

Implementasi Rasio Kontras Warna pada Aplikasi Uji Buta Warna 100-Hue Farnsworth-Munsell Berbasis Android

Implementation of Color Contrast Ratio on Android-based Farnsworth-Munsell 100-Hue Color Blindness Test Application

Fredicia¹, Ferdian Leonarta²

^{1,2}Program Studi Informatika, Universitas Kristen Krida Wacana

E-mail: ¹fredi.cia@ukrida.ac.id, ²ferdian.2014tin004@civitas.ukrida.ac.id

Abstrak

Penelitian ini menginvestigasi ketidakmampuan individu dalam mendeteksi suatu warna yang sering tidak disadari oleh penderita buta warna. Meskipun penderita masih dapat membedakan benda berdasarkan ukuran, bentuk, dan kecerahannya, kondisi buta warna tidak mengancam jiwa secara medis. Salah satu metode yang digunakan untuk menguji buta warna adalah metode Farnsworth Munsell, yang dikembangkan oleh Dean Farnsworth dan Albert H. Munsell dengan cara menyusun *chip* warna sebanyak 85 buah pada 4 buah *pallette* sesuai urutannya. Dengan memanfaatkan panduan desain antarmuka untuk pengguna yang kemungkinan merupakan penderita buta warna, antarmuka pengguna (UI) aplikasi Android dengan rasio kontras warna *foreground* dan *background* yang digunakan adalah sebesar 8.1. Keterlibatan 38 responden, termasuk 3 orang penderita buta warna menunjukkan aplikasi uji buta warna dengan UI yang didukung warna kontras dapat mendeteksi kondisi buta warna dengan baik pada penderita yang diuji. Dari responden juga, penelitian ini mendapatkan umpan balik yang positif dari kenyamanan untuk dilihat, kemudahan untuk menggunakan dan mengoperasikan, instruksi dalam aplikasi mudah dimengerti.

Kata kunci: 100 Hue Farnsworth-Munsell, Android, Aplikasi Uji Buta Warna, Buta Warna, Desain Antarmuka

Abstract

This research investigates individuals' inability to detect certain colors, which is often unnoticed by color-blind patients. Although color-blind individuals can still differentiate objects based on their size, shape, and brightness, the condition does not pose a medical threat to their lives. One of the methods used to test color blindness is the Farnsworth-Munsell method, developed by Dean Farnsworth and Albert H. Munsell, which involves arranging 85 color chips in four palettes according to their sequence. By utilizing user interface (UI) design guidelines for individuals who may be color-blind, the Android application's UI employs a color contrast ratio of 8.1 for the foreground and background colors. Involving 38 respondents, including 3 color-blind individuals, the color blindness test application with a contrast-supported UI proved to effectively detect color blindness in the tested individuals. The respondents also provided positive feedback on the comfortability of viewing the app, ease of use and operation, as well as the comprehensibility of the instructions within the application.

Keywords: 100 Hue Farnsworth-Munsell, Android, Color Blindness Test Application, Color Vision Deficiency, UI Design

1. PENDAHULUAN

Ketidakmampuan dalam melihat suatu warna sering tidak disadari oleh penderita karena masih bisa membedakan benda dari ukuran, bentuk dan kecerahannya serta secara medis buta

warna tidak mengancam jiwa. Faktor genetik, penyakit mata ataupun cedera mata merupakan penyebab seseorang mengalami buta warna. Gejala yang dialami setiap penderita berbeda dimana terdapat penderita buta warna hanya dapat melihat gradasi warna tertentu. Ketidakmampuan ini akan menyebabkan dalam melakukan aktivitas sehari-hari, seperti tidak bisa membedakan lampu lalu lintas, pekerjaan yang membutuhkan kemampuan untuk melihat warna untuk menyusun urutan warna tertentu. Buta warna dapat diagnosa dengan baik jika dilakukan diagnosa dengan metode uji buta warna.

Studi Literatur terkait penelitian metode uji buta warna yang sering digunakan dalam metode Ishihara dibandingkan metode 100 Hue Farnsworth-Munsell. Metode Ishihara sering digunakan karena proses uji buta warna cepat dan hanya menentukan jenis buta warna yang dialami, yaitu buta warna total, buta warna parsial dan mata normal [1]. Sedangkan metode 100 Hue Farnsworth-Munsell memiliki keunggulan dalam menguji buta warna dengan hasil lebih detil dengan informasi kelemahan penderita buta warna berada pada area warna tertentu [2][3].

Metode 100 Hue Farnsworth-Munsell dikembangkan oleh Dean Farnsworth dan Albert H. Munsell dimana setiap warna yang disusun dalam 4 buah pallete berisi gradasi warna yang memanfaatkan nilai *hue* (nilai warna dominan), *value* (nilai kecerahan warna) dan *chroma* (nilai intensitas warna). Ketidakmampuan pasien dalam melihat warna tertentu dinilai dari jumlah *error* dalam mengurutkan warna pada palette dan warna yang memiliki nilai *error*. Metode 100-Hue Farnsworth Munsell merupakan tes *psychotechnical* yang menentukan kemampuan seseorang untuk membedakan gradasi dari sebuah pigmen warna. Tes ini terdiri dari 85 chip warna yang didesain untuk menentukan rata-rata perbedaan gradasi warna yang dapat dilihat oleh orang dengan penglihatan normal. 85 chip tersebut disusun dalam empat buah palette seperti pada gambar 1. Penderita buta warna akan melakukan kesalahan dalam menyusun chip tersebut. Semakin banyak kesalahan dalam penyusunan chip, semakin parah buta warna yang diderita oleh orang yang melakukan tes ini [4].



Gambar 1. Peralatan tes 100-Hue Farnsworth-Munsell

Walaupun terdapat metode uji buta warna, jumlah penderita buta warna di Indonesia tidak diketahui secara pasti. Sebagai perbandingan, perkiraan penderita buta warna pria di seluruh dunia sebesar 8% dengan jumlah penderita buta warna wanita sebesar 0.4% [5]. Ketidaktahuan setiap penderita buta warna menderita buta warna dengan menggunakan teknologi informasi akan membantu metode uji buta warna dapat digunakan, terutama media tes yang digunakan sebelumnya masih menggunakan media cetak. Salah satu teknologi informasi yang saat ini banyak digunakan adalah perangkat ponsel pintar dengan perkiraan pengguna ponsel pintar di Indonesia sebanyak 192,15 juta orang [4].

Sebelum dilakukan penelitian ini, perlu menggali lebih banyak informasi penelitian yang terkait yang dapat membantu tahapan penelitian ini dan bisa memperlihatkan perbedaan yang akan dilakukan. Banyak penelitian metode uji buta warna yang dilakukan untuk mengetahui seberapa efektif implementasi teknologi informasi bersama metode tersebut. Pada tahun 2020, [2] menerapkan teknologi informasi yaitu ponsel pintar berbasis Android sebagai media tes buta warna dengan metode Farnsworth Munsell yang membantu proses perhitungan skor *error* dengan mudah. Temuan lain yang ditemukan oleh Ardi dan Muntahanah adalah

tampilan UI yang digunakan masih sederhana dan belum memenuhi kategori User Experience yang baik.

Studi pengembangan model penggunaan warna yang cocok untuk penderita buta warna yang dilakukan [6][7] mendapatkan bahwa pengguna aplikasi yang merupakan penderita buta warna perlu desain aplikasi yang menerapkan warna yang kontras. Sistem operasi yang digunakan pada perangkat seperti ponsel pintar dan komputer sudah disematkan fungsi pendukung untuk penderita buta warna, salah satunya sistem operasi Android yang dibuat oleh Google. Akan tetapi, permasalahan setiap antar muka pada aplikasi ponsel pintar masih ditemukan pada saat digunakan oleh penderita buta warna [8] yaitu kesulitan menggunakan aplikasi ponsel pintar.

[9] juga melakukan penelitian untuk mengembangkan aplikasi tes buta warna berbasis web pada tahun 2019. Tingkat akurasi sebesar 96.8% didapatkan dengan menerapkan metode Ishihara untuk penderita buta warna total dan metode Farnsworth Munsell D-15 untuk penderita buta warna parsial. Penelitian Rico Estrada et al. juga mendapatkan temuan agar aplikasi uji buta warna yang dikembangkan bisa digunakan dalam teknologi ponsel pintar dan menggunakan chip warna yang lebih banyak.

Dalam penelitian [5], penerapan teknologi pengolahan citra mampu membantu penggunaan aplikasi pada penderita buta warna sebesar 95.1%. Di tahun 2022, [10] mendapatkan temuan penerapan uji buta warna menggunakan teknologi bisa membuat fleksibilitas dalam penggunaan alat uji buta warna tetapi memerlukan koneksi internet karena menggunakan Google Form sebagai media penyampaian uji buta warna. Temuan ini sebenarnya sudah didukung dalam penelitian yang dilakukan oleh [11][12] menyimpulkan bahwa pentingnya instrumen pengujian buta warna otomatis dibandingkan menggunakan instrumen konvensional yang bisa rusak atau hilang.

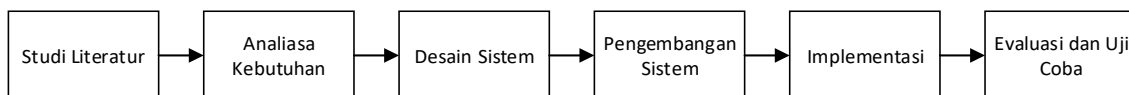
Dari latar belakang dan penelitian terkait yang pernah dilakukan, terdapat temuan dari penelitian untuk aplikasi uji buta warna yang ada yaitu:

1. Adanya metode uji buta warna, 100 Hue Farnsworth Munsell memiliki chip warna sebanyak 85 buah dibandingkan metode Farnsworth D-15 yang bisa menentukan lebih detail buta warna yang diderita.
2. Penggunaan warna kontras dalam tampilan antarmuka dapat membantu pengguna aplikasi yang kemungkinan merupakan penderita buta warna.
3. Perangkat teknologi informasi, yaitu ponsel pintar bisa digunakan sebagai media untuk uji buta warna sangat membantu dibandingkan menggunakan instrumen uji buta warna dalam bentuk fisik yang bisa rusak atau hilang.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian tentang penerapan teknologi informasi dalam mendeteksi buta warna dimulai dalam tahapan-tahapan yang terlihat pada gambar. 2.



Gambar 2. Tahapan Penelitian

Penelitian tentang penerapan teknologi informasi dalam mendeteksi buta warna dengan menggunakan metode Farnsworth Munsell memerlukan studi literatur agar mengetahui bagaimana cara kerja metode Farnsworth Munsell dan bisa mengetahui bagaimana mendesain aplikasi uji buta warna dan menerapkan desain berdasarkan kebutuhan pengguna. Metode Farnsworth Munsell menggunakan palette warna dengan chip warna sebanyak 85 buah. Desain palette dengan 85 buah chip warna terlihat pada gambar 3..



Gambar 3. Gambaran Susunan Chip Warna pada Palette

Setiap baris akan disusun *chip* warna secara acak kecuali *chip* warna yang terletak pada posisi paling kiri dan paling kanan. Pengguna aplikasi akan mengurutkan setiap *chip* warna yang diberikan setiap baris berdasarkan *chip* warna paling kiri dan paling kanan. Setiap *chip* memiliki nomor warna yang akan digunakan untuk menghitung nilai *error* dalam pengurutan warna yang dilakukan oleh pengguna aplikasi. Cara menghitung nilai *error* dapat dilihat pada persamaan (1).

$$CE_j = |C_j - C_{j-1}| + |C_j - C_{j+1}| - 2 \quad (1)$$

dengan CE_j merupakan nilai *error* posisi *chip* warna ke- j dan C_j merupakan nilai nomor warna *chip* pada posisi ke- j . [2] Kemudian setelah mendapatkan nilai *error* setiap posisi warna pada *pallette*, maka akan dijumlah dengan persamaan (2).

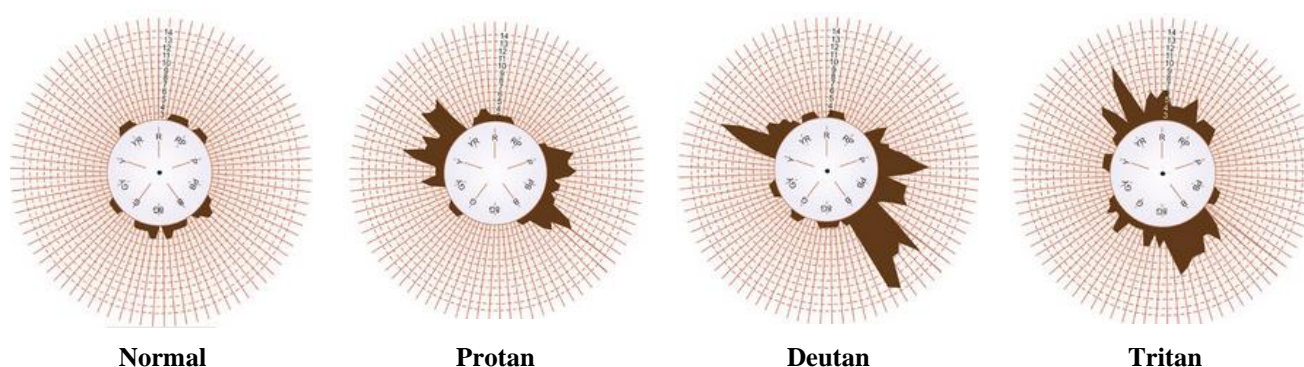
$$TES = \sum_{i=1}^4 \left(\sum_{j=1}^n CE_j \right) \quad (2)$$

dengan i merupakan posisi *chip* warna pada *pallette* ($i = 1$ merupakan *pallette* baris pertama dst.) dan n merupakan jumlah *pallette* pada baris tersebut.

Hasil penyusunan warna dan nilai *error* yang dihasilkan akan digambarkan dalam bentuk grafik yang memiliki 85 *chip* warna. Hasil detail tes berupa bagian warna apa saja yang mengalami *error* dan jenis *Color Vision Deficiency* (CVD) yang diderita oleh pengguna aplikasi akan ditentukan menggunakan grafik yang berisi 85 persegi warna beserta jumlah *error* dari masing-masing warna. Dari grafik tersebut dapat dilihat dan ditentukan seberapa parah buta warna yang diderita oleh pengguna aplikasi dan jenis CVD apa yang diderita oleh pengguna aplikasi.

Dalam penelitian Deans Farnsworth (1943)[13], jenis CVD yang diderita oleh penderita warna dapat dilihat dari kesalahan pengurutan *chip* warna sebagai berikut:

1. *Protan* sering terjadi pada kesalahan pengurutan *chip* warna ke-62 s/d 70 dan *chip* warna ke-16 s/d 22.
2. *Deutran* sering terjadi pada kesalahan pengurutan *chip* warna ke-56 s/d 61 dan *chip* warna ke-14 s/d 18.
3. *Tritan* sering terjadi pada kesalahan pengurutan *chip* warna ke-46 s/d 52 dan *chip* warna ke-84 s/d 4.



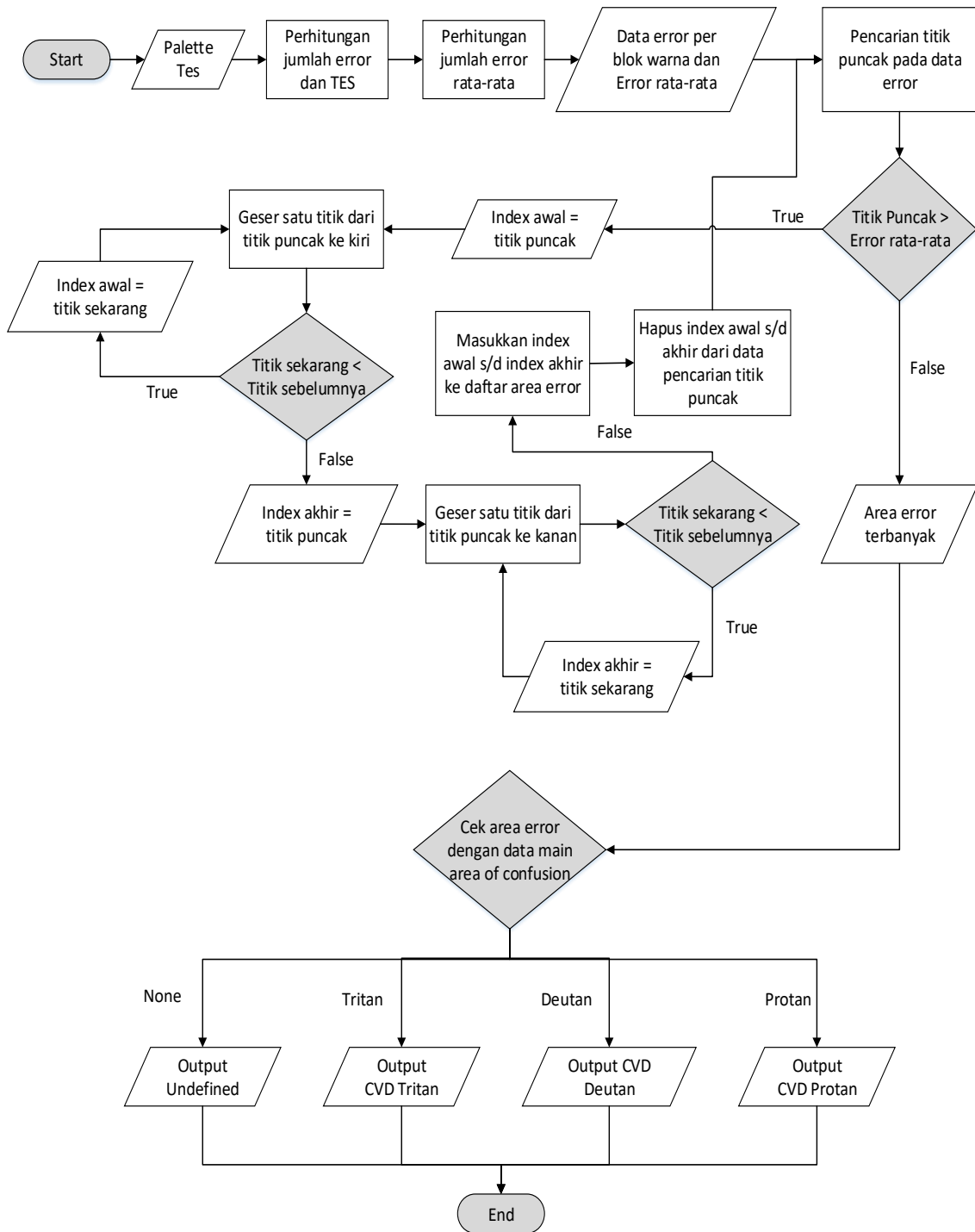
Gambar 4. Gambaran Grafik Hasil Pengujian CVD

Tahapan dalam menentukan tipe CVD yang diderita oleh pengguna aplikasi dapat dilihat pada *flowchart* di gambar 5. Adapun tahapan-tahapan aplikasi mengenali tipe CVD adalah sebagai berikut :

1. Aplikasi akan menghitung jumlah error pada setiap blok warna dan jumlah TES dari tes dengan menggunakan rumus (1) dan (2)
2. Setelah aplikasi menghitung jumlah error pada setiap blok warna, komputer akan mencari area-area dengan error terbanyak. aplikasi akan menentukan terlebih dahulu jumlah error rata-rata sebagai batas penentu minimum jumlah error yang dapat dimasukkan dalam area tersebut.
3. Pencarian area-area dengan error terbesar dilakukan dengan menggunakan algoritma Hill Climbing yang dimodifikasi untuk melakukan iterasi dari titik puncak yang kemudian turun ke bawah. Hal ini dilakukan agar area dengan jumlah error dibawah rata-rata tidak dimasukkan ke dalam area tersebut. Aplikasi akan mencari titik tertinggi dari grafik hasil test dimana titik tertinggi tersebut tidak boleh kurang dari selisih rata-rata yang sudah dihitung sebelumnya. Setelah aplikasi menemukan titik puncak, aplikasi akan melakukan pergerakan turun dari titik puncak untuk mendapatkan area dengan error terbanyak melalui titik puncak tersebut. Apabila aplikasi sudah tidak dapat melakukan pergerakan turun, maka komputer akan kembali ke titik puncak dan melakukan pergerakan turun lagi dari arah yang berlawanan dari arah sebelumnya. Setelah area “gunung” ditemukan oleh aplikasi dari titik puncak tersebut, “gunung” tersebut akan dimasukkan ke dalam array data yang mengandung area-area dengan selisih terbesar dan “gunung” tersebut juga akan dikeluarkan dari pencarian area pada iterasi berikutnya. Aplikasi akan mengulangi semua tahap ini sampai tidak dapat ditemukan lagi area dengan selisih tertinggi.
4. Setelah area-area tersebut ditemukan, maka aplikasi akan menentukan tipe CVD dengan menjumlahkan jumlah error pada setiap blok warna dalam area tersebut ke dalam setiap area warna dalam main area of confusion pada gambar 3.6 diatas. Main area of confusion dengan jumlah error paling dominan akan menjadi tipe CVD yang diderita oleh pengguna aplikasi. Apabila skor yang didapatkan oleh pengguna berjumlah dibawah 100 maka tipe CVD nya adalah “None”.

Sedangkan untuk tingkat keparahan buta warna yang dialami pasien, Deans Farnsworth mengelompokkan kedalam 3 klasifikasi tingkat keparahan buta warna berdasarkan nilai *Total Error Score* (TES), yaitu:

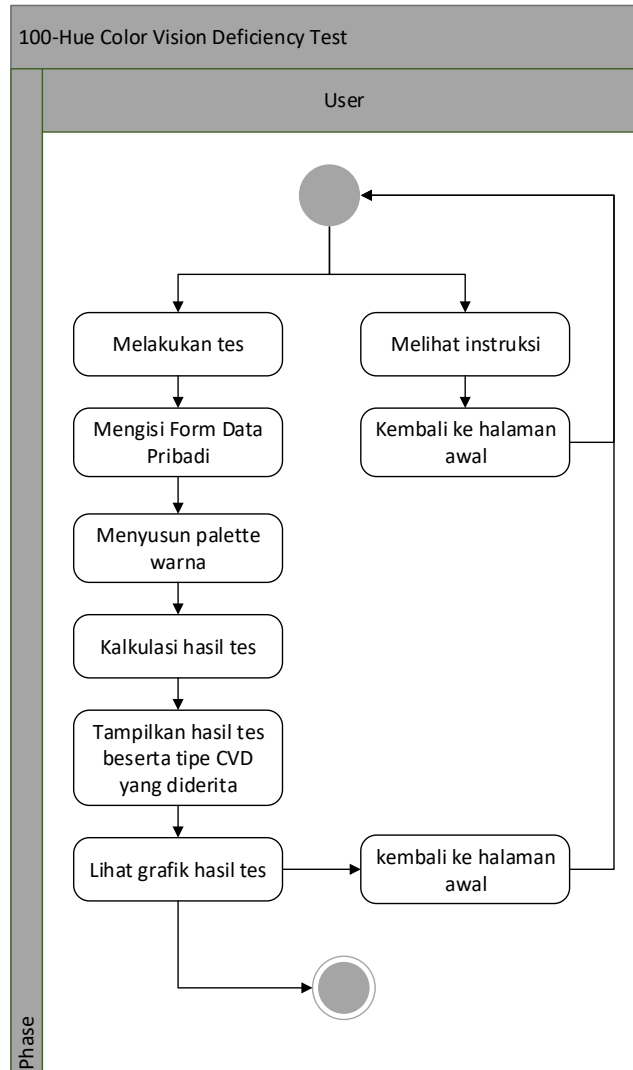
1. *Good Color Discrimination*, dengan nilai TES dibawah dari 16 poin.
2. *Average Color Discrimination*, dengan nilai TES diantara 17- 100 poin.
3. *Bad Color Discrimination*, dengan nilai TES diatas dari 100 poin.



Gambar 5. Alur Penentuan Tipe CVD

2.2 Perancangan Aplikasi

Rancangan aplikasi uji buta warna yang akan menerapkan 100 Hue Farnsworth Munsell dimulai melakukan aktivitas pengguna yang dapat dilihat pada activity diagram di gambar 6.

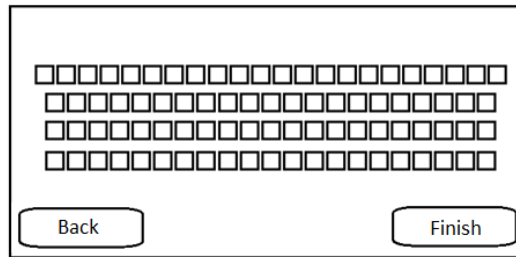


Gambar 6. Activity Diagram Aplikasi

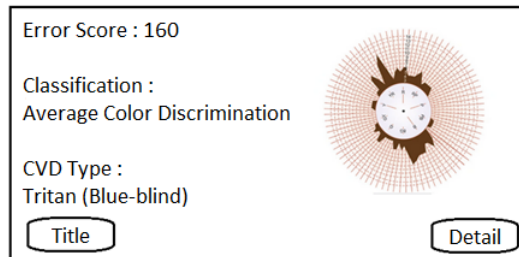
Dalam penelitian yang dilakukan oleh Aytac (2017)[6], kontras warna di antarmuka aplikasi akan sangat membantu penderita buta warna dalam menelusuri aplikasi. Pada penelitian yang dilakukan oleh Eler et al.[14](2018), minimal nilai rasio kontras warna foreground dan background adalah 3(tiga). Berikut tampilan desain antarmuka aplikasi yang digunakan sebagai media pengujian buta warna dapat dilihat pada gambar 7, gambar 8 dan gambar 9.



Gambar 7. Desain Antarmuka Halaman Utama Aplikasi Uji Buta Warna



Gambar 8. Desain Antarmuka Halaman Uji Buta Warna Aplikasi Uji Buta Warna



Gambar 9. Desain Antarmuka Halaman Hasil Uji Buta Warna Aplikasi Uji Buta Warna

3. HASIL DAN PEMBAHASAN

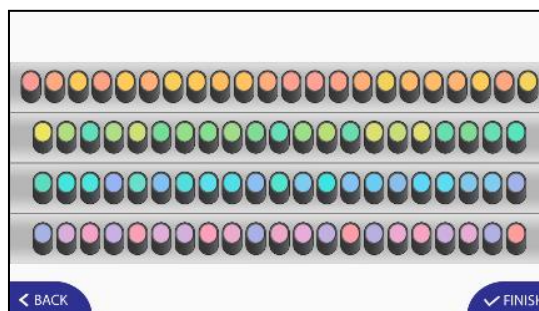
Berdasarkan rancangan aplikasi uji buta warna yang telah dibuat sebelumnya, aplikasi uji buta warna yang dihasilkan adalah sebagai berikut:

a. Antarmuka Aplikasi Uji Buta Warna

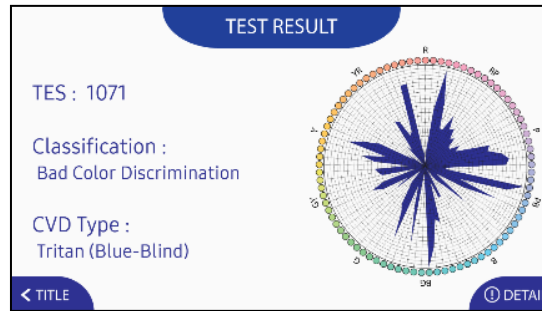
Untuk membantu penggunaan aplikasi uji buta warna, terutama pengguna adalah penderita buta warna, aplikasi didesain antarmuka aplikasi menggunakan warna biru (#2E3192) dan putih (#FFFFFF) agar dapat membantu pengguna dengan memanfaatkan kontras warna yang bisa dilihat pada gambar 8.



Gambar 10. Antarmuka Halaman Utama Aplikasi Uji Buta Warna



Gambar 11. Antarmuka Halaman Uji Buta Warna



Gambar 12. Antarmuka Halaman Hasil Uji Buta Warna

Untuk mengetahui rasio kontras terhadap warna yang digunakan pada aplikasi, penelitian ini memanfaatkan aplikasi pengecekan rasio kontras berbasis *web*, Contrast Checker [15]. Aplikasi ini merupakan aplikasi pengecekan rasio kontras terhadap desain *web* yang mengikuti panduan WCAG (Web Content Accessibility Guidelines)[16]. Beberapa penelitian tentang aksesibilitas aplikasi ponsel pintar menggunakan panduan WCAG, yaitu Ballantyne [17], Vargas [18] dan [19]. Hasil pengecekan rasio kontras warna yang digunakan pada aplikasi uji buta warna sebesar 8.59. Jika mengacu pada penelitian yang dilakukan oleh Eler et al. [14] dimana nilai rasio kontras warna sebesar 3, maka warna yang digunakan sudah memenuhi aturan penggunaan warna yang diperlukan dalam membantu penderita buta warna menggunakan aplikasi.

Pada tabel 1 menampilkan hasil pengujian antarmuka untuk ukuran dan resolusi layar *ponsel pintar* tertentu. Untuk ukuran layar 5 s/d 5.5 inch, tampilan antarmuka pada layar sesuai dengan pada saat melakukan perancangan antarmuka aplikasi. Sedangkan untuk ukuran 6 inch, tampilan antarmuka untuk halaman uji buta warna, *chip* warna pada *pallette* terlihat tidak proposional dengan ukuran layar yang disebabkan pada saat ponsel pintar melakukan *scaling* pada objek yang ditampilkan tidak melakukan *scaling* pada objek *chip* warna.

Tabel 1 Hasil Pengujian Antarmuka dengan Ukuran dan Resolusi Layar

Ukuran Layar	Hasil Pengujian Antarmuka
5.0 inch (1280 x 760 pixels)	Tampilan antarmuka pada layar tetap terlihat rapi dan konsisten.
5.5 inch (1280 x 760 pixels)	
5.5 inch (1920 x 1080 pixels)	
6.0 inch (2220 x 1080 pixels)	Tampilan antarmuka pada layar ini terlihat rapi namun pada bagian tes <i>chip</i> warna terlihat kecil dikarenakan lebar layar terlalu kecil dibandingkan dengan panjang layar sehingga objek <i>palette</i> pada bagian tes tidak dapat melakukan <i>scaling</i> dengan baik.

b. Cara Perhitungan Aplikasi Uji Buta Warna

Pada halaman uji buta warna, aplikasi dibuat dengan memiliki empat buah *pallette* seperti pada gambar 8. *Palette* warna baris pertama akan disusun 22 *chip* warna, sedangkan baris kedua s/d keempat memiliki 21 *chip* warna. Setiap *chip* warna memiliki penomoran yang merupakan posisi *chip* warna sebenarnya. Sebagai contoh, susunan *chip* warna yang dilakukan oleh pengguna aplikasi uji buta warna terdapat pada gambar 9 dengan nilai penomoran sebagai berikut:

[85, 6, 18, 4, 17, 9, 19, 15, 13, 14, 7, 2, 1, 3, 8, 20, 11, 12, 10, 16, 5, 21]
 [22, 28, 41, 29, 25, 36, 32, 33, 30, 35, 40, 31, 27, 38, 24, 26, 23, 37, 34, 39, 42]
 [43, 48, 46, 62, 44, 59, 49, 51, 50, 61, 45, 55, 47, 58, 54, 60, 53, 52, 57, 56, 63]
 [64, 72, 78, 69, 82, 73, 71, 81, 77, 65, 79, 74, 68, 83, 67, 76, 80, 70, 75, 66, 84]

Gambar 9. Screenshot Susunan *Chip* Warna Pengguna Aplikasi Uji Buta Warna

Perhitungan *error* setiap *chip* warna dilakukan menggunakan rumus (1), sedangkan untuk menghitung total *error score* dari hasil pengurutan warna yang dilakukan oleh pengguna menggunakan rumus (2). Sebagai contoh, perhitungan *error* dari 5 buah *chip* warna hasil uji pada *pallette* pertama ditunjukkan pada tabel 2.

Tabel 2 Perhitungan Error dari 5 buah Chip Warna Pallette Baris Pertama

Chip Ke	Ref	Tes	Cara Menghitung
1	85	85	$ 85 - 84 + 0 - 6 - 2 = 5$
2	1	6	$ 6 - 0 + 6 - 18 - 2 = 16$
3	2	18	$ 18 - 6 + 18 - 4 - 2 = 24$
4	3	4	$ 4 - 18 + 4 - 17 - 2 = 25$
5	4	17	$ 17 - 4 + 17 - 9 - 2 = 19$

c. Akurasi Uji Buta Warna

Setelah aplikasi uji buta warna selesai dikembangkan, dilakukan pengujian akurasi dari aplikasi dalam mendeteksi buta warna yang diderita dengan jumlah responden sebanyak 38 orang diantaranya terdapat 3 orang penderita buta warna. Sebelum melakukan pengujian, responden mengisi data lewat formulir untuk mendapatkan informasi apakah merupakan penderita buta warna atau tidak. Data yang didapatkan akan dibandingkan dengan pengujian aplikasi uji buta warna yang dikembangkan dapat dilihat pada tabel III. Responden yang menjawab bukan penderita buta warna memiliki nilai yang sama untuk tingkat keparahan CVD “Good Color Discrimination” dan “Average Color Discrimination”. Dari pengujian tersebut, didapatkan aplikasi uji buta warna dapat mendiagnosa penderita buta warna dengan baik.

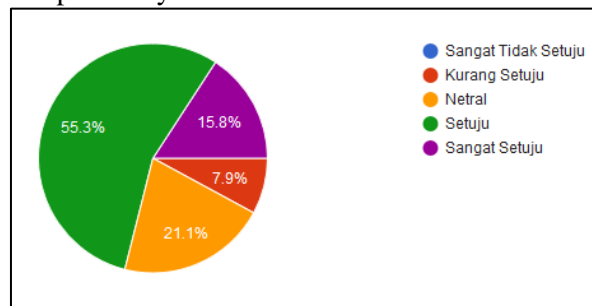
Tabel 3 Perbandingan Hasil Pengujian Aplikasi Uji Buta Warna

Tingkat Keparahan CVD	Sistem Komputer	Formulir
Good Color Discrimination	5	35
Average Color Discrimination	30	0
Bad Color Discrimination	3	3

d. Pengujian Usability Aplikasi

Untuk mengetahui kinerja aplikasi uji buta warna, setiap responden memberikan umpan balik dengan menjawab pertanyaan sebagai berikut:

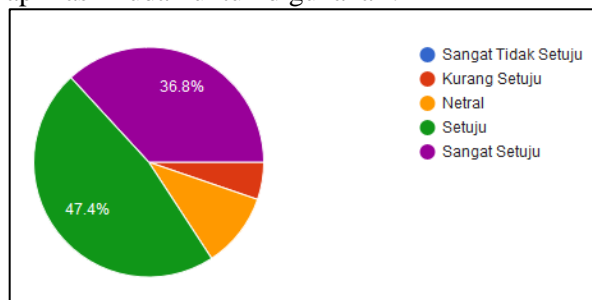
1. Apakah tampilan aplikasi nyaman untuk dilihat?



Gambar 13. Hasil Responden Kenyamanan untuk Dilihat

Kebanyakan responden menyatakan setuju bahwa tampilan dari aplikasi nyaman untuk dilihat sebesar 71.1% dapat dilihat pada gambar 13.

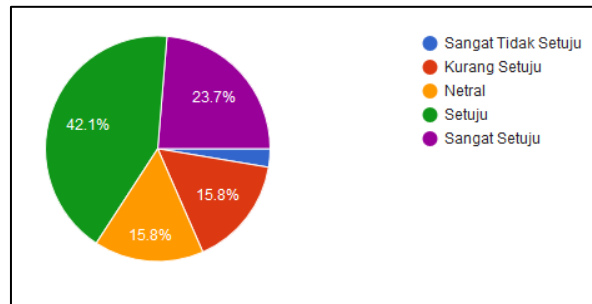
2. Apakah menu aplikasi mudah untuk digunakan?



Gambar 14. Hasil Responden Kemudahan untuk Digunakan

Gambar 14 menunjukkan responden yang setuju bahwa menu dari aplikasi mudah untuk digunakan dengan jumlah yang memilih “setuju” dan “sangat setuju” sebesar 84.2%.

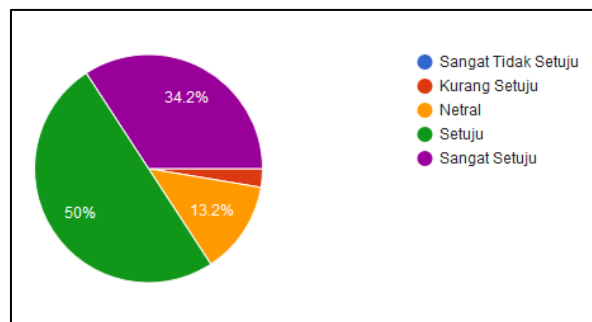
3. Apakah instruksi dalam aplikasi mudah untuk dimengerti?



Gambar 15. Hasil Responden Kemudahan untuk Dimengerti

Responden yang setuju bahwa instruksi pada aplikasi mudah untuk dimengerti sebesar 65,8%. Namun jumlah responden yang kurang setuju juga cukup banyak sebesar 34.2%. Detil hasil dapat dilihat pada gambar 15.

1. Apakah bagian tes aplikasi ini mudah untuk dioperasikan?



Gambar 16. Hasil Responden Kemudahan untuk Dioperasikan

Umpan balik responden yang setuju bahwa aplikasi ini mudah untuk dioperasikan sebesar 84.2%. dapat dilihat pada gambar 16.

4. KESIMPULAN DAN SARAN

Pemanfaatan ponsel pintar untuk melakukan uji buta warna bisa menjadi salah satu media alternatif untuk menghindari instrumen uji yang rusak atau hilang. Untuk membantu penggunaan aplikasi Uji Buta Warna yang memanfaatkan metode 100 Hue Farnsworth Munsell dapat menerapkan desain antarmuka dengan minimal rasio kontras warna sebesar 3, dimana penelitian ini memanfaatkan warna biru dengan kode warna sebagai warna foreground dan warna background putih (). Agar tampilan antarmuka bisa dihasilkan dengan baik, layar dengan full-display perlu dilakukan optimisasi. Responden juga memberikan tanggapan yang positif setelah menggunakan aplikasi pada saat pengujian dari sisi kemudahan untuk menggunakan dan mengoperasikan, nyaman untuk dilihat, dan kemudahan untuk mengikuti instruksi yang diberikan dari aplikasi. Untuk penelitian selanjutnya, metode uji buta warna perlu dikembangkan dengan memanfaatkan metode pengolahan citra dengan memanfaatkan konsep machine learning dalam menghitung tingkat keparahan dan kategori buta warna yang diderita.

DAFTAR PUSTAKA

- [1] M. A. Suryawan and M. Safei, "Implementasi Metode Ishihara pada Aplikasi Tes Buta Warna Berbasis Android," *J. Inform.*, vol. 7, no. 2, 2018.
- [2] A. Wijaya and M. Muntahanah, "APLIKASI TES BUTA WARNA DENGAN METODE FARNSWORTH MUNSELL BERBASIS ANDROID," *JSAI (Journal Sci. Appl. Informatics)*, vol. 3, no. 1, 2020, doi: 10.36085/jsai.v3i1.848.
- [3] J. S. Ng, "The FM100-Hue Test Can Detect Poor Color Vision Undetected by Color Vision Screening," *Optometry Vis. Perform.*, vol. 10, no. 1, pp. 53–56, Apr. 2022.
- [4] S. Sadya, "Pengguna Smartphone Indonesia Terbesar Keempat Dunia pada 2022," <https://dataindonesia.id/>, 2023. <https://dataindonesia.id/digital/detail/pengguna-smartphone-indonesia-terbesar-keempat-dunia-pada-2022> (accessed Jun. 02, 2023).
- [5] A. P. Putra, F. A. Fiolana, and D. A. W. Kusumastutie, "Penerapan Koreksi Warna Pada Citra Bagi Penyandang Buta Warna Parsial," *J. Tek. Elektro dan Komput. TRIAC*, vol. 8, no. 1, 2021, doi: 10.21107/triac.v8i1.10246.
- [6] S. Aytac, "Using Color Blindness Simulator During User Interface Development for Accelerator Control Room Applications; Using Color Blindness Simulator During User Interface Development for Accelerator Control Room Applications," in *Proceedings of the International Conference on Accelerator and Large Experimental Control Systems*, 2017, pp. 1958–1963. [Online]. Available: <https://accelconf.web.cern.ch/icaleps2017/papers/thsh103.pdf>
- [7] H. Hristov, T. Glushkova, S. Cheresharov, and M. Stoeva, "A Model for Designing Accessible Color and Contrast for Users with Visual Deficiency and Color Blindness," in *2022 IEEE 11th International Conference on Intelligent Systems, IS 2022*, 2022. doi: 10.1109/IS57118.2022.10019637.
- [8] M. W. Iqbal, S. K. Shahzad, N. Ahmad, A. Amelio, and D. Brodic, "Adaptive interface for color-blind people in mobile-phones," in *2018 International Conference on Advancements in Computational Sciences, ICACS 2018*, 2018. doi: 10.1109/ICACS.2018.8333488.
- [9] R. Estrada and E. A. Sarwoko, "Aplikasi Tes Buta Warna Dengan Metode Ishihara dan Farnsworth Munsell D-15 (Studi Kasus : Puskesmas Rowosari)," *J. Masy. Inform.*, vol. 10, no. 1, 2019, doi: 10.14710/jmasif.10.1.31486.
- [10] S. Saini, E. Febriani Dunga, and I. Sulistiani, "Evaluasi Pemeriksaan Tes Buta Warna Menggunakan Metode Ishihara Berbasis Google Form Menggunakan Buku Ishihara," *Indones. J. Pharm. Educ.*, vol. 1, no. 3, 2022, doi: 10.37311/ijpe.v2i1.15855.
- [11] A. French *et al.*, "The Evolution of Colour Vision Testing," *Aust. Orthopt. J.*, vol. 39, no. 2, 2008.
- [12] S. Agusta, T. Mulia, and M. Sidik, "Instrumen Pengujian Buta Warna Otomatis," *J. Ilm. Elit. Elektro*, vol. 3, no. 1, 2012.
- [13] D. Farnsworth, "The Farnsworth-Munsell 100-Hue and Dichotomous Tests for Color Vision*," *J. Opt. Soc. Am.*, vol. 33, no. 10, 1943, doi: 10.1364/josa.33.000568.
- [14] M. M. Eler, J. M. Rojas, Y. Ge, and G. Fraser, "Automated Accessibility Testing of Mobile Apps," in *Proceedings - 2018 IEEE 11th International Conference on Software Testing, Verification and Validation, ICST 2018*, 2018. doi: 10.1109/ICST.2018.00021.
- [15] P. and P. Institute for Disability Research, "Contrast Checker," *WebAIM*, Aug. 04, 2023. <https://webaim.org/resources/contrastchecker/> (accessed Aug. 04, 2023).
- [16] World Wide Web Consortium (W3C), "Web Content Accessibility Guidelines (WCAG) 2.1," Aug. 04, 2023. <https://www.w3.org/TR/WCAG21/> (accessed Aug. 04, 2023).
- [17] M. Ballantyne, A. Jha, A. Jacobsen, J. Scott Hawker, and Y. N. El-Glaly, "Study of accessibility guidelines of mobile applications," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3282894.3282921.
- [18] P. Acosta-Vargas, B. Salvador-Acosta, L. Salvador-Ullauri, W. Villegas-Ch, and M. Gonzalez, "Accessibility in native mobile applications for users with disabilities: A scoping review," *Applied Sciences (Switzerland)*, vol. 11, no. 12. 2021. doi:

- 10.3390/app11125707.
- [19] R. Bingham, S. W. Dietrich, and D. Goelman, "Strategies to improve accessibility for learners with color vision deficiency," *ACM Inroads*, vol. 10, no. 1, 2019, doi: 10.1145/3306140.