DETEKSI FITUR GAWANG PADA ROBOT BERBASIS *HOUGH TRANSFORM*

TUGAS AKHIR

Program Studi

S1 Sistem Komputer

Oleh :

BUDI SUSIANTO

14.41020.0036

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2018

DETEKSI FITUR GAWANG PADA ROBOT BERBASIS *HOUGH TRANSFORM*

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer

Disusun Oleh :

Nama	:	Budi Susianto
NIM	:	14.41020.0036
Program	:	S1 (Strata Satu)
Jurusan	:	Sistem Komputer

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2018

Mata kanal 'Adzmu Jalilan...

Wal Juhdu Dalilan...

Wa Dzikruallahi Mutawasilan...

Asbaha limtiyazdu Kholilan...

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

TUGAS AKHIR

DETEKSI FITUR GAWANG PADA ROBOT BERBASIS *HOUGH TRANSFORM*

Dipersiapkan dan disusun oleh

Budi Susianto

NIM : 14.41020.0036

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada :

Susunan Dewan Penguji

Pembimbing

I. Dr. Susijanto Tri Rasmana, S.Kom, M.T.

NIDN. 0727097302

II. Ira Puspasari, S.Si., M.T.

NIDN. 0710078601

Pembahas

I. Weny Indah Kusumawati, S.Kom., M.MT.

NIDN. 0721047201

Tugas Akhir ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEIKHLASAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama	: Budi Susianto
NIM	: 14.41020.0036
Program Studi	: S1 Sistem Komputer
Fakultas	: Fakultas Teknologi dan Informatika
Jenis Karya	: Tugas Akhir
Judul Karya	: **DETEKSI FITUR GAWANG PADA ROBOT BERBASIS *HOUGH TRANSFORM***

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right)* atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialih mediakan 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, Januari 2018

Yang menyatakan

Budi Susianto

NIM : 14.41020.0036

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

ABSTRAKSI

Permainan sepak bola merupakan sebuah kejadian yang bukan biasa-biasa saja di dunia, dengan penggemar dari berbagai umur mulai dari anak kecil sampai dengan orang tua. Kepopuleran permainan ini membuat para ilmuan untuk membuat sebuah robot yang dapat bermain sepak bola. Di Indonesia saat ini telah diadakan KRI (Kontes Robot Indonesia), yang salah satu divisi didalamnya mengambil tema sepak bola,yaitu KRSBI Beroda (Kontes Robot Sepak Bola Indonesia Beroda). Pada lomba ini, robot diharuskan mampu mendeteksi bola, menggiringnya bola dan menendangnya ke gawang.

Pada penelitian ini, penulis mengembangkan robot KRSBI Beroda 2017 yang masih memiliki beberapa kekurangan terutama dalam mendeteksi gawang. Adapun beberapa metode yang digunakan, yaitu konversi ruang warna HSV, *Color Filtering* untuk menyeleksi warna pada obyek, dan *Edge Detection* untuk mengubah gambar menjadi garis tepi, *Hough Transform* untuk membaca garis dan langkah terakhir yaitu mendeteksi sudut menggunakan Harris *Corner Detection.* Sehingga akan menghasilkan titik-titik pada setiap sudutnya. Koordinat dari setiap titik akan dikirim ke mikrokontroler sebagai acuan yang menentukan posisi robot untuk bergerak.

Hasil dari penelitian ini, dilakukan beberapa kali percobaan sehingga didapatkan data-data berupa posisi dan jarak antara robot dan gawang. Pendeteksian gawang tampak jelas secara visual pada jarak di atas 50 cm dan posisi antara -50^{o} sampai 50^{o}.

Kata Kunci : *Hough Transform,* Harris *Corner Detection*, KRSBI Beroda.

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur ke haddirat Allah SWT yang telah memberikan kekuatan, kesehatan lahir dan batin sehingga penulis dapat menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul "*Deteksi Fitur Gawang pada Robot Berbasis Hough Transform*" ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Institut Bisnis dan Informatika Stikom Surabaya

Pada kesempatan kali ini penulis juga ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu dan Kakak tercinta yang telah memberikan dukungan dan doa selama mengerjakan Tugas Akhir ini.
2. Pimpinan Stikom Surabaya yang telah banyak memberikan motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi Sistem Komputer Stikom Surabaya yang senantiasa memberikan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
4. Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., dan Ibu Ira Puspasari, S.Si., M.T., selaku dosen pembimbing satu dan dua yang telah membantu serta mendukung setiap kegiatan sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

5. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku penguji yang senantiasa memberikan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
6. Seluruh dosen Pengajar Program Studi S1 Sistem Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Institut Bisnis dan Informatika Stikom Surabaya.
7. Teman-teman dari komunitas Cyber Robotic Stikom Surabaya khususnya divisi KRSBI Beroda SYNA SWC-8, teman-teman seperjuangan Angkatan 2014, adik dan kakak angkatan Jurusan S1 Sistem Komputer yang mendukung dan membantu penulis selama masa dan penyusunan buku Tugas Akhir ini.
8. Seluruh pihak yang tidak dapat penulis tuliskan satu persatu yang telah membantu penulis secara langsung maupun tidak langsung.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang besar jika terdapat kata-kata yang salah serta menyinggung perasaan pembaca. Akhir kata penulis ucapkan banyak terima kasih yang besar kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, Januari 2018

Penulis

DAFTAR ISI

MOTTO iii

SURAT PERNYATAAN v

ABSTRAK vi

KATA PENGANTAR vii

DAFTAR GAMBAR xii

DAFTAR TABEL xiv

DAFTAR LAMPIRAN xv

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Perumusan Masalah 2

1.3 Pembatasan Masalah 2

1.4 Tujuan 3

BAB II LANDASAN TEORI 4

2.1 Konversi Warna HSV 4

2.2 *Color Filtering* 5

2.3 *Edge Detection* 8

2.4 *Hough Transform* 14

2.5 Harris *Corner Detection* 19

2.6 OpenCV 22

2.7 Raspberry Pi 3 23

2.8 Robot KRSBI Beroda Stikom Surabaya 2017 25

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

BAB III METODE PENELITIAN 27

3.1 Metode Penelitian 27

3.2 Pengolahan Obyek pada Rasberry Pi 28

3.2.1 Konversi HSV 29

3.2.2 *Color Filtering* 31

3.2.3 *Edge Detection* 32

3.2.4 *Hough Transform* 33

3.2.5 Harris *Corner Detection* 35

3.3 Flowchart Program 37

BAB IV HASIL PENGUJIAN DAN PENGAMATAN 39

4.1 Pengujian Program *Color Filtering* 39

4.1.1 Tujuan Pengujian Program *Color Filtering* 39

4.1.2 Alat yang Digunakan pada Pengujian Program *Color Filtering* 39

4.1.3 Prosedur Pengujian Program *Color Filtering* 39

4.1.4 Hasil Pengujian Program *Color Filtering* 40

4.2 Pengujian Program *Edge Detection* 44

4.2.1 Tujuan Pengujian Program *Edge Detection* 44

4.2.2 Alat yang Digunakan pada Pengujian Program *Edge Detection* 44

4.2.3 Prosedur Pengujian Program *Edge Detection* 44

4.2.4 Hasil Pengujian Program *Edge Detection* 45

4.3 Pengujian Program *Hough Transform* 46

4.3.1 Tujuan Pengujian Program *Hough Transform* 46

4.3.2 Alat yang Digunakan pada Pengujian Program *Hough Transform* 47

4.3.3 Prosedur Pengujian Program *Hough Transform* 47

4.3.4 Hasil Pengujian Program *Hough Transform* 48

4.4 Pengujian Program Harris *Corner Detection* 50

4.4.1 Tujuan Pengujian Program Harris *Corner Detection* 50

4.4.2 Alat yang Digunakan pada Pengujian Program Harris *Corner Detection* 50

4.4.3 Prosedur Pengujian Program Harris *Corner Detection* 50

4.4.4 Hasil Pengujian Program Harris *Corner Detection* 51

4.5 Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot 52

4.5.1 Tujuan Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot 52

4.5.2 Alat yang Digunakan pada Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot 52

4.5.3 Prosedur Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot 53

4.5.4 Hasil Pengujian Program Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot 53

BAB V PENUTUP 59

5.1 Kesimpulan 59

5.2 Saran 60

DAFTAR PUSTAKA 61

LAMPIRAN 62

BIODATA PENULIS 65

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

DAFTAR GAMBAR

Gambar 2.1 Model Warna HSV 4
Gambar 2.2 *Color Filtering* 6
Gambar 2.3 Proses Deteksi Tepi 8
Gambar 2.4 Sebelum Proses Deteksi Tepi 9
Gambar 2.5 Setelah Proses Deteksi Tepi 9
Gambar 2.6 Hasil Deteksi Tepi 10
Gambar 2.7 Cadar Gausian 11
Gambar 2.8 Matrik 5x5 12
Gambar 2.9 Arah Derajat Tepi 12
Gambar 2.10 Penggolongan Nilai Piksel 14
Gambar 2.11 Grafik Kartesian Persamaan 17
Gambar 2.12 (a) Garis Lurus pada X-Y (b) Representasi dalam Ruang Parameter *m-c* 18
Gambar 2.13 Hasil Persamaan (2.7) 18
Gambar 2.14 Hasil *Hough Transform* 19
Gambar 2.15 Harris *Corner Detection* 21
Gambar 2.16 Gambar Struktur OpenCV 23
Gambar 2.17 Raspberry Pi 23
Gambar 2.18 Robot KRSBI Beroda 2017 26
Gambar 3.1 Blok Diagram Penelitian 27
Gambar 3.2 Proses Pengolahan pada *Image Processing* 28
Gambar 3.3 Ruang Warna RGB 30
Gambar 3.4 Ruang Warna HSV 30

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Gambar 3.5 Hasil *Color Filtering* 32
Gambar 3.6 Hasil Pengolahan *Edge Detection* 33
Gambar 3.7 Hasil Deteksi Tepi Garis 34
Gambar 3.8 Hasil Harris *Corner Detection* 36
Gambar 3.9 Hasil Tampilan Koordinat 36
Gambar 3.10 Flowchart Deteksi Gawang 38
Gambar 4.1 Hasil Sebelum Konversi 40
Gambar 4.2 Hasil Setelah Konversi 41
Gambar 4.3 Trackbar Sebelum Filter 41
Gambar 4.4 Proses Pemfilteran 42
Gambar 4.5 Hasil Setelah Filter 43
Gambar 4.6 Hasil Pemfilteran 43
Gambar 4.7 Proses Deteksi Tepi 46
Gambar 4.8 Hasil Deteksi Tepi 46
Gambar 4.9 Nilai Resolusi Piksel Adalah 1 48
Gambar 4.10 Nilai Resolusi Piksel Adalah 0,5 49
Gambar 4.11 Nilai Resolusi Piksel Adalah 0,1 49
Gambar 4.12 Hasil Deteksi Sudut 51
Gambar 4.13 Nilai Koordinat 52
Gambar 4.14 Variasi Jarak Gawang dan Robot 54
Gambar 4.15 Variasi Posisi Gawang dan Robot 56

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

DAFTAR TABEL

Tabel 3.1 Nilai RGB pada Sebuah Gambar 29

Tabel 3.2 Nilai HSV pada Sebuah Gambar 29

Tabel 4.1 Hasil dari Tiap Percobaan Color Filtering 42

Tabel 4.2 Hasil Pengujian Deteksi Gawang dengan Variasi Jarak 54

Tabel 4.3 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 100 cm 56

Tabel 4.4 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 200 cm 57

Tabel 4.5 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 300 cm 57

Tabel 4.6 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 400 cm 58

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

DAFTAR LAMPIRAN

Lampiran 1 Program Keseluruhan 62

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi robotika di masa sekarang sudah menjadi bagian penting dalam kehidupan manusia. Robot dalam beberapa tahun ini ada yang dapat menggantikan peran manusia, hal ini terlihat pada robot-robot yang diterapkan dalam berbagai bidang seperti industri, kesehatan (*health*), pertahanan *(defense)*, pemadam api (*fire extinguisher*), penelitian *(research)*, pemainan *(game)*, dan lain-lain. Bahkan selain membantu dalam hal pekerjaan, robot juga bisa menjadi media hiburan. Programer memerintahkan robot untuk bergerak seperti halnya manusia. Bisa melakukan kegiatan yang dilakukan manusia, begitu juga dengan olahraga, salah satunya sepak bola.

Permainan sepak bola saat ini merupakan fenomena yang luar biasa di dunia. Jenis permainan ini telah melanda hampir seluruh penjuru dunia, dengan penggemar dari berbagai segmen mulai dari anak-anak sampai dengan kaum lanjut usia. Kepopuleran permainan sepak bola tersebut mendorong para ilmuwan untuk menciptakan robot yang bisa memainkan sepak bola. Ide itu kemudian terealisasikan sekitar satu setengah dekade lalu, tepatnya tahun 1998 saat dilangsungkannya putaran final piala dunia di Prancis. Sampai dengan saat ini kejuaraan sepak bola robot tingkat dunia digelar setiap tahun, dengan penyelenggara yang berbeda-beda tiap tahunnya. Para ilmuwan menargetkan pada tahun 2050 juara piala dunia sepak bola robot akan diadu dengan juara piala dunia sepak bola manusia (Hadi, 2012).

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Di Indonesia saat ini telah diadakan KRI (Kontes Robot Indonesia), yang salah satu divisi didalamnya mengambil tema sepak bola,yaitu KRSBI Beroda (Kontes Robot Sepak Bola Indonesia Beroda). Pada lomba ini, robot diharuskan mampu mendeteksi bola, menggiringnya bola dan menendangnya ke gawang.

Pada penelitian ini, penulis mengembangkan robot KRSBI Beroda 2017 yang masih memiliki beberapa kekurangan terutama dalam mendeteksi gawang. Sensor yang digunakan untuk mendeteksi gawang adalah sebuah kamera, dimana hasil pembacaan kamera akan diolah dengan menggunakan metode *Hough Transform*. Data hasil pengolahan akan dikirim ke mikrokontroler sebagai masukan untuk proses selanjutnya.

1.2. Perumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah:

1. Bagaimana menerapkan metode *Houge Transform* untuk mendeteksi gawang.
2. Bagaimana menghasilkan titik koordinat dari gawang yang sudah diolah dengan metode *Hough Transform*.
3. Menganalisis hasil deteksi gawang dalam beberapa variasi posisi robot terhadap gawang.

1.3. Pembatasan Masalah

Pada penelitian ini, terdapat beberapa batasan masalah, antara lain:

1. Menggunakan Robot KRSBI Beroda 2017.
2. Gawang berbentuk persegi panjang.
3. Gawang hanya memiliki 1 warna (kuning).

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4. Pencahayaan tidak boleh berubah karena dapat mempengaruhi warna pada gawang.
5. Robot berjalan pada lapangan dengan permukaan datar.
6. Tidak terdapat halangan antara robot dan gawang.

1.4. Tujuan

Adapun tujuan pada penelitian ini adalah sebagai berikut:

1. Menerapkan metode *Houge Transform* untuk mendeteksi gawang.
2. Menghasilkan titik koordinat dari gawang yang sudah diolah dengan metode *Hough Transform*.
3. Analisis deteksi gawang dalam beberapa variasi posisi robot terhadap gawang.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

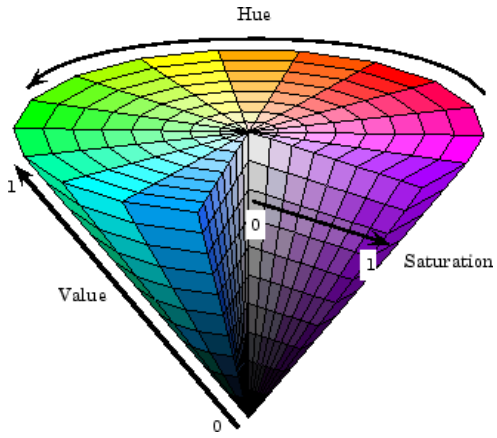
BAB II

LANDASAN TEORI

2.1. Konversi Warna HSV

Warna adalah hasil persepsi dari cahaya dalam spectrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400 nm sampai dengan 700 nm. Sedangkan ruang warna adalah model matematis abstrak yang menggambarkan cara agar suatu warna dapat direpresentasikan sebagai baris angka biasanya dengan nilai-nilai dari tiga atau empat buah warna atau komponen, contohnya adalah ruang warna RGB, ruang warna CMY/CMYK, ruang warna YIQ, ruang warna YCbCr, ruang warna HSI, HSL, HSV, ruang warna CIELAB.

HSV mendefinisikan warna dalam terminology Hue, Saturation dan Value. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. Sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer (Putra, 2017). Model warna HSV dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model Warna HSV

Model HSV, pertama kali diperkenalkan oleh A.R. Smith pada tahun 1978, yang ditunjukkan pada gambar. HSV memiliki 3 karakteristik pokok, yaitu **Hue**, **Saturation,** dan **Value.**

- **Hue :** menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahan (redness), kehijauan (greeness), dsb.
- **Saturation :** kadang disebut chroma, adalah kemurnian atau kekuatan warna.
- **Value :** kecerahan dari warna. Nilainya berkisar antara 0-100 %. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.

Adapun rumus untuk melakukan konversi pada persamaan (2.1)

$$V = \max(R,G,B)$$

$$S = \begin{cases} \frac{V-\min(R,G,B)}{V} & if\ V \neq 0 \\ 0 & otherwise \end{cases}$$

$$H = \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & if\ V = R \\ 120 + \frac{60(B-R)}{V-\min(R,G,B)} & if\ V = G \\ 240 + \frac{60(R-G)}{V-\min(R,G,B)} & if\ V = B \end{cases}$$

Output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$ (2.1)

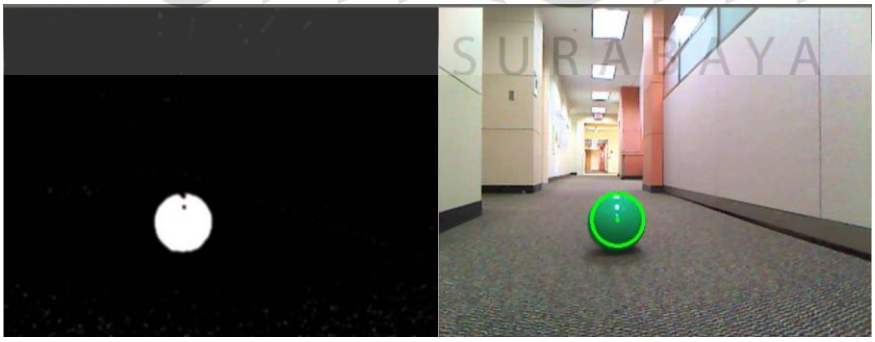
2.2. *Color Filtering*

Color Filtering adalah metode yang digunakan untuk menyeleksi atau menemukan sebuah warna tertentu dari gambar. Dengan ditemukannya warna yang dicari pada sebuah gambar, maka dapat melanjutkan proses apa yang selanjutnya harus dilakukan.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Pada dasarnya metode ini menggunakan kombinasi dari komponen warna-warna tertentu yang terdapat pada sebuah gambar. Nilai dari masing-masing komponen ini didapatkan dari hasil beberapa eksperimen. Hasil dari eksperimen ini adalah sebuah kombinasi warna-warna tertentu. Kombinasi warna ini kemudian dijadikan sebuah filter yang merupakan penentu sebuah warna diloloskan atau tidak (Prabowo, 2013).

Keluaran dari metode ini akan langsung menghasilkan sebuah gambar biner. Gambar biner sendiri adalah sebuah gambar yang hanya memiliki 2 derajat keabuan yaitu hitam dan putih. Dimana warna putih pada gambar adalah merupakan representai dari sebuah piksel yang warnanya diloloskan. Sedagkan warna hitam pada gambar adalah representasi dari piksel yang warnanya tidak diloloskan. Contoh hasil *Color Filtering* dapat dilihat pada Gambar 2.2, sehingga dapat disimpulkan bahwa warna putih pada gambar adalah obyek yang dicari. Karena warna putih pada gambar merupakan warna pada piksel yang sesuai dengan filter yang didapatkan dari hasil eksperimen.



Gambar 2.2 *Color Filtering*

Adapun jenis *Color Filtering* yang digunakan adalah *Band Pass Filter* (BPF), yaitu jenis filter yang melewatkan frekuensi pada daerah tertentu di antara

frekuensi cut-off pertama dan frekuensi cut-off kedua dan meredam frekuensi di luar daerah tersebut. Selain itu *Band Pass Filter* merupakan sebuah metode yang dirancang untuk melewatkan frekuensi dalam batasan tertentu dan menolak frekuensi lain di luar frekuensi yang dikehendaki. Dan *Band Pass Filter* merupakan gabungan antara H*igh Pass* dan *Low Pass Filter*. Filter ini akan meneruskan sinyal-sinyal dengan frekuensi antara (*median frequency*) dan menahan frekuensi di bawah dan di atas median tersebut.

Dalam *Band Pass Filter* ini dikenal 2 jenis rangkaian yaitu *Band Pass Filter* (BPF) bidang lebar dan *Band Pass Filter* (BPF) bidang sempit. Untuk membedakan kedua rangkaian ini adalah dengan melihat dari nilai ***figure of merit* (FOM)** atau **Faktor kualitas (Q).**

- Bila Q < 10, maka digolongkan sebagai *Band Pass Filter* (BPF) bidang lebar.
- Bila Q > 10, maka digolongkan sebagai *Band Pass Filter* (BPF) bidang sempit.

Perhitungan faktor kualitas (Q) adalah :

1. *Band Pass Filter* Bidang Lebar

Syarat BPF bidang lebar adalah Q<10, biasanya didapat dari 2 rangkaian filter HPF dan LPF yang mereka saling diseri dengan urutan tertentu dan frekuensi cut off harus tertentu.

2. *Band Pass Filter* Bidang Sempit

Syarat BPF bidang sempit adalah Q > 10, Rangkaian ini sering disebut multiple feedback filter karena satu rangkaian menghasilkan 2 batasan fL dan fH .

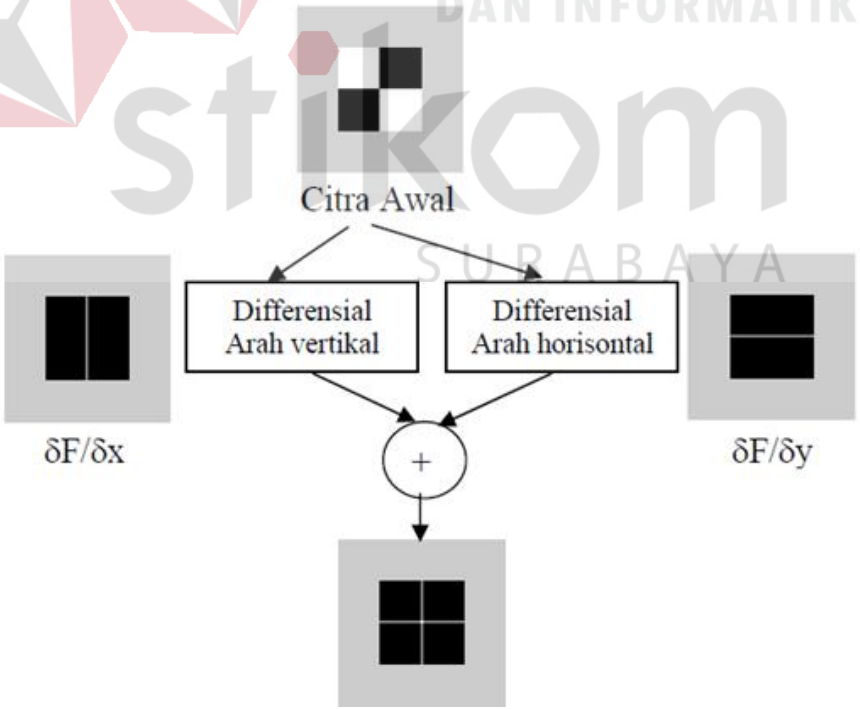
2.3. *Edge Detection*

Deteksi tepi *(Edge detection)* adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Proses deteksi tepi citra dilakukan dengan mencari lokasi-lokasi intensitas piksel-piksel yang *discontinue* dengan intensitas piksel- piksel yang berdekatan (bertetanggaan/*neighborhood*). Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra, bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya (Yunus, n.d.). Tujuannya adalah :

- Untuk menandai bagian yang menjadi detail citra.
- Untuk memperbaiki detail dari citra yang kabur, yang terjaid karena error atau adanya efek dari proses akuisi citra

Proses deteksi tepi dapat dilihat pada Gambar 2.3.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA



Gambar 2.3 Proses Deteksi Tepi

Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan *High Pass Filter* (HPF). Misalkan terdapat fungsi suatu citra sebagai f (x, y) sebagai berikut:

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

Gambar 2.4 Sebelum Proses Deteksi Tepi

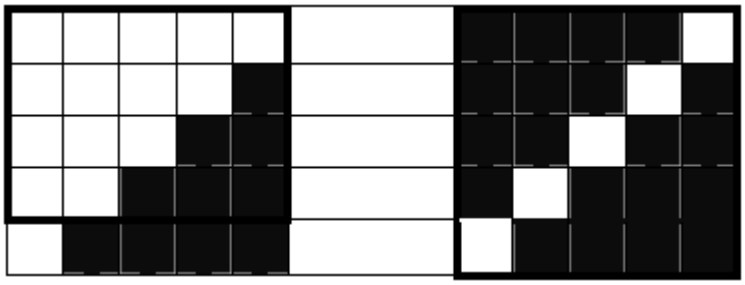
Dengan menggunakan fungsi filter H (x,y) = [-1 1], akan diperoleh matrik citra sebagai berikut:

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

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Gambar 2.5 Setelah Proses Deteksi Tepi

sehingga bila digambarkan maka proses filter tersebut memiliki piksel-piksel masukan dan keluaran sebagai berikut:



Gambar 2.6 Hasil Deteksi Tepi

Proses deteksi tepi (*Edge Detection*) sendiri dapat dikelompokkan berdasarkan operator atau metode yang digunakan dalam proses pendeteksian tepi suatu citra (*Edge Detection*) untuk memperoleh citra hasil. Pada penelitian ini akan digunakan metode deteksi tepi jenis *Canny*. Operator *Canny*, yang dikemukakan oleh John Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan yang rendah, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi.

Terdapat enam langkah yang dilakukan untuk mengimplementasikan deteksi tepi *Canny*. Keenam langkah tersebut dijabarkan berikut ini;

Langkah 1:

Pertama-tama dilakukan penapisan terhadap citra dengan tujuan untuk menghilangkan derau. Hal ini dapatdilakukan dengan menggunakan filter *Gaussian* dengan cadar sederhana. Cadar yang digunakan berukuran jauh lebih kecil daripada ukuran citra. Contoh ditunjukkan pada Gambar 2.7.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

1 / 115

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Gambar 2.7 Cadar Gausian

Langkah 2:

Setelah penghalusan gambar terhadap derau dilakukan, dilakukan proses untuk mendapatkan kekuatan tepi (*edge strength*). Hal ini dilakukan dengan menggunakan operator *Gaussian*. Selanjutnya, gradien citra dapat dihitung melalui persamaan (2.2).

$$|G| = |Gx| + |Gy| \quad (2.2)$$

Langkah 3:

Langkah ketiga berupa penghitungan arah tepi, maka digunakan persamaan (2.3).

$$theta = \tan^{-1}(Gy, Gx) \quad (2.3)$$

Langkah 4:

Setelah arah tepi diperoleh, perlu menghubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra. Sebagai contoh, terdapat susunan piksel berukuran 5x5 seperti terlihat pada Gambar 2.8. Dengan melihat piksel “a” tampak bahwa a hanya memiliki 4 arah berupa 0^o, 45^o, 90^o, dan 135^o.

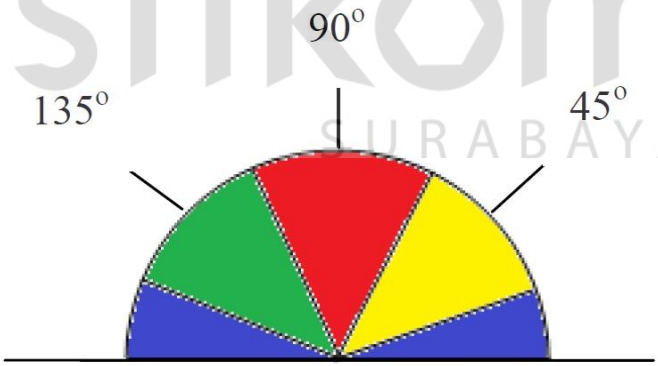
INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

X	X	X	X	X
X	X	X	X	X
X	X	a	X	X
X	X	X	X	X
X	X	X	X	X

Gambar 2.8 Matrik 5x5

Selanjutnya, arah tepi yang diperoleh akan dimasukkan ke dalam salah satu kategori dari keempat arah tadi berdasarkan area yang tertera pada persamaan (2.4).

$$arah = \begin{cases} 0 \\ 45 \\ 90 \\ 135 \end{cases} \begin{matrix} y1(H)=f(H)\left(\sum_j Wij(H)Xj\right) \\ y1=f(T)\left(\sum_j Wij(TH)yj(H)\right) \end{matrix} \quad (2.4)$$



Gambar 2.9 Arah Derajat Tepi

Semua arah tepi yang berkisar antara 0 dan 22,5 serta 157,5 dan 180 derajat (warna biru) diubah menjadi 0 derajat. Semua arah tepi yang berkisar antara 22,5 dan 67,5 derajat (warna kuning) diubah menjadi 45 derajat. Semua arah tepi yang

438 berkisar antara 67,5 dan 112,5 derajat (warna merah) diubah menjadi 90 derajat. Semua arah tepi yang berkisar antara 112,5 dan 157,5 derajat (warna hijau) diubah menjadi 135 derajat.

Langkah 5

Setelah arah tepi diperoleh, penghilangan non-maksimum dilaksanakan. Penghilangan non-maksimum dilakukan di sepanjang tepi pada arah tepi dan menghilangkan piksel-piksel (piksel diatur menjadi 0) yang tidak dianggap sebagai tepi. Dengan cara seperti itu,diperoleh tepi yang tipis.

Langkah 6:

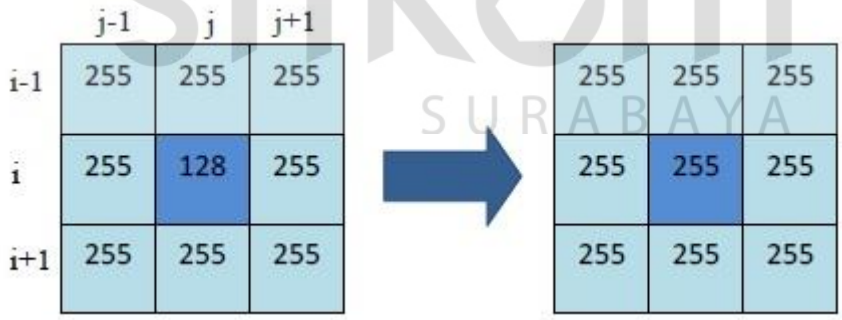
Langkah keenam berupa proses yang disebut *hysteresis*. Proses ini menghilangkan garis-garis yang seperti terputus-putus pada tepi obyek. Caranya adalah dengan menggunakan dua ambang T1 dan T2. Lalu, semua piksel citra yang bernilai lebih besar daripada T1 dianggap sebagai piksel tepi. Selanjutnya, semua piksel yang terhubung dengan piksel tersebut dan memiliki nilai lebih besar dari T2 juga dianggap sebagai piksel tepi.

Bagian penting yang perlu dijelaskan adalah penghilangan non-maksimum dan peng-ambangan histeresis. Penghilangan non-maksimum dilakukan dengan mula-mula menyalin isi larik *Grad* (yang berisi besaran gradien) ke `Non_max`. Selanjutnya, penghilangan non-maksimum dilaksanakan dengan memperhatikan dua titik tetangga yang terletak pada arah tepi (yang tersimpan dalam *Theta*). Misalnya, arah tepi adalah 0. Apabila titik yang menjadi perhatian mempunyai koordinat (r, c), dua titik tetangga berupa (r, c-1) dan (r, c+1). Apabila gradien titik perhatian lebih besar daripada gradien kedua tetangga, nilainya akan

dipertahankan. Sebaliknya, jika nilai titik perhatian lebih kecil daripada nilai salah satu atau kedua gradien tetangga, nilainya akan diabaikan (diubah menjadi nol).

Pengembangan histeresis dilakukan dengan melibatkan dua ambang T1 (ambang bawah) dan ambang T2 (ambang atas). Nilai yang kurang dari T1 akan diubah menjadihitam (nilai 0) dan nilai yang lebih dari T2 diubah menjadi putih (nilai 255). Lalu, bagaimana nilai yang lebih dari atau sama dengan T1 tetapi kurang dariT2? Oleh karena itu, untuk sementara nilai pada posisi seperti itu diberi nilai 128, yang menyatakan nilai abu-abu atau belum jelas, akan dijadikan 0 atau 255.

Selanjutnya, dilakukan pengujian untuk mendapatkan kondisi seperti tercantum pada Gambar 2.10. Apabila kondisi seperti itu terpenuhi, angka 128 diubah menjadi 255. Proses pengujian seperti itu dilakukan sampai tidak ada lagi perubahan dari nilai 128 menjadi 255. Tahap selanjutnya, semua piksel yang bernilai 128 yang tersisa diubah menjadi nol.



Gambar 2.10 Penggolongan Nilai Piksel

2.4. *Hough Transform*

Hough Transform pertama kali diperkenalkan oleh **Paul Hough** pada tahun 1962 untuk mendeteksi garis lurus. *Hough Transform* adalah teknik transformasi

citra yang dapat digunakan untuk mengisolasi atau dengan kata lain memperoleh fitur dari sebuah citra. Karena tujuan dari sebuah transformasi adalah mendapatkan suatu fitur yang lebih spesifik, *Classical Hough Transform* merupakan teknik yang paling umum digunakan untuk mendeteksi obyek yang berbentuk kurva seperti garis, lingkaran, elips dan parabola. Keuntungan utama dari transformasi *Hough* adalah dapat mendeteksi sebuah tepian dengan celah pada batas fitur dan secara relative tidak dipengaruhi oleh derau atau *noise* (Arief, 2012).

Teori dasar *Hough Transform* menggunakan bentuk parametrik dan menggunakan pemungutan suara terbanyak (*voting*) untuk menentukan nilai parameter yang tepat. Apabila dalam citra terdapat beberapa garis yang saling berpotongan pada suatu titik, maka apabila kemudian titik tersebut ditransformasi ke dalam ruang parameter *m-c* kemudian untuk mengetahui transformasi titik tersebut adalah sebuah garis lurus atau bukan dilakukan voting dengan persamaan garis yang dinyatakan dalam persamaan (2.5).

$$y_i = mx_i + c \tag{2.5}$$

Di dalam sebuah citra yang terdapat garis lurus maka garis tersebut mempunyai persamaan yang sama dengan persamaan (2.5), maka apabila ditransformasi ke dalam ruang parameter *m-c* akan diperoleh hasil beberapa garis yang saling berpotongan dalam suatu titik. Persamaan transformasi diperoleh dengan cara memanipulasi persamaan garisnya yaitu persamaan (2.5) menjadi bentuk parametrik menjadi persamaan (2.6).

$$c = y_i - mx_i \tag{2.6}$$

Misalnya dari persamaan garis 4x + 3y = 12 kita dapat menggambar grafiknya pada **koordinat Cartesius**, dengan cara;

- Langkah pertama adalah membuat table koordinat dan cukup dipilih dua pasangan koordinat yang mudah, yaitu titik potong terhadap sumbu x dan sumbu y.
- Jika diubah menjadi persamaan $c = y - mx$ menjadi: $4 = y - (-4/3x)$ atau $4 = y + 4/3x$.
- Titik potong terhadap sumbu x, y = 0

  $y + 4/3x = 4$

  $0 + 4/3x = 4$

  $4/3x = 4$

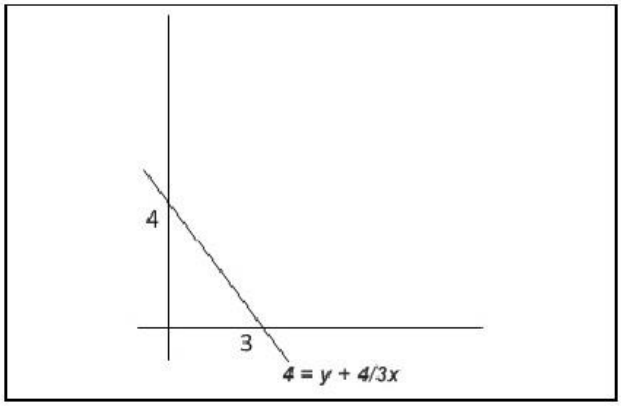
  $x = 3$, didapat titik (3, 0)
- Titik potong terhadap sumbu y, x = 0

  $y + 4/3x = 4$

  $y + 0 = 4$

  $y = 4$, didapat titik (0, 4)
- Jadi hasil grafik kartesian persamaan $4x + 3y = 12$ terlihat seperti Gambar 2.11.

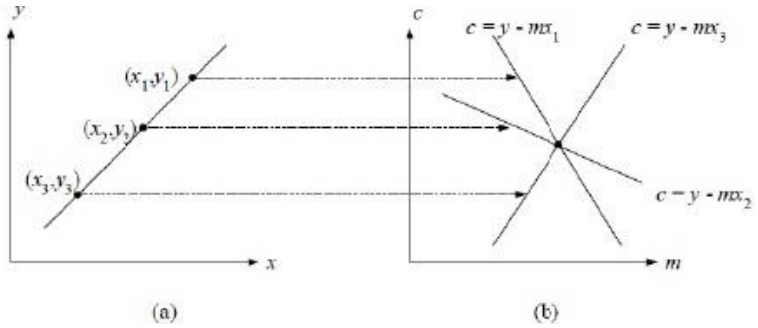
INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA



Gambar 2.11 Grafik Kartesian Persamaan

Pemisalan lain, terdapat 3 buah titik pada sebuah garis lurus. Sembarang garis yang melalui titik *(x1, y1)* berkoresponden dengan garis *c = y1 – mx1* pada ruang parameter *m-c*. Begitu juga, sembarang garis lurus yang melalui *(x2, y2)* berkoresponden dengan garis *c = y2 – mx2* dan sembarang garis lurus yang melalui *(x3, y3)* berkoresponden dengan garis *c = y3 – mx3* pada ruang *m-c*. Perpotongan *(m', c')* dari ketiga garis pada ruang *m-c* tersebut menentukan garis unik yang melalui *(xi, yi), i = 1, 2, 3*, di bidang *X - Y*.

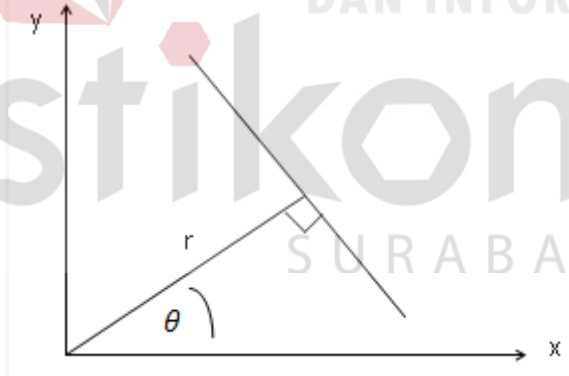
Dengan cara ini, maka setiap pixel pada garis lurus di bidang citra berkoresponden dengan sejumlah garis lurus yang melalui satu titik tertentu di ruang parameter *m-c*. Sifat ini dimanfaatkan untuk mendeteksi garis lurus. Jika setiap piksel tepi melakukan "*pemungutan suara*" pada ruang parameter, maka keberadaan garis lurus pada citra ditandai dengan penumpukan suara pada tempat-tempat tertentu di ruang parameter.



Gambar 2.12 (a) Garis Lurus pada X-Y (b) Representasi dalam Ruang Parameter *m-c*

Dalam kenyataannya, apabila ditemui sebuah garis vertikal, maka akan terjadi masalah dalam penghitungannya dikarenakan garis vertikal mempunyai nilai gradien kemiringan yang besarnya tak berhingga (∞). Sebagai alternatifnya digunakan persamaan (2.7).

$$r = x \cos + y \sin \quad (2.7)$$



Gambar 2.13 Hasil Persamaan (2.7)

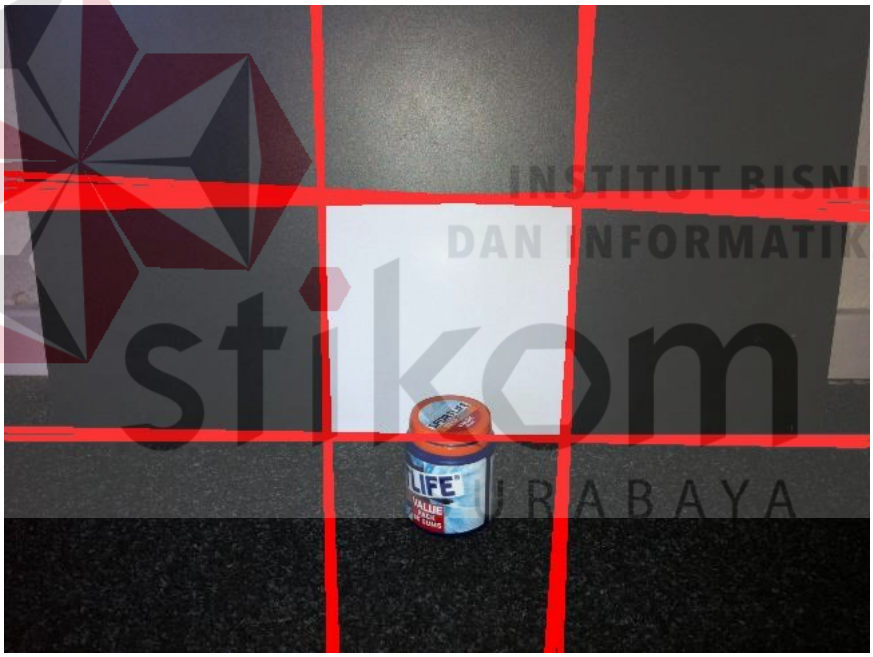
Keterangan :

- ***x*** **:** titik koordinat
- ***y*** **:** titik koordinat
- ***θ*** **:** sudut yang dibentuk antara obyek garis dengan sumbu x
- ***r*** **:** jarak antara garis dengan titik pusat (0,0)

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Apabila di dalam citra terdapat suatu garis lurus, maka jika garis lurus citra ditransformasi kedalam ruang parameter r - θ akan terjadi suatu titik penumpukan antar kurva sinusoida hasil pentransformasian masing-masing komponen piksel garis yang membentuk garis lurus tersebut.

Hough Transform memiliki beberapa perbedaan rumus yang diterapkan. Semuanya tergantung pada jenis obyek yang dicari. Pada penelitian ini jenis yang dipakai adalah *Hough Transform Line* karena untuk mendeteksi sebuah garis. Contoh hasil pengolahan citra dengan metode *Hough Transform* dapat dilihat pada Gambar 2.14.



Gambar 2.14 Hasil *Hough Transform*

2.5. Harris *Corner Detection*

Harris *Corner Detection* adalah sistem pendeteksi sudut yang sering digunakan karena mampu menghasilkan nilai yang konsisten pada citra yang mengalami rotasi, penskalaan, variasi pencahayaan maupun memiliki banyak

derau pada gambar. Dalam deteksi sudut menggunakan Harris, konvolusi dengan fungsi Gaussian merupakan langkah paling penting karena akan menentukan keseluruhan proses dan hasil akhir. Tapi justru ini yang membuat keseluruhan komputasi semakin besar karena proses yang dijalani adalah fungsi Gaussian dikalikan terlebih dahulu dengan piksel lalu dijumlahkan satu demi satu. Semakin besar mask Gaussian yang digunakan, semakin tinggi beban komputasi. Hal ini yang menjadi pendorong munculnya algoritma-algoritma terbaru yang memfasilitasi Harris dalam mendeteksi sudut. Sebagai contoh penggunaan Integral Image dan Box Filter dan lainnya (Shandy, 2012).

Harris *Corner Detection* memiliki kelebihan yakni hasil deteksi nyaris tidak berubah untuk kasus: rotasi terhadap citra, penskalaan citra, variasi cahaya dan derau pada citra. Harris didasarkan pada autokorelasi lokal isyarat yang mana mengukur perubahan lokal dari isyarat dengan nilai delta tertentu (nilainya kecil) yang digeser ke arah berbeda. Autokorelasi ini memiliki persamaan (2.8).

$$c(x,y)=\sum_W [I(xi,yi) - I(xi+\Delta x, yi+\Delta y)]^2 \quad (2.8)$$

dimana (xi, yi) merupakan titik pada jendela W (fungsi gaussian) yang berpusat di (x,y), I(.,.) menyatakan fungsi citra. (Δx, Δy) merupakan pergeseran yang dimaksud.

Sementara itu, citra yang digeser dirumuskan dengan ekspansi Taylor yakni dengan persamaan (2.9).

$$I(xi+\Delta x,yi+\Delta y) = I(xi,yi)+[Ix(xi,yi)Iy(xi,yi)]\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} \quad (2.9)$$

Dengan substitusi persamaan (2.9) ke persamaan (2.8) maka kita bisa menurunkan persamaan (2.10).

$$c(x,y) \quad = \sum_W [I(xi, yi) - I(xi + \Delta x, yi + \Delta y)]^2$$

$$= \sum_W (I(xi, yi) - I(xi,yi) - [Ix(xi,yi)Iy(xi,yi)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= \sum_W (-[Ix(xi, yi)Iy(xi, yi)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^2$$

$$= \sum_W ([Ix(xi, yi)Iy(xi, yi)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^2$$

$$= [\Delta x \Delta y] \begin{bmatrix} \sum_w (Ix(xi, yi))2 & \sum_w Ix(xi, yi)Iy(xi, yi) \\ \sum_w Ix(xi, yi)Iy(xi, yi) & \sum_w (Iy(xi, yi))2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= [\Delta x \Delta y] C(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad (2.10)$$

Adapun contoh dari hasil Harris *Corner Detection* ditunjukkan pada Gambar 2.15.

Gambar 2.15 Harris *Corner Detection*

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

2.6. OpenCV

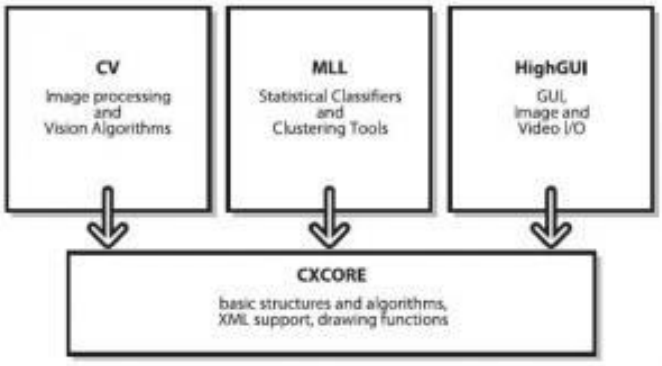
OpenCV (*Open Computer Vision*) adalah sebuah **API** (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision. Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan computer dapat melihat seperti manusia. Dengan vision tersebut computer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu obyek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition, Face Detection, Face/Object Tracking, Road Tracking dll*. OpenCV adalah library *Open Source* untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video (Priawadi, 2017).

OpenCV sendiri terdiri dari 5 library, yaitu:

- CV : untuk algoritma Image processing dan Vision.
- ML : untuk machine learning library
- Highgui : untuk GUI, Image dan Video I/O.
- CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.
- CvAux

Struktur dan Konten OpenCV:

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA



Gambar 2.16 Gambar Struktur OpenCV

2.7. Raspberry Pi 3

Raspberry Pi 3 Model B adalah generasi ketiga Raspberry Pi. Raspberry Pi 3 ini berukuran tunggal board komputer dapat digunakan untuk banyak aplikasi dan menggantikan asli Raspberry Pi Model B+ dan Raspberry Pi 2 Model B. Sementara mempertahankan bentuk mini PC Raspberry Pi 3 B membawa processer yang lebih cepat, 10x lebih cepat dari generasi pertama Raspberry Pi. Selain itu menambah nirkabel konektivitas Wifi, LAN & Bluetooth menjadikannya solusi ideal untuk desain komunikasi yang bagus (ThePiHut, 2017).

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Gambar 2.17 Raspberry Pi

Spesifikasi ;

Processor	Broadcom BCM2387 *chipset.* 1.2GHz *Quad-Core* ARM Cortex-A53 802.11 *b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)*
GPU	*Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES* 2.0, *hardware-accelerated OpenVG, and* 1080p30 H.264 *high-profile decode. Capable of* 1G piksel /s, 1.5Gtexel/s *or* 24GFLOPs *with texture filtering and* DMA *infrastructure*
Memory	1GB LPDDR2
Operating System	*Boots from Micro SD card, running a version of the* Linux *operating system or* Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A

Konektor ;

Ethernet	10/100 *BaseT Ethernet socket*
Video Output	HDMI *(rev* 1.3 & 1.4 *Composite* RCA (PAL and NTSC)
Audio Output	*Audio Output* 3.5mm *jack,* HDMI USB 4 x USB 2.0 *Connector*
GPIO Connector	40-pin 2.54 mm (100 mil) *expansion header:* 2x20 *strip* *Providing* 27 GPIO *pins as well as* +3.3 V, +5 V *and* GND *supply lines*

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Camera Connector	15-*pin* MIPI *Camera Serial Interface* (CSI-2*)*
Display Connector	*Display Serial Interface* (DSI) 15 *way flat flex cable connector with two data lanes and a clock lane*
Memory Card Slot	*Push/pull Micro* SDIO

2.8. Robot KRSBI Beroda Stikom Surabaya 2017

Robot KRSBI Beroda merupakan robot dari Tim SYNA-SWC8 yang ikut serta pada Kontes Robot Indonesia tahun 2017 lalu. Robot ini memiliki fitur *Image Processing* menggunakan Raspberry Pi yang disertai Raspicam, yaitu modul kamera yang disediakan khusus untuk Raspberry Pi dan Library OpenCV yang sudah diinstal didalamnya.

Robot dengan ukuran 52x52 cm memiliki fitur sistem gerak menggunakan roda Omni-Directional Wheels, yaitu suatu roda unik karena memiliki kemampuan bergerak bebas dua arah. Terpasang pada Motor DC yang dikendalikan oleh driver motor EMS (*Embedded Module Series*) 30 A H-Bridge melalui Arduino Due. Pada Robot ini juga terdapat mekanisme penendang dengan tenaga udara. Melalui sebuah Pneumatik yang dikendalikan dari Solenoid valve, yaitu katup yang dikendalikan dengan arus listrik baik AC maupun DC melalui kumparan/selenoida.

Gambar 2.18 Robot KRSBI Beroda 2017

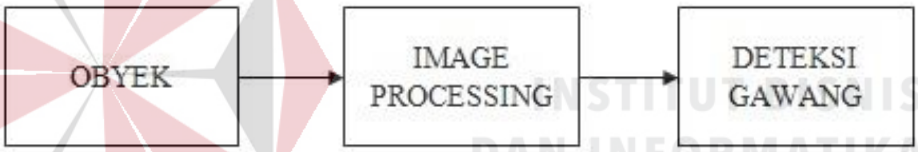
INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Metode penelitian yang digunakan dalam pengerjaan Tugas Akhir ini adalah studi literatur, pembuatan program serta melakukan deteksi, mengolah gambar dan *tracking* obyek. Pada penelitian Tugas Akhir ini, terdapat obyek berupa sebuah gawang berbentuk persegi atau persegi panjang yang hanya memiliki satu warna. Blok diagram penelitian ditunjukkan pada Gambar 3.1.



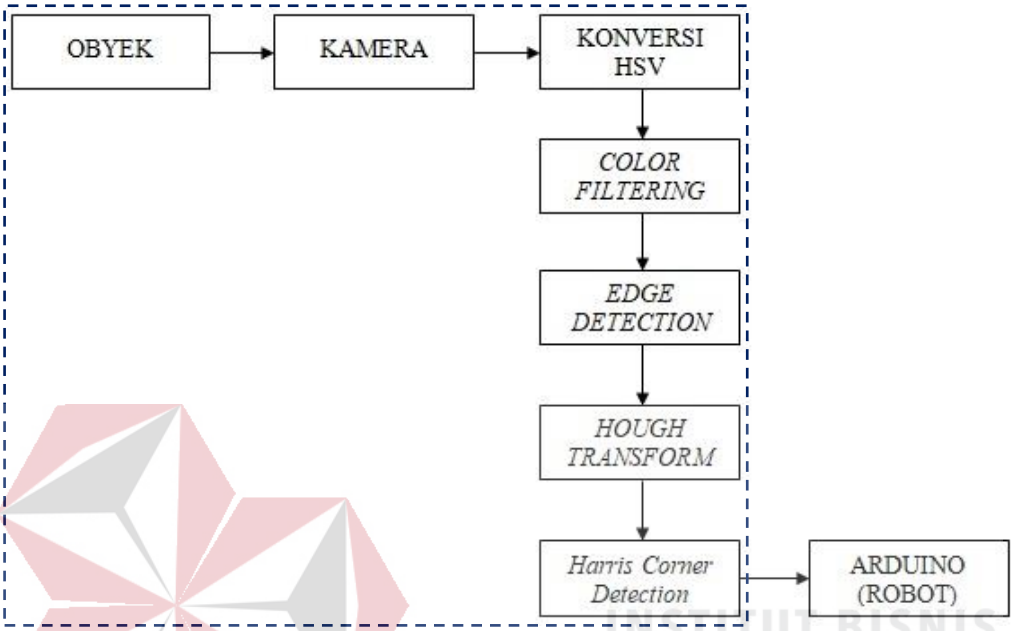
Gambar 3.1 Blok Diagram Penelitian

Penelitian diawali dengan mengambil input, yaitu dengan mencari obyek menggunakan kamera. Setelah obyek dapat dideteksi oleh kamera, obyek yang tampak pada gambar akan diolah dengan metode-metode tertentu. Langkah selanjutnya adalah melakukan *tracking* posisi keberada obyek terhadap robot. *Tracking* akan menghasilkan output berupa koordinat dari hasil *Image Processing* yang akan menentukan posisi robot akan bergerak.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

3.2. Pengolahan Obyek pada Rasberry Pi

Pengolahan obyek yang digunakan pada pengerjaan Tugas Akhir ini ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Proses Pengolahan pada *Image Processing*

Kamera akan menangkap gambar secara *real-time* yang berupa ruang warna RGB. Ruang warna yang masih berupa RGB akan dikonversi menjadi ruang warna HSV. Kemudian dilakukan *Color Filtering* untuk menyeleksi warna yang akan dideteksi. Gambar yang sudah difilter akan diubah menggunakan metode *Edge Detection*, yaitu menjadikan suatu gambar berupa garis tepi. Lalu setelah gambar berubah menjadi garis tepi, dilakukanlah metode *Hough Transform* khusus untuk mendeteksi garis. Setelah gambar diolah sesuai keinginan, data yang dihasilkan akan dikirim ke Arduino secara serial.

3.2.1. Konversi HSV

Pada tahap ini dilakukan konversi model warna pada citra atau frame, yang semula frame mempunyai model warna RGB (*Red, Green, Blue*) diubah ke model warna HSV (*Hue, Saturation, Value*). Piksel ruang warna RGB memiliki nilai yang terdiri dari R, G, dan B. Nilai-nilai ini akan digunakan untuk mencari nilai ruang warna HSV. Dengan persamaan (2.1) akan mengubah nilai R,G,B menjadi bentuk H,S,V.

Jika sebuah gambar memilki nilai RGB dengan nilai seperti berikut;

Tabel 3.1 Nilai RGB pada Sebuah Gambar

R	100	R	50	R	50	R	82
G	80	G	45	G	92	G	46
B	70	B	60	B	84	B	12
R	61	R	46	R	6	R	84
G	16	G	61	G	8	G	56
B	31	B	84	B	61	B	4
R	46	R	98	R	61	R	9
G	61	G	51	G	69	G	5
B	9	B	66	B	64	B	78

Dengan persamaan (2.1) akan menghasilkan nilai HSV seperti berikut;

Tabel 3.2 Nilai HSV pada Sebuah Gambar

H	-40	H	260	H	169	H	29.1
S	30	S	15	S	42	S	70
V	100	V	60	V	92	V	82
H	-20	H	216	H	238	H	39
S	45	S	38	S	55	S	80
V	61	V	84	V	61	V	84
H	77.3	H	-19	H	143	H	243
S	52	S	47	S	8	S	73
V	61	V	98	V	69	V	78

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

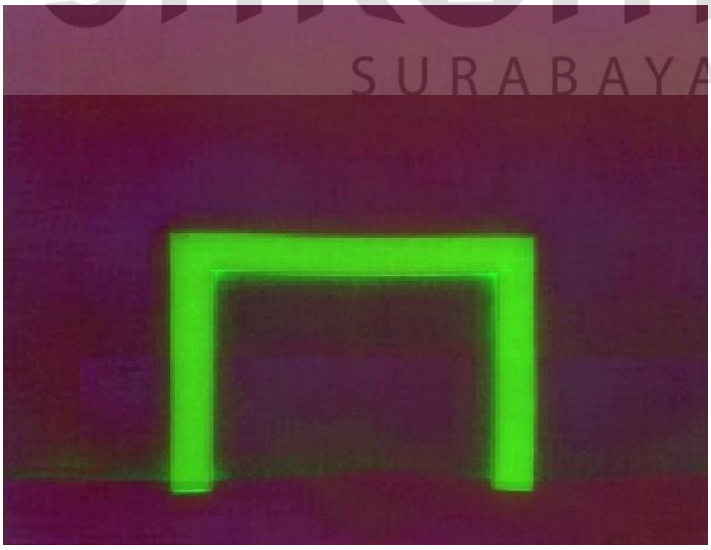
Pada program, proses pengkonversian model warna ini menggunakan bantuan dari *library* OpenCV. Berikut merupakan program untuk pengkonversian model warna RGB ke HSV:

```
cvCvtColor(imgRGB, imgHSV, CV_BGR2HSV);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

Hasil konversi ruang warna RGB ke HSV dapat dilihat pada Gambar 3.3 dan 3.4.

Gambar 3.3 Ruang Warna RGB



Gambar 3.419 Ruang Warna HSV

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

3.2.2. *Color Filtering*

Setelah proses pengkonversian pada model warna citra, selanjutnya dilakukan proses pencarian nilai minimum dan maksimum dari setiap elemen HSV yang mempresentasikan warna dari obyek. Pencarian nilai tersebut bertujuan untuk proses pemfilteran warna pada citra dengan warna sesuai yang diinginkan.

Pada proses filter ini, mengubah citra yang semula mempunyai 3 layer, yaitu HSV (*Hue, Saturation, Value*) menjadi citra biner atau hitam putih yang mempunyai 1 layer. Sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas. Nilai yang digunakan untuk *color filtering* adalah nilai minimum dan maksimum dari setiap elemen HSV. Nilai piksel yang mempunyai nilai diluar dari *range* nilai minimum dan maksimum tersebut, akan diubah nilainya menjadi sama dengan 0. Sedangkan nilai piksel yang berada pada *range* nilai minimum dan maksimum tersebut, nilai pikselnya diubah menjadi sama dengan 255. Dimana piksel yang mempunyai nilai 0, pikselnya berwarna hitam, sedangkan, piksel yang mempunyai nilai 255, pikselnya berwarna putih.

Tahap ini bertujuan untuk menyeleksi warna obyek dengan warna benda yang tertangkap oleh kamera. Dengan menggunakan metode *Color Filtering* jenis *Band Pass Filter*, dicarilah nilai-nilai batasan bawah dan batasan atas untuk sebuah warna pada obyek. Pada penelitian ini untuk mendeteksi warna dari obyek yaitu kuning, nilai dari filter untuk batas atas dan batas bawah adalah sebagai berikut:

H_MIN : 0

H_MAX : 40

S_MIN : 110

S_MAX : 255

V_MIN : 121

V_MAX : 255

Selain itu pada proses filter, dilakukan filter Gaussian. Filter Gaussian digunakan untuk menghilangkan *noise-noise* kecil yang ada disekitar obyek sehingga citra menjadi lebih *smooth*. Berikut merupakan program *color filtering* pada citra:

```
cvInRangeS(imgHSV, cvScalar(H_MIN, S_MIN, V_MIN),
cvScalar(H_MAX, S_MAX, V_MAX), imgFilter);
cvSmooth(imgFilter, imgFilter, CV_GAUSSIAN, 9, 9);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

Hasil filter warna dapat dilihat pada Gambar 3.5

Gambar 3.5 Hasil *Color Filtering*

3.2.3. *Edge Detection*

Untuk mendeteksi garis, obyek harus berupa garis. Pada proses *color filtering*, obyek yang dideteksi masih berupa suatu blob/gambar padat. Untuk itu pada langkah selanjutnya adalah merubah blob tersebut menjadi gambar garis tepi. Gambar hasil *Color Filtering* akan diolah lagi menggunakan metode *Edge*

Detection, sehingga gambar akan berubah menjadi garis tepi. Berikut merupakan program *Edge Detection* pada citra:

```
IplImage* imgEdge = cvCreateImage(cvGetSize(imgFilter), IPL_DEPTH_8U, 0);
cvCanny(imgFilter, imgEdge, 50, 200, 3);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

Hasil pengolahan gambar menjadi garis tepi dapat dilihat pada Gambar 3.6.

Gambar 3.6 Hasil Pengolahan *Edge Detection*

3.2.4. *Hough Transform*

Setelah gambar berubah mejadi garis tepi, maka akan memudahkan proses yang selanjutnya yaitu deteksi garis. Dengan mengunakan persamaan-persamaan yang ada pada proses *Hough Transform*, gambar yang berupa garis tepi akan diolah sesuai koordinat dari garis tepi tersebut.

Sehingga dengan metode *Hough Transform*, gambar hasil pengolahan *Edge Detection* yang berupa garis tepi akan dideteksi garis-garis tersebut. Berikut merupakan program *Hough Transform* pada citra:

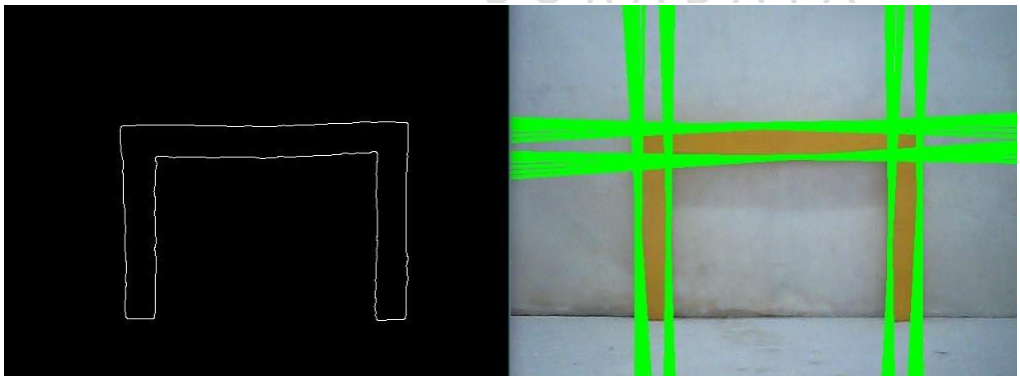
```
CvMemStorage* Storage = cvCreateMemStorage(0);
```

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

```
CvSeq *lines = cvHoughLines2(imgEdge, Storage,
CV_HOUGH_STANDARD, 1, CV_PI / 180, 50, 0, 0);
for (int i = 0; i < MIN(lines->total, 100); i++)
{
    float *line = (float *)cvGetSeqElem(lines, i), rho = line[0], theta =
    line[1];
    double a = cos(theta), b = sin(theta), x0 = a * rho, y0 = b * rho;
    CvPoint pt1, pt2;
     pt1.x = cvRound(x0 + 1000 * (-b));
     pt1.y = cvRound(y0 + 1000 * (a));
     pt2.x = cvRound(x0 - 1000 * (-b));
     pt2.y = cvRound(y0 - 1000 * (a));
    cvLine(imgRGB, pt1, pt2, CV_RGB(0, 255, 0), 3, 8, 0);
}
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

Hasil deteksi garis dapat dilihat pada Gambar 3.7.



Gambar 3.7 Hasil Deteksi Tepi Garis

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

3.2.5. Harris *Corner Detection*

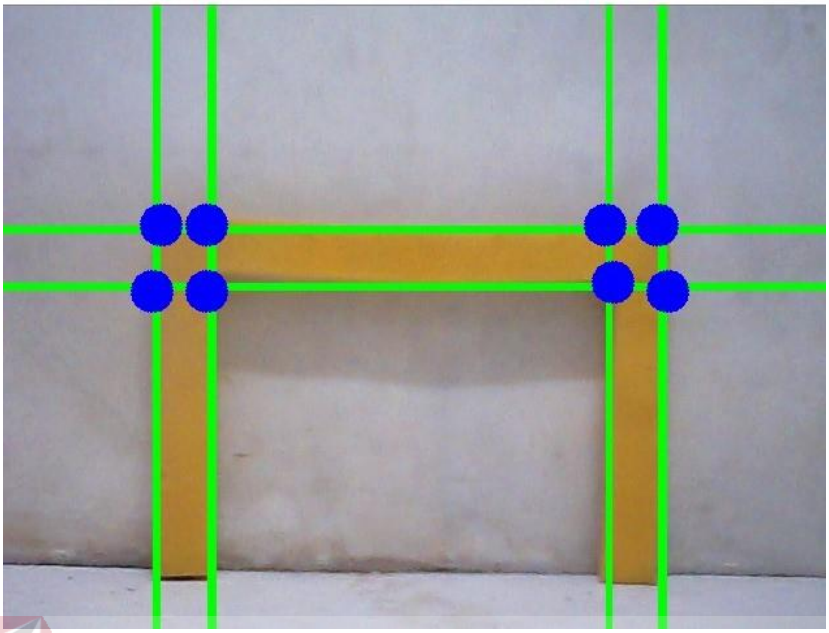
Obyek yang sudah diolah dengan *Hough Transform* akan menghasilkan garis vertikal dan horizontal. Pada pertemuan kedua garis ini akan membentuk sebuah sudut. Sudut inilah yang akan dideteksi menggunakan Harris *Corner Detection.* Berikut merupakan program *Hough Transform* pada citra:

```
cvNormalize(imgEdge, imgEdge, 0, 255, NORM_MINMAX);
int corner_count = 8;
CvPoint2D32f *corners = (CvPoint2D32f *)cvAlloc(corner_count * sizeof(CvPoint2D32f));
cvGoodFeaturesToTrack(imgHough, eig_img, temp_img, corners, &corner_count, 0.1, 15, NULL, 3, 1, 0.01);
cvFindCornerSubPix(imgHough, corners, corner_count, cvSize(3, 3), cvSize(-1, -1), cvTermCriteria(CV_TERMCRIT_ITER | CV_TERMCRIT_EPS, 20, 0.03));
int i;
for (i = 0; i < corner_count; i++)
cvCircle(imgRGB, cvPointFrom32f(corners[i]), 3, CV_RGB(0, 0, 255), 25);
```

Program lengkap secara keseluruhan dapat dilihat pada Lampiran 1.

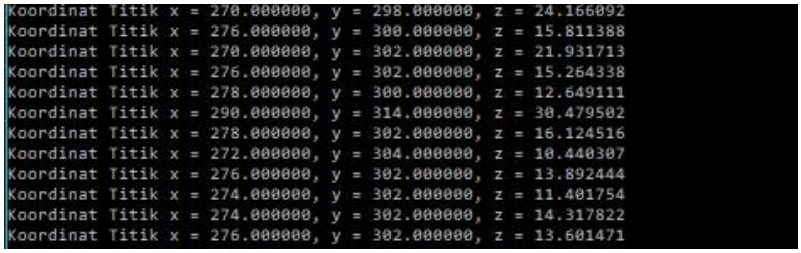
Adapun hasil dari Harris *Corner Detection* dapat dilihat pada Gambar 3.8.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA



Gambar 3.8 Hasil Harris *Corner Detection*

Titik-titik yang terlihat pada Gambar 3.8 merupakan titik-titik hasil dari deteksi sudut pada gawang. Titik-titik tersebut berada pada piksel-piksel tertentu dengan koordinat x, y, dan z. Nilai dari koordinat ini akan ditampilkan dan akan menentukan proses selanjutnya sesuai dengan data yang dikirim pada mikrokontroler. Hasil koordinat x, y, dan z ditampilkan pada terminal komputer seperti Gambar 3.9.



Gambar 3.9 Hasil Tampilan Koordinat

3.3. Flowchart Program

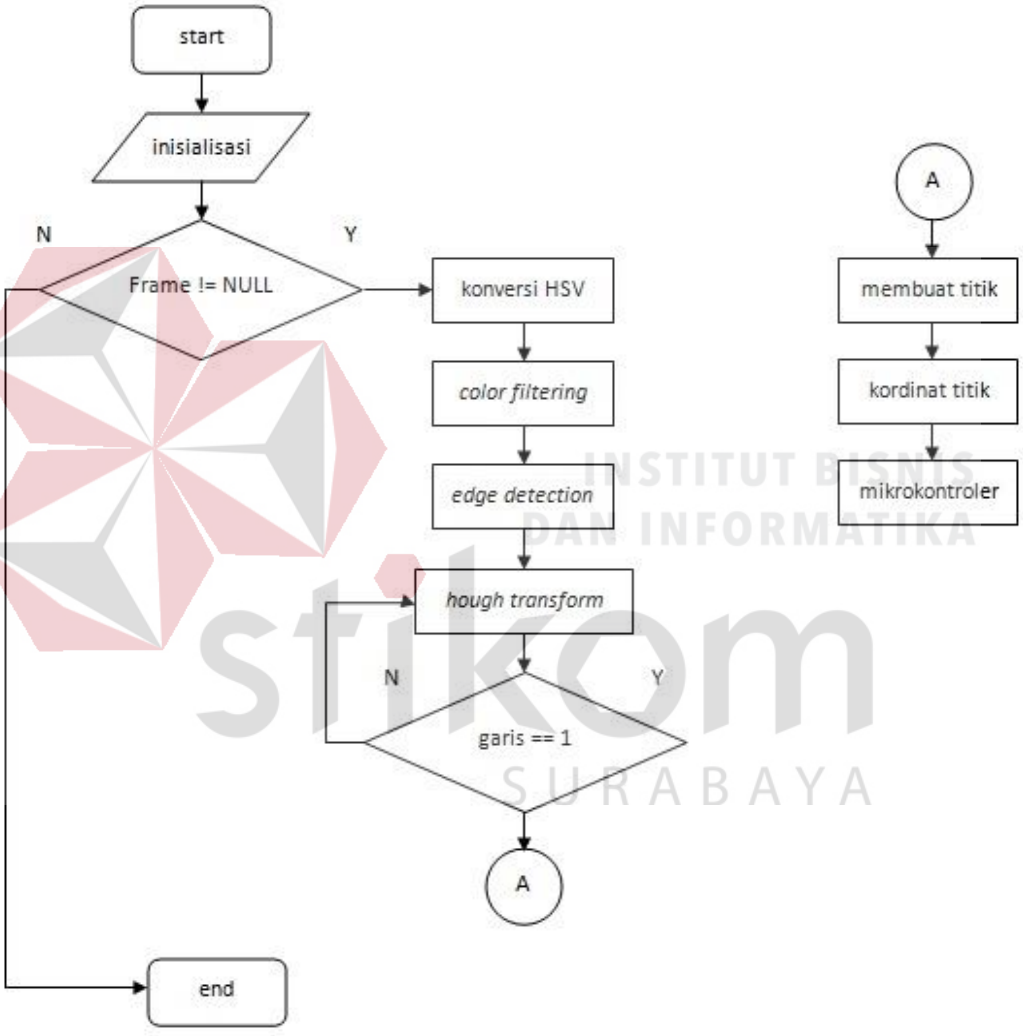
Pada penelitian ini mempunyai flowchart program seperti yang ditunjukkan pada Gambar 3.10.

Penjelasan dari Flowchart pada Gambar 3.10 adalah sebagai berikut:

1. Inisialisasi merupakan proses deklarasi variabel-variabel yang digunakan di dalam program. Pada penelitian ini variabel-variabel seperti variabel 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.
4. *Color Filtering* merupakan proses penyeleksian warna obyek. Proses ini bertujuan untuk mencari nilai filter minimal dan maksimal dari model warna HSV pada frame. Sehingga dapat diketahui bagian mana yang termasuk obyek dan mana yang bukan obyek.
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.
6. *Hough Transform* untuk mendeteksi garis pada gambar yang sudah diolah sebelumnya menggunakan metode *Edge Detection*.
7. Ketika obyek terdeteksi, maka nilai dari lingkaran menjadi 1.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

8. Ketika nilai garis sama dengan 1, maka program akan melanjutkan ke perintah berikutnya.
9. Proses selanjutnya yaitu membuat titik dari sudut pada garis yang terdeteksi.
10. Setelah titik dibuat, dapat diketahui koordinat dari titik tersebut. Koordinat yang keluar berupa nilai koordinat x, y, dan z.



Gambar 3.10 Flowchart Deteksi Gawang

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

BAB IV

HASIL PENGUJIAN DAN PENGAMATAN

4.1. Pengujian Program *Color Filtering*

Pengujian *Color Filtering* untuk melakukan seleksi warna obyek dengan warna benda yang ada disekitarnya.

4.1.1. Tujuan Pengujian Program *Color Filtering*

Tujuan dari pengujian ini untuk mengetahui proses deteksi warna pada obyek bekerja dengan baik.

4.1.2. Alat yang Digunakan pada Pengujian Program *Color Filtering*

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. Raspberry Pi
2. Raspicam
3. Program OpenCV
4. Laptop
5. Kabel LAN
6. *Power Supply* 1000mA – 5V.

4.1.3. Prosedur Pengujian Program *Color Filtering*

Langkah-langkah untuk melakukan pengujian program deteksi warna adalah sebagai berikut:

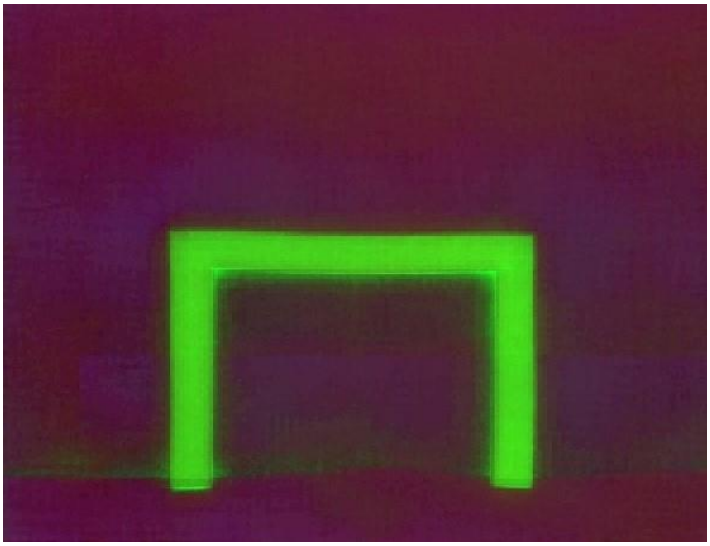
1. Menghubungkan Raspicam pada Raspberry Pi

2. Menghubungkan *power supply* dengan Raspberry Pi.
3. Menghubungkan kabel LAN antara Laptop dan Raspberry Pi
4. Laptop sudah terinstal *software* Remote Dekstop dan IP Scanner.
5. Menggunakan IP Scanner untuk mencari alamat IP pada Raspberry Pi.
6. Menjalankan Remote Dekstop pada Laptop untuk mengoperasikan Raspberry Pi dengan IP yang sudah didapatkan.
7. Menjalankan program OpenCV pada terminal Raspberry Pi.
8. Memposisikan obyek pada jangkauan tangkapan kamera.
9. Melakukan kalibrasi warna terhadap warna obyek dengan menggunakan *Trackbar* pada program.

4.1.4. Hasil Pengujian Program *Color Filtering*

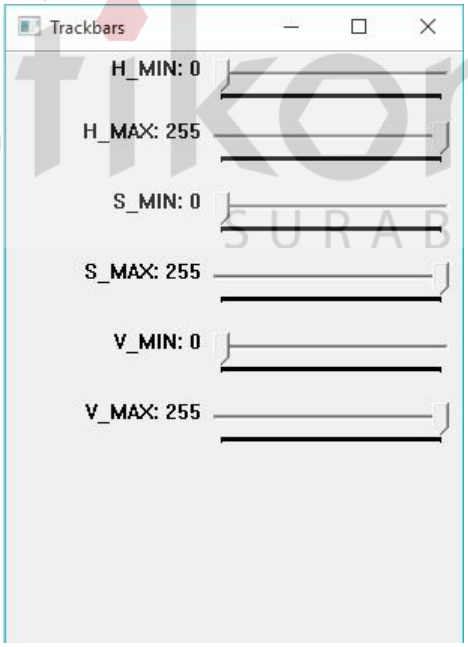
Inilah awal mula tampilan atau hasil frame sebelum proses dijalankan/ ruang warna RGB, ditunjukkan pada Gambar 4.1. Sebelum proses *Color Filtering* dilakukan, gambar akan dikonversikan terlebih dahulu ke dalam ruang warna HSV, ditunjukkan pada Gambar 4.2.

Gambar 4.1 Hasil Sebelum Konversi



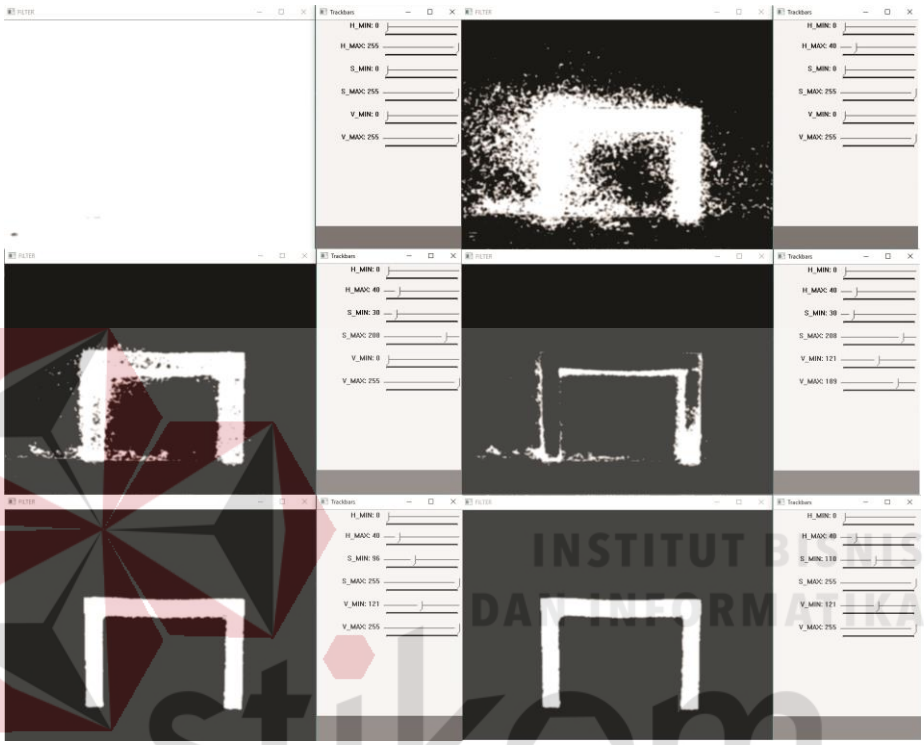
Gambar 4.2 Hasil Setelah Konversi

Setelah gambar dikonversikan ke warna HSV, gambar akan difilter. Cara pemfilteran dapat dilakukan pada *Trackbar* dengan mengganti dengan nilai yang sesuai. Tampilan *Trackbar* pada program ditunjukkan pada Gambar 4.3.



Gambar 4.3 Trackbar Sebelum Filter

Dilakukan kalibrasi untuk mendapatkan hasil yang bagus, sehingga membutuhkan percobaan-percobaan untuk mendapatkan hasil yang bagus. Beberapa percobaan dapat dilihat pada Gambar 4.4 dari awal kalibrasi hingga mendapatkan hasil yang bagus.



Gambar 4.4 Proses Pemfilteran

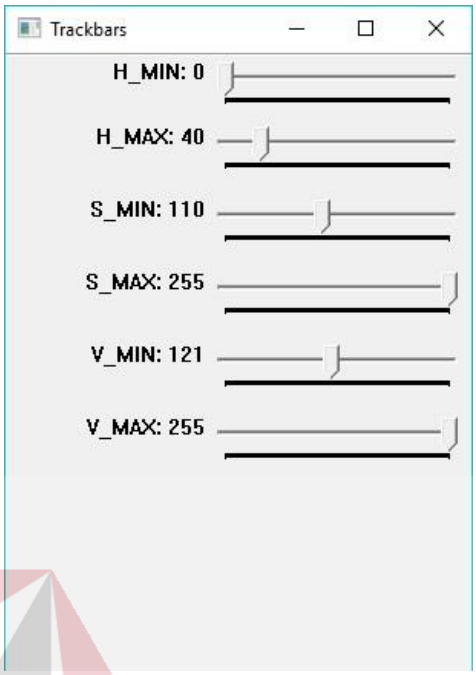
INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Nilai dari H_MIN, H_MAX, S_MIN, S_MAX, V_MIN, V_MAX dari tiap percobaan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil dari Tiap Percobaan Color Filtering

Percobaan ke-	H_MIN	H_MAX	S_MIN	S_MAX	V_MIN	V_MAX	**Keterangan**
1	0	255	0	255	0	255	Obyek Tidak Terlihat
2	0	40	0	255	0	255	Obyek Tidak Terlihat
3	0	40	30	208	0	255	Obyek Tidak Terlihat
4	0	40	30	208	121	189	Obyek Tidak Terlihat
5	0	40	96	255	121	255	Obyek Terlihat
6	0	40	110	255	121	255	Obyek Terlihat

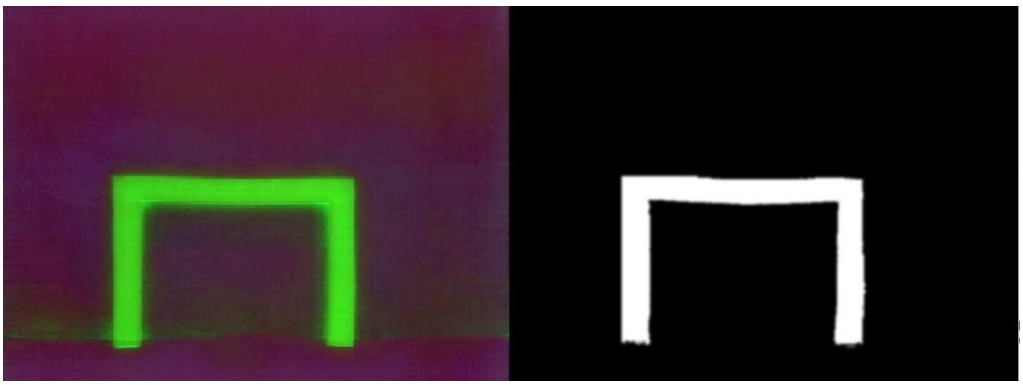
Tampilan nilai akhir pada *Trackbars* untuk *Color Filtering* ditunjukkan pada Gambar 4.5.



Gambar 4.5 Hasil Setelah Filter

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Setelah *Color Filtering* dilakukan maka akan terlihat jelas perbedaan pada frame antara frame sebelum proses dan frame setelah proses. Hasil ditunjukkan pada Gambar 4.6.



Gambar 4.6 Hasil Pemfilteran

Pada Gambar 4.6 dapat dilihat bahwa sistem mampu mendeteksi warna dari obyek yang telah ditentukan. Dengan proses filter, warna yang lolos seleksi filter berupa warna putih sedangkan yang tidak lolos berupa warna hitam.

4.2. Pengujian Program *Edge Detection*

Pengujian *Edge Detection* untuk melakukan pengubahan gambar blob dari hasil *Color Filtering* menjadi garis tepi.

4.2.1. Tujuan Pengujian Program *Edge Detection*

Tujuan dari pengujian ini untuk mengetahui proses pengubahan menjadi garis tepi pada obyek bekerja dengan baik.

4.2.2. Alat yang Digunakan pada Pengujian Program *Edge Detection*

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. Raspberry Pi
2. Raspicam
3. Program OpenCV
4. Laptop
5. Kabel LAN
6. *Power Supply* 1000mA – 5V.

4.2.3. Prosedur Pengujian Program *Edge Detection*

Langkah-langkah untuk melakukan pengujian program deteksi warna adalah sebagai berikut:

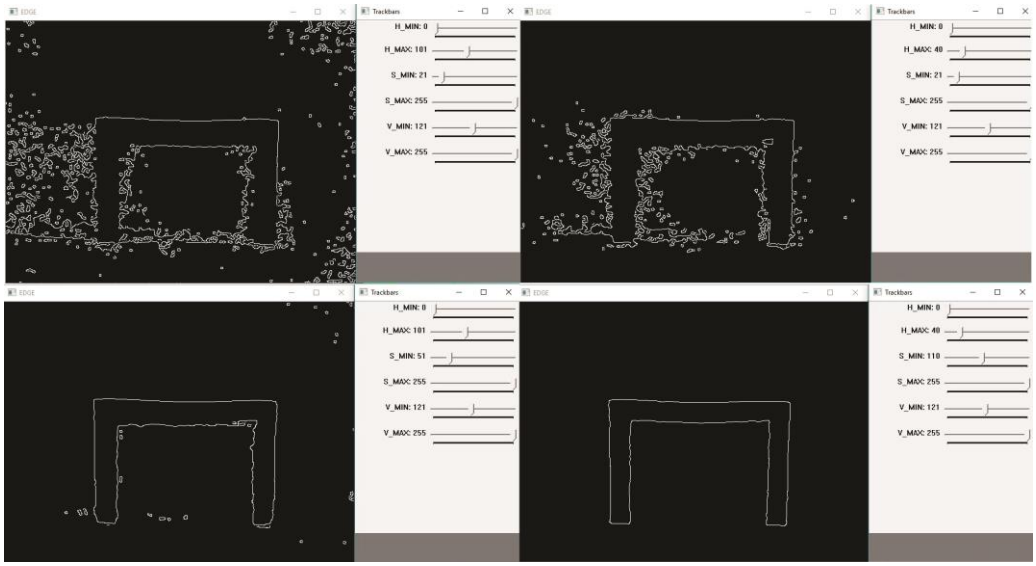
INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

1. Menghubungkan Raspicam pada Raspberry Pi
2. Menghubungkan *power supply* dengan Raspberry Pi.
3. Menghubungkan kabel LAN antara Laptop dan Raspberry Pi
4. Laptop sudah terinstal *software* Remote Dekstop dan IP Scanner.
5. Menggunakan IP Scanner untuk mencari alamat IP pada Raspberry Pi.
6. Menjalankan Remote Dekstop pada Laptop untuk mengoperasikan Raspberry Pi dengan IP yang sudah didapatkan.
7. Menjalankan program OpenCV pada terminal Raspberry Pi.
8. Memposisikan obyek pada jangkauan tangkapan kamera.
9. Mengamati hasil perubahan blob ke garis tepi, apabila belum mendapatkan hasil yang baik perbaiki hasil *Color Filtering.*

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.2.4. Hasil Pengujian Program *Edge Detection*

Setelah melewati proses pemfilteran, gambar akan diubah menjadi garis tepi. *Edge Detection* hanya mengubah gambar yang sudah difilter. Apabila hasil gambar belum baik, maka perlu melakukan lagi pemfilteran. Beberapa percobaan dapat dilihat pada Gambar 4.7.



Gambar 4.7 Proses Deteksi Tepi

Pada Gambar 4.8 dapat dilihat hasil *Edge Detection* dari hasil gambar pemfilteran.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Gambar 4.8 Hasil Deteksi Tepi

4.3. Pengujian Program *Hough Transform*

Pengujian *Hough Transform* untuk melakukan deteksi garis pada gambar hasil yang sudah diolah dengan metode *Edge Detection* menjadi garis tepi.

4.3.1. Tujuan Pengujian Program *Hough Transform*

Tujuan dari pengujian ini untuk mendeteksi garis pada gambar dengan baik.

4.3.2. Alat yang Digunakan pada Pengujian Program *Hough Transform*

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. Raspberry Pi
2. Raspicam
3. Program OpenCV
4. Laptop
5. Kabel LAN
6. *Power Supply* 1000mA – 5V.

4.3.3. Prosedur Pengujian Program *Hough Transform*

Langkah-langkah untuk melakukan pengujian program deteksi warna adalah sebagai berikut:

1. Menghubungkan Raspicam pada Raspberry Pi
2. Menghubungkan *power supply* dengan Raspberry Pi.
3. Menghubungkan kabel LAN antara Laptop dan Raspberry Pi
4. Laptop sudah terinstal *software* Remote Dekstop dan IP Scanner.
5. Menggunakan IP Scanner untuk mencari alamat IP pada Raspberry Pi.
6. Menjalankan Remote Dekstop pada Laptop untuk mengoperasikan Raspberry Pi dengan IP yang sudah didapatkan.
7. Menjalankan program OpenCV pada terminal Raspberry Pi.
8. Memposisikan obyek pada jangkauan tangkapan kamera.
9. Mengamati hasil deteksi garis pada obyek.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.3.4. Hasil Pengujian Program *Hough Transform*

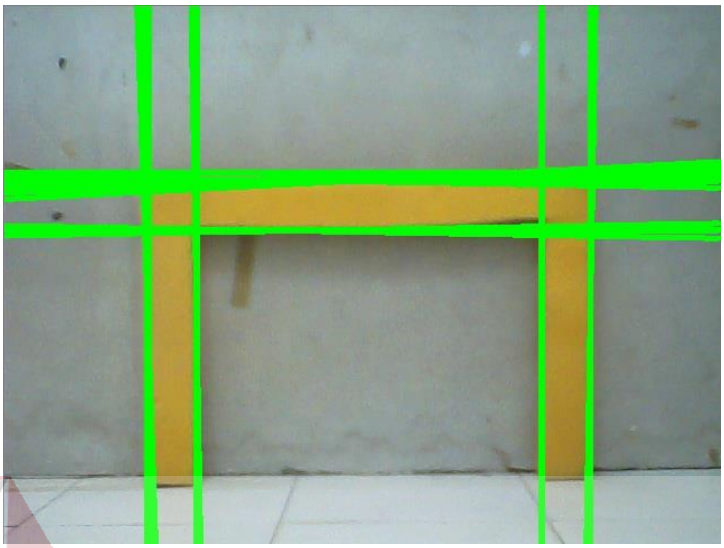
Pada pengujian program *Hough Transform* menggunakan parameter normal untuk garis lurus berupa persamaan, dengan sebagai jarak suatu garis terhadap titik asal dan sebagai orientasi garis tersebut terhadap sumbu. Setiap titik pada citra ditransformasikan menjadi suatu kurva sinusiodal pada ruang. Akumulator mencatat semua perpotongan kurva dan mengambil perpotongan terbanyak.

Adapun hasil dari *Hough Transform* dengan nilai resolusi jarak antar piksel adalah 1 yaitu pada Gambar 4.9.

Gambar 4.9 Nilai Resolusi Piksel Adalah 1

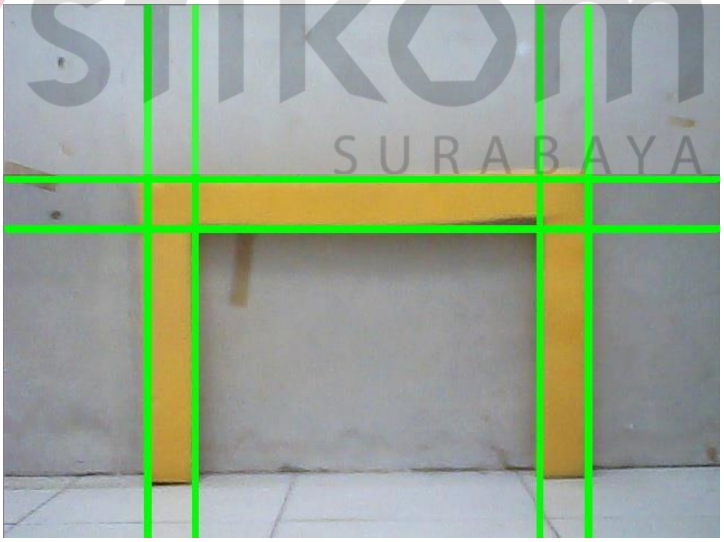
INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Hasil dari *Hough Transform* dengan nilai resolusi jarak antar piksel adalah 0,5 yaitu pada Gambar 4.10.



Gambar 4.10 Nilai Resolusi Piksel Adalah 0,5

Hasil dari *Hough Transform* dengan nilai resolusi jarak antar piksel adalah 0,1 yaitu pada Gambar 4.11.



Gambar 4.11 Nilai Resolusi Piksel Adalah 0,1

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.4. Pengujian Program Harris *Corner Detection*

Pengujian Harris *Corner Detection* untuk membuat titik pada sudut obyek dengan baik.

4.4.1. Tujuan Pengujian Program Harris *Corner Detection*

Tujuan dari pengujian ini untuk membuat titik pada sudut dari gambar yang sudah diolah menggunakan *Hough Transform*.

4.4.2. Alat yang Digunakan pada Pengujian Program Harris *Corner Detection*

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. Raspberry Pi
2. Raspicam
3. Program OpenCV
4. Laptop
5. Kabel LAN
6. *Power Supply* 1000mA – 5V.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.4.3. Prosedur Pengujian Program Harris *Corner Detection*

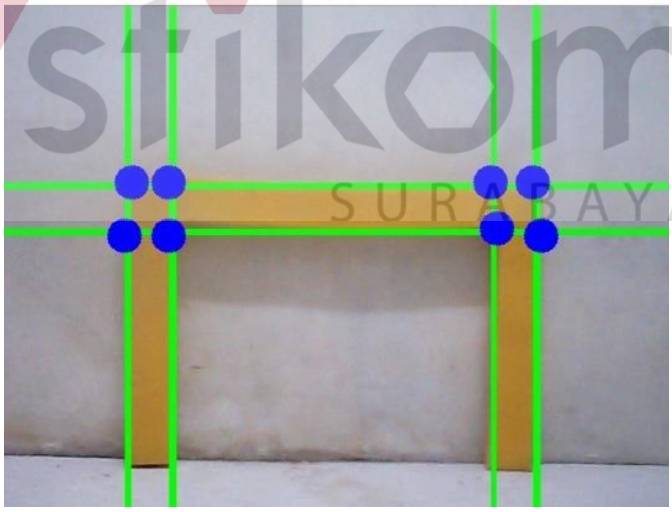
Langkah-langkah untuk melakukan pengujian titik pada sudut adalah sebagai berikut:

1. Menghubungkan Raspicam pada Raspberry Pi
2. Menghubungkan *power supply* dengan Raspberry Pi.
3. Menghubungkan kabel LAN antara Laptop dan Raspberry Pi

4. Laptop sudah terinstal *software* Remote Dekstop dan IP Scanner.
5. Menggunakan IP Scanner untuk mencari alamat IP pada Raspberry Pi.
6. Menjalankan Remote Dekstop pada Laptop untuk mengoperasikan Raspberry Pi dengan IP yang sudah didapatkan.
7. Menjalankan program OpenCV pada terminal Raspberry Pi.
8. Memposisikan obyek pada jangkauan tangkapan kamera.
9. Melakukan deteksi sudut pada obyek hasil pengolahan dari gambar *Hough Transform*.

4.4.4. Hasil Pengujian Program Harris *Corner Detection*

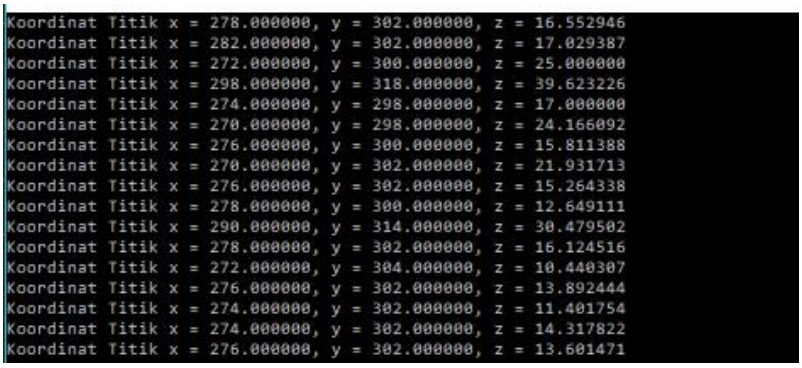
Harris *Corner Detection* akan mendeteksi sudut dan membuat titik pada sudut hasil pengolahan garis pada *Hough Transform*. Tampilan hasil Harris *Corner Detection* dapat dilihat pada Gambar 4.12.



Gambar 4.12 Hasil Deteksi Sudut

Pada Gambar 4.12 terdapat titik-titik hasil dari deteksi sudut pada gawang. Titik-titik tersebut berada pada piksel- piksel tertentu dengan koordinat x, y, dan

z. Nilai dari koordinat ini akan ditampilkan dan akan menentukan proses selanjutnya sesuai dengan data yang dikirim pada mikrokontroler. Ditunjukkan pada Gambar 4.13.



Gambar 4.13 Nilai Koordinat

4.5. Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot

Pengujian deteksi gawang pada robot dengan penempatan pada jarak dan posisi yang bervariasi.

4.5.1. Tujuan Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot

Tujuan dari pengujian ini untuk mengetahui jarak dan posisi dari beberapa variasi jarak dan posisi robot dapat mendeteksi gawang dengan baik.

4.5.2. Alat yang Digunakan pada Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. Program OpenCV
2. Laptop
3. Kabel LAN
4. Robot KRSBI Beroda

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.5.3. Prosedur Pengujian Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot

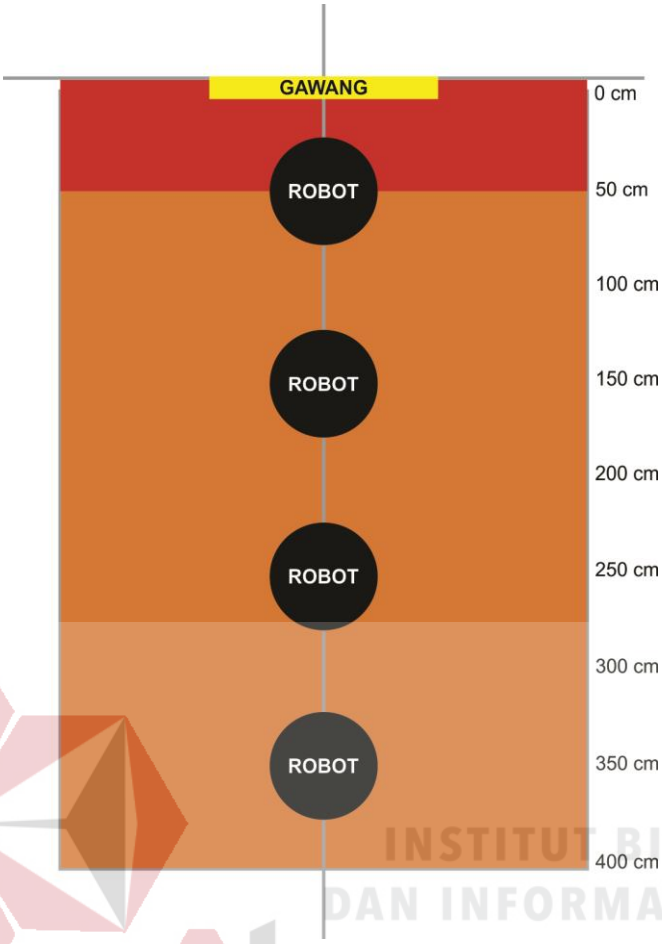
Langkah-langkah untuk melakukan deteksi gawang adalah sebagai berikut:

1. Menghubungkan kabel LAN antara Laptop dan Raspberry Pi yang terdapat pada Robot KRSBI Beroda.
2. Laptop sudah terinstal *software* Remote Dekstop dan IP Scanner.
3. Menggunakan IP Scanner untuk mencari alamat IP pada Raspberry Pi.
4. Menjalankan Remote Dekstop pada Laptop untuk mengoperasikan Raspberry Pi dengan IP yang sudah didapatkan.
5. Menjalankan program OpenCV pada terminal Raspberry Pi.
6. Memposisikan robot dengan variasi jarak dan posisi yang berbeda untuk mengetahui jarak dan posisi mana robot dapat mendeteksi dan tidak dapat mendeteksi gawang.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

4.5.4. Hasil Pengujian Program Deteksi Gawang dengan Variasi Jarak dan Posisi dari Robot

Pengujian deteksi gawang dilakukan pada variasi jarak tertentu sehingga dapat diketahui jarak minimal dan jarak maksimal mendeteksi sudut.



Gambar 4.14 Variasi Jarak Gawang dan Robot

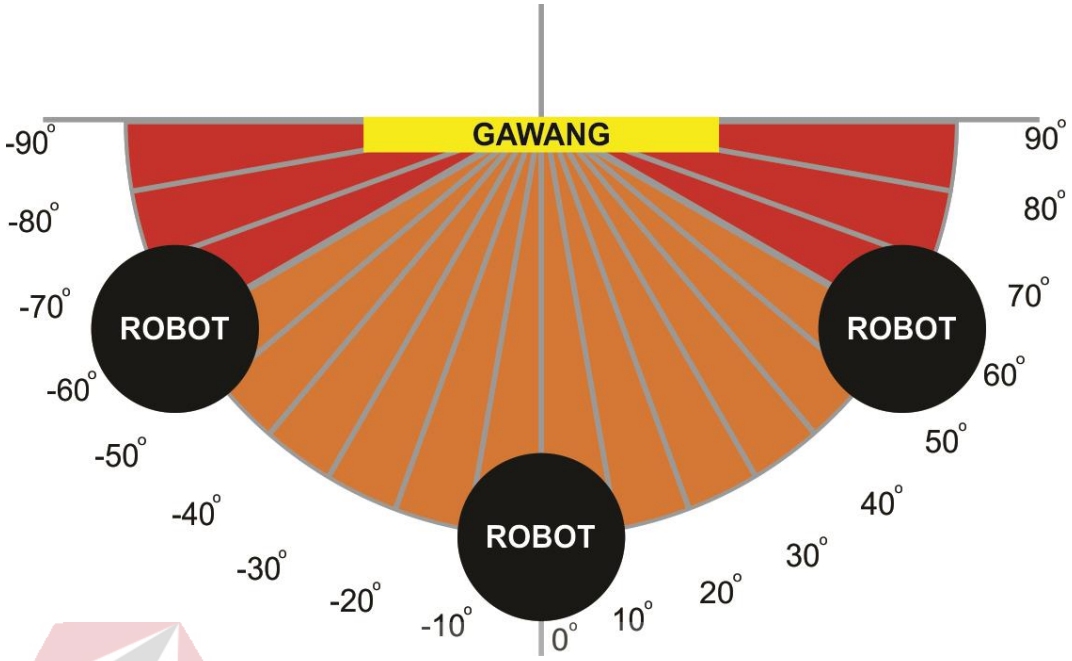
Adapun hasil pengujian deteksi sudut dengan variasi jarak dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Deteksi Gawang dengan Variasi Jarak

Percobaan ke-	Jarak	Keterangan
1	10 cm	Tidak mendeteksi gawang
2	20 cm	Tidak mendeteksi gawang
3	30 cm	Tidak mendeteksi gawang
4	40 cm	Tidak mendeteksi gawang
5	50 cm	Tidak mendeteksi gawang
6	60 cm	Mendeteksi gawang
7	70 cm	Mendeteksi gawang
8	80 cm	Mendeteksi gawang
9	90 cm	Mendeteksi gawang
10	100 cm	Mendeteksi gawang
11	110 cm	Mendeteksi gawang
12	120 cm	Mendeteksi gawang

Percobaan ke-	Jarak	Keterangan
13	130 cm	Mendeteksi gawang
14	140 cm	Mendeteksi gawang
15	150 cm	Mendeteksi gawang
16	160 cm	Mendeteksi gawang
17	170 cm	Mendeteksi gawang
18	180 cm	Mendeteksi gawang
19	190 cm	Mendeteksi gawang
20	200 cm	Mendeteksi gawang
21	210 cm	Mendeteksi gawang
22	220 cm	Mendeteksi gawang
23	230 cm	Mendeteksi gawang
24	240 cm	Mendeteksi gawang
25	250 cm	Mendeteksi gawang
26	260 cm	Mendeteksi gawang
27	270 cm	Mendeteksi gawang
28	280 cm	Mendeteksi gawang
29	290 cm	Mendeteksi gawang
30	300 cm	Mendeteksi gawang
31	310 cm	Mendeteksi gawang
32	320 cm	Mendeteksi gawang
33	330 cm	Mendeteksi gawang
34	340 cm	Mendeteksi gawang
35	350 cm	Mendeteksi gawang
36	360 cm	Mendeteksi gawang
37	370 cm	Mendeteksi gawang
38	380 cm	Mendeteksi gawang
39	390 cm	Mendeteksi gawang
40	400 cm	Mendeteksi gawang

Selain itu dilakukan pula pengujian deteksi gawang dengan variasi posisi dan jarak untuk mengetahui posisi-posisi robot dapat mendeteksi gawang atau tidak.



Gambar 4.15 Variasi Posisi Gawang dan Robot

Adapun hasil pengujian deteksi gawang dengan variasi posisi dan jarak dapat dilihat pada Tabel 4.3, Tabel 4.4, Tabel 4.5, dan Tabel 4.6.

Tabel 4.3 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 100 cm

Percobaan ke-	Sudut	Keterangan
1	- 90 º	Tidak mendeteksi gawang
2	- 80 º	Tidak mendeteksi gawang
3	- 70 º	Tidak mendeteksi gawang
4	- 60 º	Tidak mendeteksi gawang
5	- 50 º	Mendeteksi gawang
6	- 40 º	Mendeteksi gawang
7	- 30 º	Mendeteksi gawang
8	- 20 º	Mendeteksi gawang
9	- 10 º	Mendeteksi gawang
10	0 º	Mendeteksi gawang
11	10 º	Mendeteksi gawang
12	20 º	Mendeteksi gawang
13	30 º	Mendeteksi gawang
14	40 º	Mendeteksi gawang
15	50 º	Mendeteksi gawang
16	60 º	Tidak mendeteksi gawang
17	70 º	Tidak mendeteksi gawang

Percobaan ke-	Sudut	Keterangan
18	80 º	Tidak mendeteksi gawang
19	90 º	Tidak mendeteksi gawang

Tabel 4.4 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 200 cm

Percobaan ke-	Sudut	Keterangan
1	- 90 º	Tidak mendeteksi gawang
2	- 80 º	Tidak mendeteksi gawang
3	- 70 º	Tidak mendeteksi gawang
4	- 60 º	Tidak mendeteksi gawang
5	- 50 º	Mendeteksi gawang
6	- 40 º	Mendeteksi gawang
7	- 30 º	Mendeteksi gawang
8	- 20 º	Mendeteksi gawang
9	- 10 º	Mendeteksi gawang
10	0 º	Mendeteksi gawang
11	10 º	Mendeteksi gawang
12	20 º	Mendeteksi gawang
13	30 º	Mendeteksi gawang
14	40 º	Mendeteksi gawang
15	50 º	Mendeteksi gawang
16	60 º	Tidak mendeteksi gawang
17	70 º	Tidak mendeteksi gawang
18	80 º	Tidak mendeteksi gawang
19	90 º	Tidak mendeteksi gawang

Tabel 4.5 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 300 cm

Percobaan ke-	Sudut	Keterangan
1	- 90 º	Tidak mendeteksi gawang
2	- 80 º	Tidak mendeteksi gawang
3	- 70 º	Tidak mendeteksi gawang
4	- 60 º	Tidak mendeteksi gawang
5	- 50 º	Mendeteksi gawang
6	- 40 º	Mendeteksi gawang
7	- 30 º	Mendeteksi gawang
8	- 20 º	Mendeteksi gawang
9	- 10 º	Mendeteksi gawang
10	0 º	Mendeteksi gawang
11	10 º	Mendeteksi gawang
12	20 º	Mendeteksi gawang
13	30 º	Mendeteksi gawang
14	40 º	Mendeteksi gawang
15	50 º	Mendeteksi gawang
16	60 º	Tidak mendeteksi gawang

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

Percobaan ke-	Sudut	Keterangan
17	70 º	Tidak mendeteksi gawang
18	80 º	Tidak mendeteksi gawang
19	90 º	Tidak mendeteksi gawang

Tabel 4.6 Hasil Pengujian Deteksi Gawang dengan Variasi Posisi dan Jarak 400 cm

Percobaan ke-	Sudut	Keterangan
1	- 90 º	Tidak mendeteksi gawang
2	- 80 º	Tidak mendeteksi gawang
3	- 70 º	Tidak mendeteksi gawang
4	- 60 º	Tidak mendeteksi gawang
5	- 50 º	Mendeteksi gawang
6	- 40 º	Mendeteksi gawang
7	- 30 º	Mendeteksi gawang
8	- 20 º	Mendeteksi gawang
9	- 10 º	Mendeteksi gawang
10	0 º	Mendeteksi gawang
11	10 º	Mendeteksi gawang
12	20 º	Mendeteksi gawang
13	30 º	Mendeteksi gawang
14	40 º	Mendeteksi gawang
15	50 º	Mendeteksi gawang
16	60 º	Tidak mendeteksi gawang
17	70 º	Tidak mendeteksi gawang
18	80 º	Tidak mendeteksi gawang
19	90 º	Tidak mendeteksi gawang

Pada Gambar 4.14 dan Gambar 4.15, daerah warna merah adalah daerah robot tidak dapat mendeteksi sudut sedangkan pada daerah warna orange robot dapat mendeteksi sudut. Adapun nilai dari pengujian deteksi sudut berdasarkan variasi jarak dan posisi dapat dilihat pada Tabel 4.2. untuk variasi jarak dan Tabel 4.3, Tabel 4.4, Tabel 4.5, dan Tabel 4.6.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengujian diperoleh hasil sebagai berikut :

1. Tahapan awal pendeteksian gawang adalah pengkonversian warna dari RGB ke HSV.
2. Proses *Color Filtering* dapat menghasilkan filter yang sangat bagus apabila pencahayaan stabil. Karena apabila pencahayaan berubah-ubah, dapat mengganggu proses *Color Filtering*.
3. Proses pendeteksian gawang setelah *Color Filtering* adalah *Edge Detection* yang merupakan metode untuk mendeteksi garis tepi gawang.
4. *Hough Transform* mendeteksi garis hasil pengolahan gambar dengan metode *Edge Detection.*
5. *Harris Corner Detection* akan mendeteksi sudut dan membuat titik pada sudut hasil pengolahan garis pada *Hough Transform*.
6. Hasil pengujian jarak pada deteksi gawang dengan hasil visual yang baik adalah pada jarak di atas 50 cm. Sedangkan pada jarak di bawah 50 cm, pendeteksian gawang kurang baik.
7. Hasil pengujian jarak dan posisi pada deteksi gawang dengan hasil visual yang baik adalah pada saat robot berada pada posisi antara -50^{o} sampai 50^{o}. Sedangkan pada posisi di bawah dan di atas derajat tersebut, pendeteksi gawang kurang baik.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

5.2. Saran

Berikut ini terdapat beberapa saran yang penulis berikan untuk peneliti berikutnya apabila ingin mengembangkan penelitian yang telah dibuat agar menjadi lebih baik adalah sebagai berikut:

1. Peneliti berikutnya diharapkan dapat mengembangkan program *Color Filtering* agar masalah pencahayaan tidak menjadi kendala.
2. Peneliti berikutnya diharapkan dapat mengembangkan program *Hough Transform* pada deteksi gawang tanpa hambatan dalam segi posisi antara robot dan gawang.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA

DAFTAR PUSTAKA

Arief, 2012. *Transformasi Hough.* [Online] Available at: http://informatika.web.id/transformasi-hough.htm [Accessed 14 Desember 2012].

Hadi, M. S., 2012. *Mengenal Robot Sepakbola dan Prinsip Kerjanya.* [Online] Available at: https://msholihulh.wordpress.com/2012/06/08/mengenal-robot-sepakbola/ [Accessed 8 Juni 2012].

Prabowo, H., 2013. *Objek Tracking With OpenCV.* [Online] Available at: http://wayangoneone.blogspot.co.id/2013/01/tracking-objek-with-opencv.html [Accessed 10 Januari 2013].

Priawadi, O., 2017. *OpenCV.* [Online] Available at: http://www.priawadi.com/opencv [Accessed 2017].

Putra, D. E. W., 2017. *Deteksi dan Tracking Objek Berbentuk Bola menggunakan Kalman Filter.* Surabaya: Stikom Surabaya.

Shandy, 2012. HARRIS CORNER DETECTION ALGORITHM. *image processing,* p. 1.

ThePiHut, 2017. *Raspberry Pi 3 - Model B.* [Online] Available at: https://thepihut.com/collections/raspberry-pi/products/raspberry-pi-3-model-b [Accessed 2017].

Yunus, M., n.d. PERBANDINGAN METODE-METODE EDGE DETECTION UNTUK PROSES. *Jurnal Teknologi Informasi,* Volume 3, p. 2.

INSTITUT BISNIS DAN INFORMATIKA stikom SURABAYA