Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODE DAN PERANCANGAN APLIKASI

3.1 Metode Penelitian

Penelitian ini dilakukan menggunakan metode-metode berikut.

1. Studi Pustaka

Studi pustaka merupakan suatu tindakan yang dilakukan guna mendalami konsep dari hal-hal yang berhubungan dengan penelitian yang sedang dilakukan seperti CVD, *Ishihara Plate*, *Flood Fill*, dan *phythagoras*. Hal ini dilakukan dengan melakukan observasi melalui buku, jurnal, artikel, manual media pengujian CVD maupun referensi ilmiah lainnya.

2. Perancangan dan Pembangunan Aplikasi

Proses parancangan aplikasi akan dilakukan guna menganalisa prosedur dan proses apa saja yang akan dilakukan oleh aplikasi. Selain itu proses perancangan antar muka pada aplikasi juga dilakukan pada tahapan ini. Rancangan aplikasi beserta algoritma *Flood Fill* akan didokumentasikan ke dalam bentuk diagram alir, diagram alir akan merepresentasikan alur kerja aplikasi dan algoritma *Flood Fill*. Kemudian pada tahapan ini akan dipilih Bahasa pemrograman yang dapat menyediakan fungsionalitas yang dibutuhkan, sehingga aplikasi dapat segera dibangun.

3. *Testing* dan *Debugging* Aplikasi

Selama proses *development* aplikasi dilakukan, akan diadakan serangkaian pengujian terhadap fungsi-fungsi aplikasi. Kecocokan aplikasi dalam menghasilkan gambar merupakan salah satu fungsi yang diuji pada proses pengujian .

4. Pengumpulan Sampel Data

Sampel data didapatkan dengan melibatkan responden yang diambil secara acak untuk menguji kemampuannya dalam membedakan sebuah warna. Proses pengumpulan sampel data dilakukan dengan cara mengisi kuesioner setelah menggunakan aplikasi. Hal ini dilakukan untuk menguji tingkat akurasi dari aplikasi yang sudah dibuat. Untuk setiap penggunaan aplikasi yang dilakukan, *feedback user* akan disimpan secara manual melalui layanan *online form* yang akan diisi oleh responden.

5. Analisis Sampel Data

Sampel data yang telah dikumpulkan akan dianalisis untuk menentukan akurasi dari aplikasi. Pada tahapan ini akan dilakukan uji skenario, untuk memastikan bahwa algoritma *Flood Fill* berhasil menghasilkan gambar yang sesuai dengan pola yang diberikan.

6. Penulisan Laporan

Sebuah laporan mengenai hasil observasi yang akan ditulis secara bertahap, dimulai dari studi pustaka hingga penarikan kesimpulan berdasarkan hasil analisis data yang sudah didapatkan.

3.2 Rancangan Aplikasi

Aplikasi dimulai dengan menampilkan empat buah pilihan dalam melakukan pengujian buta warna yang dikategorikan sebagai berikut. Setiap pilihan akan mengarahkan pengguna pada halaman tes dengan set pertanyaan untuk pilihan tersebut. Pemilihan jumlah pertanyaan untuk setiap tes berdasar manual *Ishihara Plate*. Menurut manual, terdapat 24 buah gambar yang diujikan pada tes *Ishihara*

Plate standar. Sedangkan untuk tes pendek disarankan untuk memberikan minimal 6 buah gambar untuk proses pengujian *Ishihara Plate*.

1. *Standard Color Blind Test*

Terdapat 24 gambar yang akan dihasilkan secara acak pada tes ini. Tes ini disimulasikan sesuai dengan tes *Ishihara Plate.* Pada tes ini, kemampuan individu akan diuji berdasarkan standar yang sudah ditetapkan. Tes ini tidak dapat melalukan pengujian terhadap penderita *tritanopia*.

2. *Protanopia Color Blind Test*

Pada tes ini terdapat 10 gambar yang berfokus pada pengujian *protanopia* yang akan dihasilkan secara acak. Pengujian ini menentukan kemampuan seorang individu dalam mengenali warna merah.

3. *Deuteranopia Color Blind Test*

Di dalam tes ini terdapat 10 gambar yang akan dihasilkan secara acak. Tes ini menentukan kemampuan seorang individu dalam mengenali warna hijau.

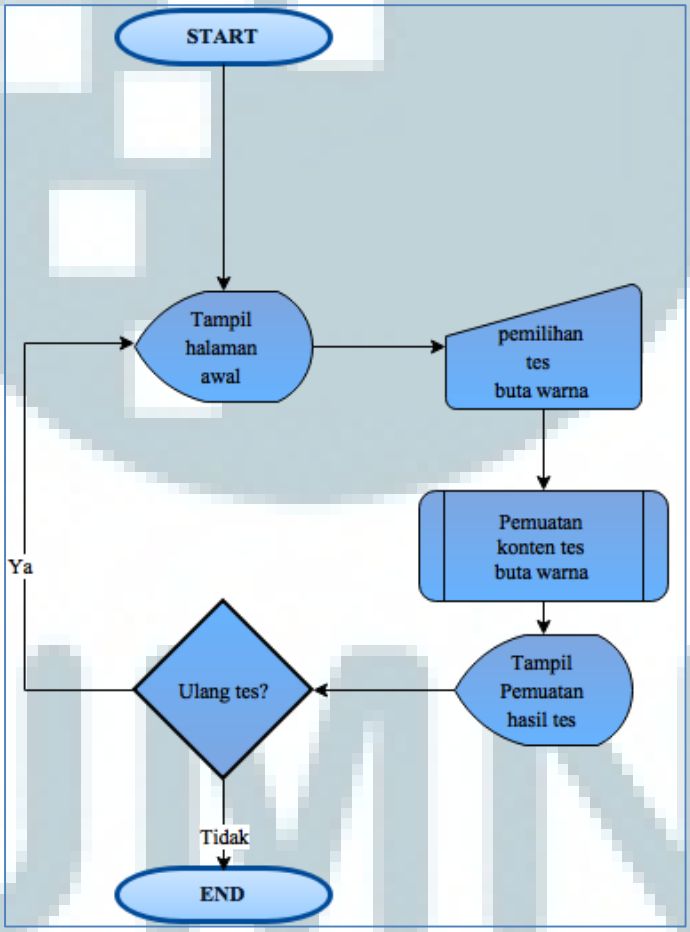
4. *Tritanopia Color Blind Test*

Akan dihasilkan 10 gambar secara acak pada tes ini. Tes ini menentukan kemampuan seorang individu dalam mengenali warna yang berada di antara spektrum warna biru dan warna kuning.

Setelah pengguna melakukan pemilihan tes, aplikasi akan menampilkan serangkaian gambar yang dihasilkan secara acak. Selanjutnya gambar pada halaman pengujian tes, akan dihasilkan dengan cara menerapkan algoritma *Flood Fill* untuk menghasilkan warna yang baru, sedangkan untuk polanya sendiri dihasilkan secara *random*. Setiap Gambar yang dihasilkan oleh aplikasi, akan

ditampilkan selama lima detik. Hal ini dilakukan agar individu yang diuji tidak berusaha memeriksa pola yang memang tidak mudah dilihat. Setiap pertanyaan yang ditampilkan, akan disertai sebuah *form* yang digunakan untuk menuliskan pola yang terlihat. Setelah semua pertanyaan dijawab, akan ditampilkan prosentase akurasi yang didapatkan berdasarkan tes yang dilakukan.

Secara garis besar aplikasi dapat dibagi menjadi dua buah proses utama dengan satu prosedur utama. Berikut diagram alir untuk aplikasi ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram Alir Aplikasi Secara Garis Besar

3.2.1 Proses Pemuatan Halaman Awal

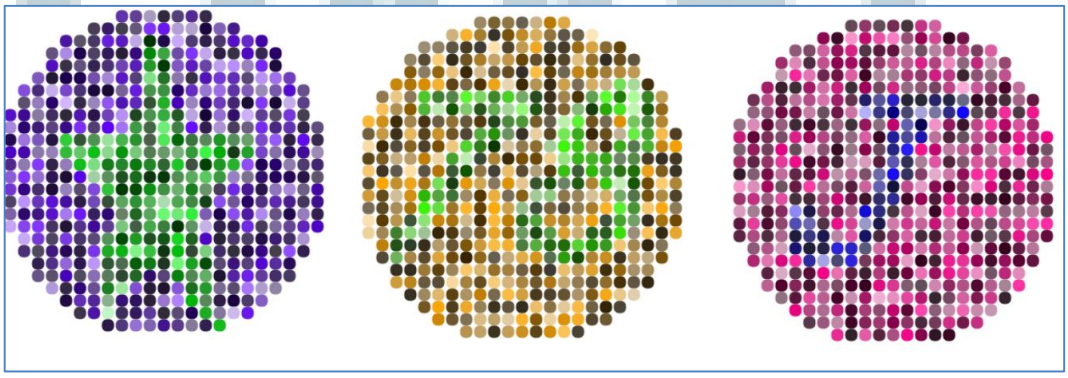
Proses ini dipanggil saat aplikasi dimulai dan saat aplikasi diminta mengulangi proses pengujiannya kembali. Proses akan menampilkan pilihan yang mendeskripsikan tes yang akan diambil. Pada proses ini, semua bahan pengujian sudah disimpan secara lokal pada aplikasi tersebut.

3.2.2 Prosedur Pemuatan Konten Tes Buta Wana

Prosedur ini memiliki peranan untuk menghasilkan konten dari tes buta warna yang dipilih. Konten yang ada dibuat secara *real time*. Hal ini dilakukan untuk memastikan bahwa gambar yang dihasilkan selalu baru. Selain itu, pola yang dihasilkan pada gambar dipilih secara acak dengan cakupan angka, huruf, dan gambar *vector*.

Prosedur ini akan menjalankan prosedur inisialisasi elemen *dots* dan memanggil prosedur inisialisasi *canvas*. Diagram alir prosedur inisialisasi elemen *dots* dapat dilihat pada gambar 3.6 sedangkan diagram alir prosedur inisialisasi *canvas* dapat dilihat pada gambar 3.7. Prosedur ini dipanggil untuk setiap soal yang akan dibuat. Dalam proses pembuatan gambar, prosedur akan menciptakan sebuah gambar dasar menggunakan prosedur *generate* gambar dasar. Diagram alir prosedur *generate* gambar dasar dapat dilihat pada gambar 3.8. Gambar ini tidak akan terlihat pada hasil akhir. Gambar ini hanya digunakan sebagai sebuah dasar untuk menentukan warna yang akan muncul pada lingkaran kecil jika ditemukan sebuah *pixel* dengan warna tertentu. *Pixel* direpresentasikan ke dalam bentuk lingkaran kecil. Terdapat tiga buah macam warna yang dapat ditemukan pada gambar dasar. Warna hitam digunakan sebagai warna untuk bilangan pertama beserta huruf dan gambar, sedangkan warna hijau digunakan sebagai representasi bilangan satuan, dan warna

putih sebagai penentu warna *background* dari gambar yang akan dihasilkan. Penentuan warna pada lingkaran didapatkan melalui prosedur *Flood Fill*. Penentuan warna dilakukan dengan cara mengambil *pixel* gambar dasar yang berada di koordinat titik tengah lingkaran kecil. Setiap titik tengah pada elemen lingkaran kecil akan mewakili *pixel* warna sebesar radius lingkaran kecil. Diagram alir untuk prosedur *Flood Fill* dapat dilihat pada gambar 3.5. Setelah warna untuk setiap lingkaran ditentukan, gambar dasar akan dihapus dan digantikan dengan gambar baru yang dihasilkan berdasarkan prosedur *Flood Fill* yang telah dijalankan melalui prosedur pengisian *canvas* dengan data *Flood Fill*. Diagram alir prosedur pengisian canvas dengan data *Flood Fill* dapat dilihat pada gambar 3.9. Gambar ini dibentuk oleh sekumpulan lingkaran kecil yang akan membentuk gambar berbentuk lingkaran besar. Contoh gambar lingkaran yang dihasilkan oleh *Flood Fill* dapat dilihat pada gambar 3.2, contoh gambar dasar dapat dilihat pada gambar 3.3, serta diagram alir prosedur pemuatan konten pada gambar 3.4. Prosedur penghitungan *score* akan dipanggil setelah pengisian data gambar dilakukan. Berdasarkan masukan pengguna, maka prosedur penghitungan *score* akan melakukan penambahan *score* untuk pengguna tersebut. Diagram alir prosedur penghitungan score dapat dilihat pada gambar 3.10.

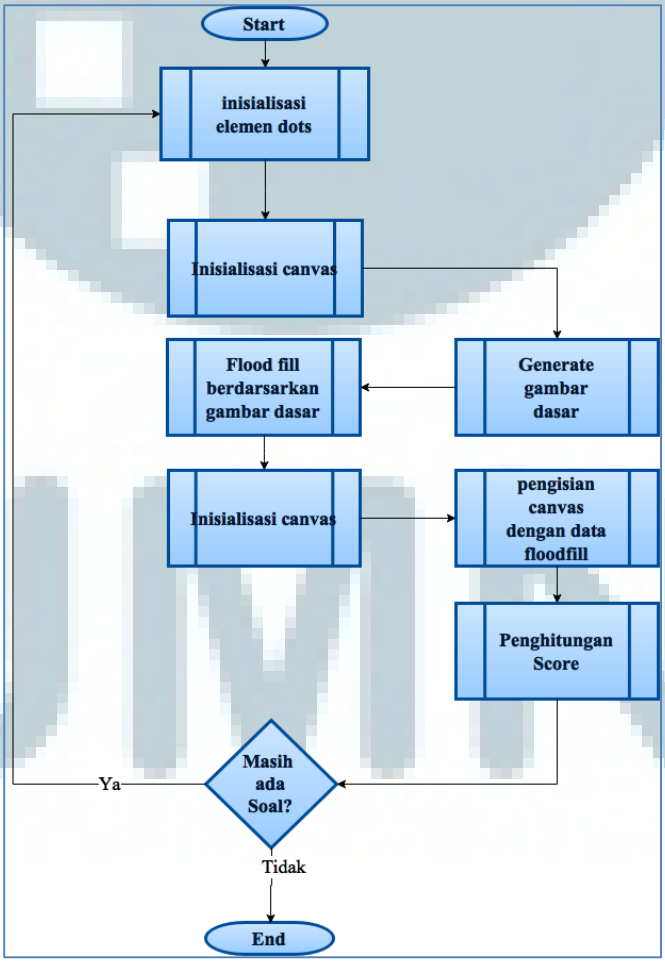


Gambar 3.2 Ilustrasi Gambar *Ishihara Plate* dengan file gambar, angka, dan huruf

Gambar 3.3 Gambar Dasar

3.2.3 Prosedur Pemuatan Konten

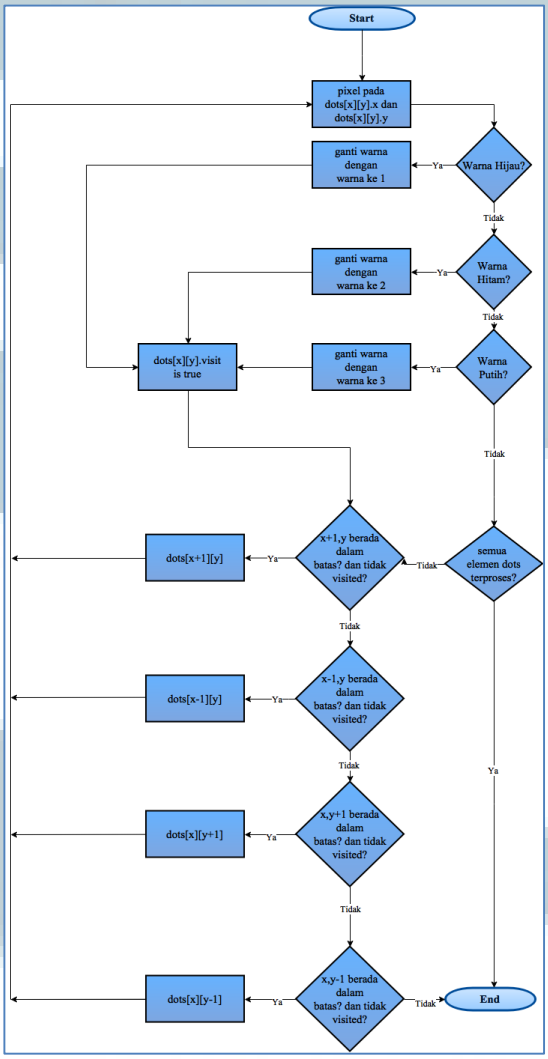
Prosedur ini bertujuan untuk memanggil subrutin-subrutin yang berfungsi untuk memuat konten tes buta warna. Subrutin-subrutin ini dibahas pada subbab selanjutnya. Diagram alir untuk prosedur ini dapat dilihat pada gambar 3.4.



Gambar 3.4 Diagram Alir Prosedur Pemuatan Konten

3.2.4 Prosedur Flood Fill Berdasarkan Gambar Dasar

Dalam prosedur ini, elemen *array dots* akan diisi menggunakan algoritma *Flood Fill* dengan *pixel* gambar dasar sebagai penentu nilai yang akan diisi. Prosedur akan mengambil koordinat titik tengah pada elemen *array* yang akan diproses. Setelah didapatkan koordinat titik tengah, prosedur ini akan mengganti warna pada elemen *array dots* dengan acuan warna *pixel* gambar dasar pada koordinat titik tengah elemen yang hendak diproses.

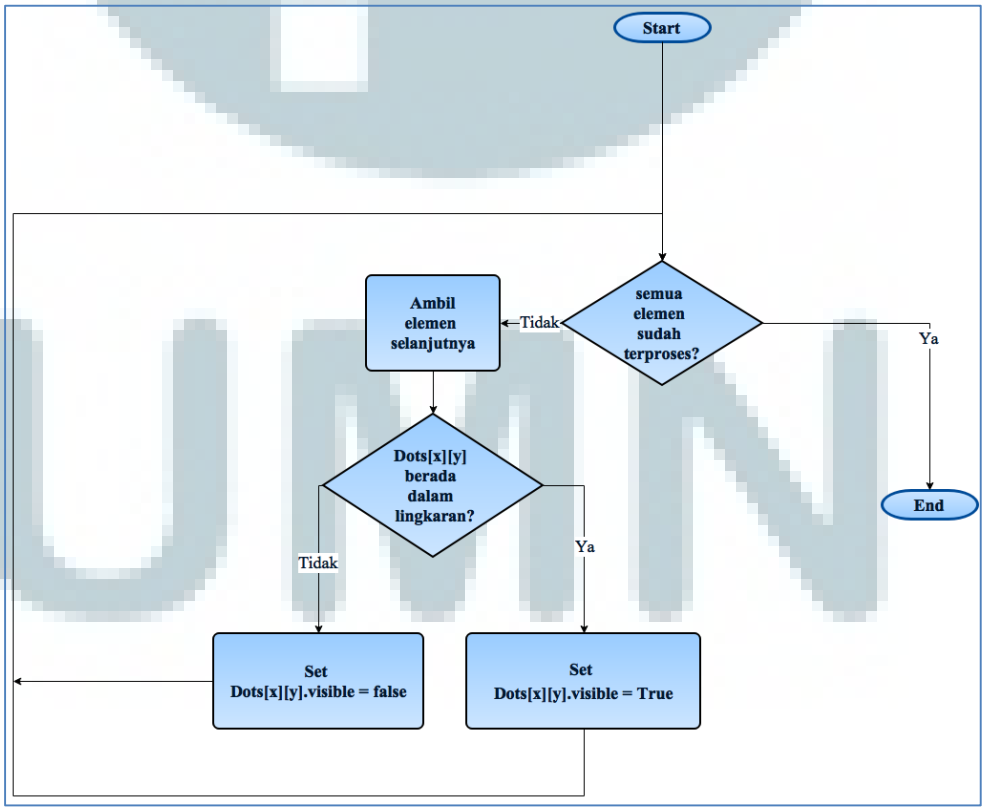


Gambar 3.5 Diagram Alir Prosedur *Flood Fill* Berdasarkan Gambar Dasar

Setelah warna elemen tersebut terganti, maka prosedur ini akan mencoba memproses elemen yang bersebelahan dengan elemen yang sedang diproses. Diagram prosedur ini dapat dilihat pada gambar 3.5.

3.2.5 Prosedur Inisialisasi Elemen *Dots*

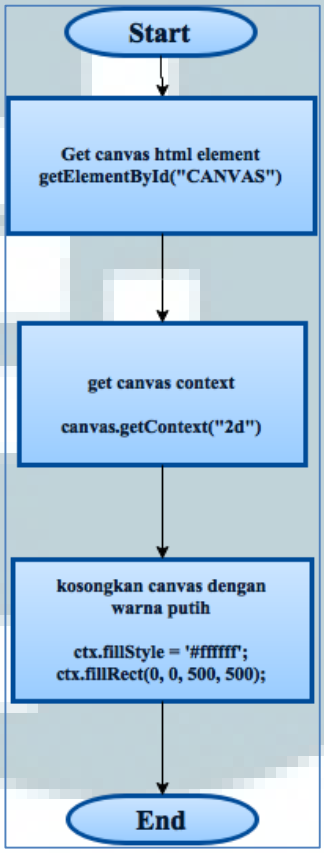
Prosedur ini akan melakukan proses inisialisasi pada *array dots* untuk selanjutnya diproses oleh prosedur selanjutnya. Proses ini akan mengambil setiap elemen *array dots* dan menentukan apakah elemen tersebut memiliki titik tengah yang berada pada daerah lingkaran besar. Apabila elemen tersebut berada di dalam daerah lingkaran, maka atribut *visible* pada elemen tersebut akan memiliki *value true* sedangkan jika elemen tersebut berada di luar daerah lingkaran, maka atribut *visible* pada elemen tersebut akan memiliki *value false*. Diagram alir untuk prosedur ini dapat dilihat pada gambar 3.6.



Gambar 3.6 Diagram Alir Prosedur Inisialisasi Elemen *Dots*

3.2.6 Prosedur Inisialisasi Canvas

Pada prosedur ini gambar yang ada pada *canvas* akan dibersihkan. *Canvas* akan dibersihkan dengan cara memperbaharui elemen *canvas*, sehingga didapatkan elemen *canvas* kosong. diagram alir dapat dilihat pada gambar 3.7.

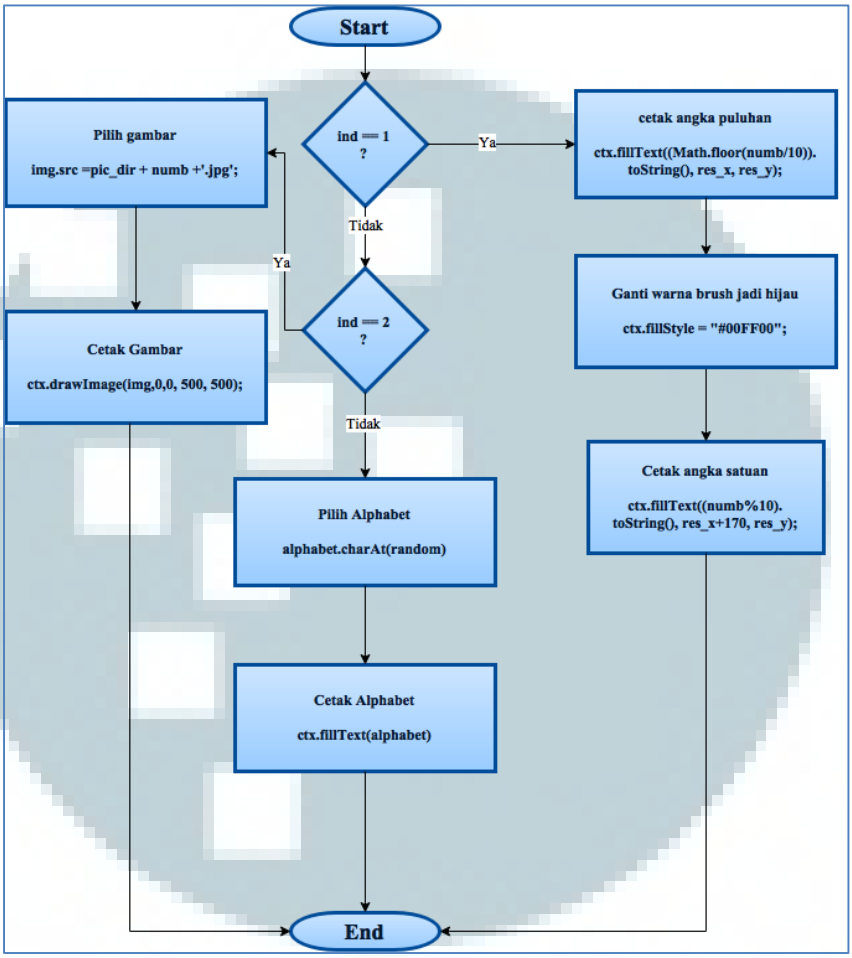


Gambar 3.7 Diagram Alir Prosedur Inisialisasi *Canvas*

3.2.7 Prosedur Generate Gambar Dasar

Prosedur ini melakukan pemilihan pola yang akan ditampilkan pada gambar dasar. Gambar dasar ini nantinya akan dijadikan sebagai pola untuk gambar pada pengujian tes buta warna. Pola yang digunakan berupa huruf, angka, dan objek. Prosedur akan menentukan jenis pola manakah yang akan digunakan sebagai dasar dari gambar dasar menggunakan *variable random*. Berdasarkan pola yang telah

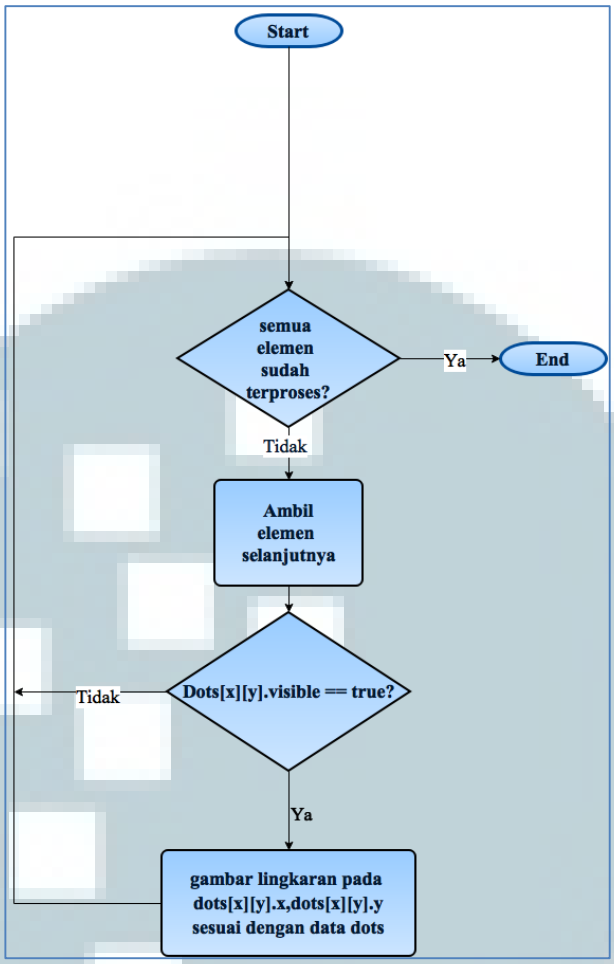
ditentukan maka gambar tersebut akan digunakan sebagai gambar dasar untuk diproses oleh algoritma *Flood Fill*. Diagram alir prosedur dapat dilihat pada 3.8.



Gambar 3.8 Diagram Alir Prosedur *Generate* Gambar Dasar

3.2.8 Prosedur Pengisian Canvas dengan Data Flood Fill

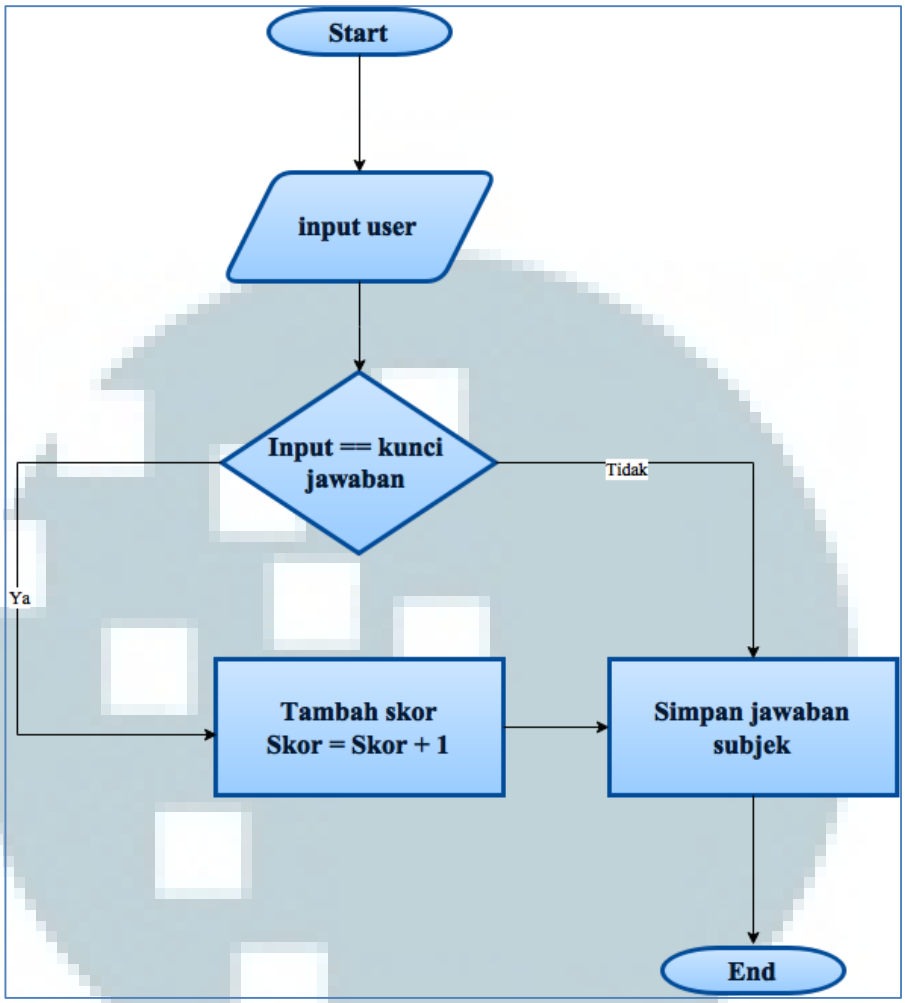
Prosedur pengisian *canvas* dengan data *Flood Fill* memiliki peran untuk mengisi *canvas* dengan data elemen *array dots* yang telah disimpan oleh prosedur *Flood Fill* berdasarkan gambar dasar. Prosedur akan mengecek atribut *visible* pada elemen tersebut dan mencetak lingkaran kecil apabila atribut itu bernilai *true*. Diagram alir prosedur ini dapat dilihat pada gambar 3.9



Gambar 3.9 Diagram Alir Prosedur Pengisian *Canvas* dengan Data *Flood Fill*

3.2.9 Prosedur Penghitungan Score

Prosedur ini memiliki peran untuk menghitung *score* yang akan didapatkan oleh *user*. Penambahan *score* akan dilakukan apabila *input* yang dimasukkan sesuai dengan kunci jawaban soal. Jawaban pada aplikasi ini bersifat *case sensitive*, hal ini dilakukan karena beberapa *alphabet* memiliki bentuk kapital yang berbeda dengan bentuk non-kapitalnya. Diagram prosedur ini dapat dilihat pada gambar 3.10.



Gambar 3.10 Diagram Alir Prosedur Penghitungan *Score*

3.2.10 Proses Pemuatan Hasil Tes

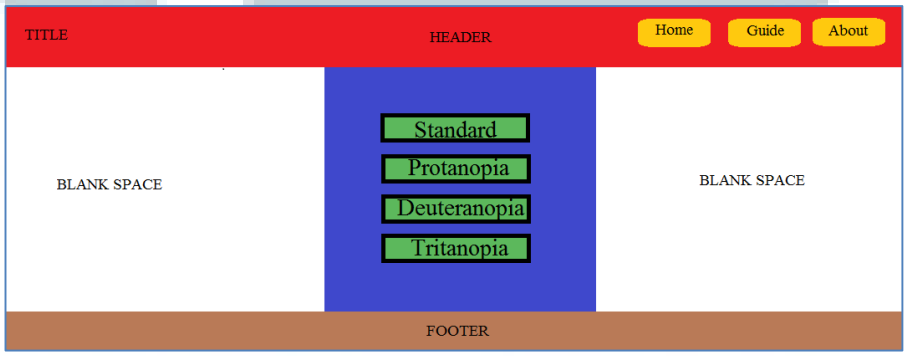
Proses ini memiliki peran menampilkan hasil dari pengujian yang sudah disimpan pada prosedur sebelumnya. Pada hasil yang sudah tersedia akan ditampilkan jawaban benar dari tiap pertanyaan yang telah disajikan disamping itu, akan ditampilkan jawaban yang telah dimasukkan oleh individu yang diuji, sehingga dapat terlihat perbedaan antara jawaban dengan kunci jawaban. Proses ini juga menampilkan prosentase ketepatan dalam menjawab pertanyaan yang diujikan.

3.3 Rancangan Tampilan Antarmuka Aplikasi

Aplikasi ini memiliki tiga buah tampilan utama, yaitu tampilan halaman muka, tampilan halaman tes, dan tampilan halaman hasil. Berikut penjelasan mengenai tiga tampilan utama aplikasi. Rancangan antarmuka aplikasi dibuat menjadi responsif, namun aplikasi ini tidak ditujukan perangkat mobile. Hal ini disebabkan ukuran gambar yang kecil menjadi kendala dalam melakukan pengujian.

1. Tampilan Halaman Muka

Tampilan halaman muka merupakan halaman awal dari aplikasi. Halaman ini menampilkan navigasi yang akan mengarahkan *user* menuju halaman tes dengan kombinasi warna yang sesuai dengan pilihan yang telah dipilih. Rancangan tampilan halaman muka dapat dilihat pada gambar 3.11.

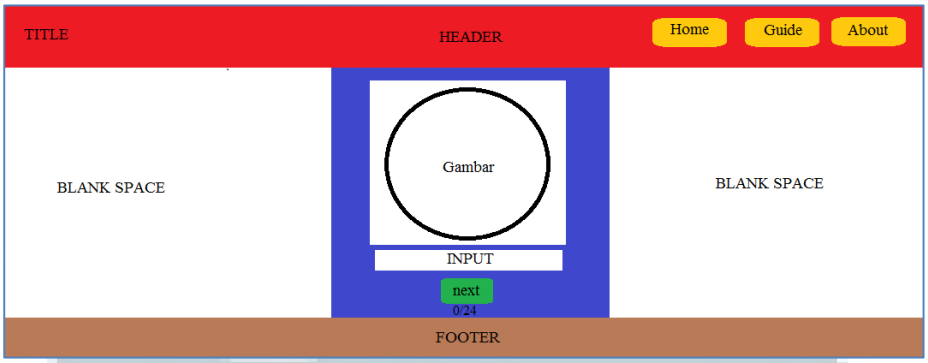


Gambar 3.11 Rancangan Tampilan Halaman Muka

2. Tampilan Halaman Tes

Tampilan halaman tes merupakan halaman yang akan menampilkan soal yang akan diujikan. Halaman ini mempunyai elemen *canvas* yang digunakan untuk menampilkan gambar yang akan diujikan. Halaman tes dilengkapi dengan *form* untuk menjawab pertanyaan dan sebuah tombol. Tombol *submit* memiliki fungsi untuk menjawab pertanyaan dan mengarahkan pengguna ke soal selanjutnya. Pada

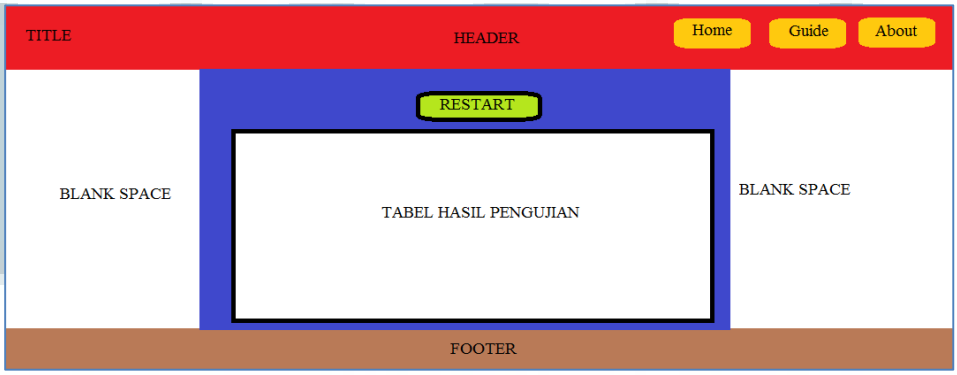
halaman ini terdapat text untuk mengetahui progress dari pengujian. Rancangan tampilan halaman tes dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan Tampilan Halaman Uji

3. Tampilan Halaman Hasil

Halaman hasil merupakan halaman yang menampilkan hasil dari pengujian yang telah dilakukan. Halaman hasil akan menampilkan tabel yang merepresentasikan hasil tes yang sudah dilakukan. Selain itu, Prosentase jawaban benar yang didapatkan melalui pengujian tersebut juga ditampilkan pada halaman ini. Selain itu halaman hasil dilengkapi tombol yang akan mengarahkan *user* menuju halaman awal. Rancangan tampilan halaman hasil dapat dilihat pada gambar 3.13.



Gambar 3.13 Rancangan Tampilan Halaman Hasil