



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - EE 184801**

## **IMPLEMENTASI LIDAR PADA ROBOT SEPAKBOLA BERODA UNTUK TENDANGAN KE GAWANG**

Maulana Falithurrahman  
NRP 0711154000087

Dosen Pembimbing  
Dr. Muhammad Rivai, ST., MT.  
Fajar Budiman, ST., M.Sc.

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





**ITS**

Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - EE 184801**

**IMPLEMENTASI LIDAR PADA ROBOT SEPAKBOLA  
BERODA UNTUK TENDANGAN KE GAWANG**

Maulana Falithurrahman  
NRP 0711154000087

Dosen Pembimbing  
Dr. Muhammad Rivai, ST., MT.  
Fajar Budiman, ST., M.Sc.

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





**ITS**

Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - EE 184801**

**LIDAR IMPLEMENTATION FOR KICKING TOWARDS  
GOAL FOR WHEELED SOCCER ROBOT**

Maulana Falithurrahman  
NRP 0711154000087

Supervisor  
Dr. Muhammad Rivai, ST., MT.  
Fajar Budiman, ST., M.Sc.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Electrical Engineering  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019



## PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Implementasi LiDAR pada Robot Sepakbola Beroda untuk Tendangan ke Gawang**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 3 Juli 2019



Maulana Falithurrahman  
NRP 0711154000087

*Halaman ini sengaja dikosongkan*



**IMPLEMENTASI LIDAR PADA ROBOT  
SEPAKBOLA BERODA UNTUK TENDANGAN KE  
GAWANG**

**TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

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

**Menyetujui :**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

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

**Fajar Budiman, ST., M.Sc.**  
**NIP. 198607072014041001**



*Halaman ini sengaja dikosongkan*





# **Implementasi LiDAR pada Robot Sepakbola Beroda untuk Tendangan ke Gawang**

Nama : Maulana Falithurrahman  
Pembimbing : 1. Dr. Muhammad Rivai, ST., MT.  
2. Fajar Budiman, ST., M.Sc.

## **ABSTRAK**

Robot sepakbola beroda merupakan salah satu kompetisi pada bidang robotik yang sudah dilombakan baik pada tingkat nasional maupun tingkat internasional. Pada pertandingan robot sepakbola, kemampuan untuk mencetak gol ke gawang lawan merupakan kemampuan yang penting. Hal ini karena dengan mencetak gol dapat membawa untuk memenangkan pertandingan. Umumnya banyak tim yang menggunakan sudut yang didapatkan melalui kompas digital untuk melakukan tendangan ke gawang. Pada tugas akhir ini akan dirancang metode untuk mendeteksi gawang menggunakan sensor kamera dan menentukan jarak ke gawang dengan LiDAR. Jaringan syaraf tiruan sebelumnya telah dilatih untuk dapat mendeteksi objek berupa gawang. Kemudian citra digital yang didapatkan dari kamera yang dipasang pada robot digunakan sebagai masukan jaringan syaraf tiruan. Heading robot akan bergerak untuk mencari objek gawang pada lapangan. Apabila gawang telah ditemukan, heading robot akan diarahkan ke bagian gawang dan sensor LiDAR akan menembakkan pulsa laser untuk mengukur jarak. Tendangan yang dihasilkan dari robot akan dimodelkan dengan lintasan gerak parabola dengan kekuatan yang berbeda tergantung jarak yang diukur oleh sensor LiDAR. Dari hasil pengujian sistem, didapatkan tingkat kesuksesan untuk mencetak gol dengan persentase rata-rata dari 89% tanpa penjaga gawang dan 77% dengan penjaga gawang. Tingkat persentase keberhasilan paling tinggi didapatkan saat pengujian dengan jarak sejauh 4meter dari gawang. Diharapkan dengan adanya metode ini dapat meningkatkan peluang untuk mencetak gol karena dapat membantu tendangan agar dapat mengarah ke gawang. Menambahkan dataset pada pelatihan dan membuat mekanisme penendang yang baik dapat dilakukan untuk meningkatkan tingkat kesuksesan mencetak gol ke gawang lawan.

**Kata Kunci:** LiDAR, robot sepakbola beroda, tendangan ke gawang

*Halaman ini sengaja dikosongkan*

# **LiDAR Implementation for Kicking Towards Goal for Wheeled Soccer Robot**

*Student's Name* : Maulana Falithurrahman  
*Supervisor* : 1. Dr. Muhammad Rivai, ST., MT.  
2. Fajar Budiman, ST., M.Sc.

## **ABSTRACT**

*Soccer wheeled robot is one of the robotic competitions, which already held not only at national scale but also international. In a soccer wheeled robot match, the ability to score a goal at your opponent is a vital ability. This ability becomes essential because by scoring a goal it can carry your team to win the match. Mostly, teams in soccer wheeled robot use digital compass to attempt shoot on goal. In this final project, a method to detect goal by using camera and define the distance using LiDAR will be made. Artificial neural network has been trained previously to detect a goal. Then digital image captured by the camera which is mounted at the robot will be used as the input to the neural network. Heading of the robot will move to search the goal on the field. If the robot detects the goal, robot's heading will be headed to the part of the goal and LiDAR will shoot laser pulse to measure the distance. Projectile motion will be used as the model for the kick which has different speed, it depends on the distance measured by LiDAR. From the system test, we got success rate to score a goal on the 20 test points at the field with 89% percentage without the presence of goalkeeper and 77% percentage with the presence of goalkeeper. Highest success rate is gained when we test the kick 4m from the goal. With this method, we expect that the opportunity to score a goal will be higher because it helps to head the kicking direction of the robot towards goal. This method is not flawless and still can be improved by adding dataset to the artificial neural network and upgrading the kicking mechanism to boost success rate.*

**Keywords:** *LiDAR, soccer wheeled robot, kicking towards goal*

*Halaman ini sengaja dikosongkan*



## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan berkah, rahmat, serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul: Implementasi LiDAR pada Robot Sepakbola Beroda untuk Tendangan ke Gawang

Penelitian ini diusulkan sebagai lanjutan dari penelitian tentang robot sepak bola beroda. Selain itu, penelitian ini dilakukan sebagai prasyarat penulis untuk mendapatkan gelar Sarjana Teknik dari Jurusan Teknik Elektro ITS, bidang studi Elektronika. Penulis ingin menyampaikan terima kasih kepada:

1. Bapak Dr. Muhammad Rivai, ST., MT. dan Fajar Budiman, ST., M.Sc. selaku dosen pembimbing yang selalu memberikan saran serta bantuan dalam penelitian ini.
2. Bapak Dr. Ir. Totok Mujiono, M.Kom., Bapak Astria Nur Irfansyah, ST., M.Eng., Ph.D., Bapak Ir. Harris Pirngadi, MT., dan Bapak Ir. Tasripan, MT. sebagai dosen penguji yang telah memberikan saran dan masukan dalam penulisan tugas akhir ini.
3. Bapak Ardyono Priyadi, S.T., M.Eng., Dr.Eng. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
4. Keluarga penulis, khususnya bapak dan ibu yang senantiasa memberikan dukungan baik dukungan material maupun non-material.
5. Bapak ibu dosen pengajar Departemen Teknik Elektro, khususnya bidang studi Elektronika.
6. Seluruh rekan beserta dosen pembimbing Tim Robotika ITS.
7. Seluruh anggota tim IRIS yang telah membantu dalam pengerjaan robot.

Kesempurnaan hanya milik Allah SWT, penyusunan tugas akhir ini tentu masih banyak kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun.

Surabaya, 3 Juli 2019

Maulana Falithurrahman

*Halaman ini sengaja dikosongkan*

## Daftar Isi

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR .....	v
Daftar Isi .....	vii
Daftar Gambar .....	xi
Daftar Tabel .....	xv
BAB I.....	1
PENDAHULUAN .....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Batasan Masalah .....	3
1.5. Metodologi Penelitian.....	3
1.6. Sistematika Penulisan .....	4
1.7. Relevansi .....	4
BAB II.....	7
TINJAUAN PUSTAKA .....	7
2.1. Gerak Parabola .....	7
2.2. LiDAR .....	9
2.3. Citra Digital .....	13
2.3.1. Citra RGB .....	13
2.3.2. OpenCV .....	14
2.4. Jaringan Saraf Tiruan.....	15
2.4.1 Convolutional Neural Network (CNN) .....	16

2.4.2	Single Shot Detection .....	19
2.5.	Robot Operating System .....	22
2.5.1.	ROS Node.....	23
2.5.2.	ROS Topic .....	23
2.5.3.	ROS Message.....	24
2.5.4.	ROS Publisher.....	24
2.5.5.	ROS Subscriber.....	24
2.6.	Kamera .....	24
2.7.	Komputer mini Intel NUC.....	27
2.8.	Microcontroller STM32F4 Discovery .....	28
2.9.	Driver Motor .....	31
2.10.	Motor DC .....	32
2.11.	Motor Brushless DC.....	35
BAB III.....		43
PERANCANGAN SISTEM.....		43
3.1.	Perancangan Mekanik dan Elektronik.....	44
3.1.1.	Perancangan Mekanik .....	44
3.1.2.	Perancangan Elektronik .....	47
3.2.	Pendeteksian Objek Gawang.....	49
3.2.1.	Pelatihan Jaringan Syaraf Tiruan .....	49
3.2.2.	Implementasi Jaringan Syaraf Tiruan .....	53
3.3.	Pengukuran Jarak dengan LiDAR .....	55
3.4.1.	Pembacaan data sensor LiDAR.....	55
3.4.2.	Pengiriman data ke komputer mini .....	57
3.4.	Implementasi Tendangan ke Gawang .....	58
3.4.1.	Memodelkan gerak parabola sebagai lintasan tendangan	58
3.4.2.	Menentukan kekuatan tendangan yang digunakan.....	59

3.4.3. Merangkai keseluruhan sistem .....	60
BAB IV .....	65
PENGUJIAN DAN ANALISA.....	65
4.1. Pengujian Deteksi Gawang.....	65
4.2. Pengujian Pembacaan Sensor LiDAR .....	68
4.3. Pengujian Kekuatan Tendangan .....	71
4.4. Pengujian Tendangan ke Gawang .....	76
BAB 5 .....	81
PENUTUP.....	81
5.1. Kesimpulan.....	81
5.2. Saran .....	81
Daftar pustaka .....	83
Lampiran A .....	87
Lampiran B .....	97
Biodata Penulis .....	105

*Halaman ini sengaja dikosongkan*

## Daftar Gambar

<b>Gambar 2. 1</b>	Gerak parabola dan bentuk lintasannya .....	7
<b>Gambar 2. 2</b>	Cara Kerja LiDAR [7] .....	10
<b>Gambar 2. 3</b>	Gambar LiDAR yang diambil saat malam hari di hutan tropis .....	10
<b>Gambar 2. 4</b>	Waktu tempuh pulsa laser .....	11
<b>Gambar 2. 5</b>	Pola pindaian pulsa laser pada LiDAR satelit .....	12
<b>Gambar 2. 6</b>	Citra digital direpresentasikan dengan nilai 8bit .....	13
<b>Gambar 2. 7</b>	Colorspace RGB digambarkan secara 3 dimensi .....	14
<b>Gambar 2. 8</b>	Pengenalan objek menggunakan modul DNN pada OpenCV .....	15
<b>Gambar 2. 9</b>	Perbandingan antara neuron biologis dan artifisial [11].	15
<b>Gambar 2. 10</b>	Arsitektur CNN secara umum .....	16
<b>Gambar 2. 11</b>	Layer konvolusi pada CNN .....	17
<b>Gambar 2. 12</b>	Pooling layer pada CNN .....	18
<b>Gambar 2. 13</b>	Contoh multi layer perceptron sederhana .....	18
<b>Gambar 2. 14</b>	Pendekatan single shot detection [13] .....	19
<b>Gambar 2. 15</b>	Arsitektur Single Shot Multibox Detector .....	20
<b>Gambar 2. 16</b>	IoU dari ground truth dan predicted bounding box .....	21
<b>Gambar 2. 17</b>	Perbedaan layer konvolusi MobileNet dan CNN pada umumnya .....	22
<b>Gambar 2. 18</b>	Hubungan publisher, subscriber dan message .....	23
<b>Gambar 2. 19</b>	Bagian-bagian dasar penyusun kamera .....	25
<b>Gambar 2. 20</b>	Gambar yang diambil dengan aperture kamera yang berbeda [22] .....	26
<b>Gambar 2. 21</b>	Gambar yang diambil dengan shutter speed berbeda .....	26
<b>Gambar 2. 22</b>	Bagian sistem komputer secara umum .....	27
<b>Gambar 2. 23</b>	Blok diagram konektivitas STM32F407 .....	30
<b>Gambar 2. 24</b>	Konfigurasi rangkaian H-bridge sederhana [30] .....	31
<b>Gambar 2. 25</b>	Arah putaran motor ketika diberikan input .....	32
<b>Gambar 2. 26</b>	Gaya tolak saat elektromagnet dialiri arus .....	32
<b>Gambar 2. 27</b>	Sikat dan cincin belah pada motor DC .....	33
<b>Gambar 2. 28</b>	Rangkaian ekuivalen motor DC [29] .....	33
<b>Gambar 2. 29</b>	Bagian utama pada BLDC .....	35
<b>Gambar 2. 30</b>	Medan magnet pada BLDC .....	35
<b>Gambar 2. 31</b>	Bentuk gelombang pada fasa BLDC .....	36
<b>Gambar 2. 32</b>	Contoh aliran arus pada fasa motor brushless .....	37

<b>Gambar 2. 33</b>	Keluaran pembacaan sensor hall-effect .....	37
<b>Gambar 2. 34</b>	Keluaran pembacaan EMF .....	38
<b>Gambar 2. 35</b>	Jenis-jenis BLDC motor .....	38
<b>Gambar 2. 36</b>	Pulsa PWM yang dihasilkan melalui komparator .....	39
<b>Gambar 2. 37</b>	Waktu on dan off pada pulsa PWM.....	40
<b>Gambar 3. 1</b>	Penempatan komponen yang digunakan pada robot.....	45
<b>Gambar 3. 2</b>	Penempatan motor penggerak pada robot.....	46
<b>Gambar 3. 3</b>	Desain penempatan kamera dan LiDAR .....	46
<b>Gambar 3. 4</b>	Desain penempatan komponen elektronik yang digunakan .....	47
<b>Gambar 3. 5</b>	Diagram blok hubungan antar komponen yang digunakan .....	48
<b>Gambar 3. 6</b>	Diagram alir pelatihan jaringan syaraf tiruan .....	50
<b>Gambar 3. 7</b>	Jalur pengambilan gambar gawang untuk dijadikan dataset .....	51
<b>Gambar 3. 8</b>	Gambar gawang sebelum dan sesudah diberi label .....	52
<b>Gambar 3. 9</b>	Keluaran file xml yang dihasilkan.....	52
<b>Gambar 3. 10</b>	Diagram alir program deteksi gawang.....	54
<b>Gambar 3. 11</b>	Diagram blok pengiriman posisi gawang ke program utama .....	55
<b>Gambar 3. 12</b>	Diagram alir pengukuran jarak dengan LiDAR.....	56
<b>Gambar 3. 13</b>	Data serial yang dikirimkan oleh LiDAR.....	56
<b>Gambar 3. 14</b>	Pinout diagram sensor LiDAR .....	57
<b>Gambar 3. 15</b>	Pinout diagram modul FTDI232.....	57
<b>Gambar 3. 16</b>	Diagram blok pengiriman sensor LiDAR ke program utama .....	58
<b>Gambar 3. 17</b>	Diagram alir penentuan kekuatan tendangan.....	59
<b>Gambar 3. 18</b>	Diagram blok rangkaian program keseluruhan sistem..	61
<b>Gambar 3. 19</b>	Diagram alir pengujian keseluruhan sistem.....	63
<b>Gambar 4. 1</b>	Penempatan kamera untuk pengujian deteksi gawang.....	65
<b>Gambar 4. 2</b>	Titik pengujian deteksi gawang pada lapangan .....	68
<b>Gambar 4. 3</b>	Penempatan sensor LiDAR pada robot .....	69
<b>Gambar 4. 4</b>	Pengukuran jarak pada robot menggunakan LiDAR .....	69
<b>Gambar 4. 5</b>	Grafik perbandingan jarak hasil pembacaan LiDAR dan sebenarnya.....	71
<b>Gambar 4. 6</b>	Posisi penempatan mekanisme penendang .....	72
<b>Gambar 4. 7</b>	Grafik hubungan jarak tempuh proyektil maksimal dengan PWM.....	74



<b>Gambar 4. 8</b>	Grafik hubungan tinggi proyektil maksimal dengan PWM .....	75
<b>Gambar 4. 9</b>	Grafik hubungan kecepatan linear dengan PWM .....	75
<b>Gambar 4. 10</b>	Penempatan sensor kamera dan LiDAR pada robot .....	76

*Halaman ini sengaja dikosongkan*

## Daftar Tabel

<b>Tabel 4. 1</b> Hasil pengujian akurasi model jaringan syaraf tiruan .....	66
<b>Tabel 4. 2</b> Hasil pengujian degree of confidence pada titik-titik.....	67
<b>Tabel 4. 3</b> Hasil pengukuran sensor LiDAR ke gawang.....	70
<b>Tabel 4. 4</b> Tabel pengujian kekuatan tendangan .....	74
<b>Tabel 4. 5</b> Hasil Pengujian tanpa Penjaga Gawang .....	78
<b>Tabel 4. 6</b> Hasil Pengujian dengan Menggunakan Penjaga Gawang ....	80

*Halaman ini sengaja dikosongkan*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Robot merupakan alat yang diciptakan dengan tujuan agar dapat membantu kebutuhan manusia baik secara otomatis maupun dengan kendali oleh manusia dengan menggunakan program yang telah ditanamkan pada robot. Untuk menumbuhkan iklim kompetitif yang positif pada lingkungan perguruan tinggi di Indonesia, setiap tahunnya diselenggarakan kompetisi robot dengan kategori yang beragam. Salah satu kategori kompetisi yang diadakan oleh Kementerian Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia adalah Kontes Robot Sepakbola Indonesia Beroda.

Robot sepakbola beroda merupakan kompetisi pada bidang robotik yang sudah memiliki kompetisi tingkat internasional dimana kompetisi ini menggabungkan kecerdasan buatan pada robot dengan sepakbola. Sebagai salah satu pendiri dari robot sepakbola di dunia, penyelenggaraan *Robocup Middle Size League (MSL)* telah dimulai sejak tahun 1997. Permainan dilakukan pada lapangan dalam ruangan yang ukurannya dibuat lebih kecil dengan tim yang terdiri dari lima buah robot autonomous melawan tim lain [1]. Secara singkat, bidang ini mempelajari tentang bagaimana robot dapat dibuat dan dilatih untuk memainkan pertandingan sepakbola. Pada pertandingan robot sepakbola, terdapat beberapa kemampuan dasar yang penting yang harus dimiliki oleh robot. Kemampuan dasar tersebut melingkupi kemampuan untuk mendeteksi keberadaan bola menggunakan kamera, menggiring bola menggunakan mekanisme penggiring dan menendang bola menggunakan mekanisme penendang yang sudah terdapat pada robot.

Kontes robot sepakbola Indonesia beroda diselenggarakan berdasarkan aturan yang berlaku pada *Robocup Middle Size League (MSL)* dengan menyesuaikan kondisi di Indonesia. Untuk memenangkan pertandingan sepakbola, baik pada pemain robot sepakbola maupun pada pemain manusia, salah satu tim harus mencetak gol lebih banyak dari tim lawan. Kemampuan mencetak gol menjadi salah satu kemampuan penting yang harus dimiliki oleh robot. Saat ini sudah ada penelitian mengenai cara untuk mendeteksi fitur yang digunakan pada pertandingan seperti keberadaan gawang tim lawan meningkatkan keberhasilan mencetak gol [2]. Sebagian tim juga menggunakan *odometry* dan kompas digital untuk

mengetahui posisi robot dan sudut terhadap gawang untuk menghasilkan tendangan. Tetapi pembacaan *odometry* dan kompas digital akan mengalami pergeseran nilai setelah dimainkan dan pergeseran nilai tersebut tidak bisa diprediksi kapan akan terjadi. Pergeseran nilai dari *odometry* dan kompas digital akan berpengaruh ke tendangan yang dihasilkan, karena orientasi robot terhadap gawang sudah berbeda. Untuk mengatasi pergeseran nilai *odometry* dan kompas digital diperlukan kalibrasi ulang untuk me-*reset* hasil pembacaan sensor tersebut. Namun untuk melakukan kalibrasi diperlukan waktu yang tidak sebentar.

Dengan menggunakan sensor kamera dan LiDAR, pada tugas akhir ini dirancang metode baru untuk mendeteksi keberadaan gawang dan melakukan tendangan ke gawang. Sensor kamera akan digunakan untuk mendapatkan citra digital yang kemudian diolah untuk dijadikan input dari model jaringan syaraf tiruan yang telah dilatih sebelumnya untuk mendeteksi keberadaan gawang. Setelah mengetahui keberadaan gawang, heading robot akan diarahkan ke bagian gawang dan selanjutnya sensor LiDAR akan ditembakkan ke arah gawang untuk mendapatkan jarak dari robot ke gawang. Tendangan ke arah gawang dimodelkan menggunakan gerak parabola karena dengan gerak parabola akan lebih sulit dijangkau oleh robot penjaga gawang tim lawan. Dengan memodelkan tendangan ke arah gawang sebagai gerak parabola dengan sudut elevasi awal yang dibuat konstan, maka kecepatan tendangan yang dihasilkan bisa divariasikan tergantung jarak yang didapatkan oleh LiDAR.

## 1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah sebagai berikut:

1. Melakukan deteksi terhadap objek gawang
2. Menentukan jarak dari robot ke gawang
3. Mencari posisi gawang dan melakukan tendangan dari robot ke gawang

## 1.3. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Mendeteksi objek gawang menggunakan kamera
2. Menggunakan lidar untuk menentukan jarak ke gawang
3. Mengarahkan *heading* robot ke gawang dan memodelkan tendangan ke gawang menggunakan gerak parabola

## **1.4. Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Tidak ada halangan lain dari robot penyerang ke gawang.
2. Lapangan berada di dalam ruangan tanpa ada interferensi cahaya dari luar.
3. Ukuran gawang dan jarak maksimal tendangan disesuaikan dengan peraturan kontes robot sepakbola Indonesia divisi beroda [3], [4].
4. Fungsi- fungsi dasar yang digunakan pada robot sudah tersedia.

## **1.5. Metodologi Penelitian**

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

- **Studi Literatur**

Pada tahap ini, dilakukan pengumpulan terhadap dasar teori yang mendukung dalam penulisan tugas akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, paper dan artikel ilmiah serta konsultasi dengan dosen pembimbing.
- **Perancangan Perangkat Keras**

Pada tahap ini dilakukan perancangan posisi penempatan sensor LiDAR dan kamera pada robot sepakbola beroda.
- **Perancangan Perangkat Lunak**

Perancangan perangkat lunak terdiri dari perancangan sistem mikrokontroler dan mini PC. Mikrokontroler berfungsi sebagai perangkat slave yang akan menerima perintah dari komputer mini untuk mengendalikan motor penggerak dan sensor-sensor yang terdapat pada robot. Pengolahan data kamera dan LiDAR dilakukan pada mini PC. Citra digital dari kamera akan dijadikan masukkan terhadap model jaringan syaraf tiruan yang sudah dilatih untuk mendeteksi keberadaan gawang. Sensor LiDAR dihubungkan dengan komputer mini menggunakan komunikasi serial.
- **Pengujian Sistem**

Pengujian sistem dilakukan untuk menilai keandalan dari sistem yang telah diselesaikan. Pengujian bertujuan untuk melihat apakah perangkat keras dan lunak telah berjalan dengan baik.

- Analisa  
Data-data yang diperoleh pada pengujian akan dianalisa lebih lanjut agar didapatkan kelebihan dan kekurangan dari sistem.
- Penyusunan laporan tugas akhir  
Tahap ini merupakan tahap terakhir dari proses pengerjaan laporan tugas akhir. Laporan tugas akhir berisi seluruh hal yang berkaitan terhadap tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka, terori penunjang, perancangan sistem, pengujian dan penutup.

## **1.6. Sistematika Penulisan**

Dalam buku tugas akhir ini, pembahasan akan dibuat terbagi menjadi lima bab yang kemudian akan disusun dengan menggunakan sistematika penulisan sebagai berikut:

- Bab I: Pendahuluan  
Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi
- Bab II: Tinjauan pustaka  
Pada bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya yang berhubungan robot yang akan dikerjakan.
- Bab III: Perancangan Sistem  
Bab ini berisi tentang perencanaan sistem yang meliputi perangkat keras dan perangkat lunak untuk pembuatan robot ini.
- BAB IV: Pengujian dan Analisis  
Bab ini menjelaskan tentang pengujian alat pada robot sepakbola beroda dan analisa hasil pengujian
- BAB V: Penutup  
Bab ini berisi kesimpulan yang didapatkan dari pembuatan alat dan saran untuk pengembangan selanjutnya.

## **1.7. Relevansi**

Hasil yang diharapkan dari tugas akhir ini adalah sebuah sistem robot sepakbola beroda yang dapat mendeteksi gawang sehingga dapat menentukan arah tendangan ke bagian gawang. Selain mendeteksi



gawang, robot juga dapat melakukan deteksi terhadap keberadaan penjaga gawang dan menendang ke bagian yang tidak dijaga. Tendangan robot ini dimodelkan dengan gerak parabola. Kekuatan tendangan ke gawang ini akan memiliki nilai yang bergantung pada jarak robot terhadap gawang. Sehingga tendangan yang dihasilkan bervariasi terhadap jarak agar menghasilkan tendangan yang efektif. Tugas akhir ini diharapkan dapat berguna bagi pengembangan penelitian robot sepakbola beroda.

*Halaman ini sengaja dikosongkan*

## BAB II

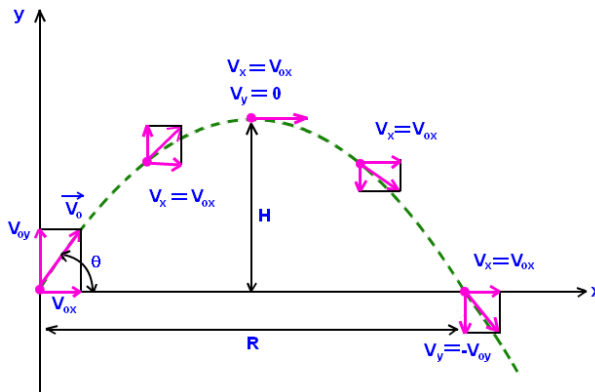
### TINJAUAN PUSTAKA

Tinjauan pustaka dalam bab ini akan menjelaskan tentang sistem-sistem yang berkaitan dengan tugas akhir ini dan sudah diterapkan pada penelitian-penelitian oleh penulis sebelumnya. Bagian teori penunjang akan menjelaskan tentang dasar teori yang berhubungan dengan keseluruhan sistem yang dirancang pada tugas akhir ini.

#### 2.1. Gerak Parabola

Gerak parabola merupakan gerak yang membentuk sudut terhadap bidang horizontal. Pada gerak parabola, gaya gesek udara diabaikan sehingga gaya yang bekerja hanya gaya berat yang dihasilkan oleh percepatan gravitasi. Gerak ini merupakan gerak yang spesial pada gerak dua dimensi. Sebuah partikel bergerak pada bidang datar dengan kecepatan awal tetapi selalu mengalami percepatan jatuh bebas yang mengarah ke bawah [5].

Gerak parabola dipelajari pada bahasan kinematika, ilmu yang membahas tentang gerak benda tanpa memperhatikan penyebabnya, sehingga gerakan benda hanya ditinjau setelah diberi kecepatan awal dan benda bergerak dalam lintasan melengkung dimana hanya ada pengaruh gravitasi. Hal ini menyebabkan pengaruh lain seperti gesekan benda dengan udara dapat diabaikan.



Gambar 2. 1 Gerak parabola dan bentuk lintasannya

Persamaan-persamaan kecepatan awal gerak parabola dapat diuraikan menjadi komponen horizontal dan komponen vertikal dengan mengalikan kecepatan awal dengan cosinus sudut elevasi untuk komponen horizontal dan sinus sudut elevasi untuk komponen vertikal. Secara matematis persamaan tersebut dapat dilihat pada (2.1) dan (2.2) di bawah. Sedangkan persamaan-persamaan jarak pada gerak parabola yang diuraikan menjadi komponen horizontal dan vertikal dapat dilihat pada persamaan (2.3) dan (2.4). Lalu persamaan kecepatan gerak parabola berupa gerak lurus berubah beraturan pada komponen vertikal dapat dilihat pada persamaan (2.5), persamaan kecepatan gerak parabola pada sumbu horizontal yang merupakan gerak lurus beraturan akan ekuivalen dengan persamaan (2.1).

$$V_{ox} = V_o \cos \theta \quad (2.1)$$

$$V_{oy} = V_o \sin \theta \quad (2.2)$$

$$x = V_o t \cos \theta \quad (2.3)$$

$$y = V_o t \sin \theta - \frac{1}{2} g t^2 \quad (2.4)$$

$$V_y = V_o \sin \theta - g t \quad (2.5)$$

Pada gerak parabola, ada dua komponen waktu yaitu waktu yang diperlukan untuk ke titik tertinggi dimana posisi y maksimum, maka vektor kecepatan sumbu y akan bernilai 0 dan menyisakan vektor kecepatan horizontal. Komponen waktu yang kedua adalah waktu yang diperlukan untuk mencapai jarak maksimum dimana waktu ini bernilai dua kali waktu yang diperlukan untuk mencapai titik tertinggi. Secara matematis waktu untuk mencapai titik tertinggi ( $t_p$ ) dan titik terjauh ( $t_{maks}$ ) dirumuskan pada persamaan (2.6) dan (2.7).

$$t_p = \frac{V_o \sin \theta}{g} \quad (2.6)$$

$$t_{maks} = 2t_p \quad (2.7)$$

Jarak pada sumbu-x saat proyektil akan mencapai titik dimana menempuh jarak terjauh didapatkan dengan mengalikan vector kecepatan pada sumbu x dengan waktu yang dibutuhkan proyektil untuk mencapai titik terjauh ( $t_{maks}$ ). Oleh karena itu persamaan (2.7) akan disubstitusikan

ke dalam persamaan (2.3). Secara matematis persamaan untuk mencari jarak tersebut dituliskan dalam persamaan (2.8)

$$x_{maks} = (V_o \cos \theta) \left( 2 \frac{V_o \sin \theta}{g} \right)$$
$$x_{maks} = \frac{V_o^2 \sin 2\theta}{g} \quad (2.8)$$

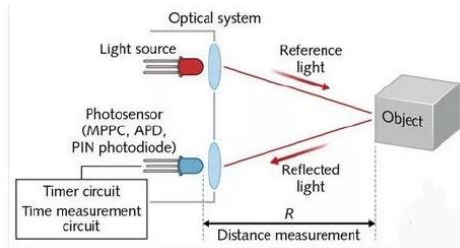
Titik tertinggi proyektil atau jarak maksimum yang dapat ditempuh proyektil pada sumbu  $y$  didapatkan saat proyektil sudah tidak memiliki vektor kecepatan pada sumbu  $y$  karena mengalami perlambatan oleh gravitasi. Nilai titik tertinggi ini akan didapatkan dengan substitusi persamaan (2.6) ke dalam persamaan (2.4). Secara matematis titik tertinggi proyektil dapat dirumuskan oleh persamaan (2.9)

$$y_p = 0 + \frac{1}{2} g \left( \frac{V_o \sin \theta}{g} \right)^2$$
$$y_p = \frac{V_o^2 \sin^2 \theta}{2g} \quad (2.9)$$

Pada gerak parabola, udara yang dilewati oleh proyektil diasumsikan tidak memberikan pengaruh terhadap lintasan gerakanya. Tetapi pada banyak situasi perbedaan antara perhitungan dan kenyataan bisa besar karena ada gesekan udara yang berlawanan dengan gerakan [5].

## 2.2. LiDAR

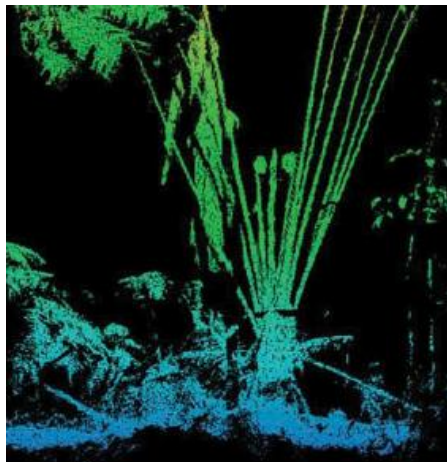
LiDAR, yang merupakan singkatan dari *light detection and ranging*, adalah teknologi yang mengukur jarak berdasarkan dari waktu pengiriman dan penerimaan sinyal laser. Jenis gelombang laser diskrit atau kontinyu dapat digunakan pada LiDAR. Gelombang laser diskrit memancarkan energi dalam waktu yang singkat dan mendeteksi jarak berdasarkan dari amplitudo sinyal yang diterima. Sebaliknya, gelombang laser kontinyu akan mendeteksi jarak berdasarkan perbedaan fase antara sinyal yang dikirimkan dan diterima [6]. Sinyal laser diskrit merupakan jenis yang paling sering digunakan pada aplikasi bidang terestrial.



**Gambar 2. 2** Cara Kerja LiDAR [7]

Satelit LiDAR yang memiliki sistem sebagai *remote sensing* pada umumnya terdiri dari *laser range finder* yang mendeteksi jarak dan sistem posisi dan orientasi yang mengukur lokasi dan orientasi dari sensor. Karena LiDAR dapat mengukur lingkungan geografis secara langsung dalam proyeksi tiga dimensi, LiDAR tidak memiliki masalah ketika menemui distorsi geometris seperti pergeseran relief. Ini merupakan salah satu keuntungan dari LiDAR.

Keuntungan lain dari penggunaan lidar adalah dapat mengambil data pada siang hari atau malam hari, selama tidak ada kabut dan asap tebal yang menghalangi atau tingkat kelembaban tinggi seperti hujan, salju dan awan diantara sistem laser dan objek yang ingin dideteksi [6].



**Gambar 2. 3** Gambar LiDAR yang diambil saat malam hari di hutan tropis

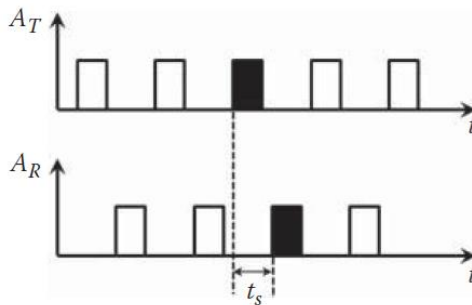
Laser dengan panjang gelombang 500-600nm pada umumnya digunakan pada sistem LiDAR yang berada di tanah. Sedangkan laser dengan panjang 1000-1600nm digunakan pada sistem LiDAR di udara. Sistem LiDAR di udara terdiri dari *laser scanner*; unit pengukuran; terdiri dari unit kendali, pemantauan dan perekaman; *differential global positioning system* (DGPS); dan *inertial measurement unit* (IMU). Sistem yang terintegrasi antara DGPS dan IMU disebut juga sistem posisi dan orientasi yang menghasilkan posisi akurat (lintang, bujur dan ketinggian) dan informasi orientasi (*roll*, *pitch*, dan *heading*). Pola pemindaian laser dapat berbentuk zigzag, parallel atau elips [6]. Pola pemindaian ini dapat dilihat pada gambar 2.5.

Pulsa sinyal laser yang kontinu dan diskrit sama-sama digunakan pada sistem LiDAR. Sistem Lidar dengan pulsa diskrit mengukur waktu perjalanan untuk pulsa cahaya yang pendek dari laser menuju target dan kembali lagi ke *receiver*. Gambar 2.4. menunjukkan amplitude sinyal cahaya yang ditransmisikan ( $A_t$ ) dan amplitude yang diterima ( $A_s$ ). Jika  $c$  merupakan konstanta dari kecepatan cahaya,  $R$  merupakan jarak antara unit pengukuran dan permukaan objek, dan  $t_s$  merupakan waktu tempuh dari pulsa laser, maka parameter-parameter berikut dapat dirumuskan secara matematis pada persamaan 2.10, 2.11, dan 2.12.

$$\text{Jarak} : R = \frac{1}{2} \times c \times t_s \quad (2.10)$$

$$\text{Ketelitian Jarak} : \Delta R = \frac{1}{2} \times c \times \Delta t_s \quad (2.11)$$

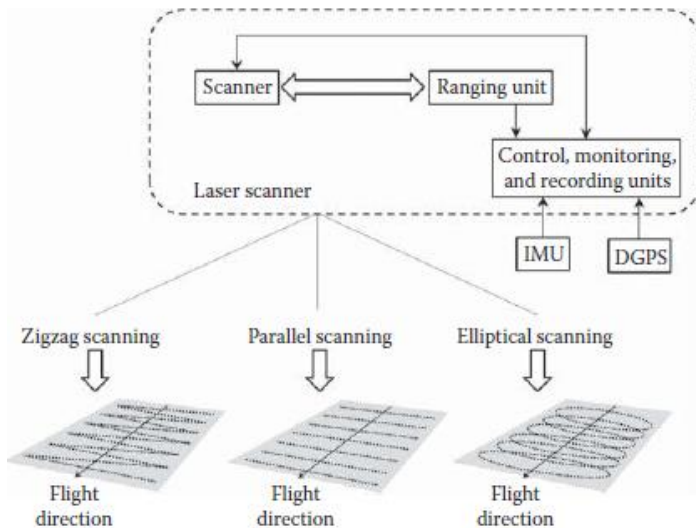
$$\text{Jarak Maksimal} : R_{max} = \frac{1}{2} \times c \times t_{s_{max}} \quad (2.12)$$



**Gambar 2. 4** Waktu tempuh pulsa laser

Akurasi LiDAR biasanya ditentukan dengan komparasi statistik antara jarak sebenarnya yang diketahui dengan jarak yang terukur. Akurasi LiDAR secara khusus diukur dengan simpangan baku ( $\sigma^2$ ) dan *root mean square error* (RMSE). Masalah kesalahan pengukuran pada LiDAR bisa disebabkan oleh kesalahan pada pulsa laser yang dihasilkan. Kesalahan yang dikarenakan pulsa laser pada normalnya disebabkan oleh adanya *noise* kecil dan perubahan pada ketinggian di titik yang memiliki sudut sempit pada permukaan tanah. Hal ini biasanya terjadi pada bukit dan parit.

LiDAR juga diimplementasikan pada bidang perkotaan, seperti ekstraksi jalan, ekstraksi dan rekonstruksi tiga dimensi dari bangunan, perkiraan populasi, penilaian terhadap kerusakan bangunan pasca bencana dan penilaian terhadap kerusakan jalan pasca bencana. Pada bidang kehutanan LiDAR diterapkan untuk memodelkan dan memetakan tinggi dari permukaan kanopi, memodelkan, memetakan dan memperkirakan produksi biomassa, serta memodelkan dan memetakan dari suatu area tertentu.



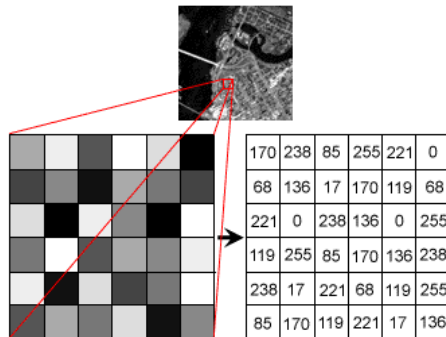
**Gambar 2. 5** Pola pindaian pulsa laser pada LiDAR satelit



### 2.3. Citra Digital

Citra digital merupakan sebuah citra yang didapatkan setelah mengalami proses digitalisasi. Proses digitalisasi merupakan proses yang diperlukan untuk menampilkan sebuah citra digital pada sebuah layer. Proses digitalisasi ini terdiri dari proses sampling dan kuantisasi. Sampling merupakan proses dimana gambar dibagi menjadi kotak-kotak kecil, sedangkan kuantisasi merupakan proses untuk mengisi kotak-kotak hasil sampling dengan nilai tertentu [8]. Gambar 2.6 menunjukkan ilustrasi citra digital dengan setiap piksel diwakilkan oleh integer .

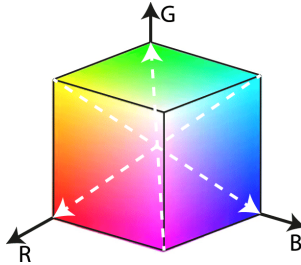
Citra digital ditampilkan oleh kumpulan titik dimana setiap titik akan diwakilkan oleh posisi  $x$ ,  $y$  dan sebuah nilai. Kumpulan titik ini dikenal dengan sebutan piksel. Karena itu, sebuah piksel dapat dinyatakan sebagai fungsi dua dimensi  $f(x,y)$  dengan sumbu  $x$  melintang ke arah horizontal dan sumbu  $y$  ke arah vertikal dengan titik asal  $(0,0)$  berada pada pojok kiri atas dari citra.



**Gambar 2. 6** Citra digital direpresentasikan dengan nilai 8bit

#### 2.3.1. Citra RGB

Citra RGB dihasilkan oleh tiga *channel* warna utama yaitu merah (*red*), hijau (*green*) dan biru (*blue*). Ketiga warna tersebut merupakan warna dasar yang dapat mewakilkan warna-warna lain. Kombinasi dari dua atau lebih warna cahaya dapat menghasilkan warna lain. Oleh karena itu, sebuah warna pada sebuah pixel dapat dinyatakan dengan kombinasi tingkat kecerahan dari masing-masing kanal. *Channel* pada *colorspace* RGB secara tiga dimensi dapat dilihat pada gambar 2.7.

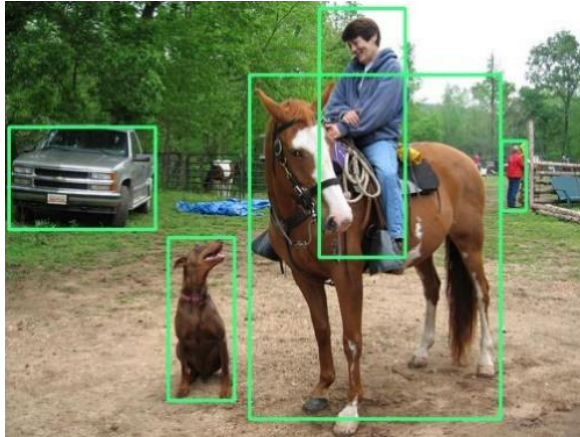


**Gambar 2. 7** Colorspace RGB digambarkan secara 3 dimensi

### 2.3.2. OpenCV

OpenCV merupakan sebuah *open-source library* yang biasa digunakan untuk visi komputer dan machine learning. OpenCV dikembangkan untuk mendukung infrastruktur umum yang digunakan pada aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin pada produk komersial. OpenCV sudah mengembangkan lebih dari 2500 algoritma yang telah dioptimalkan. Algoritma tersebut meliputi algoritma untuk mengenali wajah, mengidentifikasi objek, mengikuti jejak perpindahan objek, dan lain sebagainya, OpenCV dapat diterapkan pada pemrograman dengan bahasa C, C++, Python, Java dan Matlab. OpenCV juga mendukung sistem operasi berbasis Windows, Linux, Android dan Mac OS [9].

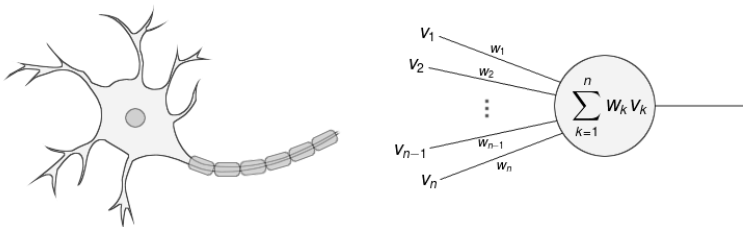
OpenCV memiliki dukungan untuk mengakses modul deep neural network (DNN) sejak meluncurkan versi 3.1. Deep learning merupakan bidang pada *computer vision* yang paling populer dan cepat berkembang pada saat ini. Modul DNN pada OpenCV dapat mengimplementasikan *forward pass* dengan *deep networks*, yaitu jaringan yang sebelumnya sudah dilatih (*pre-trained network*) menggunakan *framework* deep learning yang populer seperti Caffe dan Tensorflow. Pada OpenCV 3.3 modul ini dipindahkan dari repositori `opencv_contrib` menuju ke repositori utama [10]. Semenjak dilakukan pemindahan, modul DNN mengalami perkembangan yang semakin cepat. Gambar 2.8 menunjukkan penggunaan modul DNN pada deteksi objek menggunakan OpenCV.



**Gambar 2. 8** Pengenalan objek menggunakan modul DNN pada OpenCV

## 2.4. Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) merupakan nama yang diberikan pada cabang dari penelitian *artificial intelligence* yang bertujuan untuk menyimulasikan kecerdasan dengan mengimitasi cara kerja jaringan neuron biologis yang ada pada makhluk hidup. Jaringan saraf tiruan dibuat dari kumpulan neuron tiruan yang memiliki peran sama dengan neuron biologis [11]. Neuron tiruan ini berfungsi untuk menerima input, mengubah *internal state* tergantung dari input tersebut dan menghasilkan output tergantung dari input dan internal state. Jaringan saraf tiruan dibentuk dengan menghubungkan output dari satu neuron dengan input pada neuron lainnya.



**Gambar 2. 9** Perbandingan antara neuron biologis dan artifisial [11]

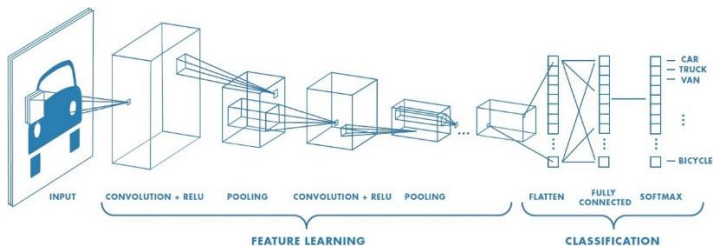
Gambar 2.9. menunjukkan neuron biologis dimana dendrit mengirimkan sinyal kepada badan sel, kemudian badan sel memproses sinyal tersebut dan axon akan mengirimkan sinyal output pada neuron lain. Sinyal input akan diakumulasikan pada badan sel di sebuah neuron. Apabila sinyal yang diakumulasikan melebihi batas tertentu, sinyal output akan dihasilkan dan dikirimkan melewati axon. Gambar 2.9 menunjukkan neuron tiruan yang mengimitasi proses tersebut. Neuron tiruan menerima input sebagai nilai vector  $(x_0, x_1, x_2, \dots, x_k)$  lalu memberikan *weight*  $(w_0, w_1, w_2, \dots, w_k)$  kepada input dan menghasilkan *weighted sum* yang secara matematis dirumuskan pada persamaan 2.13 [11], [12].

$$W_0 + \sum_{i=1}^k w_i x_i \quad (2.13)$$

Selanjutnya fungsi aktivasi  $f$  akan diberikan kepada *weighted sum* tersebut, menghasilkan output dengan nilai  $y$ . Jaringan saraf tiruan terdiri dari satu hingga banyak neuron tiruan yang berkomunikasi satu sama lain. Output dari satu neuron akan menjadi input bagi neuron lainnya. Jaringan saraf tiruan paling sederhana disebut perseptron, yang terdiri dari satu neuron tiruan seperti ditunjukkan pada gambar 2.9.

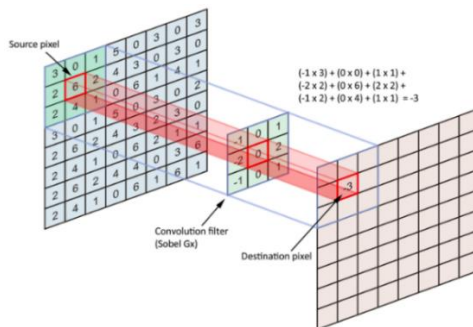
### 2.4.1 Convolutional Neural Network (CNN)

*Convolutional neural network* (CNN) merupakan salah satu jenis *neural network* yang pada umumnya digunakan pada data citra. CNN dapat digunakan untuk mendeteksi dan mengenali sebuah objek yang ada pada citra tersebut. Arsitektur CNN secara umum dibagi menjadi dua bagian, yaitu *feature extraction layer* dan *fully connected layer*.



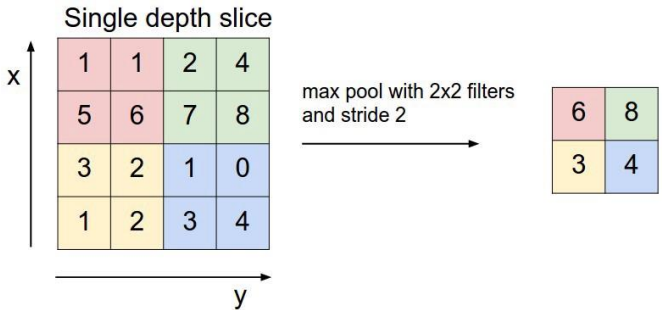
**Gambar 2. 10** Arsitektur CNN secara umum

Arsitektur dari *feature extraction* dan *fully connected layer* dapat dilihat pada gambar 2.10. Pada *feature extraction layer*, dilakukan proses *encoding* dari sebuah citra menjadi fitur berupa angka yang merepresentasikan citra tersebut. *Feature extraction layer* terdiri dari dua bagian, yaitu *convolutional layer* dan *pooling layer*. *Convolutional layer* terdiri dari neuron-neuron yang tersusun membentuk filter dengan panjang dan tinggi dalam pixel. Sebagai contoh, *convolutional layer* untuk gambar RGB dengan ukuran 5x5x3 berarti bahwa layer tersebut memiliki panjang 5 pixel, tinggi 5 piksel dengan tebal 3 yang sesuai dengan channel dari citra tersebut. Ketiga filter dengan ukuran 5x5 ini akan digeser keseluruh bagian citra. Setiap pergeseran akan dilakukan operasi dot antara input citra dan nilai dari filter tersebut sehingga menghasilkan sebuah output yang disebut activation map atau feature map. Layer konvolusi pada CNN diilustrasikan pada gambar 2.11.



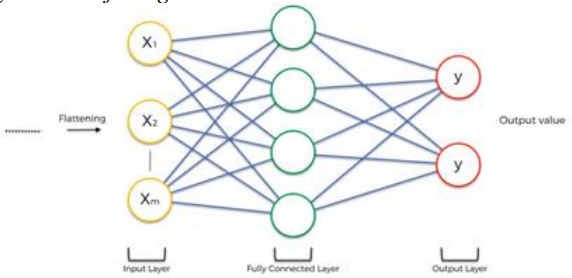
**Gambar 2. 11** Layer konvolusi pada CNN

Pergeseran filter tersebut ditentukan oleh parameter yang disebut *stride*. Jika *stride* bernilai 1, maka filter akan bergeser sebanyak 1 piksel secara horizontal lalu vertikal. Semakin kecil *stride* akan memberikan informasi yang semakin detail, namun memerlukan komputasi yang lebih tinggi dibandingkan jika menggunakan *stride* yang besar. Selain *stride*, ada parameter lain yang disebut *padding* atau *zero padding*. *Padding* merupakan parameter yang menentukan jumlah piksel bernilai 0 yang akan ditambahkan di setiap sisi dari input. *Padding* bertujuan untuk memanipulasi dimensi output dari convolutional layer.



**Gambar 2. 12** Pooling layer pada CNN

*Pooling layer* pada umumnya berada setelah *convolutional layer*. *Pooling layer* terdiri dari sebuah filter dengan dimensi dan *stride* tertentu yang akan bergeser pada seluruh luas *feature map*. *Pooling* yang biasa digunakan adalah *max pooling* dan *average pooling*. Tujuan dari *pooling layer* adalah mengurangi dimensi dari *feature map* sehingga mempercepat komputasi karena parameter yang harus diperbarui akan semakin sedikit dan mengatasi *overfitting*.



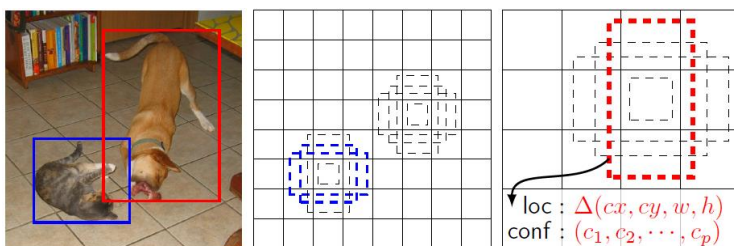
**Gambar 2. 13** Contoh *multi layer perceptron* sederhana

*Feature map* yang dihasilkan dari *feature extraction layer* berupa *array* dengan banyak dimensi, sehingga harus dilakukan *flatten feature map* menjadi vektor agar dapat dijadikan sebagai input untuk *fully connected layer*. *Fully connected layer* ini sebenarnya merupakan sebuah *multi-layer perceptron* (MLP) dengan beberapa *hidden layer*, *activation function*, *output layer* dan *loss function*.

## 2.4.2 Single Shot Multibox Detector

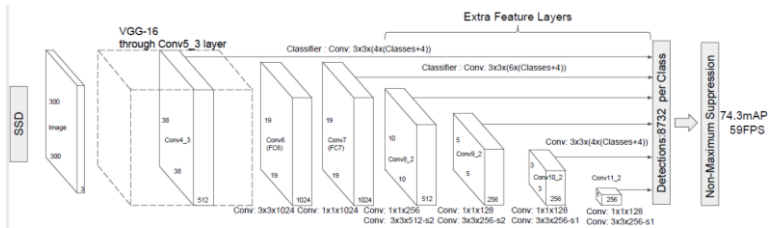
*Single Shot Multibox Detection* merupakan metode untuk mendeteksi objek di dalam sebuah citra menggunakan *single forward pass* dari *deep neural network*. Metode ini memperkirakan dimensi output dari *bounding box* terhadap kumpulan *default box* dengan perbedaan *aspect ratio* dan *scale* untuk setiap lokasi *feature map*. Pada saat prediksi, network akan menghasilkan nilai untuk keberadaan dari setiap kategori objek di setiap *default box*. Kemudian akan dilakukan penyesuaian terhadap *bounding box* untuk dapat menyesuaikan terhadap bentuk objek dengan lebih baik. Network juga akan mengombinasikan prediksi dari banyak *feature map* dengan resolusi yang berbeda-beda untuk dapat mengatasi prediksi objek dengan bentuk yang berbeda [13]. Untuk mengetahui lebih baik tentang apa itu *single shot multibox detection*, maka akan lebih baik dengan mengartikan dari setiap kata, yaitu:

- *Single Shot*, berarti bahwa pekerjaan untuk melakukan lokalisasi objek dan klasifikasi dilakukan dengan *single pass* dari jaringan syaraf tiruan tersebut.
- *Multibox*, merupakan metode untuk melakukan regresi terhadap *bounding box*.
- *Detector*, jaringan syaraf tiruan ini merupakan sebuah *object detector* yang akan mengklasifikasikan objek yang dideteksi.



**Gambar 2. 14** Pendekatan single shot detection [13]

Pendekatan single shot detection memerlukan input citra dan *ground truth box* untuk setiap objek pada saat *training*. Dengan menggunakan gaya konvolusional, metode ini akan mengevaluasi himpunan kecil dari *default box* dengan *aspect ratio* dan *scale* yang berbeda untuk setiap lokasi *feature map*. Untuk setiap *default box*, akan diprediksi *offset* dan *confidence* untuk semua kategori objek.

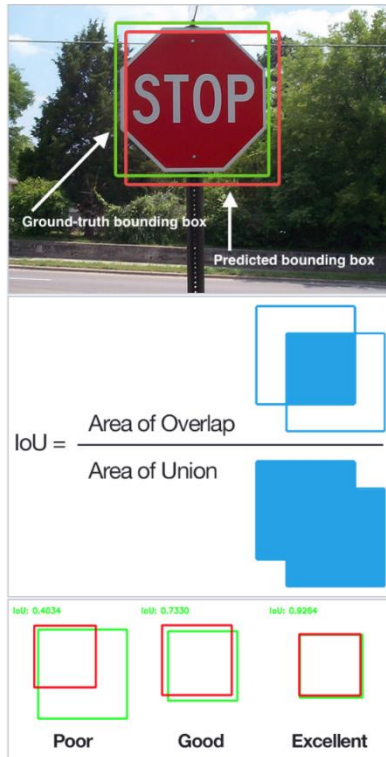


**Gambar 2. 15** Arsitektur Single Shot Multibox Detector

Arsitektur dari SSD seperti yang dapat dilihat pada gambar 2.15 tersusun dari arsitektur VGG-16 namun tanpa *fully connected layer*. VGG-16 digunakan sebagai *base network* karena memiliki performa yang bagus saat melakukan tugas klasifikasi dengan citra resolusi tinggi. *Convolutional layer* yang telah dimodifikasi, dimulai dari *conv6* dan selanjutnya, ditambahkan untuk menggantikan *fully connected layer* asli dari VGG. Layer ini bertujuan untuk melakukan *feature extraction* dan mengurangi dimensi input pada setiap layer selanjutnya.

Teknik untuk melakukan regresi terhadap *bounding box* yang ada pada SSD terinspirasi oleh penelitian tentang *multibox*, yaitu metode untuk menemukan koordinat dan *class* dari *bounding box* dengan cepat [14]. Pada *multibox*, akan dibuat dengan apa yang disebut sebagai *prior*, yaitu *bounding box* yang ditentukan sebelum dilakukan komputasi dengan ukuran tetap dan memiliki ukuran mendekati *ground truth box*. *Prior* ini kemudian dipilih agar memiliki nilai rasio *intersection over union* (IoU) yang lebih besar dari 0.5. Seperti yang dapat dilihat pada gambar 2.16, nilai IoU sebesar 0.5 tidak cukup bagus tetapi nilai ini memberikan permulaan yang bagus untuk melakukan regresi terhadap *bounding box*. Metode seperti ini lebih bagus apabila dibandingkan dengan melakukan regresi dengan koordinat pada gambar yang diacak. Oleh karena itu, *multibox* dimulai dengan *prior* yang bertugas pertama kali untuk melakukan prediksi dan kemudian akan dilakukan regresi yang semakin lama mendekati *ground truth box*. Arsitektur dari *single shot multibox detector* yang ditunjukkan pada gambar 2.15 dapat mencakup gambar *input* pada skala yang luas. Arsitektur ini juga dapat digunakan untuk mendeteksi objek dengan ukuran yang bervariasi. Pada akhirnya, *multibox detector* dapat menghasilkan nilai prediksi yang terdiri dari nilai *bounding box* di sekitar objek yang ingin dideteksi dan nilai dari *confidence* pada *bounding box* tersebut.

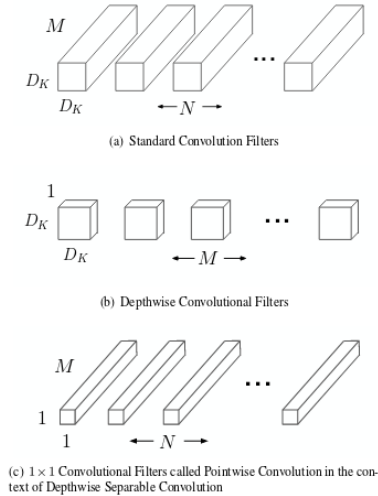




**Gambar 2. 16** IoU dari *ground truth* dan *predicted bounding box*

Jenis *single shot detection* yang digunakan pada tugas akhir ini merupakan jenis SSD MobileNet v2, yang merupakan salah satu jenis arsitektur *convolutional neural network* yang dikembangkan oleh Google dan didesain untuk bekerja pada perangkat portabel agar dapat melakukan klasifikasi, deteksi, rekognisi dan masih banyak lagi [15], [16]. Kemampuan untuk dapat menjalankan *deep network* pada perangkat portable akan meningkatkan pengalaman bagi pengguna. MobilenetV2 merupakan pengembangan dari seri sebelumnya yaitu MobilenetV1. MobilenetV2 merupakan hasil pengembangan yang sangat signifikan untuk mendukung rekognisi visual meliputi klasifikasi, deteksi objek dan segmentasi.

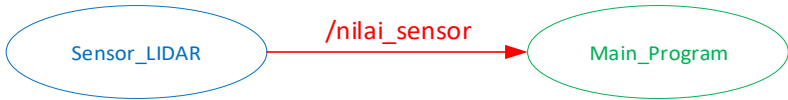
Perbedaan antara arsitektur MobileNet dengan CNN pada dasarnya adalah penggunaan *layer* konvolusi dengan ketebalan filter yang disesuaikan dengan ketebalan *input image*. MobileNet membagi *layer* konvolusi menjadi *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* melakukan konvolusi pada satu *channel* input dalam satu waktu.



**Gambar 2. 17** Perbedaan layer konvolusi MobileNet dan CNN pada umumnya

## 2.5. Robot Operating System

Robot operating system merupakan framework fleksibel yang digunakan untuk menulis program pada robot. ROS terdiri dari kumpulan tool dan library yang bertujuan untuk menyederhanakan tugas yang diperlukan untuk menghasilkan robot yang kompleks dan kuat serta bisa dijalankan pada platform robotik yang luas [17]. Untuk mengatasi variasi pada dunia nyata tentang betapa rumitnya tugas dan lingkungan yang dibebankan kepada robot merupakan hal yang sangat sulit untuk dilakukan oleh hanya satu individu, laboratorium atau institusi. Oleh karena itu, ROS dibangun dari dasar dengan semangat pengembangan perangkat lunak yang kolaboratif [18], [19].



**Gambar 2. 18** Hubungan publisher, subscriber dan message

### 2.5.1. ROS Node

*Node* merupakan proses yang melakukan komputasi dalam ROS. *Node* dapat dikombinasikan dan melakukan komunikasi antar node menggunakan *streaming topic* [18]. Sebuah robot biasanya memiliki banyak *node*. Sebagai contoh pada sebuah *mobile robot* memiliki *node* pertama yang mengendalikan *laser range finder*, *node* kedua yang mengendalikan motor penggerak, *node* ketiga melakukan lokalisasi, *node* keempat melakukan *path planning*, dan masih banyak lagi.

Penggunaan *node* dalam ROS menghasilkan beberapa manfaat untuk keseluruhan sistem. Penggunaan *node* dapat memberikan kemudahan apabila ingin melakukan *troubleshooting* karena kesalahan terjadi dalam *node* yang spesifik. Selain itu penggunaan *node* juga mengurangi kerumitan dalam pemrograman dibandingkan dengan sistem yang monolitik. *Node* dapat mengirimkan tipe data yang berbeda ke *node* lain seperti *integer*, *float* dan *string*. Tipe data berbeda yang dikirimkan disebut *ROS message*. Dengan menggunakan *ROS message*, tipe data baik *single* maupun *multiple* dapat dikirimkan antara *node*. *Message* ini dikirimkan melalui jalur yang disebut *ROS topic*. Setiap *topic* memiliki nama yang berbeda-beda [18]. *ROS topic* dan *message* akan dijelaskan lebih lanjut pada sub 2.5.2 dan 2.5.3.

Setiap *node* dalam sistem robot memiliki nama yang secara unik membedakan setiap *node* yang bekerja di dalam robot. Sebagai contoh dapat dilihat pada gambar 2.18 dimana node diwakilkan oleh bentuk elips. Pada gambar 2.18 terdapat dua node yaitu node *sensor\_LiDAR* dan *main\_program*. Node *sensor\_LiDAR* merupakan *node* yang menghasilkan nilai pembacaan sensor LiDAR.

### 2.5.2. ROS Topic

*Topic* merupakan suatu *bus* dimana *node* bertukar pesan. Secara umum, *node* tidak memperhatikan dengan siapa mereka berkomunikasi. Tetapi *node* memperhatikan data yang di *subscribe* terhadap *topic* yang relevan. Dengan kata lain *node* yang menghasilkan data yang di publish terhadap *topic* yang relevan. Dalam satu *topic* bisa terdapat banyak subscriber dan publisher [20].

*Topic* berperan untuk *streaming* komunikasi unidireksional. Pada gambar 2.18 dapat dilihat terdapat dua node yaitu `sensor_LiDAR` dan `main_program`. Kedua node ini berkomunikasi melewati satu topic yang sama yaitu `/nilai_sensor`.

### 2.5.3. ROS Message

ROS menggunakan metode yang sudah disederhanakan untuk mendeskripsikan nilai dari data yang di-*publish* oleh *node*. Data yang di *publish* ini kemudian disebut sebagai *message*. Hal ini memberikan kemudahan untuk ROS agar dapat menghasilkan kode program yang digunakan untuk mengirimkan *message* dalam beberapa bahasa pemrograman yang berbeda [21].

Komunikasi yang terjadi melalui *topic* berjalan dengan mengirimkan *messages* di antara *node*. Agar *publisher* (`sensor_LiDAR`) dan *subscriber* (`main_program`) bisa berkomunikasi, *publisher* dan *subscriber* harus mengirimkan dan menerima jenis *messages* yang sama. Ini berarti jenis *topic* didefinisikan oleh jenis *message* yang di-*publish* di dalamnya.

### 2.5.4. ROS Publisher

ROS Publisher merupakan *node* yang akan secara terus menerus mengirimkan *message* saat program dijalankan. Pada gambar 2.18, *publisher node* ditunjukkan oleh bentuk dengan warna biru. *Node* `sensor_LiDAR` akan mengirimkan nilai pembacaan LiDAR secara terus menerus ke program utama apabila program dijalankan. Ketika sebuah *node* mem-*publish* sebuah *topic*, maka *node* tersebut akan mengirimkan *ROS topic* yang berisi *ROS message* [18], [19].

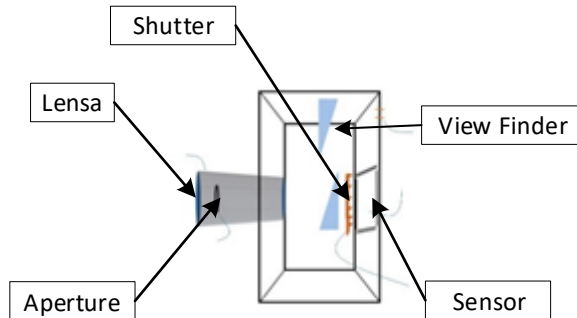
### 2.5.5. ROS Subscriber

ROS subscriber merupakan *node* yang akan menerima *message* saat menjalankan program [18], [19]. Pada gambar 2.18, *subscriber node* ditunjukkan oleh bentuk elips dengan warna hijau. *Node* `main_program` akan menerima nilai pembacaan

## 2.6. Kamera

Kamera merupakan sensor optis yang berfungsi untuk merekam gambar diam atau bergerak dan kemudian disimpan ke media penyimpanan digital atau film fotografis [22]. Jenis kamera paling sederhana dapat melakukan tugas ini adalah *pinhole camera*. *Pinhole camera* dapat bekerja hanya dengan satu atau tanpa ada bagian yang

bergerak. Sementara kamera modern dapat memiliki banyak bagian bergerak seperti kamera *single lens reflex*.



**Gambar 2. 19** Bagian-bagian dasar penyusun kamera

Kamera modern kurang lebih bekerja dengan prinsip yang sama untuk menghasilkan gambar. Bagian-bagian kamera pada umumnya ditunjukkan pada gambar 2.19. Beberapa kamera jelas bekerja dengan cara yang lebih rumit dari kamera lainnya, tetapi secara umum cahaya melewati bagian-bagian yang sama ketika di dalam kamera. Bagian-bagian ini meliputi lensa, *aperture*, *shutter*, dan *image plane* [22]. Bagian yang membedakan bagaimana gambar dapat dilihat pada kamera adalah hal yang menyebabkan adanya jenis kamera yang berbeda-beda. Gambar tersebut dapat dilihat pada layer elektronis atau jendela bidik.

Lensa merupakan bagian yang biasanya terbuat dari plastik, kaca atau kristal dan menjadi bagian pertama pada kamera yang dimasuki cahaya. Lensa akan membelokkan cahaya yang memasuki kamera menuju *image plane* atau bidang gambar. Beberapa lensa memiliki fokus yang tetap, namun ada juga beberapa kamera dengan komponen yang dapat diubah untuk mengatur fokus dari lensa.

*Aperture* atau celah adalah bagian yang secara teknis bagian dari lensa dimana merupakan ukuran dari bukaan celah lensa [22]. Banyak desain yang memiliki *aperture* yang dapat diubah-ubah sehingga dapat mengatur seberapa banyak cahaya yang masuk melalui lensa. Tetapi ada juga beberapa lensa yang memiliki *aperture* tetap sehingga ukurannya tidak dapat diubah-ubah. Semakin besar *aperture* membuka akan menyebabkan semakin banyak cahaya yang masuk, tetapi kedalaman gambar akan berkurang. Hal ini dapat dilihat pada gambar 2.20 dimana

gambar kiri diambil dengan *aperture* yang lebih kecil dibandingkan dengan gambar kanan.



**Gambar 2. 20** Gambar yang diambil dengan aperture kamera yang berbeda [22]

Pada kamera terdapat bagian yang membuka dan menutup sehingga dapat membiarkan cahaya menyinari bidang gambar dalam waktu tertentu. Bagian ini disebut *shutter* dan bekerja seperti ketika tirai yang membuka dan menutup [22]. Setelah cahaya melewati *aperture* lensa dan *shutter* yang terbuka, selanjutnya cahaya akan menabrak bidang gambar. Gambar yang diambil dengan *shutter speed* berbeda dapat dilihat pada gambar 2.21 dimana gambar kiri diambil dengan *shutter speed* lebih lambat dibandingkan gambar kanan.



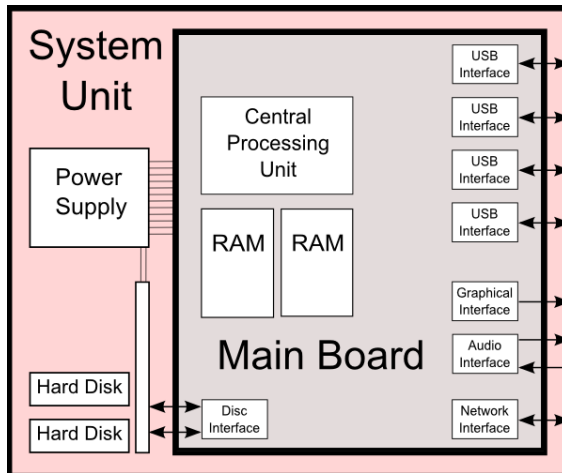
**Gambar 2. 21** Gambar yang diambil dengan shutter speed berbeda

Pada bidang gambar terdapat film yang sensitif terhadap cahaya atau sensor digital dimana gambar yang terproyeksi akan direkam. Sensor digital yang pada umumnya digunakan merupakan CCD atau CMOS. *Charge-coupled device* (CCD) merupakan sensor yang dikendalikan oleh arus. *Charge* akan dikumpulkan pada setiap piksel sensor dan kemudian akan digeser dari permukaan sensor ke output (*shift register*) untuk dilakukan sampling. *Output* dari CCD merupakan pulsa analog dimana

*charge* akan proporsional terhadap intensitas cahaya [23]. *Complimentary metal-oxide semiconductor* (CMOS) merupakan sensor yang dikendalikan oleh tegangan. Cahaya yang menabrak setiap piksel sensor akan menghasilkan tegangan yang proporsional terhadap intensitas cahaya. Tegangan kemudian mengalami *sampling* secara langsung pada setiap piksel. Hal ini menyebabkan output sensor CMOS merupakan digital [23].

## 2.7. Komputer mini Intel NUC

Komputer mini yang digunakan pada tugas akhir ini merupakan jenis Intel NUC6i7KYK. Komputer mini ini dilengkapi prosesor core i7 generasi ke-6 dengan empat *core* dan *clock speed* 2.6GHz hingga mencapai 3.5GHz [24]. Pada sisi grafis, komputer mini ini dilengkapi dengan Iris Pro Graphic 580 untuk memaksimalkan performa dari beban kerja yang banyak. Iris Pro Graphic secara performa dapat disandingkan dengan Nvidia GeForce 945M. Iris Pro Graphic juga memiliki kapabilitas untuk menampilkan gambar dengan resolusi hingga 4K [25]. Pada komputer mini ini juga memiliki satu port HDMI 2.0 dan satu port Thunderbolt 3.



**Gambar 2. 22** Bagian sistem komputer secara umum

Pada sisi memori dan penyimpanan, komputer mini ini memiliki dua socket RAM DDR4 SO-DIMM dan dua konektor M.2 tipe m yang dapat

dihubungkan dengan SATA atau SSD [24]. Di dalam komputer mini ini sudah terpasang RAM sebesar 8GB pada satu dari dua socket RAM yang tersedia. Sedangkan pada penyimpanan sudah terpasang SSD dengan kapasitas 120GB.

Pada dukungan konektivitas periferal, terdapat empat buah port USB 3.0 dengan kecepatan tinggi, satu port LAN dan dukungan konektivitas bluetooth 4.2. Untuk menyalakan komputer mini ini, diperlukan *power supply* dengan kemampuan untuk menyuplai tegangan DC 19V dan daya 120W. Penjelasan konektivitas antar komponen yang ada di dalam komputer mini dapat dilihat pada gambar 2.22.

## 2.8. Microcontroller STM32F4 Discovery

STM32F4 Discovery merupakan mikrokontroler yang menggunakan ARM 32bit. Mikrokontroler ini dibekali oleh inti prosesor ARM®Cortex®-M4 dengan kecepatan clock hingga 168 MHz. Pada board STM32F407 Discovery ini sudah tertanam beberapa fitur yang akan digunakan pada tugas akhir ini seperti *DMA controller*, *external interrupt controller*, *timer*, USART, GPIO pins dan *analog to digital conversion* (ADC) [26, p. 32], [27], [28]. Board STM32F4 Discovery ini sudah memiliki downloader ST-LINK sehingga untuk memasukkan program hanya dibutuhkan koneksi usb dengan konektor tipe usb mini. Fitur-fitur STM32F4 discovery diantaranya sebagai berikut:

- ARM cortex-M4 core dengan FPU dan *embedded Flash and SRAM*  
Prosesor RISC 32 bit ARM Cortex-M4 dengan FPU memiliki efisiensi kode yang luar biasa, memberikan kinerja tinggi yang diharapkan dari ARM *core* dalam ukuran memori biasanya dengan perangkat 8 dan 16 bit.
- *Embedded Flash Memory*  
STM32F4 memiliki memori *flash* 512 Kbytes atau 1 Mbytes tersedia untuk menyimpan program dan data.
- *Cycling Redundancy Check*  
CRC digunakan untuk mendapatkan kode CRC dari 32 bit kata data dan polynomial generator tetap.
- *Embedded SRAM*  
Semua produk stm32f4 terdapat 192 Kbytes system SRAM termasuk 63 Kbytes CCM (*Core Coupled Memory*) dan 4 Kbytes SRAM cadangan area ini hanya dapat diakses dari CPU.
- Multi AHB bus *matrix*

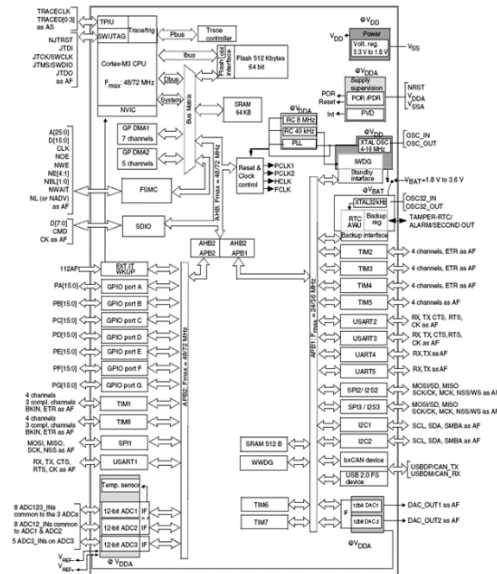


Digunakan untuk menghubungkan semua master (CPU, DMA, Ethernet, USB HS) dan *slave* (*Flash memory*, RAM, FSMC, AHB dan APB periperal).

- *DMA controller*  
Perangkat ini memiliki dua DMA, yaitu (DMA1 dan DMA2) dengan *stream* masing-masing. DMA dapat digunakan dengan periperal utama : SPI dan I2S, I2C, USART, TIMx, DAC, SDIO, DCM1 dan ADC.
- *Nested vectored interrupt controller* (NVIC)  
STM32F menanamkan pengontrol interferensi bertingkat *vector* mengelola 16 tingkat prioritas, dan menangani hingga 82 *channel* dan ditambah 16 interupsi *channel* cortex-M4 dengan FPU *core*.
- *External interrupt/event controller* (EXTI)  
Pengontrol *interrupt / event ekxternal* terdiri dari 23 garis *edge detector* yang digunakan untuk menghasilkan permintaan interupsi / *event*. Setiap baris dapat dikonfigurasi secara independen untuk memilih pemicu *event* (*rising edge, falling edge, both*).
- *Timer and watchdogs*  
STM32F4 memiliki dua *advance control timer*, delapan *timer* umum, dua *basic timer* dan dua *watchdog timer*.  
*Advance control timers* (TIM1, TIM8). Dapat dilihat sebagai generator pwm tiga fasa multiplexing pada 6 saluran. Empat saluran independen bisa digunakan untuk input *capture*, output *compare*, pwm generation dan *one pulse mode* output.  
*General purpose timer*, ada sepuluh *general purpose timer* yang dapat disinkronkan pada STM32F4. *General purpose timer* yaitu TIM2, TIM3, TIM4, TIM5, TIM9, TIM10, TIM11, TIM12, TIM13 dan TIM14.  
*Basic timers* TIM6 dan TIM7, *timer* ini digunakan untuk pemicu DAC dan pembentukan gelombang. Dan juga bisa digunakan sebagai 16 bit *time base*.  
*Independent watchdog*, didasarkan pada 12 bit *downcounter* dan 8 bit *prescaler*, dan dapat digunakan untuk menagtur ulang perangkat ketika masalah terjadi atau sebagai *free running timer* untuk batas waktu aplikasi.  
*Window watchdog*, didasarkan pada *downcounter* 7bit yang dapat diatur sebagai *free running*. Dapat untuk mengatur ulang

perangkat ketika masalah terjadi dan memiliki kemampuan interupsi peringatan dini.

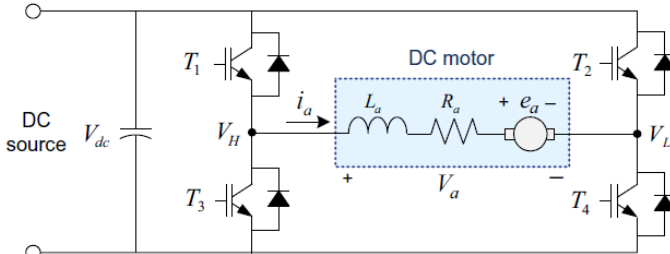
- *Universal synchronous / asynchronous receiver transmitters (USART)*  
STM32F4 terdapat empat *universal synchronous* atau *asynchronous receiver transmitter* (USART1, USART2, USART3 dan USART6), dua *universal asynchronous receiver transmitter* (UART4 dan UART5).
- *General purpose input / output (GPIO)*  
Setiap pin GPIO dapat dikonfigurasi oleh perangkat lunak sebagai output (*push pull, open drain*, dengan atau tanpa *pull up* atau *pull down*), sebagai input (*floating*, dengan atau tanpa *pull up* atau *pull down*). Sebagai fungsi *alternative peripheral*, digital atau analog. Penanganan cepat I/O maksimum 84 MHz
- *Analog to digital converters (ADC)*  
Tiga ADC 12bit tertanam dan setiap ADC memiliki 16 channels eksternal, melakukan konversi dalam mode *single shot* atau *scan*.



**Gambar 2. 23** Blok diagram konektivitas STM32F407

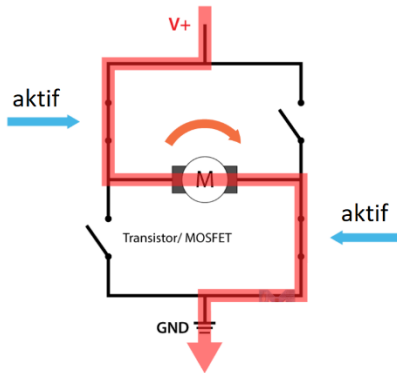
## 2.9. Driver Motor

Driver motor merupakan rangkaian yang digunakan untuk mengatur direksi dan kecepatan putar motor. Rangkaian ini akan menguatkan arus yang masuk ke motor dari unit pengendali, dalam kasus ini berupa mikrokontroller, sehingga arus yang masuk ke motor akan cukup besar untuk memutar motor. Rangkaian driver motor yang sering digunakan adalah rangkaian dengan konfigurasi H-bridge. Kata “bridge” berasal dari bentuk rangkaian yang memiliki bentuk seperti empat lengan dan dianggap sebagai jembatan [29]. Rangkaian konfigurasi H-bridge secara sederhana dapat dibuat dengan menggunakan transistor yang dapat dilihat pada gambar 2.24.



**Gambar 2. 24** Konfigurasi rangkaian H-bridge sederhana [30]

Pada gambar 2.24, konfigurasi H-bridge dibuat dengan menggunakan empat transistor BJT yang terdiri dari empat buah transistor NPN. Selain dengan menggunakan transistor BJT, rangkaian H-bridge juga dapat dibuat menggunakan transistor MOSFET [30]. Empat transistor ini disusun sehingga menyerupai bentuk huruf H. Konfigurasi H-bridge ini memiliki dua input yaitu input A dan B. Arah putaran motor dapat diatur dengan mengatur *state* dari input A dan B. Apabila *state* dari A diatur *HIGH* dan B diatur *LOW*, maka rangkaian H-bridge akan menjadi seperti gambar 2.25 dimana transistor Q1 dan Q4 akan mengalirkan arus dari VCC menuju GND. Aliran arus melewati transistor Q1 dan Q4 selanjutnya akan menyebabkan motor dapat berputar searah jarum jam. Apabila *state* dari A dan B diubah menjadi sebaliknya, maka transistor Q1 dan Q4 akan berubah menjadi tidak aktif. Arus akan mengalir melalui transistor Q2 dan Q4 melewati motor sehingga arah putaran akan berlawanan arah jarum jam. Umumnya, teknik *switching* yang digunakan adalah *pulse width modulation* (PWM) untuk mencapai tegangan yang dibutuhkan agar dapat mengendalikan kecepatan dari motor DC [30].

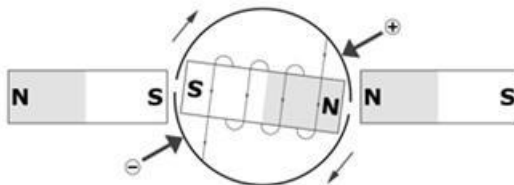


**Gambar 2. 25** Arah putaran motor ketika diberikan input

### 2.10. Motor DC

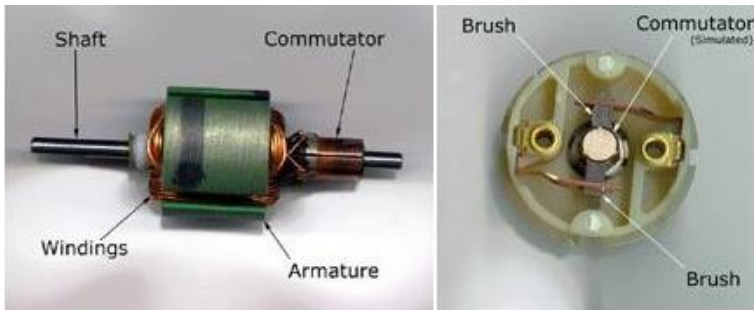
Motor DC merupakan perangkat yang berfungsi untuk mengubah energi listrik DC menjadi energi mekanik dalam bentuk gerakan rotasi. Gerakan rotasi ini disebabkan karena sifat elektromagnet. Motor DC memiliki induktor yang menghasilkan medan magnet untuk menghasilkan gerakan. Di dalam motor DC, sebuah elektromagnet, yang merupakan sebuah besi dililitkan dengan kumparan kawat, diberikan tegangan pada kedua ujung terminalnya. Apabila dua magnet permanen ditambahkan pada kedua sisinya, maka akan timbul gaya tarikan dan dorongan yang menghasilkan torsi [31].

Lalu akan timbul masalah baru, yaitu bagaimana cara untuk memberikan arus kepada elektromagnet agar tetap dapat berputar dan membalikkan arah arus pada saat yang tepat. Masalah-masalah ini dapat diatasi dengan menambahkan komutator berbentuk cincin-belah dan sepasang *brush*.



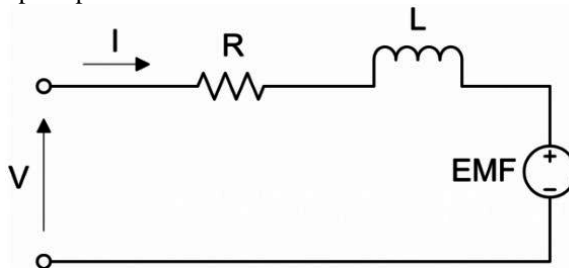
**Gambar 2. 26** Gaya tolak saat elektromagnet dialiri arus

Pada gambar 2.27, dapat dilihat bahwa komutator memiliki dua segmen dimana setiap segmen terhubung kepada setiap terminal dari elektromagnet. Di samping komutator, yang ditunjukkan oleh kedua panah, merupakan *brush* yang berfungsi untuk memberikan arus listrik kepada elektromagnet yang berputar.



**Gambar 2. 27** Sikat dan cincin belah pada motor DC

Dengan menggunakan mekanisme seperti ini, sejalan dengan elektromagnet yang bergerak maka polaritasnya juga akan berubah dan menyebabkan poros motor berputar. Walaupun mekanisme seperti ini terdengar mudah, ada beberapa isu yang menyebabkan motor ini tidak efisien dalam penggunaan energi dan secara mekanis. Secara prinsip, penyebab masalahnya adalah karena ada perbedaan waktu antara setiap pembalikan polaritas. Karena polaritas pada elektromagnet berubah secara mekanis, pada saat kecepatan tertentu polaritasnya akan berubah terlalu cepat atau terlambat sehingga menyebabkan impuls yang berlawanan. Impuls ini kemudian menghasilkan gerakan berhenti yang sementara pada putaran.



**Gambar 2. 28** Rangkaian ekuivalen motor DC [29]

Rangkaian ekuivalen motor DC dapat dilihat pada gambar 2.28. Dengan menggunakan analisa loop tegangan, didapatkan bahwa tegangan input ( $V$ ) sama dengan beda tegangan pada lilitan dan induktor ( $L$ ) dengan EMF ( $E$ ) yang dihasilkan oleh gerakan motor. Secara matematis dirumuskan pada persamaan 2.14 [29].

$$V = IR + L \frac{dI}{dt} + E \quad (2.14)$$

Tetapi untuk menunjukkan efek tegangan dan torsi kepada kecepatan, keadaan hanya diamati pada *steady-state*. Pada *steady-state*, arus bernilai konstan sehingga induktor  $L$  dapat dihilangkan dari persamaan. Kemudian karena tidak ada perubahan kecepatan, torsi yang dihasilkan oleh motor dan beban diasumsikan bernilai sama. Nilai torsi yang sama ini menghasilkan persamaan 2.15. dimana  $K_t$  mewakili konstanta yang didapatkan dari desain motor,  $\phi$  mewakili nilai flux dan  $T$  merupakan nilai torsi.

$$T = K_T \phi I \quad (2.15)$$

Dengan menyubstitusi persamaan 2.15. ke persamaan 2.14. kemudian secara matematis didapatkan persamaan baru yang dapat dilihat pada persamaan 2.16.

$$V = \frac{T}{K_t \phi} R + E \quad (2.16)$$

Selanjutnya nilai EMF ( $E$ ) merupakan nilai yang tergantung pada total flux pada motor ( $\phi$ ), sebuah konstanta  $K_E$ , dan kecepatan motor ( $n$ ). Nilai EMF secara matematis dirumuskan pada persamaan 2.16 [29]. Dengan memasukkan persamaan 2.17. pada persamaan 2.14. akan didapatkan persamaan baru pada 2.18.

$$E = K_E \phi n \quad (2.17)$$

$$V = \frac{T}{K_T \phi} R + K_E \phi n \quad (2.18)$$

Persamaan 2.18. merepresentasikan motor linear. Untuk mengadaptasikan persamaan tersebut ke motor yang berotasi, diasumsikan flux bernilai konstan yaitu pada nilai maksimal. Sehingga persamaan yang diperlukan untuk mencari kecepatan linear dan sudut dapat dilihat pada persamaan 2.19 dan 2.20.

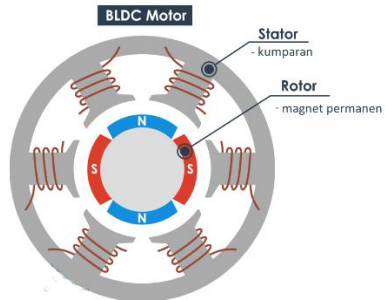
$$V = \frac{T}{k} R + k\omega \quad (2.19)$$

$$\omega = \frac{V}{k} - \frac{T}{k^2} R \quad (2.20)$$

Secara singkat, melalui persamaan terakhir dapat dilihat kecepatan motor tergantung pada variabel seperti tegangan input dan torsi beban. Untuk meningkatkan kecepatan maka tegangan input yang diberikan harus ditingkatkan [31].

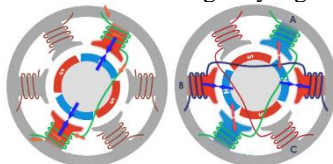
## 2.11. Motor Brushless DC

Motor brushless DC, juga dikenal sebagai motor DC sinkron, merupakan jenis motor yang banyak digunakan pada perangkat komputer dan industri dirgantara dimana sangat penting untuk memiliki motor dengan performa tinggi yang dipasangkan dengan keandalan [29]. Motor brushless terdiri dari dua bagian utama yaitu stator dan rotor. Untuk ilustrasi, rotor merupakan magnet permanen dengan dua kutub sedangkan stator merupakan kumparan. Ilustrasi mekanisme motor dan stator dapat dilihat pada gambar 2.29.



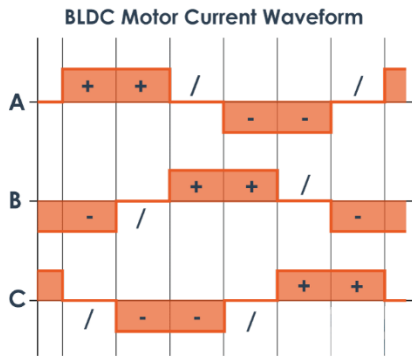
**Gambar 2. 29** Bagian utama pada BLDC

Apabila arus diberikan melalui kumparan tentu saja akan menghasilkan medan magnet yang kutub-kutubnya bergantung pada arah arusnya. Apabila arus yang tepat diberikan, maka kumparan akan menghasilkan medan magnet yang akan menarik magnet permanen pada rotor. Ketika setiap kumparan diaktifkan secara berurutan, satu kumparan setelah yang lainnya, maka rotor akan tetap berputar karena interaksi gaya antara magnet permanen dan elektromagnet yang dihasilkan [29].



**Gambar 2. 30** Medan magnet pada BLDC

Untuk meningkatkan efisiensi motor, dua kumparan yang berlawanan dapat dililitkan sebagai satu kumparan. Hal ini menghasilkan kutub yang berlawanan terhadap kutub di rotor, sehingga memberikan dua gaya tarik. Dengan konfigurasi ini, enam kutub dapat dibuat pada statur dengan hanya menggunakan tiga kumparan atau fasa. Selanjutnya efisiensi dapat ditingkatkan dengan menyalakan dua kumparan pada saat bersamaan sehingga satu kumparan akan menarik dan kumparan lainnya akan mendorong rotor. Mekanisme ini diilustrasikan pada gambar 2.30.



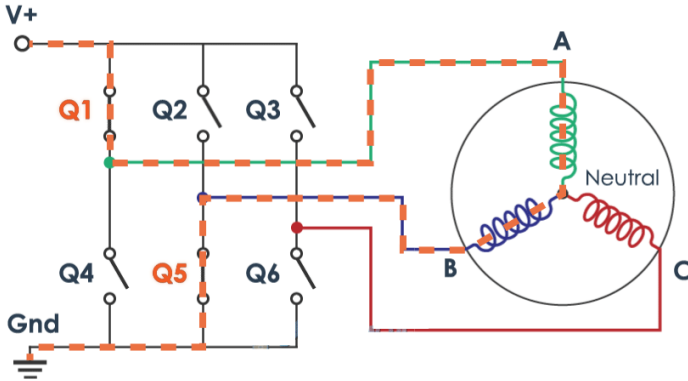
**Gambar 2. 31** Bentuk gelombang pada fasa BLDC

Untuk menghasilkan motor dengan putaran sebesar 360 derajat, diperlukan enam interval seperti ditunjukkan pada gambar 2.31. Jika diperhatikan bentuk gelombang arus, dapat dilihat bahwa pada setiap interval ada satu fase dengan arus positif, satu fase dengan arus negatif dan satu fase yang mati. Hal ini memberikan ide bahwa setiap ujung fasa dapat disambungkan secara bersamaan sehingga arus dapat dibagikan atau dengan kata lain menggunakan satu arus untuk menyalakan dua fasa pada saat yang bersamaan.

Sebagai contoh dapat dilihat pada gambar 2.32. Ketika fase A dihubungkan kepada sumber tegangan positif dan pada sisi lainnya fase B dihubungkan dengan ground, lalu diantaranya dihubungkan dengan semacam switch seperti MOSFET, maka arus akan mengalir dari sumber tegangan positif melalui fase A, titik netral, fase B dan ground. Jadi dengan hanya menggunakan satu aliran arus, empat kutub yang berbeda dapat dihasilkan untuk menggerakkan rotor. Dengan konfigurasi seperti ini, akan terbentuk konfigurasi star pada fasa-fasa motor seperti gambar 2.32., dimana titik netral secara internal meghubungkan masing-masing

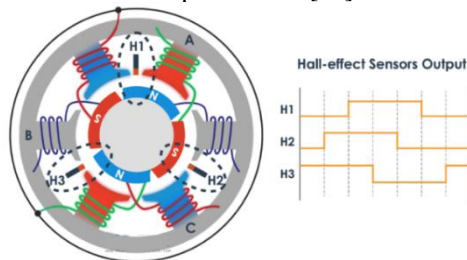


ujung fasa dan ujung lainnya keluar dari motor [29], [30]. Hal ini menyebabkan mengapa motor brushless dc memiliki tiga kabel. Jadi untuk menghasilkan siklus penuh sehingga dapat berputar, dua MOSFET yang benar harus diaktifkan pada setiap enam interval. Oleh karena itu dibutuhkan *electronic speed controller* (ESC).



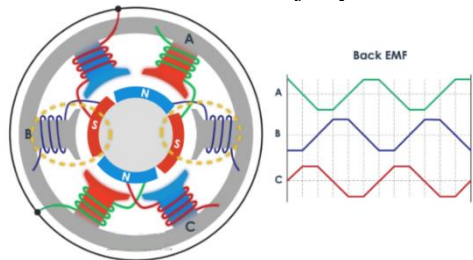
**Gambar 2. 32** Contoh aliran arus pada fasa motor brushless

ESC mengendalikan kecepatan motor brushless dengan mengaktifkan MOSFET yang diperlukan untuk menghasilkan *rotating magnetic field* (RMF) sehingga motor dapat berputar. Semakin tinggi frekuensinya, dengan kata lain semakin cepat ESC menempuh enam interval, maka semakin tinggi juga kecepatan motor yang dihasilkan. Tetapi kemudian akan muncul pertanyaan baru, yaitu bagaimana cara untuk mengetahui fasa mana yang harus diaktifkan. Jawabannya adalah dengan mengetahui posisi rotor dan ada dua metode umum yang digunakan untuk menentukan posisi rotor [32].



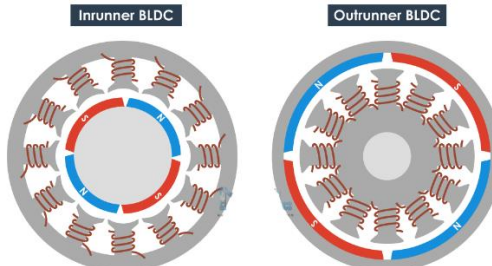
**Gambar 2. 33** Keluaran pembacaan sensor *hall-effect*

Metode pertama adalah dengan menggunakan sensor *hall-effect* yang ditanamkan pada stator dengan penempatan setiap 120 atau 60 derajat antara satu sensor dengan yang lainnya. Ketika rotor yang terdiri dari magnet permanen berputar, sensor *hall-effect* akan merasakan medan magnet dan menghasilkan logika *high* untuk satu kutub dan *low* untuk kutub lainnya [32]. Berdasarkan informasi ini ESC akan mengetahui saat untuk mengaktifkan interval komutasi selanjutnya.



**Gambar 2. 34** Keluaran pembacaan EMF

Metode kedua untuk menentukan posisi rotor adalah dengan menggunakan *electromotive force* balik yang dihasilkan, atau disebut sebagai *back EMF*. *Back EMF* dihasilkan oleh kebalikan dari proses menghasilkan medan magnet, yaitu saat menggerakkan atau mengubah medan magnet melewati kumparan maka akan menginduksi arus pada kumparan tersebut [32]. Jadi ketika medan magnet pada rotor bergerak melewati kumparan yang sedang tidak aktif, medan magnet ini akan menginduksi arus pada kumparan tersebut sehingga akan timbul beda tegangan. ESC akan menangkap beda tegangan ini saat terjadi dan berdasarkan hal tersebut ESC akan memperhitungkan kapan interval selanjutnya harus terjadi.



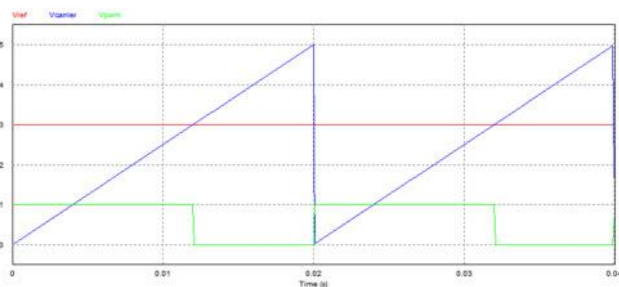
**Gambar 2. 35** Jenis-jenis BLDC motor

Motor DC brushless akan tetap memiliki fasa yang sama, yaitu tiga fasa, walaupun jumlah kutub pada stator dan rotor ditambahkan. Parameter yang berubah hanya interval yang diperlukan untuk menyelesaikan satu siklus putaran. Motor DC brushless bisa merupakan jenis *outrunner* dan *inrunner*. Motor *inrunner* brushless memiliki magnet permanen pada bagian stator, sedangkan motor *outrunner* brushless memiliki magnet permanen pada bagian rotor. Dua jenis motor ini bekerja dengan prinsip yang sama seperti yang sudah dijelaskan sebelumnya dan jenis-jenis ini memiliki keunggulan dan kekurangan masing-masing.

## 2.12. Pulse Width Modulation (PWM)

*Pulse Width Modulation* (PWM) merupakan sebuah metode untuk memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode untuk mendapatkan tegangan rata-rata yang berbeda. Aplikasi PWM berbasis mikrokontroler biasanya berupa pengendalian kecepatan motor DC, pengendalian servo atau pengaturan nyala terang LED [33].

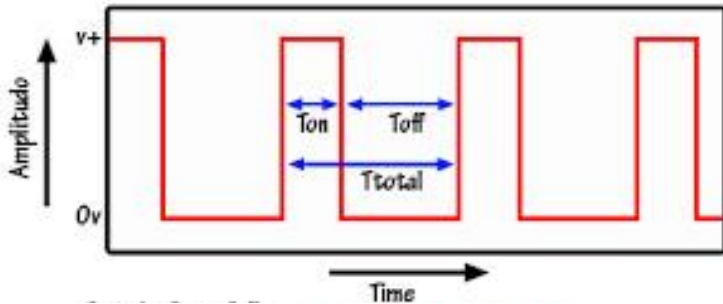
Pembangkitan sinyal PWM yang paling dasar adalah dengan cara membandingkan sinyal *sawtooth* sebagai tegangan *carrier* dengan tegangan referensi menggunakan rangkaian *op-amp comparator*. Saat nilai tegangan referensi lebih besar dari tegangan *carrier* (*sawtooth*) maka output *comparator* akan bernilai *high*. Namun saat tegangan referensi bernilai lebih kecil dari tegangan *carrier* maka output *comparator* akan bernilai *low*. Dengan memanfaatkan prinsip kerja dari *comparator* ini kita dapat mengubah *duty cycle* dari sinyal *output* dengan mengubah nilai tegangan referensi [33].



**Gambar 2. 36** Pulsa PWM yang dihasilkan melalui komparator

Pada metode pembangkitan sinyal PWM secara digital, setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Jika

terdapat PWM digital 8bit maka PWM tersebut memiliki rasio  $2^8 = 256$ . Nilai keluaran dari PWM tersebut memiliki 256 variasi dari 0-255 yang mewakili *duty cycle* dari keluaran PWM tersebut.



**Gambar 2. 37** Waktu on dan off pada pulsa PWM

Sinyal PWM pada dasarnya memiliki amplitude dan frekuensi dasar yang tetap namun lebar pulsa yang berbeda-beda. Artinya sinyal PWM memiliki nilai *duty cycle* yang bervariasi antara 0% hingga 100%.  $T_{on}$  merupakan waktu yang menunjukkan saat pulsa *high* sedangkan  $T_{off}$  merupakan waktu yang menunjukkan saat pulsa *low*. Waktu total satu periode yang penjumlahan dari  $T_{on}$  dan  $T_{off}$  selanjutnya akan disebut sebagai  $T_{total}$ . Dengan mengatur lebar pulsa on dan off dalam satu periode gelombang akan didapatkan *duty cycle* yang diinginkan. *Duty cycle* dari PWM dinyatakan secara matematis pada persamaan 2.22. *Duty cycle* 100% berarti sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 12V maka motor akan mendapatkan tegangan 12V. Pada *duty cycle* 50% tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, dan begitupun seterusnya.

$$T_{total} = T_{on} + T_{off} \quad (2.21)$$

$$D = \frac{T_{on}}{T_{total}} \times 100\% \quad (2.22)$$

Perhitungan pengaturan tegangan *output* motor dengan metode PWM cukup sederhana. Dengan menghitung *duty cycle* yang diberikan akan didapatkan tegangan output yang dihasilkan. Hal tersebut secara matematis dirumuskan pada persamaan 2.23 dan 2.24.  $V_{out}$  merupakan tegangan output pada motor yang dikendalikan oleh sinyal PWM. Dengan

menggunakan rumus tersebut maka akan didapatkan tegangan output sesuai dengan sinyal PWM yang dibangkitkan.

$$V_{out} = D \times V_{input} \quad (2.23)$$

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{input} \quad (2.24)$$

*Halaman ini sengaja dikosongkan*

## BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan perancangan sistem implementasi lidar pada robot sepakbola beroda untuk tendangan ke gawang. Perancangan sistem ini meliputi perancangan dari sisi mekanik, elektronik dan program. Pada bagian perancangan mekanik, tugas akhir ini menggunakan mekanik robot yang sudah tersedia dan mengikuti aturan Kontes Robot Sepakbola Indonesia divisi Beroda. Mekanik robot yang sudah yang sudah tersedia akan ditambahkan komponen untuk mendukung tugas akhir ini. Komponen tersebut adalah penempatan sensor LiDAR dan sensor kamera. Mekanisme penendang dan motor penggerak robot merupakan komponen yang sudah tersedia pada robot yang digunakan. Komponen elektronik yang digunakan pada tugas akhir ini juga menggunakan komponen yang sudah tersedia pada robot. Namun komponen elektronik yang sudah tersedia ditambahkan komponen untuk mendukung tugas akhir ini. Komponen tersebut meliputi komunikasi sensor LiDAR ke komputer mini melalui serial menggunakan modul FTDI232 dan kamera ke komputer mini melalui *universal serial bus*, dan pengukuran kekuatan tendangan dari mekanisme penendang. Perancangan program meliputi pelatihan jaringan syaraf tiruan yang digunakan untuk mendeteksi gawang, komunikasi *node* deteksi gawang ke program utama pada komputer mini, komunikasi *node* pembacaan sensor LiDAR ke program utama, dan pengujian sistem keseluruhan.

Sistem secara keseluruhan terdiri menjadi dua bagian, yaitu pendeteksian objek gawang dan pengukuran jarak menggunakan LiDAR. Pendeteksian objek gawang dilakukan oleh program deteksi gawang dengan menggunakan kamera yang dipasang pada bagian depan robot yang searah dengan mekanisme penendang. Kamera tersebut digunakan untuk mengambil citra yang akan dijadikan input jaringan syaraf tiruan yang sebelumnya telah dilatih untuk mendeteksi objek gawang. Output dari jaringan syaraf tiruan ini berupa *bounding box* di sekitar objek yang diprediksi sebagai gawang. Lalu di dalam *bounding box* tersebut akan dilakukan pengolahan citra untuk mendeteksi penjaga gawang. Pengolahan citra tersebut dilakukan dengan mendeteksi warna nomor pemain yang digunakan oleh penjaga gawang, yaitu warna *cyan* atau *magenta*. Untuk mendeteksi keberadaan nomor pemain digunakan metode *threshold* warna dan mencari kontur dari warna yang ditemukan. Apabila terdeteksi keberadaan nomor pemain lawan, maka koordinat

piksel nomor tersebut akan disimpan. Selanjutnya posisi nilai koordinat piksel dari nomor pemain lawan dan koordinat piksel dari *bounding box* di sekitar objek yang diprediksi sebagai gawang akan dikirimkan melalui UDP lokal dari program deteksi gawang ke *node* UDP pada program ROS. Setelah posisi diterima oleh *node* UDP, nilainya akan dikirimkan melalui *ros message* ke program utama pada program ROS. Melalui program utama pada ROS ini selanjutnya *heading* robot akan diarahkan ke tengah *bounding box* tersebut.

Setelah *heading* robot mengarah ke gawang, sensor LiDAR akan membaca hasil pengukuran jarak dari robot ke gawang. Hasil pengukuran sensor LiDAR akan dikirimkan ke *node* sensor lidar secara serial. *Node* sensor lidar kemudian mengirimkan hasil pengukuran melalui *ros message* ke program utama. Program utama mengolah jarak yang dibaca untuk menghasilkan kekuatan tendangan berdasarkan jarak tersebut. Lalu mekanisme penendang akan menghasilkan tendangan dengan kecepatan berdasarkan jarak yang terukur dengan sensor LiDAR. Tendangan yang dihasilkan akan dimodelkan dengan gerak parabola.

### **3.1. Perancangan Mekanik dan Elektronik**

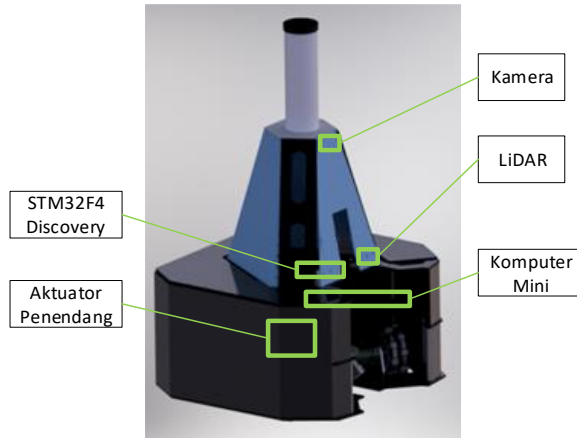
Perancangan mekanik dan elektronik robot merupakan langkah pertama yang dilakukan pada tugas akhir ini. Agar penelitian berjalan dengan baik, maka mekanik dan elektronik robot harus dapat bekerja dengan optimal. Pada tugas akhir ini digunakan robot sepakbola beroda yang sudah tersedia dan dapat berjalan secara fungsional. Robot yang digunakan merupakan properti dari Tim Robot Sepakbola Beroda Institut Teknologi Sepuluh Nopember. Perancangan mekanik termasuk penempatan sensor yang digunakan yaitu sensor kamera dan LiDAR, dan mekanisme penendang. Sementara pada perancangan elektronik meliputi komunikasi untuk mengirim data sensor kamera dan LiDAR ke komputer mini, dan komunikasi antara komputer mini dan mikrokontroler.

#### **3.1.1. Perancangan Mekanik**

Mekanik pada robot sepakbola beroda ini dibuat dengan bahan alumunium, stainless steel, dan akrilik. Untuk bagian seperti *base* robot dibuat dengan bahan plat alumunium ketebalan 5mm dan dipotong menggunakan mesin laser cutting dengan agar menghasilkan potongan plat yang rapih dan bersih. Struktur robot yang menopang dan memerlukan kekuatan lebih dibuat menggunakan plat stainless steel



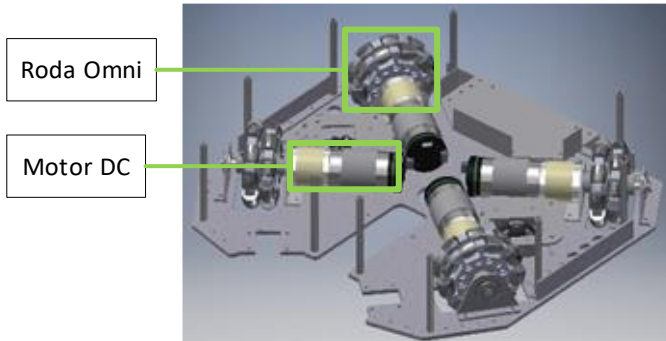
dengan ketebalan hingga 3mm. Sedangkan bagian pendukung yang tidak memerlukan tingkat kekuatan tinggi dibuat menggunakan bahan akrilik.



**Gambar 3. 1** Penempatan komponen yang digunakan pada robot

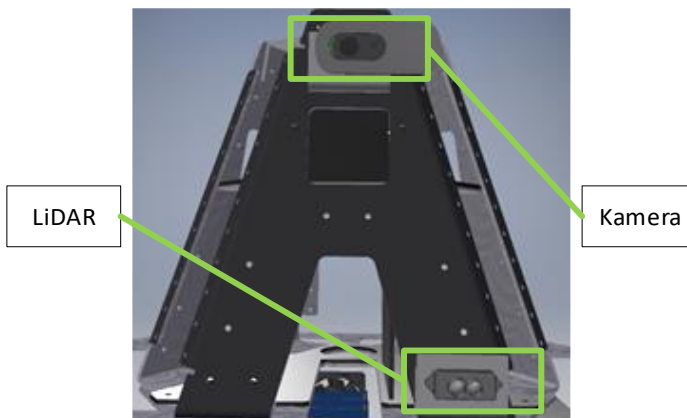
Penggerak robot menggunakan mekanisme dengan empat roda omni dengan mekanisme *holonomic drive*. Mekanisme empat roda digunakan pada robot mengingat bobot robot mencapai 25kg. Oleh karena itu mekanisme dengan menggunakan empat roda dipilih untuk menghasilkan kecepatan yang digunakan agar robot dapat melakukan manuever dengan kecepatan yang maksimal. Konfigurasi roda robot dibuat tidak simetris dengan masing-masing roda omni disusun dengan sudut 70 derajat dan 110 derajat. Hal ini bertujuan untuk mengutamakan kecepatan gerakan robot pada sumbu-y. Roda omni yang digunakan terbuat dari *frame* aluminium dengan memiliki dua lapisan.

Rancangan motor penggerak dalam proyeksi tiga dimensi dapat dilihat pada gambar 3.2. Motor penggerak pada robot menggunakan motor DC planetary gear dengan rasio gearbox 19:1. Gearbox ini digunakan agar dapat mengubah torsi yang dihasilkan motor DC menjadi lebih besar karena total beban robot ketika akan bertanding mencapai 25Kg. Motor penggerak ini dikendalikan menggunakan driver motor jenis BTN7971 yang dapat dibebani arus mencapai 70A. Baterai LiIon 6cell dengan kapasitas 10000mAh digunakan untuk memberikan suplai kepada motor penggerak.



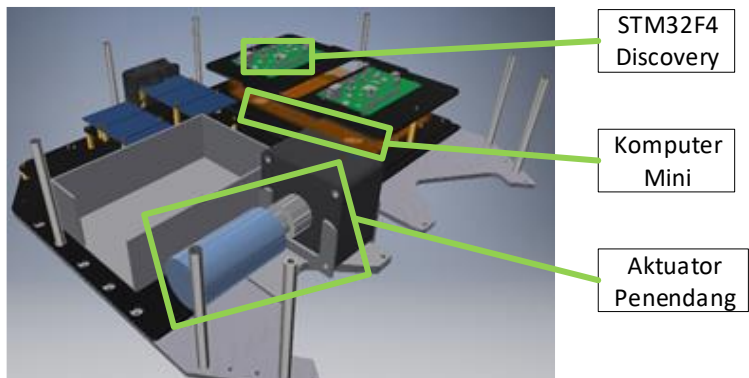
**Gambar 3. 2** Penempatan motor penggerak pada robot

Kamera akan dipasang pada bagian kepala robot dengan posisi menghadap ke depan dan dilindungi oleh penutup akrilik. Pemasangan dengan posisi ini bertujuan agar jarak pandang kamera hanya ke satu arah saja, yaitu bagian depan robot yang searah dengan mekanisme penendang. Sementara itu sensor LiDAR juga dipasang menghadap depan searah dengan mekanisme penendang dan kamera dengan diperkuat oleh frame aluminium. LiDAR dipasang pada ketinggian sekitar 20cm dan kamera pada ketinggian sekitar 60cm dari lantai. Rancangan penempatan sensor kamera dan LiDAR dapat dilihat pada gambar 3.3.



**Gambar 3. 3** Desain penempatan kamera dan LiDAR

Mekanisme penendang dirancang menggunakan motor brushless 2300kv yang dihubungkan dengan gearbox. Motor brushless yang cenderung menghasilkan kecepatan tinggi dengan torsi rendah akan diubah agar memiliki torsi yang cukup untuk menendang bola. Sementara itu kaki penendang akan dirancang dengan jari-jari sekitar 15 cm dari poros motor dan menghasilkan sudut tendangan sebesar sekitar 30 derajat terhadap bola. Kaki penendang dibuat dengan menggunakan bahan alumunium dan memiliki ketebalan sekitar 1.5cm. Bahan alumunium dipilih karena memiliki keandalan yang baik dan bobot yang tidak terlalu berat. Sudut tendangan yang dibentuk oleh gerak putaran dari kaki penendang selanjutnya akan menghasilkan gerak parabola yang digunakan untuk tendangan ke gawang. Motor brushless disuplai oleh baterai LiIon 6cell dengan kapasitas 10000mAh.

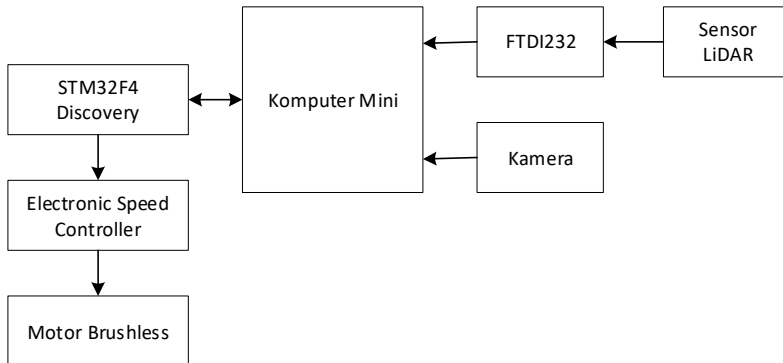


**Gambar 3. 4** Desain penempatan komponen elektronik yang digunakan

### 3.1.2. Perancangan Elektronik

Perancangan elektronik robot terdiri dari komunikasi sensor LiDAR ke komputer mini melalui serial dan kamera ke komputer mini melalui *universal serial bus*, dan pengukuran kekuatan tendangan dari mekanisme penendang. Komponen elektronik dasar seperti *power supply* dan *driver* motor merupakan komponen yang sudah tersedia pada robot yang digunakan untuk tugas akhir ini. Komponen-komponen tersebut tidak akan dibahas lebih lanjut pada bagian perancangan elektronik. Rancangan penempatan komponen elektronik pada robot dapat dilihat di gambar 3.4.

Sedangkan diagram blok yang menunjukkan hubungan antar komponen yang digunakan pada tugas akhir ini dapat dilihat pada gambar 3.5.



**Gambar 3. 5** Diagram blok hubungan antar komponen yang digunakan

Komunikasi LiDAR dengan komputer mini dijumpai oleh modul FTDI232. Data yang dibaca oleh LiDAR akan dikirimkan melalui *port usb mini* yang dimiliki oleh modul FTDI232 ke *port usb 2.0* yang terdapat pada komputer mini. Sensor kamera yang digunakan merupakan jenis Logitech C270 yang memiliki *interface usb*. USB ini kemudian dihubungkan ke *port usb 2.0* pada komputer mini untuk mengakses citra digital yang diambil. Suplai daya pada sensor kamera dan LiDAR didapatkan melalui *port usb* yang dimiliki oleh komputer mini.

Mikrokontroler dihubungkan dengan komputer mini menggunakan modul ethernet untuk mengirimkan data. Modul ethernet yang digunakan merupakan LAN8720. Modul ini dihubungkan menggunakan kabel LAN ke port LAN pada komputer. Data yang dikirimkan dari mikrokontroler ke komputer mini dengan isi berupa informasi odometry, kecepatan dan nilai pembacaan sensor. Sedangkan data yang dikirimkan dari komputer mini ke mikrokontroler memiliki informasi berupa kecepatan untuk aktuator penggerak dan kecepatan untuk mekanisme penandang yang dikirimkan ke electronic speed controller (ESC). Suplai daya untuk komponen elektronik, termasuk komputer mini dan mikrokontroler STM32F4, menggunakan baterai LiIon 6cell dengan kapasitas 10000mAh.

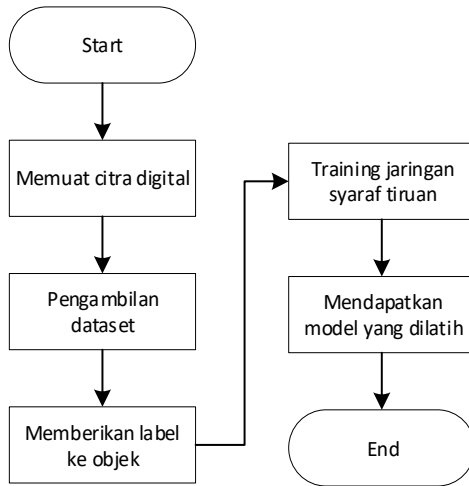
Mekanisme penendang dikendalikan dengan electronic speed controller dan sensor proximity. Electronic speed controller (ESC) akan menerima sinyal pulsa yang dikirim oleh mikrokontroler dan diubah sebagai kecepatan tendang. Sensor proximity berperan untuk menghitung jumlah putaran kaki penendang dan waktu yang diperlukan untuk menempuh satu putaran. Apabila telah melakukan tendangan dalam beberapa putaran yang diinginkan, maka ESC akan mengirimkan sinyal untuk *breaking* pada motor brushless.

### **3.2. Pendeteksian Objek Gawang**

Sistem pendeteksian objek gawang dibuat dengan tujuan agar robot dapat mengenali gawang yang terdapat pada lapangan dengan menggunakan sensor eksternal. Sensor yang digunakan merupakan sensor kamera. Kamera tersebut dipasang pada bagian depan robot yang sejajar dengan arah mekanisme penendang. Kamera yang digunakan adalah USB webcam Logitech tipe c270. Pendeteksian objek gawang secara garis besar dibagi menjadi dua, yaitu pelatihan jaringan syaraf tiruan untuk mendeteksi gawang dan implementasi jaringan syaraf tiruan untuk mendeteksi gawang.

#### **3.2.1. Pelatihan Jaringan Syaraf Tiruan**

Pelatihan jaringan syaraf tiruan merupakan langkah untuk mendapatkan model yang memiliki weight dan bias yang telah dioptimalkan untuk melakukan deteksi terhadap objek gawang. Setelah mendapatkan model yang optimal, model tersebut akan disimpan dan dimuat pada program untuk mendeteksi gawang pada robot. Untuk melakukan pelatihan jaringan syaraf tiruan diperlukan komputer dengan performa yang tinggi dan memiliki akses ke GPU untuk mempercepat proses pelatihan. Oleh karena itu pada tugas akhir ini penulis menggunakan Google Colab, platform online yang disediakan oleh Google dan dapat digunakan untuk melakukan proses pelatihan jaringan syaraf tiruan. Diagram alir pelatihan jaringan syaraf tiruan untuk mendeteksi gawang dapat dilihat pada gambar 3.7. Pelatihan jaringan syaraf tiruan pada tugas akhir ini terdiri dari langkah-langkah sebagai berikut:



**Gambar 3. 6** Diagram alir pelatihan jaringan syaraf tiruan

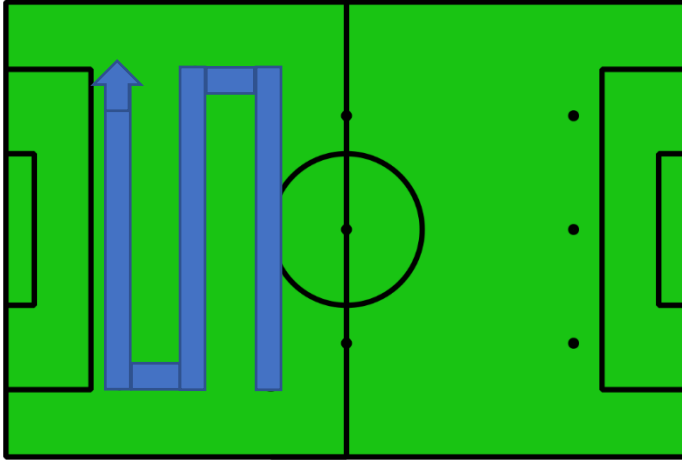
### 3.2.1.1. *Pengolahan Citra Digital*

Sebelum melakukan deteksi terhadap objek gawang, dilakukan proses pengolahan citra digital yang akan dijadikan input sistem. Citra digital tersebut didapatkan oleh sensor kamera dan diolah menggunakan library openCV. Citra digital yang dijadikan input diubah ukurannya menjadi 640 piksel x 480 piksel dengan menggunakan colorspace BGR.

### 3.2.1.2. *Pengambilan Dataset*

Pengambilan dataset dilakukan setelah pengolahan citra digital. Tahap ini dilakukan dengan merekam video dengan target utama adalah objek gawang dimulai dari jarak 2meter hingga 6meter. Jarak ini disesuaikan dengan peraturan dari kontes robot sepakbola Indonesia divisi beroda dimana robot tidak boleh memasuki kotak yang berjarak 1.8 meter di depan gawang karena akan dianggap melakukan illegal attack dan robot tidak boleh menendang dari daerah bertahan sendiri yaitu kurang dari setengah lapangan. Video akan direkam dengan menggunakan tripod yang dipindahkan secara manual sesuai dengan jalur yang ditunjukkan pada gambar 3.5. Video yang dihasilkan memiliki resolusi 640 piksel x 480 piksel. Frame dari video tersebut disimpan ke dalam format jpg setiap 500ms untuk selanjutnya diberikan label.

Pengambilan video dilakukan sebanyak dua kali, yaitu saat siang dan malam dengan total menghasilkan sekitar 3000 gambar.



**Gambar 3. 7** Jalur pengambilan gambar gawang untuk dijadikan dataset

### 3.2.1.3. *Pelabelan Gawang*

Proses melabeli objek gawang dilakukan menggunakan program. Setelah membuka program tersebut, pengguna akan memilih direktori gambar input dan direktori output. Melalui program tersebut pengguna secara manual dapat memberikan bentuk persegi panjang di sekitar objek yang diinginkan, dalam hal ini objek gawang. Selanjutnya diberikan nama terhadap label dari objek tersebut. Proses ini disebut proses *labeling*. Labeling tidak hanya dilakukan pada citra yang menampilkan objek gawang secara penuh, tetapi juga pada citra yang menampilkan objek gawang secara sebagian.

Proses *labeling* dilanjutkan untuk seluruh gambar yang dihasilkan pada tahap pengambilan dataset. Setelah selesai melakukan proses labelling, pada direktori output akan dihasilkan file dengan ekstensi .xml untuk setiap citra yang diberi label. File xml tersebut dapat dilihat pada gambar 3.10.



**Gambar 3. 8** Gambar gawang sebelum dan sesudah diberi label

```

1 <annotation>
2   <folder>gawang</folder>
3   <filename>14.jpg</filename>
4   <path>E:\maul_backup\KULIAH\SEMESTER 8\TA\garap\2 class new\gawang\14.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>640</width>
10    <height>480</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>gawang</name>
16    <pose>Unspecified</pose>
17    <truncated>1</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>1</xmin>
21      <ymin>107</ymin>
22      <xmax>123</xmax>
23      <ymin>480</ymin>
24    </bndbox>
25  </object>
26 </annotation>

```

**Gambar 3. 9** Keluaran file xml yang dihasilkan

Keterangan untuk file xml yang dihasilkan adalah sebagai berikut:

- Baris ke-2 menunjukkan nama folder file gambar yang dijadikan dataset.
- Baris ke-3 menunjukkan nama file gambar yang dijadikan dataset.
- Baris ke-4 menunjukkan direktori file gambar yang dijadikan dataset.
- Baris ke-8 hingga ke-12 menunjukkan dimensi gambar yang dijadikan dataset dan channel yang dimiliki.
- Baris ke-19 hingga ke-24 menunjukkan koordinat x dan y dari bounding box dalam file gambar yang dijadikan dataset.



#### **3.2.1.4. Proses Training Jaringan Syaraf Tiruan**

Proses training jaringan syaraf tiruan pada tugas akhir ini menggunakan Tensorflow dengan python sebagai backend. Model yang digunakan dalam pelatihan jaringan syaraf tiruan adalah SSD Mobilenet V2. Pelatihan jaringan syaraf tiruan dilakukan secara online menggunakan platform Google Colab. Sebelum melakukan pelatihan jaringan syaraf tiruan, dilakukan konfigurasi terhadap parameter-parameter yang akan menentukan proses pelatihan jaringan syaraf tiruan. Parameter-parameter tersebut antara lain:

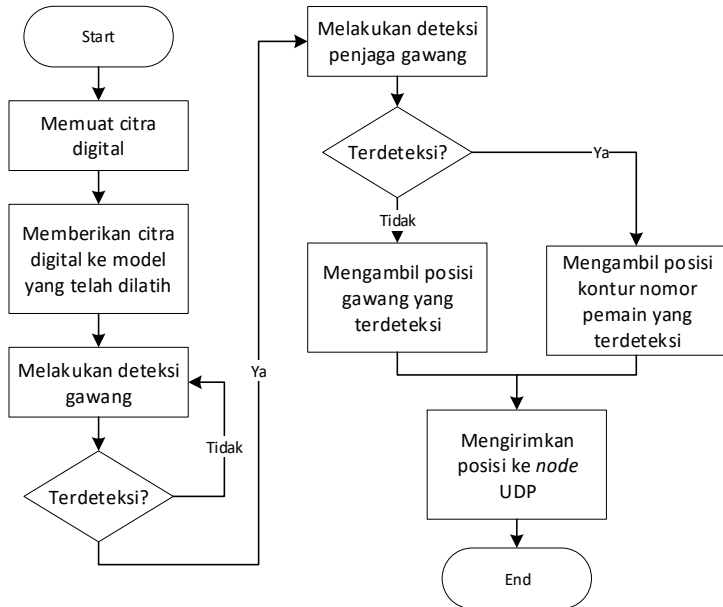
- Jumlah class dari objek yang ingin di deteksi. Pada tugas akhir ini akan dideteksi 1 class yaitu gawang.
- Direktori penyimpanan dataset yang akan digunakan sebagai training data dan test data.
- Direktori penyimpanan labelmap yang digunakan pada saat training.

#### **3.2.2. Implementasi Jaringan Syaraf Tiruan**

Pelatihan jaringan syaraf tiruan akan menghasilkan model jaringan syaraf tiruan dengan weight dan bias yang telah dioptimalkan untuk diterapkan dalam mendeteksi gawang. Model jaringan syaraf tiruan yang dihasilkan memiliki ekstensi *protobuf*. Model tersebut dimuat pada robot dengan menggunakan *library* OpenCV dengan Bahasa C++. Model jaringan syaraf tiruan yang telah dilatih ini akan diimplementasikan pada program deteksi gawang. Selanjutnya citra digital yang didapatkan secara *realtime* dari kamera akan dijadikan input jaringan syaraf tiruan yang telah dilatih.

Dengan menjadikan citra digital sebagai input model jaringan syaraf tiruan yang sebelumnya telah dilatih, komputer dapat memprediksi apakah ada objek gawang dalam satu *frame*. Dalam *frame* tersebut akan timbul *bounding box* yang berada pada sekitar objek yang diprediksi sebagai objek gawang. Setiap *bounding box* yang dihasilkan memiliki nilai *degree of confidence*. Nilai ini mewakili besar keyakinan komputer terhadap hasil prediksi objek yang dideteksi, semakin besar nilainya maka komputer semakin yakin terhadap objek yang dideteksi. Selain memiliki nilai *degree of confidence*, *bounding box* yang dihasilkan juga memiliki koordinat posisi dan ukuran dalam piksel. Kemudian di dalam *bounding box* yang dihasilkan dapat dideteksi keberadaan robot penjaga gawang. Robot penjaga gawang dapat dideteksi dengan

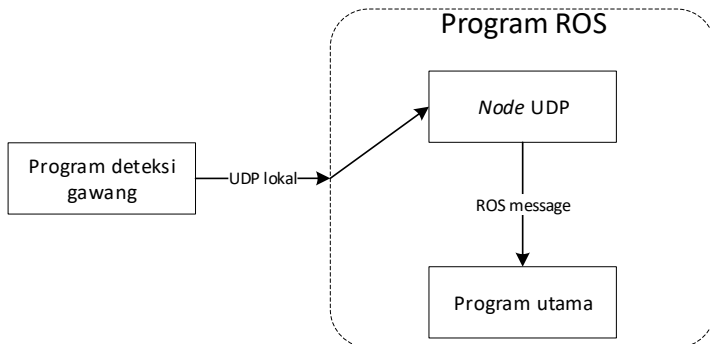
menggunakan metode ambang batas (*thresholding*) dan mencari kontur (*blob detection*) dari nomor robot yang memiliki warna *cyan* atau *magenta*. Metode ambang batas dan mencari kontur dapat dilakukan menggunakan pengolahan citra dengan *library* OpenCV.



**Gambar 3. 10** Diagram alir program deteksi gawang

Dalam kondisi tidak ada robot penjaga gawang, koordinat posisi dan ukuran *bounding box* yang pertama kali mendeteksi gawang akan disimpan dan diolah sehingga akan menghasilkan nilai yang selanjutnya dikirimkan melalui UDP lokal. Namun ketika kondisi ada robot penjaga gawang yang dideteksi dalam *bounding box*, posisi piksel dari kontur nomor pemain akan disimpan dan nilainya juga akan dikirimkan melalui UDP lokal. Nilai-nilai tersebut dikirim dari program yang memuat jaringan syaraf tiruan ke *node* UDP pada program ROS melalui UDP lokal. *Node* UDP pada program ROS ini selanjutnya akan mem-*publish* pesan yang berisi nilai tersebut ke program utama pada robot. Diagram alir yang menunjukkan implementasi jaringan syaraf tiruan yang sudah

dilatih untuk mendeteksi gawang dapat dilihat pada gambar 3.11. Sedangkan diagram blok hubungan program deteksi gawang dan program utama pada ROS dapat dilihat pada gambar 3.11.



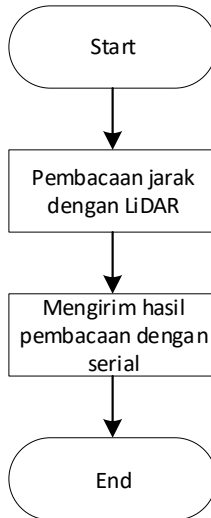
**Gambar 3. 11** Diagram blok pengiriman posisi gawang ke program utama

### 3.3. Pengukuran Jarak dengan LiDAR

Sistem pengukuran jarak dilakukan dengan menggunakan sensor LiDAR. Sensor ini digunakan dengan tujuan robot dapat mengerti jarak terhadap gawang. Sensor LiDAR yang digunakan merupakan jenis TF Mini LiDAR yang diproduksi oleh DF Robot. Sensor tersebut dipasang pada bagian depan robot dengan ketinggian 20 dari lantai dan sejajar dengan arah penandang. Diagram alir pengukuran jarak dengan LiDAR dapat dilihat pada gambar 3.13.

#### 3.4.1. Pembacaan data sensor LiDAR

Data yang dihasilkan oleh sensor LiDAR dapat diakses menggunakan komunikasi serial. Sensor LiDAR mengirim 9byte data dimana byte pertama dan byte kedua merupakan header. Byte ketiga dan keempat merupakan nilai dari pembacaan jarak dalam satuan centimeter. Byte ke-5 dan ke-6 merupakan nilai dari kekuatan sinyal LiDAR. Komunikasi serial berjalan dengan baudrate sebesar 115200. Informasi yang dikirim pada tiap byte dan penjelasannya dapat dilihat pada gambar 3.14.



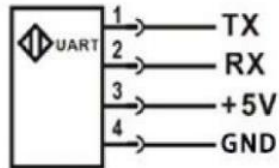
**Gambar 3. 12** Diagram alir pengukuran jarak dengan LiDAR

Byte1-2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9
0x59 59	Dist_L	Dist_H	Strength_L	Strength_H	Reserved	Raw.Qual	Checksum_L
Data encoding interpretation							
<b>Byte1</b>	0x59, frame header, all frames are the same						
<b>Byte2</b>	0x59, frame header, all frames are the same						
<b>Byte3</b>	Dist_L distance value is a low 8-bit. Note: The distance value is a hexadecimal value, for example, Distance 1,000cm = 03 E8 (HEX)						
<b>Byte4</b>	Dist_H distance value is a high 8-bit.						
<b>Byte5</b>	Strength_L is a low 8-bit.						
<b>Byte6</b>	Strength_H is a high 8-bit.						
<b>Byte7</b>	Reserved bytes.						
<b>Byte8</b>	Original signal quality degree.						

**Gambar 3. 13** Data serial yang dikirimkan oleh LiDAR

### 3.4.2. Pengiriman data ke komputer mini

Proses pengiriman data serial dari sensor LiDAR ke komputer mini dijumpai oleh modul FTDI232. Pada sensor LiDAR memiliki 4 pinout yaitu TX, RX, 5V dan GND. Pinout diagram untuk sensor lidar dapat dilihat pada gambar 3.14.



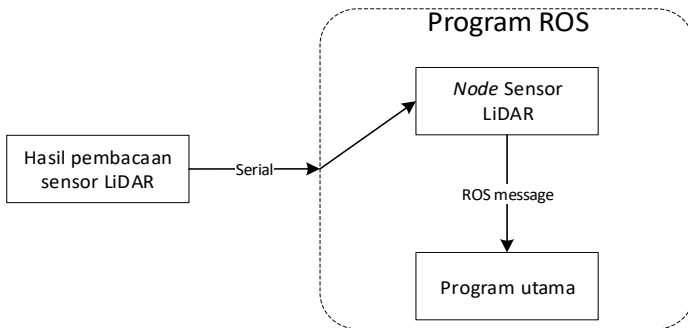
**Gambar 3. 14** Pinout diagram sensor LiDAR

Modul FTDI232 memiliki 6 pinout yaitu DTR, RX, TX, VCC, CTS, dan GND. Pada tugas akhir ini hanya empat pin yang digunakan dan dihubungkan dengan sensor LiDAR. Konektor usb mini pada modul FTDI232 akan dihubungkan dengan port usb pada komputer mini untuk mengirim data yang dilakukan secara serial. Pinout modul FTDI232 dapat dilihat pada gambar 3.15.



**Gambar 3. 15** Pinout diagram modul FTDI232

*Node* serial yang diprogram untuk membaca data hasil pengukuran sensor LiDAR melalui modul FTDI232 akan mem-*publish* pesan ke program utama. Kemudian program utama robot akan men-*subscribe* pesan yang dikirimkan oleh *node* serial. Pembacaan nilai sensor LiDAR akan digunakan ketika heading robot sudah mengarah ke gawang. Nilai sensor tersebut akan berperan untuk menentukan kekuatan tendangan ke gawang. Pembacaan nilai sensor yang berbeda akan menghasilkan kekuatan tendangan yang berbeda. Diagram blok yang menunjukkan pengiriman hasil pembacaan sensor LiDAR ke program utama ditunjukkan pada gambar 3.16.



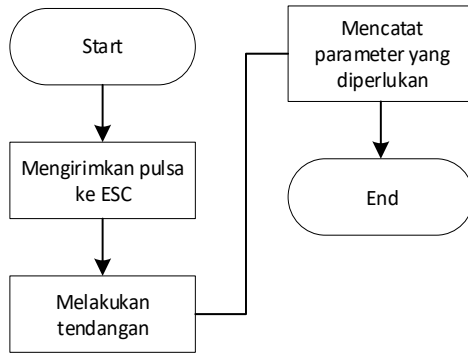
**Gambar 3. 16** Diagram blok pengiriman sensor LiDAR ke program utama

### 3.4. Implementasi Tendangan ke Gawang

Tendangan ke gawang dilakukan setelah kamera mengidentifikasi objek gawang. Sebuah bounding box akan diberikan di sekitar objek yang diprediksi sebagai gawang. Heading robot akan diarahkan menuju ke objek yang diprediksi sebagai gawang. Setelah heading sudah dipastikan ke arah objek, selanjutnya lidar akan melakukan pengukuran jarak dan mekanisme penendang akan menendang bola berdasarkan jarak yang dibaca oleh LiDAR.

#### 3.4.1. Memodelkan gerak parabola sebagai lintasan tendangan

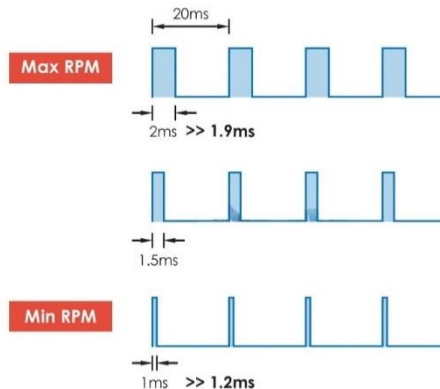
Pada tugas akhir ini akan digunakan gerak parabola untuk model tendangan ke arah gawang. Gerak parabola digunakan karena penjaga gawang akan lebih sulit untuk mengantisipasi tendangan lambung. Tendangan lambung memiliki komponen gerak pada sumbu-z. Lintasan tendangan pada sumbu-z akan lebih susah diprediksi apabila tidak menggunakan kamera yang mampu mengukur kedalaman. Salah satu jenis kamera yang dapat mengukur kedalaman dari objek adalah kamera *stereovision*. Kamera ini memiliki dua buah sensor yang digunakan untuk menangkap citra dan kemudian dilakukan pengolahan agar dapat diketahui kedalaman dari objek yang ditangkap. Gerak parabola akan terjadi karena mekanisme penendang yang dimiliki pada robot memungkinkan untuk memberikan kecepatan awal dengan sudut tertentu pada bola. Sudut ini memiliki nilai tetap dan selanjutnya akan dianggap sebagai konstanta.



**Gambar 3. 17** Diagram alir penentuan kekuatan tendangan

### 3.4.2. Menentukan kekuatan tendangan yang digunakan

Pembacaan sensor LiDAR dari robot ke gawang akan menghasilkan jarak yang berbeda untuk setiap posisi di lapangan. Kekuatan tendangan akan disesuaikan dengan jarak robot ke gawang. Hal ini dilakukan agar bola tidak melaju terlalu cepat sehingga menabrak tiang atas gawang atau terlalu pelan sehingga tidak dapat mencapai gawang. Oleh karena itu pada langkah ini akan dilakukan pengujian kekuatan tendangan.



**Gambar 3. 18** Sinyal PWM yang diberikan pada ESC

Kekuatan tendangan pada robot akan diwakilkan oleh nilai sinyal PWM yang dikirimkan ke ESC. Nilai PWM tersebut memiliki periode

sebesar 20ms. Sinyal tersebut dihasilkan oleh timer 11 yang dimiliki oleh mikrokontroler STM32F407 discovery. Untuk menghasilkan gerakan putaran maju, mikrokontroler harus menghasilkan pulsa dengan kisaran  $T_{on}$  selama 1.5ms sampai 2ms. Untuk menghasilkan putaran mundur,  $T_{pn}$  yang dihasilkan selama kisaran 1ms sampai 1.5ms. Sedangkan dengan mengirim pulsa dengan  $T_{on}$  1.5ms akan menyebabkan pengereman pada motor. Karena pada tugas akhir ini diperlukan gerakan putaran maju untuk menendang, maka PWM yang digunakan memiliki  $T_{on}$  mulai dari 1.5ms hingga 2ms. Pada program mikrokontroler, nilai-nilai  $T_{on}$  tersebut ditulis mulai dari range 1500 hingga 2000. Hal ini dilakukan karena menyesuaikan kecepatan *clock* yang ada pada timer 11 yaitu sebesar 168MHz.

Pada setiap nilai PWM yang dikirimkan akan dicatat parameter-parameter yang berkaitan dengan gerak parabola seperti periode yang dihasilkan kaki penendang untuk melakukan satu putaran. Periode ini akan diubah menjadi frekuensi dan secara matematis dapat digunakan untuk menghitung kecepatan sudut dan kecepatan linear yang dihasilkan karena nilai sudut elevasi dan panjang lengan penendang sudah menjadi konstanta. Selain itu waktu dan jarak tempuh bola hingga mencapai jarak maksimal juga akan dicatat. Tinggi maksimal bola untuk setiap nilai PWM juga dapat dihitung secara matematis. Diagram alir penentuan kekuatan tendangan yang diperlukan dapat dilihat pada gambar 3.17.

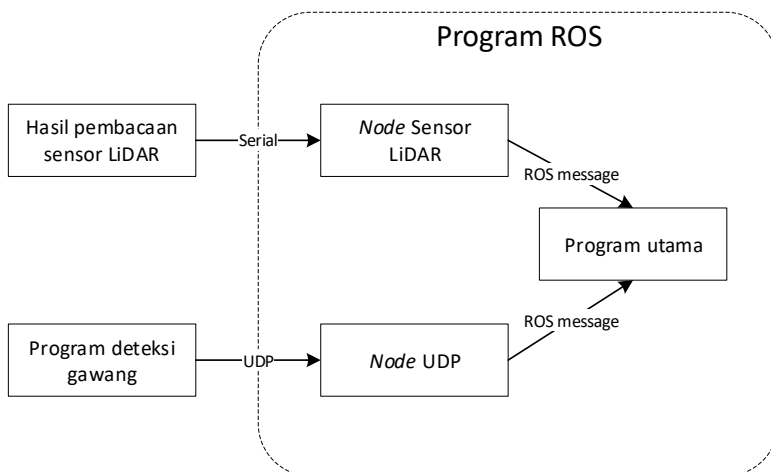
### **3.4.3. Merangkai keseluruhan sistem**

Pengujian sistem secara keseluruhan pada tugas akhir ini dilakukan dengan menjalankan dua program secara bersamaan. Program pertama ditulis dengan bahasa C++ digunakan untuk pengolahan citra dan model jaringan saraf tiruan yang sebelumnya telah dilatih. Program kedua ditulis dengan bahasa C++ menggunakan library ROS untuk menjalankan fungsi-fungsi dasar pada robot agar robot dapat bergerak. Untuk selanjutnya program pertama akan disebut program deteksi gawang dan program kedua akan disebut dengan program ROS. Program deteksi gawang dihubungkan ke program ROS menggunakan protocol UDP dalam jaringan localhost. Diagram blok yang menunjukkan hubungan antar program pada sistem keseluruhan robot ditunjukkan pada gambar 3.19.

Program deteksi gawang akan mengolah input citra digital RGB dan dijadikan masukan model jaringan saraf tiruan yang sebelumnya telah dilatih untuk mendeteksi objek gawang. *Bounding box* akan muncul di sekitar objek yang diprediksi sebagai gawang. Saat belum terdeteksi



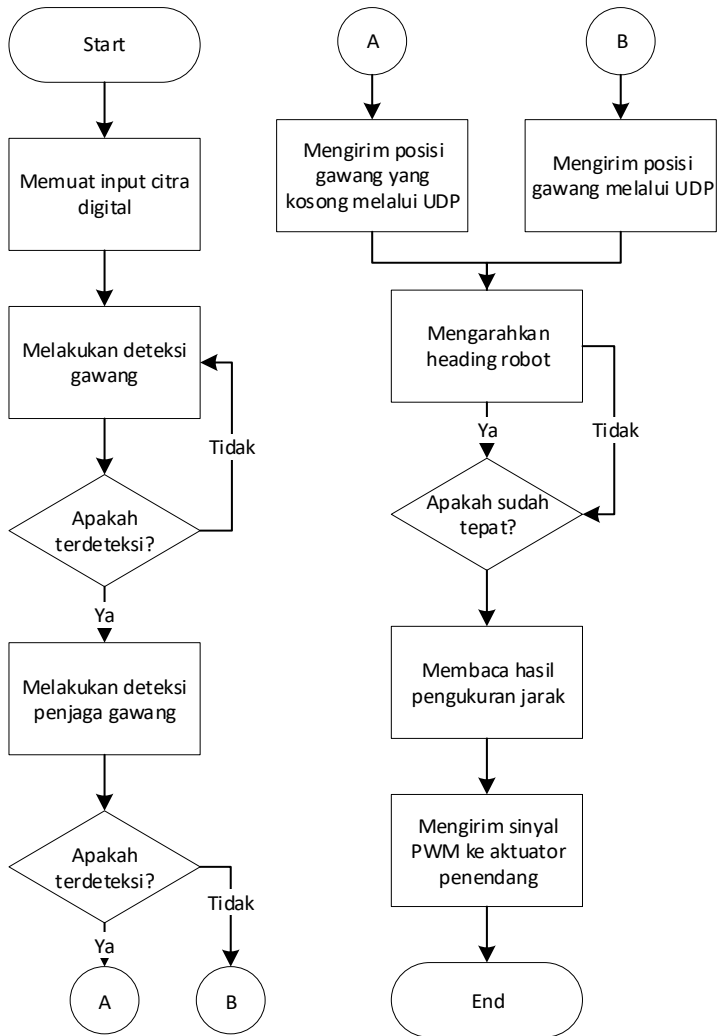
objek yang diperkirakan sebagai gawang, maka program deteksi gawang akan mengirimkan nilai awal berupa 0 ke melalui UDP local ke *node* UDP yang berjalan pada program ROS. *Node* UDP selanjutnya akan *publish* nilai awal ke program utama dan program utama akan *subscribe* nilai awal yang dikirimkan *node* UDP. Apabila nilai awal yang dikirim bernilai 0, maka heading robot akan berputar dengan kecepatan sudut untuk mengarahkan heading sampai menemukan objek yang diperkirakan sebagai gawang. Ketika sudah ditemukan objek yang diprediksi sebagai gawang, nilai awal akan berubah dan memiliki nilai yang bukan 0. Nilai tersebut berubah menjadi jarak antara tengah *bounding box* yang pertama kali dideteksi dengan heading robot. Nilai ini kemudian akan semakin kecil seiring dengan heading robot mengarah ke gawang. Namun apabila dalam *bounding box* yang pertama kali dideteksi terdapat penjaga gawang dengan nomor pemain berwarna *cyan* atau *magenta*, maka heading robot akan bergerak menjauhi posisi dari kontur nomor pemain yang terdeteksi. Dengan mengarahkan *heading* robot menjauhi kontur nomor pemain maka heading robot akan menuju ke bagian gawang yang lepas dari jangkauan penjaga gawang.



**Gambar 3. 19** Diagram blok rangkaian program keseluruhan sistem

Pembacaan nilai sensor LiDAR akan digunakan saat heading robot telah mengarah ke bagian gawang yang kosong. Hal ini disebabkan cara kerja LiDAR adalah dengan menghitung waktu *time of flight* ke objek

yang menjadi target untuk mendapatkan jarak. Apabila terdapat halangan menuju target, maka hasil pembacaan jarak akan berbeda dengan yang sebenarnya. Pesan berisi nilai sensor yang dikirim oleh *node* serial akan dipanggil di program utama. Selanjutnya pada program utama akan ditentukan kekuatan tendangan berdasarkan jarak yang dibaca dari sensor LiDAR. Untuk mengetahui kekuatan tendangan dilakukan saat pengujian kekuatan tendangan. Kekuatan tendangan akan berbeda untuk setiap pembacaan jarak yang didapatkan. Pengujian keseluruhan sistem pada robot ditunjukkan oleh diagram alir pada gambar 3.20.



**Gambar 3. 20** Diagram alir pengujian keseluruhan sistem

*Halaman ini sengaja dikosongkan*

## **BAB IV**

### **PENGUJIAN DAN ANALISA**

Bab ini membahas tentang pengujian dan analisa terhadap sistem yang telah dirancang dan dibuat. Bab ini memiliki tujuan untuk mengetahui apakah sistem dapat menyelesaikan masalah-masalah yang telah dirumuskan pada tugas akhir. Masalah-masalah tersebut meliputi pendeteksian gawang, pengukuran jarak, pengujian kekuatan tendangan, pengujian sensor LiDAR dan pengujian tendangan ke gawang. Pengujian sistem dilakukan dengan tetap mengikuti batasan-batasan masalah yang telah dirumuskan pada tugas akhir ini.

#### **4.1. Pengujian Deteksi Gawang**

Pengujian deteksi gawang dilakukan dengan cara menempatkan kamera pada titik-titik lapangan yang koordinatnya sudah diketahui. Titik-titik ini disebar pada jarak yang tidak lebih dari setengah bagian lapangan, yaitu sejauh 6meter dari garis gawang. Hal ini dilakukan untuk menyesuaikan aturan dari kontes robot sepakbola Indonesia divisi beroda. Gambar titik-titik pengujian dapat dilihat pada gambar 4.2 dan penempatan kamera pada lapangan dapat dilihat pada gambar 4.1.



**Gambar 4. 1** Penempatan kamera untuk pengujian deteksi gawang

Pengujian deteksi gawang dilakukan untuk mengetahui tingkat akurasi jaringan syaraf tiruan dalam membedakan apakah suatu objek termasuk gawang atau bukan. Pengujian dilakukan dalam ruangan tertutup dengan pencahayaan dari dalam dan tanpa ada interferensi cahaya dari luar ruangan. Tingkat pencahayaan pada saat pengujian dibuat 3

variasi pencahayaan pada ruangan. Pengujian dilakukan pada ruangan workshop di Gedung Pusat Robotika. Saat malam hari dilakukan pengujian dengan dua lampu dan satu lampu ditunjukkan pada kolom dengan nilai intensitas 48 lux dan 11 lux. Pengujian saat pagi hari ditunjukkan pada kolom dengan nilai intensitas sebesar 52 lux. Pada satu titik pengujian dilakukan dengan memberikan 100 gambar yang telah dipilih dan dijadikan input jaringan syaraf tiruan yang telah dilatih. Jaringan syaraf tiruan dikatakan berhasil mendeteksi apabila muncul *bounding box* di sekitar gawang. Apabila *bounding box* muncul di sekitar objek yang bukan gawang atau muncul saat tidak ada gawang dalam frame, maka akan dianggap salah. Hasil pengujian akurasi gawang dapat dilihat pada tabel 4.1.

**Tabel 4. 1** Hasil pengujian akurasi model jaringan syaraf tiruan

Titik	Akurasi (%)		
	Intensitas = 11 lux	Intensitas = 48 lux	Intensitas = 52 lux
A1	96	95	92
A2	84	96	83
A3	91	98	86
A4	92	97	82
A5	97	97	86
B1	96	99	89
B2	93	99	99
B3	96	88	99
B4	87	91	89
B5	99	97	94
C1	98	96	85
C2	98	92	95
C3	100	100	81
C4	100	100	99
C5	95	100	99
D1	92	99	99
D2	82	96	98
D4	94	93	99
D4	98	97	99
D5	96	85	96
Rata-rata	94.2	95.75	92.45

**Tabel 4. 2** Hasil pengujian *degree of confidence* pada titik-titik

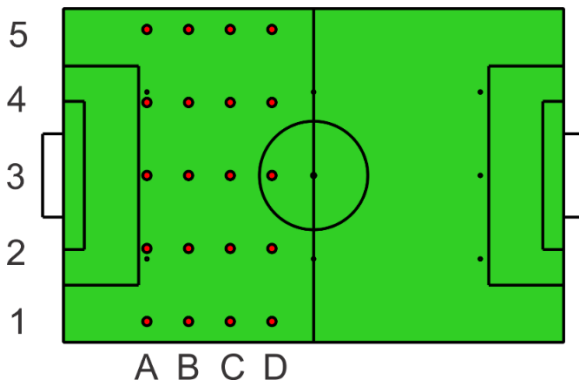
Titik	Intensitas = 11 lux	Intensitas = 48 lux	Intensitas = 52 lux
	<i>Degree of Confidence</i>	<i>Degree of Confidence</i>	<i>Degree of Confidence</i>
A1	0.999943	0.915109	0.830522
A2	0.999959	0.999881	0.999729
A3	0.999797	0.999875	0.999816
A4	0.999715	0.999705	0.999279
A5	0.999998	0.962809	0.998644
B1	0.996079	0.999674	0.999980
B2	0.999891	0.999984	0.999962
B3	0.999350	0.999729	0.999318
B4	0.999974	0.999987	0.998115
B5	0.979781	0.999947	0.972207
C1	0.999958	0.998500	0.876727
C2	0.999984	0.977600	0.865907
C3	0.999930	0.994400	0.984711
C4	0.999995	0.999951	0.998575
C5	0.999977	0.998537	0.999819
D1	0.999429	0.999999	0.851239
D2	0.999995	0.999999	0.975711
D3	0.999999	0.999999	0.999593
D4	0.999937	0.999889	0.991873
D5	0.999998	0.992836	0.829908
Rata-rata	0.998684	0.991921	0.958582

Dari pengujian yang ditunjukkan pada tabel 4.1, dapat dilihat bahwa jaringan syaraf tiruan dapat mendeteksi gawang dalam pengujian tiga kondisi pencahayaan. Saat pengujian dengan pencahayaan sebesar 11 lux, jaringan syaraf tiruan dapat mendeteksi objek gawang dengan akurasi paling besar 100% pada titik C. Saat pencahayaan diubah menjadi 48 lux, akurasi paling besar didapatkan pada titik C dengan nilai sebesar 100%. Saat pencahayaan dengan nilai 52 lux, jaringan syaraf tiruan memiliki nilai akurasi paling besar 99%.

Pengujian nilai batas *degree of confidence* ditunjukkan pada tabel 4.2. Pengujian saat kondisi pencahayaan sebesar 48 lux menghasilkan

nilai *degree of confidence* dengan rata-rata di atas 99.1921%. Ketika kondisi pencahayaan diubah menjadi 11 lux, nilai *degree of confidence* yang dihasilkan bertambah baik sampai rata-rata di atas 99.8684%. Saat pengujian pagi hari dengan pencahayaan 52 lux, nilai *degree of confidence* dimana jaringan syaraf tiruan dapat mendeteksi objek gawang adalah rata-rata lebih dari 95.8582%. Rata-rata nilai *degree of confidence* paling tinggi didapatkan pada pengujian dengan tingkat intensitas cahaya sebesar 11 lux yaitu saat pengujian dilakukan malam hari.

Hasil pengujian saat pagi hari menghasilkan nilai *degree of confidence* dan akurasi lebih rendah dibandingkan malam hari karena adanya interferensi cahaya dari luar. Interferensi cahaya dihasilkan melalui struktur ruangan yang dibentuk oleh banyak jendela kaca transparan. Interferensi cahaya ini juga menyebabkan pada saat pengujian ada objek selain gawang yang justru dideteksi sebagai gawang. Hal ini menghasilkan lebih dari satu *bounding box* pada *frame*.

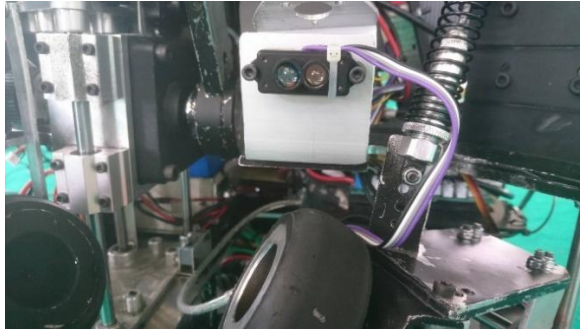


**Gambar 4. 2** Titik pengujian deteksi gawang pada lapangan

#### **4.2. Pengujian Pembacaan Sensor LiDAR**

Pengujian pembacaan sensor LiDAR dilakukan dengan membandingkan nilai yang didapatkan dari sensor dengan nilai sebenarnya. Nilai yang didapatkan dari sensor LiDAR dilihat melalui *command line terminal* pada komputer yang digunakan pada robot, yaitu Intel NUC. Nilai sebenarnya sebagai pembanding didapatkan dengan menggunakan meteran. Penempatan sensor LiDAR pada robot dapat dilihat pada gambar 4.3.





**Gambar 4. 3** Penempatan sensor LiDAR pada robot

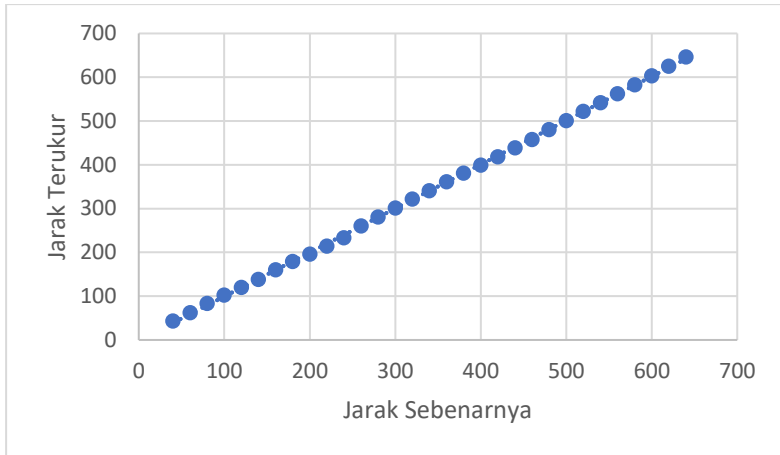
Pengujian ini dilakukan untuk mengetahui akurasi pembacaan sensor LiDAR. Sensor diletakkan pada bagian depan robot dan akan mengukur jarak dari robot hingga ke dinding belakang gawang. Pengujian dilakukan di ruangan workshop Gedung pusat robotika pada siang hari dengan pencahayaan sebesar 52 lux. Ilustrasi pengukuran jarak dengan menggunakan sensor LiDAR dapat dilihat pada gambar 4.4. Nilai yang terukur menggunakan sensor LiDAR akan dibandingkan dengan nilai sebenarnya yang dibaca pada meteran. Perbandingan pengujian sensor LiDAR dengan jarak yang sebenarnya dapat dilihat pada tabel 4.3.



**Gambar 4. 4** Pengukuran jarak pada robot menggunakan LiDAR

**Tabel 4. 3** Hasil pengukuran sensor LiDAR ke gawang

Jarak Asli (cm)	Jarak Terukur (cm)	Selisih Jarak	Error
		(cm)	RMSE
40	42	2	2.79
60	63	3	2.74
80	83	3	2.72
100	102	2	2.73
120	122	2	2.67
140	138	2	2.65
160	160	0	2.65
180	179	1	2.62
200	196	4	3.62
220	214	6	2.61
240	233	7	2.51
260	260	0	2.27
280	281	1	1.89
300	301	1	1.89
320	322	2	1.88
340	341	1	1.87
360	361	1	1.84
380	381	1	1.83
400	399	1	1.82
420	418	2	1.81
440	439	1	1.80
460	458	2	1.76
480	480	0	1.75
500	501	1	1.72
520	522	2	1.72
540	52	2	1.71
560	562	2	1.67
580	583	3	1.63
600	603	3	1.50
620	625	5	1.40
640	646	6	1.07
Rata-Rata		2.225	2.104



**Gambar 4. 5** Grafik perbandingan jarak hasil pembacaan LiDAR dan sebenarnya

Berdasarkan hasil pengujian yang ditunjukkan pada tabel 4.2., hasil pembacaan sensor LiDAR menunjukkan error yang kecil. Error merupakan selisih antara jarak yang sebenarnya dan jarak yang terukur dari sensor LiDAR. Nilai error berada pada rentang paling kecil 0 cm dan paling besar 7 cm. Rata-rata nilai error jarak yang dialami pada saat pengukuran dengan menggunakan sensor LiDAR adalah sebesar 2.225cm. Sementara itu rata-rata nilai error RMSE yang didapatkan saat pengukuran adalah sebesar 2.104cm. Hal ini disebabkan oleh beberapa hal seperti permukaan gawang memiliki tingkat pemantulan yang baik dan area permukaan gawang yang luas. Pengujian yang dilakukan di dalam ruangan dengan minimnya interferensi cahaya matahari juga menyebabkan kekuatan pulsa sinyal LiDAR masih kuat sehingga hasil pembacaan menghasilkan error yang kecil.

### 4.3. Pengujian Kekuatan Tendangan

Pengujian kekuatan tendangan dilakukan dengan memberikan sinyal PWM kepada ESC. Kemudian ESC menggerakkan motor brushless yang terhubung dengan kaki penendang dan akan menendang bola dengan sudut sehingga membentuk gerak parabola. Mekanisme penendang yang digunakan pada robot dapat dilihat pada gambar 4.6.



**Gambar 4. 6** Posisi penempatan mekanisme penendang

Pengujian kekuatan tendangan bertujuan agar dapat mengetahui tendangan yang sesuai terhadap jarak dari robot ke gawang yang bervariasi. Pengujian tahap ini dilakukan dengan menghitung waktu yang diperlukan oleh kaki penendang untuk melakukan satu putaran dan kemudian dihitung secara matematis untuk mendapatkan kecepatan sudut. Selain kecepatan sudut, jarak maksimal, ketinggian maksimal serta waktu untuk mencapai titik terjauh yang ditempuh oleh bola juga dimasukkan ke tabel hasil pengujian. Tabel hasil pengujian dapat dilihat pada tabel 4.4.

Dari pengujian kekuatan tendangan, akan didapatkan nilai dari tiga parameter. Parameter-parameter tersebut meliputi periode putaran kaki penendang, jarak tempuh maksimal dan waktu tempuh maksimal. Nilai kecepatan sudut dan kecepatan linear didapatkan dari penurunan secara matematis dengan mengasumsikan bahwa nilai jari-jari kaki penendang merupakan konstanta. Hubungan antara kecepatan sudut dan kecepatan linear dapat dilihat pada persamaan (4.1) dan (4.2). Ketinggian maksimal yang dicapai bola didapatkan melalui perhitungan secara matematis menggunakan persamaan (2.9). Sudut elevasi diasumsikan merupakan sebuah konstanta untuk setiap percobaan.

$$\omega = \frac{2\pi}{T} \quad (4.1)$$

$$V = \omega r \quad (4.2)$$

Pengujian tendangan yang dilakukan pada tugas akhir ini menggunakan nilai PWM yang disesuaikan dengan ESC motor

penendang yang dibahas pada bab 3. ESC memerlukan pulsa  $T_{on}$  untuk memutar motor maju dengan rentang dari 1.5ms sampai 2ms. Pada mikrokontroller, rentang nilai tersebut berubah menjadi 1500 dan 2000. Hal ini disebabkan karena menyesuaikan dengan besarnya *clock* yang dimiliki oleh timer 11 sebesar 168MHz. Namun pada tugas akhir ini nilai yang digunakan saat pengujian adalah 1600 sampai 1730 karena saat menggunakan nilai PWM kurang dari 1600 belum menghasilkan tendangan yang lambung dan ketika lebih dari 1730 *gearbox* yang digunakan sudah tidak kuat. Rentang nilai ini selanjutnya diubah dalam persentase dengan rentang 20% sampai 46%.

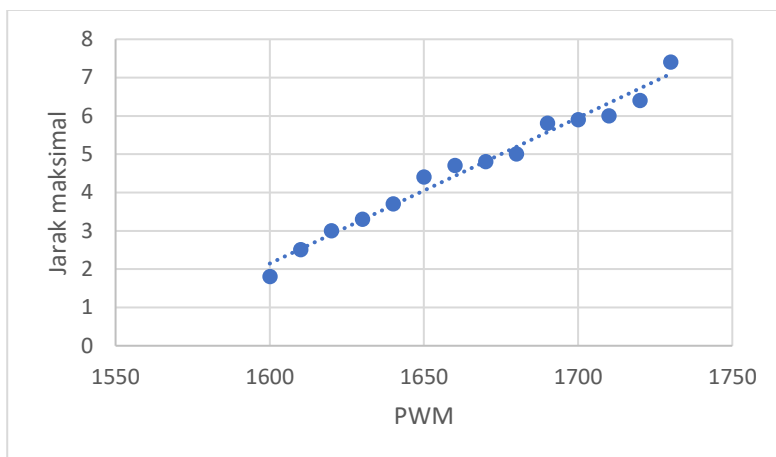
Berdasarkan hasil pengujian kekuatan tendangan yang dilakukan, didapatkan jarak tempuh dan tinggi maksimal bola yang berbeda-beda untuk setiap persentase sinyal yang diberikan. Nilai pulsa dengan persentase 20% tidak digunakan dalam tugas akhir ini karena menghasilkan jarak 1.8m, sedangkan jarak minimal dari robot ke gawang adalah 2m. Pulsa dengan persentase 22% sampai 32% tidak digunakan karena menghasilkan tinggi maksimal yang belum melewati batas maksimal ketinggian robot. Oleh karena itu nilai pulsa yang digunakan pada tugas akhir ini adalah dari 36% sampai 46%.

Pengujian pada tahap ini bertujuan untuk mendapatkan nilai kecepatan sudut, jarak tempuh proyektil maksimal dan waktu tempuh proyektil maksimal. Sedangkan kecepatan linear dan tinggi maksimal didapatkan melalui perhitungan secara matematis yang sudah dibahas pada bab 2. Pengujian pada tahap ini mendapatkan hubungan yang linear dari persentase sinyal PWM yang diberikan dengan parameter yang diamati seperti kecepatan sudut, kecepatan linear, jarak tempuh maksimal, waktu tempuh maksimal dan juga tinggi maksimal. Semakin besar persentase sinyal PWM yang diberikan juga akan menghasilkan nilai parameter yang semakin besar.

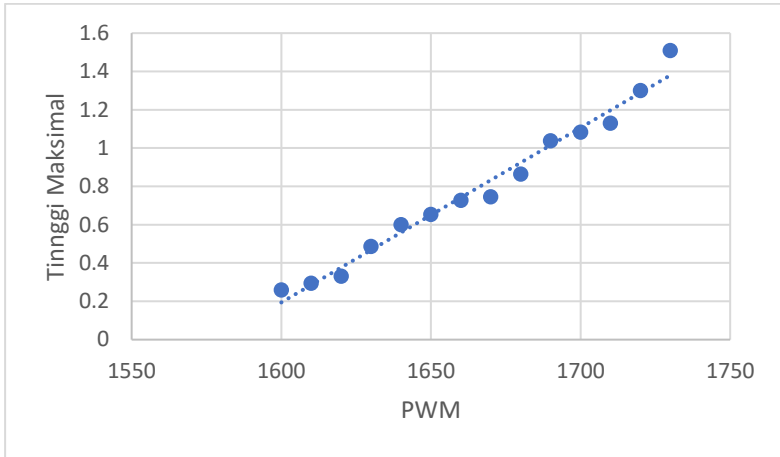
Grafik-grafik yang menunjukkan hasil percobaan pada tabel 4.4 dapat dilihat pada gambar 4.7 sampai 4.9. Gambar 4.7 menunjukkan hubungan jarak maksimal yang ditempuh pada sumbu-x dengan persentase nilai PWM yang digunakan. Gambar 4.8 menunjukkan hubungan tinggi proyektil maksimal dengan persentase nilai PWM yang digunakan. Gambar 4.9 menunjukkan hubungan antara kecepatan linear dengan persentase nilai PWM yang digunakan. Nilai persentase yang digunakan dengan rentang dari 20% hingga 46% mewakili nilai PWM dengan  $T_{on}$  mulai dari 1.6ms hingga 1.73ms dan periode totalnya dengan nilai 20ms.

**Tabel 4. 4** Tabel pengujian kekuatan tendangan

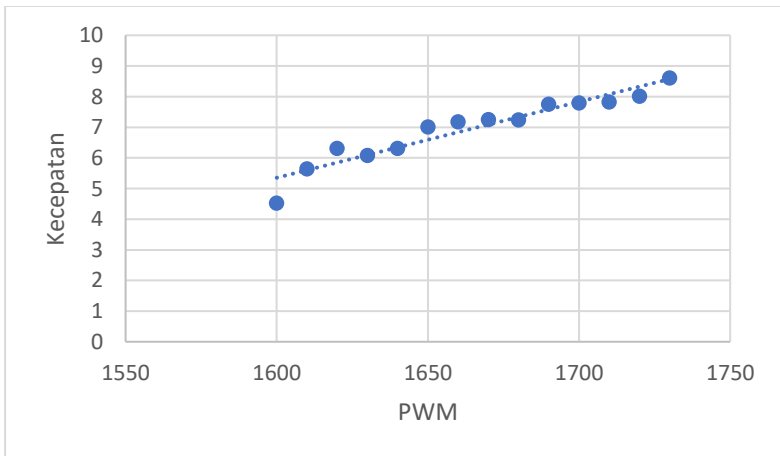
PWM (%)	Omega (rad/s)	V linear (m/s)	Jarak maksimal (m)	Waktu maksimal (s)	Tinggi maksimal (m)
20	38.76	4.52	1.8	0.46	0.26
22	42.72	5.64	2.5	0.49	0.29
24	46.51	6.31	3.0	0.52	0.33
26	49.06	6.32	3.3	0.63	0.49
28	53.22	6.35	3.7	0.70	0.60
30	57.62	7.01	4.4	0.73	0.65
32	60.97	7.18	4.7	0.77	0.73
34	65.41	7.20	4.8	0.78	0.75
36	68.26	7.24	5.0	0.84	0.87
38	70.56	7.75	5.8	0.92	1.03
40	73.02	7.79	5.9	0.94	1.08
42	76.58	7.83	6.0	0.96	1.13
44	78.50	8.01	6.4	1.03	1.30
46	80.51	8.61	7.4	1.11	1.51



**Gambar 4. 7** Grafik hubungan jarak tempuh proyektil maksimal dengan PWM



**Gambar 4. 8** Grafik hubungan tinggi proyektil maksimal dengan PWM



**Gambar 4. 9** Grafik hubungan kecepatan linear dengan PWM

## 4.4. Pengujian Tendangan ke Gawang

### 4.4.1. Pengujian tidak Menggunakan Penjaga Gawang

Pengujian tendangan ke gawang dilakukan dengan meletakkan robot pada titik-titik di lapangan. Titik-titik tersebut dapat dilihat pada gambar 4.2. Titik-titik yang dipilih disesuaikan dengan peraturan-peraturan kontes robot sepakbola beroda Indonesia divisi beroda. Peraturan-peraturan tersebut diantaranya gol dianggap sah apabila robot yang melakukan tendangan tidak berada di daerah bertahan sendiri dan robot yang melakukan tendangan tidak melakukan illegal attack. Illegal attack merupakan kondisi dimana robot penyerang melakukan tendangan pada kotak di depan gawang daerah bertahan lawan.

Pengujian pada tahap ini bertujuan untuk mengetahui seberapa besar tingkat kesuksesan mencetak gol dari titik-titik yang sudah ditentukan di lapangan. Pada pengujian ini tidak terdapat halangan pada lintasan tendangan dari robot hingga gawang. Pengujian pada setiap titik dilakukan dengan lima kali pengulangan. Tendangan akan dianggap sebagai gol apabila bola dapat masuk ke gawang dan lebih dari setengah bagian bola melewati garis gawang. Apabila bola mengenai tiang-tiang gawang atau tidak masuk ke dalam gawang maka dianggap tidak gol. Hasil pengujian tahap ini dapat dilihat pada tabel 4.5 dimana warna hijau menunjukkan pengujian yang berhasil mencetak gol sedangkan warna merah menunjukkan pengujian yang gagal.



**Gambar 4. 10** Penempatan sensor kamera dan LiDAR pada robot



Berdasarkan hasil pengujian yang ditunjukkan pada tabel 4.5, tingkat keberhasilan mencetak gol pada titik A memiliki persentase paling rendah dibandingkan dengan tiga titik lainnya. Hal ini dikarenakan titik A berada paling dekat dengan gawang dan pada percobaan titik A5 dan A1 berada di pinggir lapangan. Hal ini kemudian menyebabkan titik-titik ini memiliki sudut yang kecil terhadap mulut gawang sehingga sulit untuk mencetak gol. Sementara pengujian pada titik B mengalami persentase tingkat keberhasilan yang lebih baik daripada pengujian pada titik A. Pengujian pada titik B yang terletak di pinggir lapangan yaitu B1 dan B5 memiliki keberhasilan yang lebih tinggi, yaitu hanya mengalami satu kali kegagalan. Pengujian pada titik B3 menghasilkan pengujian yang berhasil pada lima kali percobaan. Tingkat keberhasilan mencetak gol paling tinggi berada pada titik C dengan jarak 4m dari garis gawang. Pada jarak ini, pengujian yang dilakukan pada titik-titik di pinggir lapangan seperti C1 dan C5 menghadap ke mulut gawang dengan sudut yang lebih besar dibandingkan pada titik A. Oleh karena itu menghasilkan tingkat keberhasilan mencetak gol yang lebih tinggi. Pengujian pada titik D menghasilkan persentase tingkat kesuksesan mencetak gol yang sama dengan pengujian yang dilakukan pada titik C. Tetapi pengujian yang dilakukan pada titik D3 tidak menghasilkan tingkat keberhasilan mencetak gol yang sempurna. Pengujian pada titik D3 mengalami satu kali kegagalan. Pengujian pada titik D yang mencapai tingkat keberhasilan 100% terdapat pada titik D5.

Pengujian mencetak gol dengan menendang bola ke gawang mengalami beberapa kali kegagalan karena beberapa penyebab. Penyebab pertama adalah mekanisme penggiring bola yang dimiliki robot tidak simetris. Ketidaksimetrisan ini kemudian menghasilkan tarikan bola yang cenderung ke satu sisi dan tarikan bola yang condong ke satu sisi ini apabila ditendang juga akan menghasilkan tendangan yang tidak lurus. Saat sebelum pengujian, mekanisme penggiring bola sudah diatur agar simetris dan menghasilkan tendangan yang lurus. Tetapi perubahan posisi penggiring menjadi tidak simetris di tengah pengujian tidak bisa diprediksi. Penyebab kedua merupakan adanya objek lain yang dideteksi sebagai gawang. Objek ini menyebabkan heading robot tidak mengarah ke gawang yang sebenarnya. Kondisi ini terjadi apabila nilai dari *degree of confidence* yang digunakan tidak sesuai sehingga menyebabkan adanya objek yang bukan gawang justru dideteksi sebagai gawang.

**Tabel 4. 5** Hasil Pengujian tanpa Penjaga Gawang

Titik	Percobaan					Keberhasilan
	1	2	3	4	5	
A1	Green	Green	Red	Green	Green	84%
A2	Green	Green	Green	Green	Green	
A3	Green	Red	Green	Green	Green	
A4	Green	Red	Green	Green	Green	
A5	Red	Green	Green	Green	Green	
B1	Green	Red	Green	Green	Green	88%
B2	Green	Red	Green	Green	Green	
B3	Green	Green	Green	Green	Green	
B4	Red	Green	Green	Green	Green	
B5	Green	Green	Green	Green	Green	
C1	Green	Green	Green	Green	Green	96%
C2	Green	Green	Green	Green	Green	
C3	Green	Red	Green	Green	Green	
C4	Green	Green	Green	Green	Green	
C5	Green	Green	Green	Green	Green	
D1	Green	Green	Red	Green	Green	88%
D2	Green	Red	Green	Green	Green	
D3	Green	Red	Green	Green	Green	
D4	Green	Green	Red	Green	Green	
D5	Green	Green	Green	Green	Green	
Rata-rata						89%

#### 4.4.2. Pengujian Menggunakan Penjaga Gawang

Pengujian tahap ini memiliki prosedur yang identik dengan pengujian yang dilakukan pada poin 4.4.1. Terdapat 20 titik pengujian yang digunakan di lapangan dan dapat dilihat pada gambar 4.2. Perbedaan pada percobaan ini adalah kehadiran robot penjaga gawang yang ditempatkan di dalam gawang. Robot penjaga gawang memiliki dimensi maksimal sebesar 52 x 52 cm dan lengan pada sisi kiri dan kanan yang dapat memanjang hingga mencapai dimensi maksimal 60 x 60 cm dalam waktu maksimal 1 detik. Hal ini menyebabkan hanya tersisa bagian yang lepas dari pengawalan penjaga gawang sepanjang 1.48m dari bagian dalam gawang yang memiliki panjang 2m.

Pengujian pada tahap ini bertujuan untuk mengetahui seberapa besar tingkat kesuksesan mencetak gol dari titik-titik yang sudah ditentukan di lapangan apabila terdapat robot penjaga gawang. Robot penjaga gawang akan berada di dalam gawang dan akan menghalangi agar bola tidak masuk ke gawang. Hasil pembacaan sensor LiDAR tidak akan terganggu oleh keberadaan robot penjaga gawang karena sekuen program akan menyebabkan robot untuk mencari bagian gawang yang tidak dijaga oleh penjaga gawang sebelum mengambil nilai pembacaan sensor LiDAR. Pengujian pada setiap titik dilakukan sebanyak lima kali. Pengujian pertama dan kedua penjaga gawang diletakkan pada bagian kiri gawang, pengujian ketiga penjaga gawang diletakkan pada bagian tengah gawang dan pengujian keempat dan kelima penjaga gawang diletakkan pada bagian tengah gawang.

Hasil pengujian tahap ini dapat dilihat pada tabel 4.6 dimana warna hijau menunjukkan pengujian yang menghasikan gol dan warna merah yang gagal. Pada percobaan ke-1, posisi penjaga gawang diletakkan pada posisi tengah gawang, percobaan ke-2 dan ke-4 penjaga gawang diletakkan pada kiri gawang dan percobaan ke-3 dan ke-5 penjaga gawang diletakkan pada kanan gawang. Kondisi robot penjaga gawang saat melakukan pengujian pada tahap ini tidak dalam kondisi aktif. Robot penjaga gawang dimatikan dan diletakkan pada gawang dengan posisi yang berubah-ubah.

Berdasarkan hasil pengujian yang terdapat pada tabel 4.6, tingkat keberhasilan mencetak gol dengan adanya penjaga gawang berkurang sebesar 16% pada titik A dan B, dan berkurang sebesar 8% pada titik C dan D. Tingkat keberhasilan mencetak gol pada titik A memiliki persentase paling kecil dibandingkan titik lainnya. Hal ini disebabkan karena titik A memiliki jarak yang paling dekat dengan gawang dan menyebabkan robot menghadap ke bagian mulut gawang yang sempit, terutama pada titik A1 dan A5 yang berada pada pinggir lapangan. Pengujian pada titik B memiliki tingkat keberhasilan yang lebih baik daripada di titik A. Hal ini juga disebabkan karena mulut gawang yang menghadap ke robot semakin besar seiring dengan semakin jauhnya jarak robot dari gawang. Pengujian pada titik B mengalami dua kali kegagalan di titik B2 dan B4. Tingkat keberhasilan mencetak gol pada titik C memiliki persentase paling tinggi saat pengujian ini. Persentase keberhasilan untuk mencetak gol dengan pengawalan penjaga gawang mencapai 88%. Tingkat keberhasilan mencetak gol dengan persentase paling rendah dan paling tinggi saat pengujian dengan penjaga gawang

memiliki kesamaan saat pengujian tanpa penjaga gawang, yaitu pada titik A dan titik C. Pengujian pada titik D memiliki persentase yang lebih baik daripada titik A dan B namun kurang baik jika dibandingkan dengan titik C. Pengujian pada tahap ini mengalami setidaknya satu kali kegagalan pada setiap titik percobaan yang dilakukan.

Pengujian pada tahap ini mengalami beberapa kali kegagalan dikarenakan beberapa hal. Penyebab pertama adalah keberadaan robot penjaga gawang. Robot penjaga gawang memiliki dimensi sebesar 52x52cm dan menyisakan bagian sepanjang 1.48m yang lepas dari penjagaan. Badan robot menutupi 26% dari panjang gawang yang pada percobaan sebelumnya bebas dari penjagaan.

**Tabel 4. 6** Hasil Pengujian dengan Menggunakan Penjaga Gawang

Titik	Percobaan					Keberhasilan
	1	2	3	4	5	
A1	Red	Green	Green	Green	Green	68%
A2	Green	Red	Green	Red	Green	
A3	Green	Green	Red	Green	Red	
A4	Green	Green	Red	Red	Green	
A5	Red	Green	Green	Green	Green	
B1	Red	Green	Green	Green	Green	72%
B2	Green	Red	Green	Green	Red	
B3	Green	Green	Green	Red	Green	
B4	Green	Green	Red	Green	Red	
B5	Red	Green	Green	Green	Green	
C1	Red	Green	Green	Green	Green	88%
C2	Green	Green	Green	Green	Green	
C3	Green	Green	Green	Green	Green	
C4	Red	Green	Green	Green	Green	
C5	Red	Green	Green	Green	Green	
D1	Green	Green	Green	Red	Green	80%
D2	Red	Green	Green	Green	Green	
D3	Green	Green	Red	Green	Green	
D4	Red	Green	Green	Green	Green	
D5	Green	Red	Green	Green	Green	
Rata-rata						77%

## **BAB 5**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan percobaan yang dilakukan pada pelaksanaan tugas akhir *Implementasi LiDAR pada Robot Sepakbola Beroda untuk Tendangan ke Gawang* dapat ditarik beberapa kesimpulan. Pertama, jaringan syaraf tiruan yang telah dilatih untuk mendeteksi gawang menggunakan model SSD Mobilenet v2 memiliki akurasi mencapai 100% pada jarak 4meter dari gawang dengan lux sebesar 11 dan 48. Jaringan syaraf tiruan ini berhasil mendeteksi gawang dengan nilai *degree of confidence* dari 0.829908 hingga 0.999999. Kedua, sensor LiDAR dapat menghasilkan pengukuran ke gawang dengan rata-rata error RMSE sebesar 2.104 dibandingkan dengan jarak sebenarnya. Ketiga, kekuatan tendangan yang diperlukan untuk mencetak gol pada robot yang digunakan di tugas akhir ini memerlukan pulsa dengan persentase mulai dari 36% hingga 46%. Kemudian keberhasilan pengujian tendangan ke gawang pada titik-titik yang telah ditentukan pada gambar memiliki tingkat kesuksesan lebih dari 84% dengan rata-rata 89% pada kondisi tanpa penjaga gawang dan lebih dari 68% dengan rata-rata 77% saat ada penjaga gawang. Urutan jarak yang menghasilkan tingkat keberhasilan mencetak gol paling tinggi hingga kecil untuk mencetak gol berdasakan pengujian berada pada titik C, D, B dan A saat pengujian tanpa penjaga dan pengujian dengan penjaga gawang.

#### **5.2. Saran**

Sebagai sarana pengembangan penelitian pada robot sepakbola beroda, terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan. Saran ini diharapkan dapat meningkatkan performa robot sepakbola beroda agar lebih baik. Saran-saran tersebut meliputi mekanisme penggiring bola pada robot dibuat lebih baik agar menghasilkan tendangan yang lurus. Selanjutnya dataset gawang yang digunakan pada pelatihan jaringan syaraf tiruan dibuat lebih banyak dan bervariasi agar lebih tahan terhadap perubahan lingkungan.

*Halaman ini sengaja dikosongkan*

## Daftar pustaka

- [1] “KRI 2019,” *Kontes Robot Indonesia*. [Online]. Available: <https://kontesrobotindonesia.id/kri-2019.html>. [Accessed: 25-May-2019].
- [2] and E. Rudiawan, R. Analia, P. D. Sutopo, and H. Soebakti, “The deep learning development for real-time ball and goal detection of barelang-FC,” in *2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2017, pp. 146–151.
- [3] M. Asada *et al.*, “MSL Technical Committee 1997–2019,” p. 107.
- [4] Heru, E. Pitowarno, and K. Mutijarsa, “Buku Panduan Kontes Robot Sepakbola Beroda Indonesia 2018,” Dec. 2018.
- [5] D. Halliday, R. Resnick, and J. Walker, “Fundamentals of Physics 11th.” John Wiley & Sons, 2014.
- [6] P. Dong and Q. Chen, *LiDAR remote sensing and applications*. Boca Raton London New York: CRC Press, Taylor & Francis, 2018.
- [7] J. Liu, Q. Sun, Z. Fan, and Y. Jia, “TOF Lidar Development in Autonomous Vehicle,” p. 6.
- [8] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image processing using MATLAB*. Upper Saddle River, NJ: Pearson/Prentice Hall, 2004.
- [9] “About.” [Online]. Available: <https://opencv.org/about/>. [Accessed: 25-May-2019].
- [10] *Open Source Computer Vision Library. Contribute to opencv/opencv development by creating an account on GitHub*. OpenCV, 2019.
- [11] D. Livingstone, Ed., *Artificial neural networks: methods and applications*. Totowa, NJ: Humana Press, 2008.
- [12] R. E. Neapolitan, *Artificial Intelligence: With an Introduction to Machine Learning*, 2nd ed. Chapman and Hall/CRC, 2018.
- [13] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” *ArXiv151202325 Cs*, vol. 9905, pp. 21–37, 2016.
- [14] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, “Scalable, High-Quality Object Detection,” *ArXiv14121441 Cs*, Dec. 2014.
- [15] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *ArXiv170404861 Cs*, Apr. 2017.

- [16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *ArXiv180104381 Cs*, Jan. 2018.
- [17] "ROS.org | About ROS." [Online]. Available: <https://www.ros.org/about-ros/>. [Accessed: 26-May-2019].
- [18] L. Joseph, *Robot operating system for absolute beginners: robotics programming made easy*. New York, NY: Apress, 2018.
- [19] M. Quigley, B. Gerkey, and W. D. Smart, "Programming Robots with ROS," p. 558.
- [20] "Topics - ROS Wiki." [Online]. Available: <http://wiki.ros.org/Topics>. [Accessed: 26-May-2019].
- [21] "msg - ROS Wiki." [Online]. Available: <http://wiki.ros.org/msg>. [Accessed: 26-May-2019].
- [22] D. Capel and B. Ist, "Basic principles of photography," p. 44.
- [23] J. Chouinard, "The Fundamentals of Camera and Image Sensor Technology," p. 157.
- [24] "Intel® NUC Kit NUC6i7KYK Product Specifications." [Online]. Available: <https://ark.intel.com/content/www/us/en/ark/products/89187/intel-nuc-kit-nuc6i7kyk.html>. [Accessed: 26-May-2019].
- [25] "Support for Intel® Iris™ Pro Graphics 580," *Intel*. [Online]. Available: <https://www.intel.com/content/www/us/en/support/products/88357/graphics-drivers/graphics-for-6th-generation-intel-processors/intel-iris-pro-graphics-580.html>. [Accessed: 26-May-2019].
- [26] A. Dwi Rizqi Aljabar, M. Rivai, and S. Suwito, "Rancang Bangun Robot Omni untuk Mencari Sumber Gas Menggunakan Kontrol Logika Fuzzy Berbasis Mikrokontroler ARM STM-32," *J. Tek. ITS*, vol. 7, Mar. 2018.
- [27] A. Tovar, D. Amaya, and O. Ramos, "SERIAL DATA TRANSMISSION THROUGH THE MICRO-USB PORT OF THE SYSTEM STM32F407," vol. 13, no. 2, p. 6, 2018.
- [28] Y. Zhang and L. Li, "Smart Home System Based on ZigBee Network and STM32F407 Microprocessor," *Open Autom. Control Syst. J.*, vol. 6, no. 1, pp. 1258–1266, Dec. 2014.
- [29] A. Hughes, *Electric motors and drives: fundamentals, types, and applications*, 3rd ed. Amsterdam ; Boston: Elsevier/Newnes, 2006.
- [30] S. Kim, *Electric motor control: DC, AC, and BLDC motors*. 2017.



- [31] S. J. Chapman, *Electric machinery fundamentals, fifth edition*. New York: McGraw Hill, 2012.
- [32] R. Büchi, *Brushless Motors and Controllers*. Norderstedt: Books on Demand, 2012.
- [33] R. Prayogo, “KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS BRAWIJAYA MALANG DESEMBER 2012,” p. 24.

*Halaman ini sengaja dikosongkan*

## Lampiran A

### A. Program deteksi gawang

```
//command untuk build
//g++ -ggdb gabung.cpp -o gabung `pkg-config --cflags --libs opencv`

//library udp=====
#include <iostream>
#include <arpa/inet.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
//=====

//library opencv=====
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include "opencv2/dnn.hpp"
#include <iostream>
//=====

//deklarasi udp
#define SERVICE_PORT 33333
#define BUFLen 64
//=====

using namespace cv;
using namespace std;
using namespace dnn;

//deteksi kontur robot musuh=
int low_H = 150, low_S = 101, low_V = 106;
int high_H = 166, high_S = 165, high_V = 160;
//=====
```

```

//=====variabel global
short int jarak, tengah_roi_dari_pinggir;
short int penanda = 0;           // ini penanda untuk mencari kotak
deteksi yang pertama
short int penanda_kontur = 0;    // ini penanda untuk mencari
kontur di kotak deteksi yang pertama
short int nilai_tengah_pertama; // ini nilai tengah kotak deteksi
yang pertama dideteksi
short int nilai_tengah_kontur;
short int nilai_tengah_kontur_baru;
short int nilai_tengah_kontur_pertama = 0;

String classes[2];
Scalar color[2];

int main(int argc, char **argv){

//ini bagian udp=====
int sockfd;
// char buffer[BUFLEN];
sockfd = socket(AF_INET, SOCK_DGRAM, 0);
struct sockaddr_in client;

client.sin_family = AF_INET;
client.sin_port = htons(SERVICE_PORT);
client.sin_addr.s_addr = inet_addr("127.0.0.1");

socklen_t socklen = sizeof(client);

char buffer[BUFLEN] = {'i', 't', 's'};

//=====

classes[1]="gawang";
classes[2]="robot";
color[1]=Scalar(255,0,0);
color[2]=Scalar(0,0,255);

VideoCapture stream1(1);

```

```

if (!stream1.isOpened()) { //check if video device has been initialised
cout << "cannot open camera";
}

Mat blob;
Mat outs;
Net net;
Mat cameraFrame;

//ini untuk frame yang pertama kali
Mat roi_pertama;
Mat tresold_pertama;
Mat gambar_pertama;
//=====

//ini untuk frame yang terus loop
Mat roi;
Mat tresold;
Mat gambar;
//=====

//ini untuk mendeteksi kontur
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;
//=====

net = readNetFromTensorflow("sorted_inference_graph.pb",
"label_map_2.pbtxt");

int idx, xLeftBottom=0, yLeftBottom=0, xRightTop=0, yRightTop=0;

while (true) {
    stream1.read(cameraFrame);
    // cout << cameraFrame.cols << ";" << cameraFrame.rows << endl;

    net.setInput(blobFromImage(cameraFrame, 1.0, Size(300, 300), true,
false));
    net.forward(outs);
}

```

```

    Mat      detectionMat(outs.size[2],      outs.size[3],      CV_32F,
outs.ptr<float>());

    //gambar      garis      untuk      range      tengah
frame=====
    line(cameraFrame,      Point(300,10),      Point(300,470),
Scalar(255,255,255),3);
    line(cameraFrame,      Point(340,10),      Point(340,470),
Scalar(255,255,255),3);

//=====
=====

float confidenceThreshold = 0.999;
    for (int i = 0; i < detectionMat.rows; i++)
        {
            float confidence = detectionMat.at<float>(i,
2);

            if (confidence > confidenceThreshold &&
penanda==0)
                {
                    //ini untuk findcontour di case ini
                    vector<vector<Point> > contours1;
                    vector<Vec4i> hierarchy1;

//=====

                    //untuk write file=====
                    vector<int> compression_params;

                    compression_params.push_back(CV_IMWRITE_PNG_COMPRESSIO
N);
                    compression_params.push_back(9);
                    //=====

                    cout << "kondisi 1" << endl;
                                idx                                =
static_cast<int>(detectionMat.at<float>(i, 1));

```

```

        xLeftBottom =
static_cast<int>(detectionMat.at<float>(i, 3) * cameraFrame.cols);
        yLeftBottom =
static_cast<int>(detectionMat.at<float>(i, 4) * cameraFrame.rows);
        xRightTop =
static_cast<int>(detectionMat.at<float>(i, 5) * cameraFrame.cols);
        yRightTop =
static_cast<int>(detectionMat.at<float>(i, 6) * cameraFrame.rows);

        Rect object(abs((int)xLeftBottom),
abs((int)yLeftBottom), (int)(xRightTop - xLeftBottom),(int)(yRightTop
- yLeftBottom));

        rectangle(cameraFrame, object, color[idx], 2);

        if((object.width + object.x) >= 640)
            object.width -= object.x + object.width - 640;
        if((object.height + object.y) >= 480)
            object.height -= object.y + object.height - 480;

        roi_pertama = cameraFrame(Rect(object.x, object.y,
object.width, object.height));
        gambar_pertama = Mat::zeros(roi_pertama.size(),
CV_8UC3);
        tengah_roi_dari_pinggir = object.x+(object.width/2);

        cvtColor(roi_pertama,tresold_pertama,COLOR_BGR2HSV);
        inRange(tresold_pertama, Scalar(low_H, low_S, low_V),
Scalar(high_H, high_S, high_V), tresold_pertama);
        findContours(tresold_pertama, contours1, hierarchy1,
RETR_EXTERNAL, CHAIN_APPROX_NONE, Point(0, 0));

        vector<Rect> boundrect1(contours1.size());

        for (int i = 0; i < contours1.size(); i++) {
            drawContours(gambar_pertama, contours1, i, Scalar(128,
255, 255), 2, 8, hierarchy1, 0, Point());
            cout<<contours1[i].size()<< " <--- ini contour size"<<endl;

```

```

        if (contours1[i].size()>100){
            boundrect1[i] = boundingRect(contours1[i]);
            rectangle(gambar_pertama, boundrect1[i], Scalar(255,
255, 0), 2);

            Moments m = moments(contours1[i], true);
            Point p(m.m10 / m.m00, m.m01 /
m.m00);
            circle(gambar_pertama, p, 5,
Scalar(128, 0, 0), -1);
            nilai_tengah_kontur_pertama = object.x + p.x;
            cout << p.x << "<----ini moment dari kontur" << endl;
        }
    }

    imwrite("roi_pertama.png",roi_pertama,compression_params);

    imwrite("gambar_pertama.png",gambar_pertama,compression_params)
;

    //ini di comment
    //          line(cameraFrame,Point(tengah_roi_dari_pinggir
,10),Point(tengah_roi_dari_pinggir,470),Scalar(255,255,255),3);

    nilai_tengah_pertama = tengah_roi_dari_pinggir;
    penanda_kontur=1;
    penanda=1;
} // ini tutup kurung kondisi 1

else if (confidence > confidenceThreshold && penanda == 1)
{
    //ini untuk find contour di case ini
    vector<vector<Point> > contours2;
    vector<Vec4i> hierarchy2;
    //=====

    cout << "kondisi 2" << endl;
    idx = static_cast<int>(detectionMat.at<float>(i, 1));

```



```

        xLeftBottom = static_cast<int>(detectionMat.at<float>(i, 3) *
cameraFrame.cols);
        yLeftBottom = static_cast<int>(detectionMat.at<float>(i, 4) *
cameraFrame.rows);
        xRightTop = static_cast<int>(detectionMat.at<float>(i, 5) *
cameraFrame.cols);
        yRightTop = static_cast<int>(detectionMat.at<float>(i, 6) *
cameraFrame.rows);

        Rect object(abs((int)xLeftBottom), abs((int)yLeftBottom),
(int)(xRightTop - xLeftBottom),(int)(yRightTop - yLeftBottom));

        rectangle(cameraFrame, object, color[idx], 2);

        if((object.width + object.x) >= 640)
            object.width -= object.x + object.width - 640;
        if((object.height + object.y) >= 480)
            object.height -= object.y + object.height - 480;

        roi = cameraFrame(Rect(object.x, object.y, object.width,
object.height));
        cvtColor(roi,tresold,COLOR_BGR2HSV);
        inRange(tresold, Scalar(low_H, low_S, low_V),
Scalar(high_H, high_S, high_V), tresold);
        findContours(tresold, contours2, hierarchy2,
RETR_EXTERNAL, CHAIN_APPROX_NONE, Point(0, 0));

        gambar = Mat::zeros(roi.size(), CV_8UC3);
        vector<Rect> boundrect2(contours2.size());

        tengah_roi_dari_pinggir = object.x+(object.width/2);

        for (int i = 0; i < contours2.size(); i++) {
            drawContours(gambar, contours2, i, Scalar(128, 255, 255),
2, 8, hierarchy2, 0, Point());
        }

        jarak=(cameraFrame.cols/2)-tengah_roi_dari_pinggir;

```

```

//ini di comment
//          line(cameraFrame,Point(tengah_roi_dari_pinggir
,10),Point(tengah_roi_dari_pinggir,470),Scalar(255,255,255),3);

// namedWindow("roi", WINDOW_AUTOSIZE);
// imshow("roi", roi);
// namedWindow("kotak", WINDOW_AUTOSIZE);
// imshow("kotak", gambar);
}

}

if(penanda==1&&penanda_kontur==1){
    Mat hasil_roi_pertama = imread("roi_pertama.png");
    Mat hasil_gambar_pertama = imread("gambar_pertama.png");
    // imshow("hasil_roi_pertama", hasil_roi_pertama);
    // imshow("hasil_gambar_pertama", hasil_gambar_pertama);
}

//ini di comment
//          line(cameraFrame,Point(nilai_tengah_pertama,10),
Point(nilai_tengah_pertama,470),Scalar(0,100,255),3);

if(nilai_tengah_pertama <240 ){
    // ini untuk dari kanan lapangan
    nilai_tengah_kontur_baru = nilai_tengah_kontur_pertama-50;
    short int jarak_baru = jarak-nilai_tengah_pertama;
    short      int      jarak_kontur_baru      =      jarak-
nilai_tengah_kontur_pertama;
    //   ini nilai tengah kotak      ini jarak tengah
    //   roi      kotak roi ke tengah
    //      frame
    // cout << nilai_tengah_pertama << "      " << jarak << "      " <<
jarak-nilai_tengah_pertama << "      " << nilai_tengah_kontur_pertama <<
"      " << nilai_tengah_kontur_baru << "      " << jarak-
nilai_tengah_kontur_pertama << endl;

```

```

        cout << nilai_tengah_pertama << "      " << jarak << "      " <<
jarak_baru << "      " << nilai_tengah_kontur_pertama << "      " <<
nilai_tengah_kontur_baru << "      " << jarak_kontur_baru << endl;

        // sprintf(buffer,"%d %d %d %d %d %d", nilai_tengah_pertama,
jarak, (jarak-nilai_tengah_pertama), nilai_tengah_kontur_pertama,
nilai_tengah_kontur_baru, jarak-nilai_tengah_kontur_pertama);
        // sendto(sockfd, buffer, BUFLen, 0, (struct
sockaddr *)&client, socklen);
        memcpy(buffer+3, &nilai_tengah_pertama, 2);
        memcpy(buffer+5, &jarak, 2);
        memcpy(buffer+7, &jarak_baru, 2);
        memcpy(buffer+9, &nilai_tengah_kontur_pertama, 2);
        memcpy(buffer+11, &nilai_tengah_kontur_baru, 2);
        memcpy(buffer+13, &jarak_kontur_baru, 2);

        sendto(sockfd, buffer, BUFLen, 0, (struct sockaddr *)&client,
socklen);

    }
    else if (nilai_tengah_pertama >240){
        //ini untuk dari kiri lapangan
        short      int      nilai_tengah_pertama_modified      =
abs(nilai_tengah_pertama-320);
        short int jarak_modified = abs(jarak-320)-80;
        short int delta = (abs(jarak-320)-100)-abs(nilai_tengah_pertama-
320);
        // short int jarak_baruz = jarak-nilai_tengah_kontur_pertama;
        nilai_tengah_kontur_baru=nilai_tengah_kontur_pertama+50;
        short      int      jarak_baruz=(2.1*jarak_modified)-
nilai_tengah_kontur_baru;
        // cout << nilai_tengah_pertama_modified << "      " <<
jarak_modified << "      " << delta << "      " <<
nilai_tengah_kontur_pertama << "      " << nilai_tengah_kontur_baru <<
"      " << jarak-nilai_tengah_kontur_pertama << endl;
        cout << nilai_tengah_pertama_modified << "      " <<
jarak_modified << "      " << delta << "      " <<
nilai_tengah_kontur_pertama << "      " << nilai_tengah_kontur_baru <<
"      " << jarak_baruz << endl;

```

```

        // sprintf(buffer,"%d %d %d %d %d %d", nilai_tengah_pertama,
jarak, (jarak-nilai_tengah_pertama), nilai_tengah_kontur_pertama,
nilai_tengah_kontur_baru, jarak-nilai_tengah_kontur_pertama);
        // sendto(sockfd, buffer, BUFLen, 0, (struct
sockaddr *)&client, socklen);

        memcpy(buffer+3, &nilai_tengah_pertama_modified, 2);
        memcpy(buffer+5, &jarak_modified, 2);
        memcpy(buffer+7, &delta, 2);
        memcpy(buffer+9, &nilai_tengah_kontur_pertama, 2);
        memcpy(buffer+11, &nilai_tengah_kontur_baru, 2);
        memcpy(buffer+13, &jarak_baruz, 2);

        sendto(sockfd, buffer, BUFLen, 0, (struct sockaddr *)&client,
socklen);
    }

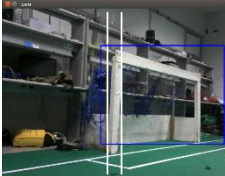


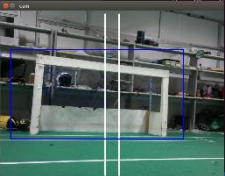
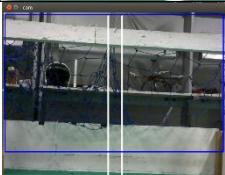
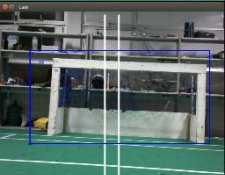

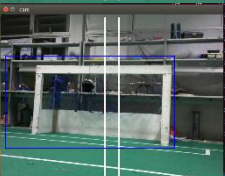
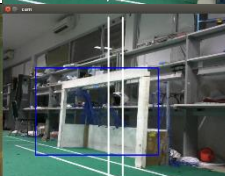

    imshow("cam", cameraFrame);


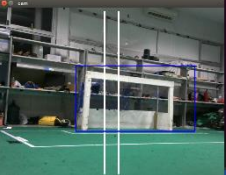
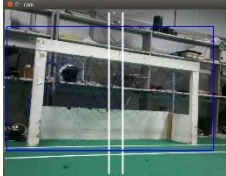
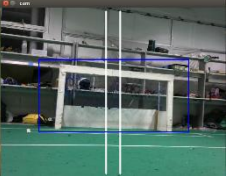



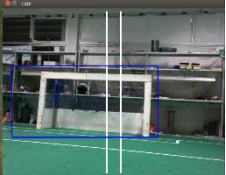
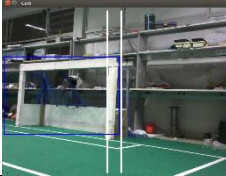

    if (waitKey(30) >= 0)
        break;
} // <-----ini tutup kurung while(true)
return 0 ;
} // <-----ini tutup kurung main program

```




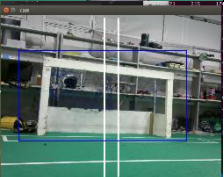
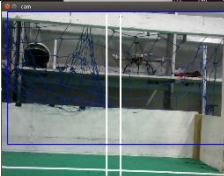
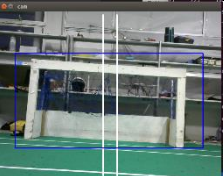

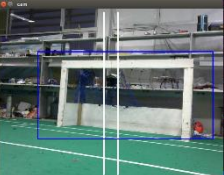



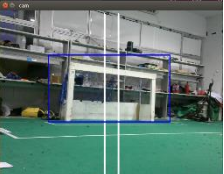
## Lampiran B Dokumentasi Data Percobaan


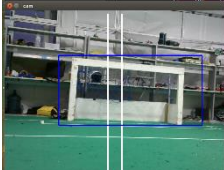

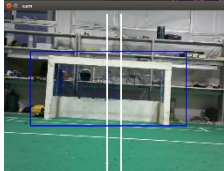



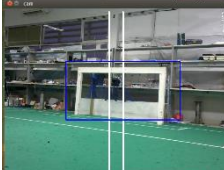
### A. Pengujian deteksi gawang pada malam hari dengan tingkat lumen 11

A1		C1	
A2		C2	
A3		C3	
A4		C4	
A5		C5	

B1		D1		
B2		D2		
B3		D3		
B4		D4		
B5		D5		

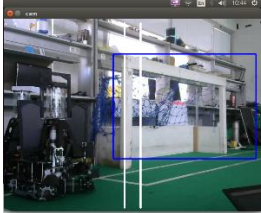


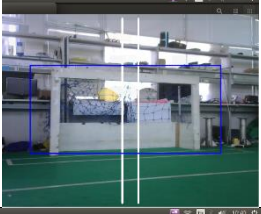

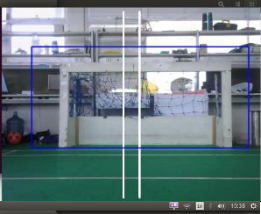

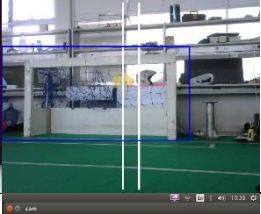

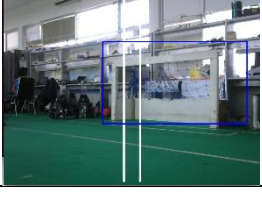
B. Pengujian deteksi gawang pada malam hari dengan tingkat lumen 11



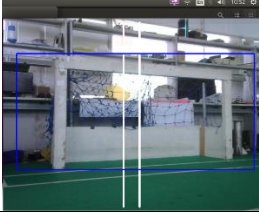
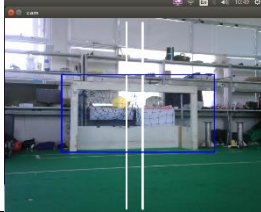
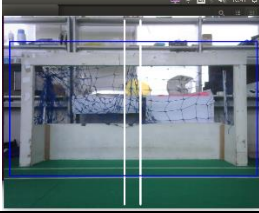
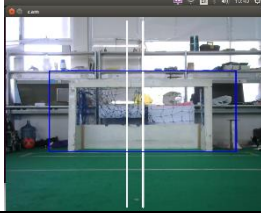
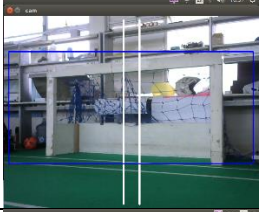
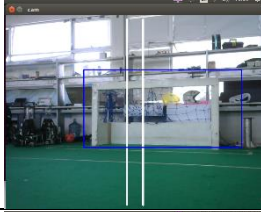
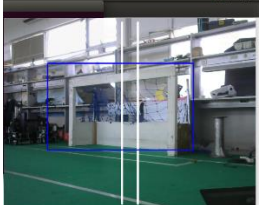

A1		C1	
A2		C2	
A3		C3	
A4		C4	
A5		C5	
B1		D1	

B2		D2	
B3		D3	
B4		D4	
B5		D5	



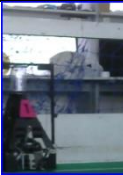
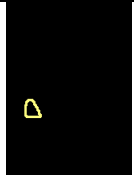

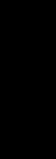









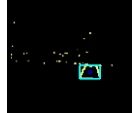













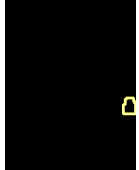












### C. Pengujian deteksi gawang pada pagi hari dengan tingkat lumen 52

A1		C1	
A2		C2	
A3		C3	
A4		C4	
A5		C5	

B1		D1	
B2		D2	
B3		D3	
B4		D4	
B5		D5	

D. Pengujian Tendangan ke Gawang dengan Penjaga Gawang

Titik	Bounding Box	Hasil	Titik	Bounding Box	Hasil
A1			C1		
A2			C2		
A3			C3		
A4			C4		
A5			C5		
B1			D1		

B2			D2		
B3			D3		
B4			D4		
B5			D5		

## Biodata Penulis



Penulis, lahir di Kota Bandar Lampung pada tanggal 24 April 1997, merupakan anak pertama dari dua bersaudara. Penulis menyelesaikan pendidikan dasar di SD Kartika II-5 Bandar Lampung, lalu melanjutkan pendidikan tingkat menengah pertama di SMPN 2 Bandar Lampung dan Pendidikan tingkat menengah atas di SMAN 2 Bandar Lampung dengan jurusan ilmu pengetahuan alam. Penulis memulai kehidupan perkuliahan pada tahun 2015 di S1 Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember di Surabaya. Selama masa perkuliahan penulis aktif di bidang teknologi dengan bergabung dengan unit kegiatan mahasiswa (UKM) robotika dan tim robot sepakbola beroda indonesia divisi beroda. Selain itu, penulis juga menjadi asisten praktikum di bidang studi elektronika.

Email : [maulana.falithurrahman@gmail.com](mailto:maulana.falithurrahman@gmail.com)

HP/WA : 08575836616

Line : falithurrahman