

KOMUNIKASI ANTAR ROBOT SEPAKBOLA BERODA MENGUNAKAN UDP *MULTICAST*

COMMUNICATION BETWEEN WHEELED SOCCER ROBOT USING UDP MULTICAST

Yonathan Darwinto Santoso¹, Saptadi Nugroho², Hartanto Kusuma Wardana³

Fakultas Teknik Elektronika dan Komputer Universitas Kristen Satya Wacana,
Salatiga, Indonesia

¹612014004@student.uksw.edu, ²saptadi_nugroho@yahoo.com,

³hartanto.kusuma@staff.uksw.edu

Abstrak

Kontes Robot Indonesia pada tahun 2017 membuka divisi baru, yaitu Kontes Robot Sepakbola Indonesia Beroda atau bisa disebut KRSBI-B. KRSBI-B adalah perlombaan sepakbola antar robot beroda. Untuk memenangkan pertandingan, setiap tim harus menyiapkan *hardware*, algoritma, dan strategi yang baik. Strategi merupakan aspek yang sangat penting di KRSBI-B. Strategi tidak akan bisa berjalan tanpa adanya komunikasi antar pemainnya. Oleh karena itu, di dalam penelitian ini akan dibahas mengenai komunikasi dan koordinasi antar robot sepakbola beroda. Setiap tim robot minimal menurunkan tiga robot dalam pertandingan, yaitu *Striker*, *Mid Fielder*, dan *Goal Keeper*. Komunikasi dalam pertandingan ini, selain digunakan untuk mengatur strategi, juga untuk mendengarkan perintah aplikasi wasit dari juri, yaitu *Referee Box*. Juri mengharuskan ada satu komputer yang digunakan untuk mem-*forward* pesan *Referee Box* ke semua robot dan mengatur strategi robot, yaitu *Base Station*. *Base Station* akan tersambung dengan *Referee Box* menggunakan TCP. Tiga robot dan *Base Station* akan tersambung menggunakan UDP *Multicast*. *Base Station* akan mengirimkan status *Referee Box* dan setiap robot akan mengirimkan status dirinya. *Referee Box* dan robot akan mengirimkan statusnya masing-masing setiap 100 milidetik ke *multicast group*. Dengan diketahuinya status pertandingan dan status masing-masing robot, maka strategi dapat dilakukan. Berdasarkan hasil pengujian, semua data *Referee Box* dan data masing-masing robot berhasil sampai di *multicast group*. Strategi pun sukses dilakukan, karena komunikasi yang dilakukan berjalan dengan lancar. Dengan demikian, dapat ditarik kesimpulan bahwa komunikasi dan strategi antar robot berhasil dilakukan.

Kata Kunci: robot, sepak bola, multicast, UDP

Abstract

The Indonesian Robot Contest in 2017 opened up a new division: Indonesian Wheeled Soccer Robot Contest (KRSBI-B). KRSBI-B is a soccer rally between wheeled robots. In order to win the game, each team must prepare hardware, algorithms and also a good strategy. Strategy is very important in KRSBI-B. The strategy will not succeed if there is no communication between the players. Therefore, this research discusses the communication and coordination between wheeled soccer robots. Each team plays three robots in the game: Striker, Mid Fielder, and Goal Keeper. Communication in this match, in addition to execute the strategies, is to listen to commands from Referee Box, a referee application from the judges. Also, the judge requires one computer to forward the message from Referee Box to all robots and to set up a robot strategy, called Base Station. The Base Station will be connected to the Referee Box using Transmission Control Protocol (TCP). Meanwhile, three robots and Base Station will be connected using UDP

Multicast. Later, Base Station will inform the Referee Box status. Every robot will inform their status as well. Referee Box and robots will inform their status every 100 milliseconds to the multicast group. By knowing the status of the game and the status of each robot, then the strategy can be executed. The test results show that all the Referee Box data and the data of each robot reached the multicast group successfully. Strategy also succeeded, because the communication was established smoothly. Finally, it is concluded that communication and strategy between robots have been successfully 100% executed.

Keywords: robot, soccer, multicast, UDP

Tanggal Terima Naskah : 15 Desember 2017

Tanggal Persetujuan Naskah : 14 Februari 2018

1. PENDAHULUAN

Pada perkembangan teknologi saat ini, banyak terjadi kemajuan dalam bidang komunikasi antar komputer. Komunikasi antar komputer saat ini banyak digunakan untuk berbagai macam keperluan, salah satunya digunakan pada Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda). Pada KRSBI Beroda komunikasi antar komputer digunakan robot untuk saling bertukar data. Pertukaran data dapat digunakan dalam menyusun strategi dan mendengarkan *Referee Box*. Dalam penelitian ini dibahas mengenai penggunaan sistem komunikasi komputer UDP *Multicast* dan penyusunan algoritma strategi pada robot sepakbola beroda.

Permasalahan yang dihadapi pada KRSBI Beroda adalah bagaimana cara agar tiga robot dan satu *base station* dapat saling berkomunikasi secara efisien, serta bagaimana cara mengatur strategi pertandingannya. Pada penelitian ini dibuat sistem komunikasi antar robot sepak bola beroda menggunakan UDP *Multicast*. Dengan menggunakan UDP *Multicast*, sebuah robot dapat mengirimkan data langsung ke beberapa robot lainnya dan menerima data dari robot lain yang berada di *multicast group* yang sama. Dengan adanya pertukaran data ini, maka setiap robot dapat mengetahui kondisi robot lainnya. Hal ini akan mempermudah dalam mengatur strategi pertandingan.

2. KOMUNIKASI

2.1 *Internet Protocol version 4*

Alamat IPv4 adalah alamat 32-bit yang secara unik dan universal mendefinisikan koneksi perangkat (misalnya komputer atau *router*) ke Internet. Alamat IPv4 unik dalam arti bahwa setiap alamat hanya mendefinisikan satu koneksi ke Internet. Dua perangkat di Internet tidak akan pernah bisa memiliki alamat yang sama pada saat bersamaan. Alamat IPv4 bersifat universal dalam arti bahwa sistem pengalamatan harus diterima oleh *host* manapun yang ingin terhubung ke internet. Panjang alamat IPv4 totalnya adalah 32-bit, dan secara teoritis dapat mengalami hingga empat miliar *host* komputer atau lebih tepatnya 4.294.967.296 *host* di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 (karena terdapat 4 oktet) sehingga nilai maksimal dari alamat IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai *host* yang dapat ditampung adalah $2^{32}=4.294.967.296$ *host* [1],[2],[3],[4].

Tabel 1. IPv4 Datagram Format [1]

<i>HEADER</i> 20 – 60 bytes				<i>DATA</i>
<i>VER</i> 4 bits	<i>HLEN</i> 4 bits	<i>Service</i> 8 bits	<i>Total length</i> 16 bits	
<i>Identification</i> 16 bits		<i>Flags</i> 3bits	<i>Fragmentation offset</i> 13 bits	
<i>Time to live</i> 8 bits	<i>Protocol</i> 8 bits	<i>Header checksum</i> 16 bits		
<i>Source IP address</i>				
<i>Destination IP address</i>				
<i>Option</i>				
-----32 bits-----				

2.2 Transfer Control Protocol (TCP)

TCP adalah protokol proses-ke-proses (program-ke-program). TCP juga menggunakan nomor *port*. Tidak seperti UDP, TCP adalah protokol yang berorientasi koneksi, yaitu menciptakan koneksi *virtual* antara dua TCP untuk pengiriman data. Selain itu, TCP menggunakan mekanisme *flow* dan *error control* pada tingkat *transport*. TCP disebut protokol *transport* berorientasi koneksi dan *reliable* (handal). TCP menambahkan fitur *connection-oriented* dan *reliability* ke layanan IP. TCP juga merupakan protokol yang berorientasi arus. TCP memperbolehkan proses pengiriman mengirimkan data sebagai aliran *byte* dan memungkinkan proses penerimaan untuk mendapatkan data sebagai aliran *byte*. TCP menciptakan lingkungan dimana kedua proses tersebut tampaknya dihubungkan oleh "tabung" imajiner yang membawa datanya ke internet [1],[2],[3],[4].

2.3 User Datagram Protocol

User Datagram Protocol (UDP) disebut protokol *transport* yang *connectionless* (tanpa koneksi) dan *unreliable* (tidak handal). UDP tidak menambahkan apapun ke layanan IP kecuali untuk menyediakan proses ke proses komunikasi bukan komunikasi *host* ke *host*. Selain itu, UDP melakukan pengecekan kesalahan yang sangat terbatas. UDP adalah protokol yang sangat sederhana dengan menggunakan *overhead* yang sangat minim. Pengiriman sebuah pesan kecil dengan menggunakan UDP membutuhkan interaksi antara pengirim dan penerima yang jauh lebih sedikit daripada menggunakan TCP atau SCTP [1],[2],[3],[4],[5],[6],[7],[8].

Tabel 2. User Datagram Format [1]

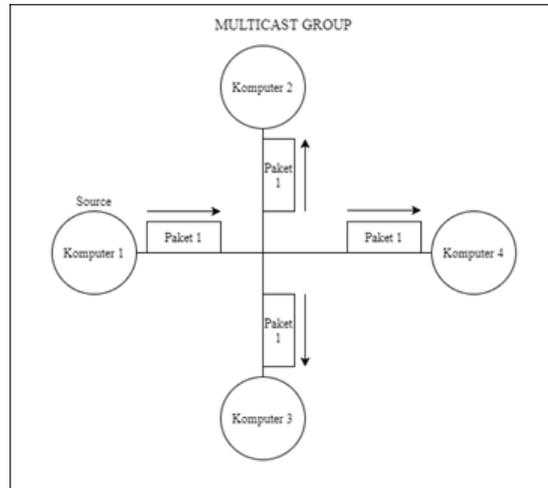
<i>Header</i> 8 bytes		<i>Data</i>
<i>Source port number</i> 16bits	<i>Destination port number</i> 16 bits	
<i>Total length</i> 16 bits	<i>Checksum</i> 16 bits	

Terlihat pada Tabel 2, format data UDP adalah *header* (8 bytes) dan data (65.507 bytes).

2.4 Multicast

Multicast atau *multicasting* adalah sebuah teknik dimana sebuah data dikirimkan melalui jaringan ke sekumpulan komputer yang tergabung ke dalam sebuah grup tertentu, yang disebut sebagai *multicast group*. *Host* dapat menerima transmisi *multicast* dengan mencari dimana lokasinya, seperti halnya ketika kita membuka sebuah stasiun radio

untuk mendengarkan siaran radio. *Multicast* sebenarnya merupakan mekanisme komunikasi *one-to-many*, atau *point-to-multipoint*, dan berbeda dengan cara transmisi *unicast*. Sebuah *multicast group* memiliki sebuah alamat *multicast*, yaitu kelas D dalam alamat IP versi 4. Pada kelas D alamat IP versi 4, alamat yang diresevasikan untuk sebuah *multicast group* adalah 224.0.0.0 hingga 239.255.255.255 [1],[2],[3],[4],[6],[7],[8].



Gambar 1. Multicasting [1]

3. PERANCANGAN SISTEM

3.1 Referee Box

Referee Box adalah aplikasi yang digunakan wasit untuk mengatur pertandingan. Operator *Referee Box* dilakukan oleh asisten wasit.[5].



Gambar 2. Referee Box

Berikut beberapa pesan yang akan dikirimkan *Referee Box* ke Robot melalui *Base Station*:

- a. *Kick Off*: *Kick Off* dilakukan saat awal babak dan setelah terjadi *goal*.
- b. *Free Kick*: *Free Kick* dilakukan jika robot lawan melakukan *illegal attack* (dua robot lawan di depan gawang) dan *illegal defense* (dua robot teman ada di gawang).

- c. *Goal Kick*: *Goal Kick* dilakukan jika lawan membuat bola keluar lapangan lewat garis gawang.
- d. *Throw In*: *Throw In* dilakukan jika status bola keluar.
- e. *Corner*: *Corner* dilakukan jika robot sendiri membuat bola keluar lapangan lewat garis gawang sendiri.
- f. *Penalty*: *Penalty* dilakukan jika lawan melakukan pelanggaran di kotak penaltinya sendiri.
- g. *Drop Ball*: *Drop Ball* dilakukan jika permainan *stuck* atau tidak ada robot yang bergerak.
- h. *Start*: dilakukan untuk memulai pertandingan atau perintah.
- i. *Stop*: dilakukan untuk memberhentikan atau menunda pertandingan.

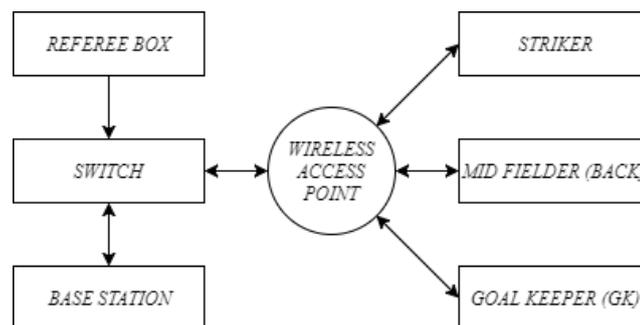
3.2 Base Station

Base Station adalah komputer yang digunakan untuk meneruskan pesan dari *Referee Box* ke robot, dalam hal ini digunakan *multicast group*. *Base Station* dapat digunakan juga untuk memantau kondisi robot [5].

3.3 Peraturan Pertandingan

Berikut peraturan lomba mengenai komunikasi robot yang ada di KRSBI Beroda 2017 [5]:

- a. Kedua AP dan kedua *base station* dan *referee box* terhubung dengan kabel *network* melalui *Switch*.
- b. Masing-masing tim harus mendesain *software* sedemikian sehingga memungkinkan hanya menggunakan satu *base station* untuk mengatur pertandingan.
- c. Perintah dari *Referee box* hanya dikirim ke *base station*. Pengiriman perintah ke robot harus dilakukan oleh *base station*.



Gambar 3. Topologi jaringan komputer antara *Referee Box*, *Base Station*, dan Robot

Sesuai peraturan KRI 2017 maka topologi jaringan komputer saat lomba adalah seperti gambar 3. *Base Station* menerima data dari *Referee Box* melalui *switch* kemudian *Base Station* mengirim data *Referee Box* ke robot melalui *Wireless Access Point* yang terhubung melalui *Switch*. Robot dan *Base Station* dapat saling bertukar data di *multicast group* melalui *Wireless Access Point* [5].

3.4 Sistem Komunikasi antara *Referee Box* dengan *Base Station*

Pada jaringan komputer di KRSBI-B 2017, *Base Station* dihubungkan dengan *Referee Box* melalui *switch*. *Base Station* akan terus menerus mendengar *Referee Box* menggunakan TCP sebagai *client* dan *Referee Box* sebagai *server*, selanjutnya data dari

Referee Box tersebut akan diteruskan ke robot melalui *UDP Multicast* [3],[4],[6],[7],[8],[9],[10].

Langkah kerja algoritma TCP:

- a. Inisialisasi *port* dan alamat IP *Referee Box*.
- b. Inisialisasi struktur *sockaddr_in*, struktur *sockaddr_in* adalah struktur yang berisi alamat internet. struktur ini didefinisikan di *library <netinet/in.h>*.
- c. Inisialisasi *socket*.
- d. Buat *socket* TCP.
- e. Mengisi komponen struktur *sockaddr_in*.
- f. Membangun koneksi dengan *server (Referee Box)*.
- g. Mendengarkan dan menerima pesan.

3.5 Sistem Komunikasi antara *Base Station* dengan Robot

Setiap robot menjalankan peran sebagai pengirim dan penerima data sedangkan *Base Station* menjalankan peran sebagai pendengar *Referee Box*, pengirim, dan penerima data. Robot maupun *Base Station* menjalankan semua perannya secara *parallel* [3],[4],[6],[7],[8],[9],[10].

Langkah kerja algoritma *UDP Multicast Sender*:

- a. Inisialisasi alamat IP dan port dari *multicast group*.
- b. Inisialisasi struktur *sockaddr_in*.
- c. Inisialisasi *socket*.
- d. Buat *socket* UDP.
- e. Mengisi komponen struktur *sockaddr_in*, tapi agar lebih aman, struktur *sockaddr_in* nya dikosongkan dahulu. Baru diisi.
- f. Kirim pesan.

Langkah kerja algoritma *UDP Multicast Listener*:

- a. Inisialisasi alamat IP dan *port* dari *multicast group*.
- b. Inisialisasi struktur *sockaddr_in*.
- c. Inisialisasi *socket*.
- d. Inisialisasi struktur *ip_mreq*.
- e. Buat *socket* UDP.
- f. Mengatur opsi *socket*, agar banyak *socket* dapat menggunakan satu *port*.
- g. Mengisi komponen struktur *sockaddr_in*, tapi agar lebih aman, struktur *sockaddr_in* nya dikosongkan dahulu. Baru diisi.
- h. Menggabungkan alamat IPv4 dengan port menjadi sebuah alamat protokol lokal.
- i. Mempersiapkan struktur *ip_mreq*.
- j. Mengatur opsi *socket* agar dapat bergabung di *multicast group*.
- k. Dengar pesan.

3.6 Sistem pengiriman data oleh Robot dan *Base Station* ke *Multicast Group*

Paket yang dikirimkan oleh robot maupun *Base Station* menggunakan format Header (xx) + id + pesan. *Header* digunakan sebagai penanda pesan baru, sedangkan id adalah penanda pengirim pesan. Robot maupun *Base Station* akan mengirim pesan terus menerus setiap 100 mili detik, tidak ada *acknowledge*-nya. Cara ini dilakukan untuk berjaga-jaga jika ada robot yang tiba-tiba mati di tengah jalan dan saat dihidupkan kembali, robot tersebut tidak mengetahui kondisi robot lain dan *Base Station*. *Striker* mengirim data lihat bola dekat kanan atau kiri ke kiper agar jika kiper tidak melihat bola maka kiper masih bisa mengetahui posisi bola di kanan atau di kiri.

Tabel 3. Daftar data robot dan *Base Station* yang akan dikirim ke *multicast group*

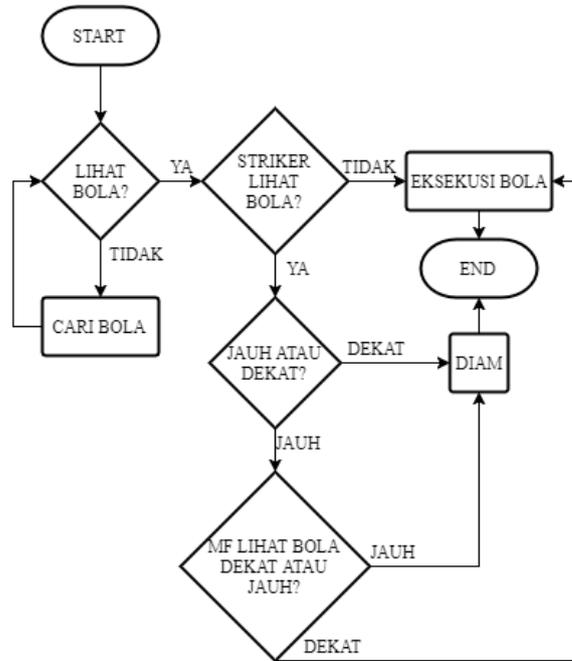
Robot	Header	ID	Data	Pesan	Pesan yang dikirim
<i>Striker</i>	xx	1	Tidak lihat bola	nb	xx1nb
			Lihat bola jauh	bj	xx1bj
			Lihat bola dekat kanan	bd1	xx1bd1
			Lihat bola dekat kiri	bd2	xx1bd2
<i>Mid Fielder</i>	xx	2	Tidak lihat bola	nb	xx2nb
			Lihat bola jauh	bj	xx2bj
			Lihat bola dekat	bd	xx2bd
<i>Base Station</i>	xx	@	<i>Start</i>	s	xx@s
			<i>Stop</i>	S	xx@S
			<i>Drop Ball</i>	N	xx@N
			<i>Park</i>	L	xx@L
			<i>Cyan Kick Off</i>	K	xx@K
			<i>Cyan Free Kick</i>	F	xx@F
			<i>Cyan Goal Kick</i>	G	xx@G
			<i>Cyan Throw In</i>	T	xx@T
			<i>Cyan Corner</i>	C	xx@C
			<i>Cyan Penalty</i>	P	xx@P
			<i>Cyan Goal</i>	A	xx@A
			<i>Magenta Kick Off</i>	k	xx@
			<i>Magenta Free Kick</i>	f	xx@f
			<i>Magenta Goal Kick</i>	g	xx@g
<i>Magenta Throw In</i>	t	xx@t			

3.7 Strategi Robot

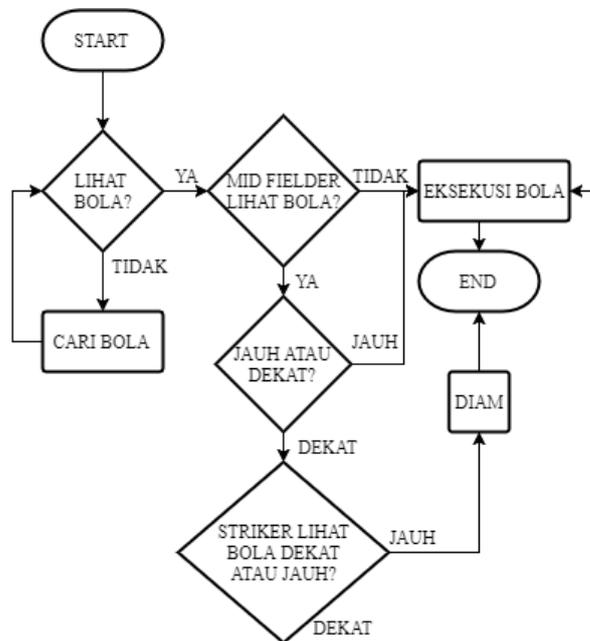
Striker adalah penyerang atau pencetak gol yang utama, sedangkan *Mid Fielder* adalah pembantu *Striker* dalam mencetak gol. Jadi yang diutamakan dalam mencetak gol adalah *Striker*, sedangkan *Mid Fielder* hanya mencetak gol jika *Striker* tidak mendapat atau melihat bola. *Goal Keeper* adalah penjaga gawang yang hanya bertugas menjaga gawang sendiri dan tidak dapat maju untuk mencetak gol. Penempatan *Striker* pada awal pertandingan adalah di bagian kiri lapangan dekat dengan bola, sedangkan *Mid Fielder* berada di bagian kanan lapangan dan tidak terlalu dekat dengan bola. Penempatan *Goal Keeper* ada di depan gawang sendiri. Pada tabel 4 terlihat, jika *striker* dan mf sama-sama tidak lihat bola maka tidak ada yang akan mengeksekusi bola, tapi jika *striker* dan mf sama-sama lihat bola dalam jarak dekat maka yang berhak mengambil bola adalah *striker* karena prioritas utamanya adalah *striker*. Demikian juga dengan kondisi lainnya.

Tabel 4. Strategi Robot

EKSEKUSI BOLA		STRIKER		
		TIDAK LIHAT BOLA	LIHAT BOLA DEKAT	LIHAT BOLA JAUH
Mid Fielder (MF)	TIDAK LIHAT BOLA	-	STRIKER	STRIKER
	LIHAT BOLA DEKAT	MF	STRIKER	MF
	LIHAT BOLA JAUH	MF	STRIKER	STRIKER



Gambar 4. Algoritma Strategi Mid Fielder



Gambar 5. Algoritma Strategi Striker

4. PENGUJIAN DAN ANALISIS

4.1 Penggunaan Data Uji

Data yang digunakan untuk menguji sistem adalah data dari *Referee Box* dan data dari robot. Semua data tersebut akan dikirimkan ke *multicast group*.

a. Data *Referee Box*

Data dari *Referee Box* didapat saat menekan salah satu tombol perintah yang ada di aplikasi *Referee Box*. Setelah menekan salah satu tombol perintah, maka aplikasi *Referee Box* akan langsung mengirimkan data berupa satu *character* ke *client* yang terhubung dengan aplikasi *Referee Box* tersebut. *Client* dalam hal ini adalah *Base Station*. Setelah *Base Station* mendapat data dari *Referee Box*, kemudian *Base Station* akan langsung mengirimkan data tersebut ke *multicast group*.

b. Data Robot

Data robot yang dikirimkan ke *multicast group* adalah data lihat bola dan data jarak bola dengan robot. Khusus untuk *Striker* ada tambahan data tentang posisi *y* dari *Striker* di lapangan. Data *y* dari *Striker* digunakan oleh kiper untuk mengetahui posisi bola ada di mana. Data lihat bola dari robot didapat dari data kamera yang ada di robot. Jika di *frame* yang ditangkap oleh kamera terdapat *pixel orange* dan berbentuk lingkaran, maka hal tersebut menandakan ada bola di *frame* tersebut. Untuk data jarak bola terhadap robot didapat dari nilai servo kamera, jika 0° sampai 45° maka jaraknya dekat dan jika 46° sampai 90° maka jaraknya jauh. Untuk data posisi *y* dari robot *Striker* dari *encoder motor*.

4.2 Hasil Pengujian dan Analisis

Semua data pengujian akan diuji apakah sampai ke tujuan atau tidak.

a. Data *Referee Box*

Data dari *Referee Box* ditujukan ke robot agar robot mengetahui perintah dari juri. Data dari *Referee Box* tidak boleh langsung dikirimkan ke robot, tapi harus melalui *Base Station* terlebih dahulu, Seperti yang terlihat di gambar 6.



Gambar 6. Alur pengiriman data dari *Referee Box* ke Robot

Tabel 5. Paket yang tertangkap oleh *wireshark*

No	Waktu	Sumber	Tujuan	Protokol	Panjang	Info
1	52.8	172.16.11 0.5	172.16.110. 1	TCP	67	PSH ,AC K
2	52.8	172.16.11 0.1	172.16.110. 5	TCP	66	AC K
3	52.8	172.16.11 0.1	224.16.32.1 10	UDP	52	

Tabel 6. Deskripsi paket no 1

67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
Ethernet II, Src: GemtekTe_1f : 3f : d1 (ac : 81 : 12 : 1f : 3f : d1), Dst: Azurewav_a9 : 67 : 13 (6c : 71 : d9 : a9 : 67 : 13)

Tabel 6. Deskripsi paket no 1 (Lanjutan)

Internet Protocol Version 4, Src: 172.16.110.5, Dst: 172.16.110.1
Transmission Control Protocol, Src Port: 28097, Dst Port: 36774, Seq: 2, Ack: 1, Len: 1
Data (1 byte)
Data: 53
[Length: 1]

Tabel 7. Deskripsi paket no 3

52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface 0
Ethernet II, Src: Azurewav_a9 : 67 : 13 (6c : 71 : d9 : a9 : 67 : 13), Dst: IPv4mcast_10 : 20 : 6e (01 : 00 : 5e : 10 : 20 : 6e)
Internet Protocol Version 4, Src: 172.16.110.1, Dst: 224.16.32.110
User Datagram Protocol, Src Port: 53663, Dst Port: 11968
Data (10 bytes)
Data: 78784053000000000000
[Length: 10]

Seperti terlihat pada gambar 6, pesan dari *Referee Box* tidak boleh langsung ditujukan ke robot, melainkan melalui *Base Station* dahulu. Jadi pada tabel 5 ada tiga paket yang tertangkap, paket nomor 1 merupakan pesan dari *Referee Box* (172.16.110.5) menuju ke *Base Station* (172.16.110.1). Paket nomor 1 berisi pesan perintah dan ACK, adanya ACK karena protokol yang digunakan adalah TCP. Paket nomor 2 adalah balasan ACK yang dikirimkan *Base Station* menuju *Referee Box*. Paket nomor 3 adalah paket dari *Base Station* (172.16.110.1) menuju *multicast group* (224.16.32.110).

b. Data Robot

Data dari robot langsung dikirimkan ke *multicast group*.

Tabel 8. Paket yang tertangkap oleh *wireshark*

No	Waktu	Sumber	Tujuan	Protokol	Panjang	Info
4	52.8	172.16.110.2	224.16.32.110	UDP	52	PSH,ACK
5	52.8	172.16.110.3	224.16.32.110	UDP	52	ACK

Tabel 9. Deskripsi paket no 4

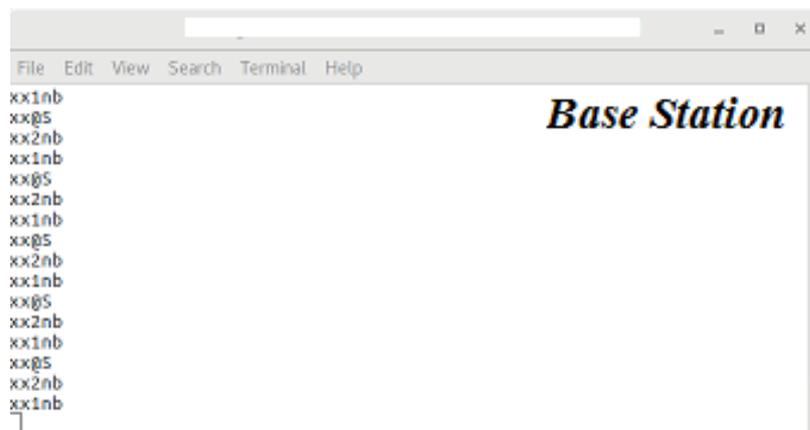
52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface 0
Ethernet II, Src: Raspberr_f8 : 2d : 05 (b8 : 27 : eb : f8 : 2d : 05), Dst: : Azurewav_a9 : 67 : 13 (6c : 71 : d9 : a9 : 67 : 13)
Internet Protocol Version 4, Src: 172.16.110.2, Dst: 224.16.32.110
User Datagram Protocol, Src Port: 54406, Dst Port: 11968
Data (10 bytes)
Data: 7878316e620000000000
[Length: 10]

Tabel 10. Deskripsi paket no 5

52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface 0
Ethernet II, Src: Raspberr_35 : 01 : 7e (b8 : 27 : eb : 35 : 01 : 7e), Dst: : Azurewav_a9 : 67 : 13 (6c : 71 : d9 : a9 : 67 : 13)
Internet Protocol Version 4, Src: 172.16.110.2, Dst: 224.16.32.110
User Datagram Protocol, Src Port: 54406, Dst Port: 11968
Data (10 bytes)
Data: 7878326e620000000000
[Length: 10]

Pada tabel 8 ada dua paket yang tertangkap oleh *wireshark*, paket nomor 4 merupakan paket yang dikirimkan *Striker* (172.16.110.2) menuju *multicast group* (224.16.32.110). Paket nomor 5 merupakan paket yang dikirimkan *Mid Fielder* (172.16.110.2) menuju *multicast group* (224.16.32.110).

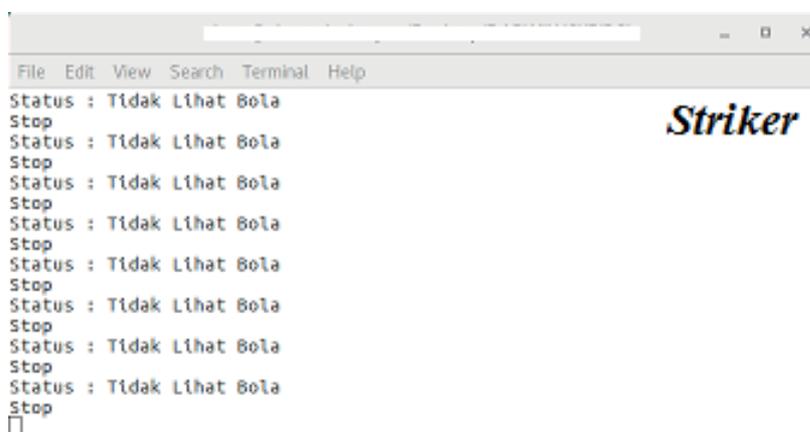
c. Data *multicast group* yang terbaca di terminal masing-masing robot dan *Base Station*



```

File Edit View Search Terminal Help
xx1nb
xx05
xx2nb
xx1nb

```

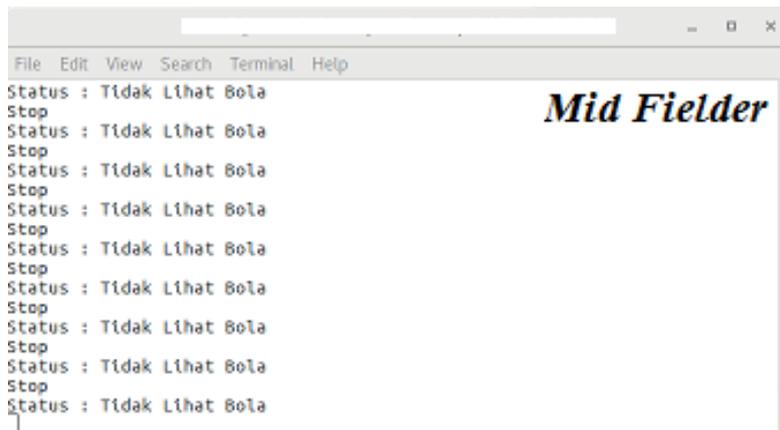
Gambar 7. Terminal *Base Station*


```

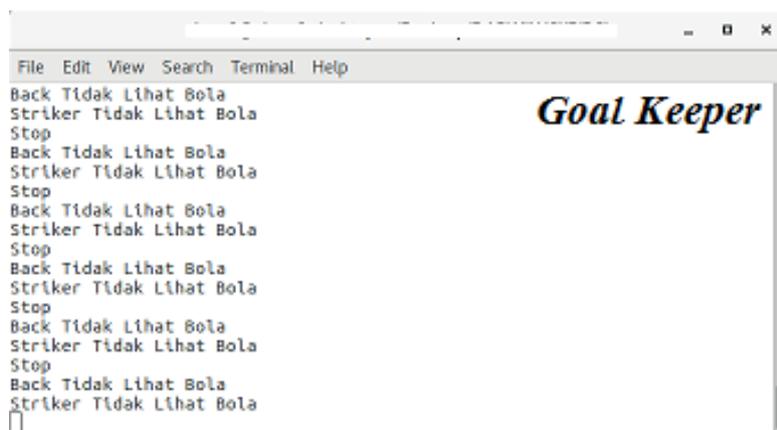
File Edit View Search Terminal Help
Status : Tidak Lihat Bola
Stop

```

Gambar 8. Terminal *Striker*



Gambar 9. Terminal *Mid Fielder*



Gambar 10. Terminal *Goal Keeper*

5. KESIMPULAN

Dari hasil pengujian dan analisis sistem ini, dapat disimpulkan sebagai berikut:

- a. Data dari *Referee Box* berhasil sampai ke semua robot melalui *Base Station*.
- b. Data dari semua robot dan *Base Station* berhasil sampai di *multicast group*.
- c. Data yang ada di *multicast group* dapat terbaca oleh semua anggota *multicast group*.

REFERENSI

- [1] Behrouz A. Forouzan. 2007. *Data Communication And Networking*, Fourth Edition, McGraw-Hill, 1221 Avenue of the Americas: New York, NY 10020.
- [2] Douglas E. Comer. "*Internetworking With TCP/IP Vol I : Principles, Protocols, and Architecture Second Edition*", PRENTICE HALL, Englewood Cliffs, New Jersey, NJ 07632, 1991.
- [3] Douglas E. Comer and David L. Stevens. "*Internetworking With TCP/IP Vol II : Design, Implementation, and Internals Second Edition*". PRENTICE HALL, Englewood Cliffs, New Jersey, NJ 07632, 1994.
- [4] William Stallings. "*Data and Computer Communications*", Pearson Education International, Upper Saddle River, New Jersey, NJ 07458, 2007.
- [5] Panitia KRI. 2017. *Panduan KRSBI Beroda 2017*. Ristekdikti: Jakarta.

- [6] Amit Keswani, Chemakura Baba Kumar Reddy, Girish Dhameja, Moqtar Ahmed Syed, Vinod Devarajan, and Navroop Singh. *Multi Robot Communication*. Department of Computer Science Lamar University: Texas, TX 77710.
- [7] Saumitra M. Das, Y. Charlie Hu, C.S. George Lee, and Yung-Hsiang Lu. “*An Efficient Group Communication Protocol For Mobile Robot*”, School of Electrical and Computer Engineering Purdue University, West Lafayette, Indiana, IN 47907-2035, 2005.
- [8] Saumitra M. Das, Y. Charlie Hu, C. S. George Lee, and Yung-Hsiang Lu. “*Mobility-Aware Ad Hoc Routing Protocols for Networking Mobile Robot Teams*”. The National Science Foundation under Grant IIS-0329061.
- [9] Samuel Alvin Hutama, Saptadi Nugroho, Darmawan Utomo. 2016. “*Features Deletion on Multiple Object Detection*”. SWCU: Salatiga.
- [10] Oei Kurniawan Utomo, Daniel Santoso, Saptadi Nugroho. 2015. “*Location and Position Determination Algorithm for Humanoid Soccer Robot*”. SWCU: Salatiga.