

**DETEKSI JARAK BOLA PADA ROBOT KIPER SEPAK BOLA
MENGUNAKAN *HOUGH CIRCLE TRANSFORMATION*
BERBASIS RASPBERRY PI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Tunggal Manda Ary Triyono
NIM: 145150301111014



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

DETEKSI JARAK BOLA PADA ROBOT KIPER SEPAK BOLA MENGGUNAKAN *HOUGH CIRCLE TRANSFORMATION* BERBASIS RASPBERRY PI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Tunggal Manda Ary Triyono
NIM: 145150301111014

Skripsi ini telah diuji dan dinyatakan lulus pada
7 November 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Hurriyatul Fitriyah, S.T, M.Sc
NIP: 19851001 201504 2 003

Mochammad Hannats Hanafi Ichsan, S.ST, M.T
NIK:201405 881229 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 November 2018



Tunggal Manda Ary Triyono

NIM: 145150301111014

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan atas rahmat dan karunia-Nya Tuhan yang Maha Esa, sehingga penulis dapat menyelesaikan skripsinya dengan judul “DETEKSI JARAK BOLA PADA ROBOT KIPER SEPAK BOLA MENGGUNAKAN *HOUGH CIRCLE TRANSFORMATION* BERBASIS RASPBERRY PI”.

Penulis menyadari bahwa pembuatan skripsi ini tidak lepas dari bantuan baik motivasi, doa dan jasa dari berbagai pihak. Maka sebab itu, penulis menyampaikan rasa hormat dan terimakasih kepada:

1. Bapak Muyono dan Ibu Trimah selaku orang tua penulis yang selalu memberikan motivasi dan semangat serta doa sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. Selaku Ketua Prodi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Ibu Hurriyatul Fitriyah, S.T, M.Sc selaku dosen pembimbing 1 yang telah memberikan arahan dan bimbingannya kepada penulis dalam menyelesaikan skripsi ini.
5. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T selaku dosen pembimbing 2 yang telah memberikan arahan dan bimbingannya kepada penulis dalam menyelesaikan laporan skripsi ini.
6. Syahrul Yoga Pradana dan Yoga Sukma Pradana yang telah membantu secara teknis dalam pengerjaan skripsi ini
7. Teman-teman Komunitas Robotiik yang telah membantu dan mendukung penulis dalam mengerjakan skripsi ini.
8. Alwinda Fischa Chalysta yang selalu menyemangati dan membantu penulis baik secara teknis maupun non teknis dalam proses pengerjaan skripsi ini.
9. Teman-teman Grup Lawas dan Grup Mungil yang telah memberikan dukungan dan motivasi kepada penulis.
10. Teman-teman tercinta program studi Teknik Komputer angkatan 2014 yang telah memberikan semangat dan dukungan kepada penulis.

Penulis menyadari bahwa dalam laporan ini masih banyak kekurangan, oleh sebab itu penulis berharap adanya pengembangan lebih lanjut oleh pihak-pihak terkait. Semoga laporan ini dapat memberikan manfaat dan referensi untuk melakukan penelitian sebagai langkah penyempurnaan sistem.

Malang, 7 November 2018

Penulis

tunggalmat@gmail.com



ABSTRAK

Robot kiper Sepak Bola merupakan robot yang sangat penting dalam tim robot Sepak Bola yang mana robot kiper harus mampu melakukan aksi penangkapan bola secara benar dan efektif. Untuk membuat aksi dari robot kiper benar dan efektif, diperlukan kemampuan robot kiper untuk mengetahui jarak antara dirinya dan bola. Untuk itu dalam penelitian ini, dibuat sistem deteksi jarak bola dengan kamera pada robot kiper sepak bola menggunakan *Hough Circle Transformation*. Penelitian ini menggunakan kamera *webcam* Logitech C525 dan Raspberry Pi 3 Model B sebagai input dan outputnya ditampilkan pada layar monitor. Perhitungan jarak dari bola menggunakan perbandingan diameter bola yang terdeteksi sistem. Citra yang diambil oleh kamera akan diolah sampai mendapatkan citra biner dengan *noise* yang minimum. Kemudian dengan menggunakan *Hough Circle Transformation*, akan didapatkan koordinat dan diameter bola yang ada pada lapangan. Data diameter objek bola yang terdeteksi inilah yang akan dibandingkan dengan diameter sebenarnya untuk menghitung jarak antara robot kiper dan bola. Selain itu, koordinat bola juga dapat dimanfaatkan untuk mengetahui posisi dan arah sudut bola. Berdasarkan pengujian, sudut pandang terbaik kamera untuk pendeteksian jarak dan sudut arah bola adalah $47,2^\circ$. Akurasi deteksi jarak bola adalah 97,69 % dan eror pendeteksian sebesar 2,57 cm. Akurasi deteksi sudut arah bola adalah 94,69 % dan eror pendeteksian sebesar $1,06^\circ$. Waktu rata-rata komputasi pendeteksian jarak bola ini adalah 158,54 ms.

Kata Kunci : robot kiper, *Hough Circle Transformation*, deteksi jarak bola.

ABSTRACT

The goalkeeper robot is very important player in the team of soccer robot which must be able to catch the ball correctly and effectively. To make the action is real, the robot need a good ability to know about the distance between the ball and itself. Therefore, inside this research, made a ball distance detection system with camera on the goalkeeper robot using Hough Circle Transformation. This research used webcam Logitech C525 and Raspberry Pi 3 Model B as the input, and the output is displayed on monitor screen . The distance calculation from the ball using diameter comparison that detected by system. The image which is taken by camera will be processed till get a binary image with minimum noise. Then by using Hough Circle Transformation method, will get the coordinate and diameter of ball in the field. Diameter data in the ball that detected will be compared with the real diameter to count the distance between goalkeeper robot and the ball. Besides that, the ball coordinate can also be used to know the position and angle direction of the ball. Based on the test, the best angle of view in camera for distance and angle of direction ball detection is $47,2^\circ$. The accuracy of ball detection is 97,69 % and error of ball detection is 2,57 cm. The accuracy of angle detection is 94,69 % and error of ball detection is $1,06^\circ$. Time average computation time in this ball distance detection ball is 158,54 ms.

Keywords: *goalkeeper robot, Hough Circle Transformation, ball distance detection*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 Robot Kiper Sepak Bola	7
2.2.2 <i>Hough Circle Transformation</i>	8
2.2.3 Kamera <i>Webcam</i>	9
2.2.4 Raspberry Pi	10
2.2.5 OpenCV	11
BAB 3 METODOLOGI	12
3.1 Metodologi Penelitian	12
3.2 Studi Literatur	13
3.3 Rekayasa Kebutuhan Sistem.....	13
3.3.1 Gambaran Umum Sistem	13
3.3.2 Kebutuhan Fungsional.....	13
3.3.3 Kebutuhan Non Fungsional.....	14



3.4 Perancangan	14
3.4.1 Perancangan Perangkat Keras	14
3.4.2 Perancangan Perangkat Lunak.....	14
3.5 Implementasi Sistem	14
3.6 Pengujian dan Analisis	14
3.7 Kesimpulan.....	15
BAB 4 Rekayasa Kebutuhan	16
4.1 Gambaran Umum Sistem.....	16
4.2 Analisis Kebutuhan Sistem.....	17
4.2.1 Kebutuhan Fungsional.....	17
4.2.2 Kebutuhan Non Fungsional.....	17
4.2.2.1 Kebutuhan Perangkat Keras.....	17
4.2.2.2 Kebutuhan Perangkat Lunak.....	18
4.3 Batasan Desain Sistem	18
BAB 5 PERANCANGAN DAN IMPLEMENTASI	19
5.1 Perancangan Sistem.....	19
5.1.1 Perancangan <i>Prototype</i> Alat Pendeteksi Jarak Bola	19
5.1.2 Perancangan Perangkat Keras	21
5.1.3 Perancangan Perangkat Lunak.....	22
5.1.3.1 Perancangan Proses Akuisisi Citra	22
5.1.3.2 Perancangan Preprocessing Citra	24
5.1.3.3 Perancangan Hough Circle Transformation.....	26
5.1.3.4 Perancangan Data Diameter Objek Pada Jarak Referensi	28
5.1.3.5 Perancangan Perhitungan Jarak Bola.....	29
5.1.3.6 Perancangan Perhitungan Sudut Arah Bola.....	33
5.1.3.7 Perancangan Menampilkan Jarak dan Sudut Arah Bola pada Frame	34
5.2 Implementasi Sistem	34
5.2.1 Implementasi <i>Prototype</i> Alat Pendeteksi Jarak Bola	35
5.2.2 Implementasi Perangkat Keras	35
5.2.3 Implementasi Perangkat Lunak.....	36
5.2.3.1 Implementasi Kode Program Proses Akuisisi Data Citra.	38

5.2.3.2 Implementasi Kode Program Preprocessing Citra	39
5.2.3.3 Implementasi Kode Program Hough Circle Transformation	40
5.2.3.4 Implementasi Kode Program Persamaan Linier Jarak Referensi	41
5.2.3.5 Implementasi Kode Program Perhitungan Jarak Bola	42
5.2.3.6 Implementasi Kode Program Perhitungan Sudut Arah Bola	44
5.2.3.7 Implementasi Kode Program Menampilkan Jarak dan Sudut Arah Bola pada Frame	45
BAB 6 PENGUJIAN DAN ANALISIS	47
6.1 Pengujian Raspberry Pi Mengakses <i>Webcam</i> Logitech C525	47
6.1.1 Tujuan Pengujian.....	47
6.1.2 Prosedur Pengujian	47
6.1.3 Hasil Pengujian	48
6.1.4 Analisis Hasil Pengujian.....	49
6.2 Pengujian Akurasi Hasil Pendeteksian Jarak Bola.....	50
6.2.1 Tujuan Pengujian.....	50
6.2.2 Prosedur Pengujian	50
6.2.3 Hasil Pengujian	55
6.2.4 Analisis Hasil Pengujian.....	58
6.3 Pengujian Akurasi Sudut Arah Bola	59
6.3.1 Tujuan Pengujian.....	59
6.3.2 Prosedur Pengujian	59
6.3.3 Hasil Pengujian	62
6.3.4 Analisis Hasil Pengujian.....	63
6.4 Pengujian Waktu Pemrosesan Sistem	64
6.4.1 Tujuan Pengujian.....	64
6.4.2 Prosedur Pengujian	64
6.4.3 Hasil Pengujian	64
6.4.4 Analisis Hasil Pengujian.....	66
BAB 7 KESIMPULAN DAN SARAN	68
7.1 Kesimpulan.....	68



7.2 Saran	68
DAFTAR PUSTAKA	70
LAMPIRAN A DATA POSISI DAN DIAMETER BOLA PADA JARAK REFERENSI	72
A.1 Data Posisi dan Diameter Bola pada Jarak Referensi 50 cm	72
A.2 Data Posisi dan Diameter Bola pada Jarak Referensi 100 cm	72
A.3 Data Posisi dan Diameter Bola pada Jarak Referensi 150 cm	72
A.4 Data Posisi dan Diameter Bola pada Jarak Referensi 200 cm	73
A.5 Data Posisi dan Diameter Bola pada Jarak Referensi 250 cm	73
A.6 Data Posisi dan Diameter Bola pada Jarak Referensi 300 cm	73
LAMPIRAN B SAMPEL FRAME HASIL PENGUJIAN	74
B.1 Sampel <i>Frame</i> Pengujian Jarak dengan Sudut Pandang Kamera 49,2°74	
B.2 Sampel <i>Frame</i> Pengujian Jarak dengan Sudut Pandang Kamera 47,2°77	



DAFTAR TABEL

Tabel 2.1 Spesifikasi Logitech C525	9
Tabel 5.1 Kode Program Inisialisasi Library Deteksi Jarak Bola	36
Tabel 5.2 Kode Program Inisialisasi Semua Variabel Global	36
Tabel 5.3 Kode Program Proses Akuisisi Data Citra	38
Tabel 5.4 Kode Program <i>Preprocessing</i> Citra	39
Tabel 5.5 Kode Program <i>Hough Circle Transformation</i>	40
Tabel 5.6 Kode Program Persamaan Linier Jarak Referensi	41
Tabel 5.7 Kode Program Fungsi Perhitungan Jarak Bola	42
Tabel 5.8 Kode Program Pemanggilan fungsi perhitungan jarak bola.....	43
Tabel 5.9 Kode Program Fungsi Perhitungan Sudut Arah Bola.....	44
Tabel 5.10 Kode Program pemanggilan Fungsi Perhitungan Sudut Arah Bola.....	44
Tabel 5.11 Kode Program Menampilkan Jarak dan Sudut Arah Bola Pada <i>Frame</i>	45
Tabel 6.1 Hasil pengujian Raspberry Pi mengakses kamera <i>Webcam</i> Logitech C525.....	48
Tabel 6.2 Hasil pengujian jarak dengan sudut pandang kamera 49,2°.....	55
Tabel 6.3 Hasil pengujian jarak dengan sudut pandang kamera 47,2°.....	56
Tabel 6.4 Hasil pengujian sudut arah bola dengan sudut pandang kamera 49,2°	62
Tabel 6.5 Hasil pengujian sudut arah bola dengan sudut pandang kamera 47,2°	62
Tabel 6.6 Waktu eksekusi program pada pengujian jarak dengan sudut pandang kamera 49,2°	64
Tabel 6.7 Waktu eksekusi program pada pengujian jarak dengan sudut pandang kamera 47,2°	65

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi penelitian pertama	5
Gambar 2.2 Gambar Penelitian kedua	6
Gambar 2.3 Gambar Penelitian Ketiga	6
Gambar 2.4 Robot KRSBI Humanoid	7
Gambar 2.5 Logitech C525 HD Webcam	9
Gambar 2.6 Raspberry Pi	10
Gambar 2.7 Logo OpenCV	11
Gambar 3.1 Diagram Alur Metodologi Penelitian	12
Gambar 4.1 Blok Diagram Sistem	16
Gambar 5.1 Gambar Robot Kiper	19
Gambar 5.2 <i>Prototype</i> Alat Pendeteksi Jarak Bola	20
Gambar 5.3 Robot Tampak Samping	21
Gambar 5.4 <i>prototype</i> alat pendeteksi jarak bola tampak samping	21
Gambar 5.5 Perancangan Perangkat Keras Deteksi Jarak Bola	21
Gambar 5.6 Flowchart Perancangan Perangkat Lunak	22
Gambar 5.7 Flowchart Akuisisi Data Citra	23
Gambar 5.8 Diagram Alir Perancangan <i>Preprocessing</i> Citra	24
Gambar 5.9 <i>Structure element ellipse</i> 5 x 5	26
Gambar 5.10 Diagram Alir Perancangan <i>Hough Circle Transformation</i>	26
Gambar 5.11 Diagram Alir Pencarian Data Diameter Objek Pada Jarak Referensi	28
Gambar 5.12 Diagram Alir Perhitungan Jarak Bola	30
Gambar 5.13 Contoh Frame Untuk Perhitungan Jarak Bola	31
Gambar 5.14 Diagram Alir Perhitungan Sudut Arah Bola	33
Gambar 5.15 Diagram Alir Perancangan Menampilkan Jarak Dan Sudut Arah Bola Pada <i>Frame</i>	34
Gambar 5.16 Implementasi <i>Prototype</i> Alat Pendeteksi Jarak Bola	35
Gambar 5.17 Implementasi Hubungan Raspberry Pi Dengan Webcam	35
Gambar 6.1 Sudut pandang kamera pengujian jarak skema 1	51
Gambar 6.2 Sudut pandang kamera pengujian jarak skema 2	51
Gambar 6.3 Nilai batas Y pada pengujian jarak dengan sudut pandang kamera 49,2°	51
Gambar 6.4 Nilai batas Y pada pengujian jarak dengan sudut pandang kamera 47,2°	52
Gambar 6.5 Posisi bola pada pengujian jarak dengan sudut pandang kamera 49,2°	53
Gambar 6.6 Posisi bola pada pengujian jarak dengan sudut pandang kamera 47,2°	54
Gambar 6.7 Bagian lapangan yang dapat dideteksi jaraknya pada pengujian jarak dengan sudut pandang kamera 49,2°	56



Gambar 6.8 Bagian lapangan yang dapat dideteksi jaraknya pada pengujian jarak dengan sudut pandang kamera 47,2° 58

Gambar 6.9 Posisi bola pada pengujian sudut arah bola dengan sudut pandang kamera 49,2° 60

Gambar 6.10 Posisi bola pada pengujian sudut arah bola dengan sudut pandang kamera 47,2° 61



DAFTAR LAMPIRAN

LAMPIRAN A DATA POSISI DAN DIAMETER BOLA PADA JARAK REFERENSI	72
A.1 Data Posisi Dan Diameter Bola Pada Jarak Referensi 50 cm	72
A.2 Data Posisi Dan Diameter Bola Pada Jarak Referensi 100 cm	72
A.3 Data Posisi Dan Diameter Bola Pada Jarak Referensi 150 cm	72
A.4 Data Posisi Dan Diameter Bola Pada Jarak Referensi 200 cm	73
A.5 Data Posisi Dan Diameter Bola Pada Jarak Referensi 250 cm	73
A.6 Data Posisi Dan Diameter Bola Pada Jarak Referensi 300 cm	73
LAMPIRAN B SAMPEL FRAME HASIL PENGUJIAN	74
B.1 Sampel Frame Hasil Pengujian Jarak dengan sudut pandang kamera 49,2°	74
B.2 Sampel Frame Hasil Pengujian Jarak dengan sudut pandang kamera 47,2°	77



BAB 1 PENDAHULUAN

Pada bab ini memaparkan terkait latar belakang permasalahan sehingga dijadikan sebagai bahan penelitian, rumusan masalah, tujuan dari penelitian, manfaat penelitian yang dilakukan, batasan masalah serta sistematika pembahasan

1.1 Latar Belakang

Dunia Teknologi saat ini berkembang sangat pesat, khususnya di dunia pengolahan citra digital. Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti pengenalan pola, pengindraan jarak jauh melalui satelit atau pesawat udara, dan *machine vision* (Dwi, 2016). Banyak bidang yang membutuhkan pengolahan citra digital, salah satunya adalah di bidang robotika. Salah satu jenis robot yang sangat membutuhkan pengolahan citra digital adalah robot sepak bola. Robot sepak bola (*robosoccer*) adalah sebuah robot yang dirancang agar dapat bermain sepak bola seperti layaknya tim sepak bola manusia. Kemampuan yang dimiliki oleh sebuah *robosoccer* adalah identifikasi terhadap bola, pemain dalam satu tim, pemain lawan, dan gawang (Najmurokhman, et al., 2016). Salah satu robot yang memiliki banyak masalah dalam kemampuannya sebagai *robosoccer* adalah robot yang berposisi sebagai kiper atau penjaga gawang. Kendala terbesar robot kiper adalah bagaimana robot dapat menangkap bola dengan benar dan efektif. Saat ini kemampuan robot kiper yang dimiliki oleh tim robot Fakultas Ilmu Komputer UB hanya sebatas mendeteksi ada tidaknya bola untuk ditangkap. Hal ini tentu belum cukup untuk menjadi kiper yang baik. Oleh karena itu, diperlukan suatu kemampuan robot kiper untuk mendeteksi jarak dari bola.

Sensor yang digunakan untuk mendeteksi bola adalah kamera. Kamera inilah yang harus dimanfaatkan untuk dapat mendeteksi tidak hanya koordinat bola namun juga jaraknya. Informasi jarak bola sangat penting bagi robot kiper. Jika sebuah robot kiper tidak mengetahui jarak bola, maka robot tidak tau kapan harus menangkap bola karena tidak tau bola tersebut masih jauh atau sudah dekat dengan gawang. Jarak bola dapat diketahui dengan mengetahui terlebih dahulu koordinat serta diameter bola pada *frame* citra. Maka dari itu *frame* citra tersebut yang harus diproses untuk menghasilkan input jarak. Dengan adanya pengetahuan jarak bola, maka robot kiper dapat menangkap bola hanya pada kondisi bola bergerak kearah gawang sehingga pergerakan robot kiperpun menjadi lebih efektif.

Sebelumnya, telah ada penelitian terkait dengan pendeteksian jarak objek terhadap kamera. Penelitian ini dilakukan oleh Isnan Nur Rifai dan Budi Sumanto yang berjudul *Computer Vision Untuk Penghitungan Jarak Obyek Terhadap*

Kamera. Pada penelitian tersebut objek di deteksi dengan menggunakan algoritma haar. Setelah terdeteksi maka dilakukan perhitungan jarak objek tersebut dengan cara membandingkan jari jari objek berdasarkan *pixel* dan jari jari objek sebenarnya. Sistem perhitungan jarak objek ini memiliki tingkat kesalahan 8,59 cm dengan jarak 100-200 cm. tingkat kesalahan tersebut masih tergolong tinggi. Untuk itu perlu sistem perhitungan jarak yang lebih baik dengan tingkat kesalahan kurang dari 5 cm. Hal ini dapat diperoleh dengan menambah besarnya ukuran citra yang diproses dan pengolahan sinyal yang tepat juga (Isnain & Budi, 2013).

Keakuratan pendeteksian jarak objek yang berbentuk bulat seperti bola bergantung pada keakuratan pendeteksian koordinat dan diameter bola. Salah satu metode yang digunakan untuk pendeteksian lingkaran adalah *Hough Circle Transformation*. Penulis ingin menerapkan *Hough Circle Transformation* dalam penelitian ini dikarenakan keakuratan hasil penggunaan metode ini yang terbilang cukup tinggi. Keakuratan *Hough Circle Transformation* lebih tinggi dibandingkan dengan metode *Blob Detection* (Pamungkas, et al., 2017). Keuntungan lain penggunaan *Hough Circle Transformation* untuk pendeteksian lingkaran adalah toleran terhadap celah dalam deskripsi batas fitur dan relatif tidak terpengaruh oleh noise gambar, tidak seperti detektor tepi (Cherabit, et al., 2012). Namun ada kekurangan dari metode *Hough Circle Transformation* ini, yaitu komputasinya yang berat dan membutuhkan penyimpanan yang banyak (Jen, et al., 2006). *Hough Circle Transformation* ini memiliki karakteristik yang mampu mendeteksi sebuah lingkaran walaupun lingkaran yang ada pada *frame* tidak lengkap (ImageJ, 2017).

Pada skripsi ini akan dibuat suatu sistem pendeteksian bola pada robot sepak bola menggunakan metode *Hough Circle Transformation*. *Hough Circle Transformation* adalah transformasi gambar yang memungkinkan objek melingkar di ekstraksi dari sebuah gambar bahkan jika lingkarannya tidak lengkap atau tidak sempurna (ImageJ, 2017). Pendeteksian bola menggunakan metode *Hough Circle Transformation* memiliki waktu yang lebih cepat dibandingkan dengan pendeteksian bola menggunakan *Blob Detection* (Pamungkas, et al., 2017). Dengan metode maka bisa diperoleh lokasi bola serta ukuran bola tersebut. Kemudian dilakukan perhitungan jarak dengan data ukuran lingkaran yang terdeteksi. Sistem ini diharapkan dapat bekerja dengan cepat, akurat dan *realtime* dimana jarak bola dapat tetap terdeteksi walaupun bola bergerak sehingga robot kiper dapat menentukan mengambil keputusan dengan baik dan efektif.

1.2 Rumusan Masalah

Rumusan masalah berdasarkan dari latar belakang yang ada di penelitian ini adalah sebagai berikut :

1. Bagaimana merancang Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi?

2. Bagaimana mengimplementasikan Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi?
3. Bagaimana akurasi Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi?
4. Bagaimanakah waktu komputasi Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang sistem pendeteksian jarak bola pada robot kiper sepak bola menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi.
2. Mengimplementasikan *Hough Circle Transformation* untuk pendeteksian jarak kamera dengan bola pada robot kiper sepak bola berbasis Raspberry Pi.
3. Mengetahui akurasi performansi Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi.
4. Mengetahui waktu komputasi Deteksi Jarak Bola Pada Robot Kiper Sepak Bola Menggunakan *Hough Circle Transformation* Berbasis Raspberry Pi

1.4 Manfaat

Manfaat dari penelitian ini adalah agar robot kiper sepak bola dapat menentukan jarak dirinya dengan bola. Informasi jarak ini membantu robot kiper untuk menentukan keputusan aksi yang akan dilakukannya, apakah perlu menangkap bola atau tidak. Sistem ini dibuat untuk dapat berjalan *real time* agar bisa diimplementasikan langsung pada saat robot kiper sepak bola bertanding.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Sistem bersifat statis atau tidak bergerak pada saat pendeteksian jarak bola.
2. Bola yang dideteksi adalah bola dengan diameter 15,1 cm dengan warna sebagian besar putih.
3. Pengujian dilakukan pada lapangan pertandingan KRSBI Humanoid.
4. Pembahasan difokuskan pada keakuratan jarak bola yang dihasilkan oleh sistem.

1.6 Sistematika Pembahasan

Sistematika penulisan penelitian ditujukan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab sebagai berikut :

BAB I PENDAHULUAN

Pada Bab I ini, dipaparkan mengenai latar belakang mengapa deteksi jarak bola ini diperlukan, rumusan masalah deteksi jarak bola yang berdasar latar

belakang, tujuan serta juga manfaat penelitian bagi perkembangan robot sepak bola, batasan atau ruang lingkup dari penelitian, kemudian sistematika penulisan laporan dari penelitian pendeteksian jarak bola ini.

BAB II LANDASAN KEPUSTAKAAN

Pada bab 2 ini, akan dipaparkan terkait dengan teori yang melandasi adanya penelitian deteksi jarak bola ini. Pada bab landasan kepastakaan akan dicantumkan pula 3 penelitian sebelumnya sudah dikerjakan yang serupa dan terkait dengan penelitian pendeteksian jarak bola.

BAB III METODOLOGI PENELITIAN

Pada bab metode penelitian akan dibahas terkait langkah kerja dalam penelitian deteksi jarak bola ini yakni studi literatur, rekayasa kebutuhan untuk membangun sistem, perancangan sistem yang meliputi perancangan perangkat keras dan perancangan perangkat lunak, implementasi, pengujian dan analisisnya, serta kesimpulan yang dapat diambil dari penelitian deteksi jarak bola ini.

BAB IV REKAYASA KEBUTUHAN

Pada bab ini dipaparkan kebutuhan kebutuhan yang harus disediakan atau dipenuhi untuk penelitian ini, yakni : kebutuhan fungsional, kebutuhan non fungsional yang meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak. Pada bab ini juga akan dipaparkan mengenai gambaran umum deteksi jarak bola serta batasan desain dari sistem ini.

BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab perancangan dan implementasi akan dipaparkan perancangan sistem, dan juga pengimplementasian dari sistem yang sudah dirancang. Pada perancangan sistem akan di lakukan perancangan sampai sesuai dengan rancangan dan mampu bekerja sesuai tujuannya, sedangkan pada implementasi sistem, akan dipaparkan bagaimana metode *Hough Circle Transformation* diimplementasikan pada deteksi jarak bola ini.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ini akan dipaparkan terkait dengan skenario pengujian terhadap deteksi jarak bola. Kemudian dilakukan analisis apakah deteksi jarak bola ini sudah bekerja sesuai tujuannya atau belum berdasarkan hasil dari pengujian.

BAB VII PENUTUP

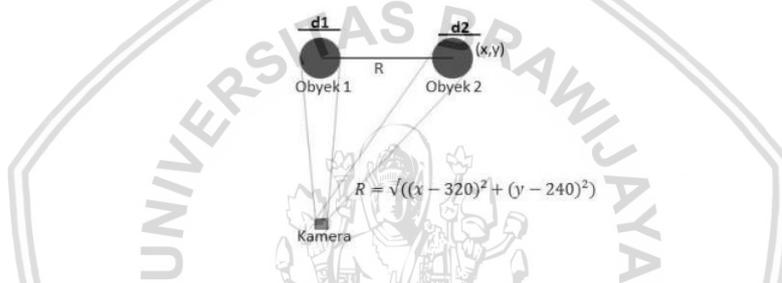
Pada bab ke 5 yakni bab penutup ini, akan dipaparkan kesimpulan dari penelitian deteksi jarak bola yang sudah dilakukan. Dari hasil kesimpulan tersebut akan di berikan saran agar nantinya deteksi jarak bola ini dapat dikembangkan menjadi lebih baik.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepustakaan yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian ini. Kajian pustaka membahas penelitian sebelumnya yang terkait dengan penelitian yang diusulkan. Adapun dasar teori membahas terkait literatur yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Tinjauan Pustaka

Tinjauan Pustaka membahas perbandingan antara penelitian yang pernah dilakukan sebelumnya yang terkait dengan penelitian ini. Pada penelitian kali ini, tinjauan pustaka diambil dari 3 penelitian yang pernah dilakukan sebelumnya. Tujuan dari adanya tinjauan pustaka ini adalah untuk mengkaji metode dan hasil dari penelitian sebelumnya untuk dijadikan dasar dalam penelitian ini.



Gambar 2.1 Ilustrasi penelitian pertama

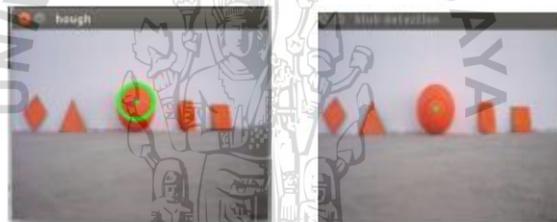
Sumber : (Isnan & Budi, 2013)

Pada penelitian pendeteksian jarak objek menggunakan kamera sebelumnya yang dilakukan oleh Isnan Nur Rifai dan Budi Sumanto dengan judul “*Computer Vision Untuk Penghitungan Jarak Obyek Terhadap Kamera*”, digunakan algoritma haar untuk mendeteksi objek. Setelah koordinat posisi target ditemukan, lalu dihitung jari jari objek tersebut berdasarkan *pixel* yang digunakan. Besar jari jari ini yang nantinya akan dibandingkan dengan jari jari sebenarnya untuk menghitung jarak objek. Pada Gambar 2.1 terlihat bahwa di penelitian ini posisi objek yang tidak lurus juga diperhitungkan. Pada penelitian ini tingkat eror yang dihasilkan masih tergolong tinggi yakni sebesar 8,59 cm dengan jarak 100 – 200 cm (Isnan & Budi, 2013).



Gambar 2.2 Gambar penelitian kedua

Penelitian kedua berjudul “*A Real-Time Face to Camera Distance Measurement Algorithm Using Object Classification*”. Seperti yang terlihat pada Gambar 2.2, penelitian tersebut mengukur jarak antara kamera dengan wajah. Kedua iris mata dideteksi menggunakan *Hough Transform*. Jarak antar iris inilah yang akan digunakan untuk menghitung seberapa dekat wajah dan kamera. Hasil penelitian ini adalah bahwa jarak yang dapat dideteksi tidak lebih dari 40 cm. Jika di atas 40 cm maka sistem akan menghasilkan nilai jarak yang salah (Md, 2015).



Gambar 2.3 Gambar penelitian ketiga

Penelitian ketiga yang terkait dengan pendeteksian bola dengan menggunakan metode *Hough Circle Transformation* “Sistem Pendeteksi dan Pelacakan Bola dengan Metode *Hough Circle Transformation*, *Blob Detection*, dan *Camshift* Menggunakan AR.Drone”. Pada penelitian ini dilakukan pendeteksian bola menggunakan 2 metode yakni *Blob Detection* dan *Hough Circle Transformation*. Hasil yang didapatkan adalah bahwa menggunakan *Hough Circle Transformation* membutuhkan waktu komputasi yang lebih cepat dibandingkan dengan metode *Blob Detection*. Tingkat keberhasilan pendeteksian bola menggunakan *Hough Circle Transformation* pun mencapai 100% (Pamungkas, et al., 2017).

Berdasarkan penelitian-penelitian yang telah tersebut diatas, penulis ingin mengatasi kekurangan yang ada pada penelitian pertama dengan membuat deteksi jarak bola pada robot kiper sepak bola menggunakan *Hough Circle Transformation* berbasis Raspberry Pi. Pada penelitian pertama, digunakan algoritma *Haar* dan untuk pendeteksian objek. Dengan algoritma ini sistem dapat menentukan jarak bola namun hasilnya kurang akurat. Oleh karena itu, penulis

ingin menggunakan metode lain yakni *Hough Circle Transformation* untuk mendapatkan hasil jari jari objek yang lebih akurat. Pada penelitian kedua merupakan penelitian deteksi jarak suatu objek yakni mata menggunakan kamera. Sedangkan Pada penelitian ini penulis melakukan deteksi objek yang berbeda yakni bola namun dengan pengolahan citra digital juga. Hasil dari penelitian kedua kurang baik dimana objek mata maksimal yang dapat dideteksi adalah pada jarak 40 cm. Oleh karena itu penulis ingin membuat Deteksi jarak bola dengan kemampuan pendeteksian jarak yang lebih jauh. Selanjutnya penelitian ketiga merupakan penelitian terkat deteksi bola menggunakan algoritma *Hough Circle Transformation* dan *Blob Detection* pada *Air Drone*. Berdasarkan penelitian ketiga, penggunaan *Hough Circle Transformation* lebih akurat dibandingkan dengan *Blob Detection*. Oleh karena itu, penulis ingin menerapkan salah satu metode yang ada pada penelitian tersebut yakni *Hough Circle Transformation* untuk mendeteksi jarak bola pada robot kiper sepak bola.

2.2 Dasar Teori

2.2.1 Robot Kiper Sepak Bola



Gambar 2.4 Robot KRSBI Humanoid

Sumber : (UMY, 2018)

Robot kiper Sepak Bola merupakan robot humanoid yang dipertandingkan dalam pertandingan Kontes Robot Sepak Bola Indonesia (KRSBI) Humanoid. Dalam KRSBI Humanoid, pertandingan dilangsungkan pada lapangan berukuran 4 meter x 6 meter. Lapangan tersebut berwarna hijau dan terbuat dari rumput sintesis. Warna garis lapangan dan warna gawang adalah putih. Warna bola yang digunakanpun sebagian besar putih (KRSBI, 2018). Hal ini tentu menambah kesulitan tersendiri bagi robot untuk mengidentifikasi objek bola.

Dalam 1 tim robot Sepak Bola terdapat maksimal 5 robot yang bermain. Salah satu robot berposisi sebagai kiper atau penjaga gawang. Tugas utama dari robot kiper Sepak Bola adalah menepis bola yang mengarah ke gawangnya agar tidak terjadi *goal*. Untuk bisa menepis bola robot harus tau posisi bola dan seberapa jauh posisi bola itu terhadap dirinya.

2.2.2 Hough Circle Transformation

Hough Circle Transformation adalah transformasi gambar yang memungkinkan objek melingkar di ekstraksi dari sebuah gambar bahkan jika lingkarannya tidak lengkap atau tidak sempurna. Transformasi secara efektif mencari objek dengan tingkat simetri radial yang tinggi. Metode ini bekerja dengan mengubah gambar di sekeliling lingkaran. Setiap kali piksel yang ditransformasikan dengan intensitas yang lebih besar dari nol pada koordinat Cartesian, koordinat tersebut mendapat satu suara. Saat gambar terus berubah dalam lingkaran radius tertentu, jika lingkaran pada gambar memiliki radius yang sama, maka suara akan terakumulasi di pusat massa lingkaran ini (ImageJ, 2017). Dengan menggunakan metode *Hough Circle Transformation* kita bisa mendapatkan radius serta luasan dari bola.

Persamaan 2.2 adalah Persamaan karakteristik lingkaran r dan pusat (a, b) .

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.2)$$

Pada **Persamaan 2.2** diatas, A dan b merupakan titik pusat lingkaran, sedangkan r adalah jari-jari lingkaran

Lingkaran juga dapat didefinisikan sebagai **Persamaan 2.3** dan **Persamaan 2.4**

$$x = a + r \cos \theta \quad (2.3)$$

$$y = b + r \sin \theta \quad (2.4)$$

Berdasarkan **Persamaan 2.3** dan **Persamaan 2.4**, Lingkaran memiliki tiga bagian informasi yakni pusat (a, b) dan jari-jari (r). dengan demikian, tugas dari *Hough Circle Transformation* adalah mencari triplet (a, b, r) yang menentukan point (x_i, y_i) . Jika (θ) bervariasi dari 0 sampai 360, maka sebuah ilngkaran yang lengkap dengan radius R akan terbentuk. (Cherabit, et al., 2012)

Jika kita mencari lingkaran dengan jari jari yang sudah diketahui, parameter untuk pencarian dikurangi menjadi sepasang (a, b) dan ruang H^2 dimensi. Alur pencarian lingkaran dengan R yang sudah diketahui adalah sebagai berikut :

1. Muat gambar
2. Deteksi tepiannya dan hasilkan gambar biner
3. Untuk setiap *pixel* 'tepi', buat lingkaran di ruang ab
4. Untuk setiap titik pada lingkaran di ruang ab , berikan 'votes' di sel akumulator
5. Sel-sel dengan jumlah suara yang lebih banyak adalah pusat

Ketika R tidak diketahui, solusi untuk mencari lingkaran adalah dengan berasumsi. Asumsikan $R = 1$, lalu jalankan algoritma diatas. Kemudian asumsikan $R = 2$ dan jalankan lagi. Setelah itu asumsikan $R = 3$, dan seterusnya. Batas asumsi

nilai R ini adalah Panjang diagonal dari sebuah gambar atau citra. Hal ini dikarenakan tidak mungkin lingkaran pada gambar dapat memiliki radius lebih besar dari atau sama dengan diagonal. (Sinha, n.d.)

2.2.3 Kamera Webcam



Gambar 2.5 Logitech c525 HD Webcam

Sumber: Logitech (2018)

Webcam adalah singkatan dari *web camera*. *Webcam* ini merupakan sebuah kamera digital yang bisa terhubung langsung ke computer. Laptop lazimnya sudah dilengkapi dengan kamera *webcam* dan mikrofon, tetapi jika belum ada, kita dapat menambahkannya sendiri. Ada beberapa jenis *webcam*, ada yang langsung ditancapkan di *USB port* pada computer dan ada juga melalui *Wireless* (Anon., 2018).

Webcam yang digunakan di penelitian ini adalah *Webcam Logitech c525*. Fitur di *webcam* ini yakni : dapat merekam video yang mempunyai kualitas HD 720p. *webcam* ini memiliki resolusi 8 megapixel dan memiliki mikrofon yang sudah termasuk di dalamnya. *Webcam* ini memiliki fitur *automatic light correction* dan *premium autofocus*. Port usbnya bertipe usb 2.0 (Logitech, 2018).

Tabel 2.1 Spesifikasi Logitech C525

Sumber : (Logitech, 2018)

Fitur	Keterangan
Dimensi kamera	40.4mm X 68.5mm X 31.75mm
Berat Kamera	88g
Tipe Koneksi	USB
Tipe USB	USB 2.0
USB VID_PID	081A
Mendukung UVC	Ya
<i>Microphone</i>	Ya
Tipe <i>Microphone</i>	Mono
Lensa dan Tipe Sensor	Plastik
Tipe Fokus	Auto
<i>Optical Resolution</i>	2MP, True, 8MP <i>Software enhanced</i>
<i>Diagonal Field of View (FOV)</i>	69°
<i>Focal Length</i>	N/A

Fitur	Keterangan
<i>Image Capture</i> (4:3 SD)	320x240, 640x480, 2MP, 8.0MP
<i>Image Capture</i> (16:9 W)	360p, 480p, 720p
<i>Video Capture</i> (4:3 SD)	320x240, 640x480, 2MP
<i>Video Capture</i> (16:9 W)	360p, 480p, 720p
<i>Frame Rate</i> (max)	30fps@640x480
<i>Right Light</i>	Ya
Indicator Lights (LED)	Ya
Privacy Shade	Tidak
Tripod Mounting Option	Tidak
Universal Clip Adjustability (range)	48.94mm
Cable Length	183 CM

2.2.4 Raspberry Pi



Gambar 2.6 Raspberry Pi

Sumber: (pi, 2016)

Raspberry Pi adalah komputer mini berukuran cukup kecil kecil (hanya sebesar kartu kredit) dimana pada awalnya dirancang untuk edukasi. Sang pencipta Raspberry Pi yakni Eben Upton menyatakan bahwa tujuan dari pembuatan Raspberry Pi adalah untuk membuat atau menciptakan perangkat berbiaya rendah untuk meningkatkan skill pemrograman perangkat keras di tingkat pra-universitas. Karena ukuran Raspberry Pi yang kecil dan harganya yang terjangkau, produk ini sangat amat disukai oleh para *developer* elektronik yang proyeknya membutuhkan lebih dari *mikrokontroler* dasar. (opensource.com, 2018)

Pada Penelitian ini, tipe Raspberry Pi yang dipakai adalah Raspberry Pi 3 Model B. model memiliki *Random Access Memory* sebesar 1 GB dan untuk cpunya memakai *processor Quad Core 1.2GHz Broadcom BCM2837 64bit*. Raspberry Pi 3 Model B ini juga dilengkapi dengan *HDMI port*, 4 *USB port*, *ethernet port*, empat puluh pin GPIO, dan *Micro SD Card slot*. Raspberry Pi 3 ini identik dengan generasi sebelumnya yakni Raspberry Pi 2 dan diberi tambahan fitur *on board BCM43438 wireless LAN* dan *on board Bluetooth Low Energy (BLE)*. (pi, 2016)

2.2.5 OpenCV



Gambar 2.7 Logo OpenCV

Sumber: (OpenCV, 2017)

OpenCV (*Open Source Computer Vision Library*) adalah sebuah *library* atau pustaka *computer vision* dan *machine learning* yang bersifat *open source*. *Library* ini memiliki lebih dari 2500 algoritma yang telah dioptimasi. Algoritma OpenCV dapat digunakan untuk mendeteksi objek, *tracking* objek, *face recognition* dan lain lain. OpenCV dirilis dibawah lisensi BSD sehingga dapat digunakan secara bebas bahkan untuk komersil tanpa menulis kode sumbernya (team, 2017).'

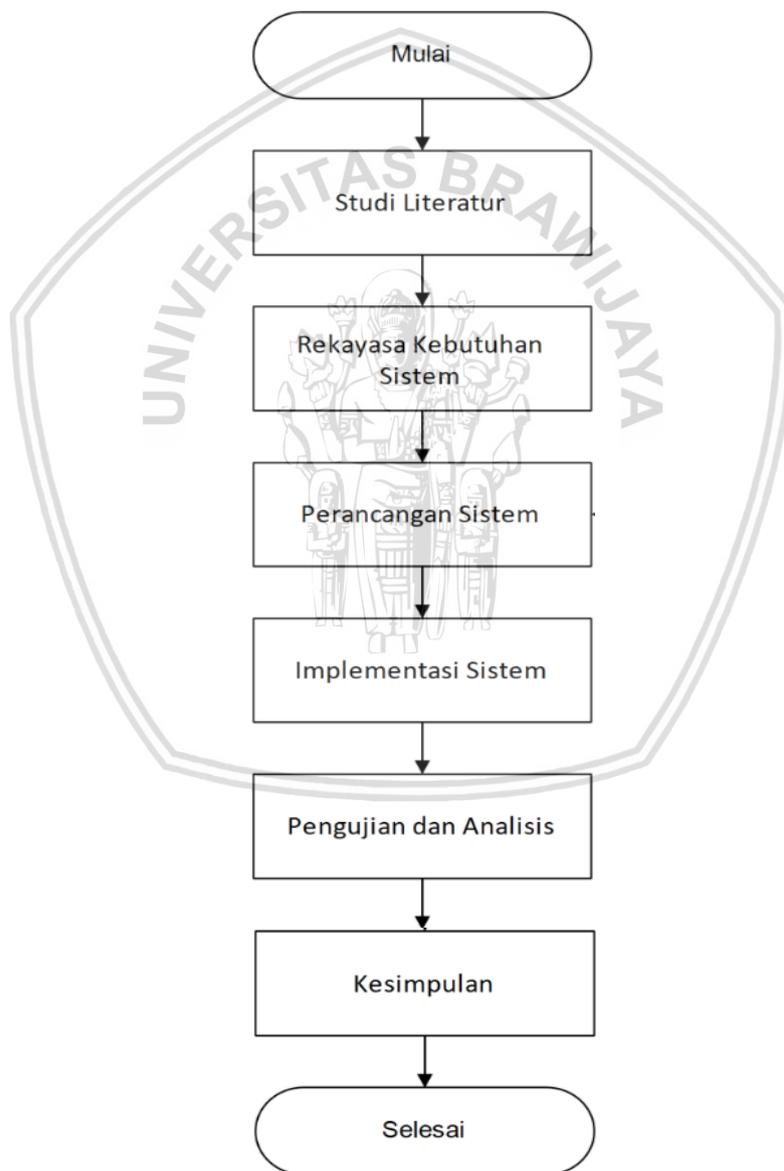
OpenCV didesain untuk dapat berjalan di berbagai macam *platform*. *Library* ini ditulis dalam bahasa C sehingga dapat menjadi *library* yang *portable* pada hampir semua sistem. OpenCV juga dapat berjalan di *desktop* (Windows, Linux, Android, MacOS, FreeBSD, OpenBSD) dan *mobile* (Android, Maemo, iOS) (OpenCV, 2018).

Versi OpenCV yang digunakan dalam penelitian ini adalah versi 2.4.13. versi ini merupakan cabang pengembangan dari versi 2.4 yang telah dilakukan beberapa perbaikan bug. Penelitian ini menggunakan versi OpenCV ini karena versi ini telah digunakan pada proyek-proyek sebelumnya terkait dengan robot sepak bola.

BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Penelitian ini tergolong dalam implementatif. Penelitian dilakukan dengan terlebih dahulu mencari studi literatur yang mengenai perhitungan jarak menggunakan kamera. Kemudian akan dilakukan analisis kebutuhan sistem. Setelah itu sistem akan dirancang dan juga di implementasikan serta diuji. Untuk lebih jelasnya, diagram alir dibawah ini akan menggambarkan bagaimana metodologi dari penelitian yang dilakukan.



Gambar 3.1 Diagram alur metodologi penelitian

3.2 Studi Literatur

Teori dalam penelitian ini didukung dari beberapa referensi. Teorinya meliputi:

- a. Teori Terkait Robot kiper Sepak Bola
Melakukan kajian terhadap robot kiper Sepak Bola yang bertanding pada Kontes Robot Sepak Bola Indonesia Humanoid
- b. Teori terkait metode *Hough Circle Transformation*
Melakukan kajian terkait apa itu metode *Hough Circle Transformation* dan bagaimana penggunaan metode ini dapat berpengaruh terhadap hasil dari Deteksi jarak ini.
- c. Teori terkait kamera *webcam*
Melakukan kajian terhadap kamera *webcam* yang spesifikasinya mampu untuk mengambil gambar secara baik sehingga hasil penelitianpun akan lebih akurat serta juga lebih baik lagi.
- d. Teori terkait Raspberry Pi
Melakukan kajian terhadap Raspberry Pi, terutama terkait bagaimana pengolahan citra pada mini pc Raspberry Pi.
- e. Teori terkait OpenCV
Melakukan kajian terhadap OpenCV terkait dengan versi OpenCV yang tepat untuk digunakan pada sistem ini.

3.3 Rekayasa Kebutuhan Sistem

Di sub bab ini terdapat analisis apa saja yang dibutuhkan untuk penelitian ini.

3.3.1 Gambaran Umum Sistem

Sistem ini akan merekam secara langsung pergerakan bola pada lapangan pertandingan. Bola yang terdeteksi akan diproses oleh sistem lokasi dan besar diameternya. Diameter yang dihitung sistem akan di bandingkan dengan diameter bola yang asli. Setelah dilakukan perhitungan matematis, maka akan diperoleh jarak robot dengan bola. Fungsi dari sistem ini adalah untuk membantu pengambilan keputusan yang harus dilakukan setelah robot mendeteksi adanya bola. Dengan dideteksinya jarak bola maka diharapkan pergerakan robot menjadi lebih efektif dan efisien.

3.3.2 Kebutuhan Fungsional

Kebutuhan fungsional dalam sistem ini yakni sebagai berikut :

- a. Kamera dapat mendeteksi lokasi bola
- b. Mini pc dapat mengolah data yang dihasilkan oleh kamera dan mengalkulasinya menjadi jarak dalam satuan cm.
- c. Jarak dapat ditampilkan secara *realtime* pada *frame*.

3.3.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional dalam sistem ini yakni sebagai berikut :

1. Kebutuhan antarmuka *hardware* (perangkat keras)
Perangkat Keras yang digunakan dalam penelitian ini antara lain sebagai berikut :
 - a. 1 buah kamera *webcam*
 - b. 1 buah Raspberry Pi 3 dengan Model B
2. Kebutuhan antarmuka *software* (perangkat lunak)
Perangkat lunak yang digunakan dalam penelitian ini antara lain sebagai berikut :
 - a. *Real VNC Viewer*
 - b. *Library OpenCV*

3.4 Perancangan

Perancangan dalam pembuatan sistem ini meliputi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak.

3.4.1 Perancangan Perangkat Keras

Pada perancangan perangkat keras ini, terdapat 3 bagian berupa input proses dan output. Terdapat 1 buah sensor input yaitu *Logitech c525 HD Webcam*. Pada bagian proses terdapat *Raspberry Pi 3 Model B* yang berfungsi sebagai pengolah data untuk dihitung menjadi nilai jarak. Pada bagian *output* terdapat layar monitor dimana jarak yang dihasilkan oleh sistem akan ditampilkan.

3.4.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak terdapat algoritma jalannya perangkat lunak dari sistem. Perancangan perangkat lunak dimulai dari pembacaan *frame* oleh kamera, kemudian melakukan perhitungan jarak dan sudut arah bola memanfaatkan *Hough Circle Transformation*, kemudian *output* yang berupa jarak dan sudut arah bola.

3.5 Implementasi Sistem

Pada tahap ini, sistem yang telah dibangun serta dirancang akan diterapkan atau diimplementasikan secara langsung penggunaannya. sistem akan diletakkan pada lapangan pertandingan kemudian sistem akan mendeteksi adanya bola dan secara langsung menampilkan jarak dari bola tersebut.

3.6 Pengujian dan Analisis

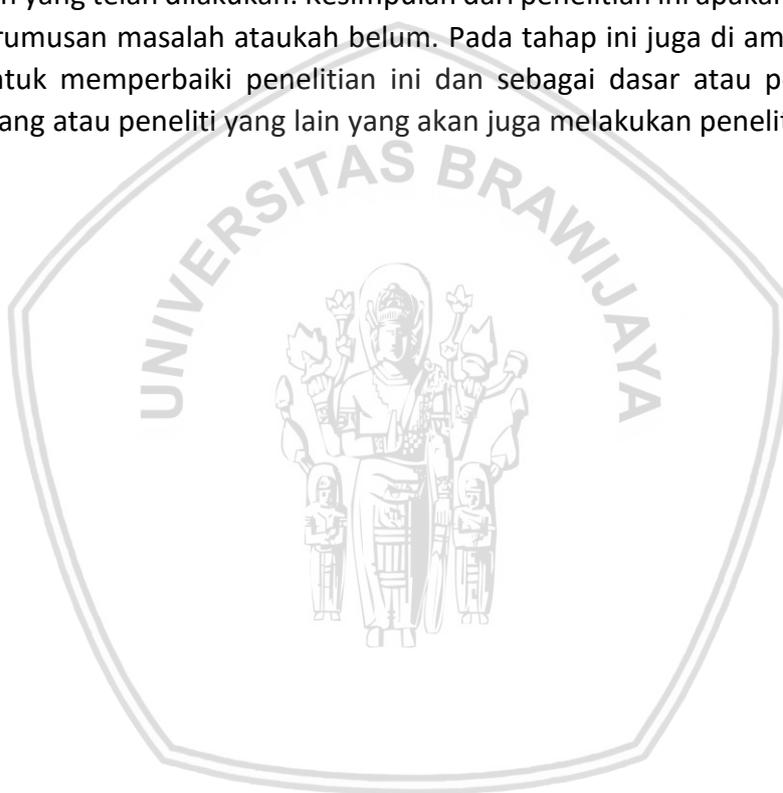
Pada tahap ini, sistem yang telah di implementasikan akan diuji dan dianalisis. Pengujian disini meliputi pengujian terhadap bagaimana kecepatan dan keakuratan kamera dalam menangkap gambar dan menentukan lokasi pergerakan

bola. Pengujian juga dilakukan terhadap ketepatan jarak yang dihasilkan oleh sistem dengan jarak yang sebenarnya.

Setelah diuji, sistem akan di analisis. Analisisnya meliputi analisis kecepatan dalam pengambilan gambar, analisis keakuratan pendeteksian bola dan analisis tingkat ketepatan nilai jarak yang dihasilkan sistem. Analisis disini berguna agar sistem ini dapat dibenahi kekurangannya untuk menjadi lebih baik dan lebih akurat.

3.7 Kesimpulan

Setelah tahap terakhir yaitu tahap analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Kesimpulan dari penelitian ini apakah telah sesuai dengan rumusan masalah ataukah belum. Pada tahap ini juga di ambil beberapa saran untuk memperbaiki penelitian ini dan sebagai dasar atau pertimbangan untuk orang atau peneliti yang lain yang akan juga melakukan penelitian serupa.

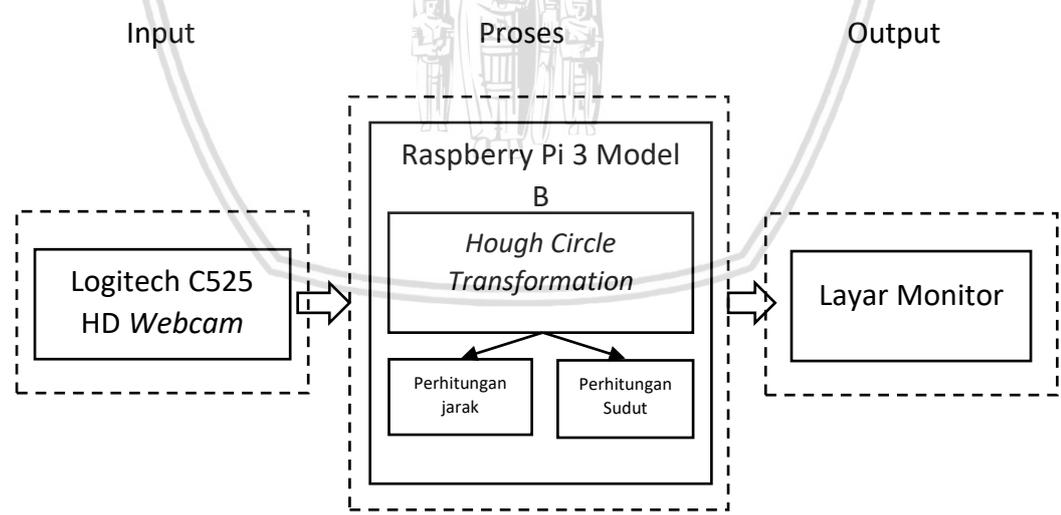


BAB 4 REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan menjelaskan terkait gambaran umum sistem, kebutuhan sistem meliputi kebutuhan fungsional, dan kebutuhan non fungsional yang terdiri dari kebutuhan perangkat lunak dan kebutuhan perangkat keras, dan batasan desain sistem.

4.1 Gambaran Umum Sistem

Deteksi jarak bola pada robot kiper Sepak Bola menggunakan *Hough Circle Transformation* berbasis Raspberry Pi merupakan suatu sistem yang dapat menentukan jarak bola dengan posisi robot kiper Sepak Bola pada lapangan Kontes Robot Sepak Bola Indonesia (KRSBI), dengan parameter berupa koordinat lokasi bola dan diameter bola yang terdeteksi oleh sistem. Koordinat lokasi bola dan diameter bola didapatkan menggunakan sensor kamera Logitech C525 HD Webcam. Berdasarkan nilai dari kedua parameter tersebut maka dilakukan perhitungan jarak dengan menggunakan perbandingan besar diameter bola di setiap koordinat. Sistem ini dapat membantu menyelesaikan permasalahan robot kiper Sepak Bola yang harus mengetahui jarak bola yang tepat untuk ditepis. Hasil dari sistem ini sangat berguna agar gerakan dari robot kiper Sepak Bola dapat lebih efisien dan tepat dalam menangkis bola. Penggunaan metode *Hough Circle Transformation* pada sistem ini dikarenakan metode ini dapat mendeteksi bola beserta diameternya sehingga dapat dicari jarak bola tersebut. Hasil dari pengolahan sistem ini akan secara otomatis ditampilkan pada monitor.



Gambar 4.1 Blok diagram sistem

Gambar 4.1 merupakan diagram blok sistem dimana input dari sistem adalah citra yang diambil menggunakan Logitech C525 HD Webcam. Input ini akan diproses pada Raspberry Pi 3 Model B yang mana pada pemrosesan ini akan dilakukan proses *Hough Circle Transformation* untuk menghasilkan nilai koordinat dan diameter bola. Nilai koordinat dan diameter bola akan digunakan untuk perhitungan jarak dan Sudut Arah Bola. Kemudian *output* dari sistem ini akan ditampilkan pada layar monitor.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menggali semua kebutuhan yang diperlukan untuk Deteksi jarak bola ini. Dalam melakukan analisis kebutuhan sistem terdiri atas beberapa kebutuhan yang perlu dijabarkan yakni kebutuhan fungsional dan kebutuhan non fungsional, dimana kebutuhan non fungsional terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.1 Kebutuhan Fungsional

Berikut ini adalah kebutuhan fungsional yang harus mampu dilakukan oleh sistem :

1. Raspberry Pi 3 Model B dapat mengakses kamera untuk mengambil citra
Pada fungsionalitas ini, Raspberry Pi 3 Model B harus mampu mengakses kamera untuk mengambil citra.
2. Sistem dapat mendeteksi koordinat dan diameter bola menggunakan *Hough Circle Transformation*
Pada fungsionalitas ini, sistem harus mampu mengolah citra untuk dapat mendeteksi bola beserta nilai koordinat (x,y) dan diameter bola menggunakan *Hough Circle Transformation*.
3. Sistem dapat menghitung nilai jarak bola
Sistem harus dapat mengolah data koordinat (x,y) dan diameter bola yang terdeteksi sehingga dapat menghasilkan nilai jarak dalam satuan sentimeter.
4. Sistem dapat menghitung nilai sudut arah bola
Selain jarak sistem juga harus mampu menentukan sudut arah bola. Sudut tersebut didapatkan berdasarkan koordinat bola. Sudut dinyatakan dalam satuan derajat.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan secara rinci dibawah ini.

4.2.2.1 Kebutuhan Perangkat Keras

Guna mendukung implementasi pembuatan sistem maka diperlukan beberapa perangkat lunak yakni sebagai berikut :

1. Raspberry Pi 3 Model B
Mini PC Raspberry Pi 3 Model B digunakan sebagai otak pengolah informasi dalam perancangan sistem
2. Kamera Logitech C525 HD *Webcam*
Kamera ini digunakan sebagai sensor untuk mengambil citra dari bola yang berada pada lapangan kontes robot Sepak Bola Indonesia
3. Kabel UTP
Kabel UTP digunakan sebagai kabel yang menghubungkan Raspberry Pi 3 Model B dengan laptop yang bertindak sebagai monitor
4. Laptop

Laptop dalam sistem ini berfungsi sebagai media untuk mengontrol Raspberry Pi 3 Model B dan menampilkan GUI Raspbian pada Raspberry Pi 3 Model B.

Adapun Spesifikasi laptop yang digunakan antara lain

- Model Perangkat : Asus A43E-VX709D
- Prosesor : Intel Celeron CPU B815
- Sistem Operasi : Windows 10 Enterprise 64-bit

4.2.2.2 Kebutuhan Perangkat Lunak

Guna mendukung implementasi pembuatan sistem maka diperlukan beberapa perangkat lunak yakni sebagai berikut :

1. Microsoft Windows 10 Enterprise 64-bit
Perangkat lunak ini digunakan sebagai sistem operasi yang digunakan oleh laptop.
2. Real VNC Viewer
Perangkat lunak ini digunakan untuk mengontrol Raspberry Pi 3 Model B pada laptop dan menampilkan *GUI* dari Raspbian Jessie pada Monitor Laptop
3. Raspbian Jessie
Perangkat lunak ini digunakan sebagai sistem operasi yang digunakan oleh Raspberry Pi 3 Model B.
4. Library OpenCV 2.4.13
Library ini digunakan untuk memudahkan membuat program untuk menemukan Koordinat dan lokasi Bola pada *frame* citra

4.3 Batasan Desain Sistem

Dalam pembuatan Deteksi jarak bola ini terdapat beberapa Batasan sehingga lingkup pembahasan, perancangan, maupun pengimplementasiannya tidak terlalu luas. Adapun batasan-batasan desain sistem ini adalah sebagai berikut :

1. Menggunakan kamera Logitech c525 HD *Webcam* untuk pengambilan citra.
2. Menggunakan *mini PC* Raspberry Pi 3 Model B untuk pengolah citra.
3. Menggunakan *library* OpenCV untuk proses pendeteksian koordinat dan jari jari bola.
4. Bola yang dideteksi adalah bola dengan diameter 15,1 cm dengan warna sebagian besar putih.
5. Menggunakan metode *Hough Circle Transformation* untuk mendeteksi koordinat dan jari-jari bola.
6. Pengujian dilakukan pada lapangan pertandingan KRSBI
7. Sistem tidak bergerak pada saat pendeteksian jarak bola
8. Pembahasan difokuskan pada keakuratan jarak bola yang dihasilkan oleh sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

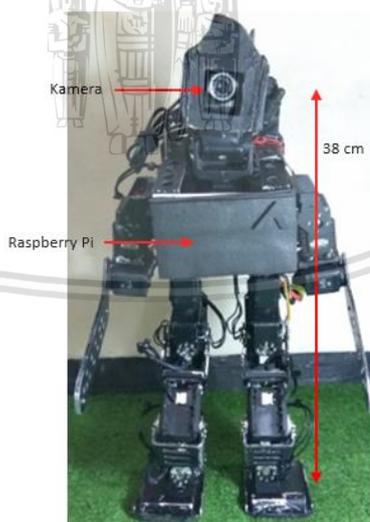
Pada bagian bab 5 ini akan dipaparkan terkait proses perancangan dan implementasi sistem dengan rinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak.

5.1 Perancangan Sistem

Pada sub bab ini akan dipaparkan cara perancangan sistem dimulai dari perancangan *prototype* alat, perancangan perangkat keras hingga perancangan perangkat lunak.

5.1.1 Perancangan *Prototype* Alat Pendeteksi Jarak Bola

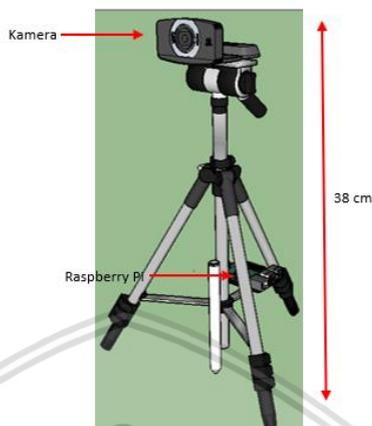
Robot kiper sepak bola Fakultas Ilmu Komputer Universitas Brawijaya memiliki tinggi total 49 cm dengan tinggi robot saat posisi siap dilapangan adalah 43 cm. Posisi kamera saat robot dalam posisi siap di lapangan adalah 38 cm diatas permukaan lapangan. Kamera pada robot kiper bersifat statis dan hanya melihat pada area tertentu di lapangan dan tidak dapat menoleh ke kanan maupun ke kiri. Saat posisi siap, kamera pada robot kiper stabil dan tidak terpengaruh getaran atau gerakan servo karena robot kiper diam atau statis saat mengamati datangnya bola. Sebagai otak pemroses, terdapat Raspberry Pi 3 Model B ditempatkan pada badan robot. Bentuk robot kiper Sepak Bola Fakultas ilmu komputer Universitas Brawijaya ditunjukkan pada Gambar 5.1 berikut.



Gambar 5.1 Gambar robot kiper

Pada Gambar 5.1 terlihat bahwa Robot kiper adalah sebuah sistem yang kompleks yang terdiri dari banyak bagian. Pada penelitian ini, penulis hanya berfokus pada *vision* dari robot, sehingga hanya beberapa bagian dari robot yang digunakan untuk penelitian ini. Bagian-bagian tersebut adalah kamera Web (*Webcam*) dan Raspberry Pi 3 Model B. Penempatan kamera disesuaikan dengan

penempatannya pada robot kiper saat berada di lapangan pertandingan yakni 38 cm di atas lapangan. Sudut pandang dari kamera tidak tegak lurus kedepan, namun diatur sedikit menghadap ke bawah. Desain alat dari deteksi jarak bola ditunjukkan pada Gambar 5.2 berikut.



Gambar 5.2 Prototype alat pendeteksi jarak bola

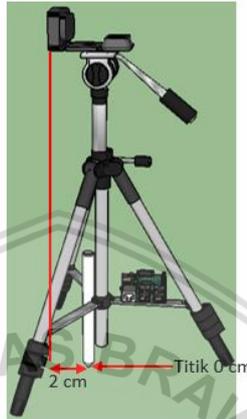
Dalam melakukan desain *prototype* dari sistem deteksi jarak bola pada robot kiper ini perlu diperhatikan peletakkan tiap-tiap komponen serta ukuran alat yang akan dikembangkan. Pada Gambar 5.2 diatas, terlihat bahwa posisi kamera adalah 38 cm di atas permukaan lapangan. Sudut pandangan kamera adalah ke depan bawah dengan sudut terbaik yang akan dicari pada pengujian. Sudut pandang kamera ini sangat penting agar dapat memaksimalkan pandangan dari robot. Untuk posisi Raspberry Pi 3 Model B tidak sama dengan posisinya pada robot kiper. Hal ini dikarenakan posisi dari Raspberry Pi tidak berpengaruh terhadap hasil dari sistem.

Pengukuran jarak ini adalah jarak dari ujung telapak kaki robot menuju ke permukaan depan bola. Pada robot kiper, posisi ujung telapak kaki adalah 2 cm dibelakang titik tegak lurus kamera terhadap lapangan. Robot tampak samping dapat dilihat pada Gambar 5.3 berikut.



Gambar 5.3 Robot tampak samping

Pada Gambar 5.3 dapat terlihat bahwa posisi kamera pada robot tidak sejajar dengan ujung telapak kaki robot yang merupakan titik 0 cm pada sistem. Oleh karena itu pada sistem ini dirancang titik 0 cm yang sama dengan yang ada pada robot. Sistem pendeteksi jarak bola tampak samping dapat dilihat pada Gambar 5.4 berikut.

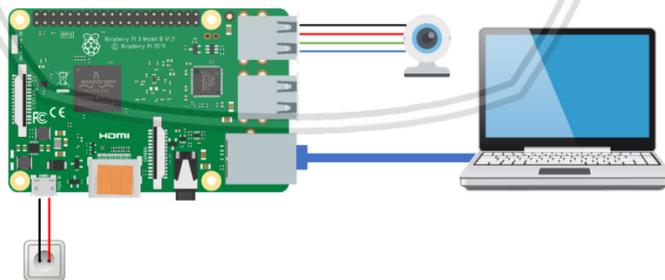


Gambar 5.4 Prototype alat pendeteksi jarak bola tampak samping

Pada Gambar 5.4 terlihat bahwa titik 0 cm pada sistem pendeteksi jarak bola ini disamakan dengan yang ada pada robot kiper yakni 2 cm dibelakang titik tegak lurus kamera terhadap lapangan.

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan berdasarkan analisis kebutuhan perangkat keras serta spesifikasi dari masing-masing dari perangkat keras agar dapat membangun sistem sesuai dengan yang diharapkan.

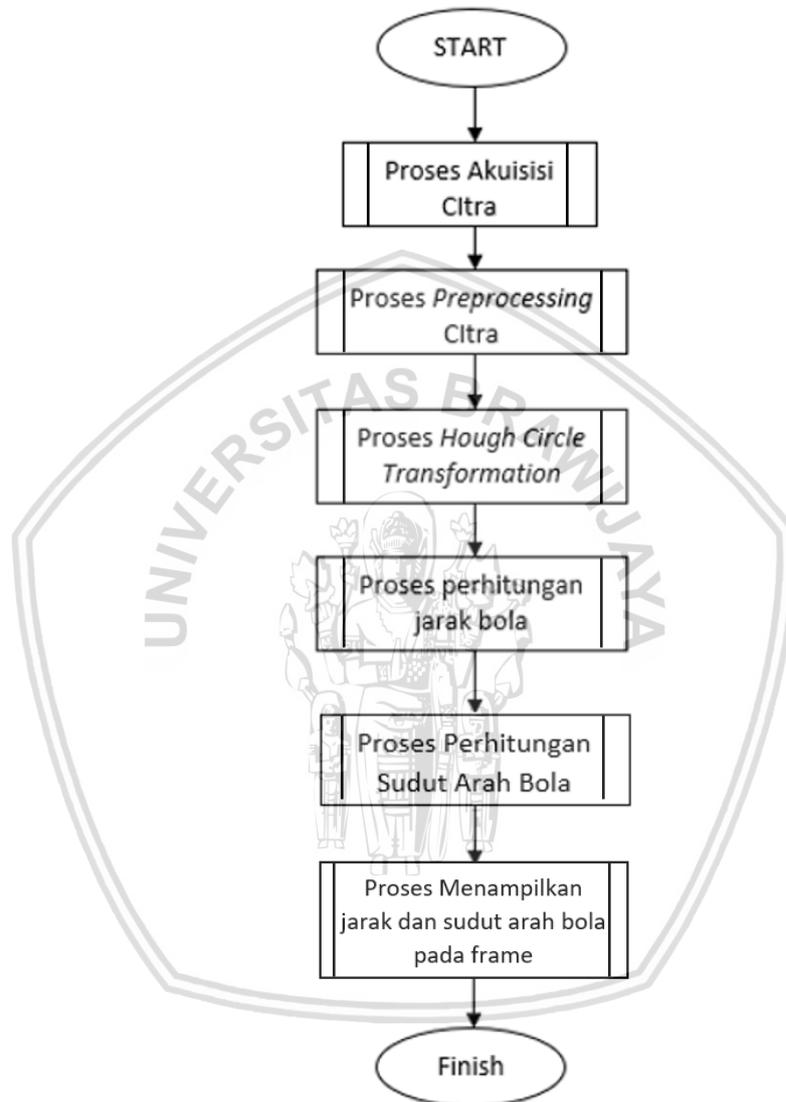


Gambar 5.5 Perancangan perangkat keras deteksi jarak bola

Seperti yang ditunjukkan pada Gambar 5.5, *webcam Logitech c525* dihubungkan dengan *Raspberry Pi 3* menggunakan kabel *USB*. Kabel *USB* dari *webcam* di koneksikan pada salah satu *port USB* yang ada pada *Raspberry Pi 3 Model B*. *Raspberry Pi* dihubungkan ke monitor laptop dengan menggunakan kabel *LAN/UTP*. Pada laptop digunakan aplikasi *Real VNC Viewer* untuk memonitor *GUI* dari *Raspberry Pi*. *Raspberry Pi* mendapatkan daya langsung dari stopkontak menggunakan adaptor dengan *output* sebesar *5 volt* dan *2 ampere*.

5.1.3 Perancangan Perangkat Lunak

Pada subbab perancangan perangkat lunak akan dipaparkan mengenai perancangan perangkat lunak untuk pengolahan citra digital dan juga perancangan perangkat lunak untuk perhitungan jarak bola. Diagram alir perancangan perangkat lunak dapat dilihat pada Gambar 5.6 berikut.

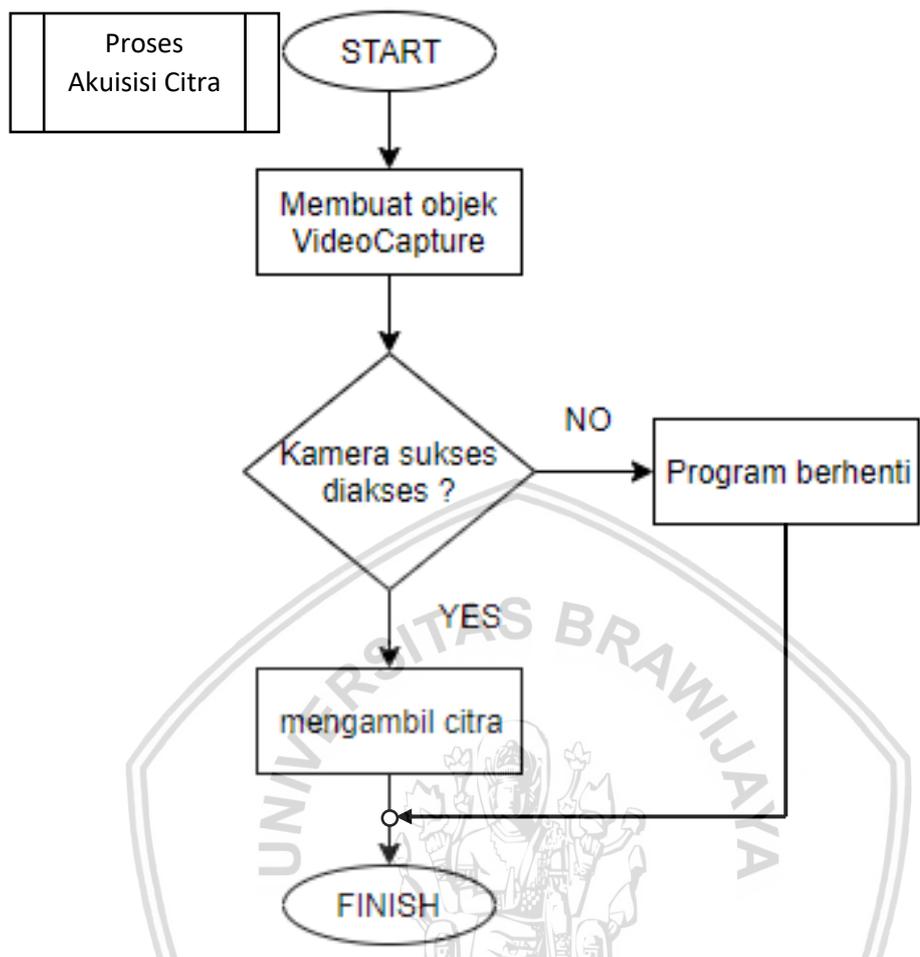


Gambar 5.6 Flowchart perancangan perangkat lunak

5.1.3.1 Perancangan Proses Akuisisi Citra

Dalam penelitian ini, untuk pemrosesan citra menggunakan *library* OpenCV 2.4.13 yang terinstall pada Raspberry Pi 3 Model B. Dalam proses pengambilan akuisisi data citra menggunakan fungsi *videcapture()*. Fungsi ini akan dapat melakukan *capturing* citra pada video maupun perangkat seperti kamera. Untuk mengakuisisi data citra melalui kamera, argumen dari fungsi *videcapture()* adalah index dari perangkat kamera. Proses akuisisi data dapat dilihat pada Gambar 5.3 berikut.

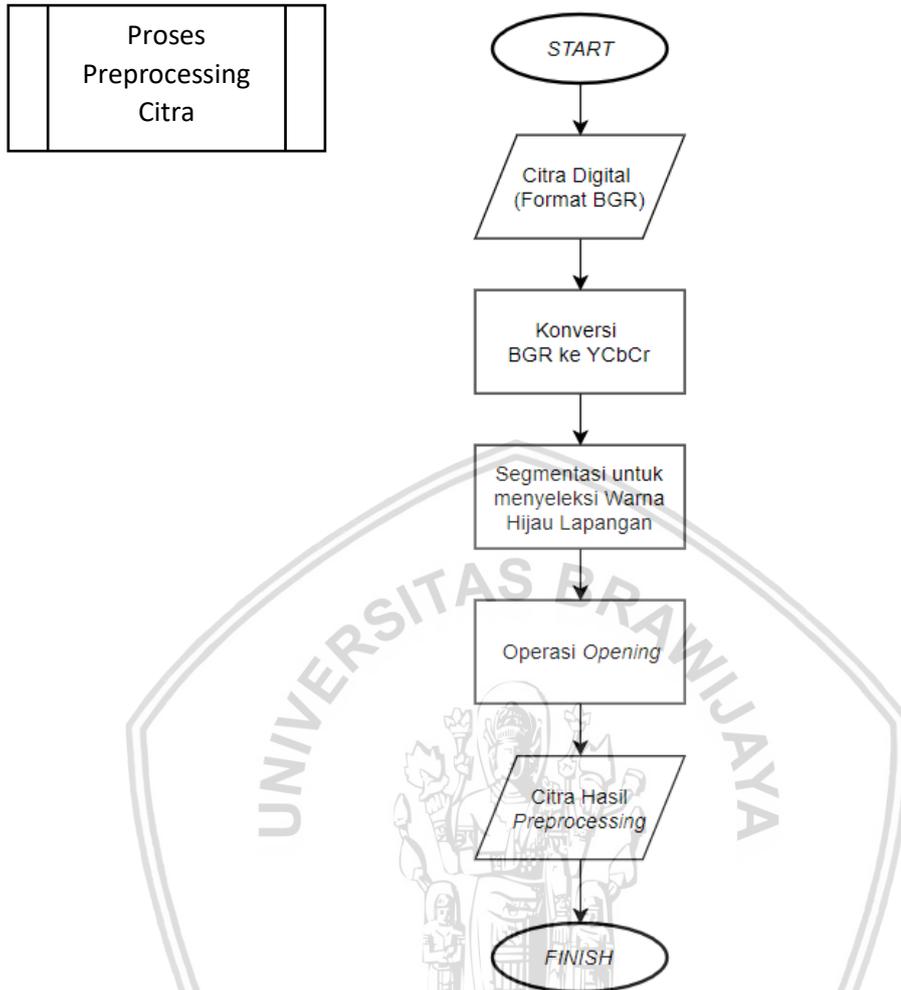




Gambar 5.7 Flowchart akuisisi data citra

Pada Gambar 5.7 dapat dilihat bahwa untuk melakukan akuisisi data citra secara *real time* menggunakan *webcam*, terlebih dahulu harus membuat objek *videocapture*. Dalam penelitian ini objek *videocapture* diberi nama *cap*. Argumen dari fungsi *videocapture* adalah 0. Hal ini dikarenakan hanya ada 1 kamera yang terhubung ke Raspberry Pi dan kamera tersebut memiliki nilai index 0. Setelah objek *videocapture* terbuat, dilakukan pengecekan terhadap sukses tidaknya program mengakses kamera. Jika sukses maka resolusi citra yang diambil akan diatur menjadi 320 x 240 *pixel*. Namun jika gagal mengakses kamera maka program akan berhenti dan menampilkan pesan error : *Webcam not accessed successfully*. Kemudian dilakukan pembacaan data citra yang ditangkap kamera.

5.1.3.2 Perancangan Preprocessing Citra



Gambar 5.8 Diagram alir perancangan preprocessing citra

Dalam melakukan pendeteksian bola, terdapat beberapa tahap yang harus dilakukan agar hasil deteksi baik. Salah satu tahap yang penting adalah *proprocessing* citra. *Preprocessing* ini harus mampu membuat citra yang akan diproses pada pemrosesan utama menjadi lebih baik dan meminimalisir *noise*. Pada OpenCV, hasil pengambilan citranya memiliki format *color space* standar yakni BGR (*Blue Green Red*). *Color space* ini akan dikonversi menjadi YCbCr. Dalam pendeteksian bola di lapangan, digunakan *color space* YcbCr karena warna dan kecerahan cahaya yang dihasilkan *color space* ini lebih stabil daripada *color space* lain sehingga pendeteksian bolapun bisa lebih stabil. Konversi BGR menjadi YcbCr dilakukan dengan menggunakan Persamaan 5.1, Persamaan 5.2 dan Persamaan 5.3 (Hamilton, 1992).

$$Y = 0,114 B + 0,587 G + 0,299 R \tag{5.1}$$

$$Cb = 0,5 B - 0,3313 G - 0,1687 R + 128 \tag{5.2}$$

$$Cr = - 0,0813 B - 0,4187 G + 0,5 R + 128 \tag{5.3}$$

Misalnya diketahui nilai suatu *pixel* adalah B = 105, G = 59, R = 95, maka perhitungan Konversi BGR menjadi YcbCr adalah sebagai berikut :

$$\begin{aligned} Y &= 0,114 B + 0,587 G + 0,299 R \\ &= (0,114 \times 105) + (0,587 \times 59) + (0,299 \times 95) \\ &= 11,97 + 34,633 + 28,405 \\ &= 75,008 \\ &= 75 \end{aligned}$$

Jadi nilai Y adalah 75

$$\begin{aligned} C_b &= 0,5 B - 0,3313 G - 0,1687 R + 128 \\ &= (0,5 \times 105) - (0,3313 \times 59) - (0,1687 \times 95) + 128 \\ &= 52,5 - 19,5467 - 16,0265 + 128 \\ &= 144,9268 \\ &= 145 \end{aligned}$$

Jadi nilai C_b adalah 145

$$\begin{aligned} C_r &= -0,0813 B - 0,4187 G + 0,5 R + 128 \\ &= -(0,0813 \times 105) - (0,4187 \times 59) + (0,5 \times 95) + 128 \\ &= -8,5365 - 24,7033 + 47,5 + 128 \\ &= 142,2602 \\ &= 142 \end{aligned}$$

Jadi nilai C_r adalah 145

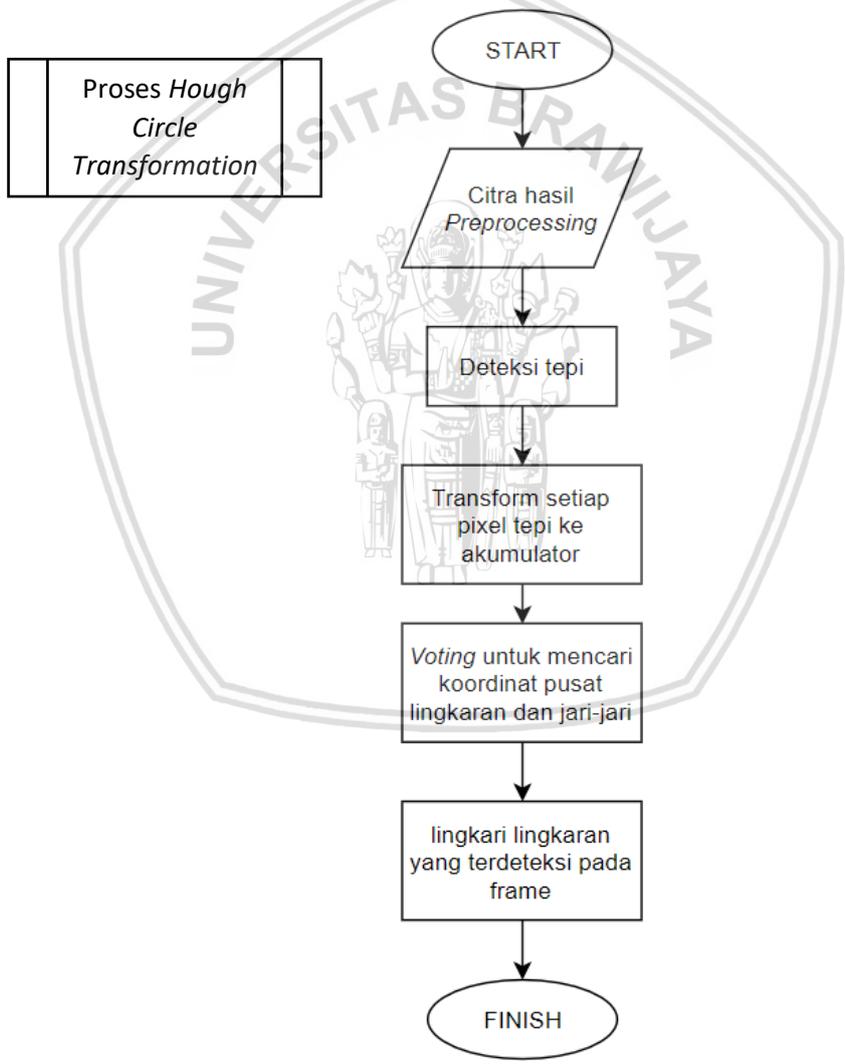
Kemudian citra hasil konversi akan disegmentasi (*thresholding*) menjadi citra biner. Warna hijau yang merupakan warna dari lapangan akan di ubah menjadi nilai 0 sedangkan warna lain yang ada pada lapangan akan bernilai 1. Hal ini akan mempermudah pendeteksian objek berbentuk bulat yang ada di lapangan. Proses *thresholding* masih memiliki beberapa *noise* kecil yang harus dihilangkan. Oleh karena itu operasi *opening* juga diperlukan pada pendeteksian ini. Operasi *opening* digunakan untuk menghilangkan objek berukuran kecil pada background gambar. Operasi *opening* dilakukan dengan urutan erosi kemudian dilasi. Erosi adalah mengecilkan atau menipiskan ukuran bentuk geometris dalam citra biner. Sedangkan dilasi adalah membesarkan atau menebalkan ukuran bentuk geometris dalam citra biner. *Structure element* yang digunakan untuk operasi *opening* ini berbentuk elipse dengan ukuran 5 x 5 *pixel*. *Structure element* pada penelitian ini dapat dilihat pada Gambar 5.9 berikut

0	0	1	0	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	0	1	0	0

Gambar 5.9 Structure element elipse 5 x 5

Setelah *noise* telah diminimalisir lalu citra akan dilanjutkan ke proses utama yakni *Hough Circle Transformation*.

5.1.3.3 Perancangan *Hough Circle Transformation*



Gambar 5.10 Diagram alir perancangan *Hough Circle Transformation*

Pada perancangan *preprocessing* citra, akan dihasilkan citra biner hasil thresholding. Citra inilah yang akan menjadi input untuk proses pengolahan citra

utama yaitu *Hough Circle Transformation*. Citra akan di deteksi tepi-tepinya untuk mempermudah mencari objek lingkaran. Pada OpenCV, deteksi citra pada *Hough Circle Transformation* menggunakan menggunakan metode *canny edge detection*. Setelah itu dibentuk akumulator dengan ukuran 1,5 kali lebih besar dari ukuran citra yang diolah. Kemudian setiap titik pada citra akan ditransformasikan ke akumulator menggunakan Persamaan 5.4 dan 5.5 berikut :

$$a = x - R \cos \theta \quad (5.4)$$

$$b = y - R \sin \theta \quad (5.5)$$

Keterangan :

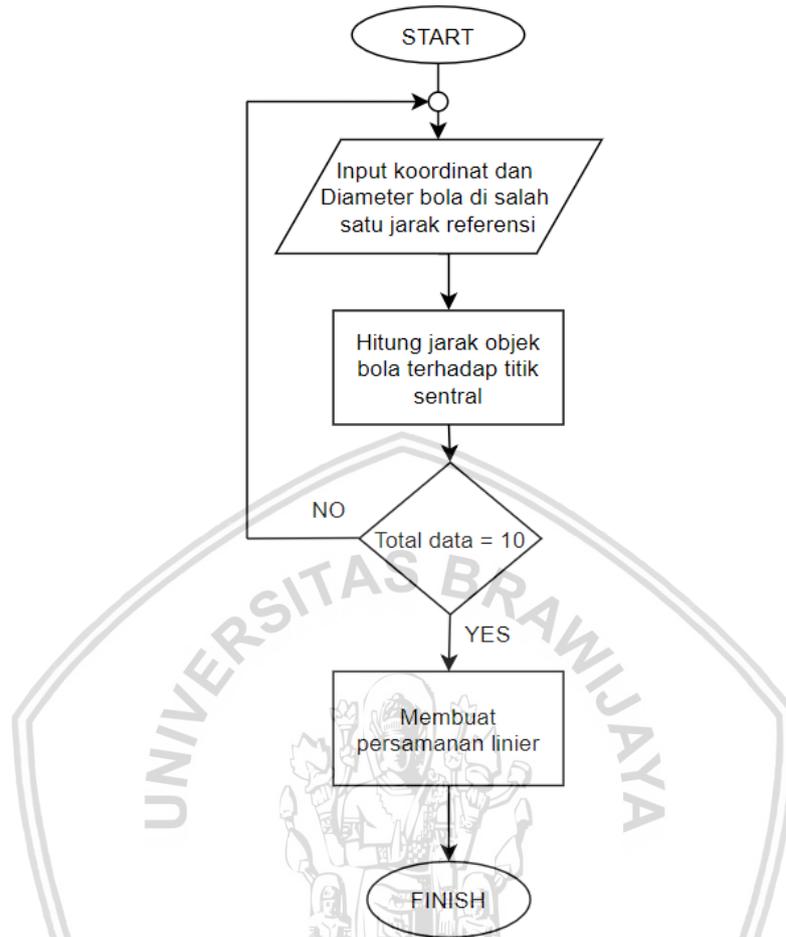
a dan b = titik lingkaran pada akumulator

x dan y = koordinat titik pada citra yang ditransformasikan

θ = sudut dari 0° sampai 360°

Perhitungan nilai a dan b dilakukan pada setiap *pixel* dan menggunakan parameter θ setiap sudut mulai dari 0° sampai dengan 360° . Hasil transformasi ini akan menghasilkan lingkaran baru pada akumulator. Setiap titik pada citra asli akan menghasilkan lingkaran baru pada akumulator sehingga jumlah lingkaran yang baru pada akumulator akan sangat banyak. Karena pada penelitian ini, jari-jari lingkaran yang dicari nilainya belum diketahui, jadi untuk nilai R pada Persamaan 5.4 dan 5.5 diisi dengan berbagai kemungkinan nilai *radius* mulai dari kemungkinan terkecil sampai terbesar. Beberapa lingkaran yang ada pada akumulator akan saling berpotongan pada suatu titik tertentu. Kemudian dilakukan *vote* dan *cell* yang memiliki nilai terbesar akan dideteksi sebagai pusat lingkaran yang sebenarnya.

5.1.3.4 Perancangan Data Diameter Objek Pada Jarak Referensi



Gambar 5.11 Diagram alir pencarian data diameter objek pada jarak referensi

Untuk menghitung jarak menggunakan perbandingan diameter lingkaran, diperlukan beberapa diameter bola pada *frame* sebagai referensi pada 2 jarak berbeda. Pada penelitian ini digunakan 60 diameter referensi dengan 10 diameter pada jarak 50 cm 10 diameter pada jarak 100 cm, 10 diameter pada jarak 150 cm, 10 diameter pada jarak 200 cm, 10 diameter pada jarak 150 cm dan 10 diameter pada jarak 300 cm. Setelah didapat 10 data pada setiap jarak referensi, maka akan dibuat fungsi linier pada setiap jarak referensi. Format fungsi linier yang dihasilkan dapat dilihat pada Persamaan 5.4 berikut :

$$d = b r + a \quad (5.6)$$

Keterangan :

d = diameter berdasarkan jarak referensi

r = variabel Jarak objek bola terhadap titik pusat *frame* pada perhitungan jarak utama

a dan b = nilai Konstanta

Pada Persamaan 5.6 terdapat nilai r adalah variabel dan nilai a dan b adalah konstanta. Konstanta inilah yang perlu dicari terlebih dahulu pada setiap jarak referensi. Perhitungan nilai Konstanta a dan b diawali dengan menghitung nilai R yang merupakan jarak objek bola terhadap titik pusat frame . nilai R didapatkan berdasarkan koordinat (x,y) dari objek bola yang terdeteksi. Perhitungan jarak bola terhadap titik pusat *frame* dilakukan dengan menggunakan Persamaan 5.7 (Isnain & Budi, 2013):

$$R = \sqrt{(x - 160)^2 + (y - 120)^2} \tag{5.7}$$

Keterangan :

R = Jarak objek bola terhadap titik pusat *frame*

x = koordinat x objek bola

y = koordinat y objek bola

Setelah didapat 10 nilai R dari Persamaan 5.7 diatas dan 10 nilai diameter Dari masing-masing jarak referensi, maka selanjutnya dapat dihitung nilai konstanta a dan b menggunakan nilai R dan diameter. Untuk mencari nilai a dan b, digunakan Persamaan 5.8 dan Persamaan 5.9 sebagai berikut :

$$a = \frac{(\sum D)(\sum R^2) - \sum(R)\sum(RD)}{n\sum R^2 - (\sum R)^2} \tag{5.8}$$

$$b = \frac{n\sum RD - \sum(R)\sum(D)}{n\sum R^2 - (\sum R)^2} \tag{5.9}$$

Keterangan :

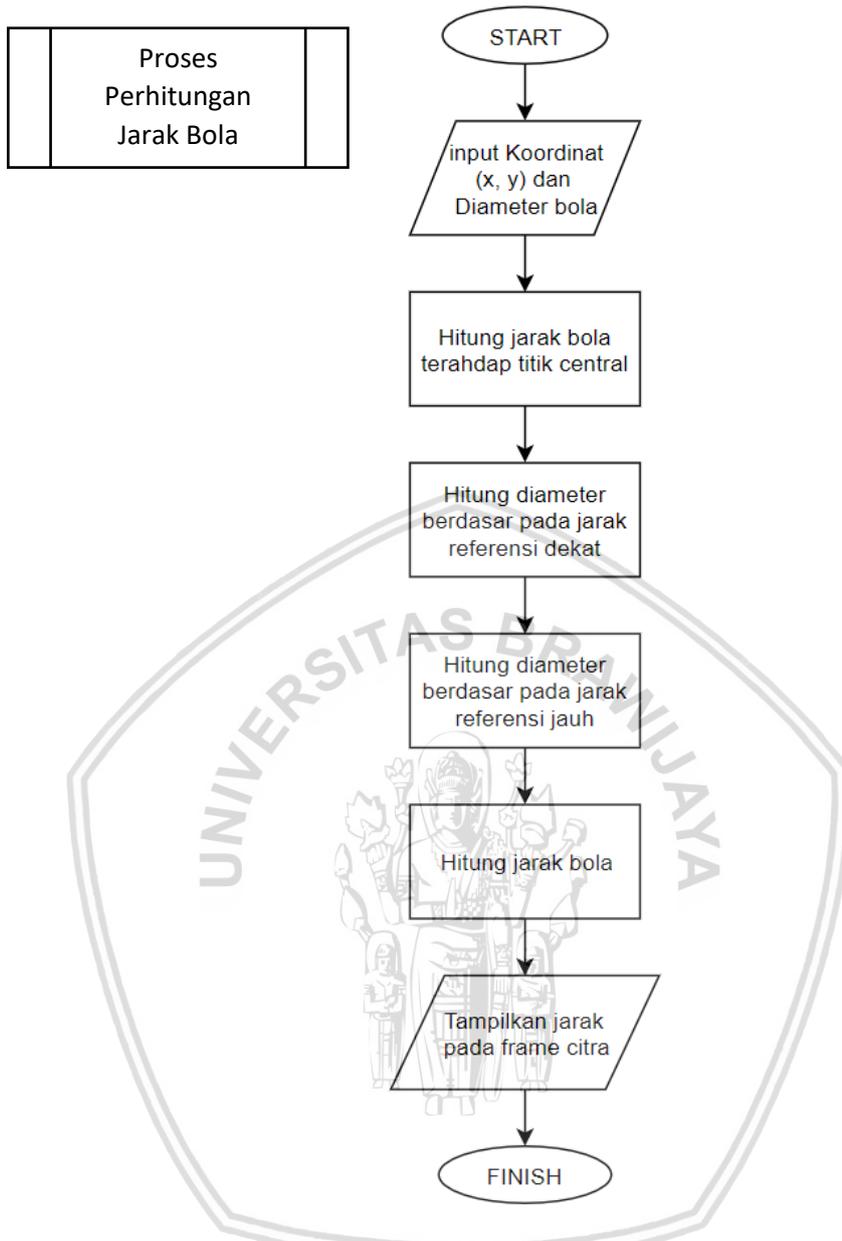
R = Jarak objek bola terhadap titik pusat *frame*

D = Diameter bola yang terdeteksi

5.1.3.5 Perancangan Perhitungan Jarak Bola

Dalam perhitungan jarak bola menggunakan citra digital, diperlukan 2 parameter yakni diameter bola pada *frame* dan koordinat bola pada *frame*. Diameter dan koordinat tersebut didapatkan melalui proses pendeteksian bola dengan menggunakan *Hough Circle Transformation*. Setelah 2 parameter ini di dapatkan, kita dapat menghitung jaraknya dengan menggunakan metode perbandingan. Oleh karena itu, keakurasian diameter dan koordinat bola sangat diperlukan karena sangat berpengaruh terhadap keakurasian pendeteksian jarak bola. Pada Gambar 5.7 dapat dilihat diagram alir proses pendeteksian jarak bola.





Gambar 5.12 Diagram alir perhitungan jarak bola

Sesuai dengan diagram alir pada Gambar 5.11, proses perhitungan jarak bola diawali dengan melakukan perhitungan jarak bola terhadap titik pusat. Perhitungan jarak bola terhadap titik sentral menggunakan Persamaan 5. Proses selanjutnya adalah pencarian diameter bola berdasarkan 2 referensi jarak yaitu jarak dekat dan jauh menggunakan fungsi yang dihasilkan dari perancangan data diameter bola pada jarak referensi. Nilai R yang dihasilkan pada Persamaan (5.4) akan dimasukkan pada fungsi linier jarak referensi dekat dan jauh. Setelah didapat diameter objek berdasarkan diameter referensi jarak dekat dan jauh maka dilakukan perhitungan jarak objek bola menggunakan Persamaan 5.7 dibawah ini (Isnain & Budi, 2013):



$$\frac{d_2 - d_1}{x_2 - x_1} = \frac{D - d_1}{X - x_1} \tag{5.7}$$

Keterangan :

D = Diameter terukur

X = Jarak terukur

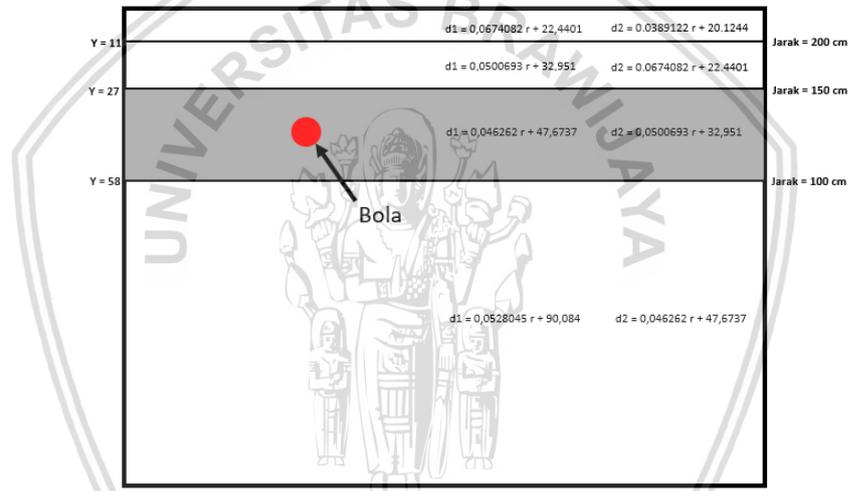
d_2 = Diameter berdasarkan jarak referensi jauh

d_1 = Diameter berdasarkan jarak referensi dekat

x_1 = Jarak referensi dekat

x_2 = Jarak referensi jauh

dari Persamaan 5.7 diatas akan didapatkan hasil berupa jarak bola dalam satuan sentimeter. Contoh perhitungan jarak bola setelah didapatkan data diameter dan koordinat bola adalah sebagai berikut :



Gambar 5.13 Contoh frame untuk perhitungan jarak bola

misal berdasarkan hasil *hough circle transformation*, diketahui beberapa data sebagai berikut :

X = 125

Y = 47

Diameter = 45,607.

Berdasarkan nilai Y yang berada diantara 27 dan 58 maka dapat diketahui nilai-nilai dan fungsi-fungsi sebagai berikut :

$d_1 = 0,046262 r + 47,6737$

$d_2 = 0,0500693 r + 32,951.$

$x_1 = 100$

$x_2 = 150$



Perhitungan jarak bola dilakukan dengan menghitung nilai r terlebih dahulu. Perhitungan nilai r sama dengan perhitungan nilai R dengan menggunakan Persamaan 5.4.

$$\begin{aligned}
 r &= \sqrt{(x - 160)^2 + (y - 120)^2} \\
 &= \sqrt{(125 - 160)^2 + (47 - 120)^2} \\
 &= \sqrt{(-35)^2 + (-73)^2} \\
 &= \sqrt{1225 + 5329} \\
 &= \sqrt{6554} \\
 &= 80,9568
 \end{aligned}$$

Setelah mendapatkan nilai r maka nilai d1 dan d2 dapat dihitung menggunakan fungsi linier jarak referensi dekat dan jarak referensi jauh. Perhitungan nilai d1 dan d2 adalah sebagai berikut :

$$\begin{aligned}
 d1 &= 0,046262 r + 47,6737 \\
 &= (0,046262 \times 80,9568) + 47,6737 \\
 &= 51,4189
 \end{aligned}$$

$$\begin{aligned}
 d2 &= 0,0500693 r + 32,951 \\
 &= (0,0500693 \times 80,9568) + 32,951 \\
 &= 37,0044
 \end{aligned}$$

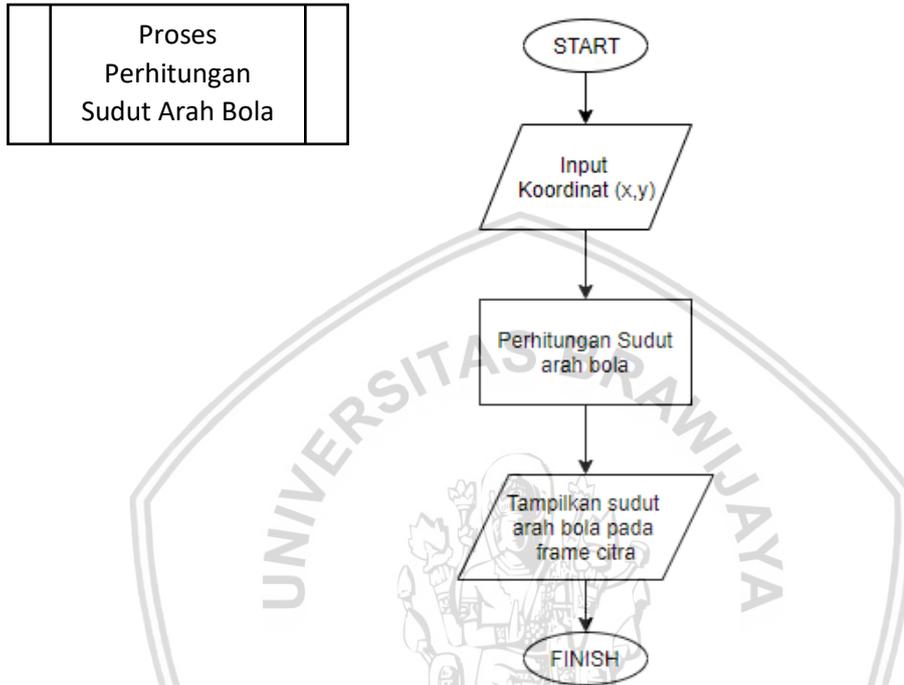
Setelah mendapatkan nilai r, d1 dan d2 kita dapat menghitung jarak bola menggunakan Persamaan 5.7. Perhitungan jarak bola adalah sebagai berikut :

$$\begin{aligned}
 \frac{d_2 - d_1}{x_2 - x_1} &= \frac{D - d_1}{X - x_1} \\
 \frac{37,0044 - 51,4189}{150 - 100} &= \frac{45,607 - 51,4189}{X - 100} \\
 X &= \frac{(45,607 - 51,4189) \times (150 - 100)}{37,0044 - 51,4189} + 100 \\
 X &= \frac{-5,8119 \times 50}{-14,4145} + 100 \\
 X &= 120,16
 \end{aligned}$$

Jadi Jarak bola sebenarnya dilapangan adalah 120,16 cm

5.1.3.6 Perancangan Perhitungan Sudut Arah Bola

Sudut arah bola merupakan sudut antara garis lurus dari koordinat titik 0 cm ke depan sistem dengan garis lurus antara titik 0 cm sistem dengan koordinat bola. Perhitungan sudut ini memanfaatkan nilai koordinat bola yang dihasilkan oleh *Hough Circle Transformation*. Diagram alir proses perhitungan sudut arah bola dapat dilihat pada Gambar 5.14.



Gambar 5.14 Diagram alir perhitungan sudut arah bola

Untuk Perhitungan Sudut arah dari bola, digunakan Persamaan 5.8 (Loughnane, 2012).

$$Sudut = \left(\text{atan} \frac{X_{Sudut} - X_{Bola}}{Y_{Sudut} - Y_{Bola}} \right) * \frac{180}{PI} \quad (5.8)$$

Pada Persamaan 5.8 X_{Sudut} dan Y_{Sudut} merupakan koordinat dari sudut yang akan diukur yang nilainya diperoleh dari hasil kalibrasi sistem. X_{Bola} dan Y_{Bola} merupakan koordinat dari bola yang terdeteksi pada frame. Untuk posisi bola, jika nilai Y_{Bola} lebih dari 160 maka posisi bola adalah disebelah kiri Sistem, Jika Selain itu maka Posisi Bola adalah di sebelah kiri sistem. Contoh perhitungannya adalah sebagai berikut :

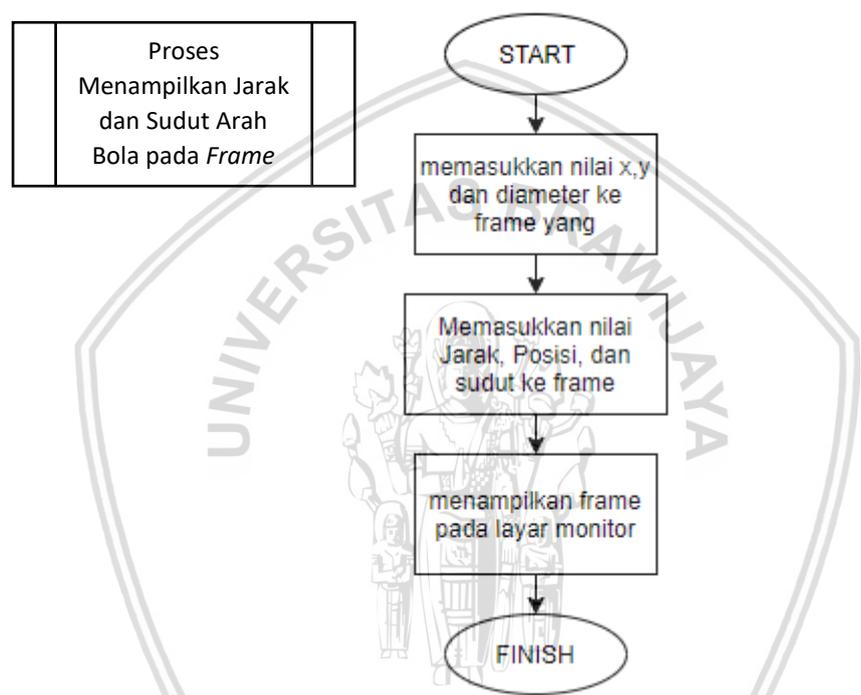
Misal diketahui $X_{Bola} = 125$, $Y_{Bola} = 47$, $X_{Sudut} = 160$ dan $Y_{Sudut} = 450$, maka perhitungan sudutnya adalah sebagai berikut :

$$\begin{aligned}
 Sudut &= \left(\text{atan} \frac{X_{Sudut} - X_{Bola}}{Y_{Sudut} - Y_{Bola}} \right) * \frac{180}{PI} \\
 &= \left(\text{atan} \frac{165 - 125}{450 - 47} \right) * \frac{180}{3,14159265}
 \end{aligned}$$

$$\begin{aligned}
 &= \left(\text{atan} \frac{165 - 125}{450 - 47} \right) * \frac{180}{3,14159265} \\
 &= (\text{atan } 0,08685) * 57,29578 \\
 &= 0,086633 * 57,29578 \\
 &= 4,9637^\circ
 \end{aligned}$$

Nilai $Y_{bola} = 47$ yang mana $47 < 160$, maka posisi bola berada di sebelah kiri sistem. Jadi Sudut Arah Bola tersebut adalah $4,9637^\circ$ di sebelah kiri sistem.

5.1.3.7 Perancangan Menampilkan Jarak dan Sudut Arah Bola pada *Frame*



Gambar 5.15 Diagram alir perancangan menampilkan jarak dan sudut arah bola pada *frame*

Pada Output sistem, akan ditampilkan *frame* citra yang didalam *frame* tersebut barisi nilai hasil dari perhitungan jarak dan sudut arah bola. Selain itu pada *frame* juga kan ditampilkan x,y dan diameter obek bola dengan tulisan yang kecil. Untuk menampilkan *frame* pada layar monitor digunakan fungsi *imshow()*. Selain *output* pada *frame*, *user* juga dapat melihat output dari sistem melalui terminal.

5.2 Implementasi Sistem

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada subbab ini menjelaskan satu per satu secara rinci terkait implementasi *prototype*, implementasi perangkat keras serta implementtasi perangkat lunak.

5.2.1 Implementasi *Prototype* Alat Pendeteksi Jarak Bola

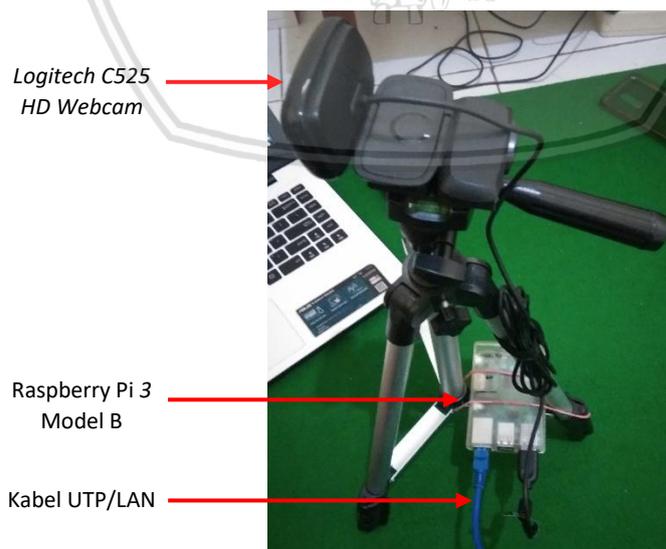
Pengimplementasian *prototype* alat pendeteksi jarak bola mengacu pada perancangan yang ada pada sub bab 5.1.1. *prototype* ini terdiri dari sebuah tripod untuk menyangga kamera yang berada di atasnya. Tinggi dari tripod adalah 38 cm. tinggi ini disamakan dengan tinggi dari sebuah robot kiper. Selain untuk menyangga kamera, tripod ini juga berfungsi untuk meletakkan Raspberry Pi. Hasil dari implementasi *prototype* dan peletakkan komponen komponen alat pendeteksi jarak bola dapat dilihat pada Gambar 5.12 berikut.



Gambar 5.16 Implementasi *prototype* alat pendeteksi jarak bola

5.2.2 Implementasi Perangkat Keras

Pada tahap ini akan dipaparkan proses pengimplementasian perangkat keras yakni, *webcam Logitech c525* dan Raspberry Pi 3 Model B. Komponen ini dihubungkan menjadi 1 berdasarkan pada perancangan yang telah dijelaskan pada sub bab 5.1.2.



Gambar 5.17 Implementasi hubungan Raspberry Pi dengan *webcam*

Pada Gambar 5.13 dapat dilihat hasil implementasi kedua perangkat keras yakni *webcam Logitech c525* dan Raspberry Pi 3 Model B. *Webcam Logitech c525* diletakkan diatas tripod dan menghadap sedikit kebawah. Sedangkan Raspberry Pi 3 mode b diletakkan di bagian bawah dari tripod. Raspberry Pi dan *webcam* dihubungkan melalui kabel USB.

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak memaparkan proses realisasi program untuk Deteksi jarak bola ini berdasarkan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam implementasi perangkat lunak ini, proses pengkodean program dilakukan pada Raspberry Pi menggunakan bahasa pemrograman C++ dan memanfaatkan beberapa *library* OpenCV untuk mempermudah pengolahan citra. Pengimplementasian beberapa *library* yang digunakan pada sistem ini dapat dilihat pada tabel 5.1 berikut.

Tabel 5.1 Kode program inisialisasi library deteksi jarak bola

Baris	Kode Program
1	<code>#include <OpenCV2/highgui/highgui.hpp></code>
2	<code>#include <OpenCV2/imgproc/imgproc.hpp></code>
3	<code>#include <iostream></code>
4	<code>#include <math.h></code>
5	<code>#include <iomanip.h></code>

Berdasarkan tabel 5.1 terdapat 4 *library* yang digunakan pada sistem ini. *Library highui.hpp* dan *imgproc.hpp* adalah *library* OpenCV yang digunakan untuk pendeteksian bola. *Highui.hpp* berfungsi untuk mengaktifkan fungsi fungsi yang terkait dengan UI sedangkan *imgproc.hpp* berfungsi untuk mengaktifkan fungsi fungsi yang terkait dengan pengolahan citra pada *frame*. *Library iostream* digunakan untuk mengaktifkan perintah yang terkait dengan *input output* seperti *cin*, *cout* dan *endl*. *Library math* digunakan untuk mengaktifkan perintah matematika. Operasi matematika yang memerlukan *library math* adalah pengkat (*pow()*) dan akar (*sqrt()*). *Library iomanip* digunakan untuk mengaktifkan perintah perintah yang terkait dengan manipulasi parameter. Dalam penelitian ini *library iomanip* digunakan untuk mengaktifkan fungsi *setprecision()* yang digunakan untuk membatasi nilai angka dibelakang koma menjadi 2 angka saja.

Tabel 5.2 Kode program inisialisasi semua variabel global

Baris	Kode Program
1	<code>#define PI 3.14159265</code>
2	<code>using namespace cv;</code>
3	<code>using namespace std;</code>
4	<code>char charCheckForEscKey = 0;</code>

Baris	Kode Program
5	Mat matOriginal;
6	Mat matProcessed;
7	int morph_pixel = 5;
8	int Y1 = 7;
9	int Cr1 = 0;
10	int Cb1 = 0;
11	int Y2 = 170;
12	int Cr2 = 144;
13	int Cb2 = 131;
14	vector<Vec3f> v3fCircles;
15	int acc_res = 2;
16	int min_dis = 500;
17	int high_thres = 50 ;
18	int low_thres = 35 ;
19	int min_rad = 10;
20	int max_rad = 110;
21	string posisi;
22	int Sudut;
23	string S_Sudut;
24	float Jarak;
25	string S_Jarak;
26	float Diameter;
27	int x_center;
28	int y_center;
29	float R;
30	float d_dekat;
31	float d_jauh;
32	float x_Sudut=160;
33	float y_Sudut=390;
	//y = ar + b
34	float a_dekat1 = 0.0528045;
35	float b_dekat1 = 90.084;
36	float a_jauh1 = 0.046262;
37	float b_jauh1 = 47.6737;
38	float j_ref_dekat1 = 50;
39	float j_ref_jauh1 = 100;
40	int batas1 =77;
41	float a_dekat2 = 0.046262;
42	float b_dekat2 = 47.6737;
43	float a_jauh2 = 0.0500693;
44	float b_jauh2 = 32.951;
45	float j_ref_dekat2 = 100;
46	float j_ref_jauh2 = 150;
47	int batas2 =45;
48	float a_dekat3 = 0.0500693;
49	float b_dekat3 = 32.951;
50	float a_jauh3 = 0.0674082;

Baris	Kode Program
51	float b_jauh3 = 22.4401;
52	float j_ref_dekat3 = 150;
53	float j_ref_jauh3 = 200;
54	int batas3 =29;
55	float a_dekat4 = 0.0674082;
56	float b_dekat4 = 22.4401;
57	float a_jauh4 = 0.0389122;
58	float b_jauh4 = 20.1244;
59	float j_ref_dekat4 = 200;
60	float j_ref_jauh4 = 250;
61	int batas4 =17;
62	float a_dekat5 = 0.0389122;
63	float b_dekat5 = 20.1244;
64	float a_jauh5 = 0.0232639;
65	float b_jauh5 = 18.7136;
66	float j_ref_dekat5 = 250;
67	float j_ref_jauh5 = 300;

Pada tabel 5.2 terdapat inisialisasi seluruh variabel global yang digunakan dalam sistem ini. Baris ke-1 merupakan pendefinisian nilai Pi yakni 3.14159265. baris ke-4 merupakan variabel yang berfungsi pada saat user akan memberhentikan program. Baris ke-5 dan ke-6 merupakan inisialisasi variabel mat yang berfungsi untuk menyimpan citra. Pada baris ke-7 terdapat inisialisasi ukuran dari *structure element* yang digunakan untuk proses *opening*. Baris ke-8 sampai ke-13 merupakan inisialisasi nilai YCbCr dari warna hijau lapangan. Y1, Cr1, dan Cb1 adalah nilai YCbCr minimal sedangkan Y2, Cr2, dan Cb2 adalah nilai YcbCr Maksimal. Baris ke-14 digunakan untuk membuat variabel yang berfungsi menyimpan informasi lingkaran yang terdeteksi. Informasi tersebut adalah jari-jari lingkaran, dan koordinat (x,y) lingkaran pada frame. Baris ke-15 sampai ke-20 merupakan inisialisasi nilai parameter yang digunakan pada fungsi *Hough Circle Transformation*. Baris ke-21 sampai ke-67 adalah inisialisasi variabel-variabel yang digunakan pada proses perhitungan jarak serta sudut arah bola.

5.2.3.1 Implementasi Kode Program Proses Akuisisi Data Citra

Tabel 5.3 Kode program proses akuisisi data citra

Baris	Kode Program
1	VideoCapture cap(1);
2	cap.set(CV_CAP_PROP_FRAME_WIDTH, 320);
3	cap.set(CV_CAP_PROP_FRAME_HEIGHT, 240);
4	cap.set(CV_CAP_PROP_FPS, 100);
5	if (cap.isOpened() == false) {
6	std::cout << "error: Webcam not accessed successfully\n\n";
7	return(0);
8	}



Baris	Kode Program
9	<code>while (charCheckForEscKey != 27 && cap.isOpened()) {</code>
10	<code>bool blnFrameReadSuccessfully=cap.read(matOriginal);</code>
11	<code>if (!blnFrameReadSuccessfully matOriginal.empty()){</code>
12	<code>std::cout << "eror: frame not read from webcam\n";</code>
13	<code>break;</code>
14	<code>}</code>

Tabel 5.3 adalah kode program untuk mengambil citra. Sesuai dengan sub bab 5.1.3.1, proses pengakuisisian data citra dimulai dengan pembuatan objek *videocapture* seperti yang ditunjukkan pada baris ke-1. Ukuran citra yang diambil sesuai dengan baris ke-2 dan ke-3 pada kode program tersebut yakni 320 x 240 *pixel*. Kemudian program akan mengecek apakah kamera berhasil di akses atau tidak. Jika gagal mengakses kamera maka program akan menampilkan pesan error: *Webcam no accessed sucessfully*. Namun jika berhasil maka akan lanjut ke proses akuisisi citra yang akan disimpan pada variabel dengan tipe *mat* yang bernama *matOriginal*. Fungsi untuk mengakuisisi data citra terdapat pada baris ke-10. Jika proses akuisisi ini gagal maka program akan menampilkan pesan error: *frame not read from webcam*.

5.2.3.2 Implementasi Kode Program *Preprocessing* Citra

Tabel 5.4 Kode program *preprocessing* citra

Baris	Kode Program
1	<code>cvtColor(matOriginal,matProcessed,CV_BGR2YCrCb);</code>
2	<code>inRange(matProcessed, Scalar(Y1, Cr1, Cb1) ,Scalar(Y2, Cr2, Cb2),matProcessed);</code>
3	<code>bitwise_not (matProcessed, matProcessed);</code>
4	<code>erode (matProcessed, matProcessed, getStructuringElement (MORPH_ELLIPSE, Size(morph_pixel, morph_pixel)));</code>
5	<code>dilate(matProcessed, matProcessed, getStructuringElement (MORPH_ELLIPSE, Size(morph_pixel, morph_pixel)));</code>

Seperti pada perancangan *preprocessing* citra, proses implementasi program *preprocessing* citra dimulai dengan proses konversi *color space* dari *BGR* ke *YCbCr*. Konversi ini menggunakan fungsi *cvtColor()* seperti yang terlihat pada tabel 5.11 baris ke 1. Pada baris ke-2 dilakukan *thresholding* atau *segmentasi* warna hijau menggunakan fungsi *inRange()*. Nilai warna yang disegmentasi adalah nilai yang di deklarasikan pada variabel global *Y1, Cr1, Cb1* dan *Y2, Cr2, Cb2*. Nilai tersebut merupakan nilai dari warna hijau lapangan rumput sintetis.pada penelitian ini, warna hijau lapangan yang terdeteksi akan didefinisikan sebagai 0 dan objek berwarna lain dilapangan akan bernilai 1. Hal ini utuk memudahkan pencarian objek bola yang berbentuk bulat pada lapangan pertandingan. Setelah dilakukan *thresholding*, pada *frame* masih nampak beberapa *noise* yang tidak



seharusnya ada. Oleh karena itu dilakukan operasi opening untuk meminimalisir *noise* kecil yang ada. Operasi *opening* ada pada baris ke-4 dan ke-5. Operasi *opening* adalah kombinasi proses dimana suatu citra digital dikenai operasi erosi dilanjutkan dengan dilasi. Operasi *opening* pada citra bertujuan memperhalus batas-batas objek, memisahkan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil daripada ukuran *structuring*. Operasi *opening* disini menggunakan *structure element* berbentuk *elips* dengan ukuran 5x5.

5.2.3.3 Implementasi Kode Program *Hough Circle Transformation*

Tabel 5.5 Kode program *Hough Circle Transformation*

Baris	Kode Program
1	<code>HoughCircles(matProcessed, v3fCircles, CV_HOUGH_GRADIENT, acc_res, min_dis, high_thres, low_thres, min_rad, max_rad);</code>
2	<code>for (int i = 0; i < v3fCircles.size(); i++) {</code>
3	<code> Diameter = (v3fCircles[i][2])*2; //Diameter bola</code>
4	<code> x_center =(int)v3fCircles[i][0]; //koordinat x bola</code>
5	<code> y_center =(int)v3fCircles[i][1]; //koordinat y bola</code>
6	<code> circle(matOriginal, Point(x_center,y_center), 1, Scalar(0, 255, 0), CV_FILLED);</code>
7	<code> circle(matOriginal, Point(x_center, y_center), (int)v3fCircles[i][2], Scalar(0, 0, 255),2);</code>
8	<code> line(matOriginal, Point(160,240), Point(x_center,y_center), Scalar(0, 255, 0), 2, CV_AA);</code>
9	<code>}</code>

Tabel 5.5 barisi kode program yang merupakan kode program utama dalam pendeteksian bola yaitu *Hough Circle Transformation*. Proses *Hough Circle Transformation* ini menggunakan fungsi OpenCV yakni *houghcircle()*. Pada fungsi tersebut terdapa beberapa parameter. Parameter pertama adalah citra yang akan dideteksi lingkarannya. Citra ini harus merupakan citra *8 bit single channel*. Parameter kedua adalah variabel vektor yang digunakan untuk menyimpan hasil dari proses *Hough Circle Transformation*. Vektor ini akan menyimpan informasi berupa koordinat x, koordinat y dan jari jari lingkaran. Parameter ketiga adalah metode *Hough Circle Transformation* yang digunakan pada OpenCV. Parameter keempat adalah ukuran akumulator yang digunakan untuk menyimpan lingkaran yang ditransformasi. Pada penelitian ini menggunakan ukuran 2 yang berarti ukuran akumulator 1,5 kali lebih besar dari ukuran citra. Parameter ke lima adalah minimum jarak dari tiap lingkaran yang terdeteksi. Semakin besar nilainya maka semakin sedikit lingkaran yang mungkin terdeteksi. Parameter ke enam adalah nilai *threshold* dari proses *canny edge detection* atau deteksi tepi citra. Parameter ke tujuh adalah *threshold vote* yang merupakan nilai minimal dari jumlah vote yang harus ada untuk sebuah titik pada akumulator dapat dikatakan sebagai titik



pusat lingkaran. Parameter ke delapan dan ke sembilan adalah parameter untuk menentukan nilai minimal dan maksimal jari-jari pada lingkaran yang dideteksi. Setelah terdeteksi adanya lingkaran, maka lingkaran tersebut akan dilingkari pada citra original menggunakan fungsi *circle()* seperti pada baris ke 7. Selain itu pada baris ke 8 juga akan menggambar garis antara posisi robot dan lingkaran yang terdeteksi.

5.2.3.4 Implementasi Kode Program Persamaan Linier Jarak Referensi

Tabel 5.6 Kode program Persamaan linier jarak referensi

Baris	Kode Program
1	<code>#include <iostream></code>
2	<code>#include <conio.h></code>
3	<code>#include <iomanip></code>
4	<code>#include <math.h></code>
5	<code>using namespace std;</code>
6	<code>int main(){</code>
7	<code>int korX[300];</code>
8	<code>int korY[300];</code>
9	<code>float X[300];</code>
10	<code>float Y[300];</code>
11	<code>float data, totalX, totalY, totalXY, totalXkuadrat,</code>
	<code>kuadrattotalX;</code>
12	<code>float a,b ;</code>
13	<code>int i, ulang;</code>
14	<code>cout << "FORMAT d = R r + D" ;</code>
15	<code>do{</code>
16	<code>totalX=0, totalY=0, totalXY=0, totalXkuadrat=0;</code>
17	<code>cout << endl << endl ;</code>
18	<code>cout << "Banyak data : " ;</code>
19	<code>cin >> data ;</code>
20	<code>cout << endl ;</code>
21	<code>for (i=0 ; i<data ; i++){</code>
22	<code>cout << "X " << i+1 << " : " ;</code>
23	<code>cin >> korX[i];</code>
24	<code>cout << "Y " << i+1 << " : " ;</code>
25	<code>cin >> korY[i];</code>
26	<code>X[i] = sqrt((pow((korX[i]-160),2)) + (pow((korY[i]-120),2)));</code>
27	<code>cout << "R " << i+1 << " : " << X[i] << endl;</code>
28	<code>cout << "D " << i+1 << " : " ;</code>
29	<code>cin >> Y[i];</code>
30	<code>cout << endl;</code>
31	<code>}</code>
32	<code>cout << endl;</code>
33	<code>for (i=0 ; i<data ; i++)</code>
34	<code>totalXY=totalXY+(X[i]*Y[i]);</code>
35	<code>for (i=0 ; i<data ; i++){</code>

Baris	Kode Program
36	totalX=totalX+X[i];
37	totalY=totalY+Y[i];
38	}
39	for (i=0 ; i<data ; i++)
40	totalXkuadrat=totalXkuadrat+(X[i]*X[i]);
41	for (i=0 ; i<data ; i++)
42	kuadrattotalX=totalX*totalX;
43	float b=((data*totalXY)-(totalX*totalY)) /
44	((data*totalXkuadrat)-(kuadrattotalX));
45	
46	float a=((totalXkuadrat*totalY)-(totalX*totalXY)) /
	((data*totalXkuadrat)-(kuadrattotalX));
47	cout << "Hasil : d = " << endl << b << "R"
	<< setiosflags(ios::showpos) << a << endl;
48	getch();
49	cout << "Ingin mengulang (Y/T) ?" ;
50	ulang=getch();
51	} while (ulang=='Y' ulang=='y');
52	return 0;
53	}

Pada tabel 5.6 terdapat kode program untuk mencari Persamaan linier dari masing-masing data pada jarak referensi. Pada program tersebut, user dapat menginputkan nilai koordinat x dan y dari objek dan menginputkan diameternya (D). Untuk mendapatkn nilai R dari setiap koordinat, digunakan Persamaan 5.4. Setelah nilai R di dapatkan maka program akan mencari nilai a dan b menggunakan Persamaan 5.5 dan 5.6. Setelah didapat nilai a dan b, maka sistem akan menghasilkan output Persamaan linier dengan format seperti persamaaan 5.3. Persamaan yang di dapat akan digunakan dalam perhitungan jarak bola yang sebenarnya.

5.2.3.5 Implementasi Kode Program Perhitungan Jarak Bola

Tabel 5.7 Kode program fungsi perhitungan jarak bola

Baris	Kode Program
1	class perhitungan {
2	public :
3	float jarakKeTengah(float x, float y){
4	float R = sqrt((pow((x-160),2)) + (pow((y-120),2)));
5	return R;
6	}
7	float diaDekat (float aDekat, float bDekat, float r){



Baris	Kode Program
8	float dDekat = ((aDekat * r) + bDekat);
9	return dDekat;
10	}
11	float diaJauh (float aJauh, float bJauh, float r){
12	float dJauh = ((aJauh * r) + bJauh);
13	return dJauh;
14	}
15	float jarak (float Dia, float d_dekat, float d_jauh,
16	float jr_dekat, float jr_jauh){
17	float Jar = (((jr_jauh - jr_dekat)*(Dia -
18	d_dekat))/(d_jauh - d_dekat))+
	jr_dekat;
	return Jar;
	};
	};

Tabel 5.7 barisi kode program yang merupakan sebuah kelas yang terdiri dari beberapa fungsi atau *method* yang digunakan untuk perhitungan jarak bola. Fungsi pertama pada baris ke-3 sampai ke-6 adalah fungsi Jarak ketengah. Fungsi ini digunakan untuk menghitung jarak koordinat objek ke titik pusat *frame* (160,120). Fungsi kedua pada baris ke-7 sampai ke-10 adalah fungsi diaDekat. Fungsi ini digunakan untuk menghitung diameter objek bola yang terdeteksi berdasarkan diameter referensi jarak dekat. Fungsi ketiga pada baris ke-11 sampai ke-14 adalah fungsi diaJauh. Fungsi ini digunakan untuk menghitung diameter objek bola yang terdeteksi berdasarkan diameter referensi jarak jauh. Selanjutnya fungsi yang terdapat pada baris ke-15 sampai ke-18 adalah fungsi jarak. Fungsi ini digunakan perhitungan jarak bola berdasarkan parameter-parameter yang sudah didapatkan sebelumnya. Fungsi ini menggunakan prinsip perbandingan untuk mendeteksi jarak dari objek bola. Fungsi jarak ini adalah fungsi utama dalam proses pendeteksian jara bola.

Tabel 5.8 Kode program pemanggilan fungsi perhitungan jarak bola

Baris	Kode Program
1	if (y_center >= batas1){
2	R = hitung.jarakKeTengah(x_center, y_center);
3	d_dekat = hitung.diaDekat(a_dekat1, b_dekat1, R);
4	d_jauh = hitung.diaJauh(a_jauh1, b_jauh1, R);
5	Jarak = hitung.jarak(Diameter , d_dekat, d_jauh,
6	j_ref_dekat1, j_ref_jauh1);
7	}
8	else if (y_center < batas1 && y_center >= batas2) {
9	R = hitung.jarakKeTengah(x_center, y_center);
10	d_dekat = hitung.diaDekat(a_dekat2, b_dekat2, R);
11	d_jauh = hitung.diaJauh(a_jauh2, b_jauh2, R);
12	Jarak = hitung.jarak(Diameter , d_dekat, d_jauh,
13	j_ref_dekat2, j_ref_jauh2);
14	}
15	else if (y_center < batas2 && y_center >= batas3) {
16	R = hitung.jarakKeTengah(x_center, y_center);



Baris	Kode Program
15	d_dekat = hitung.diaDekat(a_dekat3, b_dekat3, R);
16	d_jauh = hitung.diaJauh(a_jauh3, b_jauh3, R);
17	Jarak = hitung.jarak(Diameter, d_dekat, d_jauh, j_ref_dekat3, j_ref_jauh3);
18	}
19	else if (y_center < batas3 && y_center >= batas4) {
20	R = hitung.jarakKeTengah(x_center, y_center);
21	d_dekat = hitung.diaDekat(a_dekat4, b_dekat4, R);
22	d_jauh = hitung.diaJauh(a_jauh4, b_jauh4, R);
23	Jarak = hitung.jarak(Diameter, d_dekat, d_jauh, j_ref_dekat4, j_ref_jauh4);
24	}
25	else {
26	R = hitung.jarakKeTengah(x_center, y_center);
27	d_dekat = hitung.diaDekat(a_dekat5, b_dekat5, R);
28	d_jauh = hitung.diaJauh(a_jauh5, b_jauh5, R);
29	Jarak = hitung.jarak(Diameter, d_dekat, d_jauh, j_ref_dekat5, j_ref_jauh5);
30	}

Tabel 5.8 diatas merupakan kode program untuk memanggil fungsi perhitungan jarak dan sudut arah bola. Baris ke-1 sampai dengan baris ke-30 merupakan kode program untuk memanggil fungsi perhitungan jarak bola yan terdapat pada tabel 5.7. Untuk menghitung jarak bola, Frame akan dibagi menjadi beberapa bagian tergantung dari jarak bola maksimal yang dapat dideteksi sistem. Setiap bagian mewakili 50 cm jarak bola di lapangan. Jadi jika jarak bola yang dapat dideteksi sistem adalah 50 cm sampai 300 cm, maka frame akan dibagi menjadi 5 bagian dengan ukuran yang berbeda-beda tergantung kalibrasi dengan jarak sebenarnya di lapangan. Setiap bagian tersebut memiliki data posisi dan diameter bola yang berbeda beda.

5.2.3.6 Implementasi Kode Program Perhitungan Sudut Arah Bola

Tabel 5.9 Kode program fungsi perhitungan sudut arah bola

Baris	Kode Program
1	float sudut (float x_center, float y_center){
2	float Sud = float(atan((x_Sudut-x_center) /
3	(y_Sudut-y_center))*180/PI);
4	return Sud;
5	}

Pada Tabel 5.9 terdapat kode program yang merupakan fungsi untuk menghitung sudut arah bola. Berdasarkan perancangan, perhitungan sudut menggunakan parameter koordinat titik 0 cm yakni pada variabel x_Sudut dan y_sudut serta koordinat bola yakni x_center dan y_center. Untuk nilai PI sendiri adalah 3.14159265.

Tabel 5.10 Kode program pemanggilan fungsi perhitungan sudut arah bola

Baris	Kode Program
1	if (x_center > 160){

Baris	Kode Program
2	posisi = "kanan";
3	}
4	else {
5	posisi = "kiri";
6	}
7	Sudut = hitung.sudut(x_center,y_center);
8	if (Sudut < 0){
9	Sudut = -Sudut;
10	}
11	else {
12	Sudut = Sudut;
13	}

Tabel 5.10 merupakan tabel kode program untuk memanggil fungsi perhitungan arah sudut bola. Baris ke-1 sampai ke-6 merupakan kode program untuk menentukan posisi dari bola apakah di sebelah kanan atau di sebelah kiri sistem. Baris ke-7 merupakan pemanggilan fungsi perhitungan sudut. Baris ke-8 sampai ke 13 merupakan baris untuk membuat nilai dari perhitungan sudut menjadi nilai positif.

5.2.3.7 Implementasi Kode Program Menampilkan Jarak dan Sudut Arah Bola pada *Frame*

Tabel 5.11 Kode program menampilkan jarak dan sudut arah bola pada *frame*

Baris	Kode Program
1	stringstream j;
2	j << fixed << setprecision(2) <<Jarak;
3	S_Jarak = j.str();
4	putText(matOriginal, S_Jarak, Point(x_center, (y_center + 10)) , FONT_HERSHEY_SIMPLEX, .7, Scalar(255, 0, 0), 2,8,false);
5	putText(matOriginal, "cm", Point((x_center + 80), (y_center + 10)) , FONT_HERSHEY_PLAIN, .7, Scalar(255, 0, 0), 1,8,false);
6	stringstream x;
7	x << x_center;
8	string S_x_center = x.str();
9	putText(matOriginal, "X ", Point(185, 220), FONT_HERSHEY_PLAIN, .7, Scalar(0, 0, 0), 1,4, false);
10	putText(matOriginal, S_x_center, Point(195, 220) , FONT_HERSHEY_PLAIN, .7, Scalar(255, 255, 255), 1,4,false);
11	stringstream y;
12	y << y_center;
13	string S_y_center = y.str();
14	putText(matOriginal, "Y ", Point(185, 230) , FONT_HERSHEY_PLAIN, .7, Scalar(0, 0, 0), 1,4, false);
15	putText(matOriginal, S_y_center, Point(195, 230) , FONT_HERSHEY_PLAIN, .7, Scalar(255, 255, 255),

Baris	Kode Program
	<code>1,4,false);</code>
16	<code>stringstream d;</code>
17	<code>d << Diameter;</code>
18	<code>string S_Diameter = d.str();</code>
19	<code>putText(matOriginal, "D ", Point(245, 230) , FONT_HERSHEY_PLAIN, .7, Scalar(0, 0, 0), 1,4, false);</code>
20	<code>putText(matOriginal, S_Diameter, Point(285, 230), FONT_HERSHEY_PLAIN, .7, Scalar(255, 255, 255), 1,4,false);</code>
21	<code>putText(matOriginal, "Posisi ", Point(245, 220) , FONT_HERSHEY_PLAIN, .7, Scalar(0, 0, 0), 1,4, false);</code>
22	<code>putText(matOriginal, posisi, Point(285, 220) , FONT_HERSHEY_PLAIN, .7, Scalar(255, 255, 255), 1,4,false);</code>
23	<code>stringstream s;</code>
24	<code>s << fixed << setprecision(2) << Sudut;</code>
25	<code>S_Sudut = s.str();</code>
26	<code>putText(matOriginal, "Sudut ", Point(230, 210) , FONT_HERSHEY_PLAIN, .7, Scalar(0, 0, 0), 1,4, false);</code>
27	<code>putText(matOriginal, S_Sudut, Point(270, 210) , FONT_HERSHEY_PLAIN, .7, Scalar(255, 255, 255), 1,4,false);</code>
28	<code>line(matOriginal, Point(160,240), Point(160, 0), Scalar(0, 0, 0), 1, CV_AA);</code>
29	<code>imshow("OUTPUT", matOriginal);</code>
30	<code>cout << "\t X= " << x_center << "\t Y= " << y_center << "\t DIAMETER= " << Diameter << "\t JARAK= " << S_Jarak << "\t SUDUT= " << S_Sudut << endl;</code>

Tabel 5.9 diatas merupakan kode program untuk menampilkan hasil *output* dari Deteksi jarak bola ini. Pada frame output, bola yang terdeteksi akan dilingkari dan diberi keterangan jarak bola di sebelah kanannya. Pada frame juga akan diberi garis lurus yang membagi tampilan frame menjadi 2 bagian yakni kanan dan kiri. Selain informasi jarak bola, pada frame juga ditampilkan beberapa informasi lain seperti koordinat (x,y) bola, diameter bola yang terdeteksi, posisi arah bola dan sudut arah bola. Selain ditampilkan pada frame output, semua informasi tersebut juga dapat dilihat user melalui terminal.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dipaparkan proses pengujian, hasil pengujian, serta analisis hasil dari pengujian yang dilakukan berdasarkan sistem yang dibuat. Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Pengujian yang akan dilakukan antara lain pengujian fungsional yakni pengujian terhadap perangkat keras dalam hal ini adalah sensor kamera *webcam* Logitech C525 apakah dapat bekerja dengan baik, pengujian akurasi sistem yakni menguji seberapa akurat sistem yang telah dirancang dan diimplementasikan dibandingkan dengan nilai atau hasil yang sebenarnya, dan pengujian kecepatan sistem yakni menguji waktu pemrosesan ketika sistem mulai dijalankan hingga menghasilkan output nilai jarak. Berikut dijelaskan beberapa skenario pengujian yang dilakukan untuk menguji sistem.

6.1 Pengujian Raspberry Pi Mengakses *Webcam* Logitech C525

Webcam Logitech C525 adalah sensor utama dalam sistem ini yang berfungsi untuk mengambil citra. Pada sistem ini Raspberry Pi harus mampu mengakses *webcam* untuk mengambil citra.

6.1.1 Tujuan Pengujian

Tujuan dilakukannya Pengujian ini adalah untuk mengetahui apakah Raspberry Pi dapat mengakses kamera untuk mengambil citra atau tidak.

6.1.2 Prosedur Pengujian

Berikut Prosedur pengujian yang dilakukan untuk menguji Raspberry Pi dalam mengakses kamera *webcam* logitech C525 :

1. Menghubungkan Raspberry Pi 3 Model B dengan laptop menggunakan kabel UTP
2. Meremote GUI Raspberry Pi 3 Model B dengan menggunakan aplikasi Real VNC Viewer
3. Menjalankan kode program sistem pendeteksi jarak bola yang ada pada Raspberry Pi 3
4. Mengamati hasil pembacaan gambar obyek di monitor laptop untuk mengetahui tingkat keberhasilan dan kegagalan.

Adapun untuk mengukur jumlah keberhasilan dengan menggunakan **Persamaan (6.1)**

$$\text{Persentasi eror} = \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} \times 100 \% \quad (6.1)$$

6.1.3 Hasil Pengujian

Tabel 6.1 Hasil pengujian Raspberry Pi mengakses kamera Webcam Logitech C525

Pengujian Ke-	Hasil	Keterangan	Error (%)
1		Berhasil	0 %
2		Berhasil	0 %
3		Berhasil	0 %
4		Berhasil	0 %
5		Berhasil	0 %

Pengujian Ke-	Hasil	Keterangan	Error (%)
6		Berhasil	0 %
7		Berhasil	0 %
8		Berhasil	0 %
9		Berhasil	0 %
10		Berhasil	0 %

6.1.4 Analisis Hasil Pengujian

Berdasarkan Persamaan (6.1), untuk menentukan persentase eror pada setiap pengujian, diperoleh rata rata eror sebanyak 0%. Hal ini menunjukkan bahwa Raspberry Pi dapat mengakses kamera untuk mengambil citra dengan baik. Perhitungan persentase eror pengujian Raspberry Pi dalam mengakses kamera webcam C525 adalah sebagai berikut :



$$\begin{aligned}\text{Persentasi eror} &= \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} \times 100 \% \\ &= \frac{0}{10} \times 100 \% \\ &= 0\%\end{aligned}$$

Tidak ada eror yang dihasilkan, sehingga dapat dikatakan bahwa kamera logitech C525 dapat bekerja dengan sangat baik untuk mengambil citra.

6.2 Pengujian Akurasi Hasil Pendeteksian Jarak Bola

Sistem pendeteksi jarak bola dengan kamera pada robot kiper Sepak Bola menggunakan *Hough Circle Transformation* akan menghasilkan output berupa jarak antara robot dengan bola. Tujuan dari sistem ini adalah untuk mengetahui jarak antara robot kiper dengan bola sehingga robot dapat melakukan aksi yang efektif berdasarkan informasi jarak dari bola, oleh karena itu perlu diketahui tingkat keakuratan sistem ini dalam melakukan perhitungan jarak bola.

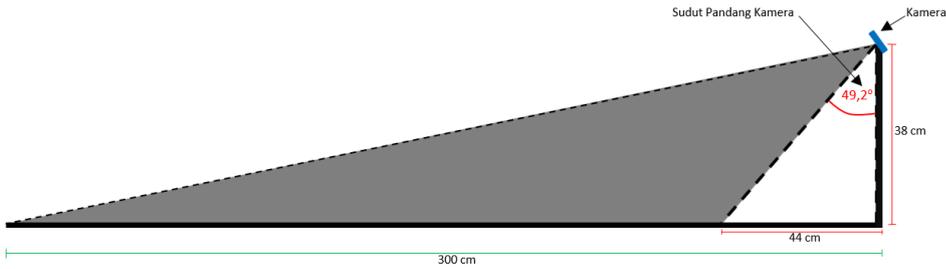
6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian terhadap akurasi hasil jarak pendeteksian bola adalah untuk menentukan nilai akurasi penggunaan metode *Hough Circle Transformation* untuk menentukan jarak dari bola berdasarkan nilai diameter bola yang terdeteksi.

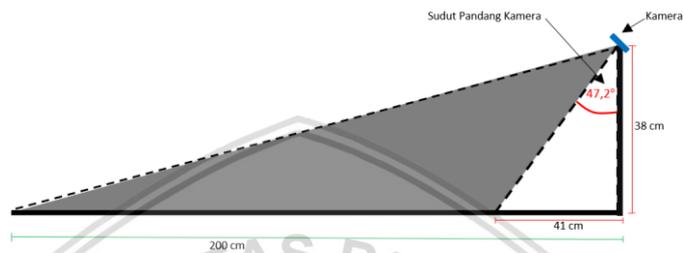
6.2.2 Prosedur Pengujian

Pengujian akurasi jarak bola dilakukan dengan tahapan sebagai berikut :

1. Menempatkan sistem pada posisi robot kiper sebenarnya dilapangan Pertandingan KRSBI Humanoid. Terdapat arah 2 skema pengujian yakni pengujian jarak skema 1 dengan sudut pandang kamera $49,2^\circ$ seperti pada Gambar 6.1. Pada Gambar tersebut dapat dilihat bahwa jarak terdekat yang ditangkap kamera adalah 44 cm dan jarak maksimal bola yang dapat di deteksi adalah 300 cm. Kemudian terdapat pengujian jarak skema 2 dengan sudut pandang kamera $47,2^\circ$ seperti pada Gambar 6.2. Pada Gambar tersebut dapat dilihat bahwa jarak terdekat yang ditangkap kamera adalah 41 cm dan jarak maksimal bola yang dapat di deteksi adalah 200 cm.



Gambar 6.1 Sudut pandang kamera pengujian jarak skema 1



Gambar 6.2 Sudut pandang kamera pengujian jarak skema 2

- Mengatur nilai batas Y di *frame* pada setiap jarak referensi. Nilai batas ini menentukan nilai d1 dan nilai d2 yang digunakan untuk perhitungan jarak bola. Untuk pengujian jarak dengan sudut pandang kamera 49,2° terdapat 4 nilai batas Y dan untuk pengujian dengan sudut pandang kamera 47,2° terdapat 3 nilai batas Y. Nilai batas Y pada setiap jarak referensi dapat dilihat pada Gambar 6.3 dan Gambar 6.4 berikut.

Y = 17	$d1 = 0,0389122 r + 20,1244$	$d2 = 0,0232639 r + 18,7136$	Jarak = 250 cm
Y = 29	$d1 = 0,0674082 r + 22,4401$	$d2 = 0,0389122 r + 20,1244$	Jarak = 200 cm
Y = 45	$d1 = 0,0500693 r + 32,951$	$d2 = 0,0674082 r + 22,4401$	Jarak = 150 cm
Y = 77	$d1 = 0,046262 r + 47,6737$	$d2 = 0,0500693 r + 32,951$	Jarak = 100 cm
	$d1 = 0,0528045 r + 90,084$	$d2 = 0,046262 r + 47,6737$	

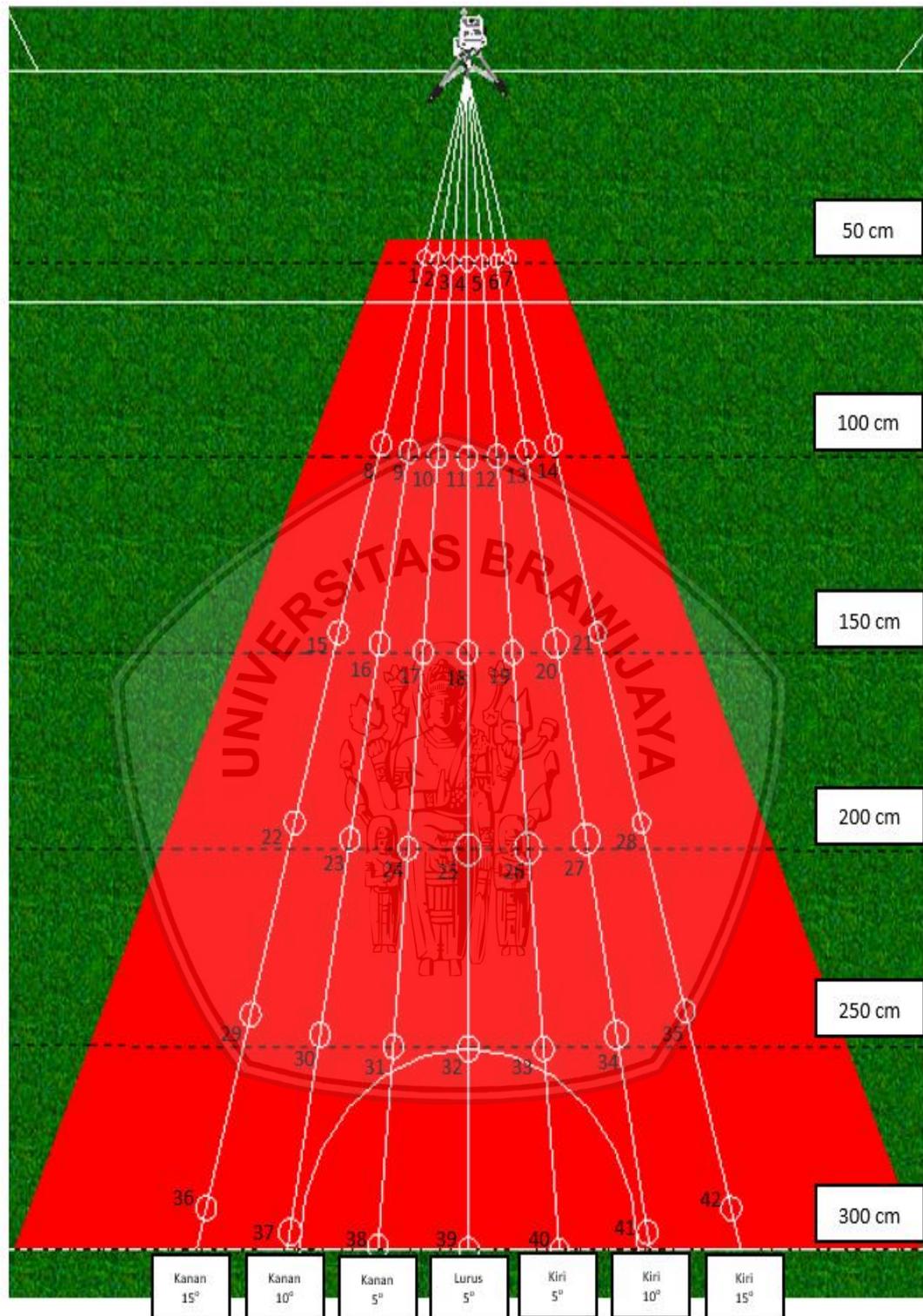
Gambar 6.3 Nilai batas Y pada pengujian jarak dengan sudut pandang kamera 49,2°



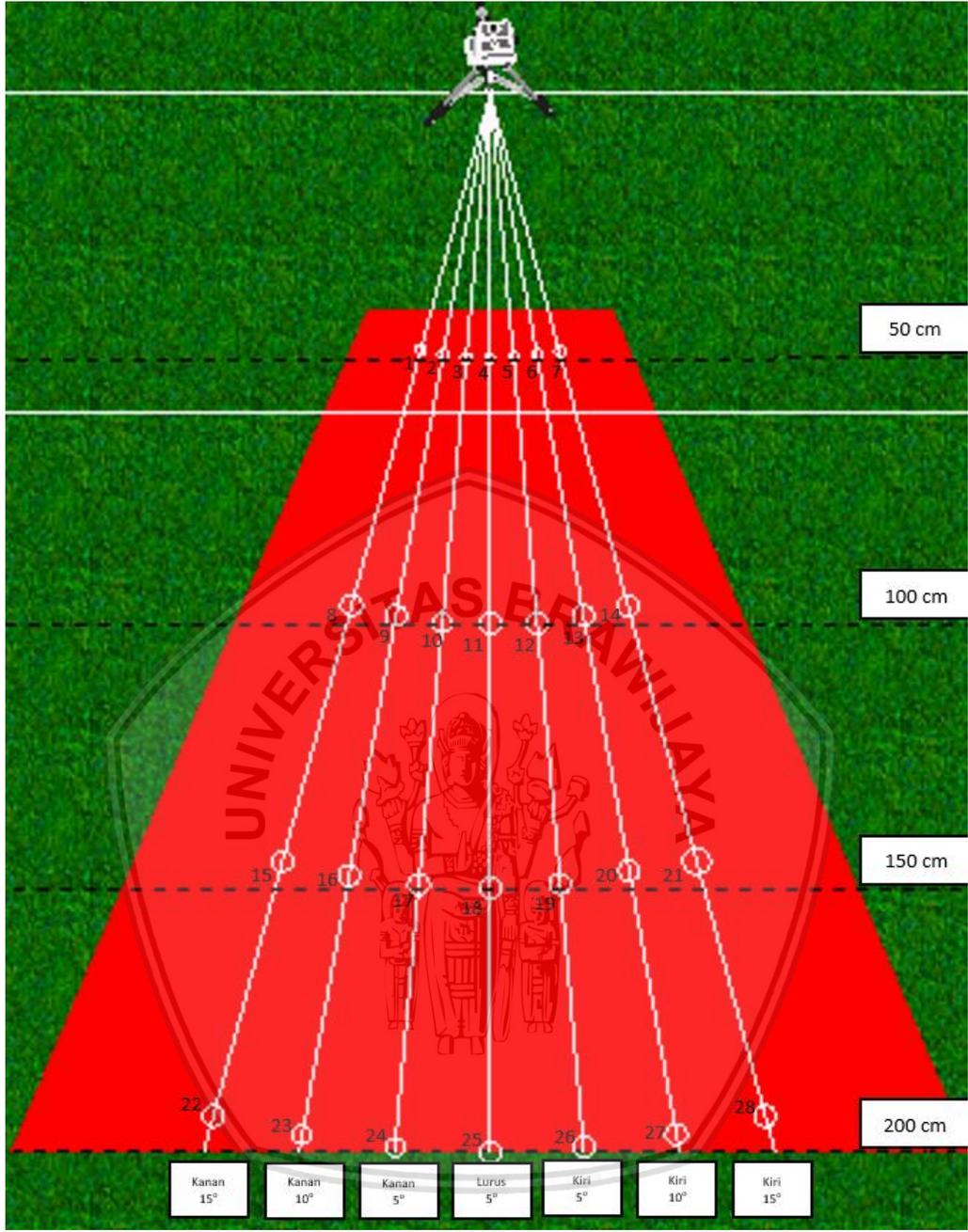
Y = 11	$d1 = 0,0674082 r + 22,4401$	$d2 = 0,0389122 r + 20,1244$	Jarak = 200 cm
Y = 27	$d1 = 0,0500693 r + 32,951$	$d2 = 0,0674082 r + 22,4401$	Jarak = 150 cm
Y = 58	$d1 = 0,046262 r + 47,6737$	$d2 = 0,0500693 r + 32,951$	Jarak = 100 cm
	$d1 = 0,0528045 r + 90,084$	$d2 = 0,046262 r + 47,6737$	

Gambar 6.4 Nilai batas Y pada pengujian jarak dengan sudut pandang kamera 47,2°

- Menjalankan program utama Deteksi jarak bola yang ada pada Raspberry Pi
- Menempatkan bola secara bergantian pada jarak bola yang dapat terlihat pada kamera dengan interval jarak kelipatan 50 cm. Arah bola dibagi menjadi 7 arah yakni lurus, 5° di sebelah kanan, 10° di sebelah kanan, 15° di sebelah kanan 5° di sebelah kiri, 10° di sebelah kiri, dan 15° di sebelah kiri. Penempatan posisi bola pada pengujian jarak dengan sudut pandang kamera 49,2° dan pengujian jarak dengan sudut pandang kamera 47,2° dapat dilihat pada Gambar 6.5 dan Gambar 6.6 berikut.



Gambar 6.5 Posisi bola pada pengujian jarak dengan sudut pandang kamera 49,2°



Gambar 6.6 Posisi bola pada pengujian jarak dengan sudut pandang kamera 47,2°

5. Mencatat 5 hasil pendeteksian jarak bola dari 5 frame di setiap posisi bola dan kemudian dihitung rata-ratanya sehingga mendapatkan rata-rata nilai jarak bola pada setiap posisi bola.
6. Untuk perhitungan persentase eror pada setiap jarak menggunakan **Persamaan 6.2** berikut

$$error = \frac{Jarak\ Hasil\ Ukur - Jarak\ Sebenarnya}{Jarak\ Sebenarnya} \times 100\% \tag{6.2}$$

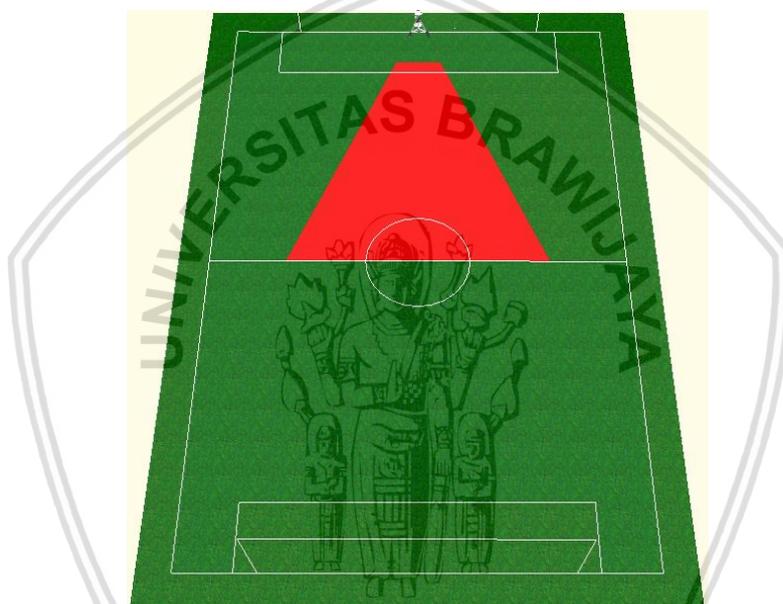
6.2.3 Hasil Pengujian

Tabel 6.2 Hasil pengujian jarak dengan sudut pandang kamera 49,2°

No	Posisi	Jarak Sebenarnya (cm)	Rataan Jarak Hasil Ukur (cm)	Eror Deteksi Jarak Bola (cm)	Persentase Eror (%)	Persentase Akurasi Deteksi Jarak Bola (%)
1	1	50	52,74	2,74	5,48	94,52
2	2	50	56,37	6,37	12,74	87,26
3	3	50	52,69	2,69	5,38	94,62
4	4	50	53,65	3,65	7,3	92,7
5	5	50	54,61	4,61	9,22	90,78
6	6	50	51,34	1,34	2,68	97,32
7	7	50	49,79	0,21	0,03	99,97
8	8	100	102,21	2,21	2,21	97,79
9	9	100	99,66	0,34	0,34	99,66
10	10	100	100,51	0,51	0,51	99,49
11	11	100	100,27	0,27	0,27	99,73
12	12	100	99	1	1	99
13	13	100	98,76	1,24	1,24	98,76
14	14	100	99,19	0,81	0,81	99,19
15	15	150	149,83	0,17	0,11	99,89
16	16	150	149,95	0,05	0,03	99,97
17	17	150	155,02	5,02	3,35	96,65
18	18	150	151,96	1,96	1,31	98,69
19	19	150	150,19	0,19	0,13	99,87
20	20	150	155,21	5,21	3,47	96,53
21	21	150	147,98	2,02	1,35	98,65
22	22	200	202,76	2,76	1,38	98,62
23	23	200	211,33	11,33	5,67	94,33
24	24	200	203,09	3,09	1,55	98,45
25	25	200	197,49	2,51	1,26	98,74
26	26	200	198,38	1,62	0,81	99,19
27	27	200	201,4	1,4	0,7	99,3
28	28	200	194,93	5,07	2,54	97,46
29	29	250	258,5	8,5	3,4	96,6
30	30	250	195,45	54,55	21,82	78,18
31	31	250	204,77	45,23	18,09	81,91
32	32	250	258,44	8,44	3,38	96,62
33	33	250	276,63	26,63	10,65	89,35
34	34	250	261,25	11,25	4,5	95,5
35	35	250	275,3	25,3	10,12	89,88
36	36	300	318,39	18,39	6,13	93,87
37	37	300	323,92	23,92	7,97	92,03
38	38	300	176,65	123,35	41,12	58,88

No	Posisi	Jarak Sebenarnya (cm)	Rataan Jarak Hasil Ukur (cm)	Eror Deteksi Jarak Bola (cm)	Persentase Eror (%)	Persentase Akurasi Deteksi Jarak Bola (%)
39	39	300	236,23	63,77	21,26	78,74
40	40	300	329,88	29,88	9,96	90,04
41	41	300	325,25	25,25	8,42	91,58
42	42	300	272,47	27,53	9,18	90,82
Rata-Rata				13,39	5,93	94,07

Tabel 6.2 merupakan Tabel hasil pengujian akurasi deteksi jarak bola pada pengujian jarak dengan sudut pandang kamera 49,2°. Pada tabel tersebut dapat dilihat bahwa nilai rata-rata eror pendeteksian jarak bola adalah sebesar 13,39 cm. Nilai rata-rata akurasi deteksi jarak bola pada pengujian jarak dengan sudut pandang kamera 49,2° adalah 94,07%.



Gambar 6.7 Bagian lapangan yang dapat dideteksi jaraknya pada pengujian jarak dengan sudut pandang kamera 49,2°

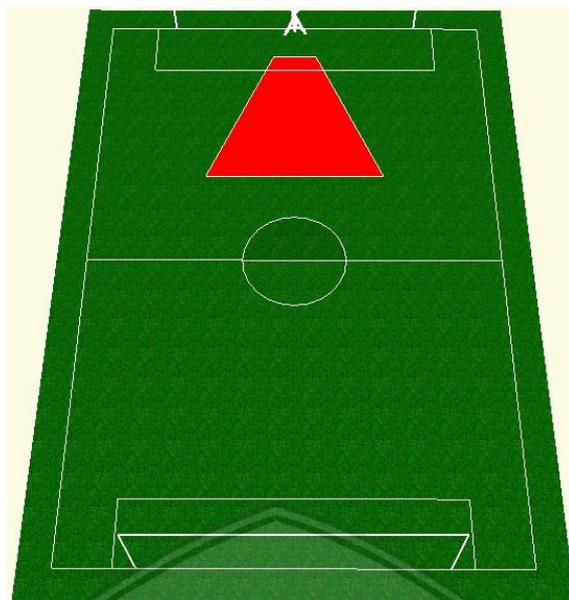
Gambar 6.7 merupakan Gambar bagian lapangan yang dapat dideteksi jaraknya oleh sistem. Jarak bola maksimal yang dapat dideteksi adalah 300 cm atau $\frac{1}{2}$ panjang lapangan.

Tabel 6.3 Hasil pengujian jarak dengan sudut pandang kamera 47,2°

No	Posisi	Jarak Sebenarnya (cm)	Rataan Jarak Hasil Ukur (cm)	Eror Deteksi Jarak Bola (cm)	Persentase Eror (%)	Persentase Akurasi Deteksi Jarak Bola (%)
1	1	50	53,72	3,72	7,44	92,56
2	2	50	49,03	0,97	1,94	98,06
3	3	50	54,9	4,9	9,8	90,2
4	4	50	49,67	0,33	0,66	99,34
5	5	50	50,12	0,12	0,24	99,76

No	Posisi	Jarak Sebenarnya (cm)	Rataan Jarak Hasil Ukur (cm)	Eror Deteksi Jarak Bola (cm)	Persentase Eror (%)	Persentase Akurasi Deteksi Jarak Bola (%)
6	6	50	50,17	0,17	0,34	99,66
7	7	50	53,87	3,87	7,74	92,26
8	8	100	102,27	2,27	2,27	97,73
9	9	100	101,66	1,66	1,66	98,34
10	10	100	98,53	1,47	1,47	98,53
11	11	100	101,07	1,07	1,07	98,93
12	12	100	97,58	2,42	2,42	97,58
13	13	100	101,68	1,68	1,68	98,32
14	14	100	99,02	0,98	0,98	99,02
15	15	150	150,66	0,66	0,44	99,56
16	16	150	149,27	0,73	0,49	99,51
17	17	150	149,48	0,52	0,35	99,65
18	18	150	149,55	0,45	0,3	99,7
19	19	150	154,55	4,55	3,03	96,97
20	20	150	153,23	3,23	2,15	97,85
21	21	150	150,83	0,83	0,55	99,45
22	22	200	210,32	10,32	5,16	94,84
23	23	200	197,44	2,56	1,28	98,72
24	24	200	196,73	3,27	1,64	98,36
25	25	200	196,55	3,45	1,73	98,27
26	26	200	203,53	3,53	1,77	98,23
27	27	200	204,15	4,15	2,08	97,92
28	28	200	208,02	8,02	4,01	95,99
Rata-rata				2,57	2,31	97,69

Tabel 6.3 merupakan Tabel hasil pengujian akurasi deteksi jarak bola pada pengujian jarak dengan sudut pandang kamera $47,2^\circ$. Pada tabel tersebut dapat dilihat bahwa nilai rata-rata eror pendeteksian jarak bola adalah sebesar 2,57 cm. Nilai rata-rata akurasi deteksi jarak bola pada pengujian jarak dengan sudut pandang kamera $47,2^\circ$ adalah 97,69%.



Gambar 6.8 Bagian lapangan yang dapat dideteksi jaraknya pada pengujian jarak dengan sudut pandang kamera $47,2^\circ$

Gambar 6.8 merupakan Gambar bagian lapangan yang dapat dideteksi jaraknya oleh sistem. Jarak bola maksimal yang dapat dideteksi adalah 200 cm atau $\frac{1}{3}$ panjang lapangan

6.2.4 Analisis Hasil Pengujian

Berdasarkan Hasil Pengujian jarak dengan sudut pandang kamera $49,2^\circ$ dan pengujian jarak dengan sudut pandang kamera $47,2^\circ$, maka akurasi tertinggi dan eror terendah pendeteksian jarak bola adalah pada pengujian jarak dengan sudut pandang kamera $47,2^\circ$. Pada pengujian jarak dengan sudut pandang kamera $47,2^\circ$ akurasi pendeteksian jarak bola adalah 97,69 %. Rata-rata eror pendeteksian jarak bola adalah 2,57 cm.

Pada pengujian dengan sudut pandang kamera $49,2^\circ$, akurasi yang dihasilkan cukup tinggi yakni 94,07 % dengan rata-rata eror 13,39 cm. Namun pada jarak di atas 200 cm tingkat eror pendeteksian bola cukup tinggi yakni 35,14 cm. dengan nilai eror yang besar tersebut, maka pendeteksian bola pada jarak di atas 200 cm kurang baik untuk diterapkan pada robot kiper Sepak Bola.

Akurasi jarak bola sangat ditentukan dengan diameter yang terdeteksi oleh sistem. Diameter yang dideteksi sangat bergantung pada pencahayaan di lapangan. Pencahayaan yang baik akan menghasilkan nilai jarak yang baik pula. Sistem mampu menghasilkan jarak yang cukup akurat dengan rata-rata eror yang cukup rendah, namun jarak yang dihasilkan pada setiap frame selalu berubah-ubah dan tidak stabil. Hal ini dikarenakan pendeteksian diameter lingkaran selalu berubah saat frame berganti.

6.3 Pengujian Akurasi Sudut Arah Bola

Dalam sitem deteksi jarak bola ini juga ditampilkan hasil pengukuran sudut dari arah bola. Sudut tersebut adalah sudut kemiringan bola terhadap arah lurus robot kedepan.

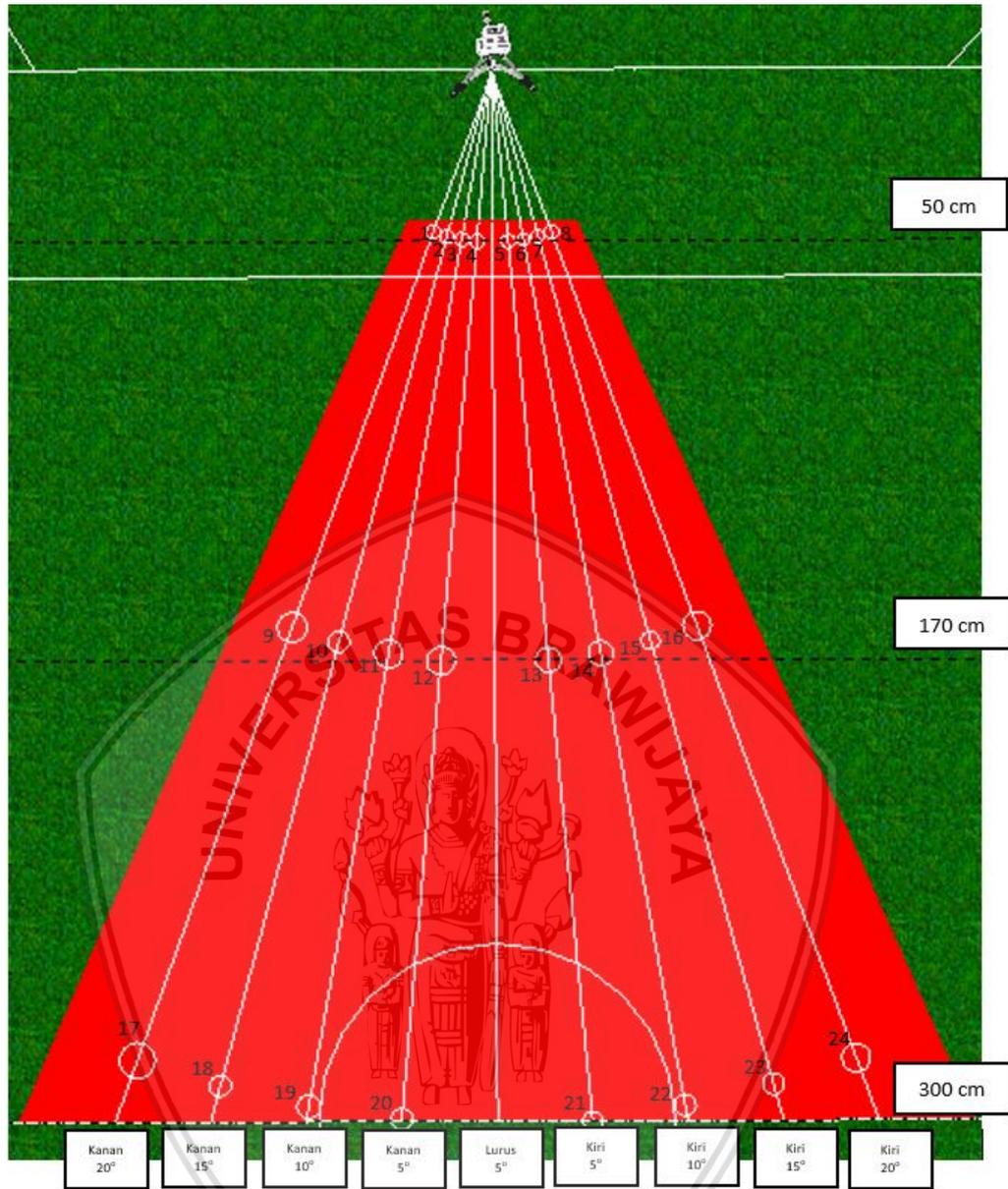
6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian akurasi sudut arah bola ini adalah untuk menentukan nilai akurasi perhitungan sudut arah bola pada sistem yang telah dibuat.

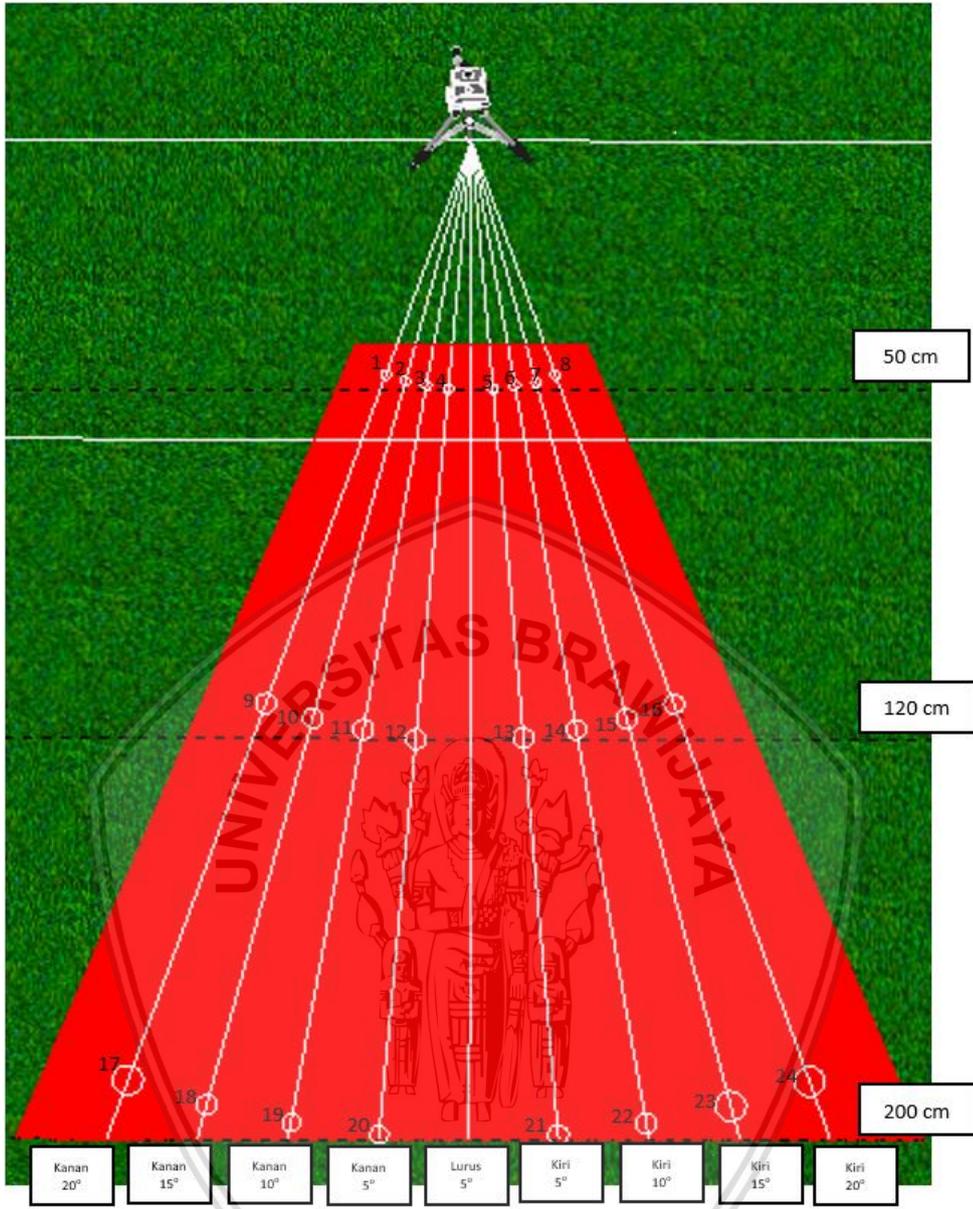
6.3.2 Prosedur Pengujian

Pengujian akurasi sudut arah bola dilakukan dengan tahapan sebagai berikut :

1. Menempatkan sistem pada posisi robot kiper sebenarnya dilapangan Pertandingan KRSBI Humanoid. Terdapat arah 2 skema pengujian yakni pengujian sudut arah bola skema 1 dengan sudut pandang kamera $49,2^\circ$ dan pengujian sudut arah bola skema 2 dengan sudut pandang kamera $47,2^\circ$. Sudut pandang kamera pada pengujian sudut arah bola skema 1 sama dengan sudut pandang kamera pada pengujian jarak skema 1 seperti pada Gambar 6.1. Sudut pandang kamera pada pengujian sudut arah bola skema 2 sama dengan sudut pandang kamera pada pengujian jarak skema 2 seperti pada Gambar 6.2.
2. Nilai batas Y pada pengujian sudut arah bola skema 1 sama dengan pengujian jarak skema 1 yakni sesuai Gambar 6.3. Sedangkan nilai batas Y pada pengujian sudut arah bola skema 2 sama dengan pengujian jarak skema 2 yakni sesuai Gambar 6.4. Pada pengujian dengan sudut pandang kamera $49,2^\circ$, untuk perhitungan sudut arah bola digunakan nilai X_{sudut} sebesar 160 dan Y_{sudut} sebesar 490, sedangkan pada pengujian dengan sudut pandang kamera $47,2^\circ$, untuk perhitungan sudut arah bola digunakan nilai X_{sudut} sebesar 160 dan Y_{sudut} sebesar 450.
3. Menjalankan program utama Deteksi jarak bola yang ada pada Raspberry Pi
4. Menempatkan bola secara bergantian di arah kanan dengan sudut 5° , 10° , 15° , 20° dan diarah kiri dengan sudut 5° , 10° , 15° , 20° . Pada pengujian dengan sudut pandang kamera $49,2^\circ$ dilakukan pengujian sudut arah bola dengan jarak bola 50 cm, 170 cm, dan 300 cm. Pada pengujian dengan sudut pandang kamera $47,2^\circ$ dilakukan pengujian sudut arah bola dengan jarak bola 50 cm, 120 cm, dan 200 cm. Penempatan posisi dan arah bola pada pengujian akurasi sudut dapat dilihat pada Gambar 6.9 dan Gambar 6.10 berikut.



Gambar 6.9 Posisi bola pada pengujian sudut arah bola dengan sudut pandang kamera 49,2°



Gambar 6.10 Posisi bola pada pengujian sudut arah bola dengan sudut pandang kamera 47,2°

5. Mencatat 5 hasil perhitungan sudut arah bola dari 5 frame di setiap posisi bola dan kemudian dihitung rata-ratanya sehingga mendapatkan rata-rata nilai sudut arah bola pada setiap posisi bola.
6. Untuk perhitungan eror pada setiap sudut menggunakan **Persamaan 6.3** berikut

$$error = \frac{Sudut\ Hasil\ Ukur - Sudut\ Sebenarnya}{Sudut\ Sebenarnya} \times 100\% \quad (6.3)$$

6.3.3 Hasil Pengujian

Tabel 6.4 Hasil pengujian sudut arah bola dengan sudut pandang kamera 49,2°

No	Posisi	Arah Bola	Sudut Sebenarnya (°)	Rata-Rata Sudut Hasil Ukur (°)	Eror Deteksi Sudut Arah Bola (°)	Persentase Eror (%)	Persentase Akurasi Deteksi Sudut Arah Bola (%)
1	1	Kanan	20	20,06	0,06	0,3	99,7
2	2	Kanan	15	14,53	0,47	2,35	97,65
3	3	Kanan	10	9,56	0,44	2,2	97,8
4	4	Kanan	5	5,07	0,07	0,35	99,65
5	5	Kiri	5	4,4	0,6	3	97
6	6	Kiri	10	8,88	1,12	5,6	94,4
7	7	Kiri	15	14,31	0,69	3,45	96,55
8	8	Kiri	20	19,3	0,7	3,5	96,5
9	9	Kanan	20	17,14	2,86	14,3	85,7
10	10	Kanan	15	13,22	1,78	8,9	91,1
11	11	Kanan	10	8,66	1,34	6,7	93,3
12	12	Kanan	5	4,67	0,33	1,65	98,35
13	13	Kiri	5	4,17	0,83	4,15	95,85
14	14	Kiri	10	7,91	2,09	10,45	89,55
15	15	Kiri	15	12,25	2,75	13,75	86,25
16	16	Kiri	20	16,93	3,07	15,35	84,65
17	17	Kanan	20	16,75	3,25	16,25	83,75
18	18	Kanan	15	12,95	2,05	10,25	89,75
19	19	Kanan	10	8,23	1,77	8,85	91,15
20	20	Kanan	5	4,2	0,8	4	96
21	21	Kiri	5	3,72	1,28	6,4	93,6
22	22	Kiri	10	7,29	2,71	13,55	86,45
23	23	Kiri	15	11,37	3,63	18,15	81,85
24	24	Kiri	20	15,69	4,31	21,55	78,45
Rata-rata					1,63	8,13	91,88

Tabel 6.4 merupakan Tabel hasil pengujian akurasi deteksi sudut arah bola pada pengujian sudut arah bola dengan sudut pandang kamera 49,2°. Pada tabel tersebut dapat dilihat bahwa nilai rata-rata eror pendeteksian sudut arah bola adalah sebesar 1,63°. Nilai rata-rata akurasi deteksi sudut arah bola pada pengujian sudut arah bola dengan sudut pandang kamera 49,2° adalah 91,88%.

Tabel 6.5 Hasil pengujian sudut arah bola dengan sudut pandang kamera 47,2°

No	Posisi	Arah Bola	Sudut Sebenarnya (°)	Rata-Rata Sudut Hasil Ukur (°)	Eror Deteksi Sudut Arah Bola (°)	Persentase Eror (%)	Persentase Akurasi Deteksi Sudut Arah Bola (%)
1	1	Kanan	20	22,15	2,15	10,75	89,25
2	2	Kanan	15	15,94	0,94	4,7	95,3

No	Posisi	Arah Bola	Sudut Sebenarnya (°)	Rata-Rata Sudut Hasil Ukur (°)	Eror Deteksi Sudut Arah Bola (°)	Persentase Eror (%)	Persentase Akurasi Deteksi Sudut Arah Bola (%)
3	3	Kanan	10	10,32	0,32	1,6	98,4
4	4	Kanan	5	5,2	0,2	1	99
5	5	Kiri	5	4,54	0,46	2,3	97,7
6	6	Kiri	10	10,51	0,51	2,55	97,45
7	7	Kiri	15	15,2	0,2	1	99
8	8	Kiri	20	21,94	1,94	9,7	90,3
9	9	Kanan	20	18,35	1,65	8,25	91,75
10	10	Kanan	15	13,94	1,06	5,3	94,7
11	11	Kanan	10	8,94	1,06	5,3	94,7
12	12	Kanan	5	4,28	0,72	3,6	96,4
13	13	Kiri	5	4,91	0,09	0,45	99,55
14	14	Kiri	10	9,1	0,9	4,5	95,5
15	15	Kiri	15	14,1	0,9	4,5	95,5
16	16	Kiri	20	18,64	1,36	6,8	93,2
17	17	Kanan	20	17,94	2,06	10,3	89,7
18	18	Kanan	15	13,79	1,21	6,05	93,95
19	19	Kanan	10	9,63	0,37	1,85	98,15
20	20	Kanan	5	4,9	0,1	0,5	99,5
21	21	Kiri	5	4,03	0,97	4,85	95,15
22	22	Kiri	10	8,66	1,34	6,7	93,3
23	23	Kiri	15	12,83	2,17	10,85	89,15
24	24	Kiri	20	17,17	2,83	14,15	85,85
Rata-rata					1,06	5,31	94,69

Tabel 6.5 merupakan Tabel hasil pengujian akurasi deteksi sudut arah bola pada pengujian sudut arah bola dengan sudut pandang kamera 47,2°. Pada tabel tersebut dapat dilihat bahwa nilai rata-rata eror pendeteksian sudut arah bola adalah sebesar 1,06°. Nilai rata-rata akurasi deteksi sudut arah bola pada pengujian sudut arah bola dengan sudut pandang kamera 47,2° adalah 94,69%.

6.3.4 Analisis Hasil Pengujian

Berdasarkan Hasil Pengujian sudut sudut arah bola dengan sudut pandang kamera 49,2° dan pengujian sudut arah bola dengan sudut pandang kamera 47,2°, maka akurasi tertinggi dan eror terendah pendeteksian sudut bola adalah pada pengujian sudut arah bola dengan sudut pandang kamera 49,2°. Pada pengujian sudut arah bola dengan sudut pandang kamera 47,2° akurasi pendeteksian sudut arah bola adalah 94,69 %. Rata-rata eror pendeteksian sudut arah bola adalah 1,06°.

Berdasarkan hasil pengujian akurasi sudut, akurasi sudut yang dihasilkan sistem cukup tinggi namun sistem hanya mampu mendeteksi bola dengan jarak

maksimal 20° . Hal ini dikarenakan jika sudah diatas 20° bola yang terlihat oleh kamera hanya sebagian sehingga sulit untuk dideteksi sudutnya.

6.4 Pengujian Waktu Pemrosesan Sistem

Waktu pemrosesan sistem adalah waktu yang dibutuhkan sistem untuk melakukan 1 kali siklus perhitungan jarak bola.

6.4.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui waktu yang dibutuhkan sistem untuk melakukan proses pendeteksian jarak menggunakan *Hough Circle Transformation*. Hal ini diperlukan untuk mengetahui performansi dari sistem yang telah dibuat.

6.4.2 Prosedur Pengujian

Pengujian waktu pemrosesan sistem dilakukan dengan menguji waktu yang dibutuhkan sistem untuk melakukan pemrosesan sampai dengan mendapatkan jarak bola pada setiap *frame* citra. Pengujian ini dilakukan dengan menggunakan fungsi `clock_gettime` yang dapat mengukur waktu dengan ketelitian hingga *nanosecond*. Fungsi `clock_gettime` menggunakan struktur data *timespec* yang mana didefinisikan pada *header time.h*.

6.4.3 Hasil Pengujian

Tabel 6.6 Waktu eksekusi program pada pengujian jarak dengan sudut pandang kamera $49,2^\circ$

No	Posisi	Jarak Bola (cm)	Sudut Arah Bola ($^\circ$)	Rata-Rata Waktu Komputasi (ms)
1	1	50	Kanan 15°	161,8
2	2	50	Kanan 10°	274,2
3	3	50	Kanan 5°	132,6
4	4	50	Lurus $^\circ$	143
5	5	50	Kiri 5°	142,2
6	6	50	Kiri 10°	114
7	7	50	Kiri 15°	137,8
8	8	100	Kanan 15°	123,8
9	9	100	Kanan 10°	181,4
10	10	100	Kanan 5°	160
11	11	100	Lurus $^\circ$	174,6
12	12	100	Kiri 5°	153,6
13	13	100	Kiri 10°	152,4
14	14	100	Kiri 15°	152,4
15	15	150	Kanan 15°	187,6
16	16	150	Kanan 10°	148

No	Posisi	Jarak Bola (cm)	Sudut Arah Bola (°)	Rata-Rata Waktu Komputasi (ms)
17	17	150	Kanan 5°	149,4
18	18	150	Lurus°	119,8
19	19	150	Kiri 5°	151,8
20	20	150	Kiri 10°	140
21	21	150	Kiri 15°	229,2
22	22	200	Kanan 15°	241,4
23	23	200	Kanan 10°	152,8
24	24	200	Kanan 5°	146,8
25	25	200	Lurus°	150,4
26	26	200	Kiri 5°	141,8
27	27	200	Kiri 10°	125
28	28	200	Kiri 15°	195
29	29	250	Kanan 15°	125,8
30	30	250	Kanan 10°	215,8
31	31	250	Kanan 5°	185,6
32	32	250	Lurus°	150,2
33	33	250	Kiri 5°	192,4
34	34	250	Kiri 10°	110,2
35	35	250	Kiri 15°	145,8
36	36	300	Kanan 15°	130
37	37	300	Kanan 10°	125,8
38	38	300	Kanan 5°	170,6
39	39	300	Lurus°	127,8
40	40	300	Kiri 5°	152,8
41	41	300	Kiri 10°	160
42	42	300	Kiri 15°	138,4
Rata-Rata				157,48

Tabel 6.2 merupakan tabel hasil pengujian waktu perhitungan jarak bola pada pengujian jarak dengan sudut pandang kamera 49,2° oleh sistem pada setiap frame yang diambil oleh kamera. Pada tabel tersebut terlihat bahwa rata rata waktu pemrosesan sistem pada pengujian jarak dengan sudut pandang kamera 49,2° adalah 157,48 ms.

Tabel 6.7 Waktu eksekusi program pada pengujian jarak dengan sudut pandang kamera 47,2°

No	Posisi	Jarak Bola (cm)	Sudut Arah Bola (°)	Rata-Rata Waktu Komputasi (ms)
1	1	50	Kanan 15°	188,4
2	2	50	Kanan 10°	153,6
3	3	50	Kanan 5°	122,4
4	4	50	Lurus°	170,2

No	Posisi	Jarak Bola (cm)	Sudut Arah Bola (°)	Rata-Rata Waktu Komputasi (ms)
5	5	50	Kiri 5°	152,2
6	6	50	Kiri 10°	202,8
7	7	50	Kiri 15°	162,8
8	8	100	Kanan 15°	231,6
9	9	100	Kanan 10°	158
10	10	100	Kanan 5°	113,4
11	11	100	Lurus°	175,8
12	12	100	Kiri 5°	154
13	13	100	Kiri 10°	111,6
14	14	100	Kiri 15°	118,2
15	15	150	Kanan 15°	148,8
16	16	150	Kanan 10°	146,6
17	17	150	Kanan 5°	110,6
18	18	150	Lurus°	207,2
19	19	150	Kiri 5°	161,6
20	20	150	Kiri 10°	135,8
21	21	150	Kiri 15°	150,2
22	22	200	Kanan 15°	148,8
23	23	200	Kanan 10°	150,8
24	24	200	Kanan 5°	188,6
25	25	200	Lurus°	158,6
26	26	200	Kiri 5°	160,4
27	27	200	Kiri 10°	217,2
28	28	200	Kiri 15°	168,2
Rata-rata				159,59

Tabel 6.3 merupakan tabel hasil pengujian waktu perhitungan jarak bola pada pengujian jarak dengan sudut pandang kamera 47,2° oleh sistem pada setiap frame yang diambil oleh kamera. Pada tabel tersebut terlihat bahwa rata rata waktu pemrosesan sistem pada pengujian jarak dengan sudut pandang kamera 47,2° adalah 159,59 ms.

6.4.4 Analisis Hasil Pengujian

Berdasarkan hasil pengujian waktu pemrosesan sistem, maka dapat dilihat bahwa waktu pemrosesan pengujian jarak dengan sudut pandang kamera 47,2° lebih cepat daripada waktu pemrosesan pengujian jarak dengan dengan sudut pandang kamera 49,2°. Pada Tabel 6.26 dan Tabel 6.27 diperoleh 2 waktu rata rata pada 2 skema pengujian jarak. Jadi waktu rata rata pemrosesan deteksi jarak bola ini adalah sebagai berikut :

$$\text{Rata-Rata} = \frac{\text{waktu pengujian jarak dengan sudut pandang kamera } 49,2^\circ + \text{waktu pengujian jarak dengan sudut pandang kamera } 47,2^\circ}{2}$$

$$= \frac{157,48 + 159,59}{2}$$
$$= 158,54 \text{ ms}$$

Hasil tersebut cukup baik mengingat dalam pengimplementasian pada pertandingan, dibutuhkan sistem yang cepat dan *realtime*. Sistem yang *realtime* dapat membuat aksi robot menjadi responsif terhadap bola.



BAB 7 KESIMPULAN DAN SARAN

Pada bab ini akan dipaparkan mengenai kesimpulan dan saran yang didapat dari hasil pengujian sistem dan analisis dari hasil pengujian yang telah dilakukan.

7.1 Kesimpulan

Berdasarkan tahapan penelitian yang telah dilakukan yakni perancangan, implementasi, pengujian dan juga analisis dari hasil pengujian yang telah dilakukan, maka penulis dapat menyimpulkan bahwa :

1. Deteksi jarak bola ini dirancang berdasarkan bentuk dan ukuran dari robot kiper Sepak Bola Fakultas Ilmu Komputer Universitas Brawijaya. Penempatan kamera berada pada 38 cm diatas permukaan lapangan. Titik 0 cm dari pengukuran ini sesuai dengan posisi bagian telapak kaki robot kiper Sepak Bola Fakultas Ilmu Komputer Universitas Brawijaya yakni 2 cm dibelakang titik tegak lurus kamera terhadap lapangan pertandingan.
2. Deteksi jarak bola ini diimplementasikan menggunakan kamera Logitech C525 dan Raspberry Pi 3 Model B. Metode *Hough Circle Transformation* digunakan untuk mencari diameter dan koordinat dari bola pada *frame*. Kemudian untuk mengimplementasikan perhitungan jarak bola, digunakan Persamaan perbandingan diameter dan koordinat bola pada setiap jarak dilapangan yang terdeteksi.
3. Setelah melakukan pengujian akurasi jarak dan sudut arah bola pada 2 sudut pandang kamera yang berbeda, pendeteksian sudut arah bola dengan akurasi terbaik adalah pada saat sudut pandang kamera $47,2^\circ$ dengan tingkat akurasi yaitu 97,69 % dan rata rata eror pendeteksian jarak bola yaitu 2,57 cm. Untuk deteksi sudut arah bola, pendeteksian sudut arah bola dengan akurasi terbaik adalah pada saat sudut pandang kamera $47,2^\circ$ dengan tingkat akurasi yaitu 94,69 % dan rata rata eror pendeteksian sudut arah bola yaitu $1,06^\circ$.
4. Performansi Deteksi jarak bola pada robot kiper Sepak Bola menggunakan *Hough Circle Transformation* ini memiliki kecepatan rata-rata untuk memproses nilai jarak dan sudut arah bola pada setiap frame yakni 158,54 ms.

7.2 Saran

Terdapat beberapa saran yang dapat menjadi acuan untuk pengembangan penelitian yang terkait dan sejenis dengan penelitian ini yaitu sebagai berikut :

1. Menggunakan metode lain untuk mendeteksi ukuran diameter suatu bola yang lebih stabil dan akurat untuk hasil pengukuran jarak yang lebih akurat.

2. Menggunakan algoritma lain untuk perhitungan jarak bola menggunakan citra digital.
3. Menggunakan algoritma lain untuk perhitungan sudut arah bola sehingga nilai sudut yang dihasilkan lebih akurat
4. Untuk penelitian kedepan disarankan untuk membuat Deteksi jarak bola dengan kondisi kamera bergerak sehingga pandangan robot kiper menjadi lebih luas namun tetap dengan memperhatikan akurasi jarak bola.



DAFTAR PUSTAKA

- Anon., 2018. *What is a webcam?*. [Online]
Available at: <https://www.digitalunite.com/guides/email-skype/webcams/what-webcam>
- Cherabit, N., Chelali, . F. Z. & Djeradi, A., 2012. Circular Hough Transform for Iris localization.
- Dwi, M., 2016. *pengertian pengolahan citra digital*. [Online]
Available at: <http://mahadwi.com/pengertian-pengolahan-citra-digital/>
- Hamilton, E., 1992. *JPEG File Interchange Format*. Version 1.02 ed. Milpitas: s.n.
- ImageJ, 2017. *ImageJ*. [Online]
Available at: <https://imagej.net/Hough Circle Transform>
- Isnain, N. R. & Budi, S., 2013. Computer Vision Untuk Penghitungan Jarak Objek Terhadap Kamera. p. 6.
- Jen, J. R., Shie, M. C. & Chen, C., 2006. A Circular Hough Transform Hardware for Industrial Circle Detection Applications.
- KBBI, K., 2018. *Arti kata jarak menurut KBBI*. [Online]
Available at: <https://kbbi.kata.web.id/jarak/>
- KRSBI, P. P., 2018. *Buku Panduan KRSBI Humanoid 2018*. s.l.:s.n.
- Logitech, 2017. *HD Webcam c525*. [Online]
Available at: <https://www.logitech.com/id-id/product/hd-webcam-c525>
- Logitech, 2018. *Logitech*. [Online]
Available at: http://support.logitech.com/en_us/product/hd-webcam-c525/specs
[Accessed 2018].
- Loughnane, C., 2012. *measuring angles in opencv*. [Online]
Available at: <https://pdnotebook.com/measuring-angles-in-opencv-2f0551b8dd5a>
[Accessed 29 mei 2018].
- M. M., 2015. *A Real-Time Face to Camera Distance Measurement Algorithm Using Object Classification*. s.l.:s.n.
- Najmurokhman, A., Kusnandar, Wibowo, B. H. & Abiddin, . S., 2016. Desain dan Realisasi Pemrosesan Citra dengan CMUcam4 untuk Meningkatkan Kinerja Visual Robosoccer.

- OpenCV, 2017. *OpenCV*. [Online]
Available at: <https://opencv.org/>
- OpenCV, 2018. *Platforms*. [Online]
Available at: <https://opencv.org/platforms/>
- opensource.com, 2018. *What is a Raspberry Pi?*. [Online]
Available at: <https://opensource.com/resources/what-raspberry-pi>
- Pamungkas, E. M., Sumbodo, B. A. A. & Candradewi, I., 2017. Sistem Pendeteksi dan Pelacakan Bola dengan Metode Hough Circle Transform, Blob Detection, dan Camshift Menggunakan AR.Drone. *IJEIS*, p. 11.
- pi, R., 2016. *Raspberry Pi 3 Model B*. [Online]
Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Sinha, U., n.d. *Circle Hough Transform*. [Online]
Available at: <http://www.aishack.in/tutorials/circle-hough-transform/>
[Accessed 18 April 2018].
- team, O., 2017. *OpenCV*. [Online]
Available at: <https://opencv.org/about.html>
- Uji, A., 2018. *Sensor Jarak*. [Online]
Available at: <http://www.alatuji.com/kategori/300/sensor-jarak>
- UMY, 2018. *8 Tim KRSBI Humanoid Lakukan Uji Coba Bersama*. [Online]
Available at: <http://www.umy.ac.id/8-tim-krsbi-humanoid-lakukan-uji-coba-bersama.html>
[Accessed 19 Juli 2018].
- UMY, 2018. *UMY Berikan Kualitas Terbaik untuk Venue KRI 2018*. [Online]
Available at: <http://www.umy.ac.id/umy-berikan-kualitas-terbaik-untuk-venue-kri-2018.html>
[Accessed 19 Juli 2018].