



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA:  
PENDETEKSIAN WARNA DAN TAMPILAN INFORMASI  
WARNA DENGAN *PLATFORM .NET* DAN *EMGUCV  
LIBRARY***

**SKRIPSI**

**BURHAN ADI WICAKSANA  
0706275952**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER  
DEPOK  
JULI 2011**



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA:  
PENDETEKSIAN WARNA DAN TAMPILAN INFORMASI  
WARNA DENGAN *PLATFORM .NET* DAN *EMGUCV  
LIBRARY***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana**

**BURHAN ADI WICAKSANA**

**0706275952**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER  
DEPOK  
JULI 2011**

i

**Universitas Indonesia**

---

**HALAMAN PERNYATAAN ORISINALITAS**

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Burhan Adi Wicaksana  
NPM : 0706275952  
Tanda Tangan :   
Tanggal : 11 Juli 2011

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Burhan Adi Wicaksana  
NPM : 0706275952  
Program Studi : Teknik Komputer  
Judul Skripsi : Implementasi Sistem Bantuan Penderita Buta Warna:  
Pendeteksian Warna, dan Tampilan Informasi Warna  
dengan *Platform .NET* dan *EmguCV Library*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc. (RFS)

Penguji : Prof. Dr.-Ing. Ir. Kalamullah Ramli, M. Eng. (KR)

Penguji : Prima Dewi Purnamasari, S.T., M.T., M.Sc. (PDP)

Ditetapkan di : Depok  
Tanggal : 11 Juli 2011

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

1. Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc. selaku pembimbing akademis dan dosen pembimbing atas segala bimbingan, ilmu, dan arahan baik dalam penulisan skripsi maupun selama masa studi di Teknik Komputer.
2. Bapak, Ibu, dan adik-adik yang selalu menjadi sumber inspirasi dan semangat.
3. Teman-teman di program studi Teknik Komputer, Teknik Elektro, dan Universitas Indonesia atas segala dukungan dan kerja samanya.

sehingga penulisan skripsi ini dapat diselesaikan dengan baik.

Depok, 11 Juli 2011

Penulis,



Burhan Adi Wicaksana  
NPM.0706275952

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Burhan Adi Wicaksana

NPM : 0706275952

Program Studi : Teknik Komputer

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis karya : Tugas akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA:  
PENDETEKSIAN WARNA, DAN TAMPILAN INFORMASI WARNA  
DENGAN PLATFORM .NET DAN EMGUCV LIBRARY**

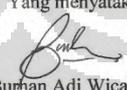
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal: 11 Juli 2011

Yang menyatakan,

  
(Burhan Adi Wicaksana)

## ABSTRAK

Nama : Burhan Adi Wicaksana  
Program Studi : Teknik Komputer  
Judul : Implementasi Sistem Bantuan Penderita Buta Warna :  
Pendeteksian Warna, dan Tampilan Informasi Warna  
dengan *Platform .NET* dan *EmguCV Library*  
Pembimbing : Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc.

Buta warna merupakan salah satu penyakit yang disebabkan karena faktor keturunan yang mengakibatkan penderitanya tidak dapat mengenali warna, baik warna tertentu (parsial) atau semua warna (total). Ketidakmampuan mengenali warna ini berpotensi menyebabkan berbagai kesulitan bagi para penderitanya dalam kehidupan sehari-hari maupun dalam lingkup yang lebih khusus, karena banyak hal dalam kehidupan sehari-hari yang diasosiasikan dengan warna. Untuk mengatasi masalah tersebut, dirancang sebuah sistem bantuan untuk penderita buta warna dalam mengenali warna. Sistem ini dirancang dalam bentuk aplikasi yang dikembangkan untuk sistem tertanam (*embedded system*) dan Windows Phone 7 dengan menerapkan konsep *augmented reality*. Perancangan modul pendeteksian warna yang dibuat diimplementasikan dengan *.NET Framework* dan *EmguCV Library*. Hasil implementasi berupa modul pendeteksian warna yang terintegrasi ke dalam sistem bantuan buta warna Chromophore sebagai aplikasi untuk sistem tertanam dan aplikasi *mobile* pada platform Windows Phone 7. Hasil pengujian pada sistem tertanam menunjukkan bahwa modul dapat mengenali sampel warna yang ada dengan persentase sebesar 69.33% dengan model warna HSV dan 90.67% dengan model warna HLS. Pengujian pada aplikasi *mobile* menunjukkan kesuksesan pendeteksian warna sampel sebesar 11.33% dengan model warna RGB dan 95.33% dengan model warna HLS.

Kata kunci: *buta warna, augmented reality, embedded system, pengenalan warna.*

## ABSTRACT

Name : Burhan Adi Wicaksana  
Study Program : Computer Engineering  
Title : Implementation of Color Blind Aid System : Color Detection, and Color Information Display Using .NET Platform and EmguCV Library  
Supervisor : Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc.

Color blindness is a form of generative deficiency where the patient loses the ability to recognize color, either particular color (partial color blind) or the whole color (total color blind). The disability of the patient to recognize color is potential to cause problems to the patient in daily life or in more specific area. To help the patient cope with the problem in recognizing colors, a color blindness helping system is designed. This system is designed for embedded system and Windows Phone 7 using the concept of augmented reality. This work proofs design and implementation of color detection module using .NET Framework and EmguCV Library. In this work we have implemented an integrated color detection module integrated in the Chromophore color blind aid system in the form of application for embedded system and mobile application for Windows Phone 7. Testing result on embedded system shows that the module is able to detect color samples with percentage of 69.33% using HSV color model, and 90.67% using HLS color model. Testing on mobile application resulted that the module is able to detect color samples with percentage of 11.33% using RGB color model and 95.33% using HLS color model.

Keywords: *color blindness, augmented reality, embedded system, color detection.*

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
LEMBAR PENGESAHAN .....	iii
UCAPAN TERIMA KASIH.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	v
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penulisan .....	2
1.4 Batasan Masalah.....	2
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	3
<b>2. AUGMENTED REALITY DAN SISTEM BANTUAN PENGENALAN WARNA UNTUK BUTA WARNA.....</b>	<b>5</b>
2.1 Buta Warna Total Dan Buta Warna Parsial.....	5
2.1.1 Buta Warna Total.....	7
2.1.1 Buta Warna Parsial .....	7
2.2 Teknologi Augmented Reality .....	8
2.3 Pengolahan Citra Untuk Pengenalan Warna .....	10
2.3.1 Pengolahan Citra untuk Pengenalan Warna .....	10
2.3.2 Model Warna RGB, HSV, dan HLS.....	12
2.4 <i>.Net Framework</i> , Bahasa Pemrograman C#, dan Emgucv Library Serta Windows Phone 7 Sebagai Salah Satu <i>Platform</i> Dalam Implementasi Sistem.....	13
2.4.1 NET Framework dan Perkembangannya.....	13
2.4.2 Bahasa Pemrograman C#.....	16
2.4.3 OpenCV dan EmguCV Library .....	17
2.4.4 Windows Phone 7.....	19

<b>3. PERANCANGAN MODUL PENDETEKSIAN WARNA DAN TAMPILAN INFORMASI WARNA .....</b>	<b>21</b>
3.1 Rancangan Cara Kerja Program .....	21
3.1.1 Deskripsi .....	21
3.1.2 Diagram Alir Cara Kerja Program .....	24
3.2 Diagram-Diagram Unified Modelling Language (UML).....	27
3.2.1 Use Case Diagram .....	27
3.2.2 Activity Diagram .....	28
3.2.3 Sequence Diagram .....	30
3.2.4 Class Diagram.....	31
3.2.5 Deployment Diagram.....	32
<b>4. IMPLEMENTASI MODUL DAN INTEGRASI SISTEM BANTUAN PENDERITA BUTA WARNA .....</b>	<b>33</b>
4.1 Implementasi Modul pada Aplikasi Untuk Sistem Tertanam .....	33
4.1.1 Perangkat Pendukung Pengembangan Modul .....	33
4.1.2 Pemrograman Modul Pendeteksian Warna dan Penjelasan Program ...	34
4.1.3 Hasil Implementasi Modul dan Penggunaan Modul.....	36
4.1.4 Integrasi Modul.....	38
4.1.5 Hasil Integrasi Sistem Bantuan Penderita Buta Warna .....	38
4.2 Implementasi Modul Pada Aplikasi <i>Mobile</i> Pada Platform Windows Phone 7.....	39
4.2.1 Instalasi Modul-Modul Pendukung Pengembangan Aplikasi Pada Platform Windows Phone 7 .....	39
4.2.2 Pemrograman Modul Dan Penjelasan Kode Program .....	40
4.2.3 Hasil Implementasi Modul.....	42
4.2.4 Integrasi Modul ke Dalam Sistem .....	42
4.2.5 Hasil Integrasi Sistem pada Windows Phone 7 .....	43
<b>5. PENGUJIAN DAN ANALISA MODUL PENDETEKSIAN WARNA .....</b>	<b>45</b>
5.1. Pengujian Fungsionalitas Modul Pendeteksian Warna .....	45
5.1.1 Pengujian Modul Untuk Sistem Tertanam.....	45
5.1.1.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.....	45
5.1.1.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan. ....	46

5.1.1.3 Pengujian pada Sampel Pengguna. ....	46
5.1.2 Pengujian Modul untuk Aplikasi Mobile .....	46
5.1.2.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna. ....	46
5.1.2.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan. ....	47
5.1.2.3 Pengujian Pada Sampel Pengguna. ....	47
5.2 Hasil Pengujian dan Analisis .....	47
5.2.1. Hasil Pengujian dan Analisis Modul untuk Sistem Tertanam .....	47
5.2.1.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.....	47
5.2.1.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan. ....	51
5.2.1.3 Pengujian Pada Sampel Pengguna. ....	55
5.2.2 Hasil Pengujian dan Analisis Modul Untuk Aplikasi Mobile .....	60
5.2.2.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.....	61
5.2.2.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan. ....	65
5.2.2.3 Pengujian pada Sampel Pengguna. ....	68
<b>6. PENUTUP.....</b>	<b>75</b>
<b>DAFTAR ACUAN.....</b>	<b>76</b>

## DAFTAR GAMBAR

Gambar 2.1. Panjang Gelombang yang Diterima Sel-Sel Kerucut [2] .....	6
Gambar 2.2. Vektor Pada Analisis CDV [2].....	6
Gambar 2.3. Piringan Tes Warna Ishihara [2] .....	7
Gambar 2.4. AR <i>Building Block</i> [8].....	9
Gambar 2.5. Pengenalan Warna dengan CECH [14].....	11
Gambar 2.6. Proses <i>managed compilation</i> oleh .NET Framework .....	14
Gambar 2.7. Arsitektur EmguCV [19].....	19
Gambar 3.2. Diagram Alir Cara Kerja Modul .....	23
Gambar 3.3. Diagram Alir Program dalam Sistem Bantuan Penderita Buta Warna .....	26
Gambar 3.4. <i>Use Case Diagram</i> .....	28
Gambar 3.5. <i>Activity Diagram</i> .....	29
Gambar 3.6. <i>Sequence Diagram</i> .....	30
Gambar 3.7. <i>Class Diagram</i> .....	31
Gambar 3.8. <i>Deployment Diagram</i> .....	32
Gambar 4.1. Algoritma dalam Implementasi Modul .....	34
Gambar 4.2. Fungsi <code>GetColor ()</code> pada Program.....	35
Gambar 4.3. Fungsi <code>DetectContour ()</code> pada Program.....	36
Gambar 4.4. Gambar Antarmuka Modul Pendeteksian Warna untuk Sistem Tertanam .....	37
Gambar 4.5. Gambar Modul Pendeteksian Warna Sedang Berjalan .....	37
Gambar 4.6. Modul Pendeteksian Warna yang Telah Terintegrasi ke dalam Sistem.....	39
Gambar 4.7.. Kode Program Modul Pendeteksian Warna pada Platform Windows Phone 7.....	41
Gambar 4.8. Antarmuka Modul Program pada Aplikasi <i>Mobile</i> .....	42
Gambar 4.9. Antarmuka Aplikasi <i>Mobile</i> Hasil Integrasi.....	43
Gambar 4.10. Modul Ishihara Test .....	44
Gambar 4.11. Modul Pendeteksian Warna Hasil Integrasi .....	44
Gambar 4.12. Modul Transformasi Warna Hasil Integrasi.....	44
Gambar 5.1. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HSV pada Modul Sistem Tertanam.....	50
Gambar 5.2. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS pada Modul Sistem Tertanam. ....	50
Gambar 5.3. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HSV pada Modul Sistem Tertanam dengan Variasi Pencahayaan.....	53
Gambar 5.4. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS pada Modul Sistem Tertanam dengan Variasi Pencahayaan. ....	54

Gambar 5.5. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Sistem Tertanam.....	57
Gambar 5.6. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Sistem Tertanam. ....	58
Gambar 5.7. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Buta Warna Parsial pada Sistem Tertanam.....	60
Gambar 5.8. Grafik Hasil Pengujian Pengalaman Pengguna Buta Warna Parsial pada Sistem Tertanam. ....	61
Gambar 5.9. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna RGB.....	64
Gambar 5.10. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS.....	64
Gambar 5.11. Grafik Pengujian Variasi Pencahayaan pada Modul Untuk Aplikasi <i>Mobile</i> dengan Model Warna RGB .....	67
Gambar 5.12. Grafik Pengujian Variasi Pencahayaan pada Modul Untuk Aplikasi <i>Mobile</i> dengan Model Warna HLS.....	68
Gambar 5.13. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Aplikasi <i>Mobile</i> .....	70
Gambar 5.14. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Aplikasi <i>Mobile</i> .....	71
Gambar 5.15. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Aplikasi <i>Mobile</i> .....	73
Gambar 5.16. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Aplikasi <i>Mobile</i> .....	73

## DAFTAR TABEL

Tabel 5.1. Hasil Pengujian Modul Untuk Sistem Tertanam Menggunakan Model Warna HSV .....	48
Tabel 5.2. Hasil Pengujian Modul Untuk Sistem Tertanam Menggunakan Model Warna HLS.....	49
Tabel 5.3. Hasil Pengujian Variasi Pencahayaan pada Modul Untuk Sistem Tertanam Menggunakan Model Warna HSV .....	52
Tabel 5.4. Hasil Pengujian Variasi Pencahayaan pada Modul Untuk Sistem Tertanam Menggunakan Model Warna HLS .....	52
Tabel 5.5. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Normal Terhadap Sampel Warna dalam Sistem Tertanam .....	56
Tabel 5.6. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Normal pada Sistem Tertanam.....	56
Tabel 5.7. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Buta Warna Parsial Terhadap Sampel Warna dalam Sistem Tertanam .....	59
Tabel 5.8. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Buta Warna Parsial pada Sistem Tertanam .....	59
Tabel 5.9. Hasil Pengujian Modul Untuk Aplikasi Mobile dengan Menggunakan Model Warna RGB .....	62
Tabel 5.10. Hasil Pengujian Modul Untuk Aplikasi Mobile dengan Menggunakan Model Warna HLS .....	63
Tabel 5.11. Hasil Pengujian Variasi Pencahayaan pada Modul Pendeteksian Warna Untuk Aplikasi Mobile dengan Model Warna RGB .....	66
Tabel 5.12. Hasil Pengujian Variasi Pencahayaan pada Modul Pendeteksian Warna Untuk Aplikasi Mobile dengan Model Warna HLS.....	66
Tabel 5.13. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Normal Terhadap Sampel Warna dalam Aplikasi <i>Mobile</i> .....	69
Tabel 5.14. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Normal pada Aplikasi Mobile.....	69
Tabel 5.15. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Buta Warna Parsial Terhadap Sampel Warna dalam Aplikasi <i>Mobile</i> .....	72
Tabel 5.16. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Buta Warna Parsial pada Aplikasi Mobile.....	72

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Buta warna (*color blindness*) merupakan penyakit yang banyak ditemukan kasusnya di dunia. Terdapat bermacam buta warna, yaitu buta warna total dan buta warna parsial. Kasus buta warna total termasuk jarang dan mengakibatkan penderitanya tidak dapat mengenali warna secara total. Buta warna parsial menyebabkan penderitanya kesulitan dalam mengenali warna, baik biru, merah, atau hijau. Warna tersebut berkaitan dengan sel kerucut yang berada pada sistem penglihatan manusia yang memiliki kemampuan untuk mengenali warna. Pengenalan atau persepsi terhadap warna juga dipengaruhi oleh tiga hal: *hue*, *saturation*, dan *brightness* [1]. *Hue* adalah perbedaan warna yang kontras, seperti merah dan hijau. *Saturation* merupakan perbedaan antara warna murni dan warna-warna campuran. Sedangkan *brightness* adalah intensitas terang-gelapnya cahaya pantulan dari objek.

Sebagian besar kasus buta warna ditemukan pada kaum pria, dengan persentase mencapai 7-10%, sementara pada wanita yang ditemukan kurang dari 1% [2]. Penyakit ini terjadi secara alami dan biasanya terjadi karena faktor keturunan. Dari sisi genetis, buta warna terjadi karena mutasi genetis pada kromosom-X. Pria hanya memiliki satu kromosom-X, sehingga bila satu kromosom ini mengalami mutasi, maka dia akan menjadi penderita buta warna. Di sisi lain, wanita memiliki lebih dari satu kromosom-X, sehingga bila satu kromosom mengalami mutasi, kromosom yang lain dapat mengatasi masalah yang terjadi. Karena hal inilah, kasus buta warna lebih banyak ditemukan pada pria, seperti telah disampaikan di atas.

Buta warna mengakibatkan penderitanya berkemungkinan mengalami kesulitan dalam berbagai hal, baik dalam kehidupan sehari-hari, maupun dalam lingkup lingkungan yang lebih khusus. Hal tersebut dikarenakan dalam kehidupan seringkali warna diasosiasikan dengan hal tertentu atau difungsikan sebagai penanda. Sebagai contoh, digunakannya warna sebagai penanda dalam rambu-rambu lalu lintas, lampu lalu lintas, penanda blok pada suatu lokasi, dan lain-lain.

Dalam lingkup yang lebih khusus, warna digunakan sebagai penanda untuk bahan kimia tertentu, penanda dalam komponen elektronik, dan lain sebagainya.

Kekurangan berupa buta warna tersebut akan menghambat seseorang dalam kehidupannya. Berawal dari hal inilah, dirancang sebuah sistem bantuan untuk penderita buta warna menggunakan teknologi perangkat *embedded* dan *augmented reality* untuk memberikan informasi warna dengan interaktif.

### **1.2 Rumusan Masalah**

Pada tulisan ini, permasalahan yang menjadi fokus adalah perancangan serta implementasi modul pendeteksian warna dan penambahan informasi menggunakan prinsip *augmented reality*. Perancangan dan implementasi dilakukan untuk sistem bantuan buta warna pada *embedded system*, serta sistem bantuan buta warna dalam bentuk aplikasi *mobile* pada platform Windows Phone 7.

### **1.3 Tujuan Penulisan**

Tujuan penelitian yang tertuang dalam penulisan skripsi ini adalah untuk membuat modul pendeteksian dan integrasi sistem bantuan untuk penderita buta warna. Sistem bantuan ini diharapkan dapat membantu penderita buta warna mengenali warna yang sesungguhnya, yang tidak dapat dikenali akibat buta warna, dengan menggunakan media informasi berupa teks dan suara. Interaktivitas dan fleksibilitas merupakan kunci yang ingin dicapai dalam perancangan dan implementasi sistem ini.

Perancangan sistem ini juga didasari dengan tujuan yang tercantum dalam *Millennium Development Goals* (MDGs) yang dirumuskan oleh PBB. Salah satu tujuan yang ada dalam MDGs tersebut adalah mewujudkan pendidikan yang universal bagi semua kalangan. Dengan adanya sistem bantuan ini diharapkan penderita buta warna mendapatkan kesempatan yang sama, baik dalam kehidupan sehari-hari, maupun dalam lingkup yang lebih khusus yang melibatkan warna sebagai unsur yang signifikan.

### **1.4 Batasan Masalah**

Permasalahan dalam penulisan skripsi ini dibatasi pada perancangan dan implementasi modul pendeteksian warna, serta integrasinya pada sistem bantuan

penderita buta warna pada *embedded system* serta aplikasi *mobile* pada *platform* Windows Phone 7.

Bagian lain dari sistem ini, yaitu pengembangan perangkat *embedded system* dan modul *finger detection* dibahas pada [4]. Penyediaan antarmuka yang sesuai bagi pengguna termasuk sistem kustomisasi warna dibahas pada [5]. Pengembangan sistem pengolahan suara baik untuk keperluan input maupun output dibahas pada [6].

### 1.5 Metode Penelitian

Metode penelitian yang digunakan dalam penulisan skripsi ini adalah studi literatur serta implementasi dan integrasi modul berdasarkan desain sistem dengan menggunakan kaidah rekayasa perangkat lunak. Studi literatur dilakukan untuk mendapatkan latar belakang masalah untuk merumuskan solusi dari masalah tersebut serta sebagai dasar teori dan teknis dalam implementasi modul.

Dalam perancangan sistem yang dilakukan, juga digunakan kaidah dalam rekayasa perangkat lunak, antara lain adalah siklus hidup perangkat lunak yang meliputi:

1. *User Requirement*
2. *System Design*
3. *Implementation*
4. *Testing*
5. *Maintenance*

Penulisan skripsi ini mencakup tahapan *user requirement*, *system design*, *implementation*, dan *testing*. Dalam perancangan sistem yang telah dilakukan, digunakan pula *Unified Modelling Language* (UML) sebagai bahasa standar dalam permodelan rancangan perangkat lunak.

### 1.6 Sistematika Penulisan

Sistematika dari penulisan skripsi ini adalah sebagai berikut:

Bab 1 Pendahuluan

Bab Pendahuluan ini berisi Latar Belakang, Tujuan Penulisan, Batasan Penulisan, dan Sistematika Penulisan.

## Bab 2 *Augmented Reality* dan Sistem Bantuan Pengenalan Warna Untuk Buta Warna

Bab 2 ini memberikan pengantar mengenai berbagai teori yang digunakan dalam perancangan sistem, yang terdiri dari bahasan mengenai Buta Warna Total dan Buta Warna Parsial; *Augmented Reality* dan Perkembangannya; *.NET Framework*; Bahasa Pemrograman C#, dan *EmguCV library*; dan *Windows Phone 7*.

## Bab 3 Perancangan Modul Pendeteksian Warna, Pelacakan Gerakan (*Motion Tracking*) dan Tampilan Informasi Warna

Bab 3 membahas mengenai rancang bangun salah satu bagian dari sistem bantuan penderita buta warna yang berisi Rancangan Cara Kerja Program, Diagram Alir Program dan Diagram-diagram *Unified Modelling Language* (UML).

## Bab 4 Implementasi Modul dan Integrasi Sistem Bantuan Penderita Buta Warna

Pada Bab 4 ini dibahas mengenai implementasi modul pendeteksian warna dan integrasi modul ke dalam sistem bantuan penderita buta warna. Bahasan implementasi modul mencakup implementasi modul pada *embedded system* dan pada aplikasi *mobile* dengan platform *Windows Phone 7*.

## Bab 5 Pengujian dan Analisis Modul Pendeteksian Warna

Dari hasil implementasi yang dilakukan, dilakukan pengujian dengan hasil yang dibahas pada Bab 5. Pengujian dan analisis hasil dilakukan berdasarkan hasil pengujian teknis dan pengujian oleh penguji dan dilakukan terhadap modul pada *embedded system* dan aplikasi *mobile*.

## Bab 6 Penutup

Bab 6 berisi penutup dari penulisan skripsi ini.

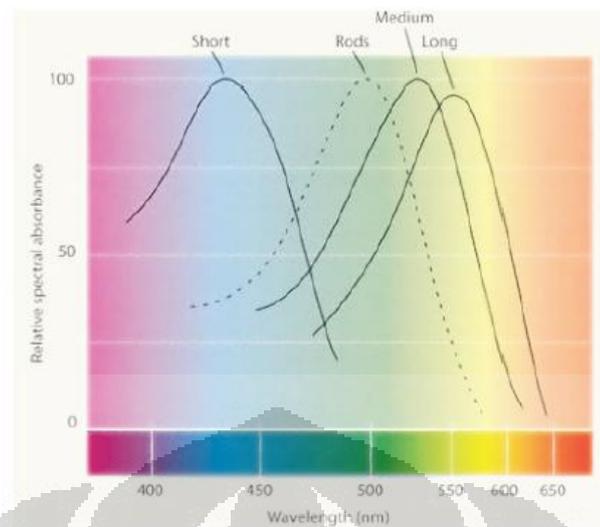
## **BAB 2**

### **AUGMENTED REALITY DAN SISTEM BANTUAN PENGENALAN WARNA UNTUK BUTA WARNA**

#### **2.1 Buta Warna Total Dan Buta Warna Parsial**

Buta warna adalah mutasi genetik yang menjadikan penglihatan subjek penderita berkurang kepekaannya terhadap panjang gelombang warna tertentu, yang terjadi secara alami dalam sebuah populasi [2]. Penyakit buta warna ini berpotensi menyebabkan kesulitan bagi penderita dalam kehidupannya. Dengan penyakit semacam ini, penderita akan mengalami kesulitan dalam mengenali warna yang banyak digunakan sebagai penanda dalam kehidupan modern, seperti warna lampu lalu lintas, atau dalam lingkup yang lebih khusus, warna zat kimia, warna penanda dalam komponen elektronik, dan lain sebagainya.

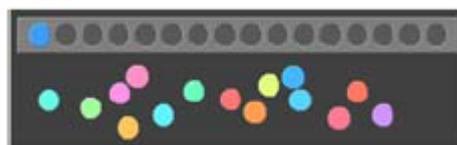
Sistem penglihatan manusia di dalam mata memiliki dua jenis reseptor, yakni sel-sel batang (*rods*) dan sel-sel kerucut (*cones*) [2]. Sel-sel batang merupakan reseptor yang peka terhadap rangsangan cahaya dengan intensitas rendah, dan tidak dapat membedakan warna. Sel-sel kerucut merupakan sel-sel yang sensitif terhadap intensitas cahaya yang terang dan memiliki kemampuan membedakan warna. Warna yang dimengerti oleh manusia itu sendiri merupakan hasil pengolahan sel-sel kerucut. Sel-sel kerucut tersebut dibagi menjadi tiga jenis yakni sel kerucut dengan panjang gelombang panjang, menengah, dan pendek. Ketiga panjang gelombang tersebut berkaitan dengan tiga warna utama, yakni merah (gelombang panjang), hijau, (gelombang menengah) dan biru (gelombang pendek). Daerah warna dan panjang gelombang yang dapat ditangkap oleh sel-sel kerucut tersebut dapat dilihat pada Gambar 2.1. Dengan kemampuan reseptor menerima tiga warna dengan panjang gelombang yang berbeda-beda tersebut, sistem penglihatan ini disebut sebagai trikromasi (*trichromatic vision*). Buta warna sendiri dapat dikategorikan menjadi dua macam, buta warna total dan buta warna parsial.



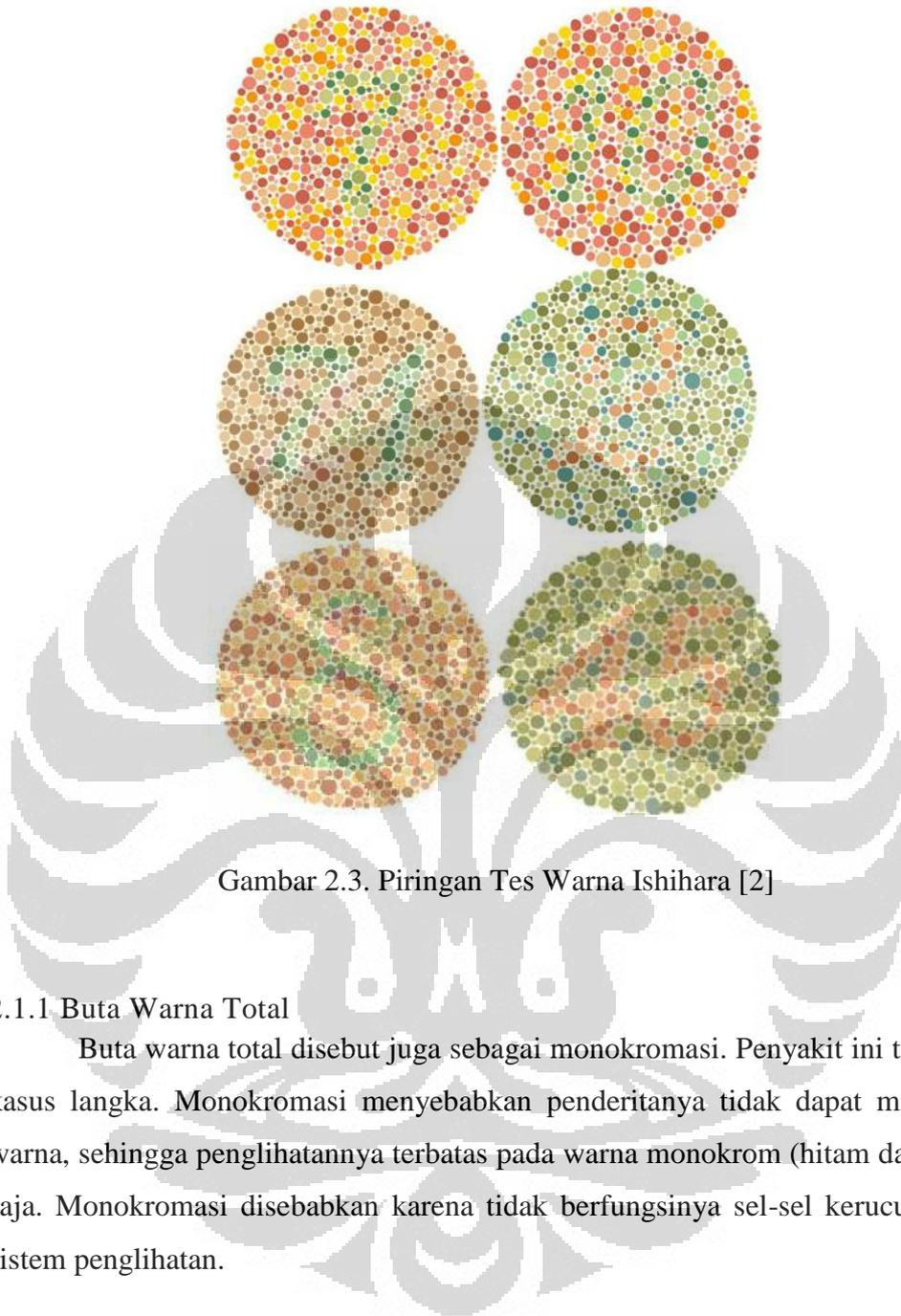
Gambar 2.1. Panjang Gelombang yang Diterima Sel-Sel Kerucut [2]

Untuk mengetahui jenis buta warna yang diderita, salah satu metode yang populer adalah menggunakan tes warna Ishihara [2]. Metode ini dinamakan sesuai penemunya, Dr. Shinobu Ishihara dari University of Tokyo. Metode ini menggunakan piringan-piringan warna yang di dalamnya terdapat lingkaran-lingkaran dengan ukuran dan warna yang tersebar secara acak. Di dalam sebaran lingkaran-lingkaran tersebut dibentuk sebuah angka atau benda tertentu yang akan terlihat oleh mata normal, tetapi tidak terlihat bagi penderita buta warna. Contoh piringan warna untuk tes Ishihara terlihat pada Gambar 2.3.

Metode lainnya adalah menggunakan *Color Difference Vectors* (CDV) [2]. Dengan metode ini dapat ditentukan jenis buta warna yang diderita, serta derajat dari buta warna yang diderita. Dengan metode ini, subjek diminta mengisi urutan warna. Warna yang berdekatan adalah warna yang paling mirip. Vektor yang digunakan dalam analisis CDV terdapat pada Gambar 2.2.



Gambar 2.2. Vektor Pada Analisis CDV [2]



Gambar 2.3. Piringan Tes Warna Ishihara [2]

#### 2.1.1 Buta Warna Total

Buta warna total disebut juga sebagai monokromasi. Penyakit ini termasuk kasus langka. Monokromasi menyebabkan penderitanya tidak dapat mengenali warna, sehingga penglihatannya terbatas pada warna monokrom (hitam dan putih) saja. Monokromasi disebabkan karena tidak berfungsinya sel-sel kerucut dalam sistem penglihatan.

#### 2.1.1 Buta Warna Parsial

Selain buta warna total, terdapat pula buta warna parsial. Salah satu bentuk dari buta warna parsial ini disebut sebagai dikromasi (*dichromachy*). Dikromasi disebabkan karena ketiadaan salah satu dari tiga sel kerucut yang seharusnya dimiliki, yang pada akhirnya menyebabkan kepekaan terhadap salah satu panjang gelombang tersebut diatas hilang secara total. Kelainan ini merupakan sebuah kasus yang langka. Bentuk lain dari buta warna parsial disebut sebagai *anomalous*

*trichromachy*, dengan ketidaksempurnaan dalam sel-sel kerucut yang ada. Kelainan ini merupakan kasus yang paling sering ditemukan. Dalam dunia medis, kelemahan mengenali warna merah disebut sebagai protanomali (*protanomaly*), kelemahan warna hijau disebut sebagai deutanomali (*deutanomaly*), sedangkan kelemahan pada sel kerucut yang peka terhadap warna biru disebut sebagai tritanomali (*tritanomaly*). Tritanomali menyebabkan penderitanya mengalami kesulitan dalam membedakan warna biru dan kuning.

Secara umum, penderita buta warna parsial akan mengalami kesulitan membedakan warna merah, kuning, atau hijau, serta membedakan warna tersebut yang memiliki intensitas rendah.

## 2.2 Teknologi Augmented Reality

*Augmented Reality* (AR) adalah sebuah teknologi baru di bidang multimedia. AR merupakan pengembangan dari teknologi pendahulunya, yaitu *Virtual Reality* (VR). Istilah VR diperkenalkan pada 1986. VR sendiri memiliki beberapa definisi, antara lain:

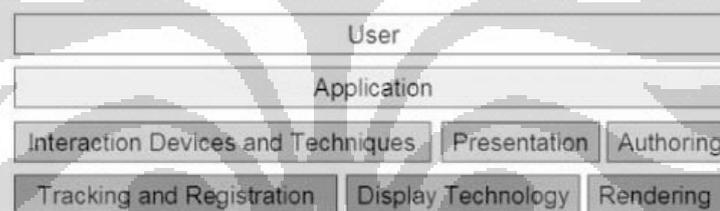
- a. VR adalah sebuah teknologi yang dapat meyakinkan penggunanya bahwa sebenarnya dia sedang berada di tempat lain dengan menggunakan data yang dihasilkan oleh computer [7].
- b. VR adalah simulasi elektronik dari sebuah lingkungan yang dapat dirasakan pengguna dengan menggunakan *head-mounted-goggle* dan pakaian khusus, sehingga pengguna merasakan keadaan yang realistis [7].

Dari pengertian-pengertian di atas, dapat ditarik sebuah kesimpulan bahwa pada intinya, VR merupakan sebuah teknologi yang memungkinkan pengguna merasakan lingkungan buatan (*virtual*) seakan seperti lingkungan yang nyata. Pada umumnya, teknologi VR menggunakan perangkat-perangkat khusus, seperti *head-mounted goggle* ataupun perangkat lainnya, untuk memungkinkan penggunanya merasakan lingkungan virtual tersebut sebagai lingkungan yang mendekati nyata.

Dengan berjalannya waktu dan berkembangnya teknologi multimedia, muncul teknologi baru sebagai penerus VR, yaitu *Augmented Reality* (AR). Dengan AR, pengguna dapat merasakan kondisi atau lingkungan nyata dengan

tambahan satu atau lebih objek virtual yang dibuat dan ditambahkan oleh computer. Objek tambahan tersebut secara umum menambahkan (*augmenting*) informasi pada lingkungan nyata yang sebenarnya. Informasi yang ditambahkan tersebut disesuaikan dengan tujuan pembuatan dari aplikasi AR itu sendiri. Titik utama perbedaan antara VR dan AR adalah, pada AR, lingkungan nyata tidak sepenuhnya digantikan oleh lingkungan buatan yang bersifat virtual seperti pada VR, akan tetapi lingkungan nyata tersebut masih memiliki peran yang signifikan [8].

Masalah paling mendasar dalam pengembangan teknologi atau aplikasi AR adalah bagaimana menggabungkan lingkungan nyata dengan objek-objek virtual yang ingin ditambahkan ke dalam lingkungan nyata tersebut. Terdapat tiga Unsur dasar dalam pengembangan sistem AR [8], seperti terlihat dalam *building block* pada Gambar 2.4.



Gambar 2.4. AR *Building Block* [8]

Unsur pertama adalah *registration* dan *tracking*. *Registration* adalah bagaimana membentuk hubungan antara lingkungan asli dengan penambahan objek maya. *Tracking* adalah bagaimana menangkap dan melacak lingkungan nyata. *Tracking* merupakan hal yang krusial dalam aplikasi AR, karena diperlukan proses *tracking* yang cepat, dapat diandalkan, untuk menghasilkan aplikasi AR yang realistis. Beberapa metode *tracking* telah dikembangkan, antara lain adalah *mechanical and electromagnetic tracking*, *optical tracking*. Salah satu bentuk dari *optical tracking* adalah dengan menggunakan sensor infra merah yang menawarkan *tracking* dengan akurasi tinggi namun dengan harga yang tinggi. Metode lain dari *optical tracking* adalah *marker-based tracking*, yakni *tracking* dengan menggunakan penanda khusus, biasanya berupa kertas yang tercetak

dengan pola tertentu. Terdapat juga metode tanpa menggunakan penanda khusus yang disebut sebagai *markerless tracking*.

Unsur penting kedua adalah teknologi *display (display technology)*. Teknologi *display* merupakan teknologi yang menjadi antarmuka langsung antara pengguna dengan sistem AR, sehingga menjadi salah satu faktor yang cukup krusial. Pengembangan teknologi *display* dibatasi oleh keterbatasan dari segi optik, teknis, serta faktor manusia. Salah satu teknologi *display* yang sering digunakan dalam pengembangan aplikasi AR sekarang ini adalah *head-mounted display*.

Unsur terakhir adalah *real-time rendering*. Proses ini terkait bagaimana menggabungkan kondisi nyata dengan objek artifisial yang dihasilkan oleh komputer menjadi satu kesatuan utuh. Untuk menghasilkan tampilan AR yang realistis, diperlukan proses *rendering* yang bersifat *real-time*, sehingga apa yang dilihat oleh pengguna mendekati kondisi nyata.

Pada perancangan dan implementasi modul pendeteksian warna yang dibahas dalam skripsi ini, prinsip *augmented reality* digunakan untuk menampilkan informasi warna dari objek yang tertangkap oleh kamera pengguna.

## **2.3 Pengolahan Citra Untuk Pengenalan Warna**

### **2.3.1 Pengolahan Citra untuk Pengenalan Warna**

Pengolahan citra dalam bentuk pendeteksian warna merupakan salah satu bagian vital dari sistem ini. Dalam sistem ini, pengenalan warna diperlukan untuk mengolah input berupa *video stream* yang didapat dari kamera, sehingga didapatkan hasil berupa informasi warna di daerah atau objek tertentu dari masukan *video stream* tersebut. Informasi warna yang ingin didapatkan tersebut disesuaikan dengan pilihan pengguna sesuai pilihan jenis buta warna yang dideritanya, yang menentukan warna apa yang perlu dimunculkan informasinya. Informasi warna ini diolah dan ditambahkan dengan masukan *video stream* secara *real-time* sesuai tujuan dari sistem ini, memberikan informasi warna kepada penderita buta warna.

Pengolahan citra dalam bentuk pendeteksian warna digunakan dalam berbagai aplikasi, antara lain untuk melakukan pendeteksian rambu-rambu lalu lintas [12], sistem pendeteksian wajah [13], dan pengenalan objek yang memiliki karakteristik atau warna tertentu [14]. Aplikasi-aplikasi tersebut menggunakan pengenalan warna sebagai komponen utamanya.

Pengolahan warna terkait dengan pengelompokan warna-warna yang dikenal sebagai *colorspace*. Untuk mendeteksi objek berdasarkan warna biasanya dilakukan transformasi dari satu *colorspace* ke *colorspace* yang lain. *Colorspace* yang paling umum ditemui adalah RGB (Red, Green, Blue). Pada [12], dalam melakukan deteksi warna, objek dengan RGB *colorspace* ditransformasikan menjadi HSV (*Hue, Saturation, Value*) *colorspace*, yang kemudian dibagi-bagi menjadi kelompok yang lebih kecil yang dinamakan *sub-space*. Komponen *saturation* dan *value* hasil transformasi tersebut digunakan untuk menentukan lokasi dari komponen *hue* yang merupakan komponen dengan informasi warna yang paling banyak.

Pengenalan objek yang memiliki karakteristik pola warna tertentu juga mungkin untuk dilakukan. Salah satu metode yang digunakan untuk melakukan hal tersebut dinamakan *Color Co-occurrence Histogram* (CCH) [14]. Metode ini merepresentasikan objek dengan karakteristik warna yang dimilikinya. Dengan cara ini, walaupun objek mengalami perubahan bentuk atau arah, objek tersebut tetap akan dapat dikenali. Pengembangan lebih lanjut dari metode ini disebut sebagai *Color Edge Co-occurrence Histogram* (CECH) [14]. Hasil pengenalan warna dengan metode ini dapat dilihat pada Gambar 2.5.



Gambar 1.5. Pengenalan Warna dengan CECH [14]

Salah satu tantangan proses pengenalan warna adalah informasi warna itu sendiri. Informasi warna pada dasarnya sensitif variasi kondisi pencahayaan, yang dapat diakibatkan karena adanya bayangan, efek awan, dan pencahayaan matahari. Tantangan lainnya adalah banyaknya variasi perbedaan persepsi warna yang dikenali oleh sistem penglihatan manusia. Untuk mengurangi variasi persepsi ini, dapat dilakukan kuantisasi warna menjadi kelompok warna yang lebih kecil lingkupnya. Kuantisasi warna dapat dilakukan pada RGB *colorspace*, namun lebih baik dilakukan pada *colorspace* yang lain seperti LUV, atau HSV. Untuk memetakan warna dari banyaknya variasi persepsi warna digunakan kamus warna seperti ISCC-NBS, yang dapat menghasilkan pengelompokan warna dengan lingkup yang lebih kecil. Langkah ini digunakan pada [14].

### 2.3.2 Model Warna RGB, HSV, dan HLS

#### 2.3.2.1 Model Warna Red, Green, Blue (RGB)

RGB merupakan model warna yang paling umum dikenal. RGB menggunakan tiga warna dasar sebagai komponen utama untuk menyusun warna-warna lainnya. Tiga warna dasar tersebut adalah merah, hijau, dan biru (red, green, blue). Warna dasar tersebut kemudian dikombinasikan untuk menghasilkan warna-warna turunan yang lain. Rentang nilai masing-masing komponen berada pada rentang 0 sampai dengan 255.

#### 2.3.2.2 Model Warna Hue, Saturation, Value (HSV)

HSV adalah model warna yang didapat dari transformasi linear dari model warna RGB. Model warna ini diperkenalkan pertama kali pada tahun 1978 oleh Alvy Ray Smith. Dengan menggunakan HSV, perhitungan dan pengolahan warna menjadi lebih sederhana dibandingkan menggunakan model warna RGB, sehingga sesuai dalam berbagai aplikasi pengolahan citra.

Dalam HSV, warna direpresentasikan ke dalam tiga komponen yaitu hue, saturation, dan value. Hue menunjukkan komponen utama yang mendefinisikan warna. Rentang hue standar adalah  $0^\circ$  sampai dengan  $360^\circ$ . Komponen saturation menunjukkan kepekatan sebuah warna, dengan rentang nilai standar antara 0-

100%. Komponen value menunjukkan tingkat kecerahan warna, dengan rentang nilai 0-100%.

Pada OpenCV, terdapat perbedaan rentang nilai hue, saturation, dan value yang digunakan. Nilai ketiga komponen tersebut berada dalam rentang antara 0 sampai dengan 255.

### 2.3.2.3 Model Warna Hue, Lightness, Saturation (HLS)

HLS atau yang disebut juga dengan HSL merupakan model warna dengan konsep yang mirip dengan HSV. Walaupun demikian, terdapat perbedaan dalam definisi komponen yang ada. Hue pada HLS menunjukkan komponen utama warna, sama dengan HSV, pada umumnya digunakan rentang nilai 0° sampai dengan 360° untuk komponen ini. Komponen lightness menunjukkan terang gelap dari warna yang ada. Nilai komponen lightness berkisar antara 0-100%. Nilai lightness 0 menunjukkan warna hitam, sedangkan nilai lightness 100% atau 1 menunjukkan warna putih. Komponen saturation pada HLS menunjukkan tingkat keabuan dari warna. Nilai 0 pada komponen saturation menunjukkan warna abu-abu, sedangkan nilai 100% atau 1 berarti komponen saturation tidak memberikan pengaruh pada warna yang ditunjukkan oleh komponen hue. Sama halnya dengan model warna HSV, pada OpenCV, nilai ketiga komponen dari HLS juga berkisar pada rentang 0 sampai dengan 255.

## 2.4 .Net Framework, Bahasa Pemrograman C#, dan Emgucv Library Serta Windows Phone 7 Sebagai Salah Satu Platform Dalam Implementasi Sistem

### 2.4.1 NET Framework dan Perkembangannya

.NET *Framework* merupakan sebuah *framework* yang digunakan dalam pengembangan aplikasi, baik aplikasi *desktop* maupun aplikasi berbasis web. Pengembangan .NET *Framework* diawali oleh Microsoft yang membentuk .NET Initiative pada tahun 2000. Salah satu aspek penting dari .NET *Framework* adalah independensi terhadap bahasa pemrograman tertentu [5]. Hal ini dimungkinkan dengan proses kompilasi program yang berbeda dengan proses kompilasi program pada umumnya.

Proses kompilasi program pada umumnya dilakukan oleh *compiler* yang mengkompilasi kode program menjadi berkas program yang dapat dijalankan

(*executable*). Dalam *.NET Framework*, metode yang digunakan untuk mengubah kode program menjadi program yang dapat dijalankan berbeda, yang disebut sebagai *managed compilation* [15]. Dengan metode ini, kode program tidak secara langsung dikompilasi menjadi berkas program yang dapat dijalankan, tetapi dikompilasi menjadi *assembly* dengan komponen di dalamnya berupa metadata dan *Intermediate Language (IL)*. IL tersebut kemudian diterjemahkan oleh *Common Language Runtime (CLR)* menjadi kode biner yang dimengerti oleh sistem operasi. Proses penerjemahan ini dilakukan oleh *Just-In-Time (JIT) compiler*. *JIT compiler* melakukan kompilasi kode hanya pada saat pertama kali kode tersebut dijalankan saja, kode hasil kompilasi tersebut kemudian disimpan oleh CLR. Proses *managed compilation* tersebut dapat diilustrasikan pada Gambar 2.6.



Gambar 2.6. Proses *managed compilation* oleh *.NET Framework*

Bila dicermati lebih dalam, proses *managed compilation* tersebut tentunya lebih kompleks dibandingkan dengan proses kompilasi tradisional. Di sisi lain, *managed compilation* menawarkan kelebihan seperti portabilitas antar sistem operasi, interoperabilitas antar bahasa pemrograman, dan fitur-fitur untuk melakukan manajemen eksekusi program seperti manajemen memori dan keamanan [16].

Selain kelebihan tersebut, *.NET Framework* memungkinkan program yang dibuat di atas *framework* tersebut independen terhadap *platform* tertentu (*platform independent*). Artinya, program yang dibuat tidak tergantung pada *platform* tertentu untuk berjalan dengan semestinya. Selama *.NET Framework* terpasang pada *platform* tersebut, maka program yang berbasis *.NET Framework* dapat dijalankan.

.NET *Framework* juga memungkinkan interoperabilitas antar bahasa pemrograman. Dengan adanya *Intermediate Language* (IL) dalam proses kompilasi yang dilakukan .NET *Framework*, program berbasis .NET menjadi program yang bersifat independen terhadap bahasa pemrograman tertentu. Bahasa-bahasa pemrograman yang dapat dikompilasi menjadi IL oleh .NET *Framework* disebut sebagai *.NET-compliant* language [16]. Independensi terhadap bahasa pemrograman tersebut memungkinkan pengembang mengembangkan program dengan bahasa pemrograman yang berbeda-beda yang kemudian dapat dikompilasi menjadi IL dan digabungkan menjadi satu program.

Standardisasi .NET dilakukan oleh European Computer Manufacturers Association (ECMA) dan Open Standards Institute (OSI). Standar yang dikeluarkan disebut sebagai *Common Language Infrastructure* (CLI). Standar ini salah satunya terkait dengan IL. IL yang digunakan untuk produk .NET buatan Microsoft dikenal sebagai MSIL. Implementasi lain dari IL antara lain adalah Mono, yang merupakan produk dari Novell.

Bahasa-bahasa pemrograman *.NET-compliant* diantaranya adalah sebagai berikut:

- a. C#
- b. Visual C++ .NET
- c. Visual Basic .NET
- d. Java
- e. Pascal
- f. Perl
- g. Python

Untuk pengembangan program berbasis .NET, Microsoft membuat aplikasi khusus sebagai lingkungan pengembangan program terintegrasi (*Integrated Development Environment*; IDE) yang dikenal dengan Microsoft Visual Studio. Visual Studio telah mengalami berbagai pengembangan dimulai dari versi pertamanya, yaitu Visual Studio 97, hingga versi terbaru yang diluncurkan pada tahun 2010, yaitu Visual Studio 2010. .NET *Framework* untuk

sistem operasi Windows juga telah berkembang mulai dari *.NET Framework 1.0* hingga versi terbarunya *.NET Framework 4.0*.

#### 2.4.2 Bahasa Pemrograman C#

C# adalah salah satu bahasa pemrograman tingkat tinggi yang digunakan untuk membangun aplikasi di atas *.NET Framework*. C# merupakan bahasa pemrograman berorientasi objek yang memungkinkan pengembang mengembangkan aplikasi *desktop*, maupun aplikasi untuk *platform Windows Phone 7* dengan menggunakan *development kit* tambahan.

C# tergolong bahasa pemrograman baru yang menawarkan kesederhanaan, dan performa [17]. C# didesain untuk mengatasi kelemahan pada bahasa pemrograman lain. Fitur-fitur dari bahasa pemrograman ini antara lain adalah sebagai berikut [17]:

##### 1. Orientasi komponen (*Component Orientation*)

C# memungkinkan pengembangan aplikasi untuk membangun berbagai komponen seperti *properties*, *events*, dan *attributes*. Komponen-komponen tersebut dibuat dengan tujuan untuk melakukan manajemen kompleksitas sebuah program.

##### 2. *One-stop coding*

Dengan C#, bila pengembang melakukan deklarasi, baik variabel, maupun yang lainnya, maka deklarasi tersebut akan dilokalisasi pada deklarasi itu sendiri tidak diletakkan pada beberapa berkas kode program yang berbeda atau diletakkan pada berbagai tempat dalam sebuah berkas kode program.

##### 3. *Versioning*

C# memungkinkan komponen dengan versi yang lebih baru bekerja bersama dengan komponen lama yang dibutuhkannya.

##### 4. *Type safety*

*Type-safe* memiliki makna bahwa variabel yang ada dapat diakses hanya dengan mengetikkan huruf yang bersesuaian dengan variabel tersebut.

## 5. Manajemen Memori

C# memiliki *runtime* khusus yang dapat melakukan manajemen memori secara otomatis yang meniadakan keharusan bagi pemrogram untuk melakukan proses-proses untuk menghapus objek yang telah dijalankan.

Dalam mengembangkan program dengan bahasa pemrograman C#, pengembang dapat memanfaatkan *library* yang berisi berbagai fungsi dasar untuk dipanggil dan digunakan dalam programnya. C# tidak memiliki *library* yang khusus dibuat untuk bahasa pemrograman ini, C# menggunakan *library* dari .NET *Framework* yang dinamakan sebagai .NET *Framework Class Library* (FCL) yang berisi berbagai fungsi yang dapat digunakan oleh berbagai bahasa program yang dijalankan di atas .NET *Framework*.

### 2.4.3 OpenCV dan EmguCV *Library*

OpenCV adalah sebuah *library* yang menyediakan fungsi untuk membangun aplikasi yang berkaitan dengan *computer vision* [18]. OpenCV dikembangkan dengan tujuan untuk menyediakan infrastruktur *computer vision* yang mudah digunakan, yang memungkinkan pengembangan aplikasi *computer vision* dengan cepat. OpenCV sendiri didesain untuk mengefektifkan proses komputasi serta mendukung aplikasi yang bersifat *real-time*. *Library* ini memiliki sekitar 500 fungsi terkait *computer vision* yang terbagi ke dalam berbagai area.

*Computer vision* sendiri didefinisikan sebagai transformasi data dari kamera atau kamera video menjadi bentuk yang lain atau untuk melakukan suatu fungsi tertentu [18]. *Computer vision* merupakan bidang yang cukup rumit dikaitkan dengan keterbatasan komputer dalam mengolah gambar atau video dibandingkan dengan kemampuan manusia. Sebagai contoh adalah bagaimana merekonstruksi bentuk tiga dimensi (3D) dari suatu benda dari gambar yang bersifat dua dimensi (2D). Masalah lainnya adalah adanya derau dalam gambar atau video, sementara pengolahan dari gambar atau video tersebut membutuhkan akurasi data yang tinggi.

OpenCV ditujukan untuk menyediakan perangkat yang dapat menjadi dasar mengatasi masalah dalam *computer vision* tanpa adanya keharusan bagi pengembang aplikasi untuk mengembangkan sesuatu yang telah ada.

OpenCV dikembangkan pertama kali oleh Intel bekerja sama dengan Software Performance Libraries dari Rusia. Dengan tujuan awal pengembangannya adalah sebagai berikut:

1. Menyediakan infrastruktur dasar untuk *computer vision*.
2. Menyediakan infrastruktur umum yang memungkinkan kode program dapat dengan mudah dibaca dan ditransfer.
3. Menyediakan *library* bersifat gratis dan terbuka yang tidak membutuhkan lisensi komersial.

Versi pertama, *alpha*, dari OpenCV diluncurkan pada 1999. OpenCV telah dikembangkan dari tahun ke tahun, dengan versi terakhir adalah versi 2.2 yang dirilis pada Desember 2010.

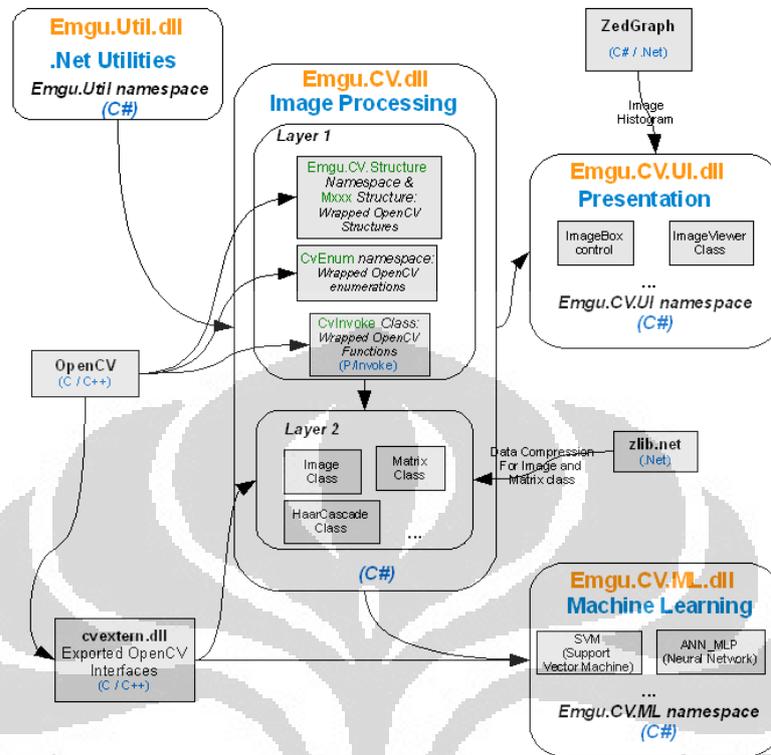
OpenCV dikembangkan untuk digunakan dengan bahasa pemrograman C, C++, Python, Ruby, dan Matlab. OpenCV dapat dijalankan pada sistem operasi Windows, Linux, dan Mac OS. Untuk menggunakan fungsi-fungsi yang terdapat pada OpenCV pada *platform* .NET, diperlukan semacam adaptor yang disebut sebagai *wrapper*. *Wrapper* yang dapat digunakan untuk menjalankan OpenCV di atas platform .NET ini antara lain adalah EmguCV, SharperCV, OpenCVDotNet, dan Code Project. Pada tulisan ini, yang akan dibahas adalah EmguCV.

EmguCV adalah .NET *wrapper* untuk OpenCV yang bersifat *cross-platform*, sehingga dapat dijalankan pada berbagai sistem operasi. Dengan EmguCV, fungsi-fungsi yang terdapat pada OpenCV dapat dipanggil dari bahasa pemrograman yang kompatibel dengan .NET (*.NET-compliant*), seperti C#, Visual Basic, Visual C++, dan lain sebagainya.

Arsitektur dari Emgu CV dapat diilustrasikan pada Gambar 2.7 [19]. Dari Gambar 2.4, terlihat bahwa *wrapper* EmguCV memiliki dua lapisan (*layer*) sebagai berikut:

- a. Lapisan dasar (*layer 1*)  
Merupakan lapisan yang terdiri atas fungsi, struktur, dan enumerasi yang mengacu secara langsung pada yang terdapat dalam OpenCV.
- b. Lapisan kedua (*layer 2*)

Lapisan kedua ini berisi kelas-kelas yang berkaitan dengan .NET.



Gambar 2.7. Arsitektur EmguCV [19]

#### 2.4.4 Windows Phone 7

Windows Phone 7 (WP 7), generasi baru penerus Windows Mobile yang diluncurkan pada Oktober 2010. Sebagai sistem operasi *mobile* baru, WP 7 membawa perubahan yang cukup radikal dibandingkan dengan pendahulunya, yaitu generasi Windows Mobile. Dari segi perangkat keras (*handset*), Microsoft tidak melakukan produksi sendiri, melainkan tetap diproduksi oleh vendor perangkat genggam. Akan tetapi, Microsoft menetapkan spesifikasi perangkat keras yang harus dibenamkan dalam WP 7, yang disebut sebagai *chassis* [20].

Perangkat dan fitur yang dimiliki oleh Windows Phone 7 antara lain adalah sebagai berikut:

- a. Wi-Fi : menyediakan koneksi Wi-Fi untuk akses internet.
- b. Kamera: ditanamkan kamera dengan resolusi minimal 5 megapixel dengan flash.

- c. Akselerometer: perangkat yang dapat mendeteksi akselerasi dan dapat digunakan untuk mendeteksi pergerakan perangkat genggam.
- d. Lokasi: terdapat perangkat GPS, dan kompas yang terhubung dengan web untuk menyediakan informasi lokasi bagi pengguna.
- e. *Multitouch user-input*: sebagai antarmuka dengan pengguna, WP 7 menggunakan perangkat masukan yang bersifat *multitouch*.

Pengembang dapat membuat aplikasi untuk WP 7 dengan bahasa pemrograman berbasis .NET *Framework*, yaitu C#. WP 7 mendukung pengembangan aplikasi dengan *platform* pengembangan yang dibuat Microsoft, yaitu Silverlight dan XNA. Silverlight dikenal sebagai *platform* untuk pengembangan aplikasi berbasis web yang menawarkan kemudahan dalam membuat antarmuka yang indah, dengan menggabungkan grafik vektor, teks, media, animasi, dan kontrol. Di sisi lain, XNA merupakan *platform* pengembangan permainan yang mendukung pembuatan permainan baik 2D atau 3D. Program yang dibuat pengembang selanjutnya dapat disebarluaskan atau dipasarkan melalui Windows Phone Marketplace [20].

Untuk melakukan pengembangan aplikasi pada platform Windows Phone 7, diperlukan perangkat Windows Phone Development Tools. Windows Phone Tools berisi perangkat dasar pengembangan aplikasi untuk Windows Phone mencakup emulator, Visual Studio 2010 Express, dan Expression Blend 4 for Windows Phone. *Emulator* digunakan untuk mensimulasikan jalannya aplikasi pada *device* sebenarnya. Visual Studio 2010 Express digunakan sebagai lingkungan pengembangan terintegrasi, sedangkan Expression Blend 4 digunakan sebagai sarana perancangan antarmuka aplikasi.

## BAB 3

### PERANCANGAN MODUL PENDETEKSIAN WARNADAN TAMPILAN INFORMASI WARNA

#### 3.1 Rancangan Cara Kerja Program

##### 3.1.1 Deskripsi

Seperti telah dituliskan pada bagian awal tulisan ini, sistem bantuan untuk penderita buta warna ini dirancang memiliki perangkat keras dan perangkat lunak yang bekerja membentuk satu kesatuan yang utuh. Dengan sistem ini, pengguna mendapatkan bantuan dalam mengenali objek dengan warna yang menjadi masukan ke dalam sistem.

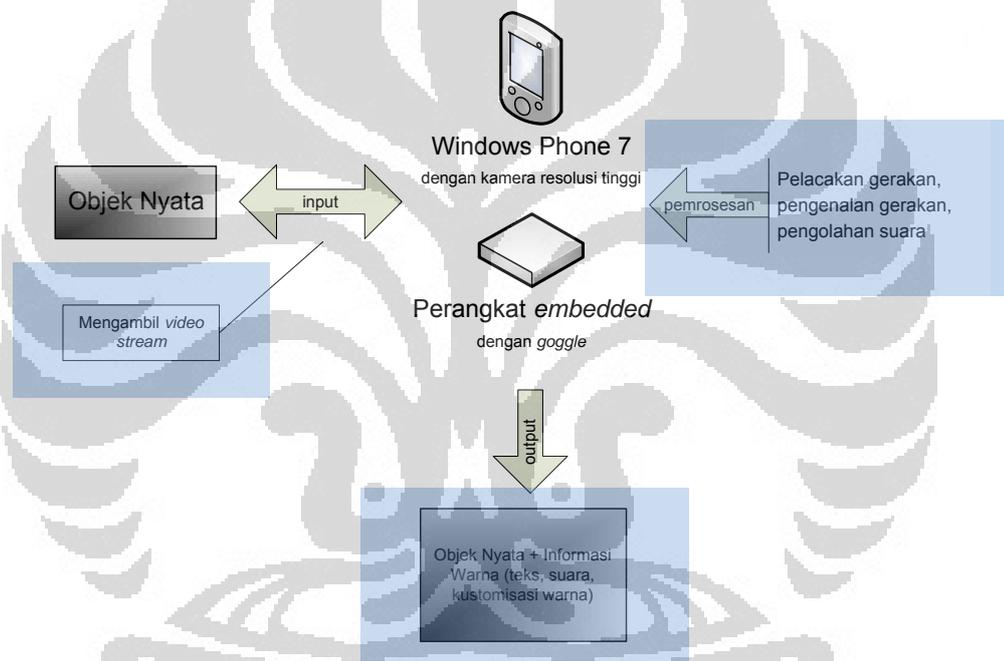
Sistem akan melakukan pendeteksian warna, terhadap objek yang ditunjuk oleh pengguna menggunakan perangkat keras yang dirancang dalam sistem ini. Perangkat keras yang dirancang adalah sebuah perangkat *embedded system* yang dilengkapi perangkat masukan dan keluaran. Perangkat keras dalam sistem ini juga akan memanfaatkan EBOX dari Microsoft sebagai *single-purpose computer*. EBOX yang akan digunakan dijalankan dengan menggunakan sistem operasi Windows Embedded Standard 2009. Windows Embedded Standard sendiri adalah sistem operasi yang khusus dikembangkan untuk perangkat *embedded system*. Sistem operasi ini dapat dikustomisasi sedemikian hingga pengembang dapat menentukan sendiri modul dan fitur yang ingin dimiliki sistem operasi yang dirancang, kemudian mengkompilasinya menjadi sistem operasi yang siap dijalankan pada perangkat *embedded*.

*Embedded system* dirancang sedemikian rupa untuk mensimulasikan *requirement* awal, yang menggunakan *head-mounted display* dengan kamera sebagai perangkat keluaran sekaligus masukan yang digunakan pengguna. Untuk mensimulasikan kondisi ini, digunakan EBOX dilengkapi *webcam*, dengan keluaran berupa tampilan pada monitor yang mendukung masukan VGA.

Aplikasi yang dibuat akan melakukan operasi terhadap *video stream* yang diambil terhadap objek yang dipilih oleh pengguna dengan operasi yang dipilih dalam perangkat lunak. Direncanakan, operasi yang dapat dilakukan dengan perangkat lunak ini antara lain adalah sebagai berikut:

- a. Meningkatkan intensitas warna tertentu yang dipilih oleh pengguna untuk mempermudah pengguna mengenali warna.
- b. Memberikan informasi warna dalam bentuk tulisan yang muncul dalam video stream yang menerapkan prinsip *augmented reality*.
- c. Informasi warna diberikan melalui media suara dengan memanfaatkan teknologi *text-to-speech*.

Sistem ini secara umum dapat digambarkan pada Gambar 1.1. Pengerjaan sistem ini dikerjakan oleh sebuah tim yang terdiri atas Alfa Sheffildi M. [4], Bayu Sri A. [5], Burhan Adi W., dan Ruki Harwahyu [6]. Modul atau bagian dari sistem yang dibahas dalam tulisan ini ditandai dengan warna pada Gambar 3.1

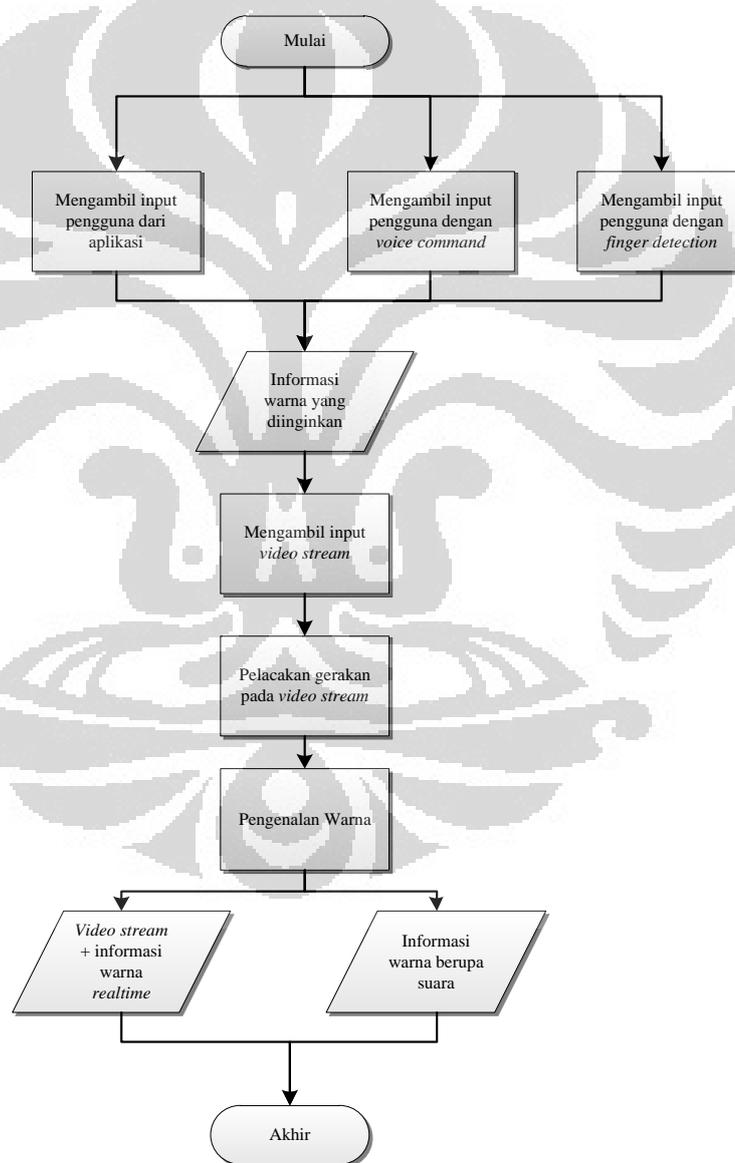


Gambar 3.1. Arsitektur Umum Sistem Bantuan Penderita Buta Warna

Modul ini merupakan bagian dari keseluruhan sistem secara umum yang dirancang untuk melakukan tugas pendeteksian warna dan pelacakan gerakan (*motion tracking*). Bahasa pemrograman yang akan digunakan adalah C# yang digunakan bersama dengan EmguCV sebagai *wrapper* library OpenCV. Secara singkat, cara kerja modul ini adalah sebagai berikut. Dengan menggunakan library DirectShow.NET, video diambil dari perangkat input, dapat berupa webcam ataupun *customized input device* yang akan dibuat. Selanjutnya, dengan library

EmguCV, *video stream* yang masuk akan diproses dan dilakukan pendeteksian warna yang sesuai dengan warna tertentu yang ditentukan sebelumnya. Kemudian dilakukan pula pemberian label berupa teks untuk warna yang telah dideteksi tersebut. Langkah-langkah yang dilakukan dalam modul ini dapat diwakili dengan Gambar 3.2.

Untuk pengembangan selanjutnya, teks tersebut juga akan diolah dengan *library text-to-speech* untuk menghasilkan keluaran berupa suara yang menginformasikan warna tertentu pada *video stream* tersebut.



Gambar 3.2. Diagram Alir Cara Kerja Modul

### 3.1.2 Diagram Alir Cara Kerja Program

Program yang dibahas disini merupakan modul bagian dari sistem besar yang digambarkan di atas untuk bantuan penderita buta warna. Modul program ini digunakan untuk mengambil input dari kamera yang merupakan kondisi nyata yang akan diolah untuk ditambahkan dengan informasi warna yang ditujukan untuk membantu penderita buta warna. Dalam rancangan program yang dibuat, informasi warna tersebut akan ditampilkan melalui media teks serta suara yang terintegrasi bersama masukan video dari kamera berisi kondisi nyata yang merepresentasikan apa yang dilihat oleh pengguna. Diagram alir cara kerja program diberikan pada Gambar 3.3.

Dari diagram alir pada Gambar 3.3, pada saat program pertama kali dijalankan, program akan menampilkan sebuah menu. Menu pilihan ini memungkinkan pengguna memilih jenis buta warna yang dideritanya, hal ini dikaitkan dengan informasi warna yang akan diberikan nantinya. Warna-warna ini berkaitan dengan buta warna parsial, yakni merah, hijau, dan biru. Bila pengguna belum mengetahui jenis buta warna yang dideritanya, di menu ini juga akan disediakan pilihan lain untuk menuju bagian program yang lain yang dapat digunakan untuk memeriksa jenis buta warna yang diderita.

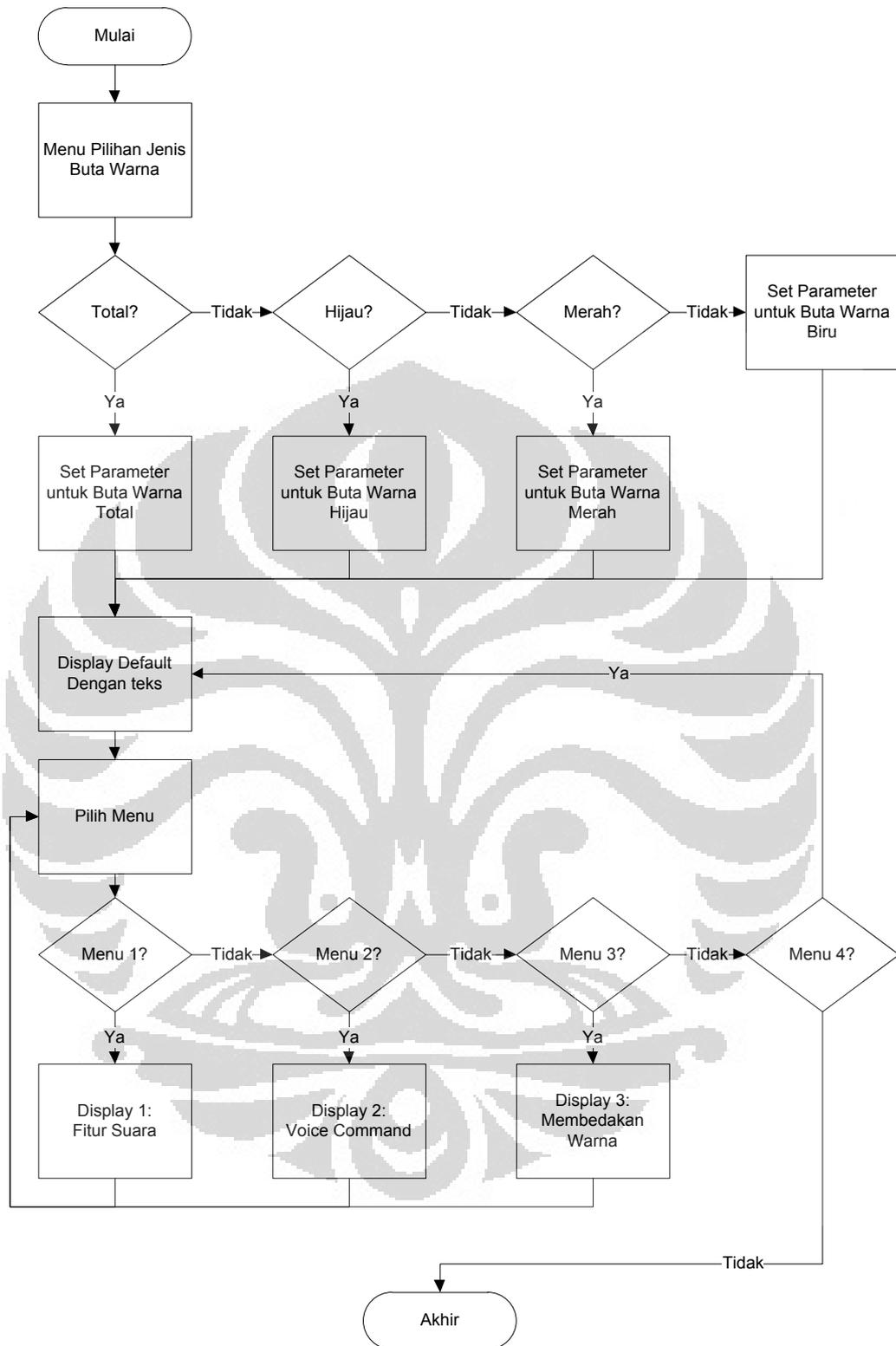
Selanjutnya, program akan mulai menangkap *video stream* dari kamera. Disinilah modul yang dirancang dalam tulisan ini bekerja, input berupa *video stream* tersebut diolah untuk dideteksi daerah dengan warna yang sesuai dengan warna yang dipilih oleh pengguna. Dilakukan juga pelacakan gerakan untuk melacak dan menandai pergerakan dari *video stream* yang mungkin terjadi, sehingga daerah yang dimaksudkan untuk diketahui warnanya juga dapat terlacak dengan akurat secara *real-time*.

Hasil pengolahan warna tersebut kemudian menghasilkan output berupa daerah dengan warna yang ditentukan tersebut. Daerah tersebut kemudian akan ditandai dengan tulisan, untuk menunjukkan informasi warna bagi penderita buta warna yang tidak dapat mengenali warna tersebut. Tampilan *video stream* yang telah ditambah dengan informasi warna ini menjadi tampilan akhir hasil olahan program. Pengguna masih dapat memanfaatkan fitur lain dari sistem melalui menu yang tersedia pada program.

Menu pertama memungkinkan pengguna mengaktifkan fitur suara dari program. Fitur ini didapat dengan mengolah hasil dari pengolahan dan pendeteksian warna yang dilakukan, seperti dijelaskan pada paragraf di atas, menjadi suara. Hal ini dilakukan dengan mengubah keluaran berupa teks yang didapat menjadi suara dengan menggunakan teknologi *text-to-speech*. Proses ini menjadi modul lain dari program dan tidak dibahas pada tulisan ini.

Menu kedua memberikan fitur *voice-command*. Dengan fitur ini, pengguna dapat memilih warna yang akan dideteksi dengan menggunakan perintah berupa suara. Perintah suara yang diberikan pengguna tersebut diolah oleh program dan digunakan untuk menentukan warna yang akan dideteksi.

Sistem ini juga dirancang memiliki perangkat input yang dapat mendeteksi jari pengguna yang menunjuk ke suatu daerah dengan warna tertentu. Warna tertentu disini yang dimaksudkan adalah warna yang tidak dapat dimengerti pengguna. Daerah dengan warna tertentu yang ditunjuk ini kemudian akan diproses untuk mendapatkan informasi warnanya. Bila informasi warna yang dimaksud telah didapat, maka informasi warna tersebut akan ditampilkan bersama dengan masukan berupa *video stream* berisi kondisi nyata. Hal ini merupakan prinsip dari *augmented reality*.



Gambar 3.3. Diagram Alir Program dalam Sistem Bantuan Penderita Buta Warna

### 3.2 Diagram-Diagram Unified Modelling Language (UML)

*Unified Modelling Language* (UML) merupakan sebuah metode untuk merepresentasikan atau mendeskripsikan desain perangkat lunak ke dalam notasi-notasi grafis yang terstandarisasi. UML dengan notasinya yang universal memudahkan pengembang melakukan kolaborasi maupun mendokumentasikan rancangan perangkat lunaknya. Dengan UML, rancangan perangkat lunak dapat direpresentasikan ke dalam diagram-diagram yang memiliki fungsi masing-masing. Berikut adalah diagram-diagram yang merepresentasikan rancangan dari modul yang akan dibuat, meliputi *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *deployment diagram*.

#### 3.2.1 Use Case Diagram

*Use Case Diagram* digunakan untuk mewakili fungsionalitas dari sistem [22]. Diagram ini menggambarkan interaksi antara pengguna, yang diwakili dengan notasi *actor*, dengan sistem untuk mencapai tujuannya. *Use Case Diagram* untuk sistem yang akan dibuat terdapat pada Gambar 3.4.

Dari diagram tersebut, dapat dilihat bahwa pada dasarnya pengguna dapat melakukan beberapa fungsi dengan sistem bantuan penderita buta warna ini. Pertama, pengguna dapat memilih jenis buta warna yang dideritanya, diberikan pilihan buta warna total dan buta warna parsial. Pada pilihan buta warna parsial terdapat sub-pilihan buta warna parsial, yaitu merah, hijau, dan biru. Pilihan tersebut disesuaikan dengan jenis buta warna yang diderita pengguna. Fungsi kedua, pengguna dapat menggunakan perangkat keras yang didesain sedemikian rupa sehingga pengguna dapat menunjuk daerah atau objek yang ingin diketahui warnanya.



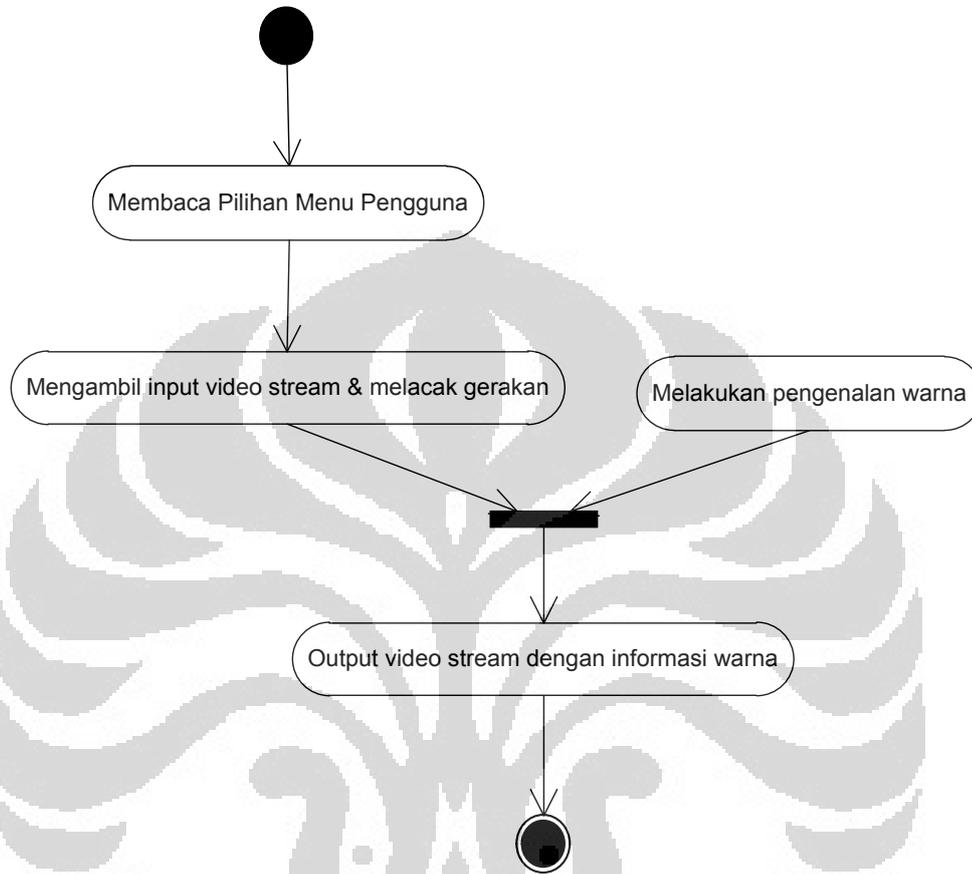
Gambar 3.4. Use Case Diagram

Fungsi ketiga, pengguna dapat mengaktifkan fitur seperti perintah suara, informasi warna dengan suara, dan fitur manipulasi warna yang dirancang untuk menaikkan intensitas warna tertentu sehingga dapat memudahkan penderita mengenali warna tersebut. Fungsi-fungsi yang ada dalam diagram tersebut pada dasarnya melibatkan perangkat lunak dan perangkat keras yang didesain dalam sistem.

### 3.2.2 Activity Diagram

*Activity diagram* merupakan diagram UML yang digunakan untuk merepresentasikan logika, *business process*, atau alur kerja program [22]. Pada

Gambar 3.5 diberikan *activity diagram* dari modul pelacakan gerakan dan pengenalan warna yang dirancang.



Gambar 3.5. *Activity Diagram*

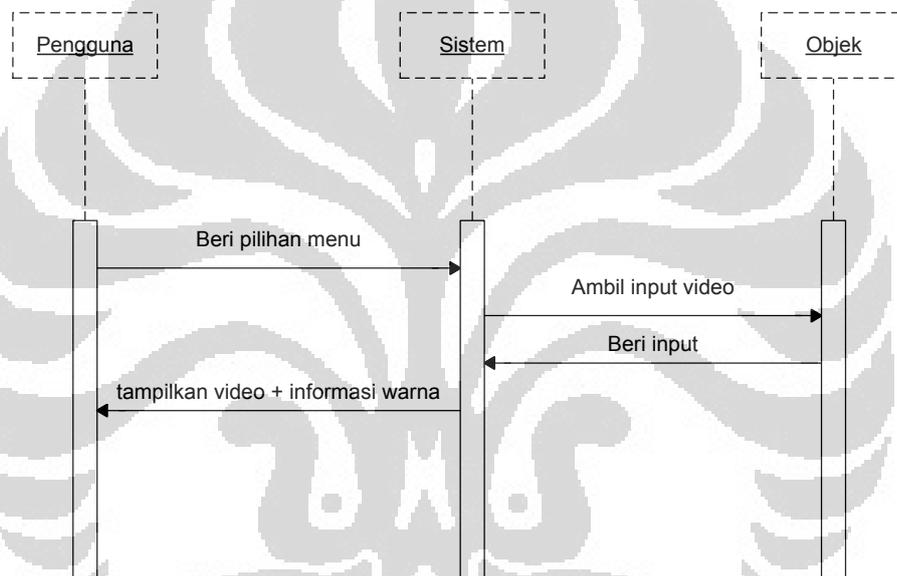
*Activity diagram* pada gambar tersebut menjelaskan alur kerja modul ini, yaitu sebagai berikut:

1. Modul akan membaca pilihan pengguna dari menu yang ditampilkan mengenai jenis buta warna yang diderita, pilihan ini menentukan warna yang harus dideteksi oleh modul.
2. Modul mengambil video dari kamera yang mewakili kondisi asli dan melakukan pengenalan warna sesuai yang ditentukan dari poin 1 di atas.
3. Modul mengolah dan menampilkan kembali *video stream* yang telah ditambahkan informasi warna berupa teks.

Informasi warna yang ditampilkan diatas, diolah pada modul lainnya, yaitu modul *text-to-speech* yang memungkinkan pengguna menggunakan fitur *voice information*.

### 3.2.3 Sequence Diagram

*Sequence Diagram* termasuk salah satu *interaction diagram* yang digunakan untuk menggambarkan perilaku dari objek-objek yang terlibat dalam sistem dan pesan-pesan antar objek yang berada dalam *use case* [22].

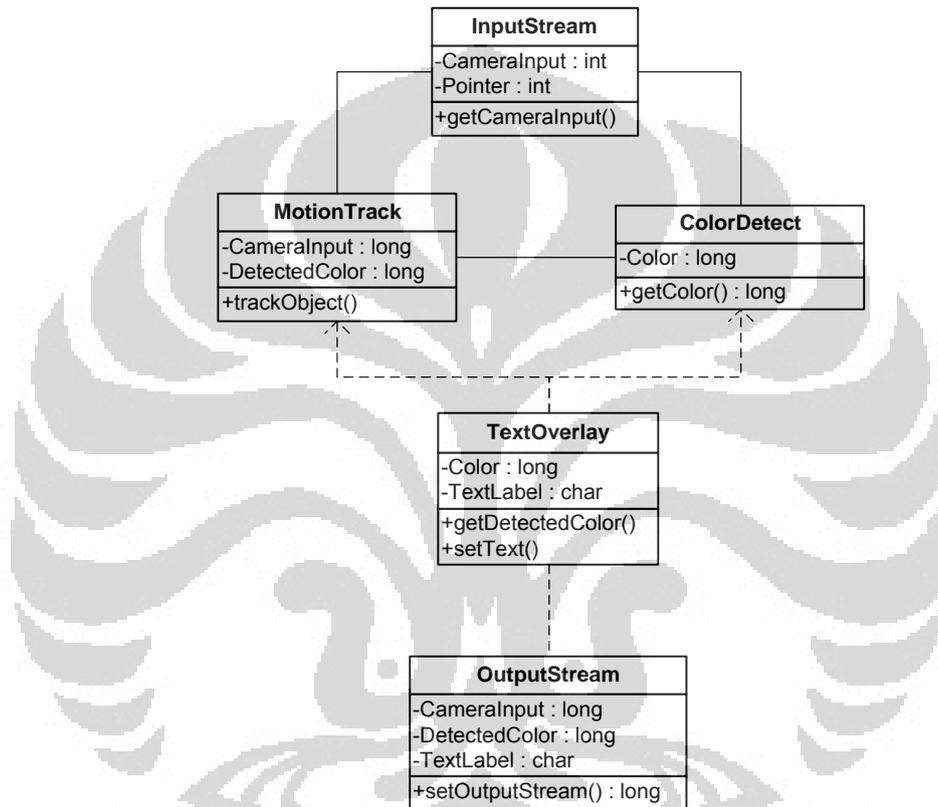


Gambar 3.6. *Sequence Diagram*

Gambar 3.6 mewakili *sequence diagram* rancangan modul. Terlihat pada diagram tersebut bahwa terdapat tiga entitas yang saling berinteraksi, yaitu pengguna, sistem, dan objek. Interaksi antara pengguna dan sistem meliputi pemberian masukan kepada sistem dan pengembalian hasil pengolahan dari masukan kembali kepada pengguna. Sistem juga berinteraksi dengan objek yang dilihat oleh pengguna. Sistem mengambil *video stream* dari objek untuk diolah dan diberikan hasilnya kepada pengguna.

### 3.2.4 Class Diagram

*Class diagram* digunakan untuk menggambarkan objek-objek dalam sistem dan hubungan-hubungan dari objek-objek tersebut [22]. Pada modul ini, setidaknya terdapat empat kelas, yang memiliki fungsi masing-masing. Kelas-kelas tersebut, seperti terlihat pada Gambar 3.7, adalah `InputStream`, `MotionTrack`, `ColorDetect`, `TextOverlay`, dan `OutputStream`.



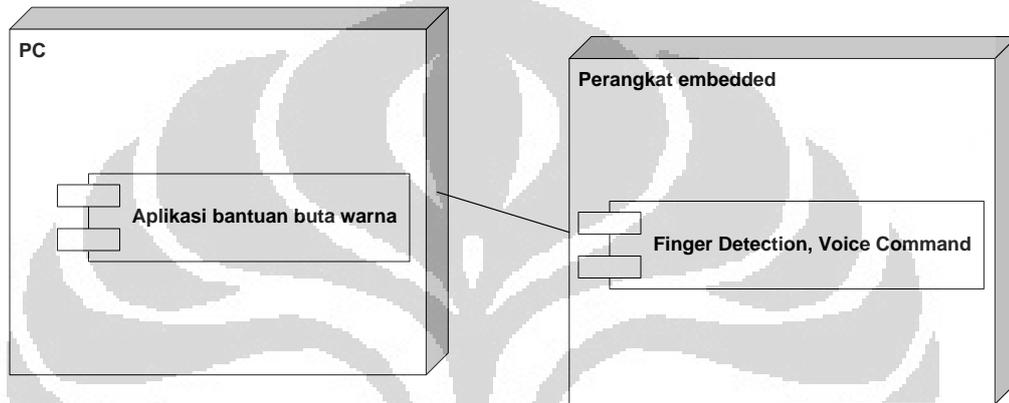
Gambar 3.7. *Class Diagram*

Kelas `InputStream` digunakan untuk mengambil masukan video dari kamera untuk diolah oleh sistem. `MotionTrack` digunakan untuk melacak pergerakan dari video yang diambil. Kelas `ColorDetect` melakukan fungsi pengenalan warna tertentu pada objek-objek yang terekam dalam video. Hasil pengenalan warna tersebut kemudian diolah oleh kelas `TextOverlay` menjadi informasi warna berupa teks untuk digabungkan dengan *video stream* yang masuk. Penggabungan dilakukan oleh kelas `OutputStream` dengan hasil akhir berupa video objek asli dengan tambahan informasi warna di dalamnya.

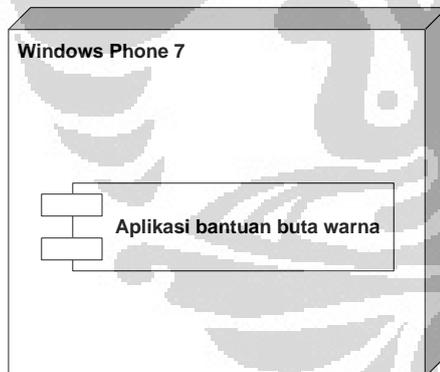
### 3.2.5 Deployment Diagram

*Deployment diagram* digunakan untuk merepresentasikan rancangan tata letak fisik dari sistem yang menunjukkan perangkat keras dan perangkat lunak dalam sistem tersebut [22]. Sistem bantuan penderita buta warna yang dirancang merupakan gabungan dari perangkat keras dan perangkat lunak seperti digambarkan pada Gambar 3.8 Perangkat lunak dari sistem ini juga dibuat pada platform Windows Phone 7.

1. Pada Platform Desktop PC



2. Pada Platform Windows Phone 7



Gambar 3.8. *Deployment Diagram*

## BAB 4

### IMPLEMENTASI MODUL DAN INTEGRASI SISTEM BANTUAN PENDERITA BUTA WARNA

#### 4.1 Implementasi Modul pada Aplikasi Untuk Sistem Tertanam

##### 4.1.1 Perangkat Pendukung Pengembangan Modul

Dalam proses implementasi dan pengembangan modul pendeteksian warna ini digunakan lingkungan pengembangan terintegrasi (*Integrated Development Environment; IDE*) sebagai sarana melakukan pengembangan program yang efektif dan efisien. Modul program akan diimplementasikan dalam .NET *Framework* dan bahasa pemrograman C# yang merupakan produk Microsoft.

Untuk melakukan pengembangan pada .NET *Framework* dan bahasa pemrograman tersebut, digunakan lingkungan pengembangan terintegrasi Visual Studio. Visual Studio merupakan produk Microsoft yang mendukung pengembangan aplikasi dalam bahasa pemrograman berbasis .NET seperti Visual Basic, C++, dan C #.

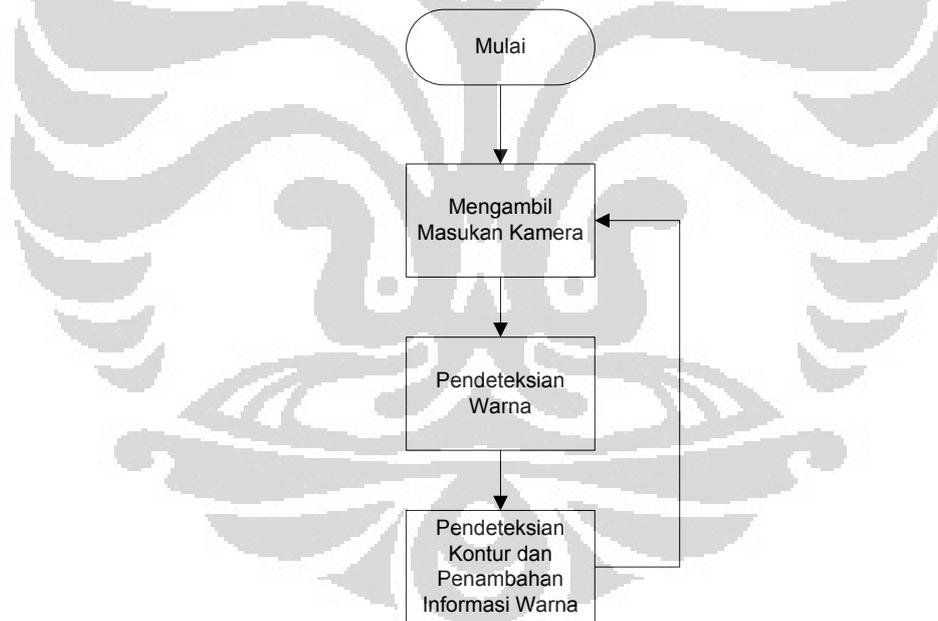
Terdapat dua macam lisensi Visual Studio, gratis dan berbayar. Lisensi gratis diberikan untuk program Visual Studio Express. Secara fungsional, Visual Studio Express cukup untuk digunakan dalam pengembangan aplikasi. Namun terdapat fungsi-fungsi pada Visual Studio yang hanya didukung oleh versi berbayar saja. Dalam pengembangan modul pendeteksian warna ini digunakan Visual Studio 2010 dengan lisensi akademik dari *Microsoft Software Developer Network-Academic Alliance (MSDN-AA)*.

Dalam pengembangan modul pendeteksian warna ini, digunakan *library* pihak ketiga (*third party library*) untuk pengolahan citra, yaitu EmguCV. Seperti telah dikemukakan sebelumnya, EmguCV merupakan wrapper .NET untuk *library* OpenCV yang merupakan *library open source* untuk aplikasi *computer vision* yang dikembangkan oleh Intel. *Library* ini digunakan untuk melakukan fungsi pengolahan citra untuk melakukan pengenalan warna.

#### 4.1.2 Pemrograman Modul Pendeteksian Warna dan Penjelasan Program

Setelah Visual Studio2010 dan EmguCV terpasang, selanjutnya implementasi sistem dimulai dengan pengembangan modul pendeteksian warna. Dengan mengacu pada rancangan sistem yang ada, modul ini dibuat pada sebuah *project* dengan jenis aplikasi Windows Presentation Foundation (WPF). Pada fase ini, pengembangan modul yang dilakukan berfokus untuk menghasilkan modul yang berjalan secara *stand-alone* untuk keperluan pengujian fungsi sebelum diintegrasikan ke dalam sistem secara keseluruhan.

Algoritma yang diterapkan dalam modul pendeteksian warna ini dapat dilihat pada Gambar 4.1. Langkah pertama adalah mengambil masukan gambar bergerak dari kamera. Selanjutnya, modul membaca masukan dari pengguna berupa warna yang akan dideteksi. Dalam implementasi modul ini masukan dari pengguna didapat dari *combobox* yang berisi pilihan warna yang dapat dideteksi.



Gambar 4.1. Algoritma dalam Implementasi Modul

Selanjutnya, pada modul terdapat method `GetColor()`, seperti terlihat pada Gambar 4.2. Modul ini dilakukan untuk melakukan pengenalan warna pada gambar yang masuk. Daerah yang mengandung warna diberi warna putih, dan daerah sisanya ditandai dengan warna hitam untuk pemrosesan selanjutnya. Di

akhir langkah ini, bagian pada gambar masukan yang mengandung warna yang diinginkan telah terdeteksi, termasuk derau (*noise*) yang terdapat pada masukan tersebut.

```

for (int i = hlsimage.Rows - 1; i >= 0; i--)
{
    for (int j = hlsimage.Cols - 1; j >= 0; j--)
    {

        if ((data[i, j, 0] > RefHue - ThreshHue &&
            data[i, j, 0] < RefHue + ThreshHue) &&
            (data[i, j, 1] > RefL - ThreshL &&
            data[i, j, 1] < RefL + ThreshL)

        )
        {
            data[i, j, 0] = 255;    //hue
            data[i, j, 1] = 255;    //lightness
            data[i, j, 2] = 0;      //saturation
        }
    }
}

```

Gambar 4.2 Fungsi GetColor() pada Program

Untuk menghasilkan pendeteksian warna objek yang diinginkan dengan lebih baik, derau ini diminimalisasi dengan langkah selanjutnya dengan method `DetectContour()`. Pada langkah ini dilakukan filterisasi dan pendeteksian kontur terbesar pada daerah yang telah dideteksi warnanya, dengan asumsi bahwa benda yang ingin dideteksi didekatkan ke kamera dan memiliki kontur terbesar. Kontur tersebut kemudian ditandai dan diberikan label warna. Program pada method `DetectContour()` ini terdapat pada Gambar 4.3.

```

CvInvoke.cvSmooth(hlsimage, hlsimage,
Emgu.CV.CvEnum.SMOOTH_TYPE.CV_MEDIAN, 17, 0, 0, 0);
CvInvoke.cvSplit(hlsimage, IntPtr.Zero, grayimage, IntPtr.Zero,
IntPtr.Zero);
CvInvoke.cvCanny(grayimage, cannyimage, 100, 50, 3);

Contour<System.Drawing.Point> contours =
cannyimage.FindContours(Emgu.CV.CvEnum.CHAIN_APPROX_METHOD.CV_CHAI
N_APPROX_SIMPLE,
                        Emgu.CV.CvEnum.RETR_TYPE.CV_RETR_TREE,
stor);

for (; contours != null; contours = contours.HNext)

```

```

{
if (contours.Area > 100) //only consider contours with area
greater than 100px
{

System.Drawing.Rectangle kotak = contours.BoundingRectangle;
int posisix = kotak.X;
int posisiy = kotak.Y;

MCvFont font = new
MCvFont (Emgu.CV.CvEnum.FONT.CV_FONT_HERSHEY_PLAIN, 0.5, 0.5);
OriginalHlsImage.Draw(kotak, new Hls(255,255,255), 1);
OriginalHlsImage.Draw(colorname, ref font, new
System.Drawing.Point(posisix, posisiy + 20), new
Hls(255,255,255));
}
}

```

Gambar 4.3. Fungsi DetectContour () pada Program

Implementasi modul ini dilakukan dalam dua model warna yaitu *Hue, Saturation, Value* (HSV) dan *Hue, Lightness, Saturation* (HLS). Penggunaan dua model warna ini digunakan untuk membandingkan hasil pendeteksian warna antara keduanya. Pada integrasi modul dengan sistem, digunakan model warna HLS.

Dalam implementasi yang dilakukan pada modul, terdapat perbedaan dengan rancangan sistem yang dibuat pada Bab 3. Perbedaan tersebut terletak pada rancangan algoritma yang digunakan. Pada implementasi yang dilakukan, tidak diterapkan algoritma pelacakan gerakan serta informasi berupa suara. Fitur suara diimplementasikan dalam integrasi modul ke dalam sistem, fitur suara ini memungkinkan pengguna menjalankan aplikasi dengan perintah suara (*voice command*).

#### 4.1.3 Hasil Implementasi Modul dan Penggunaan Modul.

Pemrograman modul pada subbab 4.1.2 menghasilkan modul *stand-alone* yang dapat digunakan. Sebelum program dijalankan, *listing* program terlebih dahulu harus dikompilasi dengan memilih menu Build > Build Solution pada Visual Studio. Selanjutnya program dapat dijalankan pada mode *debug* dengan

memilih menu Debug > Start Debugging. Program yang dihasilkan akan memiliki tampilan sederhana seperti terdapat pada Gambar 4.4.



Gambar 4.4. Gambar Antarmuka Modul Pendeteksian Warna untuk Sistem Tertanam

Seperti dijelaskan sebelumnya, pada antarmuka modul disediakan kontrol berupa *combobox* yang dapat digunakan pengguna untuk memilih warna yang ingin dideteksi dari objek yang ditangkap oleh kamera. Pada implementasi sistem yang telah terintegrasi, fungsi pemilihan warna ini dilakukan oleh modul *voice command*, sehingga untuk memilih warna yang ingin dideteksi, pengguna cukup menyebutkan nama warnanya saja.

Contoh penggunaan modul ini terdapat pada Gambar 4.5. Pada gambar tersebut ditunjukkan hasil pengenalan warna yang dipilih pengguna pada objek yang ditangkap kamera. Informasi warna ditampilkan dalam bentuk teks yang ditambahkan pada gambar bergerak yang ditampilkan oleh kamera.



Gambar 4.5. Gambar Modul Pendeteksian Warna Sedang Berjalan

#### 4.1.4 Integrasi Modul

Modul pengenalan warna untuk sistem tertanam ini diintegrasikan dengan modul lainnya yaitu pengenalan warna pada objek yang ditunjuk pengguna [4], peningkatan intensitas warna [5] serta modul *speech recognition* [6]. Proses integrasi yang dilakukan adalah sebagai berikut:

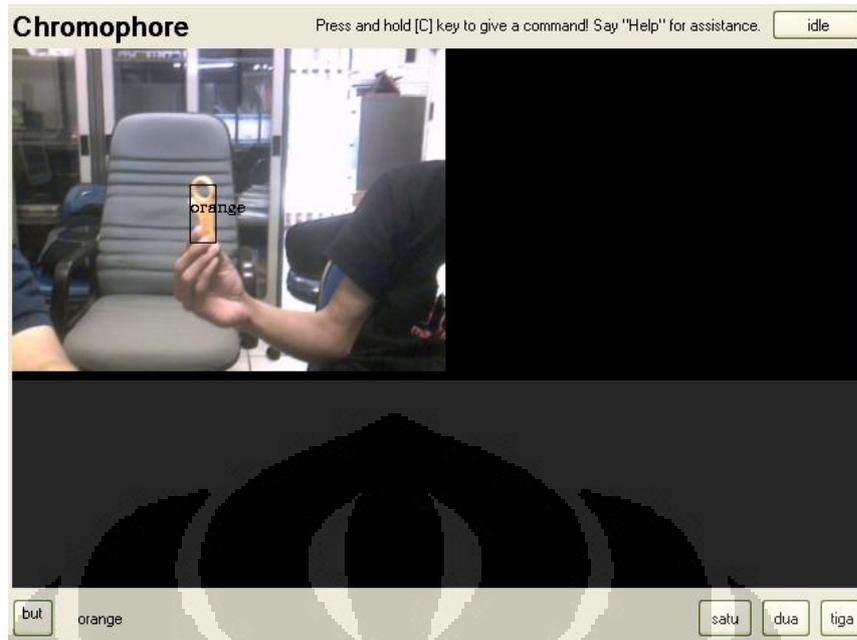
1. Pembuatan *class* terpisah untuk modul-modul yang ada. Pemisahan *class* ini dimaksudkan untuk memudahkan pengorganisasian fungsi serta memudahkan pemecahan masalah.
2. Pembuatan antarmuka sederhana untuk pengujian sistem.

Antarmuka program dibuat sederhana karena hanya ditujukan untuk keperluan pengujian. Implementasi sistem ini didesain sebagai aplikasi yang dapat berjalan dengan perangkat *head-mounted display*. Antarmuka yang ada digunakan sebagai alat bantu integrasi, sedangkan fungsi input untuk sistem dari pengguna yang sebenarnya dilakukan oleh modul *speech recognition*. Aplikasi yang telah diintegrasikan dipasang pada sebuah sistem tertanam yang terdiri atas perangkat Ebox dan periferal pendukung-pendukungnya

Sistem tertanam tersebut menggunakan sistem operasi Windows Embedded Standard 2009 yang dapat mendukung .NET Framework 3.5 yang digunakan untuk membuat aplikasi bantuan penderita buta warna ini. Windows Embedded merupakan sistem operasi untuk sistem tertanam yang dapat dikustomisasi sesuai kebutuhan pengguna. Pada sistem bantuan penderita buta warna ini, integrasi Windows Embedded sebagai sistem operasi untuk sistem tertanam dibahas secara detail pada [6].

#### 4.1.5 Hasil Integrasi Sistem Bantuan Penderita Buta Warna

Dengan langkah integrasi yang dilakukan sebelumnya dihasilkan satu sistem bantuan penderita buta warna yang lengkap yang dinamakan dengan Chromophore. Pada Gambar 4.6 diperlihatkan antarmuka program pada sistem yang sedang berjalan.



Gambar 4.6. Modul Pendeteksian Warna yang Telah Terintegrasi ke dalam Sistem.

## 4.2 Implementasi Modul Pada Aplikasi *Mobile* Pada Platform Windows Phone 7

Selain diimplementasikan ke dalam aplikasi untuk sistem tertanam, modul ini juga dibuat untuk diimplementasikan menjadi bagian dari sistem bantuan buta warna pada platform Windows Phone 7. Pada bab subbab ini dibahas mengenai instalasi perangkat pendukung pengembangan dan implementasi modul pada Windows Phone 7.

### 4.2.1 Instalasi Modul-Modul Pendukung Pengembangan Aplikasi Pada Platform Windows Phone 7

Untuk melakukan implementasi modul pada platform Windows Phone 7, pada dasarnya diperlukan langkah yang sama dengan langkah yang dilakukan pada implementasi modul pada sistem tertanam dengan *platform* .NET. Implementasi modul dilakukan dengan menggunakan lingkungan pengembangan terintegrasi Visual Studio 2010. Agar Visual Studio 2010 dapat digunakan untuk melakukan pengembangan aplikasi pada platform Windows Phone 7, diperlukan program tambahan yang disebut sebagai Windows Phone Development Tools.

Windows Phone Development Tools ini dapat diunduh secara gratis pada [23]. Tools ini disediakan di situsnya dalam bentuk berkas ISO dan dalam bentuk web installer. Berkas ISO memungkinkan pengguna melakukan instalasi secara *stand-alone* tanpa koneksi internet, sedangkan web installer mengharuskan pengguna terkoneksi ke internet saat melakukan instalasi.

#### 4.2.2 Pemrograman Modul Dan Penjelasan Kode Program

Metode yang digunakan dalam implementasi modul pada platform Windows Phone 7 berbeda dengan metode yang digunakan pada aplikasi untuk sistem tertanam. Hal ini disebabkan karena tidak didukungnya Emgu CV sebagai *library* untuk pengolahan citra pada platform Windows Phone 7.

Pada implementasi disini, pengolahan citra kamera tidak dilakukan sepenuhnya secara *real-time* terhadap citra yang tertangkap kamera. Proses pendeteksian warna dimulai dengan melakukan *capture* gambar objek yang diinginkan pengguna untuk dideteksi warnanya. Gambar inilah yang kemudian diolah dan ditunjukkan informasi warnanya kepada pengguna. *Listing* program untuk melakukan fungsi ini ditunjukkan pada Gambar 4.7.

```

for (h = 0; h < FrameHeight; h++)
{
    for (w = 0; w < FrameWidth; w++)
    {
        HLSColor HLSConvertedPxl =
        HLSColor.FromColor(writeableBmp.GetPixel(w, h));

        if ((HLSConvertedPxl.H > RefHue - ThreshHue &&
        HLSConvertedPxl.H < RefHue + ThreshHue) &&
            // HLSConvertedPxl.S > RefSat - ThreshSat &&
            // HLSConvertedPxl.S < RefSat + ThreshSat
            HLSConvertedPxl.L > refL - ThreshL &&
            HLSConvertedPxl.L < refL + ThreshL
        )
        {
            //set color of detected area
            writeableBmp.SetPixel(w, h, Colors.White);

            // set x & y for drawing rectangle bounding area with targeted
            color
            if (w > maxx) maxx = w; if (h > maxy) maxy = h;
            if (w < minx) minx = w; if (h < miny) miny = h;
        }
    }
}

```

```

else { writeableBmp.SetPixel(w, h, Colors.Black); }
}
}

if (maxx < 1 || maxy < 1 )
{
writeableBmp.Render(notfound, new TranslateTransform() { X = 50, Y
= 50 });
writeableBmp.Invalidate();
}
if (minx < 800 && miny < 800)
{
writeableBmp.DrawRectangle(minx, miny, maxx, maxy, Colors.White);
writeableBmp.Render(notif, new TranslateTransform() { X = minx +
10, Y = miny + 10 });
writeableBmp.Invalidate();

// DEBUG PURPOSE:
/* HLSColor DetectedColor =
HLSColor.FromColor(writeableBmp.GetPixel(maxx, maxy));
redbox.Text = DetectedColor.H.ToString();
greenbox.Text =
DetectedColor.S.ToString();
bluebox.Text =
DetectedColor.L.ToString();

//detectedcolor info display
Htxt.Opacity = 1;
Stxt.Opacity = 1;
Ltxt.Opacity = 1;
detectedtxt.Opacity = 1;
redbox.Opacity = 1;
greenbox.Opacity = 1;
bluebox.Opacity = 1; */
}
}
}

```

Gambar 4.7. Kode Program Modul Pendeteksian Warna pada Platform Windows Phone 7.

Cara kerja program dapat dilihat berdasarkan *listing* program tersebut. Ketika modul dijalankan, program akan mengambil masukan dari kamera. *Frame* masukan kamera tersebut kemudian diubah ke dalam model warna HLS. Model warna HLS dipilih dengan hipotesis bahwa model warna ini menghasilkan tingkat pengenalan warna yang lebih baik. Selanjutnya *frame* gambar dibaca dan diolah untuk mengetahui daerah dengan daerah warna yang dikehendaki. Daerah dengan warna yang dimaksud kemudian ditandai dan kemudian ditambahkan informasi teks nama warna pada daerah tersebut

#### 4.2.3 Hasil Implementasi Modul

Dengan pemrograman modul pada bagian sebelumnya, didapatkan hasil berupa modul pengenalan warna. Gambar 4.8 menunjukkan modul yang sedang berjalan yang menunjukkan hasil pengenalan warna pada gambar hasil *capture* yang dipilih oleh pengguna. Dengan antarmuka modul program tersebut, pengguna dapat memilih warna yang akan dideteksi, melakukan *capture* gambar pada daerah yang diinginkan dan reset hasil *capture* gambar.



Gambar 4.8. Antarmuka Modul Program pada Aplikasi *Mobile*

#### 4.2.4 Integrasi Modul ke Dalam Sistem

Modul pengenalan warna pada Windows Phone 7 ini juga merupakan bagian dari sistem bantuan penderita buta warna Chromophore. Fitur-fitur yang dimiliki oleh Chromophore ini adalah pengenalan warna, pengetesan jenis buta warna dan peningkatan intensitas warna.

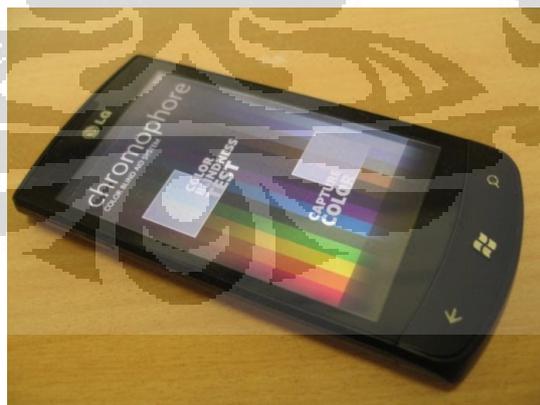
Modul pengujian jenis buta warna dan peningkatan jenis buta warna diuraikan secara lebih rinci pada [5]. Modul pengujian jenis buta warna ditujukan untuk membantu pengguna mengetahui apakah pengguna memiliki penglihatan normal atau memiliki kelainan buta warna. Pengujian jenis buta warna ini dilakukan dengan piringan Ishihara. Modul peningkatan intensitas warna ditujukan untuk membantu penyandang buta warna untuk mengenali warna-warna objek kurang kontras, sehingga menjadi ambigu untuk dikenali oleh penyandang buta warna.

Integrasi pada Windows Phone 7 dilakukan dengan langkah-langkah berikut:

1. Membuat *project* baru pada Visual Studio 2010 dengan jenis Silverlight for Windows Phone.
2. Pada Chromophore ini akan digunakan layout halaman dengan jenis Panorama, yang merupakan tata letak halaman khas pada Windows Phone 7, sehingga pada Application Type perlu dipilih jenis aplikasi Panorama.
3. Penggabungan kode program dari modul-modul yang ada dan melakukan pengecekan apakah terdapat pesan kesalahan akibat integrasi program yang dilakukan.
4. Penyesuaian antarmuka program untuk mengakomodasi fungsi

#### 4.2.5 Hasil Integrasi Sistem pada Windows Phone 7

Dari hasil integrasi yang dilakukan, dihasilkan aplikasi Chromophore sebagai sistem bantuan penderita buta warna. Pada Gambar 4.9 diperlihatkan tampilan awal program ketika dijalankan. Pada halaman awal tersebut terdapat pilihan bagi pengguna untuk melakukan pengetesan buta warna, atau menggunakan fungsi pengenalan warna dan transformasi warna.



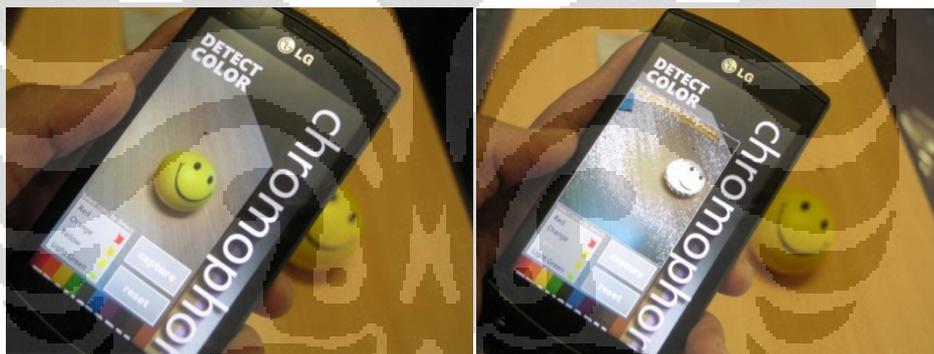
Gambar 4.9. Antarmuka Aplikasi *Mobile* Hasil Integrasi

Jika pengguna memilih pilihan melakukan pengetesan buta warna, akan terdapat antarmuka program yang diperlihatkan pada Gambar 4.10. Bila pengguna

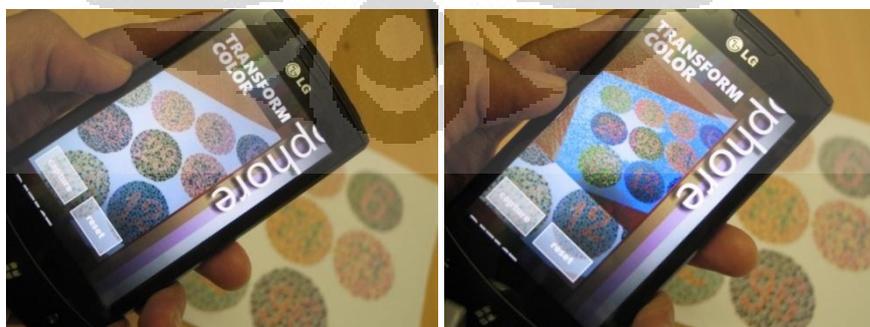
memilih fungsi pengenalan warna, maka pengguna dapat melakukan fungsi-fungsi yang hasilnya ditunjukkan masing-masing pada Gambar 4.11 dan Gambar 4.12. Pada Gambar 4.11, objek dengan warna yang terdeteksi ditandai dengan warna putih serta batas berupa kotak. Pada Gambar 4.12, terlihat bahwa warna objek yang ditangkap telah dinaikkan nilai kontrasnya.



Gambar 4.10. Modul Ishihara Test



Gambar 4.11. Modul Pendeteksian Warna Hasil Integrasi



Gambar 4.12. Modul Transformasi Warna Hasil Integrasi

## **BAB 5**

### **PENGUJIAN DAN ANALISA MODUL PENDETEKSIAN WARNA**

#### **5.1. PengujianFungsionalitas Modul Pendeteksian Warna**

##### **5.1.1 Pengujian Modul Untuk Sistem Tertanam**

Pada modul yang dibuat untuk diintegrasikan pada sistem bantuan penderita buta warna dilakukan dua macam pengujian, yakni pengujian teknis (kuantitatif) dan pengujian kepada sampel pengguna (kualitatif). Pengujian teknis yang dilakukan adalah pengujian terhadap tingkat kesuksesan pengenalan warna dengan menggunakan dua model warna yang berbeda, yaitu HSV dan HLS. Pengujian ini ditujukan untuk membandingkan tingkat kesuksesan pengenalan warna antara kedua model warna tersebut. Pengujian kedua adalah pengujian untuk mengetahui pengaruh kondisi pencahayaan terhadap tingkat kesuksesan pendeteksian warna.

Pada pengujian yang berkaitan dengan kesuksesan pendeteksian warna, digunakan kertas sampel warna yang menjadi dasar nilai komponen *hue*, *lightness* dan *saturation* yang diatur pada modul program. Walaupun terdapat pergeseran nilai warna akibat pencetakan, penggunaan kertas sampel warna ini dimaksudkan untuk mensimulasikan kondisi ideal pendeteksian warna yang mampu dilakukan oleh program. Selain faktor pergeseran warna dari warna sampel, faktor eksternal yang mempengaruhi kesuksesan pendeteksian tersebut adalah kualitas kamera yang digunakan dalam pengujian. Pada pengujian untuk sistem tertanam ini, digunakan kamera *webcam built-in* dengan kualitas menengah-rendah dengan resolusi 1.3 *megapixel*.

##### **5.1.1.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.**

Uji kesuksesan pengenalan warna ini dilakukan dengan menggunakan sampel warna tercetak dengan warna yang mendekati daerah warna yang diatur pada program. Pengujian dilakukan terhadap dua modul yang menggunakan dua model warna berbeda, yaitu HSV dan HLS. Pengujian dilakukan sebanyak 10 kali untuk masing-masing sampel warna. Pengujian dilakukan dengan kondisi pencahayaan ruangan yang terang yang menggunakan lampu *flourescent*.

#### 5.1.1.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan.

Pengujian dengan variasi kondisi pencahayaan ini ditujukan untuk mengetahui pengaruh intensitas cahaya terhadap tingkat kesuksesan pengenalan warna. Pengujian kedua ini juga dilakukan terhadap dua model warna berbeda, HSV dan HLS. Pengujian dilakukan sebanyak 10 kali untuk masing-masing kondisi pencahayaan dan model warna. Kondisi cahaya yang diuji meliputi kondisi gelap, sedang dan terang.

#### 5.1.1.3 Pengujian pada Sampel Pengguna.

Pengujian pada sampel pengguna ditujukan untuk mengetahui aspek pada modul yang terkait dengan penggunaan oleh sampel pengguna. Aspek tersebut termasuk persepsi warna yang digunakan oleh sistem dari sudut pandang pengguna, kemudahan penggunaan, dan lain sebagainya.

### 5.1.2 Pengujian Modul untuk Aplikasi *Mobile*

Pada modul untuk aplikasi *mobile* juga dilakukan pengujian yang hampir sama dengan pengujian pada modul untuk sistem tertanam, yaitu pengujian teknis (kuantitatif) dan pengujian oleh pengguna (kualitatif).

#### 5.1.2.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.

Uji kesuksesan pengenalan warna ini dilakukan dengan menggunakan sampel warna tercetak dengan warna yang mendekati daerah warna yang diatur pada program untuk aplikasi *mobile* ini. Sampel warna tercetak yang digunakan sedikit berbeda dengan sampel warna yang digunakan pada pengujian yang dilakukan pada subbab 5.1.1. Perbedaan ini disebabkan perbedaan rentang warna yang digunakan pada model warna pada kedua sistem. Pengujian dilakukan terhadap dua modul yang menggunakan dua model warna berbeda, yaitu HLS dan RGB. Pengujian dilakukan sebanyak 10 kali untuk masing-masing sampel warna. Pengujian dilakukan dengan kondisi pencahayaan ruangan yang terang yang menggunakan lampu *flourescent*.

Sama halnya dengan pengujian pada modul untuk sistem tertanam, pada pengujian yang berkaitan dengan kesuksesan pendeteksian warna ini, digunakan kertas sampel warna dengan rentang nilai warna yang dipakai pada modul

program itu sendiri. Faktor eksternal yang mempengaruhi pengujian ini adalah pergeseran warna akibat proses pencetakan sampel warna dan kamera yang digunakan dalam pengujian. Pada pengujian yang dilakukan, digunakan kamera pada *handheld* LG E900 Optimus 7, dengan resolusi sebesar 5.0 *megapixel*.

#### 5.1.2.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan.

Pengujian dengan variasi kondisi pencahayaan ini ditujukan untuk mengetahui pengaruh intensitas cahaya terhadap tingkat kesuksesan pengenalan warna. Pengujian kedua ini juga dilakukan terhadap dua model warna berbeda, HLS dan RGB. Pengujian dilakukan sebanyak 10 kali untuk masing-masing kondisi pencahayaan dan model warna. Kondisi cahaya yang diuji meliputi kondisi gelap, sedang dan terang.

#### 5.1.2.3 Pengujian Pada Sampel Pengguna.

Pengujian pada sampel pengguna ditujukan untuk mengetahui aspek-aspek pada modul yang terkait dengan penggunaan oleh sampel pengguna. Aspek tersebut termasuk persepsi warna yang digunakan oleh sistem dari sudut pandang pengguna, kemudahan penggunaan, rancangan antarmuka program, dan lain sebagainya. Pengujian dilakukan dengan sampel lima pengguna normal dan lima pengguna buta warna parsial untuk membandingkan hasil pengujian antara pengguna normal dan pengguna dengan buta warna parsial.

### 5.2 Hasil Pengujian dan Analisis

#### 5.2.1. Hasil Pengujian dan Analisis Modul untuk Sistem Tertanam

##### 5.2.1.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.

Pada pengujian yang dilakukan terhadap modul pengenalan warna untuk sistem tertanam ini, didapatkan hasil seperti terdapat pada Tabel 5.1 dan Tabel 5.2. Tabel 5.1 berisi hasil pengujian dengan menggunakan model warna HSV, sedangkan Tabel 5.2 menunjukkan hasil pengujian dengan model warna HLS. Kolom pertama pada tabel menunjukkan sampel warna yang diuji. Kolom kedua

menunjukkan jumlah ketidaksuksesan pendeteksian pada warna yang bersesuaian. Kolom ketiga berisi jumlah pendeteksian warna yang berhasil. Kolom keempat menunjukkan persentase kesuksesan dalam pendeteksian warna dari sepuluh kali pengujian yang dilakukan.

Tabel 5.1. Hasil Pengujian Modul Untuk Sistem Tertanam Menggunakan Model Warna HSV

Warna Referensi	Tidak Sukses	Sukses	Persentase Kesuksesan (%)
Merah	1	9	90
Jingga	1	9	90
Kuning	0	10	100
Hijau Muda	0	10	100
Hijau	3	7	70
Hijau Tosca	6	4	40
Biru Muda	7	3	30
Biru	1	9	90
Ungu	0	10	100
Magenta	0	10	100
Hijau Tua	2	8	80
Jingga Tua	2	8	80
Hijau-Kuning	3	7	70
Abu-abu Tua	10	0	0
Merah Tua	10	0	0
	<b>Rata-rata</b>		69.33

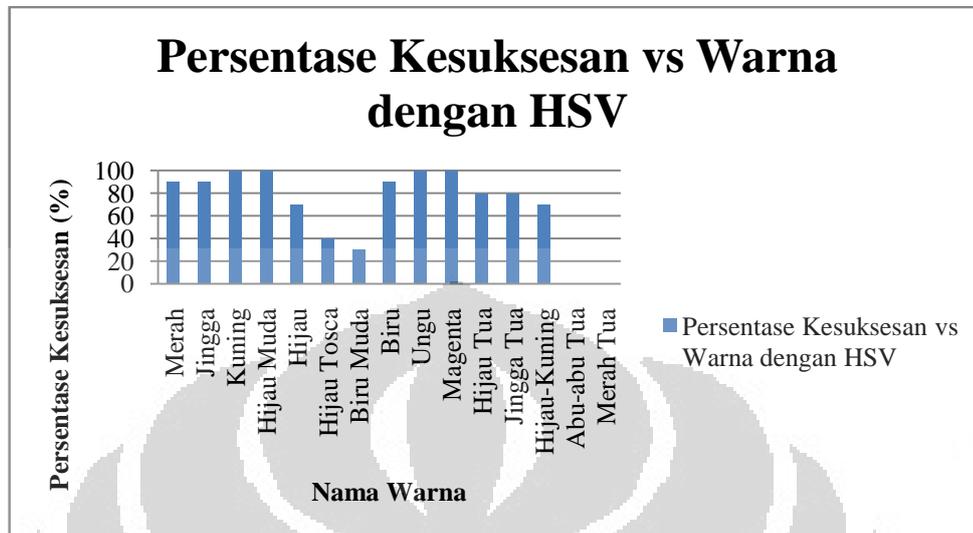
Tabel 5.2. Hasil Pengujian Modul Untuk Sistem Tertanam Menggunakan Model Warna HLS

Warna Referensi	Tidak Sukses	Sukses	Persentase Kesuksesan (%)
Merah	1	9	90
Jingga	0	10	100
Kuning	0	10	100
Hijau Muda	0	10	100
Hijau	2	8	80
Hijau Tosca	0	10	100
Biru Muda	5	5	50
Biru	1	9	90
Ungu	0	10	100
Magenta	0	10	100
Hijau Tua	2	8	80
Jingga Tua	0	10	100
Hijau-Kuning	1	9	90
Abu-abu Tua	2	8	80
Merah Tua	0	10	100
<b>Rata-rata</b>			90.67

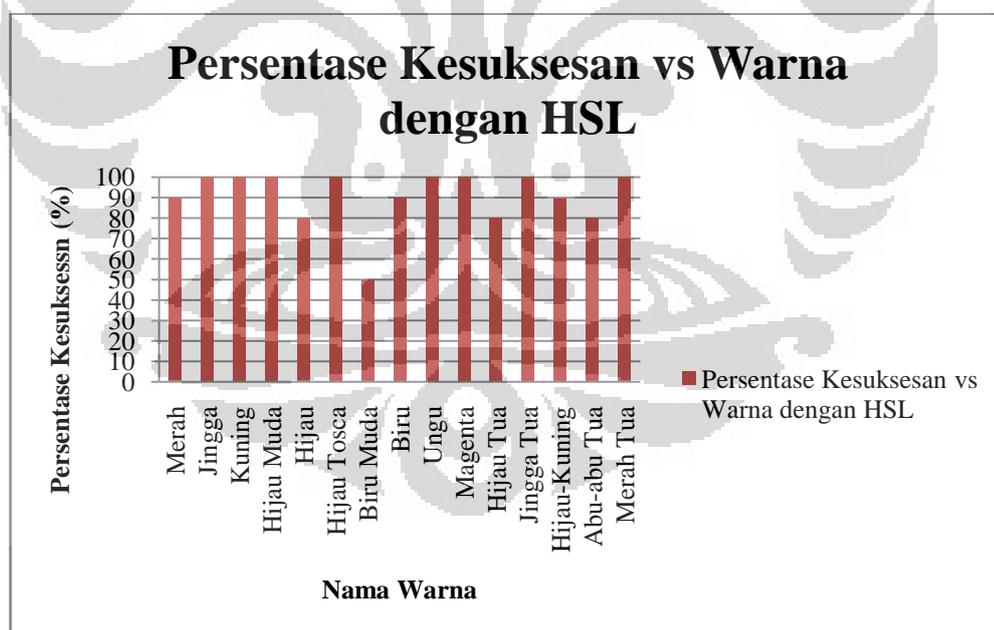
Berdasarkan tabel data pengujian tersebut dapat dihasilkan grafik persentase kesuksesan pendeteksian untuk masing-masing sampel warna yang diuji. Sumbu X pada grafik menunjukkan nama pada sampel warna yang diuji, sedangkan sumbu Y menunjukkan persentase keberhasilan pendeteksian warna pada 10 kali percobaan pada pengujian yang dilakukan. Gambar 5.1 menunjukkan grafik hasil pengujian dengan menggunakan model warna HSV. Grafik hasil pengujian dengan menggunakan model warna HLS ditunjukkan pada Gambar 5.2.

Dengan mengacu pada tabel dan grafik pada pengujian modul menggunakan sampel warna, terlihat bahwa kedua model warna dapat menghasilkan tingkat kesuksesan pengenalan di atas 50%. Namun demikian, dapat dilihat bahwa model warna HLS menghasilkan tingkat kesuksesan pengenalan warna rata-rata lebih tinggi dibandingkan dengan model warna HSV. Dari 15 sampel warna yang diujikan, HLS menghasilkan persentase rata-rata

sebesar 90,67% sedangkan HSV menghasilkan persentase rata-rata sebesar 69,33%.



Gambar 5.1. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HSV pada Modul Sistem Tertanam.



Gambar 5.2. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS pada Modul Sistem Tertanam.

Perbedaan persentase kesuksesan ini disebabkan oleh perbedaan karakteristik model warna itu sendiri. Pada modul yang diimplementasikan dengan HSV, komponen *hue* dan *saturation*, digunakan sebagai parameter penentuan warna objek. *Hue* merupakan komponen utama yang menentukan warna, sedangkan *saturation* menyatakan tingkat kecerahan warna. Dengan hanya digunakannya dua komponen tersebut, maka modul hanya mampu mendeteksi warna yang masuk ke dalam rentang *hue* dan *saturation* yang ditentukan dengan toleransi yang diatur secara statis. Dengan kata lain hanya dapat mendeteksi rentangan warna antara warna yang ditentukan dan warna putih saja.

Pada modul dengan model warna HLS, digunakan komponen Hue dan Lightness. Sama halnya dengan model warna HSV, Hue pada HLS merupakan komponen utama yang berisi informasi warna. Komponen *lightness* pada HLS dapat menentukan tingkat kecerahan dan tingkat kegelapan dari warna yang ditentukan oleh komponen Hue. Dengan demikian, HLS mampu mendeteksi warna dalam rentangan warna yang ditentukan ke arah hitam atau ke arah putih. Dengan kata lain, HLS dapat mendeteksi sebuah warna dan warna-warna yang lebih gelap dari warna tersebut, atau sebuah warna dan warna-warna lain yang lebih cerah dari warna tersebut.

#### 5.2.1.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan.

Pengujian ini dilakukan untuk mengetahui pengaruh tingkat pencahayaan terhadap modul pendeteksian warna untuk sistem tertanam. Pengujian juga dilakukan dengan menggunakan dua model warna, HSV dan HLS. Tabel 5.3 menunjukkan hasil pengujian dengan model warna HSV dan Tabel 5.4 menunjukkan hasil pengujian dengan HLS. Kolom pertama tabel menunjukkan nama sampel warna yang diujikan. Kolom kedua, ketiga, dan keempat menunjukkan jumlah kesuksesan pendeteksian pada warna yang bersesuaian, masing-masing pada kondisi cahaya kurang, sedang, dan terang. Persentase keberhasilan pendeteksian warna dalam sepuluh kali pengujian yang dilakukan ditunjukkan pada kolom kelima, keenam, dan ketujuh.

Tabel 5.3. Hasil Pengujian Variasi Pencahayaan pada Modul Untuk Sistem Tertanam Menggunakan Model Warna HSV

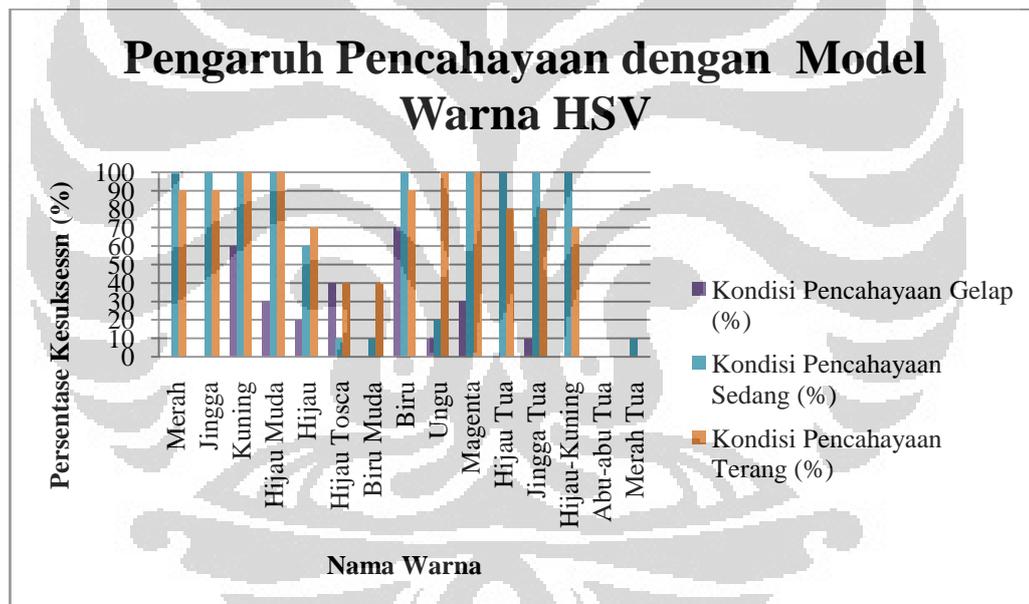
Warna	Kondisi Pencahayaan		
	Gelap (%)	Sedang (%)	Terang (%)
Merah	0	100	90
Jingga	0	100	90
Kuning	60	100	100
Hijau Muda	30	100	100
Hijau	20	60	70
Hijau Tosca	40	10	40
Biru Muda	0	10	30
Biru	70	100	90
Ungu	10	20	100
Magenta	30	100	80
Hijau Tua	0	100	80
Jingga Tua	10	100	70
Hijau-Kuning	0	100	0
Abu-abu Tua	0	0	0
Merah Tua	0	10	0
<b>Rata-rata</b>	18	67.33	69.33

Tabel 5.4. Hasil Pengujian Variasi Pencahayaan pada Modul Untuk Sistem Tertanam Menggunakan Model Warna HLS

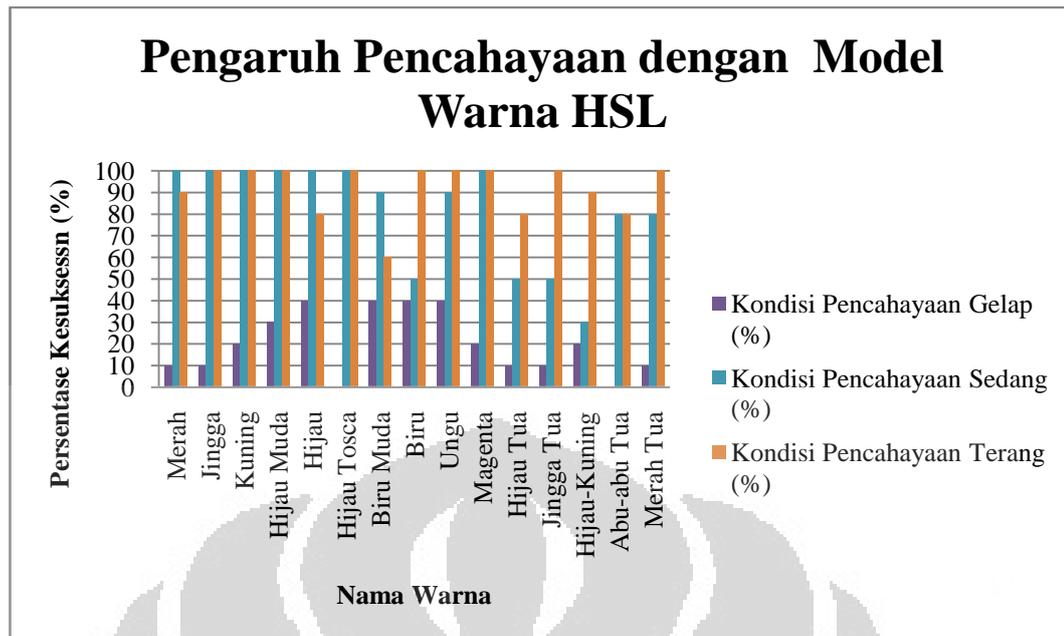
Warna	Kondisi Pencahayaan		
	Gelap (%)	Sedang (%)	Terang (%)
Merah	10	100	90
Jingga	10	100	100
Kuning	20	100	100
Hijau Muda	30	100	100
Hijau	40	100	80
Hijau Tosca	0	100	100
Biru Muda	40	90	50
Biru	40	50	90
Ungu	40	90	100
Magenta	20	100	100
Hijau Tua	10	50	80

Jingga Tua	10	50	100
Hijau-Kuning	20	30	90
Abu-abu Tua	0	80	80
Merah Tua	10	80	100
<b>Rata-rata</b>	20	81.33	90.67

Dari Tabel 5.3 dan Tabel 5.4 tersebut dihasilkan grafik tingkat kesuksesan pendeteksian pada sampel warna dengan kondisi cahaya yang berbeda-beda. Gambar 5.3 menunjukkan grafik untuk model warna HSV dan Gambar 5.4 memperlihatkan grafik untuk model warna HLS. Sumbu X dan sumbu Y masing-masing menyatakan persentase kesuksesan dan nama warna sampel yang diujikan.



Gambar 5.3. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HSV pada Modul Sistem Tertanam dengan Variasi Pencahayaan.



Gambar 5.4. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS pada Modul Sistem Tertanam dengan Variasi Pencahayaan.

Berdasarkan tabel dan grafik dari pengujian dengan variasi pencahayaan di atas dapat dilakukan beberapa analisis. Pada kondisi cahaya terang dan sedang, kedua model warna masih dapat melakukan pendeteksian warna dengan baik. Tabel 5.3 menunjukkan bahwa HSV dapat menghasilkan persentase kesuksesan rata-rata sebesar 69.33% pada kondisi cahaya terang dan 67.33% pada kondisi cahaya sedang. Tabel 5.4 menunjukkan bahwa HLS menghasilkan persentase kesuksesan sebesar 90.67% pada pencahayaan terang dan 81.33% pada pencahayaan sedang. Persentase rata-rata pada kedua model warna masih tinggi karena pada pencahayaan terang dan sedang, sampel warna yang terbaca masih mendekati kondisi aslinya, dimana nilai *hue* dan *saturation* pada HSV maupun *hue* dan *lightness* pada HLS tidak banyak bergeser dari rentang nilai yang ditentukan pada modul. Dari kedua tabel tersebut juga terlihat bahwa HLS masih menunjukkan persentase kesuksesan lebih tinggi dibandingkan HSV.

Pada kondisi pencahayaan gelap, persentase kesuksesan pada kedua model warna menurun. HSV menghasilkan persentase kesuksesan rata-rata sebesar 18%, sedangkan HLS menghasilkan persentase kesuksesan rata-rata sebesar 20%. Hal

ini menunjukkan bahwa kondisi cahaya gelap memberikan pengaruh signifikan. Pada model warna HSV, pencahayaan gelap mempengaruhi nilai komponen *hue*, *saturation* dan *value*. Namun demikian, *saturation* dan *value* lebih terpengaruh signifikan dibandingkan dengan Hue. Pada modul yang ada hanya digunakan komponen *saturation* saja, sehingga bila warna sampel yang akan dideteksi cenderung berwarna gelap, modul dengan model warna HSV tidak dapat melakukan pendeteksian dengan baik.

Dengan model warna HLS, pencahayaan gelap mempengaruhi komponen Hue, Lightness dan Saturation. Pada model ini, Lightness merupakan komponen yang terpengaruh secara signifikan terhadap pencahayaan gelap. Walaupun pada modul dengan model warna HLS digunakan komponen *hue* dan *lightness* sebagai faktor penentu dalam pendeteksian warna, persentase kesuksesan pendeteksian dengan HLS juga menunjukkan angka yang rendah, yaitu sebesar 20%. Hal ini disebabkan karena nilai toleransi *lightness* yang diatur secara statis pada program, sehingga kemampuan pendeteksian terbatas pada nilai toleransi tersebut.

#### 5.2.1.3 Pengujian Pada Sampel Pengguna.

Dari pengujian penggunaan modul yang telah terintegrasi pada sistem tertanam terhadap sampel pengguna, didapatkan beberapa hasil. Dari hasil pengujian pengguna tersebut, dapat dilakukan perhitungan 95% *confidence interval* (95 % Interval Keyakinan) dengan persamaan sebagai berikut:

$$\text{Rata - rata} \pm \left( \frac{1,96 \times \text{Standar Deviasi}}{\sqrt{\text{Populasi}}} \right) \quad (5.1)$$

Tabel 5.5 menunjukkan hasil pengujian terhadap persepsi warna pengguna normal terhadap sampel warna yang digunakan dalam sistem. Tabel 5.7 menunjukkan hasil pengujian terhadap persepsi warna pengguna dengan buta warna parsial terhadap sampel warna yang digunakan dalam sistem. Pada Tabel 5.6 ditunjukkan hasil pengujian terhadap pengalaman pengguna normal dalam menggunakan sistem, sedangkan pada Tabel 5.8 ditunjukkan hasil pengujian terhadap pengalaman pengguna dengan buta warna parsial.

Tabel 5.5. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Normal Terhadap Sampel Warna dalam Sistem Tertanam

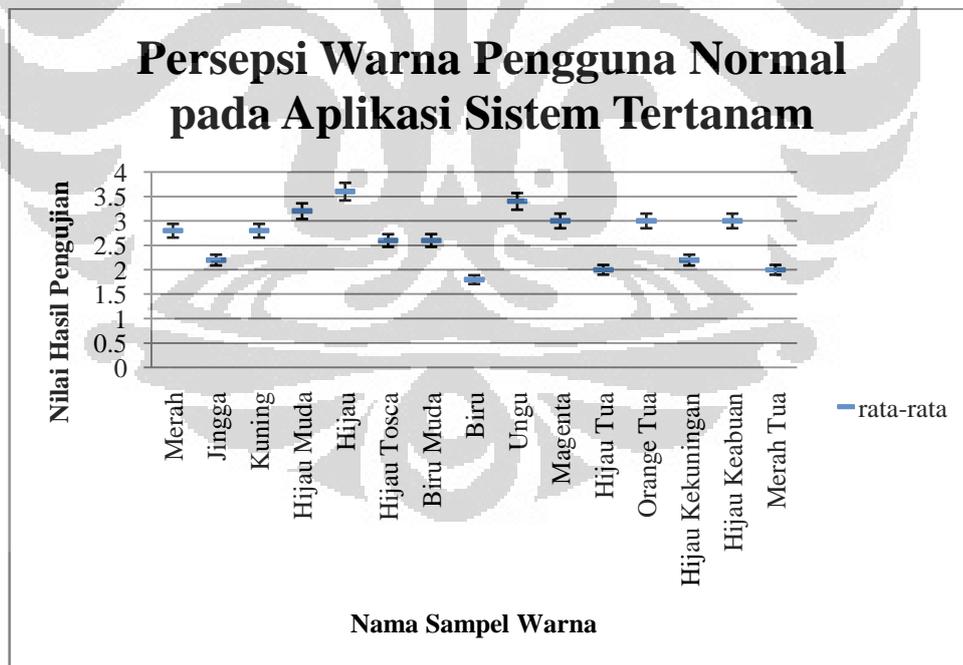
Nama Warna	Perhitungan Interval Keyakinan 95 %			
	Rata-Rata	Toleransi	Batas Bawah	Batas Atas
<b>Merah</b>	2.8	0.392	2.408	3.192
<b>Jingga</b>	2.2	0.392	1.808	2.592
<b>Kuning</b>	2.8	0.733	2.067	3.533
<b>Hijau Muda</b>	3.2	0.733	2.467	3.933
<b>Hijau</b>	3.6	0.480	3.120	4.080
<b>Hijau Tosca</b>	2.6	0.480	2.120	3.080
<b>Biru Muda</b>	2.6	0.480	2.120	3.080
<b>Biru</b>	1.8	0.392	1.408	2.192
<b>Ungu</b>	3.4	0.480	2.920	3.880
<b>Magenta</b>	3	0.000	3.000	3.000
<b>Hijau Tua</b>	2	0.620	1.380	2.620
<b>Orange Tua</b>	3	0.000	3.000	3.000
<b>Hijau Kekuningan</b>	2.2	0.392	1.808	2.592
<b>Hijau Keabuan</b>	3	1.240	1.760	4.240
<b>Merah Tua</b>	2	0.000	2.000	2.000

Tabel 5.6. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Normal pada Sistem Tertanam

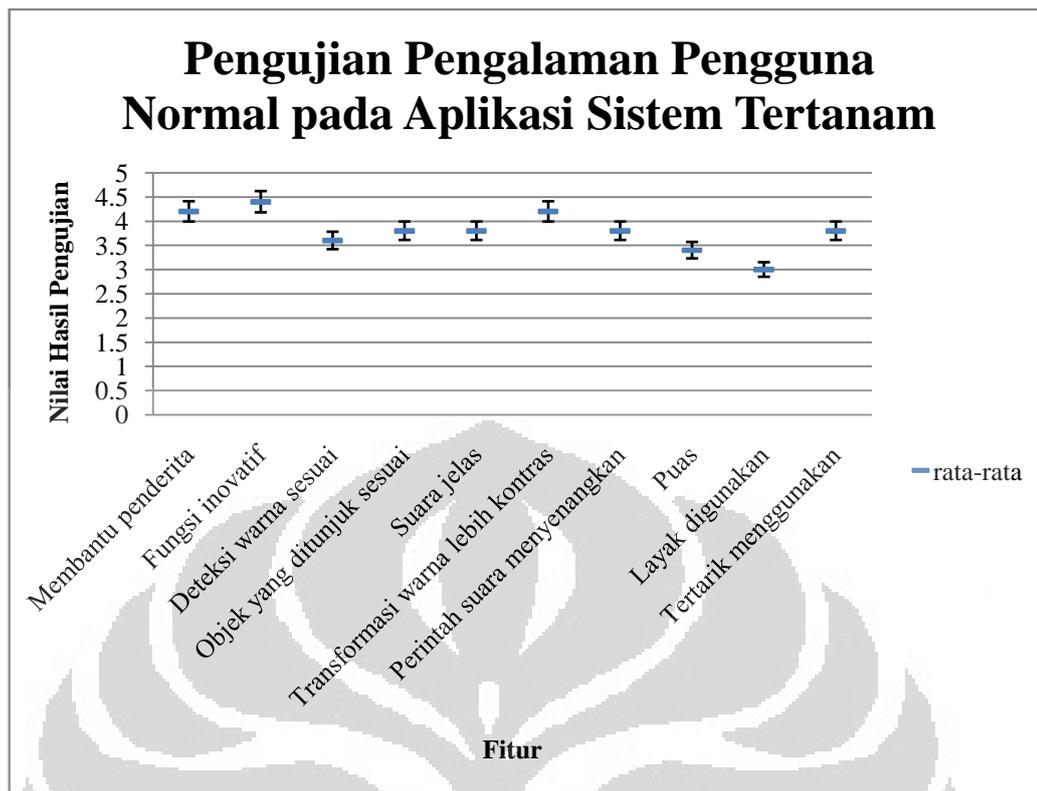
No.	Pernyataan	Perhitungan Interval Keyakinan 95 %			
		Rata-Rata	Toleransi	Batas Bawah	Batas Atas
1	Program membantu penderita buta warna	4.2	0.392	3.808	4.592
2	Fungsi-fungsi program inovatif	4.4	0.480	3.920	4.880
3	Pada Detect Color (mode 1), hasil program sesuai dengan warna sebenarnya	3.6	0.784	2.816	4.384
4	Pada Finger Detection ((mode 2), objek yang ditunjuk dan warnanya sesuai dengan kondisi sebenarnya.	3.8	0.733	3.067	4.533
5	Suara yang diucapkan di mode 2 jelas bagi saya	3.8	0.392	3.408	4.192
6	Pada Transform Color (mode 3), warna yang diubah menjadi lebih kontras	4.2	0.733	3.467	4.933
7	Memerintah dengan suara lebih menyenangkan	3.8	0.960	2.840	4.760
8	Anda puas menggunakan program ini	3.4	0.480	2.920	3.880
9	Aplikasi sudah layak digunakan banyak pengguna	3	0.877	2.123	3.877
10	Jika dipublikasikan, Anda tertarik menggunakan aplikasi ini	3.8	0.733	3.067	4.533

Berdasarkan Tabel 5.5 dan Tabel 5.6 di atas, dihasilkan grafik pada Gambar 5.5 dan Gambar 5.6. Gambar 5.5 merupakan grafik pengujian persepsi sampel warna terhadap pengguna normal. Sumbu X menyatakan nama sampel warna dan sumbu Y menyatakan nilai hasil pengujian pada pengguna. Gambar 5.6 menunjukkan grafik pengujian pengalaman pengguna pada sistem tertanam, sumbu X pada grafik tersebut menunjukkan pernyataan yang diujikan, nilai hasil pengujian pengguna ditunjukkan pada sumbu Y.

Pada pengujian terhadap persepsi warna di modul untuk sistem tertanam ini, dari pengolahan data didapatkan bahwa rata-rata pengujian normal memberikan nilai rata-rata 2.68 pada sampel warna yang ada. Berdasarkan proses pengujian yang dilakukan, didapatkan hasil bahwa persepsi sampel pengujian terhadap warna berbeda-beda, nama dari sampel warna yang ada pun turut memberikan pengaruh terhadap persepsi warna pengujian. Ketidaksesuaian persepsi warna juga mungkin diakibatkan oleh pergeseran warna saat pencetakan.



Gambar 5.5. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Sistem Tertanam.



Gambar 5.6. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Sistem Tertanam.

Dari pengujian yang berkaitan dengan pengalaman pengguna normal, kisaran nilai rata-rata sebesar 3.8 yang berarti sebagian besar pernyataan yang ada disetujui oleh pengguna. Dalam kaitannya dengan modul pendeteksian warna, Pernyataan 3, disetujui oleh pengguna dengan nilai rata-rata sebesar 3.6. Nilai tersebut menunjukkan bahwa modul sudah cukup baik dalam memenuhi ekspektasi pengguna normal.

Pengujian juga dilakukan terhadap pengguna dengan buta warna parsial. Pengujian ini ditujukan untuk membandingkan persepsi warna pengguna dengan buta warna parsial dengan persepsi pengguna normal serta pengalaman penggunaan aplikasi oleh penderita buta warna parsial. Hasil pengujian pada penderita buta warna ditunjukkan pada Tabel 5.7 dan Tabel 5.8.

Tabel 5.7. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Buta Warna Parsial Terhadap Sampel Warna dalam Sistem Tertanam

Nama Warna	Perhitungan Interval Keyakinan 95 %			
	Rata-Rata	Toleransi	Batas Bawah	Batas Atas
<b>Merah</b>	3.6	0.480	3.120	4.080
<b>Jingga</b>	2.6	0.999	1.601	3.599
<b>Kuning</b>	3	0.620	2.380	3.620
<b>Hijau Muda</b>	3.4	0.480	2.920	3.880
<b>Hijau</b>	3.8	0.392	3.408	4.192
<b>Hijau Tosca</b>	2.4	0.480	1.920	2.880
<b>Biru Muda</b>	3.2	0.733	2.467	3.933
<b>Biru</b>	2.8	0.733	2.067	3.533
<b>Ungu</b>	3.4	0.784	2.616	4.184
<b>Magenta</b>	3.4	0.480	2.920	3.880
<b>Hijau Tua</b>	2	0.000	2.000	2.000
<b>Orange Tua</b>	2.6	0.999	1.601	3.599
<b>Hijau Kekuningan</b>	1.8	1.143	0.657	2.943
<b>Hijau Keabuan</b>	2.4	0.480	1.920	2.880
<b>Merah Tua</b>	2.6	0.999	1.601	3.599

Tabel 5.8. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Buta Warna Parsial pada Sistem Tertanam

No.	Pernyataan	Perhitungan Interval Keyakinan 95 %			
		Rata-Rata	Toleransi	Batas Bawah	Batas Atas
1	Program membantu penderita buta warna	4.2	0.733	3.467	4.933
2	Fungsi-fungsi program inovatif	4.6	0.480	4.120	5.080
3	Pada Detect Color (mode 1), hasil program sesuai dengan warna sebenarnya	3.8	0.733	3.067	4.533
4	Pada Finger Detection (mode 2), objek yang ditunjuk dan warnanya sesuai dengan kondisi sebenarnya.	3.6	0.480	3.120	4.080
5	Suara yang diucapkan di mode 2 jelas bagi saya	3	0.877	2.123	3.877
6	Pada Transform Color (mode 3), warna yang diubah menjadi lebih kontras	4.2	0.392	3.808	4.592
7	Memerintah dengan suara lebih menyenangkan	4.4	0.480	3.920	4.880
8	Anda puas menggunakan program ini	3.8	0.733	3.067	4.533
9	Aplikasi sudah layak digunakan banyak pengguna	3.8	0.360	3.440	4.160
10	Jika dipublikasikan, Anda tertarik menggunakan aplikasi ini	3.8	1.143	2.657	4.943

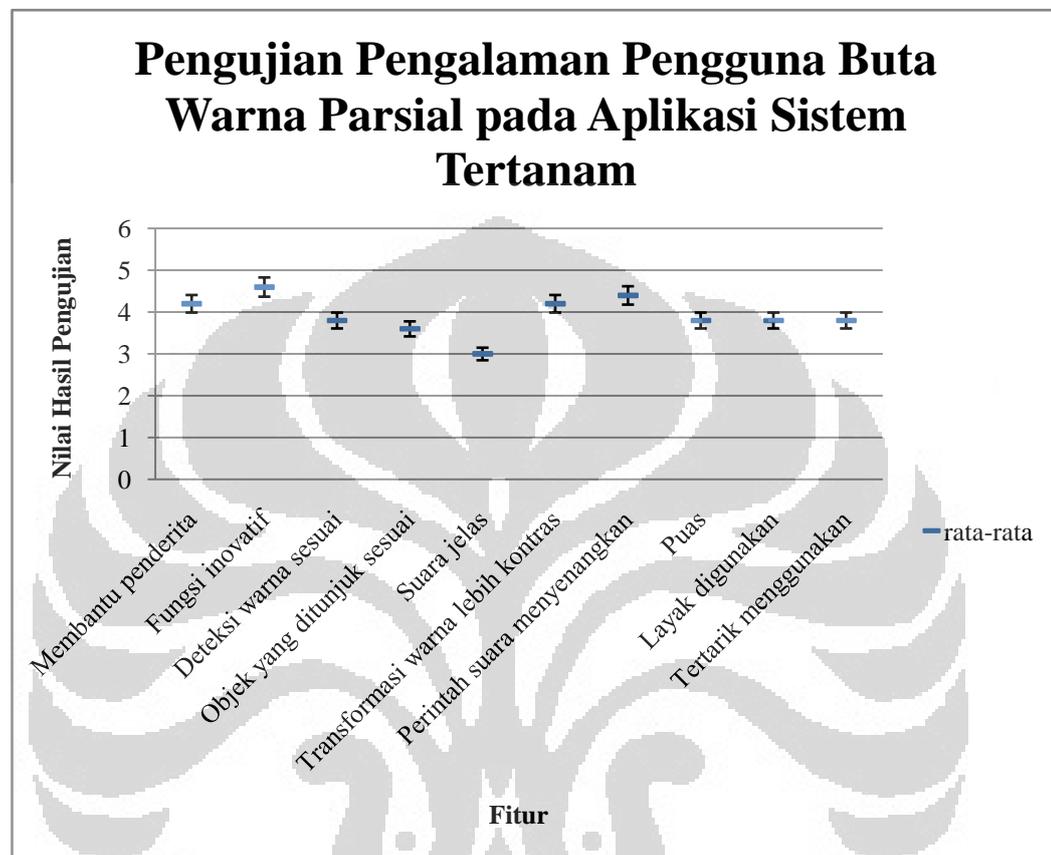
Berdasarkan Tabel 5.7 dan Tabel 5.8 di atas, dihasilkan grafik pada Gambar 5.7 dan Gambar 5.8. Pada pengujian terhadap persepsi warna di modul untuk sistem tertanam ini, dari pengolahan data didapatkan bahwa rata-rata penguji dengan buta warna parsial memberikan nilai rata-rata 2.86 pada sampel warna yang ada. Berdasarkan proses pengujian yang dilakukan, didapatkan hasil bahwa persepsi sampel penguji terhadap warna berbeda-beda, nama dari sampel warna yang ada pun turut memberikan pengaruh terhadap persepsi warna penguji. Persentase yang lebih tinggi dibandingkan pengguna normal diakibatkan karena ketiadaan persepsi warna tertentu pada pengguna dengan buta warna parsial. Ketidaksesuaian persepsi warna juga mungkin diakibatkan oleh pergeseran warna saat pencetakan.



Gambar 5.7. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Buta Warna Parsial pada Sistem Tertanam.

Dari pengujian yang berkaitan dengan pengalaman pengguna buta warna parsial, kisaran nilai rata-rata sebesar 4.2 yang berarti sebagian besar pernyataan yang ada disetujui oleh pengguna. Dalam kaitannya dengan modul pendeteksian warna, Pernyataan 3, disetujui oleh pengguna dengan nilai rata-rata sebesar 3.8.

Nilai tersebut menunjukkan bahwa modul sudah cukup baik dalam memenuhi ekspektasi pengguna buta warna parsial.



Gambar 5.8. Grafik Hasil Pengujian Pengalaman Pengguna Buta Warna Parsial pada Sistem Tertanam.

## 5.2.2 Hasil Pengujian dan Analisis Modul Untuk Aplikasi *Mobile*

### 5.2.2.1 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Model Warna.

Pada pengujian yang dilakukan pada modul pengenalan warna untuk aplikasi *mobile*, didapatkan hasil seperti terdapat pada Tabel 5.7 dan Tabel 5.8. Pada modul untuk aplikasi *mobile* ini digunakan model warna HLS dan RGB. Tabel 5.7 berisi hasil pengujian dengan menggunakan model warna RGB, sedangkan Tabel 5.8 menunjukkan hasil pengujian dengan model warna HLS. Seperti halnya pada tabel hasil pengujian pada pengujian modul untuk sistem

tertanam, kolom pertama tabel menunjukkan nama warna yang diuji. Kolom kedua menunjukkan jumlah kegagalan dalam pendeteksian warna yang bersangkutan. Kolom ketiga berisi jumlah keberhasilan dalam pendeteksian. Kolom keempat berisi persentase keberhasilan pendeteksian warna dalam sepuluh kali pengujian yang dilakukan.

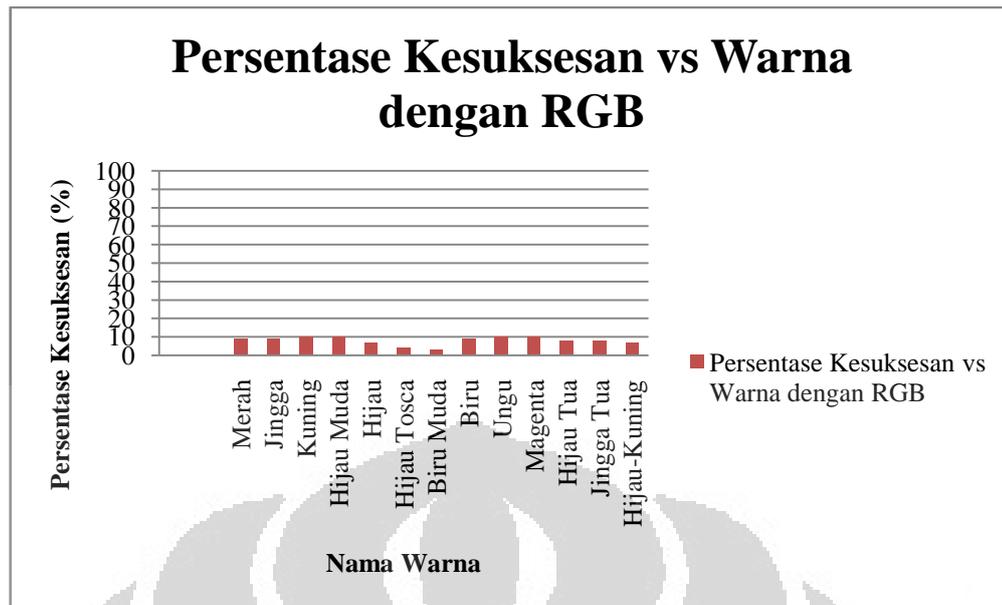
Tabel 5.9. Hasil Pengujian Modul Untuk Aplikasi Mobile dengan Menggunakan Model Warna RGB

Warna Referensi	Tidak Sukses	Sukses	Persentase Kesuksesan (%)
Merah	10	0	0
Jingga	10	0	0
Kuning	10	0	0
Hijau Muda	10	0	0
Hijau	10	0	0
Hijau Tosca	10	0	0
Biru Muda	10	0	0
Biru	10	0	0
Ungu	10	0	0
Magenta	10	0	0
Hijau Tua	5	5	50
Jingga Tua	10	0	0
Hijau-Kuning	10	0	0
Abu-abu Tua	1	9	90
Merah Tua	7	3	30
<b>Rata-rata</b>			<b>11.33</b>

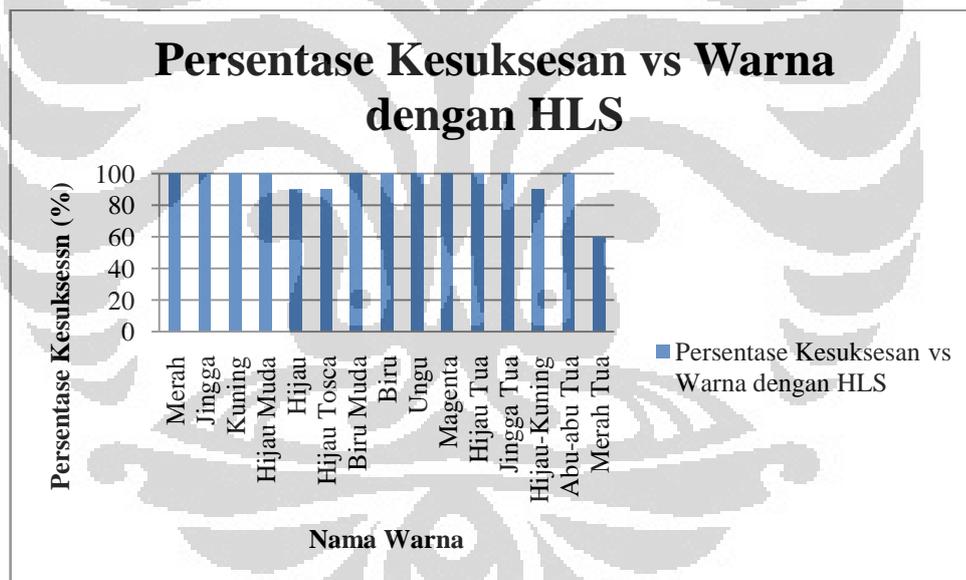
Tabel 5.10. Hasil Pengujian Modul Untuk Aplikasi Mobile dengan Menggunakan Model Warna HLS

Warna Referensi	Tidak Sukses	Sukses	Persentase Kesuksesan (%)
Merah	0	10	100
Jingga	0	10	100
Kuning	0	10	100
Hijau Muda	0	10	100
Hijau	1	9	90
Hijau Tosca	1	9	90
Biru Muda	0	10	100
Biru	0	10	100
Ungu	0	10	100
Magenta	0	10	100
Hijau Tua	0	10	100
Jingga Tua	0	10	100
Hijau-Kuning	1	9	90
Abu-abu Tua	0	10	100
Merah Tua	4	6	60
<b>Rata-rata</b>			95.33

Berdasarkan tabel data pengujian tersebut dapat dihasilkan grafik persentase kesuksesan pendeteksian untuk masing-masing sampel warna yang diuji pada modul tersebut. Sumbu X pada grafik menunjukkan nama pada sampel warna yang diuji, sedangkan sumbu Y menunjukkan persentase keberhasilan pendeteksian warna pada 10 kali pengujian yang dilakukan. Gambar 5.7 menunjukkan grafik hasil pengujian dengan menggunakan model warna RGB. Grafik hasil pengujian dengan menggunakan model warna HLS ditunjukkan pada Gambar 5.8.



Gambar 5.9. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna RGB.



Gambar 5.10. Grafik Persentase Kesuksesan Pendeteksian Warna dengan Model Warna HLS.

Berdasarkan Tabel 5.9 dan Gambar 5.9, terlihat bahwa model warna RGB hanya memberikan persentase kesuksesan pengenalan rata-rata sebesar 11.33 %, sedangkan Tabel 5.10 menunjukkan model warna HLS menunjukkan persentase rata-rata sebesar 95.33%. Hasil ini menunjukkan bahwa modul dengan model

warna HLS jauh lebih baik dalam mengenali warna. Pada modul dengan model warna RGB, digunakan tiga komponen yaitu *red*, *green*, dan *blue* untuk menentukan warna yang akan dideteksi. Nilai komponen-komponen tersebut ditentukan berdasarkan warna-warna sampel yang ditentukan. Dengan model warna RGB, agar berhasil dideteksi warna objek harus memenuhi kriteria ketiga komponen tersebut. Walaupun telah diberikan nilai toleransi, kondisi ini sulit dicapai dan mengakibatkan persentase kesuksesan yang rendah dengan RGB.

Pada Tabel 5.9 ditunjukkan bahwa persentase kesuksesan dengan model warna HLS rata-rata bernilai 95.33%. Serupa dengan modul untuk sistem tertanan dengan menggunakan model warna HLS, pada modul untuk aplikasi *mobile* ini juga digunakan dua komponen, yaitu *hue* dan *lightness* sebagai faktor penentu warna yang akandideteksi, dimana *hue* merupakan komponen utama berisi informasi warna, sedangkan *lightness* menentukan terang-gelapnya warna tersebut.

#### 5.2.2.2 Pengujian Tingkat Kesuksesan Pendeteksian Warna dengan Variasi Kondisi Pencahayaan.

Pengujian ini dilakukan untuk mengetahui pengaruh tingkat pencahayaan terhadap modul pendeteksian warna untuk aplikasi *mobile*. Pengujian juga dilakukan dengan menggunakan dua model warna, RGB dan HLS. Tabel 5.11 menunjukkan hasil pengujian dengan model warna RGB dan Tabel 5.12 menunjukkan hasil pengujian dengan HLS. Kolom pertama tabel menunjukkan nama sampel warna yang diujikan. Kolom kedua, ketiga, dan keempat menunjukkan jumlah kesuksesan pendeteksian pada warna yang bersesuaian, masing-masing pada kondisi cahaya kurang, sedang, dan terang. Persentase keberhasilan pendeteksian warna dalam sepuluh kali pengujian yang dilakukan ditunjukkan pada kolom kelima, keenam, dan ketujuh.

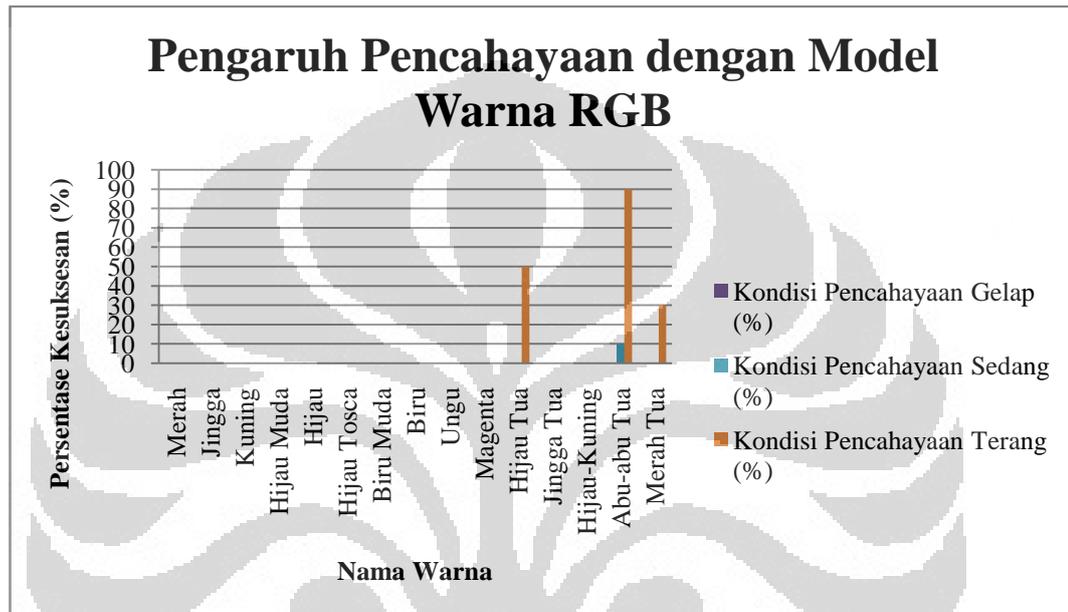
Tabel 5.11. Hasil Pengujian Variasi Pencahayaan pada Modul Pendeteksian Warna Untuk Aplikasi Mobile dengan Model Warna RGB

Warna	Kondisi Pencahayaan		
	Gelap (%)	Sedang (%)	Terang (%)
Merah	0	0	0
Jingga	0	0	0
Kuning	0	0	0
Hijau Muda	0	0	0
Hijau	0	0	0
Hijau Tosca	0	0	0
Biru Muda	0	0	0
Biru	0	0	0
Ungu	0	0	0
Magenta	0	0	0
Hijau Tua	0	0	50
Jingga Tua	0	0	0
Hijau-Kuning	0	0	0
Abu-abu Tua	0	10	90
Merah Tua	0	0	30
<b>Rata-rata</b>	0	0.67	11.33

Tabel 5.12. Hasil Pengujian Variasi Pencahayaan pada Modul Pendeteksian Warna Untuk Aplikasi Mobile dengan Model Warna HLS

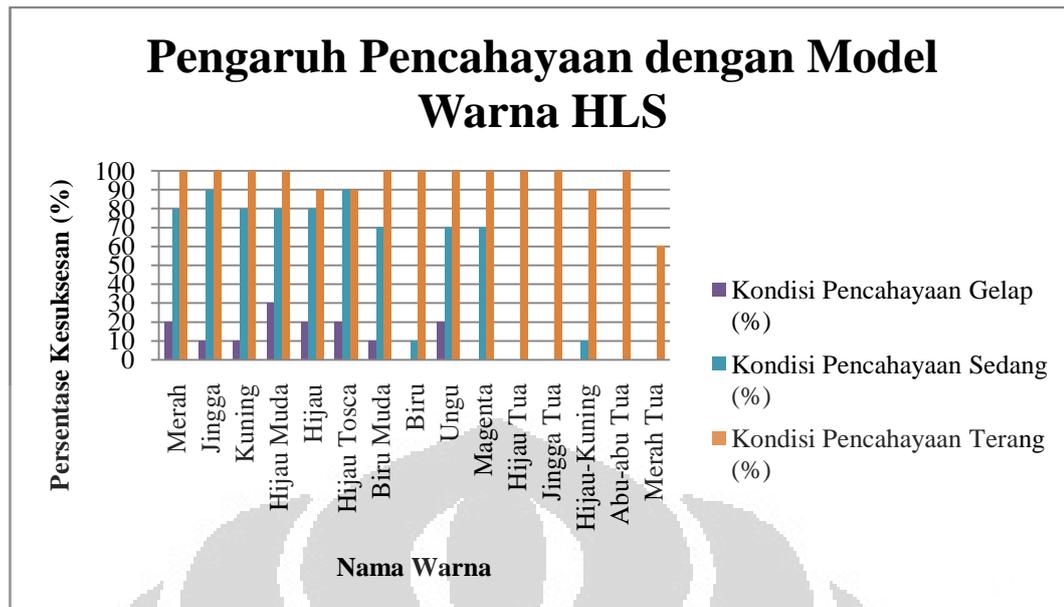
Warna	Kondisi Pencahayaan		
	Gelap (%)	Sedang (%)	Terang (%)
Merah	20	80	100
Jingga	10	90	100
Kuning	10	80	100
Hijau Muda	30	80	100
Hijau	20	80	90
Hijau Tosca	20	90	90
Biru Muda	10	70	100
Biru	0	10	100
Ungu	20	70	100
Magenta	0	70	100
Hijau Tua	0	0	100
Jingga Tua	0	0	100
Hijau-Kuning	0	10	90
Abu-abu Tua	0	0	100
Merah Tua	0	0	60
<b>Rata-rata</b>	9.33	48.67	95.33

Berdasarkan Tabel 5.11 dan Tabel 5.12 di atas, dapat dibuat grafik persentase kesuksesan pendeteksian warna pada kondisi pencahayaan yang berbeda-beda. Gambar 5.11 menunjukkan grafik untuk model warna RGB, sedangkan Gambar 5.12 menunjukkan grafik pengujian model warna HLS. Sumbu X dan sumbu Y masing-masing menyatakan persentase kesuksesan dan nama warna sampel yang diujikan.



Gambar 5.11. Grafik Pengujian Variasi Pencahayaan pada Modul Untuk Aplikasi *Mobile* dengan Model Warna RGB

Berdasarkan tabel dan grafik pengujian variasi pencahayaan pada modul untuk aplikasi *mobile*, terlihat bahwa pada semua kondisi pencahayaan, modul dengan model warna RGB menunjukkan persentase pengenalan sampel warna yang rendah, yaitu 0%, 0.67%, dan 11.33%, masing-masing pada pencahayaan gelap, sedang, dan terang. Seperti telah dijelaskan dalam analisis sebelumnya, hal ini diakibatkan karena dengan model warna RGB, ketiga komponen warna tersebut harus terpenuhi agar warna objek dapat terdeteksi. Namun demikian, terlihat bahwa warna hijau tua, abu-abu tua dan merah tua masih dapat terdeteksi pada pencahayaan terang, hal ini menunjukkan bahwa pada pencahayaan terang, warna yang terbaca oleh kamera tidak banyak bergeser dengan warna referensi yang ada pada modul.



Gambar 5.12. Grafik Pengujian Variasi Pencahayaan pada Modul Untuk Aplikasi *Mobile* dengan Model Warna HLS.

Pada hasil pengujian variasi pencahayaan pada modul dengan model warna HLS, didapatkan persentase kesuksesan sebesar 95.33% pada pencahayaan terang dan 48.67% pada pencahayaan sedang. Sedangkan pada pencahayaan gelap, persentase kesuksesan hanya tercapai sebesar 9.33%. Hal ini disebabkan karena kondisi pencahayaan gelap mempengaruhi nilai komponen *lightness* dari objek yang akan dideteksi. Modul memberikan persentase pendeteksian rendah untuk kondisi pencahayaan gelap karena pada kondisi pencahayaan tersebut, nilai *lightness* dari objek cenderung meningkat sehingga objek yang dideteksi tidak masuk dalam rentang nilai toleransi *lightness* yang diatur pada program.

### 5.2.2.3 Pengujian pada Sampel Pengguna.

Dari pengujian penggunaan modul yang telah terintegrasi pada sistem bantuan penderita buta warna pada platform *mobile* terhadap sampel pengguna, didapatkan beberapa hasil. Hasil tersebut kemudian diolah dengan menggunakan persamaan yang sama dengan persamaan pada subbab 5.2.1.3 pada pengujian pengguna sistem tertanam. Tabel 5.11 memperlihatkan hasil pengujian persepsi warna pengguna normal terhadap sampel warna dalam sistem, sedangkan Tabel

5.12 menunjukkan hasil pengujian terhadap pengalaman pengguna normal dalam penggunaan sistem.

Tabel 5.13. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Normal Terhadap Sampel Warna dalam Aplikasi *Mobile*

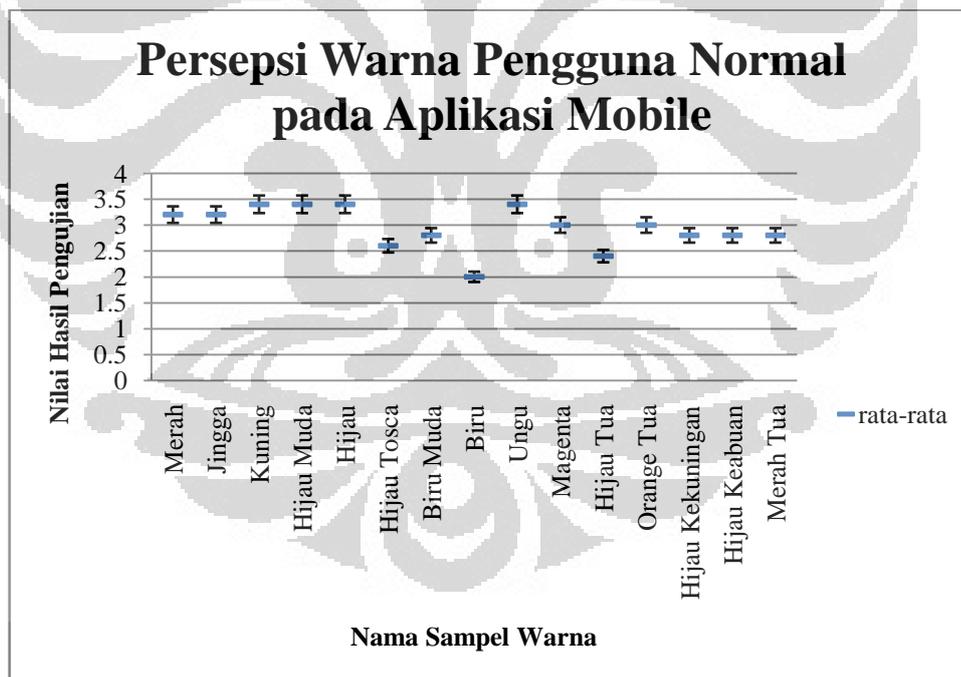
Nama Warna	Perhitungan Interval Keyakinan 95 %			
	Rata-Rata	Toleransi	Batas Bawah	Batas Atas
<b>Merah</b>	3.2	0.392	2.808	3.592
<b>Jingga</b>	3.2	0.392	2.808	3.592
<b>Kuning</b>	3.4	0.480	2.920	3.880
<b>Hijau Muda</b>	3.4	0.480	2.920	3.880
<b>Hijau</b>	3.4	0.480	2.920	3.880
<b>Hijau Tosca</b>	2.6	0.480	2.120	3.080
<b>Biru Muda</b>	2.8	0.733	2.067	3.533
<b>Biru</b>	2	0.000	2.000	2.000
<b>Ungu</b>	3.4	0.480	2.920	3.880
<b>Magenta</b>	3	0.620	2.380	3.620
<b>Hijau Tua</b>	2.4	0.784	1.616	3.184
<b>Orange Tua</b>	3	0.000	3.000	3.000
<b>Hijau Kekuningan</b>	2.8	0.733	2.067	3.533
<b>Hijau Keabuan</b>	2.8	0.733	2.067	3.533
<b>Merah Tua</b>	2.8	1.143	1.657	3.943

Tabel 5.14. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Normal pada Aplikasi *Mobile*

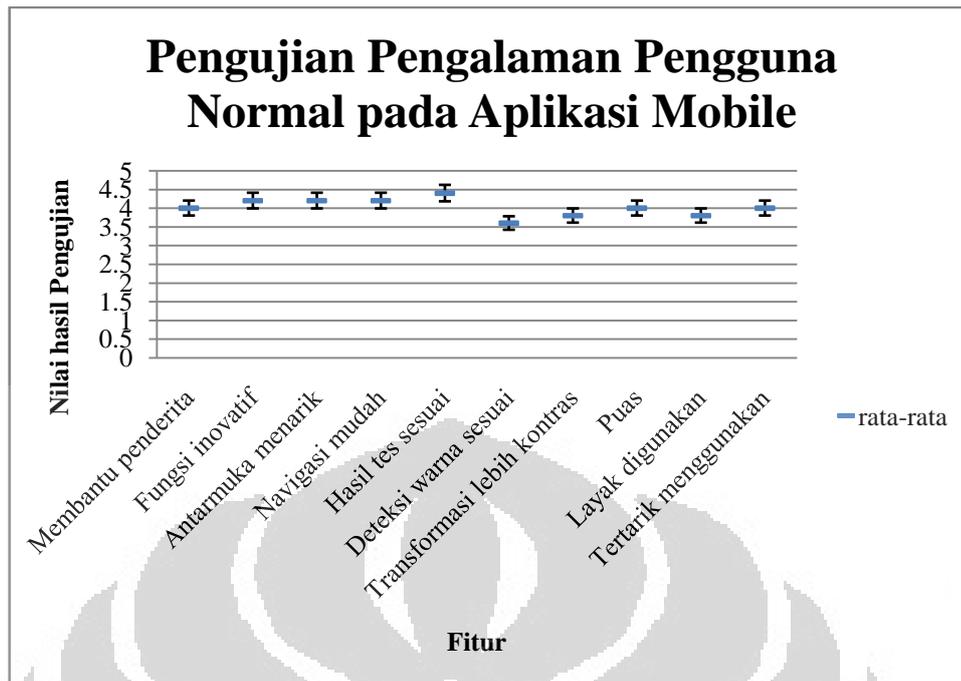
No.	Pernyataan	Perhitungan Interval Keyakinan 95 %			
		Rata-Rata	Toleransi	Batas Bawah	Batas Atas
<b>1</b>	Program membantu penderita buta warna	4	0.620	3.380	4.620
<b>2</b>	Fungsi-fungsi program inovatif	4.2	0.392	3.808	4.592
<b>3</b>	Antarmuka (tampilan) program menarik	4.2	0.392	3.808	4.592
<b>4</b>	Antarmuka (tampilan) dan navigasi program mudah digunakan	4.2	0.733	3.467	4.933
<b>5</b>	Pada Color Blindness Test, hasil program sesuai dengan kondisi Anda	4.4	0.480	3.920	4.880
<b>6</b>	Pada Detect Color, hasil program sesuai dengan warna sebenarnya	3.6	0.480	3.120	4.080
<b>7</b>	Pada Transform Color, warna yang diubah menjadi lebih kontras dan terlihat	3.8	0.392	3.408	4.192
<b>8</b>	Anda puas menggunakan program ini	4	0.000	4.000	4.000
<b>9</b>	Aplikasi sudah layak digunakan banyak pengguna	3.8	0.392	3.408	4.192
<b>10</b>	Jika dipublikasikan, Anda tertarik menggunakan aplikasi ini	4	0.620	3.380	4.620

Dari Tabel 5.13 dan Tabel 5.14 hasil pengujian di atas, dihasilkan grafik pada Gambar 5.13 dan Gambar 5.14. Gambar 5.13 merupakan grafik pengujian persepsi sampel warna terhadap pengguna normal, dimana sumbu X menyatakan nama sampel warna dan sumbu Y menyatakan nilai hasil pengujian pada pengguna. Gambar 5.14 menunjukkan grafik pengujian pengalaman pengguna, sumbu X pada grafik tersebut menunjukkan pernyataan yang diujikan, nilai hasil pengujian pengguna ditunjukkan pada sumbu Y.

Berdasarkan pengujian persepsi warna pengguna terhadap sampel warna, rata-rata sebesar 2.94. Pada pengujian dengan aplikasi mobile ini digunakan sampel warna yang berbeda dengan sampel warna pada pengujian modul untuk sistem tertanam Nilai rata-rata hasil pengujian persepsi warna pada aplikasi mobile lebih tinggi dibandingkan nilai pengujian pada modul sistem tertanam. Hal ini menunjukkan bahwa rentang warna yang digunakan pada aplikasi *mobile* lebih mendekati persepsi sampel pengguna normal.



Gambar 5.13. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Aplikasi *Mobile*



Gambar 5.14. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Aplikasi *Mobile*

Pada pengujian pengalaman pengguna, terkait dengan modul pendeteksian warna didapatkan nilai rata-rata  $3.6 \pm 0.480$ . Hal ini menunjukkan bahwa modul pendeteksian cukup memenuhi harapan sampel penguji normal, walaupun dengan metode pendeteksian yang sederhana.

Pengujian juga dilakukan terhadap penderita buta warna parsial untuk membandingkan persepsi warna pengguna buta warna parsial dan pengalaman penggunaan sistem oleh penderita buta warna parsial. Hasil pengujian pada penderita buta warna parsial diperlihatkan pada Tabel 5.15 dan 5.16.

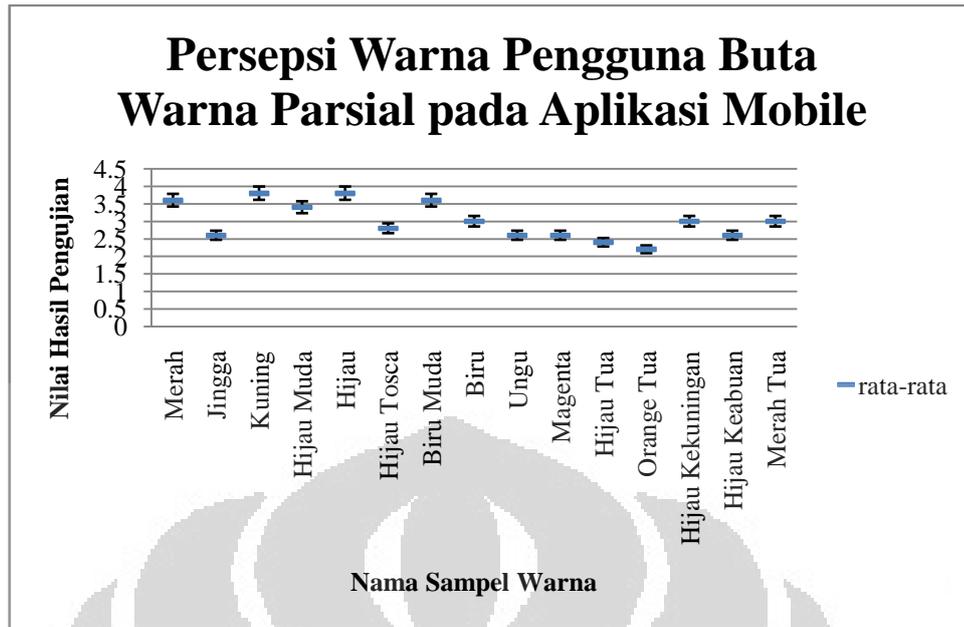
Dari Tabel 5.15 dan Tabel 5.16 hasil pengujian, dihasilkan grafik pada Gambar 5.15 dan Gambar 5.16. Gambar 5.15 merupakan grafik pengujian persepsi sampel warna terhadap pengguna normal, dimana sumbu X menyatakan nama sampel warna dan sumbu Y menyatakan nilai hasil pengujian pada pengguna. Gambar 5.16 menunjukkan grafik pengujian pengalaman pengguna, sumbu X pada grafik tersebut menunjukkan pernyataan yang diujikan, nilai hasil pengujian pengguna ditunjukkan pada sumbu Y.

Tabel 5.15. Hasil Pengujian dan Pengolahan Data Persepsi Pengguna Buta Warna Parsial Terhadap Sampel Warna dalam Aplikasi *Mobile*

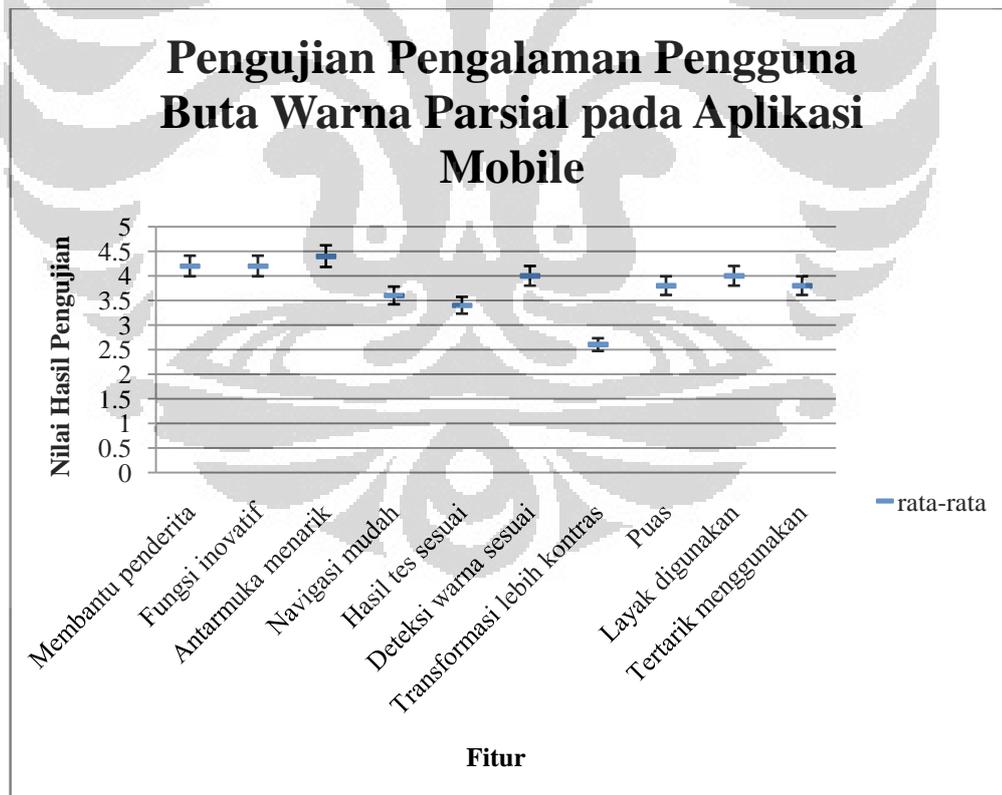
Nama Warna	Perhitungan Interval Keyakinan 95 %			
	Rata-Rata	Toleransi	Batas Bawah	Batas Atas
<b>Merah</b>	3.6	0.480	3.120	4.080
<b>Jingga</b>	2.6	0.784	1.816	3.384
<b>Kuning</b>	3.8	0.392	3.408	4.192
<b>Hijau Muda</b>	3.4	0.480	2.920	3.880
<b>Hijau</b>	3.8	0.392	3.408	4.192
<b>Hijau Tosca</b>	2.8	0.733	2.067	3.533
<b>Biru Muda</b>	3.6	0.480	3.120	4.080
<b>Biru</b>	3	0.877	2.123	3.877
<b>Ungu</b>	2.6	0.784	1.816	3.384
<b>Magenta</b>	2.6	0.480	2.120	3.080
<b>Hijau Tua</b>	2.4	0.784	1.616	3.184
<b>Orange Tua</b>	2.2	0.733	1.467	2.933
<b>Hijau Kekuningan</b>	3	0.620	2.380	3.620
<b>Hijau Keabuan</b>	2.6	0.999	1.601	3.599
<b>Merah Tua</b>	3	0.620	2.380	3.620

Tabel 5.16. Hasil Pengujian Pengalaman Penggunaan Sistem oleh Pengguna Buta Warna Parsial pada Aplikasi *Mobile*

No.	Pernyataan	Perhitungan Interval Keyakinan 95 %			
		Rata-Rata	Toleransi	Batas Bawah	Batas Atas
1	Program membantu penderita buta warna	4.2	0.733	3.467	4.933
2	Fungsi-fungsi program inovatif	4.2	1.143	3.057	5.343
3	Antarmuka (tampilan) program menarik	4.4	0.480	3.920	4.880
4	Antarmuka (tampilan) dan navigasi program mudah digunakan	3.6	0.999	2.601	4.599
5	Pada Color Blindness Test, hasil program sesuai dengan kondisi Anda	3.4	1.329	2.071	4.729
6	Pada Detect Color, hasil program sesuai dengan warna sebenarnya	4	1.074	2.926	5.074
7	Pada Transform Color, warna yang diubah menjadi lebih kontras dan terlihat	2.6	0.784	1.816	3.384
8	Anda puas menggunakan program ini	3.8	0.392	3.408	4.192
9	Aplikasi sudah layak digunakan banyak pengguna	4	0.620	3.380	4.620
10	Jika dipublikasikan, Anda tertarik menggunakan aplikasi ini	3.8	0.960	2.840	4.760



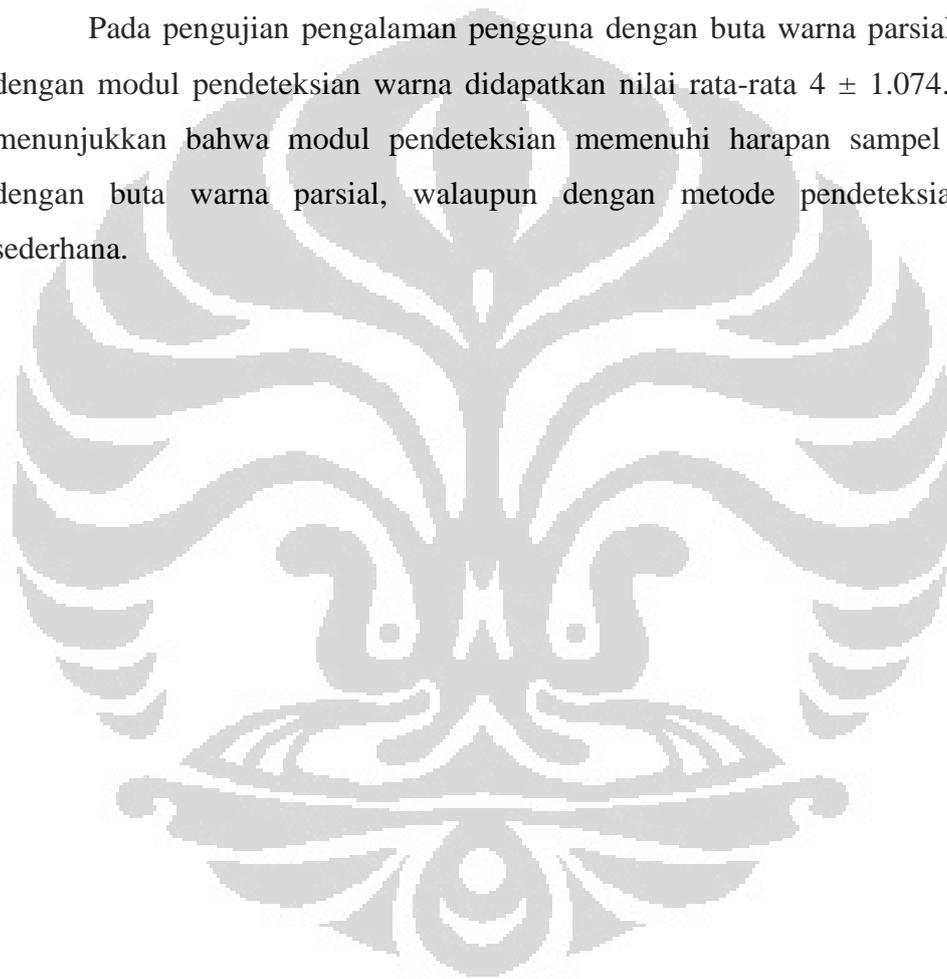
Gambar 5.15. Grafik Hasil Pengujian Persepsi Sampel Warna Terhadap Pengguna Normal pada Aplikasi *Mobile*



Gambar 5.16. Grafik Hasil Pengujian Pengalaman Pengguna Normal pada Aplikasi *Mobile*

Berdasarkan pengujian persepsi warna pengguna terhadap sampel warna, rata-rata sebesar 3. Hal ini menunjukkan bahwa rentang warna yang digunakan pada aplikasi *mobile* lebih mendekati persepsi sampel pengguna. Seperti halnya pada pengujian aplikasi pada sistem tertanam, persepsi warna pada penderita buta warna memberikan skor yang lebih tinggi dibandingkan pada pengguna normal. Hal ini dimungkinkan karena ketiadaan persepsi warna tertentu pada penderita buta warna.

Pada pengujian pengalaman pengguna dengan buta warna parsial, terkait dengan modul pendeteksian warna didapatkan nilai rata-rata  $4 \pm 1.074$ . Hal ini menunjukkan bahwa modul pendeteksian memenuhi harapan sampel pengujian dengan buta warna parsial, walaupun dengan metode pendeteksian yang sederhana.



## BAB 6 PENUTUP

Berikut adalah kesimpulan dari penulisan skripsi ini:

1. Buta warna merupakan penyakit generatif yang mengakibatkan penderitanya tidak dapat mengenali warna. Buta warna terbagi atas buta warna total dan buta warna parsial.
2. Teknologi *augmented reality* dapat digunakan sebagai salah satu alternatif dalam memberikan bantuan informasi warna.
3. EmguCV sebagai wrapper OpenCV untuk platform .NET hanya mampu berjalan pada .NET Framework, pada .NET Compact Framework seperti platform yang digunakan dalam Windows Phone 7, EmguCV tidak dapat digunakan.
4. Pada modul pendeteksian warna untuk sistem tertanam dengan model warna HSV didapatkan rata-rata pengenalan warna sebesar 69.33%, sedangkan model warna HLS menghasilkan kesuksesan sebesar 90.67%.
5. Pada modul pendeteksian warna untuk aplikasi mobile, model warna RGB memberikan pengenalan sebesar 11.33%, sedangkan modul dengan model warna HLS menunjukkan pengenalan sebesar 95.33%.
6. Pencahayaan memberikan pengaruh yang signifikan terhadap kesuksesan pendeteksian warna, baik pada modul untuk sistem tertanam maupun modul pada aplikasi *mobile*.
7. Sampel penguji memiliki persepsi yang bervariasi terhadap warna. Pada pengujian yang telah dilakukan rata-rata penguji cukup puas dengan sistem yang telah diimplementasikan.

Untuk meningkatkan pengalaman penggunaan pada modul yang telah diimplementasikan pada sistem tertanam dapat ditambahkan fitur *speech*. Pada modul untuk aplikasi *mobile*, dapat dilakukan fungsi pengolahan citra lebih lanjut untuk meminimalisasi derau serta memperbaiki tingkat pengenalan. Sistem yang ada dapat pula dirancang untuk diimplementasikan pada perangkat *head-mounted display* di masa yang akan datang.

## DAFTAR ACUAN

- [1] J. Case, Betsy. (2003). *Color Blindness*. USA : Pearson Education.
- [2] Poret, S., Jony, R. D., & Gregori, S. (2009). *Image Processing for Colour Blindness Correction*. Science and Technology for Humanity, 2009 IEEE Toronto International Conference. Toronto, Canada, 26 – 27 September 2009.
- [3] Daniel Flück: *15 Tools If You Are (Not) Colorblind*.  
<http://www.colblindor.com/2008/12/23/15-tools-color-blindness/> diakses pada 18 Desember 2010.
- [4] Manaf, Alfa Sheffildi. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Perancangan Sistem Tertanam Berbasis Konsep Realitas Tertambah Suara dengan Metode Interaksi Langsung Pengguna dengan Objek Warna*. Depok: Universitas Indonesia.
- [5] Ananto, Bayu Sri. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Perancangan Antarmuka Pengguna dan Integrasi Hasil Pengolahan Sistem Realitas Tertambah*. Depok: Universitas Indonesia.
- [6] Harwahyu, Ruki. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Perancangan Sistem Tertanam dengan Interaksi Suara*. Depok: Universitas Indonesia.
- [7] Yoh, Myeung-Sook. (2001). *The Reality of Virtual Reality*. Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM'01). Berkeley, California, USA, 25 – 27 Oktober 2001.
- [8] Bimber, Oliver, & Ramesh, Raskar. (2005). *Spatial Augmented Reality*. Massachusetts : A K Peters.
- [9] Motwani, Mukesh, et al., (2010). *Towards Benchmarking of Video Motion Tracking Algorithms*. 2010 International Conference on Signal Acquisition and Processing. Bangalore, India, 9 -10 Februari 2010.
- [10] Han, Guang-liang, Meng, Bo. (2010). *A New Real Time Multi-Target Tracking Algorithm in Video*. 2010 International Conference on Computer Application and System Modelling (ICCASM 2010). Taiyuan, Shanxi, China, 22 – 24 Oktober 2010.

- [11] Shin Shi-xu, Zheng Qi-lun, Huang Han. (2007). *A Fast Algorithm for Real-time Video Tracking*. Workshop in Intelligent Information Technology Application. Zhang Jiajie, China, 2 – 3 Desember 2007.
- [12] Fleyeh, Hasan. (2004). *Color Detection and Segmentation for Road and Traffic Signs*. Conference on Cybernetics and Intelligent Systems. Singapura, 1 – 3 Desember 2004.
- [13] Qieshi Zhang, Sei-Ishiro Kamata, Jun Zhang. (2009). *Face Detection and Tracking in Color images Using Color Centroids Segmentation*. International Conference on Robotics and Biometrics. Thailand, 22 – 25 Februari 2009.
- [14] Jiebo Luo. (2006). Color Object Detection Using Spatial-Color Joint Probability Functions. *IEEE Transactions on Image Processing, Vol. 15, No. 6, June 2006*.
- [15] Mayo, Joseph. (2008). *C# 3.0 Unleashed With the .NET Framework 3.5*. USA : Pearson Education.
- [16] Deitel, Harvey M. & Paul J. Deitel. (2008). *Visual C# 2008 How To Program*. USA : Pearson Education Inc.
- [17] Albahari, Ben, Peter Drayton, & Bradd Merrill. (2001). *C# Essentials, Second Edition*. USA : O'Reilly.
- [18] Bradski, Gary, & Adrian Kaehler. (2008). *Learning OpenCV*. USA : O'Reilly Media.
- [19] [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page) diakses pada 7 Desember 2010
- [20] Petzold, Charles. (2010). *Programming Windows Phone 7*. USA : Microsoft Corp.
- [21] Pramudya, Puja. (2010). *Silverlight for Windows Phone, Learn & Practice*. Bandung : MIC ITB.
- [22] Fowler, Martin. (2003). *UML Distilled: A Brief Guide to the Standard Object Modelling Language, Third Edition*. USA: Addison Wesley.

- [23] <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce> diakses pada 10 Juni 2011.
- [24] Li Shuhua, Guo Gaizhi. (2010). *The Application of Improved HSV Color Space Model in Image Processing*. 2nd International Conference on Future Computer and Communication. Cina, 21 – 24 Mei 2010.

