



**RANCANG BANGUN SISTEM
ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA
CMUCAM**

SKRIPSI

Oleh

**Slamet Purwanto
NIM 121910201126**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2015**



**RANCANG BANGUN SISTEM
ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA
CMUCAM**

SKRIPSI

Diajukan guna melengkapi Skripsi dan memenuhi salah satu syarat
Untuk menyelesaikan Program Studi Strata 1 Teknik Elektronika
Dan mencapai gelar Sarjana

Oleh

**Slamet Purwanto
NIM 121910201126**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN



*Skripsi ini merupakan sebuah awal, langkah kecil menuju lompatan besar guna
menggapai kesuksesan yang lebih baik lagi.*

Slamet Purwanto mengucapkan rasa syukur kepada...

*Allah SWT, dengan segala Keagungan dan Keajaiban-Nya yang senantiasa
mendengar do'a ku, menuntunku dari kegelapan, serta senantiasa menaungiku
dengan rahmat dan hidayah-Nya dan junjunganku
Nabi Besar Muhammad SAW dan para sahabat-Nya yang telah menjadi penerang di
duniadan suri tauladan bagi kita semua*

Karya ini saya persembahkan sebagai tanda bakti kepada

Ibu Yati, Bapak Bahar,

Kakak perempuanku Sri Wahyu Ningsih dan teman-temanku semua.

*Guru-guru SD Negeri 1 Kajarharjo, SMP Negeri 1 Kalibaru dan SMK Negeri 1
Glagah sampai Perguruan Tinggi yang terhormat, terima kasih telah memberikan
ilmu dan mendidik dengan penuh keikhlasan dan kesabaran*

Almamater Fakultas Teknik Universitas Jember.

MOTO

“Dan orang-orang yang bersungguh-sungguh berjuang di jalan Kami, niscaya akan kami tunjukkan jalan kami.”

(QS: Al-Ankabut: 69)

“Barang siapa mengenali diri sendiri maka ia akan mengenali Tuhannya, bila ia mengenali kelemahan dirinya maka ia akan tahu kekuasaan Allah SWT.”

(Sabda Nabi Muhammad Saw)

“Keridhaan Allah itu tergantung dari adanya keridhaan ibu-bapak, dan kemurkaan Allah itu tergantung dari adanya kemurkaan ibu-bapak.”

(H. R Tirmidzi)

“ Sesungguhnya guru dan dokter tidak akan berguna nasehatnya bila tidak dihormati. Bersabarlah dengan penyakitmu bila kamu menentang dokter. Dan bersabarlah dengan kebodohanmu bila kamu menentang guru.”

(Ta’limul Muta’allim)

“Cita-cita akan tercapai sejauh orang-orang akan bercita-cita. Kemuliaan akan tercapai sejauh seseorang berbuat mulia. Sesuatu yang kecil akan tampak besar bagi orang-orang yang bercita-cita kecil. Dan sesuatu yang besar akan tampak kecil bagi orang-orang yang bercita-cita besar.”

(Abu Thayyib)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Slamet Purwanto

NIM : 121910201126

Menyatakan dengan sesungguhnya bahwa Skripsi yang berjudul: “*RANCANG BANGUN SISTEM ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA CMUCAM 4*” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi mana pun serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 30 juni 2015

Yang menyatakan,

Slamet Purwanto

NIM 121910201126

SKRIPSI

**RANCANG BANGUN SISTEM
ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA
CMUCAM**

Oleh

**Slamet Purwanto
NIM 121910201126**

Pembimbing

Dosen Pembimbing Utama : Sumardi, S.T.,M.T.

Dosen Pembimbing Anggota : Satriyo Budi Utomo, S.T.,M.T.

LEMBAR PENGESAHAN

Laporan Skripsi berjudul **“RANCANG BANGUN SISTEM ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA CMUCAM ”** oleh Slamet Purwanto NIM: 121910201126 telah diuji dan disahkan oleh Jurusan Teknik Elektro Fakultas Teknik Universitas Jember pada :

Hari, tanggal : 30, Juni 2015

Tempat : Ruang Ujian 1 Lantai 3 Gedung Dekanat
Fakultas Teknik Universitas Jember

Tim Penguji

Ketua,

Sekretaris,

Sumardi, S.T.,M.T.
NIP. 19610414 198902 1 001

Satriyo Budi Utomo, S.T.,M.T
NIP. 19850126 200801 1 002

Anggota I,

Anggota II,

Widjonarko, S.T., M.T.
NIP. 19710908 199903 1 001

M. Agung Prawira N, S.T., M.T.
NIP. 19871217 201212 1 003

Mengesahkan
Dekan,

Ir. Widyono Hadi, M.T.
NIP. 19610414 198902 1 001

*Rancang Bangun Sistem Robot Humanoid Soccer Menggunakan Kamera Cmucam
(Design of Humanoid Robot Soccer System Using Camera Cmucam)*

Slamet Purwanto

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

Abstrak

Robot Humanoid Soccer adalah salah satu bentuk implementasi teknologi dalam bidang robotika yang memiliki kemampuan menirukan salah satu atau beberapa kegiatan manusia untuk bermain sepakbola. Pemrograman motion pada robot *humanoid* ini bertujuan untuk mendapatkan gerakan robot yang kompleks dari gerakan robot yang lebih pendek dan sederhana. Robot ini terdiri dari 2 bagian utama yaitu perangkat keras dan perangkat lunak. Perangkat keras terdiri dari konstruksi badan robot, servo Dynamixel XL-320, sensor kamera *CMUCam4*, mikrokontroler OpenCM9.04 dan Mikrokontroler Arduino Uno. Perangkat lunak terdiri dari program robot dengan menggunakan *software* Roboplus. Robot ini bekerja secara otomatis dengan pengontrol menggunakan input data dari OpenCM9.04 dan arduino. Pengujian sistem pada robot, dihasilkan gerakan robot sebagai pemain bola. Robot dapat bergerak dengan seimbang dengan memanfaatkan posisi servo dan design bentuk robot sehingga data bergerak secara seimbang. Gerakan robot yang dimaksud adalah : (1) robot berjalan maju, (2) robot duduk, (3) robot bergeser, (4) robot menendang ,(5) robot mampu bangun dari jatuh. Robot *humanoid soccer* dapat menendang bola dengan jarak $5 \text{ cm} \geq 75 \text{ pixels}$. Robot *humanoid* dapat melakukan gerakan bangkit berdiri ketika terjatuh yang dideteksi menggunakan sensor photodiode dengan rata-rata pembacaan $> 4.5 \text{ volt}$ untuk dalam ruangan dan Robot dapat mendeteksi bola *soccer* dengan bentuk warna merah.

Kata kunci : CMUCAM4, humanoid, humanoid soccer, robot

*Rancang Bangun Sistem Robot Humanoid Soccer Menggunakan Kamera Cmucam
(Design of Humanoid Robot Soccer System Using Camera Cmucam)*

Slamet Purwanto

Electronics Engineering Departement, Engineering Faculty, Jember University

ABSTRACT

Humanoid Robot Soccer is one form of implementation in the field of robotics technology that has the ability to mimic one or a few human activities to play football. Programming motion on the humanoid robot aims to obtain a complex robotic movements of the robot's movements are shorter and simpler. This robot consists of two main parts: hardware and software. The hardware consists of the construction of the robot, servo Dynamixel XL-320, CMUCAM4 camera sensor, microcontroller and microcontroller Arduino Uno OpenCM9.04. The software consists of a robot by using a software program Roboplus. This robot works automatically with the controller using data input from OpenCM9.04 and arduino. Testing the system on the robot, the robot motion is generated as a footballer. The robot can move in by utilizing the servo positioning and design of the robot so that data moves in a balanced manner. The movement of the robot in question is: (1) the robot walk forward, (2) the robot sits, (3) the robot shifts, (4) robots kicking, (5) the robot is able to get up from a fall. Humanoid robot can kick the soccer ball at a distance of $5\text{ cm} > = 75\text{ pixels}$. Humanoid robot can perform the movement stood up when fall is detected using a sensor photodiode with an average reading $> 4.5\text{ volts}$ for indoor and robot can detect the shape of soccer balls in red

Keywords: CMUCAM, humanoid, humanoid soccer, robot

RINGKASAN

RANCANG BANGUN SISTEM ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA CMUCAM ; Slamet Purwanto 121910201126; 2015:

120 halaman; Program Studi Strata Satu (S1) elektro, Jurusan Teknik Elektro, Fakultas Teknik Universitas Jember.

Skripsi ini sebagai penerapan teknologi elektronika pada masyarakat terutama dalam sebuah pendidikan dan teknologi. Robot *Humanoid Soccer* adalah salah satu bentuk implementasi teknologi dalam bidang robotika yang memiliki kemampuan menirukan salah satu atau beberapa kegiatan manusia seperti bermain sepakbola. Robot ini terdiri dari 2 bagian utama yaitu perangkat keras dan perangkat lunak. Perangkat keras terdiri dari konstruksi badan robot, servo Dynamixel XL-320, sensor Kamera *CMUCAM4*, mikrokontroler OpenCM9.04 dan Mikrokontroler Arduino Uno. Perangkat lunak terdiri dari program robot dengan menggunakan software Roboplus. Robot ini bekerja secara otomatis dengan pengontrol menggunakan input data dari OpenCM9.04 dan arduino. Pada pemrograman dari dua buah mikrokontroler yang merupakan pusat pengolahan data digunakan masukan dan keluaran data yang saling terhubung satu sama lain. Mikrokontroler berfungsi sebagai pembaca data masukan dari kamera *cmucam4* dan OpenCM9.04 berfungsi sebagai pemberi perintah untuk menggerakkan servo sehingga robot bergerak.

Robot dapat bergerak dengan seimbang dengan memanfaatkan posisi servo dan desain bentuk robot sehingga data bergerak secara seimbang. Gerakan robot yang dimaksud adalah : (1) robot berjalan maju, (2) robot duduk, (3) robot bergeser, (4) robot menendang ,(5) robot mampu bangun dari jatuh. Robot *humanoid soccer* dapat menendang bola dengan jarak $5 \text{ cm} \geq 75 \text{ pixels}$. Robot *humanoid* dapat melakukan gerakan bangkit berdiri ketika terjatuh yang dideteksi menggunakan sensor photodiode dengan rata-rata pembacaan $> 4.5 \text{ volt}$ untuk dalam ruangan dan Robot dapat mendeteksi bola *soccer* dengan bentuk warna merah.

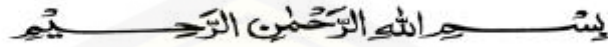
SUMMARY

Design of Humanoid Robot Soccer System Using Camera Cmucam ; Slamet Purwanto 121910201126; 2015: 120 pages; Strata satu (S1) Electro, Department of Electrical Engineering, Faculty of Engineering, University of Jember.

This thesis as the application of electronics technology to the public, especially in an education and technology. Humanoid Robot Soccer is one form of implementation in the field of robotics technology that has the ability to mimic one or a few human activities such as playing football. This robot consists of two main parts: hardware and software . The hardware consists of the construction of the robot , servo Dynamixel XL - 320 , CMUCAM4 camera sensor, microcontroller Arduino Uno and microcontroller OpenCM9.04 . The software consists of a robot by using a software program Roboplus . This robot works automatically with the controller using data input from OpenCM9.04 and Arduino . In the programming of microcontroller which is two data processing center in use input and output data are connected to each other . Microcontroller serves as a reader input data from the camera OpenCM9.04 cmucam4 and serves as a conduit commands to move the robot servo that moves.

The robot can move in by utilizing the servo positioning and design of the robot so that data moves in a balanced manner . The movement of the robot in question is : (1) the robot walk forward , (2) the robot sits , (3) the robot shifts , (4) robots kicking , (5) the robot is able to get up from a fall. Humanoid robot can kick the soccer ball at a distance of $5 \text{ cm} \geq 75 \text{ pixels}$. Humanoid robot can perform the movement stood up when fall is detected using a sensor photodiode with an average reading $> 4.5 \text{ volts}$ for indoor and robot can detect the shape of soccer red ball .

PRAKATA



Puji syukur ke hadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga laporan skripsi yang berjudul “**RANCANG BANGUN SISTEM ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA CMUCAM**”, dapat terselesaikan dengan baik. Laporan Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata Satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Terselesaikannya laporan Skripsi ini tidak terlepas dari bantuan berbagai pihak, oleh karena itu disampaikan ucapan terimakasih kepada:

1. Allah SWT, dengan segala Keagungan dan Keajaiban-Nya yang senantiasa mendengar do'aku, menuntunku dari kegelapan, serta senantiasa menaungiku dengan rahmat dan hidayah-Nya;
2. Nabi Muhammad SAW, yang telah menjadi penerang di dunia dan suri tauladan bagi kita semua;
3. Ir. Widyono Hadi, M.T., selaku Dekan Fakultas Teknik Universitas Jember;
4. Mahros Darsin, S.T., M.Sc selaku pembantu Dekan I Fakultas Teknik Universitas Jember;
5. Dr. Triwahju Hardiyanto, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Jember;
6. Bapak Dedy Kurnia Setiawan, S.T., M.T. selaku Ketua Program Studi Strata Satu (S1) Teknik Elektro Universitas Jember;
7. Bapak Sumardi, S.T., M.T.,selaku Dosen Pembimbing Utama dan Bapak Satriyo Budi Utomo, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu dan pikiran serta perhatiannya guna memberikan bimbingan dan pengarahan demi terselesaikannya skripsi ini;

8. Dr. Azmi Shaleh, S.T., M.T., selaku Dosen Pembimbing Akademik (DPA) yang telah memberikan bimbingan dan pengarahannya;
9. Widjonarko, S.T., M.T. dan M. Agung Prawira N, S.T., M.T. selaku Tim Penguji Tugas Akhir yang telah meluangkan waktu dan pikiran serta perhatiannya guna memberikan pengarahan demi terselesaikannya penulisan laporan skripsi ini;
10. Teman Alumni Diploma Tiga (DIII) Yoga Pratama atas bimbingan dan kerjasama.
11. Sivitas Akademika Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.
12. Teman-teman seperjuangan S1 Elektro 2012 Universitas Jember, "tanpa kalian saya bukan apa-apa";
13. Semua pihak yang telah membantu dalam penulisan karya serta laporan skripsi ini.

Semoga laporan Skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro, kritik dan saran diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan diharapkan dapat dikembangkan untuk penelitian-penelitian selanjutnya.

Jember, 30 Juni 2015

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN	vii
ABSTRAK	viii
<i>ABSTRACT</i>	ix
RINGKASAN	x
<i>SUMMARY</i>	xi
PRAKATA	xii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Sistem Aktifasi dan kontrol robot	4

2.2 Pengolahan Citra	5
2.2.1 Color Filtering	5
2.2.2 Sistem Warna RGB	6
2.2.3 Segmentasi Warna	8
2.2.4 Thresholding	8
2.2.5 Erosi	9
2.2.6 Dilasi	10
2.2.7 Canny Edge Detection	11
2.3 Arduino	11
2.3.1 Arduino Uno R3	13
2.2.1.1 Power	13
2.2.1.2 Memory	14
2.2.1.3 Input Dan Output	15
2.2.1.4 IComSat v1.1 -SIM900 GSM/GPRS shield	15
2.2.1.4.1 Fitur IComSat v1.1 SIM900 GSM/GPRS shield	16
2.2.1.4.2 Spesifikasi IComSat SIM900 GSM/GPRS shield	16
2.3.2 IO Expansion Shield For Arduino(V5)	16
2.4 Mikrokontroler OpenCM 9.04	17
2.5 CMUCam4	20
2.6 Motor Servo Dynamixel XL-320	25
2.7 Sensor Photodiode	27
BAB 3. METODOLOGI PENELITIAN	29
3.1 Tempat dan Waktu Penelitian	29
3.2 Studi Literatur	29
3.3 Pengumpulan dan Pengujian Data	29

3.4 Perancangan Sistem	30
3.4.1 Sistem Pencarian	32
3.4.2 Perangkat Keras	32
3.4.2.1 Komponen yang di gunakan pada perancangan	
Robot <i>humanoid soccer</i>	35
3.4.2.2 Rangkaian Arduino UNO	45
3.4.2.3 OpenCM 9.04	46
3.4.2.3 Rangkain photodioda	47
3.4.3 PERANGKAT LUNAK	48
3.4.3.1 Proses citra bentuk bola <i>soccer</i>	48
3.4.3.2 <i>Flowchart</i> Robot <i>Humanoid Soccer</i>	48
3.4.3.2 <i>Flowchart</i> Scanning Bola <i>Soccer</i>	49
3.4.3.2 <i>Flowchart</i> program kontrol servo robot	50
BAB 4. HASIL DAN PEMBAHASAN	52
4.1 Pengujian dan Analisis	52
4.1.1 Analisis <i>posisi 16 motor servo pada robot soccer</i>	52
4.1.2 Pengujian Sensor kamera <i>CMUCAM4</i>	71
4.1.3 Pengujian Sensor photodioda	73
4.1.4 Pengujian keseluruhan Robot <i>Humanoid soccer</i>	75
BAB 5. PENUTUP	78
5.1 Kesimpulan	78
5.2 Saran	79
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Sistem.....	4
Gambar 2.2 Gambar Derajat Keabuan.....	6
Gambar 2.3 Nilai warna RGB dalam <i>Hexadesimal</i>	7
Gambar 2.4 Komposisi warna RGB	7
Gambar 2.5 Contoh segmentasi api pada gambar	8
Gambar 2.6 Hasil <i>Tresholding</i> warna merah	9
Gambar 2.7 Proses Erosi	10
Gambar 2.8 Proses Dilasi	10
Gambar 2.9 Arduino UNO R3 tampak belakang.....	12
Gambar 2.10 Arduino UNO R3 tampak depan	12
Gambar 2.11 <i>IO Expansion Shield</i> untuk Arduino	17
Gambar 2.12 Opencm 9.04	18
Gambar 2.13 Pin out opencm 9.04	18
Gambar 2.14 Bentuk <i>hardware cmucam4</i>	23
Gambar 2.15 <i>Cmucam4</i>	24
Gambar 2.16 Bentuk Fisik Motor Servo	25
Gambar 2.17 Konfigurasi posisi sudut servo	26
Gambar 2.18 Rangkaian sensor photodiode	28
Gambar 3.1 Diagram Blok Sistem	30
Gambar 3.2 Diagram sistem <i>Scanning</i> bola <i>soccer</i>	31
Gambar 3.3 Diagram Sistem kerja robot	32
Gambar 3.4 Desain Badan Robot <i>Humanoid Soccer</i>	33
Gambar 3.5 Servo Dynamixel XL-320	35

Gambar 3.6 <i>Batery Li-ion LB-040</i>	35
Gambar 3.7 OpenCM 9.04 C	36
Gambar 3.8 <i>Batery Li-ion Box LB-040</i>	36
Gambar 3.9 DMF B04	36
Gambar 3.10 Dmf B05	37
Gambar 3.11 SPL 2b2	37
Gambar 3.12 SPD 3B6.....	37
Gambar 3.13 SPD 3B7.....	38
Gambar 3.14 DMF B03	38
Gambar 3.15 SPU 5.....	38
Gambar 3.16 DMF B07.....	38
Gambar 3.17 SPD 3B3.....	39
Gambar 3.18 DMF B01.....	39
Gambar 3.19 DMF B06.....	39
Gambar 3.20 SPD 4B5.....	39
Gambar 3.21 DMF F01.....	40
Gambar 3.22 DMF B02.....	40
Gambar 3.23 XL horn	40
Gambar 3.24 SPD 7V4.....	41
Gambar 3.25 SPO 5 gear	41
Gambar 3.26 STR 20.....	41
Gambar 3.27 Skrup SRV 1LSH	42
Gambar 3.28 CHS M2 0x04 TAP	42
Gambar 3.29 CHS M2 0x06 TAP	42
Gambar 3.30 Horn screw	43

Gambar 3.31 SRV 1SP	43
Gambar 3.32 SRV 1SH	43
Gambar 3.33 SRV 1 LSP	44
Gambar 3.34 DXL cable small 13 cm	44
Gambar 3.35 DXL cable small 16 cm	44
Gambar 3.36 DXL small 11 cm	45
Gambar 3.37 <i>Battery cable</i>	45
Gambar 3.38 Design <i>scematic</i> arduino uno.....	45
Gambar 3.39 Rangkaian OpenCM9.04.....	46
Gambar 3.40 Rangkaian sensor photodiode.....	47
Gambar 3.41 <i>Flowchart robot humnoid soccer</i>	48
Gambar 3.42 <i>Flowchart scanning bola soccer</i>	49
Gambar 3.43 <i>Flowchart</i> kontrol servo robot	50
Gambar 4.1 Robot dalam kondisi jalan pada waktu 0 detik.....	59
Gambar 4.2 Robot dalam kondisi jalan pada waktu 0.05 detik.....	59
Gambar 4.3 Robot dalam kondisi jalan pada waktu 0.5 detik	60
Gambar 4.4 Robot dalam kondisi jalan pada waktu 0.2 detik	60
Gambar 4.5 Robot dalam kondisi jalan pada waktu 0.3 detik	61
Gambar 4.6 Robot dalam kondisi jalan pada waktu 0.4 detik	61
Gambar 4.7 Robot dalam kondisi jalan pada waktu 0.5 detik.....	62
Gambar 4.8 Robot dalam kondisi jalan pada waktu 0.6 detik	62
Gambar 4.9 Robot kondisi bangun dari jatuh pada waktu 0.2 detik....	63
Gambar 4.10 Robot kondisi bangun dari jatuh pada waktu 0.6 detik.....	63
Gambar 4.11 Robot kondisi bangun dari jatuh pada waktu 1.2 detik	64
Gambar 4.12 Robot kondisi bangun dari jatuh pada waktu 1.4 detik....	64

DAFTAR TABEL

Tabel 2.1	Kombinasi warna dalam <i>Hexsadesimal</i>	8
Tabel 3.1	Keterangan Waktu Pelaksanaan.....	29
Tabel 3.2	Komponen perancangan robot	36
Tabel 4.1a	Data sudut servo robot dalam kondisi jalan	53
Tabel 4.1b	Data sudut servo robot dalam kondisi jalan	54
Tabel 4.1c	Data sudut servo robot dalam kondisi jalan	54
Tabel 4.2a	Data sudut servo robot dalam bangun dari jatuh	55
Tabel 4.2b	Data sudut servo robot dalam bangun dari jatuh	55
Tabel 4.2c	Data sudut servo robot dalam bangun dari jatuh	56
Tabel 4.3a	Data sudut servo robot dalam menendang	56
Tabel 4.3b	Data sudut servo robot dalam menendang	57
Tabel 4.4	Data sudut servo robot dalam posisi duduk	57
Tabel 4.5	Data sudut servo robot dalam posisi bergeser	58
Tabel 4.6a	Data sudut servo dan gambar robot kondisi jalan.....	59
Tabel 4.6b	Data sudut servo dan gambar robot kondisi jalan.....	59
Tabel 4.6c	Data sudut servo dan gambar robot kondisi jalan.....	60
Tabel 4.6d	Data sudut servo dan gambar robot kondisi jalan.....	60
Tabel 4.6e	Data sudut servo dan gambar robot kondisi jalan.....	61
Tabel 4.6f	Data sudut servo dan gambar robot kondisi jalan.....	61
Tabel 4.6g	Data sudut servo dan gambar robot kondisi jalan.....	62
Tabel 4.6h	Data sudut servo dan gambar robot kondisi jalan.....	62
Tabel 4.7a	Data sudut servo dan gambar robot bangun dari jatuh.....	63
Tabel 4.7b	Data sudut servo dan gambar robot bangun dari jatuh.....	63

Tabel 4.7c	Data sudut servo dan gambar robot bangun dari jatuh.....	64
Tabel 4.7d	Data sudut servo dan gambar robot bangun dari jatuh.....	64
Tabel 4.7e	Data sudut servo dan gambar robot bangun dari jatuh.....	65
Tabel 4.7f	Data sudut servo dan gambar robot bangun dari jatuh.....	65
Tabel 4.7g	Data sudut servo dan gambar robot bangun dari jatuh.....	66
Tabel 4.8a	Data sudut servo dan gambar robot dalam menendang	66
Tabel 4.8b	Data sudut servo dan gambar robot dalam menendang	67
Tabel 4.8c	Data sudut servo dan gambar robot dalam menendang	67
Tabel 4.8d	Data sudut servo dan gambar robot dalam menendang	68
Tabel 4.8e	Data sudut servo dan gambar robot dalam menendang	68
Tabel 4.9a	Data sudut servo dan gambar robot dalam duduk.....	69
Tabel 4.9b	Data sudut servo dan gambar robot dalam duduk.....	69
Tabel 4.9c	Data sudut servo dan gambar robot dalam duduk.....	70
Tabel 4.10a	Data sudut servo dan gambar robot bergeser	70
Tabel 4.10b	Data sudut servo dan gambar robot bergeser	71
Tabel 4.10c	Data sudut servo dan gambar robot bergeser	71
Tabel 4.11	Data <i>pixel CMUCAM4</i> dengan jarak posisi bola	72
Tabel 4.12	Data <i>X axis CMUCAM4</i> dengan jarak posisi bola Bergeser Ke kanan ke kiri.....	73
Tabel 4.13	Data <i>Y axis CMUCAM4</i> dengan jarak posisi bola bergeser ke atas dan kebawah	73
Tabel 4.14	Data pengujian saat robot jatuh kebelakang dan jatuh ke depan.....	75

BAB I. PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan teknologi elektronika begitu cepat, diiringi dengan semakin bertambahnya kebutuhan manusia di bidang elektronika, dimana teknologi elektronika mulai sudah sangat dibutuhkan pada kehidupan sehari-hari. Robotika merupakan salah satu contoh bidang elektronika yang saat ini sedang berkembang pesat. Robotika yang dikembangkan saat ini merupakan robot yang memiliki fungsi-fungsi tertentu yang dapat membantu manusia. Berbagai bentuk robot telah diciptakan saat ini seperti robot berbentuk: mobile, tank, serangga, anjing, ular maupun *humanoid*.

Robot *humanoid soccer* memiliki persendian selayaknya sendi manusia yang berfungsi untuk membantu kemudahan dalam bergerak. Sendi pada robot ini dirancang dengan besar derajat kebebasan tertentu sesuai dengan fungsinya. Robot *humanoid* memiliki keseimbangan yang akurat. Sehingga tanpa bantuan sensor keseimbangan robot ini dapat bergerak bebas.

Gerakan pada robot *humanoid* dapat dihasilkan dengan cara mengatur putaran motor servo yang berfungsi sebagai sendi robot. Serangkaian motor servo yang digunakan pada robot *humanoid* dapat dikontrol dengan menggunakan piranti *programable controller*. Gerakan robot yang kompleks dan panjang dapat diperoleh dengan cara membuat gerakan-gerakan robot yang lebih pendek dan sederhana yang dihubungkan secara runtut dengan waktu yang tepat.

Penelitian ini memiliki referensi dari berbagai sumber, salah satunya paper yang berjudul IMPLEMENTASI KAMERA CMUCAM3 PADA ROBOT TANK YANG DAPAT MENGIKUTI OBJEK BERWARNA (Riky Tri Yunardi). Dari sumber di atas akan diambil sebuah penelitian yang berjudul “ RANCANG BANGUN SISTEM ROBOT HUMANOID SOCCER MENGGUNAKAN KAMERA CMUCAM ” Penelitian ini bertujuan untuk merancang dan mengimplementasikan Robot *Humanoid soccer* untuk menghasilkan gerakan seperti pemain bola, dengan cara pengendalian motor

servo dan rangkaian robot yang akurat sehingga keseimbangannya terjaga. Gerakan yang dimaksud meliputi robot berjalan, menendang bola, robot bergeser, dan robot bangun dari jatuh . Kamera *CMUCAM4* berfungsi sebagai pendeteksi *object* bola *soccer* oleh robot *humanoid* dengan metode *image processing*.

1.2 Rumusan Masalah

Adapun perumusan masalah dari perancangan robot *humanoid soccer* adalah sebagai berikut :

1. Bagaimana mengimplementasikan kamera *CMUCAM4* pada pengendalian robot *soccer*.
2. Bagaimana mengatur gerakan motor servo pada pergerakan robot *humanoid soccer* dan berpindah posisi.
3. Bagaimana merancang dan membuat sistem elektronika yang menunjang penggunaan kamera *CMUCAM4* sebagai pendeteksi bola *soccer*.
4. Bagaimana merancang dan membuat sistem elektronika yang menunjang penggunaan kamera *CMUcam4* sebagai pengontrol robot.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini antara lain :

1. Gerak robot adalah gerak maju, bangun dari jatuh, bergeser, duduk dan menendang.
2. Pandangan kamera menghadap depan
3. Komunikasi yang digunakan antara *CMUCAM4* dengan *Robot humanoid soccer* menggunakan Arduio uno
4. Jumlah servo yang dipakai total 16 buah

1.4 Tujuan

Dalam mengerjakan penelitian Tugas Akhir ini, penulis mempunyai beberapa tujuan yakni:

1. Merancang sistem pengontrol Robot *Humanoid soccer* menggunakan kamera *CMUCAM4*.
2. Menganalisa sistem pengontrol Robot *Humanoid soccer* menggunakan kamera *CMUCAM4*
3. Memaksimalkan kinerja robot sebagai robot *soccer*.

1.5 Manfaat

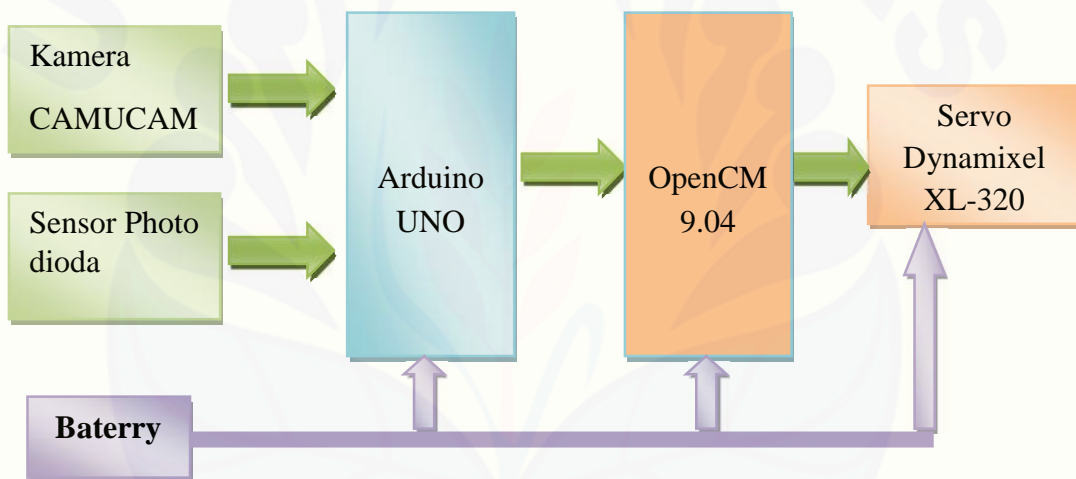
Manfaat alat ini yang diharapkan dari penelitian adalah sebagai berikut:

1. Dapat memberikan referensi tambahan dalam ilmu robotika pada pembaca atau pembuatan robot lainnya .
2. Dapat memberikan inspirasi bagi para pembuat robot lainnya sehingga dunia robot dan elektronika dapat lebih maju

BAB II. TINJAUAN PUSTAKA

2.1 Sistem Aktifasi dan kontrol robot

Sistem aktifasi dan kontrol robot terdiri dari dua bagian *software* dan *hardware*. Perangkat lunak berisi program untuk mengatur mikrokontroler dalam proses pengiriman data, verifikasi *image processing* dengan bentuk bola *soccer*, verifikasi pengendalian sistem gerak robot. *Software* yang digunakan untuk pemrograman *hardware* adalah Arduino. *Hardware* terdiri dari krontruksi robot *humanoid soccer*, sistem mekanis penggerak robot dengan motor servo, Arduino mikrokontroler dan OpenCM 9.04 mikrokontroler.



Gambar 2.1 Diagram Blok Sistem

Dalam proses pengenalan bentuk warna bola robot melakukan proses *image processing* dengan metode pengolahan citra bentuk warna bola *soccer* yang akan di deteksi, Arduino berfungsi sebagai modul *CMUCAM4* yang akan memberikan data masukan ke mikrokontroler OpenCM 9.04 sebagai penggerak servo motor. Proses pergerakan robot akan dikendalikan oleh mikrokontroler OpenCM 9.04 yang merupakan mikrokontroler dari Servo motor dynamixel XL-320. Sensor photodiode berfungsi sebagai pembacaan robot saat keadaan jatuh sehingga robot dapat berdiri sendiri.

2.2 Pengolahan Citra

Pengolahan citra adalah suatu metode yang digunakan untuk mengolah gambar sehingga menghasilkan gambar yang sesuai dengan keinginan kita. Pengambilan gambar bisa dilakukan dengan menggunakan kamera atau alat lain yang bisa digunakan untuk mentransfer gambar misalnya *scanner* atau kamera digital. Sehingga citra yang dihasilkan sudah berbentuk sinyal digital dan mudah dikenali atau dibaca komputer.

Citra digital adalah citra kontinyu yang sudah didiskritkan baik koordinat spasial maupun kecerahannya. Citra digital dianggap matrik dengan ukuran $M \times N$ dimana baris dan kolom menunjukkan titik-titiknya. Citra berwarna menggunakan metode RGB. Adapun masing masing warna dalam tabel memiliki 3 buah kombinasi angka yaitu R, G, dan B yang menentukan proporsi warna merah, warna hijau dan warna biru dari warna tersebut. RGB masing-masing memiliki *range* antara 0 hingga 63 sehingga jumlah warna yang dapat kita pilih untuk mengisi warna pada sebuah *cell* di tabel ialah $63 \times 63 \times 63 = 16$ juta warna. Tetapi seluruh tabel hanya dapat diisi dengan 256 pilihan warna. Kita dapat mengubah intensitas warna dari sebuah warna pada tabel dengan cara menggunakan *interrupt*. (Gary Bradski and Adrian. 2008)

Seringkali citra yang dihasilkan atau didapatkan memiliki kualitas yang kurang bagus. Hal ini dapat dikarenakan gangguan teknis peralatan yang digunakan atau adanya *noise* sekitarnya. Dapat juga dikarenakan kurang bagusnya pencahayaan. Untuk ini dilakukan manipulasi data-data untuk memperbaiki kualitas citra.

2.2.1 Color Filtering

Color Filtering adalah metode yang berguna untuk menemukan sebuah warna yang terdapat pada sebuah gambar. Dengan ditemukannya warna yang kita cari pada sebuah gambar, kita dapat menentukan proses apa yang selanjutnya harus dilakukan.

Pada dasarnya pencarian ini menggunakan kombinasi dari komponen *Red*, *Green*, dan *Blue* yang terdapat pada gambar. Nilai dari masing-masing komponen ini didapat dari hasil beberapa kali eksperimen. Hasil dari eksperimen ini adalah sebuah kombinasi dari komponen warna *Red*, *Green*, dan *Blue*. Kombinasi warna ini yang kemudian dijadikan *filter* yang merupakan penentu sebuah warna diloloskan atau tidak.

Keluaran dari metode ini langsung akan menghasilkan sebuah gambar biner. Gambar *biner* sendiri adalah sebuah gambar yang hanya memiliki 2 derajat keabuan yaitu hitam dan putih seperti pada Gambar 2.2. Dimana warna putih pada gambar adalah merupakan representasi dari sebuah pixel yang warnanya diloloskan. Sedangkan warna hitam pada gambar merupakan representasi dari pixel yang warnanya tidak diloloskan. Sehingga dari proses diatas, kita dapat menyimpulkan bahwa warna putih pada gambar adalah objek yang sedang kita cari. Karena warna putih pada gambar merupakan warna pada *pixel* yang sesuai dengan *filter* yang kita dapat dari hasil eksperimen



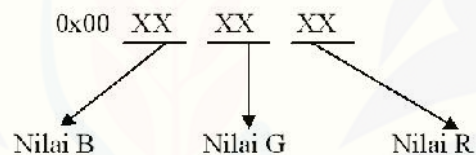
Gambar 2.2. Contoh Gambar Derajat Keabuan (Sumber: Usman, 2005)

2.2.2 Sistem Warna RGB

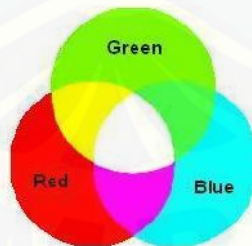
Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna diatas seperti Gambar 2.3 di bawah, variabel 0x00 menyatakan angka di belakangnya adalah hexadesimal.

Terlihat pada Gambar 2.3 bahwa setiap warna memiliki *range* nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 256), atau mempunyai nilai derajat keabuan $256 = 2^8$. Dengan demikian range warna yang digunakan adalah $(2^8)(2^8)(2^8) = 2^{24}$, 2 (atau yang dikenal dengan istilah *True Colour* pada *Windows*). Nilai warna yang digunakan diatas merupakan gabungan warna cahaya merah, warna hijau dan warna biru seperti yang terlihat pada gambar dibawah. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.

Dari definisi Gambar 2.4 di bawah untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB. Berikut merupakan contoh-contoh kombinasi warna yang bisa digunakan dapat dilihat pada Tabel 2.1.



Gambar 2.3. Nilai warna RGB dalam *Hexadesimal* (Sumber: Usman, 2005)



Gambar 2.4. Komposisi warna RGB (Sumber: Usman, 2005)

Tabel 2.1. Kombinasi warna dalam *Hexsadesimal*

Nilai	Warna	Nilai	Warna
0x00000000	Hitam	0x0000A0FF	Orange
0x000000FF	Merah	0x00888888	Abu-abu
0x0000FF00	Hijau	0x00FF00AA	Ungu
0x00FF0000	Biru	0x00A0FF00	Hijau Muda
0x0000FFFF	Kuning	0x00AA00FF	Merah Muda
0x00FF00FF	Magenta	0x00A0FFFF	Kuning Muda
0x00FFFFFF00	Cyan	0x000088AA	Coklat
0x00FFFFFFF	Putih	0x00AA0088	Ungu

2.2.3 Segmentasi Warna

Segmentasi merupakan proses untuk memisahkan objek yang kita ambil dengan latar belakang menggunakan teknik *Colour Filtering* pada warna-warna yang mendekati warna objek yang kita tentukan.

**Gambar 2.5.** Contoh segmentasi api pada gambar (Sumber: Usman, 2005)

2.2.4 Thresholding

Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses *thresholding* ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus:

$$x = w / b$$

dimana :

x adalah nilai derajat keabuan setelah *thresholding*

w adalah nilai derajat keabuan sebelum *thresholding*

b adalah jumlah derajat keabuan yang diinginkan

Berikut ini contoh *thresholding* untuk mengubah citra berwarna menjadi 2 warna.



Gambar 2.6. Hasil *Thresholding* warna merah (Sumber: Usman, 2005)

2.2.5 Erosi

Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan *structuring element* yang digunakan. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Cara yang dapat dilakukan ada 2:

- Dengan mengubah semua titik batas menjadi titik latar
- Dengan menset semua titik di sekeliling titik latar menjadi titik latar.

Algoritma untuk operasi *Erosi*:

Untuk semua titik dalam citra

Cek apakah tersebut titik latar

→ Jika ya maka ubah semua tetangganya menjadi titik latar

→ Jika tidak maka lanjutkan



Gambar 2.7. Proses Erosi (Sumber: Munir, 2004)

2.2.6 DILASI

Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan. Proses ini adalah kebalikan dari erosi, yaitu merubah latar sekeliling objek menjadi bagian dari objek tersebut.

Terdapat 2 cara untuk melakukan operasi ini, yaitu:

- Dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek, atau lebih mudahnya set setiap titik yang tetangganya adalah titik objek menjadi titik objek.
- Dengan mengubah semua titik di sekeliling titik batas menjadi titik objek, atau lebih mudahnya set semua titik tetangga sebuah titik objek menjadi titik objek.

Algoritma untuk operasi dilasi citra biner:

Untuk semua titik dalam citra
 Cek apakah tersebut titik obyek
 → Jika ya maka ubah semua tetangganya menjadi titik obyek
 → Jika tidak maka lanjutkan



Gambar 2.8. Proses Dilasi (Sumber: Munir, 2004)

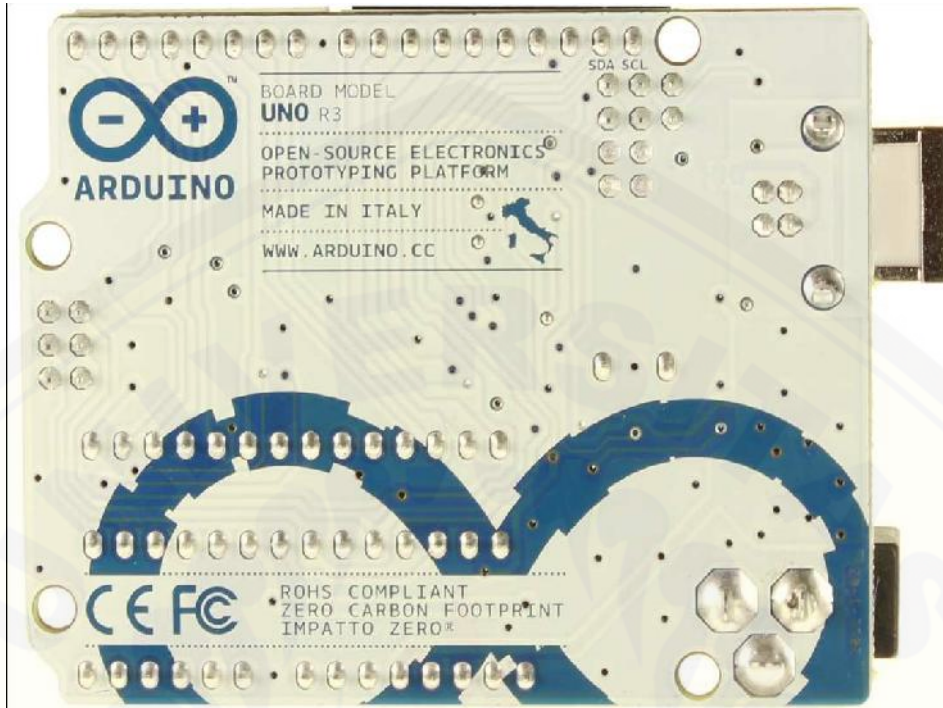
2.2.7 Canny Edge Detection

Metode *Canny* ini tidak memiliki operator khusus, namun metode ini terdiri dari beberapa langkah khusus. Metode *Canny* akan mendeteksi tepi dengan mencari nilai *gradien* maksimal lokal dari sebuah citra. *Gradien* tersebut dihitung menggunakan turunan dari *Gaussian filter*.

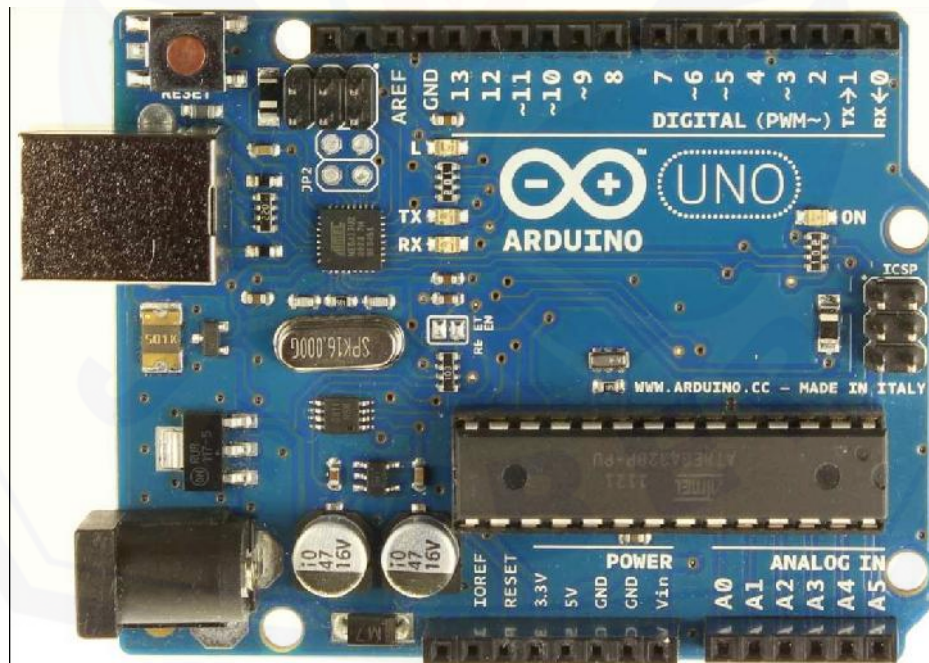
Metode *Canny* menggunakan dua *thresholds*, yang berguna untuk mendeteksi tepian yang terlihat jelas, dan tepian yang kurang jelas atau lemah, termasuk juga tepian yang kurang jelas yang terlihat pada output yang terhubung dengan tepian yang jelas. Metode ini lain dengan metode sebelumnya, yang sebelum dilakukan proses deteksi tepi harus dilakukan terlebih dahulu proses penghilangan noise. Metode *Canny* lebih utama akan mendeteksi tepian yang kurang jelas, yang tidak dapat diperoleh dengan menggunakan metode lain.

2.3 Arduino

Arduino adalah sebuah sistem minimum dimana menggunakan mikrokontroler ATmega 328 sebagai pengendalinya. *Arduino* memiliki 14 pin *input/output* yang terdiri dari 6 pin dapat digunakan sebagai *output* berupa PWM, 6 *input* yang bersifat *analog*, *crystal osilator* 16 Mhz, koneksi USB, *Jack power*, Kepala ICSP, dan tombol reset. *Arduino* mempunyai perangkat lunak tersendiri yang berfungsi sebagai *developer* program untuk memasukan program ke dalam mikrokontroler



Gambar 2.9 Arduino UNO R3 tampak belakang (sumber: Arduino Indonesia, 2013)



Gambar 2.10 Arduino UNO R3 tampak depan (sumber: Arduino Indonesia, 2013)

2.3.1 *Arduino Uno R3*

Arduino uno R3 mempunyai spesifikasi alat yang tidak jauh berbeda dengan *arduino* versi sebelumnya yaitu memiliki 14 pin *input/output* yang terdiri dari 6 pin dapat digunakan sebagai *output* berupa PWM, 6 *input* yang bersifat *analog*, *crystal* osilator 16 Mhz koneksi USB, *Jack power*, Kepala ICSP, dan tombol reset. Perbedaan *arduino uno R3* mengalami penambahan pin yaitu pin SDA dan SCL yang berada di dekat pin AREF. Dua pin penambahan ini terletak dekat tombol reset di sebelah pin AREF.

Berikut konfigurasi dari *Arduino Uno R3* :

- Mikrokontroler ATmega 328.
- Beroperasi pada tegangan 5V.
- Tegangan *input* (rekomendasi) 7-12V.
- Batas tegangan *input* 6-20V.
- *Pin digital input/output* 14 (6 menggunakan *output* PWM).
- *Pin analog input* 6.
- Arus *pin* per *input/output* 40mA.
- Arus untuk pin 3.3V adalah 50mA.
- *Flash memory* 32KB (ATmega 328) yang 2 KB digunakan oleh *bootloader*.
- Kecepatan clock 16 MHz.

2.3.1.1 *Power*

Arduino UNO R3 mempunyai 2 buah sumber untuk *power*. Dapat menggunakan koneksi USB atau menggunakan *power supply*. *Power* dapat bergantian secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan menyambungkan jack adaptor pada koneksi port *input supply*. *Board* *arduino* dapat dioperasikan menggunakan *power supply* sebesar 6-20 Volt. Jika *supply* kurang dari 7 volt, terkadang akan terjadi ketidak stabilan pada *arduino*. Ini di sebabkan oleh kurangnya pasokan daya

yang di berikan oleh pin 5 Volt yang kurang dari 5 volt. Jika menggunakan lebih dari 12 volt, regulator akan menjadi sangat panas dan menyebabkan kerusakan pada papan arduino. Direkomendasikan tegangan yang digunakan hanya pada kisaran 7 - 12 volt. *Power* pada arduino menggunakan 4 pin yaitu pin Vin, pin 5V, pin 3V3 dan pin *ground*.

- **Vin**

Tegangan *input* ke *arduino* ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan) pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan menggunakan *power jack* dapat menggunakan pin ini juga untuk mengaksesnya.

- **5V**

Regulasi *power supply* digunakan untuk power mikrokontroler dan komponen lain pada *board arduino*. Pin 5V ini dapat melalui pin Vin menggunakan regulator pada *board* atau *supply* oleh USB dan atau *supply* regulasi 5V lain.

- **3V3**

Supply 3.3 Volt didapat oleh FTDI chip yang ada di board. Arus maksimalnya adalah 50 mA

- **Ground**

Pin *ground* ini berfungsi sebagai jalur *ground* pada *arduino*

2.3.1.2 Memory

Memory pada *arduino* menggunakan *memory* yang terdapat pada mikrokontroler nya yaitu ATmega 328 yang memiliki 32 KB dengan 0.5 KB digunakan untuk *bootloader*. ATmega 328 juga memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

2.3.1.3 *Input dan Output*

Setiap 14 pin digital pada arduino dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Input* atau *output* dapat dioperasikan dengan tegangan 5 Volt. Setiap pin dapat menyediakan atau menerima maksimal 40 mA dan memiliki *internal pull-up* resistor (diputuskan secara standar) 20-50 KOhms. Beberapa pin memiliki fungsi sebagai berikut :

- Serial 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial Pin ini terhubung pada pin yang *responding* dari USB *FTDI* ke TTL chip serial.
- *Interrupt eksternal* 2 dan 3 pin ini dapat dikonfigurasi untuk *trigger* sebuah *interrupt* pada *low value*, *rising* atau *falling edge*, atau perubahan nilai.
- PWM : 3, 5, 6, 10, dan 11. Mendukung 8 bit *output* PWM dengan fungsi *analogWrite()*.
- *SPI* : 10 (*SS*), 11 (*MOSI*), 12 (*MISO*), 13 (*SCK*). Pin ini mensupport komunikasi *SPI*, yang mana masih mendukung *hardware*, yang tidak termaksud pada bahasa arduino.
- LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai *HIGH*, LED hidup, ketika pin *LOW*, LED mati.

2.3.1.4 *IComSat v1.1 -SIM900 GSM/GPRS shield*

IComSat merupakan suatu modul yang cocok dengan arduino, yaitu modul *SIM900 quad-band GSM/GPRS*. *IComSat* digunakan untuk pengiriman data yang menggunakan sistem SMS (*Short Message Service*). *Icomsat* dikontrol dengan menggunakan *ATcommands*.

2.3.1.4.1 Fitur *IComSat v1.1 -SIM900 GSM/GPRS shield*

IComSat v1.1 -SIM900 GSM/GPRS shield memiliki fitur sebagai berikut :

- Memiliki 4 tingkat frekuensi jaringan 850/900/1800/1900MHz.
- Paket data *GPRS* kelas 10/8.
- Di kontrol dengan *AT commands* (*GSM 07.07, 07.05 dan SIMCOMenhanced AT Commands*).
- *SMS (Short message service)*
- *Power ON/OFF* dan fungsi *reset* di dukung oleh arduino

2.3.1.4.2 Spesifikasi *IComSat v1.1 -SIM900 GSM/GPRS shield*

Spesifikasi *IComSat v1.1 -SIM900 GSM/GPRS shield* sebagai berikut :

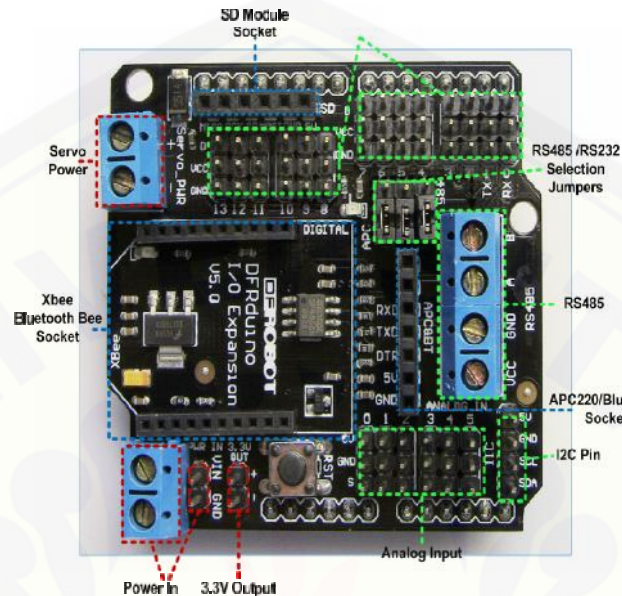
- Ukuran *board IComSat* memiliki ukuran *board* dengan 77.2mm X 66.0mm X 1.6mm.
- Indikator yang terdapat pada *IComSat* yaitu *PWR, status LED, net status LED*.
- *Power supply IComSat* dapat di jalankan dengan power supply 9-20 volt yang sesuai dengan arduino.
- Protokol komunikasi dalam *IComSat* menggunakan protokol *UART*.

2.3.2 *IO Expansion Shield For Arduino(V5)*

IO Expansion Shield For Arduino adalah perangkat tambahan yang digunakan untuk *interface* beberapa modul yang *compatible* dengan modul arduino. Modul- modul yang cocok dan sesuai dengan *platform* arduino yaitu:

- Mendukung modul *RS485*.
- Mendukung modul *Xbee (Xbee pro)*.
- Mendukung modul *Bluetooth*.

- Mendukung modul APC220.

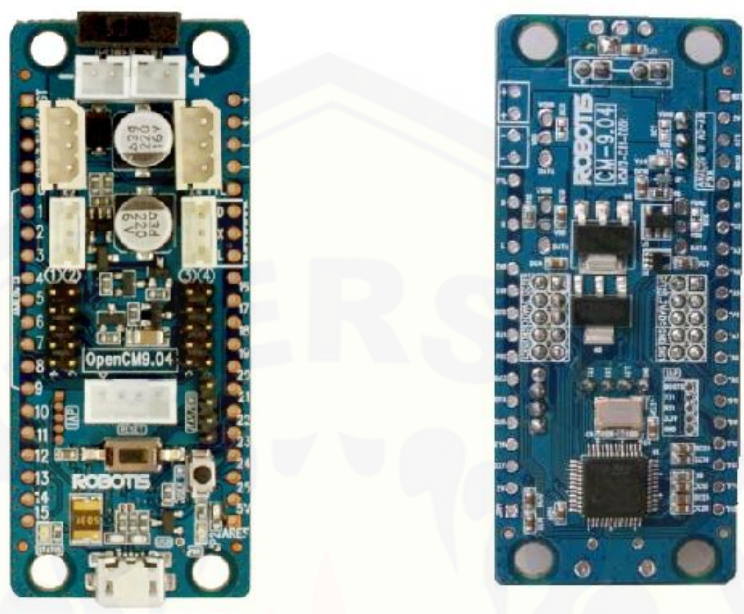


Gambar 2.11 IO Expansion Shield untuk Arduino (sumber: *Arduino Indonesia, 2013*)

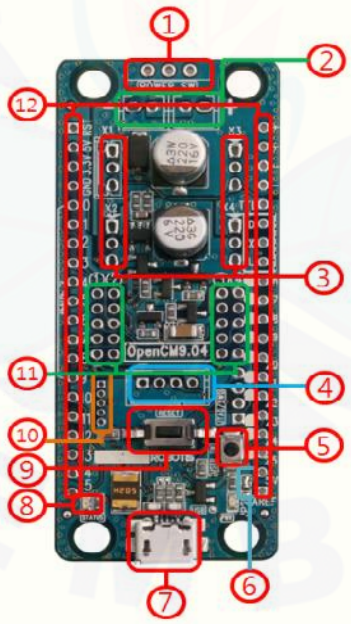
2.4 Mikrokontroler OpenCM 9.04

OpenCM 9,04-B *Microcontroller* menawarkan empat 3 pin TTL serial port untuk operasi Dynamixel AX, MX dan XL seri servo pintar. Banyak kekuatan pemrosesan disediakan oleh CPU ARM Cortex M3 STM32F103CD dengan 128 KB flash dan 20 KB dari SRAM. Board ini juga memiliki 26 GPIOs dengan input analog 10, USB, BISA, SPI, I2C dan serial port. Ada juga JTAG / port SWD untuk hardware debugging. Robotis telah merilis perangkat keras dan perangkat lunak desain sebagai open source.

Arduino seperti IDE yang tersedia untuk C / C ++ pemrograman dan Anda dapat men-*download* kode Anda dengan mudah melalui USB. Controller board ini adalah pilihan yang cocok untuk robot berukuran kecil dan menengah dan juga bisa digunakan sebagai Dynamixel pengendali cerdas pada robot besar.



Gambar 2.12 OpenCM 9.04 (sumber :www.robotsource.org)



Gambar 2.13 Pin out OpenCM 9.04 (sumber :www.robotsource.org)

1. Daya *Switch*: koneksi baterai (*input power*).
2. Baterai *socket* : *socket* untuk menghubungkan *Li-Ion Battery* (ies).
[Catatan: setiap baterai menyediakan 3.7V; baterai di kedua pin menyediakan 7.4V]
3. *Dynamixel* TTL BUS: port untuk berbasis TTL *Dynamixels*. *Dynamixels* terhubung dapat *daisy*-dirantai.
4. PIN USART: menghubungkan perangkat 4-pin seperti BT-110, BT-210, ZIG-110, LN-
5. 101 untuk komunikasi UART (catatan: *firmware* LN-101 hanya memungkinkan komunikasi dengan PC).
6. Pengguna *Switch*: user-programmable saklar; OpenCM9.04 pemulihan-mode
7. Analog Referensi Seleksi Jumper: perubahan 5V untuk sinyal analog.
8. Micro-B USB: Hubungkan OpenCM9.04 untuk komunikasi, download, dan 5V *input* daya pasokan. Setiap kabel micro USB tipe-B untuk ponsel Android berguna.
9. Status LED: Uji LED untuk pemrograman OpenCM9.04 ini. The LED berkedip dengan / sinyal rendah tinggi untuk pin D16. ulang saklar: ulang CPU.
10. JTAG / SWD 4 PIN: Via JTAG / terminal SWD menerapkan program lain seperti IAR, Keil. The OpenCM9.04 memiliki total 128Kbytes memori, download dan menyimpan awal biner bootloader ini di 0x08000000. (Bootloader: 0x08000000 ke ruang pemrograman Pengguna 0x08002FFF.: 0x08003000 untuk 0x08020000): extenal Sensor PIN: pin untuk sensor Robotis '.
11. 2.54 mm GPIO Header: perangkat eksternal Antarmuka ke OpenCM9.04 ini STM32F103CB CPU

Spesifikasi :

1. CPU : STM32F103CB (ARM Cortex-M3)
2. Tegangan : 5V ~ 16V (USB 5V, DXL 12V, XL-Series 7.4V)
3. I / O : GPIO 26
4. *Timer* : 8 (16bit)
5. Analog Dalam (ADC) : 10 (12bit)
6. Berkedip : 128 Kbytes
7. SRAM : 20 Kbytes
8. *Clock* : 72MHz (9 X 8 Mhz)
9. USB : 1 (2.0 FullSpeed) Micro tipe B
10. CAN : 1
11. USART : 3
12. SPI : 2
13. I2C : (TWI) 2
14. Debug : JTAG & SWD
15. Extenal SENSOR : 4
16. 3 Pin TTL : 4 (XL combo 3 PIN)
17. SW Alat : Robotis OpenCM
18. UKURAN : 27mm X 66,5 mm

2.5 CMUCAM4

Komputer visi serius teknologi keren namun butuh banyak daya proses, itu akan baik untuk memberikan proyek mikroprosesor berikutnya kekuatan untuk mengidentifikasi gerakan dan warna, tetapi sesuatu seperti Arduino akan berjuang untuk mengikutinya. Apa yang dapat Anda lakukan adalah memiliki prosesor khusus melakukan berat - angkat dan kemudian pipa yang data ke Arduino. *CMUCAM4* adalah sepenuhnya diprogram tertanam sensor visi komputer

.Prosesor utama adalah Parallax P8X32A (Propeller Chip) terhubung ke OmniVision 9665 CMOS modul sensor kamera.

CMUCAM4 dapat digunakan untuk melacak warna atau mengumpulkan statistik gambar dasar. Kinerja terbaik dapat dicapai bila ada warna yang sangat kontras dan intens . Sebagai contoh, dengan mudah dapat melacak bola merah di latar belakang putih, tapi itu akan sulit untuk membedakan antara nuansa yang berbeda coklat dalam mengubah cahaya. Pelacakan benda berwarna-warni dapat digunakan untuk melokalisasi daerah , mengikuti garis , atau mengejar bergerak beacon . Menggunakan statistik warna, adalah mungkin untuk *CMUCAM4* untuk memantau adegan , mendeteksi warna tertentu , atau melakukan deteksi gerak. *CMUCAM4* mendeteksi perubahan warna drastis , maka kemungkinan sesuatu dalam adegan berubah . Menggunakan "mode line", yang *CMUCAM4* dapat menghasilkan resolusi rendah gambar biner benda berwarna-warni . Hal ini dapat digunakan untuk melakukan pengolahan gambar yang lebih canggih yang mencakup lini berikut dengan deteksi cabang , atau bentuk pengakuan bahkan sederhana. Operasi-operasi yang lebih maju membutuhkan algoritma kustom untuk mengirim proses gambar biner yang dikirim dari *CMUCAM4* . Seperti halnya dengan kamera digital biasa , jenis ini mungkin memerlukan pengolahan komputer atau setidaknya mikrokontroler cepat.

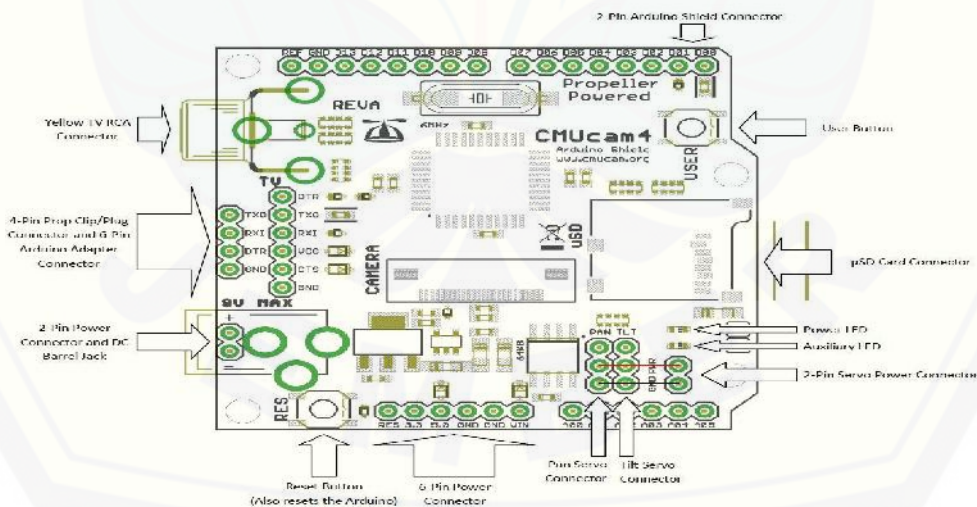
Konfigurasi yang paling umum untuk *CMUCAM4* adalah untuk memilikinya berkomunikasi dengan prosesor utama melalui port serial TTL standar. Ini " prosesor master" bisa menjadi komputer (melalui USB atau RS232) , Arduino , Basic Stamp, PIC, atau mikrokontroler serupa. *CMUCAM4* cukup kecil untuk menambah visi sederhana untuk embedded system yang tidak mampu ukuran atau kekuatan berbasis komputer sistem visi standar. Protokol komunikasi dan baud rate yang dirancang untuk mengakomodasi bahkan yang paling lambat dari prosesor . Untuk prosesor lebih lambat , yang *CMUCAM4* dapat beroperasi dalam mode " jajak pendapat " . Prosesor host dapat meminta *CMUCAM4* untuk hanya satu paket data. Hal ini memberikan prosesor lebih lambat kemampuan untuk lebih

mudah tetap disinkronkan dengan data. Hal ini juga memungkinkan untuk menambahkan penundaan antara individu karakter data serial menggunakan " mode delay " perintah . Karena keterlambatan komunikasi , kedua modus jajak pendapat dan mode delay akan menurunkan jumlah frame yang dapat diproses dalam satu detik. RCA dan *jack* barel tidak dihuni karena konektor dapat mempengaruhi bidang pandang saat pemasangan unit di sebuah kandang. Juga, bus listrik servo terputus dalam keadaan default, Anda akan perlu untuk menutup jumper ' Servo EN ' di sisi belakang papan untuk mengaktifkan daya ke servos Pan / Tilt. Untuk program Arduino dengan perisai *CMUCAM4* terhubung, Anda perlu baik menggunakan fitur ' HALT ', atau cabut jumper serial (SJ4 dan SJ5) , menghubungkan dua pin header (J1) ke pin Arduino digital, dan kemudian gunakan perpustakaan SoftwareSerial termasuk dalam instalasi Arduino .

FITUR :

- Sepenuhnya *open source* dan *re - programmable* menggunakan Perangkat Propeller
- Arduino Perisai Kompatibel / Pendukung Antarmuka Perpustakaan dan Demo Aplikasi untuk Arduino dan BASIC Stamp
- *Resolusi* VGA (640x480) RGB565/YUV655 sensor warna tingkat pengolahan gambar 30 *frame* per detik .Gambar baku pembuangan atas serial atau untuk kartu flash
 - (640:320:160:80) x (480:240:120:60) resolusi gambar
 - Ruang warna RGB565/YUV655
- *Image Processing Onboard* (QQVGA 160x120)
 - Jalur ditetapkan pengguna gumpalan warna dalam ruang warna RGB / YUV
 - Berarti , median , modus dan standar deviasi pengumpulan data - sampel dari resolusi 40x120

- *Segmented* (pengambangan) pengambilan gambar untuk melacak visualisasi (atas serial atau ke kartu flash)
 - resolusi gambar 80x60
 - ruang warna *Monochrome*
- Generasi *Histogram* (hingga 128 Bins) - sampel dari resolusi 40x120
- Gambar kliping Sewenang-wenang (windowing)
- USD / USDHC slot kartu *flash* dengan dukungan FAT16/32 sistem file penuh pengemudi o w / Direktori dan File manipulasi
- *I/O Interface*
 - Dua -port servo controller (pan dan tilt w / resolusi 1us pada refresh rate 50 Hz)
 - Pan dan / atau saluran servo Tilt dapat dikonfigurasi sebagai GPIOs Indikator pengguna dikontrol LED (merah) dan LED daya (hijau) TTL UART (sampai dengan 250.000 baud - 19.200 baud secara default)
- *Output video monokrom baseband analog* (NTSC / PAL) resolusi 160x120 untuk visualisasi pelacakan (tersegmentasi (pengambangan) *image* w / warna centroid dan melompat-lompat kotak overlay pada 30 FPS)

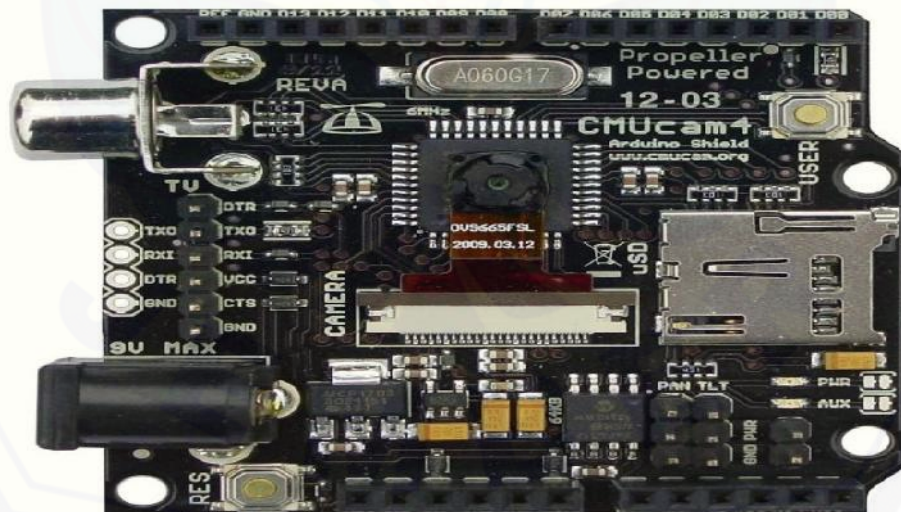


Gambar 2.14 Bentuk harware *CMUCAM4* (sumber :www.robotsource.org)

Sebanding dengan sensor video cerdas dapat mengenali warna, " *CMUCAM4* " akan memungkinkan (dalam hubungan dengan mikrokontroler) mobile robot menyenangkan mampu bereaksi dengan lingkungan mereka untuk mengikuti atau menghindari objek yang berbeda. Untuk melakukan hal ini akan cukup untuk mikrokontroler (PIC ® , AVR ® , ARM ™ , modul PICBASIC , CUBLOC , Basic Stamp ™ , Arduino ™ , dll ...) untuk mengirim (melalui port serial) urutan perintah yang sesuai ASCII tindakan yang akan Anda ingin lakukan . Sebagai contoh, perintah :

L1 - > menyalakan LED dari " *CMUCAM4* " L0 - > mematikan LED di " *CMUCAM4* "

Dengan prinsip yang sama , perintah yang lebih kompleks dapat dikirim untuk mengelola deteksi Warna yang mengindikasikan " *CMUCAM4* " batas luar komponen RGB dari warna yang ingin Anda cari . Oleh karena itu , setiap pixel gambar akan dianalisa dan dibandingkan dengan nilai- nilai Anda untuk mengembalikan koordinat piksel di kiri atas (x_1 , y_1) dan yang paling kanan bawah (x_2 , y_2) yang sesuai dengan permintaan Anda . Dengan mikrokontroler informasi Anda dapat menemukan objek berwarna dan bereaksi sesuai .



Gambar 2.15 *CMUCAM4* (sumber : www.robotsource.org)

2.6 Motor Servo Dynamixel XL-320

Servo berasal dari kata latin “*servus*” yang berarti budak. Servo adalah sebuah sistem kendali otomatis yang dipakai untuk mengatur kecepatan sebuah motor DC. Singkatnya, atur kecepatan dan tinggalkan, kecepatan motor akan konstan sampai kapanpun karena adanya servo ini. Kecepatan Motor servo di atur oleh besarnya frekuensi yang dikirimkan dari program melalui kabel data pada motor servo. Ada dua jenis motor servo yaitu motor servo yang berputar secara *kontinyu* dengan kecepatan tertentu dan motor servo yang bergerak menunjukkan sudut tertentu.

Bentuk motor servo dapat dilihat pada gambar 2.7. Terdapat tiga buah kabel dengan warna merah, hitam, dan kuning. Kabel merah dan hitam harus dihubungkan dengan sumber tegangan 4-6 volt DC agar motor servo dapat bekerja normal. Sedangkan kabel berwarna kuning adalah kabel data yang dipakai untuk mengatur arah gerak dan posisi servo.

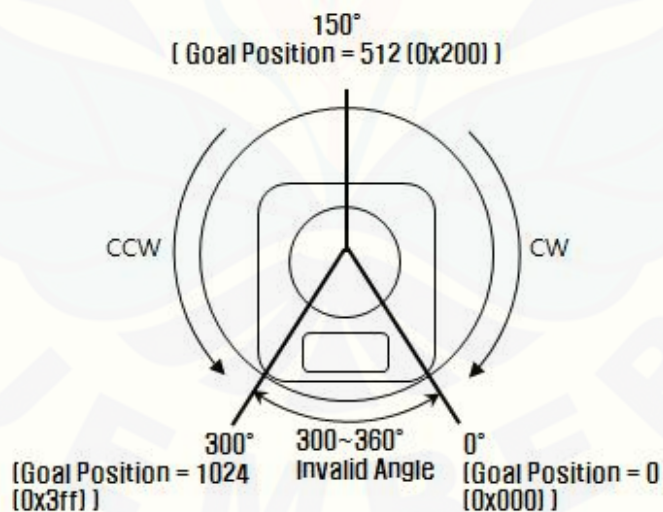


Gambar 2.16 Bentuk Fisik Motor Servo (Sumber: Mechatronicscrew, 2013)

Spesifikasi :

1. Berat : 16.7g
2. Dimensi : 24mm * 36mm * 27mm
3. Resolusi : 0.29 ° motor: Cored motor

4. Gigi Pengurangan Ratio : 238: 1
5. *Stall* Torsi: 0,39 Nm (di 7.4V) ada beban kecepatan : 114 rpm (di 7.4V)
Menjalankan Gelar § 0 ° ~ 300 ° § Aktifkan tak berujung
6. Menjalankan Suhu : -5 °C ~ + 70 °C
7. Tegangan : 6 ~ 8.4V (Direkomendasikan
Voltage 7.4V) Command Signal: Digital Packet
8. Protokol Jenis : *Half duplex Asynchronous*
Serial
9. Komunikasi (8bit, 1stop, ada Parity) Link (Fisik): TTL Tingkat multi drop (rantai jenis *Connector daisy*)
10. D : 253 ID (0 ~ 252)
11. Komunikasi Kecepatan : 7343bps ~ 1 Mbps
12. Tanggapan : Posisi, Suhu, beban, *Input Voltage*, dll Bahan: Mesin Plastik



Gambar 2.17 konfigurasi posisi sudut servo (Sumber: Mechatronicscrew, 2013)

- konfigurasi kecepatan.

0 ~ 2047 (0X7FF) dapat digunakan. Jika nilai yang di kemarahan 0 ~ 1023, itu berarti bahwa motor berputar ke arah CCW. Jika nilai yang di kemarahan 1024 ~ 2047, itu berarti bahwa motor berputar ke arah CW. Artinya, 10 bit menjadi sedikit arah untuk mengontrol arah, dan 0 dan 1024 adalah sama. Unit nilai ini bervariasi tergantung pada mode operasi.

- Joint Mode

Unit ini sekitar 0.111rpm. Sebagai contoh, jika sudah diatur untuk 300, itu berarti bahwa motor bergerak ke arah CCW pada tingkat sekitar 33.3rpm.

- Wheel Mode

Unit ini sekitar 0,1%. Sebagai contoh, jika sudah diatur untuk 512, itu berarti bahwa torsi dikendalikan oleh 50% dari torsi maksimum ke arah CCW.

2.7 Sensor Photodiode

Photodiode dibuat dari semikonduktor dengan bahan yang populer adalah silikon (Si) atau galium arsenida (GaAs), dan yang lain meliputi InSb, InAs, PbSe. Material ini menyerap cahaya dengan karakteristik panjang gelombang mencakup: 2500 Å – 11000 Å untuk silikon, 8000 Å – 20,000 Å untuk GaAs. Ketika sebuah photon (satu satuan energi dalam cahaya) dari sumber cahaya diserap, hal tersebut membangkitkan suatu elektron dan menghasilkan sepasang pembawa muatan tunggal, sebuah elektron dan sebuah hole, di mana suatu hole adalah bagian dari kisi-kisi semikonduktor yang kehilangan elektron. Arah Arus yang melalui sebuah semikonduktor adalah kebalikan dengan gerak muatan pembawa. cara tersebut didalam sebuah photodiode digunakan untuk mengumpulkan photon – menyebabkan pembawa muatan (seperti arus atau tegangan) mengalir/terbentuk di bagian-bagian elektroda.

Prinsip kerja photodiode :

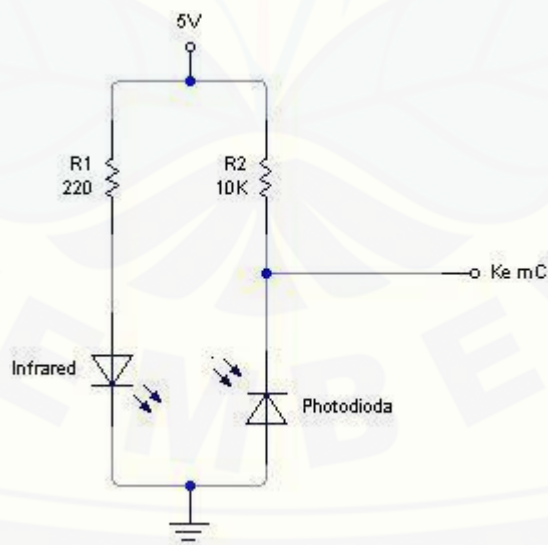
- Cahaya yang diserap oleh photodiode
- Terjadinya pergeseran foton
- Menghasilkan pasangan electron-hole di kedua sisi
- Electron menuju [+] sumber & hole menuju [-] sumber
- Sehingga arus akan mengalir di dalam rangkaian

Saat Photodiode terkena cahaya, maka akan bersifat sebagai sumber tegangan dan nilai resistansinya akan menjadi kecil.

Saat Photodiode tidak terkena cahaya, maka nilai resistansinya akan besar atau dapat diasumsikan tak hingga.

Ket : besarnya tegangan atau arus listrik yang dihasilkan oleh photodiode tergantung besar kecilnya radiasi yang dipancarkan oleh infrared

Photodiode digunakan sebagai penangkap gelombang cahaya yang dipancarkan oleh Infrared. Besarnya tegangan atau arus listrik yang dihasilkan oleh photodiode tergantung besar kecilnya radiasi yang dipancarkan oleh infrared



Gambar 2.18 Rangkaian sensor photodiode (sumber: irma 2013)

BAB III. METODOLOGI PENELITIAN

3.1 Lokasi Dan Tempat Penelitian

Adapun tempat dan waktu penelitian-analisis dilakukan secara umum di:

Tempat : Laboratorium Elektronika Terapan Fakultas Teknik Universitas
Jember
Alamat : Jl. Slamet Riyadi No. 62
Kecamatan Patrang Kabupaten Jember
Waktu : Maret 2015 - Juni 2015

3.2 Studi Literatur

Tahap ini merupakan tahap awal dari proses keseluruhan penelitian, pada tahap ini digunakan untuk mencari literatur dari penelitian sebelumnya, diharapkan dengan menemukan literatur dari penelitian sebelumnya dapat memberikan keyakinan bahwa penelitian ini dapat dilaksanakan dan memberikan arahan untuk mengurangi kesalahan dalam penelitian. Studi literatur pada Skripsi ini akan meliputi hal-hal sebagai berikut:

- Studi pengoperasian alat pendeteksi pengenalan bentuk bola *soccer* dalam pengaplikasian ke Robot *Humanoid soccer*.
- Studi penggabungan masukan data pengolahan citra

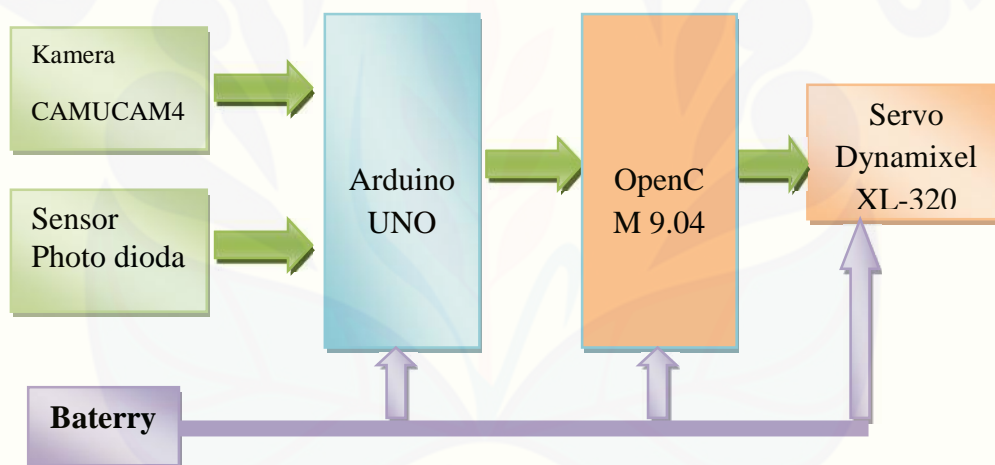
3.3 Pengumpulan dan Pengujian Data

Pengambilan data yang dimaksudkan disini adalah dengan melakukan pengujian terhadap kecocokan dalam pengenalan bentuk warna bola *soccer* setiap masukan dari beberapa bentuk bola. Variabel yang dilakukan penelitian meliputi dua hal, yaitu pengidentifikasian pola bentuk bola *soccer*. Identifikasi pola bola *soccer* dapat dilakukan dengan *hardware* berupa alat *CAMUCAM4* dengan rancangan tempat agar mudah diproses dalam pengolahan citra, kemudian data

masukan dari *CAMUCAM4* diolah dengan *software* buatan menggunakan sistem perhitungan pengolahan citra agar data yang masuk mikrokontroler dapat terbaca.

3.4 Perancangan Sistem

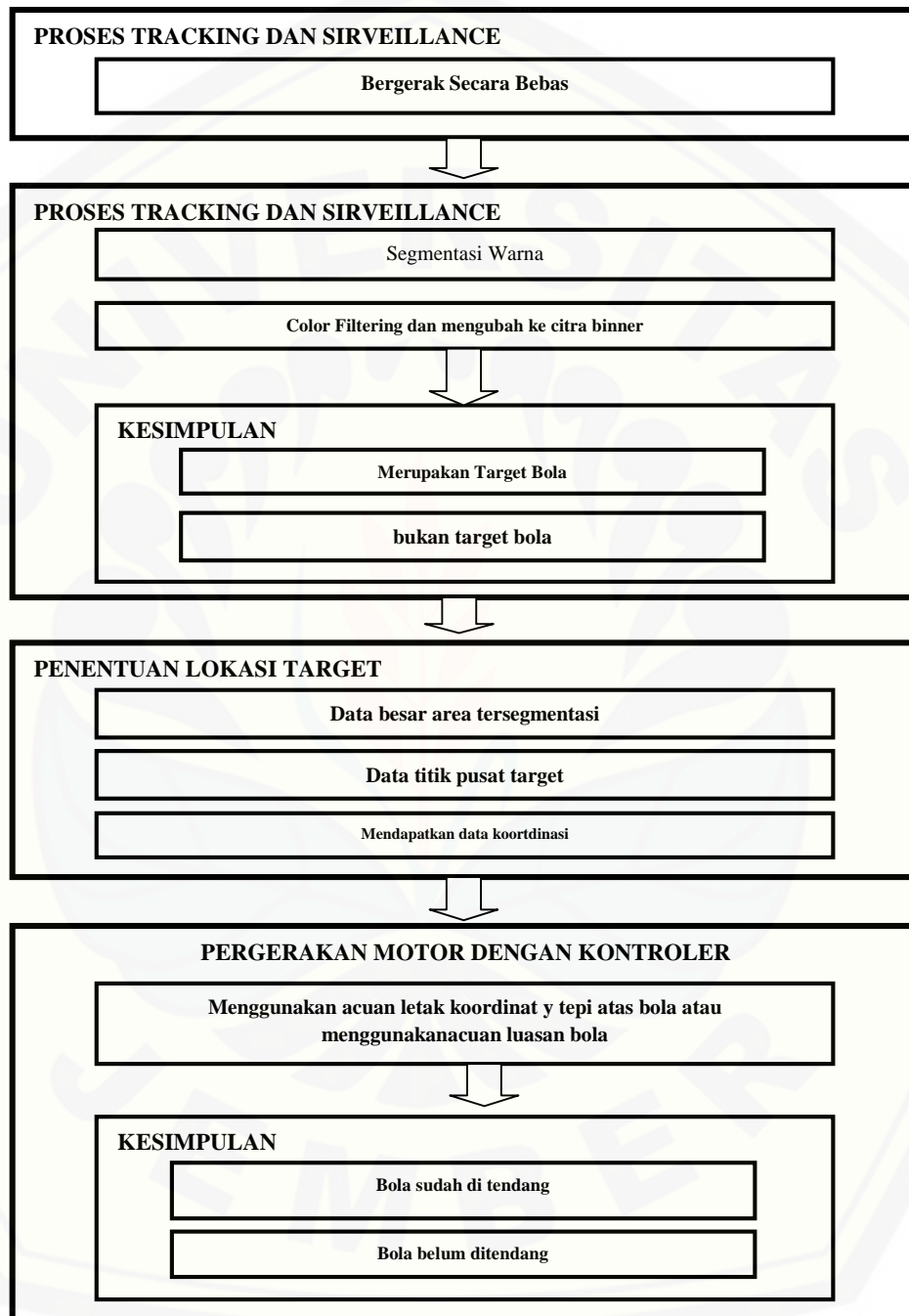
Sistem aktifasi dan kontrol robot terdiri dari dua bagian *software* dan *hardware*. Perangkat lunak berisi program untuk mengatur mikrokontroler dalam proses pengiriman data, verifikasi *image processing* dengan bentuk bola *soccer*, verifikasi pengendalian sistem gerak robot. *Software* yang digunakan untuk pemrograman *hardware* adalah Arduino. *Hardware* terdiri dari krontruksi robot *humanoid soccer*, sistem mekanis penggerak robot dengan motor servo, Arduino mikrokontroler dan OpenCM 9.04 mikrokontroler.



Gambar 3.1 Diagram Blok Sistem

Dalam proses pengenalan bentuk bola robot melakukan proses *image prosesing* dengan metode pengolahan citra bentuk bola *soccer* yang akan di deteksi, data kemudian berlanjut untuk proses pergerakan dan pengaktifan robot dan apabila sesuai data-data akan dikirim ke arduino. Arduino akan memberikan masukan data berupa digital I/O kepada mikrokontroler OpenCM 9.04, dimana OpenCM 9.04 akan memberikan perintah pada pergerakan robot melalui pengaturan posisi motor servo.

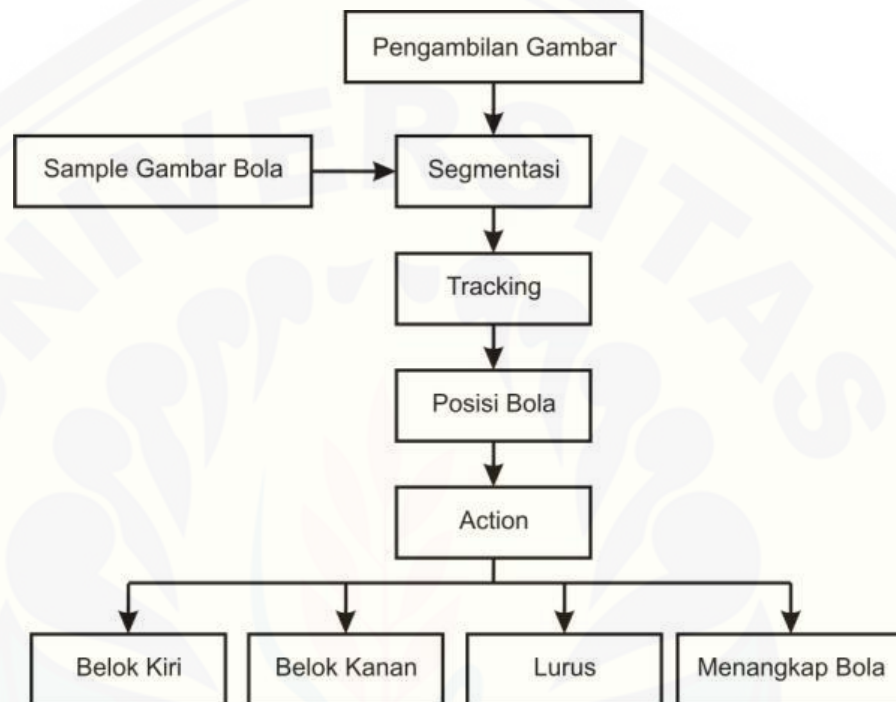
Secara garis besar sistem terbagi 2 proses, yaitu *image processing* dan robot kontrol.



Gambar 3.2 Diagram sistem *scanning* bola *soccer*

3.4.1 Sistem Pencarian

Dalam pembuatan program pencarian objek bola berwarna merah menggunakan blok diagram secara umum dapat dilihat pada gambar di bawah :



Gambar 3.3. Diagram Sistem kerja robot

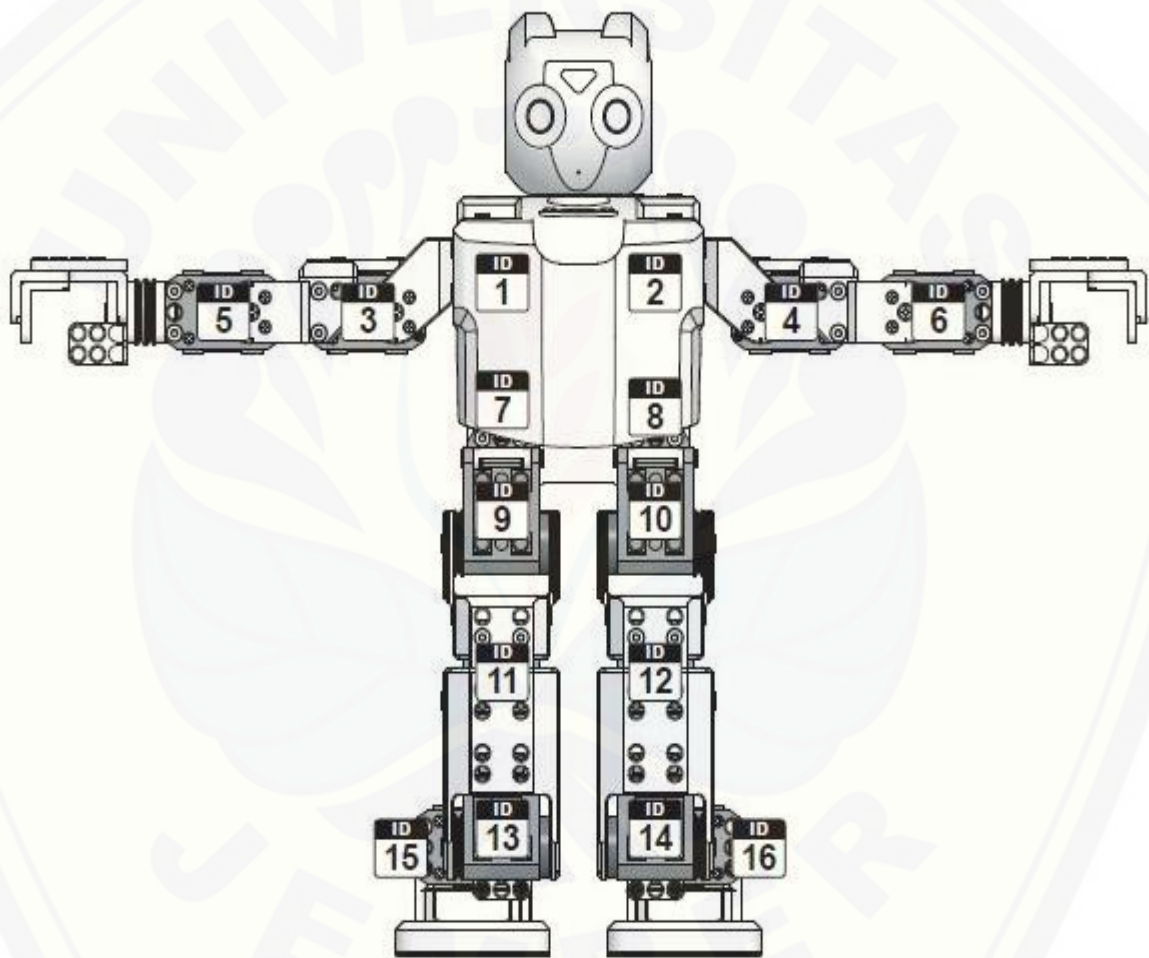
3.4.2 PERANGKAT KERAS

Secara garis besar perancangan perangkat keras adalah perancangan atau perakitan badan robot yang terdiri dari motor servo dynamixel XL-320 yang disambung dengan sedemikian rupa menggunakan rangka robot, sehingga akan membentuk badan robot.

Konstruksi robot sangat berpengaruh pada keseimbangan gerakannya, terutama pada titik keseimbangan berat badan robot ketika bergerak. Konstruksi robot terdiri dari sambungan antar rangka robot dan menggunakan motor servo dynamixel XL-320.

Pada proyek akhir ini robot *humanoid* terdiri dari 16 motor servo yang berfungsi sebagai sendi. 16 motor servo/sendai tersebut diantaranya 3 buah sendi pada tangan

kanan, 3 buah sendi pada tangan kiri, 5 buah sendi pada kaki kiri dan 5 buah sendi pada kaki kanan. Susunan motor servo dan rangka robot disambung sedemikian rupa, sehingga didapatkan konstruksi bentuk robot yang proporsional, seperti bentuk tubuh manusia, gambar desain badan robot *humanoid soccer* dapat dilihat pada gambar 3.4



Gambar 3.4. Desain Badan Robot *Humanoid Soccer* (sumber; www.support.robotis.com)

Berikut adalah perincian fungsi dari masing-masing motor servo:




- a. Motor servo 1, memberikan gerakan putar secara vertikal pada lengan kanan robot.
- b. Motor servo 2, memberikan gerakan putar secara vertikal pada lengan kiri robot.
- c. Motor servo 3, memberikan gerakan 2 arah pada lengan kanan robot di bagian atas. Jika dikombinasikan dengan gerakan motor servo 1, akan menghasilkan gerakan lengan kanan robot ke segala arah.
- d. Motor servo 4, memberikan gerakan 2 arah pada lengan kiri robot di bagian atas. Jika dikombinasikan dengan gerakan motor servo 2, akan menghasilkan gerakan lengan kiri robot ke segala arah.
- e. Motor servo 5, memberikan gerakan 2 arah pada lengan robot kanan bagian bawah. Motor ini berfungsi sebagai sendi siku lengan kanan robot.
- f. Motor servo 6, memberikan gerakan 2 arah pada lengan robot kiri bagian bawah. Motor ini berfungsi sebagai sendi siku lengan kiri robot.
- g. Motor servo 7, memberikan gerakan putar secara horisontal pada kaki kanan robot.
- h. Motor servo 8, memberikan gerakan putar secara horisontal pada kaki kiri robot.
- i. Motor servo 9, memberikan gerakan 2 arah secara vertikal pada kaki kanan robot bagian atas. Jika dikombinasikan dengan gerakan motor servo 7, akan menghasilkan gerakan kaki kanan robot ke segala arah.
- j. Motor servo 10, memberikan gerakan 2 arah secara vertikal pada kaki kiri robot bagian atas. Jika dikombinasikan dengan gerakan motor servo 8, akan menghasilkan gerakan kaki kiri robot ke segala arah.
- k. Motor servo 11, memberikan gerakan 2 arah pada kaki kanan robot bagian bawah. Motor ini berfungsi sebagai sendi lutut kaki kanan robot
- l. Motor servo 12, memberikan gerakan 2 arah pada kaki kiri robot bagian

- bawah. Motor ini berfungsi sebagai sendi lutut kaki kiri robot.
- m. Motor servo 13, memberikan gerakan 2 arah secara vertikal pada telapak kaki kanan robot.
 - n. Motor servo 14, memberikan gerakan 2 arah secara vertikal pada telapak kaki kiri robot.
 - o. Motor servo 15, memberikan gerakan 2 arah secara horisontal pada telapak kaki kanan robot.
 - p. Motor servo 16, memberikan gerakan 2 arah secara horisontal pada telapak kaki kiri robot.


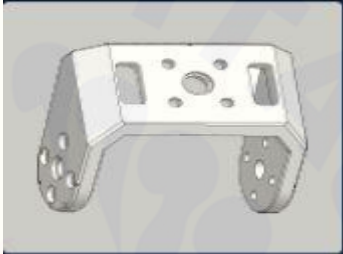


3.4.2.1 Komponen yang di gunakan pada perancangan robot *humanoid soccer*





Tabel 3.2 komponen perancangan robot

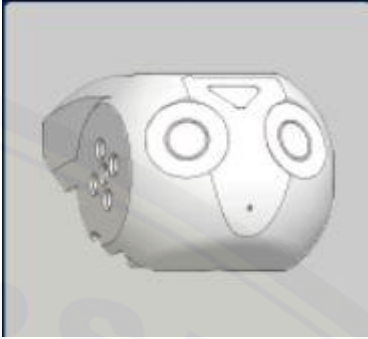


NO	NAMA	Jumlah	Gambar bentuk
1	Servo Dynamixel XL-320	16	 Gambar 3.5
2	Batery Li- ion LB-040	2	 Gambar 3.6

3	OpenCM 9.04 C	1	 <p data-bbox="984 632 1133 667">Gambar 3.7</p>
4	Batory Li- ion Box LB-040	2	 <p data-bbox="984 1058 1133 1094">Gambar 3.8</p>
5	DMF B04	4	 <p data-bbox="984 1503 1133 1539">Gambar 3.9</p>

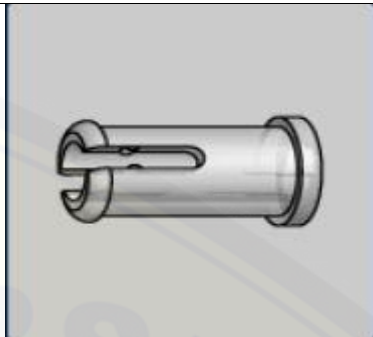
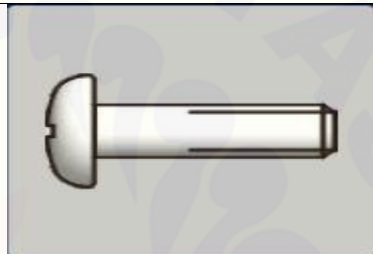

6	Dmf B05	2	 <p data-bbox="976 657 1143 688">Gambar 3.10</p>
7	SPL 2b2	14	 <p data-bbox="976 1108 1143 1140">Gambar 3.11</p>
8	SPD 3B6	6	 <p data-bbox="976 1549 1143 1581">Gambar 3.12</p>



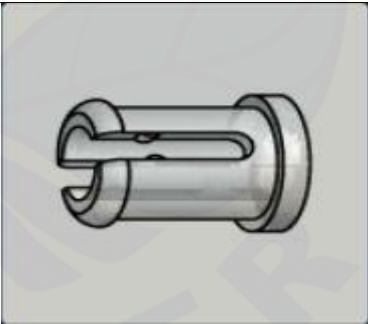
9	SPD 3B7	2	 <p>Gambar 3.13</p>
10	DMF B03	2	 <p>Gambar 3.14</p>
11	SPU 5	8	 <p>Gambar 3.15</p>
12	DMF B07	2	 <p>Gambar 3.16</p>

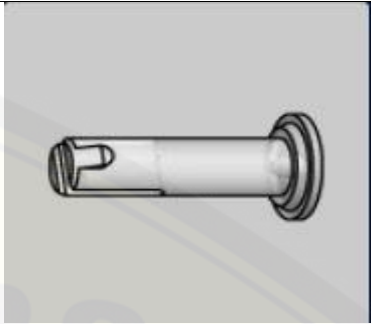
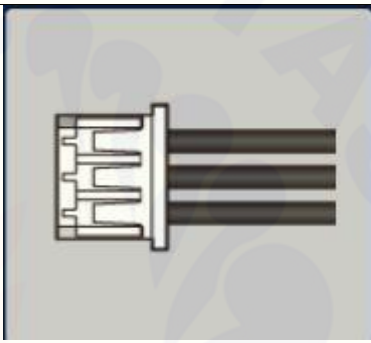
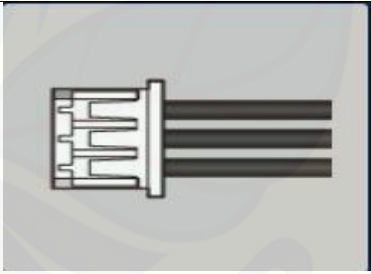
13	SPD 3B3	2	 <p data-bbox="966 598 1153 640">Gambar 3.17</p>
14	DMF B01	1	 <p data-bbox="966 997 1153 1039">Gambar 3.18</p>
15	DMF B06	2	 <p data-bbox="966 1396 1153 1438">Gambar 3.19</p>
16	SPD 4B5	2	 <p data-bbox="966 1774 1153 1816">Gambar 3.20</p>

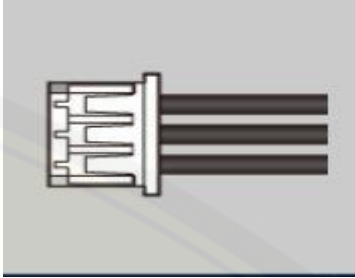
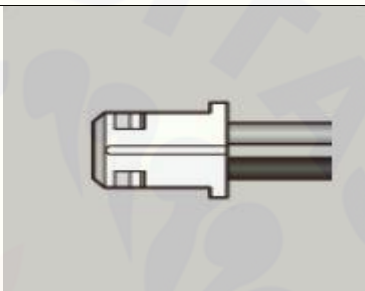
17	DMF F01	1	 <p data-bbox="976 684 1146 716">Gambar 3.21</p>
18	DMF B02	1	 <p data-bbox="976 1171 1146 1203">Gambar 3.22</p>
19	XL horn	17	 <p data-bbox="976 1633 1146 1665">Gambar 3.23</p>

20	SPD 7V4	4	 <p data-bbox="976 590 1146 621">Gambar 3.24</p>
21	SPO 5 gear	14	 <p data-bbox="976 1010 1146 1041">Gambar 3.25</p>
22	STR 20	2	 <p data-bbox="976 1434 1146 1465">Gambar 3.26</p>

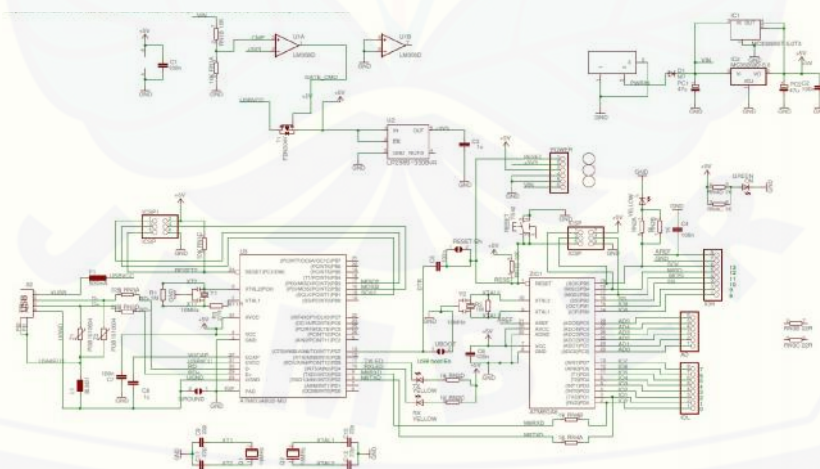
23	Skrup SRV 1LSH	40	 <p data-bbox="971 680 1146 716">Gambar 3.27</p>
24	CHS M2 0x04 TAP	72	 <p data-bbox="971 1058 1146 1094">Gambar 3.28</p>
25	CHS M2 0x06 TAP	4	 <p data-bbox="971 1491 1146 1526">Gambar 3.29</p>

26	Horn screw	17	 <p data-bbox="976 705 1146 737">Gambar 3.30</p>
27	SRV 1SP	166	 <p data-bbox="976 1150 1146 1182">Gambar 3.31</p>
28	SRV 1SH	166	 <p data-bbox="976 1598 1146 1629">Gambar 3.32</p>

29	SRV 1 LSP	40	 <p>Gambar 3.33</p>
30	DXL cable small 13 cm	6	 <p>Gambar 3.34</p>
31	DXL cable small 16 cm	2	 <p>Gambar 3.35</p>

32	DXL small 11 cm	8	 <p data-bbox="976 625 1146 659">Gambar 3.36</p>
33	Battery cable	2	 <p data-bbox="976 1035 1146 1068">Gambar 3.37</p>

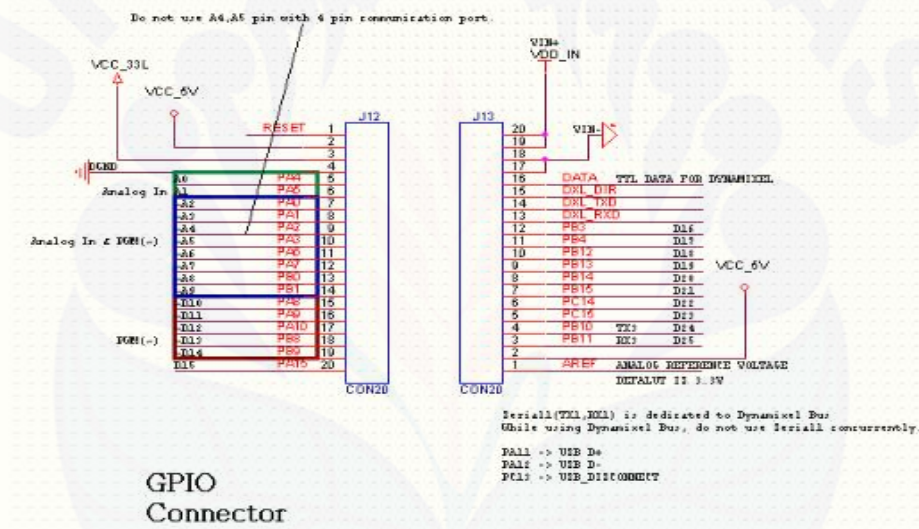
3.4.2.2 Rangkaian Arduino UNO



Gambar 3.38 Desain *scematic* arduino uno (sumber; Arduino indonesia, 2015)

Arduino adalah sebuah sistem minimum dimana menggunakan mikrokontroler ATmega 328 Arduino memiliki 14 pin input/output yang terdiri dari 6 pin dapat digunakan sebagai output berupa PWM, 6 input yang bersifat analog, *crystal* osilator 16 Mhz, koneksi USB, Jack power, Kepala ICSP, dan tombol reset. Arduino mempunyai perangkat lunak tersendiri yang berfungsi sebagai *developer* program untuk memasukan program ke dalam mikrokontroler. Data hasil pembacaan dari cmucam akan di proses oleh arduino.

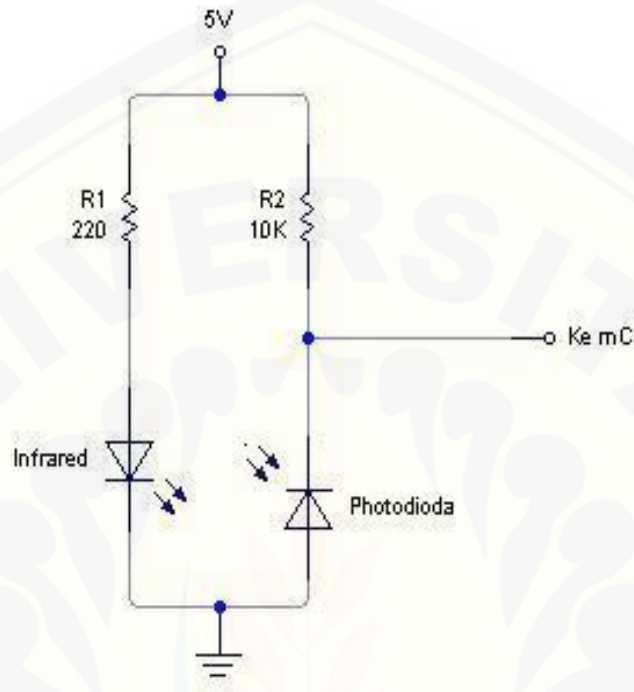
3.4.2.3 OpenCM 9.04



Gambar 3.39 Rangkaian OpenCM9.04 (sumber : www.robotsource.org)

Gambar 3.6 adalah Rangkaian pin konektor yang terhubung dengan port-port yang ada. Port opencm9.04 berfungsi sebagai penghubung antara servo dan penghubung arduino uno.

3.4.2.4 Rangkain photodioda



Gambar 3.40 Rangkaian sensor photodioda

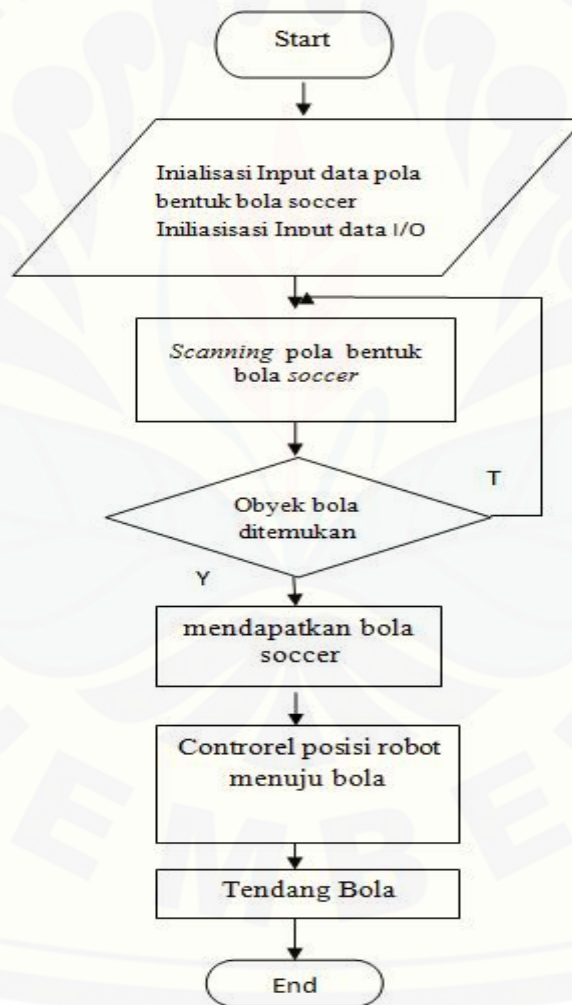
Pada gambar 3.7 Rangkaian Photodioda digunakan sebagai penangkap gelombang cahaya yang dipancarkan oleh Infrared. Besarnya tegangan atau arus listrik yang dihasilkan oleh photodioda tergantung besar kecilnya radiasi yang dipancarkan oleh infrared pada Saat *photodiode* terkena cahaya, maka akan bersifat sebagai sumber tegangan dan nilai resistansinya akan menjadi kecil. Saat *photodiode* tidak terkena cahaya, maka nilai resistansinya akan besar atau dapat diasumsikan tak hingga. Besarnya tegangan atau arus listrik yang dihasilkan oleh photodiode tergantung besar kecilnya radiasi yang dipancarkan oleh infrared atau Led.

3.4.3 PERANGKAT LUNAK

3.4.3.1 Proses citra bentuk bola *soccer*

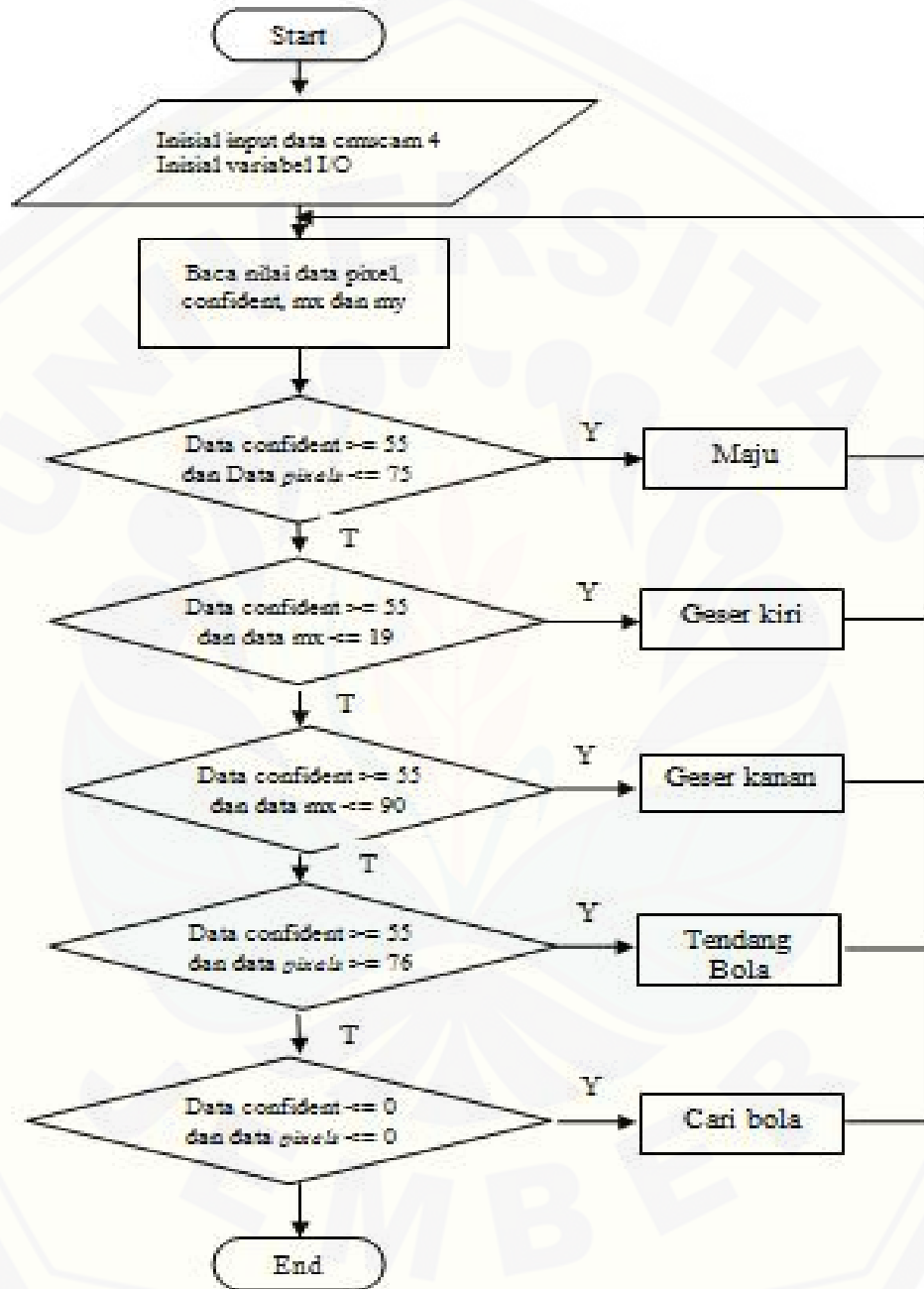
Setelah proses *scanning* bentuk pola bola *soccer* oleh kamera *CMUCAM4*. Mikrokontroler arduino akan mengolah data pola bola *soccer* tersebut ke bentuk data *pixels*, *mx* dan *my*. Pada saat nilai sudah terbaca maka arduino akan mengirim data ke OpenCM 9.04 yang kemudian mikrokontroler tersebut akan memberi perintah pada servo.

3.4.3.2 Flowchart robot *humanoid Soccer*



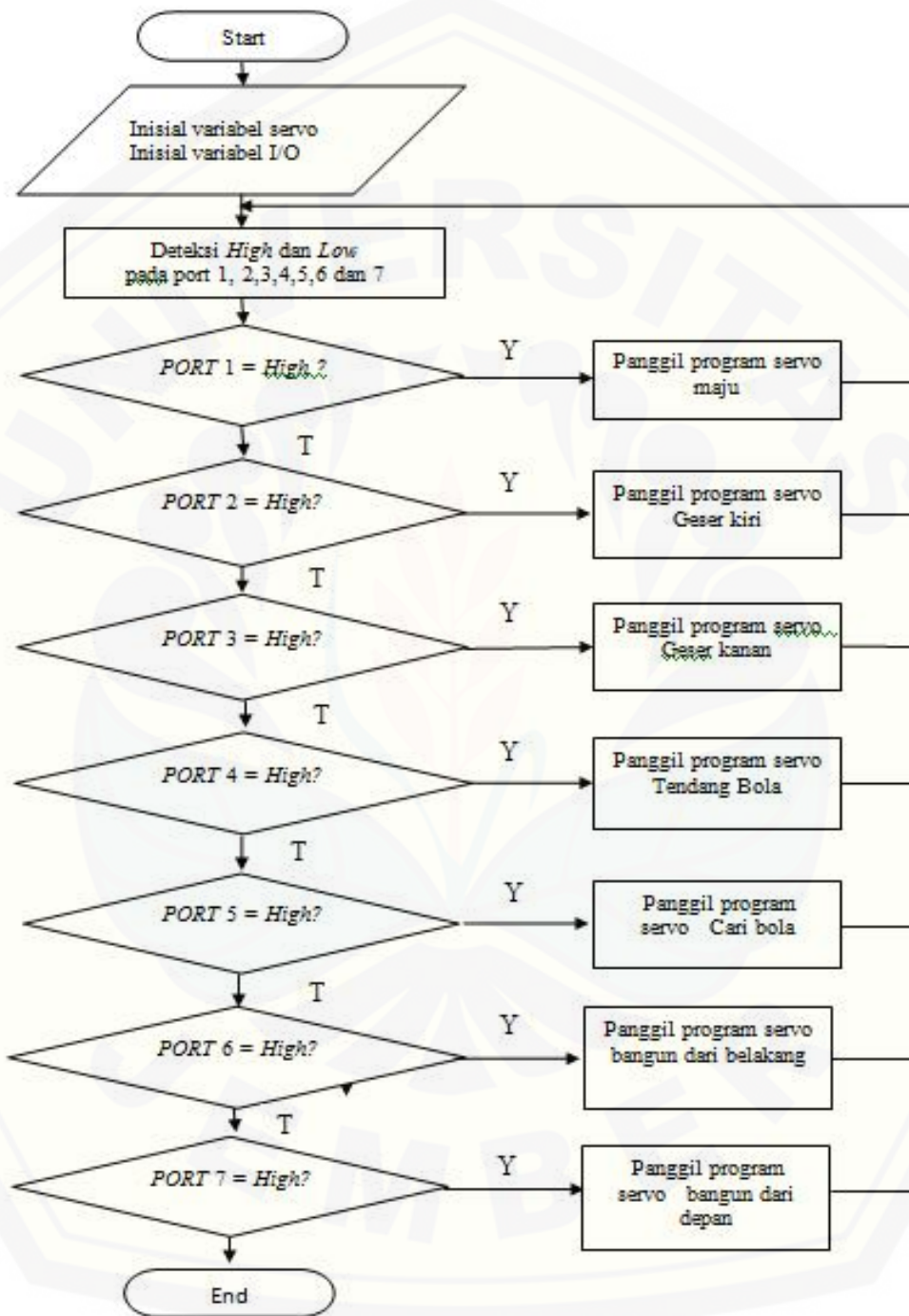
Gambar 3.41 Flowchart robot *humnoid soccer*

3.4.3.3 Flowchart scanning bola soccer



Gambar 3.42 Flowchart scanning bola soccer

3.4.3.4 Flowchart program kontrol servo robot



Gambar 3.43 Flowchart kontrol servo robot

➤ Penjelasan flowchart scanning bola *soccer*

Pada gambar *flowchart* 3.9 di jelaskan dimana pada awal *start* program melakukan pengenalan variabel masukan dan inialisasi. Setelah melakukan pengenalan atau inialisasi maka program berlanjut untuk mendeteksi nilai- nilai dan data dari masukan *CMUCAM4* yang berupa data *confident*, data *pixels* , data sumbu X dan data sumbu Y.

Data –data yang sudah terbaca kemudian akan di atur dan di kriteria sesuai dengan posisi robot yang diinginkan. Untuk posisi maju kriteria data yang di gunakan menggunakan data *pixels* ≤ 75 dengan acuan bila data tersebut terpenuhi maka program maju akan berjalan, demikian dengan data yang didapat oleh *CMUCAM4* sebagai pendeteksi objek bola *soccer* maka dapat di tentukan nilai - nilai batas sebagai acuan untuk memberikan perintah pada pergerakan robot *humanoid soccer*. Setelah semua program berjalan maka program melakukan *End* atau selesai

➤ Penjelasan *Flowchart* kontrol servo robot

Pada gambar *flowchart* 3.10 di jelaskan dimana pada awal *start* program melakukan pengenalan variabel masukan dan inialisasi. Setelah melakukan pengenalan atau inialisasi maka program berlanjut untuk mendeteksi nilai- nilai masukan dari mikrokontroler arduino ke mikrokontroler *OpenCM9.04*. Data yang digunakan merupakan data digital *High* dan *Low*.

Pada saat mikrokontroler arduino mengkodekan posisi salah satu port High maka *opencm9.04* akan merespond masukan tersebut dan kemudian mengkonfersikan dalam sebuah perintah untuk menjalankan robot ke posisi maju atau mundur. Dengan data yang di atur dan diinginkan maka robot akan bekerja sesuai data yang di baca oleh *OpenCM9.04* dan selanjutnya program akan *End*

BAB IV. HASIL DAN PEMBAHASAN

4.1 Pengujian dan Analisis

Dalam bab ini akan menjelaskan tentang pengujian alat dan sistem yang telah dirancang dan dibuat, selain itu juga akan mencantumkan data hasil pengujian serta pembahasannya. Pengujian mekanik Robot *humanoid soccer* akan meliputi posisi servo pada robot, sensor kamera cmucam4, sensor cahaya photodiode dan kinerja robot sebagai robot *soccer*.

4.1.1 Analisis posisi 16 motor servo pada robot soccer

Perancangan posisi servo pada robot *humanoid soccer* di hubungkan secara berurutan dengan label yang sudah di indifikasi tiap servo. Posisi servo pada robot *humanoid* akan memberikan sebuah pergerakan pada robot sehingga robot akan bergerak sesuai perintah yang diberikan oleh mikrokontroler OpenCm9.04

Tabel 4.1a Data sudut servo robot dalam kondisi jalan

No ID	Langkah pergerakan robot					
	Waktu 0 detik		Waktu 0.05 detik		Waktu 0.1 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	163°	556,37	163°	556,37	163°	556,37
2	137°	467,62	137°	467,62	137°	467,62
3	76,76°	262,007	76,76°	262,007	76,76°	262,007
4	223,24°	761,992	223,24°	761,992	223,24°	761,992
5	150°	512	150°	512	150°	512
6	150°	512	150°	512	150°	512
7	150°	512	150°	512	147,07°	501,998
8	150°	512	150°	512	170,51°	582,007
9	123,68°	422,161	117,77°	401,988	123,63°	421,9904
10	176,37°	602,009	170,51°	582,007	191,02°	652,014
11	179,3°	612,010	179,3°	612,010	179,3°	612,010
12	120,7°	411,989	120,7°	411,989	91,41°	312,012
13	163,18°	556,987	160,25°	546,986	166,11°	566,9888
14	136,82°	467,012	128,03°	437,009	119,24°	407,005
15	150°	512	152,93°	522,001	152,93°	522,001
16	150°	512	161,72°	552,004	158,79°	542,0032

Tabel 4.1b Data sudut servo robot dalam kondisi jalan

No ID	Langkah pergerakan robot					
	Waktu 0.2 detik		Waktu 0.3 detik		Waktu 0.4 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	163°	556,37	163°	556,37	163°	556,37
2	137°	467,62	137°	467,62	137°	467,62
3	76,76°	262,007	76,46°	260,983	76,46°	260,983
4	223,24°	761,992	222,95°	761,002	77,05°	262,997
5	150°	512	149,71°	511,010	149,71°	511,010
6	150°	512	149,71°	511,010	149,71°	511,010
7	150°	512	149,71°	511,010	146,78°	501,009
8	150°	512	149,71°	511,010	152,64°	521,011
9	129,49	441,992	129,2°	441,002	108,69°	370,995
10	182,23°	622,011	181,93°	620,987	176,07°	600,985
11	179,3°	612,010	179°	610,986	208,3°	710,997
12	120,7°	411,989	120,41°	410,999	120,41°	410,999
13	171,97°	586,990	171,68°	586,001	180,47°	616,004
14	139,75°	477,013	139,45°	475,989	133,59°	455,987
15	150°	512	137,99°	471,005	140,92°	481,006
16	150°	512	146,78°	501,009	146,78°	501,009

Tabel 4.1c Data sudut servo robot dalam kondisi jalan

No ID	Langkah pergerakan robot			
	Waktu 0.5 detik		Waktu 0.6 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	163°	556,37	163°	556,37
2	137°	467,62	137°	467,62
3	76,76°	262,007	76,76°	262,007
4	223,24°	761,992	223,24°	761,992
5	150°	512	150°	512
6	150°	512	150°	512
7	150°	512	150°	512
8	150°	512	150°	512
9	117,77°	401,988	117,77°	401,988
10	170,51°	582,007	170,51°	582,007
11	179,3°	612,010	179,3°	612,010
12	120,7°	411,989	120,7°	411,989
13	160,25°	546,986	160,25°	546,986
14	128,03°	437,009	128,07°	437,145
15	150°	512	152,93°	522,001
16	150°	512	161,72°	552,004

Tabel 4.2a Data sudut servo robot dalam bangun dari jatuh

No ID	Langkah pergerakan robot					
	Waktu 0.2 detik		Waktu 0.6 detik		Waktu 1.2 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	69,14°	235,997	93,46°	319,010	72,95°	249,002
2	230,86°	788,002	206,54°	704,989	227,05°	774,997
3	152,93°	522,001	71,78°	245,009	161,43°	551,014
4	147,07°	501,998	228,22°	778,990	161,43°	551,014
5	148,24°	505,992	149,12°	508,996	148,24°	505,992
6	148,24°	505,992	150,88°	515,003	151,76°	518,007
7	61,52°	209,988	61,52°	209,988	61,52°	209,988
8	238,48°	814,011	238,48°	814,011	238,48°	814,011
9	123,63°	421,990	123,63°	421,990	123,63°	421,990
10	176,37°	602,009	176,37°	602,009	176,37°	602,009
11	179,3°	612,010	179,3°	612,010	179,3°	612,010
12	120,7°	411,989	120,7°	411,989	120,7°	411,989
13	163,18°	556,987	163,18°	556,987	163,18°	556,987
14	136,82°	467,012	136,82°	467,012	136,82°	467,012
15	150°	512	150°	512	150°	512
16	150°	512	150°	512	72,95°	249,002

Tabel 4.2b Data sudut servo robot dalam bangun dari jatuh

No ID	Langkah pergerakan robot					
	Waktu 1.4 detik		Waktu 1.6 detik		Waktu 1.8 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	235,93°	805,307	150°	512	235,22°	802,884
2	64,07°	218,692	150°	512	64,78°	221,115
3	153,52°	524,014	76,76°	262,007	61,52°	209,988
4	146,48°	499,985	223,24°	761,992	238,48°	814,011
5	146,78°	501,009	150°	512	145,9°	498,005
6	153,22°	522,990	150°	512	154,1°	525,994
7	61,52°	209,988	150°	512	61,52°	209,988
8	238,48°	814,011	150°	512	238,48°	814,011
9	123,63°	421,990	123,63°	421,990	100,55°	343,210
10	176,37°	602,009	176,37°	602,009	199,45°	680,789
11	179,3°	612,010	179,3°	612,010	264,26°	902,007
12	120,7°	411,989	120,7°	411,989	35,74°	121,992
13	163,18°	556,987	163,18°	556,987	210,06°	717,004
14	136,82°	467,012	136,82°	467,012	89,94°	306,995
15	150°	512	150°	512	53,03°	181,009
16	150°	512	150°	512	246,97°	842,990

Tabel 4.2c Data sudut servo robot dalam bangun dari jatuh

No ID	Langkah pergerakan robot	
	Waktu 2.0 detik	
	Posisi sudut	Nilai ADC
1	235,15°	802,645
2	64,85°	221,354
3	62,4°	212,992
4	237,6°	811,008
5	150,88°	515,003
6	150,88°	515,003
7	150°	512
8	150°	512
9	85,55°	292,010
10	214,45°	731,989
11	264,26°	902,007
12	35,74°	121,992
13	210,06°	717,004
14	89,94°	306,995
15	150°	512
16	150°	512

Tabel 4.3a Data sudut servo robot dalam menendang

No ID	Langkah pergerakan robot					
	Waktu 1.4 detik		Waktu 1.6 detik		Waktu 1.8 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	150°	512	150°	512	116,6°	397,994
2	150°	512	150°	512	116,6°	397,994
3	76,46°	260,983	76,46°	260,983	98,44°	336,008
4	223,54°	763,016	223,54°	763,016	193,95°	662,016
5	149,71°	511,010	149,71°	511,010	139,75°	477,013
6	149,71°	511,010	149,71°	511,010	170,51°	582,007
7	149,71°	511,010	140,92°	481,006	143,85°	491,008
8	149,71°	511,010	149,71°	511,010	155,86°	532,002
9	123,63°	421,99	118,07°	403,012	73,83°	252,006
10	176,37°	602,002	176,37°	602,0096	182,23°	622,011
11	179°	610,986	245,21°	836,983	135,94°	464,008
12	120,41°	410,999	120,41°	410,999	120,41°	410,999
13	166,11°	566,988	166,11°	566,9888	149,71°	511,010
14	133,89°	457,0112	130,89°	446,7712	142,68°	487,0144
15	135,06°	461,0048	135,06°	461,0048	135,06°	461,0048
16	140,92°	481,006	135,06°	461,0048	135,06°	461,0048

Tabel 4.3b Data sudut servo robot dalam menendang

No ID	Langkah pergerakan robot			
	Waktu 1.4 detik		Waktu 1.6 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	149,71°	511,010	150°	512
2	149,71°	511,010	150°	512
3	76,46°	260,983	76,76°	262,007
4	222,95°	761,002	223,24°	761,992
5	149,71°	511,010	150°	512
6	149,71°	511,010	150°	512
7	143,85°	491,008	150°	512
8	155,86°	532,002	150°	512
9	113,09°	386,013	123,63°	421,990
10	176,37°	602,0096	176,37°	602,009
11	189,55°	646,997	179,3°	612,010
12	120,41°	410,999	120,7°	411,989
13	161,43°	551,014	163,18°	556,987
14	133,89°	457,011	136,82°	467,012
15	129,2°	441,002	150°	512
16	140,92°	481,006	150°	512

Tabel 4.4 Data sudut servo robot dalam posisi duduk

No ID	Langkah pergerakan robot					
	Waktu 0.1 detik		Waktu 0.15 detik		Waktu 0.3 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	150°	512	150°	512	150°	512
2	150°	512	150°	512	150°	512
3	223,24°	761,99	150°	512	223,24°	761,99
4	223,24°	761,99	223,24°	761,99	223,24°	761,99
5	150°	512	223,24°	761,99	150°	512
6	150°	512	150°	512	150°	512
7	150°	512	150°	512	150°	512
8	150°	512	150°	512	150°	512
9	88,45°	301,9	150°	512	85°	290
10	211,55°	722,09	86,08°	293,81	215,4°	734
11	259,24°	884,87	213,92°	730,18	267,19°	912
12	40,76°	139,12	264,65°	903,3	32,81°	111,99
13	208,49°	711,64	35,35°	120,6	212,99°	727
14	208,49°	711,64	211,55°	722	212,99°	727
15	150°	512	211,55°	722	150°	512
16	150°	512	150°	512	150°	512

Tabel 4.5 Data sudut servo robot dalam posisi bergeser

No ID	Langkah pergerakan robot					
	Waktu 0.1 detik		Waktu 0.15 detik		Waktu 0.3 detik	
	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC	Posisi sudut	Nilai ADC
1	150°	512	150,29°	512,98	150°	512
2	150°	512	149,71°	511,01	150°	512
3	76,76°	262,007	76,76°	262,007	76,76°	262,007
4	223,24°	761,992	223,24°	761,99	223,24°	761,992
5	150°	512	149,71°	511,01	150°	512
6	150°	512	149,71°	511,01	150°	512
7	150°	512	144,73°	494,01	150°	512
8	150°	512	155,27°	529,98	150°	512
9	123,68°	422,161	207,58°	708,5	123,68°	422,161
10	176,37°	602,009	185,16°	632,01	176,37°	602,009
11	179,3°	612,010	179°	610,98	179,3°	612,010
12	120,7°	411,989	120,41°	410,99	120,7°	411,989
13	163,18°	556,987	171,97°	586,99	163,18°	556,987
14	136,82°	467,012	145,61°	497,015	136,82°	467,012
15	150°	512	155,86°	532,002	150°	512
16	150°	512	155,57°	531,012	150°	512

Dari data tabel yang telah didapatkan dimana posisi ke 16 servo akan memberikan sebuah pergerakan pada robot. Servo Dynamixel Xl-320 sudah terintegrasi dengan mikrokontroler OpenCM9.04 dimana pada pengaturan perubahan sudut tiap servo bisa diatur. Nilai ADC didapatkan dengan cara mengkonversi nilai sudut ke nilai ADC. Dengan menggunakan perbandingan nilai

Dimana nilai Sudut $0^\circ - 300^\circ$ sama dengan $0 - 1024$ adc, dari perbandingan tersebut dapat di hitung nilai ADC bila didapatkan nilai sudut yang di inginkan. Sebagai contoh::

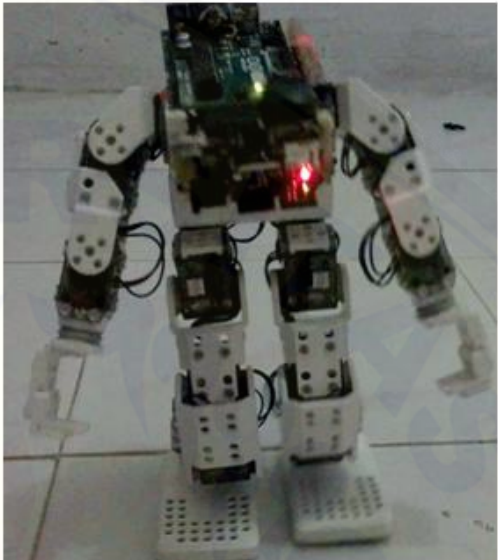
Diketahui Sudut $76,76^\circ$

Maka $ADC = (76,76 \times 1024) / 300$

$ADC = 262,007$


Tabel 4.6a Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0 detik		Gambar 4.1
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	123,68°	422,161	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	




Tabel 4.6b Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.05 detik		Gambar 4.2
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	117,77°	401,988	
10	170,51°	582,007	
11	179,3°	612,010	
12	120,7°	411,989	
13	160,25°	546,986	
14	128,03°	437,009	
15	152,93°	522,001	
16	161,72°	552,004	




Tabel 4.6c Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.1 detik		Gambar 4.3
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	147,07°	501,998	
8	170,51°	582,007	
9	123,63°	421,9904	
10	191,02°	652,014	
11	179,3°	612,010	
12	91,41°	312,012	
13	166,11°	566,9888	
14	119,24°	407,005	
15	152,93°	522,001	
16	158,79°	542,0032	




Tabel 4.6d Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.2 detik		Gambar 4.4
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	129,49	441,992	
10	182,23°	622,011	
11	179,3°	612,010	
12	120,7°	411,989	
13	171,97°	586,990	
14	139,75°	477,013	
15	150°	512	
16	150°	512	




Tabel 4.6e Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.3 detik		Gambar 4.5
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,46°	260,983	
4	222,95°	761,002	
5	149,71°	511,010	
6	149,71°	511,010	
7	149,71°	511,010	
8	149,71°	511,010	
9	129,2°	441,002	
10	181,93°	620,987	
11	179°	610,986	
12	120,41°	410,999	
13	171,68°	586,001	
14	139,45°	475,989	
15	137,99°	471,005	
16	146,78°	501,009	



Tabel 4.6f Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.4 detik		Gambar 4.6
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,46°	260,983	
4	77,05°	262,997	
5	149,71°	511,010	
6	149,71°	511,010	
7	146,78°	501,009	
8	152,64°	521,011	
9	108,69°	370,995	
10	176,07°	600,985	
11	208,3°	710,997	
12	120,41°	410,999	
13	180,47°	616,004	
14	133,59°	455,987	
15	140,92°	481,006	
16	146,78°	501,009	



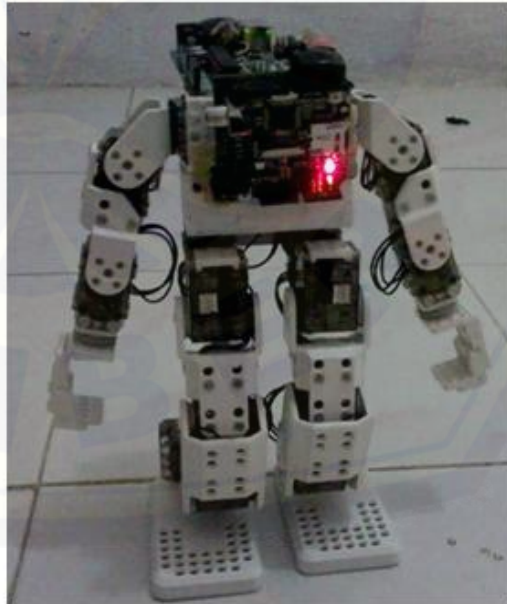
Tabel 4.6g Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.5 detik		Gambar 4.7
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	117,77°	401,988	
10	170,51°	582,007	
11	179,3°	612,010	
12	120,7°	411,989	
13	160,25°	546,986	
14	128,03°	437,009	
15	150°	512	
16	150°	512	



Tabel 4.6h Data sudut servo dan gambar robot dalam kondisi jalan

No ID	Posisi pada waktu 0.6 detik		Gambar 4.8
	Total Sudut	Total ADC	
1	163°	556,37	
2	137°	467,62	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	117,77°	401,988	
10	170,51°	582,007	
11	179,3°	612,010	
12	120,7°	411,989	
13	160,25°	546,986	
14	128,07°	437,145	
15	152,93°	522,001	
16	161,72°	552,004	



Tabel 4.7a Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 0.2 detik		Gambar 4.9
	Total Sudut	Total ADC	
1	69,14°	235,997	
2	230,86°	788,002	
3	152,93°	522,001	
4	147,07°	501,998	
5	148,24°	505,992	
6	148,24°	505,992	
7	61,52°	209,988	
8	238,48°	814,011	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	



Tabel 4.7b Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 0.6 detik		Gambar 4.10
	Total Sudut	Total ADC	
1	93,46°	319,010	
2	206,54°	704,989	
3	71,78°	245,009	
4	228,22°	778,990	
5	149,12°	508,996	
6	150,88°	515,003	
7	61,52°	209,988	
8	238,48°	814,011	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	



Tabel 4.7c Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 1.2 detik		Gambar 4.11
	Total Sudut	Total ADC	
1	72,95°	249,002	
2	227,05°	774,997	
3	161,43°	551,014	
4	161,43°	551,014	
5	148,24°	505,992	
6	151,76°	518,007	
7	61,52°	209,988	
8	238,48°	814,011	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	



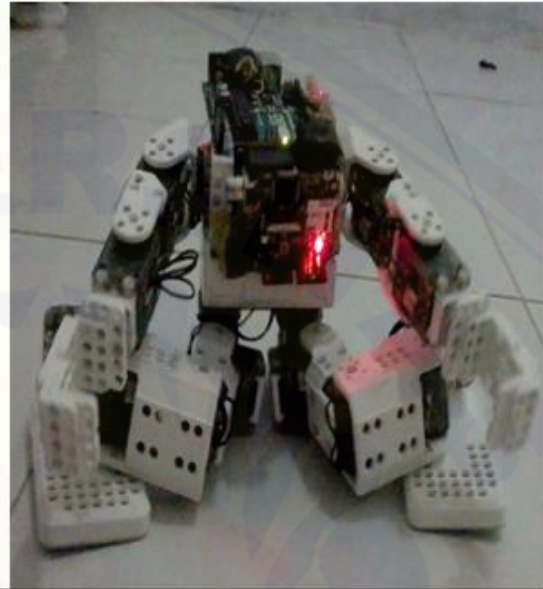
Tabel 4.7d Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 1.4 detik		Gambar 4.12
	Total Sudut	Total ADC	
1	235,93°	805,307	
2	64,07°	218,692	
3	153,52°	524,014	
4	146,48°	499,985	
5	146,78°	501,009	
6	153,22°	522,990	
7	61,52°	209,988	
8	238,48°	814,011	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	



Tabel 4.7e Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 1.6 detik		Gambar 4.13
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	




Tabel 4.7f Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 1.8 detik		Gambar 4.14
	Total Sudut	Total ADC	
1	235,22°	802,884	
2	64,78°	221,115	
3	61,52°	209,988	
4	238,48°	814,011	
5	145,9°	498,005	
6	154,1°	525,994	
7	61,52°	209,988	
8	238,48°	814,011	
9	100,55°	343,210	
10	199,45°	680,789	
11	264,26°	902,007	
12	35,74°	121,992	
13	210,06°	717,004	
14	89,94°	306,995	
15	53,03°	181,009	
16	246,97°	842,990	




Tabel 4.7g Data sudut servo dan gambar robot dalam bangun dari jatuh

No ID	Posisi pada waktu 1.9 detik		Gambar 4.15
	Total Sudut	Total ADC	
1	235,15°	802,645	
2	64,85°	221,354	
3	62,4°	212,992	
4	237,6°	811,008	
5	150,88°	515,003	
6	150,88°	515,003	
7	150°	512	
8	150°	512	
9	85,55°	292,010	
10	214,45°	731,989	
11	264,26°	902,007	
12	35,74°	121,992	
13	210,06°	717,004	
14	89,94°	306,995	
15	150°	512	
16	150°	512	

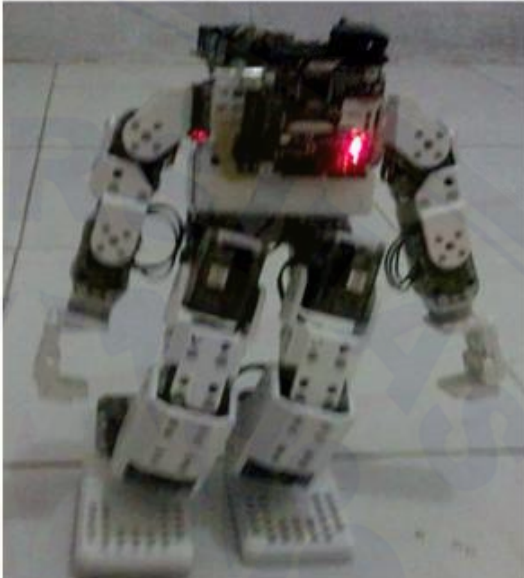


Tabel 4.8a Data sudut servo dan gambar robot dalam menendang

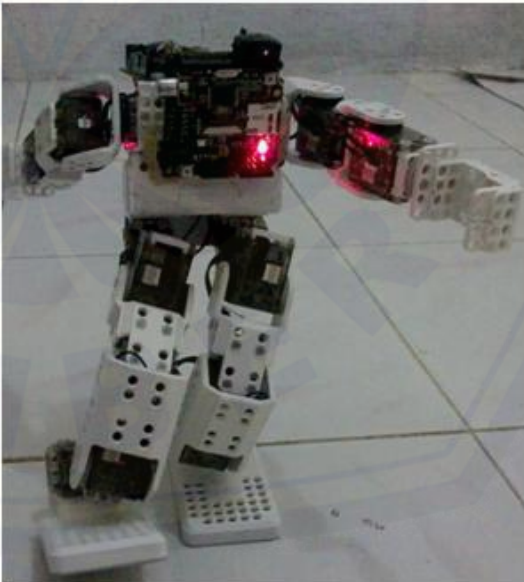
No ID	Posisi pada waktu 0.1 detik		Gambar 4.16
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	76,46°	260,983	
4	223,54°	763,016	
5	149,71°	511,010	
6	149,71°	511,010	
7	149,71°	511,010	
8	149,71°	511,010	
9	123,63°	421,99	
10	176,37°	602,002	
11	179°	610,986	
12	120,41°	410,999	
13	166,11°	566,988	
14	133,89°	457,0112	
15	135,06°	461,0048	
16	140,92°	481,006	



Tabel 4.8b Data sudut servo dan gambar robot dalam menendang

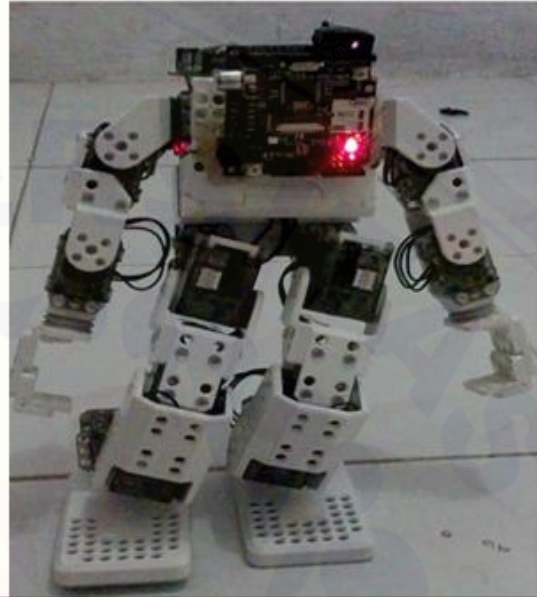
No ID	Posisi pada waktu 0.15 detik		Gambar 4.17
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	76,46°	260,983	
4	223,54°	763,016	
5	149,71°	511,010	
6	149,71°	511,010	
7	140,92°	481,006	
8	149,71°	511,010	
9	118,07°	403,012	
10	176,37°	602,0096	
11	245,21°	836,983	
12	120,41°	410,999	
13	166,11°	566,9888	
14	130,89°	446,7712	
15	135,06°	461,0048	
16	135,06°	461,0048	

Tabel 4.8c Data sudut servo dan gambar robot dalam menendang

No ID	Posisi pada waktu 0.3 detik		Gambar 4.18
	Total Sudut	Total ADC	
1	116,6°	397,994	
2	116,6°	397,994	
3	98,44°	336,008	
4	193,95°	662,016	
5	139,75°	477,013	
6	170,51°	582,007	
7	143,85°	491,008	
8	155,86°	532,002	
9	73,83°	252,006	
10	182,23°	622,011	
11	135,94°	464,008	
12	120,41°	410,999	
13	149,71°	511,010	
14	142,68°	487,0144	
15	135,06°	461,0048	
16	135,06°	461,0048	

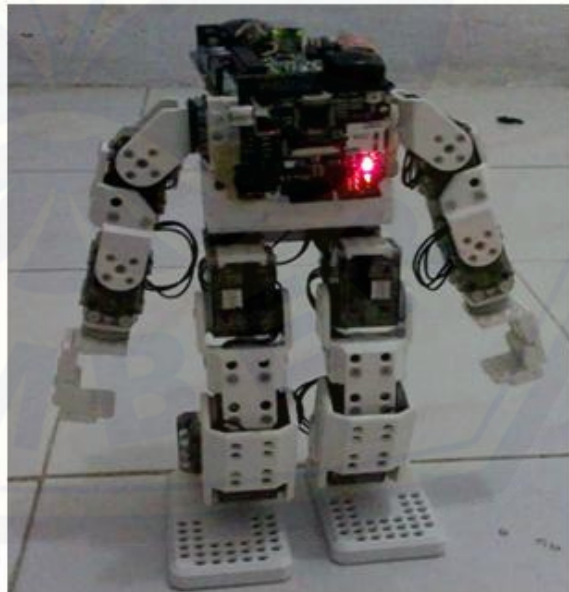
Tabel 4.8d Data sudut servo dan gambar robot dalam menendang

No ID	Posisi pada waktu 1 detik		Gambar 4.19
	Total Sudut	Total ADC	
1	149,71°	511,010	
2	149,71°	511,010	
3	76,46°	260,983	
4	222,95°	761,002	
5	149,71°	511,010	
6	149,71°	511,010	
7	143,85°	491,008	
8	155,86°	532,002	
9	113,09°	386,013	
10	176,37°	602,0096	
11	189,55°	646,997	
12	120,41°	410,999	
13	161,43°	551,014	
14	133,89°	457,011	
15	129,2°	441,002	
16	140,92°	481,006	




Tabel 4.8e Data sudut servo dan gambar robot dalam menendang

No ID	Posisi pada waktu 1.2 detik		Gambar 4.20
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	123,63°	421,990	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	




Tabel 4.9a Data sudut servo dan gambar robot dalam duduk

No ID	Posisi pada waktu 0.1 detik		Gambar 4.21
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	223,24°	761,99	
4	223,24°	761,99	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	88,45°	301,9	
10	211,55°	722,09	
11	259,24°	884,87	
12	40,76°	139,12	
13	208,49°	711,64	
14	208,49°	711,64	
15	150°	512	
16	150°	512	




Tabel 4.9b Data sudut servo dan gambar robot dalam duduk

No ID	Posisi pada waktu 0.15 detik		Gambar 4.22
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	223,24°	761,99	
4	223,24°	761,99	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	86,08°	293,81	
10	213,92°	730,18	
11	264,65°	903,3	
12	35,35°	120,6	
13	211,55°	722	
14	211,55°	722	
15	150°	512	
16	150°	512	



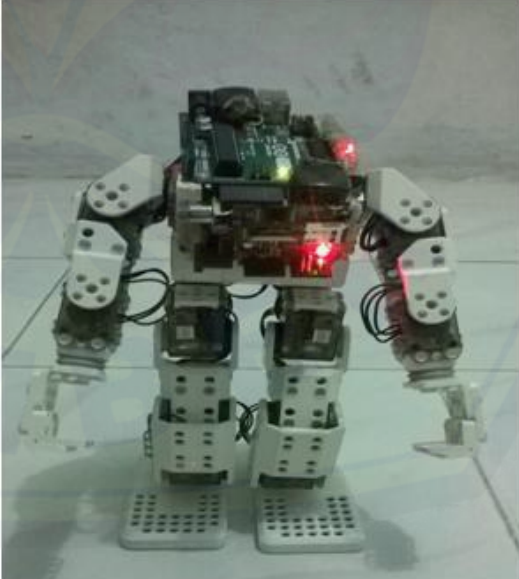
Tabel 4.9c Data sudut servo dan gambar robot dalam duduk

No ID	Posisi pada waktu 0.3 detik		Gambar 4.23
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	223,24°	761,99	
4	223,24°	761,99	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	85°	290	
10	215,4°	734	
11	267,19°	912	
12	32,81°	111,99	
13	212,99°	727	
14	212,99°	727	
15	150°	512	
16	150°	512	

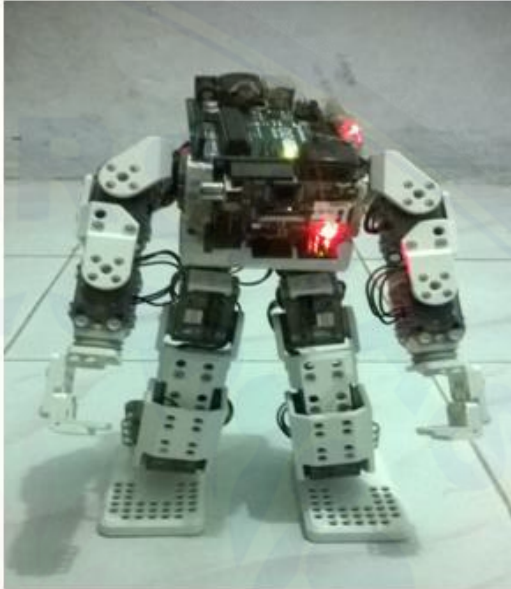


Tabel 4.10a Data sudut servo dan gambar robot dalam bergeser

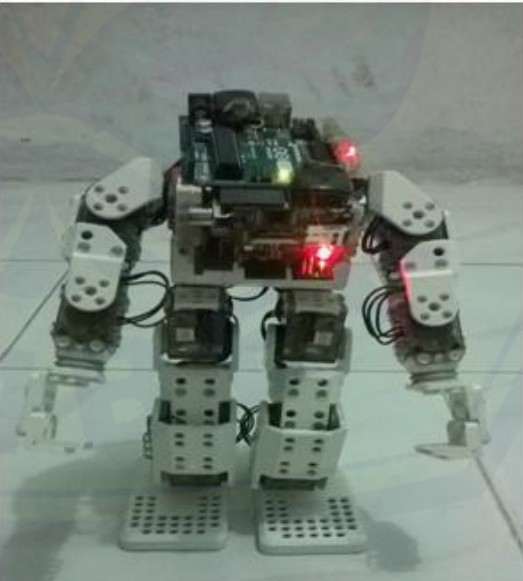
No ID	Posisi pada waktu 0.1 detik		Gambar 4.24
	Total Sudut	Total ADC	
1	150°	512	
2	150°	512	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	123,68°	422,161	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	



Tabel 4.10b Data sudut servo dan gambar robot dalam bergeser

No ID	Posisi pada waktu 0.15 detik		
	Total Sudut	Total ADC	Gambar 4.25
1	150,29°	512,98	
2	149,71°	511,01	
3	76,76°	262,007	
4	223,24°	761,99	
5	149,71°	511,01	
6	149,71°	511,01	
7	144,73°	494,01	
8	155,27°	529,98	
9	207,58°	708,5	
10	185,16°	632,01	
11	179°	610,98	
12	120,41°	410,99	
13	171,97°	586,99	
14	145,61°	497,015	
15	155,86°	532,002	
16	155,57°	531,012	

Tabel 4.10c Data sudut servo dan gambar robot dalam bergeser

No ID	Posisi pada waktu 0.3 detik		
	Total Sudut	Total ADC	Gambar 4.26
1	150°	512	
2	150°	512	
3	76,76°	262,007	
4	223,24°	761,992	
5	150°	512	
6	150°	512	
7	150°	512	
8	150°	512	
9	123,68°	422,161	
10	176,37°	602,009	
11	179,3°	612,010	
12	120,7°	411,989	
13	163,18°	556,987	
14	136,82°	467,012	
15	150°	512	
16	150°	512	

4.1.2 Pengujian Sensor kamera *CMUCAM4*

Pada pengujian kamera *CMUCAM4* diambil menggunakan serial komunikasi mikrokontroler Arduino dan ditampilkan menggunakan LCD sebagai penampil data.

Tabel 4.11 Data pixel cmucam 4 dengan jarak posisi bola

No	Jarak (cm)	<i>Pixels</i>	<i>Confident</i>
1	5	75	80
2	10	62	77
3	15	34	57
4	20	18	33
5	25	12	21
6	30	9	16
7	35	6	11
8	40	4	8
9	45	3	6

Pada data tabel 4.10 diperoleh nilai- nilai batas sebagai acuan untuk pergerakan robot *humanoid soccer* dimana data *pixels* dan *confident* digunakan untuk mendeteksi keberadaan *object* bola *soccer* sebagai target. Pada data di dapat pada jarak posisi bola 5 cm di peroleh data *pixels* 75 dan *confident* 80. Dan pada saat bola berposisi pada jarak 25 cm dari kamera di peroleh data *pixels* 12 dan *confident* 21. Nilai – nilai yang di peroleh selanjutnya di gunakan untuk mengetahui keberadaan bola seberapa jauh posisi bola yang terdeteksi. Semakin jauh bola terdeteksi maka angka nilai yang di peroleh semakin kecil dan semakin dekat posisi *object* bola maka *pixels* dan data *confident* semakin besar.

Tabel 4.12 Data sumbu X axis *CMUCAM4* dengan jarak posisi bola bergeser kekanan dan ke kiri

NO	JARAK (cm)	Mx tengah	Mx ke kiri 5 cm	Mx ke kanan 5 cm
1	10	55	16	130
2	15	64	19	121
3	20	72	31	121
4	25	70	39	104
5	30	70	43	104

Pada data tabel 4.11 di peroleh nilai- nilai batas sebagai acuan untuk pergerakan robot *humanoid soccer* dimana data mx digunakan untuk mendeteksi keberadaan object bola *soccer* di posisi kanan atau kiri. Pada data yang telah di peroleh dengan jarak 10 cm di peroleh data mx axis 55 dan bila bola di geser ke kiri sejauh 5 cm maka nilai mx akan semakin kecil denga nilai 16 apabila bola di geser ke kanan sejauh 5 cm maka nilai akan berubah semakin besar 130. Data – data pada tabel akan memberikan nilai – nilai batas sebagai acuan atau batas perintah apakah robot aharus bergeser/berbelok ke kanan atau kekiri.

Tabel 4.13 Data sumbu Y axis *cmucam 4* dengan jarak posisi bola berubah ke atas dan kebawah

NO	JARAK (cm)	My tengah	My ke atas 5 cm	My ke bawah 5 cm
1	25	65	-	-
2	25	-	-	38
3	25	-	-	14
4	25	-	92	-
5	25	-	116	-

Pada data tabel 4.12 di peroleh nilai- nilai batas sebagai acuan untuk pergerakan robot *humanoid soccer*. Data my digunakan untuk mendeteksi objek bola *soccer* dengan posisi dibawah atau diatas kamera *CMUCAM4*. Dengan jarak stabil 25 cm di peroleh nilai tengah my 65 apabila bola di geser keatas sejauh 5 cm maka nilai my semakin besar dengan nilai 92 bila bola di geser ke bawah di dapatkan nilai semakin kecil 38. Nilai – nilai yang telah di dapat kan dapat digunakan untuk posisi robot *humanoid soccer* untuk duduk atau berdiri.



Gambar 4.27 Pengujian *CMUCAM4*

4.1.3 Pengujian Sensor photodiode

Photodiode berfungsi sebagai sensor pembaca saat robot jatuh ke depan maupun jatuh ke belakang. Pengujian sensor elektroda dilakukan cara membaca nilai tegangan yang masuk ke arduino. Saat photodiode terkena cahaya, maka akan bersifat sebagai sumber tegangan dan nilai resistansinya akan menjadi kecil. Saat photodiode tidak terkena cahaya, maka nilai resistansinya akan besar atau dapat diasumsikan tak hingga.

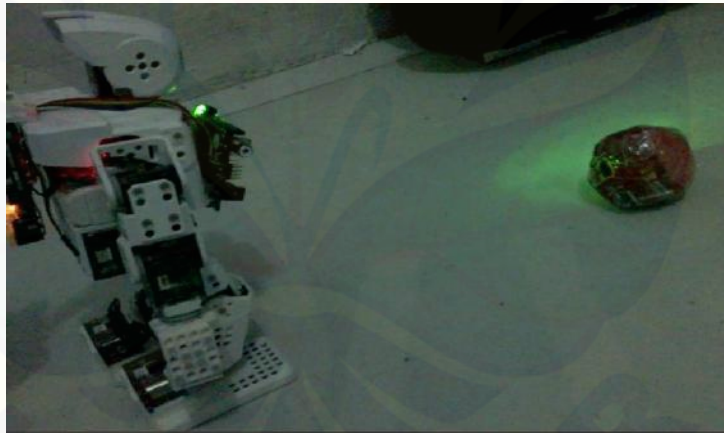
Tabel 4.14 Data pengujian saat robot jatuh kebelakang dan jatuh ke depan

NO	Sensor Depan		Sensor Belakang		Kondisi cahaya
	Bangun	Jatuh	Bangun	Jatuh	
1	4.4 V	4.8 V	4.5 V	4.8 V	Dalam ruangan
2	3.4 V	4.6 V	3.3 V	4.8 V	Luar ruangan

Pada tabel 4.14 diketahui pada saat robot kondisi bangun photodiode bagian depan mendeteksi nilai tegangan pada saat berada dalam ruangan dengan tegangan 4.4 volt dan bila terjatuh maka nilai akan naik 4.8 volt. Hasil pembacaan ini dapat di buat sebagai data program robot bangun dari jatuh. Bila nilai adc terbaca > 4.4 volt maka program akan memanggil program bangun dari jatuh.

4.1.4 Pengujian keseluruhan Robot *Humanoid soccer*

Pengujian robot *soccer* secara keseluruhan dilakukan dengan cara robot bergerak mencari bola, mendeteksi bola, mendekati bola dan menendang bola.

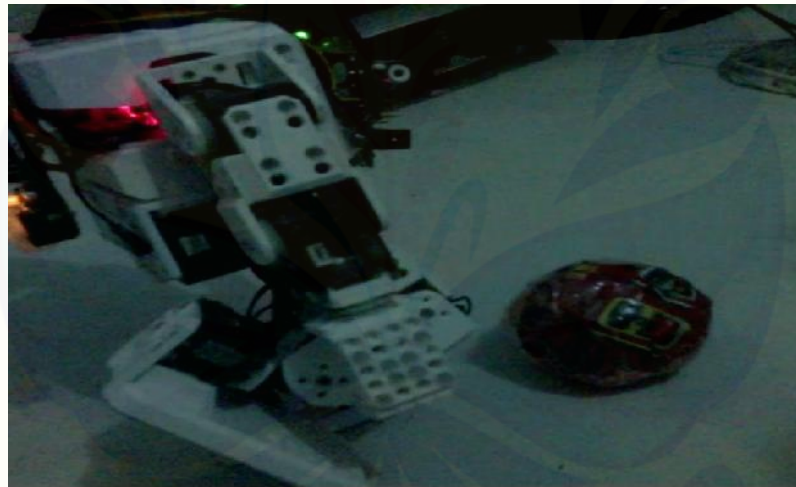
Gambar 4.28 Robot mendekati bola *soccer*

Robot mula- mula akan mencari dan mendeteksi apakah di daerah terdekat ada bola *soccer* apa tidak. Bila terdeteksi citra warna bola maka robot akan mendekati bola tersebut sesuai dengan data *pixels* yang terbaca. Apabila data *pixels* sudah terpenuhi maka robot akan bersiap untuk menendang bola dengan cara perlahan mendekati bola satu langkah dan terus menendang.



Gambar 4.29 Robot membaca nilai *pixels*

Pembacaan robot dengan jarak 5 cm dengan *pixels* 75 maka robot akan menendang bola *soccer* dengan waktu jeda 10 detik.



Gambar 4.30 Robot siap menendang bola



Gambar 4.31 Robot menendang bola



Gambar 4.32 Robot menghempaskan bola *soccer*

Dari hasil pengambilan data dan pergerakan robot di dapat posisi robot akan menendang bola dengan nilai *pixels* 75. Nilai ini akan di gunakan sebagai acuan robot untuk menendang bola.

Apabila robot tidak menemukan atau tidak mendeteksi adanya suatu citra bola *soccer* maka robot akan terus bergerak maju atau bergeser sampai ditemukan suatu objek bola yang sudah di gunakan sebagai target.



BAB V. PENUTUP

Pada bagian akan dibahas mengenai hasil dan kelemahan dari sistem yang telah dibuat. Setelah melakukan perencanaan, pembuatan dan implementasi robot *humanoid soccer*, kemudian dilakukan pengujian dan analisa maka dapat diambil kesimpulan dan saran- saran sebagai berikut:

5.1. Kesimpulan

Dengan memperhatikan data pengamatan dan analisis pada bab sebelumnya dapat disimpulkan bahwa:

1. Sensor kamera *CMUCAM4* dapat digunakan untuk melacak warna pada bola dengan rentang nilai yang diprogram dengan jarak 5cm dan 10cm dengan pembacaan data $mx \leq 68$ dan data $pixels < 65$. (Tabel.4.11 halaman 72)
2. Robot *humanoid* dapat melakukan gerakan bangkit berdiri ketika terjatuh yang dideteksi menggunakan sensor photodiode dengan pembacaan rata-rata pembacaan > 4.5 volt untuk dalam ruangan. (Tabel 4.14 halaman 75)
3. Robot *humanoid* berhasil menendang bola dengan jarak 5 cm $\geq 75 pixels$

5.2. Saran

Saran-saran yang dapat diberikan untuk perbaikan dan pengembangan dari Penelitian ini adalah sebagai berikut:

1. Struktur robot yang lebih ringan dan kuat agar robot *humanoid* pemain bola dapat berjalan lebih cepat.
2. Dibutuhkan metode untuk mendeteksi posisi tengah dari gawang sehingga robot dapat lebih tepat menentukan arah gerak misalnya dengan bantuan sensor kompas.
3. Dibutuhkan algoritma untuk menambah fungsi robot untuk mendeteksi posisi gawang dan posisi bola secara terus menerus dengan proses yang cepat.
4. Robot harus melakukan aksi jalan dan *scanning* ketika tidak mendeteksi bola, yaitu pada saat bola jauh di depan robot.

DAFTAR PUSTAKA

- Ahmad, Usman. “*Pengolahan Citra Digital dan Teknik Pemrogramannya*”. Graha Ilmu, Yogyakarta. 2005.
- Anjaswati, Irma., 2013, Sensor Photodiode,
http://irmatrianjaswatifst11.web.unair.ac.id/artikel_detail-84996
Sensorsensor%20photodiode.html, diakses pada tanggal 27 Mei 2015.
- Arduino Indonesia, 2013, *arduino uno*,
http://arduino.or.id/hardware/detail_hardware/8, diakses 26 oktober 2013
- Embedded Vision module for Bioloid Quick Start [cited 2015 Mei 13]. Available from:<http://robosavvy.com/RoboSavvyPages/Support/Hamid/HaViMoQuickStart.pdf>
- HaViMo2, Image Processing Module March 13, 2010 [cited 2015 May 7] Available :
from: <http://robosavvy.com/RoboSavvyPages/Support/Hamid/HaViMo2.pdf>
- Kenji Kaneko, et.al, *Design of Prototye Humanoid Robotics Platform forHRP*”Proceedings of the 2002 IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, 2002.
- Kaehler, Gary Bradski and Adrian. 2008. *Learning OpenCV - Computer Vision with the OpenCV Library*. O’Reilly Media, Inc. CA.
- Mechatronicscrew, 2013. Pemilihan Motor Servo Dynamixel
(Sumber:<http://ksatriaunisi.wordpress.com/2013/01/11/pemilihan-motor-servo-dynamixel/#more-29>, diakses pada tanggal 21 Mei 2015)
- Munir, Rinaldi. “*Pengolahan Citra Digital dengan Pendetakan Algoritmik*”. Informatika Bandung, Bandung. 2004.
- ROBOTIS Co., Ltd. ROBOTIS: robot actuator dynamixel AX-12+ [cited 2015 Mei 13]. Available: http://support.robotis.com/en/product/dynamixel/ax_series/ax-12.htm

ROBOTIS Co., Ltd. ROBOTIS: roboplus task software [cited 2015 Mei 13].

Available:

http://support.robotis.com/en/software/roboplus/roboplus_task_main.htm

ROBOTIS Co., Ltd. ROBOTIS: roboplus manager software [cited 2015 Mei 13].

Available :

http://support.robotis.com/en/software/roboplus/roboplus_manager_main.htm

ROBOTIS Co., Ltd. ROBOTIS: roboplus motion software [cited 2015 Mei 13].

Available:

from:http://support.robotis.com/en/software/roboplus/roboplus_motion_main.htm

Tim IE AN206, Pan Tilt Color Tracking menggunakan HaViMo 2.0 dan DT-AVR
Low Cost Micro System, *Application*

Note(Sumber:http://www.innovativeelectronics.com/innovative_electronics/download_files/artikel/AN206.pdf, diakses pada tanggal 19 Mei 2015)



LAMPIRAN

1. Listing program mikrokontroler Arduino Uno

```
#include <CMUcam4.h>
#include <CMUcom4.h>
#include <LiquidCrystal.h>
// RED COLOR TRACKING
DEFINITION*****
#define RED_MIN 100
#define RED_MAX 255
#define GREEN_MIN 5
#define GREEN_MAX 100
#define BLUE_MIN 5
#define BLUE_MAX 100
//*****
*****

#define LED_BLINK 5 // 5 Hz
#define WAIT_TIME 5000 // 5 seconds

#define PIXELS 75 // The percent of tracked pixels needs to be greater than this
0=0% - 255=100%.
#define CONFIDENCE 11 // The percent of tracked pixels in the bounding box
needs to be greater than this 0=0% - 255=100%.

#define NOISE_FILTER 2 // Filter out runs of tracked pixels smaller than this in
length 0 - 255.

CMUcam4 cam(CMUCOM4_SERIAL1);
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
const int analogPin = A1;
const int analogPin1 = A0;
float threshold = 4.7;
float threshold1 = 4.7;
const int bangun1 = 13;
int bangun2=12;
int maju=11;
int geser=10;
int cari=9;
int tendang=8;
```



```
//Setup
*****

void setup(){
  pinMode(bangun1, OUTPUT);
  pinMode(bangun2, OUTPUT);//set mlf pin as output pin
  pinMode(maju, OUTPUT);//set mlf pin as output pin
  pinMode(geser, OUTPUT);//set m2f pin as output pin
  pinMode(cari, OUTPUT);
  pinMode(tendang, OUTPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  cam.begin();

  cam.automaticPan(true, false);
  cam.automaticTilt(true, false);
  // Wait for auto gain and auto white balance to run.

  cam.LEDOn(LED_BLINK);
  delay(WAIT_TIME);
  // Turn auto gain and auto white balance off.

  cam.colorTracking(false); // Go to YUV mode! False for RGB mode!

  cam.autoGainControl(false);
  cam.autoWhiteBalance(false);

  // See automatic pan and tilt for more details.

  cam.LEDOn(CMUCAM4_LED_ON);

  cam.noiseFilter(NOISE_FILTER);

  cam.setTrackingParameters(RED_MIN, RED_MAX, GREEN_MIN,
  GREEN_MAX, BLUE_MIN, BLUE_MAX);
}
//*****
****

void loop(){
  CMUcam4_tracking_data_t data;
```

```
for(;;){
  cam.trackColor();
  cam.getTypeTDataPacket(&data);
//lcd
  lcd.setCursor(0, 0);
  lcd.print(data.confidence);
  lcd.print(" ");
  lcd.setCursor(6, 0);
  lcd.print("CV");

  lcd.setCursor(9, 0);
  lcd.print(data.pixels);
  lcd.print(" ");
  lcd.setCursor(14, 0);
  lcd.print("PX");

  lcd.setCursor(0, 1);
  lcd.print(data.mx);
  lcd.print(" ");
  lcd.setCursor(6, 1);
  lcd.print("MX");

  lcd.setCursor(9, 1);
  lcd.print(data.my);
  lcd.print(" ");
  lcd.setCursor(14, 1);
  lcd.print("MY");

// INISIALISASI SENSOR PHOTODIODA
int analogValue = analogRead(analogPin);
float voltage = analogValue * (5.0 / 1023.0);
int analogValue1 = analogRead(analogPin1);
float voltage1 = analogValue1 * (5.0 / 1023.0);

// btas

if (data.confidence >= 1)
  { if (voltage < threshold)
    {
      digitalWrite(bangun1, HIGH);
```

```
digitalWrite(bangun2,LOW);
digitalWrite(maju,LOW);
digitalWrite(geser,LOW);
digitalWrite(cari,LOW);
digitalWrite(tendang,LOW); }

else if (voltage1 < threshold1)
{
digitalWrite(bangun1, LOW);
digitalWrite(bangun2,HIGH);
digitalWrite(maju,LOW);
digitalWrite(geser,LOW);
digitalWrite(cari,LOW);
digitalWrite(tendang,LOW); }

else if ((data.mx <= 68) && (data.pixels < 65))
{
digitalWrite(bangun1, LOW);
digitalWrite(bangun2,LOW);
digitalWrite(maju,HIGH);
digitalWrite(geser,LOW);
digitalWrite(cari,LOW);
digitalWrite(tendang,LOW);
}

else if ((data.mx <= 50)&& (data.pixels < 65))
{
digitalWrite(bangun1, LOW);
digitalWrite(bangun2,LOW);
digitalWrite(maju,LOW);
digitalWrite(geser,HIGH);
digitalWrite(cari,LOW);
digitalWrite(tendang,LOW);
}
else if ((data.mx >= 90) && (data.pixels < 65))
{
digitalWrite(bangun1, LOW);
digitalWrite(bangun2,LOW);
digitalWrite(maju,LOW);
digitalWrite(geser,HIGH);
digitalWrite(cari,LOW);
digitalWrite(tendang,LOW);
}
}
```

```
        cam.automaticPan(true, true);
        cam.automaticTilt(true, true);
    }
else if (data.confidence <= 0){
    if (voltage < threshold)
    {
        digitalWrite(bangun1, HIGH);
        digitalWrite(bangun2,LOW);
        digitalWrite(maju,LOW);
        digitalWrite(geser,LOW);
        digitalWrite(cari,LOW);
        digitalWrite(tendang,LOW); }

    else if (voltage1 < threshold1)
    {
        digitalWrite(bangun1, LOW);
        digitalWrite(bangun2,HIGH);
        digitalWrite(maju,LOW);
        digitalWrite(geser,LOW);
        digitalWrite(cari,LOW);
        digitalWrite(tendang,LOW); }
else
    {
        digitalWrite(bangun1, LOW);
        digitalWrite(bangun2,LOW);
        digitalWrite(maju,LOW);
        digitalWrite(geser,LOW);
        digitalWrite(cari,HIGH);
        digitalWrite(tendang,LOW);
    }
}
}
```

2. Listing program mikrokontroler OpenCM 9.04

```
/*
 * Robot soccer
 * Created By: Slamet Purwanto
 * University of Jember, Indonesian
 */

/* Dynamixel ID defines */
****LEFT LEG****/
#define ID_NUM_8 8
#define ID_NUM_10 10
#define ID_NUM_12 12
#define ID_NUM_14 14
#define ID_NUM_16 16
****RIGHT LEG****/
#define ID_NUM_7 7
#define ID_NUM_9 9
#define ID_NUM_11 11
#define ID_NUM_13 13
#define ID_NUM_15 15
****LEFT ARM****/
#define ID_NUM_2 2
#define ID_NUM_4 4
#define ID_NUM_6 6
****RIGHT ARM****/
#define ID_NUM_1 1
#define ID_NUM_3 3
#define ID_NUM_5 5

/* Control table defines */
#define P_GOAL_POSITION 30
#define P_GOAL_SPEED 32
#define P_GAIN 29

/* Define Global Variables */
#define setupDelay 100
#define DELAY 500
/* Serial device defines for dxl bus */
#define DXL_BUS_SERIAL1 1 // <-OpenCM9.04

int value = digitalRead(1); // read #1, value assigned
int value1 = digitalRead(2);
```

```
int value2 = digitalRead(3);
int value3 = digitalRead(4);
int value4 = digitalRead(5);

Dynamixel Dxl(DXL_BUS_SERIAL1);

word SyncPag1[48]= //initial pose
{
  //left leg
  ID_NUM_8,512,1000,
  ID_NUM_10,542,1000, //50 offset
  ID_NUM_12,461,1000, //100 offset
  ID_NUM_14,437,1000, //75 offset
  ID_NUM_16,512,1000,
  //right leg
  ID_NUM_7,512,1000,
  ID_NUM_9,462,1000, //50 offset
  ID_NUM_11,612,1000, //100 offset
  ID_NUM_13,587,1000, // 75 offset
  ID_NUM_15,512,1000,
  //left arm
  ID_NUM_2,512,1000,
  ID_NUM_4,762,1000, //250 offset
  ID_NUM_6,512,1000,
  //right arm
  ID_NUM_1,512,1000,
  ID_NUM_3,262,1000, //250 offset
  ID_NUM_5,512,1000};

word Bangun1[100]=
{
  ID_NUM_3,512,1000,
  //right arm
  ID_NUM_4,512,1000,
  ID_NUM_7,187,1000, //250 offset
  ID_NUM_8,830,1000
}; //50
word Bangun2[100]=
{
  ID_NUM_1,256,1000,
  ID_NUM_2,768,1000};
word Bangun3[100]=
{
```

```
ID_NUM_3,300,1000,  
ID_NUM_4,700,1000};  
word Bangun4[100]=  
{  
ID_NUM_3,512,1000,  
ID_NUM_4,512,1000};
```

```
word Bangun5[100]=  
{ ID_NUM_1,768,1000,  
ID_NUM_2,256,1000,
```

```
};  
word Bangun6[100]=  
{  
ID_NUM_3,262,1000,  
ID_NUM_4,762,1000};
```

```
word Bangun7[100]=  
{  
ID_NUM_12,187,1000,  
ID_NUM_11,830,1000,  
ID_NUM_9,256,1000,  
ID_NUM_10,768,1000,  
ID_NUM_15,187,1000,  
ID_NUM_16,830,1000};
```

```
word Bangun8[100]=  
{  
ID_NUM_7,522,1000,  
ID_NUM_8,502,1000,  
ID_NUM_15,512,1000,  
ID_NUM_16,512,1000  
};
```

```
word Bangun9[100]=  
{  
ID_NUM_9,187,1000,  
ID_NUM_10,830,1000,  
};
```

```
word Bangun10[100]=  
{  
ID_NUM_9,462,1000,
```

```
ID_NUM_10,562,1000,  
  ID_NUM_11,612,1000,  
ID_NUM_12,412,1000  
};
```

```
// program jalan  
word Jalan2[48]=
```

```
{  
  ID_NUM_1,687,1000,  
  ID_NUM_2,387,1000,  
  ID_NUM_9,401,1000,  
  ID_NUM_10,582,1000,  
  //right leg  
  ID_NUM_13,547,1000,  
  ID_NUM_14,437,1000,  
  ID_NUM_15,520,1000,  
  ID_NUM_16,548,1000,  
  }  
;
```

```
word Jalan3[48]=
```

```
{  
  //left leg  
  ID_NUM_1,687,1000,  
  ID_NUM_2,387,1000,  
  ID_NUM_9,422,1000,  
  ID_NUM_10,662,1000,  
  ID_NUM_12,312,1000,  
  //right leg  
  ID_NUM_13,566,1000,  
  ID_NUM_14,407,1000,  
  ID_NUM_15,520,1000,  
  ID_NUM_16,520,1000,  
  }  
;
```

```
word Jalan4[48]=
```

```
{  
  //left leg  
  ID_NUM_1,687,1000,  
  ID_NUM_2,387,1000,  
  ID_NUM_7,512,1000,  
  ID_NUM_8,512,1000,
```



```
ID_NUM_9,442,1000,  
ID_NUM_10,642,1000,  
ID_NUM_12,411,1000,  
//right leg  
ID_NUM_13,587,1000,  
ID_NUM_14,477,1000,  
ID_NUM_15,510,1000,  
ID_NUM_16,510,1000,  
}  
;
```

```
word Jalan5[48]=  
{  
//left leg  
ID_NUM_1,687,1000,  
ID_NUM_2,387,1000,  
ID_NUM_9,401,1000,  
ID_NUM_10,582,1000,  
//right leg  
ID_NUM_13,542,1000,  
ID_NUM_14,437,1000,  
ID_NUM_15,471,1000,  
ID_NUM_16,505,1000  
}  
;
```

```
word Jalan6[48]=  
{  
//left leg  
ID_NUM_1,687,1000,  
ID_NUM_2,387,1000,  
ID_NUM_11,712,1000,  
ID_NUM_9,352,1000,  
ID_NUM_10,602,1000,  
ID_NUM_12,412,1000,  
//right leg  
ID_NUM_13,602,1000,  
ID_NUM_14,457,1000,  
ID_NUM_15,482,1000  
}
```

```
word Jalan7[48]=  
{  
//left leg  
ID_NUM_1,687,1000,
```

```
ID_NUM_2,387,1000,  
ID_NUM_7,512,1000,  
ID_NUM_8,512,1000,  
ID_NUM_9,402,1000,  
ID_NUM_10,582,1000,  
ID_NUM_12,412,1000,  
//right leg  
ID_NUM_11,612,1000,  
ID_NUM_13,547,1000,  
ID_NUM_14,437,1000,  
ID_NUM_15,510,1000,  
ID_NUM_16,510,1000,  
}  
;  
  
// program nendang  
word Nendang1[100]=  
{  
  ID_NUM_16,457,1000,  
  //right arm  
  ID_NUM_15,467,1000,  
  ID_NUM_7,462,1000,  
  ID_NUM_8,487,1000  
};  
  
  word Nendang2[100]=  
{  
  ID_NUM_13,547,1000,  
  ID_NUM_16,440,1000,  
  //right arm  
  ID_NUM_15,442,1000,  
  ID_NUM_7,442,1000,  
};  
  
word Nendang5[100]=  
{  
  ID_NUM_11,812,1000  
};  
  
word Nendang3[100]=  
{  
  ID_NUM_9,312,1000,
```

```
ID_NUM_2,256,1000,  
//right arm  
ID_NUM_1,256,1000,  
ID_NUM_11,512,1000  
};
```

```
word Nendang4[100]=  
{  
  ID_NUM_9,462,1000,  
  ID_NUM_2,512,1000,  
  //right arm  
  ID_NUM_1,512,1000,  
  ID_NUM_11,612,1000  
};
```

```
// program duduk  
word Duduk2[100]=  
{  
  ID_NUM_12,187,1000,  
  ID_NUM_11,830,1000,  
  ID_NUM_9,226,1000,  
  ID_NUM_10,798,1000,  
};
```

```
word Duduk3[100]=  
{  
  ID_NUM_1,768,1000,  
  ID_NUM_2,256,1000,  
  ID_NUM_12,250,1000,  
  ID_NUM_11,760,1000,  
  ID_NUM_9,262,1000,  
  ID_NUM_10,768,1000,  
};
```

```
// program cari bola  
word Carbol4[100]=  
{ ID_NUM_1,768,1000,  
  ID_NUM_2,256,1000,  
  ID_NUM_7,462,1000,  
  ID_NUM_8,572,1000,  
  ID_NUM_15,492,1000,  
  ID_NUM_16,542,1000,  
};
```

```
word Carbol5[100]=
{ ID_NUM_1,768,1000,
  ID_NUM_2,256,1000,
  ID_NUM_7,512,1000,
  ID_NUM_8,512,1000,
  ID_NUM_15,512,1000,
  ID_NUM_16,512,1000,
};

void setup(){
  // Dynamixel 2.0 Protocol -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps
  pinMode(1, INPUT);
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);

  Dxl.begin(3);
  //Set all dynamixels as same condition.
  delay(setupDelay);
  Dxl.writeWord( BROADCAST_ID, P_GOAL_SPEED, 1000 );
  delay(setupDelay);
  Dxl.writeWord( BROADCAST_ID, P_GAIN, 64);
  delay(setupDelay);
  Dxl.syncWrite(30,2,SyncPag1,48);
  delay(setupDelay);
}

void loop(){
  int value = digitalRead(1);
  int value1 = digitalRead(2);
  int value2 = digitalRead(3);
  int value3 = digitalRead(4);
  int value4 = digitalRead(5);

  if ( value == HIGH)
  {
    Dxl.syncWrite(30,2,SyncPag1,48);
    delay(5000);
    Dxl.syncWrite(30,2,Bangun1,48);
    delay(DELAY);
    Dxl.syncWrite(30,2,Bangun2,48);
    delay(DELAY);
  }
}
```

```
Dxl.syncWrite(30,2,Bangun3,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun4,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun5,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun6,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun7,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun8,48);
delay(1000);
Dxl.syncWrite(30,2,Bangun9,48);
delay(1000);
Dxl.syncWrite(30,2,Bangun10,48);
delay(1000);
Dxl.syncWrite(30,2,SyncPag1,48);
delay(1000);
}

// push1
else if ( value1 == HIGH)
{
Dxl.syncWrite(30,2,SyncPag1,48);
delay(5000);
Dxl.syncWrite(30,2,Bangun1,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun2,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun3,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun4,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun5,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun6,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun7,48);
delay(DELAY);
Dxl.syncWrite(30,2,Bangun8,48);
delay(1000);
Dxl.syncWrite(30,2,Bangun9,48);
delay(1000);
}
```

```
Dxl.syncWrite(30,2,Bangun10,48);
delay(1000);
Dxl.syncWrite(30,2,SyncPag1,48);
delay(1000);
}
// push 2
else if ( value2 == HIGH)
{

    Dxl.syncWrite(30,2,Jalan2,48);
    delay(150);
    Dxl.syncWrite(30,2,Jalan3,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan4,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan5,48);
    delay(150);
    Dxl.syncWrite(30,2,Jalan6,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan7,48);
    delay(75);

    Dxl.syncWrite(30,2,SyncPag1,48);
    delay(5000);
    Dxl.syncWrite(30,2,Jalan2,48);
    delay(150);
    Dxl.syncWrite(30,2,Jalan3,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan4,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan5,48);
    delay(150);
    Dxl.syncWrite(30,2,Jalan6,48);
    delay(75);
    Dxl.syncWrite(30,2,Jalan7,48);
    delay(75);

    Dxl.syncWrite(30,2,SyncPag1,48);
    delay(2500);
    Dxl.syncWrite(30,2,Nendang1,48);
    delay(500);
    Dxl.syncWrite(30,2,Nendang2,48);
    delay(500);
```

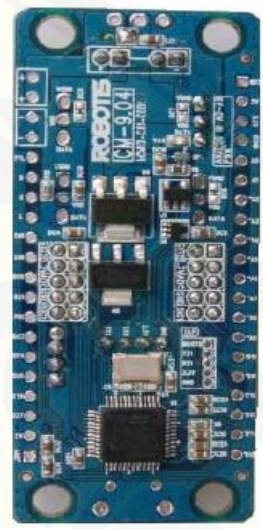
```
Dxl.syncWrite(30,2,Nendang5,48);
delay(500);
Dxl.syncWrite(30,2,Nendang3,48);
delay(500);
Dxl.syncWrite(30,2,Nendang4,48);
delay(500);
Dxl.syncWrite(30,2,Nendang2,48);
delay(500);
Dxl.syncWrite(30,2,Nendang1,48);
delay(500);
Dxl.syncWrite(30,2,SyncPag1,48);
delay(2500);
}
// push 3
else if ( value3 == HIGH)
{
  Dxl.syncWrite(30,2,Carbol4,48);
  delay(200);
  Dxl.syncWrite(30,2,Carbol5,48);
  delay(200);
}
// psh 4
else if ( value4 == HIGH)
{
  Dxl.syncWrite(30,2,SyncPag1,48);
  delay(5000);
  Dxl.syncWrite(30,2,Jalan2,48);
  delay(150);
  Dxl.syncWrite(30,2,Jalan3,48);
  delay(75);
  Dxl.syncWrite(30,2,Jalan4,48);
  delay(75);
  Dxl.syncWrite(30,2,Jalan5,48);
  delay(150);
  Dxl.syncWrite(30,2,Jalan6,48);
  delay(75);
  Dxl.syncWrite(30,2,Jalan7,48);
  delay(75);
}
else {
  Dxl.syncWrite(30,2,SyncPag1,48);
  delay(100);
} }
```

4. Datasheet OpenCm9.04 Hardware

① Illustration of OpenCM9.04

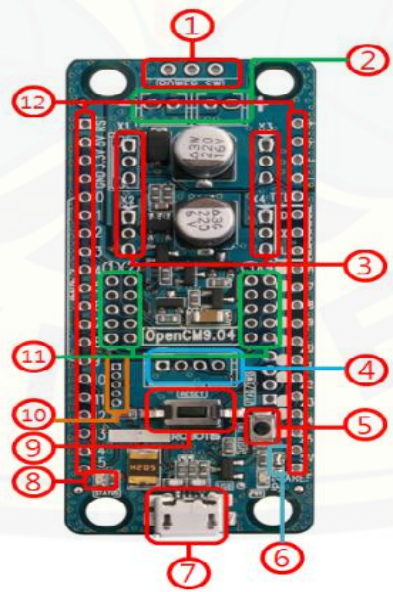


[TOP]



[BOTTOM]

② Parts label



1. Power Switch: battery connection (power input).
2. Battery socket: sockets to connect Li-Ion Battery(ies). [Note: each battery provides 3.7V; batteries in both pins provide 7.4V]
3. DYNAMIXEL TTL BUS: ports for TTL-based Dynamixels. Connected dynamixels can be daisy-chained.
4. USART PIN: connect 4-pin devices such as the BT-110, BT-210, ZIG-110, LN-101 for UART communications (note: the LN-101 firmware only allows communications with the PC).
5. User Switch: user-programmable switch; OpenCM9.04 recovery-mode
6. Analog Reference Selection Jumper : change to 5V for analog signals.
7. 7. Micro-B USB: Connect the OpenCM9.04 for communications, downloads, and
8. 5V input power supply. Any type-B micro USB cable for Android phone is useful.
9. 8. Status LED: Test LED for OpenCM9.04's programming. The LED blinks with a high/low signal to pin D16.
10. 9. Reset switch: resets the CPU.
11. 10. JTAG/SWD 4 PIN: Via JTAG/SWD terminal implement other programs such as IAR, Keil. The OpenCM9.04 has a total of 128Kbytes of memory, downloads and stores the bootloader's binary starting at 0x08000000. (Bootloader: 0x08000000 to

0x08002FFF. User programming space:

12. 0x08003000 to 0x08020000).

13. 11. External Sensor PIN: pins for Robotis' sensors.

14. 12. 2.54 mm GPIO Header: Interface external devices to the OpenCM9.04's

15. STM32F103CB CPU.

③ Product package

A Type Board Only

Package		quantity
controller	OpenCM9.04	1
Manual	User Guide	1

B type

Package		quantity
controller	OpenCM9.04	1
PIN Header	1x20 Pin Header	2
BOX Header	1x20 BOX Header	2
USB cable	Micro B Cable	1
Manual	User Guide	1

④ Product specifications

	OpenCM9.04
CPU	STM32F103CB (ARM Cortex-M3)
Op Voltage	5V~16V(USB 5V, DXL 12V, XL-Series 7.4V)
I/O	GPIO 26
Timer	8 (16bit)
Analog In(ADC)	10 (12bit)
Flash	128 Kbytes
SRAM	20 Kbytes
Clock	72Mhz(9 X 8 Mhz)
USB	1 (2.0 FullSpeed) Micro B type
CAN	1
USART	3
SPI	2
I2C(TWI)	2
Debug	JTAG & SWD
Extenal SenSor	4
3 Pin TTL	4(XL combo 3 PIN)
SW Tool	ROBOTIS OpenCM
SIZE	27mm X 66.5 mm

⑤ Power

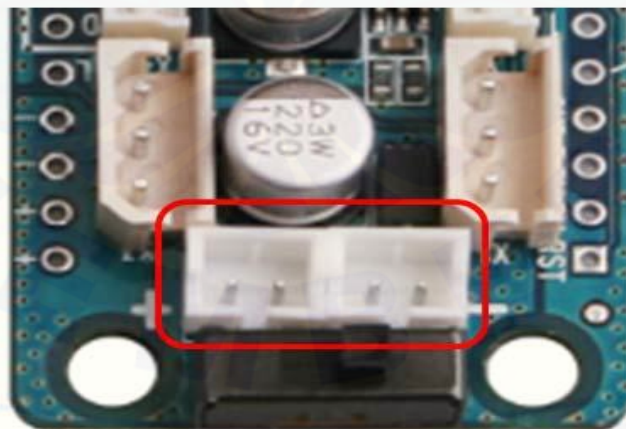
A. Connect to USB power

For programming simply connect the OpenCM9.04 via USB. The LED blinks I/O control under 5V becomes available.



B. Connect to battery power

To control Dynamixel(s) connect the proper battery product. Afterwards the board runs the pre-downloaded mode.



⑥ How-to-operate

- A. The OpenCM9.04 executes the user code (0x08003000) when powered under battery or USB.
- B. Connect to the PC via USB, write the program in the integrated development environment (IDE), compile and download the program.



<connect the OpenCM9.04
to PC>

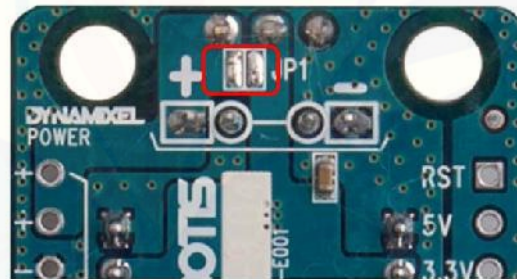
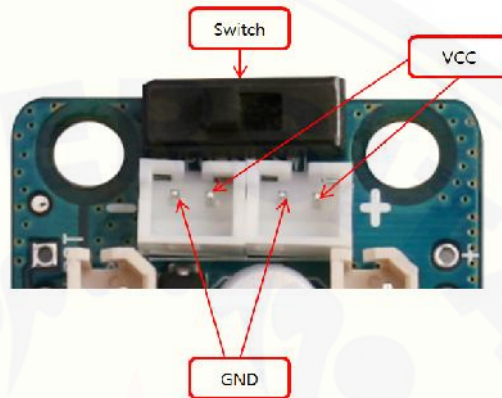
A screenshot of the ROBOTIS OpenCM IDE. The window title is "a_Blink2 | ROBOTIS OpenCM v1.0.1". The code editor shows a C++ program for a blink LED. The code includes comments in Korean and English, and defines setup and loop functions. The status bar at the bottom indicates "ROBOTIS OpenCM 04 on 0.044".

```
a_Blink2 | ROBOTIS OpenCM v1.0.1
파일 편집 스킴 도구 도움말
c_Blink2
//
Blink2
Tur is on the built-in(LED) LED on for 0.1 second, then off for 0.1 second
repeatedly using digitalWrite() which is function only for built-in LED.
Ported to OpenCM Scripts from the Arduino example 27 May 2011
By ROBOTIS, LTD
*/
void setup() {
  // Set up the built in LED pin as an output:
  pinMode(EQ-A33_LED_PIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_PIN, HIGH); // Turns digital value based on current value.
  delay(100); // Wait for 0.1 second.
```

<OpenCM9.04 IDE ROBOTIS OpenCM>

⑦ Pin information

A. Power and switch

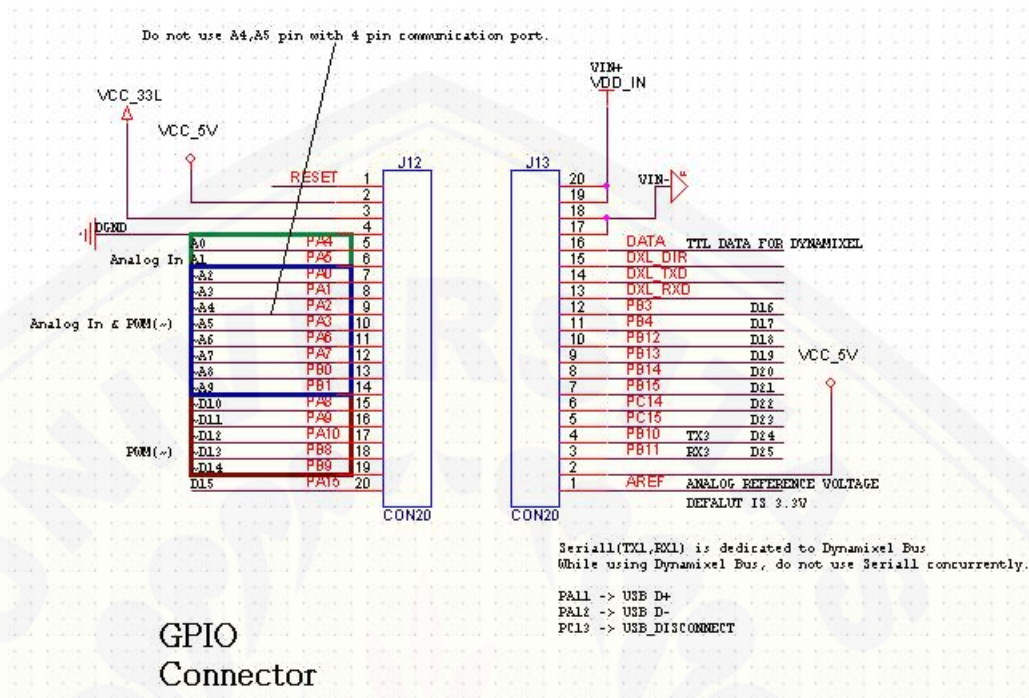


For switch use de-solder JP1

B. Refer to the OpenCM9.04's schematic for GPIO header pins to interface to the STM32F103CB CPU

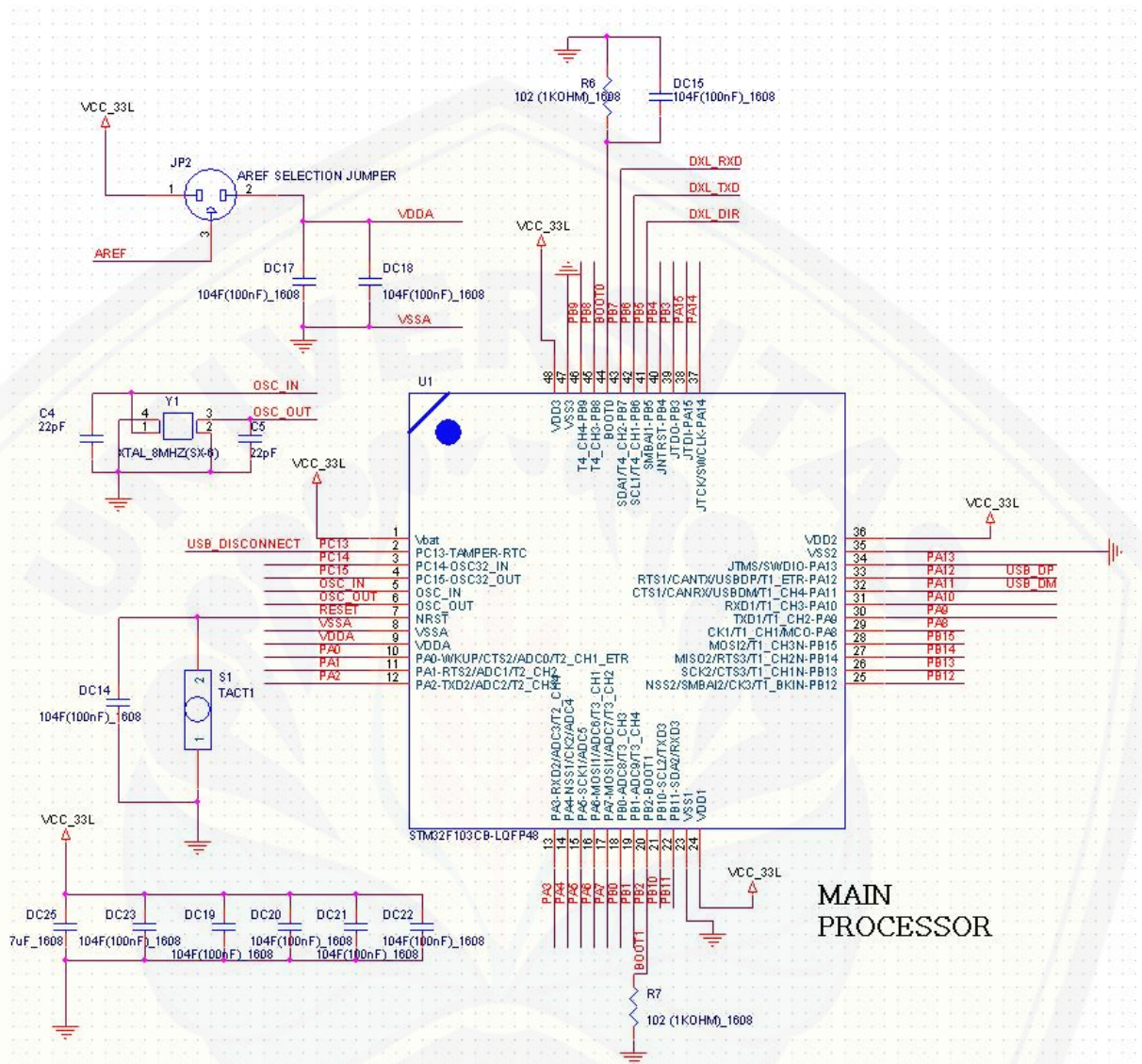


[OpenCM9.04
GPIO]



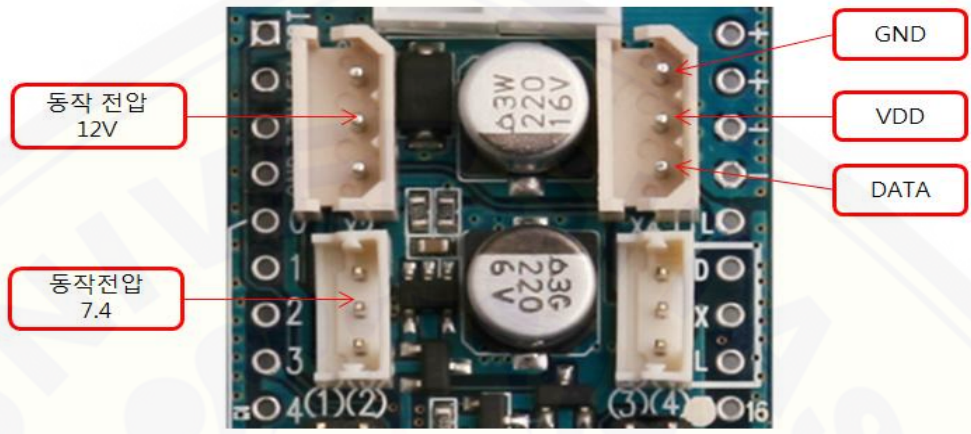
[OpenCM9.04 GPIO schematic]

For reference VDD is 12V; Dynamixel-related portions (pins PB5, PB6, PB7) have been wired separately so they are not available for use

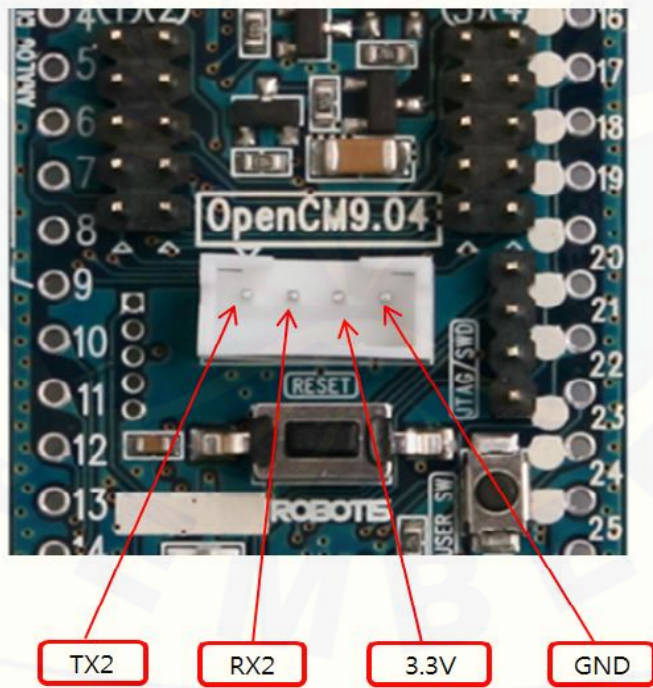


< STM32F103CB(LQFP48 Package) CPU connection schematic >

C. DYNAMIXEL TTL BUS



D. 4-pin communication devices



The same labels shown on the reverse side.



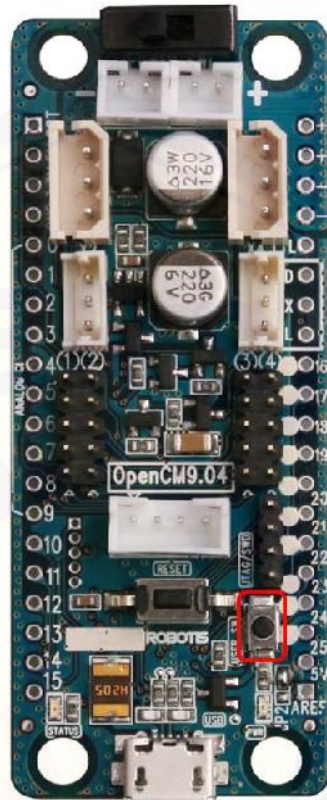
⑧ Schematic and PCB Gerber file (Schematic & Gerber Design)

The OpenCM9.04 resources (both hardware and software) are 100% open. Resources can be obtained via gitHub with the link below.

<https://github.com/robotis-pandora/ROBOTIS-OpenCM.git>

⑨ Emergency recovery mode

A. Whether the OpenCM9.04 USB drivers are not initialized nor device detected press the switch and connection will reestablish.



B. When connected via USB check the STATUS LED.



C. Go to File -> examples -> Digital -> Blink example; download it and

press the button (any other example OK).



6. Datasheet servo dynamixel xl-320



XL-320

H/W Specification

- Weight : 16.7g
- Dimension : 24mm *36mm * 27mm
- Resolution : 0.29°
- Motor : Cored Motor
- Gear Reduction Ratio : 238 : 1
- Stall Torque : 0.39 N.m (at 7.4V)
- No load speed : 114 rpm (at 7.4V)
- Running Degree
 - 0° ~ 300°
 - Endless Turn
- Running Temperature : -5°C ~ +70°C
- Voltage : **6 ~ 8.4V (Recommended Voltage 7.4V)**
- Command Signal : Digital Packet
- Protocol Type : Half duplex Asynchronous Serial Communication (8bit,1stop,No Parity)
- Link (Physical) : TTL Level Multi Drop (daisy chain type Connector)
- ID : 253 ID (0~252)
- Communication Speed : 7343bps ~ 1 Mbps
- Feedback : Position, Temperature, Load, Input Voltage, etc.
- Material : Engineering Plastic

Area	Address (Hexadecimal)	Size(byte)	Name	Description	Access	Initial Value	Min	Max
EEPROM	0	2	Model Number	Model number	R	350	-	-
	2	1	Version of Firmware	Information on the version of firmware	R	-	-	-
	3	1	ID	ID of Dynamixel	RW	1	0	252
	4	1	Baud Rate	Baud Rate of Dynamixel	RW	3	0	3
	5	1	Return Delay Time	Return Delay Time	RW	250	0	254
	6	2	CW Angle Limit	clockwise Angle Limit	RW	0	0	1023
	8	2	CCW Angle Limit	counterclockwise Angle Limit	RW	1023	0	1023
	11	1	Control Mode	Control Mode	RW	2	1	2
	12	1	Limit Temperature	Internal Limit Temperature	RW	65	0	150
	13	1	lower Limit Voltage	Lowest Limit Voltage	RW	60	50	250
	14	1	Upper Limit Voltage	Upper Limit Voltage	RW	90	50	250
	15	2	Max Torque	Lowest byte of Max. Torque	RW	1023	0	1023
	17	1	Return Level	Return Level	RW	2	0	2
	18	1	Alarm Shutdown	Shutdown for Alarm	RW	3	0	7
R7AM	24	1	Torque Enable	Torque On/Off	RW	0	0	1
	25	1	LED	LED On/Off	RW	0	0	7
	27	1	D Gain	D Gain	RW	0	0	254
	28	1	I Gain	I Gain	RW	0	0	254
	29	1	P Gain	P Gain	RW	32	0	254
	30	2	Goal Position	Goal Position	RW	-	0	1023
	32	2	Goal Velocity	Goal Speed	RW	-	0	2047
	35	2	Goal Torque	Goal Torque	RW	-	0	1023
	37	2	Present Position	Current Position	R	-	-	-
39	2	Present Speed	Current Speed	R	-	-	-	
41	2	Present Load	Current Load	R	-	-	-	

45	1	Present Voltage	Current Voltage	R	-	-	-
46	1	Present Temperature	Present temperature	R	-	-	-
47	1	Registered Instruction	Registered Instruction	R	0	-	-
49	1	Moving	Moving	R	0	-	-
50	1	Hardware Error Status	Hardware error status	R	0	-	-
51	2	Punch	Punch	RW	32	0	1023

Control Table

Control Table consists of data regarding the current status and operation, which exists inside of Dynamixel. The user can control Dynamixel by changing data of Control Table via Instruction Packet.

EEPROM and RAM

Data in RAM area is reset to the initial value whenever the power is turned on while data in EEPROM area is kept once the value is set even if the power is turned off.

Address

It represents the location of data. To read from or write data to Control Table, the user should assign the correct address in the Instruction Packet.

Access

Dynamixel has two kinds of data: Read-only data, which is mainly used for sensing, and Read-and-Write data, which is used for driving.

Initial Value

In case of data in the EEPROM Area, the initial values on the right side of the below Control Table are the factory default settings. In case of data in the RAM Area, the initial values on the right side of the above Control Tables are the ones when the power is turned on.

Highest/Lowest Byte

In the Control table, some data share the same name, but they are attached with (L) or (H) at the end of each name to distinguish the address. This data requires 16bit, but it is divided into 8bit each for the addresses (low) and (high). These two addresses should be written with one Instruction Packet at the same time.

Address Function Help

EEPROM Area

Model Number

It represents the Model Number.

Firmware Version

It represents the firmware version.

ID

It is a unique number to identify Dynamixel.

The range from 0 to 252 (0xFC) can be used, and, especially, 254(0xFE) is used as the Broadcast ID.

If the Broadcast ID is used to transmit Instruction Packet, we can command to all Dynamixels.

Please be careful not to duplicate the ID of connected Dynamixel.

Baud Rate

It represents the communication speed.

0: 9600, 1:57600, 2:115200, 3:1Mbps

Note : Maximum Baud Rate error of 3% is within the tolerance of UART communication.

Return Delay Time

It is the delay time per data value that takes from the transmission of Instruction Packet until the return of Status Packet.

0 to 254 (0xFE) can be used, and the delay time per data value is 2 usec.

That is to say, if the data value is 10, 20 usec is delayed. The initial value is 250 (0xFA) (i.e., 0.5 msec).

CW/CCW Angle Limit

The angle limit allows the motion to be restrained.

CONTROL MODE

Value	Return of Status Packet
1	Wheel Mode
2	Join Mode

The wheel mode can be used to wheel-type operation robots since motors of the robots spin infinitely.

The joint mode can be used to multi-joints robot since the robots can be controlled with specific angles.

The Highest Limit Temperature

Caution : Do not set the temperature lower/higher than the default value.

When the temperature alarm shutdown occurs, wait 20 minutes to cool the temperature before re-use.

Using the product when the temperature is high may and can cause damage.

The Lowest (Highest) Limit Voltage

It is the operation range of voltage.

50 to 250 (0x32 ~ 0x96) can be used. The unit is 0.1V.

For example, if the value is 80, it is 8V.

If Present Voltage (Address42) is out of the range, Voltage Range Error Bit (Bit0) of Status Packet is returned as '1' and Alarm is triggered as set in the addresses 17 and 18.

Max Torque

It is the torque value of maximum output. 0 to 1023 (0x3FF) can be used, and the unit is about 0.1%.

For example, Data 1023 (0x3FF) means that Dynamixel will use 100% of the maximum torque it can produce while Data 512 (0x200) means that Dynamixel will use 50% of the maximum torque. When the power is turned on, Torque Limit (Addresses 34 and 35) uses the value as the initial value.

Status Return Level

It decides how to return Status Packet. There are three ways like the below table.

Value	Return of Status Packet
0	No return against all commands (Except PING Command)
1	Return only for the READ command
2	Return for all commands

When Instruction Packet is Broadcast ID, Status Packet is not returned regardless of Status Return Level.

Alarm LED

Alarm Shutdown

Dynamixel can protect itself by detecting errors occur during the operation.

The errors can be set are as the table below.

it	Name	Contents
bit7	-	-
bit6	-	-
bit5	-	-
bit4	-	-
bit3	-	-
bit2	ERROR_INPUT_VOLTAGE	Voltage is out of operational voltage range
bit1	ERROR_OVER_HEATING	Temperature is out of operational temperature range
bit0	ERROR_OVERLOAD	Motor cannot output max load due to load being applied continuously

It is possible to make duplicate set since the function of each bit is run by the logic of 'OR'. That is, if 0X05 (binary 00000101) is set, both Input Voltage Error and Overheating Error can be detected.

If errors occur, in case of Alarm LED, the LED blinks; in case of Alarm Shutdown, the motor output becomes 0 % by making the value of Torque Limit(Address 34, 35) as 0.




RAM Area

Torque Enable

Value	Meaning
0	Keeps Torque from generating by interrupting the power of motor.
1	Generates Torque by impressing the power to the motor.

LED

Bit	명칭	내용
bit0+bit1+bit2	WHITE LED	 <p>When the bit is set the white LED turns on</p>
bit0+bit2	PINK LED	 <p>When the bit is set the pink LED turns on</p>
bit1+bit2	BLUE-GREEN LED	 <p>When the bit is set the blue-green LED</p>

		turns on
bit0+bit1	YELLOW LED	 When the bit is set the yellow LED turns on
bit2	BLUE LED	 When the bit is set the blue LED turns on
bit1	GREEN LED	 When the bit is set the green LED turns on

bit0	RED LED	 <p>When the bit is set the red LED turns on</p>
------	---------	---

PID Gain

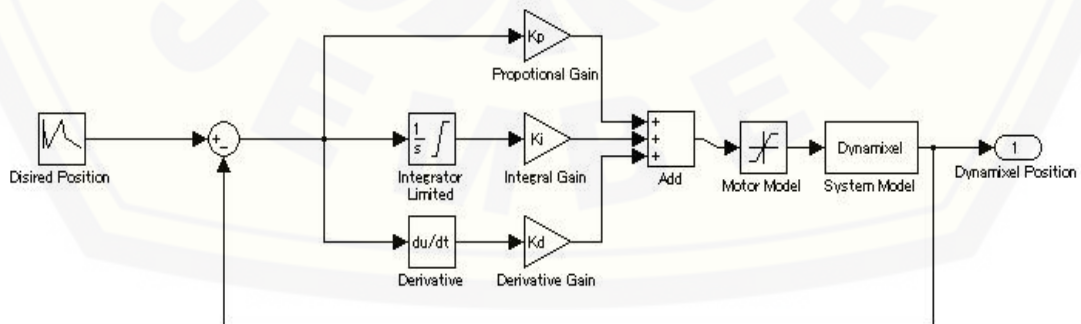
XL series will use the PID controller as a main control method.

P gain refers to the value of proportional band.

I gain refers to the value of integral action.

D Gain refers to the value of derivative action.

Gains values are in between 0~254.



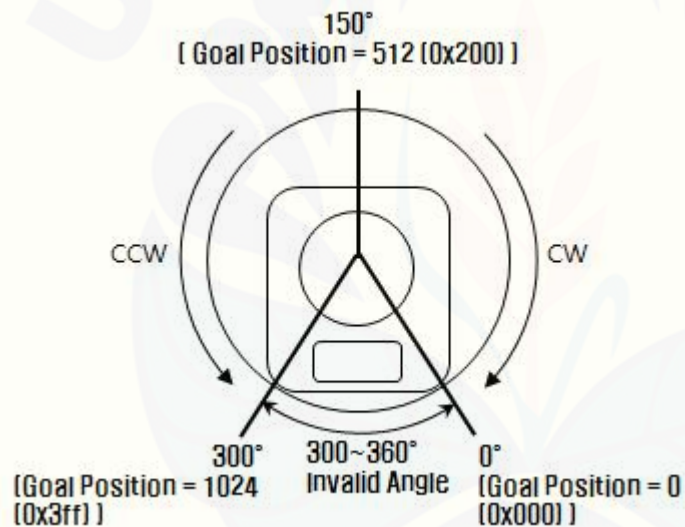
$$K_p = \text{P Gain} / 8$$
$$K_i = \text{I Gain} * 1000 / 2048$$
$$K_d = \text{D Gain} * 4 / 1000$$

Goal Position

It is a position value of destination.

0 to 1023 (0x3FF) is available. The unit is 0.29 degree.

If Goal Position is out of the range, Angle Limit Error Bit (Bit1) of Status Packet is returned as '1' and Alarm is triggered as set in Alarm LED/Shutdown.



<The picture above is based on the front of relevant model>

If it is set to Wheel Mode, this value is not used.

Moving Speed

It is a moving speed to Goal Position.

The range and the unit of the value may vary depending on the operation mode.

- **Join Mode**
0~1023 (0X3FF) can be used, and the unit is about 0.111rpm.
If it is set to 0, it means the maximum rpm of the motor is used without controlling the speed.
If it is 1023, it is about 114rpm.
For example, if it is set to 300, it is about 33.3 rpm.

Notes: Please check the maximum rpm of relevant model in Joint Mode.

Even if the motor is set to more than maximum rpm, it cannot generate the torque more than the maximum rpm.

- **Wheel Mode**
0~2047(0X7FF) can be used, the unit is about 0.1%.
If a value in the range of 0~1023 is used, it is stopped by setting to 0 while rotating to CCW direction.
If a value in the range of 1024~2047 is used, it is stopped by setting to 1024 while rotating to CW direction.
That is, the 10th bit becomes the direction bit to control the direction.
In Wheel Mode, only the output control is possible, not speed.
For example, if it is set to 512, it means the output is controlled by 50% of the maximum output.

Torque Limit

It is the value of the maximum torque limit.

0 to 1023 (0x3FF) is available, and the unit is about 0.1%.

For example, if the value is 512, it is about 50%; that means only 50% of the maximum torque will be used.

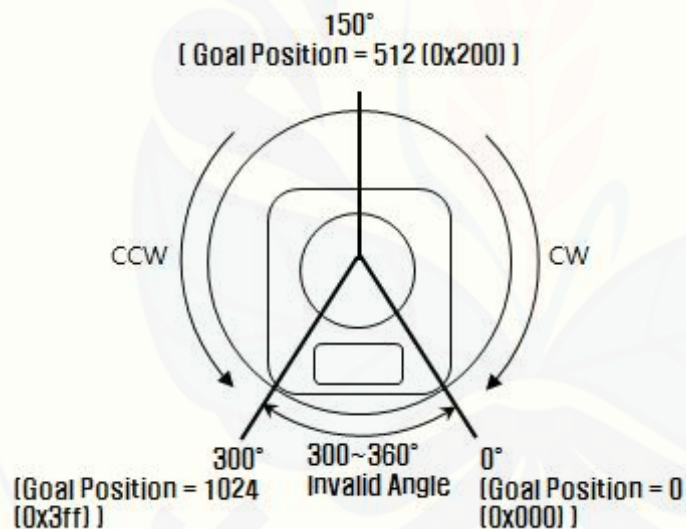
If the power is turned on, the value of Max Torque (Address 14, 15) is used as the initial value.

Notes: If the function of Alarm Shutdown is triggered, the motor loses its torque because the value becomes 0. At this moment, if the value is changed to the value other than 0, the motor can be used again.

Present Position

It is the current position value of Dynamixel.

The range of the value is 0~1023 (0x3FF), and the unit is 0.29 degree.



<The picture above is based on the front of relevant model>

Caution: If it is set to Wheel Mode, the value cannot be used to measure the moving distance and the rotation frequency.

Present Speed

It is the current moving speed.

0~2047 (0X7FF) can be used.

If a value is in the range of 0~1023, it means that the motor rotates to the CCW direction.

If a value is in the range of 1024~2047, it means that the motor rotates to the CW direction.

That is, the 10th bit becomes the direction bit to control the direction, and 0 and 1024 are equal.

The unit of this value varies depending on operation mode.

- **Joint Mode**
The unit is about 0.111rpm.
For example, if it is set to 300, it means that the motor is moving to the CCW direction at a rate of about 33.3rpm.
- **Wheel Mode**
The unit is about 0.1%.
For example, if it is set to 512, it means that the torque is controlled by 50% of the maximum torque to the CCW direction.

Present Load

It means currently applied load.

The range of the value is 0~2047, and the unit is about 0.1%.

If the value is 0~1023, it means the load works to the CCW direction.

If the value is 1024~2047, it means the load works to the CW direction.

That is, the 10th bit becomes the direction bit to control the direction, and 1024 is equal to 0.

For example, the value is 512, it means the load is detected in the direction of CCW about 50% of the maximum torque.

BIT	15~11	10	9	8	7	6	5	4	3	2	1	0
Value	0	Load Direction	Data (Load Ratio)									

Load Direction = 0 : CCW Load, Load Direction = 1: CW Load

Notes: Current load is inferred from the internal torque value, not from Torque sensor etc.

For that reason, it cannot be used to measure weight or torque; however, it must be used only to detect which direction the force works.

Present Voltage

It is the size of the current voltage supplied.

This value is 10 times larger than the actual voltage. For example, when 10V is supplied, the data value is 100 (0x64)

Present Temperature

It is the internal temperature of Dynamixel in Celsius.

Data value is identical to the actual temperature in Celsius. For example, if the data value is 85 (0x55), the current internal temperature is 85°C.

Registered Instruction

Value	Meaning
0	There are no commands transmitted by REG_WRITE

1	There are commands transmitted by REG_WRITE.
---	--

Notes: If ACTION command is executed, the value is changed into 0.

Moving

Value	Meaning
0	Goal position command execution is completed.
1	Goal position command execution is in progress.

HARDWARE ERROR STATUS

Current hardware error status

Alarm shutdown error value

Punch

Current to drive motor is at minimum.

Can choose values from 0x20 to 0x3FF.

Dimension

