



**TUGAS AKHIR -TE 141599**

# **IMPLEMENTASI AUTO DOCKING SYSTEM PADA MOBILE ROBOT**

Novan Eka Prayoga  
NRP 2211106078

Dosen Pembimbing  
Rudy Dikairono, ST.,MT  
Ir. Tasripan, MT

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



**FINAL PROJECT -TE 141599**

**IMPLEMENTATION OF AUTO DOCKING SYSTEM FOR  
MOBILE ROBOT**

Novan Eka Prayoga  
NRP 2211106078

Supervisor  
Rudy Dikairono, ST.,MT  
Ir. Tasripan, MT

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

**IMPLEMENTASI AUTO DOCKING SYSTEM PADA  
MOBILE ROBOT**

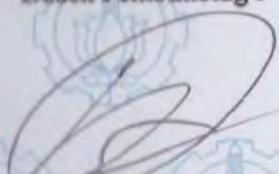
**TUGAS AKHIR**

Diajukan untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada

Bidang Studi Elektronika  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

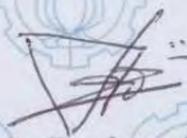
Menyetujui :

Dosen Pembimbing I



Rudy Dikairono, ST., MT.  
NIP: 198103252005011002

Dosen Pembimbing II



Ir. Tasripan, MT.  
NIP: 196204181990031004



# IMPLEMENTASI AUTODOCKING SYSTEM PADA MOBILE ROBOT

**Nama** : Novan Eka Prayoga  
**Pembimbing I** : Rudy Dikairono, ST.,MT.  
**Pembimbing II**: Ir. Tasripan, MT.

## ABSTRAK

Pada Tugas akhir ini perancangan robot menyusuri ruangan dengan ruangan yang sudah ditentukan. pada proses ini robot akan berjalan maju dari docking station dengan posisi docking station berada pada sebelah kiri ruangan pada saat proses robot kembali menuju docking station, yang digunakan sebagai acuan adalah rotary encoder jadi pada saat robot bergerak menyusuri ruangan, rotary encoder akan menghitung pulsa pada setiap putaran rodanya. ketika baterai habis pada saat robot saat bergerak maju dengan pembacaan rotary encoder, maka robot akan bergerak mundur dengan counter pembacaan rotary encoder, lalu robot akan berhenti. Setelah robot berhenti, agar konektor baterai pada robot dapat terhubung dengan konektor charger pada docking station terdapat push button tujuannya adalah untuk menghentikan robot agar tidak bergerak dan terdapat pir yang gunanya untuk mencocokkan antar konektor pir pada robot dan kepingian PCB pada *docking station*, untuk keperluan *charger* baterai.

Pengujian pada Tugas Akhir ini menunjukkan bahwa sensor rotary encoder mampu menavigasi robot dalam satuan pulsa dengan *error* sebesar 5 dan 7 pulsa setiap pada rotary encoder. Pada pembacaan jarak untuk penghindar menggunakan sensor ultrasonic memiliki *error* dalam satuan cm dengan *error* rata rata sebesar 0,6cm. kemudian untuk pembacaan motor servo sebagai tempat ip cam untuk pengawas memiliki *error* putaran sebesar 1%. hasil *error* terhadap rotary encoder saat menuju docking station mengalami gangguan, maka dibutuhkan sensor photodiode untuk membantu mendekatkan robot dengan docking station. nilai pembacaan sensor tegangan berubah ubah. untuk sensor tegangan pada robot memiliki *error* sebesar 0,43 V. Ip cam membutuhkan perangkat router untuk mengakses ip cam dan PC

**Kata Kunci** : *autodocking system, docking station, charger, navigasi*

# **IMPLEMENTATION OF AUTO DOCKING SYSTEM FOR MOBILE ROBOT**

**Name** : Novan Eka Prayoga  
**1<sup>st</sup> Advisor** : Rudy Dikairono, ST.,MT.  
**2<sup>nd</sup> Advisor** : Ir. Tasripnan, MT.

## **ABSTRACT**

*At this final project design of the robot down the room with a room that was already determined. in this process the robot will run forward from the docking station with a docking station position is on the left in the room by the time the robot back into the docking station, which is used as a reference is the rotary encoder so that when the robot moves through the room, the rotary encoder will count pulses on each putran wheels. when the battery is discharged at the time when the robot moves forward with rotary encoder readings, then the robot will move backward by the rotary encoder counter readings, then the robot will stop. After the robot stops, so that the robot can konektor battery connected to the charger connector on the docking station there is a push button to stop the goal is for the robots to not move and there is a pear that point to match the connector between the robot and kepingian pear PCB on the docking station, for the purposes of battery charger.*

*Testing at this final project shows that the rotary encoder sensor able to navigate the robot in units of pulses with an error of 5 and 7 each on a rotary pulse encoder. In the reading distance for the avoidance using ultrasonic sensors has an error in saatuan cm with an average error of 0,6cm. then to the reading of a servo motor as a watchdog memiliki ip cam for rotation error of 1%. the result of an error on the rotary encoder on the way to the docking station impaired, it is necessary to help close the photodiode sensor robot with a docking station. the value of the voltage sensor readings fluctuations. for voltage sensors on the robot has an error of 0.43 V. Ip cam need the router to access ip cam and PC*

**Keywords** : auto docking system, docking station, charger, navigation

## **KATA PENGANTAR**

Puji syukur penulis haturkan atas kehadiran Allah SWT atas segala rahmat, karunia, petunjuk, serta hidayah-Nya yang telah diberikan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

### **IMPLEMENTASI AUTO DOCKING SYSTEM PADA MOBILE ROBOT**

Tugas Akhir ini ditujukan untuk memenuhi salah satu persyaratan dalam menyelesaikan Studi Strata-1 di Jurusan Teknik Elektro, ITS Surabaya. Penulis menyadari bahwa dalam penyelesaian pengerjaan Tugas Akhir ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Untuk itu dengan segala hormat, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah berperan serta dalam penyelesaian Tugas Akhir ini dari awal sampai akhir, khususnya pada dosen pembimbing, Rudy Dikairono, ST.,MT. dan Ir. Tasripan, MT. di Teknik Elektro ITS, orang tua dan seluruh keluarga yang memberikan dukungan penuh kepada penulis serta kepada semua pihak yang telah membantu dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari masih terdapat kekurangan pada Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan ilmu pengetahuan dan teknologi di kemudian hari.

Surabaya, Januari 2015

Penulis

## DAFTAR ISI

|  |      |
|--|------|
| HALAMAN JUDUL  |      |
| PERNYATAAN KEASLIAN TUGAS AKHIR                      |      |
| HALAMAN PENGESAHAN                                   |      |
| ABSTRAK .....  | i    |
| ABSTRACTK .....                                      | iii  |
| KATA PENGANTAR .....                                 | v    |
| DAFTAR ISI .....                                     | vii  |
| DAFTAR GAMBAR .....                                  | xi   |
| DAFTAR TABEL .....                                   | xiii |
| <br>   |      |
| BAB I PENDAHULUAN .....                              | 1    |
| 1.1 Latar Belakang .....                             | 2    |
| 1.2 Perumusan Masalah .....                          | 2    |
| 1.3 Batasan Masalah .....                            | 2    |
| 1.4 Tujuan Penelitian .....                          | 3    |
| 1.5 Metodologi Penelitian .....                      | 3    |
| 1.6 Sistematika Penulisan .....                      | 5    |
| 1.7 Relevansi .....                                  | 6    |
| <br>   |      |
| BAB II TINJAUAN PUSTAKA DAN DASAR TEORI .....        | 7    |
| 2.1 Robot .....                                      | 7    |
| 2.1.1 Klasifikasi umum robot .....                   | 7    |
| 2.1.2 Omniwheel robot .....                          | 8    |
| 2.2 Baterai Lipo .....                               | 9    |
| 2.2.1 Tegangan (voltage) .....                       | 11   |
| 2.2.2 Kapasitas (Capacity) .....                     | 11   |
| 2.2.3 Discharge rate .....                           | 12   |
| 2.2.4 Hambatan dalam (Internal resistance) .....     | 12   |
| 2.3 Charger dan balancer .....                       | 13   |
| 2.3.1 waktu pengisian baterai .....                  | 14   |
| 2.3.2 contoh proses autodocking .....                | 15   |
| 2.4 Mikrokontroler Arduino Mega 2560 .....           | 16   |
| 2.4.1 Diagram Blok fungsi PIN pada kit Arduino ..... | 17   |
| 2.5 Rotary Encoder .....                             | 20   |
| 2.5.1 Prinsip kerja Rotary encoder .....             | 20   |
| 2.6 Sensor ultrasonik .....                          | 23   |
| 2.6.1 Sensor ultrasonic HC-SR04 .....                | 27   |

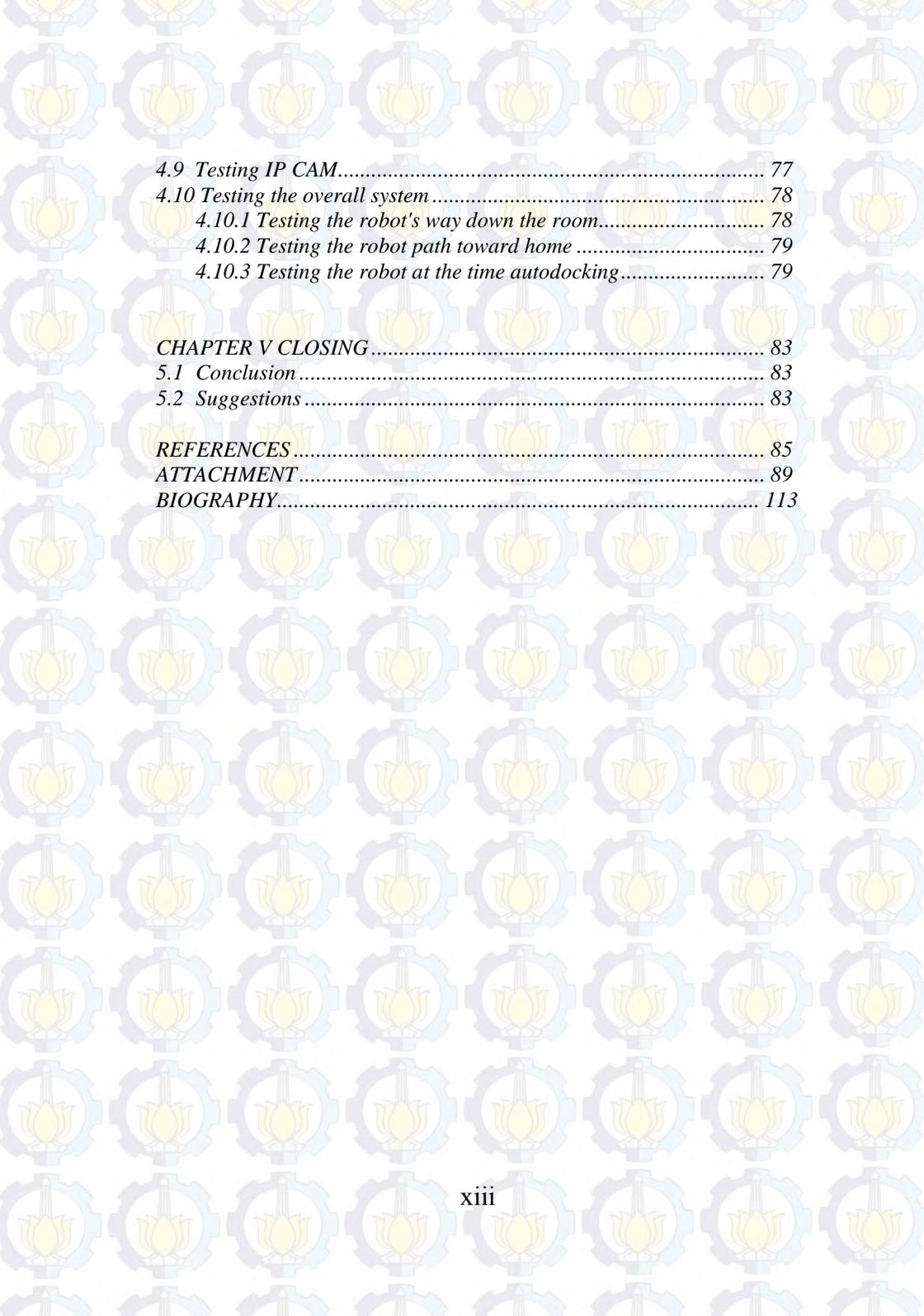
|  |           |
|--|-----------|
| 2.6.2 Prinsip kerja sensor ultrasonic .....                  | 28        |
| 2.7 LCD M1632 .....  | 30        |
| 2.7.1 Konfigurasi pin LCD M1632 .....                        | 30        |
| 2.8 Motor DC .....   | 32        |
| Driver Motor DC dengan IC L293D .....                        | 33        |
| 2.9 Motor Servo .....  | 34        |
| 2.9.1 Konfigurasi Pin Motor Servo .....                      | 35        |
| 2.9.2 Prinsip kerja motor servo .....                        | 35        |
| 2.10 Sensor photodioda .....                                 | 37        |
| Prinsip kerja sensor photodioda .....                        | 38        |
| 2.11 Sensor Tegangan .....                                   | 39        |
| 2.12 IP camera .....   | 40        |
| <br>   |           |
| <b>BAB III PERANCANGAN SISTEM .....</b>                      | <b>43</b> |
| 3.1 Perancangan Mekanik Robot dan docking station .....      | 46        |
| 3.1.1 Perancangan Mekanik robot .....                        | 46        |
| 3.1.2 Perancangan docking station .....                      | 48        |
| 3.2 Perancangan Rotary Encoder .....                         | 48        |
| 3.3 Perancangan Motor DC dan driver motor .....              | 50        |
| 3.4 Perancangan Ultrasonic .....                             | 53        |
| 3.5 Perancangan LCD .....                                    | 54        |
| 3.6 Perancangan sensor tegangan dan baterai lipo .....       | 56        |
| 3.7 Perancangan motor servo .....                            | 59        |
| 3.8 Perancangan sensor photodioda .....                      | 60        |
| 3.9 Perancangan sistem controller dan interface sistem ..... | 62        |
| 3.10 Perancangan IP CAM .....                                | 63        |
| 3.11 Perancangan pergerakan robot .....                      | 64        |
| 3.12 Diagram Alir .....                                      | 69        |
| <br>   |           |
| <b>BAB IV PENGUJIAN DAN ANALISIS .....</b>                   | <b>71</b> |
| 4.1 Pengujian Rotary encoder .....                           | 71        |
| 4.2 Pengujian Motor DC .....                                 | 72        |
| 4.3 Pengujian Driver motor .....                             | 73        |
| 4.4 Pengujian Sensor ultrasonic .....                        | 73        |
| 4.5 Pengujian LCD .....                                      | 74        |
| 4.6 Pengujian sensor tegangan .....                          | 75        |
| 4.7 Pengujian motor servo .....                              | 75        |
| 4.8 Pengujian sensor photodioda .....                        | 76        |
| 4.9 Pengujian IP CAM .....                                   | 77        |

|   |     |
|---|-----|
| 4.10 Pengujian Sistem secara keseluruhan.....           | 78  |
| 4.10.1 Pengujian jalan robot menyusuri ruangan.....     | 78  |
| 4.10.2 Pengujian jalan robot menuju pulang.....         | 79  |
| 4.10.3 Pengujian jalan robot pada saat autodocking..... | 79  |
| <br>  |     |
| BAB V PENUTUP.....                                      | 83  |
| 5.1 Kesimpulan.....                                     | 83  |
| 5.2 Saran.....  | 83  |
| <br>  |     |
| DAFTAR PUSTAKA.....                                     | 85  |
| LAMPIRAN.....   | 89  |
| RIWAYAT PENULIS.....                                    | 113 |

## **TABLE OF CONTENTS**

|  |             |
|--|-------------|
| <b>TITLE PAGE</b>  |             |
| <b>FINAL DECLARATION OF AUTHENTICITY</b>                     |             |
| <b>ENDORSEMENT PAGE</b>                                      |             |
| <b>ABSTRAK</b> .....   | <i>i</i>    |
| <b>ABSTRACTK</b> .....                                       | <i>iii</i>  |
| <b>FOREWORD</b> .....  | <i>v</i>    |
| <b>DAFTAR ISI</b> .....                                      | <i>vii</i>  |
| <b>TABLE OF CONTENTS</b> .....                               | <i>xi</i>   |
| <b>LIST OF FIGURES</b> .....                                 | <i>xv</i>   |
| <b>LIST OF TABLES</b> .....                                  | <i>xvii</i> |
| <br>   |             |
| <b>CHAPTER I INTRODUCTION</b> .....                          | <i>1</i>    |
| 1.1 <i>Background</i> .....                                  | <i>2</i>    |
| 1.2 <i>Problem Formulation</i> .....                         | <i>2</i>    |
| 1.3 <i>Limitation Problem</i> .....                          | <i>2</i>    |
| 1.4 <i>Objective</i> .....                                   | <i>3</i>    |
| 1.5 <i>Research Methodology</i> .....                        | <i>3</i>    |
| 1.6 <i>Systematics Writing</i> .....                         | <i>5</i>    |
| 1.7 <i>Relevance</i> .....                                   | <i>6</i>    |
| <br>   |             |
| <b>CHAPTER II LITERATURE AND BASIC THEORY</b> .....          | <i>7</i>    |
| 2.1 <i>Robot</i> .....                                       | <i>7</i>    |
| 2.1.1 <i>The general classification robot</i> .....          | <i>7</i>    |
| 2.1.2 <i>Omniwheel robot</i> .....                           | <i>8</i>    |
| 2.2 <i>Battery Lipo</i> .....                                | <i>9</i>    |
| 2.2.1 <i>Voltage</i> .....                                   | <i>11</i>   |
| 2.2.2 <i>Capacity</i> .....                                  | <i>11</i>   |
| 2.2.3 <i>Discharge rate</i> .....                            | <i>12</i>   |
| 2.2.4 <i>Internal resistance</i> .....                       | <i>12</i>   |
| 2.3 <i>Charger dan balancer</i> .....                        | <i>13</i>   |
| 2.3.1 <i>Battery charging time</i> .....                     | <i>14</i>   |
| 2.3.2 <i>example autodocking process</i> .....               | <i>15</i>   |
| 2.4 <i>Mikrokontroler Arduino Mega 2560</i> .....            | <i>16</i>   |
| 2.4.1 <i>Block Diagram PIN function on Arduino kit</i> ..... | <i>17</i>   |
| 2.5 <i>Rotary Encoder</i> .....                              | <i>20</i>   |
| 2.5.1 <i>The principle of Rotary encoder</i> .....           | <i>20</i>   |
| 2.6 <i>Sensor ultrasonik</i> .....                           | <i>23</i>   |

|  |   |           |
|--|---|-----------|
| 2.6.1  | <i>Sensor ultrasonic HC-SR04</i> .....                            | 27        |
| 2.6.2  | <i>The working principle of sensor ultrasonic</i> .....           | 28        |
| 2.7  | <i>LCD M1632</i> .....  | 30        |
| 2.7.1  | <i>Configuration M1632 LCD pin</i> .....                          | 30        |
| 2.8  | <i>Motor DC</i> .....   | 32        |
|  | <i>Driver Motor DC dengan IC L293D</i> .....                      | 33        |
| 2.9  | <i>Motor Servo</i> .....  | 34        |
| 2.9.1  | <i>Pin Configuration Motor Servo</i> .....                        | 35        |
| 2.9.2  | <i>The working principle of servo motors</i> .....                | 35        |
| 2.10   | <i>Sensor photodiode</i> .....                                    | 37        |
|  | <i>The working principle of sensors photodiode</i> .....          | 38        |
| 2.11   | <i>Volatile Sensor</i> .....                                      | 39        |
| 2.12   | <i>IP camera</i> .....  | 40        |
| <br><b>CHAPTER III DESIGN SYSTEM</b> .....       |   | <b>43</b> |
| 3.1  | <i>Design Mechanical Robot and docking station</i> .....          | 46        |
| 3.1.1  | <i>Design of Mechanical robot</i> .....                           | 46        |
| 3.1.2  | <i>The design of the docking station</i> .....                    | 48        |
| 3.2  | <i>Design of Rotary Encoder</i> .....                             | 48        |
| 3.3  | <i>Design of Motor DC and driver motor</i> .....                  | 50        |
| 3.4  | <i>Design of Ultrasonic</i> .....                                 | 53        |
| 3.5  | <i>Design of LCD</i> .....  | 54        |
| 3.6  | <i>Design of voltage sensor and battery lipo</i> .....            | 56        |
| 3.7  | <i>Design of motor servo</i> .....                                | 59        |
| 3.8  | <i>Design of sensor photodiode</i> .....                          | 60        |
| 3.9  | <i>Design of the system controller and system interface</i> ..... | 62        |
| 3.10   | <i>Design of IP CAM</i> .....                                     | 63        |
| 3.11   | <i>The design of the movement of the robot</i> .....              | 64        |
| 3.12   | <i>Flowchart</i> .....  | 69        |
| <br><b>CHAPTER IV TESTING AND ANALYSIS</b> ..... |   | <b>71</b> |
| 4.1  | <i>Testing Rotary encoder</i> .....                               | 71        |
| 4.2  | <i>Testing Motor DC</i> .....                                     | 72        |
| 4.3  | <i>Testing Driver motor</i> .....                                 | 73        |
| 4.4  | <i>Testing Sensor ultrasonic</i> .....                            | 73        |
| 4.5  | <i>Testing LCD</i> .....  | 74        |
| 4.6  | <i>Testing voltage sensor</i> .....                               | 75        |
| 4.7  | <i>Testing motor servo</i> .....                                  | 75        |
| 4.8  | <i>Testing sensor photodiode</i> .....                            | 76        |



|   |            |
|---|------------|
| 4.9 Testing IP CAM.....                               | 77         |
| 4.10 Testing the overall system.....                  | 78         |
| 4.10.1 Testing the robot's way down the room.....     | 78         |
| 4.10.2 Testing the robot path toward home.....        | 79         |
| 4.10.3 Testing the robot at the time autodocking..... | 79         |
| <b>CHAPTER V CLOSING</b> .....                        | <b>83</b>  |
| 5.1 Conclusion.....                                   | 83         |
| 5.2 Suggestions.....                                  | 83         |
| <b>REFERENCES</b> .....                               | <b>85</b>  |
| <b>ATTACHMENT</b> .....                               | <b>89</b>  |
| <b>BIOGRAPHY</b> .....                                | <b>113</b> |

## DAFTAR GAMBAR

|             |   |    |
|-------------|---|----|
| Gambar 1.1  | Metodologi pengerjaan tugas akhir .....                                   | 3  |
| Gambar 2.1  | Mobile robot.....   | 8  |
| Gambar 2.2  | Jenis robot menggunakan roda standar dan yang menggunakan omniwheel ..... | 9  |
| Gambar 2.3  | Baterai LiPo.....   | 11 |
| Gambar 2.4  | Charger dan Balancer .....  | 14 |
| Gambar 2.5  | Foto plat pada bagian bawah robot football .....                          | 16 |
| Gambar 2.6  | Board arduino mega2560 .....  | 17 |
| Gambar 2.7  | Diagram Blok KIT arduino .....  | 17 |
| Gambar 2.8  | Rotary encoder type E40H-A-360-24.....                                    | 20 |
| Gambar 2.9  | Sistem blok penyusunan rotary encoder .....                               | 21 |
| Gambar 2.10 | Susunan piringan untuk incremental rotary encoder ..                      | 22 |
| Gambar 2.11 | Contoh pola keluaran incremental rotary encoder .....                     | 22 |
| Gambar 2.12 | Output dan arah putaran pada resolusi yang berbeda beda.....              | 23 |
| Gambar 2.13 | Prinsip kerja sensor ultrasonik .....                                     | 26 |
| Gambar 2.14 | Prinsip pemantulan gelombang ultrasonik .....                             | 27 |
| Gambar 2.15 | pemantulan sensor ultrasonik .....  | 28 |
| Gambar 2.16 | Bentuk fisik HC-SR04.....   | 28 |
| Gambar 2.17 | Timing diagram HC-SR04.....   | 29 |
| Gambar 2.18 | Pin LCD M1632 .....   | 30 |
| Gambar 2.19 | Konstruksi motor DC .....   | 32 |
| Gambar 2.20 | Motor DC dan jembatan H .....   | 34 |
| Gambar 2.21 | Konfigurasi pin pada motor servo .....                                    | 35 |
| Gambar 2.22 | Bentuk sinyal pengontrolan motor servo.....                               | 36 |
| Gambar 2.23 | Skematis pengiriman pulsa .....   | 37 |
| Gambar 2.24 | Percobaan warna garis dan lintasan.....                                   | 38 |
| Gambar 2.25 | Cahaya pemantulan sedikit dan cahaya pemantulan banyak .....              | 39 |
| Gambar 2.26 | Bentuk fisik sensor tegangan.....   | 39 |
| Gambar 2.27 | Blok diagram IP camera .....  | 40 |
| Gambar 2.28 | Blok diagram I/O.....   | 40 |
| Gambar 3.1  | Diagram blok sistem mobile robot .....                                    | 43 |
| Gambar 3.2  | (a)Design mekanik robot tampak samping.....                               | 47 |
|             | (b)Design mekanik robot tampak atas .....                                 | 47 |
|             | (c)Design mekanik robot tampak bawah .....                                | 47 |

|             |  |    |
|-------------|--|----|
| Gambar 3.3  | Design <i>docking station</i> .....                                    | 48 |
| Gambar 3.4  | Flowchart forward dan reverse rotary encoder .....                     | 49 |
| Gambar 3.5  | Hubungan antara rotary encoder dengan mikrokontroler .....             | 50 |
| Gambar 3.6  | Hubungan antara driver motor dengan mikrokontroler .....               | 51 |
| Gambar 3.7  | Hubungan antara driver motor dengan Motor DC.....                      | 52 |
| Gambar 3.8  | Hubungan antara ultrasonic dengan mikrokontroler ..                    | 53 |
| Gambar 3.9  | Flowchart pembacaan jarak sensor ultrasonic .....                      | 54 |
| Gambar 3.10 | Hubungan antara LCD dengan mikrokontroler .....                        | 55 |
| Gambar 3.11 | Hubungan antara baterai Lipo, sensor tegangan dan mikrokontroler ..... | 56 |
| Gambar 3.12 | Hubungan antara motor servo dan mikrokontroler.....                    | 59 |
| Gambar 3.13 | Flowchart pergerakan pada motor servo.....                             | 60 |
| Gambar 3.14 | Hubungan antara sensor photodiode dengan mikrokontroler .....          | 61 |
| Gambar 3.15 | Diagram alir pembacaan sensor photodiode.....                          | 61 |
| Gambar 3.16 | Diagram alir hubungan antara ip cam dengan PC .....                    | 63 |
| Gambar 3.17 | Tampilan IP cam pada web browser .....                                 | 64 |
| Gambar 3.18 | tempat simulasi robot .....  | 65 |
| Gambar 3.19 | Ilustrasi dari pergerakan robot.....                                   | 66 |
| Gambar 3.20 | tempat docking station.....  | 67 |
| Gambar 3.21 | tempat percobaan robot .....   | 67 |
| Gambar 3.22 | diagram alir sistem pergerakan robot .....                             | 69 |
| Gambar 3.23 | diagram alir sistem saat menuju <i>docking station</i> .....           | 70 |
| Gambar 4.1  | Hasil pengujian LCD.....   | 74 |
| Gambar 4.2  | Tampilan pengujian ip cam pada web browser.....                        | 77 |
| Gambar 4.3  | Ilustrasi dari pergerakan robot.....                                   | 78 |
| Gambar 4.4  | tempat simulasi percobaan robot.....                                   | 81 |
| Gambar 4.5  | tempat <i>docking station</i> .....                                    | 82 |

## DAFTAR TABEL

|            |   |    |
|------------|---|----|
| Tabel 2.1  | kecepatan bunyi dalam medium jaringa .....            | 24 |
| Tabel 2.2  | Konfigurasi pin LCD M1632 .....                       | 31 |
| Tabel 3.1  | Hubungan antar pin rotary encoder .....               | 49 |
| Tabel 3.2  | Hubungan antar pin driver motor L298d .....           | 51 |
| Tabel 3.3  | Hubungan antar motor DC .....                         | 52 |
| Tabel 3.4  | Hubungan antar sensor ultrasonic .....                | 53 |
| Tabel 3.5  | Hubungan antar LCD .....                              | 56 |
| Tabel 3.6  | Hubungan antar sensor tegangan dan baterai LiPO ..... | 57 |
| Tabel 3.7  | Hubungan antar motor servo .....                      | 59 |
| Tabel 3.8  | Hubungan antar sensor photodiode .....                | 61 |
| Tabel 4.1  | Hasil pengujian rotary encoder .....                  | 71 |
| Tabel 4.2  | Pengujian motor DC terhadap arah putaran .....        | 72 |
| Tabel 4.3  | Pengujian PWM motor DC .....                          | 73 |
| Tabel 4.4  | Pengujian driver motor .....                          | 73 |
| Tabel 4.5  | Pengujian sensor ultrasonic .....                     | 74 |
| Tabel 4.6  | Pengujian sensor tegangan .....                       | 75 |
| Tabel 4.7  | Pengujian motor servo .....                           | 76 |
| Tabel 4.8  | Pengujian sensor photodiode .....                     | 76 |
| Tabel 4.9  | Pengujian robot saat menyusuri ruangan .....          | 79 |
| Tabel 4.10 | Pengujian robot saat menuju pulang .....              | 80 |

# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Teknologi robot semakin berkembang. Pada negeara-negara maju teknologi robot menjadi salah satu teknologi yang terus direset. Salah satu negara maju yang mengembangkan teknologi robot, diantaranya Jepang dengan salah satu product robotnya adalah Asimo(*advanced step in innovative mobility*) yang diperkenalkan pertamakali oleh honda pada tahun 2000. Asimo didesain dapat beradaptasi di dunia nyata dan mampu berjalan dengan kedua kakinya, SOINN(*Self-Organizing Incremental Neural Network*) adalah robot pintar yang sedang dikembangkan oleh *Tokyo Institute Of Technology*. SOINN merupakan robot pintar yang dapat berpikir dengan sendirinya. Kemudian negara berkembang yang melakukan riset dan mengembangkan teknologi robot adalah China dengan salah satu product robotnya *Rover Yutu*("Jade Rabbit") merupakan robot yang betugas untuk mengeksplorasi keadaan bulan. Sebagai tempat tinggal seseorang tentu didalamnya terdapat banyak sekali barang yang harus dijaga keamanannya. Salah satu alat elektronik yang banyak digunakan dewasa ini adalah CCTV (*Closed-Circuit Television*). Tetapi bagi kalangan tertentu tidak semua orang dapat memanfaatkan piranti ini dikarenakan kurang canggih dan tidak bisa mengirimkan data secara akurat. Disamping itu bila mempunyai robot untuk *mobile* robot tidak ada pengisian baterai secara otomatis. Mulai dari rumah tangga, industri, medis sampai pada militer hampir mempunyai *mobile* robot. Kelemahan pada robot pada umumnya adalah sumber energi yang akan habis apabila terus digunakan. Sumber energi atau yang biasa disebut baterai perlu terus diganti ketika baterai tersebut habis. Hal ini menyebabkan seringnya terjadi pergantian baterai dan pemborosan energi dan biaya. Pada penelitian *mobile* Robot sudah banyak dilakukan penelitian, tentang pengisian baterai otomatis atau yang disebut juga *auto docking*. Pada robot tentunya ada sebuah tempat pengisian atau yang disebut juga *docking station*. Oleh karena itu sebagai mahasiswa perguruan tinggi di Indonesia(*Institut Teknologi Sepuluh Nopember*) diperlukan keikutsertaan dalam riset dan pengembangan robotika maka dalam Tugas Akhir ini berjudul "*Implementasi Auto Docking System Pada Mobile Robot*". Ini

merupakan robot dalam katagori *mobile* robot yaitu robot bergerak. Prinsip kerja Robot ini dapat dipakai hanya untuk proses mobile robot dan pengisian baterai secara otomatis saja. Dalam aplikasi ini, Robot akan melakukan pencarian dalam map, yang sudah ditentukan. Bila mendeteksi pergerakan manusia maka robot akan mengirimkan data. Apabila daya robot sudah habis maka robot akan melakukan proses *auto docking*.

## 1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah :

1. Bagaimana robot yang dapat melakukan tugas navigasi menggunakan incremental Rotary Encoder pada saat bergerak menelusuri ruangan.
2. Bagaimana robot dapat bergerak ke *docking station* secara otomatis untuk melakukan proses *charger* baterai pada robot.
3. Bagaimana robot dengan menggunakan IP cam pada saat menelusuri ruangan.

## 1.3. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

1. Pada saat menggunakan incremental rotary encoder, apabila baterai habis maka jalur yang ditempuh robot untuk kembali sama dengan saat robot melakukan menelusuri ruangan.
2. Pada saat di *docking station* robot membutuhkan waktu yang lama dikarenakan, robot melakukan proses *penchargeran* pada baterai.
3. pada saat menggunakan rotary encoder, jalur robot saat menelusuri alas sudah ditentukan sudah ditentukan
4. pada saat robot menelusuri tempat yang sudah disediakan landasan pergerakan robot harus datar, hal ini dikarenakan agar jalan robot tidak berbelok belok

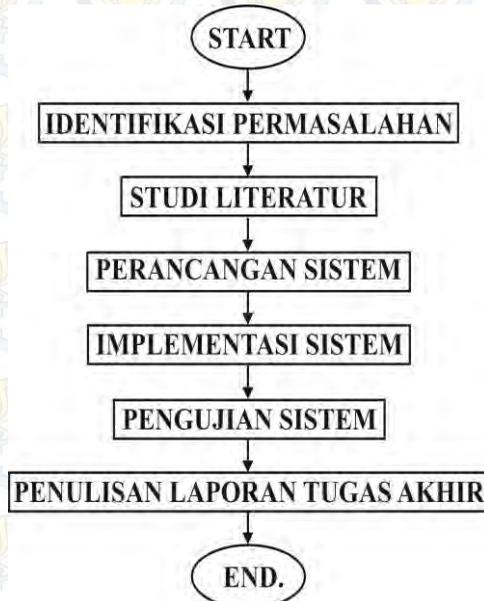
#### 1.4. Tujuan

Penelitian pada tugas akhir ini bertujuan sebagai berikut :

1. Mendesain sistem robot yang dapat melakukan tugas navigasi menggunakan incremental Rotary Encoder hasil dari proses navigasi menggunakan incremental Rotary Encoder yang dilakukan oleh robot pada saat bergerak menelusuri ruangan.
2. Mendesain sistem agar robot dapat bergerak ke *docking station* secara otomatis hasil dari pergerakan robot ke *docking station* secara otomatis digunakan untuk melakukan proses *charger* baterai pada robot.

#### 1.5. Metodologi

Langkah-langkah untuk mengerjakan tugas akhir ini adalah sebagai berikut :



**Gambar 1.1** Metodologi pengerjaan tugas akhir.

## IDENTIFIKASI PERMASALAHAN

Tahap ini merupakan proses penelusuran terhadap permasalahan-permasalahan yang terdapat dalam *auto docking* robot serta aplikasinya untuk bergerak menelusuri denah yang sudah disediakan dan kembali ke *docking station* secara otomatis. Dimana penelusuran tempat yang sudah disediakan menggunakan navigasi dari rotary encoder dan untuk kembali ke *docking station* menggunakan sensor photo dioda .

## STUDI LITERATUR

Hal-hal yang dipelajari dalam tahap studi literatur, meliputi :

- Mempelajari penggunaan sensor ultrasonic untuk mengukur jarak.
- Mempelajari penggunaan rotary encoder untuk navigasi robot.
- Mempelajari pergerakan dari robot yang menggunakan motor DC.
- Mempelajari penggunaan sensor tegangan untuk mengetahui tegangan pada baterai.
- mempelajari penggunaan pada motor servo untuk pergerakan dari tempat ip cam untuk bergerak secara per 90°.
- mempelajari penggunaan LCD untuk memonitoring tegangan dan pulsa pada setiap incremental rotary encoder.
- mempelajari penggunaan sensor photo dioda agar robot bergerak secara otomatis kembali ke *docking station*.
- mempelajari penggunaan baterai lippo yang digunakan sebagai sumber dari robot
- mempelajari *power supply* dan *charger balancer* untuk pengisian pada baterai pada robot

## PERANCANGAN SISTEM

Sistem yang dirancang memiliki dua bagian utama, yaitu sitem pemetaan navigasi robot dan kembali ke *docking station* secara otomatis. Pada navigasi robot menggunakan rotary encoder untuk bergerak menelusuri tempat yang sudah disediakan dengan memanfaatkan perhitungan pulsa. Dan kembali ke *docking station* menggunakan sensor photo dioda, dimana pada *docking station* akan di tempelkan landasan trek berwarna hitam agar robot bisa kembali ke *docking station* secara presisi.

## PENGUJIAN SISTEM

Pengujian sistem ini meliputi pengujian sensor ultrasonic, pengujian rotary encoder, pengujian motor DC, pengujian sensor tegangan, pengujian *power supply* sebagai *charger* baterai, pengujian motor DC, pengujian LCD, pengujian motor servo, dan hasil pergerakan robot otomatis yang bergerak pada tempat yang sudah disediakan,

## PENULISAN LAPORAN TUGAS AKHIR

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang dikerjakan, meliputi pendahuluan, tinjauan pustaka, perancangan sistem, pengujian dan penutup.

### 1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat tebagi menjadi lima bab dengan sistematika sebagai berikut :

#### BAB I : PENDAHULUAN

Pada bagian ini menjelaskan tentang dasar penyusunan tugas akhir yang terdiri dari beberapa sub bagian, yaitu latar belakang, permasalahan dan batasannya, tujuan yang diharapkan, metodologi pengerjaan tugas akhir, relevansi serta sistematika penulisan tugas akhir ini.

#### BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bagian ini menjelaskan teori penunjang dan beberapa literatur yang berguna bagi pembuatan tugas akhir ini, yang meliputi robot, batterai lipo, *Power Supply* dan *Charger balancer*, Arduino Mega 2560, incremental rotary encoder, Motor DC, ultrasonic HC-SR04, Sensor tegangan, LCD 16x2, Motor Servo, Sensor Photo Dioda dan ip cam

#### BAB III : PERANCANGAN SISTEM

Pada bagian ini berisi perancangan sistem yang terdiri dari, perancangan mekanik robot, perancangan mekanik *Docking Station*, perancangan perangkat lunak sistem kontroller, perancangan motor DC, perancangan driver motor, perancangan sensor ultrasonic, perancangan perangkat rotary encoder, perancangan sensor tegangan, perancangan sensor photo dioda, perancangan LCD, perancangan perangkat motor servo dan perancangan perancangan perangkat lunak ip cam

#### BAB IV : PENGUJIAN

Pada bagian ini akan memberikan hasil-hasil pengujian yang meliputi, hasil pengujian menggunakan sensor ultrasonic, hasil pengujian menggunakan rotary encoder, hasil pengujian menggunakan motor DC, hasil pengujian menggunakan sensor tegangan, hasil pengujian menggunakan *power supply* sebagai *charger* baterai, hasil pengujian menggunakan motor DC, hasil pengujian menggunakan LCD, hasil pengujian menggunakan motor servo, hasil pengujian IP Cam dan hasil pergerakan robot otomatis yang bergerak pada tempat yang sudah disediakan.

## BAB V : PENUTUP

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

### 1.7. Relevansi

Mata kuliah yang mendukung dalam pembuatan tugas akhir ini adalah sistem mikroprosesor dan mikrokontroler yang mendukung dalam pemrogramannya juga dalam pemrograman LCD dan motor servo kemudian sistem perancangan rangkaian elektronika analog yang mendukung dalam pembuatan rangkaian driver motor, rangkaian sensor tegangan dan rangkaian *power supply* kemudian sensor dan transduser yang mendukung dalam mempelajari sensor ultrasonic dan sensor photodiode kemudian elektronika industri yang mendukung dalam mempelajari incremental rotary encoder dan motor DC dan topik khusus elektronika yang mendukung dalam pembuatan program untuk pengolahan gambar, serta beberapa pengetahuan yang diperoleh dari dosen pembimbing dan referensi-referensi berupa handbook serta beberapa jurnal yang mendukung untuk menyelesaikan tugas akhir ini.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Robot

Istilah robot berasal dari kata *robot* (bahasa Czech) yang berarti bekerja. Kamus Webster mendefinisikan robot sebagai “perangkat *automatik* yang menunjukkan fungsi yang dapat dilakukan manusia”[1]. Robot Institute of America mendeskripsikan istilah robot sebagai berikut: “Suatu robot adalah manipulator multi fungsi yang dirancang untuk memindahkan benda, peralatan atau perangkat khusus dengan pergerakan yang dapat diprogram secara berulang untuk menunjukkan berbagai macam tugas[2].

#### 2.1.1 Klasifikasi Umum Robot

Berdasarkan fungsinya robot dapat dikelompokkan menjadi dua bagian, yaitu:

##### 1. *Service* robot

*Service* robot adalah robot yang berfungsi untuk mempermudah manusia dalam melakukan tugasnya seperti membersihkan rumah, membantu dokter dalam melakukan operasi dan lain-lain.

##### 2. *Mobile* robot

*Mobile* robot adalah robot yang memiliki ruang kerja yang luas. Dikatakan luas, karena bagian dasar dari robot ini diletakkan alat untuk bergerak, seperti roda atau kaki. Bentuk dari *mobile* robot dapat dilihat pada gambar 2.1

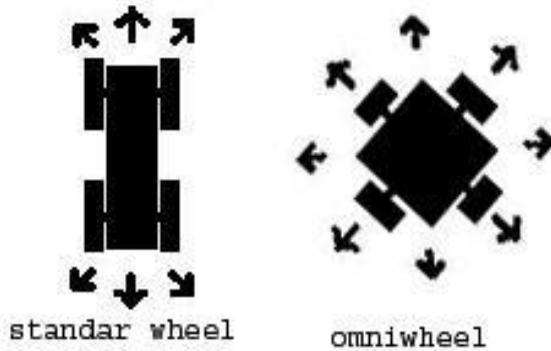


**Gambar 2.1** Mobile robot

### **2.1.2 Omniwheel Robot**

Roda baik standar atau roda *omniwheel* membuat perbedaan yang besar hanya ada dua jenis mobil robot, robot *omniwheel* dan robot *non-omniwheel*. Robot *non-omniwheel* adalah robot yang tidak bisa bergerak ke segala arah, seperti mobil pada gambar 2.2 Jenis robot untuk melakukan serangkaian gerakan untuk mengubah posisi. Pada mobil standar wheel, Jika anda ingin mobil Anda untuk bergerak ke samping, Anda harus melakukan 'gerak kompleks paralel' parkir. Anda harus memutar roda dan berkendara ke depan. Jenis roda robot akan '1 .5' derajat kebebasan, yang berarti bahwa ia tidak dapat bergerak baik dalam arah X dan Y, tetapi membutuhkan gerakan kompleks untuk mencapai arah X. [3]

Sebuah robot *omniwheel* pada gambar 2.1 mobil dapat bergerak ke segala arah. Ini tidak perlu melakukan gerakan yang kompleks untuk mencapai posisi tertentu. Jenis robot akan memiliki 2 derajat kebebasan karena dapat bergerak baik di X dan Y bebas pesawat. Sebuah lengan robot biasanya adalah *omniwheel* juga, dengan lebih banyak derajat kebebasan. [4]



**Gambar 2.2** jenis robot menggunakan roda standar dan yang menggunakan omniwheel

Keuntungan dari roda *omniwheel* adalah memiliki kemampuan pergerakan yang bebas, sedangkan roda standar memiliki pergerakan yang terbatas. Pengaturan posisi roda mempengaruhi pergerakan robot secara signifikan, jika dengan roda standar semakin jauh jarak roda depan dengan roda belakang maka semakin sulit untuk memutar posisi robot. Dengan menggunakan empat buah roda omniwheel maka di posisikan secara merata di ke empat sisi robot

Kekurangan dari roda *omniwheel*, roda omniwheel kurang efisiensi karena tidak semua roda berputar dalam arah gerak robot dan roda juga memiliki gaya gesek yang cukup besar.

## 2.2 Baterai Lipo

Baterai Lithium Polimer atau biasa disebut dengan LiPo merupakan salah satu jenis baterai yang sering digunakan dalam dunia RC. Utamanya untuk RC tipe pesawat dan helikopter. Tiga kelebihan yang ditawarkan oleh baterai berjenis LiPo dibanding baterai jenis NiCad atau NiMH yaitu:

1. Baterai LiPo memiliki bobot ringan dan dalam berbagai macam bentuk dan ukuran,
2. Baterai LiPo memiliki kapasitas penyimpanan energi listrik yang besar,

3. Baterai LiPo memiliki tingkat *discharge* rate energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang RC.

Selain keuntungan baterai jenis ini juga memiliki beberapa kelemahan yaitu:

1. Harga baterai LiPo tergolong mahal dibandingkan dengan baterai jenis NiCad dan NiMH,
2. Performa yang tinggi dari baterai LiPo harus dibayar dengan umur yang lebih pendek. Usia baterai LiPo sekitar 300-400 kali siklus pengisian ulang. Sesuai dengan perlakuan yang diberikan pada baterai.
3. Baterai LiPo menggunakan bahan elektrolit yang mudah terbakar.
4. Baterai LiPo membutuhkan penanganan khusus agar dapat bertahan lama. Charging, Discharging, maupun penyimpanan dapat mempengaruhi usia dari baterai jenis ini.

Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate. Masalah ini sebenarnya bisa diatasi dengan memanasakan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari. Seandainya para ilmuwan dapat memecahkan masalah ini maka risiko keamanan pada baterai jenis lithium akan sangat berkurang.



**Gambar 2.3** Baterai LiPo

### **2.2.1 Tegangan (Voltage)**

Pada baterai jenis NiCad atau NiMH tiap sel memiliki 1,2 volt sedangkan pada baterai Lipo memiliki rating 3,7 volt/sel. Keuntungannya adalah tegangan baterai yang tinggi dapat dicapai dengan menggunakan jumlah sel yang lebih sedikit.

Pada setiap paket baterai LiPo selain tegangan ada label yang disimbolkan dengan “S”. Disini “S” berarti sel yang dimiliki sebuah paket baterai (*battery pack*). Sementara bilangan yang berada didepan simbol menandakan jumlah sel dan biasanya berkisar antar 2-6S (meskipun kadang ada yang mencapai 10S). Berikut adalah beberapa contoh notasi baterai LiPo.

3.7 volt battery = 1 cell x 3.7 volts

7.4 volt battery = 2 cells x 3.7 volts (2S)

11.1 volt battery = 3 cells x 3.7 volts (3S)

14.8 volt battery = 4 cells x 3.7 volts (4S)

18.5 volt battery = 5 cells x 3.7 volts (5S)

22.2 volt battery = 6 cells x 3.7 volts (6S)

### **2.2.2 Kapasitas (Capacity)**

Kapasitas baterai menunjukkan seberapa banyak energi yang dapat disimpan oleh sebuah baterai dalam miliampere *hours* (mAh). Notasi ini adalah cara lain untuk mengatakan seberapa banyak beban yang dapat diberikan kepada sebuah baterai selama 1 jam,

dimana setelah 1 jam baterai akan benar-benar habis.

Sebagai contoh sebuah baterai RC LiPo yang memiliki rating 1000 mAh akan benar-benar habis apabila diberi beban sebesar 1000 miliampere selama 1 jam. Apabila baterai yang sama diberi beban 500 miliampere, maka baterai akan benar-benar habis setelah selama 2 jam. Begitu pun apabila beban ditingkatkan menjadi 15.000 miliampere (15 Amps) maka energi di dalam baterai akan habis terpakai setelah selama 4 menit saja. (15 Amp merupakan jumlah beban yang umum digunakan pada RC kelas 400). Seperti yang telah

dijelaskan, dengan beban arus yang begitu besar maka merupakan sebuah keuntungan apabila menggunakan baterai dengan kapasitas yang lebih besar (misal 2000 mAh). Dengan begitu maka waktu *discharge* akan meningkat menjadi 8 menit

### 2.2.3 Discharge Rate

*Discharge rate* biasa disimbolkan dengan “C” merupakan notasi yang menyatakan seberapa cepat sebuah baterai untuk dapat dikosongkan (*discharge*) secara aman. Sesuai dengan penjelasan diatas bahwa energi listrik pada baterai LiPo berasal dari pertukaran ion dari anoda ke katoda. Semakin cepat pertukaran ion yang dapat terjadi maka berarti semakin nilai dari “C”.

Sebuah baterai dengan *discharge rate* 10C berarti baterai tersebut dapat di *discharge* 10 kali dari kapasitas baterai sebenarnya. begitu juga 15C berarti 15 kali, dan 20C berarti 20 kali. dsb.

Mari gunakan contoh baterai 1000 mAh diatas sebagai contoh. Jika baterai tersebut memiliki rating 10C maka berarti baterai tersebut dapat menahan beban maksimum hingga 10.000 miliampere atau 10 Ampere. ( $10 \times 1000$  miliampere = 10 Ampere). Angka ini berarti sama dengan 166 mA per menit, maka energi baterai 1000 mAh akan habis dalam 6 menit. Angka ini berasal dihitung dengan mengkalkulasi jumlah arus per menitnya. 1000 mAh dibagi 60 menit = 16,6 mA per menit. Lalu kemudian kalikan 16,6 dengan C rating (dalam hal ini 10) = 166 mA beban per menit. Lalu bagi 1000 dengan 166 = 6,02 menit.

### 2.2.4 Hambatan Dalam (*Internal Resistance*)

Hambatan dalam (*Internal Resistance*) adalah bilangan yang menyatakan nilai tahanan yang ada didalam komponen baterai. Hambatan ini akan menentukan kecepatan pertukaran ion dari anoda ke katoda.[5]

### 2.3 Charger dan balancer

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah robot. Tanpa bagian ini robot tidak akan berfungsi. Begitu juga bila pemilihan catu daya tidak tepat, maka robot tidak akan bekerja dengan baik.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh banyak faktor, diantaranya :

1. Tegangan

Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap disain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

2. Arus

Arus memiliki satuan Ah (*Ampere-hour*).Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

3. Teknologi Baterai

Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong, dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong. [6]

Banyak jenis baterai yang bisa dipakai untuk mensuplai tegangan untuk robot antara lain (Alkaline, Fuel Cell, Lead Acid,Lithium, NiCad, NiMH, baterai

*Lithium Polymer* (LIPO)). Namun catu daya yang digunakan dalam robot ini menggunakan baterai *Lithium Polymer* (LIPO). Daya yang dimiliki baterai lipo adalah 3,7 V adalah tegangan nominal 1 cell dari kimia LIPO , Tegangan sebenarnya dapat dinaikan sampai pada pengisian ( charging ) maksimum pada 4,25 V dan dapat diturunkan sampai minimum 3,0 V. Serta berat beban baterai lipo cukup ringan di bandingkan jika menggunakan baterai yang jenis lain.

Untuk baterai 2 cell volt yang dibuat SERI adalah  $2 \times 3,7 \text{ V} = 7,4$

Untuk baterai 3 cell volt yang dibuat SERI adalah  $3 \times 3,7 \text{ V} = 11,1$

Untuk baterai 3 cell volt yang dibuat SERI adalah  $4 \times 3,7 \text{ V} = 14,8 \text{ V}$

Untuk baterai 6 cell tegangan yang dibuat SERI hanya maksimum adalah 3 cell yang kemudian di paralel agar hanya memiliki maksimum tegangan 11,1 V. Total Ampere akan mengikuti pembuatan SERI dan PARAREL dari baterai tersebut. Range dengan minimum 3,7 V sampai pada maksimum 4,2 V adalah range yang paling baik digunakan untuk menjaga kemampuan kimiawi dari cell baterai LIPO tersebut agar tidak mudah rusak. Oleh karena itu normalnya alat-alat *charger* LIPO akan otomatis melakukan cut off charging pada 4,2 V per cellnya. Kelemahannya adalah *charger* LIPO tidak dapat melakukan pembagian pengisian arus setiap cellnya dengan rata. [7]



**Gambar 2.4** Charger dan balancer

### 2.3.1 Waktu Pengisian Baterai

Lamanya pengisian baterai ditentukan oleh kapasitas baterai dan juga besarnya arus yang digunakan untuk pengisian. Namun pada umumnya arus pengisian dibatasi sebesar 5 sampai 10 % dari kapasitas arus yang dapat dikeluarkan oleh baterai. Hal ini dilakukan untuk menghindari kemungkinan baterai cepat masak.

untuk *charging time* pada masing-masing jenis alat *charge* sebenarnya mempunyai perhitungan dasar ideal yang dapat dihitung dengan menggunakan persamaan[8] (2-1).

$$t_h = \text{mahB} / \text{mAC} \quad (2-1)$$

Keterangan:

mahB = kapasitas maksimum baterai

mAC = besarnya *ampere* yang diberikan *charger*

$t_h$  = total waktu dalam jam

Contoh: Jika baterai berkapasitas 1800 mAh dan *Ampere Chargernya* 100 mA, berarti  $1800 / 100 = 18$  jam.

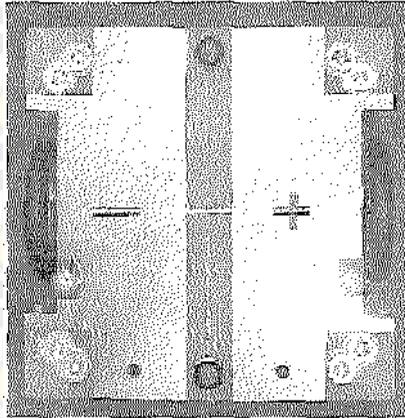
Waktu yang diperlukan untuk *chargingnya* pada kondisi ideal adalah 18 jam

Dari contoh di atas terlihat bahwa proses pengisian baterai terlalu lama. Pada beberapa baterai terdapat arus pengisian maksimum. jadi tidak terlalu berpatokan pada prinsip pengisian di atas, seperti harus mengisi 5% sampai 100% dari kapasitas arus baterai. Baterai dapat diisi dengan arus pengisian maksimum dari baterai, sehingga waktu pengisian tidak terlalu lama.

### 2.3.2 Contoh proses *autodocking*

*Robot Football* yang merupakan salah satu proyek *engineering university of Straihclyde*[9]. Pada robot ini terdapat dua bagian yang digunakan dalam proses pengisian baterai secara otomatis yaitu pertama. plat yang terdapat di bagian bawah robot yang masing-masing diberi tanda '+' dan '-', seperti yang ditunjukkan pada Gambar 2.4 yang digunakan untuk mengisi baterai. Dan yang kedua adalah *switch* yang terdapat di *Printed Circuit Board (PCB)* memiliki posisi *charge* yang digunakan untuk menggabungkan *charger plat* ke baterai melalui *charging port*. Dalam proyek ini telah diyakini bahwa mekanisme pengisian baterai dapat melakukan pengisian baterai tanpa harus memindahkan baterai dari robot. Namun pada robot

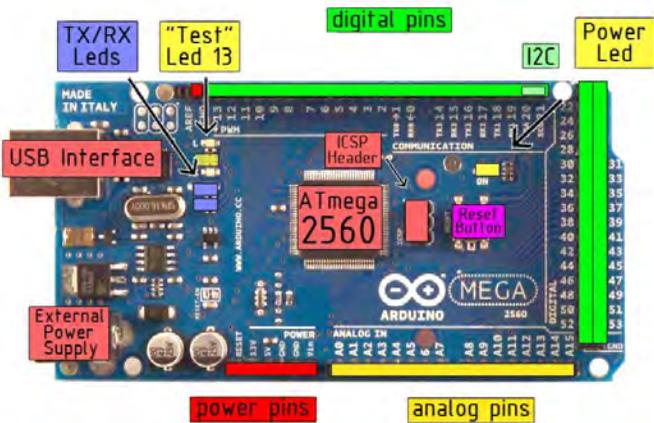
ini belum ada mekanisme *feedback* dari status pengisian dan penggunaan dari plat ini ditunjukkan untuk penggunaan yang akan datang.



**Gambar 2.5** Foto plat pada bagian bawah Robot *Football*

#### **2.4 Mikrokontroler Arduino Mega2560**

Arduino mega2560 adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino adalah sebuah board mikrokontroller yang berbasis ATmega2560. Arduino memiliki 54 pin input/output yang mana 14 pin dapat digunakan sebagai output PWM, 16 analog input, crystal osilator 16 MHz, koneksi USB, jack *power*, kepala ICSP, dan tombol reset. Arduino mampu *men-support* mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB. Berikut gambar 2.6 adalah pin-pin pada kit arduino uno yang digunakan pada rancangan alat ini:

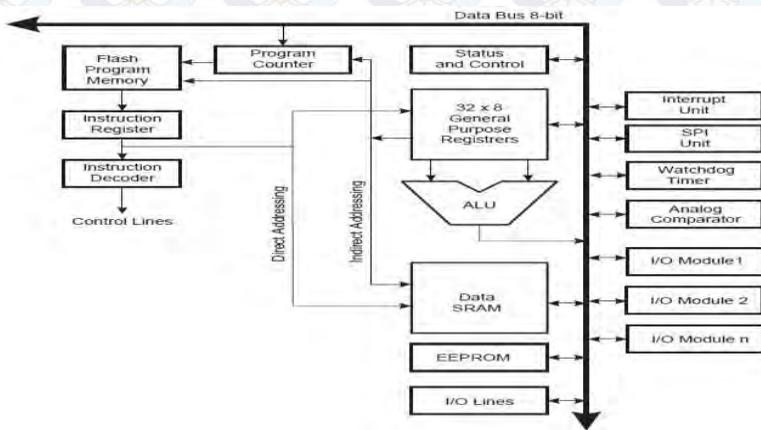


**Gambar 2.6** Board arduino mega2560

Pada gambar 2.6[10] terdapat 54 pin output/input yang mana 14 pin dapat digunakan sebagai output PWM, 16 analog input, kristal osilator 16 MHz dan tombol reset. Arduino tersebut digunakan sebagai chip mikrokontroler ATmega2560, sebagai pengendali gerakan motor DC.

### 2.4.1 Diagram Blok dan Fungsi PIN Pada Kit Arduino

Berikut gambar 2.4 adalah bentuk diagram blok dari kit arduino:



### **Gambar2.7.** Diagram Blok KIT arduino

Fungsi PIN pada kit Arduino uno pada gambar 2.7 adalah sebagai berikut:

- **PIN Power**

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Powernya* diselek secara otomatis. *PIN power* terdapat pada kaki 1 sampai kaki 6

- **Power Supply**

*Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok jack adaptor pada koneksi port input *supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada board. Rekomendasi tegangan ada pada 7 sampai 12 volt

**Penjelasan pada pin *power* adalah sebagai berikut :**

- **Vin**

Tegangan input ke board arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.

- **5V**

Regulasi *power supply* digunakan untuk *power* mikrokontroler dan komponen lainnya pada board. 5V dapat melalui Vin menggunakan regulator pada board, atau supply oleh USB atau *supply* regulasi 5V lainnya.

- **3V3**

Suplai 3.3 volt didapat oleh FTDI chip yang ada di board. Arus maximumnya adalah 50mA

- **Pin Ground**

berfungsi sebagai jalur ground pada arduino

- **Memori**

ATmega328 memiliki 32 KB flash memori untuk menyimpan kode,

juga 2 KB yang digunakan untuk bootloader. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM. Input dan Output Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20- 50 KOHms.

### **Beberapa pin memiliki fungsi sebagai berikut :**

- Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB FTDI ke TTL chip serial.
- Interupt eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi analogWrite().
- LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

#### **• Konektor USB**

Konektor USB adalah soket untuk kabel USB yang disambungkan ke komputer atau laptop. Berfungsi untuk mengirimkan program ke Arduino dan juga sebagai port komunikasi serial.

#### **• Input / output digital**

Input / output digital adalah pin pin untuk menghubungkan Arduino dengan komponen atau rangkaian digital. Input/Output didital pada KIT arduino terdapat pada kaki 1 samapai kaki 13. Misalnya kalau ingin membuat LED berkedip, LED tersebut bisa dipasang pada salah satu pin I/O digital dan ground. Komponen lain yang menghasilkan output digital atau menerima input digital bisa disambungkan ke pin-pin ini.

#### **• Input Analog**

Input Analog atau analog pin adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog. Misalnya dari potensiometer, sensor suhu, sensor cahaya, dsb.

#### **• Baterai /Adaptor**

Soket baterai atau adaptor digunakan untuk menyuplai Arduino

dengan tegangan dari baterai/adaptor 9V pada saat Arduino sedang tidak disambungkan ke komputer. Kalau Arduino sedang disambungkan ke komputer melalui USB, Arduino mendapatkan suplai tegangan dari USB, jadi tidak perlu memasang baterai/adaptor saat memprogram Arduino.

## 2.5 Rotary Encoder

*Rotary encoder* adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian *robot*, *motor drive* dan sebagainya. Bentuk *rotary encoder type E40H-A-360-24* dapat dilihat pada gambar 2.8

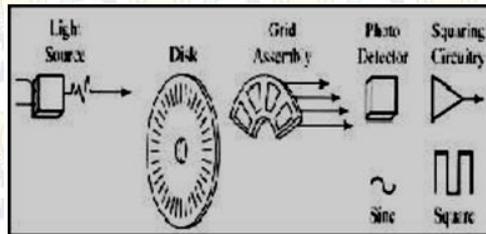


**Gambar 2.8** Rotary Encoder Type E40H-A-360-24

### 2.5.1 Prinsip Kerja Rotary Encoder

*Rotary encoder* tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. *LED* ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju piringan. Di sisi yang lain suatu *photo-transistor* diletakkan sehingga *photo-transistor* ini dapat mendeteksi cahaya dari *LED* yang berseberangan. Piringan tipis tadi dipasangkan dengan poros motor, atau perangkat berputar lainnya yang ingin kita ketahui posisinya sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari *LED* dapat mencapai *photo-transistor* melalui lubang-lubang yang ada, maka *photo-transistor* akan

mengalami *saturasi* dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut. Sistem blok penyusunan *rotary encoder* dilihat pada gambar 2.9[11]

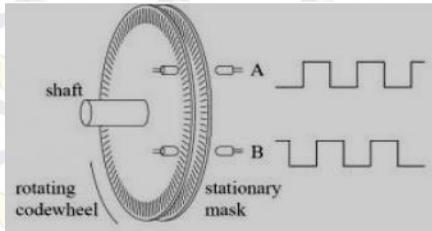


**Gambar 2.9** Sistem Blok Penyusunan Rotary Encoder

Pada penelitian ini *rotary encoder* yang digunakan termasuk dalam kategori *incremental encoder*. *Incremental encoder* terdiri dari dua *track* atau *single track* dan dua *sensor* yang disebut *channel A* dan *B*, ketika poros berputar, deretan pulsa akan muncul pada masing-masing *channel* pada *frekuensi* yang proporsional dengan kecepatan putar sedangkan hubungan *phase* antara *channel A* dan *B* menghasilkan arah putaran. Putaran dapat dihitung dengan cara jumlah pulsa yang terjadi *resolusi* piringan. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat kita tentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda *phase* seperempat putaran (*quadrature signal*).

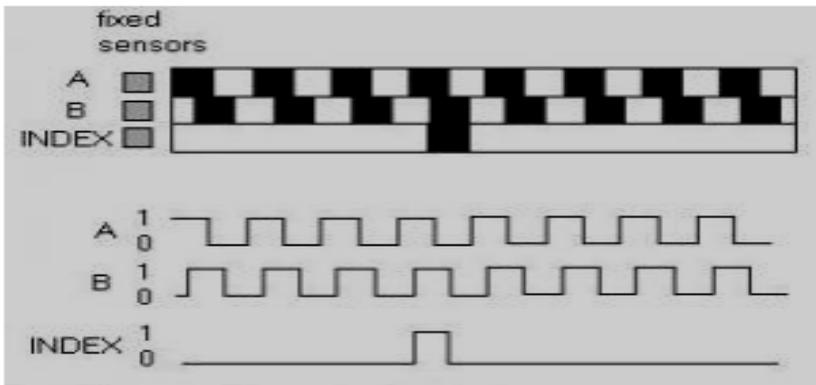
*Quadrature* sinyal atau sinyal berpangkat sebenarnya terdiri dari 2 sinyal yang biasanya dimisalkan dengan *I* (untuk *phase biasa*) dan *Q* (untuk *phase kuadrat*) atau dengan *RE* (untuk *sinyal Real*) dan *IM* (untuk *sinyal imajiner*). Letak perbedaan dari sinyal sebenarnya dengan sinyal *kuadrat* adalah terletak di *spektrum* dari sinyal *kuadrat* tidak *simetri* dengan *frekuensi 0*.

Seringkali terdapat *output channel* ketiga, disebut *INDEX*, yang menghasilkan satu *pulsa* per putaran berguna untuk menghitung jumlah putaran yang terjadi. Susunan piringan untuk *incremental encoder* dapat dilihat pada gambar 2.10

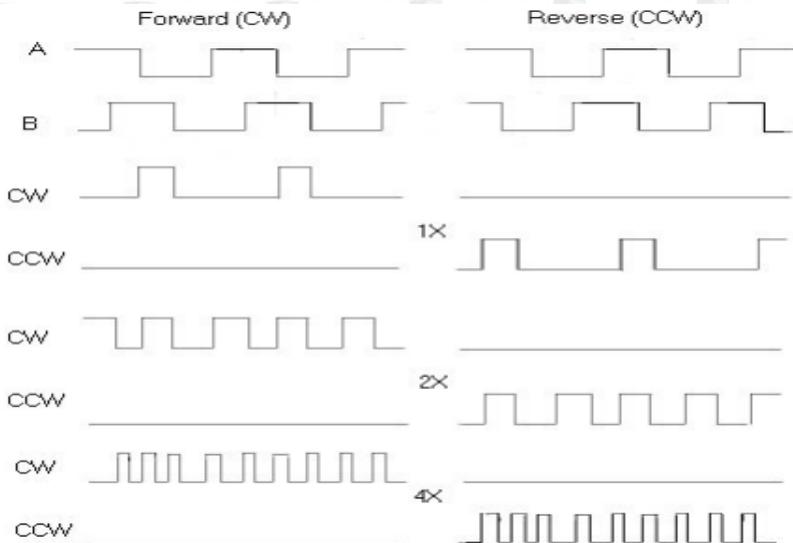


**Gambar 2.10** Susunan Piringan Untuk Incremental Encoder

*Resolusi* keluaran dari sinyal *quadrature* A dan B dapat dibuat beberapa macam yaitu *1X*, *2X* dan *4X*. *Resolusi 1X* hanya memberikan *pulsa* tunggal untuk setiap *siklus* salah satu *sinyal* A dan B, sedangkan *resolusi 4X* memberikan pulsa setiap *transisi* pada kedua *sinyal* A dan B menjadi empat kali *resolusi 1X* arah putaran dapat ditentukan melalui *level* salah satu *sinyal* selama *transisi* terhadap *sinyal* yang kedua. Pada contoh *resolusi 1X*, A = arah bawah dengan B = 1 menunjukkan arah putaran searah jarum jam, sebaliknya B = arah bawah dengan A = 1 menunjukkan arah berlawanan jarum jam. Skema pola keluaran *incremental encoder* dapat dilihat pada gambar 2.11 dan skema *output* dan arah putaran pada *resolusi* yang berbeda-beda dapat dilihat pada gambar 2.11



**Gambar 2.11** Contoh Pola Keluaran Incremental Encoder



**Gambar 2.12** Output Dan Arah Putaran Pada Resolusi Yang Berbeda-beda

## 2.6 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar penginderaannya. Perbedaan waktu antara gelombang suara yang dipancarkan dengan gelombang suara yang ditangkap kembali berbanding lurus dengan jarak atau tinggi objek yang memantulkannya. Jenis objek yang dapat diindera diantaranya adalah : objek padat, cair, butiran maupun tekstil.

Gelombang ultrasonik timbul akibat getaran mekanik dengan frekuensi diatas batas ambang pendengaran manusia yakni diatas 20Khz. Jika gelombang bolak-balik terjadi terus menerus secara periodik maka akan menghasilkan deretan gelombang periodik dimana pada setiap gerak periodik, partikel-partikel yang berada pada titik-titik yang sama pada gelombang tersebut akan berada dalam fase yang sama.

Jarak antara dua nilai puncak gelombang yang berurutan (gelombang transversal) atau jarak dari dua bagian pemampatan gelombang yang berurutan ( gelombang longitudinal) disebut panjang

gelombang ( $\lambda$ ). Waktu yang dibutuhkan untuk menempuh satu gelombang penuh atau waktu yang ditempuh sepanjang gelombang tersebut disebut periode (T). hubungan antara panjang gelombang dengan periode ini adalah[10] :

$$\lambda = c \cdot T \quad (2-2)$$

frekuensi gelombang (f) adalah banyaknya getaran yang terjadi persatuan waktu periode sehingga persamaan diatas dapat juga ditulis dalam bentuk frekuensi terhadap laju gelombang bunyi dalam medium (c) yakni

$$\lambda = c / f \quad (2-3)$$

Besarnya panjang gelombang ini sangat penting untuk menentukan batas resolusi pencitraan sistem. Dua bentuk struktur yang berdekatan panjang gelombangnya tidak bisa diidentifikasi secara terpisah pada pencitraan ultrasonik.

Kecepatan bunyi dalam medium berbeda-beda. Material dengan molekul berat cenderung bergerak lebih lambat dibandingkan molekul ringan saat terjadinya perubahan tekanan dalam medium. Material yang sangat kompresibel seperti gas akan meneruskan gelombang bunyi lebih lambat sehingga penambahan intensitas atau kompresibilitas cenderung akan menurunkan kecepatan bunyi. Kecepatan bunyi dalam medium/jaringan ditunjukkan pada Tabel 2.1[12]

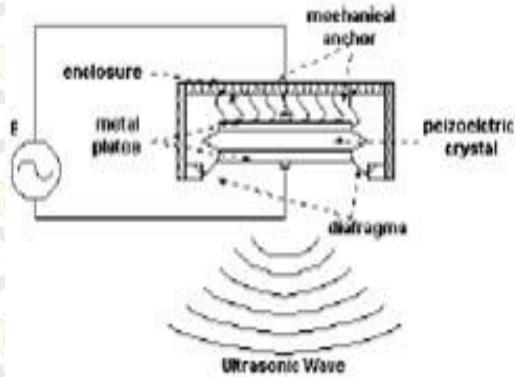
**Tabel 2.1** Kecepatan bunyi dalam medium/jaringan.

| Medium/jaringan            | Kecepatan bunyi(m/s) |
|----------------------------|----------------------|
| Udara                      | 334                  |
| Lemak                      | 1.450                |
| Air                        | 1.495                |
| Jaringan lunak (rata-rata) | 1.540                |
| Ginjal                     | 1.561                |

|        |       |
|--------|-------|
| Otot   | 1.585 |
| Tulang | 4.080 |

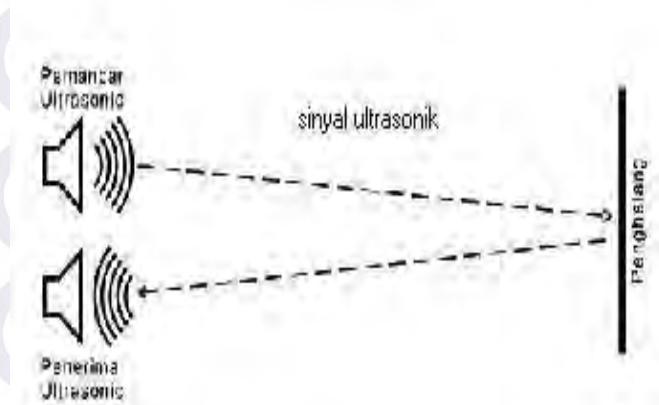
Pada Tabel 2.1 memperlihatkan kecepatan bunyi melalui beberapa medium, dimana kecepatan bunyi bergantung kepada kerapatan dan kompressibilitas medium. Material dengan molekul berat, kecepatan bunyinya cenderung lebih lambat dibandingkan molekul ringan. Dan material yang sangat kompressibel seperti gas dengan jarak simpangan molekul yang panjang akan memperlambat gelombang menjadi lebih lambat.

Sensor ultrasonik terdiri dari dua unit yaitu unit pemancar dan unit penerima. Struktur dari unit pemancar dan penerima yaitu sebuah kristal *piezoelectric* dihubungkan dengan mekanik yang kuat dan hanya dihubungkan dengan diafragma penggetar. Tegangan bolak-balik yang memiliki frekuensi 40KHz-400KHz diberikan pada plat logam. Struktur atom dari kristal *piezoelectric* akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan, dan ini yang disebut dengan efek *piezoelectric*. Kontraksi yang terjadi diteruskan ke diafragma penggetar sehingga terjadi gelombang ultrasonik yang dipancarkan ke udara (tempat sekitarnya) dan pantulan gelombang ultrasonik akan terjadi bila ada objek tertentu kemudian pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek *piezoelectric* menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama. Untuk lebih jelas tentang prinsip kerja dari sensor ultrasonik dapat dilihat prinsip dari sensor ultrasonik pada Gambar 2.13



**Gambar 2.13.** Prinsip kerja sensor ultrasonik

Besar amplitudo sinyal elektrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses *sensing* yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian Tx sampai diterima oleh rangkaian Rx, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara. Prinsip pantulan dari sensor ultrasonik ini dapat dilihat pada Gambar 2.14



**Gambar 2.14** Prinsip pemantulan gelombang ultrasonik

Pada Gambar 2.14 diatas dapat dilihat suatu proses pemantulan gelombang ultrasonik, dimana unit pemancar ultrasonik memancarkan sinyal ultrasonik yang kemudian sinyal ultrasonik tersebut di pantulkan oleh penghalang hingga pantulannya kembali lagi dan diterima oleh unit penerima ultrasonik. hasil sinyal yang diterima oleh transduser penerima akan dikonversikan menjadi jarak. Dengan mengukur selang waktu antara saat pulsa dikirim dan pulsa diterima , jarak antara obyek dengan alat pengukur akan dapat dihitung berdasarkan persamaan :

$$D = V \cdot T \quad (2-4)$$

Keterangan : V = kecepatan suara ( m/ det )

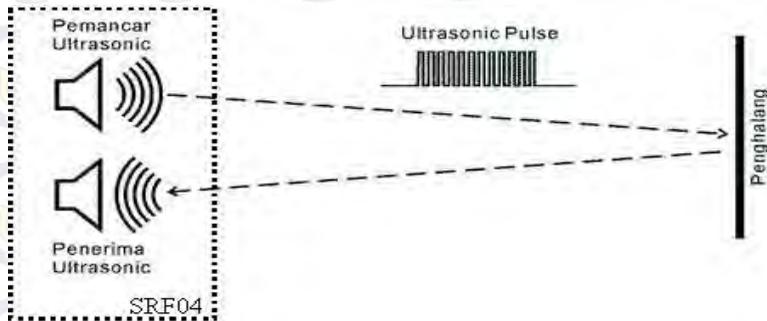
T = selang waktu ( detik )

D = jarak antara sumber dengan obyek ( m )

### 2.6.1 Sensor Ultrasonik HC-RS04

HC-RS04 adalah merupakan modul yang berisi transmitter dan receiver ultrasonic, modul dapat digunakan untuk mengukur jarak. Modul ini mengukur jarak dengan cara menghitung selisih waktu antara saat pemancaran sinyal dan saat penerimaan sinyal pantul. Seperti diketahui, kecepatan rambat suara di udara adalah 34399.22 cm/detik, berarti untuk merambat sejauh 1 cm suara membutuhkan waktu 29 mikro detik. Misalkan waktu antara pengiriman dan penerimaan sinyal ultrasonik adalah 5800 mikro detik, maka jarak antara sensor dan benda (

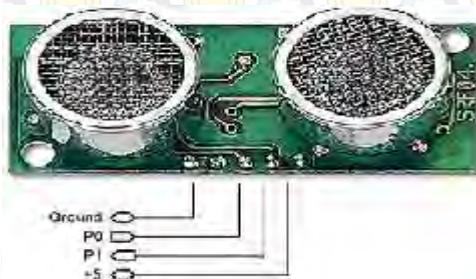
penghalang ) adalah  $100 \text{ cm} \cdot (2 \times 100 \text{ cm} \times 29 \text{ mikro-detik/cm} = 5800 \text{ mikro detik})$ .( gambar 2.15 )



**Gambar 2.15** Pemantulan sensor ultrasonic

### 2.6.2 Prinsip kerja sensor ultrasonic

HC-RS04 hanya menggunakan 2 port I/O untuk berhubungan dengan mikrokontroler, sehingga sangat ideal untuk aplikasi-aplikasi robotika, HC-RS04 dapat mengukur jarak mulai 3 cm sampai 3 m, dan dapat mengukur benda dengan diameter 3 cm pada jarak kurang dari 2 meter. ( gambar 2.16 ).

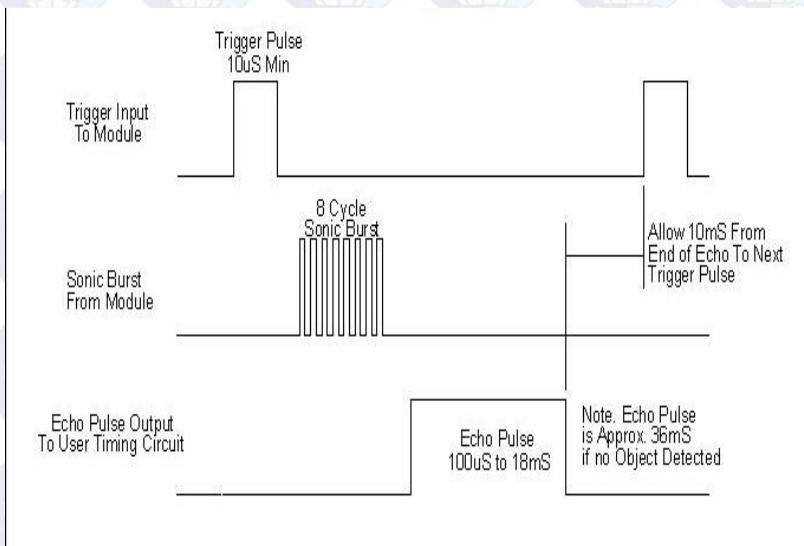


**Gambar 2.16** Bentuk fisik HC-RS04

Pulsa Ultrasonic yang dikirim oleh HC-RS04 adalah sinyal ultrasonic dengan frekwensi 40 KHz sebanyak 8 periode setiap kali pengiriman. Ketika pulsa mengenai benda penghalang, maka pulsa ini

akan dipantulkan kembali dan diterima kembali oleh penerima Ultrasonic. Dengan mengukur selang waktu antara saat pulsa dikirim dan pulsa pantul diterima, maka jarak benda penghalang bisa dihitung.

Apabila PI ( trigger pulse input ) diberi logika 1 ( high ) selama minimal 10 uS maka HC-RS04 akan memancarkan sinyal ultrasonic, setelah itu pin PO ( echo pulse output ) akan berlogika high selama 100 uS – 18 mS. ( tergantung jarak sensor dan penghalang ) dan apabila tidak ada penghalang maka PO akan berlogika 1 selama kurang lebih 38 mS ( gambar 2.17[17] )



**Gambar 2.17** Timing Diagram HC-RS04

Misalkan lama Echo Pulse adalah  $T$ , maka untuk mengetahui jaraknya dapat diketahui dengan cara membagi  $T$  dengan 58 (  $T/58$  ) untuk satuan centi meter dan dibagi dengan 148 (  $T/148$  ) untuk satuan inci. Misalkan panjang Echo Pulse adalah 5800 mikro detik maka jarak benda adalah 1 meter (  $5800/58 = 100\text{ cm} = 1\text{ meter}$  ).



Gambar 2.18 [13] merupakan konfigurasi dari pin LCD M1632, dimana setiap pin memiliki fungsi tersendiri seperti yang terdapat pada Tabel 2.2

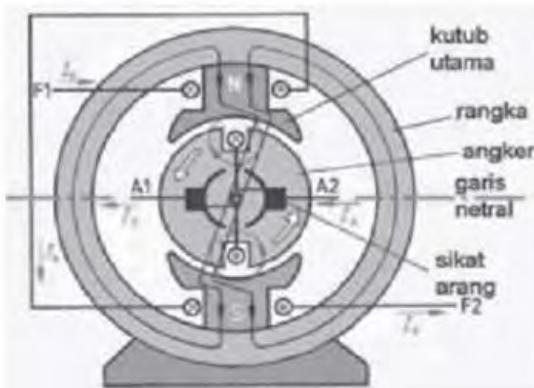
**Tabel 2.2** Konfigurasi pin LCD M1632

| No | <u>Simbol</u> | Level | <u>Fungsi</u>                                       |
|----|---------------|-------|---|
| 1  | <u>Vss</u>    | -     | 0 Volt  |
| 2  | <u>Vcc</u>    | -     | 5 + 10% Volt  |
| 3  | <u>Vee</u>    | -     | <u>Penggerak LCD</u>                                |
| 4  | RS            | H/L   | H= <u>memasukan data</u><br>L= <u>memasukan Ins</u> |
| 5  | R/W           | H/L   | H= <u>Baca</u><br>L= <u>Tulis</u>                   |
| 6  | E             |       | Enable Signal                                       |
| 7  | DB0           | H/L   | Data Bus  |
| 8  | DB1           | H/L   |   |
| 9  | DB2           | H/L   |   |
| 10 | DB3           | H/L   |   |
| 11 | DB4           | H/L   |   |
| 12 | DB5           | H/L   |   |
| 13 | DB6           | H/L   |   |
| 14 | DB7           | H/L   |   |
| 15 | V-BL          |       | <u>Kecerahan LCC</u>                                |
| 16 | V-BL          |       |   |

M1632 merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang di desain khusus untuk mengendalikan LCD.

## 2.8 Motor DC

Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula seperti pada gambar 2.19[19] Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor.



**Gambar 2.19** Kontruksi motor DC

Konstruksi motor DC pada gambar 2.19 memiliki 2 bagian dasar,yaitu :

1. Bagian yang tetap/stasioner yang disebut stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektromagnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.

Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh megnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet

dari kutub utara ke kutub selatan. Menurut hukum gaya Lourentz, arus yang mengalir pada penghantar yang terletak dalam medan magnet akan menimbulkan gaya. Gaya  $F$ , timbul tergantung pada

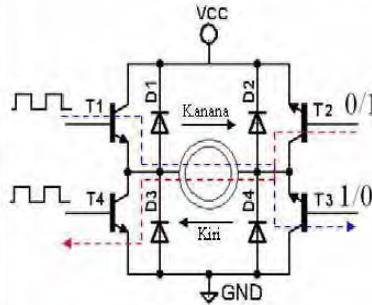
arah arus I, dan arah medan magnet B.

### **2.8.1 Driver Motor DC dengan IC L293D**

Pada dasarnya beberapa aplikasi yang menggunakan motor DC harus dapat mengatur kecepatan dan arah putar dari motor DC itu sendiri. Untuk dapat melakukan pengaturan kecepatan motor DC dapat menggunakan metode PWM (Pulse Width Modulation) sedangkan untuk mengatur arah putarannya dapat menggunakan rangkaian H-bridge yang tersusun dari 4 buah transistor. Tetapi dipasaran telah disediakan IC L293D sebagai driver motor DC yang dapat mengatur arah putar dan disediakan pin untuk input yang berasal dari PWM untuk mengatur kecepatan motor DC. Untuk lebih memahami tentang membangkitkan sinyal PWM menggunakan fitur Timer pada mikrokontroler AVR dapat membacanya pada postingan tutorial AVR tentang PWM. Sebelum membahas tentang IC L293D, alangkah baiknya jika kita membahas driver motor DC menggunakan rangkaian analog terlebih dahulu.

Jika diinginkan sebuah motor DC yang dapat diatur kecepatannya tanpa dapat mengatur arah putarnya, maka kita dapat menggunakan sebuah transistor sebagai driver. Untuk mengatur kecepatan putar motor DC digunakan PWM yang dibangkitkan melalui fitur Timer pada mikrokontroler. Sebagian besar *power supply* untuk motor DC adalah sebesar 12 V, sedangkan output PWM dari mikrokontroler maksimal sebesar 5 V.

Oleh karena itu digunakan transistor sebagai penguat tegangan. Dibawah ini adalah gambar driver motor DC menggunakan transistor. Salah satu jenis motor yang sering digunakan dalam bidang kontrol yaitu Motor DC. Motor DC akan berputar jika dialiri tegangan dan arus DC. Berikut gambar 2.20 adalah motor DC dan jembatan H yang digunakan pada rancangan alat ini:



**Gambar 2.20** Motor DC dan Jembatan H

Sistem pengaturan motor DC yang sering digunakan pada sistem kontrol seperti pada gambar 2.20[14] yaitu dengan *H-Bridge* yang pada dasarnya adalah 4 buah transistor yang difungsikan sebagai saklar. Pengaturan motor DC yaitu meliputi kecepatan dan arah. Pengaturan arah yaitu dengan cara membalik tegangan logika masukan *H-bridge*. Sedangkan sistem pengendalian kecepatan motor DC digunakan prinsip PWM (*Pulse Width Modulator*) yaitu suatu metode pengaturan kecepatan putaran motor DC dengan mengatur lamanya waktu pensaklaran aktif (*Duty Cycle*). Motor DC merupakan sebuah komponen yang memerlukan arus yang cukup besar untuk menggerakannya. Oleh karena itu motor DC biasanya memiliki penggerak tersendiri. Pada tugas akhir ini motor DC akan digerakkan dengan menggunakan PWM yang telah terintegrasi dengan rangkaian *HBridge*. Dengan rangkaian *H-Bridge* yang memiliki input PWM ini, maka selain arah kita juga bisa mengendalikan kecepatan putar motor DC tersebut.

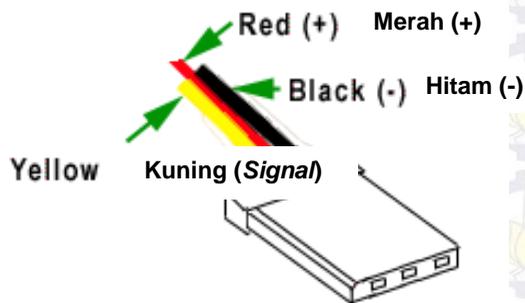
## 2.9 Motor Servo

Motor servo adalah sebuah alat yang dapat mengubah energi listrik menjadi energi mekanik. Motor servo sebenarnya digerakkan oleh sebuah motor DC. Karena motor servo digerakkan oleh motor DC maka jenis arus yang dipergunakan adalah jenis arus searah. Selain terdapat komponen motor DC, di dalam motor servo pun terdapat roda gigi yang berfungsi untuk mereduksi putaran dan untuk memperbesar torsi yang dihasilkan oleh motor DC tersebut.

Berbeda dengan motor DC yang akan langsung berputar ketika diberi tegangan, motor servo tidak akan berputar tanpa adanya perintah tertentu walaupun telah diberi tegangan. Motor servo dapat bergerak bila dialiri arus DC dan kaki sinyal diberi isyarat sinyal atau pulsa berupa arus listrik. Dengan diberi pulsa tertentu, motor servo akan berputar pada sudut tertentu.

### 2.9.1 Konfigurasi Pin Motor Servo

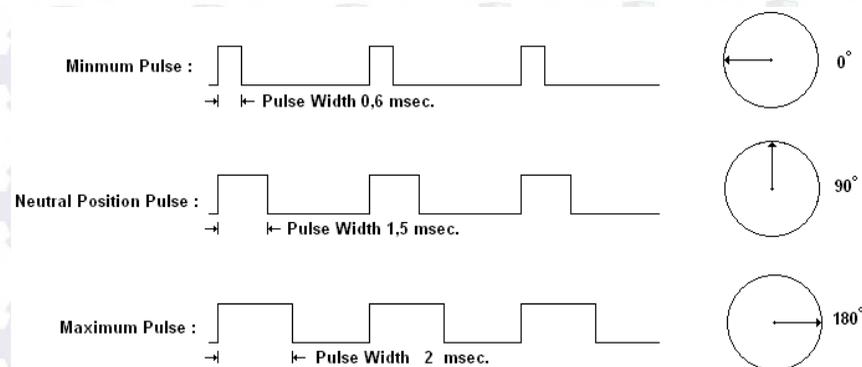
Motor servo hanya memiliki 3 kabel yang mana masing-masing fungsinya terdiri dari positif (*Vcc*), negatif (*Ground*) dan kontrol (*Signal*). Motor servo mampu bergerak searah jarum jam ataupun berlawanan arah jarum jam tanpa membalik pin konektor pada motor servo, hal ini disebabkan bahwa pada motor servo telah terdapat *driver* untuk membalik polaritas motor DC yang ada pada motor servo. Konfigurasi pin pada motor servo dapat dilihat pada gambar 2.21[15]



**Gambar 2.21** Konfigurasi pin pada motor servo

### 2.9.2 Prinsip Kerja Motor Servo

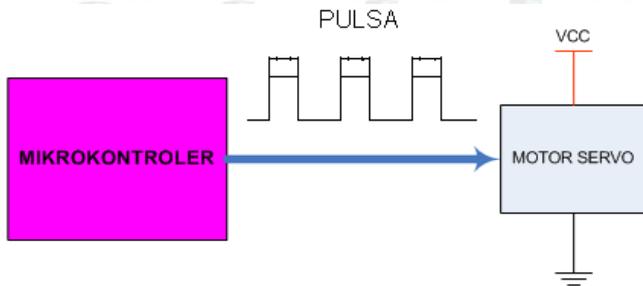
Salah satu jenis motor servo adalah motor servo jenis standar. Motor servo jenis standar hanya dapat berputar  $180^\circ$  searah atau berlawanan arah jarum jam. Prinsip kerja motor servo standar adalah kaki sinyal motor servo diberi sinyal digital dengan lebar sinyal antara 0,60 milidetik sampai 2,00 milidetik. Sinyal akan dideteksi setiap 20 milidetik. Apabila dalam selang waktu lebih dari 20 milidetik motor servo tidak mendeteksi sinyal maka motor servo akan slip. Bentuk sinyal pengontrolan motor servo dapat dilihat pada gambar 2.22[20]



**Gambar 2.22** Bentuk sinyal pengontrolan motor servo

Untuk memposisikan poros motor servo pada sudut tertentu dan selama waktu tertentu, lebar dan jumlah pulsa tertentu harus diberikan pada kaki sinyal motor servo. Agar motor servo dapat diposisikan pada sudut tertentu terlebih dahulu lebar pulsa untuk posisi 0° dan posisi 180° harus diketahui. Setelah mengetahui lebar pulsa pada kedua posisi tersebut, lebar pulsa untuk posisi sudut yang lain bisa diketahui dengan cara interpolasi.

Karena lamanya pulsa yang diberikan sangat singkat maka pemberian pulsa pada kaki sinyal tidak mungkin dilakukan secara manual. Selain itu sejumlah pulsa harus diberikan selama selang waktu tertentu. Oleh karena itu, pemberian pulsa pada kaki sinyal motor servo dilakukan dengan menggunakan mikrokontroler. Mikrokontroler akan terus memberikan sejumlah pulsa ke kaki sinyal pada motor servo sesuai dengan program. Skematis pengiriman pulsa dapat dilihat pada gambar 2.23



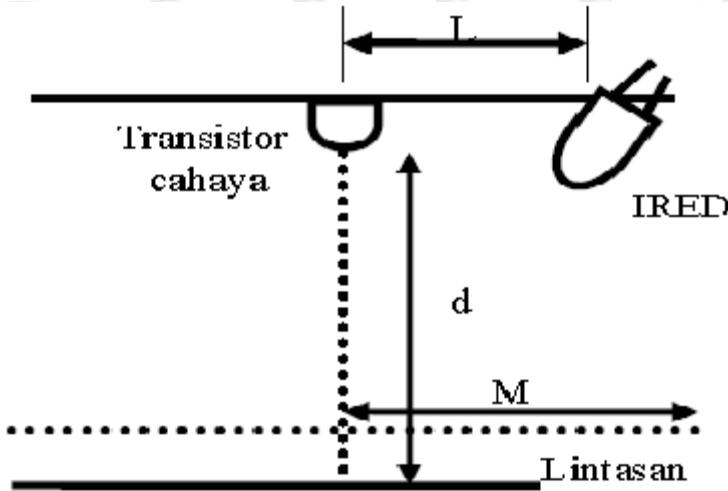
**Gambar 2.23** Skematis pengiriman pulsa

## 2.10 Sensor Photodioda

Terdapat dua sensor untuk mendeteksi garis, yaitu dengan menggunakan Infra Red dan dengan menggunakan mikrokontroller. Sensor pendeteksi garis yang digunakan dalam robot biasanya didasarkan pada prinsip pemantulan cahaya untuk membedakan warna garis dengan latar belakangnya. Pada warna gelap penyerapan cahaya lebih besar daripada warna putih sehingga cahaya terpantul ke sensor menjadi lebih kecil. Cahaya yang digunakan untuk pengenalan garis biasanya adalah cahaya tampak dan infra-merah.

Metode untuk membuat Robot Pengikut Garis dapat mengikuti garis ada beberapa cara. Cara pertama adalah membuat garis berada di antara deretan sensor. Kedua, dengan membuat sensor berada di atas garis. Ketiga, dengan cara selalu mendeteksi tepi garis. Cara keempat adalah dengan selalu melalui garis secara bolak-balik.

Percobaan untuk mencari karakteristik transistor cahaya untuk mendeteksi perbedaan warna garis dengan warna latar belakangnya. Jalannya percobaan ditunjukkan pada gambar 2.25[16].



**Gambar 2.24** percobaan warna garis dan lintasa

Berdasarkan gambar diatas sensor yang digunakan untuk mengamati dunia luar dari sistem yang dirancang adalah sensor aktif infra merah. Parameter yang diinginkan diamati oleh robot adalah keberadaan garis yang diikuti di dalam arena. Keluaran sensor masih berupa data mentah dalam bentuk tegangan selanjutnya diekstraksi oleh bagian persepsi dengan untai pengkondisi isyarat berupa komparator sehingga dihasilkan isyarat digital dengan standar TTL (Transistor Transistor Logic).

### 2.10.1 Prinsip kerja sensor photodioda

Prinsip kerja dari sensor photodioda ini apabila menerima cahaya konduktifitasnya semakin baik dan apabila mengenai garis putih maka cahaya akan dipancarkan, jika mengenai warna hitam maka cahaya akan diserap. Sifat dari photodioda adalah jika semakin banyak cahaya yang diterima, maka nilai resistansi diodanya semakin kecil. Dengan melakukan sedikit modifikasi, yaitu menggunakan rangkaian resistor pembagi tegangan maka besaran resistansi tersebut dapat diubah menjadi tegangan. Sehingga jika sensor berada diatas garis hitam, maka tegangan keluaran sensor akan kecil, demikian pula sebaliknya. Pada penggunaan sensor cahaya biasanya menggunakan dua metode

pembacaan sensor yaitu dengan Komparator dan ADC (Analog to Digital Converter).



**Gambar 2.25** cahaya pemantulan sedikit dan cahaya pemantulan banyak

## 2.11 Sensor tegangan

Tujuan penggunaan sensor tegangan adalah pengukuran tegangan keluaran pada nilai ADC yang terbaca oleh mikrokontroler. Modul ini didasarkan pada prinsip tekanan titik resistensi, dan dapat membuat tegangan input (dari terminal yang berwarna merah) mengurangi sebesar 5 kali dari tegangan asli. pada Arduino input analog tegangan 5 V, sehingga tegangan input dari modul ini harus tidak lebih dari  $5\text{ V} \times 5 = 25\text{ V}$  (jika untuk 3,3 V sistem, tegangan input harus tidak lebih dari  $3,3\text{ V} \times 5 = 16,5\text{ V}$ ). Karena chip AVR Arduino memiliki 10 bit AD, jadi resolusi simulasi modul  $0,00489\text{ V}$  ( $5\text{ V} / 1023$ ), dan tegangan input dari modul ini harus lebih dari  $0,00489\text{ V} \times 5 = 0,02445\text{ V}$  [17].

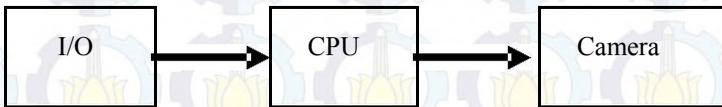


**Gambar 2.26** bentuk fisik sensor tegangan

pada robot input dari sensor adalah langsung dari baterai lipo 11.1 volt 2200mAh. Output dari sensor adalah 5 volt, sedangkan tegangan input sensor maksimal adalah 25 V atau 2500 mV.

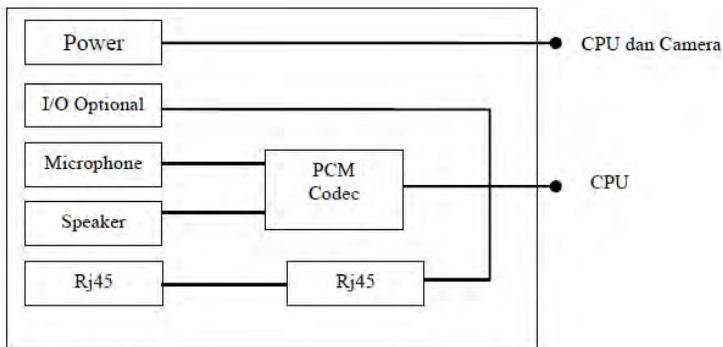
## 2.12 IP Camera (*Internet Protocol Camera*)

*IP camera* adalah perangkat *peng-capture* dan *recording* objek terkini yang memiliki kemampuan memproses visual dan audio serta dapat diakses komputer secara langsung atau melalui LAN, internet dan jaringan telepon selular. *IP camera* memiliki tiga blok utama yaitu blok CPU, I/O, dan camera. Ketiga bagian utama tersebut digambarkan dalam bentuk blok diagram seperti pada Gambar 2.28[22]



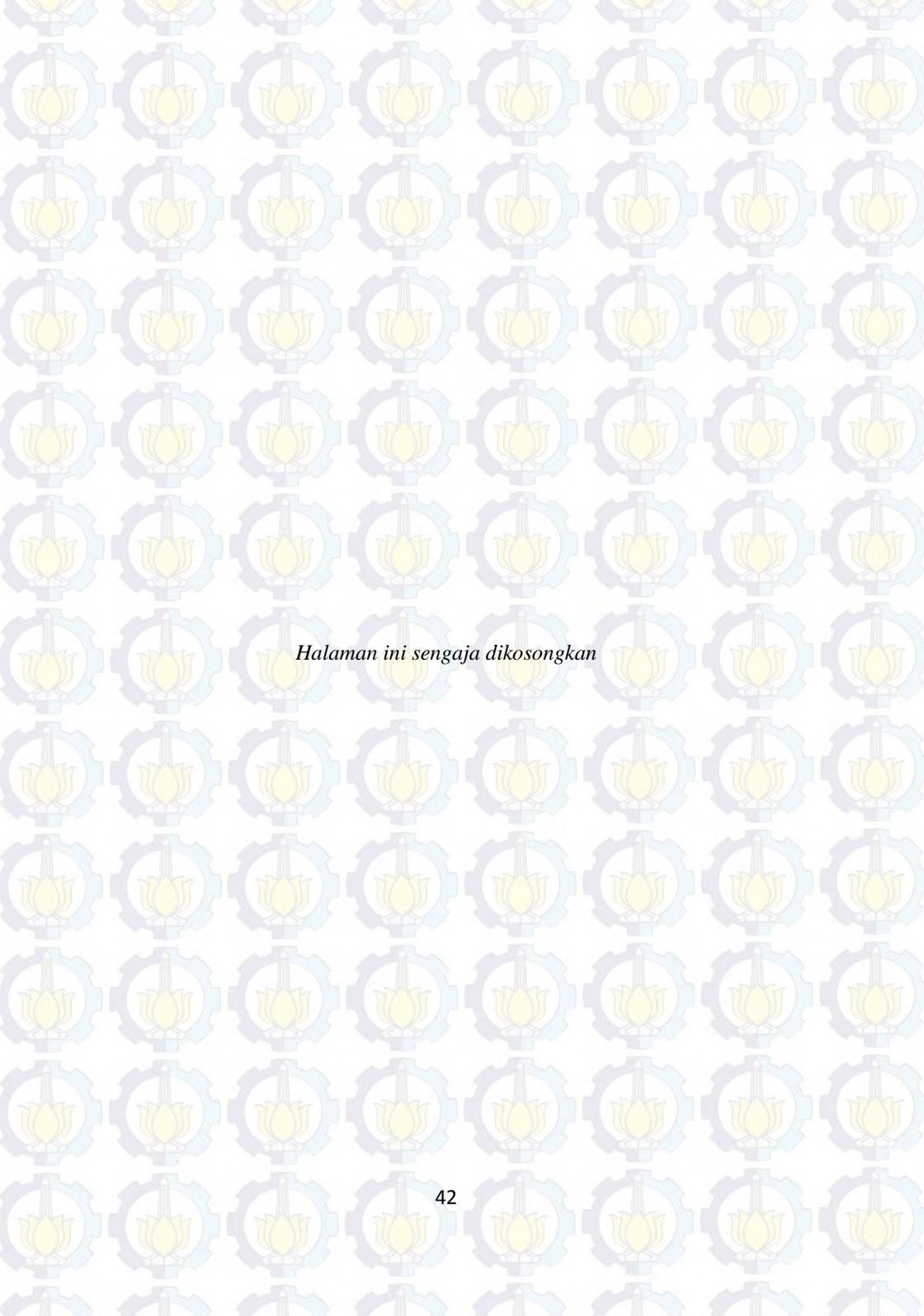
**Gambar 2.27** Blok Diagram IP Camera

Blok I/O (*input/output*) merupakan bagian yang menangani hubungan dengan peripheral *network* dan *gateway* untuk koneksi dengan LAN atau internet. Bagian ini dapat dengan mudah dikenali dengan adanya konektor *female* RJ45 dan beberapa jack input untuk *microphone* serta *speaker* yang merupakan penghubung dengan komputer seperti pada Gambar 2.28[18]



**Gambar 2.28** Blok Diagram I/O

- Keterangan
- I/O Optinal : Bagian input dan output berupa jack yang berfungsi sebagai penghubung antara *peripheral networking* dengan IP camera.
- Power Supply : Bagian yang mendistribusikan *power supply* atau catu daya dari *adaptor* ke seluruh bagian agar perangkat dapat beroperasi.
- Microphone : Bagian yang berfungsi mengubah sinyal akustik menjadi sinyal listrik
- Speaker : Bagian yang berfungsi mengubah sinyal listrik menjadi sinyal akustik yang dapat di dengar manusia
- RJ45 : Bagian yang berfungsi sebagai gateway IP camera untuk koneksi dengan komputer atau LAN/internet
- PCM Codec (*Pulse Code Modulation Coder and Decoder*) : Bagian yang berfungsi mengubah sinyal analog menjadi sinyal digital (*microphone to IP camera*) dan mengubah sinyal digital menjadi sinyal analog (*IP camera to speaker*) menggunakan teknik modulasi PCM
- LAN Buffer : Bagian yang berfungsi sebagai penyangga *traffic* data dari dan ke CPU IP camera yang melewati connector RJ45

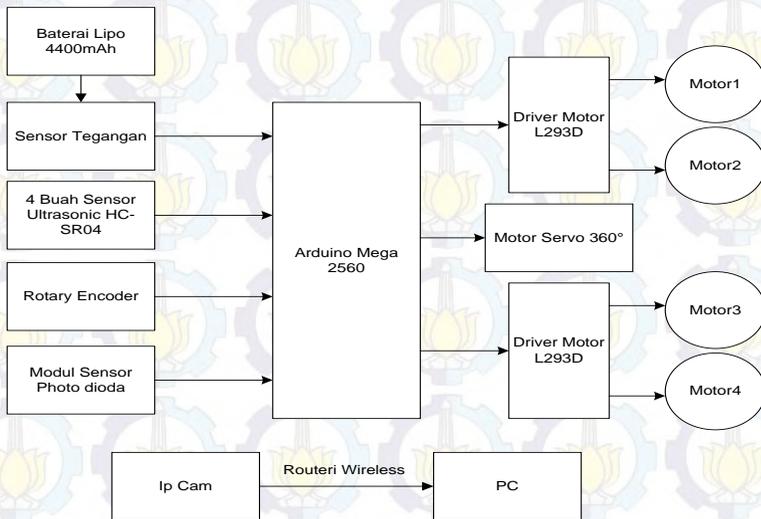


*Halaman ini sengaja dikosongkan*

### BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai ilustrasi pada sistem *implementasi auto docking system pada mobile robot*. Perancangan sistem ini meliputi perancangan perangkat keras dan rangkaian elektronika yang membahas tentang Baterai Lipo, ArduinoMega2560, Rotary Encoder, motor, Driver motor, sensor Ultrasonic HC-SR04, sensor tegangan, sensor Photodioda, *Charger* dan *Balancer*, *Power Supply*, motor Servo serta LCD 16x2. dan membahas mekanik robot, *docking station* serta perancangan perangkat lunak untuk pengawas menggunakan IP CAM.

Pada Tugas Akhir ini hanya fokus pada pembuatan sistem pada robot yaitu pada bagian penelusuran tempat yang telah disediakan serta robot melakukan *autodocking* agar kembali ke *docking station* secara benar.



**Gambar 3.1** Diagram blok sistem mobile robot.

Sistem yang dirancang memiliki diagram blok yang ditunjukkan pada gambar 3.1 Pada gambar 3.1 terdapat beberapa bagian, setiap bagian memiliki fungsi masing-masing, yaitu sebagai berikut :

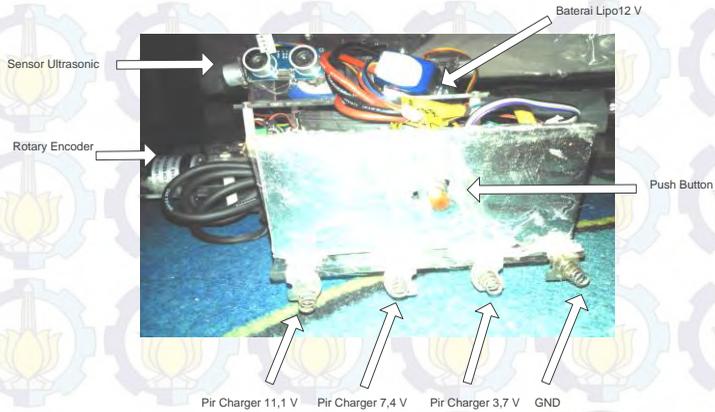
1. *Batterai Lipo* : berfungsi sebagai sumber tenaga dari robot
2. *ARDUINOMEGA2560* : pusat kendali dari robot dan juga pengontrolan dari robot
3. *Rotari Encoder* : berfungsi sebagai sensor yang dapat menghitung berapa jarak atau posisi mobile robot bergerak berdasarkan jumlah pulsa yang dihasilkannya
4. *Motor* : merupakan aktuator yang berfungsi untuk merubah energi listrik ke energi kinetik(gerak).
5. *Driver Motor* : berfungsi untuk interface antara mikrokontroller dengan aktuator(motor). Pada *driver motor* terdapat rangkaian yang berfungsi untuk mengatur arah pergerakan motor dan kecepatan motor serta untuk mensuplay tegangan pada motor.
6. *Sensor Ultrasonic HC-SR04* : berfungsi sebagai sensor jarak, sensor ultrasonik selain sebagai pendeteksi objek
7. *Sensor Tegangan* : berfungsi untuk sebagai PIN masukan baterai LIPO dan juga sebagai pembatas tegangan agar, tegangan baterai pada Baterai LIPO tidak kurang dari 9V
8. *Sensor Photodiode* : berfungsi untuk sebagai pencocokan antara keping PCB di robot dan keping PCB saat kembali ke *docking station*
9. *Charger dan Balancer* : adalah sebuah modul yang digunakan untuk *mencharger* Baterai dan *discharge* baterai
10. *Power Supply* : berfungsi untuk sebagai sumber tenaga dari modul charger dan *balancer*
11. *Motor Servo* : berfungsi untuk sebagai tempat dan menentukan arah dari IP Cam
12. *IP Cam* : berfungsi untuk sebagai pengawas.

cara kerja dari sitem robot *autodocking* ini memiliki 2 cara kerja: cara kerja perancangan sistem robot penelusuran tempat yang sudah disediakan serta perancangan sistem robot saat *autodocking* dan perancangan perangkat lunak pengawas tempat yang telah disediakan menggunakan IP CAM. Pada saat robot penelusuran seluruh tempat yang telah disediakan serta perancangan sistem robot saat *autodocking*. saat robot pertama kali robot berjalan mikrokontroller mengirimkan perintah kepada motor DC yang berada pada kanan dan kiri motor. secara

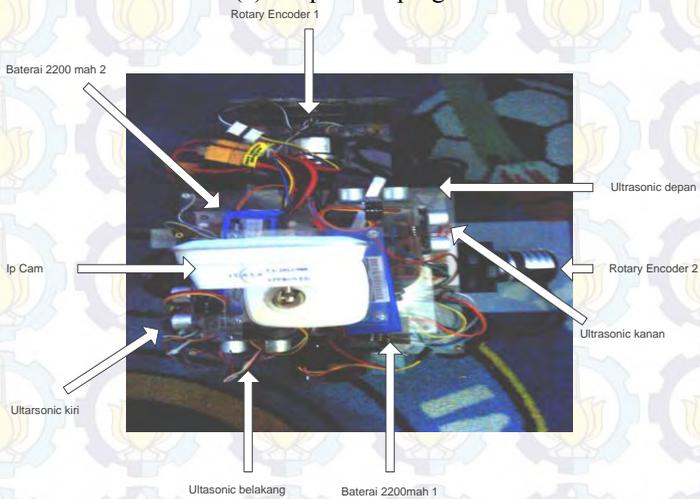
bersamaan pula pada sebelah kiri pada robot terdapat rotary encoder yang dimana akan menghitung pulsa pada setiap pergerakan robot. apabila robot di depan terkena halangan maka ultrasonic akan mengirimkan data, di depan akan ada halangan. secara bersamaan pula mikrokontroller akan mengirimkan perintah kepada Motor DC yang berada pada depan dan belakang pada robot, yang dimana motor ini akan bekerja bergerak ke kanan. secara bersamaan pula pada sebelah belakang robot terdapat rotary encoder yang dimana akan menghitung pulsa pada setiap pergerakan robot. Apabila pada robot tegangan baterai mendekati 11,09V maka, sensor tegangan akan mengirimkan data analog ke mikrokontroller, yang dimana mikrokontroller ini memiliki pin ADC. pergerakan robot pulang ke *docking station* sama dengan pergerakan robot saat melakukan penelusuran, hal ini dikarenakan karena pembacaan pulsa rotary encoder saat penelusuran di simpan di array, bila baterai mendekati habis maka akan program akan memanggil rotary yang disimpan pada array, robot akan berjalan pada *docking station* sesuai dengan data array. pada pembacaan rotary encoder terdapat nilai error, nilai error tersebut akan akan dimanfaatkan oleh modul sensor photodiode yang dimana untuk menangani error pembacaan terhadap rotary encoder yang diakibatkan slip As pada omni wheel. begitu pula saat robot mendekati *docking station* terdapat push button tujuannya adalah untuk untuk menghentikan robot agar tidak bergerak dan terdapat pir yang gunanya untuk mencocokkan antar konektor pir pada robot dan kepingan PCB pada *docking station*, untuk keperluan *charger* baterai. pada saat *charger* baterai LIPO yang terdiri dari 3 cell akan menggunakan *charger balancer* dari buatan industri, hal ini dimaksud untuk *mencharge* dan *discharge* pada baterai secara maksimal. untuk sumber *charger* dan *ballancer* digunakan *power supply* 12V 3A. pada robot terdapat LCD, hal ini digunakan untuk memonitoring tegangan baterai pada robot. pada saat pengawas tempat yang telah disediakan menggunakan pada bagian atas robot terdapat motor servo, motor servo dalam robot digunakan untuk membatu pergerakan dari IP CAM serta tempat dari IP CAM, motor servo bergerak mengikuti pergerakan dari. jadi motor servo bergerak secara 90 derajat. pada ip cam digunakan pengawas.

### 3.1. Perancangan Mekanik Robot dan *Docking Station*

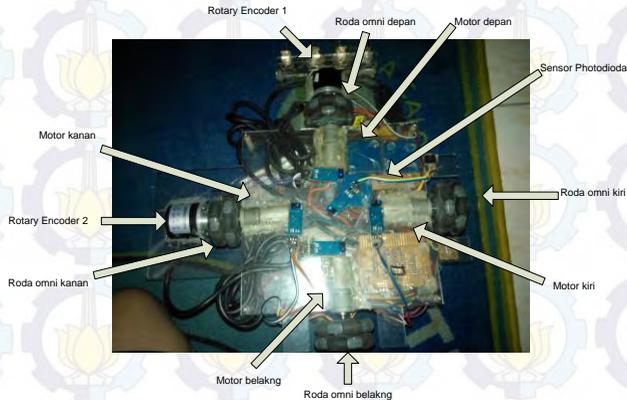
#### 3.1.1 Perancangan Mekanik Robot



(a) tampak samping



(b) tampak atas

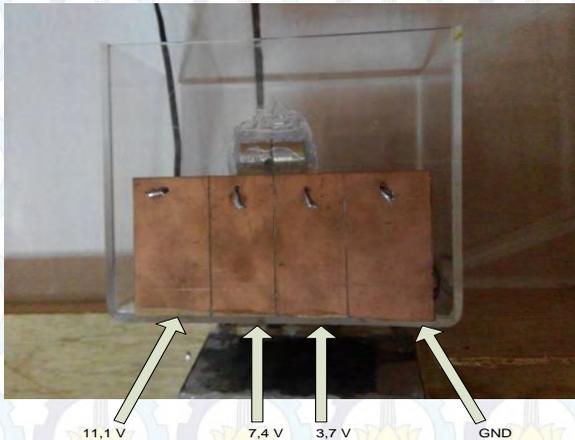


(c) tampak bawah

**Gambar 3.2** (a)Design mekanik robot tampak samping.  
 (b)Design mekanik robot tampak atas.  
 (c)Design mekanik robot tampak bawah.

Mobile robot yang dirancang, memiliki base berbentuk persegi dengan 4 buah roda penggerak menggunakan motor DC 4 buah. Roda penggerak menggunakan roda omni agar mobile robot dapat bergerak secara fleksibel. Kemudian terdapat rotary encoder dengan jumlah sensor rotari 2 buah dan di atasnya lagi terdapat sistem controller yang meliputi: mikrokontroler, sensor tegangan, sensor photodiode, LCD dan baterai . Dan di atas sendiri disediakan tempat untuk sensor ultrasonic, motor servo dan ip cam. Gambar 3.2 adalah bentuk design mekanik robot yang dirancang.

### 3.1.2 Perancangan *Docking Station*



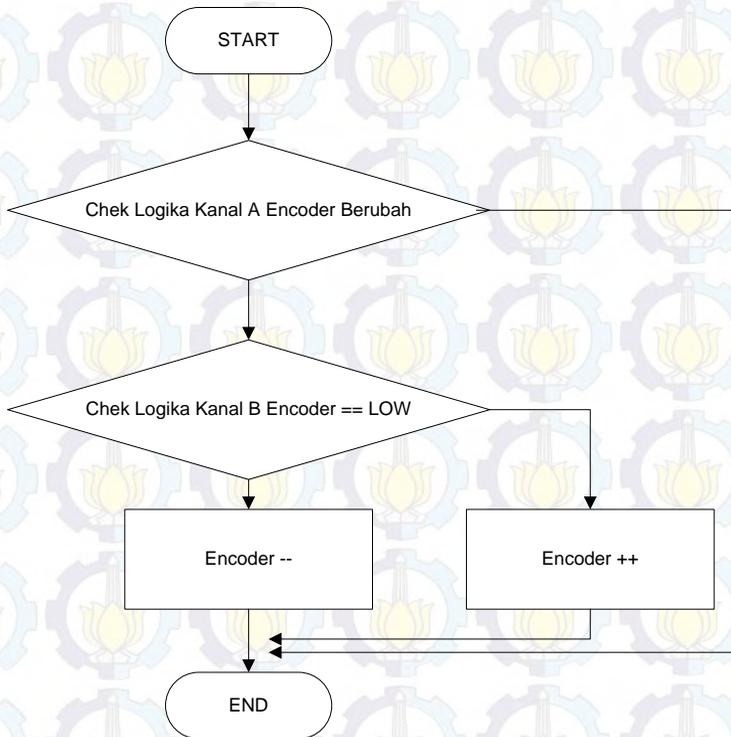
**Gambar 3.3** Design *Docking Station*

*Docking station* yang dirancang, memiliki base berbentuk kubus yang didalamnya terdapat *power supply* dan *charger ballancer*. dimana pada Gambar 3.3 terdapat lempengan pcb yang terpisah dan memiliki sumber tegangan yang berbeda beda pula. Gambar 3.3 adalah bentuk design mekanik robot yang dirancang.

### 3.2. Perancangan Rotary Encoder.

Sitem sensor rotiri encoder ini dibuat dengan bentuk segi empat dengan setiap sudutnya terdapat rotary encoder. Rotary encoder ini menggunakan jenis incremental rotary encoder.

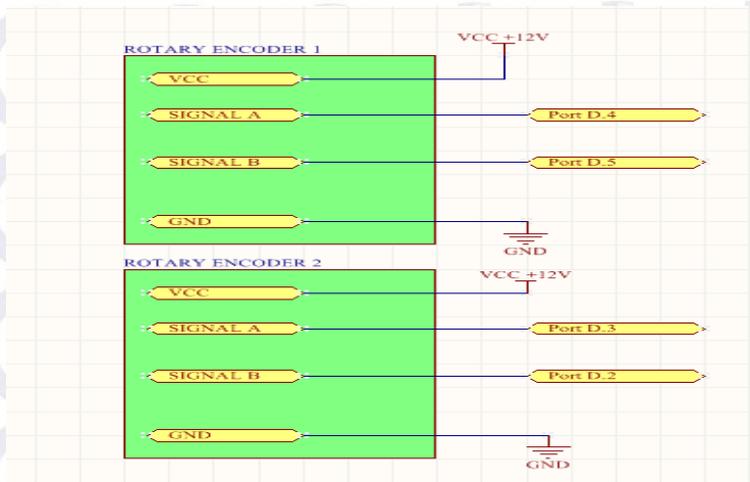
untuk mendeteksi forward/reverse rotary encoder yaitu dengan memanfaatkan perbedaan fasa antara kana A dan B. gambar 3. menjelaskan pertama dengan memperhatikan perubahan pulsa dari transisi low ke high pada kanal A kemudian memperhatikan pulsa pada kanal B apakah berlogika high atau low, pulsa high pada kanal B menandakan rotary encoder bergerak CW apabila pulsa low pada kanal B menandakan rotary encoder bergerak CCW.



**Gambar 3.4** Flowchart CW dan CCW reverse rotary encoder.

| No | Rotary Encoder | Pin | Arduino |
|----|----------------|-----|---------|
| 1  | Satu           | A   | PIND-4  |
| 2  |                | B   | PIND-5  |
| 3  | Dua            | A   | PIND-3  |
| 4  |                | B   | PIND-2  |

**Tabel 3.1** Hubungan antar pin Rotary Encoder

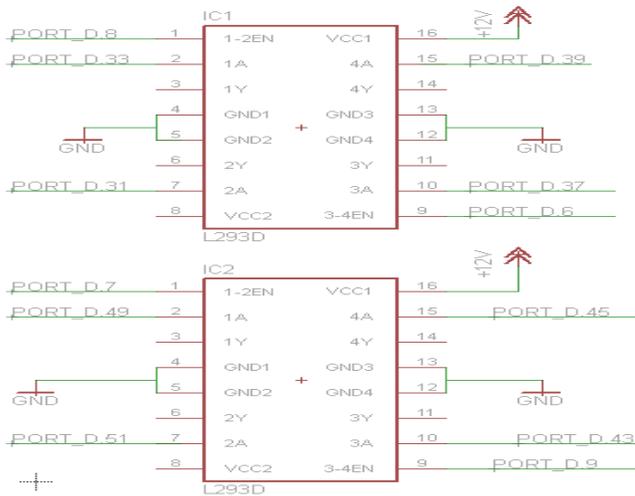


**Gambar 3.5** Hubungan Antara Rotary encoder dengan mikrokontroler.

### 3.3. Perancangan Motor DC dan Driver motor.

Untuk mengatur arah putaran dua buah motor, maka harus menggunakan rangkaian *dual full bridge*. Rangkaian ini bertindak sebagai saklar yang dapat mengubah arah arus yang melalui motor. Rangkaian ini sudah berada pada satu IC yaitu L298 yang diproduksi oleh STElectronics.

Apabila kita memasukan nilai logika *high* pada pin EnA, maka kita dapat mengubah arah arus dari OUT1 ke OUT2 atau sebaliknya dengan mengubah keadaan logika pada pin In1 dan In2 dengan catatan kedua pin harus mempunyai keadaan logika yang berbeda. Untuk mengalirkan arus dari OUT1 ke OUT2, maka pin In1 harus berlogika *high* dan pin In2 berlogika *low* Sedangkan untuk mengalirkan arus dari OUT2 ke OUT1, keadaan logika pin In1 dan In2 harus dibalik.



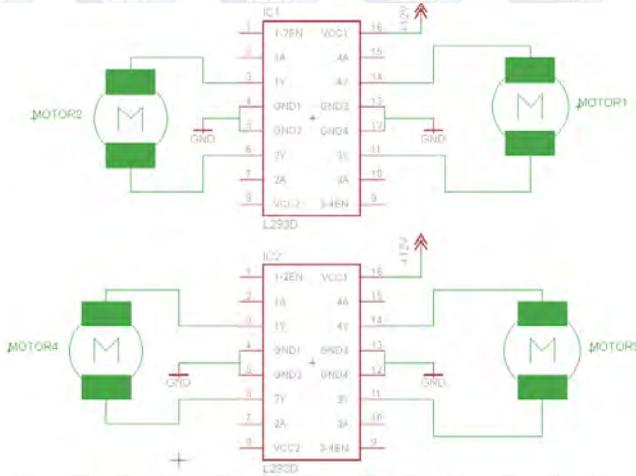
**Gambar 3.6** Hubungan Antara driver Motor dengan mikrokontroler

| No | L298d | Motor    | Pin    | Arduino |
|----|-------|----------|--------|---------|
| 1  | Satu  | Kiri     | 1-En1  | PIND-8  |
| 2  |       |          | 2-In1  | PIND-33 |
| 3  |       |          | 7-in2  | PIND-31 |
| 4  |       | Kanan    | 9-En2  | PIND-6  |
| 5  |       |          | 10-in3 | PIND-37 |
| 6  |       |          | 15-in4 | PIND-39 |
| 7  | Dua   | Depan    | 1-En1  | PIND-7  |
| 8  |       |          | 2-In1  | PIND-51 |
| 9  |       |          | 7-in2  | PIND-49 |
| 10 |       | Belakang | 9-En2  | PIND-9  |
| 11 |       |          | 10-in3 | PIND-43 |
| 12 |       |          | 15-in4 | PIND-45 |

**Tabel 3.2** Hubungan antar pin driver motor L298d

PWM terhadap motor DC sangatlah penting hal ini, digunakan untuk mengatur kecepatan dan gerak dari suatu robot. motor DC telah dapat berputar sesuai dengan teori bahwa ketika pin positif motor diberi

tegangan negatif maka motor akan berputar clock wise / searah jarum jam, begitu juga sebaliknya pin motor diberi tegangan kebalikannya. motor DC yang kedua didapatkan bahwa motor dapat merespon PWM yang diberikan oleh program. pengaturan pada setiap motor berbeda beda hal ini dikarenakan berat robot pada setiap sisi berbeda pada perancangan ini ini tidak membahas berapa RPM motor karena pada perancangan robot ini tidak ada sensor yang digunakan untuk mengukur kecepatan motor



**Gambar 3.7** Hubungan Antara Driver Motor dengan Motor DC

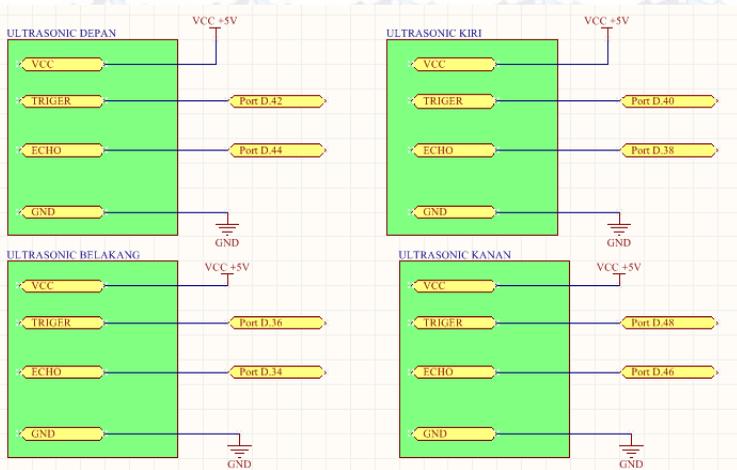
| No | L293d | Pin     | Motor   |
|----|-------|---------|---------|
| 1  | satu  | 3-out1  | Motor 2 |
| 2  |       | 6-out2  |         |
| 3  |       | 11-out3 | Motor 1 |
| 4  |       | 14-out4 |         |
| 5  | dua   | 3-out1  | Motor 4 |
| 6  |       | 6-out2  |         |
| 7  |       | 11-out3 | Motor3  |
| 8  |       | 14-out4 |         |

**Tabel 3.3** Hubungan antar Motor DC

### 3.4. Perancangan Ultrasonic

Sensor ultrasonik ini memiliki rangkaian yang sudah jadi yaitu berasal dari modul parallax. Pada alat pemantau jarak ini menggunakan 4 buah sensor ultrasonik yang dipasang pada keempat atas bagian robot.

*Scanning* sensor dilakukan secara bergantian agar data yang didapat *valid*. *Output* dari modul ini berupa sinyal sehingga dengan mudah kita dapat mengelola data tersebut untuk ditampilkan ke serial monitor. Untuk mengaktifkan sensor ini ketika ada pulsa *trigger* dari mikrokontroler (Pulsa *high* selama 5uS). Suara ultrasonik dengan *frekuensi* sebesar 40KHz akan dipancarkan selama 400uS. Suara yang akan merambat di udara, mengenai objek untuk kemudian terpantul kembali ke Pin PO ( echo pulse output ) akan berlogika high selama 100 uS – 18 mS.. Selama menunggu pantulan, sensor ultrasonik akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (*low*) ketika suara pantulan terdeteksi oleh Sensor ultrasonik. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara sensor ultrasonik dengan objek.



**Gambar 3.8** Hubungan Antara Ultrasonic dengan mikrokontroler

| No | Ultrasonic | Pin  | Arduino |
|----|------------|------|---------|
| 1  | Depan      | Trig | PIND-42 |
| 2  |            | Echo | PIND-44 |
| 3  | Kanan      | Trig | PIND-48 |

|   |          |      |         |
|---|----------|------|---------|
| 4 | Kiri     | Echo | PIND-46 |
| 5 |          | Trig | PIND-40 |
| 6 |          | Echo | PIND-38 |
| 7 | Belakang | Trig | PIND-36 |
| 8 |          | Echo | PIND-34 |

**Tabel 3.4** Hubungan antar Sensor Ultrasonic



**Gambar 3.9** Flowchart pembacaan jarak sensor ultrasonic.

### 3.5 Perancangan LCD (Liquid Cristal Display)

LCD Display Module M1632 buatan *Seiko instrument inc.* terdiri dari 2 bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf atau angka 2 baris, masing masing baris bisa menampung 16 huruf atau angka.

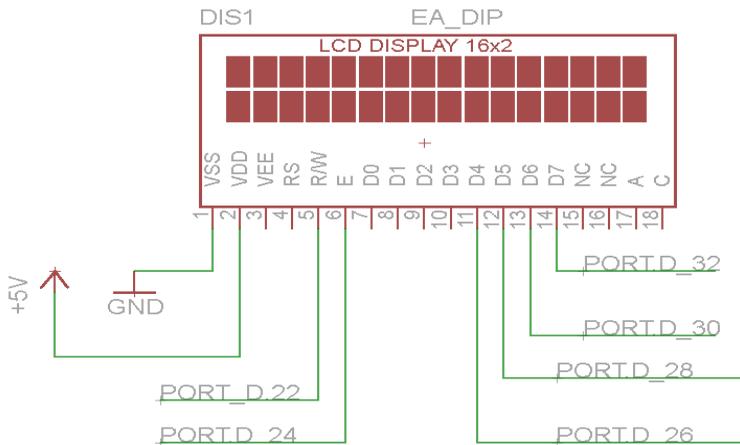
Rangkaian LCD M1632 ini adalah komponen *display* yang umum digunakan. *Display LCD* M1632 ini memiliki ROM sebagai penyimpanan karakter sebanyak 192 buah. Sebelum mengoperasikan

LCD sebagai penampil karakter, terlebih dahulu ditentukan format penulisan LCD.

Dalam penulisan format *LCD* terdapat beberapa aturan yang diberikan oleh pabrik pembuatannya (dalam data sheet) yaitu:

1. Menentukan jalur *bit* data yang akan digunakan.
2. Membersihkan layar *display* dari karakter *blank*.
3. Menentukan alamat baris pertama dan baris kedua .
4. Dalam penulisan karakter menggunakan *cursor* atau tidak.

jika penginialisasian sudah selesai, langkah selanjutnya adalah menulis karakter yang diinginkan adalah “123”, maka format data yang ditransfer ke jalur data adalah format data BCD angka 123 yang masing-masing disertai data posisi baris. Data yang dikirim ke LCD cukup satu kali, selanjutnya data akan terus tampil berulang-ulang oleh LCD itu sendiri selama tidak ada instruksi untuk membersihkan layar. Hubungan pin data dengan pin kontrol LCD dengan MCU ditunjukkan pada Gambar 3.10



**Gambar 3.10** Hubungan Antara LCD dengan mikrokontroler

| No | Pin | Arduino |
|----|-----|---------|
| 1  | VDD | VCC     |
| 2  | VSS | GND     |
| 3  | RW  | PIND-22 |

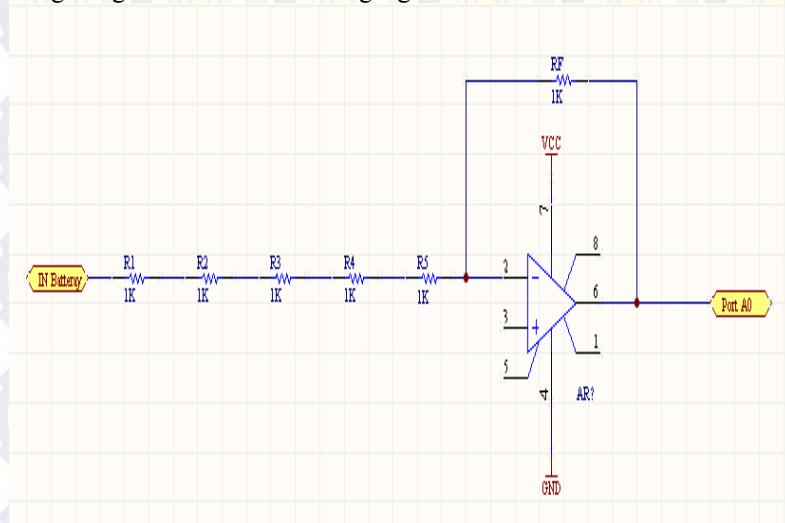
|   |    |         |
|---|----|---------|
| 4 | E  | PIND-24 |
| 5 | D4 | PIND-26 |
| 6 | D5 | PIND-28 |
| 7 | D6 | PIND-30 |
| 8 | D7 | PIND-32 |

**Tabel 3.5** Hubungan antar LCD

1. Pin RW dihubungkan dengan *port* D.22 dari MCU untuk membedakan sinyal antara instruksi program atau instruksi penulisan data.
2. Pin D4 – D7 dihubungkan dengan *port* D.26 – D.32 dari MCU untuk memberikan data ke LCD.
3. Pin E dihubungkan dengan *port* D.24 dari MCU untuk memberikan instruksi penulisan pada alamat LCD.

### 3.6 Perancangan Module Sensor Tegangan dan Baterai Lipo

Modul ini didasarkan pada prinsip tekanan titik resistensi, dan dapat membuat tegangan input (dari terminal yang berwarna merah) mengurangi sebesar 5 kali dari tegangan asli.



**Gambar 3.11** hubungan antara baterai Lipo sensor tegangan dan mikrokontroler

| No | Arduino | Sensor Tegangan |
|----|---------|-----------------|
| 1  | Pin-A0  | Out             |

**Tabel 3.6** Hubungan antar sensor tegangan dan Baterai Lipo

Berikut Penurunan rumus sensor tegangan:

pada Arduino input analog tegangan 5 V, sehingga tegangan input dari modul ini harus tidak lebih dari 5 V x 5 = 25 V ( jika untuk 3,3 V sistem, tegangan input harus tidak lebih dari 3,3 V x 5 = 16.5 V)

$$\frac{V_i - V_a}{R_1 + R_2 + R_3 + R_4 + R_5} = \frac{V_a - V_o}{R_f}$$

$$\frac{V_i - V_a}{R_t} = \frac{V_a - V_o}{R_f}$$

Karena daya impedansi Op Amp sama, maka persamaanya:

$$V_a = V_b = 0V$$

disederhanakan dari rumus diatas:

$$\frac{V_i}{R_t} = -\frac{V_o}{R_f}$$

$$-\frac{V_o}{V_i} = \frac{R_f}{R_t}$$

Maka di dapat rumus penguatan Op amp 5 kali

$$V_o = -\frac{R_f}{R_t} \cdot V_i$$

Chip AVR Arduino memiliki 10 bit AD, jadi ini resolusi simulasi modul  $0,00489 \text{ V}$  ( $5 \text{ V} / 1023$ ), dan tegangan input dari modul ini harus lebih dari  $0,00489 \text{ V} \times 5 = 0,02445 \text{ V}$ .

Catu daya merupakan faktor penting dalam suatu sistem rangkaian elektronika, pemilihan catu daya yang baik akan mempengaruhi kinerja dari robot tersebut. Baterai Lithium Polymer atau biasa juga disebut Lipo merupakan catu daya yang baik karena termasuk catu daya arus searah dan juga merupakan baterai *rechargeable*.

Daya yang dimiliki baterai lipo adalah  $3,7\text{V}$  untuk nominal 1 cell dari kimia LIPO, Cell merupakan teknologi konversi energi elektrokimia yang mampu mengubah senyawa hidrogen dan oksigen menjadi air, dan dalam prosesnya menghasilkan listrik. Tegangan dapat dinaikan sampai pada pengisian (charging) maksimum pada  $4,25\text{V}$  dan dapat diturunkan sampai minimum  $3,0\text{V}$ . Range dengan minimum  $3,7\text{V}$  sampai pada maksimum  $4,2\text{V}$  adalah range yang paling baik digunakan untuk menjaga kemampuan kimiawi dari cell baterai LiPo tersebut agar tidak mudah rusak

Untuk baterai 3 cell tegangan yang dipasang seri adalah  $3 \times 3,7\text{V} = 11,1\text{V}$ . oleh karena itu baterai LiPo yang digunakan memiliki tegangan  $11,1\text{V}$  dengan arus sebesar  $2200\text{mAh}$ . Jika tegangan baterai turun mendekati batas tegangan  $9\text{V}$  maka pemakaian baterai harus dihentikan atau dilepas dan harus diisi ulang agar melebihi tegangan  $9\text{V}$ .

untuk *charging time* pada masing-masing jenis alat *charge* sebenarnya mempunyai perhitungan dasar ideal yang dapat dlhitung dengan menggunakan persamaan (3-1).

$$t_h = \text{mahB} / \text{mA}C \quad (3-1)$$

Keterangan:

mahB = kapasitas maksimum baterai

mA C = besarnya *ampere* yang diberikan *charger*

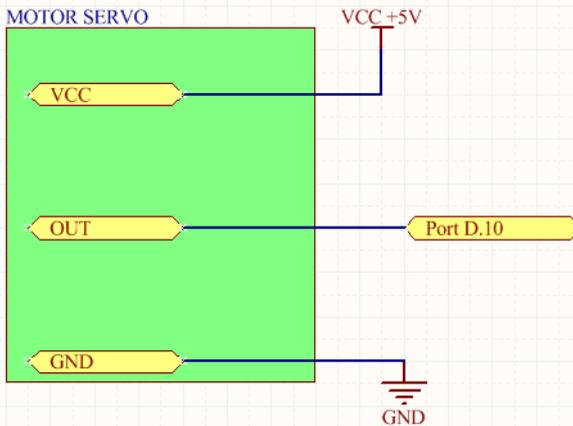
t<sub>h</sub> = total waktu dalam jam

Proses pengisian baterai terlalu lama. Pada beberapa baterai. terdapat arus pengisian maksimum. jadi tidak terlalu berpatokan pada prinsip pengisian di atas, seperti harus mengisi 5% sampai 100% dari kapasitas arus baterai. Baterai dapat diisi dengan arus pengisian maksimum dari baterai, sehingga waktu

pengisian tidak terlalu lama.

### 3.7 Perancangan Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah, dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. perancangan motor servo 360 derajat atau continue servo dilakukan dengan memberikan sejumlah pulsa pada input motor servo, dan delay untuk menentukan beberapa derajat servo akan berputar.

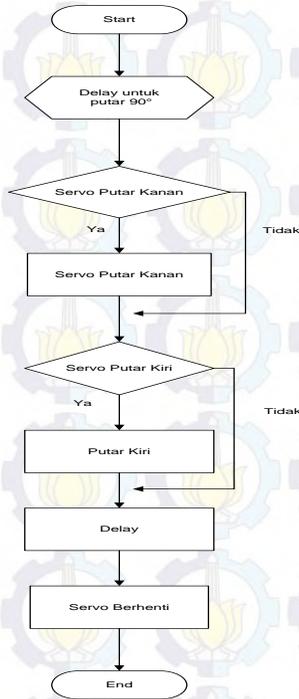


**Gambar 3.12** hubungan antara Motor servo dan mikrokontroler

| No | Arduino | Servo   |
|----|---------|---------|
| 1  | Pin-A0  | PIND-10 |

**Tabel 3.7** Hubungan antar Motor Servo

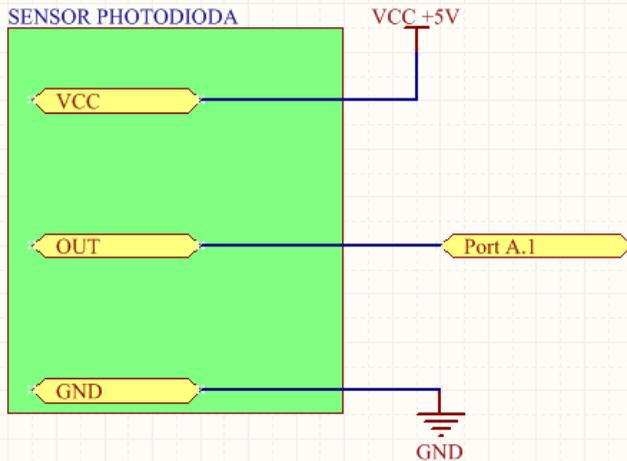
Motor servo hanya memiliki 3 kabel yang mana masing-masing fungsinya terdiri dari positif (*Vcc*), negatif (*Ground*) dan kontrol (*Signal*). Motor servo mampu bergerak searah jarum jam ataupun berlawanan arah jarum jam tanpa membalik pin konektor pada motor servo, hal ini disebabkan bahwa pada motor servo telah terdapat *driver* untuk membalik polaritas motor DC yang ada pada motor servo. Konfigurasi pin pada motor servo dapat dilihat pada gambar 2.21



**Gambar 3.13** Flowchart pergerakan motor servo

### 3.8 Perancangan Module Sensor Photodiode

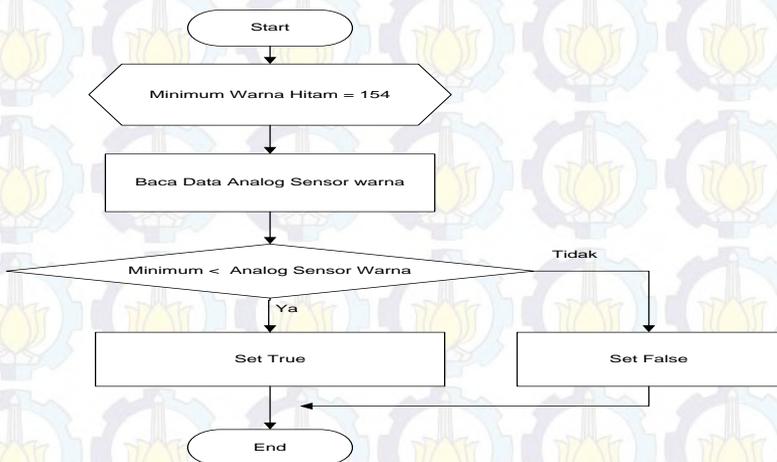
Prinsip kerja dari Module sensor photodiode , Memancarkan Cahaya Infra Merah untuk di pantulkan kembali ke sensor Photodiode . Cahaya yang di pantulkan mempengaruhi output tegangan . Pemantulan Cahaya sesuai dengan warna dari alas yang memantulkan. Karakteristik Module Sensor Photodiode Jika terkena warna Gelap memiliki Output Tegangan 3.3V dan Warna Terang 1 V.



**Gambar 3.14** hubungan antara Sensor Photodioda dan mikrokontroler

| No | Arduino | Sensor Photodioda |
|----|---------|-------------------|
| 1  | Pin-A1  | Out               |

**Tabel 3.8** Hubungan antar Sensor Photodioda



**Gambar 3.15** Diagram alir pembacaan sensor photodioda.

### 3.9 Perancangan Sistem Controller dan Interface Sistem.

Mikrokontroller di dalam perancangan ini merupakan komponen utama, dan merupakan otak dari sistem agar dapat bekerja dengan baik dan optimal.

Mikrokontroller ATmega2560 digunakan sebagai kendali utama sistem. alat ini harus didukung oleh beberapa rangkaian penunjang agar dapat melakukan fungsinya, antara lain rangkaian *clock*. Selain itu juga harus ditentukan *port-portnya* untuk rangkaian pendukung yang lain.

Agar pengiriman data dari masing-masing bagaian dapat terjadi maka diperlukan interface. Pada komunikasi data antara PC ke MEGA256 interface yang digunakan adalah USB. Sedangkan interface sensor rotari digunakan pin output interup pada Mega2560 untuk output sensor channel B, dan digunakan port output untuk sensor channel A. Sedangkan untuk mengendalikan driver motor menggunakan PORT mode output pada Arduino dan mode PWM untuk mengatur kecepatan motor. untuk sensor ultrasonic digunakan penghindar objek. kemudian sensor tegangan menggunakan port ADC digunakan sebagai indikasi baterai pada robot. untuk motor servo digunakan untuk sebagai tempat dari IP cam yang akan berputar. dan yang terakhir sensor photodioda untuk perjalanan menuju *docking station*.

#### 1. Port A

*Port A.0* (pin0) digunakan sebagai *port* masukan dan keluaran (I/O) untuk pembacaan ADC dari sensor tegangan.

*Port A.1* (pin1) digunakan sebagai *port* masukan dan keluaran (I/O) untuk pembacaan ADC dari sensor photodioda.

*Port A.0* (pin10) digunakan sebagai *port* masukan dan keluaran (I/O) pengiriman data pulsa ke motor servo.

#### 2. Port D

*Port D.2 – Port D.5* (pin 14-16) untuk digunakan sebagai interup dari rotary encoder

*Port D.34 – Port D.42* (pin 34-42) digunakan sebagai *port* masukan dan keluaran (I/O) pengiriman data sensor ultrasonic.

*Port D.22 – Port D.32* (pin 22-32) digunakan sebagai *port* masukan dan keluaran (I/O) pembacaan LCD data sensor ultrasonic.

Port D.22 – Port D.32 (pin 22-32) digunakan sebagai port masukan dan keluaran (I/O) pembacaan LCD data sensor ultrasonic.

3. Pin (Vcc) dihubungkan dengan tegangan baterai lewat regulator +5 volt
4. Pin (GND) dihubungkan dengan tegangan ground pada baterai

### 3.10 Perancangan IP CAM

*IP Camera DCS-930L* adalah kamera pengintai yang dapat diakses menggunakan jaringan TCP/IP *network*. Kamera ini memiliki slot untuk RJ-45 ethernet, dimana apabila terhubung ke jaringan akan mendapatkan *IP address* sendiri, yang mana kita dapat mengakses untuk melihat kamera tersebut dari komputer-komputer yang terhubung ke jaringan melalui Internet Explorer. Untuk penginstalasi maka dibutuhkan router sebagai penghubung antara PC dan IP CAM

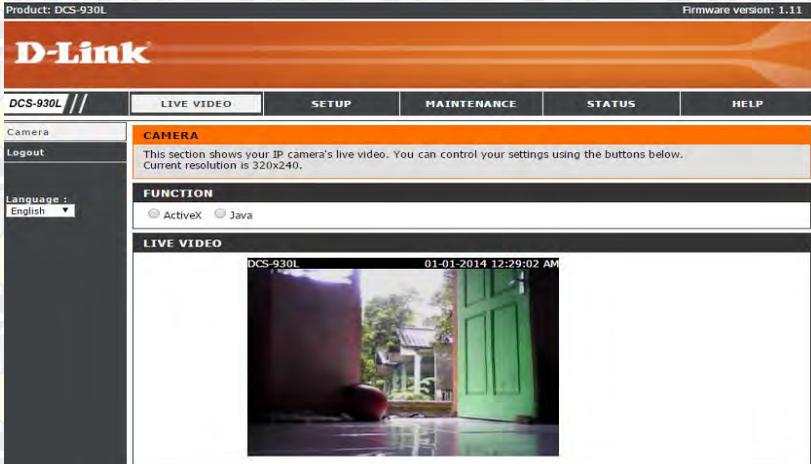
Router yang memiliki kemampuan melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur diantara keduanya. Proses routing dilakukan secara hop by hop. IP tidak mengetahui jalur keseluruhan menuju tujuan setiap paket. IP routing hanya menyediakan IP address dari router berikutnya yang menurutnya lebih dekat ke host tujuan.



**Gambar 3.16** hubungan antara ip cam dengan router.

perancangan IP cam dilakukan dengan menghubungkan ip cam dengan router yang telah terkoneksi internet, lalu menghubungkannya router yang sudah terkoneksi dengan ip cam ke PC dengan menggunakan wifi. setelah ip cam terhubung dengan PC, lalu hasil view

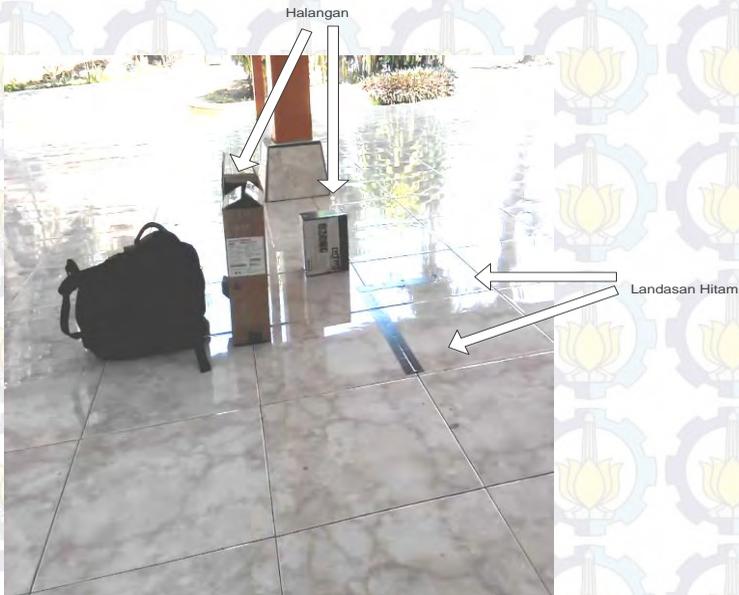
dari ip cam dapat diakses dengan alamat <http://192.168.0.100/top.htm> pada web browser. seperti pada gambar 3.17



Gambar 3.17 tampilan ip cam pada web browser

### 3.11 Perancangan pergerakan robot

perancangan robot dengan alas tempat yang sudah disediakan robot bergerak maju dan kanan saja, hal ini dikarenakan karena sensor tegangan berubah ubah, sehingga saat robot melakukan penulusaran robot cepat kembali dikarenakan baterai robot akan habis. panjang lintasan gerak robot saat maju 200 cm dan kanan 75 cm. pada proses ini robot akan berjalan maju dari *docking station* dengan posisi *docking station* berada pada sebelah kiri ruangan / seperti pada perancangan. Pada setiap sudut terdapat landasan hitam, hal ini digunakan agar robot mencoba menempatkan ke posisi lurus dan jalan untuk selanjutnya tidak berbelok.



**Gambar 3.18** tempat simulasi percobaan robot

Penjelasan:

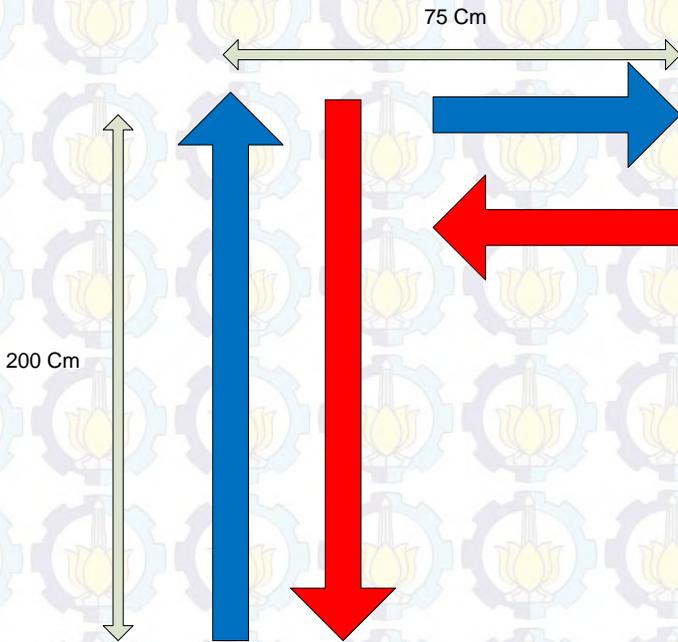
1. landasan hitam : untuk mensempurnakan posisi robot dan me reset rotary encoder
2. kerdus : sebagai halangan dan robot agar bergerak ke kiri

jika kondisi baterai mendekati habis atau sudah mencapai batas yang ditentukan, maka robot akan bergerak kembali menuju *docking station* dengan arah yang berlawanan dengan arah saat robot menyusuri tempat yang sudah disediakan. pada saat proses robot kembali menuju *docking station*, yang digunakan sebagai acuan adalah rotary encoder

pergerakan robot pulang ke *docking station* sama dengan pergerakan robot saat melakukan penelusuran, hal ini dikarenakan karena pembacaan pulsa rotary encoder saat penelusuran di simpan di array, bila baterai mendekati habis maka akan program akan memanggil rotary yang disimpan pada array, robot akan berjalan pada *docking station* sesuai dengan data array. pada pembacaan rotary encoder terdapat nilai error, nilai error tersebut akan akan dimanfaatkan oleh

modul sensor photodioda yang dimana untuk menangani error pembacaan terhadap rotary encoder yang diakibatkan slip As pada omni wheel.

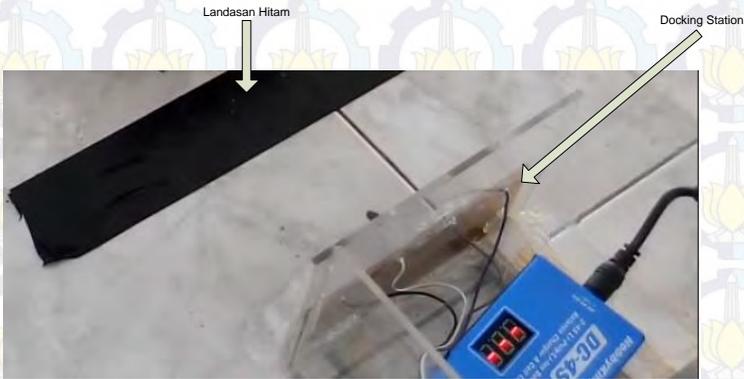
jadi pada saat robot bergerak dengan alas tempat yang sudah disediakan, rotary encoder akan menghitung pulsa pada setiap putaran rodanya.



**Gambar 3.19** ilustrasi dari pergerakan robot

Penjelasan:

1. warna biru : arah robot menyusuri ruangan
2. warna merah : arah robot saat kembali pulang



**Gambar 3.20** tempat *docking station*

Penjelasan:

1. landasan hitam : mencocokkan posisi robot agar tepat di *docking station*
2. *docking station* : tempat untuk men *charge* robot

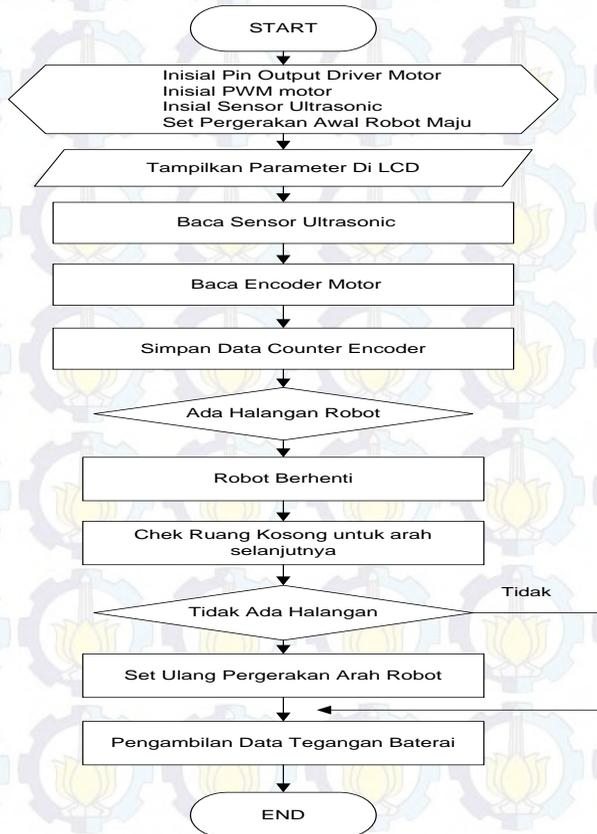


**Gambar 3.21** tempat percobaan robot

ketika baterai habis pada saat robot bergerak saat bergerak maju dengan pembacaan rotary encoder, maka robot akan bergerak mundur dengan counter pembacaan rotary encoder, lalu robot akan berhenti. Pada saat mendekati *docking station* saat kembali sensor photo dioda belum mendeteksi adanya landasan warna hitam, maka pergerakan robot akan dilanjutkan kearah belakang dan akan berhenti lagi ketika sensor photodiode mendeteksi landasan warna hitam. Lalu agar konektor baterai pada robot dapat terhubung dengan konektor *charger* pada *docking station*, digunakan push button, agar robot berhenti dan melakukan proses *charger* dan juga di robot terdapat pir yang gunanya untuk mencocokkan antar konektor pir pada robot dan kepingian PCB pada *docking station*, untuk keperluan *charger* baterai.

### 3.12 Diagram Alir

Untuk memberikan gambaran umum jalannya program dan memudahkan pembuatan perangkat lunak, maka dibuat diagram alir yang menunjukkan jalannya program, diagram alir yang menunjukkan jalannya program, diagram alir terdiri dari blok-blok dan simbol yang dihubungkan dengan anak panah. Setiap blok mengandung penjelasan tentang pekerjaan yang dilakukan oleh mikrokontroller



**Gambar 3.22** diagram alir sistem pergerakan robot



**Gambar 3.23** diagram alir sistem saat menuju *docking station*

## **BAB IV**

### **PENGUJIAN DAN ANALISIS**

Pengujian dalam tugas akhir ini dilakukan untuk mengetahui kinerja sistem. Pengujian sistem ini terdiri dari :

1. Pengujian Rotary encoder.
2. Pengujian Motor DC.
3. Pengujian Driver Motor
4. Pengujian sensor Ultrasonic.
5. Pengujian LCD.
6. Pengujian sensor tegangan.
7. Pengujian Motor Servo
8. Pengujian Sensor Photodiode.
9. Pengujian IP CAM.
10. Pengujian sistem secara keseluruhan

#### **4.1. Pengujian Rotary Encoder.**

Pengujian rotary encoder dilakukan dengan melakukan pengambilan data pulsa pada satu rotasi. Dengan membandingkan dengan datasheet akan didapatkan eror pembacaan pulsa pada rotary encoder tersebut. Berikut adalah hasil dari pengujian rotary encoder yang ditunjukkan oleh tabel 4.1.

| No | Percobaan ke | Rotary encoder 1 | Rotary encoder 2 |
|----|--------------|------------------|------------------|
| 1  | 1            | 204              | 202              |
| 2  | 2            | 208              | 208              |
| 3  | 3            | 215              | 205              |
| 4  | 4            | 202              | 202              |
| 5  | 5            | 211              | 209              |
| 6  | 6            | 204              | 204              |

**Tabel 4.1.** hasil pengujian rotary encoder

Dari enam kali uji coba pengambilan data rotary encoder pada satu kali rotasi didapatkan nilai yang berbeda-beda, dari hasil itu dapat dibuat perhitungan eror rata-rata pembacaan pulsa pada tiap kali rotary encoder berotasi. Berikut adalah hasil perhitungan eror rata-ratanya dengan jumlah pulsa pada masing-masing rotary encodernya adalah 200 pulsa (pada datasheet):

Rotary encoder 1 (A38S6 – made in china)

$$\text{error rata - rata} = \frac{\text{percobaan } 1 + 2 + 3 + 4 + 5 + 6}{6}$$

$$\begin{aligned} \text{error rata - rata} &= \frac{204 + 208 + 215 + 202 + 211 + 204}{6} \\ &= 207 - 200 = 7 \text{ pulsa} \end{aligned}$$

Rotary encoder 2 (*autonic*)

$$\text{error rata - rata} = \frac{\text{percobaan } 1 + 2 + 3 + 4 + 5 + 6}{6}$$

$$\begin{aligned} \text{error rata - rata} &= \frac{202 + 208 + 205 + 202 + 209 + 204}{6} \\ &= 205 - 200 = 5 \text{ pulsa} \end{aligned}$$

Dari hasil perhitungan rata-rata eror dapat dianalisa bahwa rotary encoder *autonic* memiliki tingkat eror pembacaan yang lebih kecil dari pada merek china. Dan dari hasil eror tersebut juga dapat disimpulkan apabila pada saat robot kembali menuju *docking station* akan mengalami gangguan dalam penempatannya pada *docking station*, untuk itu diperlukan sensor photodiode untuk membantu robot menuju *docking station* ketika robot sudah dekat dengan *docking station*.

#### 4.2. Pengujian Motor DC.

Pengujian Motor DC ini dilakukan untuk mengetahui arah putaran motor DC terhadap perubahan teganganyang dibolak balik dan juga dilakukan dengan menguji kecepatannya dengan PWM pada program yang dibuat. Tabel 4.2 adalah pengujian motor DC terhadap arah putaran dan 4.3 adalah pengujian PWM motor DC.

**Tabel 4.2** pengujian motor DC terhadap arah putaran.

| No. | Sumber tegangan | Pin Motor | Arah Putar                  |
|-----|-----------------|-----------|-----------------------------|
| 1   | Positif         | Positif   | Searah Jarum Jam            |
| 2   | Negatif         | Negatif   |                             |
| 3   | Positif         | Negatif   | Berlawanan dengan jarum jam |
| 4   | Negatif         | Positif   |                             |

dari hasil pengujian diatas didapatkan hasil bahwa motor DC telah dapat berputar sesuai dengan teori bahwa ketika pin positif motor diberi tegangan negatif maka motor akan berputar clock wise / searah jarum jam, begitu juga sebaliknya pin motor diberi tegangan kebalikannya

**Tabel 4.3** pengujian PWM motor DC.

| No. | PWM | Kecepatan Relatif  |
|-----|-----|--------------------|
| 1   | 255 | Maksimal           |
| 2   | 175 | Kecepatan maksimal |

pada pengujian motor DC yang kedua dilakukan dengan menguji motor dengan PWM. dan hasil dari pengujian didapatkan hasil bahwa motor dapat merespon PWM yang diberikan oleh program. pada pengujian ini tidak membahas berapa RPM motor karena pada perancangan robot ini tidak ada sensor yang digunakan untuk mengukur kecepatan motor

#### 4.3. Pengujian Driver Motor

Pengujian ini digunakan untuk pergerakan dari motor DC. hasil dari pergerakan motor DC ini diterapkan pada gerak robot.

**Tabel 4.4** Pengujian Driver Motor.

| No. | Input 1 | Input 2 | Enable 1 | Kondisi motor |
|-----|---------|---------|----------|---------------|
| 1   | High    | Low     | High     | CW            |
| 2   | Low     | High    | High     | CCW           |
| 3   | High    | High    | High     | Stop          |
| 4   | Low     | Low     | High     | Stop          |
| 5   | High    | Low     | Low      | Stop          |
| 6   | Low     | High    | Low      | Stop          |
| 7   | High    | High    | Low      | Stop          |
| 8   | Low     | Low     | Low      | Stop          |

Dari hasil pengujian driver motor L293D diatas dapat terlihat bahwa arah putaran motor ditentukan oleh input 1, input 2 dan enable nya. pada enable itulah pwm akan digunakan untuk mengatur kecepatan putar motor

#### 4.4. Pengujian sensor ultrasonic

Pengujian sensor ultrasonic dilakukan untuk pendeteksi apabila ada benda yang ada di depannya

**Tabel 4.5** Pengujian sensor ultrasonic

| jarak yang terbaca (cm) |         |          |             | jarak sesungguhnya (cm) |         |          |             | Error (cm) |
|-------------------------|---------|----------|-------------|-------------------------|---------|----------|-------------|------------|
| UI kanan                | UI kiri | UI depan | UI belakang | UI kanan                | UI kiri | UI depan | UI belakang |            |
| 10                      | 10      | 10       | 10          | 10                      | 10      | 10       | 10          | 0          |
| 20                      | 20      | 20       | 20          | 20                      | 20      | 20       | 20          | 0          |
| 30                      | 30      | 30       | 30          | 30                      | 30      | 30       | 30          | 0          |
| 40                      | 40      | 40       | 40          | 39,5                    | 39,5    | 39,5     | 39,5        | 0,5        |
| 50                      | 50      | 50       | 50          | 49,5                    | 49,5    | 49,5     | 49,5        | 0,5        |
| 60                      | 60      | 60       | 60          | 59,5                    | 59,5    | 59,5     | 59,5        | 0,5        |
| 70                      | 70      | 70       | 70          | 69                      | 69      | 69       | 69          | 1          |
| 80                      | 80      | 80       | 80          | 79                      | 79      | 79       | 79          | 1          |
| 90                      | 90      | 90       | 90          | 89                      | 89      | 89       | 89          | 1          |
| 100                     | 100     | 99,5     | 99,5        | 98,5                    | 98,5    | 98,5     | 98,5        | 1,5        |
| Rata rata error (cm)    |         |          |             |                         |         |          |             | 0,6        |

Tabel 4.5 menunjukkan besarnya pengukuran jarak obyek dengan menggunakan sensor ultrasonic, besarnya persentase error pada pengukuran kurang dari 30 cm nilainya 100 pengukuran dengan jarak 40 cm s/d 100 cm nilai persentase errornya kurang dari 0,6. Sehingga dari hasil pengukuran ini sensor kinect memiliki nilai jarak minimum pengukuran adalah 30 cm dan jarak maximum pengukuran adalah 100 cm.

#### 4.5 Pengujian LCD

Berdasarkan hasil pengujian yang telah dijalankan, dapat disimpulkan bahwa minimum sistem dan LCD dalam kondisi baik, hal ini ditunjukkan dengan tampilan pada LCD. Hasil dari pengujian minimum sistem ini ditunjukkan pada Gambar 4.1



**Gambar 4.1** Hasil pengujian LCD

#### 4.6 Pengujian Sensor Tegangan

Untuk pengujian sensor tegangan adalah dengan mengukur tegangan keluaran pada output sensor menggunakan avo meter dan melihat nilai ADC yang terbaca oleh arduino. Pada pengujian ini input dari sensor adalah langsung dari baterai lipo 11.1 volt 2200mAh. Output dari sensor adalah 5 volt, sedangkan tegangan input sensor maksimal adalah 25 V atau 2500 mV. Berikut adalah hasil pengujian sensor tegangan yang ditunjukkan tabel 4.6

**Tabel 4.6** pengujian sensor tegangan

| No                  | Tegangan baterai | Tegangan output sensor | Nilai ADC | Tegangan baterai sesungguhnya | Error (V) |
|---------------------|------------------|------------------------|-----------|-------------------------------|-----------|
| 1                   | 11.1 V           | 2.19 V                 | 458       | 11,5 V                        | 0,4 V     |
| 2                   | 10.8 V           | 2.16 V                 | 442       | 11,3 V                        | 0,5 V     |
| 3                   | 10.5 V           | 2.13 V                 | 436       | 10,9 V                        | 0,4 V     |
| Rata rata error (V) |                  |                        |           |                               | 0,43 V    |

Nilai adc dari sensor tegangan tersebut didapatkan dari pembacaan nilai analog pada arduino dan dikonversi menjadi tegangan.

$25000 / 1024 = 24.43$  adc per mV

Jadi rumus yang dimasukkan dalam program adalah:

$(24.43 \times \text{nilai ADC}) / 1000$  agar menjadi satuan volt

Namun pada keadaan realnya, nilai adc dari pembacaan sensor tegangan tersebut berubah-ubah, hal ini bisa dikarenakan sensor yang digunakan adalah sensor rakitan sendiri yang kemungkinan memiliki akurasi yang kecil jika dibandingkan dengan modul sensor tegangan buatan pabrik

#### 4.7 Pengujian Motor Servo

pengujian motor servo 360 derajat atau continue servo dilakukan dengan memberikan sejumlah pulsa pada output motor servo, dan delay untuk menentukan beberapa derajat servo akan berputar.

**Tabel 4.7** pengujian motor servo

| No. | Pulsa | Delay   | Kondisi Servo     | Pengukuran sudut dengan busur |
|-----|-------|---------|-------------------|-------------------------------|
| 1   | 1700  | 800 ms  | Berputar CCW 90°  | Berputar CCW 89°              |
| 2   | 1300  | 800 ms  | Berputar CW 90°   | Berputar CW 89°               |
| 3   | 1500  | 2000 ms | Berhenti          | -                             |
| 4   | 1700  | 1600 ms | Berputar CCW 180° | Berputar CCW 89°              |
| 5   | 1300  | 1600 ms | Berputar CW 180°  | Berputar CW 89°               |
| 6   | 1500  | 2000 ms | Berhenti          | -                             |

Dari hasil pengujian motor servo pada tabel 4.7 diatas, terlihat bahwa arah putaran motor servo, jika motor servo diberi pulsa sebesar 1700 maka motor servo akan berputar berlawanan dengan arah jarum jam (CCW), sedangkan jika motor servo diberi pulsa sebesar 1300 maka motor servo akan berputar searah jarum jam (CW), dan jika motor servo diberi pulsa 1500 maka akan berhenti atau tidak akan berputar. lalu untuk mengatur beberapa besar sudut putar motor servo menggunakan delay pada program yang ditentukan secara trial and error. dengan mencoba memberikan delay 800ms, motor servo berputar sebesar 90°. namun pada putaran berikutnya motor servo mengalami error putaran sebesar 1% pada tiap putarannya.

#### 4.8 Pengujian Sensor Photodioda

Pengujian sensor photodioda digunakan untuk mengarahkan sensor pada bidang datar dengan dua warna yang berbeda yaitu hitam dan putih untuk melihat nilai tegangan output dari sensor. berikut adalah hasil uji coba dari sensor photodioda yang ditunjukkan tabel 4.8

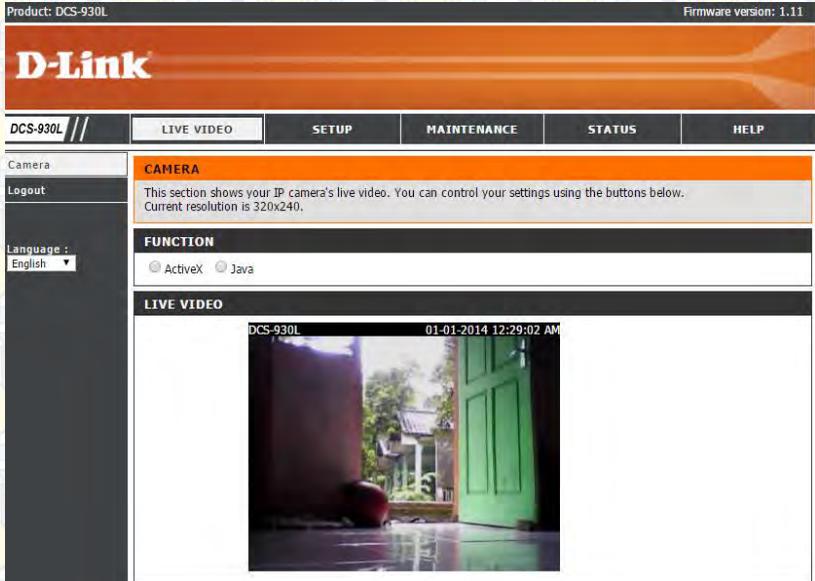
**Tabel 4.8** pengujian sensor photodioda

| No. | Warna Lantai | Tegangan Output  |
|-----|--------------|------------------|
| 1   | Putih        | 0 sampai 50mV    |
| 2   | Hitam        | 4,8 V sampai 5 V |

dari hasil pengujian diatas dapat dianalisa bahwa sensor photodioda akan aktif high jika bidang pantulnya berwarna hitam (warna gelap) dan bernilai low jika bidang pantulannya berwarna putih. jadi dengan karakteristiknya itu dapat digunakan untuk membantu proses *autodocking* apabila robot sudah mendekati *docking stationnya*.

#### 4.9 Pengujian IP cam

pengujian IP cam dilakukan dengan menghubungkan ip cam dengan router yang telah terkoneksi internet, lalu menghubungkannya router yang sudah terkoneksi dengan ip cam ke PC dengan menggunakan wifi. setelah ip cam terhubung dengan PC, lalu hasil view dari ip cam dapat diakses dengan alamat <http://192.168.0.100/top.htm> pada web browser. seperti pada gambar 4.2

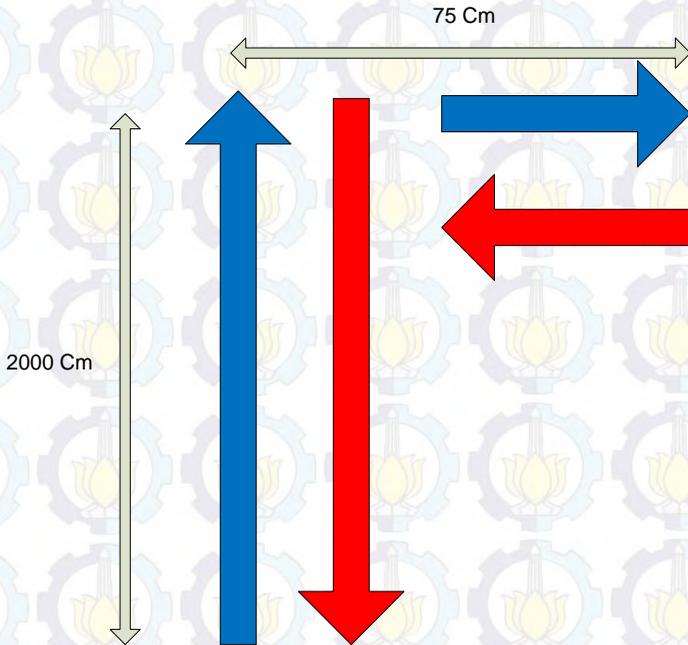


**Gambar 4.2** tampilan pengujian ip cam pada web browser

dari hasil uji coba diatas didapatkan kesimpulan bahwa ip cam telah berhasil diakses jarak jauh dengan menggunakan wifi dan internet pada router

## 4.10 Pengujian sistem secara keseluruhan

### 4.10.1 pengujian jalan robot menyusuri tempat yang sudah disediakan



**Gambar 4.3** ilustrasi pergerakan robot

Penjelasan:

1. garis biru : jalur pergerakan robot, baterai sudah terisi
2. garis merah : jalur pergerakan robot, batterai akan habis
3. garis abu abu: jarak pergerakan robot

pengujian sistem secara keseluruhan yang pertama adalah dengan menguji jalannya robot menyusuri tempat yang sudah disediakan. pada proses ini robot akan berjalan maju dari *docking station* dengan posisi *docking station* berada pada sebelah kiri ruangan / seperti pada perancangan. berikut adalah hasil yang ditunjukkan oleh tabel 4.9

**Tabel 4.9** pengujian robot saat menyusuri tempat yang sudah disediakan

| No. | Arah jalan robot | Ultrasonic yg aktif | Motor yang aktif | Arah putar motor       | Arah putar servo |
|-----|------------------|---------------------|------------------|------------------------|------------------|
| 1   | Maju             | Depan               | Kanan & kiri     | Kanan: CCW<br>Kiri: CW | CW 90°           |
| 2   | Kanan            | Kanan               | Depan & belakang | Depan: CCW<br>Kiri: CW | CW 90°           |

dari hasil pengujian pada tabel 4.9 diatas dapat dijelaskan bahwa robot mula-mula bergerak maju dari sudut kiri ruangan, lalu saat ultrasonic depan mendeteksi adanya halangan maka robot akan bergerak kekananan. Pada setiap sudut terdapat landasan hitam, hal ini digunakan agar robot mencoba menempatkan ke posisi lurus dan jalan untuk selanjutnya tidak berbelok. dengan begitu robot akan selalu berputar menyusuri tempat yang sudah disediakan selama baterai belum terdeteksi habis

#### **4.10.2 pengujian jalan robot menuju pulang**

pengujian sistem keseluruhan yang kedua adalah ketika baterai pada robot mendekati habis atau mencapai batas yang sudah ditentukan. jika kondisi baterai mendekati habis atau sudah mencapai batas yang ditentukan, maka robot akan bergerak kembali menuju *docking station* dengan arah yang berlawanan dengan arah saat robot menyusuri tempat yang sudah disediakan. pada saat proses robot kembali menuju *docking station*, yang digunakan sebagai acuan adalah rotary encoder

jadi pada saat robot bergerak menyusuri tempat yang sudah disediakan, rotary encoder akan menghitung pulsa pada setiap putaran rodanya. berikut adalah hasil pengujian dari sistem jalan robot saat baterai akan habis.

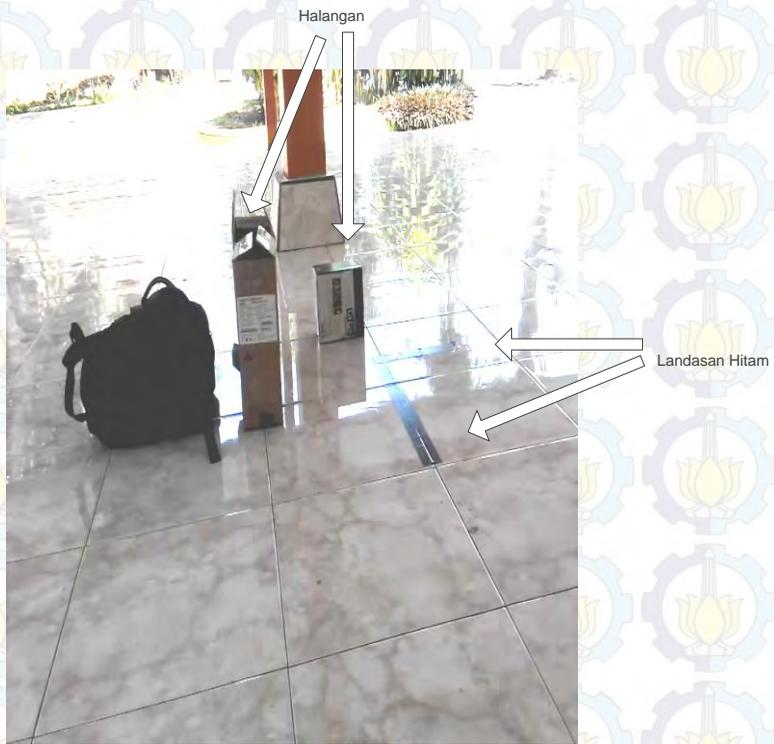
**Tabel 4.10** pengujian robot saat menuju pulang

| No. | Tegangan baterai | Kondisi baterai | Pergerakan robot | Arah pulsa rotary 1 (maju) | Arah pulsa rotary 2 (kanan) |
|-----|------------------|-----------------|------------------|----------------------------|-----------------------------|
| 1   | 12,5             | penuh           | Maju             | 8549                       | 0                           |
| 2   | 12,3             |                 | Kanan            | 8549                       | 8347                        |
| 4   | 12,1             | Akan habis      | Kiri             | 8549                       | 0                           |
| 6   | 12,0             |                 | Mundur           | 0                          | 0                           |

dari hasil uji coba yang ditunjukkan pada tabel 4.10 di atas dapat dijelaskan bahwa pada saat robot melakukan pergerakan menyusuri tempat yang sudah disediakan, program menyediakan 4 aray sebagai tempat penyimpanan data pulsa dari rotary encoder. untuk setiap arah pergerakan robot menentukan di aray mana pulsa akan disimpan untuk sementara. jika dilihat dari hasil pengujian diatas, terlihat bahwa ketika robot melakukan putaran pertama, setiap data pulsa yang terbaca di masukan dalam masing masing arah yang sudah ditentukan. dan pada saat putaran kedua pada saat robot bergerak maju , kondisi baterai habis. maka pada saat robot akan berhenti sejenak dan akan berjalan mundur sebanyak jumlah pulsa pada saat maju tadi menuju *docking station* untuk melakukan pengecasan, dan juga semua isi aray akan dihapus setiap kali robot selesai melakukan putaran menyusuri tempat yang sudah disediakan.

#### **4.10.3 pengujian jalan robot pada saat *autodocking***

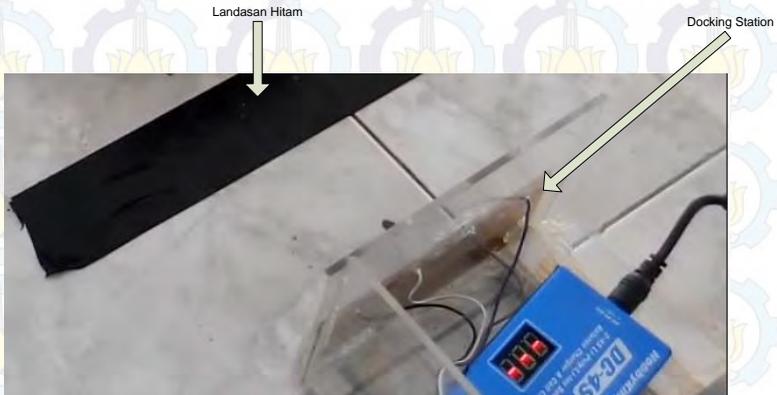
untuk proses *autodocking* robot akan dibantu oleh sensor photodiode agar dapat kembali dengan tepat menuju *docking station* nya.



**Gambar 4.4** tempat simulasi percobaan robot

Penjelasan:

1. landasan hitam : untuk mensempurnakan posisi robot dan me reset rotary encoder
2. kerdus : sebagai halangan dan robot agar bergerak ke kiri



**Gambar 4.5** tempat *docking station*

Penjelasan:

1. landasan hitam : mencocokkan posisi robot agar tepat di *docking station*
2. *docking station* : tempat untuk men *charge* robot

Seperti hasil pengujian pada point dua (sistem robot pada saat baterai akan habis), ketika baterai mendekati tegangan 12,1 V robot akan bergerak kekiri dengan counnter sebesar 8347, lalu robot akan bergerak mundur dengan counter sebesar 8549 lalu robot akan berhenti. Namun jika sensor photo dioda belum mendeteksi adanya landasan warna hitam, maka pergerakan robot akan dilanjutkan kearah belakang dan akan berhenti lagi ketika sensor photodiode mendeteksi landasan warna hitam. robot akan berhenti lagi tepat didepan *docking station*. Lalu agar konektor baterai pada robot dapat terhubung dengan konektor charger pada *docking station*, robot akan bergerak mundur degann delay 1000 ms, dan robot akan melakukan charging pada *docking station*, digunakan push button, agar motor berhenti dan melakukan charger.

## BAB V PENUTUP

Setelah dilakukan rangkaian kegiatan perancangan sistem dan pengujian alat, penulis memperoleh kesimpulan dan memberikan beberapa saran sebagai berikut.

### 5.1. Kesimpulan

Kesimpulan yang diperoleh dalam Tugas Akhir ini adalah:

- a. Sistem navigasi menggunakan rotary encoder dapat dianalisa bahwa rotary encoder *autonic* memiliki tingkat error rata rata sebesar 5 pulsa, pembacaan pulsa lebih kecil dari pada merek china yaitu memiliki error rata sebesar 7 pulsa.
- b. Hasil error terhadap rotary encoder juga dapat disimpulkan apabila pada saat robot kembali menuju *docking station* akan mengalami gangguan dalam penempatannya pada *docking station*, untuk itu diperlukan sensor photodioda untuk membantu robot menuju *docking station* ketika robot sudah dekat dengan *docking station*.
- c. Jarak pembacaan pada sensor ultrasonic memiliki error rata rata sebesar 0,6 cm.
- d. Pada Motor DC menggunakan PWM, hal ini bertujuan untuk agar gerak dari robot itu bisa lurus.
- e. Nilai tegangan dari pembacaan sensor tegangan tersebut berubah-ubah dan memiliki error sebesar 0,43 V
- f. motor servo berputar sebesar 90° dan memiliki error putaran sebesar 1% pada tiap putarannya.
- g. ip cam harus diakses dengan perangkat keras lainnya yaitu menggunakan wifi dan internet pada router
- h. Pada percobaan tidak bisa dilakukan di tempat biasa saja dikarenakan landasannya tidak rata, maka harus di tentukan tempat dengan landasan yang rata

### 5.2. Saran

Beberapa saran yang dapat penulis berikan untuk pengembangan Tugas Akhir adalah:

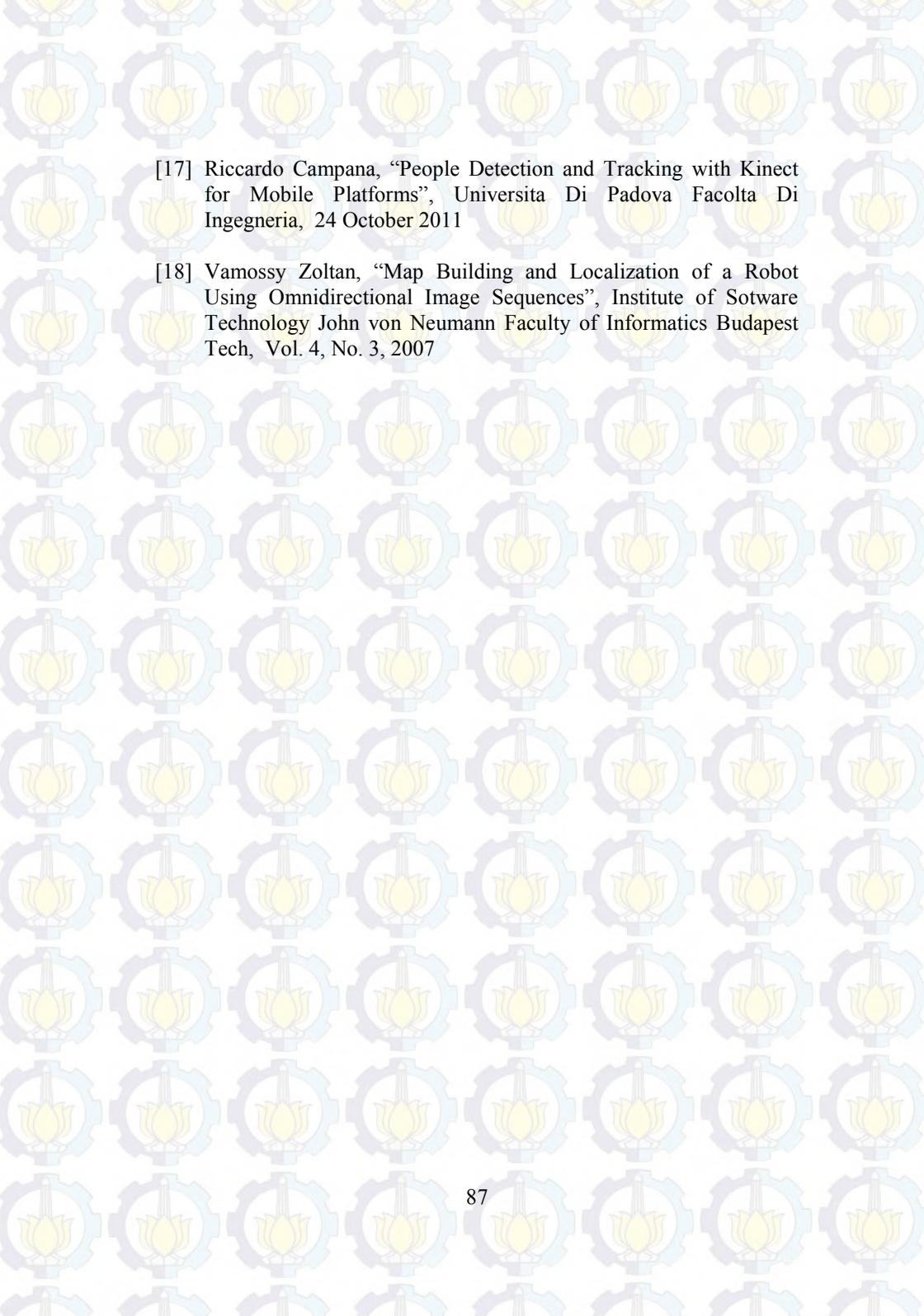
- a. Sitem navigasi menggunakan rotary encoder rata rata memiliki error yang berbeda beda kedepannya dapat dikembangkan agar memiliki error pembacaan yang sedikit.

- 
- b. pada saat menuju *docking station* robot untuk kedepannya dapat menggunakan kamera untuk image processing sebagai *autodocking*
  - c. pada saat robot berada di *docking station* sangat lama, karena melakukan proses pen chargeran untuk kedepannya dapat dikontrol menggunakan pc agar tidak lama robot berada di *docking station*

## DAFTAR PUSTAKA

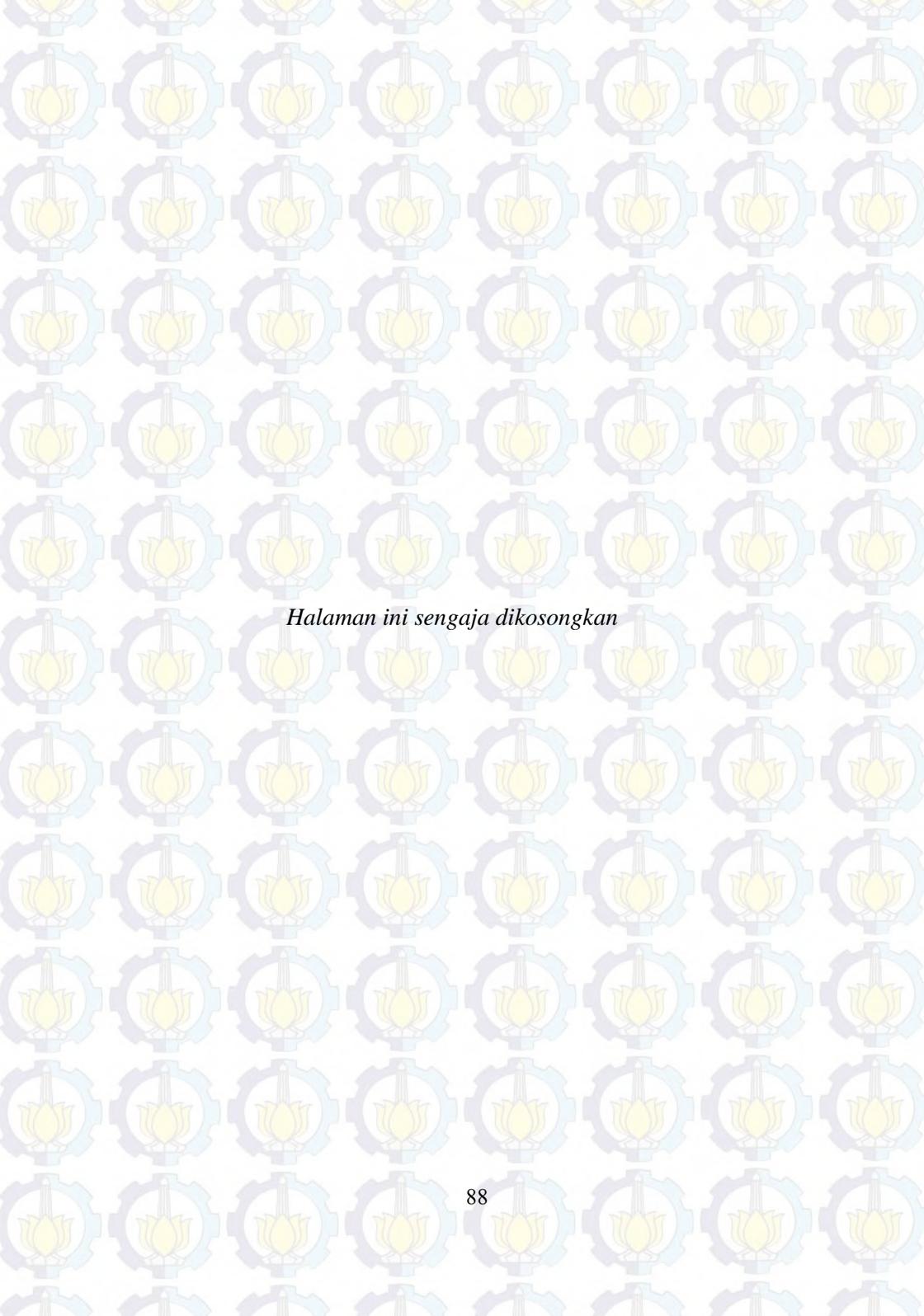
- [1] Tsai Ching-Chih, Jiang Li-Bin, Wang Tai-Yu, Wang Tung-Sheng, “Kinematics Control of an Omnidirectional Mobile Robot”, Proceedings of 2005 CACS Automatic Control Conference Tainan, Taiwan, Nov 18-19,2005
- [2] AL-Amri A. Salam, and Ahmed Iman, “Control of Omni-Directional Mobile Robot Motion”, Al-Khwarizmi Engineering Journal, Vol. 6, No. 4, PP 1-9, 2010
- [3] Sulistyoyo J, Widyanto T, Gustica Abi .N, "Penentuan Koordinat Posisi Robot Pada Bidang Kartesian", Laporan Kerja Praktek, Institut Teknologi Sepuluh Nopember, 11 Nopember 2013
- [4] Marta Bayu Sandi, Ardilla Fernando, Besari A.R. Anom, “Path Tracking Pada Mobile Robot Dengan Umpan Balik Odometri”, Politeknik Negeri Surabaya, Surabaya 2010
- [5] Kusumah H, "Perancangan alat uji masa hidup dan analisis reliabilitas baterai", Tugas Akhir, Universitas Komputer Indonesia, 21 Mei 2013
- [6] Rizal S, “NiCd battery charger berbasis MAX713 dengan pengontrolan mikrokontroler aTmega128”, Tesis, Universitas Indonesia, 2011
- [7] Prastya B, “Rancang bangun charger laptop pada mobil ”, Tugas Akhir, Institut Teknologi Sepuluh Nopember, Surabaya 10 Maret 2010
- [8] Yuliani L, “Rancang Bangun AC-DC Battery Charger Pada Penyediaan Daya Cadangan Rumah Tangga”, Politeknik Negeri Surabaya, Surabaya 2007

- [9] Jonathan Fox, Alasdair Muntz, “Robotic Table Football”<URL: <http://www.strath.ac.uk/eee/studenttradeshows2014/robotictablefootball/>>, 2012
- [10] Pradhan Ninad, “Mobile Robot Navigation for Person Following in Indoor Environments” Graduate School of Clemson University, August 2013
- [11] Benavidez Patrick, and Jamshidi Mo, ”Mobile Robot Navigation and Target Tracking System ”, Proc. of the 2011 6th International Conference on System of System Engineering, Albuquerque, New Mexico, USA –June 27-30, 2011
- [12] Biswas Joydeep, and Veloso Manuela, “Depth Camera Based Indoor Mobile Robot Localization and Navigation”, IEE International Conference on Robotics and Automation RiverCentre, Saint Paul, Minnesota, USA May 14-18, 2012
- [13] Bailey and Durrant Hugh, “Simultaneous Localization and Mapping(SLAM)”, IEEE Robotics and Automation Magazine, September 2006
- [14] Wolf Denis F, and Sukhatme Gaurav S, “Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments”, Autonomous Robots 19, 53-65, 2005
- [15] S. Thrun, and A. Bucken, “Learning maps for indoor mobile robot navigation”, Technical Report CMU-CS-96-121, Carnegie Mellon University, School of Computer Science, Pittsburgh, P A 15213, April 1996
- [16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking”, In Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11), pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society



[17] Riccardo Campana, “People Detection and Tracking with Kinect for Mobile Platforms”, Università Di Padova Facoltà Di Ingegneria, 24 October 2011

[18] Vamossy Zoltan, “Map Building and Localization of a Robot Using Omnidirectional Image Sequences”, Institute of Software Technology John von Neumann Faculty of Informatics Budapest Tech, Vol. 4, No. 3, 2007



*Halaman ini sengaja dikosongkan*

## LAMPIRAN

Program Arduino IDE

```
#include <LiquidCrystal.h> //library LCD
#include <NewPing.h> // library ultrasonic
#include <Servo.h> //library servo
Servo myServo;

LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
//ultrasonic depan
#define TRIGGER_PIN_1 42 // deklarasi pin triger pada ultrasonic
#define ECHO_PIN_1 44 // deklarasi pin echo pada ultrasonic
#define MAX_DISTANCE_1 200 // deklarasi jarak maksimal
jangkauan ultrasonic

//iltrasonic kanan
#define TRIGGER_PIN_2 48 // deklarasi pin triger pada ultrasonic
#define ECHO_PIN_2 46 // deklarasi pin echo pada ultrasonic
#define MAX_DISTANCE_2 200 // deklarasi jarak maksimal
jangkauan ultrasonic

//ultrasonic kiri
#define TRIGGER_PIN_3 40 // deklarasi pin triger pada ultrasonic
#define ECHO_PIN_3 38 // deklarasi pin echo pada ultrasonic
#define MAX_DISTANCE_3 200 // deklarasi jarak maksimal
jangkauan ultrasonic

//ultrasonic belakang
#define TRIGGER_PIN_4 36 // deklarasi pin triger pada ultrasonic
#define ECHO_PIN_4 34 // deklarasi pin echo pada ultrasonic
#define MAX_DISTANCE_4 200 // deklarasi jarak maksimal
jangkauan ultrasonic

NewPing sonar_depan(TRIGGER_PIN_1, ECHO_PIN_1,
MAX_DISTANCE_1); // deklarasi penggunaan variabel pada
ultrasonic depan
NewPing sonar_kanan(TRIGGER_PIN_2, ECHO_PIN_2,
MAX_DISTANCE_2); // deklarasi penggunaan variabel pada
ultrasonic kanan
```

```

NewPing      sonar_kiri(TRIGGER_PIN_3,      ECHO_PIN_3,
MAX_DISTANCE_3); // deklarasi penggunaan variabel pada
ultrasonic kiri
NewPing      sonar_belakang(TRIGGER_PIN_4,  ECHO_PIN_4,
MAX_DISTANCE_4); // deklarasi penggunaan variabel pada ultrasonic
belakang

//deklarasi pin L293d pada arduino
int enable_satu = 8;
int enable_dua = 6;
int enable_tiga = 7;
int enable_empat = 9;
int A_satu= 33;
int A_dua = 37;
int A_tiga = 51;
int A_empat = 43;
int B_satu = 31;
int B_dua = 39;
int B_tiga = 49;
int B_empat = 45;

int val;
//deklarasi pin encoder pada pin arduino
int encoder0PinA = 4;
int encoder0PinB = 5;
int encoder1PinA = 3;
int encoder1PinB = 2;

//deklarasi variabel tambahan untuk encoder
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int encoder1Pos = 0;
int encoder1PinALast = LOW;
int n = LOW;
int m = LOW;

unsigned long intervalPing = 0;
double depan = 0;
double kanan = 0 ;
double kiri = 0 ;
double belakang = 0 ;

```

```
//jarak minimal tabrakan objek dengan sensor ping satuan cm  
double jarakminimal = 27 ;
```

```
//variabel untuk mengunci pergerakan  
bool lockmotorchangemovemant = true ;
```

```
//variabel status arduino  
int statusMove = 1 ;
```

```
//variabel tegangan baterai  
bool lockactivesimulatebat = false ;  
bool lockChek = false ;  
bool simulationActive = false ;  
int countPutar = 0 ;
```

```
//berapa kali sudut putaran  
int jumlahPutaran = 1 ;  
double voltageBaterai = 14 ;  
double minimal_tegangan = 11.5 ;  
unsigned long timeBat = millis();
```

```
//waktu delay baterai habis  
unsigned long maxtimeBat = 2000;
```

```
unsigned long timeLcd = millis();  
unsigned long time = millis();
```

```
double pwm = 200;
```

```
double counting = 1 ;
```

```
//save rotary encoder  
int count = 1 ;  
int rotary[4] = { 0 , 0 , 0 , 0 } ;  
double saveIdentify[4];
```

```
int warnahitam = 0 ;
```

```
double milisServoRun = 0.00 ;  
double chekMillis = millis();
```

```

bool servoRunNow = false ;
bool isServoRun = false ;
int indexServoRun = 0 ;

unsigned long timeChekEnco = millis();
unsigned long maxTm = 200 ;

int lastEnco = 0 ;
int RotaryPerCode[] = { 0 , 0 , 0 , 0 } ;

unsigned long dataTimeEncode[] = { 0 , 0 , 0 , 0 } ;

void setup()
{
  Serial.begin(9600); //deklarasi baudrate komunikasi serial
  lcd.begin(16, 2);
  //loopShowBat();
  myServo.attach(10); // output servo pada pin 9
  insialmotor();
  //deklarasi pin arduino untuk encoder digunakan sebagai input
  pinMode (encoder0PinA,INPUT);
  pinMode (encoder0PinB,INPUT);
  pinMode (encoder1PinA,INPUT);
  pinMode (encoder1PinB,INPUT);

  jumlahPutaran = jumlahPutaran * 4 ;
  jumlahPutaran = jumlahPutaran - 1 ;

  time= millis();
  getBlackColor();
  delay(1000); //delay 1 detik untuk memberikan waktu sebentar
  setelah tombol di nyalakan , agar user menarik tngan
  startright();
  startmaju();
}

//fungsi insial motor -> fungsi ini untuk insial pin digital output
//---> Proses Awal Set Inisial Pin sebagai Output
//---> Proses Kedua Set Low Pada semua Output . tujuan agar motor
tidak menyala

```

//---> Proses Ketiga Set Output analog untuk mengendalikan Kecepatan motor dengan PWM

```
void insialmotor() {
```

```
//---> Proses Awal
```

```
pinMode(enable_satu, OUTPUT);  
pinMode(enable_dua, OUTPUT);  
pinMode(enable_tiga, OUTPUT);  
pinMode(enable_empat, OUTPUT);  
pinMode(A_satu, OUTPUT);  
pinMode(A_dua, OUTPUT);  
pinMode(A_tiga, OUTPUT);  
pinMode(A_empat, OUTPUT);  
pinMode(B_satu, OUTPUT);  
pinMode(B_dua, OUTPUT);  
pinMode(B_tiga, OUTPUT);  
pinMode(B_empat, OUTPUT);
```

```
//---> Proses Kedua
```

```
digitalWrite(A_satu, LOW);  
digitalWrite(A_dua, LOW);  
digitalWrite(B_satu, LOW);  
digitalWrite(B_dua, LOW);  
digitalWrite(A_tiga, LOW);  
digitalWrite(A_empat, LOW);  
digitalWrite(B_tiga, LOW);  
digitalWrite(B_empat, LOW);
```

```
//--->Proses Ketiga
```

```
analogWrite(enable_satu, 180); //rotary encoder 400rpm ( kiri )  
analogWrite(enable_dua, 150); //motor 400rpm ( kanan )  
analogWrite(enable_tiga, 150); //motor 300 rpm ( depan )  
analogWrite(enable_empat, 180); //rotary encoder 300rpm ( belakang )  
}
```

//fungsi startright . Fungsi ini akan diaktifkan ketika robot mulai menjalankan program]

//---> lakukan pergerakan kekiri . Sampai Terdeteksi Halangan Pada ultrasonic kiri

//---> Proses Melalui Tahapanj Pengecekan Awal Ping Jika Tidak terdapat halangan Maka Program Bisa lanjut

```
void startright() {  
  off();delay(100);  
  while(kiri < jarakminimal){
```

```

    readpingsensor2());
}
right();
bool stsColusion = chekColusionByLockMovement(); //chek jika
terjadi tabrakan atau halangan
while( !stsColusion ){
    stsColusion = chekColusionByLockMovement();
    readpingsensor2();
}
off();runServoxx1( 90 , 1 );delay(100);
}

```

//fungsi startmaju . fungsi program robot keliling . atau program yang memulai robot untuk berkeliling

```

void startmaju(){
off();delay(100);
while(depan < jarakminimal){
    readpingsensor2());
}
maju();
start2chekRotariPer();
}

```

```

void pwmBatFull()
{
analogWrite(enable_satu, 174); //rotary encoder 400rpm ( kiri )
analogWrite(enable_dua, 146); //motor 400rpm ( kanan )
analogWrite(enable_tiga, 146); //motor 300 rpm ( depan )
analogWrite(enable_empat, 174); //rotary encoder 300rpm ( belakang )
}

```

```

void pwmBatLow()
{
analogWrite(enable_satu, 145); //rotary encoder 400rpm ( kiri )
analogWrite(enable_dua, 118); //motor 400rpm ( kanan )
analogWrite(enable_tiga, 115); //motor 300 rpm ( depan )
analogWrite(enable_empat, 145); //rotary encoder 300rpm ( belakang )
}

```

```

void pwmAfterHb()

```

```

{
  analogWrite(enable_satu, 200); //rotary encoder 400rpm ( kiri )
  analogWrite(enable_dua, 170); //motor 400rpm ( kanan )
  analogWrite(enable_tiga, 170); //motor 300 rpm ( depan )
  analogWrite(enable_empat, 200); //rotary encoder 300rpm ( belakang )
}

void loop()
{
  pwmBatFull();
  virtualBat();
  if( ( millis() - timeLcd ) > 100 ){ //delay waktu untuk timer update 600
ms sekali , atau 0.6 detik sekali
    timeLcd = millis();
    lcd.clear();
    ShowBat();
    showping2serial();
    sensorPing();
    //show2Terminal();
    //showDataTime2Terminal();
    //showSaveMove();
  }
  if(lockmotorchangemovemant){readpingsensor();} //baca sensor ping
hanya ketika motor sedang berjalan , bukan saat pergantian arah robot
saat menemui halangan
  readencoder(); //baca encoder
  saveMoveMant(); //panggil fungsi untuk mneyimpan data encoder
  if( voltageBaterai < minimal_tegangan ){ //jika tegangan robot telah
mencapai tegangan minimum
    pwmBatLow();
    robotCameback(); //panggil kembali robot ke strat awal ( maju )
    back2homebase(); //panggil fungsi ini untuk memandu robot kembali
ke home base
    runServoxx1( 90 , 0 );

    DelayRobotAlready2HomeBase(5000); //delay robot sebesar 5000
detik untuk memberikan waktu simulasi baterai telah terisi
    resetSomeParameterRobot(); //reset beberapa parameter agar robot
siap di aktifkan kembali
    pwmAfterHb();
    majuAwalKembaliKeHomeBase();

```

```

pwmBatFull();
startright();off();delay(300);
startmaju();delay(400);
readpingsensor2();delay(500);
maxtimeBat = 2300 ;
} else {
  robotmove(); //panggil fungsi untuk keliling . saat baterai full maka
  robot berkeliling
}
}
}

```

```

void majuAwalKembaliKeHomeBase(){
  maju();delay(700);
  off();delay(300);
}

```

//fungsi setServoActive di gunakan untuk mengaktifkan servo

```

void setServoActive( double derajat ){
  isServoRun = true ;
  servoRunNow = true ;
  milisServoRun = derajat2miliServo( derajat );
}

```

//fungsi runServo di gunakan untuk menjalankan servo

```

void runServo(){
  if( !isServoRun ){
    if( servoRunNow ){
      isServoRun = true ;
      chekMillis = millis();
      putarServo( indexServoRun );
    }
  } else {
    if( (millis() - chekMillis) > milisServoRun ){
      isServoRun = false ;
      servoRunNow = false ;
      putarServo( 2 );
    }
  }
}
}
}

```

//fungsi runServoxx1 di gunakan untuk menjalankan motor servo sebesar x derajat .

//---> index putar 0 untuk memutar berlawanan arah jarum jam

//---> index putar 1 untuk memutar se arah jarum jam

```
void runServoxx1( double derajat , int indexPutar ){  
    double timeDelay = derajat2miliServo( derajat );  
    putarServo( indexPutar );  
    delay( timeDelay );  
    putarServo( 2 ); //servo berhenti  
    delay( 200 );  
}
```

//fungsi derajat2miliServo digunakan untuk merubah data derajat ke dalam mili second servo akan di putar

```
double derajat2miliServo( double derajat ){  
    double sender = 300 ;  
    double constantDer = 90.00 ;  
    return derajat * ( sender / constantDer ) ;  
}
```

//fungsi putarServo di gunakan untuk memberikan perintah pergerakan servo

//---> lebar pulsa 1700 untuk pergerakan servo arah berlawanan

//---> lebar pulsa 1300 untuk pergerakan servo searah jarum jam

//---> lebar pulsa 1500 untuk memberhentikan servo

```
void putarServo( int index ){  
    int data = 0 ;  
    if( index == 0 ){  
        data = 1700 ;  
    }else if( index == 1 ){  
        data = 1300 ;  
    }else if( index == 2 ){  
        data = 1500 ;  
    }  
    myServo.writeMicroseconds(data);  
}
```

//fungsi DelayRobotAlready2HomeBase di gunakan untuk memberhentikan robot sementara waktu

```
void DelayRobotAlready2HomeBase( unsigned long maxTime){  
    unsigned long tms = millis();
```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Charging");
while((millis() - tms ) < maxTime ){}
}

```

//fungsi resetSomeParameterRobot di gunakan untuk mereset beberapa parameter

//--> peresetan parameter guna mengaktifkan program agar robot siap kembali keliling

```

void resetSomeParameterRobot(){
    voltageBaterai = 14 ;
    encoder0Pos = 0 ;
    encoder1Pos = 0 ;
    for( int i = 0 ; i < 4 ; i++ ){
        rotary[i] = 0 ;
    }
    readpingsensor2();
    while(depan < jarakminimal){
        readpingsensor2();
    }
    timeBat = millis();

```

```

    lockactivesimulatebat = false ;
    lockChek = false ;
    simulationActive = false ;
    countPutar = 0 ;
    voltageBaterai = 14 ;
    minimal_tegangan = 12 ;
    timeBat = millis();
    maxtimeBat = 2400;
}

```

//fungsi getBlackColor --> tujuan untuk mengambil data awal warna hitam

//nilai awal akan disimpan dalam variabel double warnahitam

//nilai hasil pengambilan data di kyurangi sebesar 30% .

//-> pengambilan guna menurunkan nilai kemungkinan dari pengecekan warna hitam

//-> jika didapat nilai analog sebesar 300 maka dituurnkan menjadi 300 \* ( 1 - 0.3 )

//-> nilai yang disimpan 210 . jika di temui lagi nilai diatas ini maka di anggap hitam

```
void getBlackColor(){
    double persen_pengurangan = 0.3 ;
    warnahitam = analogRead( 2 );
    warnahitam = warnahitam * ( 1 - persen_pengurangan );
    setWarnaHitam( 154 );
    //proTestWarna();
}
```

```
void setWarnaHitam( double value ){
    warnahitam = value ;
}
```

```
void proTestWarna(){
    while( true ){
        Serial.print( analogRead( 2 ) ); Serial.print("\n");
        Serial.print( warnahitam ); Serial.print("\n");
        if( isBlack() ){
            Serial.print("Hitam \n");
        }else{
            Serial.print("Putih \n");
        }
        delay(2000);
    }
}
```

//fungsi isBlack . fungsi ini mengecek warna papan atau lantai  
//---> Jika Warna Lantai Putih Maka Output Dari fungsi Adalah False  
//---> Jika Warna Lantai Hitam Maka Output Dari fungsi Adalah True  
//---> Untuk pengecekan Fungsi Memakai data warnahitam .

```
bool isBlack(){
    bool sts = false ;
    if( warnahitam < analogRead( 2 ) ){
        sts = true ;
    }
    return sts;
}
```

//fungsi back2homebase di gunakan untuk kembali ke posisi awal robot .

//--> fungsi ini diaktifkan ketika pembacaan ingatan rotari telah selesai di laksanakan

//--> karena tempat awal robot mulai maju berbeda . ( program robot baru diaktifkan ketika program geser kiri selesai di laksanakan )

//--> program ini akan memandu robot kembali ketempat semula .

```
void back2homebase() {
  if( !isBlack() ) {
    maju();
    while( !isBlack() ) {}
    off();
  }
  if( isBlack() ) {
    off(); delay(1000);
    left(); delay(100);
    while( isBlack() ) {} ;
    off(); delay(1000);
    mundur(); delay(100); while( !isButtonPressed() ) {} ;
    off();
  }
}
```

//fungsi saveMoveMant di gunakan untuk penyimpanan data rotary encoder tiap pergerakan robot

//--> rotary index 0 untuk menyimpan jumlah perpindahan robot maju

//--> rotary index 1 untuk menyimpan jumlah perpindahan robot mundur

//--> rotary index 2 untuk menyimpan jumlah perpindahan robot kanan

//--> rotary index 3 untuk menyimpan jumlah perpindahan robot kiri

```
void saveMoveMant() {
  if( statusMove == 0 || statusMove == 1 ) {
    rotary[ statusMove ] = encoder0Pos ;
  } else if( statusMove == 2 || statusMove == 3 ) {
    rotary[ statusMove ] = encoder1Pos ;
  }
}
```

//fungsi saveMoveMant1 sama dengan saveMoveMant

//---> fungsi yang bertugas untuk menyimpan data encoder .

//---> perbedaan adalah program ini khusus untuk robot saat kembali ke home base

```
void saveMoveMant1(int idxs) {
  if( idxs == 0 || idxs == 1 ) {
```

```

    rotary[ statusMove ] = encoder0Pos ;
} else if ( idxs == 2 || idxs == 3 ) {
    rotary[ statusMove ] = encoder1Pos ;
}
}

```

//fungsi showSaveMove di gunakan untuk menampilkan data rotari encoder pada serial port

```

void showSaveMove(){
    Serial.println("-----");
    Serial.println( rotary[0] );
    Serial.println( rotary[1] );
    Serial.println( rotary[2] );
    Serial.println( rotary[3] );
    Serial.println( encoder0Pos );
    Serial.println( encoder1Pos );
    Serial.println( statusMove - 1 );
}

```

```

void ShowBat(){
    lcd.setCursor(0,0);
    lcd.print("E:");
    lcd.setCursor(3,0);
    lcd.print( voltageBaterai );
    lcd.setCursor(14,0);
    lcd.print("V");
    Serial.print("----- encoder ----- \n");
    Serial.print(encoder0Pos); Serial.print("\n");
    Serial.print(encoder1Pos); Serial.print("\n");
}

```

//fungsi virtualBat digunakan untuk monitoring data tegangan robot sekarang

//--> program ini memberikan parameter robot tegangan sekarang

//--> data tegangan di simpan pada variabel voltageBaterai

```

void virtualBat(){
    //if( (millis() - timeBat ) > maxtimeBat ){ //chek jika baterai telah melebihi waktu yang telah di tentukan
    //voltageBaterai = 9.9; //
    //}
    voltageBaterai = (23.43 * analogRead(7)-0.9) / 1000;
}

```

```

    voltageBaterai = voltageBaterai + 2.1 ;
}

//fungsi robotmove di gunakan untuk robot bergerak keliling area .
//----> fungsi yang diaktifkan ketika robot belum kehabisan baterai .
//----> fungsi ini inti dari program keliling . terdapat beberapa fungsi
didalamnya .
//-----> 1. fungsi untuk pengecekan halangan pada saat berjalan .
//-----> 2. fungsi untuk pengecekan bahwa langkah robot selanjutnya
ketika ada halangan kosong
//-----> 3. fungsi untuk memindahkan pergerakan saat ada halangan
void robotmove(){
    if( lockmotorchangemovemant ){ // cek jika motor sedang bergerak
        bool stsColusion = cekColusionByLockMovement(); //cek jika
        terjadi tabrakan atau halangan
        if( stsColusion ){ //jika terjadi halangan set parameter untuk mencari
        posisi terbaru
            lockmotorchangemovemant = false ;
            off();
            runServoxx1( 90 , 1 ); //panggil fungsi untuk menggerakkan servo
            endMonSet();
            countPutar = countPutar + 1 ;
            delay(1000);

            encoder0Pos = 0 ;
            encoder1Pos = 0 ;
        }
    }else{ // terjadi halngan jadi robot akan mencari posisi terbaru untuk
    melanjutkan perjalanan
        int indexArahBaru = cekFreeSpace(); //cek ruang kosong tersedia
        setmovemotor( indexArahBaru );
        if( countPutar <= jumlahPutaran ){delayPing();ChekPingSensor(
        indexArahBaru );}
        //start2cekRotariPer();
        //startMonSet();
        lockmotorchangemovemant = true ;
    }
}

void delayPing(){
    if( statusMove == 0 || statusMove == 1 ){

```

```

    delay(200);
} else if( statusMove == 2 || statusMove == 3 ){
    delay(200 );
}
}

```

//fungsi robotCameback di gunakan saat robot telah kehabisan baterai  
//---> fungsi ini akan emandu robot kembali ke home base dengan membaca data encoder  
//---> fungsi ini memiliki prosedur membaca data .  
//---> data encoder akan disimpan dalam buffer encoder . setelah itu data simpanan dalam rotary akan di hapus  
//---> saat proses selanjutnya data buffer encoder akan di dibandingkan dengan data rotary yang baru . mengenai pergerakan nya  
//---> pembandingan ini memberikan gambaran pada robot berapa kali roda berputar dan pergerakan maju , mundur , kiri atau kanan

```

void robotCameback(){
    int data1[] = { 0 , 3 , 1 , 2 } ;
    int data2[] = { 1 , 2 , 0 , 3 } ;
    int index = 4 ;

    for( int i = 0 ; i < 4 ; i++ ){
        if( statusMove == data1[i] ){
            index = i ;
            break ;
        }
    }
    off();delay(100);
    bool lockre = true ;
    for( int i = index ; i >= 0 ; i-- ){
        int idx1 = data2[ i ] ;
        setmovemotor( idx1 ) ;
        if( !lockre ){while( isBlack() )}
        while( !isBlack() ){}
        off();delay(100);
        lockre = false ;
    }
}
}

```

//fungsi readpingsensor digunakan untuk membaca sensor ping .

//---> fungsi ini hanya akan melakukan pengecekan data sebanyak 1 kali dalam jangka waktu 200 mili second

//---> pemberian waktu membaca sensor ping . guna memberikan waktu untuk sensor encoder membaca pergerakan

//---> Data hasil pembacaan di simpan dalam variabel jarak ( depan , kanan , kiri dan belakang )

```
void readpingsensor(){
  if( (millis() - intervalPing) > 20 ){
    intervalPing = millis();
    unsigned int uS1 = 0 ;
    if( countping == 1 ){
      uS1 = sonar_depan.ping();
    }else if( countping == 2 ){
      uS1 = sonar_kanan.ping();
    }else if( countping == 3 ){
      uS1 = sonar_kiri.ping();
    }else if( countping == 4 ){
      uS1 = sonar_belakang.ping();
    }
    double jarak = uS1 / US_ROUNDTRIP_CM ;
    if( countping == 1 ){
      if( jarak > 0 ){ depan = jarak ; }
    }else if( countping == 2 ){
      if( jarak > 0 ){ kanan = jarak ; }
    }else if( countping == 3 ){
      if( jarak > 0 ){ kiri = jarak ; }
    }else if( countping == 4 ){
      if( jarak > 0 ){ belakang = jarak ; }
    }
    if( countping == 4 ){
      countping = 1 ;
    }else{
      countping = countping + 1 ;
    }
  }
}
```

//fungsi readpingsensor2 di gunakan untuk membaca sensor ping

//---> fungsi ini sama dengan fungsi readpingsensor . perbedaan tidak ada waktu delay .

//---> fungsi ini hanya di pakai dalam keadaan tertentu . yang membutuhkan update data cepat

```
void readpingsensor2(){  
    unsigned int uS1 = sonar_depan.ping();delay(2);  
    unsigned int uS2 = sonar_kanan.ping();delay(2);  
    unsigned int uS3 = sonar_kiri.ping();delay(2);  
    unsigned int uS4 = sonar_belakang.ping();delay(2);
```

```
    double bdepan = uS1 / US_ROUNDTRIP_CM ;  
    if( bdepan > 0 ){ depan = bdepan ; }  
    double bkanan = uS2 / US_ROUNDTRIP_CM ;  
    if( bkanan > 0 ){ kanan = bkanan ; }  
    double bkiri = uS3 / US_ROUNDTRIP_CM ;  
    if( bkiri > 0 ){ kiri = bkiri ; }  
    double bbelakang = uS4 / US_ROUNDTRIP_CM ;  
    if( bbelakang > 0 ){ belakang = bbelakang ; }  
}
```

//fungsi readencoder di gunakan untuk membaca encoder motor

//---> fungsi untuk menghitung putaran roda .

//---> data disimpan dalam encoder0Pos dan encoder1Pos

```
void readencoder(){  
    n = digitalRead(encoder0PinA);  
    if((encoder0PinALast == LOW) && (n == HIGH)) {  
        if(digitalRead(encoder0PinB) == LOW) {  
            encoder0Pos--;  
        } else {  
            encoder0Pos++;  
        }  
    }  
    encoder0PinALast = n;
```

```
    m = digitalRead(encoder1PinA);  
    if((encoder1PinALast == LOW) && (n == HIGH)) {  
        if(digitalRead(encoder1PinB) == LOW) {  
            encoder1Pos--;  
        } else {  
            encoder1Pos++;  
        }  
    }  
    encoder1PinALast = m;
```

```
}
```

```
void showing2serial(){  
  double jarak[] = { depan , belakang , kiri , kanan };  
  Serial.print("----- ping sensor ----- \n");  
  Serial.print(" Sensor Ping Depan ");Serial.print("\n");  
  Serial.print(jarak[0]);Serial.print("\n");  
  Serial.print(" Sensor Ping Belakang ");Serial.print("\n");  
  Serial.print(jarak[1]);Serial.print("\n");  
  Serial.print(" Sensor Ping kiri ");Serial.print("\n");  
  Serial.print(jarak[2]);Serial.print("\n");  
  Serial.print(" Sensor Ping kanan ");Serial.print("\n");  
  Serial.print(jarak[3]);Serial.print("\n");  
}
```

```
void sensorPing(){  
  double jarak[] = { depan , belakang , kiri , kanan };  
  double jarakMinimal[] = { 23 , 30 , 23 , 30 }; //merubah jarak batasan  
  pembacaan sensor ultrasonic  
  lcd.setCursor(0,1);  
  lcd.print("Ping:");  
  lcd.setCursor(8,1);  
  if( ( jarak[ statusMove ] > 0.1 ) && ( jarak[ statusMove ] <  
  jarakMinimal[statusMove] ) ){  
    lcd.print("ON");  
  }else{  
    lcd.print("OFF");  
  }  
}
```

```
//fungsi chekColusionByLockMovement di gunakan untuk melakukan  
pengecekan halangan ketika bergerak  
//---> fungsi ini akan mengeluarkan output berupa boolean true ketika  
ada halangan
```

```
//---> output false ketika tidak ada halangan
```

```
bool chekColusionByLockMovement(){  
  bool sts = false ;  
  double jarak[] = { depan , belakang , kiri , kanan };  
  double jarakMinimal[] = { 20 , 27 , 20 , 27 };  
  //if( ( jarak[ statusMove ] > 0.1 ) && ( jarak[ statusMove ] <  
  jarakminimal ) ){
```

```

    if( ( jarak[ statusMove ] > 0.1 ) && ( jarak[ statusMove ] <
jarakMinimal[statusMove] )){
        sts = true ;
    }
    return sts ;
}

```

//fungsi chekFreeSpace digunakan untuk mengecek ruang kosong saat robot akan menentukan pergerakan selanjutnya.

//---> fungsi akan mengeluarkan integer 0 --- > 4

//---> 0 -> 54 adalah kode langkah selanjutnya

```

int chekFreeSpace(){
    double jarak[] = { depan , belakang , kiri , kanan } ;
    double indexfirst[] = { 3 , 1 , 2 , 0 } ;
    if( statusMove == 0 ){
        indexfirst[0] = 3 ;
        indexfirst[1] = 3 ;
        indexfirst[2] = 3 ;
        indexfirst[3] = 3 ;
    }else if( statusMove == 1 ){
        indexfirst[0] = 2 ;
        indexfirst[1] = 2 ;
        indexfirst[2] = 2 ;
        indexfirst[3] = 2 ;
    }else if( statusMove == 2 ){
        indexfirst[0] = 0 ;
        indexfirst[1] = 0 ;
        indexfirst[2] = 0 ;
        indexfirst[3] = 0 ;
    }else if( statusMove == 3 ){
        indexfirst[0] = 1 ;
        indexfirst[1] = 1 ;
        indexfirst[2] = 1 ;
        indexfirst[3] = 1 ;
    }
    int index = indexfirst[1] ;
    for( int i = 0 ; i < 4 ; i++ ){
        int indx = indexfirst[i] ;
        if( jarak[indx] > ( jarakminimal + ( jarakminimal * 0.08 ) )){
            index = indx ;
            break ;
        }
    }
}

```

```

    }
}
return index ;
}

```

//fungsi setmovemotor di gunakan untuk mengaktifkan pergerakan motor

//---> fungsi akan mengambil data dari index 0 --> 3

//---> index 0 untuk maju

//---> index 1 untuk mundur

//---> index 2 untuk kiri

//---> index 3 untuk kanan

```
void setmovemotor( double indexMove ) {
```

```
int data[] = { 0 , 1 , 2 , 3 } ;
```

```
if( indexMove == data[0] ) {
```

```
    maju();
```

```
}else if( indexMove == data[1] ) {
```

```
    mundur();
```

```
}else if( indexMove == data[2] ) {
```

```
    right();
```

```
}else if( indexMove == data[3] ) {
```

```
    left();
```

```
}
```

```
}
```

//fungsi ChekPingSensor untuk mengecek kinerja sensor ping

```
void ChekPingSensor( int index ) {
```

```
int data[] = { 0 , 1 , 2 , 3 } ;
```

```
if( index == data[0] ) {
```

```
    while(depan < jarakminimal) {
```

```
        readpingsensor2();
```

```
    }
```

```
}else if( index == data[1] ) {
```

```
    while(belakang < jarakminimal) {
```

```
        readpingsensor2();
```

```
    }
```

```
}else if( index == data[2] ) {
```

```
    while(kiri < jarakminimal) {
```

```
        readpingsensor2();
```

```
    }
```

```
}else if( index == data[3] ) {
```

```
    while(kanan < jarakminimal) {
```

```

        readpingsensor2();
    }
}

void monitoringFastDataEncode(){
    chekRotariPer();
    setLastDataEncode();
}

void show2Terminal(){
    Serial.print("----- data encode per -----");
    Serial.print( RotaryPerCode[0] );
    Serial.print("\n");
    Serial.print( RotaryPerCode[1] );
    Serial.print("\n");
    Serial.print( RotaryPerCode[2] );
    Serial.print("\n");
    Serial.print( RotaryPerCode[3] );
    Serial.print("\n");
}
//fungsi untuk memulai perhitungan data encoder
void start2chekRotariPer(){
    timeChekEnco = millis();
    RotaryPerCode[ statusMove ] = 0 ;
}
//fungsi untuk melakukan perhiungan data encoder
void chekRotariPer(){
    if( (millis() - timeChekEnco) > maxTm ){
        timeChekEnco = millis();
        if( statusMove == 0 || statusMove == 1 ){
            int buff = abs(encoder0Pos) - lastEnco;
            if( buff > RotaryPerCode[ statusMove ] ){
                RotaryPerCode[ statusMove ] = buff ;
            }
        }
        else if( statusMove == 2 || statusMove == 3 ){
            int buff = abs(encoder1Pos) - lastEnco;
            if( buff > RotaryPerCode[ statusMove ] ){
                RotaryPerCode[ statusMove ] = buff ;
            }
        }
    }
}

```

```

}
}

//fungsi setLastDataEncode di gunakan untuk menset data terakhir
encoder
void setLastDataEncode(){
  if( statusMove == 0 || statusMove == 1 ){
    int lastEnco = encoder0Pos ;
  }else if( statusMove == 2 || statusMove == 3 ){
    int lastEnco = encoder1Pos;
  }
}

void showDataTime2Terminal(){
  Serial.print("----- data time Encode per -----");
  Serial.print( dataTimeEncode[0] );
  Serial.print("\n");
  Serial.print( dataTimeEncode[1] );
  Serial.print("\n");
  Serial.print( dataTimeEncode[2] );
  Serial.print("\n");
  Serial.print( dataTimeEncode[3] );
  Serial.print("\n");
}

void startMonSet(){
  dataTimeEncode[statusMove] = millis();
}

void endMonSet(){
  dataTimeEncode[statusMove] = millis() -
dataTimeEncode[statusMove];
}

//fungsi maju di gunakan untuk mengaktifkan robot untuk berjalan maju
//---> fungsi ini akan berstatus 0
void maju(){
  statusMove = 0 ;
  digitalWrite(A_satu, LOW);
  digitalWrite(A_dua, HIGH);
  digitalWrite(B_satu, HIGH);
}

```

```

digitalWrite(B_dua, LOW);

digitalWrite(A_tiga, LOW);
digitalWrite(A_empat, LOW);
digitalWrite(B_tiga, LOW);
digitalWrite(B_empat, LOW);
}

//fungsi mundur di gunakan untuk mengaktifkan robot untuk berjalan
mundur
//---> fungsi ini akan berstatus 1
void mundur(){
    statusMove = 1 ;
    digitalWrite(A_satu, HIGH);
    digitalWrite(A_dua, LOW);
    digitalWrite(B_satu, LOW);
    digitalWrite(B_dua, HIGH);

    digitalWrite(A_tiga, LOW);
    digitalWrite(A_empat, LOW);
    digitalWrite(B_tiga, LOW);
    digitalWrite(B_empat, LOW);
}

//fungsi right di gunakan untuk mengaktifkan robot untuk berjalan ke
arah kiri
//---> fungsi ini akan berstatus 2
void right(){
    statusMove = 2 ;
    digitalWrite(A_satu, LOW);
    digitalWrite(A_dua, LOW);
    digitalWrite(B_satu, LOW);
    digitalWrite(B_dua, LOW);

    digitalWrite(A_tiga, LOW);
    digitalWrite(A_empat, HIGH);
    digitalWrite(B_tiga, HIGH);
    digitalWrite(B_empat, LOW);
}

```

//fungsi left di gunakan untuk mengaktifkan robot untuk berjalan ke arah kanan

//---> fungsi ini akan berstatus 3

```
void left(){  
    statusMove = 3 ;  
    digitalWrite(A_satu, LOW);  
    digitalWrite(A_dua, LOW);  
    digitalWrite(B_satu, LOW);  
    digitalWrite(B_dua, LOW);
```

```
    digitalWrite(A_tiga, HIGH);  
    digitalWrite(A_empat, LOW);  
    digitalWrite(B_tiga, LOW);  
    digitalWrite(B_empat, HIGH);  
}
```

//fungsi off di gunakan untuk mengaktifkan robot untuk off / berhenti

//---> fungsi ini akan berstatus tidak ada

```
void off(){  
    digitalWrite(A_satu, LOW);  
    digitalWrite(A_dua, LOW);  
    digitalWrite(B_satu, LOW);  
    digitalWrite(B_dua, LOW);
```

```
    digitalWrite(A_tiga, LOW);  
    digitalWrite(A_empat, LOW);  
    digitalWrite(B_tiga, LOW);  
    digitalWrite(B_empat, LOW);  
}
```

```
bool isButtonPressed(){  
    bool sts = false ;  
    if( 600 < analogRead( 3 ) ){  
        sts = true ;  
    }  
    return sts;  
}
```

## RIWAYAT PENULIS



Novan Eka Prayoga adalah nama lengkap penulis dengan nama panggilan Novan. Penulis dilahirkan di Malang tanggal 13 Nopember 1990 yang merupakan putra pertama dari dua bersaudara. Penulis memulai pendidikannya dari TK Bustanul alfa Malang, kemudian melanjutkan studinya di SDN Purwodadi I Malang, SMPN 11 Malang, dan SMKN 8 Malang.

Setelah menamatkan SMK, penulis melanjutkan studinya di Politeknik Negeri Malang (POLINEMA) tepatnya pada Program Studi Teknik Elektronika dan lulus pada tahun 2011. Selanjutnya penulis meneruskan studi sarjana di Teknik Elektro ITS (*Institut Teknologi Sepuluh Nopember*), kemudian fokus pada bidang studi Teknik Elektro. Pada bulan Januari 2014 penulis mengikuti seminar dan ujian Tugas Akhir sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro dari Institut Teknologi Sepuluh Nopember Surabaya.

*Email : no\_fun\_legion@yahoo.co.id*