

Pembuatan Aplikasi Buta warna untuk penderita buta warna

Adi Hartanto¹, Nur Ismawati²

¹Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

²Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Mercu Buana

E-mail: adi.hartanto@mercubuana.ac.id, nurismawati@mercubuana.ac.id

ABSTRACT

Color blindness is the reason a person is unable to distinguish certain colors. Color blindness is experienced by someone from birth because of lack or absence of pigment in the retinal cone cells of the eye. So that someone will be difficult to distinguish and identify colors. The purpose of making this color-blind application is to help people with color blindness and know the color by making color transfusion and color correction to display names, RGB and HSV values of a color. This Color Blind application is an Android OS application that will be embedded in a smartphone that can be easily used by people with color blindness.

Keywords: *Android, transformation, conversion, color*

ABSTRAK

Buta warna merupakan penyebab seseorang tidak mampu membedakan warna tertentu. Buta warna dialami seseorang sejak lahir karena kurangnya atau tidak adanya pigmen pada sel kerucut retina mata. Sehingga seseorang akan sulit untuk membedakan dan mengenali warna. Tujuan untuk membuat aplikasi buta warna ini adalah membantu penderita buta warna dan dapat mengetahui warna dengan cara memuatkan transformasi warna dan koreksi warna untuk menampilkan nama, nilai RGB dan HSV suatu warna. Aplikasi Buta Warna ini merupakan suatu aplikasi OS Android yang akan ditanamkan pada smartphone yang dapat dengan mudah digunakan oleh penderita buta warna.

Kata kunci: *Android, transformasi, konversi, warna*

1. PENDAHULUAN

Buta warna adalah sebuah ketidakmampuan seseorang dalam hal mengenali warna dengan cara yang normal / biasa, baik itu mengenali akan satu warna atau pun dalam seluruh warna.

Penyebab buta warna atau kondisi akan hal ini adalah kurangnya atau tidak adanya pigmen pada sel kerucut di lapisan retina mata. Fungsi mata khususnya dari pigmen ini sendirilah yang memungkinkan orang dapat mengenali berbagai macam warna.

Warna primer yaitu warna dasar yang dapat memberikan jenis warna yang terlihat dengan campuran ukuran tertentu. Pada sel kerucut terdapat 3 macam pigmen yang dapat membedakan warna dasar merah (Red), hijau (Green) dan biru (Blue).

Bagian sel kerucut adalah sebagai berikut :

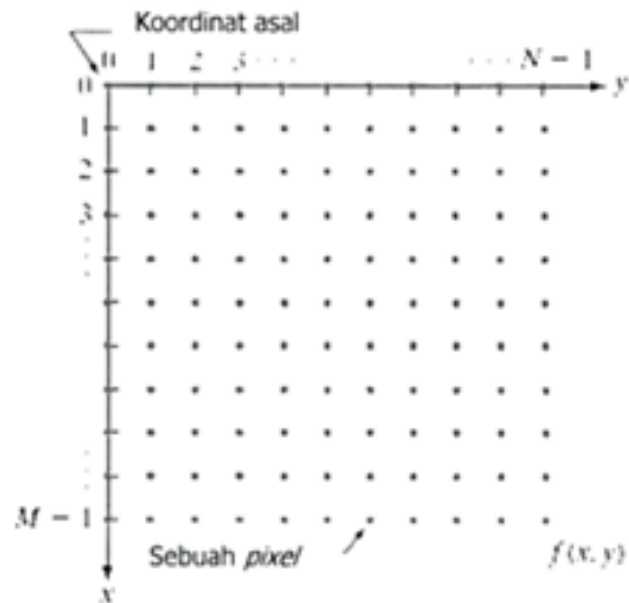
- Sel kerucut yang menyerap long-wavelength light (red)
- Sel kerucut yang menyerap middle-wavelength light (green)
- Sel kerucut yang menyerap short-wavelength light (blue)

Buta warna dikenal berdasarkan istilah Yunani protos (pertama), deutros (kedua), dan tritos (ketiga) yang ada pada warna 1. Merah, 2. Hijau, 3. Biru.

2. TEKNIK TRANSFORMASI WARNA

2.1 Pengolahan Citra

Sebuah citra didefinisikan sebagai fungsi $f(x,y)$ yang berukuran N baris dan M kolom, dimana x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas.



Gambar 1. Koordinat Citra Digital

Citra digital yang berukuran $N \times M$ lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut (Kutipan 1) :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,M) \\ f(1,0) & f(1,1) & f(1,M) \\ f(N-1,0) & f(N-1,1) & f(N-1,M-1) \end{bmatrix} \tag{1}$$

2.2 Akuisisi Citra

Akuisisi citra dilakukan untuk memperoleh data citra digital yang direpresentasikan dalam matriks dua dimensi.



Gambar 2. Hasil sampling citra

2.3 Warna

Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Identitas suatu warna ditentukan panjang gelombang cahaya tersebut. Dalam peralatan optis, warna bisa pula

berarti interpretasi otak terhadap campuran tiga warna primer cahaya: merah, hijau, biru yang digabungkan dalam komposisi tertentu.

2.4 Simulasi Buta Warna

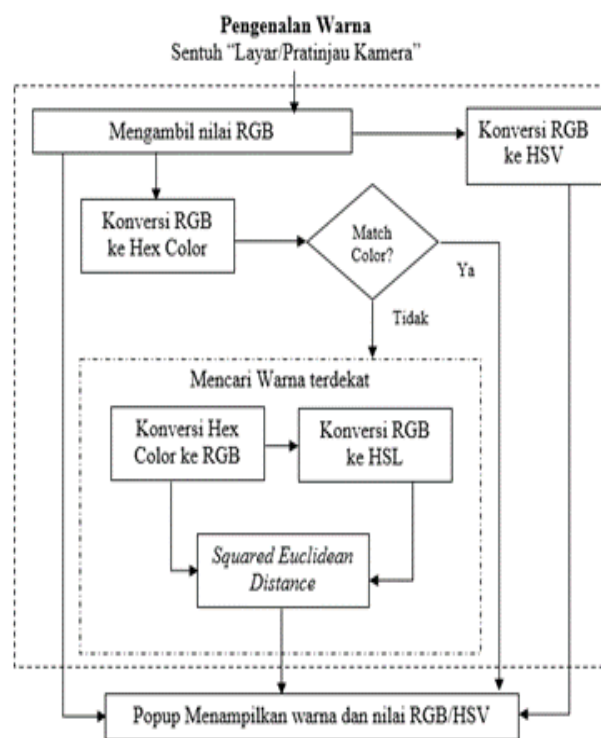
Simulasi penglihatan buta warna dikromasi dengan menggunakan pengolahan citra sudah pernah dilakukan oleh Hans Brettel, Gary W. Meyer and Donald P. Greenberg, Vischeck. Simulasi buta warna yang dilakukan adalah dengan menggunakan teori LMS Cone. Penglihatan warna dicapai melalui L, M dan S kerucut di retina manusia. Reseptor fotosensitif yang sensitif terhadap panjang, menengah dan pendek rentang panjang gelombang dari spektrum yang terlihat, masing-masing.

2.5 Koreksi warna

Daltonization adalah suatu prosedur untuk mengadaptasi warna pada gambar atau urutan gambar untuk meningkatkan persepsi warna dengan menampilkan kekurangan pada warna hasil simulasi buta warna. Metode daltonization adalah menggabungkan RGB ke Matriks transformasi LMS.

3. PERANCANGAN SISTEM

3.1 Pengenalan Warna



Gambar 3. Blok Diagram Proses Pengenalan Warna

Mengambil Nilai RGB

Proses tersebut dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Konversi citra YUV ke bitmap.
2. Ambil nilai RGB dengan pindai bitmap dari kiri ke kanan dan atas ke bawah.
3. Jumlahkan nilai yang terdapat pada komponen R, G, dan B.
4. Hitung jumlah piksel pada bitmap.
5. Hitung rata-rata nilai pada komponen R, G, dan B dengan cara hasil penjumlahan nilai komponen RGB dibagi dengan jumlah piksel.

Konversi RGB ke HSV

Untuk memperoleh nilai yang tepat untuk ruang warna HSV, dihitung satu persatu menggunakan rumus secara bertahap sebagai berikut :

1. Definisikan nilai- nilai untuk S dan V pada HSV berikut :

$$V (Value) = \max \tag{2}$$

$$S (Saturation) = \begin{cases} 0, & \text{jika } \max = \min \\ \frac{\max - \min}{V}, & \text{otherwise} \end{cases} \tag{3}$$

2. Untuk memperoleh nilai Hue [0,360] menggunakan rumus seperti berikut :

$$H (Hue) = \begin{cases} 0, & \text{jika } \max = \min \\ 60^\circ \times \left(\frac{G-B}{\max - \min} \bmod 6 \right), & \text{jika } \max = R \\ 60^\circ \times \left(\frac{B-R}{\max - \min} + 2 \right), & \text{jika } \max = G \\ 60^\circ \times \left(\frac{R-G}{\max - \min} + 4 \right), & \text{jika } \max = B \end{cases} \tag{4}$$

Metode pencarian jarak yang digunakan adalah metode Squared Euclidean Distance.

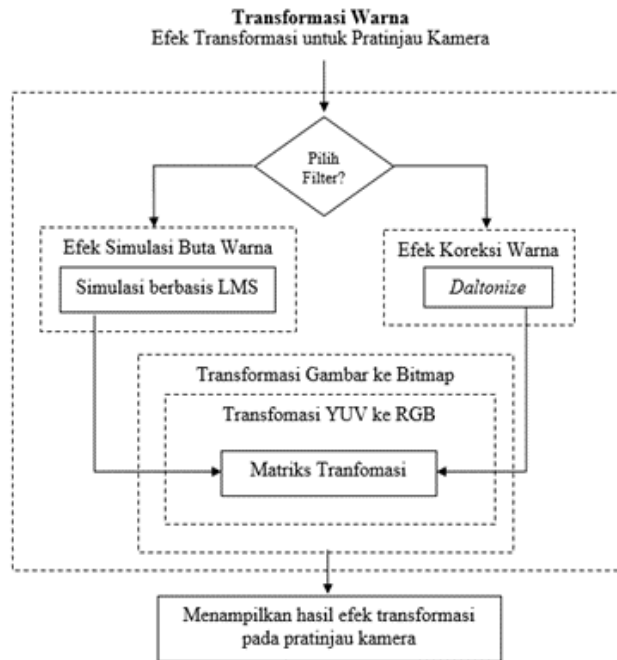
$$CD_{RGB} = (P_r - C_{ir})^2 + (P_g - C_{ig})^2 + (P_b - C_{ib})^2 \tag{5}$$

$$CD_{HSL} = (P_h - C_{ih})^2 + (P_s - C_{is})^2 + (P_l - C_{il})^2 \tag{6}$$

Kemudian nilai jarak terkecil diambil dari hasil color distance RGB ditambah dengan hasil color distance HSL.

$$Dist_{Total} = CD_{RGB} + CD_{HSL} * 2 \tag{7}$$

3.2 Transformasi Warna

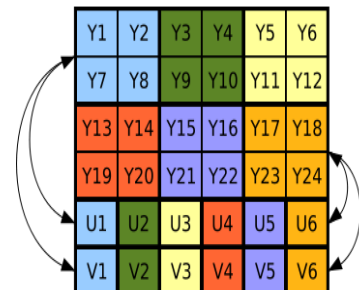


Gambar 4. Blok Diagram Proses Transformasi Warna

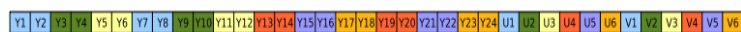
Transformasi YUV ke RGB

Format gambar standar untuk pratinjau kamera pada Android menggunakan YUV420sp (NV21). Gambar planar YUV 4: 2: 0, dengan sampel 8 bit Y, diikuti dengan menyisipkan U/V dengan sampel 8bit 2x2 subsampled kroma seperti pada Gambar 5.

Single Frame YUV420:



Position in byte stream:



Gambar 5. Frame YUV420

Setelah membaca nilai YUV, sistem melakukan transformasi YUV ke RGB dengan menggunakan persamaan berikut :

$$R=Y+(V \times 1.40763)$$

$$G=Y-(U \times 0.345518)-(V \times 0.717)$$

$$B=Y+(U \times 2.03)$$

4. IMPLEMENTASI SISTEM

Efek Simulasi Buta Warna

Pada bagian proses transformasi YUV ke RGB, sistem melakukan transformasi warna untuk memberikan efek simulasi buta warna setelah mendapatkan nilai YUV dan sebelum ditransformasikan ke RGB.

Seluruh proses simulasi buta warna dijelaskan pada langkah-langkah berikut ini untuk setiap piksel gambar yang direpresentasikan pada koordinat RGB :

- Transformasi RGB ke XYZ ke LMS dengan menggunakan matriks transformasi:

$$\begin{pmatrix} L \\ M \\ S \end{pmatrix} = \begin{pmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{8}$$

- Transformasi ruang LMS 3D ke ruang 2D dengan menghitung nilai LMS dikromasi untuk protanopia:

$$\begin{pmatrix} L_p \\ M_p \\ S_p \end{pmatrix} = \begin{pmatrix} 0 & 2.02344 & -2.52581 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{9}$$

- Untuk deuteranopia:

$$\begin{pmatrix} L_D \\ M_D \\ S_D \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.494207 & 1 & 1.24827 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L \\ M \\ S \end{pmatrix} \tag{10}$$

- Invers transformasi LiMiSi ke XYZ ke RGB, dimana $i = \{Protanopia, Deuteranopia\}$ untuk mengembalikan nilai RGB dari nilai LMS dikromasi :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 0.080944 & -0.130504 & 0.116721 \\ -0.0102485 & 0.0540194 & -0.113615 \\ -0.000365294 & -0.00412163 & 0.693513 \end{pmatrix} \begin{pmatrix} L_i \\ M_i \\ S_i \end{pmatrix} \tag{11}$$

- Transformasi YUV ke XYZ ke RGB dengan menggunakan matriks transformasi:

$$\begin{pmatrix} R_{Dic} \\ G_{Dic} \\ B_{Dic} \end{pmatrix} = \begin{pmatrix} 1 & -0.3066864 & -0.47822571 \\ 1 & -0.3066864 & -0.47822571 \\ 1 & 1.7805481 & 0.008514404 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \tag{12}$$

$$\begin{pmatrix} R_{Dic} \\ G_{Dic} \\ B_{Dic} \end{pmatrix} = \begin{pmatrix} 65536 & -20099 & -31341 \\ 65536 & -20099 & -31341 \\ 65536 & 116690 & 558 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \tag{13}$$

- Nilai RGB ditransformasikan pada rentang 2^{16} atau 65536 untuk koordinat warna YUV420 sesuai standar ITU-R BT.601-7 sehingga menghasilkan matriks transformasi YUV ke RGB dengan efek simulasi buta warna:

$$\begin{pmatrix} R_{Dal} \\ G_{Dal} \\ B_{Dal} \end{pmatrix} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0.7 & 0 & 1 \end{pmatrix} \begin{pmatrix} R - R_{Dic} \\ G - G_{Dic} \\ B - B_{Dic} \end{pmatrix} \tag{14}$$



Gambar 6. (Kiri) Hasil Simulasi Buta Warna, (Kanan) Hasil Koreksi Warna

Efek Koreksi Warna

Metode yang digunakan aplikasi untuk memberikan efek koreksi warna agar penderita buta warna merah-hijau dapat membedakan warna dengan menggunakan metode Daltonize.

Untuk proses daltonize akan dijelaskan pada langkah-langkah berikut ini :

- Menghitung matriks kesalahan (error) dengan mencari selisih nilai RGB dari nilai RGB citra asli dikurangi citra simulasi
- Menambahkan matriks error dikalikan dengan matriks daltonize ke nilai RGB

$$\begin{pmatrix} R_{Dal} \\ G_{Dal} \\ B_{Dal} \end{pmatrix} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0.7 & 0 & 1 \end{pmatrix} \begin{pmatrix} R - R_{Dic} \\ G - G_{Dic} \\ B - B_{Dic} \end{pmatrix} \tag{15}$$

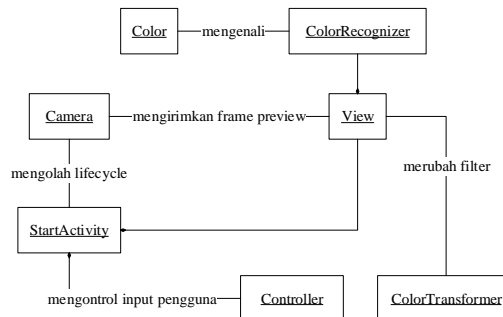


Gambar 7. (Kiri) Hasil Simulasi Buta Warna, (Kanan) Hasil Koreksi Warna

5. DESAIN SISTEM

Arsitektur Sistem

Untuk arsitektur sistem, penulis menggunakan pola MVC, yaitu sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Pada framework Android, sebuah Activity dibutuhkan, yang merupakan inti dari aplikasi. Dari Activity ini, elemen yang dibutuhkan (Controller dan View) dipakai dan dikelola. Controller bereaksi terhadap input pengguna dan merespon Activity pada Start Activity. Start Activity akan mengintruksikan View jika terdapat perubahan. Model domain aplikasi dapat dilihat seperti pada Gambar 8.



Gambar 8. Model Domain Aplikasi ButaWarna

Model domain pada Gambar 8 yang digunakan untuk aplikasi ButaWarna mempunyai 7 objek pada aplikasi, yaitu (1) Controller merupakan antarmuka untuk mengontrol interaksi pengguna. (2) View mempunyai tanggung jawab menampilkan pratinjau kamera dan pengenalan warna ketika menyentuh lokasi koordinat pada gambar. (3) ColorRecognizer merupakan objek pembantu yang berfungsi untuk pengenalan warna. Dan akan berinteraksi dengan objek Color. (4) Color merupakan fungsi pengenalan warna. (5) ColorTransformer bertanggung jawab untuk menangani frame preview pada kamera. Dan menyediakan beberapa filter yang berbeda, yaitu filter simulasi buta warna dan daltonize. (6) StartActivity merupakan inti dari aplikasi yang mengelola lifecycle aplikasi. (7) Camera hanya bertanggung jawab untuk menampilkan frame preview, dan objek ini merupakan properti dari framework Android.

Algoritma Proses Pengenalan Warna

Pada proses pengenalan warna pengguna menyentuh layar handphone terhadap objek yang diinginkan dan aplikasi akan menampilkan pop up atau pesan berisi informasi warna, nilai RGB, dan nilai HSV. Hasil proses pengenalan warna dapat dilihat pada Gambar 9. Dengan menyentuh layar aplikasi untuk menampilkan informasi warna.



Gambar 9. Proses Pengenalan Warna pada Aplikasi

Algoritma Proses Transformasi Warna

Semua proses transformasi warna ditulis dalam bahasa C++ dan di-compile di Android Native Development Kit (NDK) yang mempunyai kinerja lebih baik daripada menggunakan bahasa java untuk proses format kamera android.

Secara default, kamera android menggunakan format warna NV21 (yuv420sp). NV21 adalah format gambar dalam ruang warna YUV dengan kroma subsampling 4:2:0, empat untuk faktor sampling rate dari sinyal pencahayaan Y, dua untuk sinyal U dan dua sinyal V. Sebelum melakukan proses transformasi warna, ruang warna YUV harus dikonversi dulu ke ruang warna RGB pada saat memanggil metode onPreviewFrame yang digunakan untuk menampilkan pratinjau kamera.



Gambar 10. (Kiri) Simulasi Buta Warna, (Kanan) Koreksi Warna

Proses Simulasi Buta Warna

Pada proses simulasi buta warna merah-hijau (red-green colorblind) ini akan diambil informasi tentang RGB gambar yang telah dikonversi dari format YUV. Koordinat warna yang semula RGB akan disesuaikan dan diubah menjadi koordinat LMS. Dengan koordinat LMS, warnanya akan diubah agar dapat dilihat penderita buta warna merah-hijau atau penderita protanopia dan deuteranopia.

Proses Koreksi Warna

Proses koreksi warna menggunakan algoritma daltonize di vischeck digunakan untuk mengubah warna dari pratinjau kamera agar penderita dikromasi bisa melihat warna lebih baik.

6. KESIMPULAN

Dalam penelitian ini, maka dapat disimpulkan hal-hal sebagai berikut :

1. Dengan menggunakan metode YUV, aplikasi buta warna ini dapat diterapkan pada handphone android.
2. Aplikasi buta warna yang telah dibuat dapat menghasilkan simulasi buta warna dan koreksi warna, sehingga akan sangat membantu penderita buta warna untuk mengenali warna dan membedakan warna.

DAFTAR PUSTAKA

- [1] Stockman, D. I. A. MacLeod, and N. E. Johnson (1993), Spectral sensitivities of the human cones," J. Opt. Soc. Am. A 10, 2491–2521
- [2] Acharya, T., (1997). Image Processing Principles and Applications. Willey Interscience
- [3] Benedictus Yoga Budi Putranto, Widi Hapsari, Katon Wijana (2010). Segmentasi warna citra dengan deteksi warna hsv untuk mendeteksi objek diakses tanggal 30 Agustus 2015 dari <http://ti.ukdw.ac.id/ojs/index.php/informatika/article/view/81>
- [4] Castleman K.R. (1996). Digital image processing. Prentice Hall, New Jerse
- [5] Darma Putra (2010). Pengolahan Citra Digital, Penerbit Andi Yogyakarta
- [6] Engineering Guideline EG 28 (1993). "Annotated Glossary of Essential Terms for Electronic Production," SMPTE

- [7] F. Vienot, H. Brettel, J. Mollon (1999). Digital video colourmaps for checking the legibility of displays by dichromats, *Color Research and Application*, vol. 24, pp.243-252
- [8] Forsyth, D.A., Ponce, J. (2003). *Computer vision: A modern approach*. Prentice Hall, New Jersey
- [9] Giannakopoulos, T. (2008). *Matlab color detection software*, Department of Informatics and Telecommunications, University of Athens, Greece
- [10] Gdansk Univ. of Technol., Gdańsk, Poland (2010). Human System Interactions (HSI), Color Transformation Methods for Dichromats diakses tanggal 30 Agustus 2015 dari http://www.researchgate.net/publication/224158059_Color_transformation_methods_for_dichromats
- [11] G. R. Kuhn, M. M. Oliveira, and L. A. F. Fernandes (2008). An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE TVCG*, 14(6):1747–1754
- [12] G. Sharma, W. Wu, E. N. Dalal (2004). The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations, *Color Research & Application*, vol. 30, pp. 21 – 30
- [13] Hans Brettel, Françoise Vienot, John Mollon, (1997). "Computerized simulation of color appearance for dichromats ", *J. Opt. Soc. Am. A*, Vol. 14, No. 10, pp. 2647-2655
- [14] J.B. Huang, Yu-Cheng Tseng, Se-In Wu, and Sheng-Jyh Wang (2007). Information Preserving Color Transformation for Protanopia and Deuteranopia, *IEEE Signal Processing Letters*, vol. 14, No. 10, pp.711-714
- [15] J.D.Mollon (2015). Computerized simulation of color appearance for dichromats diakses tanggal 22 Agustus 2015 dari http://vision.psychol.cam.ac.uk/jdmollon/papers/Dichromat_simulation.pdf
- [16] J. Ruminski, J. Wtorek, J. Ruminska, M. Kaczmarek, A. Bujnowski, T. Kocejko, A. Polinski, (2010). "Color transformation methods for dichromats", *Proc. of 3rd Conference on Human System Interactions (HSI'10)*, pp. 634-641
- [17] Rahmadi Kurnia (2009). Penentuan tingkat buta warna berbasis his pada citra ishihara, Jurusan Elektro Fakultas Teknik Universitas Andalas, Padang
- [18] Ruchi Kulshrestha, R.K. Bairwa (2013). Review of Color Blindness Removal Methods using Image Processing diakses tanggal 1 September 2015 dari <http://www.ijrr.com/papers6/paper4.pdf>