



**PENGEMBANGAN KOMUNIKASI DAN VISUALISASI PADA ROBOT  
SEPAK BOLA BERODA**



UNIVERSITAS  
**Dinamika**

Oleh :

**Moch. Angga Putra Wicaksono**  
**16410200038**

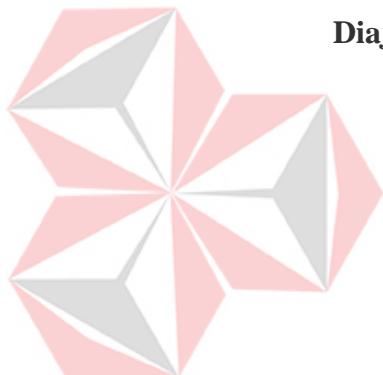
---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**UNIVERSITAS DINAMIKA**  
**2020**

**PENGEMBANGAN KOMUNIKASI DAN VISUALISASI PADA ROBOT  
SEPAK BOLA BERODA**

**TUGAS AKHIR**



Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik

UNIVERSITAS  
**Dinamika**

Oleh:

Nama : Moch. Angga Putra Wicaksono

NIM : 16410200038

Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2020**

## **TUGAS AKHIR**

## PENGEMBANGAN KOMUNIKASI DAN VISUALISASI BASE STATION PADA ROBOT SEPAK BOLA BERODA

Dipersiapkan dan disusun oleh  
**MOCH. ANGGA PUTRA WICAKSONO**  
**NIM : 16410200038**

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahasan  
Pada : 22 Januari 2020

## Susunan Dewan Pembahas

#### Pembimbing:

- I. Dr. Susijanto Tri Rasmana, S.Kom., M.T.  
NIDN: 0727097302

II. Ira Puspasari, S.Si., M.T.

Pemahasan

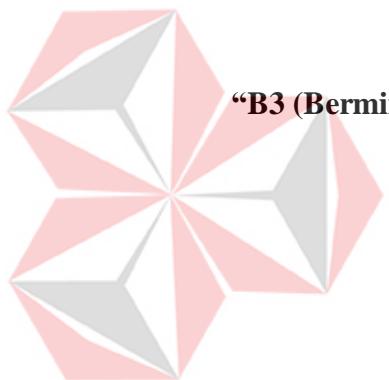
Heri Pratikno, M.T., MTCNA., MTCRE.  
NIDN: 0716117302

as  
R. Y. K. G.  
~~Y. K. G.~~  
1½<sup>20</sup>  
D. M. R.

Tugas Akhir ini telah diterima sebagai salah satu persyaratan



Dekan Fakultas Teknologi dan Informatika  
FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA



“B3 (Bermimpi, Berdoa, Berusaha) Sebuah Kunci Sebuah Kesuksesan”

UNIVERSITAS  
Dinamika  
~ Moch. Angga Putra Wicaksono ~



**Kupersembahkan Kepada**

**ALLAH SWT**

**Ayah, Ibu dan semua keluarga tercinta,**

**Yang selalu mendukung, memotivasi dan menyisipkan nama saya dalam  
doa-doa terbaiknya.**

**Beserta semua orang yang selalu membantu, mendukung dan memotivasi  
agar tetap berusaha menjadi lebih baik.**

UNIVERSITAS  
**Dindamika**

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Moch. Angga Putra Wicaksono  
NIM : 16410200038  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **Pengembangan Komunikasi Dan Visualisasi Base Station  
Pada Robot Sepak Bola Beroda**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 4 Februari 2020



Moch. Angga Putra W.

Nim : 16410200038

## ABSTRAK

Pengendalian robot secara manual membuat pergerakan robot KRSBI menjadi tidak fleksibel dan tidak relevan seiring berjalananya waktu. Dengan penelitian ini bisa menjawab dari permasalahan tersebut. Robot dapat langsung mengetahui perintah dari *RefereeBox*, aplikasi juga mampu memberikan informasi status robot membawa bola yang nantinya dapat digunakan sebagai strategi bagi robot. Selain itu visualisasi dibuat sangat *smooth* sejalan dengan jalan robot yang sebenarnya. Pada tugas akhir ini penulis mendapatkan hasil yang diperoleh dari proses uji validasi komunikasi antara robot dengan aplikasi *base station*, terbukti data yang dikirimkan dan diterima memiliki kesamaan dan tingkat uji coba tersebut adalah 100%, hasil yang diperoleh dari hasil uji kesamaan pergerakan robot sebenarnya dengan pergerakan robot pada aplikasi *base station* dengan tingkat rata-rata *error* yang tidak besar dan dapat ditoleransi dengan ketentuan 1 *pixel* = 2 cm, dan hasil uji kecepatan pengiriman data robot ke aplikasi *base station* juga menyimpulkan bahwa *range* periode pengiriman terbaik adalah dari 33 milidetik sampai 200 milidetik. Kesimpulannya aplikasi yang dibuat dapat digunakan untuk robot sepak bola beroda.

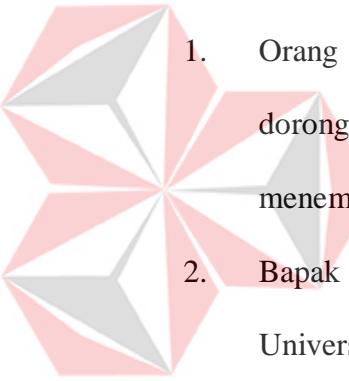
**Kata kunci :** *basestation, visualisasi, robot, KRI, KRSBI.*

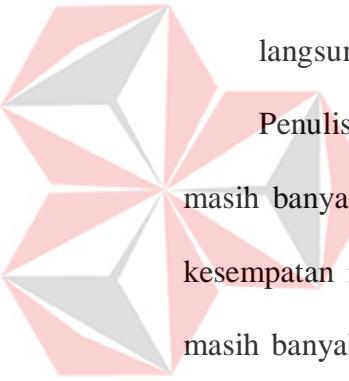


## KATA PENGANTAR

Segala puji syukur kehadirat Allah SWT, karena dengan rahmatnya dan hidayah-Nya penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “PENGEMBANGAN KOMUNIKASI DAN VISUALISASI BASE STATION PADA ROBOT SEPAK BOLA BERODA”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Teknik pada program studi S1 Teknik Komputer Universitas Dinamika.

Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

- 
1. Orang Tua dan Saudara-saudara saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir ini.
  2. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan Baik.
  3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan Baik.
  4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembahas atas ijin dan masukkan dalam menyusun Tugas Akhir ini.
  5. Bapak Dr.Susijanto Tri Rasmana, S.Kom., M.T. dan Ibu Ira Puspasari, S.Si., M.T. selaku Dosen Pembimbing yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.

- 
6. Seluruh dosen pengajar Pogram Studi S1 Teknik Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Universitas Dinamika.
  7. Rekan-rekan Komunitas Dinamika Robotik yang memberikan motivasi serta bantuan dalam penyelesaian Tugas Akhir ini.
  8. Teman- teman seperjuangan Teknik Komputer angkatan 2016 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.
  9. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung,

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun Bahasa yang digunakan.

Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 4 Februari 2020

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN SYARAT .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>Error! Bookmark not defined.</b>
<b>MOTTO .....</b>	<b>iv</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>v</b>
<b>HALAMAN PERNYATAAN .....</b>	<b>vi</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah .....	4
1.4    Tujuan.....	4
<b>BAB II LANDASAN TEORI.....</b>	<b>5</b>
2.1    Robot <i>Holonomic</i> .....	5
2.2 <i>External Rotary Encoder</i> .....	5
2.3    Kartesius Pada Visualisasi Robot Sepak Bola.....	6
2.4    Perhitungan Skala Perbandingan Lapangan .....	7
2.5    Visual Studio.....	8
2.6    NodeMCU ESP8266 .....	11
2.7    Arduino Due .....	12



2.8	Referee Box .....	13
2.9	Split Data Atau Pemisah Data .....	14
<b>BAB III METODE PENELITIAN.....</b>		<b>15</b>
3.1	Analisa Kebutuhan .....	15
3.1.1	Analisa Kebutuhan Perangkat Keras.....	15
3.1.2	Analisa Kebutuhan Perangkat Lunak .....	15
3.2	Blok Diagram Alur Komunikasi .....	16
3.3	Desain Aplikasi dan Implementasi.....	17
3.3.1	Algoritma Aplikasi.....	18
3.3.2	Rancangan Lapangan Pada Visualisasi .....	21
3.3.3	Rancang Robot Pada Aplikasi <i>Base Station</i> .....	22
3.3.4	Rancangan Pisah Data Aplikasi <i>Base Station</i> .....	22
3.4	Peracangan Komunikasi Data.....	23
3.4.1	Terima Data Koordinat.....	25
<b>BAB IV PENGUJIAN DAN PEMBAHASAN .....</b>		<b>26</b>
4.1	Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi <i>Base Station</i> .....	26
4.1.1	Prosedur Pengujian Pada Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi <i>Base Station</i> .....	26
4.1.2	Hasil Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi <i>Base Station</i> . .....	27
4.1.3	Kesimpulan Dari Semua Hasil Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi <i>Base Station</i> . .....	29
4.2	Uji Status Robot Membawa Bola Pada Aplikasi <i>Base Station</i> .....	30
4.2.1	Prosedur Pengujian Pada Uji Status Robot Membawa Bola Pada Aplikasi <i>Base Station</i> . .....	30



4.2.3 Hasil Uji Status Robot Membawa Bola Pada Aplikasi <i>Base Station</i> .....	31
4.2.4 Kesimpulan Dari Semua Hasil Uji Status Robot Membawa Bola Pada Aplikasi <i>Base Station</i> .....	33
4.3 Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi <i>Base Station</i> .....	33
4.3.1 Prosedur Pengujian Pada Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi <i>Base Station</i> . ....	33
4.3.2 Hasil Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi <i>Base Station</i> . ....	34
4.3.3 Kesimpulan Dari Semua Hasil Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi <i>Base Station</i> . ....	37
4.4 Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi <i>Base Station</i> .....	37
4.4.1 Prosedur Pengujian Pada Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi <i>Base Station</i> . 37	37
4.4.2 Hasil Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi <i>Base Station</i> . ....	38
4.4.3 Kesimpulan Dari Semua Hasil Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi <i>Base Station</i> . 45	45
4.5 Uji Kecepatan Pengiriman Data Robot Ke Aplikasi <i>Base Station</i> .....	45
4.5.1 Prosedur Pengujian Pada Uji Kecepatan Pengiriman Data Robot Ke Aplikasi <i>Base Station</i> . ....	45
4.5.2 Hasil Uji Kecepatan Pengiriman Data Robot Ke Aplikasi <i>Base Station</i> .....	46
4.5.3 Kesimpulan Dari Semua Hasil Uji Kecepatan Pengiriman Data Robot Ke Aplikasi <i>Base Station</i> . ....	47
<b>BAB V PENETUP.....</b>	<b>48</b>
5.1 Kesimpulan .....	48

5.2 Saran.....	49
<b>DAFTAR PUSTAKA.....</b>	<b>50</b>
<b>LAMPIRAN .....</b>	<b>52</b>
<b>BIODATA PENULIS.....</b>	<b>103</b>



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Robot <i>Holonomic</i> .....	5
Gambar 2.2 <i>External Rotary Encoder</i> .....	6
Gambar 2.3 Kartesius Visualisasi Dan Robot Sepak Bola Beroda .....	7
Gambar 2.4 Lapangan .....	8
Gambar 2.5 Tampilan <i>form GUI</i> .....	9
Gambar 2.6 Tampilan <i>form editor</i> .....	10
Gambar 2.7 NodeMCU ESP8266.....	11
Gambar 2.8 <i>Mapping Pin</i> NodeMCU.....	12
Gambar 2.9 Arduino Due.....	12
Gambar 2.10 Tampilan <i>Referee Box</i> .....	13
Gambar 2.11 Pemisah Data X, Y, $\Theta$ .....	14
Gambar 2.12 Format Pemisah Data.....	14
Gambar 3.1 Blok Diagram Alur Komunikasi .....	16
Gambar 3.2 Aplikasi Base Station.....	18
Gambar 3.3 (a) Flowchart Alur Komunikasi, (b) Flowchart Penerimaan Data Dari <i>Referee Box</i> .....	19
Gambar 3.4 Flowchart Kontrol Simulator .....	20
Gambar 3.5 Flowchart Kirim Data Dari robot .....	21
Gambar 3.6 (a) Robot Sebenarnya, (b) Gambar Robot Pada Aplikasi .....	22
Gambar 3.7 Proses Pisah Data.....	23
Gambar 1 Tampilan Grup <i>RefeereBox</i> Pada Aplikasi <i>Basestation</i> .....	72
Gambar 2 Tampilan Grup Robot Pemain Pada Aplikasi <i>Basestation</i> .....	73
Gambar 3 Tampilan Grup Koordinat Manual Pada Aplikasi <i>Basestation</i> .....	73
Gambar 4 Tampilan Gambar Robot Pertama Didalam Lapangan.....	74
Gambar 5 Tampilan Grup Robot Pertama Pada Aplikasi <i>Basestation</i> .....	75
Gambar 6 Tampilan Grup Status Bola pada Aplikasi <i>Basestation</i> .....	76
Gambar 7 Tampilan Grup Terima Karakter Pada Aplikasi <i>Basestation</i> .....	77

## DAFTAR TABEL

	Halaman
Tabel 2.1 Ukuran Lapangan Dalam Satuan Meter .....	8
Tabel 4.1 Uji Validasi Dan Komunikasi Menu Utama Pada Aplikasi <i>Refeerebox</i> . .....	27
Tabel 4.2 Uji Validasi Dan Komunikasi Menu Tim <i>Cyan</i> Pada Aplikasi <i>Refeerebox</i> .....	28
Tabel 4.3 Uji Validasi Dan Komunikasi Menu Tim <i>Magenta</i> Pada Aplikasi <i>Refeerebox</i> .....	28
Tabel 4.4 Uji Status Robot Membawa Bola Pada Aplikasi <i>Base Station</i> . ....	31
Tabel 4.5 Uji Ketepatan Gerak Robot Maju Pada Aplikasi <i>Base Station</i> .....	34
Tabel 4.6 Uji Ketepatan Gerak Robot Samping Kanan Pada Aplikasi <i>Base Station</i> . .....	35
Tabel 4.7 Uji Ketepatan Gerak Robot Serong Pada Aplikasi <i>Base Station</i> . ....	36
Tabel 4.8 Uji Coba Kesamaan Gerak Robot Maju. ....	38
Tabel 4.9 Uji Coba Kesamaan Gerak Robot Maju. ....	39
Tabel 4.10 Uji Coba Kesamaan Gerak Robot Maju. ....	41
Tabel 4.11 Uji Coba Kesamaan Gerak Robot Maju. ....	42
Tabel 4.23 Uji Kecepatan Pengiriman Data Robot Ke Aplikasi <i>Base Station</i> .....	46
Tabel 4.12 Uji Coba Kesamaan Gerak Robot Maju. ....	80
Tabel 4.13 Uji Coba Kesamaan Gerak Robot Kesamping Kanan.....	83
Tabel 4.14 Uji Coba Kesamaan Gerak Robot Kesamping Kanan.....	84
Tabel 4.15 Uji Coba Kesamaan Gerak Robot Kesamping Kanan.....	85
Tabel 4.16 Uji Coba Kesamaan Gerak Robot Kesamping Kanan.....	87
Tabel 4.17 Uji Coba Kesamaan Gerak Robot Kesamping Kanan.....	90
Tabel 4.18 Uji Coba Kesamaan Gerak Robot Serong. ....	92
Tabel 4.19 Uji Coba Kesamaan Gerak Robot Serong. ....	94
Tabel 4.20 Uji Coba Kesamaan Gerak Robot Serong. ....	95
Tabel 4.21 Uji Coba Kesamaan Gerak Robot Serong. ....	97
Tabel 4.22 Uji Coba Kesamaan Gerak Robot Serong. ....	100



## **DAFTAR LAMPIRAN**

	Halaman
Lampiran 1 Program Viual Studio.....	52
Lampiran 2 Program Robot.....	66
Lampiran 3 Program ESP.....	70
Lampiran 4 Penjelasan Aplikasi Base Station.....	72
Lampiran 5 <i>Flowchart</i> Terima Data Koordinat Robot .....	78
Lampiran 6 Blok Diagram Perancangan Komunikasi Data .....	79
Lampiran 7 Lapangan .....	79
Lampiran 8 Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi <i>Base Station</i> . ....	80



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kontes Robot Indonesia (KRI) adalah suatu *event* kontes robot bagi perguruan tinggi di indonesia yang dapat di ikuti oleh mahasiswa yang rutin dilaksanakan setiap tahun sekali. KRI diselenggarakan oleh Kementrian Riset, Teknologi dan Pendidikan Tinggi untuk tujuan meningkatkan bakat dan kemampuan mahasiswa dalam bidang robotika. Salah satu divisi dari Kontes Robot Indonesia (KRI) adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI) (Kementerian Riset, 2019).



Pada saat pertandingan, robot akan dikontrol dan diperintah berdasarkan instruksi wasit melalui perangkat lunak yang disebut *Refbox* (*Referee Box*). *Refbox* merupakan *game controller* salah satu teknologi yang digunakan untuk mempermudah wasit dalam mengatur jalannya pertandingan KRSBI. Fungsi dari *Refbox* adalah untuk menyampaikan keputusan wasit kepada robot yang sedang bermain, mencatat skor, dan sebagai penampilan waktu pertandingan. Protokol komunikasi *Refbox* mengirimkan beberapa pesan ke komputer kedua tim (*magenta* dan *cyan*) (Ardhiansyah & Syarifuddin, 2017).

Universitas Dinamika yang dahulu masih memakai nama Institut Bisnis Dan Informatika Stikom Surabaya telah mengikuti tiga kali kontes tersebut yaitu pada tahun 2017, 2018, dan 2019. Pada tahun pertama robot sepak bola tim hanya mampu mendeteksi warna bola dan juga menendang bola. Hal ini masih jauh belum memenuhi syarat kriteria dari *rule*

*competition* dan tim belum mampu membuat aplikasi *Base Station* untuk dapat berkomunikasi dengan baik antar robot maupun menerima perintah dari *Refree Box* (kontrol wasit).

Pada tahun kedua robot sepak bola Universitas Dinamika mengalami kemajuan dalam hal mendekripsi warna bola lebih akurat, menentukan arah serta dapat membuat aplikasi *Base Station* untuk komunikasi antar robot dan menerima perintah dari *Refree Box* ke *Base Station*, adanya aplikasi *Base Station* ini mempermudah tim dalam pengontrolan robot dari laptop pada saat itu hanya dapat maju, mundur, rotasi kanan dan rotasi kiri.



Pada tahun ketiga tim terdapat kemajuan dalam hal menentukan arah lawan lebih akurat, *base* robot yang lebih presisi, penendang menggunakan motor dan dapat mengontrol robot sesuai dengan koordinat dalam skala lapangan yang sebenarnya. Tetapi terdapat beberapa kelemahan dari sisi komunikasi. Hingga saat ini, untuk melakukan perintah *start*, *stop*, dan lain-lain yang seharusnya diberikan oleh *Refree Box* milik wasit masih dilakukan secara manual menggunakan tombol yang terdapat pada *Base Station*, sebagai contoh ketika wasit memberi perintah *start* seketika itu *Base Station* menerima perintah dan melanjutkan perintah ke robot dengan menekan tombol *start* secara manual. Seharusnya tidak boleh terjadi karena melanggar aturan dari Dikti sebagaimana yang telah diatur di rule tersebut seharusnya *Base Station* dapat meneruskan perintah dari *Refree Box* milik wasit ke robot secara otomatis tanpa harus menekan tombol secara manual. Kelemahan yang

lain adalah belum mampu mengetahui pergerakan robot secara lebih detail di aplikasi *Base Station*.

Untuk mengatasi beberapa permasalahan di atas maka penulis ingin mengembangkan aplikasi *Base Station* yang sudah ada. Tujuan dari penelitian yaitu dapat mencatat dan menampilkan pergerakan robot secara lebih detail. Pengembangan ini dapat digunakan untuk meningkatkan strategi kedepannya. Pengembangan selanjutnya yaitu dapat meneruskan perintah dari *Refree Box* ke *Base Station* lalu di *broadcast* ke robot secara otomatis yang menjadi *rule* yang sebenarnya. Visualisasi ini menerima data paket pulsa yang dikirim secara langsung dari robot berupa data pulsa putaran (rotary) kedua roda untuk kemudian diolah menjadi data koordinat yang digunakan untuk memvisualisasikan pergerakan robot pada bidang kartesian (Wahyuni, 2015).

## 1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, rumusan masalah dalam pelaksanaan tugas akhir ini adalah :

1. Bagaimana cara membuat aplikasi *Base Station* yang dapat menerima perintah dari *Refree Box* ke *Base Station* lalu di *broadcast* ke robot secara otomatis dengan komunikasi *wireless*?
2. Bagaimana cara mencatat pergerakan robot lebih detail dan ditampilkan di simulator?

### 1.3 Batasan Masalah

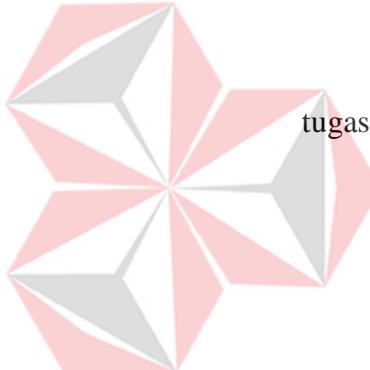
Dalam pembuatan dan perancangan alat ini, terdapat beberapa batasan masalah, antara lain :

1. Tidak membahas jaringan komunikasi yang digunakan.
2. Hanya menggerakan robot sesuai dengan koordinat yang ditentukan, tidak membahas traking bola, dan menendang bola.
3. Kondisi lapangan sesuai aturan.
4. Robot berjalan tanpa memperdulikan halangan yang ada.
5. Hanya menggunakan satu robot.

### 1.4 Tujuan

Dari latar belakang yang telah diuraikan, tujuan dalam penggeraan tugas akhir ini adalah :

1. Dapat membuat aplikasi *Base Station* yang dapat meneruskan perintah dari wasit ke *Base Station* lalu ke robot secara otomatis dengan komunikasi *wireless*.
2. Dapat mencatat pergerakan robot lebih detail dan ditampilkan disimulator.



## BAB II

### LANDASAN TEORI

#### 2.1 Robot *Holonomic*

Robot *holonomic* adalah robot yang bergerak ke segala arah tanpa melakukan rotasi sehingga robot memiliki ruang gerak yang bebas dalam melakukan pergerakan robot. Dimana konsep pergerakan ini mempunyai kelebihan dibandingkan dengan konsep pergerakan robot lainnya.

Roda yang diigunakan adalah roda *omnidirectional*. Roda khusus ini mempunyai roda kecil tambahan yang berporos tegak lurus pada roda inti, sehingga dapat bergerak ke segala arah. Desain robot *holonomic* dapat dilihat pada gambar 2.1.

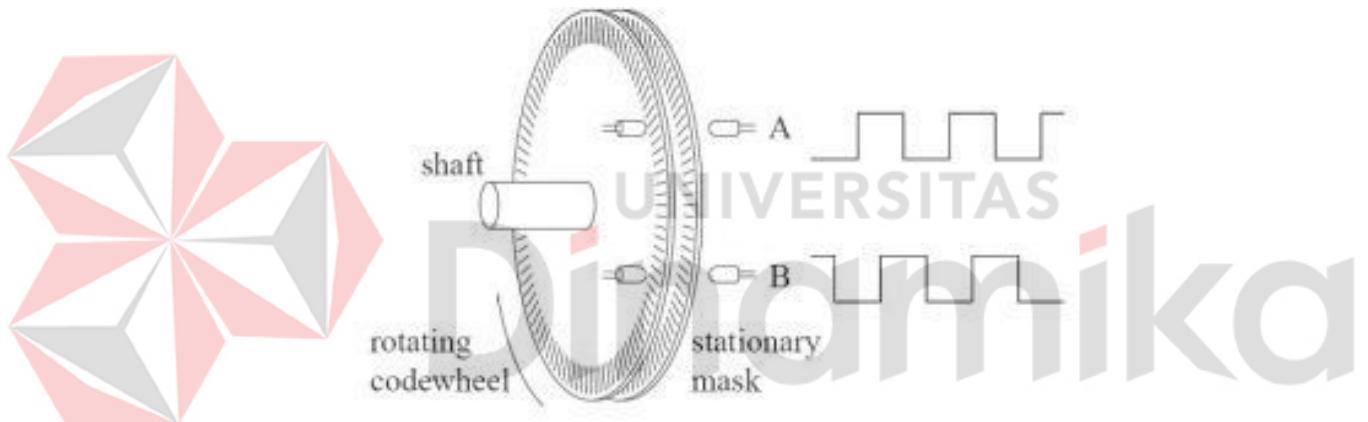


Gambar 2.1 Robot *Holonomic* (*Thingbits*, 2019)  
(Sumber : <https://www.thingbits.net/products/3wd-48mm-omni-directional-triangle-mobile-robot-chassis>)

#### 2.2 External Rotary Encoder

*Rotary Encoder* (ES403-100-3-T-24) merupakan sensor yang digunakan untuk menghitung putaran poros, sehingga digunakan untuk mendeteksi

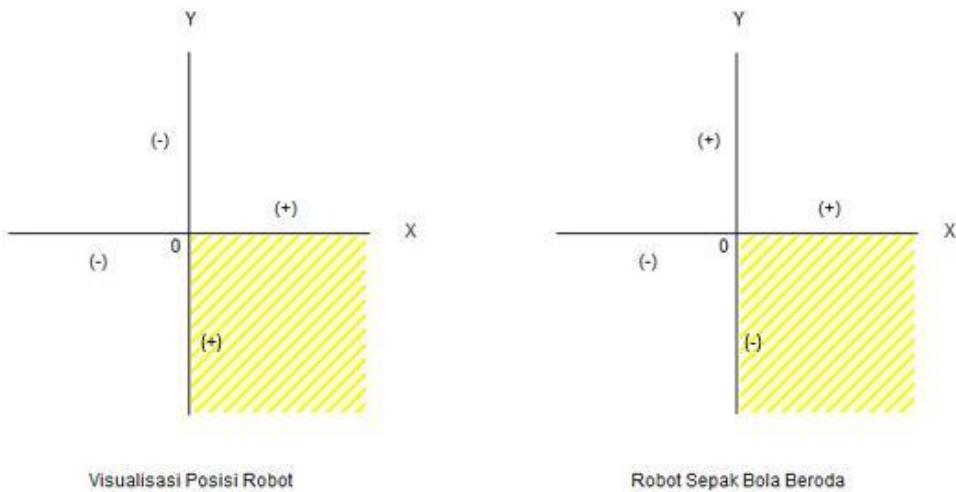
pergerakan posisi. Terdiri dari *single track* dan *double track* dan dua channel A dan B (gambar 2.2). Ketika poros berputar, deretan pulsa akan muncul di masing-masing *channel* pada *frekuensi* yang proporsional dengan kecepatan putar sedangkan hubungan fasa antara channel A dan B menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur. 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 fasa seperempat putaran (Karnadi, 2019).



Gambar 2.2 *External Rotary Encoder*  
 (Sumber : [http://hades.mech.northwestern.edu/index.php/Rotary\\_Encoder](http://hades.mech.northwestern.edu/index.php/Rotary_Encoder))

### 2.3 Kartesius Pada Visualisasi Robot Sepak Bola

Menurut penelitian dari Viky Indrarta, koordinat objek pada visualisasi dan koordinat robot sepak bola beroda pada lapangan memerlukan konversi terlebih dahulu, karena perhitungan koordinat objek robot sebenarnya dengan koordinat robot pada visualisasi sangatlah berbeda (Indrarta, 2019). Konversi yang dilakukan adalah dengan menghitung dari kedua persamaannya.



*Gambar 2.3 Kartesius Visualisasi Dan Robot Sepak Bola Beroda (Indrarta, 2019)*  
 (Sumber : Vicky Indrarta )

Nilai dari visualisasi sebelum dikirimkan ke robot maka perlu dilakukan konversi dengan menggunakan persamaan (2.1). Sedangkan untuk persamaan (2.2) adalah konversi nilai dari robot ke visualisasi (Indrarta, 2019).

$$\begin{aligned} x &= x_{\text{akhir}} \text{ (cm)} - x_{\text{awal}} \text{ (cm)} \\ y &= - (y_{\text{akhir}} \text{ (cm)} - y_{\text{awal}} \text{ (cm)}) \end{aligned} \quad (2.1)$$

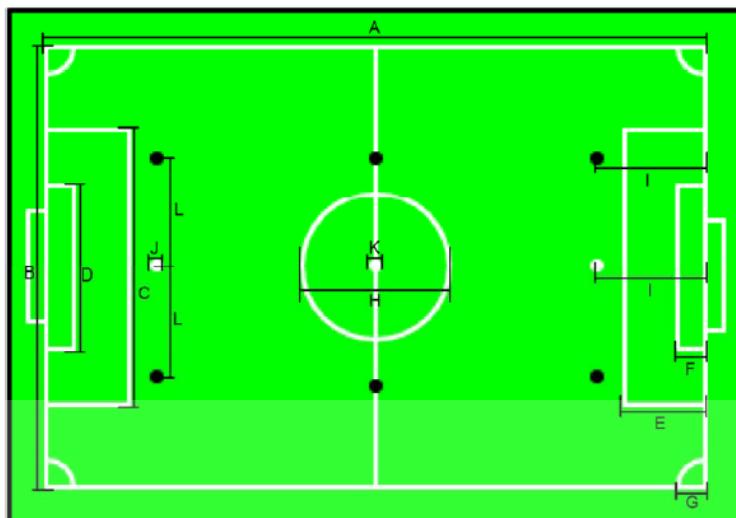
$$\begin{aligned} x &= x_{\text{akhir}} \text{ (cm)} + x_{\text{awal}} \text{ (cm)} \\ y &= y_{\text{akhir}} \text{ (cm)} + y_{\text{awal}} \text{ (cm)} \end{aligned} \quad (2.2)$$

## 2.4 Perhitungan Skala Perbandingan Lapangan

Skala pada Gambar 2.4 adalah perbandingan paling sederhana antara jarak pada gambar dengan jarak pada objek sebenarnya yang menggunakan satuan jarak pada gambar. Contohnya jika diketahui jarak antara dua titik pada gambar 9 cm sedangkan jarak sebenarnya adalah 9 m. Jika kedua jarak tersebut menggunakan satuan yang jarak pada gambar yaitu cm, maka jarak pada gambar adalah 9 m dan jarak sebenarnya adalah 900 cm. Perbandingan jarak pada gambar

dengan jarak sebenarnya adalah 9:900, dan perbandingan paling sederhana adalah 1:100.

Agar pembuatan jarak pada gambar dapat sesuai dengan jarak sebenarnya terdapat acuan dari aturan kompetisi yang ada pada tabel 2.1



*Gambar 2.4 Lapangan*  
(Sumber : <http://kontesrobotindonesia.id/tentang-kri.html>)

Tabel 2.1 Ukuran Lapangan Dalam Satuan Meter

A	9	G	0,375
B	6	H	2
C	Lebar gawang + -2,25	I	1,5
D	Lebar gawang + -0,75	J	0,05
E	1,125	K	0,075
F	0,375	L	1,5

(Sumber : <http://kontesrobotindonesia.id/tentang-kri.html>)

## 2.5 Visual Studio

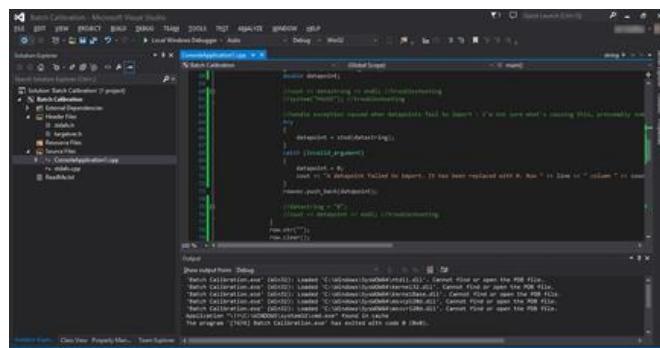
Microsoft visual studio merupakan sebuah IDE (*Intregrated Development Environment*) dari Microsoft untuk pengembangan aplikasi. IDE sendiri merupakan program komputer yang memiliki fasilitas-fasilitas yang diperlukan untuk pembangunan perangkat lunak. Dengan aplikasi Visual Studio

ini, bisa dibangun aplikasi GUI, aplikasi konsole, aplikasi web, maupun aplikasi *mobile*.

Microsoft Visual Studio memiliki beberapa edisi untuk pengembangan aplikasi. Edisi Microsoft Visual Studio diantaranya ada Community, Professional, Enterprise, Test Professional, dan Express.

Pada rancangan ini untuk membuat simulasi menggunakan fasilitas GUI (*Graphical User Interface*) yang ada di Microsoft Visual Studio yaitu Windows Forms Designer. Dan bahasa yang digunakan yaitu bahasa C#. Pada tampilan windows forms ada dua bagian yang digunakan dalam pengerjaan sebuah aplikasi. Pertama, bagian untuk mendesain tampilan GUI. Kedua, bagian untuk mengkoding program buat GUI agar tampil dinamis.

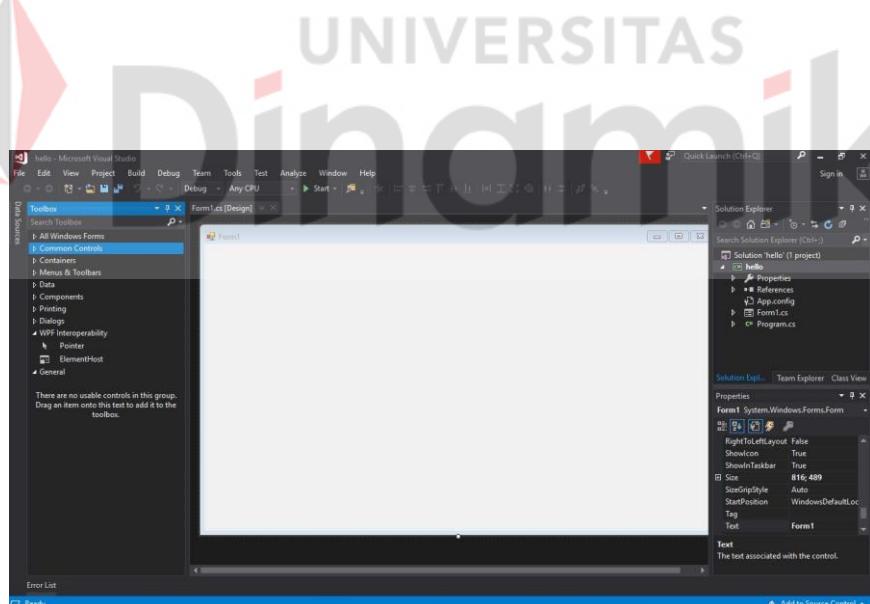
Antar muka atau yang disebut GUI (*Graphical User Interface*) adalah antarmuka yang menggunakan metode interaksi antara pengguna dengan perangkat komputer menggunakan grafis (tidak perintah teks). Pada Visual Studio terutama di windows forms terdapat menu-menu yang bisa digunakan yaitu *menubar*, *toolbar*, *toolbox*, form untuk *designer* dan *editor*, *property*, dan *solution explorer*. Berikut Gambar 2.5 tampilan windows form pada Visual Studio.



**Gambar 2.5 Tampilan form GUI**  
 (Sumber: <https://www.iqbalpambudi.com/2018/05/cara-membuat-aplikasi-kalkulator.html>)

Pembuatan projek di Visual Studio dimulai dengan membuat tampilan designer di form, kemudian diberi *script* atau coding sesuai komponen yang ada di form editor. Pada form *designer* seperti Gambar 2.6 untuk desainnya disediakan fasilitas-fasilitas untuk pengembangannya di bagian *toolbox*. Komponen yang ada di *toolbox* diantaranya *Button*, *Label*, *TextBox*, dan masih banyak lainnya.

Kemudian pengaturan fungsi-fungsi *toolbox* yang diletakkan di form berada di *property* bagian kanan windows. Pada menubar terdapat operasional standar windows seperti membuat projek baru dan menyimpannya debugging program yang telah dibuat dan lainnya. Sedangkan pada *solution explorer* terdapat berbagai referensi, *file*, dan *folder* yang dibutuhkan untuk membangun aplikasi.



*Gambar 2.6 Tampilan form editor  
(Sumber: [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio))*

## 2.6 NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT (*Internet of Things*) yang bersifat opensource. Terdiri dari perangkat keras berupa System On Chip ESP8266 dari ESP8266 buatan Esperessif System.



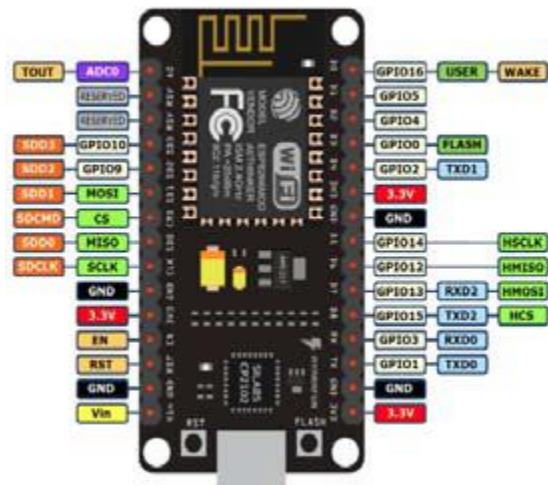
*Gambar 2.7 NodeMCU ESP8266 (Efendi & Chandra, 2019)*  
(Sumber : <https://en.wikipedia.org/wiki/NodeMCU>)

NodeMCU bisa dianalogikan sebagai board arduino yang terkoneksi dengan ESP8266. NodeMCU telah me-package ESP8266 ke dalam sebuah board yang sudah terintegrasi dengan berbagai feature selayaknya microkontroler dan kapalitas akses terhadap wifi dan juga chip komunikasi yang berupa USB to serial. Sehingga dalam pemrograman hanya dibutuhkan kabel data USB.

Karena sumber utama dari NodeMCU adalah ESP8266 khususnya seri ESP-12 yang termasuk ESP-12E. Maka fitur – fitur yang dimiliki oleh NodeMCU akan lebih kurang serupa dengan ESP-12. Beberapa fitur yang tersedia antara lain (Circuit, 2019).

1. 10 Port GPIO dari D0 – D10
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 Wire

## 5. ADC



Gambar 2.8 Mapping Pin NodeMCU (*Efendi & Chandra, 2019*)  
 (Sumber : <https://www.nyebarilmu.com/apa-itu-module-nodemcu-esp8266/>)

## 2.7 Arduino Due

Arduino Due adalah papan mikrokontroler yang berbasis pada *CPU Atmel SAM3X8E ARM Cortex-M3*. Ini adalah papan Arduino pertama yang berbasis pada mikrokontroler *ARM* inti 32-bit. Memiliki 54 pin *input/output* digital (12 dapat digunakan sebagai output PWM), 12 *input analog*, 4 UART (port serial perangkat keras), *clock* 84 MHz, koneksi USB OTG yang mampu, 2 DAC (*digital to analog*), 2 TWI, colokan listrik, *header SPI*, *header JTAG*, tombol reset dan tombol hapus (Due, 2017).



Gambar 2.9 Arduino Due (*Due, 2017*)  
 (Sumber : <https://store.arduino.cc/usa/arduino-due>)

## 2.8 Referee Box

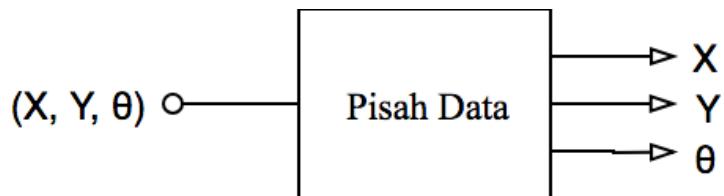
*RefBox* atau *Referee Box* adalah sebuah perangkat lunak untuk mengatur dan mengendalikan pertandingan. *Referee Box* dioperasikan oleh wasit. *Referee Box* mengirimkan instruksi ke *Base Station*, untuk diteruskan ke Robot. Instruksi mencakup memulai pertandingan *START*, menghentikan pertandingan *STOP*, instruksi *Kick Off*, *Free Kick*, *Drop Ball*, dan status pertandingan lainnya. Robot mendengarkan instruksi dari *Referee Box* melalui *Base Station*. Robot beroperasi secara otomatis berdasarkan instruksi yang diterima dari *Referee Box*. Format instruksi yang dikirim dari *Referee Box* ke *Base Station* dalam bentuk satu buah karakter (*char*), sudah sangat rinci berada pada *mailing list* dari panitia KRSBI

Beroda.



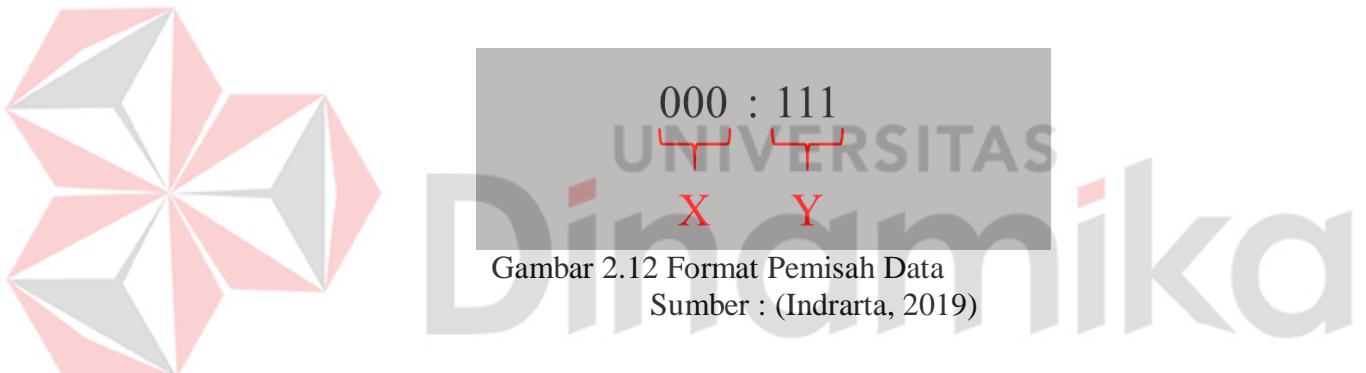
Gambar 2.10 Tampilan *Referee Box*  
(Sumber : <http://kontesrobotindonesia.id/tentang-kri.html>)

## 2.9 Split Data Atau Pemisah Data



Gambar 2.11 Pemisah Data X, Y,  $\theta$   
Sumber : (Abseno, 2019)

Dengan proses ini data yang semula bertipe data string dan belum dikelompokkan menjadi data individu yang akan diproses menjadi beberapa data yang siap dilakukan perhitungan (Abseno, 2019).



Pada Gambar 2.12. menjelaskan bahwa karakter ":" digunakan untuk pemisah data antara koordinat X dan Y. Data X memiliki nilai 000 dan ditampung ke sebuah *variable* baru, dan data Y bernilai 111 dan ditampung ke *variable* baru yang nanti data tersebut akan diproses oleh robot dan sebagai data koordinat yang harus dicapai.

## **BAB III**

### **METODE PENELITIAN**

Berikut akan dijelaskan tentang tahapan rancangan dari penelitian ini yaitu, analisa kebutuhan, blok digram alur komunikasi, desain aplikasi dan implementasi, dan perancangan komunikasi data.

#### **3.1 Analisa Kebutuhan**

Pada tahap ini akan dilakukan analisa terhadap kebutuhan – kebutuhan sistem dan perangkat keras terhadap sebuah pembuatan aplikasi *basestation*. Sistem yang di analisa adalah sistem yang berisi informasi tentang segala sesuatu berkaitan dengan pembuatan aplikasi *base station*.

##### **3.1.1 Analisa Kebutuhan Perangkat Keras**

Spesifikasi komponen perangkat keras yang digunakan untuk pembuatan aplikasi basestation ini adalah sebagai berikut:

1. PC/Laptop dengan prosesor core 2 duo.
2. Router.
3. NodeMCU ESP8266.
4. Robot.
5. Switch Hup.

##### **3.1.2 Analisa Kebutuhan Perangkat Lunak**

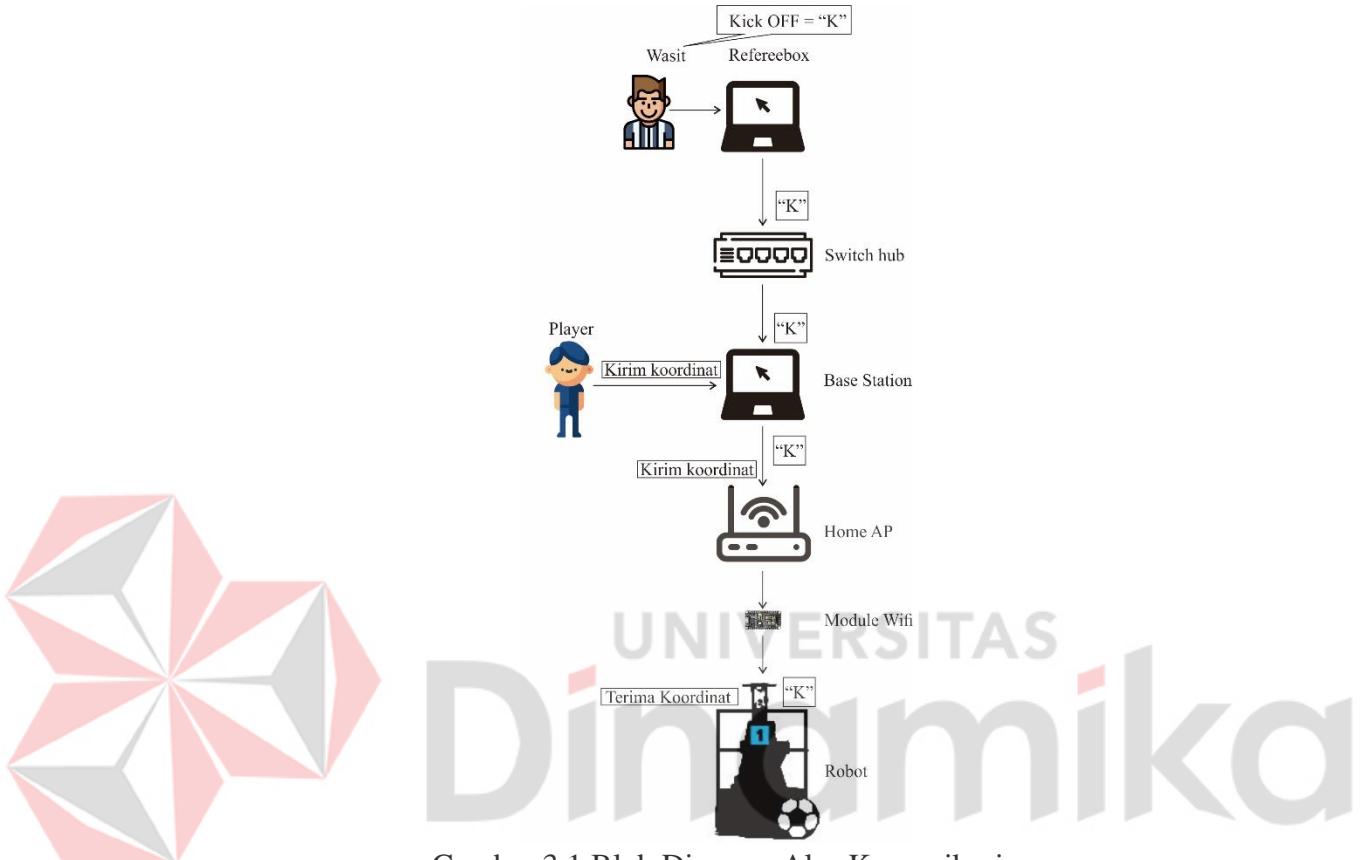
Perangkat lunak yang dibutuhkan pada perancangan aplikasi adalah sebagai berikut:

1. Sistem operasi Windows 7/8/10.

2. Visual Studio 2017.

3. Aplikasi *RefereeBox*.

### 3.2 Blok Diagram Alur Komunikasi



Gambar 3.1 Blok Diagram Alur Komunikasi  
(Sumber: Olahan Penulis)

Pada penelitian Tugas Akhir ini, peneliti membuat sebuah aplikasi *Base Station* yang berfungsi untuk mengontrol pergerakan robot lebih *smooth* dan memvisualisasikan pada tampilan lapangan, memonitoring status dimana robot membawa bola atau tidak dan meneruskan perintah dari *Referee Box* secara otomatis dan robot memberikan feedback nilai yang meliputi nilai koordinat, status, karakter dari *Referee Box*. Agar dapat berkomunikasi antara aplikasi dan robot diperlukan koneksi jaringan yang akan langsung terkoneksi ke robot sepak bola beroda.

Berdasarkan Gambar 3.1 di atas dapat dijelaskan bahwa komputer wasit menjalankan *Referee Box* mengirim perintah Kick Off, yang mewakilkan satu karakter huruf ‘K’. Karakter tersebut akan diterima oleh *Base Station*, kemudian karakter ‘K’ di *broadcast* oleh *Base Station* ke robot secara otomatis tanpa perlu penekanan tombol. Char ‘K’ kemudian diterjemahkan menjadi perintah gerak oleh robot sesuai dengan program yang sudah ada.

Disini *player* (Tim) dapat mengirim data berupa koordinat melalui aplikasi *Base Station* yang telah dikonversi agar data dapat diterima oleh robot dengan baik dan robot akan bergerak sesuai dengan koordinat yang telah diinputkan oleh *player*. *Base station* menerima nilai balik dari pergerakan robot dilapangan yang sebenarnya untuk visualisasi robot di aplikasi *Base Station*.

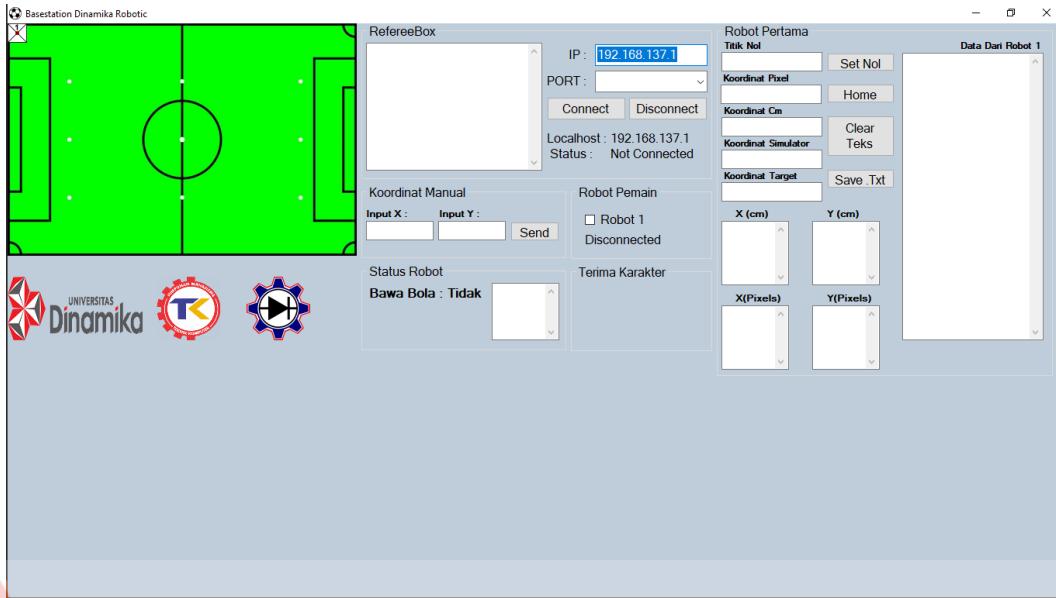


Satu robot yang digunakan dalam Tugas Akhir ini, memerlukan *device* NodeMCU ESP8266 sebagai jalur komunikasi antara robot dan aplikasi *Basestation*. Posisi awal robot pada sudut horizontal koordinat X adalah 0, dan sudut vertikal Y adalah 0. Berikut penjelasan tentang proses pengiriman data dari aplikasi *Base Station* menuju ke robot sampai mengirimkan nilai umpan balik untuk pergerakan robot.

### 3.3 Desain Aplikasi dan Implementasi.

*Base Station* adalah komputer, *notebook*, atau laptop yang menjalankan *software* yang digunakan untuk memantau dan mengatur kerja robot secara otomatis berdasarkan instruksi yang diterima dari *Referee Box* (Ardhiansyah & Syarifuddin, 2017). Pada Gambar 3.2 tersebut adalah *graphical user interface* (GUI). Terdapat gambar lapangan, gambar robot, grupbox *referee box*, grup koordinat manual, grup robot pemain, grup status bola, grup terima karakter,

grup robot pertama, dimana masing-masing dari grup didalamnya terdapat *toolbox – toolbox*.

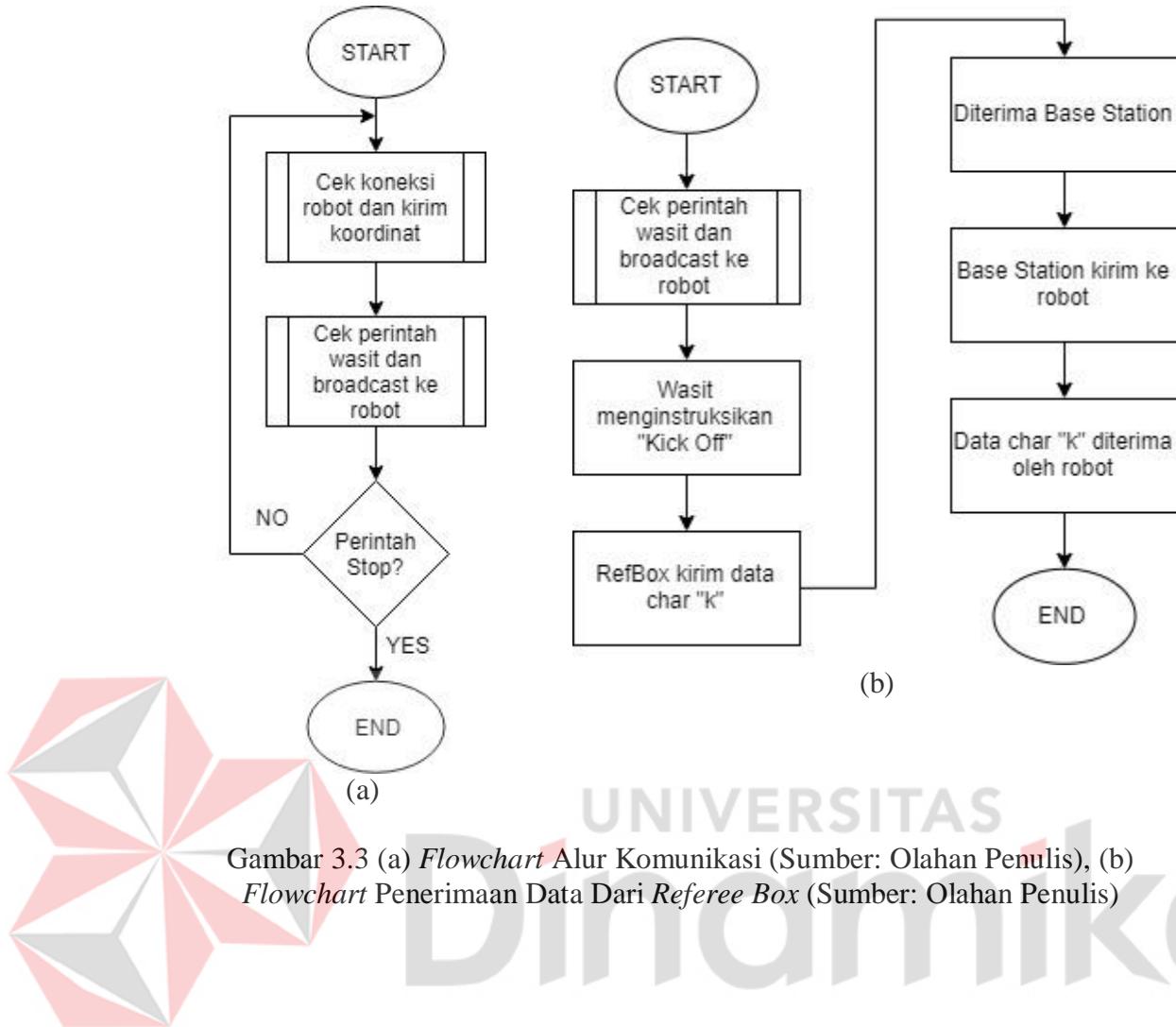


Gambar 3.2 Aplikasi *Base Station*  
(Sumber: Olahan Penulis)

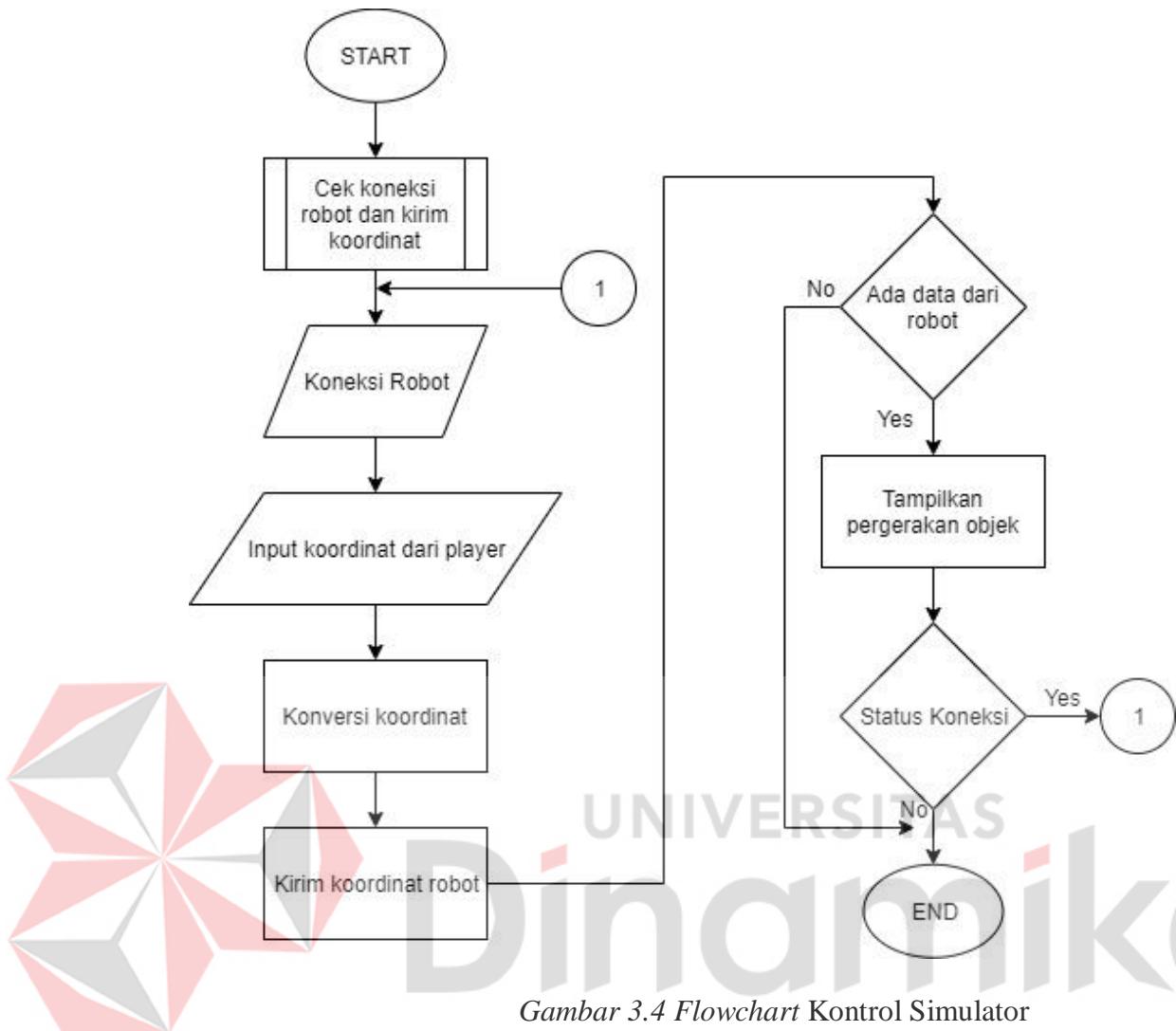
### 3.3.1 Algoritma Aplikasi

Untuk memudahkan pemahaman menggunakan aplikasi *Base Station* ini alangkah baiknya memahami *flowchart* dari perancangan aplikasi *Base Station* ini. Berikut beberapa *flowchart* yang digunakan untuk merancang sebuah aplikasi *Base Station*.

Berikut penjelasan *flowchart* Gambar 3.3 (a) dan Gambar 3.3 (b) Setelah aplikasi terkoneksi dengan komputer wasit, maka wasit dapat mengirimkan sebuah instruksi kepada tim yang bertanding melalui masing - masing aplikasi *Base Station* dari kedua tim, dari *flowchart* di atas wasit megirimkan perintah Kick Off yang diwakili dengan karakter ‘K’ melalui *Referee Box*, lalu diterima oleh *Base Station* kemudian dibroadcast ke robot dan diterjamahkan oleh robot dan dikirim balik ke aplikasi *Base Station*.

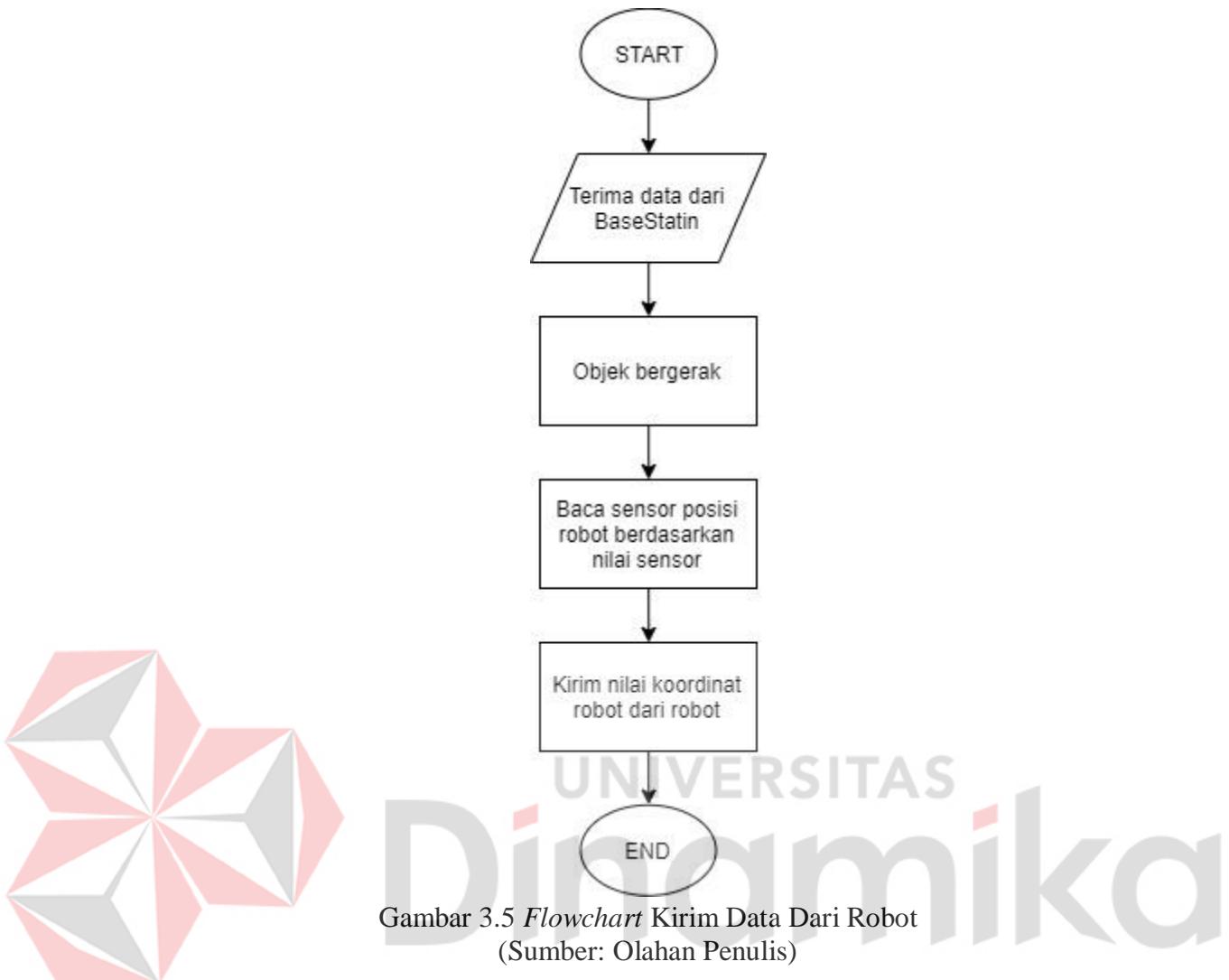


Gambar 3.3 (a) *Flowchart* Alur Komunikasi (Sumber: Olahan Penulis), (b) *Flowchart* Penerimaan Data Dari *Referee Box* (Sumber: Olahan Penulis)



Gambar 3.4 Flowchart Kontrol Simulator  
(Sumber: Olahan Penulis)

Gambar *flowchart* 3.4 dan 3.5 adalah menjelaskan mengenai proses pada yang berjalan sekuensial. Langkah awal untuk mengoperasikan aplikasi ini adalah user harus koneksi pada Robot 1. Setelah terkoneksi antara aplikasi *Base Station* dan robot lalu user dapat menginputkan data koordinat X dan Y dan mengirimkan data tersebut ke robot. Robot akan menerjemahkan inputan tersebut menjadi pergerakan robot sesuai dengan *input user* dan mengirimkan nilai koordinat robot kepada aplikasi *Base Station*. Aplikasi *Base Station* akan memproses data yang diterima dari robot dan menerjemahkan data tersebut menjadi pergerakan simulator robot sesuai dengan gerak robot yang sebenarnya.



### 3.3.2 Rancangan Lapangan Pada Visualisasi

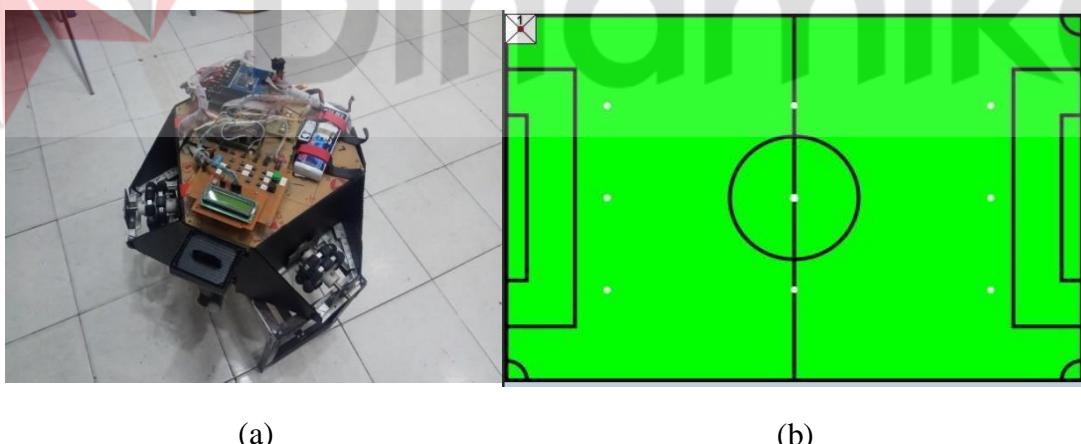
Adapun rancangan pada sebuah lapangan yang memiliki ukuran yang sebenarnya adalah panjang 9 meter dan lebar 6 meter. Pada perancangan lapangan pada aplikasi *Base Station* menggunakan ketentuan  $2 \text{ cm} = 1 \text{ pixel}$  yaitu dengan merubah panjang 900 cm lalu dikali dengan 0,5 dan menghasilkan nilai 450 cm, dan sisi lebar 600 cm juga dikali dengan 0,5 dan menghasilkan nilai 300 cm. Karena untuk memperkecil ukuran lapangan pada aplikasi *Base Station* agar untuk menghemat *space* (ruang) untuk keperluan lainnya. Dan juga untuk mendesain dalam sebuah aplikasi visual studio menggunakan satuan *pixel* maka dari cm perlu

dikonversikan menjadi *pixel* dengan pedoman  $2 \text{ cm} = 1 \text{ pixel}$ , pedoman tersebut berasal dari  $900 \text{ cm}$  dibagi dengan  $450 \text{ pixel}$  dan menghasilkan nilai 2, maka itu yang penulis menjadikan acuan dari penelitian ini. Dapat dilihat lapangan sebenarnya pada lampiran 7a dan perancangan lapangan pada aplikasi dapat dilihat pada lampiran 7b.

### **3.3.3 Rancang Robot Pada Aplikasi *Base Station***

Robot yang digunakan pada Tugas Akhir ini memiliki dimensi panjang 50 *cm* dan lebar juga 50 *cm*, maka perlu penyesuaian seperti rancangan lapangan perlu dikali dengan nilai 0,5 sehingga mendapat panjang 25 *cm* dan lebar 25 *cm*.

Maka langsung diimplementasikan ke aplikasi *Base Station* dengan panjang dan lebar 25 *pixel*. Untuk lebih jelasnya dapat ukuran robot sebenarnya pada dilihat pada Gambar 3.6 (a) dan untuk gambar robot pada aplikasi bisa dilihat pada Gambar 3.6 (b).



Gambar 3.6 (a) Robot Sebenarnya (Sumber: Olahan Penulis), (b) Gambar Robot Pada Aplikasi (Sumber: Olahan Penulis)

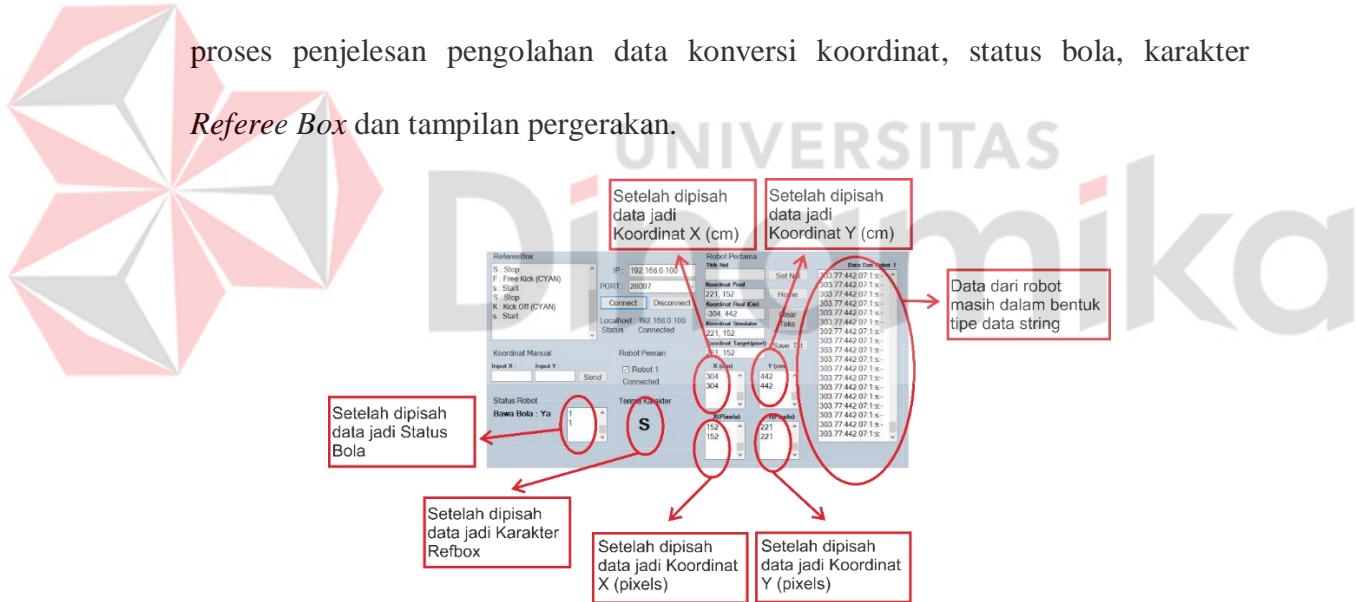
### **3.3.4 Rancangan Pisah Data Aplikasi *Base Station***

Data dari robot yang diterima dan ditampung oleh aplikasi *Base Station* masih dalam bentuk tipe data *string*, maka perlu proses pemisahan data dana

konversi tipe data, agar lebih mudah dalam mengolah data. Data yang dikirim dari robot dan diterima oleh aplikasi *Base Station* adalah dalam bentuk koordinat X : koordinat Y : StatusBola : Karakter *Referee Box*. Data yang ditampung akan dipisah atau displit oleh karakter ":" yang nantinya disimpan dalam bentuk array agar lebih mudah untuk mengolahh data yang nantinya akan membedakan nilai koordinat X, koordinat Y, Status Bola, Karakter *Referee Box*.

Data yang diterima aplikasi *Base Station* dari robot bisa dilihat pada Gambar 3.7.

Dapat dilihat dari proses pertama ke proses kedua dan ketiga terdapat pembulatan data inputan yakni jika kurang dari 0,5 maka pembulatan kebawah, jika sebaliknya lebih dari sama dengan 0,5 maka pembulatan keatas. Berikut proses penyelesaian pengolahan data konversi koordinat, status bola, karakter *Referee Box* dan tampilan pergerakan.



Gambar 3.7 Proses Pisah Data  
(Sumber: Olahan Penulis)

### **3.4 Peracangan Komunikasi Data**

Pada perancangan robot sepak bola beroda ini menggunakan mikrokontroler arduino untuk sebagai proses pergerakan robot atau kontrol robot, sensor proximity untuk sebagai sensor bahwa robot dalam kondisi membawa bola atau tidak dan NodeMCU ESP8266 sebagai pengirim data dari

mikrokontroler ke PC ataupun sebaliknya. Untuk lebih jelasnya dapat dilihat pada lampiran 6.

Pada lampiran 6 terdapat dua catu daya yang digunakan yaitu 5 *volt* untuk NodeMCU ESP8266, motor driver, external rotary. Sedangkan 12-24 *volt* digunakan untuk daya aktuator yaitu motor.

Motor driver yang penulis gunakan adalah tipe EMS-30A dengan kapasitas arus sebesar 30 *ampere* yang fungsinya untuk mengendalikan motor DC dengan sumber tegangan 12V yang berasal dari baterai lipo tiga sel. Membutuhkan tiga inputan untuk menggerakan sebuah motor yaitu direksi 1, direksi 2, dan PWM. Direksi 1 dan 2 digunakan untuk menentukan arah putaran motor, jika direksi 1 kondisi *high* dan direksi 2 kondisi *low* maka motor akan bergerak sesuai dengan arah jarum jam begitu juga sebaliknya. Untuk PWM hanya perlu memberi inputan berupa pulsa dengan perbandingan *duty cycle* sesuai dengan kebutuhan guna mengatur kecepatan putar motor DC.

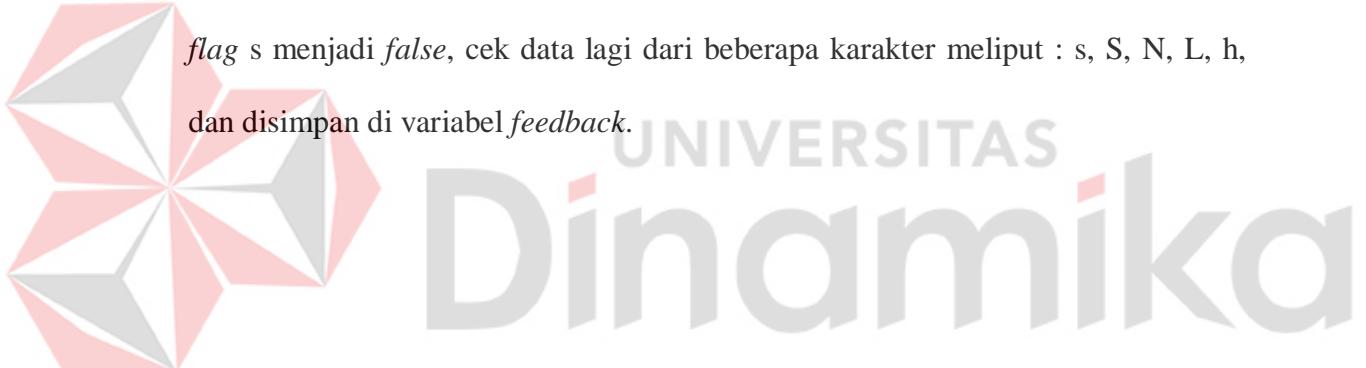
*Rotary encoder* menggunakan *rotary encoder internal* yang ada pada motor DC. Untuk membaca arah putaran motor maka digunakanlah kedua *output* dari rotary yaitu *output A* dan *output B*, karena kedua *output* mempunyai *output pulse* yang berdampingan, salah satu *output* dari rotary dihubungkan dengan pin *interrupt* untuk dapat membaca putaran.

Sensor proxymity yang digunakan penulis adalah tipe E18-D80NK, sensor ini membutuhkan catu daya sebesar 5V. Sensor ini berfungsi sebagai *switch* otomatis, jika terdapat bola maka bernilai 0 (*low*) dan jika tidak terdapat bola maka bernilai 1 (*high*) yang artinya aktif *low*.

### 3.4.1 Terima Data Koordinat

Setelah aplikasi mengirim data berupa koordinat maupun perintah dari wasit pada robot, dari sisi robot dengan nerima data dari NodeMCU ESP8266 ke mikrokontroler melalui komunikasi serial. Berikut *flowchart* dari penerimaan data koordinat pada lampiran 5.

Penjelasan dari *flowchart* lampiran 5, cek data masuk pada serial 3 jika ada data masuk disimpan pada variabel terima, cek kepala dan ekor data yang diwakili oleh karakter ‘<’ dan ‘>’ dan disimpan pada variabel data, cek ulang ekor data diwakili oleh karakter ‘>’ sebagai akhir dari sebuah data dan *set flag* s menjadi *true*, cek kondisi s jika *true* akan memanggil fungsi pisah data dan *set flag* s menjadi *false*, cek data lagi dari beberapa karakter meliput : s, S, N, L, h, dan disimpan di variabel *feedback*.



## **BAB IV**

### **PENGUJIAN DAN PEMBAHASAN**

Dalam bab ini penulis akan membahas tentang sistem yang telah dirancang pada aplikasi *base station* dan beberapa hasil pengujian. Tujuan dari bab ini untuk mengetahui tingkat keberhasilan sistem yang telah dirancang oleh penulis. Terdapat lima pengujian dalam bab ini yaitu, uji validasi dan komunikasi antara robot dan aplikasi *base station*, uji ketepatan koordinat robot sepak bola beroda dengan aplikasi *base station*, uji kesamaan pergerakan robot sebenarnya dengan pergerakan robot di aplikasi *base station*, uji status robot membawa bola pada aplikasi *base station*, uji kecepatan pengiriman data robot ke aplikasi *base station*.



#### **4.1 Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi Base Station.**

Pengujian ini berfungsi untuk mengetahui data dari perintah dari aplikasi *RefereeBox* yang terima oleh aplikasi *base station* dan dikirimkan ke robot lalu dikembalikan lagi di aplikasi *base station* telah sesuai atau tidak sesuai.

##### **4.1.1 Prosedur Pengujian Pada Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi Base Station.**

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut :

1. Mempersiapkan aplikasi *base station* pada komputer untuk kontrol robot sepak bola beroda.
2. Mempersiapkan robot sepak bola beroda pada posisi awal.
3. Mengkoneksikan antara aplikasi *base station* dengan robot.

4. Mengkoneksikan antara aplikasi *base station* dengan aplikasi *RefereeBox*.
5. Memberi perintah kepada robot melalui aplikasi *RefereeBox* dan mengirimkan kepada robot.
6. Mengamati data yang dikirimkan oleh robot kepada aplikasi *base station*, apakah tidak sesuai atau sesuai.

#### **4.1.2 Hasil Uji Validasi Dan Komunikasi Antara Robot Dan Aplikasi *Base Station*.**

Pada Tabel 4.1 adalah hasil uji validasi dan komunikasi menu utama pada aplikasi *refereebox*.

Tabel 4.1 Uji Validasi Dan Komunikasi Menu Utama Pada Aplikasi *Refereebox*.

Uji Coba Ke	<i>Referee Box to Base Station</i>	<i>Base Station to Robot</i>	<i>Robot to Base Station</i>	Base Station	Keterangan
1	START ('s')	"s"	"s"	"s"	Berhasil
2	STOP ('S')	"S"	"S"	"S"	Berhasil
3	DROPBALL ('N')	"N"	"N"	"N"	Berhasil
4	PARK ('L')	"L"	"L"	"L"	Berhasil
5	RESET ('Z')	"Z"	"Z"	"Z"	Berhasil
6	ENDPART ('h')	"h"	"h"	"h"	Berhasil

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.1 ditunjukkan bahwa data yang dikirimkan oleh *Referee Box* ke *Base Station* dan dikirimkan ke robot dan robot mengirimkan data kembali ke *Base Station* guna untuk mencocokan data. Sebagai contoh perintah *START* yang diwakili karakter 's' dan diterima oleh *Base Station* juga 's' lalu dikirimkan ke robot dan robot mengirimkan kembali kepada *Base Station*, dan hasil akhir dari data yang diterima dari robot ke *Base Station* adalah 's'. Dapat simpulkan bahwa data yang dikirimkan oleh *Referee Box* dan data yang diterima

oleh *Base Station* dari robot sesuai ditandai dengan keterangan berhasil. Pada uji coba di atas adalah tingkat keberhasilannya 100%.

Tabel 4.2 Uji Validasi Dan Komunikasi Menu Tim *Cyan* Pada Aplikasi *Refereerebox*.

Uji Coba Ke	<i>Referee Box to Base Station</i>	<i>Base Station to Robot</i>	<i>Robot to Base Station</i>	Base Station	Keterangan
1	Kick Off ("K")	"K"	"K"	"K"	Berhasil
2	Free Kick ("F")	"F"	"F"	"F"	Berhasil
3	Goal Kick ("G")	"G"	"G"	"G"	Berhasil
4	Throw In ("T")	"T"	"T"	"T"	Berhasil
5	Corner ("C")	"C"	"C"	"C"	Berhasil
6	Repair ("O")	"O"	"O"	"O"	Berhasil
7	Penalty ("P")	"P"	"P"	"P"	Berhasil
8	Goal Cyan ("A")	"A"	"A"	"A"	Berhasil
9	Red ("R")	"R"	"R"	"R"	Berhasil
10	Yellow ("Y")	"Y"	"Y"	"Y"	Berhasil

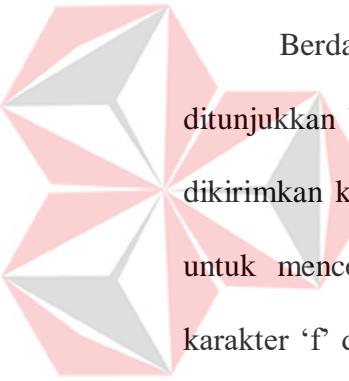
(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.2 diumpama kan sebagai tim *cyan* yang ditunjukkan bahwa data yang dikirimkan oleh *Referee Box* ke *Base Station* dan dikirimkan ke robot dan robot mengirimkan data kembali ke *Base Station* guna untuk mencocokan data. Sebagai contoh perintah *Free Kick* yang diwakili karakter 'F' dan diterima oleh *Base Station* juga 'F' lalu dikirimkan ke robot dan robot mengirimkan kembali kepada *Base Station*, dan hasil akhir dari data yang diterima dari robot ke *Base Station* adalah 'F'. Dapat simpulkan bahwa data yang dikirimkan oleh *Referee Box* dan data yang diterima oleh *Base Station* dari robot sesuai ditandai dengan keterangan berhasil. Pada uji coba di atas adalah tingkat keberhasilannya 100%.

Tabel 4.3 Uji Validasi Dan Komunikasi Menu Tim *Magenta* Pada Aplikasi *Refereerebox*.

Uji Coba Ke	<i>Referee Box to Base Station</i>	<i>Base Station to Robot</i>	<i>Robot to Base Station</i>	Base Station	Keterangan
1	Kick Off ("k")	"k"	"k"	"k"	Berhasil
2	Free Kick ("f")	"f"	"f"	"f"	Berhasil
3	Goal Kick ("g")	"g"	"g"	"g"	Berhasil
4	Throw In ("t")	"t"	"t"	"t"	Berhasil
5	Corner ("c")	"c"	"c"	"c"	Berhasil
6	Repair ("o")	"o"	"o"	"o"	Berhasil
7	Penalty ("p")	"p"	"p"	"p"	Berhasil
8	Goal Cyan ("a")	"a"	"a"	"a"	Berhasil
9	Red ("r")	"r"	"r"	"r"	Berhasil
10	Yellow ("y")	"y"	"y"	"y"	Berhasil

(Sumber: Olahan Penulis)



Berdasarkan Tabel 4.3 diumpama kan sebagai tim magenta yang ditunjukkan bahwa data yang dikirimkan oleh *Referee Box* ke *Base Station* dan dikirimkan ke robot dan robot mengirimkan data kembali ke *Base Station* guna untuk mencocokan data. Sebagai contoh perintah *Free Kick* yang diwakili karakter 'f' dan diterima oleh *Base Station* juga 'f' lalu dikirimkan ke robot dan robot mengirimkan kembali kepada *Base Station*, dan hasil akhir dari data yang diterima dari robot ke *Base Station* adalah 'f'. Dapat simpulkan bahwa data yang dikirimkan oleh *Referee Box* dan data yang diterima oleh *Base Station* dari robot sesuai ditandai dengan keterangan berhasil. Pada uji coba di atas adalah tingkat keberhasilannya 100%.

#### 4.1.3 Kesimpulan Dari Semua Hasil Uji Validasi Dan Komunikasi Antara

##### Robot Dan Aplikasi *Base Station*.

Berdasarkan dari semua hasil uji coba yang dilakukan pada hasil uji validasi dan komunikasi antara robot dan aplikasi *base station*. Terbukti bahwa

data yang dikirimkan dari aplikasi refeebox dan diterima dan diteruskan oleh aplikasi *base station* dan diterima oleh robot dan dikirimkan kembali oleh robot kepada aplikasi *base station* adalah datanya sama. Tingkat dari uji coba tersebut adalah 100% data yang dikirimkan dan diterima memiliki kesamaan.

#### **4.2 Uji Status Robot Membawa Bola Pada Aplikasi *Base Station*.**

Pengujian ini berfungsi untuk mengetahui tingkat keberhasilan pada aplikasi *base station* dalam memberikan informasi bahwa robot membawa bola atau tidak membawa bola kepada user.

##### **4.2.1 Prosedur Pengujian Pada Uji Status Robot Membawa Bola Pada Aplikasi *Base Station*.**

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut :

1. Mempersiapkan aplikasi *base station* pada komputer untuk kontrol robot sepak bola beroda.
2. Mempersiapkan robot sepak bola beroda pada posisi awal.
3. Mengkoneksikan antara aplikasi *base station* dengan robot.
4. Memberi halangan kepada sensor ir pada robot yang diibaratkan itu sebagai bola.
5. Mengamati perubahan status dan perubahan visual robot pada aplikasi *base station*.

#### 4.2.3 Hasil Uji Status Robot Membawa Bola Pada Aplikasi *Base Station*.

Pada Tabel 4.4 adalah hasil uji status robot membawa bola pada aplikasi *base station*.

Tabel 4.4 Uji Status Robot Membawa Bola Pada Aplikasi *Base Station*.

Waktu detik ke	Bola	Status	Keterangan
1	Diberi bola	1	Berhasil
2	Diberi bola	1	Berhasil
3	Diberi bola	1	Berhasil
4	Diberi bola	1	Berhasil
5	Diberi bola	1	Berhasil
6	Diberi bola	1	Berhasil
7	Diberi bola	1	Berhasil
8	Diberi bola	1	Berhasil
9	Diberi bola	1	Berhasil
10	Diberi bola	1	Berhasil
11	Tidak diberi bola	0	Berhasil
12	Tidak diberi bola	0	Berhasil
13	Tidak diberi bola	0	Berhasil
14	Tidak diberi bola	0	Berhasil
15	Tidak diberi bola	0	Berhasil
16	Tidak diberi bola	0	Berhasil
17	Tidak diberi bola	0	Berhasil
18	Tidak diberi bola	0	Berhasil
19	Tidak diberi bola	0	Berhasil
20	Tidak diberi bola	0	Berhasil
21	Diberi bola	1	Berhasil
22	Diberi bola	1	Berhasil
23	Diberi bola	1	Berhasil
24	Diberi bola	1	Berhasil
25	Diberi bola	1	Berhasil
26	Diberi bola	1	Berhasil
27	Diberi bola	1	Berhasil
28	Diberi bola	1	Berhasil
29	Diberi bola	1	Berhasil
30	Diberi bola	1	Berhasil
31	Tidak diberi bola	0	Berhasil
32	Tidak diberi bola	0	Berhasil

Waktu detik ke	Bola	Status	Keterangan
33	Tidak diberi bola	0	Berhasil
34	Tidak diberi bola	0	Berhasil
35	Tidak diberi bola	0	Berhasil
36	Tidak diberi bola	0	Berhasil
37	Tidak diberi bola	0	Berhasil
38	Tidak diberi bola	0	Berhasil
39	Tidak diberi bola	0	Berhasil
40	Tidak diberi bola	0	Berhasil
41	Diberi bola	1	Berhasil
42	Diberi bola	1	Berhasil
43	Diberi bola	1	Berhasil
44	Diberi bola	1	Berhasil
45	Diberi bola	1	Berhasil
46	Diberi bola	1	Berhasil
47	Diberi bola	1	Berhasil
48	Diberi bola	1	Berhasil
49	Diberi bola	1	Berhasil
50	Diberi bola	1	Berhasil
51	Tidak diberi bola	0	Berhasil
52	Tidak diberi bola	0	Berhasil
53	Tidak diberi bola	0	Berhasil
54	Tidak diberi bola	0	Berhasil
55	Tidak diberi bola	0	Berhasil
56	Tidak diberi bola	0	Berhasil
57	Tidak diberi bola	0	Berhasil
58	Tidak diberi bola	0	Berhasil
59	Tidak diberi bola	0	Berhasil
60	Tidak diberi bola	0	Berhasil

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.4 ditunjukkan bahwa pengujian di atas diambil pengujian selama 60 detik, dan tiap 10 detiknya penulis memberikan bola dan 10 detik selanjutnya tidak memberikan bola. Status “1” menunjukkan bahwa robot sedang membawa bola dan status “0” menunjukkan bahwa robot tidak membawa bola. Pada uji coba di atas adalah tingkat keberhasilannya 100%.

#### **4.2.4 Kesimpulan Dari Semua Hasil Uji Status Robot Membawa Bola Pada Aplikasi *Base Station*.**

Berdasarkan dari semua hasil uji coba yang dilakukan pada hasil uji status robot membawa bola pada aplikasi *base station*. Terbukti dari beberapa percobaan di atas bahwa aplikasi *base station* mampu memberikan informasi status robot membawa bola kepada user dengan akurat. Tingkat keberhasilan dari uji coba tersebut adalah 100%.

#### **4.3 Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi *Base Station*.**

Pengujian ini berfungsi untuk mengetahui ketepatan koordinat yang diterima oleh aplikasi *base station* terhadap gambar visual robot. Nilai umpan balik dari proses odometry yang dikirim ke aplikasi *base station* akan diseleksi jika kurang dari 0,50 maka akan mengikuti pembulatan bawah, jika lebih dari sama dengan 0,50 maka akan ikut pembulatan atas.

##### **4.3.1 Prosedur Pengujian Pada Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi *Base Station*.**

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut :

1. Mempersiapkan aplikasi *base station* pada pc untuk kontrol robot.
2. Mempersiapkan robot pada posisi awal.
3. Mengkoneksikan antara aplikasi *base station* dengan robot.
4. Memberikan dan mengirim *input* koordinat dari aplikasi *base station*.
5. Mengamati pengiriman data hasil dari koordinat robot yang dikirim ke aplikasi *base station*.

6. Megamati hasil koordinat robot dan menyesuaikan dengan gambar visual robot. Jika terdapat nilai kurang dari 0,50 maka gambar akan mengikuti pembulatan bawahnya yaitu 0, jika terdapat nilai lebih dari sama dengan 0,50 maka gambar akan mengikuti pembulatan atasnya yaitu 1, jika tidak terjadi pembulatan pada *base station* maka terdapat *error*.

#### **4.3.2 Hasil Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi *Base Station*.**

Pada Tabel 4.5 berikut adalah hasil uji coba dari ketepatan gerak robot maju pada aplikasi *base station*.

Tabel 4.5 Uji Ketepatan Gerak Robot Maju Pada Aplikasi *Base Station*.

No	Input Dari Basestation (cm)		Input Data Dari Robot (cm)		Output Data Dari Robot (cm)		Output Dari Basestation (cm)		Error Output (cm)	
	X	Y	X	Y	X	Y	X	Y	X	Y
1	0	150	0	150	0	149,63	0	150	0	-0,37
2	0	150	0	150	0	149,56	0	150	0	-0,44
3	0	150	0	150	0	149,59	0	150	0	-0,41
4	0	150	0	150	0	150,12	0	150	0	0,12
5	0	150	0	150	0	149,79	0	150	0	-0,21
6	0	300	0	300	0,29	299,51	0	300	0,29	-0,49
7	0	300	0	300	0,29	299,62	0	300	0,29	-0,38
8	0	300	0	300	0,29	299,53	0	300	0,29	-0,47
9	0	300	0	300	0,29	299,84	0	300	0,29	-0,16
10	0	300	0	300	0,29	299,55	0	300	0,29	-0,45
11	0	450	0	450	0	450,24	0	450	0	0,24
12	0	450	0	450	0	449,66	0	450	0	-0,34
13	0	450	0	450	0	450,18	0	450	0	0,18
14	0	450	0	450	0	450,18	0	450	0	0,18
15	0	450	0	450	0,05	450,36	0	450	0,05	0,36
Rata-rata									0,1	-0,176

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.5 ditunjukkan bahwa nilai input koordinat (0,150), (0,300), (0,450) setelah di uji menghasilkan nilai *error* yang tidak begitu besar.

Sebagai contoh pada pengujian *input* (0,300) setelah pergerakan dari posisi awalnya robot sepak bola beroda mengirimkan posisinya berada pada koordinat X (0.29) dan Y (299.51). Setelah melalui proses pembulatan pada visualisasi maka gambar robot berada pada posisi koordinat X (0) dan Y (300) dan menghasilkan nilai *error* X (0.29) dan Y (-0.49) dengan rata rata nilai *error* yang tidak begitu besar perbedaannya dengan nilai *error* x (0.1) dan y (-0.176). Pada Tabel 4.6 berikut adalah hasil uji coba dari ketepatan gerak robot samping kanan pada aplikasi *base station*.

Tabel 4.6 Uji Ketepatan Gerak Robot Samping Kanan Pada Aplikasi *Base Station*.

No	<i>Input Dari Basestation (cm)</i>		<i>Input Data Dari Robot (cm)</i>		<i>Output Data Dari Robot (cm)</i>		<i>Output Dari Basestation (cm)</i>		<i>Error Output (cm)</i>	
	X	Y	X	Y	X	Y	X	Y	X	Y
1	100	0	100	0	100,20	0,06	100	0	0,2	0,06
2	100	0	100	0	100,26	0	100	0	0,26	0
3	100	0	100	0	100,26	0	100	0	0,26	0
4	100	0	100	0	100,37	0	100	0	0,37	0
5	100	0	100	0	100,32	0	100	0	0,32	0
6	200	0	200	0	200,04	0	200	0	0,04	0
7	200	0	200	0	200,22	0	200	0	0,22	0
8	200	0	200	0	200,28	0,06	200	0	0,28	0,06
9	200	0	200	0	200,45	0	200	0	0,45	0
10	200	0	200	0	200,45	0	200	0	0,45	0
11	300	0	300	0	300,42	0,12	300	0	0,42	0,12
12	300	0	300	0	300,12	0	300	0	0,12	0
13	300	0	300	0	300,42	0	300	0	0,42	0
14	300	0	300	0	300,42	0	300	0	0,42	0
15	300	0	300	0	300,06	0,06	300	0	0,06	0,06
Rata-rata									0,286	0,02

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.6 ditunjukkan bahwa nilai *input* koordinat (100,0), (200,0), (300,0) setelah di uji menghasilkan nilai *error* yang tidak begitu besar. Sebagai contoh pada pengujian *input* (100,0) setelah pergerakan dari posisi

awalnya robot sepak bola beroda mengirimkan posisinya berada pada koordinat X (100,20) dan Y (0,06). Setelah melalui proses pembulatan pada visualisasi maka gambar robot berada pada posisi koordinat X (100) dan Y (0) dan menghasilkan nilai *error* X (0,2) dan Y (0,06) dengan rata rata nilai *error* yang tidak begitu besar perbedaannya dengan nilai *error* X (0,286) dan Y (0,02). Pada Tabel 4.7 berikut adalah hasil uji coba dari ketepatan gerak robot serong pada aplikasi *base station*.

Tabel 4.7 Uji Ketepatan Gerak Robot Serong Pada Aplikasi *Base Station*.

No	<i>Input Dari Basestation (cm)</i>		<i>Input Data Dari Robot (cm)</i>		<i>Output Data Dari Robot (cm)</i>		<i>Output Dari Basestation (cm)</i>		<i>Error Output (cm)</i>	
	X	Y	X	Y	X	Y	X	Y	X	Y
1	100	250	100	250	99,79	249,97	100	250	-0,21	-0,03
2	100	250	100	250	99,96	249,73	100	250	-0,04	-0,27
3	100	250	100	250	99,79	250,38	100	250	-0,21	0,38
4	100	250	100	250	99,9	250,44	100	250	-0,1	0,44
5	100	250	100	250	100,02	250,44	100	250	0,02	0,44
6	200	250	200	250	199,98	250,32	200	250	-0,02	0,32
7	200	250	200	250	199,93	249,79	200	250	-0,07	-0,21
8	200	250	200	250	199,57	249,49	200	250	-0,43	-0,51
9	200	250	200	250	199,93	249,97	200	250	-0,07	-0,03
10	200	250	200	250	199,87	249,61	200	250	-0,13	-0,39
11	300	250	300	250	300,24	249,55	300	250	0,24	-0,45
12	300	250	300	250	299,89	249,64	300	250	-0,11	-0,36
13	300	250	300	250	299,48	250,2	300	250	-0,52	0,2
14	300	250	300	250	299,77	250,2	300	250	-0,23	0,2
15	300	250	300	250	299,89	249,64	300	250	-0,11	-0,36
Rata-rata									-0,133	-0,042

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.7 ditunjukkan bahwa nilai *input* koordinat (100,250), (200,250), (300,250) setelah di uji menghasilkan nilai *error* yang tidak begitu besar. Sebagai contoh pada pengujian *input* (100,250) setelah pergerakan dari posisi awalnya robot sepak bola beroda mengirimkan posisinya berada pada

koordinat X (99.79) dan Y (249.97). Setelah melalui proses pembulatan pada visualisasi maka gambar robot berada pada posisi koordinat X (100) dan Y (250) dan menghasilkan nilai *error* X (-0.21) dan Y (0,38) dengan rata-rata nilai *error* yang tidak begitu besar perbedaannya dengan nilai *error* X (-0,133) dan Y (-0,042).

#### **4.3.3 Kesimpulan Dari Semua Hasil Uji Ketepatan Koordinat Robot Sepak Bola Beroda Dengan Aplikasi *Base Station*.**

Dari semua hasil uji ketepatan koordinat robot sepak bola beroda dengan aplikasi *base station* terdapat *error* yang tidak begitu besar perbedaannya dan dapat ditoleransi.

#### **4.4 Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi *Base Station*.**

Pengujian ini berfungsi untuk mengetahui Kesamaan dari nilai koordinat robot sebenarnya dan nilai koordinat robot pada aplikasi *base station* yang diharapkan terdapat Kesamaan pergerakkan robot sebenarnya dengan robot pada aplikasi *base station*.

##### **4.4.1 Prosedur Pengujian Pada Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi *Base Station*.**

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut :

1. Mempersiapkan aplikasi *base station* pada pc untuk kontrol robot.
2. Mempersiapkan robot pada posisi awal.
3. Mengkoneksikan antara aplikasi *base station* dengan robot.
4. Memberikan dan mengirim *input* koordinat dari aplikasi *base station*.

5. Mengamati pengiriman data hasil dari koordinat robot yang dikirim ke aplikasi *base station*.
6. Megamati hasil koordinat robot dan menyesuaikan dengan gambar visual robot. Jika terdapat nilai kurang dari 0,50 maka gambar akan mengikuti pembulatan bawahnya yaitu 0, jika terdapat nilai lebih dari sama dengan 0,50 maka gambar akan mengikuti pembulatan atasnya yaitu 1, jika tidak terjadi pembulatan pada *base station* maka terdapat *error*.
7. Mengamati pergerakkan robot sebenarnya dan robot pada aplikasi *base station*.

#### **4.4.2 Hasil Uji Kesamaan Pergerakan Robot Sebenarnya Dengan Pergerakan Robot Di Aplikasi *Base Station*.**

Pada Tabel 4.8 berikut adalah hasil Kesamaan pergerakan robot maju dengan koordinat X adalah 0 dan Y adalah 100 pada aplikasi *base station*.

Tabel 4.8 Uji Coba Kesamaan Gerak Robot Maju.

Waktu Detik Ke	Input koordinat X : 0 dan Y : 100							
	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
X	Y	X	Y	X	Y	X	Y	
1	0,06	0,59	0	0	0	0	0,06	0,59
	0,06	0,59	0	0	0	0	0,06	0,59
	0,06	0,59	0	0	0	0	0,06	0,59
	0,06	0,59	0	0	0	0	0,06	0,59
2	0,06	0,59	0	0	0	0	0,06	0,59
	0,06	2,29	0	1	0	2	0,06	0,29
	0,06	12,35	0	6	0	12	0,06	0,35
	0,06	16,05	0	8	0	16	0,06	0,05
3	0,06	27,28	0	14	0	28	0,06	-0,72
	0	31,11	0	16	0	32	0	-0,89
	0,06	42,51	0	21	0	42	0,06	0,51
	0,06	50,22	0	25	0	50	0,06	0,22

Input koordinat X : 0 dan Y : 100								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
	X	Y	X	Y	X	Y	X	Y
4	0,06	53,98	0	27	0	54	0,06	-0,02
	0	65,5	0	33	0	66	0	-0,5
	0	69,27	0	35	0	70	0	-0,73
	0,06	80,91	0	40	0	80	0,06	0,91
5	0,06	88,5	0	44	0	88	0,06	0,5
	0,06	92,38	0	46	0	92	0,06	0,38
	0	100,73	0	50	0	100	0	0,73
	0	100,73	0	50	0	100	0	0,73
Rata-rata						0,045	0,238	

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.8 ditunjukkan bahwa nilai *input* koordinat (0,100) setelah di uji menghasilkan nilai rata-rata *error* X (0.045) dan Y (0.238). Dari hasil uji coba di atas memerlukan waktu sebanyak 5 detik untuk mencapai sebuah target yang telah di *input* kan. Setiap 1 detiknya terdapat 4 data yang masuk. Pada Tabel 4.9 berikut adalah hasil Kesamaan pergerakan robot maju dengan koordinat X adalah 0 dan Y adalah 200 pada aplikasi *base station*.

Tabel 4.9 Uji Coba Kesamaan Gerak Robot Maju.

Input koordinat X : 0 dan Y : 200								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
	X	Y	X	Y	X	Y	X	Y
1	0	0,76	0	0	0	0	0	0,76
	0	0,76	0	0	0	0	0	0,76
	0	0,76	0	0	0	0	0	0,76
	0	0,76	0	0	0	0	0	0,76
2	0	0,76	0	0	0	0	0	0,76
	0,06	5,47	0	3	0	6	0,06	-0,53
	0,06	12,58	0	6	0	12	0,06	0,58

Input koordinat X : 0 dan Y : 200								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
	X	Y	X	Y	X	Y	X	Y
	0,06	16,35	0	8	0	16	0,06	0,35
3	0	27,64	0	14	0	28	0	-0,36
	0,06	35,28	0	18	0	36	0,06	-0,72
	0,06	42,87	0	21	0	42	0,06	0,87
	0,06	50,45	0	25	0	50	0,06	0,45
4	0	58,04	0	29	0	58	0	0,04
	0,06	65,68	0	33	0	66	0,06	-0,32
	0	73,27	0	37	0	74	0	-0,73
	0,06	81,03	0	41	0	82	0,06	-0,97
5	0,06	88,61	0	44	0	88	0,06	0,61
	0,06	96,26	0	48	0	96	0,06	0,26
	0,06	103,9	0	52	0	104	0,06	-0,1
	0,06	111,61	0	56	0	112	0,06	-0,39
6	0,06	119,31	0	60	0	120	0,06	-0,69
	0,06	127,01	0	64	0	128	0,06	-0,99
	0	134,66	0	67	0	134	0	0,66
	0	142,36	0	71	0	142	0	0,36
7	0,06	150,06	0	75	0	150	0,06	0,06
	0,06	157,76	0	79	0	158	0,06	-0,24
	0,06	165,47	0	83	0	166	0,06	-0,53
	0	173,11	0	87	0	174	0	-0,89
8	0,06	180,87	0	90	0	180	0,06	0,87
	0,06	188,52	0	94	0	188	0,06	0,52
	0,06	196,28	0	98	0	196	0,06	0,28
	0	200,63	0	100	0	200	0	0,63
Rata-rata						0,038	0,09	

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.9 ditunjukkan bahwa nilai *input* koordinat (0,200) setelah di uji menghasilkan nilai rata-rata *error* X (0,038) dan Y (0,09) . Dari hasil uji coba di atas memerlukan waktu sebanyak 8 detik untuk mencapai sebuah target yang telah di *input* kan. Setiap 1 detiknya terdapat 4 data yang masuk. Pada

Tabel 4.10 berikut adalah hasil Kesamaan pergerakan robot maju dengan koordinat X adalah 0 dan Y adalah 300 pada aplikasi *base station*.

Tabel 4.10 Uji Coba Kesamaan Gerak Robot Maju.

Input koordinat X : 0 dan Y : 300									
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)		
	X	Y	X	Y	X	Y	X	Y	
1	0,24	0,76	0	0	0	0	0,24	0,76	
	0,24	0,76	0	0	0	0	0,24	0,76	
	0,18	19,7	0	10	0	20	0,18	-0,3	
	0,18	27,17	0	14	0	28	0,18	-0,83	
2	0,24	34,63	0	17	0	34	0,24	0,63	
	0,18	42,1	0	21	0	42	0,18	0,1	
	0,18	49,57	0	25	0	50	0,18	-0,43	
	0,18	57,04	0	29	0	58	0,18	-0,96	
3	0,18	64,56	0	32	0	64	0,18	0,56	
	0,18	72,03	0	36	0	72	0,18	0,03	
	0,18	79,62	0	40	0	80	0,18	-0,38	
	0,18	87,09	0	44	0	88	0,18	-0,91	
4	0,18	94,61	0	47	0	94	0,18	0,61	
	0,18	102,14	0	51	0	102	0,18	0,14	
	0,18	109,72	0	55	0	110	0,18	-0,28	
	0,18	117,19	0	59	0	118	0,18	-0,81	
5	0,18	124,72	0	62	0	124	0,18	0,72	
	0,18	132,19	0	66	0	132	0,18	0,19	
	0,18	139,65	0	70	0	140	0,18	-0,35	
	0,18	147,24	0	74	0	148	0,18	-0,76	
6	0,18	154,77	0	77	0	154	0,18	0,77	
	0,18	162,29	0	81	0	162	0,18	0,29	
	0,18	173,52	0	87	0	174	0,18	-0,48	
	0,18	177,35	0	89	0	178	0,18	-0,65	
7	0,18	184,93	0	92	0	184	0,18	0,93	
	0,18	192,4	0	96	0	192	0,18	0,4	
	0,18	199,93	0	100	0	200	0,18	-0,07	
	0,18	211,21	0	106	0	212	0,18	-0,79	
8	0,18	214,98	0	107	0	214	0,18	0,98	
	0,18	226,27	0	113	0	226	0,18	0,27	
	0,18	230,03	0	115	0	230	0,18	0,03	

<i>Input koordinat X : 0 dan Y : 300</i>								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		<i>Error</i> (cm)	
	X	Y	X	Y	X	Y	X	Y
	0,24	237,56	0	119	0	238	0,24	-0,44
9	0,18	248,79	0	124	0	248	0,18	0,79
	0,18	252,61	0	126	0	252	0,18	0,61
	0,18	263,9	0	132	0	264	0,18	-0,1
	0,18	267,66	0	134	0	268	0,18	-0,34
10	0,18	275,13	0	138	0	276	0,18	-0,87
	0,18	286,48	0	143	0	286	0,18	0,48
	0,18	290,19	0	145	0	290	0,18	0,19
	0,18	300,18	0	150	0	300	0,18	0,18
Rata-rata						0,186	0,017	

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.10 ditunjukkan bahwa nilai *input* koordinat (0,200) setelah di uji menghasilkan nilai rata-rata *error* X (0,186) dan Y (0,017) . Dari hasil uji coba di atas memerlukan waktu sebanyak 10 detik untuk mencapai sebuah target yang telah di *input* kan. Setiap 1 detiknya terdapat 4 data yang masuk. Pada Tabel 4.11 berikut adalah hasil Kesamaan pergerakan robot maju dengan koordinat X adalah 0 dan Y adalah 400 pada aplikasi *base station*.

Tabel 4.11 Uji Coba Kesamaan Gerak Robot Maju.

<i>Input koordinat X : 0 dan Y : 400</i>								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		<i>Error</i> (cm)	
	X	Y	X	Y	X	Y	X	Y
1	0,12	0,12	0	0	0	0	0,12	0,12
	0,12	0,12	0	0	0	0	0,12	0,12
	0,12	0,12	0	0	0	0	0,12	0,12
	0,12	4,76	0	2	0	4	0,12	0,76
2	0,06	8,11	0	4	0	8	0,06	0,11
	0,12	18,88	0	9	0	18	0,12	0,88

Input koordinat X : 0 dan Y : 400								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
	X	Y	X	Y	X	Y	X	Y
3	0,12	22,52	0	11	0	22	0,12	0,52
	0,18	29,75	0	15	0	30	0,18	-0,25
	0,18	40,51	0	20	0	40	0,18	0,51
	0,18	44,1	0	22	0	44	0,18	0,1
4	0,18	54,86	0	27	0	54	0,18	0,86
	0,18	58,45	0	29	0	58	0,18	0,45
	0,18	65,62	0	33	0	66	0,18	-0,38
	0,18	76,32	0	38	0	76	0,18	0,32
5	0,18	79,91	0	40	0	80	0,18	-0,09
	0,18	90,67	0	45	0	90	0,18	0,67
	0,18	94,26	0	47	0	94	0,18	0,26
	0,18	105,02	0	53	0	106	0,18	-0,98
6	0,18	112,13	0	56	0	112	0,18	0,13
	0,18	115,72	0	58	0	116	0,18	-0,28
	0,18	126,48	0	63	0	126	0,18	0,48
	0,18	130,07	0	65	0	130	0,18	0,07
7	0,18	140,83	0	70	0	140	0,18	0,83
	0,18	147,94	0	74	0	156	0,18	-8,06
	0,18	155,12	0	78	0	156	0,18	-0,88
	0,18	162,35	0	81	0	162	0,18	0,35
8	0,12	166,06	0	83	0	166	0,12	0,06
	0	177,05	0	89	0	178	0	-0,95
	0,06	184,4	0	92	0	184	0,06	0,4
	0,18	191,75	0	96	0	192	0,18	-0,25
9	0,18	198,98	0	99	0	198	0,18	0,98
	0,12	202,63	0	101	0	202	0,12	0,63
	0	213,74	0	107	0	214	0	-0,26
	0,06	221,15	0	111	0	222	0,06	-0,85
10	0,12	228,56	0	114	0	228	0,12	0,56
	0,06	235,97	0	118	0	236	0,06	-0,03
	0,12	243,38	0	122	0	244	0,12	-0,62
	0,06	250,79	0	125	0	250	0,06	0,79
11	0,12	258,2	0	129	0	258	0,12	0,2
	0,06	265,61	0	133	0	266	0,06	-0,39
11	0,06	273,07	0	137	0	274	0,06	-0,93

Input koordinat X : 0 dan Y : 400								
Waktu Detik Ke	Gerak Robot (cm)		Gerak Visual (pixels)		Gerak Visual (cm)		Error (cm)	
	X	Y	X	Y	X	Y	X	Y
12	0,06	280,48	0	140	0	280	0,06	0,48
	0,12	287,89	0	144	0	288	0,12	-0,11
	0,06	295,3	0	148	0	296	0,06	-0,7
	0,06	302,77	0	151	0	302	0,06	0,77
13	0,12	310,18	0	155	0	310	0,12	0,18
	0,12	317,65	0	159	0	318	0,12	-0,35
	0,06	325,05	0	163	0	326	0,06	-0,95
	0,12	332,52	0	166	0	332	0,12	0,52
14	0,06	339,93	0	170	0	340	0,06	-0,07
	0,06	347,4	0	174	0	348	0,06	-0,6
	0,12	354,81	0	177	0	354	0,12	0,81
	0,06	362,22	0	181	0	362	0,06	0,22
15	0,06	369,69	0	185	0	370	0,06	-0,31
	0,12	377,09	0	189	0	378	0,12	-0,91
	0,06	388,33	0	194	0	388	0,06	0,33
	0,06	391,97	0	196	0	392	0,06	-0,03
	0,12	399,38	0	200	0	400	0,12	-0,62
	0,18	399,88	0	200	0	400	0,18	-0,12
	0	400,15	0	200	0	400	0	0,15
	Rata-rata					0,119	-0,087	

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.11 ditunjukkan bahwa nilai *input* koordinat (0,200) setelah di uji menghasilkan nilai rata-rata *error* X (0,119) dan Y (-0,087). Dari hasil uji coba di atas memerlukan waktu sebanyak 15 detik untuk mencapai sebuah target yang telah di *input* kan. Setiap 1 detiknya terdapat 4 data yang masuk.

#### 4.4.3 Kesimpulan Dari Semua Hasil Uji Kesamaan Pergerakan Robot

##### **Sebenarnya Dengan Pergerakan Robot Di Aplikasi *Base Station*.**

Berdasarkan dari semua hasil uji coba yang dilakukan pada hasil uji Kesamaan pergerakan robot sebenarnya dengan pergerakan robot di aplikasi *base station* yaitu terdapat Kesamaan nilai dari nilai pergerakan robot sebenarnya dengan robot pada aplikasi *base station*. Terdapat *error* yang tidak begitu besar perbedaanya dan dapat di toleransi.

#### 4.5 Uji Kecepatan Pengiriman Data Robot Ke Aplikasi *Base Station*.

Pengujian ini berfungsi untuk mengetahui range periode pengiriman terbaik untuk pengiriman data dari robot ke aplikasi *base station*. Pemilihan waktu periode pengiriman dapat mempengaruhi cepat lambatnya pengiriman data dan gerak robot pada aplikasi *base station*.

##### **4.5.1 Prosedur Pengujian Pada Uji Kecepatan Pengiriman Data Robot Ke Aplikasi *Base Station*.**

Langkah-langkah yang dilakukan dalam melakukan pegujian ini adalah sebagai berikut :

1. Upload program robot sesuai periode pengiriman yang telah diatur.
2. Mempersiapkan aplikasi *base station* pada pc untuk kontrol robot.
3. Mempersiapkan robot pada posisi awal.
4. Mengkoneksikan antara aplikasi *base station* dengan robot.
5. Mengamati pengiriman data dan menghitung berapa banyak data yang berhasil diterima oleh *base station*.

#### 4.5.2 Hasil Uji Kecepatan Pengiriman Data Robot Ke Aplikasi *Base Station*.

Pada Tabel 4.23 adalah hasil uji kecepatan pengiriman data robot ke aplikasi *base station*.

Tabel 4.12 Uji Kecepatan Pengiriman Data Robot Ke Aplikasi *Base Station*

Percobaan Ke	Periode Pengiriman (milidetik)	Jumlah Data Yang Terima
1	668	31
2	634	33
3	601	34
4	568	36
5	534	39
6	501	41
7	468	43
8	434	46
9	401	50
10	367	54
11	334	60
12	301	67
13	267	76
14	234	85
15	200	97
16	167	97
17	134	97
18	84	97
19	50	97
20	33	97
21	17	0

(Sumber: Olahan Penulis)

Berdasarkan Tabel 4.23 ditunjukkan bahwa setiap pengujian di atas diambil pengujian selama 20 detik. Pengujian ini hanya merubah nilai periode pengiriman untuk menguji nilai periode pengiriman yang cocok dan tidak ada data *error*.

#### 4.5.3 Kesimpulan Dari Semua Hasil Uji Kecepatan Pengiriman Data Robot

##### Ke Aplikasi *Base Station*.

Berdasarkan dari semua hasil uji coba yang dilakukan pada hasil uji kecepatan pengiriman data robot ke aplikasi *base station*. Mendapat nilai *range* periode pengiriman terbaik untuk pengiriman data yaitu, *range* periode pengiriman 33 sampai 200 milidetik. *Range* periode pengiriman 33 sampai 200 milidetik memiliki jumlah data terbanyak. Untuk percobaan 21 menyimpulkan kegagalan dalam pengiriman data, dapat terbukti data yang diterima adalah 0.



## BAB V

### PENUTUP

Dalam bab ini berisi tentang kesimpulan dan saran berdasarkan dari hasil pengujian yang telah dilakukan pada Tugas Akhir ini. Dalam pengerjaan Tugas Akhir ini diperoleh kesimpulan dan saran agar bisa mengembangkan sistem lebih baik lagi kedepannya.

#### 5.1 Kesimpulan

Berdasarkan pengujian pada Tugas Akhir diperoleh kesimpulan dari beberapa poin sebagai berikut :

1. Hasil yang diperoleh dari proses uji validasi komunikasi antara robot dengan aplikasi *base station*, terbukti data yang dikirimkan dan diterima memiliki kesamaan dan tingkat uji coba tersebut adalah 100%.
2. Hasil yang diperoleh dari hasil uji kesamaan pergerakan robot sebenarnya dengan pergerakan robot pada aplikasi *base station* dengan tingkat rata-rata *error* yang tidak besar dan dapat di toleransi, dengan ketentuan  $1 \text{ pixel} = 2 \text{ cm}$  dan hasilnya baik.
3. Hasil yang diperoleh dari uji status robot membawa bola pada aplikasi *base station*, terbukti dari beberapa percobaan yang telah dilakukan aplikasi *base station* mampu memberikan informasi status membawa bola dengan akurat, tingkat keberhasilannya adalah 100%.
4. Hasil koordinat yang dikirim dari robot tidak selalu bernilai angka bulat, maka dari itu harus dibulatkan oleh sistem agar dapat diterima

oleh visualisasi posisi robot. Pembulatan dilakukan setelah proses penerimaan data oleh aplikasi visualisasi, dimana jika kurang dari 0,50 maka akan ikut pembulatan bawah, jika lebih besar sama dengan 0,50 akan ikut pembulatan atas.

5. Hasil yang diperoleh dari hasil uji kecepatan pengiriman data robot ke aplikasi *base station* mendapat kesimpulan bahwa nilai aman buat pengiriman penelitian ini adalah *range* periode pengiriman 33 sampai 200 milidetik karena memiliki jumlah data terbanyak. Dan periode pengiriman 17 milidetik sangat tidak cocok untuk diterapkan karena data yang diterima adalah 0 atau tidak ada data yang masuk.

## 5.2 Saran

Saran dari penulis pada pengembangan Tugas Akhir ini agar kedepannya dapat dikembangkan lebih baik lagi adalah sebagai berikut :

1. Pemetaan robot terhadap garis tepi dari agar robot bisa dikendalikan tidak keluar dari garis tepi lapangan.
2. Robot dapat mengetahui daerah letak gawang lawan melalui visualisasi posisi robot.
3. Robot dapat mengetahui daerah sendiri dan daerah lawan melalui visualisasi posisi robot.

## DAFTAR PUSTAKA

- Abseno, A. P. (2019). *Penerapan Kinematika Untuk Lokalisasi Pada Robot Sepak Bola Beroda*. Surabaya: Institut Bisnis dan Informatika Stikom Surabaya.
- Ardhiansyah, T., & Syarifuddin, I. (2017). *Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Refree Box Menggunakan Base Station*. *5th Indonesian Symposium on Robotic System and Control*, 5.
- Circuit, I. (2019). *NodeMCU ESP8266: Details and Pinout*. Retrieved 11 1, 2019, from <https://www.instructables.com/id/NodeMCU-ESP8266-Details-and-Pinout/>
- Due, A. (2017). *ARDUINO DUE*. Retrieved 9 16, 2019, from <https://store.arduino.cc/usa/arduino-due>
- Efendi, M. Y., & Chandra, J. E. (2019). *Implementasi Internet of Things Pada Sistem Kendali Lampu Rumah Menggunakan Telegram Messenger Bot Dan Nodemcu Esp 8266*. *Global Journal of Computer Science and Technology*, Volume 19 (1 Version), 15-25.
- Indrarta, V. (2019). *PERANCANGAN VISUALISASI POSISI ROBOT PADA PENGENDALI ROBOT SEPAK BOLA BERODA*. *Journal of Control and Network Systems*.
- Karnadi, D. A. (2019). *RANCANG BANGUN KONTROL KECEPATAN ROBOT THREE OMNI-DIRECTIONAL MENGGUNAKAN METODE PID (Proportional Integral Derivative)*. *Journal of Control and Network Systems*.

Kementrian Riset, T. d. (2019). *Buku Panduan KRSBI Beroda 2019*. Retrieved from <http://kontesrobotindonesia.id/tentang-kri.html>

Thingbits. (2019). *3WD 48mm Omni-Directional Triangle Mobile Robot Chassis*.

Retrieved September 08, 2019, from  
<https://www.thingbits.net/products/3wd-48mm-omni-directional-triangle-mobile-robot-chassis>

Wahyuni, E. D. (2015). *Rancang Bangun Program Visualisasi Pergerakan*.

Surabaya: Politeknik Elektronika Negeri Surabaya.

