



UNIVERSITAS INDONESIA

**IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA:
DESAIN ANTARMUKA PENGGUNA, SISTEM TES BUTA
WARNA DENGAN ISHIHARA, DAN TRANSFORMASI
WARNA PADA SISTEM REALITAS TERTAMBAH**

SKRIPSI

**BAYU SRI ANANTO
0706275946**

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



UNIVERSITAS INDONESIA

**IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA:
DESAIN ANTARMUKA PENGGUNA, SISTEM TES BUTA
WARNA DENGAN ISHIHARA, DAN TRANSFORMASI
WARNA PADA SISTEM REALITAS TERTAMBAH**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Teknik

**BAYU SRI ANANTO
0706275946**

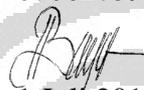
**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JULI 2011**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Bayu Sri Ananto

NPM : 0706275946

Tanda Tangan : 

Tanggal : 1 Juli 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Bayu Sri Ananto
NPM : 0706275946
Program Studi : Teknik Komputer
Judul Skripsi : Implementasi Sistem Bantuan Penderita Buta Warna:
Desain Antarmuka Pengguna, Sistem Tes Buta
Warna dengan Ishihara, dan Transformasi Warna
pada Sistem Realitas Tertambah

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. ()

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

Penguji : Prima Dewi Purnamasari, ST., MT., M.Sc. ()

Ditetapkan di : Depok

Tanggal : 1 Juli 2011

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kehadirat Allah SWT, karena atas segala rahmat dan penyertaan-Nya saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dari berbagai pihak. Oleh sebab itu, saya mengucapkan terima kasih kepada:

1. Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc., selaku dosen pembimbing yang membantu memberikan arahan dan nasihat, baik selama penulisan skripsi maupun selama masa studi di Teknik Komputer.
2. Kedua orang tua saya yang tidak pernah lelah dalam memberikan dukungan serta doa untuk kebaikan saya.
3. Teman-teman di Departemen Teknik Elektro Universitas Indonesia, khususnya program studi Teknik Komputer angkatan 2007, yang telah memberikan pengalaman dan kenangan terbaik selama masa studi.

Semoga Allah SWT berkenan membalas kebaikan semua pihak yang telah membantu dan menjadikan skripsi ini dapat bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, 1 Juli 2011

Bayu Sri Ananto

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

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

Nama : Bayu Sri Ananto
NPM : 0706275946
Program studi : Teknik Komputer
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

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:
DESAIN ANTARMUKA PENGGUNA, SISTEM TES BUTA WARNA
DENGAN ISHIHARA, DAN TRANSFORMASI WARNA
PADA SISTEM REALITAS TERTAMBAH**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif 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 sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 1 Juli 2011
Yang menyatakan



Bayu Sri Ananto

ABSTRAK

Nama : Bayu Sri Ananto
Program Studi : Teknik Komputer
Judul : Implementasi Sistem Bantuan Penderita Buta Warna: Desain Antarmuka Pengguna, Sistem Tes Buta Warna dengan Ishihara, dan Transformasi Warna pada Sistem Realitas Tertambah

Penderita buta warna umumnya sering mengalami kesulitan dalam membedakan warna tertentu, bahkan untuk penderita buta warna total, mereka hanya dapat melihat dalam warna hitam, putih, dan abu-abu saja. Kelainan persepsi warna tersebut menimbulkan permasalahan yang banyak dialami oleh penderita buta warna, mulai dari aktifitas sehari-hari sampai masalah pendidikan. Salah satu solusi untuk membantu permasalahan tersebut adalah dengan membangun sistem bantuan menggunakan teknik pengolahan citra dan menerapkan teknologi *augmented reality*. Dalam implementasi sistem bantuan tersebut, penelitian ini difokuskan pada perancangan antarmuka pengguna sistem bantuan penderita buta warna dan pengembangan sistem tes buta warna pada perangkat *Windows Phone 7*, serta pembuatan sistem transformasi warna pada perangkat bergerak, maupun sistem tertanam. Perancangan antarmuka pengguna sistem bantuan penderita buta warna untuk *platform Windows Phone 7* diimplementasikan menggunakan *Microsoft Expression Blend* berdasarkan prinsip *barrier-free*. Pengujian buta warna bagi pengguna digunakan metode Ishihara, dan untuk transformasi warna diterapkan teori Dalton dengan menggunakan pemrograman bahasa C# dengan tambahan *library* EmguCV. Analisis pengujian sistem ini membuktikan bahwa penerapan metode Ishihara untuk sistem tes buta warna memberikan hasil akurasi yang tinggi dengan persentase 100% dari 10 kali pengujian. Metode Daltonisasi untuk sistem transformasi warna memberikan hasil optimal untuk memperjelas objek pada citra berwarna ambigu. Pada perangkat tertanam, persentase kondisi citra yang dapat terlihat sangat jelas adalah 74%, dan 37% untuk perangkat bergerak.

Kata kunci: *antarmuka, augmented reality, buta warna, tes buta warna, transformasi warna.*

ABSTRACT

Name : Bayu Sri Ananto
Study Program : Computer Engineering
Title : Implementation of Color Blind Aid System: User Interface Design, Color Blindness Test Using Ishihara Method, and Color Transformation on Augmented Reality System

People with color blindness often have difficulty in distinguishing certain colors. Even for people with total color blindness, they can only see in black, white, and gray colors. Deficiency of color perception is causing many problems experienced by people with color blindness, from daily activities to educational issue. One solution to help this problem is to build an aid system using image processing techniques and applying augmented reality technologies. In the system implementation, research has focused on designing the user interface of color blind aid system and color blindness test system for Windows Phone7 device, and also developing color transformation system for mobile and embedded device. The design of color blind aid system user interfaces for Windows Phone 7 platform is implemented using Microsoft Expression Blend based on barrier-free principle. Color blindness test system using Ishihara method, and the application of Dalton's theory to transform color using C# programming language with additional EmguCV library. Analysis of the test in the implementation proves that the implementation of Ishihara method for color blindness test system provides high accuracy results with the percentage of 100% of 10 times testing. Dalton method for color transformation system provides optimal results in clarifying ambiguous colored objects, especially for color blind people. In embedded device the visibility percentage is 74% meanwhile in mobile device the visibility percentage is only 37%.

Key words: augmented reality, color blind, color blindness test, interface, transform color.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMA KASIH	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Penelitian.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Metode Penelitian	5
1.6 Sistematika Penulisan	5
BAB 2 TINJAUAN LITERATUR	7
2.1 Permodelan Ruang Warna	7
2.1.1 RGB	7
2.1.2 HSV	8
2.1.3 HSL.....	9
2.2 Karakteristik Penglihatan Buta Warna.....	9
2.3 Metode Daltonisasi	13
2.3.1 Daltonisasi Vischeck.....	13
2.3.2 Daltonisasi Fidaner	13
2.4 Konsep <i>Augmented Reality</i>	14
2.5 Pemanfaatan Teknologi untuk Implementasi.....	16
2.5.1 <i>Microsoft Expression Blend</i>	16
2.5.2 <i>Windows Phone 7 Developer Tool</i>	18
2.5.3 <i>Microsoft Visual Studio</i>	20
BAB 3 PERANCANGAN ANTARMUKA PENGGUNA, PENGETESAN BUTA WARNA, DAN TRANSFORMASI WARNA	21
3.1 Perancangan Antarmuka Pengguna Sistem Bantuan Buta Warna	21
3.1.1 Prinsip Dasar Desain Antarmuka Pengguna.....	21
3.1.2 Desain Interaksi Manusia – Komputer Prinsip <i>Barrier-free</i>	24
3.1.3 Rancangan Antarmuka Pengguna.....	25

3.2 Sistem Tes Buta Warna dengan Metode Ishihara	27
3.3 Sistem Transformasi Warna	30
3.4 Integrasi untuk Sistem Bantuan Buta Warna	32
3.5 Rancangan Cara Kerja Sistem	33
3.5.1 Diagram <i>Use Case</i>	34
3.5.2 Diagram <i>Sequence</i>	35
3.5.3 Diagram <i>Class</i>	36
BAB 4 IMPLEMENTASI RANCANGAN PENELITIAN	37
4.1 Implementasi Antarmuka Pengguna pada <i>Windows Phone 7</i>	37
4.2 Implementasi Sistem Tes Buta Warna dengan Metode Ishihara.....	40
4.3 Implementasi Transformasi Warna	41
4.3.1 Pengembangan untuk Perangkat Tertanam.....	42
4.3.2 Pengembangan untuk Perangkat Bergerak.....	47
BAB 5 ANALISIS PENGUJIAN IMPLEMENTASI SISTEM	50
5.1 Analisis Pengujian pada Perangkat Bergerak (<i>Windows Phone 7</i>).....	50
5.1.1 Penilaian Pengguna terhadap Antarmuka	50
5.1.2 Pengaruh Kamera terhadap Kesesuaian Perubahan Warna	52
5.1.3 Tingkat Kejelasan Citra dengan Transformasi Warna	54
5.1.4 Tingkat Akurasi Tes Buta Warna	57
5.1.5 Pengujian Keseluruhan Sistem Bantuan Buta Warna pada Perangkat Bergerak	59
5.2 Analisis Pengujian pada Perangkat Tertanam (<i>eBox 3310</i>).....	60
5.2.1 Tingkat Kejelasan Citra dengan Pentransformasian Warna.....	61
5.2.2 Pengujian Keseluruhan Sistem Bantuan Buta Warna pada Perangkat Tertanam.....	63
BAB 6 KESIMPULAN	66
DAFTAR REFERENSI	67

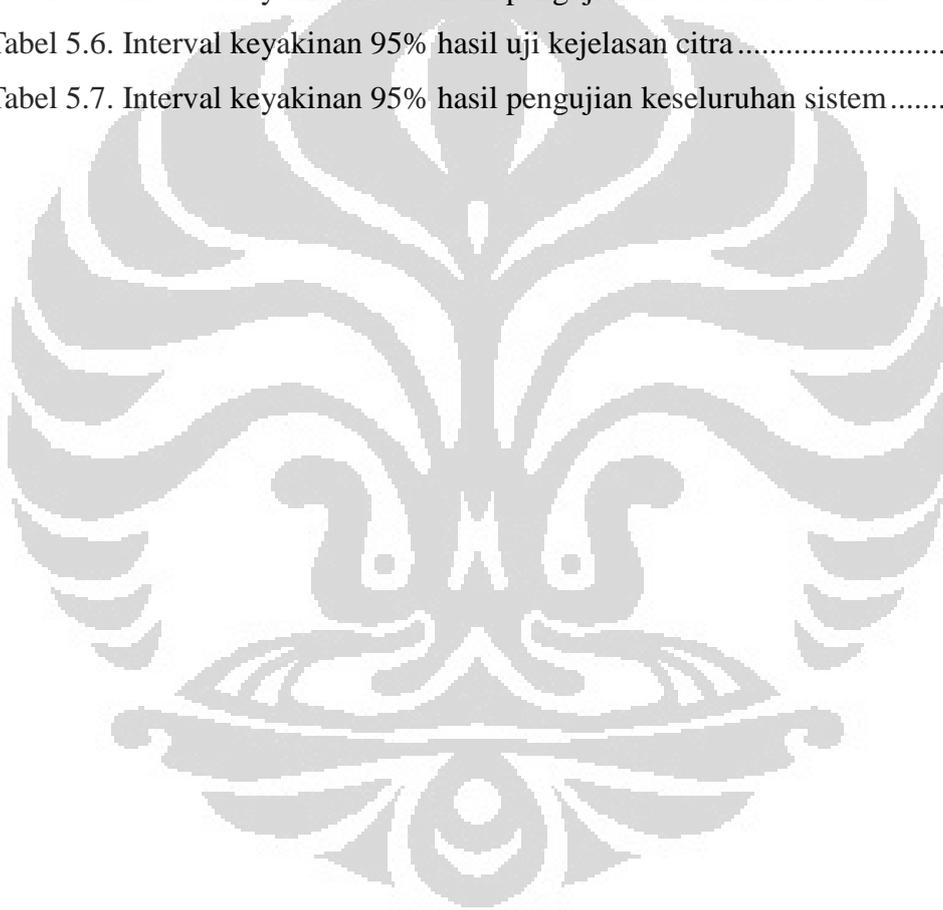
DAFTAR GAMBAR

Gambar 1.1. Desain arsitektur sistem perangkat untuk penyandang buta warna...	4
Gambar 2.1. Diagram permodelan warna dalam ruang HSV	8
Gambar 2.2. Spektrum sel kerucut pada mata normal dan buta warna.....	12
Gambar 2.3. <i>Building Block</i> pada <i>Augmented Reality</i>	15
Gambar 2.4. Contoh pengaplikasian sistem AR.....	15
Gambar 2.5. Tampilan memilih jenis proyek pada <i>Microsoft Expression Blend</i> .	17
Gambar 2.6. Proses menampilkan antarmuka dengan <i>Silverlight</i> pada <i>Blend</i>	18
Gambar 2.7. Antarmuka pengguna <i>Windows Phone 7</i> pada <i>Expression Blend</i> ...	19
Gambar 2.8. Tampilan <i>Visual Studio 2010 Express</i> untuk <i>Windows Phone</i>	20
Gambar 3.1. Hubungan interaksi manusia dan komputer	21
Gambar 3.2. Sistematisa penelitian	23
Gambar 3.3. Sketsa rancang antarmuka sistem bantu pada perangkat bergerak..	26
Gambar 3.4. Bagian-bagian pada <i>Expression Blend</i> untuk desain antarmuka.....	27
Gambar 3.5. Algoritma sistem tes buta warna	29
Gambar 3.6. Hasil proses daltonisasi menggunakan aplikasi <i>Vischeck</i>	31
Gambar 3.7. Perbandingan dua warna dilihat mata normal dan <i>deuteranomali</i> ..	31
Gambar 3.8. Diagram alir penggunaan sistem bantu penyandang buta warna	34
Gambar 3.9. <i>Use Case</i> diagram sistem bantuan bagi penyandang buta warna....	34
Gambar 3.10. <i>Sequence</i> diagram penggunaan fitur pada perangkat bergerak	35
Gambar 3.11. <i>Class</i> diagram integrasi antarmuka pengguna dengan sistem.....	36
Gambar 4.1. Pendefinisian fungsi <i>timer</i>	38
Gambar 4.2. Kode program bahasa XAML untuk penambahan animasi	39
Gambar 4.3. Kode program menampilkan info bantuan dan navigasi halaman ..	39
Gambar 4.4. Kode program untuk menentukan jenis buta warna.....	41
Gambar 4.5. Hasil implementasi sistem tes buta warna pada <i>Windows Phone 7</i> 41	
Gambar 4.6. Algoritma transformasi warna perangkat tertanam dan perangkat bergerak	42
Gambar 4.7. Kode program untuk kelas <code>ColorIntensityIncrease</code>	45
Gambar 4.8. Transformasi warna dengan pergeseran nilai <i>hue</i> dan <i>saturation</i> ...	46
Gambar 4.9. Transformasi warna pada perangkat tertanam.....	46

Gambar 4.10. Kode program untuk <i>method</i> <code>GetColor2()</code>	49
Gambar 4.11. Hasil implementasi transformasi warna pada <i>Windows Phone 7</i> ...	49
Gambar 5.1. Grafik penilaian pengguna terhadap antarmuka.....	51
Gambar 5.2. Diagram persentase pengujian sampel warna dengan kamera	53
Gambar 5.3. Grafik perbandingan kondisi sampel warna asli dengan hasil tangkapan kamera	53
Gambar 5.4. Jenis citra Ishihara yang digunakan dalam pengujian	55
Gambar 5.5. Grafik perbandingan tingkat kejelasan citra.....	56
Gambar 5.6. Grafik persentase kejelasan penglihatan terhadap citra	56
Gambar 5.7. Grafik pengujian sistem tes buta warna terhadap kondisi penglihatan pengguna.....	58
Gambar 5.8. Grafik pengujian keseluruhan sistem pada perangkat bergerak.....	60
Gambar 5.9. Grafik perbandingan tingkat kejelasan citra.....	62
Gambar 5.10. Grafik persentase kejelasan penglihatan terhadap citra	63
Gambar 5.11. Grafik hasil uji keseluruhan sistem pada perangkat tertanam.....	64

DAFTAR TABEL

Tabel 3.1. Desain dan fungsi 21 citra Ishihara	28
Tabel 3.2. Tipikal warna ambigu untuk tipe penglihatan lemah terhadap warna..	30
Tabel 5.1. Tabel data hasil pengujian antarmuka pengguna.....	50
Tabel 5.2. Tabel data hasil pengujian pengaruh kamera terhadap warna citra	52
Tabel 5.3. Interval keyakinan 95% hasil uji kejelasan citra	55
Tabel 5.4. Kondisi penglihatan penguji menggunakan sistem tes buta warna.....	58
Tabel 5.5. Interval keyakinan 95% hasil pengujian keseluruhan sistem	59
Tabel 5.6. Interval keyakinan 95% hasil uji kejelasan citra	61
Tabel 5.7. Interval keyakinan 95% hasil pengujian keseluruhan sistem	64



BAB 1

PENDAHULUAN

1.1 Latar Belakang Penelitian

Saat ini, masih banyak orang yang menganggap bahwa buta warna adalah suatu penyakit dan kemampuan penglihatan terhadap warna hanya bisa direntangan putih dan hitam saja. Namun, sebenarnya buta warna adalah kelainan bawaan atau keturunan yang dibawa oleh kromosom X orang tua penyandang buta warna, bukan suatu penyakit. Oleh sebab itu, tidak ada obat yang dapat menyembuhkan buta warna.

Secara garis besar, buta warna terdapat dua jenis, yaitu buta warna total dan buta warna parsial. Anggapan penglihatan buta warna yang hanya bisa melihat dengan warna hitam, putih, dan abu-abu saja adalah gambaran untuk penderita buta warna total, sedangkan buta warna parsial tidak dapat membedakan warna-warna tertentu tergantung dari jenis buta warna parsialnya. Hal ini terjadi karena retina mata pada penyandang buta warna parsial memiliki kedua sel yang diperlukan untuk melihat benda, yaitu sel batang yang sensitif terhadap cahaya dan sel kerucut (konus) yang sensitif terhadap warna. Pada penderita buta warna parsial, sel konus tidak berfungsi sempurna. Untuk penyandang buta warna total, retina penyandang buta warna tidak memiliki sel konus sehingga mata penyandang buta warna total tidak sensitif terhadap warna.

Berdasarkan beberapa sumber diperkirakan terdapat sekitar 9% - 12% pria dari total populasi mengalami defisiensi penglihatan terhadap warna atau buta warna [1]. Sebuah penelitian lain [2] menyebutkan bahwa ada sekitar 8% pria dan kurang dari 1% wanita yang memiliki kesalahan persepsi warna sejak lahir. Bahkan, menurut data statistik Pusat Kesehatan Nasional di Amerika yang dihimpun oleh *Center for Disease Control* (CDC), terdapat sekitar 3,4 juta penderita buta warna di USA [3]. Adapun data-data tersebut adalah data untuk semua kategori atau jenis buta warna, baik buta warna total maupun parsial.

Dari banyaknya jumlah penderita buta warna yang telah disebutkan, hanya sedikit persentase jumlah penyandang buta warna total. Namun, terlepas dari hal

tersebut, semua penyandang buta warna jenis apapun tentu memiliki masalah dalam kehidupan sehari-harinya. Banyak yang tidak mengetahui bahwa orang yang menderita buta warna parsial juga memiliki permasalahan yang terkait dengan kelainan penglihatannya terhadap warna. Berikut merupakan beberapa masalah yang dihadapi oleh beberapa penyandang buta warna:

1. Masalah sehari-hari.

Dampak kelainan buta warna tentu dihadapi oleh penyandang buta warna dalam kehidupannya sehari-hari. Misalnya saja dalam membedakan warna pakaian, warna lampu lalu lintas saat berkendara, warna simbol-simbol tertentu, dan lain sebagainya.

2. Masalah bidang pendidikan.

Buta warna tentu mempengaruhi penderitanya dalam memilih program studi untuk melanjutkan pendidikannya, bahkan dalam memilih karir selanjutnya. Misalnya, beberapa pilihan program studi dan pekerjaan mensyaratkan mahasiswa atau karyawannya tidak buta warna.

3. Masalah psikologis.

Diskriminasi terhadap penyandang buta warna masih sering terjadi, khususnya di wilayah-wilayah tertentu. Beberapa orang menganggap buta warna adalah penyakit. Selain itu, ketidakmampuan dalam membedakan warna sering kali menjadi bahan ejekan, bahkan hinaan yang dapat membuat penyandang buta warna merasa dikucilkan dan tidak percaya diri.

Permasalahan-permasalahan tersebut merupakan latar belakang tujuan skripsi ini, yaitu untuk membangun sebuah sistem aplikasi bantuan bagi penyandang buta warna. Akan dibentuk aplikasi yang dapat dijalankan pada perangkat bergerak (*Windows Phone 7*), maupun bentuk sistem tertanam (*embedded system*).

1.2 Rumusan Masalah

Perkembangan teknologi memunculkan berbagai inovasi yang mengarah kepada tujuan untuk mempermudah aktifitas manusia serta membantu orang yang memiliki kekurangan. Contohnya, teknologi *text-to-speech* merupakan salah satu

teknologi untuk membantu penyandang tunanetra. Di lain pihak, teknologi *speech-to-text* menjadi salah satu teknologi yang dapat membantu penyandang tunarungu. Masih banyak lagi program atau perangkat-perangkat yang dapat membantu para penyandang cacat dan kekurangan lainnya. Saat ini masih jarang sekali teknologi yang secara efektif dan efisien dapat digunakan untuk membantu penyandang buta warna. Penelitian ini ditujukan untuk mempelajari dan menjawab pertanyaan berikut:

- Bagaimana membuat sistem aplikasi yang berguna untuk membantu penyandang buta warna?
- Bagaimana cara membuat antarmuka pengguna yang *user-friendly*, sistem pengujian jenis penglihatan terhadap warna dengan metode Ishihara, serta mentransformasikan warna pada sistem realitas tertambah (*augmented reality*)?

1.3 Tujuan Penelitian

Tujuan penelitian ini secara umum adalah membantu penyandang buta warna dan meringankan permasalahan yang ada dengan membangun sebuah sistem aplikasi yang dapat dijalankan pada perangkat bergerak dan perangkat sistem tertanam. Hal ini terkait masalah di bidang pendidikan yang juga merupakan salah satu hal yang diproyeksikan oleh Persatuan Bangsa-Bangsa (PBB) dalam *Millenium Development Goals* (MDGs) poin kedua, yaitu persamaan hak dalam pendidikan (*universal education*).

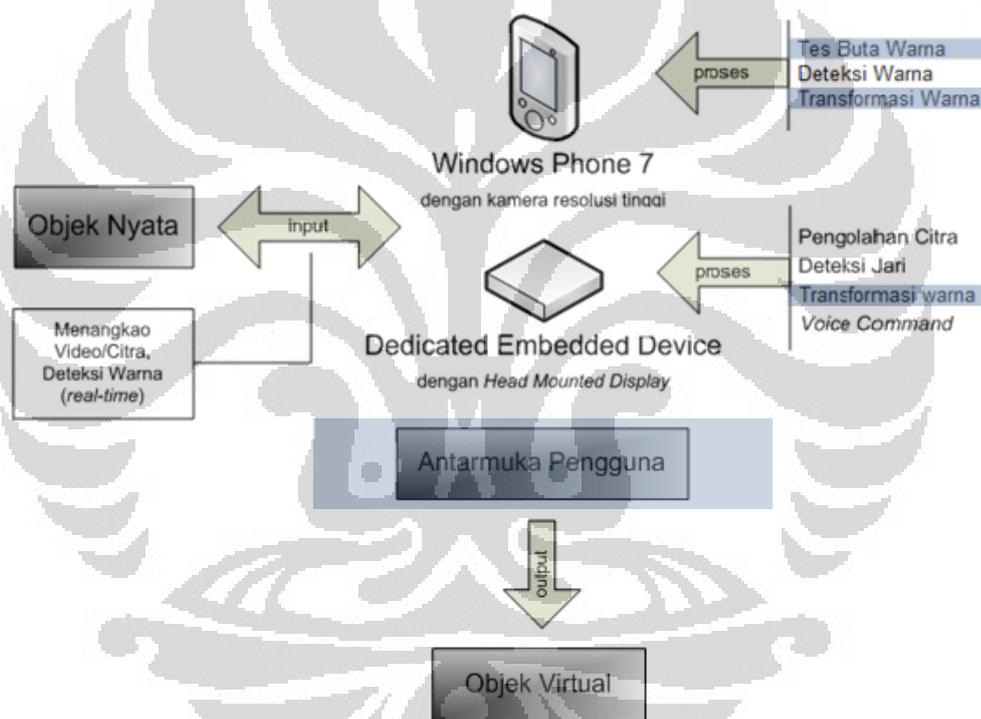
Mengacu pada tujuan utama di atas, ada pula tujuan khusus yang ingin dicapai dalam penelitian tersebut, yaitu untuk mendesain antarmuka pengguna yang efektif dan menarik, membuat sistem tes buta warna yang akurat, serta mengimplementasikan formula transformasi warna yang tepat.

1.4 Batasan Masalah

Permasalahan yang akan dibahas dalam penelitian ini dibatasi pada implementasi desain antarmuka pengguna, sistem tes buta warna, dan transformasi warna pada sistem bantuan bagi penyandang buta warna dalam

bentuk fitur untuk membedakan warna yang terlihat sama atau mirip pada penglihatan penyandang buta warna (terlihat beda pada mata normal). Perangkat lunak yang digunakan dalam implementasi sebagian besar merupakan aplikasi dari Microsoft, yaitu *Microsoft Expression Blend*, *Windows Phone 7 Developer Tool*, dan *Microsoft Visual Studio 2010*.

Secara keseluruhan, dalam penelitian ini sistem yang dikembangkan adalah dengan *platform* Microsoft .NET dengan kompatibilitas terhadap sistem operasi *Windows Phone 7* dan *Windows Embedded CE 7*. Antarmuka pengguna yang dirancang akan mendukung sistem aplikasinya.



Gambar 1.1. Desain arsitektur sistem perangkat untuk penyandang buta warna

Pada Gambar 1.1, bagian proses dan bagian yang menghubungkan langsung antara pengguna dan sistem (ditandai kotak biru) merupakan lingkup bahasan dalam penelitian ini. Lingkup penelitian mencakup hasil keluaran sistem berupa fitur tes buta warna dan transformasi warna, serta antarmuka pengguna yang diimplementasikan pada sistem realitas tertambah pada perangkat sistem tertanam dan perangkat bergerak.

1.5 Metode Penelitian

Metode yang diterapkan dalam penelitian ini adalah metode eksperimen. Penelitian dilakukan dengan cara melakukan suatu tindakan atau perlakuan terhadap subjek penelitian guna membangkitkan suatu keadaan yang akan diteliti [4]. Untuk dasar penelitian, dilakukan studi literatur yang menjadi referensi dalam menganalisis hal yang terkait dengan penyakit buta warna dan metode sistem pengolahan citra, seperti manipulasi warna.

Pelaksanaan metode penelitian eksperimen ini tidak terbatas sampai pada bentuk rancangan saja, tetapi dilakukan tindak lanjut terhadap hasil rancangan, sehingga data yang diperoleh dalam tahap perancangan dapat dianalisa untuk beralih ke tahap implementasi sehingga pada akhirnya mengarah pada pengembangan produk hasil penelitian yang lebih baik.

Sebelumnya, pengujian sistem secara keseluruhan dilakukan dengan menentukan jumlah populasi yang diteliti sebagai objek penelitian. Populasi dalam penelitian adalah mahasiswa Universitas Indonesia dan sampel yang dipilih dalam pengujian sistem digunakan teknik sampel acak. Sampel acak adalah sampel yang diambil dari suatu populasi dan setiap anggota populasi mempunyai kesempatan yang sama untuk dipilih sebagai sampel [5].

Adapun untuk mendapatkan hasil penelitian yang representatif dan akurat, sampel yang akan dipilih untuk mengetes hasil penelitian tidak hanya bersumber pada sampel kategori penderita buta warna. Akan diadakan survey atau pengisian kuesioner terhadap beberapa orang yang tidak menyandang buta warna untuk mengetahui respon dan tanggapan orang yang normal terhadap tampilan antarmuka pengguna dan tentang sistem yang dibuat.

1.6 Sistematika Penulisan

Skripsi ini terdiri dari lima bab, yang terdiri dari:

Bab 1 Pendahuluan

Bagian ini terdiri dari latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

Bab 2 Tinjauan Literatur

Bagian ini berisi tentang permodelan warna, karakteristik penglihatan buta warna, metode daltonisasi, konsep *augmented reality*, pemanfaatan teknologi untuk implementasi.

Bab 3 Perancangan Antarmuka Pengguna, Sistem Tes Buta Warna, dan Transformasi Warna

Bagian ini menjelaskan perancangan antarmuka pengguna sistem bantuan untuk buta warna, sistem tes buta warna dengan metode Ishihara, sistem transformasi warna, integrasi sistem bantuan buta warna, dan rancangan cara kerja sistem yang digambarkan menggunakan diagram *Unified Modelling Language* (UML).

Bab 4 Implementasi Rancangan dan Analisis Pengujian

Bagian ini menjelaskan implementasi antarmuka pengguna pada perangkat *Windows Phone 7*, implementasi sistem tes buta warna dengan metode Ishihara, dan implementasi transformasi warna.

Bab 5 Kesimpulan

Bagian ini berisi penutup dari penelitian.

BAB 2 TINJAUAN LITERATUR

2.1 Permodelan Ruang Warna

Warna dalam analisis citra adalah faktor penting. Pengolahan warna direpresentasikan ke dalam suatu jenis ruang warna dan tiap permodelan ruang warna tersebut memiliki sistem koordinat warna yang spesifik, bahkan setiap titik pada domain permodelan warna tersebut hanya merepresentasikan satu warna yang spesifik [6]. Pemetaan suatu warna ke dalam ruang warna tertentu dilakukan untuk tujuan spesifik dan hingga saat ini sudah terdapat lebih dari lima permodelan warna, di antaranya adalah RGB, HSV, HSL, CMYK, LUV. Tiap model warna memiliki perhitungan matematis yang berbeda-beda yang mempengaruhi tingkat kerumitan dalam pengolahan warna pada citra. Adapun model ruang warna yang sering digunakan adalah RGB, HSV, dan HSL.

2.1.1 RGB

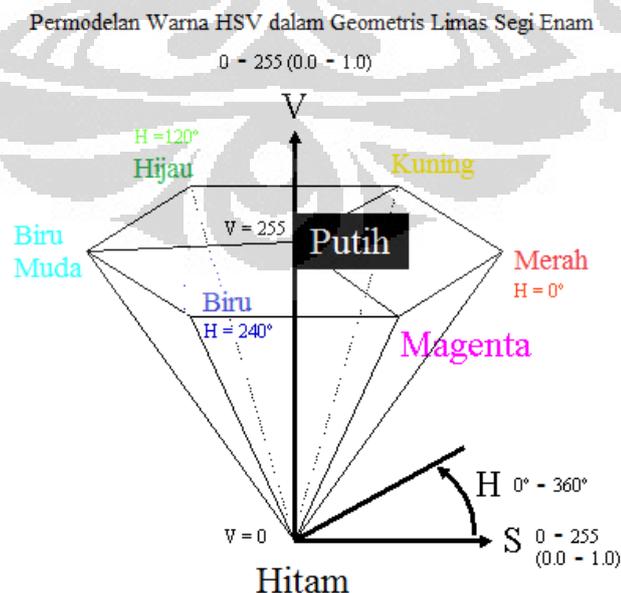
Ruang warna RGB didefinisikan oleh tiga kanal atau koordinat warna primer merah, hijau, dan biru. Tiga warna tersebut dipilih berdasarkan penelitian yang menunjukkan bahwa warna-warna yang dapat ditangkap oleh mata manusia adalah hasil perpaduan dari cahaya dengan panjang gelombang yang berbeda-beda, dan kombinasi warna yang memberikan rentang warna paling luas adalah perpaduan warna merah, hijau, dan biru [7]. Dalam permodelannya, spesifikasi lengkap dari ruang warna RGB terdapat titik kromatis putih yang ikut menentukan percampuran warna antara rentang warna merah, hijau, dan biru untuk membentuk warna lain.

RGB adalah model warna yang nyaman untuk komputer grafis, karena sistem visual manusia bekerja dengan cara yang mirip. Namun, untuk pengolahan citra, proses dengan menggunakan format RGB jarang digunakan karena memiliki perhitungan matematis yang lebih rumit dibanding ruang warna lain seperti HSL dan HSV. Hal tersebut disebabkan koordinat warna yang berasal dari tiga kanal warna primer, tidak seperti HSL atau HSV, semua koordinat yang menentukan warna inti terdapat pada kanal *hue*.

2.1.2 HSV

HSV adalah suatu permodelan warna yang dipublikasikan pada tahun 1978 oleh Alvy Ray Smith. HSV merupakan representasi lain dari ruang warna RGB (transformasi nonlinier dari model RGB) dengan persepsi gambaran hubungan warna yang lebih akurat daripada RGB. Perhitungan dan pengolahan terkait persepsi warna dalam ruang HSV lebih sederhana sehingga banyak dilakukan dalam bidang pengolahan citra [7]. Keterbatasan secara matematis menyebabkan kemungkinan terjadinya nilai akurasi yang rendah dalam klasifikasi warna sehingga dapat berpengaruh terhadap proses algoritma.

Pada ruang warna HSV, warna didefinisikan oleh *hue*, *saturation*, dan *value*. *Hue* merupakan parameter dasar untuk persepsi terhadap warna, posisi dalam ruang warna standarnya memiliki rentang antara 0° sampai 360° . *Hue* terbagi menjadi *hue* kuat dan *hue* lemah. Hal tersebut dideskripsikan dengan level saturasi. *Saturation* dalam HSV adalah komponen untuk mendeskripsikan kekuatan warna atau kepedaran suatu warna terhadap warna yang ditangkap mata, dan memiliki rentang nilai antara 0% sampai 100%. Adapun, *value* digunakan untuk menyatakan kecerahan suatu warna. Atribut ini adalah representasi jumlah banyak sedikitnya cahaya yang dipancarkan dari suatu objek. Sama seperti *saturation*, *value* memiliki rentang nilai antara 0% sampai 100%.



Gambar 2.1. Diagram permodelan warna dalam ruang HSV [8]

HSV dikembangkan untuk menampilkan warna pada sistem dengan rentang dinamis yang terbatas (tingkat piksel 0-255 untuk warna 24-bit). Dengan permodelan tersebut, warna diformulasikan menurut apa yang diterima oleh mata manusia. Penglihatan manusia dipengaruhi oleh tiga kualitas faktor warna, yaitu *hue*, *saturation*, dan *value* (atau *brightness*). Maka dengan mengkonversi atau mentransformasi warna dari RGB ke HSV, menjadi suatu langkah untuk menghindari kemungkinan persepsi penglihatan terhadap rentang warna yang tidak sempurna [9].

2.1.3 HSL

Konsep permodelan ruang warna HSL sebenarnya mirip dengan permodelan pada ruang warna HSV. Jika pada HSV, warna ditentukan juga oleh faktor *value*, pada HSL tingkat intensitas warna ditentukan oleh faktor *lightness*. Secara konseptual, HSL mewakili sebuah kerucut ganda dengan putih di bagian atas, hitam di bagian bawah, dan warna abu-abu pada pusatnya. HSL lebih merepresentasikan konsep intuitif jelas dan jenuh sebagai dua parameter independen. Oleh sebab itu, dalam pengolahan citra dengan menggunakan format model ruang warna HSL, parameter yang sering digunakan adalah *hue* dan *lightness*.

2.2 Karakteristik Penglihatan Buta Warna

Sebagai salah satu faktor yang berkaitan dengan penglihatan, buta warna kini menjadi salah satu tolak ukur untuk sebuah persyaratan masuk kuliah atau dunia kerja. Pembedaan persepsi terhadap suatu warna sangat dibutuhkan dalam bidang-bidang tertentu, seperti teknik dan kedokteran. Warna berperan penting dalam kehidupan sehari-hari, warna sering dijadikan sebagai tanda atau sekedar estetika.

Kondisi buta warna mulai diketahui sejak tahun 1879 yang dipublikasikan John Dalton dalam tulisan esainya [10]. Hal tersebut menjadi awal dikenalnya buta warna dalam literatur, sekaligus mendorong untuk memberikan bukti konklusif atas persepsi warna pada penyandang buta warna [11].

Perbedaan mendasar antara penderita buta warna dan kebanyakan orang adalah bahwa beberapa warna yang muncul berbeda jika dilihat mata normal, namun terlihat sama pada penyandang buta warna. Penderita buta warna memiliki kelemahan terhadap penglihatan, dengan kurangnya kemampuan untuk membedakan warna, saturasi, dan kecerahan.

Pada dasarnya, penglihatan terhadap warna ditentukan oleh perbedaan tiga kualitas faktor warna, yaitu *hue* (jenis warna, seperti merah atau hijau, dan sebagainya), *saturation* (warna tunggal atau warna campuran), *brightness* (intensitas warna terang atau gelap) [12]. Ketiga faktor untuk penglihatan warna tersebut dipengaruhi oleh fungsi sel konus atau sel kerucut. Sel fotosensitif pada mata yang disebut sel kerucut tersebut memungkinkan kita melihat warna. Sel kerucut terletak di tengah-tengah retina dan mengandung tiga jenis pigmen fotosensitif yang dapat mendeteksi warna merah, hijau, dan biru. Seorang buta warna memiliki kelemahan atau bahkan sama sekali tidak memiliki pigmen tersebut [13].

Sel kerucut menyerap foton dan mengirim sinyal listrik ke otak. Tiga jenis sel kerucut memiliki sensitifitas spektral yang berbeda. *L-cones* untuk menyerap (absorpsi) panjang gelombang yang tinggi, *M-cones* untuk menyerap panjang gelombang menengah, dan *S-cones* untuk menyerap panjang gelombang yang rendah secara efektif. Berdasarkan fungsi sel tersebut, akibatnya cahaya dipersepsikan terdiri dari tiga jenis (l, m, s), l (*long*), m (*middle*), dan s (*short*), merepresentasikan jumlah foton yang diserap oleh masing-masing *L-*, *M-*, dan *S-cones* [10]. Oleh karena itu, penglihatan normal disebut juga trikromasi normal (tiga warna).

Kelainan pada sel kerucut tersebut, menyebabkan kemungkinan terjadinya buta warna. Terdapat tiga jenis gangguan penglihatan terhadap warna.

1. Monokromasi

Monokromasi adalah keadaan ketika seseorang hanya memiliki sebuah sel kerucut (*cones*) atau tidak berfungsinya semua sel kerucut. Buta warna jenis ini biasa disebut buta warna total. Buta warna total sangat jarang terjadi dan dialami oleh sekitar 1 dari 10.000 penduduk di dunia [14]. Monokromasi ada

dua jenis, yaitu *rod monochromacy* dan *cone monochromacy*.

- a. *Rod monochromacy* adalah jenis buta warna yang sangat jarang terjadi, yaitu ketidakmampuan membedakan warna sebagai akibat dari tidak berfungsinya semua *cones* retina. Penderita *rod monochromacy* tidak dapat membedakan warna sehingga yang terlihat hanya hitam, putih, dan abu-abu saja.
- b. *Cone monochromacy* adalah tipe monokromasi yang disebabkan oleh tidak berfungsinya dua sel kerucut. Penderita buta warna jenis ini masih dapat melihat satu warna tertentu, karena masih memiliki satu sel kerucut yang berfungsi.

2. Dikromasi

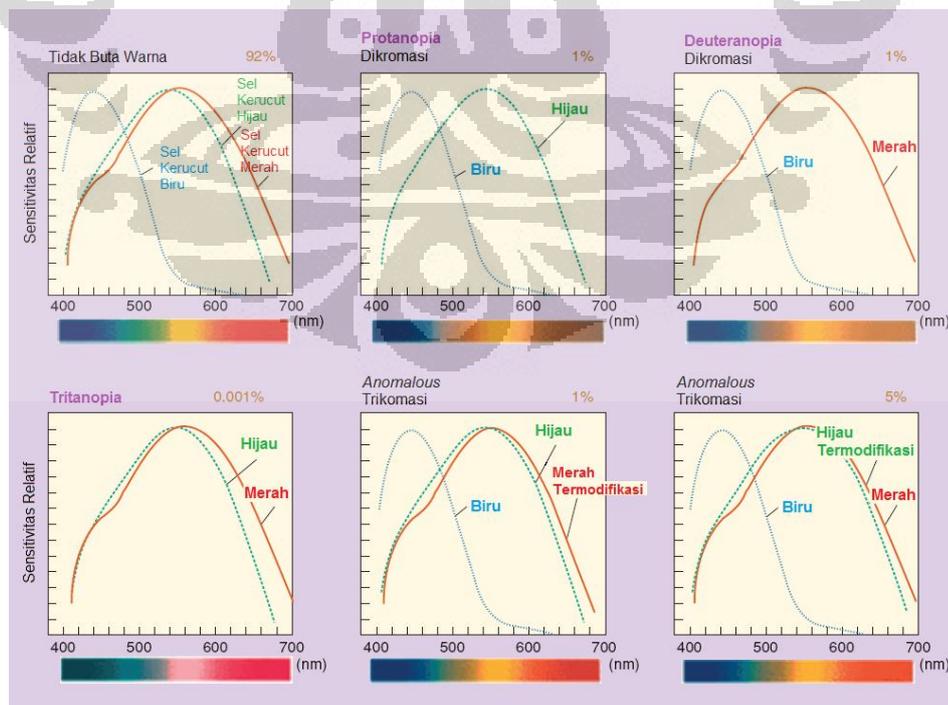
Dikromasi adalah jenis buta warna ketika salah satu dari tiga sel kerucut tidak ada atau tidak berfungsi. Dikromasi dibagi menjadi tiga bagian berdasarkan sel pigmen yang rusak.

- a. *Protanopia* adalah salah satu jenis dikromasi yang disebabkan oleh tidak adanya fotoreseptor retina merah. Pada penderita *protanopia*, penglihatan terhadap warna merah tidak ada. Dikromasi jenis ini terjadi pada 1% dari seluruh pria. *Protanopia* juga dikenal dengan buta warna merah-hijau.
- b. *Deutanopia* adalah gangguan penglihatan terhadap warna yang disebabkan tidak adanya fotoreseptor retina hijau. Hal ini menimbulkan kesulitan dalam membedakan *hue* pada warna merah dan hijau (*red-green hue discrimination*).
- c. *Tritanopia* adalah keadaan ketika seseorang tidak memiliki sel kerucut gelombang pendek (*short-wavelength cone*). Seseorang yang menderita *tritanopia* akan kesulitan membedakan warna biru dan kuning dari spektrum cahaya tampak. *Tritanopia* disebut juga buta warna biru-kuning dan merupakan jenis dikromasi yang sangat jarang dijumpai.

3. Kelainan Trikromasi (*Anomalous Trichromacy*)

Penyimpangan yang dialami dengan penglihatan trikromasi ini disebabkan faktor keturunan atau kerusakan mata setelah dewasa. Penderita *anomalous* trikromasi memiliki tiga sel kerucut, tetapi terjadi kerusakan mekanisme sensitivitas terhadap salah satu dari tiga sel reseptor warna tersebut.

- a. *Protanomali* adalah kelainan terhadap *long-wavelength pigment* (merah), sehingga menyebabkan rendahnya sensitifitas terhadap cahaya merah. Artinya, penderita *protanomali* tidak akan mampu membedakan warna dan melihat campuran warna yang dapat dilihat oleh mata normal. Penderita juga akan mengalami penglihatan yang buram terhadap warna spektrum merah. Hal ini mengakibatkan mereka dapat salah membedakan warna merah dan hitam.
- b. *Deuteranomali* disebabkan oleh kelainan pada bentuk pigmen *middle-wavelength* (hijau). Sama halnya dengan *protanomali*, *deuteranomali* tidak mampu melihat perbedaan kecil pada nilai *hue* dalam area spektrum untuk warna merah, jingga, kuning, dan hijau. Penderita salah dalam menafsirkan *hue* dalam region warna tersebut, karena *hue*-nya lebih mendekati warna merah. Penderita *deuteranomali* tidak memiliki masalah dalam hilangnya penglihatan terhadap tingkat kecerahan (*brightness*).
- c. *Tritanomali* adalah jenis *anomalous trichromacy* yang sangat jarang terjadi, baik pada pria maupun wanita. Pada *tritanomali*, kelainan terdapat pada *short-wavelength pigment* (biru). Pigmen biru ini bergeser ke area hijau dari spektrum warna.



Gambar 2.2. Spektrum sel kerucut pada mata normal dan buta warna ^[3]

2.3 Metode Daltonisasi

Daltonisasi atau suatu metode pengubahan warna citra dapat menjadi solusi untuk membantu permasalahan pembedaan warna pada suatu objek. Rentang warna yang tidak sempurna pada persepsi penglihatan orang buta warna diubah ke dalam nilai warna tertentu yang masih dapat ditoleransi oleh mata penderita buta warna. Saat ini, sudah ada beberapa metode transformasi warna yang dikembangkan, di antaranya adalah metode Daltonisasi Vischeck dan metode Daltonisasi Fidaner.

2.3.1 Daltonisasi Vischeck

Metode transformasi warna yang diajukan Vischeck adalah dengan melakukan pemisahan dimensi pada citra menjadi dimensi *bright/dark*, dimensi *red/green*, dan dimensi *blue/yellow* [15]. Untuk setiap informasi pada dimensi *red/green* (yang dianggap tidak tampak oleh penderita *deuteranopia*) akan menjadi faktor penambahan variasi pada dimensi *bright/dark* dan *blue/yellow*. Daltonisasi Vischeck dapat digunakan penderita buta warna untuk menerima informasi citra dengan lebih baik, karena bagian citra yang terlihat samar (dimensi *red/green*) diubah menjadi dimensi *bright/dark* dan *blue/yellow* yang merupakan dimensi yang terlihat lebih jelas bagi penderita *deuteranopia*.

2.3.2 Daltonisasi Fidaner

Metode transformasi warna yang diajukan Onur Fidaner, Poliang Lin, dan Nevran Ozguven, adalah dengan melakukan dua tahap untuk membedakan warna, yaitu dengan transformasi warna menjadi warna yang dapat dilihat oleh penderita buta warna, dan kemudian dilakukan proses daltonisasi [15].

Dalam metode transformasi warna Daltonisasi Fidaner, digunakan nilai variabel yang dihasilkan berdasarkan sel kerucut warna pada mata (LMS). Nilai-nilai LMS tersebut didapatkan dengan transformasi RGB ke LMS melalui perkalian matriks dengan nilai RGB warna awal. Kemudian dilakukan modifikasi nilai-nilai LMS sesuai dengan jenis buta warna yang diderita sehingga menghasilkan nilai L'M'S' yang berbeda, tergantung pada jenis buta warna. Langkah selanjutnya adalah mentransformasikan kembali LMS ke RGB melalui

perkalian matriks sehingga didapat R'G'B'. Untuk tahap berikutnya, dilakukan proses daltonisasi dengan mengukur tingkat kesalahan warna, yaitu dari warna asli dan warna yang dihasilkan setelah simulasi buta warna. Perhitungan kesalahan warna ini dilakukan secara linear dengan mengurangi RGB warna awal dengan RGB warna hasil simulasi.

2.4 Konsep *Augmented Reality*

Augmented Reality (AR) atau dikenal dengan istilah realitas tertambah merupakan sebuah teknologi baru di bidang multimedia. AR adalah hasil pengembangan dari teknologi pendahulunya, yaitu *Virtual Reality* (VR). Istilah VR diperkenalkan pada tahun 1986, yang memiliki definisi sebagai sebuah teknologi yang dapat meyakinkan penggunanya bahwa sebenarnya dia sedang berada di tempat lain dengan menggunakan data yang dihasilkan oleh komputer [16].

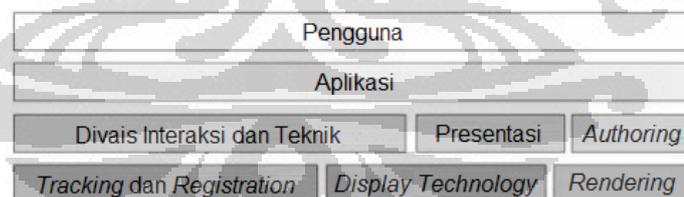
Augmented Reality sendiri adalah teknologi yang menggabungkan benda maya dalam dua dimensi atau tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi, lalu memproyeksikan benda-benda tersebut secara *real-time* [17]. Benda-benda maya yang ditambahkan menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat realitas tertambah sesuai sebagai alat untuk membantu persepsi dan interaksi penggunanya dengan dunia nyata.

Dalam penelitian ini, sistem akan melibatkan proses AR yang digabungkan dengan pengolahan citra. Salah satu fitur yang akan dirancang dengan menerapkan konsep AR ini adalah pembedaan warna yang terlihat sama pada penglihatan penyandang buta warna. Kelainan yang diderita oleh penyandang buta warna tentu mempengaruhi cara dalam membedakan beberapa warna. Sebagai contoh, untuk penderita buta warna jenis *deuteranomali* (kelemahan terhadap penglihatan warna hijau), mereka pada umumnya sulit membedakan warna coklat dan hijau dengan intensitas tertentu. Dengan konsep AR ini, sistem akan dibuat agar dapat membandingkan warna yang tidak dapat dibedakan tersebut, kemudian akan ditambahkan objek tambahan pada citra yang *real-time* dengan menaikkan

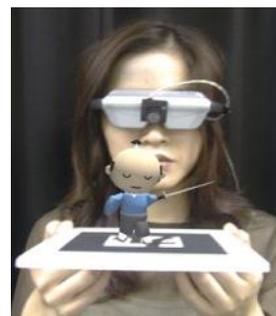
intensitas warna tertentu agar dapat dibedakan. Objek tambahan berupa perubahan intensitas warna inilah yang menjadi dasar dalam penggabungan konsep AR dan pengolahan citra untuk sistem bantuan buta warna yang akan dibangun.

Akan tetapi, yang menjadi masalah utama dalam pengembangan teknologi atau aplikasi AR adalah bagaimana menggabungkan lingkungan nyata dengan objek-objek virtual yang ingin ditambahkan ke dalam lingkungan nyata tersebut. Ada 3 unsur dasar dalam pengembangan sistem AR, yaitu *tracking* dan *registration*, *display technology*, serta *rendering*. *Tracking* adalah bagaimana menangkap dan melacak lingkungan nyata, sedangkan *registration* berhubungan dengan bagaimana membentuk hubungan antara lingkungan asli dengan penambahan objek maya. Unsur kedua, yaitu *display technology* merupakan teknologi yang menjadi antarmuka langsung antara pengguna dengan sistem AR, lingkup tersebut salah satu fokus dalam penelitian rancang sistem bantuan bagi penyandang buta warna. Unsur ketiga, yaitu *rendering* merupakan proses penggabungan kondisi nyata dengan objek *artificial* sehingga dapat menghasilkan keluaran yang utuh.

Berikut merupakan tiga unsur dasar pengembangan sistem AR yang digambarkan dalam *building block*.



Gambar 2.3. *Building Block* pada *Augmented Reality* ^[18]



Gambar 2.4. Contoh pengaplikasian sistem AR

2.5 Pemanfaatan Teknologi untuk Implementasi

Untuk merancang antarmuka pengguna sistem bantuan untuk penyandang buta warna yang kemudian diintegrasikan dengan kode program sistem transformasi warna, tes buta warna, dan fitur lainnya seperti deteksi warna, dibutuhkan beberapa aplikasi seperti *Microsoft Expression Blend*, *Windows Phone 7 Developer Tool*, dan *Microsoft Visual Studio*. Ketiga perangkat lunak tersebut merupakan produk pengembang perangkat lunak Microsoft yang terbilang masih baru. Bahkan di Indonesia, pengguna *Windows Expression Blend* dan pengembang aplikasi untuk *Windows Phone 7* masih sangat jarang. Untuk mengetahui penggunaan dan fungsinya, berikut akan dijelaskan mengenai masing-masing perangkat lunak tersebut.

2.5.1 *Microsoft Expression Blend*

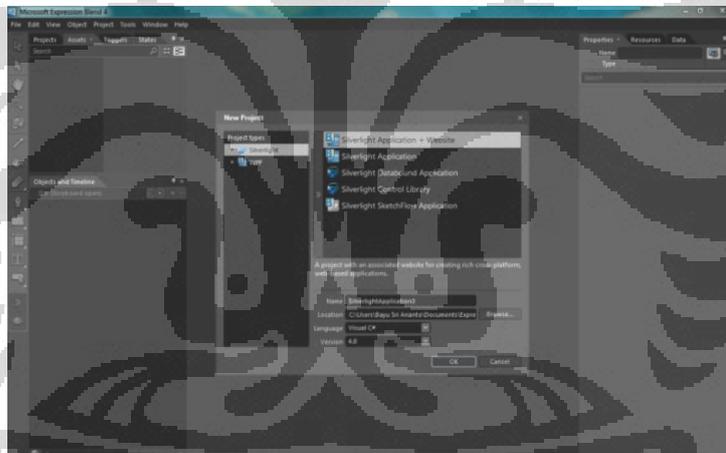
Microsoft telah menciptakan sebuah bahasa *mark-up* sederhana yang baru, yang disebut *Extensible Application Markup Language (XAML)*, yang dapat menggambarkan dengan mudah semua detail antarmuka pengguna untuk generasi ke depan pada *Windows* dan aplikasi *web* tanpa perlu kode komputer yang prosedurnya rumit [19].

Microsoft Expression Blend adalah aplikasi buatan Microsoft yang menerapkan sistem berbahasa XAML. Aplikasi ini merupakan sebuah alat bantu desain antarmuka yang digunakan untuk membuat sebuah aplikasi *web* atau aplikasi *desktop* yang *user friendly*. Perancangan desain antarmuka yang dilakukan dengan menggunakan aplikasi *Microsoft Expression Blend* berbasis *What You See Is What You Get (WYSIWYG)*. Pendesain tidak perlu membuat kode-kode program untuk menggambar sebuah komponen pada antarmuka, namun dapat menghasilkan sebuah aplikasi yang menarik dan interaktif.

Microsoft Expression Blend menyediakan berbagai perangkat komponen siap pakai, kontrol *Windows Presentation Foundation (WPF)* yang sangat fungsional, dan panel yang dapat digunakan untuk merancang antarmuka pengguna dengan lebih cepat dan mudah, bahkan bisa dilakukan tanpa pemrograman. Pengeditan dan pemrograman pada *Microsoft Expression Blend*

juga dapat diintegrasikan langsung dengan aplikasi *Visual Studio* sehingga memberikan kemudahan bagi pengembang untuk menambahkan kode prosedural untuk antarmuka pengguna.

Untuk memulai sebuah proyek pada *Expression Blend*, pengguna harus menentukan terlebih dahulu jenis aplikasi yang akan dibuat, dan secara umum terdapat 2 pilihan, yaitu WPF (untuk aplikasi *desktop*) atau *Silverlight* (lebih mengarah kepada aplikasi bergerak). Selain itu, disediakan pula pilihan desain prototipe atau disebut aplikasi *SketchFlow*. Secara umum, dalam penelitian ini, jenis aplikasi untuk antarmuka pengguna yang akan dibuat dengan menggunakan *Microsoft Expression Blend* adalah dengan menggunakan *Silverlight*. Pada dasarnya, *Silverlight* memiliki fungsi yang sama, namun *Silverlight* dapat mengintegrasikan ke dalam sebuah halaman HTML dengan menggunakan *plugin Silverlight* untuk *browser*.

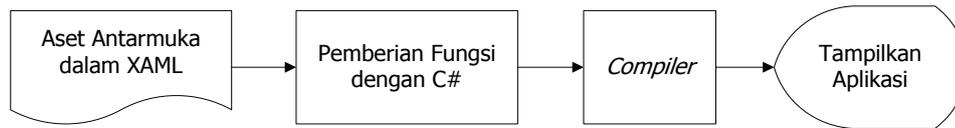


Gambar 2.5. Tampilan memilih jenis proyek pada *Microsoft Expression Blend*

Alur cara *Silverlight* dalam membuat serta menampilkan konten secara tipikal adalah seperti berikut [20]:

1. XAML menggambarkan bagaimana kontrol, gambar, video, dan aset lainnya ditampilkan.
2. C# memberikan aset-aset tersebut fungsi.
3. *Compiler* kemudian menempatkan XAML antarmuka pengguna yang dibuat beserta fungsinya menjadi sebuah *file* yang *executable* (EXE) untuk *Silverlight desktop* atau aplikasi *web*.

4. *Browser*, komputer, atau perangkat sejenisnya, kemudian akan menampilkan aplikasi.



Gambar 2.6. Proses menampilkan antarmuka dengan *Silverlight* pada *Blend*

Jadi, kekuatan sebenarnya pada pengembangan menggunakan *Silverlight*, dalam hal ini perancangan antarmuka yang menggunakan aplikasi dasar *Microsoft Expression Blend*, adalah hubungan integrasi C# dan kemampuan menggunakan XAML untuk desain.

Berdasarkan semua pemaparan mengenai *Expression Blend* tersebut, pendesain tentu dapat pula membuat objek atau *item* seperti kurva, media, animasi, *textbox*, *dropdownlist*, dan sebagainya dengan cara *drag and drop* pada area kerja. Area kerja pada aplikasi ini berupa berkas dengan tipe ekstensi .XAML, jadi berkas hasil rancangan merupakan format teks. Dengan demikian, file yang dibuat dapat dibuka dengan teks editor biasa seperti *Notepad*.

Pada *Expression Blend 4*, yang dirilis awal Juni 2010, sudah mendukung *Silverlight 4* dan WPF 4, dengan tetap kompatibel untuk versi *Silverlight* dan WPF sebelumnya. Penambahan dan pengembangan lainnya adalah bentuk-bentuk *library*, pengembangan dukungan berkas *Photoshop*, integrasi lapisan ganda pada animasi, audio, video, gambar vektor, penambahan piksel efek *shader*, *path layout*, efek transisi, dukungan pola *model view – view model*, dukungan .ZIP untuk *file* proyek dan *template*, dan *data binding* otomatis ke dalam antarmuka pengguna [21].

2.5.2 *Windows Phone 7 Developer Tool*

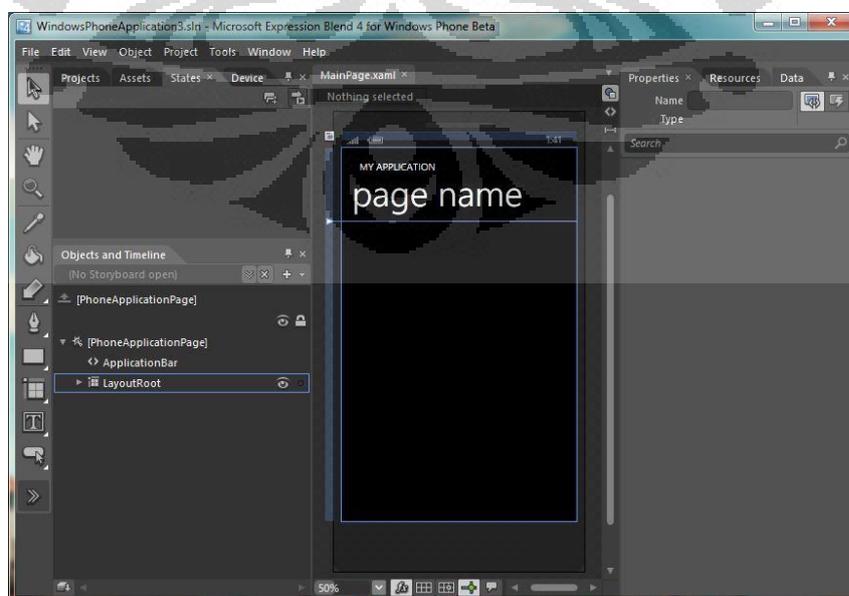
Windows Phone 7 (WP7) merupakan generasi baru penerus *Windows Mobile* yang diluncurkan pada Oktober 2010. Microsoft menyebut *Windows Phone* sebagai “*a revolutionary new platform*” [22]. Perubahan yang cukup radikal dengan berbagai kelebihanannya dan fitur yang ditawarkan, mendorong banyak pengembangan terhadap aplikasi yang dapat berjalan di atas *platform*

WP7. Untuk pengembangan aplikasi tersebut, dibutuhkan aplikasi seperti *Windows Phone 7 Developer Tool*.

Aplikasi ini merupakan semacam *plugin* yang dapat dijalankan pada perangkat lunak seperti *Microsoft Visual Studio* dan *Microsoft Expression Blend*. Pengguna harus terlebih dulu menginstal aplikasi ini sehingga ketika pengguna akan memulai untuk membangun sebuah aplikasi bergerak yang berbasis *Windows Phone 7*, jenis proyek yang dipilih saat menggunakan *Microsoft Visual Studio* atau *Microsoft Expression Blend* adalah *Windows Phone Application*. Pada perancangan antarmuka menggunakan perangkat ini, pengguna akan dimudahkan dengan disediakan beberapa *library* dan komponen yang dibutuhkan untuk mendesain. Selain itu, untuk area kerja pada aplikasi sudah disediakan *template* atau halaman yang mendefinisikan tampilan *Windows Phone 7*.

Untuk versi yang sudah dirilis, penginstalasian *Windows Phone Developer Tool* terdiri dari:

- *Microsoft Expression Blend* untuk *Windows Phone*.
- *Visual Studio 2010 Express* untuk *Windows Phone*.
- *Windows Phone Emulator Resources*.
- *Silverlight 4 Tools* untuk *Visual Studio*.
- *XNA Game Studio 4.0*.

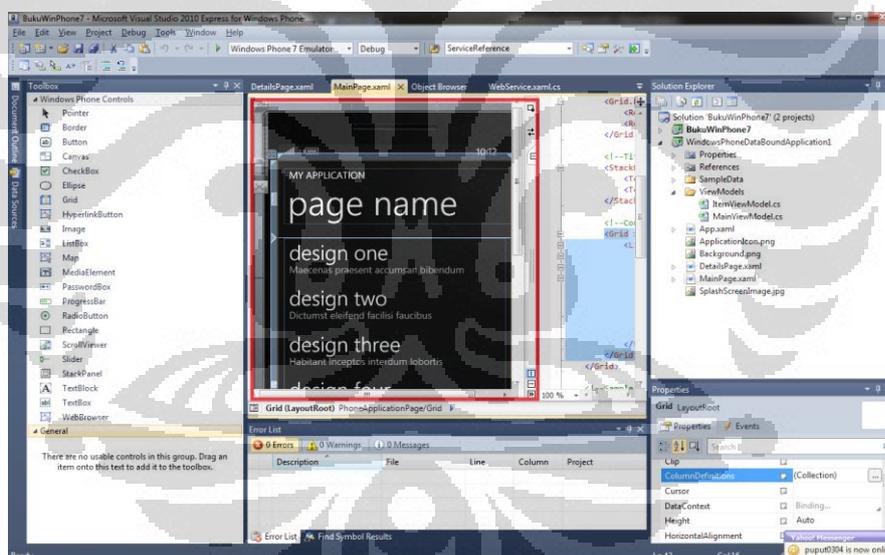


Gambar 2.7. Antarmuka pengguna *Windows Phone 7* pada *Expression Blend*

Secara umum, penelitian yang akan dilakukan untuk merancang aplikasi virtual untuk membantu penyandang buta warna adalah menggunakan *Microsoft Expression Blend* sebagai rancang antarmuka pengguna, dan *Microsoft Visual Studio* yang akan mengintegrasikan kode program untuk proses pengolahan citra dan data dengan antarmuka pengguna sehingga menjadi aplikasi yang interaktif. Adapun bahasa pemrograman yang digunakan adalah C#.

2.5.3 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi konsol, aplikasi *Windows*, ataupun aplikasi *web*. *Visual Studio* mencakup *compiler*, *SDK*, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*).



Gambar 2.8. Tampilan *Visual Studio 2010 Express* untuk *Windows Phone*

Microsoft Visual Studio juga dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

BAB 3

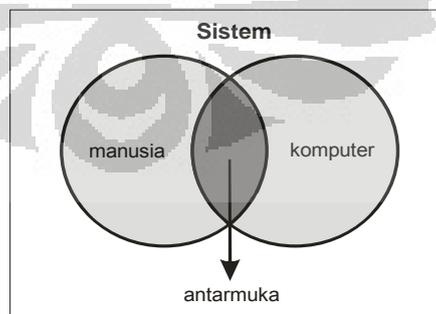
PERANCANGAN ANTARMUKA PENGGUNA, PENGETESAN BUTA WARNA, DAN TRANSFORMASI WARNA

3.1 Perancangan Antarmuka Pengguna Sistem Bantuan Buta Warna

Dalam merancang antarmuka pengguna, khususnya bagi penyandang buta warna, sebenarnya tidak berbeda dengan merancang antarmuka bagi orang berpenglihatan normal, hanya saja pemilihan tema warna pada aplikasi akan menjadi hal yang lebih serius untuk diperhatikan. Oleh sebab itu, sebelum melakukan perancangan antarmuka bagi sistem bantuan penyandang buta warna, ada prinsip-prinsip dasar yang harus menjadi acuan sehingga pengembangan sistem aplikasi yang dibuat dapat digunakan dengan mudah oleh pengguna dari berbagai kalangan, bahkan pengguna berpenglihatan normal sekalipun, dan pengguna tertarik untuk menggunakan sistem yang dibangun.

3.1.1 Prinsip Dasar Desain Antarmuka Pengguna

Antarmuka pengguna merupakan bentuk tampilan grafis yang menghubungkan antara pengguna dengan sistem sehingga memudahkan pengguna dalam menjalankan suatu program dan eksekusi perintah-perintah yang ada pada sistem. Oleh sebab itu, selain pemahaman dasar mengenai perangkat dan teknik, pengetahuan lain yang dibutuhkan oleh seseorang di bidang teknik adalah proses desain antarmuka.



Gambar 3.1. Hubungan interaksi manusia dan komputer

Ada beberapa hal yang harus dilakukan dalam perancangan antarmuka agar sistem dapat berjalan sesuai kegunaan dan fungsinya. Berikut adalah tiga prinsip dasar dalam mendesain antarmuka pengguna menurut Gould dan Lewis [22].

1. Terlebih dahulu mengetahui kebutuhan pengguna dan selalu fokus pada pengguna dan fungsi tugas sistem.
2. Pengukuran empiris berdasarkan penggunaan (seperti dengan melakukan uji coba kegunaan atau *usability testing*).
3. Desain yang iteratif.

Prinsip dasar tersebut menjadi perhatian penting untuk memahami pengguna (*end user*) dan menyesuaikan kebutuhan desain yang dapat berubah berdasarkan tanggapan pengguna. Dalam penelitian ini, antarmuka menjadi sangat penting, karena sasaran pengguna adalah penyandang buta warna. Unsur warna, tata letak, komponen, dan fungsionalitas sistem, akan turut mempengaruhi sikap pengguna dalam menjalankan dan pemakaian sistem aplikasi bantuan bagi penyandang buta warna. Dengan demikian, analisis dan perancangan antarmuka, baik dari segi teknis maupun estetika, perlu dikembangkan untuk mencapai kepuasan pengguna dalam menggunakan sistem perangkat yang dibuat.

Deborah J. Mayhew mengungkapkan tiga langkah dasar yang dapat meningkatkan rancangan antarmuka pengguna [23], hal ini sekaligus menjadi acuan penulis dalam mendesain antarmuka pada sistem bantuan bagi penyandang buta warna. Langkah pertama untuk menuju perbaikan dan pengembangan antarmuka pengguna melibatkan perubahan sikap pada perangkat lunak handal, itu berarti sebuah perangkat lunak yang profesional harus membuat antarmuka yang semakin berkualitas. Pendesain perlu memikirkan perancangan sebuah antarmuka pengguna sebagai tantangan dan menstimulasi masalah ke dalam algoritma perancangan yang elegan dengan struktur yang baik, dan kode yang dapat di modifikasi dengan mudah. Perancangan sistem bantuan bagi buta warna tentu sudah menjadi tantangan sendiri. Antarmuka yang perlu dibentuk tidak hanya terlibat pada algoritma pengolahan sistem, namun komponen pada tiap halaman antarmuka menjadi perhatian utama, termasuk masalah pemilihan warna tema aplikasi yang dibuat.

Langkah kedua untuk desain antarmuka yang baik yaitu mengamati pola tingkah laku manusia dan segi psikologisnya, kemudian memanfaatkan akumulasi pengetahuan tentang desain antarmuka. Dengan demikian, pengembang perangkat

lunak dapat membimbing penggunaanya dalam menjalankan sistem aplikasi yang dibangun. Jadi, pengguna dari kalangan manapun, bahkan seorang awam yang baru menggunakan aplikasinya dimudahkan dengan petunjuk yang menjelaskan bagaimana alur pemakaian program tersebut.

Langkah ketiga agar sukses merancang antarmuka adalah mengintegasi metode desain antarmuka pengguna dan tekniknya ke dalam standar metodologi pengembangan perangkat lunak yang umum. Semua itu dimulai dengan memperhatikan aturan desain dan diakhiri dengan tes oleh pengguna.

Dengan menerapkan prinsip dasar dalam rancang antarmuka pengguna, berikut adalah sistematika penelitian.



Gambar 3.2. Sistematika penelitian

3.1.2 Desain Interaksi Manusia – Komputer Prinsip *Barrier-free*

Pada penelitian ini, penulis menerapkan desain interaksi antara manusia berbasis prinsip *barrier-free* (bebas hambatan) yang bertujuan agar perancangan, perencanaan struktur bertingkat dari antarmuka, tata letak dan kepraktisan kontrol untuk perangkat lunak, sesuai dengan operasional kebiasaan pengguna sehingga untuk mengurangi kelelahan, mengurangi kesalahan, meningkatkan efisiensi, dan mendapatkan kemudahan ketika menggunakan perangkat lunak.

Prinsip *barrier-free* membagi jenis hambatan dalam perilaku untuk berbagai jenis orang yang memiliki kesulitan atau kelemahan ke dalam tiga kategori, yaitu [24]:

- Hambatan bergerak, contohnya orang-orang yang tidak bisa berjalan atau membutuhkan suatu perangkat khusus untuk bergerak.
- Hambatan kompleks dalam bergerak, contohnya orang-orang yang memiliki anggota tubuh tidak lengkap atau bagian tubuh yang tidak dapat terkoordinasi, seperti kelumpuhan syaraf.
- Hambatan informasi, contohnya seperti penderita buta warna atau kelemahan penglihatan terhadap warna, rabun, tuli, dan sebagainya.

Prinsip *barrier-free* menunjukkan berbagai konsep dalam desain yang bertujuan untuk meringankan hambatan perilaku manusia tersebut.

Sampai saat ini, permasalahan orang dengan defisiensi warna masih belum banyak terbantu dengan desain interaktif. Berdasarkan prinsip ini, untuk mengurangi kesulitan dalam interaksi, setidaknya ada dua kriteria yang perlu diterapkan dalam rancang antarmuka aplikasi, khususnya untuk sistem bantuan buta warna, yaitu untuk bantuan operasi (penggunaan) dan bantuan tulisan (informasi tambahan).

- Bantuan pengoperasian:
 1. Meningkatkan rasio kontras cahaya dan latar elemen-elemen kontrol yang berbeda pada antarmuka, karena penderita buta warna sensitif terhadap perbedaan luminansi kontrol, pola, atau grafik.

2. Mengurangi jumlah ikon dan tombol untuk memberikan tampilan yang lebih sederhana tapi menarik.
- Bantuan tulisan atau bacaan:
 1. Tulisan: menggunakan metode tampilan informasi tambahan, seperti tampilan bantuan saat akan menggunakan fitur tertentu atau saat fitur dijalankan.
 2. Suara: menggunakan metode yang dapat memberikan informasi berupa suara, seperti yang diterapkan pada sistem untuk perangkat tertanam.

3.1.3 Rancangan Antarmuka Pengguna

Seperti yang telah ditunjukkan pada Gambar 3.2 tentang sistematika penelitian bahwa sebelum merancang alur atau urutan-urutan skenario dalam menggunakan aplikasi, terlebih dahulu perlu ditentukan fitur-fitur apa saja yang akan disediakan.

Fitur-fitur yang tersedia dalam sistem aplikasi bantuan bagi penyandang buta warna pada prototipe perangkat *embedded* ada empat jenis. Pada fitur standar, yaitu menampilkan jenis warna pada objek yang ditangkap oleh kamera dalam bentuk teks. Kemudian pengguna dapat juga memilih fitur untuk menunjukkan warna pada objek dalam bentuk suara (tanpa teks) dengan cara menunjuk objek yang ditangkap oleh kamera. Fitur lainnya adalah fitur untuk membedakan warna yang dilihat mirip pada penglihatan penyandang buta warna dengan menaikkan intensitas salah satu warna pada objek yang terlihat mirip tersebut, dan semua program dijalankan dengan perintah suara.

Untuk aplikasi bergerak, fitur yang ada terdiri dari tes buta warna, deteksi warna, dan transformasi warna. Dari keseluruhan program yang dirancang, untuk implementasi antarmuka pengguna, penulis membatasi pada desain antarmuka pada perangkat *Windows Phone 7*.

Desain antarmuka pengguna untuk perangkat *Windows Phone 7* ini dilakukan tidak hanya berdasarkan unsur estetika, tapi juga kemudahan penggunaan, dan pemanfaatan fitur layar sentuh pada perangkat.



Gambar 3.3. Sketsa rancangan antarmuka sistem bantu pada perangkat bergerak

Sketsa rancangan tersebut adalah desain antarmuka untuk tiap-tiap halaman yang dapat diakses oleh pengguna dalam menggunakan aplikasi tersebut beserta *workflow*-nya. Navigasi tiap halaman ke halaman selanjutnya memanfaatkan komponen interaksi pengguna berupa tombol dan pada fitur *Capture Color* digunakan fungsi sentuh dan geser untuk transisi antar halaman, atau pada *Windows Phone 7* disebut dengan *Panorama Page*.

Sketsa level rancangan antarmuka yang ditunjukkan pada Gambar 3.3 di atas, diimplementasikan dengan desain yang dapat diintegrasikan langsung

dengan program. Untuk merancang kesatuan antarmuka tersebut, digunakan aplikasi *Expression Blend*. Pada *Expression Blend*, semua sketsa digambarkan dengan komponen yang tersedia pada *toolbox* dan dikerjakan di area desain. Penambahan gambar atau animasi, ada bagian *properties* yang dapat mengimpor gambar-gambar untuk digunakan dalam desain. *Interaction panel* membantu membuat tampilan yang lebih dinamis dengan menampilkan efek-efek, seperti efek transisi. Semua halaman yang telah didesain tersebut dapat langsung dihubungkan berdasarkan *workflow*-nya. Langkah selanjutnya, hanya perlu mengintegrasikan navigasi kontrol atau elemen pada antarmuka pengguna dengan kode program pada sistem.



Gambar 3.4. Bagian-bagian pada *Expression Blend* untuk desain antarmuka

3.2 Sistem Tes Buta Warna dengan Metode Ishihara

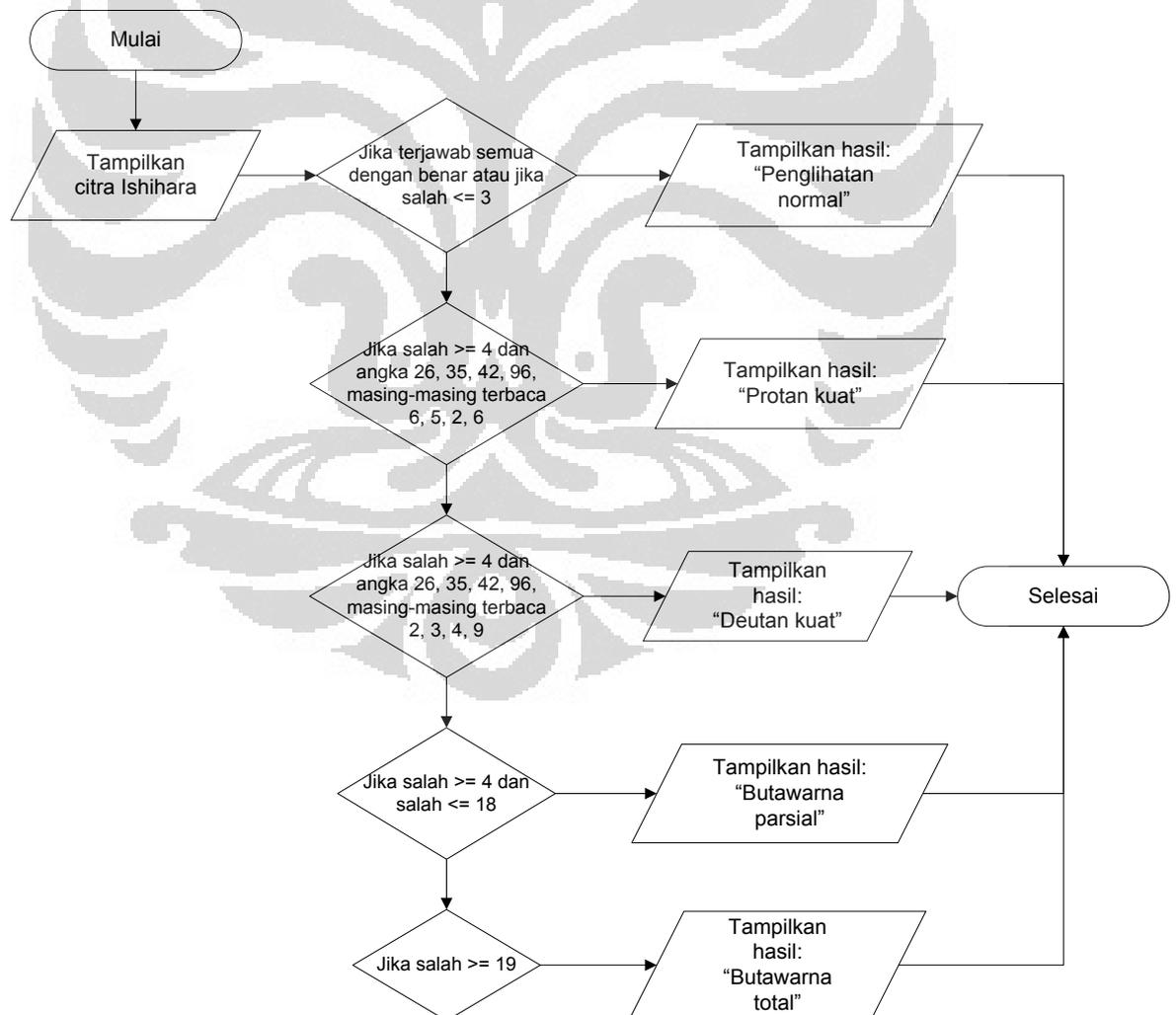
Metode Ishihara hasil pengembangan Dr. Shinobu Ishihara yang pertama kali dipublikasikan pada tahun 1917 di Jepang merupakan metode yang paling banyak digunakan untuk menentukan seseorang menderita buta warna atau tidak. Metode tes buta warna Ishihara menggunakan lembaran yang didalamnya terdapat titik-titik dengan berbagai warna dan ukuran atau disebut *Ishihara plates*. Titik berwarna tersebut disusun sehingga membentuk suatu lingkaran. Di bagian tengahnya membentuk suatu angka atau objek. Pola demikian tersebut dibuat sedemikian rupa sehingga orang buta warna tidak dapat melihat perbedaan warna.

Pada tes buta warna, lembaran Ishihara yang digunakan biasanya berjumlah 38 lembar, atau 24 lembar. Ada juga yang menggunakan 14 lembar dengan model warna dan bentuk yang berbeda. Dalam penelitian ini digunakan 21 citra Ishihara yang telah disesuaikan dengan tes buta warna pada umumnya. Pembagian fungsionalitas 21 citra Ishihara yang digunakan pada sistem bantuan untuk penyandang buta warna ditunjukkan pada Tabel 3.1.

Tabel 3.1. Desain dan fungsi 21 citra Ishihara ^[25]

FUNGSI CITRA ISHIHARA	
Halaman pada citra Ishihara 38 Lembar	1
Urutan pada aplikasi Chromophore	5
Angka pada citra Ishihara	12
Tipe Desain	Pendahuluan
Fungsi	Citra dapat dibaca oleh orang berpenglihatan normal dan orang buta warna.
Halaman pada citra Ishihara 38 Lembar	2 – 9
Urutan pada aplikasi Chromophore	2, 4, 6, 9, 13, 15, 18, 20
Angka pada citra Ishihara	3, 8, 15, 29, 57, 74, 5, 6
Tipe Desain	Pentransformasian
Fungsi	Angka pada citra yang terlihat dengan orang berpenglihatan normal berbeda dengan angka yang terlihat pada orang yang berpenglihatan lemah terhadap warna merah-hijau.
Urutan pada citra Ishihara 38 Lembar	10 – 17
Urutan pada aplikasi Chromophore	1, 3, 7, 12, 14, 17, 19, 21
Angka pada citra Ishihara	2, 7, 16, 45, 73, 6, 97, 5
Tipe Desain	Penghilangan
Fungsi	Angka pada citra terlihat oleh orang yang berpenglihatan normal, tetapi tidak terlihat oleh orang dengan kelemahan penglihatan terhadap warna merah-hijau.
Urutan pada citra Ishihara 38 Lembar	22 – 25
Urutan pada aplikasi Chromophore	8, 10, 11, 16
Angka pada citra Ishihara	26, 35, 42, 96
Tipe Desain	Pengklasifikasian
Fungsi	Ada dua angka pada tiap citra Ishihara tipe ini. <i>Protan</i> hanya dapat melihat angka di sebelah kanan. <i>Deutan</i> hanya dapat melihat angka di sebelah kiri. Catatan: Jika kedua angka terlihat, tipe <i>protan</i> lemah akan lebih jelas melihat angka sebelah kanan dibandingkan angka sebelah kiri, dan sebaliknya untuk tipe <i>deutan</i> .

Citra Ishihara yang biasa digunakan memiliki warna dominan antara merah dan hijau sehingga hanya dapat digunakan untuk mengetahui buta warna parsial terhadap warna merah-hijau. Buta warna parsial terhadap warna biru-kuning akan sulit diketahui dari tes ini karena citra Ishihara sedikit sekali menggunakan warna biru dan kuning. Dengan demikian, fitur tes buta warna yang diimplementasikan pada perangkat bergerak, hanya akan menentukan jenis karakter penglihatan terhadap warna tipe *protan* kuat (kelemahan penglihatan terhadap warna merah), *deutan* kuat (kelemahan penglihatan terhadap warna hijau), buta warna total, normal, dan apabila kondisi di atas tidak ada yang terpenuhi atau dengan kata lain ada kemungkinan pengguna adalah tipe *protan* lemah, *deutan* lemah, atau *tritan* (kelemahan terhadap warna biru), maka sistem akan mengenali pengguna sebagai buta warna parsial.



Gambar 3.5. Algoritma sistem tes buta warna

3.3 Sistem Transformasi Warna

Seorang penyandang buta warna parsial terkadang dihadapkan dengan kesulitan membedakan warna. Berikut merupakan tipikal keambiguan warna pada tipe yang berbeda untuk penglihatan lemah terhadap warna.

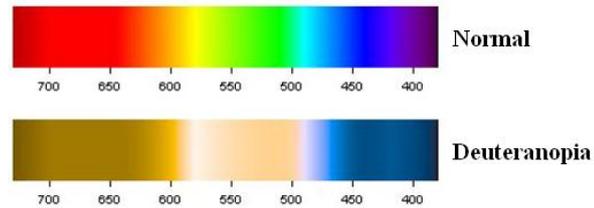
Tabel 3.2. Tipikal warna ambigu untuk tipe penglihatan lemah terhadap warna ^[25]

WARNA AMBIGU		
Buta Warna Jenis Protan	Buta Warna Jenis Deutan	Buta Warna Jenis Tritan
Merah/kuning/ kuning-hijau	Merah/kuning/ kuning-hijau	
Merah-ungu/ abu-abu/hijau	Ungu/abu-abu/ biru-hijau	Violet/abu-abu/ kuning-hijau
Merah/hitam	Hijau/hitam	Biru/hitam
Ungu/biru	Ungu/biru	Ungu/merah
Jingga/kuning	Jingga/kuning	
Merah/coklat, hijau/coklat	Merah/coklat, hijau/coklat	
Merah/biru dan coklat/violet	Merah/biru dan coklat/violet	Hitam/putih dan biru/kuning

Pembedaan warna pada sistem transformasi warna hanya mengubah kontras warna. Penyandang buta warna dapat mengetahui bahwa objek tersebut memiliki warna yang berbeda dan penyandang buta warna pun bisa sedikit terbantu untuk mengetahui warna sebenarnya pada objek. Dengan menaikkan intensitas warna, kelemahan penglihatan warna tertentu akan disamarkan dengan memunculkan warna yang lebih tajam yang dapat ditoleransi dan diterima oleh mata penyandang buta warna. Prinsip inilah yang dilandasi dari teori dasar Dalton, seorang ilmuwan asal Inggris yang menjadi orang pertama yang meneliti buta warna.

Teknik yang digunakan dalam memperbaiki olah citra penglihatan buta warna agar dapat dibedakan adalah dengan menggabungkan 2 metode yang disebut sebagai proses daltonisasi [26]. Berikut merupakan proses daltonisasi:

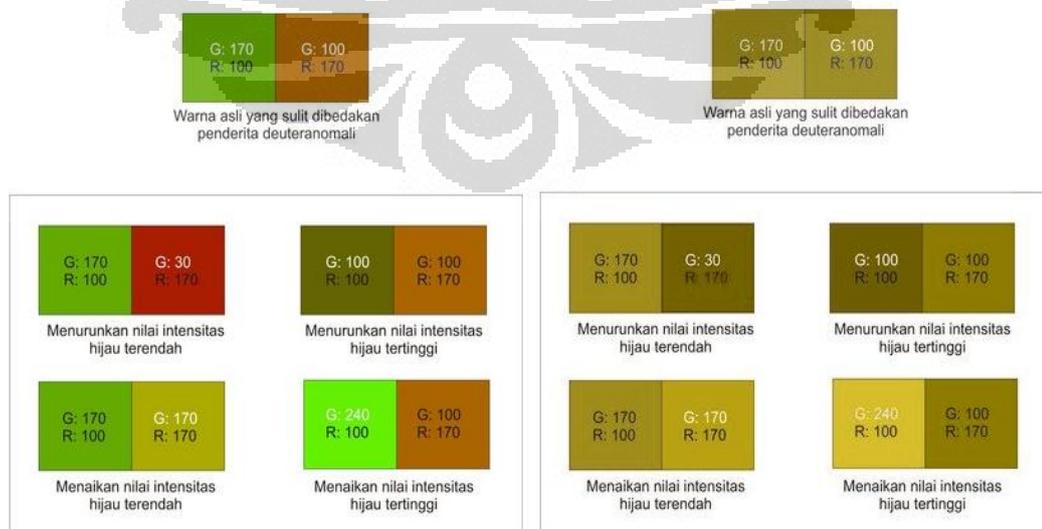
1. Menaikkan kontras warna asli sehingga terlihat perbedaan variasi warna yang ada pada citra yang sebenarnya.
2. Analisis informasi citra yang kemudian dipetakan dalam dimensi warna yang dapat dilihat oleh orang-orang buta warna.



Gambar 3.6. Hasil proses daltonisasi menggunakan aplikasi *Vischeck*

Berdasarkan contoh hasil proses daltonisasi pada Gambar 3.6, dapat diketahui bahwa hilang atau berkurangnya fungsi penglihatan terhadap warna tertentu akan berdampak pada munculnya warna-warna yang terlihat sama. Seperti contoh perbandingan penglihatan mata normal dan *deuteranopia* (hilang fungsi penglihatan warna hijau) pada gambar tersebut, maka penderita buta warna jenis ini akan melihat warna hijau sebagai warna krem. Apabila ada objek berwarna hijau dan coklat krem berdekatan, maka mereka akan sulit membedakan warna kedua objek tersebut.

Sebagai analisa untuk membuktikan keefektifan pembedaan warna dengan cara meningkatkan intensitas pada salah satu warna yang ditentukan berdasarkan perbandingan tersebut, digunakan aplikasi *Vischeck* [27] untuk mensimulasikan penglihatan buta warna *deuteranomali* sebagai contoh. Berikut merupakan hasil simulasi perbandingan 2 warna yang dilihat dengan mata normal dan dilihat dengan penglihatan buta warna jenis *deuteranomali* atau mengalami kelemahan melihat warna hijau.



Gambar 3.7. Perbandingan dua warna dilihat mata normal dan *deuteranomali*

Berdasarkan pada Gambar 3.7, menaikkan warna intensitas tertinggi ternyata lebih efektif dibandingkan dengan menurunkan nilai intensitas terendah. Jadi, dengan algoritma ini, penyandang buta warna akan dapat mengetahui bahwa dua objek memiliki warna yang berbeda. Kelemahan sistem ini adalah kadang pengguna kalangan buta warna hanya dapat mengetahui bahwa ada dua objek atau lebih dengan warna yang berbeda yang sebelumnya terlihat sama. Mereka tidak dapat mengetahui warna asli objek tersebut secara akurat. Untuk mengatasi masalah tersebut, pengguna dapat memilih fitur standar (deteksi warna) yang akan menampilkan label warna pada tiap objek.

3.4 Integrasi untuk Sistem Bantuan Buta Warna

Untuk implementasi dan integrasi rancangan digunakan bahasa pemrograman C# pada Visual Studio 2010. C# merupakan salah satu bahasa pemrograman tingkat tinggi yang digunakan untuk membangun aplikasi di atas .NET *Framework*. C# juga termasuk bahasa pemrograman berorientasi objek yang memungkinkan pengembang perangkat lunak untuk mengembangkan aplikasi *desktop*, maupun aplikasi untuk *platform Windows 7*. C# dirancang untuk mengatasi kelemahan pada bahasa pemrograman lain, dan dengan kesederhanaan dan performa yang ditawarkan oleh penggunaan bahasa C# [28]. Dalam penelitian yang mengintegrasikan realitas tertambah dengan sistem bantuan bagi penyandang buta warna digunakanlah bahasa pemrograman tersebut.

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. OpenCV pada dasarnya dibutuhkan untuk mengefektifkan proses komputasi serta mendukung aplikasi yang bersifat *real-time*. Open CV sendiri merupakan sebuah *library* yang menyediakan fungsi-fungsi untuk membangun aplikasi yang berkaitan dengan *computer vision* [29].

Perancangan sistem secara keseluruhan dilakukan berdasarkan fitur-fitur yang telah ditetapkan pada rencana awal. Implementasi fitur dikerjakan oleh 4

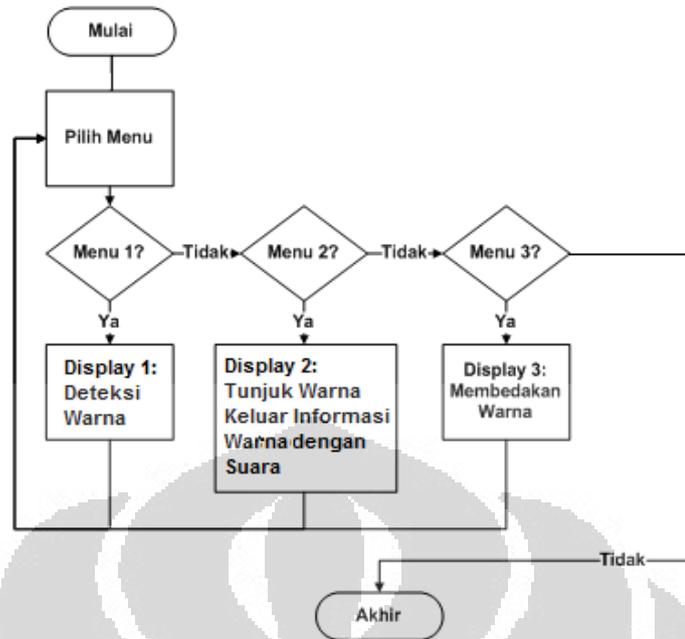
anggota kelompok penelitian dengan tugas masing-masing terdiri dari:

1. Implementasi sistem deteksi jari pada sistem tertanam sehingga pengguna dapat menjalankan perintah dengan mengarahkan kamera ke arah jari yang memberikan instruksi dan mengeksekusi perintah [30].
2. Implementasi sistem konversi suara menjadi perintah menjalankan program dan memberikan informasi warna objek dengan suara [31].
3. Implementasi sistem deteksi warna dengan memberikan informasi berupa teks pada objek dengan warna tertentu [32]
4. Implementasi sistem tes buta warna pada perangkat bergerak serta sistem transformasi warna pada perangkat bergerak dan perangkat tertanam [33].

Selanjutnya, semua sistem untuk masing-masing fitur tersebut akan diintegrasikan dengan antarmuka pengguna. Pengintegrasian dilakukan dengan menggunakan aplikasi *Microsoft Expression Blend* yang akan menghubungkan kode program masing-masing sistem dengan komponen kontrol antarmuka pengguna dengan memasukkan fungsi pemanggil. Untuk memasukkan fungsi pemanggil ini, digunakan *Microsoft Visual Studio* yang menghubungkan dengan kode program sistem masing-masing fitur.

3.5 Rancangan Cara Kerja Sistem

Untuk memvisualisasikan cara kerja sistem bantuan penyandang buta warna, digunakan pemodelan dengan *Unified Modelling Language* (UML). UML adalah pemodelan sistem yang digunakan secara universal dalam dunia teknologi informasi (TI). UML menggunakan notasi-notasi yang terstandar [34] sehingga memudahkan pengembangan proyek TI skala besar yang melibatkan banyak orang, baik pengguna, pelaksana proyek/*programmer*, penganalisa, dan penguji coba.

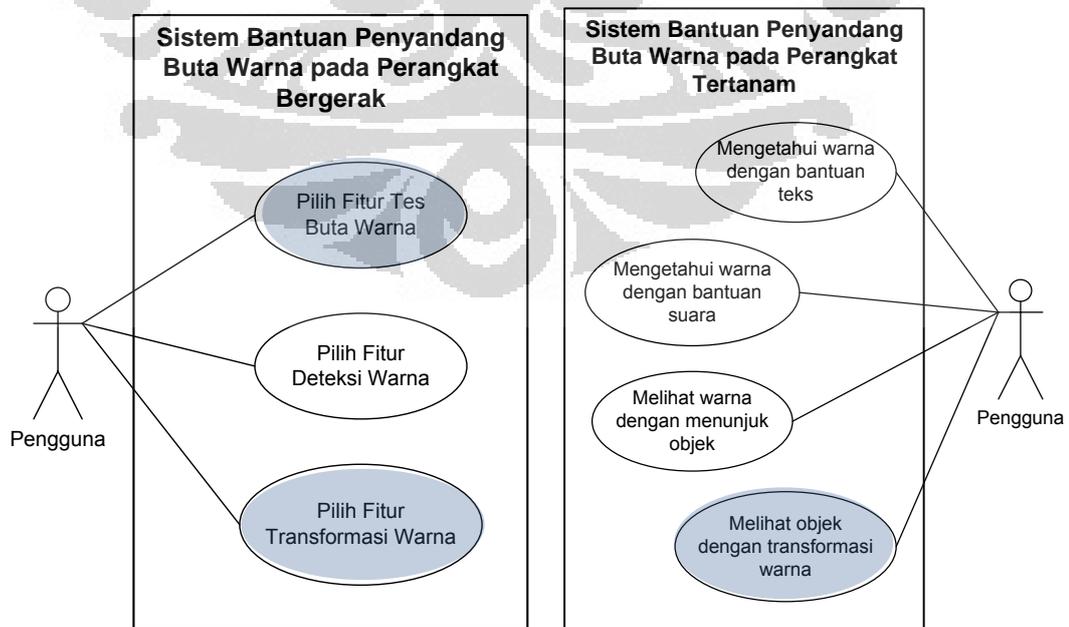


Gambar 3.8. Diagram alir penggunaan sistem bantu penyandang buta warna

Selanjutnya akan dijelaskan mengenai rancangan kerja sistem untuk tiap-tiap jenis UML.

3.5.1 Diagram *Use Case*

Use case diagram digunakan untuk menggambarkan proses interaksi yang terjadi antara pengguna dengan fungsionalitas utama sistem.

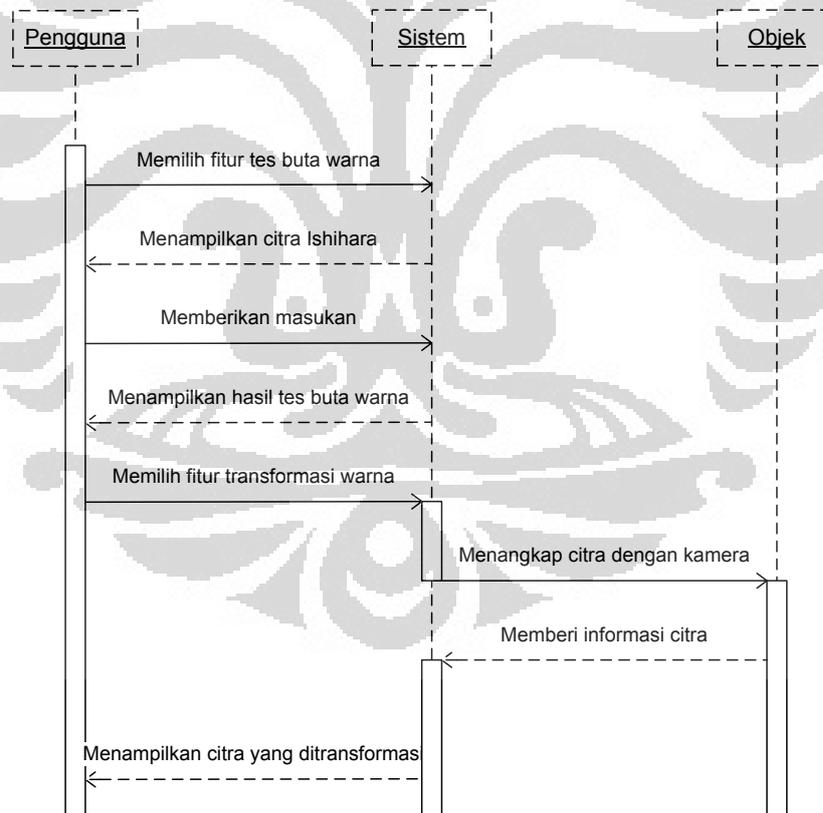


Gambar 3.9. *Use Case* diagram sistem bantuan bagi penyandang buta warna

Use Case diagram menggambarkan interaksi antara pengguna dan sistem untuk mencapai tujuan [35]. Diagram di atas mewakili fungsionalitas dari sistem bantuan penderita buta warna dengan fitur-fitur yang ada pada masing-masing perangkat.

Pengguna dapat memilih menu fitur yang tersedia untuk menjalankan sistem yang diinginkan. Ada 3 fitur utama pada perangkat bergerak, yaitu tes buta warna, deteksi warna, dan transformasi warna. Pada perangkat tertanam, ada 3 fitur utama yang dapat dijalankan, yaitu deteksi warna, mengetahui warna objek tertentu dengan menggunakan jari sebagai penunjuk dan memberikan informasi warna dengan suara, transformasi warna. Program pada perangkat tertanam dijalankan dengan fungsi perintah suara.

3.5.2 Diagram Sequence

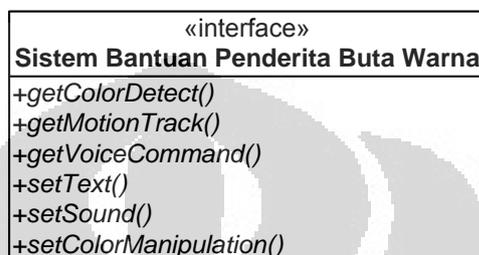


Gambar 3.10. *Sequence* diagram penggunaan fitur pada perangkat bergerak

Sequence diagram digunakan untuk menggambarkan perilaku objek-objek yang terlibat dalam sistem dan pesan antar objek yang terdapat pada *use case* [24].

Ada tiga entitas pada sistem bantuan yang saling berinteraksi, yaitu pengguna, sistem, dan objek. Pengguna akan mengeksekusi program pada sistem, dan sistem akan mengolah data informasi serta melakukan interaksi atau tindakan terhadap objek sesuai perintah pengguna, kemudian sistem memberikan hasil keluaran dari olah masukan kepada pengguna.

3.5.3 Diagram Class



Gambar 3.11. *Class* diagram integrasi antarmuka pengguna dengan sistem

Class diagram digunakan untuk menggambarkan objek-objek dalam sistem dan hubungannya dari objek-objek tersebut [24]. Adapun diagram di atas menggambarkan pengintegrasian sistem pada antarmuka pengguna. Jadi, objek yang terlibat pada sistem diintegrasikan terhadap antarmuka pengguna sehingga semua kelas adalah perintah dari masing-masing objek pada fitur sistem bantuan penderita buta warna.

`ColorDetect` merupakan kelas yang mendefinisikan sistem untuk mengolah citra dan warna-warna objek. `MotionTrack` digunakan untuk melacak pergerakan video yang diambil sekaligus sistem yang dapat digunakan pula untuk fitur deteksi jari pada perangkat *embedded*. `VoiceCommand` digunakan untuk mengatur fungsi berupa perintah suara. Pada akhirnya, semua sistem yang tergabung dapat mengeluarkan dan menampilkan hasil pengolahan data dan informasi pada antarmuka pengguna.

BAB 4

IMPLEMENTASI RANCANGAN PENELITIAN

4.1 Implementasi Antarmuka Pengguna pada *Windows Phone 7*

Desain antarmuka pengguna untuk sebuah sistem yang dirancang sebagai sistem bantu penglihatan, dalam hal ini untuk penyandang buta warna. Segi estetika, kemudahan penggunaan program, serta waktu eksekusi program yang cepat, karakter penglihatan penyandang buta warna, menjadi perhatian pada sistem yang dibuat.

Pada tahap implementasi ada batasan-batasan dalam desain antarmuka, seperti tampilan layar keseluruhan yang berukuran 480x768 piksel dan konsep metro yang diterapkan pada *Windows Phone 7*. Adapun lingkungan implementasi perangkat keras pada *Windows Phone 7*, di antaranya adalah sebagai berikut:

- Tipe:
LG Optimus E900
- Display:
TFT layar sentuh kapasitif, 16 M warna,
480 x 800 piksel, 3,8 inci
- CPU:
Prosesor 1 GHz Scorpion, Adreno 200 GPU,
Qualcomm QSD8650 Snapdragon *chipset*

Berdasarkan spesifikasi yang ada, pengimplementasian sistem bantuan buta warna dengan fungsi utama memproses citra dan warna dengan menggunakan jenis perangkat telepon pintar dapat dilakukan. Untuk langkah awal implementasi, digunakan 2 jenis tipe halaman aplikasi yang tersedia pada *Windows Phone 7 Developer Tools*, yaitu *Windows Phone Page* dan *Windows Phone Panorama Page*. *Windows Phone Page* adalah jenis halaman aplikasi biasa yang hanya terdapat satu halaman tampilan aplikasi, sedangkan *Windows Phone Panorama Page* digunakan untuk memberikan cara yang unik untuk menampilkan kontrol, data dan layanan melalui sebuah kanvas horizontal yang panjang dan melebihi ukuran layar. Alasan lain digunakannya tipe panorama adalah untuk menghemat

penggunaan komponen antarmuka, dan untuk lebih menghemat sumber daya yang disediakan sistem pada tipe panorama, penulis menyiasati menggunakan 1 komponen untuk menampilkan citra dari kamera dan 1 komponen menampilkan citra yang telah diproses oleh sistem.

Berikut merupakan fungsi *timer* yang berjalan saat program dieksekusi atau halaman aplikasi sedang dalam kondisi *load*, dengan tujuan dapat mengatur waktu tiap *frame* yang jalan pada komponen penampil citra.

```
public PanoramaPage()
{
    InitializeComponent();
    InitializeCamera();
    // 100 Milliseconds
    timerdetect.Interval = new TimeSpan(0, 0, 0, 0, 150);
    timerdetect.Tick += new EventHandler(timerdetect_Tick);
    // 100 Milliseconds
    timertransform.Interval = new TimeSpan(0, 0, 0, 0, 150);
    timertransform.Tick += new EventHandler(timertransform_Tick);
}
```

Gambar 4.1. Pendefinisian fungsi *timer*

Pembangunan aplikasi yang efektif dan efisien, difokuskan pada fitur sistem dan daya prosesnya, serta desain antarmuka dan komponen yang digunakan untuk menjembatani antara pengguna dan sistem.

Adapun komponen yang dipakai terdiri dari:

- *Button* sebagai kontrol pengguna.
- *Textbox* untuk memasukkan input dari pengguna.
- *Listbox* sebagai daftar informasi atau keterangan yang memanfaatkan fungsi *slider*.
- *Picture* untuk menempatkan gambar.
- *Imageviewer* untuk menampilkan hasil gambar yang ditangkap oleh kamera dan telah diproses oleh sistem.
- *Cameravisualizer* untuk memunculkan gambar dari kamera secara *realtime*.

Penambahan objek baru pada aplikasi dilakukan dengan jenis mode desain dan memasukan aset kontrol yang tersedia, sehingga diperoleh rancangan antarmuka dengan sistem ”*What You See Is What You Get*”. Akan tetapi dalam pemberian kontrol lebih atau pengaturannya, tetap dibutuhkan pemrograman

bahasa XAML. Animasi atau efek transisi dilakukan dengan menambahkan elemen *storyboard* pada bagian kolom *object and timeline*. Setelah mengimpor animasi yang telah dibuat dengan *frame-by-frame*, maka pada kode XAML antarmuka akan muncul baris kode seperti berikut:

```
<Storyboard x:Name="...">
    <DoubleAnimationUsingKeyFrames      Storyboard.TargetProperty ...
                                       Storyboard.TargetName ...>
        <EasingDoubleKeyFrame KeyTime ... Value ...>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
```

Gambar 4.2. Kode program bahasa XAML untuk penambahan animasi

Bagian kosong pada baris kode di atas adalah parameter untuk masing-masing elemen properti. Selanjutnya, ketika tahap penambahan komponen atau aset kontrol selesai, maka hal yang paling pokok dalam integrasi sistem aplikasi yang dapat berjalan adalah menambahkan *event* atau kondisi saat program harus menjalankan fungsi yang ada. Setiap penambahan aset harus diberikan pula nama masing-masing ketika menambahkan *event*. Sistem dapat mengenali identitas masing-masing aset pada *method* untuk kondisi eksekusi program.

Melengkapi desain antarmuka pengguna, ditambahkan pula informasi atau fungsi bantuan untuk membantu penggunaan fitur aplikasi. Implementasi fungsi bantuan dirancang dengan memunculkan jendela *pop-up* yang kemudian akan mengarahkan pengguna ke halaman aplikasi selanjutnya.

```
private void help1_MouseLeftButtonUp(object sender,
                                     System.Windows.Input.MouseButtonEventArgs e)
{
    //memunculkan jendela bantuan dengan animasi transisi
    trans4.Begin();

    //navigasi ke halaman aplikasi selanjutnya
    Uri targetUri = new Uri
    ("/PanoramaPage.xaml", System.UriKind.Relative);
    NavigationService.Navigate(targetUri);
}
```

Gambar 4.3. Kode program menampilkan info bantuan dan navigasi halaman

4.2 Implementasi Sistem Tes Buta Warna dengan Metode Ishihara

Secara umum, algoritma sistem tes buta warna telah dijelaskan pada bab 3, yaitu bagian perancangan. Jadi, instruksi program yang berjalan telah disesuaikan dengan alur pengujian buta warna dengan metode Ishihara pada umumnya. Berdasarkan perancangannya, hasil keluaran yang akan diperoleh dari implementasi sistem tes buta warna ini adalah 5 jenis tipe karakter penglihatan terhadap warna, yaitu penglihatan normal, buta warna parsial, *protan* kuat (kelemahan penglihatan warna merah), *deutan* kuat (kelemahan penglihatan warna hijau), dan buta warna total.

Sistem tes buta warna ini terdiri dari 21 citra Ishihara dan tiap pergantian citra tidak berpindah halaman aplikasi. Semuanya dikembangkan dalam satu halaman aplikasi yang sama sehingga dapat menghemat sumber daya. Untuk mengatasi meminimalisasian tersebut, semua aset diletakkan pada halaman yang sama, namun digunakan fungsi *hidden* dengan mengubah nilai *opacity* objek menjadi 0. Apabila aset siap dimunculkan, maka *opacity* diubah kembali menjadi 100. Adapun untuk pemrograman sistem tes buta warna ini sendiri dapat direpresentasikan dengan *pseudocode* berikut.

```
int x = 0;
if (TextBox1.Text == "2") {x = x + 1;}
if (TextBox2.Text == "3") {x = x + 1;}
:
:
:
if (TextBox21.Text == "5") {x = x + 1;}

if (x > 17)
{ Hasil.Content = "Berpenglihatan Normal"; }
else
if (x < 4)
{ Hasil.Content = "Buta Warna Total"; }
else
if (x > 3 && x < 18 && TextBox8.Text == "2"
    && TextBox10.Text == "3"
    && TextBox11.Text == "4"
    && TextBox16.Text == "9")
{ Hasil.Content = "Deutan Kuat"; }
```

```

else
if (x > 3 && x < 18 && TextBox8.Text == "6"
    && TextBox10.Text == "5"
    && TextBox11.Text == "2"
    && TextBox16.Text == "6")
{ Hasil.Content = "Protan Kuat"; }
else
{ Hasil.Content = "Buta Warna Parsial"; }

```

Gambar 4.4. Kode program untuk menentukan jenis buta warna

Kode semu di atas prinsipnya adalah sama dengan kode program yang sesungguhnya. Terdapat 21 *textbox* yang mewakili tiap-tiap citra Ishihara dan kondisi-kondisi tertentulah yang akan menjadi parameter penentuan hasil tes. Selain itu, untuk mengurangi tingkat kesalahan dibuat sebuah nilai toleransi, seperti apabila seseorang menjawab salah, masih ada kemungkinan hasil tes buta warna pengguna adalah berpenglihatan normal, namun dengan syarat tidak lebih dari 4 jawaban yang salah. Hal tersebut dilakukan untuk meminimalisasi kesalahan yang dilakukan tidak sengaja dalam penggunaan aplikasi oleh pengguna.

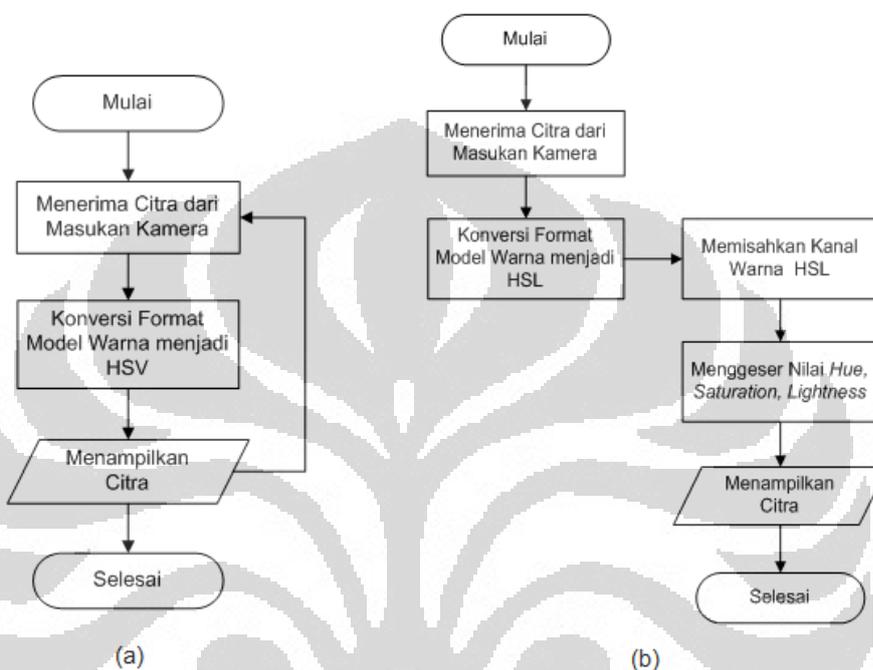


Gambar 4.5. Hasil implementasi sistem tes buta warna pada *Windows Phone 7*

4.3 Implementasi Transformasi Warna

Pengembangan sistem transformasi warna pada citra yang diimplementasikan pada penelitian ini menggunakan dua metode berbeda dengan tujuan yang sama. Hal tersebut dikarenakan pengembangan kedua perangkat memiliki fungsionalitas dan keterbatasan yang berbeda pula. Akan tetapi,

algoritma untuk masing-masing pengembangan sistem tersebut sama-sama menghasilkan citra yang lebih kontras agar penderita buta warna dapat menangkap objek dengan warna yang sebenarnya berbeda, tetapi tampak serupa bagi penglihatan buta warna. Berikut merupakan algoritma masing-masing sistem transformasi warna untuk tiap perangkat:



Gambar 4.6. Algoritma transformasi warna pada (a) perangkat tertanam, dan (b) perangkat bergerak

Pemrograman modul transformasi warna memanfaatkan aplikasi produk Microsoft, yaitu *Visual Studio 2010* yang memiliki fungsionalitas lengkap untuk dapat mengembangkan sistem modul ini pada kedua perangkat, yaitu untuk perangkat Windows Phone 7 dan perangkat tertanam. Untuk lisensi, semua pengembangan yang dilakukan dengan produk Microsoft didapatkan dengan lisensi akademik dari *Microsoft Software Developer Network-Acadamic Alliance (MSDN-AA)*.

4.3.1 Pengembangan untuk Perangkat Tertanam

Pengaplikasian transformasi warna pada perangkat tertanam dilakukan dengan metode yang berbeda dengan yang diterapkan pada perangkat *Windows Phone 7*. Hal tersebut dilakukan setelah pengujian metode lain yang masih belum

dapat menghasilkan pengkonversian warna yang akurat. Sebelum menjelaskan tahap implementasi, berikut merupakan spesifikasi lingkungan implementasi pada perangkat tertanam:

Divais : eBox-3310

Platform : *.NET Framework 3.5*

Sistem Operasi : *Windows Embedded Standard 2009*

Pengembangan sistem bantuan bagi penyandang buta warna ini terdiri beberapa modul yang dikerjakan terpisah. Proses implementasi dan pengembangan modul transformasi warna digunakan lingkungan pengembangan terintegrasi (*Integrated Development Environment, IDE*) sebagai sarana melakukan pengembangan program yang efektif dan efisien. Selain itu, tiap modul yang dibuat harus memiliki dukungan penuh terhadap divais yang akan digunakan. Dalam modul transformasi warna, target *framework* yang digunakan disesuaikan dengan *framework* pada divais yaitu *.NET Framework 3.5*.

Pengadaptasian *platform* pada divais dengan menggunakan *framework* tersebut disebabkan karena pemrograman yang dilakukan adalah menggunakan bahasa C# dengan tambahan *library* EmguCV. EmguCV menggunakan *Windows Communication Foundation (WCF)*, sebuah *Application Programming Interface (API)* pada *.NET Framework* untuk menghubungkan integrasi aplikasi berbasis layanan [36] sehingga dalam implementasi program pada *Visual Studio 2010* dengan menambahkan *library* EmguCV, mengharuskan dijalankan di atas *platform .NET Framework 3.5*.

Berdasarkan literatur dan penelitian yang dilakukan, untuk tahap awal memulai suatu pengolahan citra, dilakukan konversi format citra dari format RGB ke dalam format HSV. Hal ini dilakukan untuk mempermudah perhitungan olah warna. Permodelan dalam format warna HSV mengubah cara penentuan warna menjadi klasifikasi yang dapat dipisahkan, seperti penentuan warna pada kanal *hue*, penentuan kejenuhan warna pada kanal *saturation*, dan penentuan terang-gelap pada kanal *value*.

Pengkonversian warna tersebut, tidak akan secara otomatis menjadikan warna yang sama dengan format yang berbeda. Warna yang tampak akan turut

berubah sesuai dengan pergeseran yang direpresentasikan pada persamaan berikut [7]:

$$h = \begin{cases} 0^0 & \text{jika } \max = \min \\ 60^0 \times \frac{g-b}{\max-\min} + 0^0, & \text{jika } \max = r \text{ dan } g \geq b \\ 60^0 \times \frac{g-b}{\max-\min} + 360^0, & \text{jika } \max = r \text{ dan } g < b \\ 60^0 \times \frac{b-r}{\max-\min} + 120^0, & \text{jika } \max = g \\ 60^0 \times \frac{r-g}{\max-\min} + 240^0, & \text{jika } \max = b \end{cases} \quad (4.1)$$

$$s = \begin{cases} 0, & \text{jika } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \end{cases} \quad (4.2)$$

$$v = \max \quad (4.3)$$

Hue (h) secara umum dipersepsikan dengan batasan antara 0 hingga 360, sedangkan *saturation* (s) dan *value* (v) di antara 0 dan 1. Hasil analisis untuk definisi persamaan di atas, apabila warna utama adalah merah ($\max = r$), h akan berada di rentang (0, 60) dan (300, 360). Apabila warna utama adalah hijau atau biru, dengan kata lain saat kondisi $\max = g$ atau $\max = b$, maka variasi rentang warna masing-masing adalah 60 sampai 120, dan 180 sampai 300, dengan nilai maksimum tiap rentang adalah 120 dan 240.

Untuk mengimplementasikan metode pengkonversian di atas, perhitungan atau pemrograman matematis tidak lagi dilakukan. Dengan mengintegrasikan *library* EmguCV, algoritma untuk mengkonversikan format warna sudah tersedia pada OpenCV dengan EmguCV yang merupakan *wrapper*-nya. Akan tetapi, sebelum memanggil fungsi konversi warna dari RGB ke HSV, pada pembuatan program di *Visual Studio* perlu ditambahkan terlebih dahulu referensi EmguCV. Penambahan referensi tersebut membutuhkan instalasi EmguCV dan memerlukan dokumen seperti bundel *binary* OpenCV 2.0 dan ekstraksi *windows installer* untuk EmguCV. Berkas *project* hasil ekstraksi tadi, yaitu `Emgu.CV.Example.sln` di *build solution*. Pemrograman dapat dilakukan pada *file project* tersebut yang telah berhasil diimpor *library* EmguCV pada bagian *reference*.

Setelah tahap di atas, pembangunan sistem transformasi warna dapat diimplementasikan pada perangkat eBox.

```

namespace ChromphoreIntegrate
{
    class ColorIntensityIncrease
    {
        public bool terbuka=false;
        Capture capture;

        public ColorIntensityIncrease()
        {
            capture = new Capture();
        }

        public void GetColor()
        {
            Image<Bgr, Byte> capturedimage = capture.QuerySmallFrame();
            Image<Hsv, Byte> sipimage = capturedimage.Convert<Hsv, Byte>();

            CvInvoke.cvNamedWindow("hsv");
            CvInvoke.cvShowImage("hsv", sipimage);
            terbuka = true;
        }

        public void stopView()
        {
            CvInvoke.cvDestroyWindow("hsv");
            capture.Dispose();
            terbuka = false;
        }
    }
}

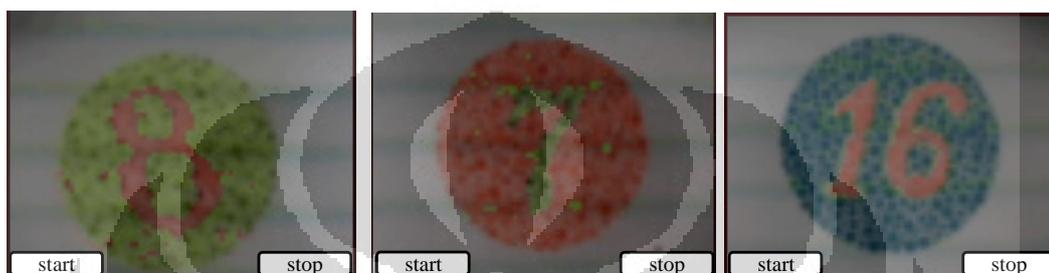
```

Gambar 4.7. Kode program untuk kelas ColorIntensityIncrease

Kode program di atas adalah kelas ColorIntensityIncrease yang berisi program konversi warna dari RGB menjadi HSV. Adapun bagian tersebut dieksekusikan dengan metode GetColor().

Berdasarkan percobaan saat implementasi, tahap awal yang pernah dilakukan adalah menambahkan proses pergeseran nilai *hue* dan *saturation*. Hal tersebut dilakukan setelah memisahkan kanal (*channel*), kemudian menentukan rentang warna yang perlu digeser dengan nilai baru. Dengan demikian, warna yang telah ditransformasikan tidak berbeda terlalu jauh dengan warna yang sebenarnya. Akan tetapi, setelah tahap implementasi dan melalui pengujian sistem, algoritma tersebut sebenarnya memiliki hasil yang bagus, tetapi sangat terpengaruh oleh faktor seperti kualitas kamera dan pencahayaan. Pada bahasa C#

dalam *Visual Studio*, kisaran nilai *hue* berada di antara 0 – 255 padahal umumnya adalah 0 – 360. Perangkat kamera sebagai prototipe pengganti visualisasi yang ditangkap oleh *Head Mounted Display (HMD)* pada pengujian pun kurang memadai. Selain itu, faktor pencahayaan ruangan sangat mempengaruhi tingkat akurasi. Gambar 4.8 menunjukkan hasil transformasi warna menggunakan metode pergeseran nilai *hue* dan *saturation* dengan batasan perataan warna merah, kuning, hijau, dan biru.



Gambar 4.8. Transformasi warna dengan pergeseran nilai *hue* dan *saturation*

Karena hasil transformasi warna kurang maksimal, dalam implementasi pada perangkat tertanam, algoritma program hanya sampai pada pengkonversian format citra. Hasil konversi tersebut telah menghasilkan peningkatan intensitas warna citra sehingga menjadi lebih kontras dan dalam rentang warna yang dapat ditoleransi oleh penglihatan buta warna. Gambar 4.9 merupakan hasil metode sistem transformasi warna yang akhirnya diimplementasikan pada perangkat tertanam.



Gambar 4.9. Transformasi warna pada perangkat tertanam

4.3.2 Pengembangan untuk Perangkat Bergerak

Pemrograman untuk perangkat bergerak pada platform *Windows Phone 7* memiliki ruang lingkup pengembangan yang lebih sederhana untuk proses olah citra. Tidak seperti pada implementasi pada perangkat tertanam, eksekusi fungsi sistem pengolahan citra dipermudah dengan penambahan *library* EmguCV. Pada pengembangan untuk perangkat bergerak, walaupun sama-sama diprogram menggunakan *Visual Studio*, jenis proyek yang dipilih saat awal pembuatan dokumen proyek baru adalah jenis pemrograman aplikasi untuk *Windows Phone 7* yang berjalan di atas *platform .NET Framework 4*. Selain itu, keterbatasan pengembangan aplikasi untuk *Windows Phone 7* pada penelitian ini adalah tidak tersedianya fungsi emulator untuk kamera. Bahkan penambahan referensi atau *library* pihak ketiga seperti EmguCV tidak bisa dilakukan karena tidak didukung penuh untuk dapat dijalankan di atas *platform Windows Phone 7*. Keterbatasan yang dipaparkan di atas membuat hasil keluaran berbeda dengan hasil keluaran pada perangkat tertanam.

Sebagai pengganti *library* EmguCV, digunakan referensi *Microsoft.Phone.Media.Extended*. Seperti penggunaan EmguCV pada *Visual Studio*, penggunaan referensi *Microsoft.Phone.Media.Extended* juga membutuhkan tahap instalasi agar dapat digunakan. Tahap-tahap instalasi agar dapat membangun solusi dari *file project* adalah terdiri dari:

1. Duplikat *Microsoft.Phone.Media.Extended.dll* ke *C:\Program Files\Reference Assemblies\Microsoft\Framework\Silverlight\v4.0\Profile\WindowsPhone*
2. Tambahkan *Microsoft.Phone.Media.Extended* pada *FrameworkList.xml*.
3. Pada *command prompt* *Visual Studio 2010*, masukkan perintah “*SN-Vr Microsoft.Phone.Media.Extended.dll*”.

Setelah tahap di atas, *library* sudah dapat digunakan. Dalam implementasinya, fitur transformasi warna yang juga merupakan suatu sistem pengolahan citra dari kamera tidak dilakukan sepenuhnya secara *real-time* terhadap citra yang tertangkap kamera. Proses transformasi warna dimulai dengan melakukan pengambilan gambar objek yang diinginkan pengguna untuk kemudian diolah warnanya. Adapun kode program untuk perintah transformasi

warna berada dalam *method* GetColor2().

```

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

        If (HslConvertedPxl.H >= 0 && HslConvertedPxl.H <= 10 &&
            HslConvertedPxl.L >= 0.5 && HslConvertedPxl.L <= 0.6)
        {
            writeableBmp.SetPixel(w, h, 255, 50, 50);
        }

        if (HslConvertedPxl.H >= 0 && HslConvertedPxl.H <= 9 &&
            HslConvertedPxl.L >= 0.2 && HslConvertedPxl.L <= 0.4)
        {
            writeableBmp.SetPixel(w, h, 150, 0, 0);
        }

        if (HslConvertedPxl.H >= 25 && HslConvertedPxl.H <= 35 &&
            HslConvertedPxl.L >= 0.65 && HslConvertedPxl.L <= 0.85)
        {
            writeableBmp.SetPixel(w, h, 250, 214, 165);
        }

        if (HslConvertedPxl.H >= 25 && HslConvertedPxl.H <= 35 &&
            HslConvertedPxl.L >= 0.3 && HslConvertedPxl.L <= 0.45)
        {
            writeableBmp.SetPixel(w, h, 123, 63, 0);
        }

        if (HslConvertedPxl.H >= 55 && HslConvertedPxl.H <= 60 &&
            HslConvertedPxl.L >= 0.75 && HslConvertedPxl.L <= 0.85)
        {
            writeableBmp.SetPixel(w, h, 255, 255, 0);
        }

        if (HslConvertedPxl.H >= 70 && HslConvertedPxl.H <= 80 &&
            HslConvertedPxl.L >= 0.30 && HslConvertedPxl.L <= 0.50)
        {
            writeableBmp.SetPixel(w, h, 141, 188, 0);
        }

        if (HslConvertedPxl.H >= 90 && HslConvertedPxl.H <= 135 &&
            HslConvertedPxl.L >= 0.4 && HslConvertedPxl.L <= 0.6)
        {
            writeableBmp.SetPixel(w, h, 92, 135, 49);
        }

        if (HslConvertedPxl.H >= 200 && HslConvertedPxl.H <= 220 &&
            HslConvertedPxl.L >= 0.45 && HslConvertedPxl.L <= 0.7)
        {
            writeableBmp.SetPixel(w, h, 0, 127, 255);
        }

        if (HslConvertedPxl.H >= 225 && HslConvertedPxl.H <= 240 &&
            HslConvertedPxl.L >= 0.3 && HslConvertedPxl.L <= 0.5)
        {
            writeableBmp.SetPixel(w, h, 0, 0, 255);
        }
    }
}

```

```

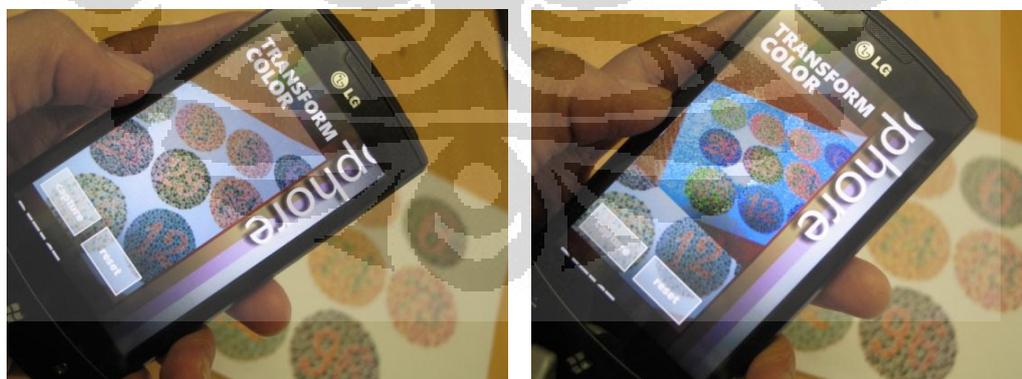
    }
    if (HslConvertedPx1.H >= 260 && HslConvertedPx1.H <= 275 &&
        HslConvertedPx1.L >= 0.4 && HslConvertedPx1.L <= 0.65)
    {
        writeableBmp.SetPixel(w, h, 143, 0, 255);
    }
    if (HslConvertedPx1.H >= 0 && HslConvertedPx1.H <= 360 &&
        HslConvertedPx1.L >= 0.70 && HslConvertedPx1.L <= 1)
    {
        writeableBmp.SetPixel(w, h, 255, 255, 255);
    }
}
}
}

```

Gambar 4.10. Kode program untuk *method* `GetColor2()`

Pergeseran yang dilakukan untuk masing-masing nilai *hue* dan *lightness*, dilakukan dengan beberapa kali pengujian. Berdasarkan percobaan terhadap persepsi warna untuk pergeseran nilai *hue* dan *lightness* yang diterapkan pada program tersebut dapat menaikkan kontras warna. Dengan demikian, warna-warna lemah atau warna-warna ambigu dapat dihilangkan atau diminimalisasi.

Pada pemrograman di atas, pergeseran dari hasil transformasi dilakukan dengan mengubah rentang warna dengan format HSL menjadi warna dengan format RGB.



Gambar 4.11. Hasil implementasi transformasi warna pada *Windows Phone 7*

BAB 5

ANALISIS PENGUJIAN IMPLEMENTASI SISTEM

5.1 Analisis Pengujian pada Perangkat Bergerak (*Windows Phone 7*)

Pengujian sistem bertujuan untuk membuktikan hipotesa dan mengetahui tanggapan pengguna terhadap sistem bantuan buta warna pada perangkat bergerak. Pengujian terdiri dari penilaian pengguna terhadap antarmuka, pengaruh kamera terhadap kesesuaian perubahan warna asli, tingkat kejelasan citra dengan transformasi warna, tingkat akurasi tes buta warna, dan pengujian sistem secara keseluruhan pada perangkat bergerak.

5.1.1 Penilaian Pengguna terhadap Antarmuka

Dalam pengujian ini, diambil sampel sebanyak 10 pengguna untuk memberikan penilaian terhadap tampilan antarmuka serta kemudahan penggunaan dan navigasi sistem bantuan penyandang buta warna pada perangkat bergerak. Setelah pengguna selesai menggunakan keseluruhan fitur dan melihat tampilan antarmuka, pengguna diminta mengisi *form* yang terdiri dari tingkat penilaian terhadap pernyataan dengan parameter sangat tidak setuju, tidak setuju, agak setuju, setuju, sangat setuju, masing-masing bernilai 1, 2, 3, 4, dan 5.

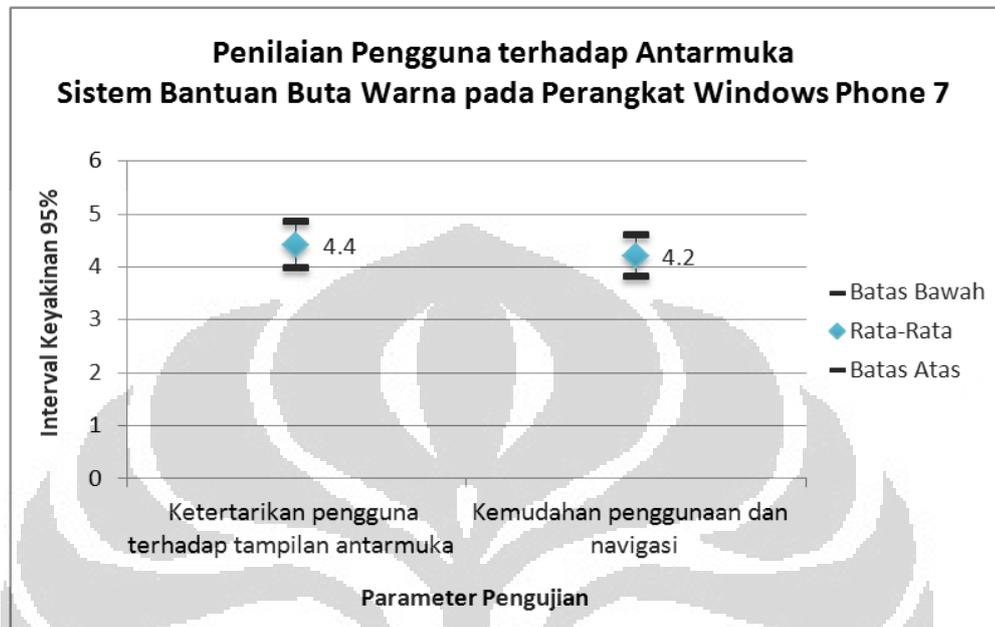
Berdasarkan pengujian yang telah dilakukan terhadap 10 pengguna, diperoleh nilai tingkat kepercayaan pengujian 95% (*95% confidence interval*) dari hasil perhitungan dengan persamaan berikut:

$$\text{Rata-rata penilaian penguji} \pm \left(1.96 \times \frac{\text{Standar deviasi}}{\sqrt{\text{Jumlah penguji}}} \right) \quad (5.1)$$

Tabel 5.1. Tabel data hasil pengujian antarmuka pengguna

Parameter Pengujian	Hasil
Ketertarikan pengguna terhadap tampilan antarmuka	4,4 ± 0,43
Kemudahan penggunaan dan navigasi	4,2 ± 0,39

Berdasarkan data yang diperoleh dalam pengujian tersebut, hasil rata-rata tingkat kepercayaan pengujian 95% dapat direpresentasikan ke dalam bentuk grafik di bawah ini.



Gambar 5.1. Grafik penilaian pengguna terhadap antarmuka

Pada grafik Gambar 5.1, terlihat bahwa rata-rata pengguna menganggap tampilan antarmuka menarik dan antarmuka mendukung kemudahan bagi pengguna dalam navigasi dan pemakaian sistem bantuan ini. Hal tersebut dapat terlihat dari pengujian yang dilakukan terhadap dua parameter mengenai antarmuka yang masing-masing memiliki nilai 4,4 dan 4,2 dalam interval keyakinan 95% dari skala maksimal 5. Dengan demikian, kepuasan pengguna terhadap pengimplementasian antarmuka pada sistem ini berada di atas nilai tengah interval keyakinan 95%.

Berdasarkan hasil perhitungan data di atas, faktor besarnya ukuran *file* XAP juga mempengaruhi kepuasan pengguna terhadap pemakaian aplikasi. Adapun besar *file* XAP untuk aplikasi sistem bantuan ini adalah 1.389 KB. Maksimum paket XAP yang diizinkan tidak boleh lebih dari 400 MB. Hal ini menyebabkan sistem berjalan dengan ringan dan dapat menghasilkan gerakan-gerakan transisi atau animasi yang halus, yang tentunya mendukung penilaian lebih baik dari antarmuka pengguna.

5.1.2 Pengaruh Kamera terhadap Kesesuaian Perubahan Warna

Sistem transformasi warna pada perangkat bergerak menghasilkan keluaran dengan warna asli diubah menjadi lebih kontras atau dengan kata lain warna berubah menjadi warna lain yang lebih tinggi intensitasnya, namun masih dalam rentang yang sama. Pada penggunaan sistem ini, warna citra yang diolah adalah warna pada citra hasil pengambilan gambar oleh kamera perangkat. Oleh sebab itu, pengujian ini bertujuan untuk menganalisis pengaruh kamera terhadap perubahan warna objek sesungguhnya dan warna yang ditangkap oleh kamera.

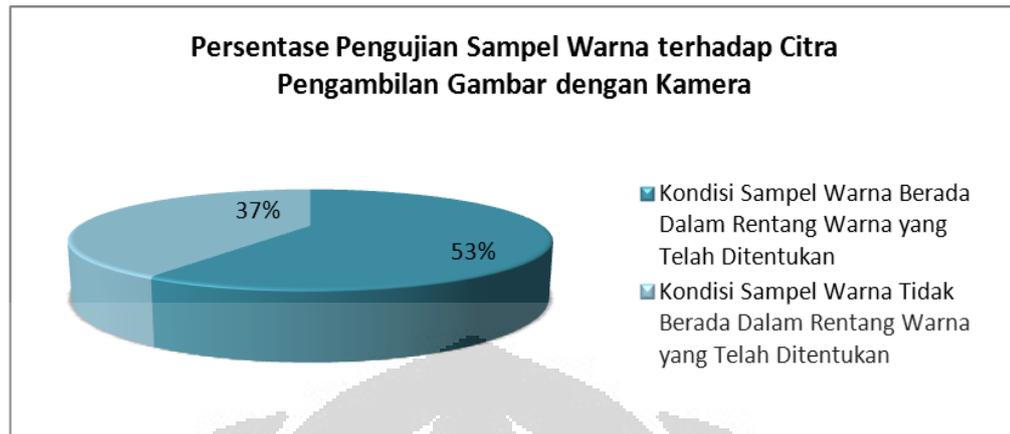
Metode pengujian ini dilakukan dengan mengambil sampel warna yang berada dalam rentang warna yang ditentukan di awal (pada pemrograman), kemudian membandingkan warna tersebut dengan warna pada citra hasil pengambilan gambar oleh kamera *Windows Phone 7* sebanyak 10 kali pengambilan gambar. Adapun perbandingan warna citra asli dengan warna citra hasil tangkapan kamera menggunakan bantuan fitur *eyedropper* dan model palet warna pada perangkat lunak *CorelDraw*. Kedua citra dibandingkan dengan melihat nilai *channel* warna dan kemudian dicocokkan dengan rentang warna pada program. Berikut merupakan tabel data hasil pengujian tersebut.

Tabel 5.2. Tabel data hasil pengujian pengaruh kamera terhadap warna citra

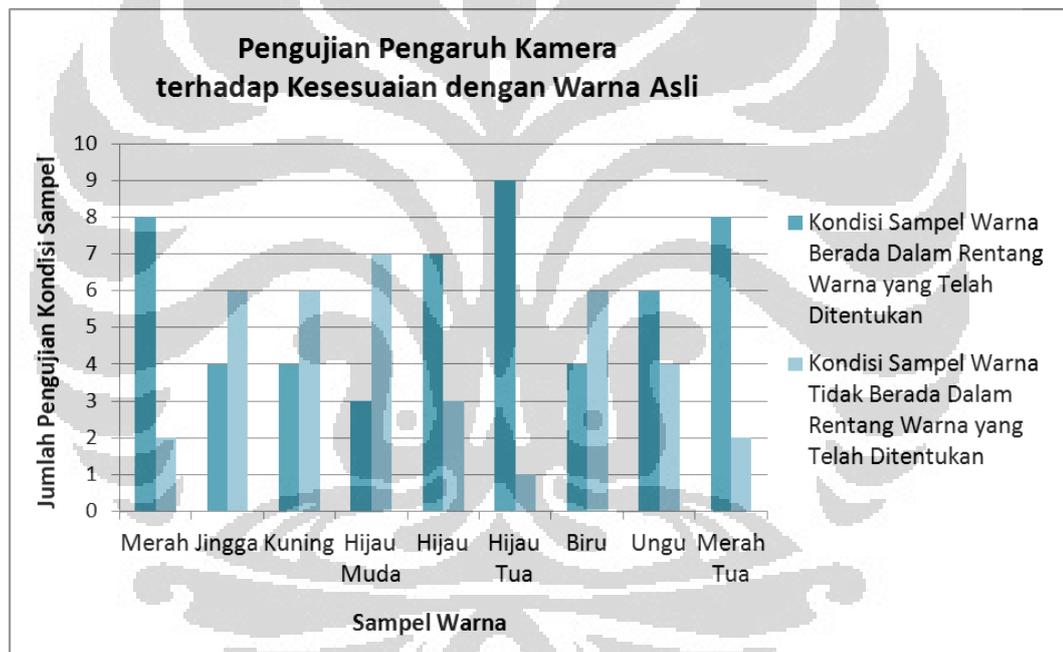
Warna Asli	Jumlah Pengujian	
	Kondisi Sampel Warna Berada Dalam Rentang Warna yang Telah Ditentukan	Kondisi Sampel Warna Tidak Berada Dalam Rentang Warna yang Telah Ditentukan
Merah	8	2
Jingga	4	6
Kuning	4	6
Hijau Muda	3	7
Hijau	7	3
Hijau Tua	9	1
Biru	4	6
Ungu	6	4
Merah Tua	8	2

Melalui pengolahan data, hasil pengujian pada Tabel 5.2 dapat direpresentasikan ke dalam bentuk diagram yang dapat menunjukkan perbandingan persentase pengujian sampel warna terhadap citra pengambilan

gambar dengan kamera pada perangkat *Windows Phone 7*.



Gambar 5.2. Diagram persentase pengujian sampel warna dengan kamera



Gambar 5.3. Grafik perbandingan kondisi sampel warna asli dengan hasil tangkapan kamera

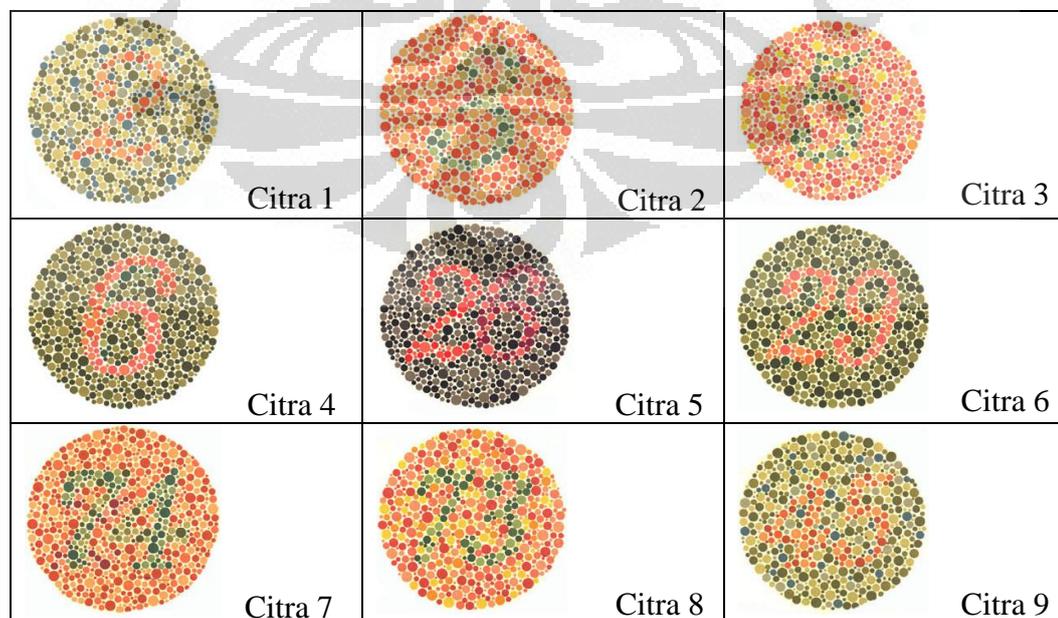
Kalkulasi hasil pengujian di atas menunjukkan bahwa kamera akan sangat mempengaruhi hasil olah sistem. Hal tersebut dibuktikan dengan percobaan pengambilan gambar dengan sebanyak 10 kali pengulangan dan membandingkan dengan warna citra yang asli. Dari grafik terlihat bahwa dalam 10 kali percobaan tersebut tidak menghasilkan pola yang baik dalam kondisi sampel warna yang diambil oleh kamera. Ada saja saat ketika warna berubah menjadi berada di luar

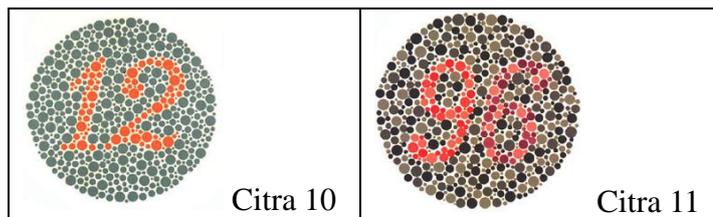
rentang warna yang seharusnya. Berdasarkan hal tersebut, penulis menyimpulkan bahwa kamera yang turut menjadi faktor berubahnya rentang warna pada citra asli tidak hanya menjadi faktor tunggal, karena ada faktor lain seperti pencahayaan dan posisi pengambilan gambar. Hipotesa ini didukung dengan jumlah pengujian yang membentuk pola tidak teratur yang disebabkan adanya faktor pengganggu untuk saat terjadinya kondisi perubahan rentang warna.

5.1.3 Tingkat Kejelasan Citra dengan Transformasi Warna

Inti pengujian ini adalah menentukan apakah sistem transformasi warna yang diimplementasikan menghasilkan keluaran yang diharapkan. Pengguna diharapkan dapat dengan jelas membedakan objek yang memiliki warna hampir sama atau bahkan terlihat sama pada penglihatan buta warna.

Metode yang dilakukan dalam pengujian ini adalah dengan mengambil sampel 10 pengguna untuk menguji persepsi kejelasan citra yang terlihat setelah dilakukan transformasi warna. Untuk mendapatkan data yang lebih valid, objek atau citra yang diuji adalah citra yang sama. Dalam pengujian ini, objek yang digunakan untuk dilakukannya transformasi warna adalah citra Ishihara. Adapun citra Ishihara yang digunakan adalah 11 citra dengan tipe desain yang berbeda. Berikut adalah citra Ishihara yang digunakan dalam pengujian.





Gambar 5.4. Jenis citra Ishihara yang digunakan dalam pengujian

Tabel 5.3. Interval keyakinan 95% hasil uji kejelasan citra

Citra Ishihara	Hasil	
	Tanpa Sistem Bantuan	Dengan Sistem Bantuan
Citra 1	$3 \pm 0,773$	$3,7 \pm 0,510$
Citra 2	$4,3 \pm 0,926$	$4,1 \pm 0,457$
Citra 3	$4,3 \pm 0,776$	$3,9 \pm 0,457$
Citra 4	$4,5 \pm 0,669$	$4,7 \pm 0,299$
Citra 5	$4,1 \pm 0,742$	$3,7 \pm 0,510$
Citra 6	$4,8 \pm 0,392$	$4,8 \pm 0,261$
Citra 7	$4,1 \pm 0,898$	$4,6 \pm 0,320$
Citra 8	$4 \pm 1,012$	$3,6 \pm 0,699$
Citra 9	$3,8 \pm 0,960$	$3,3 \pm 0,823$
Citra 10	5 ± 0	5 ± 0
Citra 11	$4,2 \pm 1,033$	$3,8 \pm 0,422$
Rata-rata	$4,19 \pm 0,744$	$4,11 \pm 0,432$

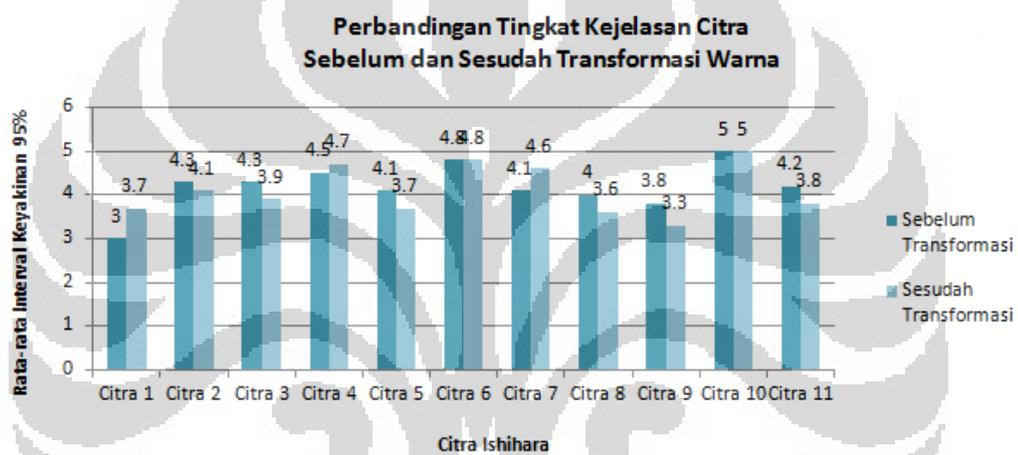
Parameter pengujian kejelasan citra Ishihara terdiri dari kondisi citra yang tidak terlihat atau salah terbaca, sulit terlihat, cukup terlihat, jelas terlihat, dan sangat jelas terlihat.

Dalam hasil pengujian penglihatan citra tanpa bantuan transformasi warna terdapat 2 responden yang mengalami kondisi tidak dapat atau sulit melihat angka pada citra Ishihara yang diujikan, hal tersebut disebabkan 2 responden dari 10 penguji adalah penderita buta warna parsial. Setelah penggunaan sistem transformasi warna pada perangkat, tidak ada kondisi angka yang tidak terlihat dan mengurangi persentase munculnya kondisi kesulitan penglihatan angka pada citra Ishihara.

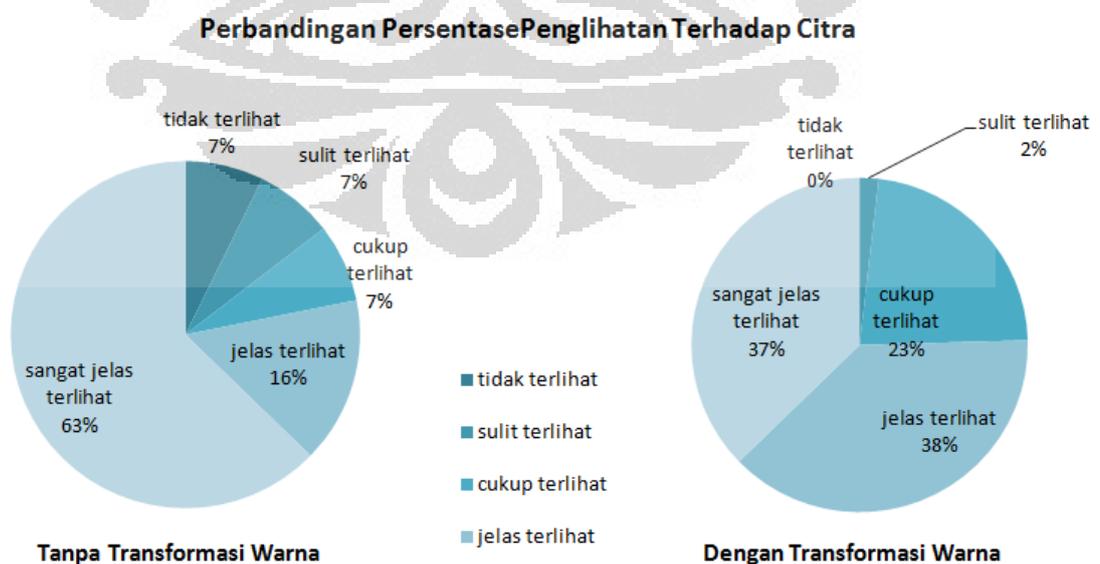
Pada Tabel 5.3, nilai rata-rata interval keyakinan pengujian saat tanpa menggunakan bantuan sistem transformasi warna hampir sama dengan saat menggunakan transformasi warna. Tanpa penggunaan sistem transformasi warna,

nilai rata-rata interval kepercayaan pengujian terhadap 10 responden adalah $4,19 \pm 0,744$, sedangkan dengan penggunaan sistem transformasi warna adalah $4,11 \pm 0,432$. Nilai hasil pengujian tanpa menggunakan transformasi warna lebih besar sedikit dibanding dengan menggunakan transformasi warna disebabkan sampel penguji. Dominan penguji adalah orang berpenglihatan normal dengan kondisi dapat melihat jelas citra, walaupun tanpa penggunaan sistem transformasi warna.

Berdasarkan data yang diperoleh saat pengujian, dengan perhitungan nilai *confidence interval 95%*, didapat grafik perbandingan tingkat kejelasan citra sebelum dan sesudah transformasi warna, sebagai berikut.



Gambar 5.5. Grafik perbandingan tingkat kejelasan citra



Gambar 5.6. Grafik persentase kejelasan penglihatan terhadap citra

Data pengujian di atas menunjukkan bahwa dengan penggunaan sistem transformasi warna yang diimplementasikan pada perangkat *Windows Phone 7* dapat menghilangkan kemungkinan kondisi tidak terlihatnya objek dengan warna ambigu dan memperkecil persentase kondisi sulit terlihat.

Seperti yang telah dijelaskan pada bagian Bab 3, tentang teori Dalton, peningkatan intensitas warna asli menjadi lebih kontras dapat menaikkan pula nilai toleransi sel kerucut pada mata manusia. Ketika seseorang yang mengalami kelemahan penglihatan terhadap warna, warna ambigu yang telah dikontraskan ke arah warna primer aslinya masih dapat terlihat dalam rentang warna yang tidak terlalu jauh dengan warna pada citra sebenarnya. Namun, dalam pengujian terkait pengaruh penggunaan sistem transformasi warna terhadap kejelasan citra. Faktor-faktor lain dapat menurunkan kualitas hasil pengujian, di antaranya adalah kualitas warna tinta dalam penyetakan citra Ishihara, cahaya ruangan pengujian, serta kualitas kamera perangkat yang analisisnya telah dijelaskan pada bagian pengujian pengaruh kamera terhadap kesesuaian warna.

5.1.4 Tingkat Akurasi Tes Buta Warna

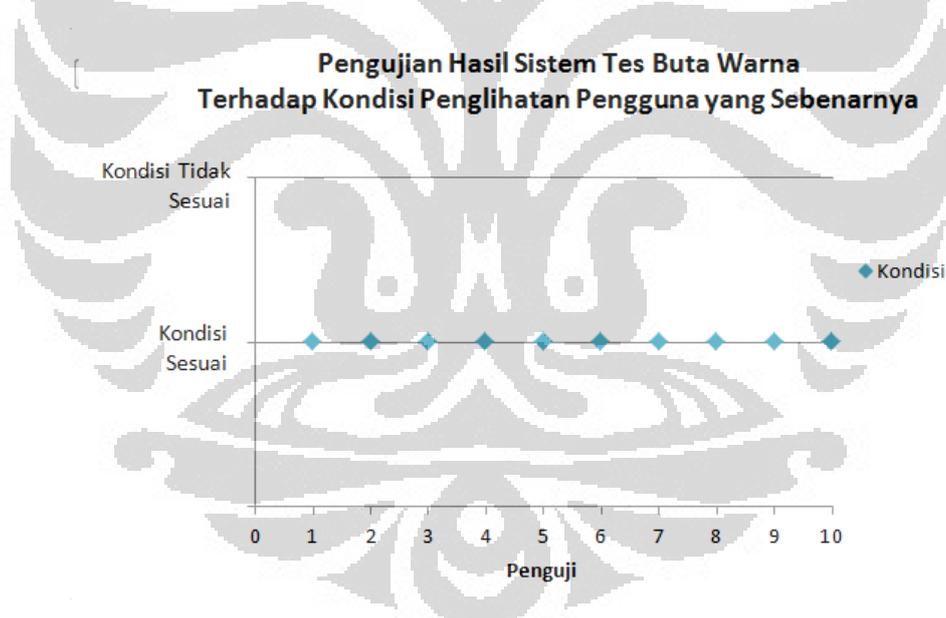
Sebagai salah satu tes yang terkait dengan hal medis, pengetesan buta warna harus memiliki akurasi kebenaran yang tinggi dalam pengujian, karena tes medis seperti tes buta warna sering dijadikan suatu persyaratan kelengkapan dokumen. Pengimplementasian sistem tes buta warna dengan metode Ishihara pada *Windows Phone 7* harus dapat mewakili tes buta warna yang biasa dilakukan secara manual.

Metode pengujian dilakukan dengan mengambil sampel sebanyak 10 pengguna untuk melakukan tes buta warna hasil implementasi penelitian yang sudah diintegrasikan pada perangkat bergerak. Untuk menambahkan tingkat validasi data yang diperoleh, sampel yang diambil terdiri dari pengguna yang berpenglihatan normal dan penyandang buta warna. Persentase perbandingan jumlah penguji yang berpenglihatan normal dan buta warna adalah 8 banding 2. Parameter pengujian tingkat akurasi tes buta warna adalah kesamaan kondisi hasil tes menggunakan sistem bantuan penyandang buta warna dengan kondisi hasil tes buta warna yang dilakukan secara manual dengan seorang ahli medis. Apabila

hasil pengujian sama dengan kondisi karakter penglihatan pengguna, maka dinyatakan sesuai kondisi. Apabila hasil pengujian buta warna pada perangkat tidak menghasilkan kondisi yang sama dengan penglihatan warna pada penguji, maka sistem tes buta warna dinyatakan tidak sesuai kondisi.

Tabel 5.4. Kondisi penglihatan penguji menggunakan sistem tes buta warna

Sampel	Kondisi Penglihatan Penguji terhadap Warna
Penguji 1	Kondisi sesuai dengan kondisi sebenarnya
Penguji 2	Kondisi sesuai dengan kondisi sebenarnya
Penguji 3	Kondisi sesuai dengan kondisi sebenarnya
Penguji 4	Kondisi sesuai dengan kondisi sebenarnya
Penguji 5	Kondisi sesuai dengan kondisi sebenarnya
Penguji 6	Kondisi sesuai dengan kondisi sebenarnya
Penguji 7	Kondisi sesuai dengan kondisi sebenarnya
Penguji 8	Kondisi sesuai dengan kondisi sebenarnya
Penguji 9	Kondisi sesuai dengan kondisi sebenarnya
Penguji 10	Kondisi sesuai dengan kondisi sebenarnya



Gambar 5.7. Grafik pengujian sistem tes buta warna terhadap kondisi penglihatan pengguna

Berdasarkan grafik pengujian pada Gambar 5.7, terbukti bahwa sistem tes buta warna yang diimplementasikan pada perangkat bergerak memberikan hasil yang optimal dan memiliki tingkat akurasi yang tinggi. Dari 10 penguji atau responden yang mencoba fitur tes buta warna, semuanya memiliki kondisi yang sesuai antara hasil tes dengan sistem tes buta warna pada perangkat *Windows*

Phone 7 dengan hasil tes buta warna manual oleh ahli medis. Jadi, penggunaan 21 citra Ishihara sudah merepresentasikan metode tes buta warna dengan hasil yang akurat.

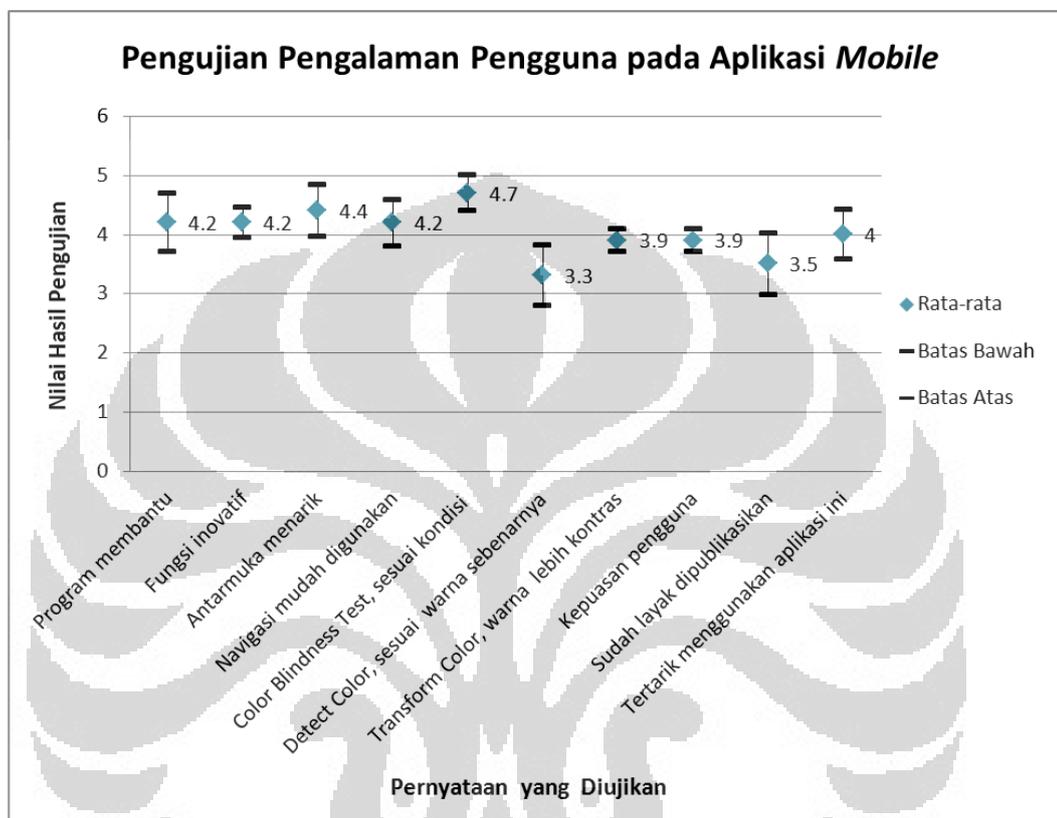
5.1.5 Pengujian Keseluruhan Sistem Bantuan Buta Warna pada Perangkat Bergerak

Selain melakukan pengujian untuk masing-masing modul, dilakukan juga pengujian keseluruhan sistem yang bersifat *user experience*. Pengujian dilakukan oleh 10 responden untuk memberikan penilaian terhadap implementasi seluruh fitur sistem bantuan tes buta warna pada penelitian ini. Parameter yang digunakan dalam pengujian terdiri dari 5 kondisi, yaitu sangat tidak setuju, tidak setuju, agak setuju, setuju, dan sangat setuju, dengan masing-masing nilai untuk tiap kondisi adalah 1, 2, 3, 4, dan 5. Dengan pengolahan data tersebut, diperoleh interval keyakinan 95% hasil pengujian keseluruhan sistem, seperti pada Tabel 5.5.

Tabel 5.5. Interval keyakinan 95% hasil pengujian keseluruhan sistem

No.	Pernyataan	Hasil
1	Program membantu penderita buta warna	4,2 ± 0,489
2	Fungsi-fungsi program inovatif	4,2 ± 0,261
3	Antarmuka (tampilan) program menarik	4,4 ± 0,433
4	Antarmuka (tampilan) dan navigasi program mudah digunakan	4,2 ± 0,392
5	Pada <i>Color Blindness Test</i> , hasil program sesuai dengan kondisi pengguna	4,7 ± 0,299
6	Pada <i>Detect Color</i> , hasil program sesuai dengan warna sebenarnya	3,3 ± 0,510
7	Pada <i>Transform Color</i> , warna yang diubah menjadi lebih kontras dan terlihat	3,9 ± 0,196
8	Pengguna puas menggunakan program ini	3,9 ± 0,196
9	Aplikasi sudah layak digunakan banyak pengguna	3,5 ± 0,527
10	Jika dipublikasikan, pengguna tertarik menggunakan aplikasi ini	4 ± 0,413

Dari hasil pengujian tersebut, bagian yang menjadi lingkup penelitian penulis, yaitu pernyataan tentang antarmuka pengguna, sistem tes buta warna, dan transformasi warna. Perbandingan nilai interval kepercayaan 95% terhadap keseluruhan hasil pengujian dapat dilihat pada grafik Gambar 5.8.



Gambar 5.8. Grafik pengujian keseluruhan sistem pada perangkat bergerak

Berdasarkan grafik Gambar 5.8, nilai hasil pengujian tertinggi adalah pada pernyataan bahwa fitur tes buta warna sesuai kondisi atau dengan kata lain memiliki hasil pengujian yang akurat. Secara keseluruhan, nilai hasil pengujian memiliki nilai di atas rata-rata yang mengindikasikan responden puas terhadap sistem bantuan buta warna yang diimplementasikan pada perangkat bergerak.

5.2 Analisis Pengujian pada Perangkat Tertanam (eBox 3310)

Pengujian sistem bantu penyandang buta warna pada perangkat tertanam bertujuan untuk membuktikan hipotesa dan mengetahui tanggapan pengguna terhadap akurasi tingkat kejelasan citra setelah transformasi warna dan dibandingkan dengan sebelum penggunaan transformasi warna. Pengujian sistem

bantuan penyandang buta warna secara keseluruhan juga dilakukan dengan mengambil sampel 10 pengguna untuk menguji sistem yang sudah diintegrasikan.

5.2.1 Tingkat Kejelasan Citra dengan Pentransformasian Warna

Seperti pada pengujian yang dilakukan untuk sistem pada perangkat bergerak, pengujian ini juga bertujuan untuk mengetahui tingkat akurasi terhadap persepsi pengguna dalam menilai kejelasan citra yang terlihat setelah dilakukan transformasi warna. Untuk itu, metode pengujian yang dijalankan sama pula, yaitu menggunakan citra Ishihara yang terdapat pada Gambar 5.4.

Metode transformasi warna yang digunakan pada perangkat bergerak berbeda dengan perangkat tertanam. Pada perangkat tertanam hanya menggunakan pengkonversian format warna ke dalam format HSV. Pengujian ini dalam analisis akhirnya dapat menunjukkan perbandingan terhadap kedua metode tersebut. Data hasil pengujian tingkat kejelasan citra dengan pentransformasian warna pada perangkat tertanam dapat dilihat berdasarkan tabel di bawah dan direpresentasikan dengan grafik perbandingan pengaruh penggunaan sistem.

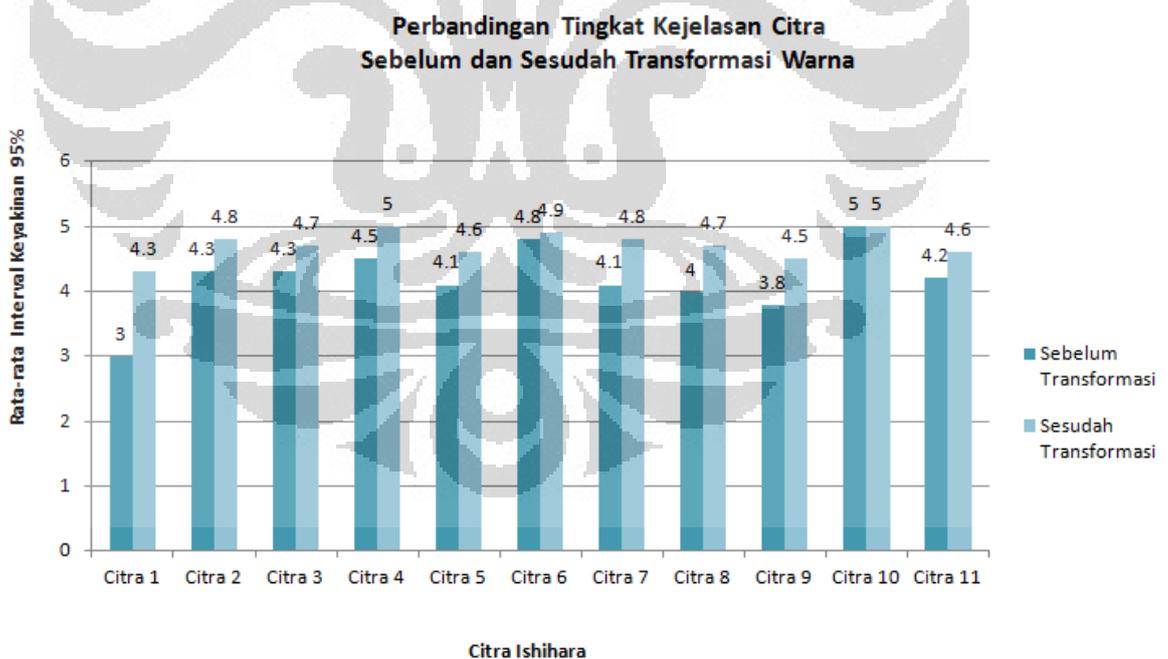
Tabel 5.6. Interval keyakinan 95% hasil uji kejelasan citra

Citra Ishihara	Hasil	
	Tanpa Sistem Bantuan	Dengan Sistem Bantuan
Citra 1	$3 \pm 0,773$	$4,3 \pm 0,510$
Citra 2	$4,3 \pm 0,926$	$4,8 \pm 0,261$
Citra 3	$4,3 \pm 0,776$	$4,7 \pm 0,299$
Citra 4	$4,5 \pm 0,669$	5 ± 0
Citra 5	$4,1 \pm 0,742$	$4,6 \pm 0,320$
Citra 6	$4,8 \pm 0,392$	$4,9 \pm 0,196$
Citra 7	$4,1 \pm 0,898$	$4,8 \pm 0,261$
Citra 8	$4 \pm 1,012$	$4,7 \pm 0,299$
Citra 9	$3,8 \pm 0,960$	$4,5 \pm 0,438$
Citra 10	5 ± 0	5 ± 0
Citra 11	$4,2 \pm 1,033$	$4,6 \pm 0,320$
Rata-rata	$4,19 \pm 0,744$	$4,72 \pm 0,264$

Berdasarkan data hasil pengujian, pada citra 10 memiliki nilai interval keyakinan 95% maksimum, yaitu pengguna sangat setuju bahwa dengan dan

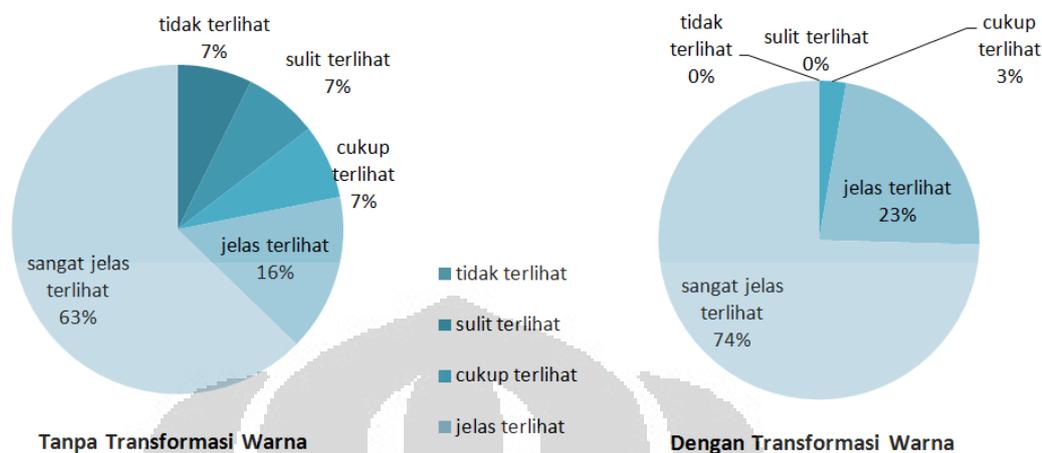
tanpa penggunaan sistem transformasi warna, citra yang terlihat oleh pengguna sangat jelas, karena citra Ishihara tersebut memiliki variasi warna yang jelas terlihat, baik pada penderita buta warna, maupun orang berpenglihatan normal.

Perbandingan hasil rata-rata nilai interval kepercayaan pengujian menunjukkan bahwa dengan menggunakan sistem transformasi warna, pengguna dapat lebih jelas melihat angka pada citra Ishihara dibandingkan tanpa sistem bantuan transformasi warna. Artinya, implementasi metode transformasi warna pada perangkat tertanam membantu pengguna, baik yang berpenglihatan normal maupun buta warna sebagian, untuk mengenali objek dengan warna ambigu. Hal tersebut dibuktikan dengan nilai rata-rata interval keyakinan 95% untuk kondisi tanpa sistem bantu transformasi warna adalah $4,19 \pm 0,744$, sedangkan untuk kondisi menggunakan sistem transformasi warna adalah $4,72 \pm 0,264$ dari skala nilai maksimum interval sebesar 5 poin. Jadi, penggunaan sistem transformasi warna pada perangkat tertanam memberikan pengaruh yang sangat tinggi dalam memperjelas citra berwarna ambigu.



Gambar 5.9. Grafik perbandingan tingkat kejelasan citra

Perbandingan Persentase Penglihatan Terhadap Citra



Gambar 5.10. Grafik persentase kejelasan penglihatan terhadap citra

Data pengujian di atas menunjukkan bahwa dengan penggunaan sistem transformasi warna yang diimplementasikan pada perangkat tertanam dapat menghilangkan kemungkinan kondisi sulit dan tidak terlihatnya objek dengan warna ambigu. Perbandingan penggunaan sistem transformasi warna pada perangkat tertanam dan perangkat bergerak tidak terlalu berbeda, hal tersebut dapat dilihat pada diagram Gambar 5.10 dan Gambar 5.6. Berdasarkan perbandingan olah data tersebut, dapat disimpulkan bahwa metode transformasi warna yang diterapkan pada perangkat tertanam memberikan hasil yang lebih baik daripada perangkat bergerak dengan persentase kondisi citra yang terlihat sangat jelas lebih banyak pada kondisi penggunaan sistem transformasi warna pada perangkat tertanam. Pada implementasi metode transformasi warna untuk perangkat tertanam, persentase kondisi citra yang sangat jelas terlihat adalah 74%, dan kondisi citra yang jelas terlihat adalah 23%. Pada implementasi transformasi warna untuk perangkat bergerak, persentase kondisi citra yang sangat terlihat jelas sebesar 37%, dan kondisi citra yang jelas terlihat adalah 38%.

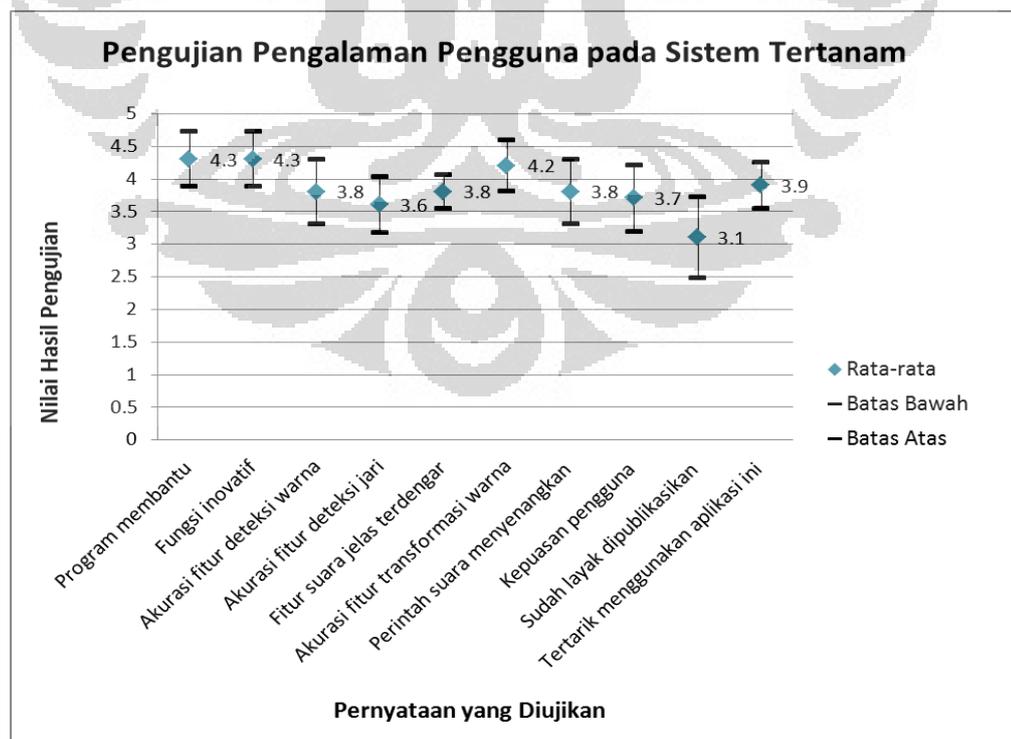
5.2.2 Pengujian Keseluruhan Sistem Bantuan Buta Warna pada Perangkat Tertanam

Pengujian keseluruhan sistem bantu bagi penyandang buta warna dilakukan berdasarkan penilaian oleh 10 pengguna. Data hasil pengujian diolah untuk

mendapatkan nilai interval keyakinan pengujian 95%, seperti pada Tabel 5.7.

Tabel 5.7. Interval keyakinan 95% hasil pengujian keseluruhan sistem

No.	Pernyataan	Hasil
1	Program membantu penderita buta warna	$4,3 \pm 0,418$
2	Fungsi-fungsi program inovatif	$4,3 \pm 0,418$
3	Pada <i>Detect Color</i> (mode 1), hasil program sesuai dengan warna sebenarnya	$3,8 \pm 0,489$
4	Pada <i>Finger Detection</i> (mode 2), objek yang ditunjuk warnanya sesuai dengan kondisi sebenarnya	$3,6 \pm 0,433$
5	Suara yang diucapkan di mode 2 jelas terdengar	$3,8 \pm 0,261$
6	Pada <i>Transform Color</i> (mode 3), warna yang diubah menjadi lebih kontras dan lebih terlihat	$4,2 \pm 0,392$
7	Memerintah dengan suara lebih menyenangkan	$3,8 \pm 0,489$
8	Pengguna puas menggunakan program ini	$3,7 \pm 0,510$
9	Aplikasi sudah layak digunakan banyak pengguna	$3,1 \pm 0,616$
10	Jika dipublikasikan, pengguna tertarik menggunakan aplikasi ini	$3,9 \pm 0,352$



Gambar 5.11. Grafik hasil uji keseluruhan sistem pada perangkat tertanam

Berdasarkan perbandingan grafik yang dihasilkan pada Gambar 5.8 dan Gambar 5.11, menunjukkan bahwa pengembangan keseluruhan sistem bantuan bagi penderita buta warna pada perangkat bergerak lebih memuaskan pengguna dibanding dengan pengimplementasian pada perangkat tertanam. Hal tersebut dapat terlihat dari tren pola grafik. Pada perangkat bergerak, nilai interval kepercayaan terendah berkisar antara nilai 3,3 dari nilai maksimum 5, yang berarti pengguna sudah cukup puas terhadap sistem bantu buta warna yang dibangun pada perangkat bergerak. Nilai interval kepercayaan terendah untuk pengimplementasian pada perangkat tertanam adalah berkisar antara nilai 3,1 yang berarti pengguna juga cukup puas terhadap pengembangan sistem bantu buta warna.

Untuk pengujian terhadap parameter penggunaan fitur transformasi warna, metode yang diterapkan pada perangkat tertanam lebih baik dibanding pada perangkat bergerak. Pada perangkat tertanam, nilai rata-rata interval kepercayaan 95% adalah 4,2, sedangkan pada perangkat bergerak, adalah 3,9. Hal tersebut menunjukkan bahwa sistem transformasi warna yang diimplementasikan pada kedua perangkat sudah memenuhi harapan pengguna dengan perbandingan penerapan metode transformasi warna pada perangkat tertanam 8,9% lebih baik dibanding pada perangkat bergerak.

BAB 6

KESIMPULAN

1. Aplikasi sistem bantuan untuk penderita buta warna dapat diimplementasikan dengan menerapkan konsep *augmented reality*.
2. Pengujian antarmuka pengguna terhadap 10 responden didapatkan nilai interval kepercayaan 95% untuk masing-masing parameter sebagai berikut:
 - Ketertarikan pengguna terhadap antarmuka : $4,4 \pm 0,43$ (dari skala 1 – 5)
 - Kemudahan penggunaan dan navigasi : $4,2 \pm 0,39$ (dari skala 1 – 5)
3. Pengujian sistem tes buta warna pada 10 responden dengan perbandingan 2 orang penyandang buta warna dan 8 orang berpenglihatan normal, menunjukkan hasil keluaran sistem yang akurat terhadap kondisi penglihatan pengguna, yaitu 100% dari 10 kali pengujian responden.
4. Kualitas kamera pada perangkat, pencahayaan, dan posisi pengambilan citra dengan kamera, sangat mempengaruhi sistem transformasi warna.
5. Dengan meningkatkan nilai *saturation* dan *value* pada suatu warna, serta menggeser *hue* menuju titik kebutaan seseorang ternyata dapat meningkatkan kontras warna dan penderita buta warna dapat mengambil informasi citra dengan lebih baik.
6. Hasil analisis pengujian menunjukkan bahwa metode transformasi warna pada perangkat tertanam, yaitu hanya dengan mengkonversikan format citra ke dalam format HSV ternyata lebih baik daripada metode transformasi warna yang diterapkan pada perangkat bergerak.
7. Sistem transformasi warna pada perangkat tertanam menghasilkan persentase kondisi citra yang sangat terlihat jelas adalah 74% dan kondisi citra yang terlihat jelas adalah 23%. Pada perangkat bergerak, persentase kondisi citra yang sangat terlihat jelas adalah 37% dan kondisi citra yang terlihat jelas adalah 38%.

DAFTAR REFERENSI

- [1] Hoffman, Paul S. (1999, October). Accommodating Color Blindness. *Usability Interface, The Usability SIG Newsletter*, Vol. 6, No.2.
- [2] Ohkubo, Tomoyuki, & Kobayashi, Kazuyuki. (2008). A Color Compensation Vision System for Color-blind People. *SICE Annual Conference 2008, Japan*, 1286-1289.
- [3] McDowell, Jason. (2008). Design of a Color Sensing System to Aid the Color Blind. *IEEE Potentials*, 34-39.
- [4] Marzuki, C. (1999). *Metodologi Riset*. Jakarta: Erlangga.
- [5] Singarimbun, Masri, & Sofyan, Efendi. (1987). *Metode Penelitian Survey*, LP3E.
- [6] Fidaner, O., Lin, P., Ozguven, N. (2004). Analysis of Colorblindness. Departement of Electrical Engineering The Standford University.
- [7] Shuhua, Li, & Gaizhi, Guo. (2010). The Application of Improved HSV Color Space Model in Image Processing. 2nd International Conference on Future Computer and Communication. China.
- [8] Efg's Computer Lab Color. *HSV Lab Report*. <http://www.efg2.com/Lab/Graphics/Colors/HSV.htm>. Diakses pada tanggal 14 April 2011.
- [9] Ohkubo, Tomoyuki, & Kobayashi, Kazuyuki. (2008). A Color Compensation Vision System for Color-blind People. *SICE Annual Conference 2008, Japan*, 1286-1289.
- [10] Wakita, Ken, & Shimamura, Kenta. (2005, October). *SmartColor: Disambiguation Framework for The Colorblind*. Paper presented at the ASSETS 2005 Conference, Baltimore, Maryland, USA.
- [11] Graham, C.H., & Hsia, Yun. (1958). The Spectral Luminosity Curves for a Dichromatic Eye and a Normal Eye in The Same Person (vol. 44). *Communicated November 6, 1957*. Department of Psychology, Columbia University.
- [12] Arditi, A. (1999). *Effective Color Contrast: Designing for People with Partial Sight and Congenital Color Deficiencies*. New York: Lighthouse International.
- [13] Case, Betsy J. (2003). *Color Blindness*. Texas: Pearson Education, Inc.
- [14] Contractors to Lift Colour Blind Bar. (2004, April). *IEE Review*, p. 16.

- [15] Indrawan, Eri. (2008). *Perangkat Lunak Transformasi Warna untuk Penderita Buta Warna*. Institut Teknologi Bandung: Bandung.
- [16] Yoh, Myeung-Sook. (2001). The Reality of Virtual Reality. *Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM'01)*. IEEE.
- [17] Jacko, Julie A., & Sears, Andrew. (2010, December 16). Handbook of Research on Ubiquitous Computing Technology for Real Time Enterprises. *CRC Press*, p. 459.
- [18] Bimber, Oliver, dan Ramesh, Rakar. (2005). *Spatial Augmented Reality*. Massachusetts: A.K. Peters.
- [19] Leete, Gurdy, & Leete, Mary. (2007). *Microsoft Expression Blend Bible*. Indianapolis: Wiley Publishing, Inc.
- [20] Gaudioso, Victor. (2009). *Foundation Expression Blend 3 with Silverlight*. United States of America: Springer-Verlag New York, Inc.
- [21] Pramudya, Puja. (2010). *Silverlight for Windows Phone, Learn & Practice*. Institut Teknologi Bandung, Microsoft Innovation Center.
- [22] Gould, John D., & Lewis, Clayton. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM*, 28(3): 300-311.
- [23] Mayhew, Deborah J. (1992). *Principles and Guidelines in Software User Interface Design*. New Jersey: Prentice-Hall, Inc.
- [24] Shu, Yang, & Yang Zheng. (2008). Research on Human-Computer-Interaction Design of Software Based on Barrier-free Principle. *IEEE Journal*.
- [25] Colour Vision Examination. (2006, October). Guidance Note MS7 (Third Edition). Published by the Health and Safety Executive.
- [26] Dougherty, Bob, & Wade, Alex. (2005, May 3). *Color Blind Image Correction*. Vischeck, a General Partnership, Menlo Park, California. <http://www.vischeck.com/daltonize/>. Diakses pada tanggal 2 Desember 2010.
- [27] Vischeck. (2008). *Try Vischeck on Your Image Files*. <http://www.vischeck.com/vischeck/vischeckImage.php>. Diakses pada tanggal 20 April 2011.
- [28] Albahari, Ben, Peter Drayton, dan Bradd Merrill. (2001). *C# Essentials, Second Edition*. USA: O'Reilly.
- [29] Bradski, Gray, dan Adrian Kaehler. (2008). *Learning OpenCV*. USA: O'Reilly Media.

- [30] 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.
- [31] Harwahyu, Ruki. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Perancangan Sistem Tertanam dengan Interaksi Suara*. Depok: Universitas Indonesia.
- [32] Wicaksana, Burhan Adi. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Pendeteksian Warna, Pelacakan Gerakan (Motion Tracking), dan Tampilan Informasi Warna dengan Platform .NET dan EmguCV Library*. Depok: Universitas Indonesia.
- [33] Ananto, Bayu Sri. (2010). *Perancangan Sistem Bantuan Penderita Buta Warna: Perancangan Antarmuka Pengguna dan Integrasi Hasil Pengolahan Sistem Realitas Tertambah*. Depok: Universitas Indonesia.
- [34] _____. *Introduction to Unified Modelling Language (UML)*. [PowerPoint Presentation]. Depok: Faculty of Engineering University of Indonesia, 2008.
- [35] Fowler, Martin. (2003). *UML Distilled: A Brief Guide to the Standard Object Modelling Language, Third Edition*. USA: Addison Wesley.
- [36] Steve Resnick, Richard Crane, Chris Bowen: *Essential Windows Communication Foundation (WCF): For .NET Framework 3.5*, Addison-Wesley, February 11, 2008, ISBN 0-321-44006-4