
pin 3

int pwm_b = 11; //PWM control for motor outputs 3 and 4 is on digital
pin 11

int dir_a = 12; //direction control for motor outputs 1 and 2 is on digital
pin 12

int dir_b = 13; //direction control for motor outputs 3 and 4 is on digital
pin 13

int val = 0; //value for fade

int nilai = 0;

int nilai1 = 0;

float nilai2 = 0;

int kp = 0;

boolean arah1;

boolean arah2;

const int pingPin = 7;

//int val = 0;
```

```

void setup()
{
  //pinMode(13,OUTPUT);

  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  long duration, inches, cm;

  //lcd.begin(16, 2);

  // Print a message to the LCD.
  //lcd.print("cek serial");

  pinMode(3, OUTPUT); //Set control pins to be outputs
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);

  analogWrite(pwm_a, kp); //set both motors to run at (100/255 = 39)%
  duty cycle (slow)

  analogWrite(pwm_b, kp);
  pinMode(2,OUTPUT);
}

void loop() // run over and over
{
  //stopped();
}

```

```

if (nilai2 < (-0.40))
{
    arah1=false;
    arah2=true;
    kp=(((nilai2*(-1.00))*10)+80);
    //kp = nilai2;
    //digitalWrite(dir_a, arah1); //Reverse motor direction, 1 high, 2 low
    //digitalWrite(dir_b, arah2); //Reverse motor direction, 3 low, 4 high
    //analogWrite(pwm_a, kp); //set both motors to run at (100/255 =
39)% duty cycle
    //analogWrite(pwm_b, kp);
}
else if (nilai2 > 0.40)
{
    //nilai1=(nilai1*(-1));
    arah1=true;
    arah2=false;
    kp =(((nilai2*1.00)*10)+80);
    //kp = nilai2;
    //digitalWrite(dir_a, arah1); //Reverse motor direction, 1 high, 2 low
    //digitalWrite(dir_b, arah2); //Reverse motor direction, 3 low, 4 high

```

```

//analogWrite(pwm_a, kp); //set both motors to run at (100/255 =
39)% duty cycle

//analogWrite(pwm_b, kp);
}

else if (nilai2 > (-0.40)&& nilai2<(0.40))
{
//nilai1=(nilai1*(-1));

if (a[3]=='f'){

    arah1=false;

    arah2=false;

    kp=90;

}

if (a[3]=='b'){

    arah1=true;

    arah2=true;

    kp=90;

}

if (a[3]=='s'){

    arah1=false;

    arah2=false;

    kp=0;

```

```

    }
}
else
{
    arah1=false;
    arah2=false;
    kp=0;
}

digitalWrite(dir_a, arah1); //Reverse motor direction, 1 high, 2 low
digitalWrite(dir_b, arah2); //Reverse motor direction, 3 low, 4 high
analogWrite(3,kp);
analogWrite(11,kp);

digitalWrite(2,LOW);
val = analogRead(A0); // read the input pin
if (val > 205){
    digitalWrite(2,HIGH);
    delay(1000);
}
else {

```

```
    digitalWrite(2,LOW);  
  }  
  //delay(1000);  
  }  
void serialEvent() {  
  while (Serial.available()) {  
    // get the new byte:  
    if (i<=3)  
    { if(i==0)  
      {  
        a[0]= Serial.read();  
      }  
      else if(i==1)  
      {  
        a[1]= Serial.read();  
      }  
      else if(i==2)  
      {  
        a[2]= Serial.read();  
      }  
      else if(i==3)
```

```

    {
        a[3]= Serial.read();
        i=-1;
    }
    i++;
}
}

nilai = (((int)a[0]-48)*100 + ((int)a[1]-48)*10 + ((int)a[2]-48));
nilai2 = (nilai-160.00)/160.00;

//char c[0] = a[3];
}

void forw() // no pwm defined
{
    digitalWrite(dir_a, HIGH); //Reverse motor direction, 1 high, 2 low
    digitalWrite(dir_b, HIGH); //Reverse motor direction, 3 low, 4 high
}

void back() // no pwm defined
{
    digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high
    digitalWrite(dir_b, LOW); //Set motor direction, 3 high, 4 low
}

```

```

void forward() //full speed forward
{
    digitalWrite(dir_a, HIGH); //Reverse motor direction, 1 high, 2 low
    digitalWrite(dir_b, HIGH); //Reverse motor direction, 3 low, 4 high
    analogWrite(pwm_a, nilai); //set both motors to run at (100/255 =
39)% duty cycle
    analogWrite(pwm_b, nilai);
}

void backward() //full speed backward
{
    digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high
    digitalWrite(dir_b, LOW); //Set motor direction, 3 high, 4 low
    analogWrite(pwm_a, 100); //set both motors to run at 100% duty
cycle (fast)
    analogWrite(pwm_b, 100);
}

void stopped() //stop
{
    digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high
    digitalWrite(dir_b, LOW); //Set motor direction, 3 high, 4 low

```

```

    analogWrite(pwm_a, 0); //set both motors to run at 100% duty cycle
(fast)

    analogWrite(pwm_b, 0);

}

void right() //full speed backward

{

    digitalWrite(dir_a, LOW); //Set motor direction, 1 low, 2 high

digitalWrite(dir_b, HIGH); //Set motor direction, 3 high, 4 low

    analogWrite(pwm_a, nilai); //set both motors to run at 100% duty
cycle (fast)

    analogWrite(pwm_b, nilai);

}

void left() //full speed backward

{

    digitalWrite(dir_a, HIGH); //Set motor direction, 1 low, 2 high

    digitalWrite(dir_b, LOW); //Set motor direction, 3 high, 4 low

    analogWrite(pwm_a, kp); //set both motors to run at 100% duty cycle
(fast)

    analogWrite(pwm_b, kp);

}

void fadein()

```

```

{
  // fade in from min to max in increments of 5 points:
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5)
  {
    // sets the value (range from 0 to 255):
    analogWrite(pwm_a, fadeValue);
    analogWrite(pwm_b, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}

```

```

void fadeout()

```

```

{
  // fade out from max to min in increments of 5 points:
  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5)
  {
    // sets the value (range from 0 to 255):
    analogWrite(pwm_a, fadeValue);
    analogWrite(pwm_b, fadeValue);
    // wait for 30 milliseconds to see the dimming effect

```

```

    delay(30);
}
}
void astop()          //stop motor A
{
    analogWrite(pwm_a, 0); //set both motors to run at 100% duty cycle
    (fast)
}
void bstop()         //stop motor B
{
    analogWrite(pwm_b, 0); //set both motors to run at 100% duty cycle
    (fast)
}
long microsecondsToInches(long microseconds)
{
    // According to Parallax's datasheet for the PING))), there are
    // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
    // second). This gives the distance travelled by the ping, outbound
    // and return, so we divide by 2 to get the distance of the obstacle.
    // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-
    v1.3.pdf
    return microseconds / 74 / 2;
}

```

```
}  
long microsecondsToCentimeters(long microseconds)  
{  
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.  
    // The ping travels out and back, so to find the distance of the  
    // object we take half of the distance travelled.  
    return microseconds / 29 / 2;  
}
```

BIODATA PENULIS



Penulis bernama lengkap Hanifar Kahira lahir pada tanggal 08 Maret 1993. Anak pertama dari dua bersaudara. Penulis mengawali kegiatan pendidikan formal di SD Khadijah II Surabaya, yang kemudian dilanjutkan di SMP Negeri 3 Surabaya, kemudian SMA Negeri 6 Surabaya dan pada tahun 2011 penulis diterima sebagai mahasiswa di jurusan Teknik Elektro ITS. Saat kuliah, penulis turut berpartisipasi sebagai asisten praktikum di bidang studi elektronika serta aktif pula dalam unit kegiatan mahasiswa workshop Elektro ITS. Selain itu penulis pernah menjadi bagian dari tim Robotika ITS Surabaya.

Email : vanagongokil@gmail.com

Perancangan Robot Beroda Untuk Mendeteksi Gas Berbahaya Menggunakan Image Processing Berbasis *Single Board Computer*

Hanifar Kahira, Muhammad Rivai¹⁾, Rudy Dikairono.²⁾

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: muhammad_rivai@ee.its.ac.id

Abstrak - Gas dapat dideteksi melalui indra penciuman. Akan tetapi hal tersebut dapat membahayakan manusia dikarenakan gas yang di hirup dapat berupa racun. Banyak benda yang mengandung suatu gas berbahaya yang dapat menyebabkan suatu bencana apabila terjadi kebocoran. Oleh karena itu dibutuhkan suatu sistem yang mampu mendeteksi suatu kebocoran gas tanpa merugikan manusia.

Teknologi robot beroda memungkinkan untuk dapat melakukan manuver secara cepat sehingga dapat dimanfaatkan untuk membuat mekanisme kerja robot yang dapat mendeteksi gas berbahaya. Hal ini dimaksudkan untuk mengganti peran manusia dalam mendeteksi bahan berbahaya agar tidak terjadi penambahan korban jiwa. *Single Board Computer* digunakan karena memiliki kelebihan bersifat *open source* dan prosesor yang lebih tinggi dari pada mikrokontroler. Selain itu robot ini dilengkapi dengan kamera webcam sebagai media untuk menangkap obyek sehingga robot dapat berjalan menuju obyek dan mendeteksi kebocoran gas pada benda yang dituju. Sensor gas yang digunakan sensitif terhadap gas LPG, alkohol, benzena dan heksana. Pergerakan robot menggunakan kontrol proporsional untuk menuju ke target dengan error rata-rata sebesar 34,806%. Sedangkan robot ini sendiri jika diuji seluruhnya mampu mencapai target dengan kesalahan sebesar 40% dari total pengujian yang dilakukan.

Kata kunci : Gas Berbahaya, Robot Beroda, *Single Board Computer*

PENDAHULUAN

Banyak sekali benda asing mencurigakan yang berpotensi membahayakan keselamatan manusia di sekitar dan mengandung gas yang berbahaya. Identifikasi suatu benda yang dianggap berbahaya adalah ditemukannya obyek mencurigakan kemudian didekati secara perlahan oleh manusia. Hal ini masih bersifat konvensional dan membahayakan manusia saat proses pengidentifikasian karena perlu mendekati pada bahan berbahaya dan itu dapat meledak, jika terlambat dapat mengakibatkan luka hingga meninggal. Karena membahayakan bagi manusia, perlu dibuat suatu robot yang mampu menggantikan fungsi manusia untuk mengidentifikasi suatu objek apakah objek tersebut mengandung gas berbahaya atau tidak sehingga dapat mencegah terjadinya hal-hal yang tidak diinginkan. Robot ini juga nantinya diharapkan mampu melewati halangan yang ada sehingga mampu menjangkau tempat yang sulit. Robot ini dilengkapi dengan sensor gas yang mampu mengenali objek tersebut apakah mengandung gas berbahaya atau tidak. Robot ini juga dilengkapi dengan kamera sehingga dapat membedakan objek yang akan dituju dengan halangan yang ada. Diharapkan implementasi pada robot ini dapat mampu menuju ke arah sumber bahan peledak dan mengidentifikasi objek tertentu apakah itu berbahaya atau tidak.

I. TEORI PENUNJANG

A. *Mobile Robot*

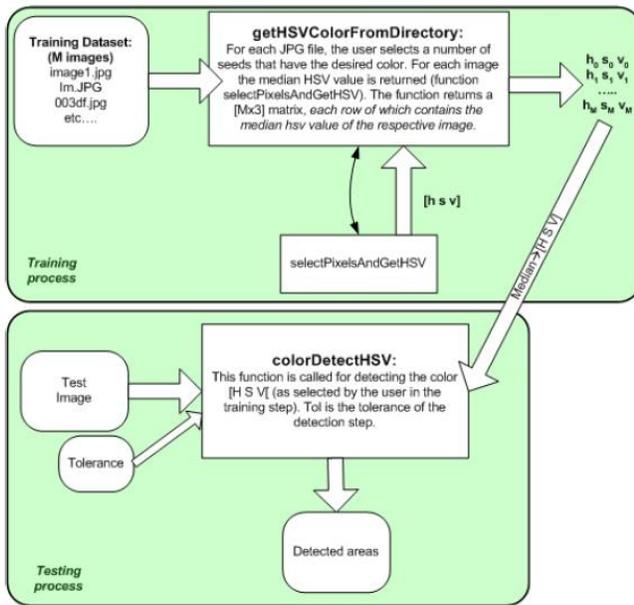
mobile robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik lain. *Mobile robot* adalah tipe robot yang paling populer dalam dunia penelitian robot. Dari segi manfaat, penelitian tentang berbagai tipe *mobile robot* diharapkan dapat membantu manusia dalam melakukan otomasi dalam transportasi, platform bergerak untuk robot industri, eksplorasi tanpa awak dan masih banyak lagi.

B. *Image Detection*

Pada metode segmentasi dengan deteksi warna HSV dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Citra digital menggunakan model warna RGB sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna RGB ke HSV. Untuk membentuk segmen sesuai dengan warna yang diinginkan maka ditentukan nilai toleransi pada setiap dimensi warna HSV, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses adaptive threshold. Hasil dari proses threshold tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan. Secara garis besar, gambaran proses segmentasi adalah sebagai berikut [1]:

- Tentukan citra RGB yang menjadi obyek deteksi, nilai warna HSV yang menjadi acuan (hasil proses pelatihan data) dan nilai toleransi HSV yang digunakan.
- Transpose citra RGB ke HSV
- Lakukan filter warna pada citra berdasarkan nilai acuan (T) dan nilai toleransi (tol). Dengan x sebagai warna HSV pada piksel yang ada maka warna yang tidak termasuk dalam rentang $T - tol < x < T + tol$ diberi warna hitam.

Transpose kembali citra ke RGB, tampilkan hasil filter



Gambar 2. 1 Skema Deteksi Warna HSV

Citra dari target dapat dipengaruhi oleh sumber cahaya. Disini digunakan metode untuk mengubah RGB dan CYM ke dalam HSV yang dapat secara signifikan mampu mengurangi dampak cahaya. Model HSV terdiri dari tiga komponen penyusun yaitu: hue (H), saturation (S), and value (V). H adalah sudut dari 0 derajat hingga 360 derajat. Saturation menunjukkan kisaran abu-abu di ruang warna. Hal ini berkisar dari 0 hingga 100%. Kadang-kadang nilai dihitung dari 0 hingga 1. Value adalah kecerahan warna dan bervariasi dengan saturasi warna. Hal ini berkisar antara 0 hingga 100%. Nilai RGB ditransfer ke HSV dengan mengikuti persamaan sebagai berikut:

$$H = \cos^{-1} \frac{(R-G)+(R-B)}{\sqrt{(R-B)^2 + (R-B)(G-B)}}, B \leq G$$

$$H = 2\pi - \cos^{-1} \frac{(R-G)+(R-B)}{\sqrt{(R-B)^2 + (R-B)(G-B)}}, B > G$$
(1)

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R + B + G)}$$
(2)

$$V = \frac{\max(R, G, B)}{255}$$
(3)

Citra biner adalah mengubah kecerahan gambar menjadi 0 atau 255, 0 mewakili hitam dan 255 mewakili putih. Jadi gambar asli dapat disederhanakan menjadi latar belakang dan depan. Metode ini dapat mengurangi jumlah pengolahan data citra. Dalam gambar biner, bagaimana menemukan threshold yang tepat sangat penting. Ambang batas akan mempengaruhi citra yang diambil pada objek pelacakan. Citra biner adalah gambar dalam pixel per titik dibandingkan dengan nilai ambang batas. Jika nilai pixel lebih kecil dari ambang batas, itu diatur menjadi 0. [1]

C. Algoritma CamShift

CamShift merupakan singkatan dari *Continuously Adaptive Mean-Shift*. Algoritma CamShift merupakan

pengembangan dari algoritma dasar Mean-Shift yang dilakukan secara berulang untuk dapat melacak pergerakan dari obyek. Algoritma Mean-Shift beroperasi berdasarkan distribusi probabilitas dari citra. Untuk melacak citra berwarna pada video sequence, maka citra berwarna tersebut harus direpresentasikan dalam bentuk distribusi probabilitas citra dengan menggunakan distribusi histogram dari citra tersebut. Distribusi warna dari video sequence selalu berubah tiap waktu, maka algoritma Mean-Shift harus dikembangkan supaya dapat digunakan untuk melacak obyek. Perkembangan dari algoritma ini dinamakan algoritma CamShift.

Tahap pertama dari algoritma CamShift adalah dengan melakukan penghitungan terhadap nilai hue dari obyek sampel. Nilai histogram hue sebagai hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai lookup table.

Tahap kedua, dilakukan pemilihan lokasi awal dari Mean-Shift 2D search window. Kemudian dilakukan penghitungan nilai histogram hue yang menjadi nilai probabilitas untuk tiap-tiap pixel pada citra.

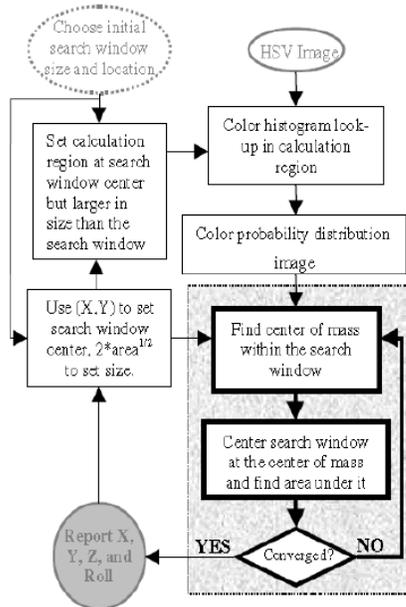
Tahap ketiga, dijalankan algoritma Mean-Shift untuk mencari pusat dan besar search window yang baru. Lokasi tengah dan besar daerah yang dihasilkan disimpan. Lalu dilakukan looping ke tahap dua dimana lokasi awal dan search window menggunakan hasil algoritma Mean-Shift pada tahap tiga.

Tahap algoritma Mean-Shift dapat dijabarkan sebagai berikut:

1. Tentukan ukuran dari search window.
2. Tentukan lokasi awal dari search window.
3. Hitung lokasi mean dalam search window.
4. Pusatkan search window pada lokasi mean yang sudah dihitung dengan langkah ke-3.
5. Ulangi langkah ke-3 dan langkah ke-4 sampai search window konvergen, yaitu sampai window tersebut berpindah dengan jarak yang kurang dari nilai ambang yang sudah ditentukan (preset threshold) .[2]

Alur dari algoritma CamShift dapat dilihat pada Gambar 2.2. Tahap pertama adalah melakukan penghitungan terhadap nilai hue dari objek target. Nilai histogram hue hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai lookup table (LUT) distribusi probabilitas citra. Tahap kedua adalah penginisiasian ukuran search window awal yaitu seukuran dengan frame pertama yang datang dari kamera. Tahap kedua atau preprocessing adalah pengestraksian fitur dari setiap frame yang diterima. Tahap ketiga merupakan inti dari algoritma CAMShift. Tidak sama dengan algoritma MeanShift yang didesain untuk gambar statis, algoritma CAMShift didesain untuk distribusi gambar yang bersifat dinamis. Hal ini terjadi karena objek yang dideteksi sebagai video bergerak yang akan menyebabkan lokasi dan ukuran search window berubah terhadap waktu. Oleh karena itu algoritma CAMShift akan selalu mengeset

ukuran search window pada setiap pengolahan frame dengan kata lain search window akan lebih adaptif. [3]



Gambar 2. 2 Alur Algoritma CamShift

Tahap algoritma *CamShift* dapat dijabarkan sebagai berikut:

- Langkah 1 : Tentukan ukuran awal search window.
- Langkah 2 : Tentukan lokasi awal dari search window.
- Langkah 3 : Tentukan daerah kalkulasi (calculation region) pada bagian tengah search window dengan ukuran lebih besar dari search window.
- Langkah 4 : Frame citra video dikonversi ke dalam sistem warna HSV(Hue, Saturation, Value), kemudian langkah selanjutnya adalah membuat histogram dari citra untuk mengetahui distribusi probabilitas warna.
- Langkah 5 : Lakukan algoritma mean shift seperti di atas (satu atau banyak iterasi) dengan input berupa ukuran dan lokasi search window serta citra distribusi probabilitas warna dan simpan zeroth moment.
- Langkah 6 : Set nilai x, y, z yang diperoleh dari langkah (5).
- Langkah 7 : Nilai x dan y dipakai untuk menentukan titik tengah search window, sedangkan $(2 \cdot \text{area}^{1/2})$ untuk menentukan ukuran search window.
- Langkah 8 : Ulangi langkah (3) untuk setiap pergantian frame citra video. [3]

D. Single Board Computer

Single Board Computer adalah komputer papan tunggal yang lengkap dengan mikroprocessor, memori, input/output (I/O) dan fitur lain yang dibutuhkan dari sebuah komputer fungsional. SBC menggunakan memori eksternal seperti kartu SD dan kartu mikro SD. Raspberry Pi adalah komputer papan tunggal (Single Board Computer/SBC) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan seperti spreadsheet, game, bahkan sebagai media player. Raspberry Pi dikembangkan oleh Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli

komputer dari Universitas Cambridge, Inggris. Raspberry Pi memiliki dua model yaitu model A dan model B. Perbedaan model A dan B terletak pada memory yang digunakan. Model A menggunakan 256 MB dan model B menggunakan 512 MB. Selain itu terdapat Raspberry Pi 2 yang memiliki memory yang lebih yaitu 1GB. Pada model B terdapat ethernet port yang tidak terdapat pada model A.

E. Sensor Gas MQ-3

Sensor MQ3 memiliki karakteristik sebagai berikut :

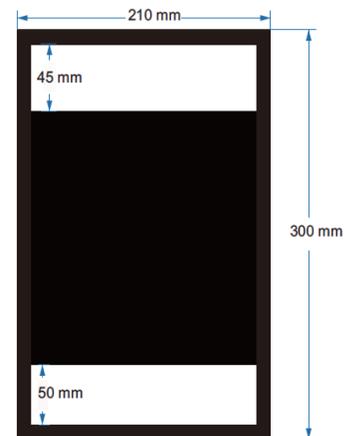
- Catu daya heater : 5V AC/DC
- Catu daya rangkaian : 5VDC
- Range pengukuran : 0,05mg/L – 10mg/L
- Cocok untuk aplikasi pengetes kadar alkohol di udara
- Output : analog (perubahan tegangan) Alcohol Gas Sensor MQ-3 merupakan Sensor alkohol yang cocok untuk mendeteksi kadar alkohol secara langsung, misal pada nafas kita. Sensor alkohol MQ-3 memiliki sensitifitas tinggi dan waktu respon yang cepat. Output dari Sensor alkohol MQ-3 ini berupa tegangan analog yang sebanding dengan kadar alkohol yang diterima.[6]

II. DESAIN SISTEM DAN IMPLEMENTASI

Perancangan perangkat keras pada sistem ini:

A. Base Robot

Pada sistem ini dibuat base robot yang nantinya akan menjadi dasar bentuk robot. Base robot ini dibuat menggunakan bahan aluminium dan acrylic. Berikut adalah desain base robot:



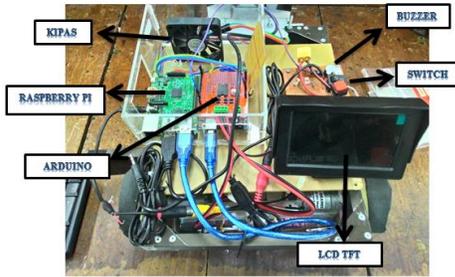
Gambar 3. 1 Desain base robot

B. Kamera

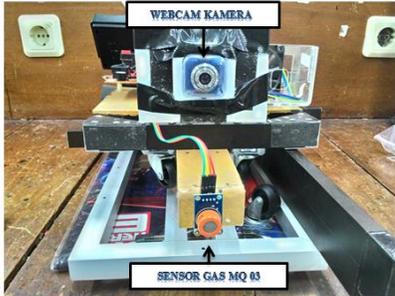
Kamera disini digunakan untuk menangkap gambar dari obyek yang dituju. Kamera yang digunakan pengaturan fokusnya dapat diubah dan tidak memiliki fitur auto fokus. Kamera ini terhubung dengan Raspberry Pi sebagai *processing unit* untuk mengolah data gambar melalui kabel USB. Kamera yang digunakan adalah kamera webcam standar dengan spesifikasi minimum :

- Resolution : 1920 x 1080 pixels
- Frame Rate : 30 fps
- Computer Interface : USB 2.0

Bentuk keseluruhan robot dapat seperti berikut :



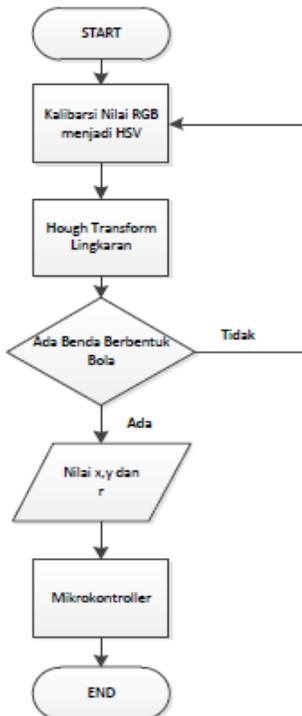
Gambar 3. 2 Bagian Robot dari Atas



Gambar 3. 3 Bagian Robot tampak depan

C. Perancangan Perangkat Lunak Raspberry Pi

Perancangan program untuk *Single Board Computer* yang akan dirancang adalah *image processing* untuk mendapatkan warna dari target yang dituju dan software yang digunakan adalah *OpenCV*.

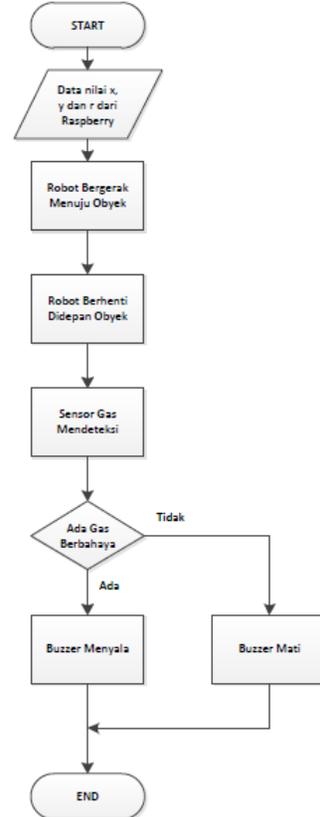


Gambar 3. 4 Flowchart Deteksi Bola

D. Perancangan Perangkat Lunak Mikrokontroler

Mikrokontroler Arduino menggunakan bahasa pemrograman C dan dapat diprogram menggunakan Arduino IDE. Seperti yang dijelaskan sebelumnya, Arduino berfungsi

untuk menerima data dari Raspberry Pi yang kemudian diolah untuk mengatur pergerakan dari robot.



Gambar 3. 5 Flowchart Gerakan Robot

III. ANALISIS DAN SIMULASI

A. Pengujian Sensor Gas

Pengujian ini bertujuan untuk mengetahui seberapa jauh sensor gas yang digunakan dapat mendeteksi dan sensitivitas dari sensor tersebut.

Tabel 4. 1 Tabel Pengujian Sensor Gas

Jarak	Tegangan
1 cm	2.7 v
2 cm	2 v
3 cm	1.7 v
4 cm	1.4 v
5 cm	1.4 v
6 cm	1.3 v
7 cm	1.1 v

Pengujian dilakukan dengan jarak yang berbeda dan menggunakan jenis cairan yang sama yaitu Alkohol 70%. Dari hasil data yang didapatkan, yaitu jika gas berada pada jarak 1 cm di depan sensor gas tegangan yang terbaca sebesar 2.7 volt. Sedangkan apabila gas berada pada jarak 5 cm didepan gas, tegangan yang terbaca 1.4 volt. Hal ini dapat disimpulkan bahwa semakin jauh gas yang terbaca maka besarnya tegangan yang keluar semakin kecil. Oleh karena itu sensor gas harus tepat diletakkan didepan sumber gas dengan jarak kurang lebih 5 cm.

B. Pengujian Driver Motor

Pengujian driver motor dengan cara memberi supply 12 V dan mengetahui seberapa besar tegangan yang dikeluarkan apabila nilai PWM berubah-ubah.

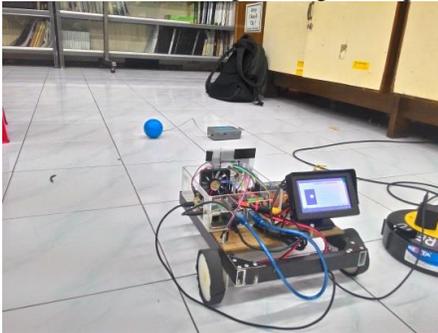
Tabel 4. 2 Tabel PWM terhadap Tegangan Output

PWM	Output
19% duty cycle	1.6 V
24% duty cycle	2.8 V
27% duty cycle	3.9 V
31% duty cycle	4.8 V
39% duty cycle	5.7 V
58% duty cycle	8.5 V
78% duty cycle	9.5 V

Hasil yang didapat, yaitu jika PWM bernilai 0, maka tegangan outputnya 0V, sedangkan apabila PWM yang diberikan sebesar 19% dari *duty cycle*, tegangan outputnya sebesar 1.6 V. Pada saat PWM bernilai 39% dari *duty cycle*, tegangan outputnya sebesar 6.3 V, dan apabila PWM yang diberikan sebesar 78% dari *duty cycle*, tegangan outputnya sebesar 9.5 V. Dari data yang didapatkan diketahui bahwa semakin besar PWM yang digunakan maka tegangan yang digunakan untuk menggerakkan motor juga semakin besar sehingga kecepatan pada motor DC akan semakin cepat. Tetapi pada motor yang digunakan kecepatan maksimum yang dicapai yaitu sebesar 38rpm sehingga apabila PWM mencapai maksimum yaitu 100% *duty cycle* maka kecepatan yang dicapai adalah sebesar 38rpm.

C. Pengujian Deteksi Benda

Pada pengujian ini akan dilakukan pendeteksian benda yang berupa bola tetapi menggunakan warna yang berbeda-beda. Pengujian ini dilakukan untuk mengetahui nilai Threshold HSV untuk tiap masing-masing warna.



Gambar 4. 1 Pengujian Nilai Hue Dari Obyek

Tabel 4. 3 Tabel Pengujian Deteksi Warna

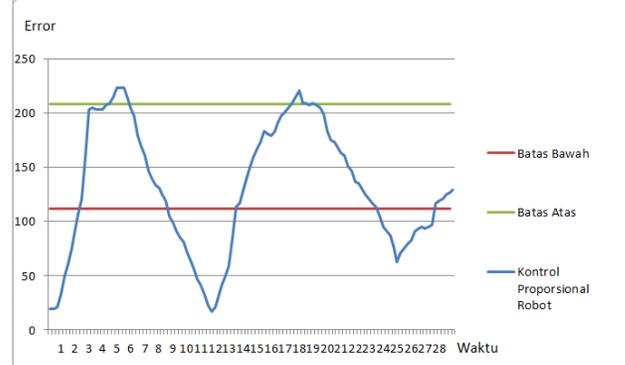
Warna	Nilai Hue
Orange	0 - 22
Kuning	22 - 38
Hijau	38 - 75
Biru	75 - 130
Violet	130 - 160
Merah	160 - 179

Dari hasil pengujian yang dilakukan untuk mendapatkan nilai warna sesuai dengan HSV maka yang harus diatur adalah besaran nilai *Hue* sedangkan *Saturation* (*S*) dan *Value* (*V*) bergantung dengan pencahayaan. Apabila kondisi ketika cahaya cukup maka nilai *Saturation* dan *Value* relatif kecil

sedangkan apabila kondisi cahaya redup nilai *Saturation* dan *Value* relatif tinggi untuk mengatur sensitivitas warna terhadap ruangan. Selain itu pemantulan cahaya juga berpengaruh sehingga apabila cahaya yang dipantulkan sangat banyak maka lantai juga akan menimbulkan pantulan pada obyek yang dituju sehingga di kamera akan terlihat dua obyek yang memiliki nilai HSV yang sama dan hal ini dapat mempengaruhi dalam pendeteksian obyek.

D. Pengujian Kontrol Proporsional

Pada pengujian ini akan dilakukan kontrol proporsional pada sistem untuk memperkecil error. Error ini merupakan error posisi piksel pada frame yang diambil. Berikut adalah grafik hasilnya



Pada grafik dapat dilihat hasilnya yaitu ketika benda berada di sebelah kiri maka error yang dihasilkan relatif besar dibandingkan dengan apabila benda yang berada sebelah kanan. Apabila benda berada di sebelah kanan maka nilai error piksel diatas 160 dan apabila benda berada di sebelah kiri maka error yang dihasilkan berada dibawah 160. Error dibawah 160 lebih sulit untuk kembali daripada saat error diatas 160. Berarti dapat disimpulkan robot ini sulit untuk kembali saat berbelok ke kiri daripada belok kanan. Disini menggunakan batas atas sebesar 208 dan batas bawah 112. Hal ini dilakukan untuk kompensasi error saat robot mendekati benda. Apabila robot mendekati pada benda maka gambar dari obyek yang di deteksi akan menjadi besar sehingga posisi benda tidak lagi tepat di tengah. Hal ini untuk mengatasi kekeliruan apabila benda dianggap tidak berada di tengah. Pada pengujian ini terdapat error rata-rata yaitu sebesar 34,806%.

E. Pengujian Keseluruhan Sistem

Pada pengujian ini akan dilakukan pengujian keseluruhan sistem untuk mengetahui apakah sistem ini mencapai tujuan yang diberikan yaitu menuju pada benda yang ditentukan dengan menggunakan algoritma *CamShift* dan mampu mendeteksi kebocoran gas pada benda tersebut.

Tabel 4. 4 Tabel Pengujian Sistem

Pengujian	Hasil
Pengujian 1	Berhasil
Pengujian 2	Gagal
Pengujian 3	Gagal
Pengujian 4	Berhasil
Pengujian 5	Berhasil

Dari hasil pengujian yang dilakukan maka dapat disimpulkan bahwa robot mampu menuju benda yang dituju dan mendeteksi kebocoran gas alkohol pada benda tersebut. Akan tetapi masih terdapat error yaitu sekitar 40% dari total percobaan yang dilakukan

IV. KESIMPULAN

Kesimpulan yang diperoleh dalam penelitian ini adalah:

- Robot mampu mendeteksi gas alkohol pada benda yang berjarak kurang lebih 3cm.
- Untuk dapat memaksimalkan Raspberry Pi maka digunakan image processing dengan lebar frame yang kecil sehingga dapat melakukan pemrosesan gambar meskipun tidak maksimal dikarenakan memori yang digunakan pada Raspberry Pi kecil.
- Penggunaan algoritma CamShift memudahkan untuk mendeteksi obyek dengan warna yang berbeda dengan menyesuaikan histogram yang dibentuk saat menentukan ROI (Region Of Interest).
- Kontrol Proporsional yang digunakan pada robot mampu membuat robot mencapai target dengan error rata-rata sebesar 34,806%.
- Sistem ini masih terdapat error sekitar 40% dari total percobaan yang dilakukan.

DAFTAR PUSTAKA

- [1] Benedictus Yoga Budi Putranto, Widi Hapsari, Katon Wijana, "Segmentasi Warna Citra Dengan Deteksi Warna HSV Untuk Mendeteksi Objek", Fakultas Teknik Program Studi Teknik Informatika Universitas Kristen Duta Wacana Yogyakarta, 2010.
- [2] Resmana Lim, Yulia Roy Otniel Pantouw, "Pelacakan dan Pengenalan Wajah Menggunakan Webcam & Metode Gabor Filter", Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra, ____
- [3] Kurnia Khafidhatur Rafiah, Rita Magdalena, Ir., M.T., Nur Andini, S.T., MT., "Deteksi Rambu-Rambu Lalu Lintas Untuk Membantu Pengguna Jalan Raya Dengan Algoritma CamShift", Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung, ____
- [4] Hani Avrilyantama, Muhammad Rivai, Djoko Purwanto, 2015. Pengembangan Robot Hexapod Untuk Melacak Sumber Gas. JURNAL TEKNIK ITS Vol. 4, No. 1, pp. 1-6
- [5] Abi Nawang Gustica, Muhammad Rivai, Tasripan, 2014. Implementasi Sensor Gas pada Kontrol Lengan Robot untuk Mencari Sumber Gas. JURNAL TEKNIK POMITS Vol. 3, No. 1, pp.F.90-95
- [6] Dimas Herjuno, Muhammad Rivai, Tri Arief Sardjono, 2012. Teledeteksi Gas pada Mobile Robot yang Dikendalikan Gelombang Radio. JURNAL TEKNIK POMITS Vol. 1, pp. A.160-163
- [7] Harianto, Muhammad Rivai, Djoko Purwanto, 2013. Implementation of Electronic Nose in Omni-directional Robot. International Journal of Electrical and Computer Engineering (IJECE), Vol. 3, No. 3, June 2013, pp. 399-406

PERANCANGAN ROBOT BERODA UNTUK MENDETEKSI GAS BERBAHAYA *MENGGUNAKAN IMAGE PROCESSING BERBASIS SINGLE BOARD COMPUTER*

HANIFAR KAHIRA

2211100161

Pembimbing :

Dr. Muhammad Rivai ST.,MT.

Rudy Dikairono ST.,MT.



Latar Belakang



Rumusan Masalah

- Bagaimana cara mengolah data dari sensor gas
- Bagaimana cara mengolah data kamera ke Raspberry Pi
- Bagaimana cara mengirim data dari Raspberry Pi ke Arduino.
- Bagaimana cara mengontrol motor menggunakan Machine Vision.

Batasan masalah

- Kamera yang digunakan USB Webcam.
- Single Board Computer yang digunakan Raspberry Pi b+.
- Mikrokontroller yang digunakan adalah Arduino UNO R3
- Motor yang digunakan adalah motor DC 12 Volt 38 rpm.
- Sensor gas yang digunakan adalah MQ-3.
- Data gambar yang dikirimkan diolah dengan Raspberry Pi b+
- Obyek yang dituju tidak terkena halangan atau *obstacle*.
- Berada di ruangan tertutup dengan cahaya yang cukup.

Tujuan dan Manfaat

Tujuan tugas akhir ini diharapkan mampu membuat robot pendeteksi suatu objek dan mengidentifikasi objek tersebut mengandung gas berbahaya atau tidak. Dengan adanya perancangan ini diharapkan dapat memberikan banyak manfaat dan mampu mencegah hal-hal yang tidak diinginkan seperti terkena penyakit dan terjadinya ledakan dan kebakaran.

Mengapa Menggunakan Metode Ini?

- ❑ Robot Beroda merupakan salah satu jenis robot yang menggunakan roda sebagai alat penggerak sehingga memiliki kelebihan dalam bermanuver dalam menjangkau tempat-tempat yang jauh. Selain itu robot beroda dapat menuju ke target dalam waktu yang relatif cepat.
- ❑ Raspberry Pi merupakan komputer papan tunggal yang memiliki ukuran sebesar kartu kredit. Sehingga ukurannya relatif kecil dan tidak sebesar komputer maupun laptop tetapi performa dari Raspberry Pi mampu untuk mengolah data dari kamera.
- ❑ Hough Transform digunakan karena mampu membentuk suatu deteksi berupa lingkaran yang relatif stabil. Hal ini dikarenakan Transformasi Hough menggunakan sistem voting piksel yaitu semakin banyak piksel yang tertangkap maka semakin mendekati bentuk lingkaran sempurna. Kelebihan Transformasi Hough adalah mampu mendeteksi bentuk geometri dengan kumpulan tepi yang terputus dan relatif tidak terpengaruh derau.
- ❑ Pendeteksian menggunakan lingkaran dikarenakan banyak benda berbahaya yang memiliki bentuk hampir mirip dengan lingkaran contohnya seperti granat. Selain itu dapat digunakan untuk mendeteksi kebocoran pada tabung karena dasar tabung berbentuk lingkaran.
- ❑ Metode transformasi HSV dipilih karena pengaruh cahaya sedikit berpengaruh. Metode HSV dapat diatur berdasarkan nilai dari Hue, Saturation, dan Value. Nilai pada setiap elemen ini berubah bergantung dari cahaya untuk nilai Saturation dan Value. Sedangkan untuk nilai Hue bergantung dari warna yang akan dideteksi.

Robot beroda

Robot mobil atau mobile robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik lain. Mobile robot adalah tipe robot yang paling populer dalam dunia penelitian robot. Sebutan ini biasa digunakan sebagai kata kunci utama untuk mencari rujukan atau referensi yang berkaitan dengan robotik di internet. Publikasi dengan judul yang berkaitan mobile robot sering menjadi daya tarik, tidak hanya bagi kalangan peneliti, tapi juga bagi kalangan awam. Dari segi manfaat, penelitian tentang berbagai tipe mobile robot diharapkan dapat membantu manusia dalam melakukan otomasi dalam transportasi, platform bergerak untuk robot industri, eksplorasi tanpa awak dan masih banyak lagi.



RASPBERRY PI

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal (Single Board Circuit /SBC) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memory yang digunakan, Model A menggunakan memory 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan ethernet port (kartu jaringan) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang.



Arduino Uno

Arduino Uno adalah *board* berbasis mikrokontroler pada ATmega 328. *Board* ini memiliki 14 digital *input* / output *pin* (dimana 6 *pin* dapat digunakan sebagai output PWM), 6 *input* analog, 16 MHz osilator kristal, koneksi USB, *jack* listrik dan tombol reset. *Pin – pin* ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya (Arduino, Inc., 2009).

Arduino Uno R3 berbeda dengan semua *board* sebelumnya karena *Arduino Uno R3* ini tidak menggunakan *chipdriver* FTDI USB-to-serial. Melainkan menggunakan fitur dari ATmega 16U2 yang diprogram sebagai konverter *USB-to-serial*.



Driver Motor Ardumoto

Ardumoto adalah shield driver motor Arduino yang akan mengontrol 2 motor DC. Driver ini menggunakan L-298P H-Bridge yang mampu meng-drive motor hingga 2 A untuk tiap channelnya. Output kontrol motor 1/2 terhubung dengan pin 12 (Direction A) dan pin 3 (PWM A). Sedangkan output kontrol motor 3/4 terhubung dengan pin 13 (Direction B) dan pin 11 (PWM B). Vin pada shield Ardumoto terhubung secara langsung dengan Vin pada Arduino sehingga harus berhati-hati dalam menyambungkan jangan sampai supply dari kedua board terjadi *short*.



Sensor Gas MQ 03

Sensor gas MQ 03 ini mampu mendeteksi gas yang mengandung alkohol. Selain itu sensor ini juga dapat digunakan untuk mendeteksi kadar alkohol pada mulut sehingga dapat mengetahui kondisi orang tersebut mabuk atau tidak. Sensor ini mempunyai 2 keluaran dapat berupa digital maupun analog.

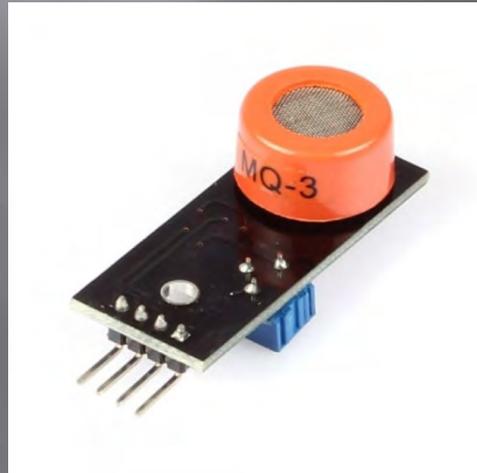


Image Processing

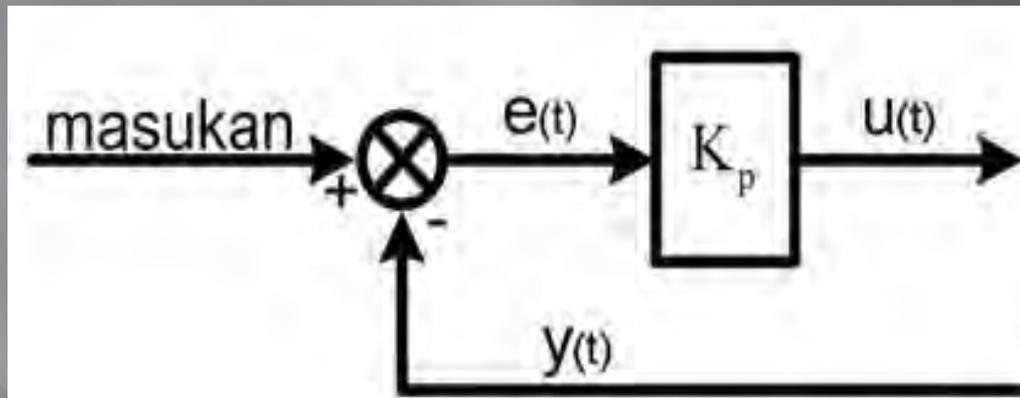
Metode yang digunakan pada image processing yaitu merubah dari RGB (Red, Green, Blue) menjadi HSV (Hue, Saturation, Value). Hue adalah apa yang biasanya kita sebut sebagai 'warna' dalam bahasa sehari-hari. Istilah 'hue' berarti kombinasi dari warna-warna dasar, dengan kata lain, merah, hijau, biru atau kuning (RGB). Saturation menunjukkan intensitas dari hue. Warna-warna dasar yang terang adalah warna dengan saturation tinggi, sementara warna-warna pastel saturation-nya rendah. Monochrome (hitam dan putih) seluruhnya tidak memiliki saturation karena tidak punya intensitas warna di dalamnya. Value berhubungan dengan tajam atau tidaknya sebuah warna – atau tingkat hitam atau putih pada skala warna. Sebuah warna dengan value yang rendah berarti lebih dekat dengan hitam, sementara yang memiliki value tinggi lebih dekat dengan putih.

Algoritma *CamShift*

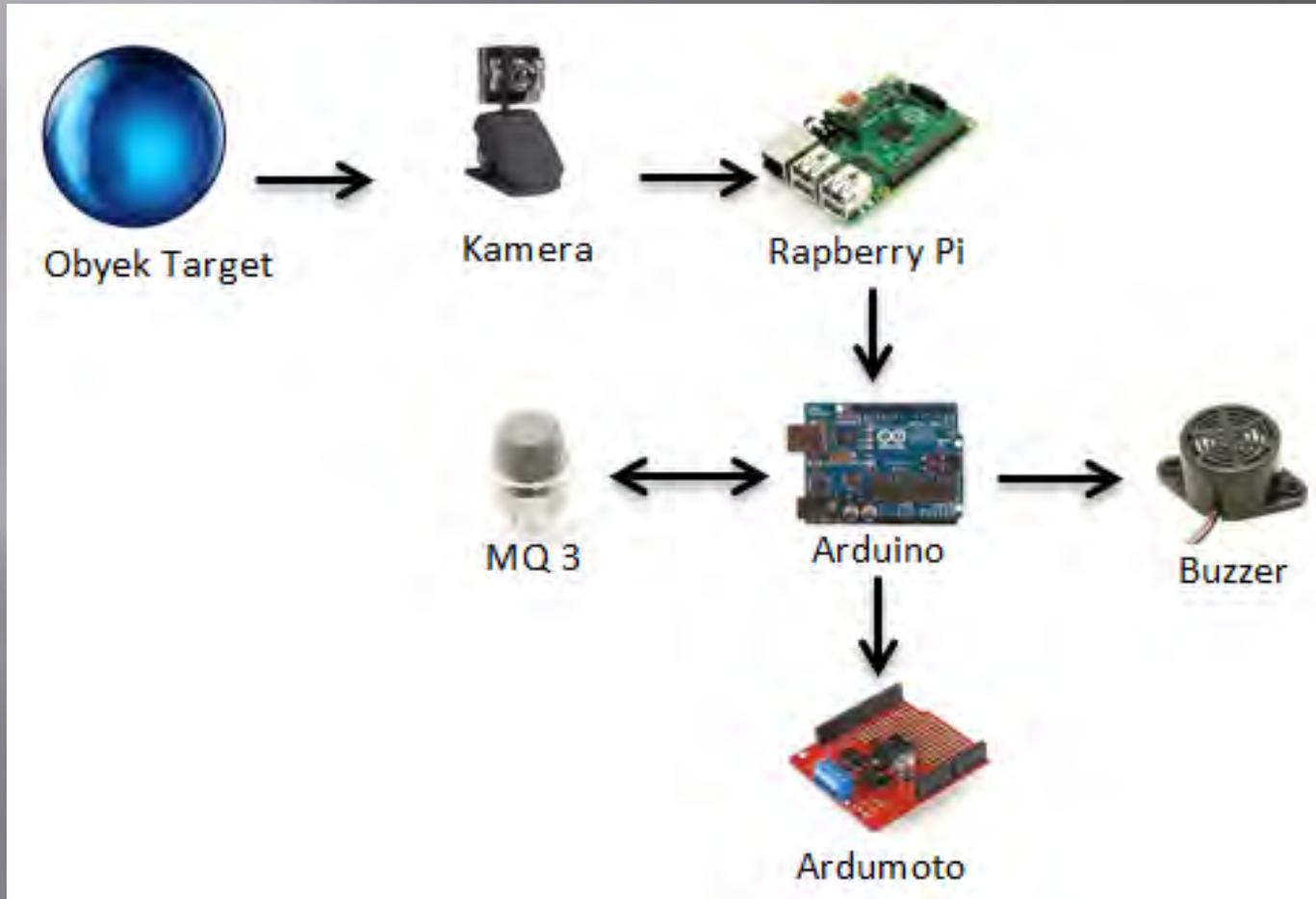
- ❑ CamShift merupakan singkatan dari *Continuously Adaptive Mean-Shift*. Algoritma CamShift merupakan pengembangan dari algoritma dasar Mean-Shift yang dilakukan secara berulang untuk dapat melacak pergerakan dari obyek. Algoritma Mean-Shift beroperasi berdasarkan distribusi probabilitas dari citra. Untuk melacak citra berwarna pada video sequence, maka citra berwarna tersebut harus direpresentasikan dalam bentuk distribusi probabilitas citra dengan menggunakan distribusi histogram dari citra tersebut. Distribusi warna dari video sequence selalu berubah tiap waktu, maka algoritma Mean-Shift harus dikembangkan supaya dapat digunakan untuk melacak obyek. Perkembangan dari algoritma ini dinamakan algoritma CamShift.
- ❑ Tahap pertama dari algoritma CamShift adalah dengan melakukan penghitungan terhadap nilai hue dari obyek sampel. Nilai histogram hue sebagai hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai lookup table.
- ❑ Tahap kedua, dilakukan pemilihan lokasi awal dari Mean-Shift 2D search window. Kemudian dilakukan penghitungan nilai histogram hue yang menjadi nilai probabilitas untuk tiap-tiap pixel pada citra.
- ❑ Tahap ketiga, dijalankan algoritma Mean-Shift untuk mencari pusat dan besar search window yang baru. Lokasi tengah dan besar daerah yang dihasilkan disimpan. Lalu dilakukan looping ke tahap dua dimana lokasi awal dan search window menggunakan hasil algoritma Mean-Shift pada tahap tiga.

Kontrol Proporsional

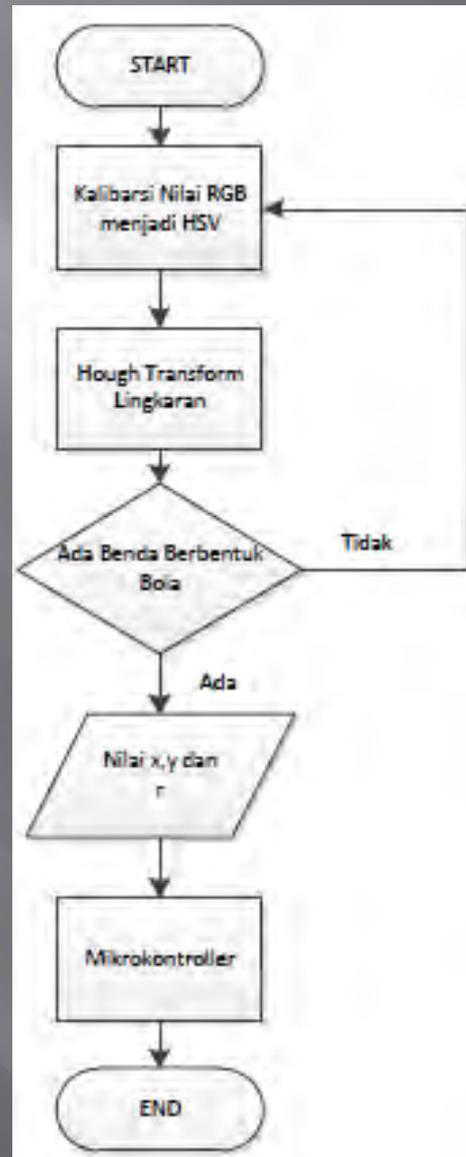
- Secara lebih sederhana dapat dikatakan, bahwa keluaran controller proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.



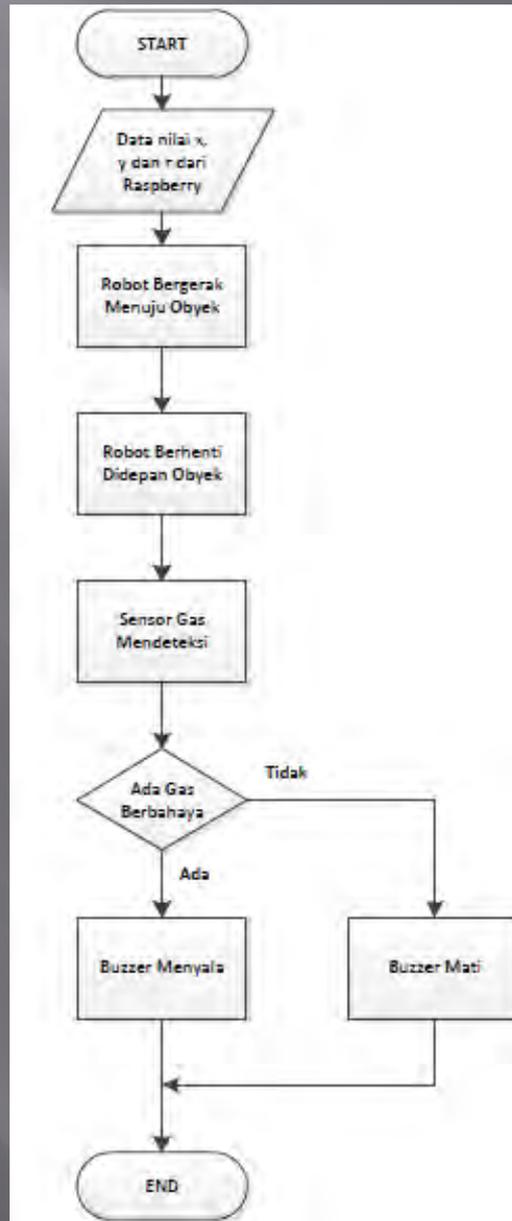
CARA KERJA SISTEM



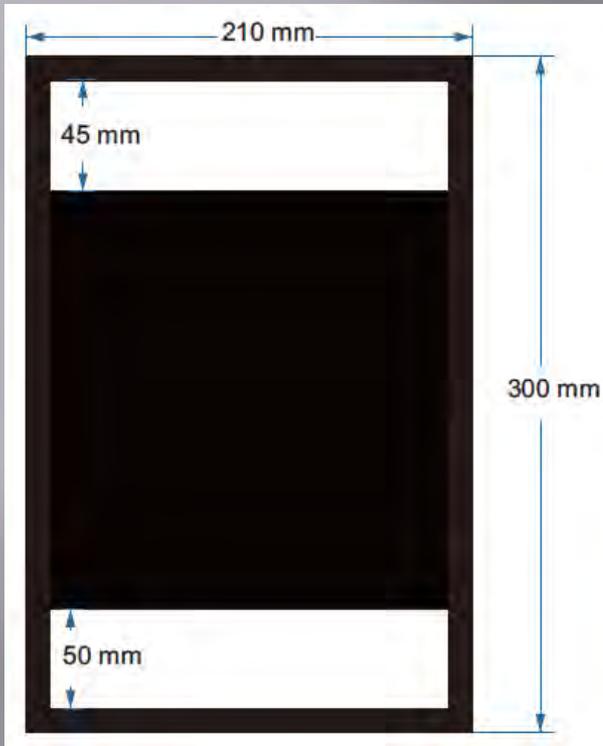
Flowchart Deteksi Benda



Flowchart Gerakan Robot



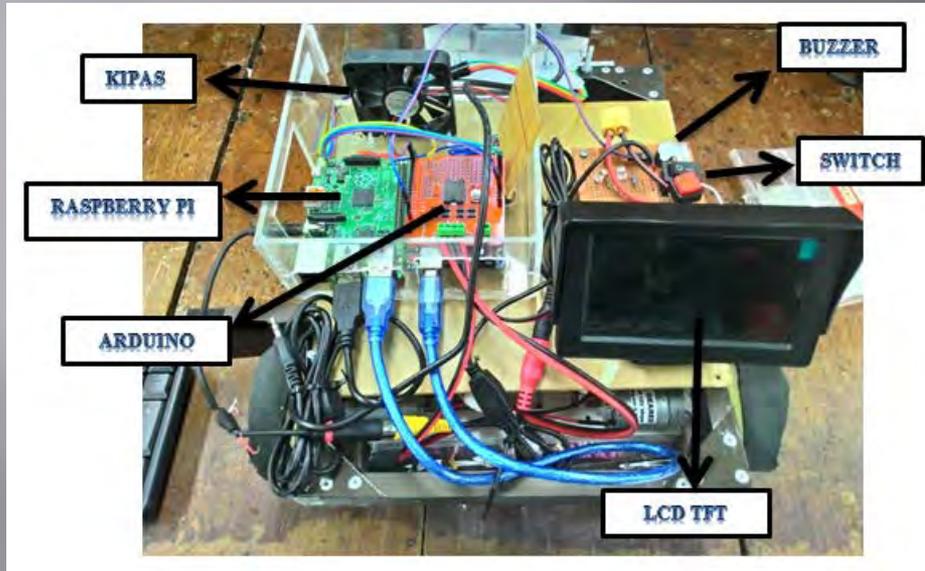
Desain Base Robot



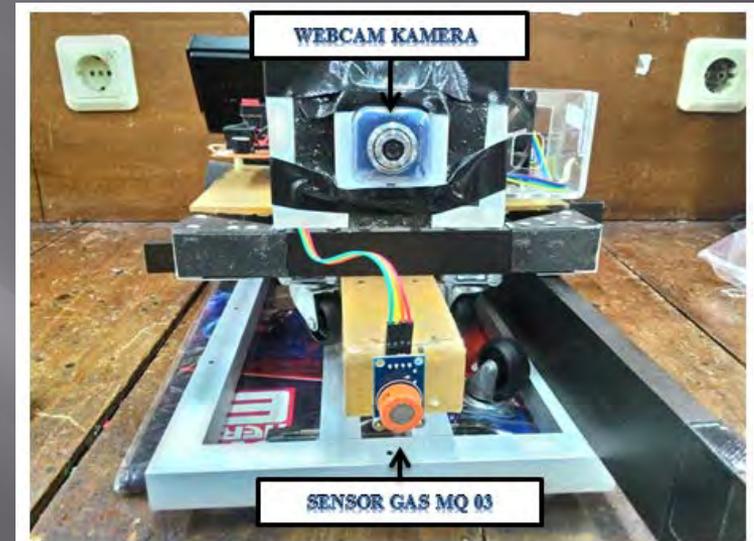
Mekanik robot dibuat menggunakan bahan dasar alumunium dan arcrylic. Bahan alumunium dipilih dikarenakan kuat dan lebih ringan dari besi. Sedangkan arcrylic digunakan untuk menjadi bahan base pada elektronik robot.

Motor yang digunakan adalah motor DC 38rpm dengan torsi sebesar 7.8kg. Motor ini mampu menahan beban mendekati 8kg tetapi kecepatannya rendah yaitu sebesar 38rpm. Robot ini menggunakan dua roda penggerak dan dua roda bebas. Motor ini menggunakan supply baterai Li-Po 1300mAH 12V. Baterai ini juga di paralel dengan LCD TFT untuk menampilkan Raspberry Pi. Sedangkan Raspberry Pi menggunakan supply Power Bank 11000mAH 5V.

BENTUK MEKANIK ROBOT



Tampak Atas



Tampak Depan

DETEKSI BOLA MENGGUNAKAN TRANSFORMASI RGB KE HSV & ALGORITMA *CAMSHIFT*

The image displays a software interface for object tracking. On the left, a window titled "CamShiftDemo" features three vertical sliders for parameter adjustment: "Vmin: 10", "Vmax: 256", and "Smin: 132". Below these sliders is a video feed showing a person holding a red ball, which is highlighted with a red circle. On the right, a "Histogram" window shows a single red vertical bar on a black background, representing the color distribution of the tracked object. The background of the right side shows a code editor with a "Call Stack" window at the bottom.

Posisi Bola

The screenshot displays a software interface for object tracking. On the left, a terminal window shows the following output:

```
x=256 y=168  
lebar=89  
track_object1  
x=256 y=168  
lebar=89  
track_object1  
x=257 y=168  
lebar=89  
track_object1  
x=257 y=168  
lebar=89  
track_object1  
x=257 y=168  
lebar=89  
track_object1  
x=257 y=168  
lebar=89  
track_object1  
x=258 y=168  
lebar=89  
track_object1  
x=257 y=168  
lebar=89  
track_object1  
x=257 y=
```

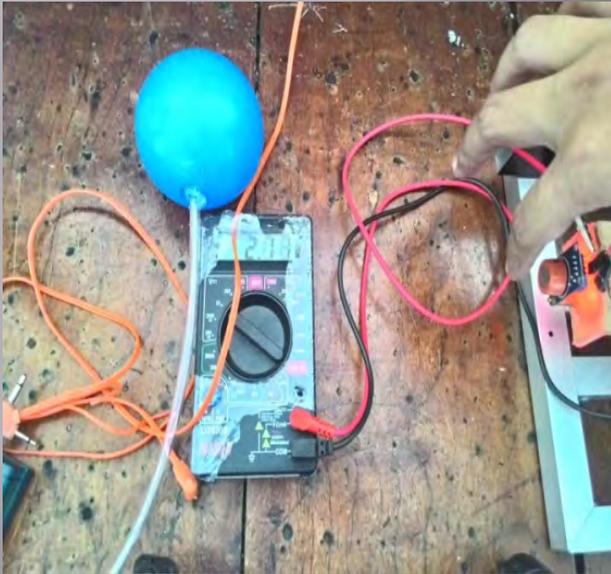
The central window, titled "CamShiftDemo", features three sliders for parameter adjustment:

- Vmin: 10
- Vmax: 256
- Smin: 132

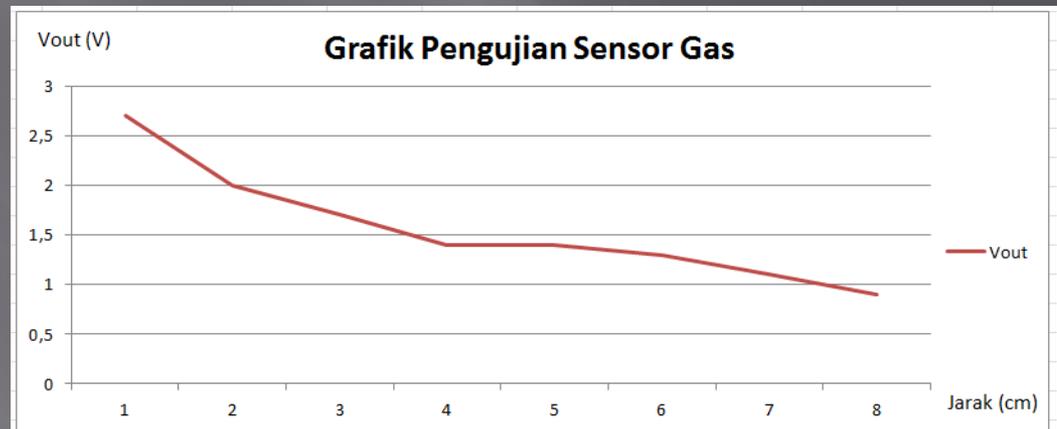
Below the sliders is a video feed showing a person holding a red ball, which is circled in red. To the right, a "Histogram" window is partially visible.

Pengujian dan Analisis

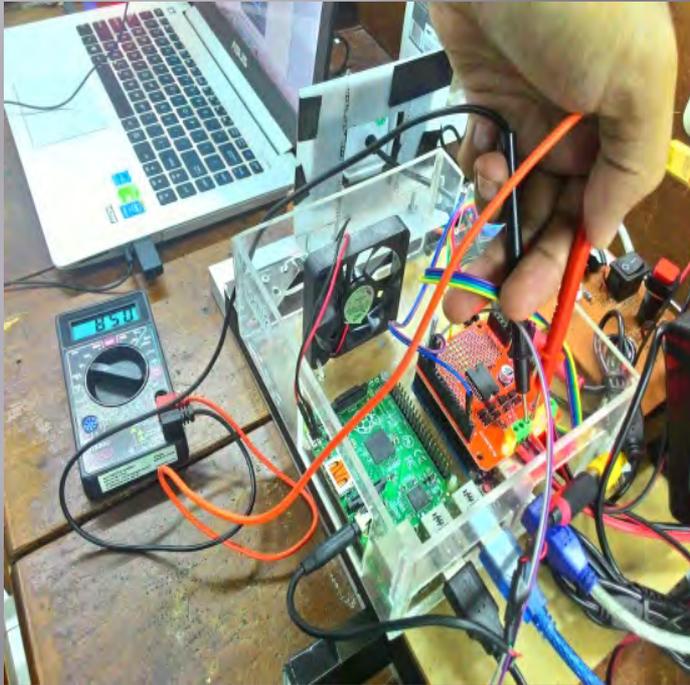
Uji Sensor Gas



Jarak	Tegangan
1 cm	2.7 v
2 cm	2 v
3 cm	1.7 v
4 cm	1.4 v
5 cm	1.4 v
6 cm	1.3 v
7 cm	1.1 v
8 cm	0.9 v

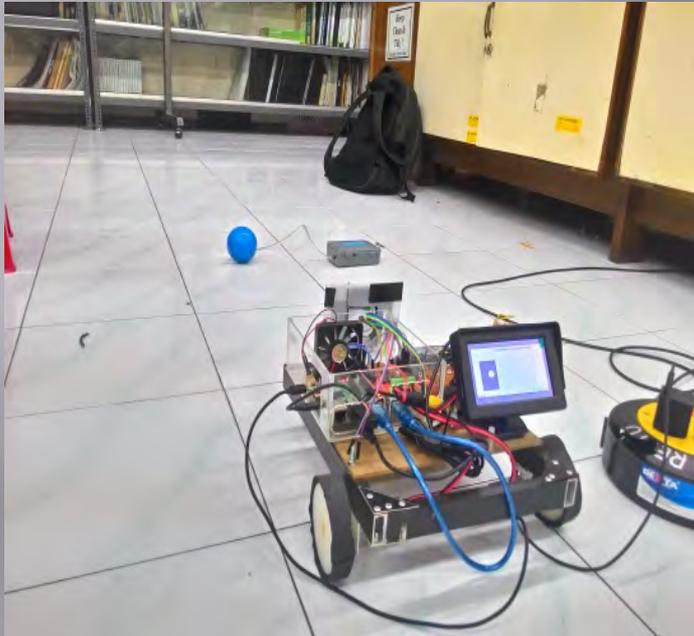


Uji Driver Motor



PWM	Output
19% duty cycle	1.6 V
24% duty cycle	2.8 V
27% duty cycle	3.9 V
31% duty cycle	4.8 V
39% duty cycle	5.7 V
58% duty cycle	8.5 V
78% duty cycle	9.5 V

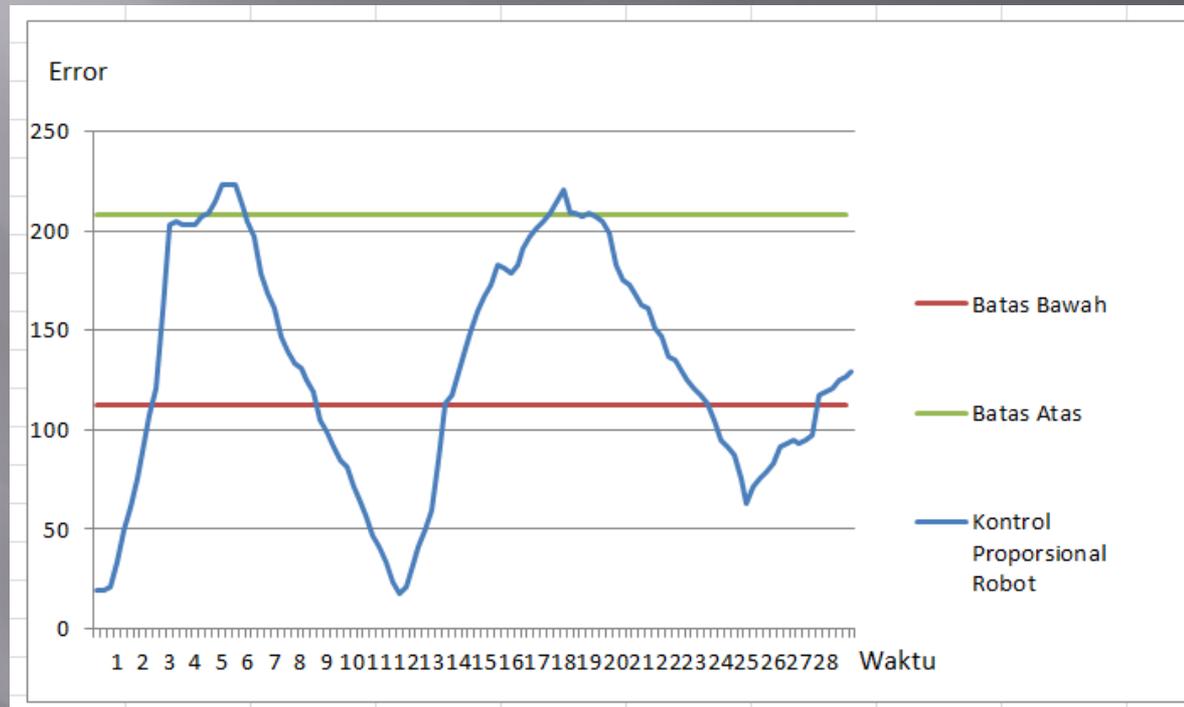
Pengujian Deteksi Bola



Warna	Nilai Hue
Orange	0 - 22
Kuning	22 - 38
Hijau	38 - 75
Biru	75 - 130
Violet	130 - 160
Merah	160 - 179

Dari hasil pengujian didapatkan untuk warna yang berbeda memiliki nilai *Hue* yang berbeda sedangkan nilai *Saturation* dan *Value* berubah bergantung pada cahaya pada lingkungan sekitar.

Uji Kontrol Proporsional



Dari grafik diatas dapat dilihat bahwa robot memiliki masalah saat error dibawah batas bawah yang berarti benda yang di deteksi berada di sebelah kiri dari frame. Robot memiliki masalah saat berbelok ke kiri hal ini dikarenakan desain mekanik yang mungkin tidak sama dan relatif berat

UJI KESELURUHAN SISTEM

PERCOBAAN	HASIL
Percobaan 1	Berhasil
Percobaan 2	Gagal
Percobaan 3	Gagal
Percobaan 4	Berhasil
Percobaan 5	Berhasil

Pada pengujian keseluruhan sistem robot mampu mendeteksi kebocoran gas pada benda pada jarak kurang lebih 1-3 cm. Hal ini dikarenakan sensor gas yang digunakan mampu mendeteksi gas dengan jarak yang relatif pendek yaitu sekitar 1cm hingga 7cm.

KESIMPULAN

- Penggunaan algoritma *CamShift* memudahkan untuk mendeteksi obyek dengan warna yang berbeda dengan menyesuaikan histogram yang dibentuk saat menentukan ROI (*Region Of Interest*).
- Robot mampu mendeteksi gas alkohol pada benda yang berjarak kurang lebih 3cm.
- Untuk dapat memaksimalkan Raspberry Pi maka digunakan image processing dengan lebar frame yang kecil sehingga dapat melakukan pemrosesan gambar meskipun tidak maksimal dikarenakan memori yang digunakan pada Raspberry Pi kecil.
- Kontrol Proporsional yang digunakan pada robot mampu membuat robot mencapai target dengan error rata-rata sebesar 34,806%.
- Sistem ini masih terdapat error sekitar 40% dari total percobaan yang dilakukan hal ini dikarenakan oleh banyak faktor seperti desain mekanik dan pemrosesan pada Single Board Computer.

TERIMA KASIH