PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

TUGAS AKHIR

ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik pada
Program Studi Teknik Elektro
Fakultas Sains dan Teknologi Universitas Sanata Dharma

disusun oleh :

GILANG WASTUNARENDRA
NIM : 215114039

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2023

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

FINAL PROJECT

AUTOMATIC LOCATION TRACKING ROBOT BASED ON RASPBERRY PI

In a partial fulfilment of the requirements
for the degree of Sarjana Teknik
Department of Electrical Engineering
Faculty of Science and Technology, Sanata Dharma University

GILANG WASTUNARENDRA
NIM : 215114039

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF SCIENCE AND TECHNOLOGY
SANATA DHARMA UNIVERSITY
2023

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LEMBAR PERSETUJUAN

TUGAS AKHIR

ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI

(AUTOMATIC LOCATION TRACKING ROBOT BASED ON RASPBERRY PI)

Oleh :

GILANG WASTUNARENDRA

NIM : 215114039

telah disetujui oleh :

Pembimbing I

Dr. Ir.Bernadeta Wuri Harini Tanggal : 21 November 2023

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LEMBAR PENGESAHAN

TUGAS AKHIR

ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI

(AUTOMATIC LOCATION TRACKING ROBOT BASED ON RASPBERRY PI)

disusun oleh :

GILANG WASTUNARENDRA

NIM : 215114039

Telah dipertahankan di depan tim penguji
pada tanggal 20 Oktober 2023
dan dinyatakan memenuhi syarat

Susunan Tim Penguji :

	Nama Lengkap	Tanda Tangan
Ketua	: Ir. Petrus Setyo Prabowo, M. T.	
Sekretaris	: Dr. Ir.B. Wuri Harini	
Anggota	: Ir. Martanto, M. T.	

Yogyakarta, 20 Oktober 2023
Fakultas Sains dan Teknologi
Universitas Sanata Dharma
Dekan,

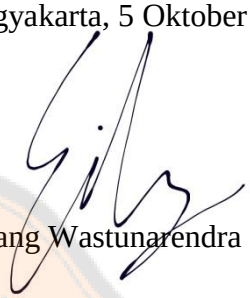
Ir. Drs. Haris Sriwindono, M.Kom., Ph.D.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa tugas akhir ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, 5 Oktober 2023



Gilang Wastunarendra

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

HALAMAN PERSEMBAHAN DAN MOTTO HIDUP

MOTTO:

“Mandiri dalam Bekerja
Merdeka dalam Berkarya”

Skripsi ini saya persembahkan untuk...
Tuhan Yang Maha Esa,
Negara Kesatuan Republik Indonesia,
Universitas Sanata Dharma,
Orang tua Saya,
Teman-teman dan orang-orang spesial yang sudah menemani saya sampai sejauh ini

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGANAKADEMIS

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma :

Nama : GILANG WASTUNARENDRA

Nomor Mahasiswa : 215114039

Demi pengembangan ilmu pengetahuan, saya memberikan kepada Perpustakaan Universitas Sanata Dharma karya ilmiah saya yang berjudul :

ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI

beserta perangkat yang diperlukan (bila ada). Dengan demikian saya memberikan kepada Perpustakaan Universitas Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas, dan mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Yogyakarta, 5 Oktober 2023

(GILANG WASTUNARENDRA)

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

INTISARI

Pengerjaan tugas akhir ini ditujukan untuk meneliti kemampuan robot beroda dalam melacak lokasi secara otomatis. Proyek tugas akhir ini mengambil inspirasi dari algoritma robot dari divisi VTOL pada perlombaan KRTI (Kontes Robot Terbang Indonesia). Robot pada divisi tersebut mampu memetakan area secara otomatis dengan menggunakan mini-pc sebagai pengendali dan kamera serta LIDAR sebagai perangkat pendeteksi objek atau area. Dalam divisi VTOL KRTI robot yang dibuat harus bisa mengantarkan lokasi ke tiga gedung sesuai dengan urutan gedung yang diminta.

Bentuk penyesuaian yang dilakukan pada robot tugas akhir ini adalah dengan menggunakan Raspberry Pi sebagai pengendali didukung dengan kamera sebagai pelacak lokasi dengan mendeteksi warna gedung dan membaca kode QR di atas gedung. Penggerak robot menggunakan empat roda yang arah geraknya diatur menggunakan metode pengacakan *delay time*. Untuk mencegah robot menabrak objek maka ditambahkan sensor ultrasonik. Adapun algoritma gerakan robot akan mengikuti algoritma divisi VTOL KRTI.

Dari hasil pengujian, kamera mampu mendeteksi warna yang diminta dan kode QR dengan keberhasilan pembacaan tertinggi pada kode QR VTOL2 dengan nilai 60% pada ketinggian 50cm. Gerakan robot menghasilkan *error* yang tinggi di setiap pengujian kesepuluh disebabkan kapasitas baterai yang menurun 7% di setiap pengujian. Robot masih kesulitan dalam memastikan titik berhenti dan batas jarak pada gedung ketika menjalankan pola gerakan karena kamera dan sensor ultrasonik belum dapat diintegrasikan dalam mengontrol motor.

Kata kunci: robot, Raspberry Pi, kamera, sensor ultrasonik, divisi VTOL KRTI.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

ABSTRACT

This final project is aimed at examining the ability of wheeled robots to track locations automatically. This final project takes inspiration from the robot algorithm from the VTOL division at the KRTI (Indonesian Flying Robot Contest) competition. The robot in the division is able to map the area automatically by using a mini-pc as a controller and camera and LIDAR as an object or area detection device. In the VTOL division of KRTI, the robot must be able to deliver locations to three buildings in accordance with the order of the building requested.

The form of adjustment made to this final project robot is to use Raspberry Pi as a controller supported by a camera as a location tracker by detecting the color of the building and reading the QR code on the building. The robot driver uses four wheels whose motion direction is set using the delay time randomization method. To prevent the robot from crashing into objects, an ultrasonic sensor is added. The robot movement algorithm will follow the KRTI VTOL division algorithm.

From the test results, the camera is able to detect the requested color and QR code with the highest reading success on the VTOL2 QR code with a value of 60% at a height of 50cm. The robot movement produces a high error in every tenth test due to the battery capacity which decreases 7% in each test. The robot still has difficulty in ensuring the stopping point and distance limit on the building when running the movement pattern because the camera and ultrasonic sensor cannot be integrated in controlling the motor.

Keywords: robot, Raspberry Pi, camera, ultrasonic sensor, VTOL division KRTI.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

KATA PENGANTAR

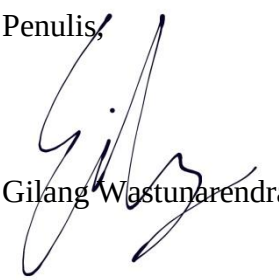
Puji syukur kepada Tuhan Yang Maha Esa atas terselesaikannya tugas akhir ini. Banyak pengalaman dan pengetahuan baru yang didapatkan penulis selama pengerjaan tugas akhir ini. Penulisan tugas akhir ini mampu terselesaikan juga karena dukungan banyak pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Bapak Ir. Drs. Haris Sriwindono, M.Kom., Ph.D. selaku Dekan Fakultas Sains dan Teknologi Universitas Sanata Dharma.
2. Bapak Ir. Martanto, M.T. selaku kepala Program Studi Teknik Elektro.
3. Ibu Ir. Bernadeta Wuri Harini selaku Dosen Pembimbing Tugas Akhir yang sudah membimbing saya sampai terselesaikannya tugas akhir ini.
4. Bapak Dr. Ir. Iswanjono selaku Dosen Pembimbing Akademik.
5. Seluruh dosen pengampu matakuliah Program Studi Teknik Elektro, FST, USD.
6. Bapak, Ibu, dan Adik saya yang selalu memberikan dukungan dan semangat.
7. Teman-teman transfer dari Mekatronika PMSD dan teman-teman Teknik Elektro angkatan 19, 20, dan 21.
8. Seluruh pihak yang tidak bisa disebutkan satu persatu yang telah mendukung dan membantu dalam banyak hal.

Penulis menyadari bahwa penulisan tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat menerima kritik dan saran dari berbagai pihak yang mampu memberi perbaikan dan pengembangan tugas akhir ini. Semoga tugas akhir ini mampu memberikan manfaat bagi pembaca, terima kasih. Salam.

Yogyakarta,5 Oktober 2023

Penulis,



Gilang Wastunarendra

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Daftar Isi

LEMBAR PERSETUJUAN iii
LEMBAR PENGESAHAN iv
PERNYATAAN KEASLIAN KARYA v
HALAMAN PERSEMBAHAN DAN MOTTO HIDUP vi
LEMBAR PERNYATAAN PERSETUJUAN PULIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS vii
INTISARI viii
ABSTRACT ix
KATA PENGANTAR x
DAFTAR ISI xi
DAFTAR GAMBAR xiv
DAFTAR TABEL xvi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Tujuan dan Manfaat 2
1.3 Batasan Masalah 3
1.4 Metodologi Penelitian 3
BAB II DASAR TEORI 5
2.1 Robot Beroda 5
2.2 Raspberry Pi 7
2.2.1 Raspberry Pi 4 B 7
2.2.2 GPIO Raspberry Pi 4 B 8
2.2.2 Bahasa Pemrograman Phyton 9
2.3 Motor DC 10
2.3.1 Gearbox Motor DC 11
2.3.2 *Driver* Motor DC 12
2.3.3 *Driver* Motor DC L298N 13
2.3.4 PWM(*pulse-width-modulation*) 14
2.3.5 Hubungan Duty Cycle PWM dengan Keluaran Tegangan 15
2.4 Kamera Webcam 16
2.5 Sensor Ultrasonik 17

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.5.1 Modul Sensor Ultrasonik HC-SR04 18
2.6 Sensor Kompas 19
2.6.1 Modul Sensor Ultrasonik HC-SR04 19
BAB III RANCANGAN PENELITIAN 20
3.1 *Block Diagram* 20
3.2 Perancangan Mekanik 21
3.3 Perancangan Perangkat Keras 23
3.3.1 Gearbox Motor DC 23
3.3.1.1 Metode Pengacakan *Delay Time Value* 24
3.3.2 Kamera *Webcam* 25
3.3.2.1 Pembacaan Kode QR 25
3.3.2.2 Pembacaan Warna 26
3.3.3 Modul Sensor Ultrasonik HC-SR04 26
3.3.4 Modul Sensor Kompas HMC5883L 27
3.4 Perancangan Elektronik 27
3.4.1 Perancangan Elektronik *Driver* Motor DC 28
3.4.2 Perancangan Elektronik Sensor Ultrasonik 31
3.4.3 Perancangan Elektronik Sensor Kompas 33
3.4.4 Perancangan Elektronik Kamera *Webcam* 34
3.4.5 Perancangan Pemakaian Daya Keseluruhan 35
3.5 Perancangan Pemrograman 35
3.5.1 Perancangan Pemrograman Kamera 37
3.5.2 Perancangan Pemrograman Sensor Kompas dan Sensor Ultrasonik 39
BAB IV HASIL DAN PEMBAHASAN 40
4.1 Perubahan Perancangan 40
4.1.1 Perubahan Perancangan Perangkat Keras 40
4.1.2 Perubahan Perancangan Pemrograman 42
4.1.3 Perubahan Perancangan Elektronik 44
4.2 Hasil Perancangan Perangkat Keras 48
4.2.1 Bentuk Fisik Robot 48
4.2.2 Sistem Kelistrikan Robot 50
4.3 Hasil Pengujian Alat 51
4.3.1 Hasil Pengujian Pola Gerakan Robot 51

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.3.2 Hasil Pembacaan Warna dan Kode QR Melalui Modul Kamera 57
4.3.3 Hasil Pengukuran Jarak Melalui Sensor Ultrasonik 62
4.3.4 Pengujian Kecepatan Motor Berdasarkan Arah Gerak Robot 65
BAB V KESIMPULAN DAN SARAN 72
DAFTAR PUSTAKA 73
LAMPIRAN

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Daftar Gambar

Gambar 1.1 Diagram Blok Sistem Robot Pelacak Lokasi....4
Gambar 2.1 Konfigurasi Sistem Kemudi Beroda Empat 7
Gambar 2.2 Raspberry Pi 4 Model B 8
Gambar 2.3 Pin pada Raspberry Pi 4 Model B....9
Gambar 2.4 Motor DC [13] 11
Gambar 2.5 *Commutator* Motor DC [13]....12
Gambar 2.6 Gearbox Motor DC dengan gambar mekanik....12
Gambar 2.7 Prinsip kerja *H-Bridge*....13
Gambar 2.8 Modul *driver* motor DC L298N [17]....14
Gambar 2.9 *Pulse Width Modulation* [15]....15
Gambar 2.10 Sinyal PWM dengan *Duty Cycle* [15] 16
Gambar 2.11 Xiaovv HD USB Webcam 6320S 18
Gambar 2.12 Sensor Ultrasonik dan prinsip kerjanya 19
Gambar 2.13 Sensor Kompas 20
Gambar 3.1 *Block Diagram* proses kerja robot 21
Gambar 3.2 Tampak samping dan depan robot....22
Gambar 3.3 Rangka atas....23
Gambar 3.4 Rangka bawah bagian atas....23
Gambar 3.5 Rangka bawah bagian bawah....23
Gambar 3.6 Perancangan elektronik Robot Pelacak Lokasi....29
Gambar 3.7 Perancangan elektronik modul *driver* L298N 30
Gambar 3.8 Perancangan elektronik modul sensor ultrasonik HC-SR04 32
Gambar 3.9 Perancangan elektronik modul sensor kompas....34
Gambar 3.10 Perancangan elektronik kamera *webcam*....35
Gambar 3.11 *Flowchart* rancangan pemrograman umum....37
Gambar 3.12 Gambaran arena pengujian [1]....38
Gambar 3.13 *Flowchart* pembacaan bentuk dan warna objek 38
Gambar 3.14 *Flowchart* pembacaan kode QR urutan dan lokasi....39
Gambar 3.15 *Flowchart* sensor kompas dan sensor ultrasonik....40
Gambar 4.1 Modul kamera Raspberry Pi OV5647 42
Gambar 4.2 *Flowchart* pemrograman utama dengan perubahan 43

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.3 Perubahan *flowchart* pemrograman sensor ultrasonik.....44
Gambar 4.4 Perubahan *flowchart* pembacaan kode QR lokasi 45
Gambar 4.5 Skematik perubahan rancangan elektronik.....45
Gambar 4.6 Baterai Lipo 11,1V 3s.....48
Gambar 4.7 Baterai Lipo 7,4V 2s dan UBEC 5V 3A 48
Gambar 4.8 Rangka robot tampak atas.....49
Gambar 4.9 Rangka robot tampak depan 49
Gambar 4.10 Rangka robot tampak kanan 49
Gambar 4.11 Rangka robot tampak kiri 50
Gambar 4.12 Rangka robot tampak bawah 50
Gambar 4.13 Diagram blok sistem kelistrikan robot.....51
Gambar 4.14 Skematik PCB sumber tegangan 51
Gambar 4.15 Kotak kuning titik *start*.....52
Gambar 4.16 Kotak kuning titik ELP.....52
Gambar 4.17 Gedung target pada arena 53
Gambar 4.18 Arena pengujian pola gerakan robot.....53
Gambar 4.19 Robot pada posisi tengah kotak *start*.....54
Gambar 4.20 Batas jarak antara robot dengan gedung.....54
Gambar 4.20 Indikator keberhasilan pembacaan kode QR 55
Gambar 4.21 Program pembacaan warna.....58
Gambar 4.22 Hasil pembacaan warna kuning 59
Gambar 4.23 Hasil pembacaan warna hijau 59
Gambar 4.24 Hasil pembacaan warna biru.....59
Gambar 4.25 Hasil pembacaan warna merah muda 59
Gambar 4.26 Pemrograman pembacaan kode QR.....60
Gambar 4.27 Indikator keberhasilan pembacaan kode QR 60
Gambar 4.28 Kamera gagal berfungsi.....61
Gambar 4.29 Grafik hasil pengukuran jarak 62
Gambar 4.30 Pengukuran jarak melalui sensor ultrasonik.....63
Gambar 4.31 Program sensor ultrasonik 63
Gambar 4.32 Program sensor ultrasonik ketika robot berjalan 64
Gambar 4.33 Sensor ultrasonik gagal berfungsi.....65
Gambar 4.34 Proses pengujian kecepatan 65

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Daftar Tabel

Tabel 2.1 Nomor dan nama pin Raspberry Pi 4 Model B 9
Tabel 2.2 Spesifikasi gearbox motor DC.....13
Tabel 2.3 spesifikasi *driver* motor DC L298N.....14
Tabel 2.4 spesifikasi Xiaovv HD USB Webcam 6320S 22
Tabel 2.5 spesifikasi HC-SR04 Ultrasonic Range Finder 19
Tabel 2.6 spesifikasi modul sensor kompas HMC5883L.....20
Tabel 3.1 Pengaruh kecepatan dan arah putaran motor pada pergerakkan robot 25
Tabel 3.2 Data pada kode QR.....26
Tabel 3.3 *Input-output* pin modul *driver* L298N.....30
Tabel 3.4 Rencana pemakaian daya *driver* motor DC L298N 32
Tabel 3.5 *Input-output* pin modul sensor ultrasonik HC-SR04.....33
Tabel 3.6 Rencana pemakaian daya modul sensor ultrasonik HC-SR04 33
Tabel 3.7 *Input-output* pin modul sensor kompas HMC5883L.....34
Tabel 3.8 Rencana pemakaian daya modul sensor ultrasonik HC-SR04 35
Tabel 3.9 Rencanan pemakaian daya keseluruhan 36
Tabel 4.1 Spesifikasi modul kamera Raspberry Pi OV5647 42
Tabel 4.2 Pengkabelan pada Raspberry Pi 46
Tabel 4.3 Pemakaian daya keseluruhan.....47
Tabel 4.4 Deteksi batas jarak gedung kanan pada pola gerakkan satu.....55
Tabel 4.5 Deteksi batas jarak gedung tengah pada pola gerakkan satu.....55
Tabel 4.6 Deteksi batas jarak gedung kiri pada pola gerakkan satu.....56
Tabel 4.7 Jarak robot berhenti pada pola gerakkan satu.....56
Tabel 4.8 Rentang nilai warna kuning, hijau, biru 57
Tabel 4.9 Keberhasilan pembacaan kode QR.....61
Tabel 4.10 Hasil pengukuran jarak melalui sensor ultarsonik.....62
Tabel 4.11 Pengujian kecepatan putaran motor.....66
Tabel 4.12 Pengujian tegangan dan *duty cycle*.....66
Tabel 4.13 Kecepatan sudut, linear, dan torsi pada motor 67
Tabel 4.14 Kecepatan dan torsi pada gedung kanan 69
Tabel 4.15 Kecepatan dan torsi pada gedung tengah 70
Tabel 4.16 Kecepatan dan torsi pada gedung kiri 70

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.17 Kecepatan dan torsi pada titik berhenti 70

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB I
PENDAHULUAN

1.1. Latar Belakang

KRTI (Kontes Robot Terbang Indonesia) 2021 merupakan perlombaan pesawat tanpa awak yang diselenggarakan oleh Pusat Prestasi Nasional Kementerian Pendidikan dan Kebudayaan Indonesia. Peserta dari perlombaan ini adalah mahasiswa dari seluruh perguruan tinggi di Indonesia. Setiap kelompok mahasiswa diminta untuk merancang pesawat tanpa awak yang mampu berfungsi sesuai dengan divisi dalam perlombaan ini. Adapun divisi dalam perlombaan KRTI 2021, yaitu Divisi *Racing Plane* (RP), Divisi *Fixed Wing* (FW), Divisi *Vertical Take-Off Landing* (VTOL), dan Divisi *Technology Developmnt* (TD) [1].

Divisi *Vertical Take-Off Landing* (VTOL) merupakan salah satu divisi pada KRTI 2021 yang mempertandingkan kecepatan dan ketepatan penghantaran barang yang dilakukan secara otomatis oleh *quadcopter drone*. Tema dari Divisi VTOL KRTI 2021 adalah *Pengiriman Barang ke model Apartemen bertingkat menggunakan wahana VTOL berbasis pengenalan alamat kode QR (Quick Response) dan marka WARNA presisi untuk mendukung Kehidupan di Era Tatanan Baru (KETB) pasca pandemi*. Pengenalan alamat barang dilakukan secara *real-time* dengan pedoman kode QR berukuran A4 yang dipasang secara vertikal di dinding gedung lokasi penghantaran barang. Pembacaan warna digunakan untuk pengenalan lokasi penjatuhan barang, gedung penghalang, gedung lokasi penghantaran barang, titik HOME, dan titik *Emergency Landing Pad* (ELP). Algoritma gerak yang harus dicapai pada divisi ini yaitu *quadcopter drone* mampu bergerak dari titik HOME kemudian menghantarkan barang ke tiga gedung sesuai dengan urutan dan mendarat di *Emergency Landing Pad* (ELP).

Pada tugas akhir ini akan dirancang sistem gerak *drone* saat mengantarkan barang. Mengingat pengerjaan *drone* yang mampu terbang dengan stabil dan dilengkapi dengan perangkat pelempar harus dilakukan terlebih dahulu dan membutuhkan konsentrasi tersendiri, maka dari itu algoritma gerakan *drone* akan disimulasikan pada robot beroda sebelum digunakan pada *drone*. Tujuan simulasi gerakan *drone* pada robot beroda adalah agar algoritma pergerakan *drone* pada divisi VTOL KRTI mampu

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

diujicoba tanpa harus menunggu *drone* yang belum sepenuhnya jadi (*drone* masih belum stabil saat terbang).

Basuki Winarno dan Meliyana Wahyu, tahun 2016 pada penelitian di Politeknik Negeri Madiun yang berjudul "Pengendalian Gerak Robot Penghindar Halangan Menggunakan Citra dengan Kontrol PID" menggunakan kamera yang digunakan sebagai pengolah citra sehingga memiliki kemampuan untuk menghindari halangan [2]. Zaki Muhammad Fathi, dkk., tahun 2016 pada penelitian untuk SEMINAR NASIONAL INOVASI DAN APLIKASI TEKNOLOGI (SENIATI) di Institut Teknologi Nasional Malang yang berjudul "Robot Pengenal dan Pencari Objek dengan Kamera Menggunakan Metode Transformasi Hough" menggunakan kamera yang dipasangkan pada robot untuk mengolah citra dalam menunjang fitur deteksi dan pencarian objek [3]. Kukuh Darmawan Setyanto, dkk., pada penelitian di Universitas Jember dengan judul "PENGENDALIAN *MOBILE ROBOT VISION* MENGGUNAKAN WEBCAM PADA OBJEK ARAH PANAH BERBASIS RASPBERRY PI" menggunakan kamera *webcam* sebagai *input* penangkap objek untuk selanjutnya dilakukan proses pengolahan citra yang akan membantu kendali gerak pada robot [4]. Dari penelitian-penelitian tersebut dapat dilihat bahwa pengembangan robot yang telah dilakukan adalah dengan menambahkan kamera untuk menangkap gambar dan pemrosesan gambar dilakukan oleh mini-komputer Raspberry Pi. Hal ini menjadi tantangan bagi tugas akhir ini, yaitu merancang kamera yang mampu menjadi pemandu pelacakan lokasi pada robot dengan pusat pemrosesan data pada Raspberry Pi. Oleh karena itu, judul pada tugas akhir ini adalah "ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI".

1.2. Tujuan dan Manfaat

Tujuan dari penelitian ini yaitu merancang robot yang mampu bergerak menuju lokasi yang diminta sesuai dengan panduan dari pembacaan kamera.

Manfaat dari penelitian ini yaitu :

1. Robot mampu bergerak sesuai dengan algoritma VTOL pada KRTI.
2. Program mampu diaplikasikan pada *drone*.
3. Mampu memberi pengetahuan mengenai pengaruh pembacaaan kamera terhadap gerakan robot.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

1.3. Batasan Masalah

Batasan masalah pada proyek ini adalah :

1. Kontroler menggunakan Raspberry Pi.
2. Pemrograman menggunakan Bahasa Pemrograman Phyton.
3. Kamera digunakan untuk mendeteksi warna dan bentuk objek, serta membaca kode QR urutan dan lokasi.
4. Sensor ultrasonik digunakan untuk menentukan batas jarak antara robot dengan gedung.
5. Sensor kompas digunakan untuk mengetahui arah robot.
6. Penggerak menggunakan 4 buah motor DC.
7. Buzzer akan berbunyi ketika kamera berhasil membaca kode QR urutan dan lokasi.
8. Tangkapan gambar dari kamera mampu dilihat secara *real-time* pada monitor komputer melalui sambungan *wifi*.

1.4. Metodologi Penelitian

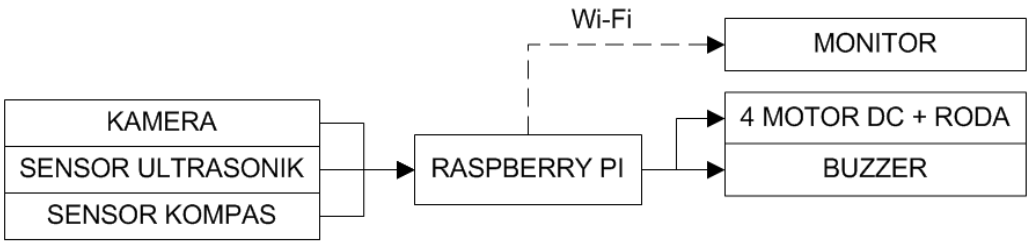
Metode-metode yang digunakan agar proses pengerjaan proyek dapat berjalan sesuai dengan yang diharapkan adalah :

1. Studi Pustaka

   Studi pustaka bertujuan untuk mempelajari pustaka yang akan digunakan sebagai referensi dalam perancangan robot, yaitu penerapan kamera dalam pembacaan benda sebagai acuan pergerakan robot, penerapan mikro-kontroler Raspberry Pi, motor DC, dan rancang bangun robot.
2. Perancangan Sistem

   Perancangan sistem dilakukan dengan tujuan merancang sistem pembacaan kamera untuk membedakan warna, membedakan bentuk objek, dan membaca kode QR urutan dan lokasi. Pada tahap ini juga dilakukan perancangan mekanik dengan membuat gambar desain robot yang mencakup posisi 4 motor dan roda penggerak, kamera, sensor ultrasonik, sensor kompas, buzzer, Raspberry Pi.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 1.1 Diagram Blok Sistem Robot Pelacak Lokasi

Acuan dalam perancangan sistem ini dapat dilihat pada diagram blok yang merupakan gambaran umum sistem Robot Pelacak Lokasi. Pada diagram blok di atas kamera berfungsi sebagai *input* pada Raspberry Pi yang kemudian akan diproses. *Output* dari Raspberry Pi yaitu bergeraknya robot dengan bantuan 4 buah roda penggerak dan aktifnya buzzer ketika kamera berhasil membaca kode QR urutan dan lokasi.

3. Pembuatan Alat

   Pada tahap ini dilakukan pembuatan sistem mekanik, kontrol, dan elektronik pada robot. Pembuatan sistem mekanik dilakukan dengan cara merakit kerangka robot dan memasang komponen kelistrikan yang dibutuhkan pada robot. Pembuatan sistem kontrol dilakukan dengan membuat program yang digunakan untuk mengatur jalannya robot sesuai dengan pembacaan kamera. Pembuatan sistem elektronik dilakukan dengan pengkabelan pada setiap komponen kelistrikan robot sekaligus memastikan tegangan dan arus yang masuk pada setiap komponen sesuai dengan yang dianjurkan serta memastikan kabel untuk transfer data dari kamera dan motor DC ke Raspberry Pi dapat bekerja dengan baik.

4. Pengujian dan Pengambilan Data

   Tahap pengujian dan pengambilan data dilakukan untuk mengetahui seberapa jauh kamera mampu mendeteksi bentuk, mendeteksi warna dan membaca kode QR urutan dan lokasi.

5. Analisis dan Kesimpulan Hasil

   Pada tahap ini data yang diperoleh pada tahap pengujian dan pengambilan data akan dianalisis. Hasil analisis data akan menjadi informasi apakah gerakan robot sudah sesuai dengan yang direncanakan atau belum. Jika belum sesuai maka akan dilakukan *troubleshooting* pada robot hingga robot mampu bergerak sesuai dengan spesifikasi yang sudah direncanakan.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB II
DASAR TEORI

Bab ini akan membahas tentang dasar teori dari robot beroda, mini-komputer Raspberry Pi beserta bahasa pemrograman yang digunakan, dan komponen-komponen elektronik yang digunakan dalam merancang robot ini.

2.1 Robot Beroda

Menurut *Robotic Institute of America* di Universitas Carnegie Mellon di kota Pittsburgh, negara bagian Pensilvania, Amerika Serikat, pada tahun 1979 mendefinisikan secara formal yang dimaksud dengan robot adalah "manipulator multi fungsi dan dapat diprogram ulang yang dirancang untuk menggerakkan material, alat, atau perangkat khusus melalui sejumlah gerakan terprogram untuk melakukan aktifitas tertentu". Menurut Basuki Winarno dan Meliyana Wahyu, tahun 2016 pada penelitian di Politeknik Negeri Madiun yang berjudul "Pengendalian Gerak Robot Penghindar Halangan Menggunakan Citra dengan Kontrol PID", mengatakan bahwa robot merupakan suatu mekanik yang dapat diperintah secara otomatis untuk memproses suatu benda agar dapat bekerja sesuai dengan permintaan tertentu sehingga dapat meringankan pekerjaan manusia [2]. Berdasarkan definisi-definisi tersebut dapat disimpulkan bahwa robot merupakan suatu alat yang memiliki empat karakteristik, yaitu memiliki sensor, memiliki sistem kecerdasan (kontrol), memiliki peralatan mekanik (aktuator), dan memiliki sumber daya (*power*) (Wisnu Jatmiko dkk., 2012: 18).

Robot mampu diklasifikasi berdasarkan fungsi, sifat, konstruksi, sistem kendali, dan pemrogramannya. Berdasarkan konstruksinya, robot dibagi menjadi robot beroda, robot manipulator (tangan), robot berkaki, robot terbang (*flying robot*), robot dalam air (*under water robot*), robot humanoid, dan robot animaloid. Robot beroda merupakan robot yang secara konstruksi menggunakan roda sebagai aktuator untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut mampu melakukan perpindahan lokasi dari satu titik ke titik yang lain. Robot beroda ini sangat disukai oleh orang yang baru

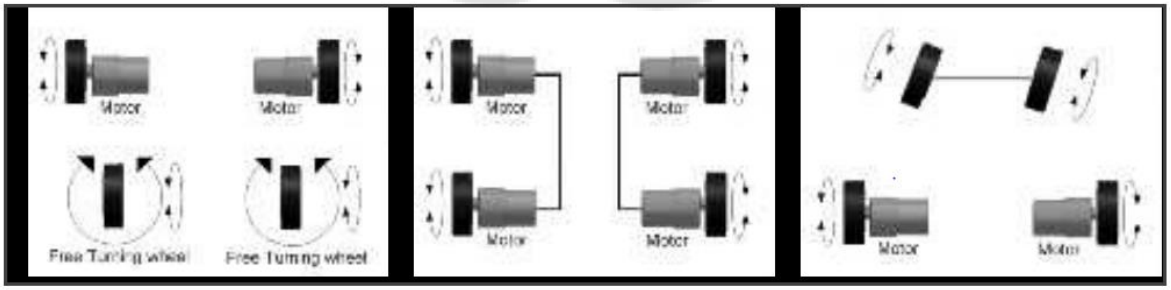
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

mempelajari robot. Hal ini dikarenakan robot beroda tidak dibuat untuk kerja fisik yang berat [5].

Robot beroda bisa diklasifikasikan berdasarkan jumlah roda yang dipakai, yaitu robot roda dua, robot roda tiga, robot roda empat, dan robot roda empat berantai [6]. Keempatnya memiliki sistem pengontrolan gerak yang berbeda. Sistem gerak yang berbeda tersebut didasari oleh jumlah roda yang dipakai. Jumlah roda pada robot memengaruhi kemudahan robot pada saat berbelok (semakin banyak roda maka semakin banyak gesekan yang harus dilawan robot) dan juga berpengaruh pada keseimbangan robot.

Tugas akhir ini akan menggunakan robot beroda empat. Robot beroda empat memiliki pusat massa pada pusat persegi sehingga badan robot data dirancang sedikit lebih besar dari robot beroda tiga (Wisnu Jatmiko dkk., 2012: 32). Beberapa konfigurasi sistem kemudi pada robot beroda empat adalah sebagai berikut.

a) Dua roda belakang yang memiliki motor terpisah dan dua roda depan yang dapat berputar ke segala arah (Gambar 2.1 a).
b) Sebuah pasangan sisi roda depan-belakang dihubungkan ke motor penggerak yang sama dan pasangan sisi roda lainnya dihubungkan ke motor penggerak terpisah (Gambar 2.1 b). Konfigurasi ini yang akan digunakan pada rancangan robot di tugas akhir ini.
c) Konfigurasi roda empat yang mengikuti sistem kemudi mobil yang hanya memerlukan sebuah motor penggerak untuk masing-masing roda belakang dan sebuah servo untuk kemudi pada pasangan roda depan (Gambar 2.1 c). Kelemahan utama sistem kemudi mobil pada robot adalah *dead reckoning* membutuhkan *effort* yang besar. *Dead reckoning* yaitu kemampuan robot untuk menghitung posisi saat ini relatif terhadap posisi sebelumnya [6].



a. b. c.

Gambar 2.1 Konfigurasi Sistem Kemudi Beroda Empat [6]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.2 Raspberry Pi

Raspberry Pi adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah [7]. Dengan ukuran yang kecil Raspberry Pi mampu mempermudah pengguna dalam menjalankan program perkantoran, permainan komputer, dan sebagai pemutar video beresolusi tinggi. Raspberry Pi dilengkapi dengan pin GPIO yang digunakan untuk menghubungkan perangkat elektronik lain ke Raspberry Pi.

2.2.1 Raspberry Pi 4 Model B

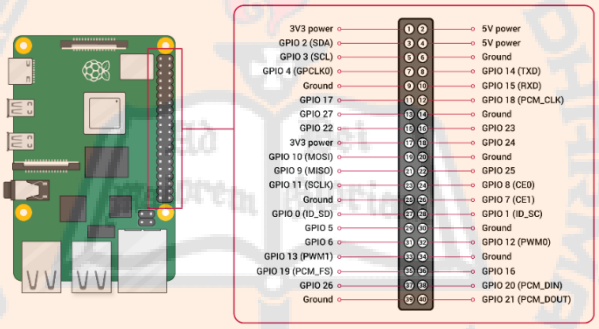
Raspberry Pi 4 Model B merupakan produk terbaru dari Raspberry Pi yang dirilis pada Juni 2019. Model ini memiliki peningkatan kecepatan pada prosesor, peforma multimedia, memori, dan konektivitas dibandingkan dengan generasi sebelumnya Raspberry Pi 3 Model B+, dengan tetap mempertahankan kompatibilitas dan besar konsumsi daya. Raspberry Pi 4 Model B memiliki prosesor 1.5GHz 64-bit *quad core* ARM Cortex-A72 dengan 8GB LPDDR4 RAM. Fitur kunci lainnya dari Raspberry Pi 4 Model B adalah *dual-display support* hingga resolusi 4K melalui sepasang port mikro-HDMI, dekode video perangkat keras hingga 4Kp60, LAN nirkabel *dual-band* 2.4/5.0GHz, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, dan kemampuan PoE (melalui *add-on* terpisah PoE HAT) [8]. Raspberry Pi 4 Model B memiliki 40 pin yang terdiri dari: 2 pin sumber tegangan 5V, 2 pin sumber tegangan 3.3V, 8 pin *ground*, 26 pin GPIO dan 2 pin serial [9]. Pada proyek tugas akhir ini Raspberry Pi 4 Model B digunakan sebagai pemroses tangkapan gambar pada kamera dan juga sebagai pusat pengontrol gerakan robot.

Gambar 2.2 Raspberry Pi 4 Model B [8]

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.2.2 GPIO Raspberry Pi 4 Model B

GPIO (*General-Purpose Input Output*) pada Raspberry Pi 4 Model B adalah pin yang digunakan sebagai *input* atau *output* dan untuk penggunaan yang lebih luas. Sebagai *output*, GPIO mampu mengeluarkan tegangan sebesar 3.3V untuk logika *HIGH* dan 0V untuk logika *LOW*. Untuk pembacaan tegangan *input*, GPIO mampu membaca tegangan input sebesar 3.3V untuk logika *HIGH* dan 0V untuk logika *LOW*. Proses *input* pada GPIO Raspberry Pi menjadi semakin mudah dengan adanya *internal pull-up* atau *internal pull-down resistors*. Pada penggunaan dalam jangkauan yang lebih luas, pin GPIO pada Raspberry Pi mampu digunakan sebagai pin PWM (*pulse-width modulation*), SPI (*Serial Peripheral Interface Bus*), I2C (*Inter-Intergreted Circuit*), dan pin serial (pin TXD dan RXD) [10]. Gambar 2.9 dan Tabel 2.1 di bawah menunjukkan posisi pin sumber tegangan, *ground*, dan pin GPIO pada Raspberry Pi.



Gambar 2.3 Pin pada Raspberry Pi 4 Model B [10]

Tabel 2.1 Nomor dan Nama Pin Raspberry Pi 4 Model B [10]

Nomor Pin Pada Papan	**Nama Pin dan Fungsi Pin**
1	3V3 power
2	5V power
3	GPIO 2 (SDA)
4	5V power
5	GPIO 3 (SCL)
6	Ground
7	GPIO 4 (GPCLK0)
8	GPIO 14 (TXD)
9	Ground
10	GPIO 15 (RXD)
11	GPIO 17

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 2.1 Nomor dan Nama Pin Raspberry Pi 4 Model B [10] (lanjutan)

Nomor Pin Pada Papan	Nama Pin dan Fungsi Pin
12	GPIO 18 (PCM_CLK)
13	GPIO 27
14	Ground
15	GPIO 22
16	GPIO 23
17	3V3 power
18	GPIO 24
19	GPIO 10 (MOSI)
20	Ground
21	GPIO 9 (MISO)
22	GPIO 25
23	GPIO 11 (SCLK)
24	GPIO 8 (CE0)
25	Ground
26	GPIO 7 (CE1)
27	GPIO 0 (ID_SD)
28	GPIO 1 (ID_SC)
29	GPIO 5
30	Ground
31	GPIO 6
32	GPIO 12 (PWM0)
33	GPIO 13 (PWM1)
34	Ground
35	GPIO 19 (PCM_FS)
36	GPIO 16
37	GPIO 26
38	GPIO 20 (PCM_DIN)
39	Ground
40	GPIO 21 (PCM_DOUT)

2.2.3 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang menggabungkan kode yang dilengkapi dengan fungsionalitas pustaka standar yang besar [11]. Python pada umumnya digunakan sebagai bahasa skrip walaupun pemakaiannya mampu mencakup konteks yang lebih luas yang pada umumnya tidak bisa dilakukan dengan menggunakan bahasa skrip. Beberapa fitur yang dimiliki Python dan menjadi keunggulan dibanding bahasa pemrograman lain adalah [12]:

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

a) memiliki kepustakaan yang luas, dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
b) memiliki sistem pengelolaan memori otomatis (*garbage collection*, seperti java).
c) modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

Berdasarkan fitur-fitur tersebut maka bahasa pemrograman Python sangat sesuai digunakan untuk pemrograman pada Raspberry Pi, mengingat penggunaan Raspberry Pi sebagai saran belajar pemrograman bagi pemula dan juga sebagai pengontrol pada sistem yang kompleks dan membutuhkan pengelolaan memori yang baik.

2.3 Motor DC

Motor listrik DC merupakan perangkat elektronik yang mampu mengubah energi listrik arus searah menjadi energi mekanik. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan torsi yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Motor DC umumnya dibatasi untuk penggunaan berkecepatan rendah dan penggunaan daya rendah, serta biayanya relatif lebih murah dari motor AC (I Nyoman Bagia & I Made Parsa, 2018: 3 – 5).

Gambar 2.4 Motor DC [13]

Bagian-bagian utama dari motor DC dapat diuraikan sebagai berikut [13].

1) Kutub Medan

   Motor DC memiliki 2 kutub medan magnet yaitu kutub utara dan kutub selatan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Garis *magnetic energy* membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan.

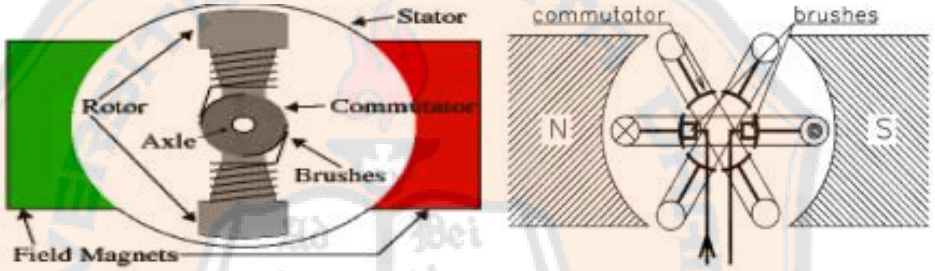
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2) Dinamo

Dinamo pada motor DC berbentuk silinder, dihubungkan kearah penggerak untuk menggerakkan beban. Arus yang masuk menuju dinamo akan menjadi elektromagnet. Pada motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan berganti lokasi. Saat hal itu terjadi arus yang masuk ke dalam motor DC akan berbalik dan merubah kutub-kutub utara dan selatan dinamo.

3) *Commutator*

Kegunaan komponen ini pada motor DC adalah untuk membalikkan arah arus listrik dalam dinamo. *Commutator* juga membantu motor DC dalam hal transmisi arus antara dinamo dan sumber daya. Proses dalam *commutator* dapat dilihat pada Gambar 2.11.

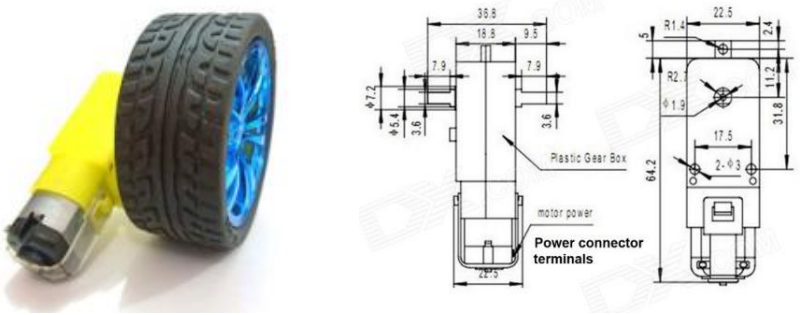


Gambar 2.5 *Commutator* Motor DC [13]

2.3.1 Gearbox Motor DC

Gearbox merupakan perangkat tambahan pada motor DC yang berisi susunan roda gigi dan berfungsi untuk mengubah torsi atau kecepatan putaran motor. Gearbox pada umumnya digunakan untuk menambah torsi atau kecepatan motor ketika motor menarik beban yang berat. Dengan adanya gearbox maka beban yang diterima oleh motor tidak langsung diterima oleh dinamo, tetapi ditransmisikan terlebih dahulu melalui mekanisme roda gigi. Pada tugas akhir ini besar perbandingan roda gigi pada gearbox ini adalah 1:48. Roda pada robot akan dipasang pada as gearbox. Spesifikasi dari gearbox motor DC ini dapat dilihat pada Tabel 2.2.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



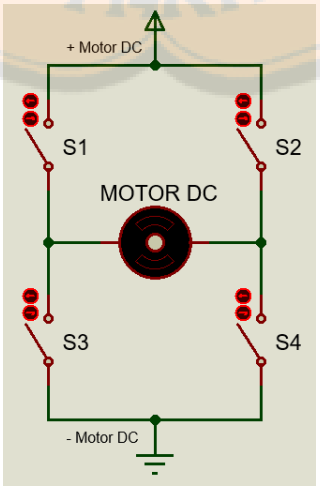
Gambar 2.6 Gearbox Motor DC dengan gambar mekanik

Tabel 2.2 Spesifikasi Gearbox Motor DC

Tegangan	Arus Tanpa Beban	Kecepatan Tanpa Beban	Torsi
3V	≤ 170mA	130 ±10% rpm	Sekitar 1kgf.cm
6V	≤240mA	290±10% rpm	Sekitar 1.5kgf.cm
7.2V	≤ 260mA	330 ± 10% rpm	Sekitar 1.8kgf.cm
Dimensi motor : 2.76 in x 1.46 in x 0.87 in (7 cm x 3.7 cm x 2.2 cm)			

2.3.2 *Driver* Motor DC

Driver motor DC digunakan untuk mengatur arah putaran motor, sehingga motor mampu berputar *forward* atau *reverse*. Pengaturan arah putaran motor tidak bisa dikontrol langsung dari mikro-kontroler karena tegangan sinyal yang keluar terlalu kecil dan mikro-kontroler tidak mampu diprogram untuk membalik arus. Salah satu jenis *driver* yang biasa digunakan adalah *H-Bridge.* Rangkaian *driver* ini menggunakan dua buah pasangan transistor jenis MOSFET sebagai saklar [14]. Prinsip kerja *H-Bridge* dapat dilihat pada Gambar 2.12.



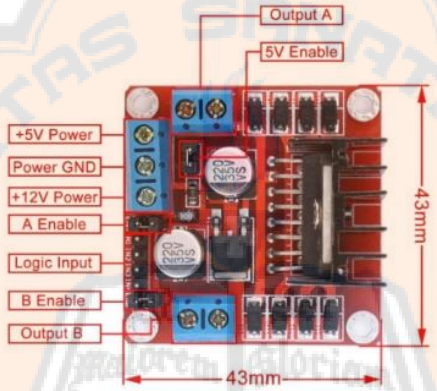
Gambar 2.7 Prinsip kerja *H-Bridge*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.3.3 *Driver* Motor DC L298N

Driver motor DC L298N merupakan *driver* motor dengan tipe *dual H-Bridge* yang mampu mengendalikan motor untuk berputar dua arah [16]. Modul L298N ini mampu mengatur arah putaran dua motor DC [17]. Pembagian pin untuk setiap motor adalah sebagai berikut.

1. Pin Enable A, *input* 1, dan *input* 2 digunakan untuk mengatur motor DC yang terhubung ke *output* 1 dan *output* 2.
2. Pin Enable B, *input* 3, dan *input* 4 digunakan untuk mengatur motor DC yang terhubung ke *output* 3 dan *output* 4.



Gambar 2.8 Modul *driver* motor DC L298N [17]

Tabel 2.2 Hubungan Pengendalian Putaran Motor DC Melalui L298N

Enable A / Enable B	***Input* 1**	***Input* 2**	**Keterangan**
> 0	0	1	Berputar satu arah
> 0	1	0	Berputar ke arah lain
> 0	0	0	Berhenti berputar
0	-	-	Motor berputar sangat pelan

Modul *driver* motor ini mampu mengendalikan 2 motor dan menghasilkan arus makasimal 2A untuk kedua arah putaran motor [16]. Modul ini sangat cocok untuk pengaplikasian pada sistem robot karena sangat baik untuk disambungkan dengan mikrokontroler. Untuk pengendaliannya, modul ini bisa dikendalikan melalui saklar manual sederhana, gerbang logika TTL, relai, dll. Modul ini dilengkapi dengan indikator

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LED, regulator *on-board* +5V dan dioda pelindung. Berdasarkan *datasheet* modul L298N dari Handson Technology, spesifikasi lebih rinci dari modul ini dapat dilihat pada Tabel 2.3 [17].

Tabel 2.3 Spesifikasi *Driver* Motor DC L298N

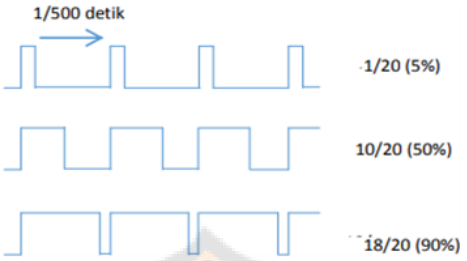
Spesifikasi	**Keterangan**
Tegangan *input*	3.2-40Vdc
Driver	L298N Dual H Bridge DC Motor *Driver*
Suplai tegangan	DC 5-35V
Peak current	2A
Operating current range and control signal input voltage range	0 ~ 36mA *Low*: -0.3V ≤ Vin ≤ 1.5V. *High*: 2.3V ≤ Vin ≤ Vss.
Enable signal input voltage range	*Low*: -0.3 ≤ Vin ≤ 1.5V (*control signal is invalid*). *High*: 2.3V ≤ Vin ≤ Vss (*control signal active*).
Maximum power consumption	20W (*when the temperature* T = 75 ℃).
Storage temperature	-25 ℃ ~ +130 ℃
On-board +5V regulated Output supply (supply to controller board i.e. Arduino).	
Ukuran	3.4cm x 4.3cm x 2.7cm

2.3.4 PWM (*pulse-width modulation*)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu perioda, untuk mendapatkan tegangan rata-rata yang berbeda (Candra & Ta'ali, 2020). PWM diperoleh dengan mengubah-ubah siklus kerja (*duty cycle*) gelombang kotak sehingga mampu mencapai tegangan keluaran yang berbeda-beda (S. & Hartono, 2018). Teknik mengubah lebar pulsa yang disebut PWM dapat diterapkan pada sinyal digital dengan lebar keadaan *high* dan *low* [15]. Berdasarkan uraian dari beberapa penelitian tersebut, penggunaan PWM

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

pada proyek tugas akhir ini ditujukan untuk mengatur kecepatan motor yang memengaruhi kecepatan gerak robot dan juga kemampuan robot untuk berbelok.

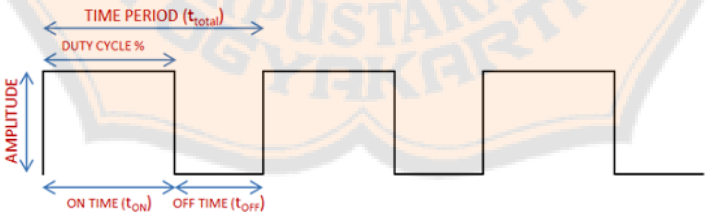


Gambar 2.9 *Pulse Width Modulation*[15]

Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi antara 0% hingga 100%. *Duty Cycle* merupakan representasi dari kondisi logika *high* dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan rentang nilai 0% sampai 100% [15]. Pada Gambar 2.13 dapat dilihat besaran *duty cycle* yang bervariasi yang dipengaruhi oleh lamanya kondisi sinyal *high* dalam satu periode.

2.3.5 Hubungan Duty Cycle PWM dengan Keluaran Tegangan

Sinyal PWM pada proyek tugas akhir ini digunakan untuk mengatur kecepatan motor yang digunakan pada robot. Sinyal PWM diatur dengan menentukan lebar *Duty Cycle* pada PWM yang merupakan lamanya siklus kerja pada satu sinyal PWM.



Gambar 2.10 Sinyal PWM dengan *Duty Cycle*[15]

Besarnya *Duty Cycle* dapat ditentukan dengan persamaan 2.1 dan 2.2 :

$$Duty\ Cycle = T_{ON} / (T_{ON} + T_{OFF}) \quad (2.1)$$

atau

$$Duty\ Cycle = T_{ON} / T_{total} \quad (2.2)$$

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Keterangan :

TON = Waktu siklus PWM pada posisi tinggi

TOFF = Waktu siklus PWM pada posisi rendah

Ttotal = Waktu satu siklus PWM atau periode satu gelombang PWM

Pada proyek ini, besarnya *Duty Cycle* akan ditentukan dalam persen sehingga hasil untuk *Duty Cycle* dalam persen dapat ditentukan dengan persamaan 2.3 :

$$Duty\ Cyle = \frac{\text{TON}}{\text{Ttotal}} \times 100\% \qquad (2.3)$$

Lebar *Duty Cycle* akan diimplementasikan pada besarnya tegangan yang diinginkan untuk menentukan kecepatan motor. Besarnya tegangan yang keluar ke motor melalui pengaturan *Duty Cycle* ini dapat ditentukan melalui persamaan 2.4 :

$$V_{OUT} = Duty\ Cycle \times V_{IN} \qquad (2.4)$$

Keterangan :

VOUT = Tegangan yang diinginkan untuk menentukan kecepatan motor

Duty Cycle = Besarnya siklus kerja PWM yang dinyatakan dalam persen (%)

VIN = Tegangan sumber yang masuk ke motor

2.4 Kamera *Webcam*

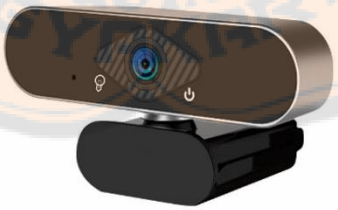
Webcam (*web camera*) adalah sebutan bagi kamera *real-time* yang gambarnya bisa diakses atau dilihat melalui *World Wide Web*, program *instant messaging*, atau aplikasi *video call*. Sebuah *webcam* sederhana terdiri dari sebuah lensa standar yang dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar; *casing (cover)*, termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar; kabel support yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector*, kabel ini dikontrol untuk menyesuaikan ketinggian, arah, dan sudut pandang *webcam.*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Kamera *Webcam* digunakan pada proyek ini sebagai alat untuk menangkap gambar dari objek pada area gerak robot. Perangkat kamera akan dilengkapi dengan tiang penyangga agar posisi kamera mampu sejajar dengan posisi kode QR yang berada di sisi atas gedung lokasi maupun gedung penghalang (*obstacle*). Kamera *webcam* pada proyek ini memiliki nama Xiaovv HD USB Webcam 6320S. Kamera ini memiliki kualitas gambar *high definition* 1080P, mampu *auto-focus* pada target yang akan diambil gambarnya, penglihatan kamera 150 derajat, dilengkapi dengan *microphone*, dan transfer data serta sumber tegangan pada kamera menggunakan sambungan USB sehingga lebih mudah untuk digunakan. Spesifikasi dari kamera ini dapat dilihat pada Tabel 3.3.

Tabel 2.4 Spesifikasi Xiaovv HD USB Webcam 6320S

Spesifikasi	**Keterangan**
Nama Perangkat	Xiaovv HD USB Webcam 6320S
Ukuran	100mm x 2mm x 50mm / 3.9*0.9*1.9inches
Berat	105g
Suplai tegangan	DC 5V
Pemakaian Arus	100mA – 2A
Pixels	200W
Wide angle	150˚
Spesifikasi	**Keterangan**
Working temperature	-10˚C~50˚C
Video format	H.264 H.265 MJPG NV12 YUY2
Supportive system	*Windows 7/8/10, Linux2.4.6 and higher, MacOS10.5 and higher*



Gambar 2.11 Xiaovv HD USB Webcam 6320S

Sumber : www.migadget.id/product

2.5 Sensor Ultrasonik

Sensor ultrasonik yang digunakan pada proyek tugas akhir ini merupakan sensor yang sudah tersusun dalam modul *ultrasonic ranging module* HC-SR04. Perangkat ini digunakan untuk mendeteksi jarak suatu objek dengan jarak yang mampu diukur sekitar 2

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

– 400 cm. Perangkat ini menggunakan dua pin digital untuk mengkomunikasikan jarak yang terbaca. Prinsip kerja sensor ultrasonik ini adalah dengan mengirimkan pulsa ultrasonik sekitar 40kHz, kemudian menerima pantulan dari pulsa *echo*, dan menghitung waktu yang diambil dalam mikrodetik seperti yang digambarkan pada Gambar 2.14

Gambar 2.12 Sensor Ultrasonik dan prinsip kerjanya

Sumber : www.jogjarobotika.com/HC-SR04

2.5.1 Modul Sensor Ultrasonik HC-SR04

Sensor ultrasonik digunakan pada proyek tugas akhir ini untuk membantu robot dalam mengetahui batas jarak antara robot dengan gedung lokasi atau gedung *obstacle*. Pada tugas akhir ini, sensor ultrasonik yang digunakan sudah dalam bentuk modul dengan nama HC-SR04 Ultrasonic Range Finder. Modul sensor ini menggunakan IC sensor HC-SR04 yang mampu digunakan untuk mengukur jarak dengan membaca nilai pantulan gelombang ultrasonik yang mengenai suatu objek. Keunggulan dari sensor ini adalah mampu mendeteksi suatu objek dengan jarak 2-400cm dengan resolusi 1cm. Modul ini dilengkapi dengan empat pin, yaitu pin VCC, Gnd, Trig, dan Echo. Spesifikasi dari modul sensor ini dapat dilihat pada Tabel 3.4

Tabel 2.5 Spesifikasi HC-SR04 Ultrasonic Range Finder

Spesifikasi	**Keterangan**
Nama Perangkat	HC-SR04 Ultrasonic Range Finder
Suplai tegangan	DC 3.3-5V
Konsumsi arus	Kurang dari 2mA
Tingkat *output*	5V tinggi
Tingkat output pada akhir	0V
Induksi sudut	Tidak lebih dari 15 derajat
Jarak deteksi	2- 400cm
Presisi tinggi	3mm

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.6 Sensor Kompas

Sensor kompas digital merupakan modul sensor magnetometer dengan keluaran berupa sudut yang menyatakan arah hadap. Sedangkan sensor magnetometer keluarannya berupa besar medan magnet bumi yang dapat diukur dalam tiga sumbu yang dapat digunakan untuk menentukan sudut arah hadap.

Gambar 2.13 Sensor Kompas

Sumber : www.jogjarobotika.com/HMC5883L

2.6.1 Modul Sensor Kompas HMC5883L

Sensor kompas pada proyek tugas akhir ini digunakan untuk mengetahui arah dan posisi robot pada saat bergerak. Sensor kompas pada tugas akhir ini menggunakan IC HMC5883L dan sudah tersusun dalam suatu modul. IC HMC 5883L merupakan IC kompas digital 3 axis yang memiliki 2 pin I2C sebagai penghubung antar-muka, memiliki sensor *magneto-resistive* HMC118X *series* dengan resolusi tinggi, serta memiliki ASIC dengan konten *amplification, automatic degaussing strap driver, offset cancelation* dan 12 bit ADC yang memungkinkan keakuratan pembacaan kompas antara 1 sampai 2 derajat. Sensor ini bekerja dengan membaca nilai dan arah medan magnet bumi. Modul sensor ini memiliki 5 pin, yaitu pin VCC, Gnd, SDA, SCL, dan DRDY. Spesifikasi modul sensor kompas HMC5883L dapat dilihat pada Tabel 3.5

Tabel 2.6 Spesifikasi Modul Sensor Kompas HMC5883L

Spesifikasi	Keterangan
Nama Perangkat	Modul HMC5883L (modul medan magnet tiga axis)
Model	GY-271
IC yang digunakan	HMC5883L
Suplai tegangan	3-5V
Operating Current	130 μA
Metode komunikasi	Protokol komunikasi I2C
Jarak pengukuran	±1.3-8 Gauss

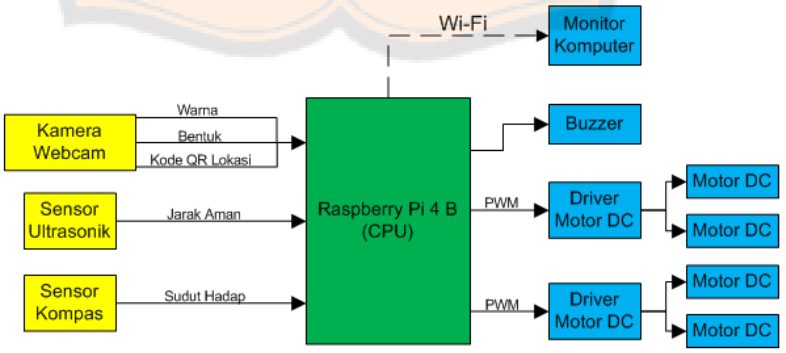
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB III
RANCANGAN PENELITIAN

Bab ini berisi perancangan dari proyek tugas akhir yang berjudul *"ROBOT PELACAK LOKASI BERBASIS RASPBERRY PI"*, yang terdiri dari *block diagram*, perancangan mekanik, perancangan perangkat keras, perancangan elektronik, dan *flowchart* dari keseluruhan proses kerja dari robot ini.

3.1 *Block Diagram*

Pada bagian ini disajikan *block diagram* pada gambar 3.1 yang akan menjelaskan proses kerja "ROBOT PELACAK LOKASI OTOMATIS BERBASIS RASPBERRY PI" dengan perangkat dan jenis data yang digunakan. Perangkat yang digunakan pada proses ini terdiri dari perangkat *input, process,* dan *output*. Perangkat yang digunakan sebagai *input* yaitu kamera *webcam*, sensor ultrasonik, dan sensor kompas. Masing-masing perangkat tersebut akan memberikan data *input* pada Raspberry Pi: *input* data warna dan bentuk objek, serta hasil pembacaan kode QR urutan dan lokasi; data hasil deteksi jarak aman yang ditentukan antara robot dengan objek yang berdiri di dekat robot; data hasil deteksi sudut hadap dari robot. Semua *input* tersebut akan diproses pada Raspberry Pi dan *output* dari proses tersebut berupa suara yang dihasilkan dari *buzzer* ketika berhasil membaca kode QR urutan dan lokasi, tampilan gambar dari kamera yang dapat dilihat pada monitor komputer, dan kecepatan serta arah putar motor.

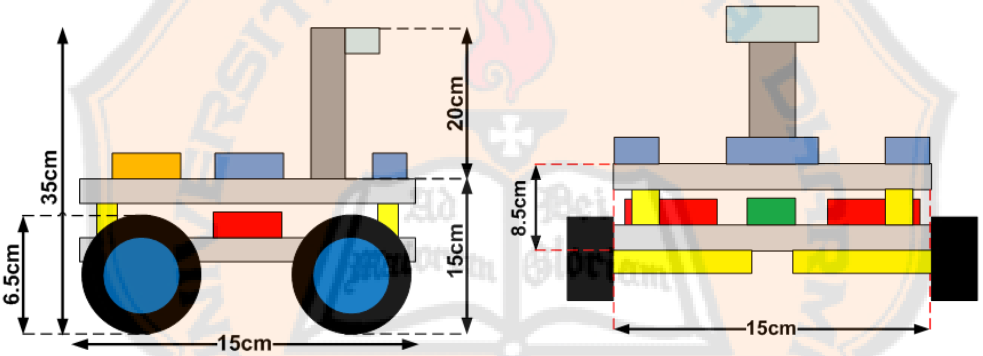


Gambar 3.1 *Block diagram* proses kerja robot

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.2 Perancangan Mekanik

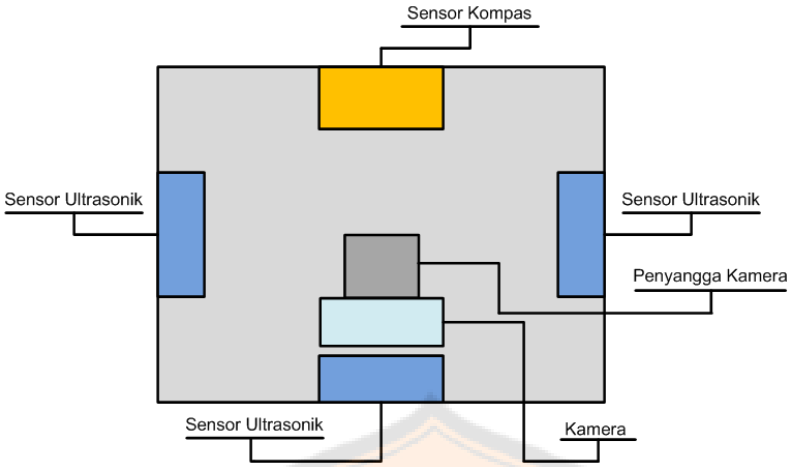
Bagian ini berisi rancangan rangka robot yang akan digunakan pada tugas akhir ini. Rangka robot akan dibuat menggunakan besi akrilik yang lebih mudah digunakan dalam perakitan karena mudah untuk dibuat ulir, struktur bahannya ringan dan kuat. Rangka robot terdiri dari dua bagian, yaitu bagian atas dan bagian bawah. Rangka bagian atas digunakan untuk meletakan kamera dengan tiang penopangnya, tiga buah sensor ultrasonik, dan sensor kompas. Sedangkan, rangka bagian bawah digunakan untuk meletakkan Raspberry Pi 4 B, *driver* motor DC, dan di bagian bawah akan dipasang empat buah motor DC yang sudah dilengkapi dengan roda. Pada Gambar 3.2 dapat dilihat desain rangka robot secara umum beserta ukuran panjang, lebar, dan tinggi robot, serta ukuran diameter roda dan tinggi tiang penyangga kamera.



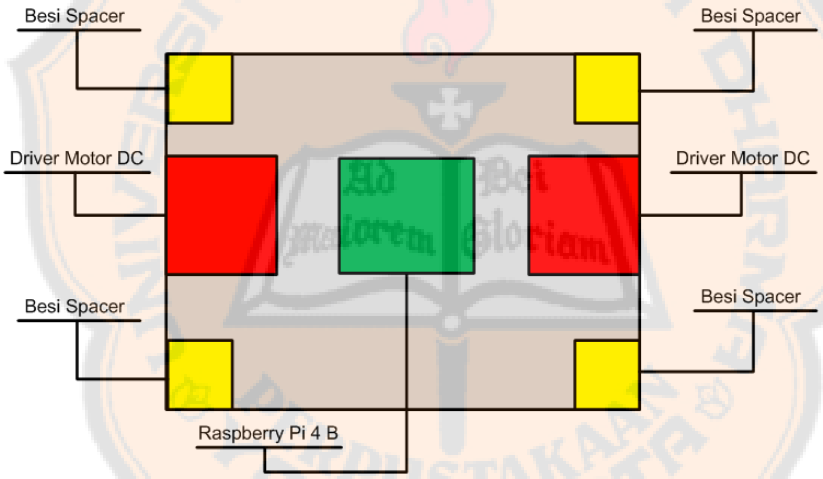
Gambar 3.2 Tampak samping dan depan robot

Pada Gambar 3.3, Gambar 3.4, dan Gambar 3.5 dapat dilihat posisi peletakkan perangkat keras yang digunakan pada robot. Perangkat keras pada rancangan robot ini diletakkan pada bagian atas dan bawah rangka. Rangka atas digunakan untuk meletakkan sensor ultrasonik di bagian depan, kanan, dan kiri, kamera dengan penyangga, dan sensor kompas di bagian belakang. Pada rangka bawah bagian atas digunakan untuk meletakkan Raspberry Pi 4 B dengan dua *driver* motor DC di sisi kanan dan kiri. Besi *spacer* di pasang pada sudut rangka bawah untuk menyangga rangka atas. Pada bagian bawah rangka bawah digunakan untuk memasang empat buah motor DC.

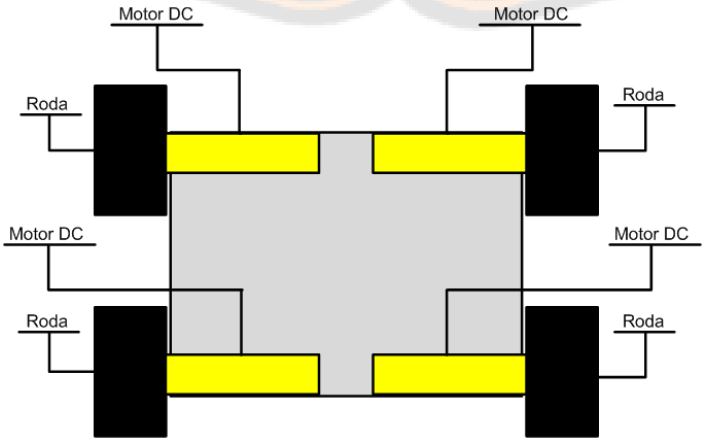
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.3 Rangka atas



Gambar 3.4 Rangka bawah bagian atas



Gambar 3.5 Rangka bawah bagian bawah

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.3 Perancangan Perangkat Keras

Bagian ini berisi uraian spesifikasi dari perangkat keras yang akan digunakan dalam proyek tugas akhir ini. Pemilihan tipe perangkat keras pada tugas akhir ini berdasarkan pada kebutuhan alat yang akan dibuat dan penelitian yang sudah pernah memakai perangkat-perangkat keras tersebut.

3.3.1 Gearbox Motor DC

Gearbox motor DC terdiri dari sebuah motor DC dengan tegangan input 3 – 7,2V dan gearbox dengan rasio roda gigi 1:48. Motor DC dengan rentang tegangan tersebut digunakan pada proyek tugas akhir ini karena pemakaian daya yang digunakan cukup ringan. Hal ini disesuaikan dengan proyek ini yang merupakan *prototype* dari sistem pergerakan *drone*. Besarnya daya yang digunakan pada motor DC 3 – 7,2V dapat diketahui melalui persamaan 3.1 :

$$P = V \times I \tag{3.1}$$

Keterangan :

P = Daya yang digunakan pada motor DC (Watt)

V = Tegangan yang diberikan pada motor DC (Volt)

I = Arus yang dikeluarkan melalui *driver* menuju motor DC (Ampere)

Pada proyek tugas akhir ini digunakan modul *driver* motor DC L298N yang mampu mengeluarkan arus sampai 2A. Dengan menggunakan tegangan maksimal pada motor DC sebesar 7,2V maka besarnya daya yang digunakan adalah sebesar 14,4 Watt. Untuk menentukan besarnya daya dapat digunakan persamaan 3.1 sehingga memperoleh hasil sebagai berikut.

$$P = V \times I$$

$$P = 7{,}2\ V \times 2\ A$$

$$P = 14{,}4\ Watt$$

Pada proyek tugas akhir ini digunakan empat buah gearbox motor DC sehingga besarnya daya pada keempat motor tersebut adalah 57,6 Watt.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rasio roda gigi pada gearbox untuk meningkatkan torsi *input* dari putaran motor sehingga menghasilkan torsi *output* pada roda yang lebih tinggi dari torsi putaran motor. Pada proyek ini digunakan gearbox dengan rasio 1:48 sehingga torsi yang dihasilkan pada roda menjadi 48 kali lebih besar dari torsi yang dihasilkan motor. Hubungan antara torsi dengan rasio gearbox dapat dilihat pada persamaan 3.2 :

$$\text{Torsi Roda} = \text{Torsi Motor} \times \text{Rasio Roda}$$

$$\text{Torsi Roda} = \text{Torsi Motor} \times \frac{48}{1} \quad (3.2)$$

3.3.1.1 Metode Pengacakan *Delay Time Value*

Delay Time merupakan sebuah parameter nilai dalam program yang fungsi utamanya adalah memberikan waktu tunda untuk kondisi HIGH dan LOW pada *driver* motor DC. Perbedaan dari *delay time* HIGH dan *delay time* LOW akan mempengaruhi kecepatan dan arah motor [18]. Perubahan kecepatan dan arah motor akan mempengaruhi pergerakan robot dalam mencapai titik yang dituju dan juga mampu membuat robot berbelok. Jika keempat motor berada pada kecepatan dan arah yang sama maka robot mampu bergerak lurus, sedangkan jika kecepatan dan arah dari dua motor di bagian kanan berbeda dari dua motor bagian kiri (maupun sebaliknya) maka robot mampu berbelok. Pada proyek ini robot akan bergerak maju jika semua roda bergerak *counter- clockwise* (CCW) sementara robot akan bergerak mundur jika semua roda berputar *clockwise* (CW). Untuk pengaruh kecepatan dan arah putaran motor pada pergerakan robot dapat dilihat pada Tabel 3.1

Tabel 3.1 Pengaruh kecepatan dan arah putaran motor pada pergerakkan robot

Kecepatan		Arah Putaran		Pergerakan Robot
Motor Kanan	Motor Kiri	Motor Kanan	Motor Kiri	
Sama	Sama	CCW	CCW	Maju
Sama	Sama	CW	CW	Mundur
Lebih cepat	Lebih lambat	CCW	CW	Belok kiri
Lebih lambat	Lebih cepat	CW	CCW	Belok kanan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.3.2 Kamera *Webcam*

Pada proyek tugas akhir ini kamera digunakan untuk menuntun jalannya robot dengan membaca warna objek dan kode QR yang diletakkan di depan objek. Hasil pembacaan warna dan kode QR akan menentukan arah gerak robot pada saat berjalan dari titik *start* sampai berhenti pada titik *Emergency Landing Point*.

3.3.2.1 Pembacaan Kode QR

Kode QR (*Quick Response Code*) merupakan pengembangan dari *Bar-Code* yang dulunya merupakan kode satu dimensi menjadi kode dua dimensi dengan kemampuan menyimpan data lebih besar dibandingkan dengan *Bar-Code* sehingga kode QR mampu menyimpan data berupa kode angka, huruf, *binary* serta huruf kanji [19]. Kode QR ini sudah banyak digunakan untuk berbagai bidang. Pada proyek tugas akhir ini kode QR yang digunakan merupakan representasi data dari masing-masing gedung yang dituju. Kode QR pada proyek ini berjumlah tiga sesuai dengan jumlah gedung yang harus dituju. Data pada kode QR tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Data pada kode QR

Kode QR	Data
	VTOL 1
	VTOL 2
	VTOL 3

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Kode QR pada proyek ini akan dibaca oleh kamera kemudian kode-kode QR tersebut akan diterjemahkan sehingga data pada kode-kode QR mampu terbaca oleh robot. Penerjemahan kode QR pada robot menggunakan metode zbar yang merupakan pengimplementasian dari Pyzbar *library* pada pemrograman Python. Kemampuan dari Pyzbar *library* ini adalah mampu menerjemahkan *Bar-Code* maupun kode QR dari berbagai sumber [19].

3.3.2.2 Pembacaan Warna

Pembacaan warna pada proyek tugas akhir ini menggunakan metode segmentasi warna yang merupakan proses segmentasi dengan cara menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan [20]. Proses segmentasi warna pada kamera menggunakan metode deteksi HSV yang terdiri dari tiga elemen yaitu *Hue*, *Saturation*, dan *Value*. *Hue* merupakan suatu ukuran panjang gelombang pada warna dominan yang diterima oleh penglihatan, *Saturation* merupakan ukuran banyaknya cahaya putih yang bercampur pada *Hue*, sedangkan *Value* mengacu pada kecerahan warna. Jika dibandingkan dengan segmentasi RGB (*Red, Green, Blue*) maka segmentasi HSV masih memiliki tingkat akurasi yang lebih tinggi [20].

3.3.3 Modul Sensor Ultrasonik HC-SR04

Modul sensor ultrasonic HC-SR04 pada proyek tugas akhir ini digunakan untuk mengetahui jarak robot dengan objek di sekitarnya. Modul sensor ini bekerja dengan mengeluarkan gelombang ultrasonik kemudian membaca pantulan dari gelombang yang mengenai objek di depan robot seperti yang ditunjukkan pada persamaan 3.3 :

$$s = \frac{v \times t}{2} \tag{3.3}$$

Keterangan :

s = Jarak (cm)

v = Kecepatan suara (34300 cm/detik)

t = Waktu tempuh (detik)

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Waktu tempuh pada persamaan 3.3 dapat ditentukan dari selisih antara waktu ketika gelombang mulai dikirim dengan waktu penerimaan gelombang pantulan. Hasil perkalian antara kecepatan suara dengan waktu tempuh kemudian dibagi dua karena waktu tempuh menunjukkan waktu untuk satu gelombang yang terdiri dari gelombang terkirim dan gelombang terima.

Modul sensor ultrasonik yang digunakan pada robot berjumlah tiga yang diposisikan pada bagian depan, kiri, dan kanan. Pemosisian modul sensor ultrasonik di bagian depan, kiri, dan kanan bertujuan untuk mengetahui jarak benda di depan, kiri, dan kanan robot. Ketika jarak objek di depan, kiri, dan kanan robot sudah berhasil ditentukan maka robot akan bergerak menghindari objek di sekitarnya sehingga tidak menabrak objek tersebut.

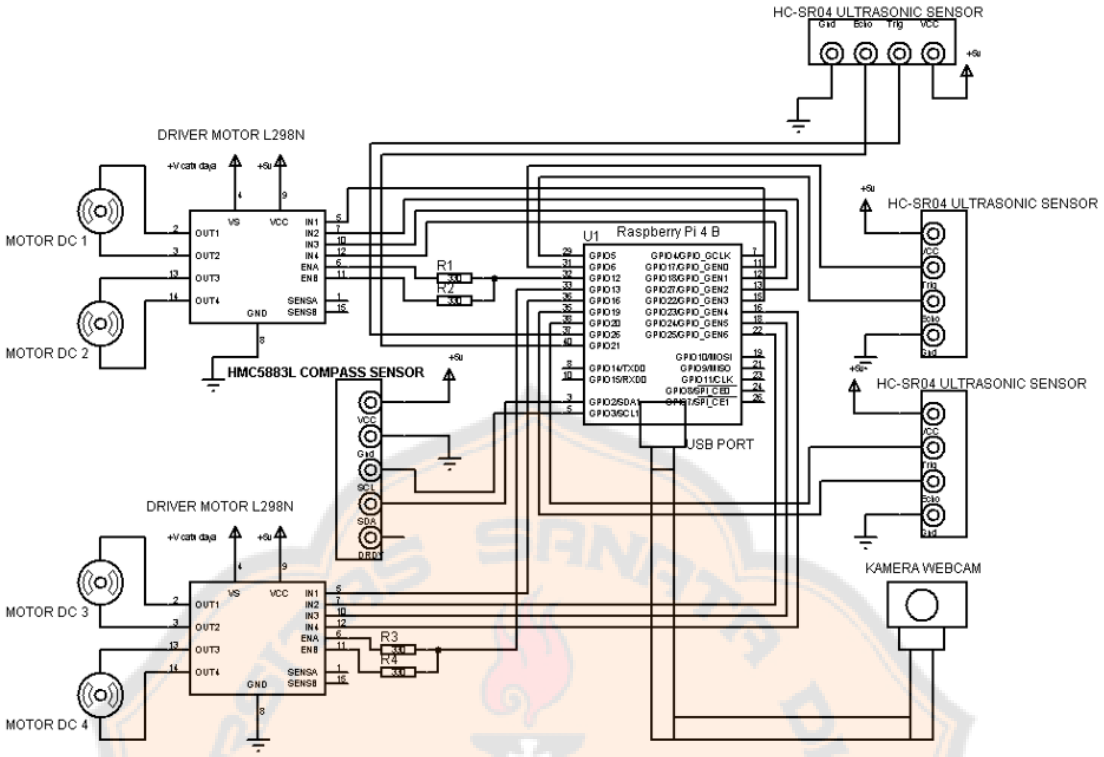
3.3.4 Modul Sensor Kompas HMC5883L

Modul sensor kompas HMC5883L pada proyek tugas akhir ini digunakan untuk menunjukkan arah hadap robot. Dengan menunjukkan arah hadap robot maka dapat diketahui arah yang akan dituju oleh robot. Sensor ini menggunakan sistem komunikasi I2C yang memiliki dua *port* data yaitu SDA dan SCL. Data yang didapat pada sensor kompas yaitu data 3 *axis* dengan panjang 16 bit data. Pengolahan data dari *axis* kompas yaitu dengan menghitung arcus tan antara sumbu X dan sumbu Y pada *axis* kompas sehingga mendapatkan nilai sudut hadap. Ketika sudut lebih dari nol maka data sudut kompas menjadi 360 dikurangi data sudut kompas sedangkan ketika data sudut kurang dari nol maka data sudut kompas ditambahkan 360 [21].

3.4 Perancangan Elektronik

Pada bab ini akan dijelaskan rancangan elektronik yang digunakan pada proyek tugas akhir ini. Rancangan elektronik terdiri dari rancangan pengkabelan perangkat keras elektronik dan rancangan pemakaian daya. Tujuan dari perancangan elektronik ini adalah agar pengkabelan perangkat keras elektronik mampu dipetakan dengan baik dan pemakaian daya mampu direncanakan dengan baik. Secara keseluruhan rencana pengkabelan perangkat keras elektronik pada robot dapat dilihat pada Gambar 3.11

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



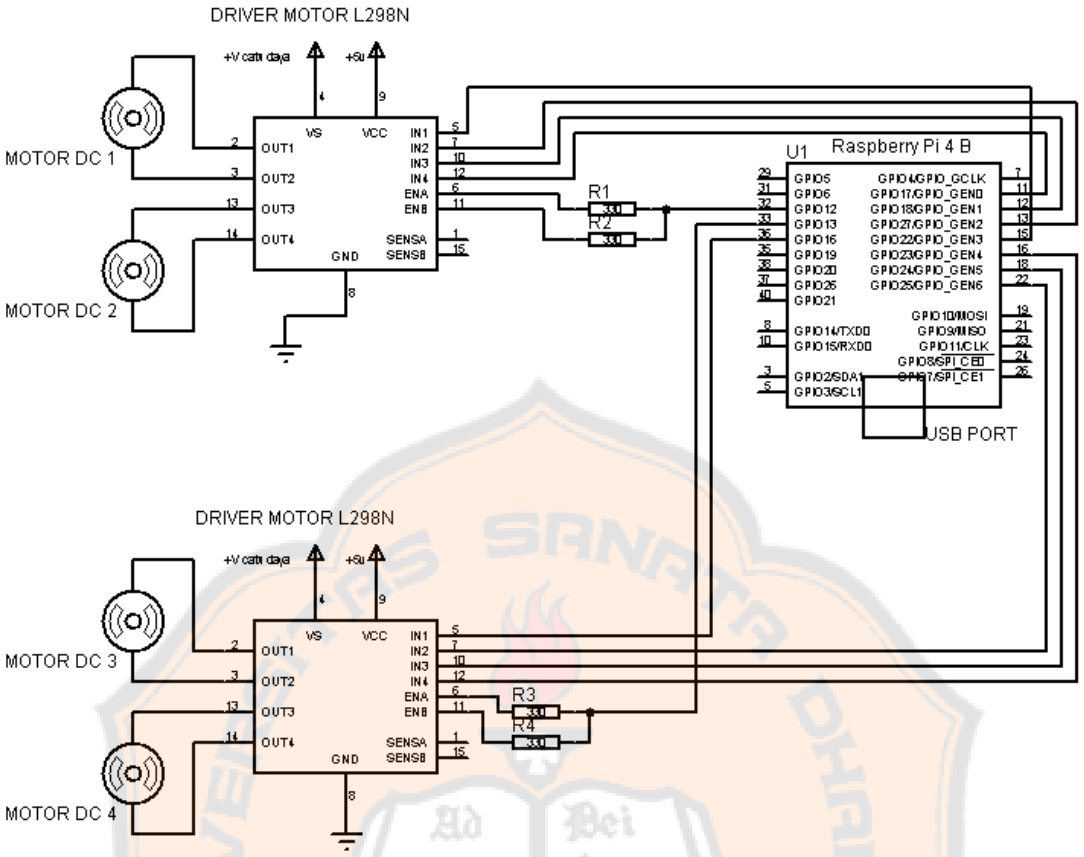
Gambar 3.6 Perancangan elektronik Robot Pelacak Lokasi

Pada perancangan elektronik di proyek ini, semua pin perangkat keras elektronik disambungkan pada pin Raspberry Pi. Setiap pin pada perangkat keras elektronik memiliki fungsi masing-masing dan dihubungkan pada pin Raspberry Pi sesuai dengan jenis data *input* dan *output* yang diperlukan.

3.4.1 Perancangan Elektronik *Driver* Motor DC

Bagian ini menjelaskan perancangan elektronik antara *driver* motor, motor DC, dan Raspberry Pi. Pengkabelan dari *driver* ini dapat dilihat pada Gambar 3.12. Motor DC yang digunakan pada proyek ini berjumlah empat dan setiap dua motor DC disambungkan pada satu *driver*. Masukan digital dan PWM pada *driver* motor ditransmisikan melalui pin keluaran pada Raspberry Pi. Nama pin pada modul *driver* motor DC L298N dan pin yang dihubungkan pada Raspberry Pi dan motor DC dapat dilihat pada Tabel 3.3

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.7 Perancangan elektronik modul *driver* L298N

Tabel 3.3 *Input-Output* Pin Modul *Driver* L298N L298N

Driver	**Nama Pin**	**Perangkat Keras yang Disambungkan**	**Pin pada Perangkat Keras**
Driver 1	VS	Catu Daya	+12V
	GND	Catu Daya dan Raspberry Pi 4 B	Ground
	VCC	Raspberry Pi 4 B	+5V
	IN 1		Pin 15 (GPIO 22)
	IN 2		Pin 13 (GPIO 27)
	IN 3		Pin 12 (GPIO 18)
	IN 4		Pin 11 (GPIO 17)
	EN A		Pin 32 (GPIO 12/PWM0)
	EN B		
	OUT 1	Motor 1	V+
	OUT 2		V-
	OUT 3	Motor 2	V+
	OUT 4		V-

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 3.3 *Input-Output* Pin Modul *Driver* L298N (lanjutan)

Driver	**Nama Pin**	**Perangkat Keras yang Disambungkan**	**Pin pada Perangkat Keras**
Driver 2	VS	Catu Daya	+12V
	GND	Catu Daya dan Raspberry Pi 4 B	Ground
	VCC	Raspberry Pi 4 B	+5V
	IN 1		Pin 36 (GPIO 16)
	IN 2		Pin 22 (GPIO 25)
	IN 3		Pin 18 (GPIO 24)
	IN 4		Pin 16 (GPIO 23)
	EN A		Pin 33 (GPIO 13/PWM1)
	EN B		
	OUT 1	Motor 1	V+
	OUT 2		V-
	OUT 3	Motor 2	V+
	OUT 4		V-

Pin IN1 sampai IN4 pada kedua *driver* dihubungkan pada pin GPIO Raspberry Pi sebagai kendali logika HIGH dan LOW pada pin-pin tersebut. Dengan kendali logika HIGH dan LOW maka motor mampu berputar *clockwise* maupun *counter clockwise*. Pin EN A dan EN B pada kedua *driver* dihubungkan dengan pin PWM pada GPIO Raspberry Pi sebagai kendali sinyal PWM yang mempengaruhi kecepatan motor. Pin VCC merupakan pin untuk mengaktifkan MOSFET L298N yang membutuhkan tegangan 5V dan dihubungkan pada pin sumber tegangan 5V Raspberry Pi. Pin VS dihubungkan dengan sumber tegangan 12V catu daya sementara pin GND dihubungkan dengan pin Ground pada catu daya dan Raspberry Pi dengan tujuan untuk menyamakan tegangan referensi antara catu daya dengan Raspberry Pi sebagai perangkat pengendali.

Pemakaian daya pada *driver* L298N ditentukan dengan besarnya tegangan dan arus yang dibutuhkan pada setiap pin. Tegangan keluaran pada setiap pin GPIO Raspberry Pi sebesar 3,3V dan arus keluaran bisa bervariasi mulai dari 2mA sampai 16mA. Pada perancangan kebutuhan daya ini digunakan arus maksimal yang dapat dikeluarkan GPIO Raspberry Pi yaitu sebesar 16mA. Pemakaian arus untuk mengaktifkan motor mampu mencapai 2A dengan tegangan maksimal yang mampu dikeluarkan pada pin OUT sebesar 7,2V. Dengan menggunakan persamaan 3.1 maka total keseluruhan rencana pemakaian daya pada *driver* L298N adalah sebesar 163,99 Watt yang ditunjukkan pada Tabel 3.4.

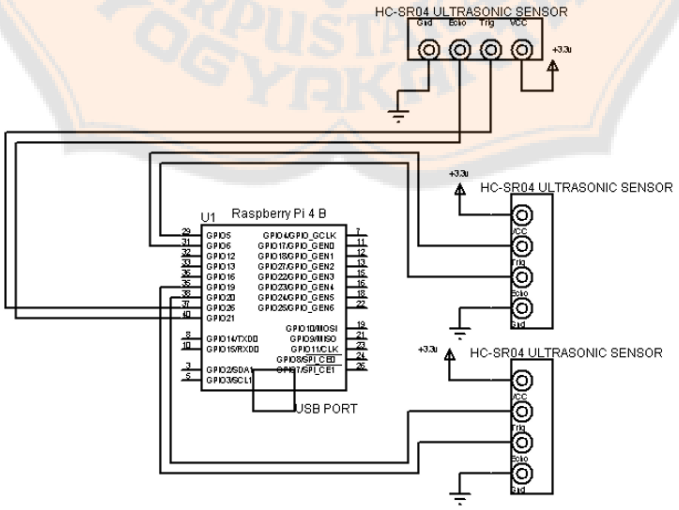
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 3.4 Rencana pemakaian daya *driver* motor DC L298N

Pin	Tegangan (V)	Arus (A)	Daya (Watt)	Jumlah Pin	Total Pemakaian Daya tiap Pin (Watt)
VCC	5	16m	0,08	2	0,16
VS	12	2	24	2	48
IN1	3,3	16m	0,053	2	0,11
IN2	3,3	16m	0,053	2	0,11
IN3	3,3	16m	0,053	2	0,11
IN4	3,3	16m	0,053	2	0,11
EN A	3,3	16m	0,053	2	0,11
EN B	3,3	16m	0,053	2	0,11
OUT 1	7,2	2	14,4	2	28,8
OUT 2	7,2	2	14,4	2	28,8
OUT 3	7,2	2	14,4	2	28,8
OUT 4	7,2	2	14,4	2	28,8
Total Pemakaian Daya (Watt)					163,99

3.4.2 Perancangan Elektronik Modul Sensor Ultrasonik HC-SR04

Bagian ini menjelaskan perancangan elektronik antara modul sensor ultrasonik HC-SR04 dengan Raspberry Pi seperti pada Gambar 3.13. Modul ini memiliki pin Trigger dan Echo yang keduanya menggunakan sinyal digital. Pin Trigger digunakan untuk memicu keluarnya gelombang ultrasonik, sedangkan pin Echo digunakan untuk menyalurkan data hasil pembacaan pantulan gelombang ke kontroler. Detail sambungan pin antara modul sensor ultrasonik HC-SR04 dengan Raspberry Pi dapat dilihat pada Tabel 3.5



Gambar 3.8 Perancangan elektronik modul sensor ultrasonik HC-SR04

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 3.5 *Input-output* pin modul sensor ultrasonik HC-SR04

Modul	Nama Pin	Perangkat Keras yang Disambungkan	Pin pada Perangkat Keras
Modul 1	VCC	Raspberry Pi 4 B	+3,3V
	GND		Ground
	Trigger		Pin 31 (GPIO 6)
	Echo		Pin 29 (GPIO 5)
Modul 2	VCC		+3,3V
	GND		Ground
	Trigger		Pin 38 (GPIO 20)
	Echo		Pin 35 (GPIO 19)
Modul 3	VCC		+3,3V
	GND		Ground
	Trigger		Pin 37 (GPIO 26)
	Echo		Pin 40 (GPIO 21)

Pin Trigger pada ketiga modul dihubungkan pada pin GPIO Raspberry Pi sebagai kendali lamanya sinyal ultrasonik yang dikeluarkan, sedangkan pin Echo yang masing-masing dihubungkan pada GPIO akan memberikan input pada GPIO sebagai hasil dari penerimaan sinyal pantulan ultrasonik. Pin VCC merupakan pin untuk mengaktifkan IC HC-SR04 yang membutuhkan tegangan dengan rentang 3,3 sampai 5V dan dihubungkan pada pin sumber tegangan 3,3V Raspberry Pi, sedangkan pin Ground dihubungkan pada pin Ground Raspberry Pi.

Pemakaian daya pada modul sensor ultrasonik HC-SR04 ditentukan dengan besarnya tegangan dan arus yang dibutuhkan pada setiap pin yang digunakan untuk *output*. Pemakaian arus dan tegangan untuk ketiga modul disesuaikan dengan tegangan dan arus maksimal yang mampu dikeluarkan oleh Raspberry Pi, yaitu 3,3V untuk tegangan dan 16mA untuk arus. Total keseluruhan pemakaian daya yang direncanakan pada modul sensor ultrasonik HC-SR04 adalah sebesar 0,32 Watt yang ditunjukkan pada Tabel 3.6.

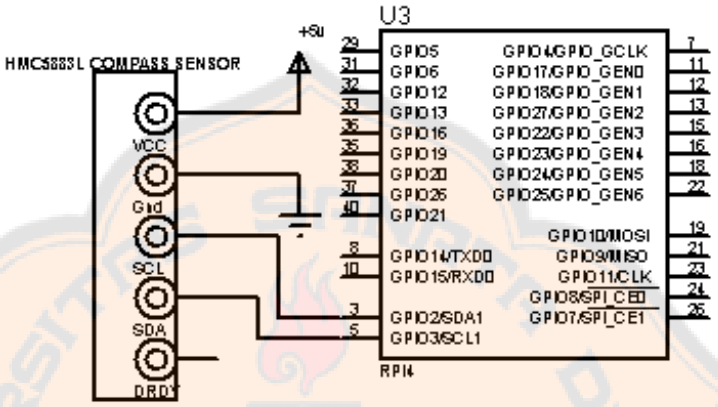
Tabel 3.6 Rencana Pemakaian Daya Modul Sensor Ultrasonik HC-SR04

Pin	Tegangan (V)	Arus (A)	Daya (Watt)	Jumlah Pin	Total Pemakaian Daya tiap Pin (Watt)
VCC	3,3	16m	0,053	3	0,158
Trigger	3,3	16m	0,053	3	0,158
Total Pemakaian Daya (Watt)					0,32

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.4.3 Perancangan Elektronik Sensor Kompas

Bagian ini menjelaskan perancangan elektronik antara modul sensor kompas HMC5883L dengan Raspberry Pi seperti pada Gambar 3.14. Modul ini memiliki pin SCL dan SDA yang akan disambungkan dengan *port* I2C pada Raspberry Pi. Detail sambungan pin antara modul sensor kompas HMC5883L dengan Raspberry Pi dapat dilihat pada Tabel 3.7



Gambar 3.9 Perancangan elektronik modul sensor kompas

Tabel 3.7 *Input-Output* Pin Modul Sensor Kompas HMC5883L

Nama Pin	Perangkat Keras yang Disambungkan	Pin pada Perangkat Keras
VCC	Catu Daya	+5V
GND	Catu Daya dan Raspberry Pi 4 B	Ground
SCL	Raspberry Pi 4 B	Pin 3 (GPIO 2/SDA)
SDA		Pin 5 (GPIO 3/SCL)

Pin SDA dan SCL pada modul sensor kompas dihubungkan pada pin SDA dan SCL GPIO Raspberry Pi sebagai komunikasi data antara modul sensor kompas dengan Raspberry Pi. Pin VCC merupakan pin untuk mengaktifkan IC HMC5883L yang membutuhkan tegangan dengan rentang 3 sampai 5V dan dihubungkan pada pin sumber tegangan 5V catu daya, sedangkan pin Ground dihubungkan pada pin Ground Raspberry Pi dan catu daya.

Pemakaian daya pada modul sensor kompas HMC5883L ditentukan berdasarkan besarnya tegangan dan arus yang dibutuhkan pada setiap pin yang digunakan untuk

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

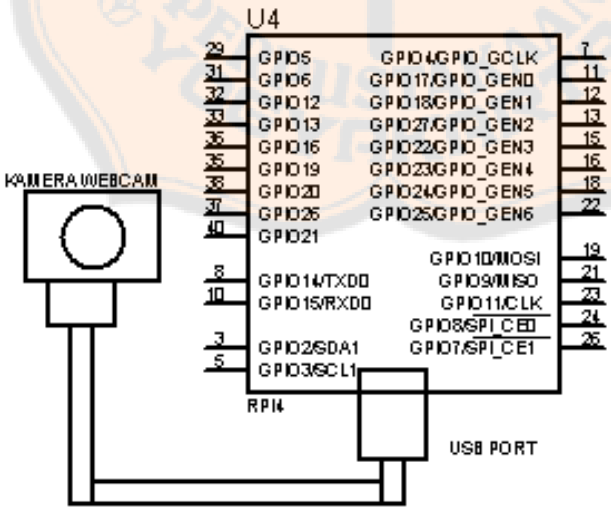
komunikasi data. Pemakaian arus dan tegangan untuk modul sensor kompas disesuaikan dengan tegangan yang mampu dikeluarkan oleh GPIO Raspberry Pi yaitu 3,3V dan arus yang dibutuhkan oleh modul sensor kompas yaitu 130 µA. Total keseluruhan pemakaian daya yang direncanakan pada modul sensor kompas HMC5883L adalah sebesar 0,00151 Watt yang ditunjukkan pada Tabel 3.8

Tabel 3.8 Rencana Pemakaian Daya Modul Sensor Kompas HMC5883L

Pin	Tegangan (V)	Arus (A)	Daya (Watt)	Jumlah Pin	Total Pemakaian Daya tiap Pin (Watt)
VCC	5	130 µ	0,00065	1	0,00065
SDA	3,3	130 µ	0,00043	1	0,00043
SCL	3,3	130 µ	0,00043	1	0,00043
Total Pemakaian Daya (Watt)					0,00151

3.4.4 Perancangan Elektronik Kamera *Webcam*

Bagian ini menjelaskan perancangan elektronik antara kamera *webcam* dengan Raspberry Pi seperti pada Gambar 3.15. Kamera *webcam* pada umumnya cukup disambungkan dengan *port* USB yang berada pada perangkat kontroler. Pada Raspberry Pi model 4 B terdapat *port* USB yang pada proyek ini digunakan sebagai sumber tegangan untuk kamera *webcam* sekaligus sebagai jalur komunikasi data.



Gambar 3.10 Perancangan elektronik kamera *webcam*

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pemakaian daya pada kamera *webcam* disesuaikan dengan besarnya tegangan dan arus yang dibutuhkan yaitu sebesar 5V untuk tegangan dan 100mA sampai 2A untuk arus. Dengan menggunakan arus maksimal yang dibutuhkan maka pemakaian daya yang direncanakan pada kamera *webcam* ini adalah sebesar 10 Watt.

3.4.5 Perancangan Pemakaian Daya Keseluruhan

Setelah mengetahui rencana pemakaian daya dari seluruh perangkat keras elektronik yang dihubungkan dengan Raspberry Pi maka total daya tersebut akan ditambahkan dengan daya yang dibutuhkan oleh Raspberry Pi. Pada Raspberry Pi 4 B dibutuhkan suplai tegangan sebesar 5V dengan suplai arus sebesar 3A. Dengan besar suplai tegangan dan arus tersebut maka pemakaian daya yang dibutuhkan oleh Raspberry Pi 4 B adalah sebesar 15 Watt. Total keseluruhan rencana penggunaan daya pada robot ini adalah sebesar 189,41 Watt yang ditunjukkan pada Tabel 3.9.

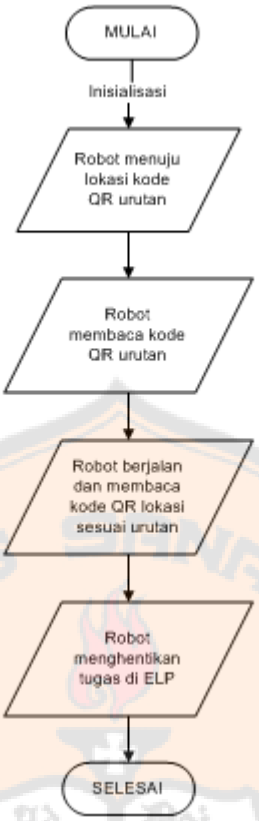
Tabel 3.9 Rencana Pemakaian Daya Keseluruhan

Perangkat	**Pemakaian Daya (Watt)**
Modul *driver* motor DC L298N	163,99
Modul sensor ultrasonik HC-SR04	0,32
Modul sensor kompas HMC5883L	0,00151
Kamera *webcam*	10
Raspberry Pi 4 B	15
Total Pemakaian Daya Keseluruhan (Watt)	189,3115

3.5 Perancangan Pemrograman

Bagian ini menjelaskan rancangan pemrograman proses kerja pada Robot Pelacak Lokasi Berbasis Raspberry Pi. Rancangan pemrograman berisi tentang gambaran dasar dari program yang akan dibuat dengan mengatur alur *input*, *process*, dan *output* pada sistem kerja robot. Penjelasan rancangan pemrograman dilengkapi dengan *flowchart* (diagram alir) untuk menjelaskan urutan program yang akan dibuat. Rancangan pemrograman akan dibagi menjadi tiga bagian, yaitu rancangan pemrograman kamera, sensor ultrasonik, dan sensor kompas. *Flowchart* rancangan pemrograman Robot Pelacak Lokasi Berbasis Raspberry Pi secara umum dapat dilihat pada Gambar 3.16.

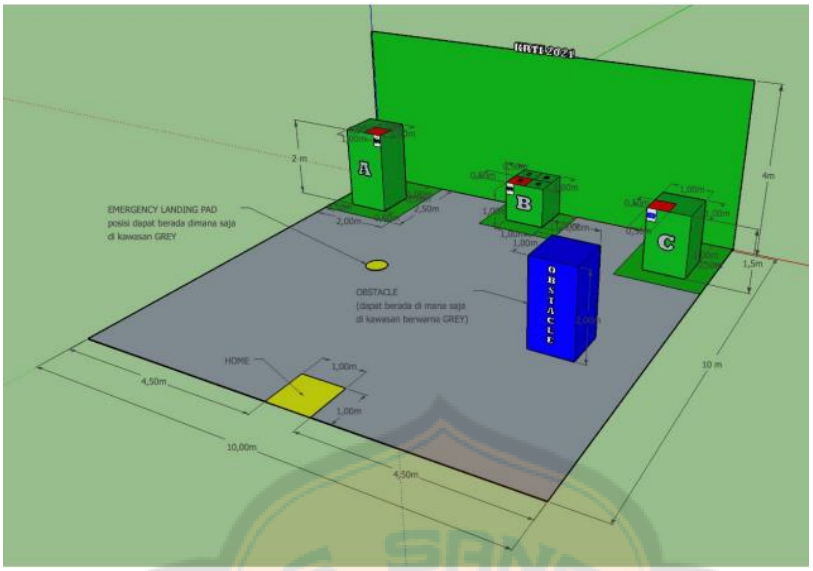
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.11 *Flowchart* rancangan pemrograman umum

Secara umum, urutan pemrograman pada Robot Pelacak Lokasi Berbasis Raspberry Pi dimulai dari proses robot berjalan menuju lokasi kode QR. Robot berjalan dari kotak kuning yang disebut HOME menuju gedung *obstacle* yang berwarna biru. Di sisi depan bagian atas gedung *obstacle* terdapat kode QR urutan yang harus dibaca oleh robot. Hasil pembacaan kode QR urutan menentukan urutan gedung yang akan dituju oleh robot. Robot kemudian berjalan menuju gedung berwarna hijau dan membaca kode QR lokasi yang berada di sisi depan bagian atas gedung. Setelah robot membaca semua kode QR lokasi, robot akan berhenti di lingkaran kuning yang disebut ELP (*Emergency Landing Point*). Gambaran arena pengujian yang akan digunakan dapat dilihat pada Gambar 3.17. Gambar tersebut merupakan arena pengujian lomba VTOL KRTI yang digunakan sebagai dasar pembuatan tugas akhir ini.

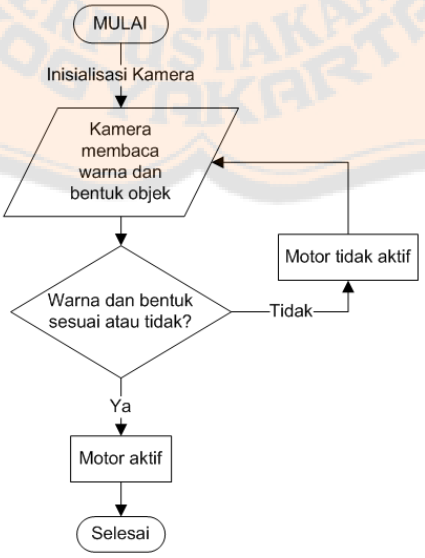
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 3.12 Gambaran arena pengujian [1]

3.5.1 Perancangan Pemrograman Kamera

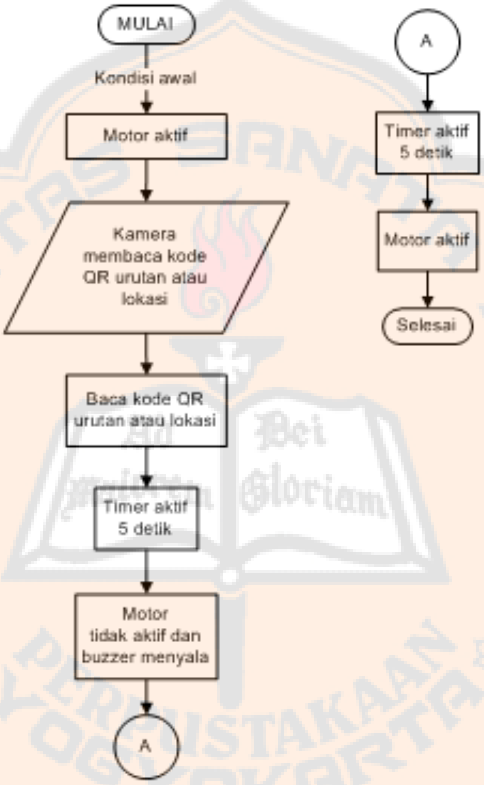
Bagian ini menjelaskan rancangan pemrograman untuk alur kerja pada kamera. Pada rancangan pemrograman ini, kamera digunakan untuk mengenali bentuk dan warna objek serta untuk membaca kode QR urutan dan kode QR lokasi. Hasil pembacaan kamera akan memicu motor untuk berputar. Gambaran alur kerja kamera pada saat membaca warna dan bentuk objek dapat dilihat melalui *flowchart* pada Gambar 3.18. Untuk *flowchart* pembacaan kode QR urutan dan kode QR lokasi dapat dilihat pada Gambar 3.19.



Gambar 3.13 *Flowchart* pembacaan bentuk dan warna objek

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Fungsi pertama kamera adalah untuk mengenali warna dan bentuk objek. Tujuan dari fungsi ini adalah agar robot mampu mengenali objek dan mampu menentukan apakah robot harus berjalan atau berhenti. Menurut *flowchart* pada Gambar 3.18 kamera akan memengaruhi pergerakan robot melalui pengendalian on-off pada penggerak robot, yaitu motor DC. Ketika kamera mendeteksi warna dan bentuk objek maka akan ditentukan apakah warna dan bentuk objek tersebut sesuai dengan perintah pada program atau tidak. Jika sesuai maka motor akan aktif, jika tidak sesuai maka motor tidak aktif.



Gambar 3.14 *Flowchart* pembacaan kode QR urutan dan lokasi

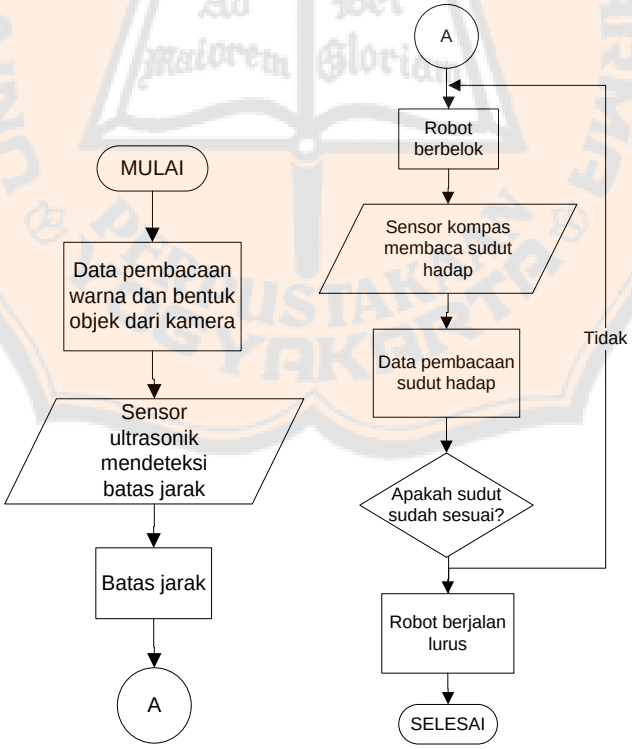
Fungsi kedua kamera adalah untuk membaca kode QR. Tujuan dari fungsi ini adalah agar robot mampu menentukan urutan gerak pada gedung lokasi melalui pembacaan kode QR urutan dan menunjukkan bahwa robot sudah mencapai gedung lokasi melalui pembacaan kode QR lokasi. Menurut *flowchart* pada Gambar 3.19 kamera akan memengaruhi pergerakan robot melalui pengendalian on-off pada motor DC dengan bantuan *timer*. Ketika kamera membaca kode QR (urutan atau lokasi) maka kamera akan mendapatkan data dari kode QR dan memicu *timer* untuk aktif selama lima detik. Setelah lima detik maka motor DC akan tidak aktif dan *buzzer* akan menyala sebagai indikator

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

kamera berhasil membaca kode QR. Setelah itu *timer* akan kembali aktif selama lima detik kemudian motor DC akan kembali aktif.

3.5.2 Perancangan Pemrograman Sensor Kompas dan Sensor Ultrasonik

Pada bagian ini akan dijelaskan rancangan pemrograman untuk alur kerja pada sensor kompas dan sensor ultrasonik. Menurut *flowchart* pada Gambar 3.20 sensor kompas digunakan untuk mendeteksi sudut hadap pada robot. Deteksi sudut tersebut akan digabung dengan data hasil deteksi warna dan bentuk objek pada kamera sehingga Raspberry Pi mampu menentukan apakah robot akan berjalan lurus atau berbelok. Robot akan terus berbelok sampai sudut yang diminta sudah terpenuhi. Ketika sudah mencapai sudut yang diminta maka robot akan berjalan lurus menuju objek. Sementara itu, sensor ultrasonik digunakan untuk mendeteksi batas jarak antara robot dengan gedung lokasi atau gedung *obstacle*. Ketika sensor ultrasonik mendeteksi batas jarak maka robot akan berbelok sampai sensor kompas mendeteksi sudut hadap yang digunakan sebagai syarat agar robot mampu berjalan menjauhi gedung tersebut.



Gambar 3.15 *Flowchart* sensor kompas dan sensor ultrasonik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB IV
HASIL DAN PEMBAHASAN

Bab ini akan membahas mengenai hasil dari perancangan melalui penelitian yang dilakukan pada robot empat roda. Adapun yang akan dibahas pada bab ini adalah perubahan perancangan, hasil perancangan perangkat keras dan pemrograman, serta analisa hasil pengujian yang telah dilakukan. Data pada pengujian akan diambil melalui pembacaan kamera dan sensor, serta hasil kerja dari penggerak robot.

4.1 Perubahan Perancangan

Perubahan perancangan pada penelitian ini meliputi perubahan perancangan perangkat keras, perancangan pemrograman, dan perancangan elektronik. Perubahan yang dilakukan pada perancangan perangkat keras meliputi pengurangan jumlah sensor ultrasonik yang sebelumnya berjumlah tiga menjadi satu, dihilangkannya sensor kompas dan *buzzer* pada robot, penggantian kamera *webcam* dengan modul kamera Raspberry Pi OV5647, serta rencana penggunaan catu daya yang sebelumnya satu sumber dibedakan menjadi dua sumber dengan keluaran tegangan yang berbeda. Pengaruh dari pengurangan perangakat keras ini adalah berubahnya perancangan pemrograman dan juga perancangan elektronik pada robot.

4.1.1 Perubahan Perancangan Perangkat Keras

Pada bagian ini akan dibahas mengenai perangkat keras yang digantikan dan dikurangi dalam perancangan robot serta pengaruhnya terhadap perubahan perancangan pemrograman. Perangkat keras yang digantikan pada robot yaitu kamera *webcam*. Penggunaan kamera *webcam* pada robot digantikan dengan modul kamera Raspberry Pi OV5647. Alasan digunakannya modul kamera ini sebagai pengganti *webcam* adalah karena modul ini lebih kompatibel pada Raspberry Pi sehingga lebih mudah untuk digunakan. Modul kamera ini menggunakan sensor Omnivision OV5647 yang merupakan sensor gambar CMOS 5 megapixel dengan kemampuan menangkap gambar pada resolusi 5 megapixel 2592 x 1944 dan menangkap video pada resolusi 720p/60 sampai 1080p/30

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

high-definition (HD). Spesifikasi lebih lengkap dari modul kamera Raspberry Pi OV5647 dapat dilihat pada Tabel 4.1

Gambar 4.1 Modul kamera Raspberry Pi OV5647

Sumber : www.arucam.com/standard raspberry camera

Tabel 4.1 Spesifikasi modul kamera Raspberry Pi OV5647

Sensor	Omnivision OV5647
Ukuran Sensor	1/4” (OV5647)
Resolusi	5MP 2592×1944
Video	1080p 30 fps, 720p 60fps, 480p 90fps
Field of View (FOV)	54°(H), 41°(V)
Tipe Fokus	Fixed Focus
Panjang Fokus	3.60mm
Jarak Fokus	1 m – Tak terbatas
Ekuivalen Lensa Full-frame SLR	35 mm
Dimensi	25×24 mm
Arus Puncak	300mA

Perangakat keras yang dikurangi pada proyek tugas akhir ini adalah sensor ultrasonik. Pengurangan tiga sensor ultrasonik menjadi satu sensor ultrasonik dilakukan karena data pembacaan jarak dari satu sensor ultrasonik sudah mampu membuat jarak aman antara robot dengan objek di depan robot. Untuk jarak aman antara robot dengan objek di kanan maupun kiri dapat tercapai dengan baik karena robot mampu berbelok tanpa mengenai objek di sekitarnya. Setelah berbelok, robot diperintahkan untuk maju sehingga robot mampu melewati objek di sisi kanan maupun sisi kiri.

Perubahan pada sensor juga dilakukan pada sensor kompas. Pada proyek tugas akhir ini sensor kompas dihilangkan karena proses kerja Raspberry Pi yang sudah cukup

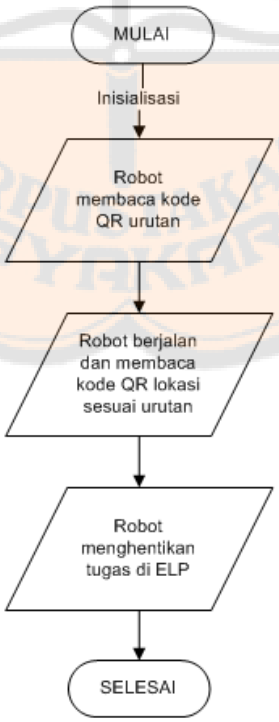
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

berat untuk mengendalikan gerakan robot melalui data pembacaan kamera dan pendeteksian sensor ultrasonik.

Perangkat terakhir yang tidak jadi digunakan pada proyek tugas akhir ini adalah *buzzer*. Fungsi dari *buzzer* adalah memberitahu pengguna bahwa robot sudah berhasil mendeteksi kode QR. Akan tetapi fungsi ini sudah tercapai melalui tangkapan kode QR pada layar kamera yang sudah disertai dengan hasil pembacaan kode QR.

4.1.2 Perubahan Perancangan Pemrograman

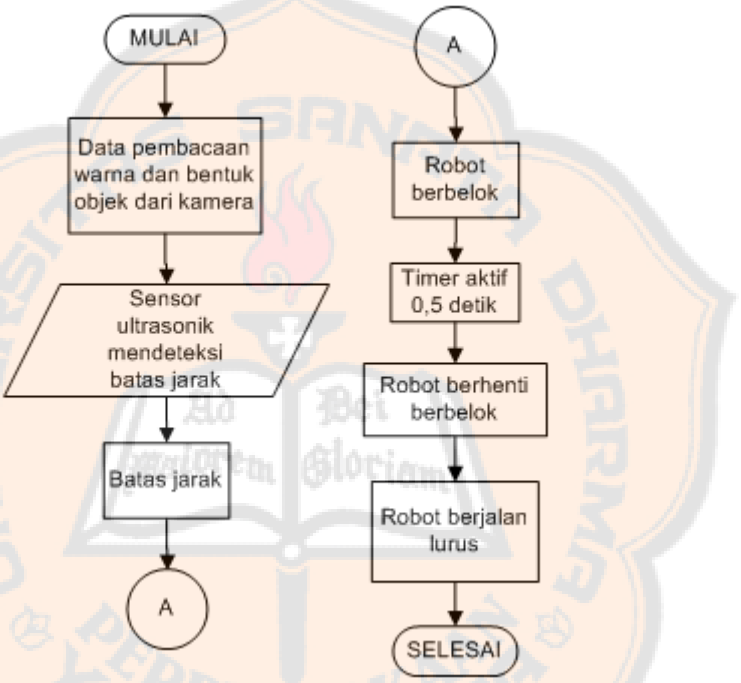
Penyebab pertama dari perubahan pemrograman pada robot adalah dihilangkannya proses pembacaan kode QR urutan sebagai syarat sebelum robot menuju ke masing-masing gedung. Kode QR urutan pada awalnya akan digunakan sebagai *input* bagi robot untuk mengetahui urutan gedung yang harus dituju. Akan tetapi, merujuk pada aturan VTOL KRTI 2021 yang tidak menuliskan kode QR urutan sebagai *input* awal untuk robot maka proses pembacaan kode QR urutan pada robot dihilangkan dan robot hanya membaca kode QR pada masing-masing gedung. Perubahan *flowchart* pemrograman utama dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Flowchart* pemrograman utama dengan perubahan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Penyebab perubahan pemrograman yang kedua adalah berkurangnya jumlah sensor ultrasonik yang digunakan serta sensor kompas yang tidak jadi digunakan pada robot. Proses pembelokan robot ke arah kanan maupun kiri dengan sudut 90 derajat dilakukan melalui pengaturan lamanya waktu yang digunakan robot untuk berbelok. Setelah berbelok, robot akan berhenti berbelok sebelum berjalan lurus menuju gedung selanjutnya. Perubahan *flowchart* pemrograman untuk pengaruh penggunaan sensor pada robot dapat dilihat pada Gambar 4.3.

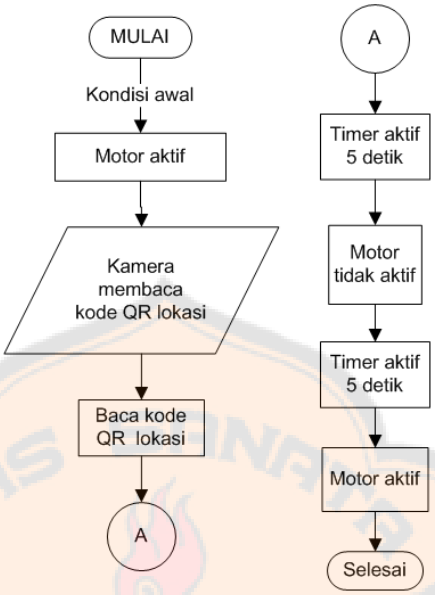


Gambar 4.3 Perubahan *flowchart* pemrograman sensor ultrasonik

Penyebab perubahan pemrograman yang terakhir adalah *buzzer* yang tidak jadi digunakan pada robot. Fungsi *buzzer* sebagai indikator bahwa robot berhasil membaca kode QR mampu digantikan dengan hasil pembacaan kode QR yang tertampil pada layar kamera dan kolom terminal aplikasi pemrograman. Dihilangkannya *buzzer* dan kode QR urutan memengaruhi perubahan pemrograman pada kamera. Dengan dua penyebab perubahan tersebut maka pemrograman kamera untuk pembacaan kode QR hanya digunakan untuk membaca kode QR lokasi dan hasil pembacaan tersebut hanya

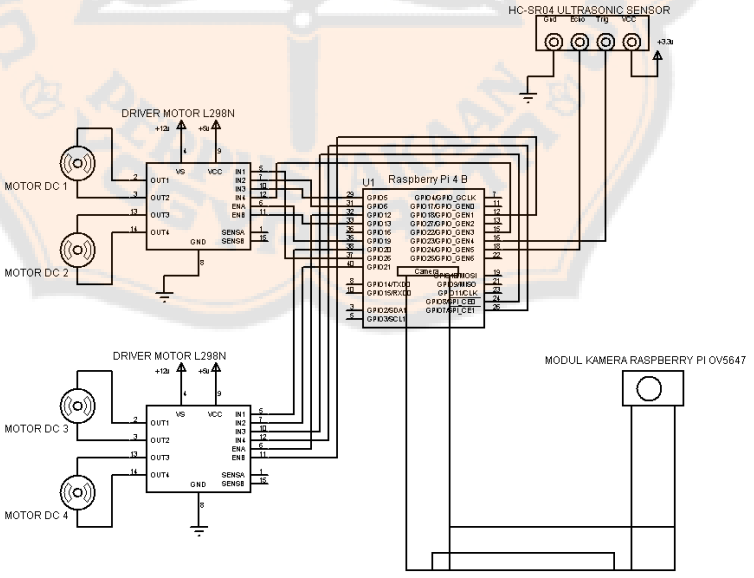
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

memengaruhi aktif tidaknya motor pada robot. Perubahan *flowchart* pemrograman untuk pembacaan kode QR lokasi melalui kamera dapat dilihat pada Gambar 4.4.



Gambar 4.4 Perubahan *flowchart* pembacaan kode QR lokasi

4.1.3 Perubahan Perancangan Elektronik



Gambar 4.5 Skematik perubahan rancangan elektronik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Perubahan perancangan elektronik yang dilakukan pada proyek tugas akhir ini meliputi perubahan rencana pengkabelan dan pemakaian daya. Skematik rangkaian setelah mengalami perubahan rencana pengkabelan dapat dilihat pada Gambar 4.5. Dengan berkurangnya komponen yang digunakan membuat ketersediaan pin pada GPIO Raspberry Pi 4 B menjadi lebih banyak sehingga urutan pengkabelan pada pin GPIO menjadi berubah. Selain itu penambahan modul kamera Raspberry Pi OV5647 pada robot juga memengaruhi perubahan urutan pengkabelan. Perubahan urutan pengkabelan pada pin ini meliputi perangkat *driver* motor DC, sensor ultrasonik, dan modul kamera Raspberry Pi OV5647 yang dapat dilihat pada Tabel 4.2. Untuk pengkabelan sumber daya pada perangkat keras dan keluaran tegangan pada motor tidak mengalami perubahan.

Tabel 4.2 Pengkabelan pada Raspberry Pi

Perangkat Keras	**Nama Sambungan**	**Sambungan pada Raspberry Pi 4 B**	**Nama Pin (BCM)**	**Nama Pin (Board)**
Modul *driver* L298N (1)	Pin IN 1	GPIO	GPIO 26	37
	Pin IN 2		GPIO 6	31
	Pin EN A		GPIO 19 (PCM_FS)	35
	Pin IN 3		GPIO 5	29
	Pin IN 4		GPIO 22	15
	Pin EN B		GPIO 13 (PWM1)	33
Modul *driver* L298N (2)	Pin IN 1	GPIO	GPIO 20 (PCM_DIN)	38
	Pin IN 2		GPIO 21 (PCM_DOUT)	40
	Pin EN A		GPIO 12 (PWM0)	32
	Pin IN 3		GPIO 8 (CE0)	24
	Pin IN 4		GPIO 7 (CE1)	26
	Pin EN B		GPIO 18 (PCM_CLK)	12
Modul sensor ultrasonik HC-SR04	Pin Trigger		GPIO 23	16
	Pin Echo		GPIO 24	18
Modul kamera Raspberry Pi OV5647	Kabel fleksibel	Slot *camera*	-	-

Perubahan rencana pemakaian daya pada robot disesuaikan dengan besarnya daya yang dibutuhkan dari setiap perangkat keras yang jadi digunakan. Dengan berkurangnya

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

perangkat keras atau komponen yang digunakan maka pemakaian daya pada robot ini juga berkurang. Rencana pemakaian perangkat keras yang sebelumnya menggunakan 3 buah sensor ultrasonik dikurangi menjadi 1 buah sensor ultrasonik sehingga rencana pemakaian daya yang sebelumnya sebesar 0,32 Watt pada proyek ini berkurang menjadi 0,11 Watt. Kemudian modul sensor kompas yang sebelumnya akan digunakan pada pengujian ini tidak jadi digunakan. Penambahan modul kamera Raspberry Pi OV5647 pada robot juga menyebabkan perubahan pada pemakaian daya. Modul kamera Raspberry Pi OV5647 membutuhkan arus maksimal sebesar 300mA dengan tegangan 5V sehingga daya yang dibutuhkan pada modul kamera Raspberry Pi sebesar 1,5 Watt. Dengan perubahan pemakaian daya tersebut maka total pemakaian daya pada robot ini adalah 180,6 Watt. Detail pemakaian daya keseluruhan pada robot proyek tugas akhir ini dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pemakaian daya keseluruhan

Perangkat	**Jumlah**	**Pemakaian Daya (Watt)**
Modul *driver* motor DC L298N	2	163,99
Modul sensor ultrasonik HC-SR04	1	0,11
Modul kamera Raspberry Pi OV5647	1	1,5
Raspberry Pi 4 B	1	15
Total Pemakaian Daya Keseluruhan (Watt)		**180,6**

Suplai daya untuk robot dibedakan menjadi suplai daya untuk empat motor DC dan suplai daya untuk Raspberry Pi. Pembedaan suplai daya ini karena kebutuhan daya untuk empat motor DC lebih besar dibandingkan Raspberry Pi sehingga jika menggunakan satu sumber daya maka keluaran daya menjadi tidak stabil untuk kebutuhan Raspberry Pi. Untuk suplai tegangan empat motor DC digunakan baterai Lipo 11,1V 3s dengan keluaran arus sebesar 400mAh. Baterai ini dipilih sebagai suplai tegangan untuk motor DC karena mampu menghasilkan tegangan yang mendekati kebutuhan tegangan maksimal pada motor DC yang digunakan, yaitu 12V. Dengan menggunakan tegangan baterai yang mendekati 12V maka RPM yang dihasilkan mampu mencapai nilai maksimal sehingga robot mampu menyelesaikan pola gerakan dengan lebih cepat. Kebutuhan arus untuk setiap motor adalah

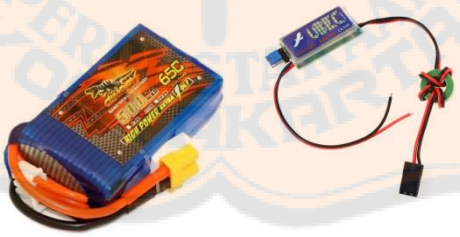
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2A, sehingga dengan menggunakan suplai arus dari baterai sebesar 400mAh maka kapasitas baterai akan habis dalam waktu 12 menit.

Gambar 4.6 Baterai Lipo 11,1V 3s

Sumber : www.ichibot.id/ baterai lipo chibo power 3s 400mah

Suplai tegangan untuk Raspberry Pi menggunakan baterai Lipo 7,4V 2s dengan keluaran arus sebesar 450mAh. Baterai dengan sepsifikasi tersebut digunakan untuk Raspberry Pi karena kebutuhan tegangan yang lebih kecil dan arus yang dibutuhkan mampu mengaktifkan Raspberry Pi sesuai dengan lama waktu pengaktifan robot. Keluaran tegangan dari baterai kemudian dihubungkan dengan UBEC 5V 3A sehingga menghasilkan tegangan dan arus yang sesuai dengan spesifikasi kebutuhan suplai tegangan dan arus pada Raspberry Pi. Dengan kebutuhan arus sebesar 3A dan suplai arus sebesar 450mAh maka kapasitas pada baterai akan habis dalam waktu 9 menit.



Gambar 4.7 Baterai Lipo 7,4V 2s dan UBEC 5V 3A

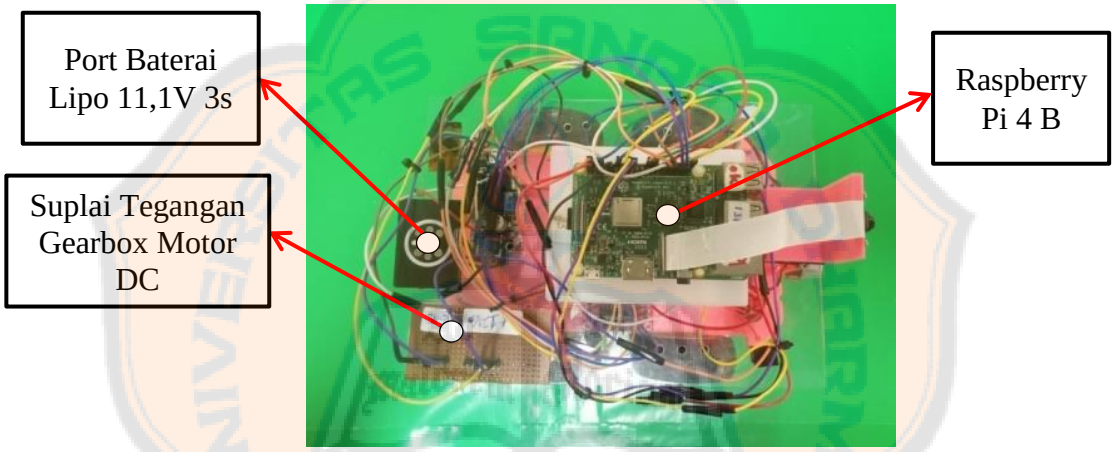
Sumber : Buaya Aerotech

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

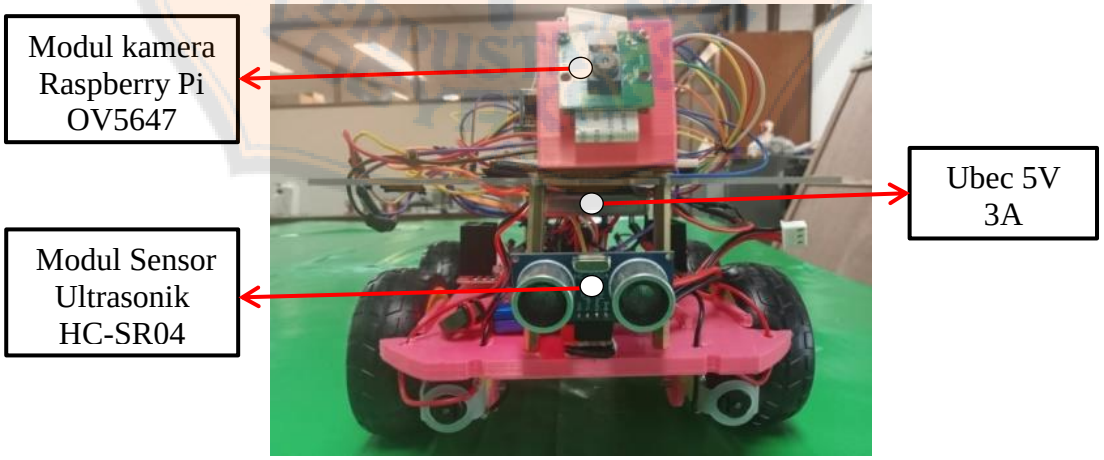
4.2 Hasil Perancangan Perangkat Keras

4.2.1 Bentuk Fisik Robot

Bagian ini menjelaskan bentuk fisik robot yang sudah terpasang dengan seluruh perangkat keras elektronik yang digunakan. Rangka robot terdiri dari rangka bagian atas dan rangka bagian bawah. Gambar 4.8 menunjukkan rangka bagian atas yang digunakan untuk meletakkan port baterai Lipo 11,1V 3s, PCB suplai tegangan untuk motor, Raspberry Pi 4 B. Adapun rangka atas sisi depan digunakan untuk meletakkan modul kamera Raspberry Pi OV5647 dan UBEC 5V 3A yang ditunjukkan pada Gambar 4.9.



Gambar 4.8 Rangka robot tampak atas

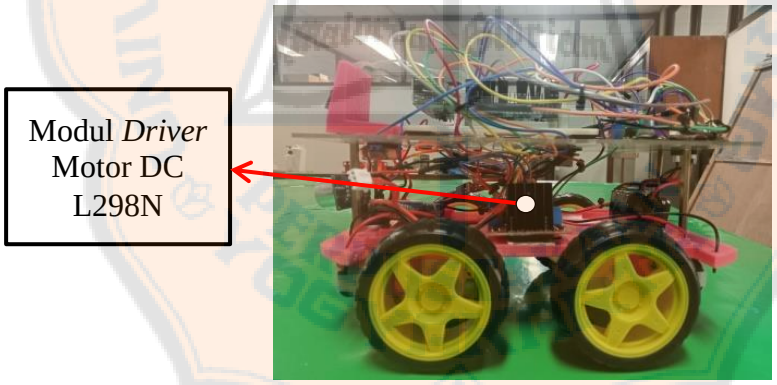


Gambar 4.9 Rangka robot tampak depan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rangka bagian bawah terdiri dari sisi atas dan sisi bawah. Gambar 4.10 dan Gambar 4.11 menunjukkan rangka bagian bawah sisi atas yang digunakan untuk meletakkan baterai Lipo 11,1V 3s, dua modul *driver* motor DC L298N, baterai Lipo 7,4V 2s, dan sebuah modul sensor ultrasonik HC-SR04. Adapun posisi peletakan modul sensor ultrasonik HC-SR04 dapat dilihat lebih jelas pada Gambar 4.9.

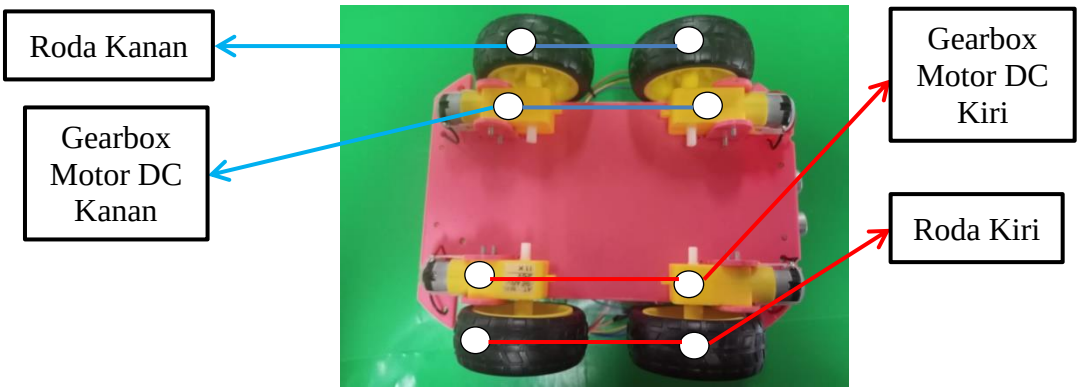
Gambar 4.10 Rangka robot tampak kanan



Gambar 4.11 Rangka robot tampak kiri

Gambar 4.12 menunjukkan rangka robot bagian bawah sisi bawah yang digunakan untuk meletakkan empat buah gearbox motor DC. Pada setiap gearbox motor DC terpasang roda dengan diameter 65mm.

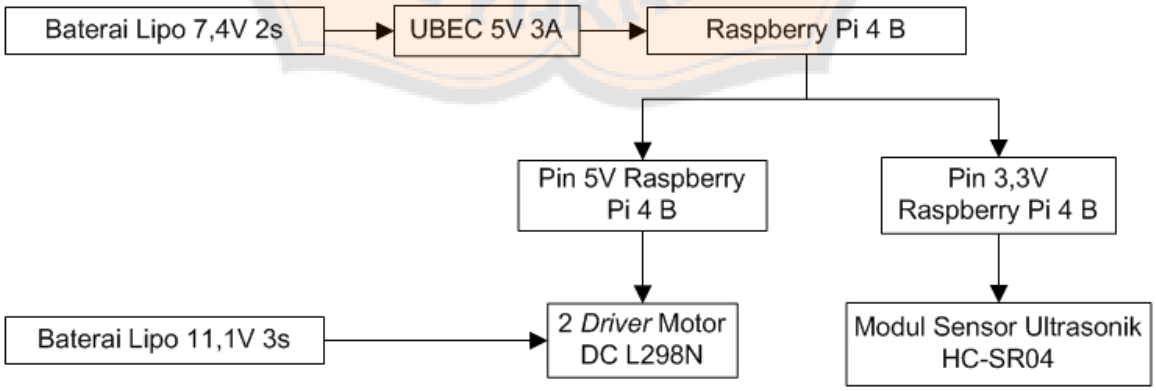
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 4.12 Rangka robot tampak bawah

4.2.2 Sistem Kelistrikan Robot

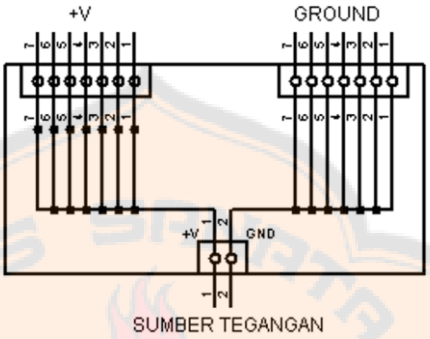
Sistem kelistrikan pada robot dibuat dengan tujuan untuk menyuplai tegangan pada setiap perangkat keras yang terpasang pada robot. Gambar 4.13 merupakan pembagian suplai tegangan pada setiap perangkat berdasarkan sumber tegangan yang digunakan. Pada robot ini sumber tegangan untuk Raspberry Pi 4 B menggunakan baterai Lipo 7,4V 2s yang keluarannya dihubungkan dengan UBEC 5V 3A sehingga menghasilkan besar tegangan dan arus yang dibutuhkan oleh Raspberry Pi 4 B. Adapun sumber tegangan untuk pin VCC *driver* motor DC menggunakan keluaran 5V dari pin Raspberry Pi, sedangkan untuk pin VS dihubungkan dengan keluaran tegangan dari baterai Lipo 11,1V 3s yang bertujuan agar motor mendapatkan tegangan yang mendekati 12V sehingga mampu menghasilkan torsi yang cukup untuk menggerakan robot. Pada modul sensor ultrasonik pin VCC sebagai sumber tegangan dihubungkan dengan keluaran 3,3V dari Raspberry Pi.



Gambar 4.13 Diagram blok sistem kelistrikan robot

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada sistem kelistrikan robot ditambahkan sebuah PCB (*Printed Circuit Board*) sebagai penghubung antara baterai Lipo 11,1V 3s dengan pin VS *driver* motor yang digunakan untuk menyuplai tegangan pada motor DC. Pin pada PCB dibagi menjadi pin sumber tegangan dan pin keluaran tegangan. Skematik dari PCB suplai tegangan untuk gearbox motor DC ditunjukkan pada Gambar 4.14.



Gambar 4.14 Skematik PCB sumber tegangan

4.3 Hasil Pengujian Alat

Dalam proyek tugas akhir ini terdapat beberapa hal yang diuji yaitu pengujian pola gerakan robot, pembacaan warna dan kode QR melalui kamera, pengukuran jarak melalui sensor ultrasonik, serta kecepatan roda berdasarkan arah gerakan robot.

4.3.1 Hasil Pengujian Pola Gerakan Robot

Pergerakan robot dalam proyek tugas akhir ini dibagi menjadi tiga pola arah gerakan sesuai dengan urutan gedung yang dituju. Pengujian ini dilakukan pada arena yang luasnya 1 : 9,4 dari luas arena yang digunakan pada lomba VTOL KRTI sehingga luas arena yang digunakan menjadi 3m x 3,55m. Pengecilan luas arena ini dilakukan karena kebutuhan arus dari baterai 11,1V 3s yang sangat banyak ketika menjalankan empat motor sehingga baterai akan cepat habis untuk sedikit pengujian jika menggunakan arena dengan luas sesuai pedoman VTOL KRTI.

Pada arena tersebut terdapat kotak kuning di sisi kanan yang merupakan titik *start* robot, sedangkan kotak kuning di sisi kiri merupakan titik berhenti robot yang pada lomba VTOL KRTI disebut *Emergency Landing Point* (ELP). Kotak kuning di sisi kanan

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

memiliki luas 64cm x 49cm, sedangkan untuk sisi kiri memiliki luas 21,5cm x 26cm. Gambar 4.15 dan Gambar 4.16 merupakan tampilan dari. kedua kotak kuning tersebut.

Gambar 4.15 Kotak kuning titik *start*

Gambar 4.16 Kotak kuning titik ELP

Target yang dituju pada arena ini adalah tiga buah gedung berwarna hijau dengan kode QR yang berada di sisi kiri atas gedung seperti yang ditampilkan pada Gambar 4.17. Ketiga gedung tersebut memiliki tinggi yang bervariasi. Gedung kanan memiliki tinggi 98cm, gedung tengah memiliki 50cm, dan gedung kiri memiliki tinggi 101cm. Luas dari tiap gedung adalah 1m x 1m dengan jarak antar gedung 24cm.

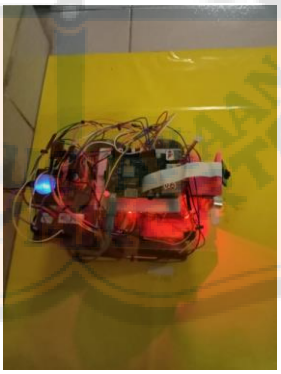
Gambar 4.17 Gedung target pada arena

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada Gambar 4.18 dapat dilihat tampilan arena secara keseluruhan. Peletakan kotak kuning untuk titik *start* diberi jarak 230cm dari gedung kanan, sedangkan untuk kotak kuning titik ELP diberi jarak 254cm dari gedung kiri.

Gambar 4.18 Arena pengujian pola gerakan robot

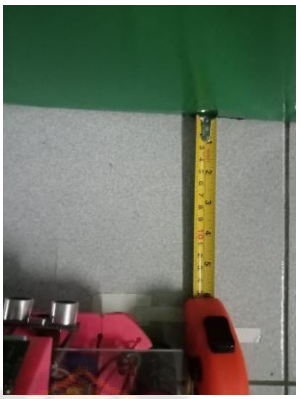
Langkah pengujian yang dilakukan pertama kali adalah memosisikan robot pada posisi tengah kotak *start*. Gambar 4.19 menunjukan pemosisian robot pada titik tengah kotak *start*, yaitu posisi ujung sensor ultrasonik sampai tepi depan kotak *start* diberi jarak 18cm dan posisi roda kanan robot sampai tepi kanan kotak *start* diberi jarak 23,5cm.



Gambar 4.19 Robot pada posisi tengah kotak *start*

Setelah robot berada pada posisi kotak *start* selanjutnya robot dijalankan menuju setiap gedung sesuai urutan pola gerak. Batas jarak yang ditentukan antara gedung dengan robot adalah 15cm seperti yang ditunjukkan pada Gambar 4.20. Setelah semua kode QR pada gedung terdeteksi maka robot akan berhenti di titik ELP. Jarak yang ditentukan untuk titik berhenti pada ELP adalah 258cm dari gedung kiri.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



Gambar 4.20 Batas jarak antara robot dengan gedung

Pada pengujian ini robot akan menjalankan pola gerakkan satu, yaitu robot akan berjalan mulai dari gedung kanan kemudian ke gedung tengah lalu gedung kiri. Tabel 4.4 menunjukkan hasil deteksi batas jarak melalui sensor ultrasonik pada gedung kanan. Pada pengujian di gedung kanan didapatkan rata-rata *error* batas jarak sebesar 38,6%.

Tabel 4.4 Deteksi Batas Jarak Gedung Kanan Pada Pola Gerakkan Satu

Pengujian	**Batas Jarak (cm)**	**Deteksi Batas Jarak (cm)**	**Kapasitas Baterai Setelah Dijalankan (%)**	***Error* (%)**
1	15	15	93	0,0
2	15	16,2	86	8,0
3	15	16,3	79	8,7
4	15	17	72	13,3
5	15	17,76	65	18,4
6	15	20,7	58	38,0
7	15	23,3	51	55,3
8	15	25,69	44	71,3
9	15	25,95	37	73,0
10	15	30	30	100,0
Rata-rata *Error* (%)				**38,6**

Deteksi batas jarak yang selanjutnya dilakukan pada gedung tengah. Tabel 4.5 menunjukkan deteksi batas jarak pada gedung tengah. Pada pengujian batas jarak gedung tengah didapatkan rata-rata *error* batas jarak sebesar 111,7%.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.5 Deteksi batas jarak gedung tengah pada pola gerakkan satu

Pengujian	Batas Jarak (cm)	Deteksi Batas Jarak (cm)	Kapasitas Baterai Setelah Dijalankan (%)	*Error* (%)
1	15	8,87	93	40,87
2	15	11,1	86	26,00
3	15	13,98	79	6,80
4	15	21,56	72	43,73
5	15	28,28	65	88,53
6	15	29,15	58	94,33
7	15	29,75	51	98,33
8	15	32,55	44	117,00
9	15	33,29	37	121,93
10	15	86,92	30	479,47
Rata-rata *Error* (%)				**111,7**

Gedung terakhir yang dilalui robot pada pola gerakan 1 adalah gedung kiri. Tabel 4.6 menunjukkan hasil deteksi batas jarak antara robot dengan gedung kiri. Dari pengambilan data pada gedung kiri didapatkan *error* batas jarak sebesar 126,14%.

Tabel 4.6 Deteksi Batas Jarak Gedung Kiri Pada Pola Gerakkan Satu

Pengujian	Batas Jarak (cm)	Deteksi Batas Jarak (cm)	Kapasitas Baterai Setelah Dijalankan (%)	*Error* (%)
1	15	5,18	93	65,47
2	15	5,24	86	65,07
3	15	6,53	79	56,47
4	15	9,13	72	39,13
5	15	15,3	65	2,00
6	15	17,92	58	19,47
7	15	25,2	51	68,00
8	15	34,3	44	128,67
9	15	44,22	37	194,80
10	15	108,35	30	622,33
Rata-rata *Error* (%)				**126,14**

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Titik terakhir yang harus dicapai robot adalah titik ELP pada kotak kuning di sebelah kiri arena. Jarak antara gedung kiri dan titik robot berhenti ditentukan dengan mengurangi lebar arena dengan deteksi jarak dari sensor ultrasonik. Pada Tabel 4.7 ditunjukkan jarak yang dicapai robot ketika berhenti. Pada pengujian ini didapatkan *error* sebesar 8,34%.

Tabel 4.7 Jarak Robot Berhenti pada Pola Gerakkan Satu

Pengujian	**Jarak Berhenti (cm)**	**Jarak Berhenti Sesungguhnya (cm)**	**Kapasitas Baterai Setelah Dijalankan (%)**	***Error* (%)**
1	258	274,92	93	6,6
2	258	265,12	86	2,8
3	258	252	79	2,3
4	258	250,32	72	3,0
5	258	241,12	65	6,5
6	258	240,15	58	6,9
7	258	237,9	51	7,8
8	258	227,8	44	11,7
9	258	218,82	37	15,2
10	258	204,8	30	20,6
Rata-rata *Error* (%)				**8,34**

Dari pengujian deteksi batas jarak antara robot dengan gedung didapatkan bahwa robot masih kesulitan untuk berhenti tepat di jarak 15cm. Hanya satu pengujian pada gedung kanan yang berhasil berhenti tepat 15cm di depan gedung kanan. Untuk pengujian batas jarak berhenti di depan gedung, *error* yang paling kecil terjadi di gedung kanan sementara *error* yang paling besar terjadi di gedung kiri. Adapun untuk pengujian jarak berhenti menghasilkan *error* yang paling kecil dibandingkan pengujian batas jarak di depan gedung.

Terjadinya *error* pada keseluruhan pola gerakan satu ini dikarenakan kemampuan pemrosesan Raspberry Pi yang terbatas jika digunakan untuk mengontrol kamera dan sensor ultrasonik. Selain itu karena keterbatasan pin PWM pada Raspberry Pi sehingga menggunakan pin GPIO lain yang bukan PWM sehingga menyebabkan keluaran PWM dari Raspberry Pi kurang stabil.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Keluaran PWM dari Raspberry Pi yang kurang stabil menyebabkan ketidakstabilan putaran motor pada robot. Akibat dari ketidakstabilan putaran motor adalah gerakan robot pada saat berbelok menjadi tidak selalu tepat 90 derajat. Hal ini menyebabkan posisi berhenti robot di depan gedung maupun di kotak ELP menjadi tidak presisi dan bahkan meleset jauh dari jarak yang ditargetkan, seperti yang terlihat pada Tabel 4.5 dan Tabel 4.6.

Selain pengaruh dari keluaran PWM, terjadinya *error* juga dipengaruhi oleh sumber tegangan untuk keempat motor dari baterai 11,1V 3s yang mengalami penurunan kapasitas sebesar 7% di setiap pengujian dan baterai tidak diisi ulang setiap kali selesai pengujian. Kapasitas baterai bernilai 100% sebelum dilakukan pengujian dan ketika akan memulai pengujian kesepuluh, kapasitas baterai bernilai 37%. Robot dijalankan selama 10 menit dan berdasarkan keluaran arus, kapasitas baterai akan habis dalam waktu 12 menit. Dengan demikian maka dapat dilihat hasil pengujian kesepuluh yang selalu berbeda jauh dengan pengujian pertama sehingga menghasilkan *error* yang cukup besar dengan pengaruh dari keluaran PWM serta kapasitas tegangan dan keluaran arus dari baterai.

4.3.2 Hasil Pembacaan Warna dan Kode QR Melalui Modul Kamera

Pembacaan warna pada modul kamera Raspberry Pi dilakukan dengan metode segmentasi HSV. Pada segmentasi HSV terdapat elemen *Hue* yang menunjukkan nilai warna yang terlihat. Nilai warna pada elemen *Hue* ditunjukkan pada rentang nilai 0 sampai 179. Adapun rentang nilai untuk elemen *Saturation* dan *Value* adalah 0 sampai 255. Pada pengujian ini nilai *Saturation* dan *Value* tidak diubah-ubah dan diposisikan pada nilai maksimal. Dari hasil pencarian nilai warna dari rentang nilai *Hue* maka didapat rentang nilai untuk warna kuning, hijau, biru yang ditunjukkan pada Tabel 4.8. Adapun rentang warna di luar warna-warna tersebut dinyatakan sebagai *undefined* atau tidak terdeteksi.

Tabel 4.8 Rentang Nilai Warna Kuning, Hijau, Biru

Warna	**Rentang Nilai**
Kuning	25 – 33
Hijau	33 – 68
Biru	68 - 120

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Rentang warna tersebut kemudian ditambahkan ke program pembacaaan warna sebagai syarat pembacaan warna kuning, hijau, dan biru. Gambar 4.21 menunjukkan potongan pemrograman dari proses pembacaan warna.

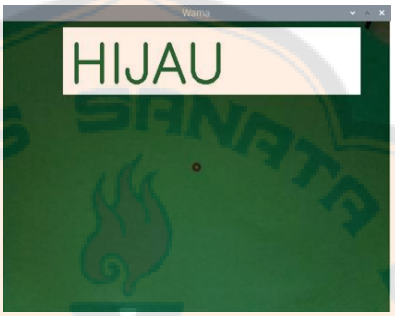
```
while True:
    _, frame = cap.read()
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    height, width, _ = frame.shape

    cx = int(width / 2)
    cy = int(height / 2)

    # Pick pixel value
    pixel_center = hsv_frame[cy, cx]
    #HSV = Hue, Saturation, Value
    hue_value = pixel_center[0]

    color = "UNDEFINED"
    if hue_value < 25:
        color = "UNDEFINED"
    elif hue_value < 33:
        color = "KUNING"
    elif hue_value < 68:
        color = "HIJAU"
    elif hue_value < 120:
        color = "BIRU"
    else:
        color = "UNDEFINED"

    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]), int(pixel_center_bgr[2])

    cv2.rectangle(frame, (cx - 220, 10), (cx + 270, 120), (255, 255, 255), -1)
    cv2.putText(frame, color, (cx - 210, 100), 0, 3, (b, g, r), 5)
    cv2.circle(frame, (cx, cy), 5, (25, 25, 25), 3)
```

Gambar 4.21 Program pembacaan warna

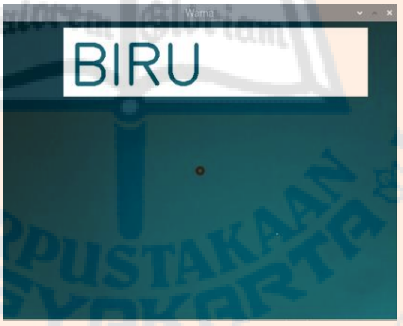
Pemrograman ini menggunakan *library* OpenCV yang mendukung dalam pemrosesan gambar pada Python. Selain itu digunakan juga *library* NumPy untuk pengolahan numerik dari gambar yang tertangkap. Layar yang dihasilkan dari pemrograman ini memiliki ukuran 640 x 480 yang memiliki titik lingkaran kecil di tengah dengan fungsi untuk mendeteksi warna. Jika titik tersebut mendeteksi warna kuning, hijau, atau biru maka akan tertampil tulisan nama warna di layar. Pada pengujian ini kamera berhasil membaca warna kuning, hijau, dan biru yang ditunjukkan pada Gambar 4.22, 4.23, dan 4.24. Ketika diberikan warna selain ketiga warna tersebut maka akan muncul tulisan "UNDEFINED" pada layar seperti pada pembacaan warna merah muda yang ditunjukkan pada Gambar 4.25.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.22 Hasil pembacaan warna kuning



Gambar 4.23 Hasil pembacaan warna hijau



Gambar 4.24 Hasil pembacaan warna biru

Gambar 4.25 Hasil pembacaan warna merah muda

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada pemrograman untuk membaca kode QR digunakan *library* Pyzbar pada pemrograman OpenCV Python. Dari *library* tersebut peneliti menggunakan fungsi *decode* untuk menerjemahkan kode QR dengan tipe pengkodean UTF-8 menjadi karakter yang bisa dibaca. Ukuran dari layar untuk tampilan hasil pembacaan kode QR adalah sama dengan layar pembacaan warna. Pada saat kode QR terdeteksi maka akan muncul kotak di sekitar kode QR dengan tulisan di atasnya. Tulisan itu menunjukkan hasil terjemahan kode QR. Untuk pemrograman kode QR dapat dilihat pada Gambar 4.26.

```
sucess, img = cap.read()
for barcode in decode(img) :
    myData = barcode.data.decode('utf-8')
    print(myData)
    pts = np.array([barcode.polygon],np.int32)
    pts = pts.reshape((-1,1,2))
    cv2.polylines(img,[pts],True,(255,0,255),5)
    pts2 = barcode.rect
    cv2.putText(img,myData,(pts2[0],pts2[1]),
                cv2.FONT_HERSHEY_SIMPLEX,0.9,
                (255,0,255),2)

cv2.imshow("Kode QR",img)
```

Gambar 4.26 Pemrograman pembacaan kode QR

Pengujian untuk pembacaan kode QR dilakukan dengan meletakkan ujung kamera robot pada jarak 100cm atau 1m dari gedung hijau yang di atasnya terdapat kode QR. Pada robot terdapat dudukan kamera yang memiliki sudut kemiringan 60 derajat menghadap ke atas. Pada Gambar 4.27 ditunjukkan indikator keberhasilan kamera dalam mendeteksi kode QR, yaitu jika pada layar muncul kotak dengan tulisan data kode QR.

Gambar 4.27 Indikator keberhasilan pembacaan kode QR

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

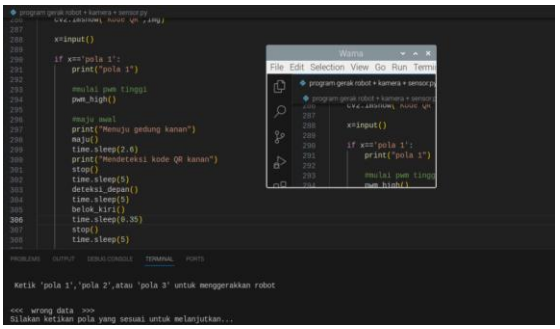
Hasil dari pengujian ini disajikan dalam Tabel 4.9 yang menunjukkan jarak pembacaan, keberhasilan pembacaan, dan data yang terbaca pada kode QR. Pengujian dilakukan dengan memajukan robot ke arah gedung dari jarak 100cm sampai 10cm di depan gedung. Data diambil setiap perpindahan jarak 10cm menuju gedung. Ketinggian dari setiap kode QR mengikuti ketinggian gedung.

Tabel 4.9 Keberhasilan Pembacaan Kode QR

No	Kode QR	Ketinggian Kode QR	Jarak (cm)										Persen Keberhasilan (%)
			100	90	80	70	60	50	40	30	20	10	
1	VTOL1	98 cm	N	N	N	Y	Y	Y	N	N	N	N	30
2	VTOL2	50 cm	Y	Y	Y	Y	Y	Y	N	N	N	N	60
3	VTOL3	101cm	N	N	N	Y	N	N	N	N	N	N	10

Kemampuan pembacaan kode QR pada tabel dinyatakan dengan huruf Y jika berhasil membaca dan N jika gagal membaca. Dari pengujian tersebut didapatkan bahwa pembacaan dengan akurasi tertinggi diperoleh dari pembacaan kode QR VTOL 2 dengan ketinggian 50cm dan rentang jarak pembacaan 50 – 70cm. Sedangkan untuk pembacaan kode QR VTOL 3 memiliki akurasi 10% dan hanya berhasil pada jarak 70cm. Adapun untuk kode QR VTOL 2 memiliki akurasi 30% dengan rentang jarak 50 – 70cm.

Pada pengujian ini fungsi kamera tidak bisa berjalan dengan baik jika digabungkan dengan pengontrolan motor dan pendeteksian sensor ultrasonik. Pada Gambar 4.28 ditunjukkan tampilan layar “Warna” pada kamera yang gagal menampilkan tangkapan gambar di depan kamera. Hal ini dikarenakan program pengolahan gambar pada kamera memerlukan proses data yang cukup berat pada Raspberry Pi sehingga untuk pemrograman kamera tidak bisa digabung dengan pemrograman motor dan sensor ultrasonik.



Gambar 4.28 Kamera gagal berfungsi

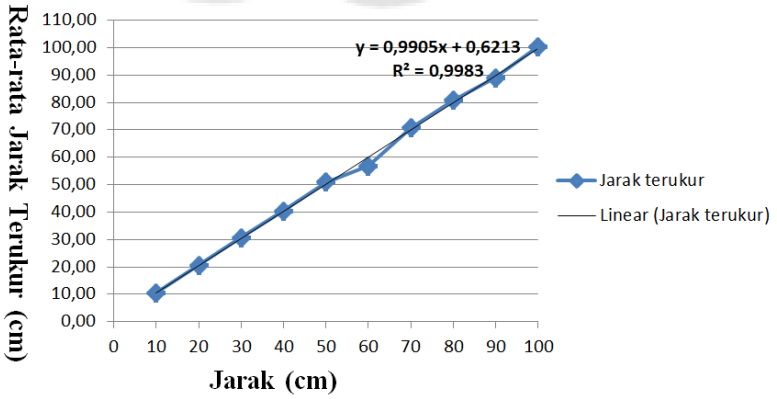
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

4.3.3 Hasil Pengukuran Jarak Melalui Sensor Ultrasonik

Tabel 4.10 Hasil Pengukuran Jarak Melalui Sensor Ultarsonik

Jarak (cm)	Hasil Pengujian Pengukuran Jarak										Rata - rata	Error (%)
	1	2	3	4	5	6	7	8	9	10		
10	10,0	10,60	10,6	10,6	10,6	10,6	10,7	10,7	10,6	10,6	10,56	5,30
20	20,8	20,70	20,7	20,7	20,8	20,7	20,4	20,4	20,8	20,6	20,66	3,19
30	30,7	30,60	30,6	30,6	30,6	30,7	30,6	30,6	30,6	30,6	30,62	2,02
40	40,5	40,50	40,5	40,4	40,5	40,4	40,4	40,0	40,5	40,1	40,38	0,94
50	51,0	50,90	51,0	51,0	51,1	51,0	51,0	51,0	51,1	50,9	51,00	1,96
60	56,9	56,90	56,9	56,9	56,9	56,9	56,9	56,9	56,9	56,9	56,90	5,45
70	70,7	70,70	70,8	70,6	70,8	70,8	70,7	70,7	70,8	70,8	70,74	1,05
80	80.8	80,70	81,0	80,8	80,8	80,8	80,9	80,8	80,8	80,7	80,81	1,00
90	88,4	88,90	89,0	88,9	89,4	89,0	89,3	88,9	89,0	88,9	88,97	1,16
100	100,5	100,10	100,6	100,6	100,3	100,2	100,2	100,1	100,5	100,6	100,37	0,37
Rata - Rata Error (%)												**2,24**

Pada Tabel 4.10 dapat dilihat hasil pengukuran jarak melalui modul sensor ultrasonik HC-SR04. Pengukuran dilakukan sebanyak sepuluh kali dengan jarak maksimal 1m atau sama dengan 100cm. Pengambilan data pengukuran dilakukan setiap perpindahan jarak 10cm dan diambil 10 data dari masing-masing perpindahahan jarak. Nilai *error* dicari melalui perbandingan antara nilai yang terukur dengan nilai yang diinginkan kemudian dicari rata-ratanya untuk *error* keseluruhan. Pada pengukuran ini peneliti mendapatkan rata-rata nilai *error* sebesar 2,24%. Gambar 4.29 merupakan grafik hasil pengukuran jarak yang menunjukkan bahwa rata-rata hasil pengukuran mendekati nilai jarak yang ditentukan.



Gambar 4.29 Grafik hasil pengukuran jarak

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pada Gambar 4.30 ditunjukkan pemosisian antara ujung sensor ultrasonik pada robot dengan gedung hijau pada jarak 100cm menggunakan meteran. Setelah sensor ultrasonik berada pada posisi yang diinginkan kemudian program dijalankan.

Gambar 4.30 Pengukuran jarak melalui sensor ultrasonik

```
while True:
        # Posisi awal trigger padaa keadaan LOW
    GPIO.output(Trigger_Tengah, GPIO.LOW)
    # Waktu untuk menyiapkan modul
    time.sleep(0.2)
    # Mengirim pulsa 10us pada trigger
    GPIO.output(Trigger_Tengah, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(Trigger_Tengah, GPIO.LOW)
    start = time.time()
    stop = time.time()
    while GPIO.input(Echo_Tengah) == 0:
        start = time.time() #lamanya echo tidak menangkap sinyal
    while GPIO.input(Echo_Tengah) == 1:
        stop = time.time() #lamanya echo menangkap sinyal
    # Menghitung waktu yang ditempuh pulsa
    waktu_tempuh = stop - start
    # Waktu tempuh dikalikan dengan kecepatan suara (34300cm/s) untuk mmendapatkan jarak
    # Jarak dibagi 2 karena ada pulsa start dan stop
    jarak = waktu_tempuh * 34300 / 2
    print ("Jarak Objek Depan : %.1f" % jarak)
```

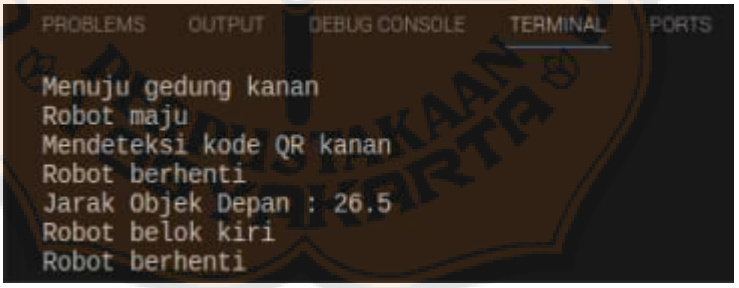
Gambar 4.31 Program sensor ultrasonik

Pemrograman pada sensor ultrasonik ini digunakan untuk mengukur jarak sesungguhnya yang ditangkap oleh sensor melalui tangkapan sinyal pantulan dari benda di depan robot. Gambar 4.31 menunjukkan pemrograman pada sensor ultrasonik yang diawali

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

dengan membuat keluaran pin Trigger pada posisi LOW agar pengukuran mampu dimulai dari nilai 0. Setelah itu pin Trigger dihidupkan atau diposisikan HIGH selama 10 mikrodetik kemudian keluaran pin Trigger turun ke posisi LOW. Pin Echo merupakan pin yang digunakan untuk memberikan data masukan ke Raspberry Pi ketika sensor ultrasonik berhasil mendeteksi sinyal pantulan. Ketika sinyal pantulan terdeteksi pin Echo bernilai 1, sedangkan ketika belum terdeteksi pin Echo bernilai 0. Dari pendeteksian pantulan tersebut peneliti mampu mendapatkan nilai waktu tempuh melalui selisih antara lamanya waktu pin Echo bernilai 0 dengan lamanya waktu ketika pin Echo bernilai 1. Waktu tempuh yang didapatkan kemudian dikalikan dengan kecepatan suara. Karena sensor ultrasonik mengirimkan sinyal dan menangkap sinyal pantulan maka hasil perkalian tersebut kemudian dibagi dua. Hasil dari persamaan pada program tersebut adalah jarak yang terukur dari objek di depan sensor ultrasonik.

Pada Gambar 4.32 ditunjukkan tampilan kolom terminal ketika sensor ultrasonik mendeteksi batas jarak antara robot dengan gedung di depan. Sensor ultrasonik mulai mendeteksi ketika robot berhenti di depan gedung dan sensor ultrasonik menghadap ke gedung. Batas jarak yang dideteksi sensor akan tertampil setelah kalimat “Jarak Objek Depan :” pada kolom terminal.

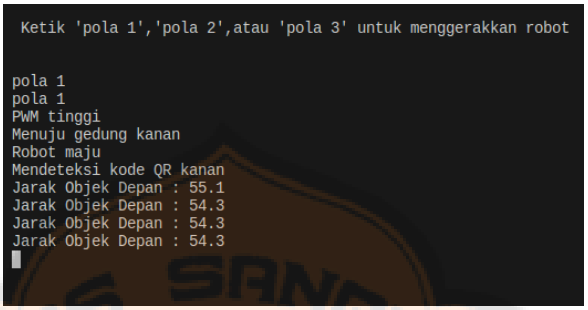


Gambar 4.32 Program sensor ultrasonik ketika robot berjalan

Sensor ultrasonik mengalami kegagalan dalam berfungsi ketika pemrograman sensor digabung dengan pemrograman untuk pembacaan dari kamera dan pengontrolan motor. Pada Gambar 4.33 terlihat terminal pada aplikasi pemrograman yang hanya menampilkan empat data hasil deteksi jarak. Data yang tertampil pada terminal belum sesuai dengan jarak yang diminta sehingga data yang tertampil seharusnya lebih dari empat sampai sensor berhasil mendeteksi jarak yang diminta. Dengan kegagalan integrasi sensor ultrasonik

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

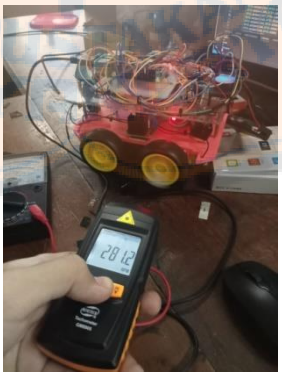
terhadap perangkat yang lain, maka sensor ultrasonik tidak jadi digunakan untuk memberi pengaruh terhadap kontrol *on-off* motor melainkan hanya diaktifkan ketika robot berhenti setelah menempuh jarak yang diminta. Motor pada robot akan aktif sesuai lama waktu yang diberikan tanpa pengaruh dari pembacaan sensor ultrasonik.



Gambar 4.33 Sensor ultrasonik gagal berfungsi

4.3.4 Pengujian Kecepatan Motor Berdasarkan Arah Gerak Robot

Pada bagian ini dilakukan pengujian terhadap kecepatan motor berdasarkan arah gerak robot. Gambar 4.34 menunjukkan proses pengujian yang dilakukan dengan membaca putaran roda pada setiap arah gerak robot menggunakan tachometer. Roda robot yang akan diuji ditempelkan selotip warna putih agar cahaya infra merah dari tachometer bisa memantul. Adapun robot pada pengujian ini diletakkan di atas kotak agar roda tidak menyentuh meja.



Gambar 4.34 Proses pengujian kecepatan

Pada Tabel 4.11 disajikan data hasil pembacaan tachometer yang menunjukkan kecepatan putaran pada roda motor dalam satuan rpm (*revolution per minute*) ketika robot tidak dijalankan pada arena. Motor yang digunakan pada proyek ini dibagi menjadi dua

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

berdasarkan peletakan motor, yaitu motor kanan dan motor kiri. Motor kanan diberi angka 1 dan 2 sementara motor kiri diberi angka 3 dan 4. Data kecepatan putaran diambil berdasarkan empat arah gerakan robot yaitu maju, mundur, belok kanan, dan belok kiri.

Tabel 4.11 Pengujian kecepatan putaran motor

Motor		**Kecepatan Putaran (rpm)**				**Rata-rata (rpm)**
		Maju	**Mundur**	**Belok Kanan**	**Belok Kiri**	
Kanan	**1**	231,7	222,3	232,2	233,2	229,85
	2	269,4	258,1	221	250,2	249,675
Rata-rata Motor Kanan (rpm)						**239,7625**
Kiri	**3**	254,1	246,8	270,6	235,4	251,725
	4	254,3	266,4	263,8	274,2	264,675
Rata-rata Motor Kiri (rpm)						**258,2**

Pada pengujian ini setiap motor diberi PWM dengan nilai *duty cycle* dan frekuensi yang sama sehingga kecepatan dari motor 1 sampai motor 4 akan bernilai sama. Akan tetapi hasil dari pengujian ini menunjukkan nilai kecepatan motor yang bervariasi. Dari variasi nilai tersebut kemudian ditentukan rata-rata dari motor kanan dan kiri sehingga dihasilkan rata-rata kecepatan motor kanan yang lebih kecil dari motor kiri. Perbedaan kecepatan ini diakibatkan oleh keluaran *duty cycle* pada pin PWM Raspberry Pi yang tidak stabil. Dengan kecepatan setiap motor yang berbeda maka robot tidak akan selalu berhasil berjalan lurus, tetapi akan sedikit berbelok ke kanan atau ke kiri. Hal demikian juga terjadi ketika robot berbelok ke kanan atau ke kiri tidak selalu tepat 90°.

Tabel 4.12 Pengujian tegangan dan *duty cycle*

Motor		**Tegangan Sumber Saat Motor Aktif (V)**	**Tegangan Keluaran *Driver* (V)**	***Duty Cycle* Saat Motor Aktif (%)**
Kanan	1	11,24	6,9	61,39
	2	10,82	7,2	66,54
Kiri	3	11,66	7,5	64,32
	4	11,46	7,5	65,45

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.12 menunjukkan keluaran *duty cycle* PWM yang tidak stabil melalui perbandingan antara tegangan sumber saat motor berputar dengan tegangan keluaran *driver*. Pada pengujian ini diberikan *duty cycle* 75% dengan frekuensi PWM 1000Hz. Dengan kapasitas baterai yang mampu mencapai 12v maka perkiraan keluaran tegangan dari *driver* adalah 9v. Akan tetapi karena kebutuhan arus yang cukup banyak pada empat motor, maka kapasitas baterai menjadi cepat menurun terutama saat keempat motor diaktifkan. Keluaran *duty cycle* pada saat motor aktif dapat ditentukan dengan menggunakan persamaan 4.1.

$$Duty\ Cycle = \frac{V \text{ motor aktif}}{V \text{ keluaran } driver} \times 100\% \quad (4.1)$$

Dari persamaan 4.1 didapatkan bahwa keluaran *duty cycle* PWM yang berbeda-beda pada setiap motor. Keluaran *duty cycle* yang paling besar ditunjukan pada motor 2 dengan nilai 66,54% sedangkan keluaran *duty cycle* terkecil ditunjukan pada motor 1 dengan nilai 61,39%. Perbedaan nilai *duty cycle* di setiap motor membuktikan bahwa keluaran PWM dari Raspberry Pi cenderung tidak stabil sehingga mempengaruhi kestabilan gerakan robot.

Tabel 4.13 Kecepatan sudut, linear, dan torsi pada motor

Motor	Kecepatan Sudut (rad/s)	Kecepatan Linear (m/s)	Torsi (kgf.cm)
Kanan	25,10	0,82	1,31
Kiri	27,02	0,88	1,41
Rata - rata	**26,06**	**0,85**	**1,36**

Tabel 4.13 menunjukkan hasil konversi rata-rata kecepatan putaran motor kanan dan kiri serta nilai torsi yang dihasilkan ketika robot tidak dijalankan. Kecepatan putaran motor dikonversi ke kecepatan sudut untuk menentukan kecepatan linear yang akan digunakan untuk menentukan lamanya robot berjalan ketika menuju gedung pertama, berpindah gedung, dan menuju ke titik berhenti. Konversi kecepatan putaran ke kecepatan sudut menggunakan persamaan 4.2 sementara untuk konversi kecepatan sudut ke kecepatan linear menggunakan persamaan 4.3.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$\omega = \frac{\text{RPM}}{60 \text{ detik}} \times 2 \times \pi \quad (4.2)$$

$$v = \frac{\omega \times r}{1000} \quad (4.3)$$

Keterangan :

ω = Kecepatan sudut (rad/s)

RPM = Kecepatan putaran (rpm)

v = Kecepatan linear (m/s)

r = Jari-jari roda (32, 5mm)

Torsi yang dihasilkan mampu ditentukan dengan membandingkan kecepatan putaran pada pengujian dan kecepatan putaran pada spesifikasi gearbox ketika tegangan sumber 7,2v. Hasil perbandingan tersebut kemudian dikalikan dengan torsi pada spesifikasi gearbox ketika tegangan sumber 7,2v. Persamaan untuk menentukan torsi dapat dilihat pada persamaan 4.4.

$$\tau = \frac{\text{RPM}}{\text{RPM spesifikasi}} \times \tau \text{ spesifikasi} \quad (4.4)$$

Keterangan :

τ = Torsi (kgf.cm)

RPM = Kecepatan putaran (rpm)

RPM spesifikasi = 330rpm

τ spesifikasi = 1,8kgf.cm

Setelah melakukan pengujian kecepatan linear, kecepatan putaran, dan torsi pada kondisi robot tidak dijalankan kemudian dilakukan pengujian dengan kondisi robot dijalankan menuju gedung kanan, gedung tengah, gedung kiri, dan titik berhenti (ELP) menggunakan pola gerakan satu. Pengujian kecepatan dan torsi pada ketiga gedung dilakukan dengan menentukan jarak 15cm sebagai batas jarak untuk berhenti di depan gedung dan jarak 248cm sebagai jarak dari titik start ke gedung. Waktu dari titik start menuju batas jarak ditentukan dengan membagi jarak yang harus ditempuh dengan kecepatan linear pada saat robot tidak dijalankan menggunakan persamaan 4.5 sehingga didapatkan waktu 2,58 detik.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

$$t = \frac{(s1-s2)}{v \times 100} \quad (4.5)$$

Keterangan :

t = Waktu dari start ke batas jarak (s)

s1 = Jarak dari titik start ke gedung (cm)

s2 = Batas jarak di depan gedung (cm)

v = Kecepatan linear saat robot tidak dijalankan (m/s)

Waktu dari titik start ke batas jarak yang sudah ditentukan kemudian dimasukan ke dalam pemrograman untuk menentukan lamanya robot berjalan menuju batas jarak. Robot dijalankan sebanyak sepuluh kali dengan menggunakan pola gerakan satu. Hasil pengujian kecepatan dan torsi untuk menuju setiap gedung dapat dilihat pada Tabel 4.14, Tabel 4.15, dan Tabel 4.16 serta pada titik berhenti yang ditunjukan pada Tabel 4.17. Dari sepuluh kali pengujian didapatkan bahwa kecepatan dan torsi yang dihasilkan mengalami penurunan sehingga menyebabkan robot tidak selalu tepat berhenti di jarak 15cm dari depan gedung.

Tabel 4.14 Kecepatan dan Torsi pada Gedung Kanan

Pengujian	Target Batas Jarak (cm)	Batas Jarak Sesungguhnya (cm)	Target Jarak Yang Ditempuh (cm)	Jarak Tertempuh (cm)	Waktu pada Program (s)	Kecepatan Linear (m/s)	Kecepatan Putaran (rpm)	Torsi (kgf.cm)
1	15	15,00	248	233,00	2,58	0,90	265,49	1,45
2	15	16,20	248	231,80	2,58	0,90	264,12	1,44
3	15	16,30	248	231,70	2,58	0,90	264,01	1,44
4	15	17,00	248	231,00	2,58	0,90	263,21	1,44
5	15	17,76	248	230,24	2,58	0,89	262,34	1,43
6	15	20,70	248	227,30	2,58	0,88	258,99	1,41
7	15	23,30	248	224,70	2,58	0,87	256,03	1,40
8	15	25,69	248	222,31	2,58	0,86	253,31	1,38
9	15	25,95	248	222,05	2,58	0,86	253,01	1,38
10	15	30,00	248	218,00	2,58	0,84	248,40	1,35
Persentase Penurunan (%)						**6,44**	**6,44**	**6,44**

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Tabel 4.15 Kecepatan dan Torsi pada Gedung Tengah

Pengujian	Target Batas Jarak (cm)	Batas Jarak Sesungguhnya (cm)	Target Jarak Yang Ditempuh (cm)	Jarak Tertempuh (cm)	Waktu pada Program (s)	Kecepatan Linear (m/s)	Kecepatan Putaran (rpm)	Torsi (kgf.cm)
1	15	8,87	248	239,13	2,58	0,93	272,47	1,49
2	15	11,10	248	236,90	2,58	0,92	269,93	1,47
3	15	13,98	248	234,02	2,58	0,91	266,65	1,45
4	15	21,56	248	226,44	2,58	0,88	258,01	1,41
5	15	28,28	248	219,72	2,58	0,85	250,36	1,37
6	15	29,15	248	218,85	2,58	0,85	249,36	1,36
7	15	29,75	248	218,25	2,58	0,85	248,68	1,36
8	15	32,55	248	215,45	2,58	0,84	245,49	1,34
9	15	33,29	248	214,71	2,58	0,83	244,65	1,33
10	15	86,92	248	161,08	2,58	0,62	183,54	1,00
Persentase Penurunan (%)						**32,64**	**32,64**	**32,64**

Tabel 4.16 Kecepatan dan Torsi pada Gedung Kiri

Pengujian	Target Batas Jarak (cm)	Batas Jarak Sesungguhnya (cm)	Target Jarak Yang Ditempuh (cm)	Jarak Tertempuh (cm)	Waktu pada Program (s)	Kecepatan Linear (m/s)	Kecepatan Putaran (rpm)	Torsi (kgf.cm)
1	15	5,18	248	242,82	2,58	0,94	276,68	1,51
2	15	5,24	248	242,76	2,58	0,94	276,61	1,51
3	15	6,53	248	241,47	2,58	0,94	275,14	1,50
4	15	9,13	248	238,87	2,58	0,93	272,18	1,48
5	15	15,3	248	232,70	2,58	0,90	265,15	1,45
6	15	17,92	248	230,08	2,58	0,89	262,16	1,43
7	15	25,2	248	222,80	2,58	0,86	253,87	1,38
8	15	34,3	248	213,70	2,58	0,83	243,50	1,33
9	15	44,22	248	203,78	2,58	0,79	232,19	1,27
10	15	108,35	248	139,65	2,58	0,54	159,12	0,87
Persentase Penurunan (%)						**42,49**	**42,49**	**42,49**

Tabel 4.17 Kecepatan dan torsi pada titik berhenti

Pengujian	Target Batas Jarak (cm)	Batas Jarak Sesungguhnya (cm)	Target Jarak Yang Ditempuh (cm)	Jarak Tertempuh (cm)	Waktu pada Program (s)	Kecepatan Linear (m/s)	Kecepatan Putaran (rpm)	Torsi (kgf.cm)
1	5,18	258	204,80	199,62	2,3	0,87	255,14	1,39
2	5,24	258	218,82	213,58	2,3	0,93	272,99	1,49
3	6,53	258	227,80	221,27	2,3	0,96	282,82	1,54
4	9,13	258	237,90	228,77	2,3	0,99	292,40	1,59
5	15,3	258	240,15	224,85	2,3	0,98	287,39	1,57
6	17,92	258	241,12	223,20	2,3	0,97	285,28	1,56
7	25,2	258	250,32	225,12	2,3	0,98	287,74	1,57
8	34,3	258	252,00	217,70	2,3	0,95	278,25	1,52
9	44,22	258	265,12	220,90	2,3	0,96	282,34	1,54
10	108,35	258	274,92	166,57	2,3	0,72	212,90	1,16
Persentase Penurunan (%)						**16,56**	**16,56**	**16,56**

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Dari keempat pengujian tersebut didapatkan persentase penurunan kecepatan dan torsi yang paling tinggi terjadi ketika robot menuju gedung kiri. Terlihat hasil deteksi batas jarak sesungguhnya pada pengujian 1 sampai 4 yang kurang dari target jarak 15cm sehingga posisi robot menjadi lebih dekat dengan gedung. Adapun pada pengujian 7 sampai 10 menampilkan hasil deteksi batas jarak sesungguhnya yang melebihi target jarak 15cm sehingga posisi robot menjadi lebih jauh dari target batas jarak yang ditentukan. Selain itu pada pengujian 10 dihasilkan deteksi batas jarak yang melebihi 100cm. Pada Tabel 4.17 terlihat hasil deteksi batas jarak di titik berhenti pada pengujian kesepuluh yang mencapai 16,92cm di atas batas jarak yang ditentukan (258cm).

Perbedaan persentase penurunan terlihat pada pengujian di gedung kanan, gedung tengah, dan titik berhenti. Pada gedung kanan terjadi persentase penurunan kecepatan dan torsi yang paling kecil. Jarak yang paling jauh dari target batas jarak pada gedung kanan adalah 30cm. Pada pengujian di gedung tengah hanya dua hasil pengujian yang mampu mendekati batas jarak 15cm sementara hasil pengujian yang lain memiliki selisih jarak yang cukup besar dengan target batas jarak 15cm. Batas jarak terjauh yang ditempuh pada pengujian di gedung tengah adalah 86,92cm. Sementara itu pada titik berhenti jarak tempuh robot yang mendekati 258cm hanya terjadi pada pengujian 5 sampai 9 dengan rentang jarak 10cm di atas atau di bawah jarak 258cm serta penurunan pada titik berhenti memiliki nilai 34,24%.

Penyebab dari penurunan kecepatan dan torsi di atas 10% pada pengujian tersebut disebabkan oleh kapasitas baterai yang sudah sangat sedikit ketika akan melakukan pengujian kesepuluh seperti yang terjadi pada pengujian pola gerakan satu dengan deteksi batas jarak.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian data terhadap alat yang sudah dilakukan maka dapat disimpulkan beberapa hal sebagai berikut.

1. Pada pengujian pola gerakan robot dengan pendeteksian batas jarak terjadi *error* yang tinggi di setiap pengujian kesepuluh dan juga ketika robot menuju gedung kiri yang disebabkan penurunan kapasitas baterai sebesar 7% di setiap pengujian sehingga kapasitas baterai tersisa 37% ketika akan memulai pengujian kesepuluh.
2. Kamera mampu mendeteksi warna kuning, hijau, dan biru serta mampu mendeteksi kode QR dengan keberhasilan tertinggi pada kode QR VTOL2 di ketinggian 50cm dengan nilai keberhasilan 60% dan keberhasilan terendah pada kode QR VTOL3 di ketinggian 101cm dengan nilai keberhasilan 10%.
3. Sensor ultrasonik mampu mengukur jarak dengan baik dan menghasilkan rata-rata *error* yang kecil, yaitu 2,24%.
4. Kecepatan dan torsi robot mengalami penurunan paling tinggi ketika menuju gedung kiri dengan nilai penurunan 42,49%.
5. Robot masih kesulitan dalam memastikan titik berhenti dan batas jarak pada gedung ketika menjalankan pola gerakan karena kamera dan sensor ultrasonik gagal diintegrasikan untuk pengontrolan motor.

5.2 Saran

Berdasarkan pengujian yang telah dilakukan pada penelitian ini, maka saran untuk pengembangan penelitian selanjutnya antara lain :

1. Menambahkan perangkat mikrokontroler untuk mengontrol aktuator dan modul sensor pada robot sehingga kerja Raspberry Pi tidak terlalu berat.
2. Mengembangkan kemampuan kamera untuk mendeteksi jarak atau menambahkan LIDAR supaya lebih mudah dalam mengontrol posisi robot.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR PUSTAKA

[1] Pedoman Kontes Robot Terbang Indonesia Tahun 2021

[2] Basuki,Winarno & Meliyana, Wahyu. (2016, Oktober). Pengendalian Gerak Robot Penghindar Halangan Menggunakan Citra dengan Kontrol PID. *Journal of Electrical Electronic Control and Automotive Engineering (JEECAE)*, *1*(1), 25-30.

[3] Fathi, Z. M., Fahmi, A., & Sunarya, U. (2016). Robot Pengenal dan Pencari Objek dengan Kamera Menggunakan Transformasi Hough. (SEMINAR NASIONAL INOVASI DAN APLIKASI TEKNOLOGI DI INDUSTRI (SENIATI) 2016, Institut Teknologi Nasional Malang) Diakses dari https://ejournal.itn.ac.id/index.php/seniati/article/download/938/837

[4] Setyanto, K. D., Fibriani, I., & Sumardi. PENGENDALIAN MOBILE ROBOT VISION MENGGUNAKAN WEBCAM PADA OBJEK ARAH PANAH BERBASIS RASPBERRY PI. *Jurnal Arus Elektro Indonesia (JAEI)*, 27-32

[5] Pashori, Aswar & Iswadi. (2014, Desember). Teknologi Robot. *Jurnal Fisika*, *1*, 82-86.

[6] Jatmiko, W., dkk. (2012). *Robotika: Teori dan Aplikasi.* Depok: Fakultas Ilmu Komputer Universitas Indonesia. Hal: 30 – 34.

[7] *Pi Setup. The MagPi Magazine. Issue 02 – June 2012*. Hal. 4 – 5

[8] *Raspberry Pi 4 Model B Product Brief*
https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf

[9] *Raspberry Pi 4 Model B Product Datasheet*
https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf

[10] *Raspberry Pi Documentation*
https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

[11] Triono, J. & Windiastik, S. P. (2022, April). Penerapan Raspberry Pi untuk Sistem Informasi Uang Elektronik (E-Money) Siswa Dengan Biaya Murah. *Journal of Computer, Information System, & Technology Management*, *5*(1), 23-34.

[12] Kadarina, T.M. & Hajar, M. H. I. (2019, September). Pengenalan Bahasa Pemrograman Python Menggunakan Aplikasi Games Untuk Siswa/i di Wilayah Kembangan Utara. *Jurnal Abdi Masyarakat (JAM)*, *5*(1), 11 - 16.

[13] Bagia, I. N. & Parsa, I. M. (2018). *Motor-Motor Listrik.* Kupang: CV. Rasi Terbit. Hal: 3 – 4.

[14] Muhammad. (2018). *Analisis Driver H-Bridge Menggunakan Relay Pada Motor BLDC Konstruksi Axial Flux (Celah Ganda)*. (Skripsi, Universitas Jember, 2018) Diakses dari https://repository.unej.ac.id/handle/123456789/89217

[15] Kurniawan, F. F., dkk. (2022, September). Rancang Bangun Pengatur Kecepatan Motor DC Dengan PWM Berbasis Arduino Nano. *JUPITER (Jurnal Pendidikan Teknik Elektro)*, *7*(2), 9 - 16.

[16] Amin, M., Ananda, R., & Eska, J. (2019, Desember). Analisis Penggunaan Driver Mini Victor L298N Terhadap Mobil Robot dengan Dua Perintah Android dan Arduino Nano. *Jurnal Teknologi dan Sistem Informasi (JURTEKSI)*, *6*(1), 53.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

[17] *Handson Technology L298N Dual H-Bridge Motor Driver Datasheet* http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

[18] Budiarso, Z., Nurraharjo, E., & Listiyono, H. (2019,Januari). RANCANG BANGUN SISTEM PENGATURAN KECEPATAN MOTOR ARUS SEARAH MENGGUNAKAN METODE DUAL H-BRIDGE BERBASIS ARDUINO. *Jurnal DINAMIK*, 24(1), 31-40

[19] Kusumayudha, S., Fitriyah, H., & Maulana, R. (2022, Juni). Pendaratan Micro-Drone berdasarkan Quick Response Code menggunakan DJI Tello dan Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(6), 2960-2966.

[20] Kusmanto, R. D., Tompunu, A. N., & Pambudi, W. S. (2011, September). Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV. *JURNAL ILMIAH ELITE ELEKTRO*, 2(2), 83-87.

[21] Al Azhar, G., Winarno, T., & Izza, S. (2022, Januari). Implementasi g-h Filter Pada SensorKompas Sebagai Peningkatan Akurasi TrajectoryTrackingRobot Differential Drive. *Journal of Mechanical and Electrical Technology*, 1(1).

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LAMPIRAN

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Pemrograman Robot

program gerak robot + kamera + sensor.py > ...

```python
#warna,QR,PWM,motor,ultrasonik
#semua pola gerakan

#setup camera
import cv2
import numpy as np
from pyzbar.pyzbar import decode

#setup motor
import RPi.GPIO as GPIO
import time

#GPIO setup
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

#Motor 1 kanan
in1 = 37
in2 = 31
en1 = 35
#Motor 2 kanan
in3 = 29
in4 = 15
en2 = 33
#Motor 3 kiri
in5 = 38
in6 = 40
en3 = 32
#Motor 4 kiri
in7 = 24
in8 = 26
in7 = 24
in8 = 26
en4 = 12

#Setup pin driver motor
#Motor 1 kanan
GPIO.setup(in1, GPIO.OUT)#forward motor1
GPIO.setup(in2, GPIO.OUT)#reverse motor1
GPIO.setup(en1, GPIO.OUT)#ENA motor1
#Motor 2 kanan
GPIO.setup(in3, GPIO.OUT)#forward motor2
GPIO.setup(in4, GPIO.OUT)#reverse motor2
GPIO.setup(en2, GPIO.OUT)#ENB motor2
#Motor 3 kiri
GPIO.setup(in5, GPIO.OUT)#forward motor3
GPIO.setup(in6, GPIO.OUT)#reverse motor3
GPIO.setup(en3, GPIO.OUT)#ENA motor3
#Motor 4 kiri
GPIO.setup(in7, GPIO.OUT)#forward motor4
GPIO.setup(in8, GPIO.OUT)#reverse motor4
GPIO.setup(en4, GPIO.OUT)#ENB motor4

#Set semua pin driver motor ke LOW
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
GPIO.output(in3,GPIO.LOW)
GPIO.output(in4,GPIO.LOW)
GPIO.output(in5,GPIO.LOW)
GPIO.output(in6,GPIO.LOW)
GPIO.output(in7,GPIO.LOW)
GPIO.output(in8,GPIO.LOW)
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
#Setup pin PWM
#frekuensi PWM = 1000Hz, periode = 1ms
p1 = GPIO.PWM(en1,1000)#PWM MOTOR 1
p2 = GPIO.PWM(en2,1000)#PWM MOTOR 2
p3 = GPIO.PWM(en3,1000)#PWM MOTOR 3
p4 = GPIO.PWM(en4,1000)#PWM MOTOR 4
#Persen PWM awal
p1.start(0)
p2.start(0)
p3.start(0)
p4.start(0)

#GPIO ultrasonik tengah
Trigger_Tengah = 16
Echo_Tengah = 18
GPIO.setup(Trigger_Tengah, GPIO.OUT)  # Trigger > Out
GPIO.setup(Echo_Tengah, GPIO.IN)      # Echo < In

print("Menjalankan Misi")

#Persen PWM
def pwm_low():
    #PWM rendah
    print("PWM rendah")
    p1.ChangeDutyCycle(25)
    p2.ChangeDutyCycle(25)
    p3.ChangeDutyCycle(25)
    p4.ChangeDutyCycle(25)

#Persen PWM
def pwm_low():
    #PWM rendah
    print("PWM rendah")
    p1.ChangeDutyCycle(25)
    p2.ChangeDutyCycle(25)
    p3.ChangeDutyCycle(25)
    p4.ChangeDutyCycle(25)

def pwm_medium():
    #PWM sedang
    print("PWM sedang")
    p1.ChangeDutyCycle(50)
    p2.ChangeDutyCycle(50)
    p3.ChangeDutyCycle(50)
    p4.ChangeDutyCycle(50)

def pwm_high():
    #PWM tinggi
    print("PWM tinggi")
    p1.ChangeDutyCycle(75)
    p2.ChangeDutyCycle(75)
    p3.ChangeDutyCycle(75)
    p4.ChangeDutyCycle(75)
def pwm_stop():
    #PWM berhenti
    print("PWM berhenti")
    p1.ChangeDutyCycle(0)
    p2.ChangeDutyCycle(0)
    p3.ChangeDutyCycle(0)
    p4.ChangeDutyCycle(0)

#Gerakan motor robot
def maju():
    print("Robot maju")
    #Semua reverse motor LOW
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in4, GPIO.LOW)
    GPIO.output(in6, GPIO.LOW)
    GPIO.output(in8, GPIO.LOW)
    #Semua motor bergerak maju
    GPIO.output(in1, GPIO.HIGH)
    GPIO.output(in3, GPIO.HIGH)
    GPIO.output(in5, GPIO.HIGH)
    GPIO.output(in7, GPIO.HIGH)
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
129  def belok_kiri():
130      print("Robot belok kiri")
131      GPIO.output(in2, GPIO.LOW)#reverse motor 1 LOW
132      GPIO.output(in4, GPIO.LOW)#reverse motor 2 LOW
133      GPIO.output(in5, GPIO.LOW)#forward motor 3 LOW
134      GPIO.output(in7, GPIO.LOW)#forward motor 4 LOW
135      #Motor kanan maju, motor kiri mundur
136      GPIO.output(in1, GPIO.HIGH)#forward motor 1 HIGH
137      GPIO.output(in3, GPIO.HIGH)#forward motor 2 HIGH
138      GPIO.output(in6, GPIO.HIGH)#reverse motor 3 HIGH
139      GPIO.output(in8, GPIO.HIGH)#reverse motor 4 HIGH
140
141  def belok_kanan():
142      print("Robot belok kanan")
143      GPIO.output(in6, GPIO.LOW)#reverse motor 3 LOW
144      GPIO.output(in8, GPIO.LOW)#reverse motor 4 LOW
145      GPIO.output(in1, GPIO.LOW)#forward motor 1 LOW
146      GPIO.output(in3, GPIO.LOW)#forward motor 2 LOW
147      #Motor kiri maju, motor kanan mundur
148      GPIO.output(in5, GPIO.HIGH)#forward motor 3 HIGH
149      GPIO.output(in7, GPIO.HIGH)#forward motor 4 HIGH
150      GPIO.output(in2, GPIO.HIGH)#reverse motor 1 HIGH
151      GPIO.output(in4, GPIO.HIGH)#reverse motor 2 HIGH
```

```
def mundur():
    print("Robot mundur")
    #Semua forward motor LOW
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in5, GPIO.LOW)
    GPIO.output(in7, GPIO.LOW)
    #Semua reverse motor HIGH
    GPIO.output(in2, GPIO.HIGH)
    GPIO.output(in4, GPIO.HIGH)
    GPIO.output(in6, GPIO.HIGH)
    GPIO.output(in8, GPIO.HIGH)

def stop():
    print("Robot berhenti")
    #Semua forward motor LOW
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in5, GPIO.LOW)
    GPIO.output(in7, GPIO.LOW)
    #Semua reverse motor LOW
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(in4, GPIO.LOW)
    GPIO.output(in6, GPIO.LOW)
    GPIO.output(in8, GPIO.LOW)
```

```
179  #Deteksi ultrasonik
180  def deteksi_depan():
181      # Posisi awal trigger padaa keadaan LOW
182      GPIO.output(Trigger_Tengah, GPIO.LOW)
183      # Waktu untuk menyiapkan modul
184      time.sleep(0.2)
185      # Mengirim pulsa 10us pada trigger
186      GPIO.output(Trigger_Tengah, GPIO.HIGH)
187      time.sleep(0.00001)
188      GPIO.output(Trigger_Tengah, GPIO.LOW)
189      start = time.time()
190      stop = time.time()
191      while GPIO.input(Echo_Tengah) == 0:
192          start = time.time() #lamanya echo tidak menangkap sinyal
193      while GPIO.input(Echo_Tengah) == 1:
194          stop = time.time() #lamanya echo menangkap sinyal
195      # Menghitung waktu yang ditempuh pulsa
196      waktu_tempuh = stop - start
197      # Waktu tempuh dikalikan dengan kecepatan cahaya (34000cm/s) untuk mmendapatkan jarak
198      # Jarak dibagi 2 karena ada pulsa start dan stop
199      jarak = waktu_tempuh * 34000 / 2
200      print ("Jarak Objek Depan : %.1f" % jarak)
201      return jarak
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```python
#Deteksi ultrasonik
def deteksi_depan():
    # Posisi awal trigger padaa keadaan LOW
    GPIO.output(Trigger_Tengah, GPIO.LOW)
    # Waktu untuk menyiapkan modul
    time.sleep(0.2)
    # Mengirim pulsa 10us pada trigger
    GPIO.output(Trigger_Tengah, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(Trigger_Tengah, GPIO.LOW)
    start = time.time()
    stop = time.time()
    while GPIO.input(Echo_Tengah) == 0:
        start = time.time() #lamanya echo tidak menangkap sinyal
    while GPIO.input(Echo_Tengah) == 1:
        stop = time.time() #lamanya echo menangkap sinyal
    # Menghitung waktu yang ditempuh pulsa
    waktu_tempuh = stop - start
    # Waktu tempuh dikalikan dengan kecepatan cahaya (34000cm/s) untuk mmendapatkan jarak
    # Jarak dibagi 2 karena ada pulsa start dan stop
    jarak = waktu_tempuh * 34000 / 2
    print ("Jarak Objek Depan : %.1f" % jarak)
    return jarak
```

```python
def cleargpios():
    print ("Membersihkan GPIO")
    GPIO.output(in1, GPIO.LOW)
    GPIO.output(in2, GPIO.LOW)
    GPIO.output(en1, GPIO.LOW)
    GPIO.output(in3, GPIO.LOW)
    GPIO.output(in4, GPIO.LOW)
    GPIO.output(en2, GPIO.LOW)
    GPIO.output(in5, GPIO.LOW)
    GPIO.output(in6, GPIO.LOW)
    GPIO.output(en3, GPIO.LOW)
    GPIO.output(in7, GPIO.LOW)
    GPIO.output(in8, GPIO.LOW)
    GPIO.output(en4, GPIO.LOW)
    GPIO.output(Trigger_Tengah, GPIO.LOW)
    print ("Semua GPIO bersih")

print("\n")
print(" Ketik 'pola 1','pola 2',atau 'pola 3' untuk menggerakkan robot")
print("\n")

#camera setting
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
while True:

    #baca warna
    _, frame = cap.read()
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    height, width, _ = frame.shape

    cx = int(width / 2)
    cy = int(height / 2)

    # Pick pixel value
    pixel_center = hsv_frame[cy, cx]
    #HSV = Hue, Saturation, Value
    hue_value = pixel_center[0]

    #warna yang dibutuhkan : hijau, biru, kuning
    color = "Undefined"
    if hue_value < 25:
        color = "UNDEFINED"
    elif hue_value < 33:
        color = "KUNING"
    elif hue_value < 68:
        color = "HIJAU"
    elif hue_value < 120:
        color = "BIRU"
    else:
        color = "UNDEFINED"

    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]), int(pixel_center_bgr[2])
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
    cv2.imshow("Warna", frame)

    sucess, img = cap.read()
    for barcode in decode(img) :
        myData = barcode.data.decode('utf-8')
        #print(myData)
        pts = np.array([barcode.polygon],np.int32)
        pts = pts.reshape((-1,1,2))
        cv2.polylines(img,[pts],True,(255,0,255),5)
        pts2 = barcode.rect
        cv2.putText(img,myData,(pts2[0],pts2[1]),
                    cv2.FONT_HERSHEY_SIMPLEX,0.9,
                    (255,0,255),2)
        #baca gedung kanan
        if myData == 'VTOL 1' :
            print(myData)
        if myData == 'VTOL 2' :
            print(myData)
        if myData == 'VTOL 3' :
            print(myData)

    cv2.imshow("Kode QR",img)
    x=input()

    if x=='pola 1':
        print("pola 1")

        #mulai pwm tinggi
        pwm_high()

        #maju awal
        print("Menuju gedung kanan")
        maju()
        time.sleep(2.6)
        print("Mendeteksi kode QR kanan")
        stop()
        time.sleep(5)
        belok_kiri()
        time.sleep(0.35)
        stop()
        time.sleep(5)

        #ke gedung tengah
        print("Menuju gedung tengah")
        maju()
        time.sleep(1.5)
        stop()
        time.sleep(5)
        belok_kanan()
        time.sleep(0.35)
        print("Mendeteksi kode QR kanan")
        stop()
        time.sleep(5)
        belok_kiri()
        time.sleep(0.35)
        stop()
        time.sleep(5)

        #ke gedung kiri
        print("Menuju gedung kiri")
        maju()
        time.sleep(1.5)
        stop()
        time.sleep(5)
        belok_kanan()
        time.sleep(0.35)
        print("Mendeteksi kode QR kanan")
        stop()
        time.sleep(5)
        belok_kiri()
        time.sleep(0.6)
        stop()
        time.sleep(5)

        #ke titik berhenti
        print("Menuju titik berhenti")
        maju()
        time.sleep(2.3)
        stop()
        pwm_stop()
        print("Titik berhenti tercapai")
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
elif x=='pola 2':
    print("pola 2")

    #mulai pwm tinggi
    pwm_high()

    #ke gedung tengah
    print("Menuju gedung tengah")
    belok_kiri()
    time.sleep(0.35)
    stop()
    time.sleep(5)
    maju()
    time.sleep(1.5)
    stop()
    time.sleep(5)
    belok_kanan()
    time.sleep(0.35)
    stop()
    time.sleep(5)
    maju()
    time.sleep(2.6)
    stop()
    belok_kanan()
    time.sleep(0.35)
    stop()
    time.sleep(5)
    maju()
    time.sleep(2.6)
    print("Mendeteksi kode QR kanan")
    stop()
    time.sleep(5)

    #ke gedung kanan
    print("Menuju gedung kanan")
    belok_kanan()
    time.sleep(0.35)
    stop()
    time.sleep(5)
    maju()
    time.sleep(1.5)
    stop()
    time.sleep(5)
    belok_kiri()
    time.sleep(0.35)
    print("Mendeteksi kode QR kanan")
    stop()
    time.sleep(5)

    #ke gedung kiri
    print("Menuju gedung kiri")
    belok_kiri()
    time.sleep(0.35)
    stop()
    time.sleep(5)
    maju()
    time.sleep(2.6)
    stop()
    time.sleep(5)
    belok_kanan()
```

Select Indentation

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
        time.sleep(0.35)
        print("Mendeteksi gedung kiri")
        stop()
        time.sleep(5)

        #ke titik berhenti
        print("Menuju titik berhenti")
        belok_kiri()
        time.sleep(0.6)
        stop()
        time.sleep(5)
        maju()
        time.sleep(2.6)
        stop()
        pwm_stop()
        print("Titik berhenti tercapai")

    elif x=='pola 3':
            print("pola 3")

            #PWM tinggi
            pwm_high()

            #ke gedung kiri
            print("Menuju gedung kiri")
            belok_kiri()
            time.sleep(0.35)
            stop()
            time.sleep(5)
            maju()
            time.sleep(2.6)
            stop()
            time.sleep(5)
            belok_kanan()
            time.sleep(0.35)
            stop()
            time.sleep(5)
            maju()
            time.sleep(2.6)
            print("Mendeteksi kode QR kiri")
            stop()
            time.sleep(5)

            #ke gedung tengah
            print("Menuju gedung tengah")
            belok_kanan()
            time.sleep(0.35)
            maju()
            time.sleep(1.5)
            stop()
            time.sleep(5)
            belok_kiri()
            time.sleep(0.35)
            print("Mendeteksi gedung tengah")
            stop()
            time.sleep(5)

            #ke gedung kanan
            print("Menuju gedung kanan")
            belok_kanan()
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI


```
            time.sleep(0.35)
            stop()
            time.sleep(5)
            maju()
            time.sleep(1.5)
            stop()
            time.sleep(5)
            belok_kiri()
            time.sleep(0.35)
            print("Mendeteksi kode QR kanan")
            stop()
            time.sleep(5)

            #ke titik berhenti
            print("Menuju titik berhenti")
            belok_kiri()
            time.sleep(0.35)
            maju()
            time.sleep(2.6)
            stop()
            time.sleep(5)
            belok_kiri()
            time.sleep(0.35)
            stop()
            time.sleep(5)
            maju()
            time.sleep(2.6)
            stop()
            pwm_stop()
            print("Titik berhenti tercapai")
    elif x=='e':
        print("Misi Selesai")
        stop()
        cleargpios()
        break

    else:
        print("<<<  wrong data  >>>")
        print("Silakan ketikan pola yang sesuai untuk melanjutkan...")

cap.release()
cv2.destroyAllWindows()
```