



**SINKRONISASI PROTOKOL UDP UNTUK KOMUNIKASI ROBOT
SEPAK BOLA BERODA INDONESIA**



Oleh :

MUHAMMAD AKBAR SUHARBI

15410200049

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2019**

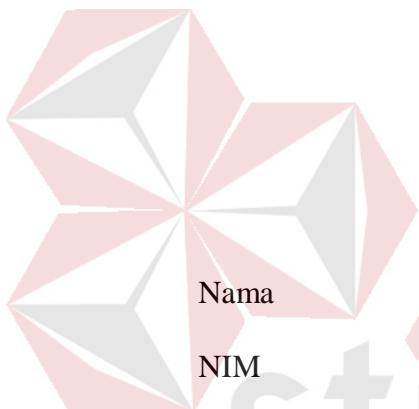
SINKRONISASI PROTOKOL UDP UNTUK KOMUNIKASI ROBOT

SEPAK BOLA BERODA INDONESIA

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Disusun Oleh :

Nama

: Muhammad Akbar Suharbi

NIM

: 15410200049

Program

: S1 (Strata Satu)

Jurusan

: Sistem Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

TUGAS AKHIR

SINKRONISASI PROTOKOL UDP UNTUK KOMUNIKASI ROBOT

SEPAK BOLA BERODA INDONESIA

Dipersiapkan dan disusun oleh

Muhammad Akbar Suharbi

NIM : 15410200049

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada :

Susunan Dewan Penguji

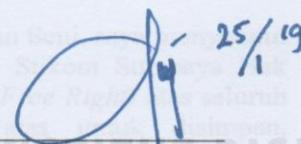
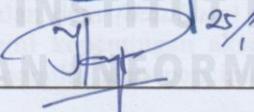
Pembimbing

I. Dr. Jusak
NIDN. 0708017101

II. Ira Puspasari, S.Si., M.T.
NIDN. 0710078601

Pembahas

I. Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501


25/1/19

25/1/19

25/01/2019



Dr. Jusak

Dekan Fakultas Teknologi dan Infomatika

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Muhammad Akbar Suharbi
NIM : 15410200049
Program Studi : S1 Sistem Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **SINKRONISASI PROTOKOL UDP UNTUK KOMUNIKASI ROBOT SEPAK BOLA BERODA INDONESIA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya 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, 28 Januari 2019
Yang menyatakan



Muhammad Akbar Suharbi
NIM : 15410200049



“Ijazah Itu Tanda Orang Pernah Bersekolah

Bukan Tanda Orang Pernah Berfikir”

~Muhammad Akbar Suharbi~

v

PERSEMBAHAN

Alhamdulillahi robbil 'alamin

Puji syukur atas karunia, rejeki dan rahmat Allah SWT berikan kepada saya, sehingga dapat menyelesaikan Tugas Akhir ini secara tepat waktu. Alhamdulillah selesai sudah masa kuliah saya yang ditempuh selama 3,5 tahun ini. Keberhasilan menyusun Tugas Akhir ini juga tidak lepas dari dukungan dan bantuan orang-orang yang selalu ada untuk saya. Tugas Akhir ini saya persembahkan untuk :

Orang Tua Tersayang

Terimakasih kepada mama dan papa tercinta yang telah merawat, mendidik saya hingga dewasa, memberikan kasih sayang yang begitu besar, membiayai sekolahku sampai sarjana, memberi semangat dan dukungan, serta selalu mendoakan di setiap waktu agar aku menjadi anak yang selalu berbakti, sukses, dan bermanfaat bagi orang lain. Memang apa yang aku lakukan tidak sebanding pengorbanan mama dan papa berikan, namun membuat kalian tersenyum adalah kebanggaan tersendiri bagiku.

Kakak Tersayang

Terimakasih kepada kakak kandungku tersayang **Rahmania Desliyanti Putri** yang selalu jahilin adik, namun selalu bikin kangen karena jarang di rumah. Terimakasih atas seluruh dukungan, doa, dan perhatian kakak kepadaku, sehingga adik bisa menyelesaikan Tugas Akhir ini tepat waktu. Meskipun tidak pernah diungkapkan lewat lisan kalau kakak menyayangiku, namun adik tahu kalau kakak sayang sama aku.

Kekasih Tercinta

Terimakasih kepada kekasihku **Andini Dwikke Wardani** yang suka bawel, jahil, dan manja. Kamu adalah teman, sahabat, sekaligus kekasih yang selalu ada saat aku senang maupun duka. Terimakasih sudah berperan penting dalam menyelesaikan buku ini, terimakasih selalu memberikanku semangat, menemaniku, medoakanku, dan perhatianmu ketika aku mulai menyerah. Terimakasih sayang berkat dukunganmu aku bisa menyelesaikan Tugas Akhir ini dan lulus 3,5 tahun.

Team Robot SWC-8

Terimakasih atas doa, dukungan, dan batuan yang kalian berikan (**Vicky, Pepe, Dio, Faris, Raga, Misky, Angga**) selama 3 tahun ini sudah menjadi teman terbaikku. Tanpa bantuan dan dukungan kalian tidak mungkin aku bisa menyelesaikan Tugas Akhir ini. Aku harap kita semua bisa lulus bersama seperti kita berjuang bersama di bengkel Robot.

ABSTRAKSI

Seiring perkembangan teknologi, banyak para peneliti yang melakukan riset di bidang olahraga sepak bola, hingga tahun 1995 munculah gagasan pertama kali untuk menciptakan sebuah robot yang dapat bermain sepak bola, dan robot itu dinamakan *robot soccer*. Kompetisi *robot soccer* di Indonesia khususnya MSL telah diselenggarakan dua kali pada tahun 2017 dan 2018. Robot yang dilombakan harus mampu mendekripsi bola, menendangnya ke gawang, serta mampu menerima perintah dari *Refree Box* untuk bergerak secara otomatis tanpa campur tangan manusia.

Dari beberapa kekurangan yang dialami tim KRSBI Beroda 2018, penulis mengangkat salah satu kekurangan tersebut untuk dilakukan penelitian lebih lanjut. Kekurangan tersebut adalah adanya permasalahan komunikasi antara *Base Station* dan robot, yaitu *delay* tinggi serta koneksi komunikasi hanya berjalan satu arah. Pada penelitian ini, penulis mengembangkan konsep baru yaitu sinkronisasi komunikasi antara *Base Station* dan robot dengan menggunakan protokol UDP.

Pengujian terhadap sinkronisasi protokol UDP dilakukan sebanyak 90 kali pengambilan data dengan parameter yang diujikan adalah *packet loss*, *delay*, dan reaksi robot. Hasil dari pengujian ini menunjukkan bahwa UDP dengan sinkronisasi mempunyai *packet loss* rata-rata sebesar 0.013% untuk robot 1 dan 0.014% untuk robot 2 sedangkan besar *delay* rata-rata 13.30 ms untuk robot 1 dan 20.62 ms untuk robot 2 dengan skenario reaksi robot yang telah ditentukan.

Kata Kunci : *Packet Loss*, *Delay*, dan KRSBI-Beroda

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur kepada Tuhan Yang Maha Esa karena berkat, rahmat, dan karuniaNyalah penulis dapat menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul “Sinkronisasi Protokol UDP Untuk Komunikasi Robot Sepak Bola Berdoa Indonesia” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Institut Bisnis dan Informatika Stikom Surabaya

Pada kesempatan kali ini penulis juga ingin mengucapkan terima kasih kepada :

1. Bapak Pauladie Susanto, S.Kom., M.T, selaku Ketua Program Studi S1 Sistem Komputer dan selaku Dosen Penguji, yang telah memberikan kritik dan saran dalam penulisan Tugas Akhir ini.
2. Dr. Jusak, selaku Dekan Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika STIKOM Surabaya dan Dosen Pembimbing pertama yang telah membimbing, memberikan saran dan kritik untuk memberikan semangat penulis dalam menyelesaikan Tugas Akhir.
3. Ibu Ira Puspasari, S.Si., M.T., selaku Dosen Pembimbing kedua yang senantiasa memberikan dukungan dan arahan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
4. Bapak Heri Pratikno, M.T., selaku Dosen Wali yang selalu memberikan nasihat, dorongan, dan dukungan selama proses perkuliahan.

5. Seluruh Dosen Pengajar Program Studi S1 Sistem Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Institut Bisnis dan Informatika STIKOM Surabaya.
6. Teman-teman angkatan 2015 maupun adik dan kakak angkatan Jurusan S1 Sistem Komputer yang mendukung dan membantu penulis selama masa dan penyusunan buku Tugas Akhir ini.
7. Seluruh pihak yang tidak dapat penulis tuliskan satu persatu yang telah membantu penulis secara langsung maupun tidak langsung.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang besar jika terdapat kata-kata yang salah serta menyenggung perasaan pembaca. Akhir kata penulis ucapkan banyak terima kasih yang besar kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, 28 Januari 2019

Penulis

DAFTAR ISI

PERSEMBAHAN.....	vi
ABSTRAKSI.....	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan	4
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 <i>User Datagram Protocol (UDP)</i>	6
2.2 <i>Multicast</i>	7
2.3 <i>Socket Programming</i>	8
2.4 <i>Thread</i>	10
2.5 <i>BackgroundWorker</i>	11
2.6 <i>Refree Box</i>	12
2.7 <i>Base Station</i>	15
2.8 <i>Packet Loss</i>	18
2.9 <i>Delay</i>	20

2.10	Wireshark	21
2.11	Visual Studio 2013	22
2.12	Robot KRSBI Beroda Stikom Surabaya 2018	23
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM		24
3.1	Metode Penelitian	24
3.2	Blok Diagram Sistem Komunikasi	24
3.3	Perancangan Sistem	26
3.3.1	Konfigurasi <i>Mulicast</i>	27
3.3.2	Konfigurasi IP <i>Address</i>	30
3.3.3	Perancangan <i>Base Station</i>	34
3.4	Diagram Waktu Sinkronisasi	36
3.5	Pengujian Perancangan Sistem	45
BAB IV HASIL PENGUJIAN DAN PEMBAHASAN		47
4.1	Pengujian <i>Quality of Service Packet Loss</i>	47
4.1.1	Tujuan Pengujian <i>Packet Loss</i>	47
4.1.2	Komponen Yang Dibutuhkan Pada Pengujian <i>Packet Loss</i>	47
4.1.3	Prosedur Pengujian <i>Packet Loss</i>	48
4.1.4	Hasil Pengujian <i>Packet Loss</i>	48
4.2	Pengujian <i>Quality of Service Delay</i>	53
4.2.1	Tujuan Pengujian <i>Delay</i>	53
4.2.2	Komponen Yang Dibutuhkan Pada Pengujian <i>Delay</i>	53
4.2.3	Prosedur Pengujian <i>Delay</i>	54
4.2.4	Hasil Pengujian <i>Delay</i>	54
4.3	Pengujian Reaksi Robot	59

4.3.1	Tujuan Pengujian Reaksi Robot	59
4.3.2	Komponen Yang Dibutuhkan Pada Pengujian Reaksi Robot	59
4.3.3	Prosedur Pengujian Reaksi Robot	60
4.3.4	Hasil Pengujian Reaksi Robot.....	60
4.4	Pengujian Jarak Jangkauan Media <i>Receiver</i>	82
4.4.1.	Tujuan Pengujian Jarak Jangkauan Media <i>Receiver</i>	82
4.4.2.	Komponen Yang Dibutuhkan Pada Pengujian Jarak Jangkauan Media <i>Receiver</i>	82
4.4.3.	Prosedur Pengujian Jarak Jangkauan Media <i>Receiver</i>	83
4.4.4.	Hasil Pengujian Jarak Jangkauan Media <i>Receiver</i>	83
BAB V PENUTUP		89
5.1	Kesimpulan.....	89
5.2	Saran	90
DAFTAR PUSTAKA		91
LAMPIRAN		93
BIODATA MAHASISWA		124

DAFTAR GAMBAR

Gambar 2. 1 <i>User Datagram</i> Format	7
Gambar 2. 2 One-to-Many Multicast	8
Gambar 2. 3 Ilustrasi menggunakan <i>Socekt Stream</i>	9
Gambar 2. 4 Ilustrasi menggunakan <i>Socekt Datagram</i>	9
Gambar 2. 5 Ilustrasi Cara Kerja Thread	10
Gambar 2. 6 Ilustrasi Cara Kerja <i>Thread Sleep</i>	11
Gambar 2. 7 Ilustrasi Cara Kerja <i>BacgroundWorker</i>	12
Gambar 2. 8 Aplikasi <i>Refree Box</i>	13
Gambar 2. 9 <i>Base Station</i> Tim Institut dan Informatika Stikom Surabaya	15
Gambar 2. 10 <i>Wireshark</i>	22
Gambar 2. 11 Logo Visual Studio	22
Gambar 2. 12 Robot Tim SYNA-SWC8	23
Gambar 3. 1 Blok Diagram Rancangan Pengujian Komunikas	24
Gambar 3. 2 Alur Komunikasi Data Robot Tim Institut dan Informatika Stikom Surabaya	26
Gambar 3. 3 Tampilan Utama Winbox	28
Gambar 3. 4 Konfigurasi <i>Multicast</i> pada <i>Interface Bridge1</i>	29
Gambar 3. 5 Tampilan IGMP Groups dari <i>Interface bridge1</i>	29
Gambar 3. 6 Robot Penyerang telah bergabung kedalam Multicast Group	30
Gambar 3. 7 Robot Bertahan telah bergabung kedalam Multicast Group	30
Gambar 3. 8 Tampilan Login pada Winbox	31
Gambar 3. 9 Konfigurasi IP Server	32

Gambar 3. 10 Konfigurasi DHCP Server pada Bridge1	33
Gambar 3. 11 Daftar Perangkat yang didaftarkan kepada DHCP Server	33
Gambar 3. 12 Diagram Waktu antara <i>Base Station</i> dan Robot Penyerang (a).....	37
Gambar 3. 12 Diagram Waktu antara <i>Base Station</i> dan Robot Penyerang (b)	38
Gambar 3. 12 Diagram Waktu antara <i>Base Station</i> dan Robot Penyerang (c).....	39
Gambar 3. 12 Diagram Waktu antara <i>Base Station</i> dan Robot Penyerang (d)	40
Gambar 3. 13 Diagram Waktu antara <i>Base Station</i> dan Robot Bertahan (a)	41
Gambar 3. 13 Diagram Waktu antara <i>Base Station</i> dan Robot Bertahan (b).....	42
Gambar 3. 13 Diagram Waktu antara <i>Base Station</i> dan Robot Bertahan (c)	43
Gambar 3. 13 Diagram Waktu antara <i>Base Station</i> dan Robot Bertahan (d).....	44
Gambar 4. 1 Posisi awal kedua robot.....	62
Gambar 4. 2 Robot 1 maju 151 cm dari posisi awal.....	62
Gambar 4. 3 Robot 2 maju 123 cm dari posisi awal.....	63
Gambar 4. 4 Robot 1 mundur 297 cm dari posisi terakhir	63
Gambar 4. 6 Robot 1 berotasi sebesar 745°	64
Gambar 4. 5 Robot 2 mundur 165 cm dari posisi terakhir	64
Gambar 4. 7 Robot 2 berotasi sebesar 275°	65
Gambar 4. 8 Robot 1 maju 136 cm dari posisi awal.....	65
Gambar 4. 9 Robot 2 maju 120 cm dari posisi awal.....	66
Gambar 4. 10 Robot 1 mundur 294 cm dari posisi terakhir	66
Gambar 4. 11 Robot 2 mundur 160 cm dari posisi terakhir	67
Gambar 4. 12 Robot 1 berotasi sebesar 720°	67
Gambar 4. 13 Robot 2 berotasi sebesar 275°	68
Gambar 4. 14 Robot 1 maju 137 cm dari posisi awal	68

Gambar 4. 15 Robot 2 maju 130 cm dari posisi awal.....	69
Gambar 4. 16 mundur 286 cm dari posisi terakhir	69
Gambar 4. 17 Robot 2 mundur 156 cm dari posisi terakhir.....	70
Gambar 4. 18 Robot 1 berotasi sebesar 740°	70
Gambar 4. 19 Robot 2 berotasi sebesar 280°	71
Gambar 4. 20 (A) Robot Penyerang, (B) Robot Bertahan	82
Gambar 4. 21 Letak posisi <i>Base Station</i> yang menjadi titik awal pengukuran	84



DAFTAR TABEL

Tabel 2. 1 Kategori Degredasi.....	19
Tabel 2. 2 Standarisasi ITU-T delay	21
Tabel 4. 1 Hasil Perhitungan <i>Packet Loss</i> (dalam %) Babak Pertama	49
Tabel 4. 2 Hasil Perhitungan <i>Packet Loss</i> (dalam %) Babak Kedua.....	50
Tabel 4. 3 Hasil Perhitungan <i>Packet Loss</i> (dalam %) Babak Ketiga.....	51
Tabel 4. 4 Kategori Degredasi Standarisasi ITU.....	53
Tabel 4. 5 Hasil Perhitungan <i>Delay</i> Babak Pertama.....	55
Tabel 4. 6 Hasil Perhitungan <i>Delay</i> Babak Kedua	56
Tabel 4. 7 Hasil Perhitungan <i>Delay</i> Babak Ketiga	57
Tabel 4. 8 Standarisasi ITU-T Delay	59
Tabel 4. 9 Reaksi robot dengan protokol TCP	72
Tabel 4. 10 Reaksi robot dengan protokol UDP Sinkronisasi.....	73
Tabel 4. 11 Reaksi robot dengan protokol UDP tanpa sinkronisasi	74
Tabel 4. 12 Reaksi robot dengan protokol TCP	75
Tabel 4. 13 Reaksi robot dengan protokol UDP dengan Sinkronisasi.....	76
Tabel 4. 14 Reaksi robot dengan protokol UDP tanpa sinkronisasi	77
Tabel 4. 15 Reaksi robot dengan protokol TCP	78
Tabel 4. 16 Reaksi robot dengan protokol UDP dengan Sinkronisasi.....	79
Tabel 4. 17 Reaksi robot dengan protokol UDP tanpa sinkronisasi	80
Tabel 4. 18 Jarak Jangkauan Media <i>Receiver</i> dengan Menggunakan TCP	85
Tabel 4. 19 Jarak Jangkauan Media <i>Receiver</i> dengan Menggunakan UDP Sinkronisasi	86
Tabel 4. 20 Jarak Jangkauan Media <i>Receiver</i> dengan Menggunakan UDP	87

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Berbagai kompetisi robot sepak bola telah banyak di gelar diberbagai negara, salah satunya di Indonesia. Terdapat dua jenis kompetisi yang diselenggarakan di Indonesia yaitu, robot sepak bola humanoid dan robot sepak bola beroda. Untuk pertandingan sepak bola beroda masih dikategorikan kompetisi baru karena baru diadakan dua kali yaitu pada tahun 2017 dan 2018, kompetisi robot sepak bola beroda berada dalam kategori Kontes Robot Sepak Bola – Beroda atau biasa disingkat dengan KRSBI-B (Kementerian Riset, 2018). Robot sepak bola beroda adalah salah satu bidang robotika yang mengkombinasikan kecerdasan buatan, karena bidang ini mempelajari bagaimana robot dapat dibuat dan dilatih untuk memainkan permainan sepak bola (Rachman, 2017). Bagian utama dari bidang ini adalah navigasi robot, menghindari rintangan, dan komunikasi baik antar robot maupun dari basestation ke robot.

Kampus Institut Bisnis dan Informatika Stikom Surabaya sendiri telah dua kali mengikuti ajang kompetisi ini yaitu pada tahun 2017 dan 2018. Pada tahun pertama robot sepak bola yang diikutsertakan dalam ajang kompetisi KRSBI-B hanya mampu mendekripsi warna bola dan juga menendang bola. Hal ini masih belum memenuhi kriteria persyaratan dari *rule competition* karena robot masih belum dapat berkomunikasi baik antar robot maupun menerima perintah dari

sebuah aplikasi dari wasit atau lebih dikenal dengan *Refree Box*, dan juga pada tahun pertama tidak adanya *Base Station* sebagai media untuk menerima setiap perintah dari *Refree Box* yang kemudian diteruskan kepada robot.

Pada tahun kedua robot sepak bola Institut Bisnis dan Informatika Stikom Surabaya mengalami kemajuan dalam hal mendeteksi bola yang lebih akurat serta telah dapat berkomunikasi dan menerima perintah yang dikirimkan dari *Refree Box* kepada robot melalui *Base Station*. Pada tahun kedua ini tim telah membuat sebuah aplikasi *Base Station* yang bertugas sebagai media komunikasi antara *Refree Box* dan robot. Dengan menggunakan socket programming komunikasi dapat dibangun dan dapat mengirimkan data dari *Base Station* menuju robot, dimana pada robot terdapat modul wifi sebagai media receiver yang terkoneksi pada akses point. Komunikasi yang terjalin antara *Base Station* dan robot berjalan diatas protokol TCP, hal ini membuat pengiriman data dari *Base Station* menuju robot terasa lambat karena TCP menggunakan aliran dan mekanisme error control pada level transportasi, dimana keakuratan data yang dikirim sangatlah penting. Sesuai dengan karakteristik TCP yang kompleks pada layer transportnya (Anggraeni, 2012).

Berdasarkan permasalahan yang muncul dan telah terjadi pada saat kompetisi, maka penggunaan protokol TCP kurang begitu efektif untuk diterapkan pada proses pengiriman data antara *Base Station* dan robot karena kecepatan pengiriman data lebih dibutuhkan dari pada keakurasaian data. Menurut (Santoso, Nugroho, & Wardana, 2018) dengan menggunakan metode UDP *Multicast* proses pengiriman data dapat terjadi secara cepat dan lancar, karena robot dan *Base Station* mengirim pesan terus menerus setiap 100 mili detik tanpa ada

acknowledge-nya dan data tersebut dapat terbaca oleh semua anggota *multicast group*. Namun karena pada proses pengiriman data tersebut tidak ada *acknowledge-nya* maka kemungkinan besar dapat terjadi tabrakan pada saat pengiriman data dan hal tersebut dapat mengakibatkan *packet loss*, sehingga perlu ada sinkronisasi pengiriman data agar pendistribusian dapat berjalan dengan efisien. Untuk itulah penulis ingin melakukan penelitian lebih lanjut mengenai sinkronisasi pengiriman data antara robot dan *Base Station* pada UDP *Multicast*, hasil dari pengiriman data tersebut kemudian akan diuji langsung kepada robot dan dianalisis melalui tiga parameter pengujian yaitu *packet loss*, *delay*, dan reaksi robot bergerak dengan menggunakan hardware yang sama. Selain pengujian terhadap protokol UDP dengan sinkronisasi, evaluasi juga akan dilakukan terhadap komunikasi yang berjalan diatas protokol TCP/IP dan UDP tanpa sinkronisasi. Kemudian hasil dari pengujian tersebut akan dibandingkan untuk mendapatkan hasil pengujian efisiensi distribusi komunikasi data antara robot dan *Base Station*.

1.2 Rumusan Masalah

Dari permasalahan yang diuraikan diatas, dapat dirumuskan permasalahan yang dihadapi yaitu sebagai berikut :

1. Bagaimana melakukan sinkronisasi protokol UDP untuk komunikasi antara robot dan Base station pada robot sepak bola beroda.
2. Bagaimana menguji efisiensi distribusi komunikasi data antara robot dan Base Station dengan menggunakan parameter uji *delay*, *packet loss*, dan reaksi robot bergerak.

1.3 Batasan Masalah

Dalam melakukan penelitian sinkronisasi ini terdapat beberapa batasan masalah, diantaranya adalah :

1. Komunikasi yang terjalin antara Robot dan Base Station berjalan diatas protokol UDP.
2. Menggunakan *software* visual studio dalam mendesain GUI aplikasi Base Station.
3. Menggunakan mikrokontroler wemos pada masing-masing robot sebagai modul wifi yang digunakan untuk media receiver.
4. Pengujian dilakukan dengan sekenario pertandingan KRSBI-Beroda.
5. Menggunakan *software* Wireshark sebagai aplikasi untuk mengukur *packet loss* dan *delay*.

1.4 Tujuan

Tujuan dari dilakukannya proses sinkronisasi ini adalah sebagai berikut :

1. Proses pengiriman data yang terjadi dapat dilakukan secara efektif dan efisien.
2. Mengurangi resiko terjadinya *packet loss* saat pengiriman data sehingga komunikasi yang terjalin antara robot dengan Base Station dapat berjalan dengan lancar.

1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini secara Garis besar tersusun dari 5 (lima) bab, yaitu diuraikan sebagai berikut:

1. BAB I PENDAHULUAN

Pada Bab ini dibahas mengenai latar belakang masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

2. BAB II LANDASAN TEORI

Pada Bab ini dibahas teori penunjang dari permasalahan, yaitu mengenai *User Datagram Protocol*, metode *Multicast*, *Refree Box*, *Base Station*, *Wireshark*, perhitungan *Packet Loss dan delay*, aplikasi GUI Visual Studio, dan Robot.

3. BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada Bab ini dibahas tentang rancangan penelitian, blok diagram sistem komunikasi, perancangan sistem yang akan dibuat, blok diagram sinkronisasi antara *base station* dan robot, serta flowchart algoritma dari reaksi robot yang mengeksekusi perintah *Refree box*.

4. BAB IV HASIL DAN PEMBAHASAN

Pada Bab ini dibahas mengenai pengujian terhadap proses sinkronisasi yang terjadi antara robot dan *base station*. Kemudian hasil data yang didapat akan dibandingkan dengan proses komunikasi yang terjadi diatas protokol UDP tanpa sinkronisasi dan proses komunikasi yang terjadi diatas protokol TCP/IP. Dari perbandingan tersebut akan dianalisis, proses komunikasi mana yang lebih efisien dan efektif untuk diterapkan pada komunikasi KRSBI-Beroda.

5. BAB V PENUTUP

Berisi kesimpulan dan saran yang didapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 *User Datagram Protocol (UDP)*

Menurut (Dwiyankuntoko, 2009) UDP merupakan *protocol* internet yang mengutamakan kecepatan data. *Protocol* ini biasanya digunakan untuk *streaming* video ataupun fasilitas *real-time* yang lain. Oleh karena itu pada UDP ini tidak memerlukan adanya *setup* koneksi terlebih dahulu karena hal tersebut dapat menyebabkan adanya tambahan *delay*. Selain itu, *protocol* ini termasuk dalam *protocol* yang sederhana, artinya antara penerima dan pengirim tidak perlu menjaga *session* atau status koneksi, ukuran *headernya* juga sederhana. UDP ini juga tidak memerlukan *congestion control* (control kemacetan) pada koneksiya. Maksudnya adalah UDP dapat mengirimkan per segment tanpa dipengaruhi oleh kesibukan jaringan (Jusak, 2016).

Karateristik dari protokol UDP sendiri diantaranya adalah *connectionless* dimana tidak ada aktifitas handshaking antara UDP dan penerimanya saat akan dilakukan pengiriman data, sehingga data tersebut dikirim melalui jaringan dan mencapai ke komputer tujuan tanpa membuat suatu koneksi langsung. *Unreliable* dimana pesan-pesan yang dikirim menggunakan protokol ini akan dikirim sebagai data gram tanpa adanya nomor urut atau pesan *Acknowledgment* (ACK), hal tersebut menyebabkan pesan-pesan yang diterima di tujuan mungkin saja diterima dalam keadaan yang tidak urut sehingga protokol yang berjalan diatasnya (layer aplikasi) berperan penting dalam memulihkan pesan-pesan tersebut.

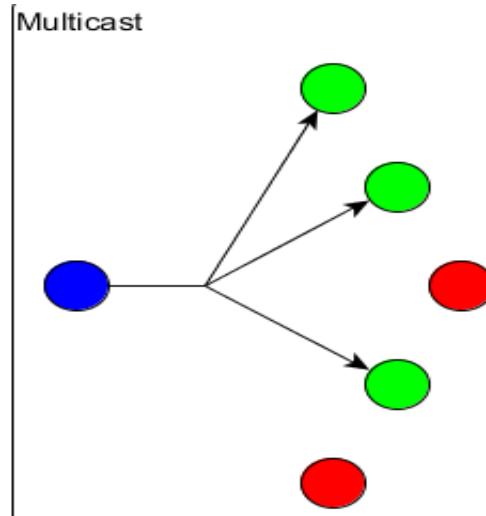
UDP Datagram Header Format								
Bit #	0	7	8	15	16	23	24	31
0	Source Port			Destination Port				
32	Length			Header and Data Checksum				

Gambar 2. 1 User Datagram Format (Microchip, 2018)

2.2 *Multicast*

Menurut (Sugeng, 2010) alamat IP *Multicast* (*Multicast IP Address*) adalah alamat yang digunakan untuk menyampaikan satu paket kepada banyak penerima. Dalam sebuah intranet yang memiliki alamat *multicast* IPv4, sebuah paket yang ditujukan ke sebuah alamat *multicast* akan diteruskan oleh *router* ke subjaringan di mana terdapat host-host yang sedang berada dalam kondisi "listening" terhadap lalu lintas jaringan yang dikirimkan ke alamat *multicast* tersebut. Dengan cara ini, alamat *multicast* pun menjadi cara yang efisien untuk mengirimkan paket data dari satu sumber ke beberapa tujuan untuk beberapa jenis komunikasi. Alamat *multicast* didefinisikan dalam RFC 1112.

Alamat-alamat *multicast* IPv4 didefinisikan dalam ruang alamat kelas D, yakni 224.0.0.0/4, yang berkisar dari 224.0.0.0 hingga 239.255.255.255. *Prefiks* alamat 224.0.0.0/24 (dari alamat 224.0.0.0 hingga 224.0.0.255) tidak dapat digunakan karena dicadangkan untuk digunakan oleh lalu lintas *multicast* dalam subnet lokal.



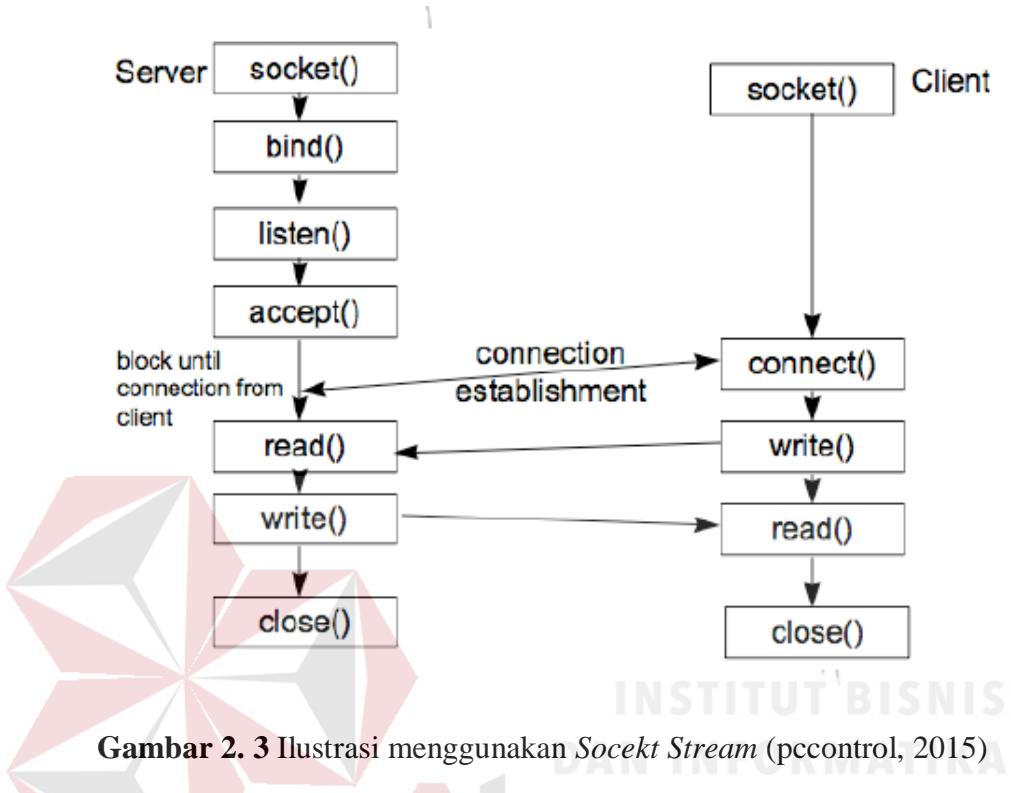
Gambar 2. 2 One-to-Many Multicast (Nurwinda, 2017)

2.3 *Socket Programming*

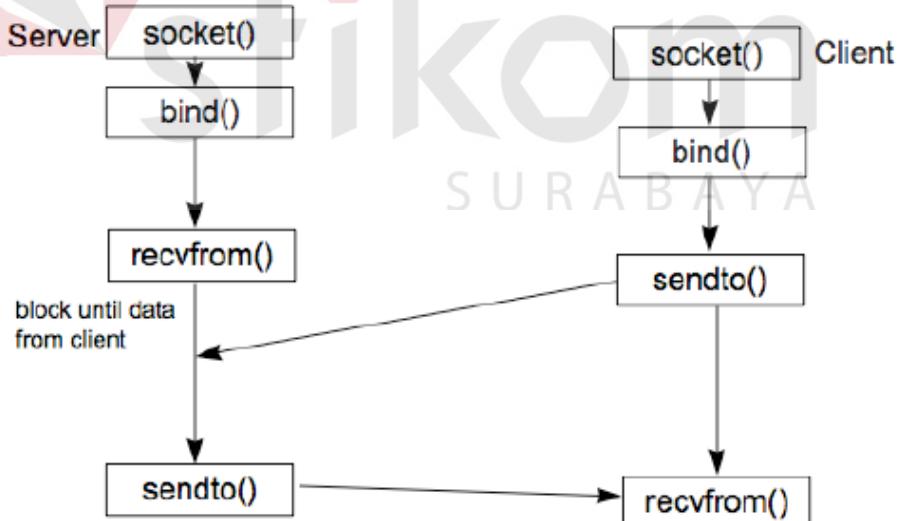
Menurut (Edwin, 2012) *socket* adalah mekanisme komunikasi yang memungkinkan terjadinya data antara program atau proses baik dalam satu mesin maupun antar mesin. Ada dua jenis *socket* yang paling umum digunakan yaitu, jenis pertama adalah *socket* Lokal atau AF_UNIX dimana *socket* ini digunakan untuk melakukan komunikasi dengan perantara sebuah *file* yang biasanya diletakan pada direktori /tmp dan lainnya. Jenis kedua adalah *socket networking* dimana *socket* ini digunakan untuk komunikasi antar aplikasi antar mesin dalam lingkungan jaringan TCP/IP maupun UDP.

Socket networking sendiri dikategorikan menjadi dua kategori berdasarkan pengiriman datanya yaitu, kategori pertama adalah *Socket Stream* yang merupakan *socket* untuk komunikasi *full-duplex* berbasis aliran (stream) data, *socket* ini dikhkususkan untuk digunakan pada protokol TCP/IP. Kategori kedua adalah *socket Datagram* dimana *socket* ini tidak membutuhkan koneksi yang tersambung dengan benar untuk mengirimkan dan menerima data. *Socket*

model ini dikhkususkan untuk digunakan pada protokol UDP. Ilustrasi *socket stream* dan *socket datagram* dapat dilihat pada gambar 2.3 dan 2.4 berikut ini.



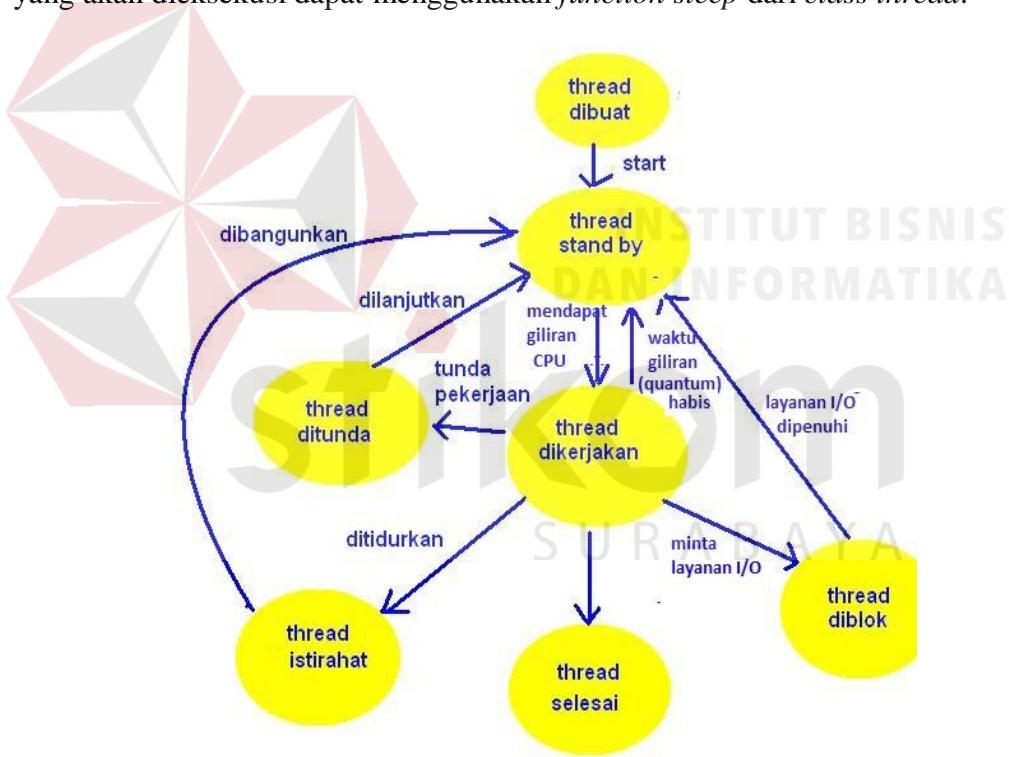
Gambar 2. 3 Ilustrasi menggunakan *Socekt Stream* (pccontrol, 2015)



Gambar 2. 4 Ilustrasi menggunakan *Socekt Datagram* (pccontrol, 2015)

2.4 Thread

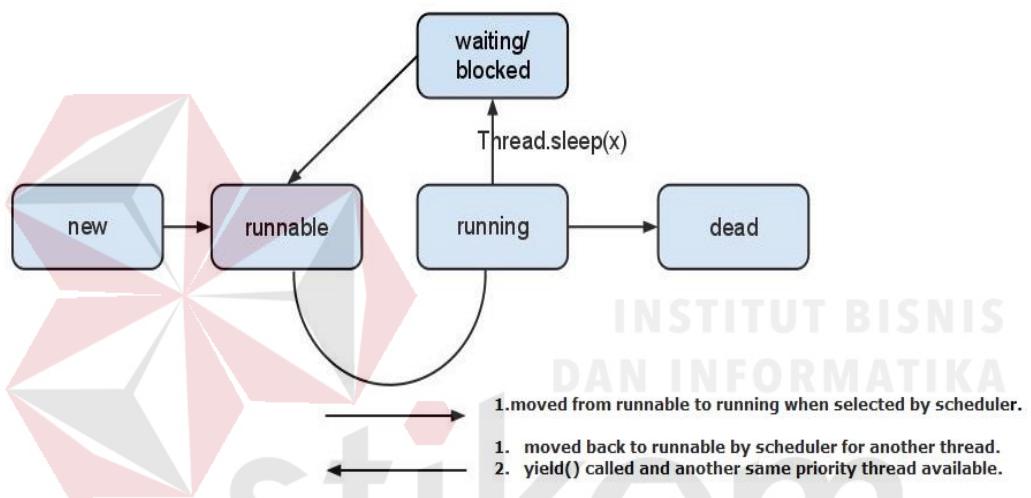
Thread adalah sekumpulan perintah (instruksi) yang dapat dilaksanakan (dieksekusi) secara sejajar dengan *thread* lainnya, dengan menggunakan cara *time slice* dimana ketika satu CPU melakukan perpindahan antara satu *thread* ke *thread* lainnya atau *multiprocess* dimana ketika *thread* tersebut dilaksanakan oleh CPU yang berbeda dalam satu sistem (Wikipedia, 2016). *Thread* sebenarnya sama dengan proses, hanya saja cara berbagai sumber daya antara proses dengan *thread* sangat berbeda. Untuk mengatur jeda waktu terhadap sebuah instruksi yang akan dieksekusi dapat menggunakan *function sleep* dari *class thread*.



Gambar 2. 5 Ilustrasi Cara Kerja Thread (pccontrol, 2013)

Dalam *class Thread* terdapat *function sleep* yang digunakan untuk menjeda sistem yang berjalan saat ini untuk jumlah waktu yang telah ditentukan. *Function thread sleep* memiliki dua macam tipe *function* yaitu *sleep(Int32)* berfungsi untuk menjeda sistem yang berjalan saat ini untuk jumlah milidetik

yang telah ditentukan, cara kerjanya sendiri adalah jika nilai dari *milliseconds timeout* adalah not, maka *threads* melepaskan sisa waktunya untuk setiap *threads* dengan prioritas yang sama dan siap disajikan, jika tidak ada *threads* lain dengan prioritas sama yang siap dijalankan dan eksekusi *threads* saat ini tidak ditangguhkan. Yang kedua adalah *sleep(TimeSpan)* berfungsi untuk menjeda sistem yang berjalan saat ini untuk jumlah detik yang telah ditentukan. Ilustrasi *thread* dapat dilihat pada gambar 2.5 dibawah ini.

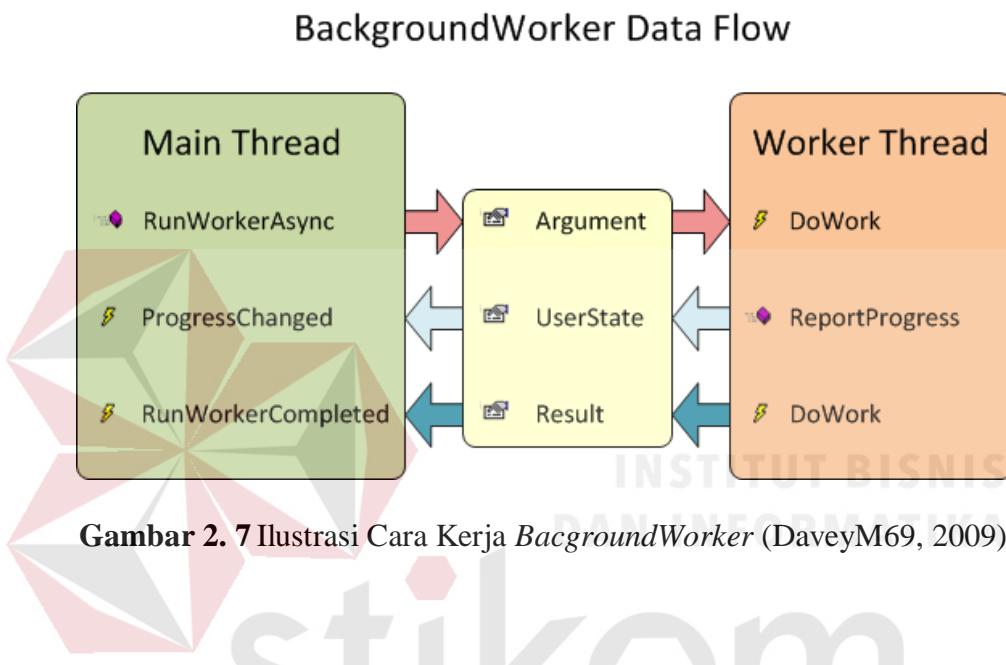


Gambar 2. 6 Ilustrasi Cara Kerja *Thread Sleep* (stackoverflow, 2011)

2.5 *BackgroundWorker*

Backgroundworker adalah salah satu *class* yang terdapat pada *namespace System.ComponentModel* yang digunakan sebagai obyek, menurut (Herdinanto, 2015) *backgroundworker* adalah komponen yang dapat menjalankan atau mengeksekusi kode pada *background* atau *secondary thread*. Dengan menggunakan *backgroundworker* sebuah sistem atau aplikasi yang dijalankan bersamaan dapat berjalan secara *multitasking* tanpa mengganggu antara satu dengan lainnya. Salah satu penggunaan *backgroundworker* adalah ketika

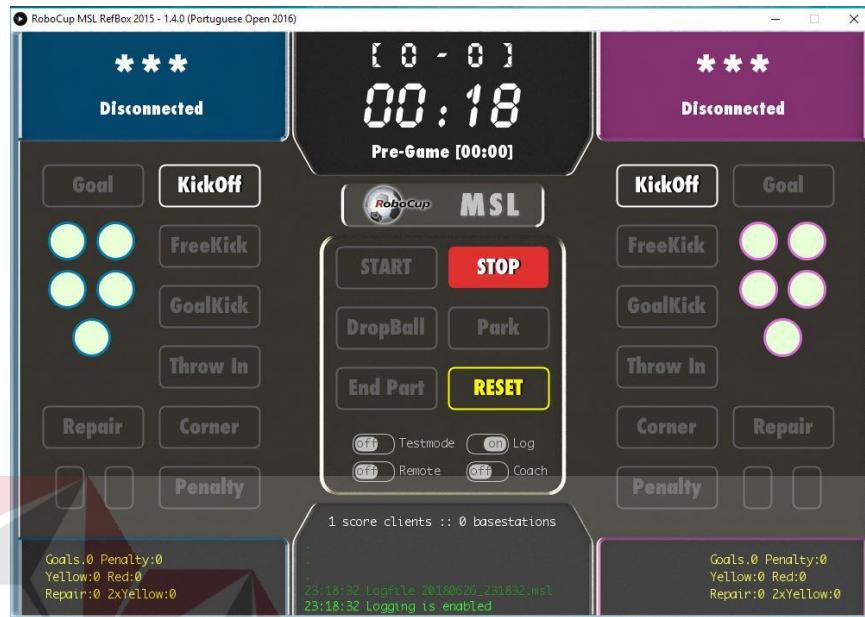
mendownload sebuah *file* atau menyalin *file* yang mempunyai ukuran besar dan membutuhkan waktu yang lama. *Backgroundworker* juga dapat menjalankan maupun mengeksekusi kode tanpa mengganggu proses pada program utama. Berikut adalah ilustrasi *backgroundworker* yang ditunjukan gambar 2.7 dibawah ini.



2.6 Refree Box

RefBox atau *Refree Box* adalah sebuah perangkat lunak untuk mengatur dan mengendalikan pertandingan. *RefBox* dioperasikan oleh wasit. *RefBox* 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 *RefBox* melalui *Base Station*. Robot beroperasi secara otomatis berdasarkan instruksi yang diterima dari *RefBox* (KRSBI-B, 2017). Format instruksi yang dikirim dari *RefBox* ke *Base Station* dalam bentuk

satu buah karakter (*char*), sudah sangat rinci berada pada *mailing list* dari panitia KRSBI Beroda.



Gambar 2. 8 Aplikasi *Refree Box* (Niemueller, 2015)

Terdapat tujuh instruksi utama yang digunakan untuk mengkontrol permainan kedua tim dan sepuluh instruksi yang digunakan untuk mengkontrol masing-masing tim. Berikut penjelasan dari masing-masing instruksi yang terdapat pada *Refree Box* :

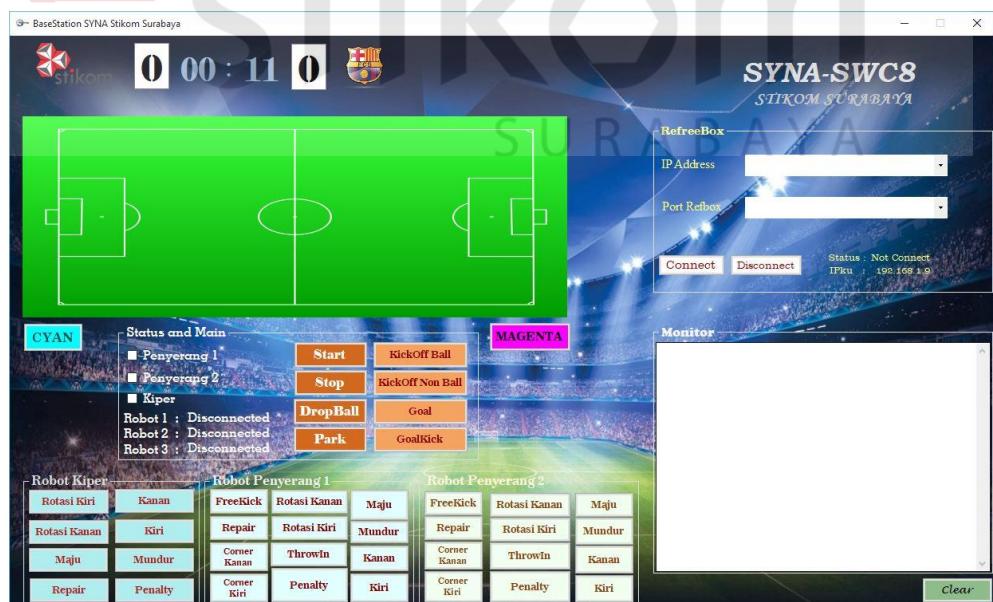
❖ Tujuh Instruksi Utama

- a. Start = s (digunakan untuk memulai pertandingan)
- b. Stop = S (digunakan untuk menghentikan pertandingan)
- c. Drop Ball = N (Ketika kedua robot tim dalam keadaan *dead lock*)
- d. Park = L (Ketika permainan telah berakhir)
- e. End Part = h (Ketika babak pertama telah usai)
- f. Reset = Z (Untuk mengatur ulang *Refree Box*)
- g. End Game = e (Ketika babak kedua telah usai)

- ❖ Sepuluh Instruksi Untuk Tim Cyan
 - a. Kick Off = K (Penempatan posisi awal)
 - b. Free Kick = F (Tendangan Bebas)
 - c. Goal Kick = G (Bola keluar melebihi gawang)
 - d. Throw In = T (Lemparan bola dalam)
 - e. Corner = C (Tendangan sudut)
 - f. Penalty = P (Tendangan dari titik putih)
 - g. Goal = A (Goal, kembali ke posisi awal)
 - h. Repair = O (Robot mengalami kerusakan dan harus diperbaiki)
 - i. Red = R (kartu merah, robot keluar)
 - j. Yellow = Y (kartu kuning, peringatan)
- ❖ Sepuluh Instruksi Untuk Tim Magenta
 - a. Kick Off = k (Penempatan posisi awal)
 - b. Free Kick = f (Tendangan Bebas)
 - c. Goal Kick = g (Bola keluar melebihi gawang)
 - d. Throw In = t (Lemparan bola dalam)
 - k. Corner = c (Tendangan sudut)
 - e. Penalty = p (Tendangan dari titik putih)
 - f. Goal = a (Goal, kembali ke posisi awal)
 - l. Repair = o (Robot mengalami kerusakan dan harus diperbaiki)
 - g. Red = r (kartu merah, robot keluar)
 - m. Yellow = y (kartu kuning, peringatan)

2.7 Base Station

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 *RefBox* (Ardhiansyah & Syarifuddin, 2017). *Base Station* melakukan koordinasi aktivitas setiap robotnya dalam pertandingan. Selama pertandingan sedang berjalan, anggota tim peserta tidak diizinkan untuk mengoperasikan *Base Station*-nya. *Base Station* menggunakan kabel *Ethernet Switch* untuk melakukan koneksi dengan *RefBox*. *Base Station* memanggil alamat IP (IP Address) *RefBox* untuk melakukan koneksi melalui *Port 28097 (default)*. Alamat IP Komputer yang menjalankan *RefBox* akan diumumkan pada saat pertandingan. Cara koneksi antara *Base Station* dengan robot yakni setiap robot memiliki alamat IP (IP address). *Base Station* dapat terhubung dengan robot secara *wireless* melalui *access point* yang khusus disediakan panitia dalam pertandingan (KRSBI-B, 2017).



Gambar 2. 9 Base Station Tim Institut dan Informatika Stikom Surabaya

Aplikasi *Base Station* tim robotik Stikom memiliki beberapa fungsi *button* diantaranya terdapat delapan instruksi utama dan tiga *check box connection*, delapan instruksi untuk robot kiper, dua belas instruksi untuk robot penyerang 1, dua belas instruksi untuk robot bertahan (penyerang 2), dua *combo box* dan dua instruksi *connect* dan *disconnect* pada *grup box Refree Box*, dua *button CYAN* dan *Magenta*, serta *button clear* pada *grup box monitor*. Berikut penjelasan dari setiap *button* pada *Base Station* :

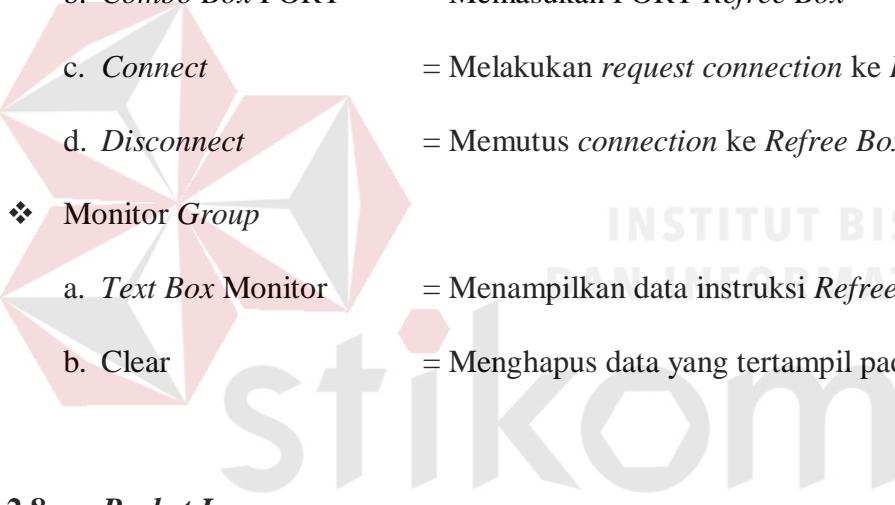
❖ *Status and Main*

- a. *Check Box Penyerang 1* = Melakukan *request connection* ke Robot 1
- b. *Check Box Penyerang 2* = Melakukan *request connection* ke Robot 2
- c. *Check Box Kiper* = Melakukan *request connection* ke Robot 3
- d. Start = s (Memerintahkan robot bergerak)
- e. Stop = S (Memerintahkan robot berhenti)
- f. Drop Ball = N (Memerintahkan robot berhenti)
- g. Park = L (Memerintahkan robot berhenti)
- h. Kick Off Ball = K (Memerintahkan robot ke posisi bola tengah)
- i. Kick Off Non Ball = k (Memerintahkan robot ke posisi tengah)
- j. Goal = A (Memerintahkan robot ke posisi tengah)
- k. Goal Kick = G (Memerintahkan robot berhenti)

❖ *Robot Kiper*

- a. Rotasi Kiri = r (Mengontrol robot berotasi kiri)
- b. Rotasi Kanan = R (Mengontrol robot berotasi kanan)
- c. Maju = M (Mengontrol robot bergerak maju)
- d. Repair = O (Memerintahkan robot berhenti)

- e. Kanan = Y (Mengontrol robot bergerak ke kanan)
- f. Kiri = y (Mengontrol robot bergerak ke kiri)
- g. Mundur = m (Mengontrol robot bergerak mundur)
- h. Penalty = P (Memerintahkan robot ke posisi gawang)
- ❖ Robot Penyerang 1
- a. Free Kick = F (Memerintah robot untuk tendangan bebas)
- b. Repair = O (Memerintahkan robot berhenti)
- c. Corner Kanan = C (Memerintahkan robot ke posisi sudut kanan)
- d. Corner Kiri = c (Memerintahkan robot ke posisi sudut kiri)
- e. Rotasi Kanan = R (Mengontrol robot berotasi kanan)
- f. Rotasi Kiri = r (Mengontrol robot berotasi kiri)
- g. Throw In = T (Memerintahkan robot mengambil bola out)
- h. Penalty = P (Memerintahkan robot ke posisi titik penalti)
- i. Maju = M (Mengontrol robot bergerak maju)
- j. Mundur = m (Mengontrol robot bergerak mundur)
- k. Kanan = Y (Mengontrol robot bergerak ke kanan)
- l. Kiri = y (Mengontrol robot bergerak ke kiri)
- ❖ Robot Penyerang 1
- a. Free Kick = F (Memerintah robot untuk tendangan bebas)
- b. Repair = O (Memerintahkan robot berhenti)
- c. Corner Kanan = C (Memerintahkan robot ke posisi sudut kanan)
- d. Corner Kiri = c (Memerintahkan robot ke posisi sudut kiri)
- e. Rotasi Kanan = R (Mengontrol robot berotasi kanan)
- f. Rotasi Kiri = r (Mengontrol robot berotasi kiri)

- 
- g. Throw In = T (Memerintahkan robot mengambil bola out)
 - h. Penalty = P (Memerintahkan robot ke posisi titik penalti)
 - i. Maju = M (Mengkontrol robot bergerak maju)
 - j. Mundur = m (Mengkontrol robot bergerak mundur)
 - k. Kanan = Y (Mengkontrol robot bergerak ke kanan)
 - l. Kiri = y (Mengkontrol robot bergerak ke kiri)
- ❖ *Refree Box Group*
- a. *Combo Box IP Address* = Memasukan IP *Refree Box*
 - b. *Combo Box PORT* = Memasukan PORT *Refree Box*
 - c. *Connect* = Melakukan *request connection* ke *Refree Box*
 - d. *Disconnect* = Memutus *connection* ke *Refree Box*
- ❖ *Monitor Group*
- a. *Text Box Monitor* = Menampilkan data instruksi *Refree Box*
 - b. *Clear* = Menghapus data yang tertampil pada monitor

2.8 *Packet Loss*

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena *retransmisi* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru

tidak akan diterima. Berikut penyebab terjadinya *packet loss* menurut (Oktavianto, 2014), yakni:

1. Congestion, disebabkan terjadinya antrian yang berlebihan dalam jaringan.
2. Node yang bekerja melebihi kapasitas buffer.
3. Memory yang terbatas pada node.
4. Policing atau kontrol terhadap jaringan untuk memastikan bahwa jumlah trafik yang mengalir sesuai dengan besarnya bandwidth. Jika besarnya trafik yang mengalir didalam jaringan melebihi dari kapasitas bandwidth yang ada maka policing control akan membuang kelebihan trafik yang ada.

Untuk menghitung *packet loss* digunakan rumus :

$$\text{Packet loss} = \left(\frac{\text{data yang dikirim} - \text{paket data yang diterima}}{\text{paket data yang dikirim}} \right) \times 100\%$$

Hasil dari perhitungan tersebut kemudian di kategorikan sebagai berikut :

Tabel 2. 1 Kategori Degredasi

KATEGORI DEGREDASI	PACKET LOSS
Sangat bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

2.9 *Delay*

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Oleh karenanya *delay* dalam suatu jaringan juga merupakan unjuk kerja yang dapat dijadikan acuan dalam menilai kemampuan dan kualitas pentransmisian data. Akibat dari *delay*, data yang diterima akan mengalami keterlambatan waktu datang sehingga hal ini menyebabkan terjadi penungguan sejenak data tersebut sampai pada tujuan. *Delay* akan sangat dirasakan ketika melakukan transmisi paket data yang bersifat UDP atau secara *realtime*. Sebagai contoh ketika menghubungi seseorang dari Surabaya yang ada di tempat sangat jauh jaraknya, di luar negri melalui VoIP misalkan, sering menjumpai *delay* suara yang cukup terlambat datang untuk merespon suara dari tempat lain. Ada beberapa macam *delay* di dalam jaringan, di antaranya paketisasi *delay*, *queuing delay*, propagasi *delay*, (Firmansyah, 2008).

Adapun rumus yang digunakan untuk menghitung *delay* adalah sebagai berikut :

$$\text{Delay} = \text{Waktu paket diterima} - \text{Waktu paket dikirimkan}$$

Untuk mencari rata-rata *delay* maka digunakan rumus :

$$\text{Rata-Rata } \textit{Delay} = \text{Total } \textit{Delay} / \text{Total paket yang diterima}$$

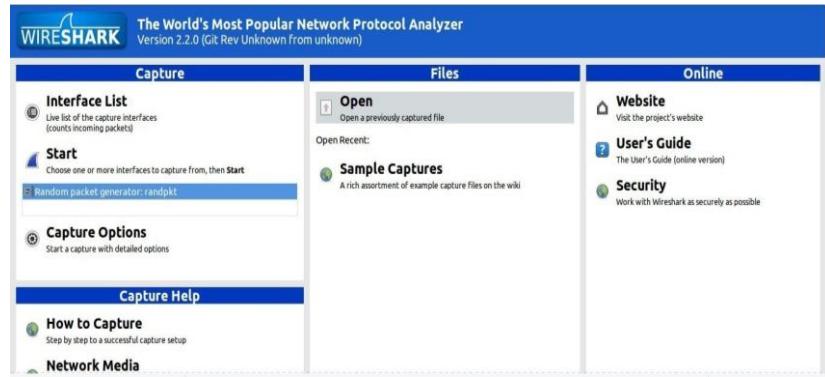
Terdapat standarisasi penilaian terhadap pengujian pada *delay* :

Tabel 2. 2 Standarisasi ITU-T delay

Kategori Delay	Besar Delay
Excellent	< 150 ms
Good	150 - 300 ms
Poor	300 - 450 ms
Unnaceptable	> 450 ms

2.10 Wireshark

Wireshark merupakan salah satu dari sekian banyak *tool Network Analyzer* yang banyak digunakan oleh *Network administrator* untuk menganalisis kinerja jaringannya termasuk protokol didalamnya. *Wireshark* banyak disukai karena *interface*-nya yang menggunakan *Graphical User Interface (GUI)* atau tampilan grafis. *Wireshark* mampu menangkap paket-paket data atau informasi yang melintas dalam jaringan. Semua jenis paket informasi dalam berbagai format protokol pun akan dengan mudah ditangkap dan dianalisis. Karenanya tak jarang *tool* ini juga dapat dipakai untuk *sniffing* (memperoleh informasi penting spt *password email* atau *account* lain) dengan menangkap paket-paket yang melintas di dalam jaringan dan menganalisisnya, (Phyong, 2010). Secara garis besar cara kerja *wireshark* terdiri dari dua tahapan, pertama merekam semua paket yang melewati *interface* yang dipilih (*interface* adalah perangkat penghubung antar jaringan, bisa melalui *wifi* atau *ethernet/lan card*). Kedua hasil rekaman tadi dapat dianalisis dengan memfilter protokol yang ingin di analisis seperti *tcp*, *http*, *udp*, dan sebagainya. *Wireshark* juga dapat mencatat *cookie*, *post*, dan *reque*.



Gambar 2. 10 Wireshark (Nestor, 2017)

2.11 Visual Studio 2013

Microsoft Visual Studio merupakan *integrated development environment* (IDE) dari Microsoft. IDE ini dapat digunakan untuk mengembangkan aplikasi antarmuka pengguna konsol dan grafis beserta aplikasi Windows Forms, situs web, aplikasi web dan layanan web di kedua kode asli beserta kode terkelola untuk semua *platform* yang didukung oleh Microsoft Windows, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework dan Microsoft Silverlight (Rachman, 2017). Berikut logo dari visual studio dapat dilihat pada gambar 2.10.



Gambar 2. 11 Logo Visual Studio (Anderson, 2013)

2.12 Robot KRSBI Beroda Stikom Surabaya 2018

Robot KRSBI Beroda merupakan robot yang didesain untuk dapat bermain sepak bola dengan memanfaatkan motor dan roda sebagai aktuatornya. Robot dengan ukuran 52x52x60 cm ini memiliki fitur sistem gerak menggunakan roda Omni-Directional Wheels, yaitu suatu roda unik karena memiliki kemampuan bergerak bebas dua arah. Terpasang pada motor DC yang dikendalikan oleh *driver* motor EMS (Embedded Module Series) 30 A *H-Bridge* melalui *Arduino Due* (Susianto, 2018). Pada robot ini juga terdapat mekanisme penendang dengan tenaga udara. Melalui sebuah Pneumatik dengan arus listrik DC yang di kendalikan oleh sebuah *relay*. Berikut gambar robot dari Tim SYNA-SWC8.



Gambar 2. 12 Robot Tim SYNA-SWC8

BAB III

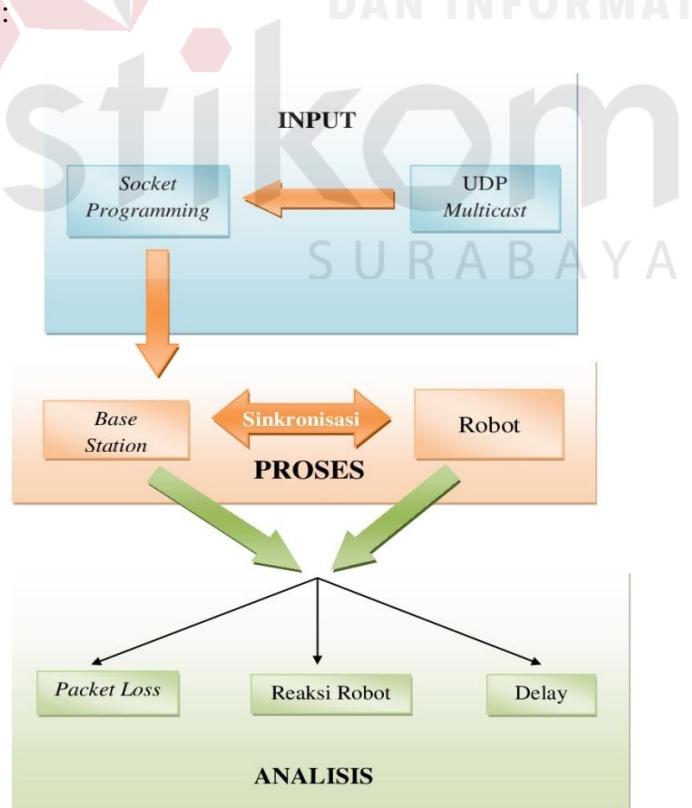
METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Berikut akan dijelaskan tentang tahapan rancangan dari penelitian ini yaitu, blok diagram sistem komunikasi, perancangan sistem komunikasi dan blok diagram sinkronisasi.

3.2 Blok Diagram Sistem Komunikasi

Secara umum perancangan perangkat lunak terdiri dari metode yang akan digunakan yaitu UDP *multicast* untuk menjadi dasar pembuatan aplikasi *Base Station* dan *tools* penerima pada *wemos*. Perancangan Blok Diagram dapat dilihat pada gambar 3.1:



Gambar 3. 1 Blok Diagram Rancangan Pengujian Komunikasi

Tiap-tiap dari diagram blok sistem pada gambar diatas dapat dijelaskan sebagai berikut :

1. Input pada *Socket Programming* :

- *UDP Multicast* :
 - a. Protokol yang berjalan pada *socket programming* untuk pengiriman data dari *Base Station* menuju robot.

2. Proses :

- Sinkronisasi :
 - a. Proses yang mengatur pengiriman data antar robot penyerang dan bertahan yang dilakukan oleh *Base Station* sebagai *server*.

▪ *Base Station* :

- a. Aplikasi yang dibuat dengan menggunakan visual studio dimana di dalamnya terdapat algoritma *socket programming* yang digunakan sebagai jembatan komunikasi antara *Refree Box* dengan *Base Station* dan *Base Station* dengan robot.

- b. Merupakan *client* yang menerima data dari *Refree Box*.

▪ Robot :

- a. Menerima data dari *Base Station* untuk mengeksekusi perintah dari *Refree Box* secara otomatis.
- b. Melakukan koordinasi dengan menerima dan mengirimkan data kepada *Base Station*.

3. Analisis :

- *Packet loss* : Berapa banyak paket data yang akan hilang pada saat proses pengiriman yang terjadi pada robot dan *Base Station*

- Reaksi robot bergerak : Bagaimana reaksi robot setelah menerima data dari *Base Station* dengan tolak ukur benar atau salah, robot mengeksekusi instruksi atau pun data yang dikirim oleh *Base Station*.
- *Delay* : Berapa besar *delay* yang dihasilkan dari pengiriman data yang terjadi.

3.3 Perancangan Sistem

Alur komunikasi yang terjadi sesuai dengan peraturan yang telah ditetapkan oleh buku panduan KRSBI Beroda, dimana *Base Station* dihubungkan dengan *Refree Box* melalui *switch*. *Base Station* akan bertindak sebagai *client* yang akan terus menerus menerima data yang dikirimkan dari *Refree Box* yang bertindak sebagai *server*. Selanjutnya data yang diterima oleh *Base Station* akan dikirimkan kepada 3 robot yang berada di lapangan untuk mengeksekusi perintah tersebut. Robot menerima data dari *Base Station* melalui Akses Point yang telah disediakan oleh juri (KRSBI-B, 2017).



Gambar 3. 2 Alur Komunikasi Data Robot Tim Institut dan Informatika Stikom Surabaya

Dari gambar diatas dapat dilihat bahwa pengiriman data dari *Refree Box* menuju *Base Station* melalui *switch* yang dihubungkan dengan kabel ethernet. Sedangkan data dari *Base Station* yang akan dikirim kepada robot melalui aksespoint, untuk itu lah harus terdapat sebuah modul wifi pada robot yang terhubung dengan akses point. Sinkronisasi dilakukan oleh *Base Station* sebagai server dari ketiga robot, *Base Station* akan mengatur proses pengiriman data antara *Base Station* dengan robot dan robot dengan robot. *Base Station* mengatur reaksi robot bergerak pengiriman dan penerimaan data, sehingga data yang dikirim melalui protokol UDP *Multicast* dapat berjalan secara efisien dan mengurangi resiko terjadinya tabrakan data saat pengiriman berlangsung. Komponen-komponen yang akan digunakan untuk membangun perancangan sistem komunikasi ini meliputi 4 komponen yaitu konfigurasi *multicast*, konfigurasi ip *address*, perancangan *Base Station*, dan diagram waktu.

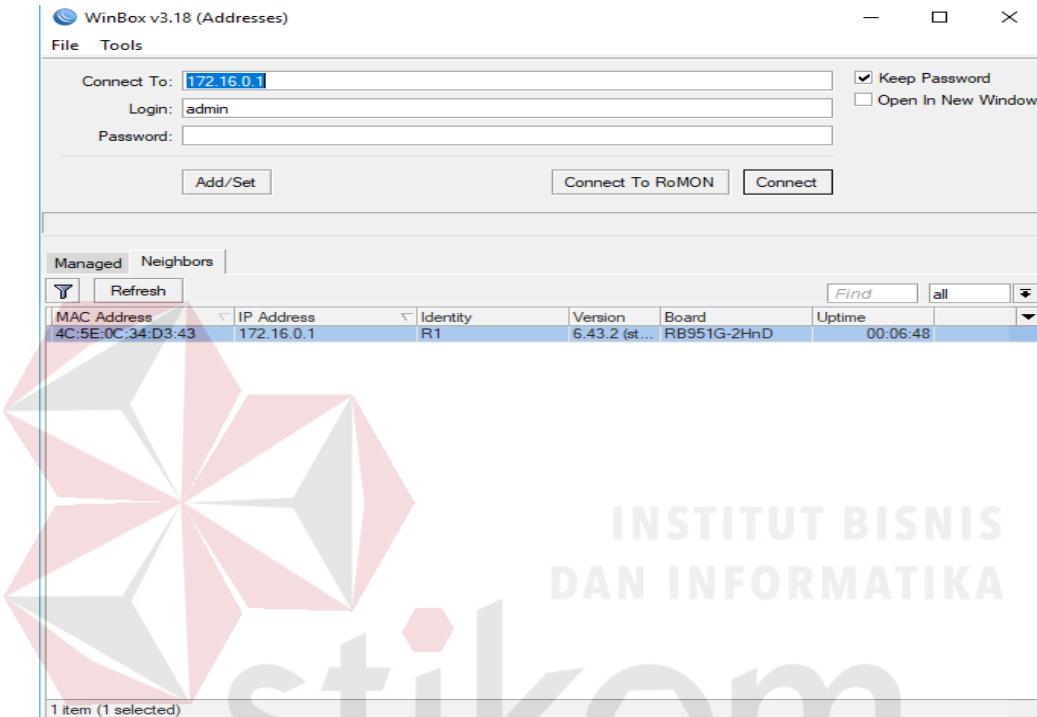
3.3.1 Konfigurasi *Mulicast*

Salah satu metode yang digunakan dalam pengujian Tugas Akhir ini adalah dengan menggunakan metode *Multicast* sebagai pengiriman data antara *Base Station* dengan robot. Tujuan dari penerapan metode *Multicast* ini adalah untuk mengelompokan *client-client* menjadi satu grup dengan server sehingga pada saat server mengirimkan sebuah *broadcast* maka hanya *client* yang masuk ke dalam grup tersebut yang menerimanya. Untuk dapat membangun *Multicast* pada sebuah koneksi maka perlu dilakukan konfigurasi pada akses point (Router). Dalam perancangan Tugas Akhir ini penulis menggunakan Router Mikrotik sebagai media akses point. Dan untuk mengkonfigurasi *Multicast* di Router Mikrotik, penulis menggunakan aplikasi Winbox yang mendukung proses

konfigurasi dari Router Mikrotik. Berikut tahap-tahap proses konfigurasi *Multicast* pada Router Mikrotik dengan menggunakan Winbox :

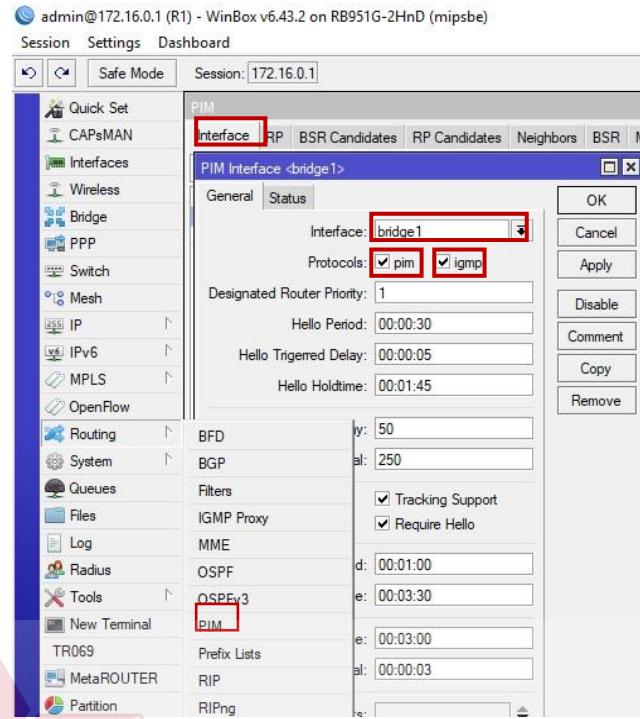
1. Buka aplikasi Winbox pilih ip server Mikrotik, kemudian tekan *Connect*.

Seperti yang terlihat pada gambar 3.3 berikut ini.



Gambar 3. 3 Tampilan Utama Winbox

2. Setelah terkoneksi dengan router maka muncul menu-menu konfigurasi pada Winbox. Langkah selanjutnya pilih menu *Routing* → *PIM* → *Interfaces* → *add* dan mencentang protokol pim dan igmp. Penulis menggunakan bridge1 sebagai interface PIM. Seperti yang terlihat pada gambar 3.4 berikut ini.



Gambar 3. 4 Konfigurasi Multicast pada Interface Bridge1

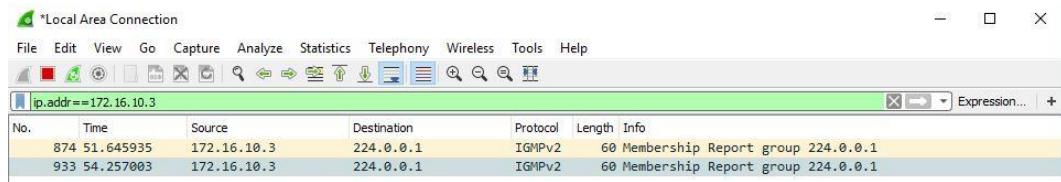
3. Setelah terkonfigurasi maka jaringan *Multicast* kini telah terbangun, dan untuk dapat melihat *client* mana saja yang bergabung pada jaringan *Multicast* dapat dilihat pada menu *IGMP grups*. Seperti yang terlihat pada gambar 3.5 berikut ini.

PIM						
RP	BSR Candidates	RP Candidates	Neighbors	BSR	MRIB	MFC
IGMP Groups						
<input type="button" value="Find"/>	<input type="button" value="+"/>	<input type="button" value="-"/>	<input type="button" value="Filter"/>			
Interface	contains	bridge1				
Interface	/	Group	State	Version	Source	Timeout ...
bridge1		224.0.0.13	exclude	IGMPv2		177
bridge1		239.255.255.250	exclude	IGMPv2		178
bridge1		224.0.0.22	exclude	IGMPv2		184
bridge1		224.0.0.1	exclude	IGMPv2		147
bridge1		224.0.0.2	exclude	IGMPv2		187

5 items

Gambar 3. 5 Tampilan IGMP Groups dari Interface bridge1.

4. Untuk melihat apakah robot penyerang dan robot bertahan telah bergabung ke dalam *Multicast Group* maka dapat dilihat melalui aplikasi *Wireshark* dengan mengkoneksikan terlebih dahulu antara *Base Station* dengan robot 1 (penyerang) maupun antara *Base Station* dengan robot 2 (bertahan). Seperti yang terlihat pada gambar 3.6 dan 3.7 berikut ini.



No.	Time	Source	Destination	Protocol	Length	Info
874	51.645935	172.16.10.3	224.0.0.1	IGMPv2	60	Membership Report group 224.0.0.1
933	54.257003	172.16.10.3	224.0.0.1	IGMPv2	60	Membership Report group 224.0.0.1

Gambar 3. 6 Robot Penyerang telah bergabung kedalam Multicast Group



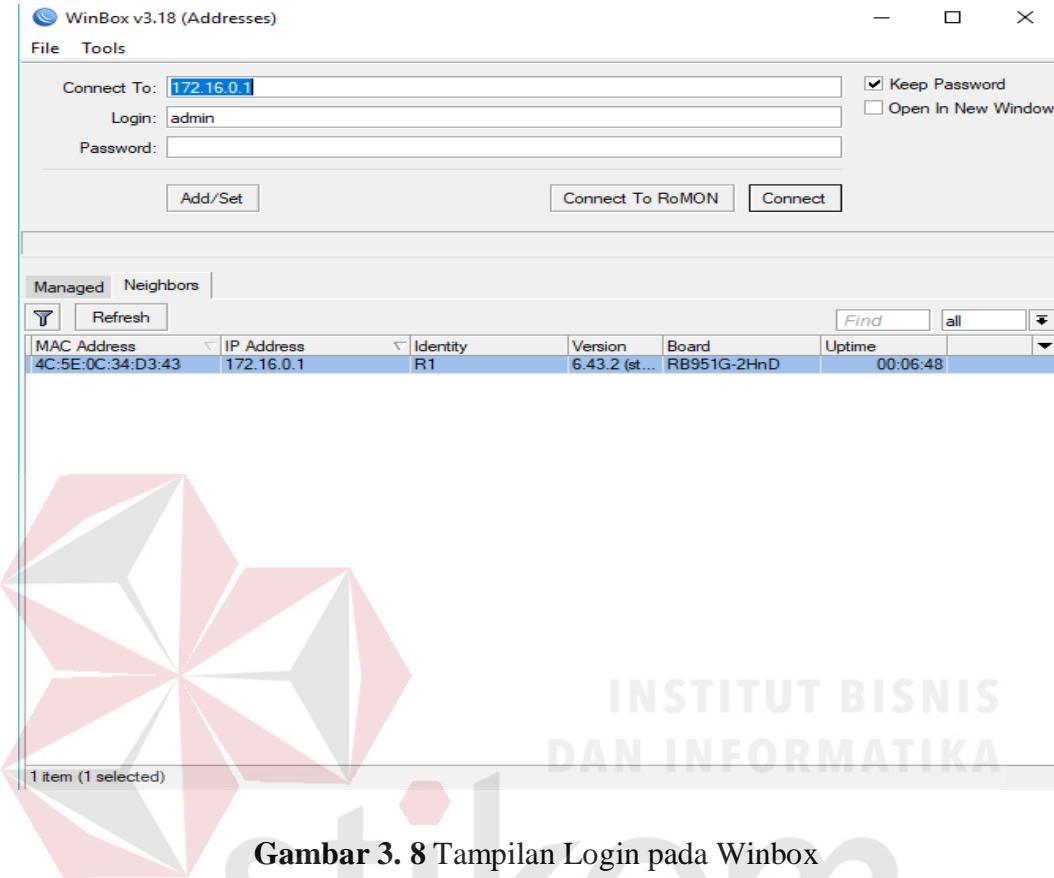
No.	Time	Source	Destination	Protocol	Length	Info
657	39.331767	172.16.10.4	224.0.0.1	IGMPv2	60	Membership Report group 224.0.0.1
1065	61.184310	172.16.10.4	224.0.0.1	IGMPv2	60	Membership Report group 224.0.0.1

Gambar 3. 7 Robot Bertahan telah bergabung kedalam Multicast Group

3.3.2 Konfigurasi IP Address

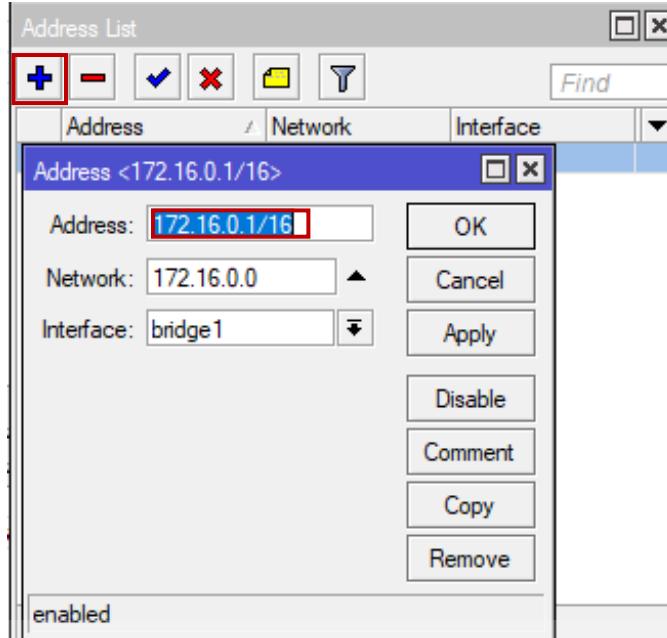
Berdasarkan IP yang diberikan oleh pihak wasit ketika pertandingan KRSBI-Beroda berlangsung, IP yang digunakan merupakan ip kelas B maka, pada penelitian Tugas Akhir ini penulis juga menggunakan IP kelas B untuk semua user baik server maupun *client*. Pembagian ip dilakukan oleh router (dalam hal ini router wasit) dimana setiap user akan menerima IP setelah mendaftarkan *mac-addressnya* kepada wasit. Pada penelitian Tugas Akhir ini penulis menggunakan router mikrotik yang berperan sebagai router wasit dan bertugas untuk memberikan IP kepada setiap user yang telah mendaftarkan *mac-addressnya*. Berikut langkah-langkah konfigurasi menggunakan aplikasi Winbox.

1. Buka aplikasi Winbox dan login dengan memilih ip server Mikrotik, kemudian tekan tombol *connect*. Dapat dilihat pada gambar 3.8.



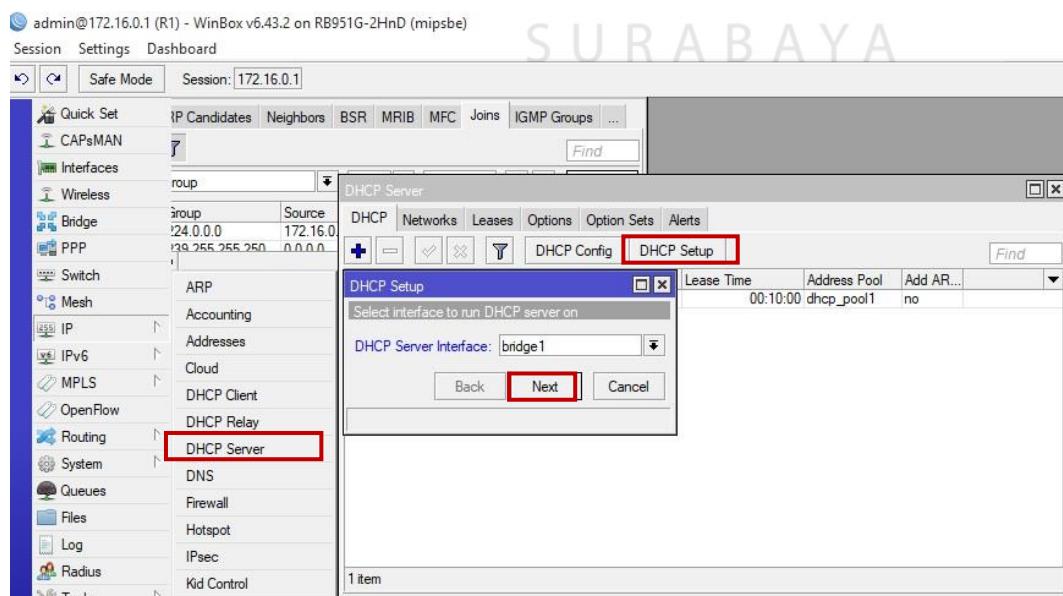
Gambar 3. 8 Tampilan Login pada Winbox

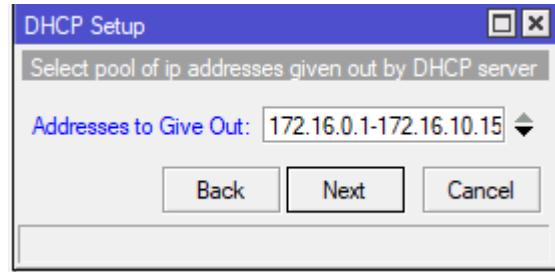
2. Setelah login muncul tampilan menu utama Winbox, pilih menu IP → Address → Address List, kemudian tambahkan ip server yang akan menjadi *gateway* untuk setiap *clinet*, dalam hal ini penulis memberikan ip server 172.16.0.1 subnet mask 255.255.0.0 (/16) dengan interface bridge1. Seperti yang dicontohkan pada gambar 3.9 berikut ini.



Gambar 3. 9 Konfigurasi IP Server

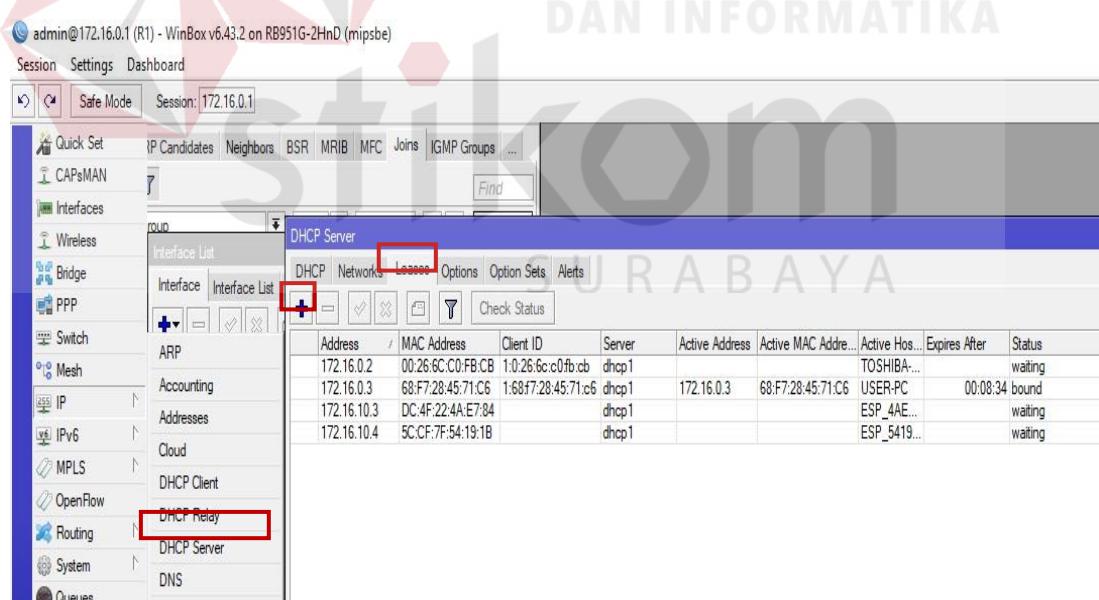
3. Langkah selanjutnya mngkonfigurasi DHCP Server untuk memberikan IP kepada client secara otomatis. Pilih IP → DHCP Server → DHCP Setup dan ikuti langkah-langkah selanjutnya. Dalam hal ini penulis menggunakan bridge1 sebagai DHCP Server Interface dengan Address to Give Out 172.16.0.2-172.16.10.15. Dapat dilihat pada gambar 3.10 berikut ini.





Gambar 3. 10 Konfigurasi DHCP Server pada Bridge1

4. Setelah menkonfigurasi DHCP Server langkah terakhir adalah memberikan IP kepada *client* yang mendaftarkan *mac-addressnya* dengan cara pilih menu IP → DHCP Server → *Leases* → pilih icon +. Dalam hal ini penulis mendaftarkan 4 perangkat (*client*) yang terdiri dari 2 perangkat komputer (1 laptop sebagai *RefreeBox*, 1 laptop sebagai *Base Station*) dan 2 perangkat wemos (1 sebagai robot penyerang, 1 sebagai robot bertahan). Seperti yang dapat dilihat pada gambar 3.11 berikut ini.



Gambar 3. 11 Daftar Perangkat yang didaftarkan kepada DHCP Server

3.3.3 Perancangan *Base Station*

Base Station dibuat dengan aplikasi visual studio 2013 dimana bahasa pemograman yang digunakan adalah C#. Algoritma *socket programming* dibuat sebagai algoritma komunikasi antara *Base Station* dan robot dengan menggunakan protokol UDP, berikut langkah-langkah kerjanya :

1. Deklarasi *library* Net Socket, berfungsi untuk dapat menggunakan *class-class* komunikasi dalam *socket*.
2. Deklarasi *library* Therading, berfungsi untuk dapat menggunakan *class* pengatur waktu (Timer).
3. Algoritma UDP *connection* :
 - a. Inisialisasi *socket* dengan *class* (*AdressFamily*, *SocketType*, *ProtocolType*), berfungsi untuk menentukan protokol yang berjalan diatas *socket*.
 - b. Inisialisasi *IPEndpoint* dengan *class* (*IPAdress* dan Port) berfungsi untuk mendeklarasikan IP dan Port yang dituju.
 - c. Menggunakan *class* *Conection* untuk membangun koneksi dari inisialisasi *socket* dan *IPEndpoint*.
4. Algoritma UDP *Sender* :
 - a. Status terkoneksi.
 - b. Inisialisasi variabel pesan dengan tipe data string.
 - c. Konversi variabel pesan dengan tipe data string menjadi byte
 - d. Mengatur waktu pengiriman dengan memanggil *class* *threed sleep*.
 - e. Memanggil *class* *socket send* untuk mengirim pesan.
 - f. Kirim pesan.

5. Algoritma UDP *Receiver* :

- a. Status terkoneksi
- b. Inisialisasi variabel dengan tipe data byte untuk menyimpan pesan yang diterima.
- c. Mengatur waktu penerimaan dengan memanggil *class thred sleep*.
- d. Memanggil *class socket recv* untuk menerima pesan.
- e. Terima pesan dan menyimpannya ke dalam variabel baru dengan tipe data int.
- f. Konversi pesan dengan tipe data int menjadi tipe data string dan menyimpannya ke dalam variabel baru.
- g. Menampilkan pesan yang diterima .

Fungsi dari *thred sleep* dalam melakukan sinkronisasi terhadap protokol UDP adalah menjeda waktu pengiriman dan penerimaan yang dilakukan oleh *Base Station* agar *Base Station* memiliki waktu untuk menerima data yang datang dari *Refree Box* maupun dari kedua robot penyerang dan bertahan. Dengan menggunakan *thred sleep*, *Base Station* dapat secara pararel melakukan pengiriman maupun penerimaan data yang berbeda secara bersamaan.

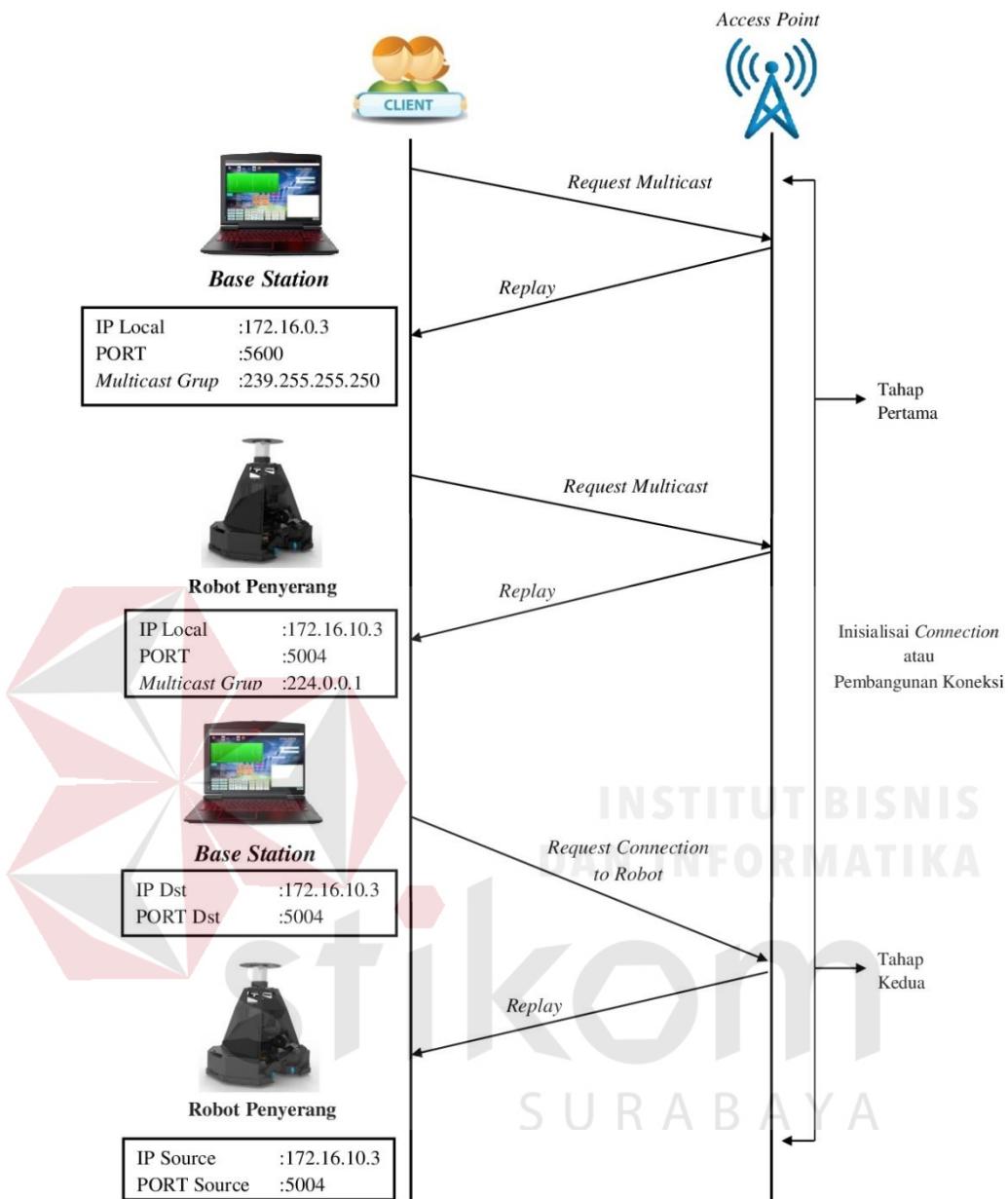
Namun dengan menggunakan *thred sleep* dalam perancangan *Base Station* ini sebagai metode yang digunakan untuk melakukan sinkronisasi terhadap protokol UDP, timbul masalah dimana fungsi dari *thred sleep* sendiri adalah mematikan sistem *Base Station* secara keseluruhan dengan jeda waktu yang telah ditentukan. Hal ini membuat sistem *Base Station* sering kali menjadi *not responding* bahkan sistem mengalami *crash* maupun *error*. Untuk itu diperlukan sistem yang dapat mengeksekusi kode tanpa mengganggu proses pada

program utama, sistem yang digunakan adalah *backgroundworker*. Seluruh pekerjaan sinkronisasi untuk menerima dan mengirim di masukan ke dalam *function backgroundworker* dan dipanggil ketika proses UDP *connection*. Tidak hanya pada pekerjaan sinkronisasi, dalam pekerjaan komunikasi antara *Base Station* dengan *Refree Box* juga dimasukan ke dalam *backgroundworker*. Dengan menggunakan *backgroundworker* seluruh sistem *Base Station* dapat bekerja secara *multitasking* tanpa mengganggu pekerjaan anatara satu dengan yang lainnya.

3.4 Diagram Waktu Sinkronisasi

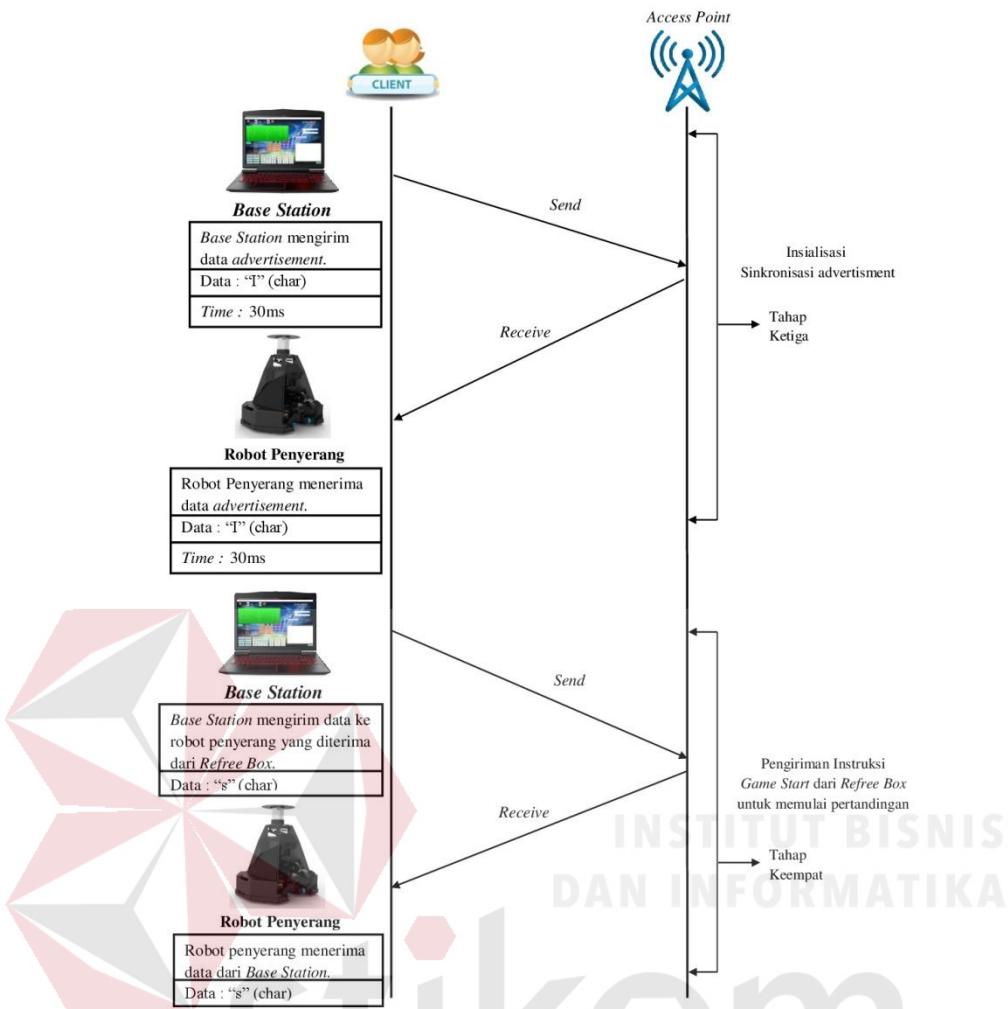
Proses sinkronisasi dilakukan oleh *Base Station* dengan mengatur waktu pengiriman dan penerimaan data dari kedua robot. Berikut akan dijelaskan secara detail diagram waktu dari proses sinkronisasi yang terjadi selama komunikasi berlangsung.

- Diagram waktu antara *Base Station* dengan robot penyerang.



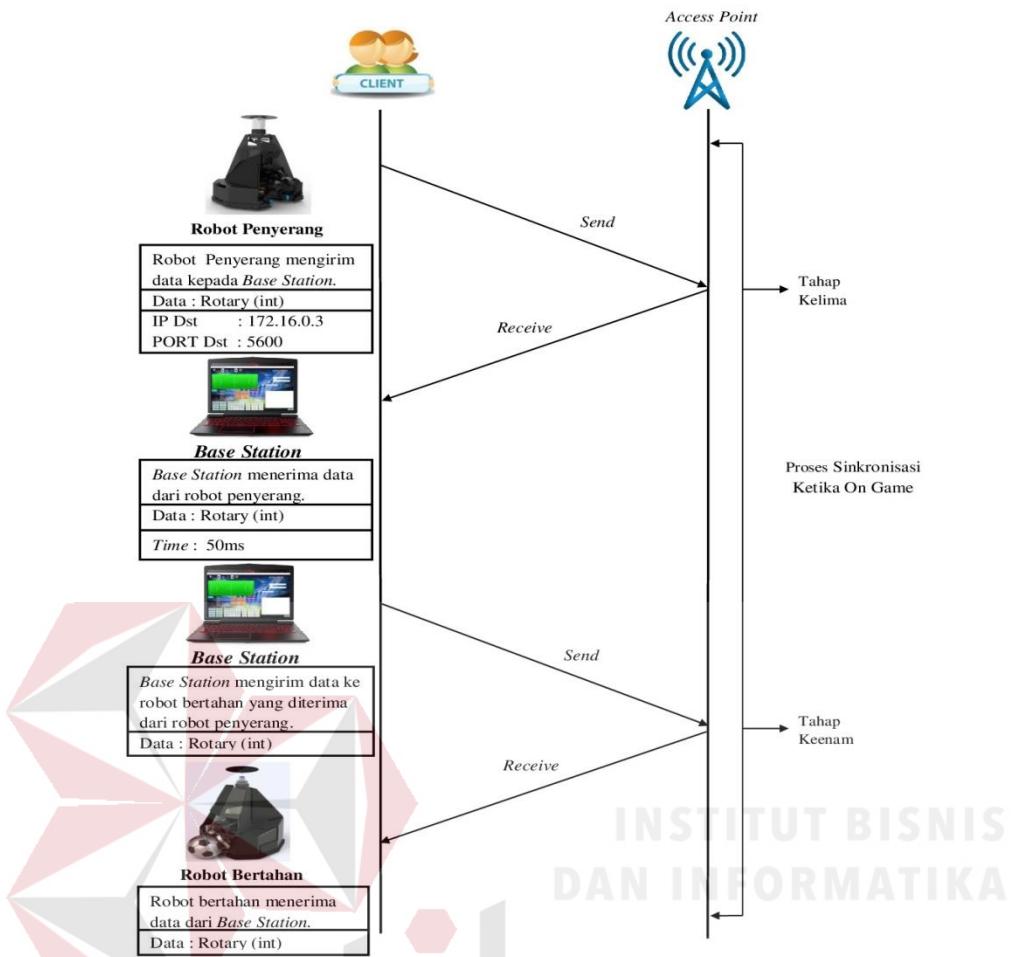
Gambar 3. 12 Diagram Waktu antara *Base Station* dan Robot Penyerang (a)

- Pada tahap pertama, *Base Station* dan robot penyerang melakukan *request connection* kepada akses point untuk mendapatkan IP, PORT, dan *Multicast Grup*. Antara *Base Station* dan robot penyerang harus dalam satu jaringan.
- Pada tahap kedua, *Base Station* melakukan *request connection* kepada robot penyerang, dengan tujuan agar *Base Station* dan robot penyerang dapat saling berkomunikasi untuk mengirim dan menerima data.



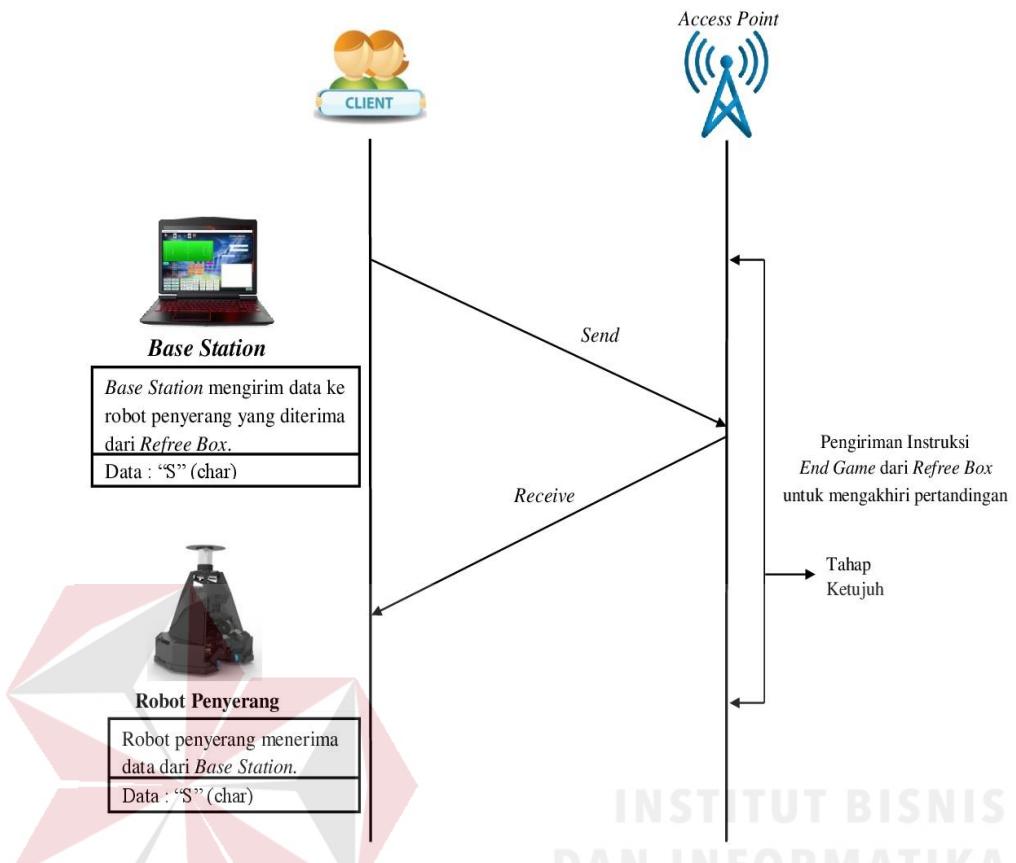
Gambar 3. 12 Diagram Waktu antara *Base Station* dan Robot Penyerang (b)

- c. Pada tahap ketiga, setelah *Base Station* terkoneksi dengan robot penyerang, *Base Station* menngirimkan data *advertisement* yaitu char ‘I’ kepada robot penyerang secara terus menerus dengan jeda waktu 30ms dan robot penyerang menerima data tersebut secara terus-menerus dengan jeda waktu 30ms. Tujuannya adalah untuk memberitahu posisi dari *Base Station* kepada robot penyerang.
- d. Pada Tahap keempat, setelah *Base Station* menerima data dari *Refree Box* berupa data start atau char ‘s’ maka data tersebut dikirimkan langsung kepada robot penyerang untuk mengeksekusinya dan permainan di mulai.



Gambar 3. 12 Diagram Waktu antara *Base Station* dan Robot Penyerang (c)

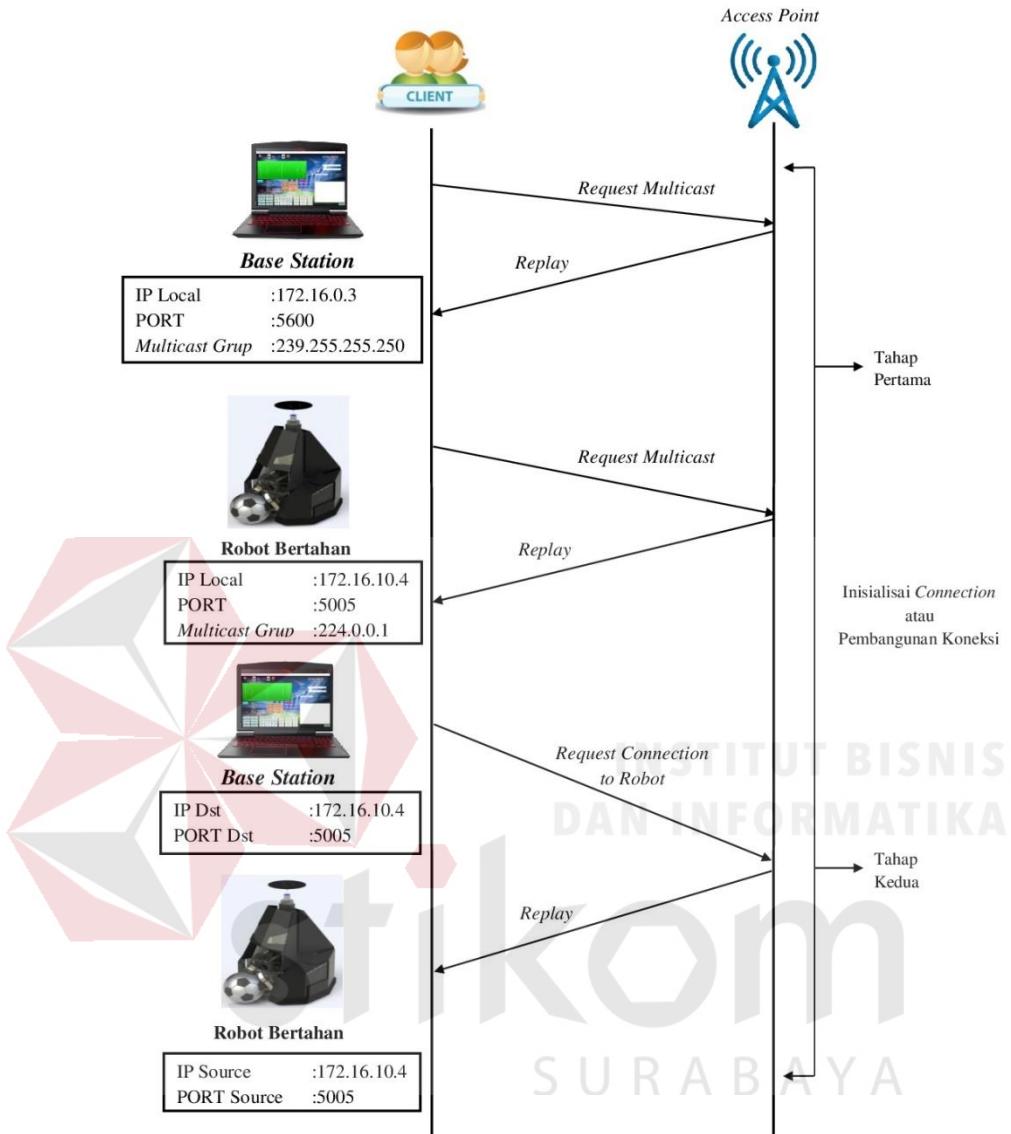
- e. Pada tahap kelima, setelah robot penyerang menjalankan instruksi *Refree Box* yang diterima dari *Base Station*, robot bergerak mengejar bola. Setiap pergerakan, robot mengirimkan data rotary kepada *Base Station* secara terus menerus dan data tersebut diterima oleh *Base Station* secara terus menerus dengan jeda waktu 50 ms. Data rotary yang diterima oleh *Base Station* langsung dikirimkan kepada robot bertahan sebagai komunikasi kedua robot.
- f. Pada tahap keenam, ketika *Base Station* menerima data rotary dari robot bertahan maka data tersebut dikirim langsung ke robot penyerang. Robot penyerang menerima data rotary tersebut sebagai informasi dari pergerakan robot bertahan.



Gambar 3. 12 Diagram Waktu antara *Base Station* dan *Robot Penyerang* (d)

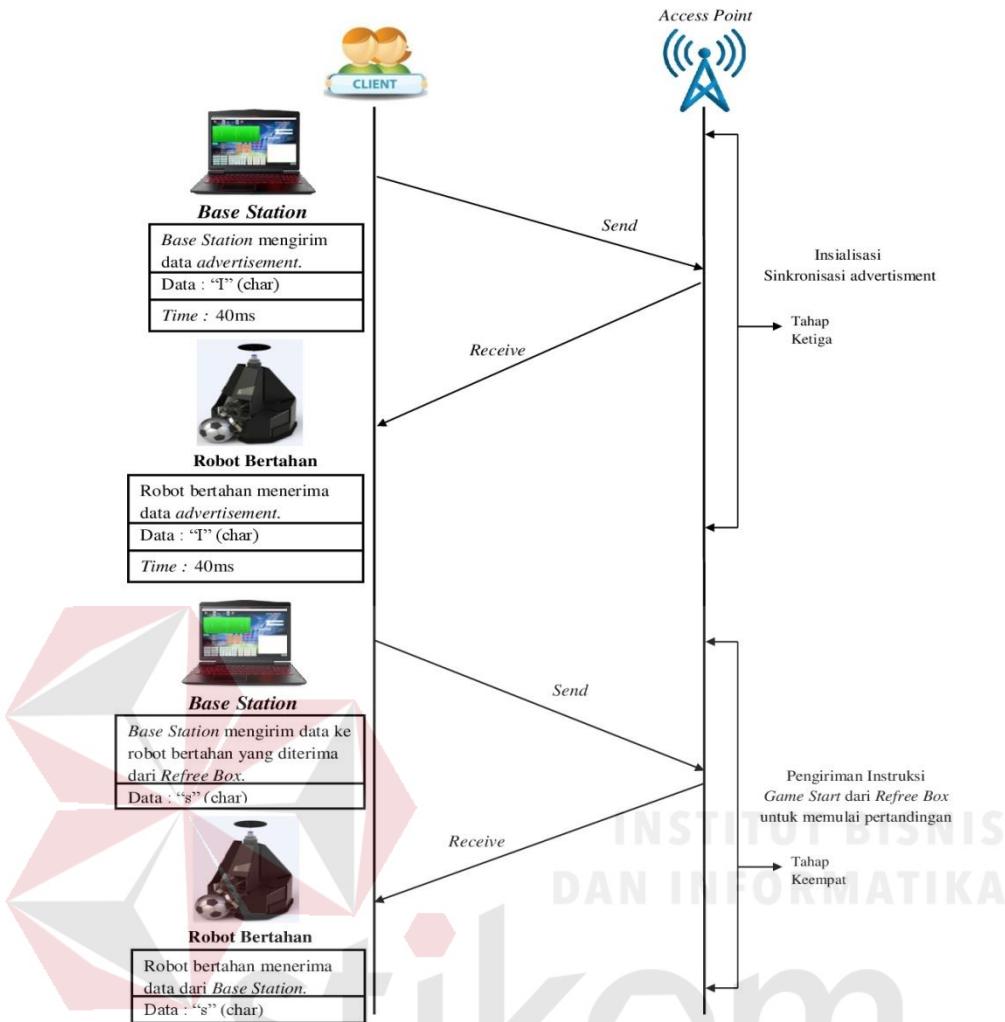
- g. Pada tahap ketujuh, saat *Base Station* menerima data dari *Refree Box* yaitu data Stop atau char 'S' maka data tersebut langsung dikirimkan kepada robot penyerang. Robot penyerang menerima data tersebut dan mengeksekusinya dengan berhenti bergerak, permainan pun telah selesai.

- b. Diagram waktu antara *Base Station* dengan robot bertahan.



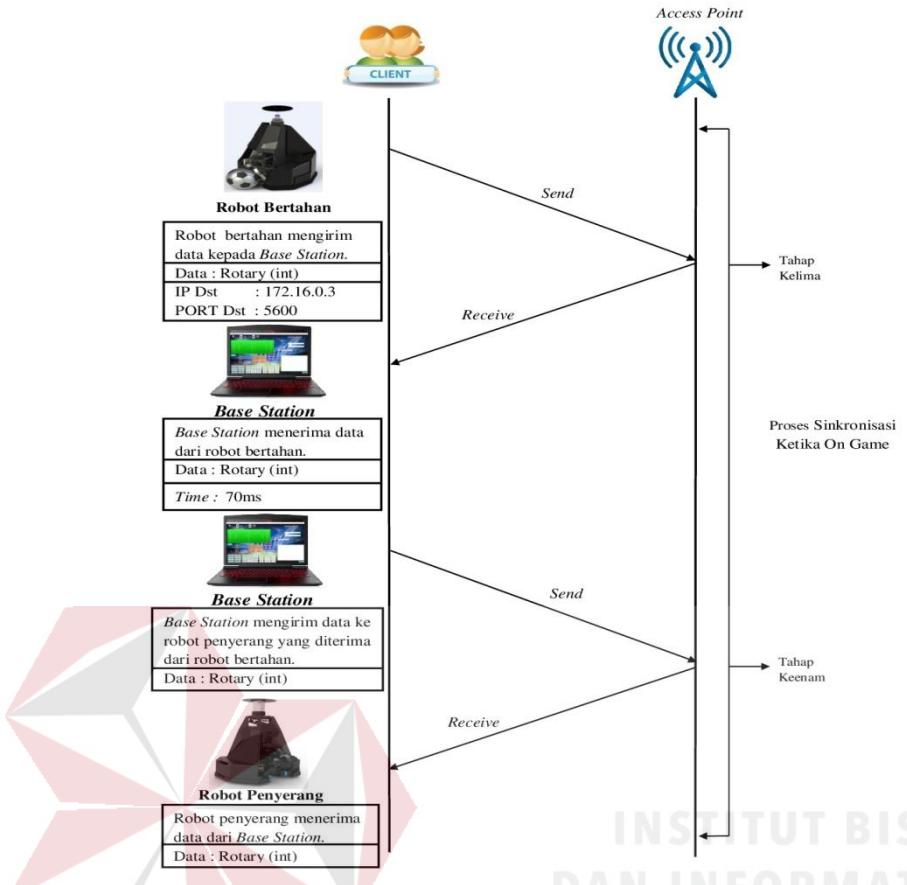
Gambar 3. 13 Diagram Waktu antara *Base Station* dan Robot Bertahan (a)

- Pada tahap pertama, *Base Station* dan robot bertahan melakukan *request connection* kepada akses point untuk mendapatkan IP, PORT, dan *Multicast Grup*. Antara *Base Station* dan robot bertahan harus dalam satu jaringan.
- Pada tahap kedua, *Base Station* melakukan *request connection* kepada robot bertahan, dengan tujuan agar *Base Station* dan robot bertahan dapat saling berkomunikasi untuk mengirim dan menerima data.



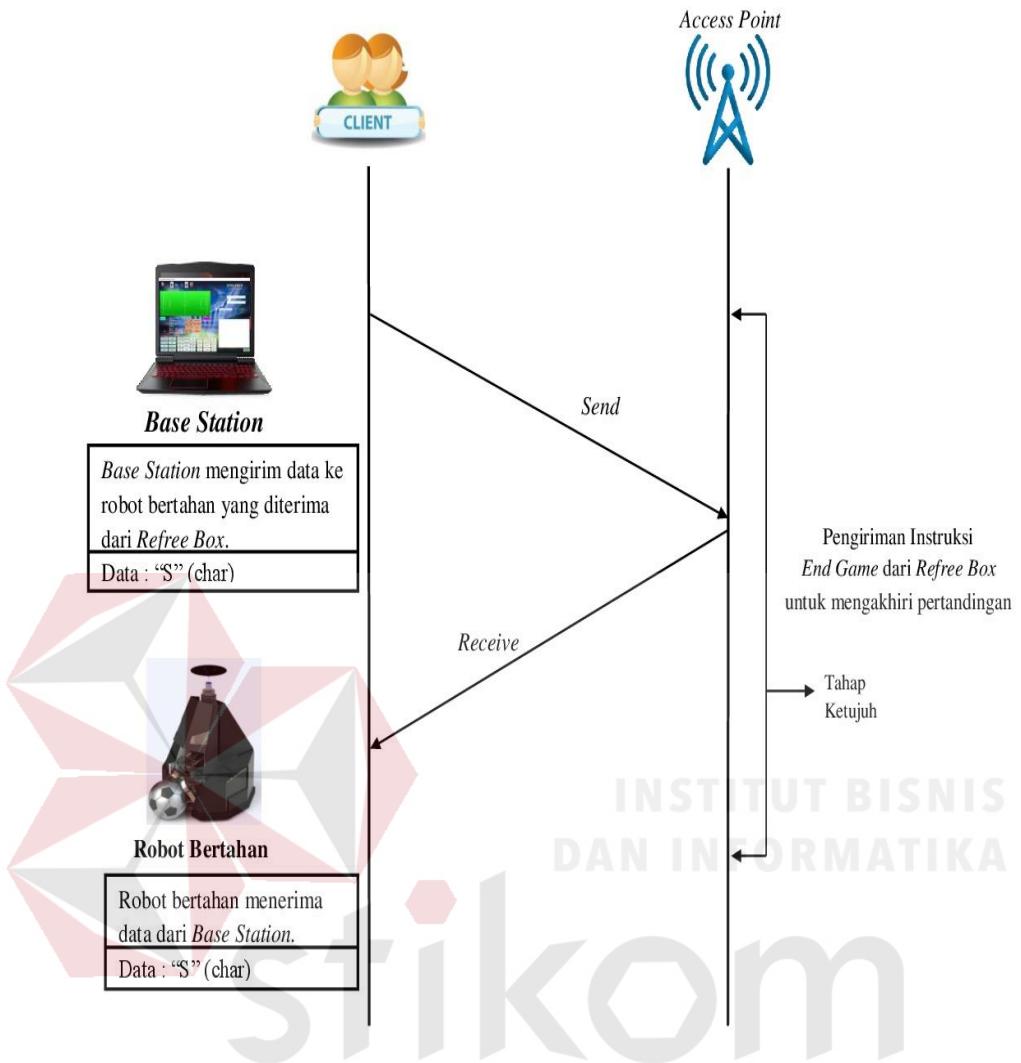
Gambar 3. 13 Diagram Waktu antara *Base Station* dan Robot Bertahan (b)

- c. Pada tahap ketiga, setelah *Base Station* terkoneksi dengan robot bertahan, *Base Station* mengirimkan data *advertisement* yaitu char ‘I’ kepada robot bertahan secara terus menerus dengan jeda waktu 40ms dan robot bertahan menerima data tersebut secara terus-menerus dengan jeda waktu 40ms. Tujuannya adalah untuk memberitahu posisi dari *Base Station* kepada robot bertahan.
- d. Pada Tahap keempat, setelah *Base Station* menerima data dari *Refree Box* berupa data start atau char ‘s’ maka data tersebut dikirimkan langsung kepada robot bertahan untuk mengeksekusinya dan permainan di mulai.



Gambar 3. 13 Diagram Waktu antara *Base Station* dan Robot Bertahan (c)

- e. Pada tahap kelima, setelah robot bertahan menjalankan instruksi *Refree Box* yang diterima dari *Base Station*, robot bergerak mengejar bola. Setiap pergerakan, robot mengirimkan data rotary kepada *Base Station* secara terus menerus dan data tersebut diterima oleh *Base Station* secara terus menerus dengan jeda waktu 70 ms. Data rotary yang diterima oleh *Base Station* langsung dikirimkan kepada robot penyerang sebagai komunikasi kedua robot.
- f. Pada tahap keenam, ketika *Base Station* menerima data rotary dari robot penyerang maka data tersebut dikirim langsung ke robot bertahan. Robot bertahan menerima data rotary tersebut sebagai informasi dari pergerakan robot penyerang.



Gambar 3. 13 Diagram Waktu antara *Base Station* dan *Robot Bertahan* (d)

- g. Pada tahap ketujuh, saat *Base Station* menerima data dari *Refree Box* yaitu data Stop atau char ‘S’ maka data tersebut langsung dikirimkan kepada robot bertahan. Robot bertahan menerima data tersebut dan mengeksekusinya dengan berhenti bergerak, permainan pun telah selesai.

3.5 Pengujian Perancangan Sistem

Penelitian Tugas Akhir ini akan dilakukan pengujian dengan sekenario seperti pada saat pertandingan KRSBI-Beroda. Skenario yang dibuat akan terdiri dari 2 robot yaitu robot penyerang dan bertahan. Kemudian akan terdapat dua laptop yang berfungsi sebagai laptop wasit dan laptop pemain, dimana laptop wasit adalah *Refree Box* dan laptop pemain adalah *Base Station*. *Switch* akan digunakan sebagai media penghubung antara *Refree Box* dan *Base Station*, terdapat satu akses point yang terhubung pula dengan *switch* yang dimana akses point ini berfungsi sebagai penghubung antara robot dengan *Base Station*. Pertandingan dimulai dengan mengkoneksikan *Base Station* ke *Refree Box* terlebih dahulu, setelah *Base Station* telah berhasil terkoneksi dengan *Refree Box* selanjutnya *Base Station* akan melakukan koneksi dengan kedua robot. Setelah semua telah terkoneksi kemudian permainan akan dimulai. *Base Station* akan mengirimkan pesan *advertisement* kepada kedua robot sebagaimana yang telah dijelaskan pada diagram waktu untuk menginformasikan posisi dari *Base Station*, kemudian permainan dimulai setelah *Refree Box* mengirimkan instruksi mulai ke *Base Station* dan instruksi tersebut kemudian diteruskan untuk dikirimkan kepada kedua robot. Setelah kedua robot menerima data tersebut maka selanjutnya kedua robot akan mengeksekusi dengan bergerak sesuai dengan program yang telah ditanamkan pada kedua robot, selanjutnya kedua robot akan terus mengirimkan pesan atau data kepada *Base Station* sebagaimana yang telah dijelaskan pada diagram waktu. Pengujian akan dilakukan sebanyak 3 babak.

Hasil dari pengujian tersebut akan dievaluasi dengan melakukan analisis melalui tiga parameter yaitu, pertama besar *packet loss* pengiriman yang

dilakukan oleh *Base Station* dan robot, kedua besar *delay* selama proses pengiriman dan penerimaan yang dilakukan oleh *Base Station* dan robot, dan yang ketiga adalah rekasi dari robot saat menerima data dari *Base Station*, bagaimana robot mengeksekusi perintah tersebut dengan tolak ukur kuantitatif. Pengujian juga akan dilakukan pada protokol TCP/IP dan UDP tanpa sinkronisasi. Sehingga akan terdapat 3 hasil pengujian untuk memperoleh kesimpulan apakah sinkronisasi yang dilakukan terhadap protokol UDP dapat menghasilkan efisiensi distribusi komunikasi data yang optimal dan efektif dari kedua protokol lainnya yaitu TCP dan UDP tanpa sinkronisasi.

Pengujian terhadap jarak jangkauan penerimaan media *receiver* dalam hal ini juga akan dilakukan pengambilan data untuk mengukur apakah robot dapat menerima setiap data yang dikirimkan oleh *Base Station* di dalam arena pertandingan sepak bola yang berukuran 9x6 m. Dari pengujian ini diharapkan *tools* yang digunakan sebagai media *receiver* dapat menjangkau semua area dari arena pertandingan sehingga tidak ada data yang tidak dapat diterima oleh robot.

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

Dalam bab ini penulis akan menguraikan dan menampilkan hasil pengujian dari analisis tiga parameter yang digunakan yaitu *packet loss*, *delay*, dan reaksi robot, sebagai output terhadap penelitian tugas akhir ini.

4.1 Pengujian *Quality of Service Packet Loss*

Seperti yang telah dijelaskan pada bab 2 bahwa salah satu parameter untuk menggambarkan jumlah paket yang hilang adalah *packet loss*. Dalam Tugas Akhir ini, *packet loss* yang dihitung merupakan paket yang hilang selama proses transaksi komunikasi berlangsung. Berikut penjabaran lebih detail.

4.1.1 Tujuan Pengujian *Packet Loss*

Pengujian dari *packet loss* ini bertujuan untuk mengetahui seberapa besar transaksi paket yang hilang pada saat proses pengiriman data antara *Base Station* dan robot berlangsung .

4.1.2 Komponen Yang Dibutuhkan Pada Pengujian *Packet Loss*

Adapun komponen yang digunakan dalam pengujian ini, antara lain :

1. Akses Point *switch*
2. Laptop
3. Robot
4. Aplikasi *Wireshark*
5. Aplikasi *Base Station*

4.1.3 Prosedur Pengujian *Packet Loss*

Berikut langkah-langkah dalam pengujian *packet loss* dari pertukaran data anatar *Base Station* dan robot.

1. Menyalakan power robot untuk mode on.
2. Membuka aplikasi *Base Station* yang telah dibuat oleh peserta.
3. Melakukan koneksi dari *Base Station* kepada robot.
4. Membuka aplikasi *wireshark*, memilih *local area connection* untuk memonitoring kinerja jaringan didalamnya, dan menjalankannya dengan memilih *start capturing packets*.
5. Lakukan *filter* pada *apply a display filter* untuk menampilkan ip kedua robot.
6. Tampilkan *identification* pada kolom pengamatan dengan memilih *internet protocol version 4* pada *box detail*.
7. Amati pengiriman serta penerimaan paket data anatara *Base Station* dan robot pada kolom *identification*.

4.1.4 Hasil Pengujian *Packet Loss*

Pengujian dilakukan terhadap 3 protokol dengan percobaan sebanyak 3 babak. Masing-masing babak dilakukan pengambilan data 15 kali (data yang diambil merupakan data transaksi komunikasi yang terekam oleh *Wireshark* selama proses pengiriman dan penerimaan berlangsung). Pengujian hasil analisis yang disajikan pada tabel 4.1, 4.2, dan 4.3 merupakan hasil perhitungan dari rumus *packet loss* yang telah dijelaskan pada bab 2 sebelumnya.

1. Babak Pertama

Tabel 4. 1 Hasil Perhitungan *Packet Loss* (dalam %) Babak Pertama

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,0004239%	0,0168873%	0,0370908%	0,0085357%	0,0339509%	0,0102204%
2.	0,0305819%	0,0608557%	0,0353397%	0,0123944%	0,0315197%	0,0507220%
3.	0,0372000%	0,0942273%	0,0146340%	0,0446774%	0,0435595%	0,0448857%
4.	0,0369180%	0,0943148%	0,0035028%	0,0008575%	0,0165963%	0,0386961%
5.	0,0237519%	0,1119933%	0,0182771%	0,0231767%	0,0187301%	0,0133742%
6.	0,0249479%	0,0888119%	0,0033685%	0,0018327%	0,0042221%	0,0066487%
7.	0,0262631%	0,0860192%	0,0063326%	0,0170496%	0,0474130%	0,0114202%
8.	0,0282023%	0,0843096%	0,0009105%	0,0071597%	0,0126620%	0,0503883%
9.	0,0808308%	0,0905153%	0,0063226%	0,0161621%	0,0365458%	0,0461393%
10.	0,0160389%	0,0363835%	0,0335774%	0,0279451%	0,0273972%	0,0383907%
11.	0,0881675%	0,0504158%	0,0162705%	0,0353139%	0,0086585%	0,0237874%
12.	0,0466280%	0,0493526%	0,0278894%	0,0051231%	0,0033464%	0,0303632%
13.	0,0015631%	0,0229777%	0,0280060%	0,0111734%	0,0453344%	0,0448109%
14.	0,0033544%	0,0514939%	0,0250955%	0,0037221%	0,0358523%	0,0393941%
15.	0,0228142%	0,0217272%	0,0255206%	0,0326757%	0,0088246%	0,0335945%
RATA-RATA PACKET LOSS						
	0,031%	0,49%	0,019%	0,017%	0,025%	0,034%

2. Babak Kedua

Tabel 4. 2 Hasil Perhitungan *Packet Loss* (dalam %) Babak Kedua

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,0071170%	0,2698774%	0,0023499%	0,0120235%	0,0147215%	0,0193416%
2.	0,1102398%	0,2471148%	0,0015118%	0,0007795%	0,0338813%	0,0147215%
3.	0,0013802%	0,2294830%	0,0089718%	0,0045941%	0,0189493%	0,0384669%
4.	0,0592416%	0,2220035%	0,0003591%	0,0181402%	0,0368183%	0,0338840%
5.	0,1450316%	0,2192169%	0,0126277%	0,0082535%	0,0320597%	0,0294381%
6.	0,1614322%	0,2110403%	0,0060967%	0,0054281%	0,0283450%	0,0046760%
7.	0,1047553%	0,2374820%	0,0047802%	0,0065645%	0,0326990%	0,0320624%
8.	0,0755883%	0,2287736%	0,0046845%	0,0055785%	0,0310493%	0,0303982%
9.	0,0078442%	0,1902069%	0,0032332%	0,0183742%	0,0217904%	0,0091011%
10.	0,1374226%	0,1842323%	0,0024735%	0,0054341%	0,0197652%	0,0112294%
11.	0,1546671%	0,3430545%	0,0017243%	0,0185103%	0,0103677%	0,0093499%
12.	0,1538986%	0,0436284%	0,0010445%	0,0086808%	0,0277830%	0,0367516%
13.	0,1381439%	0,0262146%	0,0016185%	0,0132789%	0,0038039%	0,0042796%
14.	0,1368449%	0,0792469%	0,0108739%	0,0043389%	0,0090163%	0,0342033%
15.	0,1311734%	0,5242287%	0,0010609%	0,0024589%	0,0034175%	0,0305535%
RATA-RATA PACKET LOSS						
	0,102%	0,217%	0,004%	0,009%	0,022%	0,023%

3. Babak Ketiga

Tabel 4. 3 Hasil Perhitungan *Packet Loss* (dalam %) Babak Ketiga

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,1519885%	0,0203551%	0,0134991%	0,0034807%	0,0034244%	0,0034100%
2.	0,0441572%	0,0413657%	0,0370908%	0,0194470%	0,0020205%	0,0019198%
3.	0,0334927%	0,0602919%	0,0043947%	0,0296968%	0,0006329%	0,0102024%
4.	0,0969842%	0,0598991%	0,0213455%	0,0342173%	0,0106077%	0,0078547%
5.	0,0872824%	0,0593674%	0,0191997%	0,0306673%	0,0092960%	0,0047181%
6.	0,0907608%	0,0624857%	0,0097471%	0,0031512%	0,0023447%	0,0087082%
7.	0,0579411%	0,0604207%	0,0257756%	0,0161369%	0,0018628%	0,0087873%
8.	0,0302153%	0,0581722%	0,0061634%	0,0121608%	0,0027918%	0,0070202%
9.	0,1152367%	0,0631269%	0,0024849%	0,0241025%	0,0015286%	0,0020197%
10.	0,1241256%	0,0360986%	0,0211723%	0,0038431%	0,0084845%	0,0017767%
11.	0,0209708%	0,0114037%	0,0031907%	0,0044365%	0,0062931%	0,0053716%
12.	0,0835984%	0,0302784%	0,0013185%	0,0039787%	0,0042263%	0,0014892%
13.	0,1460620%	0,0199401%	0,0309746%	0,0379350%	0,0086672%	0,0032334%
14.	0,1352955%	0,0120119%	0,0182159%	0,0067676%	0,0057371%	0,0084299%
15.	0,1274251%	0,0014416%	0,0267630%	0,0371492%	0,0043532%	0,0059227%
RATA-RATA PACKET LOSS						
	0,09%	0,04%	0,016%	0,018%	0,005%	0,005%

Berdasarkan hasil pengujian yang telah dilakukan, terdapat perlakuan khusus kepada UDP tanpa sinkronisasi. Perlakuan khusus tersebut adalah menambahkan waktu pengiriman saat *Base Station* melakukan *advertisement* kepada kedua robot (dalam hal ini ditambahkan *thread sleep* di program *Base Station*), jika UDP tanpa sinkronisasi tidak ditambahkan waktu sama sekali maka *Base Station* tidak akan bisa mengirim instruksi lainnya kepada robot, begitu pula robot tidak akan bisa mengirim balik data yang akan dikirim. Sehingga *packet loss* tidak akan bisa dihitung dan pengambilan data pada *Wireshark* pun tidak dapat dilakukan.

Pada babak pertama hasil pengujian *packet loss* yang ditunjukan oleh tabel 4.1 UDP dengan sinkronisasi mempunyai *packet loss* yang lebih kecil dari pada TCP dan UDP tanpa sinkronisasi yaitu sebesar 0.019% untuk robot 1 dan 0.017% pada robot 2. Pada babak kedua hasil pengujian *packet loss* yang ditunjukan oleh tabel 4.2 UDP dengan sinkronisasi masih tetap memiliki *packet loss* yang lebih kecil dari pada TCP dan UDP tanpa sinkronisasi yaitu sebesar 0.004% untuk robot 1 dan 0.009% untuk robot 2. Pada babak ketiga hasil pengujian *packet loss* yang ditunjukan oleh tabel 4.3 UDP dengan sinkronisasi memiliki *packet loss* yang lebih kecil dari pada TCP namun lebih besar dari pada UDP tanpa sinkronisasi yaitu sebesar 0.016% untuk robot 1 dan 0.018% untuk robot 2.

Total rata-rata pengujian *packet loss* dari ketiga babak, untuk robot 1 sebesar 0.074% robot 2 sebesar 0.249% pada protokol TCP, robot 1 sebesar 0.013% robot 2 sebesar 0.014% pada protokol UDP dengan sinkronisasi, dan robot 1 sebesar 0.017% robot 2 sebesar 0.020% pada protokol UDP tanpa sinkronisasi. Dapat disimpulkan bahwa UDP dengan sinkronisasi mempunyai

packet loss lebih kecil dari pada protokol TCP dan UDP tanpa sinkronisasi. Namun ketiganya berdasarkan kategori degredasi masuk kedalam kategori sangat bagus seperti yang ditunjukan pada tabel 4.4 dibawah ini.

Tabel 4. 4 Kategori Degredasi Standarisasi ITU

KATEGORI DEGREDASI	PACKET LOSS
Sangat bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

4.2 Pengujian *Quality of Service Delay*

Dilakukannya pengujian *delay* adalah untuk mengetahui besar *delay* pada masing-masing protokol dari transaksi komunikasi yang sedang berlangsung antara *Base Station* dengan robot 1 maupun dengan robot 2.

4.2.1 Tujuan Pengujian *Delay*

Tujuan dari pengujian *delay* ini adalah untuk menghitung besar *delay* yang terjadi selama transaksi komunikasi berlangsung dan menjadi tolak ukur dalam menilai masing-masing protokol.

4.2.2 Komponen Yang Dibutuhkan Pada Pengujian *Delay*

Adapun komponen yang digunakan dalam pengujian ini, anatara lain :

1. Akses Point dan *switch*
2. Laptop
3. Robot
4. Aplikasi *Wireshark* dan *Base Station*

4.2.3 Prosedur Pengujian *Delay*

Berikut langkah-langkah dalam pengujian *delay* dari pertukaran data antara *Base Station* dan robot.

1. Menyalakan power robot untuk mode on.
2. Membuka aplikasi *Base Station* dan melakukan koneksi kepada robot.
3. Membuka aplikasi *wireshark*, memilih *local area connection* untuk memonitoring kinerja jaringan didalamnya, dan menjalankannya dengan memilih *start capturing packets*.
4. Lakukan *filter* pada *apply a display filter* untuk menampilkan ip kedua robot.
5. Tampilkan *time since reference or first frame* pada kolom pengamatan dengan memilih *frame* pada *box detail*.
6. Amati pengiriman serta penerimaan paket data antara *Base Station* dan robot pada kolom *time delta*.

4.2.4 Hasil Pengujian *Delay*

Pengujian terhadap *delay* dilakukan percobaan sebanyak 3 babak dengan masing-masing babak dilakukan pengambilan data sebanyak 15 kali kali (data yang diambil merupakan data transaksi komunikasi yang terekam oleh *Wireshark* selama proses pengiriman dan penerimaan berlangsung). Hasil pengujian *delay* yang tersaji di dalam tabel 4.5, 4.6, dan 4.7 merupakan hasil perhitungan rumus *delay* berdasarkan standarisasi ITU-T*delay*.

1. Babak Pertama

Tabel 4. 5 Hasil Perhitungan *Delay* Babak Pertama

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,054888	0,155346	0,010776	0,007978	0,006031	0,002925
2.	0,053406	0,159695	0,008955	0,009811	0,001362	0,014540
3.	0,068743	0,156546	0,006529	0,008507	0,015317	0,005665
4.	0,069569	0,156546	0,004933	0,019562	0,004910	0,003884
5.	0,017042	0,158500	0,005707	0,014695	0,002878	0,004776
6.	0,095084	0,152049	0,013221	0,000816	0,011356	0,010382
7.	0,098087	0,217726	0,011129	0,033518	0,004364	0,008989
8.	0,151865	0,154764	0,019330	0,014781	0,013162	0,006806
9.	0,151964	0,087387	0,005781	0,034526	0,008984	0,001619
10.	0,152045	0,062500	0,014489	0,029663	0,013554	0,002340
11.	0,155110	0,126727	0,016775	0,041303	0,016803	0,007206
12.	0,152469	0,100495	0,010472	0,013971	0,006572	0,009880
13.	0,013606	0,298266	0,015053	0,043418	0,004542	0,009323
14.	0,153379	0,017030	0,004752	0,002314	0,002719	0,008158
15.	0,152016	0,032646	0,031193	0,017896	0,014273	0,003329
TOTAL DELAY						
	1,539273	2,036041	0,179095	0,179095	0,126827	0,099822
RATA-RATA DELAY (ms)						
	102,62	135,74	11,94	19.52	8,46	6,65

2. Babak Kedua

Tabel 4. 6 Hasil Perhitungan *Delay* Babak Kedua

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,151866	0,157786	0,029524	0,036927	0,008811	0,001744
2.	0,152124	0,152469	0,021614	0,002346	0,006687	0,000806
3.	0,100741	0,155105	0,028228	0,023132	0,013181	0,012525
4.	0,080634	0,157089	0,026077	0,007897	0,006984	0,003635
5.	0,049491	0,158127	0,024649	0,037076	0,003293	0,013787
6.	0,009489	0,154625	0,003795	0,004120	0,002139	0,010609
7.	0,072433	0,164994	0,006774	0,045682	0,006282	0,014034
8.	0,123376	0,155678	0,004737	0,040148	0,007935	0,005902
9.	0,004855	0,155322	0,018922	0,026030	0,001266	0,010891
10.	0,111881	0,157726	0,029116	0,010306	0,000797	0,010353
11.	0,094002	0,524173	0,008697	0,004654	0,000031	0,014052
12.	0,113726	0,155102	0,010238	0,032350	0,015642	0,008032
13.	0,014001	0,107636	0,004510	0,019970	0,010140	0,003741
14.	0,028808	0,088829	0,001728	0,032098	0,001057	0,011317
15.	0,081539	0,045990	0,016420	0,007550	0,001145	0,009988
TOTAL DELAY						
	0,081539	2,490651	0,235029	0,330286	0,085390	0,131416
RATA-RATA DELAY (ms)						
	79,26	166,04	15,67	22,02	5,69	8,76

3. Babak Ketiga

Tabel 4. 7 Hasil Perhitungan *Delay* Babak Ketiga

No	PROTOKOL					
	TCP		UDP SINKRONISASI		UDP	
	Base Station	Base Station	Base Station	Base Station	Base Station	Base Station
	To Robot 1	To Robot 2	To Robot 1	To Robot 2	To Robot 1	To Robot 2
1.	0,094004	0,109197	0,007084	0,023883	0,004779	0,014442
2.	0,077346	0,164939	0,007635	0,012992	0,000884	0,001881
3.	0,086598	0,155320	0,029113	0,022150	0,012995	0,011968
4.	0,048253	0,155194	0,024999	0,037960	0,015031	0,003009
5.	0,078249	0,155920	0,015589	0,032171	0,003925	0,014008
6.	0,111601	0,152843	0,006187	0,009319	0,003406	0,000813
7.	0,119396	0,155292	0,000827	0,009894	0,007360	0,007120
8.	0,026426	0,155431	0,020973	0,020591	0,012386	0,009012
9.	0,119696	0,166118	0,015015	0,002705	0,006703	0,006343
10.	0,041198	0,163267	0,025781	0,001966	0,007071	0,002540
11.	0,013371	0,155407	0,006491	0,004446	0,003216	0,002298
12.	0,072605	0,157040	0,001936	0,045167	0,008265	0,013579
13.	0,152238	0,156147	0,007086	0,025752	0,013752	0,004214
14.	0,151863	0,156514	0,011849	0,034400	0,011507	0,012759
15.	0,154578	0,156478	0,003707	0,021415	0,009483	0,005615
TOTAL DELAY						
	1,347422	2,315107	0,184272	0,304811	0,120763	0,109601
RATA-RATA DELAY (ms)						
	89,83	154,34	12,28	20,32	8,05	7,31

Sama seperti pengujian *packet loss*, agar pengujian *delay* terhadap protokol UDP tanpa sinkronisasi dapat menghasilkan suatu nilai maka perlu dilakukan perlakuan khusus yaitu menambahkan waktu (*thread sleep*) pada program *Base Station* saat melakukan pengiriman *advertisement* kepada kedua robot. Pada pengujian babak pertama yang disajikan oleh tabel 4.4 dapat dilihat bahwa rata-rata *delay* masing-masing protokol berbeda-beda. Protokol UDP dengan sinkronisasi rata-rata *delay* adalah 11.94 ms untuk robot 1 dan 19.52 ms untuk robot 2, lebih kecil dibandingkan rata-rata *delay* yang dihasilkan oleh protokol TCP dan lebih besar dari rata-rata *delay* yang dihasilkan oleh protokol UDP tanpa sinkronisasi.

Pada pengujian babak kedua yang disajikan oleh tabel 4.5 rata-rata *delay* yang dihasilkan oleh UDP dengan sinkronisasi sebesar 15.67 ms untuk robot 1 dan 22.02 ms untuk robot 2, masih tetap lebih kecil dari rata-rata *delay* protokol TCP namun lebih besar dari rata-rata *delay* protokol UDP tanpa sinkronisasi. Begitu pula pada pengujian babak ketiga yang disajikan oleh tabel 4.6 rata-rata *delay* yang dihasilkan oleh UDP dengan sinkronisasi sebesar 12.28 ms untuk robot 1 dan 20.32 ms untuk robot 2 masih lebih kecil dari rata-rata *delay* protokol TCP dan lebih besar dari rata-rata *delay* protokol UDP tanpa sinkronisasi.

Untuk total rata-rata dari ketiga babak, besar *delay* protokol TCP adalah 90.57 ms untuk robot 1 dan 152.04 ms untuk robot 2, besar *delay* protokol UDP dengan sinkronisasi adalah 13.30 ms untuk robot 1 dan 20.62 untuk robot 2, dan besar *delay* protokol UDP tanpa sinkronisasi adalah 7.40 ms untuk robot 1 dan 7.57 ms untuk robot 2. Sehingga dapat disimpulkan UDP tanpa sinkronisasi mempunyai *delay* lebih kecil dibanding UDP dengan sinkronisasi maupun TCP.

Namun UDP dengan sinkronisasi dan UDP tanpa sinkronisasi masuk kedalam kategori *delay Excellent* untuk robot 1 dan robot 2, sedangkan TCP masuk kedalam kategori *delay Excellent* untuk robot 1 dan *Good* untuk robot 2, seperti yang ditunjukkan pada Tabel 4.8 dibawah ini.

Tabel 4.8 Standarisasi ITU-T Delay

Kategori Delay	Besar Delay
Excellent	< 150 ms
Good	150 - 300 ms
Poor	300 - 450 ms
Unnaceptable	> 450 ms

4.3 Pengujian Reaksi Robot

Yang dimaksud pengujian terhadap reaksi robot dalam Tugas Akhir ini adalah untuk mengetahui apakah robot mampu menjalankan dan bergerak sesuai instruksi yang diberikan oleh *Base Station* dengan tolak ukur kuantitaif. Perubahan gerakan yang dilakukan oleh robot 1 dan robot 2 akan diukur berdasarkan perintah yang telah diberikan.

4.3.1 Tujuan Pengujian Reaksi Robot

Tujuan dari pengujian ini adalah untuk mengetahui dan mengukur setiap perubahan gerakan dari reaksi robot secara kuantitatif dalam menjalankan instruksi yang dikirimkan oleh *Base Station*.

4.3.2 Komponen Yang Dibutuhkan Pada Pengujian Reaksi Robot

Adapun komponen yang digunakan dalam pengujian ini, anatara lain :

1. Akses Point dan *switch*
2. Kabel UTP
3. Laptop
4. Robot
5. Aplikasi *Base Station*
6. Meteran dan Busur.

4.3.3 Prosedur Pengujian Reaksi Robot

Berikut langkah-langkah pengujian reaksi robot dalam menerima data dari *Base Station*.

1. Menyalakan power robot untuk mode on.
2. Membuka aplikasi *Base Station* dan melakukan koneksi kepada robot.
3. *Base Station* mengirim data instruksi ‘s’ (*start*) kepada robot yang diterima dari *Refree Box*.
4. *Base Station* mengirim setiap data yang diterima oleh robot 1 kepada robot 2 begitu pula sebaliknya.
5. *Base Station* mengirim data instruksi ‘S’ (*stop*) kepada robot yang diterima dari *Refree Box*.
6. Amati setiap pergerakan atau reaksi robot ketika menerima data yang dikirim oleh *Base Station*.
7. Ukuran setiap perubahan gerakan robot 1 dan robot 2.

4.3.4 Hasil Pengujian Reaksi Robot

Pengujian yang dilakukan terhadap reaksi robot dikategorikan menjadi 3 kategori yaitu reaksi robot saat menerima perintah maju, mundur, dan berotasi. Percobaan dilakukan sebanyak 3 babak, sehingga terdapat 15 kali pengukuran

pada setiap babaknya dari perubahan gerakan setiap robot. Berikut penyajian hasil pengukuran masing-masing robot.

- Hasil skenario pengujian reaksi robot
 - a. *Base Station* mengirimkan instruksi start ('s') yang diterima dari *Refree Box* kepada kedua robot.
 - b. Robot 1 dan robot 2 berhasil menerima instruksi dari *Base Station*. Robot 1 bergerak maju setelah menerima instruksi sedangkan robot 2 masih menunggu data dari robot 1, data yang dikirimkan adalah data rotari.
 - c. Setelah robot 2 menerima data rotari dari robot 1 robot 2 bergerak maju sedangkan robot 1 tetap maju hingga menerima data rotari dari robot 2.
 - d. Setelah robot 1 menerima data rotari dari robot 2, robot 1 berhenti sejenak dan kemudian bergerak mundur, sedangkan robot 2 tetap bergerak maju hingga kondisi *pulse per rotation (ppr)* terpenuhi.
 - e. Setelah *ppr* robot 2 terpenuhi maka robot 2 berhenti sejenak dan kemudian bergerak mundur, sedangkan robot 1 tetap mundur hingga kondisi *ppr* terpenuhi.
 - f. Setelah *ppr* robot 1 terpenuhi robot diam sejenak dan kemudian bergerak rotasi, sedangkan robot 2 tetap mundur hingga kondisi *ppr* terpenuhi .
 - g. Setelah *ppr* robot 2 terpenuhi robot diam sejenak dan kemudian bergerak rotasi, sedangkan robot 1 tetap bergerak rotasi hingga *ppr* terpenuhi.

- h. Robot 1 berhenti berotasi setelah menerima data rotari dari robot 2, begitu pula robot 2 berhenti berotasi setelah menerima data rotari dari robot 1.
- i. Hasil pengukuran dengan menggunakan protokol TCP :
 - Pengukuran posisi awal kedua robot dapat dilihat pada gambar 4.1.



Gambar 4. 1 Posisi awal kedua robot

- Robot 1 bergerak maju sejauh 151 cm dari posisi awal seperti yang terlihat pada gambar 4.2.



Gambar 4. 2 Robot 1 maju 151 cm dari posisi awal

- Robot 2 bergerak maju sejauh 123 cm dari posisi awal seperti yang terlihat pada gambar 4.3.



Gambar 4. 3 Robot 2 maju 123 cm dari posisi awal

- Robot 1 bergerak mundur sejauh 297 cm dari posisi terakhir robot 1 bergerak, seperti yang terlihat pada gambar 4.4.



Gambar 4. 4 Robot 1 mundur 297 cm dari posisi terakhir

- Robot 2 bergerak mundur sejauh 165 cm dari posisi terakhir robot 2 bergerak, seperti yang terlihat pada gambar 4.5.



Gambar 4. 5 Robot 2 mundur 165 cm dari posisi terakhir

- Robot 1 bergerak rotasi sebanyak 2 kali sehingga berotasi sebesar 745° , seperti yang terlihat pada gambar 4.6.



Gambar 4. 6 Robot 1 berotasi sebesar 745°

- Robot 2 bergerak rotasi sebanyak 1 kali sehingga berotasi sebesar 275° , seperti yang terlihat pada gambar 4.7.



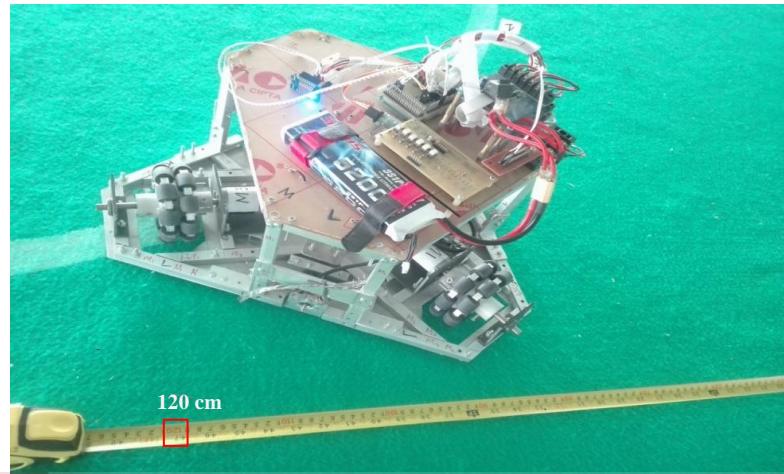
Gambar 4. 7 Robot 2 berotasi sebesar 275°

- j. Hasil pengukuran menggunakan protokol UDP Sinkronisasi :
 - Robot 1 bergerak maju sejauh 136 cm dari posisi awal seperti yang terlihat pada gambar 4.8.



Gambar 4. 8 Robot 1 maju 136 cm dari posisi awal

- Robot 2 bergerak maju sejauh 120 cm dari posisi awal seperti yang terlihat pada gambar 4.9.



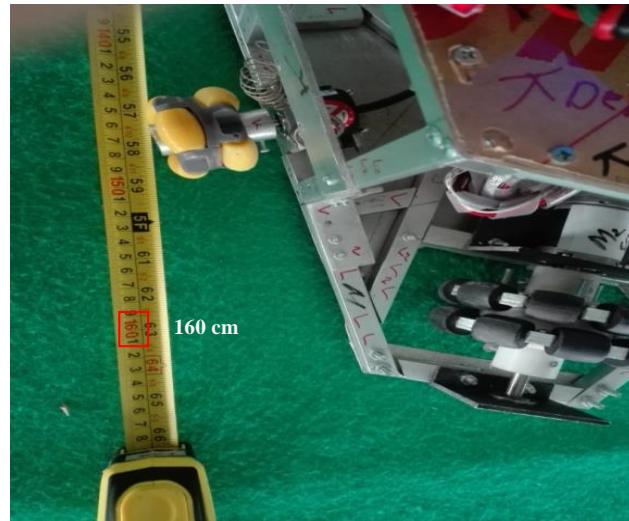
Gambar 4. 9 Robot 2 maju 120 cm dari posisi awal

- Robot 1 bergerak mundur sejauh 294 cm dari posisi terakhir robot 1 bergerak, seperti yang terlihat pada gambar 4.10.



Gambar 4. 10 Robot 1 mundur 294 cm dari posisi terakhir

- Robot 2 bergerak mundur sejauh 160 cm dari posisi terakhir robot 2 bergerak, seperti yang terlihat pada gambar 4.11.



Gambar 4. 11 Robot 2 mundur 160 cm dari posisi terakhir

- Robot 1 bergerak rotasi sebanyak 2 kali sehingga berotasi sebesar 720° , seperti yang terlihat pada gambar 4.12.



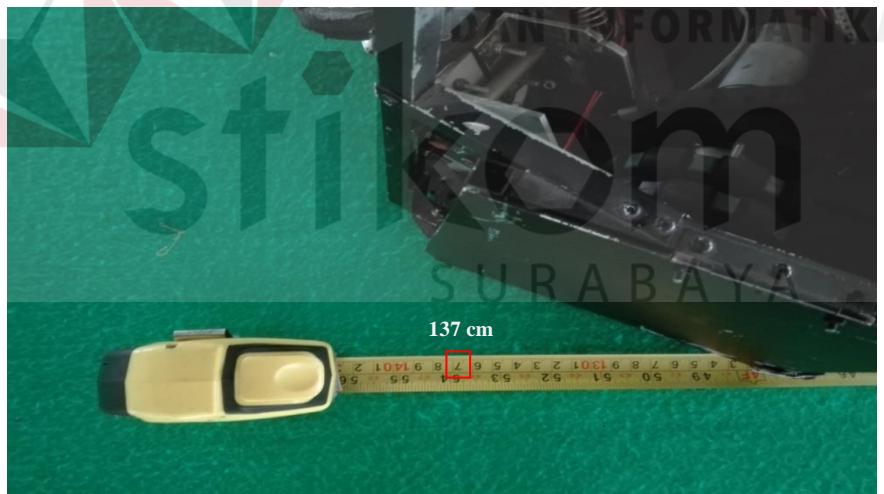
Gambar 4. 12 Robot 1 berotasi sebesar 720°

- Robot 2 bergerak rotasi sebanyak 1 kali sehingga berotasi sebesar 275° , seperti yang terlihat pada gambar 4.13.



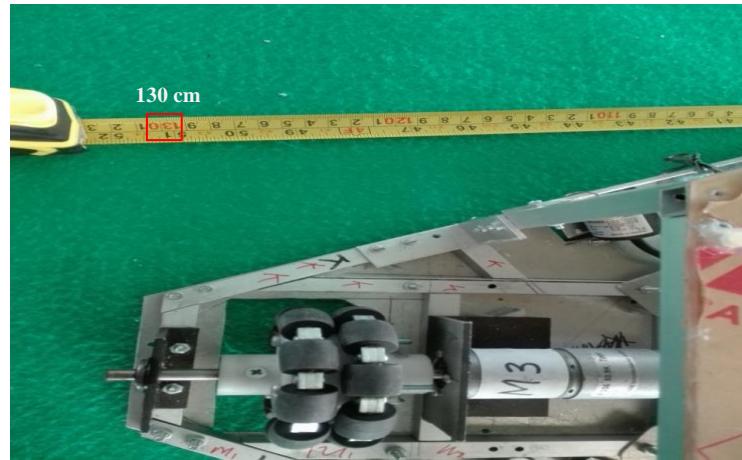
Gambar 4. 13 Robot 2 berotasi sebesar 275°

- k. Hasil pengukuran menggunakan protokol UDP tanpa sinkronisasi :
- Robot 1 bergerak maju sejauh 137 cm dari posisi awal seperti yang terlihat pada gambar 4.14.



Gambar 4. 14 Robot 1 maju 137 cm dari posisi awal

- Robot 2 bergerak maju sejauh 130 cm dari posisi awal seperti yang terlihat pada gambar 4.15.



Gambar 4. 15 Robot 2 maju 130 cm dari posisi awal

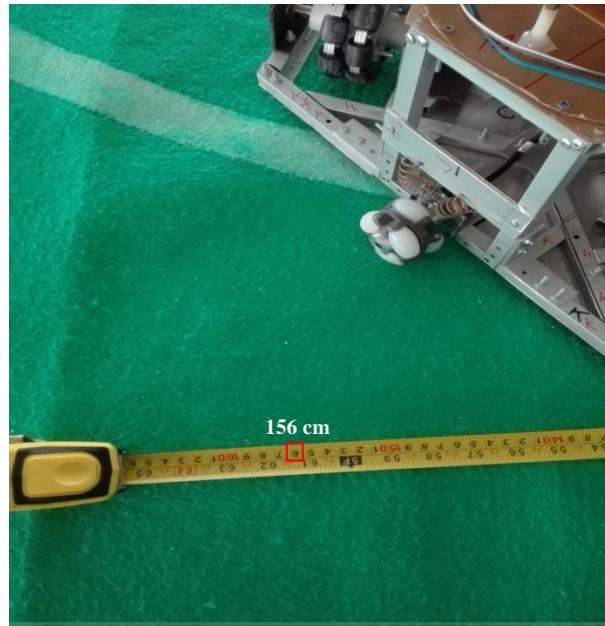
- Robot 1 bergerak mundur sejauh 286 cm dari posisi terakhir robot

1 bergerak, seperti yang terlihat pada gambar 4.16.



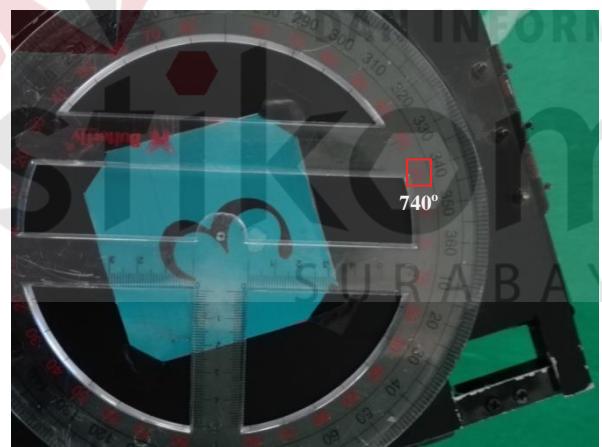
Gambar 4. 16 mundur 286 cm dari posisi terakhir

- Robot 2 bergerak mundur sejauh 156 cm dari posisi terakhir robot 2 bergerak, seperti yang terlihat pada gambar 4.17.



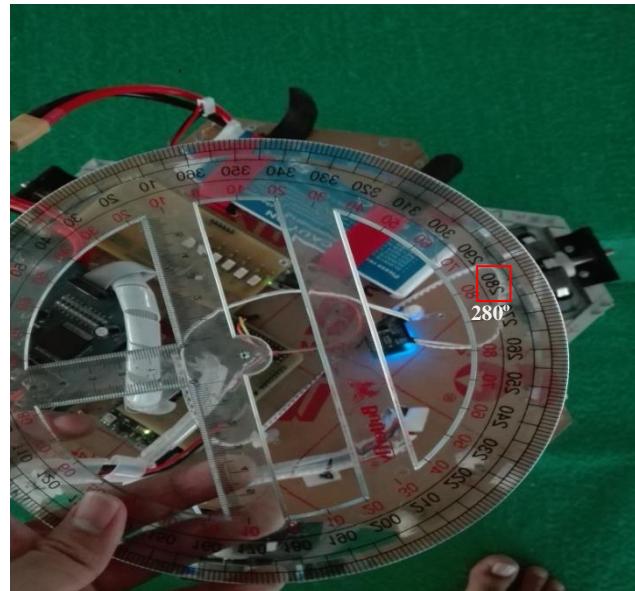
Gambar 4. 17 Robot 2 mundur 156 cm dari posisi terakhir

- Robot 1 bergerak rotasi sebanyak 2 kali sehingga berotasi sebesar 740° , seperti yang terlihat pada gambar 4.18.



Gambar 4. 18 Robot 1 berotasi sebesar 740°

- Robot 2 bergerak rotasi sebanyak 1 kali sehingga berotasi sebesar 280° , seperti yang terlihat pada gambar 4.19.



Gambar 4. 19 Robot 2 berotasi sebesar 280°

Berikut penyajian hasil rekap pengujian reaksi robot dalam bentuk tabel 4.9, 4.10, dan 4.11 untuk babak pertama, tabel 4.12, 4.13, dan 4.14 untuk babak kedua serta tabel 4.15, 4.16, dan 4.17 untuk babak ketiga dibawah ini.



1. Babak Pertama

Tabel 4. 9 Reaksi robot dengan protokol TCP

No.	TCP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	151 cm	297 cm	745°	123 cm	165 cm	275°
2.	109 cm	293 cm	740°	126 cm	163 cm	275°
3.	102 cm	290 cm	740°	120 cm	167 cm	270°
4.	147 cm	295 cm	740°	117 cm	162 cm	285°
5.	158 cm	286 cm	735°	123 cm	170 cm	280°
6.	114 cm	289 cm	745°	125 cm	165 cm	275°
7.	110 cm	295 cm	740°	130 cm	163 cm	280°
8.	153 cm	297 cm	745°	122 cm	166 cm	275°
9.	106 cm	296 cm	745°	120 cm	166 cm	285°
10.	152 cm	300 cm	740°	117 cm	162 cm	270°
11.	108 cm	284 cm	730°	127 cm	162 cm	275°
12.	104 cm	297 cm	740°	126 cm	164 cm	275°
13.	107 cm	295 cm	745°	123 cm	167 cm	285°
14.	148 cm	292 cm	740°	123 cm	165 cm	275°
15.	136 cm	296 cm	745°	122 cm	172 cm	280°
RATA-RATA REAKSI ROBOT						
	127 cm	293,47 cm	741°	122,93 cm	165,27 cm	277,33°

Tabel 4. 10 Reaksi robot dengan protokol UDP Sinkronisasi

No.	UDP Sinkronisasi					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	136 cm	290 cm	720°	120 cm	165 cm	275°
2.	132 cm	287 cm	740°	123 cm	163 cm	272°
3.	132 cm	289 cm	735°	122 cm	166 cm	270°
4.	133 cm	293 cm	740°	126 cm	162 cm	272°
5.	134 cm	286 cm	745°	118 cm	164 cm	275°
6.	132 cm	290 cm	730°	122 cm	163 cm	270°
7.	136 cm	289 cm	725°	125 cm	163 cm	275°
8.	135 cm	294 cm	735°	123 cm	167 cm	275°
9.	133 cm	290 cm	720°	123 cm	163 cm	272°
10.	130 cm	289 cm	745°	120 cm	165 cm	275°
11.	132 cm	283 cm	740°	124 cm	162 cm	275°
12.	128 cm	296 cm	735°	122 cm	165 cm	272°
13.	130 cm	287 cm	740°	128 cm	166 cm	280°
14.	132 cm	289 cm	730°	123 cm	163 cm	275°
15.	131 cm	291 cm	740°	127 cm	162 cm	272°
RATA-RATA REAKSI ROBOT						
	132,4 cm	289,53 cm	734,67°	123,07 cm	163,93 cm	273,67°

Tabel 4. 11 Reaksi robot dengan protokol UDP tanpa sinkronisasi

No.	UDP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	137 cm	286 cm	740°	130 cm	156 cm	280°
2.	135 cm	289 cm	745°	133 cm	159 cm	285°
3.	138 cm	287 cm	745°	128 cm	158 cm	280°
4.	89 cm	286 cm	740°	135 cm	162 cm	275°
5.	137 cm	293 cm	740°	131 cm	160 cm	285°
6.	142 cm	290 cm	745°	130 cm	163 cm	280°
7.	135 cm	289 cm	730°	129 cm	157 cm	280°
8.	137 cm	286 cm	745°	133 cm	158 cm	275°
9.	138 cm	288 cm	740°	130 cm	160 cm	280°
10.	144 cm	289 cm	735°	134 cm	161 cm	285°
11.	145 cm	286 cm	740°	132 cm	161 cm	285°
12.	94 cm	286 cm	735°	129 cm	157 cm	280°
13.	139 cm	287 cm	740°	130 cm	159 cm	280°
14.	136 cm	290 cm	745°	130 cm	162 cm	275°
15.	135 cm	293 cm	740°	133 cm	160 cm	280°
RATA-RATA REAKSI ROBOT						
	132,07 cm	288,33 cm	740,33°	131,13 cm	159,53 cm	280,33°

2. Babak Kedua

Tabel 4. 12 Reaksi robot dengan protokol TCP

No.	TCP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	114 cm	302 cm	740°	120 cm	162 cm	280°
2.	145 cm	297 cm	735°	122 cm	165 cm	285°
3.	102 cm	290 cm	740°	120 cm	166 cm	275°
4.	152 cm	293 cm	745°	119 cm	163 cm	275°
5.	155 cm	288 cm	745°	123 cm	165 cm	280°
6.	137 cm	289 cm	730°	125 cm	169 cm	285°
7.	102 cm	292 cm	740°	134 cm	166 cm	280°
8.	151 cm	304 cm	745°	126 cm	163 cm	275°
9.	157 cm	296 cm	745°	122 cm	162 cm	285°
10.	126 cm	297 cm	735°	119 cm	165 cm	280°
11.	133 cm	284 cm	730°	127 cm	162 cm	275°
12.	118 cm	298 cm	745°	127 cm	160 cm	275°
13.	107 cm	297 cm	745°	129 cm	169 cm	285°
14.	157 cm	286 cm	740°	130 cm	164 cm	280°
15.	107 cm	297 cm	745°	122 cm	169 cm	280°
RATA-RATA REAKSI ROBOT						
	130,87 cm	294 cm	740,33°	124,33 cm	164,67 cm	279,67°

Tabel 4. 13 Reaksi robot dengan protokol UDP dengan Sinkronisasi

No.	UDP Sinkronisasi					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	132 cm	288 cm	740°	122 cm	168 cm	270°
2.	132 cm	290 cm	740°	123 cm	162 cm	272°
3.	134 cm	289 cm	720°	122 cm	166 cm	272°
4.	136 cm	291 cm	735°	125 cm	165 cm	280°
5.	134 cm	287 cm	735°	120 cm	167 cm	275°
6.	138 cm	292 cm	730°	122 cm	163 cm	270°
7.	135 cm	289 cm	740°	123 cm	162 cm	270°
8.	130 cm	296 cm	735°	120 cm	164 cm	272°
9.	133 cm	288 cm	720°	123 cm	163 cm	272°
10.	130 cm	289 cm	740°	126 cm	169 cm	280°
11.	134 cm	283 cm	740°	124 cm	160 cm	275°
12.	131 cm	293 cm	725°	122 cm	162 cm	272°
13.	130 cm	287 cm	745°	126 cm	162 cm	285°
14.	132 cm	295 cm	740°	122 cm	163 cm	275°
15.	135 cm	287 cm	740°	119 cm	165 cm	270°
RATA-RATA REAKSI ROBOT						
	133,07 cm	289,6 cm	735°	122,60 cm	164,07 cm	274°

Tabel 4. 14 Reaksi robot dengan protokol UDP tanpa sinkronisasi

No.	UDP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	132 cm	288 cm	740°	132 cm	162 cm	280°
2.	137 cm	289 cm	740°	132 cm	161 cm	280°
3.	136 cm	283 cm	745°	133 cm	159 cm	280°
4.	146 cm	286 cm	740°	132 cm	162 cm	285°
5.	132 cm	293 cm	735°	130 cm	164 cm	275°
6.	92 cm	298 cm	745°	133 cm	162 cm	275°
7.	135 cm	289 cm	740°	135 cm	163 cm	285°
8.	133 cm	293 cm	745°	128 cm	159 cm	275°
9.	138 cm	294 cm	740°	133 cm	163 cm	275°
10.	148 cm	289 cm	745°	132 cm	163 cm	280°
11.	95 cm	287 cm	740°	137 cm	161 cm	275°
12.	115 cm	296 cm	730°	128 cm	167 cm	280°
13.	136 cm	292 cm	745°	130 cm	157 cm	285°
14.	136 cm	295 cm	745°	130 cm	162 cm	275°
15.	138 cm	289 cm	745°	133 cm	163 cm	275°
RATA-RATA REAKSI ROBOT						
	129,93 cm	290,73 cm	741,33°	131,87 cm	161,87 cm	278,67°

3. Babak Ketiga

Tabel 4. 15 Reaksi robot dengan protokol TCP

No.	TCP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	108 cm	296 cm	745°	122 cm	165 cm	275°
2.	153 cm	296 cm	745°	122 cm	165 cm	275°
3.	137 cm	287 cm	745°	123 cm	166 cm	275°
4.	149 cm	293 cm	735°	122 cm	162 cm	275°
5.	155 cm	294 cm	740°	123 cm	166 cm	285°
6.	119 cm	297 cm	730°	127 cm	165 cm	275°
7.	107 cm	289 cm	740°	121 cm	163 cm	280°
8.	155 cm	297 cm	730°	134 cm	163 cm	275°
9.	152 cm	296 cm	745°	123 cm	162 cm	280°
10.	158 cm	297 cm	745°	123 cm	163 cm	280°
11.	108 cm	293 cm	745°	122 cm	162 cm	280°
12.	128 cm	296 cm	740°	123 cm	167 cm	285°
13.	107 cm	293 cm	735°	120 cm	164 cm	275°
14.	151 cm	289 cm	745°	123 cm	164 cm	275°
15.	106 cm	294 cm	745°	122 cm	168 cm	285°
RATA-RATA REAKSI ROBOT						
	132,87 cm	293,80 cm	740,67°	123,33 cm	164,33 cm	278,33°

Tabel 4. 16 Reaksi robot dengan protokol UDP dengan Sinkronisasi

No.	UDP Sinkronisasi					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	133 cm	286 cm	735°	123 cm	167 cm	272°
2.	132 cm	289 cm	720°	123 cm	164 cm	272°
3.	132 cm	289 cm	720°	123 cm	166 cm	272°
4.	133 cm	286 cm	735°	123 cm	165 cm	275°
5.	135 cm	288 cm	735°	122 cm	167 cm	275°
6.	133 cm	290 cm	740°	126 cm	163 cm	275°
7.	137 cm	289 cm	740°	120 cm	168 cm	270°
8.	132 cm	292 cm	745°	120 cm	162 cm	285°
9.	132 cm	289 cm	720°	123 cm	162 cm	285°
10.	135 cm	289 cm	740°	126 cm	162 cm	280°
11.	133 cm	294 cm	740°	125 cm	167 cm	275°
12.	132 cm	286 cm	720°	124 cm	162 cm	272°
13.	131 cm	287 cm	720°	126 cm	162 cm	280°
14.	136 cm	289 cm	745°	122 cm	160 cm	275°
15.	133 cm	288 cm	740°	125 cm	165 cm	280°
RATA-RATA REAKSI ROBOT						
	133,27 cm	288,73 cm	733°	123,4 cm	164,13 cm	276,2°

Tabel 4. 17 Reaksi robot dengan protokol UDP tanpa sinkronisasi

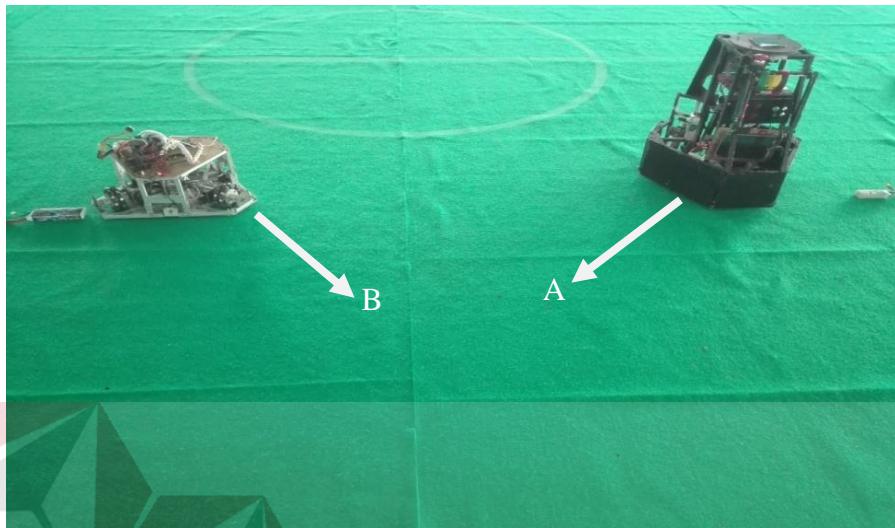
No.	UDP					
	Robot 1			Robot 2		
	Maju Sa < 60	Mundur Sb < 1000	Rotasi Sc < 1000	Maju Sb < 1000	Mundur Sa < 2500	Rotasi Sc < 1300
1.	133 cm	289 cm	745°	133 cm	167 cm	275°
2.	133 cm	289 cm	745°	133 cm	167 cm	285°
3.	136 cm	283 cm	745°	133 cm	162 cm	285°
4.	146 cm	289 cm	735°	137 cm	162 cm	275°
5.	133 cm	293 cm	735°	130 cm	162 cm	280°
6.	97 cm	298 cm	745°	133 cm	162 cm	280°
7.	133 cm	289 cm	740°	137 cm	167 cm	285°
8.	133 cm	293 cm	745°	128 cm	159 cm	275°
9.	138 cm	284 cm	740°	133 cm	163 cm	275°
10.	137 cm	289 cm	740°	138 cm	163 cm	280°
11.	96 cm	287 cm	740°	137 cm	158 cm	280°
12.	126 cm	297 cm	730°	137 cm	167 cm	280°
13.	136 cm	297 cm	740°	141 cm	162 cm	285°
14.	126 cm	294 cm	740°	133 cm	162 cm	285°
15.	133 cm	289 cm	745°	133 cm	167 cm	275°
RATA-RATA REAKSI ROBOT						
	129,07 cm	290,67 cm	740,67°	134,4 cm	163,33 cm	280°

Rata-rata reaksi robot dengan menggunakan protokol UDP sinkronisasi dari ketiga babak adalah untuk robot 1 132.91 cm kondisi maju, 289.29 cm kondisi mundur, dan 734.22° untuk kondisi rotasi. Sedangkan untuk robot 2 123.02 cm kondisi maju, 164.04 cm kondisi mundur, dan 274.62° untuk kondisi rotasi. Rata-rata rekasi robot dengan menggunakan protokol TCP dari ketiga babak adalah untuk robot 1 133 cm kondisi maju, 293.76 cm kondisi mundur, dan 743.33° untuk kondisi rotasi, sedangkan untuk robot 2 124.67 cm kondisi maju, 168 cm kondisi mundur, dan 281.67° untuk kondisi rotasi. Rata-rata reaksi robot dengan menggunakan protokol UDP dari ketiga babak adalah untuk robot 1 sebesar 135 cm kondisi maju, 291.33 cm kondisi mundur, 745° untuk kondisi rotas, sedangkan untuk robot 2 sebesar 131 cm kondisi maju, 163.67 cm kondisi mundur, dan 275° untuk kondisi rotasi.

Terdapat perbedaan dari rata-rata gerakan reaksi robot dari 3 kategori yang diuji pada setiap protokol yang digunakan, namun selisih dari setiap gerakan maju, mundur, maupun berotasi tidak lah besar, hal ini bisa dapat disimpulkan bahwa baik robot 1 maupun robot 2 berhasil menjalankan instruksi maupun menerima setiap data yang dikirimkan oleh *Base Station*. Perbedaan rata-rata reaksi dari setiap gerakan robot dapat dipengaruhi oleh kepresisian mekanik robot itu sendiri karena dalam melakukan penelitian Tugas Akhir ini penulis menggunakan dua robot yang berbeda dimensi dan berbeda motor aktuator.

Dimensi untuk robot penyerang adalah 52x52x60 cm dengan bobot 25kg dan menggunakan motor *planetary gear 36* (PG36) yang mempunyai spesifikasi torsi 6kgfcm, kecepatan 1070RPm, serta *rotary internal* dengan *pulse* 7ppr dengan suplai tegangan 12-24Vdc. Sedangkan dimensi untuk robot bertahan adalah

50x50x60cm dengan bobot 10kg dan menggunakan motor *planetary gear* 36 (PG36) yang mempunyai spesifikasi torsi 13kgfcm, kecepatan 870RPm, serta *rotary* internal dengan *pulse* 7ppr dengan suplai tegangan 12-24Vdc.



Gambar 4. 20 (A) Robot Penyerang, (B) Robot Bertahan

4.4 Pengujian Jarak Jangkauan Media Receiver

Yang dimaksud pengujian jarak jangkauan media receiver dalam Tugas Akhir ini adalah untuk mengukur jarak jangkauan penerimaan robot dalam menerima data dari Base Station, apakah robot dapat menerima semua data dalam area arena pertandingan.

4.4.1. Tujuan Pengujian Jarak Jangkauan Media Receiver

Tujuan dari pengujian ini dilakukan adalah untuk mengetahui apakah media receiver robot dapat menerima data dari Base Station berdasarkan jarak jangkauan dari area arena pertandingan.

4.4.2. Komponen Yang Dibutuhkan Pada Pengujian Jarak Jangkauan Media Receiver

Adapun komponen yang digunakan dalam pengujian ini, antara lain :

1. Akses Point dan *switch*
2. Laptop
3. Robot
4. Wemos
5. Meteran

4.4.3. Prosedur Pengujian Jarak Jangkauan Media *Receiver*

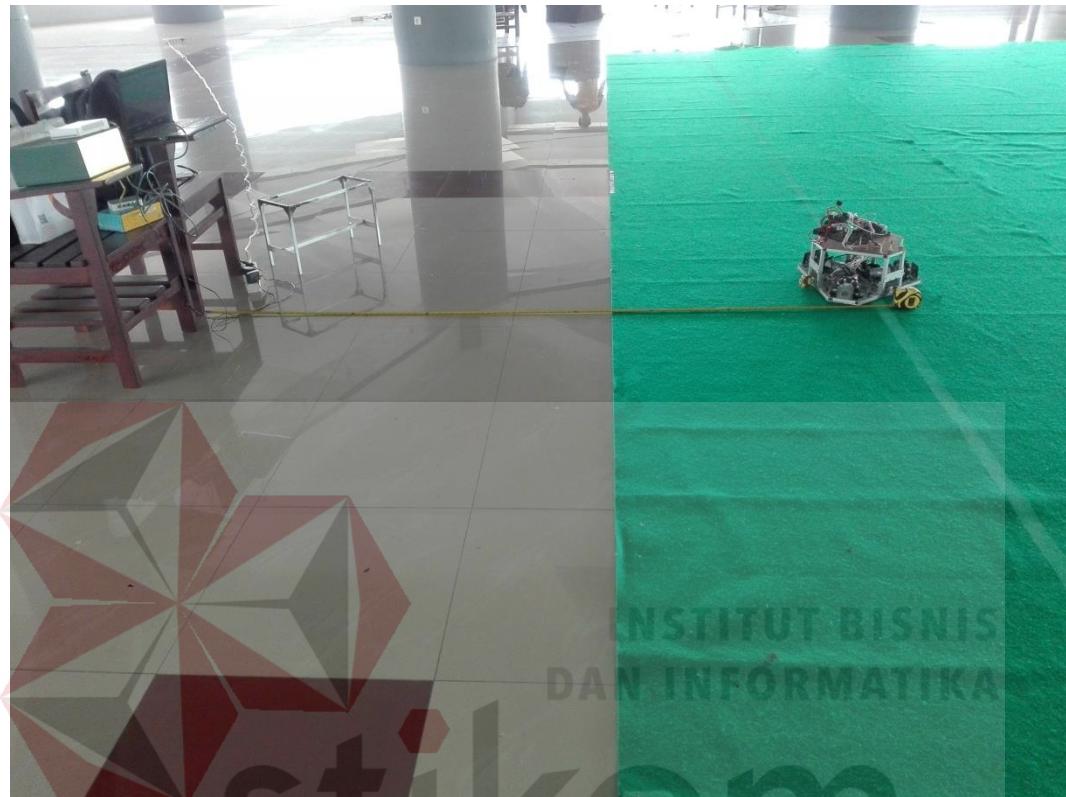
Berikut langkah-langkah pengujian reaksi robot dalam menerima data dari *Base Station*.

1. Menyalakan power robot untuk mode on.
2. Membuka aplikasi *Base Station* dan melakukan koneksi kepada robot.
3. Meletakan robot pada posisi yang telah ditentukan sebagai uji sampel
4. *Base Station* mengirim data instruksi ‘s’ (*start*) kepada robot yang diterima dari *Refree Box*.
5. Mengamati reaksi robot merespon atau tidak dan mengukur setiap peletakan robot pada sampel yang telah ditentukan dari posisi *Base Station* berada.
6. Catat hasil pengukuran.

4.4.4. Hasil Pengujian Jarak Jangkauan Media *Receiver*

Pengujian dilakukan dengan meletakan robot di 15 titik yang berbeda di dalam arena karpet. Letak dari *Base Station* berada di pinggir luar arena karpet dengan jarak 260 cm. Letak dari *Base Station* menjadi titik awal dari pengukuran ke 15 titik sampel yang telah ditentukan. Pengujian yang dilakukan adalah dengan mengirimkan data start (char ‘s’) kepada robot dan melihat apakah robot berhasil menerima data tersebut. Hasil rekap dari pengukuran 15 titik berbeda dengan

menggunakan ketiga protokol disajikan dalam bentuk tabel 4.18, 4.19, dan 4.20 berikut ini.



Gambar 4. 21 Letak posisi *Base Station* yang menjadi titik awal pengukuran

Tabel 4. 18 Jarak Jangkauan Media *Receiver* dengan Menggunakan TCP

No.	TCP		
	Jarak (cm)	Robot 1	Robot 2
1.	287 cm	‘s’ (berhasil)	‘s’ (berhasil)
2.	310 cm	‘s’ (berhasil)	‘s’ (berhasil)
3.	450 cm	‘s’ (berhasil)	‘s’ (berhasil)
4.	496 cm	‘s’ (berhasil)	‘s’ (berhasil)
5.	530 cm	‘s’ (berhasil)	‘s’ (berhasil)
6.	575 cm	‘s’ (berhasil)	‘s’ (berhasil)
7.	600 cm	‘s’ (berhasil)	‘s’ (berhasil)
8.	690 cm	‘s’ (berhasil)	‘s’ (berhasil)
9.	750 cm	‘s’ (berhasil)	‘s’ (berhasil)
10.	820 cm	‘s’ (berhasil)	‘s’ (berhasil)
11.	895 cm	‘s’ (berhasil)	‘s’ (berhasil)
12.	910 cm	‘s’ (berhasil)	‘s’ (berhasil)
13.	950 cm	‘s’ (berhasil)	‘s’ (berhasil)
14.	1120 cm	‘s’ (berhasil)	‘s’ (berhasil)
15.	1200 cm	‘s’ (berhasil)	‘s’ (berhasil)

Tabel 4. 19 Jarak Jangkauan Media *Receiver* dengan Menggunakan UDP Sinkronisasi

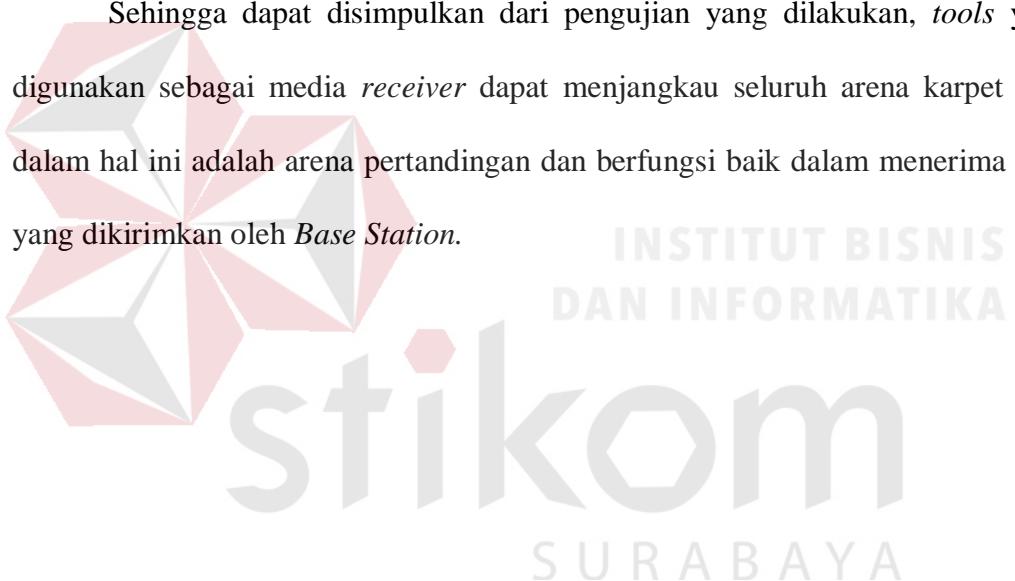
No.	UDP Sinkronisasi		
	Jarak (cm)	Robot 1	Robot 2
1.	287 cm	‘s’ (berhasil)	‘s’ (berhasil)
2.	310 cm	‘s’ (berhasil)	‘s’ (berhasil)
3.	450 cm	‘s’ (berhasil)	‘s’ (berhasil)
4.	496 cm	‘s’ (berhasil)	‘s’ (berhasil)
5.	530 cm	‘s’ (berhasil)	‘s’ (berhasil)
6.	575 cm	‘s’ (berhasil)	‘s’ (berhasil)
7.	600 cm	‘s’ (berhasil)	‘s’ (berhasil)
8.	690 cm	‘s’ (berhasil)	‘s’ (berhasil)
9.	750 cm	‘s’ (berhasil)	‘s’ (berhasil)
10.	820 cm	‘s’ (berhasil)	‘s’ (berhasil)
11.	895 cm	‘s’ (berhasil)	‘s’ (berhasil)
12.	910 cm	‘s’ (berhasil)	‘s’ (berhasil)
13.	950 cm	‘s’ (berhasil)	‘s’ (berhasil)
14.	1120 cm	‘s’ (berhasil)	‘s’ (berhasil)
15.	1200 cm	‘s’ (berhasil)	‘s’ (berhasil)

Tabel 4. 20 Jarak Jangkauan Media *Receiver* dengan Menggunakan UDP

No.	UDP		
	Jarak (cm)	Robot 1	Robot 2
1.	287 cm	‘s’ (berhasil)	‘s’ (berhasil)
2.	310 cm	‘s’ (berhasil)	‘s’ (berhasil)
3.	450 cm	‘s’ (berhasil)	‘s’ (berhasil)
4.	496 cm	‘s’ (berhasil)	‘s’ (berhasil)
5.	530 cm	‘s’ (berhasil)	‘s’ (berhasil)
6.	575 cm	‘s’ (berhasil)	‘s’ (berhasil)
7.	600 cm	‘s’ (berhasil)	‘s’ (berhasil)
8.	690 cm	‘s’ (berhasil)	‘s’ (berhasil)
9.	750 cm	‘s’ (berhasil)	‘s’ (berhasil)
10.	820 cm	‘s’ (berhasil)	‘s’ (berhasil)
11.	895 cm	‘s’ (berhasil)	‘s’ (berhasil)
12.	910 cm	‘s’ (berhasil)	‘s’ (berhasil)
13.	950 cm	‘s’ (berhasil)	‘s’ (berhasil)
14.	1120 cm	‘s’ (berhasil)	‘s’ (berhasil)
15.	1200 cm	‘s’ (berhasil)	‘s’ (berhasil)

Pada pengujian dengan menggunakan protokol TCP yang tersaji pada tabel 4.18, dapat dilihat bahwa robot berhasil menerima data yang dikirimkan oleh *Base Station* dari semua jarak jangkauan arena karpet. Begitu pula dengan menggunakan protokol UDP Sinkronisasi robot juga berhasil menerima data yang dikirimkan oleh *Base Station* dari semua jarak jangkauan area karpet, seperti yang tersaji pada tabel 4.19. Dan dengan menggunakan protokol UDP robot juga berhasil menerima semua data yang dikirimkan oleh *Base Station* seperti yang tersaji pada tabel 4.20.

Sehingga dapat disimpulkan dari pengujian yang dilakukan, *tools* yang digunakan sebagai media *receiver* dapat menjangkau seluruh arena karpet atau dalam hal ini adalah arena pertandingan dan berfungsi baik dalam menerima data yang dikirimkan oleh *Base Station*.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap protokol UDP dengan sinkronisasi dapat ditarik kesimpulan sebagai berikut :

1. Dengan menggunakan *thread sleep* untuk mengatur sinkronisasi waktu penerimaan dan pengiriman data dari kedua robot, maka menghasilkan komunikasi antara *Base Station*, robot 1 dan robot 2 menjadi efektif dan efisien. Hal ini karena saat proses komunikasi berlangsung *Base Station* dapat melayani permintaan robot 1 dan robot 2 yang datang bersamaan secara pararel sehingga tidak ada antrian data yang dikirimkan kepada *Base Station* baik dari robot 1 maupun robot 2 hal ditunjukkan dari besar *packet loss* dan besar *delay* yang mempunyai nilai rata-rata kecil dibanding protokol TCP dan UDP tanpa sinkronisasi.
2. Jika tidak dilakukan sinkronisasi pada protokol UDP maka *Base Station* tidak akan bisa menerima paket data yang datang dari robot 1 dan robot 2 secara bersamaan, selain itu *Base Station* juga harus melakukan *advertisement* dengan mengirimkan data terus menerus kepada robot 1 dan robot 2 untuk memberi tahu lokasinya, hal ini membuat *Base Station* harus mampu melakukan komunikasi dua arah secara pararel.
3. Meskipun UDP tanpa sinkronisasi memiliki besar *delay* yang lebih kecil dari pada UDP sinkronisasi, namun UDP tanpa sinkronisasi pun masih tetap memerlukan perlakuan khusus yaitu memerlukan *thread sleep* pada saat

Base Station melakukan *advertisement* (sebagaimana yang telah dijelaskan pada bab 4). Jika tidak maka UDP tanpa sinkronisasi tidak akan bisa melakukan komunikasi dua arah karena data lain yang dikirimkan oleh *Base Station* maupun data yang dikirimkan oleh robot 1 dan robot 2 akan selalu mengantri sampai *Base Station* tidak mengirimkan data *advertisement* kembali.

4. Berdasarkan data uji *packet loss* dan *delay*, maka urutan protokol komunikasi yang paling efisien dan efektif untuk diterapkan pada KRSBI-

Beroda adalah :

- a. UDP Sinkronisasi
- b. TCP
- c. UDP tanpa sinkronisasi

5.2 Saran

Adapun saran yang dapat penulis berikan kepada peneliti yang ingin mengembangkan penelitian ini selanjutnya adalah sebagai berikut :

1. Penelitian ini hanya berfokus kepada penilaian protokol yang digunakan untuk komunikasi KRSBI-Beroda. Penilaian terhadap tools yang digunakan sebagai media *transmit* maupun *receive* pada robot tidak dilakukan penelitian, sehingga besar kemungkinan peran dari *tools* yang digunakan juga akan berpengaruh terhadap komunikasi yang berlangsung.
2. Diharapkan peneliti yang ingin melakukan penelitian ini selanjutnya dapat mengembangkan strategi komunikasi kedua robot dari pertukaran data yang dilakukan.

DAFTAR PUSTAKA

- Anggraeni, R. T. (2012). Analisis Perbandingan Unjuk Kerja Protokol TCP, UDP, SCTP Menggunakan Simulasi Lalu Lintas Data Multimedia . *jurnal.stikom.edu/jcone* , 2-7.
- 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.
- Atmaja, B. (2017, September 6). "memahami-perbedaan-unicast-broadcast-dan-multicast". Dipetik Juli 11, 2018, dari jemujemu : <https://jemujemu.com/memahami-perbedaan-unicast-broadcast-dan-multicast/>
- Dwiyankuntoko, A. (2009). *Membandingkan Protokol UDP dan TCP*. Komunitas eLearning IlmuKomputer.com.
- Firmansyah, A. (2008, September 2). *Istilah dalam Jaringan Manajemen Telekomunikasi*. Dipetik Agustus 16, 2018, dari Arif Firmansyah Blog: <https://firmansyah2308.wordpress.com/2008/09/02/jaringan-manajemen-telekomunikasi/>
- Jusak. (2016). *Desain dan Analisis Unjuk Kerja Jaringan* (Vol. I). Surabaya: Revka Petra Media Surabaya.
- Kementerian Riset, T. d. (2018). "Tentang KRI". Dipetik Juli 11, 2018, dari <http://kontesrobotindonesia.id/tentang-kri.html>
- KRSBI-B, P. (2017). *Buku Panduan Kontes Robot Sepak Bola Indonesia Devisi (KRSBI BERODA) 2018*. Jakarta: Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia.

Oktavianto, M. M. (2014, June 13). *Quality Of Service*. Dipetik Agustus 16, 2018, dari muhtaroktavianto.blogspot.com: <http://muhtaroktavianto.blogspot.com/2014/06/qos-quality-of-service.html>

Phyong, F. (2010, Oktober 9). *Wireshark, fungsi, dan kegunaannya*. Dipetik Agustus 16, 2018, dari FIYA PHYONG Blogspot: <http://fiyaphyong.blogspot.com/2010/10/wireshark-fungsi-dan-kegunaanya.html>

Santoso, Y. D., Nugroho, S., & Wardana, H. K. (2018). Komunikasi Antar Robot Sepak Bola Beroda Menggunakan UDP Multicast. *Jurnal Teknik dan Ilmu Komputer*, 13.

Sugeng, M. (2010, September). "pengertian dari unicast anycast". Dipetik Juli 11, 2018, dari blogbebasok.blogspot.com: <http://blogbebasok.blogspot.com/2010/09/pengertian-dari-unicast-anycast.html>

