UNIVERSITAS INDONESIA

IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA: SISTEM TERTANAM BERBASIS KONSEP REALITAS TERTAMBAH DENGAN METODE INTERAKSI LANGSUNG PENGGUNA DENGAN OBJEK WARNA

SKRIPSI

ALFA SHEFFILDI MANAF
0706275870

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JULI 2011

UNIVERSITAS INDONESIA

IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA: SISTEM TERTANAM BERBASIS KONSEP REALITAS TERTAMBAH DENGAN METODE INTERAKSI LANGSUNG PENGGUNA DENGAN OBJEK WARNA

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Teknik

ALFA SHEFFILDI MANAF
0706275870

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JULI 2011

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan benar.

Nama : Alfa Sheffiildi Manaf

NPM : 0706275870

Tanda Tangan : 

Tanggal : 1 Juli 2011

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh	:
Nama	: Alfa Sheffildi Manaf
NPM	: 0706275870
Program Studi	: Teknik Komputer
Judul Skripsi	: Implementasi Sistem Bantuan Penderita Buta Warna: Sistem Tertanam Berbasis Konsep Realitas Tertambah dengan Metode Interaksi Langsung Pengguna dengan Objek Warna

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

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

UCAPAN TERIMA KASIH

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

1. Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc. selaku pembimbing akademis dan dosen pembimbing atas segala bimbingan, ilmu, dan arahan baik dalam penulisan seminar maupun selama masa studi di Teknik Komputer.
2. Kedua orang tua saya serta keluarga yang selalu menjadi sumber semangat dan inspirasi
3. Teman-teman di program studi Teknik Komputer dan Teknik Elektro Universitas Indonesia Angkatan 2007 atas segala dukungan dah kerja samanya.
4. Seluruh sivitas akademika Departemen Teknik Elektro Universitas Indonesia

sehingga penulisan skripsi ini dapat diselesaikan dengan baik.

Depok, 11 Juli 2011

Penulis,

Alfa Sheffildi Manaf

NPM. 0706275870

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama	: Alfa Sheffildi Manaf
NPM	: 0706275870
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 Non-eksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

IMPLEMENTASI SISTEM BANTUAN PENDERITA BUTA WARNA: SISTEM TERTANAM BERBASIS KONSEP REALITAS TERTAMBAH DENGAN METODE INTERAKSI LANGSUNG PENGGUNA DENGAN OBJEK WARNA

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Ekslusif ini Universitas Indonesia berhak meyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta sebagai pemegang Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok
Pada tanggal: 11 Juli 2011
Yang menyatakan

Alfa Sheffildi Manaf

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

ABSTRAK

Nama	:	Alfa Sheffildi Manaf
Program Studi	:	Teknik Komputer
Judul	:	Implementasi Sistem Bantuan Penderita Buta Warna: Sistem Tertanam Berbasis Konsep Realitas Tertambah dengan Metode Interaksi Langsung Pengguna dengan Objek Warna

Buta warna adalah kelainan pada retina mata yang menyebabkan penyandangnya tidak bisa mengenali atau membedakan warna tertentu. Ketidakmampuan mengenali warna ini berpotensi menyebabkan berbagai kesulitan bagi para penderitanya dalam kehidupan sehari-hari. Kelainan buta warna tidak bisa disembuhkan. Oleh karena itu, satu-satunya cara untuk membantu penyandang buta warna membedakan warna adalah dengan menggunakan alat bantu. Dalam skripsi ini, dikembangkan aplikasi bantuan penderita buta warna untuk *platform* sistem tertanam berbasis *Windows Embedded Standard 2009, .NET Framework, OpenCV library* serta *EmguCV Wrapper*. Sistem ini terdiri dari beberapa fitur pengenal warna yang diterapkan dengan konsep realitas tertambah. Implementasi sistem yang dibahas pada skripsi ini meliputi sistem bantu yang dikembangkan dengan konsep realitas tertambah suara, dengan tujuan membantu penyandang buta warna mengenali warna dengan media suara melalui interaksi jari pengguna pada objek warna. Sistem ini mendapatkan hasil yang cukup memuaskan berdasarkan pengujian dan tanggapan dari para responden. Hasil pengujian interaksi jari menunjukkan tingkat deteksi jari mencapai 89.6% untuk metode klasifikasi kulit dengan format warna HSV. Sedangkan, tingkat deteksi jari menggunakan metode klasfikasi kulit dengan format YCbCr mencapai 87.5%. Selain itu, tingkat pengenalan warna yang didapat mencapai tingkat yang baik untuk mayoritas warna-warna tertentu yang diuji.

Kata kunci: *buta warna, realitas tertambah,* sistem tertanam, *Windows Embedded Standard 2009, .NET Platform, OpenCV, EmguCV,* interaksi jari, pengenal warna

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

ABSTRACT

Name	:	Alfa Sheffildi Manaf
Study Program	:	Computer Engineering
Title	:	Implementation of Color Blind Aid System: Embedded System Based on Augmented Reality Concept with Direct Interaction Method between User and Colored Object

Color blindness is an anomaly which happened in retinal of eye(s) which prevent the patient to recognize or differentiate certain colors. The disability of the patient to recognize color is potential to cause problems to the patient in daily life. Color blind cannot be cured. Therefore, the only one method to help color blind people to recognize or differentiate color is with a vision aid kit. In this final project, color blind aid system for embedded *platform* based on *Windows Embedded Standard 2009, .NET Framework, OpenCV library* and *EmguCV Wrapper* developed. There will be kind of color recognition features implemented with augmented reality concept in the system. Specifically, this paper explains the implementation of aid system, which is developed with sound augmented reality concept and finger interaction between user and colored object. This system receives good enough result according to system testing which has been done and responses from respondents. The result of finger interaction test shows that the fingertip detection rate reaches 89.6% for skin classification method with HSV color space. Meanwhile, fingertip detection rate reaches 87.5% for skin classification method with YCbCr color space. Furthermore, color recognition rate achieved good result for majority of certain tested color types.

Keywords: *color blindness, augmented reality, embedded system, Windows Embedded, .NET Platform, OpenCV, EmguCV, finger interaction, color recognition*

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

DAFTAR ISI

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

DAFTAR GAMBAR

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

DAFTAR TABEL

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kelainan buta warna merupakan kelainan yang menyebabkan para penderitanya tidak sensitif terhadap warna-warna tertentu atau dengan kata lain tidak dapat membedakan objek warna tertentu. Dalam kehidupan, kelainan buta warna dapat berdampak buruk dalam perkembangan individu penderita. Kelainan buta warna merupakan penyakit bawaan/turunan yang dibawa oleh salah satu jenis gen warisan orang tua para penyandang buta warna. Pada kasus-kasus tertentu juga dijumpai penderita buta warna yang disebabkan oleh kerusakan saraf, otak, atau mata.

Penderita buta warna dibagi menjadi 2 jenis, yang pertama adalah penyandang buta warna parsial/sebagian dan penyandang buta warna total. Penyandang buta warna parsial tidak dapat membedakan warna tertentu tergantung dari jenis buta warna parsial yang mereka alami, sedangkan para penderita buta warna total hanya dapat membedakan warna hitam, putih dan abu-abu. Hal ini terjadi karena retina mata pada penderita buta warna parsial memiliki dua sel yang diperlukan untuk melihat benda, yaitu sel batang yang sensitif terhadap cahaya dan sel kerucut (konus) yang sensitif terhadap warna. Hanya saja pada penderita buta warna parsial sebagian sel konus yang ada tidak berfungsi sempurna. Retina penyandang buta warna total tidak memiliki sel konus. Sehingga, mata penderita buta warna total tidak sensitif terhadap warna. Menurut beberapa sumber, banyaknya penderita buta warna mencapai 4-8% populasi pria di dunia, sedangkan penderita buta warna wanita kurang dari 1% populasi wanita di dunia.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Kelainan buta warna sedikit banyak berdampak negatif pada para penyandangnya. Contohnya, hal ini mempengaruhi penyandang buta warna dalam memilih program studi untuk melanjutkan pendidikannya atau dalam memilih karir selanjutnya dalam hidupnya, saat ini beberapa pilihan program studi dan karir mensyaratkan mahasiswanya/karyawannya tidak buta warna. Selain itu, dampaknya juga muncul pada kehidupan sehari-hari penyandang buta warna, seperti kesulitan dalam membedakan warna pakaian atau warna objek-objek lainnya di sekitar manusia.

Hal ini menjadi latar belakang riset ini, karena secara umum latar belakang ini juga salah satu yang diproyeksikan oleh PBB (Persatuan Bangsa Bangsa) dalam *Millennium Development Goals* (MDGs) poin kedua, yaitu persamaan hak dalam pendidikan (*universal education*).

1.2 Tujuan Penelitian

Tujuan dari penulisan penelitian ini adalah merancang dan mengimplementasi sebuah sistem yang terdiri dari sistem fungsi tunggal (*single purpose system*) berupa sistem tertanam (*embedded system*) dan aplikasi divais bergerak (*mobile device*) yang memiliki beberapa fitur dengan konsep realitas tertambah (*augmented reality*). Sistem ditujukan untuk membantu penyandang buta warna dalam membedakan warna objek-objek di sekitar mereka. Dengan adanya alat bantuan ini diharapkan batasan (*barrier*) antara bidang pekerjaan/bidang studi dengan penyandang buta warna dapat diperkecil. Selain itu juga dilakukan implementasi dan uji coba metode interaksi manusia dan komputer menggunakan jari pada sistem realitas tertambah.

1.3. Batasan Masalah

Permasalahan yang akan dibahas pada penelitian ini dibatasi pada rancang bangun sistem fungsi tunggal yang berupa sebuah sistem tertanam bantuan untuk penyandang buta warna dengan konsep realitas tertambah melalui

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

tambahan realitas berupa sinyal suara, untuk itu diperlukan suatu perancangan sistem yang baik untuk menghasilkan sistem dengan *user experience* yang baik dan tepat guna ketika pengguna menggunakan alat ini. Sistem yang dirancang ini akan memungkinkan pengguna alat ini yang merupakan penyandang buta warna dapat mengetahui warna-warna objek dengan menunjuk atau menyentuh objek warna.

1.4. Sistematika Penulisan

Pembahasan yang dilakukan pada penelitian ini meliputi 4 bab, yaitu :

Bab 1 Pendahuluan

Bagian ini terdiri dari latar belakang masalah, tujuan penulisan, batasan masalah dan sistematika penulisan

Bab 2 Dasar Teori

Bagian ini berisi teori kelainan buta warna, sistem bantuan untuk penyandang buta warna, *Microsoft .NET* sebagai *platform* pengembangan sistem tertanam dan penggunaan *platform Microsoft .NET* untuk aplikasi *augmented reality*

Bab 3 Perancangan Sistem Bantuan Pengenal Warna dengan Konsep *Augmented Reality*

Bagian ini berisi deskripsi umum sistem, arsitektur sistem tertanam, komponen perangkat keras, *point-to-sound augmented reality*, aliran data dan pemrosesan sinyal.

Bab 4 Implementasi Sistem

Bagian ini membahas tahapan-tahapan implementasi sistem sesuai dengan perancangan

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Bab 5 Pengujian dan Analisis

Bagian ini membahas pengujian yang dilakukan terhadap sistem yang telah diimplementasi, serta analisis yang dilakukan berdasarkan pengujian yang dilakukan

Bab 6 Kesimpulan

Bagian ini berisi poin-poin kesimpulan yang dapat ditarik dari penelitian ini

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

BAB II

DASAR TEORI

2.1 Kelainan Buta Warna

Kelainan buta warna merupakan kelainan yang terjadi pada mata (retina), yang menyebabkan penderita buta warna tidak bisa membedakan warna-warna tertentu. Pada banyak kasus kelainan buta warna disebabkan oleh kelainan gen pada kromosom X yang dibawa oleh orang tua para penderita buta warna, tetapi terdapat faktor lain yang menyebabkan seseorang menderita buta warna, yaitu kerusakan langsung saraf, otak dan mata. Sebenarnya penyebutan kalimat "buta warna" untuk kelainan jenis ini kurang tepat, karena penderita kelainan ini tetap melihat warna-warna, hanya saja kemampuan melihat warna-warna tertentu menjadi terbatas.

Jumlah populasi manusia yang mengalami kelainan buta warna ini mencapai 4-8% (8% pada populasi kaukasia, 5% pada populasi asia, dan 4% pada populasi afrika) [1] di dunia untuk laki-laki. Kurang dari 1% populasi wanita dunia adalah penderita buta warna. Jumlah penderita buta warna laki-laki jauh lebih banyak dari penderita buta warna perempuan karena kelainan buta warna turunan (genetik) dibawa oleh jenis kromosom X. Perempuan mempunyai 2 kromosom X (XX), sedangkan laki-laki diwarisi 1 kromosom X dan 1 kromosom Y (XY) oleh orang tuanya.

Dari fakta tersebut, terlihat bahwa peluang laki-laki dominan terhadap penyakit buta warna lebih tinggi karena hanya dipengaruhi oleh 1 kromosom X. Pada wanita, seseorang akan mengalami buta warna jika kedua kromosom X yang diwarisi membawa kelainan buta warna. Jika hanya 1 kromosom X yang membawa informasi kelainan buta warna maka wanita tersebut tidak akan mengalami buta warna. Individu tersebut hanya akan berperan sebagai pembawa kromosom X yang membawa informasi kelainan buta warna

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

(*carrier*), ketika nantinya akan mewariskan gen tersebut ke anaknya. Kelainan buta warna diklasifikasikan menjadi 2 jenis kelainan buta warna, yaitu buta warna sebagian (parsial) dan buta warna total.

2.1.1 Buta Warna Sebagian (parsial)

Buta warna sebagian merupakan jenis kelainan buta warna yang paling sering ditemui. Pada buta warna jenis ini, para penderita tidak dapat membedakan warna-warna tertentu, seperti merah-hijau atau biru-kuning. Buta warna parsial disebabkan karena satu atau lebih sel konus pada retina mata yang sensitif terhadap warna bermasalah. Sel konus ini juga sering disebut sel kerucut. Pada retina mata manusia terdapat 2 jenis sel yang berfungsi untuk menangkap bayangan, yaitu sel batang dan sel konus. Sel batang sensitif terhadap cahaya dan aktif di saat kondisi cahaya rendah, sedangkan sel konus/kerucut sensitif terhadap warna dan aktif di saat kondisi cahaya normal. Terdapat 3 jenis sel konus pada retina normal, yaitu *S-Cones*, *M-Cones* dan *L-Cones*. Masing-masing jenis sel konus ini memiliki sensitivitas yang berbeda-beda terhadap spektrum warna tertentu, *S-Cones* sensitif terhadap cahaya dengan spektrum warna biru, *M-Cones* sensitif terhadap cahaya dengan spektrum hijau, sedangkan *L-Cones* sensitif terhadapa cahaya dengan spektrum merah. Dengan adanya 3 jenis sel konus di atas, maka otak kita dapat menerjemahkan warna-warna hasil dari kombinasi spektrum cahaya yang ditangkap oleh 3 jenis sel konus tadi. Pada retina mata manusia normal, 3 jenis sel konus tadi dapat bekerja dengan baik. Dengan begitu seseorang dapat memiliki persepsi warna yang berbeda-beda untuk mendeteksi warna yang berbeda.

Seperti yang telah dijelaskan di awal paragraf, penderita buta warna sebagian/parsial memiliki kelainan pada sel konus yang terbentuk di retina mata. Kelainan tersebut berupa satu atau lebih jenis sel konus yang tidak berfungsi sempurna ataupun tidak berfungsi sama sekali. Hal ini sangat mempengaruhi rentang spektrum yang dapat dipersepsikan oleh manusia.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Berdasarkan jenis kelainan yang terdapat pada sel konus, jenis buta warna sebagian/parsial dapat dibedakan menjadi 2 bagian besar, yaitu *dichromacy* dan *anomalous thrichromacy*.

A. *Dichromacy*

Dichromacy adalah salah satu jenis buta warna parsial. Pada jenis ini 1 dari 3 jenis sel konus tidak ada pada retina penderita *Dichromacy* atau tidak berfungsi sama sekali. Dengan absennya atau tidak berfungsinya 1 dari 3 jenis sel konus, maka hal ini mempengaruhi kemampuan individu penderita untuk membedakan warna tertentu. Persepsi otak manusia terhadap warna sangat dipengaruhi dari sinyal yang diberikan oleh kombinasi sel konus. *Dichromacy* dibagi menjadi 3 bagian lagi berdasarkan sel konus yang hilang atau tidak berfungsi, yaitu *Protanopia*, *Deutaronopia* dan *Tritanopia*.

Protanopia merupakan jenis *Dichromacy* dimana sel konus yang hilang/tidak berfungsi adalah sel konus yang sensitif terhadap warna merah (*L-Cones*). Pada jenis ini retina sama sekali tidak dapat melihat spektrum warna merah, sehingga mempengaruhi penglihatan penderita terhadap warna merah atau warna-warna dengan campuran spektrum cahaya warna merah. Kasus *Protanopia* terjadi pada 1% pria dan 0.02% wanita [2].

Jenis kedua dari *Dichromacy* adalah *Deuteranopia*, jenis *Dichromacy* ini disebabkan karena sel konus yang sensitif terhadap spektrum warna hijau (*M-Cones*) hilang atau tidak berfungsi sama sekali. Pada jenis ini penderita sama sekali tidak dapat melihat warna yang dipengaruhi spektrum warna hijau. Kasus Deutranopia terjadi pada 1% pria dan 0.01% wanita [2].

Jenis *Dichromacy* ketiga adalah *Tritanopia*. Jenis *Dichromacy* ini disebabkan karena sel konus yang sensitif terhadap spektrum warna biru (*S-Cones*) hilang atau tidak berfungsi sama sekali. Pada jenis ini penderita

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

tidak dapat melihat warna yang dipengaruhi spektrum warna biru. Kasus ini jarang ditemui, terjadi pada 0.001% pria dan 0.03% wanita [2].

Gambar 2.1 menunjukkan simulasi penglihatan penderita buta warna yang dihasilkan dari perangkat simulasi buta warna *colblindor*, yaitu sebuah situs *web* yang menyediakan informasi seputar buta warna sekaligus simulator buta warna.

Gambar 2.1. Simulasi penglihatan (normal, *Protanopia*, *Deuteranopia*, *Tritanopia*) dari perangkat simulator colblindor.com

B. *Anomalous Thrichromacy*

Anomalous Thrichromacy merupakan salah satu jenis buta warna parsial selain *Dichromacy*. Pada buta warna parsial jenis ini semua jenis sel konus (*L-Cones*, *M-Cones* dan *S-Cones*) ada pada retina mata penderita buta warna parsial jenis ini. Hanya saja salah satu dari 3 jenis sel konus tadi

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

tidak berfungsi dengan baik. Dengan kata lain salah satu jenis sel konus tingkat sensitivitasnya kurang dibandingkan sel konus normal. Kurang sempurnanya (tidak berfungsinya) salah satu sel konus, akan mempengaruhi persepsi warna yang dilihat individu penderita jenis ini, khususnya pada warna campuran. *Anomalous Thrichromacy* dibagi menjadi 3 jenis, yaitu *Protanomaly*, *Deuteranomaly* dan *Tritanomaly*.

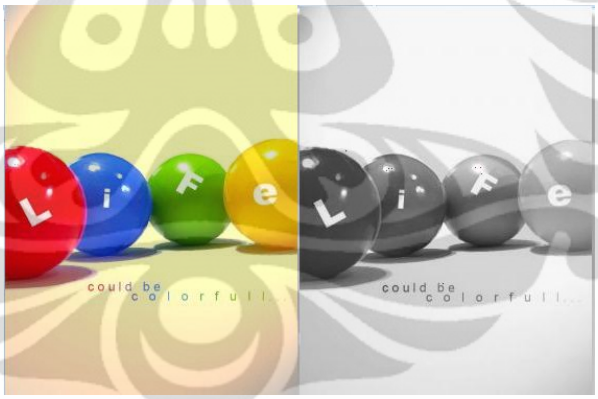
Pada *Protanomaly*, sel konus yang tidak berfungsi sempurna adalah sel konus yang sensitif terhadap spektrum warna merah, kasus *Protanomaly* terjadi pada 1% pria dan 0,01% wanita. Pada *Deuteranomaly*, sel konus yang tidak berfungsi sempurna adalah sel konus yang sensitif terhadap warna hijau. Kasus ini merupakan kasus yang paling banyak ditemukan, yang terjadi pada 6% pria dan 0.4% wanita. Selanjutnya, pada *Tritanomaly* sel konus yang tidak berfungsi sempurna adalah sel konus warna biru. Kasus ini jarang ditemukan, terjadi pada 0.01% pria dan wanita [2].

Gambar 2.2. Simulasi Penglihatan (normal, *Protanomaly*, *Deuteranomaly*, *Tritanomaly*) dari perangkat simulator colblindor.com

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

2.1.2 Buta Warna Total

Buta warna total adalah kasus yang jarang terjadi dibandingkan dengan buta warna parsial. Pada buta warna total, penderitanya benar-benar tidak dapat melihat warna, yang dapat terlihat hanyalah pantulan cahaya dari benda (bayangan benda). Penderita hanya dapat membedakan intensitas cahaya yang dipantulkan cahaya itu saja, misalnya terang, gelap atau abu-abu. Buta warna total dibagi menjadi dua bagian berdasarkan penyebabnya, yaitu *rod monochromacy* dan *cone monochromacy*. Pada kasus *rod monochromacy* retina hanya memiliki sel batang (rod) dan tidak mempunya sel konus. Retina mata tidak sensitif terhadap warna sama sekali. Jenis lainnya adalah *cone monochromacy*. Pada kasus ini retina hanya memiliki salah satu dari 3 jenis sel konus. Sehingga penderita tidak dapat membedakan warna sama sekali, padahal mereka dapat melihat warna tertentu yang sensitif di spektrum warna tertentu.



Gambar 2.3. Simulasi penglihatan (normal & *monochromacy*) dari perangkat simulator colblindor.com

2.1.3 Kesulitan yang Dihadapi Penyandang Buta Warna

Bagi penyandang kelainan buta warna terdapat beberapa kesulitan yang mereka hadapi dalam menjalani hidup. Kesulitan dimulai dari masalah yang sederhana sampai masalah yang cukup krusial dalam hidup. Contoh masalah

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

sederhana yang dihadapi sehari-hari seperti menentukan nama warna, memilih/menentukan warna pakaian, memilih warna yang tepat untuk desain/lukisan, menginterpretasikan rambu lalu-lintas dan masalah sehari-hari lainnya. Kesulitan terbesar yang dihadapi penderita buta warna kebanyakan adalah kesulitan untuk mendapatkan pekerjaan sesuai keinginan. Beberapa jenis pekerjaan mewajibkan para pekerja tidak buta warna, seperti pilot, pekerjaan yang berhubungan dengan elektronik, tentara, polisi dan pekerjaan yang berhubungan dengan kimia. Kesulitan dalam mendapatkan pekerjaan yang disebabkan karena individu menderita kelainan buta warna, sering kali membuat para penderita buta warna frustasi. Selain itu, juga ada dampak psikologis yang menyebabkan penderita tidak percaya diri dalam hal-hal tertentu.

2.2 Sistem Bantuan Penglihatan Penyandang Buta Warna

Buta warna tidak dapat disembuhkan, karena ini merupakan kelainan dan bukan sebuah penyakit. Oleh karena itu, diperlukan sesuatu yang dapat meminimalisasi dampak negatif dari kelainan buta warna, hal itu berupa alat bantu pengenal warna atau pembeda warna. Saat ini ada beberapa alat bantu pengenal warna atau pembeda warna. Alat bantu yang tersedia tersebut berupa lensa khusus dan perangkat lunak komputer personal.

Lensa khusus yang tersedia diklaim oleh pembuatnya dapat meningkatkan penglihatan penyandang buta warna. Lensa khusus tersebut berupa sebuah lensa biasa yang diberi tinta warna-warna tertentu. Lensa tersebut berfungsi mengubah panjang gelombang cahaya yang dipantulkan oleh objek-objek berwarna ke panjang gelombang yang dapat lebih mudah dilihat penderita buta warna. Walaupun berdasarkan testimoni mayoritas orang-orang yang telah sempat mencoba atau menggunakan lensa ini mengatakan bahwa penglihatan mereka terhadap warna menjadi lebih tajam tetapi mereka juga mengatakan bahwa mereka masih tidak percaya diri untuk menentukan nama warna dan membedakan warna. Alat ini mengubah spektrum warna dari warna yang biasa mereka lihat [11]. Lensa dibuat spesifik untuk penderita buta warna merah-

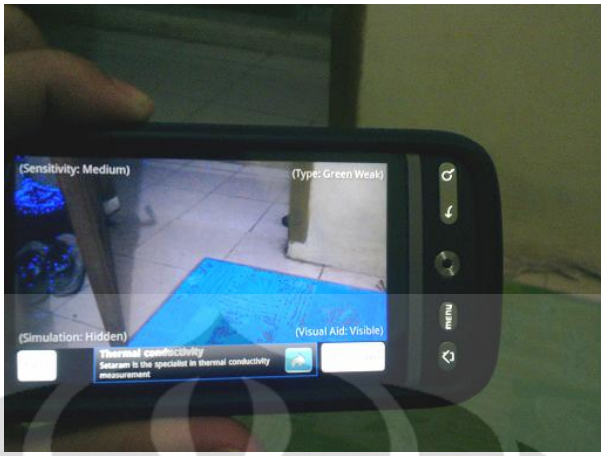
Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

hijau (*deuteranomaly*) yang merupakan jenis yang paling banyak ditemui. Salah satu perusahaan yang membuat lensa semacam ini adalah perusahaan *ChromaGen*.

Selain itu, terdapat jenis peralatan bantuan penglihatan untuk penderita buta warna berupa perangkat lunak di komputer personal (PC). Perangkat lunak ini memudahkan pengguna mengenal warna dan membedakan warna pada lingkungan *desktop* komputer personal mereka. Contoh penggunaannya seperti membantu pengguna untuk membedakan warna pada diagram warna di dokumen-dokumen. Salah satu perusahaan yang mengembangkan perangkat lunak semacam ini adalah *eyePilot* [18]. Limitasi dari perangkat lunak ini adalah, perangkat lunak ini hanya dapat membantu di lingkungan tampilan layar komputer personal.

Seiring berkembangnya teknologi informasi, terdapat beberapa sistem berbasis teknologi informasi dalam pengembangan aplikasi bergerak (*mobile application*) yang dapat dijalankan pada *smartphone* untuk membantu penyandang buta warna untuk membedakan warna-warna. Hanya saja algoritma yang diterapkan masih sederhana. Contohnya menambahkan tanda (*highlight*) di warna-warna tertentu. Contoh perangkat lunak yang menerapkan algoritma seperti ini adalah *Enliven* yang berbasis Android OS. Aplikasi ini tersedia secara gratis di *Android Market* [19]. Selain itu, terdapat aplikasi *Color Blind Aid* yang berbasis iOS dan dijual sekitar US$ 5 di *iTunes Store* [20].

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011



Gambar 2.4. Perangkat lunak bantuan buta warna *Enliven* yang berbasis pada Android OS memberi *highlight* warna biru pada objek hijau pada mode jenis buta warna lemah warna hijau

Pada skripsi ini dibahas rancang bangun alat bantu penglihatan untuk penyandang buta warna. Sistem yang dibangun berupa sebuah divais tunggal berupa *embedded system* (sistem tertanam) dengan konsep *augmented reality* (realitas tertambah) yang diharapkan dapat menjadi inovasi yang bermanfaat dalam alat bantu penglihatan bagi penyandang buta warna.

2.2.1 Manfaat Sistem Bantuan Penglihatan Penyandang Buta Warna

Berikut adalah manfaat yang diharapkan dengan adanya sistem bantuan penglihatan untuk penyandang buta warna :

1. Memudahkan kegiatan sehari-hari yang berkaitan dengan masalah warna.
2. Menurunkan ketergantungan kepada orang lain dalam kegiatan yang berkaitan dengan warna, misalnya dalam memilih warna baju.
3. Meningkatkan peluang untuk diterima atau menjalani profesi tertentu.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

2.2.2 Konsep Sistem Tertanam

Sistem tertanam adalah sebuah kombinasi perangkat keras dan perangkat lunak komputer yang dirancang untuk mengerjakan suatu fungsi khusus [3]. Sistem tertanam berbeda dengan perangkat komputer personal (PC) yang sering digunakan manusia. Komputer personal dibuat untuk kebutuhan yang bermacam-macam, tergantung kebutuhan pengguna. Pengguna dapat melakukan instalasi perangkat lunak apa saja dengan mudah sesuai kebutuhan. Pengguna juga dapat menambahkan divais *input output* (IO) dengan mudah, sehingga dapat disimpulkan komputer personal dirancang untuk kebutuhan yang universal, setiap orang dapat menggunakan personal komputer sesuai kebutuhan mereka masing-masing. Di sinilah letak perbedaan komputer personal dengan sistem tertanam. Sistem tertanam dirancang hanya untuk fungsi tertentu saja. Sistem yang berbasis *platform embedded* makin banyak digunakan pada era sekarang ini. Contoh sebuah sistem tertanam adalah telepon genggam, *microwave oven*, DVD *player*, Blu-ray *player*, dan lain-lain.

Dengan membuat suatu sistem dengan *platform embedded* maka suatu sistem dapat menjalankan suatu fungsi tertentu dengan lebih cepat dan lebih efektif. Sumber daya yang digunakan hanya untuk menjalankan suatu fungsi tersebut saja. Selain itu, sistem yang dibangun dengan *platform embedded* dapat dibuat dengan dimensi yang lebih ringkas sesuai dengan kebutuhan sistem tersebut. Dengan semakin berkembangnya perangkat-perangkat sistem tertanam pada level performansi yang dicapai dan keringkasan dimensi.

Pada skripsi ini akan dibahas sistem bantuan penglihatan untuk penyandang buta warna. Sistem berfungsi untuk memanipulasi data yang berupa bayangan objek menjadi sebuah informasi tambahan untuk para pengguna guna mengetahui warna benda atau membedakan warna-warna benda. Dengan karakteristik sistem berbasis *embedded* saat ini, diharapkan sistem bantuan penglihatan ini tepat guna dalam membantu penglihatan penyandang buta warna.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

2.2.3 Konsep *Augmented Reality*

Augmented Reality (AR) adalah sebuah konsep yang menggabungkan antara objek-objek di lingkungan nyata dengan objek-objek virtual yang dihasilkan oleh sebuah sistem komputer dengan tujuan menambah informasi pada objek nyata. Penambahan objek virtual tersebut biasanya dilakukan secara *real-time* dan dalam konteks yang berhubungan langsung dengan lingkungan sekitar yang berupa lingkungan nyata. Contoh konsep AR yang paling sederhana adalah tambahan objek virtual skor pertandingan pada siaran langsung pertandingan sepak bola di televisi. Dengan berkembangnya teknologi AR, informasi mengenai lingkungan nyata di sekitar kita menjadi lebih interaktif dan bermanfaat. Konsep AR pertama kali dikenalkan oleh Thomas Caudell pada tahun 1990, saat itu ia bekerja di perusahaan Boeing [4].

Berbagai aplikasi berbasis AR telah banyak dikembangkan di berbagai macam bidang, seperti aplikasi dalam bidang militer, pendidikan, arsitektur dan bidang-bidang lainnya. Contoh aplikasi AR dalam bidang militer mulai dikembangkan oleh angkatan militer Amerika Serikat, konsep AR diaplikasikan melalui sebuah helm yang biasa dipakai oleh tentara, helm tersebut mempunyai layar tambahan di depan mata pengguna guna menampilkan informasi tambahan (AR). Informasi tambahan yang diaplikasikan dalam bidang militer biasanya berupa informasi yang berguna untuk memberi pertolongan pertama pada tentara yang terluka, seperti informasi golongan darah [21]. Selain itu, konsep AR pada militer juga dapat menampilkan informasi strategi perang yang *real-time*, informasi keadaan sekitar, informasi jika ada keadaan darurat dan gambaran keseluruhan medan perang. Salah satu perusahaan yang membuat solusi-solusi sistem berbasis AR adalah *Arcane Technologies* [22].

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar 2.5. Contoh gambaran penggunaan AR pada bidang militer

Untuk menerapkan konsep AR dalam sebuah sistem/aplikasi diperlukan suatu komponen yang disebut *Augmented Reality Display*. Komponen ini berfungsi menampilkan objek tambahan (*augmented object*) di antara mata dari pengamat dan objek di lingkungan asli. Jenis divais yang biasa digunakan untuk menjalankan fungsi komponen ini ada 3 tipe, yaitu *Head-Attached Displays*, *Hand-Held Displays* dan *Spatial Displays*. *Head-Attached Display* adalah sebuah komponen *display* yang dapat dipakai pengguna di kepala mereka. Terdapat 3 tipe divais *Head-Attached Display*, yaitu *Retinal Display* yang dapat langsung memproyeksi gambar langsung ke retina mata dengan laser. Kedua, *Head-Mounted Display* yang mempunyai perangkat penampil (*display*) di depan mata pengguna. Terakhir, *Head-Mounted projector* yang mempunyai perangkat proyektor kecil yang dapat memproyeksikan gambar virtual ke objek di lingkungan asli [5].

Dengan terus berkembangnya dan semakin banyak digunakannya teknologi *augmented reality* yang memicu simplifikasi peralatan penunjang AR seperti *Head Mounted Display* (HMD) dalam hal bentuk dan harga, maka dalam skripsi ini dibuat suatu sistem bantuan penglihatan untuk penderita buta warna yang berbasis konsep AR. Input gambar dari kamera dapat dimanfaatkan untuk mendeteksi objek warna dan dijadikan objek tambahan (*augmented object*) yang melengkapi gambaran objek sebenarnya untuk ditampilkan menjadi sebuah kesatuan dengan antarmuka (*interface*) yang tepat guna dan nyaman.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar 2.6. Contoh penggunaan Head Mounted Display [13]

2.3 Penggunaan *Microsoft .NET Framework* sebagai *Framework* dalam Pengembangan Sistem Tertanam

Dalam pengembangan sistem tertanam, penulis akan menggunakan sebuah sistem operasi (OS) khusus yang dikembangkan oleh Microsoft untuk mengembangkan sistem dengan *platform* tertanam yang disebut *Windows Embedded* [23]. *Platform ini* dipilih untuk memudahkan pengembangan sistem bantuan penglihatan untuk penyandang buta warna menjadi sebuah aplikasi tertanam (*embedded application*). Terdapat 3 jenis *Application Programming Interface* (API) untuk mengembangkan sebuah perangkat lunak di *Windows Embedded*, yaitu Win32, MFC dan *.NET Framework*. *Framework .NET* digunakan sebagai API demi memudahkan pengembangan aplikasi. Terdapat dua jenis utama sistem operasi pada lini produk *Windows Embedded*, yaitu *Windows Embedded CE* dan *Windows Embedded Standard*.

2.3.1 *Windows Embedded CE*

Windows Embedded CE (*Compact Embedded*) adalah sistem operasi (OS) yang dikembangkan oleh perusahaan perangkat lunak raksasa *Microsoft*. *Windows Embedded CE* dirancang *Microsoft* khusus untuk mengembangkan sistem tertanam. *Windows Embedded CE* dapat dikembangkan untuk

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

mengembangkan divais bergerak, telepon seluler, telepon IP, divais multimedia, televisi, peralatan otomasi dan aplikasi-aplikasi lainnya. *Window Embedded CE* dirancang tidak hanya untuk dijalankan di atas satu arsitektur mikroprosesor saja. Arsitektur mikroprosesor yang dapat bekerja sama dengan *Windows Embedded CE* adalah ARM, MIPS, SH4 dan x86 [6]. Dilihat dari bermacam-macam arsitektur mikroprosesor yang dapat bekerja sama, sistem operasi *Windows Embedded CE* dinilai menawarkan fleksibilitas bagi pengembang divais *embedded* yang dapat menyesuaikan kebutuhan pengembangan divais *embedded* sesuai kebutuhan.

Selain fleksibilitas dalam memilih arsitektur mikroprosesor, para pengembang aplikasi di *Windows Embedded CE* juga fleksibel dalam menentukan modul-modul apa saja yang perlu disertakan ke dalam sistem mereka. Pengembang dapat menentukan sendiri desain sistem operasi, sebagai contoh suatu pengembang tidak membutuhkan fitur *networking* pada sistem yang akan dikembangkannya. Untuk menghemat sumber daya, pengembang dapat menghilangkan fitur *networking* tadi dari desain sistem operasi *Windows Embedded CE*. Untuk merancang modul-modul apa saja yang dibutuhkan untuk sistem yang akan dikembangkan dengan *Windows Embedded CE*, *Microsoft* membuat suatu *plug-in* untuk memudahkan pengembang dalam melakukan hal ini, *plug-in* ini dinamakan Platform Builder for *Windows Embedded CE*. *Platform Builder for Windows Embedded CE* dapat digunakan sebagai *plug-in Microsoft Visual Studio*, yaitu sebuah IDE (*Integrated Development Environment*) yang disediakan *Microsoft* untuk mengembangkan sistem dan aplikasi yang berkaitan dengan Microsoft, termasuk pengembangan sistem dan aplikasi berbasis *Windows Embedded CE*.

Windows Embedded CE mendukung teknologi terkini yang digunakan untuk pengembagan sistem, seperti:

1. Teknologi jaringan dan *wireless* : TCP/IP, IPv4, IPv6, IPsec, PAN, LAN, WAN, WLAN, SOAP, LDAP, RDP, VoIP, PPPoE, dll

2. Teknologi server : Telnet, FTP, SMB, RAS, PPTP, UPnP, Parental Control, Web Proxy, dll

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

3. Teknologi multimedia : DirectDraw, DirectShow, Direct3D Windows Media Player, WMA, MP3, IE, DVD Video API dan Digital Rights Management

4. Teknologi *storage* dan *file system* : FAT, TFAT, exFAT, CDFS, CEDB, dll

2.3.2 *Windows Embedded Standard 2009*

Windows Embedded Standard 2009 (WES2009) merupakan sebuah sistem operasi lini *Windows Embedded* yang terdiri dari komponen-komponen yang dapat diubah-ubah (*customized*) menjadi sebuah sistem operasi yang sesuai dengan aplikasi yang ingin dijalankan di atasnya. WES2009 mempunyai komponen-komponen yang persis sama dengan Windows XP Professional, hanya saja komponennya dapat diubah-ubah. Seperti halnya *Windows Embedded CE*, WES2009 ditujukan menjadi sebuah sistem operasi untuk keperluan menjalankan sistem fungsi tunggal. Perbedaannya terletak pada *Framework* yang dapat didukung oleh kedua sistem operasi tersebut dan komponen-komponen yang dapat didukung (*driver* perangkat keras atau perangkat lunak).

Pada *Windows Embedded CE, framework* yang dapat digunakan selain Win32 dan MFC adalah *.NET Compact Framework* yang merupakan versi padat (*compact*) yang memiliki *library* lebih ringkas daripada *.NET Framework.* Sedangkan, WES2009 mendukung *.NET Framework* yang memiliki *library* yang lebih lengkap. Dengan perbedaan kedua sistem operasi tertanam tersebut, dapat disimpulkan bahwa dengan menggunakan *Windows Embedded CE* kita dapat membuat suatu sistem operasi untuk aplikasi tertanam yang sangat ringan, baik dalam volume *file* maupun proses keseluruhan yang dijalankan. Dengan keterbatasan pengembangan aplikasi tertanam berbasis .NET yang tidak seluas WES2009, yang mengimplementasikan *.NET Framework* secara penuh.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

2.3.3 *Microsoft Visual Studio* dan *Windows Embedded Studio*

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* (IDE) yang dapat digunakan sebagai perangkat pengembangan sistem atau aplikasi yang berbasis teknologi Microsoft. Untuk mengembangkan aplikasi pada *Windows Embedded CE* digunakan sebuah jenis *project* khusus yang disebut sebagai *Smart Device Project*. Pada *Windows Embedded CE*, selain untuk mengembangkan aplikasi, *Microsoft Visual Studio* juga digunakan untuk merangcang *image* sistem operasi *Windows Embedded CE* yaitu dengan menggunakan sebuah *plugin* yang disebut *Windows Embedded CE Platform Builder*.

Untuk mengembangkan sistem tertanam yang berbasis *Windows Embedded Standard 2009* digunakan sebuah tipe *project* pengembangan aplikasi yang sama ketika mengembangkan aplikasi di dalam .*NET Framework* pada aplikasi *desktop* biasa. Tidak seperti pada *Windows Embedded CE*, perancangan *image* sistem operasi WES2009 tidak dilakukan pada *Microsoft Visual Studio*, melainkan dilakukan pada perangkat pengembang berbeda yang disebut *Windows Embedded Studio*.

Windows Embedded Studio merupakan perangkat pengembang yang khusus dikembangkan *Microsoft* untuk membuat, memilih dan memanipulasi komponen pada WES2009/*Windows XP Embedded* pada *run-time image* yang akan dibuat. Selain itu, perangkat ini juga membantu para pengembang membuat *image* dan mengimplementasikannya pada divais target [16]. Terdapat empat komponen utama perangkat pengembangan pada *Windows Embedded Studio*, yaitu *Target Analyzer, Target Designer, Component Database Manager* dan *Component Designer*.

Target Analyzer membantu para pengembang menganalisis divais target yang akan diimplementasikan WES2009 di dalamnya dari segi perangkat keras. Dengan menggunakan perangkat ini, pengembang tidak perlu repot menentukan *driver hardware* apa saja yang dibutuhkan agar WES2009 dapat berjalan dengan baik pada divais target. *Target Analyzer* yang dijalankan pada divais target pada

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

sistem operasi *Ms-Dos* ataupun *Windows XP SP2* akan mendeteksi dan mengumpulkan data semua perangkat keras yang terdapat pada divais target tersebut.

Perangkat ini akan menyimpannya pada sebuah file dengan ekstensi *pmq* yang dalam proses perancangan *run-time image* dapat langsung diimpor ke perangkat perancangan untuk memasukkan komponen *driver* perangkat keras pada divais target ke dalam rancangan *run-time image*. Perangkat utama yang kedua adalah *Target Designer, Target Designer* merupakan perangkat yang dapat digunakan pengembang untuk merancang *run-time image* sistem operasi WES2009. Pada perangkat ini pengembang dapat memasukkan komponen-komponen yang diperlukan ke dalam rancangan lalu membuatnya menjadi *run-time* image yang sesuai rancangan. Perangkat selanjutnya adalah *Component Database Manager*, perangkat ini digunakan untuk mengatur basis data komponen WES2009, seperti memasukkan komponen baru ke dalam basis data komponen WES2009. Perangkat utama terakhir dalam *Windows Embedded Studio* adalah *Component Designer*. Perangkat ini ditujukan untuk membuat komponen baru (*custom*) yang sebelumnya belum tersedia pada komponen awal WES2009, misalnya *driver* perangkat keras tertentu. Komponen baru yang dibuat dengan menggunakan *Component Designer* akan disimpan dalam sebuah *file* dengan ekstensi *sld* yang selanjutnya dapat dimasukkan ke dalam basis data komponen WES2009 dengan menggunakan perangkat *Component Database Manager*.

2.4 Penggunaan *Microsoft .NET* sebagai *Framework* dalam Pengembangan Aplikasi *Augmented Reality*

Seperti yang telah dijelaskan pada bagian sebelumnya, untuk mengembangkan aplikasi atau perangkat lunak pada *Windows Embedded* terdapat 3 jenis API yang dapat digunakan, yaitu Win32, MFC (*Microsoft Foundation Classes*) dan *.NET Compact Framework/.NET Framework*. Win32 dan MFC API adalah API yang digunakan untuk mengembangkan aplikasi *native-code* pada *Windows Embedded.* Aplikasi ini dikembangkan

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

dengan bahasa pemrograman tingkat yang lebih rendah dan spesifik untuk arsitektur perangkat keras tertentu**.** Aplikasi yang dikembangkan dengan *native-code* akan memiliki tingkat performansi yang paling tinggi, tetapi hanya kompatibel dengan satu jenis arsitektur perangkat keras saja sesuai dengan arsitektur apa yang dituju dalam pengembangan aplikasi *native-code* tersebut. Contohnya aplikasi *native-code* untuk arsitektur ARM dan untuk arsitektur x86. Aplikasi *native-code* pada *Windows Embedded* dikembangkan dengan bahasa pemrograman C/C++ [7].

Selain API untuk aplikasi *native-code*, terdapat juga API yang ditujukan untuk pengembangan aplikasi *managed-code*, yaitu aplikasi yang dikembangkan dengan menggunakan bahasa tingkat tinggi. API yang mendukung pengembangan aplikasi *managed-code* dalam pengembangan aplikasi *embedded* dengan *Windows Embedded CE* adalah *.NET Compact Framework* API. Pengembangan aplikasi pada WES2009 dilakukan menggunakan *.NET Framework* API. Penulisan program pada API ini dapat dilakukan dengan bahasa pemrograman Visual Basic dan C# dengan dukungan *.NET Compact Framework/.NET Framework Library*. Dengan menggunakan API untuk pengembangan aplikasi *managed-code*, pengembang diberi lingkungan yang efisien untuk mengembangkan suatu aplikasi *embedded* tanpa memikirkan kode level rendah (*low-level code*) [7].

Untuk melakukan pengembangan aplikasi *augmented reality* di perangkat *embedded* digunakan *Microsoft .NET Framework* sebagai API. Selain lebih mudah dalam melakukan penerapan algoritma program kedalam kumpulan kode-kode, *framework* ini juga menawarkan kompatibilitas yang baik antar sistem-sistem yang dibangun di dalam *framework* ini sehingga diharapkan memudahkan kami untuk mengembangkan suatu aplikasi *augmented reality* yang nantinya akan dijalankan di sebuah sistem tertanam yang juga dibangun dengan .NET Framework.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

2.4.1 Pemrosesan Citra dengan *C#, OpenCV* dan *EmguCV*

C# (C-Sharp) adalah bahasa pemrograman yang dikembangkan oleh *Microsoft* sebagai bagian dari *.NET Platform*. Bahasa pemrograman C# dibuat dengan dasar bahasa yang sudah ada dan sudah populer, yaitu C, C++ dan Java. Penulisan program dengan bahasa C# pada pengembangan aplikasi di *Windows Embedded* sama saja dengan menulis kode program dengan C# pada pengembangan aplikasi berbasis sistem operasi komputer personal (PC) Windows, seperti Windows XP, Vista, dan 7. C# dibuat dengan mengadaptasi fitur yang bagus dari setiap bahasa pemrograman tersebut ditambah dengan fitur baru lainnya. Berikut adalah fitur-fitur yang ditawarkan bahasa pemrograman C# bagi para pengembang perangkat lunak [8] :

1. Pemrograman Berorientasi Objek.
2. *Graphical User Interface* (GUI).
3. *Exception Handling.*
4. *Multithreading.*
5. Multimedia
6. *Database Processing.*
7. *Internet/Client-Server.*
8. *Distributed Computing.*

Dari beberapa fitur di atas ternyata C# tidak menyediakan fitur pemrosesan citra secara khusus. Ftur multimedia sebenarnya terdapat pada C# tetapi fungsi/*library* yang ada terbatas. Padahal pemrosesan citra menjadi suatu hal yang penting saat ini. Teknologi penglihatan komputer (*computer vision*) semakin banyak digunakan saat ini untuk berbagai macam hal, misalnya aplikasi deteksi gerakan, deteksi wajah, pengenalan wajah, pengenalan sidik jari, pengenalan warna, dan lainnya. Untuk memudahkan pengembang aplikasi dalam menerapkan fungsi pengolahan citra dengan bahasa pemrograman C#, para pengembang dapat menggunakan

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

fungsi/*library* tambahan pengolahan citra yang kompatibel dengan C# di *.NET Platform.*

Salah satu *library* pemrograman yang populer untuk pemrosesan citra adalah *OpenCV* (Open Computer Vision). *OpenCV* memiliki lisensi BSD, yaitu lisensi yang membebaskan pengembang untuk memanfaatkan produk lisensi untuk keperluan akademis ataupun keperluan komersial. *OpenCV* pertama kali dikembangkan oleh Intel. Dengan *OpenCV* pengembang dapat melakukan fungsi pemrosesan citra dengan mudah seperti segmentasi objek, tranformasi objek, *object tracking* dan fungsi-fungsi lainnya. *OpenCV* sangat populer di kalangan pengembang aplikasi berbasis penglihatan komputer (*computer vision*).

OpenCV dapat digunakan dengan bahasa pemrograman seperti C, C++ dan Phyton [9]. Untuk menggunakan *library OpenCV* dengan C# diperlukan suatu pembungkus (*wrapper*) yang memungkinkan *library OpenCV* dapat dipanggil menggunakan C#. Salah satu *wrapper* populer untuk memungkinkan *OpenCV* digunakan bersama-sama dengan *.NET Platform* adalah *EmguCV*. *EmguCV* memungkinkan para pengembang aplikasi berbasis platform .NET seperti C#, Visual C++, Visual Basic dan IronPhyton untuk menggunakan *OpenCV* sebagai *library* tambahan [10]. *EmguCV* dikembangkan menggunakan *Windows Communication Foundation* (WCF) yang disediakan *Microsoft* mulai dari *.NET Framework 3.5*. Untuk mengembangkan sistem bantuan penderita buta warna ini, dipilih operasi WES2009 agar dapat mendukung EmguCV.

2.4.2 *Microsoft Voice APIs*

Untuk menambah realitas tertambah yang berupa suara diperlukan sebuah API yang dapat mengubah *string* atau variabel lainnya ke dalam bentuk suara. Oleh karena itu, untuk memudahkan implementasi program *augmented reality* dengan suara pada *.NET Platform* dapat digunakan API dari *Microsoft* yang dapat menjalankan fungsi yang berhubungan dengan suara.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Terdapat beberapa API yang disediakan *Microsoft* untuk memudahkan pengembang aplikasi berbasis .NET untuk menambahkan fungsi yang berhubungan dengan suara pada aplikasinya, berikut jenis-jenis API yang ditawarkan :

1. *Speech Application Programming Interface* (SAPI), digunakan sebagai API aplikasi desktop dengan komunikasi perangkat level rendah.
2. *.NET system.Speech managed namespace*, digunakan untuk pengembangan aplikasi desktop dengan menggunakan bahasa-bahasa pemrograman di *.NET Platform.*
3. *Microsoft Unified Communication Managed* API 2.0, digunakan untuk pengembangan aplikasi di server untuk melayani interaksi komunikasi suara.
4. *Speech Server*, digunakan untuk pengembangan aplikasi *Windows Server*.
5. *Tellme Studio*, digunakan untuk pengembangan aplikasi berbasis VoiceXML yang mengikuti standar *World Wide Web Consortium* (W3C).

Pemilihan mana produk yang akan digunakan tergantung lingkungan pengembangan aplikasinya. Dari dua jenis produk yang disediakan untuk pengembangan aplikasi *desktop,* untuk penelitian penulis, penulis memilih untuk menggunakan *.NET system.Speech managed namespace*. Karena yang dikembangkan adalah aplikasi pada perangkat *embedded* (*WES2009*). Dan aplikasi serupa untuk perangkat *smartphone* yang akan berjalan di atas sistem operasi *Windows Phone 7*.

Berbeda dengan *Microsoft* SAPI yang lebih ditujukan untuk pemrograman level rendah berbasis *Component Object Model* (COM) dengan bahasa pemrograman C++, *Microsoft .NET System.Speech managed namespace* disediakan untuk pengembangan fitur suara dengan lebih mudah pada lingkungan yang telah diatur di atas *platform* .NET *(.NET managed environment*). Selain menawarkan pemrograman yang lebih mudah, .NET juga

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

menjamin kompatibilitas aplikasi agar bisa dijalankan di atas semua sistem operasi Windows yang memiliki versi .NET yang sama. *System.Speech managed namespace* ini disediakan *Microsoft* sejak .NET versi 3.0, dan *System.Speech managed namespace* ini sudah menyediakan semua metode untuk berinteraksi dengan suara, seperti *Speech Recognition, Speech Synthesis*, pengunaan *Grammar library*, dan *Speech event handler*.

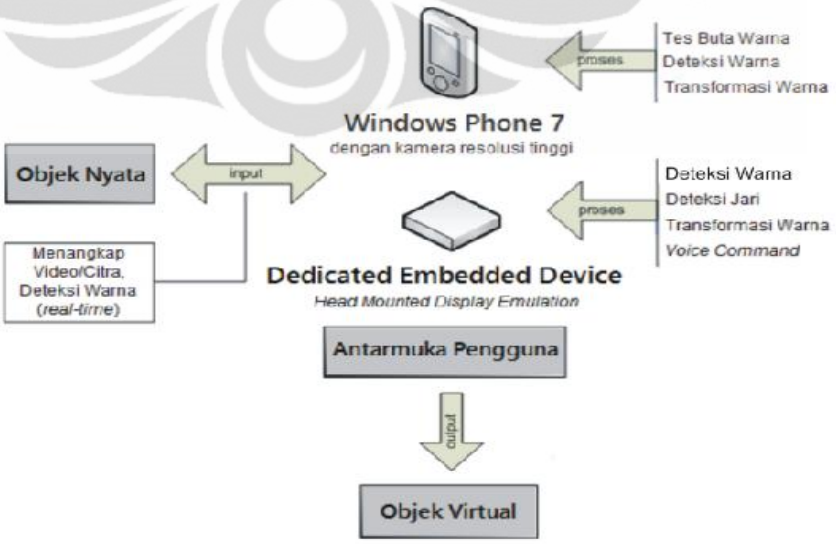
Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

BAB III

PERANCANGAN SISTEM BANTUAN PENGENAL WARNA DENGAN KONSEP *AUGMENTED REALITY*

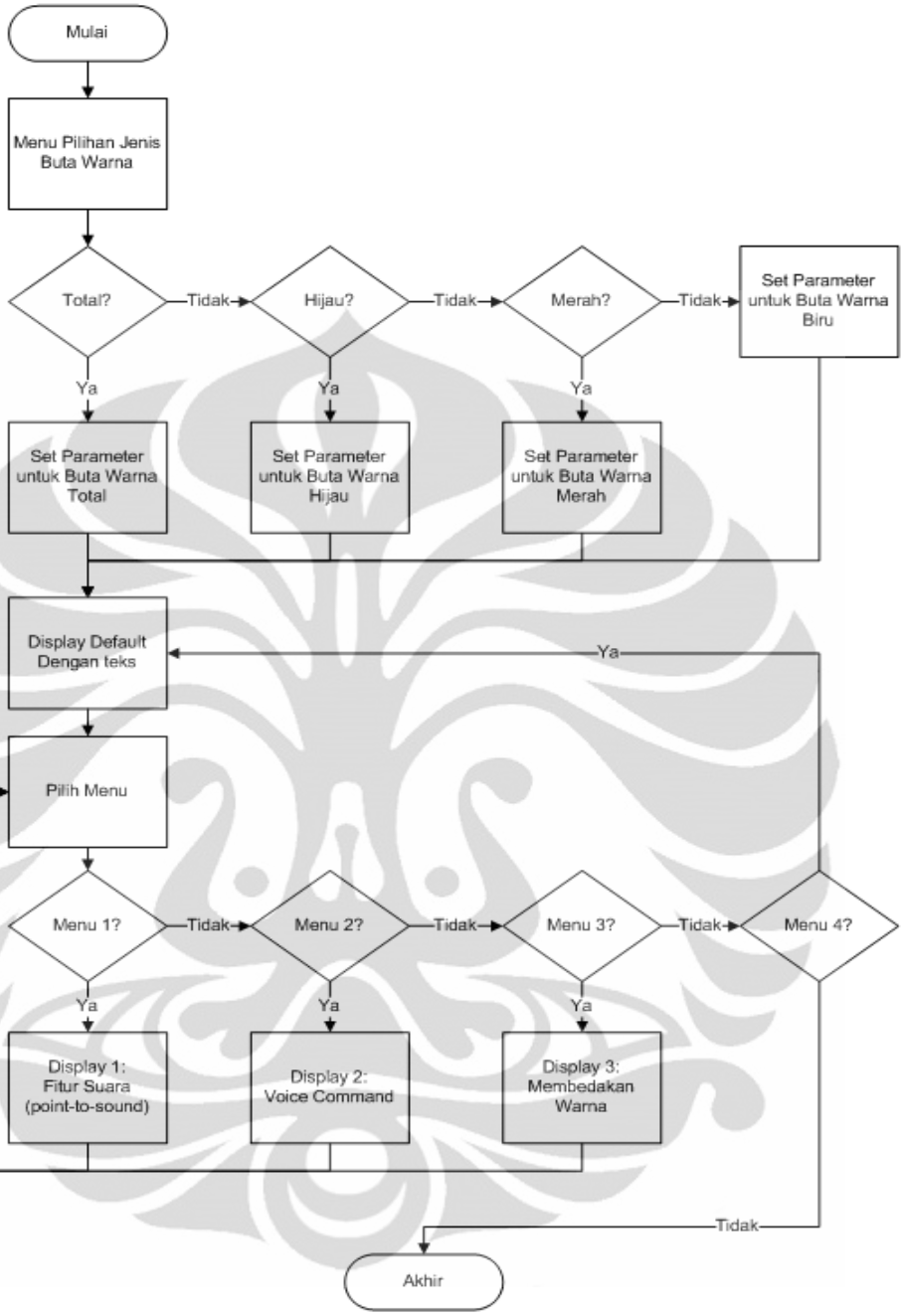
3.1 Deskripsi Umum Sistem

Sistem bantuan pengenal warna secara umum akan dibangun pada dua *platform*, yaitu sistem berbasis *Windows Embedded Standard 2009* (WES2009) dan sistem berbasis *Windows Phone 7*. Kedua sistem ini bertujuan menghasilkan keluaran berupa pengenalan warna dengan konsep realitas tertambah. Untuk memenuhi tujuan tersebut, dibangun sistem dengan berbagai fitur pengenal warna di dalamnya. Sistem ini mencakup beberapa fitur pengenal warna dengan algoritma yang berbeda-beda. Fokus penjelasan adalah pengembangan sistem bantuan pengenal warna pada divais tertanam (*embedded device*) yang berbasis WES2009 serta pembahasan fitur *point-to-sound augmented reality*. Untuk melihat gambaran umum sistem secara keseluruhan dapat dilihat sketsa dan diagram UML berikut ini.



Gambar 3.1. Gambaran umum sistem secara keseluruhan

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011



Gambar 3.2. Diagram alir umum sistem bantuan penglihatan untuk penyandang buta warna secara keseluruhan

Pada diagram alir di atas terlihat bahwa terdapat beberapa fitur yang akan dikembangkan pada sistem bantuan penglihatan untuk penyandang buta warna. Fitur yang akan dikembangkan adalah fitur fitur suara (*point-to-sound*), fitur teks/*highlight* denganvoice *command* yang dibahas pada [25] dan [26], serta fitur pembeda warna yang dibahas pada [24]. Fitur teks dengan *voice command* akan menampilkan tambahan objek virtual yang berupa teks nama warna berdasarkan objek asli yang ditangkap kamera. Teks nama ditampilkan pada objek dengan warna sesuai dengan jenis warna yang diperintahkan dengan menggunakan perintah suara.

Fitur kedua adalah fitur *point-to-sound*, fungsi ini akan memungkinkan pengguna untuk berinteraksi langsung dengan objek warna dan sistem akan mengeluarkan suatu sinyal suara berupa nama warna sesuai dengan objek warna yang ditunjuk pengguna. Selanjutnya adalah fitur *voice command*, fitur ini dimaksudkan untuk memudahkan pengguna untuk menemukan warna tertentu. Pengguna hanya perlu melakukan perintah suara (*voice command*) yang berupa nama warna, lalu sistem akan memberi suatu tanda (*highlight*) pada objek yang sesuai. Fitur yang terakhir adalah fitur pembeda warna, fitur ini digunakn untuk mempermudah pengguna mengenali warna tertentu dengan cara menaikkan saturasi warna tertentu, sesuai dengan jenis buta warna yang dialami pengguna.

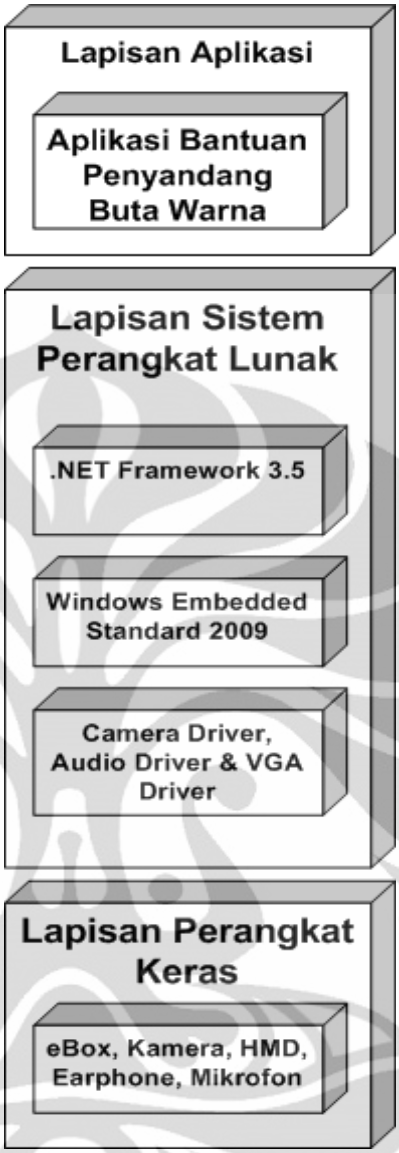
3.2 Arsitektur Sistem Tertanam

Arsitektur sistem tertanam yang akan dibangun akan meliputi tiga lapisan (*layer*) utama, yaitu lapisan perangkat keras (*hardware layer*), lapisan sistem perangkat lunak (*system software layer*) dan lapisan aplikasi *(application layer*). Lapisan perangkat keras meliputi perangkat-perangkat keras yang digunakan pada sistem ini, perangkat keras meliputi *eBox* yang berbasis arsitektur mikroprosesor x86, kamera video, mikropon dan *earphone*. *eBox* digunakan sebagai pusat pemrosesan data, sedangkan perangkat keras lainnya berperan sebagai divais masukan dan keluaran (I/O).

Lapisan selanjutnya adalah lapisan sistem perangkat lunak, lapisan ini terdiri dari tiga sub lapisan, yaitu perangkat lunak pengontrol (*driver*), sistem operasi, serta *Middleware*. Perangkat lunak pengontrol berfungsi untuk mengontrol kerja dari divais-divais I/O agar dapat bekerja dengan baik dengan pusat pemrosesan data. Sistem operasi adalah perangkat lunak yang mengatur kerja perangkat keras dan layanan-layanan (*services*) yang digunakan untuk menjalankan aplikasi agar aplikasi dapat berjalan dengan baik. Pada sistem ini akan digunakan sebuah sistem operasi khusus aplikasi tertanam untuk mengatur perangkat keras dan proses dengan baik pada sistem tertanam ini, sistem operasi yang digunakan merupakan sistem operasi khusus untuk sistem tertanam yang diproduksi oleh *Microsoft*, yaitu *WES2009*.

Komponen terakhir pada lapisan sistem perangkat lunak adalah *Middleware*, *Middleware* adalah sebuah sistem perangkat lunak yang bukan merupakan kernel OS, *driver* ataupun aplikasi. *Middleware* biasanya berperan sebagai mediator antara aplikasi perangkat lunak dengan kernel atau perangkat lunak *driver* dengan tujuan mereduksi kompleksitas aplikasi perangkat lunak, fleksibilitas, keamanan, portabilitas, konektivitas dan interoperabilitas antar aplikasi. Contoh *Middleware* adalah protokol jaringan TCP/IP dan mesin virtual seperti *Java Virtual Machine* (JVM) [12]. Pada sistem ini akan digunakan *Middleware* berupa *.NET Framework 3.5* yang kompatibel dengan sistem operasi yang akan digunakan, yaitu *Windows Embedded Standard 2009*.

Lapisan terakhir adalah lapisan aplikasi perangkat lunak, yaitu lapisan perangkat lunak yang akan berinteraksi dengan pengguna secara langsung. Aplikasi perangkat lunak sistem bantuan untuk penyandang buta warna terdapat pada bagian ini.



Gambar 3.3. *Deployment* diagram arsitektur sistem tertanam

3.3 Komponen Perangkat Keras

Dalam pengembangan sistem tertanam yang akan kami kembangkan, sistem akan terdiri dari komponen perangkat keras (*hardware*) berikut:

a. *eBox-3310A-MSJK*

eBox-3310A-MSJK adalah sebuah divais penghitung (komputer) yang berbasis mikroprosesor dengan arsitektur x86 [7]. Pada penelitian ini, *eBox* digunakan sebagai divais target yang bertindak sebagai pusat pemrosesan

sinyal dalam pengembangan sistem bantuan penglihatan untuk penyandang buta warna yang berbasis sistem tertanam dengan WES2009. Divais ini memiliki beberapa antarmuka (*interface*) I/O (input/output) sebagai berikut :

1. VGA Video Output
2. PS/2 Keyboard & Mouse Input
3. Antarmuka IDE
4. Slot CF (Compact Flash)
5. 3 x Antarmuka USB
6. 2 x Port Serial
7. Antarmuka Ethernet
8. Output Audio dan Input Mikrofon

b. *Head Mounted Display* (HMD)

Head Mounted Display adalah sebuah perangkat penampil yang digunakan di kepala. Perangkat penampil terdapat di depan salah satu mata ataupun pada kedua mata. Biasanya digunakan untuk aplikasi yang berbasis realitas tertambah atau untuk aplikasi hiburan multimedia biasa, seperti menyaksikan video/film & memainkan permainan komputer. Pada sistem ini divais HMD akan digunakan sebagai penampil kondisi asli lingkungan sekitar pengguna dan objek realitas tertambah.

c. Video Kamera Dijital

Video kamera dijital adalah sebuah divais yang berfungsi untuk menangkap gambar bergerak yang terdiri dari bingkai-bingkai (*frames*) citra 2 dimensi yang disimpan atau diproses secara dijital. Dalam sistem bantuan penglihatan untuk penyandang buta warna, divais ini akan

digunakan sebagai divais input yang akan memberikan gambar bergerak dalam format dijital. Selanjutnya akan diproses di pusat pemrosesan data untuk ditampilkan dan diolah lebih lanjut.

d. *Earphone*

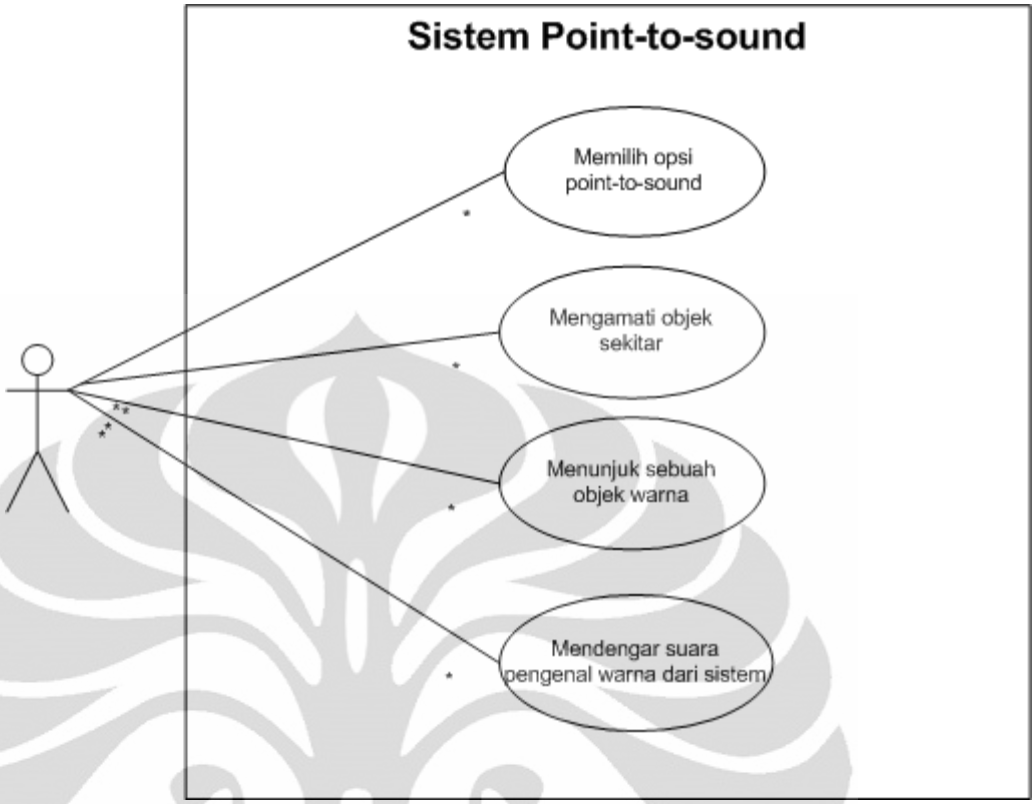
Earphone adalah sebuah divais output yang dapat mengubah sinyal elektrik menjadi sinyal suara. Pada sistem bantuan penglihatan untuk penyandang buta warna ini, *earphone* digunakan sebagai divais *output* yang akan menghasilkan objek suara yang merupakan objek tambahan realitas.

e. Mikrofon

Mikrofon adalah sebuah divais input yang dapat mengubah sinyal suara menjadi sinyal elektrik. Pada sistem ini, mikrofon akan digunakan sebagai divais input yang akan menerima sinyal masukan dari pengguna yang berupa suara. Selanjutnya sistem akan melakukan proses lebih lanjut di pusat pemrosesan data.

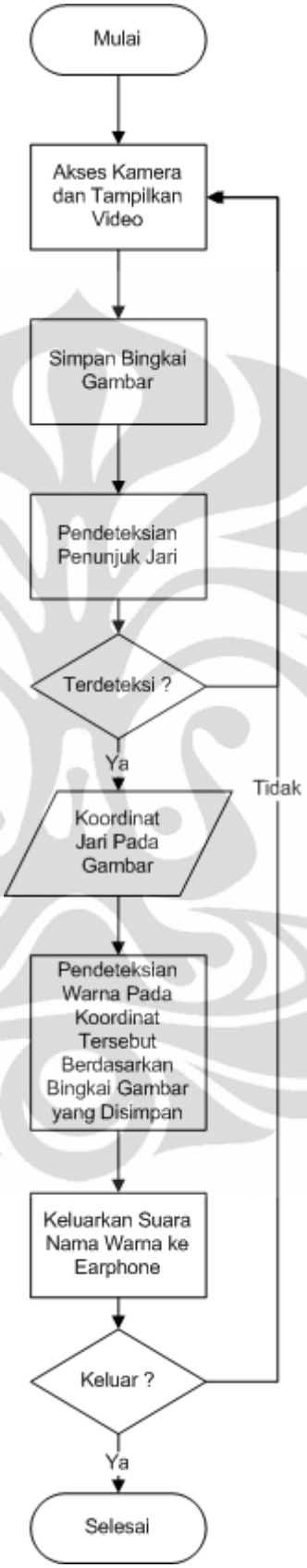
3.4 *Point-to-Sound Augmented Reality*

Point-to-Sound Augmented Reality adalah salah satu algoritma yang dengan konsep realitas tertambah yang dirancang sebagai salah satu fitur dalam implementasi sistem bantuan penglihatan untuk penyandang buta warna. Algoritma ini menawarkan interaksi langsung antara pengguna dan objek warna untuk menghasilkan tambahan realitas yang berupa suara. Pengguna akan mengetahui nama suatu warna dengan cara menunjuk ataupun menyentuh suatu objek warna. Sistem akan melakukan proses untuk menghasilkan suatu objek realitas tertambah yang berupa sinyal suara. Untuk penjelasan lebih lanjut mengenai cara kerja atau algoritma sistem dapat dilihat pada beberapa UML pada Gambar 3.4 berikut.



Gambar 3.4. *Use Case* Diagram

Gambar 3.4 merupakan diagram *use case* dari subsistem sistem bantuan Penglihatan untuk penyandang buta warna, yaitu diagram *use case* yang menunjukkan interaksi pengguna dengan sistem saat menggunakan fitur *point-to-sound*. Pertama, pengguna harus memilih fitur *point-to-sound* pada sistem, Setelah masuk ke tampilan fitur *point-to-sound* pengguna dapat secara fleksibel melihat-lihat objek sekitar saja ataupun menunjuk suatu objek warna. Jika pengguna ingin mengetahui warna tertentu dan sistem akan bekerja dengan memprosesnya menjadi suatu objek tambahan berupa suara yang menyebutkan nama warna yang dimaksud oleh pengguna.



Gambar 3.5. Diagram alir sistem *point-to-sound*

Gambar 3.5 merupakan diagram alir (*flow chart*) yang menunjukkan algoritma dasar program pada fitur *point-to-sound*. Awalnya, sistem menampilkan tampilan awal sistem. Tampilan awal hanya berupa gambar bergerak *real-time* yang ditangkap oleh kamera dijital. Selagi menampilkan antarmuka berupa video *real-time*, sistem juga akan menyimpan setiap bingkai (*frame*) gambar yang ditangkap kamera guna kepentingan proses selanjutnya, yaitu pendeteksian warna di koordinat tertentu.

Selanjutnya, sistem tidak akan melakukan proses lanjutan sebelum ada jari pengguna yang terdeteksi oleh sistem. Jika sistem mendeteksi jari pengguna, sistem akan melakukan proses berikutnya, yaitu sistem akan melakukan kalkulasi posisi jari pengguna yang terdeteksi di depan kamera. Posisi jari yang dihasilkan berupa koordinat gambar. Dari koordinat gambar yang menunjukkan posisi jari tersebut dan bingkai gambar sebelumnya yang telah disimpan oleh sistem, sistem dapat menggunakan kedua informasi tersebut untuk menentukan warna apakah yang ditunjuk oleh pengguna. Setelah informasi warna didapat, sistem akan melakukan proses berikutnya, yaitu sistem akan mengubah informasi nama warna tersebut ke bentuk suara yang dapat didengar oleh pengguna menggunakan *earphone*.

3.5 Proses Alir Data pada Sistem

Aliran data dalam fitur *point-to-sound augmented reality* akan melibatkan 3 perangkat keras, yaitu video kamera dijital, pusat pemrosesan data (eBox) dan *earphone*. Pertama, video kamera dijital akan menangkap gambar pada lingkungan sekitar dan langsung ditransmisikan ke pusat pemrosesan data secara *real-time*. Selanjutnya dilakukan pemrosesan citra oleh pusat pemrosesan data, untuk mengetahui apakah jari pengguna terdeteksi di depan kamera, jika jari terdeteksi maka pusat pemrosesan data akan melakukan pemrosesan lebih lanjut untuk mengetahui objek warna apakah yang ditunjuk oleh pengguna. Jika proses pemrosesan citra tersebut selesai, sistem akan

melanjutkan ke pemrosesan suara, yaitu mengubah nama objek warna yang telah diketahui menjadi objek suara dan dikeluarkan ke *earphone*.

Gambar 3.6. Diagram aliran data

3.6 Pemrosesan Sinyal

Pemrosesan sinyal yang akan dilakukan pada algoritma *point-to-sound augmented reality* berupa pemrosesan data *string* ke bentuk suara dan pemrosesan citra. Pemrosesan data *string* ke dalam bentuk suara akan dapat dilakukan dengan mudah dengan fungsi atau *method* yang telah tersedia pada *.NET system.Speech managed namespace* API, yaitu dengan *method SpeakAsync()* yang memiliki parameter berupa objek *string* dan akan melakukan *return* data berupa objek suara.

Pemrosesan sinyal yang paling kompleks pada algoritma ini adalah pemrosesan citra. Pemrosesan citra pada algoritma ini akan dilakukan untuk menentukan posisi jari pengguna pada lingkungan asli yang ditangkap oleh kamera serta menentukan nama warna sesuai dengna posisi jari yang terdeteksi. Pemrosesan citra ini akan dilakukan dengan menggunakan bantuan *library* khusus untuk fungsi-fungsi pemrosesan atau rekayasa citra seperti yang telah dibahas pada bagian dasar teori.

Terdapat dua metode pemrosesan citra yang dapat dilakukan untuk mendeteksi posisi jari pengguna berada. Metode pertama dilakukan dengan melakukan pendeteksian jari berdasarkan morfologi (bentuk) dan warna jari tangan. Kelebihan metode ini adalah sistem sangat *user-friendly*. Tetapi

kekurangannya adalah algoritma sulit diterapkan, tingkat ketepatan deteksi sangat tergantung dengan lingkungan sekitar dan beban proses yang lebih berat pada pusat pemrosesan data. Metode yang kedua dilakukan dengan melakukan pendeteksian serta *tracking* terhadap suatu marka yang unik dan mudah dideteksi oleh algoritma pendeteksi. Kelebihan metode ini adalah algoritma lebih mudah diterapkan, tingkat ketepatan pendeteksian baik, dan beban proses yang rendah. Kekurangan metode ini adalah sistem menjadi kurang *user-friendly*, karena diperlukan sebuah marka unik yang telah ditentukan sebelumnya yang harus diletakkan diujung jari pengguna setiap akan menggunakan fitur ini.

BAB IV

IMPLEMENTASI SISTEM

4.1 Perancangan Windows Embedded Standard 2009 Run-Time Image

Seperti yang telah dijelaskan pada bab sebelumnya, bahwa sistem akan dikembangkan di suatu *platform* tertanam (*embedded*), maka diperlukan sebuah sistem operasi *embedded* yang dapat dikustom dan dapat dijalankan di eBox-3310. Sistem operasi yang digunakan adalah *Windows Embedded Standard 2009*. Untuk merancang *Run-Time Image Windows Embedded Standard 2009* (WES2009) ada beberapa hal yang harus diperhatikan, yaitu:

1. Dukungan penuh terhadap perangkat keras yang akan menjalankan WES2009
2. Dukungan penuh terhadap aplikasi/perangkat lunak terbenam yang akan dijalankan di atas WES2009

Berdasarkan dua hal di atas, diperlukan analisis terkait perangkat keras yang perlu didukung oleh WES2009 pada eBox, untuk menentukan *driver* apa saja yang dibutuhkan untuk dimasukkan ke dalam *Run-Time Image WES 2009*. Selain itu juga diperlukan analisa terkait dukungan apa saja yang diperlukan untuk dimasukkan ke dalam Run-Time Image WES2009 untuk menjalankan aplikasi terbenam yang dikembangkan.

Analisa terkait perangkat keras dapat dilakukan dengan bantuan salah satu perangkat yang termasuk dalam perangkat pengembang yang disediakan oleh Microsoft dalam *Windows Embedded Studio*. Untuk menentukan *driver* perangkat keras apa saja yang dibutuhkan oleh eBox-3310, digunakan perangkat *Target Analyzer*. *Target Analyzer* bekerja dengan mendeteksi perangkat keras yang terdapat pada divais target, yaitu eBox-3310. Selanjutnya spesifikasi perangkat

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

keras yang terdeteksi pada divais target akan disimpan ke dalam sebuah file yang dapat digunakan untuk menambahkan komponen *driver* pada rancangan/konfigurasi *Run-Time Image WES2009* [14].

Pada skripsi ini, digunakan perangkat *Target Analyzer* pada eBox-3310 melalui *Windows XP SP3* yang dijalankan melalu *Live* USB. Proses ini menghasilkan sebuah *file* (.pmq) yang berisi instruksi kepada perangkat *Target Designer* untuk menyertakan komponen *driver* perangkat keras yang dibutuhkan oleh eBox-3310. Pada kasus eBox 3310, terdapat komponen *driver* yang diperlukan tetapi tidak tersedia dalam basis data komponen WES2009. Oleh karena itu, diperlukan file tambahan (.sld) untuk mendukung perangkat keras tersebut. Dalam kasus ini, pengembang eBox telah menyediakan file tersebut, sehingga komponen *driver* perangkat keras tersebut dapat langsung disertakan ke dalam basis data komponen WES2009 melalui perangkat *Component Database Manager*. Dengan langkah-langkah yang telah dilakukan di atas, semua *driver* perangkat keras yang dibutuhkan telah disertakan dalam konfigurasi *Run-Time Image WES2009*.

Selanjutnya analisis harus dilakukan terkait perangkat lunak yang akan dijalankan di atas WES2009. Sesuai dengan yang dijelaskan pada bab sebelumnya, perangkat lunak akan dikembangkan pada *.NET Framework* versi 3.5. Terkait hal ini, harus dilakukan konfigurasi *Run-Time Image WES2009* yang dapat mendukung penuh perangkat lunak pada *.NET Framework 3.5*. Untuk itu, komponen *.NET Framework 3.5* harus disertakan pada konfigurasi melalui perangkat *Target Designer*. Komponen *Speech API* (SAPI) juga disertakan pada konfigurasi ini karena diperlukan dalam pengembangan sistem. Konfigurasi berdasarkan kebutuhan perangkat keras dan perangkat lunak selesai, selanjutnya perlu dilakukan persiapan untuk membuat dan instalasi *Run-Time Image WES 2009* pada divais target.

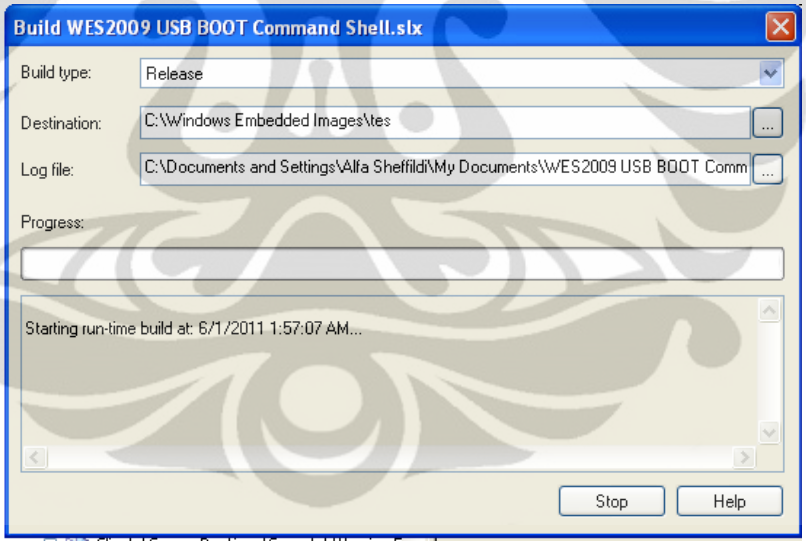
4.2. Pembuatan dan Instalasi Windows Embedded 2009 Run-Time Image

Setelah komponen pendukung perangkat keras dan perangkat lunak yang dibutuhkan telah semuanya disertakan dalam perancangan *Run-Time Image*, perlu

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

dilakukan pemeriksaan ketergantungan (*dependencies check*) terhadap komponen-komponen WES2009 yang telah disertakan ke dalam rancangan *Run-Time Image* dan pembuatan *Image* itu sendiri (*build image*). Pemeriksaan ketergantungan dilakukan untuk memeriksa apakah komponen-komponen yang disertakan dalam rancangan *Run-Time Image* memiliki ketergantungan dengan komponen lain atau tidak. Jika komponen tertentu memiliki ketergantungan terhadap komponen lain, maka komponen baru harus ditambahkan ke dalam rancangan *Run-Time Image*. Hal ini harus dilakukan agar setiap komponen awal yang disertakan ke dalam rancangan *Run-Time Image* dapat bekerja dengan seharusnya.

Proses *dependencies check* akan secara otomatis menambahkan komponen yang diperlukan ke dalam konfigurasi. Setelah itu, *Run-Time Image* siap untuk dibuat. Perangkat *Target Designer* akan membuat *Run-Time Image* dan disimpan ke dalam sebuah direktori.



Gambar 4.1 Proses Pembuatan WES2009 *Run-Time Image*

Setelah proses pembuatan *Run-Time Image* selesai, semua *file* yang dibutuhkan untuk menjalankan WES2009 di eBox-3310 akan tersedia dalam direktor tersebut, *file Run-Time Image* yang dihasilkan dari percobaan kali ini sekitar 300MB. Akan tetapi, saat dilakukan instalasi *image* tersebut pada *Internal Storage* (penyimpan internal) eBox-3310, ternyata penyimpan internal eBox-3310 tidak mampu untuk menampung sistem operasi WES2009 yang dihasilkan dari

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

proses perancangan. Hal ini terjadi karena pada saat proses instalasi WES2009 *First Boot Agent* (FBA) melakukan instalasi dan registrasi semua perangkat yang diperlukan (*driver* perangkat keras, perangkat-perangkat lunak dan *.NET 3.5 Framework*). Ternyata proses tersebut menghasilkan beban sebesar ~700MB, beban paling tinggi dihasilkan oleh instalasi *.NET Framework 3.5*. Oleh karena beban volume data tersebut melebihi kapasitas penyimpan internal *eBox-3310* (sekitar 500MB) tidak dimungkinkan untuk menjalankan WES2009 pada penyimpan internal *eBox-3310*.

Oleh karena itu, penulis memilih opsi melakukan instalasi WES2009 pada USB *Flash Drive* yang berkapasistas sekitar 4GB. Untuk melakukannya, diperlukan tambahan komponen yang harus disertakan dalam konfigurasi rancangan *Run-Time Image*, Komponen tersebut adalah *USB Boot 2.0* dan *Runtime Quick Start Helper*. Setelah kedua komponen tersebut dimasukkan ke dalam konfigurasi rancangan *Run-Time Image*, *image* WES2009 harus dibangun ulang. Selain itu, *USB Flash Drive* juga harus dipersiapkan agar mempunyai *Master Boot Record* (MBR) agar bisa melakukan *boot* WES2009 [14]. Untuk menulis MBR sekaligus melakukan format pada *USB Flash Drive* yang dituju, digunakan perangkat *UFDPrep* yang disediakan dalam perangkat *Windows Embedded Studio*. Selanjutnya, dengan mengubah konfigurasi *boot drive* pada *Basic Input Output System* (BIOS) eBox-3310 menjadi *USB Flash Drive* dan memulai *booting* dari *USB Flash Drive* yang telah dipersiapkan, WES2009 berjalan dengan sempurna yang didahului instalasi dan registrasi komponen WES2009 oleh FBA.

4.3. Implementasi Fitur *Point-to-Sound*

Implementasi fitur *Point-to-Sound* pada sistem ini dilakukan dengan cara mengimplementasikan algoritma deteksi ujung jari dengan menggunakan *OpenCV* dan *EmguCV*. Deteksi ujung jari dilakukan untuk mendapatkan posisi ujung jari pengguna dalam koordinat 2 dimensi pada citra waktu-nyata. Selanjutnya, koordinat tersebut digunakan untuk menentukan warna dari sebuah objek yang ditunjuk oleh pengguna.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Terdapat beberapa tahapan yang penulis lakukan untuk mendeteksi ujung jari dengan menggunakan bantuan *library OpenCV* dan *EmguCV*. Berikut adalah tahapan yang diimplementasikan untuk mendeteksi ujung jari berdasarkan beberapa referensi jurnal:

1. Klasifikasi objek kulit
2. Ektraksi kontur tangan
3. Analisa kontur tangan
4. Penentuan posisi ujung jari

Pada bagian selanjutnya akan dibahas masing-masing tahapan yang penulis lakukan untuk mendeteksi ujung jari dan mendapatkan posisinya dalam koordinat gambar.

4.3.1. Klasifikasi Objek Kulit

Proses klasifikasi objek kulit dilakukan untuk mendapatkan objek awal yang diharapkan berupa proyeksi objek tangan yang ditangkap kamera yang akan diproses selanjutnya dalam penentuan posisi ujung jari. Proses ini akan mengklasifikasikan objek tangan dan menghasilkan keluaran citra skala abu (*grayscale*). Objek yang dianggap sebagai representasi objek tangan akan diberi nilai maksimal dalam skala 8 bit, yaitu 255 (putih) pada keluaran citra *grayscale* hasil proses ini, sedangkan objek bukan tangan akan diberi nilai 0 (hitam). Klasifikasi objek kulit yang dilakukan pada kesempatan kali ini adalah klasifikasi objek kulit berdasarkan warna yang ditangkap oleh kamera.

Terdapat beberapa format warna yang populer digunakan untuk mengklasifikasikan warna kulit, salah satunya adalah format warna HSV (*Hue, Saturation, Value*) dan format warna YCbCr. Format warna RGB (*Red, Green, Blue*) yang populer dalam berbagai aplikasi komputer seperti pencetakan (*printing*) dan penampilan gambar (*display*) sulit diadaptasikan untuk memodelkan warna kulit karena memiliki keterkaitan yang tinggi antar tiga

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

komponen warna tersebut (Red, Blue dan Green) [15]. Format HSV dan YCbCr sering digunakan untuk mengklasifikasikan warna karena warna kulit dapat dimodelkan dengan komponen H dan S, serta Cb dan Cr. Selain itu HSV dan YCbCr bersifat *luminance independent,* sehingga warna tidak terlalu diperngaruhi oleh tingkat pencahayaan [15].

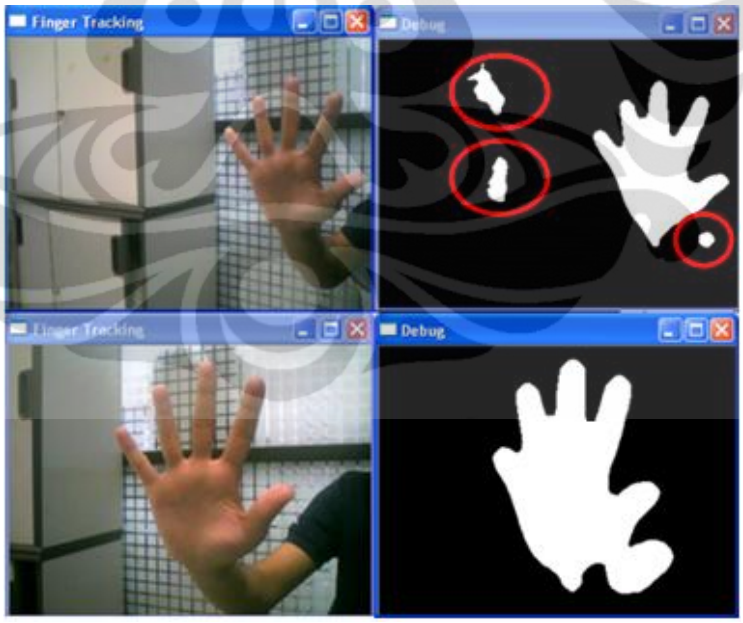
Oleh karena itu, pada skripsi ini dicoba mengimplementasikan permodelan warna kulit berdasarkan warna dengan format warna HSV dan YCbCr. Permodelan warna kulit dilakukan dengan cara memeriksa setiap pixel yang ditangkap oleh kamera atau sering disebut dengan metode *explicit thresholding*. Mengingat kamera menghasilkan gambar dengan menggunakan format warna RGB, maka diperlukan konversi gambar yang ditangkap ke format warna yang akan digunakan untuk mengklasifikasikan objek kulit, yaitu HSV dan YCbCr. Berikut adalah nilai *threshold* yang penulis gunakan untuk mengklasifikasikan atau memodelkan warna kulit untuk kedua jenis format warna tersebut.

- HSV : $H \leq 50$ & $S \geq 0.23$ & $S \leq 0.68$ [17]
- YCbCr : $Cb \geq 77$ & $Cb \leq 127$ & $Cr \geq 133$ & $Cr \leq 173$ [17]

Dengan menggunakan batas-batas nilai tersebut, setiap *pixel* pada gambar yang telah diubah kedalam format warna tersebut akan diperiksa apakah *pixel-pixel* tersebut masuk ke dalam kondisi batas-batas warna tersebut. Jika, *pixel* memenuhi kondisi tersebut, maka *pixel* akan dianggap sebagai objek kulit dan diberi nilai 255 dalam format *grayscale*. Sebaliknya, jika *pixel* tidak memenuhi kondisi tersebut, maka *pixel* tersebut dianggap sebagai objek bukan kulit dan diberi nilai 0 dalam format *grayscale*.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Selanjutnya, untuk meminimalisir derau/*noise* dan mempertegas bentuk objek kulit yang dideteksi (dalam hal ini bentuk tangan) dilakukan algoritma *smoothing*, dilasi dan erosi pada citra *grayscale* tersebut. Meminimalisir derau dan mempertegas objek sangat penting pada deteksi ujung jari untuk kasus ini, karena deteksi ujung jari yang penulis lakukan dalam kasus ini menggunakan pendekatan morfologi objek. Semakin bagus morfologi representasi objek kulit pada citra *grayscale* tersebut dibandingkan objek sebenarnya. Bentuk representasi objek kulit mendekati bentuk sebenarnya., maka deteksi ujung jari akan semakin baik dan akurat. Sebaliknya, jika algoritma program gagal menghasilkan bentuk objek pada citra *grayscale* mendekati bentuk aslinya, maka akan didapat banyak kesalahan deteksi. Dengan alasan tersebut, dapat disimpulkan bahwa proses klasifikasi kulit ini merupakan proses yang sangat penting dalam algoritma deteksi ujung jari ini. Setelah objek kulit didapatkan dan disimpan dalam citra *grayscale*, proses dilanjutkan dengan analisis kontur yang akan dibahas pada bagian selanjutnya.



Gambar 4.2. Hasil Klasifikasi dengan Menggunakan Format Warna HSV (baris atas) dan Hasil Klasifikasi dengan Menggunakan Format Warna YCbCr (baris bawah)

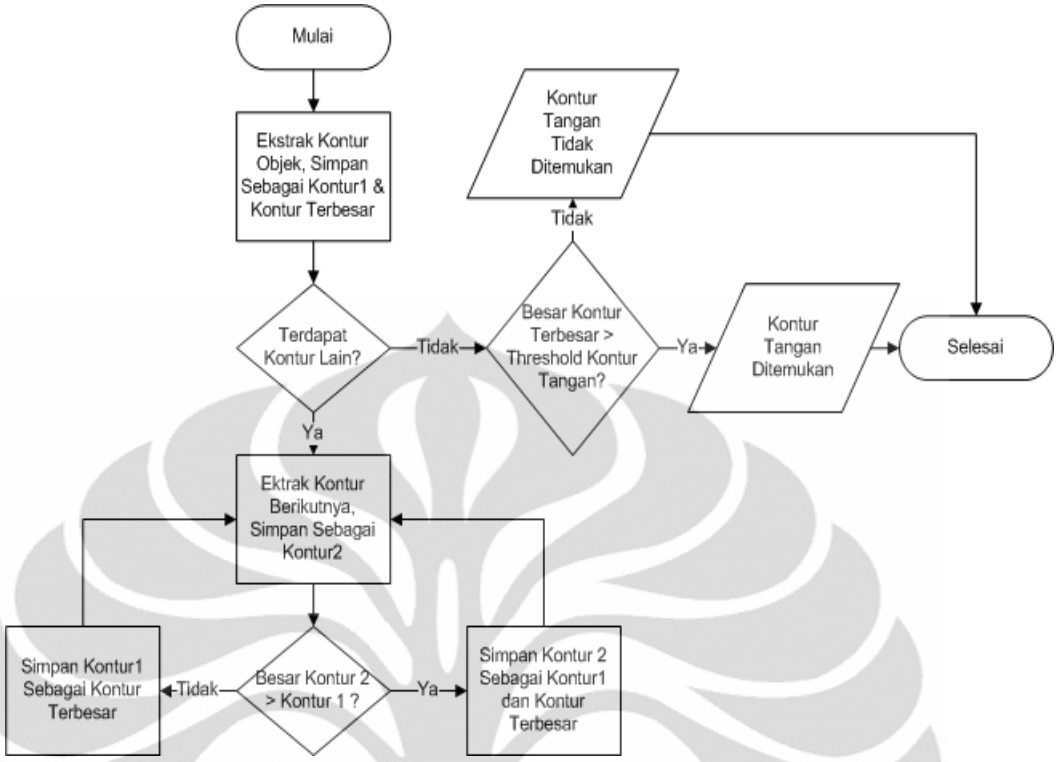
Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

4.3.2 Ekstraksi Kontur Tangan

Dengan menggunakan citra *graycale* yang hanya memiliki nilai maksimum (255) dan nilai minimum (0) dilakukan ekstraksi kontur dari objek yang dideteksi sebagai objek kulit (tangan). Adapun kontur yang diekstrak merupakan *linkedlist* dari elemen-elemen poin (terdiri dari posisi pixel dalam koordinat 2D) yang saling terhubung membentuk kontur. Untuk mendapatkan kontur pada OpenCV dan EmguCV, digunakan fungsi *cvFindContour* yang akan mengembalikan nilai berupa kontur-kontur objek yang dihasilkan dari citra *grayscale*. Kontur yang dihasilkan didapat berdasarkan batas terluar (*exterior*) dari objek-objek pada citra yang terpisah secara implisit antara wilayah yang positif (dalam kasus ini wilayah berwarna putih) dan wilayah yang negatif (dalam kasus ini wilayah berwarna hitam).

Untuk mendapatkan kontur yang diinginkan, yaitu kontur tangan, dilakukan komparasi kontur-kontur yang didapat dari citra tersebut. Kontur-kontur tersebut dibandingkan satu dengan yang lainnya berdasarkan ukuran masing-masing kontur tersebut, membandingkan ukuran kontur di sini dimaksudkan untuk menghilangkan derau yang muncul pada citra hasil keluaran proses klasifikasi kulit. Seperti terlihat pada Gambar 4.2 (marka lingkaran merah), pada citra tersebut tedapat derau yang berupa pixel-pixel berwarna putih selain pixel-pixel yang merepresentasikan objek tangan. Dengan asumsi kontur yang dihasilkan dari objek tangan relatif lebih besar dibandingkan dengan kontur objek derau. Dengan membandingkan besar kontur objek-objek tersebut dan hanya mengambil kontur yang paling besar, efek derau dapat diminamilisir. Selain itu juga dilakukan *thresholding* (pembatasan dengan nilai) berdasarkan ukuran kontur, yaitu berdasarkan estimasi ukuran kontur tangan.

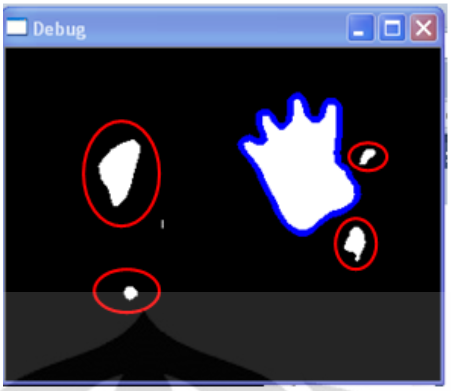
Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011



Gambar 4.3. Diagram Alir Proses Ektraksi Kontur Tangan

Dari proses ekstraksi kontur ini, diharapkan sistem dapat menemukan satu kontur yang merupakan kontur tangan yang akan dianalisa lebih lanjut berikutnya untuk menyimpulkan posisi koordinat dari ujung jari. Hasil ekstraksi kontur tangan dapat terlihat pada Gambar 4.4 di bawah ini. Pada gambar tersebut terlihat kontur yang dihasilkan digambar ke kontur tangan yang terdeteksi (garis biru) yang merupakan kontur tangan yang sebenarnya, sedangkan derau-derau (lingkaran merah) yang terdapat pada citra tersebut diabaikan untuk proses selanjutnya.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011



Gambar 4.4. Citra Hasil Ekstraksi Kontur Tangan

4.3.3. Analisis Kontur Tangan

Untuk mendapat posisi jari dari kontur tangan yang dihasilkan dari proses sebelumnya, perlu dilakukan analisa dari kontur tersebut. Analisis kontur dilakukan berdasarkan bentuk (morfologi) dari kontur, yang didapat dari proses ekstraksi kontur tangan. Sebelum dibahas lebih lanjut, perlu dipaparkan bahwa dengan alasan analisis posisi jari dilakukan berdasarkan morfologi dari kontur tangan, kesempurnaan bentuk kontur tangan yang didapat dari proses sebelumnya merupakan faktor yang sangat penting untuk hasil analisa posisi jari yang akurat pada kontur tangan.

Analisis kontur tangan dilakukan dengan cara memeriksa kontur cembung (*convex*) yang terdapat pada kontur tangan. Untuk menganalisis kontur cembung dengan menggunakan OpenCV, digunakan fungsi *cvConvexHull* dan fungsi *cvConvexityDefects*. Fungsi *cvConvexHull* digunakan untuk menghasilkan *polygon* cembung dari kontur tangan yang diproses. *Polygon* tersebut akan terdiri dari *vertice* atau tepi-tepi pada *polygon* dan *vertex* yang merupakan garis lurus yang menghubungkan antar *vertice*. Pada proses selanjutnya, tepi-tepi polygon tersebut yang akan dianalisis sebagai ujung jari. Pada Gambar 4.5 berikut ini merupakan gambar kontur tangan yang diproses menjadi sebuah bentuk *polygon*. Sebelum polygon tersebut diproses menjadi *convex polygon* yang tepi-tepinya merupakan gambaran dari ujung jari.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar 4.5. Kontur Tangan, Polygon dari Kontur Tangan dan *Convex* Polygon

Pada Gambar 4.5, terlihat jendela paling kiri meperlihatkan garis biru yang merepresentasikan kontur tangan. Sedangkan, gambar di tengah merupakan kontur tangan yang sudah diproses menjadi sebuah polygon yang terdiri dari garis-garis (*vertex*) yang membentuk banyak sudut (*vertices*) pada *polygon* tersebut. Gambar terakhir di kanan, merupakan gambar yang memperlihatkan polygon kontur tangan (gambar tengah) yang telah diproses menjadi sebuah *convex polygon*. Pada *polygon* ini, cekungan yang terbentuk dari garis-garis pada polygon sebelumnya dihilangkan, sehingga membentuk *polygon* cembung seperti terlihat pada gambar sebelah kanan. *Polygon* cembung tersebut dihasilkan dengan memeriksa setiap *vertex*, jika sudut yang terbentuk dari suatu *vertex* kurang dari 180^0 artinya *vertex* tersebut merupakan bagian dari *convex polygon*, jika tidak memenuhi kondisi tersebut maka *vertex* yang bersangkutan akan diabaikan. Dengan mendapatkan *convex* polygon, maka *vertex* atau tepi yang terdapat pada *polygon* tersebut merupakan representasi ujung jari pada kontur tangan.

Selanjutnya, untuk menemukan posisi ujung jari, penulis menggunakan fungsi *cvConvexityDefects*. Fungsi *cvConvexityDefects* dijalankan berdasarkan kontur tangan *polygon* dan *convex polygon* tangan yang telah diperoleh dari proses sebelumnya. Dengan menjalankan fungsi tersebut, didapat posisi koordinat ujung jari yang terdeteksi, yaitu lingkaran merah dan posisi-posisi cekungan antar ujung jari, yaitu lingkaran hijau. Algoritma yang digunakan pada fungsi ini, menggunakan teknik menghitung titik konveksitas dan kecekungan pada kontur yang dikembangkan oleh Kazuhiro Homma pada sebuah jurnal *Nuclear Medicine* pada tahun 1985 [27]. Hasil dari algoritma ini terlihat pada Gambar 4.6.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar.4.6. Letak Lima Ujung Jari dan Titik-titik antara Lima jari yang Terdeteksi pada Kontur Tangan

4.3.4. Penentuan Posisi Ujung Jari

Dari fungsi *cvConvexityDefects* didapat dua jenis variabel *array* yang penulis dapat manfaatkan untuk analisa posisi ujung jari, yaitu variabel yang mengembalikan posisi koordinat 2D dari titik puncak cembungan (*convexity*) dan titik cekungan (*concavity*) dari kontur tangan. Untuk menentukan posisi koordinat 2D ujung jari yang terdeteksi, digunakan kalkulasi jarak titik-titik cembungan yang disimpan di dalam sebuah variabel *array* terhadap titik pusat kontur tangan yang terdeteksi oleh sistem. Sebagai gambaran, Gambar 4.7 menunjukkan morfologi tangan yang umum digunakan manusia untuk menunjuk suatu benda. Morfologi tangan seperti ini sekaligus sebagai asumsi morfologi tangan yang digunakan sebagai representasi pengguna sedang menunjuk sebuah objek.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar 4.7. Morfologi Tangan yang Akan Diproses Sistem Ketika Pengguna Menunjuk Suatu Objek

Pada Gambar 4.7 di atas terlihat morfologi/pose tangan pengguna yang diasumsikan ketika menunjuk sebuah objek. Gambar di sebelah kanan merupakan kontur tangan yang terdeteksi sistem dan beberapa titik cekungan dan cembungan yang terdeteksi menggunakan fungsi *cvConvexityDefects*. Terlihat pada gambar tersebut bahwa banyak nilai koordinat 2D yang terdeteksi sebagai titik minimal cekungan dan titik maksimal cembungan, yaitu lingkaran merah dan hijau. Dengan demikian, untuk menemukan posisi ujung jari diperlukan satu tahap proses lagi yang harus dilakukan.

Untuk mendapatkan posisi ujung jari, dilakukan pendekatan dengan membandingkan posisi-posisi puncak di kontur cembung terhadap posisi pusat dari kontur tangan yang terdeteksi. Setiap posisi puncak yang terdeteksi dihitung jaraknya terhadap posisi pusat kontur tangan menggunakan persamaan untuk mencari jarak Euclidean dua poin pada koordinat kartesian dua dimensi. Berikut adalah Persamaan 1 yang digunakan untuk mencari jarak Euclidean.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Persamaan 1. Persamaan Untuk Mencari Jarak Euclidean D(a,b) antara poin a dan poin b

$$D(a,b) = \sqrt{(b1 - a1)^2 + (b2 - a2)^2} \quad \ldots\ldots(1)$$

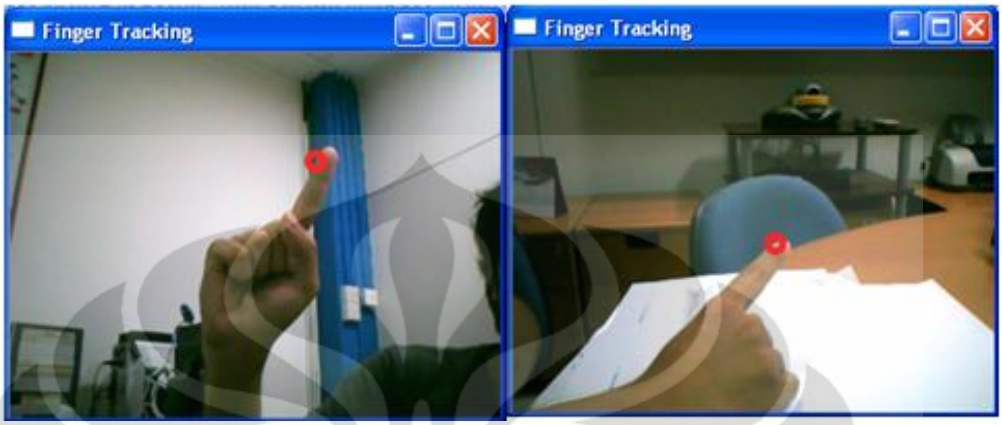
Adapun posisi pusat kontur tangan didapat dengan cara melakukan *ellipse fitting* (penyesuaian bentuk elips) dan mengambil posisi koordinat pusat dari elips yang dihasilkan. Proses *ellipse fitting* dilakukan dengan pendekatan metode kuadrat terkecil (*least square method*) terhadap sebaran poin-poin (posisi pixel) yang merupakan representasi objek tangan pada citra *grayscale*. Untuk menghasilkan elips pada kasus ini, digunakan fungsi *cvFitEllipse* pada OpenCV. Fungsi cvFitEllipse diterapkan berdasarkan metode *ellipse fitting* yang dikembangkan Andrew Fitzgibbon pada tahun 1999 [27]. Gambar 4.8 menunjukkan elips yang dihasilkan dengan menggunakan fungsi tersebut pada representasi kontur tangan.

Gambar 4.8. Elips (garis merah) pada Kontur Tangan dan Pusat Koordinat Kontur Tangan yang Terdeteksi (lingkaran biru)

Dengan mendapatkan posisi pusat koordinat dari kontur tangan yang diproses dan menghitung jarak semua titik-titik cembung maksimal kontur terhadap posisi pusat kontur tangan, maka dapat ditentukan koordinat 2D titik maksimal yang memiliki jarak Euclidean paling jauh terhadap pusat kontur

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

tangan, hal ini sekaligus merupakan representasi dari posisi ujung jari yang dicari. Hasil dari semua proses di atas dapat terlihat pada Gambar 4.8 di bawah ini.



Gambar 4.8. Posisi Ujung Jari yang Terdeteksi Oleh Sistem (lingkaran merah)

4.3.5. Pengenalan Objek Warna dengan Interaksi Jari dari Pengguna

Pengenalan objek warna pada bagian ini akan diimplementasikan sesuai dengan diagram alir pada Gambar 3.5. Warna yang akan dikenali ditentukan berdasarkan warna objek yang ditunjuk oleh pengguna, hal ini dimungkinkan dengan mendeteksi posisi koordinat ujung jari pada citra dan membandingkannya dengan bingkai citra yang disimpan sebelum tangan terdeteksi pada kamera. Posisi koordinat ujung jari pada citra sudah bisa didapat dengan implementasi yang telah dijelaskan pada bagian sebelumnya. Selanjutnya, dilakukan pengenalan warna pada koordinat tertentu pada gambar. Untuk itu, perlu dilakukan pendefinisian nama-nama warna yang dapat dideteksi oleh sistem ini.

Pendefinisan nama warna dilakukan berdasarkan kombinasi antar komponen-komponen dari format warna pada citra yang diproses (RGB, HSV, HSL, dll). Untuk kesempatan kali ini, penulis memilih pengenalan warna dilakukan dalam format HSV dan HSL. Pengenalan warna dengan HSV dan HSL dipilih dengan alasan warna dapat diklasifikasikan dengan relatif baik dalam sebuah jarak (*range*) masing-masing komponen, H, S dan V/L. HSV dan HSL memiliki permodelan warna yang mirip [24]. Pada HSV dan HSL, warna dapat diklasifikasikan dalam sebuah *range* karena komponen *Hue* dominan terhadap

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

menunjukkan tingkat intensitas warna yang dihasilkan pada *Hue* tertentu. *Value* menunjukkan tingkat kecerahan pada warna yang dideteksi. Dan *Lightness* pada HSL yang menunjukkan intensitas suatu warna dari hitam sampai putih [24]. Berbeda dengan format RGB yang memiliki keterkaitan yang tinggi antar 3 komponen yang ada pada RGB, sehingga sulit untuk mengklasifikasikan nama warna tertentu dalam *range* komponen RGB.

Klasifikasi nama warna pada sistem ini dilakukan dengan mengamati warna yang dihasilkan dari kombinasi komponen dalam format warna HSV dan HSL pada *library* OpenCV yang disesuaikan dengan persepsi nama warna sehari-hari yang dikenali manusia dengan pengelihatan warna normal (tidak buta warna parsial atau buta warna total). Nama warna yang didefinisikan pada kesempatan kali ini dibatasi berdasarkan nama warna-warna yang sering digunakan manusia dalam kehidupan sehari-hari. Nama-nama warna yang diefinisikan juga dibatasi untuk nama warna dalam bahasa Inggris. Berikut pada Tabel 4.1 dan Tabel 4.2 berturut-turut adalah nama-nama warna yang didefinisikan dalam implementasi sistem beserta *range* nilai pada format warna HSV dan HSL di OpenCV.

Tabel 4.1. Tabel Nilai Komponen HSV untuk Setiap Warna yang Didefiniskan

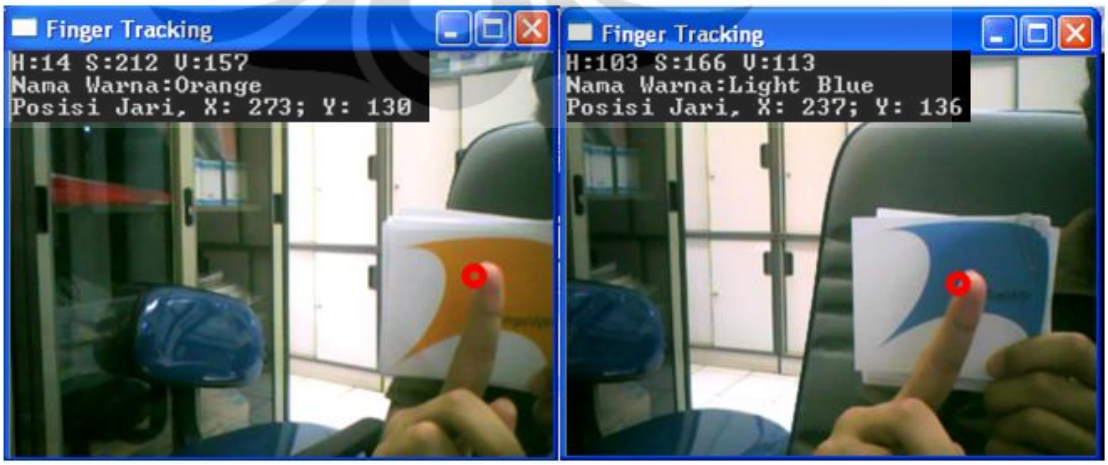
Nama Warna	Hue	Saturation	Value
White	0-255	0-17	200-255
Gray	0-255	0-17	110-200
Black	0-255	0-255	0-50
Pink	0-3 / 175-182	70-120	230-255
Dark Red	0-3 / 170-182	230-255	145-175
Red	0-10 / 165-190	0-255	0-255
Brown	15-17 / 190-197	79-255	155-190
Orange	10-23 / 191-200	0-255	0-255
Yellow	24-34 / 201-210	0-255	0-255
Light Green	35-50 / 211-230	0-255	0-255
Green	51-65 / 231-250	0-255	0-255
Tosca	66-83 / 251-255	0-255	0-255
Light Blue	84-107	0-255	0-255
Blue	108-136	91-255	0-255
Purple	108-140	0-90	0-255
Magenta	141-165	0-255	0-255

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Tabel 4.2. Tabel Nilai Komponen HSL untuk Setiap Warna yang Didefiniskan

Nama Warna	Hue	Saturation	Lightness
White	0-255	0-255	223-255
Gray	0-255	0-64	0-255
Black	0-255	0-255	0-32
Pink	0-3 / 175-182	0-255	191 - 223
Dark Red	0-3 / 170-182	0-255	95 - 140
Red	0-10 / 165-190	0-255	141 - 179
Brown	15-17 / 190-197	0-255	63 - 95
Orange	10-23 / 191-200	0-255	127 - 160
Yellow	24-34 / 201-210	0-255	127 - 178
Light Green	35-50 / 211-230	0-255	127 - 191
Green	51-65 / 231-250	0-255	63 – 127
Tosca	66-83 / 251-255	0-255	127 – 204
Light Blue	84-107	0-255	127 – 191
Blue	108-136	0-255	51 – 153
Purple	108-140	0-255	95 – 191
Magenta	141-165	0-255	95 – 140

Berdasarkan jarak nilai komponen-komponen pada HSV atau HSL yang telah didefinisikan di atas, maka sistem dapat menentukan nama warna di koordinat gambar tertentu hasil tangkapan kamera yang dikonversi terlebih dahulu dari format warna RGB (format yang dihasilkan kamera) ke format warna HSV atau HSL. Gambar 4.9 di bawah ini adalah hasil dari pengenalan warna dengan interaksi langsung pengguna dengan jari.



Gambar 4.9. Posisi Ujung Jari yang Terdeteksi Oleh Sistem(lingkaran merah)

Pada Gambar 4.9 terlihat bahwa sistem telah dapat mengenali warna yang dimaksud oleh pengguna dengan baik. Untuk mencapai tujuan dari fitur ini, yaitu menghasilkan *output* nama warna berupa suara, diperlukan satu proses yang dapat mengubah teks nama warna dalam format *string*. Untuk itu digunakan sebuah *library*, yaitu *System.Speech Managed Namespace* yang merupakan bagian dari *Microsoft .NET Framework*. Pada *library* tersebut, digunakan sebuah *Class* bernama *SpeechSynthesizer* yang memiliki fungsi untuk mengubah format data *string ke* bentuk gelombang suara. Hasilnya sistem dapat bekerja dengan baik dengan mengucapkan nama-nama warna yang telah didefinisikan dengan jelas.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

BAB V

PENGUJIAN DAN ANALISIS

Pada bagian ini, dibahas hasil pengujian dan analisis dari sistem yang telah diimplementasikan. Sistem telah berhasil diimplementasikan pada eBox-3310 dengan WES2009, hanya saja sumber daya eBox-3310 tidak cukup untuk menjalankan perangkat lunak dengan lancar, oleh karena itu digunakan komputer personal untuk pengujian. Pengujian dan analisis dibagi menjadi dua bagian, pertama adalah pengujian dan analisis performa algoritma pendeteksi ujung jari yang telah diterapkan, serta pengujian dan analisa sistem pengenalan warna dengan interaksi jari pengguna berdasarkan warna umum yang telah didefinisikan pada bagian sebelumnya.

5.1 Pengujian dan Analisis Performa Algoritma Pendeteksi Ujung Jari

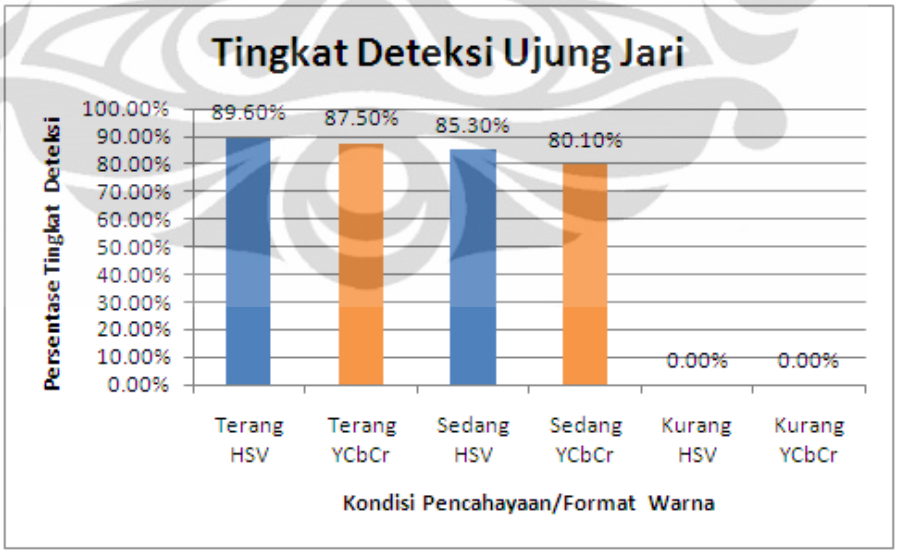
Pada bagian pengujian dan analisis pertama ini akan dibahas hasil pengujian dan analisa dari algoritma pendeteksi ujung jari yang telah diimplementasikan menggunakan *EmguCV* dan *.NET Framework 3.5*. Bagian pertama yang diuji dan dianalisis adalah pengujian tingkat deteksi ujung jari pada sistem. Pengujian dilakukan dengan membandingkan 2 metode klasifikasi objek kulit dan pada kondisi cahaya terang, sedang dan kurang. Untuk menguji tingkat deteksi ujung jari, dilakukan gerakan acak dengan menggunakan ujung jari pada area yang ditangkap oleh kamera, lalu menghitung ujung jari yang terdeteksi berdasarkan setiap bingkai (*frame*) gambar dan setiap kondisi berbeda diuji untuk 1000 bingkai gambar. Pengujian dilakukan pada kondisi di dalam ruangan (*indoor*) dan pemilihan latar belakang area pengujian yang cukup sederhana, hal ini dimaksudkan untuk memaksimalkan hasil tingkat deteksi ujung jari yang didapat. Gambaran kondisi pengujian dapat dilihat pada Gambar 5.1, selanjutnya Tabel 5.1 dan Gambar 5.2 berikut ini menunjukkan hasil pengujian yang dilakukan.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

Gambar 5.1. Gambaran Kondisi Pencahayaan pada Saat Pengujian (terang, sedang dan kurang)

Tabel 5.1. Tabel Tingkat Deteksi Ujung Jari

Tingkat Pencahyaan	Format Warna	Tingkat Deteksi
Terang	HSV	89.60%
	YCbCr	87.50%
Sedang	HSV	85.30%
	YcbCr	80.10%
Kurang	HSV	0.00%
	YCbCr	0.00%



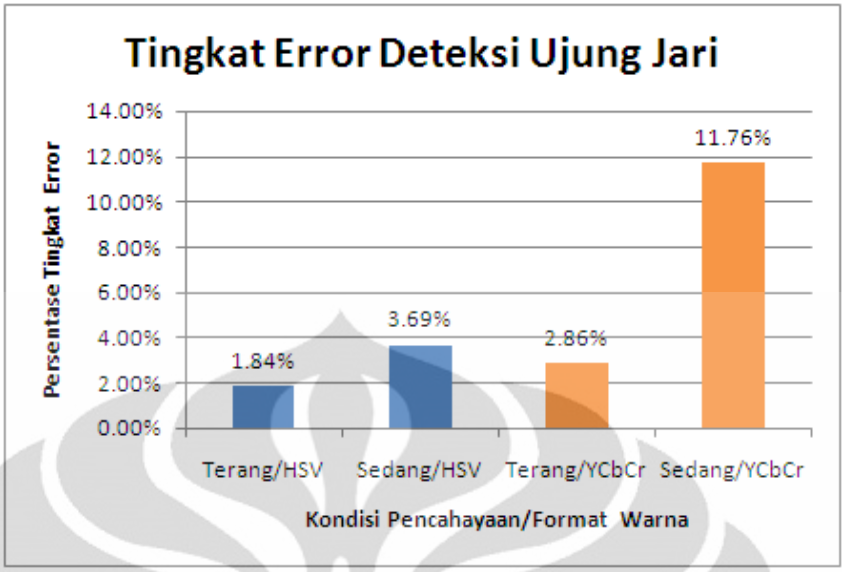
Gambar 5.2 Grafik Tingkat Deteksi Ujung Jari

Pada Tabel 5.1 dan Gambar 5.2 di atas terlihat bahwa variasi tingkat pencahayaan sangat mempengaruhi tingkat deteksi jari pada sistem ini, bahkan jari tidak terdeteksi sama sekali ketika kondisi pencahayaan kurang. Hal ini terjadi karena metode klasifikasi objek kulit yang digunakan tidak mampu membedakan warna kulit dan bukan kulit pada kondisi cahaya yang kurang. Selain itu, secara umum dapat dilihat bahwa pada kondisi terang hingga sedang kedua format warna yang digunakan untuk melakukan klasifikasi kulit dengan metode *explicit thresholding* mampu menghasilkan tingkat deteksi rata-rata di atas 80% dengan catatan latar belakang yang ada cukup sederhana. Hal ini akan sangat berbeda jika latar belakang yang digunakan cukup kompleks dengan objek-objek yang memiliki warna mirip dengan objek kulit.

Jika terdapat objek-objek yang menyerupai warna kulit seperti warna coklat atau coklat muda, maka algoritma klasifikasi objek kulit akan menyimpulkan objek mirip warna kulit tadi sebagai objek kulit, sehingga sistem akan mengeluarkan hasil posisi ujung jari yang salah. Kesalahan deteksi juga kerap terjadi jika kondisi cahaya agak kurang, sering terjadi sebagian objek kulit tidak terdeteksi oleh sistem, karena kondisi pencahayaan pada bagian tertentu objek tangan kurang atau terkena bayangan tertentu. Hasil klasifikasi objek kulit menjadi tidak sempurna dan menghasilkan kesalahan deteksi ujung jari.

Tabel 5.2 Tabel Tingkat Error Deteksi Ujung Jari

Tingkat Pencahayaan	Format Warna	Jumlah Ujung Jari Terdeteksi	Jumlah Deteksi Error	Tingkat Deteksi Error
Terang	HSV	435	8	1.84%
Sedang	HSV	407	15	3.69%
Terang	YCbCr	420	12	2.86%
Sedang	YCbCr	289	34	11.76%



Gambar 5.3 Grafik Tingkat Error Deteksi Ujung Jari

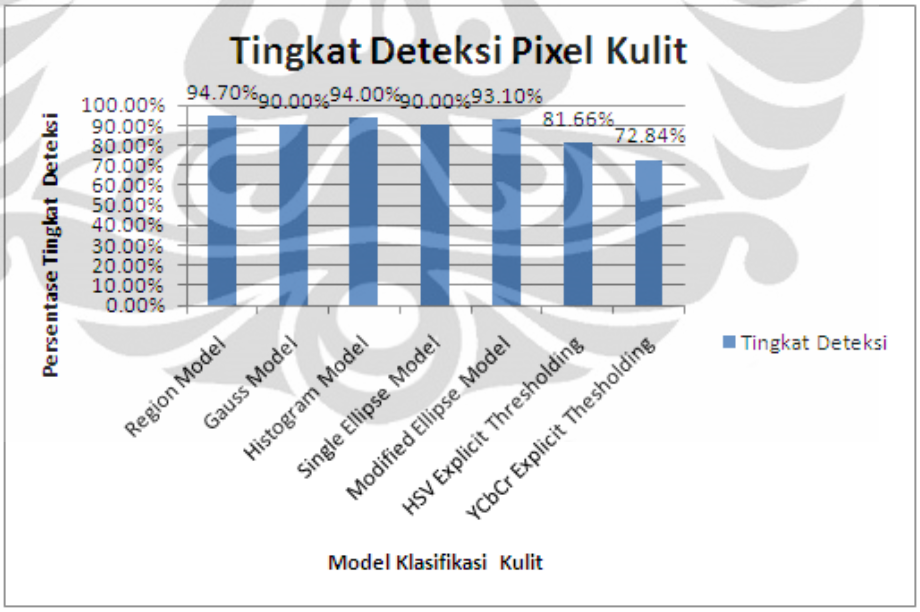
Tabel 5.2 dan Gambar 5.3 di atas menunjukkan persentase tingkat pendeteksian ujung jari yang error. Deteksi ujung jari yang error didapatkan dengan cara menghitung jumlah deteksi yang salah (poin yang terdeteksi bukan di ujung jari) pada 500 bingkai gambar yang diuji pada tingkat pencahayaan terang dan sedang dengan menggunakan metode klasifikasi objek kulit *explicit thresholding* pada format warna HSV dan YCbCr. Dari hasil yang terlihat pada tabel dan grafik, dapat dilihat bahwa format warna HSV menghasilkan tingkat error yang lebih rendah pada kedua kondisi cahaya dibandingkan dengan klasifikasi dengan menggunakan format warna YCbCr. Sedangkan, dengan menggunakan format warna YCbCr penurunan tingkat pencahayaan pada metode klasifikasi objek kulit ini sangat mempengaruhi tingkat error deteksi ujung jari secara signifikan.

Selain itu, juga dilakukan pengujian tingkat pemrosesan bingkai gambar (*frame rate*), hasilnya menunjukkan bahwa *frame rate* pada kedua jenis format warna menunjukkan tingkat pemrosesan yang sama, yaitu sekitar 7.5 bingkai per detik. Hal ini dapat menunjukkan bahwa transformasi yang tidak linier dari format warna RGB ke format warna HSV tidak telalu mempengaruhi waktu pemrosesan pada algoritma deteksi ujung jari, mengingat transformasi dari format warna RGB ke format warna YCbCr merupakan transformasi yang linier.

Sesuai dengan yang telah dibahas pada bagian sebelumnya, bahwa metode atau algoritma klasifikasi objek kulit yang dipilih akan sangat menentukan performa dari algoritma pendeteksian ujung jari, oleh karena itu, berikut penulis membandingkan performa pengklasifikasian objek kulit dengan beberapa metode klasifikasi objek kulit populer yang diperoleh dari [14].

Tabel 5.3 Perbandingan Beberapa Jenis Metode Klasifikasi Kulit dalam Deteksi Pixel Kulit [14]

Metode Klasifikasi Kulit	Tingkat Deteksi
Region Model	94.70%
Gauss Model	90.00%
Histogram Model	94.00%
Single Ellipse Model	90.00%
Modified Ellipse Model	93.10%
HSV Explicit Thresholding	81.66%
YCbCr Explicit Thesholding	72.84%



Gambar 5.4 Grafik Perbandingan Tingkat Deteksi Pixel Kulit pada Beberapa Model Klasifikasi Kulit [14]

Pada Tabel 5.3 terlihat bahwa metode klasifikasi kulit yang diterapkan pada metode ini memiliki tingat deteksi yang lebih rendah dibandingkan metode-metode klasifikasi kulit lainnya. Hal ini cukup dapat diterima mengingat metode

klasifikasi kulit dengan *explicit thresholding* merupakan metode klasifikasi warna kulit yang paling sederhana [17]. Pengujian tingkat deteksi pixel kulit yang digunakan untuk mendapatkan hasil pada Tabel 5.3 dilakukan dengan membandingkan jumlah pixel yang mampu terdeteksi sebagai *pixel* kulit oleh metode HSV dan YCbCr *explicit thresholding* dengan jumlah pixel kulit sebenarnya yang ditangkap oleh kamera. Data pengujian diambil pada kondisi cahaya terang dan latar belakang yang sederhana sebelum proses pengolahan citra dilasi dan erosi dilakukan. Data yang digunakan untuk kalkulasi tingkat deteksi pixel kulit merupakan data yang valid sesuai dengan masing-masing performa metode klasifikasi kulit yang digunakan. Kalkulasi jumlah pixel area kulit dilakukan secara manual dengan bantuan perangkat *Magic Wand* pada perangkat lunak pengolah citra *Adobe Photoshop CS3* yang dapat mendeteksi ujung dari sebaran warna dan menyeleksi area tersebut lalu menghitung jumlah pixel pada area yang diseleksi tersebut.

5.2. Pengujian dan Analisis Pengenalan Warna dengan Interaksi Jari Pengguna

Pada bagian ini akan dibahas hasil pengujian dan analisis untuk sistem pengenalan warna dengan interaksi jari pengguna. Warna yang dapat dikenali pada sistem ini merupakan warna yang telah didefinisikan menggunakan format warna HSV dan HSL sesuai dengan warna yang tercantum pada Tabel 4.1 dan Tabel 4.2 pada bab sebelumnya. Pengujian dilakukan dengan menguji ketepatan algoritma pada sistem dalam menebak atau mengenali warna dari sebuah objek yang ingin diketahui oleh pengguna. Pada pengujian kali ini, objek disimulasikan dengan menggunakan contoh warna yang dicetak pada secarik kertas. Contoh objek yang digunakan untuk pengujian dapat dilihat pada Gambar 5.5.

Gambar 5.5. Contoh Objek Warna yang Digunakan untuk Pengujian Sistem Pengenalan Warna

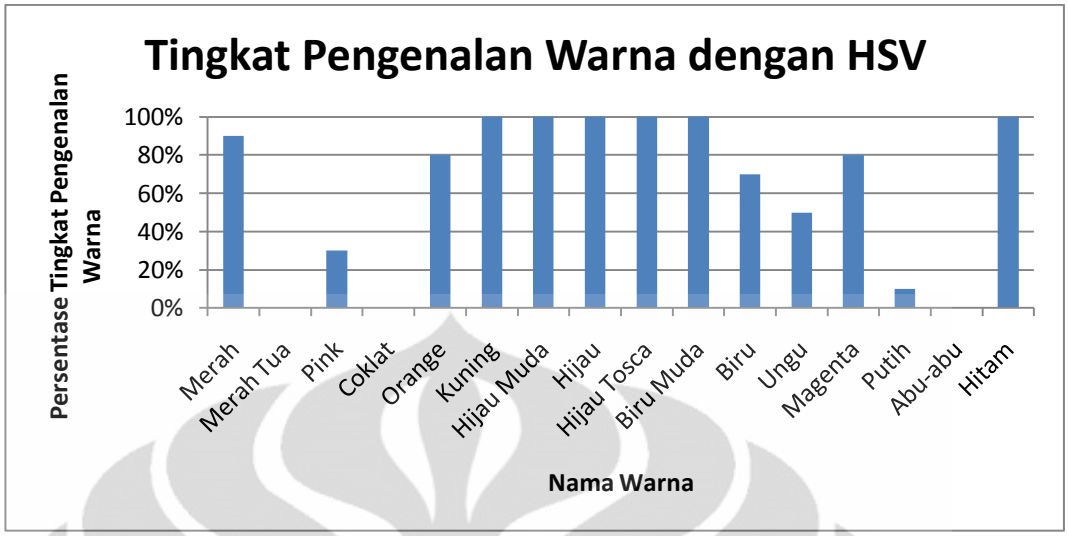
Secara keseluruhan, terdapat 16 jenis warna yang diuji. Masing-masing objek warna diuji dengan sepuluh percobaan yang dilakukan pada lingkungan dengan cahaya terang dan di dalam sebuah ruangan. Tabel 5.4 dan Tabel 5.5 menunjukkan hasil pengujian tingkat pengenalan warna dengan metode interaksi jari pengguna ke objek warna.

Tabel 5.4 Tabel Hasil Pengujian Tingkat Pengenalan Warna dengan Format Warna HSV

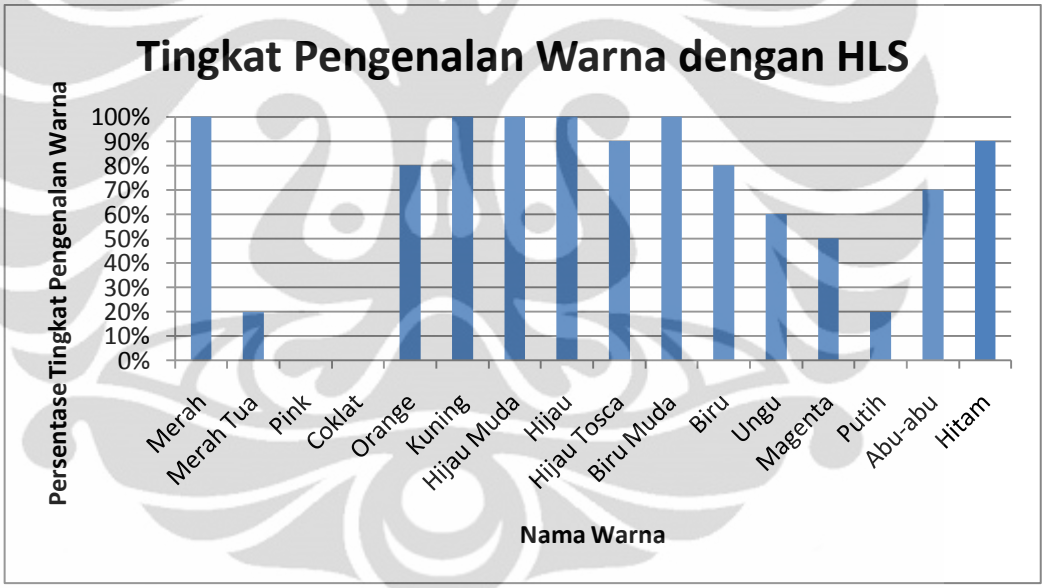
Nama Warna	Merah	Merah Tua	Pink	Coklat	Orange	Kuning	Hijau Muda	Hijau
Tingkat Pengenalan	90%	0%	30%	0%	80%	100%	100%	100%
Nama Warna	Hijau Tosca	Biru Muda	Biru	Ungu	Magenta	Putih	Abu-abu	Hitam
Tingkat Pengenalan	100%	100%	70%	50%	80%	10%	0%	100%

Tabel 5.5 Tabel Hasil Pengujian Tingkat Pengenalan Warna dengan Format Warna HSL

Nama Warna	Merah	Merah Tua	Pink	Coklat	Orange	Kuning	Hijau Muda	Hijau
Tingkat Pengenalan	100%	20%	0%	0%	80%	100%	100%	100%
Nama Warna	Hijau Tosca	Biru Muda	Biru	Ungu	Magenta	Putih	Abu-abu	Hitam
Tingkat Pengenalan	90%	100%	80%	60%	50%	20%	70%	90%



Gambar 5.6. Grafik Hasil Pengujian Tingkat Pengenalan Warna dengan Format Warna HSV



Gambar 5.7. Grafik Hasil Pengujian Tingkat Pengenalan Warna dengan Format Warna HSL

Dari hasil yang terlihat pada Tabel 5.4, Tabel 5.5, Gambar 5.6 dan Gambar 5.7, terlihat bahwa mayoritas warna yang didefiniskan pada kedua format, HSV dan HSL, memiliki tingkat pengenalan di atas 80%. Akan tetapi, juga ada beberapa jenis warna yang memiliki tingkat deteksi yang buruk (mendekati 0%). Hal ini dapat terjadi karena beberapa hal, diantaranya adalah sebagai berikut:

1. Klasifikasi nama-nama warna yang dilakukan pada domain warna HSV dan HSL ditinjau berdasarkan persepsi warna yang dihasilkan oleh kombinasi komponen H, S dan V/L pada layar komputer personal. Ada kemungkinan bahwa hasil akan lebih baik jika klasifikasi warna dilakukan berdasarkan nilai komponen H, S dan V/L pada objek warna yang ditangkap langsung oleh kamera dan disesuaikan dengan persepsi penglihatan manusia.

2. Ketidak-akuratan hasil cetak objek warna yang digunakan oleh pengujian. Alasan ini diperkuat oleh jawaban para responden yang menjawab pertanyaan mengenai kesesuaian objek warna yang dijadikan objek pengujian dengan persepsi pengelihatan mereka.

3. Kemiripan warna yang akan dikenali dengan warna yang diklasifikasikan sebagai warna kulit. Hal ini terjadi pada objek berwarna coklat dan merah tua. Ketika pengujian dilakukan untuk mengenali warna tersebut dengan jari, kesalahan deteksi posisi ujung jari sering salah, sehingga salah mengenali warna yang dimaksud oleh pengguna.

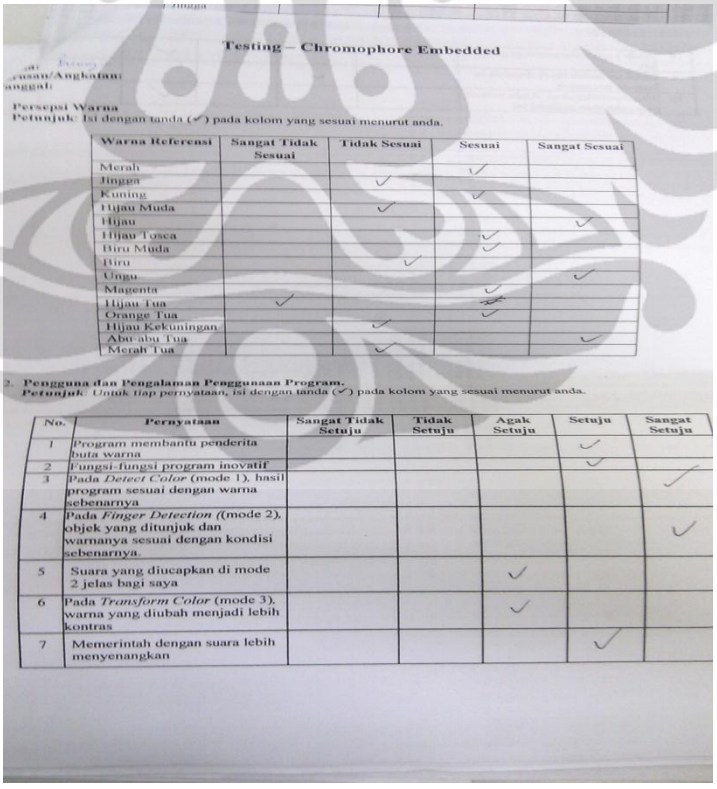
Selain itu, juga dapat dilihat bahwa jenis warna yang hanya tergantung pada komponen *Hue* saat pengklasifikasian, baik pada format warna HSV dan HSL, memiliki tingkat akurasi pengenalan warna yang tinggi (mendekati 100%). Sedangkan, jenis warna yang tergantung komponen lainnya, seperti *Saturation* pada HSV dan *Lightness* pada HSL memiliki tingkat akurasi yang rendah.

5.2.3. Pengujian Kualitatif Fitur *Point-to-Sound*

Selain menguji keakuratan sistem secara kuantitatif dengan metode dan hasil pengujian yang telah dibahas, penulis juga melakukan pengujian secara kualitatif. Pengujian secara kualitatif dilakukan dengan meminta 10 orang responden (terdiri dari responden berpengelihatan normal dan responden yang mengalami buta warna parsial) mencoba unit perangkat tertanam secara keseluruhan. Setelah responden mencoba unit tersebut secara keseluruhan, responden diminta untuk mengisi borang yang terdiri dari beberapa pertanyaan.

Salah satu pertanyaan pada borang merupakan pertanyaan mengenai pendapat responden mengenai kesesuaian nama atau jenis warna yang disimpulkan oleh sistem sesuai dengan warna yang dimaksud oleh pengguna dengan cara menunjuk objek warna tertentu. Opsi yang dapat dipilih oleh responden adalah "Sangat Setuju" dengan nilai 5, "Setuju" yang bernilai 4, "Agak Setuju" yang bernilai 3, "Tidak Setuju" yang bernilai 2 dan "Sangat Tidak Setuju" yang bernilai 1 pada interval kepercayaan 95%.

Setelah dilakukan pengolahan terhadap tanggapan dari responden, interval kepercayaan 95% yang didapat adalah 3.6 ± 0.433. Dari interval kepercayaan 95% dan lima opsi yang dapat dipilih oleh responden, dapat disimpulkan bahwa para responden cenderung setuju bahwa sistem bekerja dengan baik dengan memberikan hasil keluaran nama warna yang sesuai dengan warna yang mereka ingin ketahui melalui interaksi jari.



Testing – Chromophore Embedded

Persepsi Warna
Petunjuk: Isi dengan tanda (✓) pada kolom yang sesuai menurut anda.

Warna Referensi	Sangat Tidak Sesuai	Tidak Sesuai	Sesuai	Sangat Sesuai
Merah			✓	
Jingga		✓		
Kuning			✓	
Hijau Muda		✓		
Hijau				✓
Hijau Tosca			✓	
Biru Muda			✓	
Biru		✓		
Ungu				✓
Magenta			✓	
Hijau Tua	✓		✓	
Orange Tua			✓	
Hijau Kekuningan		✓		
Abu-abu Tua				✓
Merah Tua		✓		

2. **Pengguna dan Pengalaman Penggunaan Program.**
Petunjuk: Untuk tiap pernyataan, isi dengan tanda (✓) pada kolom yang sesuai menurut anda.

No.	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Agak Setuju	Setuju	Sangat Setuju
1	Program membantu penderita buta warna				✓	
2	Fungsi-fungsi program inovatif				✓	
3	Pada *Detect Color* (mode 1), hasil program sesuai dengan warna sebenarnya					✓
4	Pada *Finger Detection* ((mode 2), objek yang ditunjuk dan warnanya sesuai dengan kondisi sebenarnya.					✓
5	Suara yang diucapkan di mode 2 jelas bagi saya			✓		
6	Pada *Transform Color* (mode 3), warna yang diubah menjadi lebih kontras			✓		
7	Memerintah dengan suara lebih menyenangkan				✓	

Gambar 5.8. Contoh Borang yang Digunakan

BAB VI

KESIMPULAN

- Algoritma deteksi ujung jari yang diterapkan dengan metode klasifikasi kulit *explicit thresholding* pada kondisi cahaya terang dan cukup menunjukkan tingkat deteksi rata-rata 87.45% pada format warna HSV dan 83.8% pada format warna YCbCr. Metode dengan format warna HSV memiliki tingkat deteksi ujung jari lebih baik 3.65% dibandingkan format warna YCbCr.
- Penurunan intensitas cahaya menyebabkan tingkat deteksi ujung jari dan keakuratan pendeteksian ujung jari menurun karena hasil klasifikasi kulit yang tidak sempurna, penurunan intensitas cahaya terang ke sedang memberi penurunan sebesar 4.3% pada tingkat deteksi dengan format warna HSV dan 7.4% pada YCbCr, serta peningkatan sebesar 1.85% terhadap tingkat error pendeteksian dengan format HSV dan peningkatan error sebesar 8.9% dengan format YCbCr.
- Pada kasus ini, metode klasifikasi kulit dengan menggunakan HSV lebih baik 8.82% dari metode klasifikasi kulit dengan menggunakan YCbCr dalam mendeteksi pixel yang mengandung warna kulit
- Dalam sistem pengenalan warna, format warna HSV dan HSL dapat digunakan dengan cukup baik untuk mengklasifikasikan mayoritas dari 16 jenis warna yang diuji, yaitu dengan tingkat pengenalan di atas 80%. Hanya saja terdapat beberapa jenis warna yang tingkat pengenalannya buruk, yaitu di bawah 50%, contohnya warna coklat.
- Responden menyatakan bahwa mereka cenderung setuju metode interaksi jari untuk mengenali warna bekerja mengenali warna yang ditunjuk oleh pengguna dengan baik

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

DAFTAR REFERENSI

[1] Wakita, Ken, & Shimamura, Kenta. (2005). *Smart Color : Disambiguation Framework for Color-Blind*. Paper presented at the ASSETS: The International ACM SIGACCESS Conference on Computers and Accessibility 2005 Conference, 9-12 Oktober 2005, Baltimore, Maryland, USA.

[2] Kalloniatis, Michael and Luu, Charles. (2005) *Psychophysics of Vision: The Perception of Color*. University of Melbourne. U.S. National Library of Medicines National Institutes of Health.

[3] Michael Barr and Anthony J. Massa. (2006). *Programming Embedded System: with C and GNU Development tools*. O'Reilly Media Inc.

[4] Frederic P Miller, Agnes F Vandome, John McBrewster. (2009). *Augmented Reality*. VDM Publishing House.

[5] Bimber, Oliver and Raskar, Ramesh. (2005). *Spatial Augmented Reality-Merging Real and Virtual Worlds*. Massachusetts. A K Peters, Ltd.

[6] Pavlov, Stanislav and Belevsky, Pavel. (2009). *Windows Embedded CE 6.0 Fundamentals*. Microsoft Press.

[7] Phung, Samuel. (2009). *Professional Windows Embedded CE 6.0*. Indianapolis. Wiley Publishing, Inc.

[8] Deitel, M Harvey, et al. (2001). *C# How to Program*. Prentice Hall.

[9] OpenCV Wiki. http://opencv.willowgarage.com/wiki/. Diakses pada 14 Desember 2010.

[10] EmguCV Wiki. http://emgu.com/wiki/. Diakses pada 14 Desember 2010.

[11] Colblindor. http://www.colblindor.com. Diakses pada 4 Desember 2010.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

[12] Noergaard, Tammy. (2005). *Embedded System Architecture: A Comprehensive Guide for Engineers and Programmers.* Oxford: Elsevier, Inc.

[13] Head Mounted Display. http://www.igargoyle.com. Diakses pada 4 Desember 2010.

[14] Microsoft Team. (2008). *Windows Embedded Standard 2009 Help*. Microsoft.

[15] Tang, Hao-kui and Feng, Zhi-quan. (2008). *Hand's Skin Detection Based on Ellipse Clustering*. 2008 International Symposiums on Computer Science and Computational Technology, 20-22 Desember 2008, Shanghai, China.

[16] Microsoft Team. (2008). *Windows Embedded Studio Documentation*. Microsoft.

[17] Kakumanu, P, Makrogiannis, S, Bourbakis. (2007). N. *A Survey of Skin-Color Modeling and Detection Methods*. Pattern Recognition Society.

[18] EyePilot. http://www.colorhelper.com/. Diakses pada 12 Juni 2011

[19] Enliven. https://market.android.com/details?id=com.revolucian.enlivenfull. Diakses pada 12 Juni 2011.

[20] Color Blind Aid. http://itunes.apple.com/us/app/color-blind-aid/id351425486?mt=8. Diakses pada 12 Juni 2011.

[21] U.S. Army considering augmented reality wearable displays for medics. http://www.tecca.com/news/2011/05/19/army-wearable-display-medics/. Diakses pada 12 Juni 2011.

[22] Arcane Technologies. http://www.arcane-technologies.com/en/. Diakses Pada 12 Juni 2011.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011

[23] Windows Embedded. http://www.microsoft.com/windowsembedded. Diakses pada 12 Juni 2011.

[24] Ananto, Bayu Sri. (2011) *Implementasi Sistem Bantuan Penderita Buta Warna: Desain Antarmuka Pengguna, Sistem Tes Buta Warna dengan Ishihara, dan Tranformasi Warna pada Sistem Realitas Tertambah.* Depok. Universitas Indonesia.

[25] Wicaksana, Burhan Adi. (2011). *Implementasi Sistem Bantuan Penderita Buta Warna: Pendeteksian Warna dan Tampilan Informasi Warna dengan Platform .NET dan EmguCV Library*. Depok. Universitas Indonesia.

[26] Harwahyu, Ruki. (2011). *Implementasi Sistem Bantuan Penderita Buta Warna:Interaksi Suara Untuk Perangkat Tertanam dengan Sistem Operasi Tertanam Microsoft*, Depok. Universitas Indonesia.

[27] Bradski, Gary & Kaehler, Adrian. (2008). *Learning OpenCV*. California. O'Reilly Media.

Implementasi sistem ..., Alfa Sheffildi Manaf, FT UI, 2011