

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : IQBAL ABRIAN Z.
NRP : 5110100139
DOSEN WALI : DANIEL O. SIAHAAN, S.Kom., M.Sc., PD.Eng.
DOSEN PEMBIMBING : 1. DWI SUNARYONO, S.Kom., M.Kom.
2. RIDHO RAHMAN HARIADI, S.Kom., M.Sc.

2. JUDUL TUGAS AKHIR

“Implementasi Sistem Pendeteksi Warna Untuk Penderita Buta Warna Pada Kamera Menggunakan *Platform .NET* dan *EmguCV Library*.”

3. LATAR BELAKANG

Buta warna adalah suatu kelainan yang disebabkan oleh ketidakmampuan sel-sel kerucut dalam retina mata yang mengalami kelemahan atau kerusakan permanen dan tidak mampu merespon warna dengan semestinya. Buta warna merupakan salah satu penyakit yang disebabkan oleh faktor keturunan yang mengakibatkan penderitanya tidak dapat mengenali warna, baik warna tertentu (parsial) atau semua warna (total). Kebutaan warna juga dapat disebabkan oleh konsumsi obat-obatan dalam periode waktu tertentu karena penyakitnya. Ketidakmampuan mengenali warna ini dapat berpotensi menyebabkan berbagai kesulitan bagi para penderitanya dalam kehidupan sehari-hari maupun dalam lingkup yang lebih khusus, karena banyak hal dalam kehidupan ini yang diasosiasikan dengan warna.

Dalam sistem ini, pengenalan warna diperlukan untuk mengolah input berupa video stream yang didapat dari kamera, sehingga didapatkan hasil berupa informasi warna di daerah atau objek tertentu dari masukan video stream tersebut. Informasi warna yang ingin didapatkan tersebut disesuaikan dengan pilihan pengguna sesuai pilihan jenis buta warna yang dideritanya, yang menentukan warna apa yang perlu dimunculkan infomasinya. Informasi warna ini diolah dan ditambahkan dengan masukan video stream secara *realtime* sesuai tujuan dari sistem ini, yaitu memberikan informasi warna kepada penderita buta warna. Selain itu, untuk mengetahui jenis buta warna yang diderita pengguna maka diterapkan salah satu metode populer yang digunakan yaitu menggunakan tes buta warna *Ishihara* [1]. Metode ini menggunakan piringan-piringan warna yang di dalamnya terdapat lingkaran dengan ukuran dan warna yang tersebar secara acak. Di dalam sebaran lingkaran-lingkaran tersebut dibentuk sebuah angka atau benda tertentu yang akan terlihat oleh mata normal, tetapi tidak terlihat bagi penderita buta warna. Contoh piringan warna untuk tes Ishihara terlihat pada Gambar 2.

Pemrograman yang dilakukan dengan menggunakan bahasa pemrograman C# dan *library* EmguCV .NET Framework pada Microsoft Visual Studio. *Library* EmguCV digunakan karena bersifat *cross-platform*, sehingga dapat dijalankan pada berbagai sistem operasi.

Hasil yang diharapkan semoga dengan adanya sistem ini nantinya pengguna dapat mendiagnosa penyakit buta warna sejak dini dan dapat memberikan manfaat kepada penderita buta warna dalam mendeteksi warna secara akurat.

4. RUMUSAN MASALAH

Permasalahan-permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana pengguna mendeteksi warna suatu objek pada kamera video.
2. Bagaimana mendapatkan informasi warna dari hasil pemrosesan pendeteksi.
3. Bagaimana pengguna melakukan tes buta warna pada sistem.
4. Bagaimana mendapatkan hasil tes buta warna dengan menggunakan metode Ishihara pada sistem.

5. BATASAN MASALAH

Adapun batasan ruang lingkup permasalahan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Mengintegrasikan kamera video dengan perangkat lunak pendeteksi warna.
2. Warna yang akan dideteksi oleh kamera video hanya pada kondisi pencahayaan yang baik.

3. Kamera hanya mendeteksi warna yang kemudian diproses sehingga menghasilkan suatu informasi dari warna tersebut.
4. Tes buta warna pada pengguna menggunakan metode *Ishihara* berupa potongan gambar dari pelat-pelat *Ishihara*.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Membangun sebuah sistem pendeteksi warna untuk penderita buta warna agar dapat memudahkan dalam mengenali warna secara akurat menggunakan kamera video.
2. Mengimplementasikan penerapan EmguCV pada sistem pendeteksi warna.
3. Membangun perangkat lunak yang dapat mendeteksi warna melalui kamera video.
4. Membangun perangkat lunak yang dapat digunakan pengguna untuk melakukan tes buta warna.

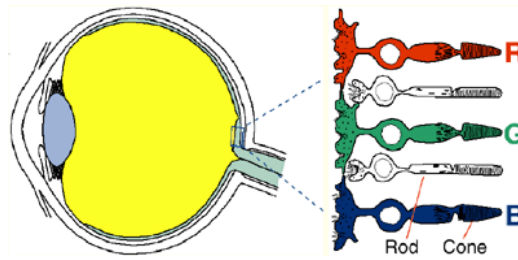
7. MANFAAT TUGAS AKHIR

Tugas akhir ini dibuat dengan harapan dapat melakukan tes buta warna terhadap para pengguna dan dapat memberikan manfaat bagi penderita buta warna dalam mengenali warna di sekitarnya. Sehingga bagi para penderita buta warna dapat mengantisipasi segala kegiatan yang diasosiasikan dengan warna dengan melakukan pendeteksian warna.

8. TINJAUAN PUSTAKA

a. Buta Warna

Mata manusia dapat melihat ketika sebuah cahaya pantulan dari suatu benda ditangkap dan disimulasikan pada retina (suatu membran saraf yang terdapat pada bagian belakang mata). Retina tersebut terdiri dari suatu rangkaian *Rods* dan *Cones*. *Rods*, yang terletak pada permukaan retina sangat peka terhadap intensitas cahaya dan berfungsi sangat baik pada malam hari, namun tidak dapat membedakan warna. *Cones*, yang terletak pada tengah retina (disebut *macula*), tidak begitu peka pada malam, tapi membuat kita dapat mengenali warna pada saat siang hari. Rangkaian struktur mata beserta sel *Rods* dan *Cones* yang telah dijelaskan dapat dilihat pada Gambar 1.



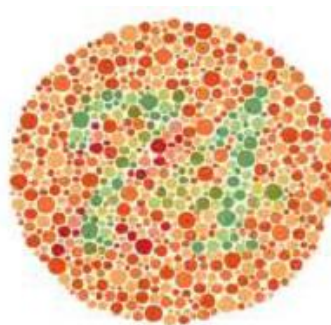
Gambar 1. Struktur Mata beserta sel *cone* dan *rod*

Penglihatan warna pada mata normal (disebut *trichromat*) sangat bergantung pada sel kerucut (*Cones*) retina yang mengandung tiga jenis *photopigments* yang memiliki tingkat kepekaan spektrum yang berbeda. Sel kerucut tersebut, masing-masing memiliki pigment cahaya yang sensitivitasnya berbeda terhadap suatu panjang gelombang tertentu (setiap warna yang terlihat oleh mata memiliki panjang gelombang yang berbeda antara 400 sampai 700 nm). Instruksi pembuatan sel kerucut ini terdapat dalam gen-gen tertentu, dan jika gen tersebut memuat instruksi yang salah maka sel kerucut tersebut akan peka terhadap panjang gelombang yang tidak semestinya, yang akhirnya mengakibatkan buta warna.

Kegagalan penglihatan terhadap suatu warna terjadi ketika satu atau lebih dari ketiga jenis sel kerucut tersebut tidak peka sebagaimana seharusnya, atau sensitivitasnya terganggu. Kegagalan penglihatan terhadap suatu warna yang lebih berat terjadi saat satu atau lebih dari ketiga jenis sel kerucut tersebut benar-benar hilang [2].

b. Metode Ishihara

Metode *Ishihara* yaitu metode yang dapat dipakai untuk menentukan dengan cepat suatu kelainan buta warna didasarkan pada penggunaan kartu bertitik-titik, seperti Gambar 2. Kartu ini disusun dengan menyatukan titik-titik yang mempunyai bermacam-macam warna.



Gambar 2. Contoh Tes Buta Warna Dengan Metode *Ishihara*

Metode *Ishihara* ini dikembangkan menjadi Tes Buta Warna *Ishihara* oleh Dr. Shinobu Ishihara. Tes ini pertama kali dipublikasi pada tahun 1917 di Jepang dan terus digunakan di seluruh dunia sampai sekarang. Tes buta warna *Ishihara* terdiri dari lembaran yang di dalamnya terdapat titik-titik dengan berbagai warna dan ukuran. Titik berwarna tersebut disusun sehingga membentuk lingkaran. Warna titik itu dibuat sedemikian rupa sehingga orang buta warna tidak akan melihat perbedaan warna seperti yang dilihat orang normal (*pseudo-isochromaticism*). Dalam tes buta warna *Ishihara* ini digunakan 38 lembar gambar. Dimana gambar-gambar tersebut memiliki urutan 1 sampai 38 [1].

c. Framework .NET

Microsoft .NET *Framework* (dibaca Microsoft Dot NET *Framework*) merupakan sebuah perangkat lunak kerangka kerja yang berjalan utamanya pada sistem operasi Microsoft Windows, saat ini .NET *Framework* umumnya telah terintegrasi dalam distribusi standar *Windows* (mulai dari *Windows Server 2003* dan versi-versi *Windows* yang lebih baru). Kerangka kerja ini menyediakan sejumlah besar pustaka pemrograman komputer dan mendukung beberapa bahasa pemrograman serta interoperabilitas yang baik sehingga memungkinkan bahasa-bahasa tersebut berfungsi satu dengan lain dalam pengembangan sistem. Perangkat lunak ini adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk *platform* *Windows*. *Framework* .NET dapat memenuhi beberapa tujuan berikut.

1. Menyediakan lingkungan pemrograman berorientasi objek, apakah kode objek disimpan dan dijalankan secara lokal, dijalankan secara lokal tetapi disebarkan melalui internet atau dijalankan secara *remote* (dijalankan dari suatu tempat).
2. Menyediakan lingkungan untuk menjalankan suatu kode yang meminimalkan konflik saat *software deployment* disebarkan dan *versioning*/tentang versi.
3. Menyediakan lingkungan untuk menjalankan suatu kode yang menjamin keamanan saat kode dijalankan, termasuk kode yang dibuat oleh pihak yang tidak diketahui atau pihak ketiga yang setengah dipercaya.
4. Menyediakan lingkungan untuk menjalankan suatu kode yang dapat mengeliminasi masalah performa dari lingkungan *scripted* dan *interpreted*.
5. Membuat pengembang memiliki pengalaman yang konsisten dalam berbagai tipe aplikasi berbasis *Windows* dan aplikasi berbasis *Web* [3].

d. OpenCV dan EmguCV Library

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

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

OpenCV ditujukan untuk menyediakan perangkat yang dapat menjadi dasar mengatasi masalah dalam *computer vision* tanpa adanya keharusan bagi pengembang aplikasi untuk mengembangkan sesuatu yang telah ada. OpenCV dikembangkan pertama kali oleh intel berkerja sama dengan Software Performance Libraries dari Rusia. Dengan tujuan awal pengembangannya adalah sebagai berikut:

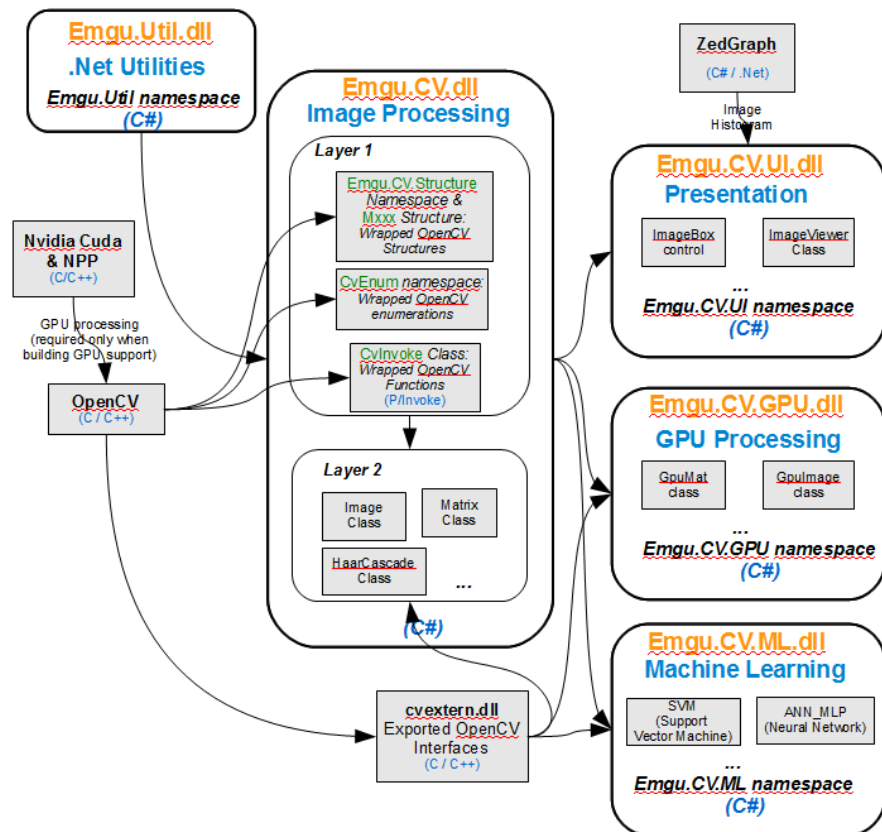
1. Menyediakan infrastruktur dasar untuk *computer vision*.
2. Menyediakan infrastruktur umum yang memungkinkan kode program dapat dengan mudah dibaca dan ditransfer.
3. Menyediakan *library* bersifat gratis dan terbuka yang tidak membutuhkan lisensi komersial.

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

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

EmguCV adalah .NET *wrapper* untuk OpenCV yang bersifat *cross-platform*, sehingga dapat dijalankan pada berbagai sistem operasi. Dengan EmguCV, fungsi-fungsi yang terdapat pada OpenCV dapat dipanggil dari bahasa pemrograman yang kompatibel dengan .NET (.NET-compliant), seperti C#, Visual Basic, Visual C++, dan lain sebagainya. Arsitektur dari EmguCV dapat diilustrasikan pada Gambar 3 [4]. Dari gambar terlihat EmguCV memiliki dua lapisan (*layer*) sebagai berikut:

1. Lapisan dasar (*layer 1*)
Merupakan lapisan yang terdiri atas fungsi, struktur dan enumerasi yang mengacu secara langsung pada yang terdapat pada OpenCV.
2. Lapisan kedua (*layer 2*)
Pada lapisan kedua ini berisi kelas-kelas yang berkaitan dengan .NET.



Gambar 3. Arsitektur EmguCV [4]

e. Pengertian C#

C# (dibaca: “*See-Sharp*”) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET *Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# *Language Specification*, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan simbol # (U+0023) [5].

9. RINGKASAN ISI TUGAS AKHIR

Dalam tugas akhir ini akan dibangun sebuah sistem aplikasi bagi penderita buta warna untuk mempermudah penderita dalam mengidentifikasi warna menggunakan kamera serta dilengkapi fitur tes buta warna dengan menggunakan metode *Ishihara* [1], Deskripsi sistem yang akan dibangun pada aplikasi ini adalah sebagai berikut:

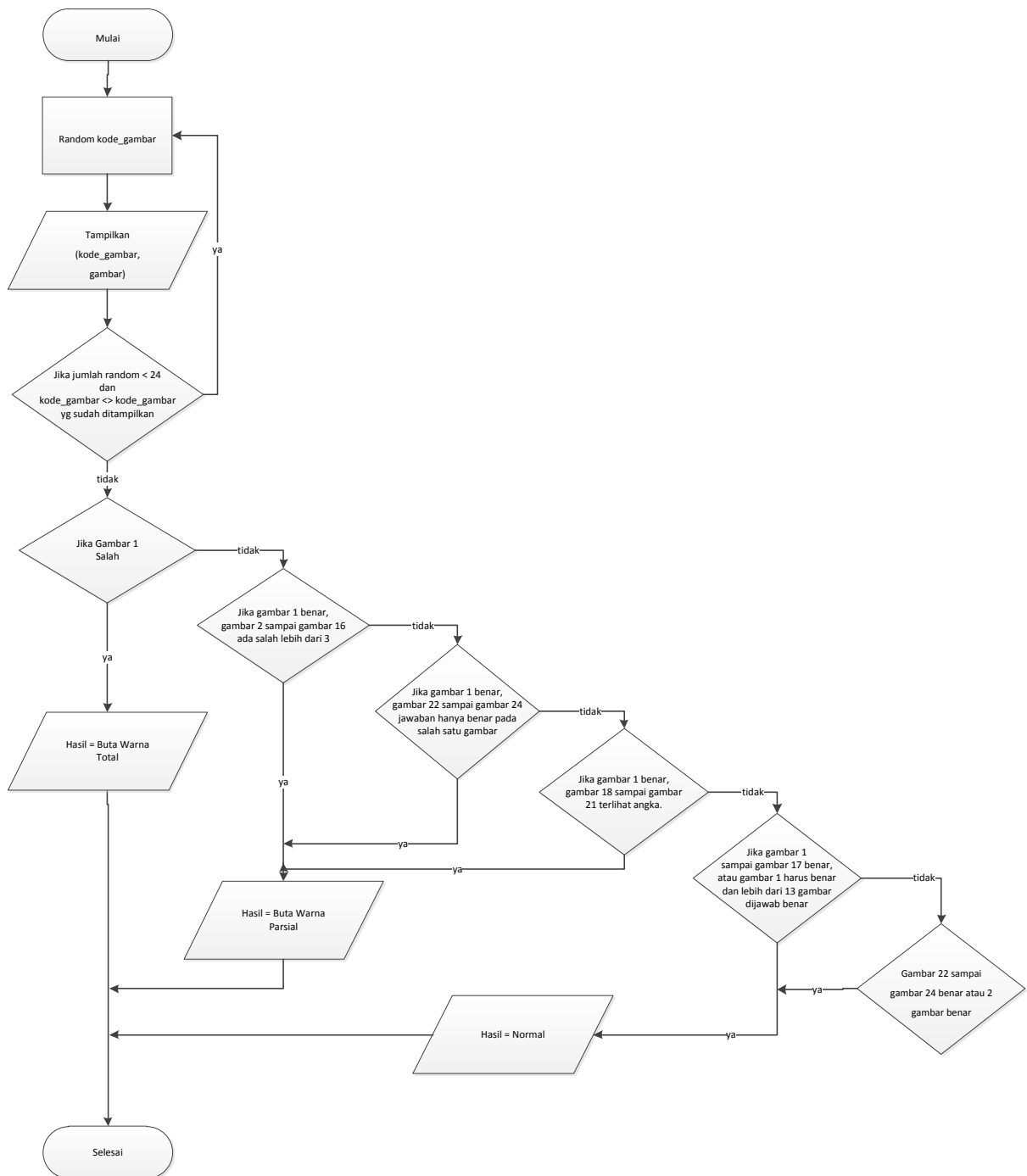
1. Pengguna dapat melakukan pengetesan buta warna dengan metode *Ishihara*.
2. Sistem menampilkan hasil dari tes buta warna yang dilakukan pengguna.
3. Pengguna melakukan pengenalan objek benda melalui kamera agar sistem dapat mendeteksi warna dari objek tersebut.
4. Sistem menampilkan hasil informasi warna yang terdeteksi melalui kamera.

a. Tes Buta Warna Ishihara

Pada tes buta warna *Ishihara* ini digunakan 38 lembar gambar, gambar-gambar tersebut memiliki urutan dari 1 sampai 38. Setiap lembar gambar memiliki bobot nilai masing-masing sehingga didapatkan hasil dengan tiga kesimpulan yaitu:

1. Normal
2. Buta warna Parsial
3. Buta warna Total

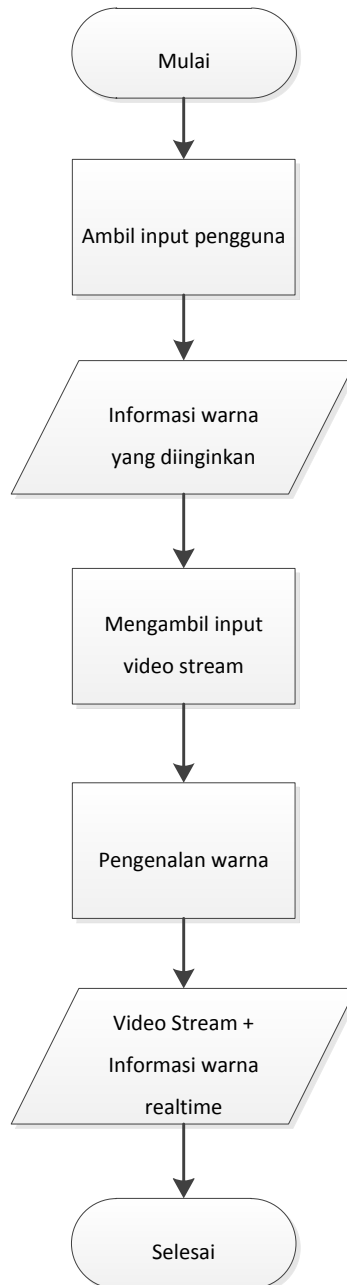
Pada sistem ini, tes buta warna menggunakan 24 lembar gambar saja yang merupakan gambar utama dari tes buta warna *Ishihara*. Dengan gambar tersebut dapat disimpulkan kondisi pengguna apakah buta warna total, parsial atau normal. Dalam proses menampilkan 24 lembar gambar, tes buta warna *Ishihara* ini dapat dilakukan secara urut atau acak. Berikut *flowchart* tes buta warna dengan metode *Ishihara* secara acak (*random*) yang terdapat pada Gambar 4 [1].



Gambar 4. flowchart tes buta warna dengan metode *Ishihara*

b. Sistem pendeteksi warna

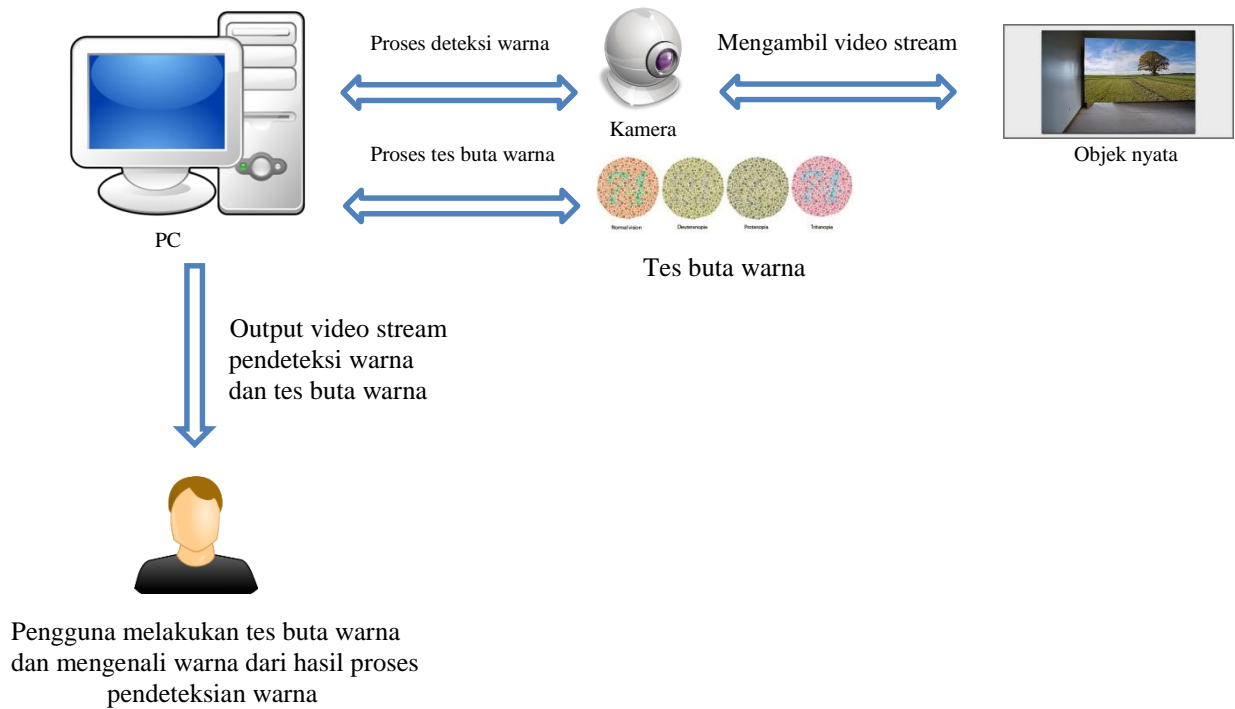
Gambar 5 di bawah ini merupakan diagram alir kerja untuk sistem pendeteksi warna diawali dengan mengambil input dari pengguna kemudian diproses sehingga menghasilkan suatu informasi.



Gambar 5. Diagram alir kerja sistem pendeteksi warna [6]

c. Diagram Arsitektur

Pada Gambar 6 berikut merupakan diagram arsitektur yang menunjukkan cara kerja keseluruhan sistem ini, mulai mengambil video stream dari kamera yang selanjutnya diproses untuk menghasilkan suatu informasi warna yang bermanfaat bagi pengguna dan pengguna juga dapat melakukan tes buta warna.



Gambar 6. Diagram arsitektur

10.METODOLOGI

a. Penyusunan proposal tugas akhir

Pada tahap ini penulis menyusun proposal tugas akhir sebagai langkah awal dalam pengerjaan tugas akhir. Proposal ini juga mengandung proyeksi dari ide tugas akhir yang diajukan. Proposal yang dibuat mengusulkan tentang sistem pendeteksi warna untuk penderita buta warna serta tes buta warna dengan menggunakan metode *Ishihara*.

b. Studi literatur

Pada tahap ini diperlukan adanya pengumpulan informasi dari berbagai sumber terkait dengan metode tes buta warna, pendeteksian warna dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang akan dibangun. Literatur yang digunakan adalah terkait dengan penerapan pendeteksi warna pada kamera video, tes buta warna dan implementasi menggunakan EmguCV.

c. Analisis dan desain perangkat lunak

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Tahap desain meliputi arsitektur perangkat lunak yang digunakan, desain kelas-kelas yang terlibat dalam aplikasi, desain antarmuka, serta diagram-diagram yang mendukung pendeksripsian sistem aplikasi.

d. Implementasi perangkat lunak

Pada tahap ini desain perangkat lunak diwujudkan ke dalam bentuk program. Tahap ini merealisasikan apa yang terdapat pada tahapan sebelumnya sehingga menjadi sebuah aplikasi yang sesuai dengan apa yang telah direncanakan. Rencana pembuatan perangkat lunak ini akan diimplementasikan dengan menggunakan:

1. Bahasa pemrograman yang digunakan adalah C#.
2. *Framework* dan *library* yang digunakan adalah .NET *Framework* dan EmguCV *library*.
3. *Tools* pemrograman yang digunakan adalah Microsoft Visual Studio.
4. *Tools* pemodelan yang digunakan adalah Microsoft Office Visio.

e. Pengujian dan evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan skenario yang telah disiapkan sebelumnya. Uji coba dan evaluasi dilakukan untuk mencari masalah yang mungkin timbul,

mengevaluasi jalannya program, dan mengadakan perbaikan jika diperlukan. Tahapan-tahapan dari uji coba yang akan dilakukan adalah sebagai berikut.

1. Pengujian *blackbox*

Pengujian *blackbox* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengujian pada spesifikasi fungsional program [7]. Pengujian ini dilakukan untuk melihat apakah data yang diminta pengguna dapat muncul secara sesuai atau tidak.

2. Pengujian tingkat akurasi, presisi

Pengujian ini adalah pengujian terhadap tingkat kesuksesan aplikasi dalam mendeteksi warna, sehingga data warna yang diminta dan ditampilkan sesuai dengan keadaan nyatanya.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibangun. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Jadwal rencana pengerjaan tugas akhir ini dijelaskan pada Tabel 1.

Tabel 1. Rencana jadwal pengerjaan tugas akhir

Tahapan	2014																							
	Februari				Maret				April				Mei				Juni							
Penyusunan Proposal																								
Studi Literatur																								
Perancangan sistem																								
Implementasi																								
Pengujian dan evaluasi																								
Penyusunan buku																								

12. DAFTAR PUSTAKA

- [1] R. Widianingsih, A. H. Kridalaksana and A. R. Hakim, "Aplikasi Tes Buta Warna Dengan Metode Ishihara Berbasis Komputer," *Jurnal Informatika Mulawarman*, vol. 5, 2010.
- [2] E. Indrawan, "Perangkat Lunak Transformasi Warna Untuk Penderita Buta Warna," *Tesis Program Magister Informatika ITB*, vol. 2, pp. 15-17, 2008.
- [3] R. Rubiantoro, "Pengertian Net Framework, Fungsi, dan Update Terbaru," [Online]. Available: <http://rubiantororizky.blogspot.com/2013/07/pengertian-net-framework-fungsi-dan.html>. [Accessed 3 Maret 2014].
- [4] "EmguCV," OpenCV, 31 Agustus 2013. [Online]. Available: http://www.emgu.com/wiki/index.php/Main_Page. [Accessed 3 Maret 2014].
- [5] Wikipedia, "C Sharp," Wikipedia, 13 Juni 2013. [Online]. Available: http://id.wikipedia.org/wiki/C_Sharp. [Accessed 3 Maret 2014].
- [6] B. A. Wicaksana and R. F. Sari, "Implementing Text Information Display of Detected Color for Partially Color Blinded Person using .NET Platform and EmguCV Library," in *IEEE International Conference on IT & Multimedia at UNITEN (ICIMU 2011) Malaysia*, Indonesia, 2011.
- [7] S. Rosa Ariani Sukanto, "Black-Box Testing," [Online]. Available: <http://www.gangsir.com/download/6-Black-BoxTesting.pdf>. [Accessed 3 Maret 2014].