

# **Implementasi Komunikasi Antar Robot Pada Team Barelang-FC**

**TUGAS AKHIR**

Oleh :

**Arif Muspita 3311201086**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM  
BATAM  
2015**

## **HALAMAN PENGESAHAN**

### **Implementasi Komunikasi Antar Robot Pada Team Barelang-FC**

**Oleh :**

**Arif Muspita (3311201086)**

Tugas Akhir ini telah diterima dan disahkan sebagai persyaratan untuk  
memperoleh gelar

Ahli Madya

Di

**PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BATAM**

Batam, 29 Januari 2015

Disetujui oleh:

Pembimbing,

**Nur Cahyono K, S.Si., M.T**

**NIK. 106044**

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311201086

Nama : Arif Muspita

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

### **“IMPLEMENTASI KOMUNIKASI ANTAR ROBOT PADA TEAM BARELANG-FC”**

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 29 Januari 2015

**Arif Muspita**

3311201086

## KATA PENGANTAR

Puji Syukur Kehadirat Allah SWT atas rahmat dan karunianya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Implementasi Komunikasi Antar Robot Pada Team Barelang-FC”. Penulis berharap program dapat bermanfaat dalam mempermudah dalam mendukung perkembangan robot sepak bola. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah memberikan bantuan baik secara langsung maupun tidak langsung dalam penyelesaian proyek ini, antara lain:

1. Allah SWT, atas anugrah yang telah diberikan kepada penulis
2. Orang tua yang selalu memberikan dukungan moril maupun materil
3. Bapak Dr. Priyono Eko Sanyoto, selaku Direktur Politeknik Negeri Batam.
4. Ibu Hilda Widyastuti, S.T., M.T selaku Ketua Jurusan Teknik Informatika
5. Ibu Meyti Eka Apriyani, MT selaku Ketua Program Studi Jurusan Informatika
6. Bapak Nur Cahyono K, S.Si., M.T selaku pembimbing tugas akhir.
7. Bapak Eko Rudiawan S.ST selaku pembimbing robot divisi sepak bola.
8. Bapak Dwi Ely Kurniawan, S.Pd., M.Kom selaku Dosen Pengampun Tugas Akhir.
9. Ibu Yeni Rokhayati, S.Si., M.Sc selaku dosen wali
10. Seluruh Dosen-dosen Teknik Informatika Politeknik Negeri Batam.
11. Seluruh Tim Robot Politeknik Batam yang selalu membantu dalam riset.
12. Seluruh Alumni Tim Robot Politeknik Batam yang ikut membimbing dalam proses pembuatan Tugas Akhir.
13. Seluruh teman-teman yang telah membantu atas terselesaikannya buku laporan ini.

Penulis menyadari sepenuhnya dalam pembuatan sistem maupun laporan ini masih banyak terdapat kekurangan, maka saran dan masukan yang bersifat membangun sangat diharapkan demi pengembangan sistem ini dimasa yang akan datang.

Batam, 29 Januari 2015

Penulis,

## **ABSTRAK**

Robot merupakan benda bergerak yang dibuat untuk pembelajaran ataupun membantu pekerjaan manusia. Di Politeknik Negeri Batam sendiri, memiliki Tim Robot yang bernama Barelang. Jika dilihat dari segi kekurangan khususnya robot sepak bola, masih terdapat kekurangan seperti komunikasi antar robot, penentuan peran robot, menentukan strategi, lambatnya eksekusi perintah, konfigurasi cahaya yang masih manual dan masih banyak lagi masalah yang lainnya. Dari berbagai masalah tersebut jika dilihat pada robot sepak bola atau robot berkelompok, masalah yang paling dasar adalah komunikasi antar robot. Dalam melakukan komunikasi antar robot adapun media yang digunakan untuk komunikasi yaitu WLAN, dengan tujuan yang diharapkan adalah robot mampu melakukan komunikasi dengan robot lainnya dan bergerak sesuai strategi. Maka dari itu penambahan program komunikasi pada sistem robot berguna untuk mengurangi kesalahan dalam perebutan bola, tabrakan antar robot dan mendukung strategi tertentu.

Kata kunci : Robot, Komunikasi

## ***ABSTRACT***

The robot is moving objects created for learning or helping people work. In Batam Polytechnic, have Team Robot and the named is Barelang. When viewed in terms of shortages in particular robot soccer, there are have some problem such as communication between the robot, the determination of the robot, to define a strategy, the slow execution, the light is still manual configuration and many other problems. Of the various problems if viewed on a soccer robot or robot groups, the most basic problem is the communication between robots. In the communication between the robot while the medium used for communication is WLAN, with the expected goal is the robot is able to communicate with other robots and move according to the strategy. The addition of the communications program robotic systems are useful to reduce errors in the taking for the ball, collisions between robots and support specific strategies.

Keywords: Robot, Communications

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
HALAMAN PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vi
DAFTAR ISI .....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL .....	x
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI.....	4
2.1 Linux.....	4
2.2 C++ .....	4
2.3 WLAN .....	5
2.3.1 Spesifikasi WLAN.....	5
2.3.2 Kelebihan dan kekurangan WLAN .....	5
2.4 Jenis Komunikasi dan Perbandingannya .....	6
2.5 Komunikasi Antar Robot .....	6
2.6 SBC Fit-PC 2i .....	7
2.7 CM730 .....	8
2.8 Sensor.....	8
2.8.1 Sensor <i>Gyroscope</i> (3-Axis LYPR540AH) .....	8
2.8.2 Sensor <i>Accelerometer</i> (3-Axis ADXL335) .....	9
2.8.3 <i>Camera</i> (C920).....	9
2.9 Aktuator (MX-28).....	10
BAB III ANALISIS DAN PERANCANGAN.....	11
3.1 Deskripsi Umum Sistem .....	11
3.2 Analisis Kebutuhan Sistem.....	12

3.2.1	Analisis kebutuhan perangkat keras robot.....	12
3.2.2	Analisis kebutuhan perangkat lunak robot .....	12
3.2.3	Analisis topologi yang digunakan saat komunikasi .....	13
3.3	Perancangan program .....	13
3.4	Flow chart Pergerakan Robot .....	14
3.4.1	<i>FlowChart</i> Pada Robot1 dan Robot2 <i>Mode Strategi Default</i> .....	15
3.4.2	<i>FlowChart</i> Pada Robot1 dan Robot2 <i>Mode Strategi Menyerang</i> .....	16
3.5	Strategi .....	17
3.5.1	Strategi Menyerang.....	17
3.5.2	Strategi <i>Default</i> .....	18
BAB IV IMPLEMENTASI DAN HASIL .....		19
4.1	Implementasi Program .....	19
4.1.1	Cara Implementasi Program Robot .....	19
4.2	Implementasi Program Komunikasi dan Strategi .....	22
4.2.1	Implementasi Program Komunikasi pada Robot1 .....	22
4.2.2	Implementasi Program Strategi pada Robot1 .....	24
4.2.3	Implementasi Program Komunikasi pada Robot2.....	26
4.2.4	Implementasi Program Strategi pada Robot2.....	27
4.3	Hasil Percobaan .....	31
4.3.1	Hasil Percobaan Komunikasi.....	31
4.3.2	Hasil Percobaan Strategi.....	37
BAB V KESIMPULAN DAN SARAN .....		38
5.1	Kesimpulan .....	38
5.2	Saran .....	38
DAFTAR PUSTAKA.....		39
LAMPIRAN .....		1.1-1
1.1	ROBOT 1 .....	1.1-1
1.2	ROBOT 2 .....	1.2-1
BIOGRAFI PENULIS .....		A

## DAFTAR GAMBAR

Gambar 1 Tampilan Fit-PC .....	7
Gambar 2 Tampilan CM-730 .....	8
Gambar 3 Sensor <i>Gyroscope</i> pada CM-730 .....	9
Gambar 4 Sensor <i>Accelerometer</i> pada CM-730 .....	9
Gambar 5 Kamera C920 .....	10
Gambar 6 Servo MX-28 .....	10
Gambar 7 Deskripsi Umum Sistem .....	11
Gambar 8 Topologi <i>Star</i> .....	13
Gambar 9 Blok Diagram Sistem Robot .....	14
Gambar 10 <i>Flowchart</i> Robot1 dan Robot2 <i>Mode Strategi Default</i> .....	15
Gambar 11 <i>Flowchart</i> Robot1 dan Robot2 <i>Mode Strategi Menyerang</i> .....	16
Gambar 12 Strategi Penyerangan .....	17
Gambar 13 Strategi <i>Default</i> .....	18
Gambar 14 Putty .....	19
Gambar 15 Proses Login .....	20
Gambar 16 Direktori Program Robot .....	20
Gambar 17 Void Pengiriman Komunikasi .....	21

## DAFTAR TABEL

Tabel 1 Spesifikasi WLAN.....	5
Tabel 2 Perbandingan Jenis Komunikasi .....	6
Tabel 3 Spesifikasi SBC-FitPC2i .....	7
Tabel 4 Spesifikasi CM-730 .....	8
Tabel 5 Spesifikasi MX-28.....	10
Tabel 6 Analisis Kebutuhan Perangkat Keras Robot .....	12
Tabel 7 Analisis Kebutuhan Perangkat Lunak Robot .....	12
Tabel 8 Percobaan Robot1 Dapat bola Robot2 Dapat Bola .....	31
Tabel 9 Percobaan Robot1 hilang bola Robot2 dapat bola .....	32
Tabel 10 Percobaan Robot1 dapat bola Robot2 hilang bola .....	34
Tabel 11 Percobaan Robot1 hilang bola Robot2 hilang bola .....	35
Tabel 12 Hasil Percobaan Strategi.....	37

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi komunikasi berdampak luas dalam kehidupan manusia. Dimulai dengan pengembangan komputer untuk mengolah informasi, kemudian dilanjutkan dengan saling menghubungkan komputer-komputer untuk kepentingan berbagi informasi. Teknologi komunikasi membawa manusia pada era yang baru, era dimana informasi menjadi sangat penting bagi kehidupan manusia dan dapat diakses dimanapun dan kapanpun. Teknologi komunikasi berperan penting sebagai media perantara untuk dapat menyampaikan informasi dan diterima dengan baik oleh penerima informasi. Media transmisi yang tersedia saat ini adalah transmisi dengan menggunakan kabel dan transmisi menggunakan udara. Media transmisi udara adalah salah satu media yang dapat diakses dimanapun dan kapanpun, adapun jenis-jenis yang terkenal yaitu *Infra-red*, *Bluetooth* dan WLAN.

WLAN merupakan teknologi komunikasi nirkabel dimana setiap hubungan antar perangkat elektronik seperti pengiriman dan penerimaan data dilakukan dengan melalui udara dengan teknik gelombang radio. Ini merupakan solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. WLAN merupakan jenis dari transmisi udara yang mempunyai kecepatan hingga 54Mbps mengalahkan *bluetooth* dan *infra-red*, selain itu WLAN mempunyai jarak jangkauan yang lebih jauh dari jenis media transmisi udara yang lainnya, untuk mencari alat WLAN juga cukup banyak dijual dipasaran sehingga untuk melakukan kegiatan-kegiatan pembelajaran maupun riset dalam hal apapun sangat mendukung. Misalkan pembelajaran dalam dunia robotika.

Robot ialah benda bergerak yang dibuat untuk membantu pekerjaan manusia. Robot biasanya digunakan untuk menyelesaikan tugas yang berat, berbahaya, pekerjaan yang berulang dan pekerjaan yang kotor. Tetapi di perguruan tinggi, robot digunakan untuk media pembelajaran dan riset yang berkepanjangan serta untuk mengikuti kompetisi-kompetisi yang berbau tentang robot. Misalkan di Politeknik Negeri Batam, merupakan lembaga pendidikan yang mempunyai UKM khusus tentang robot yaitu *Bareleng Robotic Team*. Jika di lihat dari segi kekurangan di dalam robot khususnya robot bermain sepak bola, banyak sekali kekurangannya seperti tidak adanya komunikasi antara robot, belum

ada penentuan peran masing-masing robot, menentukan strategi, lambatnya dalam menyelesaikan perintah, konfigurasi cahaya yang masih manual, pemetaan posisi robot. Dari berbagai masalah tersebut jika dilihat pada robot bermain sepak bola atau robot berkelompok, masalah yang paling mendasar adalah melakukan komunikasi dengan masing-masing robot dan bagaimana cara mengimplementasikan. Umumnya dalam permainan sepak bola manusia setiap pemain bertujuan memasukkan bola ke gawang dengan sebanyak mungkin menggunakan strategi tertentu sama halnya dalam sepak bola robot, setiap robot akan melakukan perintah yang bertujuan menciptakan goal sebanyak mungkin namun tidak menutup kemungkinan robot tersebut saling menabrak satu dengan yang lainnya

Guna meminimalisir masalah tersebut dibutuhkan penambahan program yang mengimplementasikan komunikasi antar robot. Adapun media transmisinya yaitu media udara dengan jenis WLAN, karena WLAN sendiri mempunyai kecepatan transfer hingga 54 Mbps dengan jarak jangkauan lebih dari 10 meter.

Oleh karena itu, penambahan program komunikasi didalam robot bertujuan untuk meminimalisir kesalahan dalam perebutan bola, meminimalisir tabrakan dengan team sendiri dan mendukung startegi yang dibuat sehingga menghasilkan *output* gerakan yang sesuai dengan startegi yang telah ditentukan.

## **1.2 Rumusan Masalah**

Adapun perumusan masalah yang akan dibahas pada penelitian ini adalah sebagai berikut :

1. Bagaimana robot sepak bola bisa saling berkomunikasi?
2. Bagaimana implementasi komunikasi robot sepak bola?
3. Bagaimana agar robot bergerak sesuai dengan strategi?

## **1.3 Batasan Masalah**

Adapun batasan masalah sebagai berikut :

1. Alat yang digunakan adalah robot sepak bola yang sudah jadi.
2. Komunikasinya hanya pada dua robot penyerang
3. Implementasinya hanya pada lapangan yang telah ditentukan *RoboCup*.
4. Hanya melakukan penambahan program komunikasi ke dalam sistem robot.

## **1.4 Tujuan**

Adapun tujuan yang akan dicapai sebagai berikut :

1. Menghasilkan program komunikasi antar robot.
2. Menghasilkan *output* gerakan dari kedua robot yang saling berkomunikasi.
3. Menghasilkan *output* gerakan dari strategi yang telah disesuaikan.

## **1.5 Sistematika Penulisan**

### **Bab I Pendahuluan**

Pada bab ini menjelaskan tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan dan Sistematika Penulisan.

### **Bab II Landasan Teori**

Pada bab ini menjelaskan tentang rujukan pengerjaan terhadap Studi Pustaka yang menunjang pembuatan studi kasus tentang implementasi komunikasi antar robot pada tim barelang-fc

### **Bab III Analisis dan Perancangan**

Pada bab ini menjelaskan tentang desain sistem, perangkat-perangkat penunjang

### **Bab IV Implementasi dan Pengujian**

Pada bab ini menjelaskan tentang implementasi komunikasi antar robot pada tim barelang-fc

### **Bab V Kesimpulan dan Saran**

Pada bab ini menjelaskan tentang kesimpulan dan saran.

## BAB II

### LANDASAN TEORI

#### 2.1 Linux

Linux adalah nama yang diberikan kepada sistem operasi komputer bertipe Unix. Linux merupakan salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama. Seperti perangkat lunak bebas dan terbuka, Linux dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja. Nama "Linux" berasal dari nama pembuatnya, yang diperkenalkan tahun 1991 oleh *Linus Torvalds*. Beberapa jenis varian Linux, sebagai berikut:

Ubuntu	Knoppix
OpenSUSE	Xandros
Fedora	BackTrack
Mandriva	Slackware
Debian	PCLinuxOS
Sabayon	CentOS
Red Hat	ClearOS
Chrome OS	Gentoo Linux

Adapun Operating System yang digunakan untuk implementasi komunikasi robot adalah Ubuntu Server dengan versi 12.04

#### 2.2 C++

C++ adalah bahasa pemrograman komputer yang dibuat oleh (Bjarne Stroustrup) merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (*American National Standart Institute*) Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem *embedded*, Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an. Program c++ atau c digunakan dalam sistem untuk membuat sekumpulan perintah yang nantinya akan berjalan pada sistem robot.

## 2.3 WLAN

WLAN(*Wireless Local Area Network*) adalah suatu jaringan area lokal tanpa kabel dimana media transmisinya menggunakan frekuensi radio (RF), untuk memberi sebuah koneksi jaringan ke seluruh pengguna dalam area di sekitarnya.

### 2.3.1 Spesifikasi WLAN

Teknologi ini di buat dan di kembangkan sekumpulan insinyur Ameraka Serikat yang bekerja pada *Intitute of Electrical and Electronics Engineers* (IEEE). Berdasarkan spesifikasi IEEE 802.11 sekarang ini ada 4 variasi dari 802.11 dapat dilihat pada tabel 1.

Tabel 1 Spesifikasi WLAN

Jenis	Frekuensi	Kecepatan	Perangkat mendukung
802.11 a	5.0 GHz	54 Mbps	Jenis perangkat 802.11 a
802.11 b	2.4 GHz	5,5 – 11 Mbps	Jenis perangkat 802.11 b
802.11 g	2.4 GHz	54 Mbps	Jenis perangkat 802.11 g
802.11 n	2.4 GHz dan 5.0 GHz	100Mbps	Jenis perangkat 802.11 b,g,n

Dalam sistem robot jenis perangkat *wireless* yang digunakan untuk melakukan komunikasi adalah jenis 802.11 b,g,n dan untuk melakukan pengujian komunikasi yaitu dengan menggunakan perangkat router jenis 802.11 g.

### 2.3.2 Kelebihan dan kekurangan WLAN

Adapun kelebihanannya, sebagai berikut :

- ) Mobilitas Tinggi.
- ) Kemudahan dan kecepatan instalasi
- ) Menurunkan biaya kepemilikan
- ) Fleksibel
- ) Scalable

Selain itu, WLAN juga mempunyai kekurangan, yaitu sebagai berikut :

- ) *Delay* yang besar
- ) Biaya peralatan mahal
- ) Adanya masalah propagasi radio seperti terhalang, terpantul, dan banyak sumber interferensi
- ) Kapasitas jaringan menghadapi keterbatasan spectrum
- ) Keamanan data kurang terjamin

## 2.4 Jenis Komunikasi dan Perbandingannya

Adapun jenis komunikasi yang di bandingkan yaitu *infra-red*, *bluetooth* dan wlan, hasilnya dapat di lihat pada Tabel 2.

Tabel 2 Perbandingan Jenis Komunikasi

Category	<i>Infrared</i>	<i>Bluetooth</i>	WLAN
Jarak	1m	10m	50m
Kecepatan	16Mbps	1 Mbps	1-54 Mbps
Frekuensi	1011-1014 Hz	2.4 GHz	2.4 GHz
Keamanan	-	64bit, 128 bit	SSID

Untuk *infra-red*, jarak paling jauh yaitu 1 meter dengan kecepatan transfer data 16Mbps dengan frekuensi 1011-1014 Hz. Namun *bluetooth*, spesifikasinya lebih tinggi dari perangkat komunikasi *infra-red*. *Bluetooth* mempunyai jarak transfer maksimal 10 meter dengan kecepatan transfer data 1Mbps dengan frekuensi 2.4 GHz. Dan yang terakhir ialah teknologi komunikasi yang jauh diatasnya *infra-red* yaitu wlan. Wlan merupakan perangkat dengan jarak transfer maksimal 50 meter dengan kecepatan hingga 54 Mbps dengan frekuensi sama seperti *bluetooth*.

## 2.5 Komunikasi Antar Robot

Sistem robot majemuk atau berkelompok memiliki beberapa keuntungan seperti lebih efisien waktu, lebih handal, dapat menyelesaikan masalah yang rumit yang tidak dapat dikerjakan robot tunggal. Bentuk umum interaksi antar robot pada robot sistem robot berkelompok (Parker, 2008) :

- Kolektif
- Kerja sama
- Kolaborasi
- Koordinasi

Untuk berinteraksi dengan robot lain dalam robot majemuk dibutuhkan komunikasi. Komunikasi adalah pusat dari sistem komunikasi robot karena menentukan bagaimana robot dapat berinteraksi dengan robot lain (Rui Rochaet.al, 2005). Bentuk interaksi komunikasi dibedakan menjadi tiga yaitu:

1. Melalui lingkungan, penggunaan lingkungan sendiri sebagai media komunikasi
2. Melalui sensor penggunaan sensor untuk observasi dan persepsi aksi dari kelompok
3. Melalui penggunaan sinyal komunikasi untuk pertukaran pesan antara agen. Ada banyak *research* aplikasi teknologi *wireless* untuk komunikasi antar robot. Beberapa diantaranya adalah komunikasi secara *broadcast* dalam team robot Alliance dengan sensor

*feedback* berupa infra red (Parker, 1998), IR untuk komunikasi Swarms Microbotic (Kornienko, 2011), 802.1b *bridge* (Wifi) untuk komunikasi antar robot (Ulam dan Arkin, 2004), komunikasi antar robot yang secara integrasi antara WLAN, Bluetooth dan ZigBee(Witkowski et al., Guardians Project). UWB untuk perangkat komunikasi dan navigasi antar robot (Dougherty, R.,2003).

## 2.6 SBC Fit-PC 2i

Robot memiliki sebuah pengotrol yang akan melakukan semua perintah tertentu sehingga robot tersebut menjadi cerdas. Dalam robot sepak bola ini *main controller*-nya menggunakan SBC-FitPC2i. Adapun spesifikasinya *main controller* ada pada Tabel 3.

**Tabel 3 Spesifikasi SBC-FitPC2i**

SBC fitPC2i	<i>CPU</i>	Intel Atom Z530 CPU @1.6GHz
	<i>Chipset</i>	Intel US15W
	<i>RAM</i>	1GB DDR2
	<i>Storage</i>	4GB Flash Disk
	<i>External Video Port</i>	HDMI
	<i>Audio Port</i>	HD Audio
	<i>Ethernet Port</i>	1000 BaseT
	<i>Wifi</i>	802.11b/g/n
	<i>USB Port</i>	2 PORT USB 2.0
	<i>External Storage</i>	MiniSD
	<i>BIOS</i>	Phoenix
	<i>OS</i>	Linux ( Ubuntu Server 12.04)

SBC-Fit PC 2i, dapat dilihat pada Gambar 1.



**Gambar 1 Tampilan Fit-PC**

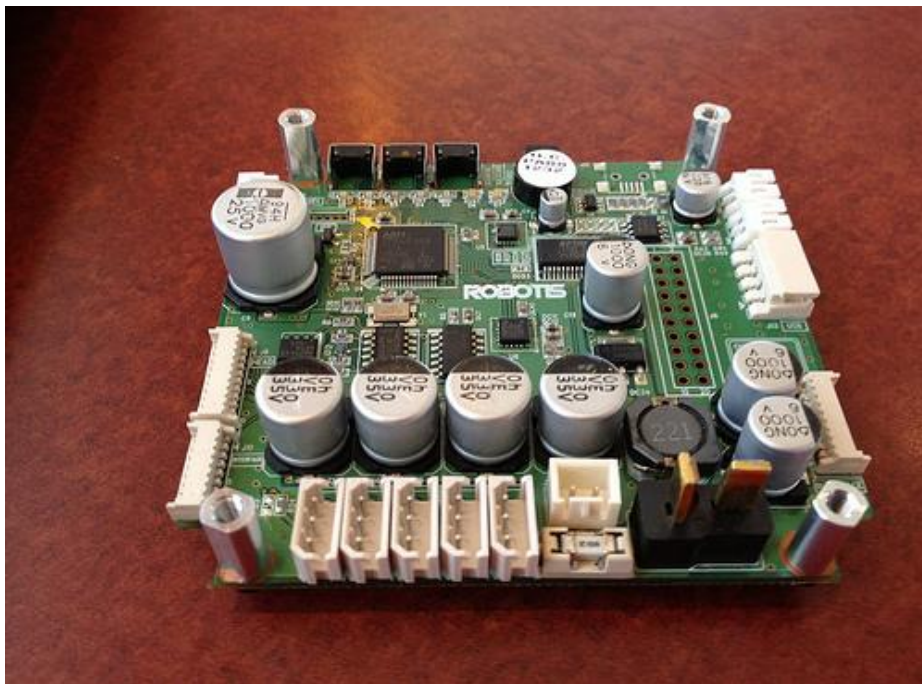
## 2.7 CM730

Hardware ini merupakan *sub controller* yang digunakan dalam robot yang terhubung dengan fit-pc dan dikeluarkan hingga menghasilkan gerakan, alatnya berupa Servo MX-28. Adapun spesifikasinya *sub controller* ada pada Tabel 4.

**Tabel 4 Spesifikasi CM-730**

CM-730	<i>CPU</i>	ARM 32-bit Cortex-M3 STM32F103RE
	<i>Frequency</i>	72 MHz
	<i>Performance</i>	1.25 DMIPS/MHz
	<i>SRAM</i>	64KB
	<i>Flash Memory</i>	512 KB
	<i>Communication</i>	TTL / RS485
	<i>Peripheral</i>	1000 BaseT

CM-730 dapat dilihat pada Gambar 2.



**Gambar 2 Tampilan CM-730**

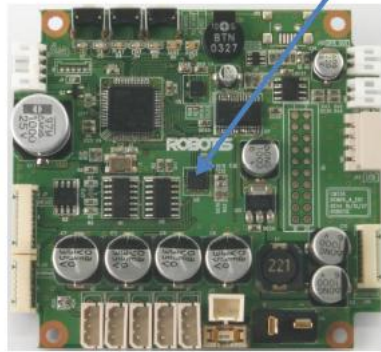
## 2.8 Sensor

Selain *main controller* dan *sub controller*, sensor juga harus ada dalam sebuah robot. Adapun sensor yang digunakan sebagai berikut:

### 2.8.1 Sensor Gyroscope (3-Axis LYPR540AH)

Fungsinya dalam robot ini adalah untuk mengukur orientasi suatu benda berdasarkan ketetapan momentum sudut sehingga mengetahui apakah dalam keadaan jatuh atau tidak. Adapun gambarnya di jelaskan pada Gambar 3

Gyroscope



Gambar 3 Sensor *Gyroscope* pada CM-730

### 2.8.2 Sensor *Accelerometer* (3-Axis ADXL335)

Fungsinya dalam sistem robot ini adalah untuk menjaga keseimbangan robot. Adapun gambarnya dapat dilihat pada Gambar 4.

Accelerometer



Gambar 4 Sensor *Accelerometer* pada CM-730

### 2.8.3 *Camera* (C920)

Fungsinya adalah untuk melihat objek tertentu sehingga dapat mengenali apa yang dilihat sesuai inisialisasi sebelumnya. Adapun gambar kamera dapat dilihat pada Gambar 5.



Gambar 5 Kamera C920

## 2.9 Aktuator (MX-28)

Robot setelah mempunyai *Controller* dasar dan sensor harus mempunyai *output* yang berupa pergerakan. Adapun *output* pergerakan yang digunakan adalah servo MX-28, untuk spesifikasinya dapat dilihat pada tabel 5.

Tabel 5 Spesifikasi MX-28

MX-28	Torsi	24 kgf.cm@12V
	Kecepatan	54 RPM No Load
	Motor	Maxon Coreless DC Motor
	Sensor Posisi	Absolute Encoder
	Ukuran	35.5mm x 50.8mm x 41.8 mm
	Berat	72 gram
	Output	Posisi / kecepatan / temperatur / alarm

Adapun gambar dari aktuatornya dapat dilihat pada Gambar 6.



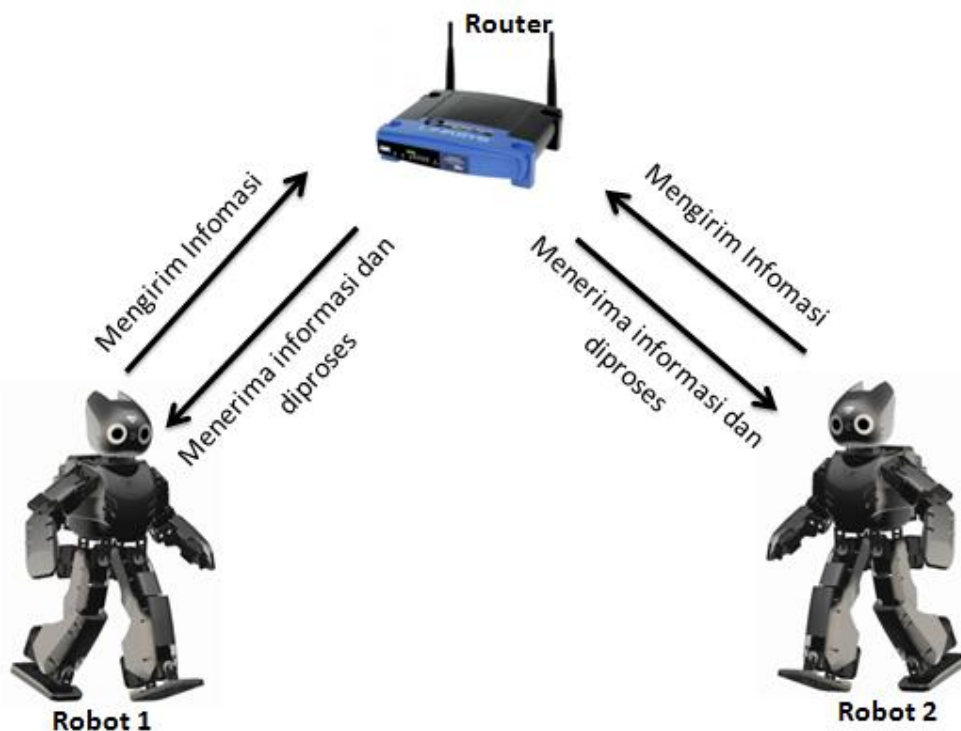
Gambar 6 Servo MX-28

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Deskripsi Umum Sistem

Deskripsi umum sistem komunikasi antar robot pada tim barealng-fc terdiri dari 2 robot dan Robot1 bertindak sebagai *server*, Robot2 bertindak sebagai *client*, Robot1 dan Robot2 awalnya terkoneksi ke router dan mendapatkan IP, sehingga pada saat pertandingan dimulai Robot1 dan Robot2 dapat terhubung dengan router. Cara kerja sistem, Robot1 mengirim data yang berupa *text* ke Robot2 via router dan diterima oleh Robot2 selanjutnya diproses. Jika pada saat tertentu Robot1 ingin meminta informasi berupa keberadaan bola maka Robot1 mengirim permintaan hilang bola ke Robot2 sehingga Robot2 membantu mencarinya, jika Robot1 dan Robot2 menemui bola secara bersamaan dan sehingga mengikuti arah bolanya maka untuk kondisi tertentu Robot1 mengirim *text* ke Robot2 via router untuk tidak mengejar bola. Gambaran deskripsi umum sistem terdapat pada gambar 7.



Gambar 7 Deskripsi Umum Sistem

### 3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah melakukan identifikasi beberapa kebutuhan dan bahan dalam sistem yang akan dipergunakan untuk menambah program pada sistem robot barelang-fc. Analisis kebutuhan sistem pada TA ini dibagi menjadi 2 yaitu analisis kebutuhan perangkat keras robot dan analisis perangkat lunak robot

#### 3.2.1 Analisis kebutuhan perangkat keras robot

Perangkat keras yang digunakan pada robot untuk menunjang pengerjaan Tugas akhir ini ada pada Tabel 6

**Tabel 6 Analisis Kebutuhan Perangkat Keras Robot**

<i>CPU</i>	<i>Main Controller</i>	Fit PC2 Intel Atom Z530 CPU @ 2GHz
	<i>Sub Controller</i>	ARM CortexSTM32F407VG@168MHz
<i>Actuator</i>	<i>Head</i>	2 Dynamixel MX-28
	<i>Arm</i>	2 x 3 Dynamixel MX-28
	<i>Leg</i>	2 x 6 Dynamixel MX-28
<i>Sensor</i>	<i>Gyroscope</i>	ITG-3205
	<i>Accelerometer</i>	ADCL-345
	<i>Camera</i>	Logitech C920
<i>Power</i>	Battery LiPo 3 Cell 4200 mAh	

#### 3.2.2 Analisis kebutuhan perangkat lunak robot

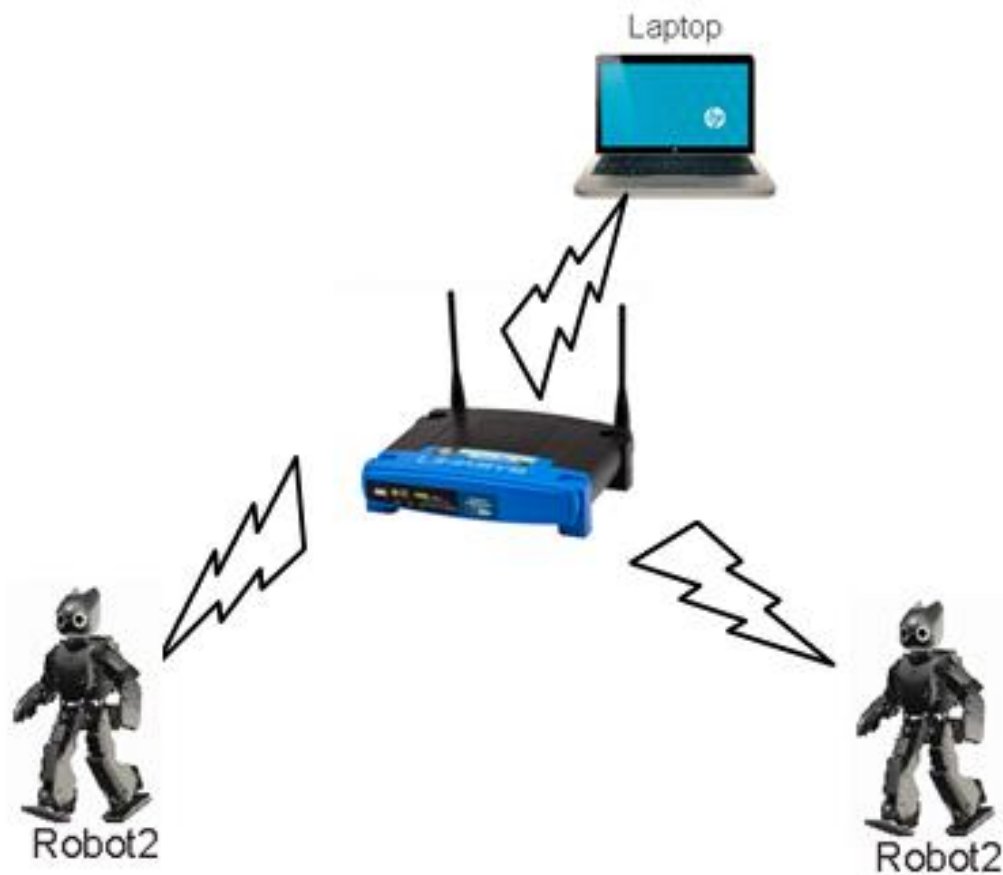
Perangkat lunak yang digunakan pada robot untuk menunjang pengerjaan Tugas akhir ini ada pada Tabel 7.

**Tabel 7 Analisis Kebutuhan Perangkat Lunak Robot**

<i>Software</i>	<i>Operating System</i>	Linux Ubuntu / Windows XP / Windows 7
	<i>Framework</i>	Open-DARwIn SDK
	<i>Programming Language</i>	C / C++ / Java / Python / Perl
	<i>Compiler</i>	Gcc / G++
	<i>Motion Editor</i>	Motion-DARwIn TCP/IP
	<i>Development Environment</i>	Eclipse / Qeditor
	<i>Code Management</i>	Subversion
	<i>Remote Management</i>	SSH
	<i>File Sharing</i>	FTP

### 3.2.3 Analisis topologi yang digunakan saat komunikasi

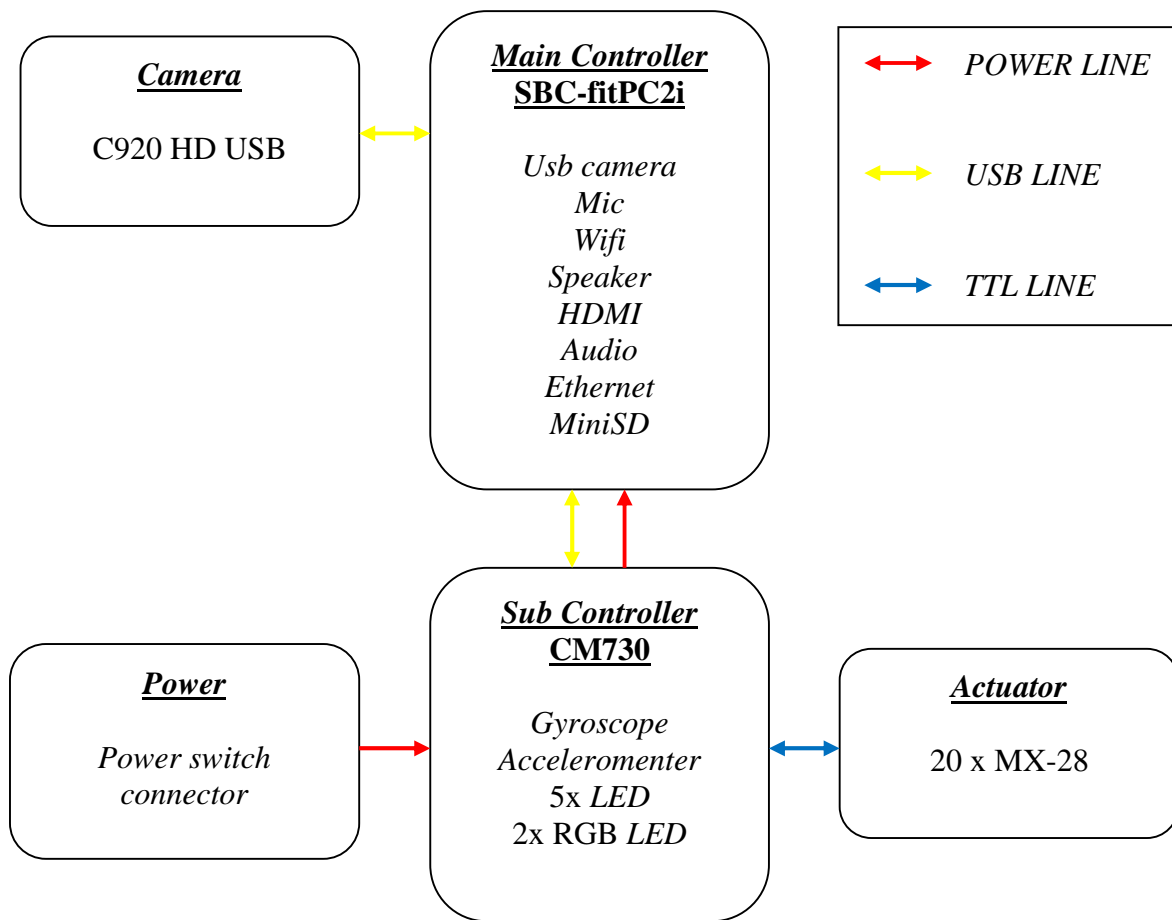
Pada Tugas Akhir ini, sifat dari komunikasi antar robot ini adalah WLAN. Untuk WLAN sendiri biasanya digunakan untuk berbagi sumber daya seperti printer dan sharing data. Jumlah user yang berada pada jaringan kurang lebih 5 user. Berdasarkan kondisi tersebut maka disarankan untuk memilih topologi jenis star, karena yang menjadi titik akses dari robot atau komputer adalah router untuk memungkinkan robot atau komputer menerima dan mengirim data ke robot atau komputer yang terhubung ke jaringan wireless. Adapun penjelasan topologi komunikasi antar robot dapat dilihat pada gambar 8.



Gambar 8 Topologi Star

### 3.3 Perancangan program

Dalam perancangan program komunikasi yang harus diperhatikan adalah peraturan pertandingan standar internasional, strategi yang digunakan dan algoritma yang akan diterapkan sehingga dapat menghasilkan program yang sesuai dengan system robot. Gambaran secara garis besar tentang diagram blok sistem robot dapat dilihat pada Gambar 9.



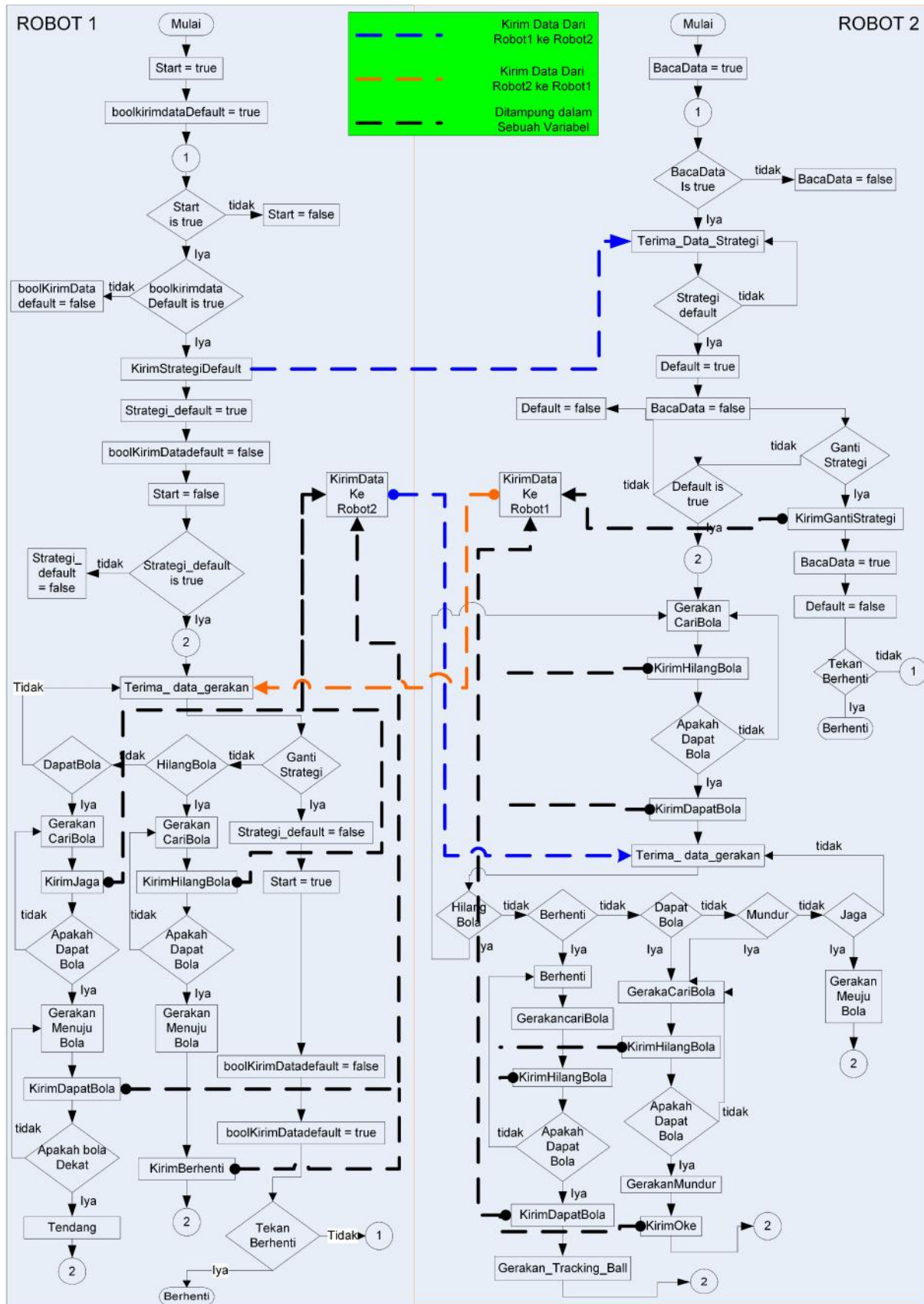
Gambar 9 Blok Diagram Sistem Robot

Pada gambar diatas menjelaskan tentang diagram blok dari alur *system* robot . Awalnya *power switch* di tekan sehingga *sub controller* CM-730 dan *Main Controller* dalam keadaan hidup selanjutnya *main controller* dan *sub controller* dapat saling berkomunikasi. Didalam *main controller* terdapat program yang akan dieksekusi sehingga dapat mengakses *usb camera* sebagai sensor penglihatan robot dan *Sub Controller* sebagai media *output* yang menghasilkan gerakan. Di dalam *Sub Controller* terdapat sensor *gyroscope* dan *accelerometer* serta tempat mengaksesnya 20 penggerak yang berupa servo MX-28.

### 3.4 Flow chart Pergerakan Robot

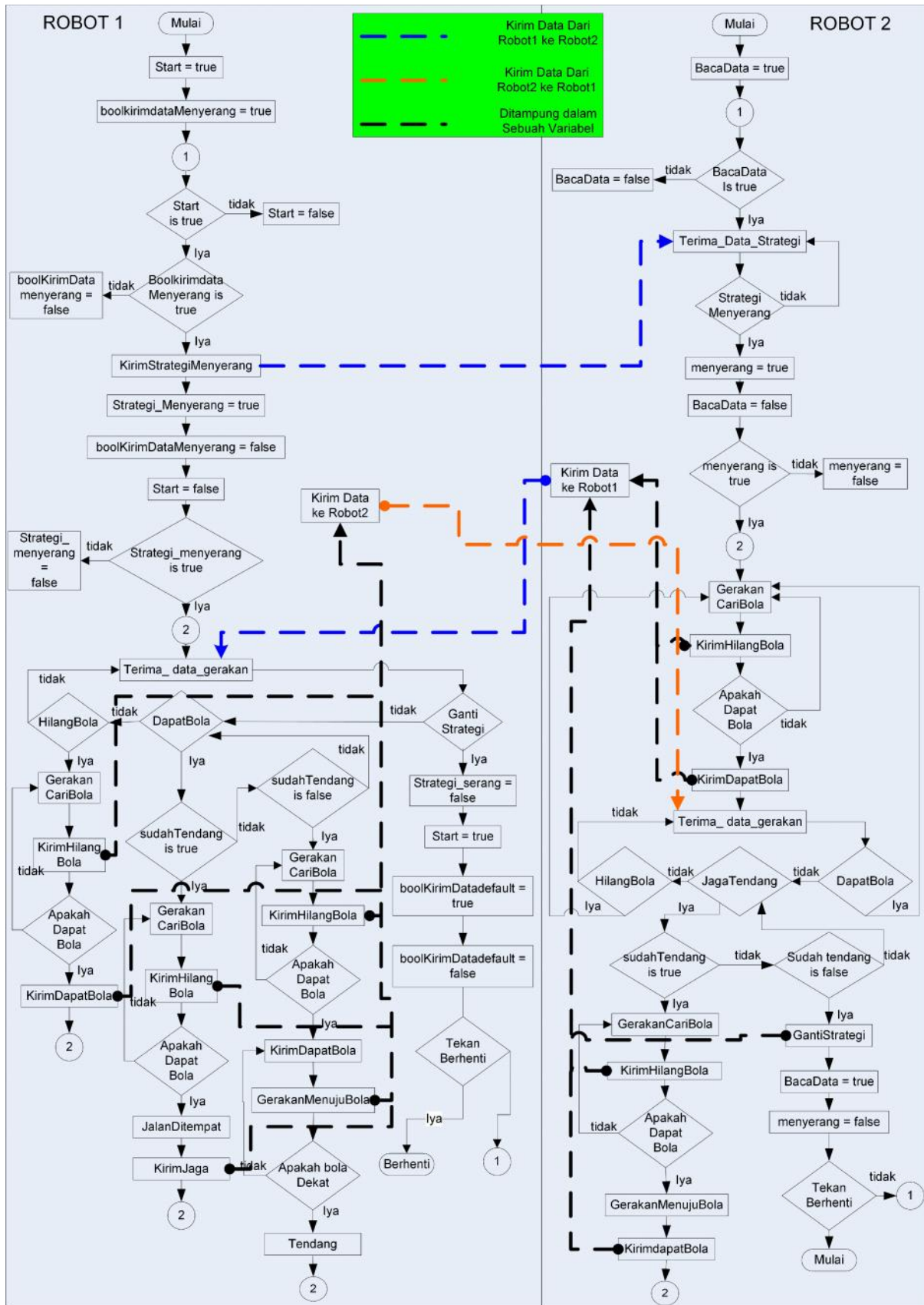
Dalam perancangan program diperlukan *flowchart* dari pergerakan robot yang saling berkomunikasi hingga membentuk formasi tertentu, flow chart terbagi menjadi 2, *flowchart* pada Robot1 dan Robot2 *mode* strategi *default*, *flowchart* pada Robot1 dan Robot2 *mode* strategi menyerang.

### 3.4.1 FlowChart Pada Robot1 dan Robot2 Mode Strategi Default



Gambar 10 Flowchart Robot1 dan Robot2 Mode Strategi Default

### 3.4.2 FlowChart Pada Robot1 dan Robot2 Mode Strategi Menyerang

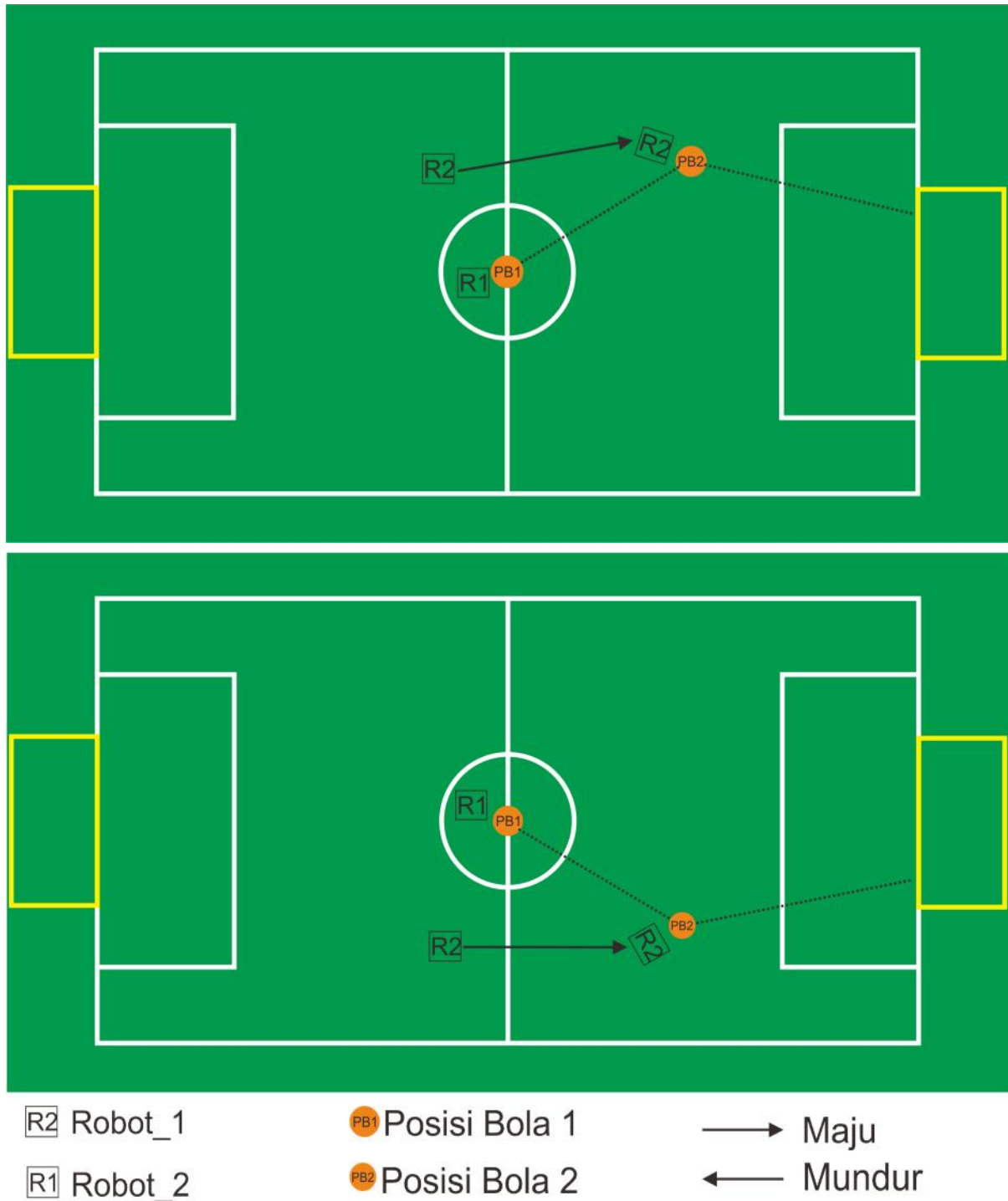


Gambar 11 Flowchart Robot1 dan Robot2 Mode Strategi Menyerang

### 3.5 Strategi

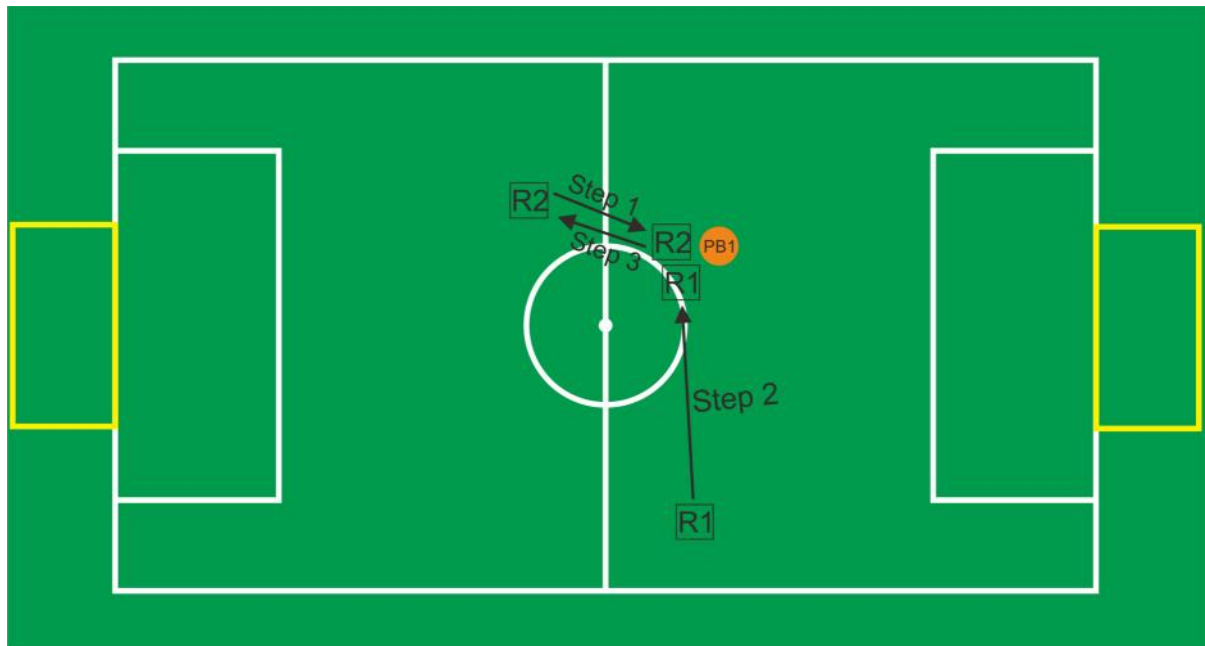
Setelah Robot siap untuk digunakan, selanjutnya adalah menentukan algoritma apa yang digunakan hingga perancangan strategi yang untuk diterapkan dalam pertandingan menentukan. Adapun strateginya sebagai berikut :

#### 3.5.1 Strategi Menyerang



Gambar 12 Strategi Penyerangan

### 3.5.2 Strategi Default



R2 Robot\_1

R1 Robot\_2

PB1 Posisi Bola 1

PB2 Posisi Bola 2

→ Maju

← Mundur

Gambar 13 Strategi Default

## BAB IV

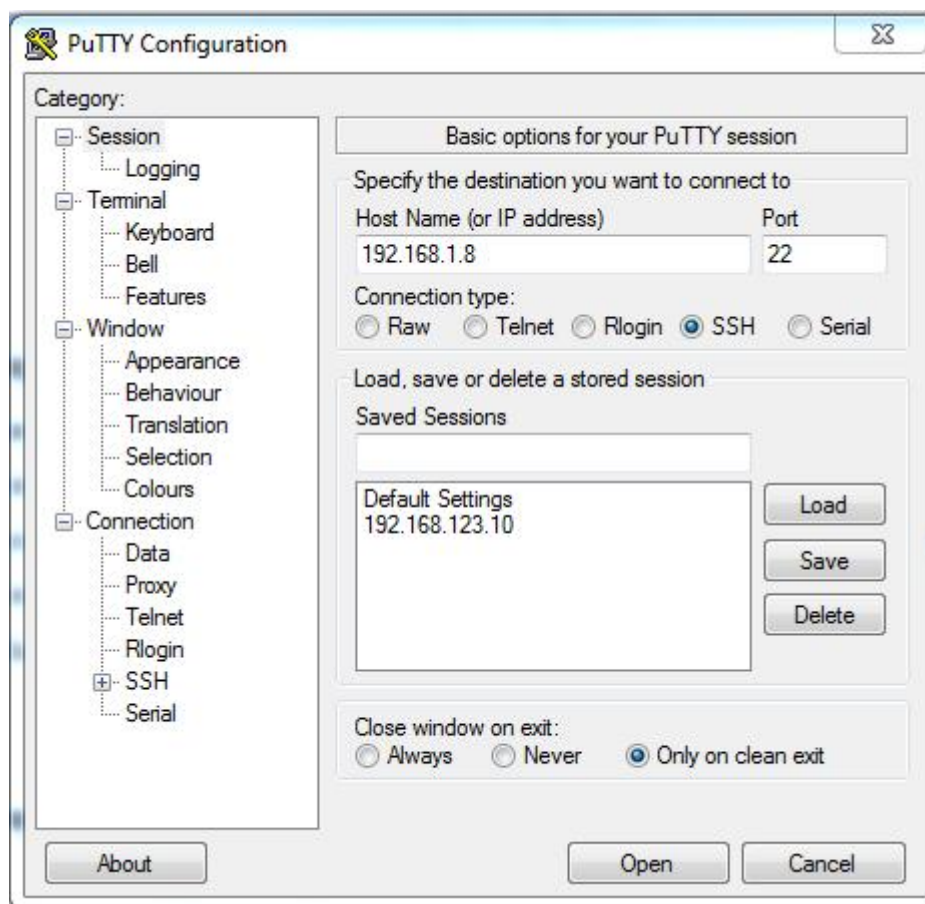
### IMPLEMENTASI DAN HASIL

#### 4.1 Implementasi Program

##### 4.1.1 Cara Implementasi Program Robot

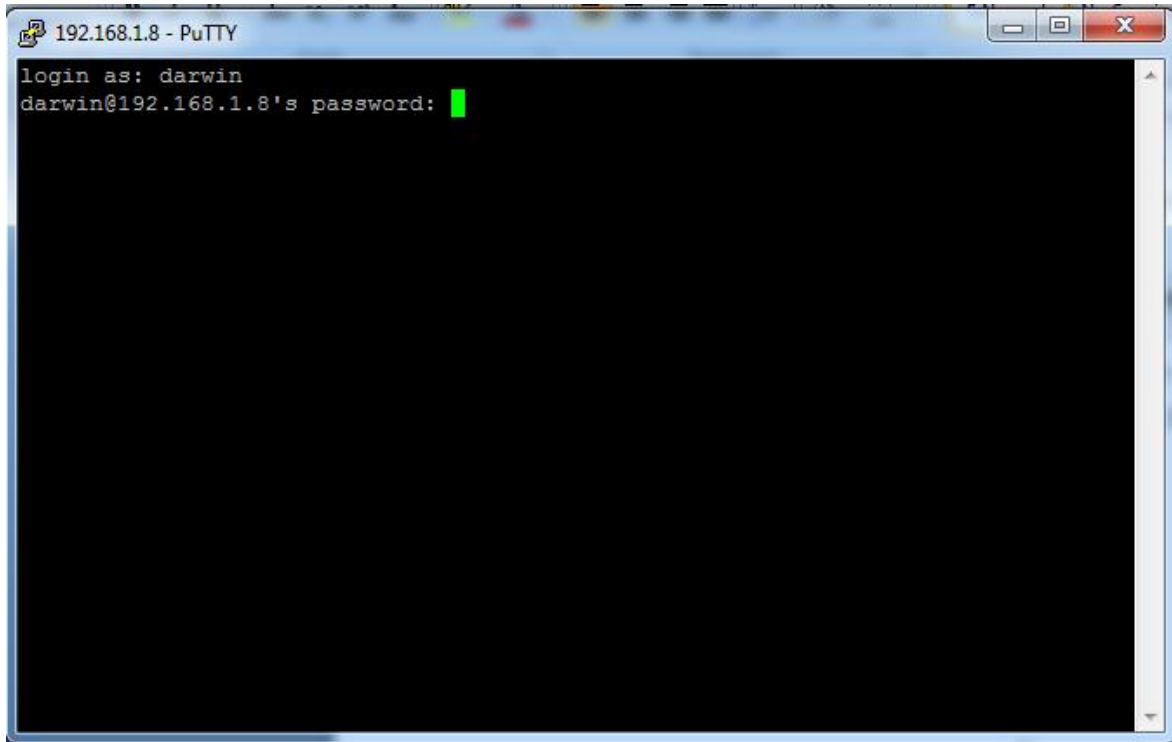
Adapun cara implementasi program robot, sebagai berikut:

1. Persiapan aplikasi putty untuk koneksi ke robot dengan jaringan kabel atau wireless, sebagai contoh IP address robot2 192.168.1.8 dengan menggunakan Wireless.
2. Selanjutnya melakukan koneksi ke robot dengan IP address 192.168.1.8



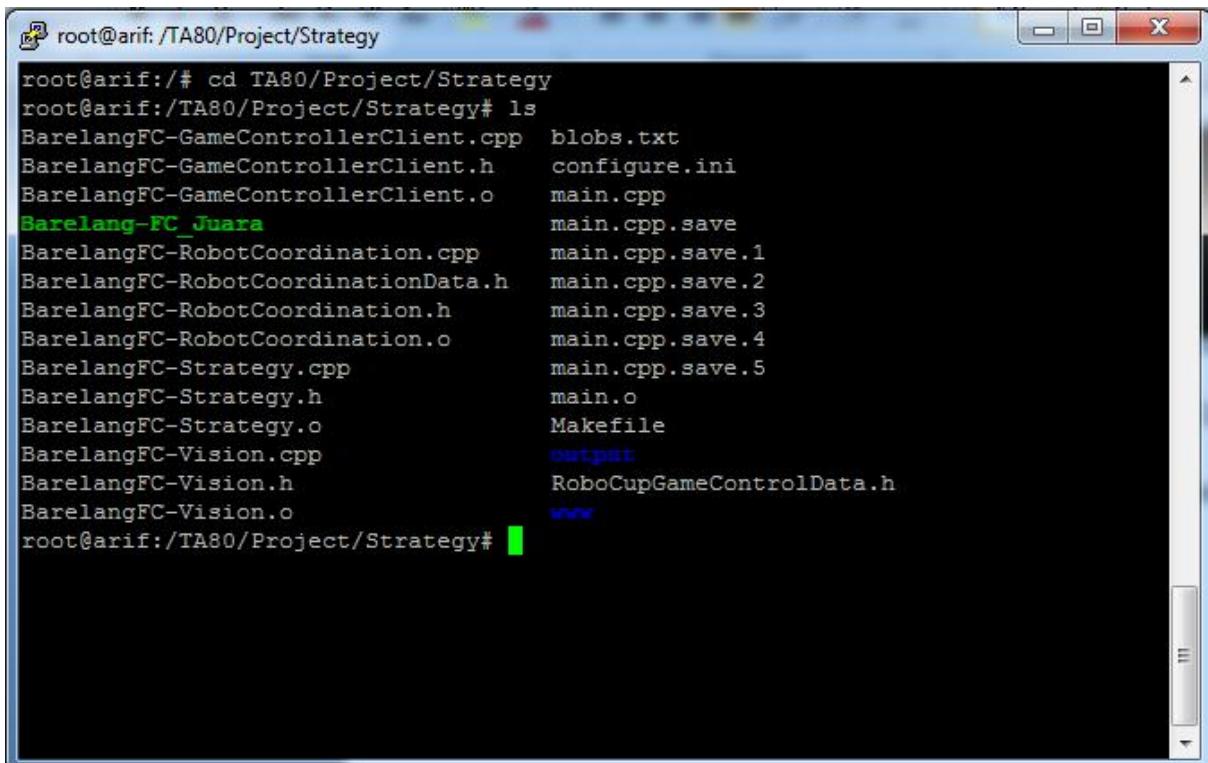
Gambar 14 Putty

3. Setelah melakukan koneksi ke IP *address* 192.168.1.8, maka akan tampil halaman terminal yang meminta memasukkan *password* robotnya.



Gambar 15 Proses Login

- Langkah selanjutnya akan langsung menuju ke direktori program *main* robot.



Gambar 16 Direktori Program Robot

- Setelah itu melakukan pembukaan *file* *main.cpp* yang isinya void gerakan robot dengan pusatnya pada `int main()`

```
root@arif: /TA80/Project/Strategy
GNU nano 2.2.6 File: main.cpp

void KirimBerhenti() {
    sprintf(motionData,"Berhenti");
    if (sendto(komunikasiB,motionData , BUFSIZE , 0 , (struct sockaddr *) &Re$
    for(int c=0; c<BUFSIZE; c++){
        motionData[c]=0;
    }
}

void KirimJaga() {
    sprintf(motionData,"Jaga");
    if (sendto(komunikasiB,motionData , BUFSIZE , 0 , (struct sockaddr *) &Re$
    for(int c=0; c<BUFSIZE; c++){
        motionData[c]=0;
    }
}

void KirimHilangBola() {
    sprintf(motionData,"Hilang Bola");
    if (sendto(komunikasiB,motionData , BUFSIZE , 0 , (struct sockaddr *) &Re$
    for(int c=0; c<BUFSIZE; c++){
        motionData[c]=0;
    }
}

void KirimDapatBola() {
    sprintf(motionData,"Dapat Bola");
    if (sendto(komunikasiB,motionData , BUFSIZE , 0 , (struct sockaddr *) &Re$
    for(int c=0; c<BUFSIZE; c++){
        motionData[c]=0;
    }
}

void KirimMundur() {
    sprintf(motionData,"Mundur");
    if (sendto(komunikasiB,motionData , BUFSIZE , 0 , (struct sockaddr *) &Re$
    for(int c=0; c<BUFSIZE; c++){
        motionData[c]=0;
    }
}

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Gambar 17 Void Pengiriman Komunikasi

Pada Gambar 17, merupakan tempat dimana void semua proses pergerakan, komunikasi, kamera hingga strategi yang nantinya akan dipanggil di dalam main program.

```

root@arif: /TA80/Project/Strategy
GNU nano 2.2.6      File: main.cpp      Modified
    }
    return 1;
}
else {
    return 0;
}
}

int main(void){
    int strategi;
    bool menyerang = false, bertahan = false, biasa = false;
    runLuaProgram();
    connect();
    motion();
    komunikasi();
    kamera.initialize(0, camWidth, camLength);
    robotReady();
    pthread_create(&threadStreaming, NULL, streamCamera, NULL);
    pthread_create(&threadCamSetting, NULL, settingCamera, NULL);
    kamera.loadconfig();
    struct timeval timerRobotIsReady, timerState1, timerState2, timermulai;
    bool robotIsReady = false;
    bool kirimdata = false;
    bool state1 = false, state2 = false, mulai=false;
    setTimer(timerRobotIsReady, 7);

// KirimModeBertahan();
// KirimModeMenyerang();
    KirimModeBiasa();

// strategi = KirimModeBertahan();
// strategi = KirimModeMenyerang();
    strategi = KirimModeBiasa();

    switch(strategi) {
        case 0 :
            bertahan = true;
            break;
        case 1 :
    }
}

```

Gambar 18 Void Main

Gambar 18 merupakan otak dari program robot ini yang akan selalu melakukan perulangan, void yang telah di buat akan di panggil dan dieksekusi pada int main().

## 4.2 Implementasi Program Komunikasi dan Strategi

### 4.2.1 Implementasi Program Komunikasi pada Robot1

Function komunikasi

- If komunikasiA = socket (.....) then die(socket) endif
- If komunikasiB = socket (.....) then die(socket) endif
- Memset ((char\*) &kom,0,sizeof(kom))
- Memset ((char\*) &RemoteKom,0,sizeof(RemoteKom))
- RemoteKom.sin\_family ← AF\_INET
- RemoteKom.sin\_port ← htons(PORT\_OUT)
- If inet\_aton("192.168.1.4",.....)=0 then write "inet\_aton() failed" endif
- Kom.sin\_family ← AF\_INET

```

Kom.sin_port ←htons(PORT_OUT)
Kom.sin_port ←htons(PORT_IN)
Kom.sin_addr.s_addr ←htonl(inaddr_and)
If bind (komunikasiA, .....)= -1 then die(bind) endif
end

```

Program pada robot1 tentang main komunikasi yang mampu mengirim dan menerima dari IP yang telah di tentukan

```

Function kirimBerhenti
  Set(motionData,"berhenti")
  If sendto(komunikasiB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "Berhenti" ke Robot2

```

Function kirimJaga
  Set(motionData,"jaga")
  If sendto(komunikasiB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "Jaga" ke Robot2

```

Function kirimHilangBola
  Set(motionData,"Hilang_Bola")
  If sendto(komunikasiB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "HilangBola" ke Robot2

```

Function kirimDapatBola
  Set(motionData,"Dapat_Bola")
  If sendto(komunikasiB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "DapatBola" ke Robot2

```
Function kirimMundur
```

```
  Set(motionData,"Mundur")
```

```
  If sendto(komunikasiB,.....) =-1 then die(sendto)
```

```
  For c←0 to buflen do motiondata[c]←0
```

```
end
```

Program komunikasi untuk mengirim *String* "Mundur" ke Robot2

```
If terima ←recvfrom(komunikasiA,.....) = -1 then die(recvfrom)
```

Program komunikasi untuk Menerima *String* dari Robot2

#### 4.2.2 Implementasi Program Strategi pada Robot1

```
If strategi_default ←true then
```

```
  If terima ←recvfrom(komunikasiA,.....) die(recvfrom) endif
```

```
  If strcmp (buf, "hilang_bola") = 0 then
```

```
    If kamera.ball_X = -1 and kamera.ball_Y ==-1) then
```

```
      KirimHilangBola()
```

```
      searchBall(0)
```

```
    else
```

```
      kirimBerhenti()
```

```
      followBall(0)
```

```
    endif
```

```
  else if strcmp (buf,"dapat_bola") = 0 then
```

```
    If kamera.ball_X = -1 and kamera.ball_Y ==-1) then
```

```
      KirimJaga()
```

```
      searchBall(0)
```

```
    else
```

```
      followBall()
```

```
      if headtilt > -1.0 then
```

```
        kirimDapatBola()
```

```
      else
```

```
        kirimJaga()
```

```
      endif
```

```
    endif
```

```
  endif
```

```
endif
```

Program strategi "*Default*" pada Robot1

```

If strategi_serang ← true then
  If terima ← recvfrom(komunikasiA,.....) die(recvfrom) endif
  If strcmp (buf, "hilang_bola") = 0 then
    If sudah_tendang ← false
      If kamera.ball_X = -1 and kamera.ball_Y = -1) then
        KirimHilangBola()
        searchBall(0)
      else
        walk(0.00,0.00,0.00)
        kirimDapatBola()
      endif
    else
      If kamera.ball_X = -1 and kamera.ball_Y = -1) then
        walk(0.00,0.00,0.00)
        KirimHilangBola()
      else
        walk(0.00,0.00,0.00)
        kirimDapatBola()
      endif
    endif
  else if strcmp (buf,"dapat_bola") = 0 then
    If sudah_tendang ← false
      If kamera.ball_X = -1 and kamera.ball_Y = -1) then
        walk(0.00,0.00,0.00)
        KirimJagaTendang()
      else
        walk(0.00,0.00,0.00)
        KirimJagaTendang()
      endif
    else
      If kamera.ball_X = -1 and kamera.ball_Y = -1) then
        searchBall(0)
        KirimHilangBola()
      else
        followBall(0)
        if headTilt > -0.4 then
          walkLeftKick()
          sudahTendang ← true
        else

```

```

        KirimDapatBola()
    endif
endif
endif
else if strcmp (buf,"GantiStrategi") = 0 then
    strategi_serang ← false
    start ← true
    boolkirimdatadefault ← true
    boolkirimdatamenyerang ← false
endif
endif

```

Program strategi “Menyerang” pada Robot1

### 4.2.3 Implementasi Program Komunikasi pada Robot2

```

Function komunikasiSend
    If comA = socket (.....) then die(socket) endif
    If comB = socket (.....) then die(socket) endif
    Memset ((char*) &com,0,sizeof(com))
    Memset ((char*) &RemoteCom,0,sizeof(Remotecom))
    RemoteCom.sin_family ← AF_INET
    RemoteCom.sin_port ← htons(PORT_OUT)
    If inet_aton("192.168.1.8",.....) = 0 then write "inet_aton() failed" endif
    Kom.sin_family ← AF_INET
    Kom.sin_port ← htons(PORT_IN)
    Kom.sin_addr.s_addr ← htonl(inaddr_any)
    If bind (komunikasiA, ..... ) = -1 then die(bind) endif
end

```

Program pada robot2 tentang main komunikasi yang mampu mengirim dan menerima dari IP yang telah di tentukan.

```

Function HilangBola
    Set(motionData,"Hilang_Bola")
    If sendto(comB,.....) =-1 then die(sendto)
    For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* “HilangBola” ke Robot1

```

Function DapatBola
  Set(motionData,"Dapat_Bola")
  If sendto(comB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "DapatBola" ke Robot1

```

Function oke
  Set(motionData,"Oke")
  If sendto(comB,.....) =-1 then die(sendto)
  For c←0 to buflen do motiondata[c]←0
end

```

Program komunikasi untuk mengirim *String* "oke" ke Robot1

```

If terima ←recvfrom(comA,.....) = -1 then die(recvfrom)

```

Program komunikasi untuk Menerima *String* dari Robot1

#### 4.2.4 Implementasi Program Strategi pada Robot2

```

If default ←true
  Write "strategi default"
  If ball_pos.X = -1 and ball_pos.Y = -1 then
    walking.stop()
    searchBallFar()
    Hilangbola
  Else
    Walking.stop()
    dapatBola
  endif
If terima ←recvfrom(comA,.....) = -1 then die(recvfrom)
If strcmp (buf, "hilang_bola") = 0 then
  If ball_pos.X = -1 and ball_pos.Y =-1) then
    HilangBola()
    searchBall(0)
  else
    DapatBola()
  endif
endif

```

```

If strcmp (buf, "dapat_bola") = 0 then
  If ball_pos.X != -1 and ball_pos.Y !=-1) then
    If headTilt>60 then
      Walking.stop()
      Oke()
    else
      jalan(-10,0,0)
      Oke()
    else
      searchBall(0)
      HilangBola()
    endif
  endif
endif
If strcmp (buf, "jaga") = 0 then
  Follow.proses(tracker.ball_position)
Endif
If strcmp (buf, "mundur) = 0 then
  If ball_pos.X != -1 and ball_pos.Y !=-1) then
    If headTilt>50 then
      Walking.stop()
    else
      jalan(-10,0,0)
      Oke()
    else
      searchBall(0)
      HilangBola()
    endif
  endif
Endif
If strcmp (buf, "berhenti") = 0 then
  If ball_pos.X != -1 and ball_pos.Y !=-1) then
    searchBall(0)
    HilangBola()
  else
    tracker.ball_position;
    DapatBola()
  endif
endif
endif

```

Program strategi "*Default*" pada Robot2

```

If default ←true
  Write “strategi default”
  If ball_pos.X = -1 and ball_posY = -1 then
    walking.stop()
    searchBallFar()
    Hilangbola
  Else
    Walking.stop()
    dapatBola
  endif
If terima ←recvfrom(comA,.....) = -1 then die(recvfrom)
If strcmp (buf, “hilang_bola”) = 0 then
  If ball_pos.X = -1 and ball_pos.Y =-1) then
    Walking.stop()
    searchBall(0)
    HilangBola()
  else
    Walking.stop()
    DapatBola()
  endif
endif

If strcmp (buf, “jagaTendang”) = 0 then
  If sudah_tendang ←false
    If kamera.ball_X = -1 and kamera.ball_Y =-1) then
      KirimHilangBola()
      searchBall(0)
    else
      follower.process(tracker.ball_position)
      if headTilt < 30 and headPan <0 then
        TendangKiri()
        If countTendang >15 then
          GantiStrategiDefault()
          SudahTendang ←true
        Else
          countTendang++
        endif
      else if headTilt < 30 and headPan>0 then
        Tendangkanan()

```

```

        If countTendang >15 then
            GantiStrategiDefault()
            SudahTendang ←true
        Else
            countTendang++
        endif
    else
        kirimDapatBola()
    endif
endif
else
    GantiStrategiDefault()
    bacaData ← true
    serang ←false
endif
endif
else if strcmp (buf,"dapat_bola") = 0 then
    If ball_pos.X != -1 and ball_pos.Y !=-1) then
        searchBall(0)
        HilangBola()
    else
        tracker.ball_position;
        DapatBola()
    Endif
endif
endif

```


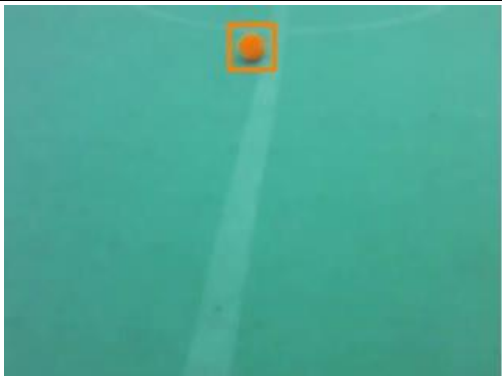


Program strategi “Menyerang” pada Robot2

### 4.3 Hasil Percobaan

#### 4.3.1 Hasil Percobaan Komunikasi

##### 4.3.1.1 Robot1 dapat bola dan Robot2 dapat bola

**Tabel 8 Percobaan Robot1 Dapat bola Robot2 Dapat Bola**





Pandangan Robot1	Pandangan Robot2	Gerakan Robot1	Gerakan Robot2	Keterangan
		Mengikuti Bola	Berhenti	Sesuai
		Mengikuti Bola	Mundur	Sesuai

		Mengikuti Bola	Mundur	Sesuai
---	--	----------------	--------	--------

#### 4.3.1.2 Robot1 hilang bola dan Robot2 hilang bola

**Tabel 9 Percobaan Robot1 hilang bola Robot2 dapat bola**

Pandangan Robot1	Pandangan Robot2	Gerakan Robot1	Gerakan Robot2	Keterangan
		Mencari Bola	Mencari Bola	Sesuai

		<p>Mencari Bola</p>	<p>Mencari Bola</p>	<p>Sesuai</p>
		<p>Mencari Bola</p>	<p>Mencari Bola</p>	<p>Sesuai</p>

### 4.3.1.3 Robot1 dapat bola dan Robot2 hilang bola

Tabel 10 Percobaan Robot1 dapat bola Robot2 hilang bola


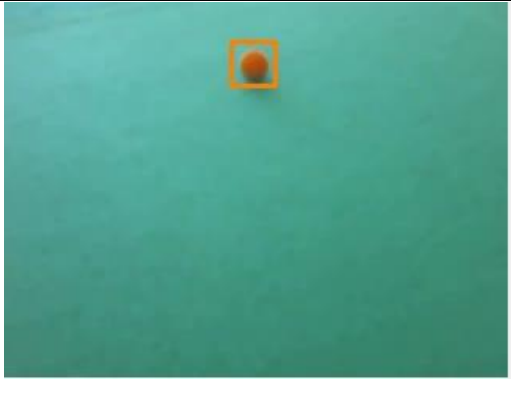

Pandangan Robot1	Pandangan Robot2	Gerakan Robot1	Gerakan Robot2	Keterangan
		Mengikuti Bola	Cari Bola	Sesuai
		Mengikuti Bola	Berhenti	Sesuai

		Mengikuti Bola	Cari Bola	Sesuai
---	--	----------------	-----------	--------

#### 4.3.1.4 Robot1 hilang bola dan Robot2 dapat bola

Tabel 11 Percobaan Robot1 hilang bola Robot2 hilang bola

Pandangan Robot1	Pandangan Robot2	Gerakan Robot1	Gerakan Robot2	Keterangan
		Cari Bola	Jaga Bola	Sesuai

		<p>Cari Bola</p>	<p>Jaga Bola</p>	<p>Sesuai</p>
		<p>Cari Bola</p>	<p>Jaga Bola</p>	<p>Sesuai</p>

### 4.3.2 Hasil Percobaan Strategi

Tabel 12 Hasil Percobaan Strategi

Strategi	Percobaan Komunikasi	Hasil	Keterangan
<i>Default</i>	Kirim komunikasi Hilang Bola dari Robot1 ke Robot2	Robot2 mampu melakukan penerimaan komunikasi Hilang Bola dan masuk ke program strategi <i>Default</i> .	Berhasil
	Kirim komunikasi Hilang Bola dari Robot2 ke Robot1	Robot1 mampu melakukan penerimaan komunikasi Hilang Bola dan masuk ke program strategi <i>Default</i>	Berhasil
Menyerang	Kirim komunikasi Dapat Bola dari Robot1 ke Robot2	Robot2 mampu melakukan penerimaan komunikasi Dapat Bola dan masuk ke program strategi Menyerang.	Berhasil
	Kirim Komunikasi Dapat Bola dari Robot2 ke Robot1	Robot2 mampu melakukan penerimaan komunikasi Dapat Bola dan masuk ke program strategi Menyerang.	Berhasil

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Implementasi komunikasi antar robot pada tim Barelang-FC dapat diselesaikan dengan output gerakan yang cukup baik. Implementasi komunikasi antar robot telah sesuai dengan tujuan awal pembuatan, melakukan perancangan program komunikasi antar robot dan menghasilkan output gerakan dari robot yang saling berkomunikasi serta gerakan robot sesuai strategi yang telah di tentukan sebelumnya.

#### **5.2 Saran**

Saran-saran yang dianggap dapat mengembangkan Komunikasi antar robot pada tim Barelang-FC antara lain:

1. Penambahan mode strategi
2. Pengembangan komunikasi antar robot menjadi 3 atau 4 Robot.
3. Menambahkan program *interrupt* pada robot.

## DAFTAR PUSTAKA

- [1] Dougherty, Robert et al. (2003). A Behavior Control Approach to Formation-Keeping Through an Obstacle Field,0-7803-8155-6, 2004 IEEE.
- [2] Hanif al fatta (2006). Dasar Pemrograman C++ disertai dengan Pengenalan Pemrograman Berorientasi Objek. ISBN 979-763-582-1.
- [3] Hendri(2009), 802.11a/b/g/n. <http://hendri.staff.uns.ac.id/2009/12/wifi-802-11-a-b-g-n/>. Diakses 26 Desember 2015.
- [4] Kornienko S.(2011), IR-based Communication and Perception in Microrobotic Swarms, arXiv:1109.3617v1 [cs.RO] 16 Sep 2011, Institute of Parallel and Distributed Systems, University of Stuttgart, Germani.
- [5] Labiod, H, Afifi, H. And De Santis, C. (2007). WiFi™, Bluetooth™, ZigBee™, And Wimax™, Netherlands, Springer.
- [6] "Linux Online Tentang Sistem Operasi Linux". Linux.org. Diakses 17 September 2014.
- [7] Parker, Lynne E. (1998), Adaptive Heterogeneous Multi-Robot Teams, Elsevier Preprint, 17 November 1998.
- [8] Parker, Lynne E.(2008), Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems, Journal Of Physical Agents, Vol. 2, No. 1, March 2008.
- [9] "Pengertian Wireless Perangkat Sistem Keamanan Jaringan Wireless". Jaringan komputer.org. Diakses 2 January 2015.
- [10] Robotis, 2007, Hardware Specification & Diagram Block, Schematic Hardware
- [11] RoMeLa, 2011, Assambly Manual Darwin-OP, Viginia Tech.
- [12] RoMeLa, 2011, Fabrication Manual Darwin-OP, Viginia Tech.

- [13] Rui Rocha, Jorge Dias, Adriano Carvalho (2005). CMRS: A Study of vision Based 3-D mapping using information theory. *Robotics and Autonomous System* 53, 282-311.
- [14] Ulam, Patrick & Arkin Ronald C, *Communications Recovery For Multi-Robot Teams*, Defense Advanced Research Projects Agency, 3701 North Fairfax Drive, Arlington, VA, 22203-1714.
- [15] Witkowski, U. et al., Ad-hoc network communication infrastructure for multi robot systems in disaster scenarios. GUARDIANS project no. 045269.
- [16] "Zigbee, Wireless, Bluetooth, infrared". [bbs.cechina.cn](http://bbs.cechina.cn). Diakses 30 Desember 2014.

# LAMPIRAN

## 1.1 ROBOT 1

```
////////////////////////////////////
////////////////////////////////////
//////////////////////////////////// ROBOT 1 //////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

#include <stdio.h>
#include "opencv2/opencv.hpp"
#include <time.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <libgen.h>
#include "BarelangFC-Vision.h"
#include "BlobResult.h"
#include <sys/time.h>
#include "minIni.h"
#include <unistd.h>
#include <pthread.h>

#define ROBOT_4

#define OUT_PORT          2001
#define OUT_PORT_MOTION  2000

#define camWidth    320
#define camLength   240
#define PI          3.1415926

////////////////////////////////////Port yang digunakan untuk melakukan komunikasi////////////////////////////////////
#define PORT_OUT    30
#define PORT_IN     31
#define BUFLLEN     100
////////////////////////////////////

using namespace BarelangFC;
```

BarelangVision kamera;

//////////Inisialisasi Variabel Komunikasi 2 Robot//////////

```
struct sockaddr_in si_me, si_other, Remote;
struct sockaddr_in Motion , RemoteMotion;
struct sockaddr_in kom, RemoteKom, komunikasi_luar;
int s, l, i, slen = sizeof(si_other), recv_len;
int sm, lm , im ;
int sme, lme, ime;
int komunikasiA, komunikasiB, com_len = sizeof(komunikasi_luar), terima;
char buf[BUFLLEN];
```

////////////////////////////////////

```
pthread_t threadStreaming, threadCamSetting;
bool sudahTendang = false;
```

```
void die(char* s){
    perror(s);
    exit(1);
}
```

//////////Pengiriman Komunikasi melalui IP address//////////

```
void komunikasi() {
    if((komunikasiA=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    if((komunikasiB=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    memset((char *) &kom, 0, sizeof(kom));
    memset((char *) &RemoteKom, 0, sizeof(RemoteKom));
    RemoteKom.sin_family = AF_INET;
    RemoteKom.sin_port = htons(PORT_OUT);
    if (inet_aton("192.168.1.3" , &RemoteKom.sin_addr) == 0){
        fprintf(stderr, "inet_aton() failed\n");
        exit(1);
    }
    kom.sin_family = AF_INET;
    kom.sin_port = htons(PORT_IN);
    kom.sin_addr.s_addr = htonl(INADDR_ANY);
    if( bind(komunikasiA , (struct sockaddr*)&kom, sizeof(kom) ) == -1)die("bind");
}
```

////////////////////////////////////

```

void runLuaProgram() {
    int a = system("cd ../Motion; screen -S dcm lua run_dcm.lua; screen -S player lua walk_server.lua;");
}

void * streamCamera(void * argument){
    kamera.streaming();
}

void * settingCamera(void * argument){
    kamera.setting();
}

void connect(){
    if ((s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    if ((l=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    memset((char *) &si_me, 0, sizeof(si_me));
    memset((char *) &Remote, 0, sizeof(Remote));
    Remote.sin_family = AF_INET;
    Remote.sin_port = htons(OUT_PORT);
    if (inet_aton("127.0.0.1" , &Remote.sin_addr) == 0){
        fprintf(stderr, "inet_aton() failed\n");
        exit(1);
    }
}

void motion(){
    if ((sm=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    if ((lm=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    memset((char *) &Motion, 0, sizeof(Motion));
    memset((char *) &RemoteMotion, 0, sizeof(RemoteMotion));
    RemoteMotion.sin_family = AF_INET;
    RemoteMotion.sin_port = htons(OUT_PORT_MOTION);
    if (inet_aton("127.0.0.1" , &RemoteMotion.sin_addr) == 0){
        fprintf(stderr, "inet_aton() failed\n");
        exit(1);
    }
}

void DisconnectMotion() {
    close(sm);
}

```

```

}

void disconnect(){
    close(s);
}

//////////Pemutusan Komunikasi Robot//////////
void DisconnectCommunication() {
    close(komunikasiA);
}

//////////

bool disableHeadMove = false;
bool disableWalkControl = false;
double headPan, headTilt;

// Head Movement
char headData[BUFLEN];
void headMove(double pan,double tilt){
    sprintf(headData,"% .2f,% .2f",pan,tilt);
    if (sendto(l,headData , BUFLEN , 0 , (struct sockaddr *) &Remote, sizeof(Remote))==1)die("sendto()");
    for(int s=0; s<BUFLEN; s++){
        headData[s]=0;
    }
    headPan = pan;
    headTilt = tilt;
}

char motionData[BUFLEN];

void jalan() {
    sprintf(motionData,"k");
    if (sendto(lm,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteMotion, sizeof(RemoteMotion))==1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void robotReady() {

```

```

    sprintf(motionData, "9");
    if (sendto(lm, motionData, BUFLLEN, 0, (struct sockaddr *) &RemoteMotion, sizeof(RemoteMotion)) == -1) die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

```

void walkLeftKick() {
    sprintf(motionData, "5");
    if (sendto(lm, motionData, BUFLLEN, 0, (struct sockaddr *) &RemoteMotion, sizeof(RemoteMotion)) == -1) die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

```

void walkRightKick() {
    sprintf(motionData, "6");
    if (sendto(lm, motionData, BUFLLEN, 0, (struct sockaddr *) &RemoteMotion, sizeof(RemoteMotion)) == -1) die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

```

void Walk(double x, double y, double z) {
    sprintf(motionData, "walk,%.2f,%.2f,%.2f", x, y, z);
    if (sendto(lm, motionData, BUFLLEN, 0, (struct sockaddr *) &RemoteMotion, sizeof(RemoteMotion)) == -1) die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

/////////////////////////////////Void-void yang digunakan untuk komunikasi Robot/////////////////////////////////

```

void KirimBerhenti() {
    sprintf(motionData, "Berhenti");
    if (sendto(komunikasiB, motionData, BUFLLEN, 0, (struct sockaddr *) &RemoteKom, sizeof(RemoteKom)) == -1) die("sendto()");
}

```

```

for(int c=0; c<BUFLEN; c++){
    motionData[c]=0;
}
}

void KirimJaga() {
    sprintf(motionData,"Jaga");
    if (sendto(komunikasiB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))== -1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void KirimHilangBola() {
    sprintf(motionData,"Hilang_Bola");
    if (sendto(komunikasiB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))== -1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void KirimDapatBola() {
    sprintf(motionData,"Dapat_Bola");
    if (sendto(komunikasiB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))== -1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void KirimModeMenyerang() {
    sprintf(motionData,"Menyerang");
    if (sendto(komunikasiB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))== -1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

```

```

void KirimModeDefault() {
    sprintf(motionData,"Default");
    if (sendto(komunikasiB,motionData , BUFLLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))===-1)die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

```

void KirimJagaTendang() {
    sprintf(motionData,"JagaTendang");
    if (sendto(komunikasiB,motionData , BUFLLEN , 0 , (struct sockaddr *) &RemoteKom,
sizeof(RemoteKom))===-1)die("sendto()");
    for(int c=0; c<BUFLLEN; c++){
        motionData[c]=0;
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

//////////////////////////////////////////////////////////////////Ball Tracking////////////////////////////////////////////////////////////////

```

```

#ifdef ROBOT_4
double posPan=0, lastPosPan;//0.00083
double posTilt = -0.6, lastPosTilt;
void trackBall() {
    if(!disableHeadMove) {
        if(kamera.Ball_X != -1 && kamera.Ball_Y != -1) {
            double errorPan = (double)kamera.Ball_X - 160.0;
            double errorTilt = (double)kamera.Ball_Y - 120.0;
            double PPan = errorPan * 0.0005; // Tune in Kp Pan //0.0009 <<---pertama 0.0005
            double PTilt = errorTilt * 0.0004; // Tune in Kp Tilt //0.0008
            posPan += PPan;
            posTilt += PTilt*-1;
            if(posPan>= 1.8) {
                posPan = 1.8;
            }
            if(posPan<= -1.8) {
                posPan = -1.8;
            }
            if(posTilt<= -1.6) {
                posTilt = -1.6;
            }
        }
    }
}

```

```

    }
    if(posTilt>= -0.2) {
        posTilt = -0.2;
    }
    headMove(posPan,posTilt);
    lastPosPan = posPan;
    lastPosTilt = posTilt;
}
}
#endif
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////Ball Searching/////////////////////////////////////////////////////////////////
#ifdef ROBOT_4
long updateInterval=0;
double panRate=0.75, tiltRate=0.4;

void searchBall(int isRobotMove){
    //printf("Search ball\r\n");
    if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
        if(!disableHeadMove){
            if(updateInterval>=5) { //5
                updateInterval=0;
                posTilt+=tiltRate;
                if(posTilt <= -1.8 || posTilt >= -0.2) {
                    tiltRate *= -1;
                    posPan += panRate;
                    if(posPan >= 1.7||posPan <=-1.7) {
                        panRate *= -1;
                    }
                }
            }
            if(posTilt <= -1.6){
                posTilt= -1.6;
            }
            if(posTilt >= -0.2) {
                posTilt = -0.2;
            }
            if(posPan >= 1.7) {
                posPan = 1.7;
            }
        }
    }
}

```

```

    }
    if(posPan <= -1.7) {
        posPan = -1.7;
    }
    headMove(posPan,posTilt);
    //printf("posPan = %.2f posTilt = %.2f\r\n",posPan, posTilt);
}
else{
    updateInterval++;
}
if(!disableWalkControl) {
    if(!isRobotMove) {
        Walk(0.00, 0.00, 0.00);
    }
    else if(isRobotMove == 1) {
        Walk(0.03, 0.00, 0.01);
    }
    else if(isRobotMove == 2) {
        Walk(0.03, 0.00, 0.5);
    }
    else if(isRobotMove == 3) {
        Walk(0.03, 0.00, 0.1);
    }
    else if(isRobotMove == 4) {
        Walk(-0.03, 0.00, 0.00);
    }
    else if(isRobotMove == 5) {
        Walk(0.00,0.00,-0.4);
    }
}
}
}
else {
    headMove(posPan, posTilt);
}
}
#endif
////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////Ball Following////////////////////////////////////////////////////////////////

```

```

#ifdef ROBOT_4
int countNear;
int countReadyKick;
int followBall(int mode){ //0 normal, 1 sambil belok
    int retVal = 0;
    double SetPointPan = 0;
    double SetPointTilt = -0.05;//-0.25
    trackBall();
    double errorfPan = posPan - SetPointPan;
    double errorfTilt = posTilt - SetPointTilt;

//    double PyMove = errorfPan*0.02; // 0.6
    double PyMove = 0; // 0.6
    double PxMove = 0;
    if(posTilt > -1.1 && posPan > -0.2 && posPan <0.2) {
        mode = 1;
    }
    // Jarak untuk nendang
    if(posTilt > -0.3 && posPan < 0.5 && posPan > -0.5 && kamera.Ball_X != -1 && kamera.Ball_Y != -1){
        countReadyKick++;
    }
    else {
        countReadyKick=0;
    }
    // Jarak untuk baca gawang
    // -0.15
    if(posTilt > -0.8 && posPan < 0.5 && posPan > -0.5 && kamera.Ball_X !=-1 && kamera.Ball_Y!=-1){
        countNear++;
    }
    else{
        countNear=0;
    }
    if(countNear>=1) {
        retVal = 1;
        countNear=0;
    }

    if(countReadyKick>=1) {
        PxMove = 0;
        if(posPan > 0) {

```

```

        retVal = 3;
    }
    else {
        retVal = 2;
    }
    countReadyKick=0;
}
else{
    if(posTilt > -0.2) {
        PxMove = 0.005;
    }
    else if(posTilt > -0.8) {
        PxMove = 0.015;
    }
    else {
        PxMove = 0.040;
    }
}

if(!disableWalkControl) {
    if(mode==0) {
        // Mode differential walking
        if(errorfPan > -0.5 && errorfPan < 0.5) {
            Walk(PxMove,0.00,PyMove);
        }
        else {
            // PID for rotation
            PyMove = errorfPan*0.25;
            Walk(0.00,0.00,PyMove);
        }
    }
    else if(mode==1) {
        // Mode omnidirectional walking
        if(errorfPan > -0.5 && errorfPan < 0.5) {
            PxMove = errorfTilt * 0.04 * -1;
            PyMove = errorfPan * 0.045;

            Walk(PxMove,PyMove,0.00);
        }
        else {

```

```

        PyMove = errorfPan*0.3;
        //PxMove = errorfTilt*0.04*-1;
        Walk(0.00,0.00,PyMove);
    }
}
}
return retVal;
}
#endif
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////Checking Lost Ball/////////////////////////////////////////////////////////////////
int countBallLost = 0;
int countBallFound = 0;
int ballLostRetVal = 0;

int ballLost(int threshold) {
    if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
        countBallFound = 0;
        countBallLost++;
        if(countBallLost>=threshold) {
            ballLostRetVal = 1;
        }
    }
    else {
        countBallLost = 0;
        countBallFound++;
        if(countBallFound>1) {
            ballLostRetVal = 0;
        }
    }
    return ballLostRetVal;
}
/////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////// Function Set Timer ///////////////////////////////////////////////////////////////////
int setTimer(struct timeval &tv, time_t sec) {
    gettimeofday(&tv,NULL);
    tv.tv_sec+=sec;
    return 1;
}

```

```

}
/////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////// Function For Check Timer/////////////////////////////////////////////////////////////////
int checkTimer(struct timeval &tv, time_t sec) {
    struct timeval ctv;
    gettimeofday(&ctv,NULL);
    if((ctv.tv_sec > tv.tv_sec)) {
        gettimeofday(&tv,NULL);
        tv.tv_sec+=sec;
        return 1;
    }
    else {
        return 0;
    }
}
/////////////////////////////////////////////////////////////////

    bool boolkirimdataMenyerang = true;
    bool boolkirimdataBertahan = false;
    bool boolkirimdataDefault = false;

int main(void){
    bool strategi_serang = false, strategi_bertahan = false, strategi_Default = false;
    runLuaProgram();
    connect();
    motion();
    komunikasi();
    kamera.initialize(5, camWidth, camLength);
    robotReady();
    pthread_create(&threadStreaming,NULL, streamCamera, NULL);
    pthread_create(&threadCamSetting,NULL, settingCamera, NULL);
    kamera.loadconfig();
    struct timeval    timerRobotIsReady, timerState1, timerState2, timermulai;
    bool robotIsReady = false;
    setTimer(timerRobotIsReady, 7);

    while(robotIsReady == false) {
        if(checkTimer(timerRobotIsReady, 7)==true) {
            robotIsReady = true;

```

```

    }
    else {
        robotIsReady = false;
    }
}
jalan();
headMove(0,-0.7);
bool boolkirimdataMenyerang = true;
bool boolkirimdataBertahan = false;
bool boolkirimdataDefault = false;
bool start = true;
while(1) {
    kamera.processing();

//    printf("Tilt %g Pan %g\n", headTilt, headPan);
    if(start == true) {
        if(boolkirimdataMenyerang) {
            ///////////Kirim data Strategi menyerang ke Robot2//////////
            KirimModeMenyerang();
            //////////////////////////////////////

            boolkirimdataMenyerang = false;

            strategi_serang = true;
            start = false;
        }
        if(boolkirimdataDefault) {
            ///////////Kirim data Strategi default ke Robot2//////////
            KirimModeDefault();
            //////////////////////////////////////

            strategi_Default = true;
            boolkirimdataDefault = false;
            start = false;
        }
    }
}

////////// Strategi DEFAULT //////////
if(strategi_Default == true) {
    printf("strategi D_faultttttttttttt\n");
}

```

```

//////////Terima data Gerakan dari Robot1//////////
if ((terima = recvfrom(komunikasiA,buf, BUFLLEN, 0, (struct sockaddr *)
&komunikasi_luar,(socklen_t*)&com_len)) == -1)die("recvfrom()");
    printf("terima data %s\n\r",buf);
//////////

//////////Terima data Komunikasi dari Robot2 berupa HilangBola//////////
if(strcmp(buf,"Hilang_Bola")==0) {
    if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
        ////////////Kirim Data Komunikasi Ke Robot2 berupa KirimHilangBola//////////
        KirimHilangBola();
        ////////////

        searchBall(0);
    }
    else if(kamera.Ball_X != -1 && kamera.Ball_Y != -1) {
        ////////////Kirim Data Komunikasi Ke Robot2 berupa KirimBerhenti//////////
        KirimBerhenti();
        ////////////

        followBall(0);
    }
}
//////////

//////////Terima data Komunikasi dari Robot2 berupa DapatBola//////////
else if(strcmp(buf,"Dapat_Bola")==0) {
    if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
        ////////////Kirim Data Komunikasi Ke Robot2 berupa KirimJaga//////////
        KirimJaga();
        ////////////

        searchBall(0);
    }
    else if(kamera.Ball_X != -1 && kamera.Ball_Y != -1) {
        followBall(0);
        if(headTilt > -1.0) {
            ////////////Kirim Data Komunikasi Ke Robot2 berupa KirimDapatBola//////////
            KirimDapatBola();

```



```

    }
    else if(sudahTendang) {
        if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
            Walk(0.00,0.00,0.00);

            ///////////Kirim Data Komunikasi Ke Robot2 berupa KirimHilangBola//////////
            KirimHilangBola();
            //////////////////////////////////////
        }
    }
    else {
        Walk(0.00,0.00,0.00);

        ///////////Kirim Data Komunikasi Ke Robot2 berupa KirimDapatBola//////////
        KirimDapatBola();
        //////////////////////////////////////
    }
}
}
}
////////////////////////////////////

//////////Terima data Komunikasi dari Robot2 berupa DapatBola//////////
if(strcmp(buf,"Dapat_Bola")==0) {
    if(sudahTendang) {
        if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {
            Walk(0.00,0.00,0.00);

            ///////////Kirim Data Komunikasi Ke Robot2 berupa KirimJagaTendang//////////
            KirimJagaTendang();
            //////////////////////////////////////
        }
    }
    else {
        Walk(0.00,0.00,0.00);

        ///////////Kirim Data Komunikasi Ke Robot2 berupa KirimJagaTendang//////////
        KirimJagaTendang();
        //////////////////////////////////////
    }
}
}
else if(!sudahTendang) {
    if(kamera.Ball_X == -1 && kamera.Ball_Y == -1) {

```

```

searchBall(0);

//////////Kirim Data Komunikasi Ke Robot2 berupa KirimHilangBola//////////
KirimHilangBola();
//////////

}
else {
followBall(0);
if(headTilt > -0.4) {
walkLeftKick();
sudahTendang = true;
}
else {
//////////Kirim Data Komunikasi Ke Robot2 berupa KirimDapatBola//////////
KirimDapatBola();
//////////
}
}
}
}
}

//////////

//////////Terima data Komunikasi dari Robot2 berupa GantiStrategi//////////
if(strcmp(buf,"GantiStrategi")==0) {
strategi_serang = false;
start = true;
boolkirimdataDefault = true;
boolkirimdataMenyerang = false;
}
//////////

}

//////////

}
kamera.keluar();
system("killall screen;");
disconnect();
DisconnectMotion();
DisconnectComunication();
delete buf;
return 0;}

```

## 1.2 ROBOT 2

```
////////////////////////////////////  
////////////////////////////////////  
//////////////////////////////////// ROBOT 2 //////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////
```

```
#include <stdio.h>  
#include "opencv2/opencv.hpp"  
#include <time.h>  
#include <sys/socket.h>  
#include <arpa/inet.h>  
#include <string.h>  
#include <pthread.h>  
#include <libgen.h>  
#include "BlobResult.h"  
#include <sys/time.h>  
#include "minIni.h"  
#include <unistd.h>  
#include <pthread.h>  
#include <fcntl.h>  
#include <errno.h>  
#include <termios.h>  
#include <termio.h>  
#include <math.h>  
#include <cmath>  
#include <stdio.h>  
#include <unistd.h>  
#include <limits.h>  
#include <string.h>  
#include <libgen.h>  
#include <signal.h>  
#include <time.h>  
#include "BarelangFC-Vision.h"  
#include "mjpg_streamer.h"  
#include "LinuxDARwIn.h"  
#include "StatusCheck.h"  
#include "VisionMode.h"  
#define camWidth 320
```

```

#define camLength 240
#define PI          3.14159265359
#define BALL_ERROR_THRESHOLD 7
#define THRESHOLD_EKSEKUSI 1
#ifdef MX28_1024
#define MOTION_FILE_PATH  "../Lib_Bareleng-FC/Data/motion_1024.bin"
#else
#define MOTION_FILE_PATH  "../Lib_Bareleng-FC/Data/motion_4096.bin"
#endif
#define INI_FILE_PATH    "../Lib_Bareleng-FC/Data/config.ini"
#define SCRIPT_FILE_PATH "script.asc"
#define U2D_DEV_NAME0    "/dev/servo"

//////////Port yang digunakan untuk melakukan komunikasi//////////
#define PORT_OUT      31
#define PORT_IN       30
#define BUFLen        100
//////////

using namespace BarelengFC;
BarelengVision kamera;

//////////Inisialisasi Variabel Komunikasi 2 Robot//////////
struct sockaddr_in KomunikasiSend, remoteKomunikasi, si_other;
int Kom1, Kom2;
int s, l, i, slen = sizeof(si_other), rcv_len;
struct sockaddr_in com, RemoteCom, komunikasi_luar;
int comA, comB, comC, com_len = sizeof(komunikasi_luar), terima;;
char buf[BUFLen];
//////////

long xCount = 0;
double faktorX = 30;
double faktorY = 60;
double posPanAngle=0.0;
double posTiltAngle=0.0;
double HeadPan;
double HeadTilt;
double tiltScanParam[]={ 60,110};
int countTilt;

```

```

unsigned char buff;
int fd;

LinuxCM730 linux_cm730(U2D_DEV_NAME0);
CM730 cm730(&linux_cm730);
Point2D ball_pos, red_pos, yellow_pos, blue_pos;
BallTracker tracker = BallTracker();
BallFollower follower = BallFollower();
pthread_t threadStreaming, threadCamSetting, threadStrategy, threadKom;

```

```

void * streamCamera(void * argument) {
    kamera.streaming();
}
void * settingCamera(void * argument) {
    kamera.setting();
}
void die(char * s) {
    perror(s);
    exit(1);
}

```

/////////////////////////////////Pengiriman Komunikasi melalui IP address////////////////////////////////

```

void komunikasiSend() {
    if((comA=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    if((comB=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)die("socket");
    memset((char *) &com, 0, sizeof(com));
    memset((char *) &RemoteCom, 0, sizeof(RemoteCom));
    RemoteCom.sin_family = AF_INET;
    RemoteCom.sin_port = htons(PORT_OUT);
    if (inet_aton("192.168.1.8" , &RemoteCom.sin_addr) == 0){
        fprintf(stderr, "inet_aton() failed\n");
        exit(1);
    }
    com.sin_family = AF_INET;
    com.sin_port = htons(PORT_IN);
    com.sin_addr.s_addr = htonl(INADDR_ANY);
    if(bind(comA,(struct sockaddr*)&com,sizeof(com))== -1)die("bind");
}

```

////////////////////////////////

```

void change_current_dir() {
    char exepath[1024] = {0};
    if(readlink("/proc/self/exe", exepath, sizeof(exepath)) != -1) {
        if(chdir(dirname(exepath))) {
            fprintf(stderr, "chdir error!! \n");
        }
    }
}

void searchBallFar() {
    if(xCount >= 10) { //4
        xCount = 0;
        posPanAngle += faktorX;
        if(posPanAngle >= 80 || posPanAngle <= -80) {
            if(posPanAngle >= 80) {
                posPanAngle = 80;
            }
            if(posPanAngle <= -80) {
                posPanAngle = -80;
            }
            faktorX *= -1;
            countTilt++;
            if(countTilt > 1) {
                countTilt = 0;
            }
            posTiltAngle = tiltScanParam[countTilt];
        }
        //printf("Pan = %.2f\t Tilt = %.2f\n", posPanAngle, posTiltAngle);
        Head::GetInstance()->MoveByAngle(posPanAngle, posTiltAngle);
    }
    else{
        xCount++;
    }
}

void jalan(double x, double y, double z) {
    Walking::GetInstance()->X_MOVE_AMPLITUDE=x;
    Walking::GetInstance()->Y_MOVE_AMPLITUDE=y;
    Walking::GetInstance()->A_MOVE_AMPLITUDE=z;
    Walking::GetInstance()->Start();
}

```

```

char motionData[BUFLEN];

/////////////////////////////////Void-void yang digunakan untuk komunikasi Robot/////////////////////////////////

void HilangBola() {
    sprintf(motionData,"Hilang_Bola");
    if (sendto(comB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteCom, sizeof(RemoteCom))==
1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void DapatBola() {
    sprintf(motionData,"Dapat_Bola");
    if (sendto(comB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteCom, sizeof(RemoteCom))==
1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void GantiStrategiDefault() {
    sprintf(motionData,"GantiStrategi");
    if (sendto(comB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteCom, sizeof(RemoteCom))==
1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

void oke() {
    sprintf(motionData, "oke");
    if (sendto(comB,motionData , BUFLEN , 0 , (struct sockaddr *) &RemoteCom, sizeof(RemoteCom))==
1)die("sendto()");
    for(int c=0; c<BUFLEN; c++){
        motionData[c]=0;
    }
}

/////////////////////////////////

/////////////////////////////////Pemutusan Komunikasi Robot/////////////////////////////////

void DisconnectCommunication() {
    close(comA);
}

```

```

}
////////////////////////////////////

void sighandler(int sig) {
    exit(0);
}

    bool bacaData = true;
    int countTendang = 0;
    bool kirimdata = false, sudahTendang = false;

int main(void) {
    signal(SIGABRT, &sighandler);
    signal(SIGTERM, &sighandler);
    signal(SIGQUIT, &sighandler);
    signal(SIGINT, &sighandler);
    change_current_dir();
    komunikasiSend();
    kamera.initialize(0, camWidth, camLength);
    pthread_create(&threadStreaming, NULL, streamCamera, NULL);
    pthread_create(&threadCamSetting, NULL, settingCamera, NULL);
    kamera.loadconfig();
    minIni* ini_file = new minIni(INI_FILE_PATH);
    bool Default = false, serang = false, kirimdata = false, sudahTendang = false;
    ////////////////////////////////// Framework Initialize //////////////////////////////////
    if(MotionManager::GetInstance()->Initialize(&cm730) == false) {
        linux_cm730.SetPortName(U2D_DEV_NAME1);
        if(MotionManager::GetInstance()->Initialize(&cm730) == false) {
            //printf("Fail to initialize Motion Manager!\n");
            return 0;
        }
    }
    Walking::GetInstance()->LoadINISettings(ini_file);
    MotionManager::GetInstance()->AddModule((MotionModule*)Action::GetInstance());
    MotionManager::GetInstance()->AddModule((MotionModule*)Head::GetInstance());
    MotionManager::GetInstance()->AddModule((MotionModule*)Walking::GetInstance());

    LinuxMotionTimer *motion_timer = new LinuxMotionTimer(MotionManager::GetInstance());
    motion_timer->Start();
    //////////////////////////////////
    MotionManager::GetInstance()->LoadINISettings(ini_file);

```

```

int firm_ver = 0;
if(cm730.ReadByte(JointData::ID_HEAD_PAN, MX28::P_VERSION, &firm_ver, 0) !=
CM730::SUCCESS) {
    fprintf(stderr, "Can't read firmware version from Dynamixel ID %d!! \n\n", JointData::ID_HEAD_PAN);
    exit(0);
}
if(0 < firm_ver && firm_ver < 27) {
    #ifdef MX28_1024
        Action::GetInstance()->LoadFile(MOTION_FILE_PATH);
    #else
        fprintf(stderr, "MX-28's firmware is not support 4096 resolution!! \n");
        fprintf(stderr, "Upgrade MX-28's firmware to version 27(0x1B) or higher.\n\n");
        exit(0);
    #endif
}
else if(27 <= firm_ver) {
    #ifdef MX28_1024
        fprintf(stderr, "MX-28's firmware is not support 1024 resolution!! \n");
        fprintf(stderr, "Remove '#define MX28_1024' from 'MX28.h' file and rebuild.\n\n");
        exit(0);
    #else
        Action::GetInstance()->LoadFile((char*)MOTION_FILE_PATH);
    #endif
}
else {
    exit(0);
}
Action::GetInstance()->m_Joint.SetEnableBody(true, true);
MotionManager::GetInstance()->SetEnable(true);
cm730.WriteByte(CM730::P_LED_PANNEL, 0x01|0x02|0x04, NULL);
Action::GetInstance()->Start(15);
while(Action::GetInstance()->IsRunning()) usleep(8*1000);
while(1) {
    StatusCheck::Check(cm730);
    kamera.processing();
    ball_pos.X = kamera.Ball_X;
    ball_pos.Y = kamera.Ball_Y;
    tracker.Process(ball_pos);
    HeadTilt = Head::GetInstance()->GetTiltAngle();
}

```

```

HeadPan = Head::GetInstance()->GetPanAngle();

Head::GetInstance()->m_Joint.SetEnableHeadOnly(true, true);
Walking::GetInstance()->m_Joint.SetEnableBodyWithoutHead(true, true);
if(bacaData) {
    ////////////Terima data Strategi dari Robot1//////////
    if ((terima = recvfrom(comA,buf, BUFLen, 0, (struct sockaddr *)
&komunikasi_luar,(socklen_t*)&com_len)) == -1)die("recvfrom()");
    //    printf("terima data strategi %s\n\r",buf);
    if(strcmp(buf,"Default")==0) {
        Default = true;
        bacaData = false;
    }
    else if(strcmp(buf,"Bertahan")==0) {
        bertahan = true;
        bacaData = false;
    }
    else if(strcmp(buf,"Menyerang")==0) {
        serang = true;
        bacaData = false;
    }
    }
    ////////////
}
//    printf("Tilt %g Pan %g\n", HeadTilt, HeadPan);
////////// Strategi DEFAULT //////////
if(Default) {
    printf("strategi D_faulttttttttttttttttttttttttttttttt\n");
    if(ball_pos.X == -1 && ball_pos.Y == -1) {
        Walking::GetInstance()->Stop();
        searchBallFar();
        ////////////Kirim data Ke Robot1 berupa HilangBola//////////
        HilangBola();
        ////////////
    }
    else if(ball_pos.X != -1 && ball_pos.Y != -1) {
        Walking::GetInstance()->Stop();
        ////////////Kirim data Ke Robot1 berupa DapatBola//////////
        DapatBola();
        ////////////
    }
}

```



```

        ///////////////////////////////////////////////////////////////////
    }
}
/////////////////////////////////////////////////////////////////
//////////Terima data Komunikasi dari Robot1 berupa Jaga//////////
else if(strcmp(buf,"Jaga")==0) {
    follower.Process(tracker.ball_position);
}
/////////////////////////////////////////////////////////////////
//////////Terima data Komunikasi dari Robot1 berupa Mundur//////////
else if(strcmp(buf,"Mundur")==0) {
    if(ball_pos.X != -1 && ball_pos.Y != -1) {
        if(HeadTilt > 50) {
            Walking::GetInstance()->Stop();
        }
        else {
            jalan(-10,0,0);
            ///////////////////////////////////////////////////////////////////
            Kirim data Ke Robot1 berupa oke//////////
            oke();
            ///////////////////////////////////////////////////////////////////
        }
    }
}
/////////////////////////////////////////////////////////////////
//////////Terima data Komunikasi dari Robot1 berupa Berhenti//////////
else if(strcmp(buf,"Berhenti")==0) {
    Walking::GetInstance()->Stop();
    if(ball_pos.X == -1 && ball_pos.Y == -1) {
        searchBallFar();
        ///////////////////////////////////////////////////////////////////
        Kirim data Ke Robot1 berupa HilangBola//////////
        HilangBola();
        ///////////////////////////////////////////////////////////////////
    }
    else if(ball_pos.X != -1 && ball_pos.Y != -1) {
        ///////////////////////////////////////////////////////////////////
        Kirim data Ke Robot1 berupa DapatBola//////////
        DapatBola();
        ///////////////////////////////////////////////////////////////////
        tracker.ball_position;
    }
}
}

```



```

////////////////////////////////////

//////////Terima data Komunikasi dari Robot1 berupa JagaTendang////////
if(strcmp(buf,"JagaTendang")==0) {
    if(!sudahTendang) {
        if(ball_pos.X == -1 && ball_pos.Y == -1) {
            searchBallFar();
            ////////////Kirim data Ke Robot1 berupa HilangBola//////////
            HilangBola();
            //////////////////////////////////////
        }
        else {
            follower.Process(tracker.ball_position);
            if(HeadTilt < 30 && HeadPan < 0) {
                Head::GetInstance()->m_Joint.SetEnableHeadOnly(true, true);
                Action::GetInstance()->m_Joint.SetEnableBodyWithoutHead(true, true);
                Action::GetInstance()->Start(12);
                if(countTendang > 15) {
                    ////////////Kirim data Ke Robot1 berupa GantiStrategiDefault//////////
                    GantiStrategiDefault();
                    //////////////////////////////////////
                    sudahTendang = true;
                }
                else {
                    countTendang++;
                }
            }
            else if(HeadTilt < 30 && HeadPan > 0) {
                Head::GetInstance()->m_Joint.SetEnableHeadOnly(true, true);
                Action::GetInstance()->m_Joint.SetEnableBodyWithoutHead(true, true);
                Action::GetInstance()->Start(13);
                if(countTendang > 15) {
                    ////////////Kirim data Ke Robot1 berupa GantiStrategiDefault//////////
                    GantiStrategiDefault();
                    //////////////////////////////////////
                    sudahTendang = true;
                }
                else {
                    countTendang++;
                }
            }
        }
    }
}

```

```

        }
        else {
            DapatBola();
        }
    }
}
else if(sudahTendang) {
    ///////////Kirim data Ke Robot1 berupa GantiStrategiDefault//////////
    GantiStrategiDefault();
    ////////////////////////////////////////
    bacaData = true;
    serang = false;
}
}
//////////////////////////////////////
//////////Terima data Komunikasi dari Robot1 berupa DapatBola////////
if(strcmp(buf,"Dapat_Bola")==0) {
    if(ball_pos.X == -1 && ball_pos.Y == -1) {
        Walking::GetInstance()->Stop();
        searchBallFar();
        ///////////Kirim data Ke Robot1 berupa HilangBola//////////
        HilangBola();
        ////////////////////////////////////////
    }
    else if(ball_pos.X != -1 && ball_pos.Y != -1) {
        tracker.ball_position;
        Walking::GetInstance()->Stop();
        ///////////Kirim data Ke Robot1 berupa DapatBola//////////
        DapatBola();
        ////////////////////////////////////////
    }
}
}
//////////////////////////////////////
}
kamera.keluar();
DisconnectCommunication();
delete buf;
return 0;}

```

## BIOGRAFI PENULIS

### Arif Muspita



Penulis dilahirkan di Batam pada tanggal 12 Februari 1994. Merupakan anak ketiga dari tiga bersaudara. Penulis telah menempuh pendidikan formal di TK AL-Hijrah, SD Negeri 007 Batam, SMP Negeri 11 Batam, SMK Teladan Batam. Setelah lulus dari SMK Teladan penulis mengikuti tes masuk di Politeknik Negeri Batam dan diterima pada jenjang Diploma III Teknik Informatika pada tahun 2012. Penulis pernah menjadi anggota dan ketua pada tim terbaik Politeknik Negeri Batam yaitu Tim Robot Barelang khususnya divisi Robot Sepak Bola. Pada tahun 2012 Polibatam menjadi juara umum tingkat regional 1 untuk semua divisi di lampung dan lolos ke nasional di semarang namun pada tahun 2013 hanya ada 2 divisi dari 4 divisi yang diikuti tim barelang yang menjadi juara dan lolos ke nasional. Selain itu penulis senang melakukan riset dalam hardware maupun software yang berkaitan dengan pemrograman. Sedikit keahlian dalam mengerti bahasa pemrograman: Visual Basic, C#, Java, HTML, PHP, C, C++. Databases : MySQL, SQLite, Sql dan Oracle. Operating System: Windows dan Linux. Hardware : Instalasi Komputer, Jaringan Komputer dan Perangkat Robot.