



**ESTIMASI JARAK BOLA BERDASARKAN DATA KAMERA
KATADIOPTRIK MENGGUNAKAN METODE *ARTIFICAL NEURAL
NETWORK***



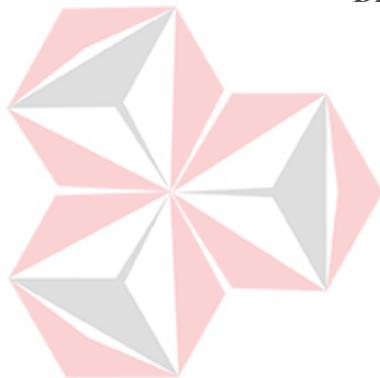
Oleh :
ZENDI ZAKARIA RAGA PERMANA
16410200033

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2020**

**ESTIMASI JARAK BOLA BERDASARKAN DATA KAMERA
KATADIOPTRIK MENGGUNAKAN METODE *ARTIFICAL NEURAL
NETWORK***

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



**UNIVERSITAS
Dinamika**

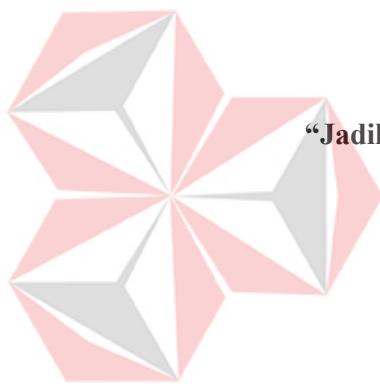
Oleh :

**Nama : Zendi Zakaria Raga Permana
NIM : 16410200033
Program Studi : S1 Teknik Komputer**

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2020



“Jadilah berkat bagi banyak orang, dimanapun dan kapanpun”

UNIVERSITAS
Dinamika

Tugas Akhir

**ESTIMASI JARAK BOLA BERDASARKAN DATA KAMERA
KATADIOPTRIK MENGGUNAKAN METODE *ARTIFICAL NEURAL
NETWORK***

Dipersiapkan dan disusun oleh

Zendi Zakaria Raga Permana

NIM : 16410200033

Telah diperiksa, diuji dan disetujui oleh Dewan Pengaji

Pada : Rabu, 2 September 2020

Susunan Dewan Pengaji

Pembimbing:

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

NIDN. 0727097302

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

NIDN. 0710078601

Pembahas:

Pauladie Susanto, S.Kom, M.T.

NIDN. 0729047501


Digitally signed by
Universitas Dinamika
Date: 2020.09.02
21:34:44 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2020.09.02
22:01:18 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2020.09.02
13:48:28 +07'00'

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


Digitally signed by
Universitas Dinamika
Date: 2020.09.03
23:26:39 +07'00'
Dr. Jusak

NIDN : 0708017101

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA

iii

 Dipindai dengan CamScanner

iii

SURAT PERNYATAAN

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI DAN KEIKHLASAN 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 : Tugas Akhir
Judul Karya : **ESTIMASI JARAK BOLA BERDASARKAN DATA
KAMERA KATADIOPTRIK MENGGUNAKAN METODE
ARTIFICIAL NEURAL NETWORK**

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 Royalty Free Right*) atas seluruh isi/sebagian karya karya ilmiah saya tersebut di atas untuk disimpan, dialih mediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau di publikasikan 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, buka 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 kemudian 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, Juli 2020
Yang menyatakan

Zendi Zakaria Raga P.
NIM : 16410200033

ABSTRAK

Pertandingan *RoboCup Middle Size tingkat Internasional yang diikuti oleh Indonesia* merupakan kategori robot sepak bola beroda. Robot ini, memiliki kecerdasan buatan dengan kemampuan membedakan objek bola, gawang, lapangan, lawan dan objek yang telah ditentukan sebelumnya. Universitas Dinamika Surabaya berkontribusi didalam pertandingan tersebut, robot yang digunakan masih memiliki keterbatasan sudut pandang hanya sebesar 78° . Keterbatasan ini, dapat diselesaikan dengan menggunakan kamera Katadioptrik dengan sudut pandang 360° .

Pada penelitian ini, penulis menerapkan Kecerdasan Buatan yang akan nantinya akan diimplementasikan pada Robot Sepak Bola Beroda pada proses prediksi jarak bola terhadap robot. Mencakup kumpulan metode yang digunakan, diantaranya melalui proses konversi ruang warna citra warna ke ruang citra RGB, minEnclosing, *Euclidean distance* dan *Artifical Neural Network* dengan proses pelatihan 1.000.000 iterasi dan mendapatkan nilai error sebesar 0,054969%, untuk memprediksi jarak piksel dengan jarak sesungguhnya dalam satuan centimeter.

Hasil dari penelitian yang telah dilakukan ini, mengalami beberapa kali percobaan pembelajaran secara *offline* dengan jarak deteksi 40 cm - 400 cm. Hasil pengujian tanpa metode ANN dengan tingkat keberhasilan 63,67%, dan tingkat kelinieran $R^2 = 0.6996$, sedangkan menggunakan metode ANN sebesar 99,03 %, dan $R^2 = 0.9982$.

Kata kunci : Robot, *Euclidean distance*, *Artifical Neural Network*, Kamera Katadioptrik.

KATA PENGANTAR

Penulis panjatkan Puji Syukur kepada Tuhan Yang Maha Esa yang telah menyertai, sehingga penulis dapat meyelesaikan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul "Estimasi Jarak Bola Berdasarkan Data Kamera Katadioptrik menggunakan metode *Artifical Neural Network*" ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Universitas Dinamika.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada :

1. Teruntuk keluarga penulis, yang telah memberikan dukungan atas apa yang penulis lakukan selama ini. Terutama dalam menyelesaikan Tugas Akhir ini.
2. Pimpinan Universitas Dinamika yang telah memberikan kesempatan penulis, untuk dapat menyelesaikan masa pendidikan S1.
3. Bapak Pauladie Susanto, S.Kom., M.T., sebagai Kepala Program Studi Teknik Komputer dan sebagai Pengaji Tugas Akhir penulis yang telah memotivasi penulis untuk terus berkarya.
4. Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., dan Ibu Ira Puspasari, S.Si., M.T., sebagai dosen Pembimbing satu dan dua telah membimbing penulis sampai sejauh ini. Terima kasih juga atas waktu yang telah diberikan kepada penulis.
5. Seluruh rekan Tim Penelitian *Mobile Robot* terima kasih atas kesempatan-nya kepada penulis untuk dapat melakukan penelitian *Omnivision* lebih jauh.
6. Seluruh rekan Tim Robotika Universitas Dinamika, terima kasih telah memberikan semangat dan kekuatan kepada penulis.

7. Untuk Yohan Prakoso Tim Robotika ITS terima kasih telah menjadi sahabat sharing penulis untuk belajar tentang dunia Robotika.

Dalam Laporan Tugas Akhir ini jauh dari kata sempurna. Maka, penulis berharap masukan berupa kritik dan saran yang membangun dari semua pihak dengan harapan dapat memperbaiki penulisan ini kedepan-nya. Penulis memohon maaf jika terdapat sebuah atau banyak kalimat kata yang menyinggung pihak lain yang telah membaca Laporan Tugas Akhir ini. Akhir kata penulis ucapkan Terima Kasih kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, 10 Juni 2020



DAFTAR ISI

Halaman

SURAT PERNYATAAN iv

ABSTRAK v

KATA PENGANTAR vi

DAFTAR ISI viii

DAFTAR GAMBAR xii

DAFTAR TABEL xiii

DAFTAR LAMPIRAN xiv

BAB I 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 3

1.4 Tujuan 3

1.5 Sistematika Penulisan 4

BAB II 5

2.1 Citra Digital 5

2.1.1 Citra RGB 6

2.1.2 Citra HSV 6

2.1.3 Citra Biner 8



| | | |
|---------|---|----|
| 2.2 | Cermin Katadioptrik Kamera 360..... | 9 |
| 2.3 | Kalibrasi Nilai Piksel | 9 |
| 2.4 | Pengukuran Euclidean..... | 10 |
| 2.5 | Morfologi Lingkungan | 10 |
| 2.6 | Jaringan Syaraf Tiruan | 11 |
| 2.6.1 | Backpropagation | 12 |
| 2.6.2 | Prosedur inisialisasi..... | 12 |
| 2.6.3 | Prosedur Perhitungan Feedforward..... | 13 |
| 2.6.4 | Error Backpropagation..... | 13 |
| 2.7 | Regresi Linier dan Korelasi | 15 |
| BAB III | | 16 |
| 3.1 | Blok Diagram Penelitian..... | 16 |
| 3.2 | Pengambilan Frame Citra..... | 16 |
| 3.3 | Kalibrasi Nilai Pixel..... | 17 |
| 3.4 | Proses Konversi ke HSV..... | 17 |
| 3.5 | Mendeteksi bola dan lapangan dengan proses Morfologi OpenCV..... | 18 |
| 3.6 | Proses penentuan letak bola dalam pixel pada kamera katadioptrik..... | 20 |
| 3.7 | Pengukuran jarak bola dengan backpropagation..... | 20 |
| 3.7.1 | Inisialisasi nilai w dan nilai b Artifical Neural Network | 20 |
| 3.7.2 | Normalisasi Input Output Artifical Neural Network | 21 |
| 3.7.3 | Forward propagation pada Matlab 2015 | 21 |
| 3.7.4 | Perhitungan Backward propagation pada Matlab 2015 | 22 |



| | |
|---|----|
| 3.7.5 Update nilai w dan b pada Matlab 2015..... | 24 |
| 3.8 Training menggunakan Neural Net Fitting Tools Matlab..... | 25 |
| BAB IV | 27 |
| 4.1 Pengujian penyesuaian Frame Kamera Katadioptrik..... | 27 |
| 4.1.1 Tujuan Pengujian penyesuaian Frame Kamera Katadioptrik..... | 27 |
| 4.1.2 Alat yang Digunakan penyesuaian Frame Kamera Katadioptrik..... | 28 |
| 4.1.3 Prosedur Pengujian penyesuaian Frame Kamera Katadioptrik..... | 28 |
| 4.1.4 Hasil Pengujian Program penyesuaian Frame Kamera Katadioptrik | 28 |
| 4.2 Pengujian Program HSV Bola dan Lapangan..... | 30 |
| 4.2.1 Tujuan Pengujian Program HSV Bola dan Lapangan..... | 30 |
| 4.2.2 Alat yang Digunakan pada Pengujian HSV Bola dan Lapangan..... | 31 |
| 4.2.3 Prosedur Pengujian Program HSV Bola dan Lapangan..... | 31 |
| 4.2.4 Hasil Pengujian Program HSV Bola dan Lapangan | 32 |
| 4.3 Pengujian Program Morfologi Bola dan Lapangan | 33 |
| 4.3.1 Tujuan Pengujian Program Morfologi Bola dan Lapangan | 33 |
| 4.3.2 Alat yang Digunakan Pengujian Morfologi Bola dan Lapangan | 34 |
| 4.3.3 Prosedur Pengujian Program Morfologi Bola dan Lapangan | 34 |
| 4.3.4 Hasil Pengujian Program Morfologi Bola dan Lapangan | 34 |
| 4.4 Pengujian Program Jarak dalam nilai piksel | 35 |
| 4.4.1 Tujuan Pengujian Program Jarak dalam nilai piksel..... | 35 |
| 4.4.2 Alat yang Digunakan pada Pengujian Jarak dalam nilai piksel | 35 |
| 4.4.3 Prosedur Pengujian Program Jarak dalam nilai piksel..... | 35 |
| 4.4.4 Hasil Pengujian Program Jarak dalam nilai piksel..... | 36 |

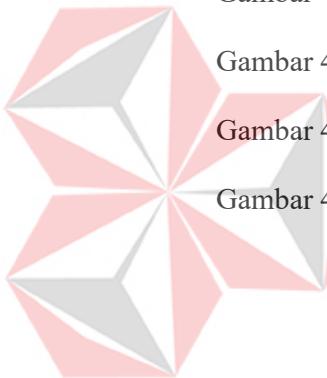


| | |
|---|----|
| 4.5 Pengujian Program Jarak ANN dengan cm | 38 |
| 4.5.1 Tujuan Pengujian Program Jarak ANN dengan cm | 38 |
| 4.5.2 Alat yang Digunakan pada Pengujian Jarak ANN dengan cm | 38 |
| 4.5.3 Prosedur Pengujian Program Jarak ANN dengan cm | 39 |
| 4.5.4 Hasil Pengujian Program Jarak ANN dengan cm..... | 39 |
| 4.6 Pengujian Waktu pada Deteksi Bola dan Prediksi Jarak Bola..... | 41 |
| 4.6.1 Tujuan Pengujian Waktu Deteksi Bola dan Prediksi Jarak Bola | 41 |
| 4.6.2 Alat yang Digunakan Waktu Deteksi Bola dan Prediksi Jarak Bola | 41 |
| 4.6.3 Prosedur Waktu pada Deteksi Bola dan Prediksi Jarak Bola..... | 42 |
| 4.6.4 Hasil Waktu pada Deteksi Bola dan Prediksi Jarak Bola | 42 |
| BAB V..... | 44 |
| PENUTUP..... | 44 |
| 5.1 Kesimpulan | 44 |
| 5.2 Saran..... | 45 |

DAFTAR GAMBAR

Halaman

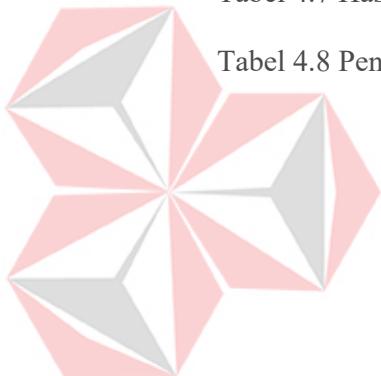
| | |
|--|----|
| Gambar 2.1 Titik awal 2D sebuah citra digital | 5 |
| Gambar 2.2 Ruang warna RGB | 6 |
| Gambar 2.3 Geometri silinder citra warna HSV..... | 7 |
| Gambar 2.4 Citra warna biner..... | 8 |
| Gambar 3.1 Blok Diagram Penelitian..... | 16 |
| Gambar 3.7 Blok Diagram Pendekripsi Bola dan Lapangan..... | 19 |
| Gambar 4.1 (a) Sebelum penyesuaian (b) Setelah penyesuaian | 30 |
| Gambar 4.2 HSV Bola (a) dan HSV Lapangan (b)..... | 33 |
| Gambar 4.3 Grafik persamaan nilai piksel dengan cm..... | 37 |
| Gambar 4.4 Grafik ANN dengan nilai cm | 40 |



UNIVERSITAS
Dinamika

DAFTAR TABEL

| | Halaman |
|---|---------|
| Tabel 4.1 Nilai Trackbar circle | 28 |
| Tabel 4.2 Nilai Trackbar circle2 | 29 |
| Tabel 4.3 Nilai HSV Bola..... | 32 |
| Tabel 4.4 Nilai HSV Lapangan..... | 32 |
| Tabel 4.5 Nilai Morfologi bola dan lapangan | 35 |
| Tabel 4.6 Nilai persamaan jarak piksel dengan cm..... | 36 |
| Tabel 4.7 Hasil nilai ANN dengan cm | 39 |
| Tabel 4.8 Pengujian Waktu Deteksi dan Prediksi Bola | 42 |



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN



Halaman

| | |
|--|----|
| Lampiran 1.1 Letak Kamera Katadioptrik | 47 |
| Lampiran 1.2 Indeks kamera pada device manager | 47 |
| Lampiran 1.3 Pengambilan Citra Visual Studio | 48 |
| Lampiran 1.4 (a) (i) Proses penyesuaian, (ii) Setelah proses penyesuaian | 48 |
| Lampiran 1.5 (b) Program penyesuaian frame citra: | 50 |
| Lampiran 1.6 Konversi RGB ke HSV Visual Studio..... | 50 |
| Lampiran 1.7 Pendektsian Bola dan Lapangan | 50 |
| Lampiran 1.8 (a) citra pada ruang biner (b) hasil dilasi dengan nilai kernel 2px . | 51 |
| Lampiran 1.9 (a) citra pada ruang biner asli (b) nilai erosi bernilai kernel 2 px .. | 52 |
| Lampiran 1.10 (a) citra ruang biner dan (b) dilasi bernilai kernel 10px | 52 |
| Lampiran 1.11.x. Program normalisasi | 52 |
| Lampiran 1.12 Arsitektur ANN Penelitian | 53 |
| Lampiran 1.13 Neural Net Fitting Matlab | 54 |
| Lampiran 1.14 Levenberg-Marquardt..... | 54 |
| Lampiran 1.15 Bayesian Regularization | 54 |
| Lampiran 1.16 Scaled Conjugate Gradient | 54 |
| Lampiran 1.17 Program Visual Studio untuk prediksi real-time pada robot | 55 |

BAB I

PENDAHULUAN

1.1 Latar Belakang



RoboCup Middle Size League adalah suatu pertandingan tingkat *Internasional* dengan kategori robot sepak bola beroda yang mampu memiliki kecerdasan berfikir membedakan objek bola, gawang, lapangan dan objek yang telah ditentukan sebelumnya. Robot Sepak Bola yang dimiliki Universitas Dinamika Surabaya saat ini, masih menggunakan kamera *Webcame* yang menghadap ke depan yang memiliki batas sudut pandang sebesar 78° . Keterbatasan sudut pandang robot tersebut dapat diselesaikan dengan menggunakan kamera Katadioptrik. Kamera Katadioptrik merupakan kamera yang memiliki sudut pandang 360° sehingga sudut pandang robot menjadi luas dan juga sangat membantu proses pengolahan citra robot menjadi jauh lebih cepat untuk menentukan posisi jarak bola terhadap robot.

Jaringan Syaraf Tiruan pada dasarnya adalah suatu metode yang dapat ditemui dalam model matematik yang di dalamnya memiliki kumpulan inti yang saling terhubung secara hubungan paralel yang dapat diilustrasikan seperti jaringan syaraf otak pada manusia. Dengan menerapkan metode yang ada ini pada pengolahan citra robot sepak bola beroda dapat menciptakan kumpulan input berupa data matriks dan output berbentuk nilai skalar yang nantinya akan dijadikan sebagai data pembelajaran sebuah robot sepak bola beroda untuk mengetahui posisi jarak bola terhadap robot. Pada penelitian judul *Artifical Neural Network Untuk Pengukuran*

Posisi Bola Menggunakan Kamera *Omnidireksional* Pada Robot Sepak Bola Beroda mampu mendeteksi keberadaan bola pada jarak antara 40-400 cm dengan nilai error rata-rata sebesar 2.25% (Prakoso, 2017). Penelitian lainnya, telah diterapkan metode *Neural Network* pada umpan balik kamera dengan pembelajaran *offline*, kumpulan data yang telah diajarkan merupakan kumpulan data error sudut dan nilai PWM yang tercatat sebelumnya pada motor kanan dan kiri memiliki, dari proses yang telah terjadi mengalami keberhasilan pengenalan data 95% dan pengontrolan dengan metode *Neural Network* 70% (Kautsar, 2013).

Pada Tugas Akhir ini akan diterapkannya metode Jaringan Syaraf Tiruan dengan perhitungan secara *backpropagation*, yang membantu robot untuk dapat berfikir secara mandiri dan menentukan sebuah nilai jarak bola terhadap robot sesuai dengan nilai jarak secara *real*. Model pembelajaran yang telah dilatih sebelumnya dengan bantuan data *training* pada Matlab diterapkan pada robot, sehingga saat terdapat lebih dari satu robot yang bertanding, robot tidak akan sampai berebut sebuah bola.



1.2 Rumusan Masalah

1. Bagaimana cara mendeteksi sebuah bola dengan menggunakan kamera katadioptrik?
2. Bagaimana cara mendapatkan nilai kalibrasi jarak pixel terhadap jarak objek yang sesungguhnya di lapangan robot sepak bola?
3. Bagaimana menentukan bola yang berada di dalam lapangan?
4. Bagaimana menerapkan *Backpropagation* dapat diterapkan dalam melakukan proses penentuan jarak bola di lapangan terhadap robot?

1.3 Batasan Masalah

1. Objek yang mengalami proses pengukuran hanyalah bola yang berwarna orange yang memiliki diameter 21 cm.
2. Nilai posisi bola merupakan nilai pixel dalam bidang dua dimensi yaitu data nilai x dan y pixel kamera katadioptrik.
3. Sistem ini hanya diterapkan pada lingkungan lapangan sepak bola beroda yang memiliki dimensi 9x6 meter yang memiliki warna hijau seperti lapangan sepak bola pada umumnya.
4. Saat adanya perubahan intensitas cahaya di lingkungan sekitar robot perlu adanya pengaturan ulang nilai dari HSV.



1.4 Tujuan

1. Mendeteksi sebuah bola menggunakan kamera katadioptrik.
2. Mengkalibrasi jarak bola dalam nilai satuan piksel terhadap jarak bola dalam sesungguhnya pada lapangan robot sepak bola yang memiliki dimensi 6x9 meter dan memiliki lingkungan berwarna hijau seperti lapangan sepak bola pada umumnya.
3. Menentukan keberadaan bola dengan metode morfologi lingkungan pada OpenCV.
4. Menerapkan metode *Backpropagation* untuk melakukan penentuan jarak sebuah bola terhadap robot secara real.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dari penelitian ini, dan sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Pada bab ini membahas teori penunjang yaitu citra digital, kamera katadioptrik, kalibrasi nilai piksel, pengukuran jarak *Euclidean*, morfologi lingkungan dan Jaringan syaraf tiruan.

BAB III METODE PENELITIAN

Bab ini berisi tentang pengujian sistem yang meliputi prosedur perancangan perangkat lunak baik secara program, metode *Backpropagation* beserta proses pelatihan *training* robot untuk mendapatkan nilai bobot yang akan digunakan untuk memprediksi nilai citra.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini dibahas tentang pengujian penyesuaian frame citra, coloring deteksi bola dan lapangan, pengujian jarak bola dalam piksel dan pengujian prediksi jarak piksel dengan jarak sesungguhnya.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan penulis serta pengembangan penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Citra Digital

Pada sebuah citra digital tercipta dari proses digitalisasi yang dari setiap bagiannya terdiri dari proses kumpulan data sampling dan nilai data kuantisasi. Nilai sampling yang diperoleh bertujuan untuk dapat membagi bagian layar untuk diubah ke dalam layar kotak-kotak kecil hasil dan memiliki nilai tertentu. Satu bagian citra terdapat sekumpulan titik yang masing-masing akan di representasikan oleh koordinat x dan y yang memiliki sebuah nilai yang disebut *pixel*. Maka, dengan ditemukannya *pixel* akan dijadikan ke bentuk sebuah fungsi dua dimensi $f(x,y)$ dengan nilai titik koordinat awal terletak pada area pojok kiri atas citra dua dimensi. Koordinat yang terbentuk dapat diilustrasikan pada Gambar 2.1 yang menjelaskan bahwa sumbu x bergerak menjauh ke kanan dari titik awal dan sumbu y bergerak menjauh ke bawah dari titik awal.

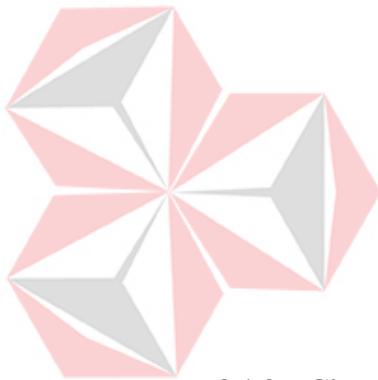


Gambar 2.1 Titik awal 2D sebuah citra digital

(Sumber : Olahan sendiri)

2.1.1 Citra RGB

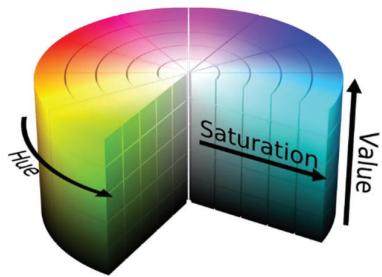
Pada dasarnya indra pengelihatan pada manusia mampu membedakan warna yang dikatakan warna merah, warna hijau, dan warna biru atau dalam istilah pada ruang citra adalah warna RGB. Metode yang terjadi pada ruang warna RGB banyak dijumpai atau ditemui pada sistem rekayasa pengolahan citra pada komputer. Dalam penelitian yang dilakukan cara untuk merepresentasikan tingkat suatu kecerahan nilai warna pada sebuah kanal matrix dapat dinyatakan dalam sebuah tingkat kedalaman warna dalam sebuah nilai dan yang dinyatakan dalam n-bit (Prakoso, 2017). Pada Gambar 2.2 merupakan ilustrasi ruang warna RGB.



Gambar 2.2 Ruang warna RGB

2.1.2 Citra HSV

Citra HSV merupakan singkatan dari *Hue*, *Saturation*, dan *Value* adalah citra warna RGB berawal pada nilai warna yang direpresentasikan ulang ke dalam bentuk geometri HSV. Citra warna HSV dalam bentuk geometri silinder yang dilihat pada Gambar 2.3. 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. Nilai dari Saturation memiliki nilai antara 0% hingga 100%. Value mengartikan sebuah nilai kecerahan dari suatu kumpulan warna. Skala value memiliki nilai antara 0% hingga 100%.



Gambar 2.3 Geometri silinder citra warna HSV

Terjadinya proses konversi RGB ke dalam ruang warna HSV dapat dirumuskan berikut :

$$H = \tan \left[\frac{3(G - B)}{(R - G) + (R - B)} \right] \quad (2.1)$$

$$S = 1 - \frac{\min(R, G, B)}{V} \quad (2.2)$$

$$V = \frac{R + G + B}{3} \quad (2.3)$$



Rumus diatas menjelaskan, apabila S bernilai 0, maka untuk mengetahui nilai H tidak dapat ditentukan secara langsung. Maka dengan persoalan diatas perlu adanya proses normalisasi RGB dengan rumus berikut :

$$r = \frac{R}{R + G + B} \quad (2.4)$$

$$g = \frac{G}{R + G + B} \quad (2.5)$$

$$b = \frac{B}{R + G + B} \quad (2.6)$$

Didapatkan dari perhitungan diatas r, g, dan b yang telah dinormalisasi, maka rumus untuk perubahan RGB ke HSV sebagai berikut :

$$V' = \max(r, g, b) \quad (2.7)$$

$$S' = \begin{cases} 0 & \text{Jika } V = 0 \\ V - \frac{\min(r, g, b)}{V} & \text{Jika } V > 0 \end{cases} \quad (2.8)$$

$$H' = \begin{cases} 0 & \text{Jika } S = 0 \\ \frac{60 * (g - b)}{S * V} & \text{Jika } V = r \\ 60 * \left[2 + \frac{(b - r)}{S * V} \right] & \text{Jika } V = g \\ 60 * \left[4 + \frac{(r - g)}{S * V} \right] & \text{Jika } V = b \end{cases} \quad (2.9)$$

$$H = H + 30 \quad \text{Jika } H < 0 \quad (2.10)$$

Perhitungan yang telah terjadi dapat memisahkan nilai warna sehingga dengan perhitungan ini merupakan cara langkah perhitungan yang ideal untuk mengembangkan algoritma ke dalam pemrosesan citra yang lebih kompleks dan ditujukan untuk klarifikasi (Rakmawati, 2013).



2.1.3 Citra Biner

Citra biner dalam pengolahan citra adalah citra yang hanya direpresentasikan ke dalam dua nilai saja yaitu 0 dan 1 yang berarti putih atau hitam. Penerapan citra biner sangat banyak dimanfaatkan untuk melakukan pemisahan antara objek yang dituju dengan latar belakang objek yang diabaikan. Pada Gambar 2.4 dijadikan sebagai ilustrasi citra dalam ruang biner.



Gambar 2.4 Citra warna biner

2.2 Cermin Katadioptrik Kamera 360

Dunia grafika banyak sekali memanfaatkan dari fungsi kamera untuk mengambil sebuah gambar dan Kamera katadioptrik adalah kamera yang memiliki luas sudut pandang 360° sehingga menjadikan objek yang tertangkap kamera mampu menangkap disekitar kamera secara keseluruhan, kamera dengan lensa katadioptrik dapat dilihat pada Gambar 2.5



Gambar 2.5 Lensa Cermin pada Kamera

2.3 Kalibrasi Nilai Piksel

Proses kalibrasi *frame* dengan perhitungan seperti yang dilakukannya penelitian Sistem Navigasi Robot Beroda Menggunakan Omnidirectional Camera (Prakoso, 2015) pada penelitian yang telah dilakukan *frame* citra berukuran 500x500 *pixel*.

$$cx = x - 250 \quad (2.11)$$

$$cy = -(y - 250)$$

Keterangan :

x : nilai pembacaan kamera pada sumbu x sebelum dikalibrasi (*pixel*)

y : nilai pembacaan kamera pada sumbu y sebelum dikalibrasi (*pixel*)

cx : nilai pembacaan kamera pada sumbu x setelah dikalibrasi (*pixel*)

cy : nilai pembacaan kamera pada sumbu y setelah dikalibrasi (*pixel*)

2.4 Pengukuran *Euclidean*

Setelah pendekalian bola, langkah selanjutnya menyelesaikan permasalahan untuk dapat menentukan posisi dimana bola berada pada citra kamera katadioptrik. Terdapat rumus *Euclidean* untuk menyelesaikan permasalahan mencari jari – jari bola terhadap titik origin kamera katadioptrik (2.12) dan jarak bola terhadap titik origin kamera katadioptrik (2.13) dimana jarak (d) dalam pixel, nilai x,y *pixel* pertama dan nilai x,y *pixel* kedua adalah sebagai berikut :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.12)$$

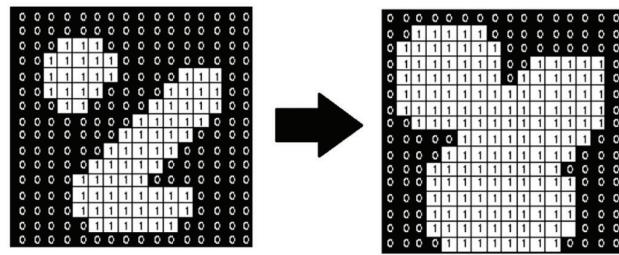
$$\theta = \tan^{-1}(y_2 - y_1) - (x_2 - x_1) \quad (2.13)$$



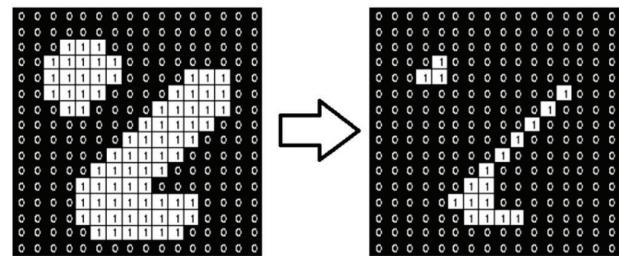
2.5 Morfologi Lingkungan

Morfologi dalam ilmu biologis merupakan teknik penyesuaian lingkungan jika dalam pengolahan citra ditujukan untuk teknik memperbaiki citra yang terhilang atau menghaluskan sebuah citra. Mengolah sebuah citra dengan teknik morfologi mampu membantu proses citra keabuan (grayscale) menjadi lebih baik. Operasi morfologi yang dapat digunakan terdapat Dilasi sebagai penebalan piksel dan Erosi sebagai pengikisan piksel.

Pengertiannya secara teori pun tetap sama bahwa dilasi ditujukan untuk menebalan citra piksel yang dirasa perlu dilakukan penebalan dan jika awal sebuah piksel bernilai 0 maka akan diubah menjadi bernilai 1 diharapkan terjadi perubahan penebalan piksel begitu juga sebaliknya sama dengan Erosi terjadi perubahan nilai 1 menjadi nilai 0. Pada Gambar 2.6 merupakan ilustrasi morfologi Dilasi dan pada Gambar 2.7 merupakan ilustrasi morfologi Erosi.



Gambar 2.6 Morfologi Dilasi



Gambar 2.7 Morfologi Erosi



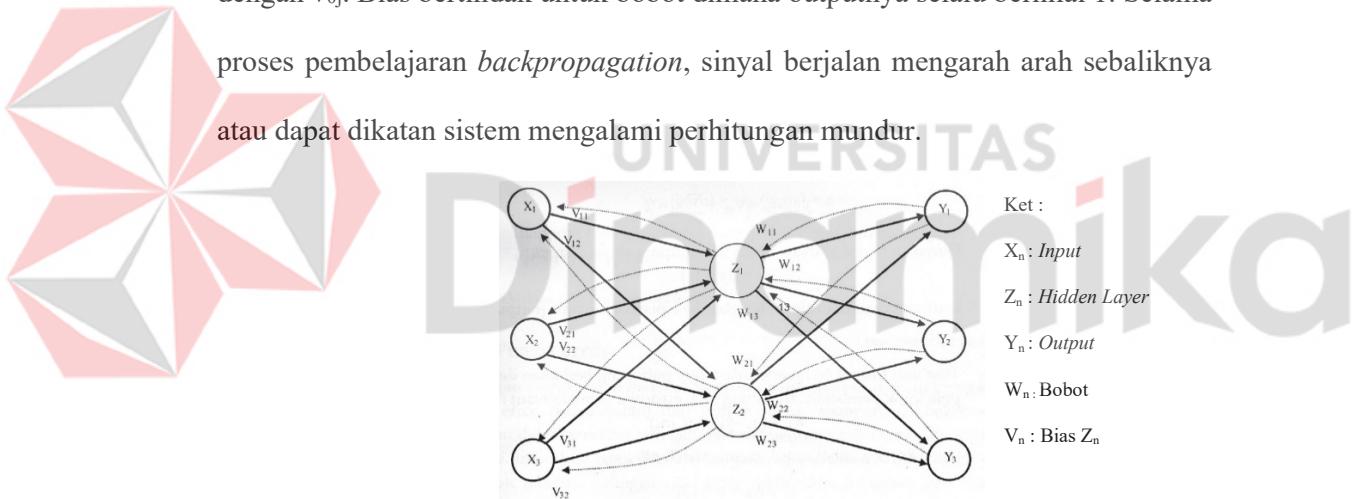
2.6 Jaringan Syaraf Tiruan

Dalam kecerdasan buatan meniru prinsip kerja dari Jaringan Syaraf pada otak manusia. Dikarenakan pada setiap neuron otak manusia saling terhubung sehingga banyak sistematis yang tercipta secara kompleks. Kumpulan dari neuron tersebut menciptakan kumpulan perceptron dan diharapkan dengan menerapkan prinsip tersebut sistem dalam mengalami proses pembelajaran maupun pengenalan objek secara berkala.

Metode Jaringan Syaraf Tiruan banyak sekali jenisnya antara lain *backpropagation*, *linear associative memory*, *clustering network* namun pada setiap metode yang ada menyesuaikan pada kebutuhan. Pada penelitian ini akan menerapkan *backpropagation*.

2.6.1 Backpropagation

Backpropagation adalah metode pembelajaran yang mengoreksi error sebuah sistem jaringan saraf tiruan, untuk sebuah tujuan memperkecil nilai error yang dihasilkan pada suatu keluaran sistem matematis. Untuk mendapatkan nilai error, tahapan perambatan maju harus dilakukan terlebih dahulu dengan mengaktifkan neuron dengan aktivasi sigmoid. Pada Gambar 2.8 merupakan Arsitektur Jaringan *Backpropagation*. x_i merupakan sebuah unit masukan yang masuk ke dalam layer tersembunyi dengan z_j dan unit keluaran berada pada y_k . Bias pada unit output y_k ditandai dengan w_{0k} , bias pada unit tersembunyi z_j ditandai dengan v_{0j} . Bias bertindak untuk bobot dimana outputnya selalu bernilai 1. Selama proses pembelajaran *backpropagation*, sinyal berjalan mengarah arah sebaliknya atau dapat dikatakan sistem mengalami perhitungan mundur.



Gambar 2.8 Arsitektur Jaringan *Backpropagation*

2.6.2 Prosedur inisialisasi

Pada buku Sistem Kecerdasan Tiruan Dengan Kemampuan Belajar dan Adaptasi memberikan penjelasan tentang proses inisialisasi penyelesaian tetang metode *backpropagation* sebagai berikut.

Langkah 0: Dilakukan inisialisasi awal bobot dan masukan setiap layer dengan metode nilai acak gaussian dengan rata-rata 0 dan deviasi 0,1.

Langkah 1: Jika kondisi terhenti dan tidak tercapai, maka ulangi langkah 2 – 9.

Langkah 2: Untuk setiap pasangan pembelajaran, lakukan langkah 3 - 8.

2.6.3 Prosedur Perhitungan Feedforward

Langkah 3: Setiap unit input (x_i , $i = 1, \dots, n$) akan menerima sinyal input x_i dan menyebarlakannya ke semua unit di layer di atasnya (*hidden layer*).

Langkah 4: Setiap unit tersembunyi (z_i , $j = 1, \dots, p$) menjumlahkan setiap sinyal input yang telah dikalikan dengan pembobotnya.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.14)$$

Tetapkan fungsi aktivasi untuk menghitung sinyal outputnya.

$$z_i = f(z_in_i) \quad (2.15)$$

Dan mengirimkan sinyal ini pada semua unit pada layer diatasnya (*layer output*).

Langkah 5: Setiap unit output (y_k , $k = 1, \dots, m$) menjumlahkan sinyal input berbobotnya,

$$y_in_k = w_{0k} + \sum_{j=1}^n z_j w_{jk} \quad (2.16)$$

dan menetapkan fungsi aktivasi untuk menghitung sinyal outputnya.

$$y_k = f(y_in_k) \quad (2.17)$$

2.6.4 Error Backpropagation

Langkah 6: Setiap unit output (y_k , $k = 1, \dots, m$) menerima pola target yang berhubungan dengan pola pelatihan input, menghitung error informasinya.

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (2.18)$$

Hitung bobot pengoreksi (digunakan untuk memperbarui w_{jk} nanti),

$$\Delta w_{0k} = \mu \delta_k z_j \quad (2.19)$$

Menghitung pengoreksi biasnya (digunakan untuk memperbarui w_{0k} nanti)

$$\Delta w_{0k} = \mu \delta_k z_j \quad (2.20)$$

Langkah 7: Kumpulan unit tersembunyi (z_i , $j = 1, \dots, p$) menjumlahkan selisih inputnya (dari unit di layer atas)

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.21)$$

Kalikan dengan deratifit dari fungsi aktivasinya untuk menghitung error informasi

$$\delta_i = \delta_{in_j} f'(z_{in_j}) \quad (2.22)$$

Hitung bobot pengoreksinya (digunakan untuk memperbarui v_{ij} nanti),

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.23)$$

Dan menghitung bias pengoreksinya (digunakan untuk memperbarui v_{0j} selanjutnya),

$$\Delta v_{0j} = \alpha \delta_j x_i \quad (2.24)$$

Langkah 8: Kumpulan unit output (y_k , $k = 1, \dots, m$) memperbarui bias dan bobotnya ($j = 0, \dots, p$)

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (2.25)$$

Setiap unit tersembunyi (z_j , $j = 1, \dots, p$) memperbarui bias dan bobotnya ($i = 0, \dots, n$):

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad (2.26)$$

Langkah 9: **Tes kondisi berhenti.**

2.7 Regresi Linier dan Korelasi

Regresi linier adalah persamaan yang menunjukkan hubungan variabel bebas (predictor (X)) dan variabel tak bebas (response (Y)) (Yuliara, 2016).

Persamaan regresi, sebagai berikut:

$$Y = a + bX \quad (2.27)$$

Y: garis regresi

a: konstanta perpotongan dengan sumbu y

b: konstanta regresi

X: variabel bebas

Beberapa data hasil prediksi jarak pada Tugas Akhir ini, akan ditampilkan dalam bentuk grafik, sehingga membutuhkan regresi linier dan korelasi.

Hubungan antara jarak image dan jarak nyata pada Tugas Akhir ini perlu diukur dengan analisis korelasi. Dimana hasil korelasi ini akan membandingkan nilai jarak image dan jarak nyata sebelum dan setelah menerapkan ANN. Persamaan koefisien korelasi:

$$R = \frac{n \sum_{i=1}^n X_i Y_i - \left(\sum_{i=1}^n X_i \right) \left(\sum_{i=1}^n Y_i \right)}{\sqrt{\left[n \sum_{i=1}^n X_i^2 - \left(\sum_{i=1}^n X_i \right)^2 \right] \left[n \sum_{i=1}^n Y_i^2 - \left(\sum_{i=1}^n Y_i \right)^2 \right]}} \quad (2.28)$$

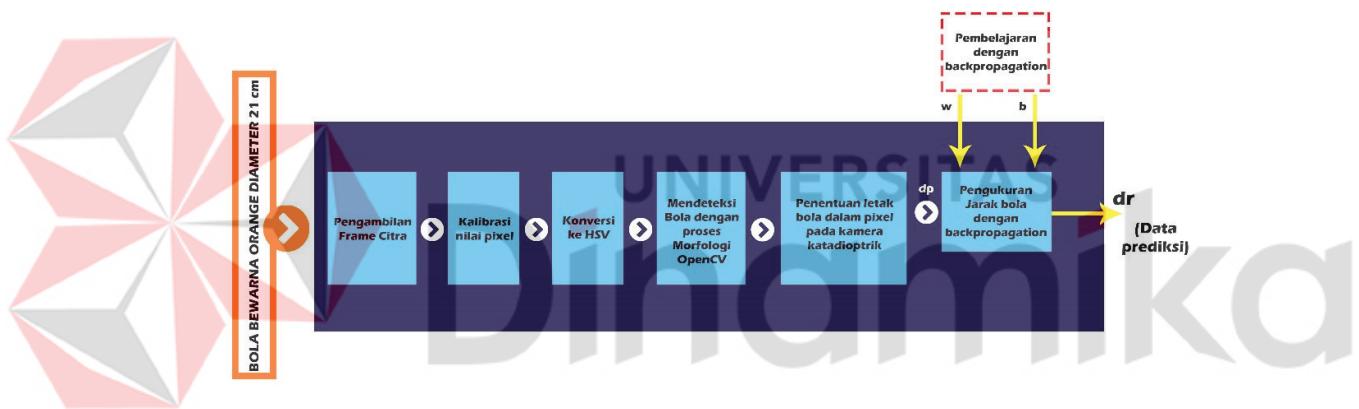
Koefisien determinasi (R^2) merupakan kuadrat koefisien korelasi. Kuatnya hubungan korelasi pada Tugas Akhir ini mengacu pada (Pratomo, 2015), dengan interpretasi nilai interval dan tingkat hubungan sebagai berikut: 0,00-0,19 (sangat rendah), 0,20-0,39 (rendah), 0,40-0,59 (sedang), 0,60-0,79 (kuat), dan 0,80-1,00 (sangat kuat).

BAB III

METODE PENELITIAN

3.1 Blok Diagram Penelitian

Blok Diagram Penelitian robot ini merupakan fokus utama dalam proses pengukuran jarak bola dalam nilai piksel terhadap robot dalam nilai sesungguhnya. Urutan proses sistem dapat dilihat pada Gambar 3.1



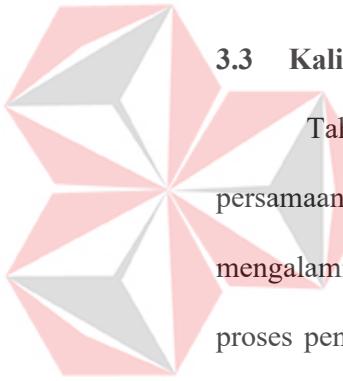
Gambar 3.1 Blok Diagram Penelitian

(Sumber : Olahan sendiri)

Penjelasan secara detail pada blok diagram penelitian adalah sebagai berikut :

3.2 Pengambilan *Frame* Citra

Pengambilan *Frame* citra ditujukan untuk melakukan tangkapan pada kamera katadioptrik. Citra yang akan ditetapkan memiliki resolusi 640x480. Kamera katadioptrik diletakkan di bagian atas robot dengan jarak maksimal 60 cm dari roda robot dapat dilihat pada Lampiran 1.1.



Untuk langkah awal pengambilan gambar pada OpenCV perlu adanya proses deklarasi terlebih dahulu untuk membuat variable yang mampu sebagai penampung gambar dari kamera yang diaktifkan yaitu dengan syntax **VideoCapture**. Penelitian ini kamera yang terbaca dan yang digunakan adalah HD USB Camera dapat dilihat pada Lampiran 1.2., citra ditampung dengan cara memindahkannya ke sebuah struktur data **Mat**, disediakan pada *library* OpenCV. Citra awal yang diambil masih dalam ruang warna RGB atau BGR. Hasil pengambilan citra pada Visual Studio dan OpenCV dapat dilihat pada Lampiran 1.3.

3.3 Kalibrasi Nilai Pixel

Tahapan berikutnya adalah kalibrasi nilai pixel yang ditulis menggunakan persamaan (2.11). Pada Gambar Lampiran 1.4 (a), (i) merupakan Citra sebelum mengalami proses penyesuaian dan (ii) merupakan Citra yang telah mengalami proses penyesuaian. Dimana, titik awal citra sudah mengalami perubahan frame citra terjadi adanya penutupan lingkaran hitam dan lingkaran putih. Cuplikan Program dan Hasil proses ini ditunjukkan pada Lampiran 1.4 (b).

3.4 Proses Konversi ke HSV

Konversi citra ke HSV ditujukan supaya citra dapat dikelompokkan ke ruang HSV dari ruang RGB. Mengingat informasi yang didapat di citra RGB belum mampu memisahkan antara ruang warna dan informasi kecerahan. Citra HSV sendiri dipisahkan menggunakan kanal *value*. Konversi warna menggunakan

sebuah fungsi pada *library* OpenCV yaitu dengan parameter **void cvtColor (InputArray src, OutputArray dst, int code, int dstCn=0)**.

InputArray **src** merupakan data masukan atau input pada sebuah citra yang akan diproses nantinya. OutputArray **dst** merupakan nilai keluaran suatu citra yang telah melalui proses konversi. int **code** yaitu untuk menentukan konversi jenis apa yang akan digunakan. int **dstCn=0** menjelaskan banyaknya jumlah kanal yang akan diterapkan pada output konversi. Pada tugas akhir menerapkan kode CV_BGR2HSV. Dapat dilihat hasil dari konversi citra dari Visual Studio pada Lampiran 1.5.

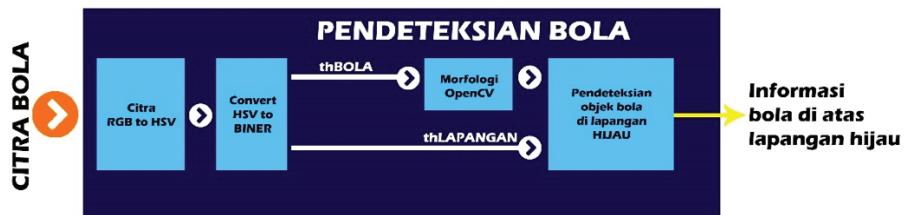
Berikut merupakan program konversi Citra RGB ke Citra HSV :

```
cvtColor(frame_asli, frame_konversi_HSV,  
        CV_BGR2HSV);  
if (frame_konversi_HSV.empty()) cerr << "Error,  
    Konversi tidak berhasil karena frame  
    kosong!\n";  
else imshow("Citra HSV", frame_konversi_HSV);
```

3.5 Mendeteksi bola dan lapangan dengan proses Morfologi OpenCV

Tahap selanjutnya adalah mendeteksi bola dengan proses morfologi yang dimiliki oleh OpenCV. Pendeksteksian memanfaatkan ciri-ciri yang dimiliki bola yaitu bola berwarna orange gelap yang hanya berada pada lapangan berwarna hijau.

Proses pendeksteksian kedua objek tersebut dengan mengubah ruang citra RGB ke ruang citra HSV dan selanjutnya konversi ke ruang citra biner. Pada Gambar 3.2 merupakan blok diagram pendeksteksian bola dengan proses morfologi sehingga informasi yang didapat hanya citra bola yang berada di atas lapangan berwarna hijau saja. Hasil pendeksteksian bola dan lapangan pada Lampiran 1.6.



Gambar 3.2 Blok Diagram Pendekatan Bola dan Lapangan

(Sumber : Olahan sendiri)

Program Morfologi Citra Bola dan Lapangan dapat dilihat pada Lampiran 1.7x.

Pada saat berlangsungnya teknik opening citra dilakukan pada variable `opening_morfologi_bola` bernilai lebih dari 0. Jika proses yang terjadi saat variable `opening_morfologi_bola` bernilai 0 maka akan terjadi sebuah kesalahan pada program atau dapat dikatakan sebagai error. Dikarenakan tidak dapat berjalan pada kernel 0. Hasil morfologi ditunjukkan pada Lampiran 1.7., 1.8, 1.9, dan 1.10.

Pada proses seleksi digunakan fungsi yang dimiliki pada opencv dengan fungsi `minEnclosingCircle` (`((Mat)contours_bola[i], center_bola[i], radius_bola[i])`). Pada fungsi yang digunakan menjadikan objek dapat ditandai dengan berupa lingkaran dengan radius kecil dan mampu memenuhi bagian objek dalam satu titik tanda yang timbul. Objek yang ditandai sebelumnya memiliki nilai radius yang disimpan pada variable `radius` dan titik tengah citra disimpan pada variable `center`. Untuk lengkapnya dapat dilihat pada program sebagai berikut :

```
minEnclosingCircle((Mat)contours_bola[i],center_bola[i],radius_bola[i]);
koreksi_x_nilai_positif=center_bola[i].x+(radius_bola[i]+5);
koreksi_x_nilai_negatif=center_bola[i].x-(radius_bola[i]+5);
koreksi_y_nilai_positif=center_bola[i].y+(radius_bola[i]+5);
```

```
koreksi_y_nilai_negatif=center_bola[i].y-  
(radius_bola[i]+5);
```

3.6 Proses penentuan letak bola dalam *pixel* pada kamera katadioptrik

Proses menentukan letak bola akan terjadi jika terdeteksi adanya bola di lapangan hijau. Perhitungan yang dihunakan mengacu pada persamaan (2.12) dan (2.13). Lampiran 1.11 (a) dan (b), merupakan program dan hasil penentuan jarak bola dalam piksel.

3.7 Pengukuran jarak bola dengan *backpropagation*

Setelah informasi radius bola terhadap titik origin dan jarak bola dalam piksel terhadap titik origin kamera katadioptrik. Perlu diketahui bahwa jarak bola yang ada pada kamera katadioptrik tidak linier dengan jarak yang sesungguhnya. Dikarenakan jarak yang terdapat pada kamera katadioptrik berupa nilai pixel dan jarak *real* dinyatakan dalam satuan cm. Salah satu metode yang dapat diterapkan pada permasalahan seperti ini adalah dengan menggunakan sistem jaringan syaraf tiruan. Untuk menentukan nilai *learning* digunakan aplikasi Matlab untuk dapat menentukan bobot w dan b. Isi dari fungsi tersebut adalah sebagai berikut :

3.7.1 Inisialisasi nilai w dan nilai b *Artifical Neural Network*

Untuk melakukan inisialisasi awal perlu melakukan proses memunculkan nilai acak dengan nilai antara -1 sampai 1 diberlakukan untuk nilai w dan b pada setiap layer yang terbentuk. Pada Matlab pengacakan nilai dilakukan dengan cara menggunakan fungsi `rand()` yang telah disediakan Matlab. Namun, perlu

diperhatikan fungsi `rand()` yang tersedia pada Matlab mengeluarkan nilai tidak beraturan atau acak dengan *range* 0 sampai 1 yang didistribusikan secara merata.

Penelitian ini menggunakan fungsi untuk menggantikan fungsi `rand()` yaitu bernama `randnew(baris,kolom)`. Karena sebuah fungsi tentunya memiliki parameter yaitu "row" sebagai baris dan "col" sebagai kolom mengartikan ukuran baris dan kolom matrik yang ingin diacak nantinya. Parameter tersebut mengartikan sebagai berikut :

```
function r = randnew(baris, kolom)
r = 2*rand(baris,kolom)-1;
```

3.7.2 Normalisasi Input Output *Artifical Neural Network* pada Matlab 2015

Untuk menentukan nilai masukan dan keluaran perlu adanya proses normalisasi terlebih dahulu. Pada proses forward propagation supaya nilai input maupun output tetap berada pada antara daerah kerja *transfer function* baik log sigmoid, tangen sigmoid. Penelitian ini menetapkan sebuah fungsi yang memiliki parameter untuk melakukan proses normalisasi yaitu `normf(x_masukan,xmin_masukan,xmax_masukan)`, rentang nilai `xmin_masukan` dan `xmax_masukan` rentang tersebut bernilai -1 sampai 1. Penulisan program pada fungsi Matlab sebagai berikut berupa data masukan, nilai target setiap keluaran dan normalisasi setiap masukan, dapat diihat pada Lampiran 1.11. x.

3.7.3 Forward propagation pada Matlab 2015

Forward propagation suatu proses memasukkan nilai data input ke dalam layer awal dan menyebarkan ke layer-layer selanjutnya. Dikarenakan perhitungan yang terjadi pada *fordward* propagation adalah perhitungan maju. Dibutuhkan juga sebuah transfer function. Terdapat beberapa jenis *transfer function* yang digunakan

dalam sistem *Artifical Neural Network* diantaranya adalah log sigmoid, tangen log sigmoid, linier, dan hard limiter. Jenis transfer function yang digunakan pada penelitian ini adalah tftype. Dianalogikan, jika tftype bernilai 0 maka transfer function yang digunakan adalah linier, jika sebaliknya yaitu bernilai 1 menandakan transfer function yang digunakan adalah sigmoid.

Dikarenakan program pelatihan ini merupakan pelatihan offline pada Matlab maka untuk dapat diterapkan dapat ditulis menjadi sebuah fungsi yaitu fpnet (*p,w,b,tftype*). Fungsi ini memiliki parameter yang digunakan yaitu *p* sebuah vektor input, *w* (bobot) dan *b*(bias) merupakan matriks yang akan digunakan pada layer. Berikut merupakan program fungsi yang dibuat:

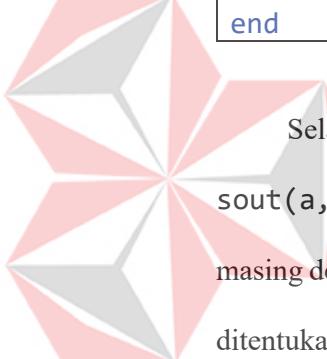
```
function [a] = fpnet (p,w,b,tftype) [~,col] = size(p);
o = ones(1,col); a = [];
n = w*p + b*o;
if tftype == 0,
a = n;
end
if tftype == 1,
a = 1 ./ (1+exp (-n));
end
if tftype == 2,
a = (exp(n)-exp(-n)) ./ (exp(n)+exp(-n));
end
if tftype == 3,
a = hardlim(n);
end
```

3.7.4 Perhitungan Backward propagation pada Matlab 2015

Proses kedua adalah proses perhitungan secara *backward* yaitu perhitungan secara mundur, dilakukan untuk mendapatkan nilai *error* yang dibandingkan terhadap target yang telah ditentukan awalnya disebut dengan sensifitas layer. Untuk mendapatkan nilai-nilai tersebut pada Matlab dapat dituliskan dengan sebuah fungsi. Fungsi dtf(*a,tftype*). Fungsi yang terbentuk memiliki 2

parameter masing-masing adalah a yang dijadikan sebagai input layer dan $tftype$ dijadikan sebagai tipe transfer function yang digunakan pada layer. Terdapat kumpulan nilai yang dapat dimasukkan ke dalam fungsi $tftype$ yaitu 0 untuk linier, 1 untuk log sigmoid dan 2 untuk tangen sigmoid. Program yang dapat diterapkan adalah sebagai berikut :

```
function [dv] = dtf(a,tftype) dv=[];  
if tftype == 0,  
dv = diag(ones(size(a)));  
end  
if tftype == 1,  
dv = diag(a .* (1-a));  
end  
if tftype == 2,  
dv = diag(1 - a .* a);  
end
```



Selanjutnya adalah proses sensifitas dari setiap layer, menggunakan fungsi $sout(a,t,tftype)$. Pada fungsi yang digunakan memiliki parameter masing-masing dengan a sebagai keluaran dari layer output, t sebuah target jarak yang telah ditentukan sebelumnya. Transfer function merupakan fungsi aktivasi yang telah ditentukan dengan fungsi $dtf(a,tftype)$ yang telah dijelaskan sebelumnya.

Penerapan untuk fungsi ditunjukkan sebagai berikut :

```
function [so] = sout(a,t,tftype)  
so = -2*dtf(a,tftype)*(t - a);
```

Nilai sensifitas dari layer yang berbeda yaitu dengan $slayer(a,w,s,tftype)$. Fungsi yang terbentuk menjadikan 4 parameter penting masing-masing adalah a merupakan keluaran dari layer, w merupakan bentuk matriks w pada layer, s merupakan nilai sensifitas layer. Berikut adalah programnya :

```
function [sl] = slayer(a,w,s,tftype)
sl = dtf(a,tftype)*(w')*s;
```

Setelah proses backward propagation dilakukan, pada setiap layer yang ada akan memiliki nilai sensifitas masing-masing. Nilai tersebut akan digunakan untuk bahan pertimbangan dan untuk memperbarui nilai w dan b pada setiap layer pada tahap selanjutnya.

3.7.5 Update nilai w dan b pada Matlab 2015

Proses selanjutnya perlu adanya pembaruan nilai bobot pada setiap layer dengan menggunakan nilai error yang telah didapat sebelumnya dari proses *backward*. Program untuk melakukan proses pembaruan bobot sebagai berikut :

```
bobot_w2 = bobot_w2 - lr*s2*a1';
bias_b2 = bias_b2 - lr*s2;
bobot_w1 = bobot_w1 - lr*s1*a0';
bobot_b1 = bobot_b1 - lr*s1;
```

Saat sistem mengalami update nilai bobot w dan b langkah berikutnya melakukan proses pengoreksian nilai output dengan melihat dari error yang dihasilkan. Jika nilai keluaran error kurang dari nilai error yang diharapkan proses pembelajaran telah selesai dan nilai bobot w dan b dapat ditentukan sebagai nilai yang dapat digunakan. Sedangkan jika nilai error yang dihasilkan masih lebih besar daripada nilai error yang diharapkan, maka proses diulangi lagi pada poin sebelumnya. Penelitian ini menggunakan arsitektur jaringan syaraf tiruan dengan susunan ANN yang digunakan terdiri dari satu layer tersembunyi dan satu keluaran layer. Layer tersembunyi sendiri memiliki satu masukan dan sepuluh keluaran. Gambar arsitektur dapat dilihat pada Lampiran 1.12.

Arsitektur ANN terdiri dari satu masukan dan satu keluaran. ANN yang digunakan terdiri dari satu layer tersembunyi dan satu keluaran layer. Layer tersembunyi sendiri memiliki satu masukan dan sepuluh keluaran. Untuk memudahkan pengolahan matrik tersebut dapat dibantu dengan adanya *library* Armadillo. *Library* ini menggunakan perhitungan yang kompleks dalam matrik tanpa melakukan secara manual untuk efisiensi dari berjalannya sebuah program.

3.8 Training menggunakan Neural Net Fitting Tools Matlab

Penelitian ini juga menggunakan *Neural Net Fitting* dengan tujuan untuk membandingkan hasil pembelajaran yang terbaik. Terdapat tiga jenis pelatihan Levenberg-Marquardt, Bayesian Regularization, dan Scaled Conjugate Gradient.

Gambar tampilan *tools Neural Net Fitting* pada Matlab, menggunakan metode Leven, Bayesian, dan metode Scaled, ditunjukkan pada Lampiran 1.13, 1.14, 1.15 dan 1.16. Program Visual Studio untuk prediksi *real-time* pada robot dapat dilihat pada Lampiran 1.17.

Setelah sudah mendapatkan nilai-nilai bobot w dan b pada Matlab maka untuk proses prediksi secara *real-time* membutuhkan perhitungan matrik kembali untuk memproses nilai yang didapatkan dari kamera nantinya. Fungsi bernama `normalisasi_masukan(float masukan, float masukan_Min, float masukan_Max)`. Dan untuk memasukkan data yang telah ternormalisasi melalui proses kedua layer adalah sebagai berikut.

```
float nilai_normalisasi =
    panggil_normalisasi(_jarak_dalam_piksel, 0, 231.00);
layer1 = (w1*nilai_normalisasi) + b1;
layer1 = 1 / (1 + exp(-layer1));
layer2 = (w2*layer1) + b2;
float out = layer2.at(0.0);
```

```
return nilai_denormalisasi(out, 0, 500);
```

Digunakan variable normalizedN untuk melakukan tahapan proses masukan kedalam layer-layer yang telah dibuat, layer1: matrik 10x1 sebagai penampung data output dari layer 1 (hidden layer), layer2: matrik 1x1 sebagai penampung data output dari layer 2(output layer), variable out: variable yang digunakan untuk data output dari pengolahan proses ANN bukan berbentuk matriks, sehingga dapat melakukan proses selanjutnya.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian penyesuaian Frame Kamera Katadioptrik

Pengujian penyesuaian Frame Kamera Katadioptrik untuk melakukan penyesuaian hasil tangkap kamera sesuai dengan kebutuhan.

4.1.1 Tujuan Pengujian Program penyesuaian Frame Kamera Katadioptrik

Tujuan dari pengujian ini untuk menutupi bagian tengah pada kamera dengan lingkaran putih dikarenakan saat proses pembacaan citra terdapat noise pada bagian tengah kamera *noise* tersebut berupa bayang-bayang pantulan dari cermin katadioptrik warna yang dipantulkan pun tergantung pada objek warna disekitar. Jika pantulan tersebut berupa warna yang sama dengan objek yang telah ditentukan akan menjadi sebuah permasalahan dimana objek seolah-olah terdapat lebih dari satu. Namun, objek pantulan terlihat samaran atau bayang-bayang maka diperlukan adanya penutupan pada bagian tengah citra. Pada bagian samping kamera juga perlu adanya penutupan citra dengan cara yang sama dikarenakan samping kamera katadioptrik terdapat efek buram yang tidak menentu warnanya. Namun menggunakan penerapan yang berbeda.



4.1.2 Alat yang Digunakan pada Pengujian Program penyesuaian Frame

Kamera Katadioptrik

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain :

1. Laptop.
2. OpenCV.
3. OpenFrameworks.
4. Visual Studio 2015.
5. Kamera Katadioptrik.

4.1.3 Prosedur Pengujian Program penyesuaian Frame Kamera Katadioptrik

Langkah-langkah untuk melakukan pengujian program penyesuaian Frame Kamera Katadioptrik adalah sebagai berikut :

1. Memasangkan *library* OpenCV kedalam laptop yang akan digunakan.
2. Memasangkan OpenFrameworks kedalam laptop yang akan digunakan.
3. Meng-*install* Visual Studio 2015 di laptop.
4. Memasukkan OpenCV dan OpenFrameworks kedalam Visual Studio 2015.
5. Memasangkan *cable* USB Kamera Katadioptrik dengan laptop.
6. Mengatur nilai pada *trackbar* untuk menyesuaikan posisi lingkaran putih pada kamera dan lingkaran hitam pada samping kamera.

4.1.4 Hasil Pengujian Program penyesuaian Frame Kamera Katadioptrik

Tabel 4.1 Nilai Trackbar circle

| Nilai Trakbar circle | | |
|----------------------|-----|----|
| x | y | r |
| 338 | 237 | 43 |

(Sumber : Olahan sendiri)

Pada Tabel 4.1 merupakan nilai koordinat untuk menutup pada bagian hasil tengah citra kamera katadioptrik. Yang nantinya akan digunakan untuk pergeseran Trackbar bernama circle pada Visual Studio. Pada Tracbar circle terdapat x,y dan r x dan y untuk melakukan pergeseran koordinat lingkaran putih ke tengah citra kamera katadioptrik. r ditujukan untuk menjadikan lingkaran putih pada citra tengah lingkaran dapat dikecilkan atau dibesarkan ukurannya. Dapat dikatakan r sebagai nilai besar kecilnya radius lingkaran putih pada tengah citra kemra katdioptrik. Semakin besar nilai x lingkaran putih akan semakin begeser ke kanan, semakin besar nilai y lingkaran putih akan semakin bergeser ke bawah. Berbeda dengan r, jika nilai r semakin besar akan semakin besar juga ukuran dari lingkaran putih utnuk menutup tengah citra kamera katadioptrik.

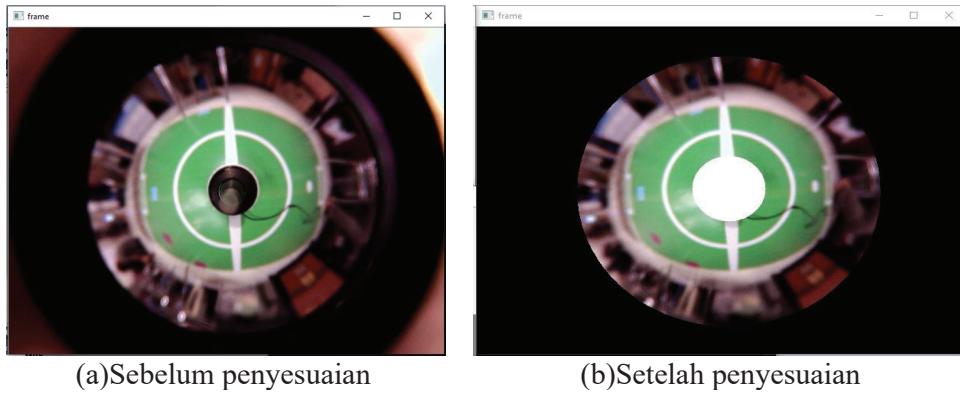
Tabel 4.2 Nilai Trackbar circle2

| Nilai Trackbar circle2 | | | |
|------------------------|-----|-----|-----|
| X | Y | R | t |
| 338 | 237 | 309 | 329 |

(Sumber : Olahan sendiri)

Pada Tabel 4.2 merupakan nilai koordinat untuk menutup pada hasil samping citra kamera katadioptrik. Terdapat x,y,r dan t untuk penjelasan tentang x,y dan r tetap sama seperti penjelasan Tabel 4.2 hanya saja pada penutupan citra ini memerlukan t yaitu *thickness* dengan tujuan untuk melebarkan lubang tengah pada lingkaran hitam yang terbentuk. Bentuk aslinya dapat di ilustrasikan seperti donat terdapat lubang di tengah lingkaran. Dan hal ini tidak diperlukan pada cara pembuatan lingkaran putih sebelumnya. Karena penutupan tengah citra membutuhkan lingkaran putih yang utuh, sedangkan penutupan pada samping citra

membutuhkan lingkaran hitam yang memiliki lubang ditengahnya. Gambar 4.1. merupakan gambar sebelum penyesuaian dan setelah penyesuaian frame.



Gambar 4.1 (a) Sebelum penyesuaian (b) Setelah penyesuaian

(Sumber : Olahan sendiri)

4.2 Pengujian Program HSV Bola dan Lapangan

Pengujian HSV Bola dan Lapangan untuk melakukan seleksi warna objek dengan warna benda yang ada disekitarnya. Pada pengujian ini warna yang diseleksi adalah warna orange dan hijau.

4.2.1 Tujuan Pengujian Program HSV Bola dan Lapangan

Tujuan dari pengujian ini untuk mengetahui proses seleksi warna pada bola berwarna orange dan hijau dengan baik.

4.2.2 Alat yang Digunakan pada Pengujian Program HSV Bola dan Lapangan

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain :

1. Laptop.
2. OpenCV.
3. OpenFrameworks.
4. Visual Studio 2015.
5. Kamera Katadioptrik.

4.2.3 Prosedur Pengujian Program HSV Bola dan Lapangan

Langkah-langkah untuk melakukan pengujian Program HSV Bola dan Lapangan sebagai berikut :

1. Hubungkan kamera katadioptrik ke laptop yang akan digunakan.
2. Buka aplikasi Visual Studio 2015 yang merupakan lembar kerja dari Program penelitian ini.
3. Jalankan program utama yang telah dibuat.
4. Pada trackbar H+, H-, S+, S-, V+, V- ball dan field dapat diatur dengan kebutuhan yang ada untuk dapat menyesuaikan warna bola berwarna orange dan lapangan berwarna hijau.
5. Nilai trackbar akan tersimpan secara otomatis ke dalam folder data berekstensi setting_katadioptrik.dat dan setting_katadioptrik_obstacle.dat

4.2.4 Hasil Pengujian Program HSV Bola dan Lapangan

Tampilan hasil HSV pada bola berwarna orange dan lapangan berwarna hijau yang telah mengalami proses dari HSV. Dapat dilihat pada Tabel 4.3 merupakan nilai HSV dari Bola berwarna orange dan mampu untuk menangkap object warna berwarna saja. Dari hasil pengolahan tersebut, jika jarak bola kurang dari 400 cm, maka prosentase terdeteksi sebesar 80%. Berarti jarak ini dapat digunakan sebagai acuan untuk mengambil data saat proses ini. Pada Tabel 4.4 merupakan nilai HSV Lapangan berwarna hijau dan nilai tersebut mampu untuk menangkap warna hijau saja. Hasil pada pengujian dapat dilihat pada Gambar 4.2 (a) untuk hasil bola dan (b) untuk hasil lapangan.



Tabel 4.3 Nilai HSV Bola

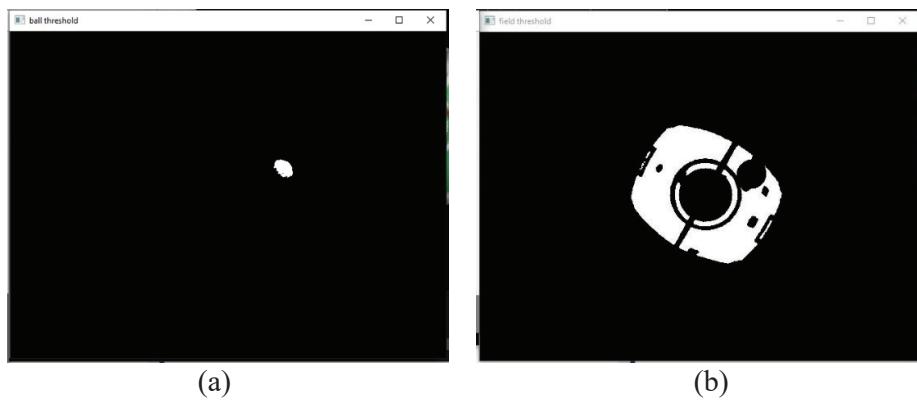
| No | Nilai HSV Bola | | | | | | Akurasi deteksi (%) | |
|-----|----------------|----|-----|-----|-----|-----|---------------------|--------------------|
| | H- | H+ | S- | S+ | V- | V+ | Jarak bola < 400cm | Jarak bola > 400cm |
| 1. | 0 | 33 | 133 | 255 | 255 | 255 | 100% | 0% |
| 2. | 0 | 31 | 130 | 255 | 255 | 255 | 100% | 0% |
| 3. | 0 | 30 | 131 | 255 | 255 | 255 | 100% | 0% |
| 4. | 0 | 20 | 131 | 255 | 255 | 255 | 100% | 0% |
| 5. | 0 | 20 | 128 | 255 | 255 | 255 | 100% | 0% |
| 6. | 0 | 20 | 110 | 255 | 255 | 255 | 100% | 0% |
| 7. | 0 | 20 | 130 | 255 | 255 | 255 | 100% | 0% |
| 8. | 0 | 20 | 109 | 255 | 255 | 255 | 100% | 0% |
| 9. | 0 | 10 | 105 | 255 | 255 | 255 | 0% | 100% |
| 10. | 0 | 9 | 100 | 255 | 255 | 220 | 0% | 100% |

(Sumber : Olahan sendiri)

Tabel 4.4 Nilai HSV Lapangan

| No | Nilai HSV Lapangan | | | | | | Akurasi deteksi (%) |
|-----|--------------------|----|----|-----|-----|-----|---------------------|
| | H+ | H- | S+ | S- | V+ | V- | |
| 1. | 38 | 96 | 73 | 250 | 155 | 255 | 100% |
| 2. | 38 | 96 | 50 | 250 | 154 | 255 | 100% |
| 3. | 38 | 96 | 80 | 250 | 155 | 255 | 100% |
| 4. | 38 | 96 | 73 | 250 | 153 | 255 | 100% |
| 5. | 38 | 96 | 70 | 250 | 152 | 255 | 100% |
| 6. | 38 | 96 | 51 | 250 | 155 | 255 | 100% |
| 7. | 38 | 96 | 52 | 250 | 150 | 255 | 100% |
| 8. | 38 | 96 | 45 | 250 | 152 | 255 | 100% |
| 9. | 38 | 96 | 10 | 250 | 151 | 255 | 100% |
| 10. | 38 | 96 | 15 | 250 | 140 | 255 | 100% |

(Sumber : Olahan sendiri)



Gambar 4.2 HSV Bola (a) dan HSV Lapangan (b)

(Sumber : Olahan sendiri)

Perubahan yang terjadi pada Nilai HSV bola maupun Nilai HSV lapangan selalu mengalami perubahan agar mendapatkan nilai yang maksimal. Intensitas cahaya yang ditangkap pada kamera juga mempengaruhi perubahan nilai HSV masing-masing objek yang ditentukan sebelumnya. Dapat dikatakan perubahan waktu pagi menuju siang mempengaruhi intensitas cahaya yang menyebabkan sistem yang telah dibuat harus menyesuaikan kembali.

4.3 Pengujian Program Morfologi Bola dan Lapangan

Pengujian Program Morfologi Bola dan Lapangan untuk menyesuaikan hasil citra lebih baik. Pengujian dilakukan setelah proses HSV Bola dan HSV Lapangan.

4.3.1 Tujuan Pengujian Program Morfologi Bola dan Lapangan

Tujuan dari pengujian ini untuk memperbaiki hasil citra yang dihasilkan oleh HSV bola dan HSV Lapangan. Ditujukan untuk menambah citra yang berlubang dan mengurangi citra data biner yang belum halus.

4.3.2 Alat yang Digunakan pada Pengujian Program Morfologi Bola dan Lapangan

1. Kamera Katadioptrik.
2. Laptop yang terpasang Visual Studio 2015.
3. Lapangan berwarna Hijau.
4. Bola berwarna orange.

4.3.3 Prosedur Pengujian Program Morfologi Bola dan Lapangan

1. Pastikan kamera katadioptrik telah terhubung pada Laptop yang akan digunakan.
2. Program omnivision pada Visual Studio 2015.
3. Jalankan program.
4. Sesuaikan citra kamera katadioptrik.
5. Kalibrasi Nilai HSV Bola dan Lapangan.
6. Kalibrasi Nilai Morfologi Bola dan Lapangan.



4.3.4 Hasil Pengujian Program Morfologi Bola dan Lapangan

Pada Tabel 4.5 merupakan nilai dari morfologi bola dan lapangan. Didapat nilai blur sebesar 3 dikarenakan saat citra mengalami proses blur dapat membantu untuk menghaluskan hasil citra. Pada +ball bernilai 1 tandanya kernel ditambah dengan ketebalan 1 piksel dan mengalami pengurangan 2 piksel pada -ball. Terjadi hal yang sama juga pada +field dan -field.

Tabel 4.5 Nilai Morfologi bola dan lapangan

| Nilai Morfologi Bola dan Lapangan | | | | |
|-----------------------------------|-------|-------|--------|--------|
| Blur | +ball | -ball | +field | -field |
| 3 | 1 | 2 | 3 | 4 |

(Sumber : Olahan sendiri)

4.4 Pengujian Program Jarak dalam nilai piksel

Pengujian Program Jarak dalam nilai piksel untuk mengetahui jarak bola terhadap robot dalam satuan nilai piksel.

4.4.1 Tujuan Pengujian Program Jarak dalam nilai piksel

Pengujian ini ditujukan untuk dapat mengetahui persamaan nilai antara nilai jarak dalam piksel dengan jarak sesungguhnya.

4.4.2 Alat yang Digunakan pada Pengujian Program Jarak dalam nilai piksel

1. Kamera Katadioptrik.
2. Visual Studio 2015.
3. Program HSV, Kalibrasi, dan Morfologi.
4. Program Euclidean pada Visual Studio.

4.4.3 Prosedur Pengujian Program Jarak dalam nilai piksel

1. Pastikan kamera katadioptrik telah terpasang pada Laptop.
2. Jalankan program Euclidean pada Visual Studio 2015.

3. Meteran.
4. Bola berwarna Orange.
5. Lapangan sepak bola berwarna hijau.

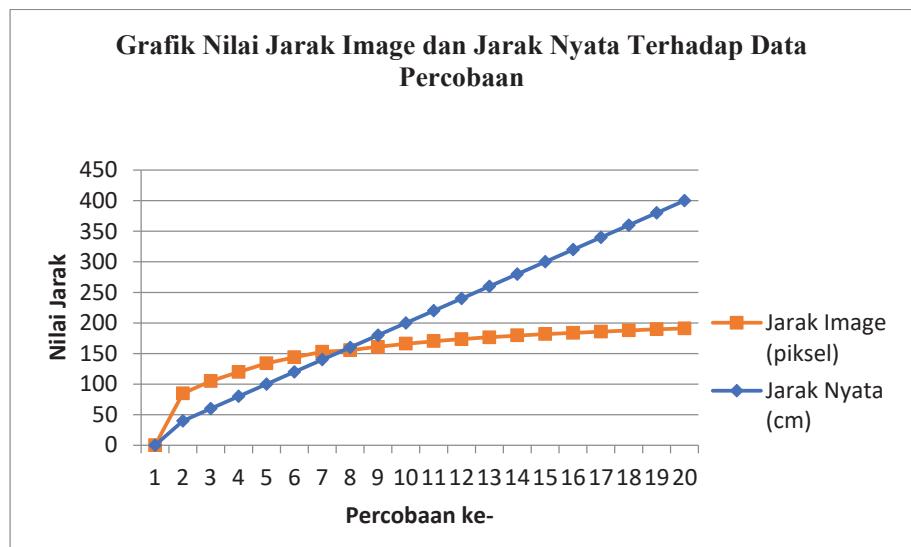
4.4.4 Hasil Pengujian Program Jarak dalam nilai piksel

Berikut ini adalah hasil pengujian, untuk melihat kesesuaian antara jarak nyata dengan piksel. Telah dilakukan 20 kali percobaan mulai dari piksel 85 dan pada jarak nyata sejauh 40 cm. Percobaan ini dilakukan dari nilai piksel 85 sampai dengan nilai piksel 191, dimana jarak nyata yang dihasilkan sejauh 400 cm.

Tabel 4.6 Nilai persamaan jarak piksel dengan cm

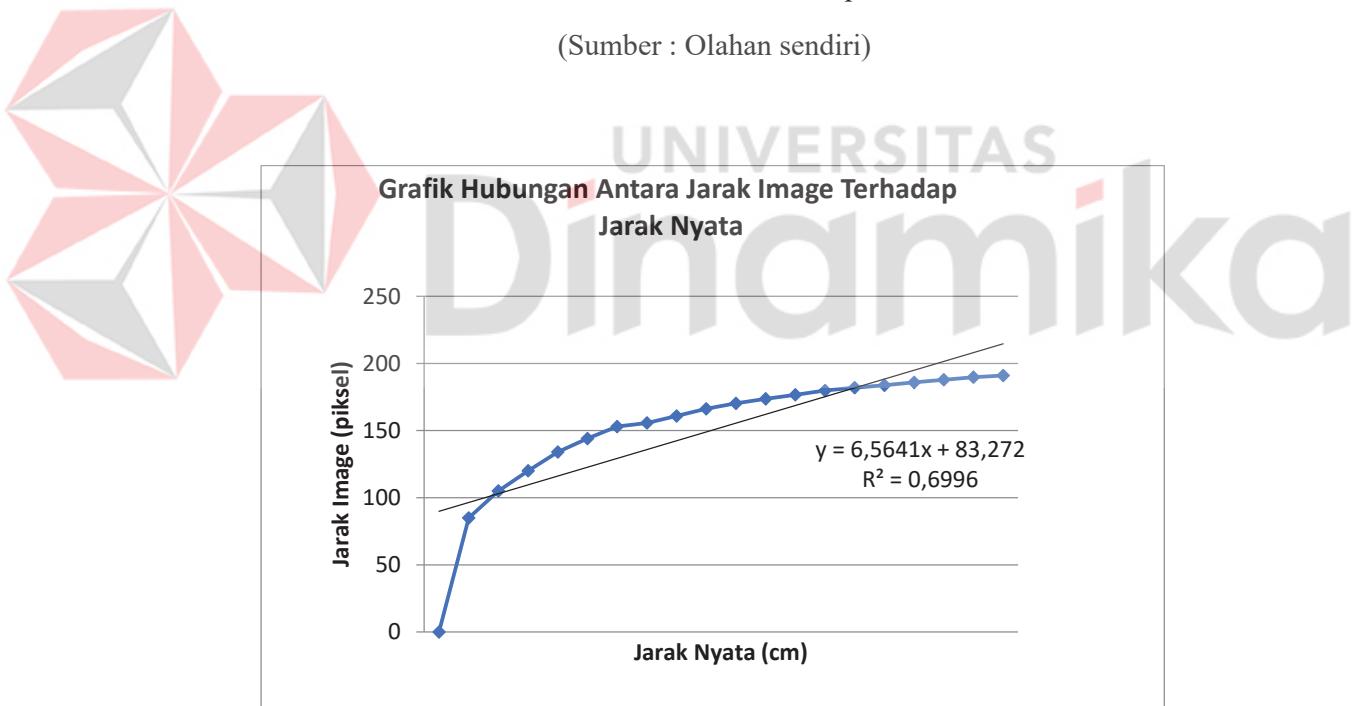
| No | Jarak <i>image</i> (piksel) | Jarak nyata (cm) | Error (%) |
|-----------------|--------------------------------|---------------------|--------------|
| 1 | 0 | 0 | 0 |
| 2 | 85,00 | 40 | 112,50 |
| 3 | 105,00 | 60 | 75,00 |
| 4 | 120,00 | 80 | 50,00 |
| 5 | 134,00 | 100 | 34,00 |
| 6 | 144,00 | 120 | 20,00 |
| 7 | 153,00 | 140 | 9,28 |
| 8 | 155,63 | 160 | 2,73 |
| 9 | 160,79 | 180 | 10,67 |
| 10 | 166,20 | 200 | 16,90 |
| 11 | 170,30 | 220 | 22,59 |
| 12 | 173,66 | 240 | 27,64 |
| 13 | 176,63 | 260 | 32,07 |
| 14 | 179,74 | 280 | 35,81 |
| 15 | 181,86 | 300 | 39,38 |
| 16 | 183,70 | 320 | 42,59 |
| 17 | 185,82 | 340 | 45,35 |
| 18 | 187,80 | 360 | 47,83 |
| 19 | 189,78 | 380 | 50,06 |
| 20 | 191,00 | 400 | 52,25 |
| Error rata-rata | | | 36.33 |

(Sumber : Olahan sendiri)



Gambar 4.3 Grafik Nilai Jarak Terhadap Data Percobaan

(Sumber : Olahan sendiri)



Gambar 4.4 Grafik Hubungan Antara Jarak Image Terhadap Jarak Nyata

(Sumber : Olahan sendiri)

Pada Gambar 4.3 menandakan bahwa nilai piksel jarak pada kamera tidak linier dengan nilai jarak sesungguhnya dalam satuan cm. Ketidaklinieran ini ditunjukkan oleh Gambar 4.4, dimana pada dua nilai berbeda tersebut memiliki nilai regresi $R^2 = 0.6996$ mengacu pada persamaan 2.28, semakin tinggi nilai regresi (mendekati nilai 1), grafik yang dihasilkan akan semakin memiliki hubungan yang linier dan kuat (Pratomo, 2015). Maka, perlu adanya perhitungan secara kompleks untuk mengatasi permasalahan tersebut dengan ANN.

4.5 Pengujian Program Jarak ANN dengan cm

Pengujian Program jarak ANN dengan cm untuk melakukan prediksi jarak ANN dengan nilai sesungguhnya.

4.5.1 Tujuan Pengujian Program Jarak ANN dengan cm

Tujuan pengujian ini ditujukan agar robot dapat mengetahui atau memprediksi nilai citra yang sesuai dengan jarak sesungguhnya yaitu dalam satuan cm bukan lagi dalam satuan piksel.

4.5.2 Alat yang Digunakan pada Pengujian Program Jarak ANN dengan cm

1. Kamera Katadioptrik
2. Laptop dan software Visual Studio 2015
3. Program ANN pada Visual Studio.
4. Bola dan Lapangan.
5. Meteran.

4.5.3 Prosedur Pengujian Program Jarak ANN dengan cm

1. Pastikan Kamera katadioptrik telah terhubung pada laptop.
2. Jalankan Program ANN pada Visual Studio 2015.
3. Lakukan kalibrasi nilai dengan jarak piksel dengan jarak sesungguhnya.

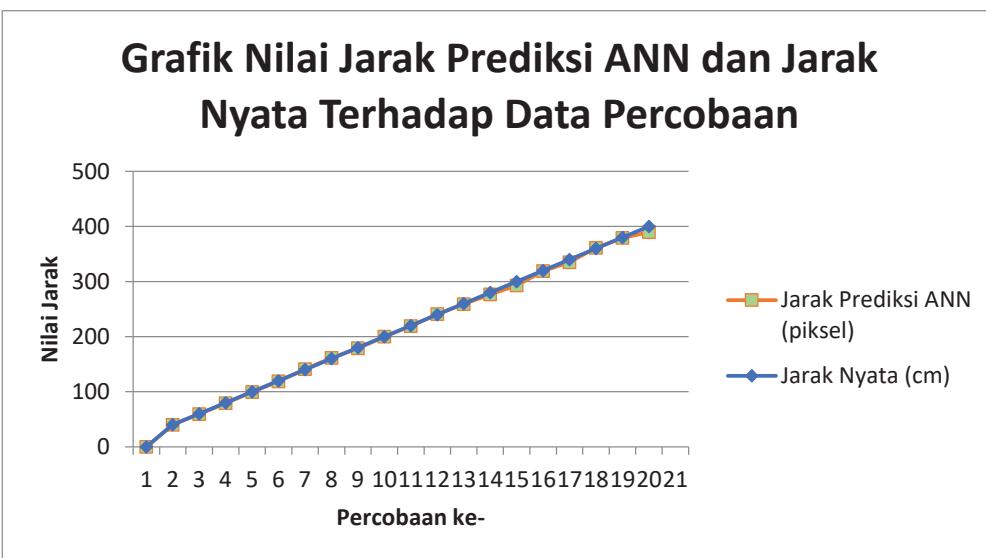
4.5.4 Hasil Pengujian Program Jarak ANN dengan cm

Berikut ini adalah hasil pengujian, untuk melihat kesesuaian antara jarak nyata dengan piksel, setelah dilakukan penyesuaian dengan ANN. Telah dilakukan 20 kali percobaan mulai dari piksel 39,77 dan pada jarak nyata sejauh 40 cm. Percobaan ini dilakukan dari nilai piksel 39,77 sampai dengan nilai piksel 389,54, dimana jarak nyata yang dihasilkan sejauh 400 cm.

Tabel 4.7 Hasil nilai ANN dengan cm

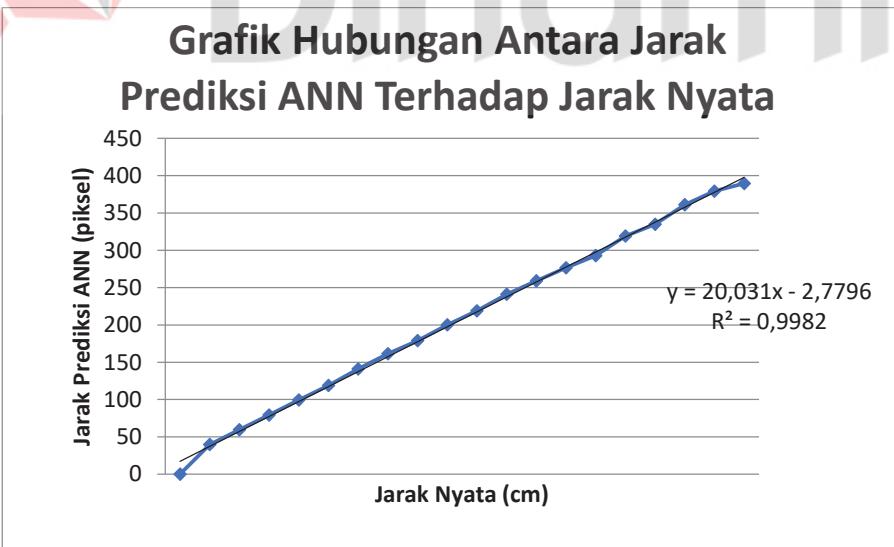
| No | Jarak prediksi ANN (piksel) | Jarak Nyata (cm) | error (%) |
|-----------------|-----------------------------|------------------|-----------|
| 1. | 0 | 0 | 0 |
| 2. | 39,77 | 40 | 0,58 |
| 3. | 59,57 | 60 | 0,71 |
| 4. | 79,31 | 80 | 0,86 |
| 5. | 99,51 | 100 | 0,49 |
| 6. | 118,92 | 120 | 0,9 |
| 7. | 140,99 | 140 | 0,71 |
| 8. | 161,34 | 160 | 0,834 |
| 9. | 179,01 | 180 | 0,55 |
| 10. | 200,11 | 200 | 0,06 |
| 11. | 218,98 | 220 | 4,46 |
| 12. | 241,02 | 240 | 0,43 |
| 13. | 258,97 | 260 | 0,39 |
| 14. | 276,66 | 280 | 1,19 |
| 15. | 292,98 | 300 | 2,34 |
| 16. | 319,12 | 320 | 0,275 |
| 17. | 334,88 | 340 | 1,51 |
| 18. | 361,09 | 360 | 0,30 |
| 19. | 379,18 | 380 | 0,22 |
| 20. | 389,54 | 400 | 2,62 |
| Error rata-rata | | | 0,97 |

(Sumber : Olahan sendiri)



Gambar 4.4 Grafik Nilai Jarak Prediksi ANN dan Jarak Nyata Terhadap Data Percobaan

(Sumber : Olahan sendiri)



Gambar 4.5 Grafik Hubungan Nilai Jarak Prediksi ANN Terhadap Jarak Nyata

(Sumber : Olahan sendiri)

Pada Gambar 4.4 menunjukkan bahwa ANN mampu membantu untuk memprediksi nilai jarak piksel ke cm menjadi lebih baik. Dimana, awal data nilai jarak piksel tidak linier dengan nilai sesungguhnya, hal ini didukung oleh Gambar 4.5, dimana dari hasil persamaan memberikan nilai regresi linier $R^2 = 0.9982$, nilai ini mendekati 1 yang berarti mendekati linier. Perhitungan prediksi ANN membantu untuk melinierkan nilai piksel dengan jarak sesungguhnya dengan tingkat keberhasilan penyesuaian sebesar 99,03 %.

4.6 Pengujian Waktu pada Deteksi Bola dan Prediksi Jarak Bola

Suatu proses pengujian untuk dapat menganalisa durasi waktu dalam melakukan proses deteksi bola dan prediksi jarak bola.

4.6.1 Tujuan Pengujian Waktu pada Deteksi Bola dan Prediksi Jarak Bola

Tujuan pengujian waktu pada Deteksi bola untuk mengetahui lama proses yang dibutuhkan untuk melakukan deteksi bola. Pada pengujian waktu pada prediksi jarak bola untuk mengetahui lama proses yang dijalankan dalam melakukan sebuah prediksi jarak bola.

4.6.2 Alat yang Digunakan Pengujian Waktu pada Deteksi Bola dan Prediksi

Jarak Bola

Alat yang digunakan untuk dapat melakukan pengujian sebagai berikut :

1. Kamera katadioptrik.
2. Laptop yang telah terinstal Visual Studio 2015.
3. Program Visual Studio 2015.
4. Bola berwarna *orange*.

4.6.3 Prosedur Pengujian Waktu pada Deteksi Bola dan Prediksi Jarak Bola

1. Pastikan kamera katadioptrik telah terhubung pada Laptop.
2. Jalankan program utama pada Visual Studio 2015.
3. Lakukan penyesuaian citra agar bola dapat terdeteksi.
4. Informasi data waktu deteksi bola dan prediksi jarak bola dapat dilihat pada terminal Visual Studio 2015.
5. Pada terminal terdapat data waktu *second*, *millisecond*, *microsecond* dan *nanosecond*.

4.6.4 Hasil Pengujian Waktu pada Deteksi Bola dan Prediksi Jarak Bola

Berikut ini adalah hasil dari Analisa data waktu dalam proses deteksi bola dan prediksi bola. Pada Tabel 4.8 merupakan data waktu dalam satuan *second(s)*, *millisecond(ms)*, *microsecond(us)* dan *nanosecond(ns)*. Pada system tergolong berjalan sangat cepat dalam melakukan proses deteksi dikarenakan baru dapat menginformasikan nilai waktu mulai pada satuan *microsecond*.

Tabel 4.8 Pengujian Waktu Deteksi dan Prediksi Bola

| No | Waktu Deteksi Bola | Waktu Prediksi Jarak Bola |
|----|------------------------|---------------------------|
| | <i>Nanosecond (ns)</i> | <i>Nanosecond (ns)</i> |
| 1 | 21900 | 50400 |
| 2 | 18200 | 65700 |
| 3 | 31100 | 52000 |
| 4 | 34100 | 51500 |
| 5 | 17100 | 58500 |
| 6 | 30900 | 56000 |
| 7 | 28900 | 68300 |
| 8 | 41300 | 87600 |

| No | Waktu Deteksi Bola | Waktu Prediksi Jarak Bola |
|-----------|--------------------|---------------------------|
| | Nanosecond (ns) | Nanosecond (ns) |
| 9 | 45000 | 61500 |
| 10 | 45700 | 95300 |
| 11 | 36100 | 73700 |
| 12 | 48000 | 63000 |
| 13 | 39700 | 61400 |
| 14 | 33900 | 92900 |
| 15 | 45000 | 57900 |
| 16 | 43100 | 76100 |
| 17 | 42200 | 74800 |
| 18 | 39100 | 72000 |
| 19 | 25600 | 77500 |
| 20 | 36000 | 57900 |
| Rata-rata | 35145 | 67700 |

Sumber : (Olahan pribadi)

Pada Tabel 4.8 diatas dapat disimpulkan bahwa waktu deteksi bola mengalami durasi yang lebih sedikit daripada durasi prediksi bola. Didapatkan pada tabel nilai rata-rata deteksi bola sebesar 35145 ns, sedangkan nilai rata-rata prediksi bola sebesar 67700 ns.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian diperoleh hasil sebagai berikut :

1. Tahapan awal untuk melakukan pendektsian bola berwarna orange maupun lapangan berwarna hijau adalah pengkonversian citra warna RGB ke citra warna HSV.
2. Proses pendektsian bola berwarna orange di atas lapangan berwarna hijau dengan perbandingan citra biner kedua objek menggunakan minEnclosingCircle dengan melihat contours disekitar objek bola berwarna orange.
3. Proses morfologi closing dan opening merupakan metode untuk memperbaiki citra biner bola maupun lapangan.
4. Proses pengukuran jarak dua objek dalam sebuah piksel menggunakan metode *Euclidean distance*.
5. Hasil pengujian pendektsian bola berwarna orange dengan hasil visual yang baik adalah pada jarak 40 cm sampai 400 cm. Saat jarak lebih dari 400 cm nilai HSV pada bola perlu mengalami perubahan.
6. Telah dilakukan pengujian tanpa metode ANN dengan tingkat keberhasilan 63,67%, dan tingkat kelinieran $R^2 = 0.6996$. Hasil pengujian prediksi jarak piksel dengan jarak sesungguhnya menggunakan pelatihan Matlab dengan 1.000.000 iterasi dan mendapatkan nilai error sebesar 0,054969%. Tingkat



UNIVERSITAS
Dinamika

keberhasilan penyesuaian setelah menggunakan metode ANN sebesar 99,03 %, dengan tingkat kelinieran sebesar $R^2 = 0.9982$.

7. Hasil pengujian waktu dalam satu proses berjalan-nya program deteksi bola dan prediksi bola memiliki rata-rata 35145 ns untuk proses waktu deteksi bola. Sedangkan proses waktu untuk prediksi jarak bola memiliki rata-rata sebesar 67700 ns. Nilai waktu yang didapat, disimpulkan bahwa durasi waktu deteksi bola lebih singkat atau kecil jika dibandingkan dengan durasi waktu prediksi jarak bola.

5.2 Saran

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

1. Peneliti berikutnya diharapkan mengembangkan program Kecerdasan Buatan yang lainnya untuk meningkatkan kecepatan proses pembelajaran sekitar dan memperbanyak objek yang akan diajarkan. Sebagai contoh menerapkan metode *Deep Learning*.
2. Peneliti berikutnya diharapkan menggunakan kamera yang lebih luas pada jangkauan tangkapan. Dikarenakan pada kamera yang sebelumnya pada jangkauan tertentu menjadi buram sehingga mengganggu proses pendektsian.
3. Peneliti berikutnya diharapkan mengembangkan program untuk dapat menyesuaikan perubahan intensitas cahaya.

DAFTAR PUSTAKA

- Akbar, E. I. (2017). *Rancang Bangun Omni Vision pada Robot ESPARTAN KRSBI Beroda*, 78-81.
- Dr.Ir.Saludin Muis, M. (2017). *Sistem Kecerdasan Tiruan Dengan Kemampuan Belajar dan Adaptasi*. Ruko Jambusari 7A Yogyakarta: TEKNOSAIN.
- Muhammad, F. A. (2015). *Sistem Navigasi Robot Beroda Menggunakan Omnidirectional Camera*, 1-5.
- Prakoso, Y., R, Rachmawan., M, Muhammad Abizhar., S, Alam Ar Raad., R, Aulia Aditya., D, Rudy., I, Muhammad Azhar., A, Helga Dendy., A Kamal. 2017. Artifical Neural Network Untuk Pengukuran Posisi Bola Menggunakan Kamera Omnidireksional Pada Robot Sepak Bola Beroda. *5th Indonesian Symposium on Robotic Systems and Control*. 169-172. Jakarta. Indonesia.
- Pratomo, DS, Erna, DA. (2015). *Analisis Regresi Dan Korelasi Antara Pengunjung Dan Pembeli Terhadap Nominal Pembelian Di Indomaret Kedungmundu Semarang Dengan Metode Kuadrat Terkecil*. Jurnal Statistika, 1-12.
- Setyawan, N. (2018). *Pengenalan Posisi Objek Dari Sistem Kamera Omni-Vision Menggunakan Pso-Nn Dan Scan Lines pada Robot Sepak Bola*, 27-33.
- Stone, A. A. (2017). *Segmentasi Citra pada Robot Sepak Bola Beroda Menggunakan Multilayer Neural Network dan Fitur Warna HSV*, A276-A281.
- Yuliara, I Made. (2016). *Modul Regresi Linier Sederhana*, Jurusan Fisika Fakultas MIPA, Universitas Udayana: 2-5.