



POLITEKNIK NEGERI
CILACAP

TUGAS AKHIR

***ROBOT TROLI BERGERAK MENGGUNAKAN HAND
GESTURE RECOGNITION BERBASIS IMAGE
PROCESSING***

***MOBILE ROBOT TROLLEY USING HAND GESTURE
RECOGNITION BASED IMAGE PROCESSING***

Oleh :

YUDI ARRASYID
NIM.20.01.01.022

DOSEN PEMBIMBING :

ARIF SUMARDIONO, S.Pd., M.T.
NIP. 198912122019031014

ERNA ALIMUDIN, S.T., M.Eng.
NIP. 199008292019032013

**PROGRAM STUDI DIPLOMA III TEKNIK ELEKTRONIKA
JURUSAN REKAYASA ELEKTRO DAN MEKATRONIKA
POLITEKNIK NEGERI CILACAP
2023**



POLITEKNIK NEGERI
CILACAP

TUGAS AKHIR

**ROBOT TROLI BERGERAK MENGGUNAKAN
*HAND GESTURE RECOGNITION BERBASIS IMAGE
PROCESSING***

***MOBILE ROBOT TROLLEY USING HAND
GESTURE RECOGNITION BASED IMAGE
PROCESSING***

Oleh :

YUDI ARRASYID
NIM.20.01.01.022

DOSEN PEMBIMBING :

ARIF SUMARDIONO, S.Pd., M.T
NIP. 198912122019031014

ERNA ALIMUDIN, S.T., M.Eng
NIP. 199008292019032013

**PROGRAM STUDI DIPLOMA III TEKNIK ELEKTRONIKA
JURUSAN REKAYASA ELEKTRO DAN MEKATRONIKA
POLITEKNIK NEGERI CILACAP
2023**

HALAMAN PENGESAHAN
ROBOT TROLI BERGERAK MENGGUNAKAN HAND
GESTURE RECOGNITION BERBASIS IMAGE PROCESSING

Oleh :

YUDI ARRASYID
NIM.20.01.01.022

Tugas Akhir ini Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Ahli Madya (A.Md)
di
Politeknik Negeri Cilacap

Disetujui Oleh

Pengaji Tugas Akhir

Pembimbing Tugas Akhir

1. Galih Mustiko Aji, S.T., M.T.
NIP.19850917201931005

1. Arif Sumardiono, S.Pd., M.T.
NIP.198912122019031014

2. Hera Susanti, S.T., M.Eng.
NIP. 198604092019032011

2. Erna Alimudin, S.T., M.Eng.
NIP.199008292019032013



**LEMBAR PERNYATAAN PERSETUJUAN
PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Politeknik Negeri Cilacap, yang bertanda tangan dibawah ini, saya :

Nama : Yudi Arrasyid
NIM : 20.01.01.022

Demi mendorong kemajuan ilmu pengetahuan, saya sepakat memberikan Hak Bebas Royalti Non-Eksklusif (**Non-Exclusive Royalty-Free Right**) kepada Politeknik Negeri Cilacap atas karya ilmiah saya yang berjudul:

**"ROBOT TROLI BERGERAK MENGGUNAKAN HAND
GESTURE RECOGNITION BERBASIS IMAGE PROCESSING"**

Politeknik Negeri Cilacap dapat menggunakan Hak Bebas Royalti Non-Eksklusif ini untuk menyimpan, mengonversi format, mengelola di dalam database, mendistribusikan, serta menampilkan dan mempublikasikan karya ini secara online atau melalui media lain untuk keperluan akademis, tanpa perlu mendapatkan izin terlebih dahulu dari saya. Namun, sangat penting bagi mereka untuk selalu mencantumkan nama saya sebagai penulis atau pencipta.

Saya dengan pribadi bertanggung jawab sepenuhnya atas segala konsekuensi hukum yang mungkin terjadi sebagai hasil dari pelanggaran hak cipta dalam karya ilmiah ini, tanpa melibatkan pihak Politeknik Negeri Cilacap. Pernyataan ini saya sampaikan dengan tulus dan sungguh-sungguh..

Dibuat di : Cilacap
Pada Tanggal : 20 Juli 2023

Yang Menyatakan,

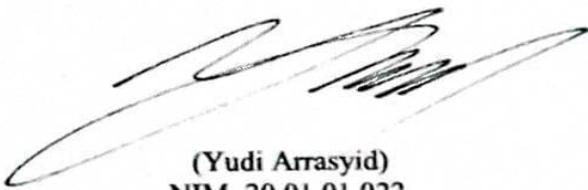


(Yudi Arrasyid)
NIM. 20.01.01.022

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan sungguh-sungguh dan tulus, saya menyatakan bahwa Laporan Tugas Akhir ini telah dirangkai berdasarkan penelitian, ide-ide, dan presentasi asli penulis mengenai perangkat keras, program, dan teks laporan yang dilampirkan dalam Laporan Akhir ini. Jika ada penggunaan karya orang lain, penulis akan dengan jelas mencantumkan sumbernya. Pernyataan ini saya sampaikan dengan sepenuh hati, dan jika di masa depan terdapat kesalahan atau ketidakbenaran dalam pernyataan ini, saya bersedia menerima konsekuensi akademik berupa pencabutan gelar yang telah saya peroleh melalui laporan ini, serta sanksi lain yang sesuai dengan standar yang berlaku di Perguruan Tinggi.

Cilacap, 20 Juli 2023
Yang menyatakan,



(Yudi Arrasyid)
NIM. 20.01.01.022

ABSTRAK

Troli supermarket adalah alat yang digunakan untuk membawa barang belanjaan saat berbelanja di supermarket atau toko serba ada. Troli ini umumnya terbuat dari logam atau plastik dan dilengkapi dengan roda sehingga mudah digerakkan di sekitar toko. Penggunaan troli sangat berguna untuk mengangkut barang dalam jumlah besar. Tetapi semakin banyak barang yang dibeli, semakin sulit mendorong atau menarik troli. Untuk mengatasi hal ini, troli otomatis dikembangkan agar dapat mengikuti gesture tangan pengguna, mengurangi kebutuhan untuk mendorong secara manual. Sistem ini menggunakan kamera yang dapat mendeteksi gesture tangan menggunakan python dan teknik pembelajaran mesin melalui mediapipe. Troli ini berhasil mengikuti perintah tangan dengan baik dalam berbagai kondisi seperti bergerak maju, kiri, kanan, mundur, dan berhenti dengan tingkat akurasi 98%. Kegagalan sebesar 2% terjadi akibat pencahayaan yang kurang baik dalam pengujian. Kecepatan rata-rata troli adalah 17,48 cm/s tanpa beban dan 7,47 cm/s dengan beban maksimal 50 kg. Troli tidak dapat melebihi batas maksimal tersebut. Baterai troli dengan batas discharge 10,5V memiliki masa pakai sekitar 50 menit tanpa beban dan 47 menit dengan beban 50 kg. Jarak maksimum antara troli dan objek adalah 650 cm. Jarak ideal untuk mendeteksi gesture tangan menggunakan kamera adalah antara 50 hingga 150 cm, dengan intensitas cahaya ideal 10 lux.

Kata kunci : robot troli, mikrokontroller, pengolah citra, isyarat tangan, mikrokomputer.

ABSTRACT

Supermarket trolley is a tool used to carry shopping items while shopping in a supermarket or department store. This trolley is usually made of metal or plastic and equipped with wheels for easy movement around the store. The use of trolleys is very useful for transporting goods in large quantities. But the more items purchased, the harder it is to push or pull the trolley. To overcome this, automatic trolleys were developed to follow the movement of the user's hand, eliminating the need to push manually. This system uses a camera that can detect hand movements using python and machine learning techniques via mediapipe. This trolley successfully follows hand commands in various conditions such as moving forward, left, right, backwards and stopping with an accuracy rate of 98%. Failure of 2% occurred due to poor lighting in the test. The average speed of the trolley is 17.48 cm/s without load and 7.47 cm/s with a maximum load of 50 kg. The cart cannot exceed this maximum limit. A trolley battery with a discharge limit of 10.5V has a service life of approx. 50 minutes without load and 47 minutes with 50 kg load. The maximum distance between the trolley and the object is 650 cm. The ideal distance for detecting hand movements using a camera is between 50 and 150 cm, with an ideal light intensity of around 10 lux.

Keywords: trolley robot, microcontroller, image processing, hand gesture, microcomputer.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullohi Wabarakatuh.

Puji dan syukur kami sampaikan kepada Allah SWT atas segala anugerah, kekuatan, bimbingan, dan rahmat-Nya. Kiranya berkah dan salam selalu dilimpahkan kepada Rasulullah SAW, keluarga, sahabat, dan semua orang yang setia mengikuti-Nya. Semoga doa ini dikabulkan oleh Allah SWT. Dengan kehendak-Nya, penulis berhasil menyelesaikan tugas akhir yang berjudul:

“ROBOT TROLI BERGERAK MENGGUNAKAN HAND GESTURE RECOGNITION BERBASIS IMAGE PROCESSING”

Dalam upaya untuk memperoleh gelar Ahli Madya (A.Md) di Politeknik Negeri Cilacap, salah satu persyaratan yang harus dipenuhi adalah melakukan pembuatan dan penyusunan tugas akhir.

Penulis menyadari bahwa karya ini belum mencapai tingkat kesempurnaan karena adanya kendala dan hambatan selama proses penulisannya. Oleh karena itu, saran yang konstruktif sangat diharapkan untuk mencapai pengembangan yang lebih optimal dan kemajuan yang lebih baik

Wassalamu'alaikum Warahmatullohi Wabarakatuh.

Cilacap, 20 Juli 2023



Yudi Arrasyid
(Penulis)

UCAPAN TERIMA KASIH

Dengan rendah hati dan penuh syukur, penulis ingin mengucapkan rasa terima kasih yang tak terhingga kepada Allah SWT atas kehadirat-Nya. Penulis juga ingin menyampaikan apresiasi yang tinggi tanpa mengurangi rasa hormat kepada semua pihak yang telah membantu menyelesaikan tugas akhir ini. Penulis juga ingin mengungkapkan rasa terima kasih kepada mereka yang telah memberikan bantuan berharga dalam proses pembelajaran di Politeknik Negeri Cilacap. Oleh karena itu, penulis dengan tulus mengucapkan terima kasih kepada:

- 1) Allah SWT yang telah memberi ridho dan barokah-Nya sehingga dapat terselesaikannya Tugas Akhir ini.
- 2) Bapak Riyadi Purwanto, S.T., M.Eng., selaku Direktur Politeknik Negeri Cilacap
- 3) Bapak Muhamad Yusuf, S.ST., M.T., selaku Ketua Jurusan Teknik Elektronika Politeknik Negeri Cilacap.
- 4) Bapak Arif Sumardiono, S.Pd., M.T., selaku pembimbing I tugas akhir, terima kasih kepada beliau yang selalu memberi masukan beserta solusi pada program dan alat serta perbaikan laporan.
- 5) Ibu Erna Alimudin, S.T., M.Eng., sebagai dosen pembimbing II tugas akhir sekaligus Wali Kelas TE3B, yang telah membina, memberi motivasi dan masukan atas laporan ini.
- 6) Seluruh Dosen Prodi Teknik Elektronika, yang telah memberi ilmu yang bermanfaat untuk bekal masa depan.
- 7) Kepada kedua orang tua yang selalu menjadi support dalam keadaan apapun.
- 8) Mbak Fera Sopiana yang telah mengeluarkan dana untuk penelitian troli belanja sebelumnya sehingga penulis dapat menyelesaikan penelitian lebih cepat.
- 9) Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberi kontribusi positif dalam bentuk apapun itu.
- 10) Terakhir untuk diri sendiri, terima kasih atas usaha dan pantang menyerah yang selama ini dikerjakan untuk suksesnya laporan ini, penulis berharap dimasa yang akan datang kegigihan dan usaha penulis tidak akan pernah luntur.

Semoga Allah SWT selalu memberikan perlindungan, rahmat, dan nikmat-Nya bagi kita semua. Aamiin.

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI	iii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	iv
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
UCAPAN TERIMA KASIH	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR ISTILAH.....	xiv
DAFTAR SINGKATAN.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan dan Manfaat	2
1.4. Batasan Masalah	2
1.5. Metodologi Penelitian.....	3
1.6. Sistematika Penelitian.....	3
BAB II DASAR TEORI	7
2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori	11
BAB III PERANCANGAN SISTEM	27
3.1 Perancangan Troli Otomatis	27
3.2 Diagram Alir.....	30
3.3 Arsitektur Sistem <i>Hand Gesture Recognition</i>	32
3.4 Perancangan Robot Troli	36

3.5	Perancangan Rangkaian Listrik Robot Troli.....	38
3.6	Sistematika Pemrograman Arduino	42
3.7	Sistematika Pemrograman Python Pada Raspberry Pi.....	48
BAB IV HASIL DAN PEMBAHASAN	55	
4.1	Pengujian Gerak Berdasarkan Perintah Dari <i>Hand Gesture</i>	55
4.2	Pengujian <i>Hand Gesture</i> Terhadap Intensitas Cahaya.....	58
4.3	Pengujian Jarak Pendeksiyan <i>Hand Gesture</i> Pada Kamera	68
4.4	Pengujian <i>Tracking</i> Servo Terhadap Objek <i>Hand Gesture</i>	71
4.5	Pengujian Gerak Lurus Troli	72
4.6	Pengujian Kecepatan Troli.....	76
4.7	Pengujian Masa Pakai Baterai	78
4.8	Pengujian Sensor Ultrasonik dan ToF VL53L0X	82
4.9	Pengujian Keseluruhan Alat	83
4.10	Analisa Seluruh Sistem.....	87
BAB V PENUTUP	91	
5.1	Kesimpulan.....	91
5.2	Saran	91
DAFTAR PUSTAKA.....	93	
LAMPIRAN A	A-1	
LAMPIRAN B.....	B-1	
LAMPIRAN C	C-1	
1.1	Biodata Penulis	C-2

DAFTAR GAMBAR

Gambar 2. 1 Troli Belanja.....	11
Gambar 2. 2 Isyarat Tangan	13
Gambar 2. 3 Skeleton Hand	13
Gambar 2. 4 MediaPipe.....	14
Gambar 2. 5 Hand Landmark Pose Mediapipe	15
Gambar 2. 6 MediaPipe ML Pipeline	15
Gambar 2. 7 Library OpenCV	16
Gambar 2. 8 Pemrograman Python	16
Gambar 2. 9 Raspberry Pi 4B.....	17
Gambar 2. 10 Arduino Mega.....	17
Gambar 2. 11 Sensor Ultrasonik HC-SR04.....	18
Gambar 2. 12 Jarak Sensor Ultrasonik	18
Gambar 2. 13 Webcam Logitech.....	19
Gambar 2. 14 Motor Servo SG90.....	20
Gambar 2. 15 Pulse Width Modulation Period	20
Gambar 2. 16 Motor DC Power Window.....	21
Gambar 2. 17 Motor Driver BTS7960	22
Gambar 2. 18 Modul Stepdown XL4015	23
Gambar 2. 19 LCD I2C 20x4	23
Gambar 2. 20 Modul Sensor MPU6050	24
Gambar 2. 21 Sensor Laser Time Of Flight VL53L0X.....	25
Gambar 2. 22 LCD Raspberry 5 Inch.....	25
Gambar 2. 23 Baterai Aki SMT-Power	26
Gambar 3. 1 Diagram Blok Sistem	27
Gambar 3. 2 Flowchart Sistem I.....	31
Gambar 3. 3 Flowchart Sistem II	32
Gambar 3. 4 Prinsip Kerja Mediapipe	33
Gambar 3. 5 Penggunaan Modul Mediapipe	33
Gambar 3. 6 3D Hand Landmark Pose.....	34
Gambar 3. 7 Gesture Tangan.....	34
Gambar 3. 8 Perancangan Mekanika Troli.....	37
Gambar 3. 9 Komponen Box.....	37
Gambar 3. 10 Komponen Box Sensor Ultrasonik	37
Gambar 3. 11 Rangkaian Pada Raspberry PI 4B	38

Gambar 3. 12 Rangkaian Arduino Mega 2560 Pada Robot	39
Gambar 3. 13 Rangkaian Keseluruhan Sistem	41
Gambar 3. 14 Program Gerak Servo Mengikuti Objek	42
Gambar 3. 15 Pemrograman Sesnor Ultrasonik	43
Gambar 3. 16 Pemrograman Driver Motor Untuk Gerak Troli	44
Gambar 3. 17 Sistematika Program LCD	44
Gambar 3. 18 Diagram Blok Sistem Kendali PD	45
Gambar 3. 19 Sistematika Troli Bergerak Lurus.....	47
Gambar 3. 20 Penerimaan Data Serial UART Arduino	48
Gambar 3. 21 Sistematika Pendektsian Tangan.....	49
Gambar 3. 22 Sistematika <i>Hand Gesture</i> Tracking.....	50
Gambar 3. 23 Sistematika <i>Hand Gesture</i> Recognition.....	52
Gambar 3. 24 Titik Koordinat Piksel Pada Computer Vision	52
Gambar 3. 25 Membaca Port Arduino	53
Gambar 3. 26 Pengiriman Data Serial UART Raspberry PI	53
Gambar 4. 1 Pendektsian Pola Gerak Tangan	55
Gambar 4. 2 Posisi Awal Servo Sudut 90°	72
Gambar 4. 3 Total Arus Saat Tidak Bergerak	80
Gambar 4. 4 Total Arus Saat Bergerak	80
Gambar 4. 5 Tegangan Baterai Saat Penuh	81
Gambar 4. 6 Persentase Baterai Saat Penuh.....	81
Gambar 4. 7 Persentase Baterai Saat 25 Menit	82
Gambar 4. 8 Tegangan Baterai Saat Habis.....	82
Gambar 4. 9 Persentase Baterai Saat Habis.....	82
Gambar 4. 10 Tampilan Jarak Sensor Pada LCD	83
Gambar 4. 11 Indikator Baterai Menyala	84
Gambar 4. 12 Tampilan LCD Saat Sistem Sudah Siap	84
Gambar 4. 13 Sistem Siap Menerima Input	85
Gambar 4. 14 Troli Bergerak Maju	85
Gambar 4. 15 Troli Bergerak Mundur	85
Gambar 4. 16 Troli Bergerak Ke Kiri	86
Gambar 4. 17 Troli Bergerak Ke Kanan	86
Gambar 4. 18 Troli Berhenti Bergerak	86
Gambar 4. 19 Respon Ultrasonik	87

DAFTAR TABEL

Tabel 3. 1 Kebutuhan Komponen Sistem	28
Tabel 3. 2 Komponen Perangkat Lunak	30
Tabel 3. 3 Langkah Pengenalan Pola	35
Tabel 3. 4 Pengaturan Input Raspberry Pi 4B	39
Tabel 3. 5 Tabel Pengaturan Komponen Arduino	40
Tabel 3. 6 Konfigurasi Komunikasi Arduino Raspberry	41
Tabel 4. 1 Pengujian Gerak Berdasarkan Perintah Dari Hand Gesture .	56
Tabel 4. 2 Tabel Pengujian Nilai Lux	58
Tabel 4. 3 Hasil Pengujian Deteksi Tangan	60
Tabel 4. 4 Histogram Low Contrast Pencahayaan Pada Metode RGB .	64
Tabel 4. 5 Metode Perbaikan Citra	66
Tabel 4. 6 Tabel Jarak Kamera Terhadap Objek	68
Tabel 4. 7 Hasil Pengujian Response Tracking	71
Tabel 4. 8 Pengujian Gerak Lurus Proporsional-Derivative	73
Tabel 4. 9 Grafik Gerak Lurus Troli	74
Tabel 4. 10 Tabel Hubungan Antara Kp, Ki, dan Kd	76
Tabel 4. 11 Berat Troli dan Komponen	76
Tabel 4. 12 Pengujian Kecepatan Tanpa Beban	77
Tabel 4. 13 Hasil Pengujian Kecepatan Dengan Beban	78
Tabel 4. 14 Tabel Daya Komponen.....	79
Tabel 4. 15 Penggunaan Lama Pakai Baterai Tanpa Beban	80
Tabel 4. 16 Penggunaan Lama Pakai Baterai Dengan Beban	80
Tabel 4. 17 Jarak Ultrasonik dan Laser Pada Halangan	83

DAFTAR ISTILAH

<i>Hand Gesture</i>	:	Gesture tangan dan jari yang digunakan untuk berkomunikasi atau menyampaikan maksud tanpa menggunakan kata-kata.
<i>Output</i>	:	Keluaran.
<i>Input</i>	:	Masukan.
<i>Modern</i>	:	Terbaru.
<i>lux</i>	:	Satuan unit pengukuran yang digunakan untuk mengukur tingkat pencahayaan atau fluks bercahaya per satuan luas.
Inisialisasi	:	Menetapkan mode operasi atau membaca konfigurasi dari file atau perangkat eksternal. inisialisasi dilakukan saat sistem atau program dimulai.
Robot	:	mesin yang dirancang untuk melakukan tugas secara otonom atau semi-otonom, yang telah diprogram untuk melakukan serangkaian instruksi.
<i>Library</i>	:	Kumpulan kode atau program yang telah ditulis sebelumnya dan disediakan untuk digunakan kembali oleh programmer.
<i>Open Source</i>	:	Kode sumber dari perangkat lunak dibuat tersedia untuk umum dan dapat diakses, digunakan, dimodifikasi, dan distribusikan secara bebas.
<i>Tracking</i>	:	Proses mengikuti objek tertentu secara otomatis dari frame ke frame dalam sebuah video atau gambar.
<i>Computer Vision</i>	:	Bidang ilmu komputer dan kecerdasan buatan yang berfokus pada memungkinkan komputer menafsirkan dan menganalisis gambar dan video dengan cara yang mirip dengan penglihatan manusia.

<i>Image Processing</i>	:	Metode pemrosesan data citra digital.
<i>Microcomputer</i>	:	Sebuah komputer yang dirancang dengan ukuran kecil dan sumber daya yang terbatas.
<i>Flowchart</i>	:	Alur kerja atau algoritma dalam suatu program atau sistem.
<i>Display</i>	:	Perangkat elektronik yang digunakan untuk menampilkan output visual, seperti teks, gambar, dan video.
<i>Pixel</i>	:	Titik-titik kecil yang membentuk gambar atau tampilan digital.
<i>Human Machine Interface</i>	:	Interaksi antara manusia dan mesin.
<i>Controller</i>	:	Mengontrol dan mengatur suatu sistem atau perangkat elektronik.
<i>BBox</i>	:	Bounding Box, kotak persegi atau persegi panjang yang mengelilingi objek di dalam gambar.
<i>Delay</i>	:	Metode dalam pemrograman yang digunakan untuk menunda eksekusi program untuk sementara waktu.

DAFTAR SINGKATAN

DC	:	<i>Direct Current</i>
cm	:	<i>Centimeter</i>
M	:	<i>Meter</i>
Nm	:	<i>Nanometer</i>
mm	:	<i>Milimeter</i>
Kg	:	<i>Kilogram</i>
ToF	:	<i>Time of Flight</i>
s	:	<i>Second</i>
PID	:	<i>Proprtional Integral Derivative</i>
PWM	:	<i>Pulse Width Modulation</i>
HMI	:	<i>Human Machine Interface</i>
OpenCV	:	<i>Open Computer Vision</i>
PC	:	<i>Personal Computer</i>
CPU	:	<i>Central Processing Unit</i>
GPU	:	<i>Graphic Processing Unit</i>
I/O	:	<i>Input/Output</i>
V	:	<i>Volt</i>
I	:	<i>Arus</i>
W	:	<i>Daya</i>
Kb	:	<i>Kilobites</i>
Hz	:	<i>Hertz</i>
D	:	<i>Degree</i>
°	:	<i>Derajat</i>
VCC	:	<i>Voltage Common Collector</i>
GND	:	<i>Ground</i>
SDA	:	<i>Serial Data</i>
SDL	:	<i>Serial Clock</i>
USB	:	<i>Universal Serial Bus</i>
GPIO	:	<i>General Pin Input Output</i>
UART	:	<i>Universal Asynchronous Receiver Transmitter</i>
SPI	:	<i>Serial Peripheral Interface</i>
12C	:	<i>Inter Integrated Circuit</i>
RAM	:	<i>Random Access Memory</i>

DAFTAR LAMPIRAN

- LAMPIRAN A *LISTING PROGRAM ARDUINO*
- LAMPIRAN B *LISTING PROGRAM PYTHON*
- LAMPIRAN C HASIL ALAT

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seorang individu umumnya memperoleh keperluan sehari-hari seperti makanan, minuman, barang-barang kebersihan, dan barang rumah tangga lainnya melalui pembelian di supermarket. Supermarket umumnya menawarkan berbagai pilihan produk, merek, dan variasi dengan harga yang bersaing. Ketika berbelanja, kita tentunya membutuhkan sarana untuk mengangkut barang belanjaan. Agar tidak lelah membawa barang, terdapat beberapa jenis alat yang dapat digunakan di supermarket, seperti keranjang belanja dan troli.

Troli supermarket merupakan peranti yang digunakan untuk mengangkut barang belanjaan saat berbelanja di supermarket atau toko serba ada^[1]. Umumnya, troli ini terbuat dari bahan logam atau plastik dan dilengkapi dengan roda agar dapat dengan mudah dipindahkan di dalam toko. Penggunaan troli ini memberikan kemudahan kepada pelanggan dalam membawa dan menggerakkan barang dalam jumlah yang besar. Sebagai pengguna, kita perlu mendorong troli tersebut agar dapat membantu pergeseran barang dengan lebih efisien^[2].

Pada umumnya, troli belanja digunakan dengan cara didorong atau ditarik oleh pengguna. Semakin banyak barang belanja yang diangkut, semakin besar energi yang dibutuhkan untuk mendorong atau menarik troli tersebut^[3]. Hal ini dapat menyebabkan kelelahan pada pengguna, terutama jika jarak yang ditempuh cukup jauh atau troli digunakan untuk mengangkut beban yang sangat berat, sehingga pembeli akan berpikir untuk mengurangi barang belanjanya^[4]. Maka dari itu, diperlukanlah sebuah robot troli yang mampu mengikuti pergeseran tangan pengguna, sehingga pengguna tidak perlu melakukan dorongan pada troli.

Penelitian saat ini akan menitikberatkan perbaikan teknologi penemuan sebelumnya dengan mengganti pendektsian dengan menggunakan *hand gesture recognition* untuk mengurangi kelemahan penelitian sebelumnya serta memperbaiki kualitas dan terobosan terkait teknologi troli otomatis khusunya pada gerak lurus troli, upaya keras telah dilakukan untuk mengembangkan antarmuka cerdas antara pengguna dan sistem berbasis komputer berdasarkan gerak tubuh manusia. *Gesture* menyediakan antarmuka yang intuitif untuk manusia dan komputer.

Dengan demikian, antarmuka berbasis gesture seperti itu tidak hanya bisa mengganti perangkat antarmuka umum, tetapi juga bisa dieksplorasi untuk memperluas fungsinya. Penggunaan sistem kendali proporsional dan derivative pada robot sangat berpengaruh dalam mengurangi error pergesture agar troli dapat bergerak lurus. Diharapkan penelitian lanjutan ini dapat memperbaiki gesture pada troli otomatis serta dapat berkontribusi pada bidang keilmuan terkait sistem kontrol dan mikrokontroller.

1.2. Rumusan Masalah

Dengan mengacu pada permasalahan yang ada dalam latar belakang, maka rumusan masalah yang akan dijelaskan adalah sebagai berikut:

- a. Bagaimana perancangan mekanik dan elektrikal pada robot troli?
- b. Bagaimana cara menyusun sistem elektrik dan algoritma untuk memungkinkan troli bergerak sesuai dengan pergesture tangan?

1.3. Tujuan dan Manfaat

1.2.1. Tujuan

Dengan mempertimbangkan latar belakang yang telah disebutkan, maka tujuan dari penelitian ini adalah untuk menggerakkan robot troli sesuai perintah pengguna menggunakan gesture tangan sehingga memudahkan pelanggan dalam membawa barang yang berat dan membantu pelanggan dalam berbelanja.

1.2.2. Manfaat

Berikut ini adalah beberapa manfaat yang dapat diperoleh dari perancangan Tugas Akhir ini:

- a. Membantu pelanggan dalam membawa barang yang banyak dan berat.
- b. Mengurangi tenaga dan mencegah kelelahan saat berbelanja.
- c. Meningkatkan penjualan pada supermarket.

1.4. Batasan Masalah

Untuk mencegah terjadinya keluaran yang tidak sesuai, tujuan utama dalam penyusunan tugas akhir ini adalah dengan menetapkan batasan masalah yang mencakup beberapa hal, antara lain.:

- a. Troli hanya diprogram untuk mendeteksi gesture tangan yang sudah diatur dalam program.
- b. Hanya membahas deteksi *hand gesture recognition* yang diterapkan pada robot troli menggunakan kamera dan raspberry pi sebagai pemroses gambar.
- c. Troli hanya mendeteksi 1 tangan pelanggan saja dan tidak dapat diarahkan dengan menggunakan 2 tangan sekaligus.

1.5. Metodologi Penelitian

Metode yang digunakan dalam pembuatan Tugas Akhir ini yaitu :

- a. Identifikasi masalah
Metode ini melakukan pengamatan dan mengobservasi penggunaan troli pada supermarket serta menemukan permasalahan terkait penggunaan troli manual.
- b. Studi literatur
Metode ini memperoleh data dan informasi yang relevan dengan sistem yang diinginkan melalui proses pencarian literatur
- c. Perancangan sistem.
Metode ini merupakan tahap perancangan yang dilanjutkan dengan pembuatan sistem hingga selesai sesuai dengan rencana yang diinginkan.
- d. Pengujian alat
Metode ini digunakan untuk memperoleh hasil pengujian dan data hasil percobaan penggunaan alat.
- e. Perbaikan alat
Metode ini digunakan untuk memperbaiki alat apabila perlu dilakukan perbaikan agar alat mencapai hasil maksimal.
- f. Penyusunan laporan
Metode ini merupakan hasil dari perancangan tugas akhir dan disusun mulai dari perancangan, percobaan, hingga kesimpulan.

1.6. Sistematika Penelitian

Untuk mendapatkan gambaran yang jelas tentang susunan materi yang tercakup dalam laporan akhir ini, maka metodologi penyusunan laporan akhir adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini mencakup hal-hal berikut:

- 1) Latar Belakang
Berisi argumentasi tentang alasan-alasan penting yang mendorong mengusulkan tugas akhir tersebut, dengan mengutip berbagai sumber literatur. Data pendukung sebanyak mungkin - informasi atau pendapat dari pihak lain untuk mendapatkan masalah..
- 2) Tujuan dan Manfaat
Menguraikan tujuan yang ingin dicapai dalam Tugas Akhir, seperti menguji, menerapkan, atau mengembangkan gejala, konsep, atau dugaan, serta membuat suatu model. Keuntungan merujuk pada efek positif atau manfaat praktis yang diperoleh dari hasil akhir Tugas Akhir, dilihat dari berbagai sudut pandang.
- 3) Rumusan Masalah
Menjelaskan pokok masalah proyek akhir, sekaligus menggambarkan masalah-masalah yang harus dipecahkan untuk mencapai tujuan.
- 4) Batasan Masalah
Menyatakan tentang keterbatasan masalah saat menulis TA sehingga pembaca memahami ruang lingkup pekerjaan yang harus dilakukan.
- 5) Metodologi
Memberikan pendekatan desain akhir melibatkan pengembangan metode, strategi, pendekatan, atau serangkaian langkah yang dapat digunakan untuk menyelesaikan atau mengatasi masalah yang ada dalam Tugas Akhir.
- 6) Sistematika Penulisan
Mengungkapkan cara penyusunan buku ini serta menjelaskan isi dari setiap bagian atau bab yang tercatat di dalamnya.

BAB II DASAR TEORI

Bagian ini menguraikan konsep-konsep fundamental dan teori-teori yang diperoleh dari sumber-sumber resmi seperti buku, jurnal, artikel, atau tesis yang telah diterbitkan sebelumnya. Tujuan dari penjelasan ini adalah untuk mendapatkan wawasan yang dibutuhkan guna menyelesaikan masalah yang dihadapi. Dalam konteks ini, referensi yang dianggap relevan adalah karya-karya yang dipublikasikan secara formal, seperti buku, jurnal, artikel, atau tesis, yang dapat memberikan landasan ilmiah yang kuat. Informasi yang bersifat tidak dipublikasikan, seperti catatan

kuliah, pendapat lisan, pengalaman pribadi, atau opini subyektif, tidak dianggap sebagai sumber referensi yang valid dalam konteks ini.

BAB III PERANCANGAN SISTEM

Bab ini menguraikan secara terperinci tentang perencanaan komponen sistem yang meliputi analisis sistem, analisis kebutuhan sistem, serta perancangan sistem yang mencakup pembuatan diagram blok dan flowchart.

BAB IV HASIL DAN PEMBAHASAN

Bab ini mengandung informasi tentang hasil yang dicapai, termasuk nilai parameter yang diuji atau simulasi, dan lain sebagainya. Hasil yang dihasilkan dianalisis dan diinterpretasikan agar pembaca dapat memahami makna penting secara kuantitatif dan kualitatif dari hasil yang diperoleh.

BAB V PENUTUP

Bab ini memuat simpulan dan rekomendasi berdasarkan pencapaian yang telah dicapai, yang akan bermanfaat bagi pengembangan sistem yang lebih baik di masa depan. Simpulan tersebut haruslah praktis dan dapat dipahami dengan mudah.

~Halaman ini sengaja dikosongkan~

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan literatur dilaksanakan dengan menghimpun informasi dari jurnal yang sudah ada sebagai rujukan dalam pengembangan teknologi pengenalan gesture tangan untuk mengendalikan robot troli bergerak.

Pada penelitian sebelumnya telah dilakukan penelitian tentang troli otomatis yang menggunakan pengenalan warna untuk pengikut manusia menggunakan kamera berjudul “Shopping Trolley Automatically Move Following Color With Camera”, penelitian tersebut memang dikatakan sukses dalam mengenali warna serta troli dapat mengikuti manusia. Namun terdapat kelemahan dalam penelitian tersebut yakni warna sekitar rentan terdeteksi oleh kamera sehingga troli dapat bergerak sendiri apabila disupermarket tersebut terdapat warna yang sama dengan yang digunakan oleh pengguna troli serta kendali troli yang tidak dapat bergerak lurus juga perlu dilakukan perbaikan agar nantinya penelitian tersebut dikatakan baik dan berjalan sesuai rencana^[5].

Pada penelitian yang berjudul “Robot Pengikut Target Berdasarkan Bentuk dan Warna Menggunakan Metode HSV Untuk Aplikasi Assistant Robot” telah berhasil menciptakan sebuah robot yang mampu secara otomatis mengikuti target dengan bantuan kamera. Dalam pengujian pendekripsi target berdasarkan jarak, tingkat keberhasilan yang dicapai mencapai 87.6%, dengan rentang jarak minimal 30 cm hingga maksimal 350 cm. Kelemahan penelitian ini yakni masih berupa prototipe kotak dengan sebuah laptop sebagai pemroses gambar. Pendekripsi masih menggunakan warna dengan objek benda. Objek mudah menghilang dan mudah terdeteksi apabila objek lain berwarna sama^[6].

Pada penelitian terkait troli yang berjudul “An Intelligent Following Sensor Shopping Cart” dijelaskan bahwa penelitian menggunakan deteksi warna hasil dari tes pemilihan warna menunjukkan bahwa warna yang cocok untuk deteksi merah yang memberikan 41,52% dari perbedaan warna dengan latar belakang pemandangan diikuti oleh Hijau, Biru, Kuning dan Putih Troli berhasil mendekripsi warna dengan

toleransi warna 18%. Kelemahan dari penelitian ini yakni terletak pada troli yang masih berupa *prototype* mobil kecil yang pendeksiannya mudah berubah ketika intensitas cahaya dalam ruangan menurun^[7].

Pada penelitian terkait troli yang berjudul “Automatic Human Following Trolley Using Raspberry” dijelaskan bahwa robot troli dapat digunakan di pusat perbelanjaan untuk pengalaman berbelanja yang lebih baik. Troli hanya mengikuti pengguna tertentu berdasarkan nilai HSV warna bajunya, sistem bekerja dengan baik di sebagian besar kondisi kecuali untuk beberapa kondisi yaitu - jika banyak orang dengan baju yang sama warna dan pola yang sama datang maka robot mungkin bingung dan pada malam hari jika seluruh latar belakang berwarna putih dan warna baju juga putih maka mungkin ada beberapa masalah. Sehingga kelemahan paling fatal dari penelitian ini terletak pada penelitian ini yakni jika baju pengguna lain sama maka troli dapat mengikuti orang lain^[8].

Pada penelitian yang berjudul “Hand Gesture Controlled Robot” membuat robot yang bergerak dengan *hand gesture* menggunakan Accelerometer dan Gyroskop gesture untuk melakukan manuver pada robot, Kontrol robot dengan *hand gesture* memberikan cara alternatif mengendalikan robot. Kelemahan terkait penelitian ini terletak pada penggunaan perangkat transmisi kecil seperti sarung tangan, tidak efisien dalam penggunaan^[9].

Pada penelitian yang berjudul “Implementasi Gesture Tangan Terhadap Navigasi Robot Beroda Menggunakan Teknik Accelerometer” mengembangkan sebuah robot yang memiliki kesamaan dengan penelitian sebelumnya mengenai robot yang telah menjadi fokus banyak penelitian, terutama dalam hal pengendalian gesture robot. Penelitian ini memanfaatkan teknik Accelerometer, dan temuan penelitian menunjukkan bahwa sensor accelerometer pada MPU 6050 dapat mengidentifikasi gesture pergelangan tangan sebagai cara untuk mengendalikan pergesture robot. Sehingga robot mampu melakukan pergesture maju dan mundur. menggunakan komunikasi *gesture* tangan dengan respon yang cepat. Kelemahan penelitian ini terletak pada penggunaan perangkat trasnmisi kecil yang tidak efisien, Semakin jauh jarak bluetooth maka komunikasi akan berhenti dan harus dipairing kembali^[10].

Pada penelitian yang berjudul “Pengendalian Gerak Robot menggunakan Semantik Citra Gestur Tangan Manusia” penelitian ini menggunakan semantik citra dimana objek pendekripsi menggunakan sebuah dataset yang sudah ditangkap sebelumnya, dengan mencocokan hasil gambar dengan dataset maka pendekripsi *gesture* tangan dapat dilakukan. Robot memiliki tingkat keberhasilan sebanyak 60% dalam mengambil objek benda dalam sepuluh percobaan, dengan kesalahan sebesar 40%. Di sisi lain, robot berhasil memindahkan objek sebanyak 50% dalam enam percobaan, dengan tingkat kesalahan sebesar 50%. Kelemahan pada penelitian ini terletak pada tingkat kesalahan dan error yang sangat besar dikarenakan jarak yang ditangkap tidak sesuai dengan dataset semantik citra grayscale sehingga robot tidak dapat mendekripsi apabila objek tidak cocok dengan dataset^[11].

Penelitian saat ini berbeda dengan penelitian sebelumnya dimana akan mengembangkan pendekripsi dengan menggunakan pengenalan isyarat tangan untuk mengurangi kelemahan penelitian sebelumnya yakni penelitian fera sopiana yang memiliki kelemahan pada pendekripsi warna dengan menggunakan kertas warna dimana warna yang sama akan mudah terdeteksi didalam supermarket dan juga troli pada penelitian sebelumnya masih menggunakan kertas untuk pendekripsi. Tugas akhir ini juga memperbaiki kualitas dan terobosan terkait teknologi troli otomatis penelitian sebelumnya khususnya pada gerak lurus troli menggunakan gyroscope dengan sistem kendali proporsional dan derivative, menggunakan LCD sebagai human machine interface pada robot, serta penggunaan motor servo juga diterapkan untuk mengikuti objek tangan pengguna, sehingga tangan pengguna tidak menghilang ketika telah dideteksi oleh kamera. Diharapkan penelitian lanjut ini dapat menutupi kekurangan-kekurangan yang telah ada pada penelitian sebelumnya.

Tabel 2. 1 Perbandingan Jurnal

Judul Jurnal	Metode	Kelemahan	Kelebihan Penulis
Shopping Trolley Automatically Move Following Color With Camera, Fera Sopiana (2022)	Pengenalan warna untuk pengikut manusia menggunakan kamera	warna sekitar rentan terdeteksi oleh kamera sehingga troli dapat bergerak sendiri, serta kendali troli yang tidak dapat bergerak lurus	Mengganti pendekripsi dengan menggunakan hand gesture recognition, serta menambahkan kendali Proporsional-Derivative pada robot

Judul Jurnal	Metode	Kelemahan	Kelebihan Penulis
Robot Pengikut Target Berdasarkan Bentuk dan Warna Menggunakan Metode HSV Untuk Aplikasi Assistant Robot, Syahri Muharom (2021)	Sistem deteksi objek berdasarkan bentuk dan warna dengan laptop	Masih berupa prototipe, Laptop sebagai pemroses gambar. Pendekripsi masih menggunakan warna mudah terdeteksi apabila objek lain berwarna sama	Mengganti Pendekripsi dengan Hand Gesture dengan keakuratan 90%, Menggunakan mikrokomputer Raspberry, Menggunakan Sistem Kendali PD untuk gerak lurus robot.
An Intelligent Following Sensor Shopping, Samidi (2019)	Menggunakan Arduino yang berkomunikasi dengan ponsel menggunakan aplikasi Android pendekripsi warna melalui Bluetooth	Harus Dipair dengan bluetooth yang terputus. Pendekripsi warna mudah hilang dengan ketika intensitas cahaya menurun. Berupa Prototipe	Hand Gesture hanya membutuhkan 10 lux dalam pendekripsi. Tidak terpengaruh penurunan cahaya
Automatic Human Following Trolley Using Raspberry, Kiran Ingole (2021)	Trolley hanya mengikuti pengguna tertentu berdasarkan nilai HSV warna bajunya dengan MobileNet SSD	Jika banyak orang dengan baju yang sama warna dan pola yang sama datang maka robot mungkin bingung	Robot penulis hanya mendekripsi 1 tangan saja. Tidak dapat menggunakan 2 tangan
Hand Gesture Controlled Robot, Kathiravan Sekar (2020)	Menggunakan accelerometer dengan arduino uno, robot dikendalikan oleh gerakan tangan manusia. Namun hal ini perlu memakai perangkat accelerometer	Robot harus dikontrol menggunakan perangkat accelerometer, Kontrol Robot tidak efisien dalam segi ekonomi dan penggunaan.	Hand Gesture Kamera tidak perlu perangkat tambahan. Pendekripsi hingga 650cm. Tidak perlu pairing dengan modul RF
Implementasi Gerakan Tangan Terhadap Navigasi Robot Beroda Menggunakan Teknik Accelerometer, Bagus Prasetyo (2021)	Teknologi robot saat ini telah banyak diteliti terutama dari sisi kontrol gerak robot. Penelitian ini mengadopsi teknik Accelerometer.	Sejak jauh jarak bluetooth maka komunikasi akan berhenti dan harus dipairing kembali	Kendali hand gesture berbasis image processing tidak perlu dipairing kembali
Pengendalian Gerak Robot menggunakan Semantik Citra Gestur Tangan Manusia, Susetyo Bagas (2020)	Penelitian ini mengembangkan robot lengan beroda pemindah benda menggunakan perintah semantik citra gestur tangan	Tingkat kesalahan dan error yang sangat besar dikarenakan jarak yang ditangkap tidak sesuai dengan dataset semantik citra grayscale sehingga robot tidak dapat mendekripsi apabila	Menggunakan Mediapipe yang merupakan library training tensorflow lite dari database google yang memiliki ribuan gambar yang telah ditraining

Judul Jurnal	Metode	Kelemahan	Kelebihan Penulis
		objek tidak cocok dengan dataset	

2.2 Landasan Teori

2.2.1. Troli Belanja

Troli biasa digunakan untuk menaruh makanan dengan jumlah banyak yang akan dibeli. Dengan menggunakan troli kita tidak perlu lagi harus mengangkut barang belanja yang banyak dan berat, cukup dengan memasukan barang belanja kedalam troli kita daat memindahkan barang belanja menjadi mudah. Kereta belanja atau *Trolley* ditemukan oleh seorang pemilik toko di Amerika bernama Sylvan Nathan Goldman pada tanggal 4 Juni 1937, Goldman memiliki sebuah toko swalayan di Kota Oklahoma, Amerika Serikat^[12]. Ia melihat pelanggan yang kesulitan membawa banyak barang belanjaan sekaligus dan memutuskan untuk menciptakan sebuah solusi yang lebih praktis. Kereta belanja atau *Trolley* berbentuk keranjang dengan empat roda dan pegangan dorong di bagian belakang^[13]. Pada tugas akhir ini troli belanja digunakan sebagai tempat pengangkut barang dan untuk menempatkan berbagai komponen seperti motor, battery, mikrokontroller, driver, kamera dan sebuah LCD.



Gambar 2. 1 Troli Belanja

2.2.2. Image Processing

Citra atau Gambar adalah representasi dari dua objek dimensi dunia visual, beberapa disiplin ilmu yang menyangkut tentang citra antara lain dalam bidang seni, penglihatan manusia, astronomi, teknologi dan sebagainya. Dimana citra ini merupakan kumpulan piksel atau titik warna yang memiliki bentuk dua dimensi^[14].

Pengolahan citra, yang juga dikenal sebagai pemrosesan gambar, merupakan suatu teknik yang digunakan untuk memanipulasi dan menganalisis gambar dengan maksud untuk mengubah informasi yang terkandung di dalamnya atau mengekstrak karakteristik khusus dari gambar tersebut^[15]. Pengolahan gambar dapat dilakukan pada gambar yang berasal dari berbagai sumber seperti kamera, scanner, atau file gambar digital^[16]. Pengolahan gambar memiliki berbagai aplikasi, seperti pengenalan pola, deteksi objek, pemrosesan medis, pengenalan wajah, pengolahan citra satelit, dan masih banyak lagi^[17]. Pada tugas akhir ini *image processing* digunakan untuk mengolah citra pada *gesture* tangan yang dideteksi sehingga *gesture* tersebut dapat menggesture troli.

2.2.3. *Computer Vision*

Computer vision adalah gabungan antara pemrosesan gambar dan pengenalan pola. Bidang ini mengadopsi kemampuan penglihatan manusia dalam memperoleh informasi dari gambar^[18]. Ujuan utama computer vision adalah menciptakan model yang dapat mengekstrak data dan informasi dari gambar^[19]. Computer Vision menggunakan gambar digital dari kamera sebagai *Input* dan *Neural Network* untuk mendeteksi dan mengklasifikasikan objek serta mengidentifikasi apa yang dilihat komputer^[20].

Pada tugas akhir ini visi komputer digunakan untuk mengolah penglihatan dari input kamera sehingga dapat mengenali objek tangan.

2.2.4. *Hand Gesture*

Isyarat tangan atau *hand gesture* merupakan salah satu aspek bahasa tubuh yang dapat disampaikan melalui bagian tengah tubuh, telapak tangan, posisi jari dan bentuk yang dibangun oleh tangan. Gesture tangan dapat dibagi menjadi dua kategori, yaitu statis dan dinamis^[21]. Seperti yang dapat disimpulkan dari namanya, gestur statis merujuk pada posisi tangan yang tidak berubah, sementara gestur dinamis terdiri dari serangkaian pergesture tangan seperti gesture melambai. Ada berbagai macam gesture tangan dalam sebuah gestur; misalnya, jabat tangan bervariasi dari satu orang ke orang lain^[22].

Pada tugas akhir ini menggunakan visi komputer sebagai pendekripsi gerak tangan. Walaupun sebenarnya pendekripsi *gesture*

tangan memiliki dua jenis, yakni menggunakan perangkat yang dipakai pada tangan dan menggunakan kamera untuk melihat.

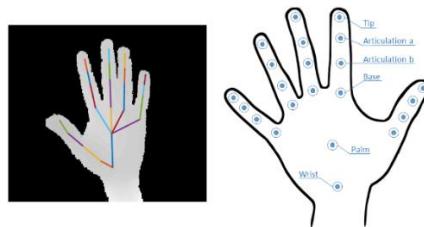
Perangkat yang dapat dikenakan pada tangan dapat menangkap data perubahan tangan yang lebih akurat dan dengan demikian mencapai pengenalan yang lebih tinggi ketepatan. Namun cara ini membutuhkan peralatan khusus yang tidak hanya mahal, tetapi juga tidak nyaman untuk penggunaan sehari-hari. Sebaliknya, visi komputer mengumpulkan gesture isyarat tangan pengguna dengan kamera dan tidak ada peralatan lainnya. Oleh karena itu, metode ini lebih nyaman. Selanjutnya, pengenalan gesture menggunakan visi komputer sederhana dan mudah digunakan, tetapi juga mendapatkan pengenalan yang tepat.



Gambar 2. 2 Isyarat Tangan

2.2.5. Skeleton Hand Gesture Landmark Recognition

Skeleton Based Recognition adalah sebuah pengenalan pola yang menggunakan sebuah kerangka model Convolutional Neural Network (CNN) yang termasuk kedalam image processing. Di mana berbagai representasi dataset atau kumpulan data tangan dapat dijadikan model agar dapat digunakan untuk klasifikasi^[23].



Gambar 2. 3 Skeleton Hand

Beberapa kerangka model mencakup orientasi sendi, ruang antara sendi, lokasi sendi kerangka, derajat sudut antara sendi dan lintasan kelengkungan sendi. Di mana menggunakan algoritma yang menghitung kecocokan dataset dengan kecocokan set titik kerangka secara langsung menggunakan kamera. Ketika gambar secara langsung menemukan kecocokan yang sama pada model kerangka dataset, gambar tersebut akan diklasifikasikan sesuai dengan dataset yang diberikan, metode ini hampir sama dengan menggunakan metode deep learning^[24].

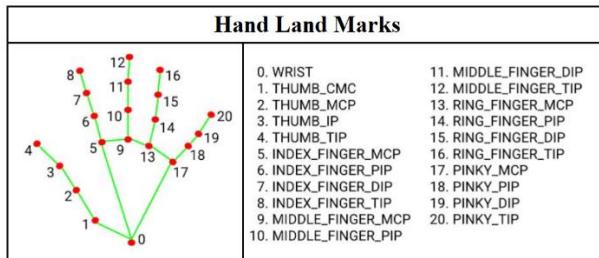
2.2.6. MediaPipe

Mediapipe adalah modul library Python yang memungkinkan pengguna untuk menggunakan fungsi dan algoritma yang ada di dalam Mediapipe untuk memproses data gambar atau video dengan mudah^[25]. Keunggulan MediaPipe adalah kemampuannya untuk menggunakan data time series dan konfigurasi pipeline yang memungkinkan melakukan multiprocessing secara paralel.



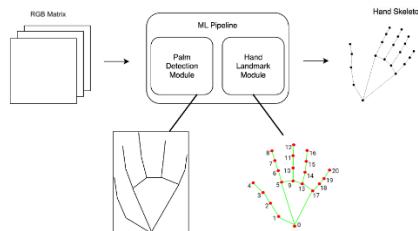
Gambar 2. 4 MediaPipe

Kumpulan data Google yang besar dan beragam telah digunakan untuk melatih sejumlah besar algoritma deteksi dan pelacakan tubuh manusia di MediaPipe^[26]. Model yang diusulkan menggunakan kerangka node dan landmark untuk melacak sendi penting pada berbagai aspek tubuh manusia^[27]. Hasil dari bentuk 3 dimensi tersebut kemudian digunakan dalam model Tensorflow lite yang dibuat oleh pengembang Google untuk membuat aliran informasi pipeline^[28]. Pipelines di MediaPipe terdiri dari node pada grafik, yang sering disediakan dalam bentuk file pbtxt^[29].



Gambar 2. 5 Hand Landmark Pose Mediapipe

Proses kerja dari Mediapipe menawarkan representasi posisi tangan dalam 2D dengan menggunakan peta gambar kedalaman yang telah diteliti oleh Google. Untuk memperoleh gambaran yang lebih rinci, diperlihatkan pada Gambar 2.6.



Gambar 2. 6 MediaPipe ML Pipeline

2.2.7. OpenCV

OpenCV, singkatan dari Open Source Computer Vision Library, adalah perpustakaan perangkat lunak dengan sumber terbuka yang mencakup pengembangan perangkat lunak untuk Teknologi Visi Komputer. Visi Komputer merupakan salah satu bidang dalam pemrosesan citra yang memungkinkan komputer untuk memiliki kemampuan penglihatan mirip dengan manusia. Dengan adanya visi komputer ini, komputer dapat mengambil keputusan, melakukan tindakan, serta mengenali berbagai objek. Beberapa contoh aplikasi visi komputer termasuk deteksi wajah, pengenalan wajah, pelacakan wajah atau objek, serta pelacakan jalur^[30]. Pada tugas akhir ini OpenCV digunakan sebagai library dalam pemrosesan *gesture* tangan manusia.



Gambar 2. 7 Library OpenCV

2.2.6 Bahasa Pemrograman Phyton

Python adalah bahasa pemrograman komputer serbaguna yang interaktif. Filosofi desain bahasa pemrograman Python sangat terfokus pada membaca satu atau lebih kode atau skrip^[31]. Dibandingkan dengan bahasa pemrograman lain, Python tentu memiliki beberapa kualitas yang baik, bahasa yang menggabungkan fitur yang jelas dengan kode dan sintaks perintah yang mudah dipahami. Dalam tugas akhir ini bahasa pemrograman python digunakan untuk melakukan pemrograman kamera komputer raspberry.



Gambar 2. 8 Pemrograman Python

2.2.7 Raspberry Pi 4B

Raspberry Pi merupakan sebuah perangkat komputer mini yang memiliki kemampuan input dan output yang serupa dengan papan mikrokontroler. Awalnya dirancang sebagai alat pendidikan yang bertujuan membantu siswa sekolah dalam mempelajari pemrograman komputer. Pada setiap tahap pengembangannya, Raspberry mengembangkan model dengan spesifikasi yang lebih baik. Dibekali prosesor Quad core 64-bit ARM-Cortex A72 1.5GHz, memiliki RAM sebesar 4GB dengan decode video H.264 sebesar 1080p 60fps support dual HDMI display sampai kualitas 4Kp 60fps^[32].

Dalam tugas akhir ini menggunakan Raspberry Pi sebagai pemroses input dari sensor dan kamera, mengklasifikasikan objek menggunakan image processing untuk mendapatkan informasi yang tepat dan mengirim perintah ke Arduino Mega 2560.



Gambar 2. 9 Raspberry Pi 4B

2.2.8 Arduino Mega 2560

Papan elektronik Arduino Mega2560 merupakan perangkat open source. Papan ini mempunyai kristal dengan kecepatan 16 MHz, memiliki 54 pin digital dan 16 pin analog, serta dilengkapi dengan tombol reset. Selain itu, Arduino Mega2560 juga memiliki 4 pasang header yang meliputi UART, USB, dan ICSP. Tegangan Operasi 5V sedangkan tegangan input dapat diberikan dari 6V-20V. Papan ini kompatibel dengan Arduino shield^[33]. Dalam penelitian ini, papan tersebut berperan sebagai pusat kendali utama sistem, yang memproses informasi dari sensor ultrasonik, sensor ToF, Raspberry Pi, dan mengendalikan aktivasi motor DC dan Servo.



Gambar 2. 10 Arduino Mega

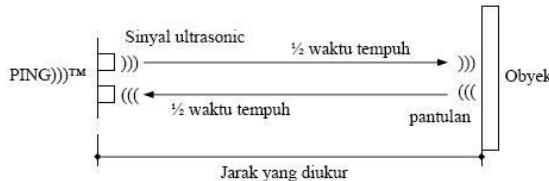
2.2.8. Sensor Ultrasonik HC-SR04

Sensor ultrasonik merupakan jenis sensor yang mengubah energi fisik berupa suara menjadi energi listrik. Nama "ultrasonik" diberikan karena sensor ini menggunakan gelombang suara dengan frekuensi ultrasonik^[34].



Gambar 2. 11 Sensor Ultrasonik HC-SR04

Sensor ultrasonik bekerja dengan mengirimkan gelombang ultrasonik (output) dan menerima gelombang ultrasonik yang dipantulkan (input) untuk mengukur jarak target di depannya. Sensor ultrasonik merupakan jenis sensor yang mengubah besaran fisik (suara) menjadi besaran listrik. Nama "sensor ultrasonik" diberikan karena sensor ini menggunakan gelombang ultrasonik.



Gambar 2. 12 Jarak Sensor Ultrasonik

Dalam penelitian ini, sensor HC-SR04 digunakan sebagai alat pendekripsi jarak menggunakan sensor jarak. Untuk mendekripsi jarak objek, sensor PING digunakan dengan cara mengirimkan gelombang ultrasonik frekuensi 40 kHz pada saat $t = 200 \mu\text{s}$, dan kemudian mendekripsi pantulannya. Sensor PING mengirimkan gelombang ultrasonik dengan mengikuti pulsa trigger yang diberikan oleh mikrokontroler sebagai pengontrol (dengan minimal pulsa trigger tout $2\mu\text{s}$).

Gelombang ultrasonik tersebut menjalar melalui udara pada kecepatan 344 m/s, kemudian mencapai objek dan memantul kembali ke sensor. Setelah mengirimkan gelombang ultrasonik, sensor PING menghasilkan pulsa keluaran tinggi ke pin SIG, dan ketika gelombang pantulan terdeteksi, sensor PING menghasilkan keluaran rendah ke pin SIG. Lebar pulsa yang dihasilkan oleh sensor PING berhubungan dengan dua kali waktu perjalanan (t_{IN}) gelombang ultrasonik, dan dari lebar pulsa tersebut dapat diukur jarak ke target^[35].

Maka jarak yang harus diukur adalah:

$$S = \frac{(tIN \times V)}{2}(1)$$

Di mana:

S = Jarak antara sensor ultrasonik dengan objek yang terdeteksi.

V = perambatan cepat gelombang ultrasonik di udara (344 m/s)

t_{in} = Waktu tempuh gelombang ultrasonik

Sensor ultrasonik HC-SR04 memiliki tegangan operasional sebesar 5V dan arus yang kecil sebesar 15mA. Sensor ini dilengkapi dengan 2 input dan output, yaitu Echo dan Trigger. Pada proyek akhir ini, HC-SR04 digunakan sebagai sensor pendeteksi halangan di depan troli, sehingga troli dapat menghindari halangan dengan baik.

2.2.9. Webcam

Kamera Web atau Webcam adalah perangkat yang merekam objek sebagai gambar yang dibuat oleh sistem lensa. Elemen di dalam kamera memiliki ribuan titik piksel. Semakin banyak piksel, semakin tinggi resolusi gambar. Warna, intensitas warna, dan kecerahan warna gambar yang ditampilkan di kamera web dapat ditentukan berdasarkan piksel yang membentuk foto. Kamera jenis ini biasanya sudah disertakan didalam perangkat komputer. Sebagai bagian dari spesifikasi laptop tersebut, biasanya webcam terhubung ke komputer melalui port USB atau COM^[36]. Webcam pada tugas akhir ini berfungsi sebagai perangkat penangkap objek gambar yang nantinya akan digunakan untuk memproses gambar pada Raspberry Pi.



Gambar 2. 13 Webcam Logitech

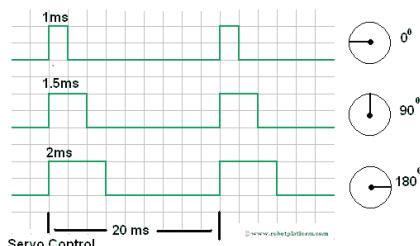
2.2.10.Motor Servo SG90

Motor servo adalah jenis motor yang telah didesain untuk mengontrol pergerakan melalui penggunaan kontrol loop tertutup. Tujuan dari motor ini adalah untuk memindahkan objek dengan sudut atau kecepatan yang ditentukan melalui PWM (Pulse Width Modulation). Komponen inti dari motor servo terdiri dari motor DC, driver, dan gearbox. Contoh yang populer dari motor servo adalah Motor Servo TowerPro SG90. Motor ini dilengkapi dengan roda gigi dan mampu menghasilkan torsi sebesar 2.5kg/cm, dengan kemampuan rotasi antara 0 hingga 180 derajat^[37].



Gambar 2. 14 Motor Servo SG90

Derajat Motor servo dapat diputar dari -90 derajat hingga 90 derajat dengan ketergantungan proporsional pada pulsa modulasi lebar (PWM) yang memicu itu. Gambar 2.15 menunjukkan modulasi pulsa dengan periode pulsa 20 ms atau 50 Hz dimana servo dapat berputar dari -90 derajat ke 90 derajat dengan mengubah duty cycle 1-2 ms dari pulsa tinggi. Motor servo pada penelitian ini digunakan sebagai penggerak pada kamera untuk mengikuti objek, sehingga pendekatan objek tidak menghilang secara tiba-tiba.



Gambar 2. 15 Pulse Width Modulation Period

2.2.11.Motor DC Power Window

Motor Power Window Gear 12 Volt adalah salah satu varian motor listrik DC yang menggunakan arus searah. Motor ini berfungsi untuk mengubah energi listrik menjadi energi mekanik dengan gesture berputar. Motor ini dilengkapi dengan angker dan memiliki satu atau lebih belitan individual. Setiap belitan dihubungkan dengan komutator, yang berperan sebagai cincin belah. Jika terdapat isolator di antara komutator, cincin pemisah dapat berfungsi sebagai saklar ganda bipolar.

Prinsip kerja Motor DC didasarkan pada hukum gaya Lorentz, yang menyatakan bahwa saat konduktor yang membawa arus ditempatkan dalam medan magnet, gaya akan timbul pada konduktor tersebut. Motor Power Window Gear menggunakan magnet ferrite yang memiliki sifat magnet permanen. Motor ini memiliki tegangan 12V, torsi 30kg, kecepatan beban 90rpm, dan arus sebesar 9 Ampere^[38]. Motor DC Power Window pada tugas akhir ini digunakan sebagai aktuator penggerak troli agar dapat berpindah.



Gambar 2. 16 Motor DC Power Window

2.2.12.Driver Motor BTS7960

Modul Driver motor BTS7960 adalah sebuah komponen yang mengatur arus tinggi pada motor dengan menggunakan H-bridge. Modul ini telah terintegrasi sepenuhnya untuk digunakan dalam berbagai aplikasi penggerak motor. Interaksi modul ini dengan mikrokontroler dipermudah oleh keberadaan IC pengendali yang terintegrasi, yang memiliki kemampuan menerima sinyal logika, mengatur kecepatan perubahan tegangan, menentukan waktu mati, serta melindungi motor dari suhu berlebih, tegangan berlebih, tegangan rendah, arus berlebih, dan hubung singkat. Penggunaan BTS7960 sebagai pengendali motor PWM

arus tinggi dengan kapasitas mencapai 43A dan voltase input yang terproteksi antara 6 hingga 27V juga memberikan solusi yang ekonomis, dengan konsumsi daya yang sangat rendah^[39]. Motor driver pada tugas akhir ini digunakan sebagai media kontrol motor dc power window agar bergerak sesuai yang diinginkan.



Gambar 2. 17 Motor Driver BTS7960

2.2.13. Modul *Stepdown XL4015*

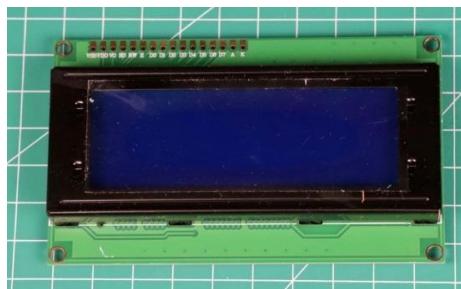
Modul stepdown adalah sebuah perangkat yang berperan dalam menurunkan tegangan dan memiliki kemampuan untuk mengatur outputnya menggunakan potensiometer. Terdapat beragam jenis modul stepdown yang memiliki fungsi-fungsi unik masing-masing. Dalam penelitian ini, digunakan modul stepdown dengan model XL4015. Modul XL4015 adalah konverter DC ke DC yang tetap beroperasi pada frekuensi 180kHz. Modul ini memiliki kapasitas hingga 5A untuk menyalakan beban, dengan rentang tegangan input antara 6 hingga 38V, serta rentang tegangan output antara 1.25 hingga 36V. Selain memiliki efisiensi yang tinggi dan riak yang rendah, modul ini juga menunjukkan performa yang sangat baik dalam mengatur saluran dan beban. Fitur-fitur lain yang dimiliki oleh modul ini termasuk pengaturan tegangan konstan dan arus konstan^[40]. Modul stepdown pada tugas akhir ini digunakan sebagai penurun tegangan dari 12V menjadi 5V untuk supply arus pada Raspberry Pi.



Gambar 2. 18 Modul Stepdown XL4015

2.2.14. LCD (Liquid Crystal Display) I2C 20x4

Layar LCD, atau Liquid Crystal Display, merupakan suatu perangkat yang menggunakan kristal cair sebagai komponen utamanya untuk menampilkan informasi. Penggunaan layar LCD sangat luas, digunakan dalam berbagai aplikasi seperti televisi elektronik, kalkulator, dan layar komputer. Kristal cair dalam layar LCD memungkinkan tampilan gambar dan karakter karena piksel-pikselnya dapat dikendalikan oleh kristal cair. Meskipun disebut "titik cahaya," kristal cair sendiri sebenarnya tidak memancarkan cahaya. Layar LCD ini beroperasi pada tegangan 5V dan memiliki format tampilan 20x4 dengan lampu latar berwarna biru^[41]. Pada tugas akhir ini LCD digunakan sebagai penampil geraknya nilai sudut dan perintah troli ketika berjalan.



Gambar 2. 19 LCD I2C 20x4

2.2.15. Sensor Gyroscope MPU6050

MPU6050 merupakan sebuah sensor Inertial Measurement Unit (IMU) yang terdiri dari tiga akselerometer dan tiga giroskop yang digabungkan dalam satu chip. Sensor MPU6050 memiliki kemampuan

untuk mengukur percepatan dan kecepatan sudut dari suatu objek dalam tiga dimensi. Selain itu, sensor ini dapat digunakan untuk mendeteksi gesture, rotasi, dan orientasi perangkat. Dengan adanya accelerometer dan gyroscope, sensor ini dapat menghasilkan pembacaan sudut kemiringan objek dalam tiga dimensi^[42].

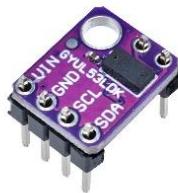
Dalam usaha untuk menjaga keseimbangan sensor robot, algoritma PID dapat digunakan untuk mengendalikan sinyal komponen Kp dan Kd secara real-time, yang kemudian dikirim ke mikrokontroller. Saat menguji sensor MPU6050 pada permukaan datar, sumbu z menunjukkan stabilitas antara 1,47 dan -1,24, sementara sumbu x dan y mengalami pergerakan naik dan turun pada sudut kemiringan 60° dan -60°. Dalam penelitian ini, sensor IMU digunakan untuk memuluskan gesture troli agar tetap bergerak lurus di jalurnya tanpa keluar.



Gambar 2. 20 Modul Sensor MPU6050

2.2.16. Sensor Laser ToF VL53L0X

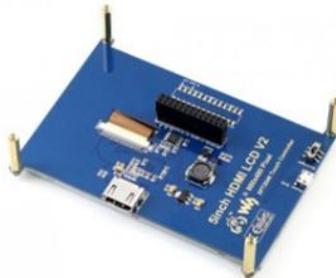
Sensor Laser VL53L0X adalah modul rentang laser Time-of-Flight (ToF) yang ditempatkan dalam paket terkecil di pasaran saat ini, memberikan pengukuran jarak akurat apa pun pantulan targetnya, tidak seperti teknologi konvensional. Sensor ini dapat mengukur jarak absolut hingga 2 m dengan kecepatan maksimum 50 Hz. VL53L0X mengintegrasikan array SPAD terdepan (single photon avalanche diodes) dan menyematkan teknologi generasi kedua yang dipatenkan FlightSense. Memiliki tegangan input 5V dengan media komunikasi I²C bus^[43]. Pada tugas akhir ini sensor VL53L0X digunakan sebagai pendekripsi pengguna pada troli, sehingga ketika troli diperintah mundur troli tidak akan menabrak pengguna.



Gambar 2. 21 Sensor Laser Time Of Flight VL53L0X

2.2.17. LCD Raspberry 5 Inch

LCD Raspberry 5 inch (atau sering juga disebut LCD HDMI 5 inch) adalah sebuah layar tampilan berukuran 5 inci yang dirancang khusus untuk digunakan dengan Raspberry Pi, sebuah papan komputer mini yang banyak digunakan untuk berbagai proyek elektronik dan komputasi. LCD Raspberry 5 inch biasanya dilengkapi dengan resolusi 800x480 piksel dan dapat menampilkan gambar dalam format 4:3 atau 16:9 dengan tegangan input sebesar 5V^[44]. Pada tugas akhir ini LCD Raspberry 5 Inch digunakan sebagai penampil deteksi *hand gesture* dari raspberry pi.



Gambar 2. 22 LCD Raspberry 5 Inch

2.2.16 Baterai Aki

Baterai aki atau akumulator, juga dikenal sebagai elemen sekunder, adalah sumber daya arus searah yang dapat mengubah energi kimia menjadi energi listrik. Baterai ini dianggap sebagai unsur sekunder karena berisi bahan elektrokimia yang mempengaruhi reaksi kimia di dalamnya. Terminal positif baterai menggunakan pelat oksida, sementara

terminal negatif menggunakan pelat timah, dan elektrolitnya terdiri dari larutan asam sulfat^[45].

Dalam penelitian ini, menggunakan baterai aki kering sebagai sumber listrik. Baterai aki kering ini menggunakan kalsium sebagai bahan anoda dan katoda. Keunggulan baterai sel kering adalah tidak memerlukan perawatan dan memiliki kinerja baterai yang lebih baik dibandingkan dengan jenis baterai lainnya. Namun, baterai kering memiliki keterbatasan dalam menahan suhu tinggi dan harganya relatif mahal. Baterai yang digunakan dalam penelitian memiliki kapasitas 9Ah dengan tegangan 12 VDC.



Gambar 2. 23 Baterai Aki SMT-Power

BAB III

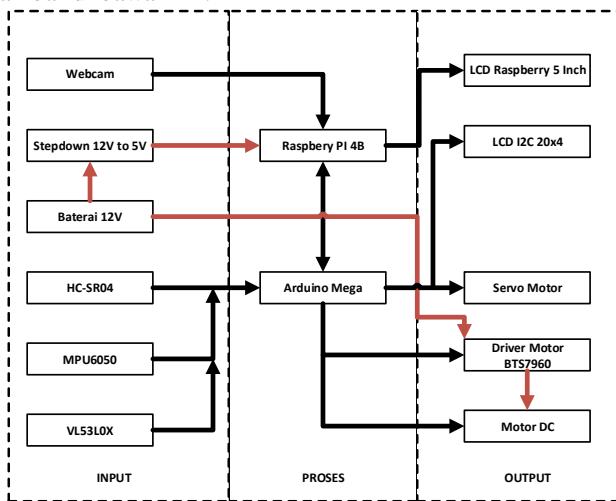
PERANCANGAN SISTEM

Perencanaan sistem tugas akhir dilakukan sebagai langkah awal, sebelum alat siap diimplementasikan, untuk memastikan sistem bekerja sesuai fungsinya. Pertama, struktur perangkat, seperti badan perangkat, dirancang dan lokasi penempatan komponen yang digunakan. Selain itu juga agar dapat mengetahui prinsip-prinsip fungsional dan karakteristik dari masing-masing komponen yang digunakan. Rincian desain proyek akhir dijelaskan di bawah ini.

3.1 Perancangan Troli Otomatis

3.1.1. Blok Diagram

Pada proses kerja suatu alat, langkah pertama adalah membuat diagram blok yang berfungsi sebagai gambaran umum dari tugas akhir yang akan dilakukan. Diagram blok ini sangat penting dalam perancangan alat karena memberikan gambaran prinsip kerja rangkaian secara keseluruhan. Keberadaan diagram blok memudahkan perancangan alat agar sistem yang dikembangkan sesuai dengan rencana yang telah ditetapkan sebelumnya. Anda dapat melihat diagram blok sistem alat ini pada gambar di bawah ini.



Gambar 3. 1 Diagram Blok Sistem

Pada tahap awal sistem akan bekerja ketika tegangan *input* baterai aki 12VDC masuk pada modul stepdown XL4015, Motor Driver BTS7960. Tegangan 12VDC pada modul stepdown akan diturunkan untuk memberi *supply* pada Raspberry Pi untuk menghidupkan arduino dan kamera webcam sedangkan arduino difungsikan untuk mengaliri *supply* tegangan seluruh perangkat input seperti sensor dan mpu6050. Ketika semua tegangan sudah disupport pada semua perangkat sensor dan LCD, selanjutnya Raspberry Pi akan memulai *booting* perangkat sedangkan arduino menunggu kirimnya perintah dari raspberry. Ketika raspberry sudah berhasil *booting* dengan tanda adanya *display* pada LCD Raspberry. Raspberry akan memulai program python secara otomatis untuk memulainya sistem, ketika program sudah berjalan akan muncul *display* kamera webcam. dan raspberry akan mengirimkan perintah menuju arduino mega yang kemudian dilakukan proses kontrol. MPU6050 akan memberikan nilai posisi dari robot pada arduino, sensor HC-SR04 akan memperingati apabila robot menemui halangan. Ketika dua proses tersebut berjalan, arduino akan mengirimkan perintah pada motor servo untuk mengikuti gesture tangan, motor driver akan menggerakkan robot sesuai perintah, dan LCD I2C untuk menampilkan nilai sensor yang diproses pada Arduino Mega.

3.1.2. Kebutuhan Perangkat Keras

Setelah menetapkan blok diagram sistem di atas, langkah selanjutnya adalah memberikan penjelasan dan menjelaskan kebutuhan tiap komponen agar sesuai dengan sistem yang akan dibuat. Di bawah ini terdapat tabel yang berisi kebutuhan perangkat keras yang akan digunakan dalam perancangan sistem.

Tabel 3. 1 Kebutuhan Komponen Sistem

No.	Nama Komponen	Jumlah	Fungsi
1.	Arduino Mega 2560	1 Pcs	Sebagai otak kontrol sistem yang difungsikan dalam pemroses gerak robot.
2.	Raspberry Pi 4B	1 Pcs	Sebagai komputer dalam menjalankan <i>image processing</i>

No.	Nama Komponen	Jumlah	Fungsi
3.	Sensor ToF VL53L0X	1Pcs	Sebagai pendekripsi jarak pengguna dibelakang troli
4.	Sensor Ultrasonik HC-SR04	3 Pcs	Sebagai pendekripsi halangan jika troli akan menabrak.
5.	<i>Webcam Logitech C270</i>	1 Pcs	Sebagai sensor pendekripsi citra gesture tangan pengguna.
6.	LCD I2C 4x20	1 Pcs	Sebagai penampil informasi nilai sensor, pendekripsi, dan nilai posisi robot
7.	Motor <i>Power Window</i>	2 Pcs	Sebagai aktuator dan penggerak robot agar robot dapat berpindah tempat.
8.	Motor Servo SG90	1 Pcs	Sebagai penggerak kamera agar kamera tetap mengikuti tangan target
9.	Modul <i>Stepdown XL4015</i>	1 Pcs	Sebagai konverter tegangan dari 12V menjadi 5V
10.	Baterai Aki	1 Pcs	Sebagai <i>supply</i> tegangan utama pada sistem robot
11.	<i>Driver Motor</i> BTS7960	2 Pcs	Sebagai pengatur dan kontrol motor power window
12.	LCD 5 Inch HDMI	1 Pcs	Sebagai penampil output <i>display raspberry pi</i>
13.	MPU6050	1 Pcs	Sebagai sensor posisi dan gerak lurus robot

3.1.3. Komponen Perangkat Lunak

Pada perancangan alat robot troli ini dibutuhkan juga beberapa perangkat lunak atau *software* untuk menunjang sistem dapat berjalan. Beberapa diantara perangkat lunak tersebut sangat dibutuhkan untuk

melakukan programming sistem robot. Diantaranya ada pada daftar Tabel 3.2.

Tabel 3. 2 Komponen Perangkat Lunak

No.	Perangkat Lunak	Fungsi
1.	Raspbian Bullseye	Sistem operasi yang dibuat oleh Raspberry Pi secara open source
3.	Arduino IDE	Dipakai untuk melakukan pemrograman kontrol pada arduino mega 2560
4.	VSCODE	Dipakai untuk melakukan pemrograman citra digital menggunakan bahasa python.

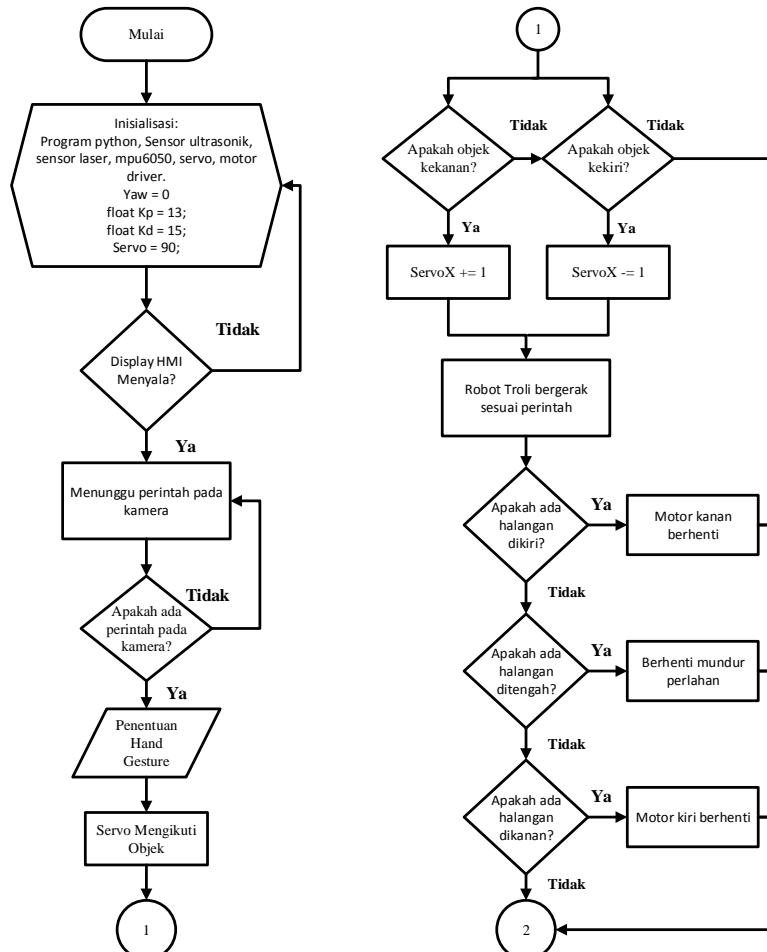
3.2 Diagram Alir

Flowchart atau diagram alir adalah cara umum untuk menggambarkan proses implementasi suatu sistem. Simbol-simbol digunakan untuk mewakili setiap langkah dalam sistem, sedangkan garis-garis dengan panah menunjukkan urutan langkah-langkah. Pada tahap ini, sebuah rencana sistem disusun yang mencakup input dan output. Input menggambarkan data yang akan diproses, sedangkan output menggambarkan informasi yang dihasilkan.. Berdasarkan dari Gambar 3.2 dan 3.3 *flowchart* merupakan gambaran umum prinsip kerja sistem.

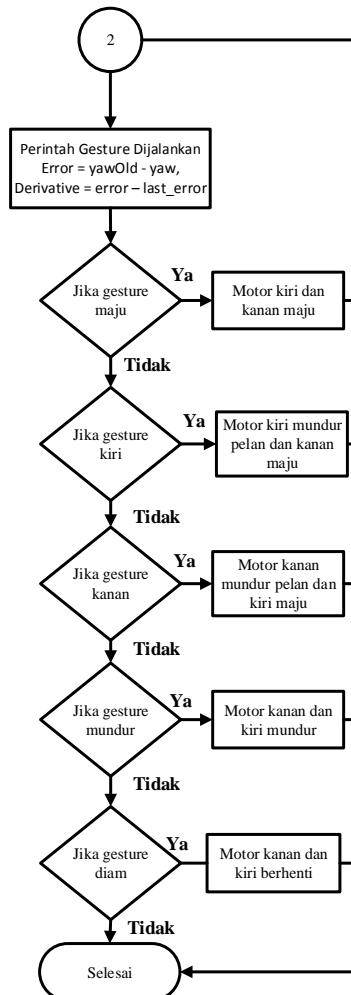
Pada tahap awal sistem bekerja dengan menginisialisasi sensor, mpu6050, menghidupkan lcd dan raspberry melakukan booting. Kemudian sistem akan melakukan pendekripsi tangan apabila lcd raspberry sudah menampilkan *display*. Raspberry akan otomatis memulai program python yang sudah diprogram sebelumnya dan siap untuk menerima *input*. Ketika kamera sudah menemukan *input* berupa deteksi tangan, servo akan mengunci kamera tepat di titik tengah kotak deteksi dan mengikuti arah gerak kotak deteksi tangan tersebut. Ketika robot berbelok, robot akan mencatat posisi yaw terakhir hingga ia berhenti. Ketika robot bergerak maju maka yaw terakhir tersebut akan digunakan untuk setpoint agar robot dapat bergerak maju secara lurus.

Ketika robot terhalang sesuatu didepan robot yang dideteksi oleh sensor ultrasonik, maka robot akan berusaha menghindari halangan hingga tercapai kondisi yang sesuai. Ketika ultrasonik bagian depan mendekripsi halangan maka robot akan mundur. Jika sensor ultrasonik

bagian kanan mendeteksi adanya benda maka motor kiri akan dihentikan sehingga troli dapat berbelok ke kiri. Ketika ultrasonik kiri mendeteksi halangan maka motor kanan akan dihentikan sehingga troli berbelok ke kiri.



Gambar 3. 2 Flowchart Sistem I

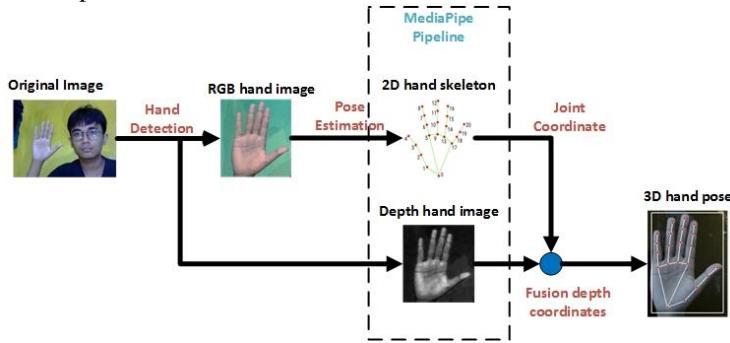


Gambar 3. 3 Flowchart Sistem II

3.3 Arsitektur Sistem *Hand Gesture Recognition*

Pada tahap awal pendekripsi tangan, mula-mula memahami struktur cara kerja dari penggunaan *library mediapipe*, dengan mengetahui dasar-dasar mediapipe. Pembuatan sistem robot troli akan

lebih mudah, maka dari itu Gambar 3.4 merupakan gambar sistem kerja dari mediapipe pipeline. Mediapipe ini tersedia pada pemrograman machine learning yang dikembangkan oleh Google Inc. Berisi kumpulan dataset 2D Hand Pose dan *Depth Map* yang telah dilakukan *training* untuk mendapatkan keakuratan *fusion* antara RGB deteksi tangan dengan 2D hand pose.



Gambar 3. 4 Prinsip Kerja Mediapipe

Dari Gambar 3.4 diatas terlihat bahwa untuk mendapatkan pendekripsi hand landmark diperlukan perubahan dari image *real-time* biasa berupa BGR menjadi image yang memiliki warna RGB, dengan tujuan agar terlihat perbedaan warna antara warna lengan dengan warna latar yang ada. Ketika gambar sudah diubah menjadi RGB, RGB tersebut akan diubah kembali menjadi Depth Image berupa kedalaman dan ketebalan objek berupa 3D hand yang belum digabung dengan hand landmark. Depth image map dan 2D hand skeleton ini sudah dilakukan training berupa model tensorflow lite oleh mediapipe, sehingga output yang didapatkan berupa model 3D hand pose yang memiliki hand landmark.

```

def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    if self.results.multi_hand_landmarks:
        for handLms in self.results.multi_hand_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, handLms,
                                           self.mpHands.HAND_CONNECTIONS)
    return img
  
```

Gambar 3. 5 Penggunaan Modul Mediapipe

Pada Gambar 3.5 merupakan cara mendapat pendekslan hand landmark menggunakan mediapipe, dengan tahap pertama mengubah image terlebih dahulu menjadi image berwarna RGB. Kemudian hasil dari imgRGB tersebut diakses menjadi self.result yang kemudian digabungkan pada multi_hand_landmark yang merupakan 2D Hand pose. Selanjutnya self.result tersebut digambar dengan pertintah self.mpDraw untuk membuat titik pada sendi.

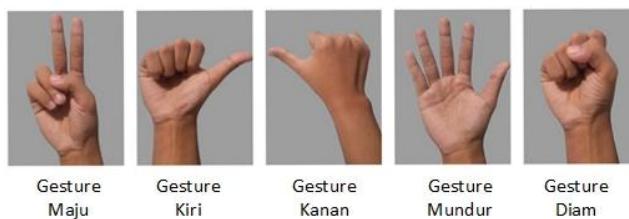
Self.mpHand.Hand_Connection akan menyambungkan tiap titik sendi menjadi sebuah hand landmark skeleton. Gambar 3.6 menunjukkan titik-titik koordinat sendi ruas jari.



Gambar 3. 6 3D Hand Landmark Pose

3.3.1. Perancangan *Gesture* Gerak Troli

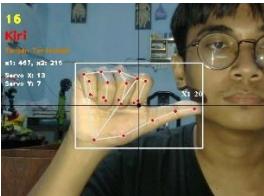
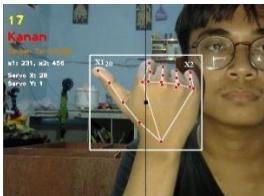
Perintah untuk pergesiture troli telah diatur pada pemroggraman python untuk mendeteksi pola tertentu pada gerak tangan, beberapa diantaranya menggunakan titik region of interest pada sendi hand landmark. Pada pemrograman robot troli ini tidak menggunakan training model keras, namun dapat mengenali pola tertentu tanpa menggunakan sebuah training dataset. Beberapa pola yang telah diatur ditampilkan pada Gambar 3.7.



Gambar 3. 7 *Gesture* Tangan

Gesture tangan tersebut telah diprogram pada pemrograman python di raspberry pi, pengenalan tersebut berdasarkan titik sendi hand landmark. menggunakan program if else didapatkan keakuratan pengenalan pola. Dengan langkah pengenalan pola yang ditampilkan pada tabel 3.3 antara lain sebagai berikut:

Tabel 3. 3 Langkah Pengenalan Pola

Pola Gesture Tangan	Langkah Pengenalan
Pola Maju 	Pola Maju menggunakan pengenalan koordinat <code>index_tip(8) && middle_tip(12) > thumb_tip(4) && ring_tip(8) && pinky_tip(20)</code>
Pola Kiri 	Pola Kiri menggunakan pengenalan koordinat <code>thumb_tip(4/X1) > pinky_tip(20/X2)</code>
Pola Kanan 	Pola Kanan menggunakan pengenalan koordinat <code>thumb_tip(4/X1) < pinky_tip(20/X2)</code>

Pola Gesture Tangan	Langkah Pengenalan
Pola Mundur 	Pola Mundur menggunakan pengenalan koordinat <code>index_tip(8) && middle_tip(12) && thumb_tip(4) && ring_tip(8) && pinky_tip(20) > wrist(0)</code>
Pola Diam 	Pola Diam menggunakan pengenalan koordinat <code>index_tip(8) && middle_tip(12) && thumb_tip(4) && ring_tip(8) && pinky_tip(20) = 0</code>

3.4 Perancangan Robot Troli

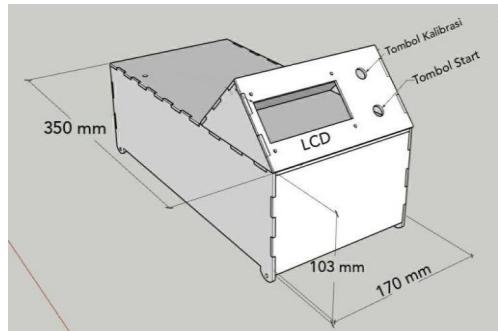
3.4.1. Perancangan Mekanika Robot Troli

Perancangan mekanik mencakup desain kerangka alat yang terbuat dari besi, akrilik, dan papan triplek. Bahan besi dipilih untuk membuat troli karena kekuatannya yang baik sehingga dapat menopang beban alat dengan baik dan tahan lama. Ukuran troli yang digunakan adalah 22 liter dan memiliki 4 roda sebagai penggeraknya. Akrilik dengan ketebalan 3mm dipilih sebagai bahan pembuatan box untuk komponen dan perangkat elektronik.

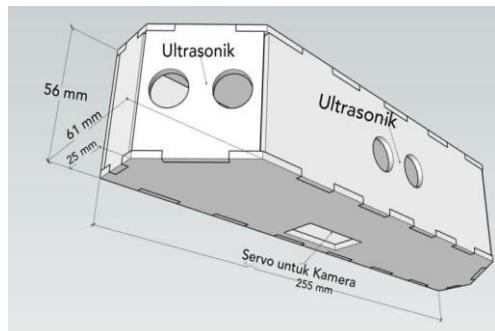
Keuntungan menggunakan akrilik adalah mudah dibentuk dan tahan terhadap air sehingga dapat melindungi komponen elektronik dengan baik. Ada 2 buah box komponen yang dibuat, yaitu box 1 untuk peletakan komponen pada Gambar 3.9 dan box 2 untuk sensor ultrasonik dan webcam pada Gambar 3.10. Sedangkan papan triplek digunakan sebagai bagian bawah dari box peletakan komponen untuk menahan beban yang cukup berat. Box komponen kemudian diletakkan di bagian bawah troli untuk meningkatkan efisiensi alat. Bentuk dari robot troli yang telah dibuat ada pada Gambar 3.8.



Gambar 3. 8 Perancangan Mekanika Troli



Gambar 3. 9 Komponen Box



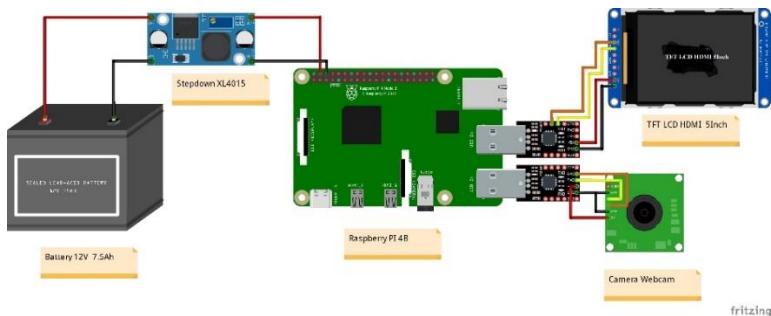
Gambar 3. 10 Komponen Box Sensor Ultrasonik

3.5 Perancangan Rangkaian Listrik Robot Troli

Perancangan rangkaian listrik robot troli ini mengacu pada gambaran keseluruhan mengenai cara menghubungkan setiap komponen elektronik pada alat yang direncanakan melalui sistem pengkabelan yang telah dirancang sebelumnya.

3.5.1. Rangkaian Raspberry Pi 4B

Pada Gambar 3.11 dapat dilihat perancangan rangkaian elektronik yang menggunakan komponen mikrokomputer Raspberry Pi 4B, MPU6050, dan Motor Servo SG90.



Gambar 3. 11 Rangkaian Pada Raspberry PI 4B

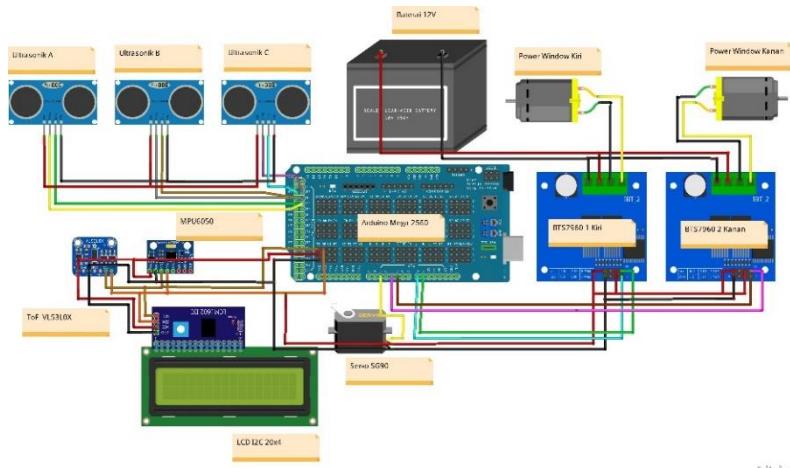
Rangkaian elektronik ini dirancang untuk mengetahui komunikasi antara komponen sensor kamera dan mini pc Raspberry Pi 4B. Komunikasi antara kedua komponen tersebut dilakukan melalui koneksi serial UART yang menggunakan kabel USB. Pada tahap awal baterai 12V akan menurunkan tegangan menjadi 5V dengan menggunakan modul stepdown XL4015 untuk *supply* tegangan raspberry pi, jika raspberry pi sudah mendapatkan *supply* tegangan, LCD TFT akan menyala karena adanya *supply* via USB. LCD menyala yang menandakan raspberry sudah siap berjalan. Konfigurasi koneksi antara Raspberry Pi 4B ditampilkan ada Tabel 3.4.

Tabel 3. 4 Pengaturan Input Raspberry Pi 4B

Baterai 12V	Modul Stepdown	Raspberry Pi 4B	Sensor Kamera	LCD TFT 5Inch
VCC	Vin +			
GND	Vin -			
		USB Type A	USB Type B	
		USB Type A		USB Type B
	Vout +	5 V		
	Vout -	GND		

3.5.2. Rangkaian Arduino Mega 2560

Rangkaian elektrik pada Arduino Mega 2560 dirancang menggunakan perangkat lunak Fritzing untuk memvisualisasikan komponen dan pengkabelan antara komponen. Gambar 3.12 menampilkan rangkaian dengan komponen dan jalur kabel yang terhubung satu sama lain.



Gambar 3. 12 Rangkaian Arduino Mega 2560 Pada Robot

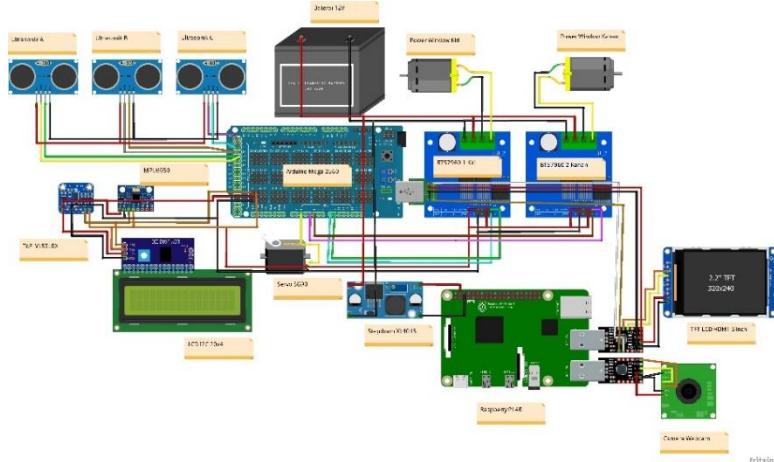
Tujuan dari membuat rangkaian elektronik tersebut adalah untuk mempelajari cara menghubungkan beberapa komponen, yaitu driver motor BTS7960, MPU6050 Gyroscope dan Accelerometer, Motor Power Window, Motor Servo, Sensor Laser ToF VL53L0X, Sensor Ultrasonik HC-SR04, LCD 20x4 dengan I2C. Disertakan tabel input antar komponen dapat ditemukan pada Tabel 3.3.

Tabel 3. 5 Tabel Pengaturan Komponen Arduino

Baterai 12 V	Arduino Mega 2560	Sensor Ultrasonik HC-SR04	BTS7960	Motor Servo SG90	Motor DC Power	LCD I2C 20x4	MPU6050	VL53L0X
	Pin 2			PWM				
	Pin 4		LPWM1					
	Pin 5		RPWM1					
	Pin 8		LPWM2					
	Pin 9		RPWM2					
	Pin 46	Trig A						
	Pin 47	Echo A						
	Pin 48	Trig B						
	Pin 49	Echo B						
	Pin 50	Trig C						
	Pin 51	Echo C						
	SDA					SDA	SDA	SDA
	SCL					SCL	SCL	SCL
	+5V	VCC	VCC	VCC		5V	5V	5V
	GND	GND	GND	GND		GND	GND	GND
			M1+		M1+			
			M2+		M2+			
			M1-		M1-			
			M2-		M2-			
VCC			Bat +					
GND			Bat -					

3.5.3. Rangkaian Keseluruhan Sistem Robot Troli

Rangkaian Keseluruhan Sistem Robot Troli dapat dilihat pada Gambar 3.13.



Gambar 3. 13 Rangkaian Keseluruhan Sistem

Rangkaian elektronik pada robot troli ini dirancang untuk menghubungkan dan berkomunikasi antara mini pc Raspberry Pi 4B dan mikrokontroler Arduino Mega. Kedua komponen tersebut saling terhubung dan berkomunikasi melalui koneksi serial UART yang menggunakan kabel USB sebagai media pengirim data. Konfigurasi koneksi antara Raspberry Pi 4B dan Arduino Mega ada pada Tabel 3.6.

Tabel 3. 6 Konfigurasi Komunikasi Arduino Raspberry

Baterai 12V	Supply 5V	Raspberry Pi 4B	Arduino Mega 2560
VCC	Vin +		
GND	Vin -		
	Vout +	+5V	
	Vout -	GND	
		USB Type C	USB Type A

3.6 Sistematika Pemrograman Arduino

3.6.1 Sistematika Pemrograman Motor Servo

Arduino Mega 2560 digunakan dalam pemrograman motor servo dengan menggunakan software Arduino IDE. Motor servo ini akan difungsikan sebagai pergesture kamera saat mendekripsi objek tangan. Fokus dari program ini adalah mengaktifkan servo untuk melacak pergesture objek tangan dan menyesuaikan posisi kamera. Berikut adalah program yang dihasilkan:

```

void pantilt() {
    if (xcenter != 404) {
        if (xcenter < 300 && xcenter > 1) {
            com = '1';
        }
        else if (xcenter > -300 && xcenter < -1) {
            com = '2';
        }
        switch (com) {
            case '1':
                a();
                delay(10);
                break;
            case '2':
                b();
                delay(10);
                break;
            default:
                servox.write(90);
                break;
        }
    }
    else {
        com = '0';
    }
}
void a() {
    servox.write(x);
    x -= 1;
}
void b() {
    servox.write(x);
    x += 1;
}

```

Gambar 3. 14 Program Gerak Servo Mengikuti Objek

Dari pemrograman diatas terdapat default servoX.write(90) yang berarti sudut posisi kamera awal terletak pada 90° servo, ketika objek mendekati koordinat $> -300 \& < -1$ yang artinya objek sedang berada pada posisi kiri, sehingga diperlukan perubahan sudut servo, maka arduino akan menambah 1° sudut hingga posisi sesuai, sedangkan ketika objek melewati $< 300 \& > 1$ yang artinya objek sedang berada pada posisi kanan sehingga diperlukan perubahan sudut servo, , maka arduino akan mengurangi 1° sudut hingga posisi sesuai.

3.6.2. Sistematika Pemrograman Sensor Ultrasonik dan Sensor VL53L0X

Program yang dibuat menggunakan Arduino IDE bertujuan untuk mengendalikan sensor ultrasonik yang terhubung dengan Arduino Mega. Tujuan utamanya adalah untuk mengukur jarak maksimum antara sensor ultrasonik dengan objek yang terdeteksi.

```

void gerak() {
    if (usB <= 15) {
        waithere();
        for (int c = 0; c < 100; c++) {
            delay(1);
        }
        backward();
        pwmmmin = 70;
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
    else if (ul <= 20) {
        pwm1 = 30;
        pwm2 = 30;
        forward();
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
    else if (usc <= 15) {
        draw = 0;
        pwmmmin = 50;
        left();
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
    else if (usa <= 15) {
        draw = 0;
        pwmmmin = 50;
        right();
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
    else if (usa <= 15 && usc <=15) {
        waithere();
        for (int c = 0; c < 100; c++) {
            delay(1);
        }
        backward();
        pwmmmin = 70;
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
    else {
        flow();
    }
}

```

Gambar 3. 15 Pemrograman Sesnor Ultrasonik

Dalam pemrograman, aturan diatur sebagai berikut: jika jarak sensor ultrasonik A kurang dari 20 cm, motor akan berbelok ke kanan. Jika jarak sensor ultrasonik B kurang dari 20 cm, troli akan berhenti dan mundur perlahan sampai kondisi tidak terpenuhi. Jika jarak sensor ultrasonik C kurang dari 20 cm, troli akan berbelok ke kiri. Jika troli mundur dan hampir menabrak pengguna dengan jarak 20 cm, troli akan mendeteksi dan berhenti, lalu maju perlahan sampai kondisi tidak terpenuhi.

3.6.3. Sistematika Pemrograman *Driver Motor BTS 7960*

Untuk mengendalikan kecepatan dan arah gerak motor DC, diperlukan pemrograman pada driver motor BTS7960 yang terhubung dengan Arduino Mega. Pemrograman ini dapat dilakukan menggunakan

Arduino IDE. Berikut ini adalah contoh program yang dapat digunakan untuk mengatur motor driver tersebut.

```

void flow() {
    if (kondisi != 404) {
        if (kondisi == 77) {
            stss = "Berhenti ";
            waithere();
        }
        if (kondisi == 66) {
            stss = "Mundur ";
            backward();
            pwmmmin += 3;
            if (pwmmmin > pwmmax) pwmmmin = pwmmax;
        }
        if (kondisi == 55) {
            stss = "Kanan ";
            right();
            target_yaw = yawold;
            pwmmmin += 1;
            draw -= 1;
            if (draw < 5) draw = 5;
            if (pwmmmin > pwmmax) pwmmmin = pwmmax;
        }
        if (kondisi == 33) {
            stss = "Maju ";
            error = yawOld - yaw; // proportional
            derivative = error - last_error; //derivative
            angle = (error * kp) + (derivative * kd);
            target_yaw = yaw;
            if (target_yaw > yawOld) {
                pm1 = mtrspd - abs(angle); //kiri
                pm2 = mtrspd + abs(angle); //kanan
                forward();
            } //setting the steering command if it is veering to the left
            else if (target_yaw < yawOld) {
                pm1 = mtrspd + abs(angle); //kiri
                pm2 = mtrspd - abs(angle); //kanan
                forward();
            }
            error = last_error;
        }
        if (kondisi == 44) {
            stss = "kiri ";
            left();
            target_yaw = yawOld;
            pmmin += 1;
            Draw -= 1;
            if (draw < 5) draw = 5;
            if (pmmin > pwmmax) pmmin = pwmmax;
        }
    }
}

```

Gambar 3. 16 Pemrograman Driver Motor Untuk Gerak Troli

Program diatas terlihat apabila data yang telah dikirim pada raspberry dan diterima oleh arduino, maka arduino langsung mengeksekusi perintah yang dikirim dari raspberry tersebut.

3.6.4. Sistematika Pemrograman LCD 20x4 I2C

Pemrograman dilakukan menggunakan Arduino IDE pada LCD I2C yang terhubung dengan Arduino Mega. Fungsi dari LCD I2C adalah menampilkan informasi berupa teks dari beberapa sumber seperti webcam, sensor ultrasonik, dan PWM motor. Program untuk mengontrol LCD I2C dapat dilihat pada kode program di bawah ini.

```

lcd.setCursor(0, 0);
lcd.print("JSON:");
lcd.print("{\""+string(xcenter) + "," + string(ycenter) + "," + string(kondisi) + \"}");

lcd.setCursor(0, 1);
lcd.print("Angle : ");
lcd.print(angle);

lcd.setCursor(0, 2);
lcd.print("kondisi : ");
lcd.print(stss);

lcd.setCursor(0, 3);
String us = ("{" + String(usA) + ',' + String(usB) + ',' + String(usC) + ',' + String(usI) + '}');
lcd.print("jarak:");
lcd.print(us);

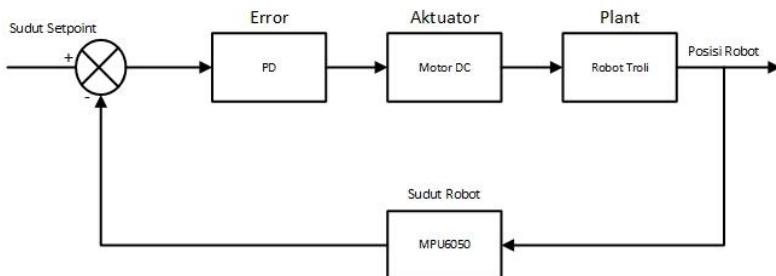
```

Gambar 3. 17 Sistematika Program LCD

Pad baris LCD pertama kolom (0,0) ditampilkan informasi kiriman dari raspberry pi, berupa titik center objek dan kondisi perintah yang dilakukan, kedua kolom (0,1) ditampilkan informasi posisi sudut robot, ketiga kolom (0,2) ditampilkan informasi kondisi robot sekarang maju, berbelok atau berhenti, keempat kolom (0,3) ditampilkan informasi seluruh sensor ultrasonik dan sensor laser.

3.6.5. Sistematika Sistem Kendali Pemrograman MPU6050

Robot troli menggunakan prinsip Proporsional dan Derivative tanpa menggunakan integral. Dengan menggunakan sistem kendali Gambar 3.18 dibawah ini:



Gambar 3. 18 Diagram Blok Sistem Kendali PD

Pada Gambar 3.18 ketika robot akan berjalan, awal mula arduino akan menetapkan variabel setpoint target, selanjutnya dilakukan pemrosesan kendali sistem PD untuk menjalankan aktuator yakni motor dc sehingga motor dc akan bergerak lurus, namun ketika robot terjadi error sehingga berpindah jalur dari sudut setpoint maka sensor mpu6050 akan mengetahui sudut robot saat ini untuk dilakukan perbaikan dengan menggunakan algoritma PD robot. Sehingga robot akan tetap ada pada titik setpoint, untuk lebih jelasnya ada pada prinsip sesuai rumus dibawah:

P – Proporsional

I – Integral

D – Derivative

$$(Proporsional * K_p) + (Integral * K_i) + (Derivative * K_d) = Angle Steering Output$$

PID adalah singkatan dari proporsional, integral, dan diferensial. Integral dan diferensial adalah istilah kalkulus, tetapi mengingat bahwa kalkulus adalah studi matematika tentang perubahan yang kontinu, hal ini hampir sempurna sesuai dengan kebutuhan. PID adalah algoritma kontrol, sehingga bisa digunakan untuk segala hal mulai dari pengontrol sensor gyro hingga pengontrol jalur pengikut. Pada akhirnya, sudut akan menjadi kesalahan dikali dengan K_p ditambah dengan integral dikalikan dengan K_i ditambah dengan diferensial dikalikan dengan K_d.

Kontrol proporsional

Variabel Deklarasi:

Target = 0 (setpoint awal)

Error = 0

Kp = 11 (Nilai Arbitrary)

Angle = 0

Loop Section:

Error = yawOld - yaw

*(Error * Kp) = angle*

Kontrol proporsional adalah tahap koreksi. Caranya direpresentasikan dalam kode adalah kesalahannya sama dengan nilai gyro target yang dalam banyak kasus adalah 0, dikurangi dengan pembacaan gyro. Kemudian error dikalikan dengan K_p untuk mendapatkan hasil koreksi yang nantinya akan kita tambahkan integral dan turunannya. Namun robot troli ini tidak menggunakan integral karena akan memiliki baca waktu yang lama dan error yang semakin besar sehingga program langsung menggunakan derivative.

Kontrol derivative

Variabel Deklarasi:

derivative = 0

last_error = 0

Kd = 13 (Nilai Arbitrary)

Loop Section:

derivative = error - last_error

*(Error * Kp) + (derivative * Kd) = angle*

Derivatif membandingkan kesalahan saat ini, dengan kesalahan dari waktu sebelumnya loop berputar. Selanjutnya dengan mengukur kesalahan dan menyimpan nilai kesalahan di akhir loop sebagai variabel yang disebut kesalahan terakhir. Sekarang kali berikutnya loop muncul, dapat secara efektif membandingkan kesalahan dari bagian proporsional dengan kesalahan dari kesalahan terakhir. Ini kemudian disimpan dalam variabel turunan dan dikalikan dengan Kd dan sekali lagi, menambahkannya dengan proporsional untuk mendapatkan keluaran sudut akhir. Perhatikan juga bahwa semua variabel yang gunakan untuk pengontrol PID adalah float, bukan bilangan bulat, karena memerlukan nilai yang sangat tepat untuk mendapatkan drive yang mulus dan lurus. Sekarang benar-benar dapat menggunakan output untuk mengendalikan motor robot troli. Implementasi tersebut ditampilkan pada Gambar 3.21.

```

//MPU6050
float target = 0;
float yawOld = 0;
float derivative = 0;
float last_error = 0;

//the 'K'
float kp = 2;
float kd = 3;
float pwm1 = 0;
float pwm2 = 0;

int mtrsPd = 45.5;
float angle = 0;

void calcYaw() {
    unsigned long currentTime = millis();
    dt = (currentTime - lastTime) / 1000.0;
    lastTime = currentTime;

    gyroZ = getRotationZ();

    float angularZ = (gyroZ - gyroZ0) / 131.0 * dt; //angular z: +=

    if (fabs(angularZ) < 0.05) {
        angularZ = 0.0;
    }
    gyroAngleZ += angularZ; //returns the absolute value of the z-axis rot;
    yaw = -gyroAngleZ;
}

if (kondisi == 33) {
    stss = "Maju ";
}

error = yawOld - yaw; //proportional
derivative = error - last_error; //derivative
angle = (error * kp) + (derivative * kd);

target_yaw = yaw;
if (target_yaw > yawOld) {
    pwm1 = mtrsPd - abs(angle); //kiri
    pwm2 = mtrsPd + abs(angle); //kanan
    forward();
}
//setting the steering command if it is veering
else if (target_yaw < yawOld) {
    pwm1 = mtrsPd + abs(angle); //kiri
    pwm2 = mtrsPd - abs(angle); //kanan
    forward();
}
error = last_error;
}

```

Gambar 3. 19 Sistematika Troli Bergerak Lurus

Metode yang digunakan dalam tugas akhir ini untuk mendapatkan parameter Kp (konstanta proporsional) dan Kd (konstanta diferensial) yang cocok adalah melalui metode penyetelan dengan metode *trial* dan *error* walaupun banyak metode seperti ziegler-nichols dan cohen-coon. Pendekatan awal yang dilakukan adalah dengan mengatur nilai setpoint dan mengatur Kd sebagai 0, sementara Kp diatur dengan nilai terkecil yaitu 1. Setelah Kp terkecil diatur, langkah selanjutnya adalah meningkatkan nilai Kp secara bertahap hingga gelombang osilasi hampir mencapai nilai setpoint yang diinginkan.

Setelah osilasi mendekati setpoint, kemudian meningkatkan nilai Kd untuk mengurangi osilasi yang terjadi pada nilai kesalahan proporsional. Hal ini bertujuan untuk mengurangi gelombang osilasi yang disebabkan oleh kesalahan proporsional dari Kp, sehingga sistem menjadi lebih stabil. Dengan mengatur parameter Kp dan Kd secara iteratif dengan memperhatikan osilasi dan kesalahan proporsional, dapat mencapai parameter yang cocok untuk sistem yang lebih stabil dan mendekati setpoint yang diinginkan.

3.6.6. Sistematika Penerimaan Data UART

Penerimaan data UART dilakukan apabila raspberry telah mengirim data serial melalui baud rate, data yang dikirim tersebut selanjutnya harus dipecah untuk digunakan dalam memproses data pengenalan *hand gesture*. Data yang dikirim tersebut masih menggunakan json parsing sehingga perlu dilakukan deserialize data, data tersebut disimpan pada void paket(). Ketika paket telah sampai pada arduino selanjutnya data akan dipecah menjadi 3 buah data, xcenter berupa titik koordinat horizontal tengah objek, ycenter berupa titik koordinat vertikal tengah objek, sedangkan kondisi merupakan perintah yang dikirim untuk menggerakkan motor apabila perintah *hand gesture* diberikan. Program tersebut telah diperlihatkan pada gambar 3.20.

```
void paket() {
    if (Serial.available()) {
        dataT = Serial.readStringUntil('\n');
        deserializeJson(doc, dataT);
        xcenter = doc["xc"].as<int>();
        ycenter = doc["yc"].as<int>();
        kondisi = doc["k"].as<int>();
    }
}
```

Gambar 3. 20 Penerimaan Data Serial UART Arduino

3.7 Sistematika Pemrograman Python Pada Raspberry Pi

3.7.1. Sistematika Pendektsian Tangan

Pada mulanya pemrograman *hand gesture* pada raspberry dengan memahami prinsip kerja dari *library mediapipe*, Untuk mendapatkan pendektsi landmark tangan, diperlukan modifikasi pada gambar *real-time* yang awalnya berformat BGR menjadi gambar dengan format RGB. Hal ini dilakukan agar perbedaan warna antara lengan dan latar belakang

terlihat jelas. Setelah gambar diubah menjadi format RGB, selanjutnya format RGB tersebut diubah menjadi gambar dengan format Depth Image yang menunjukkan kedalaman dan ketebalan objek, khususnya tangan dalam bentuk tiga dimensi (3D), tanpa mencakup landmark tangan. Depth image map dan *2D hand skeleton* telah dijadikan model dalam TensorFlow Lite melalui pelatihan oleh mediapipe. Sebagai hasilnya, didapatkan model pose tangan 3D yang mencakup landmark tangan.

```
def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    if self.results.multi_hand_landmarks:
        for handLms in self.results.multi_hand_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, handLms,
                                           self.mpHands.HAND_CONNECTIONS)
    return img
```

Gambar 3. 21 Sistematika Pendektsian Tangan

Untuk mendapatkan pendektsian landmark tangan menggunakan Mediapipe, langkah pertama adalah mengubah gambar menjadi format berwarna RGB. Setelah itu, hasil dari gambar RGB tersebut akan diakses dan disimpan dalam variabel "self.result". Kemudian, variabel "self.result" akan digabungkan dengan "multi_hand_landmark" yang merupakan representasi posisi tangan dalam 2D. Selanjutnya, titik-titik dalam "self.result" akan digambar menggunakan metode "self.mpDraw" untuk membuat titik-titik pada sendi tangan. Terakhir, metode "self.mpHand.Hand_Connection" akan menghubungkan setiap titik sendi untuk membentuk struktur rangka landmark tangan.

3.7.2. Sistematika *Hand Gesture Tracking*

Setelah dilakukan pendektsian pada tangan selanjutnya dilakukan tracking, Tracking dilakukan untuk memantau atau mengikuti pergesiture atau perubahan suatu objek atau entitas dalam jangka waktu tertentu. Secara umum, tracking dilakukan dengan menggunakan kamera untuk mengumpulkan data mengenai posisi, kecepatan, atau orientasi objek yang sedang dipantau. Data tersebut kemudian dianalisis dan digunakan untuk menghasilkan informasi mengenai pergesiture atau perubahan yang

terjadi pada objek tersebut. Hal ini dilakukan juga untuk tujuan penggunaan servo dalam melacak objek agar objek tepat pada titik tengah kamera.

```

h, w, c = img.shape
cx, cy = int(lm.x * w), int(lm.y * h)
xList.append(cx)
yList.append(cy)
cv2.putText(img, str('Tangan Terdeteksi'), (10, 120), cv2.FONT_HERSHEY_PLAIN, 1,
            (0, 128, 205), 2)
lmList.append([id, cx, cy])
xmin, xmax = min(xList), max(xList)
ymin, ymax = min(yList), max(yList)
bbox = xmin, ymin, xmax, ymax
bx, by = (bbox[0]-20 + bbox[2] + 20) //2, (bbox[3] + 20 + bbox[1] - 20)//2
b = [bx, by]
b.append(0)
if draw:
    cv2.rectangle(img, (xmin - 20, ymin - 20), (xmax + 20, ymax + 20),
                  (255, 255, 255), 2)

```

Gambar 3. 22 Sistematika *Hand Gesture Tracking*

Pada awalnya, cx (koordinat horizontal tengah) dan cy (koordinat vertikal tengah) digunakan sebagai titik tengah objek yang terdeteksi dalam gambar. Proses pendektsian ini menggunakan koordinat objek lengan dengan lebar dan tinggi piksel sebesar 640 x 480, yang memberikan informasi tentang lokasi objek dalam ruang gambar.

Setelah pendektsian dilakukan, nilai cx dan cy mengandung informasi tentang koordinat besar dari lokasi objek. Ini termasuk xmin (koordinat horizontal kiri), xmax (koordinat horizontal kanan), ymin (koordinat vertikal atas), dan ymax (koordinat vertikal bawah). Dengan menggunakan batas persegi panjang yang ditentukan oleh nilai-nilai ini, kita dapat membatasi wilayah atau kotak pembatas objek dalam gambar.

Lebih spesifik, xmin dan xmax menentukan batas kiri dan kanan kotak pembatas objek pada sumbu horizontal (sumbu x). Sementara itu, ymin dan ymax menentukan batas atas dan bawah kotak pembatas objek pada sumbu vertikal (sumbu y). Dengan demikian, koordinat-konkordinat ini membantu dalam menentukan posisi dan ukuran kotak pembatas objek yang akan digunakan dalam proses pengolahan lebih lanjut. Dalam konteks pengolahan gambar, perintah bbox (bounding box) digunakan untuk mengidentifikasi dan mengisolasi objek atau area tertentu dalam gambar.

Bounding box membatasi area yang akan diproses untuk menghasilkan mask atau melakukan penghilangan latar belakang (background subtraction) di sekitar objek yang spesifik. Dengan menggunakan kotak pembatas ini, kita dapat fokus pada area yang relevan dan meningkatkan efisiensi dalam proses pengolahan gambar. Selain itu, penggunaan bounding box juga memungkinkan pelacakan objek. Dengan menandai sisi-sisi objek yang terdeteksi melalui kotak pembatas, kita dapat mendapatkan nilai titik tengah koordinat objek.

3.7.3. Sistematika *Hand Gesture Recognition*

Setelah dilakukan pelacakan pada *hand gesture*, dilanjutkan dengan pengenalan objek menggunakan titik koordinat objek. Sebelumnya, langkah awal adalah melakukan pengenalan dengan membuat variabel yang menyimpan nilai koordinat titik persambungan pada jari-jari tangan. Variabel "fingers" mengacu pada titik persambungan "tip" pada setiap ujung jari, dengan titik "pip" pada bagian sendi jari ketiga. Ketika nilai variabel "fingers" sama dengan 0, koordinat titik "tip" lebih kecil dibandingkan dengan koordinat titik "pip". Namun, jika koordinat titik "tip" lebih besar daripada koordinat titik "pip", maka nilai variabel "fingers" menjadi 1.

Pada Gambar 3.23 *gesture* mundur tiap titik tip lebih besar dari titik pip, sehingga pengenalan jari didapatkan bahwa semua jari terbuka pada gerak mundur. Pada *gesture* kanan thumb_tip terbuka sedangkan semua jari tertutup dan kondisi selanjutnya koordinat jari jempol thumb_tip / x1 lebih kecil dari titik koordinat kelingking pinky_tip / x2. Pada *gesture* kiri thumb_tip terbuka sedangkan semua jari tertutup dan kondisi selanjutnya koordinat jari jempol thumb_tip / x1 lebih besar dari titik koordinat kelingking pinky_tip / x2. Pada *gesture* maju jari jari telunjuk index_finger_tip dan jari tengah middle_finger_tip terbuka sedangkan pada jari lainnya tertutup. Pada *gesture* diam seluruh jari tertutup untuk menghentikan motor bergerak.

```
if len(lmList) != 0:
    x1, y1 = lmList[4][1:]
    x2, y2 = lmList[20][1:]
    gx = b[0] - width // 2
    gy = b[1] - height // 2
```

```

if total == 5:
    cv2.putText(img, str("Mundur"), (10, 90), cv2.FONT_HERSHEY_PLAIN,2,
               (0, 170, 255),3)
    k = 66

elif (fingers == [1,0,0,0,0] and fingers != [1,1,1,1,1] and x1 < x2):
    cv2.putText(img, str("Kanan"), (10, 90), cv2.FONT_HERSHEY_PLAIN,2,
               (0, 0, 255),3)
    k = 55

elif (fingers == [1,0,0,0,0] and fingers != [1,1,1,1,1] and x1 > x2):
    cv2.putText(img, str("Kiri"), (10, 90), cv2.FONT_HERSHEY_PLAIN,2,
               (0, 0, 255),3)
    k = 44

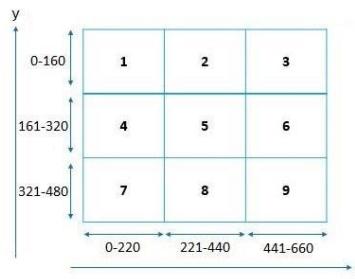
elif (fingers == [0,1,1,0,0]):
    cv2.putText(img, str("front"), (10, 90), cv2.FONT_HERSHEY_PLAIN,2,
               (0, 0, 255),3)
    k = 33

else:
    cv2.putText(img, str("Berhenti"), (10, 90), cv2.FONT_HERSHEY_PLAIN,2,
               (0, 0, 255),3)
    k = 77

```

Gambar 3. 23 Sistematika Hand Gesture Recognition

Tiap titik koordinat pada computer vision dapat dilihat pada Gambar 3.23, apabila objek kekiri maka nilai koordinat piksel horizontal x akan berkurang mendekati 0 sedangkan apabila objek kekanan maka nilai koordinat horizontal x akan bertambah hingga 640 piksel.



Gambar 3. 24 Titik Koordinat Piksel Pada Computer Vision

3.7.4. Sistematika Pengiriman Serial UART Raspberry Pi

Setelah dilakukan pengenalan *hand gesture* selanjutnya dilakukan pengiriman data melalui serial uart, awal mula dilakukan penentuan variabel Serial, berupa True dan False ketika kondisi berupa True maka

serial dapat dihubungkan dengan arduino. Perlu diperhatikan apabila serial usb pada arduino dan raspberry sedang tidak dihubungkan maka program tidak dapat dijalankan karena program mengenali bahwa perangkat sedang tidak terhubung dengan arduino. Pada perintah /dev/ttyACM0 merupakan perangkat port yang terhubung pada perangkat dengan sistem operasi Linux.

```
useSerial = False
if useSerial == True :
    ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1.11)
    time.sleep(2)
```

Gambar 3. 25 Membaca Port Arduino

Pada Ggaambar 3.25 diperlihatkan bahwa pengiriman data serial menggunakan baud rate sebesar 9600, dengan waktu delay apabila port tidak terkoneksi sebesar 2 detik. Ketika serial bernilai True maka port dibaca untuk dilakukan pengiriman data melalui serial baudrate 9600, ketika telah terjadi pengiriman arduino akan menerima data yang kemudian dilakukan permrosesan data untuk menggerakkan troli. Data tersebut berupa nilai xcenter, ycenter dan kondisi perintah *hand gesture*.

```
dataObject ={
    "xc":gx,
    "yc":gy,
    "k":k
}
else:
    gx = 404
    gy = 404
    k = 404
    dataObject ={
        "xc":gx,
        "yc":gy,
        "k":k
    }
# print(data)
if useSerial == True :
    dataJson = json.dumps(dataObject)
    print(dataJson)
    ser.write((dataJson + '\n').encode())
```

Gambar 3. 26 Pengiriman Data Serial UART Raspberry PI

Pada Gambar 3.26 merupakan data yang perlu dikirim melalui serial UART, variabel k merupakan nilai perintah pada pengenalan *hand gesture* sedangkan nilai xc dan yc merupakan nilai koordinat titik center objek yang terdeteksi oleh kamera. Langkah terakhir kemudian data tersebut diparsing menggunakan json untuk dikirimkan melalui ser.write.

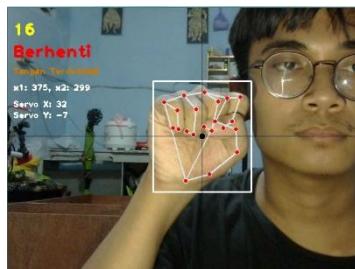
BAB IV

HASIL DAN PEMBAHASAN

Pada bab IV ini, akan diuraikan tentang pengujian sistem yang telah dirancang. Terdapat dua jenis pengujian yang dilaksanakan, yaitu pengujian individual untuk setiap komponen dan pengujian keseluruhan atau terintegrasi. Tujuan dari pengujian setiap bagian adalah untuk mengevaluasi kinerja rangkaian dan memastikan bahwa rangkaian tersebut layak untuk diimplementasikan dalam sistem. Sementara itu, tujuan dari pengujian keseluruhan adalah untuk menilai kehandalan sistem dalam menghadapi kesalahan. Rincian tentang masing-masing pengujian akan dijelaskan pada penjelasan selanjutnya.

4.1 Pengujian Gerak Berdasarkan Perintah Dari *Hand Gesture*

Pengujian pendekripsi *gesture* tangan dilakukan dengan mendekripsi telapak tangan menggunakan library mediapipe dengan modul OpenCV. Pengujian dilakukan secara berulang kali untuk mengambil sampel keakuratan deteksi, apakah pendekripsi dapat dilakukan atau tidak. Pemrograman dilakukan pada python menggunakan modul library OpenCV. Hasil pendekripsi *hand gesture* menggunakan modul library OpenCV dapat dilihat pada Gambar 4.1. Objek tangan terdeteksi ditandai dengan munculnya kotak berwarna putih yang terletak di area sekitar tangan yang terdeteksi.

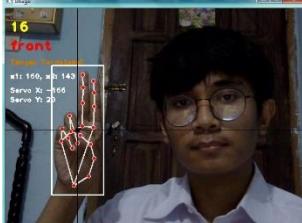
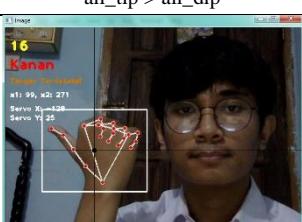


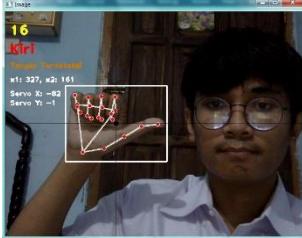
Gambar 4. 1 Pendekripsi Pola Gerak Tangan

Pengujian *hand gesture* pada alat dilakukan untuk menguji kinerja sistem pada alat. Saat kita memberikan *hand gesture* maju maka troli akan bergerak lurus. Saat kita memberikan *hand gesture* mundur maka troli

akan bergerak mundur. Saat kita memberikan *hand gesture* kiri, maka troli akan berbelok ke kiri. Saat kita memberikan *hand gesture* kanan, maka troli akan berbelok ke kanan. Sedangkan jika kita memberikan *hand gesture* diam, maka troli akan berhenti.

Tabel 4. 1 Pengujian Gerak Berdasarkan Perintah Dari *Hand Gesture*

Kondisi	Tampilan	Gerak
Maju	 Index_tip>index_dip Middle_tip>middle_dip	 Maju
Mundur	 all_tip > all_dip	 Mundur
Kanan	 thumb_tip<pinky_tip	 Kanan

Kondisi	Tampilan	Gerak
Kiri	 Thumb_tip>pinky_tip	 Kiri
Berhenti	 all_tip<all_dip	 Berhenti

Hasil pengujian sesuai Tabel 4.1 secara keseluruhan menunjukkan bahwa troli memiliki kemampuan untuk bergerak maju, belok ke kanan atau kiri, serta berputar mengikuti arah gesture objek berdasarkan gerak pola tangan. Webcam yang digunakan berhasil membaca objek *hand gesture* dengan baik, sehingga troli dapat mengikuti perintah *hand gesture* tersebut. Sensor dan motor yang digunakan beroperasi sesuai dengan program yang dijalankan, memastikan troli dapat bergerak dengan lancar sesuai instruksi. Namun, perlu diperhatikan bahwa intensitas cahaya yang tidak stabil dapat mengganggu kemampuan webcam dalam membaca objek, yang berakibat pada kehilangan deteksi objek dan menyebabkan troli tidak dapat bergerak secara efektif.

Selanjutnya untuk mengetahui sejauh mana gestur tangan manusia dapat diidentifikasi dengan rentang waktu 2 detik saat melakukan penangkapan gestur, langkah selanjutnya adalah melihat tingkat akurasi gestur yang berhasil terdeteksi. Dalam Tabel 4.2, terlihat hasil dari dua puluh percobaan pendekripsi gestur yang dilakukan, berdasarkan gestur instruksi yang tercantum pada Tabel 4.1 sebelumnya.

Tabel 4. 2 Identifikasi Gesture Tangan Pencahayaan Ideal

No	Gesture	Terdeteksi/Percobaan	Lama <i>gesture</i> terdeteksi (detik)	Kondisi Pencahayaan (Lux)	Error (%)
1	Maju	18/20	3	63	10
2	Mundur	20/20	1	67	0
3	Kanan	19/20	2	67	5
4	Kiri	19/20	1	64	5
5	Berhenti	20/20	1	63	0
Rata-Rata Error (%)					4

Kesalahan rata-rata yang diperoleh sebesar 4% kesalahan terbesar terjadi pada pembacaan *gesture* maju yaitu sebesar 10%. Hal tersebut disebabkan pencahayaan latar belakang dan jarak tangan dengan webcam. Selain itu terdapat perbedaan dalam pendekripsi hand gesture pada kondisi penurunan cahaya. Pola mundur dan diam menghasilkan hasil pengujian terbaik, sedangkan pola maju, kanan, dan kiri memiliki tingkat pendekripsi yang kurang baik ketika terjadi penurunan cahaya. Pada pola maju, terjadi ketidakmampuan dalam mendekripsi pola saat intensitas cahaya menurun, dan pola berubah menjadi pola kanan karena koordinat titik Thump_tip yang lebih besar daripada titik koordinat Pinky_tip. Sedangkan pada pola kanan dan kiri, pola dapat berubah menjadi berhenti ketika terjadi penurunan cahaya.

Selain itu, dalam skenario jika terdapat rintangan yang berjarak kurang dari 20 cm di depan troli, sistem akan secara otomatis menghentikan pergerakan troli. Sensor ultrasonik digunakan untuk mendekripsi rintangan di sebelah kanan atau kiri troli. Jika rintangan terdeteksi di sebelah kanan, troli akan berbelok ke kiri sebagai responsnya. Begitu pula sebaliknya, jika rintangan terdeteksi di sebelah kiri, troli akan berbelok ke kanan untuk menghindari rintangan tersebut.

4.2 Pengujian Hand Gesture Terhadap Intensitas Cahaya

Intensitas cahaya yang cukup dan jarak objek terhadap webcam yang ideal sangat berpengaruh terhadap hasil pendekripsi objek. Untuk mencapai hasil pendekripsi objek yang optimal, penting untuk memperhatikan nilai rata-rata lux pada saat dilakukan pengujian pada

sebuah ruangan. Untuk pengukuran intensitas cahaya, digunakan Lux Light Meter.

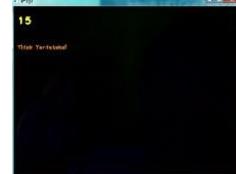
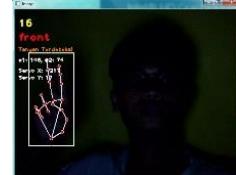
Dalam pengujian ini, dilibatkan tiga buah ruangan dengan cahaya lampu yang berbeda-beda. Variasi cahaya lampu ini bertujuan untuk mengetahui pengaruhnya terhadap deteksi objek. Dengan menggunakan Lux Light Meter, dilakukan pengukuran intensitas cahaya di setiap ruangan

Tabel 4. 3 Tabel Pengujian Nilai Lux

Deteksi	Rata- Rata Nilai Lux		
	Ruang A	Ruang B	Ruang C
Mundur	0	11	244
Maju	0	15	204
Kiri	0	16	227
Kanan	0	14	243
Berhenti	0	12	227

Langkah awal dalam proses pengujian ini adalah melakukan pengukuran nilai rata-rata lux di dalam ruangan dalam 3 kamar berbeda dengan tujuan untuk menentukan seberapa kuat intensitas cahaya yang dibutuhkan oleh kamera dalam mendeteksi objek. Dalam konteks ini, nilai lux digunakan sebagai parameter untuk mengukur intensitas cahaya yang ada di dalam ruangan, dan hasil pengukuran ini menjadi referensi penting untuk tahap pengujian berikutnya. Setelah nilai rata-rata lux berhasil diperoleh, dilakukan pengujian terhadap intensitas cahaya yang dibutuhkan dan hasilnya dicatat secara sistematis pada Tabel 4.2. Hal ini bertujuan untuk memperoleh data yang lebih lengkap dan mendetail terkait intensitas cahaya yang optimal dalam mendeteksi objek. Dan hasil pengujian pada ruangan ditampilkan pada Tabel 4.3.

Tabel 4. 4 Hasil Pengujian Deteksi Tangan

Pola Gesture	Nilai Lux		Hasil Deteksi
	Lu x	Kondisi	
Mundur	0	Tidak Terdeteksi	
	4	Terdeteksi	
	12	Terdeteksi	
	204	Terdeteksi	
Maju	2	Tidak Terdeteksi	
	7	Terdeteksi	

Pola Gesture	Nilai Lux		Hasil Deteksi
	Lu x	Kondisi	
	12	Terdeteksi	 
	220	Terdeteksi	 
Kanan	2	Tidak Terdeteksi	
	6	Tidak Terdeteksi	
	14	Terdeteksi	 
	235	Terdeteksi	 

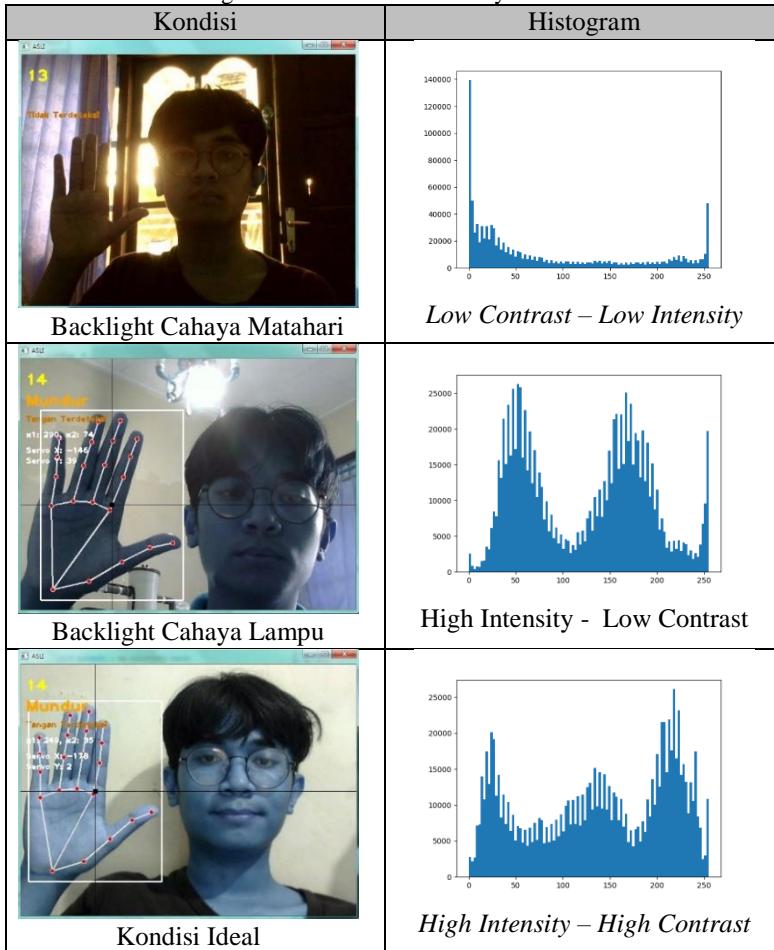
Pola Gesture	Nilai Lux		Hasil Deteksi
	Lu x	Kondisi	
Kiri	1	Tidak Terdeteksi	
	8	Tidak Terdeteksi	
	14	Terdeteksi	
	207	Terdeteksi	
Berhenti	0	Tidak Terdeteksi	
	4	Tidak Terdeteksi	

Pola Gesture	Nilai Lux		Hasil Deteksi
	Lu x	Kondisi	
	13	Terdeteksi	 
	242	Terdeteksi	 

Dari hasil pengujian didapatkan nilai lux rata-rata dibutuhkan oleh kamera dalam mendeteksi setiap objek adalah 10 lux, ruangan teras membutuhkan 60 lux sedangkan ruangan seperti ruang tamu, ruang makan dan kamar tidur membutuhkan setidaknya 120 lux. Kamera sudah dapat mendeteksi objek *hand gesture* dibawah 120 lux yakni sebesar 10 lux adalah ruangan dalam keadaan gelap dengan sedikit intensitas cahaya. Pada pola gerak mundur terlihat bahwa dengan menggunakan 4 lux dapat mendeteksi pola gerak tangan, sedangkan pada gerak maju, kanan, kiri, mundur dan diam dibutuhkan setidaknya 10 lux dalam ruangan tersebut.

Setelah melakukan pengujian intensitas cahaya di ruangan, dilakukan analisis lebih lanjut terkait kondisi backlight atau cahaya latar belakang. Dalam analisis ini, tangkapan pendektsian gesture pada mode pendektsian "imageRGB" dievaluasi dengan mempertimbangkan intensitas cahaya yang masuk ke kamera. Analisis intensitas cahaya ini menggunakan histogram kontras intensitas citra. Hasil analisis histogram tersebut kemudian ditampilkan dalam Tabel 4.4.

Tabel 4. 5 Histogram Low Contrast Pencahayaan Pada Metode RGB



Histogram digunakan dalam analisis citra untuk menampilkan distribusi intensitas piksel di dalam gambar. Dalam konteks pendekripsi sistem hand gesture saat terjadi backlight cahaya matahari, backlight cahaya lampu, dan kondisi ideal, histogram dapat memberikan wawasan tentang bagaimana tingkat cahaya yang berbeda mempengaruhi pendekripsi gesture tangan. Pada saat terjadi backlight cahaya matahari,

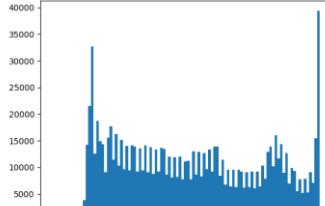
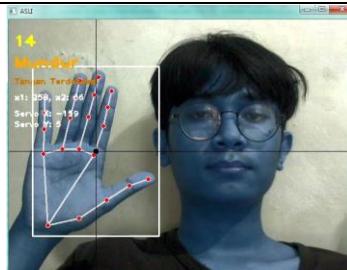
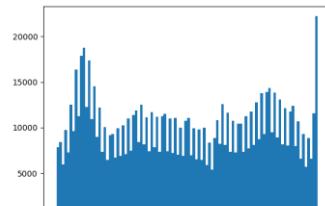
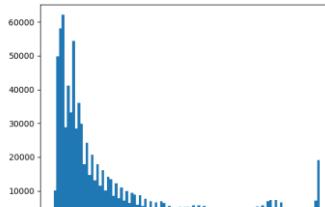
histogram menunjukkan adanya puncak yang lebih rendah pada intensitas piksel di area tangan yang terkena siluet cahaya matahari. Hal ini disebabkan oleh kontras yang rendah antara tangan dan latar belakang yang terang, sehingga objek tangan tidak dapat terdeteksi oleh kamera.

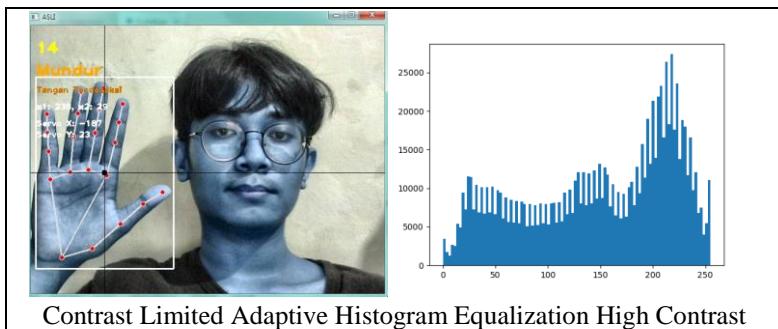
Saat terjadi backlight cahaya lampu, histogram juga menggambarkan peningkatan intensitas pada area sekitar tangan akibat siluet yang terbentuk oleh cahaya yang dipancarkan dari belakang. Histogram menunjukkan puncak yang lebih tinggi pada intensitas piksel di daerah tersebut. Namun beberapa objek yang tidak terkena cahaya akan terlihat gelap yang menunjukkan puncak yang lebih rendah dari pada intensitas piksel yang lain. Sehingga apabila semua cahaya menutupi semua objek, pendektsian akan menghilang karena tidak adanya intensitas cahaya yang dapat ditangkap oleh kamera.

Dalam kondisi ideal, histogram citra tangan akan menunjukkan distribusi intensitas yang lebih merata dan konsisten di seluruh piksel tangan. Ini akan memudahkan pendektsian dan pengenalan gesture tangan dengan tingkat akurasi yang tinggi. Histogram dengan distribusi yang lebih merata dan simetris dapat memberikan informasi tentang kualitas citra yang baik, serta memberikan landasan yang solid bagi sistem pendektsian hand gesture.

Informasi ini dapat membantu dalam mengidentifikasi tantangan yang terkait dengan backlighting dan memungkinkan pengembangan solusi teknis yang sesuai untuk meningkatkan pendektsian gesture tangan pada kondisi cahaya yang bervariasi. Beberapa Langkah yang dapat dilakukan untuk melakukan perbaikan kualitas pada citra, diantaranya sudah dicantumkan pada Tabel 4.5.

Tabel 4. 6 Metode Perbaikan Citra
Jenis Metode Perbaikan Citra

	
<p>Histogram Equalization Low Contrast – High Intensity</p>	
	
<p>Histogram Equalization High Contrast – High Intensity</p>	
	
<p>Contrast Limited Adaptive Histogram Equalization Low Contrast</p>	



Dalam analisis histogram terkait pendekripsi sistem hand gesture, metode equalization histogram dapat digunakan untuk memperbaiki citra dengan low contrast dan high contrast. Pada kondisi low contrast, histogram citra akan menunjukkan distribusi intensitas piksel yang terkompresi, dengan banyak piksel berkumpul di rentang intensitas yang sempit. Dalam hal ini, metode equalization histogram dapat digunakan untuk meningkatkan kontras citra dengan meratakan distribusi intensitas piksel.

Metode equalization histogram, histogram citra yang awalnya tersebar akan diperatakan, menghasilkan distribusi intensitas yang lebih merata dan kontras yang lebih seimbang. Namun kekurangan pada metode equalization histogram untuk hand gesture adalah banyaknya noise atau gambar yang buram akibat keterpaksaan piksel mempertinggi intensitas, sehingga seakan pendekripsi lebih cerah sedangkan pada pendekripsi gesture tangan dibutuhkan tingkat kedalaman objek yang akan dideteksi.

Metode Contrast Limited Adaptive Histogram Equalization (CLAHE) juga dapat digunakan untuk mengatasi tantangan low contrast dan high contrast. Pada kondisi high contrast, metode CLAHE dapat mempertahankan detail gambar tanpa menyebabkan kelebihan kontras yang berlebihan. Sedangkan pada kondisi low contrast, metode CLAHE dapat meningkatkan kontras secara adaptif tanpa mengorbankan detail yang penting. Dengan mempertimbangkan kondisi kontras yang spesifik, metode CLAHE dapat memberikan solusi yang lebih optimal dalam meningkatkan visualisasi gesture tangan pada citra dan mendukung

pendeksihan hand gesture dengan lebih baik dibandingkan metode equalization histogram.

Analisis histogram dan penggunaan metode equalization histogram, termasuk metode CLAHE, dapat memberikan perbaikan citra yang efektif untuk meningkatkan pendeksihan sistem hand gesture dalam situasi low contrast dan high contrast.

4.3 Pengujian Jarak Pendeksihan *Hand Gesture* Pada Kamera

Pengujian ini dilakukan dengan tujuan untuk menemukan jarak yang optimal antara kamera dan objek *gesture* tangan, sehingga dapat menghasilkan gambar yang jelas dan fokus. Dalam pengujian tersebut, kamera ditempatkan pada jarak yang berbeda-beda dari objek yang sama, dan hasilnya diamati secara teliti untuk menentukan jarak terbaik yang dapat menghasilkan gambar yang jelas dan tajam. Selanjutnya, hasil dari pengujian jarak kamera dicatat dengan cermat dan dianalisis secara mendalam untuk menentukan jarak optimal yang diperlukan. Hasil analisis kemudian diungkapkan secara rinci dalam Tabel 4.4 berupa tabel jarak kamera terhadap objek.

Tabel 4. 7 Tabel Jarak Kamera Terhadap Objek

No.	Jarak Kamera	Kondisi	Hasil Deteksi
1.	25 cm	Terdeteksi	
2.	50 cm	Terdeteksi	

No.	Jarak Kamera	Kondisi	Hasil Deteksi
3.	100 cm	Terdeteksi	
4.	150 cm	Terdeteksi	
5.	200 cm	Terdeteksi	
6.	300 cm	Terdeteksi	
7.	400 cm	Terdeteksi	

No.	Jarak Kamera	Kondisi	Hasil Deteksi
8.	500 cm	Terdeteksi	
9.	650 cm	Terdeteksi	
10.	700 cm	Tidak Terdeteksi	

Dalam Tabel 4.4, terlihat bahwa kamera memiliki kemampuan untuk mendeteksi objek pada jarak minimal 25 cm hingga jarak maksimal 700 cm. Namun, setelah melewati jarak sejauh 650 cm, kamera tidak lagi mampu mengenali objek yang ada di tangan. Hal ini menunjukkan bahwa semakin jauh tangan terdeteksi oleh kamera, semakin sulit untuk mengenali bentuk tangan tersebut. Oleh karena itu, ketika objek berada pada jarak yang terlalu jauh, bentuk sendi pada tangan tidak dapat diidentifikasi dengan jelas. Dengan demikian, dapat diketahui bahwa jarak yang ideal untuk mendeteksi objek tangan adalah berkisar antara 50 hingga 150 cm. Pada rentang ini, kamera memiliki kemampuan yang baik untuk mengenali dan mengidentifikasi bentuk tangan dengan akurasi yang memadai.

4.4 Pengujian Tracking Servo Terhadap Objek Hand Gesture

Pengujian dilakukan untuk mengetahui arah gerak motor servo saat mengikuti objek tangan, dilakukan pengujian dengan menggunakan motor servo yang memiliki sudut maksimal 180° . Pada posisi awal, motor servo diatur menghadap ke depan dengan sudut 90° . Ketika objek terdeteksi maka servo akan bergerak mengikuti objek, ketika objek berada di kiri maka servo akan mengurangi derajatnya, sehingga sudut servo kiri memiliki rentang sudut $0-90^\circ$. Sedangkan ketika objek berada di kanan maka servo akan menambah sudut derajatnya, sehingga sudut servo kanan memiliki rentang sudut $90-180^\circ$. Pengujian ini bertujuan mengukur rentang sudut saat objek berada di kiri dan di kanan untuk melihat apakah servo dapat menyesuaikan objek. Object tracking diuji dengan menggunakan busur derajat untuk mencocokkan posisi objek dengan sudut sebenarnya servo. Hasil pengujian ditampilkan pada Tabel 4.15. dan Gambar 4.19.

Pengujian kesalahan ini bertujuan untuk mengukur kemampuan sensor menerima rangsangan serta respon servo saat terdapat objek. Parameter yang digunakan adalah posisi sudut objek sebagai nilai acuan dan sudut sebenarnya sebagai nilai sebenarnya dengan melihat sudut pada LCD.

Tabel 4. 8 Hasil Pengujian Response Tracking

Tabel 4.3 Hasil Pengujian Response Tracking				
Percobaan Ke-	Posisi Sudut Objek	Sudut Sebenarnya Servo	Selisih	Error(%)
1	0°	0°	0°	0%
2	30°	28°	2°	6,6%
3	45°	47°	2°	4,4%
4	60°	61°	1°	1,6%
5	90°	90°	0°	0%
6	120°	117°	2°	2,5%
7	135°	134°	1°	0,74%
8	150°	152°	2°	1,3%
9	165°	166°	1°	0,6%
10	180°	180°	0°	0%

Percobaan Ke-	Posisi Sudut Objek	Sudut Sebenarnya Servo	Selisih	Error(%)
Rata-rata error (%)				1,77%

Berdasarkan hasil pengujian yang tercantum dalam Tabel 4.5, dapat ditarik kesimpulan bahwa ketika sudut servo mendeteksi objek *hand gesture* kurang dari 90° , servo akan mengurangi sudutnya menjadi kurang dari 90° . Sebaliknya, jika sudut objek melebihi 90° , servo akan menambah sudutnya menjadi lebih dari 90° . Selama proses pendekripsi objek, rata-rata respon error servo tercatat sebesar 1,77%. Hal ini disebabkan oleh fakta bahwa servo menggunakan perintah untuk terus-menerus menambah dan mengurangi sudut saat mengamati objek. Dengan demikian, servo berupaya untuk mengoreksi dan menyesuaikan posisinya agar sesuai dengan objek yang dideteksi.



Gambar 4. 2 Posisi Awal Servo Sudut 90°

4.5 Pengujian Gerak Lurus Troli

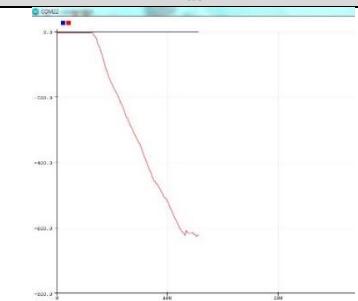
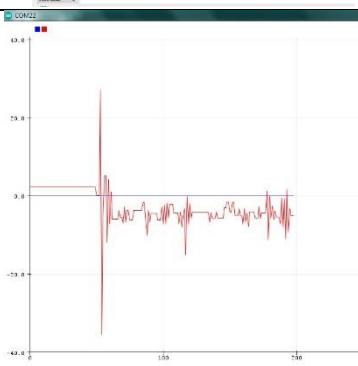
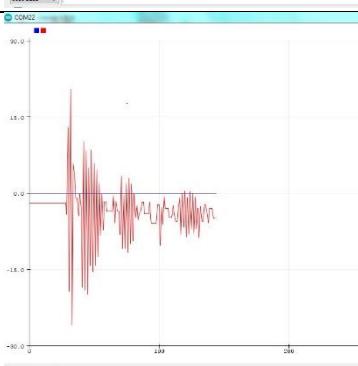
Pengujian Gerak lurus troli atau disebut pengujian error PD dilakukan untuk mengetahui apakah troli dapat bergerak lurus atau tidak. Pengujian dilakukan menggunakan nilai acuan busur derajat dan sebuah meteran, troli diuji untuk dapat bergerak sesuai lintasan yang sudah ada berjarak 150cm. Troli selanjutnya diukur persentase kesalahan dan sudut kesalahan troli bergerak tidak sesuai lintasan. Hasil pengujian tersebut kemudian dicatat pada Tabel 4.6.

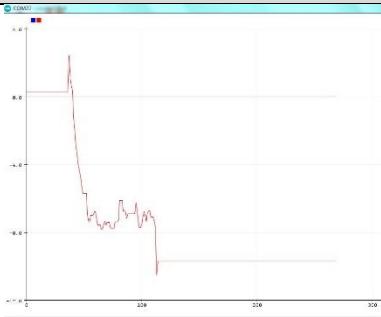
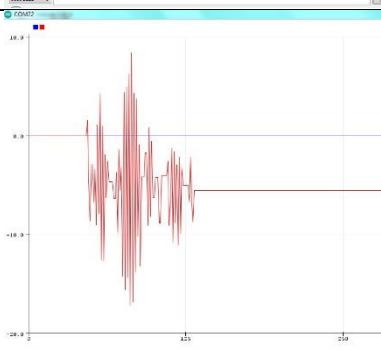
Tabel 4. 9 Pengujian Gerak Lurus Proporsional-Derivative

Percobaan Ke-	Jarak	Sudut Lintasan Acuan	Sudut Troli Sebenarnya	Selisih	error
1	150 cm	90°	85°	5°	5,5%
2	150 cm	90°	90°	0°	0%
3	150 cm	90°	92°	2°	2,2%
4	150 cm	90°	90°	0°	0%
5	150 cm	90°	95°	5°	5,5%
6	150 cm	90°	90°	0°	0%
7	150 cm	90°	89°	1°	1,1%
8	150 cm	90°	85°	5°	5,5%
9	150 cm	90°	90°	0°	0%
10	150 cm	90°	90°	0°	0%
Rata-Rata error				89,6°	1,98%

Setelah dilakukan serangkaian pengujian yang teliti dan mendalam, ditemukanlah data yang menunjukkan bahwa rata-rata sudut troli yang sebenarnya adalah 89,6°. Keberadaan sudut tersebut dapat dijelaskan oleh adanya masalah pada roda depan yang mengalami ketidakmampuan dalam melakukan manuver. Terkadang, roda depan tidak berputar secara optimal, yang pada gilirannya mengakibatkan perubahan sudut troli yang sedikit saat troli mencapai tujuan. Akan tetapi, kendati mengalami kendala tersebut, troli masih tetap berada dalam lintasan yang ditetapkan. Oleh karena itu, penggunaan sistem kontrol Proporsional-Derivatif (PD) pada troli memiliki pengaruh yang sangat signifikan dalam menjaga stabilitas dan kinerja pergerakan troli. Dengan adanya penggunaan PD, troli hanya mengalami kesalahan sudut sebenarnya sebesar 1,98%, yang dapat dikatakan sebagai angka yang sangat rendah dan mengindikasikan tingkat akurasi yang tinggi dalam melakukan pergerakan.

Tabel 4. 10 Grafik Gerak Lurus Troli

Kondisi Troli	Keterangan	Hasil
Tanpa PD	Cenderung bergerak kanan	
PD Kp = 11.2 Kd = 12	Gerak lurus	
Kp > Kd Kp = 13 Kd = 7	Gerak kanan-kiri	

Kondisi Troli	Keterangan	Hasil
Perkecil Kp dan Kd Kp = 1 Kd = 3	Respon lambat	
Perbesar Kp dan Kd Kp = 13 Kd = 15	Respon cepat	

Pada Tabel 4.7 diperlihatkan hubungan antara gerak troli dengan target setpoint, dimana garis biru yang menunjuk arah 0,0 adalah setpoint target atau garis lurus robot sedangkan garis merah merupakan nilai angle robot troli saat bergerak maju, jika grafik merah kearah bawah menandakan robot bergerak ke kanan, sedangkan jika grafik merah ke atas maka robot bergerak kekiri.

Ketika Kp dan Kd diperkecil respon robot ketika terjadi kesalahan lambat sehingga robot akan keluar dari jalur, sedangkan jika Kp dan Kd diperbesar maka trespon robot akan terlalu cepat yang mengakibatkan robot akan bergerak ke kanan dan ke kiri, maka dari itu untuk mendapatkan hasil PD yang mendekati setpoint diperlukan penyesuaian pada nilai Kp dan Kd. Ketika Kp lebih besar dari Kd maka gerak troli akan menjadi tidak stabil maka dari itu untuk menstabilkan nilai respon

transien dari error Kp diperlukan perbesaran nilai Kd untuk memberi efek redaman osilasi dari kesalahan yang terjadi pada robot oleh nilai Kp.

Tabel 4. 11 Tabel Hubungan Antara Kp, Ki, dan Kd

Respon Closed-Loop	Rise Time	Overshoot	Settling Time	SS Error
Kp	Menurun	Meningkat	Perubahan kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Menghilangkan
Kd	Perubahan kecil	Menurun	Menurun	Perubahan kecil

4.6 Pengujian Kecepatan Troli

4.4.1 Pengujian Kecepatan Tanpa Beban

Pengujian dilakukan untuk mengukur kecepatan troli ketika tidak ada beban yang diletakkan di atasnya. Pada kondisi ini, troli hanya membawa beratnya sendiri dan komponen yang digunakan untuk membangunnya. Informasi terkait berat keseluruhan troli dan komponennya terdapat pada Tabel 4.9.

Tabel 4. 12 Berat Troli dan Komponen

No.	Komponen	Jumlah	Berat
1.	Troli	1	8 Kg
2.	Webcam Logitech C270	1	0,3 Kg
3.	Raspberry Pi 4B	1	0,65 Kg
4.	Arduino Mega 2560	1	0,037 Kg
5.	Sensor Ultrasonik HCSR04	3	0,03 Kg
6.	Motor Servo SG995	1	0,06 Kg
8.	Driver Motor BTS7960	2	0,1 Kg
9.	Modul Stepdown XL4015	1	0,02 Kg
10.	LCD Raspberry 5 Inch	1	0,125 Kg
11.	Modul MPU6050	1	0,01 Kg
12.	Baterai Aki	1	2 Kg
13.	Frame LCD	1	0,013 Kg
14.	Frame Servo	1	0,01 Kg
15.	Penjepit LCD dan Servo	2	0,6 Kg

No.	Komponen	Jumlah	Berat
16.	Box Akrilik	2	2 Kg
Jumlah Berat			13,955 Kg

Berdasarkan data yang disajikan, terlihat bahwa total berat dari semua komponen pada troli adalah 13,955 Kg. Pengujian dilakukan sebanyak tiga kali untuk mengukur kecepatan troli saat bergerak sejauh 1,5 meter. Pada jarak tersebut, waktu yang diperlukan oleh troli untuk menempuh jarak 1,5 meter diukur. Data waktu yang dihasilkan akan digunakan dalam rumus fisika yang sesuai.

Rumus fisika yang berlaku pada pengujian ini menyatakan bahwa jarak (s) berbanding lurus dengan waktu (t), dan kecepatan (v) adalah perbandingan jarak terhadap waktu. Kecepatan PWM pada motor DC yang digunakan adalah 153.5. Untuk mencari rata-rata kecepatan troli, hasil pengujian dapat dilihat pada Tabel 4.10.

Tabel 4. 13 Pengujian Kecepatan Tanpa Beban

Tabel 4.18 Pengujian Kecepatan Tanpa Bahan			
Percobaan Ke-	Jarak	Waktu	Kecepatan
1	150 cm	8,32 s	18,32 cm/s
2	150 cm	9,06 s	16,55 cm/s
3	150 cm	8,36 s	17,96 cm/s
Rata – Rata Kecepatan		8,58 s	17,61 cm/s

Selama pengujian, terdapat perbedaan dalam kecepatan troli yang disebabkan oleh beberapa faktor. Faktor pertama adalah ketidakstabilan pencahayaan selama pengujian yang mengakibatkan pembacaan deteksi tangan menjadi tidak stabil, sehingga motor tidak dapat berjalan dengan lancar. Faktor kedua adalah tidak rata permukaan lantai selama pengujian. Karena alasan ini, dilakukan tiga kali percobaan untuk mencari rata-rata kecepatan troli tanpa beban, yang pada akhirnya ditemukan rata-rata kecepatan sebesar 17,61 cm/s dengan rata-rata waktu tempuh jarak 150 cm sebesar 8,58 s. Setelah didapatkan data bahwa kecepatan tanpa beban

adalah 17,61 cm/s selanjutnya dilakukan pengukuran kecepatan dengan beban untuk melihat perbedaan yang terjadi antara dua kondisi.

4.4.2 Pengujian Kecepatan Dengan Beban

Dilakukan suatu pengujian untuk mengetahui kecepatan troli pada berbagai beban. Pengujian tersebut dilakukan sebanyak 6 kali dengan menggunakan beban yang berbeda. Troli dijalankan sejauh 2 meter dengan menggunakan PWM motor sebesar 105,5. Hasil dari pengujian tersebut kemudian direkam dan disajikan dalam bentuk tabel pada Tabel 4.11.

Tabel 4. 14 Hasil Pengujian Kecepatan Dengan Beban

No.	Berat Beban	Jarak	Waktu	Kecepatan
1	5 Kg	150 cm	9,26 s	16,19 cm/s
2	10 Kg	150 cm	9,77 s	15,35 cm/s
3	15 Kg	150 cm	11,15 s	13,45 cm/s
4	20 Kg	150 cm	13,12 s	11,43 cm/s
5	30 Kg	150 cm	15,64 s	9,59 cm/s
6	50 Kg	150 cm	20,08 s	7,47 cm/s

Hasil pengujian menunjukkan bahwa terdapat hubungan antara variasi kecepatan troli dengan beban yang diujikan. Ditemukan bahwa semakin berat beban yang diangkut oleh troli, kecepatan troli akan semakin lambat. Selain itu, pengaruh ketidakstabilan pencahayaan selama pengujian menyebabkan troli terkadang tidak dapat mendeteksi tangan, yang berdampak pada pencatatan kecepatan yang tercatat dalam Tabel 4.11.

4.7 Pengujian Masa Pakai Baterai

Pengujian lama pemakaian baterai dilakukan untuk mengetahui berapa lama baterai dapat bertahan saat troli digunakan. Pengujian dilakukan ketika baterai aki dalam kondisi *full* baterai sampai baterai aki habis. Pemakaian lama baterai tergantung pada daya yang digunakan oleh baterai aki. Tabel 4.12. menunjukkan penggunaan daya oleh beban dari baterai aki.

Tabel 4. 15 Tabel Daya Komponen

No.	Beban	Arus (A)	Tegangan (V)	Daya (Watt)
1.	Raspberry Pi 4B	0,9	5,02	4
2.	Arduino Mega	0,3	5,02	2
3.	LCD Raspberry 5 inch	0,46	5,02	2,3
4.	LCD I2C 20x4	0,2	5,02	1
5.	Motor Servo MG90	0,063	5,02	0,35
6.	Sensor Ultrasonik 1	0,002	5,02	0,01
7.	Sensor Ultrasonik 2	0,002	5,02	0,01
8.	Sensor Ultrasonik 3	0,002	5,02	0,01
9.	Sensor VL53L0X	0,004	5,02	0,014
10.	Driver BTS7960 1	1,1	13,2	14,52
11.	Driver BTS7960 2	1,1	13,2	14,52

1. Total Beban Saat Tidak Bergerak
 $= 4 \text{ W} + 2 \text{ W} + 2,3 \text{ W} + 1 \text{ W} + 0,35 \text{ W} + 0,01 \text{ W} + 0,01 \text{ W} + 0,01 \text{ W} + 0,014 \text{ W}$
 $= 9,6 \text{ W}$
2. Total Beban Saat Bergerak
 $= 4 \text{ W} + 2 \text{ W} + 2,3 \text{ W} + 1 \text{ W} + 0,35 \text{ W} + 0,01 \text{ W} + 0,01 \text{ W} + 0,014 \text{ W} + 14,52 \text{ W}$
 $= 14,52 \text{ W}$
 $= 38,73 \text{ W}$
2. Kapasitas Baterai
 $= V \times I$
 $= 12V \times 7,5 \text{ Ah}$
 $= 90 \text{ Wh}$
3. DoD Baterai
 $= \text{Kapasitas Baterai} \times 50\%$
 $= 90 \times 50\%$
 $= 45 \text{ Wh}$
4. Waktu Pemakaian Tanpa Bergerak
 $= \frac{\text{DoD Baterai}}{\text{Total Beban}} = \frac{45}{9,6} = 4,7 \text{ Jam}$

$$\text{5. Waktu Pemakaian} = \frac{\text{DoD Baterai}}{\text{Total Beban}} = \frac{45}{38,73} = 1,16 \text{ Jam}$$

Saat Bergerak



Gambar 4. 3 Total Arus Saat Tidak Bergerak



Gambar 4. 4 Total Arus Saat Bergerak

Tabel 4. 16 Penggunaan Lama Pakai Baterai Tanpa Beban

No.	Lama Pakai (menit)	Tegangan Baterai (v)	Persentase Baterai (%)
1.	1 menit	13,3V	99%
2.	10 menit	12,7V	86,4%
3.	25 menit	11,8V	73,8%
4.	35 menit	11,2V	63,8%
5.	45 menit	10,9V	56,7%
6.	55 menit	10,5V	31%

Tabel 4. 17 Penggunaan Lama Pakai Baterai Dengan Beban

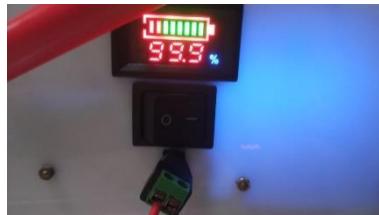
No.	Lama Pakai	Tegangan Baterai	Persentase Baterai
1.	1 menit	13,2V	99%
2.	10 menit	12,6V	84,2%
3.	25 menit	11,6V	62%
4.	35 menit	11,8V	46,8%
5.	47 menit	10,5V	31%

Dalam penghitungan, diperoleh bahwa baterai mampu bertahan selama 1,16 jam dengan Depth of Discharge sebesar 50%. Selain perhitungan, baterai juga diuji secara langsung dari kondisi penuh hingga habis saat digunakan pada troli tanpa beban. Percobaan pertama dilakukan dengan troli tanpa beban dan dari hasil pengujian tersebut, baterai mampu bertahan selama 55 menit ketika troli bergerak.

Pengujian kedua dilakukan dengan menjalankan troli sampai baterai habis berkisar 10,5V agar cycle life baterai tetap dikisaran aman. Pada awalnya baterai dalam keadaan penuh dengan tegangan sebesar 13,3V dan persentase baterai sebesar 99,9% seperti ditunjukkan pada Gambar 4.5 dan 4.6, baterai yang awalnya 99,9% pada indikator baterai berkurang menjadi 58% pada 25 menit pertama seperti ditunjukkan pada gambar 4.7. Uji terus dilakukan hingga berdurasi 47 menit berjalan terus-menerus, baterai habis pada tegangan 10,5V dengan persentase baterai sebesar 31% seperti pada Gambar 4.8 dan 4.9. Tujuan pengujian baterai dihentikan ketika baterai telah mencapai 10,5V adalah untuk menjaga siklus hidup baterai tetap aman dan baterai memiliki umur yang panjang.



Gambar 4. 5 Tegangan Baterai Saat Penuh



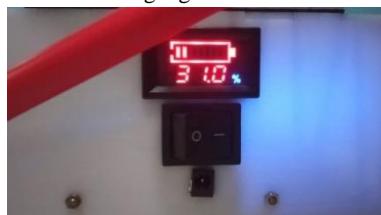
Gambar 4. 6 Persentase Baterai Saat Penuh



Gambar 4. 7 Persentase Baterai Saat 25 Menit



Gambar 4. 8 Tegangan Baterai Saat Habis



Gambar 4. 9 Persentase Baterai Saat Habis

4.8 Pengujian Sensor Ultrasonik dan ToF VL53L0X

Pengujian dilakukan untuk mengetahui pada jarak berapa troli akan berhenti ketika ada halangan, dilakukan pengujian terhadap sensor ultrasonik dan sensor laser. Sebanyak 3 buah sensor ultrasonik dan 1 buah sensor laser digunakan pada pengujian ini, yang diletakkan di bawah troli. Sensor ultrasonik diletakkan pada bagian depan, kanan, dan kiri troli, sementara sensor laser diletakkan pada bagian belakang bawah troli, agar troli tidak menabrak benda di depan, kanan, kiri, atau pengguna di belakang. Sensor digunakan untuk menghentikan dan memundurkan troli saat ada halangan yang terdeteksi pada jarak kurang dari 20cm dari posisi troli, dilakukan pemrograman untuk sensor ultrasonik dan sensor laser. Jarak ini berlaku untuk ketiga sensor ultrasonik dan sensor laser. Hasil

pengujian sensor ultrasonik dan sensor laser dapat dilihat pada Gambar 4.10 dan Tabel 4.15.

Tabel 4. 18 Jarak Ultrasonik dan Laser Pada Halangan

Jarak Sesungguhnya	Jarak Terdeteksi Sensor				Hasil Deteksi
	Ultrasonik A	Ultrasonik B	Ultrasonik C	Time Of Flight	
10 cm	10	10	11	15	Terdeteksi
20 cm	20	21	20	23	Terdeteksi
30 cm	31	31	30	32	Terdeteksi
40 cm	40	41	40	41	Terdeteksi



Gambar 4. 10 Tampilan Jarak Sensor Pada LCD

Gambar 4.10 menampilkan beberapa tampilan indikator yang menggambarkan hasil pengujian sensor ultrasonik dan sensor laser. Pada indikator tersebut, terdapat label "jarak" yang digunakan untuk menunjukkan hasil pengukuran jarak antara troli dengan benda yang terdeteksi oleh sensor. Layar LCD menampilkan angka-angka 26, 32, 58, dan 99, yang mewakili jarak antara troli dan benda yang terdeteksi oleh sensor ultrasonik pada sisi kanan (26 cm), sisi depan (32 cm), sisi kiri (58 cm), dan sisi belakang menggunakan sensor laser (99 cm). Ketiga nilai jarak ini digunakan untuk memungkinkan troli menghindari halangan saat bergerak.

4.9 Pengujian Keseluruhan Alat

Pengujian dimulai dari kondisi alat yang mati. Untuk menghidupkan perangkat listrik pada troli, pengguna harus merubah kondisi tombol saklar dari posisi off ke on. Tombol saklar terletak di sisi kanan bawah bagian troli. Setelah tombol saklar dinyalakan, persentase

baterai akan ditampilkan di atas tombol saklar. Gambaran dari tombol saklar dapat ditemukan pada Gambar 4.11.



Gambar 4. 11 Indikator Baterai Menyala

Selanjutnya sistem arduino akan memulai proses inisialisasi yang berlangsung berkisar 20 detik, Saat proses inisialisasi sistem selesai akan muncul informasi lcd arduino, sedangkan untuk inisialisasi raspberry memerlukan waktu sekitar 42 detik. Saat inisialisasi raspberry selesai maka lcd display akan mulai menampilkan display dan membuka output kamera seperti pada Gambar 4.12.



Gambar 4. 12 Tampilan LCD Saat Sistem Sudah Siap

Setelah sistem dinyalakan, sistem arduino lcd akan menampilkan beberapa informasi seperti nilai kirim raspberry, angle mpu6050, kondisi pergesure dan jarak sensor. Setelah menyala raspberry perlu melakukan booting komputer dan menampilkan lcd *display* Raspberry. Sistem telah siap ketika lcd display sudah membuka kamera seperti pada gambar 4.13.



Gambar 4. 13 Sistem Siap Menerima Input

Sistem telah siap ketika lcd display telah membuka kamera seperti pada Gambar 4.13, kita bisa langsung memberikan perintah pada troli. Servo akan mengikuti objek ketika terdapat objek tepat berada didepan kamera dengan respon servo berkisar 2 detik untuk pergesture. Saat kita memberikan *hand gesture* maju maka troli akan bergerak lurus seperti pada Gambar 4.14. Saat kita memberikan *hand gesture* mundur maka troli akan bergerak mundur seperti pada Gambar 4.15. Saat kita memberikan *hand gesture* kiri, maka troli akan berbelok ke kiri seperti pada Gambar 4.16. Saat kita memberikan *hand gesture* kanan, maka troli akan berbelok ke kanan seperti pada Gambar 4.17. Saat kita memberikan *hand gesture* kanan, maka troli akan berhenti seperti pada Gambar 4.18.



Gambar 4. 14 Troli Bergerak Maju



Gambar 4. 15 Troli Bergerak Mundur



Gambar 4. 16 Troli Bergerak Ke Kiri



Gambar 4. 17 Troli Bergerak Ke Kanan



Gambar 4. 18 Troli Berhenti Bergerak

Apabila terdapat rintangan yang berjarak kurang dari 20 cm di depan troli, maka troli akan otomatis berhenti. Jika sensor ultrasonik mendeteksi rintangan di sebelah kanan troli, maka troli akan berbelok ke kiri, begitu pula sebaliknya apabila rintangan terdeteksi di sebelah kiri, troli akan berbelok ke kanan seperti pada Gambar 4.19.



Gambar 4. 19 Respon Ultrasonik

Hasil pengujian menunjukkan bahwa troli dapat bergerak maju, belok ke kanan atau kiri, serta berputar mengikuti arah gesture objek warna. Webcam berfungsi dengan baik dalam membaca objek *hand gesture*. Sensor dan motor beroperasi sesuai dengan program yang dijalankan. Kendati demikian, intensitas cahaya yang tidak stabil dapat mengganggu pembacaan webcam dan mengakibatkan troli kehilangan deteksi objek sehingga tidak dapat bergerak.

4.10 Analisa Seluruh Sistem

Berdasarkan hasil percobaan dan pengujian yang telah dilakukan, telah didapatkan analisa bahwa :

1. Raspberry Pi 4B memerlukan waktu total 42 detik untuk menyala dan mulai beroperasi setelah dimatikan. Waktu tersebut digunakan untuk menyalakan LCD dan mengaktifkan kamera.
2. Pengujian deteksi objek *hand gesture* berhasil dilakukan dengan baik. Pada saat pengujian, terdapat 5 gesture *hand gesture* yang dapat dikenali, yaitu maju, mundur, kanan, kiri, dan berhenti. Objek *hand gesture* dapat terdeteksi dengan baik oleh kamera, ditandai dengan adanya kotak persegi putih (tracking bbox) di sekitar tangan yang terdeteksi. Intensitas cahaya yang cukup, dengan nilai minimal 10 lux, jarak objek dan pencahayaan *hand gesture* yang ideal terhadap kamera sangat mempengaruhi hasil pendekripsi objek *hand gesture* apabila tidak memenuhi troli akan bergerak namun tersendat.
3. Hasil pengujian jarak kamera menunjukkan bahwa kamera dapat mendeteksi objek pada jarak maksimal 650cm dan minimal 25cm. Namun, setelah jarak mencapai 650cm, kamera kehilangan

- kemampuan deteksinya. Jarak yang ideal untuk deteksi adalah antara 30-150cm.
4. Pengujian masa pakai baterai didapatkan data bahwa baterai dengan batas *discharge* hingga 10,5V dapat bertahan 50 menit tanpa beban sedangkan 47 menit dengan beban 5Kg. Kelemahan baterai saat ini mudah habis saat sudah mencapai penggunaan 30 menit.
 5. Hasil percobaan pengujian kecepatan menunjukkan bahwa kecepatan troli tanpa beban adalah 17,61 cm/s. Ketika troli diberi beban seberat 20 Kg, kecepatannya menurun menjadi 11,43 cm/s, dan ketika beban ditambah menjadi 50 Kg, kecepatannya semakin melambat menjadi 7,47 cm/s dan tidak kuat bergerak karena pwm rendah. Dengan demikian, dapat disimpulkan bahwa semakin berat beban yang dibawa oleh troli, maka kecepatannya akan semakin lambat.
 6. Setelah melakukan pengujian, ditemukan bahwa sensor ultrasonik berfungsi dengan baik. Ketika sensor ultrasonik bagian depan mendeteksi halangan pada jarak kurang dari 20 cm, maka troli akan berhenti dan mundur hingga posisi sebelumnya tidak tercapai. Apabila sensor ultrasonik bagian kiri atau kanan mendeteksi halangan pada jarak kurang dari 20 cm, maka troli akan berbelok ke arah yang sesuai. Selain itu, jika sensor laser mendeteksi keberadaan pengguna pada jarak kurang dari 20 cm, troli akan bergerak maju hingga posisi sebelumnya tidak tercapai.
 7. Hasil uji coba menunjukkan bahwa motor servo beroperasi dengan baik dan mampu mengikuti gesture objek *hand gesture* yang terdeteksi. Ketika objek berada pada posisi sudut 0° atau 180° dari posisi awal servo, motor servo akan bergerak untuk menyesuaikan posisi objek sehingga berada di tengah-tengah titik fokus kamera. Kelemahan juga ditemukan ketika sistem dihidupkan, servo terkadang memutar 180° sehingga objek menjadi menghilang.
 8. Pengujian gerak lurus troli menunjukkan kinerja yang baik. Rata-rata sudut dari 10 percobaan adalah $89,6^\circ$ dengan tingkat error sebesar 1,98%. Kesalahan tersebut disebabkan oleh kebutuhan roda depan untuk memutar arah sebelum bergerak maju setelah bergerak mundur.

Secara umum, sistem telah berfungsi dengan baik sesuai dengan rencana yang telah disusun. Namun, kelemahan dan kekurangan yang ada dapat dijadikan sebagai modal dan bahan penelitian untuk pengembangan lebih lanjut

~Halaman ini sengaja dikosongkan~

BAB V

PENUTUP

Dari seluruh proses perencanaan, pembuatan, dan analisis yang dilakukan pada Tugas Akhir ini, terdapat kesimpulan dan saran yang dapat diambil.

5.1 Kesimpulan

Sistem robot troli yang menggunakan kamera dan Raspberry Pi untuk mengenali gesture tangan telah terbukti efektif sesuai dengan tujuan yang telah disusun sebelumnya. Sistem ini dapat dengan tepat mendeteksi dan robot dapat bergerak sesuai perintah *hand gesture* seperti maju, mundur, kanan, kiri, dan berhenti dengan tingkat akurasi yang memadai. Namun, ada beberapa faktor yang mempengaruhi pendektsian objek, seperti intensitas cahaya dan jarak objek terhadap webcam. Webcam yang digunakan berhasil membaca objek *hand gesture* dengan baik, sehingga troli dapat mengikuti perintah *hand gesture* tersebut.

Pendeteksian *hand gesture* memiliki kemampuan untuk mendeksi objek *hand gesture* pada jarak minimal 25 cm hingga jarak maksimal 650 cm dengan intensitas cahaya ideal adalah 10 lux dengan kontras cahaya memadai. Selama proses pendektsian objek, rata-rata respon error servo tercatat sebesar 1,77%. Robot troli memiliki kecepatan rata-rata troli saat tidak ada beban adalah 17,61 cm/s. Percobaan menunjukkan bahwa baterai dengan batas discharge 10,5V dapat bertahan selama 55 menit tanpa beban dan 47 menit dengan beban 5Kg.

Temuan ini dapat digunakan sebagai landasan untuk penelitian dan pengembangan sistem robot troli yang lebih efisien, dengan memperbaiki kekurangan dan kelemahan yang ditemukan. Robot troli ini dapat digunakan untuk memudahkan pelanggan saat berbelanja, sehingga pelanggan tidak perlu lagi mendorong troli secara manual.

5.2 Saran

Disarankan untuk melakukan pengembangan sistem ini dengan memperbaiki baik dari segi fungsi maupun implementasi. Dalam rangka memperbaiki kekurangan - kekurangan yang ada, merekomendasikan beberapa saran sebagai berikut.

1. Untuk meningkatkan kecepatan dan kapasitas beban, dapat digunakan motor yang lebih kuat

2. Dalam penggunaan sensor ultrasonik, perlu memperhatikan penempatan dan sudut sensor untuk mendapatkan hasil yang lebih akurat.
3. Dalam pengembangan selanjutnya, diperlukan penggantian roda depan dengan roda roller agar dapat bermanuver dengan baik.
4. Dalam pengembangan selanjutnya, dapat dilakukan integrasi dengan teknologi lain seperti pengenalan suara atau face recognition untuk memperluas fungsionalitas robot troli.
5. Dalam pembuatan dan pengembangan robot troli, perlu memperhatikan aspek jenis servo, kapasitas baterai serta posisi kamera dan HMI yang ideal.

DAFTAR PUSTAKA

- [1] I. Febri, T. Setya, and F. Sadika, “Perancangan Troli Mandiri Untuk Pengunjung di Pasar Modern (Studi Kasus : Pasar Modern Batununggal Indah),” *e-Proceeding Art Des.*, vol. 7, no. 2, pp. 5642–5654, 2020.
- [2] J. S. Putra, “Pengembangan Robot Troli,” *Semin. Has. Elektro SI ITN Malang*, pp. 2–13, 2019.
- [3] M. Lovita and R. Mukhaiyar, “Rancang Bangun Troli Otomatis Berbasis Computer Vision,” *JTEIN J. Tek. Elektro Indones.*, vol. 3, no. 2, pp. 399–406, 2022.
- [4] F. I. Pasaribu and S. Yogen, “Perancangan Prototype Troli Pengangkut Barang Otomatis Mengikuti Pergerakan Manusia,” *RELE (Rekayasa Elektr. dan Energi) J. Tek. Elektro*, vol. 1, no. 2, pp. 82–92, 2019, doi: 10.30596/rele.v1i2.3011.
- [5] F. S. Anggreyani and E. Alimudin, “Shopping Trolley Automatically Move Following Color With Camera,” *Politek. Negeri Cilacap*, 2022.
- [6] S. Muhamrom, S. Asnawi, and A. Bachri, “Robot Pengikut Target Berdasarkan Bentuk dan Warna Menggunakan Metode HSV Untuk Aplikasi Assistant Robot,” *JE-Unisla*, vol. 6, no. 1, p. 415, 2021, doi: 10.30736/je.v6i1.571.
- [7] F. S. Samidi*, I. S. Mustafa, and N. A. M. Radzi, “An Intelligent Following Sensor Shopping Cart,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 6249–6255, 2019, doi: 10.35940/ijrte.d5103.118419.
- [8] K. Ingole and S. R. Khedkar, “Automatic Human Following Trolley Using Raspberry Pi,” *Int. Res. J. Eng. Technol.*, vol. 08, no. 07, pp. 1558–1563, 2021, [Online]. Available: www.irjet.net
- [9] K. Sekar, V. Rajkumar, R. Thileeban, and S. Sudharsan, “Hand Gesture Controlled Robot,” *Int. J. Eng. Res. Technol.*, vol. 9, no. 11, pp. 17–19, 2020, doi: 10.1007/978-981-33-4909-4_64.
- [10] B. Prasetyo, K. Exaudi, and S. Sembiring, “Implementasi Gerakan Tangan terhadap Navigasi Robot Beroda menggunakan Teknik Accelerometer,” *Emit. J. Tek. Elektro*, vol. 21, no. 2, pp. 142–147, 2021, doi: 10.23917/emit.v21i2.14228.
- [11] S. B. Bhaskoro and M. A. A. Aziz, “Pengendalian Gerak Robot

- menggunakan Semantik Citra Gestur Tangan Manusia,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 8, no. 1, p. 80, 2020, doi: 10.26760/elkomika.v8i1.80.
- [12] A. Nur, F. Mappasaile, T. Z. Muttaqien, and Y. Pujiraharjo, “Perancangan Troli Belanja Dengan Sekat Barang Belanjaan (Studi Kasus : Transmart Carrefour Buah Batu Bandung) Shopping Trolley Design With Groceries Partition (Case Study : Transmart Carrefour Buah Batu Bandung),” *e-Proceeding Art Des.*, vol. 7, no. 2, pp. 5634–5641, 2020.
- [13] A. D. Pratiwi and T. S. Pembudi, “Perancangan Trolley Belanja Mandiri Untuk Kebutuhan Belanja Mingguan Di Pasar Modern (Studi Kasus: Pasar Modern Batununggal Indah).,” *e-Proceeding Art Des.*, vol. 7, no. 2, pp. 5673–5682, 2020.
- [14] J. Jumadi, Y. Yupianti, and D. Sartika, “Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering,” *JST (Jurnal Sains dan Teknol.*, vol. 10, no. 2, pp. 148–156, 2021, doi: 10.23887/jstundiksha.v10i2.33636.
- [15] D. D. Affifah, Y. Permanasari, and R. Respitawulan, “Teknik Konvolusi pada Deep Learning untuk Image Processing,” *Bandung Conf. Ser. Math.*, vol. 2, no. 2, pp. 103–112, 2022.
- [16] S. Y. Damanik and K. Tampubolon, “Pengenalan Pola Citra Simbol Bahan Kimia Berbahaya menggunakan Metode Transformasi Wavelet,” *MEANS (Media Inf. Anal. dan Sist.)*, vol. 6, no. 2, pp. 112–116, 2021, doi: 10.54367/means.v6i2.1765.
- [17] T. Susim and C. Darujati, “Pengolahan Citra Untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV,” *J. Syntax Admiration*, vol. 2, no. 3, pp. 534–545, 2021.
- [18] A. A. Chellsya *et al.*, “Implementasi Computer Vision Dalam Mendeteksi Pelanggaran Tidak Menggunakan Helm Pada Pengendara Motor,” *e-Proceeding Appl. Sci.*, vol. 9, no. 1, pp. 55–71, 2023.
- [19] H. Septyan Nu, Y. Sofyan, and A. Rafi Al Tahtawi, “Pengendalian Robot Lengan Pemilah Benda Berdasarkan Bentuk Menggunakan Teknologi Computer Vision,” *SEMMASTERA (Seminar Nas. Teknol. dan Ris. Ter.)*, vol. 2, no. 0, pp. 42–48, 2020.
- [20] M. A. Masril and D. P. Caniago, “Optimasi Teknologi Computer

- Vision pada Robot Industri Sebagai Pemindah Objek Berdasarkan Warna,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 1, pp. 46–57, 2023.
- [21] Oguntimilehin and Olatunji, “A Review of Sign Language Recognition Techniques,” *Int. J. Inf. Syst. Comput. Sci.*, vol. 10, no. 6, p. 2021, 2021.
- [22] Y. Liu, M. Yang, J. Li, Q. Zheng, and D. Wang, “Dynamic hand gesture recognition using 2d convolutional neural network,” *Eng. Lett.*, vol. 28, no. 1, pp. 243–254, 2020.
- [23] M. J. Hussain *et al.*, “Intelligent Sign Language Recognition System for E-Learning Context,” *Comput. Mater. Contin.*, vol. 72, no. 3, pp. 5327–5343, 2022, doi: 10.32604/cmc.2022.025953.
- [24] M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” *J. Imaging*, vol. 6, no. 8, pp. 2–29, 2020, doi: 10.3390/JIMAGING6080073.
- [25] S. Budiman, S. Lestanti, S. M. Evandri, and R. K. Putri, “Pengenalan Gestur Gerakan Jari Untuk Mengontrol Volume Di Komputer Menggunakan Library OpenCV Dan MediaPipe,” *J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 223–232, 2022.
- [26] F. Zhang *et al.*, “MediaPipe Hands: On-device Real-time Hand Tracking,” 2020, [Online]. Available: <http://arxiv.org/abs/2006.10214>
- [27] I. Suyudi, S. Sudadio, and S. Suherman, “Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression (Introduction to Indonesian Sign Language Using Mediapipe With Random Forest Models and Multinomial Logistic Regression),” *J. Ilmu Siber dan Teknol. Digit.*, vol. 1, no. 1, pp. 65–80, 2022.
- [28] J. D. Asmoro, A. T. Wibowo, And M. Ridwan, “Virtual Mouse With Hand Gesture Recognition Based On,” *Jurteksi (Jurnal Teknol. Dan Sist. Informasi)*, Vol. IX, No. 2, Pp. 261–268, 2023.
- [29] R. Mohamed Abdulhamied, M. M. Nasr, and S. N. Abdul Kader, “Real-time recognition of American sign language using long-short term memory neural network and hand detection,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 1, p. 545, 2023, doi: 10.11591/ijeecs.v30.i1.pp545-556.
- [30] M. P. Pulungan, M. B. A. Rachman, and ..., “Identifikasi Warna

- Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV,” *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, no. April, pp. 279–289, 2022, [Online]. Available: <https://conference.upnvj.ac.id/index.php/senamika/article/view/2011%0Ahttps://conference.upnvj.ac.id/index.php/senamika/article/download/2011/1588>
- [31] I. W. Sukerta Wijaya, I. G. Harjumawan Wiratmaja KS., I. D. M. A. Pramana Setya Bintara, and I. K. G. Ryan Aditya Permana, “Program Menghitung Banyak Bata pada Ruangan Menggunakan Bahasa Python,” *TIERS Inf. Technol. J.*, vol. 2, no. 1, 2021, doi: 10.38043/tiers.v2i1.2840.
- [32] F. B. Setiawan, O. J. Aldo Wijaya, L. H. Pratomo, and S. Riyadi, “Sistem Navigasi Automated Guided Vehicle Berbasis Computer Vision dan Implementasi pada Raspberry Pi,” *J. Rekayasa Elektr.*, vol. 17, no. 1, pp. 7–14, 2021, doi: 10.17529/jre.v17i1.18087.
- [33] M. Fajar And A. Rahman, “Rancang Bangun Alat Pencampur Bahan Es Krim Berbasis Arduino Mega2560,” *J. J-Ensitec*, Vol. 05, No. 02, Pp. 271–277, 2019.
- [34] I. Y. Lonteng, I. Rosita, M. Simulasi, And M. Jarak, “Rancang Bangun Simulasi Alat Pendekripsi Jarak Aman Antar Kendaraan Menggunakan Sensor Ultrasonik Berbasis Arduino,” *Jeecon*, Vol. 2, No. 2, Pp. 22–26, 2020.
- [35] S. Oky, “Perancangan Robot Avoider Berbasis Arduino Uno Menggunakan Tiga Sensor Ultrasonik,” *Epic J. Electr. Power, Instrum. Control*, vol. 1, no. 2, pp. 1–11, 2018, doi: 10.32493/epic.v1i2.1529.
- [36] S. Rakasiwi, H. Kusumo, and A. Cahyo Pangestu, “Sistem Presensi Karyawan Menggunakan Raspberry Dengan Sensor Fingerprint Dan Webcam,” *J. Tek. Inform. dan Teknol. Inf.*, vol. 2, no. 2, pp. 75–83, 2022, doi: 10.55606/jutiti.v2i2.372.
- [37] E. D. Arisandi, “Controlling Multi Servo with Single Chip FPGA,” *AIP Conf. Proc.*, vol. 2542, no. 1–5, pp. 5–10, 2022, doi: 10.1063/5.0105126.
- [38] W. Sukadana, N. Agus, and W. Kusuma, “Rancang Bangun Pengendali Kursi Roda Menggunakan Mikrokontroler Atmega328 Berbasis Aplikasi Mobile,” *Semin. Nas. Peran. Iptek Menuju Ind. Masa Depan*, vol. 5, pp. 85–92, 2019, doi: 10.21063/PIMIMD5.2019.12.

- [39] T. Prilian, I. Rusmana, and T. Handayani, “Kursi Roda Elektrik dengan Kendali Gestur Kepala,” *Avitec*, vol. 3, no. 1, pp. 29–41, 2021, doi: 10.28989/avitec.v3i1.830.
- [40] A. H. Zaidan, M. K. Wail, and A. A. Yaseen, “Improving the Efficiency of the Wireless Robotic Hand by Controlling Sensory Variables,” *AIP Conf. Proc.*, vol. 2398, pp. 0–16, 2022, doi: 10.1063/5.0093385.
- [41] J. Shaik, “Smart Trolley Shopping for Automatic Billing & Assistance for Visually Impaired,” *J. Eng. Sci.*, vol. 14, no. 04, pp. 155–164, 2023.
- [42] O. B. Kharisma, A. Wildan, and F. E. Laumal, “Implementasi Sensor MPU 6050 untuk Mengukur Kesetimbangan Self Balancing Robot Menggunakan Kontrol PID,” *Semin. Nas. Teknol. Informasi, Komun. dan Ind.*, no. November, pp. 357–364, 2018.
- [43] STMicroelectronics, “VL53L0X Datasheet,” pp. 1–38, 2022.
- [44] Waveshare, “5inch HDMI LCD,” 2019.
- [45] S. M. B. Respati, A. Nugroho, And Udiawan, “Separator Serbuk Zeolit Sebagai Aki Baterai Dengan Efek Perbedaan Cairan H₂so₄, Air Cuka, Dan Nacl,” *Pros. Semin. Nas. Sains Dan Teknol.*, Vol. 12, No. 1, Pp. 98–103, 2022.

LAMPIRAN A

Listing Program Arduino

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
#include <Wire.h>
#include <ArduinoJson.h>
DynamicJsonDocument doc(2048);

int xcenter, ycenter, kondisi;
char com = '0';

//MOTOR
int RPWM1 = 9;
int LPWM1 = 8;
int LPWM2 = 4;
int RPWM2 = 5;

int pwmmmin = 30;
int pwmstop = 0;
int pwmmax = 50;
int draw = 30;
int pwm = 30;
int errorar;
int arah, arahrob;
String stss = "null";

//ULTRASONIC
#include <Ultrasonic.h>
Ultrasonic ultrasonicA(46, 47); //trig,echo
Ultrasonic ultrasonicB(48, 49);
Ultrasonic ultrasonicC(50, 51);
int usA, usB, usC;
int pos;

//ULX530 Sensor
#include <VL53L0X.h>
```

```

VL53L0X sensor;
int ul;

//Servo
#include <Servo.h>
Servo servoX;
Servo servoY;
int x = 90 ;
int y = 0 ;

//MPU6050
float target = 0;
float error = 0;
float derivative = 0;
float last_error = 0;

//the 'k' values are the ones you need to fine tune before your program
will work. Note that these are arbitrary values that you just need to
experiment with one at a time.
float Kp = 11.2;
float Kd = 12;
float pwm1 = 0;
float pwm2 = 0;

int mtrSpd = 45.5;
float angle = 0;

//Variables for Gyroscope
#include "I2Cdev.h"
#define MPU6050_ADDR 0x68 // Alternatively set AD0 to HIGH -->
Address = 0x69
#define MPU6050_RA_GYRO_ZOUT_H 0x47
float gyroZ; // Raw register values gyroscope
unsigned long lastTime = 0;
float dt; //Differential time
long gyroZ0 = 0; //Gyro offset = mean value
float yaw;

```

```
float yawOld = 0;
float target_yaw = 0;
float gyroAngleZ = 0; //Angle variable
unsigned long sebelum = 0;
void setup() {
    lcd.begin();
    lcd.backlight();
    lcd.clear();
    sensor.init();
    sensor.startContinuous();
    Serial.begin(9600);
    servoX.attach(2);
    servoY.attach(3);
    motor();
    Wire.begin();
    mpubegin();
}
void loop() {
    calcYaw();
    paket();
    ultrasonik();
    pantilt();
    gerak();

    lcd.setCursor(0, 0);
    lcd.print("JSON:");
    lcd.print("{\" " + String(xcenter) + "," + String(ycenter) + "," +
String(kondisi) + " }");

    lcd.setCursor(0, 1);
    lcd.print("Angle : ");
    lcd.print(angle);

    lcd.setCursor(0, 2);
    lcd.print("kondisi : ");
    lcd.print(stss);
```

```

lcd.setCursor(0, 3);
String us = ('{' + String(usA) + ',' + String(usB) + ',' + String(usC) + ','
+ String(ul) + "}"    );
lcd.print("jarak:" );
lcd.print(us);
}

void flow() {
if (kondisi != 404) {
  if (kondisi == 77) {
    stss = "Berhenti ";
    waithere();
  }

  if (kondisi == 66) {
    stss = "Mundur  ";
    backward();
    pwmmmin += 3;
    if (pwmmmin > pwmmmax) pwmmmin = pwmmmax;
  }

  if (kondisi == 55) {
    stss = "Kanan  ";
    right();
    target_yaw = yawOld;
    pwmmmin += 1;
    draw -= 1;
    if (draw < 5) draw = 5;
    if (pwmmmin > pwmmmax) pwmmmin = pwmmmax;
  }

  if (kondisi == 33) {
    stss = "Maju  ";
    error = yawOld - yaw;// proportional
  }
}
}

```

```

derivative = error - last_error; //derivative
angle = (error * Kp) + (derivative * Kd);

target_yaw = yaw;
if (target_yaw > yawOld) {
    pwm1 = mtrSpd - abs(angle); //kiri
    pwm2 = mtrSpd + abs(angle); //kanan
    forward();
}
//setting the steering command if it is veering to the left
else if (target_yaw < yawOld) {
    pwm1 = mtrSpd + abs(angle); //kiri
    pwm2 = mtrSpd - abs(angle); //kanan
    forward();
}
error = last_error;
}

if (kondisi == 44) {
    stss = "Kiri    ";
    left();
    target_yaw = yawOld;
    pwmmmin += 1;
    draw -= 1;
    if (draw < 5) draw = 5;
    if (pwmmmin > pwmmmax) pwmmmin = pwmmmax;
}
}
}

void motor(){
pinMode(RPWM1, OUTPUT);
pinMode(LPWM1, OUTPUT);
pinMode(RPWM2, OUTPUT);
pinMode(LPWM2, OUTPUT);
}

```

```
void forward(){
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, pwm2);
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, pwm1);//kiri
}

void backward(){
    analogWrite(RPWM1, pwmmmin);
    analogWrite(LPWM1, 0);
    analogWrite(RPWM2, pwmmmin);
    analogWrite(LPWM2, 0);
}

void left(){
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, draw);
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, pwmmmin);
}

void right(){
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, pwmmmin);
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, draw);
}

void waithere(){
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, 0);
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, 0);
}

void mpubegin(){
    Wire.begin();
}
```

```

Wire.beginTransmission(MPU6050_ADDR);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // wake up!
Wire.endTransmission(true);
delay(100);
calibration();
}

int16_t getRotationZ() {
    uint8_t buffer[14];
    I2Cdev::readBytes(MPU6050_ADDR,
MPU6050_RA_GYRO_ZOUT_H, 2, buffer);
    return (((int16_t)buffer[0]) << 8) | buffer[1];
}

void calibration()
{
    unsigned short times = 100; //Sampling times
    for (int i = 0; i < times; i++)
    {
        gyroZ = getRotationZ();
        gyroZ0 += gyroZ;
    }
    gyroZ0 /= times;
}

void calcYaw() {

    unsigned long currentTime = millis();
    dt = (currentTime - lastTime) / 1000.0;
    lastTime = currentTime;

    gyroZ = getRotationZ();

    float angularZ = (gyroZ - gyroZ0) / 131.0 * dt; //angular z: =t
    if (fabs(angularZ) < 0.05) //
    {

```

```

angularZ = 0.00;
}
gyroAngleZ += angularZ; //returns the absolute value of the z-axis
rotazion integral
yaw = - gyroAngleZ;
}

String dataT = "";

void paket() {
if (Serial.available()) {
dataT = Serial.readStringUntil("\n");
deserializeJson(doc, dataT);
xcenter = doc["xc"].as<int>();
ycenter = doc["yc"].as<int>();
kondisi = doc["k"].as<int>();
}
}

void pantilt() {
if (xcenter != 404) {
if (xcenter < 300 && xcenter > 1) {
com = '1';
}
else if (xcenter > -300 && xcenter < -1) {
com = '2';
}
switch (com) {
case '1':
a();
delay(10);
break;
case '2':
b();
delay(10);
break;
default:
}
}
}

```

```

        servoX.write(90);
        break;
    }
}
else {
    com = '0';
}
}
void a() {
    servoX.write(x);
    x -= 1;
}
void b() {
    servoX.write(x);
    x += 1;
}
void ultrasonik() {
    usA = ultrasonicA.read();
    usB = ultrasonicB.read();
    usC = ultrasonicC.read();
    ul = sensor.readRangeContinuousMillimeters() / 10;

}

void gerak() {
    if (usB <= 15) {
        waithere();
        for (int c = 0; c < 100; c++) {
            delay(1);
        }
        backward();
        pwmmmin = 70;
        for (int c = 0; c < 1000; c++) {
            delay(1);
        }
        waithere();
    }
}

```

```

else if (ul <= 20) {
    pwm1 = 30;
    pwm2 = 30;
    forward();
    for (int c = 0; c < 1000; c++) {
        delay(1);
    }
    waithere();
}
else if (usC <= 15) {
    draw = 0;
    pwmmmin = 50;
    left();
    for (int c = 0; c < 1000; c++) {
        delay(1);
    }
    waithere();
}
else if (usA <= 15) {
    draw = 0;
    pwmmmin = 50;
    right();
    for (int c = 0; c < 1000; c++) {
        delay(1);
    }
    waithere();
}
else if (usA <= 15 && usC <=15) {
    waithere();
    for (int c = 0; c < 100; c++) {
        delay(1);
    }
    backward();
    pwmmmin = 70;
    for (int c = 0; c < 1000; c++) {
        delay(1);
    }
}

```

```
    }
    waithere();
}
else {
    flow();
}
}
```


LAMPIRAN B

Listing Program Phyton

```
import cv2
import mediapipe as mp
import time
import serial
import json

useSerial = True
if useSerial == True :
    ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1.11)
    time.sleep(2)

class dataObject():
    def __init__(self, mode=False, maxHands=1, modelComplexity=1,
detectionCon=0.5, trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.modelComplex = modelComplexity
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
self.modelComplex,
                                         self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils

def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    if self.results.multi_hand_landmarks:
        for handLms in self.results.multi_hand_landmarks:
            if draw:
```

```

        self.mpDraw.draw_landmarks(img, handLms,
                                    self.mpHands.HAND_CONNECTIONS)
    return img

def findPosition(self, img, handNo=0, draw=True, flipType = True):
    xList = []
    yList = []
    bbox = []
    lmList = []
    b = []
    if self.results.multi_hand_landmarks:
        palm = self.results.multi_hand_landmarks[handNo]
        for id, lm in enumerate(palm.landmark):
            # print(id, lm)
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            xList.append(cx)
            yList.append(cy)
            cv2.putText(img, str('Tangan Terdeteksi'), (10, 120),
cv2.FONT_HERSHEY_PLAIN,1,
(0, 120, 205),2)
            lmList.append([id, cx, cy])

            xmin, xmax = min(xList), max(xList)
            ymin, ymax = min(yList), max(yList)
            bbox = xmin, ymin, xmax, ymax
            bx, by = (bbox[0]-20 + bbox[2] + 20)//2, (bbox[3] + 20 + bbox[1]
- 20)//2
            b = [bx, by]
            b.append(0)
            if draw:
                cv2.rectangle(img, (xmin - 20, ymin - 20), (xmax + 20, ymax
+ 20),
(255, 255, 255), 2)

        else:

```

```

        cv2.putText(img, str('Tidak Terdeteksi'), (10, 120),
cv2.FONT_HERSHEY_PLAIN,1,
(0, 120, 205),2)
return lmList,bbox,b

def main():
    mode = ""
    time.sleep(1)
    pTime = 0
    cTime = 0
    width, height = 640, 480
    cap = cv2.VideoCapture(0, cv2.CAP_V4L)
    cap.set(3, width)
    cap.set(4, height)
    detected = dataObject()
    k = 0
    while True:
        success, img = cap.read()
        img = detected.findHands(img)
        lmList,bbox,b = detected.findPosition(img)

        height, width = img.shape[:2]
        fingers=[]
        tipIds = [4, 8, 12, 16, 20]

        if len(lmList) != 0:
            x1, y1 = lmList[4][1:]
            x2, y2 = lmList[20][1:]
            gx = b[0] - width // 2
            gy = b[1] - height // 2

            cv2.line(img, (0, b[1]), (width, b[1]), (0, 0, 0), 1)
            cv2.line(img, (b[0], 0), (b[0], height), (0, 0, 0), 1)
            cv2.circle(img, (b[0], b[1]), 5, (0, 0, 0), cv2.FILLED)

            if lmList[tipIds[0]][1] > lmList[tipIds[0]-1][1]:
                fingers.append(1)

```

```

else:
    fingers.append(0)
for id in range(1,5):
    if lmList[tipIds[id]][2] < lmList[tipIds[id]-2][2]:
        fingers.append(1)
    else:
        fingers.append(0)
total=fingers.count(1)

if total == 5:
    cv2.putText(img, str("Mundur"), (10, 90),
cv2.FONT_HERSHEY_PLAIN,2,
(0, 170, 255),3)
k = 66

elif (fingers == [0,0,0,0,0] and x1 < x2):
    cv2.putText(img, str("Kanan"), (10, 90),
cv2.FONT_HERSHEY_PLAIN,2,
(0, 0, 255),3)
k = 55

elif (fingers == [1,0,0,0,0] and fingers != [1,1,1,1,1] and x1 > x2):
    cv2.putText(img, str("Kiri"), (10, 90),
cv2.FONT_HERSHEY_PLAIN,2,
(0, 0, 255),3)
k = 44

elif (fingers == [0,1,1,0,0]):
    cv2.putText(img, str("front"), (10, 90),
cv2.FONT_HERSHEY_PLAIN,2,
(0, 0, 255),3)
k = 33

else:
    cv2.putText(img, str("Berhenti"), (10, 90),
cv2.FONT_HERSHEY_PLAIN,2,
(0, 0, 255),3)
k = 77

```

```

        cv2.putText(img, (str("x1: ") + str(x1)+','+str(" x2: ") + str(x2)),
(10, 150), cv2.FONT_HERSHEY_PLAIN,1,
(255, 255, 255),2)
        cv2.putText(img, (str("Servo X: ") + str(gx)), (10, 180),
cv2.FONT_HERSHEY_PLAIN,1,
(255, 255, 255),2)
        cv2.putText(img, (str("Servo Y: ") + str(gy)), (10, 200),
cv2.FONT_HERSHEY_PLAIN,1,
(255, 255, 255),2)

dataObject ={  

    "xc":gx,  

    "yc":gy,  

    "k":k  

}  

else:  

    gx = 404  

    gy = 404  

    k = 404  

    dataObject ={  

        "xc":gx,  

        "yc":gy,  

        "k":k  

}  

# print(data)  

if useSerial == True :  

    dataJson = json.dumps(dataObject)  

    print(dataJson)  

    ser.write((dataJson + '\n').encode())  

cTime = time.time()  

fps = 1/(cTime-pTime)  

pTime = cTime  

cv2.putText(img,str(int(fps)),  

(10,50),
cv2.FONT_HERSHEY_PLAIN, 2, (0,255,255), 3)

```

```
cv2.namedWindow("Image",cv2.WINDOW_NORMAL)
#                                     cv2.setWindowProperty("Image",
cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
cv2.imshow("Image", img)
if cv2.waitKey(20) & 0xFF == ord('q'):
    break
cv2.destroyAllWindows()
cap.release()

if __name__ == "__main__":
    main()
```

LAMPIRAN C
Hasil Alat dan Pengujian



Gambar A Troli Tampak Depan



Gambar B. Troli Tampak Kanan



Gambar C. Troli Tampak Kiri



Gambar D. Posisi Human Machine Interface



Gambar E. Pengujian Jarak 20cm Sensor Ultrasonik

00:08.36

Gambar F. Waktu Troli Bergerak Sejauh 150cm Tanpa Beban



Gambar G. Pengujian Kecepatan Troli Dengan Beban 5Kg



Gambar H. Pengujian Kecepatan Troli Dengan Beban 30Kg

00:09.26

Gambar I. Waktu Troli Bergerak Sejauh 150cm Dengan Beban 5 Kg

00:09.77

Gambar J. Waktu Troli Bergerak Sejauh 150 cm dengan Beban 10Kg

00:11.15

Gambar K. Waktu Troli Bergerak Sejauh 150 cm dengan Beban 15Kg

00:13.12

Gambar L. Waktu Troli Bergerak Sejauh 150 cm dengan Beban 20Kg

00:15.64

Gambar M. Waktu Troli Bergerak Sejauh 150 cm dengan Beban 30Kg

00:20.08

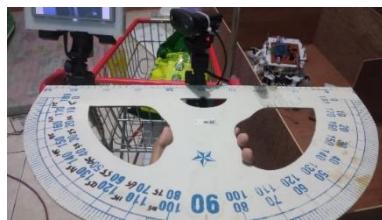
Gambar N. Waktu Troli Bergerak Sejauh 150 cm dengan Beban 50Kg



Gambar O. Posisi Awal Servo Sudut 90°



Gambar P. Posisi Awal Servo Sudut 0°



Gambar Q. Posisi Awal Servo Sudut 30°



Gambar R. Posisi Awal Servo Sudut 45°

1.1 Biodata Penulis



Nama	:	Yudi Arrasyid
Tempat/Tanggal Lahir	:	Depok, 12 Maret 2002
Alamat	:	JL. KH. Sulaeman RT02/RW02, No.6, Sawangan Baru, Depok.
Email	:	yudiarrasyid12@gmail.com
Telepon/Hp	:	085772238374
Hobi	:	Menonton Anime & Film
Moto	:	Hiduplah dan Bekerja Keras
Riwayat Pendidikan	:	
• SD Negeri 01 Sawangan		Tahun 2008 – 2011
• SMP Negeri 10 Depok		Tahun 2014 – 2017
• SMA Negeri 10 Depok		Tahun 2017 – 2020
• Politeknik Negeri Cilacap		Tahun 2020 – 2023
Prodi D3 – Teknik Elektronika		

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 13 Juli 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).