



SISTEM PAKAR DIAGNOSA BUTA WARNA BERBASIS ANDROID

Skripsi

diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer

Oleh

Puspita Prabawati NIM. 5302411215

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG

2015

PERSETUJUAN PEMBIMBING

Nama : Puspita Prabawati

NIM : 5302411215

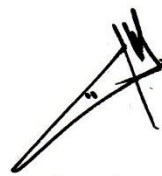
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer

Judul Skripsi : SISTEM PAKAR DIAGNOSA BUTA WARNA BERBASIS
ANDROID

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer FT UNNES.

Semarang, November 2015

Pembimbing,



Drs. Agus Suryanto, M.T.

NIP. 196708181992031004

HALAMAN PENGESAHAN

Skripsi dengan judul Sistem Pakar Diagnosa Buta Warna Berbasis Android, telah dipertahankan didepan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES pada 5 November 2015.

Oleh

Nama: Puspita Prabawati

NIM: 5302411215

Program Studi: S-1 Pendidikan Teknik Informatika dan Komputer

Panitia:

Ketua Panitia



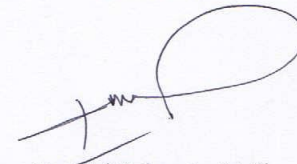
Drs. Suryono, M.T.
NIP. 195503161985031001

Penguji I



Drs. Agus Suryanto, M.Pd.
NIP. 195706051986011001

Penguji II



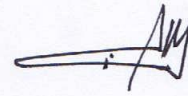
Drs. Sri Sukanto, M.Si.
NIP. 19650508199103003

Sekretaris



Feddy Setio Pribadi, S. Pd, M.T.
NIP. 197808222003121002

Penguji III/Pembimbing



Drs. Agus Suryanto, M.T.
NIP. 196708181992031004

Mengetahui:

Dekan Fakultas Teknik UNNES



Dr. Nur Qudus, M.T.
NIP. 196911301994031001

PERNYATAAN

Saya menyatakan bahwa yang tertulis di dalam skripsi ini benar-benar hasil karya sendiri berdasarkan arahan dosen pembimbing, bukan jiplakan dari karya tulis orang lain, baik sebagian atau seluruhnya. Pendapat atau temuan yang terdapat dalam tulisan ini dirujuk berdasarkan kode etik ilmiah.



Semarang, November 2015

Puspita Prabawati
NIM 5302411215

MOTTO DAN PERSEMBAHAN

Motto:

1. Karena sesungguhnya, sesudah kesulitan itu pasti ada kemudahan.
(QS.94:5)
2. Tiada daya dan upaya melainkan pertolongan Allah.

Persembahan:

1. Alm. Bp. Tri Djoko dan Ibu Sri Purwanti,
kedua orang tuaku tercinta yang
senantiasa mencurahkan kasih sayang,
bimbingan dan doa untukku
2. Kakakku tersayang Prima Anugraheni
yang selalu memberikan semangat
untukku.

PRAKATA

Puji syukur kehadiran Allah SWT, atas rahmat dan hidayahnya penyusunan skripsi ini dapat terselesaikan dengan tuntas dan baik. Skripsi ini disusun dalam rangka penyelesaian studi strata satu untuk mencapai gelar sarjana pendidikan. Penulisan skripsi ini selesai berkat bantuan beberapa pihak. Untuk itu ucapan terima kasih tersampaikan kepada:

1. Dr. Nur Qudus, M.T., Dekan Fakultas Teknik Universitas Negeri Semarang yang telah memberikan ijin penelitian dalam memperlancar penyelesaian skripsi ini.
2. Drs. Suryono, M.T., Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang yang telah memberikan kemudahan administrasi kepada penulis dalam penyusunan skripsi ini.
3. Feddy Setio Pribadi, S.T, M.T., Ketua Prodi PTIK Universitas Negeri Semarang yang telah membantu kelancaran dalam penyusunan skripsi ini.
4. Drs. Agus Suryanto, M. T., Dosen Pembimbing yang telah memberikan waktu, bimbingan, dan saran dalam menyelesaikan skripsi ini.
5. dr.Anies Setiowati, M.Gizi., Dokter/Pakar yang telah menguji Aplikasi Sistem Pakar Diagnosa Buta Warna ini.
6. Seluruh keluarga besar yang memberikan dukungan untukku.
7. Teman-teman PTIK Unnes 2011 untuk kebersamaan selama ini.
8. Teman-teman Romli (Rombel Lima) yang selalu mendukung dan memberi semangat.

9. Sabahat RMLH terima kasih atas kebersamaan dan persahabatan. Perjalanan bersama kalian menorehkan banyak pengalaman dan pelajaran yang begitu berarti.

10. Semua pihak yang telah memberikan dukungan dan bantuannya.

Peneliti sadar bahwa skripsi ini masih jauh dari kesempurnaan, penulis mengharapkan saran dan kritik dari semua pihak. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak yang memerlukan.

Semarang, Oktober 2015

Peneliti

ABSTRAK

Puspita Prabawati. 2015. *Sistem Pakar Diagnosa Buta Warna Berbasis Android*. Skripsi, Pendidikan Teknik Informatika dan Komputer, Fakultas Teknik, Universitas Negeri Semarang. Drs. Agus Suryanto, M.T.

Kemajuan pada ilmu kedokteran dan ilmu pengetahuan pada umumnya memunculkan peralatan medis yang semakin canggih dalam memerangi penyakit atau melakukan deteksi lebih dini pada kondisi-kondisi tertentu. Salah satunya gangguan yang terjadi pada mata yaitu buta warna. Permasalahan dalam penelitian ini yaitu bagaimana pengembangan Aplikasi Diagnosa Buta Warna Berbasis Android, apakah aplikasi tersebut valid/dapat digunakan sebagai alat bantu tes buta warna sehingga bermanfaat dalam hal mengembangkan teknologi dalam bidang kedokteran.

Metode perancangan dan pengembangan aplikasi ini menggunakan model *Waterfall*. Objek yang diteliti yaitu aplikasi ini sendiri dengan variabel penelitian *sensitivitas* dan *spesifisitas* sistem. Metode pengambilan data dengan menguji aplikasi ini kepada pakar kemudian diuji cobakan kepada masyarakat umum. Pengambilan data dilakukan kepada penderita buta warna total, buta warna sebagian dan mata normal. Hasil akurasi aplikasi ini ditentukan oleh nilai *sensitivitas* dan *spesifisitas* dengan rumus yang telah ada.

Hasil penelitian menunjukkan bahwa sistem pakar diagnosa buta warna mampu mendiagnosa buta warna secara valid kepada masyarakat umum dengan *sensitivitas* yang dibangun berada pada kategori valid dengan nilai 100%, tingkat *spesifisitas* pada kategori valid sebesar 80%, nilai akurasi dari keseluruhan aplikasi sebesar 90%. Kesimpulan dari penelitian ini adalah aplikasi yang dibuat merupakan aplikasi dengan tingkat *sensitivitas* dan *spesifisitas* kesehatan dapat diterapkan sebagai pengganti tes buta warna secara manual. Diajukan saran penelitian lanjutan dengan mengubah database yang online menjadi offline namun dengan soal tes yang tetap acak, perlunya uji media dan dikembangkan lagi menggunakan metode lain.

Kata kunci: *sistem pakar diagnosa buta warna, android, metode waterfall*.

DAFTAR ISI

SISTEM PAKAR DIAGNOSA BUTA WARNA BERBASIS ANDROID	i
PERSETUJUAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN.....	iv
MOTTO DAN PERSEMBAHAN	v
PRAKATA.....	vi
ABSTRAK	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
LAMPIRAN.....	xiii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Identifikasi Masalah	4
1.3 Batasan Masalah.....	4
1.4 Rumusan Masalah	5
1.5 Tujuan Penelitian.....	5
1.6 Manfaat Penelitian.....	5
BAB II.....	6
KAJIAN PUSTAKA.....	6
2.1 Kajian Teori.....	6
2.1.1 Definisi Sistem Pakar.....	6
2.1.2 Komponen Sistem Pakar.....	9
2.1.3 Teknik Representasi Pengetahuan.....	11
2.1.4 Metode <i>Forward Chaining</i>	12
2.1.5 Diagnosa.....	15
2.1.6 Versi Android.....	15
2.1.7 Buta Warna.....	21
2.1.8 Alat Tes Kebutaan Warna.....	22

2.2	Kerangka Berpikir	27
2.3	Hipotesis.....	30
BAB III		31
METODE PENELITIAN		31
3.1	Metode Pengembangan Perangkat Lunak	31
3.2	Alat dan Bahan Pembuatan Aplikasi.....	31
3.3	Tahapan Perancangan Aplikasi	32
3.4	Tahapan Pengembangan Perangkat Lunak/Aplikasi	32
3.5	Tahapan Penelitian	42
BAB IV		44
HASIL PENELITIAN DAN PEMBAHASAN		44
4.1.	HASIL PENELITIAN	44
4.2.	PEMBAHASAN	62
BAB V.....		67
PENUTUP.....		67
5.1	Kesimpulan.....	67
5.2	Saran.....	67
DAFTAR PUSTAKA		69
Lampiran		71

DAFTAR GAMBAR

Gambar 1 Komponen Sistem Pakar	9
Gambar 2 Contoh Tes Buta Warna dengan Metode Ishihara	22
Gambar 3 Koin-koin Warna Farnsworth Munsell	27
Gambar 4 Model Waterfall	32
Gambar 5 Use Case Diagram	34
Gambar 6 Activity Diagram	35
Gambar 7 Tampilan Halaman Intro	35
Gambar 8 Tampilan Halaman Utama	36
Gambar 9 Tampilan Menu Daftar	36
Gambar 10 Tampilan Menu Login	37
Gambar 11 Tampilan Mulai Tes	37
Gambar 12 Tampilan Hasil Tes	38
Gambar 13 Tampilan Menu Seputar Buta Warna	38
Gambar 14 Tampilan Menu About	39
Gambar 15 Tahapan Penelitian	43
Gambar 16 Halaman Intro	45
Gambar 17 Halaman Home	46
Gambar 18 Halaman Menu Daftar	47
Gambar 19 Halaman Menu Login	48
Gambar 20 Halaman Mulai Tes Ishihara	49
Gambar 21 Halaman Hasil Tes Ishihara	50
Gambar 22 Halaman Utama Tes Farnsworth Munsell	51
Gambar 23 Halaman Mulai Tes Farnsworth Munsell	51
Gambar 24 Halaman Hasil Tes Farnsworth Munsell	52
Gambar 25 Halaman Menu Seputar Buta Warna	53
Gambar 26 Halaman Hasil Tes Ishihara	54

DAFTAR TABEL

Tabel 1 Versi Android.....	15
Tabel 2 Pengambilan kesimpulan Tes Buta Warna	25
Tabel 3 Tabel Nomor Plate Ishihara	55
Tabel 4 Tabel Urutan Warna Farnsworth Munsell	56
Tabel 5 Implementasi Antarmuka.....	58
Tabel 6 Pengujian Black Box.....	59
Tabel 7 Daftar Validator	60
Tabel 8 Data Hasil Uji Coba Aplikasi.....	61
Tabel 9 Nilai Sensitivitas dan Spesifisitas Aplikasi	63

LAMPIRAN

Lampiran 1 Surat Keterangan telah Melakukan Penelitian.....	71
Lampiran 2 Data Pasien Uji Coba Aplikasi	72
Lampiran 3 Dokumentasi	72
Lampiran 4 Source Code Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android.....	72

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan jaman yang semakin maju seperti sekarang ini membuat kebutuhan manusia semakin meningkat, terlebih lagi didorong dengan adanya kemajuan ilmu pengetahuan dan teknologi yang sangat cepat. Sebagai contoh, dengan adanya komputer segala kegiatan dapat dilakukan dengan cepat dan resiko kesalahan dapat dikurangi. Di dalam perkembangan komputer, para ahli komputer mencoba untuk menciptakan suatu sistem yang diharapkan dapat memiliki kemampuan memecahkan suatu permasalahan seperti seorang ahli. Konsep tersebut yang mendorong lahirnya sistem pakar. Sistem Pakar tidak untuk menggantikan kedudukan seorang pakar tetapi untuk memasyarakatkan atau meminjam pengetahuan dan pengalaman seorang pakar tersebut.

Kemajuan pada ilmu kedokteran dan ilmu pengetahuan pada umumnya memunculkan peralatan medis yang semakin canggih dalam memerangi penyakit atau melakukan deteksi lebih dini pada kondisi-kondisi tertentu. Salah satunya gangguan yang terjadi pada mata yaitu buta warna. Buta warna adalah suatu keadaan dimana seseorang tidak dapat membedakan warna tertentu yang bisa dibedakan oleh orang yang mempunyai mata normal. Seseorang yang menderita buta warna dapat disebabkan oleh kelainan gen sejak lahir atau akibat gangguan dari mengkonsumsi obat-obatan secara berlebihan. Buta warna

umumnya diderita oleh laki-laki, sedangkan perempuan sebagai pembawa gen/sifat keturunan.

Deteksi dini pada buta warna dapat dilakukan dengan menggunakan Tes Ishihara dan Tes Farnsworth Munsell, karena sebagian besar dunia kerja membutuhkan pegawai yang mempunyai mata normal, misalnya pada dunia industri, pendidikan maupun dalam pemerintahan. Oleh sebab itu tes Ishihara dan Farnsworth Munsell berperan penting dalam hal ini.

Tes Ishihara adalah sebuah metode pengetesan buta warna yang dikembangkan oleh Dr. Shinobu Ishihara. Tes ini pertama kali dipublikasikan pada tahun 1917 di Jepang. Sejak saat itu, tes ini terus digunakan di seluruh dunia, sampai sekarang. Tes buta warna Ishihara terdiri dari lembaran yang didalamnya 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.

Tes berikutnya adalah tes Farnsworth Munsell. Tes ini berfungsi sebagai tes lanjutan dari tes Ishihara yang hanya dapat menentukan kelainan partial atau tidaknya. Sedangkan tes Farnsworth Munsell, bisa melakukan skrining kelemahan warna tertentu, seperti kelemahan terhadap warna merah (*protan*), kelemahan terhadap warna hijau (*deutan*).

Kedua tes Ishihara dan Farnsworth Munsell ini mempunyai kelemahan yaitu berupa media tes. Media yang digunakan adalah lembaran kertas bagi Ishihara dan koin-koin warna dari kertas bagi tes Farnsworth Munsell. Media tes ini

sendiri hanya dapat dilakukan pada ruangan bercahaya putih dengan intensitas penerangan yang cukup, sehingga melakukan tes buta warna ini tidak bisa di sembarang tempat/ruangan dengan bercahaya redup dan menggunakan cahaya kemerahan atau lampu pijar. Hal ini merupakan salah satu dari kelemahan tes konvensional, karena jika penerangan ruangan tidak sesuai dengan ketentuan standar, maka warna pada media tes pun akan berubah. Media lembaran kertas bagi tes Ishihara pun mempunyai kelemahan berupa pemudaran warna, mudah robek, dan bisa saja salah satu dari lembaran tes terselip ataupun hilang. Sedangkan media koin-koin warna pada tes Farnsworth Munsell sendiri, memiliki kelemahan berupa pemudaran warna, mudah robek, dan bentuk koin yang sangat kecil, sehingga bisa hilang.

Untuk mengatasi masalah tersebut peneliti merancang suatu sistem tes buta warna yang sudah terkomputerisasi, dimana sistem tersebut menggunakan aplikasi yang berbasis Android agar mudah diakses oleh siapapun pengguna *smartphone* android.

Berdasarkan latar belakang masalah diatas, peneliti tertarik untuk melakukan penelitian terhadap permasalahan yang berkaitan dengan penyakit buta warna untuk dapat memberikan suatu alternatif tes dalam menangani permasalahan yang ada. Oleh karena itu, maka dalam penyusunan Skripsi ini peneliti mengambil judul “**Sistem Pakar Diagnosa Buta Warna Berbasis Andorid**”.

1.2 Identifikasi Masalah

Berdasarkan dari latar belakang masalah diatas, maka dapat diidentifikasi masalah sebagai berikut:

1. Pengujian buta warna menggunakan Tes Ishihara masih menggunakan media buku/lembaran kertas yang mudah robek dan luntur warnanya.
2. Pengujian buta warna menggunakan Tes Farnsworth Munsell masih menggunakan koin-koin kertas dalam pengujiannya dan membutuhkan ruangan bercahaya putih sehingga melakukan tes buta warna ini tidak bisa di sembarang tempat/ruangan dengan bercahaya redup dan menggunakan cahaya kemerahan atau lampu pijar.
3. Dibutuhkan media tes Ishihara dan Farnsworth Munsell yang lebih memudahkan seorang pakar dalam pengujiannya dan tahan lama dalam penggunaannya.

1.3 Batasan Masalah

Dalam penyusunan Skripsi ini, peneliti membatasi pembatasan hanya pada:

1. Aplikasi ini diperuntukkan kepada pengguna untuk mengetahui jenis buta warna yang ada pada dirinya.
2. Pembangunan sistem pakar menggunakan metode *forward chaining*.
3. Aplikasi ini berbasis android, sehingga diharapkan dapat digunakan secara bersamaan oleh banyak pengguna *smartphone* android.
4. Aplikasi sistem pakar diagnosa buta warna dibuat menggunakan penyimpanan data-data penunjang menggunakan database SQLite Editor Android dan bahasa pemrograman android.

1.4 Rumusan Masalah

Sesuai dengan permasalahan yang diangkat pada latar belakang penulisan, maka masalah yang akan dibahas dapat dirumuskan sebagai berikut:

1. Bagaimana membuat sistem pakar yang dapat digunakan sebagai alat bantu diagnosa buta warna secara *valid* menggunakan *smartphone* android?
2. Bagaimana mengimplementasikan tes buta warna berbasis android sehingga dapat diakses dengan mudah menggunakan *smartphone* android?

1.5 Tujuan Penelitian

Tujuan dari penyusunan skripsi ini adalah:

1. Membuat sistem pakar yang dapat digunakan sebagai alat bantu diagnosa buta warna secara *valid* menggunakan *smartphone* android.
2. Mengimplementasikan/mengaplikasikan tes buta warna berbasis android sehingga dapat diakses dengan mudah menggunakan *smartphone* android.

1.6 Manfaat Penelitian

1. Bagi Peneliti

Dapat meningkatkan pengetahuan dan kreativitas dalam menyelesaikan permasalahan dalam bidang sistem pakar berdasarkan keilmuan yang dimilikinya.

2. Bagi Masyarakat

Dapat mempermudah masyarakat untuk melakukan pengecekan mata dan dapat mengetahui hasil diagnosa dari tes buta warna tersebut, sehingga apabila mengalami buta warna dapat ditindak lanjuti oleh dokter mata.

BAB II

KAJIAN PUSTAKA

2.1 Kajian Teori

2.1.1 Definisi Sistem Pakar

Sistem pakar pertama kali dikembangkan oleh priset kecerdasan buatan pada dasawarsa 1960-an dan 1970-an dan diterapkan secara komersial pada tahun 1980-an. Bentuk umum sistem pakar adalah suatu program yang dibuat berdasarkan suatu set aturan yang menganalisis informasi (biasanya diberikan oleh pengguna suatu sistem) mengenai suatu kelas masalah spesifik serta analisis matematis dari masalah tersebut. Tergantung dari desainnya, sistem pakar juga mampu merekomendasikan suatu rangkaian tindakan pengguna untuk dapat menerapkan koreksi. Sistem ini memanfaatkan kapabilitas penalaran untuk mencapai suatu kesimpulan.

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. (Martin dan Oxman dalam Kusriani 2006:11)

Sistem Pakar atau *Expert System* adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar. (Darkin dalam Muhammad Arhami 2003:109)

Dari kedua pernyataan diatas menunjukkan bahwa Sistem Pakar (*Expert System*) adalah suatu sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Atau dengan kata lain sistem pakar adalah sistem yang didesain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli. Diharapkan dengan sistem ini, orang awam dapat menyelesaikan masalah tertentu baik ‘sedikit’ rumit ataupun rumit sekalipun ‘tanpa’ bantuan para ahli dalam bidang tersebut. Sedangkan bagi para ahli, sistem ini dapat digunakan sebagai asisten yang berpengalaman.

Sistem pakar merupakan cabang dari *Artificial Intelligence* (AI) yang cukup tua karena sistem ini telah mulai dikembangkan pada pertengahan tahun 1960. Sistem pakar yang muncul pertama kali adalah General-purpose problem solver (GPS) yang dikembangkan oleh Newl dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN, DENDRAL, XCON & XSEL, SOPHIE, Prospector, FOLIO, DELTA, dan sebagainya (Kusumadewi, 2003).

Suatu sistem dikatakan sistem pakar apabila memiliki ciri-ciri sebagai berikut (Kusumadewi, 2003):

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak pasti
3. Dapat mengemukakan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami
4. Berdasarkan pada kaidah atau rule tertentu
5. Dirancang untuk dikembangkan secara bertahap
6. Keluarannya atau output bersifat anjuran.

Adapun banyak manfaat yang dapat diperoleh dengan mengembangkan sistem pakar, antara lain (Kusumadewi, 2003):

1. Masyarakat awam non-pakar dapat memanfaatkan keahlian di dalam bidang tertentu tanpa kesadaran langsung seorang pakar
2. Meningkatkan produktivitas kerja, yaitu bertambahnya efisiensi pekerjaan tertentu serta hasil solusi kerja
3. Penghematan waktu dalam menyelesaikan masalah yang kompleks
4. Memberikan penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang
5. Pengetahuan dari seorang pakar dapat dikombinasikan tanpa ada batas waktu
6. Memungkinkan penggabungan berbagai bidang pengetahuan dari berbagai pakar untuk dikombinasikan.

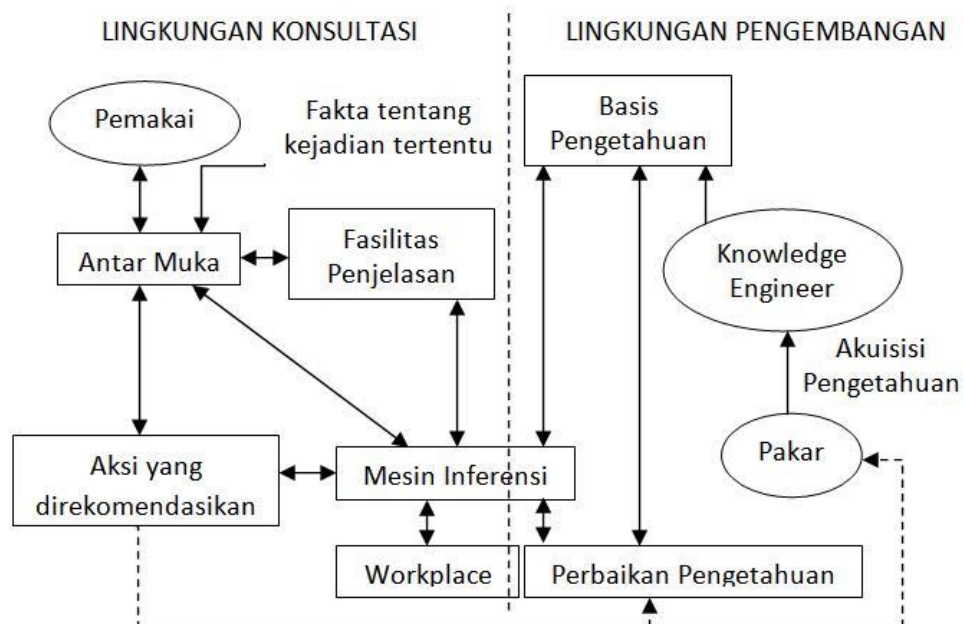
Selain banyak manfaat yang diperoleh, ada juga kelemahan pengembangan sistem pakar, yaitu (Kusumadewi, 2003):

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional.

Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak.

2.1.2 Komponen Sistem Pakar

Sistem pakar biasanya terdiri atas beberapa komponen yang masing-masing berhubungan seperti terlihat pada Gambar 1. (Kadir,A., 2003:2004)



Gambar 1 Komponen Sistem Pakar

Basis pengetahuan, berisi pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Basis pengetahuan tersusun atas 2 elemen dasar:

1. Fakta, misalnya : situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu
2. Aturan, yang mengarahkan penggunaan pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus.

Mesin inferensi (*Inference Engine*), merupakan otak dari sistem pakar. Juga dikenal sebagai penerjemah aturan (*rule interpreter*). Kerja mesin inferensi meliputi:

1. Menentukan aturan mana akan dipakai
2. Menyajikan pertanyaan kepada pemakai, ketika diperlukan
3. Menambahkan jawaban ke dalam memori Sistem Pakar
4. Menyimpulkan fakta baru dari sebuah aturan
5. Menambahkan fakta tadi ke dalam memori.

Papan Tulis (*Blackboard/Workplace*) adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data antarmuka pemakai (*User Interface*). Sistem Pakar mengatur komunikasi antara pengguna dan komputer.

Subsistem Penjelasan (*Explanation Facility*) adalah kemampuan untuk menjejak (*tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting

untuk transfer pengetahuan dan pemecahan masalah. Komponen subsistem penjelasan harus dapat menyediakannya yang secara interaktif menjawab pertanyaan pengguna, misalnya:

1. Mengapa pertanyaan tersebut anda tanyakan?
2. Seberapa yakin kesimpulan tersebut diambil?
3. Mengapa alternatif tersebut ditolak?
4. Apa yang akan dilakukan untuk mengambil suatu kesimpulan?
5. Fakta apalagi yang diperlukan untuk mengambil kesimpulan akhir?

Sistem Penghalusan Pengetahuan (*Knowledge Refining System*). Seorang pakar mempunyai sistem penghalusan pengetahuan, artinya, mereka bisa menganalisa sendiri performa mereka, belajar dari pengalaman, serta meningkatkan pengetahuannya untuk konsultasi berikutnya.

2.1.3 Teknik Representasi Pengetahuan

Teknik representasi pengetahuan digunakan untuk merepresentasikan basis pengetahuan yang diperoleh ke dalam suatu skema/diagram tertentu sehingga dapat diketahui relasi/keterhubungan antara suatu data dengan data yang lain.

Terdapat beberapa teknik representasi pengetahuan

yang biasa digunakan dalam pengembangan suatu sistem pakar, yaitu:

1. *Rule-Based Knowledge*

Pengetahuan direpresentasikan dalam suatu bentuk fakta (*facts*) dan aturan (*rules*). Bentuk representasi ini terdiri atas premis dan kesimpulan.

2. *Frame-Based Knowledge*

Pengetahuan direpresentasikan dalam suatu bentuk hirarki atau jaringan *frame*.

3. *Object-Based Knowledge*

Pengetahuan direpresentasikan sebagai jaringan dari objek-objek. Objek adalah elemen data yang terdiri dari data dan metoda (proses).

4. *Case-Base Reasoning*

Pengetahuan direpresentasikan dalam bentuk kesimpulan kasus (*cases*).

2.1.4 Metode *Forward Chaining*

Metode *Forward chaining* dimulai dari sejumlah fakta-fakta yang telah diketahui, untuk mendapatkan suatu fakta baru dengan memakai *rule-rule* yang memiliki ide dasar yang cocok dengan fakta dan terus dilanjutkan sampai mendapatkan tujuan atau sampai tidak ada *rule* yang punya ide dasar yang cocok atau sampai mendapatkan fakta.

Metode ini digunakan untuk mendapatkan solusi dari problem berdasarkan kondisi yang ada, atau suatu proses yang memulai pencarian dari premis atau data menuju konklusi (*data-driven*). Cara kerjanya adalah *inference engine* menyalakan atau memilih *rule-rule* dimana bagian premisnya cocok dengan informasi yang ada pada bagian *working memory*.

Sebagai contoh, asumsikan *listing* berikut ini valid, masing-masing variable dari setiap *rule* menginginkan nilai benar (*true*) dan tujuannya adalah variabel G :

R1 : JIKA A DAN C MAKA E;

R2 : JIKA D DAN C MAKA H;

R3 : JIKA B DAN E MAKA F;

R4 : JIKA B MAKA C;

R5 : JIKA F MAKA G;

Langkah-langkah dari komputer adalah sebagai berikut :

1. Komputer mengambil *rule* yang pertama (R1).

Terdapat A pada posisi JIKA karena nilai A belum ada pada memori dan tidak ada *rule* yang memuat konklusi A, maka komputer akan menanyakan jawaban dari A kepada *user* (diasumsikan benar).

2. Setelah A terpenuhi maka giliran C yang akan

- diperiksa nilainya, tetapi tidak ada nilai C pada memori. Meski demikian C merupakan konklusi dari *rule* R4. Sistem akan beralih ke *rule* R4.
3. Terdapat B pada posisi JIKA dari *rule* R4. Karena tidak terdapat pada memori dan bukan merupakan konklusi dari *rule*, maka komputer akan menanyakan jawaban untuk B (diasumsikan dijawab benar). Dengan demikian konklusi C diinputkan ke memori.
 4. Dengan diinputkannya konklusi C pada memori, maka syarat untuk konklusi E pada *rule* R4 terpenuhi juga. Konklusi E diinputkan ke memori, kemudian komputer akan mencari *rule* dengan E pada posisi JIKA dan akan mendapatkan *rule* R3.
 5. Pada *rule* R3 nilai B dan E terdapat pada memori dengan nilai benar, maka konklusi F terpenuhi dan akan diinputkan ke memori. Komputer kemudian mencari lagi *rule* dengan F pada posisi JIKA dan akan mendapatkan *rule* R5.
 6. Konklusi G pada *rule* R5 terpenuhi, karena F bernilai benar dan sistem pakar akan menghasilkan kesimpulan G.



2.1.5 Diagnosa



Diagnosa atau Diagnosis dalam kamus besar bahasa Indonesia adalah penentuan suatu penyakit dengan meneliti (memeriksa) gejala-gejalanya (Sugono, 2008:350).

2.1.6 Versi Android


Perkembangan versi Android dalam Murya (2014: 8-11) dapat dilihat pada tabel 1 berikut:



Tabel 1 Versi Android

Versi	Simbol dan Nama	Penjelasan
1.1		Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
1.5	 <i>Cupcake</i>	Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung


		dari telepon, Bluetooth A2DP support, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar.
1.6	 <p><i>Donut</i></p>	Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang dintegrasikan, Gestures, dan Text-to-speech engine, kemampuan dial kontak, teknologi text tochange speech (tidak tersedia pada semua ponsel; pengadaan resolusi WVGA).
2.0/2.1	 <p><i>Eclair</i></p>	Perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI (user interface) dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk

		kamera 3,2 MP, digital zoom, dan bluetooth 2.1.
2.2	 <p><i>Frozen Yoghurt</i></p>	<p>Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.</p>
2.3	 <p><i>Gingerbread</i></p>	<p>Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang,</p>

		dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.
3.0/3.1	 <p><i>Honeycomb</i></p>	Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.
4.0		Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan

	 <p><i>Ice Cream Sandwich</i></p>	<p>fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan Near Field Communication(NFC).</p>
4.1	 <p><i>Jelly Bean</i></p>	<p>Pada versi Android Jelly Bean terdapat penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI (User Interface) yang baru dan pencarian melalui melalui Voice Search yang lebih cepat. Google now juga menjadi bagian yang diperbaharui. Google now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lau lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1,</p>

		muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.
4.4	 <p><i>KitKat</i></p>	<p>Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Pembaruan UI dengan bar status dan navigasi transparan pada layar depan. NFC Host Card Emulation sebagai emulator kartu pintar. WebViews berbasis Chromium. Perluasan fungsionalitas bagi layanan pendengar notifikasi. Kemampuan untuk menentukan aplikasi SMS standar. Terdapat Sensor batching, Step Detector, dan Counter API. Peningkatan tampilan mode layar penuh, tombol perangkat lunak dan status bar bisa diakses dari tepi dengan cara menggesek. Penyeimbang audio, pemantauan audio, dan peningkatan suara audio. Perekam aktivitas layar yang terintegrasi Inframerah.</p>

		Dukungan Bluetooth Message Access Profile (MAP).
5.0	 <p style="text-align: center;"><i>Lollipop</i></p>	<p>Google kembali meluncurkan versi Android terbarunya yang diberi nama <i>Android Lollipop</i> atau disingkat dengan Android L. Beberapa media terkemuka seperti Tech Times menuturkan bahwa Android Lollipop ini akan dirilis pada tanggal 3 November 2014. Android Lollipop 5.0 kabarnya memiliki fitur super canggih dan sudah tertanam dalam Tablet Nexus 9 (Google Nexus 9).</p>

2.1.7 Buta Warna

Menurut Ganong (2003) Buta warna merupakan penyakit keturunan yang terekspresi pada para pria, tetapi tidak pada wanita. Wanita secara genitis sebagai carrier. Istilah buta warna atau colour blind sebetulnya salah pengertian dan menyesatkan, karena seorang penderita buta warna tidak buta terhadap seluruh warna. Akan lebih tepat bila disebut gejala defisiensi daya melihat warna tertentu saja atau colour vision difiency. Orang yang mengalami buta warna tidak hanya melihat warna hitam putih saja,

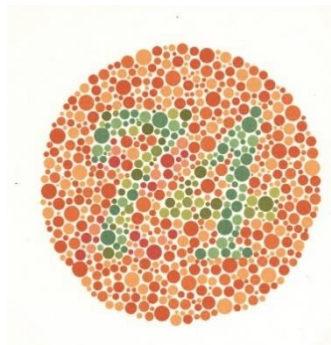
tetapi yang terjadi adalah kelemahan/penurunan pada penglihatan warna-warna tertentu misalnya kelemahan pada warna merah, hijau, kuning, dan biru. Buta warna permanen biasanya terjadi karena faktor keturunan. Sedangkan orang yang tidak mengalami buta warna dapat mengalami buta warna apabila terjadi faktor-faktor tertentu seperti kecelakaan. Tipe buta warna ada 3 yaitu monokromat/buta warna total (*monochromacy*), dikromat/buta warna parsial (*dichromacy*) dan anomaly trikromat (*anomalous trichromacy*).

2.1.8 Alat Tes Kebutaan Warna

Dalam penelitian ini, peneliti menggunakan dua jenis tes buta warna, yaitu:

1. Tes Ishihara

Menurut Guyton (1997) Metode ishihara yaitu metode yang dapat dipakai untuk menentukan dengan cepat suatu kelainan buta warna didasarkan pada penggunaan kartu bertitik-titik, seperti Gambar 2.



Gambar 2 Contoh Tes Buta Warna dengan Metode Ishihara

Kartu ini disusun dengan menyatukan titik-titik yang mempunyai bermacam-macam warna. Pada gambar 1. orang normal akan melihat angka “74”, sedangkan penderita buta warna merah-hijau akan melihat angka “21”.

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 didalamnya 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-isochromatism*). Tabel 1 menunjukkan contoh kartu tes buta warna dengan metode ishihara. Dalam tes buta warna ishihara ini digunakan 38 plate atau lembar gambar. Di mana gambar-gambar tersebut memiliki urutan 1 sampai 38. Tahapan dalam pemeriksaan buta warna dengan metode ishihara, yaitu:

1. Menggunakan buku Ishihara 38 plate.
2. Yang perlu diperhatikan:

- a. Ruangan pemeriksaan harus cukup pencahayaannya.
 - b. Lama pengamatan untuk membaca angka masing-masing lembar maksimum 10 detik.
3. Pada tes pembacaan buku Ishihara dapat disimpulkan:
- a. Normal
 - b. Buta warna Parsial:
 - 1) Bila plate no. 1 sampai dengan no 17 hanya terbaca 13 plate atau kurang.
 - 2) Bila terbaca angka-angka pada plate no. 18, 19, 20 dan 21 lebih mudah atau lebih jelas dibandingkan dengan plate no. 14, 10, 13, dan 17.

Bila ragu-ragu kemungkinan buta warna parsial dapat dites dengan:

- 1) Membaca angka-angka pada plate no. 22, 23, 24, dan 25. Pada orang normal, akan terbaca dengan benar angka-angka pada plate-plate tersebut diatas secara lengkap (dua rangkap). Pada penderita buta warna parsial hanya terbaca satu angka pada tiap-tiap plate tersebut diatas.

2) Menunjuk arah alur pada plate no. 26, 27, 30, 31, 32, 33, 34, 35, 36, 37, dan 38.

Untuk orang normal bisa menunjuk alur secara benar sedangkan untuk buta warna parsial dapat menunjukkan adanya alur dari satu sisi yang lainnya.

c. Buta warna total

Pada plate no.28 dan 29, untuk orang normal, tidak bisa menunjukkan adanya alur, sedangkan untuk penderita buta warna parsial dapat menunjukkan adanya alur dari satu sisi.

Tabel 2 Pengambilan kesimpulan Tes Buta Warna

Kesimpulan Tes	Pengambilan Kesimpulan
Buta Warna Total	1. Jika gambar 1 salah dan jawaban gambar lain diabaikan
Buta Warna Parsial	<ol style="list-style-type: none"> 1. Jika gambar 1 benar, gambar 2 sampai gambar 16 ada salah lebih dari 3 atau 2. Jika gambar 1 benar, gambar 22 sampai gambar 24 jawaban hanya benar pada salah satu gambar atau 3. Jika gambar 1 benar, Jika gambar 18 sampai gambar 21 terlihat angka.
Normal	<ol style="list-style-type: none"> 1. Jika gambar 1 sampai gambar 17 benar, atau gambar 1 harus benar dan lebih dari 13 gambar dijawab benar 2. Gambar 22 sampai gambar 24 benar atau 2 gambar benar

2. Tes Farnsworth Munsell

Peralatan berikutnya adalah tes Farnsworth Munsell. Tes ini merupakan tes kelanjutan dari tes Ishihara. Pada tes Ishihara, hasil yang didapat hanyalah mendiagnosis apakah pasien mengalami buta warna parsial atau tidak. Sedangkan pada tes Farnsworth Munsell, tes ini bisa mendiagnosis dengan melakukan skrining kelemahan warna tertentu, seperti kelemahan terhadap warna merah (*protan*), kelemahan terhadap warna hijau (*deutan*). Untuk pengujian tes Farnsworth Munsell D-15 ini pun tidaklah sulit. Pasien diminta untuk menghafal urutan-urutan warna pada koin-koin yang sudah disiapkan. Lalu melakukan acak warna pada koin-koin warna tersebut. Setelah koin-koin warna tersebut diacak, maka pasien diminta untuk mengurutkan kembali warna-warna yang ada. Setelah selesai, maka kita bisa menyocokkan urutan warna yang telah disusun kembali oleh pasien. Untuk lebih jelas mengenai koin-koin warna pada tes Farnsworth Munsell, bisa dilihat pada Gambar 7.

(sumber:http://www.unitycolor.com/images/product_images/pup_images/171_0.jpg)



Gambar 3 Koin-koin Warna Farnsworth Munsell

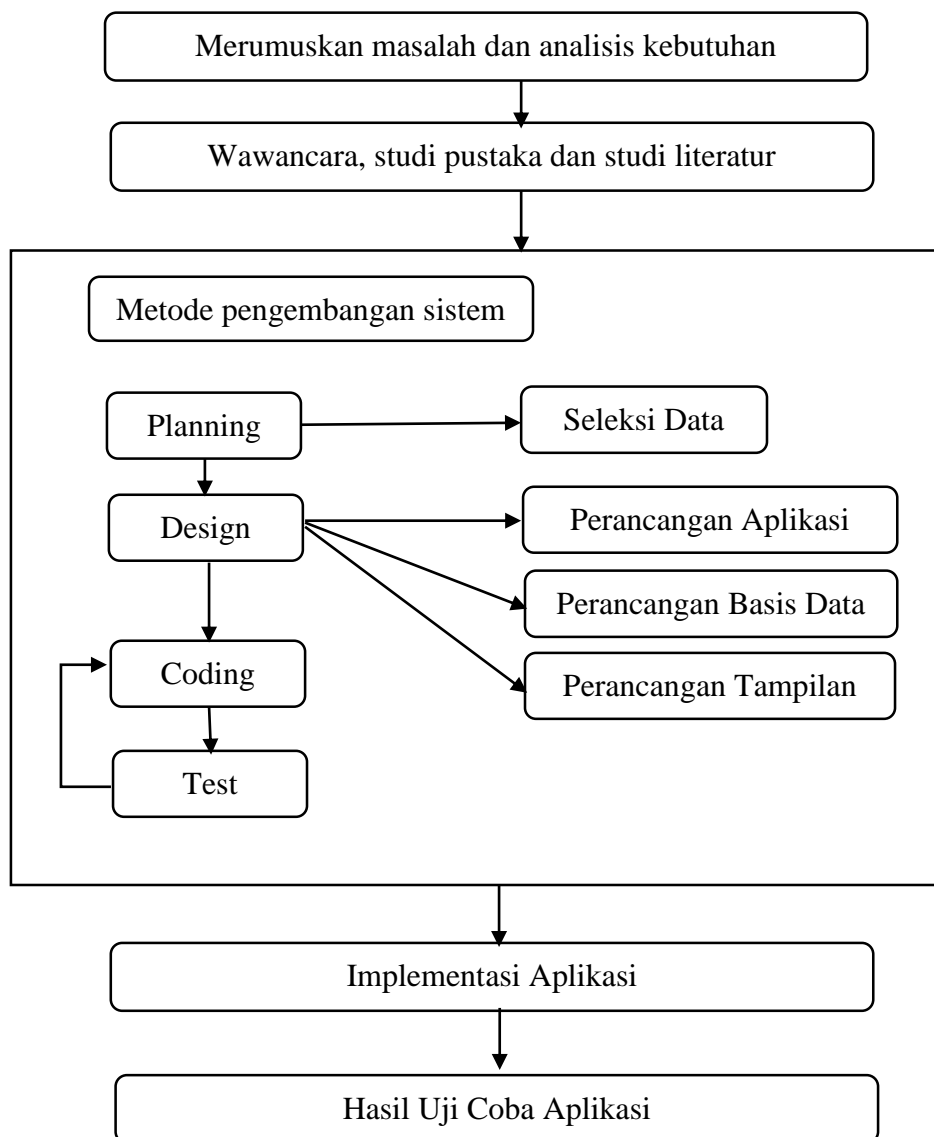
2.2 Kerangka Berpikir

Sistem pakar untuk diagnosa buta warna merupakan suatu sistem pakar yang dirancang sebagai alternatif untuk membantu masyarakat dalam memberikan solusi terhadap dugaan adanya buta warna pada orang tersebut. Pengetahuan ini didapat dari berbagai

sumber diantaranya penelitian dan seminar yang dilakukan pakar dalam bidangnya serta buku yang berhubungan dengan buta warna. Basis pengetahuan disusun sedemikian rupa ke dalam suatu database dengan beberapa tabel diantaranya tabel gejala dan tabel aturan untuk mempermudah kinerja sistem dalam penarikan kesimpulan. Penarikan kesimpulan dalam pakar ini menggunakan metode inferensi *forward chaining*.

Sistem pakar ini akan melakukan pengtesan buta warna menggunakan plat-plat tes Ishihara terlebih dahulu, dimana setiap hasil pelat tes akan membawa *user* kepada tes selanjutnya sampai mendapatkan hasil akhir. Kemudian dilihat dari hasil tes Ishihara tersebut, apabila hasil tes Ishihara menunjukkan pasien mengalami buta warna parsial maka pasien akan melanjutkan tes berikutnya yaitu tes Farnsworth Munsell. Pada tes Ishihara, hasil yang didapat hanyalah mendiagnosis apakah pasien mengalami buta warna parsial atau tidak. Sedangkan pada tes Farnsworth Munsell, tes ini bisa mendiagnosis dengan melakukan skrining kelemahan warna tertentu, seperti kelemahan terhadap warna merah (*protan*), kelemahan terhadap warna hijau (*deutan*). Pasien akan diminta untuk melanjutkan tes Farnsworth Munsell apabila hasil yang didapat dari tes Ishihara yaitu buta warna parsial. Apabila hasil menunjukkan bahwa pasien tidak buta warna/mata normal maka akan muncul hasil dari tes tersebut. Ada hasil akhir sistem pakar akan

menampilkan jenis buta warna yang diderita.. Dengan berbasiskan sistem operasi Android, diharapkan tingkat kemudahan pengaksesan sistem pakar ini lebih mudah dan efisien, dilihat dari perkembangan teknologi telepon selular dewasa ini. Hampir semua lapisan masyarakat sudah tidak asing lagi dengan sistem operasi berbasis Andorid.



Gambar 4 Kerangka Berpikir

2.3 Hipotesis

Sugiono (2008:93) menyatakan bahwa: “Hipotesis merupakan jawaban sementara terhadap rumusan masalah penelitian, oleh karena itu rumusan masalah biasanya disusun dalam bentuk kalimat pertanyaan.” Dikatakan sementara karena jawaban yang diberikan baru berdasarkan teori yang relevan, belum pada fakta-fakta empiris yang diperoleh melalui pengumpulan data. Jadi hipotesis dapat juga dinyatakan sebagai jawaban teoritis terhadap rumusan masalah penelitian dan belum jawaban yang empirik.

Berdasarkan rumusan masalah penelitian yang dikemukakan diatas, maka dirumuskan hipotesis sebagai berikut: Sistem pakar diagnosa buta warna mampu mendiagnosa buta warna secara *valid* dan pengguna aplikasi ini bisa menjalankan aplikasi tersebut tanpa kendala.

BAB III

METODE PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak

Proses pengembangan aplikasi ini akan menggunakan metode sekuensial linier (*waterfall*). Metode *waterfall* merupakan metode pengembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test, dan pemeliharaan.

3.2 Alat dan Bahan Pembuatan Aplikasi

Dalam proses pembuatan aplikasi ini, alat dan bahan yang digunakan adalah perangkat keras dan perangkat lunak sebagai berikut:

1. Laptop dengan spesifikasi:
 - a. Processor: Intel(R) Core(TM) i3-2310M
 - b. Memory (RAM): 2048MB
 - c. Hardisk 225GB
2. Perangkat lunak:
 - a. *Operating System*: Windows 8
 - b. ADT Eclipse
 - c. SQLite Editor Android
 - d. *Browser*: Google Chrome/Mozilla Firefox

3.3 Tahapan Perancangan Aplikasi

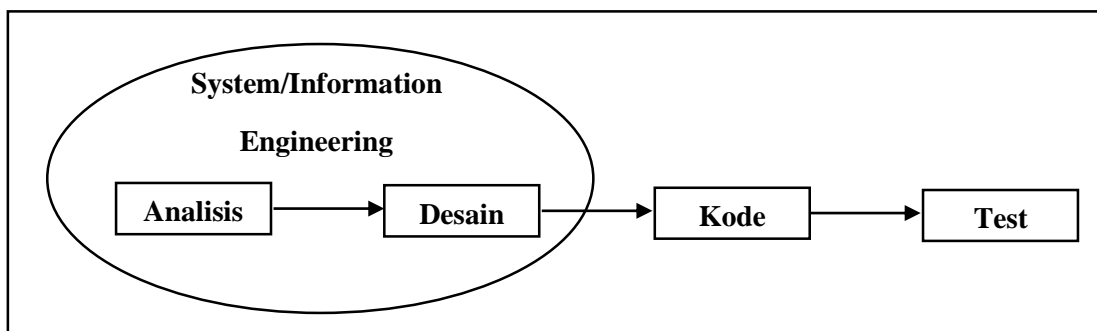
Dalam perancangan Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android ini menggunakan metode *forward chaining* yaitu sebagai berikut:

1. Membuat basis pengetahuan berupa penyakit buta warna, gejala-gejala yang timbul serta solusi dari buta warna tersebut.
2. Mengelompokkan jenis-jenis buta warna menjadi kode-kode yang digunakan untuk merancang tabel keputusan dan pohon keputusan.
3. Merancang tabel keputusan sistem pakar diagnosa buta warna sebagai representasi pengetahuan.
4. Mengubah representasi pengetahuan tersebut menjadi aturan dalam bentuk IF-Then *rule*.
5. Memasukkan rule ke dalam script program Eclipse ADT (Android Development Tool), source code aplikasi bisa dilihat pada lampiran 4.

Penjelasan lebih lengkap dari tahap ini dapat dilihat pada sub bab 4.1.

3.4 Tahapan Pengembangan Perangkat Lunak/Aplikasi

Berikut ini adalah tahapan dari model *waterfall* (Roger S. Pressman, 2002:37):



Gambar 4 Model Waterfall

1. Analisis

Pada tahap ini, peneliti melakukan analisis dan pengumpulan data tentang buta warna, mengumpulkan berbagai informasi tentang segala sesuatu yang akan berhubungan dengan pengembangan aplikasi buta warna akan diproduksi. Informasi ini diperoleh dari berbagai teori penunjang keadaan di lapangan. Sesuai dengan kegiatan peneliti yang akan meneliti apakah hasil diagnosis aplikasi ini sesuai dengan hasil diagnosis manual menggunakan buku isihara dan koin-koin warna Farnsworth Munsell.

2. Desain

a. Perancangan UML

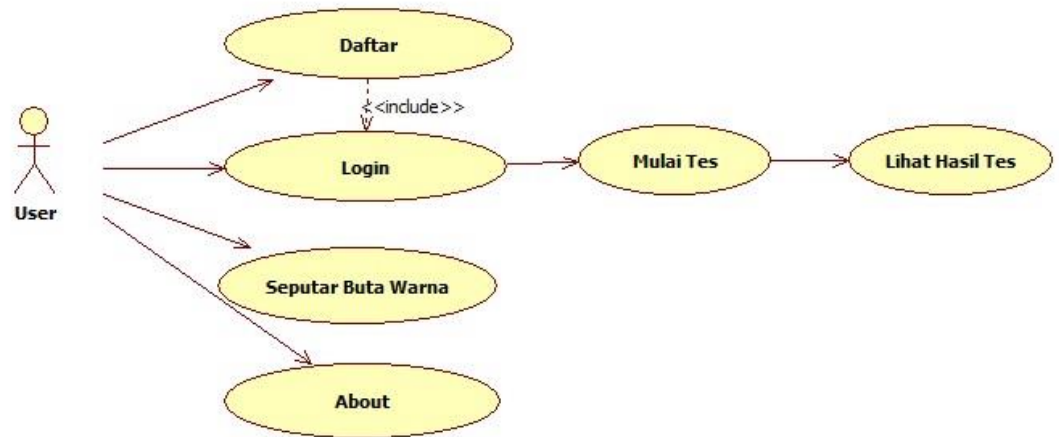
UML (*Unified Modeling Language*) adalah sebuah bahasa yang menggunakan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan dari sebuah sistem pengembangan *software* berbasis *object-oriented*. Tahap ini akan menjelaskan tentang dua model diagram pada UML dari aplikasi diagnosa buta warna berbasis android:

1. *Use Case Diagram*

2. *Activity Diagram*

Gambar 11 dan Gambar 12 adalah penjelasan dari UML tersebut:

1. Use Case Diagram

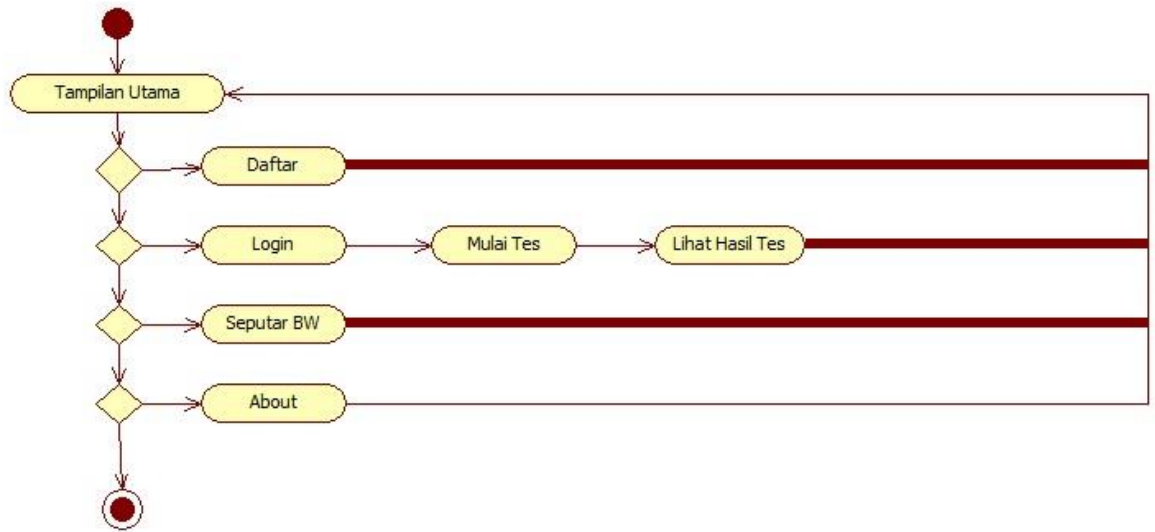


Gambar 5 Use Case Diagram

Pengguna dapat menjalankan menu menu berupa Menu Daftar, Login, Seputar Bua Warna, dan About, serat menu lanjutan Mulai Tes dan Hasil Tes di dalam menu lanjutan Login. Ketika pengguna memilih menu Mulai Tes maka akan tampil hasil diagnosa dari tes tersebut.

2. Activiy Diagram

Activity Diagram ini menggambarkan alur aktivitas sistem dari aplikasi diagnosa buta warna berbasis android, bagaimana masing-masing alur berawal, *desicion* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Dibawah ini akan dibahas, *activity diagram* program yang dilakukan *user* terhadap aplikasi yang sudah digunakan:



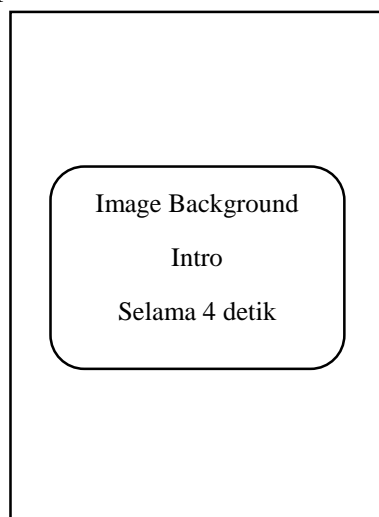
Gambar 6 Activity Diagram

b. Desain Tampilan Aplikasi

Desain layout Aplikasi Sitem Pakar Diagnosa Bua Warna Berbasis

Android adalah sebagai berikut:

a. Tampilan halaman intro



Gambar 7 Tampilan Halaman Intro

b. Tampilan halaman utama

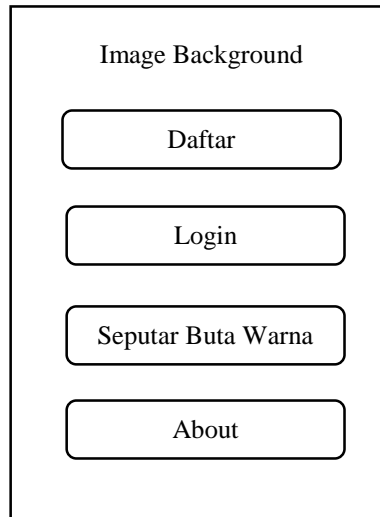


Image Background

Daftar

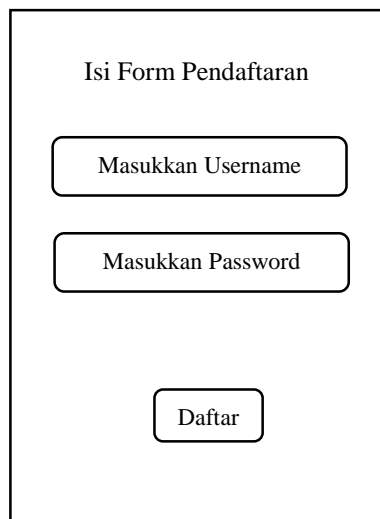
Login

Seputar Buta Warna

About

Gambar 8 Tampilan Halaman Utama

c. Tampilan Menu Daftar



Isi Form Pendaftaran

Masukkan Username

Masukkan Password

Daftar

Gambar 9 Tampilan Menu
Daftar

d. Tampilan menu Login

The diagram shows a vertical rectangular container representing the login menu. At the top, the text "Isi form Login" is centered. Below this, there are three rounded rectangular input fields stacked vertically. The first field is labeled "Masukkan Username", the second is labeled "Masukkan Password", and the third is labeled "Login dan Mulai Tes".

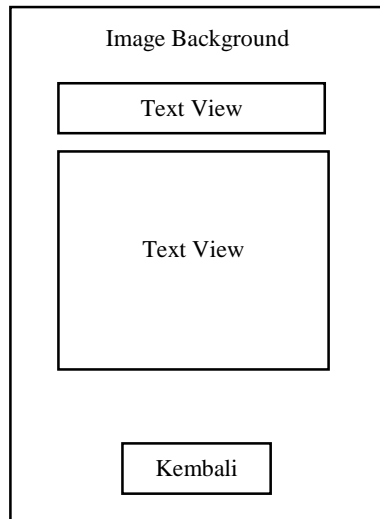
Gambar 10 Tampilan Menu Login

e. Tampilan Mulai Tes

The diagram shows a vertical rectangular container representing the start test screen. At the top, the text "Image Background" is centered. Below this, there is a small rectangular box labeled "Gambar ke-". Underneath that is a larger rounded rectangular box labeled "Image View". Below the "Image View" is a rectangular box labeled "Text View". At the bottom, there is a rectangular box labeled "Option Buton" which contains three small circles arranged horizontally.

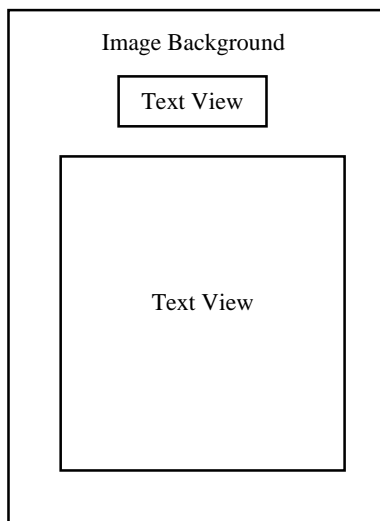
Gambar 11 Tampilan Mulai Tes

f. Tampilan Hasil Tes

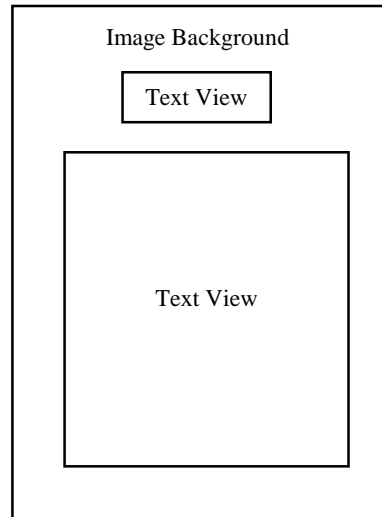


Gambar 12 Tampilan Hasil Tes

g. Tampilan menu Seputar Buta Warna

Gambar 13 Tampilan Menu
Seputar Buta Warna

h. Tampilan menu About



Gambar 14 Tampilan Menu About

3. Kode

Pengkodean (*coding*) merupakan proses menerjemahkan desain ke dalam suatu bahasa yang bisa dimengerti oleh komputer. Perancangan sistem ini terdiri dari perancangan basis data, perancangan antar muka pemakai dan sarana-sarana pendukung sistem. Pada sistem pakar ini menggunakan bahasa pemrograman Android menggunakan ADT Eclipse dan SQLite Editor Android sebagai alat bantu dalam pengimplementasian kode kedalam aplikasi pembangunan sistem pakar.

4. Test

Proses pengujian berfokus pada logika internal software, memastikan bahwa semua pertanyaan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-

kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

Produk penelitian ini berupa aplikasi diagnosa buta warna berbasis android diuji cobakan pada 22 orang yang terdiri dari penderita buta warna total, buta warna parsial dan mata normal. Tujuan dari uji coba produk ini adalah untuk mendapatkan nilai validitas penggunaan aplikasi diagnosa buta warna berbasis android. Apabila pada hasil uji coba menggunakan aplikasi diagnosa buta warna sesuai dengan hasil diagnosa dokter, maka aplikasi tersebut *valid* dan dapat digunakan untuk tes buta warna. Beberapa metode dapat digunakan untuk melakukan uji validitas, pada penelitian ini menggunakan metode *One Feature: Single Decision Threshold* (Bemmel, 1997 dalam Informatika Kesehatan, 2009 : 83).

Model Keputusan

		+	-	
Kenyataan	+	TP	FN	100%
	-	FP	TN	100%

Perhitungan ini diukur oleh besarnya sensitivitas dan spesifisitas uji coba, dengan rumus sebagai berikut :

$$\text{Sensitivitas} = \frac{\sum \text{True Positive (TP)}}{\sum \text{True Positive (TP)} + \sum \text{False Negative (FN)}}$$

$$\text{Spesifisitas} = \frac{\sum \text{True Negative (TN)}}{\sum \text{True Negative (TN)} + \sum \text{False Positive (FP)}}$$

Metode ini menggunakan empat parameter diagnosis, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).

Keterangan sebagai berikut :

- 1) True Positive (TP) menunjukkan jumlah pasien yang teridentifikasi terserang penyakit X baik berdasarkan data riil maupun diagnosis,
- 2) True Negative (TN) menunjukkan jumlah pasien yang tidak terserang penyakit X baik data riil maupun diagnosis,
- 3) False Positive (FP) menunjukkan jumlah pasien yang tidak menderita penyakit X berdasarkan data riil namun diagnosis memutuskan terserang penyakit,
- 4) False Negative (FN) menunjukkan jumlah pasien yang menderita penyakit X berdasarkan data riil namun diagnosis memutuskan tidak terserang penyakit.

Pada penelitian ini nilai Akurasi dari aplikasi diagnosis buta warna sangat dipengaruhi oleh tingginya nilai TP dan TN, atau rendahnya nilai FP dan FN.

Secara matematis dapat dirumuskan sebagai berikut:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Keterangan:

TP: True Positive

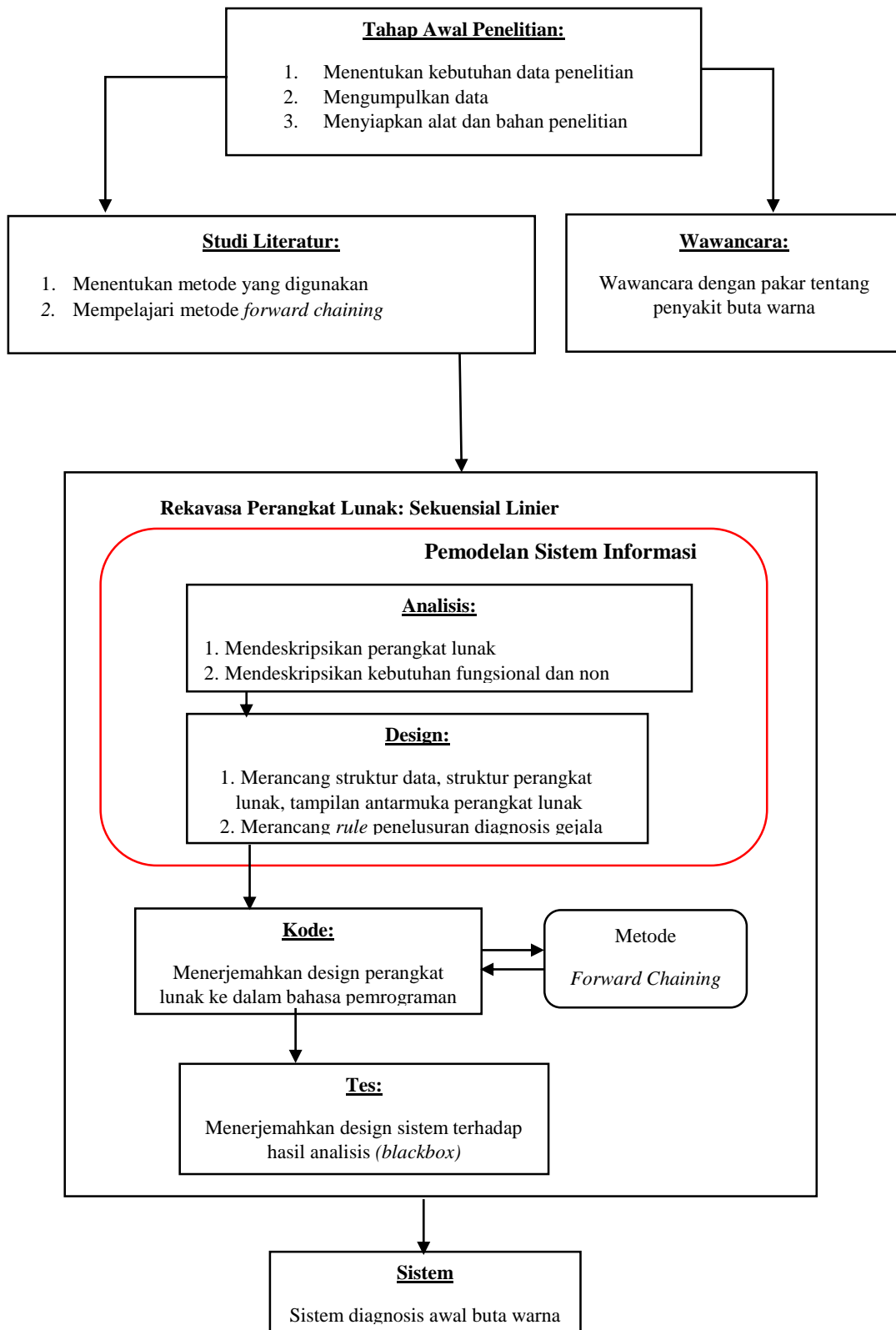
TN: True Negatives

FP: False Positive

FN: False Negative

3.5 Tahapan Penelitian

Tahapan penelitian merupakan tahapan yang akan dilakukan peneliti untuk mempermudah dalam melakukan penelitian. Tahapan penelitian sistem pakar diagnosa buta warna berbasis android digambarkan seperti pada Gambar 15. Sebelum melakukan penelitian, terlebih dahulu peneliti melakukan uji coba aplikasi kepada dokter. Hal ini ditujukan untuk mengetahui seberapa besar keakurasian dari aplikasi buta warna ini dibandingkan dengan tes buta warna buku ishihara dan Farnsworth Munsell. Responden yang terpilih melakukan tes buta warna secara manual menggunakan buku ishihara dan Farnsworth Munsell dengan pengamatan dokter mata dan tes buta warna menggunakan aplikasi ini. Setelah pengujian aplikasi dilakukan oleh dokter mata dan dinyatakan aplikasi ini layak dan *valid* digunakan sebagai alat bantu diagnosa buta warna, maka dilanjutkan dengan melakukan penelitian kepada 22 orang yang terdiri dari buta warna parsial, buta warna total dan mata normal yang lebih jelasnya akan dijelaskan pada sub bab 4.1.



Gambar 15 Tahapan Penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. HASIL PENELITIAN

Pada subbab ini akan dijelaskan tentang tentang perancangan aplikasi sistem pakar diagnosa buta warna berbasis android. Aplikasi ini diharapkan dapat memberikan informasi diagnosa buta warna secara valid. Aplikasi ini dapat berjalan pada perangkat *mobile* dengan catatan perangkat yang digunakan telah *support* dengan android. Pada aplikasi ini menggunakan spesifikasi android dengan sistem operasi android ice cream sandwich 4.0.

4.1.1 Implementasi Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis

Android

Hasil implementasi dari Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android setelah melakukan tahap-tahap penelitian dalam prosedur yang telah ditetapkan:

a. Tampilan Halaman Intro

Tampilan halaman *intro* ini merupakan halaman yang paling utama ketika membuka aplikasi sistem pakar diagnosa buta warna. Pada gambar ini menjelaskan judul dari aplikasi yang berlangsung sekitar 4 detik saja. Bentuk dari halaman intro ini dapat dilihat pada Gambar 16.



SISTEM PAKAR BUTA WARNA



Gambar 16 Halaman Intro

b. Tampilan Halaman *Home*

Tampilan halaman home ini merupakan implementasi dari rancangan home. Pada gambar ini menjelaskan menu apa saja yang ada pada aplikasi sistem pakar ini, halaman home berisi menu utama user yang meliputi menu daftar, menu login, menu seputar buta warna dan menu about. Bentuk dari halaman home ini dapat dilihat pada Gambar 17.



Gambar 17 Halaman Home

c. Tampilan Menu Daftar

Tampilan menu daftar ini merupakan implementasi dari rancangan menu daftar. Halaman ini berisi form pendaftaran user yang berguna untuk pengisian data user sebelum melakukan tes agar datanya dapat tersimpan kedalam database. Bentuk menu ini dapat dilihat pada Gambar 18.

The image shows a mobile registration form with two columns of input fields. The left column contains fields for Username, Password, Jenis Kelamin (Gender), Tanggal Lahir (Date of Birth), Golongan Darah (Blood Type), and Alamat (Address). The right column contains fields for Password, Jenis Kelamin (Gender), Tanggal Lahir (Date of Birth), Golongan Darah (Blood Type), and Alamat (Address). A red 'Daftar' button is located at the bottom right of the form. The form is displayed on a mobile device with a status bar at the bottom showing the time 05:52 and various icons.

Field	Value
Username	[Redacted]
Password	[Redacted]
Jenis Kelamin	<input checked="" type="radio"/> Laki-Laki <input type="radio"/> Perempuan
Tanggal Lahir :	Pilih Tanggal [Redacted]
Golongan Darah	[Redacted]
Alamat	[Redacted]

Gambar 18 Halaman Menu Daftar

d. Tampilan Menu Login

Tampilan menu login ini merupakan implementasi dari rancangan menu login. Halaman ini berisi form login user dengan memasukkan username dan password yang sudah dibuat di menu daftar. Menu login ini berguna untuk melakukan tes isihara yang dilanjutkan dengan tes farnsworth munsell. Bentuk menu ini dapat dilihat pada gambar 19.

Username

Password

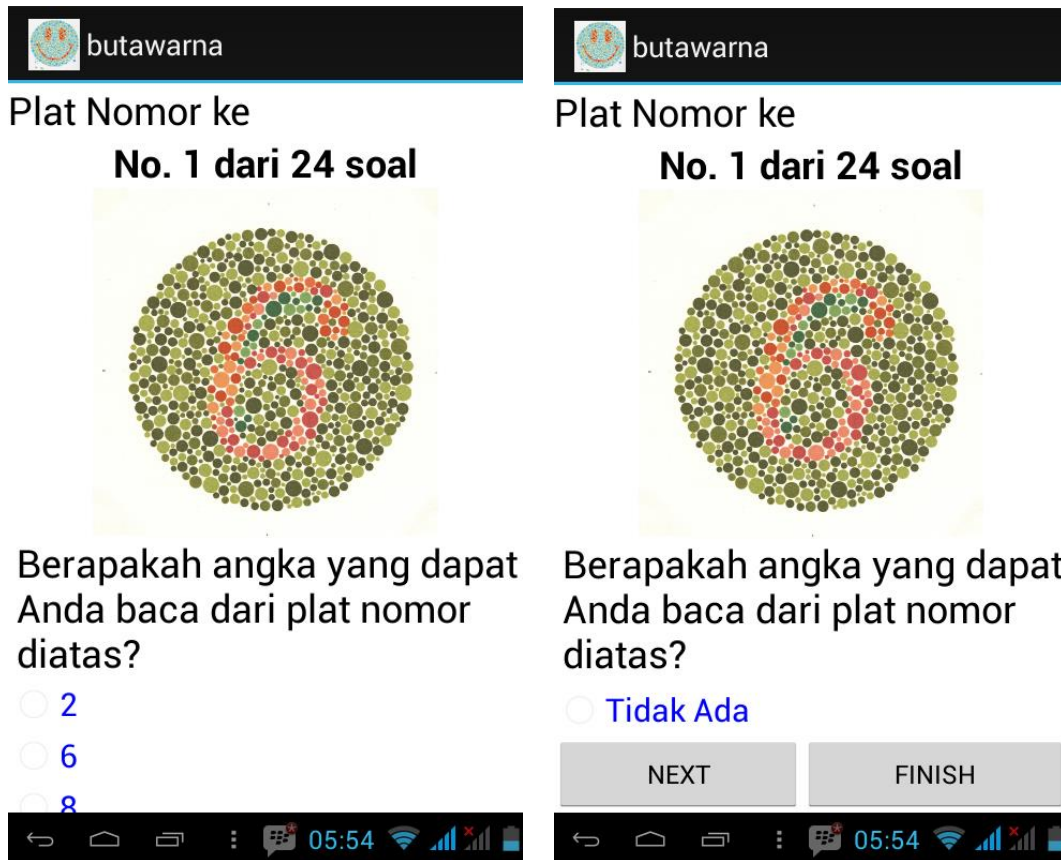
Login dan Mulai Tes



Gambar 19 Halaman Menu Login

e. Tampilan Menu Mulai Tes Ishihara

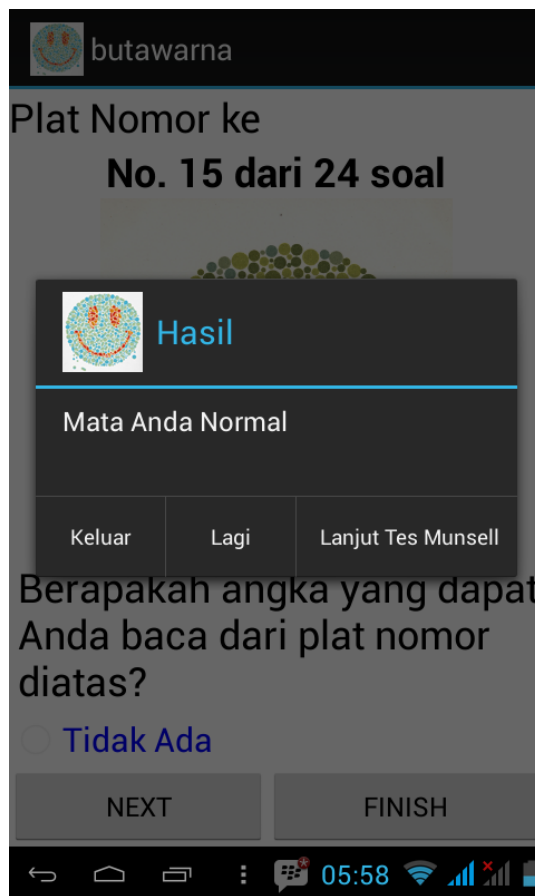
Tampilan menu mulai tes ini merupakan implementasi dari rancangan menu mulai tes. Halaman ini merupakan kelanjutan dari menu login yang kemudian langsung dilakukan pengetesan buta warna yang dimulai dengan tes isihara. Bentuk tampilan tes isihara dapat dilihat pada Gambar 20.



Gambar 20 Halaman Mulai Tes Ishihara

f. Tampilan Halaman Hasil Tes Ishihara

Tampilan hasil tes ishihara ini merupakan implementasi dari rancangan hasil tes. Halaman ini berisi hasil tes ishihara yang sudah dilakukan. Bentuk tampilan hasil tes ishihara dapat dilihat pada gambar 21.



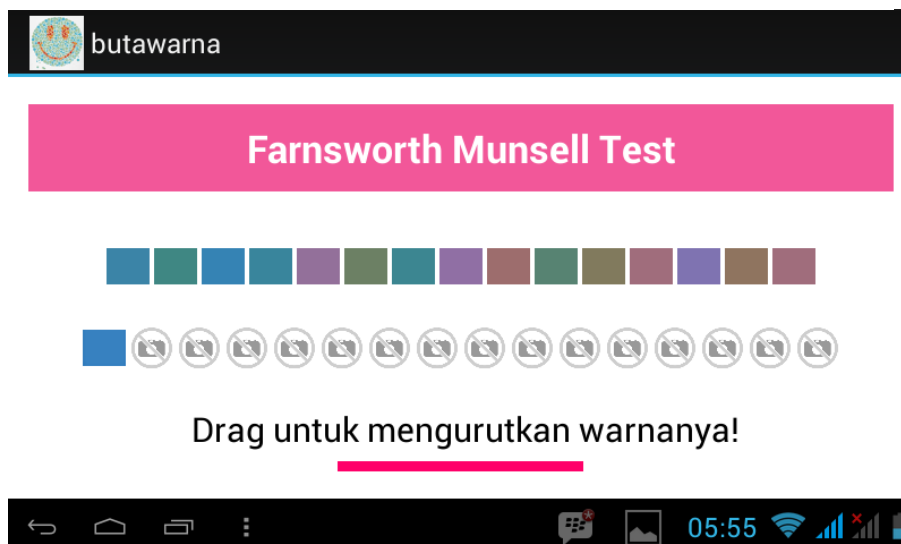
Gambar 21 Halaman Hasil Tes Ishihara

g. Tampilan Tes Farnsworth Munsell

Tampilan tes farnsworth munsell ini merupakan implementasi dari rancangan tes farnsworth munsell. Halaman ini berisi kelanjutan tes dari tes ishihara yang bertujuan untuk mengetahui kelemahan user yang mengalami buta warna, masuk dalam kategori lemah terhadap warna merah atau biru. Bentuk tampilan hasil tes ishihara dapat dilihat pada gambar 22 dan gambar 23.



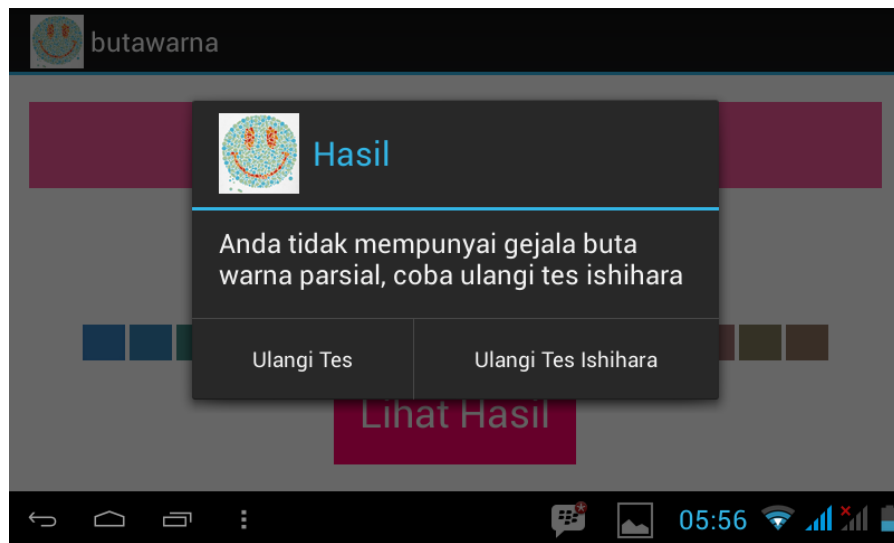
Gambar 22 Halaman Utama Tes Farnsworth Munsell



Gambar 23 Halaman Mulai Tes Farnsworth Munsell

h. Tampilan Halaman Hasil Tes Farnsworth Munsell

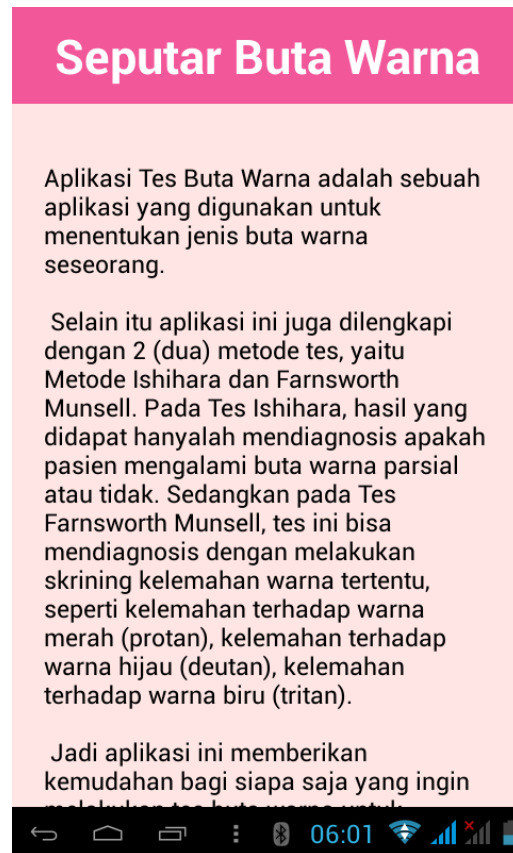
Tampilan hasil tes farnsworth munsell ini merupakan implementasi dari rancangan hasil tes farnsworth munsell. Halaman ini berisi hasil tes farnsworth munsell yang sudah dilakukan. Bentuk tampilan hasil tes isihara dapat dilihat pada gambar 24.



Gambar 24 Halaman Hasil Tes Farnsworth Munsell

i. Tampilan Halaman Menu Seputar Buta Warna

Tampilan menu seputar buta warna ini merupakan implementasi dari rancangan menu seputar buta warna. Halaman ini berisi pengetahuan seputar buta warna dan jenis-jenis buta warna. Bentuk tampilan seputar buta warna dapat dilihat pada gambar 25.



Gambar 25 Halaman Menu Seputar Buta Warna

j. Tampilan Halaman Menu About

Tampilan halaman menu about ini merupakan implementasi dari rancangan menu about. Halaman ini berisi tentang tujuan dibuatnya aplikasi ini dan tentang developernya. Bentuk tampilan menu about dapat dilihat pada gambar 26.



Gambar 26 Halaman Hasil Tes Ishihara

4.1.2 Implementasi Basis Pengetahuan Sistem Pakar Diagnosa Buta Warna Berbasis Android

Pada tahap ini peneliti menentukan konsep yang akan dikembangkan untuk menjadi sistem pakar diagnosa buta warna berbasis android yang dikonsultasikan kepada pakar. Inti pada tahapan ini adalah pengumpulan data dan informasi yang akan menjadi basis pengetahuan dari sistem pakar yang akan dikembangkan. Basis pengetahuan ini berisi seputar pengetahuan tes buta warna ishihara dan farnsworth munsell dan cara kerjanya untuk bisa dijadikan sistem pakar.

Hasil dari analisis peneliti terhadap pakar adalah basis pengetahuan dari plate tes ishihara:

Tabel 3 Tabel Nomor Plate Ishihara

Kode	No. Plate Ishihara
A	Plate No. 1
B	Plate No. 2
C	Plate No. 3
D	Plate No. 4
E	Plate No. 5
F	Plate No. 6
G	Plate No. 7
H	Plate No. 8
I	Plate No. 9
J	Plate No. 10
K	Plate No. 11
L	Plate No. 12
M	Plate No. 13
N	Plate No. 14
O	Plate No. 15
P	Plate No. 16
Q	Plate No. 17
R	Plate No. 18
S	Plate No. 19
T	Plate No. 20
U	Plate No. 21
V	Plate No. 22
W	Plate No. 23
X	Plate No. 24
Y	Plate No. 25
ZA	Buta Warna Total
ZB	Buta Warna Parsial/Sebagian
ZC	Mata Normal

Rules yang diketahui adalah:

1. Jika bisa menjawab 9-10 angka dengan benar, maka mata normal IF BENAR ≥ 9 THEN ZC
2. Jika bisa menjawab 4-8 angka dengan benar, maka buta warna parsial IF BENAR ≥ 4 AND BENAR ≤ 8 THEN ZB
3. Jika bisa menjawab 1-3 angka dengan benar, maka buta warna total IF BENAR ≤ 3 THEN ZA
4. Jika Plat 18 dibaca 5, Plat 19 dibaca 2, Plat 20 dibaca 45, Plat 21 dibaca 73, maka buta warna sebagian IF R = 5, S = 2, T = 45, U = 73 THEN ZB

Setelah melakukan pengujian tes isihara, bagi pasien yang mengalami buta warna parsial/sebagian akan melanjutkan tes farnsworth munsell untuk mengetahui kelemahan terhadap skrining warna. Pada tabel 3 adalah basis pengetahuan dari tes Farnsworth Munsell:

Tabel 4 Tabel Urutan Warna Farnsworth Munsell

Kode	Urutan Warna Farnsworth Munsell
AA	Urutan warna no.1 (warna dasar patokan tidak boleh diubah-ubah)
BB	Urutan warna no.2
CC	Urutan warna no.3
DD	Urutan warna no.4
EE	Urutan warna no.5
FF	Urutan warna no.6
GG	Urutan warna no.7

HH	Urutan warna no.8
II	Urutan warna no.9
JJ	Urutan warna no.10
KK	Urutan warna no.11
LL	Urutan warna no.12
MM	Urutan warna no.13
NN	Urutan warna no.14
OO	Urutan warna no.15
PP	Urutan warna no.16
QAA	Hasil Protan
QAB	Hasil Deutan

Rules yang diketahui adalah:

1. Jika warna urutannya 16-2-15-3-14-13-4-5-12-11-6-10-7-9-8,
maka protan

Variabel TANDAPROTAN = PP-BB-OO-CC-NN-MM-DD-
EE-LL-KK-FF-JJ-GG-II-HH

IF ANSWER = TANDAPROTAN THEN QAA

2. Jika warna urutannya 2-16-3-4-15-14-5-13-6-12-7-8-11-10-9,
maka deutan

Variabel TANDADEUTAN = BB-PP-CC-DD-OO-NN-EE-
MM-FF-LL-GG-HH-KK-JJ-II

IF ANSWER = TANDADEUTAN THEN QAB

4.1.3 Implementasi Antarmuka

Dari hasil analisis dan desain yang telah dilakukan, peneliti mengubahnya kedalam bentuk *code* dengan cara mengimplementasikan hasil analisis dan desain tersebut kedalam bahasa pemrograman ADT Eclipse. Implementasi dilakukan dalam bentuk *file* program. Implementasi antarmuka disusun dengan memanfaatkan bahasa pemrograman XML dan JAVA. Hasil implementasi antarmuka dapat dilihat pada tabel 4:

Tabel 5 Implementasi Antarmuka

Menu	Deskripsi	Nama File
Halaman Intro	Halaman berisi gambar singkat dari program diagnosa buta warna berbasis android	splash.xml
Halaman Utama	Halaman utama pada program ini berisi empat menu utama yaitu daftar, login, seputar buta warna, dan about	main.xml
Halaman Daftar	Halaman ini berisi form pendaftaran bagi yang ingin mengikuti tes buta warna	daftar.xml
Halaman Login	Halaman ini berisi form login bagi yang sudah mendaftar di menu daftar dan dilanjutkan dengan memulai tes buta warna dan diakhiri dengan hasil tes tersebut	login.xml
Halaman Tes Ishihara	Halaman ini berisi soal tes ishihara 24 plate	menuutama.xml
Halaman Tes Farnsworth Munsell	Halaman ini berisi tes farnsworth munsell yaitu kelanjutan dari tes ishihara bagi yang	tes_munsell.xml

	mengalami buta warna parsial	
Halaman Seputar Buta Warna	Halaman berisi tentang pengetahuan seputar buta warna, diantaranya pengertian buta warna dan faktor penyebab buta warna	seputar.xml
Halaman About	Halaman berisi tentang profil pembuat aplikasi dan tujuan dibuat aplikasi ini	about.xml

4.1.4 Pengujian Alpha

Pengujian alpha merupakan pengujian fungsional yang digunakan untuk menguji sistem yang telah dibuat dengan metode *black box*. Pengujian *black box* terfokus pada persyaratan fungsional perangkat lunak. Pada tabel 5 merupakan pengujian fitur Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android:

Tabel 6 Pengujian Black Box

Tampilan Halaman	Fitur yang diuji	Sesuai yang diharapkan?	
		Ya	Tidak
Halaman Intro:	Background muncul selama 5 detik	V	
Halaman Utama: Daftar	<i>Button</i> daftar tes menampilkan form yang berisi kolom nama dan username	V	
	Selesai mengisi form pendaftaran menampilkan pemberitahuan bahwa berhasil terdaftar	V	
	<i>Button</i> kembali untuk menampilkan halaman utama	V	
Halaman Utama: Login	<i>Button</i> Login menampilkan form yg berisi username dan password yang sudah didaftarkan di menu daftar	V	

	Selesai mengisi form Login kemudian dilanjutkan dengan Mulai Tes, Tes Ishihara bagi yang mempunyai mata normal, Tes Ishihara dilanjutkan farnsworth Munsell bagi yang menderita buta warna	V	
	Selesai dilakukan tes akan langsung muncul Hasil Tes	V	
	<i>Button</i> kembali menampilkan halaman utama	V	
Halaman Utama: Seputar Buta Warna	<i>Button</i> Seputar Buta Warna menampilkan informasi tentang buta warna diantaranya pengertian dan faktor penyebab buta warna	V	
Halaman Utama: About	<i>Button</i> About menampilkan tentang profil pembuat dan tujuan aplikasi ini dibuat	V	

4.1.5 Validitas Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android

Uji kelayakan aplikasi sistem pakar diagnosa buta warna dilakukan oleh dokter umum. Berikut ini merupakan validator dari aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android:

Tabel 7 Daftar Validator

No.	Nama	Jabatan	Instansi
1.	dr. Anies Setiowati, M.Gizi	Dokter umum	Poliklinik UNNES

Dari uji kelayakan produk yang telah dilakukan oleh validator, dinyatakan bahwa aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android bisa digunakan untuk tes buta warna atau valid. Setelah

dilakukan uji validitas pada pakar, maka aplikasi ini akan digunakan kepada sejumlah user untuk mengetahui validitas dari aplikasi inidan sesuai dnegan diagnosa pakar dibidang kesehatan. Pada tabel 8 berikut adalah data pengguna (*user*) aplikasi diagnosa buta warna berbasis android:

Tabel 8 Data Hajil Uji Coba Aplikasi

No.	Nama	Tes Aplikasi	BW Parsial
		Total / Parsial / Normal	Protan / Deutan
1	Pengguna 1	Normal	-
2	Pengguna 2	Normal	-
3	Pengguna 3	Normal	-
4	Pengguna 4	Normal	-
5	Pengguna 5	Normal	-
6	Pengguna 6	Parsial	Deutan
7	Pengguna 7	Normal	-
8	Pengguna 8	Total	-
9	Pengguna 9	Parsial	Deutan
10	Pengguna 10	Normal	Parsial
11	Pengguna 11	Normal	-
12	Pengguna 12	Normal	-
13	Pengguna 13	Parsial	Protan
14	Pengguna 14	Total	-
15	Pengguna 15	Parsial	Deutan
16	Pengguna 16	Normal	-
17	Pengguna 17	Normal	-
18	Pengguna 18	Parsial	Deutan
19	Pengguna 19	Total	-
20	Pengguna 20	Normal	-
21	Pengguna 21	Parsial	Protan
22	Pengguna 22	Normal	Parsial

Dari hasil uji coba aplikasi sistem pakar diagnosa buta warna diatas menunjukkan bahwa:

1. Sebanyak 9 orang dinyatakan positif menderita buta warna (buta warna parsial dan buta warna total), baik berdasarkan kenyataan maupun hasil pemeriksaan aplikasi (TP=9).
2. Sebanyak 11 orang dinyatakan negatif menderita buta warna (buta warna parsial dan buta warna total), baik berdasarkan kenyataan maupun hasil pemeriksaan aplikasi (TN=11).
3. Tidak ada sejumlah orang yang sebenarnya positif menderita buta warna (buta warna parsial dan buta warna total), namun hasil pemeriksaan menyatakan negative (FN=0).
4. Sebanyak 2 orang sebenarnya negatif menderita buta warna (buta warna parsial dan buta warna total), namun hasil pemeriksaan aplikasi menyatakan positif (FP=2).

4.2. PEMBAHASAN

Validitas pakar diperoleh setelah peneliti melakukan beberapa tahapan-tahapan uji validitas program, yang diantaranya adalah mencari pakar yang akan dijadikan sebagai validator/aplikasi ini, setelah mendapatkan hasil apa yang diperoleh dari pakar mengenai aplikasi ini kemudian diuji cobakan kepada 22 orang yang terdiri dari mata normal, buta warna parsial dan buta warna total.

Dari hasil uji validitas yang diperoleh menunjukkan bahwa pakar bidang kesehatan menyatakan aplikasi yang dibuat valid/bisa digunakan untuk tes buta warna. Pada tabel 9 berikut adalah Nilai Sensitifitas dan Spesifitsitas dari Aplikasi Diagnosa Buta Warna:

Tabel 9 Nilai Sensitifitas dan Spesifitsitas Aplikasi

		Kenyataan	
		Positive	Negative
Keputusan Hasil Uji Coba Aplikasi	Positive	9 (TP)	2 (FP)
	Negative	0 (FN)	11 (TN)
		Sensitifitas	Spesifisitas
		=TP / (TP+FN)	=TN / (FP+TN)
		=9 / (9+0)	=11 / (2+11)
		=100%	=85%

$$\text{Akurasi} = \frac{\text{TP (True Positive)} + \text{TN (True Negative)}}{\text{TP (True Positive)} + \text{TN (True Negative)} + \text{FP (False Positive)} + \text{FN (False Negative)}}$$

$$= \frac{9+11}{9+11+2+0}$$

$$= 0,90$$

Akurasi dalam prosentase = 90 %

Jadi Sistem Pakar Diagnosa Buta Warna Berbasis Android mampu mendiagnosa buta warna secara valid menggunakan *smartphone* android karena aplikasi ini mampu memberikan hasil pemeriksaan dengan nilai akurasi

90%. Berikut adalah penelitian buta warna dan hasilnya yang digunakan sebagai acuan dan penguat pada penelitian ini, yaitu hasil penelitian sistem pakar kebutaan warna yang dilakukan oleh Hari Murti, S.Kom, M.Cs dan Rina Candra Sani, S.Pd, M.Kom (2009) dalam Model Pendiagnosa Kebutaan Warna dengan Menggunakan Metode Ishihara antara lain sebagai berikut:

1. Rancangan model tersebut digunakan sebagai alat bantu untuk test pemeriksaan gangguan penglihatan terhadap warna, karena dari rancangan sistem pakar kebutaan warna memberikan hasil pemeriksaan yang sama seperti hasil pemeriksaan secara manual dengan buku/alat test Ishihara yang dilakukan oleh seorang dokter mata.
2. Analisa dan rancangan Sistem pakar kebutaan warna dapat digunakan sebagai pengganti seorang pakar dalam menjalankan tugas dalam melakukan pemeriksaan gangguan penglihatan, sehingga apabila pakar sedang tidak dapat menjalankan tugas untuk melakukan pemeriksaan maka tugas pakar dapat dibantu/digantikan oleh orang lain.
3. Dalam menganalisa kebutaan warna disediakan fasilitas pemutahiran basis pengetahuan sehingga dapat melakukan penambahan pengetahuan secara terus menerus, menambah plat soal baru disesuaikan dengan penyempurnaan buku/alat tes Ishihara edisi baru.

Tes buta warna Ishihara menggunakan 38 plate gambar, tetapi dalam penelitian Aplikasi Tes Buta Warna dengan Metode Ishihara berbasis Komputer oleh Ratri Widianingsih, Awang Harsa Kridalaksana, Ahmad Rofiq Hakim (2010) hanya ditampilkan 24 plate yang merupakan gambar-gambar

utama dari tes buta warna Ishihara. Dengan 24 plate ini sudah dapat disimpulkan kondisi orang yang dites apakah mengalami buta warna total, parsial atau normal. Hasil penelitian dari Aplikasi Tes Buta Warna Dengan Metode Ishihara berbasis Komputer ini antara lain adalah:

1. Untuk mengetahui apakah seseorang mengalami buta warna dapat dilakukan dengan salah satu metode Ishihara.
2. Aplikasi tes buta warna dengan metode Ishihara menggunakan komputer dapat digunakan untuk tes buta warna yang hasilnya sama dengan tes Ishihara secara manual.
3. Tes buta warna dengan metode Ishihara menggunakan komputer dapat mengidentifikasi penderita buta warna total, buta warna parsial, dan normal.

Selain itu juga ada Instrumen Pengujian Buta Warna Otomatis oleh Sofiar Augusta, Tony Mulia, dan M. Sidik yang hasil penelitiannya adalah sebagai berikut:

1. Percobaan menggunakan metode tes buta warna secara otomatis menggunakan perangkat lunak berbasis visual basic .Net menggunakan *notebook* dan *tablet* tidak berbeda dari hasil yang didapat dengan menggunakan Ishihara *book test*. Dengan kata lain, perangkat lunak yang digunakan, bisa dijadikan acuan dan pegangan dalam melakukan tes buta warna dan dapat membantu pengguna maupun dokter agar pekerjaan dalam melakukan tes buta warna menjadi mudah dan cepat.

2. Pada metode tes buta warna secara otomatis ini, masih terdapat kesalahan-kesalahan kecil yang diakibatkan kurang pemahannya pasien dalam menggunakan teknologi terbaru berupa layar sentuh pada *tablet*. Hal ini bisa diedukasikan terlebih dahulu kepada pasien agar dapat melakukan tes buta warna secara benar dengan menggunakan layar sentuh. Penelitian ini masih jauh dari sempurna dikarenakan berbagai keterbatasan yang ada, oleh karena itu direkomendasikan untuk melakukan penelitian lebih lanjut dengan menampilkan edukasi yang diperlukan, agar pasien bisa menggunakan teknologi terbaru berlayar sentuh. Memperbaiki hasil akhir pada tes Farnsworth Munsell, sehingga hasil yang diperoleh dapat ditampilkan secara otomatis.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian dan pengembangan aplikasi yang telah peneliti lakukan selama ini, dapat disimpulkan bahwa:

1. Sistem Pakar Diagnosa Buta Warna Berbasis Android mampu digunakan sebagai alat bantu diagnosa buta warna secara valid menggunakan *smartphone* android karena aplikasi ini mampu memberikan hasil pemeriksaan dengan nilai akurasi 90% dan telah diuji cobakan kepada 22 pengguna tanpa kendala.
2. Sistem Pakar Diagnosa Buta Warna Berbasis Android dapat diimplementasikan/diaplikasikan disemua perangkat android minimal versi API level 14 atau setara dengan Android 4.0-4.0.2 Ice Cream Sandwich.

5.2 Saran

Berdasarkan penelitian yang dilakukan, peneliti berharap agar aplikasi ini dapat dikembangkan lebih lanjut, sehingga menjadi lebih sempurna lagi. Saran terhadap pengembangan aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android ini adalah sebagai berikut:

1. Pengembangan sistem pakar ini selanjutnya dapat lebih dikembangkan lagi dengan menggunakan metode lain dan dapat menyimpan database yang tidak secara online tetapi dengan soal-

soal tes yang tetap random sehingga soal-soal tes tidak bergantung pada koneksi internet.

2. Pengembangan sistem pakar ini selanjutnya dapat memerlukan validasi dari pakar yang lebih sempurna.
3. Semua kekurangan yang tidak disadari oleh peneliti, dapat disempurnakan lagi.

DAFTAR PUSTAKA

- Murti, H., dan Rina Candra Noor Santi. 2011. *Aplikasi Pendiagnosa Kebutaan Warna dengan Menggunakan Pemrograman Borland Delphi*. Jurnal Teknologi Informasi DINAMIK Volume 16, No.2, Juli 2011 : 160170.
- Widianingsih, R., Awang Harsa Kridalaksana, dan Ahmad Rofiq Hakim. 2010. *Aplikasi Tes Buta Warna Dengan Metode Ishihara Berbasis Komputer*. Jurnal Informatika Mulawarman.
- Agusta, Sofiar., dan Tony Mulia, dan M. Sidik. 2012. *Impelentasi Pengujian Buta Warna*. JURNAL ILMIAH ELITE ELEKTRO, VOL. 3, NO. 1, MARET 2012: 15-22.
- Putra., Firmansyah. 2011. *Perancangan Sistem Pakar Identifikasi Penyakit Paru-paru Menggunakan Metode Forward Chaining*. Skripsi UIN Syarif Hidayatullah Jakarta.
- Sugiono., Prof. Dr. .2008. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung : Alfabeta.
- Mediafire. 2015. *Ishihara 38 Plate 768x789 Pixel.rar*. <http://www.mediafire.com/download/nz8hxujc09cnubo/Ishihara+38+Plate+768x789+Pxel.rar> [diakses, 30 Januari 2015].
- Wikipedia. 2014. Sensitivity and Specificity. http://en.wikipedia.org/wiki/Sensitivity_and_Specificity. [diakses, 28 Agustus 2015].
- Wikipedia. 2014. *Sensitivity and Specificity*. http://en.wikipedia.org/wiki/Sensitivity_and_Specificity. [diakses, 28 Agustus 2015].

Qurays, Khamid. 2015. *Jenis dan Macam-macam OS Android Terbaru*.
<http://juraganpulsa.my.id/apa-itu-android-dan-macam-macam-versi-android/>
[diakses, 12 November 2015]

Rohman, Feri Fahrur., dan Ami Faujjzah. 2008. *Rancang Bangun Aplikasi Sisem Pakar Untuk Menentukan Jenis Gangguan Perkembangan Pada Anak*. Media Informatika, VOL. 6, NO. 1, JUNI 2008: 1-23.

Lampiran 1 Surat Keterangan telah Melakukan Penelitian**SURAT PERNYATAAN**

Yang bertanda tangan dibawah ini :

Nama Pakar : dr. Anies Setiowati, M. Gizi

NIP :

Menerangkan bahwa sebenarnya bahwa :

Nama : Puspita Prabawati

NIM : 5302411215

Prodi : Pendidikan Teknik Informatika dan Komputer

Universitas : Universitas Negeri Semarang

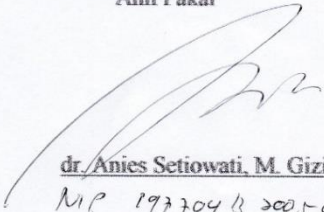
Adalah benar-benar telah melaksanakan penelitian dan uji validasi kepada pakar terkait tanggal 02 September 2015 dalam rangka skripsi dengan judul :

"Sistem Pakar Diagnosa Buta Warna Berbasis Android".

Demikian surat keterangan ini saya buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Semarang, 03 September 2015

Ahli Pakar


dr. Anies Setiowati, M. Gizi

NIP 19770413 2005-012003

Lampiran 2 Data Pasien Uji Coba Aplikasi

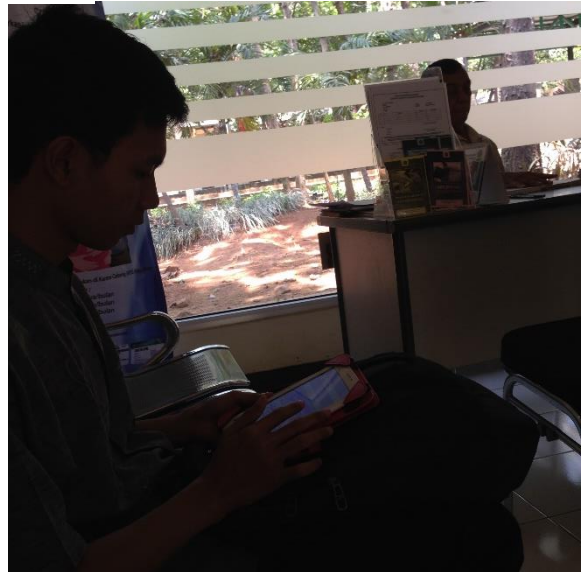
DATA UJI COBA APLIKASI SISTEM PAKAR DIAGNOSA BUTA WARNA BERBASIS ANDROID

PUSAT PELAYANAN KESEHATAN UNNES

No	Nama Pengguna	Kenyataan	Tes Aplikasi
			Total / Parsial / Normal
1	Cindykia	Normal	Normal
2	Arif S		Normal
3	Panqa P.		Normal
4	Ayu Paska S		Normal
5	Deni Novizar		Normal
6	Syamsul Sigit		Parsial / Deutan
7	Karamita		Normal
8	Marini		Total
9	Nugroho		Parsial / Deutan
10	Istian	Normal	Parsial
11	Su manto		Normal
12	Su prap to		Normal
13	SU HARTO		Parsial / Protan
14	Martinal		Total
15	Damiri		Parsial / Deutan
16	yuli		Normal
17	DALMUDI		Normal
18	Mulyanto		Parsial / Deutan
19	Margant		Total

20	NGATINEM		Normal
21	Wan sito		Parcial / Protan
22	Sujanti	Normal	Parcial.
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			

Lampiran 3 Dokumentasi



Uji Pengguna Aplikasi kepada Mata Normal dan Buta Warna



Uji Validitas oleh Pakar

Lampiran 4 Source Code Aplikasi Sistem Pakar Diagnosa Buta Warna Berbasis Android

MenuUtama.java

```

package com.android.butawarna;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.*;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class MenuUtama extends Activity {

    TextView No, soalIshihara;
    Button btnNext, btnSelesai;
    RadioGroup jawabIshihara;
    RadioButton radioA, radioB, radioC, radioD;
    ImageView img;
    int jawabnYgDiPilih[] = null;
    int jawabanYgBenar[] = null;
    boolean cekPertanyaan = false;
    int urutanPertanyaan = 0;
    List<Soal_Ishihara> listSoal;
    JSONArray jsonArray = null;
    private ProgressDialog pDialog;
    private static final String url =
"http://192.168.43.1/buta_warna/soal_ishihara.php";
    private static final String TAG_DAFTAR = "daftar_soal";
    private static final String TAG_ID = "id_soal";
    private static final String TAG_SOAL = "soal";
    private static final String TAG_A = "jwbn_a";
    private static final String TAG_B = "jwbn_b";
    private static final String TAG_C = "jwbn_c";
    private static final String TAG_D = "jwbn_d";
    private static final String TAG_BENAR = "benar_ishihara";
    private static final String TAG_GAMBAR = "gambar";
    public ImageLoader imageLoader;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.menuutama);
    listSoal = new ArrayList<Soal_Ishihara>();
    imageLoader = new ImageLoader(getApplicationContext());
    img = (ImageView) findViewById(R.id.img);
    No = (TextView) findViewById(R.id.No);
    soalIshihara = (TextView) findViewById(R.id.soalIshihara);
    btnNext = (Button) findViewById(R.id.btnNext);
    btnSelesai = (Button) findViewById(R.id.btnSelesai);
    jawabIshihara = (RadioGroup)
findViewById(R.id.jawabIshihara);
    radioA = (RadioButton) findViewById(R.id.radioA);
    radioB = (RadioButton) findViewById(R.id.radioB);
    radioC = (RadioButton) findViewById(R.id.radioC);
    radioD = (RadioButton) findViewById(R.id.radioD);

    btnSelesai.setOnClickListener(klikSelesai);
    btnNext.setOnClickListener(klikBerikut);
    new GetSoal().execute();
}

// @Override
// public void onClick(View v) {

//}

private class GetSoal extends AsyncTask<Void, Void, Void> {
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // menampilkan progress dialog
        pDialog = new ProgressDialog(MenuUtama.this);
        pDialog.setMessage("Silahkan Tunggu . . .");
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected Void doInBackground(Void... arg0) {
        // Membuat Service handler class instance
        ServiceHandler sh = new ServiceHandler();
        // membuat request le ur; dan meminta response
        String jsonStr = sh.makeServiceCall(url,
ServiceHandler.GET);
        Log.d("Response: ", ">" + jsonStr);
        if (jsonStr != null) {
            try {
                JSONObject jsonObject = new
JSONObject(jsonStr);
                // GET JSON Array Mode
                jsonArray =
jsonObject.getJSONArray(TAG_DAFTAR);
                Soal_Ishihara s = null;
                // Pembuatan Looping Semua Contacts
                for (int i = 0; i < jsonArray.length(); i++) {

```

```

JSONObject c = jsonArray.getJSONObject(i);
s = new Soal_Ishihara();

String id_soal = c.getString(TAG_ID);
String soal = c.getString(TAG_SOAL);
String jwb_n_a = c.getString(TAG_A);
String jwb_n_b = c.getString(TAG_B);
String jwb_n_c = c.getString(TAG_C);
String jwb_n_d = c.getString(TAG_D);
String benar = c.getString(TAG_BENAR);
String gambar = c.getString(TAG_GAMBAR);

s.setId_soal(id_soal);
s.setSoal(soal);
s.setJwb_n_a(jwb_n_a);
s.setJwb_n_b(jwb_n_b);
s.setJwb_n_c(jwb_n_c);
s.setJwb_n_d(jwb_n_d);
s.setBenar_ishihara(benar);
s.setGambar(gambar);
listSoal.add(s);
}
} catch (JSONException e) {
    e.printStackTrace();
}
} else {
    Log.e("ServiceHandler", "Couldn't get any data
from the url");
}
return null;
}

@Override
protected void onPostExecute(Void result) {
    super.onPostExecute(result);
    // Matikan Progress Dialog
    if (pDialog.isShowing())
        pDialog.dismiss();

    jawabnYgDiPilih = new int[listSoal.size()];
    java.util.Arrays.fill(jawabnYgDiPilih, -1);
    jawabanYgBenar = new int[listSoal.size()];
    java.util.Arrays.fill(jawabanYgBenar, -1);
    setUpSoal();
}

private void setUpSoal() {
    Collections.shuffle(listSoal);
    tunjukkanPertanyaan(0, cekPertanyaan);
}

private void tunjukkanPertanyaan(int urutan_soal_soal, boolean
review) {
    if (urutan_soal_soal == 0) ;

```

```

try {
    jawabIshihara.clearCheck();
    Soal_Ishihara soal_ishihara = new Soal_Ishihara();
    soal_ishihara = listSoal.get(urutan_soal_soal);
    if (jawabanYgBenar[urutan_soal_soal] == -1) {
        jawabanYgBenar[urutan_soal_soal] =
Integer.parseInt(soal_ishihara.getBenar_ishihara());
    }

    String soalnya = soal_ishihara.getSoal();
    soalIshihara.setText(soalnya);
    jawabIshihara.check(-1);
    radioA.setTextColor(Color.BLUE);
    radioB.setTextColor(Color.BLUE);
    radioC.setTextColor(Color.BLUE);
    radioD.setTextColor(Color.BLUE);
    imageLoader.DisplayImage(soal_ishihara.getGambar(),
img);

    radioA.setText(soal_ishihara.getJwb_n_a());
    radioB.setText(soal_ishihara.getJwb_n_b());
    radioC.setText(soal_ishihara.getJwb_n_c());
    radioD.setText(soal_ishihara.getJwb_n_d());

    Log.d("", jawabanYgDiPilih[urutan_soal_soal] + "");
    if (jawabanYgDiPilih[urutan_soal_soal] == 1)
        jawabIshihara.check(R.id.radioA);
    if (jawabanYgDiPilih[urutan_soal_soal] == 2)
        jawabIshihara.check(R.id.radioB);
    if (jawabanYgDiPilih[urutan_soal_soal] == 3)
        jawabIshihara.check(R.id.radioC);
    if (jawabanYgDiPilih[urutan_soal_soal] == 4)
        jawabIshihara.check(R.id.radioD);

    pasangLabelDanNomorUrut();

    if (urutan_soal_soal == (listSoal.size() - 1)) {
        btnNext.setEnabled(true);

        if (review) {
            Log.d("periksa",
jawabnYgDiPilih[urutan_soal_soal] + "" +
jawabanYgBenar[urutan_soal_soal]);
            if (jawabnYgDiPilih[urutan_soal_soal] !=
jawabanYgBenar[urutan_soal_soal]) {
                if (jawabnYgDiPilih[urutan_soal_soal] ==
1)
                    radioA.setTextColor(Color.RED);
                if (jawabnYgDiPilih[urutan_soal_soal] ==
2)
                    radioB.setTextColor(Color.RED);
                if (jawabnYgDiPilih[urutan_soal_soal] ==
3)
                    radioC.setTextColor(Color.RED);
                if (jawabnYgDiPilih[urutan_soal_soal] ==
4)

```

```

        radioD.setTextTextColor(Color.RED);
    }
    if (jawabanYgBenar[urutan_soal_soal] == 1)
        radioA.setTextTextColor(Color.GREEN);
    if (jawabanYgBenar[urutan_soal_soal] == 2)
        radioB.setTextTextColor(Color.GREEN);
    if (jawabanYgBenar[urutan_soal_soal] == 3)
        radioC.setTextTextColor(Color.GREEN);
    if (jawabanYgBenar[urutan_soal_soal] == 4)
        radioD.setTextTextColor(Color.GREEN);
    }
    }
} catch (Exception e) {
    Log.e(this.getClass().toString(), e.getMessage(),
e.getCause());
}
}

private View.OnClickListener klikSelesai = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        aturJawaban_nya();
        // hitung berapa jawaban yang benar
        int jumlahJawabanYgBenar = 0;
        for (int i = 0; i < jawabanYgBenar.length; i++) {
            if ((jawabanYgBenar[i] != -1)
                && (jawabanYgBenar[i] ==
jawabnYgDiPilih[i]))
                jumlahJawabanYgBenar++;
        }
        AlertDialog tampilKotakAlert;
        tampilKotakAlert = new
AlertDialog.Builder(MenuUtama.this).create();
        tampilKotakAlert.setTitle("Hasil");
        tampilKotakAlert.setIcon(R.drawable.ic_launcher);
        if (jumlahJawabanYgBenar >= 9)
            tampilKotakAlert.setMessage("Mata Anda Normal");
        else
            if (jumlahJawabanYgBenar >= 4 && jumlahJawabanYgBenar
<=8)
                tampilKotakAlert.setMessage("Anda Mempunyai Gejala
buta Warna Sebagian Atau Parsial, Silahkan Lanjut ke Tes
Berikutnya");
            else tampilKotakAlert.setMessage("Anda Mempunyai
Gejala Buta Warna Total");

        tampilKotakAlert.setButton(AlertDialog.BUTTON_NEUTRAL,
"Lagi", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                java.util.Arrays.fill(jawabnYgDiPilih, -1);
                cekPertanyaan = false;
            }
        });
    }
}

```



```

        urutanPertanyaan = 0;
        tunjukkanPertanyaan(0, cekPertanyaan);
    }
});

tampilKotakAlert.setButton(AlertDialog.BUTTON_POSITIVE, " Lanjut
Tes Munsell", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int
which) {
        Intent i = new Intent(getApplicationContext(),
TesMunsell.class);
        finish();
        startActivity(i);
    }
});

tampilKotakAlert.setButton(AlertDialog.BUTTON_NEGATIVE, "Keluar",
new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int
which) {
        cekPertanyaan = false;
        finish();
    }
});

tampilKotakAlert.show();
};

private void aturJawaban_nya() {
    if (radioA.isChecked())
        jawabnYgDiPilih[urutanPertanyaan] = 1;
    if (radioB.isChecked())
        jawabnYgDiPilih[urutanPertanyaan] = 2;
    if (radioC.isChecked())
        jawabnYgDiPilih[urutanPertanyaan] = 3;
    if (radioD.isChecked())
        jawabnYgDiPilih[urutanPertanyaan] = 4;

    Log.d("", Arrays.toString(jawabnYgDiPilih));
    Log.d("", Arrays.toString(jawabanYgBenar));
}

private View.OnClickListener klikBerikut = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        aturJawaban_nya();
        urutanPertanyaan++;
        if (urutanPertanyaan >= listSoal.size())
            urutanPertanyaan = listSoal.size() - 1;
    }
};

```

```

        tunjukkanPertanyaan(urutanPertanyaan, cekPertanyaan);
    }
};

    private void pasangLabelDanNomorUrut () {
        No.setText("No. " + (urutanPertanyaan + 1) + " dari" +
listSoal.size() + "soal");
    }
}

```

DaftarUser.java

```

package com.android.butawarna;

import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.*;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class DaftarUser extends Activity implements
View.OnClickListener {
    // inisialisasi

    EditText EditUser, EditPass, EditGoldar, EditAlamat, tanggal;
    TextView txtTgl;
    Button btnDaftar;
    RadioGroup rg1;
    RadioButton radioLaki, radioPerempuan;
    String jns, tgl_lahir ;
    private DatePickerDialog datepicker;
    SimpleDateFormat format, format2;

    // inisialisasi progress bar
    private ProgressDialog pDialog;

    // memanggil Class JSONParser

```

```

JSONParser jsonParser = new JSONParser();
private static final String LINK_DAFTAR =
"http://192.168.43.1/buta_warna/registrasi.php";

private static final String TAG_BERHASIL = "sukses";
private static final String TAG_PESAN = "pesan";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    this.requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.daftar);

    format = new SimpleDateFormat("dd-MM-yyyy", Locale.US);
    format2 = new SimpleDateFormat("yyyy-MM-dd", Locale.US);

    EditUser = (EditText) findViewById(R.id.editUser);
    EditPass = (EditText) findViewById(R.id.editPass);
    EditGoldar = (EditText) findViewById(R.id.editGoldar);
    EditAlamat = (EditText) findViewById(R.id.editAlamat);
    tanggal1 = (EditText) findViewById(R.id.tampil_tanggal);
    txtTgl = (TextView) findViewById(R.id.tampil_tanggal);
    rg1 = (RadioGroup) findViewById(R.id.rg1);
    radioLaki = (RadioButton) findViewById(R.id.radioLaki);
    radioPerempuan = (RadioButton)
findViewById(R.id.radioPerempuan);

    btnDaftar = (Button) findViewById(R.id.btnDaftar);
    btnDaftar.setOnClickListener(this);
}
public void tanggal(View v) {
    settingTanggal();
}

public void settingTanggal() {
    Calendar c = Calendar.getInstance();
    datepicker = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker datePicker, int
tahun, int bulan, int hari) {
            Calendar tanggal = Calendar.getInstance();
            tanggal.set(tahun, bulan, hari);
            txtTgl.setText(format.format(tanggal.getTime()));
            tgl_lahir = format2.format(tanggal.getTime());
            Log.d("tgl_pesan", tgl_lahir);
        }
    }, c.get(Calendar.YEAR), c.get(Calendar.MONTH),
c.get(Calendar.DAY_OF_MONTH));
    datepicker.show();
}

@Override

```

```

public void onClick(View v) {

    if (v.getId() == R.id.btnDaftar) {

        String hasil = "";
        // jika yg di seleksi laki laki
        if (radioLaki.isChecked()) {
            //to String berfungsi mengkonversi ke String
            hasil = radioLaki.getText().toString();
            Log.d("laki", hasil);
        }
        else if (radioPerempuan.isChecked()) {
            hasil = radioPerempuan.getText().toString();
        }
        jns = hasil;
    }
    // aksi tombol setelah di klik
    new DaftarUserBaru().execute();
}

class DaftarUserBaru extends AsyncTask<String, String, String>
{

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(DaftarUser.this);
        progressDialog.setMessage("Save User...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(true);
        progressDialog.show();
    }

    @Override
    protected String doInBackground(String... args) {
        // periksa jika TAG_BERHASIL

        int berhasil;
        String nama_user = EditUser.getText().toString();
        String password = EditPass.getText().toString();
        String gol_darah = EditGoldar.getText().toString();
        String alamat = EditAlamat.getText().toString();

        try{
            List<NameValuePair> params = new
ArrayList<NameValuePair>();
            params.add (new BasicNameValuePair("nama_user",
nama_user));
            params.add (new BasicNameValuePair("password",
password));
            params.add(new
BasicNameValuePair("tgl_lahir_user", tgl_lahir));
            params.add(new BasicNameValuePair("gol_darah",
gol_darah));
            params.add(new BasicNameValuePair("alamat_user",
alamat));
            params.add(new

```

```

BasicNameValuePair("jns_kelamin_user", jns));
    Log.d ("jns",jns);

    // mengirim data dari user ke server
    JSONObject jsonObject =
jsonParser.makeHttpRequest (LINK_DAFTAR, "POST", params);

    // json response
    Log.d("Coba Login", jsonObject.toString());

    //json berhasil
    berhasil = jsonObject.getInt (TAG_BERHASIL);
    if ( berhasil == 1){
        Log.d("Proses Registrasi Berhasil",
jsonObject.toString());
        finish();
        return jsonObject.getString (TAG_PESAN);
    } else {
        Log.d("Registrasi Gagal ..",
jsonObject.getString (TAG_PESAN));
        return jsonObject.getString (TAG_PESAN);
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute (String s) {
    progressDialog.dismiss();
    if (s !=null) {
        Toast.makeText (DaftarUser.this, s,
Toast.LENGTH_LONG).show();
    }
}
}
}

```

LoginUser.java

```

package com.android.butawarna;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;

```

```

import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class LoginUser extends Activity implements
View.OnClickListener {
    private EditText isiUser, isiPass;
    private Button btnLogin;

    // progress bar
    private ProgressDialog pDialog;

    // memasang class JSON Parser
    JSONParser jsonParser = new JSONParser();

    // lokasi script untuk login
    private static final String LINK_UNTUK_LOGIN =
"http://192.168.43.1/buta_warna/login.php";

    // inisialisasi response JSON
    private static final String TAG_SUKSES = "sukses";
    private static final String TAG_PESAN = "tampilkan_pesan";
    private static final String TAG_ID = "id";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.login);

        // inisialisasi username dan password
        isiUser = (EditText) findViewById(R.id.isiUser);
        isiPass = (EditText) findViewById(R.id.isiPass);

        // inisialisasi button
        btnLogin = (Button) findViewById(R.id.btnLogin);

        // aksi button setelah diklik
        btnLogin.setOnClickListener(this);
    }
}

```

```

@Override
public void onClick(View v) {
    new MencobaLogin().execute();
}

class MencobaLogin extends AsyncTask<String, String, String> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(LoginUser.this);
        pDialog.setMessage("Loading...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected String doInBackground(String... params) {
        // jika sukses maka

        int jikaSukses;
        String username = isiUser.getText().toString();
        String password = isiPass.getText().toString();
        Log.d("user", username);
        Log.d("password", password);

        try {
            // menyusun parameter
            List<NameValuePair> usernameDanPassword = new
ArrayList<NameValuePair>();
            usernameDanPassword.add(new
BasicNameValuePair("nama_user", username));
            usernameDanPassword.add(new
BasicNameValuePair("password", password));

            // untuk mengecek username dan password sudah
terkirim ke server apa belum

            Log.d("Request", "start");
            // membuat HTTP request
            JSONObject json =
JSONParser.makeHttpRequest(LINK_UNTUK_LOGIN, "POST",
usernameDanPassword);
            // memeriksa log jawaban dari JSON
            Log.d("Coba Login", json.toString());

            //respon JSON tentang TAG_SUKSES
            jikaSukses = json.getInt(TAG_SUKSES);
            if (jikaSukses == 1) {
                Log.d("Login Sukses", json.toString());
                SharedPreferences sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(LoginUser.this);

```

```

        SharedPreferences.Editor editor =
sharedPreferences.edit();

        Intent i = new Intent(LoginUser.this,
MenuUtama.class);
        finish();
        startActivity(i);
        return json.getString(TAG_PESAN);
    } else {
        Log.d("Login Gagal...",
json.getString(TAG_PESAN));
        return json.getString(TAG_PESAN);
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(String s) {
    pDialog.dismiss();
    if (s != null) {
        Toast.makeText(LoginUser.this, s,
Toast.LENGTH_LONG).show();
    }
}
}
}
}

```

Seputar.java

```

package com.android.butawarna;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.View;
import android.view.Window;

/**
 * Created by Yudi Ardiansah on 7/21/2015.
 */
public class Seputar extends Activity implements
View.OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.seputar);
    }
}

```



```
        @Override
        public void onClick(View v) {

    }
}
```

About.java

```
package com.android.butawarna;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.View;
import android.view.Window;

/**
 * Created by Yudi Ardiansah on 7/21/2015.
 */
public class About extends Activity implements
View.OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.about);
    }

    @Override
    public void onClick(View v) {

    }
}
```