SISTEM PEMBACAAN BOLA MENGGUNAKAN OMNIDIRECTIONAL CAMERA UNTUK PERGERAKAN ROBOT SEPAK BOLA BERODA



Disusun Oleh:

N a m a : Agus Darmawan
NIM : 14523122

PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2018

ii

HALAMAN PENGESAHAN DOSEN PEMBIMBING

SISTEM PEMBACAAN BOLA MENGGUNAKAN OMNIDIRECTIONAL CAMERA UNTUK PERGERAKAN ROBOT SEPAK BOLA BERODA

TUGAS AKHIR

Disusun Oleh:

N a m a : Agus Darmawan
NIM : 14523122

Yogyakarta, 16 Oktober 2018

Pembimbing,

(Izzati Muhimmah, S.T., M.Sc., Ph.D.)

iii

HALAMAN PENGESAHAN DOSEN PENGUJI

SISTEM PEMBACAAN BOLA MENGGUNAKAN OMNIDIRECTIONAL CAMERA UNTUK PERGERAKAN ROBOT SEPAK BOLA BERODA

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 16 Oktober 2018

Tim Penguji

Izzati Muhimmah, S.T., M.Sc., Ph.D. _____

Anggota 1

Elyza Gustri Wahyuni, S.T., M.Cs. _____

Anggota 2

Septia Rani, S.T., M.Cs. _____

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

iv

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Agus Darmawan

NIM : 14523122

Tugas akhir dengan judul:

SISTEM PEMBACAAN BOLA MENGGUNAKAN OMNIDIRECTIONAL CAMERA UNTUK PERGERAKAN ROBOT SEPAK BOLA BERODA

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 16 Oktober 2018



(Agus Darmawan)

HALAMAN PERSEMBAHAN

Dengan mengucap syukur Alhamdulillah, saya persembahkan karya ini yang diperuntunkan bagi semua saudara-saudara tercintaku, yang selalu bertanya :

"*Gimana skripsimu?*"

Tidak lupa saya persembahkan kepada kedua orang tua saya yang tercinta yang selama ini mendoakan yang terbaik bagi saya, saudara sekandung saya yang selalu mendukung saya, dan semua kerabat-kerabat saya.

HALAMAN MOTO

Jangan menjelaskan tentang dirimu kepada siapapun. Karena yang menyukaimu tidak membutuhkan itu, dan yang membencimu tidak akan mempercayai itu (Ali bin Abi Thalib).

Karena sesungguhnya sesudah kesulitan itu ada kemudahan(QS. Alam Nasyroh: 5).
Sesungguhnya sesudah kesulitan itu ada kemudahan(QS. Alam Nasyroh: 6).

KATA PENGANTAR

Alhamdulillah, penulis panjatkan segenap rasa syukur kehadirat Allah Subhanahu wa Ta'ala yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga laporan tugas akhir ini dapat diselesaikan. Shalawat dan salam selalu kita haturkan kepada junjungan kita Nabi Muhammad Shalallahu ‘alaihi Wassalam, dengan syafaat dari beliaulah kita dapat terbebas dari zaman kejahiliyahan.

Tugas akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Allah Subhanahu Wata’ala yang telah memberikan kehidupan, keselamatan dan kesehatan baik jasmani dan rohani.
2. Nabi Muhammad Shalallahu ‘alaihi Wassalam yang senantiasa menjadi panutan kita.
3. Orang tua dan keluarga penulis atas segala doa dan dukungan selama melaksanakan tugas akhir.
4. Keluarga besar “UNISI ROBOTIC” yang selama ini telah membantu dalam pengembangan robot yang menjadi objek penelitian.
5. Ibu Izzati Muhimmah selaku pembimbing tugas akhir ini, yang telah meluangkan waktunya untuk membimbing penulis dalam menyelesaikan penelitian ini.
6. Serta semua teman-teman yang telah membantu penulis, baik itu berupa doa maupun ilmu.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaman dalam melakukan penulisan laporan dan penelitian. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan tugas akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi banyak pihak.

Yogyakarta, 16 Oktober 2018

(Agus Darmawan)

SARI

Setiap tahunnya Indonesia rutin mengadakan Kontes Robot Indonesia oleh Ristekdikti, salah satu perlombaannya adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBIB) yang merupakan perlombaan robot yang bermain bola layaknya manusia namun robot yang digunakan merupakan robot yang bergerak atau bermanuver menggunakan roda dan melakukan permainan secara otomatis tanpa adanya campur tangan manusia dengan memanfaatkan sensor kamera sebagai alat visualisasi dan pencitraan bola.

Beberapa hal yang menjadi tantangan pada perlombaan ini diantaranya robot harus mampu mendeteksi bola, robot mampu melacak posisi bola, robot mampu mengenali garis-garis pada lapangan untuk dapat melakukan lokalisasi posisi terhadap lapangan, robot mampu melakukan lokalisasi target terhadap lapangan, dan robot mampu melakukan identifikasi penghalang.

Penelitian yang dilakukan kali ini berfokus pada pengolahan citra untuk medeteksi dan melacak keberadaan bola dan menggerakkan robot sepak bola beroda berdasarkan hasil penangkapan citra dengan teknik *omnidirectional camera*, selain itu akan digunakan teknik *color-filtering* yang diantaranya menggunakan metode segmentasi, morfologi, dan metode kontur untuk proses pengolahan citranya, selanjutnya metode sistem invers kinematik dan sistem *holonomic* sebagai teknik pergerakan robot, sehingga robot harus mampu mendeteksi objek bola yang ada di sekitar robot dan juga mampu bergerak ke arah letak atau posisi objek bola berada berdasarkan hasil citra tersebut.

Implementasi pada penelitian ini akan menghasilkan beberapa sistem yang mendukung selesainya penelitian, di antaranya sistem kalibrasi warna, sistem pengolahan citra, dan sistem pergerakan robot, di mana sistem-sistem tersebut kemudian akan dilakukan pengujian untuk mengetahui proses perancangan dan implementasi sistem pada penelitian ini, bekerja sesuai dengan semestinya.

Dari hasil implementasi tersebut, sistem mampu membaca bola dan melacak keberadaan bola yang berada didalam lapangan, dan sistem mampu menggerakan robot agar dapat bergerak ke arah posisi bola berada dengan tepat.

Kata kunci: KRSBIB, *color-filtering*, robot, bola.

GLOSARIUM

Real-time	merupakan suatu kondisi berjalannya suatu sistem yang memiliki teggat waktu dalam pengoperasiannya.
Humanoid	jenis suatu objek atau robot yang memiliki bentuk dan pergerakan yang menyerupai pergerakan manusia.
Omnidirectional	suatu sistem yang bergerak maupun menampilkan kondisi dari segala arah.
Mono	suatu sistem yang hanya bergerak dalam satu arah .
Camera	alat untuk menangkap citra.
Distorsi	suatu gangguan maupun perubahan bentuk yang tidak diinginkan.
Holonomic	sistem yang mampu memberikan pergerakan ke segala arah.
Package	paket yang menanmpung sejumlah library.
Trackbar	menu pilihan yang dapat digunakan dengan cara menggeserkan panel penggeser sehingga mendapatkan nilai tertentu.

DAFTAR ISI

HALAMAN JUDUL..... i
HALAMAN PENGESAHAN DOSEN PEMBIMBING..... iii
HALAMAN PENGESAHAN DOSEN PENGUJI..... **Error! Bookmark not defined.**
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**Error! Bookmark not defined.**
HALAMAN PERSEMBAHAN..... v
HALAMAN MOTO..... vi
KATA PENGANTAR..... vii
SARI..... viii
GLOSARIUM..... ix
DAFTAR ISI..... x
DAFTAR TABEL..... xiii
DAFTAR GAMBAR..... xiv
BAB I PENDAHULUAN..... 1
1.1 Latar Belakang..... 1
1.2 Rumusan Masalah..... 2
1.3 Batasan Masalah..... 3
1.4 Tujuan Penelitian..... 3
1.5 Manfaat Penelitian..... 3
1.6 Sistematika Penulisan Tugas Akhir..... 4
BAB II TINJAUAN PUSTAKA..... 5
2.1 Robot Sepak Bola Beroda..... 5
2.2 Pengenalan Objek Menggunakan OpenCV..... 6
2.3 Invers Kinematik untuk Pergerakan Robot..... 7
2.4 Software Terkait Perlombaan Robot Sepak Bola..... 8
BAB III LANDASAN TEORI..... 9
3.1 Perlombaan Robot Sepak Bola Beroda..... 9
3.2 Pengolahan Citra..... 10
3.2.1 Citra..... 10
3.2.2 Karakteristik Lingkungan..... 11
3.2.3 Watak *Omnidirectional Camera*..... 11
3.2.4 Warna..... 13

3.2.5 Warna Dasar HSV 14
3.3 Perangkat Keras dan Perangkat Lunak 14
3.3.1 Motor DC 14
3.3.2 Arduino 15
3.3.3 Komputer 16
3.3.4 Python ..... **Error! Bookmark not defined.**
3.3.5 OpenCV 16
3.3.6 Nanpy 17
3.4 Teknik Pengolahan Citra 17
3.4.1 Segmentasi 17
3.4.2 Morfologi 17
3.4.3 Kontur 18
3.5 Teknik Pergerakan Robot 19
3.5.1 Pergerakan *Holonomic* 19
3.5.2 Invers Kinematik 20
BAB IV METODOLOGI PENELITIAN 21
4.1 Analisis Masalah 21
4.2 Perancangan Sistem 22
4.3 Overview Sistem 24
4.3.1 Sistem *Omnidirectional Camera* 24
4.3.2 Sistem *Holonomic* 25
4.3.3 Sistem Multi-Threading 26
4.3.4 Sistem Kalibrasi Warna 26
4.3.5 Sistem *Color Filtering* 27
4.3.6 Sistem *Find-Contours* 27
4.3.7 Sistem Pembacaan Lapangan Hijau 28
4.3.8 Sistem *Tracking Ball* 29
4.3.9 Sistem Invers Kinematik 30
4.4 Metode Pengujian 30
BAB V HASIL DAN PEMBAHASAN 32
5.1 Implementasi Sistem 32
5.1.1 Instalasi Program Python 32
5.1.2 Instalasi *Library* OpenCV 33

5.1.3 Instalasi *Library* Nanpy....34
5.1.4 Implementasi Sistem Kalibrasi Warna....34
5.1.5 Implementasi Sistem Pengolahan Citra....37
5.1.6 Implementasi Sistem Pergerakan Robot....41
5.2 Menjalankan Sistem....44
5.3 Hasil dan Pengujian Sistem....46
5.4 Kelebihan Dari Sistem Hasil Penelitian....72
5.5 Kekurangan dari Sistem....72
BAB VI KESIMPULAN DAN SARAN....74
6.1 Kesimpulan....74
6.2 Saran....74
DAFTAR PUSTAKA....75
LAMPIRAN....77

DAFTAR TABEL

Tabel 5.1 Pengujian perubahan pencahayaan..... 47
Tabel 5.2 Pengujian perubahan pencahayaan..... 48
Tabel 5.3 Pengujian untuk menentukan jumlah iterasi pada operasi erosi dan dilasi..... 50
Tabel 5.4 Pengujian jarak pandang deteksi objek bola..... 55
Tabel 5.5 Pengujian hasil pengolahan citra untuk pergerakan robot..... 63

DAFTAR GAMBAR

Gambar 2.1 Robot sepak bola berbasis android.....5
Gambar 2.2 Definisi parameter untuk robot motor 4.....7
Gambar 3.1 Lapangan pertandingan robot sepak bola beroda.....9
Gambar 3.2 Koordinat spasial pada citra.....10
Gambar 3.3 Pengaruh pencahayaan lingkungan pada citra.....11
Gambar 3.4 Hasil Penangkapan dengan teknik *omnidirectional camera*.....12
Gambar 3.5 Hasil penangkapan citra dengan menggunakan kamera depan.....13
Gambar 3.6 Histogram pada citra.....13
Gambar 3.7 Perbedaan citra antara gabungan warna RGB dengan HSV.....14
Gambar 3.8 Prinsip kerja motor dc.....15
Gambar 3.9 Perangkat Arduino.....15
Gambar 3.10 Hasil operasi erosi dan delasi.....18
Gambar 3.11 Hasil pengambilan kontur dari lapangan hijau.....19
Gambar 3.12 Roda *omniwheel*.....20
Gambar 3.13 Representasi invers kinematik pada pergerakan motor tiga.....20
Gambar 4.1 Robot yang dijadikan objek penelitian.....21
Gambar 4.2 Berubahan bentuk bola menjadi lonjong (oval) pada citra.....22
Gambar 4.3 Rancangan sistem penelitian.....23
Gambar 4.4 Alur kerja sistem.....24
Gambar 4.5 Sistem *omnidirectional camera*.....25
Gambar 4.6 Sistem *holonomic* robot.....25
Gambar 4.7 Sistem kalibrasi warna.....26
Gambar 4.8 Hasil *color filtering*.....27
Gambar 4.9 Hasil *find-contours* dari citra.....28
Gambar 4.10 Contoh penggunaan operasi *opening* dan *closing*.....29
Gambar 4.11 Hasil pembatasan pembacaan lapangan hijau.....29
Gambar 4.12 Tangkapan gambar dari robot mengikuti bola.....30
Gambar 4.13 Rumus invers kinematik.....30
Gambar 5.1 Sintak pemasangan program Python.....32
Gambar 5.2 Program Python.....33
Gambar 5.3 Sintak pemasangan program pip.....33

Gambar 5.4 Sintak pemasangan *library* opencv..... 34
Gambar 5.5 Pemeriksaan versi OpenCV..... 34
Gambar 5.6 Sintak pemasangan *library* nanpy.....34
Gambar 5.7 Jendela kalibrasi warna untuk menentukan rentang nilai warna objek.....35
Gambar 5.8 Kalibrasi warna untuk warna bola..... 36
Gambar 5.9 Kalibrasi warna untuk warna lapangan.....36
Gambar 5.10 Sintak variabel penyimpan nilai masing-masing warna..... 37
Gambar 5.11 Alur kerja sistem kalibrasi warna.....37
Gambar 5.12 Sintak pengambilan nilai masing-masing warna yang disimpan..... 38
Gambar 5.13 Pembatasan pembacaan untuk lapangan hijau dan *tracking ball* di dalam lapangan hijau.....39
Gambar 5.14 Alur sistem pengolahan citra.....41
Gambar 5.15 Sintak pengaksesan arduino menggunkan *library* nanpy..... 43
Gambar 5.16 Alur sistem pergerakan robot.....43
Gambar 5.17 Sintak untuk algoritma pergerakan motor.....44
Gambar 5.18 Tiga jendela *trackbar* untuk menentukan nilai warna dari objek target.. 45
Gambar 5.19 tiga jendela sistem pengolahan citra..... 46

BAB I
PENDAHULUAN

1.1 Latar Belakang

Robot merupakan sebuah mesin atau salah satu perangkat teknologi yang dikembangkan oleh manusia dengan tujuan dapat melakukan berbagai macam pekerjaan manusia dan mempermudah manusia untuk menyelesaikan segala kegiatannya, pemanfaatan tenaga robot untuk membantu pekerjaan manusia sangat banyak, mulai dari pekerjaan yang tidak berbahaya contohnya seperti robot pelayan rumah makan, robot layanan informasi dan bahkan ada juga robot yang digunakan untuk menjaga bayi, sampai dengan pekerjaan yang notabene memiliki tingkat resiko kecelakan yang lumayan tinggi, contohnya seperti pekerjaan dibidang industri otomotif (industri pembuatan mobil, industri pembuatan pesawat, dan lain-lain) maupun bidang industri kimia.

Indonesia merupakan salah satu negara berkembang yang sudah beralih menggunakan robot sebagai salah satu mesin yang membantu dalam mengerjakan berbagai macam kegiatan. Selain itu Indonesia juga merupakan salah satu negara yang ikut serta dalam perlombaan robot tingkat internasional untuk bertanding melawan robot-robot dari negara lain. Di Indonesia sendiri, setiap tahunnya rutin mengadakan Kontes Robot Indonesia untuk mencari kandidat-kandidat yang akan mewakili Indonesia dalam perlombaan tingkat internasional. Perlombaan ini merupakan perlombaan yang diadakan oleh Kementerian Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia (Ristekdikti) yang diikuti oleh beberapa perguruan-perguruan tinggi di Indonesia.

Salah satu perlombaan robot yang diadakan oleh (Ristekdikti) adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBIB), di mana perlombaan ini merupakan perlombaan robot yang bermain bola layaknya manusia namun robot yang digunakan merupakan robot yang bergerak atau ber-manuver menggunakan roda dan melakukan permainan secara otomatis tanpa adanya campur tangan manusia dengan memanfaatkan sensor kamera sebagai alat visualisasi dan pencitraan bola. Sensor ini layaknya mata sehingga robot mampu melihat posisi bola berada di dalam lapangan.

Pada perlombaan KRSBIB yang diadakan oleh Ristekdikti ini ada beberapa tantangan yang perlu diperhatikan, di antaranya robot harus mampu mendeteksi bola, robot mampu melacak posisi bola, robot mampu mengenali garis-garis pada lapangan untuk dapat

melakukan lokalisasi posisi terhadap lapangan, robot mampu melakukan lokalisasi target terhadap lapangan, dan robot mampu melakukan identifikasi penghalang.

Penelitian sebelumnya yang dilakukan oleh Suryawan dengan judul skripsi "Perancangan Robot Sepak Bola Beroda Berbasis Android", pada penelitian yang dilakukan oleh suryawan ini menggunakan sebuah smartphone berbasis android sebagai mesin komputasinya, versi android yang digunakan sendiri merupakan versi sistem operasi 1.6 Donut, dengan total penyimpanan internal sebesar 128MB. Selain itu chipset yang digunakan merupakan chipset Qualcomm MSM7227 600MHz ARM 11 dengan GPU Adreno 200. Sedangkan untuk penangkapan citra, dilakukan dengan pengambilan citra secara landscape dan menghadap langsung ke arah bola atau kearah depan robot dan menghasilkan citra yang hanya menampilkan kondisi didepan robot (Suryawan, 2014).

Sedangkan robot yang akan digunakan untuk penelitian ini, menggunakan sebuah komputer bersistem operasi Linux (distro Ubuntu) yang digunakan untuk melakukan pemprosesan gambar yang ditangkap oleh sebuah kamera. Spesifikasi komputer yang digunakan berupa processor "Intel Core i3", dengan ram 4GB dan memori penyimpanan sebesar 128GB. Sedangankan untuk kamera yang digunakan memiliki resolusi 1920 x 1080 piksel. Untuk penangkapan citra sendiri dilakukan dengan teknik *omnidirectional camera* yang memanfaatkan cermin cembung dengan cara kamera dihadapkan ke arah atas dan cermin cembung dihadapkan ke arah bawah, sehingga dapat menghasilkan citra yang menampilkan keadaan di sekitar robot dari segala arah.

Maka dari itu, pada penelitian kali ini berfokuskan pada pendeteksian dan pelacakan bola, yaitu melakukan pengolahan citra untuk medeteksi dan melacak bola sebagai parameter pergerakan robot sepak bola beroda berdasarkan hasil penangkapan citra *omnidirectional camera*, sistem diharuskan mampu mendeteksi objek bola yang ada di sekitar robot dan juga mampu bergerak ke arah letak atau posisi objek bola berada berdasarkan hasil citra tersebut.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang dimunculkan di atas, rumusan masalah berikut ini akan dijawab oleh penelitian yang akan dilakukan, di antaranya :

1. Bagaimana cara robot mendeteksi objek bola berdasarkan hasil citra dari *omnidirectional camera*?
2. Bagaimana cara robot melacak keberadaan bola berdasarkan hasil citra dari *omnidirectional camera*?

3. Bagaimana robot dapat bergerak ke arah objek bola secara otomatis?

1.3 Batasan Masalah

Untuk menjaga fokus penelitian ini, ada beberapa batasan masalah yang perlu diperhatikan, hal ini bertujuan agar penelitian yang dilakukan tidak menyimpang dari pokok pembahasan yang diangkat, maka berikut batasan masalah dari penelitian ini :

1. Deteksi objek bola dilakukan berdasarkan cakupan tangkap citra di sekitar robot yang dihasilkan dari *omnidirectional camera*.
2. Wilayah yang digunakan untuk deteksi objek bola merupakan wilayah berwarna hijau yang sesuai dengan lapangan sepak bola pada perlombaan robot sepak bola beroda.
3. Invers kinematik merupakan persamaan matematika untuk mendapatkan pergerakan motor, dalam penelitian ini tidak membahas invers kinematik lebih lanjut.
4. Untuk proses kalibrasi warna, akan dilakukan kalibrasi terhadap warna dari setiap bola yang memiliki warna yang berbeda dengan warna bola yang sebelumnya.

1.4 Tujuan Penelitian

Tujuan dari penelitian yag akan dilakukan :

1. Robot mampu mendeteksi objek bola yang berada di sekitar robot berdasarkan hasil tangkapan citra *omnidirectional camera*.
2. Robot mampu melacak keberadaan bola berdasarkan hasil citra dari *omnidirectional camera*.
3. Robot mampu bergerak ke arah objek bola berdasarkan letak dan posisi objek bola secara otomatis.

1.5 Manfaat Penelitian

Manfaat dari penelitian yang dilakukan ini, robot mampu bergerak kearah letak objek bola secara otomatis, tanpa harus menggerakan robot secara manual, sehingga robot bisa bermain layaknya pemain sepak bola yang sesungguhnya.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan laporan berguna untuk mempermudah pembaca dalam memahami laporan tugas akhir ini. Secara garis besar, sistematika penulisan laporan tugas akhir ini dapat diuraikan sebagai berikut :

BAB I PENDAHULUAN

Pada bagian bab ini akan membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan dari penelitian yang dilakukan, manfaat dari penelitian yang dilakukan dan sistematika penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab tinjauan pustaka ini akan memaparkan penelitian-penelitian yang sebelumnya telah dilakukan dan berhubungan dengan permasalahan yang diangkat, dan mengacu pada daftar pustaka.

BAB III LANDASAN TEORI

Pada bab landasan teori ini akan membahas mengenai teori dan metode-metode yang berhubungan dengan penelitian yang dilakukan, yang kemudian akan digunakan untuk dapat memecahkan permasalahan yang dihadapi dalam melakukan penelitian.

BAB IV METODOLOGI

Bab ini memaparkan analisis masalah, overview sistem, perancangan sistem, implementasi sistem, dan bentuk pengujian sistem.

BAB V HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan ini akan dibahas mengenai sistem yang dibuat berdasarkan hasil dari implementasi sistem.

BAB VI KESIMPULAN DAN SARAN

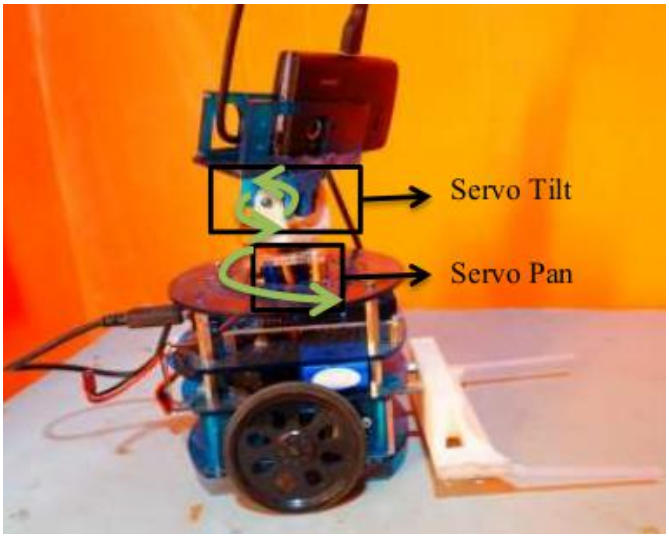
Pada bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan, serta saran-saran yang dapat diterapkan dari hasil implementasi sistem yang dapat menjadi masukan yang berguna untuk kedepannya.

BAB II
TINJAUAN PUSTAKA

Penelitian sejenis yang sebelumnya telah dilakukan, akan dijadikan sebagai bahan kajian untuk pembahasan saat ini, hal ini bertujuan untuk menunjukkan bahwa penelitian yang dilakukan saat ini memiliki konstribusi tersendiri terhadap ilmu pengetahuan dan juga akan menghindari dari proses duplikasi yang kemungkinan dapat terjadi.

2.1 Robot Sepak Bola Beroda

Penelitian yang dilakukan oleh (Suryawan, 2014) akan menjadi salah satu studi literatur dari penelitian ini, penelitian suryawan memiliki fokus utama yang hampir mirip dengan penelitian saat ini, yaitu berfokus pada robot sepak bola beroda. Pada penelitian yang dilakukannya ini, pengolahan citra dan proses-proses untuk mendapatkan koordinat dari objek menggunakan smartphone android sebagai pusat komputasi, untuk teknik pengambilan citranya digunakan teknik landscape yang langsung dihadapkan ke arah depan robot, yang menghasilkan citra satu arah tanpa mampu menampilkan kondisi yang ada di arah yang berlawanan seperti yang terlihat pada Gambar 2.1. Untuk deteksi objek yang digunakan ialah deteksi objek berdasarkan warna, dan memanfaatkan operasi erosi dan dilasi agar deteksi objek berfokus pada warna objek yang menjadi target deteksi.



Gambar 2.1 Robot sepak bola berbasis android
(sumber : Suryawan, 2014)

2.2 Pengenalan Objek Menggunakan OpenCV

Pengenalan objek yang diterapkan pada robot sepak bola beroda ini amatlah sangat penting, menilik pada navigasi robot untuk bergerak sesuai dengan letak dan posisi bola berada, dan menyesuaikan dengan hasil tangkapan citra dari *omnidirectional camera*.

Tinjauan pustaka yang didapatkan mengenai deteksi objek untuk pengolahan citra yang menjadi bahan rujukan, salah satunya deteksi objek yang menggunakan "Algoritma Greedy untuk Mendeteksi Bola pada Robot Sepak Bola Beroda" (Asyrofi, 2017). Selain itu juga, terdapat penelitian mengenai pengenalan objek untuk dijadikan rujukan, yang dilakukan oleh Mahardika "Identifikasi Basic Shape dengan Metode Harris Corner Detection pada Raspberry PI" (Mahardika, 2015) yang membedakan jenis objek berdasarkan bentuk dari citra objek yang dihasilkan, dan "Aplikasi Penghitung Telur Ikan Gurami menggunakan Deteksi Blob berbasis Android" oleh (Setiawan dan Firdausy, 2016).

Penelitian yang dilakukan oleh (Asyrofi, 2017), untuk dapat mendeteksi objek bola pada robot sepak bola beroda menggunakan beberapa teknik, dintaranya teknik gaussian blur, teknik canny edge, teknik hough circle transform, dan teknik convex hull. Penggunaan teknik gaussian blur pada penelitian Asyrofi ini dimanfaatkan untuk mendapatkan citra dengan distribusi warna yang rata dengan tujuan memperbaiki struktur gambar pada skala tertentu, teknik canny edge detector digunakan untuk mendeteksi atau mengambil garis tepian objek dan juga mengurangi data yang akan di olah, teknik hough circle transform digunakan untuk menganalisis objek dengan bentuk tertentu pada citra.

Sedangkan untuk penelitian yang dilakukan (Mahardika, 2015) menggunakan teknik harris corner detection untuk melakukan pengenalan terhadap objek berdasarkan sudut yang dihasilkan dari citra tiap objek. Selain itu juga, sebelum masuk teknik harris corner, Mahardika menggunakan beberapa teknik untuk pengolahan citranya, diantaranya teknik grayscale yang mengubah citra berwarna menjadi citra keabuan, di mana pada dasarnya citra berwarna memiliki nilai intensitas yang berupa warna hitam, keabuan dan putih. Selanjutnya teknik gaussian blur, thresholding dan canny edge detector.

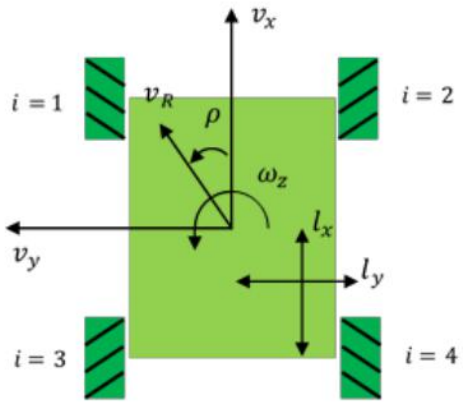
Pada penelitian yang dilakukan oleh (Setiawan dan Firdausy, 2015) menggunakan teknik segmentasi, teknik thresholding, dan teknik deteksi blob. Teknik segmentasi merupakan teknik yang melakukan pemisahan atau membedakan antara objek yang akan di deteksi dengan latar belakang dari objek pada citra, teknik ini akan sangat berpengaruh terhadap tingkat keberhasilan terhadap pengenalan pola yang akan dilakukan, teknik thresholding disini digunakan untuk mengubah dari citra sebelumnya (dari citra asli ke citra

keabuan) menjadi citra dengan nilai 1 dan 0 atau citra biner. Dan penggunaan teknik blob berguna untuk mendeteksi objek berdasarkan perbedaan warna pada piksel-piksel yang muncul (lebih gelap atau lebih terang), biasanya teknik ini menggunakan operasi dilasi untuk pendeteksian objeknya dari citra biner.

2.3 Invers Kinematik untuk Pergerakan Robot

Sedangkan untuk teknik pergerakan motor menggunakan persamaan invers kinematik dalam melakukan konversi dari koordinat objek pada citra menjadi pergerakan bagi motor, beberapa penelitian yang dilakukan di antaranya "Implementasi Sistem Gerak *Holonomic* Pada Robot KRSBI Beroda 2017" (Muliady dan Arisandy, 2017) dan juga penelitian "Implementasi Invers Kinematics Pada Sistem Pergerakan Mobile Robot Roda Mekanum" oleh (Hendrayawan, 2014)

Pada penelitian yang dilakukan oleh (Muliady dan Arisandy, 2017) dan (hendrayawan, 2014) ini berhasil melakukan implementasi invers kinematik pada robot beroda, penelitian yang mereka lakukan sama-sama menggunakan 4 buah motor untuk perpindahan robot dan juga sama-sama menggunakan roda mekanum yang mampu bergerak ke segala arah dan termasuk kedalam jenis pergerakan sistem *holonomic*, di mana pada perputaran motor yang dihasilkan menggunakan invers kinematik terhadap robot roda empat memiliki tingkat 40 – 100 % untuk mendekati bola yang ada di depan robot, namun keberhasilan yang didapatkan 0 – 20 % untuk mendekati bola yang ada di belakang robot, hal ini tidak luput dipengaruhi oleh jaraknya antara robot dengan bola (Muliady dan Arisandy, 2017).



Gambar 2.2 Definisi parameter untuk robot motor 4

(sumber : Muliady dan Arisandy, 2017)

Sedangkan penelitian yang dilakukan oleh (hendrayawan, 2014), pergerakan roda dengan menggunakan persamaan invers kinematik pada roda mekanum dari beberapa sudut pengujian menghasilkan rata-rata kesalahan kecepatan putaran sebesar 3,16 rpm dari masing-masing motor, dengan motor 1 sebesar 3,38 rpm, motor 2 sebesar 3,6 rpm, motor 3 sebesar 4,25 rpm dan motor 4 sebesar 3,18 rpm.

2.4 Software Terkait Perlombaan Robot Sepak Bola

Perlombaan robot sepak bola umumnya diikuti oleh berbagai universitas diseluruh Indonesia, hal ini menunjukkan perbedaan di antara robot-robot yang diajukan dari masing-masing universitas, baik itu desain sampai dengan perangkat lunak pendukung untuk memprogram robot yang akan diperlombakan.

Penelitian oleh (Suryawan, 2014) menggunakan perangkat lunak arduino IDE sebagai pengirim program kontrol ke dalam mikro kontroler sebagai alat kendali pergerakan robot, selain itu perangkat lunak java digunakan untuk memprogram aplikasi yang akan mengelola citra hasil tangkap kamera pada smartphone android.

Penelitian yang dilakukan oleh (Asyrofi, 2017) menggunakan dasar pemprograman C++ untuk menggunakan *library* OpenCV dan melakukan pengolahan terhadap citra, melihat bahwa struktur dasar dari *library* OpenCV sendiri merupakan bahasa pemprograman C++.

Selain itu, penelitian yang dilakukan oleh (Fajar, 2017) menggunakan perangkat lunak Python sebagai bahasa pemprogramannya untuk pengolahan citra, selain itu pemilihan bahasa pemprograman Python juga dikarenakan bahasa pemprograman Python ini sudah terdapat suatu modul yang berguna untuk membuat suatu komunikasi antara satu robot dengan robot yang lainnya, modul komunikasi tersebut adalah modul socket.

BAB III
LANDASAN TEORI

3.1 Perlombaan Robot Sepak Bola Beroda

Perlombaan robot sepak bola yang diadakan oleh Ristekdikti, memiliki beberapa peraturan yang telah menjadi kesepakatan bersama, baik itu ukuran lapangan, posisi bola, posisi robot maupun aturan permainannya. Untuk penggunaan lapangan sendiri, dipilih warna hijau sebagai dasar permukaan lapangan, hal ini menyesuaikan dengan lapangan yang biasa digunakan untuk bermain sepak bola. Lapangan yang digunakan pada pertandingan robot sepak bola beroda sendiri memiliki ukuran yang berbeda dengan sepak bola untuk manusia, di mana luas lapangan untuk sepak bola manusia memiliki ukuran 105 x 68 m, sedangkan luas dari lapangan yang digunakan untuk pertandingan robot sepak bola beroda memiliki ukuran 6 x 9 m, seperti yang terlihat pada Gambar 3.1 berikut ini.



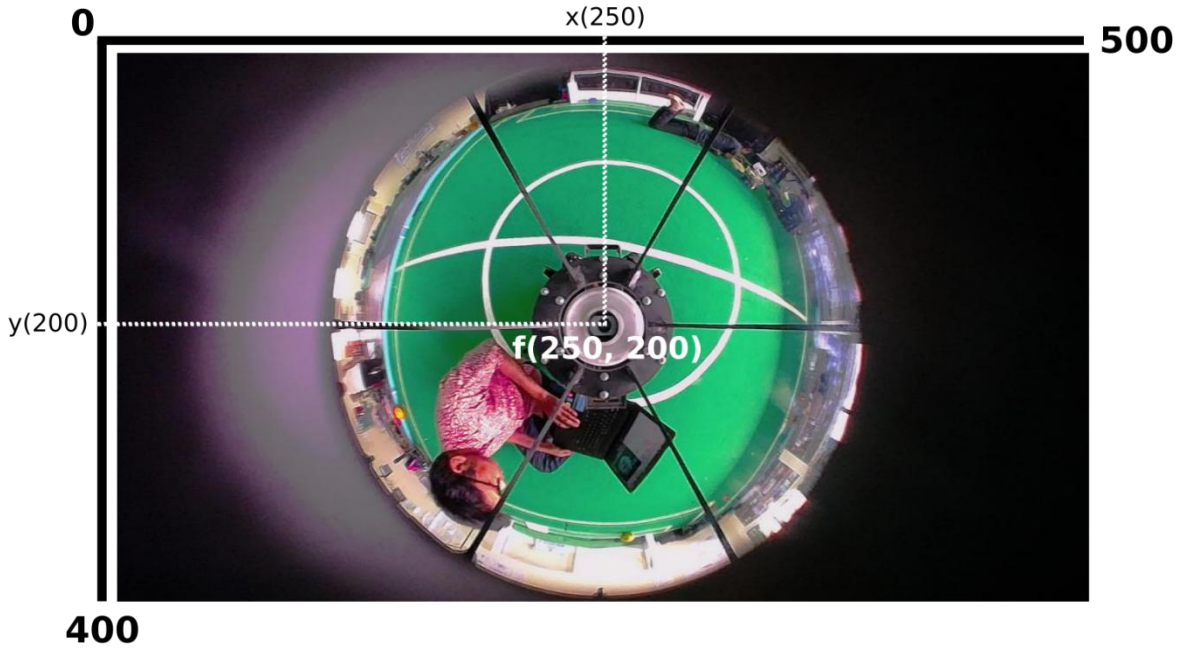
Gambar 3.1 Lapangan pertandingan robot sepak bola beroda

Untuk posisi bola, sebelum pertama kali permainan dimula, bola akan diletakkan oleh juri ke posisi tengah lapangan, dengan begitu permainan baru akan dapat dimulai, namun lain halnya dengan beberapa pelanggaran yang dilakukan atau diakibatkan oleh pemain, maka posisi bola akan menyesuaikan sesuai dengan peraturan yang telah ditentukan. Sedangkan untuk posisi awal pemain, boleh mengambil posisi di mana saja (otomatis) namun dengan satu syarat yaitu masih mencakup wilayah lapangan sendiri dan robot memiliki jarak tertentu terhadap bola.

3.2 Pengolahan Citra

3.2.1 Citra

Citra atau bisa juga disebut dengan gambar merupakan salah satu media informasi yang memiliki struktur penyusun atas kombinasi warna-warna dalam tiap piksel-nya dan berupa gambar dua dimensi. Citra dibedakan menjadi dua jenis, yaitu citra analog dan citra digital, namun untuk melakukan pengolahan citra akan digunakan citra digital yang merupakan sebuah gambar yang menampilkan susunan 3 kombinasi warna dalam tiap pikselnya, di antaranya warna merah, hijau dan biru (red, green, dan blue – RGB) sampai membentuk satu citra dan tersusun satu per-satu dalam sebuah frame atau array yang dapat di definisikan sebagai fungsi f (x dan y), dengan nilai x dan y masing-masing merupakan koordinat spasial serta nilai fungsi pada (x dan y) intensitas suatu citra (Mahardika, 2015).

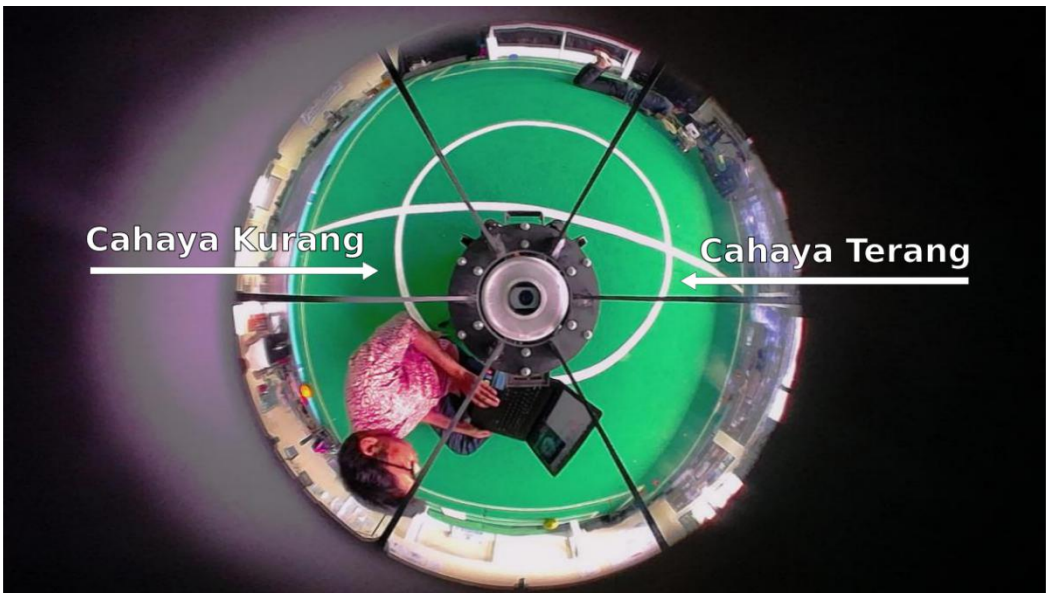


Gambar 3.2 Koordinat spasial pada citra

Alasan digunakannya citra digital dikarenakan dapat diakses atau digunakan melalui komputer, dan dapat diolah secara langsung maupun disimpan kedalam memori, pada pengolahan citra, koordinat nilai 0 untuk sumbu x dan y berada pada posisi atas kiri citra, hal ini dapat dilihat seperti yang ditampilkan pada Gambar 3.2.

3.2.2 Karakteristik Lingkungan

kondisi lingkungan pada pengambilan citra menggunakan kamera sangat mempengaruhi hasil dari pengolahan citra. Hal-hal dasar yang akan mempengaruhi citra, salah satunya yaitu keadaan pencahayaan lingkungan yang masuk ke dalam citra yang ditangkap, pencahayaan yang berbeda akan sangat mempengaruhi warna yang ditangkap oleh citra tersebut. Perbedaan warna inilah yang akan mempersulit kita untuk melakukan penyaringan berdasarkan warna dari objek.



Gambar 3.3 Pengaruh pencahayaan lingkungan pada citra

Pada Gambar 3.3 di atas terlihat bahwa posisi lapangan disisi kiri dan disisi kanan robot memiliki warna hijau yang cukup berbeda, salah satu sisi berwarna hijau tua dan satu sisi lainnya memiliki warna hijau muda. Untuk hal penyaringan warna pada kasus seperti ini masih bisa di sesuaikan rentang nilai warnanya, namun bagaimana halnya jika pencahayaan disisi kiri lebih terang lagi, maka akan sangat mungkin warna hijau akan berubah menjadi warna putih.

3.2.3 Watak *Omnidirectional Camera*

Omnidirectional camera merupakan konsep pengambilan gambar atau citra yang menggunakan kamera dan menghasilkan gambar yang dapat menampilkan citra dari berbagai arah, baik itu arah depan, arah belakang, samping kanan maupun samping kiri. Pada penelitian ini, teknik pengambilan citra dengan *omnidirectional camera* memanfaatkan sebuah cermin cembung yang dihadapkan ke bawah dan kamera yang dihadap ke atas (kearah

cermin cembung), sehingga citra yang akan ditangkap oleh kamera merupakan pantulan cermin cembung yang mampu memperlihatkan keadaan atau kondisi disekitar robot dan dari segala arah seperti yang terlihat pada Gambar 3.3.

Seperti yang terlihat pada Gambar 3.2. di atas, citra yang dihasilkan oleh *omnidirectional camera* sendiri memiliki *distorsi* pada segi perseptif sudut pandang, di mana citra yang dihasilkan memiliki pandangan yang cekung (tidak datar atau tidak standar), hal ini sangat mempengaruhi proses deteksi objek berdasarkan bentuk, dengan menggunakan teknik *omnidirectional camera* objek yang akan terlihat pada pantulan cermin akan memiliki bentuk yang berbeda dengan bentuk aslinya, hal ini dapat dilihat pada Gambar 3.4 berikut ini, terlihat bahwa objek bola memiliki bentuk yang sudah tidak sama seperti bentuk aslinya, yaitu objek bola menjadi lebih melonjong (oval).



Gambar 3.4 Hasil Penangkapan dengan teknik *omnidirectional camera*

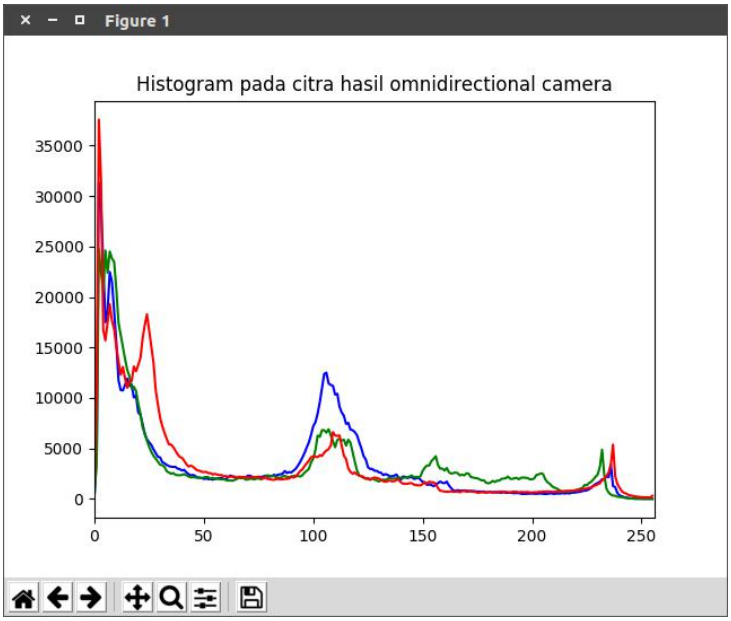
Namun dibalik kekurangannya tersebut, *omnidirectional camera* memiliki kelebihan, yaitu dapat menampilkan citra dari segala arah hanya dengan satu kali pengambilan citra, berbeda dengan teknik *mono camera,* dengan cara kamera yang digunakan diarahkan langsung terhadap objek atau diarahkan kedepan, untuk dapat menghasilkan citra dari segala arah seperti *omnidirectional camera* membutuhkan 6 kali pengambilan citra, hal ini dapat dibandingkan antara Gambar 3.4 dan Gambar 3.5.



Gambar 3.5 Hasil penangkapan citra dengan menggunakan kamera depan

3.2.4 Warna

Warna merupakan sebuah pantulan cahaya yang dapat kita lihat ketika cahaya memantul atau mengenai suatu benda yang memiliki spektrum tertentu dengan panjang gelombang masing-masing berdasarkan warna (Wikipedia, 2018). Pada citra digital, warna diterapkan kedalam bentuk histogram berdasarkan jumlah munculnya nilai frekuensi intensitas tiap piksel yang merupakan kombinasi antara warna merah (red), hijau (green), dan biru (blue).



Gambar 3.6 Histogram pada citra

pada Gambar 3.6 di atas, terlihat hasil histogram dari citra Gambar 3.2, menampilkan hasil dari tingkatan nilai intensitas tiap piksel berdasarkan susunan warna RGB.

3.2.5 Warna Dasar HSV

Pada dasarnya citra tersusun dari gabungan warna antara merah, hijau, dan biru (RGB). Namun pada penelitian ini, pencarian untuk nilai warna dari objek akan menggunakan gabungan warna antara Hue, Saturation, dan Value (HSV), hal ini dikarenakan penggunaan gabungan warna HSV memiliki keunggulan pada toleransi perubahan jarak warna yang dipengaruhi oleh karakteristik lingkungan berupa intensitas cahaya disekitar robot (Mahesa, Azizul, dan Meliyana. 2016).



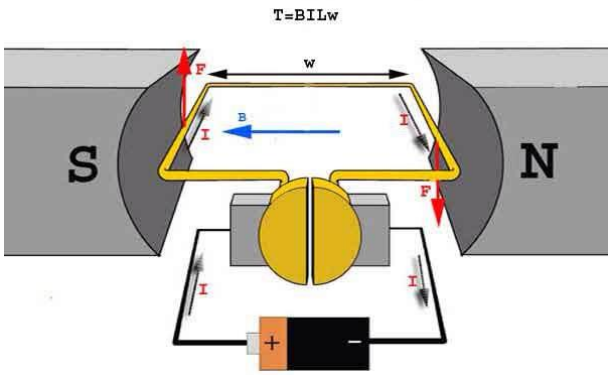
Gambar 3.7 Perbedaan citra antara gabungan warna RGB dengan HSV

Pada gambar di atas terlihat bahwa perbedaan penangkapan citra dari gabungan warna RGB dengan HSV, dapat dilihat bahwa citra pada gabungan warna HSV memiliki rentang nilai warna yang berbeda pada citra dengan intensitas cahaya yang berbeda, hal ini memberikan pilihan lebih banyak untuk menentukan rentang nilai terendah dan rentang nilai tertinggi dari warna objek yang akan menjadi target pengolahan citra.

3.3 Perangkat Keras dan Perangkat Lunak

3.3.1 Motor DC

Motor DC (Direct Curent) merupakan jenis motor listrik yang menggunakan catu daya tegangan searah pada kumparan medan yang kemudian akan diubah menjadi energi gerak atau energi kinetik. Terdapat dua bagian utama untuk pergerakan pada motor dc, di antaranya kumparan medan yang ada pada motor dc yang disebut dengan “stator” (bagian yang tidak berputar), dan kumparan jangkar yang disebut dengan “rotor“ (bagian yang berputar) (Setiaji, 2014).



Gambar 3.8 Prinsip kerja motor DC

(sumber : Setiaji, 2014)

prinsip kerja motor dc sendiri hanya memanfaatkan tarikan medan magnet dari kumparan medan yang memiliki sifat medan magnet utara dan selatan, untuk pergerakannya kumparan jangkar akan dialiri listrik sehingga mengahasilkan medan magnet yang mengakibatkan tarik-menarik antara masing-masing kumparan (onny, 2014).

3.3.2 Arduino

Arduino adalah perangkat mikro kontroler yang bersifat open source dan memberikan kemudahan dalam penggunaannya bagi siapa. Arduino dapat digunakan untuk mengakses atau membaca berbagai macam sensor, mulai dari sensor suara, jarak, gerakan, sampai dengan mengakses lampu (LED). Selain itu, arduino juga memberikan kemudahan bagi pengguna dalam memasukkan program-program yang diinginkan ke dalam mikro kontroler arduino, di mana tersedianya Arduino IDE, yang merupakan suatu perangkat lunak dan dapat digunakan untuk memprogram dan memasukkan program-program ke dalam mikro kontroler arduino (Arduino, 2018).



Gambar 3.9 Perangkat Arduino

(sumber : Arduino, 2018)

3.3.3 Komputer

Komputer merupakan mesin perhitungan yang diciptakan khusus oleh manusia untuk membantu pekerjaan manusia dalam melakukan berbagai hal yang berhubungan dengan kegiatan komputasi. Salah satu kegiatan komputasi yang dapat dikerjakan oleh komputer dengan cepat adalah proses pengolahan citra secara *real-time*. Perkembangan komputer yang sangat cepat menjadikan komputer sangat berpengaruh terhadap kehidupan manusia, selain itu perkembangan kecerdasan buatan membuat komputer menjadi bagian utama atau hal terpenting untuk melakukan komputasinya, baik itu pengenalan objek sampai dengan proses pengenalan wajah.

3.3.4 Python

Python merupakan salah satu bahasa pemprograman komputer yang pembuatannya di fokuskan pada tingkat keterbacaan kode program. Bahasa pemprograman Python merupakan bahasa yang menggabungkan kapabilitas, kemampuan, sintaks program yang jelas, dan didasari dengan *library* standar bawaan yang sangat banyak. Penggunaan bahasa pemprograman ini tidak dibatasi dengan paradigma yang semestinya, namun dengan menggunakan bahasa pemprograman ini para pengembang (developers) dapat menentukan struktur aplikasi yang akan dibuat, tidak mesti berorientasi objek maupun imperatif.

3.3.5 OpenCV

OpenCV (Open Source Computer Vision) merupakan salah satu pustaka perangkat lunak (*library*) yang dikembangkan khusus untuk digunakan dalam pengolahan citra baik itu citra berupa gambar maupun video dan juga lebih ditujukan atau difokuskan pada proses pengolahan citra secara *real-time* (OpenCV team, 2018). *Library* ini merupakan *library* open source, yang berarti *library* ini bersifat bebas untuk dikembangkan oleh setiap pengguna tanpa adanya koordinasi dari individu atau perusahaan tertentu dan juga source-codenya tersedia untuk setiap orang yang ingin mengembangkannya.

Selain itu *library* yang berbasis C++ ini juga merupakan *library* yang multi-platform, di mana multi-platform berarti dapat digunakan di beberapa sistem komputer yang berbeda, namun tetap menggunakan *library* yang sama, seperti Linux, Windows, maupun MacOS. Dengan menggunakan *library* ini, komputer mampu mencitrakan keadaan sekitar, seperti halnya manusia, komputer akan mampu meakukan analisis sampai dengan pengenalan terhadap objek-objek yang menjadi target deteksi. OpenCV memdukung penggunaan *library*-

nya untuk beberapa bahasa, diantaranya bahasa pemprograman Python, C / C++, dan bahasa pemprograman java.

3.3.6 Nanpy

Nanpy merupakan *library* yang dapat di akses menggunakan bahasa pemprograman Python yang dapat digunakan untuk mengakses arduino secara langsung melalui komputer, nanpy memberikan kemudahan bagi programer dengan secara langsung mengakses pin-pin yang ada pada arduino dan mengaksesnya hanya perlu menggunakan kabel USB yang terpasang pada arduino.

Untuk menggunakan *library* ini, sebelumnya pengguna harus memasukkan program firmware dari nanpy ke dalam mikro kontroler arduino atau perangkat arduino, sehingga kita dapat menerima input maupun melakukan output melalui bahasa pemprograman Python secara langsung terhadap perangkat arduino.

3.4 Teknik Pengolahan Citra

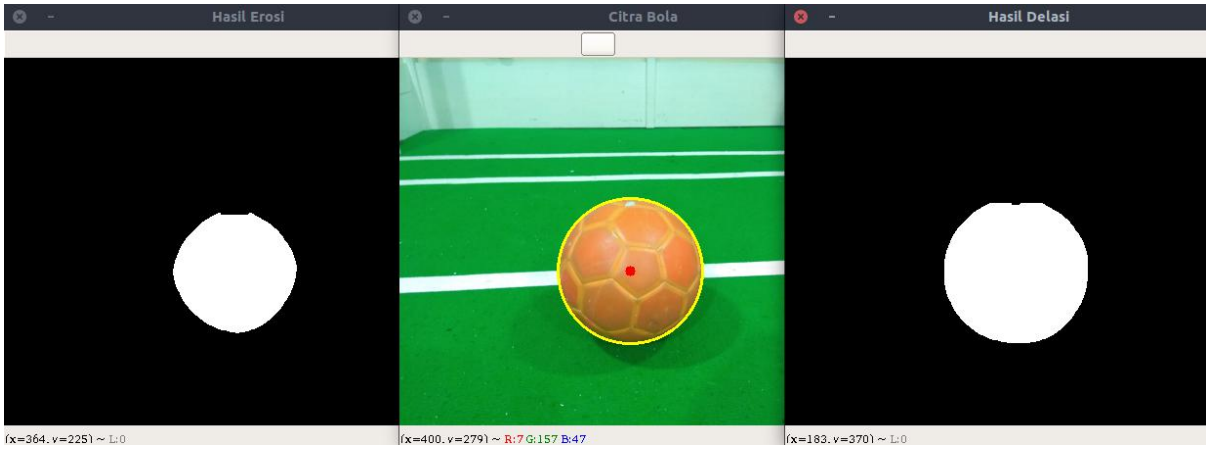
3.4.1 Segmentasi

Segmentasi merupakan teknik yang melakukan pemisahan atau membedakan antara objek yang akan dideteksi dengan latar belakang dari objek pada citra, teknik ini akan sangat berpengaruh terhadap tingkat keberhasilan pengenalan pola yang akan dilakukan dalam penelitian ini (Setiawan dan Firdausy, 2016). Pemisahan objek terhadap latar belakangnya dapat dilakukan dengan memanfaatkan beberapa indikasi, diantaranya berdasarkan warna maupun berdasarkan bentuk ataupun pola dari objek..

3.4.2 Morfologi

Morfologi merupakan salah satu operasi pengolahan citra yang digunakan untuk menormalkan citra objek yang ditargetkan, citra yang dapat diolah pada operasi ini merupakan citra yang telah diubah menjadi citra biner. Pada operasi ini menggunakan dua operasi dasar yaitu operasi dilasi dan operasi erosi (OpenCV team, 2018), operasi dasar erosi merupakan operasi yang menghasilkan pengurangan piksel berdasarkan proses operasi “and”, dengan menggunakan operasi ini piksel yang akan terbaca (aktfif) akan dikurangi dari jumlah piksel sebelumya, sedangkan operasi dilasi berlawanan dengan operasi erosi yaitu operasi ini menggunakan operasi “or”, operasi ini akan menambahkan jumlah piksel dari jumlah yang sebelumnya, hal ini terjadi karena proses operasi “or” yang digunakan tidak mempengaruhi

jumlah piksel yang ada, untuk melihat hasil dan perbedaan pada operasi ini dapat dilihat pada Gambar 3.10. Untuk menggunakan operasi morfologi ini dibutuhkan 2 jenis masukkan, di antaranya citra biner yang akan menggunakan operasi morfologi, dan kernel (structuring element) yang akan digunakan sebagai piksel pusat dari citra tersebut (Kadir & Susanto, 2013).



Gambar 3.10 Hasil operasi erosi dan dilasi

Operasi erosi digunakan untuk mengurangi bentuk objek dari citra asli yang didapatkan dan bertujuan untuk menghilangkan citra-citra yang sejenis namun bukan objek tujuan yang akan diambil operasi ini dirumuskan menggunakan persamaan berikut (Gonzalez & Woods, 2014).

$$A \ominus B = \{ Z \mid (B)_z \subseteq A \}$$

Sedangkan operasi dilasi memperbesar atau melebarkan ukuran dari objek, dapat juga digunakan untuk mengembalikan ukuran asli dari objek yang dideteksi setelah pengurangan piksel hasil operasi erosi, operasi ini dirumuskan sebagai berikut (Gonzalez & Woods, 2014).

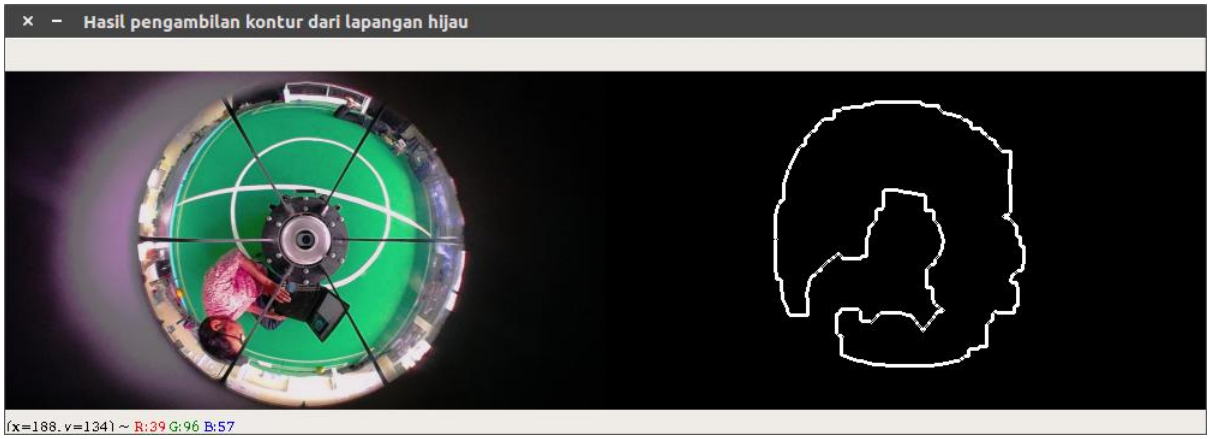
$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}$$

3.4.3 Kontur

Kontur merupakan kumpulan piksel-piksel yang berada pada tepian objek yang membentuk kurva yang mengikuti batasan luar dari piksel yang memiliki warna maupun intesitas yang sama. Kontur biasanya digunakan untuk melakukan analisis terhadap suatu

bentuk objek dari citra dan bahkan digunakan untuk pengenalan dan deteksi objek (OpenCV team, 2018).

Untuk dapat melakukan pencarian kontur, diharuskan menggunakan gambar biner, di mana gambar tersebut sudah menjadi gambar yang hanya memuat warna hitam dan putih yang didapatkan dari hasil thresholding maupun teknik segmentasi. Konsep untuk mendapatkan kontur sendiri sebenarnya sama seperti mencari warna putih di dalam wadah berwarna hitam, yang kemudian hanya akan di ambil koordinat pinggiran dari warna putih tersebut.



Gambar 3.11 Hasil pengambilan kontur dari lapangan hijau

Pada gambar di atas, dilakukan pengambilan kontur untuk area berwarna hijau, sehingga kontur yang didapatkan hanya akan mengambil tepian dari area yang berwarna hijau yang sebelumnya sudah dibatasi hanya akan membaca warna hijau di dalam citra.

3.5 Teknik Pergerakan Robot

3.5.1 Pergerakan *Holonomic*

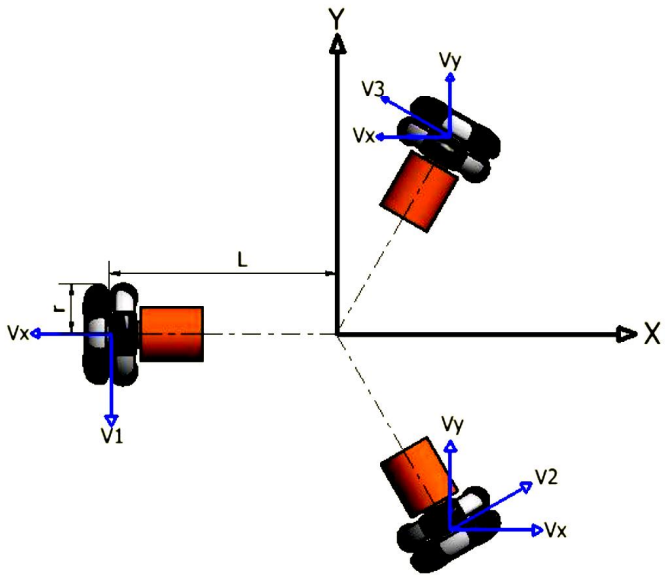
Didunia saat ini mulai banyak dikembangkannya robot yang disebut dengan "Three *omnidirectional drive*", dengan menggunakan tiga buah motor sebagai alat penggerak dari robot. Robot jenis ini merupakan robot yang pergerakannya memiliki tipe *holonomic,* yang berarti robot mampu bergerak ke segala arah, dan bahkan mampu berputar ditempat dengan memanfaatkan satu jenis roda, yaitu roda "*omniwheel*"(Affandi, Sulistijono, dan Ardilla, 2014).



Gambar 3.12 Roda *omniwheel*

3.5.2 Invers Kinematik

Invers kinematik merupakan persamaan matematika yang digunakan untuk menghitung pergerakan robot agar memiliki nilai yang sama pada masing-masing motor dengan nilai masukkan berupa koordinat x, y, dan Θ. Penggunaan persaaman ini bertujuan agar motor memiliki nilai pergerakan yang tidak hanya searah saja, melainkan dapat mengolah data masukkan x, y, dan Θ menjadi kecepatan dan arah pergerakan untuk masing-masing motor, agar robot memiliki gerakan yang luwes kesegala arah.



Gambar 3.13 Representasi invers kinematik pada pergerakan motor tiga
(sumber : Affandi, Sulistijono, dan Ardilla, 2014)

BAB IV
METODOLOGI PENELITIAN

4.1 Analisis Masalah

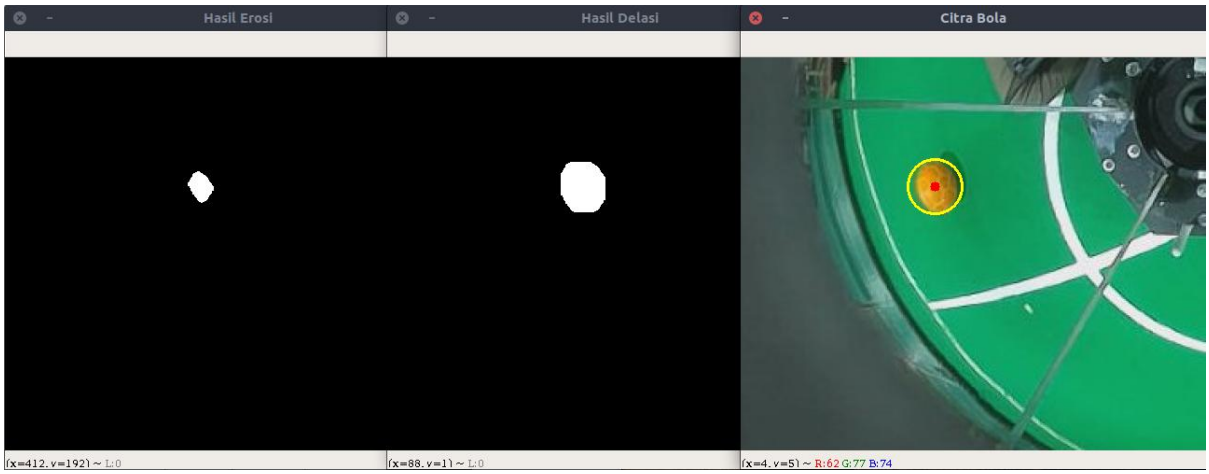
Pada latar belakang dari penelitian ini, sudah dipaparkan beberapa tantangan yang ada pada perlombaan KRSBIB, namun pada penelitian ini hanya akan menjawab dua tantangan, di antaranya robot harus mampu mendeteksi bola, dan robot mampu mengejar bola sesuai dengan posisi bola berada. Robot yang digunakan pada penelitian ini menggunakan teknik *omnidirectional camera* untuk pengambilan citra, yaitu teknik yang mampu menampilkan atau menghasilkan citra dengan kelebihan jangkauan pandang dari berbagai arah, baik itu arah depan, arah belakang, samping kanan, maupun samping kiri. Untuk penampakan dari robot yang digunakan dapat penelitian ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Robot yang dijadikan objek penelitian

Sistem dari robot yang dibuat ini, akan mampu melakukan pembacaan terhadap lapangan hijau yang digunakan, maka dari itu batasan pembacaan bola hanya akan dilakukan di dalam lapangan hijau saja, selain itu sistem akan mampu melakukan pembacaan dengan jenis pencahayaan yang berbeda dengan syarat dilakukan kalibrasi untuk masing-masing pencahayaan. Dari hasil citra yang menggunakan teknik *omnidirectional camera* ini

didapatkan citra yang memiliki sifat cekung untuk sudut pandangnya, di mana hal ini akan mempengaruhi bentuk maupun bidang dari objek yang terdeteksi. Contoh pendeteksian bola dengan menggunakan teknik *omnidirectional camera* ini, menghasilkan bola berubah bentuk menjadi lonjong (oval), seperti yang terlihat pada Gambar 4.2 berikut ini.



Gambar 4.2 Perubahan bentuk bola menjadi lonjong (oval) pada citra

Dari penampakkan citra di atas, dilakukanlah penelitian untuk dapat mendeteksi bola dari hasil penangkapan citra *omnidirectional camera* dan robot dapat bergerak ke arah bola berada sesuai dengan hasil pengolahan citra, yang kemudian pengolahan citra ini akan menghasilkan koordinat posisi dari objek bola. Kemudian dari koordinat yang didapat ini, akan diubah menjadi kecepan dan pergerakan bagi masing-masing motor yang terdapat pada robot, sehingga robot mampu mengarahkan pergerakannya ke arah posisi bola.

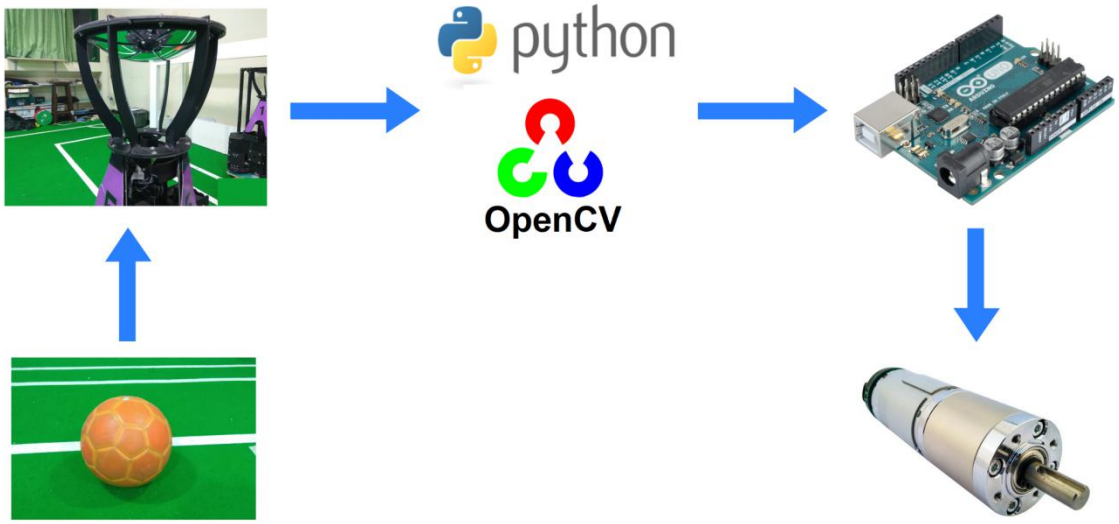
4.2 Perancangan Sistem

Berdasarkan dari hasil penangkapan citra pada Gambar 4.2 di atas, penelitian yang berfokuskan pada pendeteksian objek bola dan pergerakan robot ke arah bola ini, akan dilakukan perancangan terhadap sistemnya, maka dari itu rancangan penelitian ini akan menggunakan sistem *omnidirectioal camera* sebagai teknik pengambilan citra, namun dikarenakan citra yang didapatkan menghasilkan sudut pandang citra yang cekung dan mempengaruhi bentuk dari objek yang akan dideteksi, maka diambillah cara lain untuk melakukan deteksi objek terhadap bola.

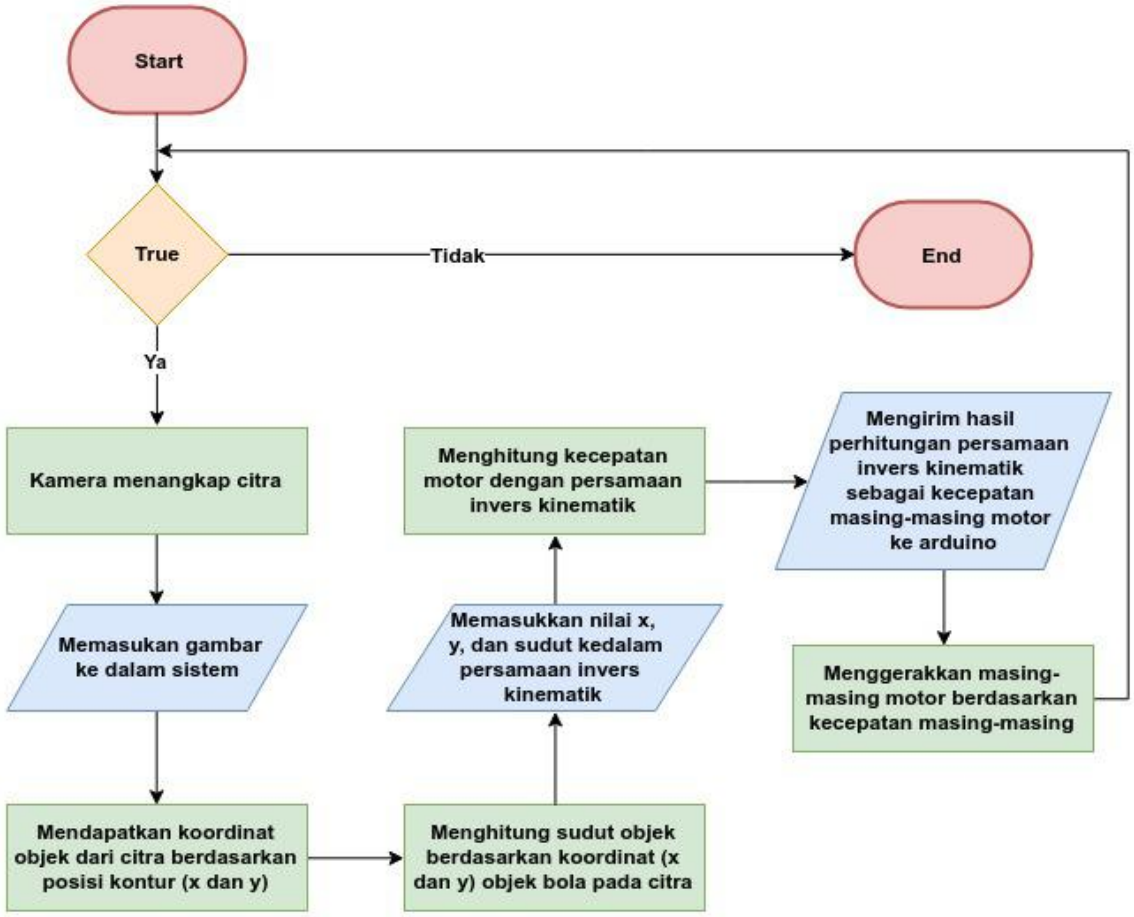
Indikasi yang dapat dimanfaatkan untuk mendeteksi objek selain dari bentuk bola yaitu warna dari objek bola tersebut. Melihat bahwa warna bola dengan warna lapangan hijau, sangatlah kontras dan terlihat berbeda jauh, di dalam lapangan hijau juga tidak terdapat warna

yang mirip dengan warna bola, namun untuk warna yang mirip dengan warna bola akan sangat mungkin terlihat bila warnanya ada di luar lapangan.

Dari permasalahan salah deteksi objek bola ini akan dilakukan pembatasan untuk melakukan deteksi terhadap objek yang memiliki warna serupa dengan bola yang ada di luar lapangan hijau, sehingga sistem hanya akan membaca objek bola yang ada di dalam lapangan hijau. Dan selain itu juga dalam permainan sepak bola, di dalam lapangan hanya akan ada satu buah bola, sehingga tidak perlu khawatir untuk robot mendeteksi objek lain yang memiliki warna serupa dengan bola di dalam lapangan. Rancangan dari sistem dan alur kerja sistem yang akan dibuat dapat dilihat pada Gambar 4.3 dan Gambar 4.4 berikut ini.



Gambar 4.3 Rancangan sistem penelitian



Gambar 4.4 Alur kerja sistem pembacaan bola

4.3 Overview Sistem

4.3.1 Sistem *Omnidirectional Camera*

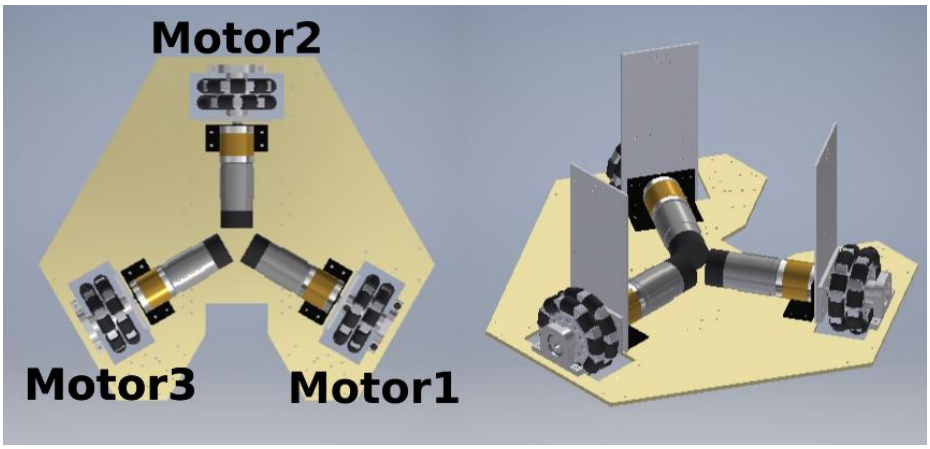
Sistem *omnidirectional carmera* yang digunakan pada penelitian ini, memanfaatkan cermin cembung yang dihadapkan kearah bawah dan kamera yang dihadapkan kearah atas, sehingga citra yang dihasilkan merupakan pantulan citra dari cermin cembung dan mampu menampilkan keadaan ataupun objek-objek yang ada disekitar robot serta menjadikannya citra dengan tampilan dari segala arah, untuk hasil citra yang didapatkan dapat dilihat pada Gambar 3.4 dan untuk overview sistem *omnidirectional camera*-nya sendiri dapat dilihat pada Gambar 4.5 berikut ini.



Gambar 4.5 Sistem *omnidirectional camera*

4.3.2 Sistem *Holonomic*

Sistem *holonomic* yang digunakan pada penelitian ini memanfaatkan tiga buah motor dengan menggunakan jenis roda bertipe *omniwheel*, desain dasar untuk penempatan motor dan pergerakan robot yang menjadi target penelitian ini dapat dilihat pada Gambar 4.6.



Gambar 4.6 Sistem *holonomic* robot

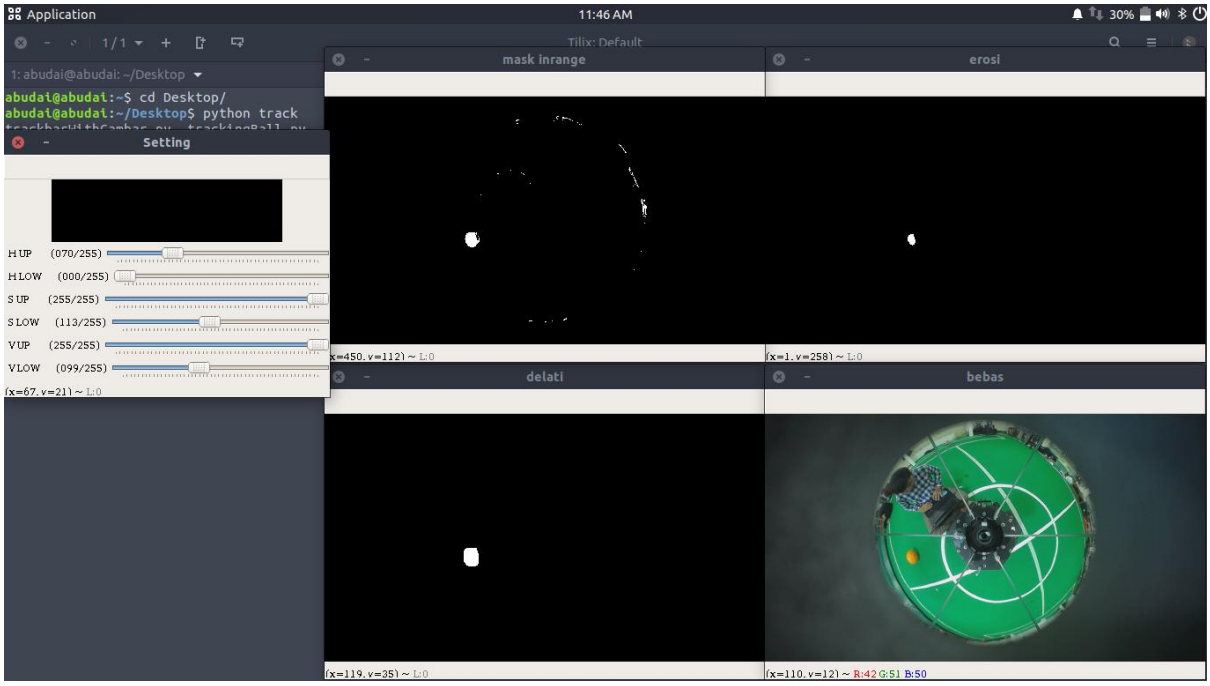
Pada gambar di atas menunjukkan jarak antara masing-masing motor memiliki sudut 120 derajat. Dengan posisi yang miring tersebut, kemungkinan besar robot akan mampu untuk bergerak ke segala arah.

4.3.3 Sistem Multi-Threading

Pada umumnya, sistem multi-threading ini merupakan sistem yang terdapat pada basis komputer, di mana sistem ini bekerja untuk dapat melakukan beberapa pekerjaan secara bersamaan atau secara paralel. Sistem ini memberikan kemudahan bagi penggunanya yang ingin menjalankan suatu proses secara bersamaan dan tidak membebani jalannya proses utama, dikarenakan cara kerja dari sistem ini berjalan dengan menggunakan core yang berbeda dengan program utama, sehingga saat menjalankan program utama, multi-threading ini akan menggunakan core yang lain dan berjalan berdampingan secara bersamaan.

4.3.4 Sistem Kalibrasi Warna

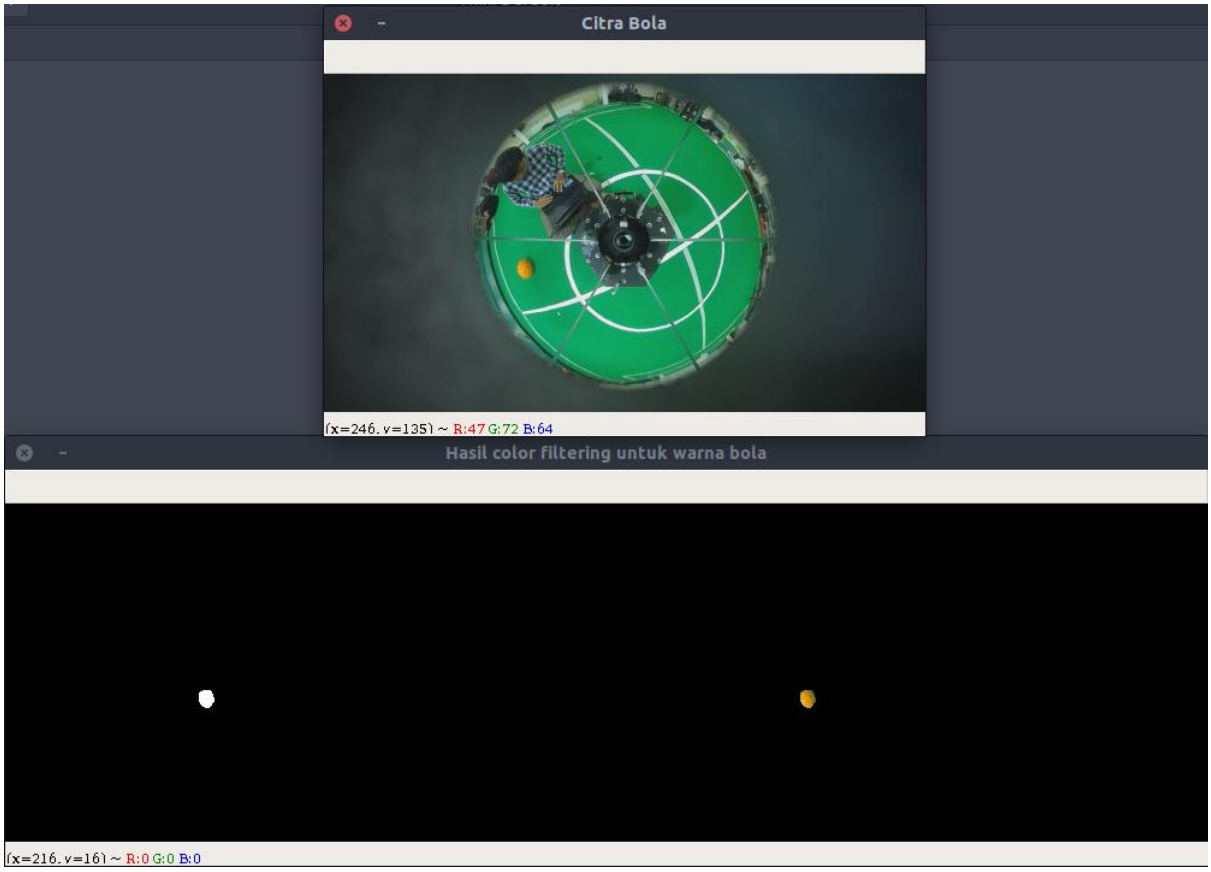
Sistem ini digunakan untuk melakukan pencarian atau penyesuaian terhadap warna dari objek yang akan dideteksi di dalam citra, karena penelitian ini menggunakan metode *color filtering*, maka untuk mendapatkan warna dari objek yang akan dideteksi haruslah memiliki yang namanya sistem kalibrasi, di mana sistem ini akan memberikan pengguna sebuah *trackbar* yang akan digunakan untuk memudahkan pengguna dalam menentukan rentang warna dari objek yang akan dideteksi.



Gambar 4.7 Sistem kalibrasi warna

4.3.5 Sistem *Color Filtering*

Untuk sistem *color filtering* digunakan untuk memisahkan objek-objek berdasarkan warna dari objek yang ingin dideteksi, dengan menggunakan *color filtering* kita dapat lebih fokus untuk mengolah objek, metode tambahan yang digunakan pada proses *color filtering* ini diantaranya operasi erosi yang dimanfaatkan untuk menghapus penangkapan citra lain yang rentang nilai warnanya hampir mirip dengan objek target dan operasi dilasi yang dimanfaatkan untuk mengembalikan ukuran objek ke ukuran yang normal hasil pengurangan piksel dari operasi erosi. Selain itu proses pengolahan citra akan semakin ringan dikarenakan jumlah array yang dikelola akan semakin sedikit, hal ini dikarenakan citra yang diolah hanya pada citra objek target. Penggunaan sistem *color filtering* dapat dilihat pada Gambar 4.8 berikut ini.



Gambar 4.8 Hasil *color filtering*

4.3.6 Sistem *Find-Contours*

Sistem *find-contours* akan mempengaruhi objek apa yang dideteksi, baik itu mempengaruhi pola atau bentuk maupun banyaknya sudut dari objek, kontur yang akan didapatkan dari sistem *find-contours* ini dapat digunakan untuk menemukan koordinat titik

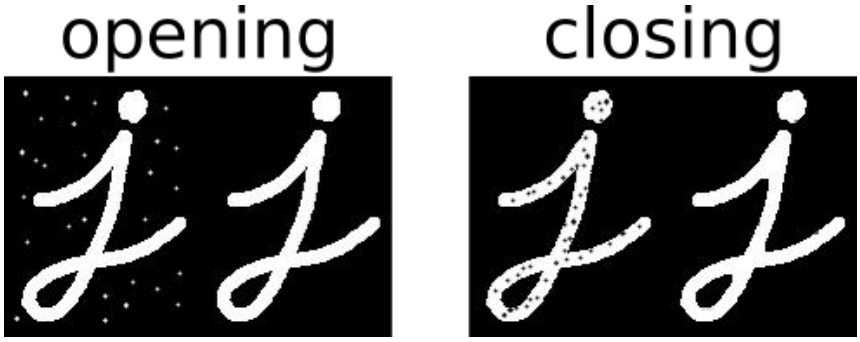
tengah dari objek yang dideteksi, yang kemudian dari koordinat titik tengah ini akan didapatkan nilai koordinat x dan y dari objek, selain itu juga dapat digunakan untuk mendapatkan luas dari objek, sehingga dapat dilakukan penggambaran lingkaran terhadap objek agar menampilkan bahwa objek berhasil terdeteksi, seperti yang terlihat ada Gambar 4.9.



Gambar 4.9 Hasil *find-contours* dari citra

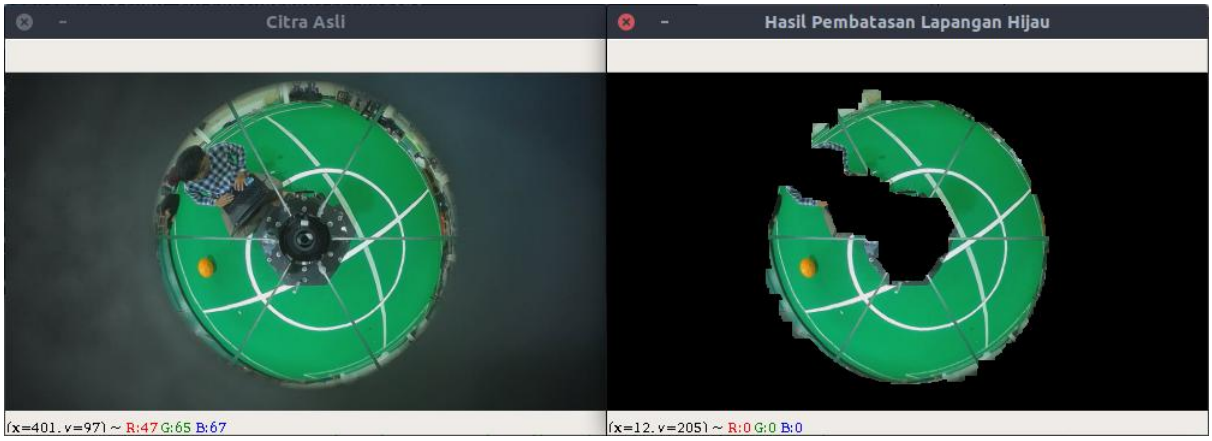
4.3.7 Sistem Pembacaan Lapangan Hijau

Sistem ini berguna untuk membatasi pembacaan daerah lapangan hijau, dengan tujuan agar sistem tidak akan melakukan pembacaan lingkungan atau objek yang berada di luar lapangan hijau dan hanya akan medeteksi objek bola yang berada di dalam lapangan. Sistem ini menggunakan operasi erosi dan delasi untuk menghilangkan daerah yang ada di luar lapangan, di mana untuk melakukan proses penghapusan wilayah luar lapangan yang terdeteksi pada citra, memanfaatkan jumlah iterasi (perulangan untuk proses penambahan maupun pengurangan jumlah piksel) yang akan dilakukan pada dua operasi tersebut. Selain kedua operasi tersebut, juga digunakan ‘operasi closing dan operasi opening’, kedua operasi ini dapat digunakan untuk menghapus objek-objek yang tidak termasuk objek target, untuk operasi closing berfungsi untuk menghapus piksel non-aktif (piksel hitam atau piksel bernilai 0) yang berada di dalam piksel aktif, sedangkan opening berguna untuk menghilangkan piksel aktif yang berada di luar piksel objek target (piksel aktif yang utama), seperti yang terlihat pada Gambar 4.10.



Gambar 4.10 Contoh penggunaan operasi opening dan closing
(sumber : OpenCV, 2018)

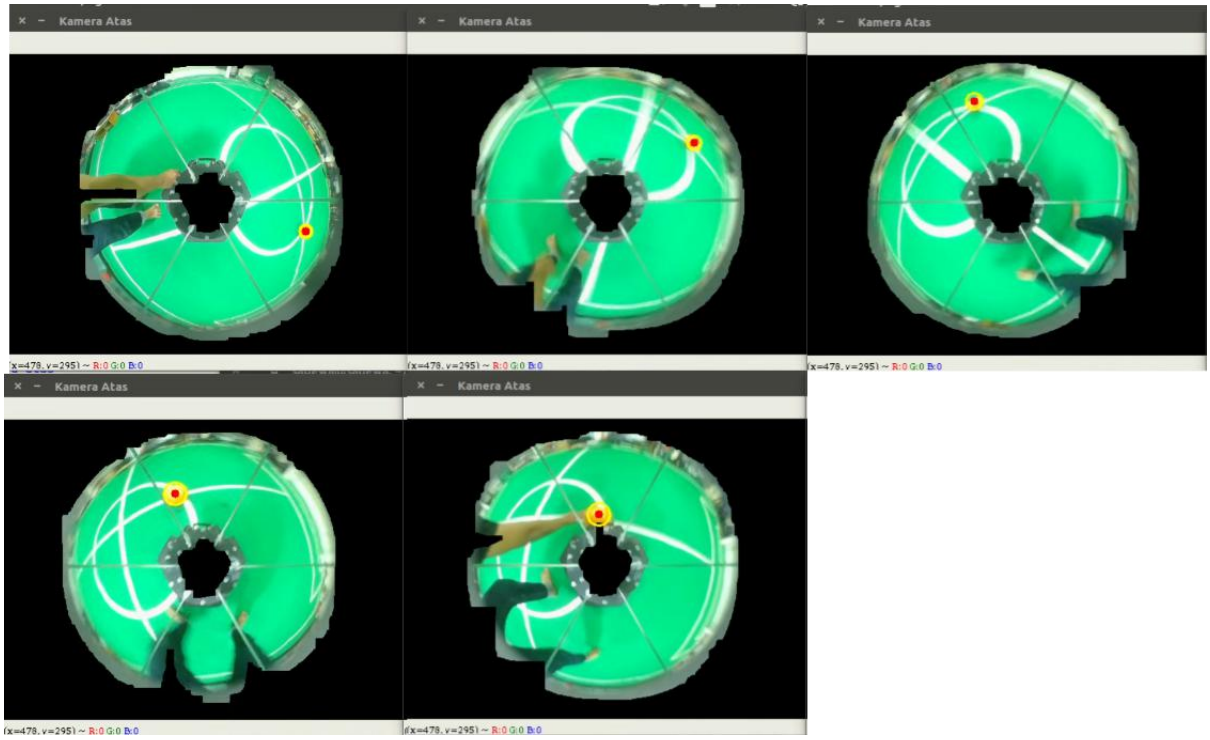
Dan hasil dari sistem ini dapat dilihat pada Gambar 4.11 berikut ini dengan menggunakan jumlah iterasi sebanyak 1 kali iterasi pada operasi erosi dan 3 kali iterasi pada operasi delasi.



Gambar 4.11 Hasil pembatasan pembacaan lapangan hijau

4.3.8 Sistem *Tracking Ball*

Sesuai dengan judulnya sistem ini berperan untuk mengikuti atau bergerak kearah objek bola berada, berdasarkan koordinat (x dan y) objek bola dari citra, kemudian akan diubah atau dimasukkan kedalam persamaan invers kinematik untuk menghasilkan kecepatan motor dan juga arah pergerakan robot agar dapat bergerak mendekati objek bola. Sistem ini akan terus meghasilkan koordinat (x dan y) dari objek pada citra selama bola masih terdeteksi oleh kamera. Untuk sistem *tracking ball* dapat dilihat pada Gambar 4.12 berikut ini.



Gambar 4.12 Tangkapan gambar dari robot mengikuti bola

4.3.9 Sistem Invers Kinematik

Dari koordinat titik tengah objek yang dideteksi (bola), didapatlah nilai koordinat x dan y objek dalam citra, setelah itu akan dilakukan pencarian terhadap nilai sudut dari objek bola dengan menggunakan perhitungan 'arcus tangen' dari koordinat y terhadap x, berikut rumus untuk mendapatkan sudut dari objek bola

$$\text{sudut} = \text{atan}\ (y / x) \text{ atau sudut} = \tan^{-1}\ (y / x)$$

Kemudian nilai x, y dan sudut ini akan dimasukkan ke dalam persamaan invers kinematik untuk dapat menghasilkan gerakan motor yang memiliki arah tujuan pergerakan yang sama pada masing-masing motor agar dapat bergerak kearah bola, berikut persamaan invers kinematik yang digunakan pada penelitian ini.

```
motor1 = -(x * 0.2756) - (y * 0.8612) + (sudut * -0.469)
motor2 = (x * 0.5512) + (sudut * -0.469)
motor3 = -(x * 0.2756) + (y * 0.8612) + (sudut * -0.469)
```

Gambar 4.13 Rumus invers kinematik

4.4 Metode Pengujian

Pengujian sistem dilakukan dengan tujuan untuk membuktikan bahwa sistem yang telah diimplementasikan berjalan dengan baik dan sebagai mana mestinya, untuk melakukan

pengujian terhadap sistem ini akan menjalankan program dari sistem yang sebelumnya telah dilakukan kalibrasi terhadap warna dari bola maupun lapangan hijau yang menjadi target pengolahan citra, kemudian akan dilakukan proses pengolahan terhadap citra hasil tangkapan *omnidirectional camera* dan akan menghasilkan koordinat posisi bola dari sudut pandang robot .

Dari hasil pengolahan citra inilah yang akan diuji untuk membuktikan dan dilihat bahwa sistem pengolahan citra mampu beradaptasi terhadap karakteristik lingkungan, yang memiliki tingkat pencahayaan yang berbeda dengan menggunakan satuan candela sebagai patokan tingkat pencahayaan lapangan. Selain itu, proses pengujian juga akan dilakukan terhadap banyaknya jumlah penggunaan iterasi pada operasi erosi dan operasi dilasi, jarak pandang dari citra yang dihasilkan menggunakan *omnidirectional camera* terhadap keberhasilan sistem mendeteksi objek bola di dalam lapangan hijau, dan sistem pengolahan citra yang menghasilkan koordinat posisi objek untuk membuktikan bahwa robot berhasil bergerak kearah objek bola berada.

BAB V
HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil sistem pembacaan bola, mulai dari proses implementasi kebutuhan perangkat lunak, kalibrasi warna sampai dengan sistem *tracking ball* yang telah di implementasikan agar dapat menjelaskan lebih dalam lagi mengenai sistem yang menjadi objek penelitian saat ini. Selain itu juga pada bab ini akan dibahas mengenai hasil pengujian terhadap sistem. Untuk menyesuaikan penulisan laporan, bab ini akan dibagi menjadi beberapa subbab sesuai dengan sistem yang telah diimplementasikan.

5.1 Implementasi Sistem

5.1.1 Instalasi Program Python

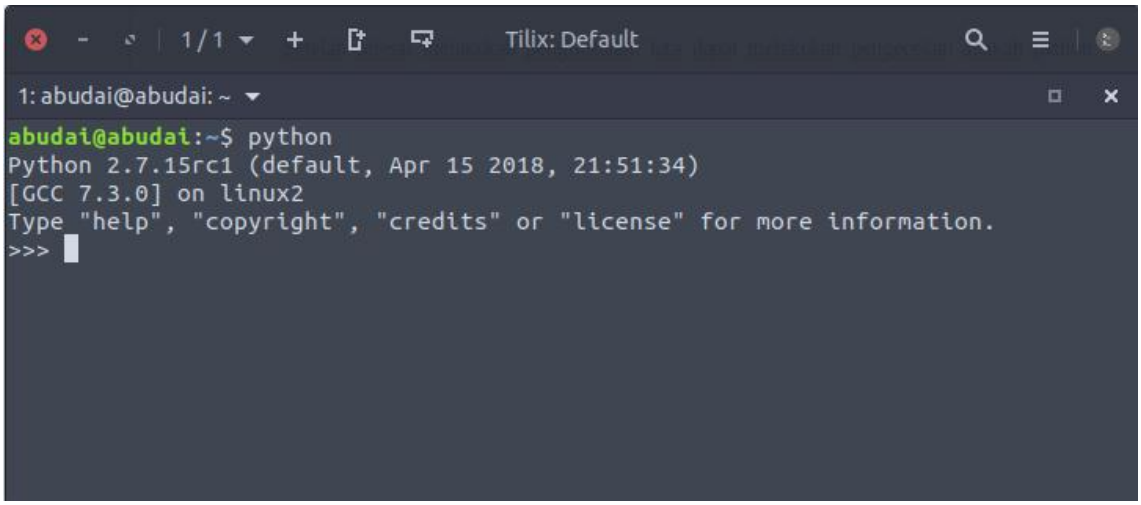
Pada dasarnya *package* instalasi dari ubuntu sudah membawa program Python kedalam sistem operasinya, sehingga pengguna tidak perlu lagi melakukan instalasi untuk program Python, namun pada penelitian ini diasumsikan bahwa Python belum ter-instal sama sekali pada sistem operasi komputer robot. Untuk melakukan penginstalan terhadap program Python , kita dapat menginstalnya melalui terminal.

Instalasi program-program pada ubuntu dapat dilakukan dalam beberapa metode, salah satunya dengan mendapatkannya secara langsung melalui *repository* ubuntu pada terminal. Untuk dapat mengaksesnya, komputer yang digunakan dipastikan telah terhubung dengan koneksi internet, penginstalan program Python dapat dilakukan dengan memasukkan sintak-sintak berikut ini ke dalam terminal.

```
sudo apt install python
```

Gambar 5.1 Sintak pemasangan program Python

Setelah selesai melakukan penginstalan, kita dapat mengecek apakah Python sudah terinstal, dan juga mencari tahu versi Python yang terinstal dengan mengetik “Python” pada terminal. Jika muncul tampilan seperti berikut, maka Python sudah terinstal dan sudah dapat digunakan.



Gambar 5.2 Program Python

Dalam hal ini, kita disarankan untuk menggunakan beberapa *library* yang sesuai dengan kebutuhan , yang berguna untuk mempermudah menjalankan sebuah proses, misalkan *library* untuk mengelola gambar maupun video dapat menggunakan *library* OpenCV. Untuk menambahkan *library* yang dibutuhkan dan mengintegrasikannya dengan program Python, kita membutuhkan satu program yang bernama "get-pip.py", program ini merupakan sebuah *Package* yang berguna untuk menambahkan secara otomatis *library* dari *repository* 'pip' agar dapat langsung ter-integrasi dengan Python. Untuk menginstalnya dapat dilakukan dengan mengetik sintak berikut ini.

```
sudo apt update
sudo apt install python-pip
```

Gambar 5.3 Sintak pemasangan program pip

Pada sintak di atas, dilakukan update sistem terlebih dahulu pada sistem operasi ubuntu, hal ini bertujuan agar memperbarui pembacaan beberapa sistem tambahan (program Python yang baru saja ter-instal), dengan begitu baru dapat dilakukannya penginstalan terhadap *package* pip.

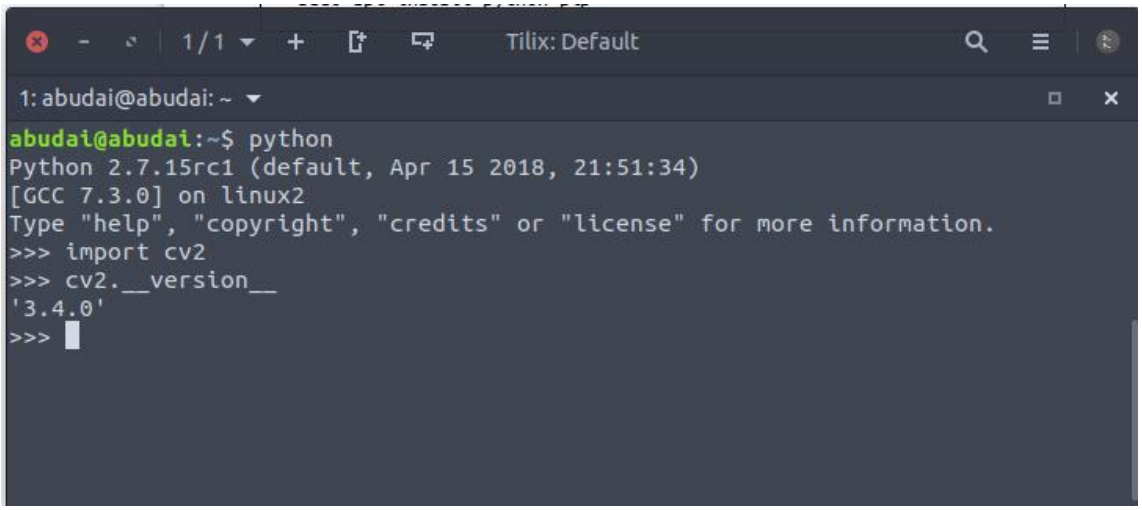
5.1.2 Instalasi *Library* OpenCV

Untuk menambahkan dan mengintegrasikan *library* OpenCV ke dalam Python, dapat juga dilakukan dengan menginstalnya melalui *repository* 'pip' yang sudah tersedia. Dalam *repository* 'pip' sudah tersedia berbagai macam pilihan-pilihan *library* yang di peruntunkan

untuk bahasa pemprograman Python, salah satunya *library* OpenCV. Untuk menginstall OpenCV dapat menggunakan sintak berikut ini.

```
sudo pip install opencv-python
```

Gambar 5.4 Sintak pemasangan *library* opencv



Gambar 5.5 Pemeriksaan versi OpenCV

Pada Gambar 5.4 di atas, melakukan pemeriksaan terhadap OpenCV untuk memastikan bahwa OpenCV berhasil ter-instal dan ter-integrasi dengan program Python yang kita gunakan. Selain itu dilakukan juga pemeriksaan untuk versi OpenCV, guna memudahkan dalam mendapatkan informasi maupun contoh-contoh mengenai versi OpenCV yang digunakan.

5.1.3 Instalasi *Library* Nanpy

Seperti yang telah dijelaskan sebelumnya, nanpy merupakan *library* yang memudahkan kita untuk dapat megakses pin-pin arduino secara langsung dan hanya perlu menggunakan bahasa Python, untuk menginstal *library*-nya sendiri juga dapat menggunakan *repository* 'pip', dengan menggunakan sintak berikut ini.

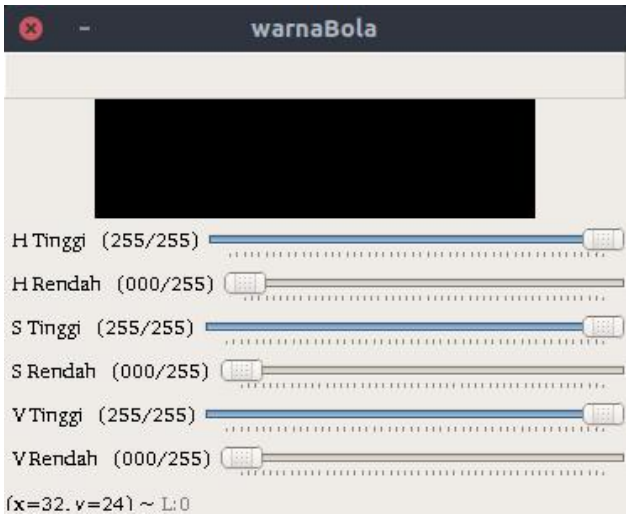
```
sudo pip install nanpy
```

Gambar 5.6 Sintak pemasangan *library* nanpy

5.1.4 Implementasi Sistem Kalibrasi Warna

Untuk sistem kalibrasi warna sendiri, akan disediakan satu jendela baru yang berfungsi untuk dapat menentukan rentang dari nilai warna batas bawah objek sampai dengan rentang

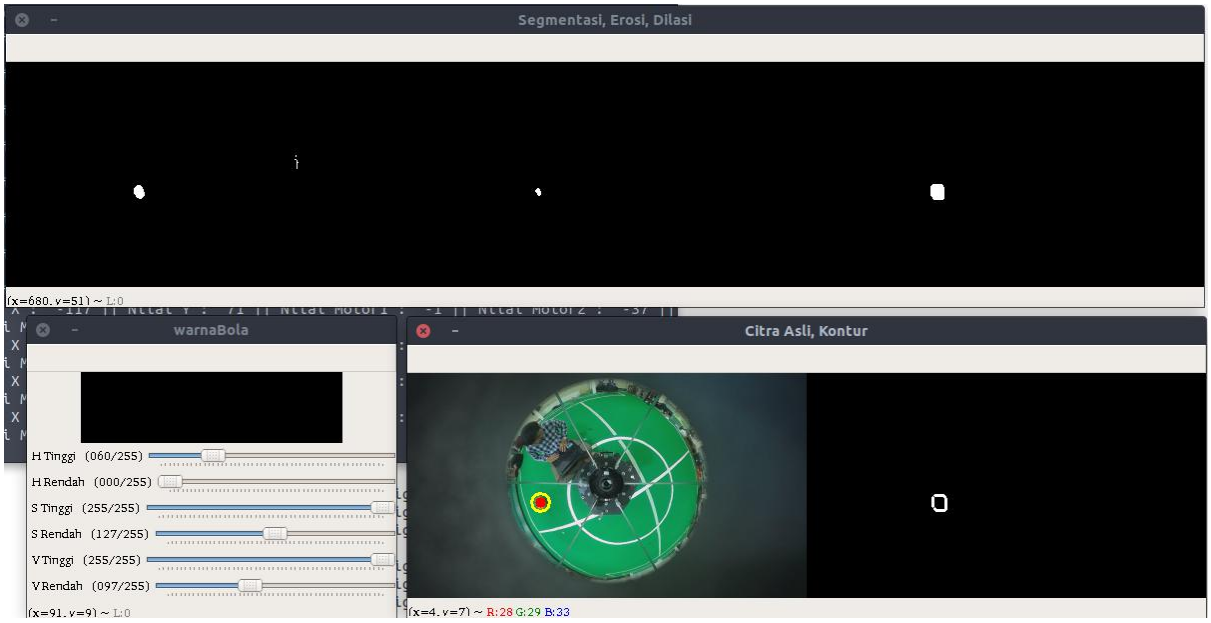
nilai warna batas atas objek yang akan menjadi target pengolahan citra, di antaranya warna objek bola, warna hijau lapangan dan warna hitam objek lain.



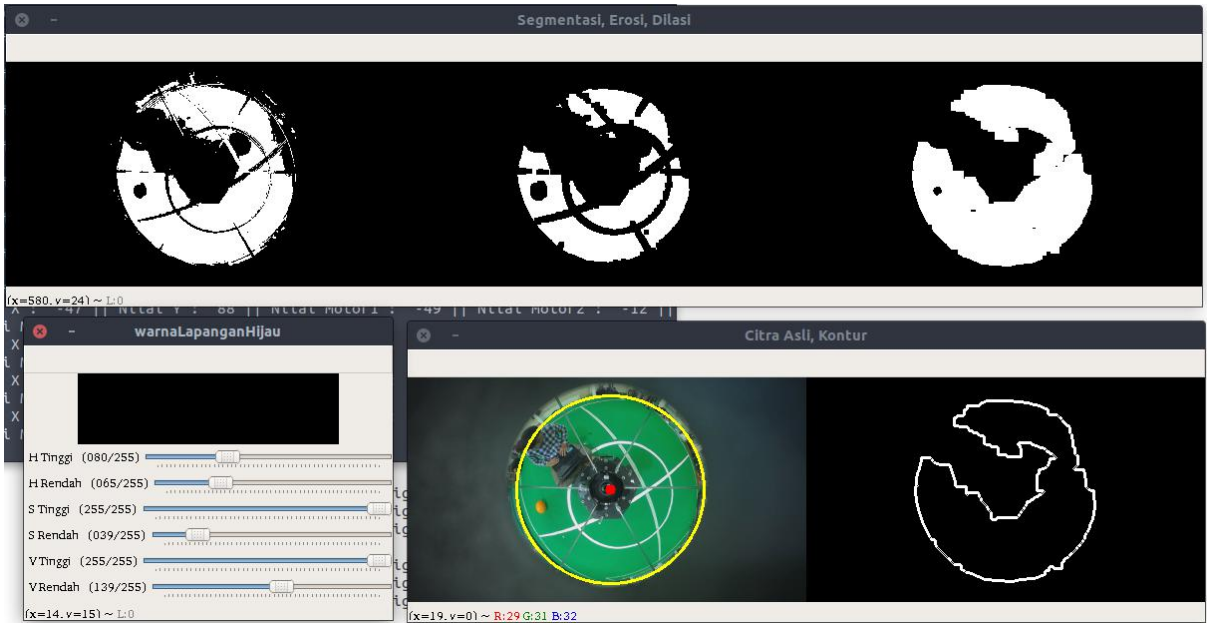
Gambar 5.7 Jendela kalibrasi warna untuk menentukan rentang nilai warna objek bola

Citra yang dihasilkan dari penangkapan kamera, akan berupa citra yang tersusun atas gabungan warna merah, hijau, dan biru (RGB). Namun pada sistem kalibrasi ini, pencarian untuk nilai warna dari objek menggunakan gabungan warna antara Hue, Saturation, dan Value (HSV). Untuk mendapatkan citra dengan gabungan warna HSV, dilakukan konversi warna dari RGB ke HSV dengan menggunakan fungsi "cvtColor" pada *library* OpenCV. Sehinga perubahan citra yang dihasilkan dapat dilihat pada Gambar 3.7.

Untuk proses kalibrasi warna terhadap masing-masing objek yang akan menjadi target pengolahan citra dapat dilihat pada Gambar 5.8 dan Gambar 5.9 berikut ini.



Gambar 5.8 Kalibrasi warna untuk warna bola



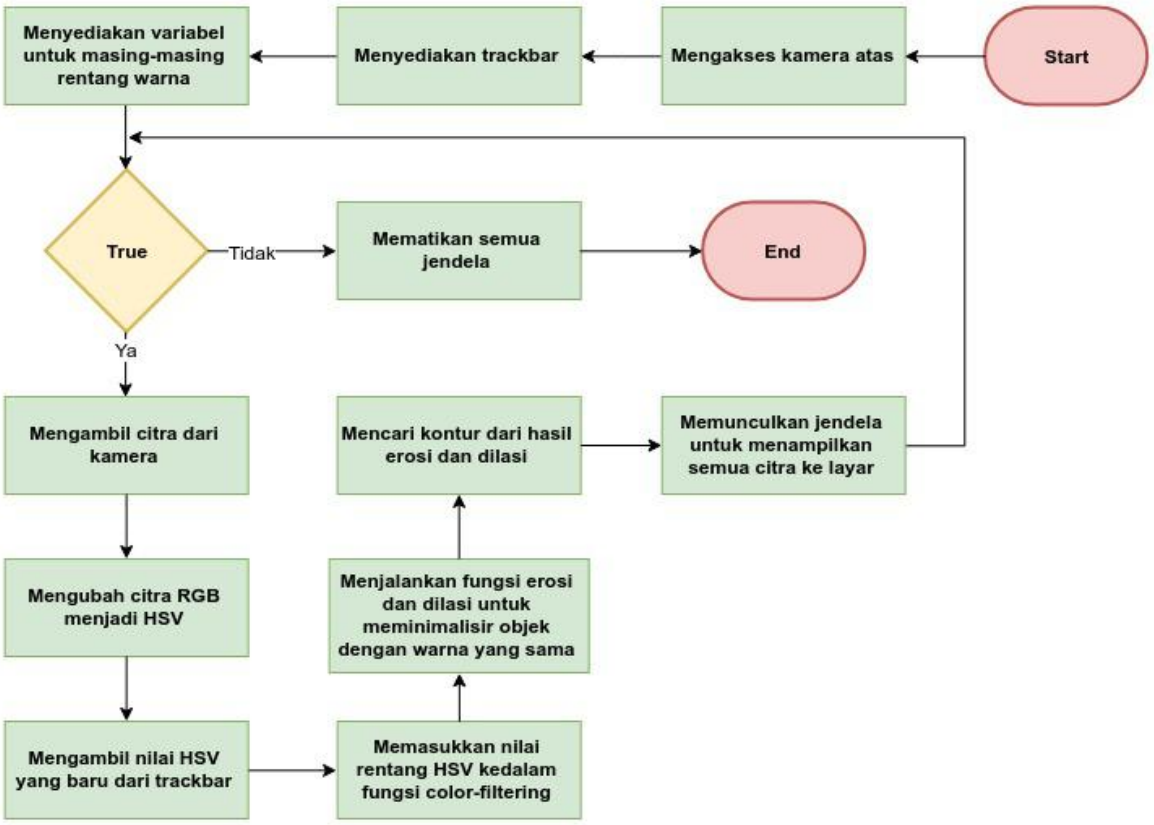
Gambar 5.9 Kalibrasi warna untuk warna lapangan

Cara kerja dari sistem kalibrasi warna ini, dimulai dari penentuan rentang nilai batasan warna terendah dan rentang nilai batasan warna tertinggi dari warna objek dengan menggunakan *trackbar*. Setelah itu, nilai *trackbar* akan dimasukkan kedalam varibel masing-masing, seperti yang terlihat pada sintak dibawah ini.

```
ht = cv2.getTrackbarPos('H Tinggi', config_name)
st = cv2.getTrackbarPos('S Tinggi', config_name)
vt = cv2.getTrackbarPos('V Tinggi', config_name)
hr = cv2.getTrackbarPos('H Rendah', config_name)
sr = cv2.getTrackbarPos('S Rendah', config_name)
vr = cv2.getTrackbarPos('V Rendah', config_name)
```

Gambar 5.10 Sintak variabel penyimpan nilai masing-masing warna

Kemudian nilai yang kita dapatkan tersebut akan dimasukkan ke dalam fungsi “inRange” yang berguna untuk proses segemntasi, agar dapat memisahkan warna dari objek yang menjadi target kita, dengan latar belakang dari objek yang memiliki warna yang berbeda, seperti yang terlihat pada Gambar 5.9 dan Gambar 5.10. Selain itu juga, nilai dari tiap variabel akan disimpan kedalam file JSON, sehingga data hasil kalibrasi warna ini dapat juga di akses oleh file program yang lain.



Gambar 5.11 Alur kerja sistem kalibrasi warna

5.1.5 Implementasi Sistem Pengolahan Citra

Sistem pengolahan citra tersusun dari beberapa metode yang digunakan untuk melacak dan menentukan koordinat dari objek, di antaranya metode segmentasi, morfologi, dan kontur. Metode-metode tersebut digunakan secara bertahap untuk mendapatkan koordinat dari objek target, mulai dari proses pemisahan warna objek dengan warna latar belakang, meminimalisir objek dengan warna yang sama, dan mengambil nilai koordinat(x dan y) dari objek target.

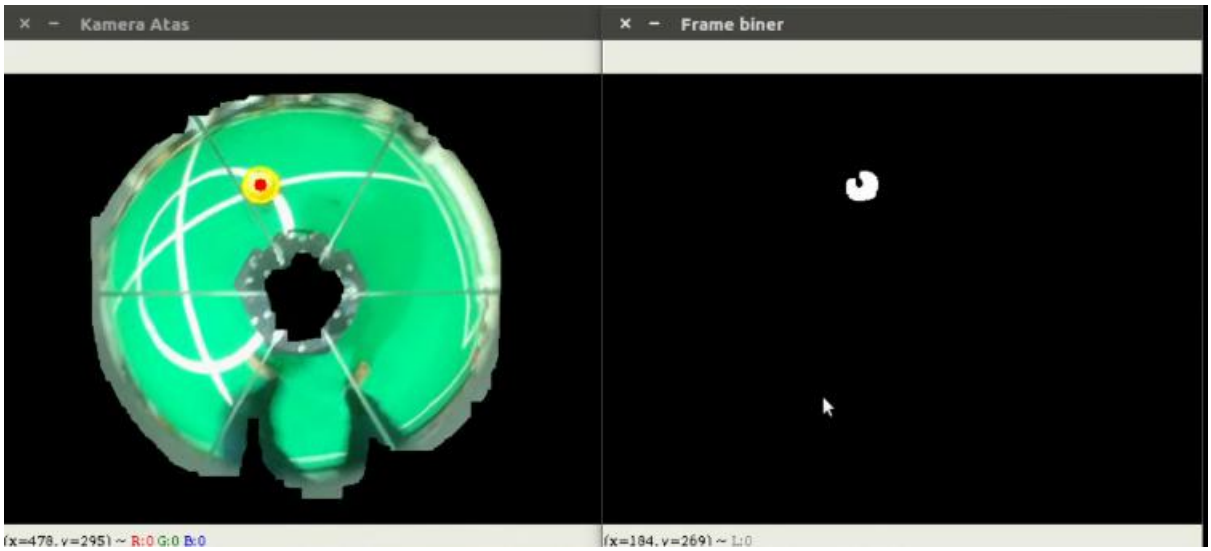
Pada sistem ini sama seperti sistem kalibrasi warna, yaitu sama-sama menggunakan citra dengan gabungan warna HSV, hal ini bertujuan agar nilai warna HSV yang didapatkan dari hasil kalibrasi warna dapat juga bekerja sesuai dengan objek yang di tangkap dari sistem pengolahan citra. Nilai yang didapatkan dari hasil kalibrasi warna yang kemudian disimpan ke dalam file json tersebut, akan di akses oleh sistem pengolahan citra dan diambil berdasarkan variabel masing-masing, seperti yang terlihat pada sintak dibawah ini.

```
from JConfig import JConfig
Config = JConfig()
```

```
#warna bola pada kamera atas
Config.setSection('warnaBola')
warnaRangeAtasBola = np.array(Config.get('rangeAtas'))
warnaRangeBawahBola = np.array(Config.get('rangeBawah'))
#warna Lapangan pada kamera atas
Config.setSection('warnaLapanganHijau')
warnaRangeAtasLapangan = np.array(Config.get('rangeAtas'))
warnaRangeBawahLapangan = np.array(Config.get('rangeBawah'))
#warna objek hitam pada kamera atas
Config.setSection('warnaObjekHitam')
warnaRangeAtasObjekHitam = np.array(Config.get('rangeAtas'))
warnaRangeBawahObjekHitam = np.array(Config.get('rangeBawah'))
```

Gambar 5.12 Sintak pengambilan nilai masing-masing warna yang disimpan

Nilai yang ditampung melalui varibel masing-masing tersebut akan digunakan sebagai parameter untuk pemisahan warna atau *color-filtering* menggunakan fungsi "inRange", sehingga menghasilkan citra yang hanya akan menampilkan objek dari warna yang telah ditentukan seperti yang terlihat pada Gambar 4.2, selanjutnya akan dilakukan beberapa operasi morfologi, yang diantaranya operasi erosi, dilasi, closing, dan operasi opening, penggunaan operasi ini akan digunakan untuk melakukan pembatasan pembacaan lapangan hanya diarea berwarna hijau, sedangkan untuk mendeteksi objek bola hanya akan menggunakan operasi erosi dan operasi delasi. Selanjutnya citra hasil dari proses morfologi ini akan dicarikan kontur dari area hijau pada lapangan, setelah didapatkan tepian lapangan hijau dengan fungsi *find-contours*, kemudian hasil citra ini akan digunakan sebagai citra utama untuk mencari keberadaan bola, sehingga pembacaan bola tidak akan dilakukan di luar lapangan hijau.



Gambar 5.13 Pembatasan pembacaan untuk lapangan hijau dan *tracking ball* di dalam lapangan hijau

Selanjutnya setelah di batasi pembacaan wilayah hijau, akan dilakukan deteksi bola di dalam lapangan hijau. Sebelumnya nilai warna bola hasil kalibrasi warna akan dimasukkan ke dalam fungsi *color-filtering*, setelah proses pemisahan objek dengan latar belakang objek, kemudian dilakukan operasi erosi dan dilasi, hal ini bertujuan untuk meminimalisir masuknya objek dengan warna sama namun bukan menjadi objek tujuan, sehingga hanya akan berfokus pada objek bola dan menghasilkan citra yang mendeteksi objek bola, seperti yang terlihat pada Gambar 5.14. Citra hasil dari proses erosi dan dilasi tersebut, kemudian akan dicari kontur dari objek untuk dapat menentukan dan mendapatkan koordinat (x dan y) dari objek yang dideteksi, dengan koordinat objek inilah yang kemudian akan dijadikan sebagai kecepatan dan ke mana arah motor DC harus bergerak.

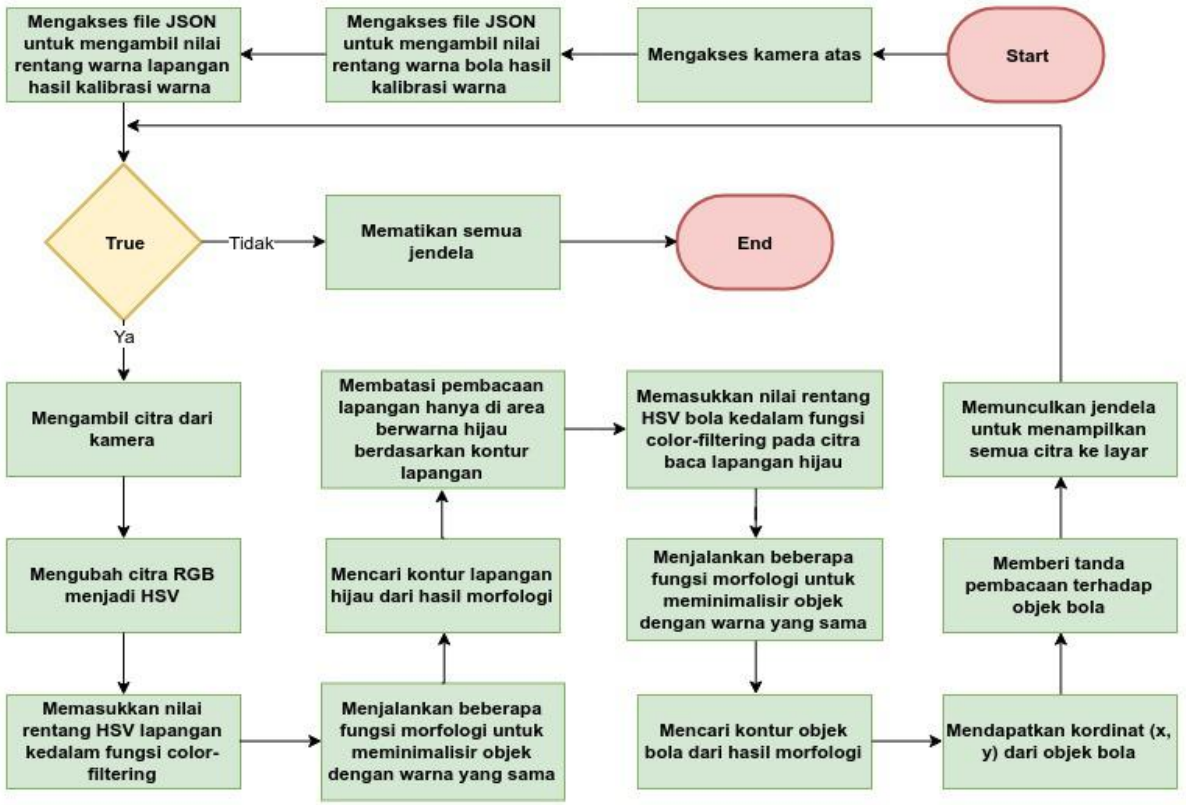
Namun jika objek bola tidak dideteksi oleh sistem, maka sistem akan memberikan nilai koordinat x = 10, dan nilai koordinat y akan bernilai 0, hal ini bertujuan agar nilai invers kinematik yang dihasilkan akan memberikan kecepatan dan pergerakan motor yang dapat membuat robot bergerak secara parabola. Gerakan parabola dipilih, karena diasumsikan bahwa objek bola tidak terdeteksi dipengaruhi oleh jarak pandang dari citra yang dihasilkan oleh *omnidirectional camera* memiliki keterbatasan pembacaan, dengan begitu diharapkan gerakan parabola ini dapat melakukan pencarian atau pelebaran berputar untuk robot sehingga objek bola yang memiliki posisi di luar jarak cakupan *omnidirectional camera* dapat kembali ter deteksi oleh sistem. Pada bagian sistem invers kinematik di atas, terdapat rumus untuk mendapatkan sudut, dijelaskan bahwa dilakukan perhitungan sudut berdasarkan ‘arcus

tangen' dari koordinat y terhadap x, namun pada penelitian ini ada perubahan parameter yang dimasukkan untuk mendapatkan nilai dari sudut objek target, sehingga rumus di atas akan berubah menjadi seperti berikut.

$$sudut = atan\ (x / y)\ atau\ sudut = tan^{-1}\ (x / y)$$

Perubahan menjadi 'arcus tangen' dari koordinat x terhadap y didasari oleh teknik pengambilan citra yang dilakukan, di mana pengambilan gambar dengan teknik pantulan cermin cembung ini menghasilkan citra yang terbalik dari kondisi aslinya dilapangan, dari hal inilah dilakukannya pengembalian keadan citra menjadi normal dan menghasilkan koordinat yang berlawan dari citra hasil pantulan cermin cembung.

Selain itu, pada sistem ini dibatasi juga agar saat robot mendapatkan bola, maka akan memberhentikan segala aktifitas motor, sehingga saat bola telah masuk ke dalam pegangan robot sistem ini akan memberikan nilai pada motor yang berguna untuk memberhentikan pergerakan dengan artian bahwa robot sudah mendapatkan bola.



Gambar 5.14 Alur sistem pengolahan citra

5.1.6 Implementasi Sistem Pergerakan Robot

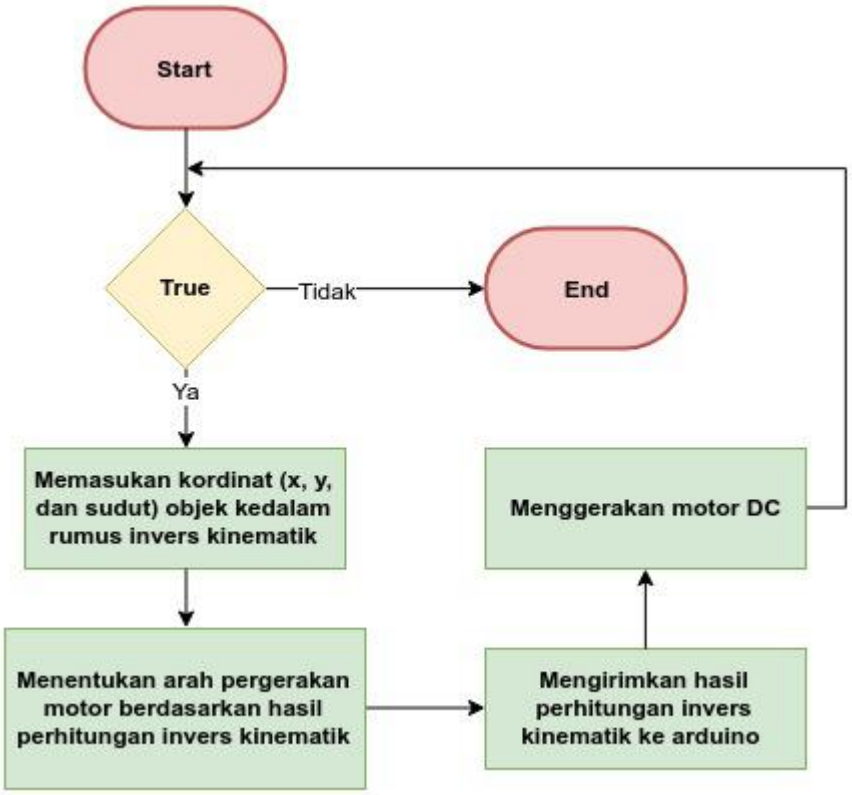
Pergerakan robot didapatkan dari hasil perhitungan menggunakan invers kinematik dari koordinat (x, y, dan sudut) dari objek bola yang telah berhasil dideteksi dengan sistem pengolahan citra, kemudian hasil dari perhitungan inilah yang akan menjadi nilai kecepatan dari motor DC dan arah bergerak motor untuk dapat bergerak mendekati bola. Nilai ini akan dikirimkan melalui arduino yang menjadi mikro kontroler untuk mengendalikan motor DC agar bergerak sesuai dengan nilai hasil dari perhitungan invers kinematik. Untuk perintah pengiriman ke arduino sendiri menggunakan bantuan *library* nanpy yang mengijinkan pengguna untuk dapat mengaksesnya hanya dengan bahasa Python, seperti yang terlihat pada sintak berikut.

```
from nanpy import ArduinoApi, SerialManager
import sys
#================Koneksi ke arduino==================
try:
connection = SerialManager()
arduino = ArduinoApi (connection = connection)
print("Terhubung dengan Arduino")
except :
print("Gagal Terhubung dengan Arduino")
sys.exit(0)
#===============Deklarasi Pin=====================
#motor 1
motor1PWM1 = 2
motor1PWM2 = 3
motor1IN1 = 14
#motor 2
motor2PWM1 = 4
motor2PWM2 = 5
motor2IN1 = 15
#motor 3
motor3PWM1 = 6
motor3PWM2 = 7
motor3IN1 = 16
#drible 1
```

```
dribleKiriPWM1 = 9
dribleKiriPWM2 = 8
dribleKiriIN1 = 17
#drible 2
dribleKananPWM1 = 11
dribleKananPWM2 = 10
dribleKananIN1 = 18
i = 2
while i <= 18:
arduino.pinMode(i, arduino.OUTPUT)
arduino.digitalWrite(i, arduino.LOW)
i = i+1
tombolOnOff = 44
arduino.pinMode(tombolOnOff, arduino.INPUT)
arduino.digitalWrite(tombolOnOff, arduino.HIGH)
potensio1 = 0
potensio2 = 1
arduino.pinMode(potensio1, arduino.INPUT)
arduino.pinMode(potensio2, arduino.INPUT)
```

Gambar 5.15 Sintak pengaksesan arduino menggunkan *library* nanpy

Pada dasarnya, *library* ini bekerja atau berkomunikasi dengan arduino dengan menggukan komunikasi serial, di mana komunikasi ini dapat digunakan untuk melakukan pertukaran informasi dari satu perangkat ke perangkat yang lain yang merupakan perangkat yang berbeda.



Gambar 5.16 Alur sistem pergerakan robot

Untuk pengiriman nilai hasil perhitungan invers kinematik dan pengaksesan mikro kontroler arduino dapat dilihat pada sintak berikut ini.

```
if motor1 >= 0 :
dpa.arduino.analogWrite(dpa.motor1PWM1, motor1)
dpa.arduino.analogWrite(dpa.motor1PWM2, 0)
dpa.arduino.digitalWrite(dpa.motor1IN1, dpa.arduino.HIGH)
elif motor1 < 0 :
dpa.arduino.analogWrite(dpa.motor1PWM1, 0)
dpa.arduino.analogWrite(dpa.motor1PWM2, motor1*-1)
dpa.arduino.digitalWrite(dpa.motor1IN1, dpa.arduino.HIGH)


if motor2 >= 0:
dpa.arduino.analogWrite(dpa.motor2PWM1, 0)
dpa.arduino.analogWrite(dpa.motor2PWM2, motor2)
dpa.arduino.digitalWrite(dpa.motor2IN1, dpa.arduino.HIGH)
elif motor2 < 0 :
```

```
dpa.arduino.analogWrite(dpa.motor2PWM1, motor2*-1)
dpa.arduino.analogWrite(dpa.motor2PWM2, 0)
dpa.arduino.digitalWrite(dpa.motor2IN1, dpa.arduino.HIGH)


if motor3 >= 0 :
dpa.arduino.analogWrite(dpa.motor3PWM1, motor3)
dpa.arduino.analogWrite(dpa.motor3PWM2, 0)
dpa.arduino.digitalWrite(dpa.motor3IN1, dpa.arduino.HIGH)
elif motor3 < 0 :
dpa.arduino.analogWrite(dpa.motor3PWM1, 0)
dpa.arduino.analogWrite(dpa.motor3PWM2, motor3*-1)
dpa.arduino.digitalWrite(dpa.motor3IN1, dpa.arduino.HIGH)
```

Gambar 5.17 Sintak untuk algoritma pergerakan motor

5.2 Menjalankan Sistem

Sebelum dimulainya sebuah permainan robot sepak bola, terlebih dahulu akan dilakukan beberapa persiapan, salah satunya akan dilakukan kalibrasi terhadap warna dari objek bola yang akan menjadi target perebutan dengan lawan. Kalibrasi ini berguna untuk memastikan bahwa robot akan menemukan dan melacak bola target, mampu membedakan ataupun membatasi pembacaan objek yanga ada di dalam area lapangan dengan area di luar lapangan, sehingga hanya akan bergerak ke arah objek bola yang ada di dalam lapangan dengan mencari warna bola yang telah ditentukan berdasarkan hasil kalibrasi warna.

Saat menjalankan program sistem kalibrasi warna ini, akan dilakukan pengambilan gambar secara *realtime* dengan teknik pengambilan gambar menggunakan teknik *omnidirectional camera*, kemudian gambar *realtime* tersebut akan dimasukkan ke dalam sistem dan dikelola, sehingga sistem akan memunculkan 3 jendela baru seperti yang terlihat pada Gambar 5.8 dan Gambar 5.9, di antaranya yang pertama jendela untuk menampilkan citra asli dan kontur, yang kedua jendela yang akan menampilkan hasil segmentasi, erosi dan dilasi, dan jendela yang ketiga akan menampilkan jendela *trackbar* yang akan ditampilkan sebanyak 3 tahap untuk 3 pengaturan yang berbeda, di antaranya *trackbar* untuk nilai warna bola sebagai pengaturan yang akan muncul pertama kali saat menjalankan sistem, *trackbar* untuk nilai warna lapangan, dan terakhir *trackbar* untuk nilai warna objek hitam (lawan) seperti yang terlihat pada Gambar 5.18.

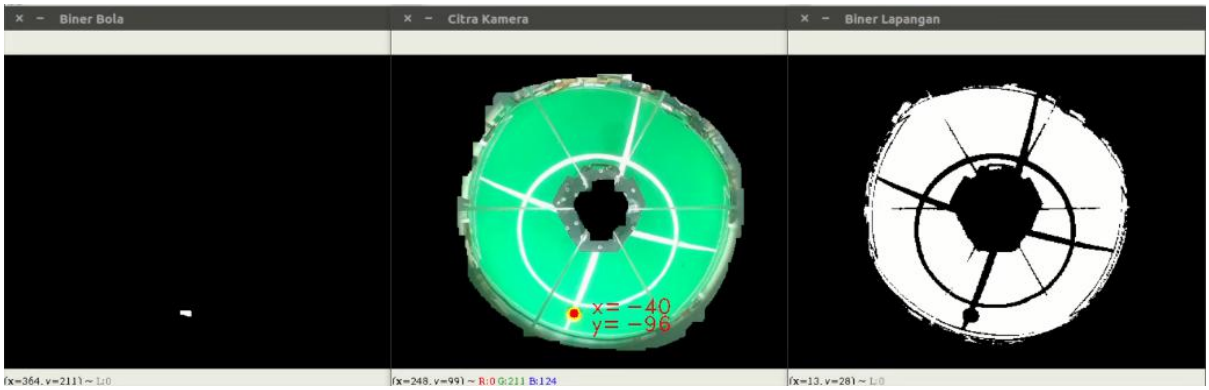


Gambar 5.18 Tiga jendela *trackbar* untuk menentukan nilai warna dari objek target

Trackbar ini akan digunakan untuk melakukan kalibrasi warna dengan tujuan mempermudah penentuan rentang nilai warna HSV dari objek bola dan lapangan. Untuk mendapatkan nilai warna batas atas dan nilai warna batas bawah dari objek target (rentang nilai warna HSV) dengan menggunakan *trackbar* yang disediakan pada sistem, untuk mendapatkan nilai HSV ini dilakukan menggunakan teknik "Trial and Error", di mana pencarian nilai terhadap warna dilakukan secara terus menerus dengan menggerakan masing-masing bar pada susunan hue (batas atas dan bawah), saturation (batas atas dan bawah) dan value (batas atas dan bawah) ke kiri (nilai semakin mengecil bergerak ke arah 0) mapun ke kanan (nilai semakin membesar bergerak ke arah 255), sampai dengan kondisi untuk mendapatkan rentang nilai yang cocok dengan objek yang menjadi target deteksi. Setelah mendapatkan rentang nilai dari warna objek, untuk menyimpan nilai tersebut dapat dilakukan dengan menekan tombol 's' pada *keyboard*, yang berarti rentang nilai warna tersebut akan tersimpan kedalam file json yang disediakan sistem, sedangkan untuk melanjutkan ke pengaturan berikutnya (mencari rentang nilai warna untuk warna lapangan) dapat dilakukan dengan menekan tombol 'n' pada *keyboard*, dengan begitu akan memunculkan jendela *trackbar* untuk mengatur rentang nilai warna dari pengaturan berikutnya (baik itu lapangan maupun objek hitam) dan nilai pada *trackbar*-nya akan kembali ke nilai bawaan sistem, yaitu 0 (nol), dan untuk memberhentikan jalannya sistem kalibrasi warna dapat dilakukan dengan menekan tombol 'Esc'.

Setelah selesai melakukan proses kalibrasi warna terhadap objek yang menjadi target deteksi dan menyimpannya ke dalam file json, selanjutnya menjalankan sistem pengolahan citra. Saat pertama kali menjalankan sistem pengolahan citra ini, sistem akan menjalankan program yang berguna untuk menghubungkan komunikasi antara komputer (bahasa pemprograman Python) dengan perangkat arduio dan menggunakan *library* nanpy, kemudian saat komunikasi berhasil dibuat, sistem akan memunculkan 3 jendela baru, di antaranya yang

pertama jendela citra biner dari pendeteksian lapangan, yang kedua citra asli dari penangkapan gambar menggunakan *omnidirectional camera*, dan yang ketiga merupakan citra biner dari pendeteksian objek bola, seperti yang terlihat pada Gambar 5.19.



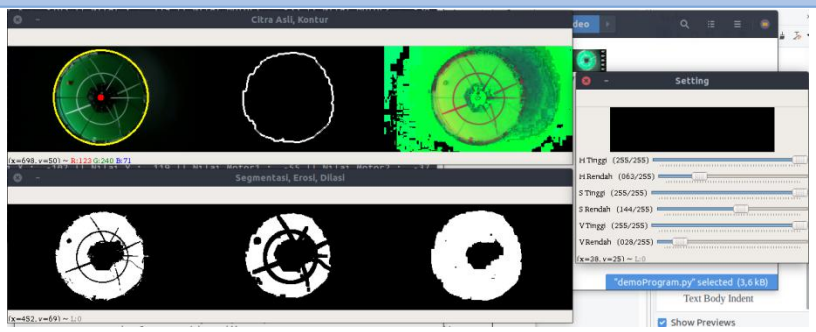
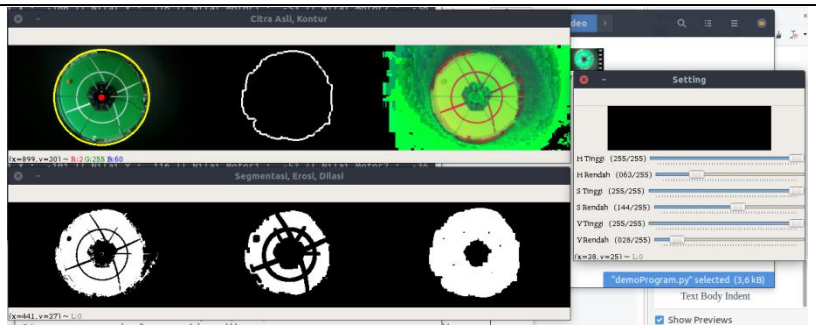
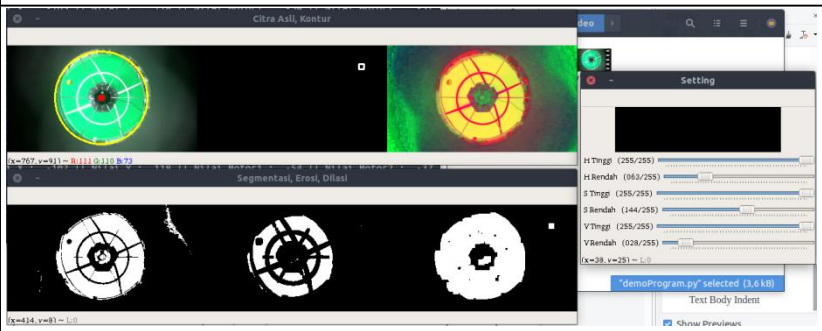
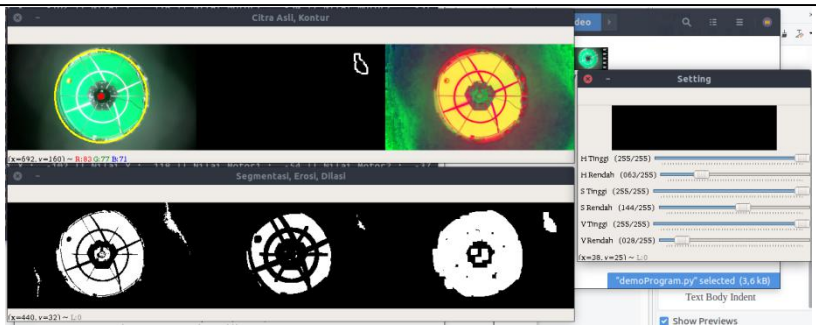
Gambar 5.19 tiga jendela sistem pengolahan citra

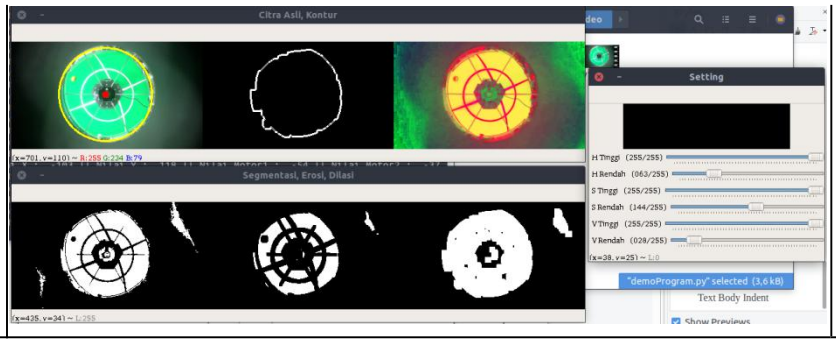
Sistem ini akan mengambil gambar secara *realtime* dan dimasukkan kedalam sistem untuk kemudian diproses citra tersebut, setelah itu mengambil rentang nilai warna dari hasil kalibrasi warna yang tersimpan di dalam file json dan melakukan proses segementasi maupun morfologi untuk masing-masing objek (objek lapangan dan bola) berdasarkan warna yang telah di dapatkan, selanjutnya setelah mendapatkan citra biner dari masing-masing objek, dicarilah kontur objek untuk dapat mengetahui titik tengah dari objek target (bola), dari titik tengah objek inilah kita dapat mendapatkan koordinat (x dan y) dari objek target. Koordinat (x, y dan sudut) dari objek bola dimasukkan kedalam persamaan invers kinematik, sehingga akan menghasilkan nilai kecepatan untuk masing-masing perputaran motor dan arah gerakan motor, selanjutnya akan dikomunikasikan dengan perangkat arduino agar dapat mengendalikan motor sesuai dengan kecepatan dan pergerakan hasil dari perhitungan invers kinematik agar dapat bergerak ke arah posisi bola berada.

5.3 Hasil dan Pengujian Sistem

Pengujian pertama akan dilakukan untuk mengetahui pengaruh dari perubahan pencahayaan yang terjadi diarea lapangan, dimulai dari pencahayaan redup sampai dengan pencahayan yang terang dengan nilai kalibrasi yang sama, pengukuran tingkat kecerahan yang terdapat pada area robot berada menggunakan sebuah perangkat lunak yang bernama “Lux Light Meter” yang terdapat pada smartphone berbasis android yang memanfaatkan sensor “Ambient Light Sensor” pada smartphone. Hasil dan pengujian dari perubahan pencahayaan dapat dilihat pada tabel dibawah ini.

Tabel 5.1 Pengujian perubahan pencahayaan

No.	Citra *omnidirectional camera*	Keterangan daerah lampu menyala
1		
2		
3		
4		

5		

Pengujian di atas merupakan pengujian yang dilakukan untuk menguji sistem pembacaan lapangan hijau, apakah sistem masih mampu mengenali lapangan hijau dengan tingkat pencahayaan yang berubah-ubah. Untuk hasil pengujian pada citra diatas dapat dilihat pada tabel berikut ini.

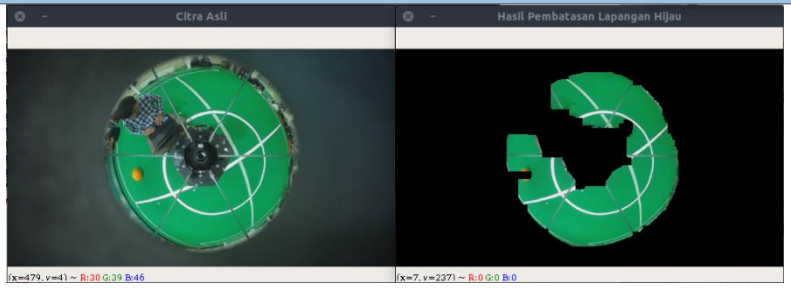
Tabel 5.2 Pengujian perubahan pencahayaan

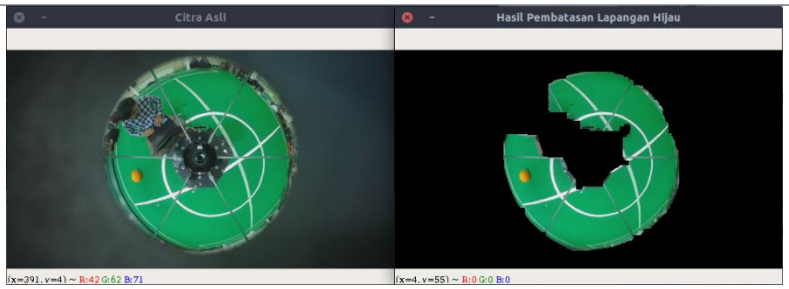
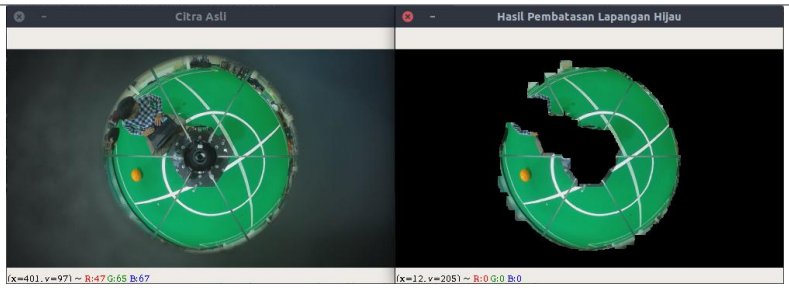
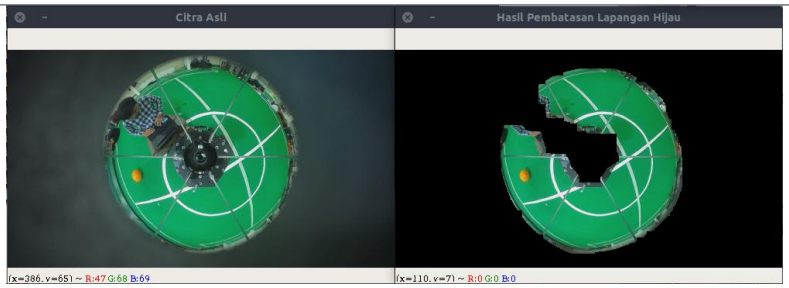
No.	Keterangan Daerah Lampu Menyala	Tingkat Kecerahan (Candela)	Hasil
1	Pada baris ke-1 tabel, terlihat bahwa hanya 1 area lampu neon panjang yang menyala (posisi lampu kanan lapangan) sehingga menghasilkan penerangan yang kurang merata di arena lapangan.	antara 35 - 37 Cd	Sistem masih mengenali lapangan hijau, hal ini terlihat pada hasil citra biner yang menampilkan deteksi pada lapangan hijau
2	pada baris ke-2 tabel, dua area lampu neon panjang diatas lapangan dinyalakan (posisi lampu kiri dan kanan lapangan dinyalakan) terlihat arena lapangan dengan pencahayaan yang cukup merata.	antara 39 - 42 Cd	Tingkat pencahayaan masih mirip atau rentang nilainya masih mencakup dengan baris ke-1 tabel, sehingga kemungkinan pembacaan lapangan masih dapat dilakukan dengan baik.
3	pada baris ke-3 semua lampu neon panjang dimatikan dan hanya menyalakan lampu spiral yang menghasilkan cahaya yang	antara 130 - 133 Cd	Hasil citra biner berbeda, namun tetap dapat melakukan pendeteksian terhadap lapangan hijau,

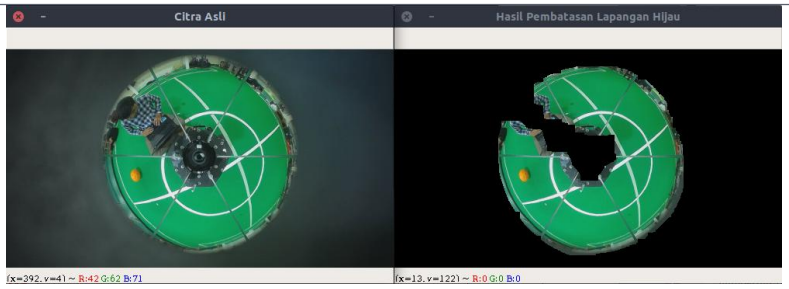
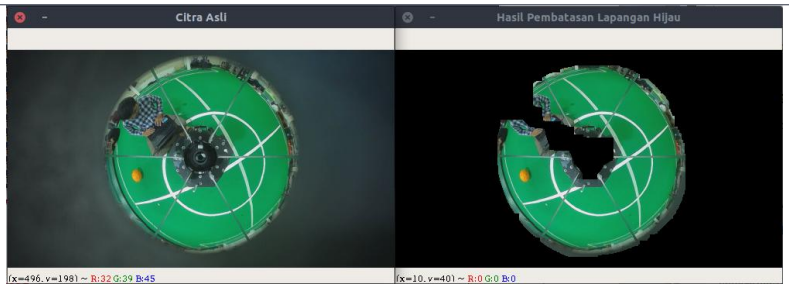
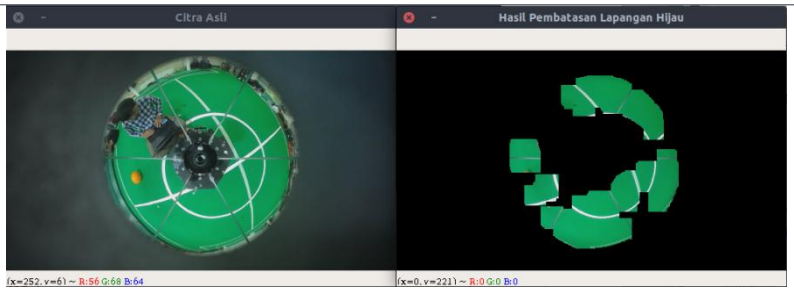
	lebih terang dibandingkan dengan lampu neon panjang yang dihidupkan dengan hasil pencahayaannya yang mereta di seluruh area lapangan namun dengan hasil citra yang memiliki kelebihan pencahayaan.		walaupun perubahan rentang nilai pencahayaan berjarak sangat jauh dengan rentang nilai yang sebelumnya.
4	pada baris ke-4 dilakukan pengujian dengan kondisi lampu spiral masih menyala terang namun pencahayaan ditambahkan dengan menghidupkan area lampu neon panjang (posisi lampu kiri lapangan)	antara 133 - 136 Cd	Hasil citra biner sama dengan hasil citra biner pada baris ke-3, dan masih berhasil mendeteksi lapangan hijau.
5	pada baris ke-5 kemudian semua lampu di area lapangan hijau dihidupkan	antara 165 - 169Cd	Hasil citra biner sama dengan hasil citra biner pada baris ke-3 dan baris ke-4, dan masih berhasil mendeteksi lapangan hijau.

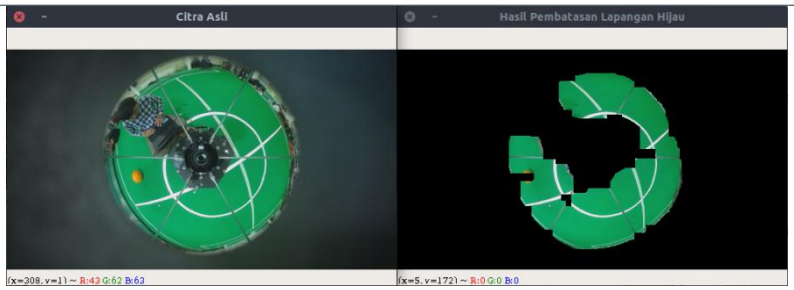
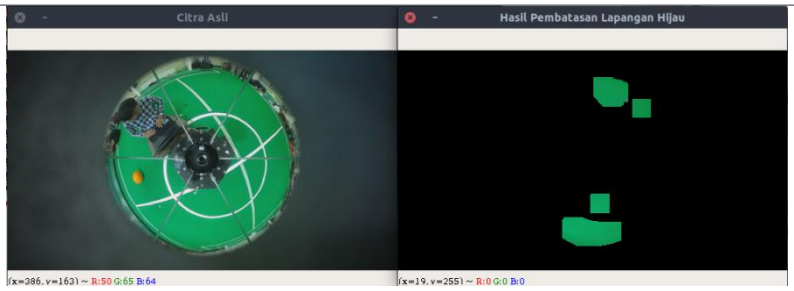
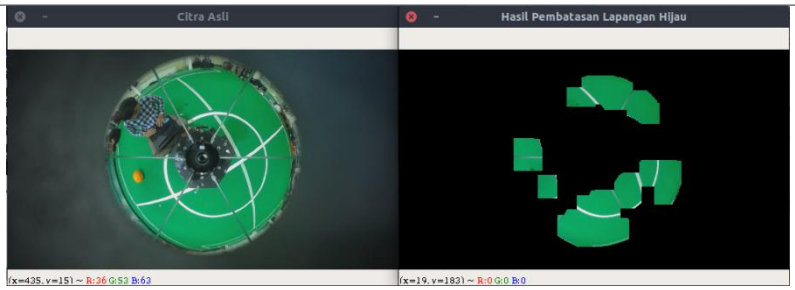
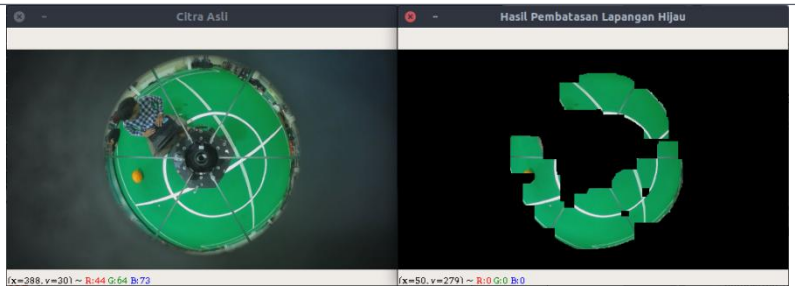
Pengujian yang kedua akan dilakukan untuk menentukan jumlah iterasi yang akan digunakan untuk proses operasi erosi dan delasi.

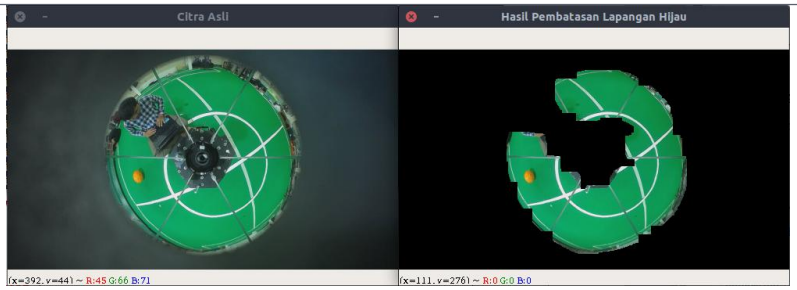
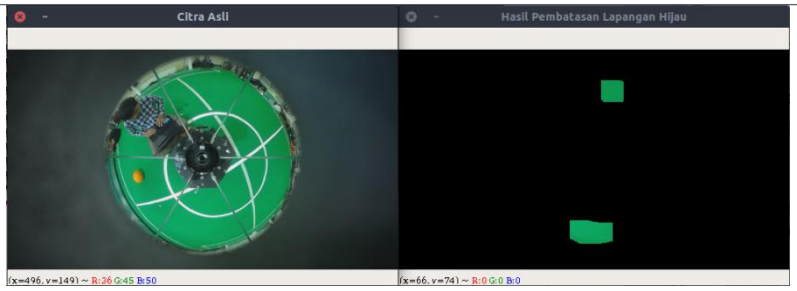
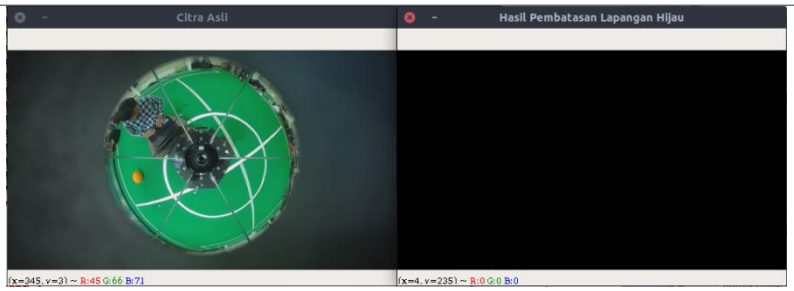
Tabel 5.3 Pengujian untuk menentukan jumlah iterasi pada operasi erosi dan dilasi

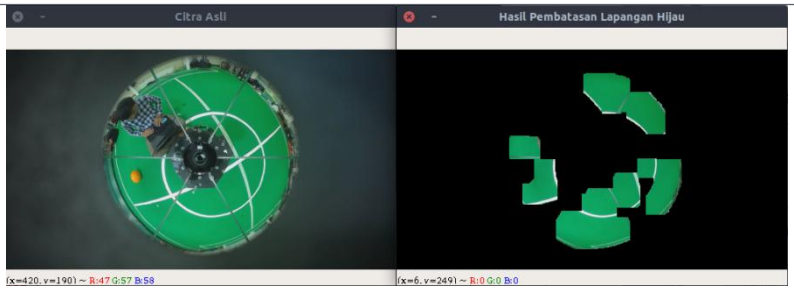
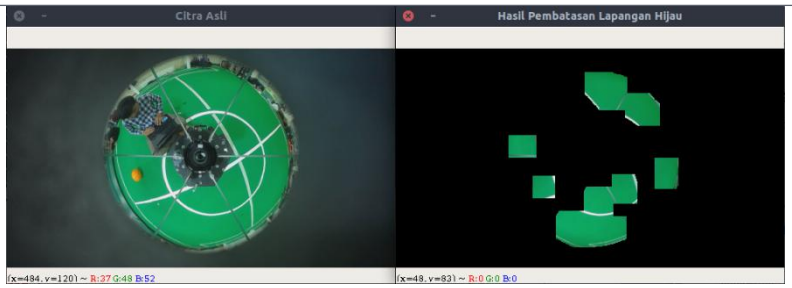
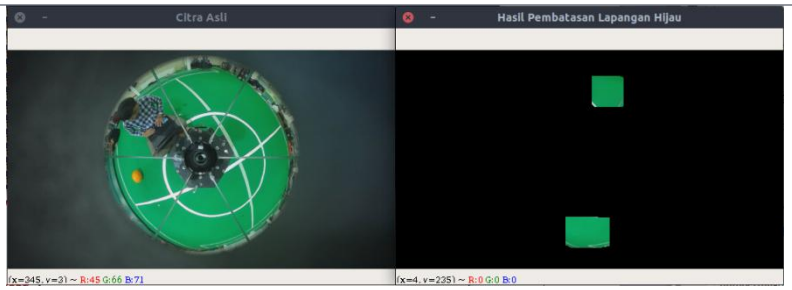
No	Citra Pengujian Jumlah Iterasi	Keterangan
1		Iterasi erosi : 1 Iterasi dilasi : 1 Status : dengan nilai ini, warna bola tidak dapat dideteksi oleh sistem, karena dihapus pada

		citra.
2		Iterasi erosi : 1 Iterasi dilasi : 2 Status : warna bola terdeteksi pada citra, namun dinding dari lapangan tidak terlihat pada citra.
3		Iterasi erosi : 1 Iterasi dilasi : 3 Status : warna bola terdeteksi pada citra, dan dinding dari lapangan juga terlihat pada citra dengan hasil yang sangat minim.
4		Iterasi erosi : 1 Iterasi dilasi : 4 Status : warna bola terdeteksi pada citra, dan dinding dari lapangan juga terlihat pada citra dengan hasil yang cukup untuk mengetahui atau mendeteksi bola jika bola berada di tepian dinding lapangan.

5		Iterasi erosi : 1 Iterasi dilasi : 5 Status : warna bola terdeteksi pada citra, dan dinding dari lapangan juga terlihat pada citra dengan hasil pembacaan citra melampaui dinding lapangan, sehingga deteksi objek di luar lapangan sangat mungkin terjadi.
6		Iterasi erosi : 1 Iterasi dilasi : 6 Status : warna bola terdeteksi pada citra, dan dinding dari lapangan juga terlihat pada citra dengan hasil pembacaan citra melampaui dinding lapangan, sehingga deteksi objek di luar lapangan sangat mungkin terjadi.
7		Iterasi erosi : 2 Iterasi dilasi : 1 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau, sehingga objek

		bola yang ada di dalam lapangan tidak terlihat.
8		Iterasi erosi : 2 Iterasi dilasi : 2 Status : dengan nilai ini, warna bola tidak dapat dideteksi oleh sistem, karena dihapus pada citra.
9		Iterasi erosi : 3 Iterasi dilasi : 1 Status : sistem hanya menampilkan sebagian kecil lapangan hijau, dan tidak menampilkan kondisi lingkungan di dalam lapangan.
10		Iterasi erosi : 3 Iterasi dilasi : 2 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau, sehingga objek bola tidak terlihat di dalam lapangan.
11		Iterasi erosi : 3 Iterasi dilasi : 3 Status : warna bola tidak dapat dideteksi oleh sistem, karena dihapus pada citra.

12		Iterasi erosi : 3 Iterasi dilasi : 5 Status : warna bola terdeteksi pada citra, dan dinding dari lapangan juga terlihat pada citra dengan hasil yang sangat minim.
13		Iterasi erosi : 4 Iterasi dilasi : 1 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.
14		Iterasi erosi : 4 Iterasi dilasi : 4 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.
15		Iterasi erosi : 5 Iterasi dilasi : 1 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.

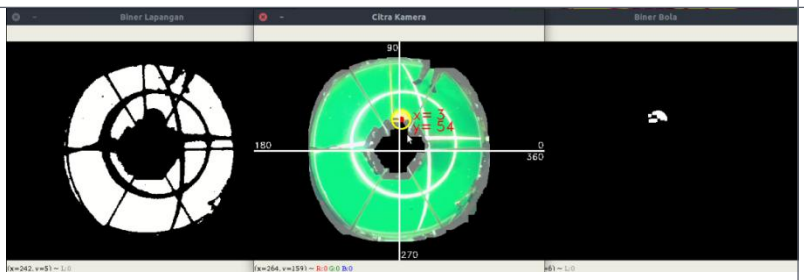
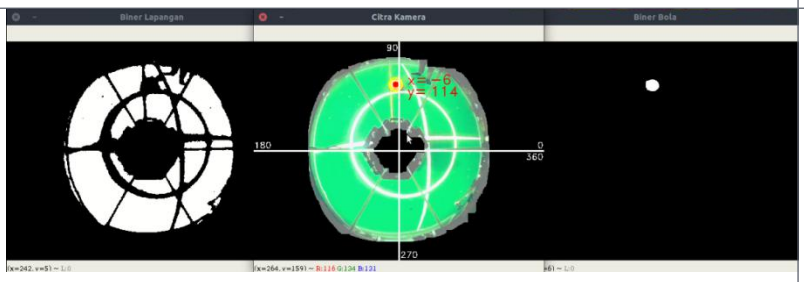
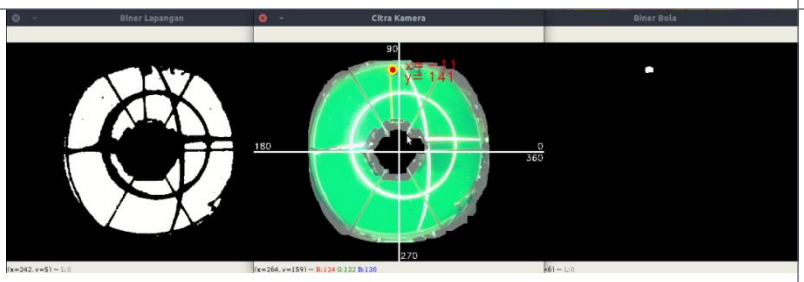
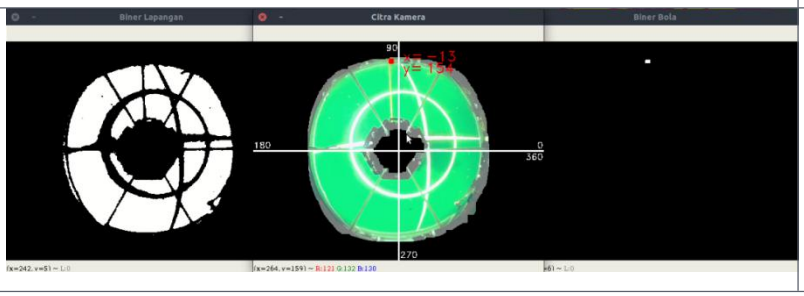
16		Iterasi erosi : 5 Iterasi dilasi : 5 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.
17		Iterasi erosi : 6 Iterasi dilasi : 6 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.
18		Iterasi erosi : 7 Iterasi dilasi : 7 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.
19		Iterasi erosi : 8 Iterasi dilasi : 8 Status : sistem tidak mampu mengembalikan ukuran dari objek lapangan berwarna hijau.

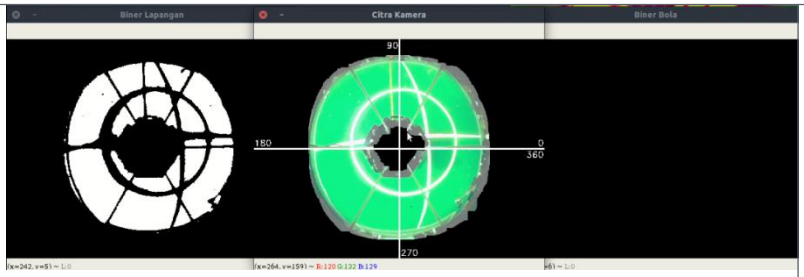
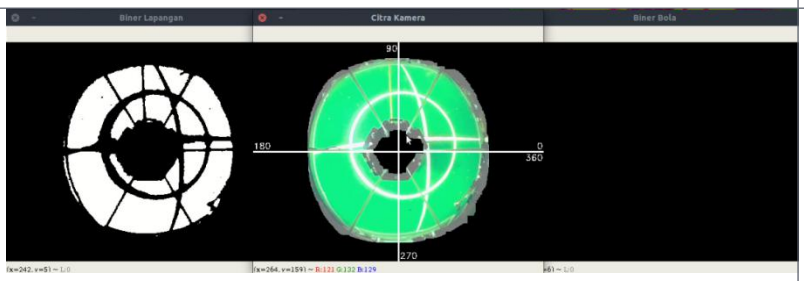
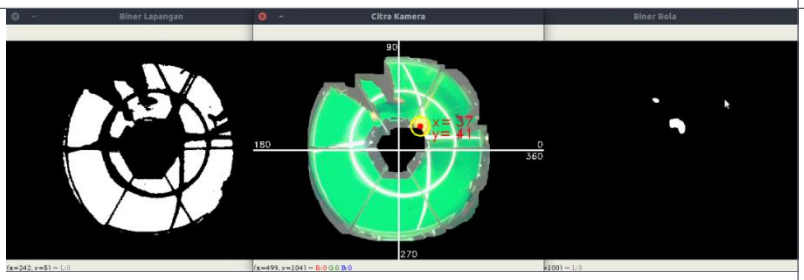
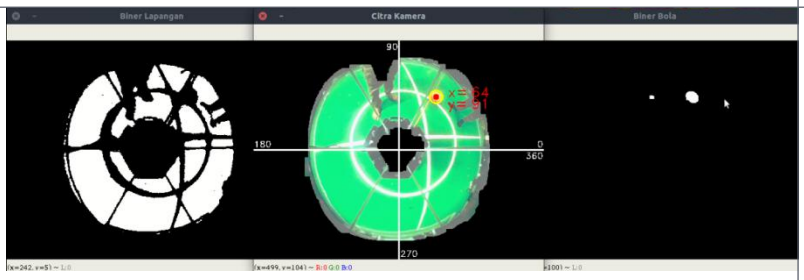
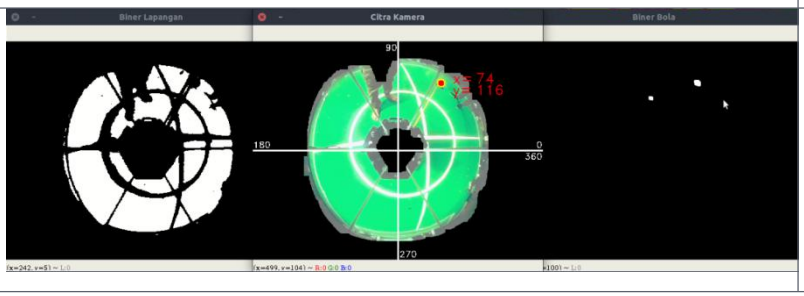
Pada pengujian di atas, terlihat bahwa jumlah iterasi yang sangat sesuai ataupun yang tepat untuk digunakan untuk melakukan pembatasan pembacaan terhadap lapangan hijau yaitu jumlah iterasi erosi sebanyak 1 kali dan jumlah iterasi dilasi sebanyak 4 kali. Dengan iterasi sebanyak itu, dihasilkan citra yang mampu menampilkan objek-objek yang ada di

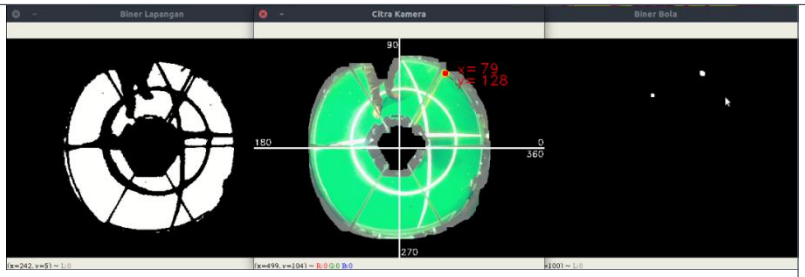
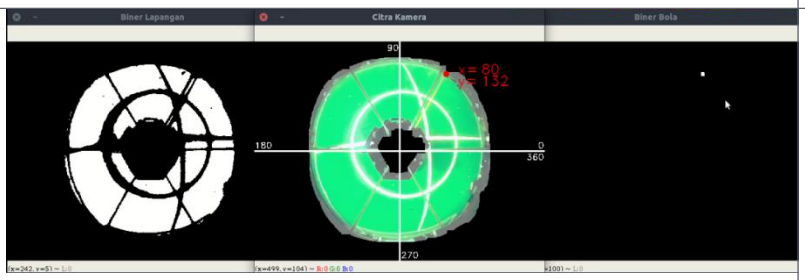
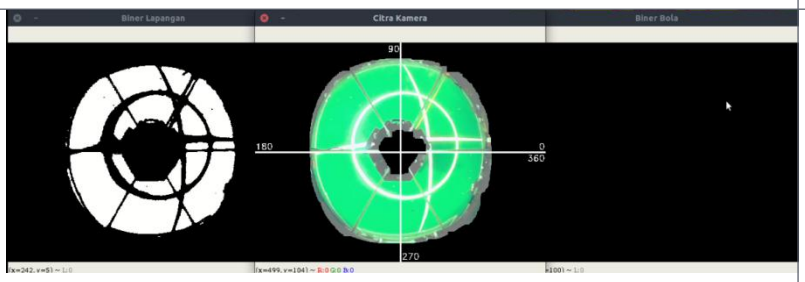
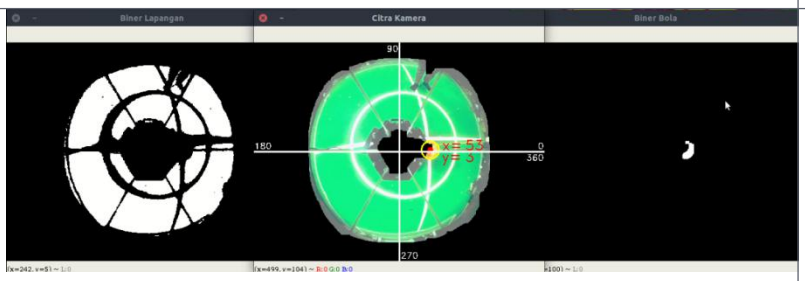
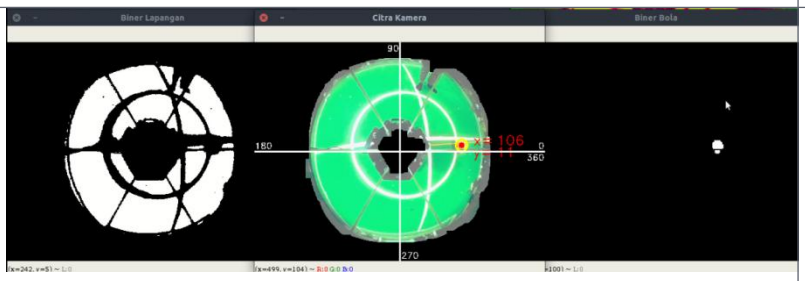
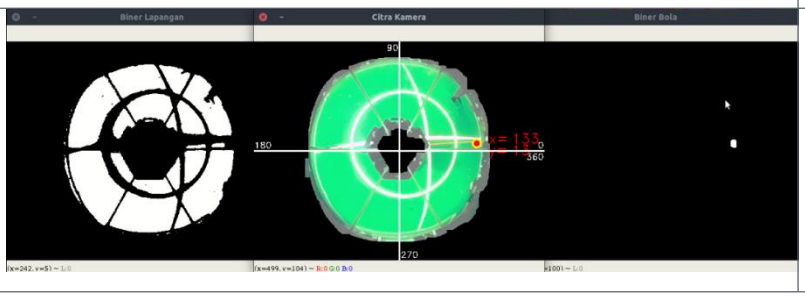
dalam lapangan hijau, salah satunya adalah bola dan mampu meminimalisir untuk pembacaan selain bola, contohnya seperti robot lawan atau manusia seperti yang terlihat pada tabel pengujian.

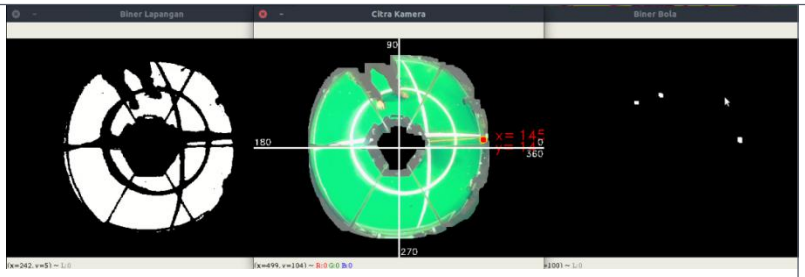
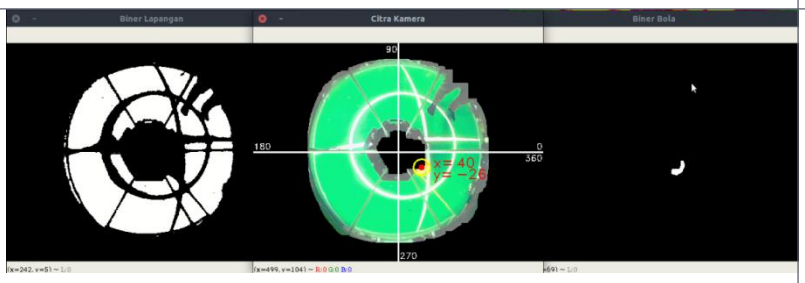
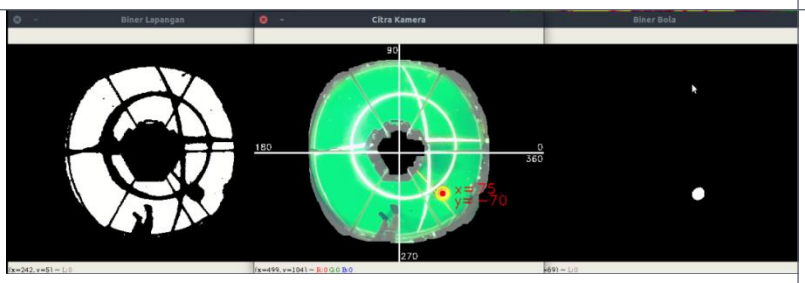
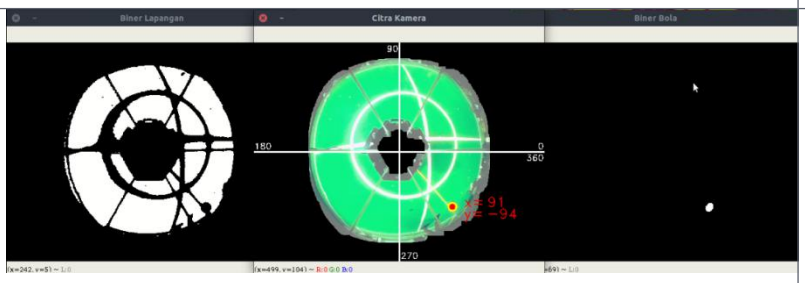
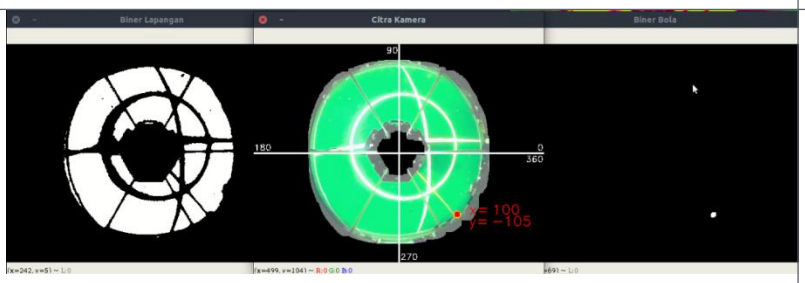
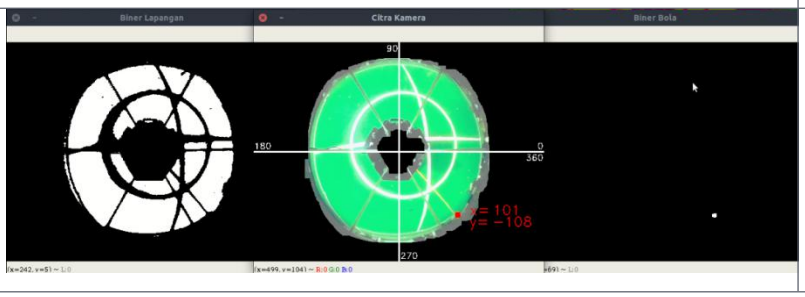
Selanjutnya akan dilakukan pengujian untuk menguji jarak tangkap yang dapat dilakukan sistem dengan hasil citra dari *omnidirectional camera*. Pengukuran jarak dilakukan dimulai dari dimensi robot ketitik tengah bola.

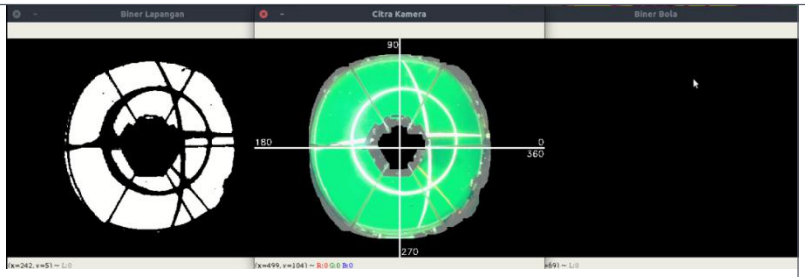
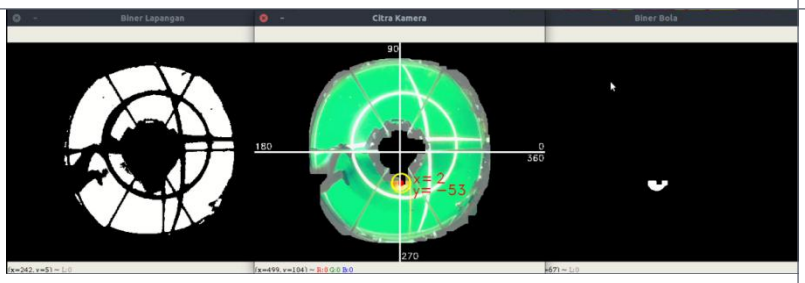
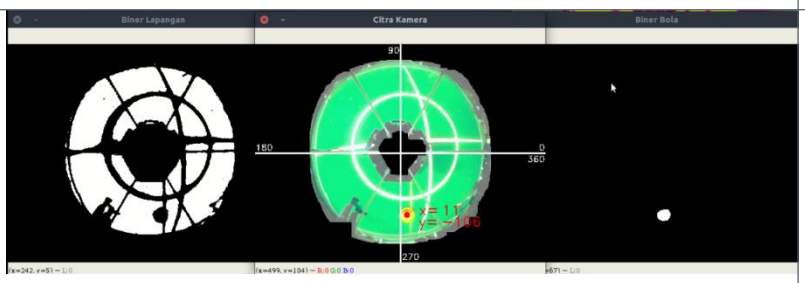
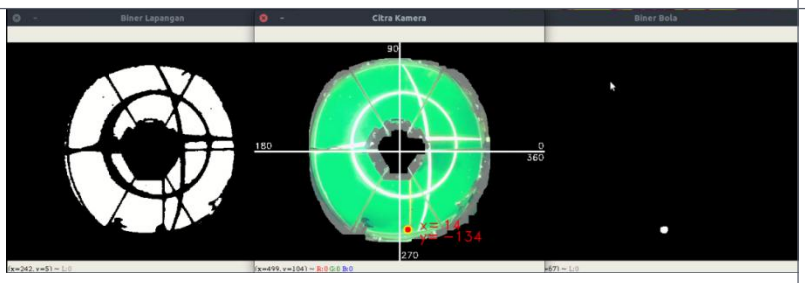
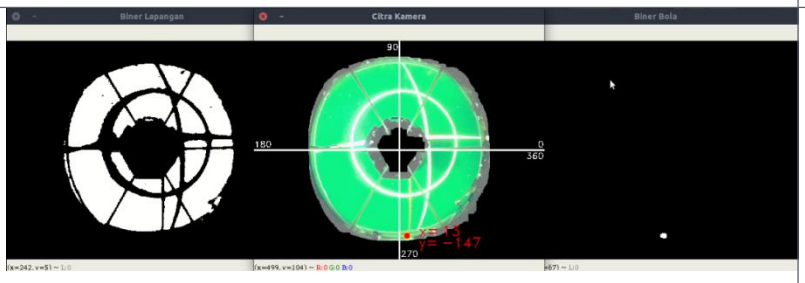
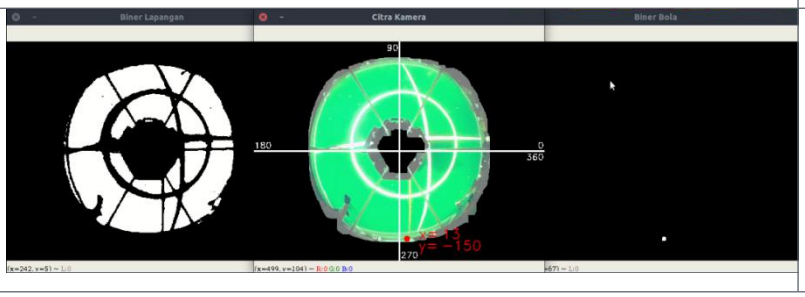
Tabel 5.4 Pengujian jarak pandang deteksi objek bola

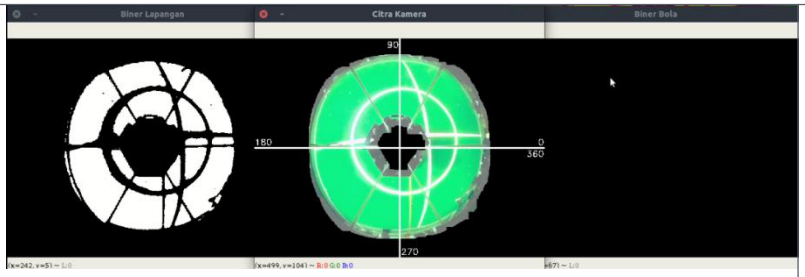
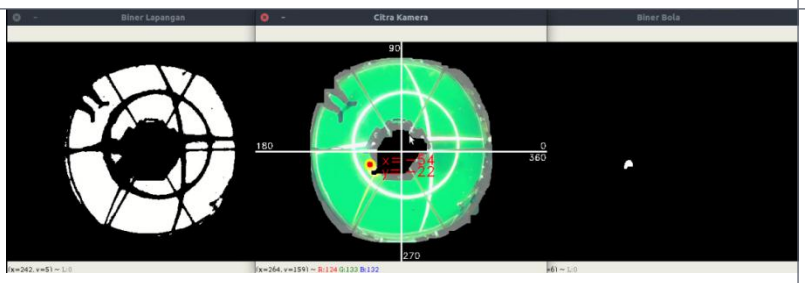
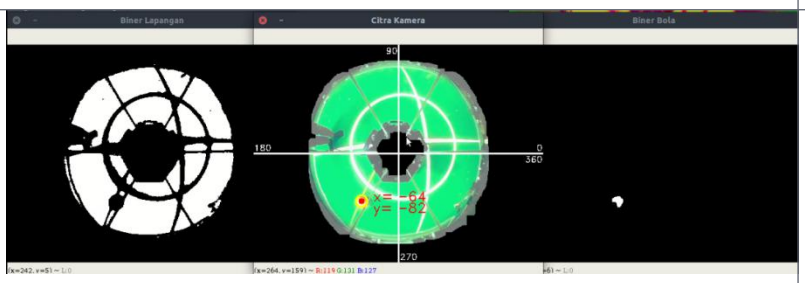
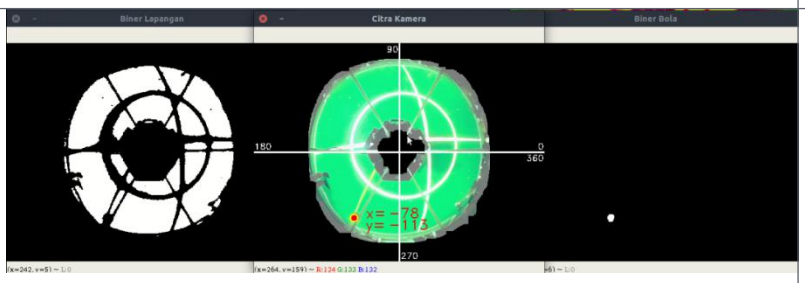
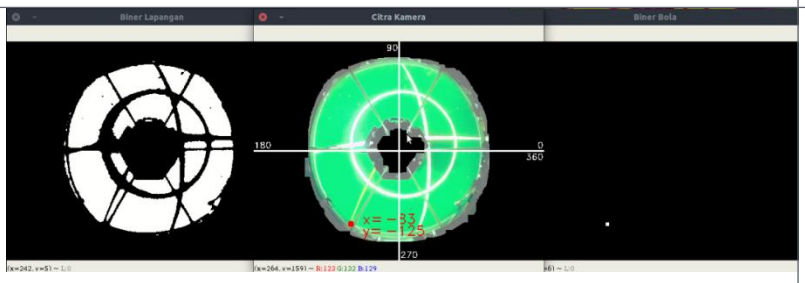
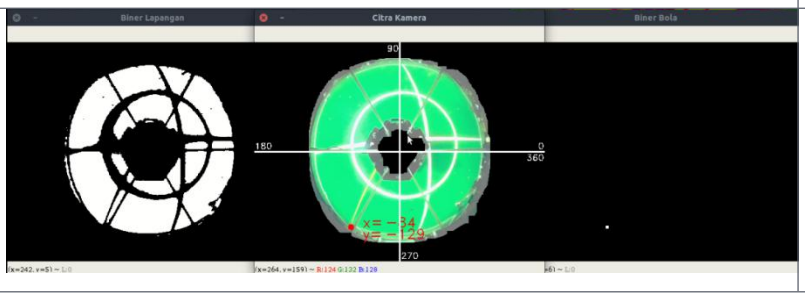
No	Citra Jarak Pandang Robot	Keterangan
	Daerah depan robot	
1		Jarak : 10cm X : 3 Y : 54 Status : bola terdeteksi
2		Jarak : 70cm X : -6 Y : 114 Status : bola terdeteksi
3		Jarak : 150cm X : -11 Y : 141 Status : bola terdeteksi
4		Jarak : 240cm X : -13 Y : 154 Status : bola terdeteksi

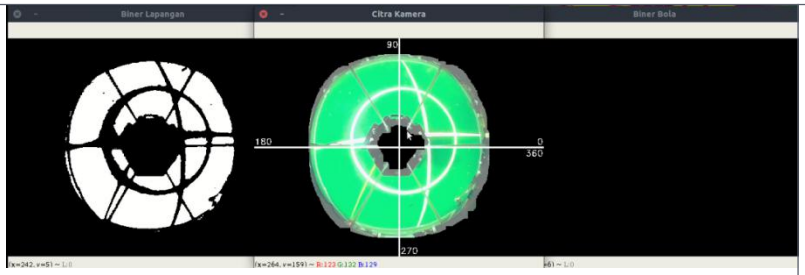
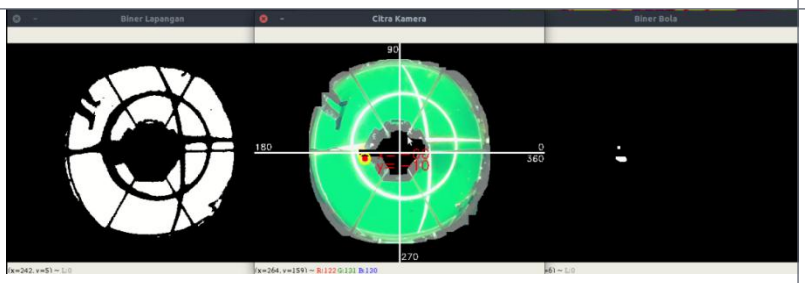
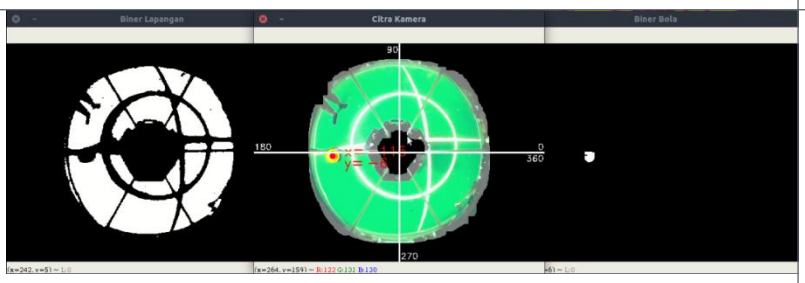
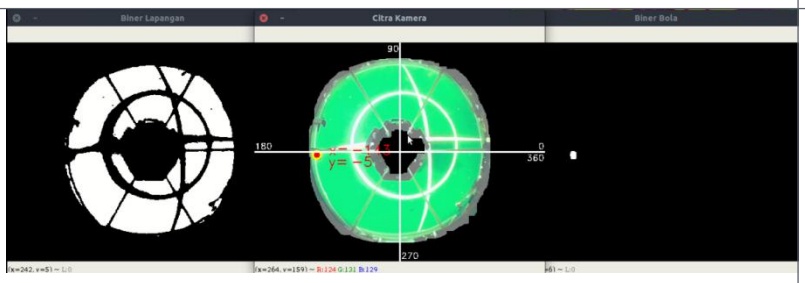
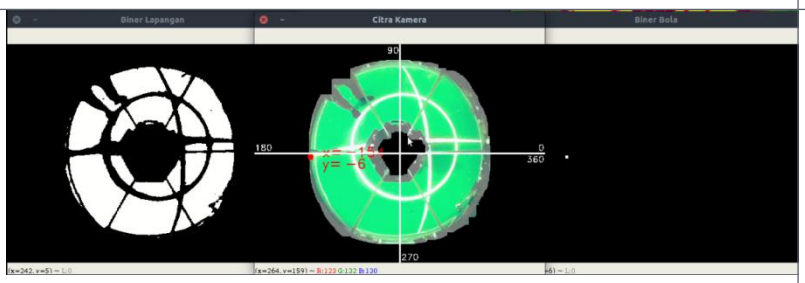
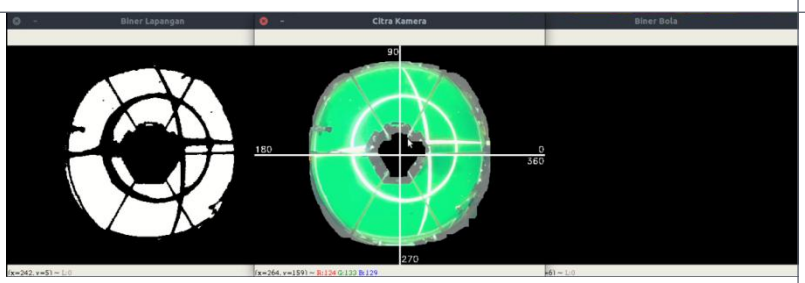
5		Jarak : 280cm X : 0 Y : 0 Status : bola tidak terdeteksi
6		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah kuadran 1	
7		Jarak : 8cm X : 37 Y : 41 Status : bola terdeteksi
8		Jarak : 70cm X : 64 Y : 91 Status : bola terdeteksi
9		Jarak : 150cm X : 74 Y : 116 Status : bola terdeteksi

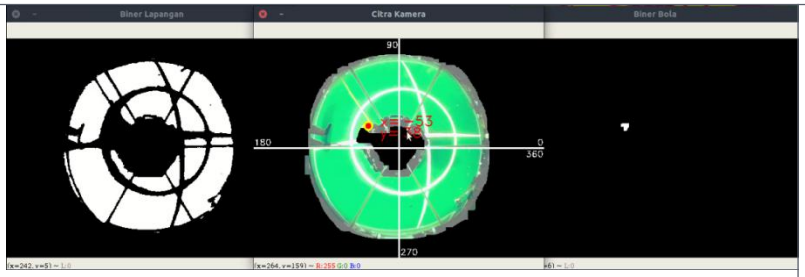
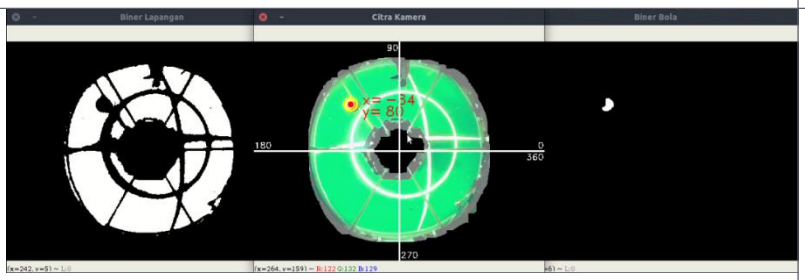
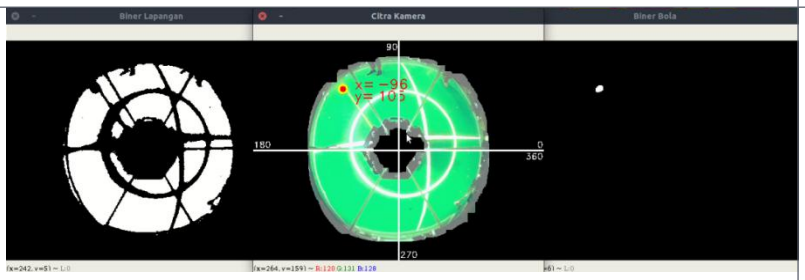
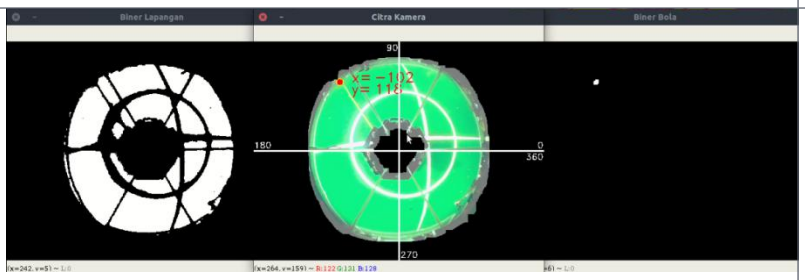
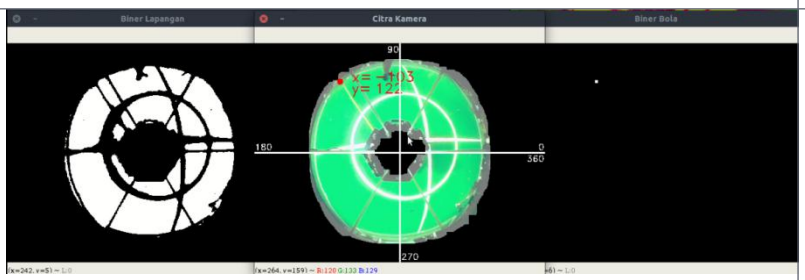
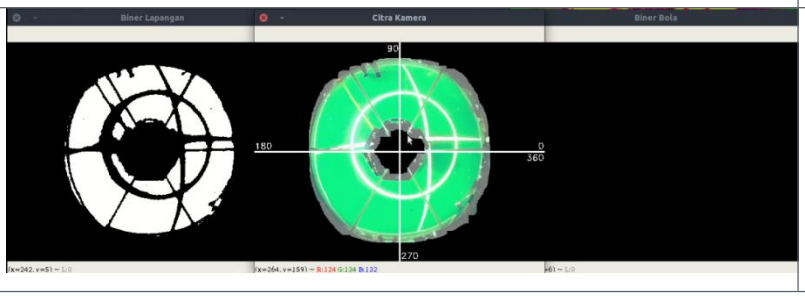
10		Jarak : 240cm X : 79 Y : 128 Status : bola terdeteksi
11		Jarak : 280cm X : 80 Y : 132 Status : bola terdeteksi
12		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah kanan robot	
13		Jarak : 10cm X : 53 Y : 3 Status : bola terdeteksi
14		Jarak : 70cm X : 106 Y : 11 Status : bola terdeteksi
15		Jarak : 150cm X : 133 Y : 13 Status : bola terdeteksi

16		Jarak : 280cm X : 145 Y : 14 Status : bola terdeteksi
	Daerah kuadran 4	
17		Jarak : 8cm X : 40 Y : -26 Status : bola terdeteksi
18		Jarak : 70cm X : 75 Y : -70 Status : bola terdeteksi
19		Jarak : 150cm X : 91 Y : -94 Status : bola terdeteksi
20		Jarak : 240cm X : 100 Y : -105 Status : bola terdeteksi
21		Jarak : 280cm X : 101 Y : -108 Status : bola terdeteksi

22		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah belakang robot	
23		Jarak : 10cm X : 2 Y : -53 Status : bola terdeteksi
24		Jarak : 70cm X : 11 Y : -106 Status : bola terdeteksi
25		Jarak : 150cm X : 14 Y : -134 Status : bola terdeteksi
26		Jarak : 240cm X : 13 Y : -147 Status : bola terdeteksi
27		Jarak : 280cm X : 13 Y : -150 Status : bola terdeteksi

28		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah kuadran 3	
29		Jarak : 10cm X : -54 Y : -22 Status : bola terdeteksi
30		Jarak : 70cm X : -64 Y : -82 Status : bola terdeteksi
31		Jarak : 150cm X : -78 Y : -113 Status : bola terdeteksi
32		Jarak : 240cm X : -83 Y : -125 Status : bola terdeteksi
33		Jarak : 280cm X : -84 Y : -129 Status : bola terdeteksi

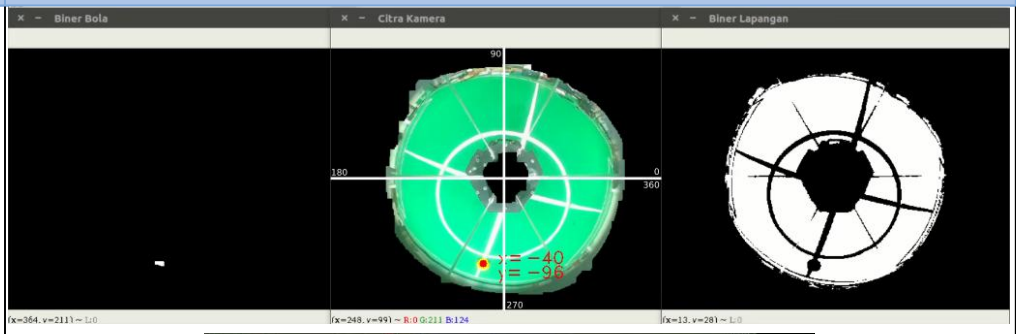
34		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah kiri robot	
35		Jarak : 7cm X : 0 Y : -10 Status : bola terdeteksi
36		Jarak : 70cm X : -115 Y : -6 Status : bola terdeteksi
37		Jarak : 150cm X : -113 Y : -5 Status : bola terdeteksi
38		Jarak : 240cm X : -151 Y : -6 Status : bola terdeteksi
39		Jarak : 280cm X : 0 Y : 0 Status : bola tidak terdeteksi
	Daerah kuadran 2	

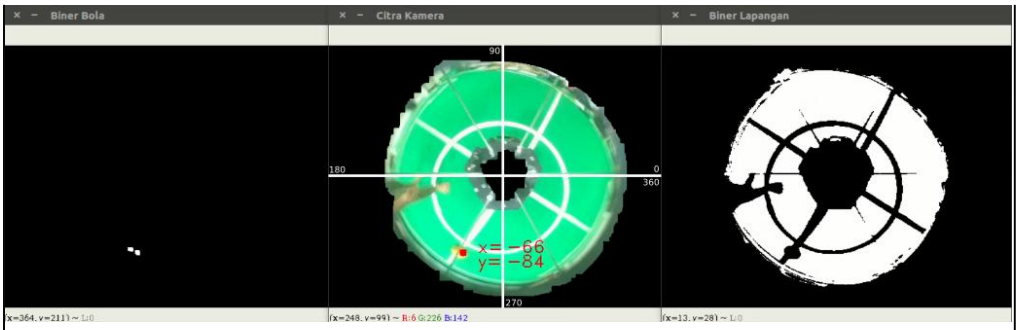
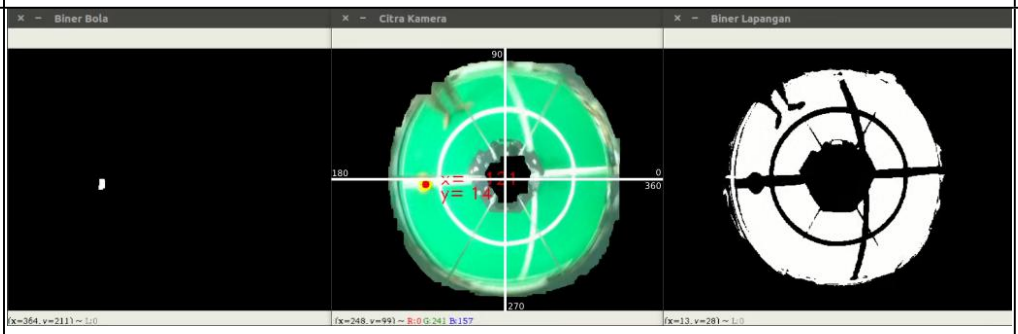
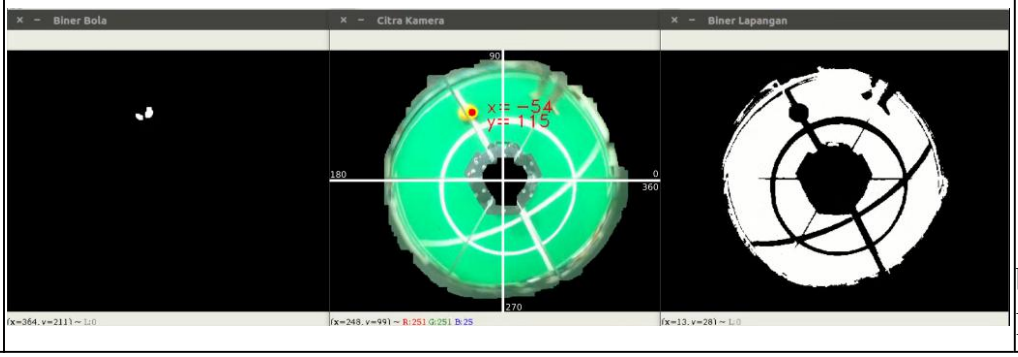
40		Jarak : 7cm X : -53 Y : 38 Status : bola terdeteksi
41		Jarak : 70cm X : -34 Y : 80 Status : bola terdeteksi
42		Jarak : 150cm X : -96 Y : 105 Status : bola terdeteksi
43		Jarak : 240cm X : -102 Y : 118 Status : bola terdeteksi
44		Jarak : 280cm X : -103 Y : 122 Status : bola terdeteksi
45		Jarak : 300cm X : 0 Y : 0 Status : bola tidak terdeteksi

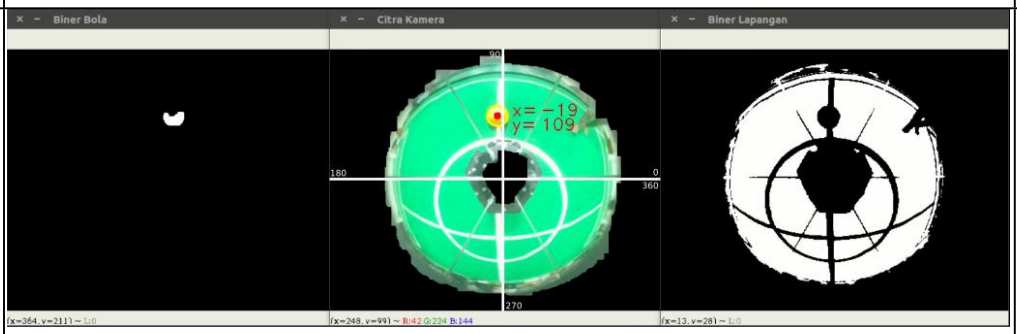
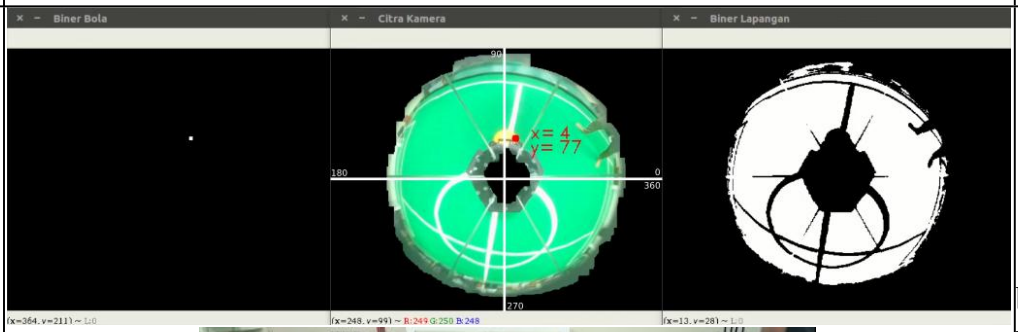
Pada pengujian jarak di atas, objek bola berhasil dideteksi dengan rata-rata jarak sejauh 240 cm. Untuk jarak pandang sendiri, beberapa hal yang sangat mempengaruhinya, yang pertama kondisi pencahayaan yang kurang merata yang ada dilapangan, hal ini akan mempengaruhi nilai rentang warna yang telah ditentukan dari hasil kalibrasi warna bola. Yang kedua adalah bentuk cermin cembung yang tidak proposional untuk dapat menghasilkan bentuk objek yang utuh pada pantulan cermin, sehingga *distorsi* bentuk yang terjadi pada pinggiran cermin mempengaruhi besarnya pembacaan warna terhadap objek (objek tidak terlihat jelas) dan akan mempengaruhi warna objek tersebut, bisa terlalu terang maupun gelap).

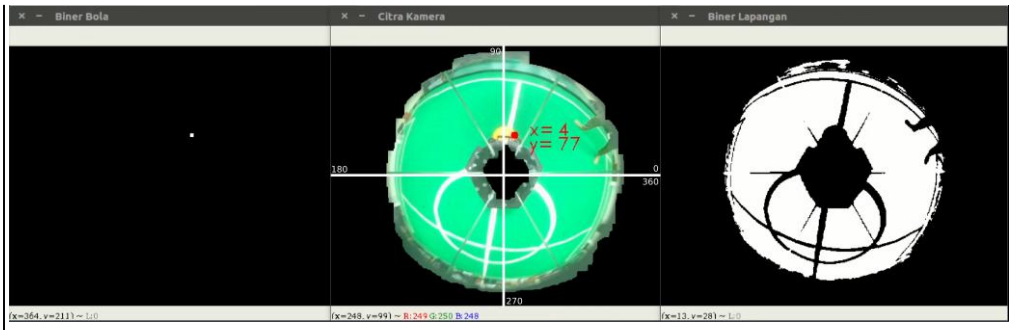
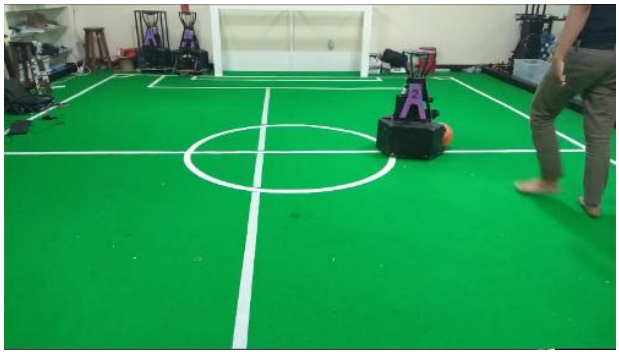
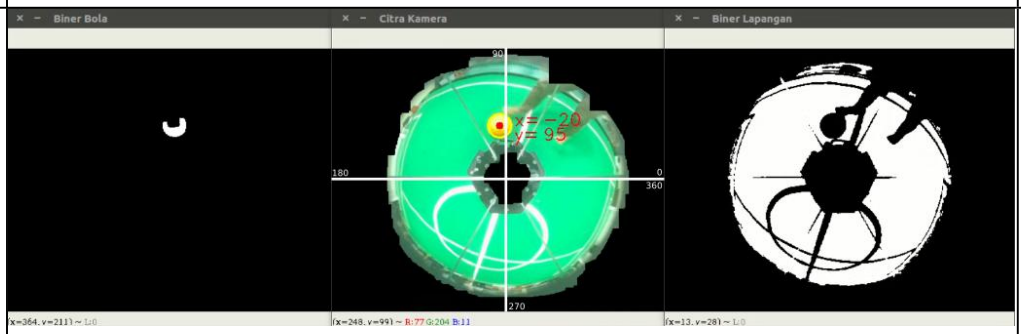
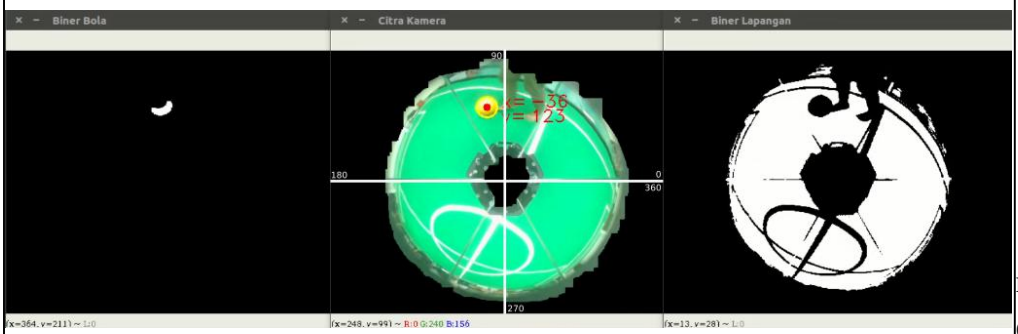
Dan yang terakhir akan dilakukan pengujian untuk memastikan bahwa penangkapan citra dan pengambilan koordinat dari citra menghasilkan nilai kecepatan dan arah gerak motor yang benar, sehingga dapat dipastikan bahwa robot menghampiri posisi dan keberadaan bola dengan tepat. Hasil dan pengujian dari sistem ini dapat dilihat pada tabel berikut.

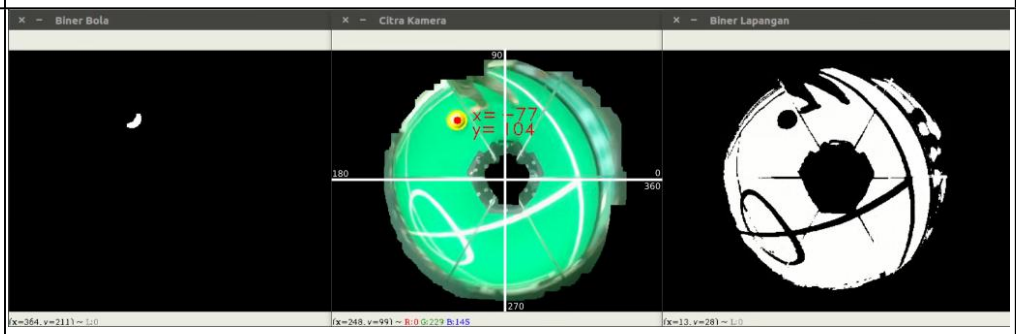
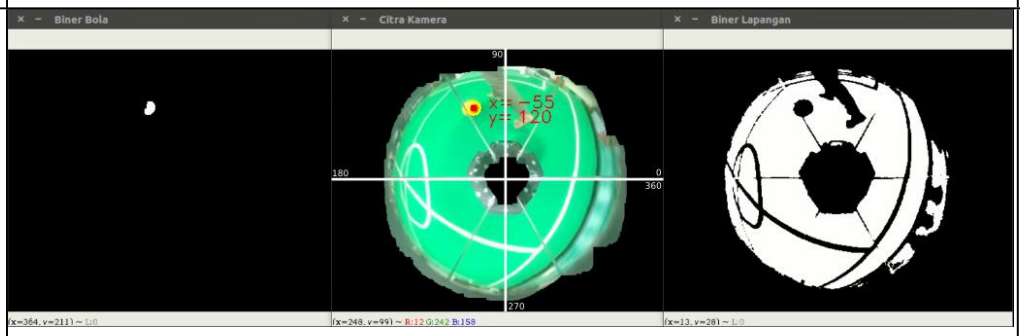
Tabel 5.5 Pengujian hasil pengolahan citra untuk pergerakan robot

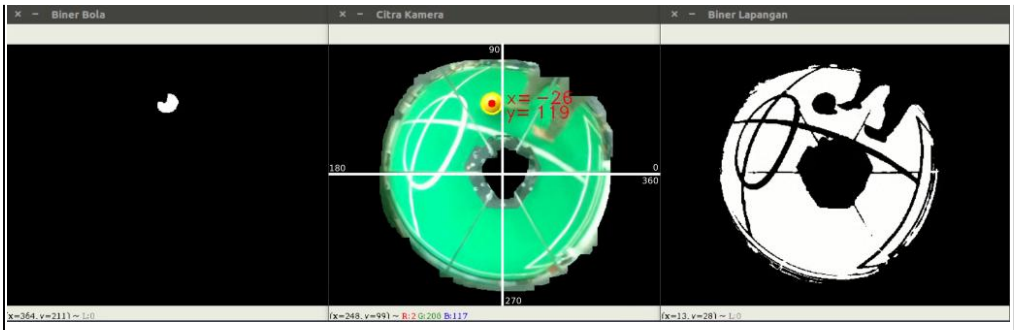
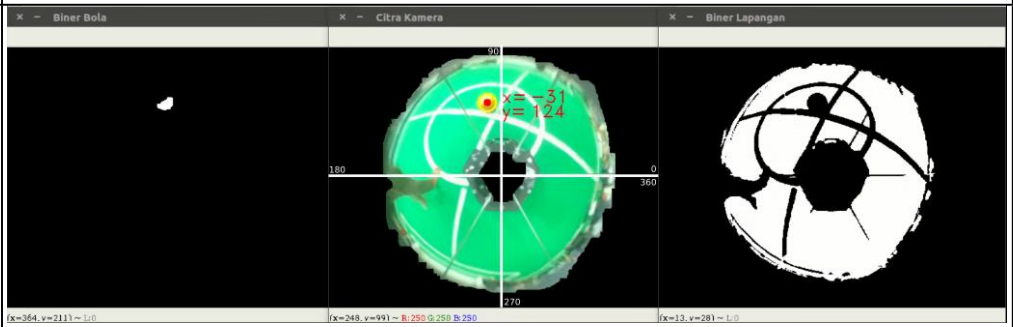
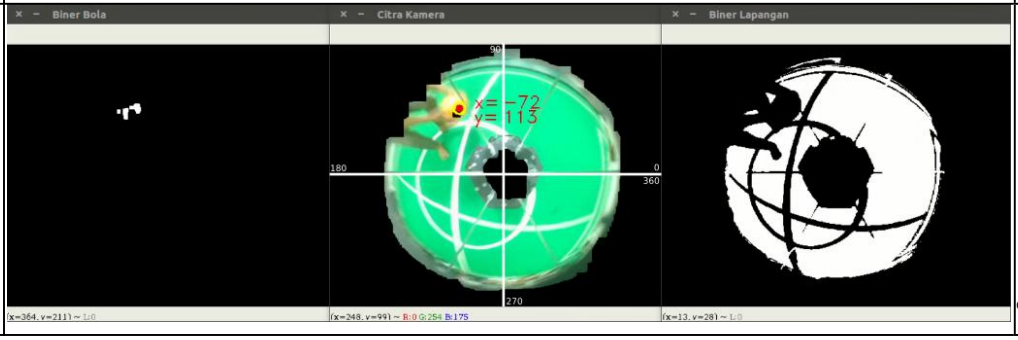
No.	Citra dan keterangan pergerakan robot dan bola	Nilai x, y, dan kecepatan Motor
1	 	X : -40 Y : -96 Motor 1 : 0 Motor 2 : 0 Motor 3 : 0 Status : robot menjalankan program berhenti

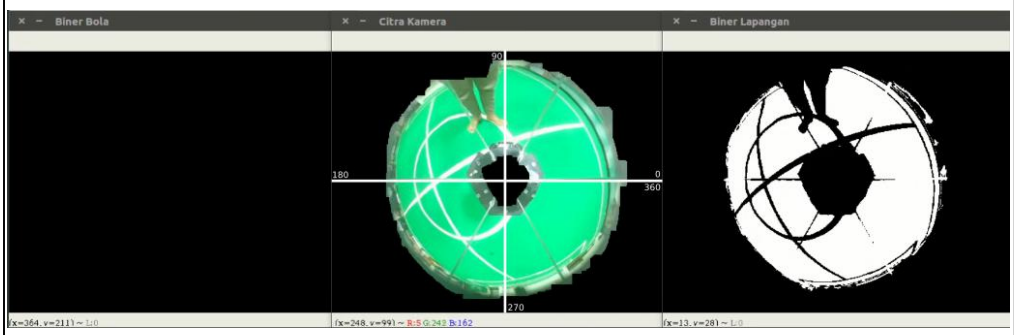
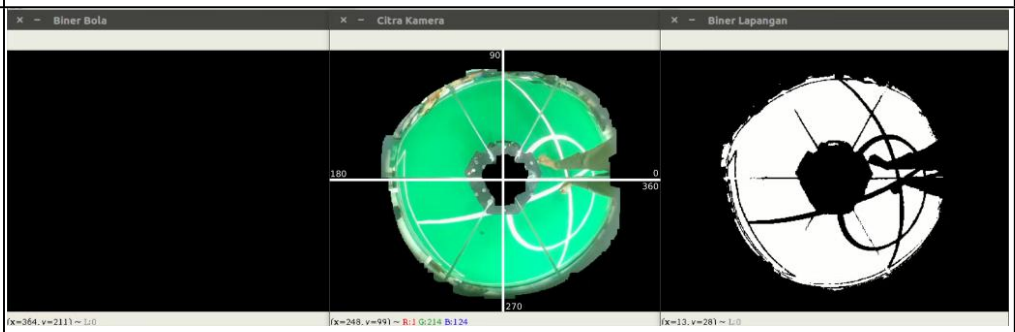
2	 	X : -66 Y : -84 Motor 1 : 137 Motor 2 : 12 Motor 3 : -8 Status : Robot mulai program tarcking ball dan berputar ke arah posisi bola
3	 	X : -121 Y : 14 Motor 1 : 41 Motor 2 : -32 Motor 3 : 96 Status : Robot bergerak memutar kearah posisi bola
4		X : -54 Y : 115 Motor 1 : -72 Motor 2 : -18 Motor 3 : 125 Status : Robot bergerak maju kearah bola

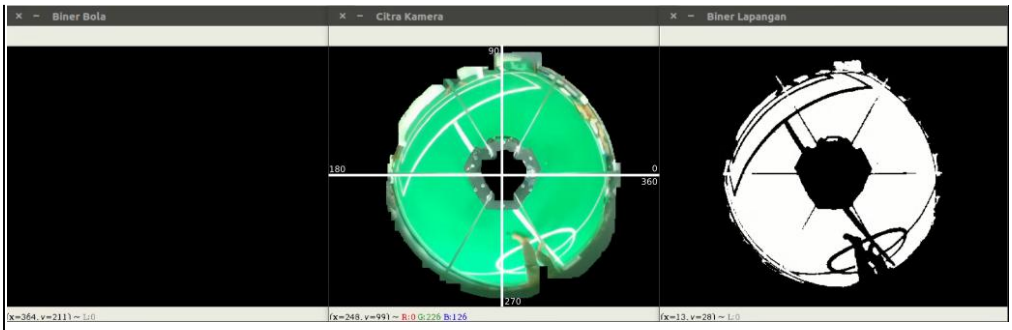
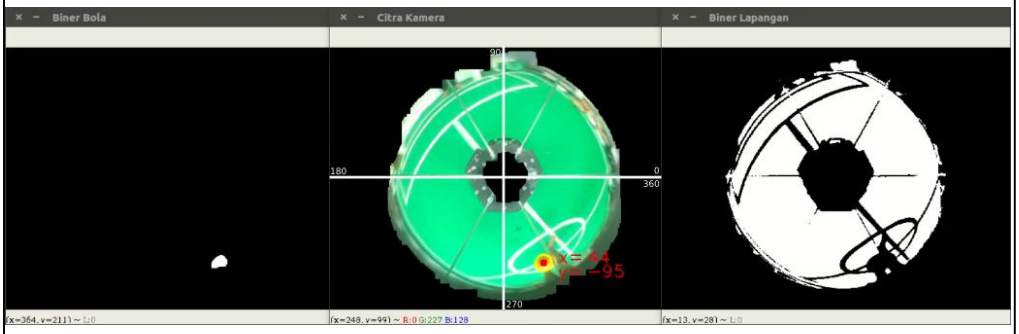
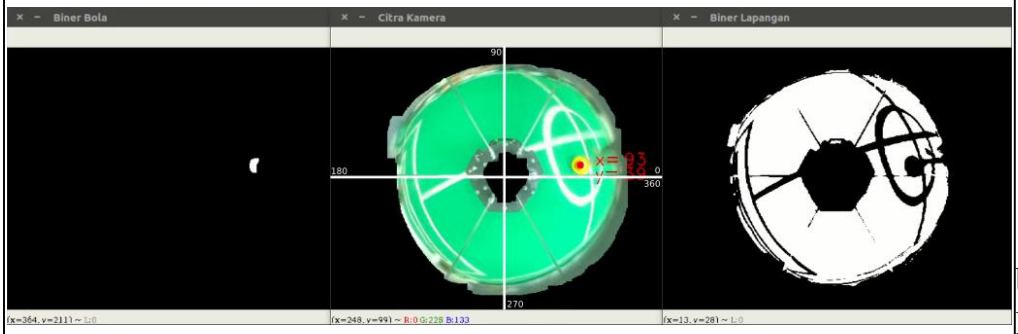
		
5	 	X : -19 Y : 109 Motor 1 : -82 Motor 2 : -5 Motor 3 : 100 Status : Robot bergerak maju kearah bola
6	 	X : 4 Y : 77 Motor 1 : 0 Motor 2 : 0 Motor 3 : 0 Status : Robot berhasil menangkap bola dan berhenti bergerak, dikarenakan pada pengujian ini dibuat agar robot tidak melakukan gerakan pada saat robot mendapatkan bola

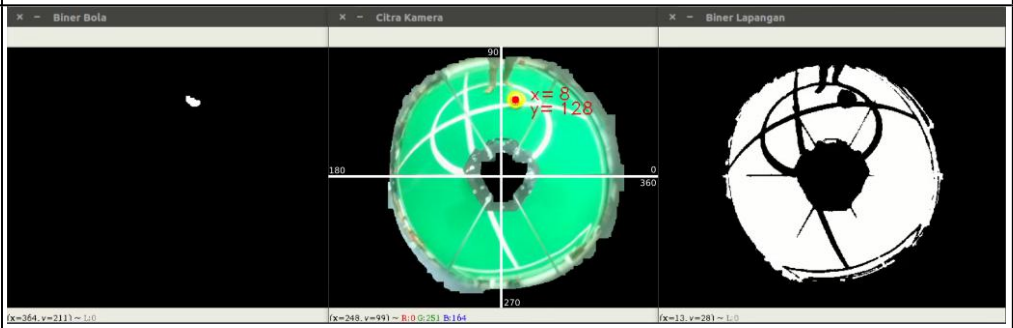
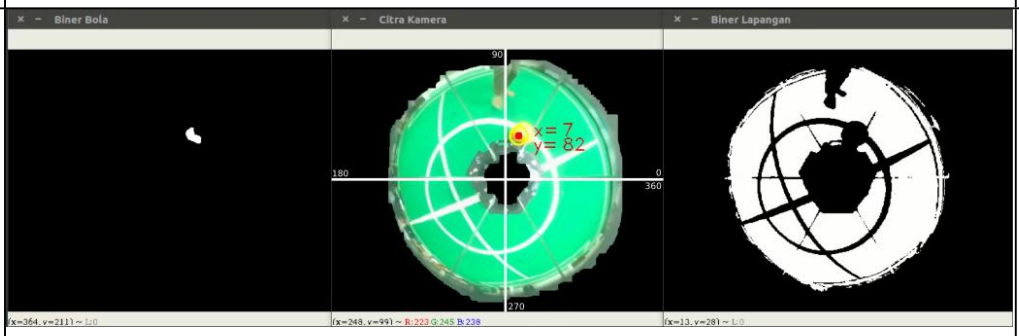
7	 	X : 4 Y : 77 Motor 1 : 0 Motor 2 : 0 Motor 3 : 0 Status : Robot masih dalam keadaan menangkap bola
8	 	X : -20 Y : 95 Motor 1 : -71 Motor 2 : -5 Motor 3 : 92 Status : Bola ditarik dari robot, dan robot kembali melacak bola dan mengejar bola
9		X : -36 Y : 123 Motor 1 : -88 Motor 2 : -12 Motor 3 : 123 Status : Robot mengejar bola yang digiring oleh penguji

		
10	 	X : -77 Y : 104 Motor 1 : -53 Motor 2 : -25 Motor 3 : 128 Status : Robot mengejar bola yang digiring oleh penguji
11	 	X : -55 Y : 120 Motor 1 : -77 Motor 2 : -19 Motor 3 : 120 Status : Robot mengejar bola yang digiring oleh penguji

12	 	X : -26 Y : 119 Motor 1 : -89 Motor 2 : -8 Motor 3 : 115 Status : Robot mengejar bola yang digiring oleh penguji
13	 	X : -31 Y : 124 Motor 1 : -91 Motor 2 : -10 Motor 3 : 121 Status : Bola ditendang oleh penguji, robot tetap mengejar bola
14		X : -72 Y : 113 Motor 1 : -57 Motor 2 : -28 Motor 3 : 135 Status : bola di angkat oleh penguji

		
15	 	X : 10 Y : 0 Motor 1 : 39 Motor 2 : 47 Motor 3 : 39 Status : Robot tidak mendeteksi keberadaan posisi bola, namun pada pengujian ini robot akan berputar secara parabola untuk mencari keberadaan bola dengan kecepatan motor seperti yang tertera di atas
16	 	X : 10 Y : 0 Motor 1 : 39 Motor 2 : 47 Motor 3 : 39 Status : Robot masih berputar secara parabola untuk mencari keberadaan bola dengan kecepatan motor yang sama

17	 	X : 10 Y : 0 Motor 1 : 39 Motor 2 : 47 Motor 3 : 39 Status : Robot masih berputar secara parabola untuk mencari keberadaan bola dengan kecepatan motor yang sama
18	 	X : 44 Y : -95 Motor 1 : 25 Motor 2 : -22 Motor 3 : -143 Status : Bola dijatuhkan oleh penguji, robot langsung mendeteksi keberadaan bola, dan bergerak putar kearah posisi bola
19		X : 93 Y : 39 Motor 1 : -94 Motor 2 : 21 Motor 3 : -15 Status : Robot bergerak putar kearah posisi bola

		
20	 	X : 8 Y : 128 Motor 1 : -113 Motor 2 : 3 Motor 3 : 106 Status : Robot bergerak menghadap bola
21	 	X : 7 Y : 82 Motor 1 : -106 Motor 2 : 2 Motor 3 : 100 Status : Robot bergerak maju kearah posisi bola

Pada baris ke-2 sampai dengan baris ke-7 terlihat bahwa robot mendeteksi posisi keberadaan bola dan bergerak untuk menghadap dan menghampiri bola sesuai dengan koordinat yang didapatkan, dan bergerak kearah bola berada. Kemudian bola diambil dan digiring oleh penguji seperti yang terlihat pada baris ke-8 sampai dengan baris ke-13, dan robot masih mampu mendeteksi bola dan langsung bergerak untuk mengejar bola. Sedangkan pada baris ke-14 sampai dengan baris ke-17, bola dihilangkan dari area lapangan hijau, sehingga robot tidak mampu mendeteksi keberadaan posisi bola dan melakukan pencarian bola atau menunggu bola terdeteksi dengan bergerak secara parabola. Terakhir pada baris ke-18 sampai dengan baris ke-21, bola dimasukkan kembali ke dalam area lapangan hijau dan diletakkan di belakang robot, kemudian robot kembali mendeteksi bola dan robot bergerak memutar untuk menghadap dan menghampiri keberadaan bola.

5.4 Kelebihan Dari Sistem Hasil Penelitian

Setelah dilakukannya beberapa tahap pengujian terhadap sistem yang menjadi objek penelitian ini, didapati beberapa kelebihan yang dapat disampaikan, diantaranya :

1. Pemanfaatan cermin cembung mampu menghasilkan citra secara *omnidirectional camera* atau citra dari segala arah, sehingga sistem mampu membaca kondisi dan keadaan disekitar robot dengan jarak rata-rata hingga 240 cm.
2. Sistem masih mampu mengenali warna objek target, dengan kondisi cahaya yang berubah-ubah dan tingkat kecerahan cahaya yang berbeda-beda.
3. Sistem mampu membatasi pembacaan untuk lapangan hijau, dengan tujuan sistem hanya akan membaca objek-objek yang berada di dalam lapangan hijau, sehingga objek-objek yang terdapat di luar lapangan hijau tidak terlihat atau terbaca pada citra.
4. Sistem mampu melacak keberadaan bola dan posisi bola secara *real-time*.
5. Mudahnya melakukan kalibrasi warna dari objek yang menjadi target deteksi objek dengan menggunakan *trackbar*.

5.5 Kekurangan dari Sistem

Kekurangan yang muncul dari hasil pengujian terhadap sistem, yaitu terdapat pada teknik pengolahan citra yang digunakan masih menggunakan pengolahan citra dasar, ialah teknik *color-filtering*, hal ini menimbulkan munculnya beberapa kekurangan pada sistem, di antaranya :

1. sistem belum mampu melakukan pengenalan terhadap objek berdasarkan bentuk dari objek itu sendiri. Hal ini didasari oleh dua hal, yang pertama terjadi karena penggunaan warna sebagai indikator pengenalan objek, sehingga objek yang memiliki warna yang hampir sama bisa di kenali sebagai objek target, dan yang kedua diakibatkan oleh *distorsi*-nya citra yang dihasilkan menggunakan *omnidirectional camera* dan membuat bentuk dari citra objek tidak sesuai dengan bentuk aslinya.
2. Pengenalan warna objek dengan pengaruh tingkat pencahayaan yang berbeda, memiliki hasil deteksi objek yang kurang maksimal, sehingga warna dengan rentang yang sama dengan objek target masih terdeteksi.
3. Jarak pandang dari citra yang dihasilkan menggunakan cermin pada robot tidak terlalu jauh, dan juga *distorsi* yang dihasilkan pada pinggiran citra sangat besar, dan sangat mempengaruhi warna dari objek yang akan dideteksi.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian di atas, dapat diambil kesimpulan sebagai berikut :

1. Sistem mampu mendeteksi objek bola dengan menggunakan teknik pengambilan gambar *omnidirectional camera*, dan mendapatkan titik koordinat posisi keberadaan bola dilapangan, sehingga robot mampu bergerak kearah objek bola berada.
2. Sistem kalibrasi warna bekerja dengan baik, hal ini terlihat dari rentang nilai warna hasil kalibrasi warna yang berhasil digunakan untuk mendeteksi objek pada sistem pengolahan citra atau sistem *tracking ball*.
3. Teknik pengambilan citra dengan *omnidirectional camera* yang memanfaatkan cermin cembung menghasilkan citra yang terbalik, hal ini mempengaruhi koordinat objek yang dideteksi dan jarak pandang dari citra yang dihasilkan.
4. Untuk jumlah iterasi pada proses operasi erosi dan dilasi, akan digunakan iterasi sebanyak 1 kali untuk operasi erosi dan iterasi sebanyak 4 kali untuk operasi dilasi, jumlah iterasi ini mampu menampilkan citra yang membatasi pembacaan hanya di dalam area lapangan hijau, dan area yang ada di luar lapangan tidak terlihat pada citra, selain itu bola masih dapat dideteksi di dalam lapangan.

6.2 Saran

Dari hasil penelitian yang dilakukan, sistem ini masih memiliki banyak kekurangan dan dimungkinkan untuk dilakukan pengembangan lebih lanjut lagi, diantaranya sebagai berikut :

1. Penggunaan cermin cembung khusus yang lebih proposional agar menghasilkan citra dengan bentuk objek yang minim *distorsi*.
2. Pengembangan untuk sistem pembacaan bentuk dari objek, dengan menggunakan teknik *omnidirectional camera* dengan metode " *camera calibration*".
3. Penggunaan metode "*select area"* untuk sistem kalibrasi warna, agar lebih mudah dalam menentukan warna objek hanya dengan memilih area atau warna yang akan menjadi target deteksi objek.

DAFTAR PUSTAKA

Mahardika, P, N. (2015). Identifikasi Basic Shape dengn Metode Harris Corner Detection pada Raspberry PI. Skripsi, tidak dipublikasikan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Setiawan, M.T., & Firdausy, K. (2016). Aplikasi Penghitung Telur Ikan Gurami menggunakan Deteksi Blob berbasis Android. Seminar Nasional Aplikasi Teknologi Informasi(SNATi). Yogyakarta.

Asyrofi, M., H. (2017). Penerapan Algoritma Greedy untuk Mendeteksi Bola pada Robot Sepak Bola Beroda. Makalah IF221 Strategi Algoritma. Bandung : Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

OpenCV team. (2018). OpenCV. Diakses dari https://opencv.org/about.html

OpenCV team. (2018). Morphological Transformations. Diakses dari https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

OpenCV team. (2018). Contours : Getting Started. Diakses dari https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html

Wikipedia. (2018). Warna. Diakses dari https://id.wikipedia.org/wiki/Warna

Kadir, A., & Susanto, A. (2013). Teori dan Aplikasi Pengolahan Citra. Yogyakarta : CV.ANDI OFFSET

Heinemann, P. (2007). Cooperative Multi-Robot Soccer in a Highly Dynamic Environment. Disertasi. Tübingen : Universitas Tübingen

Setiaji, B, W, B. (2014). Robot Pemadam Api Berkaki. Skripsi, tidak dipublikasikan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Arduino. (2018). What is Arduino?. Diakses dari https://www.arduino.cc/en/Guide/Introduction

Onny. (2014). Prinsip Kerja Motor Listrik. Diakses dari https://artikel-teknologi.com/prinsip-kerja-motor-listrik/

Muliady., & Arisandy, G. (2017). Implementasi Sistem Gerak *Holonomic* pada Robot KRSBI Beroda 2017. Jurnal Teknik dan Ilmu Komputer. Bandung : Universitas Kristen Maranatha.

Suryawan, D. (2014). Perancangan Robot Sepak Bola Beroda Berbasis Android. Skripsi, tidak dipublikasikan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia.

Affandi, I., Sulistijono, I, A., & Ardilla, F. (2014). Kontrol Posisi Robot Omni-Directional Menggunakan Metode Gyrodometry. Simposium KRSBI. Surabaya : Politeknik Elektronika Negeri Surabaya.

Gonzalez, R, C., & Woods, R, E. (2014). Digital Image Processing, Third Edition. India : Pearson.

Hendrayawan, V. (2014). Implementasi Invers Kinematics Pada Sistem Pergerakan Mobile Robot Roda Mekanum. Publikasi Jurnal Skripsi. Malang : Universitas Brawijaya.

Mahesa, J., Azizul., & Meliyana. (2016). Rancang Bangun Sistem Tracking Pada Robot Sepak Bola Dengan Metode Thresholding Menggunakan Open-CV. In M. Udin Harun Al Rasyid and Fahim Nur Cahya Bagar. ISRSC 2016 : The 4th Indonesian Symposium on Robot Soccer Competition, Gedung Pascasarjana Politeknik Elektronika Negeri Surabaya, Surabaya, 1 juni (ISBN: 978-602-72251-2-1). Surabaya: Politeknik Elektronika Negeri Surabaya.

Fajar, S, D. (2017). Robot Soccer Beroda Berbasis Raspberry Pi 3 Sebagai Prototype ERSBIB 2017. Skripsi, tidak dipublikasikan. Yogyakarta : Universitas Sanata Dharma

LAMPIRAN