

RANCANG BANGUN HARVEST ASSISTING MOBILE FIELD  
ROBOT BERBASIS COMPUTER VISION DENGAN METODE  
DEEP LEARNING



PROGRAM STUDI SARJANA TERAPAN TEKNIK MEKATRONIKA  
JURUSAN TEKNIK MESIN  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR  
2023

## HALAMAN PENGESAHAN

Skripsi dengan judul “Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning*” oleh Franklin Delano Exel Mainda NIM 444 19 006 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan Teknik pada Program Studi Teknik Mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.

Makassar, 14 Agustus 2023

Menyetujui,

Pembimbing I,

Pembimbing II,



Dr.Eng. Abdul Kadir Muhammad,  
S.T., PG.Dipl., M.Eng.  
NIP. 19750402 200312 1 002

Dr. Ir. Simon Ka'ka, M.T.  
NIP. 19590913 198803 1 001

Mengetahui,

Koordinator Program Studi









Dr.Eng. Akhmad Taufik, S.T., M.T.  
NIP. 19760413 200812 1 003

## HALAMAN PENERIMAAN

Pada hari ini, Rabu tanggal 16 Agustus 2023, tim penguji ujian sidang skripsi telah menerima skripsi mahasiswa: Franklin Delano Exel Mainda NIM 444 19 006 dengan judul: “Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning*”.

Makassar, 16 Agustus 2023

Tim Penguji Ujian Sidang Skripsi:

- |  |               |   |
|--|---------------|---|
| 1. Ir. Lewi, M.T.  | Ketua         | (  )   |
| 2. Paisal, S.T., M.T.                                      | Sekretaris    | (  )  |
| 3. Imran Habriansyah, S.ST., M.T.                          | Anggota I     | (  ) |
| 4. Prof. A.M. Shiddiq Yunus, S.T.,<br>M.Eng.Sc., Ph.D.     | Anggota II    | (  ) |
| 5. Dr.Eng. Abdul Kadir Muhammad,<br>S.T., PG.Dipl., M.Eng. | Pembimbing I  | (  ) |
| 6. Dr. Ir. Simon Ka'ka, M.T.                               | Pembimbing II | (  ) |

## PRAKATA

Puji Syukur kita panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning*” dengan baik.

Dalam penulisan skripsi ini, tidak sedikit hambatan yang penulis alami. Namun berkat bantuan dari berbagai pihak, tantangan itu bisa teratasi. Oleh sebab itu, pada kesempatan ini penulis menyampaikan terima kasih dan penghargaan kepada:

1. Seluruh keluarga penulis terutama kedua Orang Tua tercinta yang senantiasa selalu memberi dukungan baik moril maupun materiil untuk tercapainya Tugas Akhir ini.
2. Bapak Ir. Ilyas Mansur, M.T., selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Dr. Ir. Syaharuddin Rasyid, M.T., selaku Ketua Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.
4. Bapak Dr.Eng. Akhmad Taufik, S.T., M.T., selaku Koordinator Program Studi S1T Teknik Mekatronika Politeknik Negeri Ujung Pandang.
5. Bapak Dr.Eng. Abdul Kadir Muhammad, S.T., M.Eng., selaku Dosen Pembimbing I yang telah memberikan perhatian dan kesempatannya untuk Penulis dalam menyelesaikan skripsi ini.



6. Bapak Dr. Ir. Simon Ka'ka, M.T., selaku Dosen Pembimbing II yang telah memberikan perhatian dan kesempatannya untuk Penulis dalam menyelesaikan skripsi ini.
7. Teman-teman seperjuangan kelas 4 SIT Teknik Mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang, yang telah banyak memberikan

masukan dan bantuan kepada penulis selama proses pembuatan Skripsi.

Semoga Tuhan Yang Maha Esa membalas kebaikan pada siapa pun yang terlibat dalam penyusunan skripsi ini dengan rahmat dan karunia yang melimpah. Besar harapan penulis, dengan hadirnya skripsi ini dapat memberikan sumbangsih yang berarti demi kemajuan ilmu pengetahuan bangsa terutama pada bidang Mekatronika.

Penulis menyadari ketidaksempurnaan skripsi ini baik dari bahasa yang digunakan maupun dari teknik penyajiannya. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan skripsi ini dan demi perbaikan pada masa mendatang.

Makassar, 10 Agustus 2023

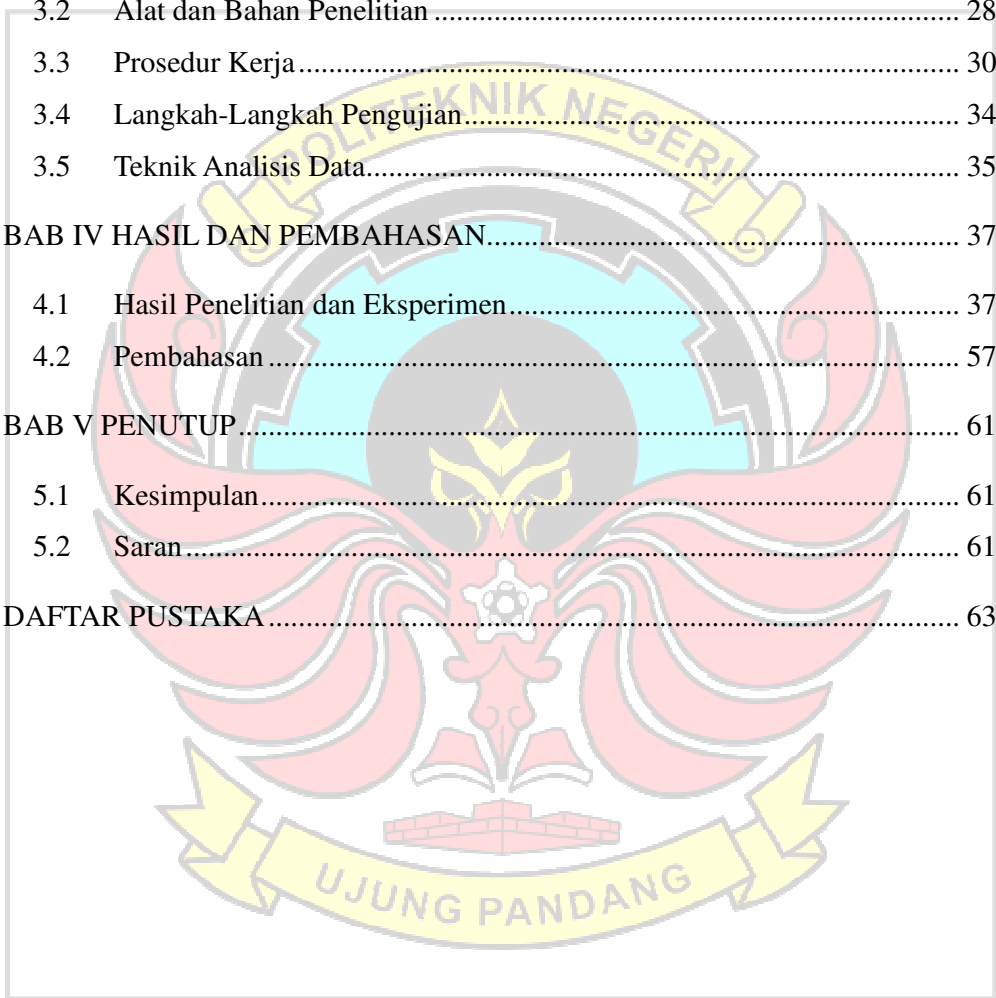


Penulis

## DAFTAR ISI

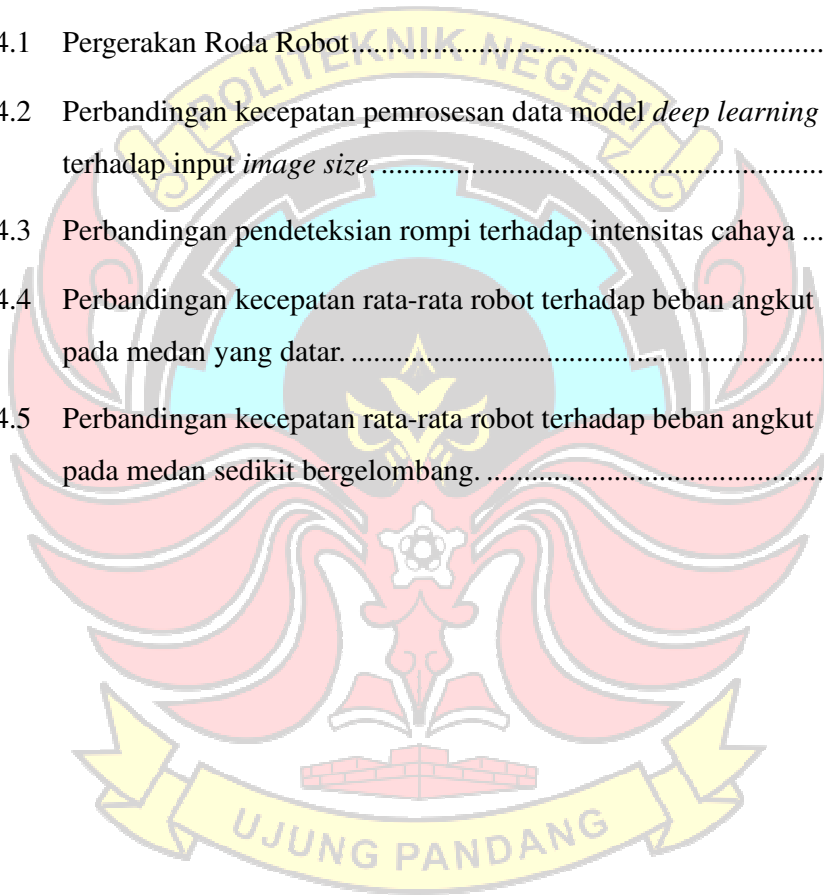
	hlm.
HALAMAN SAMPUL .....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN .....	iii
PRAKATA.....	iv
DAFTAR ISI .....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR .....	ix
DAFTAR LAMPIRAN.....	xii
SURAT PERNYATAAN .....	xiii
RINGKASAN .....	xiv
SUMMARY .....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Ruang Lingkup Penelitian .....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1 Penggunaan Robot dalam Sektor Pertanian .....	5
2.2 <i>Computer Vision</i> .....	6
2.3 <i>Deep Learning</i> .....	7
2.4 Sistem Catu Daya .....	10
2.5 Perangkat Kontrol.....	12

2.6	Aktuator.....	16
2.7	Sensor.....	19
2.8	Penelitian Terkait.....	22
BAB III METODE PENELITIAN.....		28
3.1	Tempat dan Waktu Penelitian.....	28
3.2	Alat dan Bahan Penelitian.....	28
3.3	Prosedur Kerja.....	30
3.4	Langkah-Langkah Pengujian.....	34
3.5	Teknik Analisis Data.....	35
BAB IV HASIL DAN PEMBAHASAN.....		37
4.1	Hasil Penelitian dan Eksperimen.....	37
4.2	Pembahasan.....	57
BAB V PENUTUP.....		61
5.1	Kesimpulan.....	61
5.2	Saran.....	61
DAFTAR PUSTAKA.....		63



## DAFTAR TABEL

	hlm.
Tabel 2.1 Spesifikasi Raspberry Pi 4 Model B .....	13
Tabel 2.2 Spesifikasi Arduino Mega 2560 .....	15
Tabel 3.1 <i>Datasheet</i> motor DC 37GB31ZY.....	33
Tabel 4.1 Pergerakan Roda Robot.....	51
Tabel 4.2 Perbandingan kecepatan pemrosesan data model <i>deep learning</i> terhadap input <i>image size</i> .....	52
Tabel 4.3 Perbandingan pendeteksian rompi terhadap intensitas cahaya .....	52
Tabel 4.4 Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan yang datar.....	54
Tabel 4.5 Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan sedikit bergelombang.....	55



## DAFTAR GAMBAR

	hlm.
Gambar 1.1 <i>Greenhouse</i> .....	1
Gambar 2.1 Petani tradisional .....	5
Gambar 2.2 Traktor tanpa awak .....	6
Gambar 2.3 Aplikasi <i>computer vision</i> .....	7
Gambar 2.4 Diagram venn mengenai <i>deep learning</i> dalam teknologi AI.....	8
Gambar 2.5 Perbedaan <i>machine learning</i> dan <i>deep learning</i> .....	9
Gambar 2.6 Arsitektur <i>convolutional neural network</i> .....	9
Gambar 2.7 Akumulator 12VDC .....	11
Gambar 2.8 Modul Step Down LM2596.....	12
Gambar 2.9 Raspberry Pi 4 Model B .....	12
Gambar 2.10 Arduino Mega 2560.....	14
Gambar 2.11 Driver Motor BTS7960 .....	16
Gambar 2.12 Motor DC 37GB31ZY.....	16
Gambar 2.13 Buzzer piezoelektrik.....	19
Gambar 2.14 Eyesec Webcam 1080P.....	19
Gambar 2.15 Sensor Ultrasonik HC-SR04.....	20
Gambar 2.16 Sensor <i>Load Cell</i> dan modul HX711.....	22
Gambar 2.17 Robot asisten pemanen di <i>greenhouse</i> .....	23
Gambar 2.18 Robot lengan 6DOF.....	24
Gambar 2.19 Robot penyemprot pestisida .....	24
Gambar 2.20 Robot pengumpul bola tenis meja .....	25

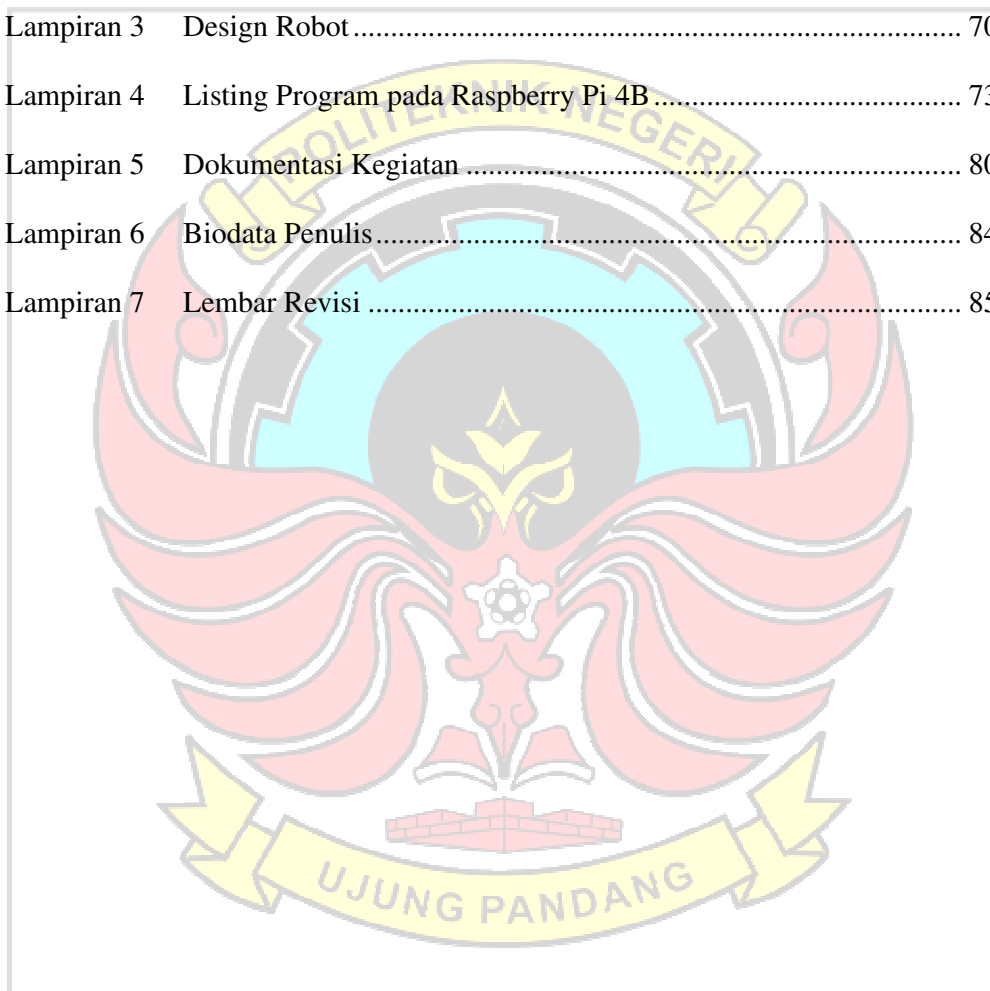
Gambar 2.21	Robot <i>Active Caster</i> .....	27
Gambar 3.1	Diagram alir prosedur perancangan.....	30
Gambar 3.2	Diagram blok sistem elektronik robot .....	31
Gambar 3.3	Desain sistem mekanik robot.....	33
Gambar 3.4	Diagram alir langkah-langkah pengujian .....	35
Gambar 4.1	Pelat besi dan akrilik .....	37
Gambar 4.2	Kerangka robot.....	38
Gambar 4.3	Poros Roda.....	39
Gambar 4.4	Hasil Pekerjaan Mekanika .....	39
Gambar 4.5	Rangkaian sistem catu daya.....	40
Gambar 4.6	Rangkaian pada perangkat kontrol .....	41
Gambar 4.7	Rangkaian pada aktuator .....	42
Gambar 4.8	Rangkaian pada sensor .....	43
Gambar 4.9	Proses anotasi gambar .....	44
Gambar 4.10	Proses augmentasi gambar .....	44
Gambar 4.11	<i>Dataset</i> yang dihasilkan .....	45
Gambar 4.12	Proses instalasi Ultralytics.....	46
Gambar 4.13	Arsitektur YOLOv8n.....	46
Gambar 4.14	Proses training model .....	47
Gambar 4.15	Confusion matrix pada model .....	47
Gambar 4.16	Grafik mAP@50 dan mAP@50-95 model.....	48
Gambar 4.17	<i>Listing program</i> pada Raspberry Pi.....	49
Gambar 4.18	<i>Listing program</i> pada Arduino Mega .....	49

Gambar 4.19 Bentuk lintasan ..... 53  
Gambar 4.20 Grafik kecepatan pemrosesan data ..... 59



## DAFTAR LAMPIRAN

	hlm.
Lampiran 1: Lembar Asistensi Pembimbing I .....	66
Lampiran 2: Lembar Asistensi Pembimbing II .....	68
Lampiran 3 Design Robot .....	70
Lampiran 4 Listing Program pada Raspberry Pi 4B .....	73
Lampiran 5 Dokumentasi Kegiatan .....	80
Lampiran 6 Biodata Penulis .....	84
Lampiran 7 Lembar Revisi .....	85





## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Franklin Delano Exel Mainda

NIM : 444 19 006

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning*” merupakan gagasan dan hasil karya saya sendiri melalui arahan dari pembimbing dan belum pernah diajukan dalam bentuk apa pun pada perguruan tinggi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang telah ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 14 Agustus 2023



Franklin Delano Exel Mainda  
NIM. 444 19 006

# **RANCANG BANGUN HARVEST ASSISTING MOBILE FIELD ROBOT BERBASIS COMPUTER VISION DENGAN METODE DEEP LEARNING**

## **RINGKASAN**

Proses panen hasil tani dalam rumah kaca khususnya di Indonesia saat ini mayoritas masih menggunakan metode manual yang membutuhkan banyak tenaga. Hal tersebut rentan menyebabkan menurunnya performa petani. Tujuan penelitian ini ialah merancang sebuah asisten robot lapangan seluler yang mampu membantu petani dalam proses pemanenan seperti pengangkut, pemindah, dan pembawa hasil pertanian. Sebab kemampuannya sebagai asisten untuk mengikuti petani dan membawa hasil panen, maka diperlukan kemampuan kognitif pada robot seperti kemampuan penglihatan (visi).

Robot asisten bekerja dengan cara mendeteksi keberadaan (deteksi objek) rompi keamanan yang digunakan petani. Deteksi objek ini dibangun berbasis visi komputer dengan metode pembelajaran mendalam dalam arsitektur jaringan saraf tiruan konvolusi dari YOLOv8n menggunakan bahasa pemrograman Python dengan beberapa modul pustaka pendukung lainnya. Pengolahan citra kemudian diolah pada Raspberry Pi 4B sebagai perangkat kontrol primer untuk mengolah data dari webcam secara waktu riil dan mengatur pergerakan robot berdasarkan letak objek yang dideteksi melalui komunikasi serial dengan Arduino Mega sebagai perangkat kontrol sekunder untuk mengirim sinyal dengan modulasi lebar pulsa dan menggerakkan motor DC pada robot. Sesuai dengan prinsip rrobot lapangan seluler, maka dirancang mekanik robot yang mampu bergerak pada permukaan datar dan sedikit bergelombang. Dari hasil pengujian didapatkan robot mampu berjalan di area datar dan sedikit bergelombang dengan kecepatan yang stabil. Selain itu, tingkat keberhasilan pendeteksian rompi dipengaruhi oleh intensitas cahaya.

# **HARVEST ASSISTING MOBILE FIELD ROBOT DESIGN BASED ON COMPUTER VISION WITH DEEP LEARNING METHODS**

## **SUMMARY**

The process of harvesting agricultural products in greenhouses, especially in Indonesia, currently the majority still use manual methods that require a lot of energy. This is vulnerable to causing a decrease in the performance of farmers. The purpose of this research is to design a mobile field robot assistant that is able to assist farmers in the harvesting process such as transporting, moving, and carrying agricultural products. Because of its ability as an assistant to follow farmers and bring crops, cognitive abilities are needed in robots such as vision abilities.

The assistant robot works by detecting the presence (object detection) of the safety vests used by farmers. This object detection is built based on computer vision with deep learning methods in the convolutional neural network architecture from YOLOv8n using the Python programming language with several other supporting libraries. Image processing is then processed on the Raspberry Pi 4B as the primary control device to process data from the webcam in real-time and regulate the robot's movement based on the location of the detected object via serial communication with Arduino Mega as the secondary control device to send PWM signals and drive the DC motor on the robot. In accordance with the mobile field robot principle, a mechanical robot is designed that is capable of moving on flat and slightly wavy surfaces. From the test results, it was found that the robot was able to walk on flat and slightly wavy areas with a stable speed. Besides that, the success rate of vest detection influenced by light intensity.

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Di banyak negara, pertanian memainkan peran penting dalam perekonomiannya, salah satunya Indonesia. Indonesia terkenal sebagai negara agraris karena sebagian penduduknya bekerja di sektor pertanian. Di dalamnya, petani merupakan pelaku utama dalam sektor pertanian yang berperan penting dalam mewujudkan ketahanan pangan. Sektor pertanian merupakan sektor yang mendasari kehidupan setiap masyarakat di Indonesia. Potensi dari sektor pertanian di Indonesia didukung oleh ketersediaan sumber daya alam, serta kondisi iklim yang sangat baik untuk bertani. Sehingga, sektor pertanian layak untuk dikembangkan secara berkelanjutan demi kelangsungan hidup suatu bangsa. Sektor pertanian dewasa ini sudah mengalami perkembangan yang cukup pesat. Modernisasi sektor pertanian itu tidak lepas dari perkembangan ilmu pengetahuan dan teknologi, di antaranya penggunaan teknologi *greenhouse* (rumah kaca) dan teknologi robotika.



Gambar 1.1 *Greenhouse*

Sumber: <http://pustaka-pertanian.blogspot.com/>

*Greenhouse* adalah sebuah bangunan yang berfungsi untuk membudidayakan tanaman yang tidak sesuai dengan kondisi lingkungan lokal, seperti tanaman hias, beberapa jenis sayur dan buah-buahan yang sulit dibudidayakan di lahan luar, tunas tanaman yang baru saja tumbuh. Penggunaan *greenhouse* diharapkan mampu meningkatkan produktivitas pertanian para petani. Namun kenyataannya, terkadang

produktivitas pertanian tersebut justru tidak mengalami peningkatan atau malah mengalami penurunan. Salah satu kendalanya dapat dijumpai pada saat proses panen. Proses panen hasil tani dalam *greenhouse* khususnya di Indonesia saat ini mayoritas masih menggunakan metode manual yang membutuhkan banyak tenaga dan waktu. Hal tersebut rentan menyebabkan menurunnya performa petani. Oleh karena itu, diperlukan solusi yang efisien dan dapat mengoptimalkan proses panen.

Afghani dan Javed (2013) memelopori penelitian tentang pengembangan *Mobile Robot AI* dengan mengembangkan sebuah *Person Following Robot* yang dapat mengikuti manusia berdasarkan panas yang dipantulkan oleh manusia menggunakan sensor IR Beacon. Sementara itu, Wang dkk. (2015) mengembangkan *Person Following Robot* lainnya dengan menggunakan sensor Kinect yang mampu secara cerdas mendeteksi orang yang tepat dan secara otomatis mengikuti orang tersebut.

Untuk mengatasi masalah tersebut, penulis merancang sebuah robot yang mampu membantu petani terkhusus yang menerapkan metode *greenhouse* dalam proses pertanian. Adanya *mobile field robot* pengangkut, pemindah, dan pembawa tanaman merupakan salah satu pekerjaan yang dapat mengurangi faktor kelelahan pada proses pemanenan manual yang dilakukan manusia. Sebab kemampuannya

dalam mengikuti petani, maka diperlukan kemampuan kognitif pada robot yang mampu menjadi penghubung antara robot dan manusia agar dapat mengikuti petani seperti kemampuan penglihatan (*vision*) pada robot. Oleh karena itu, prasyarat untuk mengembangkan robot ini adalah mengimplementasikan *computer vision* dengan metode *deep learning*.

Dengan demikian, skripsi ini bertujuan untuk mengembangkan sebuah robot yang dapat mengikuti petani selama proses pemanenan. Penelitian ini diharapkan dapat memberikan solusi yang efektif dan inovatif yang membantu sektor pertanian Indonesia dalam meningkatkan efisiensi dan akurasi dalam proses pemanenan.

## 1.2 Rumusan Masalah

Dari uraian latar belakang di atas, maka rumusan masalah penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang bangun *mobile field robot* sebagai asisten dalam kegiatan pemanenan pada *greenhouse*?
2. Bagaimana membuat sistem kecerdasan buatan *deep learning* untuk *harvest assisting mobile field robot* berbasis *computer vision*?

## 1.3 Ruang Lingkup Penelitian

Adapun ruang lingkup penelitian ini yaitu sebagai berikut:

1. Implementasi *object detection* sebagai tugas dari *computer vision* dengan metode *deep learning* pada robot menggunakan Raspberry Pi 4 Model B.
2. Uji coba dan validasi sistem pada *mobile field robot* dilakukan pada daerah berlantai datar dan sedikit gelombang.

3. Pengujian untuk parameter beban angkut menggunakan buah tomat.
4. Petani yang akan dideteksi mengenakan sebuah penanda yaitu rompi.

#### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini, yaitu:

1. Merancang bangun *mobile field robot* sebagai asisten dalam kegiatan pemanenan pada *greenhouse*.
2. Membuat sistem kecerdasan buatan *deep learning* untuk *harvest assisting mobile field robot* berbasis *computer vision*.

#### 1.5 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini antara lain:

1. Meningkatkan produktivitas petani.
2. Untuk mempelajari *computer vision* dan *deep learning* sebagai salah satu teknologi kecerdasan buatan.
3. Dapat menjadi referensi atau sumber informasi bagi penelitian sejenis di masa yang akan datang.
4. Sebagai bentuk kontribusi untuk perkembangan ilmu pengetahuan, terutama bidang ilmu mekatronika.



## BAB II TINJAUAN PUSTAKA

### 2.1 Penggunaan Robot dalam Sektor Pertanian

Menurunnya minat generasi muda terhadap pertanian konvensional berbanding lurus dengan penurunan ketersediaan pangan. Kebutuhan pangan akan terus bertambah seiring bertambahnya jumlah penduduk. Jika masih menggunakan sistem pertanian konvensional, tidak tertutup kemungkinan target pemenuhan pangan tidak akan tercapai. Untuk mengatasi masalah tersebut, maka teknologi dibidang pertanian perlu dikembangkan selain untuk mempermudah juga untuk meningkatkan minat generasi penerus pertanian di Indonesia.



Gambar 2.1 Petani tradisional  
Sumber: <https://www.mosintuwu.com/>

Kemajuan teknologi yang semakin modern menuntut sektor pertanian untuk mengikuti kemajuan yang terjadi pada dinamika perkembangan teknologi di industri. Fokus terbaru dari agro-robotika adalah penerapan aplikasi otomatisasi untuk tugas yang berulang lebih efisien atau efektif dari pada pendekatan manusia atau mesin besar tradisional. Penggunaan teknologi robotika untuk membantu para petani menyelesaikan tugas menanam dan merawat hingga memanen sudah banyak digunakan di negara maju. Robot juga dapat digunakan sebagai bagian dari sistem



pengelolaan hama terpadu, misalnya, untuk penyiangan gulma yang akurat dan murah untuk menangkal hama dan penyakit tanaman.



Gambar 2.2 Traktor tanpa awak  
Sumber: <https://www.futurefarming.com/>

## 2.2 *Computer Vision*

*Computer vision* adalah bidang yang sangat luas yang mencakup berbagai macam cara untuk memproses gambar (*image processing*) dan keragaman aplikasi yang menakjubkan. Aplikasi *computer vision* berkisar dari mereproduksi kemampuan visual manusia, seperti mengenali wajah, hingga membuat kategori kemampuan visual yang sama sekali baru. “Sebagai contoh dari *kategori* terakhir, salah satu aplikasi *computer vision* baru-baru ini adalah mengenali gelombang suara dari getaran yang ditimbulkannya pada objek yang terlihat dalam video” (Davis, A. dkk., 2014).

Menurut Ballard dkk. (1983), *computer vision* menjadi salah satu bidang penelitian yang paling aktif untuk aplikasi *deep learning*, karena penglihatan adalah tugas yang mudah bagi manusia dan banyak hewan tetapi menantang bagi komputer. Banyak dari tugas tolok ukur standar yang paling populer untuk

algoritma *deep learning* adalah bentuk pengenalan objek atau pengenalan karakter optik.



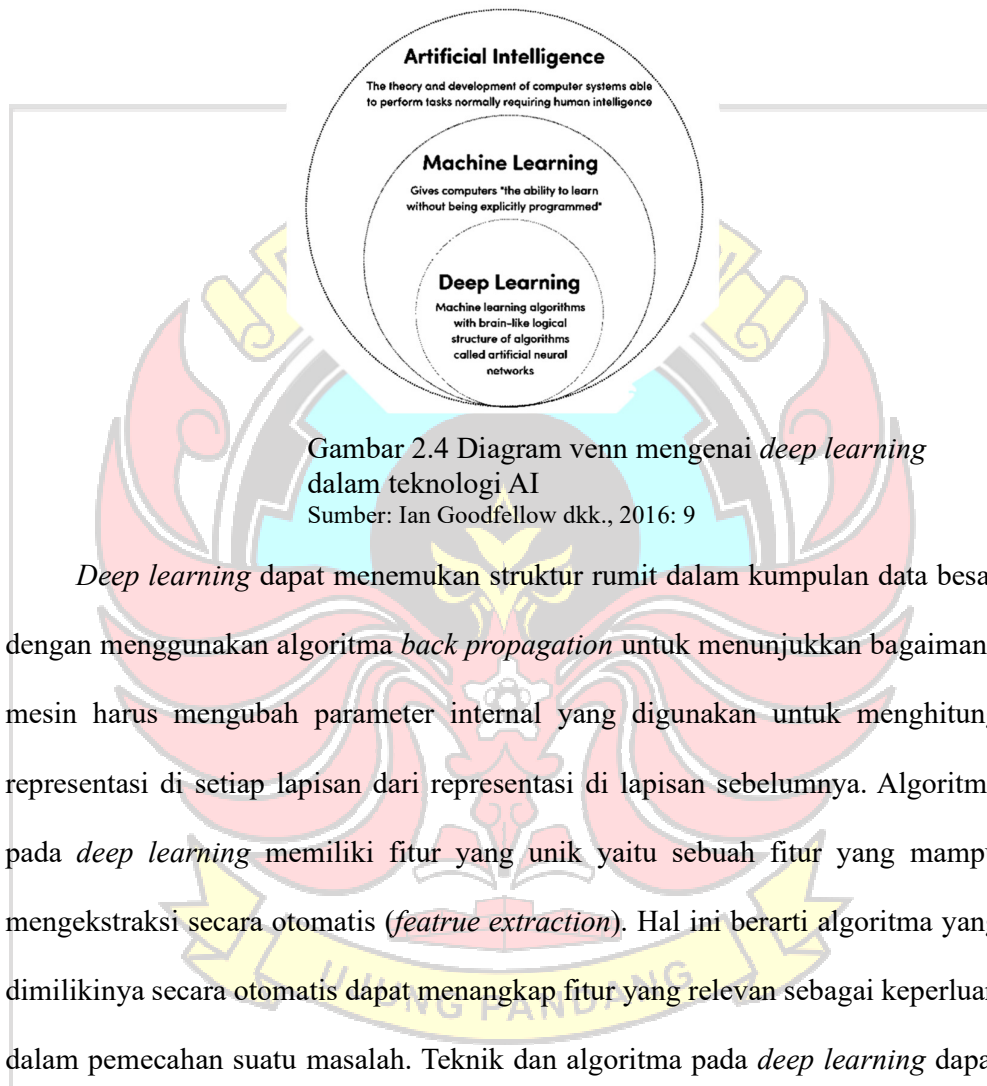
Gambar 2.3 Aplikasi *computer vision*  
Sumber: <https://www.netapp.com/>

*Computer vision* bekerja dengan cara memecah gambar yang ada menjadi banyak bagian-bagian yang berbeda. Jaringan neuron akan mengidentifikasi setiap bagian-bagian yang ada, barulah kemudian menggabungkannya menjadi satu bagian yang utuh. Secara otomatis komputer akan melabeli gambar tersebut. Itulah mengapa semakin banyak data yang dijadikan bahan pembelajaran oleh komputer, kemampuan untuk memprediksi gambar tersebut secara benar juga akan meningkat. Komputer akan memanfaatkan algoritma yang mampu untuk menganalisis warna, bentuk, jarak antar objek, dan seterusnya hingga komputer mampu melabeli seluruh objek sebagai benda tertentu.

### 2.3 *Deep Learning*

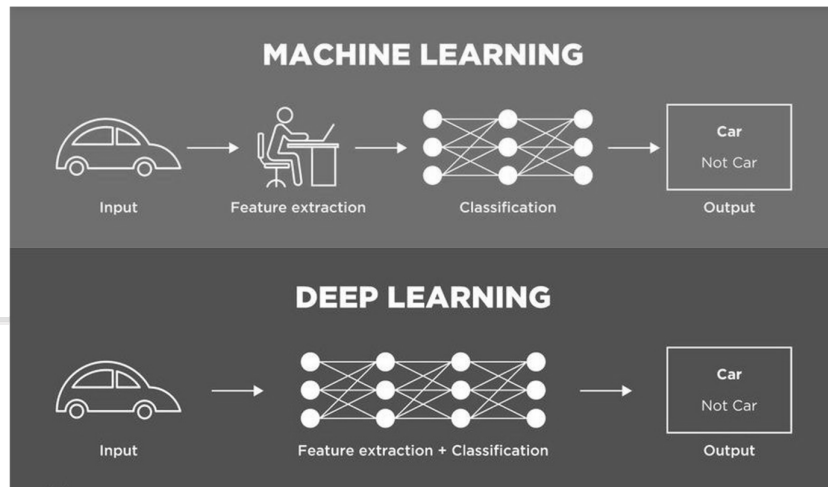
*Deep learning* adalah salah satu cabang dari ilmu pembelajaran mesin (*machine learning*) yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi. *Deep learning* menggunakan sekumpulan fungsi transformasi non-linear yang

ditata berlapis-lapis dan mendalam. *Deep learning* disebut sebagai *deep* (dalam) karena struktur dan jumlah lapisan (*layer*) jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan.



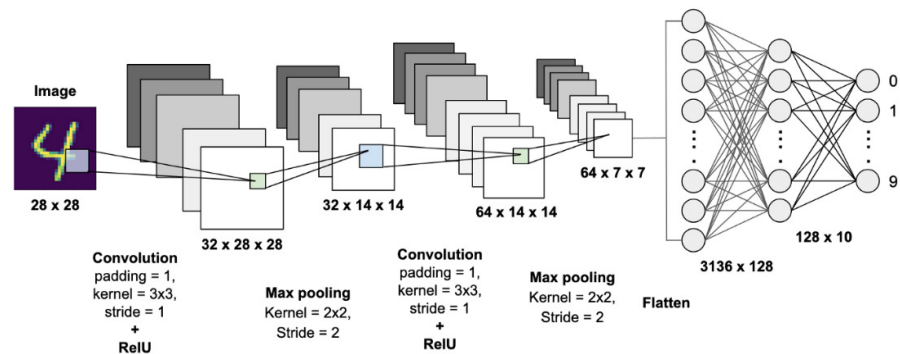
Gambar 2.4 Diagram venn mengenai *deep learning* dalam teknologi AI  
Sumber: Ian Goodfellow dkk., 2016: 9

*Deep learning* dapat menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma *back propagation* untuk menunjukkan bagaimana mesin harus mengubah parameter internal yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya. Algoritma pada *deep learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis (*feature extraction*). Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Teknik dan algoritma pada *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*), atau semi-terarah (*semi-supervised learning*).



Gambar 2.5 Perbedaan *machine learning* dan *deep learning*  
 Sumber: <https://inmarketing.id>

Dalam *Deep Learning*, model *Neural Network* yang terdiri dari banyak lapisan digunakan untuk mempelajari representasi data yang semakin kompleks dari data masukan. Setiap lapisan memproses data masukan dan menghasilkan keluaran yang dijadikan masukan untuk lapisan berikutnya. Proses ini diulang sampai keluaran dari lapisan terakhir yang merupakan output model. *Deep Learning* sering kali digunakan untuk memproses data yang bersifat kompleks, seperti data citra, suara, dan bahasa alami dan telah digunakan dalam berbagai bidang.



Gambar 2.6 Arsitektur *convolutional neural network*  
 Sumber: <https://jackdani.medium.com/>

### 2.3.1 *Object Detection*

*Object detection* merupakan salah satu penerapan *deep learning* dalam kehidupan sehari-hari yang paling umum namun menantang untuk dilakukan oleh sebuah sistem. *Object detection* mengacu pada kemampuan komputer untuk mendeteksi sejumlah objek pada suatu gambar. Hal ini dapat dilakukan dengan cara mengambil *image feature* seperti garis, sudut, kontur dan warna dari sebuah gambar.

### 2.3.2 You Only Look Once (YOLO)

*You Only Look Once (YOLO)*, merupakan sebuah metode *object detection state of the art* yang diusulkan pada tahun 2015 oleh Redmon *et al.* Versi YOLO yang digunakan pada penelitian ini yaitu YOLOv8. YOLOv8 merupakan pengembangan dari versi YOLO sebelumnya. YOLOv8 memiliki jaringan ekstraktor yang lebih kuat dan memiliki fitur deteksi *multi-scale* dibandingkan dengan versi sebelumnya. Selain itu YOLOv8 ini dapat mendeteksi target yang lebih banyak dari target kecil hingga besar.

## 2.4 Sistem Catu Daya

### 2.4.1 Akumulator

Akumulator (aki) adalah sebuah alat yang dapat menyimpan energi (umumnya energi listrik) dalam bentuk energi kimia. Contoh-contoh akumulator adalah baterai dan kapasitor. Pada umumnya di Indonesia, kata akumulator (sebagai aki atau accu) hanya dimengerti sebagai "baterai" kendaraan. Sedangkan di bahasa Inggris, kata akumulator dapat mengacu kepada baterai, kapasitor, kompulsator, dll.

Di dalam standar internasional, setiap satu sel akumulator memiliki tegangan sebesar 2 [V]. Sehingga aki 12 [V] memiliki 6 sel, sedangkan aki 24 [V] memiliki 12 sel.

Aki merupakan sel yang banyak kita jumpai karena banyak digunakan pada sepeda motor maupun mobil. Aki termasuk sel sekunder, karena selain menghasilkan arus listrik, aki juga dapat diisi arus listrik kembali. Secara sederhana aki merupakan sel yang terdiri dari elektrode Pb sebagai anode dan PbO<sub>2</sub> sebagai katode dengan elektrolit H<sub>2</sub>SO<sub>4</sub>



Gambar 2.7 Akumulator 12VDC  
Sumber: <https://otoklix.com/>

#### 2.4.2 Modul *Step Down* LM2596

Modul *Step Down* LM2596 adalah sebuah modul yang berfungsi untuk menurunkan tegangan dari tegangan input 4 – 38 [V<sub>DC</sub>] menjadi 2 – 36 [V<sub>DC</sub>] dengan arus keluaran hingga 5 [A] serta efisiensi kerja hingga 96%. Namun modul ini tidak dilengkapi dengan perlindungan polaritas terbalik, sehingga harus berhati-hati dalam pemasangannya.





Gambar 2.8 Modul Step Down LM2596  
Sumber: <https://www.blibli.com/>

## 2.5 Perangkat Kontrol

### 2.5.1 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B adalah komputer papan tunggal (*single-board computer*) seukuran kartu kredit kecil yang dikembangkan di Inggris oleh Raspberry Pi Foundation. Tujuannya adalah untuk mendukung dan mempromosikan pengajaran ilmu komputer dasar di sekolah-sekolah dan di negara-negara berkembang. Raspberry Pi telah digunakan untuk banyak hal, seperti robotika, pengembangan super komputer, tablet *portable*, dan masih banyak lagi.



Gambar 2.9 Raspberry Pi 4 Model B  
Sumber: <https://id.wikipedia.org/>

Nama Raspberry Pi diambil dari nama buah, yaitu buah Raspberry, sedangkan Pi diambil dari kata Python, yaitu nama dari sebuah bahasa pemrograman. Python dijadikan bahasa pemrograman utama dari Raspberry Pi, namun tidak tertutup

kemungkinan untuk menggunakan bahasa pemrograman lain pada Raspberry Pi. Spesifikasi dari Raspberry Pi 4 Model B dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2.1 Spesifikasi Raspberry Pi 4 Model B

<b>Spesifikasi</b>	
<b>SoC</b>	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit @ 1.5GHz
<b>RAM</b>	1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depends on model)
<b>Konektivitas</b>	<ul style="list-style-type: none"> <li>• 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless</li> <li>• Bluetooth 5.0 BLE</li> <li>• Gigabit Ethernet</li> </ul>
<b>Multimedia</b>	<ul style="list-style-type: none"> <li>• H.265 (4kp60 decode)</li> <li>• H264 (1080p60 decode, 1080p30 encode)</li> <li>• OpenGL ES 3.1, Vulkan 1.0</li> </ul>
<b>Memori</b>	microSD
<b>Port</b>	<ul style="list-style-type: none"> <li>• 2 × USB 3.0 ports; 2 USB 2.0 ports. (Lanjutan tabel 2.1)</li> <li>• 2 × micro-HDMI ports (up to 4kp60 supported)</li> <li>• 2-lane MIPI DSI display port</li> <li>• 2-lane MIPI CSI camera port</li> <li>• 4-pole stereo audio and composite video port</li> </ul>
<b>GPIO</b>	Raspberry Pi standard 40 pin GPIO header
<b>Power</b>	<ul style="list-style-type: none"> <li>• 5V DC via USB-C connector (upto 3A*)</li> <li>• 5V DC via GPIO header (upto 3A*)</li> <li>• Power over Ethernet (PoE) enabled (requires separate PoE HAT)</li> </ul>
<b>Dimensi</b>	85 mm × 56 mm

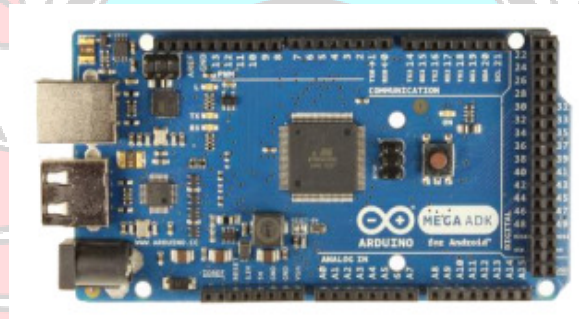
Sumber: <https://www.raspberrypi.com/>



### 2.5.2 Arduino Mega 2560

Arduino merupakan sebuah perangkat elektronik yang bersifat *open source* dan sering digunakan untuk merancang dan membuat perangkat elektronik serta *software* yang mudah untuk digunakan. Arduino ini dirancang sedemikian rupa untuk mempermudah penggunaan perangkat elektronik di berbagai bidang.

Arduino ini memiliki beberapa komponen penting di dalamnya, seperti pin I/O, IC mikrokontroler, osilator, dan beberapa konektor. Selain itu, Arduino juga sudah menggunakan bahasa pemrograman Arduino Language yang sedikit mirip dengan bahasa pemrograman C++.



Gambar 2.10 Arduino Mega 2560

Sumber: <https://store.arduino.cc/>

Dalam penelitian ini digunakan Arduino Mega 2560 sebagai kontroler yang akan mengontrol semua fungsi aktuator dan sensor mengingat bahwa jenis Arduino ini memiliki 54 pin I/O digital dan 16 pin I/O analog sehingga mampu mengontrol banyak komponen atau modul sekaligus. Spesifikasi dari Arduino Mega 2560 dapat pada Tabel 2.2 di bawah ini.

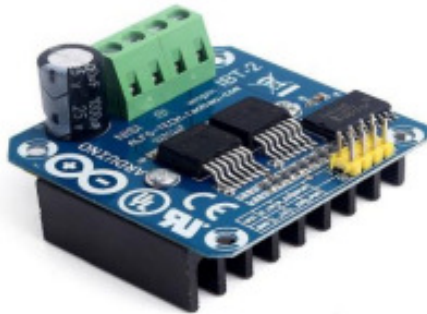
Tabel 2.2 Spesifikasi Arduino Mega 2560

<b>Spesifikasi</b>	
<b>Tegangan operasi</b>	5V
<b>Tegangan input (rekomendasi)</b>	7V – 12V
<b>Tegangan input (limit)</b>	6V – 20V
<b>Pin I/O digital</b>	53 pin (ada 15 pin PWM)
<b>Pin Input analog</b>	16 pin
<b>Arus DC per pin I/O</b>	20 mA
<b>Arus DC pin 3.3V</b>	50 mA
<b>Flash Memory</b>	256 KB (8 KB digunakan oleh <i>bootloader</i> )
<b>SRAM</b>	8 KB
(Lanjutan tabel 2.2)	
<b>EEPROM</b>	4 KB
<b>Kecepatan Clock</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Berat</b>	37 g
<b>Dimensi</b>	101.52 mm × 53.3 mm

Sumber: <https://store.arduino.cc/>

### 2.5.3 Driver Motor BTS7960

*Driver* yang digunakan memiliki tegangan input 12VDC, berfungsi sebagai pembalik arah putaran motor DC sekaligus mengatur kecepatan putaran motor DC. Dimana polaritas *input* tegangan motor DC dibalik melalui IC yang terdapat pada papan kontrol *driver* tersebut dengan bantuan Arduino Mega 2560.

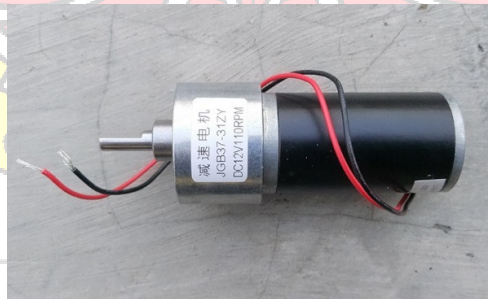


Gambar 2.11 Driver Motor BTS7960  
Sumber: <https://kampungtengah.blogspot.com/>

## 2.6 Aktuator

### 2.6.1 Motor DC

Motor DC adalah sebuah perangkat yang mengubah energi listrik menjadi energi gerak. Motor DC memiliki dua terminal dan memerlukan tegangan arus searah atau *direct current* (DC) agar dapat beroperasi. Operasi motor tergantung pada interaksi dua medan magnet. Secara sederhana dikatakan bahwa motor listrik bekerja dengan prinsip bahwa dua medan magnet dapat dibuat berinteraksi untuk menghasilkan gerakan (torsion).



Gambar 2.12 Motor DC 37GB31ZY  
Sumber: <https://shopee.co.id/>

Arah putaran motor DC dapat diatur untuk berputar searah jarum jam (*clockwise*) atau berlawanan arah jarum jam (*counter-clockwise*) apabila polaritas

listrik yang diberikan pada motor DC tersebut dibalikkan. Untuk mempermudah pengaturan kecepatan dan arah putaran motor, maka digunakanlah teknik modulasi lebar pulsa (*pulse width modulation*).

*Pulse Width Modulation* (PWM) adalah teknik modulasi dengan mengubah lebar pulsa (*duty cycle*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi *high* kemudian berada di zona transisi ke kondisi *low*. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. *Duty Cycle* merupakan representasi dari kondisi logika *high* dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan *range* 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi *high* terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan *high* sama dengan keadaan *low* maka sinyal mempunyai *duty cycle* sebesar 50%. Semakin besar nilai *duty cycle* yang diberikan maka putaran motor akan semakin cepat.

Untuk menghitung daya motor dapat dihitung menurut persamaan berikut:

$$P = \omega \cdot \tau \dots\dots\dots(1)$$

dimana:

$P$  = Daya atau *power* [W]

$\omega$  = Kecepatan sudut [rad/s]

$\tau$  = Torsi (*torque*) [N.m]

Rumus untuk mencari torsi adalah:

$$\tau = F \cdot l \dots\dots\dots(2)$$

dimana:

$\tau$  = Torsi (*torque*) [N.m]

$F$  = Gaya penggerak [N]

$l$  = Jarak [m]

Rumus untuk mencari kecepatan sudut adalah:

$$\omega = \frac{2\pi n}{60} \dots\dots\dots(3)$$

dimana:

$\omega$  = Kecepatan sudut [rad/s]

$n$  = Putaran [rpm]

Sehingga dapat dituliskan:

$$P = \frac{2\pi n}{60} \cdot F \cdot l \dots\dots\dots(4)$$

Untuk menghitung beban maksimal motor DC jika digunakan pada permukaan dengan kemiringan tertentu dari persamaan di bawah ini:

$$Motor PL_{Max} = \frac{\frac{1}{\sin \alpha} \times \tau_{motor}}{r} \times n \dots\dots\dots(5)$$

Sumber: <https://robotics.stackexchange.com/>

dimana:

$Motor PL_{Max}$  : Beban maksimal motor DC [kg]

$\tau_{motor}$  : Torsi motor [kg.cm]

$\alpha$  : kemiringan permukaan [deg]

$r$  : Jari-jari roda [cm]

$n$  : Jumlah motor DC yang digunakan

### 2.6.2 Buzzer

Buzzer adalah perangkat persinyalan audio, yang dapat berupa perangkat mekanis, elektromekanis, atau piezoelektrik. Penggunaan umum buzzer biasanya

diaplikasikan pada perangkat alarm, *timer*, atau konfirmasi input pengguna seperti klik mouse atau penekanan tombol. Pada penelitian ini buzzer digunakan sebagai alarm atau pembuat tanda peringatan pada robot.



Gambar 2.13 Buzzer piezoelektrik  
Sumber: <https://squishycircuits.com/>

## 2.7 Sensor

### 2.7.1 Eyesec Webcam 1080P

Eyesec Webcam 1080P mengadopsi sensor CMOS-2MP sebagai jeroannya. Kamera ini dapat digunakan untuk mengambil gambar definisi tinggi. Kamera ini mendukung mode video 1080p30, 720p30 dan masih banyak lagi. Kamera ini dapat dihubungkan melalui kabel USB3.0 ke *port* USB pada Raspberry Pi

Kamera berfungsi dengan semua model Raspberry Pi. Kamera ini dapat diakses melalui V4L dan ada banyak *library* pihak ketiga yang tersedia di internet. Kamera ini dapat diatur posisinya secara 360 derajat.



Gambar 2.14 Eyesec Webcam 1080P  
Sumber: <https://tokopedia.link/ZRgiV8zmcCb>

### 2.7.2 Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 sesuai namanya merupakan sensor yang menggunakan gelombang ultrasonik. Gelombang ultrasonik suara atau getaran dengan frekuensi yang terlalu tinggi untuk bisa didengar oleh telinga manusia, yaitu kira-kira di atas 20.000 [Hz]. Hanya beberapa hewan yang menggunakan konsep ultrasonik, seperti lumba-lumba menggunakannya untuk komunikasi, sedangkan kelelawar menggunakan gelombang ultrasonik untuk navigasi. Dalam hal ini, gelombang ultrasonik merupakan gelombang ultra (di atas) frekuensi gelombang suara (sonik).



Gambar 2.15 Sensor Ultrasonik HC-SR04

Sumber: <https://www.nn-digital.com/>

Sensor ini berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Sensor akan mengirim atau memancarkan gelombang ultrasonik terus menerus oleh transmiter kemudian gelombang ultrasonik tersebut dipantulkan oleh suatu benda di depannya dan diterima oleh *receiver* kemudian selisih waktu antara sesaat setelah memancarkan dan sesaat sebelum menerima gelombang dihitung dengan rumus kecepatan. Kecepatan gelombang ultrasonik itu sekitar 340 [m/s] dimana untuk menempuh jarak 1 [m] memerlukan waktu  $1/340$  atau 0,00294 [s]. Jika menempuh jarak 1 [cm] maka akan membutuhkan waktu 0,0000294 [s] (29,4 [ $\mu$ s]). Gelombang ultrasonik melakukan perjalanan pulang pergi (*transmit* –

*receive*) sehingga waktu yang dibutuhkan menjadi 2 kali lipat. Hal ini berpengaruh pada perhitungan jaraknya. Waktu tempuh juga menjadi 2 kali lipat, sehingga untuk menempuh jarak 1 [cm] diperlukan waktu  $29,4 \text{ } [\mu\text{s}] \times 2 = 58,8 \text{ } [\mu\text{s}]$ . Jadi untuk menghitung jarak :

$$X = t/58.8$$

dimana:  $X$  = Jarak actual [cm]

$t$  = Selisih waktu sesaat setelah transmiter melepas gelombang dan sebelum *receiver* menerimanya [s]

### 2.7.3 Load Cell

Sensor *load cell* adalah transduser yang dapat mengubah tekanan oleh beban menjadi sinyal listrik. Konversi terjadi secara tidak langsung dalam dua tahap. Melalui pengaturan mekanis, gaya tekan dideteksi berdasarkan deformasi dari matriks pengukur regangan (*strain gauges*) dalam bentuk resistor *planar*. Regangan ini mengubah hambatan efektif (*effective resistance*) empat pengukur regangan yang disusun dalam konfigurasi jembatan Wheatstone (*Wheatstone bridge*) yang kemudian dibaca berupa perbedaan potensial (tegangan). Tegangan keluaran yang dihasilkan dari *load cell* sangat kecil, sehingga dibutuhkan sebuah penguat khusus. Penguat yang dapat digunakan adalah modul HX711.





Gambar 2.16 Sensor *Load Cell* dan modul HX711

Sumber: <https://www.amazon.eg>

HX711 adalah sebuah konverter ADC 24-bit yang dirancang untuk timbangan dan aplikasi kontrol industri. Input multiplexer akan memilih saluran input diferensial A atau B ke penguat PGA (*Programmable Gain Amplifier*). Saluran A dapat diprogram dengan *gain* penguatan 128 atau 64, yang sesuai dengan tegangan input diferensial skala penuh masing-masing yaitu kurang lebih 20 [mV] atau kurang lebih 40 [mV]. Saluran B memiliki gain penguatan tetap sebesar 32. *On-chip power supply* menghilangkan kebutuhan regulator eksternal untuk menyediakan daya bagi ADC dan sensor. Input *clock* fleksibel, karena bisa di dapat dari sumber *clock* eksternal, Kristal atau dari *on-chip oscillator*.

## 2.8 Penelitian Terkait

- 1) Sistem Navigasi Berbasis Pelacakan Objek pada Operasi Robot Asisten Pemanenan di Dalam *Greenhouse*. (2017)

Tujuan dari penelitian ini adalah merancang robot asisten (*assisting robot*) dalam operasi pemanenan yang dapat membawakan hasil panen dan terus

mengikuti pemanen (*person following*) agar pemanen dapat terus memanen dan tidak cepat mengalami kelelahan.

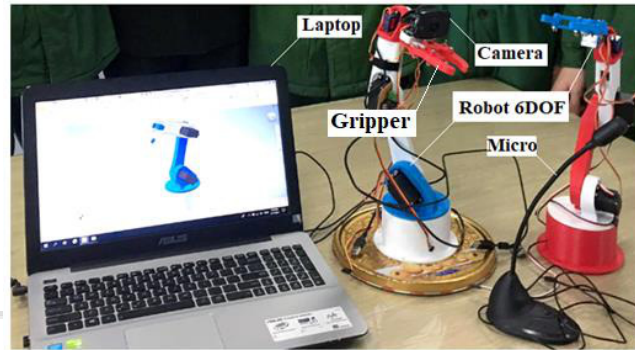


Gambar 2.17 Robot asisten pemanen di *greenhouse*  
Sumber: Awal dkk., 2017

Hasil penelitian ini adalah robot dapat berfungsi dengan baik. Walaupun robot yang dibuat merupakan permodelan dari desain sebenarnya, namun secara fungsional sudah menyerupai desain yang sebenarnya. Berdasarkan uji kinerja, robot dapat mengikuti objek warna yang sudah ditentukan. Namun dengan batasan-batasan yang telah diuji. Selanjutnya akan dibuat prototipe dari robot ini yang sesuai dengan desain yang diinginkan sehingga dapat diuji di lapangan, kemudian dapat diketahui keefektifan dan efisiensi dari robot ini dalam meningkatkan kapasitas pemanenan di dalam *greenhouse*.

- 2) *Deep Learning based 6DOF Robot Arm Control with Computer Vision.* (2021)

Tujuan dari penelitian ini adalah untuk membangun sistem kontrol lengan robot 6DOF yang melakukan tugas pengambilan objek berdasarkan teknik *computer vision* menggunakan model *deep learning*.



Gambar 2.18 Robot lengan 6DOF  
 Sumber: Bien Xuan Duong dkk., 2021

Hasil penelitian ini adalah robot menunjukkan keakuratan dan keandalan teknik *computer vision* dengan akurasi pengenalan sebesar 92%. Namun penelitian ini belum mempertimbangkan evaluasi akurasi gerak lengan robot karena adanya kesalahan fabrikasi, kecepatan pemrosesan sistem kontrol secara *real-time*.

3) Rancang Bangun Robot Penyemprot Pestisida Otonom dengan Sistem *Wall-follower* pada Penyemprotan Tanaman Cabai. (2021)

Tujuan dari penelitian ini adalah untuk membuat robot penyemprot pestisida otomatis yang mampu menyusuri dinding bedengan dengan sistem *wall-follower* yang mampu diimplementasikan pada lahan pertanian konvensional dengan tanaman cabai yang ditanam di bedengan.

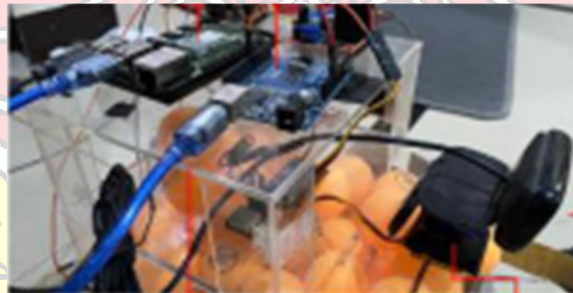


Gambar 2.19 Robot penyemprot pestisida  
 Sumber: Budiono, 2021

Hasil penelitian ini adalah 1) robot berhasil menyusuri bedengan dengan sistem *wall-follower* pada lahan pertanian konvensional berdasarkan pembacaan sensor ultrasonik. 2) Penyemprotan pestisida otomatis menggunakan sensor ultrasonik dapat mendeteksi tanaman cabai pada jarak maksimum 90 [cm]. 3) Penyemprotan robot berhasil melakukan penyemprotan ke seluruh tanaman cabai tanpa menghasilkan penyemprotan berlebih dengan kecepatan robot 0,3125 [m/s], ketinggian semprotan 60 [cm], lebar semprotan 115 [cm], dan debit semprotan 17,94 [mL/s].

4) Penerapan Deteksi dan Pelacakan Objek pada Robot Otonom Pengumpul Bola Tennis Meja Menggunakan Pengolahan Citra. (2021)

Tujuan dari penelitian ini adalah untuk merancang suatu robot otonom (*autonomous robot*) yang bertujuan untuk mengumpulkan bola tenis meja tanpa harus mengeluarkan lebih banyak energi dari pemain tersebut.



Gambar 2.20 Robot pengumpul bola tenis meja  
Sumber: Naufalimam dkk., 2021

Hasil penelitian ini adalah 1) robot mampu mendeteksi bola tenis meja berdasarkan bentuk dan warna dengan menggunakan metode *contour detection*. 2) Adanya ROI digunakan sebagai daerah tengah untuk deteksi bola serta untuk menggerakkan kamera agar posisi kamera berada di daerah tengah *frame*. 3) Pada uji skenario deteksi jarak bola dalam keadaan diam, kamera mendeteksi empat bola

dengan masing-masing jarak, yaitu: (i) 45 [cm], (ii) 65 [cm],(iii) 85 [cm], dan (iv) 105 [cm]. Akurasi deteksi pada jarak 45 [cm] sebesar 99.607%, pada jarak 65 [cm] didapatkan akurasi sebesar 97.631%, pada jarak 85 [cm] didapatkan akurasi sebesar 99.238%, dan pada jarak 105 [cm] didapatkan akurasi sebesar 99.567%. 4) Pada uji skenario deteksi jarak bola dalam keadaan bergerak dari  $0^{\circ}$ – $90^{\circ}$ , kamera mendeteksi empat bola dengan masing-masing jarak, yaitu: (i) 45 [cm], (ii) 65 [cm], (iii) 85 [cm], dan (iv) 105 [cm]. Akurasi deteksi pada jarak 45 [cm] sebesar 82.936%, pada jarak 65 [cm] didapatkan akurasi sebesar 88.304%, pada jarak 85 [cm] didapatkan akurasi sebesar 86.850%, dan pada jarak 105 [cm] didapatkan akurasi sebesar 87.902%. 5) Pada uji *tracking* bola, motor servo membutuhkan waktu tempuh sebesar 1.7071 [s] untuk *tracking* bola dari sudut  $0^{\circ}$ – $45^{\circ}$ , 7.7985 [s] untuk *tracking* bola dari sudut  $0^{\circ}$ – $90^{\circ}$ , dan 13.0151 [s] untuk *tracking* bola dari sudut  $0^{\circ}$ – $135^{\circ}$ .

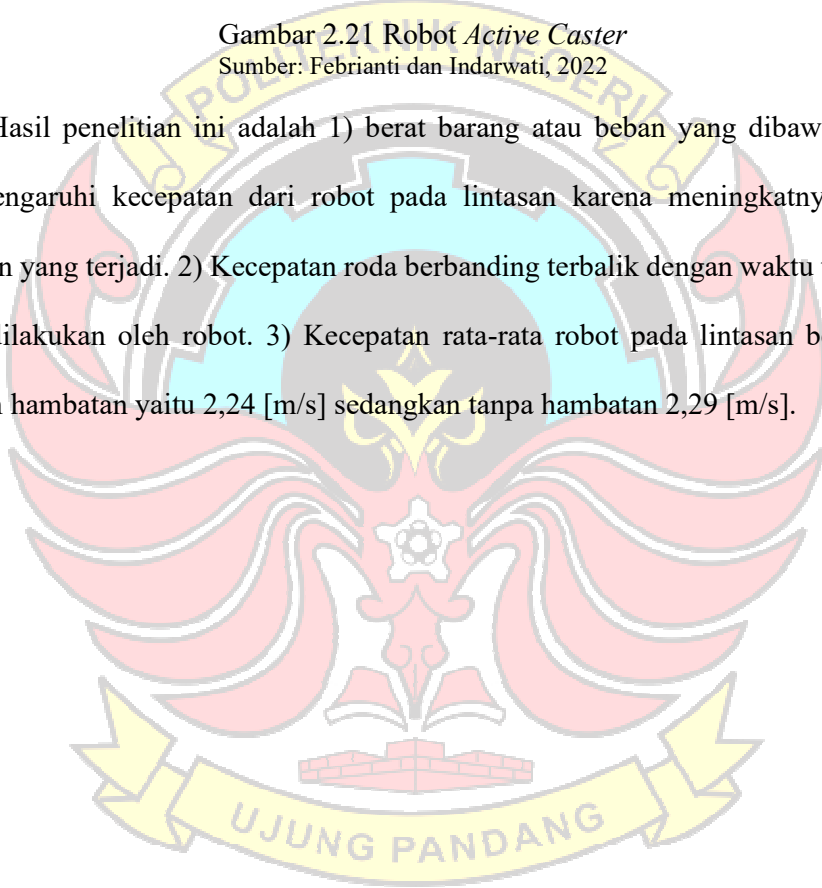
5) Pengembangan Robot *Active Caster* Berbasis *Deep Learning* dan *Internet of Things* Untuk Aplikasi *Warehouse*. (2022)

Tujuan dari penelitian ini adalah untuk mengembangkan robot dengan mengubah penerapan robot dari *Delivery Robot* menjadi *Warehouse Robot*, meningkatkan pergerakan dan pengolahan *database* menggunakan raspberry pi, membuat program yang dapat membuat pergerakan robot lebih stabil, dan pembuatan *jobsheet* yang memuat langkah-langkah pengoperasian robot dan panduan pemrograman sehingga lebih mudah untuk dipahami.



Gambar 2.21 Robot *Active Caster*  
Sumber: Febrianti dan Indarwati, 2022

Hasil penelitian ini adalah 1) berat barang atau beban yang dibawa robot mempengaruhi kecepatan dari robot pada lintasan karena meningkatnya gaya gesekan yang terjadi. 2) Kecepatan roda berbanding terbalik dengan waktu tempuh yang dilakukan oleh robot. 3) Kecepatan rata-rata robot pada lintasan berkelok dengan hambatan yaitu 2,24 [m/s] sedangkan tanpa hambatan 2,29 [m/s].



## BAB III METODE PENELITIAN

### 3.1 Tempat dan Waktu Penelitian

Seluruh kegiatan baik perancangan dan pengembangan maupun eksperimen akan dilakukan di Kampus I Politeknik Negeri Ujung Pandang tepatnya pada Laboratorium Mekatronika dan Sistem Otomasi serta Laboratorium Riset Pascasarjana. Waktu pelaksanaan penelitian dimulai pada bulan Februari – Agustus 2023.

### 3.2 Alat dan Bahan Penelitian

#### 3.2.1 Alat yang digunakan

Alat-alat yang akan digunakan dalam pelaksanaan penelitian terdaftar pada Tabel 3.1.

Tabel 3.1 Alat yang digunakan

No.	Nama Alat	Jumlah (buah)
1.	PC dengan OS Windows	1
2.	Obeng Plus	1
3.	Obeng Minus	1
4.	Tang Kombinasi	1
5.	Palu	1
6.	Gerinda	1
7.	Mesin Las	1
8.	Bor	1
9.	Meteran	1
10.	Multimeter	1
11.	Solder	1
12.	Penyedot timah	1
13.	Cas aki	1



### 3.2.2 Bahan yang digunakan

Bahan-bahan yang akan digunakan dalam pelaksanaan penelitian terdaftar pada Tabel 3.2.

Tabel 3.2 Bahan yang digunakan

No.	Nama Bahan	Jumlah
1.	Raspberry Pi 4 Model B	1 buah
2.	Cooling Fan Raspberry Pi 4B	1 buah
3.	MicroSD 32GB	1 buah
4.	Arduino Mega 2560	1 buah
5.	EyeSec 1080p USB Webcam	1 buah
6.	Sensor Ultrasonik HC-SR04	1 buah
7.	Sensor Load Cell 10kg	1 buah
8.	Modul HX711	1 buah
9.	Driver Motor BTS7960	4 buah
10.	Motor DC	4 buah
11.	Ban Troli Karet	4 buah
12.	LCD Raspberry Pi	1 buah
13.	Buzzer Aktif	1 buah
14.	Shock absorber	4 buah
15.	Aki 12[V] 9 [AH]	1 buah
16.	Sakelar DPDT	1 buah
17.	Push Button	1 buah
18.	Modul Step Down LM2596 Display	1 buah
19.	Modul Port USB2.0 Tipe A Female	4 buah
20.	Kabel USB Tipe A Male	4 buah
21.	Spacer Brass M3x50	24 buah
22.	Spacer Brass M3x10	64 buah
23.	Ampelas	Disesuaikan
24.	Lem	Disesuaikan
25.	Besi plat 2 [mm]	Disesuaikan
26.	As Stainless dia. 13mm	Disesuaikan
27.	Besi Hollow 30mm x 30mm	Disesuaikan
28.	Kabel jumper	Disesuaikan

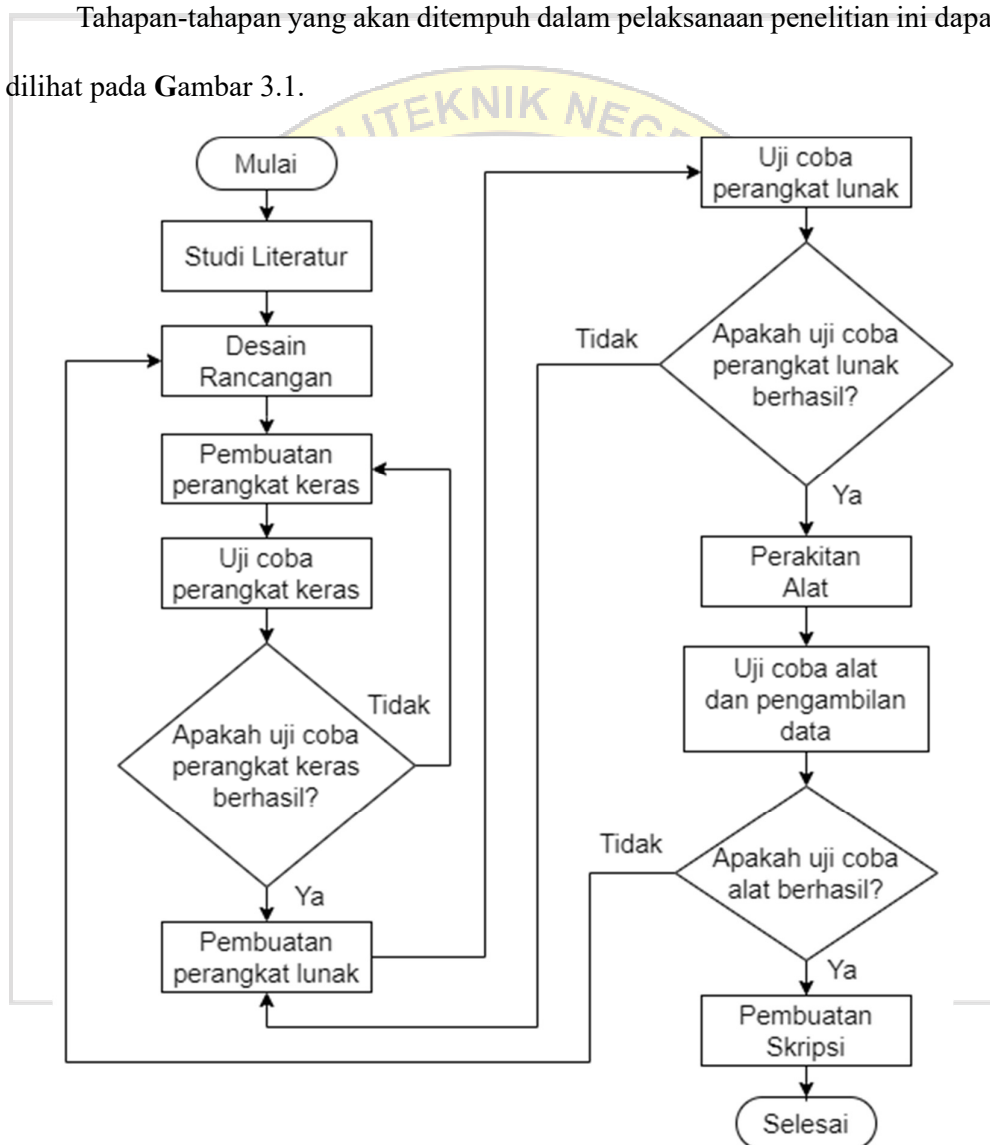


(Lanjutan Tabel 3.2)

29.	Elektroda 2mm	Disesuaikan
30.	Timah	Disesuaikan
31.	Mur dan baut	Disesuaikan

### 3.3 Prosedur Kerja

Tahapan-tahapan yang akan ditempuh dalam pelaksanaan penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir prosedur perancangan

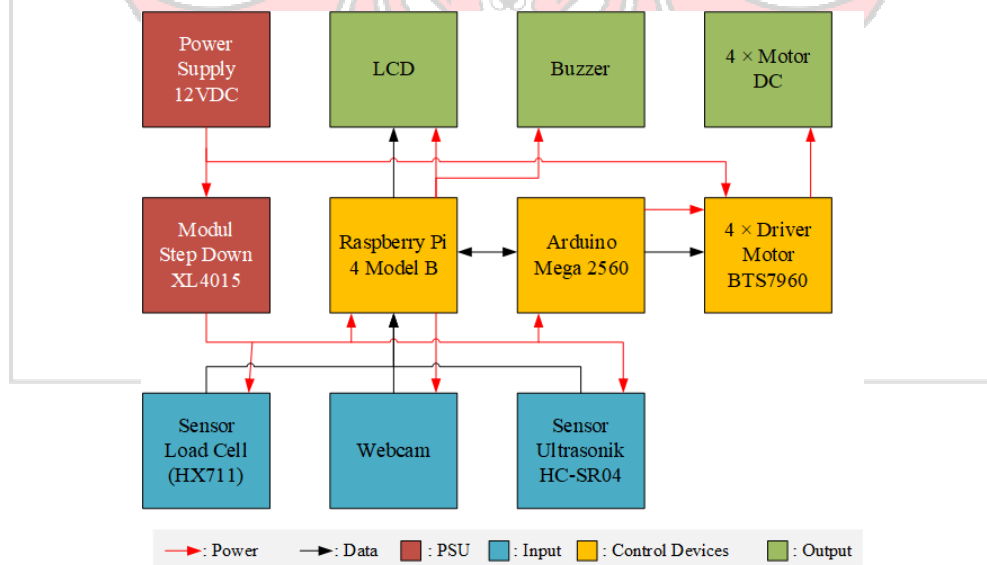
### 3.3.1 Studi Literatur

Studi literatur dimaksudkan untuk mencari data serta informasi berupa skripsi, jurnal, artikel, dan buku, dimana informasi tersebut harus relevan dengan alat yang akan dibuat. Setelah itu dilakukan pembuatan suatu rancangan penelitian dalam bentuk proposal, berisi tentang kerangka dasar yang menjadi acuan bagi penulis dalam melaksanakan penelitian.

### 3.3.2 Perancangan dan Pembuatan Perangkat Keras

#### 1) Sistem Elektronik

Robot akan dibuat dengan menggunakan beberapa modul untuk memperkecil tempat atau ruang dalam peletakannya dan memperkecil ukuran dari alat ini. Selain menggunakan Raspberry Pi, robot ini juga menggunakan Arduino Mega untuk mengontrol aktuator. Hal ini bertujuan untuk mengurangi beban kerja Raspberry Pi akibat proses *computer vision* dengan metode *deep learning*.



Gambar 3.2 Diagram blok sistem elektronik robot

## 2) Sistem Mekanik

Pada perancangan sistem mekanik, dirancang dengan menggunakan prinsip mobile, sesuai dengan fungsinya yang akan digunakan pada permukaan tidak rata. Dari segi mekanis, robot akan dilengkapi dengan *shock absorber* di masing-masing lengan ayun (*swing arm*) tempat dudukan motor DC. Dari hasil perancangan sistem mekanik diperkirakan massa robot mencapai 13 [kg] dan beban angkut yang direncanakan sebesar 5 [kg] sehingga massa total robot sebesar 18 [kg]. Dengan jari-jari roda yang akan digunakan sebesar 5 [cm] dan dengan berat total tersebut maka torsi yang diperlukan yaitu:

$$\tau = F \times r = (m \times a) \times r = (18 \times 9,81) \times 0,05 = 8,829 \text{ [Nm]}$$

dimana  $\tau$  = Torsi yang diperlukan [Nm],

$F$  = Gaya yang bekerja [N],

$m$  = Massa total robot [kg],

$a$  = Percepatan gravitasi [ $\text{m/s}^2$ ],

dan  $r$  = jari-jari roda [m].

Oleh karena itu, dilakukan survei mengenai motor DC yang sesuai. Berdasarkan beberapa jenis motor yang tersedia maka penulis memilih motor DC seri 37GB31ZY karena memiliki *stall torque* sebesar 27 [kg.cm] atau setara 2,65 [Nm]. Agar torsi yang diperlukan dapat terpenuhi, maka digunakan 4 buah motor DC 37GB31ZY, sehingga torsi yang dihasilkan sebesar 10,6 [Nm].

Motor DC ini bekerja pada tegangan 12 VDC serta dilengkapi dengan transmisi *gearbox* yang tersusun atas *spur gear* dengan perbandingan gir 56:1

dengan panjang poros 15 [mm] dan diameter poros 6 [mm]. Berikut *datasheet* dari motor DC dengan seri 37GB31ZY.

Tabel 3.1 *Datasheet* motor DC 37GB31ZY

<i>Rated</i>	<i>No Load</i>			<i>AT Load</i>			<i>Stall</i>		<i>Gearbox</i>
<i>Volt</i>	<i>Speed</i>	<i>Current</i>	<i>Torque</i>	<i>Speed</i>	<i>Current</i>	<i>Output</i>	<i>Torque</i>	<i>Current</i>	<i>Length</i>
V	RPM	mA	kg.cm	RPM	A	W	kg.cm	A	mm
12	140	350	11.2	110	1	10	27	6.5	24

Sumber: <https://id.aliexpress.com/>

Berikut ini gambar desain sistem mekanik robot.



Gambar 3.3 Desain sistem mekanik robot

### 3.3.3 Perancangan dan Pembuatan Perangkat Lunak

Pembuatan perangkat lunak ini bertujuan memberikan perintah untuk melakukan suatu fungsi spesifik kepada Raspberry Pi dan Arduino Mega yang mana akan mengontrol perangkat-perangkat elektronik. Raspberry Pi berfungsi memproses dan mengolah data *computer vision* dengan metode *deep learning*, sedangkan Arduino Mega berfungsi untuk mengontrol aktuator. Bahasa pemrograman yang digunakan Raspberry Pi adalah *Python* dan untuk Arduino Mega menggunakan C atau C++. Raspberry Pi dan Arduino Mega akan

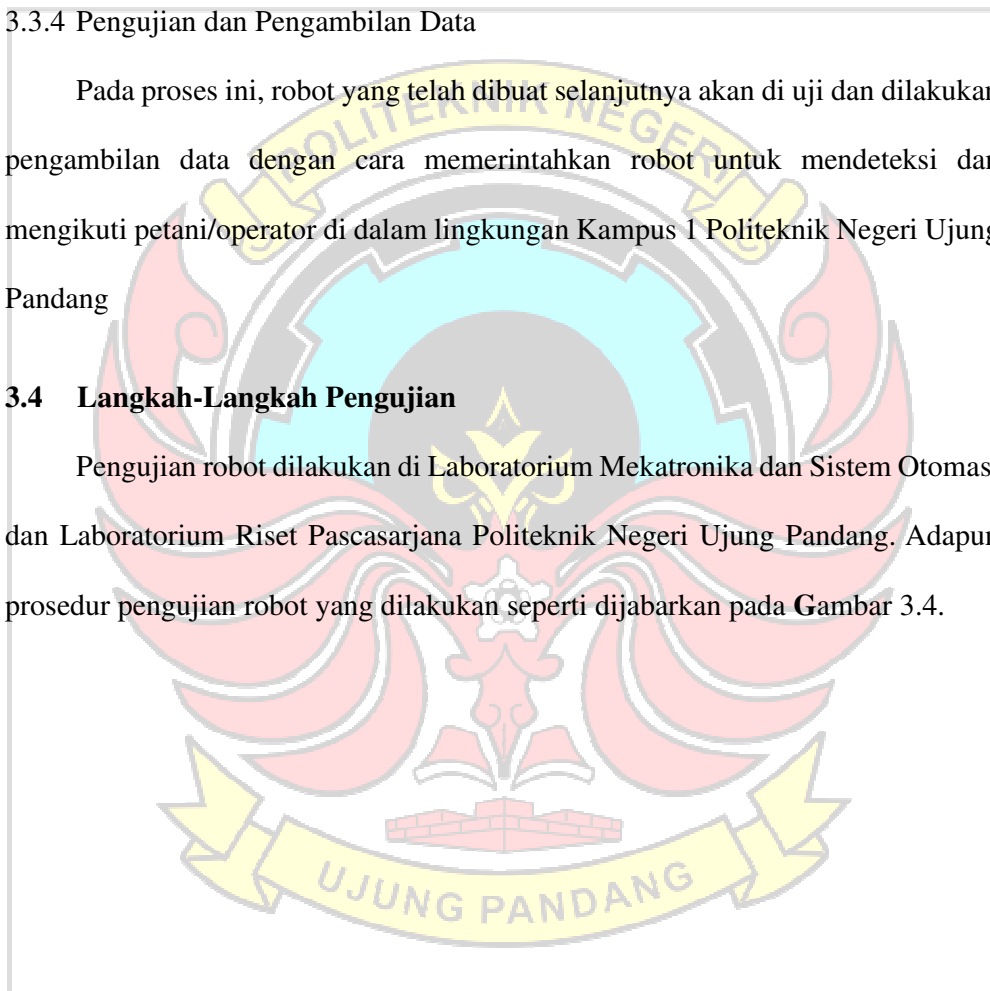
dihubungkan dengan menggunakan komunikasi serial. Perancangan perangkat lunak pada penelitian ini berupa program yang telah dibuat dalam komputer kemudian dimasukkan ke dalam *hardware* elektronik yang telah dibuat sebelumnya.

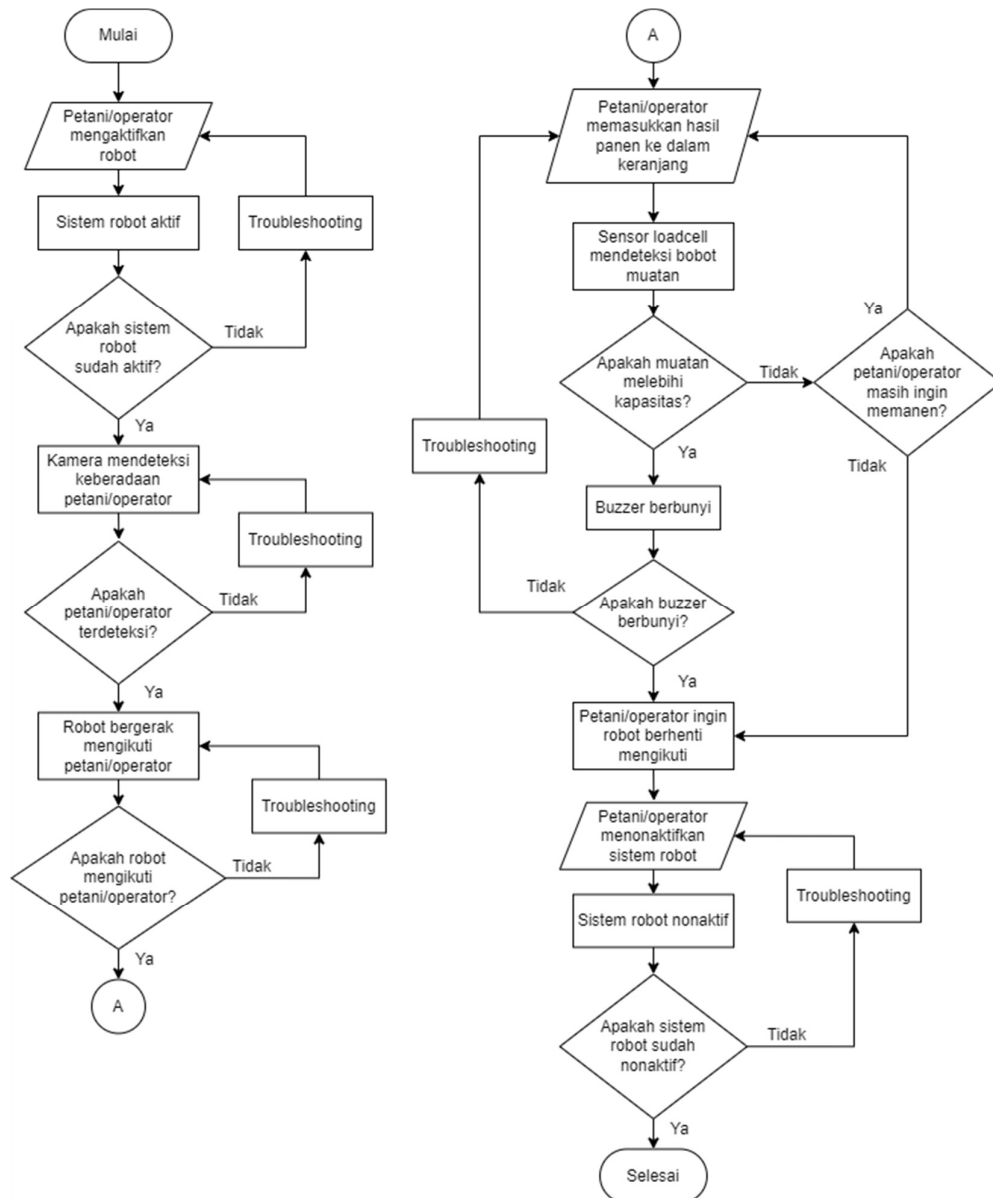
#### 3.3.4 Pengujian dan Pengambilan Data

Pada proses ini, robot yang telah dibuat selanjutnya akan di uji dan dilakukan pengambilan data dengan cara memerintahkan robot untuk mendeteksi dan mengikuti petani/operator di dalam lingkungan Kampus 1 Politeknik Negeri Ujung Pandang

### 3.4 Langkah-Langkah Pengujian

Pengujian robot dilakukan di Laboratorium Mekatronika dan Sistem Otomasi dan Laboratorium Riset Pascasarjana Politeknik Negeri Ujung Pandang. Adapun prosedur pengujian robot yang dilakukan seperti dijabarkan pada Gambar 3.4.





Gambar 3.4 Diagram alir langkah-langkah pengujian

### 3.5 Teknik Analisis Data

Setelah melakukan langkah-langkah pengujian, maka didapatkan data hasil pengujian. Kemudian dari data yang telah terkumpul dilakukan proses analisis pada data-data tersebut meliputi:

- 1) Melakukan analisa kecepatan pemrosesan data *deep learning* dengan input *image size* yang bervariasi.
- 2) Melakukan analisa respons robot untuk mendeteksi rompi dengan intensitas cahaya yang bervariasi.
- 3) Melakukan analisa terhadap pergerakan robot dengan parameter beban angkut bobot bervariasi dari buah tomat pada lintasan datar dan sedikit bergelombang.



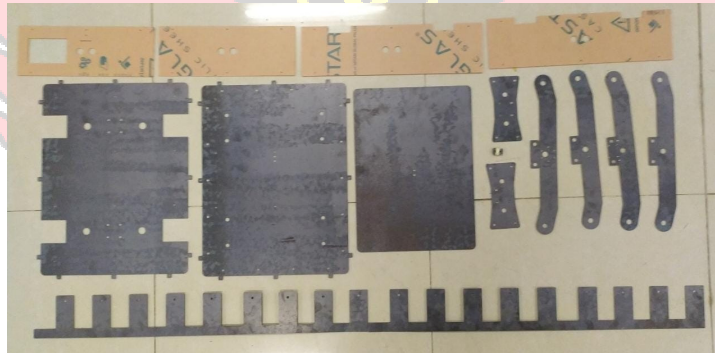
## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Hasil Penelitian dan Eksperimen

Hasil penelitian ini terbagi menjadi beberapa bagian yaitu hasil pekerjaan mekanika, hasil pekerjaan elektronika, hasil pekerjaan informatika, dan hasil pengujian dari Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning*.

#### 4.1.1 Hasil Pekerjaan Mekanika

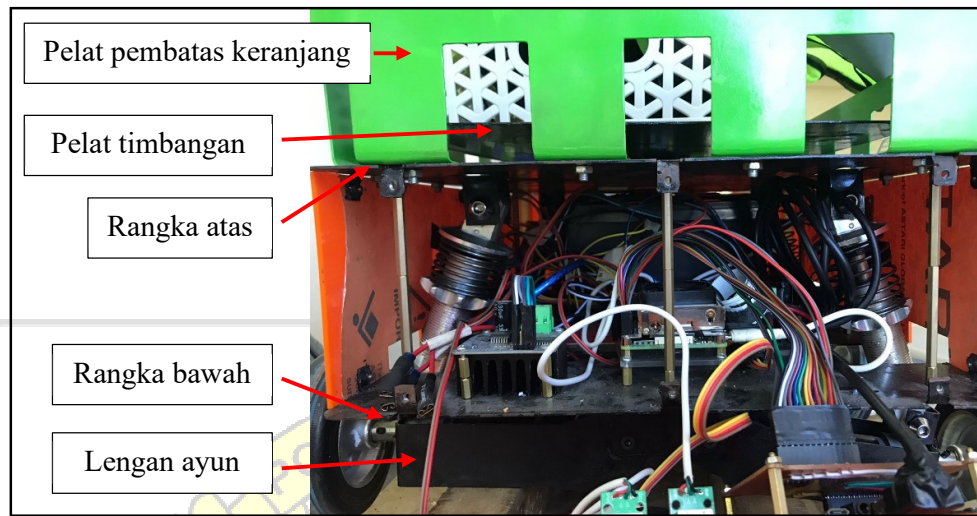
Robot ini dibangun dengan dua material utama yaitu pelat besi 1,5 [mm] dan akrilik 2 [mm]. Pelat besi digunakan pada bagian rangka atas, rangka bawah, lengan ayun, pagar pembatas, dan alas timbangan. Sedangkan akrilik digunakan pada bagian bodi robot.



Gambar 4.1 Pelat besi dan akrilik

Kedua material tersebut dipotong menggunakan mesin pemotong laser agar hasilnya lebih presisi. Pelat besi yang sudah dipotong tadi, kemudian ditebuk menggunakan mesin tekuk sesuai dengan desain yang telah dibuat. Setelah ditebuk, pelat besi kemudian dicat agar tahan terhadap korosi dan menambah nilai estetika robot. Selanjutnya, pelat besi tersebut dirakit hingga membentuk kerangka yang utuh.





Gambar 4.2 Kerangka robot

Rangka bawah dan rangka atas dihubungkan menggunakan *spacer* yang terbuat dari bahan kuningan. Rangka bawah merupakan tempat dipasangnya sebagian besar komponen elektronik, sedangkan rangka atas merupakan tempat penyimpanan barang yang akan diangkut serta tempat untuk tiang kamera di bagian belakang. Pada masing-masing lengan ayun, dipasangkan sebuah motor DC dan dihubungkan *shock absorber* agar robot dapat berjalan di area yang sedikit bergelombang.

Untuk bagian roda penggerak robot, dibuatkan sebuah poros yang terbuat dari material *stainless steel* berdiameter 13 [mm]. Salah satu ujung poros disambungkan ke roda troli karet berdiameter 10 [cm] dengan cara dilas. Kemudian, ujung poros yang lain dibor dengan diameter 6 [mm] sesuai dengan diameter poros motor DC. Selanjutnya dibuatkan pengunci sebanyak 4 buah.



Gambar 4.3 Poros Roda

Untuk bagian bodi robot, terdapat sensor ultrasonik di setiap sisi. Selain itu, pada sisi depan terdapat komponen antarmuka robot seperti LCD, sakelar ON/OFF, *push button*, dan port USB. Pada sisi kiri terdapat indikator tegangan baterai. Pada sisi kanan terdapat buzzer. Kemudian pada sisi belakang terdapat port USB dan soket kabel cas aki. Setelah semuanya dirakit, robot ini memiliki dimensi panjang 490 [cm], lebar 420 [cm], tinggi 540 [cm], dan berat tanpa beban 13,6 [kg]. Berikut ini tampilan hasil akhir pekerjaan mekanika.



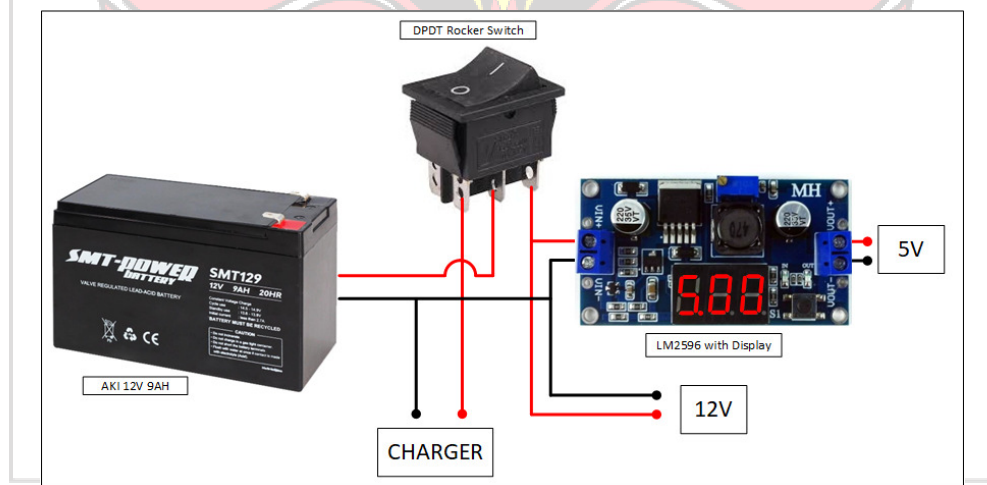
Gambar 4.4 Hasil Pekerjaan Mekanika

#### 4.1.2 Hasil Pekerjaan Elektronika

Hasil pekerjaan elektronika terbagi menjadi empat bagian yaitu rangkaian pada perangkat kontrol, rangkaian pada sensor, rangkaian pada aktuator, dan rangkaian pada sistem catu daya.

##### 1) Rangkaian pada sistem catu daya

Rangkaian pada sistem catu daya merupakan rangkaian yang cukup vital karena nantinya akan menyuplai daya ke setiap komponen pada robot. Rangkaian ini terdiri dari sebuah Aki 12 [V] dengan kapasitas 9 [AH] yang menjadi sumber daya utama. Aki kemudian dihubungkan ke saklar DPDT agar saat saklar pada posisi OFF aki dapat dicas. Sedangkan pada saat posisi ON aki tidak dapat dicas. Selain itu, saat posisi ON tegangan dari aki diparalelkan ke modul *step down* LM2596 untuk menghasilkan tegangan 5V dan tegangan 12V.

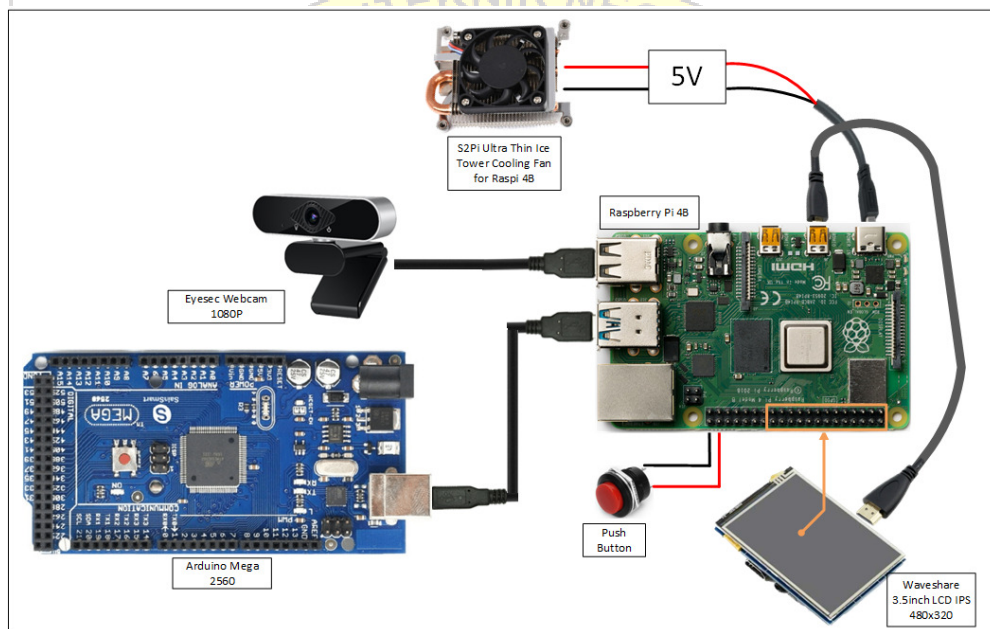


Gambar 4.5 Rangkaian sistem catu daya

##### 2) Rangkaian pada perangkat kontrol

Rangkaian pada perangkat kontrol terdiri dari Raspberry Pi yang merupakan komponen utama yang mengatur seluruh proses kerja robot. Raspberry Pi

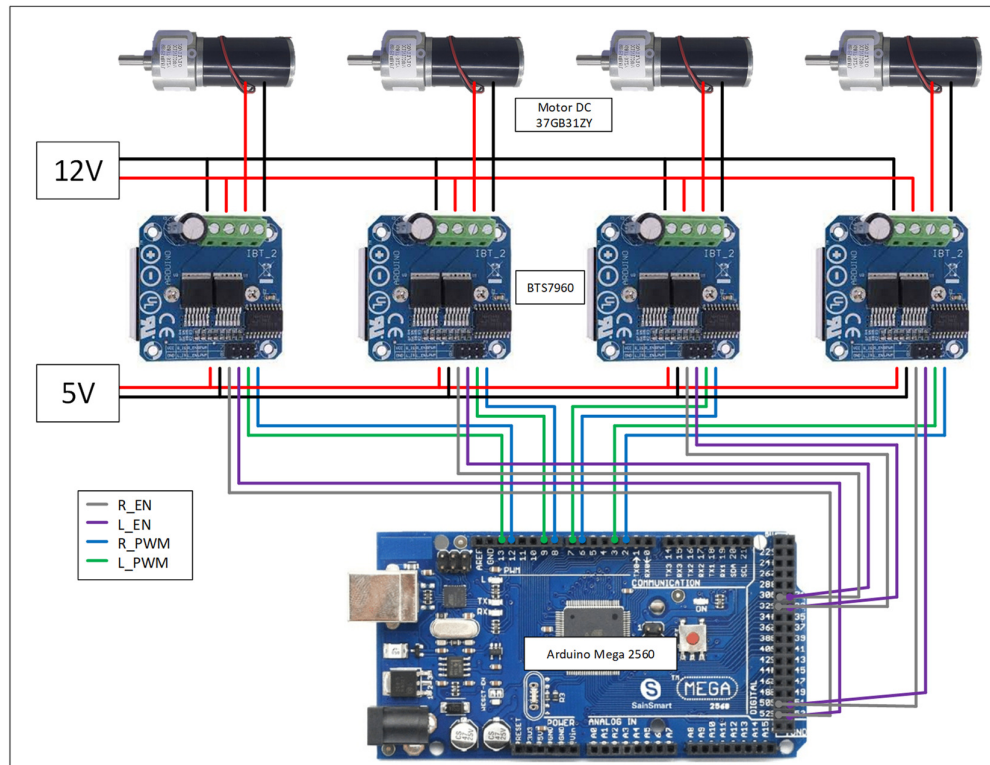
dihubungkan ke Webcam 1080p dan Arduino Mega menggunakan kabel USB, serta dipasang sebuah kipas pendingin agar tidak mengalami *overheat*. Arduino Mega berfungsi untuk mengendalikan kerja aktuator. Selain itu, Raspberry Pi dihubungkan ke LCD TFT sebagai antarmuka pengoperasian robot. Terdapat sebuah *push button* yang dihubungkan ke GPIO Raspberry Pi yang berfungsi untuk menghentikan program yang sedang berjalan.



Gambar 4.6 Rangkaian pada perangkat kontrol

### 3) Rangkaian pada aktuator

Rangkaian pada aktuator terdiri dari 4 buah motor DC 37GB31ZY. Masing-masing motor DC terhubung ke sebuah *driver* motor BTS7960. Kemudian setiap driver motor dihubungkan ke Arduino Mega. Arduino Mega menerima perintah dari Raspberry Pi kemudian mengolah dan meneruskan perintah berupa sinyal PWM ke driver motor untuk mengatur arah dan kecepatan putaran motor DC.

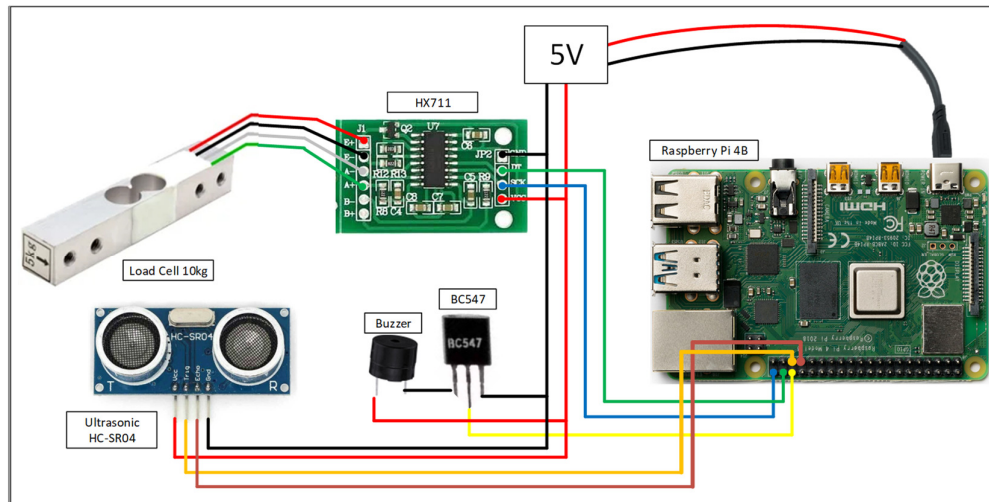


Gambar 4.7 Rangkaian pada aktuator

#### 4) Rangkaian pada sensor

Rangkaian pada sensor terdiri dari sebuah load cell 10 [kg] beserta *amplifier* HX711 dan sebuah sensor ultrasonik HC-SR04 yang dihubungkan ke Raspberry Pi. Karena sinyal yang dikeluarkan oleh load cell sangat kecil, maka keluaran load cell dihubungkan ke HX711 untuk menguatkan sinyal tersebut sebelum ke GPIO Raspberry Pi. Selain itu, terdapat sebuah buzzer untuk memberikan indikator kelebihan muatan. Buzzer tersebut tidak langsung dihubungkan ke Raspberry Pi melainkan melalui transistor BC547 karena tegangan keluaran pada GPIO Raspberry Pi hanya sebesar 3,3 [V]. Transistor BC547 berfungsi sebagai saklar agar buzzer dapat terhubung ke 5 [V] dan GND.





Gambar 4.8 Rangkaian pada sensor

#### 4.1.3 Hasil Pekerjaan Informatika

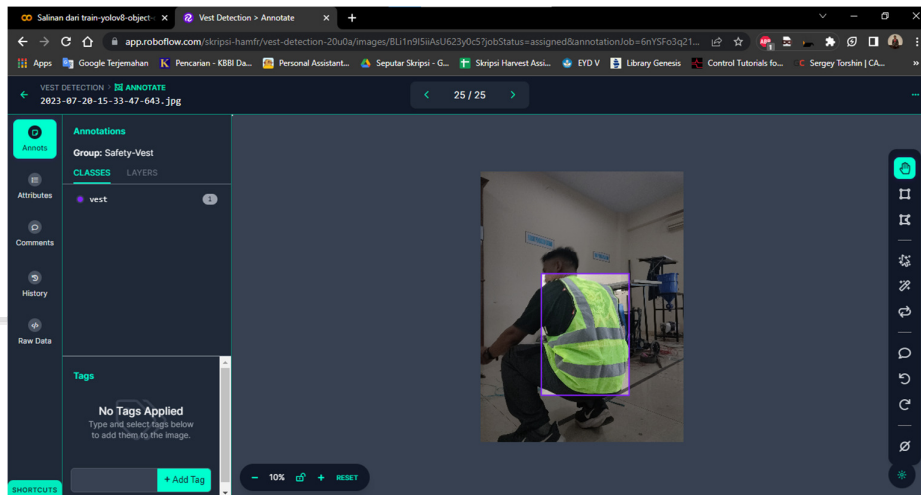
Hasil pekerjaan informatika terbagi menjadi tiga bagian yaitu pembuatan model *deep learning*, pembuatan program pada Raspberry Pi 4B, dan pembuatan program pada Arduino Mega 2560.

##### 1) Pembuatan model *deep learning*

Pembuatan model *deep learning* dilakukan di luar komponen yang digunakan pada robot sebab prosesnya sangat berat. Adapun kegiatan yang dilakukan dalam pembuatan model *deep learning* adalah sebagai berikut:

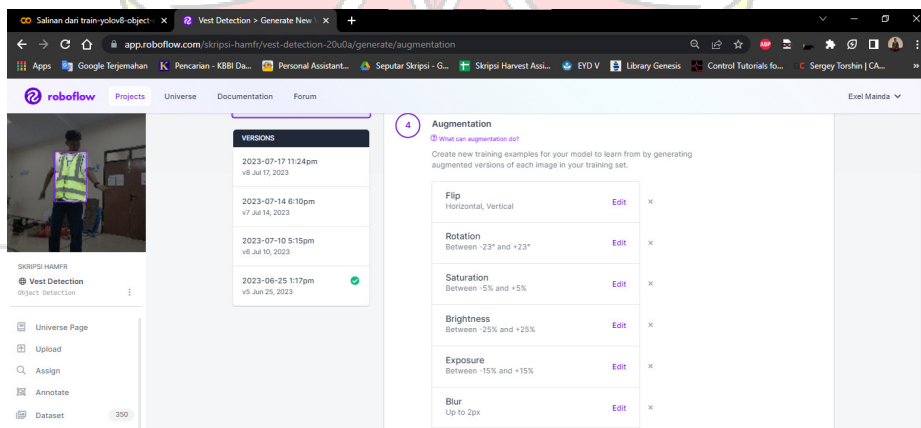
##### a) Pembuatan *dataset*

Pembuatan *dataset* ini dilakukan pada laman Roboflow. Tahap pertama yaitu mengumpulkan (*upload*) gambar yang mengandung rompi. Di sini penulis mengumpulkan 527 gambar asli. Selanjutnya, gambar tersebut diberikan keterangan (*annotate*) berupa kotak pembatas (*bounding box*) pada rompi yang terlihat.



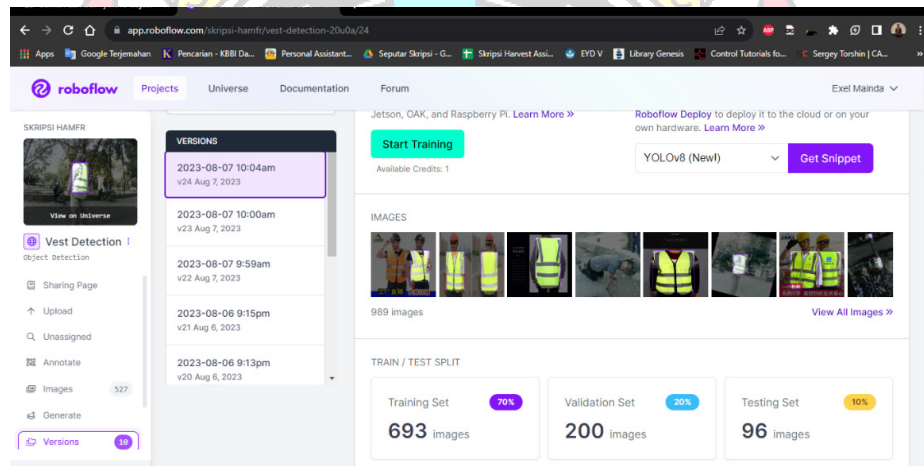
Gambar 4.9 Proses anotasi gambar

Tahap ketiga, mengatur atau membagi jumlah gambar untuk *training*, *validation*, dan *testing*. Di sini penulis membagi 231 gambar untuk *training*, 200 gambar untuk *validation*, dan 96 gambar untuk *testing*. Selanjutnya, melakukan *preprocessing* dan *augmentation* terhadap gambar-gambar tersebut. Hal ini dilakukan untuk melakukan transformasi pada gambar yang sudah ada sehingga dapat membuat variasi baru dan menambah jumlah gambar dalam *dataset*.



Gambar 4.10 Proses augmentasi gambar

Tahap terakhir yaitu *generate dataset*. Pada tahap ini, Roboflow akan memproses gambar-gambar tersebut dengan *preprocessing* dan *augmentation* yang telah diatur tadi dan juga akan menggandakan jumlah gambar (*maximum version size*). Disini penulis hanya dapat memilih versi 3x karena terbatas untuk akun pengguna gratis. Setelah selesai, proses *generate dataset* akan menghasilkan 989 gambar yang terbagi menjadi 70% (693 gambar) untuk *training*, 20% (200 gambar) untuk *validation*, dan 10% (96 gambar) untuk *test*, serta *snippet* untuk mengekspor *dataset* yang dibuat.

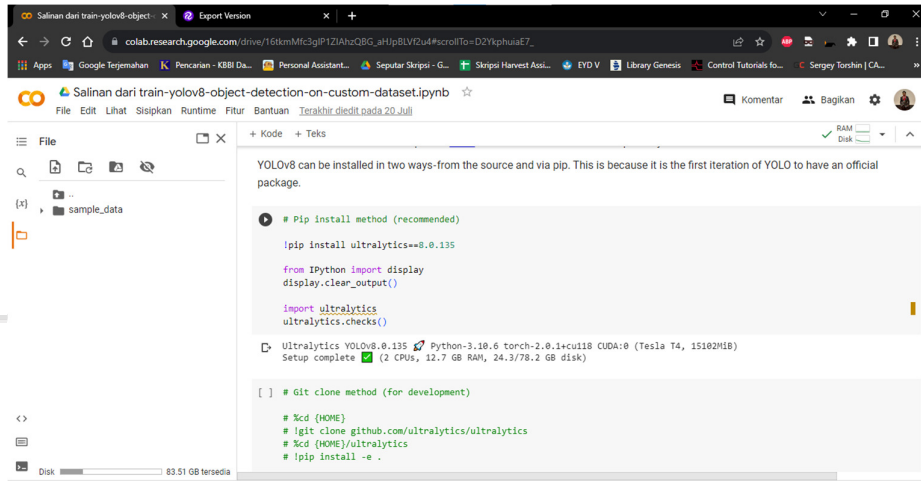


Gambar 4.11 *Dataset* yang dihasilkan

b) Proses *training*

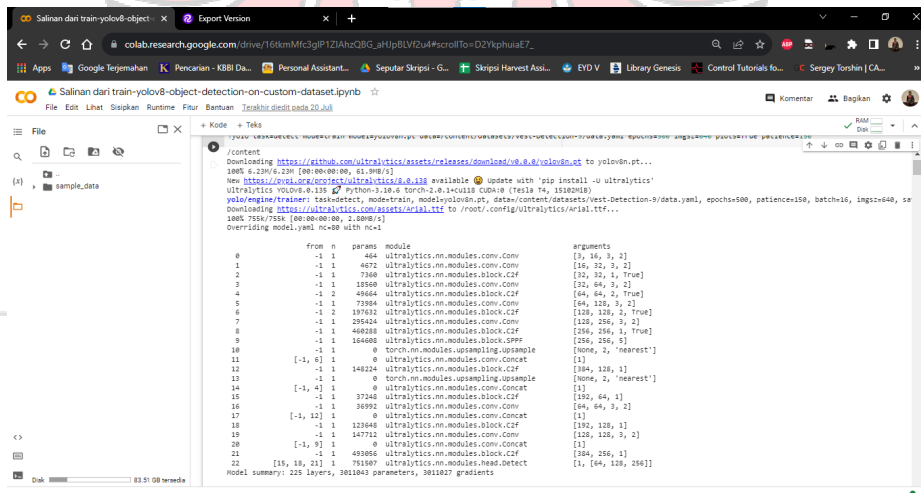
Proses *training* ini dilakukan pada laman Google Colab. Tahap pertama yaitu mengatur jenis *runtime* menjadi Python 3 dan akselerator *hardware* menjadi GPU dengan jenis GPU yaitu T4. Selanjutnya melakukan instalasi *library* yang akan digunakan yaitu Ultralytics. Saat proses instalasi, Ultralytics akan secara otomatis menginstal *library* lain yang dibutuhkan seperti Matplotlib, OpenCV, Torch, Torchvision, dll.





Gambar 4.12 Proses instalasi Ultralytics

Tahap selanjutnya yaitu menginstal *library* Roboflow dan mengimpor *dataset* yang telah dibuat. Setelah mengimpor, selanjutnya proses *training* dimulai. Pada proses *training*, digunakan *pre-trained model* yaitu YOLOv8n dengan jumlah *epochs* sebesar 500, *image size* sebesar 640, dan *patience* sebesar 150. YOLOv8n sendiri memiliki jumlah *hidden layer* sebanyak 255 lapis dan 3.011.043 parameter.



Gambar 4.13 Arsitektur YOLOv8n

Proses *training* secara otomatis selesai setelah 471 *epochs* karena tidak ada peningkatan yang diamati dalam 150 *epochs* terakhir (efek *patience*) dan memakan waktu selama 2,34 jam.

```

Epoch      GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
470/500    2.34G    0.4839    0.2589    0.8849     12         640: 100% 44/44 [00:13:00:00, 3.23it/s]
Class      Images  Instances  Box(P)    R
all         200     234       0.887    0.949    0.957    0.691

Epoch      GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
471/500    2.33G    0.4857    0.2582    0.886     14         640: 100% 44/44 [00:14:00:00, 3.07it/s]
Class      Images  Instances  Box(P)    R
all         200     234       0.891    0.944    0.958    0.695

Stopping training early as no improvement observed in last 100 epochs. Best results observed at epoch 371, best model saved as best.pt.
To update EarlyStopping(patience=100) pass a new patience value, i.e. 'patience=300' or use 'patience=0' to disable EarlyStopping.

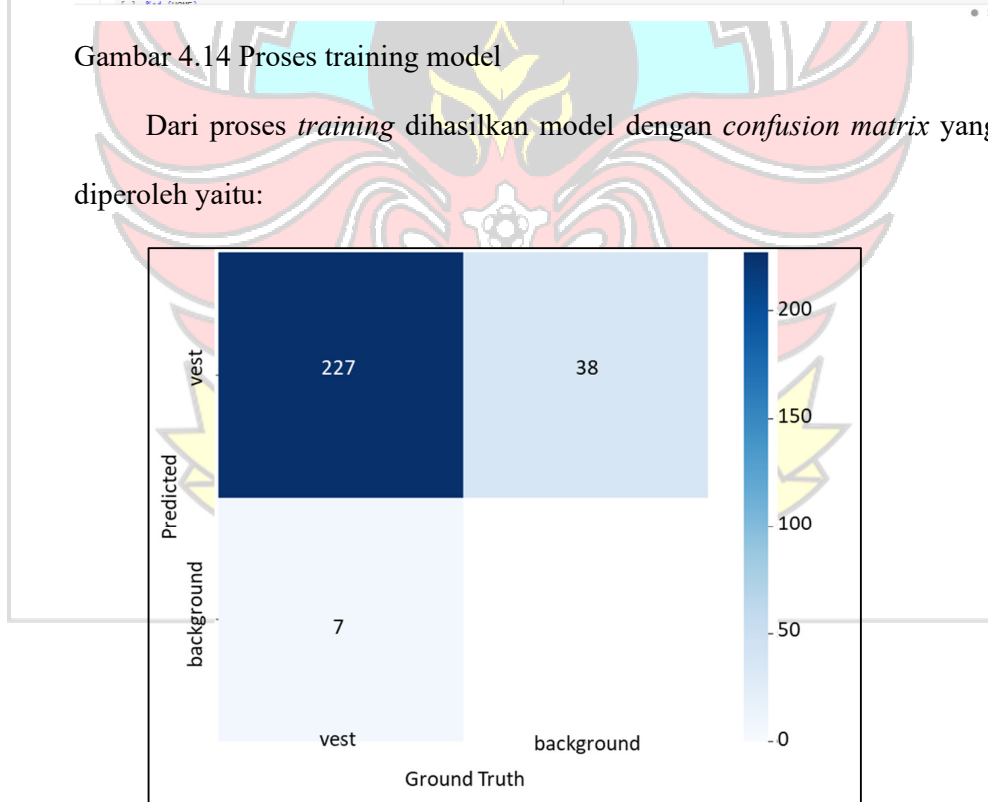
471 epochs completed in 2.338 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8 8.1.35 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3005843 parameters, 0 gradients
Class      Images  Instances  Box(P)    R
all         200     234       0.919    0.932    0.965    0.716

Speed: 3.7ms preprocess, 3.1ms inference, 0.8ms loss, 2.3ms postprocess per image
Results saved to runs/detect/train
  
```

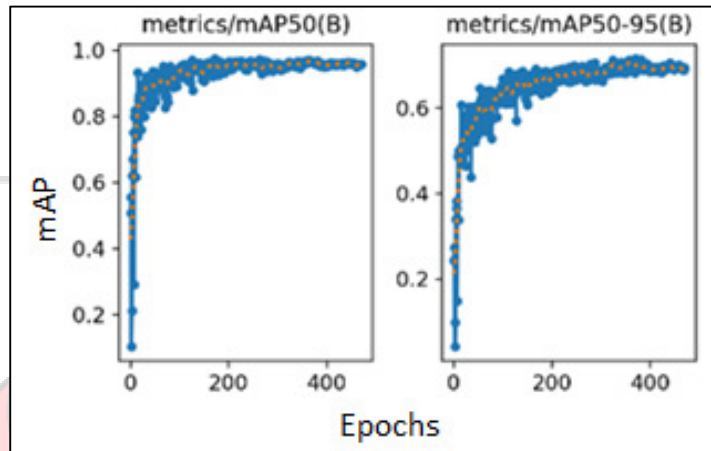
Gambar 4.14 Proses training model

Dari proses *training* dihasilkan model dengan *confusion matrix* yang diperoleh yaitu:



Gambar 4.15 Confusion matrix pada model

Selain itu, diperoleh juga nilai  $mAP@50$  sebesar 96,5% dan nilai  $mAP@50-95$  sebesar 71,6%.



Gambar 4.16 Grafik  $mAP@50$  dan  $mAP@50-95$  model

Model yang dihasilkan kemudian diunduh untuk digunakan pada perangkat kontrol Raspberry Pi 4B.

## 2) Program pada Raspberry Pi 4B

Raspberry Pi 4B digunakan untuk mengambil dan memproses data gambar ke dalam proses *computer vision* dengan metode *deep learning* yang diterima melalui Webcam, dan membuat komunikasi serial antara Raspberry Pi 4B dan Arduino Mega 2560. Selain itu, Raspberry Pi 4B juga digunakan untuk mengatur proses kerja dari komponen yang telah dirangkai. Program-program pada Raspberry Pi 4B dibuat dengan menggunakan aplikasi Visual Studio Code menggunakan bahasa Python dengan *library* tambahan seperti OpenCV.

```

le Edit Selection View Go Run Terminal Help real-time inference.py - Fix - Visual Studio C
real-time inference.py X
real-time inference.py
1 import cv2 # OpenCV library
2 import time # time library
3 from threading import Thread # library for multi-threading
4
5 from ultralytics import YOLO
6 import supervision as sv
7 import numpy as np
8
9 import RPi.GPIO as GPIO
10 from hx711 import HX711
11 GPIO.setmode(GPIO.BCM)
12 GPIO.setwarnings(False)
13 from pyfirmata import pyfirmata, ArduinoMega, util
14 import sys
15
16 # defining a helper class for implementing multi-threading
17 class WebcamStream :
18     # initialization method
19     def __init__(self, stream_id):
20         self.stream_id = stream_id # default is 0 for main camera

```

Gambar 4.17 Listing program pada Raspberry Pi

3) Program pada Arduino Mega 2560

Pemrograman pada Arduino Mega bertujuan untuk mengatur proses kerja dari aktuator robot. Arduino Mega dikontrol melalui Raspberry Pi dengan bahasa Python menggunakan *library* pyFirmata pada sisi Raspberry Pi dan *library* Firmata pada sisi Arduino Mega. Firmata adalah protokol generik untuk berkomunikasi dengan mikrokontroler dari perangkat lunak pada komputer host.

```

StandardFirmata $
1 #include <Servo.h>
2 #include <Wire.h>
3 #include <Firmata.h>
4
5 #define I2C_WRITE B00000000
6 #define I2C_READ B00001000
7 #define I2C_READ_CONTINUOUSLY B00010000
8 #define I2C_STOP_READING B00011000
9 #define I2C_READ_WRITE_MODE_MASK B00011000
10 #define I2C_10BIT_ADDRESS_MODE_MASK B00100000
11 #define I2C_END_TX_MASK B01000000
12 #define I2C_STOP_TX 1
13 #define I2C_RESTART_TX 0
14 #define I2C_MAX_QUERIES 8
15 #define I2C_REGISTER_NOT_SPECIFIED -1
16
17 // the minimum interval for sampling analog input

```

Gambar 4.18 Listing program pada Arduino Mega

#### 4) Sistem Navigasi Robot

Sensor webcam yang dihubungkan dengan Raspberry Pi akan menangkap gambar secara *real-time*. Kemudian gambar yang ditangkap oleh kamera diproses oleh Raspberry Pi melalui proses *computer vision* dengan metode *deep learning* dengan target/objek yang telah dikehendaki. Lalu Raspberry Pi akan memberikan perintah pada Arduino Mega melalui komunikasi serial sesuai dengan hasil dari proses *computer vision* tersebut. Arduino Mega kemudian memberikan umpan pada *driver* motor untuk menggerakkan motor sesuai dengan arah dari target/objek.

Mengenai arah pergerakan robot, penentuan arah pergerakan robot ditentukan berdasarkan pada posisi target/objek yang ditangkap oleh kamera. Untuk perintah gerak maju atau mundur, digunakan metode berdasarkan ukuran area objek (*bounding box*) yang terdeteksi. Bila ukuran area objek lebih kecil dari 95000 [px], maka robot akan bergerak maju. Bila ukuran area objek lebih besar dari 110000 [px], maka robot akan bergerak mundur. Selain itu, robot juga akan bergerak mundur bila sensor ultrasonic mendeteksi adanya benda di depan sejauh 18 [cm]. Sementara perintah gerak ke kiri atau ke kanan digunakan metode berdasarkan perbedaan antara titik tengah dari lebar (koordinat x) daerah tangkapan gambar (*frame*) dan titik tengah dari lebar (koordinat x) objek yang terdeteksi. Resolusi yang digunakan yaitu lebar 640 [px] dan tinggi 480 [px], sehingga titik tengah dari lebar daerah tangkapan gambar yaitu 320 [px]. Bila titik tengah lebar objek yang terdeteksi lebih kecil dari titik tengah daerah tangkapan gambar, maka robot akan bergerak ke kiri. Sedangkan bila titik tengah lebar objek yang terdeteksi lebih besar

dari titik tengah daerah tangkapan gambar, maka robot akan bergerak ke kanan. Adapun objek yang dimaksud merupakan Rompi *safety*.

Pergerakan robot menggunakan metode *differential drive steering*, yaitu dengan mengombinasikan gerakan roda kanan dan roda kiri sehingga membuat pergerakan yang diinginkan. Tabel 4.1 menunjukkan kombinasi gerakan roda kiri dan roda kanan untuk menghasilkan suatu pergerakan yang diinginkan. Pada penelitian ini kecepatan motor diatur dengan mengatur tegangan masukan masing-masing motor yang diatur dengan *Pulse Width Modulation* (PWM).

Tabel 4.1 Pergerakan Roda Robot

No.	Perintah	Roda Kiri	Roda Kanan
1.	Maju	Maju	Maju
2.	Berhenti	Berhenti	Berhenti
3.	Belok kiri	Maju lambat	Maju cepat
4.	Belok kanan	Maju cepat	Maju lambat
5.	Mundur	Mundur	Mundur

#### 4.1.4 Hasil Pengujian

Adapun hasil yang diperoleh berdasarkan pengujian yang telah dilakukan yaitu sebagai berikut:

- 1) Perbandingan kecepatan pemrosesan data model *deep learning* terhadap input *image size*.

Pengujian ini dilakukan dengan 6 kali percobaan dengan input *image size* model yang bervariasi dan parameter *confidence threshold* 50% serta *IoU Threshold* 70% sehingga mendapatkan hasil berupa:

Tabel 4.2 Perbandingan kecepatan pemrosesan data model *deep learning* terhadap input *image size*.

Percobaan	Image size model [px]	Latensi [ms]	Jarak maksimal objek [m]
1	640	± 800	± 5,5
2	480	± 600	± 4,7
3	320	± 400	± 3,6
4	256	± 300	± 2,4
5	224	± 200	± 2,2
6	192	± 150	± 1,6

Berdasarkan hasil di atas, maka dipilihlah *image size* berukuran 224 [px] karena memiliki latensi yang cukup rendah dan jarak maksimal objek yang cukup memadai. Dengan demikian, pengujian selanjutnya akan menggunakan *image size* berukuran 224 [px].

2) Perbandingan pendeteksian rompi terhadap intensitas cahaya (lux).

Pengujian ini dilakukan dengan 8 kali percobaan dengan intensitas cahaya yang bervariasi dengan parameter *image size* 224 [px], *confidence threshold* 50%, dan *IoU threshold* 70%. Pengukuran nilai intensitas cahaya menggunakan sensor cahaya pada ponsel Xiaomi Redmi Note 10 Pro sehingga mendapatkan hasil berupa:

Tabel 4.3 Perbandingan pendeteksian rompi terhadap intensitas cahaya

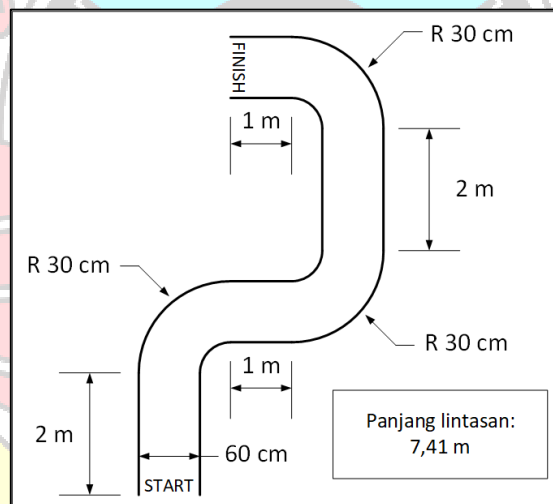
Percobaan	Waktu	Lux [lx]	Hasil Pembacaan
1	10.05	2041	Terdeteksi
2	12.25	83	Terdeteksi
3	12.55	5981	Terdeteksi

Lanjutan tabel 4.3

4	14.13	70	Terdeteksi
5	15.19	1429	Terdeteksi
6	16.14	63	Terdeteksi
7	17.50	11	Terdeteksi
8	18.46	0	Tidak terdeteksi

3) Perbandingan kecepatan rata-rata robot terhadap beban angkut.

Pengujian ini dilakukan pada medan yang datar dan sedikit bergelombang. Adapun bentuk lintasan untuk kedua medan tersebut dapat dilihat pada Gambar 4.18.



Gambar 4.19 Bentuk lintasan

A) Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan yang datar.

Pengujian ini dilakukan dengan 12 kali percobaan dengan beban angkut yang bervariasi dari 0-5 [kg] dengan hasil berupa:



Tabel 4.4 Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan yang datar.

Percobaan	Beban Angkut [kg]	Tegangan		Waktu Tempuh [s]
		Awal [V]	Akhir [V]	
1	0	12,6	12,6	24,73
2	0	12,6	12,5	27,36
3	1	12,4	12,3	24,23
4	1	12,5	12,5	28,5
5	2	12,8	12,7	23,09
6	2	12,7	12,7	31,99
7	3	12,6	12,5	30,73
8	3	12,6	12,5	26,75
9	4	12,5	12,4	25,05
10	4	12,4	12,3	22,23
11	5	12,2	12,1	29,47
12	5	12,2	12,1	24,59

Dari data Tabel 4.4 dapat dihitung kecepatan rata-rata yang dihasilkan oleh robot terhadap beban angkut pada medan yang datar dengan panjang lintasan ( $s$ ) 7,41 [m] yaitu:

a) Pada beban angkut 0 [kg]:

$$t_0 = \frac{\sum t_0}{2} = \frac{24,73+27,36}{2} = 26,04 \text{ [s]}$$

$$v_0 = \frac{s}{t_0} = \frac{7,41}{26,04} = 0,28 \text{ [m/s]}$$

b) Pada beban angkut 1 [kg]:

$$t_1 = \frac{\sum t_1}{2} = \frac{24,23+28,5}{2} = 26,36 \text{ [s]}$$

$$v_1 = \frac{s}{t_1} = \frac{7,41}{26,36} = 0,28 \text{ [m/s]}$$

c) Pada beban angkut 2 [kg]:

$$t_2 = \frac{\sum t_2}{2} = \frac{23,09+31,99}{2} = 27,54 \text{ [s]}$$

$$v_2 = \frac{s}{t_2} = \frac{7,41}{27,54} = 0,27 \text{ [m/s]}$$

d) Pada beban angkut 3 [kg]:

$$t_3 = \frac{\sum t_3}{2} = \frac{30,73+26,75}{2} = 28,74 \text{ [s]}$$

$$v_3 = \frac{s}{t_3} = \frac{7,41}{28,74} = 0,26 \text{ [m/s]}$$

e) Pada beban angkut 4 [kg]:

$$t_4 = \frac{\sum t_4}{2} = \frac{25,05+22,23}{2} = 23,64 \text{ [s]}$$

$$v_4 = \frac{s}{t_4} = \frac{7,41}{23,64} = 0,31 \text{ [m/s]}$$

f) Pada beban angkut 5 [kg]:

$$t_5 = \frac{\sum t_5}{2} = \frac{29,47+24,59}{2} = 27,03 \text{ [s]}$$

$$v_5 = \frac{s}{t_5} = \frac{7,41}{27,03} = 0,27 \text{ [m/s]}$$

B) Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan sedikit bergelombang.

Pengujian ini dilakukan dengan 12 kali percobaan dengan beban angkut yang bervariasi dari 0-5 [kg] dengan hasil berupa:

Tabel 4.5 Perbandingan kecepatan rata-rata robot terhadap beban angkut pada medan sedikit bergelombang.

Percobaan	Beban Angkut [kg]	Tegangan		Waktu Tempuh [s]
		Awal [V]	Akhir [V]	
1	0	12,5	12,4	32,55
2	0	12,5	12,4	33,74

Lanjutan tabel 4.5

3	1	12,4	12,4	32,24
4	1	12,4	12,4	29,24
5	2	12,9	12,8	22,24
6	2	12,8	12,7	27,23
7	3	12,7	12,5	32,01
8	3	12,6	12,5	31,23
9	4	12,6	12,5	26,48
10	4	12,5	12,4	38,73
11	5	12,7	12,7	22,23
12	5	12,7	12,6	24,23

Dari data Tabel 4.5 dapat dihitung kecepatan rata-rata yang dihasilkan oleh robot terhadap beban angkut pada medan yang sedikit bergelombang dengan panjang lintasan ( $s$ ) 7,41 [m] yaitu:

a) Pada beban angkut 0 [kg]:

$$t_0 = \frac{\sum t_0}{2} = \frac{32,55+33,74}{2} = 33,14 \text{ [s]}$$

$$v_0 = \frac{s}{t_0} = \frac{7,41}{33,14} = 0,22 \text{ [m/s]}$$

b) Pada beban angkut 1 [kg]:

$$t_1 = \frac{\sum t_1}{2} = \frac{32,24+29,24}{2} = 30,74 \text{ [s]}$$

$$v_1 = \frac{s}{t_1} = \frac{7,41}{30,74} = 0,24 \text{ [m/s]}$$

c) Pada beban angkut 2 [kg]:

$$t_2 = \frac{\sum t_2}{2} = \frac{22,24+27,23}{2} = 24,73 \text{ [s]}$$

$$v_2 = \frac{s}{t_2} = \frac{7,41}{24,73} = 0,3 \text{ [m/s]}$$

d) Pada beban angkut 3 [kg]:

$$t_3 = \frac{\sum t_3}{2} = \frac{32,01+31,23}{2} = 31,62 \text{ [s]}$$

$$v_3 = \frac{s}{t_3} = \frac{7,41}{31,62} = 0,23 \text{ [m/s]}$$

e) Pada beban angkut 4 [kg]:

$$t_4 = \frac{\sum t_4}{2} = \frac{26,48+38,73}{2} = 32,6 \text{ [s]}$$

$$v_4 = \frac{s}{t_4} = \frac{7,41}{32,6} = 0,23 \text{ [m/s]}$$

f) Pada beban angkut 5 [kg]:

$$t_5 = \frac{\sum t_5}{2} = \frac{22,23+24,23}{2} = 23,23 \text{ [s]}$$

$$v_5 = \frac{s}{t_5} = \frac{7,41}{23,23} = 0,32 \text{ [m/s]}$$

## 4.2 Pembahasan

1) Hasil *training* model *deep learning*.

Dari data hasil pembuatan model *deep learning* yang terdapat pada Gambar 4.15, dapat diketahui nilai *True Positive* (TP) sebesar 227, nilai *False Positive* (FP) sebesar 38, nilai *False Negative* (FN) sebesar 7, dan nilai *True Negative* sebesar 0. Berdasarkan data tersebut, dapat diperoleh nilai *Accuracy* dan nilai *Precision*. “*Accuracy* adalah parameter atau *metric* seberapa akurat model mengklasifikasikan dengan benar” (Ardikatama, H., 2021). Secara matematis, *metric* ini dapat diperoleh dengan rumus:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% = \frac{(227+0)}{(227+38+7+0)} \times 100\% = 83,45\%$$

“*Precision* adalah rasio prediksi benar dibagi hasil seluruh prediksi pada kelas positif yang mana hal ini dapat diartikan seberapa akurat model memprediksi kelas tersebut” (Ardikatama, H., 2021). Secara matematis, *metric* ini dapat diperoleh dengan rumus:

$$Precision = \frac{TP}{TP+FP} \times 100\% = \frac{227}{227+38} \times 100\% = 85,66\%$$

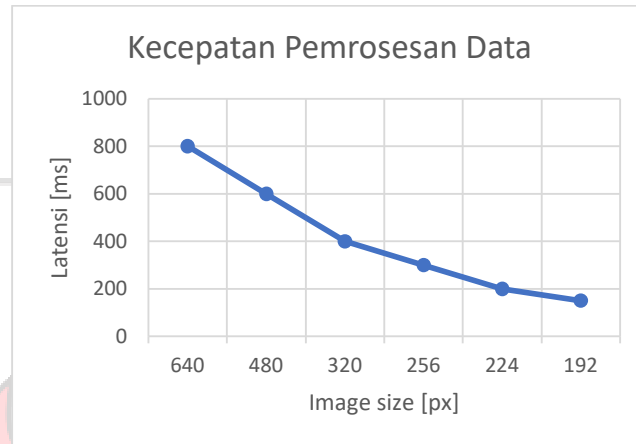
Sehingga nilai *accuracy* yang dimiliki oleh model yaitu 83,45% dan nilai *precision* 85,66%. Selain kedua parameter tadi, diperoleh juga nilai *mAP* (*Mean Average Precision*). “*mAP* adalah parameter yang biasa digunakan untuk melihat performa model *object detection*. *mAP* menggambarkan seberapa presisi model kita memprediksi setiap kemungkinan yang ada pada data *test* pada nilai batasan (*threshold*) *IoU* yang ditentukan” (Ardikatama, H., 2021). *IoU threshold* sendiri merupakan suatu proses pengukuran berbasis *Jaccard Index* untuk mengevaluasi overlap di antara *ground truth bounding boxes* (*bounding boxes* yang memang ada pada *dataset*) dan *bounding boxes* prediksi.

Nilai *mAP* dihitung dari setiap *epochs* yang terjadi dari awal *training* hingga *training* selesai. *mAP@50* berarti seberapa besar nilai *mAP* dengan *IoU threshold* 50% dan *mAP@50-95* berarti seberapa besar nilai *mAP* dengan *IoU threshold* 50-95%. Adapun nilai *mAP@50* sebesar 96,5% dan nilai *mAP@50-95* sebesar 71,6%.

2) Perbandingan kecepatan pemrosesan data model *deep learning* terhadap input *image size*.

Berdasarkan data pengujian yang telah penulis lakukan, dapat diketahui kecepatan pemrosesan data model *deep learning* yang dilakukan pada perangkat

kontrol Raspberry Pi 4B. Adapun grafik perbandingan kecepatan pemrosesan data yaitu:



Gambar 4.20 Grafik kecepatan pemrosesan data

Dari Gambar 4.20 dapat diketahui bahwa ukuran *image size* pada model *deep learning* mempengaruhi kecepatan pemrosesan datanya. Sehingga semakin besar ukuran *image size* yang digunakan, maka semakin lambat pemrosesan datanya. Selain itu, kecepatan pemrosesan data tersebut berbanding lurus dengan jarak maksimal objek yang dapat dideteksi.

3) Perbandingan pendeteksian rompi terhadap intensitas cahaya (lux).

Berdasarkan data pengujian yang telah penulis lakukan, dapat diketahui bahwa intensitas cahaya (lux) mempengaruhi kemampuan robot dalam mendeteksi rompi. Adapun besarnya intensitas cahaya yang dapat diterima oleh robot yaitu dari 11 [lx] hingga 5981 [lx].

- 4) Perbandingan kecepatan rata-rata robot terhadap beban angkut dengan medan yang datar dan sedikit bergelombang.

Berdasarkan data pengujian yang telah penulis lakukan, robot bergerak melalui lintasan yang berkelok sejauh 7,41 [m] pada medan yang datar dengan kecepatan yang cukup stabil dari 0,26 – 0,31 [m/s] dan pada medan yang sedikit bergelombang dengan kecepatan 0,22 – 0,32 [m/s].



## BAB V PENUTUP

### 5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Rancang bangun robot ini memakai Raspberry Pi 4B sebagai *primary control device* untuk tempat pengolahan data dan pengontrolan robot menggunakan program kecerdasan buatan *deep learning* dan juga memakai Arduino Mega sebagai *secondary control device* untuk menggerakkan roda yang dihubungkan dengan motor DC.
2. Robot ini telah mampu untuk membuat prediksi gerakan dari data gambar *real-time* yang diolah oleh algoritma *deep learning* dalam sistem *computer vision* sehingga robot mampu menentukan kecepatan dan arah gerakannya agar sesuai dengan lintasan yang dilalui dan beban yang diangkut. Keberhasilan robot untuk memprediksi gerakannya dipengaruhi oleh intensitas cahaya.

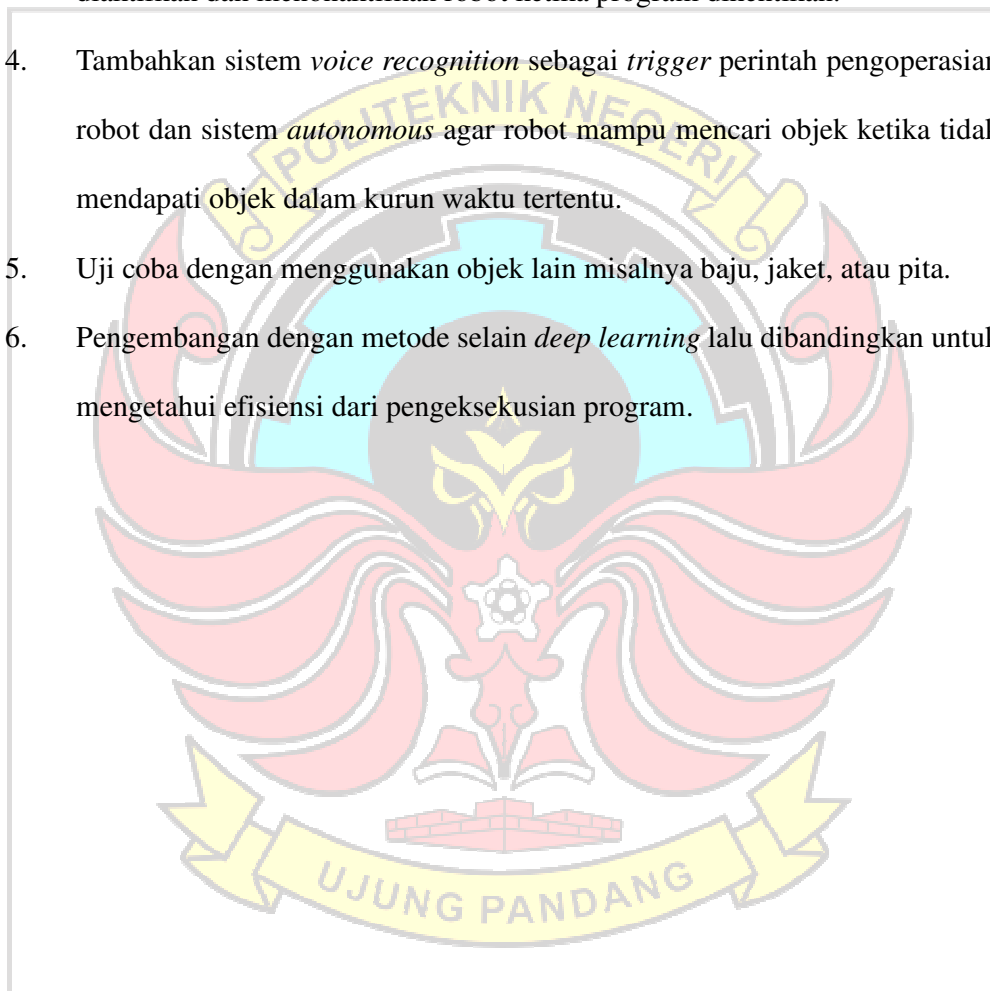
### 5.2 Saran

Rancang Bangun *Harvest Assisting Mobile Field Robot* Berbasis *Computer Vision* dengan Metode *Deep Learning* ini masih memiliki banyak kekurangan, oleh sebab itu saran ke depannya untuk penelitian ini adalah sebagai berikut:

1. Tingkatkan kualitas kamera yang digunakan seperti adanya kemampuan *auto exposure* dan kejernihan gambar agar input gambar *real-time* terlihat lebih jelas.



2. Gunakan *control device* yang memiliki spesifikasi lebih tinggi dari Raspberry Pi 4B terutama dalam segi GPU seperti agar kecepatan pemrosesan data *deep learning* lebih cepat.
3. Buat mekanisme agar robot langsung mengeksekusi program ketika robot diaktifkan dan menonaktifkan robot ketika program dihentikan.
4. Tambahkan sistem *voice recognition* sebagai *trigger* perintah pengoperasian robot dan sistem *autonomous* agar robot mampu mencari objek ketika tidak mendapati objek dalam kurun waktu tertentu.
5. Uji coba dengan menggunakan objek lain misalnya baju, jaket, atau pita.
6. Pengembangan dengan metode selain *deep learning* lalu dibandingkan untuk mengetahui efisiensi dari pengeksekusian program.



## DAFTAR PUSTAKA

Afghani, S. dkk. 2013. Follow Me Robot Using Infrared Beacons. *SAVAP Internasional Journal*. 4(3): 63–72.

Ardikatama, H. 2021. Jenis-Jenis Performance Metrics yang Wajib Kamu Ketahui!. (Daring), (<https://medium.com/>), diakses 10 Agustus 2023.

Awal, S. M. S. dkk. 2017. Sistem Navigasi Berbasis Pelacakan Objek pada Operasi Robot Asisten Pemanenan di dalam *Greenhouse*. Prosiding. *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*. 17–21.

Ballard, D. H. dkk. 1983. *Parallel Vision Computation*. Edisi ke-306. 21–26. California: Springer Nature.

Bien Xuan Duong dkk. 2021. Deep Learning based 6DOF Robot Arm Control with Computer Vision. *Internasional Journal of Engineering Research & Technology (IJERT)*. 10(2): 531–536.

Budiono, M. 2021. Rancang Bangun Robot Penyemprot Pestisida Otonom dengan Sistem Wall-follower pada Penyemprotan Tanaman Cabai. Skripsi. Pekanbaru: Universitas Islam Negeri Sultan Syarif Kasim Riau.

Davis, A. dkk. 2014. The Visual Microphone: Passive Recovery of Sound From Video. Prosiding. *ACM Transactions on Graphics*. 33(4). 79:1–79:10.

Febrianti dan Indarwati. 2022. Pengembangan Robot Active Caster Berbasis Deep Learning dan Internet of Things Untuk Aplikasi Warehouse. Skripsi. Makassar: Politeknik Negeri Ujung Pandang.

Goodfellow, I. dkk. 2016. *Deep Learning*. Cambridge: The MIT Press.

Jocher, G. dkk. 2023. Ultralytics YOLOv8

Naufalimam, S. M. dkk. 2021. Penerapan Deteksi dan Pelacakan Objek pada Robot Otonom Pengumpul Bola Tennis Meja Menggunakan Pengolahan Citra. Prosiding. *e-Proceeding of Engineering*. 8(5): 4355.

Riadi, M. 2020. Raspberry Pi (Definisi, Fungsi, Jenis, Spesifikasi dan Pemrograman). (Daring), (<https://www.kajianpustaka.com/>), diakses 29 Januari 2023.

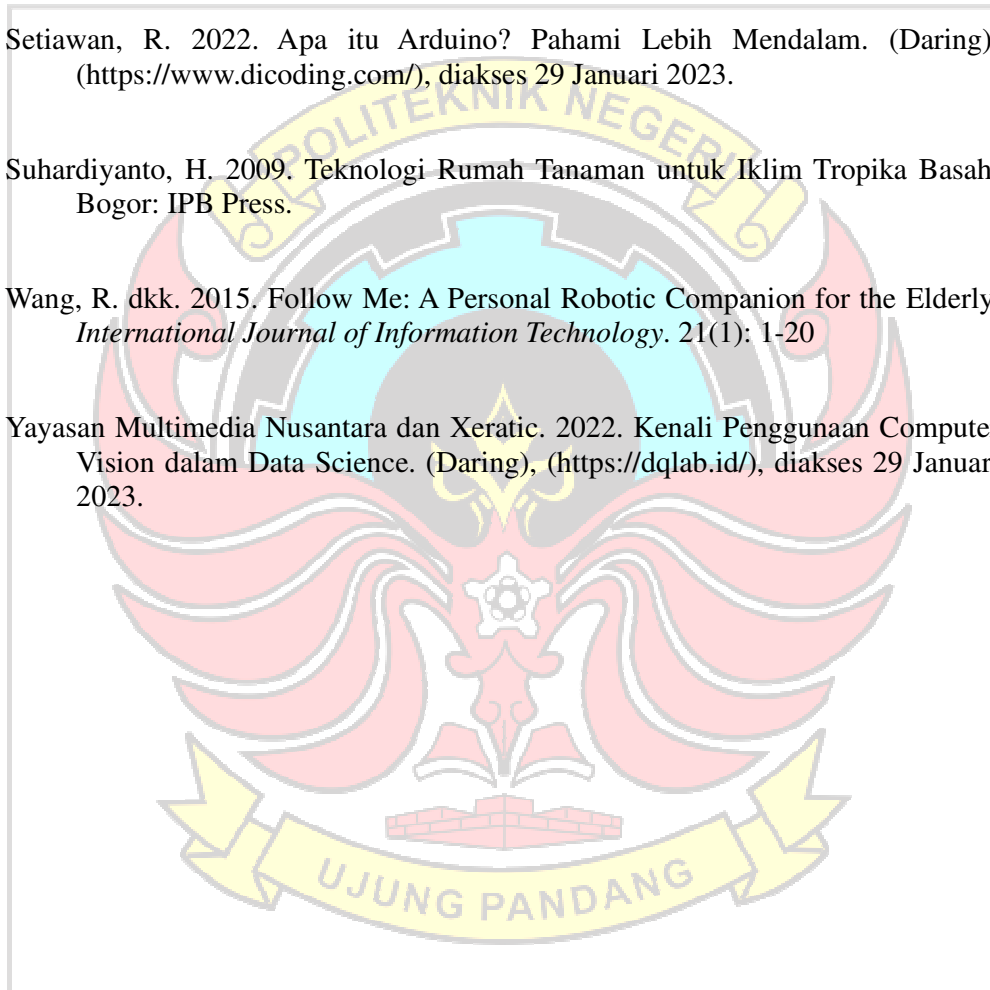
Rusdi, M. dkk. 2016. Pedoman Penulisan Proposal dan Skripsi Program Diploma (D-4) Bidang Rekayasa dan Tata Niaga. Makassar: Politeknik Negeri Ujung Pandang.

Setiawan, R. 2022. Apa itu Arduino? Pahami Lebih Mendalam. (Daring), (<https://www.dicoding.com/>), diakses 29 Januari 2023.

Suhardiyanto, H. 2009. Teknologi Rumah Tanaman untuk Iklim Tropika Basah. Bogor: IPB Press.

Wang, R. dkk. 2015. Follow Me: A Personal Robotic Companion for the Elderly. *International Journal of Information Technology*. 21(1): 1-20

Yayasan Multimedia Nusantara dan Xeratic. 2022. Kenali Penggunaan Computer Vision dalam Data Science. (Daring), (<https://dqlab.id/>), diakses 29 Januari 2023.





Lampiran 1: Lembar Asistensi Pembimbing I



**JURUSAN TEKNIK MESIN  
PROGRAM STUDI SARJANA TERAPAN TEKNIK MEKATRONIKA  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR 2023**

**KARTU ASISTENSI**

Judul Tugas Akhir : "RANCANG BANGUN *HARVEST ASSISTING MOBILE FIELD ROBOT* BERBASIS *COMPUTER VISION* DENGAN METODE *DEEP LEARNING*"

Nama : Franklin Delano Exel Mainda 444 19 006

Kelas : 4A Teknik Mekatronika

Dosen Pembimbing I : Dr.Eng. Abdul Kadir Muhammad, S.T., PG.Dipl., M.Eng.

Dosen Pembimbing II : Dr. Ir. Simon Ka'ka, M.T.

No	Tanggal	Kegiatan	Uraian/Revisi	Paraf Pembimbing
1	Kamis <del>Selasa</del> 23/3/2023	Seminar Progress I	- Rancangan mekanik, elektronik - RAB	AB
2	Jumat 14/4/23	Seminar Progress II	- Revisi RAB - Pengecekan komponen sebelumnya - Pengadaan alat dan bahan	AB
3	Kamis 27/4/23	Seminar Progress III	- Pengadaan dan pengecekan alat dan bahan - Pengerjaan mekanik	AB
4	27/5/23	Seminar Progress IV	- Perakitan rangka, bending plat Pembuatan as roda - Pengerjaan perangkat lunak (YoloV8)	AB



JURUSAN TEKNIK MESIN  
PROGRAM STUDI SARJANA TERAPAN TEKNIK MEKATRONIKA  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR 2023

5	Sabtu, 10/6/23	Seminar Progress V	- Pembuatan program pendeteksian rompi (vision) - Program motor - Pergerakan wiring, as roda.	AB
6	Senin, 26/6/23	Seminar Progress VI	- Pemrograman load cell dan buzzer - Pergerakan tiang untuk kamera dan sensor - Pengujian elektronik dan mekanik.	AB
7	Selasa, 11/7/23	Seminar Progress VII	- <del>Senas</del> perbaikan roda (balancing) - Lanjut program vision -	AB
8	Senin, 7/8/23	Asistensi	- Lanjut pengambilan data	AB
9	Kamis 10/8/23	Asistensi	- Asistensi RGB IV - Video uji coba	AB
10	11/8/23	Kasihan	App untuk vision	AB

Disahkan, 11 Agustus 2023  
Dosen Pembimbing I

*Abdul Kadir*

**Dr.Eng. Abdul Kadir Muhammad,**  
**S.T., PG.Dipl., M.Eng.**

NIP. 19750402 200312 1 002

Lampiran 2: Lembar Asistensi Pembimbing II



JURUSAN TEKNIK MESIN  
PROGRAM STUDI SARJANA TERAPAN TEKNIK MEKATRONIKA  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR 2023

**KARTU ASISTENSI**

Judul Tugas Akhir : "RANCANG BANGUN *HARVEST ASSISTING MOBILE FIELD ROBOT* BERBASIS *COMPUTER VISION* DENGAN METODE *DEEP LEARNING*"

Nama : Franklin Delano Exel Mainda 444 19 006

Kelas : 4A Teknik Mekatronika

Dosen Pembimbing I : Dr.Eng. Abdul Kadir Muhammad, S.T., PG.Dipl., M.Eng.

Dosen Pembimbing II : Dr. Ir. Simon Ka'ka, M.T.

No	Tanggal	Kegiatan	Uraian/Revisi	Paraf Pembimbing
1	02/4/2023	Asistensi	penjastangan identifikasi fungsi komponen	
2	06/4/2023	Asistensi	* Skema gambar struktur komponen mekanik per selok	
3	14/4/2023	Asistensi	+ RAB * pengadaan alat/bhn	
4	27/5/2023	Asistensi	+ pengerjaan permesin + pemasangan di lab. * instal library	





JURUSAN TEKNIK MESIN  
PROGRAM STUDI SARJANA TERAPAN TEKNIK MEKATRONIKA  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR 2023

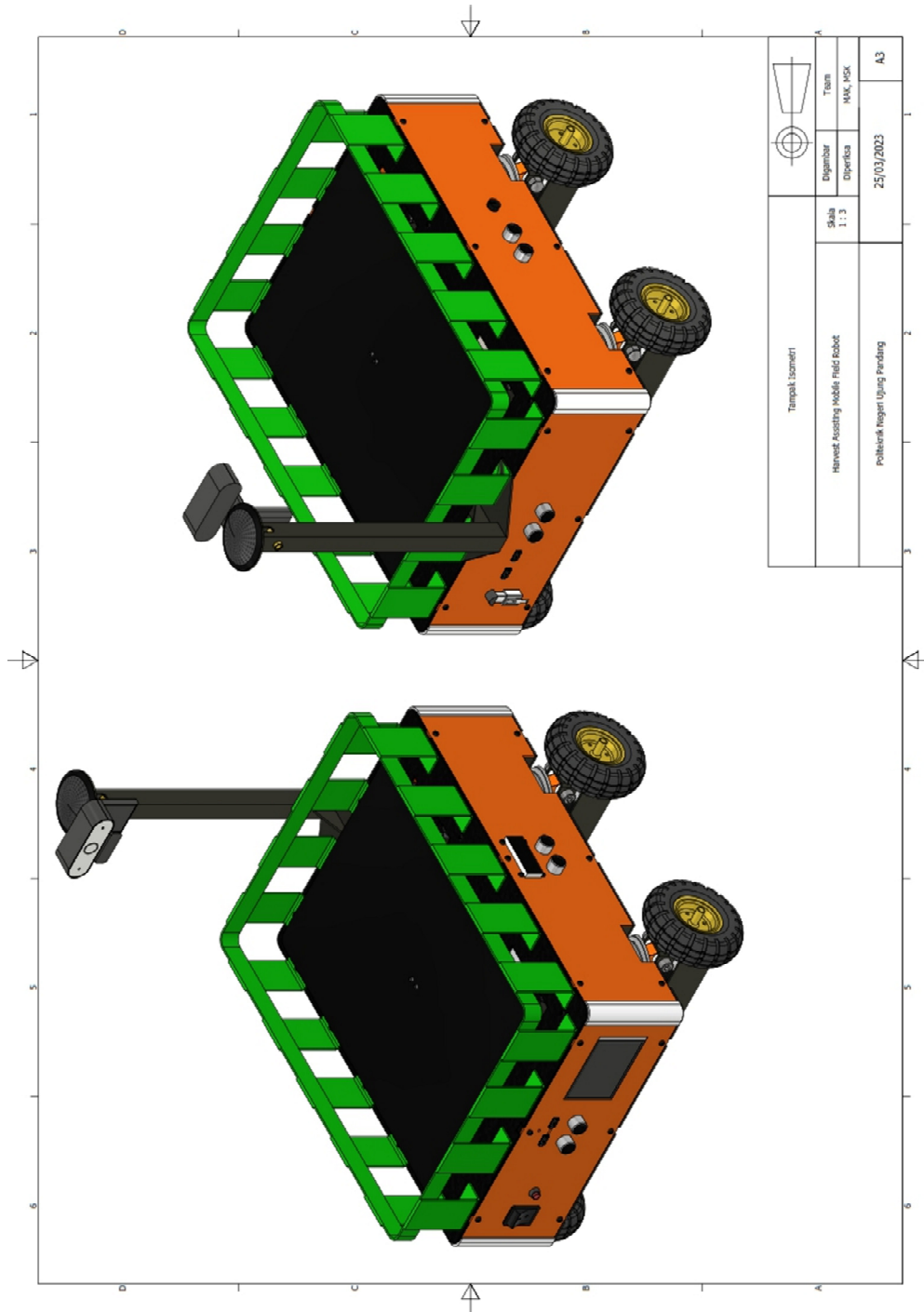
5	10/06/2023	Asisten	- Instalasi wiring - pengujian - pembuatan PCB. - Uji putaran motor	
6	26/06/2023	Asisten	- Mandukasi rumus - program wordCell & buzzer. - pengujian motor.	
7	10/07/2023	Asisten	- wiring - koneksi USB. - indikator daya. - pengujian daya.	
8	3/8/2023	Asisten	- perbaikan Tabel <sup>2</sup> - Lampiran <sup>2</sup>	
9	14/8/2023	Asisten	All of seminar kamu	
10				

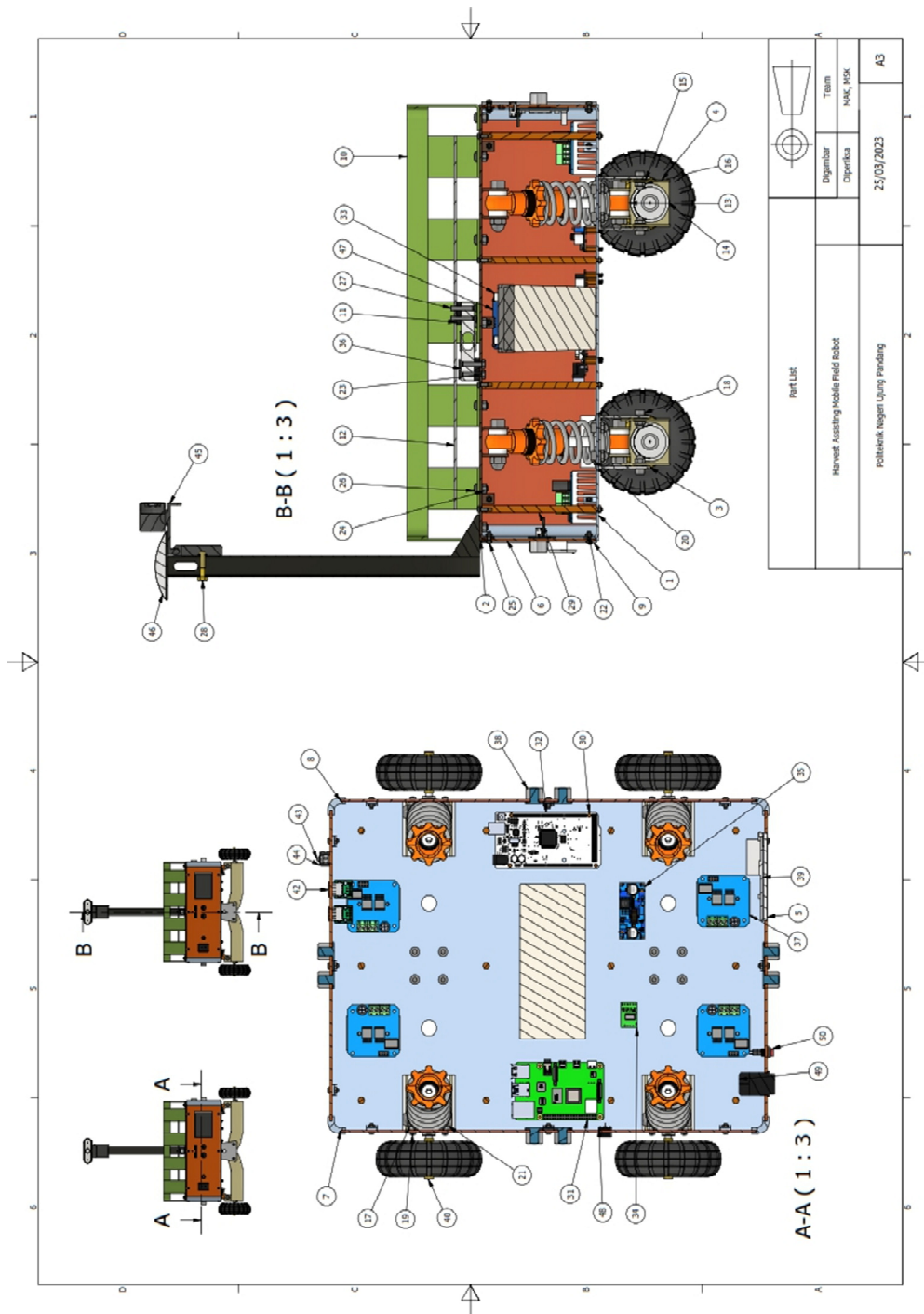
Disahkan, 14 Agustus 2023  
Dosen Pembimbing II

**Dr. Ir. Simon Ka'ka, M.T.**  
NIP. 19590913 198803 1 001




Lampiran 3: Design Robot





Revisi List		Disiplin	Team
Harvest Assisting Mobile Field Robot		Diperiksa	MAR, RSK
Politeknik Negeri Lampung Pandang		25/03/2023	
			A3

No.	Qty	Nama Bagian	Material	Keterangan
1	1	Lower Plate	Steel, Mild	Plat 2mm
2	1	Upper Plate	Steel, Mild	Plat 2mm
3	2	Arm Bracket	Steel, Mild	Plat 2mm
4	4	Swing Arm	Steel, Mild	Plat 2mm
5	1	Front Cover	Acrylic	2mm
6	1	Rear Cover	Acrylic	2mm
7	1	Right Cover	Acrylic	2mm
8	1	Left Cover	Acrylic	2mm
9	4	Corner Cover	PLA	3D Print
10	1	Rallings	Steel, Mild	Plat 2mm
11	2	Loadcell Stand	Steel, Mild	Plat 2mm
12	1	Basket Stand	Steel, Mild	Plat 2mm
13	4	Bike Rear Shock Absorber	Steel	125mm
14	4	Motor DC 37GB	Steel	12V
15	8	SKF618/5	Steel, Mild	Deep groove ball bearings
16	16	KS B 1026 Type 3 - M5	Steel, Mild	Hexagon Domed Cap Nuts
17	8	KS B 1026 Type 3 - M8	Steel, Mild	Hexagon Domed Cap Nuts
18	8	ISO 4762 - M5 x 10	Stainless Steel	Hex socket head cap screws
19	8	ISO 4762 - M3 x 8	Stainless Steel	Hex socket head cap screws
20	8	ISO 4762 - M5 x 8	Stainless Steel	Hex socket head cap screws
21	8	ISO 4762 - M8 x 35	Stainless Steel	Hex socket head cap screws
22	37	ISO 4032 - M3	Stainless Steel	Hexagon nuts, style 1
23	2	ISO 4032 - M4	Stainless Steel	Hexagon nuts, style 1
24	18	ISO 4032 - M5	Stainless Steel	Hexagon nuts, style 1
25	40	ISO 7045 - M3 x 8 - 4.8 - H	Steel	Pan head screw
26	18	ISO 7045 - M5 x 10 - 4.8 - H	Steel	Pan head screw
27	4	ISO 7046-1 - M4 x 25 - 4.8 - H	Steel	Countersunk flat head screws
28	1	ISO 4015 - M5 x 25	Stainless Steel	Hexagon head bolts
29	24	Spacer Bolts M3 L55	Brass	
30	10	Spacer Bolts M3 L10	Brass	
31	1	Raspberry Pi 4 Model B	Composite	RAM 4GB
32	1	Arduino Mega 2560	Composite	
33	1	Battery 12V 9AH	Lead Acid	
34	1	Modul HX711	Composite	
35	1	Modul XL4015	Composite	Step Down DC-DC
36	1	Sensor Loadcell 10kg	Composite	Weight Sensor
37	4	Modul BTS7960	Composite	Motor Driver
38	4	Modul HC-SR04	Composite	Distance Sensor
39	1	LCD TFT Touchscreen 3.5"	Composite	
40	4	Roda Karet dia 10cm	Rubber	
41	1	Tiang Kamera dan Mikrofon	Steel	Hollow 30mm x 30mm
42	4	Modul USB A Female	Composite	
43	1	Socket Aki 2P	Composite	
44	1	Pelindung Socket	Rubber	
45	1	Eyesecc Webcam	Composit	1080P
46	1	Talfaudio Mic	Steel	Omnidirectional
47	1	Modul 7 Segment	Composit	4 Digit
48	1	Active Buzzer	Composit	5V
49	1	Switch DPDT	Composit	ON/OFF Rocker
50	1	Push Button	Composit	

Part List			
Harvest Assisting Mobile Field Robot		Digambar	Team
		Diperiksa	MAK, MSK
Politeknik Negeri Ujung Pandang		A4	

Lampiran 4: Listing Program pada Raspberry Pi 4B

```
import cv2
import time
from threading import Thread
from ultralytics import YOLO
import supervision as sv
import numpy as np
import RPi.GPIO as GPIO
from hx711 import HX711
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
from pyfirmata import pyfirmata, ArduinoMega, util
import sys

class WebcamStream :
    def __init__(self, stream_id):
        self.stream_id = stream_id # default is 0 for main camera
        self.vcap = cv2.VideoCapture(self.stream_id, cv2.CAP_V4L2)
        self.vcap.set(cv2.CAP_PROP_GAIN, 0)
        self.vcap.set(cv2.CAP_PROP_BRIGHTNESS, 0)
        self.vcap.set(cv2.CAP_PROP_CONTRAST, 0)
        self.vcap.set(cv2.CAP_PROP_FRAME_WIDTH, frame_width)
        self.vcap.set(cv2.CAP_PROP_FRAME_HEIGHT, frame_height)
        self.vcap.set(cv2.CAP_PROP_FPS, 10)
        if self.vcap.isOpened() is False :
            print("[Exiting]: Error accessing webcam stream.")
            exit(0)
        self.grabbed , self.frame = self.vcap.read()
        if self.grabbed is False :
            print("[Exiting] No more frames to read")
            exit(0)
        self.stopped = True
        self.t = Thread(target=self.update, args=())
        self.t.daemon = True # daemon threads run in background
    def start(self):
        self.stopped = False
        self.t.start()
    def update(self):
        while True :
            if self.stopped is True :
                break
            self.grabbed , self.frame = self.vcap.read()
            if self.grabbed is False :
                print("[Exiting] No more frames to read")
                self.stopped = True
                break
```

```

        self.vcap.release()
    def read(self):
        return self.frame
    def stop(self):
        self.stopped = True
"===== Arduino initialization ====="
board = ArduinoMega("/dev/ttyUSB0")
it = util.Iterator(board)
it.start()
print("Communication Successfully started")
"===== MOTOR =====" # Deklarasi pin motor
# MOTOR 1 (Kiri Depan)
R_PWM1 = board.get_pin("d:12:p")
L_PWM1 = board.get_pin("d:13:p")
R_EN1 = board.get_pin("d:52:o")
R_EN1.write(1)
L_EN1 = board.get_pin("d:53:o")
L_EN1.write(1)
# MOTOR 2 (Kanan Depan)
R_PWM2 = board.get_pin("d:8:p")
L_PWM2 = board.get_pin("d:9:p")
R_EN2 = board.get_pin("d:30:o")
R_EN2.write(1)
L_EN2 = board.get_pin("d:31:o")
L_EN2.write(1)
# MOTOR 3 (Kiri Belakang)
R_PWM3 = board.get_pin("d:2:p")
L_PWM3 = board.get_pin("d:3:p")
R_EN3 = board.get_pin("d:50:o")
R_EN3.write(1)
L_EN3 = board.get_pin("d:51:o")
L_EN3.write(1)
# MOTOR 4 (Kanan Belakang)
R_PWM4 = board.get_pin("d:6:p")
L_PWM4 = board.get_pin("d:7:p")
R_EN4 = board.get_pin("d:32:o")
R_EN4.write(1)
L_EN4 = board.get_pin("d:33:o")
L_EN4.write(1)
def robotStop(duration):
    print("Robot BERHENTI")
    R_PWM1.write(0)
    L_PWM1.write(0)
    R_PWM2.write(0)
    L_PWM2.write(0)
    R_PWM3.write(0)

```

```

L_PWM3.write(0)
R_PWM4.write(0)
L_PWM4.write(0)
board.pass_time(duration)
def robotMundur(speed, duration):
    print("Robot MUNDUR")
    R_PWM1.write(speed)
    L_PWM1.write(0)
    R_PWM2.write(0)
    L_PWM2.write(speed)
    R_PWM3.write(speed)
    L_PWM3.write(0)
    R_PWM4.write(0)
    L_PWM4.write(speed)
    board.pass_time(duration)
"===== HX711 ====="
referenceUnit = -204
def cleanAndExit():
    print("Cleaning...")
    GPIO.cleanup()
    print("Bye!")
    sys.exit()
hx = HX711(20, 21)
hx.set_reading_format("MSB", "MSB")
hx.set_reference_unit(referenceUnit)
hx.reset()
hx.tare()
print("Tare done! Add weight now...")
"===== ULTRASONIC ====="
TRIG = 19
ECHO = 13
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
class ultrasonic(Thread):
    def __init__(self, trig=TRIG, echo=ECHO):
        Thread.__init__(self)
        self.trig = trig
        self.echo = echo
        self.val = None
        self.stop_thread = False
    def run(self):
        while self.stop_thread == False:
            GPIO.output(TRIG, False)
            time.sleep(0.1)
            GPIO.output(TRIG, True)
            time.sleep(0.00001)

```

```

GPIO.output(TRIG, False)
while GPIO.input(ECHO)==0:
    pulse_start = time.time()
while GPIO.input(ECHO)==1:
    pulse_end = time.time()
    pulse_duration = pulse_end - pulse_start
    distance = (pulse_duration * 34300)/2
    distance = round(distance,2)
    # print("Distance: %f cm" %distance)
    self.val = distance

def get_dist(self):
    return self.val
def stop(self):
    self.stop_thread = True
    self.join()

"===== BUZZER ====="
buzzerPin = 16
GPIO.setup(buzzerPin, GPIO.OUT)
"===== EXIT BUTTON ====="
exitPin = 26
GPIO.setup(exitPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
"===== "

global frame_width, frame_height
frame_width = 640
frame_height = 480
imsize = 224
webcam_stream = WebcamStream(0)
webcam_stream.start()
model = YOLO("/home/pi/Program-Exel/models/best"+ str(imsize)
+"Nint8.onnx", task="detect")
box_annotator = sv.BoxAnnotator(thickness=1, text_thickness=1, text_scale=1)
horizontal_center = frame_width / 2
min_area = 95000
max_area = 110000
prev_error = 0.
prev_error_yaw = 0.
sum_error = 0.
sum_error_yaw = 0.
ultra = ultrasonic()
ultra.start()
while ultra.get_dist() == None:
    time.sleep(0.01)
while True :
    val = max(0, float(hx.get_weight(5)))
    print("%.2f %val, 'g')
    hx.power_down()

```



```

hx.power_up()
if val > 5000:
    GPIO.output(buzzerPin, GPIO.HIGH)
elif val > 4000:
    kp = 0.4
    kd = 0.3
    ki = 0.03
    Kp = 0.004
    Kd = 0.003
    Ki = 0.0003
    pwm_max = 0.32
    pwm_max_yaw = 0.45
elif val > 2000:
    kp = 0.3
    kd = 0.2
    ki = 0.02
    Kp = 0.003
    Kd = 0.002
    Ki = 0.0002
    pwm_max = 0.32
    pwm_max_yaw = 0.39
else:
    kp = 0.2
    kd = 0.1
    ki = 0.01
    Kp = 0.002
    Kd = 0.001
    Ki = 0.0001
    pwm_max = 0.25
    pwm_max_yaw = 0.35
    GPIO.output(buzzerPin, GPIO.LOW)
if webcam_stream.stopped is True :
    break
else :
    frame = webcam_stream.read()
    time.sleep(0.03) # delay value in seconds. so, delay=1 is equivalent to 1 second
    result = model(frame, imgsz=(imgsize), conf=0.5,iou=0.7, max_det=1,
agnostic_nms=False)[0]
    detections = sv.Detections.from_yolov8(result)
    frame = box_annotator.annotate(
        scene=frame, detections=detections
    )
if result.boxes is not None:
    box = result.boxes.xywh.cpu().numpy().astype(float)
    if box.size > 0:
        x_center = box[0][0]

```



```

y_center = box[0][1]

# yaw pid
error_yaw = horizontal_center - x_center
D_error_yaw = error_yaw - prev_error_yaw
yaw = (error_yaw * kp) + (D_error_yaw * kd) + (sum_error_yaw * ki)
pwmka_yaw = pwm_max_yaw - yaw
pwmki_yaw = pwm_max_yaw + yaw
if pwmka_yaw > pwm_max_yaw:
    pwmka_yaw = pwm_max_yaw
elif pwmka_yaw < 0:
    pwmka_yaw = 0
if pwmki_yaw > pwm_max_yaw:
    pwmki_yaw = pwm_max_yaw
elif pwmki_yaw < 0:
    pwmki_yaw = 0

# #fw-rv pids
error = min_area - detections.area
D_error = error - prev_error
accel = (error * Kp) + (D_error * Kd) + (sum_error * Ki)
pwmka = pwm_max + abs(accel)
pwmki = pwm_max + abs(accel)
if pwmka > pwm_max:
    pwmka = pwm_max
elif pwmka < 0:
    pwmka = 0
if pwmki > pwm_max:
    pwmki = pwm_max
elif pwmki < 0:
    pwmki = 0
speedka = pwmka + pwmka_yaw
speedki = pwmki + pwmki_yaw
if detections.area < min_area:
    print("Robot Manuver MAJU")
    R_PWM1.write(0)
    L_PWM1.write(speedka)
    R_PWM2.write(speedki)
    L_PWM2.write(0)
    R_PWM3.write(0)
    L_PWM3.write(speedka)
    R_PWM4.write(speedki)
    L_PWM4.write(0)
elif ultra.get_dist() < 18 or detections.area > max_area:
    robotMundur(0.5, 0.02)

```

```

print("PWM KIRI = %.2f" %pwmki)
print("PWM KIRI YAW = %.2f" %pwmki_yaw)
print("PWM KANAN = %.2f" %pwmka)
print("PWM KANAN YAW = %.2f" %pwmka_yaw)
prev_error = error
sum_error += error
prev_error_yaw = error_yaw
sum_error_yaw += error_yaw
else:
    robotStop(0.01)
    if ultra.get_dist() < 18:
        robotMundur(0.5, 0.02)
    cv2.imshow('Real-time Vest Detection', frame)
    if cv2.waitKey(5) == 27 or GPIO.input(exitPin) == GPIO.LOW:
        break
ultra.stop()
robotStop(0.01)
cleanAndExit()
cv2.destroyAllWindows()

```



Lampiran 5: Dokumentasi Kegiatan

1. Proses Pengerjaan Mekanika



Bending plat



Proses pengecatan

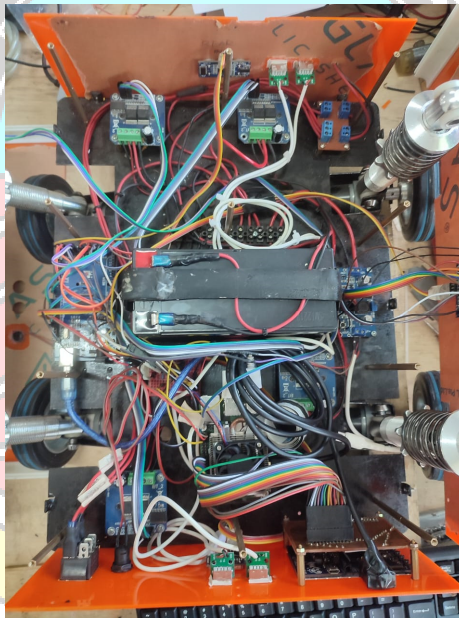


Proses pembubutan poros roda



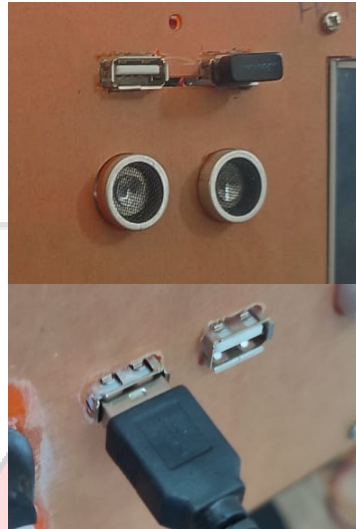
Proses perakitan rangka

## 2. Proses Pengerjaan Elektronika

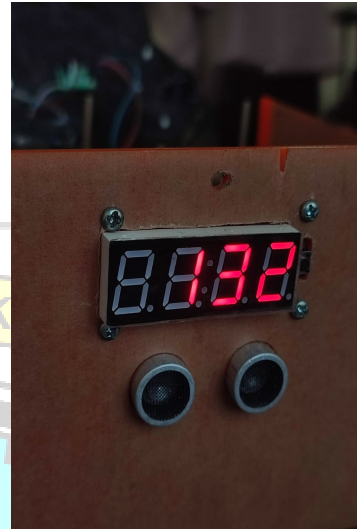


Proses instalasi perkabelan

Port USB



Indikator Aki

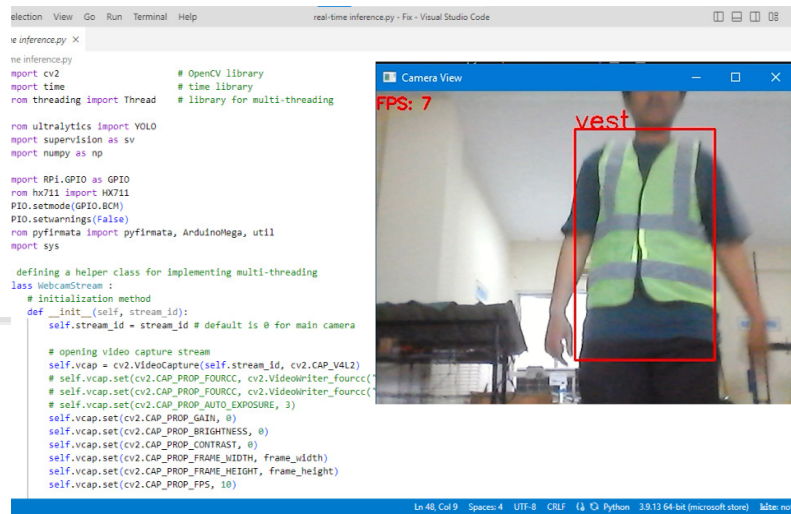


Proses pemasangan port USB dan indikator aki

### 3. Proses Pengerjaan Informatika

```
C:\WINDOWS\system32\cmd.exe
C:\Users\F. D. Exel Mainda>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117
Looking in indexes: https://download.pytorch.org/whl/cu117
Collecting torch
  Downloading https://download.pytorch.org/whl/cu117/torch-2.0.1%2Bcu117-cp39-cp39-win_amd64.whl (2343.7 MB)
----- 2.3/2.3 GB ? eta 0:00:00
Collecting torchvision
  Downloading https://download.pytorch.org/whl/cu117/torchvision-0.15.2%2Bcu117-cp39-cp39-win_amd64.whl (4.9 MB)
----- 4.9/4.9 MB 1.3 MB/s eta 0:00:00
Collecting torchaudio
  Downloading https://download.pytorch.org/whl/cu117/torchaudio-2.0.2%2Bcu117-cp39-cp39-win_amd64.whl (2.5 MB)
----- 2.5/2.5 MB 1.4 MB/s eta 0:00:00
Collecting filelock (from torch)
  Downloading https://download.pytorch.org/whl/filelock-3.9.0-py3-none-any.whl (9.7 kB)
Requirement already satisfied: typing-extensions in c:\users\f. d. exel mainda\appdata\local\packages\pythonsoftwarefound
ation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from torch) (3.7.4.3)
Collecting sympy (from torch)
  Downloading https://download.pytorch.org/whl/sympy-1.11.1-py3-none-any.whl (6.5 MB)
----- 6.5/6.5 MB 1.4 MB/s eta 0:00:00
Collecting networkx (from torch)
  Downloading https://download.pytorch.org/whl/networkx-3.0-py3-none-any.whl (2.0 MB)
----- 2.0/2.0 MB 1.4 MB/s eta 0:00:00
Requirement already satisfied: Jinja2 in c:\users\f. d. exel mainda\appdata\local\packages\pythonsoftwarefoundation.pyth
on.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from torch) (3.0.1)
Requirement already satisfied: numpy in c:\users\f. d. exel mainda\appdata\local\packages\pythonsoftwarefoundation.pyth
on.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from torchvision) (1.19.5)
Requirement already satisfied: requests in c:\users\f. d. exel mainda\appdata\local\packages\pythonsoftwarefoundation.py
thon.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from torchvision) (2.26.0)
Collecting pillow<8.3.*>5.3.0 (from torchvision)
  Downloading https://download.pytorch.org/whl/Pillow-9.3.0-cp39-cp39-win_amd64.whl (2.5 MB)
----- 2.5/2.5 MB 1.6 MB/s eta 0:00:00
```

Proses instalasi library



### Proses uji coba program

#### 4. Proses pengambilan data

Dokumentasi proses pengambilan data dapat diakses melalui tautan di bawah:





## Lampiran 6: Biodata Penulis



Franklin Delano Exel Mainda. lahir di Kota Makassar pada tanggal 27 Desember 1999. Putra tunggal dari ayah Yohannes Mainda, S.ST. dan ibu Alm. Merly. Pada tahun 2012, Penulis menyelesaikan pendidikan Sekolah Dasar di SD Inpres Baraya I. Pada tahun yang sama, penulis melanjutkan pendidikan di SMP Frater Makassar dan lulus pada tahun 2015. Setelah itu, Penulis melanjutkan pendidikan Sekolah Menengah Kejuruan di SMK Negeri 5 Makassar dengan jurusan Teknik Elektronika Industri dan lulus pada tahun 2019. Pada tahun 2019, Penulis diterima di Perguruan Tinggi Negeri Politeknik Negeri Ujung Pandang sebagai Mahasiswa Jurusan Mesin Program Studi S1 Terapan Teknik Mekatronika. Penulis pernah melaksanakan kegiatan Praktik Kerja Lapangan di Balai Teknik Perkeretaapian Kelas I Bandung, Kota Bandung, Provinsi Jawa Barat.



Lampiran 7: Lembar Revisi

Lembar 1

LAMPIRAN BERITA ACARA PELAKSANAAN  
UJIAN SIDANG SKRIPSI

Nama Mahasiswa : Franklin Delano Exel Mainda

No. Induk Mahasiswa : 444 19 006

Catatan/Daftar Revisi Penguji:

No.	Nama Dosen	Uraian	Tanda Tangan
1.	Prof. A.M. Siddiq Yunus ST. MENG Ph.D.	<ul style="list-style-type: none"> <li>* Penulisan konsisten →</li> <li>* Perbaiki flowchart</li> <li>* Tambahkan saran 2 yg dibedakan</li> </ul>	
2.	Imam Habrianyah S.T, MT.	<ul style="list-style-type: none"> <li>* Saran perbaikan v/ flowchart</li> <li><del>* perbaiki bagan hal 54.</del></li> <li>* Tambahkan saran                             <ul style="list-style-type: none"> <li>- metode pemrosesan</li> <li>- pengembangan dgn metode lain dan dibandingkan.</li> <li>* - dicoba dgn objek lain.</li> </ul> </li> </ul>	
3.	Paisal ST-MT.	<ul style="list-style-type: none"> <li>* perbaiki flowchart</li> <li>* bagaimana memilih motor DC.</li> <li>* perbaiki perhitungan jarak.</li> </ul>	
4.	Er. Lwi, MT.	<ul style="list-style-type: none"> <li>* Saran                             <ul style="list-style-type: none"> <li>- perancangan yg dibuat</li> </ul> </li> <li>* Rumus perhitungan daya (P) perlu ada representasi</li> <li>* Buatkan alasan pemilihan part (motor DC).</li> <li>* Tambahkan saran tuliskan kekurangan &amp; fokus kedepan.</li> </ul>	

FS-