LAPORAN KERJA PRAKTIK

PENGOLAHAN CITRA PADA ROBOT SEPAK BOLA BERODA

KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

Oleh:

ZENDI ZAKARIA RAGA PERMANA

16410200033

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2019

LAPORAN KERJA PRAKTIK

PENGOLAHAN CITRA PADA ROBOT SEPAK BOLA BERODA

Diajukan sebagai salah satu syarat untuk menempuh ujian Tahap Akhir

Program Strata Satu(S1)

Disusun Oleh :

Nama	:	Zendi Zakaria Raga Permana
Nim	:	16.41020.0033
Program	:	S1 (Strata Satu)
Jurusan	:	Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2019

“Salah satu cara untuk menikmati hidup adalah dengan mengucap Syukur”

UNIVERSITAS
Dinamika

Kupersembahkan untuk Ibu, Ayah dan Kakak saya, terimakasih untuk kasih sayangnya sampai detik ini. Dan juga ketiga adik saya terimakasih atas dukungan dan keceriaan yang selalu dibagikan. Bersama teman-teman Tim DINAMIKA ROBOTIC Terimakasih atas bantuan dan dukunganya yang menjadikan saya sampai sejauh ini.

UNIVERSITAS Dinamika

LAPORAN KERJA PRAKTIK

PENGOLAHAN CITRA PADA ROBOT SEPAK BOLA BERODA

Laporan Kerja Praktik oleh

Zendi Zakaria Raga Permana

NIM : 16.41020.0033

Telah diperiksa, diuji dan disetujui

Surabaya, 11 Desember 2019

Disetujui :

Dosen Pembimbing,	Penyelia,
Pauladie Susanto,.S.Kom,.M.T	**Dr. Susijanto Tri Rasmana, S.Kom.,M.T.**
NIDN. 0729047501	NIDN. 0727097302

Mengetahui :

Ketua Prodi S1 Teknik Komputer

Fakultas Teknologi dan Informatika
UNIVERSITAS Dinamika

Pauladie Susanto,.S.Kom,.M.T

NIDN. 0729047501

UNIVERSITAS Dinamika

Scanned with CamScanner

v

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama	: Zendi Zakaria Raga Permana
NIM	: 16410200033
Program Studi	: S1 Teknik Komputer
Fakultas	: Fakultas Teknologi dan Informatika
Jenis Karya	: Laporan Kerja Praktik
Judul Karya	: **PENGOLAHAN CITRA PADA ROBOT SEPAK BOLA BERODA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right)* atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data *(database)* untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 29 Agustus 2019

Yang menyatakan

Zendi Zakaria Raga Permana
NIM : 16410200033

vi

ABSTRAK

Penggunaan pengolahan citra pada bidang Robotika bukanlah hal yang baru di kalangan pelajar maupun dalam Industri. Pengolahan citra membantu robot untuk menentukan objek yang dipilih sebagai target robot dengan memanfaatkan fungsi dari kamera. Akan tetapi pengolahan citra mempunyai tingkat kesulitan dan kerumitan sendiri bagi orang yang masih awan atau mulai mempelajarinya.

Kerja praktik ini akan menjelaskan bagaimana melakukan proses pengolahan citra yang akan diterapkan pada robot sepak bola beroda. Dimana robot dapat mendeteksi bola berwarna orange menggunakan kamera Logitech 920 Pro dan menggunakan Mini PC Raspberry Pi 3 B+. Proses dari pendeteksian bola berwarna orange menerapkan Citra HSV yang digunakan untuk membedakan warna bola dengan warna sekitar bola dan dilanjutkan ke dalam proses Citra Biner yang digunakan untuk menentukan bola yang menjadi objek pendeteksian.

Pengujian dilakukan dengan cara bola diletakkan pada jarak yang bervariasi dari kamera. *Edge detection* dengan metode *Canny* dilakukan untuk menguji apakah dengan perubahan posisi bola, sistem mampu menangani perubahan bentuk dari letak dan posisi bola. Hasil dari pengujian sistem mampu mendeteksi bola dengan jarak minimal 60 cm dan jarak maksimal 640 cm.

Kata kunci : Pengolahan Citra, Robot Sepak Bola Beroda, *Canny edge detection*.

UNIVERSITAS Dinamika

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas segala berkat dan rahmat yang telah diberikan, sehingga penulis dapat menyelesaikan Laporan Kerja Praktik. Penulisan Laporan ini adalah sebagai salah satu syarat menempuh Tugas Akhir pada Program Studi S1 Teknik Komputer Universitas Dinamika.

Dalam usaha menyelesaikan penulisan Laporan Kerja Praktik ini penulis banyak mendapat bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Tuhan YME, karena berkat dan rahmatnya penulis dapat menyelesaikan Laporan Kerja Praktik ini.
2. Orang Tua dan Seluruh Keluarga penulis tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Kerja Praktik serta Laporan ini.
3. Kepada Pauladie Susanto, S.Kom., M.T. selaku Ketua Program Studi Teknik Komputer Surabaya atas ijin yang telah diberikan untuk dapat melaksanakan Kerja Pratek meriset robot di Lab Robotika.
4. Kepada Dr.Susijanto Tri Rasmana, S.Kom., M.T. yang telah membimbing selama Kerja Praktik ini yang telah memberi banyak masukan tentang Robotika.
5. Teman-teman seperjuangan Teknik Komputer angkatan 2016.
6. DINAMIKA ROBOTIC atas segalanya kesempatan dan waktunya selama ini yang telah menemani saya dalam Kerja Praktik ini.

UNIVERSITAS Dinamika

7. Kepada adik keempat saya terimakasih atas selalu memberikan sewnyuman disaat suka maupun duka.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh, karena intu penulis sanngat mengharapkan sran dan kritik untuk memperbaiki kekurangan untuk lebih tidak lagi.

Surabaya, 29 Agustus 2019

Penulis

UNIVERSITAS Dinamika

DAFTAR ISI

Halaman

ABSTRAK vi

KATA PENGANTAR viii

DAFTAR ISI x

DAFTAR TABEL xiii

DAFTAR GAMBAR xiv

DAFTAR LAMPIRAN xvi

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Perumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan 2

1.5 Kontribusi 3

BAB II GAMBARAN UMUM PERUSAHAAN 4

2.1 Sejarah Singkat Universitas Dinamika 4

2.2 Visi dan Misi Universitas Dinamika 6

2.2.1 Visi 6

2.2.2 Misi 6

2.3 Lokasi Perusahaan 7

BAB III LANDASAN TEORI 8

3.1 Citra Digital 8

UNIVERSITAS Dinamika

3.1.1 Citra RGB 9

3.1.2 Citra HSV 9

3.1.3 Citra Biner 11

3.2 OpenCV 12

3.3 Camera Logitech 13

3.4 Raspberry Pi 3 B+ 14

3.5 Robot SYNA SWC8 16

3.5.1 Base Robot 17

3.5.2 Mekanika Dribbling 19

3.5.3 Motor penggerak 19

3.5.4 Letak kamera robot 20

BAB IV DISKRIPSI PEKERJAAN 21

4.1 Prosedur Penelitian 21

4.2 Analisa Kebutuhan 21

4.2.1 Pengisian *Operating System* Raspberry Pi B 3+ 22

A. Menggunakan Aplikasi Win32 Imager 22

B. Download Operating System Raspberry Pi 3 B+ 22

C. Pengisian OS ke SD Card 23

4.2.2 Konfigurasi OpenCV 24

4.3 Desain Flowchart Pengolahan Citra 26

4.4 Implementasi 31

4.5 Analisa Hasil 31

UNIVERSITAS Dinamika

4.6 Hasil dan Pembahasan 31
4.6.1 Pembacaan Webcame dengan OpenCV 31
4.6.2 Pembahasan Pengolahan Citra Robot 35
4.6.3 Nilai HSV 47
4.6.4 Nilai Pegeseran *Pixel* 48
4.6.5 Nilai pergseseran *pixel* menjauhi kamera 49
4.6.6 Grafik Pergeseran *pixel* dengan Koordinat x 50
4.6.7 Grafik Pergeseran *pixel* dengan Koordinat y 51
4.6.8 Grafik Pergeseran *pixel* x menjauhi kamera 52
4.6.9 Grafik Pergeseran *pixel* y menjauhi kamera 53
BAB V PENUTUP 54
5.1 Kesimpulan 54
5.2 Saran 55
DAFTAR PUSTAKA 56
LAMPIRAN 57
BIODATA PENULIS 65

UNIVERSITAS Dinamika

DAFTAR TABEL

Tabel 3.1 Spesifikasi Logitech C922 13

Tabel 4.1 Penjelasan Fungsi Citra OpenCV 39

Tabel 4.2 Nilai HSV bola dan Koordinat Bola 47

Tabel 4.3 Nilai pergeseran Nilai pixel 48

Tabel 4.4 Nilai pergeseran pixel menjauhi kamera..... 49

UNIVERSITAS Dinamika

DAFTAR GAMBAR

Gambar 2.1 Universitas Dinamika 7

Gambar 3.1 Titik awal 2D pada Citra Digital 8

Gambar 3.2 Ruang Warna Citra RGB 9

Gambar 3.3 Geometri Silinder Ruang Warna HSV 10

Gambar 3.4 Citra Warna Biner 11

Gambar 3.5 Logo OpenCV 13

Gambar 3.6 Camera Logitech C922 Pro 14

Gambar 3.7 Raspberry Pi 3 B+ 15

Gambar 3.8 PIN GPIO Raspberry Pi 3 B+ 16

Gambar 3.9 Robot SYNA SWC8 17

Gambar 3.10 Base Robot 18

Gambar 3.11 Base Robot tampak depan 18

Gambar 3.12 Dribbling Robot SYNA SWC8 19

Gambar 3.13 Motor Penggerak Robot 20

Gambar 3.14 Letak Kamera Robot 20

Gambar 4.0.1 Prosedur Penelitian 21

Gambar 4.0.2 Logo Win32 Disk Imager 22

Gambar 4.0.3 Website Raspberry Pi 3 B+ 23

Gambar 4.0.4 SD Card Adapter dan SD Card 16 GB 23

Gambar 4.0.5 Tampilan Win32 Disk Imager 24

Gambar 4.0.6 Perintah persiapan OpenCV 24

Gambar 4.0.7 Download OpenCV 25

Gambar 4.0.8 Proses Unzip OpenCV 25

UNIVERSITAS Dinamika

Gambar 4.0.9 Pencarian File OpenCV 25

Gambar 4.0.10 Penyisipan OpenCV 25

Gambar 4.0.11 Compile OpenCV 26

Gambar 4.0.12 Proses akhir penginstalan OpenCV 26

Gambar 4.0.13 Konversi Ruang RGB (a) ke Ruang HSV (b) 27

Gambar 4.0.14 Penentuan Objek HSV (a) ke target objek (b)..... 28

Gambar 4.0.15 Filtering (a), pendeteksian garis tepi (b) 28

Gambar 4.0.16 Deteksi garis tepi (a), deteksi lingkaran (b) 29

Gambar 4.0.17 Flowchart Pengolahan Citra 30

Gambar 4.0.18 Percobaan Baca Webcame 32

Gambar 4.0.19 Trackbar OpenCV 37

Gambar 4.0.20 Tampilan Deteksi bola 47

UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1 Form KP-3 (Surat Balasan Perusahaan) 57

Lampiran 2 Form KP-5 (Acuan Kerja) 58

Lampiran 3 Form KP-6 (Log Harian dan Catatan Perubahan Acuan Kerja)..... 60

Lampiran 4 Form KP-7 (Kehadiran Kerja Praktik) 62

Lampiran 5 Kartu Bimbingan Kerja Praktik..... 64

UNIVERSITAS Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

RoboCup Middle Size League adalah suatu pertandingan tingkat Internasional dimana yang diperlombakan adalah robot sepak bola beroda yang mampu memiliki kecerdasan untuk dapat berfikir untuk dapat membedakan objek bola, gawang, lapangan dan objek yang telah ditentukan sebelumnya. Karena robot yang diperlombakan adalah robot yang harus dapat berfikir secara berkembang dimana robot dapat menentukan keputusan dimana terdapat objek yang robot cari dan yang bukan robot cari di dalam lingkungan yang seharusnya. Hal seperti ini akan menjadi suatu masalah jika robot tidak mempunyai kemampuan untuk menentukan mana objek yang telah ditentukan dan yang bukan ditentukan.

Robot Sepak Bola yang dimiliki Universitas Dinamika saat ini menggunakan camera webcame Logitech C922 Pro dimana memiliki sudut pandang 78°. Dengan keterbatasan sudut pandang robot perlu adanya bantuan pergerakan mekanik robot dengan tujuan robot dapat mencari bola saat robot kehilangan object bola yang dicari. Titik koordinat bola pada suatu citra dapat dipastikan dengan pembacaan data nilai x, y dan r dari bola tersebut dengan satuan nilai *pixel*. Namun, data nilai x,y dan r yang telah didapat dari pembacaan nilai *pixel* dari kamera tidak dapat langsung dipergunakan perlu adanya proses kalibrasi dan pengolahan citra yang lebih lanjut sehingga menghasilkan nilai citra yang dibutuhkan untuk melakukan pergerakan dari robot.

UNIVERSITAS Dinamika

Maka, perlu adanya proses pengolahan citra lebih lanjut yang mampu diterapkan oleh robot yang nantinya robot akan hanya mendeteksi objek bola saja yang berwarna orange. Terdapat banyak metode yang dapat dipergunakan dalam pengolahan Citra adalah Segmentasi HSV.

Metode segmentasi warna dengan deteksi warna HSV untuk mengasilkan objek segmen citra berupa blob sehingga dapat terdeteksi oleh perangkat komputer. Proses deteksi objek akan mengolah segmen warna yang dihasilkan oleh proses segmentasi sehingga dapat diketahui banyaknya objek terdeteksi, luas area dan titik pusat tiap objek.

1.2 Perumusan Masalah

Bagaimana cara mendeteksi posisi bola terhadap robot berdasarkan data kamera menggunakan Raspberry Pi 3 B+?

1.3 Batasan Masalah

Melihat permasalahan yang ada, maka penulis membatasi masalah dari kerja praktik, yaitu:

1. Menggunakan Camera Webcame C922 Pro.
2. Mini PC Raspberry Pi 3 B+.
3. Menggunakan Library pengolahan citra OpenCV.

1.4 Tujuan

Tujuan dari kegiatan Kerja Praktik yang dilaksanakan oleh mahasiswa adalah agar mahasiswa dapat melihat secara langsung bagaimana kondisi dan kenyataan di lapangan. Serta melatih analisa, tentang bagaimanakah cara menyelesaikan

UNIVERSITAS Dinamika

permasalahan menggunakan ilmu yang didapatkan pada perkuliahan. Tujuan khusus adalah sebagai berikut :

1. Mengakses Webcame C9220 Pro menggunakan Raspberry Pi 3 B+.
2. Memahami teori pengolahan citra yang terdapat pada OpenCV.
3. Mencoba dan menganalisa hasil pengolahan citra dan hasil deteksi bola.
4. Memberikan cara mengkonfigurasi dan proses pada pendeteksian robot sepak bola untuk mendeteksi objek yaitu bola berwarna orange.

1.5 Kontribusi

Memberikan kontribusi ke Tim Robotika Stikom Surabaya yaitu dengan membuat dokumen yang dapat dibaca dan dapat di implementasikan untuk Tim supaya dapat dikembangkan lebih lagi atas apa yang telah penulis capai.

UNIVERSITAS Dinamika

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah Singkat Universitas Dinamika

Di tengah kesibukan derap Pembangunan Nasional, kedudukan informasi semakin penting. Hasil suatu pembangunan sangat ditentukan oleh materi informasi yang dimiliki oleh suatu negara. Kemajuan yang dicitakan oleh suatu pembangunan akan lebih mudah dicapai dengan kelengkapan informasi. Cepat atau lambatnya laju pembangunan ditentukan pula oleh kecepatan memperoleh informasi dan kecepatan menginformasikan kembali kepada yang berwenang.

Kemajuan teknologi telah memberikan jawaban akan kebutuhan informasi, komputer yang semakin canggih memungkinkan untuk memperoleh informasi secara cepat, tepat dan akurat. Hasil informasi canggih ini telah mulai menyentuh kehidupan kita. Penggunaan dan pemanfaatan komputer secara optimal dapat memacu laju pembangunan. Kesadaran tentang hal inilah yang menuntut pengadaan tenaga-tenaga ahli yang terampil untuk mengelola informasi, dan pendidikan adalah salah satu cara yang harus ditempuh untuk memenuhi kebutuhan tenaga tersebut.

Atas dasar pemikiran inilah, maka untuk pertama kalinya di wilayah Jawa Timur dibuka Pendidikan Tinggi Komputer, Akademi Komputer & Informatika Surabaya (AKIS) pada tanggal 30 April 1983 oleh Yayasan Putra Bhakti berdasarkan SK Yayasan Putra Bhakti No. 01/KPT/PB/III/1983. Tokoh pendirinya pada saat itu adalah :

1. Laksda. TNI (Purn) Mardiono

2. Ir.Andrian A. T
3. Ir. Handoko Anindyo
4. Dra. Suzana Surojo
5. Dra. Rosy Merianti, Ak

Kemudian berdasarkan rapat BKLPTS tanggal 2-3 Maret 1984 kepanjangan AKIS dirubah menjadi Akademi Manajemen Informatika & Komputer Surabaya yang bertempat di jalan Ketintang Baru XIV/2. Tanggal 10 Maret 1984 memperoleh Ijin Operasional penyelenggaraan program Diploma III Manajemen Informatika dengan surat keputusan nomor: 061/Q/1984 dari Direktorat Jendral Pendidikan Tinggi (Dikti) melalui Koordinator Kopertis Wilayah VII. Kemudian pada tanggal 19 Juni 1984 AKIS memperoleh status TERDAFTAR berdasar surat keputusan Direktorat Jendral Pendidikan Tinggi (Dikti) nomor: 0274/O/1984 dan kepanjangan AKIS berubah lagi menjadi Akademi Manajemen Informatika & Teknik Komputer Surabaya. Berdasar SK Dirjen DIKTI nomor: 45/DIKTI/KEP/1992, status DIII Manajemen Informatika dapat ditingkatkan menjadi DIAKUI.

Waktu berlalu terus, kebutuhan akan informasi juga terus meningkat. Untuk menjawab kebutuhan tersebut AKIS ditingkatkan menjadi Sekolah Tinggi dengan membuka program studi Strata 1 dan Diploma III jurusan Manajemen Informatika. Dan pada tanggal 20 Maret 1986 nama AKIS berubah menjadi STIKOM SURABAYA , singkatan dari Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya berdasarkan SK Yayasan Putra Bhakti nomor: 07/KPT/PB/03/86 yang selanjutnya memperoleh STATUS TERDAFTAR pada tanggal 25 Nopember 1986 berdasarkan Keputusan Mendikbud nomor:

0824/O/1986 dengan menyelenggarakan pendidikan S1 dan D III Manajemen Informatika. Di samping itu STIKOM SURABAYA juga melakukan pembangunan gedung Kampus baru di jalan Kutisari 66 yang saat ini menjadi Kampus II STIKOM SURABAYA . Peresmian gedung tersebut dilakukan pada tanggal 11 Desember 1987 oleh Bapak Wahono Gubernur Jawa Timur pada saat itu.

2.2 Visi dan Misi Universitas Dinamika

2.2.1 Visi

Menjadi Perguruan Tinggi yang Berkualitas, Unggul, dan Terkenal.

2.2.2 Misi

1. Mengembangkan ipteks sesuai dengan kompetisi.
2. Membentuk SDM yang profesional, unggul dan berkompetensi.
3. Menciptakan corporate yang sehat dan produktif.
4. Meningkatkan kepedulian sosial terhadap kehidupan bermasyarakat.
5. Menciptakan lingkungan hidup yang sehat dan produktif.

UNIVERSITAS Dinamika

2.3 Lokasi Perusahaan

Gambar 2.1 Universitas Dinamika

UNIVERSITAS
Dinamika

BAB III

LANDASAN TEORI

3.1 Citra Digital

Pada sebuah citra digital tercipta dari proses digitalisasi yang dari setiap bagiannya terdiri dari proses sampling dan kuantisasi. Sampling bertujuan untuk membagi bagian layar untuk menjadi layar kotak-kotak kecil hasil dan memiliki nilai tertentu. Satu bagian citra dihasilkan oleh sekumpulan titik yang masing-masing di representasikan oleh koordinat x dan y yang memiliki sebuah nilai. Bisa katakan lain dengan sebutan *pixel*. Maka, dengan ditemukannya *pixel* dapat dinyatakan dengan sebuah fungsi dua dimensi *f(x,y)* dengan titik awal terletak pada pada pojok kiri atas citra. Dapat diilustrasikan seperti pada Gambar 3.1 yang menjelaskan bahwa sumbu x bergerak menjauh ke kanan dari titik awal dan sumbu y bergerak menjauh ke bawah dari titik awal.

UNIVERSITAS Dinamika

Gambar 3.1 Titik awal 2D pada Citra Digital

3.1.1 Citra RGB

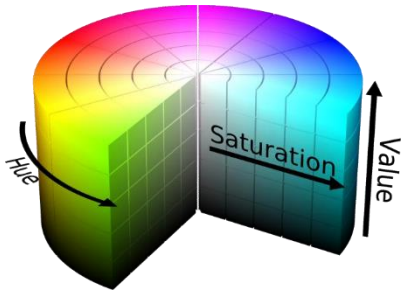
Pada dasarnya mata manusia mampu membedakan warna merah, hijau, dan biru *(Red, Green, Blue)*. Maka dari itu, warna merupakan kombinasi dari ketiga warna tersebut. Metode ruang warna RGB ini seringkali ditemui pada sistem grafika komputer. Bahwa dalam cara merepresentasikan tingkat suatu kecerahan sebuah kanal matrix dapat dinyatakan dalam sebuah tingkat kedalaman warna yang dinyatakan dalam n-bit (Prakoso, 2017). Pada Gambar 3.2 merupakan ilustrasi tuang warna RGB.

Gambar 3.2 Ruang Warna Citra RGB

3.1.2 Citra HSV

Citra HSV merupakan sebuah singkatan dari Hue, Saturation, dan Value adalah yang awalnya citra warna RGB yang direpresentasikan kembali ke dalam geometri HSV. Dengan proses ini bertujuan untuk lebih dijabarkan atau dibedakan. Citra warna HSV direpresentasikan dalam geometri silinder yang dapat dilihat pada Gambar 3.3. Saturation merupakan variable yang vibrancy dari suatu warna, saturation bisa disebut juga dengan purity. Semakin kecil nilai dari Saturation, maka warna yang ditampilkan condong ke warna abu – abu. Skala nilai dari Saturation

berkisar 0% hingga 100%. Value menunjukkan nilai kecerahan dari suatu warna. Skala value berkisar antara 0% hingga 100%.



Gambar 3.3 Geometri Silinder Ruang Warna HSV

Perhitungan konversi RGB menjadi HSV dapat dirumuskan sebagai berikut :

$$H = \tan\left[\frac{3(G-B)}{(R-G)+(R-B)}\right] \tag{3.1}$$

$$S = 1 - \frac{min(R,G,B)}{V} \tag{3.2}$$

$$V = \frac{R+G+B}{3} \tag{3.3}$$

Pada rumus di atas, apabila S = 0, maka H tidak dapat ditentukan. Untuk itu perlu normalisasi RGB terlebih dahulu dengan rumus sebagai berikut :

$$r = \frac{R}{R+G+B} \tag{3.4}$$

$$g = \frac{G}{R+G+B} \tag{3.5}$$

$$b = \frac{B}{R+G+B} \tag{3.6}$$

Setelah nilai r,g, dan b sudah dinormalisasi, maka rumus untuk transformasi RGB ke HSV sebagai berikut :

$$V = \max(r,g,b) \tag{3.7}$$

UNIVERSITAS Dinamika

$$S = \begin{cases} 0 & \text{Jika V = 0} \\ V - \dfrac{min(r,g,b)}{V} & \text{Jika V > 0} \end{cases} \quad (3.8)$$

$$H = \begin{cases} 0 & \text{Jika S = 0} \\ \dfrac{60 * (g-b)}{S * V} & \text{Jika V = r} \\ 60 * \left[2 + \dfrac{(b-r)}{S * V}\right] & \text{Jika V = g} \\ 60 * \left[4 + \dfrac{(r-g)}{S * V}\right] & \text{Jika V = b} \end{cases} \quad (3.9)$$

$$H = H + 30 \qquad \text{Jika H < 0} \quad (3.10)$$

Model memisahkan komponen intensitas dari citra warna, sehingga model ini merupakan model yang ideal untuk mengembangkan algoritma pemrosesan citra yang intuitif dan natural (Rakmawati, 2013).

3.1.3 Citra Biner

Citra biner adalah citra yang hanya memiliki 2 nilai saja yaitu 0 dan 1 yang berarti putih atau hitam. Pada penerapan citra biner umumnya digunakan untuk dapat memisahkan antara objek yang dituju dengan latar belakang objek yang diabaikan. Objek ditunjukan pada warna putih dan latar belakang ditunjukan dengan warna hitam. Pada Gambar 2.4 merupakan ilustrasi citra biner.

Gambar 3.4 Citra Warna Biner

UNIVERSITAS Dinamika

3.2 OpenCV

OpenCV merupakan sebuah library open source untuk kepentingan vision komputer maupun untuk kepentingan robotika yang telah disediakan dengan link http://SourceForge.net/projects/opencvlibrary. Library yang ditulis dalam bahasa C dan C++ dan dapat dijalankan pada sistem operasi Linux, Windows dan Mac OS X. Selain itu terdapat pengembangan pada Python dan Ruby. OpenCV adalah singkatan dari Open Source Computer Vision. OpenCV merupakan sebuah software bebas atau open source yang digunakan untuk operasi computer vision dan machine leraning. OpenCV telah dibangun untuk menyediakan sebuah infrastruktur umum untuk beberapa aplikasi computer vision dan untuk mempercepat penggunaan dari mesin persepsi dalam produk komersial. OpenCV mempermudah bisnis-bisnis untuk memanfaatkan dan memodifikasi kode. Library OpenCV mempunyai lebih dari 2500 algoritma yang telah dioptimalkan dimana meliputi sebuah himpunan menyeluruh dari keduanya yaitu klasik dan seni beberapa algoritma computer vision dan machine learning. Algoritma-algoritma tersebut dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasi tindakan manusia dalam video, mengikuti jejak perpindahan objek, mengekstrak model-model 3D objek, menghasilkan titik awan 3D dari kamera stereo, dan lain sebagainya.

OpenCV dapat diterapkan pada pemograman C++, C dan Python, Java dan MATLAB. OpenCV mendukung untuk sistem operasi Window, Linux, Android dan Mac OS. Logo OpenCV dilihat pada Gambar 3.5

UNIVERSITAS Dinamika

Gambar 3.5 Logo OpenCV

3.3 Camera Logitech

Camera Logitech C922 Pro Stream merupakan jenis Camera Webcame yang seringkali digunakan untuk keperluan pengambilan data video *streaming* maupun sekedar untuk mengambil gambar. Bentuk fisik camera dapat dilihat pada Gambar 3.6. Camera ini mempunyai spesifikasi ukuran 1080p pada 30 fps, *autofocus*, dan bidang pandang 78 derajat. Untuk mengetahui lebih jelas terdapat Tabel sebagai spesifikasi dari camera webcame logitech C922 ini.

Tabel 3.1 Spesifikasi Logitech C922

Video quality : Full HD 1080p @30fps, HD 720p @30/60fps
Compatibility : Windows 7, Windows 8, Windows 10, macOS X 10.9 or higher, Xbox, Chrome OS.
Works with : Xsplit, OBS
Height (without clip): 1.14″ x 3.74″ x 0.94″
Height (with clip): 1.73″ x 3.74″ 2.80″
Weight (including clip and cable): 5.71oz (162g)
Cable Length: 5ft
Diagonal Field of View (FOV): 78 degrees

UNIVERSITAS Dinamika

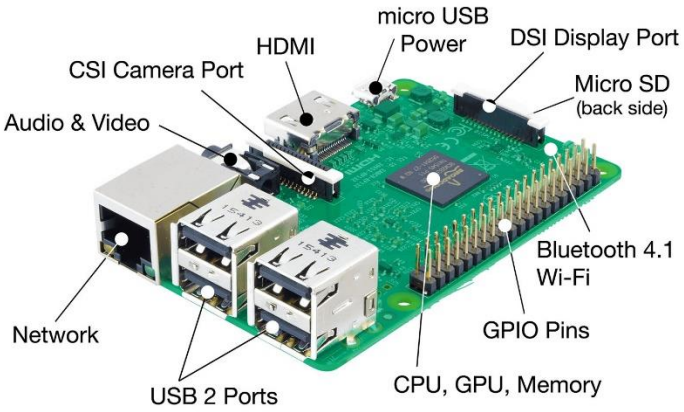
Gambar 3.6 Camera Logitech C922 Pro

3.4 Raspberry Pi 3 B+

Raspberry Pi, sering disingkat dengan nama Raspi, adalah komputer papan tunggal (*single-board circuit; SBC*) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Rasberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris.

Raspberry Pi 3 Model B+ adalah mini komputer versi terbaru yang di keluarkan Raspberry Pi sebagai pengembangan dari versi Raspberry Pi sebelumnya dan Menjadi penerus Raspberry Pi 3 Model B yang sudah release,Secara design fisik Raspberry Pi 3 Model B dengan Raspberry Pi 3 model B+ sama sehingga untuk case versi Raspberry Pi 3 Model B bisa di pakai. Bentuk fisik dari Raspberry Pi 3 B+ dapat dilihat pada Gambar 3.7

UNIVERSITAS Dinamika



Gambar 3.7 Raspberry Pi 3 B+

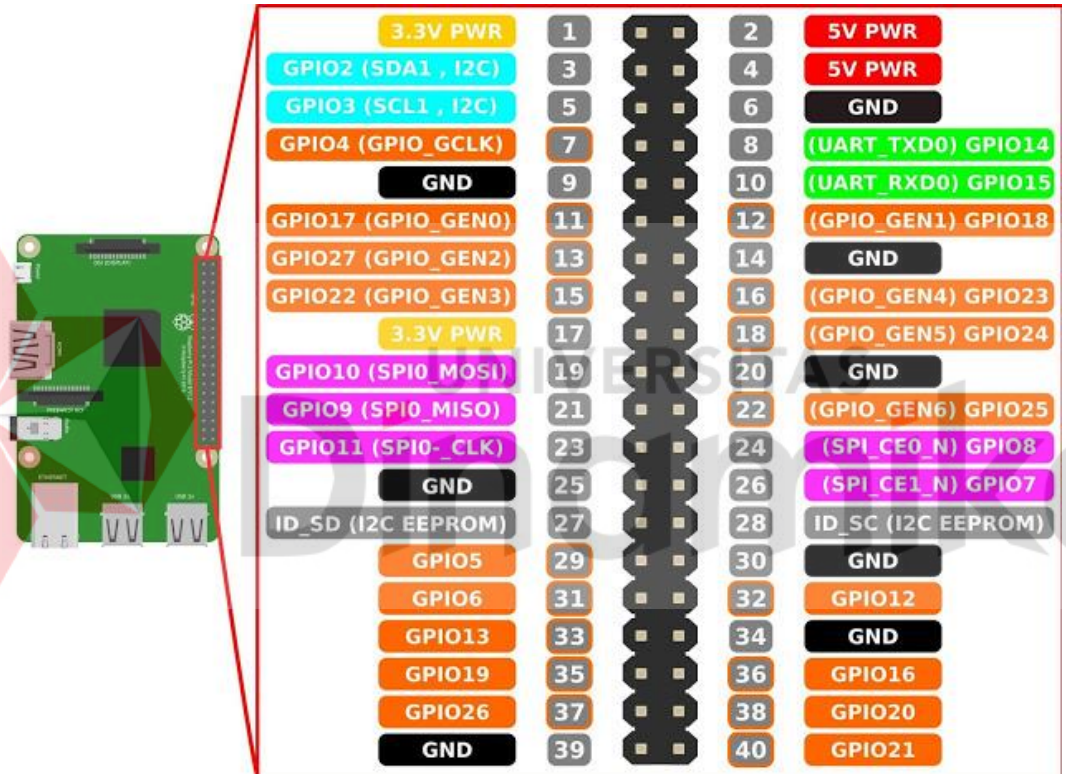
Spesifikasi Raspberry Pi 3 B+ adalah sebagai berikut :

Processor :	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
Memory :	1GB SDRAM
Power Supply :	5V/2.5A DC power input (micro USB)
Wireless :	Bluetooth 4.2 BLE
Ethernet :	Maximum throughput 300 Mbps
GPIOs :	Extended 40 pin GPIO header
Video Output :	Full size HDMI
Audio Output :	Stereo output and composite video port
USB Port :	USB 2.0 Port
CSI :	For Raspberry Pi Camera
DSI :	For touchscreen display
Micro SD :	For loading Operating System

UNIVERSITAS Dinamika

PoE :	Support PoE HAT
Operating Temperature :	0~50 derajat Celcius
Dimension :	120mm x 75mm x 34mm
Weight :	75g

Skematik PIN GPIO Raspberry Pi 3 B+ dapat dilihat pada Gambar 3.8



Gambar 3.8 PIN GPIO Raspberry Pi 3 B+

3.5 Robot SYNA SWC8

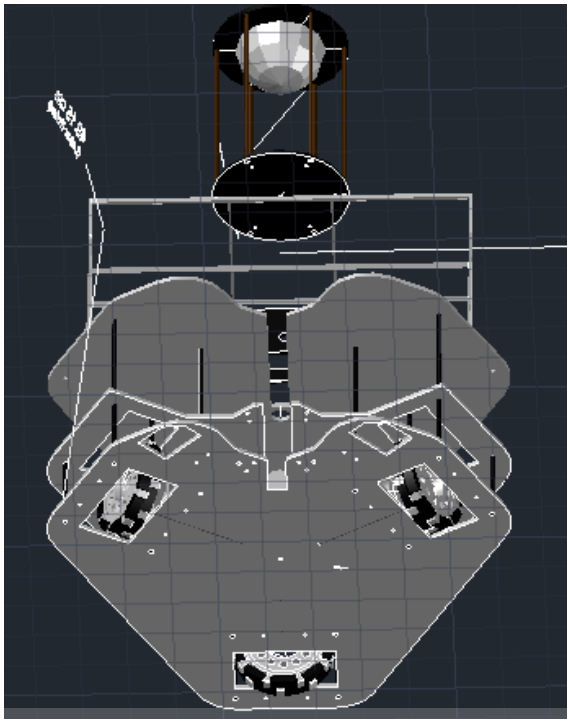
Robot SYNA SWC8 merupakan robot yang diciptakan oleh Tim Robot komunitas CYBER ROBOTIC dengan tujuan untuk perlombaan KRI pada Tahun 2018. Bentuk dari *desain* robot dapat dilihat pada Gambar 3.9 jika diperhatikan

robot ini merupakan robot omni yang memiliki tiga buah roda dimana jarak antara roda sebesar 120°.

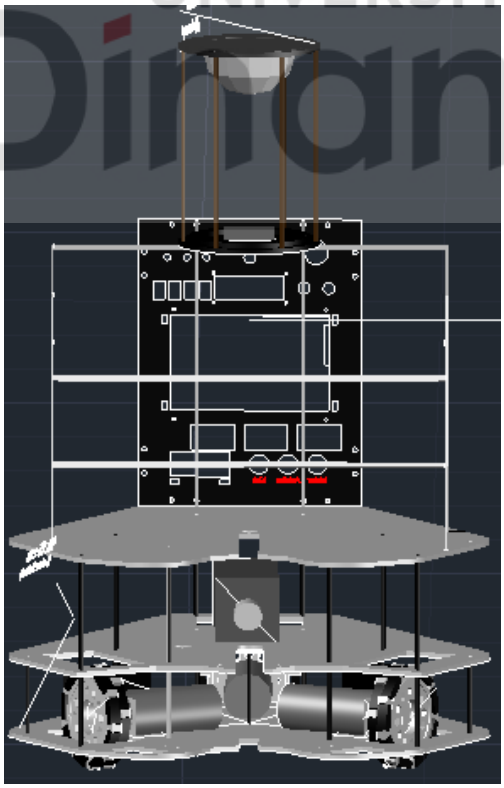
Gambar 3.9 Robot SYNA SWC8

3.5.1 Base Robot

Base robot terlebih dahulu di desain ke dalam AutoCAD dengan tujuan mempermudah proses pembangunan dari kerangka mekanika dari robot. Berikut pada Gambar 3.10 merupakan desain robot tampak bawah. Dan pada Gambar 3.11 merupakan base robot tampak depan.



Gambar 3.10 Base Robot



Gambar 3.11 Base Robot tampak depan

UNIVERSITAS Dinamika

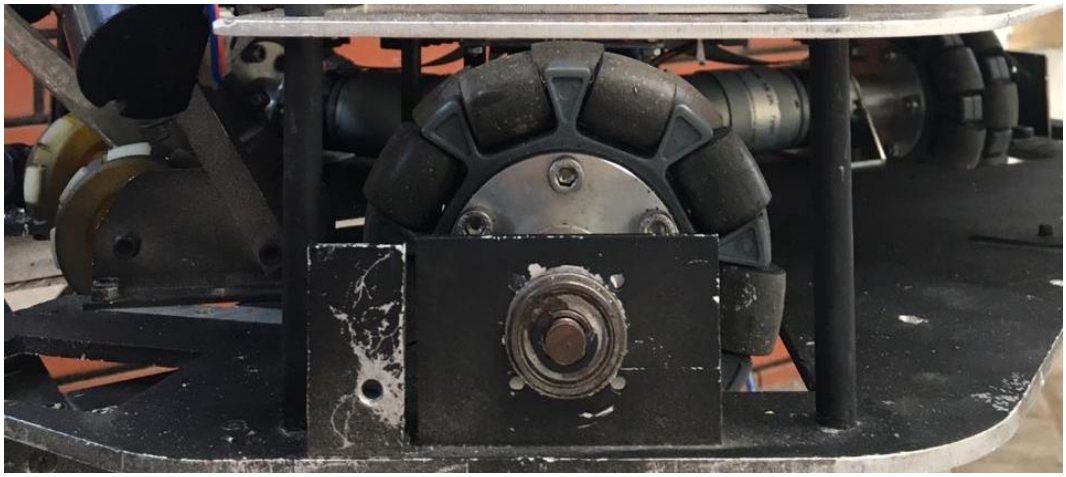
3.5.2 Mekanika Dribbling

Mekanika Dribbling pada robot SYNA SWC8 menggunakan motor DC dengan type PG-36 dikarenakan PD-36 mempunyai kekuatan torsi yang besar dan juga memiliki *gear box* yang mampu menahan dorongan yang melebihi 2 kg mengingat object adalah bola yang diam. Pada Gambar 3.12 merupakan dribbling robot SYNA SWC8.

Gambar 3.12 Dribbling Robot SYNA SWC8

3.5.3 Motor penggerak

Motor penggerak robot SYNA SWC8 juga menggunakan motor DC dengan type PG-36 berjumlah tiga buah. Masing-masing dari motor terpasang juga roda *Omnidirectional*. Berikut pada Gambar 3.13 merupakan roda dan Motor DC.



Gambar 3.13 Motor Penggerak Robot

3.5.4 Letak kamera robot

Kamera Robot SYNA SWC8 terletak dibagian atas robot dengan menghadap kedepan hanya mampu memiliki arah pandang kedepan saja. Dengan menggunakan kamera webcame Logitech C922 Pro. Pada Gambar 3.14 merupakan letak kamera pada Robot SYNA SWC8.

Gambar 3.14 Letak Kamera Robot

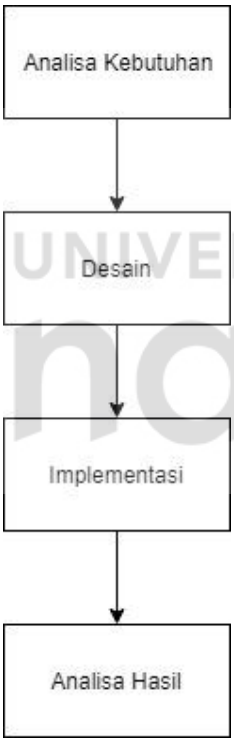
UNIVERSITAS Dinamika

BAB IV

DISKRIPSI PEKERJAAN

4.1 Prosedur Penelitian

Pada pengerjaan *Image Processing* pada robot sepak bola ini dibutuhkan prosedur penelitian untuk mengerjakan dan menentukan tahapan yang akan dilalui untuk mengerjakan *project* serta kebutuhan yang akan dibutuhkan untuk mengerjakan *project* ini. Tahapan dapat dilihat pada Gambar 4.1



Gambar 4.1 Prosedur Penelitian

4.2 Analisa Kebutuhan

Analisa kebutuhan ini dibutuhkan sebagai bagian awal untuk mengidentifikasi masalah serta kebutuhan yang dibutuhkan untuk menyelesaikan

masalah yang ada. Kebutuhan ini bisa berupa metode dan kebutuhan sistem yang akan digunakan, untuk kebutuhan sistem sendiri berupa software dan hardware.

4.2.1 Pengisian *Operating System* Raspberry Pi B 3+

Raspberry Pi B 3+ tidak memiliki memory internal sebagai *storage Operating System*. Namun, disediakan sebuah *slot* memory SD Card untuk menjadi *Storage*. Perlu ada tahapan untuk melakukan pengisian ke SD Card. Yaitu sebagai berikut :

A. Menggunakan Aplikasi Win32 Imager

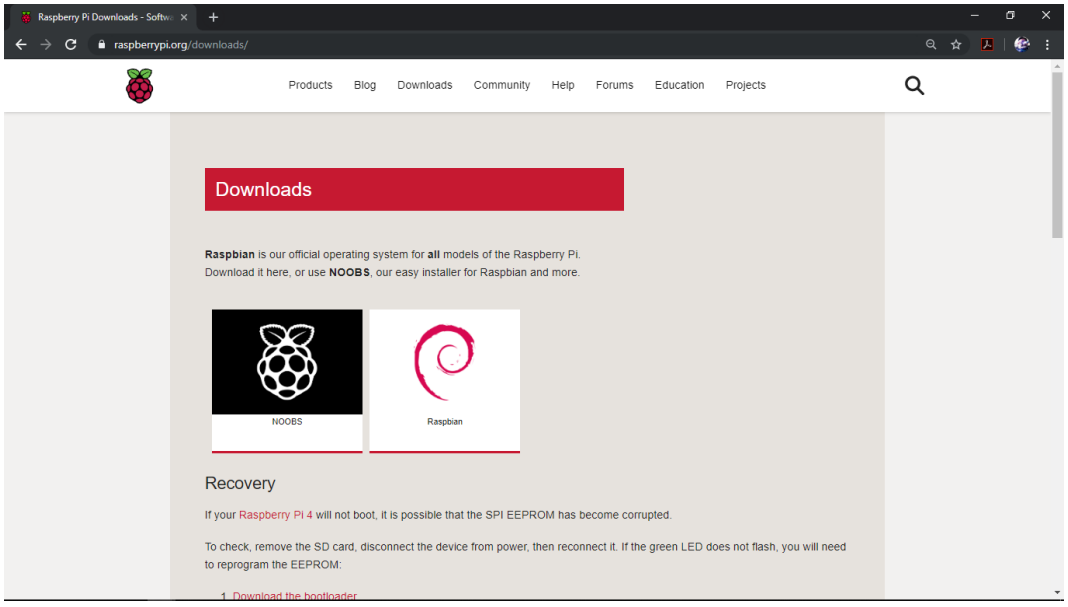
Aplikasi ini bertujuan untuk melakukan *booting Operating System* Raspberry Pi 3 B+ yang masih dalam berbentuk .zip dapat tersalin ke SD Card yang akan digunakan nantinya. Pada Gambar 4.2 merupakan logo Win32 Disk Imager.

Gambar 4.2 Logo Win32 Disk Imager

B. Download Operating System Raspberry Pi 3 B+

Sebelum melakukan pengisian perlu men-*download* terlebih dahulu OS Raspberry Pi yang diinginkan dapat men-*download* pada *link* : https://www.raspberrypi.org/downloads/ . Pada Gambar 4.3 merupakan tampilan website.

UNIVERSITAS Dinamika



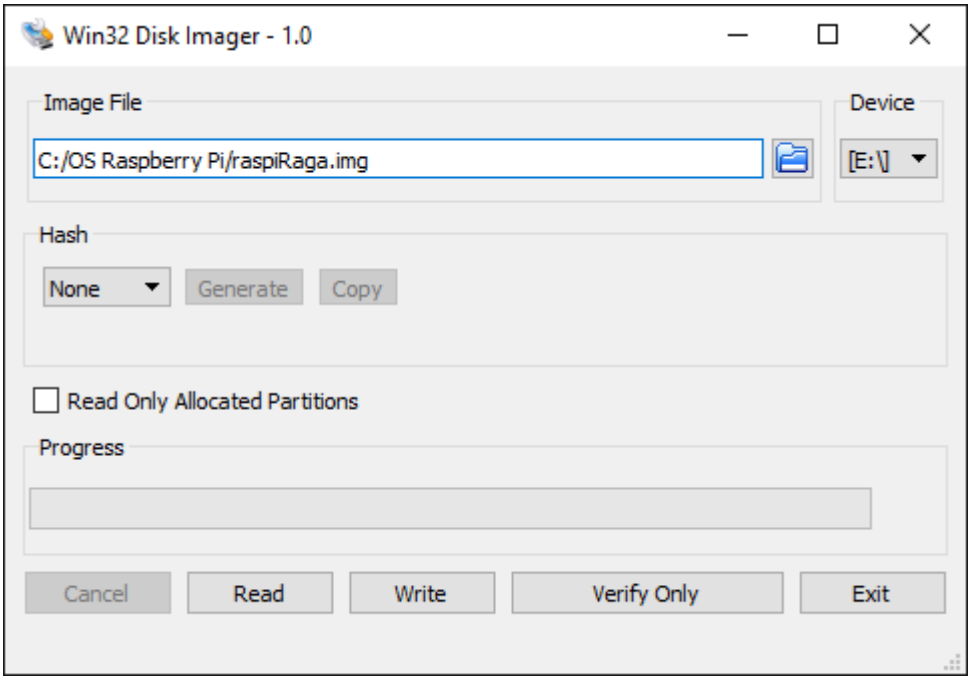
Gambar 4.3 Website Raspberry Pi 3 B+

C. Pengisian OS ke SD Card

Siapkan Adapter SD Card dan juga memory SD Card yang akan digunakan. Penulis menyarankan menggunakan kapasitas memori minimal 16GB dikarenakan untuk mengoptimalkan pemakaian.

Gambar 4.4 SD Card Adapter dan SD Card 16 GB

Membuka Win32 Imager ditujukan untuk melakukan proses *booting* kedalam SD Card yang masih kosong.



Gambar 4.5 Tampilan Win32 Disk Imager

Pastikan memiliki direktori berformat .img supaya dapat dikenali oleh aplikasi Win32 Disk Imager. Sebagai contoh penulis menggunakan raspRaga.img seperti pada Gambar 4.5 jika sudah lakukan proses Write dengan tujuan proses penulisan raspiRaga.img ke SD Card yang terletak di [E:\].

4.2.2 Konfigurasi OpenCV

Sebelum melakukan proses *compile* OpenCV, terdapat beberapa software yang dibutuhkan terlebih dahulu. Berikut pada Gambar 4.6 merupakan perintah untuk tahap awal :

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install build-essential cmake unzip pkg-config -y
sudo apt-get install libjpeg-dev libpng-dev libtiff-dev -y
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
-y
sudo apt-get install libxvidcore-dev libx264-dev -y
sudo apt-get install libgtk-3-dev -y
sudo apt-get install libcanberra-gtk* -y
sudo apt-get install libatlas-base-dev gfortran -y
sudo apt-get install python3-dev -y
```

Gambar 4.6 Perintah persiapan OpenCV

Kemudian, download file OpenCV melalui terminal. Terdapat pada Gambar 4.7

```
cd ~
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.0.zip
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.
1.0.zip
```

Gambar 4.7 Download OpenCV

File yang telah terdownload dan tersimpan pada Directory Raspberry Pi 3 B+ perlu adanya ekstrak isi file yang terdapat pada Gambar 4.8

```
unzip opencv.zip
unzip opencv_contrib.zip
```

Gambar 4.8 Proses Unzip OpenCV

Untuk mempermudah proses pencarian data, dapat menggunakan akses jaringan yang baik. Untuk me-*rename* pada Gambar 4.9 raspberry Pi 3 B+ adalah sebagai berikut:

```
mv opencv-4.1.0 opencv
mv opencv_contrib-4.1.0 opencv_contrib
```

Gambar 4.9 Pencarian File OpenCV

Memodifikasi hasil program sebelumnya, dan pada selanjutnya proses penyisisipan virtual :

```
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

Gambar 4.10 Penyisipan OpenCV

UNIVERSITAS Dinamika

Setelah semua perintah diatas terpenuhi proses waktunya *compile* OpenCV. Perlu diperhatikan proses *Compile* ini akan menggunakan semua core prosesor Raspberry Pi 3 B+, pastikan terdapat Kipas maupun heatsink pada Mini PC. Proses *compile* terdapat pada Gambar 4.11

```
make -j4
```

Gambar 4.11 Compile OpenCV

Setelah proses selesai ketikan perintah berikut untuk proses penyelesaian akhir. Dapat dilihat pada Gambar 4.12

```
sudo make install
sudo ldconfig
```

Gambar 4.12 Proses akhir penginstalan OpenCV

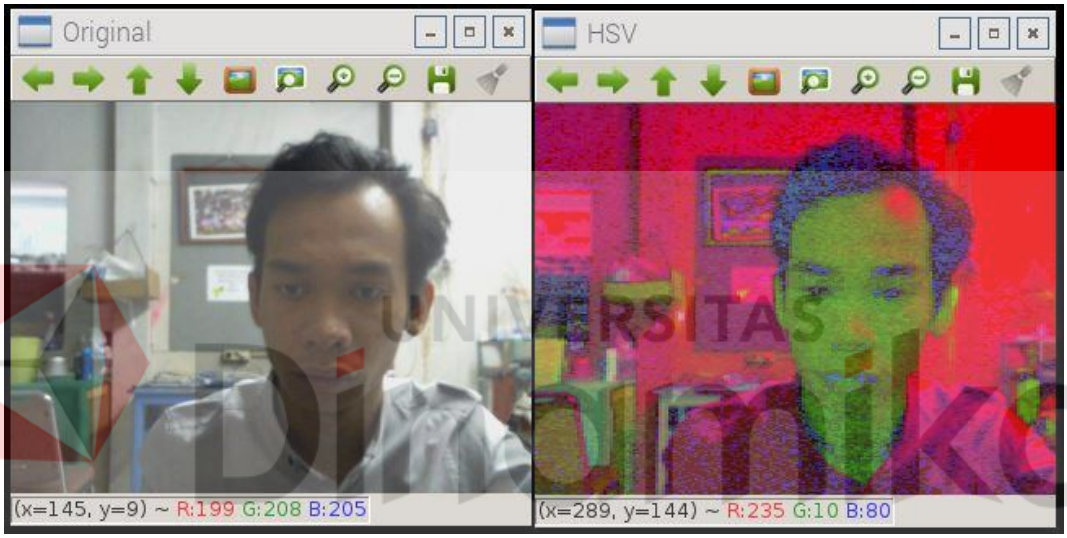
UNIVERSITAS Dinamika

4.3 Desain Flowchart Pengolahan Citra

Pada Flowchart Pengolahan Citra diharapkan dapat memberikan gambaran tentang alur proses pengolahan data gambar nantinya. Desain pengolahan Citra merupakan sebuah Blok Vision. Pada project ini mempunyai Flowchart program seperti yang ditunjukkan pada Gambar 4.17 . Penjelasan dari Flowchart pada Gambar 4.17 adalah sebagai berikut :

1. Inisialisasi merupakan proses deklarasi variable-variable yang digunakan di dalam program. Pada penelitian ini variable-variable seperti variable yang digunakan untuk menampung gambar tangkapan kamera.

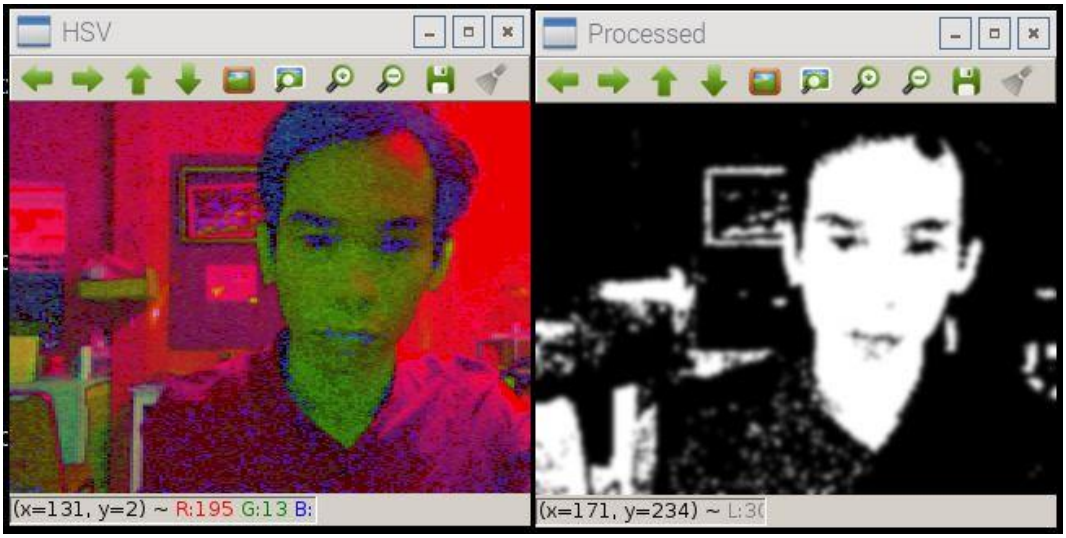
2. Menampilkan status pada Frame. Ketika Frame bernilai NULL maka program akan berhenti. Sedangkan, ketika Frame bernilai tidak NULL maka program akan menjalankan perintah berikutnya.
3. Konversi RGB ke HSV merupakan pengkonversian model warna pada Frame. Proses ini bertujuan untuk mengubah model warna Frame yang semula mempunyai model warna RGB ke HSV. Berikut adalah contoh gambar hasil proses RGB ke HSV terdapat pada Gambar 4.13



(a) (b)

Gambar 4.13 Konversi Ruang RGB (a) ke Ruang HSV (b)

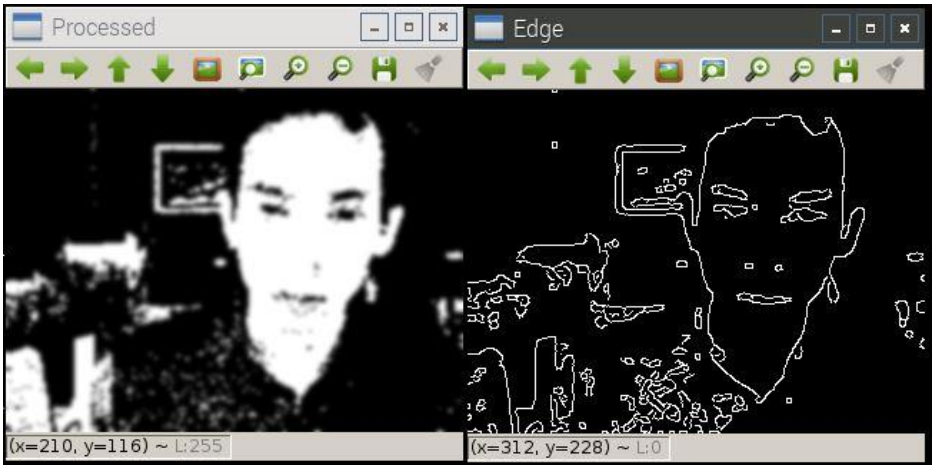
4. *Color Filtering* merupakan proses penyeleksian warna objek. Proses ini bertujuan untuk mencari nilai filter minimal dan maksimal dari model warna HSV pada Frame. Sehingga dapat diketahui bagian mana yang termasuk objek dan mana yang bukan objek.



(a) (b)

Gambar 4.14 Penentuan Objek HSV (a) ke target objek (b)

5. *Edge Detection* merupakan proses mengubah gambar pada Frame menjadi garis tepi. Proses ini bertujuan untuk mengubah hasil *Color Filtering* yang masih berupa blob menjadi garis tepi. Sehingga setelah menjadi garis akan memudahkan proses selanjutnya.



(a) (b)

Gambar 4.15 Filtering (a), pendeteksian garis tepi (b)

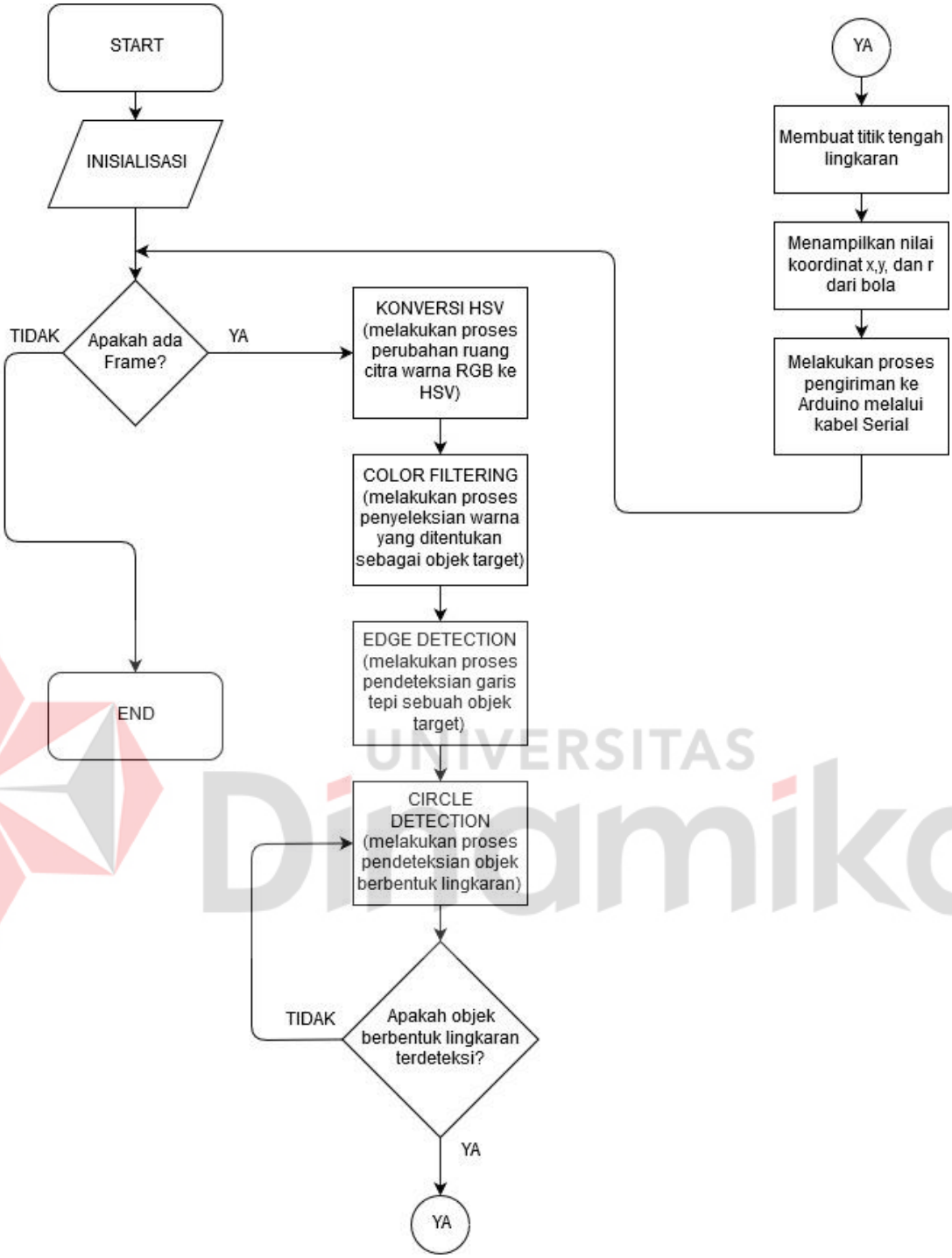
6. *Circle Detection* untuk mendeteksi garis yang berbentuk lingkaran pada gambar yang sudah diolah sebelumnya menggunakan metode *Edge Detection*.

(a) (b)

Gambar 4.16 Deteksi garis tepi (a), deteksi lingkaran (b)

7. Ketika objek terdeteksi, maka nilai dari lingkaran menjadi 1.
8. Ketika nilai garis sama dengan 1, maka program akan melanjutkan ke perintah berikutnya.
9. Proses selanjutnya yaitu membuat titik tengah dari bola yang terdeteksi.
10. Setelah titik tengah dibuat, dapat diketahui koordinat dari titik tersebut. Koordinat yang keluar berupa nilai koordinat x, y, dan r.

UNIVERSITAS Dinamika



Gambar 4.17 Flowchart Pengolahan Citra

4.4 Implementasi

Dalam tahap ini rancangan yang akan dibuat akan diterapkan pada Robot SYNA SWC8 sebagai pendeteksian objek bola. Pada tahapan ini dilakukan beberapa tahapan yaitu :

1. Menggunakan Mini PC Raspberry Pi 3 B+ yang sudah terdapat di Lab Robotika Stikom Surabaya.
2. Mengkonfigurasi Mini PC Raspberry Pi 3 B+ dengan OpenCV.
3. Implementasi Mini PC Raspberry Pi 3 B+ untuk melakukan pendeteksian bola.

4.5 Analisa Hasil

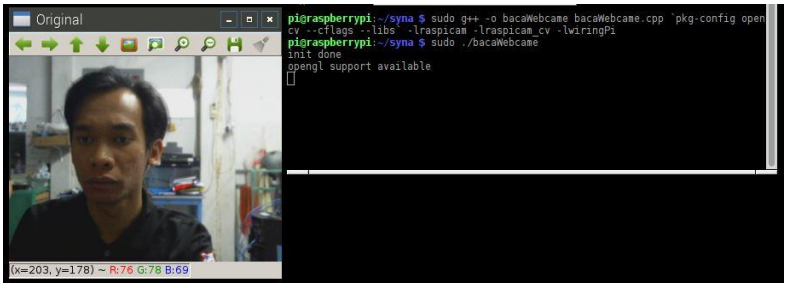
Pada tahap yang terakhir ini nantinya akan dilakukan analisa hasil, dimana pada percobaan di Tahap Software OpenCV akan di implementasikan. Setelah itu hasil akan dianalisa dan disimpulkan. Nantinya hasil analisa berupa nilai yang telah ditentukan sebagai nilai kalibrasi awal. Dan dari analisa tersebut akan dibuktikan dengan kondisi lingkungan sekitar untuk proses pendeteksian bola.

4.6 Hasil dan Pembahasan

4.6.1 Pembacaan Webcame dengan OpenCV

Percobaan awal menggunakan OpenCV dalam membaca kamera webcame eksternal Logitech C922 Pro. Terdapat pada Gambar 4.18 merupakan hasil dari pembacaan webcame menggunakan OpenCV dan pada bagian bawah ini merupakan program untuk pembacaan kamera webcame.

UNIVERSITAS Dinamika



Gambar 4.18 Percobaan Baca Webcame

Berikut adalah *source code* dan langkah-langkah untuk melakukan proses pembacaan Kamera *Webcame* menggunakan *library* OpenCV :

Langkah yang pertama harus dilakukan adalah memasukkan *header* yang dibutuhkan untuk proses pembacaan kamera *Webcame*. *Header* bertujuan untuk memanggil pustaka yang terdapat pada *library* opencv yang dapat digunakan untuk proses pengolahan citra yang akan dilakukan nanti. Tidak hanya pustaka opencv saja melainkan pustaka yang mendukung proses dari sistem nantinya. Berikut kumpulan nama *header* yang akan digunakan untuk proses pembacaan kamera *Webcame* :

```
#include <iostream>
#include “cv.h”
#include “opencv2/highgui/highgui.hpp”
```

Langkah yang kedua perlu adanya proses pembacaan alamat dari usb kamera *Webcame*. Ditujukan saat program dijalankan alamat yang telah ditentukan dapat memanggil kamera *Webcame* untuk difungsikan/dihidupkan. Berikut adalah penulisan program untuk menentukan alamat dari usb kamera *Webcame* :

```
CvCapture* img = cvCaptureFromCAM(0);
```

UNIVERSITAS Dinamika

CvCapture* merupakan sebuah type data yang mempunyai pointer yang disediakan OpenCV untuk melakukan tugas khusus dalam melakukan pengambilan gambar yang bergerak secara *real-time* atau dapat dikatakan lain adalah video. **img** merupakan variable yang *user* buat untuk menampung proses pengambilan citra nantinya. **cvCaptureFromCAM(0)** merupakan fungsi yang dibuat oleh openCV untuk mengarahkan bahwa citra yang akan diolah berupa data video yang bersumber dari CAM dan angka 0 merupakan letak alamat kamera webcame berada untuk melakukan proses pengambilan citra.

Langkah yang ketiga perlu adanya pendeklarasian variabel yang digunakan untuk menampung citra yang akan ditampilkan. Berikut adalah penulisan program untuk melakukan pendeklarasian :

```
IplImage* p_imgOriginal;
```

IpImage* merupakan type data yang memiliki sebuah pointer yang telah disediakan openCV untuk menangani data matriks, dikarenakan data yang diolah nantinya merupakan data matrik 2D. **p_imgOriginal** merupakan variabel yang dibuat oleh user untuk diolah nantinya dan sudah menjadi data variable yang berbentuk matriks.

Langkah yang keempat perlu adanya pembuat *frame* GUI yang bertujuan untuk menampilan citra yang ditangkap oleh kamera *Webcame*. Berikut adalah penulisan program pembuatan *frame* GUI pada OpenCV :

```
cvSetCaptureProperty(img, CV_CAP_PROP_FRAME_WIDTH, 320);
cvSetCaptureProperty(img, CV_CAP_PROP_FRAME_HEIGHT, 240);

cvNamedWindow("Original", CV_WINDOW_AUTOSIZE);
```

UNIVERSITAS Dinamika

cvSetCaptureProperty(img, CV_CAP_PROP_FRAME_WIDTH, 320) merupakan fungsi OpenCV untuk melakukan pembuatan properti pada citra dan di dalamnya terdapat **img** yang merupakan variabel yang akan diproses dalam pengolahan citra. **CV_CAP_PROP_FRAME_WIDTH, 320** ditujukan untuk menjadikan citra nantinya memiliki lebar sebanyak 320 pixel atau dapat dikatakan lain terdapat 320 kotak pixel yang akan diolah. **CV_CAP_PROP_FRAME_HEIGHT, 320** ditujukan untuk menjadikan citra nantinya memiliki tinggi sebanyak 240 pixel atau dapat dikatakan lain terdapat 240 kotak pixel yang akan diolah nantinya. **cvNamedWindow("Original", CV_WINDOW_AUTOSIZE)** merupakan proses pemberian nama pada Frame GUI yang telah dibuat dengan nama Original dan **CV_WINDOW_AUTOSIZE** ditujukan saat frame terbuat dengan ukuran yang otomatis atau menyesuaikan.

Setelah penjelasan diatas berikut adalah penulisan program secara keseluruhan untuk proses pembacaan menggunakan kamera *Webcame* :

UNIVERSITAS Dinamika

```
#include <iostream>
#include "cv.h"
#include "opencv2/highgui/highgui.hpp"
using namespace cv;
using namespace std;
int main(int argc, char* argv[]){
        IplImage* p_imgOriginal;
        CvCapture* img = cvCaptureFromCAM(0);
        cvSetCaptureProperty(img, CV_CAP_PROP_FRAME_WIDTH, 320);
        cvSetCaptureProperty(img, CV_CAP_PROP_FRAME_HEIGHT, 240);
        cvSetCaptureProperty(img, CV_CAP_PROP_FPS, 100);
        cvNamedWindow("Original", CV_WINDOW_AUTOSIZE);
        while (1) {
                p_imgOriginal = cvQueryFrame(img);
                p_strStorage = cvCreateMemStorage(0);
                cvShowImage("Original", p_imgOriginal);
                cvReleaseMemStorage(&p_strStorage);
                cvWaitKey(1);
        }
        serialClose(fd);
        cvDestroyAllWindows();
        cvReleaseCapture(&img);
        cvReleaseImage(&p_imgOriginal);
        return(0);
}
```

4.6.2 Pembahasan Pengolahan Citra Robot

Penulisan program untuk *Image Processing* robot untuk melakukan pendeteksian bola sangatlah berbeda. Namun, untuk melakukan proses awal yaitu membuka kamera *Webcame* sangatlah sama dengan penjelasan program sebelumya. Program ini memiliki beberapa tahapan proses yang akan dilakukan yaitu : pengubahan citra RGB ke citra HSV, dari HSV mengalami perubahan proses *Filtering* atau penentuan warna yang akan menjadi target objek. Setelah itu citra mengalami proses pendeteksian tepi dimana objek yang telah ditentukan tadi diambil garis tepi terluar. Dan yang proses terakhir penandaan bahwa objek tersebut dipastikan memiliki bentuk lingkaran. Dikarenakan juga objek berupa bola berbentuk bulat dan memiliki warna orange.

Langkah pertama perlu memasukkan terlebih dahulu *header* yang akan dibutuhkan. *Header* dapat dituliskan sebagai berikut :

```
#include <iostream>
#include "cv.h"
#include <stdio.h>
#include "opencv2/highgui/highgui.hpp"
#include <wiringPi.h>
#include <wiringSerial.h>
```

<iostream> merupakan singkatan dari input output stream header yang digunakan sebagai standart input output operasi yang digunakan oleh bahasa C++. **cv.h** merupakan header yang disediakan OpenCV untuk melakukan penulisan dalam fungsi-fungsi yang terdapat pada OpenCV, sehingga user mengetahui fungsi-fungsi yang tersedia dan dapat digunakan. **"opencv2/highgui/highgui.hpp"** merupakan pemanggilan pada folder yang telah dituju dimana terdapat fungsi-fungsi yang dibutuhkan. **<wiringPi.h>** ditujukan untuk megaktifkan GPIO dari mini PC Raspberry Pi 3 B+. **<wiringSerial.h>** merupakan header untuk melakukan kominikasi serial dalam mode ON.

Langkah Kedua perlu proses pembuatan *Trackbar* tujuan dibuatnya kumpulan *Trackbars* ditujukan untuk proses filtering atau penyesuaian warna objek yang akan ditentukan. Dimana objek yang akan dipilih adalah objek bola yang berwarna orange dideklarasikan terlebih dahulu variabel-variabel yang digunakan. Jika dilihat pada program variabel yang digunakan menggunakan jenis data **int (integer)** yang dinamakan H_MIN, H_MAX, V_MIN, V_MAX, S_MIN, S_MAX disesuaikan dengan nilai-nilai warna yang akan dicari yaitu terdapat *range* 0-255. Berikut pada Gambar adalah hasil dari proses pembuatan *Trackbars* pada OpenCV. Berikut merupakan penulisan program untuk menciptakan *Trackbar* :

UNIVERSITAS Dinamika

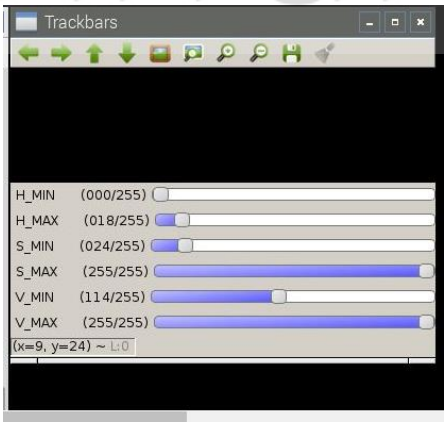
```
int H_MIN = 0;
int H_MAX = 43;
int S_MIN = 122;
int S_MAX = 255;
int V_MIN = 134;
int V_MAX = 255;

void on_trackbar(int, void*)
{}

void createTrackbars()
{
cvNamedWindow("Trackbars", 0);
char TrackbarName[50];
printf(TrackbarName, "H_MIN", H_MIN);
printf(TrackbarName, "H_MAX", H_MAX);
printf(TrackbarName, "S_MIN", S_MIN);
printf(TrackbarName, "S_MAX", S_MAX);
printf(TrackbarName, "V_MIN", V_MIN);
printf(TrackbarName, "V_MAX", V_MAX);

createTrackbar("H_MIN", "Trackbars", &H_MIN, H_MAX, on_trackbar);
createTrackbar("H_MAX", "Trackbars", &H_MAX, H_MAX, on_trackbar);
createTrackbar("S_MIN", "Trackbars", &S_MIN, S_MAX, on_trackbar);
createTrackbar("S_MAX", "Trackbars", &S_MAX, S_MAX, on_trackbar);
createTrackbar("V_MIN", "Trackbars", &V_MIN, V_MAX, on_trackbar);
createTrackbar("V_MAX", "Trackbars", &V_MAX, V_MAX, on_trackbar);
}
```

Pada program diatas saat dijalankan akan menghasilkan output sebagai berikut yang terdapat pada Gambar 4.19



Gambar 4.19 Trackbar OpenCV

Langkah ketiga untuk mendeteksi diperlukan beberapa fungsi yang akan digunakan untuk proses perubahan citra RGB ke citra HSV dan juga nantinya akan

ditambahkan proses *Edge* untuk mengetahui garis tepi pada sebuah objek yang di dapat dan yang sudah ditentukan. Dapat dijabarkan fungsi dari OpenCV **`cvQueryFrame`** adalah fungsi OpenCV dengan tujuan untuk menggunakan lokasi memori yang sama untuk melakukan proses pengambilan gambar secara berurutan. **`p_imgOriginal`** adalah variabel yang menampung data berurutan dari webcame data berupa citra ruang warna RGB. **`cvCvtColor (InputArray src, OutputArray dst, int code, int dstCn=0)`** merupakan fungsi untuk melakukan konversi citra ruang warna RGB ke HSV data akan akhir berupa citra yang memiliki ruang warna HSV yang tersimpan pada variabel **`p_imgHSV`**. **`cvInRange(Frame_HSV, Scalar(low_H, low_S, low_V), Scalar(high_H, high_S, high_V), frame_threshold)`** bertujuan untuk membantu pengaturan nilai trackbar yang akan dibuat nantinya mampu merubah nilai dari filtering nantinya. **`cvSmooth(const CvArr* src, CvArr* dst, int smoothtype=CV_GAUSSIAN, int size1=3, int size2=0, double sigma1=0, double sigma2=0)`** pada OpenCV bertujuan untuk membuat citra menjadi blur dimana terdapat **`CV_GAUSSIAN(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY=0, int borderType=BORDER_DEFAULT)`** dan bernilai 9,9 ditujukan untuk mengurangi kedalaman sebuah informasi dari sebuah citra untuk meringankan memori juga. Sehingga terjadi proses pengolahan citra yang lebih ringan. **`cvCanny(detected_edge, detected_edge, lowThreshold, lowThreshold*ratio, kernel_size)`** mempunyai fungsi untuk menyeleksi nilai citra agar yang terambil hanya citra yang berada pada tepi saja. **`cvHoughCircle`** digunakan untuk mendeteksi objek yang berbentuk lingkaran sehingga proses penyeleksian ditambahkan fitur pendeteksian objek yang

berbentuk orange dalam penulisan program **cvHoughCircles(InputArray image, OutputArray circles, int method, double dp, double minDist, double param1=100, double param2=100, int minRadius = 0, int maxRadius=0).**

```
p_imgOriginal = cvQueryFrame(img);
cvCvtColor(p_imgOriginal, p_imgHSV, CV_BGR2HSV);
cvInRangeS(p_imgHSV, cvScalar(H_MIN, S_MIN, V_MIN), cvScalar(H_MAX, S_MAX,
V_MAX), p_imgProcessed);

cvSmooth(p_imgProcessed, p_imgProcessed, CV_GAUSSIAN,9 ,9);
cvCanny(p_imgProcessed, p_imgEdge, 100, 150, 3);
p_seqCircles = cvHoughCircles(p_imgProcessed, p_strStorage,
CV_HOUGH_GRADIENT, 2, p_imgProcessed->height / 4, 100, 30, 0, 50);
```

Tabel 4.1 Penjelasan Fungsi Citra OpenCV

Penulisan Program	Penjelasan
CV_RGB2HSV	Konversi gambar RGB atau BGR ke representasi warna HSV (hue saturation value) atau sebaliknya.
CV_BGR2HSV	
CV_GAUSSIAN	Jumlahkan lebih dari a param1 × param2 lingkungan.
CV_BLUR	Jumlahkan lebih dari param1 × param2neokal dengan yang berikutnya scaling dengan 1 / (param1 × param2).
CV_MEDIAN	Temukan median di atas a param1 × param1 kuadrat lingkungan.
cvCvtColor	Ubah kumpulan array dari satu ruang warna ke ruang warna lainnya.

UNIVERSITAS Dinamika

cvInRanges	Uji apakah elemen array berada dalam nilai dua array lainnya.
cvQueryFrame	Merupakan fungsi OpenCV yang menggunakan lokasi memori yang sama untuk pengambilan gambar seacara berurutan atau *real time*.
cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0)	Parameter : • **src** merupakan data input citra yang memiliki nilai 8-bit dengan unsigned, dan juga dapat menggunakan 16-bit unsigned (**CV_16UC...**) atau juga dapat menggunakan satuan nilai dalam bentuk float. • **dst** merupakan data output citra yang menghasilkan ukuran yang sama dengan data input citra dan nilai kedalaman citra juga sama dengan data input **src**. • **code** merupakan konversi jarak. Sebagai contohnya adalah konversi ruang warna RGB ke ruang warna GRAY

UNIVERSITAS Dinamika

	dimana membutuhkan sebuah konversi citra **(CV_BGR2GRAY, CV_RGB2GRAY, CV_GRAY2BGR).** Pada OpenCV sudah tersedia fungsi untuk melakukan konversi sebuah citra. • **dstCn** merupakan penomoran sebuah channel sebuah citra. Jika parameter bernomor 0 maka, berada pada channel 0. Yang juga otomatis berasal dari **src** dan **code**.
cvInRangeS(src, lower, upper, dst)	Parameter : • **src** merupakan data input citra yang akan diproses nantinya. • **lower** merupakan nilai terendah dalam array sebuah nilai yang akan menjadi *range* terendah sampai *range* yang tertinggi.

UNIVERSITAS Dinamika

	• **upper** merupakan nilai tertinggi dalam array sebuah nilai yang akan menjadi *range* terendah sampai *range* yang tertingi. • **dst** merupakan data output citra yang menghasilkan ukuran yang sama dengan data input citra dan nilai kedalaman citra juga sama dengan data input **src**.
cvSmooth(src, dst, smoothtype=CV_GAUSSIAN, int size1, int size2, double sigma1, double sigma2)	Parameter : • **src** merupakan data input citra yang memiliki nilai 8-bit dengan unsigned, dan juga dapat menggunakan 16-bit unsigned (**CV_16UC...**) atau juga dapat menggunakan satuan nilai dalam bentuk float. • **dst** merupakan data output citra yang menghasilkan ukuran yang sama dengan data input citra dan nilai kedalaman citra

UNIVERSITAS Dinamika

	juga sama dengan data input **`src`**. • **`smoothtype`** terdapat banyak macam yaitu : **`CV_BLUR_NO_SCALE`** merupakan konvolusi linear dengan **`size1 x size2`** dalam sebuah kernel. Jika ingin menghaluskan piksel yang berbeda dengan kernel kotak yang memiliki ukuran yang berbeda dapat menggunakan citra integral yang menggunakan fungsi **`integral()`**. • **`size1`** adalah parameter pertama untuk melakukan proses penghalusan sebuah citra. Yang menggunakan nilai positif. • **`size2`** merupakan parameter kedua untuk melakukan proses penghalusan dan juga ketinggian. Cara ini diabaikan

UNIVERSITAS Dinamika

	oleh metode **CV_MEDIAN** dan **CV_BILATERAL**. Dalam kasus blur skala / non skala dan Gaussian sederhana jika size2 adalah nol maka diatur ke size1. Kalau tidak, itu merupakan angka ganjil possitif. • **sigma1** dalam sebuah kondisi parameter ini dapat menentukan gaussian atau sigma standart deviasi.
cvCanny(image, edges, threshold1, threshold2, apertureSize, L2gradient)	Parameter : • **image** merupakan citra input yang memiliki 8-bit input image. • **edges** merupakan data output map yang merupakan single channels 8-bit image, dimana besarnya ukuran sama dengan data citra input yaitu **image**. • **threshold1** merupakan nilai threshold ambang pertama untuk prosedur histeresis.

UNIVERSITAS Dinamika

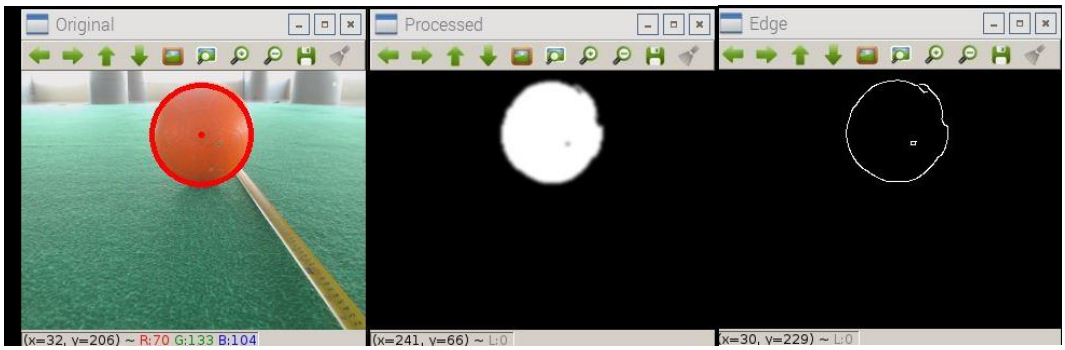
	• **threshold2** merupakan nilai threshold ambang kedua untuk prosedur histeresis. • **aperture Size** merupakan ukuran dalam pengoperasian sobel. • **L2gradient** menunjukan pernilaian keakuratan.
cvHoughCircles(src_gray, circles, CV_HOUGH_GRADIENT, dp, min_dist, param_1, param2, min_radius, max_radius)	Parameter : • **src_gray** merupakan data input citra yang mempunyai ruang warna keabuan. • **circle** untuk membuat sebuah lingkaran jika objek yang dideteksi merupakan objek lingkaran dilihat dari nilai x,y dan r. • **CV_HOUGH_GRADIENT** metode untuk menentukan deteksi. Saat ini yang tersedia hanya fungsi ini yang terdapat pada OpenCV.

UNIVERSITAS Dinamika

	• **dp** merupakan kebalikan dari ratio dari resolusi. • **min_dist** merupakan hasil bagi src_gray.row/8 • **param_1** merupakan nilai tertinggi dari nilai threshold yang telah diambil sebelumnya. • **param_2** merupakan data nilai tengah dari threshold yang telah didapat sebelumnya. • Min_radius merupakan nilai minimum yang akan dideteksi nantinya. • **max_radius** merupakan nilai maksimum yang akan dideteksi nantinya.

Semua fungsi dan informasi untuk melakukan proses pendeteksian bola berwarna orange. Pada Gambar 4.20 adalah hasil dari pendeteksian robot untuk memilih objek yang telah ditentukan sebelunya.

UNIVERSITAS Dinamika



Gambar 4.20 Tampilan Deteksi bola

4.6.3 Nilai HSV

Dikarenakan objek yang akan diteksi merupakan bentuk bola yang memiliki warna orange maka perlu adanya penyesuaian nilai HSV atas dan bawah. Pada Tabel 4.2 merupakan Nilai HSV untuk bola berdiameter 21cm dan memiliki warna Orange. Pada tabel 4.3 merupakan nilai pergeseran pixel saat bola digeserkan setiap 2cm dan posisi kamera diam tetap pada posisinya.

Tabel 4.2 Nilai HSV bola dan Koordinat Bola

NILAI HSV							**Koordinat**	
H_MIN	**H_MAX**	**S_MIN**	**S_MAX**	**V_MIN**	**V_MAX**	**KET**	**X**	**Y**
0	12	24	255	114	255	Bola terdeteksi	167	59
0	12	24	255	114	255	Bola terdeteksi	167	39

UNIVERSITAS Dinamika

0	12	24	255	114	255	Bola terdeteksi	163	39
0	12	24	255	114	255	Bola terdeteksi	159	35
0	12	24	255	114	255	Bola terdeteksi	157	31
0	12	24	255	114	255	Bola terdeteksi	157	29

UNIVERSITAS Dinamika

4.6.4 Nilai Pegeseran *Pixel*

Pada Tabel 4.3 merupakan hasil dari pergeseran nilai pixel pada bola terhadap kamera. Bola diletakkan di depan kamera dan bola digeser dengan selisih 2cm tiap pergeseran, pergeseran dilakukan dengan menggeser ke arah kanan kamera.

Tabel 4.3 Nilai pergeseran Nilai pixel

Data Pergeseran Nilai Pixel Bola				
X (px)	**Y (px)**	**Radius (px)**	**Pergeseran (cm)**	**Ket**
145	129	16	2cm	Bola Terdeteksi
147	127	16	4cm	Bola Terdeteksi

155	127	16	6cm	Bola Terdeteksi
161	127	16	8cm	Bola Terdeteksi
167	127	15	10cm	Bola Terdeteksi
172	127	15	12cm	Bola Terdeteksi
181	127	15	14cm	Bola Terdeteksi
189	129	16	16cm	Bola Terdeteksi
193	127	15	18cm	Bola Terdeteksi
199	129	16	20cm	Bola Terdeteksi
285	129	16	22cm	Bola Terdeteksi
213	127	15	24cm	Bola Terdeteksi
219	127	15	26cm	Bola Terdeteksi
225	127	15	28cm	Bola Terdeteksi
233	129	16	30cm	Bola Terdeteksi

UNIVERSITAS Dinamika

4.6.5 Nilai pergseseran *pixel* menjauhi kamera

Pada Tabel 4.4 merupakan hasil dari pergeseran nilai pixel yang menjauhi kamera. Bola diletakkan tepat di depan kamera dan jarak pergeseran memiliki selisih 60cm.

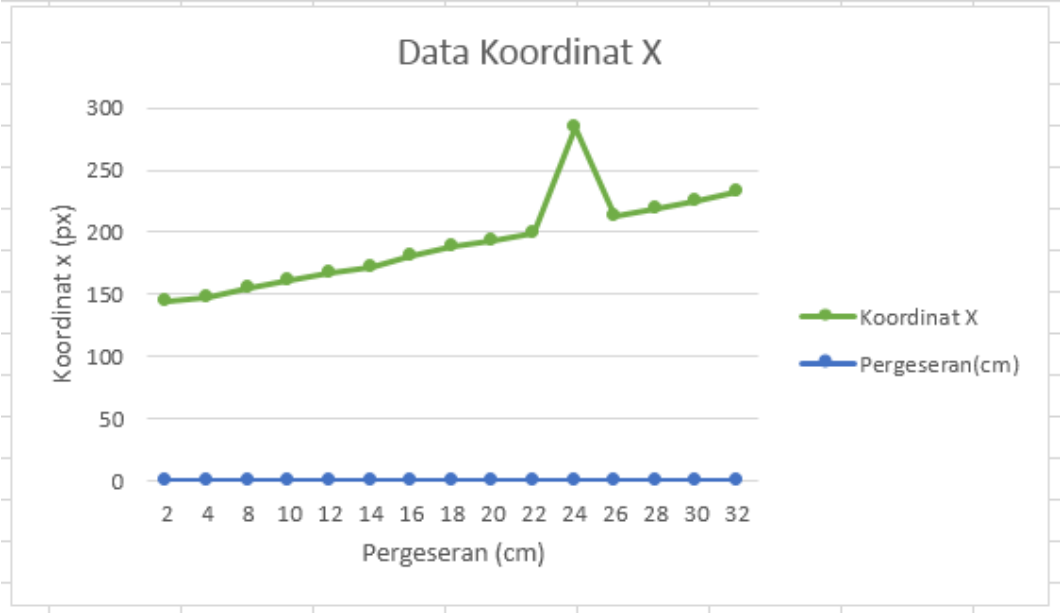
Tabel 4.4 Nilai pergeseran pixel menjauhi kamera

Data pergeseran bola menjauhi kamera				
X (px)	**Y (px)**	**Radius (px)**	**Pergeseran (cm)**	**Ket**
167	59	45	60cm	Bola Terdeteksi
167	39	20	120cm	Bola Terdeteksi

163	39	15	180cm	Bola Terdeteksi
159	35	12	240cm	Bola Terdeteksi
157	31	10	300cm	Bola Terdeteksi
157	29	8	360cm	Bola Terdeteksi
153	29	8	420cm	Bola Terdeteksi
153	29	6	480cm	Bola Terdeteksi
153	27	6	520cm	Bola Terdeteksi
181	3	24	600cm	Bola Terdeteksi
153	27	5	660cm	Bola Terdeteksi

4.6.6 Grafik Pergeseran *pixel* dengan Koordinat x

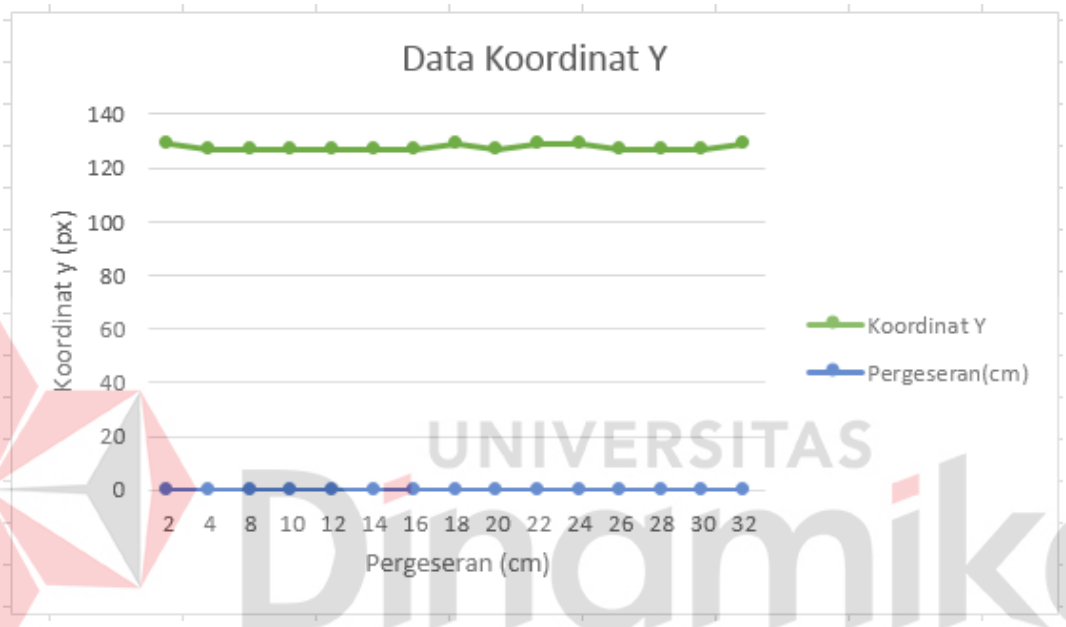
Jika dilihat pada grafik pada Gambar 4.21 untuk menentukan nilai pergeseran antara nilai sesungguhnya cm ke pixel sangatlah sulit bagi manusia untuk memperkirakan posisi letak yang tepat baik koordinat x maupun koordinat y.



Gambar 4.21 Nilai pergeseran pixel terhadap koordinat x

4.6.7 Grafik Pergeseran *pixel* dengan Koordinat y

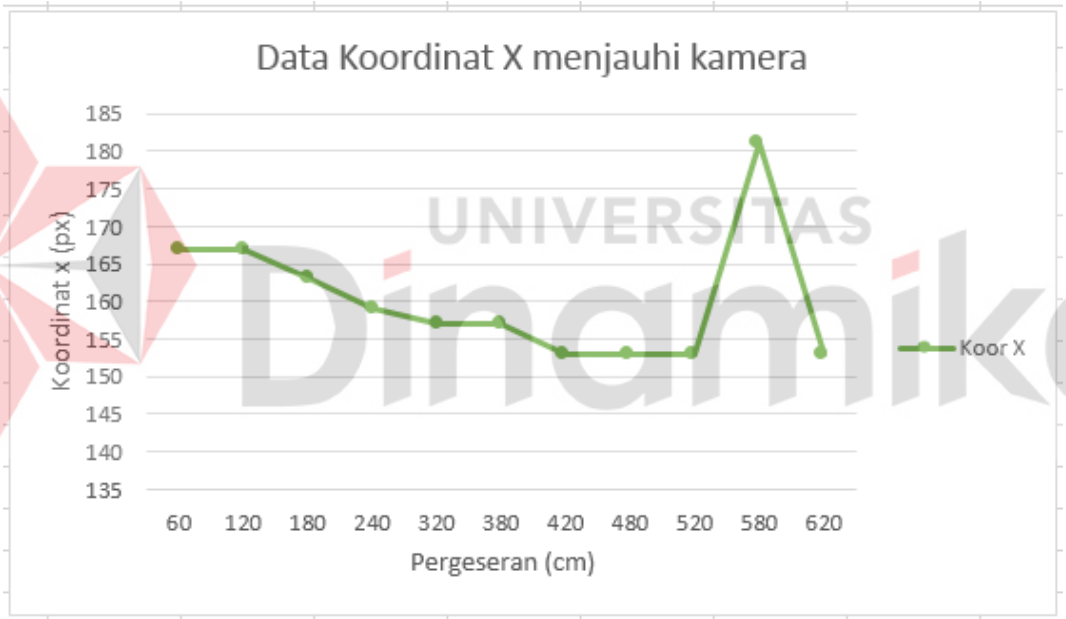
Pada Gambar 4.22 merupakan grafik dari pergeseran piksel sebesar 2cm jika mengacu pada nilai sesungguhnya. Namun perubahan pada nilai pixel masih belum dapat dikatakan sama. Dikarenakan terdapat loncatan – loncatan piksel yang tidak beraturan.



Gambar 4.22 Nilai pergeseran pixel terhadap koordinat y

4.6.8 Grafik Pergeseran *pixel* x menjauhi kamera

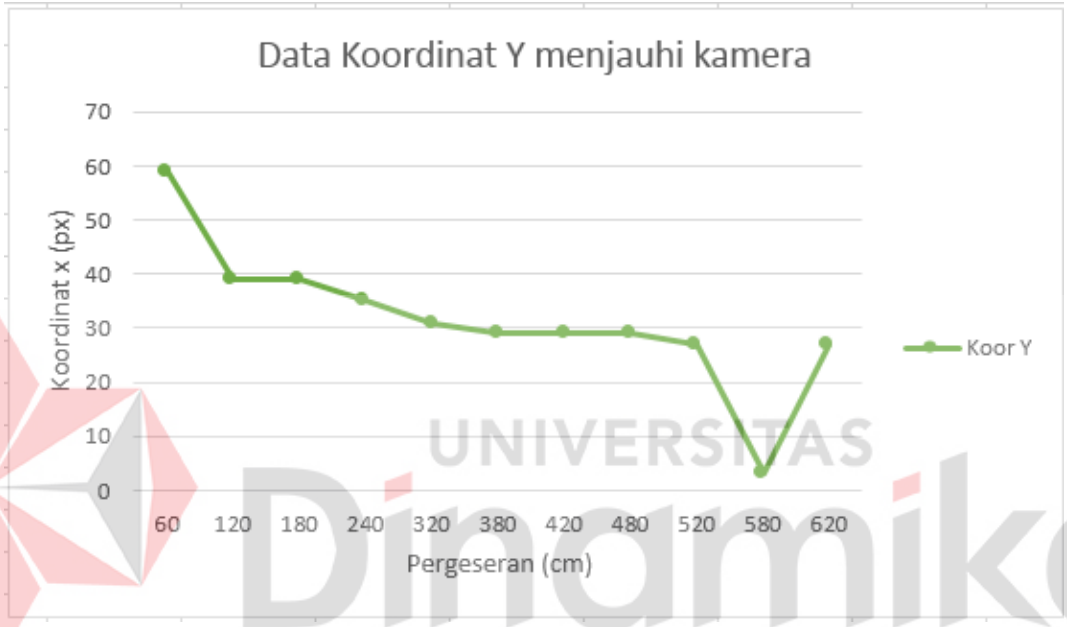
Pada Gambar 4.23 merupakan hasil dari pergeseran nilai piksel dengan cara bola yang dijadikan objek digeser menjahui kamera dengan jarak tiap 60cm. Dapat dikatakan bahwa jarak bola semakin jauh nilai piksel dari koordinat X juga mengalami perubahan yaitu semakin mengecil. Perlu diingat bahwa koordinat X pada kamera merupakan kolom piksel dan koordinat Y kamera merupakan baris dari piksel. Pada grafik jarak 520cm ke 620cm terdapat perubahan loncatan yang sangat tinggi.



Gambar 4.23 Nilai pergeseran pixel terhadap koordinat x

4.6.9 Grafik Pergeseran *pixel* y menjauhi kamera

Pada Gambar 4.24 merupakan nilai dari pergeseran piksel koordinat Y yang menjauhi kamera. Dimana yang menjauhi kamera merupakan objek bola yang telah ditentukan. Dapat dilihat bahwa semakin jauh bola diletakkan dan mengalami pergeseran dengan jarak tiap 60cm nilai dari piksel Y semakin kecil.



Gambar 4.24 Nilai pergeseran pixel terhadap y

BAB V

PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan dan saran tentang Pengolahan Citra pada robot SYNA SWC8 yang dimiliki oleh STIKOM CYBER ROBOTIC.

5.1 Kesimpulan

Kesimpulan yang diperoleh aelama melakukan Perancangan sistem Pengolahan Citra pada robot SYNA SWC8 adalah :

1. Untuk mengakses webcame C9220 sangatlah berat bagi Raspberry Pi 3 B+, maka dari itu perlu adanya pengurangan nilai pixel menjadi 640x480 pixel. Untuk meringankan kerja dari Raspberry Pi 3 B+.
2. Pada penelitian yang telah dilakukan ini proses pendeteksian bola hanya berdasarkan warna saja dengan ketentuan objek bola berwarna orange. Jika ada object yang memiliki warna yang sama akan terjadi proses pendeteksian yang salah tidak sesuai yang diharapkan.
3. Intensitas Cahaya yang berubah mempengaruhi dari proses pendeteksian bola.

UNIVERSITAS Dinamika

5.2 Saran

1. Dikarenakan sudut pandang robot terbatas karena menggunakan kamera depan saja, perlu adanya solusi untuk dapat menjadikan robot mampu melihat sekelilingnya tanpa dengan bantuan pergerakan mekanik robot. Yaitu menggunakan kamera katadioptrik yang memiliki sudut pandang 360°.
2. Perlu adanya perluasan teori dalam pengolahan citra telah dilakukan penelitian ini. Seperti halnya membedakan keberadaan bola di area lapangan hijau atau tidak.
3. Pada proses pendeteksian bola sangatlah sensitif dalam perubahan nilai warna yang menjadi target. Jadi perlu adanya cara untuk penyesuaian.
4. Robot perlu memiliki kemampuan untuk mendeteksi bola yang berada di dalam lapangan saja. Saat mendeteksi bola di luar lapangan diabaikan oleh sistem.
5. Robot juga perlu memiliki kemampuan untuk dapat mengerti jarak sesungguhnya dalam satuan cm. Bukan lagi informasi pixel yang didapat. Yang menggunakan semua informasi koordinat pada kamera yaitu x dan y. Dapat dilakukan dengan metode *Artifical Neural Network*.

UNIVERSITAS Dinamika

DAFTAR PUSTAKA

Akbar, E. I. (2017). *Rancang Bangun Omni Vision pada Robot ESPARTAN KRSBI Beroda*, 78-81.

Andono, P. N. (2017). *Pengolahan Citra Digital.* Yogyakarta: Andi.

Dr.Ir.Saludin Muis, M. (2017). *Sistem Kecerdasan Tiruan Dengan Kemampuan Belajar dan Adaptasi.* Ruko Jambusari 7A Yogyakarta: TEKNOSAIN.

Fahmizal. (2019). *Trajectory Tracking pada Robot Omni dengan Metode Odometry*, 35-44.

Muhammad, F. A. (2015). *SISTEM NAVIGASI ROBOT BERODA MENGGUNAKAN OMNIDIRECTIONAL CAMERA*, 1-5.

Prakoso, Y. (2017). *Artifical Neural Network Untuk Pengukuran Posisi Bola Menggunakan Kamera Omnidireksional Pada Robot Sepak Bola Beroda*.

Setyawan, N. (2018). *PENGENALAN POSISI OBJEK DARI SISTEM KAMERA OMNI-VISION MENGGUNAKAN PSO-NN DAN SCAN LINES PADA ROBOT SEPAK BOLA*, 27-33.

Siang, J. J. (2005). *Jaringan Syaraf Tiruan.* Yogyakarta: Andi.

Stone, A. A. (2017). *Segmentasi Citra pada Robot Sepak Bola Beroda Menggunakan Multilayer Neural Network dan Fitur Warna HSV*, A276-A281.

Widodo, F. a. (2017). *Rancang Bangun Sistem Lokomosi, Penggiring, dan Penendang pada Robot Sepak Bola*, 224-229.

Zhao, S. (2010). *Fisheye Lens Camera System Calibration And Localizaion Error Analysis*, V4-441-445.

UNIVERSITAS Dinamika