



**PENERAPAN METODE ISHIHARA UNTUK
PERANCANGAN APLIKASI TES BUTA WARNA
BERBASIS ANDROID**

SKRIPSI

Oleh

Septian Nanda Perdana

NIM 102410101007

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**PENERAPAN METODE ISHIHARA UNTUK
PERANCANGAN APLIKASI TES BUTA WARNA
BERBASIS ANDROID**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Komputer

oleh

Septian Nanda Perdana

NIM 102410101007

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Bapak Kusdi dan Ibu Sri Lestariwati;
2. Kakek Suwarsono dan Nenek Puji Astuti;
3. Adik Sindi Dwi Agustin;
4. Om Didik Suryawan;
5. Semua anggota keluarga yang saya cintai;
6. Guru TK, guru SD, guru SMP, guru SMA;
7. Dosen-dosen Program Studi Sistem Informasi, Bapak Anang Andrianto, Ibu Nelly Oktavia Adiwijaya;
8. Seluruh teman-teman sejak taman kanak-kanak hingga perguruan tinggi;
9. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

Tiada suatu usaha yang besar akan berhasil tanpa dimulai dari usaha yang kecil.¹



¹ Joeniarto, 1967 dalam Mulyono, E. 1998. *Beberapa Permasalahan Implementasi Konvensi Keanekaragaman Hayati dalam Pengelolaan Taman Nasional Meru Betiri*. Tesis magister, tidak dipublikasikan.

PERNYATAAN

Saya yang bertandatangan di bawah ini:

nama : Septian Nanda Perdana

NIM : 102410101007

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Penerapan Metode Ishihara untuk Perancangan Aplikasi Tes Buta Warna Berbasis Android” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggungjawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 13 Juli 2017

Yang menyatakan,

Septian Nanda Perdana

NIM 102410101007

SKRIPSI

**PENERAPAN METODE ISHIHARA UNTUK PERANCANGAN
APLIKASI TES BUTA WARNA BERBASIS ANDROID**

oleh

Septian Nanda Perdana

NIM 102410101007

Pembimbing:

Dosen Pembimbing Utama : Anang Andrianto, S.T., M.T.

Dosen Pembimbing Anggota : Nelly Oktavia Adiwijaya, S.Si., M.T.

PENGESAHAN

Skripsi berjudul “Penerapan Metode Ishihara untuk Perancangan Aplikasi Tes Buta Warna Berbasis Android” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 13 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

Drs. Antonius Cahya P., M.App.Sc., Ph.D.

Nova El Maidah, S.Si., M.Cs.

NIP 196909281993021001

NIP 198411012015042001

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamir, M.Comp.Sc., Ph.D.

NIP 196704201992011001

PENGESAHAN PEMBIMBING

Skripsi berjudul “Penerapan Metode Ishihara untuk Perancangan Aplikasi Tes Buta Warna Berbasis Android” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 13 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Dosen Pembimbing Utama,

Dosen Pembimbing Anggota,

Anang Andrianto, S.T., M.T.

Nelly Oktavia Adiwijaya, S.Si., M.T.

NIP 196906151997021002

NIP 19841024209122008

RINGKASAN

Penerapan Metode Ishihara untuk Perancangan Aplikasi Tes Buta Warna Berbasis Android; Septian Nanda Perdana, 102410101007; 2017; 227 halaman; Program Studi Sistem Informasi Universitas Jember.

Begitu pentingnya mata bagi kehidupan, oleh karena itu kesehatannya wajib dijaga. Namun, seiring berkembangnya waktu muncul berbagai penyakit pada mata. Salah satunya adalah buta warna. Buta warna adalah kelainan pada mata sehingga penderitanya tidak bisa membedakan warna-warna tertentu. Terdapat beberapa cara untuk mendeteksi apakah seseorang mengalami buta warna atau tidak. Salah satunya adalah Metode Ishihara. Tes buta warna Ishihara terdiri dari lembaran-lembaran yang didalamnya terdapat titik-titik dengan pola tertentu. Warna titik-titik tersebut dibuat sedemikianrupa sehingga orang buta warna tidak akan berhasil melihat pola yang ada. Umumnya tes Ishihara memakai buku sebagai medianya. Selanjutnya dengan memanfaatkan perkembangan teknologi saat ini, khususnya dalam bidang *smartphone* android, dalam penelitian ini akan dibangun aplikasi tes buta warna menggunakan Metode Ishihara berbasis android.

Metode pengembangan perangkat lunak yang digunakan adalah Luther. Aplikasi tes buta warna Metode Ishihara berbasis android ini akan diujikan pada dua puluh responden sebagai sampelnya. Kemudian setelah hasil tes aplikasi dari dua puluh responden tersebut terkumpul barulah dilakukan analisis dengan menggunakan metode pengujian validitas dan reliabilitas.

PRAKATA

Puji syukur ke hadirat Allah Swt. atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode Ishihara untuk Perancangan Aplikasi Tes Buta Warna Berbasis Android”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Anang Andrianto, S.T., M.T. selaku dosen pembimbing utama dan Nelly Oktavia Adiwijaya, S.Si., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Prof. Drs. Slamir, M.Comp.Sc., Ph.D. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
3. bapak dan ibu dosen beserta karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Bapak Kusdi dan Ibu Sri Lestariwati yang telah memberikan kasih sayang, doa, semangat, waktu, dan usaha untuk penulis;
5. teman-teman mahasiswa yang sudah memberikan bantuan demi terselesaikannya skripsi ini;
6. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 13 Juli 2017

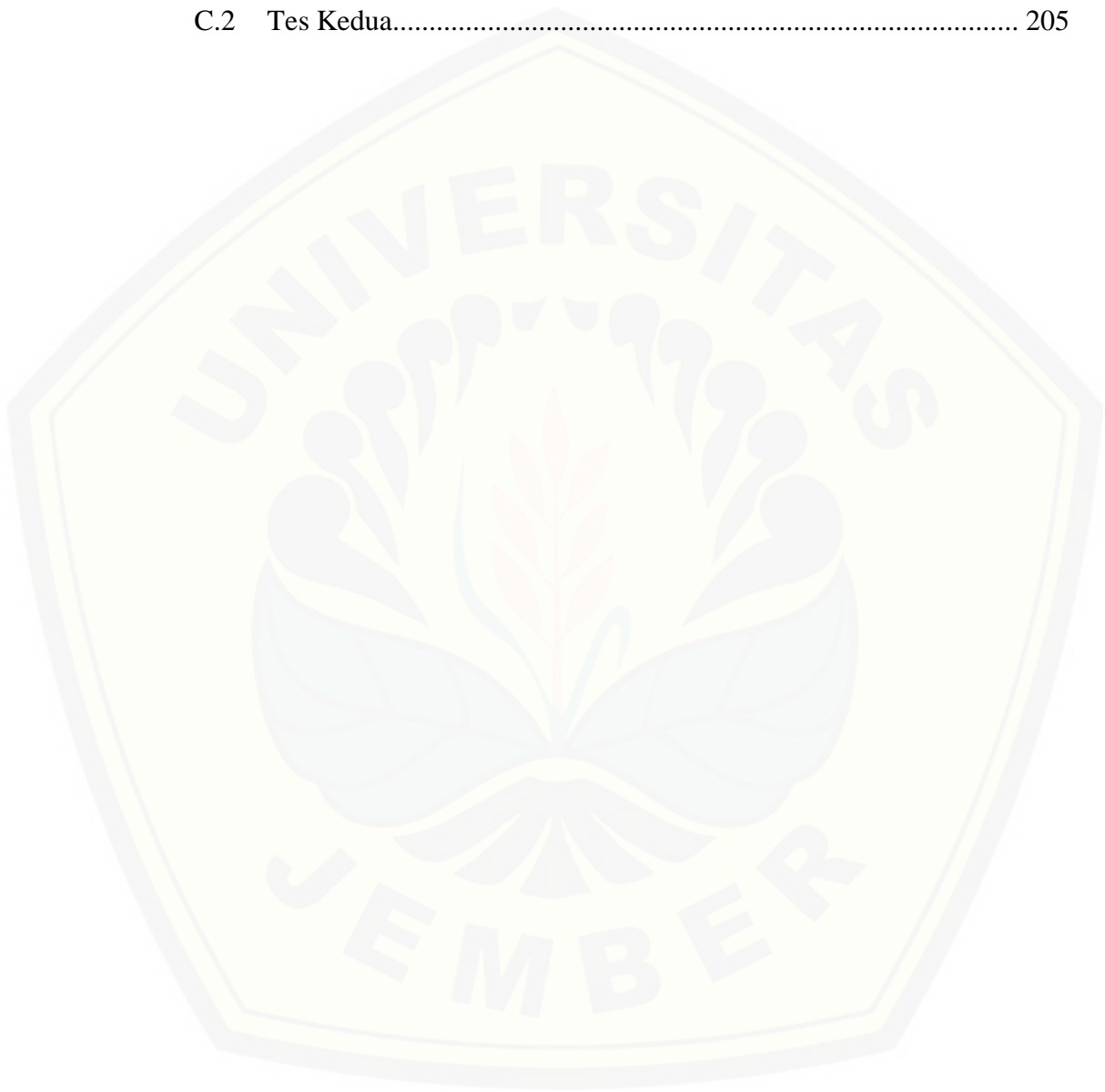
Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSEMBAHAN.....	ii
HALAMAN MOTO.....	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
HALAMAN PENGESAHAN PEMBIMBING	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Penelitian Terdahulu	4
2.2 Pengertian Buta Warna	5
2.3 Pemeriksaan Buta Warna.....	6
2.4 Tes Buta Warna Metode Ishihara	8
2.5 Android	10
2.6 Metode Pengembangan Perangkat Lunak Multimedia Luther	10
2.7 Metode Pengujian Validitas dan Reliabilitas.....	12
BAB 3. METODOLOGI PENELITIAN	16
3.1 Metode Pengembangan Perangkat Lunak.....	16
3.2 Metode Pengujian	17
BAB 4. PENGEMBANGAN APLIKASI.....	20

4.1	Konsep Alur Bisnis dan <i>Flowchart</i> Metode Ishihara	20
4.1.1	Alur Bisnis.....	20
4.1.2	<i>Flowchart</i> Metode Ishihara	21
4.2	<i>Material Collecting</i>	24
4.3	Desain <i>Mockup</i> Aplikasi dan <i>Coding</i>	27
4.4	Pengujian <i>White Box</i>	73
4.5	Pengujian <i>Black Box</i>	75
4.6	Pengujian Validitas dan Reliabilitas	78
4.6.1	Pengujian Validitas.....	78
4.6.2	Pengujian Reliabilitas.....	85
BAB 5.	HASIL DAN PEMBAHASAN	90
5.1	Tinjauan Aplikasi Tes Buta Warna.....	90
5.2	Hasil Pengembangan Aplikasi Tes Buta Warna Metode Ishihara dengan Metode Pengembangan Perangkat Lunak Multimedia Luther	90
5.3	Hasil Pengujian Validitas dan Reliabilitas Terhadap Aplikasi Tes Buta Warna Metode Ishihara.....	103
5.3.1	Validitas.....	103
5.3.2	Reliabilitas.....	103
5.4	Pembahasan Penggunaan Metode Pengembangan Perangkat Lunak Multimedia Luther Terhadap Pengembangan Aplikasi Tes Buta Warna Metode Ishihara Berbasis Android	104
5.5	Pembahasan Penggunaan Metode Pengujian Validitas dan Reliabilitas Pada Hasil Tes Aplikasi Buta Warna Metode Ishihara Berbasis Android	106
BAB 6.	PENUTUP	108
6.1	Kesimpulan	108
6.2	Saran	109
	DAFTAR PUSTAKA	110
	LAMPIRAN.....	112
A.	Penjelasan <i>Plate</i> Ishihara	112
B.	Hasil Tes 20 Responden.....	119

B.1	Tes Pertama.....	119
B.2	Tes Kedua.....	159
C.	Pengumpulan Hasil Tes 20 Responden.....	199
C.1	Tes Pertama.....	199
C.2	Tes Kedua.....	205



DAFTAR TABEL

Tabel 2.1 Nilai-nilai r product moment	14
Tabel 4.1 Pengambilan keputusan tes buta warna Metode Ishihara	23
Tabel 4.2 Data tes buta warna 20 responden.....	78
Tabel 4.3 Perhitungan validitas 1	79
Tabel 4.4 Perhitungan validitas 2.....	80
Tabel 4.5 Perhitungan validitas 3.....	80
Tabel 4.6 Perhitungan validitas 4.....	81
Tabel 4.7 Perhitungan validitas 5.....	82
Tabel 4.8 Perhitungan validitas 6.....	83
Tabel 4.9 Hasil validitas.....	84
Tabel 4.10 Data tes pertama buta warna 20 responden.....	85
Tabel 4.11 Data tes kedua buta warna 20 responden.....	86
Tabel 4.12 Data tabulasi 20 responden tes pertama dan kedua.....	87
Tabel 4.13 Penolong uji reliabilitas	88

DAFTAR GAMBAR

Gambar 2.1 Contoh tes buta warna dengan Metode Ishihara 9

Gambar 2.2 Tahap pengembangan perangkat lunak multimedia Luther 11

Gambar 4.1 *Flowchart* aplikasi tes buta warna Metode Ishihara..... 22

Gambar 4.2 *Plate* Ishihara..... 26

Gambar 4.3 *Splashscreen* aplikasi tes buta warna 26

Gambar 4.4 Logo aplikasi tes buta warna 26

Gambar 4.5 *Splashscreen*..... 27

Gambar 4.6 Desain *mockup* home 28

Gambar 4.7 *Source code* home html 30

Gambar 4.8 *Source code* home ts..... 31

Gambar 4.9 Desain *form input* biodata 31

Gambar 4.10.a *Source code form input* biodata html..... 33

Gambar 4.10.b *Source code form input* biodata html 34

Gambar 4.11 *Source code form input* biodata ts 35

Gambar 4.12 Desain *form input* tanggal 36

Gambar 4.13 *Source code form input* tanggal html 37

Gambar 4.14 Desain *form input* jenis kelamin..... 38

Gambar 4.15 *Source code form input* jenis kelamin html..... 39

Gambar 4.16 Desain *form input* tes *plate* angka 39

Gambar 4.17 *Source code form input* tes *plate* angka html 41

Gambar 4.18 Desain *form input* tes *plate* pola..... 42

Gambar 4.19 *Source code form input* tes *plate* pola html..... 44

Gambar 4.20 *Source code form input* tes *plate* angka dan pola ts 45

Gambar 4.21 Desain hasil tes Ishihara 49

Gambar 4.22 *Source code* hasil tes Ishihara bagian 1 html 51

Gambar 4.23 *Source code* hasil tes Ishihara bagian 2 html 52

Gambar 4.24 *Source code* hasil tes Ishihara bagian 3 html 53

Gambar 4.25 *Source code* hasil tes Ishihara bagian 4 html 54

Gambar 4.26 *Source code* hasil tes Ishihara bagian 1 ts 55

Gambar 4.27 <i>Source code</i> hasil tes Ishihara bagian 2 ts	57
Gambar 4.28 <i>Source code</i> hasil tes Ishihara bagian 3 ts	58
Gambar 4.29 <i>Source code</i> hasil tes Ishihara bagian 3 ts	60
Gambar 4.30 <i>Source code</i> hasil tes Ishihara bagian 4 ts	61
Gambar 4.31 Desain modal hasil tes Ishihara	62
Gambar 4.32 <i>Source code</i> hasil tes Ishihara html	63
Gambar 4.33 <i>Source code</i> modal hasil tes Ishihara ts	64
Gambar 4.34 Desain deskripsi Metode Ishihara	65
Gambar 4.35 <i>Source code</i> deskripsi Metode Ishihara html	67
Gambar 4.36 <i>Source code</i> deskripsi Metode Ishihara ts	67
Gambar 4.37 Desain tentang aplikasi Ishihara	68
Gambar 4.38 <i>Source code</i> tentang aplikasi Ishihara html	70
Gambar 4.39 <i>Source code</i> deskripsi aplikasi Ishihara ts	71
Gambar 4.40 Proses <i>building</i> aplikasi	71
Gambar 4.41 Contoh simulasi <i>real mockup</i> pada <i>device mobile</i>	72
Gambar 4.42 Kode fungsi <i>firstload</i> pada <i>file test.ts</i>	73
Gambar 4.43 Kompleksitas dari kode pada Gambar 4.42	73
Gambar 4.44 Kode fungsi <i>constructor</i> pada <i>file hasil.ts</i>	74
Gambar 4.45 Kompleksitas dari kode pada Gambar 4.44	74
Gambar 4.46 Pengujian <i>black box</i> aplikasi tes buta warna Ishihara bagian 1	75
Gambar 4.47 Pengujian <i>black box</i> aplikasi tes buta warna Ishihara bagian 2	76
Gambar 4.48 Pengujian <i>black box</i> informasi Metode Ishihara	77
Gambar 4.49 Pengujian <i>black box</i> informasi tentang aplikasi	77
Gambar 5.1 <i>Splashscreen</i>	91
Gambar 5.2 Home / menu utama	92
Gambar 5.3 <i>Form</i> biodata	93
Gambar 5.4 Dialog <i>pop up</i> tanggal lahir	94
Gambar 5.5 Dialog <i>pop up</i> jenis kelamin	95
Gambar 5.6 Soal bentuk angka	96
Gambar 5.7 Soal bentuk pola	97
Gambar 5.8 Hasil detail tes buta warna bagian pertama	98

Gambar 5.9 Hasil detail tes buta warna bagian kedua	99
Gambar 5.10 Hasil detail tes buta warna bagian kedua	100
Gambar 5.11 Halaman informasi Metode Ishihara	101
Gambar 5.12 Halaman informasi Metode Ishihara	102



BAB 1. PENDAHULUAN

Bab ini akan menjelaskan tentang latar belakang pemilihan topik dan perumusan masalah yang akan dijadikan bahasan pada penelitian ini.

1.1 Latar Belakang

Begitu pentingnya mata bagi manusia untuk mengenali lingkungan di sekitarnya. Oleh karena itu manusia harus menjaga kesehatan mata dengan baik dan cermat. Namun sangat disayangkan, seiring dengan gaya hidup yang mengikuti perkembangan zaman, banyak manusia yang tidak menjaga kesehatan matanya dengan baik, bahkan seolah tidak peduli dengan kesehatan mata mereka. Hal ini tentunya menimbulkan berbagai penyakit pada mata.

Adapun diantaranya beberapa penyakit mata, antara lain katarak, rabun jauh, rabun dekat, rabun senja, *presbyopia* (mata tua), buta warna, *pterygium* (pertumbuhan selaput tipis di konjungtiva), *pinguecula* (benjolan kecil di ujung bola mata dekat dengan kornea dan berwarna kuning), *astigmatism* (mata silindris), dan kebutaan (Ilyas, 2012).

Kemampuan mata dalam membedakan warna merupakan salah satu fungsi penglihatan yang penting. Namun, tidak semua manusia dianugerahi kemampuan penglihatan warna secara normal. Salah satunya adalah penderita buta warna. Buta warna merupakan penyakit keturunan yang terekspresi pada pria, tetapi tidak pada wanita (Ganong, 2003). Kelainan buta warna dapat mengganggu aktivitas penderitanya sehari-hari. Contohnya dalam membedakan warna lampu lalu lintas, warna pakaian, warna makanan dan minuman, warna gambar ketika menonton film.

Pada dunia pendidikan pun demikian, beberapa fakultas di Universitas Jember mewajibkan calon mahasiswanya untuk memiliki penglihatan mata normal. Tentunya hal ini akan menjadi sebuah masalah untuk calon mahasiswa baru yang memiliki penglihatan buta warna. Selama ini proses pengecekan kemampuan mata calon mahasiswa baru dalam melihat warna dilakukan melalui *Unej Medical Centre (UMC)* dan menggunakan alat tes kertas gambar Ishihara.

Tingkat mobilitas dan kesibukan saat ini membuat rendahnya kesadaran dan kurangnya perhatian masyarakat mengenai kelainan buta warna serta untuk melakukan tes buta warna sejak dini. Pemeriksaan secara langsung untuk mengetahui kelainan buta warna yang ditangani dokter spesialis mata sangat jarang dilakukan. Biaya yang dikeluarkan juga relatif mahal. Oleh karena itu, dibutuhkan adanya sebuah inovasi baru dalam melakukan tes buta warna yang dapat digunakan dalam membantu proses pemeriksaan sejak dini oleh seseorang yang dapat dilakukan kapan saja, di mana saja, dan dengan biaya yang relatif murah.

Ada beberapa metode yang dapat digunakan untuk tes buta warna. Salah satunya adalah Metode Ishihara. Metode Ishihara masih menjadi salah satu pilihan utama di hampir semua negara untuk mengidentifikasi seseorang yang menderita buta warna. Metode Ishihara secara relatif dapat dipercaya dalam membedakan antara defisit warna merah dan defisit warna hijau. Dengan memanfaatkan perkembangan teknologi yang ada sekarang ini, untuk mengidentifikasi buta warna diharapkan akan lebih mudah dilakukan. Salah satu teknologi yang sekarang berkembang pesat di masyarakat adalah *smartphone*. Saat ini *smartphone* banyak menggunakan sistem operasi android.

Pada penelitian ini akan dibangun sebuah aplikasi tes buta warna menggunakan Metode Ishihara dengan memanfaatkan sistem operasi android sebagai basisnya, sehingga hasil dari rancangan aplikasi ini diharapkan mampu digunakan sebagai media pendeteksian buta warna secara efisien.

1.2 Rumusan Masalah

Menindaklanjuti seperti yang dijelaskan sebelumnya, ada beberapa rumusan masalah yang dapat dipaparkan, antara lain:

- a. Bagaimana merancang aplikasi tes buta warna dengan Metode Ishihara yang berbasis android.
- b. Apakah aplikasi tes buta warna berbasis android dengan Metode Ishihara mampu mendeteksi buta warna.

1.3 Tujuan

Penelitian ini memiliki beberapa tujuan yang diantaranya adalah:

- a. Merancang aplikasi tes buta warna dengan Metode Ishihara yang berbasis android.
- b. Untuk mendeteksi buta warna

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang dijadikan sebagai pedoman dalam melakukan penelitian ini, diantaranya sebagai berikut.

- a. Metode yang digunakan peneliti dalam mengembangkan aplikasi adalah metode pengembangan perangkat lunak multimedia Luther.
- b. Proses uji coba pengukuran aplikasi menggunakan metode validitas dan reliabilitas.
- c. Penelitian ini ditujukan untuk mengetahui apakah seseorang memiliki mata normal, buta warna parsial, dan buta warna total dengan menggunakan Metode Ishihara.
- d. Aplikasi yang akan dikembangkan adalah berbasis android.

BAB 2. TINJAUAN PUSTAKA

Tinjauan pustaka merupakan landasan berpikir bagi penulis untuk mendapatkan dasar-dasar ilmiah yang diperlukan sebagai penetapan dalam melakukan suatu penelitian.

2.1 Penelitian Terdahulu

Penelitian terdahulu mengenai tes buta warna telah dilakukan oleh Widianingsih *et al.* (2010) dalam penelitiannya yang berjudul Aplikasi Tes Buta Warna Dengan Metode Ishihara Berbasis Komputer. Dalam penelitian ini menghasilkan suatu aplikasi tes buta warna berbasis komputer yang digunakan untuk tes buta warna di Poltabes Samarinda. Hasil keluaran berupa *print out* surat keterangan kesehatan dengan menyebutkan hasil tes buta warna, yaitu normal, buta warna parsial, atau buta warna total.

Penelitian lain telah dilakukan oleh Murti dan Santi (2011) dalam penelitiannya yang berjudul Aplikasi Pendiagnosa Kebutaan Warna dengan Menggunakan Pemrograman Borland Delphi. Penelitian ini menghasilkan alat bantu tes 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 tes Ishihara yang dilakukan oleh seorang dokter mata.

Penelitian yang dilakukan Agusta *et al.* (2012), bertujuan memudahkan pengguna, dokter, maupun pelayan kesehatan dalam melakukan tes buta warna secara massal, dengan membuat suatu program berbasis visual basic 6.0. Metode yang digunakan dengan membandingkan hasil tes buta warna yang dilakukan secara konvensional menggunakan instrumen pengujian tes buta warna otomatis menggunakan *software* berbasis visual basic dengan perangkat bantuan *notebook* dan *tablet* (Tab). Kesimpulan instrumen pengujian tes buta warna otomatis dapat berfungsi dengan baik dalam melakukan tes buta warna secara otomatis, mudah digunakan, dan memenuhi syarat untuk mengajukan hak atas kekayaan intelektual.

Penelitian lain mengenai buta warna juga dilakukan oleh Emerson *et al.* (2010) yang berjudul *Prevalence of Color-Vision Deficiency Among Male High-School Students*, menyatakan bahwa rata-rata penderita buta warna adalah siswa sekolah menengah atas yang berjenis kelamin laki-laki. Hal tersebut ditunjukkan dengan persentase 5,17% atau 65 siswa dari 1258 siswa.

Penelitian terdahulu yang relevan merupakan hal yang cukup penting untuk dijadikan data pendukung dalam melaksanakan penelitian karena didalamnya terdapat acuan yang merupakan teori atau hasil dari berbagai penelitian. Pada penelitian ini khususnya adalah Metode Ishihara yang digunakan dalam mengambil keputusan. Namun, penelitian ini dalam pengembangan aplikasi tes buta warnanya digunakan metode pengembangan perangkat lunak multimedia Luther, serta pada analisisnya menggunakan metode pengukuran validitas dan reliabilitas dan *platform* yang digunakan adalah android.

2.2 Pengertian Buta Warna

Buta warna adalah ketidakmampuan seseorang untuk membedakan warna tertentu. Pada retina mata normal terdapat dua jenis sel yang sensitive terhadap cahaya. Sel batang (*rod cell*) yang aktif pada cahaya rendah, lalu sel kerucut (*cone cell*) yang aktif pada cahaya yang memiliki intensitas tinggi (terang). Sel kerucut inilah yang membuat manusia dapat melihat warna-warna dan membedakannya (Yanuarita, 2012).

Buta warna merupakan penyakit keturunan yang terekspresi pada pria, tetapi tidak pada wanita. Wanita secara genetik 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 atau *colour vision deficiency* (Ganong, 2003).

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

Nugroho (2013) menyatakan bahwa klasifikasi buta warna, yaitu:

a. *Trichromacy*

Trichromacy adalah gangguan penglihatan warna yang dapat disebabkan oleh faktor keturunan atau dapat diakibatkan karena adanya kerusakan pada sel mata setelah dewasa. Ada tiga macam *trichromacy*, antara lain:

- 1) *Protanomaly*, penderita akan sulit mengenali warna merah
- 2) *Deuteranomaly*, penderita akan sulit mengenali warna hijau
- 3) *Tritanomaly*, penderita akan sulit mengenali warna biru

b. *Dichromacy*

Dichromacy adalah keadaan di mana salah satu dari tiga sel kerucut tidak ada atau tidak berfungsi. Ada tiga klasifikasi untuk *dichromacy*, yaitu:

- 1) *Protanopia* (buta warna merah), sel kerucut warna merah tidak berfungsi dengan baik
- 2) *Deuteranopia* (buta warna hijau), sel kerucut tidak peka terhadap warna hijau
- 3) *Tritanopia* (buta warna biru), sel kerucut untuk warna biru tidak berfungsi dengan baik

c. *Monochromacy*

Monochromacy adalah kondisi retina mata yang mengalami kerusakan total dalam merespon warna. *Monochromacy* ditandai dengan hilangnya atau berkurangnya semua penglihatan warna, sehingga yang terlihat hanya putih dan hitam.

2.3 Pemeriksaan Buta Warna

Rokhim (2015) mengemukakan untuk melakukan tes buta warna terdapat beberapa metode yang sering dijumpai, antara lain:

a. *Pseudoisochromatic Plate Test*

Tes ini digunakan terutama untuk memeriksa adanya kelainan melihat warna pada anak-anak usia 3 – 6 tahun. Kebanyakan anak-anak usia tiga tahun sudah dapat menyebutkan macam-macam bidang bangunan yang sederhana secara mudah

seperti lingkaran, kotak, bintang, dan segitiga. Jika dengan pemeriksaan di atas seorang anak tidak dapat menyebutkan gambar tersebut, maka digunakan gambar dengan warna hitam putih dan mencocokkan gambar dengan berbagai macam warna yang berbeda. Jadi seorang anak dituntun untuk mengenal gambar apa yang tertera.

b. *Color Pencil Discrimination*

Tes ini dapat digunakan untuk memeriksa ada tidaknya defisiensi melihat warna terhadap anak-anak yang lebih besar dan sudah bersekolah. Hal ini terlihat saat seorang anak sulit membedakan macam-macam warna dan pensil warna yang begitu banyak.

c. *Holmgren-Thompson Wool Test for Color Blindness*

Tes ini dilakukan dengan cara:

- 1) Empat puluh gulungan benang dikumpulkan pada suatu tempat. Label warna tertutup.
- 2) Pilih sepuluh gulungan benang yang warnanya paling mendekati pola A warna hijau muda.
- 3) Gulungan benang yang tersisa, pilih lima buah yang paling mendekati pola C warna merah.
- 4) Kemudian, sisa dua puluh lima gulungan benang, pilih lima buah yang paling mendekati pola B warna biru.
- 5) Catat label angka dari gulungan benang yang dipilih dengan urutan mulai dari warna yang paling mendekati sampai yang kurang mendekati warna pola gulungan benang.

d. *Anomaloscope*

Digunakan untuk menemukan defisiensi sebagian warna. Selain itu, juga digunakan untuk mendiagnosis kelainan trikromat. Pemeriksaan ini dapat menentukan dengan lebih pasti adanya gangguan penglihatan warna merah dan hijau.

e. *D-15 Farnsworth Test*

Diskriminasi warna ditentukan oleh tiga faktor, yaitu *hue*, *saturation*, dan derajat terang/luminasi (*brightness*). Tes ini menggunakan kepingan-kepingan

berwarna dengan *hue* yang berbeda namun *saturation* dan derajat terang yang sama. Terdapat dua versi tes ini, yaitu D-15 yang terdiri atas keping warna, dan FM—100 yang terdiri dan 85 keping warna. Prinsip tes ini sama, yaitu pasien diminta mengurutkan kepingan-kepingan warna sesuai gradasi warna, dimulai dari keping dengan warna paling mendekati keping referensi (keping yang terfiksasi pada kotak wadah keping). Pada bagian bawah keping terdapat nomor yang merupakan urutan berapa seharusnya keping tersebut disusun.

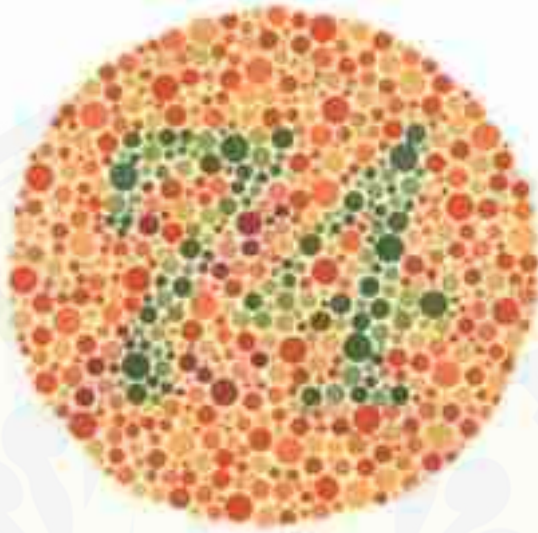
f. Metode Ishihara

Metode Ishihara digunakan untuk mendeteksi gangguan persepsi warna. Berupa tabel warna khusus terdiri dari gambar-gambar pseudoisokromatik yang disusun oleh titik-titik dengan kepadatan warna berbeda yang dapat dilihat dengan mata normal, tapi tidak bisa dilihat oleh mata yang mengalami defisiensi sebagian warna. Metode ini sering digunakan untuk mendiagnosis defisiensi warna merah-hijau. Setiap kartu memiliki bermacam-macam gambar dan latar belakang dengan warna yang berkombinasi. Tes Ishihara adalah tes paling populer dan efektif untuk penderita buta warna.

2.4 Tes Buta Warna Metode Ishihara

Metode Ishihara dikembangkan oleh Dr. Shinobu Ishihara pada tahun 1917. Dia merupakan seorang profesor yang berasal dari University of Tokyo. Tes buta warna Ishihara terdiri dari lembaran (*plate*) yang didalamnya terdapat titik-titik dengan berbagai warna dan ukuran (Yanuarita, 2012). 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. Gambar diletakkan di tempat dengan pencahayaan yang baik dan pasien diminta untuk mengidentifikasi angka atau mengikuti jejak garis yang terdapat pada titik-titik warna berbentuk lingkaran tersebut. Hasil tes seseorang akan dibandingkan dengan kunci jawaban, selanjutnya diidentifikasi dan diklasifikasikan untuk menentukan tingkat buta warnanya.

Guyton & Hall (1997) mengemukakan Metode Ishihara, yaitu metode yang dapat dipakai untuk menentukan dengan cepat suatu kelainan buta warna berdasarkan pada penggunaan kartu bertitik seperti Gambar 2.1.



Gambar 2.1 Contoh tes buta warna dengan Metode Ishihara
(Sumber: Ishihara:1972)

Pada Gambar 2.1, orang normal akan membaca angka 74, sedangkan penderita buta warna merah-hijau akan membaca angka 21. Mereka yang buta warna total tidak dapat membaca sama sekali. Beberapa hal yang perlu diperhatikan dalam pemeriksaan tes buta warna menggunakan Metode Ishihara adalah ruangan pemeriksaan harus cukup pencahayaannya, lama pengamatan untuk membaca masing-masing *plate* maksimum sepuluh detik. Seseorang dinyatakan buta warna total apabila pada *plate* satu salah dan jawaban lain diabaikan. Seseorang dinyatakan buta warna parsial apabila *plate* satu benar, *plate* dua sampai *plate* enam belas ada salah lebih dari tiga, atau jika *plate* satu benar, *plate* dua puluh dua sampai *plate* dua puluh empat jawaban hanya benar pada salah satu *plate*, atau jika *plate* satu benar, *plate* delapan belas sampai *plate* dua puluh satu salah. Seseorang dikatakan normal apabila *plate* satu sampai *plate* tujuh belas benar, atau *plate* satu harus benar dan lebih dari tiga belas *plate* dijawab benar, *plate* dua puluh dua sampai *plate* dua puluh empat benar atau dua *plate* benar.

2.5 Android

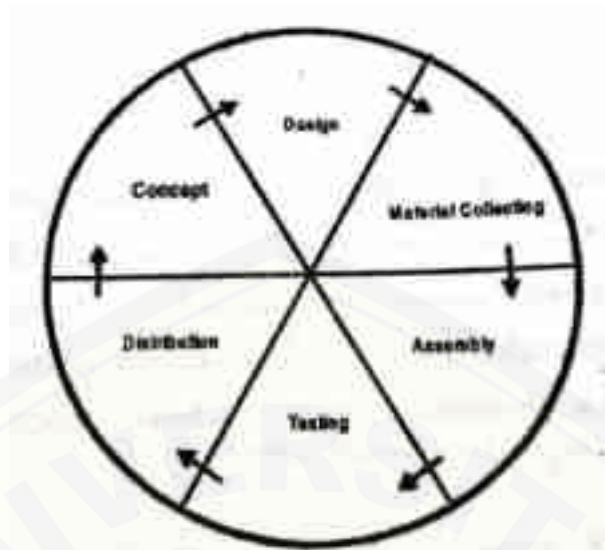
Safaat (2015) mengemukakan android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari android dilisensikan di bawah GNU, *General Public License Version 2* (GPLv2), yang sering dikenal dengan istilah “*copyleft*” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah *terms*.

Android didistribusikan di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen *handset* khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan.

Juhara (2016) menyatakan aplikasi android dibangun menggunakan Java. Meskipun memiliki *Java Virtual Machine* bernama *Dalvik* VM yang berjalan di atas *kernel* linux, tetapi android bukan implementasi *Java* ME. *Dalvik* sendiri sudah mengalami optimasi dan diubah sedemikian rupa sehingga kode *intermediate* yang dihasilkan *compiler java* buatan Oracle tidak akan dapat dijalankan. Optimasi ini bertujuan supaya *Dalvik* berjalan optimal pada perangkat *embedded* yang minim daya komputasi dan listrik.

2.6 Metode Pengembangan Perangkat Lunak Multimedia Luther

Luther (1994) mendefinisikan langkah-langkah pengembangan perangkat lunak multimedia dengan enam tahap, di mana setiap tahapannya tidak harus berurutan, tetapi dapat dikerjakan secara paralel dengan tahapan perencanaan (*concept* dan *design*) harus dimulai dulu. Adapun tahapan dari metode pengembangan perangkat lunak multimedia Luther seperti Gambar 2.2.



Gambar 2.2 Tahap pengembangan perangkat lunak multimedia Luther
(Sumber: Sutopo:2003)

a. *Concept*

Tujuan konsep pengembangan perangkat lunak multimedia didefinisikan pada tahapan ini. Melingkupi identifikasi pengguna aplikasi, jenis aplikasi (presentasi, interaktif), tujuan aplikasi (pemberitahuan, hiburan, pengajaran) dan hal-hal umum. Aturan dasar untuk tahapan *design* juga sudah diatur dan ditentukan pada tahapan ini, seperti gaya, ukuran aplikasi, *platform* yang digunakan, dan lainnya. Luaran dari tahapan ini semacam dokumen laporan yang merupakan deskripsi dari perangkat lunak multimedia yang dikembangkan.

b. *Design*

Tujuan tahapan ini adalah menentukan secara detil arsitektur, gaya, dan semua material yang akan digunakan pada perangkat lunak multimedia yang akan dikembangkan. Tahapan ini diharapkan cukup detil sehingga tahapan-tahapan berikutnya, yaitu *material collecting* dan *assembly* dapat dilakukan tanpa perlu keputusan-keputusan alternatif lain. Tetapi, tidak semuanya berjalan sesuai yang diharapkan dan sudah biasa ada material baru atau bagian dari aplikasi yang ditambahkan, dihapus, atau diubah di tahapan yang paling akhir pada pengembangan ini.

Tugas lain pada tahapan ini adalah memutuskan semua *material content*, termasuk *text* dan *file database*, *audio*, *video*, dan gambar. Tujuannya untuk membuat daftar material yang akan dicari atau dibuat sehingga semuanya akan dapat dikumpulkan pada tahapan berikutnya.

c. *Material Collecting*

Material collecting adalah tahap di mana pengumpulan bahan yang sesuai dengan kebutuhan yang dilakukan. Tahap ini dapat dikerjakan paralel dengan tahap *assembly*. Pada beberapa kasus, tahap *material collecting* dan tahap *assembly* akan dikerjakan secara linear tidak paralel.

d. *Assembly*

Tahap *assembly* (pembuatan) adalah tahap di mana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap *design*.

e. *Testing*

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian *alpha (alpha test)*, di mana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Sebenarnya yang terpenting dalam *testing* adalah untuk melihat apakah aplikasi yang dibangun berjalan dengan baik di lingkungan yang sebenarnya untuk memastikan semuanya dapat berjalan dengan baik. Selain itu, juga diperlukan *testing* kepada pengguna untuk meyakinkan bahwa pengguna dapat menggunakan aplikasi dengan baik.

f. *Distribution*

Tahapan di mana aplikasi disimpan dalam suatu media penyimpanan agar bisa digunakan pengguna.

2.7 Metode Pengujian Validitas dan Reliabilitas

Pengujian ini dilakukan bertujuan untuk mengetahui seberapa valid dan reliabilitasnya pada aplikasi yang telah dikembangkan terhadap penentuan pengambilan keputusan apakah seseorang mengalami buta warna atau tidak.

a. Validitas

Validitas adalah tingkat keandalan dan kesahihan alat ukur yang digunakan. Instrumen dikatakan valid berarti menunjukkan alat ukur yang dipergunakan untuk mendapatkan data itu valid atau dapat digunakan untuk mengukur apa yang seharusnya diukur (Sugiyono, 2004). Instrumen yang valid merupakan instrumen yang benar-benar tepat untuk mengukur apa yang hendak diukur. Atau bisa dikatakan Validitas (*Validity*) yaitu sejauh mana suatu alat ukur tepat dalam mengukur suatu data, dengan kata lain apakah alat ukur yang dipakai memang mengukur sesuatu yang ingin diukur.

Ada beberapa kriteria yang dapat digunakan untuk mengetahui instrumen yang digunakan sudah tepat untuk mengukur apa yang ingin diukur, yaitu:

- 1) Jika koefisien korelasi *product moment* melebihi 0,3.
- 2) Jika koefisien korelasi *product moment* $> r_{tabel} (\alpha ; n - 2)$, $n =$ jumlah sampel.
- 3) Nilai sig. $\leq \alpha$

Rumus yang digunakan untuk uji validitas dengan teknik korelasi *product moment* (r_{hitung}), yaitu:

$$r_{hitung} = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[n(\sum X^2) - (\sum X)^2][n(\sum Y^2) - (\sum Y)^2]}} \dots\dots\dots \text{(Persamaan 1)}$$

di mana:

$n =$ jumlah responden

$X =$ skor variabel (jawaban responden)

$Y =$ skor total dari variabel (jawaban responden)

b. Reliabilitas

Reliabilitas adalah ukuran yang menunjukkan bahwa alat ukur yang digunakan dalam penelitian mempunyai keandalan sebagai alat ukur, diantaranya diukur melalui konsistensi hasil pengukuran dari waktu ke waktu jika fenomena yang diukur tidak berubah (Zulganef, 2006).

Pada umumnya teknik korelasi yang digunakan untuk menguji reliabilitas menggunakan *product moment*. Bila koefisien korelasi (r_{hitung}) lebih besar (r_{tabel}), maka pengukuran pertama dan kedua konsisten, sehingga instrumen tersebut dinyatakan *reliable*, akan tetapi bila (r_{hitung}) lebih kecil (r_{tabel}), maka

hasil pengukuran pertama dan kedua tidak konsisten, sehingga dapat disimpulkan instrumen yang digunakan tidak *reliable*.

Tahapan perhitungan uji reliabilitasnya, yaitu:

- 1) Membuat hipotesis dalam bentuk kalimat

Ho: Pengukuran pertama dan pengukuran kedua tidak konsisten (tidak *reliable*)

Ha: Pengukuran pertama dan pengukuran kedua konsisten (*reliable*)

- 2) Menentukan taraf signifikan (α)

Taraf signifikan adalah seberapa besar menerima hipotesis yang salah

- 3) Kaidah pengujian

Jika $r_{hitung} \leq r_{tabel}$ maka Ho diterima

Jika $r_{hitung} > r_{tabel}$ maka Ho ditolak

- 4) Menghitung r_{hitung} dan r_{tabel}

- a) Menghitung r_{hitung} mengacu pada Persamaan 1

- b) Menentukan nilai koefisien korelasi (r_{tabel})

Nilai r_{tabel} dapat dilihat di tabel *Product Moment* dengan ketentuan $r_{(a,n-2)}$

- 5) Membandingkan r_{tabel} dan r_{hitung}

Tujuan membandingkan r_{hitung} dan r_{tabel} adalah untuk mengetahui hipotesis mana yang akan diterima berdasarkan kaidah pengujian

- 6) Membuat keputusan

Menerima atau menolak Ho

Untuk menentukan koefisien korelasi (r_{tabel}) pada metode validitas dan reliabilitas dapat didasarkan pada Tabel 2.1, kemudian disesuaikan dengan jumlah responden (N) dan taraf signifikan (%).

Tabel 2.1 Nilai-nilai r product moment

N	Taraf Signifikan		N	Taraf Signifikan		N	Taraf Signifikan	
	5%	1%		5%	1%		5%	1%
3	0,997	0,999	27	0,381	0,487	55	0,266	0,345
4	0,950	0,990	28	0,374	0,478	60	0,254	0,330
5	0,878	0,959	29	0,367	0,470	65	0,244	0,317

6	0,811	0,917	30	0,361	0,463	70	0,235	0,306
7	0,754	0,874	31	0,355	0,456	75	0,227	0,296
8	0,707	0,834	32	0,349	0,449	80	0,220	0,286
9	0,666	0,798	33	0,344	0,442	85	0,213	0,278
10	0,632	0,765	34	0,339	0,436	90	0,207	0,270
11	0,602	0,735	35	0,334	0,430	95	0,202	0,263
12	0,576	0,708	36	0,329	0,424	100	0,195	0,256
13	0,553	0,684	37	0,325	0,418	125	0,176	0,230
14	0,532	0,661	38	0,320	0,413	150	0,159	0,210
15	0,514	0,641	39	0,316	0,408	175	0,148	0,194
16	0,497	0,623	40	0,312	0,403	200	0,138	0,181
17	0,482	0,606	41	0,308	0,398	300	0,113	0,148
18	0,468	0,590	42	0,304	0,393	400	0,098	0,128
19	0,456	0,575	43	0,301	0,389	500	0,088	0,115
20	0,444	0,561	44	0,297	0,384	600	0,080	0,105
21	0,433	0,549	45	0,294	0,380	700	0,074	0,097
22	0,423	0,537	46	0,291	0,376	800	0,070	0,091
23	0,413	0,526	47	0,288	0,372	900	0,065	0,086
24	0,404	0,515	48	0,284	0,368	1000	0,062	0,081
25	0,396	0,505	49	0,281	0,364			
26	0,388	0,496	50	0,279	0,361			

BAB 3. METODOLOGI PENELITIAN

Pada bagian ini akan diuraikan mengenai metode pengembangan perangkat lunak serta metode pengujian validitas dan reliabilitas yang digunakan dalam penelitian.

3.1 Metode Pengembangan Perangkat Lunak

Pada penelitian ini pengembangan aplikasi merujuk pada metode pengembangan perangkat lunak multimedia Luther yang dijelaskan pada subbab 2.6. Untuk tahapan pengembangannya didasarkan pada Gambar 2.2, kemudian dijabarkan sebagai berikut.

a. *Concept*

Tujuan konsep pengembangan aplikasi tes buta warna berbasis android yang menggunakan Metode Ishihara ini adalah untuk mendeteksi buta warna. Untuk pengguna aplikasinya sendiri adalah masyarakat umum yang berusia minimal 15 tahun dan sudah mengenal tentang *smartphone*. Jenis aplikasi yang dikembangkan adalah interaktif, karena pengguna dan aplikasi saling melakukan interaksi atau terdapat respon antara aplikasi dan pengguna. Aplikasi ini dikategorikan sebagai aplikasi kesehatan karena dapat menentukan seseorang mengalami buta warna atau tidak. Untuk spesifikasi android yang digunakan dalam menjalankan aplikasi ini minimum android versi 4.4+, ukuran aplikasi tergantung dari *device* yang digunakan.

b. *Design*

Tahap ini membuat desain antarmuka dalam bentuk *mockup* dan membuat alur Metode Ishihara yang diterapkan di aplikasi dalam bentuk *flowchart*.

c. *Material Collecting*

Pada *material collecting* dilakukan pengumpulan 24 *Plate* Ishihara, Adobe Photoshop CC 2017 untuk membuat *icon* dan *splashscreen*, CorelDraw 2017 untuk membuat desain *mockup*, Microsoft Visio Professional 2016 untuk membuat desain *flowchart*, Java SE Development Kit sebagai ruang lingkup untuk pengembangan aplikasi menggunakan bahasa pemrograman Java, Android SDK untuk

membangun aplikasi android, Cordova sebagai HTML5 *hybrid platform*, Ionic untuk framework angular 2, dan Webstorm untuk menulis kode aplikasi.

d. *Assembly*

Tahap *assembly* ini merangkai seluruh objek seperti *Plate Ishihara*, *icon*, *splashscreen*, penulisan *source code*, dan seluruh objek yang digunakan dalam aplikasi berdasarkan pada tahap *design* dan *material collecting*.

e. *Testing*

Pada tahapan pengetesan dilakukan pengujian terhadap aplikasi yang telah dibangun apakah terjadi kesalahan atau tidak menggunakan metode pengetesan *blackbox* dan *whitebox*. *White box testing* adalah pengujian aplikasi dengan melihat isi atau kode aplikasi. Dalam penelitian ini penulis menggunakan *cyclomatic complexity* untuk mengetahui suatu kompleksitas dari aplikasi yang dirancang. *Black box testing* menguji perangkat lunak dari segi spesifikasi fungsional, tanpa menguji desain dan kode aplikasi. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak, berjalan sesuai dengan spesifikasi yang dibutuhkan.

f. *Distribution*

Tahapan di mana aplikasi didistribusikan melalui Play Store agar dapat diakses masyarakat.

3.2 Metode Pengujian

Metode yang digunakan dalam pengujian menggunakan validitas dan reliabilitas sebagaimana yang telah dijelaskan pada subbab 2.7. Pengujian ini dilakukan setelah semua data hasil tes buta warna yang didapat dari aplikasi yang digunakan oleh responden terkumpul. Terdapat empat puluh anggota dalam sebuah populasi masyarakat yang akan diujikan untuk menggunakan aplikasi. Berdasarkan populasi tersebut digunakan teknik pengambilan sampel, yaitu *simple random sampling*. Dikatakan *simple* (sederhana) karena pengambilan anggota sampel dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi itu. Cara ini dilakukan karena anggota populasi dianggap homogen, yakni memiliki usia lima belas tahun ke atas. Pengambilan sampel secara acak dilakukan

dengan undian, maka setiap anggota populasi diberi nomor terlebih dahulu, sesuai dengan jumlah anggota populasi. Menurut Roscoe (dalam Sugiyono, 2016:90), untuk penelitian eksperimen yang sederhana, yang menggunakan kelompok eksperimen dan kelompok kontrol, maka jumlah anggota sampel antara 10 s/d 20. Pada penelitian ini menggunakan dua puluh sampel. Karena teknik pengambilan sampel adalah random, maka setiap anggota populasi mempunyai peluang sama untuk dipilih menjadi anggota sampel. Untuk penelitian ini peluang setiap anggota populasi adalah $2/40$. Cara pengambilannya adalah dua anggota populasi mengambil secara bersamaan nomor undiannya, kemudian diteruskan dua anggota lagi untuk mengambil, dan begitu seterusnya. Hanya ada dua puluh nomor undian yang sudah ditandai. Jika anggota populasi tersebut mendapat nomor undian yang sudah ditandai, artinya dia sudah terpilih menjadi sampel. Setelah terkumpul dua puluh sampel atau responden, maka aplikasi siap untuk diujikan. Hasil dari semua tes sifatnya masih acak. Kemudian diurutkan sesuai *plate* Ishihara. Untuk jawaban benar diwakilkan dengan nilai 1, sedangkan untuk jawaban salah diwakilkan dengan nilai 0. Selanjutnya dilakukan pengujian validitas dan reliabilitas.

a. Validitas

Ada beberapa kriteria yang dapat digunakan untuk mengetahui aplikasi yang digunakan sudah tepat untuk mengukur apa yang ingin diukur, yaitu:

- 1) Jika koefisien korelasi *product moment* melebihi 0,3.
- 2) Jika koefisien korelasi *product moment* $> r_{tabel} (\alpha ; n - 2)$, $n =$ jumlah sampel.
- 3) Nilai sig. $\leq \alpha$

Rumus yang digunakan untuk uji validitas dengan teknik korelasi *product moment* (r_{hitung}), yaitu menggunakan Persamaan 1.

b. Reliabilitas

Tahapan perhitungan uji reliabilitasnya, yaitu:

- 1) Membuat hipotesis dalam bentuk kalimat

Ho: Pengukuran pertama dan pengukuran kedua tidak konsisten (tidak *reliable*)

Ha: Pengukuran pertama dan pengukuran kedua konsisten (*reliable*)

- 2) Menentukan taraf signifikan (α)

Taraf signifikan adalah seberapa besar menerima hipotesis yang salah

3) Kaidah pengujian

Jika $r_{hitung} \leq r_{tabel}$ maka H_0 diterima

Jika $r_{hitung} > r_{tabel}$ maka H_0 ditolak

4) Menghitung r_{hitung} dan r_{tabel}

a) Menghitung r_{hitung} mengacu pada Persamaan 1

b) Menentukan nilai koefisien korelasi (r_{tabel})

Nilai r_{tabel} dapat dilihat di Tabel 2.1 dengan ketentuan $r_{(a,n-2)}$

5) Membandingkan r_{tabel} dan r_{hitung}

Tujuan membandingkan r_{hitung} dan r_{tabel} adalah untuk mengetahui hipotesis mana yang akan diterima berdasarkan kaidah pengujian

6) Membuat keputusan

Menerima atau menolak H_0

BAB 4. PENGEMBANGAN APLIKASI

Pada bab ini akan diuraikan mengenai pengembangan aplikasi tes buta warna dengan Metode Ishihara yang merupakan pengembangan aplikasi berbasis multimedia, oleh karena itu peneliti menggunakan metode pengembangan perangkat lunak multimedia Luther. Pengembangan aplikasi ini akan melewati beberapa tahapan mulai dari konsep, desain, pengkodean dan pembangunan serta distribusi aplikasi.

4.1 Konsep Alur Bisnis dan *Flowchart* Metode Ishihara

Pada subbab ini akan dijelaskan mengenai konsep dasar dari alur atau proses interaksi antara pengguna dengan aplikasi. Konsep ini mempunyai tujuan akhir memberikan hasil akhir yakni status mata normal, mata buta warna parsial dan mata buta warna total.

4.1.1 Alur Bisnis

Alur bisnis yang dimaksud di sini didasarkan pada hasil wawancara dan studi pustaka yang telah dilakukan. Berdasarkan konsep dasar yang telah ditetapkan sebelumnya, maka alur proses interaksi antara pengguna dan aplikasi dimulai dari pengguna menjalankan aplikasi pada telepon genggam berbasis android dengan cara menekan *icon* atau logo aplikasi. Kemudian pengguna akan mendapatkan layar pembuka berupa *splashscreen* yang kemudian dilanjutkan pada menu utama atau *home*.

Pada menu utama akan terdapat logo aplikasi dan beberapa menu untuk melakukan beberapa fasilitas untuk keperluan pengguna, antara lain tombol Mulai Test, Metode Ishihara, Tentang Aplikasi dan Keluar. Keempat tombol tersebut, tombol Mulai Test adalah tombol yang terpenting. Hal ini dikarenakan pengguna diharuskan menekan tombol ini saat memulai sebuah sesi tes mata.

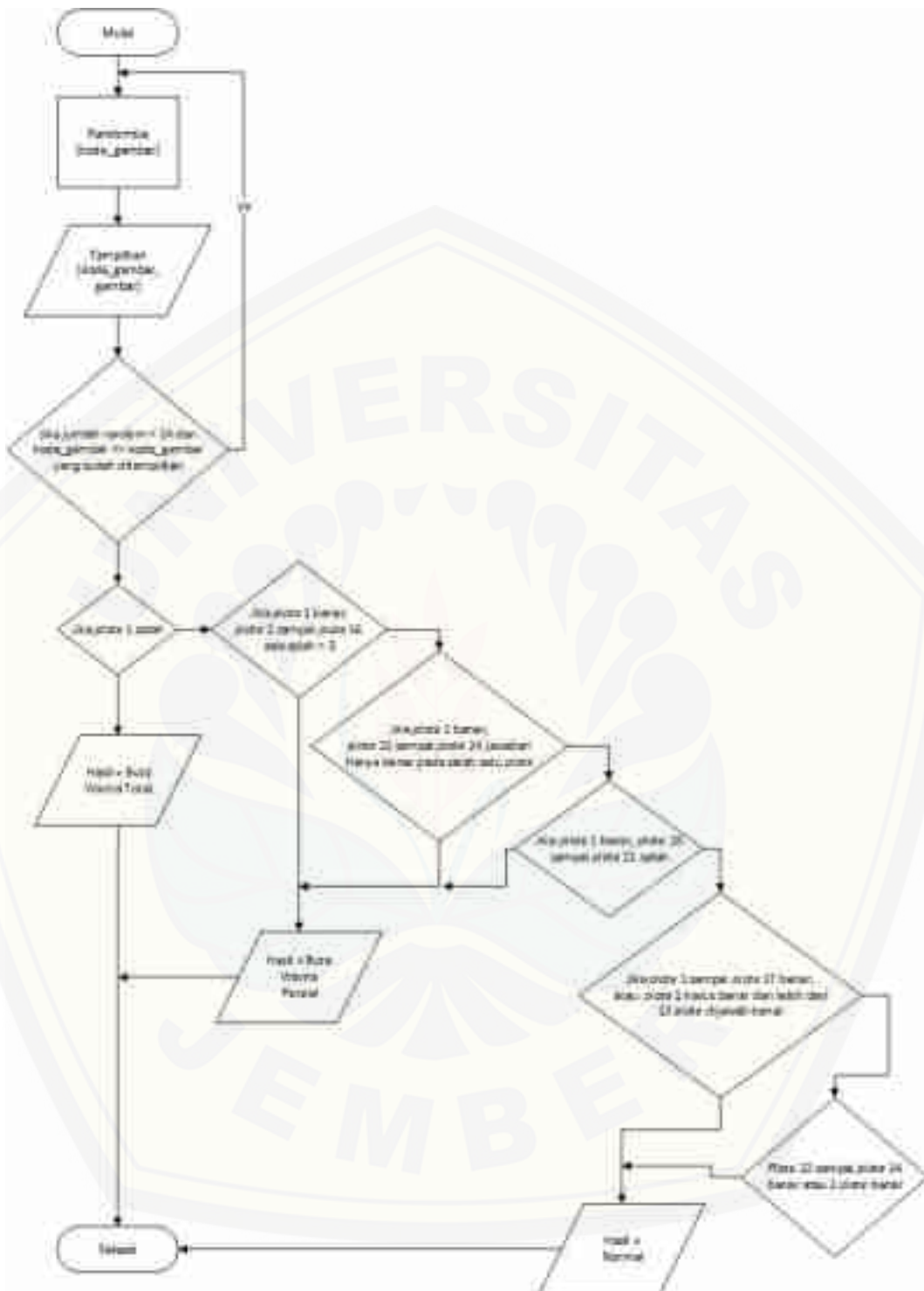
Prosedur melakukan sesi tes untuk seorang pengguna adalah sebagai berikut.

- a. Pengguna menekan tombol Mulai Test.
- b. Pengguna mengisi data diri.
- c. Pengguna menekan tombol Submit.
- d. Sesi tes akan dimulai secara otomatis di mana pengguna akan bertemu dengan 24 macam *plate* dan diharuskan menjawab setiap pertanyaan yang muncul dengan waktu tidak boleh lebih dari 10 detik. Tiap pertanyaan yang lebih dari 10 detik tidak bisa diulang dan akan menghasilkan status jawaban kosong.
- e. Jika semua pertanyaan telah selesai ditampilkan dan terjawab oleh pengguna, maka aplikasi secara otomatis akan menampilkan hasil dari perhitungan atas jawaban pengguna. Hasil tes ini akan mengeluarkan satu dari tiga macam status level mata berdasarkan nilai atau skor yang didapatkan oleh pengguna.
- f. Pengguna juga dapat melihat riwayat jawaban dengan menggunakan fasilitas yang tersedia dalam aplikasi.
- g. Pengguna dapat kembali ke layar utama jika telah cukup untuk sesi tes.

Beberapa prosedur di atas sangat mudah dilakukan oleh pengguna karena akan dipandu oleh aplikasi. Selain tombol untuk memulai suatu sesi tes, aplikasi ini menyediakan tiga tombol lainnya antara lain tombol Metode Ishihara yang berguna untuk mengetahui keterangan detail tentang metode tersebut. Kemudian tombol Tentang Aplikasi, yakni berguna untuk mengetahui informasi tentang aplikasi tes buta warna, dan yang terakhir adalah tombol Keluar yang berguna untuk mengakhiri dan sekaligus keluar dari aplikasi.

4.1.2 *Flowchart* Metode Ishihara

Flowchart pada Gambar 4.1 menggambarkan alur algoritma aplikasi buta warna Metode Ishihara. Dimulai dari *input*, proses, dan pengambilan keputusan. Beberapa istilah yang dipakai dalam *flowchart* tersebut antara lain kode gambar adalah nomor indeks yang menandai urutan sebuah *plate* Ishihara. Gambar adalah tampilan *plate* Ishihara yang ada pada aplikasi saat menjalankan sebuah sesi tes. *Plate* adalah elemen dalam Metode Ishihara sebagai parameter untuk mengambil keputusan apakah seseorang mengalami buta warna atau tidak.



Gambar 4.1 Flowchart aplikasi tes buta warna Metode Ishihara

Gambar 4.1 adalah alur algoritma yang menggambarkan cara kerja aplikasi tes buta warna secara umum. Pada aplikasi ini terdapat beberapa tahapan aliran data hasil tes buta warna yang dijalani oleh seorang pengguna antara lain:

a. Tahap verifikasi jawaban benar atau salah

Tahap ini merupakan tahapan di mana setiap jawaban yang dimasukkan oleh seorang pengguna ke dalam aplikasi akan langsung diproses untuk menentukan status jawaban tersebut menjadi benar atau salah. Tahapan ini terletak pada bagian *randomize* pada *flowchart* Gambar 4.1.

b. Tahap pengumpulan sekaligus pengurutan jawaban benar atau salah

Tahap ini merupakan tahapan di mana setiap jawaban yang telah berstatus benar atau salah akan dikumpulkan sekaligus diurutkan berdasarkan kode gambar ke dalam sebuah data dalam bentuk array. Tahapan ini terletak pada bagian *randomize* pada *flowchart* Gambar 4.1.

c. Tahap pengecekan sekaligus pengambilan keputusan

Tahap ini merupakan tahapan di mana setiap jawaban yang telah diurutkan sebelumnya akan dicek berdasarkan hasil dari tiap status jawaban sesuai dengan Metode Ishihara. Metode Ishihara yang digunakan sebagai acuan pengambilan keputusan tes buta warna tersaji dalam Tabel 4.1.

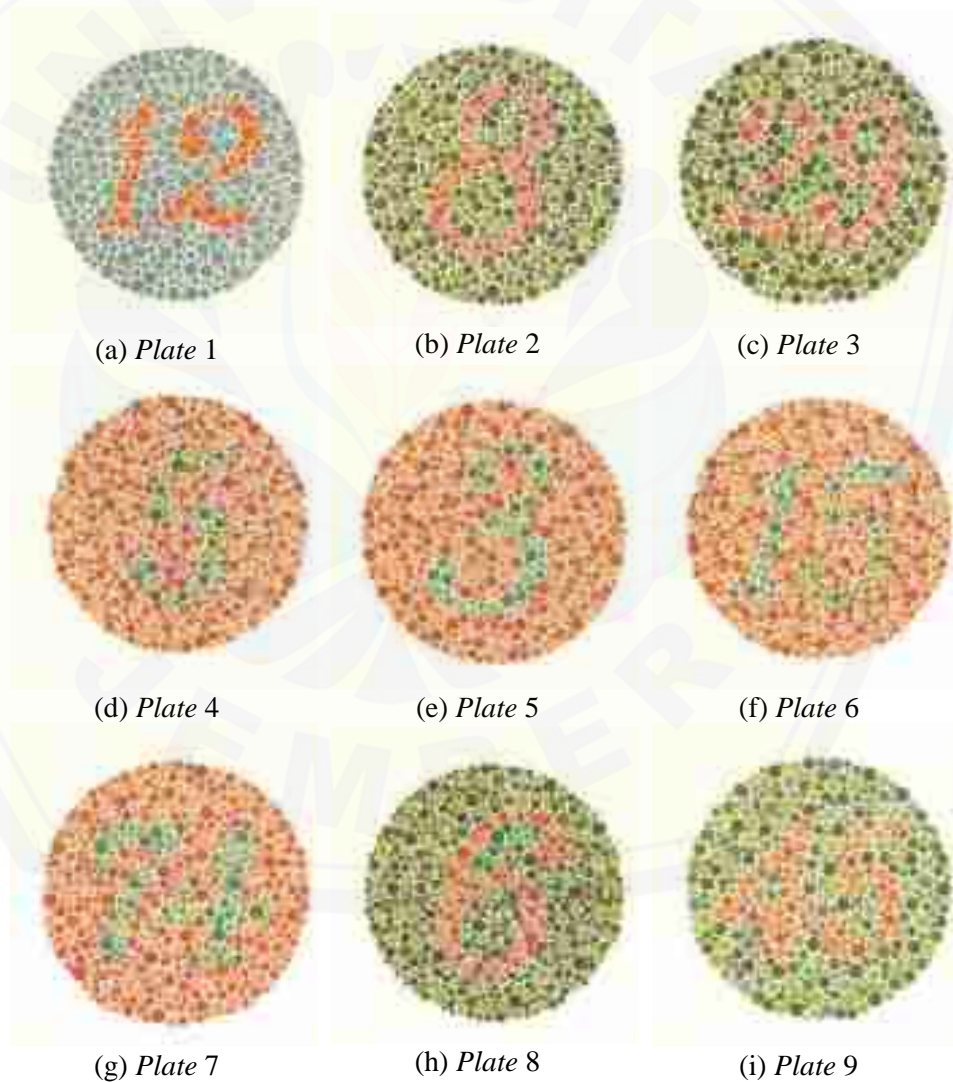
Tabel 4.1 Pengambilan keputusan tes buta warna Metode Ishihara

Kesimpulan Tes	Pengambilan Keputusan
Buta Warna Total	1. Jika <i>plate</i> 1 salah dan jawaban <i>plate</i> lain diabaikan
Buta Warna Parsial	1. Jika <i>plate</i> 1 benar, <i>plate</i> 2 sampai <i>plate</i> 16 ada salah lebih dari 3, atau 2. Jika <i>plate</i> 1 benar, <i>plate</i> 22 sampai <i>plate</i> 24 jawaban hanya benar pada salah satu <i>plate</i> , atau 3. Jika <i>plate</i> 1 benar, <i>plate</i> 18 sampai <i>plate</i> 21 salah

Normal	<ol style="list-style-type: none">1. Jika <i>plate</i> 1 sampai <i>plate</i> 17 benar, atau <i>plate</i> 1 harus benar dan lebih dari 13 <i>plate</i> dijawab benar2. <i>Plate</i> 22 sampai <i>plate</i> 24 benar atau 2 <i>plate</i> benar
--------	---

4.2 *Material Collecting*

Dalam *material collecting* dikumpulkan kebutuhan yang diperlukan untuk membuat aplikasi. Adapun diantaranya sesuai yang tertulis di subbab 4.2.

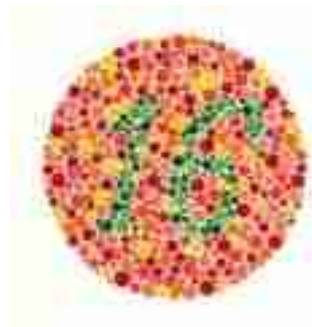




(j) Plate 10



(k) Plate 11



(l) Plate 12



(m) Plate 13



(n) Plate 14



(o) Plate 15



(p) Plate 16



(q) Plate 17



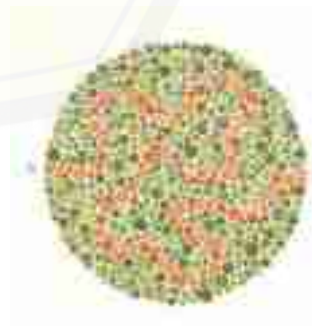
(r) Plate 18



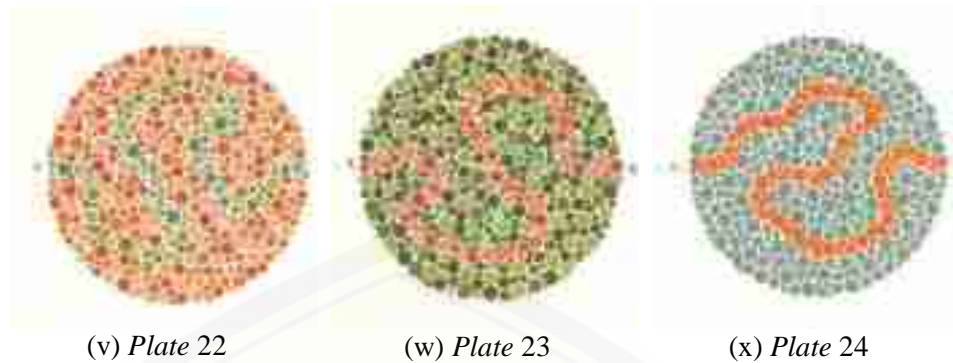
(s) Plate 19



(t) Plate 20



(u) Plate 21



Gambar 4.2 *Plate* Ishihara

Plate Ishihara ini menggunakan 24 macam *plate* sesuai dengan hasil penelitian Dr. Shinobu Ishihara dalam buku yang berjudul “*The Series of Plates Designed as a Test for Colour-Blindness*”.



Gambar 4.3 *Splashscreen* aplikasi tes buta warna



Gambar 4.4 Logo aplikasi tes buta warna

Logo dan *splashscreen* ini dipersiapkan terlebih dahulu yang berasal dari hasil desain manual berformat .png dan .psd kemudian dilanjutkan dengan beberapa langkah proses yakni dengan menyertakan *file* tersebut ke dalam framework ionic tepat pada *folder* resources kemudian untuk mendapatkan ukuran yang sesuai dengan kebutuhan aplikasi berbasis android, maka dilakukan *generate* gambar secara otomatis melalui bantuan *command line* ionic yakni `ionic resources` dengan perintah ini akan didapatkan beberapa ukuran gambar yang sesuai.

4.3 Desain Mockup Aplikasi dan Coding

Pada subbab ini dijabarkan tentang desain tampilan *mockup* aplikasi serta kode aplikasi sesuai desain *mockup* tersebut.



Gambar 4.5 *Splashscreen*

Splashscreen adalah layar pembuka aplikasi yang berguna untuk menampilkan *brand* atau sesuatu hal lain yang berkaitan dengan *credit point* dari si pembuat atau perusahaan pengembang aplikasi. Desain dari *splashscreen* ini

sengaja dibuat dengan konsep minimalis dengan tujuan agar lebih menarik bagi pengguna. Hal ini senada dengan konsep *user interface* (UI) dan *user experience* (UX). *Splashscreen* aplikasi tes buta warna ini menampilkan beberapa elemen antara lain logo aplikasi, nama aplikasi, nama metode yang digunakan, tahun pembuatan, dan nama pembuat aplikasi. Semua elemen ini disusun secara vertikal karena mengacu pada kebanyakan telepon genggam yang berorientasi layar *portrait*.

Splashscreen ini akan tampil secara otomatis pada saat pertama kali aplikasi dijalankan oleh pengguna selama beberapa detik, kemudian setelah masa tampil dari *splashscreen* telah habis, maka akan tampil halaman *home* atau menu utama seperti pada Gambar 4.6.



Gambar 4.6 Desain *mockup* home

Desain *mockup* home ini mempunyai konsep dasar seperti halnya konsep desain pada *splashscreen* namun perbedaan dari kedua desain layar ini adalah terletak pada elemen yang menempel. Pada desain layar home ini ada beberapa tombol fasilitas yang siap membantu pengguna untuk melakukan sesi tes. Ada

sedikit hal yang berbeda dari desain tersebut, yakni terlihat tombol yang menonjol. Konsep ini menganut *material design* dengan ciri khas tombol yang mempunyai drop shadow dengan maksud seolah-olah melayang di atas layar dasar.

Adapun alur kerja desain *mockup* home ini adalah saat tombol Mulai Test ditekan akan tampil halaman untuk mengisi biodata pengguna secara lengkap seperti pada Gambar 4.9. Kemudian jika pengguna menekan tombol Metode Ishihara, maka akan tampil halaman informasi mengenai Metode Ishihara seperti pada Gambar 4.34. Selain itu, saat pengguna menekan tombol Tentang Aplikasi, maka akan tampil halaman tentang informasi aplikasi, seperti deskripsi aplikasi, pembuat aplikasi dan tahun pembuatan aplikasi. Tampilan halaman ini sama seperti pada Gambar 4.37. Tombol terakhir yakni tombol Keluar. Jika tombol ini ditekan, maka aplikasi akan tertutup sekaligus mengakhiri segala aktivitas pada aplikasi secara otomatis.

```

20 </ion-col >
21     <ion-col width=80>
22         <ion-list no-lines>
23             <ion-item class="item item-trns">
24                 <button ion-button icon-left full
25 (click)="mulai ()">
26                     <ion-icon name="create"></ion-icon>
27                     MULAI TEST
28                 </button>
29             </ion-item>
30             <ion-item class="item item-trns">
31                 <button ion-button icon-left full
32 (click)="informasi ()">
33                     <ion-icon name="information-
34 circle"></ion-icon>
35                     METODE ISHIHARA
36                 </button>
37             </ion-item>
38             <ion-item class="item item-trns">

```



```

36         <button ion-button icon-left full
(click)="tentang()">
37             <ion-icon name="leaf"></ion-icon>
38             TENTANG APLI KASI
39         </button>
40     </ion-item>
41     <ion-item class="item item-trns">
42         <button ion-button icon-left full
(click)="keluar()">
43             <ion-icon name="exit"></ion-icon>
44             KELUAR
45         </button>

```

Gambar 4.7 Source code home html

Gambar 4.7 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.6, sebagai contoh pada kode aplikasi tersebut tepatnya baris 23 sampai dengan 29 adalah kode yang membangun elemen tombol dengan tulisan mulai tes. Hal serupa terjadi pada baris-baris selanjutnya yang mana akan membuat tampilan tombol dengan bentuk yang sama namun dengan tulisan yang berbeda.

```

14     constructor(public navCtrl: NavController, public modalCtrl:
Modal Controller, public platform: Platform) {
15     }
16
17     mulai () {
18         let modal = this.modalCtrl.create(Modal bi oPage);
19         modal.onDi dDi smi ss(data => { . . .
20         });
21         modal.present();
22     }
23
24     informasi () {
25         this.navCtrl.push(Informasi Page);
26     }
27

```

```
38     tentang(){
39         this.navCtrl.push(TentangPage);
40     }
41
42     keluar(){
43         this.platform.exitApp();
44     }
```

Gambar 4.8 *Source code home ts*

Gambar 4.8 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi perintah-perintah yang diinginkan oleh pengguna kepada aplikasi. Baris ke 17 adalah perintah untuk menampilkan halaman *form* isian biodata sebagai awal untuk memulai sesi tes. Baris ke 34 adalah perintah untuk menampilkan halaman informasi pengertian Metode Ishihara. Baris ke 38 adalah perintah untuk menampilkan halaman tentang aplikasi, meliputi deskripsi, tahun pembuatan dan informasi mengenai pembuat aplikasi. Baris ke 42 adalah perintah untuk mengakhiri aplikasi dan sekaligus untuk keluar dari aplikasi.



The image shows a mobile application interface for a biodata form. At the top, there is a title bar with the text "isi Biodata" on the left and a "BATAL" button on the right. Below the title bar, there are several input fields: "Nama", "Tempat Lahir", "Tanggal Lahir", "Jenis Kelamin", and "Alamat". Each field has a corresponding label and a text input area. At the bottom of the form, there is a "SELESAI" button.

Gambar 4.9 Desain *form input* biodata

Gambar 4.9 adalah desain *mockup form input* biodata. Desain ini mempunyai konsep dasar seperti halnya konsep desain pada layar home, namun perbedaan dari kedua desain layar ini adalah terletak pada elemen yang menempel.

Pada desain layar *input* biodata ini ada beberapa elemen penyangga layar, yakni *bar* navigasi yang berisi tentang nama halaman yang aktif dan sebuah tombol untuk membatalkan pengisian *form* biodata. Kemudian elemen selanjutnya adalah beberapa bidang isian antara lain adalah untuk nama, tempat lahir, tanggal lahir, jenis kelamin, alamat dan profesi. Bidang – bidang isian ini dimaksudkan untuk mengetahui identitas dari pengguna yang melakukan sesi tes mata. Selanjutnya elemen di bawahnya adalah tombol Submit, yang mana tombol ini berfungsi untuk menyerahkan data pengguna ke aplikasi untuk diolah sebagai *input* awal. Selain itu tombol Submit ini berfungsi untuk memulai sesi tes mata sehingga dengan menekan tombol Submit, maka pengguna dinyatakan siap untuk memulai sesi tesnya.

Adapun alur kerja desain *mockup* home ini adalah jika pengguna menekan tombol batal, maka akan kembali ke halaman home seperti pada Gambar 4.9. Kemudian pengguna mengisi biodata dimulai dari nama, tempat lahir dan bidang isian yang lainnya secara berurutan. Pada saat pengguna menekan tanggal lahir, maka aplikasi akan menampilkan dialog *pop up* tanggal lahir seperti yang terlihat pada Gambar 4.12, sehingga pengguna dapat menentukan tanggal lahirnya dengan mudah. Kemudian dilanjutkan dengan mengisi jenis kelamin, hal serupa akan terjadi seperti saat mengisi tanggal lahir, yakni akan tampil dialog *pop up* jenis kelamin dengan tampilan halaman seperti Gambar 4.14, sehingga pengguna dapat memilih jenis kelamin dengan mudah. Setelah semua bidang *form* biodata selesai dan terisi dengan baik, maka pengguna harus menekan tombol Submit. Jika tombol Submit ditekan, maka aplikasi akan menampilkan halaman tes sekaligus memulai soal yang pertama. Tampilan halaman ini akan sama seperti halaman pada Gambar 4.16 atau 4.18 karena soal akan ditampilkan secara acak.

```
1 <i on-header>  
2   <i on-navbar color="primary">  
3     <i on-title>Isi Biodata :</i on-title>  
4     <i on-buttons end>
```

```

5     <button i on-button
      (click)="dismiss()">BATAL</button>
6     </i on-buttons>
7   </i on-navbar>
8 </i on-header>

```

Gambar 4.10.a Source code form input biodata html

Gambar 4.10.a adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan navigasi desain Gambar 4.9. Kode aplikasi tersebut tepatnya baris 1 sampai dengan 8, perlu dicermati pada bagian `(click)="dismiss()"`, kode ini merupakan *event click* yang mana perintah untuk membatalkan sekaligus menutup halaman yang saat itu sedang terbuka.

```

9 <i on-content>
10 <i on-list>
11 <i on-item>
12 <i on-label stacked>Nama</i on-label >
13 <i on-input type="text" [(ngModel)]="biodata.nama"></i on-
   input>
14 </i on-item>
15 <i on-item>
16 <i on-label stacked>Tempat Lahir</i on-label >
17 <i on-input type="text"
   [(ngModel)]="biodata.tempatlahir"></i on-input>
18 </i on-item>
19 <i on-item>
20 <i on-label stacked>Tanggal Lahir</i on-label >
21 <i on-datetime [(ngModel)]="biodata.tglahir"></i on-
   datetime>
22 </i on-item>
23 <i on-item>
24 <i on-label >Jenis Kelamin</i on-label >
25 <i on-select [(ngModel)]="biodata.jenis kelamin">
26 <i on-option value="Laki-laki">Laki-laki </i on-option>
27 <i on-option value="Perempuan">Perempuan</i on-option>

```

```

28         </i on-select>
29     </i on-item>
30     <i on-item>
31         <i on-label stacked>Alamat</i on-label >
32         <i on-textarea [(ngModel)]="bi odata. al amat" ></i on-
textarea>
33     </i on-item>
34     <i on-item>
35         <i on-label stacked>Profesi </i on-label >
36         <i on-textarea [(ngModel)]="bi odata. profesi " ></i on-
textarea>
37     </i on-item>
38     <i on-item>
39         <button i on-button full
(click)="si mpan()" >SUBMI T</button>
40     </i on-item>
41 </i on-list>

```

Gambar 4.10.b *Source code form input biodata html*

Gambar 4.10.b adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.9 sebagai contoh pada kode aplikasi tersebut tepatnya baris 11 sampai dengan 15 adalah kode yang membangun elemen dengan label nama dan *form input* nama. Perlu dicermati pada bagian `[(NgModel)]="biodata.nama"`, kode ini akan mentransfer data yang dimasukkan oleh pengguna ke dalam aplikasi untuk selanjutnya diproses oleh subkode lainnya. Hal serupa terjadi pada baris-baris berikutnya yang mana akan membuat tampilan elemen dengan bentuk yang sama, namun dengan tulisan yang berbeda. Tetapi ada perbedaan pada beberapa blok kode baris yang akan mendengarkan perintah klik, artinya aplikasi akan merespon interaksi klik dan bukan masukan data seperti pada elemen-elemen sebelumnya. Kode aplikasi baris pada blok baris 1 sampai 8, kode ini akan membatalkan *input* biodata pengguna, kemudian blok baris 38 sampai 41 akan menyimpan data masukan pengguna.

```
1 import { Component } from '@angular/core';
2 import { ViewController, NavController } from 'ionic-
  angular';
3
4 /*
5   Generated class for the Modal bio page.
6
7   See
8   http://ionicframework.com/docs/v2/components/#navigation for
9   more info on
10  Ionic pages and navigation.
11 */
12 @Component({
13   selector: 'page-modal bio',
14   templateUrl: 'modal bio.html'
15 })
16 export class Modal bioPage {
17
18   biodata: any = {};
19
20   constructor(public navCtrl: NavController, public viewCtrl:
  ViewController) {}
21
22   simpan(){
23     this.viewCtrl.dismiss(JSON.stringify(this.biodata));
24   }
25
26   dismiss(){
27     this.biodata = {"nama" : ""};
28     this.viewCtrl.dismiss(JSON.stringify(this.biodata));
29   }
30 }
```

Gambar 4.11 Source code form input biodata ts

Gambar 4.11 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk menyimpan satu set biodata dari pengguna. Kode aplikasi ini berupa kelas yang mempunyai sebuah konstruktor dengan membutuhkan dua kelas yang berbeda sebagai komponen pembangunnya, antara lain NavController dan ViewController. Kedua kelas ketergantungan tersebut diperoleh dari hasil perintah `import { ViewController, NavController } from 'ionic-angular';` pada baris 2. Kemudian subkode pada Modalbiopage dapat mengambil dan menyimpan data *input* yang berasal dari Gambar 4.10. dengan perintah sebagai berikut:

```
simpan(){      this.viewCtrl.dismiss(JSON.stringify(this.biodata));}
```

yang disimpan pada variabel `this.biodata`, terkadang pengguna juga membatalkan pendaftaran biodatanya. Jika hal itu terjadi aplikasi akan menggunakan kode aplikasi berikut ini untuk membatalkan masukan dari pengguna.

```
dismiss(){  
  this.biodata = {"nama" : ""};  
  this.viewCtrl.dismiss(JSON.stringify(this.biodata));  
}
```



Month	Day	Year
Jan	1	2011
Feb	2	2011
Mar	3	2011

Gambar 4.12 Desain *form input* tanggal

Gambar 4.12 adalah desain *mockup form input* tanggal. Desain ini mempunyai konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain dua tombol untuk persetujuan berupa tombol (Cancel dan Done) dan elemen pilihan tanggal, bulan, tahun yang berada di bawahnya.

Adapun alur kerja dari desain ini adalah awalnya pengguna bisa memilih tanggal, bulan, tahun dengan cara geser ke atas dan ke bawah, kemudian jika sudah tepat pada tanggal yang dipilih, maka pengguna dapat memberikan persetujuan *input* dengan menekan tombol Done, dan sebaliknya pengguna dapat membatalkan *input* tanggal lahir dengan menekan tombol Cancel. Menekan tombol Cancel ataupun Done, maka tampilan aplikasi akan kembali pada halaman *form* biodata seperti pada Gambar 4.9, perbedaan dari kedua aksi ini adalah jika pengguna menekan tombol Cancel bidang isian tanggal lahir akan tetap kosong, sebaliknya jika pengguna menekan tombol Done, maka bidang isian tanggal lahir akan terisi dengan tanggal lahir yang dipilih oleh pengguna.

```
19     <i on-i tem>  
20         <i on-l abel  stacked>Tanggal  Lahi r</i on-l abel  >  
21         <i on-dateti me  [(ngModel )]="bi odata. tgl l ahi r"></i on-  
    dateti me>  
22     </i on-i tem>
```

Gambar 4.13 *Source code form input* tanggal html

Gambar 4.13 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.12. Kode ini akan memberikan tampilan label tanggal lahir dan bidang isian *input* tanggal lahir. Perlu dicermati pada baris ke 21 terdapat kode yang bertugas untuk sinkronisasi dengan kode typescript untuk menerima dan mengirim data tanggal lahir.

A mobile application mockup showing a form titled "Jenis Kelamin". The form is centered on the screen and has a white background. It features a title "Jenis Kelamin" at the top, followed by two radio button options: "Laki-laki" and "Perempuan". Below the options are two buttons: "CANCEL" and "OK". The background of the mockup is a blurred image of a mobile app interface with a large watermark of the Universitas Jember logo.

Gambar 4.14 Desain *form input* jenis kelamin

Gambar 4.14 adalah desain *mockup form input* jenis kelamin. Desain ini mempunyai konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain judul panel yang bertuliskan jenis kelamin dan dua masukan jenis radio untuk memilih salah satu jenis kelamin antara laki-laki atau perempuan. Selain itu terdapat elemen untuk memberikan persetujuan atau pembatalan *input*, yakni berupa tombol Ok dan Cancel.

Adapun alur kerja dari desain ini adalah jika pengguna sudah cocok pada jenis kelamin yang dipilih, maka pengguna dapat memberikan persetujuan *input* dengan menekan tombol Ok, dan sebaliknya pengguna dapat membatalkan *input* jenis kelamin dengan menekan tombol Cancel. Jika sudah memilih jenis kelamin dan menekan tombol Ok, bidang isian jenis kelamin akan terisi dengan nama jenis kelamin yang dipilih, setelah itu halaman akan kembali pada bidang isian *form* biodata seperti pada Gambar 4.9. Jika pengguna sudah memilih atau belum memilih

jenis kelamin, lalu menekan tombol Cancel, maka akan kembali pada bidang isian seperti Gambar 4.9 dengan *form* jenis kelamin yang kosong.

```
23     <i on-i tem>  
24         <i on-l abel >Jeni s Kel ami n</i on-l abel >  
25         <i on-sel ect [(ngModel)]="bi odata. j eni skel ami n" >  
26             <i on-opti on val ue="Laki -l aki ">Laki -l aki </i on-opti on>  
27             <i on-opti on val ue="Perempuan">Perempuan</i on-opti on>  
28         </i on-sel ect>  
29     </i on-i tem>
```

Gambar 4.15 *Source code form input jenis kelamin html*

Gambar 4.15 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.14. Baris 25 adalah kode yang membangun tampilan label jenis kelamin dan blok baris 25 sampai dengan 28 untuk menampilkan pilihan jenis kelamin laki-laki dan perempuan. Perlu dicermati pada baris ke 25 terdapat kode yang bertugas untuk sinkronisasi dengan kode typescript untuk menerima dan mengirim data tanggal lahir.



Gambar 4.16 *Desain form input tes plate angka*

Gambar 4.16 adalah desain *mockup form input plate* angka. Desain ini masih menggunakan konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain bar navigasi yang mempunyai dua elemen yakni judul yang berfungsi memberikan informasi layar yang aktif saat ini dan waktu yang bertugas memberikan informasi waktu yang dibutuhkan untuk menyelesaikan satu pertanyaan, pengguna harus cepat dalam memberikan jawaban atas pertanyaan tersebut karena waktu akan terus berjalan mundur secara otomatis.

Elemen selanjutnya yang terdapat pada desain ini adalah *Plate Ishihara* yang mana akan menampilkan gambar sebagai bahan dasar tes mata berupa angka atau pola. Pengguna diwajibkan menebak visualisasi dari gambar tersebut dalam kurang dari 10 detik. Selain elemen-elemen yang telah disebutkan sebelumnya pada desain ini masih terdapat satu set elemen lagi yang sangat berpengaruh yaitu *form input* dan satu tombol untuk menyimpan jawaban dan menampilkan halaman soal yang baru. Berdasar satu set desain yang utuh seperti ini, maka pengguna dapat dengan mudah menentukan satu jawaban dengan mengisi bidang *form* isian kurang dari 10 detik, lalu dilanjutkan dengan menekan tombol Next untuk melanjutkan pertanyaan selanjutnya.

Adapun alur kerja dari desain ini adalah pengguna harus mengisi bidang isian yang telah tersedia kemudian dilanjutkan dengan menekan tombol Next. Jika pengguna telah melakukan kegiatan tersebut aplikasi akan menampilkan soal berikutnya dengan tampilan halaman seperti pada Gambar 4.16 atau 4.18, hal ini bisa terjadi dikarenakan soal akan ditampilkan secara acak oleh aplikasi. Jika seluruh halaman soal selesai ditampilkan, maka aplikasi akan menghitung hasil sesi tes, kemudian hasil dari tes ini akan ditampilkan ke dalam sebuah halaman seperti pada Gambar 4.21.

14 <i on-content >

15 <i on-gri d >

16 <i on-row >

17 <i on-col >


```

18         
19     </ion-col>
20 </ion-row>
21 <ion-row>
22     <ion-col width=10>
23 </ion-col>
24     <ion-col width=80>
25
26     <ion-list *ngIf="nm <= 17" no-lines>
27         <ion-item>
28             <ion-label floating>Angka yang terlihat : </ion-
label>
29                 <ion-input type="number"
[(ngModel)]="jawaban"></ion-input>
30             </ion-item>
31             <ion-item>
32                 <button *ngIf="nm <= 17" ion-button icon-right full
(click)="next(jawaban)">
33                     NEXT
34                 <ion-icon name="arrow-dropright"></ion-icon>
35             </button>
36         </ion-item>
37     </ion-list>

```

Gambar 4.17 Source code form input tes plate angka html

Gambar 4.17 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.16. Sebagai contoh pada kode aplikasi tersebut tepatnya blok baris 16 sampai dengan 20 adalah kode yang membangun elemen gambar *Plate* Ishihara dan kemudian blok baris 26 sampai 37 adalah kode baris yang membangun elemen *form input* angka sekaligus membangun tombol dengan tulisan next. Perlu dicermati pada bagian kode baris 29 terdapat kode `<ion-input type="number" [(ngModel)]="jawaban"></ion-`

`input>`, kode ini bertugas untuk sinkronisasi jawaban ke komponen model dengan nama variabel jawaban. Dan kode `<button *ngIf="nm <= 17" ion-button icon-right full (click)="next(jawaban)">` dengan interaksi klik bertugas untuk memanggil fungsi `next` dengan parameter variabel jawaban yang ada pada file `typescript`.



Gambar 4.18 Desain *form input* tes *plate* pola

Gambar 4.18 adalah desain *mockup form* input *plate* pola. Desain ini pada dasarnya sama dengan Gambar 4.16 akan tetapi ada beberapa elemen yang berbeda yakni pada tombol masukan. Pada desain sebelumnya (Gambar 4.16) hanya terdapat satu tombol saja namun pada desain tes pola ini elemen tombol ada tiga buah dengan fungsi hampir sama tetapi dengan nilai yang berbeda sesuai nilai tulisan pada desain tersebut yakni nol, satu dan dua.

Adapun alur kerja dari desain ini adalah pengguna harus mengisi bidang isian yang telah tersedia kemudian dilanjutkan dengan menekan salah satu dari tiga tombol yang tersedia. Jika pengguna telah melakukan kegiatan tersebut, maka

aplikasi akan menampilkan soal berikutnya dengan tampilan halaman seperti pada Gambar 4.16 atau 4.18, hal ini bisa terjadi dikarenakan soal akan ditampilkan secara acak oleh aplikasi. Jika seluruh halaman soal selesai ditampilkan, maka aplikasi akan menghitung hasil sesi tes kemudian hasil dari tes ini akan ditampilkan ke dalam sebuah halaman seperti pada Gambar 4.21.

```

17 <i on-col >
18     
19     </i on-col >
20 </i on-row>
21 <i on-row>
22     <i on-col wi dth-10>
23     </i on-col >
24     <i on-col wi dth-80>
25
26     . . .
27
28     <i on-l i st *ngl f="nm >= 18" no-l i nes>
29         Berapa juml ah pol a/al ur yang terli hat ?
30     <i on-i tem>
31         <butt on i on-but ton i con-ri ght full
    (cl i ck)="next(0)" >
32
33         </butt on>
34         <butt on i on-but ton i con-ri ght full
    (cl i ck)="next(1)" >
35
36         </butt on>
37         <butt on i on-but ton i con-ri ght full
    (cl i ck)="next(2)" >
38

```

0
1
2

```

39         </button>
40     </ion-item>
41 </ion-list>

```

Gambar 4.19 *Source code form input tes plate pola html*

Gambar 4.19 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain Gambar 4.18. Kode html ini pada dasarnya sama dengan kode aplikasi pada Gambar 4.17, namun perbedaan yang paling mendasar adalah terdapat pada blok baris yang membangun tombol next. Pada desain ini terdapat tiga tombol, elemen ini dibangun oleh blok kode baris 28 sampai dengan 41. Masing – masing tombol mempunyai fungsi yang sama, namun tiap tombol berbeda nilai yang dibawa. Sebagai contoh pada kode aplikasi tersebut tepatnya blok baris 31 sampai dengan 33 dengan kode aplikasi `<button ion-button icon-right full (click)="next(0)">0</button>` adalah kode yang membangun elemen tombol dengan nilai yang dibawa adalah nilai 0. Hal seperti ini sama pada tombol – tombol lainnya hanya saja nilai parameternya adalah 1 dan 2.

```

15 export class TestPage {
16
17   nm: any;
18   img: any;
19   pos: any = 0;
20   arr_nm: any[] = [];
21   jawaban: any;
22   waktu: any = 10;
23   timer: any;
24   arr_jawaban: any[] = [];
25   kunci_normal =
[0, 12, 8, 29, 5, 3, 15, 74, 6, 45, 5, 7, 16, 73, '', '', 26, 42, 2, '', 1, 1, 1, 1, 1];
26   kunci_redgreen = [0, 12, 3, 70, 2, 5, 17, 21, '', '', '', '',
'', '', 5, 45, "S", "S", 1, 1, '', '', '', '', '']
27   cek_jawaban = [];
28

```

```
29   constructor(public navCtrl: NavController, public navParams:
NavParams) {
30     this.firstload();
31   }
32
33   firstload(){
41     if(this.pos == 0){ . . .
42     }
43   }
44
45   next(a){ . . .
100  }
101
102   randomGambar(){ . . .
111  }
110
113   countdown(){ . . .
124  }
```

Gambar 4.20 *Source code form input tes plate angka dan pola ts*

Gambar 4.20 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk serangkaian satu sesi tes mulai dari awal sampai perhitungan tes dijalankan sehingga didapatkan hasil yang diinginkan. Pada awalnya aplikasi mendeklarasi beberapa variabel awal yang digunakan untuk menyimpan data masukan saat tes dan data statis untuk acuan tes, deklarasi ini terlihat pada blok kode 17 sampai 26. Kemudian terdapat konstruktor(baris kode ke 29 sampai dengan 31), yakni kelas yang berfungsi untuk membangun konstruksi sebuah kelas dengan beberapa kelas lain yang ia butuhkan, hal ini dilakukan sama dengan beberapa proses sebelumnya, yakni dengan perintah import seperti ini `import { NavController, NavParams } from 'ionic-angular';`. Pada saat objek atau class ini dipanggil, maka fungsi `firstload()` akan otomatis dipanggil pertama kali.

Pada baris 33 terdapat fungsi `firstload()` fungsi ini adalah fungsi yang bertugas mempersiapkan variabel dan memanggil fungsi lain jika beberapa kondisi

yang dibutuhkan telah terpenuhi. Algoritma yang terkandung dalam fungsi `firstload()` ini adalah sebagai berikut:

- a. Jika variabel `pos` (posisi random) sama dengan 0, maka
- b. Variabel jawaban diisi dengan nilai ‘’, kemudian variabel `pos` ditambahkan dengan nilai 1, kemudian variabel `nm` (nama plate) diisi dengan fungsi `randomGambar()`.
- c. Jika nilai variabel `nm` lebih besar atau sama dengan 18, maka nilai variabel `nm` diisi dengan alamat gambar pola dengan nama `nm` berekstensi `jpg` selain itu jika tidak memenuhi kondisi itu, maka variabel `nm` diisi dengan alamat gambar angka dengan nama `nm` berekstensi `jpg`, kemudian
- d. Variabel waktu diisi dengan nilai 10, kemudian panggil fungsi `clearInterval(this.timer)` dengan parameter variabel `timer`. Kemudian panggil fungsi `countdown()`.

Begitulah alur yang dijalankan oleh fungsi `firstload()`, sehingga halaman tes akan siap digunakan oleh pengguna. Setiap halaman tes pengguna dipastikan akan menekan tombol `next` atau `[0,1,2]`, maka pada dasarnya pengguna akan menjalankan fungsi `next(parameter)`.

Algoritma yang terkandung pada fungsi `next(parameter)` terdapat banyak percabangan kondisi yang berfungsi menentukan arah dan nilai pada hasil akhir tes. Kode aplikasi fungsi ini terdapat pada blok baris kode 44 sampai dengan 97. Alur algoritma dari fungsi ini adalah sebagai berikut.

- a. Deklarasikan variabel `e` dan `n`.
- b. Kemudian kondisikan:
 - 1) Jika nilai variabel `a` sebagai parameter dari fungsi `next` sama dengan nilai dari variabel `array_kunci_normal` dengan indeks nilai dari variabel `nm`, maka nilai variabel `e` diisi dengan nilai “Benar”,
 - 2) Selain itu jika nilai variabel `nm` sama dengan 16, maka kondisikan:
 - a) Jika nilai variabel `a` sama dengan 2, maka nilai variabel `n` diisi dengan nilai “p”
 - b) Selain itu jika nilai variabel `a` sama dengan 6, maka nilai variabel `n` diisi dengan “D”
nilai variabel `e` diisi dengan nilai “Salah”.

- 3) Selain itu jika nilai variabel nm sama dengan 17, maka kondisikan:
 - a) Jika nilai variabel a sama dengan 4, maka nilai variabel n diisi dengan nilai "p"
 - b) Selain itu jika nilai variabel a sama dengan 2, maka nilai variabel n diisi dengan "D"
Selanjutnya nilai variabel e diisi dengan nilai "Salah".
- 4) Selain itu, maka nilai variabel e diisi dengan nilai "Salah".
- c. Nilai variabel a sama dengan "" (tanpa nilai pada tipe data string), maka variabel nilai a diisi dengan nilai string "Kosong"
- d. Deklarasikan variabel dengan nama param yang diisi dengan objek sebagai berikut.
 - no = nilai dari variabel nm,
 - jawaban = nilai dari variabel a,
 - waktu = 10 - nilai variabel waktu,
 - res = nilai dari variabel e,
 - t = nilai dari variabel n,
 - kunci = nilai dari variabel array kunci_normal berindeks nilai variabel nm,
 - url = nilai dari variabel img
- e. Masukkan nilai variabel param ke variabel array arr_jawaban dengan metode push, kemudian nilai variabel jawaban diisi dengan tipe string dengan nilai sama dengan ""
- f. Jika nilai variabel pos kurang dari 24, maka kondisikan:
 - 1) Nilai variabel pos ditambahkan dengan nilai 1, kemudian
 - 2) Nilai variabel nm diisi dengan hasil dari fungsi randomGambar(), kemudian kondisikan:
 - 3) Jika nilai variabel nm lebih besar atau sama dengan 18, maka nilai variabel nm diisi dengan alamat gambar pola dengan nama nm berekstensi jpg, selain itu jika tidak memenuhi kondisi itu, maka variabel nm diisi dengan alamat gambar angka dengan nama nm berekstensi jpg, kemudian
 - 4) Variabel waktu diisi dengan 10, kemudian
 - 5) Panggil fungsi clearInterval(waktu) dengan parameterannya adalah variabel waktu, kemudian

- 6) Panggil fungsi `countdown()`
- g. Selain itu jika nilai variabel `pos` sama dengan 24, maka variabel `nm` diisi dengan nilai tipe string sama dengan "FINISH", kemudian variabel waktu diisi dengan tipe string sama dengan "", kemudian panggil fungsi `push()` yang terdapat pada variabel `navCtrl` dengan parameter `HasilPage` dan variabel objek bernama `hasil` yang berisi data array variabel `arr_jawaban`.

Pada blok baris kode 102 sampai dengan 111 terdapat fungsi `randomGambar()` yang bertugas sebagai penampil gambar secara acak, fungsi ini dimanfaatkan untuk menampilkan gambar secara acak tiap sesi soal saat tes. Fungsi ini mempunyai algoritma sendiri karena fungsi ini bertindak sebagai subkode yang dimanfaatkan oleh fungsi lain. Adapun algoritma fungsi `randomGambar()` adalah sebagai berikut.

- a. Deklarasikan variabel `x` dengan tipe `any`
- b. Selama bernilai benar, maka nilai variabel `x` diisi dengan hasil fungsi `Math.Floor(Mat.random() dikalikan dengan 24)` ditambahkan dengan 1, kemudia jika nilai indeks dari variabel array `arr_nm` sama dengan nilai -1, maka tambahkan nilai variabel `x` ke dalam variabel array `arr_nm` dengan metode `push()`, kemudian kembalikan nilai variabel `x` sebagai hasil akhir.

Memfaatkan fungsi ini, maka soal dapat ditampilkan secara acak, sehingga pengguna tidak akan dapat menjiplak jawaban dari pengguna lain atau histori jawaban lainnya.

Pada blok baris kode 110 sampai dengan 121 terdapat fungsi `countdown()` yang bertugas sebagai penghitung mundur, fungsi ini dimanfaatkan untuk membatasi waktu tiap sesi soal saat tes. Fungsi ini mempunyai algoritma sendiri karena fungsi ini bertindak sebagai subkode yang dimanfaatkan oleh fungsi lain. Adapun algoritma dari fungsi ini adalah sebagai berikut.

- a. Deklarasikan variabel waktu, yang mana nilai variabel waktu adalah hasil dari fungsi `setInterval()` dengan parameter yang mana kondisikan terlebih dahulu.
 - 1) Jika nilai variabel waktu tidak sama dengan 0 dan waktu tidak sama dengan -1, maka nilai variabel waktu diisi dengan nilai satu,
 - 2) Selain itu jika waktu sama dengan -1 tidak ada yang harus dilakukan

- 3) Selain itu semua panggil fungsi `clearInterval()` dengan parameter variabel waktu kemudian panggil fungsi `nex()` dengan parameter tipe string bernilai "" dan parameter 1000.



Gambar 4.21 Desain hasil tes Ishihara

Gambar 4.21 adalah desain *mockup* dari hasil tes Ishihara. Desain ini masih menggunakan konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain bar navigasi yang mempunyai dua elemen, yakni judul yang berfungsi memberikan halaman yang sedang aktif, dan dua buah *icon*, yaitu Plates yang mana berfungsi untuk menampilkan *plate-plate* yang telah dijawab sebelumnya, dan Home yang berfungsi untuk kembali ke halaman utama atau awal.

Kemudian blok selanjutnya adalah blok biodata. Pada desain blok biodata ini terdapat informasi data identitas diri dari pengguna yang melakukan sesi tes mata. Biodata yang dimaksud terdiri atas nama, tempat lahir, tanggal lahir, jenis kelamin,

alamat, profesi, mulai test dan akhir test. Biodata ini dimaksudkan sebagai salah satu bahan untuk validasi data pengguna.

Selanjutnya terdapat blok informasi yang memuat hasil dari sesi tes pengguna, blok ini disebut dengan blok hasil tes Metode Ishihara. Blok ini berfungsi untuk mengetahui hasil akhir dari serangkaian sesi tes yang telah dilakukan. Jika melihat informasi pada blok ini pengguna dapat memastikan status mata mereka saat itu juga, Hasil tes yang ditampilkan pada blok hasil ini tidak lepas dari perhitungan yang dilakukan oleh aplikasi berdasarkan pada beberapa masukan yang telah diberikan oleh pengguna sesaat sebelumnya. Oleh karena itu penting agar pengguna memberikan informasi atau jawaban dengan teliti dan dengan sebenar-benarnya.

Kemudian blok informasi selanjutnya adalah berupa tabel detail dari hasil serangkaian tes dari pengguna. Dalam tabel detail ini terdapat empat kolom yang terdiri atas nomor gambar, jawaban, waktu, dan hasil. Deskripsi masing-masing fungsi dari kolom tersebut adalah sebagai berikut.

- a. Kolom nomor gambar akan mewakili nomor gambar yang muncul saat tes
- b. Kolom jawaban akan mewakili jawaban pengguna atas gambar yang muncul saat tes
- c. Kolom waktu akan mewakili waktu yang dibutuhkan oleh pengguna untuk menjawab satu pertanyaan yang muncul saat tes berlangsung
- d. Kolom hasil akan mewakili keputusan jawaban dari pengguna atas soal yang diberikan adalah berstatus salah atau benar. Hal ini didasarkan pada kecocokan antara kunci jawaban dan jawaban dari pengguna

Jika melihat halaman hasil tes ini pengguna sangat mudah dalam memanfaatkan hasil tes karena informasi yang disajikan sudah cukup lengkap.

Alur kerja halaman ini adalah jika pengguna ingin melihat riwayat dari seluruh hasil jawaban yang telah dikerjakan, maka pengguna cukup menekan tombol *plates* dengan aksi ini, maka halaman akan berubah seperti pada gambar 4.31. Lain halnya dengan pengguna menekan tombol Home, maka jika itu terjadi aplikasi akan menampilkan halaman home seperti pada Gambar 4.6.

```

1 <i on-header>
2   <i on-navbar color="primary">
3     <i on-title>Test Buta Warna</i on-title>
4     <i on-buttons *ngIf="waktu > -1" end>
5       <button i on-button>
6         <h6>
7           <i on-icon name="timer"></i on-icon>
8           : {{ waktu }} detik
9         </h6>
10        </button>
11      </i on-buttons>
12    </i on-navbar>
13  </i on-header>
14

```

Gambar 4.22 *Source code* hasil tes Ishihara bagian 1 html

Gambar 4.22 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Baris 3 adalah kode yang membangun informasi halaman yang sedang aktif, kemudian blok baris kode 4 sampai dengan 10 adalah kode yang membangun elemen *icon* Plate dan *icon* Home.

```

15 <i on-content>
16   <i on-card>
17     <i on-card-header>Bi odata Peserta Test :</i on-card-header>
18     <i on-card-content style="text-align: left;">
19       <i on-row>
20         <i on-col width-25>Nama</i on-col >
21         <i on-col width-10>:</i on-col >
22         <i on-col >{{ bi odata. nama }}</i on-col >
23       </i on-row>
24       <i on-row>
25         <i on-col width-25>Tempat Lahir</i on-col >
26         <i on-col width-10>:</i on-col >
27         <i on-col >{{ bi odata. tempatLahir }}</i on-col >

```

```

28     </i on-row>
29     <i on-row>
30         <i on-col width-25>Tanggal Lahir</i on-col >
31         <i on-col width-10>: </i on-col >
32         <i on-col >{{ bi odata. tgl l ahi r }}</i on-col >
33     </i on-row>
34     <i on-row>
35         <i on-col width-25>Jeni s Kel ami n</i on-col >
36         <i on-col width-10>: </i on-col >
37         <i on-col >{{ bi odata. j eni skel ami n }}</i on-col >
38     </i on-row>
39     <i on-row>
40         <i on-col width-25>Al amat</i on-col >
41         <i on-col width-10>: </i on-col >
42         <i on-col >{{ bi odata. al amat }}</i on-col >
43     </i on-row>

```

Gambar 4.23 *Source code* hasil tes Ishihara bagian 2 html

Gambar 4.23 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Baris 17 sampai dengan 43 adalah kode yang membangun informasi biodata pengguna antara lain nama, tempat lahir, tanggal lahir, jenis kelamin dan alamat pengguna.

```

43     </i on-row>
44     <i on-row>
45         <i on-col width-25>Profesi </i on-col >
46         <i on-col width-10>: </i on-col >
47         <i on-col >{{ bi odata. profesi }}</i on-col >
48     </i on-row>
49     <i on-row>
50         <i on-col width-25>Mul ai Tes</i on-col >
51         <i on-col width-10>: </i on-col >
52         <i on-col >{{ bi odata. mul ai test }}</i on-col >

```



```

53     </i on-row>
54     <i on-row>
55         <i on-col width-25>Akhir Tes</i on-col >
56         <i on-col width-10>: </i on-col >
57         <i on-col >{{ akhirstest }}</i on-col >
58     </i on-row>
59 </i on-card-content>
60 </i on-card>
61 <i on-card>
62     <i on-card-header>Kesimpulan : </i on-card-header>
63     <i on-card-content style="text-align: center; ">
64         <h1 style="font-weight: bold; ">{{ kesimpulan }}</h1>
65     </i on-card-content>
66 </i on-card>

```

Gambar 4.24 *Source code* hasil tes Ishihara bagian 3 html

Gambar 4.24 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Baris 44 sampai dengan 61 adalah kode yang membangun informasi biodata pengguna antara lain profesi, mulai test, akhir test. Sedangkan pada blok kode 61 sampai dengan 66 adalah kode yang membangun elemen kesimpulan hasil tes. Perlu dicermati pada baris kode ke 47 terdapat kode aplikasi `{{biodata.profesi}}` kode ini berfungsi untuk menampilkan data profesi dari variabel objek biodata sehingga bidang biodata profesi akan terisi dengan data tersebut.

```

67 <i on-card>
68     <i on-card-header>Hasil Jawaban Test : </i on-card-header>
69     <i on-card-content>
70         <i on-grid>
71             <i on-row class="header" >
72                 <i on-col >
73                     Gambar
74                 </i on-col >
75                 <i on-col >
76                     Jawaban
77                 </i on-col >
78                 <i on-col >
79                     Waktu
80                 </i on-col >
81                 <i on-col >

```

```

82         Hasil
83     </ion-col >
84 </ion-row>
85 <ion-row *ngFor="let item of hasil; let i = index">
86     <ion-col >
87         {{ (i+1) }}
88     </ion-col >
89     <ion-col >
90         {{ item.jawaban }}
91     </ion-col >
92     <ion-col >
93         {{ item.waktu }} detik
94     </ion-col >
95     <ion-col >
96         {{ item.res }}
97     </ion-col >
98 </ion-row>
99 </ion-grid>
100 </ion-card-content>

```

Gambar 4.25 *Source code* hasil tes Ishihara bagian 4 html

Gambar 4.25 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Baris 67 sampai dengan 100 adalah kode yang membangun informasi tabel detail hasil tes mata. Perlu dicermati beberapa baris kode aplikasi seperti pada ke 85, baris kode ini akan melakukan proses *looping* sebanyak jumlah data yang ada pada variabel hasil kemudian data akan disesuaikan dengan nama kolom seperti variabel `item.jawaban` akan dipassing dengan kolom jawaban dan seterusnya. Kemudian akan didapatkan beberapa baris sesuai dengan banyak data dari variabel hasil.

```

1  import {Component} from '@angular/core';
2  import {NavController, NavParams, ModalController} from
3  'ionic-angular';
4  import {HomePage} from '../home/home';
5  import {ModalGambarPage} from '../modal-gambar/modal-gambar';
6  /*
7  Generated class for the Hasil page.
8

```

```

    See http://ionicframework.com/docs/v2/components/#navigation
  9 for more info on
 10 Ionic pages and navigation.
 11 */
 12 @Component({
 13   selector: 'page-hasil',
 14   templateUrl: 'hasil.html'
 15 })
 16 export class HasilPage {
 17
 18   hasil: any;
 19   kesimpulan: any;
 20   cek15: boolean = false;
 21   biodata: any;
 22   akhirstest: any;
 23   cek: any;
 24
 25   constructor(public navCtrl: NavController, public
 26     navCtrl: NavController, public modalCtrl: ModalController) {
 27     this.hasil = this.navCtrl.get('hasil');
 28     this.cek = this.navCtrl.get('cek');
 29     console.log(this.hasil);
 30     this.biodata =
 31     JSON.parse(window.localStorage.getItem('biodata'));
 32     var tzoffset = (new Date()).getTimezoneOffset() * 60000;
 33     //offset in milliseconds
 34     var localISOTime = (new Date(Date.now() -
 35     tzoffset)).toISOString().slice(0, -1);
 36     this.akhirstest = localISOTime.replace("T", " ");
 37     this.prosesKesimpulan(this.cek);
 38   }
 39 }

```

Gambar 4.26 Source code hasil tes Ishihara bagian 1 ts

Gambar 4.26 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk memproses hasil dari satu sesi tes mulai dari awal sampai akhir. Terlihat pada gambar tersebut ada banyak baris kode yang membangun pemrosesan data dari dan untuk pengguna. Sebenarnya semua cara kerja halaman pada bagian kode typescript pada aplikasi ini adalah sama pada saat mengimpor *library code* dari paket satu ke paket yang lain tetapi khusus untuk halaman hasil ini ditampilkan pembahasannya. Aktivitas impor paket ini dimulai dari blok baris kode 1 sampai dengan 4, kode aplikasi ini adalah proses impor *library code* mulai dari Component, NavController, HomePage,

ModalGambarPage dari beberapa paket pada lingkungan aplikasi ini. Sebagai contoh paket kelas Component diimpor dari paket @angular/core dan seterusnya. Hal ini dikarenakan kelas HasilPage membutuhkan layanan dari masing-masing *library code* yang telah diimpor tadi.

Selanjutnya pada blok baris kode 12 sampai dengan 15, maka aplikasi menentukan *selector* dan alamat *template* yang ditandai dengan perintah @component yang mempunyai parameter sebuah objek yang berisi selector: 'page-hasil' dan templateUrl: 'hasil.html'. *Selector* akan mencari *tag* html dengan nama page-hasil untuk selanjutnya akan dikenai tindakan oleh kelas HasilPage entah berupa penambahan elemen atau transfer data variabel. Sedangkan templateUrl adalah alamat *file* yang berisi desain *layout* dari aplikasi yang berformat html.

Kemudian kode baris 15 adalah perintah untuk mengekspor kelas HasilPage agar dapat digunakan oleh kelas lain yang membutuhkan layanannya, tentu saja kelas lain dapat dengan mudah melakukannya dengan perintah impor dengan menyebut nama HasilPage dengan cara seperti yang telah dijelaskan pada paragraf sebelumnya.

Pada kelas HasilPage sendiri terdapat beberapa aktivitas yang bisa dilihat, yakni rangkaian deklarasi variabel awal yang berfungsi sebagai tempat penyimpanan data sementara selama aplikasi berjalan. Deklarasi yang dilakukan antara lain variabel hasil, kesimpulan, cek15, biodata, akhirtest, dan cek yang mempunyai tipe data yang sama, yakni any kecuali variabel cek15 yang bertipe boolean dengan nilai false. Kode baris dari deklarasi ini dapat dilihat pada blok kode baris 17 sampai dengan 24.

Kemudian kelas HasilPage ini mempunyai sebuah konstruktor dengan parameter yang dibutuhkan adalah kelas public navCtrl: NavController, public navParams: NavParams, public modalCtrl: ModalController yang mana kelas-kelas yang dibutuhkan ini sudah dapat dipakai karena telah diimpor diawal kode tadi. Sifat dasar dari konstruktor ini adalah semua kode aplikasi yang terdapat di dalamnya, maka akan dijalankan pada saat kelas ini diinstansiasi atau penciptaan objek dari kelas ini. Kemudian blok baris kode 25 sampai dengan 34 akan dijalankan pertama

kali secara otomatis. Penjelasan algoritma dari blok baris kode 25 sampai dengan 34 adalah sebagai berikut.

- a. Variabel hasil diisi dengan hasil dari fungsi `this.navParams.get('hasil')` dengan parameter data bertipe string 'hasil' kemudian (baris ke 26)
- b. Variabel cek diisi dengan hasil dari fungsi `this.navParams.get('cek')` dengan parameter data bertipe string 'cek' kemudian panggil perintah untuk menulis log dari aplikasi dengan fungsi `console.log(this.hasil)` dengan parameter variabel hasil.(baris ke 27 - 28)
- c. Variabel biodata diisi dengan data yang diambil dari penyimpanan lokal dengan nama parameter string "biodata" yang telah melewati parsing format data json (baris ke 29)
- d. Menciptakan variabel baru dengan nama `tzoffset` yang mana diisi dengan hasil dari selisih waktu antara lokal dan zona UTC yang kemudian dikalikan dengan angka 60000 dengan maksud menjadikannya milidetik (baris ke 30)
- e. Menciptakan variabel baru dengan nama `localISOTime` yang mana diisi dengan hasil selisih antara waktu sekarang dikurangi dengan nilai `tzoffset` kemudian diubah menjadi format string dengan memilih sebagian dari data array indeks 0 dan -1 (baris ke 31)
- f. Variabel `akhirtest` diisi dengan hasil dari metode `replace` karakter "T" dengan "" pada variabel `localISOTime` (baris ke 32)
- g. Panggil fungsi `prosesKesimpulan(cek)` dengan parameter variabel `cek` (baris ke 33).

```
1 plate() {
2     let modal = this.modalCtrl.create(Modal GambarPage,
3         {hasil: this.hasil});
4     modal.present();
5 }
```

Gambar 4.27 *Source code* hasil tes Ishihara bagian 2 ts

Gambar 4.27 adalah potongan kode aplikasi lanjutan dari gambar sebelumnya, yakni Gambar 4.26 yang berformat typescript, di mana kode inilah yang menjalankan beberapa fungsi yang dipanggil untuk menjalankan fitur tertentu salah

satunya adalah fungsi `plate()`. Adapun penjelasan algoritma dari fungsi ini adalah mendeklarasikan variabel modal yang mana akan diisi dengan hasil penciptaan objek modal kontroler dengan parameter kelas `ModalGambarPage` dan variabel hasil, kemudian variabel modal menampilkan modal *plate* dengan perintah `present()`.

```

41   prosesKesimpulan(a) {
42     console.log(a);
43     if (a[1].res == 'Salah') {
44       // Jika Gambar 1 Salah, maka Hasil Buta Warna
45       Total
46       this.kesimpulan = 'BUTA WARNA TOTAL';
47     } else if(a[1].res == 'Benar') {
48       let salah1 = 0;
49       let salah2 = 0;
50       let salah3 = 0;
51       let benar1 = 0;
52       let benar2 = 0;
53       a.filter((b) => {
54         if (b != null) {
55           if (b.kode_gambar > 1 && b.kode_gambar <= 16)
56             {
57               //console.log(b);
58               if (b.res == 'Salah') {
59                 salah1++;
60                 if (salah1 > 3) {
61                   console.log("BWP 1");
62                   this.kesimpulan = 'BUTA WARNA PARSIAL';
63                 }
64               } else{
65                 benar1++;
66               }
67             }
68         }
69       });
70     }
71   }

```

Gambar 4.28 Source code hasil tes Ishihara bagian 3 ts

Gambar 4.28 adalah potongan kode aplikasi lanjutan dari gambar sebelumnya, yakni Gambar 4.26 yang berformat typescript, di mana kode inilah yang menjalankan beberapa fungsi yang dipanggil pada konstruktor salah satunya adalah fungsi `prosesKesimpulan(a)`. Adapun penjelasan algoritma dari fungsi ini adalah sebagai berikut.

- a. Tampilkan log konsol parameter variabel a
- b. Jika nilai dari variabel objek a pada indeks ke 1 bagian res sama dengan 'Salah', maka variabel kesimpulan diisi dengan nilai string 'BUTA WARNA TOTAL' (dengan arti jika gambar 1 salah, maka hasil Buta Warna Total)
- c. Selain itu jika res sama dengan 'Benar', maka kondisikan:
 - 1) Deklarasikan variabel salah1 yang mana diisi dengan nilai angka 0
 - 2) Deklarasikan variabel salah2 yang mana diisi dengan nilai angka 0
 - 3) Deklarasikan variabel salah3 yang mana diisi dengan nilai angka 0
 - 4) Deklarasikan variabel benar1 yang mana diisi dengan nilai angka 0
 - 5) Deklarasikan variabel benar2 yang mana diisi dengan nilai angka 0
 - 6) Variabel a menjalankan fungsi filter dengan tujuan memilah beberapa variabel yang diproses di dalamnya yang dimulai dengan menjalankan fungsi anonymous dengan parameter variabel b. Di dalam fungsi anonymous tersebut ada beberapa logika dan algoritma, antara lain sebagai berikut.
 - a) Jika variabel b tidak sama dengan null, maka kondisikan:
 - (1) Jika nilai properti dari variabel objek b, yakni kode_gambar lebih dari 1 dan kode_gambar kurang dari atau sama dengan 16, maka kondisikan:
 - (a) Jika nilai properti dari variabel objek b, yakni res sama dengan 'Salah', maka nilai variabel salah1 ditambahkan dengan nilai 1, kemudian jika nilai variabel salah1 lebih besar dari 3, maka Panggil perintah log konsol dengan parameter "BWP 1" kemudian nilai variabel kesimpulan diisi dengan nilai "BUTA WARNA PARSIAL"
 - (b) Selain itu nilai variabel benar ditambahkan dengan nilai 1

```

67     else if(b.kode_gambar == 22 || b.kode_gambar == 23 ||
68     b.kode_gambar == 24){
69         if(b.res == 'Salah'){
70             salah2++;
71             if(salah2 == 2){
72                 console.log("BWP 2");
73                 this.kesimpulan = 'BUTA WARNA PARSIAL';
74             }
75         }else{
76             benar2++;

```

```

76         if((benar1 != 16 || benar1 < 13) && benar2 >= 2){
77             console.log("MN 2");
78             this.kesimpulan = 'MATA NORMAL';
79         }
80     }
81 }
82     else if(b.kode_gambar == 18 || b.kode_gambar == 19 ||
83 b.kode_gambar == 20 || b.kode_gambar == 21){
84         if(b.res == 'Salah') {
85             salah3++;
86             if (salah3 == 4) {
87                 console.log("BWP 3");
88                 this.kesimpulan = 'BUTA WARNA PARSIAL';
89             }
90         }

```

Gambar 4.29 Source code hasil tes Ishihara bagian 3 ts

Gambar 4.29 adalah potongan kode aplikasi lanjutan dari gambar sebelumnya, dan lanjutan algoritmanya adalah:

- (2) Selain itu jika nilai properti dari variabel objek b, yakni kode_gambar sama dengan 22 atau kode_gambar sama dengan 23 atau kode_gambar sama dengan 24, maka kondisikan:
 - (a) Jika nilai properti dari variabel objek b, yakni res sama dengan 'Salah', maka nilai variabel salah2 ditambahkan dengan nilai 1 kemudian jika nilai variabel salah 2 sama dengan 2, maka jalankan perintah konsol log dengan parameter "BWP 2" kemudian nilai variabel kesimpulan dengan nilai string "BUTA WARNA PARSIAL"
 - (b) Selain itu, maka nilai variabel benar2 ditambahkan dengan nilai 1 kemudian jika nilai variabel benar1 tidak sama dengan nilai 16 atau nilai variabel benar1 kurang dari 13 dan jika nilai variabel benar2 lebih dari atau sama dengan nilai 2 maka panggil perintah log konsol dengan parameter "MN 2" kemudian nilai variabel kesimpulan diisi dengan string "MATA NORMAL"
- (3) Selain itu jika nilai properti dari variabel objek b, yakni kode_gambar sama dengan 18 atau jika kode_gambar sama dengan 19 atau jika kode_gambar sama dengan 20 atau jika kode_gambar sama dengan 21, maka kondisikan:

- (a) Jika nilai properti dari variabel objek b, yakni res sama dengan 'Salah', maka nilai variabel salah3 ditambahkan dengan nilai 1 kemudian jika nilai variabel salah3 sama dengan nilai 4, maka panggil perintah log konsol dengan parameter "BWP 3" kemudian nilai variabel kesimpulan diisi dengan nilai string "BUTA WARNA PARSIAL"

```

92         else if(b.kode_gambar > 1 && b.kode_gambar <= 17){
93             if(b.res == 'Benar'){
94                 benar1++;
95                 if(benar1 == 16 || benar1 > 13){
96                     console.log("MN 1");
97                     this.kesimpulan = 'MATA NORMAL';
98                 }
99             }
100         }

        console.log(salah1+"..." +salah2+"..." +salah3+"..." +benar1+"...
101 ."+benar2);
102     }
103 });
104     console.log(this.kesimpulan);
105 }
106 }
107
108 goHome() {
109     this.navCtrl.setRoot(HomePage);
110 }
111
112 }

```

Gambar 4.30 Source code hasil tes Ishihara bagian 4 ts

Gambar 4.30 adalah potongan kode aplikasi lanjutan dari gambar sebelumnya, dan lanjutan algoritmanya adalah:

- (4) Selain itu jika nilai properti dari variabel objek b, yakni kode_gambar lebih dari 1 dan jika kode_gambar kurang dari atau sama dengan 17, maka kondisikan:
- (a) Jika nilai properti dari variabel objek b, yakni res sama dengan 'Benar', maka nilai variabel benar1 ditambahkan nilai 1 jika nilai benar1 sama dengan 16 atau jika benar1 lebih besar 13 maka panggil perintah konsol log dengan parameter

string nilai “MN 1” kemudian nilai variabel kesimpulan diisi dengan string nilai “MATA NORMAL”

(5) Panggil perintah konsol log dengan parameter variabel kesimpulan

Kemudian pada kelas Hasil ini terdapat pula fungsi goHome() untuk menentukan root halaman. Fungsi goHome() ini memanfaatkan variabel navCtrl dengan fungsi setRoot dengan parameter objek kelas yang akan dituju. Dalam hal ini objek kelas yang digunakan sebagai parameter adalah HomePage. Jadi ketika seorang pengguna akan kembali ke halaman Home ketika dia mengnekan tombol yang mempunyai fungsi goHome() ini;



Gambar 4.31 Desain modal hasil tes Ishihara

Gambar 4.31 adalah desain *mockup* desain modal hasil tes Ishihara. Desain ini masih menggunakan konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain bar navigasi yang mempunyai dua elemen, yakni judul bertuliskan plate uji yang berfungsi memberikan informasi layar yang aktif saat ini

dan tombol tutup dengan desain *flat* yang berfungsi untuk menutup halaman. Pada blok desain berikutnya adalah tampilan *plate* dari hasil tes buta warna di mana akan ada sebanyak 24 gambar sesuai dengan banyaknya soal pada saat tes.

Alur kerja dari desain ini adalah jika pengguna menekan tombol tutup aplikasi akan kembali ke halaman hasil tes seperti pada Gambar 4.21.

```
1 <i on-header>
2   <i on-navbar color="primary">
3     <i on-title>Plate Uji</i on-title>
4     <i on-buttons end>
5       <button i on-button (click)="dismiss()">TUTUP</button>
6     </i on-buttons>
7   </i on-navbar>
8 </i on-header>
9 <i on-content>
10  <i on-card *ngFor="let item of hasil">
11    <i on-card-header>Plate Uji ke - {{ item.no }}</i on-card-
header>
12    <i on-card-content style="text-align: center;">
13      
14    </i on-card-content>
15  </i on-card>
16 </i on-content>
17
```

Gambar 4.32 Source code hasil tes Ishihara html

Gambar 4.32 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Blok baris 1 sampai dengan 8 adalah kode yang membangun informasi status halaman yang aktif dan elemen tombol tutup. Kemudian blok baris blok baris 9 sampai 16 adalah kode aplikasi yang membangun daftar elemen gambar hasil tes buta warna. Perlu dicermati bahwa kode pada baris 10 adalah perulangan untuk menampilkan seluruh data hasil tes pada konteks ini, maka data berada pada variable item, sehingga untuk menampilkan gambar, maka dibutuhkan alamat atau url daari gambar tersebut. Url ini dapat diperoleh dari variabel item.url pada kode potongan kode aplikasi {{ item.url }}.

```
1 import { Component } from '@angular/core';
2 import { NavController, NavParams, ViewController } from
  'ionic-angular';
3
4 /*
5   Generated class for the Modal Gambar page.
6
7   See
8   http://ionicframework.com/docs/v2/components/#navigation for
9   more info on
10  Ionic pages and navigation.
11 */
12 @Component({
13   selector: 'page-modal-gambar',
14   templateUrl: 'modal-gambar.html'
15 })
16 export class ModalGambarPage {
17
18   hasil: any;
19
20   constructor(public navCtrl: NavController, public
21     navCtrl: NavParams, public viewCtrl: ViewController) {
22     this.hasil = this.navCtrl.get('hasil');
23     console.log(this.hasil);
24   }
25
26   di smi ss(){
27     this.viewCtrl.di smi ss();
28   }
29 }
```

Gambar 4.33 Source code modal hasil tes Ishihara ts

Gambar 4.33 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk memproses tampilan gambar dari hasil tes. Kelas ModalGambarPage ini mempunyai variabel hasil (baris ke 16) yang bertugas untuk menyimpan atau menampung sementara hasil tes. Kemudian terdapat juga sebuah konstruktor dengan parameter public navCtrl: NavController, public navCtrl: NavParams, public viewCtrl: ViewController. Kelas-kelas yang dibutuhkan oleh konstruktor yang digunakan sebagai parameter didapatkan

dari impor paket lain yang dijalankan pada blok kode baris 1 sampai dengan 2. Saat kelas ini digunakan oleh kelas lain, maka secara otomatis semua metode atau aktivitas yang berada dalam konstruktor akan dijalankan, dalam konteks ini kelas ModalGambarPage akan menjalankan kode baris 19 sampai dengan 20, yakni

```
this.hasil = this.navParams.get('hasil');  
console.log(this.hasil);
```

Penjelasan dari kode ini adalah variabel hasil akan diisi oleh data hasil dari fungsi `this.navParams()` dengan parameter 'hasil'. Kemudian kelas ini juga mempunyai fungsi `dismiss()` yang bertugas untuk membatalkan aksi saat ini.



Gambar 4.34 Desain deskripsi Metode Ishihara

Gambar 4.34 adalah desain *mockup* deskripsi Metode Ishihara. Desain ini masih menggunakan konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain bar navigasi yang mempunyai dua elemen, yakni judul bertuliskan Metode Ishihara yang berfungsi memberikan informasi layar yang aktif saat ini, dan tombol kembali berupa anak panah ke kiri dengan desain *flat* yang

berfungsi untuk menutup halaman. Pada blok desain berikutnya adalah tampilan deskripsi dari Metode Ishihara.

Alur kerja dari desain halaman ini adalah jika pengguna menekan tombol panah, maka aplikasi akan kembali ke halaman home seperti pada Gambar 4.6.

```

1 <i on-header>
2   <i on-navbar color="primary">
3     <i on-title>Metode Ishihara</i on-title>
4   </i on-navbar>
5 </i on-header>
6 <i on-content>
7   <i on-grid>
8     <i on-row>
9       <i on-col >
10        <p style="text-align: justify">Metode Ishihara
        di kembangkan oleh Dr. Shinobu Ishihara pada tahun 1917. Beliau
        merupakan seorang profesor dari University of Tokyo. Tes Metode
        Ishihara adalah tes yang digunakan untuk mendeteksi gangguan
        persepsi warna, berupa tabel warna khusus berupa gambar-gambar
        pseudoisokromatik yang disusun oleh titik-titik dengan
        kepadatan warna berbeda yang dapat dilihat dengan mata normal,
10 tapi tidak bisa dilihat oleh mata yang mengalami defisiensi
        sebagian warna. Tes buta warna Ishihara terdiri dari lembaran
        yang di dalamnya terdapat titik-titik dengan berbagai warna dan
        ukuran. Titik-titik berwarna tersebut disusun sehingga
        membentuk lingkaran yang di dalamnya terdapat titik-titik dengan
        pola membentuk angka maupun garis berkelok. Warna titik-titik
        itu dibuat sedemikian rupa sehingga orang buta warna tidak akan
        berhasil melihat angka maupun garis yang ada.</p>
11       </i on-col >
12     </i on-row>

```

```
13 </ion-grid>
14 </ion-content>
15
```

Gambar 4.35 *Source code* deskripsi Metode Ishihara html

Gambar 4.35 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi blok. Baris 1 sampai dengan 5 adalah kode yang membangun navigasi. Kemudian blok baris 6 sampai 14 adalah kode aplikasi yang membangun informasi detail mengenai deskripsi Metode Ishihara.

```
1 import { Component } from '@angular/core';
2 import { NavController, NavParams } from 'ionic-angular';
3
4 @Component({
5   selector: 'page-informasi',
6   templateUrl: 'informasi.html'
7 })
8 export class InformasiPage {
9
10  constructor(public navCtrl: NavController, public navParams:
11  NavParams) {}
12
13  ionViewDidLoad() {
14    console.log('ionViewDidLoad InformasiPage');
15  }
```

Gambar 4.36 *Source code* deskripsi Metode Ishihara ts

Gambar 4.36 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk memproses tampilan deskripsi

Metode Ishihara melalui konstruktor yang memanggil fungsi dialog dari bawaan framework ionic, yakni `ionViewDidLoad()`.



Gambar 4.37 Desain tentang aplikasi Ishihara

Gambar 4.37 adalah desain *mockup* tentang aplikasi Ishihara. Desain ini masih menggunakan konsep dasar yang sederhana yang mana dimaksudkan agar pengguna lebih mudah dalam berinteraksi dengan aplikasi. Pada desain tersebut beberapa elemen antara lain bar navigasi yang mempunyai dua elemen, yakni judul bertuliskan tentang aplikasi yang berfungsi memberikan informasi layar yang aktif saat ini dan tombol kembali berupa anak panah ke kiri dengan desain *flat* yang berfungsi untuk menutup halaman. Pada blok desain berikutnya adalah tampilan logo dan informasi tentang aplikasi, pembuat aplikasi dan tahun pembuatan aplikasi.

Alur kerja dari desain halaman ini adalah jika pengguna menekan tombol panah, maka aplikasi akan kembali ke halaman home seperti pada Gambar 4.6.

```

1 <i on-header>
2   <i on-navbar color="primary">
3     <i on-title>Tentang Aplikasi </i on-title>
4   </i on-navbar>
5 </i on-header>
6 <i on-content>
7   <i on-grid>
8     <i on-row>
9       <i on-col width-10>
10      </i on-col >
11      <i on-col width-80 style="text-align: center;" >
12        
14        <!--<i on-icon name="eye" style="font-
15        size: 150px; color: white;"></i on-icon-->
16      </i on-col >
17      <i on-col width-10>
18      </i on-col >
19    </i on-row>
20  </i on-grid>
21  <i on-grid>
22    <i on-row>
23      <i on-col >
24        <p style="text-align: justify">Tujan konsep
25        pengembangan aplikasi ini adalah untuk mendeteksi buta
26        warna menggunakan Metode Ishihara. Untuk pengguna
27        aplikasinya sendiri adalah masyarakat umum yang berusia
28        minimal 15 tahun dan sudah mengenal tentang smartphone.
29        Jenis aplikasi yang dikembangkan adalah interaktif,
30        karena pengguna dan aplikasi saling melakukan interaksi
31        atau terdapat respon antara aplikasi dan pengguna.
32        Aplikasi ini dikategorikan sebagai aplikasi kesehatan
33        karena dapat menentukan seseorang mengalami buta warna
34        atau tidak. Untuk spesifikasi android yang digunakan

```

dal am menj al ankan apl i kasi i ni mi ni mum androi d versi 4. 4+, ukuran apl i kasi tergan tung dari devi ce yang di gunakan.

```

23         <br/><br/><br/>
24         © Copyri ght 2017. N2d. </p>
25     </i on-col >
26 </i on-row>
27 </i on-gri d>
28 </i on-content>

```

Gambar 4.38 *Source code* tentang aplikasi Ishihara html

Gambar 4.38 adalah potongan kode aplikasi berformat html, di mana kode inilah yang membangun tampilan desain untuk navigasi. Blok baris 1 sampai dengan 5 adalah kode yang membangun navigasi. Kemudian blok baris 9 sampai dengan 15 adalah kode yang membangun tampilan logo aplikasi. Kemudian blok baris blok baris 20 sampai 26 adalah kode aplikasi yang membangun konten tentang aplikasi.

```

1  import { Component } from '@angular/core' ;
2  import { NavController, NavParams } from 'ionic-angular' ;
3
4  @Component({
5  selector: 'page-tentang' ,
6  templateUrl: 'tentang.html '
7  })
8  export class TentangPage {
9
10 constructor(public navCtrl: NavController, public navParams:
    NavParams) {}
11

```

```
12  ionViewDidLoad() {  
13  console.log('ionViewDidLoad TentangPage');  
14  }  
15  
16  }
```

Gambar 4.39 *Source code* deskripsi aplikasi Ishihara ts

Gambar 4.39 adalah potongan kode aplikasi berformat typescript, di mana kode inilah yang menjalankan interaksi untuk memproses tampilan deskripsi Metode Ishihara melalui konstruktor yang memanggil fungsi dialog dari bawaan framework ionic yakni `ionViewDidLoad()`.



Gambar 4.40 Proses *building* aplikasi

Gambar 4.40 adalah potongan kode aplikasi *command line* dari konsol berupa perintah *native* dari framework ionic yang bertujuan untuk membangun aplikasi menjadi sebuah *file* apk yang siap dipasang ke dalam sistem operasi android. Langkah pertama yang harus dilakukan adalah masuk ke dalam *folder* proyek

D:\Mobile Projects\butawarna aplikasi kemudian ketik perintah `ionic build android` kemudian dilanjutkan dengan menekan tombol ENTER, maka ionic akan menjalankan perintah tersebut dan selanjutnya tinggal menunggu sampai proses pembangunan aplikasi selesai.

Setelah proses pembangunan selesai, maka *file* apk siap untuk diuji dan didistribusikan. *File* ini bisa ditemukan di dalam *folder* `D:/Mobile Projects/butawarna/platforms/android/build/outputs/apk/android-debug.apk`



Gambar 4.41 Contoh simulasi *real mockup* pada *device mobile*

Gambar 4.41 adalah contoh simulasi *real mockup* pada *device mobile*. Saat aplikasi dipasang pada sebuah *smartphone* berbasis android, maka *prototype*-nya adalah seperti yang ada pada gambar.

4.4 Pengujian White Box

Dalam penelitian ini penulis menggunakan *cyclomatic complexity* untuk mengetahui suatu kompleksitas dari aplikasi yang dirancang. Berikut ini adalah kompleksitas dari aplikasi tes buta warna Metode Ishihara berbasis android.

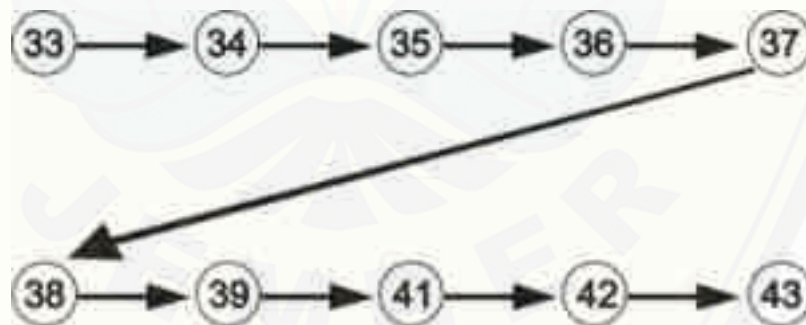
```

33  firstload(){
34      if(this.pos == 0){
35          this.jawaban = '';
36          this.pos += 1;
37          this.nm = this.randomGambar();
38          this.nm >= 18 ? this.img = "assets/pola" + this.nm +
".jpg" : this.img = "assets/angka" + this.nm + ".jpg";
39          this.waktu = 10;
40          clearInterval(this.timer);
41          this.countdown();
42      }
43  }

```

Gambar 4.42 Kode fungsi firstload pada file test.ts

Kode di atas merupakan bagian dari file test.ts. Berikut ini adalah *complexity* dari kode tersebut.



Gambar 4.43 Kompleksitas dari kode pada Gambar 4.42

Hasil dari kompleksitas dari alur Gambar 4.43 bernilai 1. Nominal tersebut didapatkan dari mengurangi jumlah tanda panah, dengan jumlah lingkaran, dijumlahkan dengan 2. Jadi jumlah tanda panah dikurangi dengan jumlah lingkaran bernilai -1. Kemudian nilai tersebut dijumlahkan dengan 2, maka akan memiliki hasil akhir yang bernilai 1.

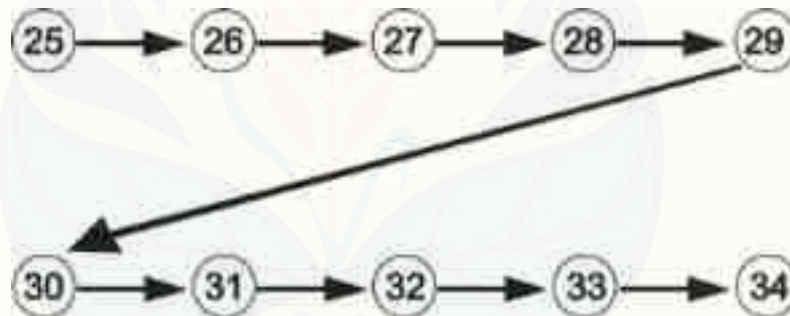
```

constructor(public navCtrl: NavController, public navParams:
25 NavParams, public modalCtrl: ModalController) {
26   this.hasil = this.navParams.get('hasil');
27   this.cek = this.navParams.get('cek');
28   console.log(this.hasil);
   this.biodata =
29   JSON.parse(window.localStorage.getItem('biodata'));
   var tzoffset = (new Date()).getTimezoneOffset() * 60000;
30   //offset in milliseconds
   var localISOTime = (new Date(Date.now() -
31   tzoffset)).toISOString().slice(0, -1);
32   this.akhirstest = localISOTime.replace("T", " ");
33   this.prosesKesimpulan(this.cek);
34 }

```

Gambar 4.44 Kode fungsi constructor pada file hasil.ts

Kode di atas merupakan bagian dari file hasil.ts. Berikut ini adalah *complexity* dari kode tersebut.



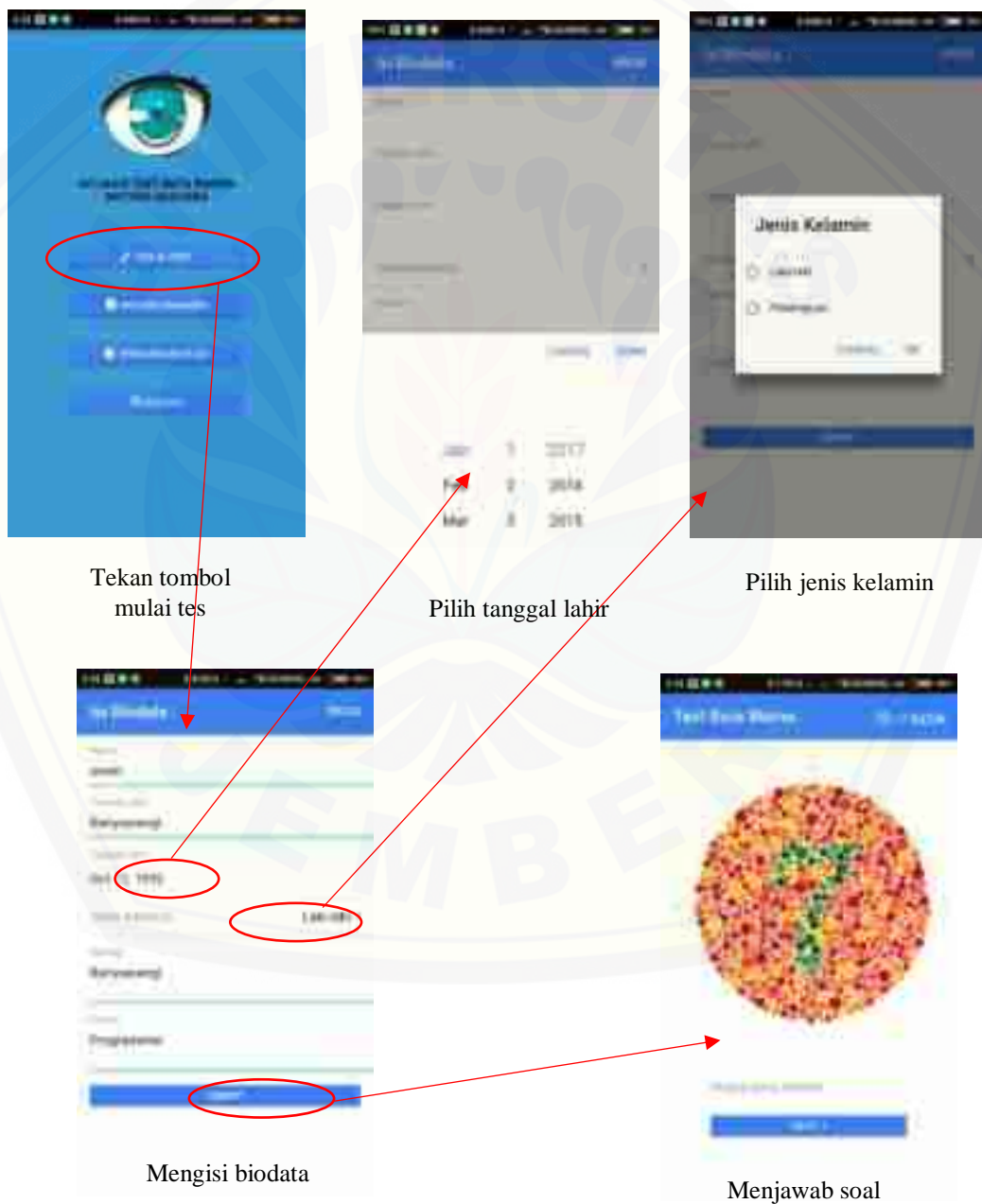
Gambar 4.45 Kompleksitas dari kode pada Gambar 4.44

Hasil dari kompleksitas dari alur Gambar 4.45 bernilai 1. Nominal tersebut didapatkan dari mengurangi jumlah tanda panah, dengan jumlah lingkaran, dijumlahkan dengan 2. Jadi jumlah tanda panah dikurangi dengan jumlah lingkaran bernilai -1. Kemudian nilai tersebut dijumlahkan dengan 2, maka akan memiliki hasil akhir yang bernilai 1. Alur aplikasi yang memiliki nilai kompleksitas 1, maka aplikasi tersebut dapat dikatakan sederhana.

4.5 Pengujian *Black Box*

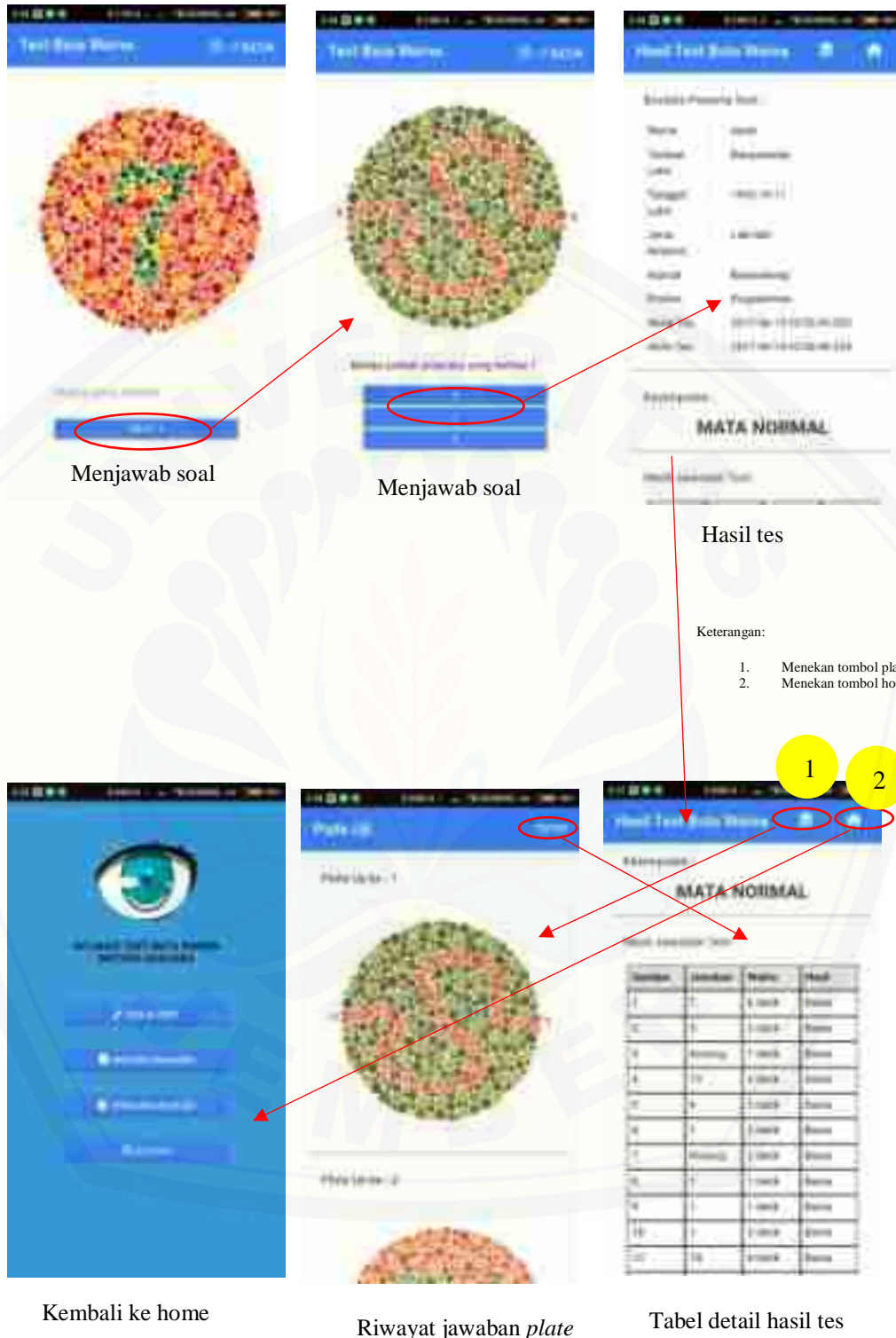
Black box testing menguji perangkat lunak dari segi spesifikasi fungsional, tanpa menguji desain dan kode aplikasi. Pengujian dimaksudkan untuk mengetahui apakah fungsi – fungsi masukan dan keluaran perangkat lunak berjalan sesuai dengan spesifikasi yang dibutuhkan. Berikut ini adalah gambar pengujian aplikasi tes buta warna Metode Ishihara berbasis android dengan *black box testing*.

Memulai Tes Ishihara



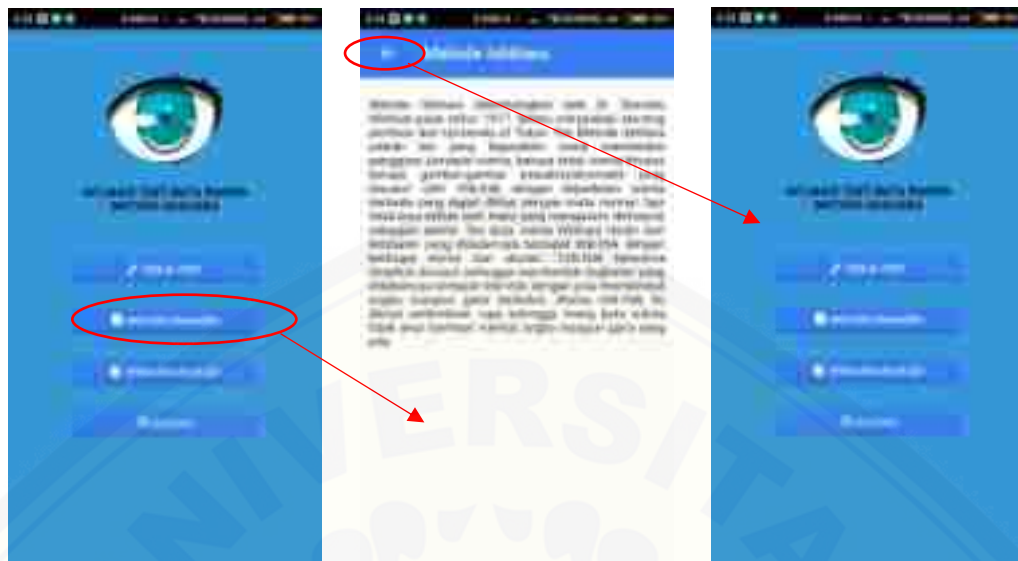
Gambar 4.46 Pengujian *black box* aplikasi tes buta warna Ishihara bagian 1

Lanjutan dari bagian 1



Gambar 4.47 Pengujian *black box* aplikasi tes buta warna Ishihara bagian 2

Menampilkan informasi Metode Ishihara



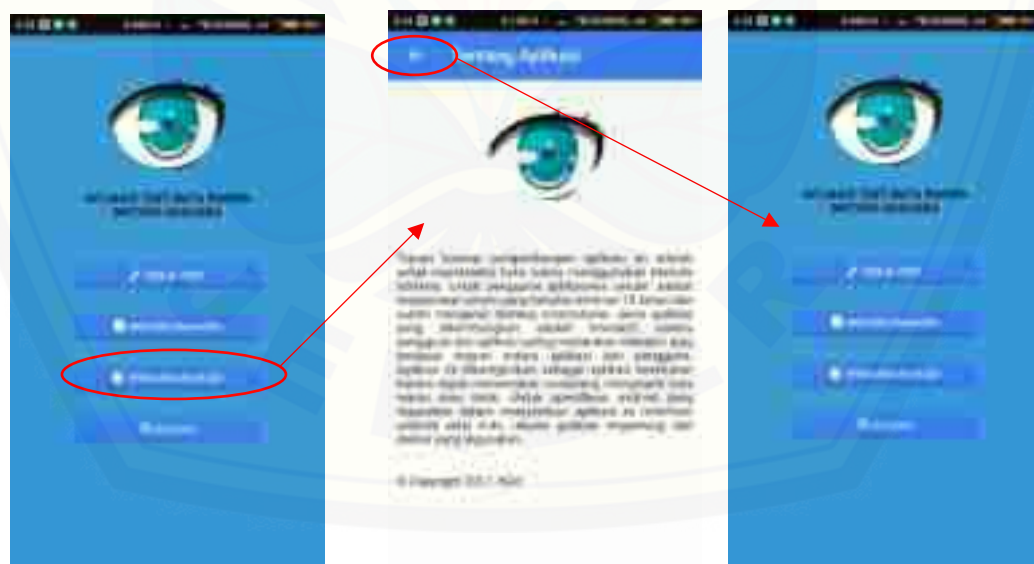
Tekan tombol Metode Ishihara

Informasi Metode Ishihara, Tekan tombol anak panah

Kembali ke home

Gambar 4.48 Pengujian *black box* informasi Metode Ishihara

Menampilkan informasi Metode Ishihara



Tekan tombol tentang aplikasi

Informasi tentang aplikasi, Tekan tombol anak panah

Kembali ke home

Gambar 4.49 Pengujian *black box* informasi tentang aplikasi

Ada beberapa kriteria yang dapat digunakan untuk mengetahui instrumen yang digunakan sudah tepat untuk mengukur apa yang ingin diukur, yaitu:

- Jika koefisien korelasi *product moment* melebihi 0,3.
- Jika koefisien korelasi *product moment* $> r_{tabel} (\alpha ; n - 2)$, n = jumlah sampel.
- Nilai sig. $\leq \alpha$

Rumus yang digunakan untuk uji validitas dengan teknik korelasi *product moment* (r_{hitung}) mengacu pada Persamaan 1. Kemudian berdasarkan Tabel 4.2 didapatkan hasil seperti Tabel 4.3.

Tabel 4.3 Perhitungan validitas 1

X	ΣX	$(\Sigma X)^2$	ΣY	$(\Sigma Y)^2$
1	20	400		
2	19	361		
3	18	324		
4	20	400		
5	19	361		
6	20	400		
7	18	324		
8	19	361		
9	20	400		
10	19	361		
11	20	400	466	217156
12	20	400		
13	20	400		
14	20	400		
15	20	400		
16	18	324		
17	19	361		
18	18	324		
19	19	361		
20	20	400		
21	20	400		

22	20	400		
23	20	400		
24	20	400		

Berdasar Tabel 4.2 didapatkan hasil seperti Tabel 4.4.

Tabel 4.4 Perhitungan validitas 2

n	XY																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
2	19	19	0	19	19	19	0	19	19	19	19	19	19	19	19	0	0	0	19	19	19	19	19	19
3	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
4	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
5	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
6	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
7	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
8	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	0	23	23	23	23	23
9	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
10	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
11	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
12	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
13	16	0	0	16	0	16	0	0	16	0	16	16	16	16	16	0	16	0	16	16	16	16	16	16
14	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
15	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
16	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
17	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
18	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
19	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
20	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24

Berdasar Tabel 4.4 diperoleh hasil seperti Tabel 4.5.

Tabel 4.5 Perhitungan validitas 3

XY	ΣXY
1	466
2	450
3	431
4	466

11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
13	1	0	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	256
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	576

Berdasarkan Tabel 4.6 diperoleh hasil sesuai Tabel 4.7

Tabel 4.7 Perhitungan validitas 5

X	ΣX^2	ΣY^2
1	20	10938
2	19	
3	18	
4	20	
5	19	
6	20	
7	18	
8	19	
9	20	
10	19	
11	20	
12	20	
13	20	
14	20	
15	20	
16	18	
17	19	
18	18	
19	19	
20	20	

21	20
22	20
23	20
24	20

Berdasarkan Tabel 4.3, Tabel 4.5, Tabel 4.7 didapatkan hasil sesuai Tabel 4.8.

Tabel 4.8 Perhitungan validitas 6

Plate	Rumus								
	n	ΣX	ΣY	ΣXY	ΣX^2	ΣY^2	$(\Sigma X)^2$	$(\Sigma Y)^2$	r_{hitung}
1	20	20	466	466	20	10938	400	217156	1
2	20	19	466	450	19	10938	361	217156	0,836
3	20	18	466	431	18	10938	324	217156	0,965
4	20	20	466	466	20	10938	400	217156	1
5	20	19	466	450	19	10938	361	217156	0,836
6	20	20	466	466	20	10938	400	217156	1
7	20	18	466	431	18	10938	324	217156	0,965
8	20	19	466	450	19	10938	361	217156	0,836
9	20	20	466	466	20	10938	400	217156	1
10	20	19	466	450	19	10938	361	217156	0,836
11	20	20	466	466	20	10938	400	217156	1
12	20	20	466	466	20	10938	400	217156	1
13	20	20	466	466	20	10938	400	217156	1
14	20	20	466	466	20	10938	400	217156	1
15	20	20	466	466	20	10938	400	217156	1
16	20	18	466	431	18	10938	324	217156	0,965
17	20	19	466	447	19	10938	361	217156	0,492
18	20	18	466	431	18	10938	324	217156	0,965
19	20	19	466	443	19	10938	361	217156	0,034
20	20	20	466	466	20	10938	400	217156	1
21	20	20	466	466	20	10938	400	217156	1

22	20	20	466	466	20	10938	400	217156	1
23	20	20	466	466	20	10938	400	217156	1
24	20	20	466	466	20	10938	400	217156	1

Pengujian menggunakan taraf signifikan (α) 5%, dan dibandingkan dengan nilai-nilai r *product moment* pada Tabel 2.1.

Dalam hal ini diperoleh $r_{tabel} (\alpha ; n - 2)$

$$n = 20$$

$$\alpha = 5\%$$

$$\text{Maka, nilai } r_{tabel} = 20 - 2 = 18$$

Jadi, r_{tabel} 18 dengan taraf signifikan 5% dari tabel *product moment* adalah 0,468

Tabel 4.9 Hasil validitas

<i>Plate</i>	Korelasi Product Moment	Hubungan dengan $r_{tabel} = 0,468$	Validitas
1	1	Lebih dari	Valid
2	0,836	Lebih dari	Valid
3	0,965	Lebih dari	Valid
4	1	Lebih dari	Valid
5	0,836	Lebih dari	Valid
6	1	Lebih dari	Valid
7	0,965	Lebih dari	Valid
8	0,836	Lebih dari	Valid
9	1	Lebih dari	Valid
10	0,836	Lebih dari	Valid
11	1	Lebih dari	Valid
12	1	Lebih dari	Valid
13	1	Lebih dari	Valid
14	1	Lebih dari	Valid
15	1	Lebih dari	Valid
16	0,965	Lebih dari	Valid
17	0,492	Lebih dari	Valid

Berdasarkan Lampiran C.2 didapatkan hasil sesuai Tabel 4.11.

Tabel 4.11 Data tes kedua buta warna 20 responden

n	Plate																								Y	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
2	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	17
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
13	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	17
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	24

Tahapan perhitungan uji reliabilitasnya, yaitu:

- a. Membuat hipotesis dalam bentuk kalimat

Ho: Pengukuran pertama dan pengukuran kedua tidak konsisten (tidak *reliable*)

Ha: Pengukuran pertama dan pengukuran kedua konsisten (*reliable*)

- b. Menentukan taraf signifikan (α)

Taraf signifikan adalah seberapa besar menerima hipotesis yang salah

- c. Kaidah pengujian

Jika $r_{hitung} \leq r_{tabel}$ maka Ho diterima

Jika $r_{hitung} > r_{tabel}$ maka Ho ditolak

- d. Menghitung r_{hitung} dan r_{tabel}

- 1) Menghitung r_{hitung} mengacu pada Persamaan 1

- 2) Menentukan nilai koefisien korelasi (r_{tabel})

Nilai r_{tabel} dapat dilihat di tabel *Product Moment* dengan ketentuan $r_{(a,n-2)}$

5) Membandingkan r_{tabel} dan r_{hitung}

Tujuan membandingkan r_{hitung} dan r_{tabel} adalah untuk mengetahui hipotesis mana yang akan diterima berdasarkan kaidah pengujian

6) Membuat keputusan

Menerima atau menolak H_0

Berdasarkan Tabel 4.10 dan Tabel 4.11 didapatkan sesuai Tabel 4.12.

Tabel 4.12 Data tabulasi 20 responden tes pertama dan kedua

Responden (n)	Tes Pertama (X)	Tes Kedua (Y)
1	24	24
2	19	17
3	24	24
4	24	24
5	24	24
6	24	24
7	24	24
8	23	24
9	24	24
10	24	24
11	24	24
12	24	24
13	16	17
14	24	24
15	24	24
16	24	24
17	24	24
18	24	24
19	24	24
20	24	24

Berdasarkan Tabel 4.12 didapatkan hasil sesuai Tabel 4.13.

Tabel 4.13 Penolong uji reliabilitas

Responden	X	Y	XY	X ²	Y ²
1	24	24	576	576	576
2	19	17	323	361	289
3	24	24	576	576	576
4	24	24	576	576	576
5	24	24	576	576	576
6	24	24	576	576	576
7	24	24	576	576	576
8	23	24	552	529	576
9	24	24	576	576	576
10	24	24	576	576	576
11	24	24	576	576	576
12	24	24	576	576	576
13	16	17	272	256	289
14	24	24	576	576	576
15	24	24	576	576	576
16	24	24	576	576	576
17	24	24	576	576	576
18	24	24	576	576	576
19	24	24	576	576	576
20	24	24	576	576	576
Σ	466	466	10939	10938	10946

Berdasarkan Tabel 4.13, lalu menghitung r_{hitung} dengan menggunakan Persamaan 1.

$$r_{hitung} = \frac{20(10939) - (466)(466)}{\sqrt{[20(10938) - (466)^2][20(10946) - (466)^2]}}$$

$$r_{hitung} = 0,965$$

Pengujian menggunakan taraf signifikan (α) 5%, dan dibandingkan dengan nilai-nilai r *product moment* pada Tabel 2.1.

Dalam hal ini diperoleh $r_{tabel} (\alpha ; n - 2)$

$$n = 20$$

$$\alpha = 5\%$$

$$\text{Maka, nilai } r_{tabel} = 20 - 2 = 18$$

Jadi, r_{tabel} 18 dengan taraf signifikan 5% dari tabel *product moment* adalah 0,468

Membandingkan r_{tabel} dan r_{hitung} :

$$r_{hitung} = 0,965$$

$$r_{tabel} = 0,468$$

Ternyata $r_{hitung} = 0,965 > r_{tabel} = 0,468$, sehingga H_0 ditolak.

Karena nilai $r_{hitung} > r_{tabel}$, maka keputusannya tes pertama dan tes kedua konsisten (*reliable*).

BAB 6. PENUTUP

Bab ini merupakan bagian akhir dari penulisan skripsi, berisi kesimpulan dan saran. Kesimpulan yang ditulis adalah sebuah pernyataan dari hasil penulisan yang telah dilakukan, sedangkan saran adalah anjuran atau opini yang dapat dijadikan bahan untuk pengembangan lanjutan pada penulisan selanjutnya.

6.1 Kesimpulan

Adapun kesimpulan pada penelitian ini antara lain:

- a. Pembuatan aplikasi tes buta warna Metode Ishihara berbasis android dapat dirancang dengan menggunakan metode pengembangan perangkat lunak multimedia Luther. Hal ini dikarenakan kesesuaian konsep dari Metode Ishihara yang informasinya diberikan secara sekuensial dimulai dari satu halaman ke halaman lain dengan desain *mockup* yang telah dibuat pada tahap *design* dalam Metode Luther yang didasarkan pada ciri khas aplikasi-aplikasi yang berbasis android, yakni terdapatnya menu-menu dalam satu halaman kemudian menu tersebut akan mengantarkan penggunaannya menuju halaman lain.
- b. Aplikasi tes buta warna Metode Ishihara berbasis android terbukti valid dan reliabel, yang artinya mampu mendeteksi buta warna. Ini dibuktikan dengan pengujian pada 20 responden menggunakan metode pengujian validitas dan reliabilitas. Berdasarkan hasil uji yang diperoleh tersebut, untuk pengujian validitas terdapat dua responden yang mengalami buta warna parsial, sedangkan delapan belas adalah mata normal. Kemudian untuk pengujian reliabilitas dilakukan sebanyak dua kali. Pada pengujian reliabilitas pertama terdapat dua responden yang mengalami buta warna parsial dan delapan belas adalah mata normal. Selanjutnya untuk pengujian reliabilitas kedua hasilnya terdapat dua orang buta warna parsial dan delapan belas adalah mata normal. Lalu, dari kedua hasil uji reliabilitas tersebut dianalisis dan hasilnya menunjukkan bahwa nilai $r_{hitung} > r_{tabel}$, yang berarti pengujian pertama dan kedua reliabel.

6.2 Saran

- a. Pada aplikasi tes buta warna menggunakan Metode Ishihara berbasis android yang telah dikembangkan ini, hasil tes yang ditampilkan masih sebatas tampilan saja. Oleh karena itu pengembangan lebih lanjut untuk aplikasi ini dapat menambahkan fitur simpan dalam bentuk *file* dengan *extension* PDF. Sehingga pengguna dapat mencetak hasil tes ke dalam kertas atau digunakan untuk keperluan lain sesuai kebutuhan pengguna dengan mudah.
- b. Hal lain yang bisa dilakukan untuk penelitian lebih lanjut, yaitu untuk pengujiannya dapat melakukan pemilihan populasi responden yang mampu menjangkau seluruh kategori buta warna menurut Metode Ishihara, yakni mata normal, buta warna parsial, dan buta warna total.

DAFTAR PUSTAKA

- Agusta, S., et al. 2012. "Instrumen Pengujian Buta Warna Otomatis." *Jurnal Ilmiah Elite Elektro* (Universitas Indonesia) 3.
- Arthur, H. 1996. *A Testing Methodology Using The Cyclomatic Complexity Metric*. Gaithersburg: National Institute of Standard and Technology.
- Emerson, M.Cruz. et al. 2010. "Prevalence of Color-Vision Deficiency Among Male High-School Students." *Philippine Journal of Ophthalmology* (Department of Ophthalmology) 35.
- Ganong, William F. 2003. *Buku Ajar Fisiologi Kedokteran*. Jakarta: EGC.
- Guyton, A. C., dan J. E. Hall. 1997. *Buku Ajar Fisiologi Kedokteran*. Jakarta: EGC.
- Ilyas, Sidarta. 2012. *Ilmu Penyakit Mata*. Jakarta: Fakultas Kedokteran Universitas Indonesia.
- Ishihara, Shinobu. 1972. *Test for Colour-Blindness*. Tokyo: Kanehara Shuppan Co. Ltd.
- Juhara, Zamrony P. 2016. *Panduan Lengkap Pemrograman Android*. Yogyakarta: Andi.
- Luther, A. C. 1994. *Authoring Interactive Multimedia*. Massachusetts: Academic Press,.
- Murti, Hari & Santi, Rina Candra Noor. 2011. "Aplikasi Pendiagnosa Kebutaan Warna dengan Menggunakan Pemrograman Borland Delphi." *Jurnal Teknologi Informasi DINAMIK* 16: 160-170.
- Nugroho, Taufan. 2013. *Buta Warna dan Strabismus (Mata Juling)*. Yogyakarta: Nuha Medika.
- Pressman, Roger S. 1997. *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku I)*. Yogyakarta: Andi.
- Riduwan, dan E. A. Kuncoro. 2014. *Cara Menggunakan dan Memaknai Path Analysis (Analisis Jalur)*. Bandung: Alfabeta.
- Rokhim, Akhmad Nur. 2015. *Mengenal Tes Buta Warna*. Surabaya: Rona Publishing.
- Rosa, A. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bogor: Modula.



- Safaat, Nazruddin. 2015. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Siregar, Syofian. 2013. *Metode Penelitian Kuantitatif Dilengkapi Dengan Perbandingan Perhitungan Manual & SPSS*. Jakarta: Prenadamedia Group.
- Sugiyono. 2016. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- . 2004. *Metode Penelitian Kualitatif dan R&D*. Jakarta: Alfabet.
- Sutopo, Ariesto Hadi. 2003. *Multimedia Interaktif Dengan Flash*. Yogyakarta: Graha Ilmu.
- Widianingsih, Ratri. Kridalaksana, Awang Harsa. Hakim, Ahmad Rofiq. 2010. "Aplikasi Tes Buta Warna Dengan Metode Ishihara Berbasis Komputer." *Jurnal Informatika Mulawarman* 5: 36.
- Yanuarita, Andri. 2012. *Tes Buta Warna*. Yogyakarta: Rona Publishing.
- Zulganef. 2006. *Pemodelan Persamaan Struktural & Aplikasinya Menggunakan Amos 5*. Bandung: Pustaka.






LAMPIRAN





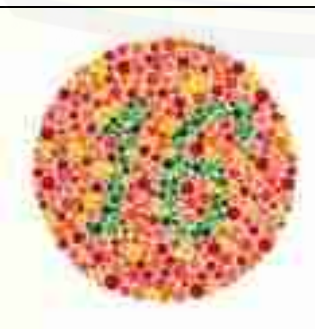
Pada bagian ini dijelaskan tentang lampiran yang memuat uraian secara rinci tentang dokumentasi pendukung yang tidak disampaikan pada bagian utama penelitian karena dapat mengganggu kontinuitas pembaca.

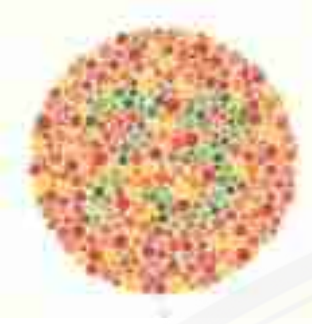



A. Penjelasan *Plate* Ishihara



Plate Ishihara yang digunakan dalam penelitian ini memakai 24 *plate* untuk mengetahui apakah seseorang mengalami buta warna atau tidak. Terdapat *plate* yang menunjukkan angka dan pola tertentu. Kemudian pengguna diminta untuk mengidentifikasi *plate* tersebut. Hal ini sesuai dengan buku tes buta warna Metode Ishihara yang diantaranya jika *plate* tersebut dilihat oleh orang normal, buta warna parsial, dan buta warna total akan tampak berbeda. Penjelasan mengenai *plate* Ishihara yang digunakan dijelaskan sebagai berikut.



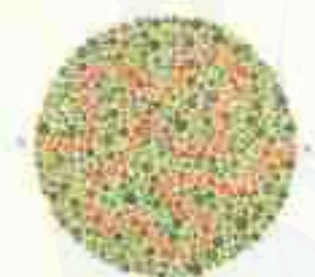

<i>PLATE</i>	KETERANGAN
	<p>Semua orang mampu melihat angka 12 termasuk mereka dengan buta warna total. Jika seseorang mengatakan mereka tidak bisa melihat sesuatu atau melihat sesuatu yang lain, maka mereka berbohong</p>
	<ol style="list-style-type: none"> a. Mereka dengan mata normal melihat angka 8 b. Mereka dengan buta warna parsial melihat angka 3 c. Mereka dengan buta warna total tidak akan melihat apapun



 A circular Ishihara color blindness test plate. The background is composed of green and black dots. The number 29 is formed by red and yellow dots.	<ul style="list-style-type: none">a. Mereka dengan mata normal melihat angka 29b. Mereka dengan buta warna parsial melihat angka 70c. Mereka dengan buta warna total tidak melihat apa-apa
 A circular Ishihara color blindness test plate. The background is composed of orange and brown dots. The number 5 is formed by green and blue dots.	<ul style="list-style-type: none">a. Mereka dengan mata normal melihat angka 5b. Mereka dengan buta warna parsial melihat angka 2c. Mereka dengan buta warna total tidak melihat apa-apa
 A circular Ishihara color blindness test plate. The background is composed of orange and brown dots. The number 3 is formed by green and blue dots.	<ul style="list-style-type: none">a. Mereka dengan mata normal melihat angka 3b. Mereka dengan buta warna parsial melihat angka 5c. Mereka dengan buta warna total tidak melihat apa-apa
 A circular Ishihara color blindness test plate. The background is composed of orange and brown dots. The number 15 is formed by green and blue dots.	<ul style="list-style-type: none">a. Mereka dengan mata normal melihat angka 15b. Mereka dengan buta warna parsial melihat angka 17c. Mereka dengan buta warna total tidak melihat apa-apa
 A circular Ishihara color blindness test plate. The background is composed of orange and brown dots. The number 74 is formed by green and blue dots.	<ul style="list-style-type: none">a. Mereka dengan mata normal melihat angka 74b. Mereka dengan buta warna parsial melihat angka 21c. Mereka dengan buta warna total tidak melihat apa-apa

	<p>a. Mereka dengan mata normal melihat angka 6</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>
	<p>a. Mereka dengan mata normal melihat angka 45</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>
	<p>a. Mereka dengan mata normal melihat angka 5</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>
	<p>a. Mereka dengan mata normal melihat angka 7</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>
	<p>a. Mereka dengan mata normal melihat angka 16</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>

	<p>a. Mereka dengan mata normal melihat angka 73</p> <p>b. Mereka dengan buta warna parsial dan buta warna total tidak melihat apa-apa</p>
	<p>a. Orang dengan mata normal atau buta warna total tidak bisa melihat angka berapapun</p> <p>b. Mereka dengan buta warna parsial melihat angka 5</p>
	<p>a. Orang dengan mata normal atau buta warna total tidak bisa melihat angka berapapun</p> <p>b. Mereka dengan buta warna parsial melihat angka 45</p>
	<p>a. Mereka dengan mata normal melihat angka 26</p> <p>b. Orang dengan buta warna merah kuat melihat angka 6, orang dengan buta warna merah ringan melihat angka 6, serta juga secara samar melihat angka 2</p> <p>c. Orang dengan buta warna hijau kuat melihat angka 2, orang dengan buta warna hijau ringan melihat angka 2, serta juga secara samar melihat angka 6</p>

	<p>d. Orang dengan buta warna total tidak bisa melihat angka berapapun</p>
	<p>a. Mereka dengan mata normal melihat angka 42</p> <p>b. Orang dengan buta warna merah kuat melihat angka 2, Orang dengan buta warna merah ringan melihat angka 2, serta juga secara samar melihat angka 4</p> <p>c. Orang dengan buta warna hijau kuat melihat angka 4, orang dengan buta warna hijau ringan melihat angka 4 serta juga secara samar melihat angka 2</p> <p>d. Orang dengan buta warna total tidak bisa melihat angka berapapun</p>
	<p>a. Mereka dengan mata normal dapat melihat 2 garis ungu dan merah</p> <p>b. Mereka dengan buta warna merah kuat dapat melihat garis ungu, mereka dengan buta warna merah ringan dapat melihat garis ungu, secara samar kemungkinan bisa melihat garis merah</p> <p>c. Mereka dengan buta warna hijau kuat dapat melihat garis merah, mereka dengan buta warna hijau ringan dapat melihat garis merah, secara samar kemungkinan bisa melihat garis ungu</p> <p>d. Orang dengan buta warna total tidak bisa melihat apapun</p>

	<ul style="list-style-type: none">a. Mereka dengan mata normal atau buta warna total tidak bisa melihat garisb. Orang dengan buta warna parsial dapat melihat garis secara berkelok tergantung parahnya kondisic. Orang dengan buta warna total tidak bisa melihat apapun
	<ul style="list-style-type: none">a. Mereka dengan mata normal dapat melihat garis hijau secara berkelokb. Mereka dengan buta warna parsial atau buta warna total tidak dapat melihat garis
	<ul style="list-style-type: none">a. Mereka dengan mata normal dapat melihat garis oranye berkelokb. Mereka dengan buta warna parsial atau buta warna total tidak dapat melihat garis
	<ul style="list-style-type: none">a. Mereka dengan mata normal dapat melihat garis biru-hijau atau kuning-hijau secara berkelokb. Mereka dengan buta warna parsial dapat melihat garis biru-hijau dan merahc. Mereka dengan buta warna total tidak bisa melihat garis manapun

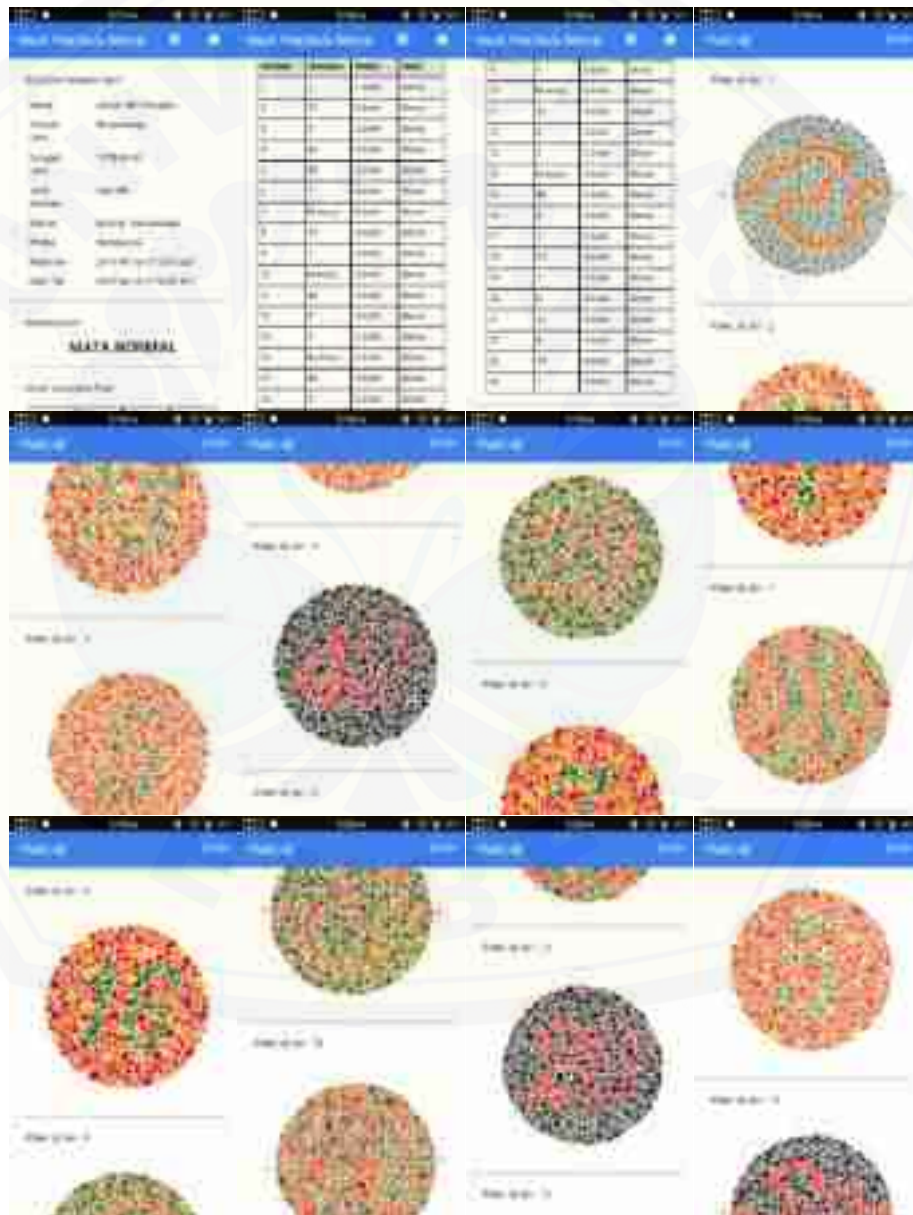
	<ol style="list-style-type: none">a. Mereka dengan mata normal dapat melihat garis merah dan oranye secara berkelokb. Merka dengan buta warna parsial dapat melihat garis merah dan biru-hijau secara berkelokc. Mereka dengan buta warna total tidak bisa melihat garis manapun
	<p>Semua orang mampu melihat garis berkelok</p>

B. Hasil Tes 20 Responden

Pada bagian ini memaparkan hasil tes aplikasi buta warna Metode Ishihara yang dilakukan pada dua puluh responden, yaitu berupa *screenshot* hasil tes buta warna yang dijalankan pada android.

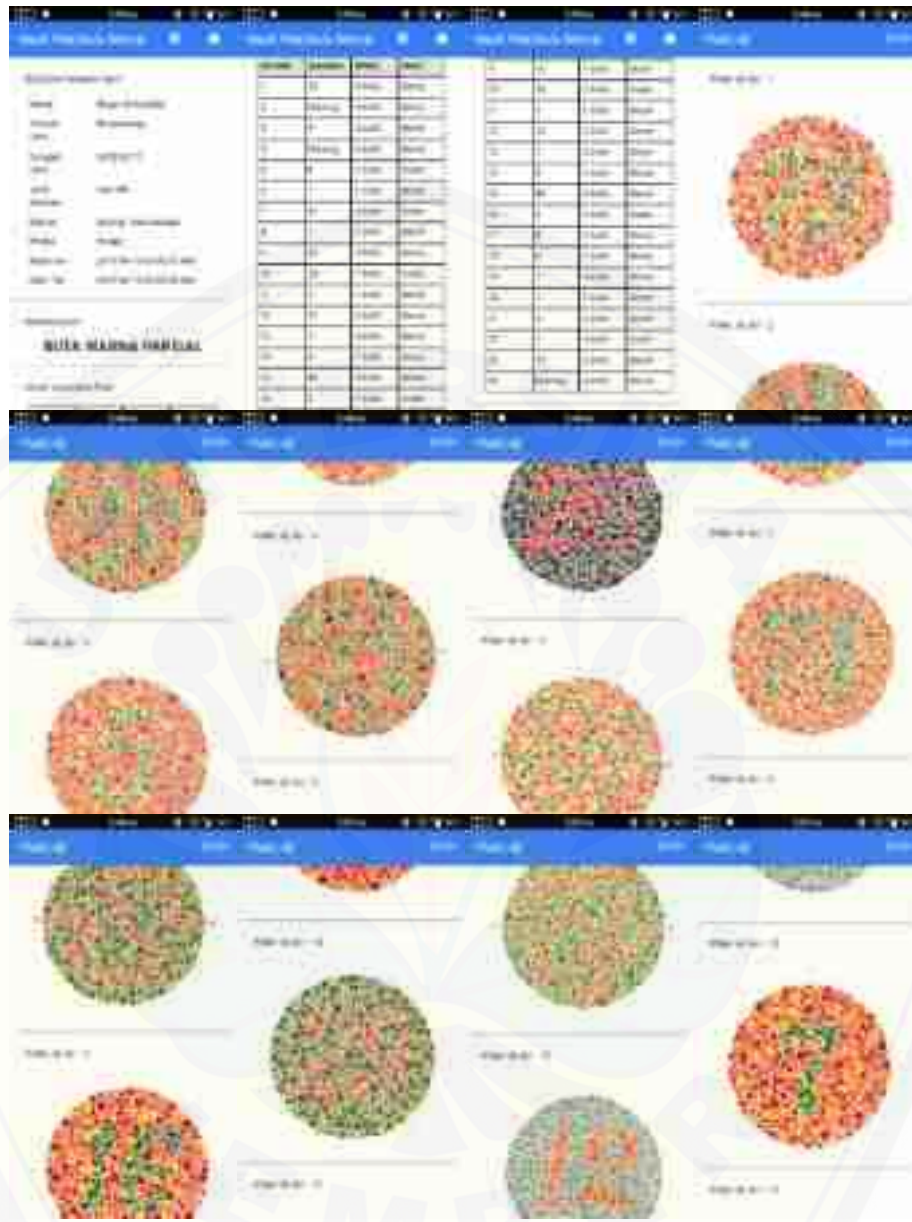
B.1 Tes Pertama

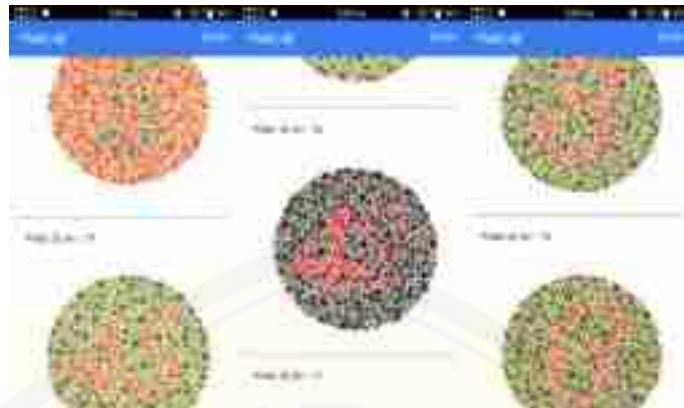
1. Abrian Eko Prasetyo



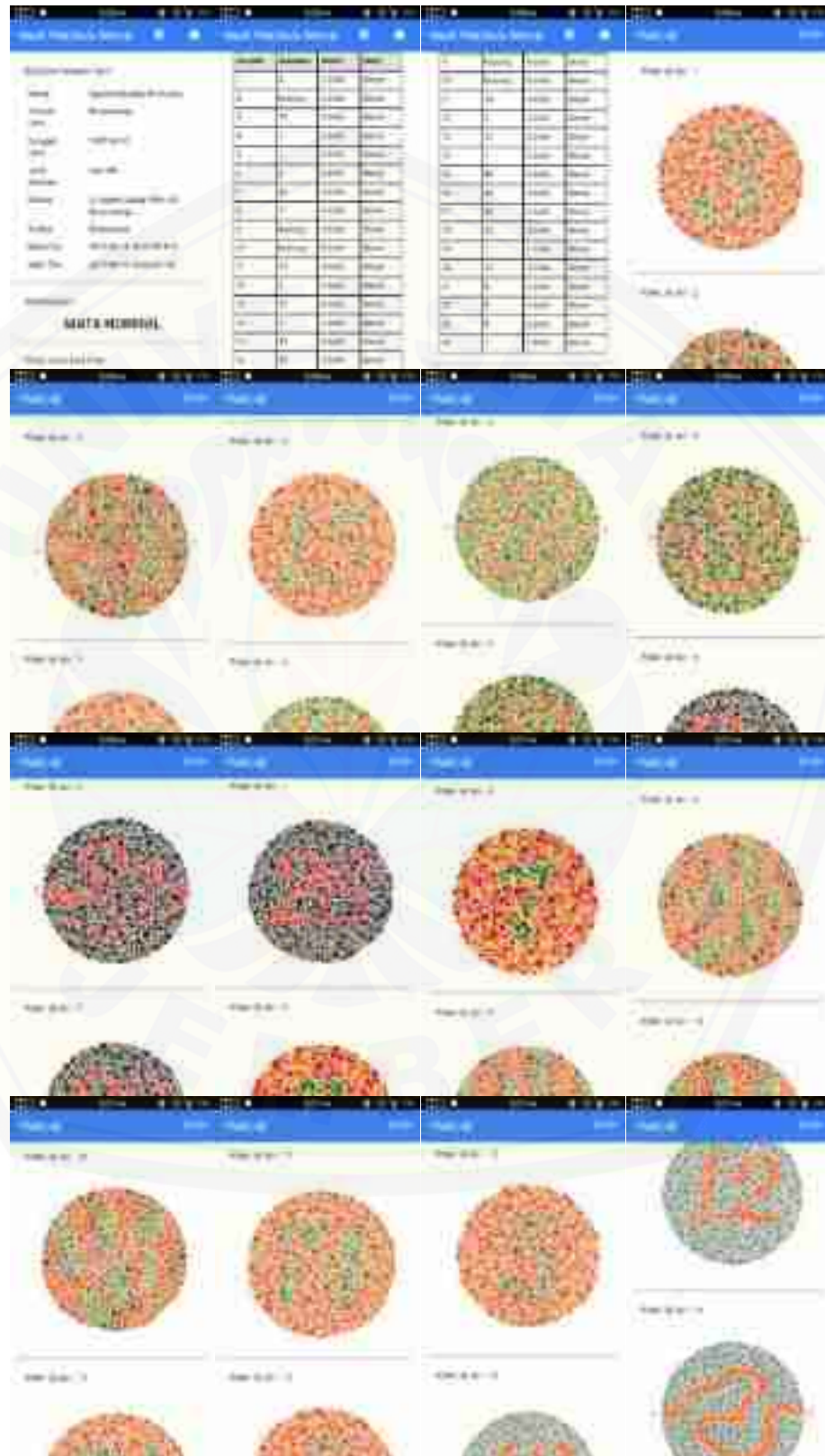


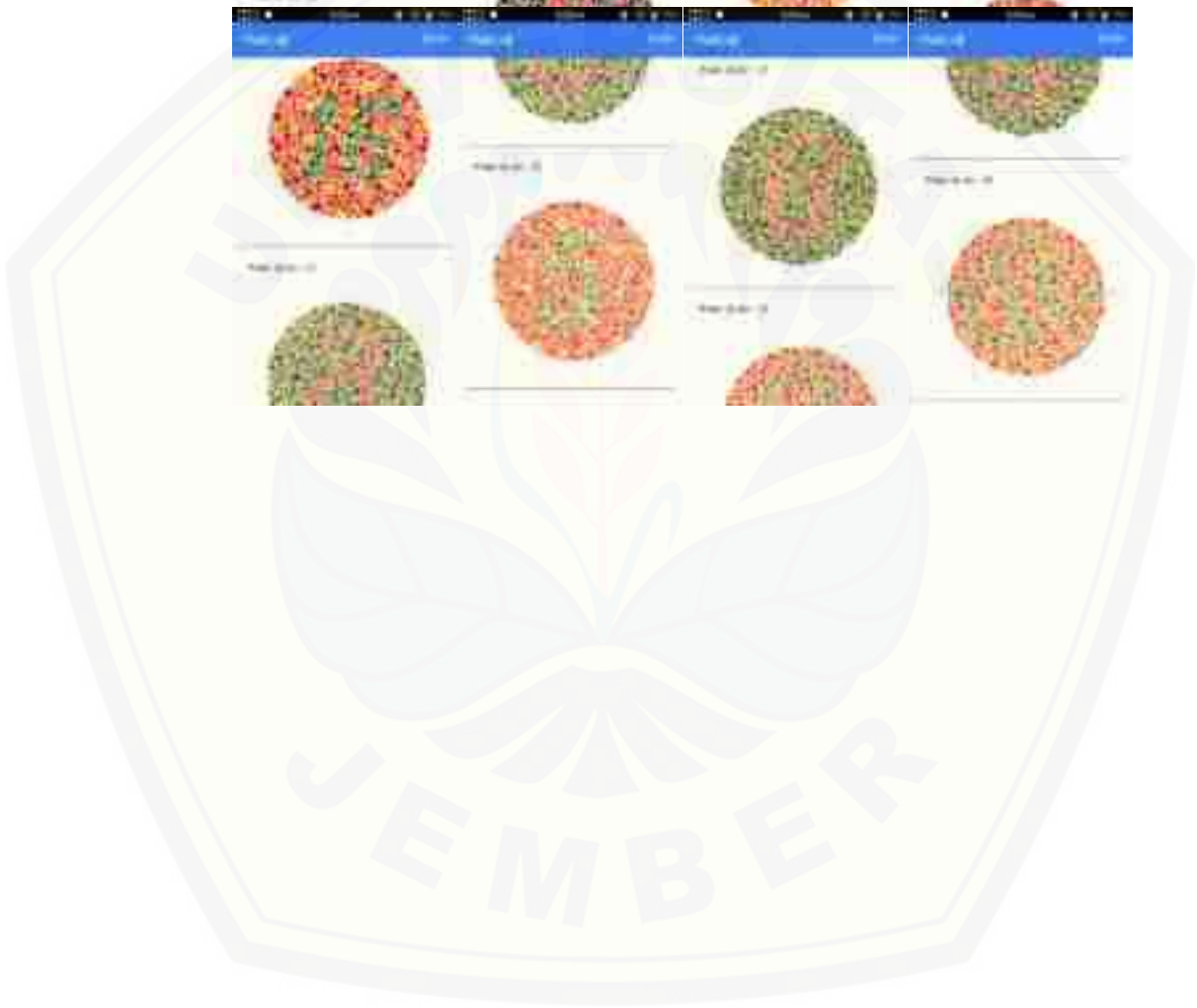
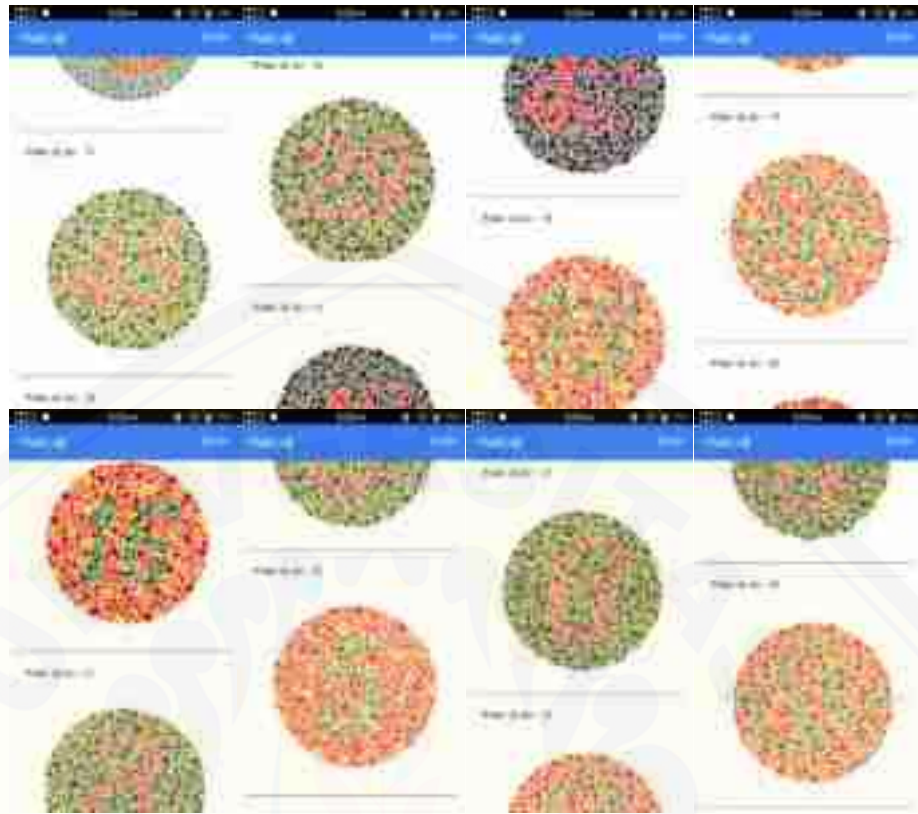
2. Bagas Di Saputra



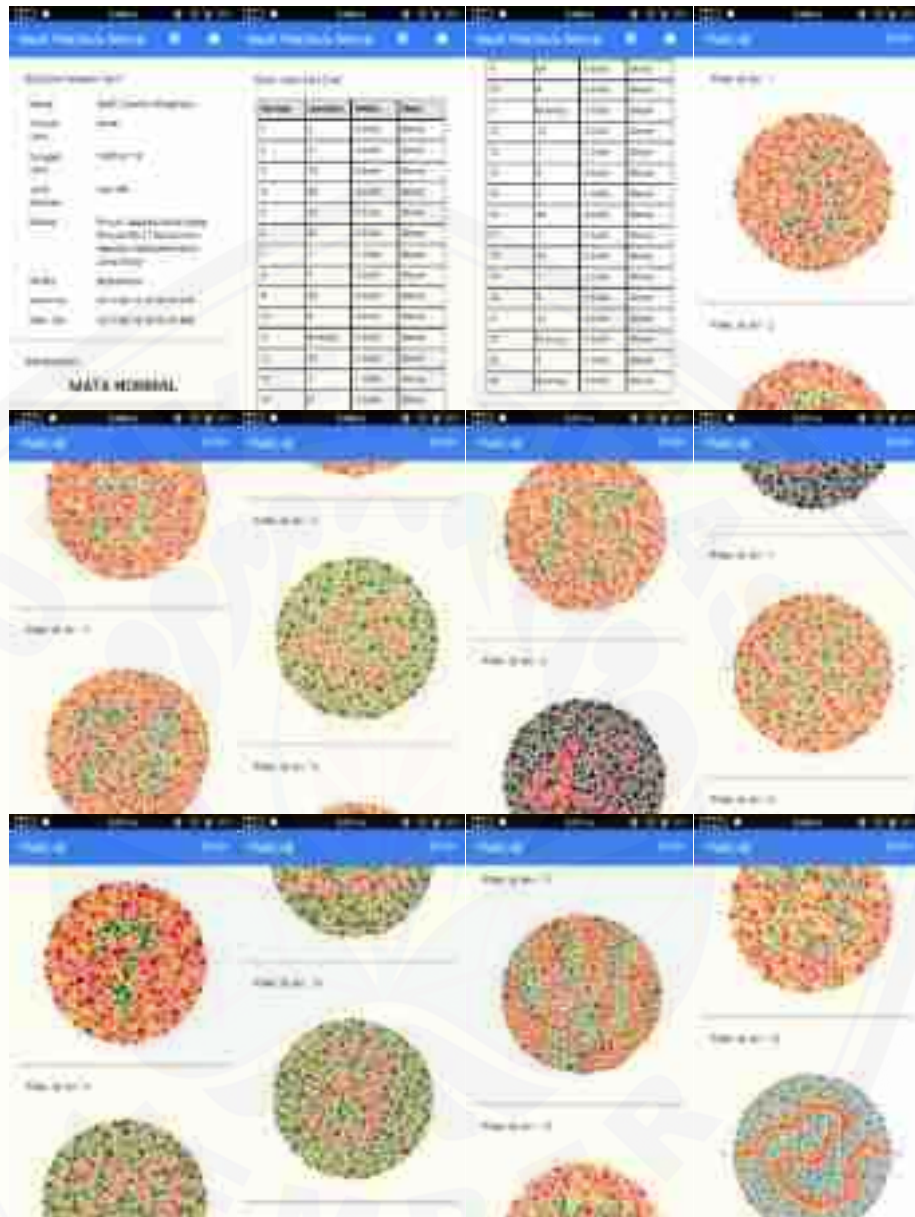


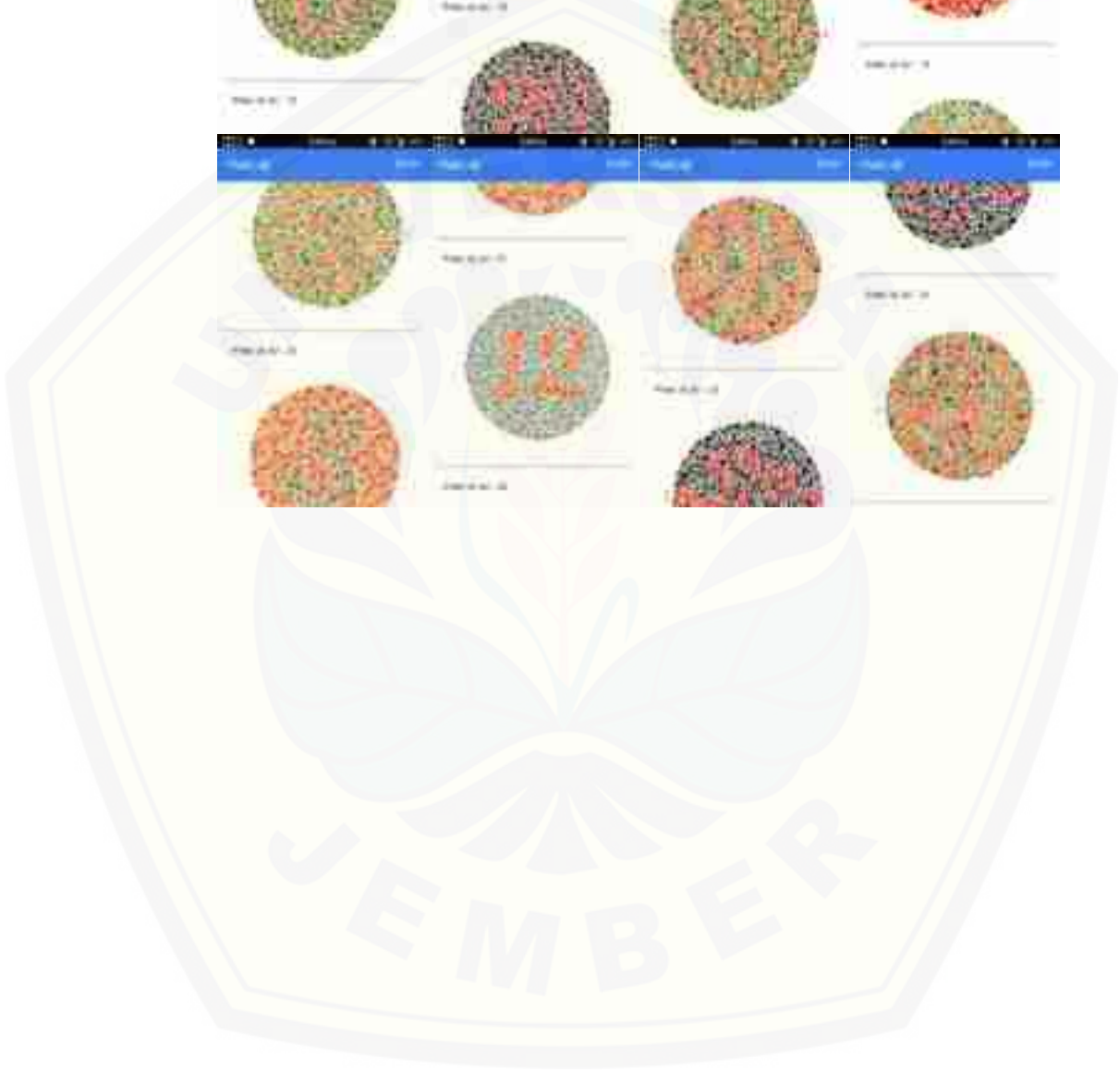
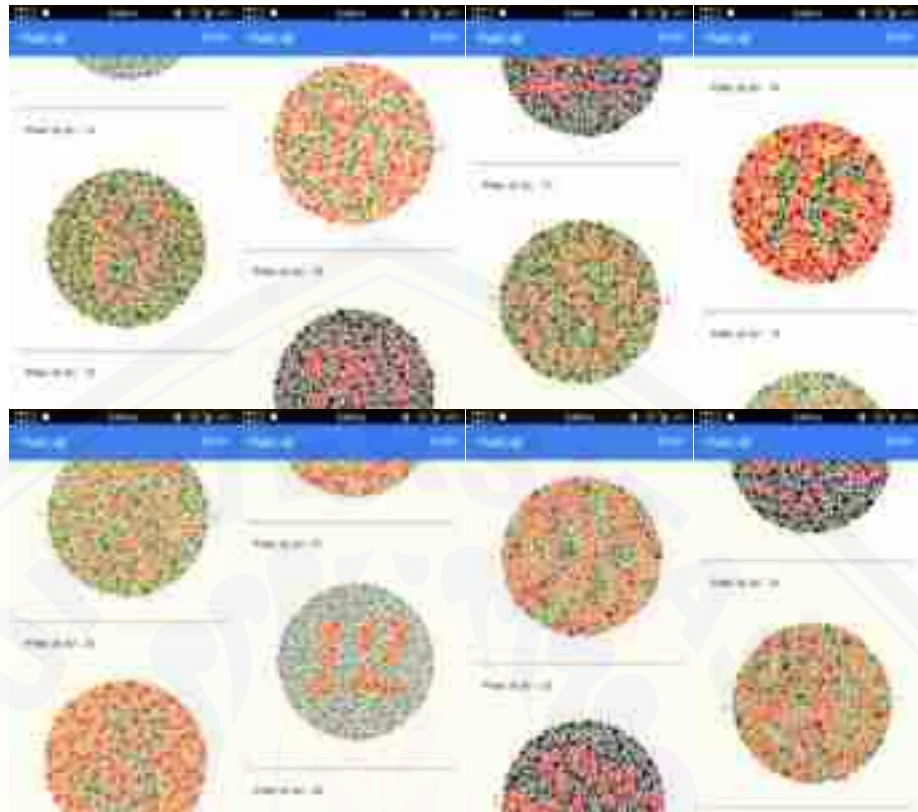
3. Agusty Maulana Bramasta



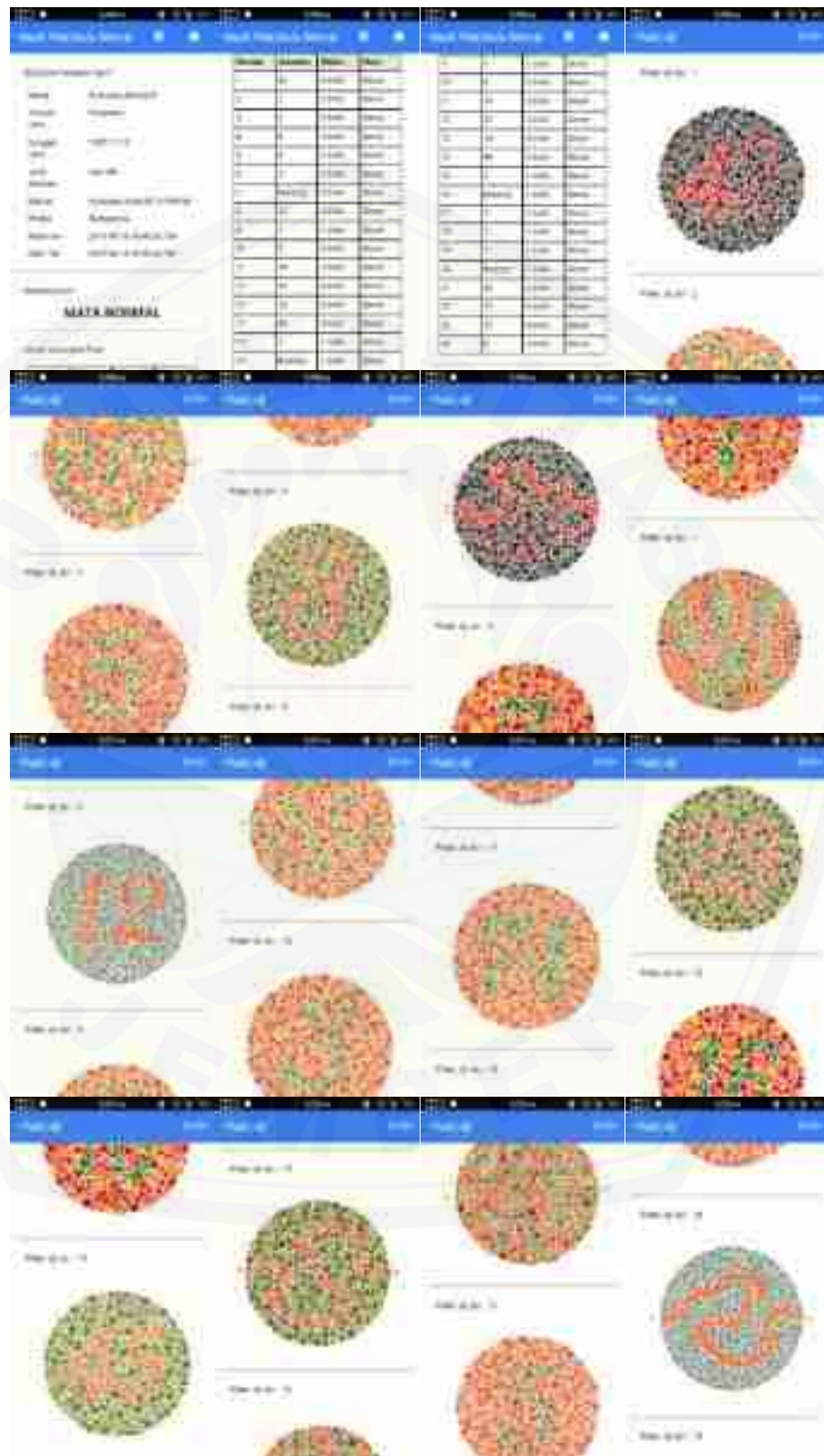


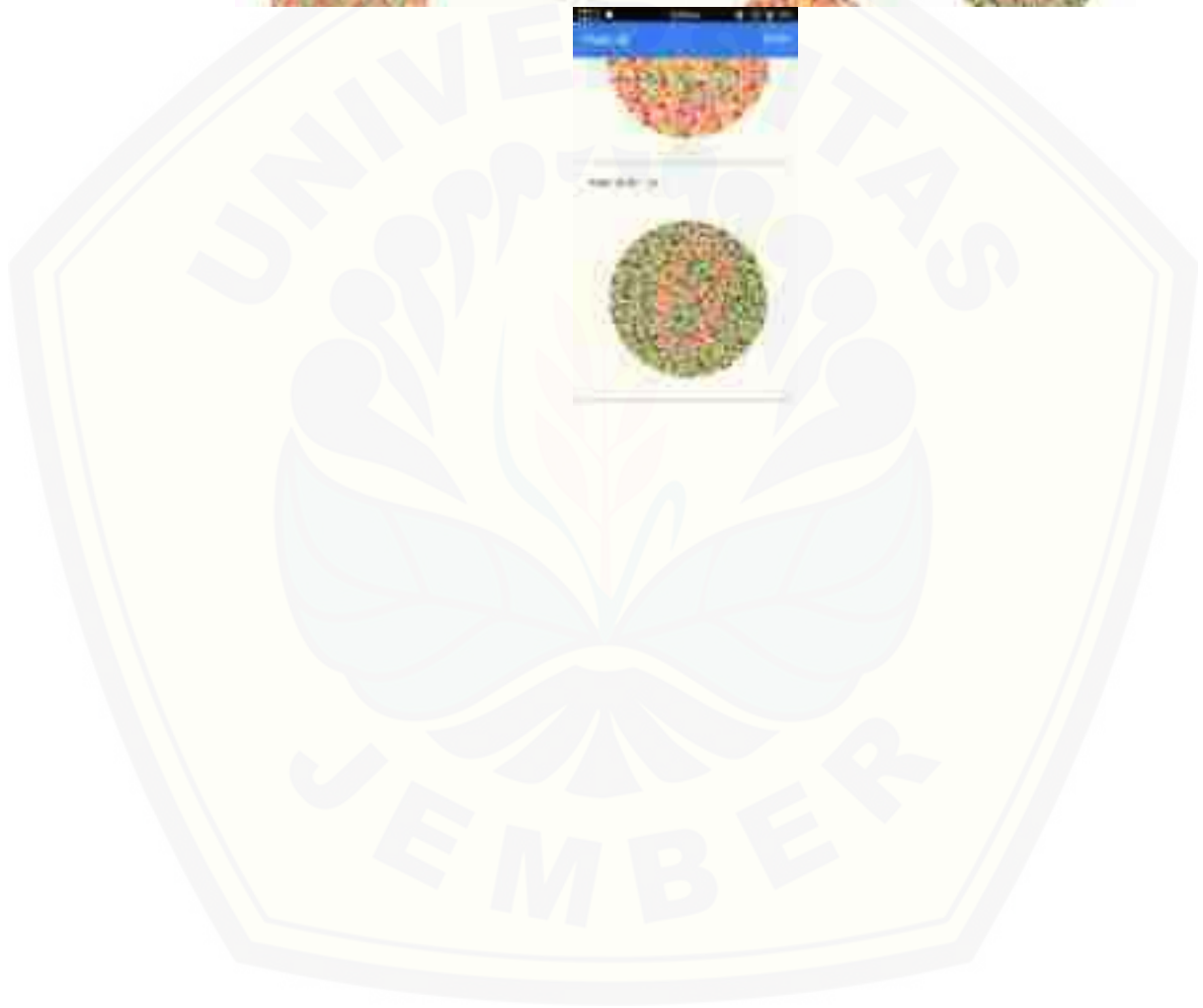
4. Galih Chandra Wiradhika



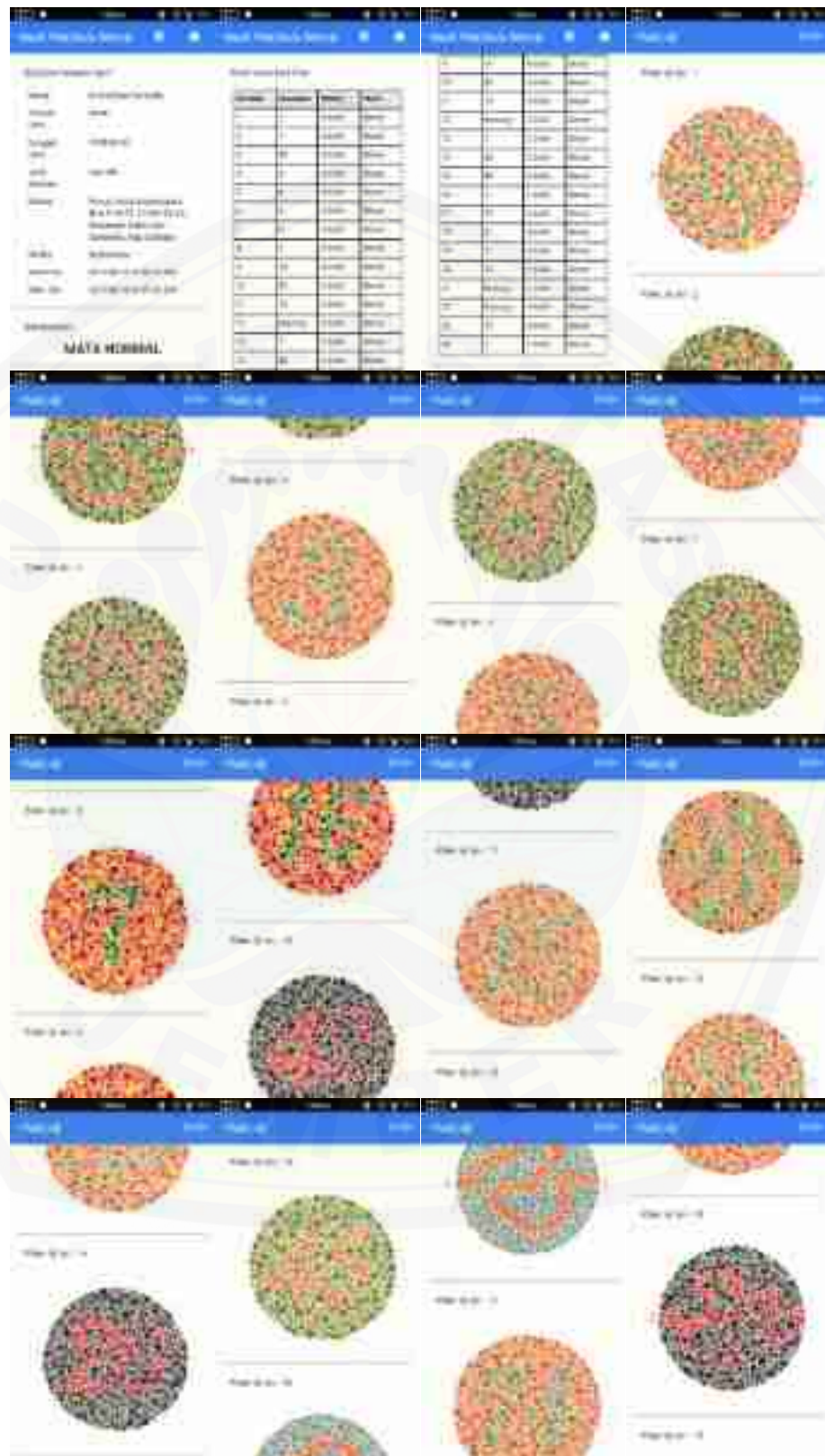


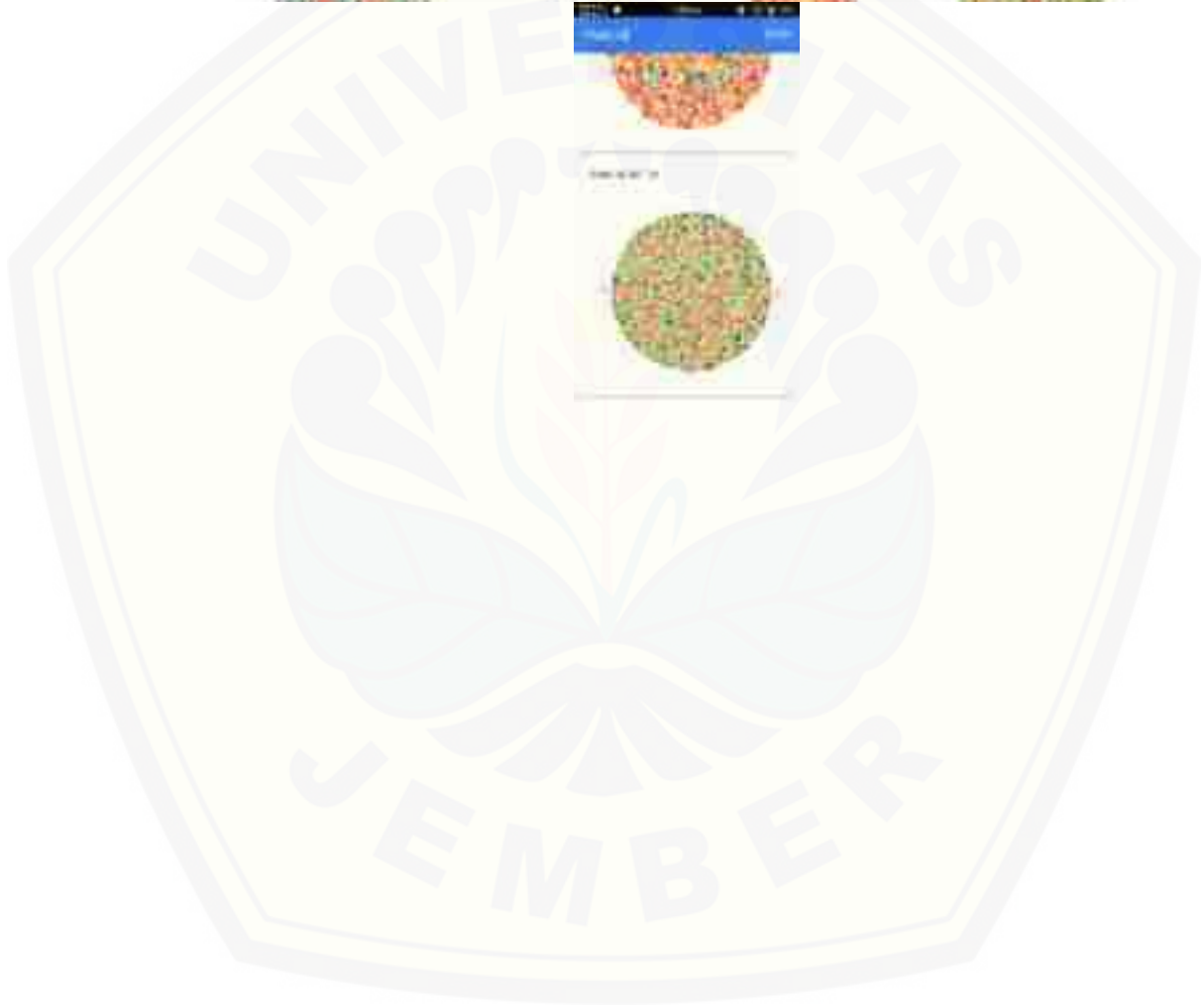
5. Kharisma Ahmad P.



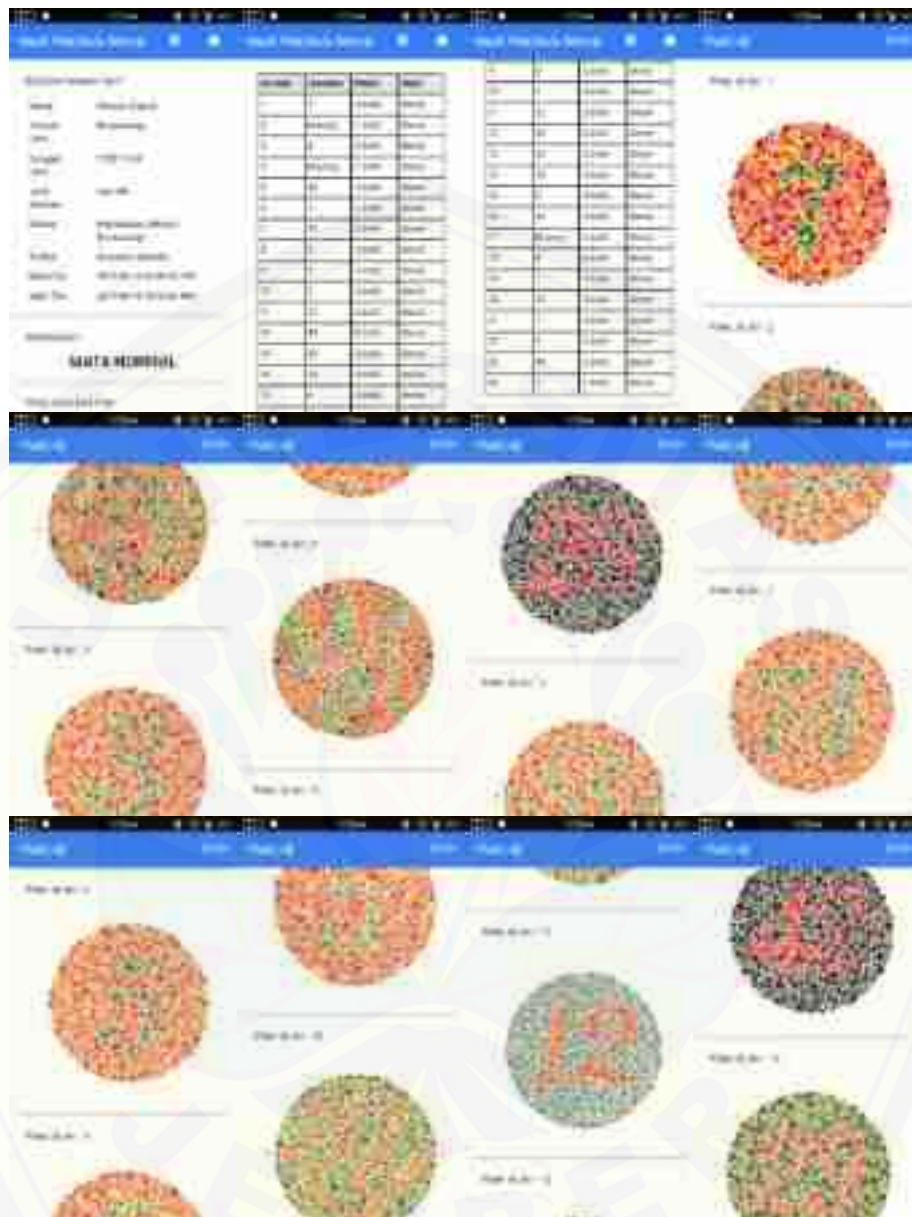


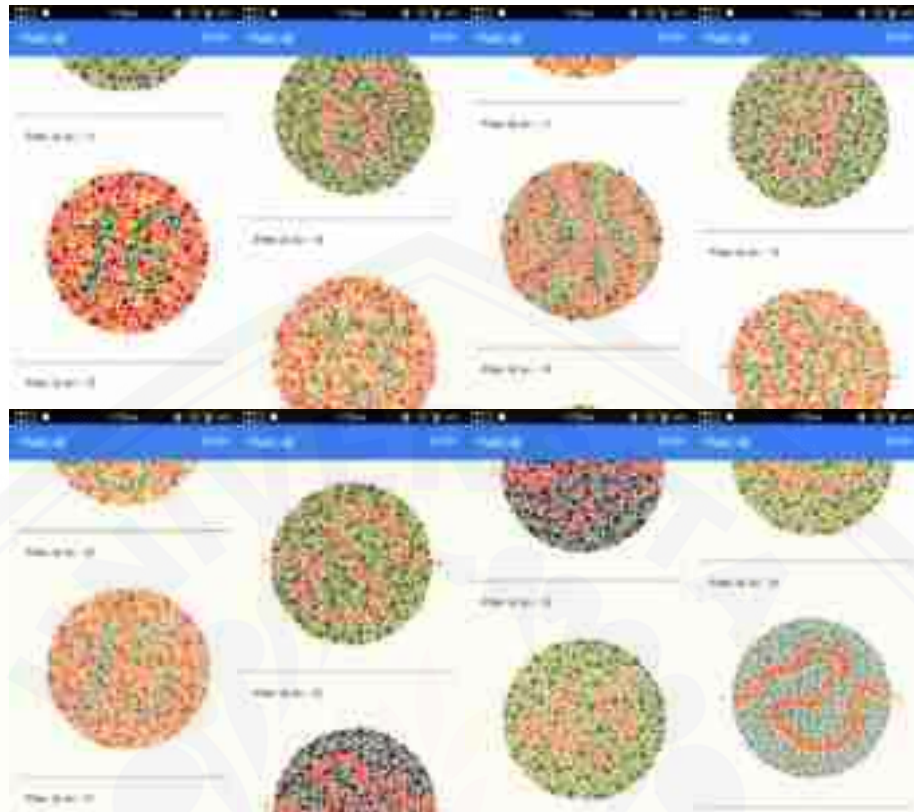
6. Emha Dinar Samudro



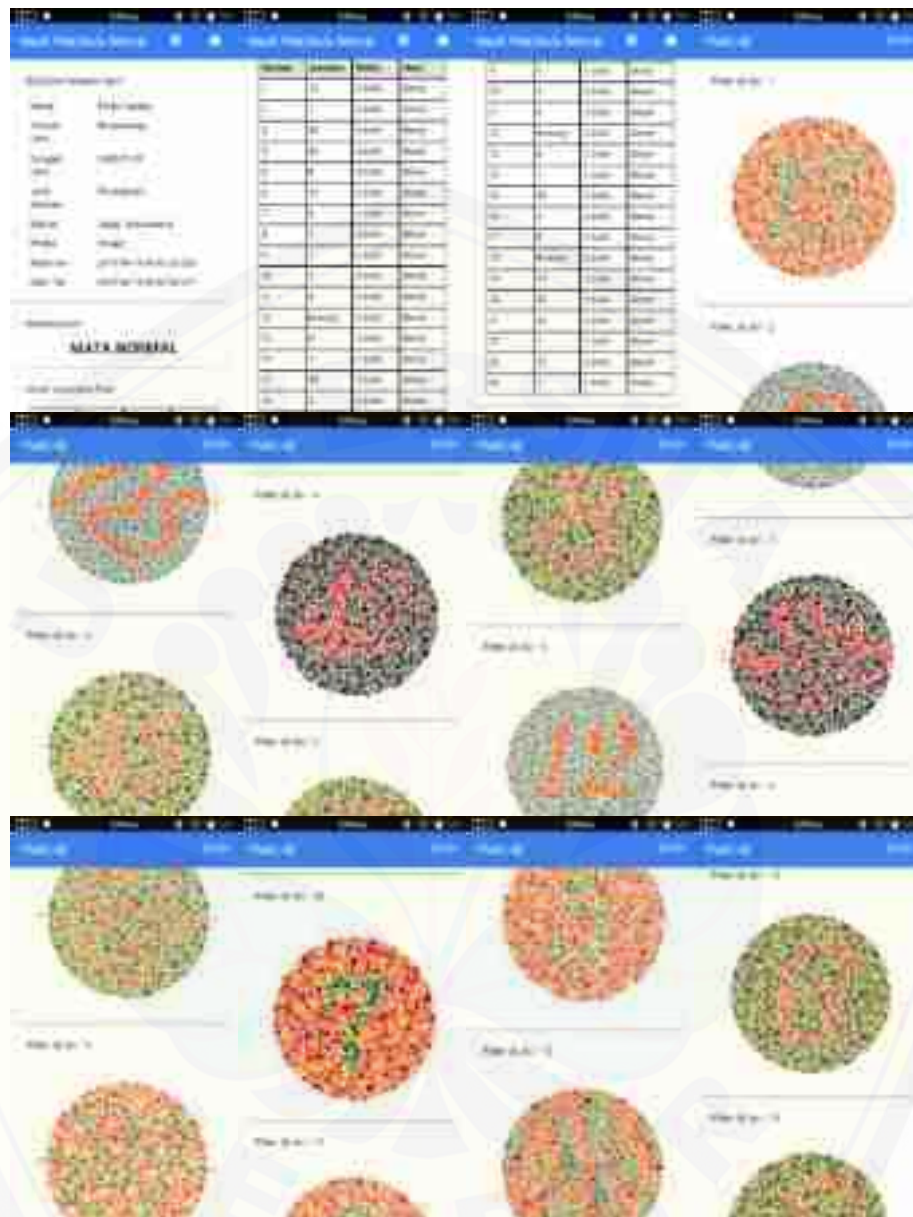


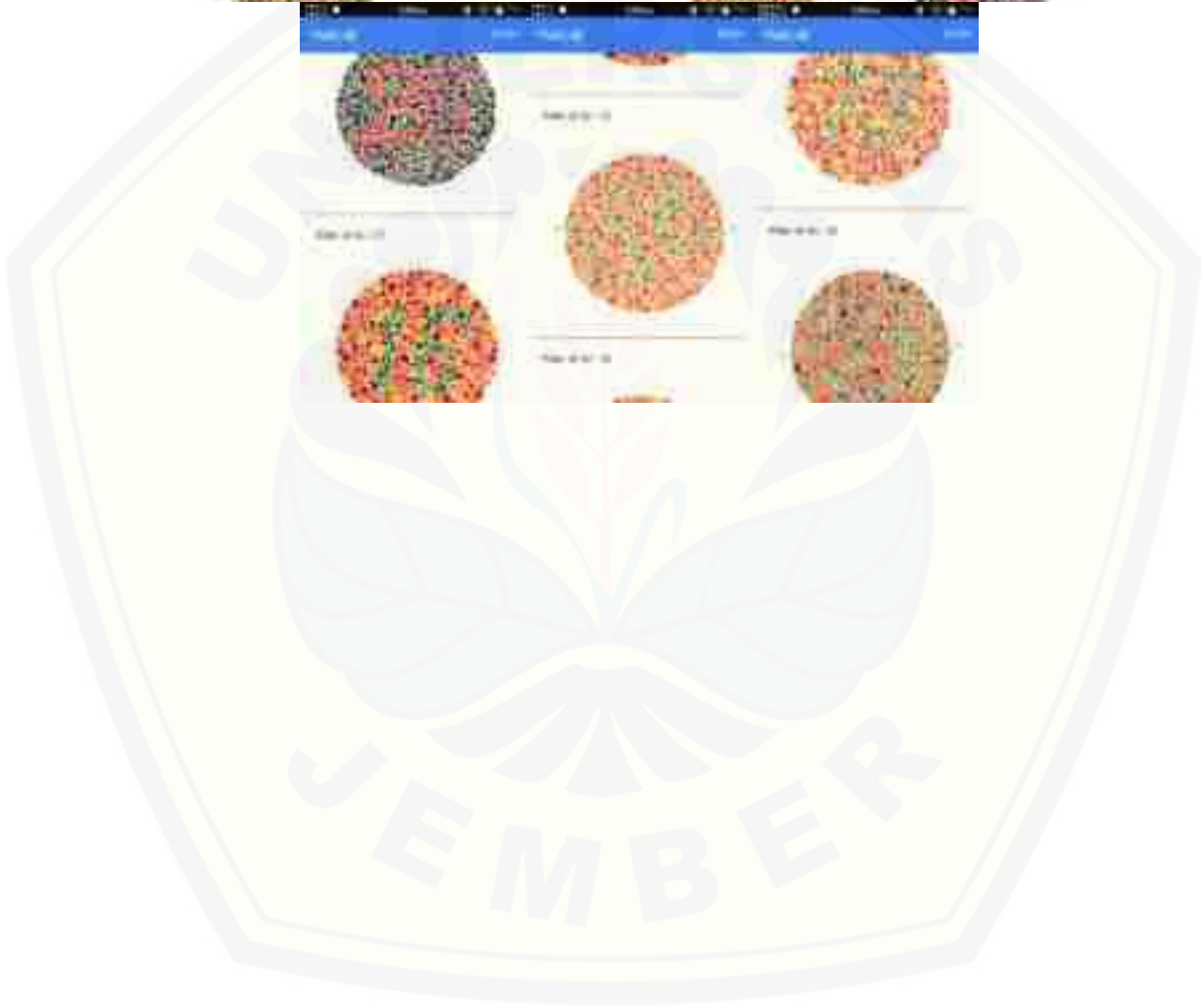
7. Novado Caesar



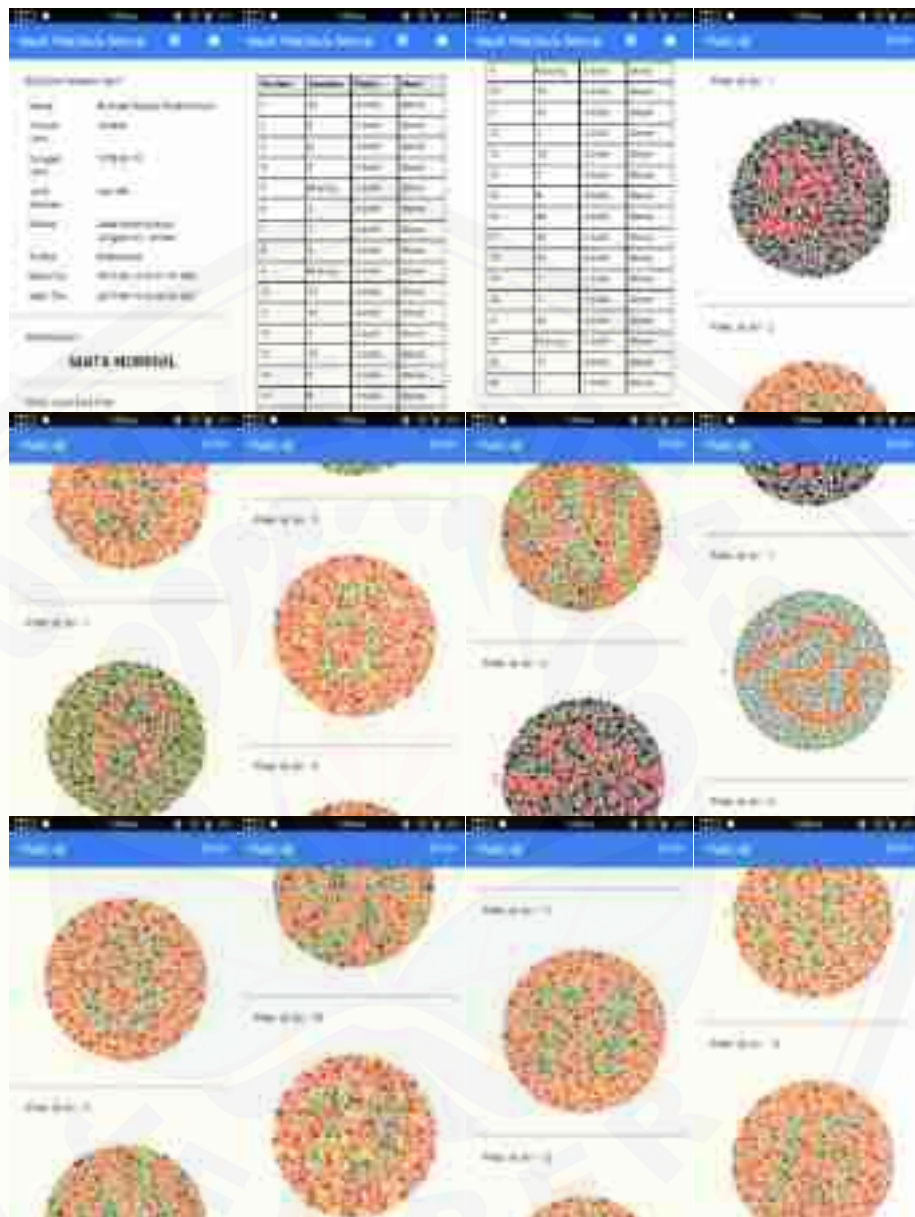


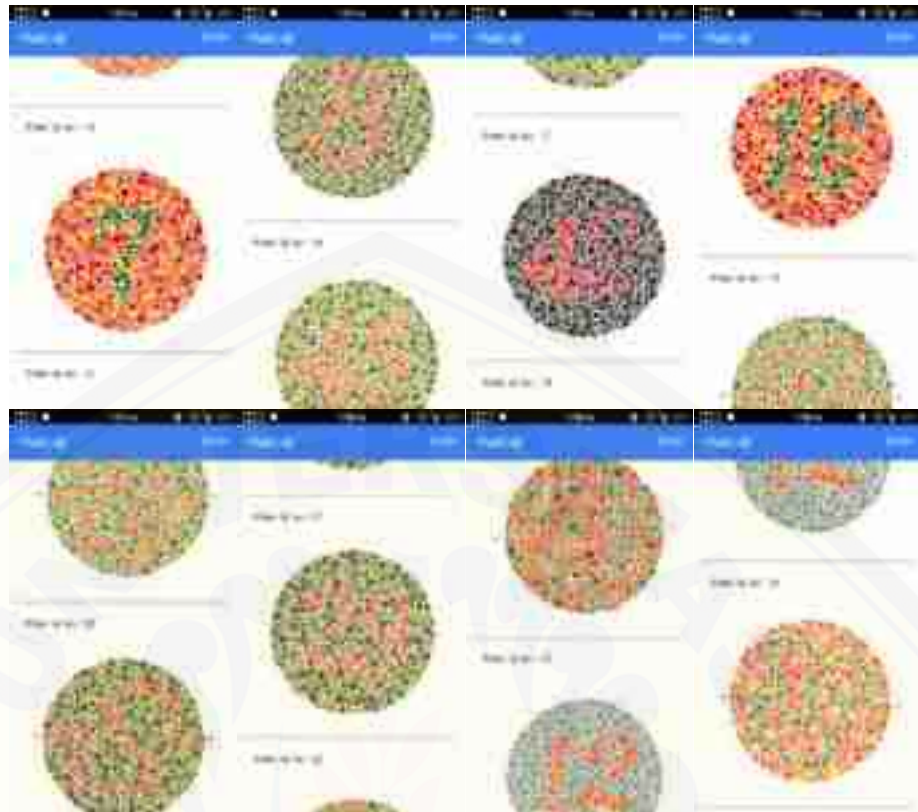
8. Ferika Sandra



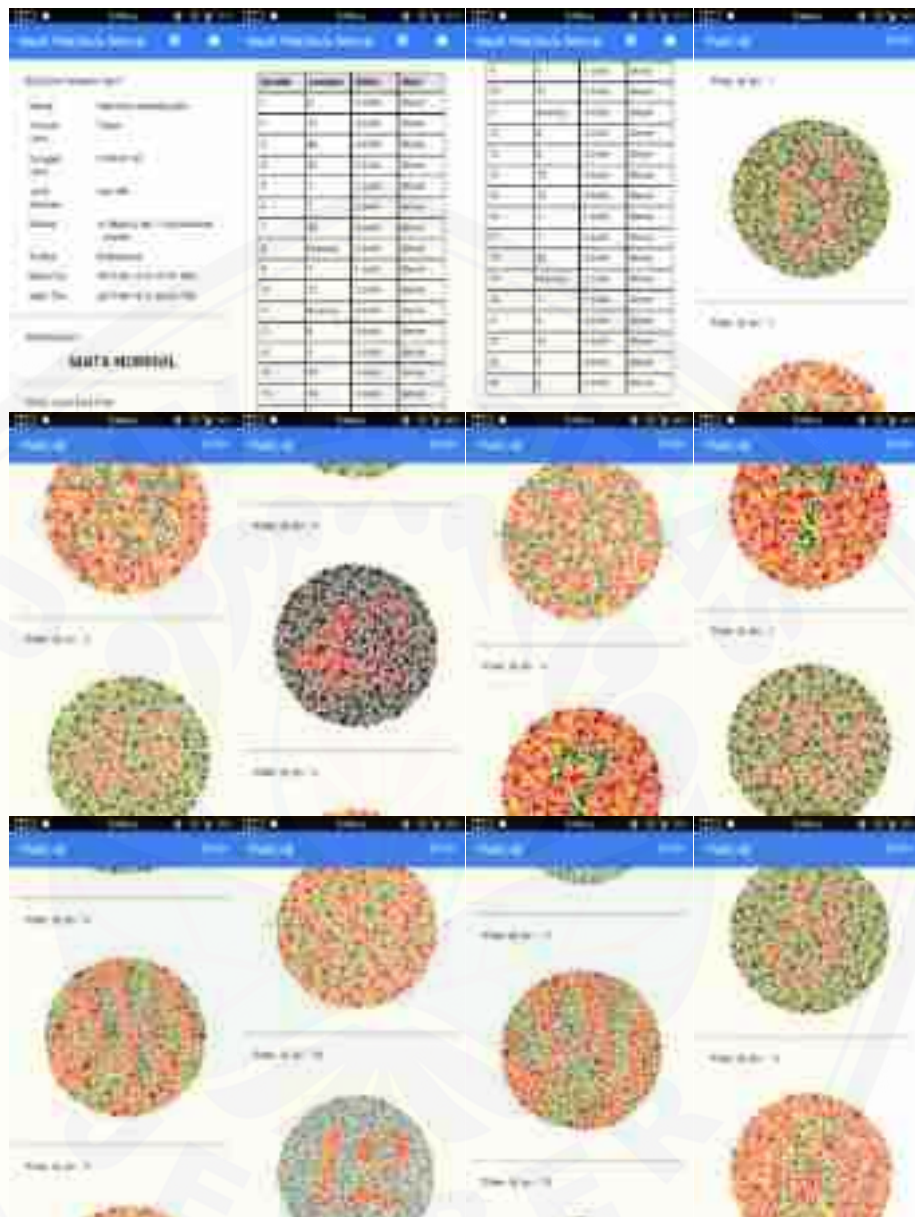


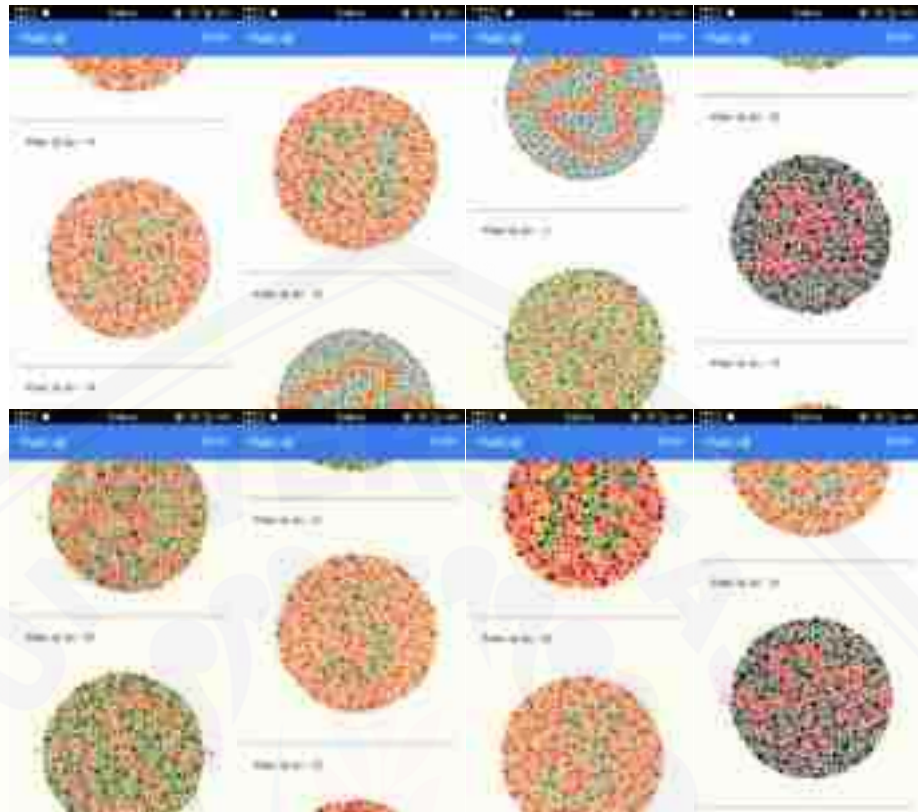
9. Rahmad Hidayat Hadi Subroto

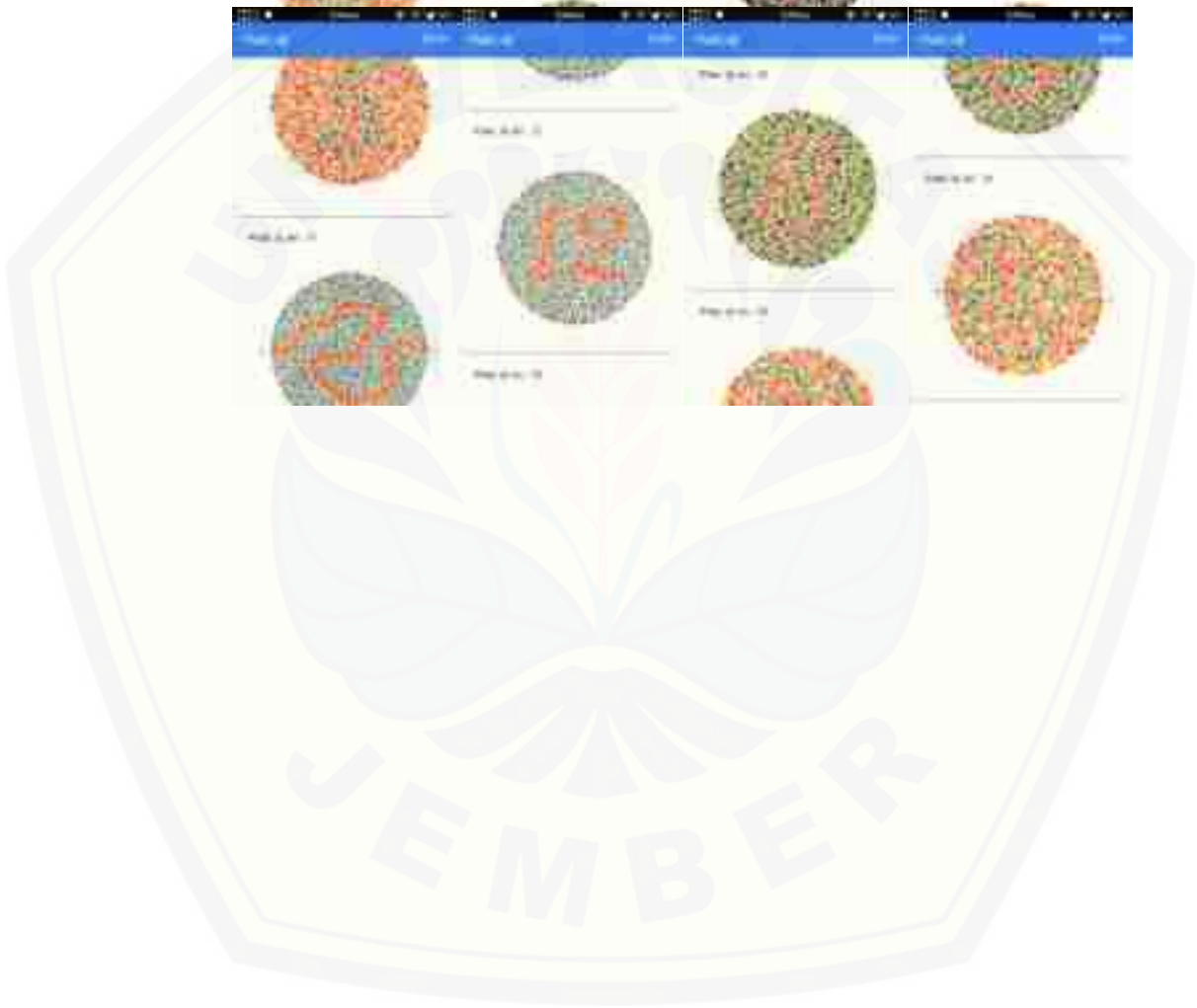
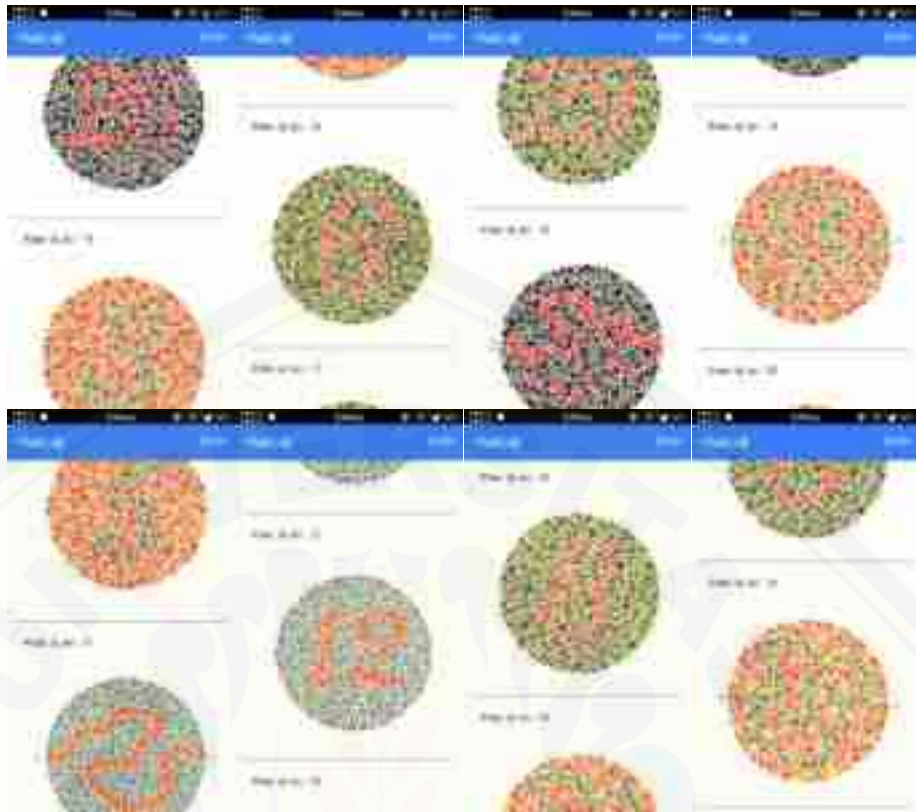




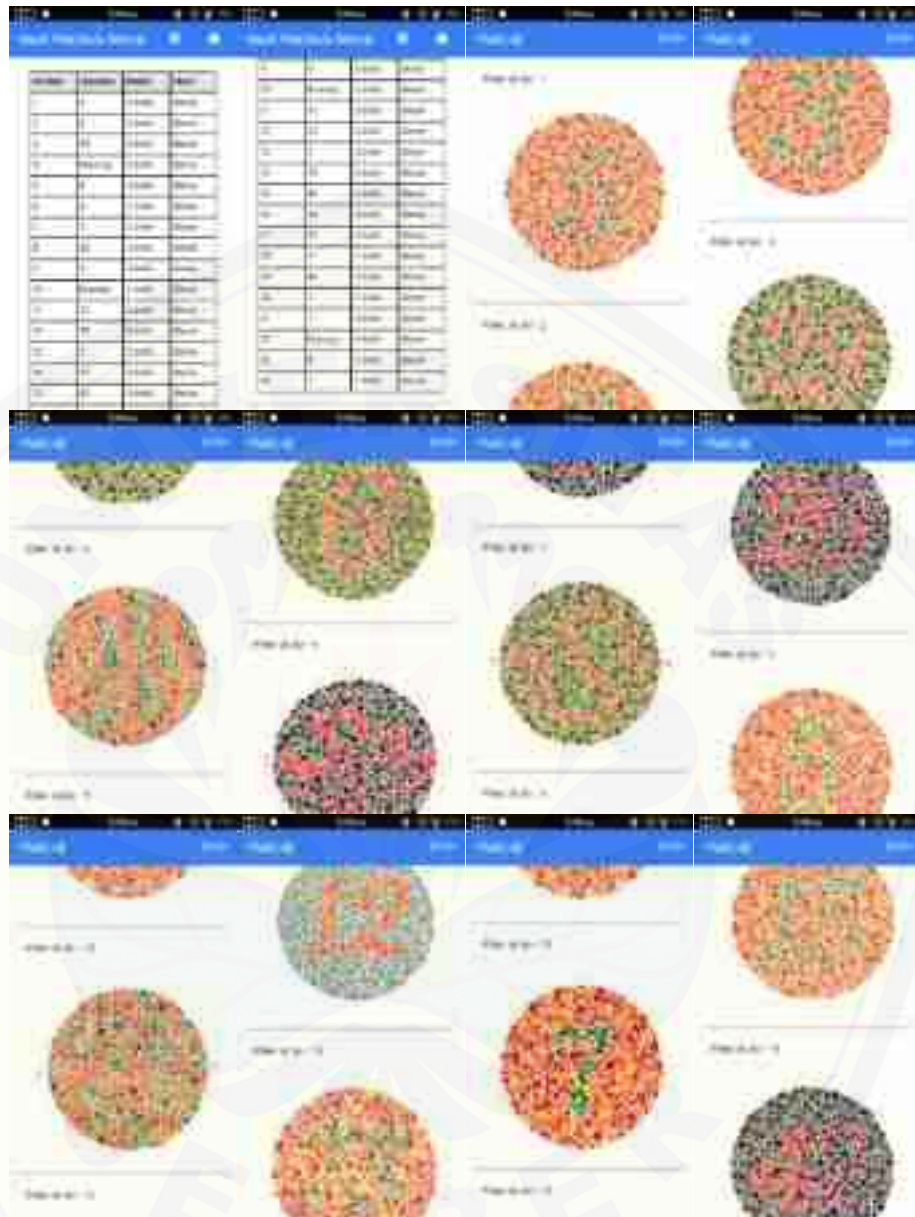
10. Hammam Iqomatuddin

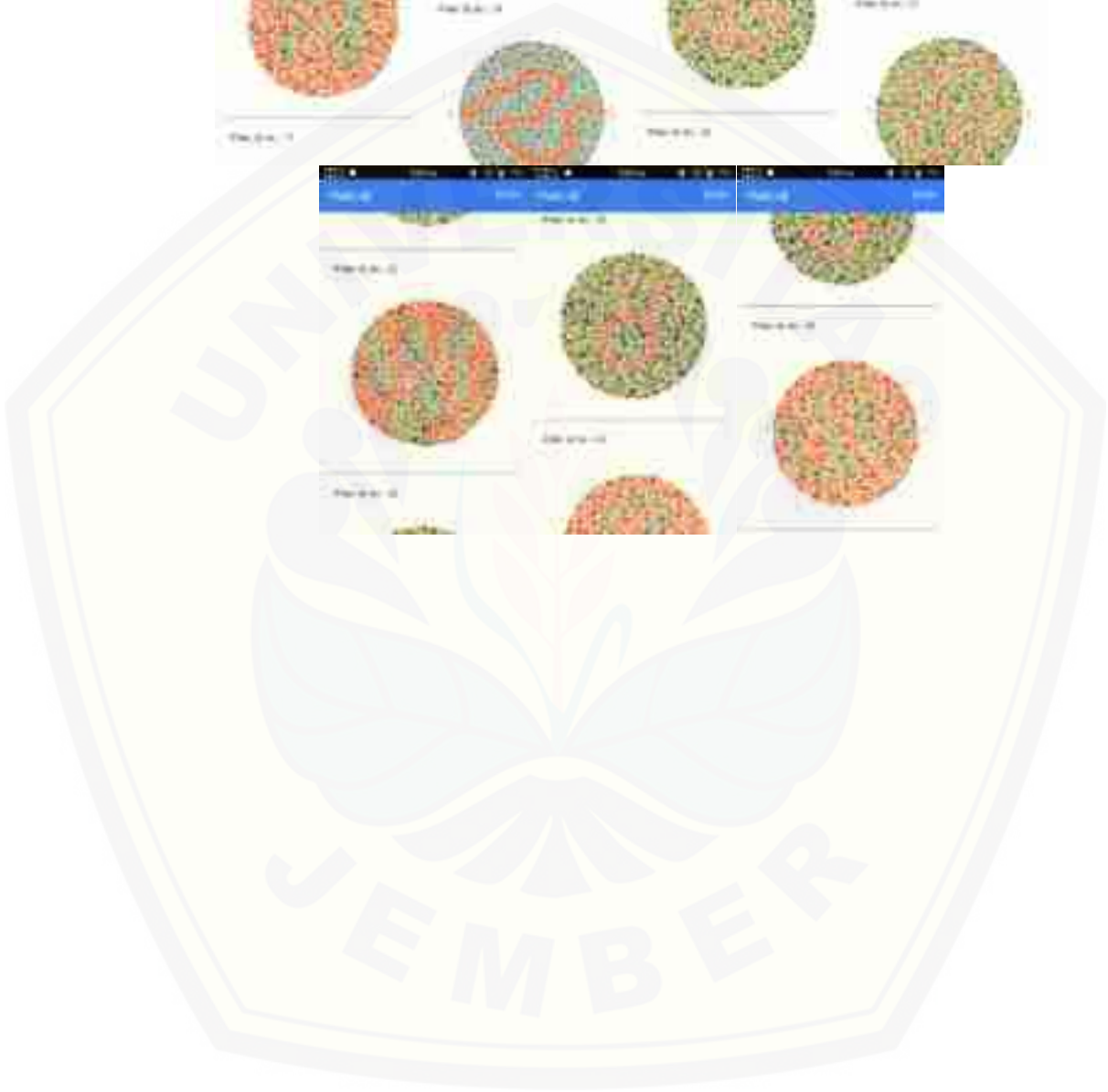
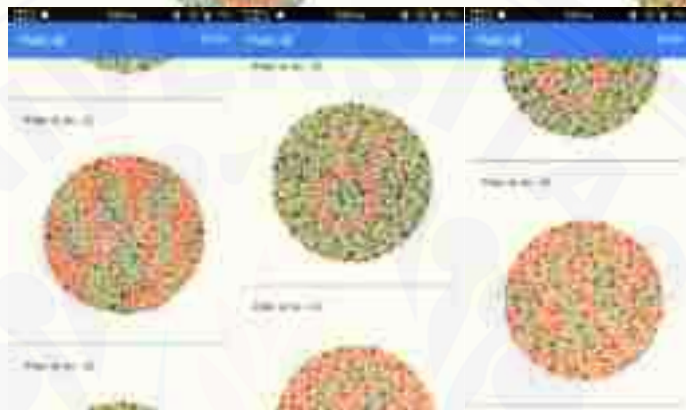




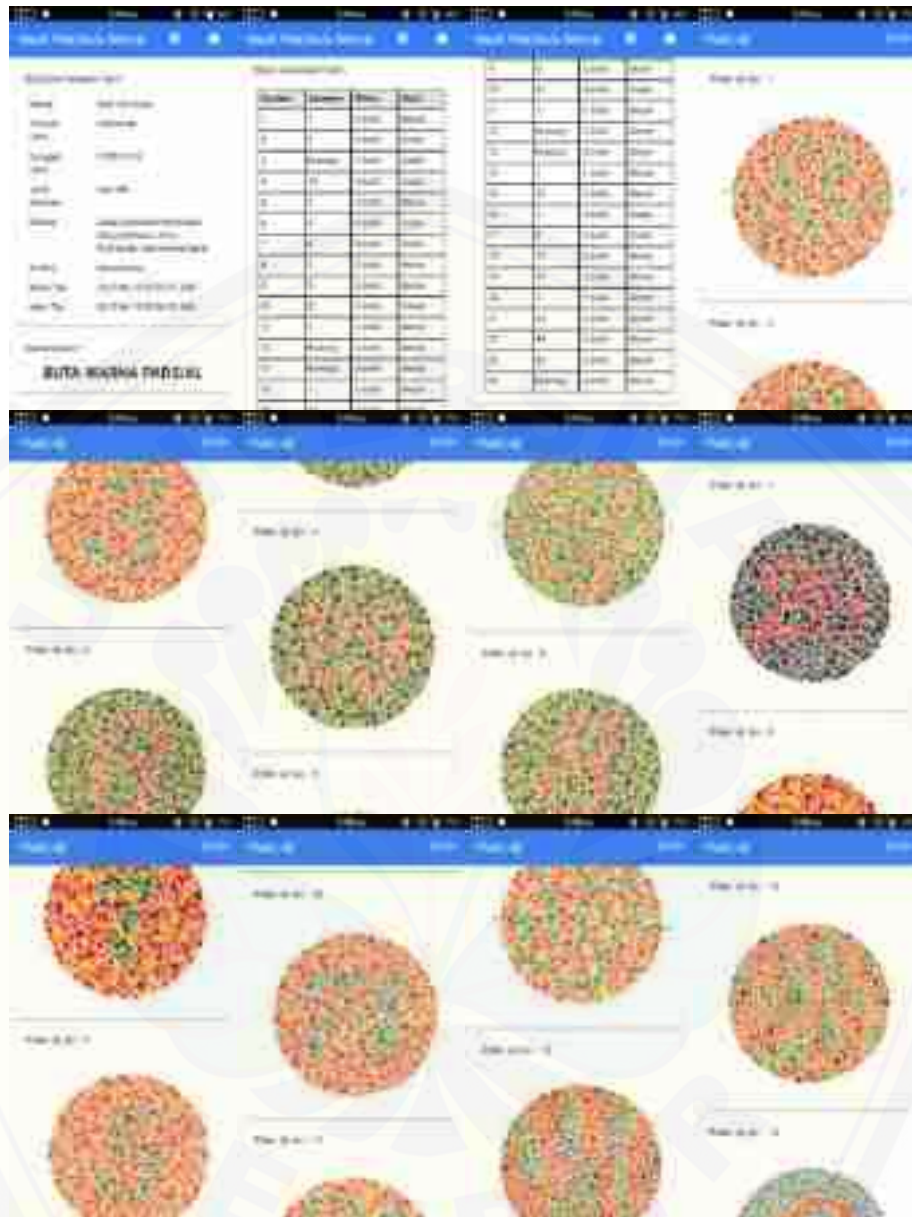


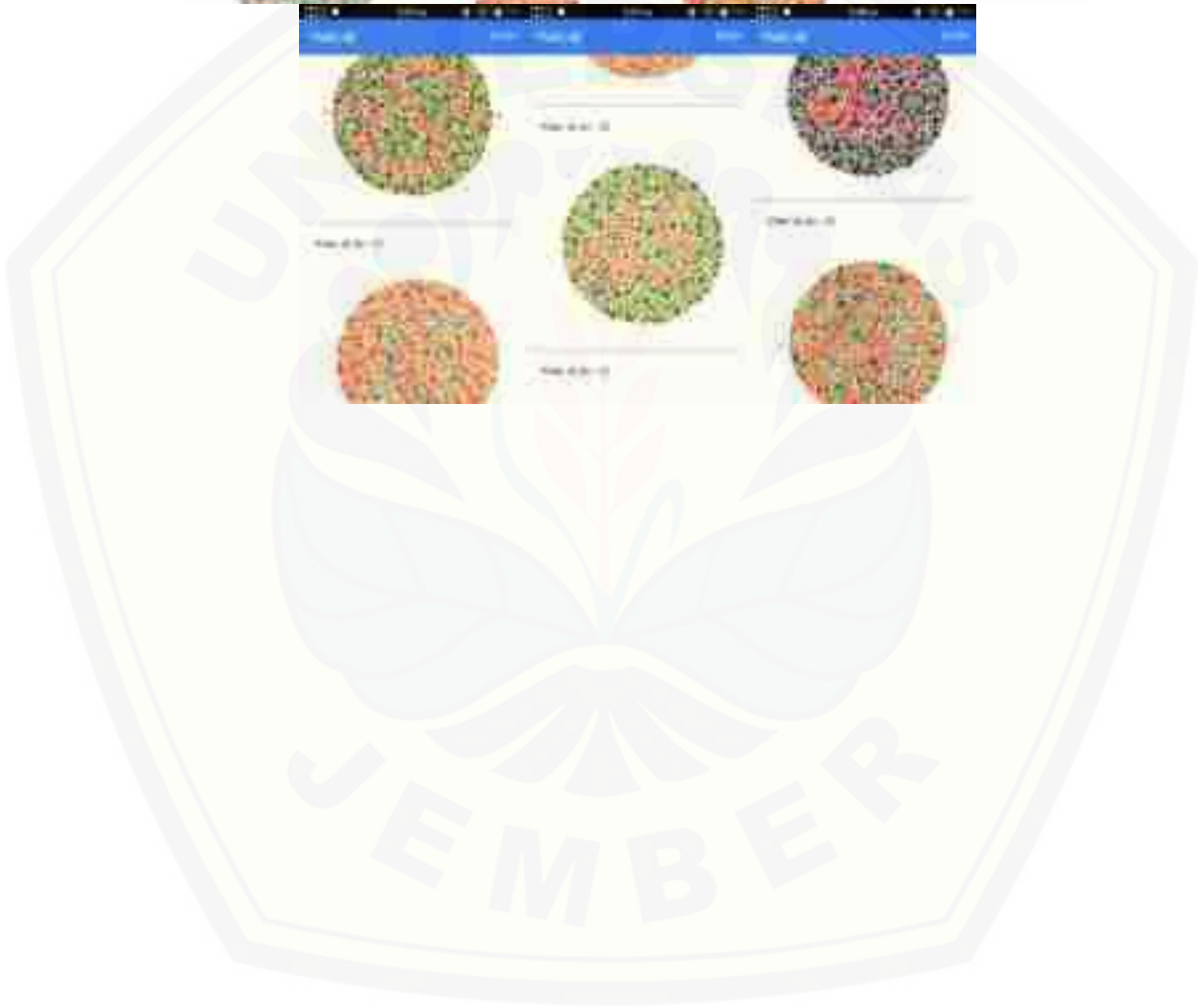
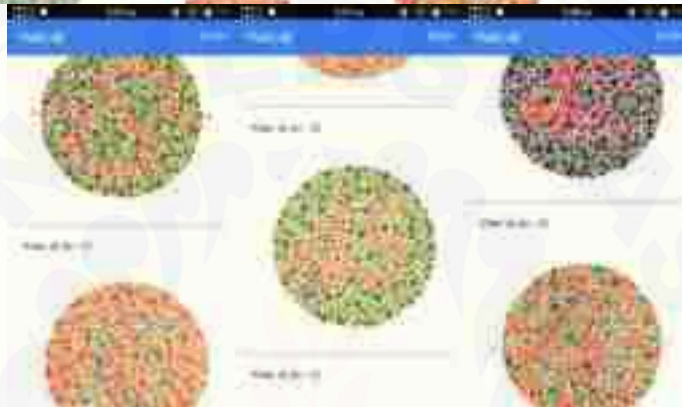
12. Muliyanah Suciono



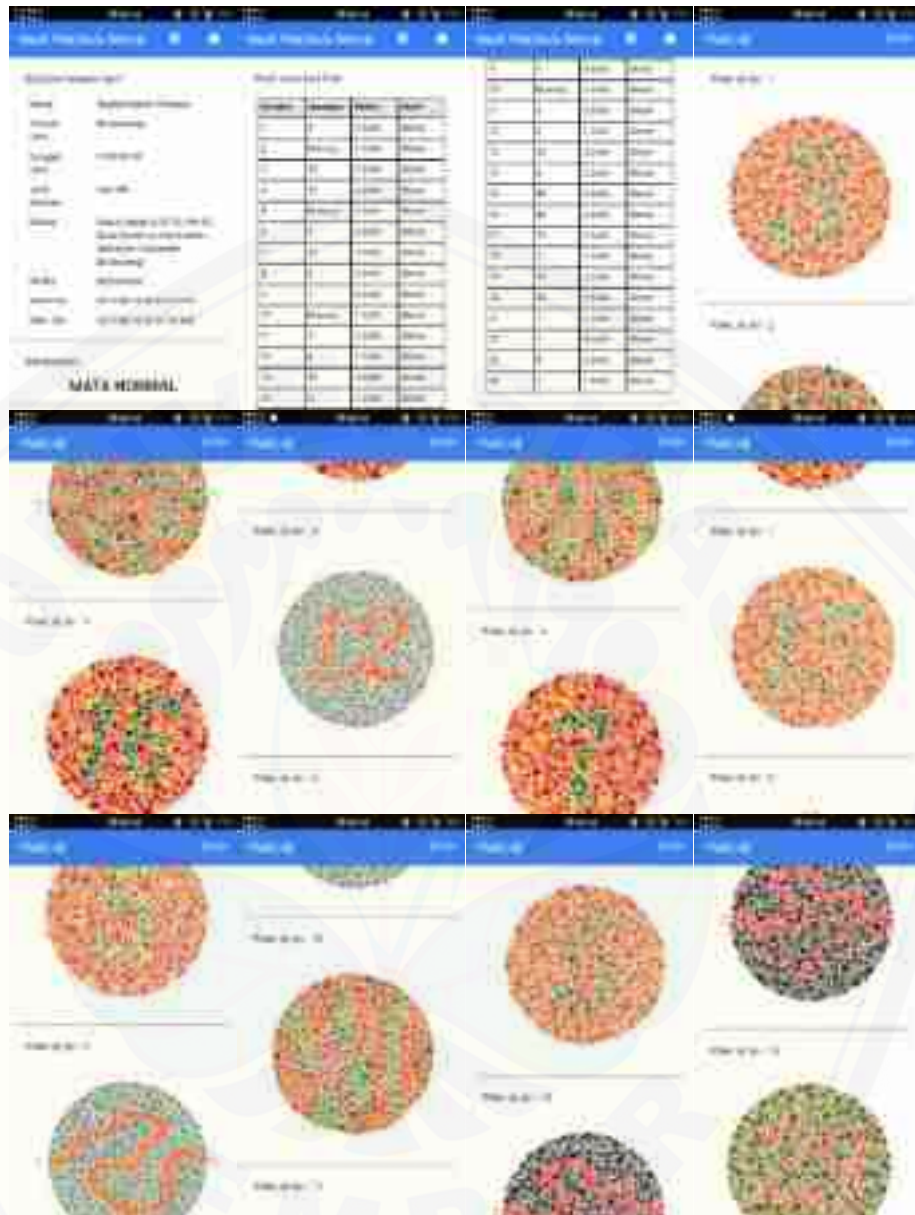


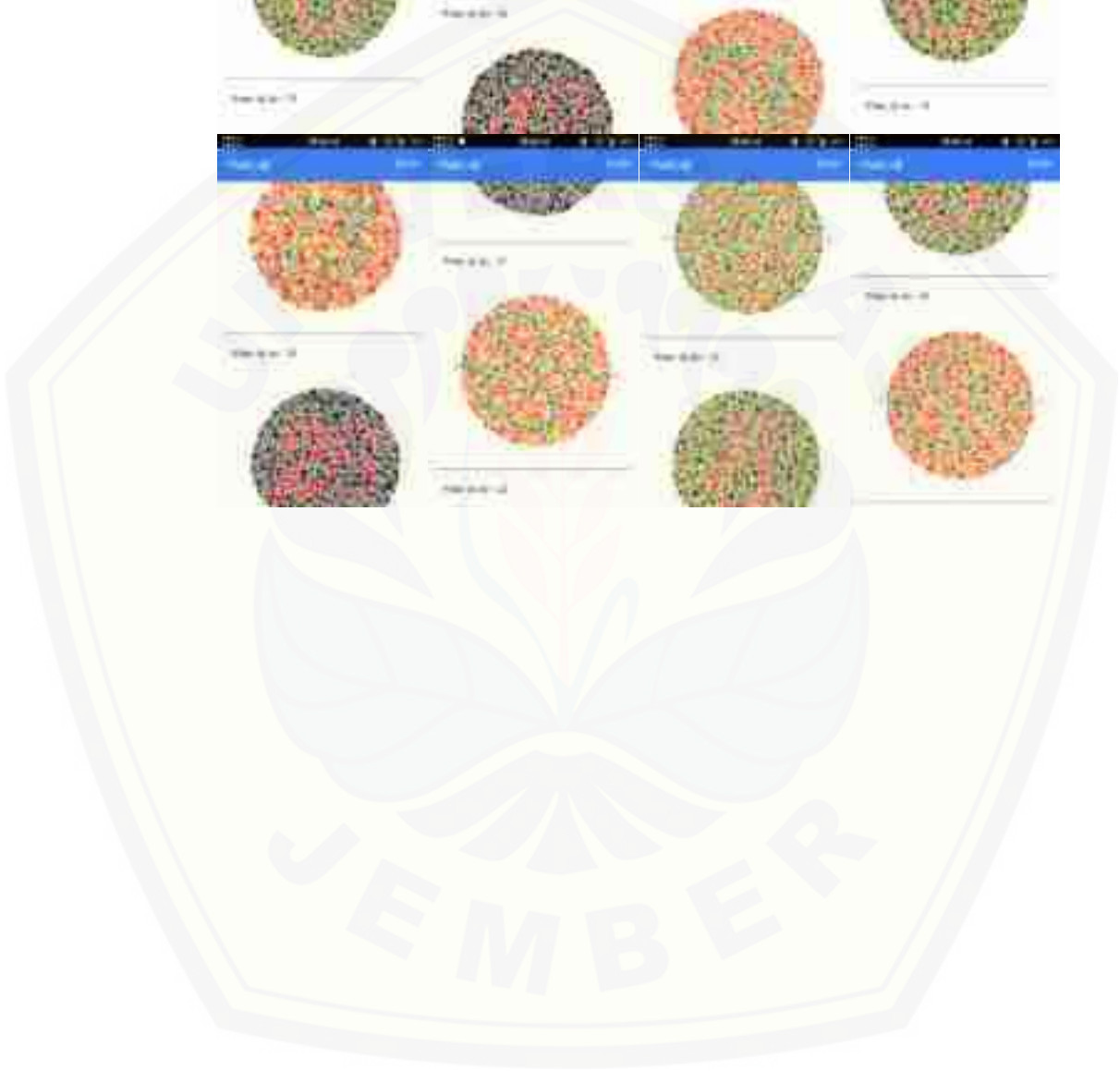
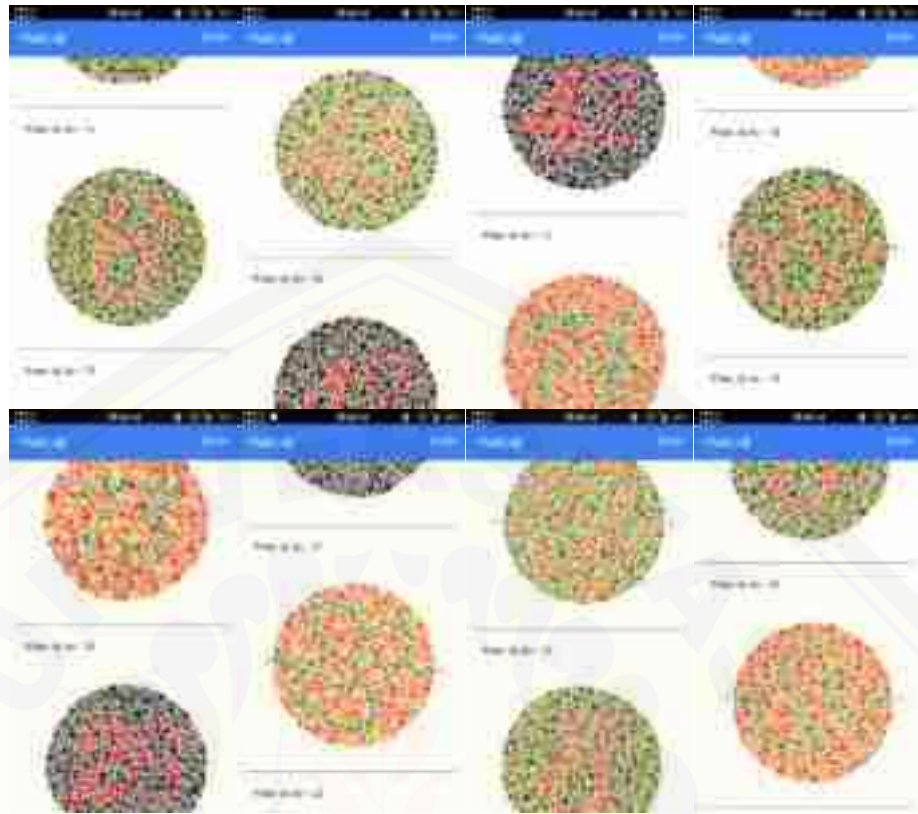
13. Andri Armanto



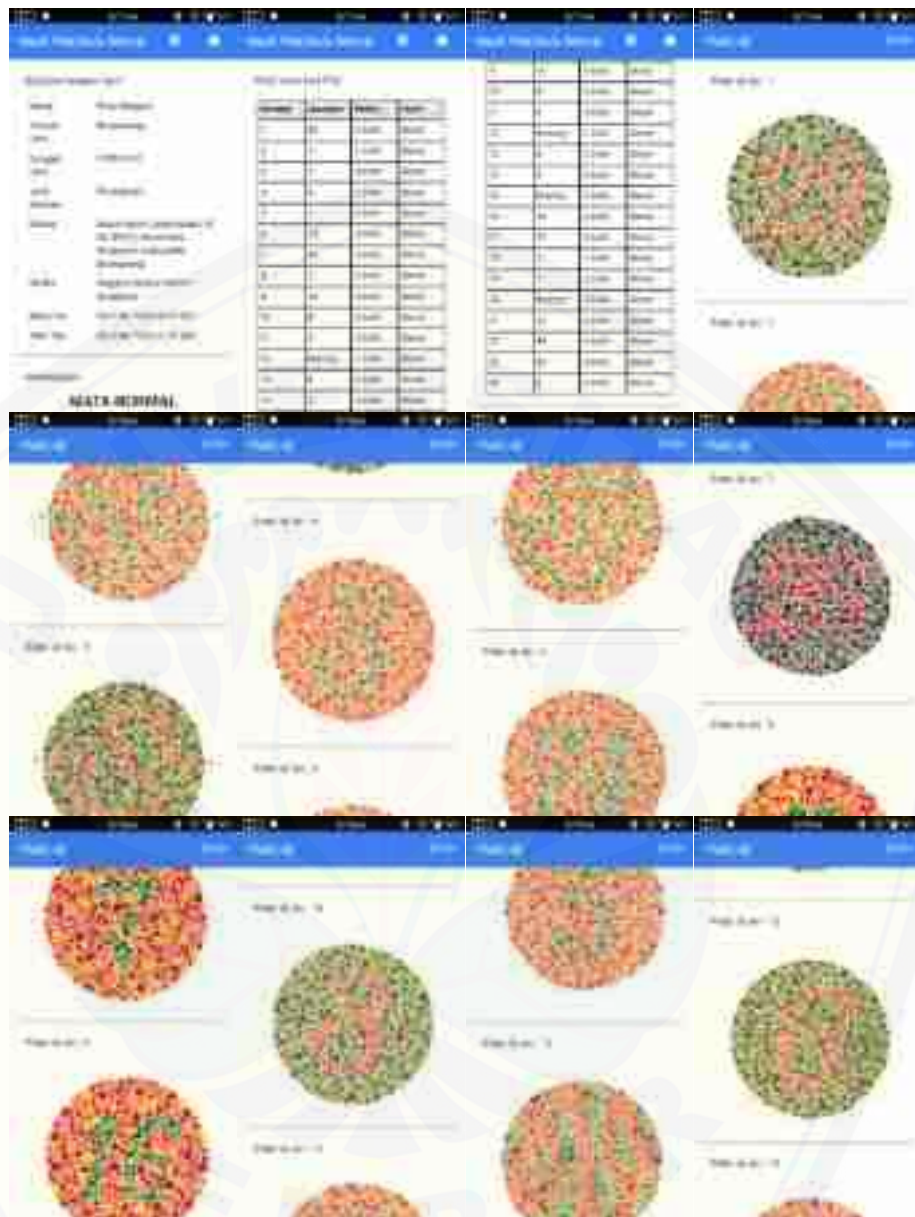


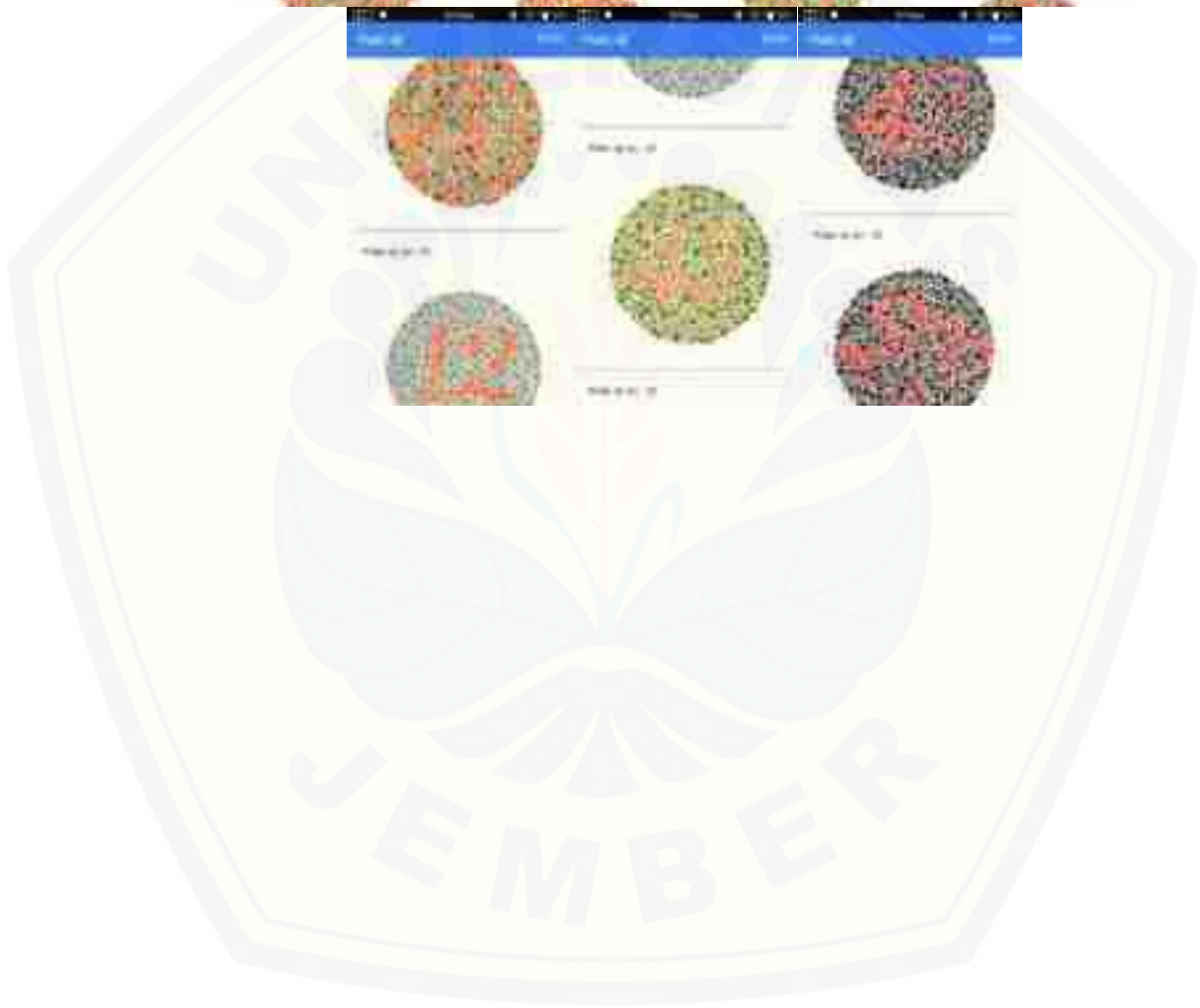
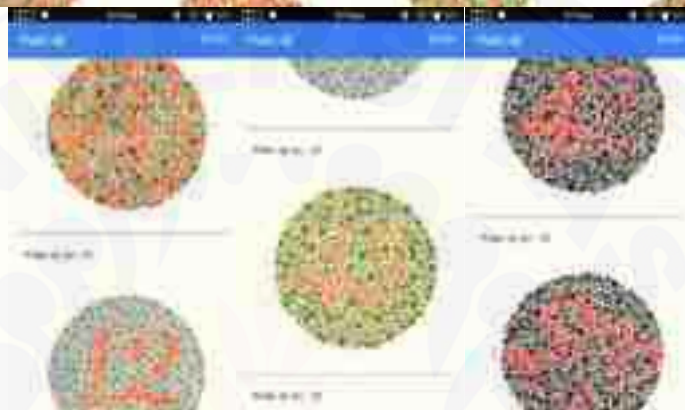
14. Septian Nanda Perdana



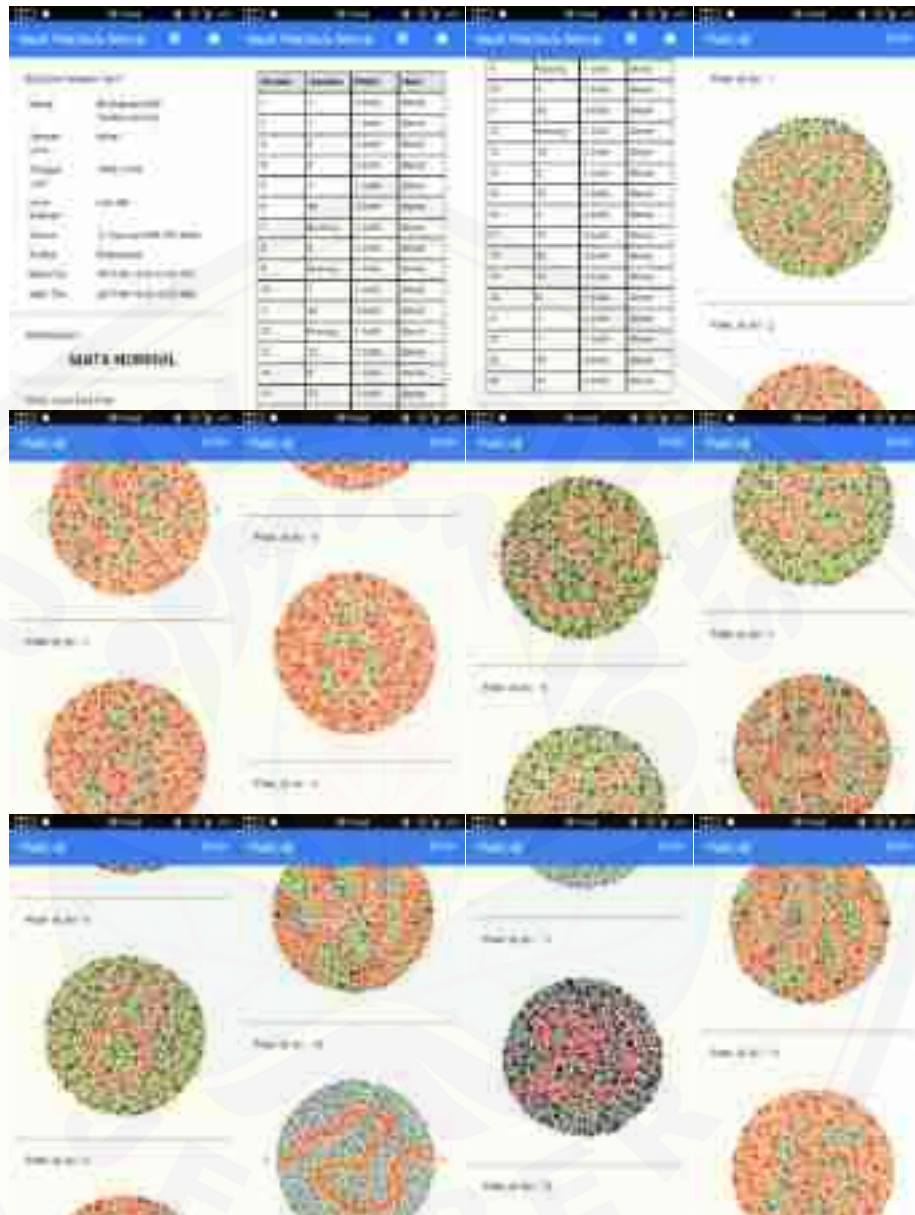


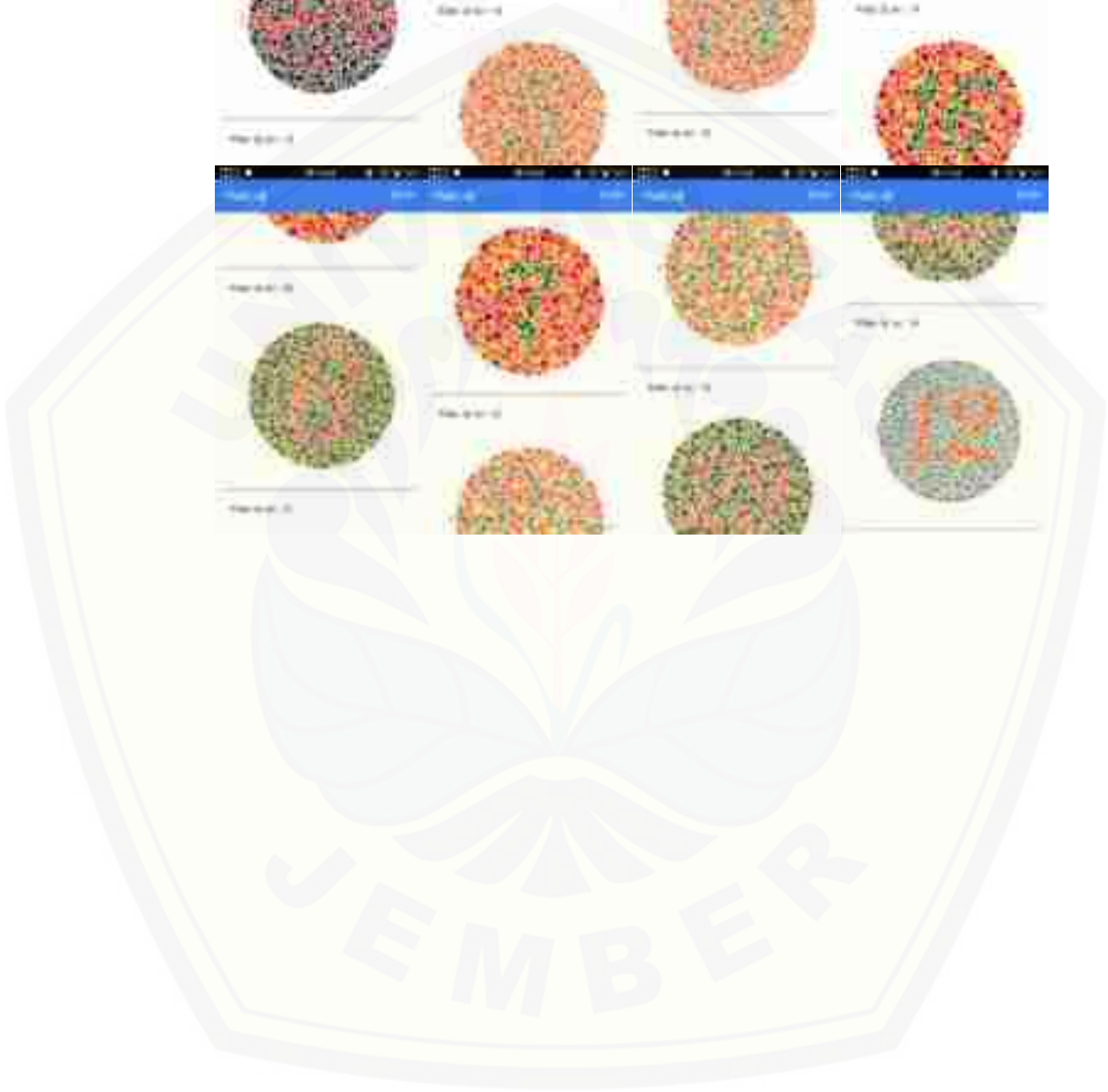
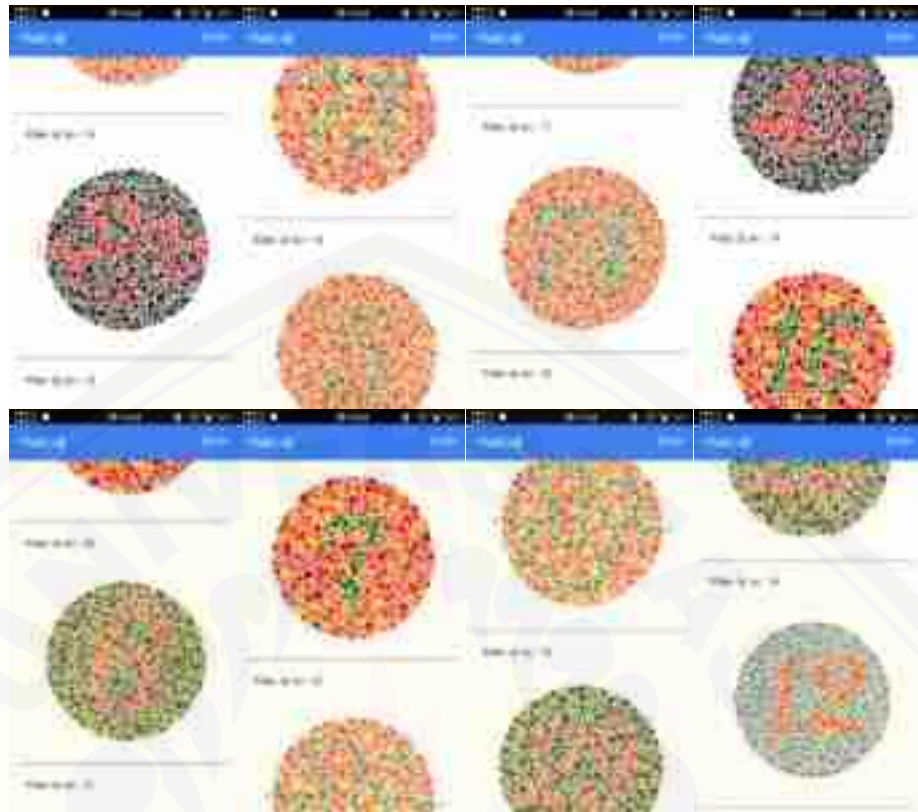
15. Pipin Setiyanti



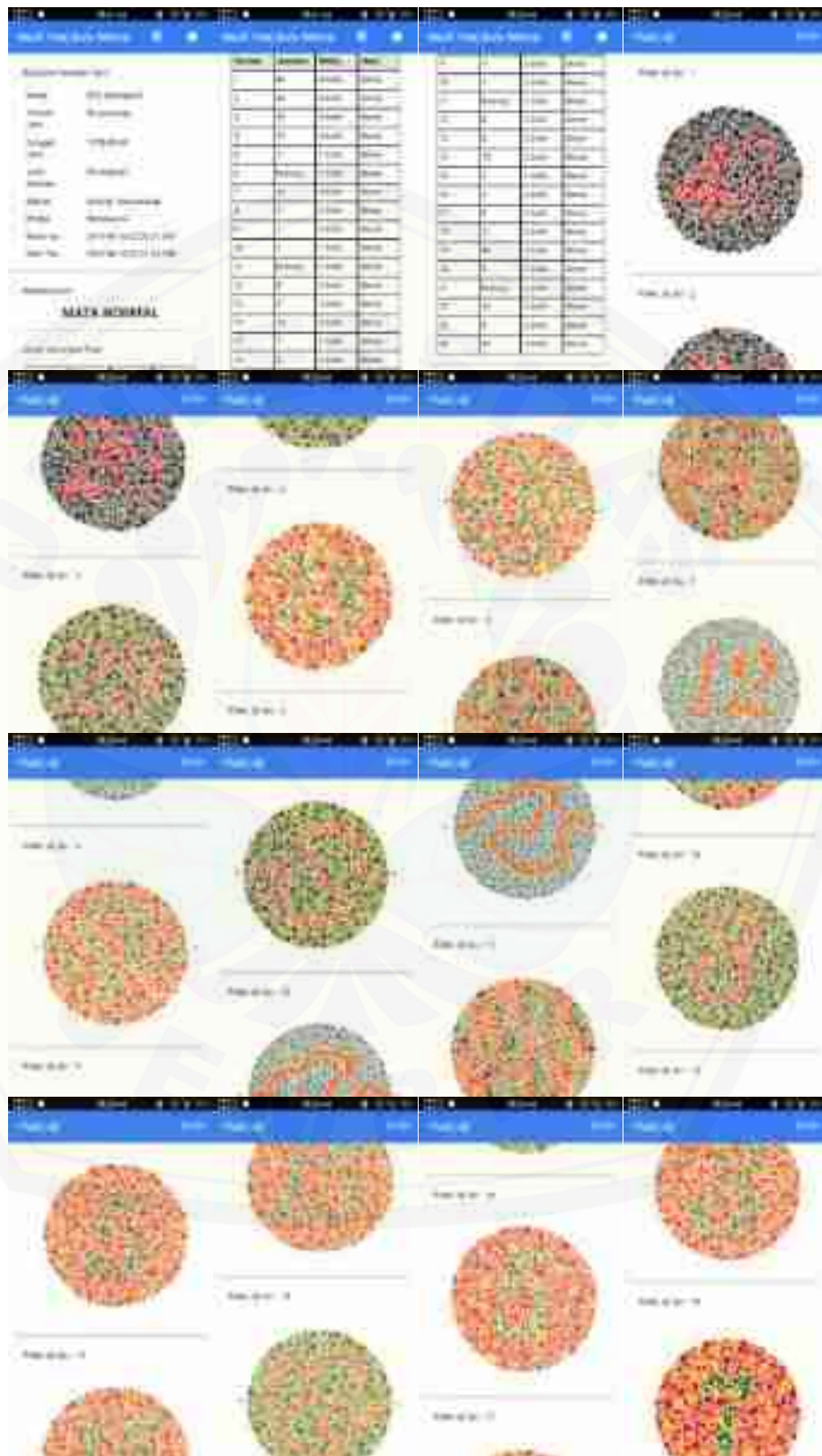


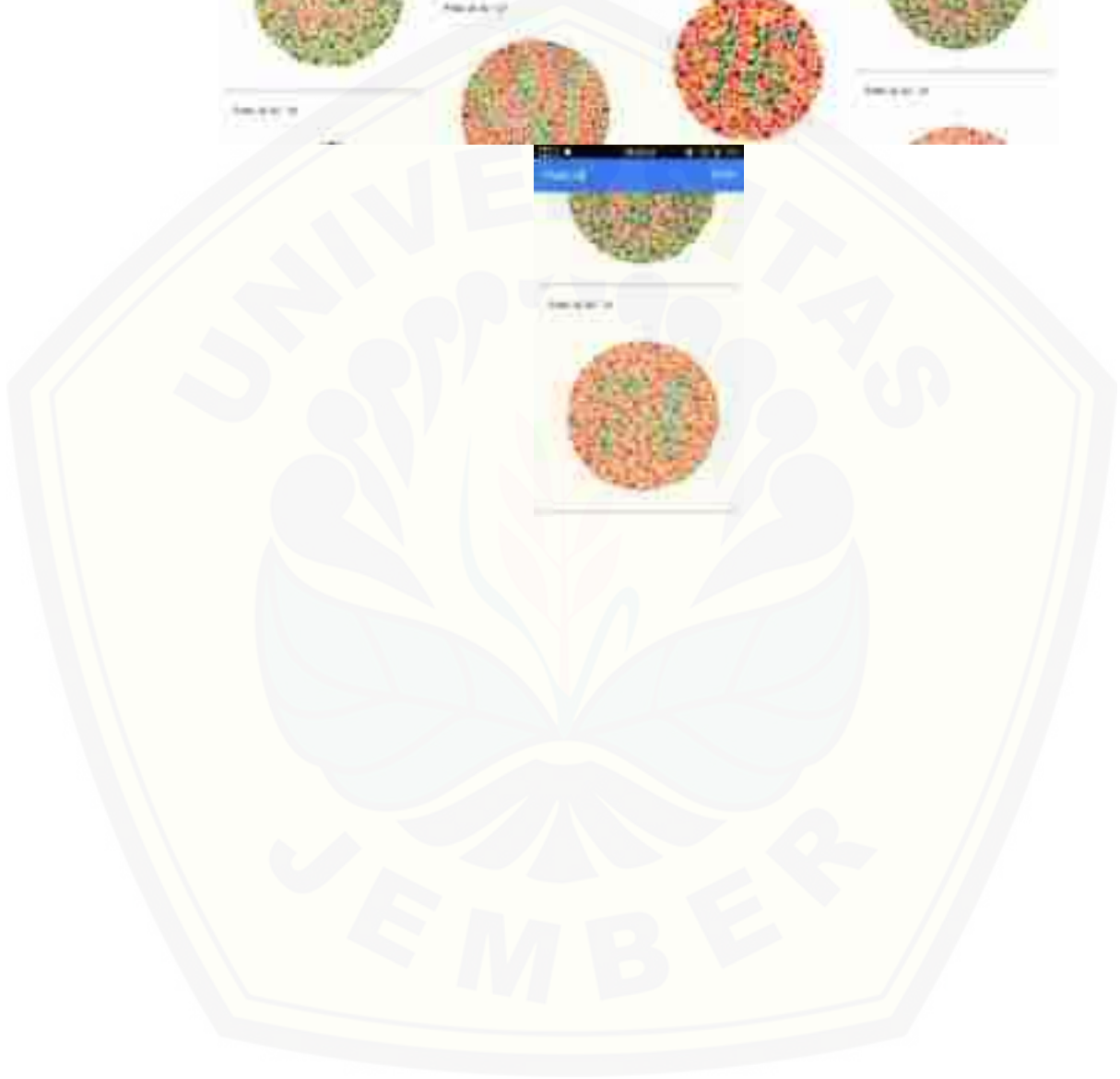
16. Mochamad Rafli Tuafikurrohman



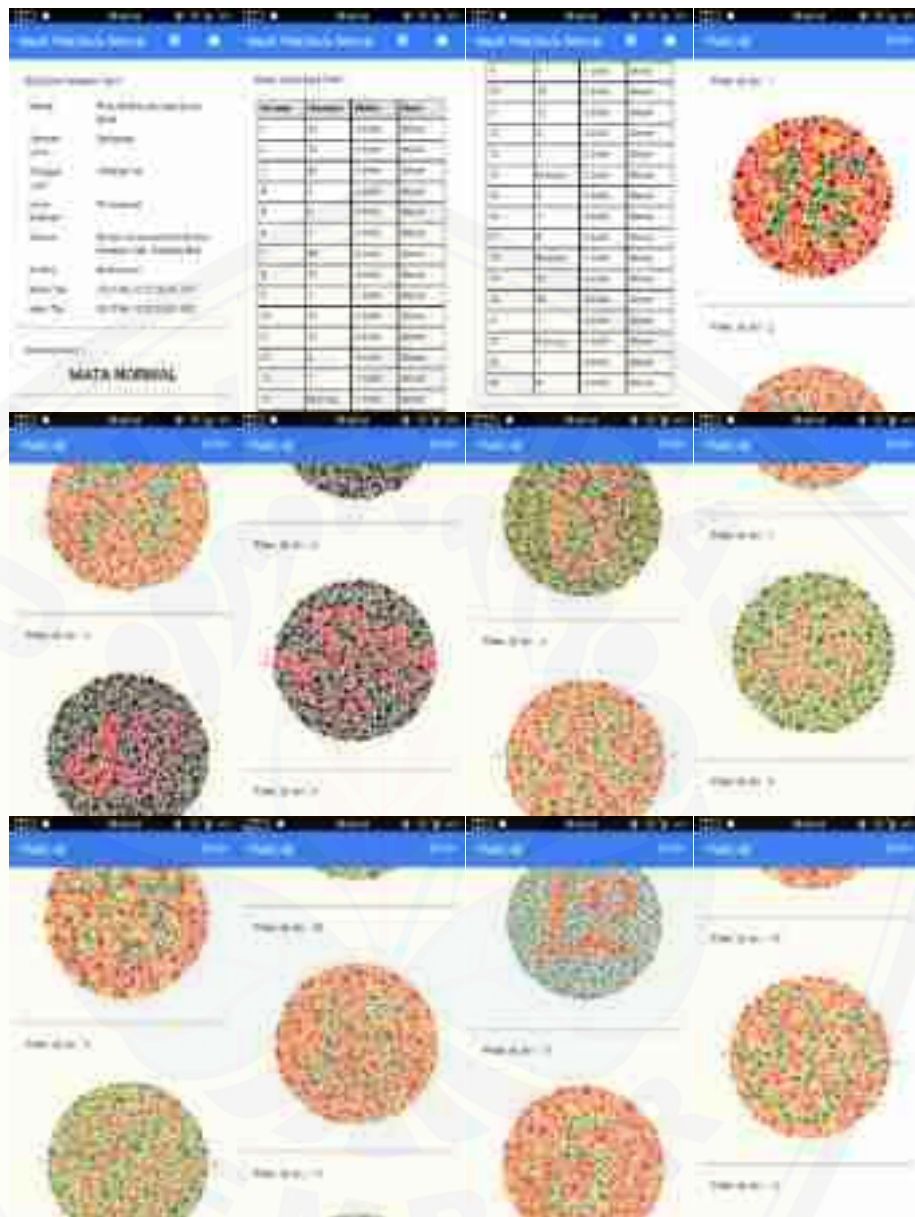


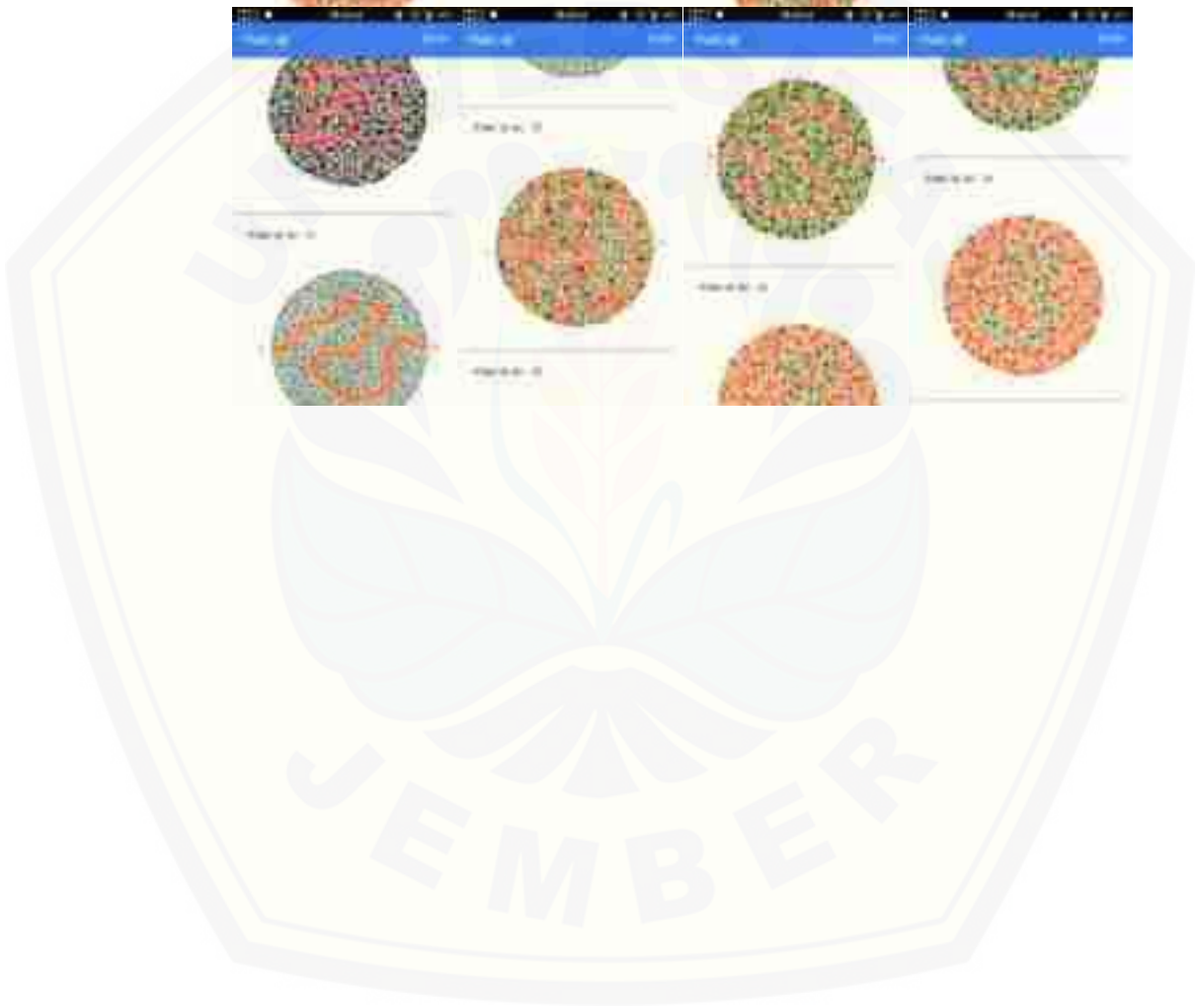
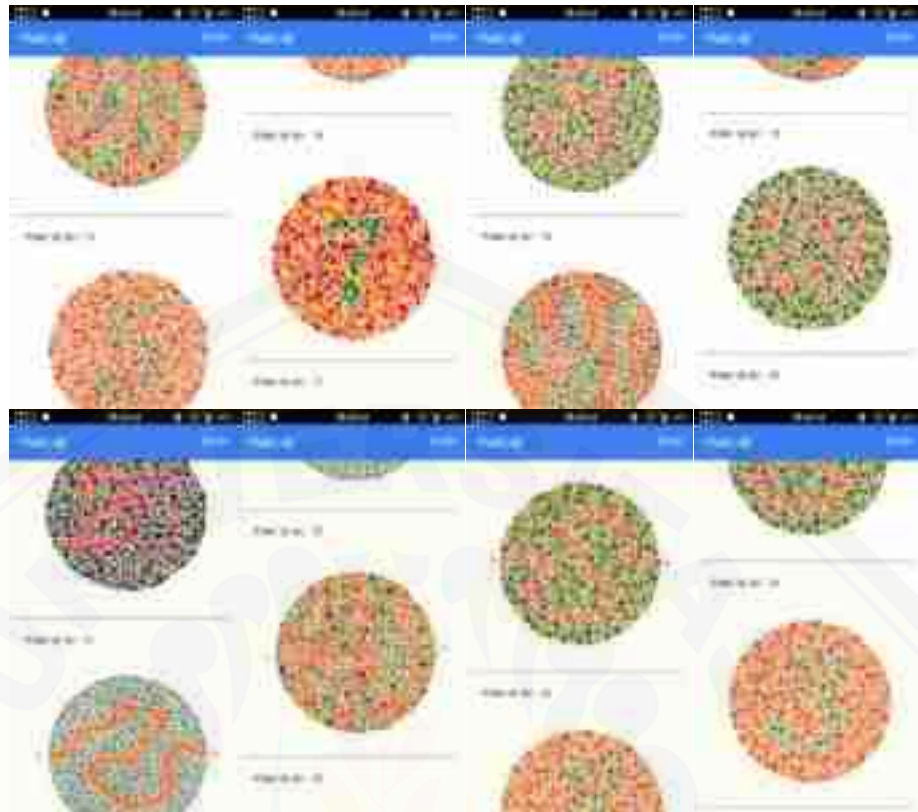
17. Ulfa Rahmawati





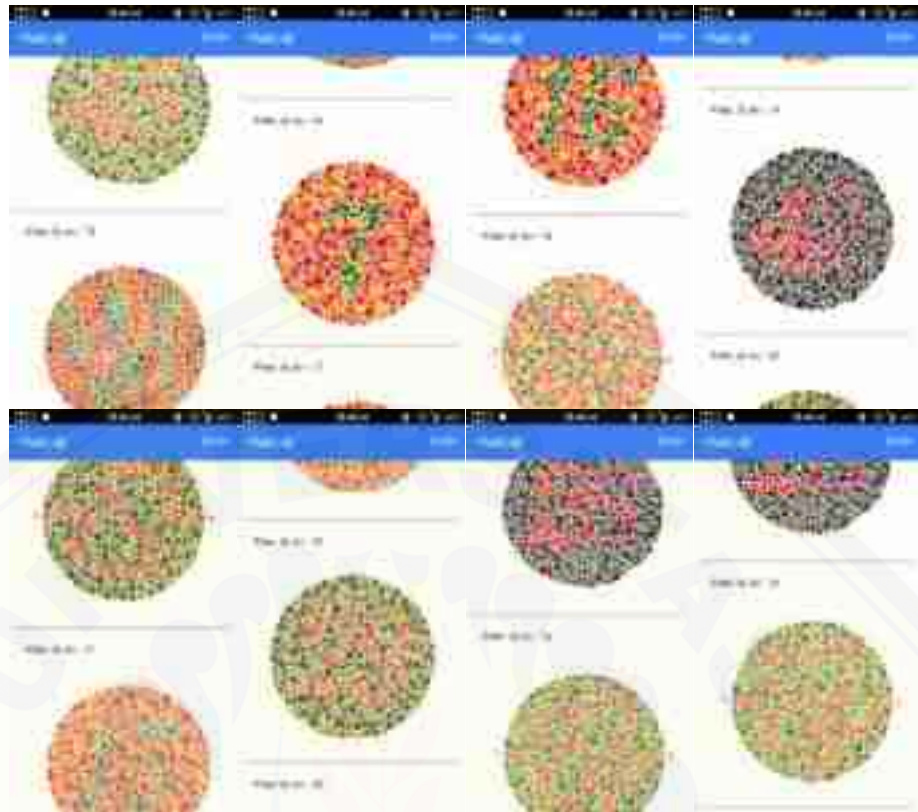
18. Putu Shintia Apriana Sucia Dewi



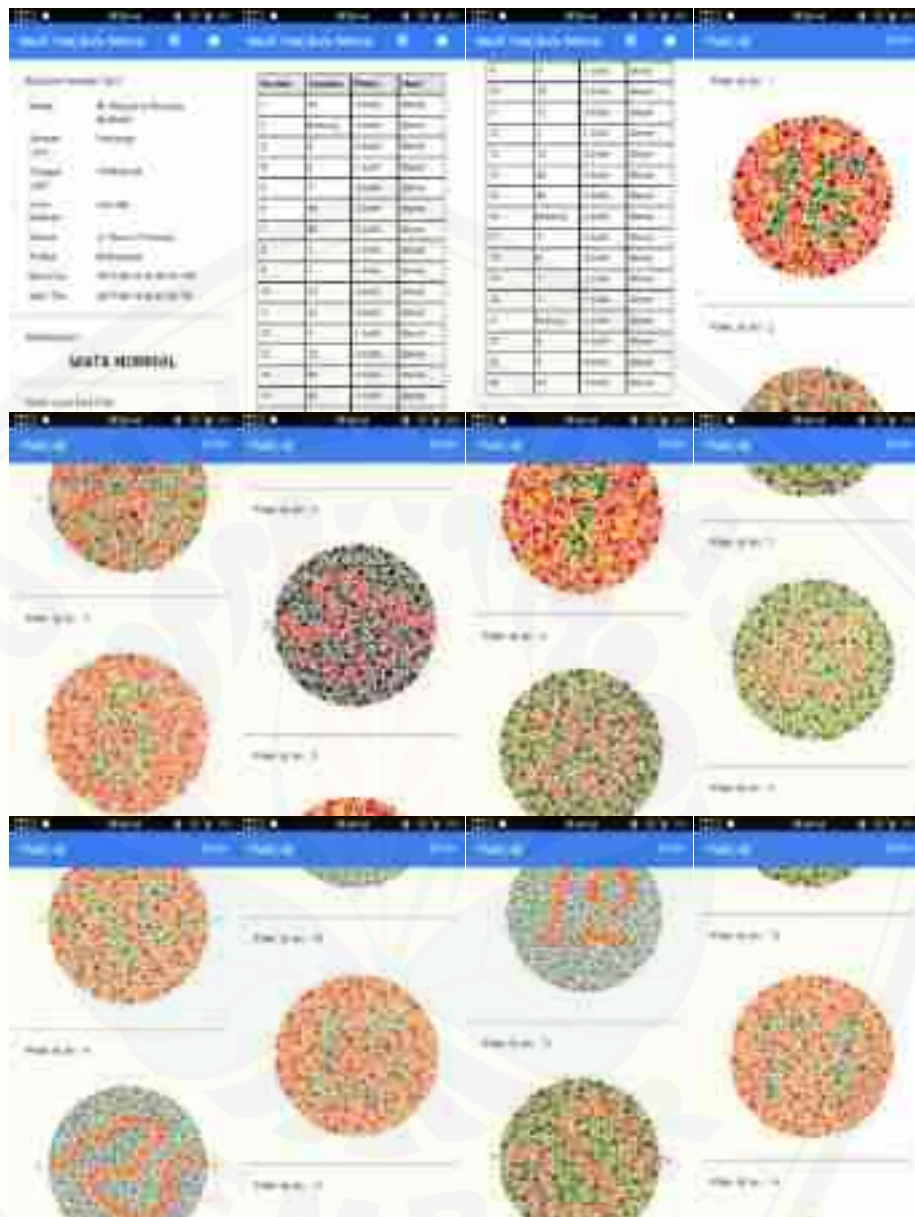


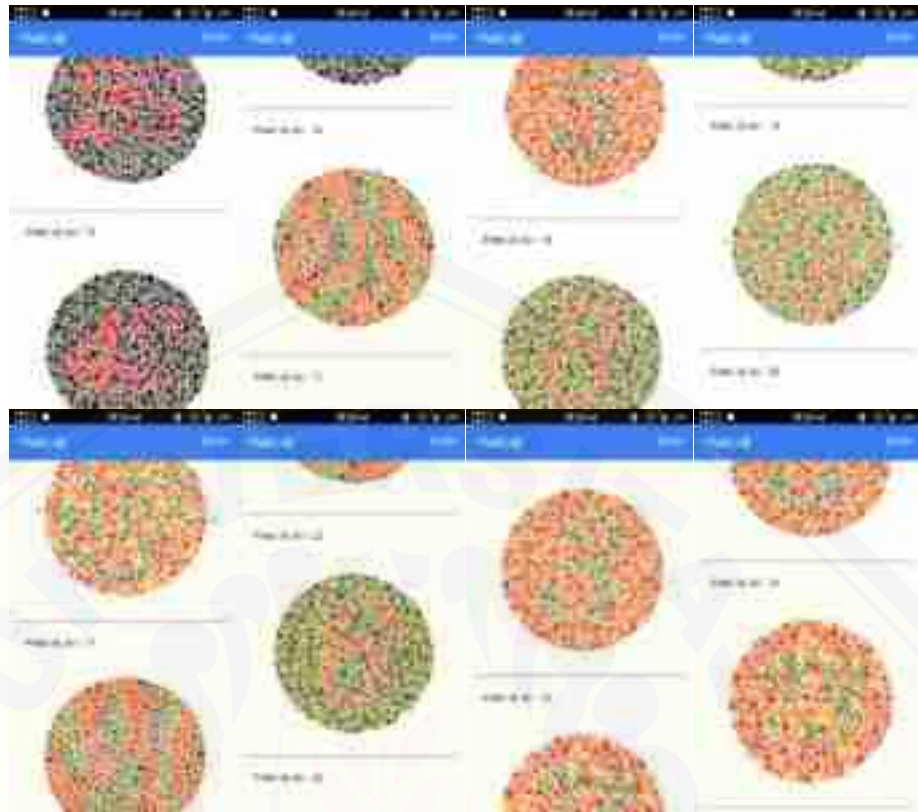
19. Sindi Dwi Agustin





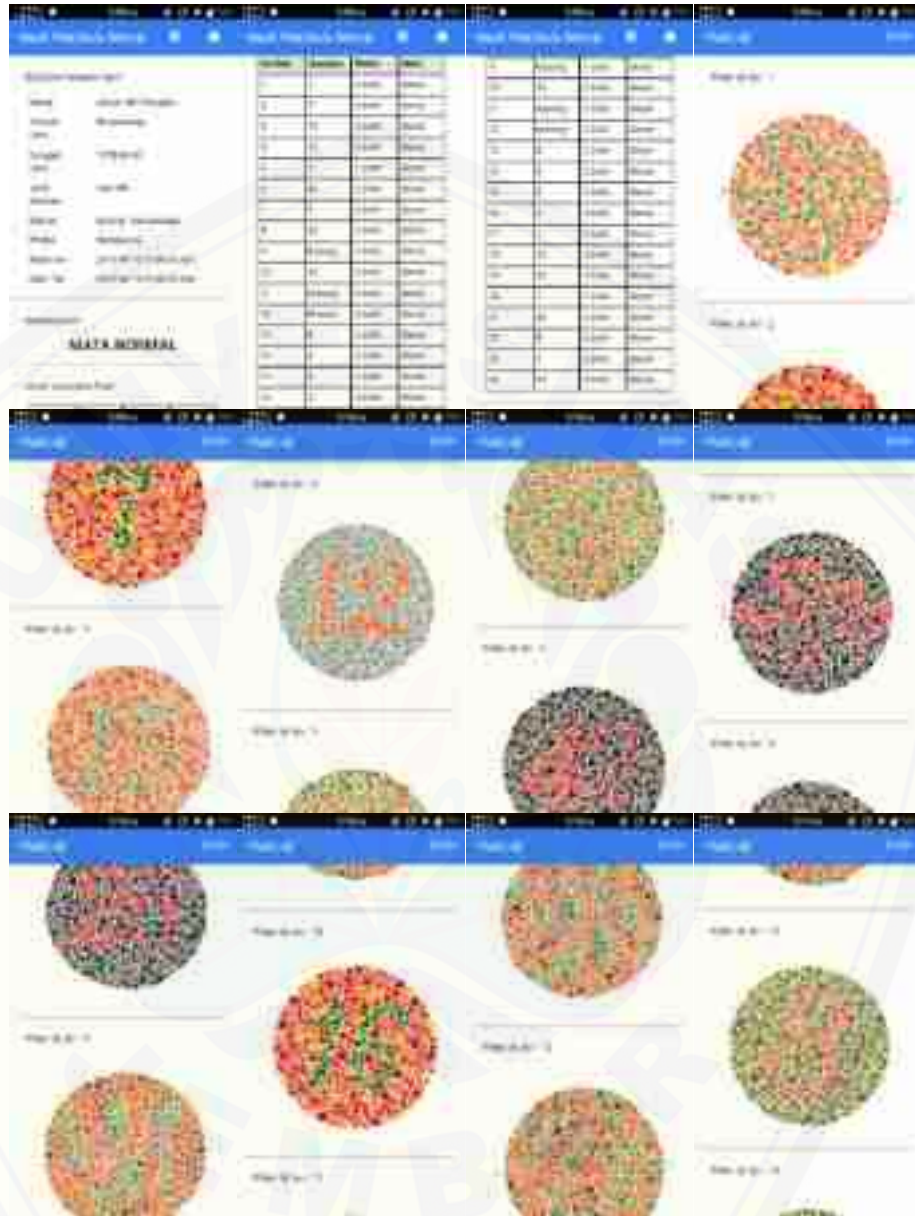
20. M, Khuzaimy Rurroziq Basthoni

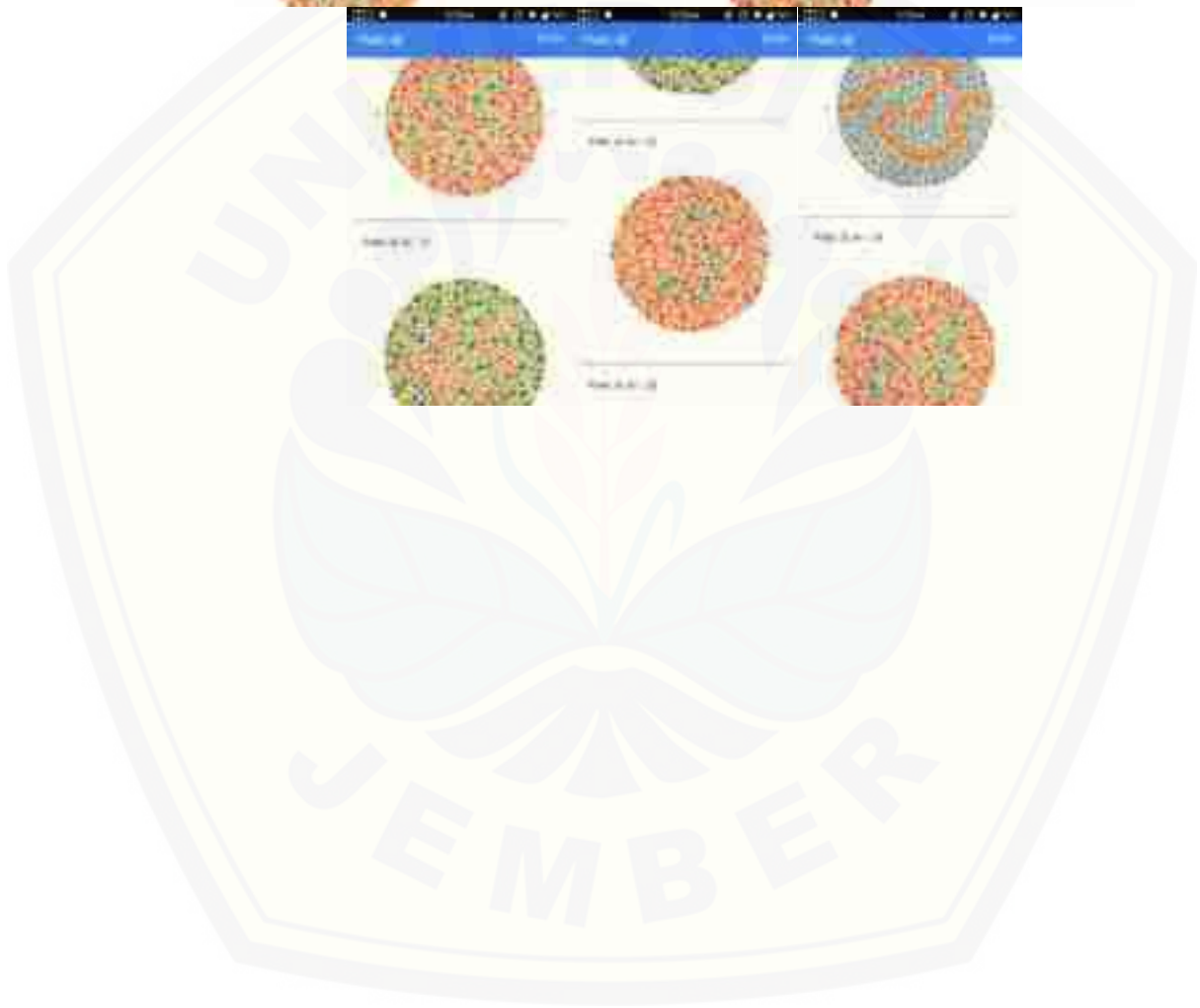
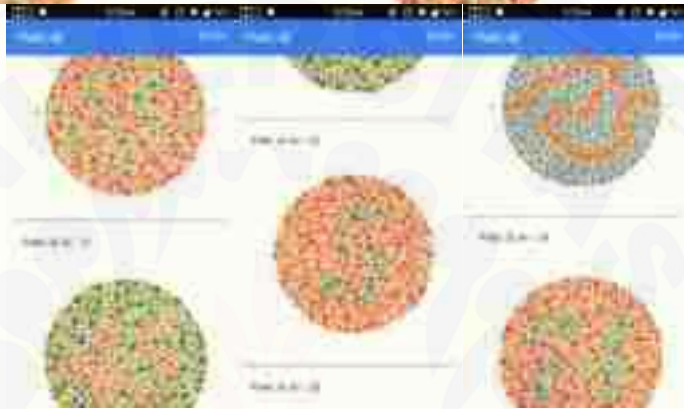




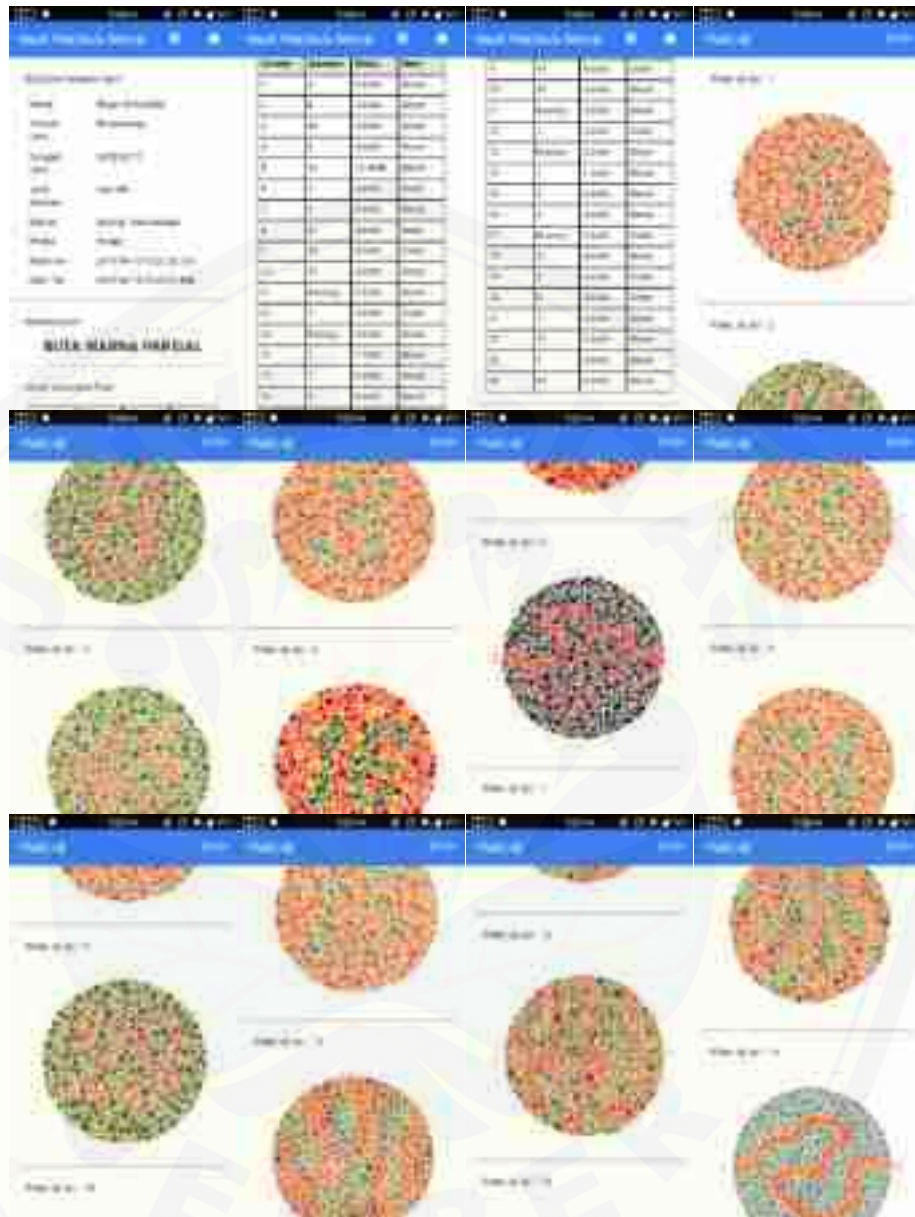
B.2 Tes Kedua

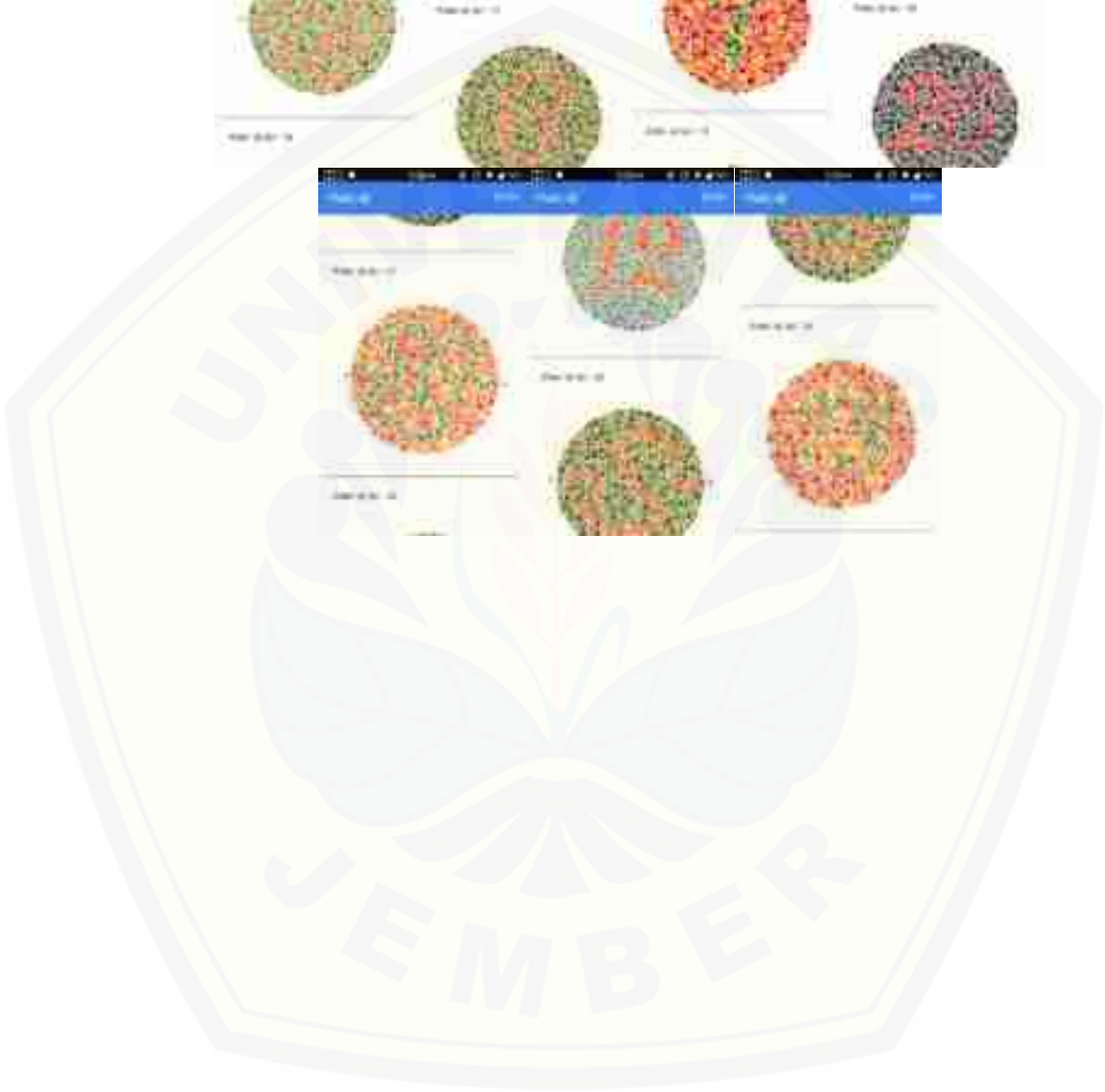
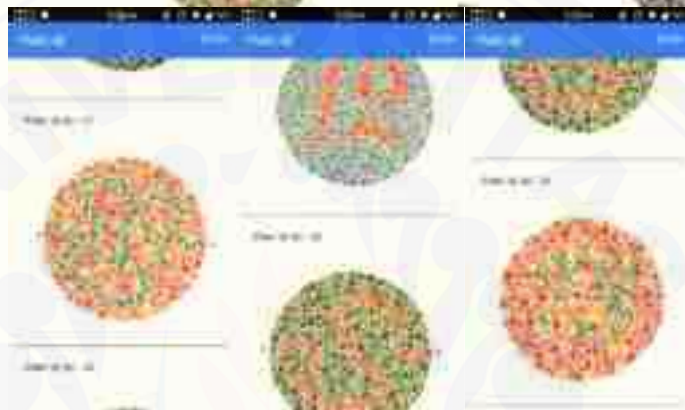
1. Abrian Eko Prasetyo



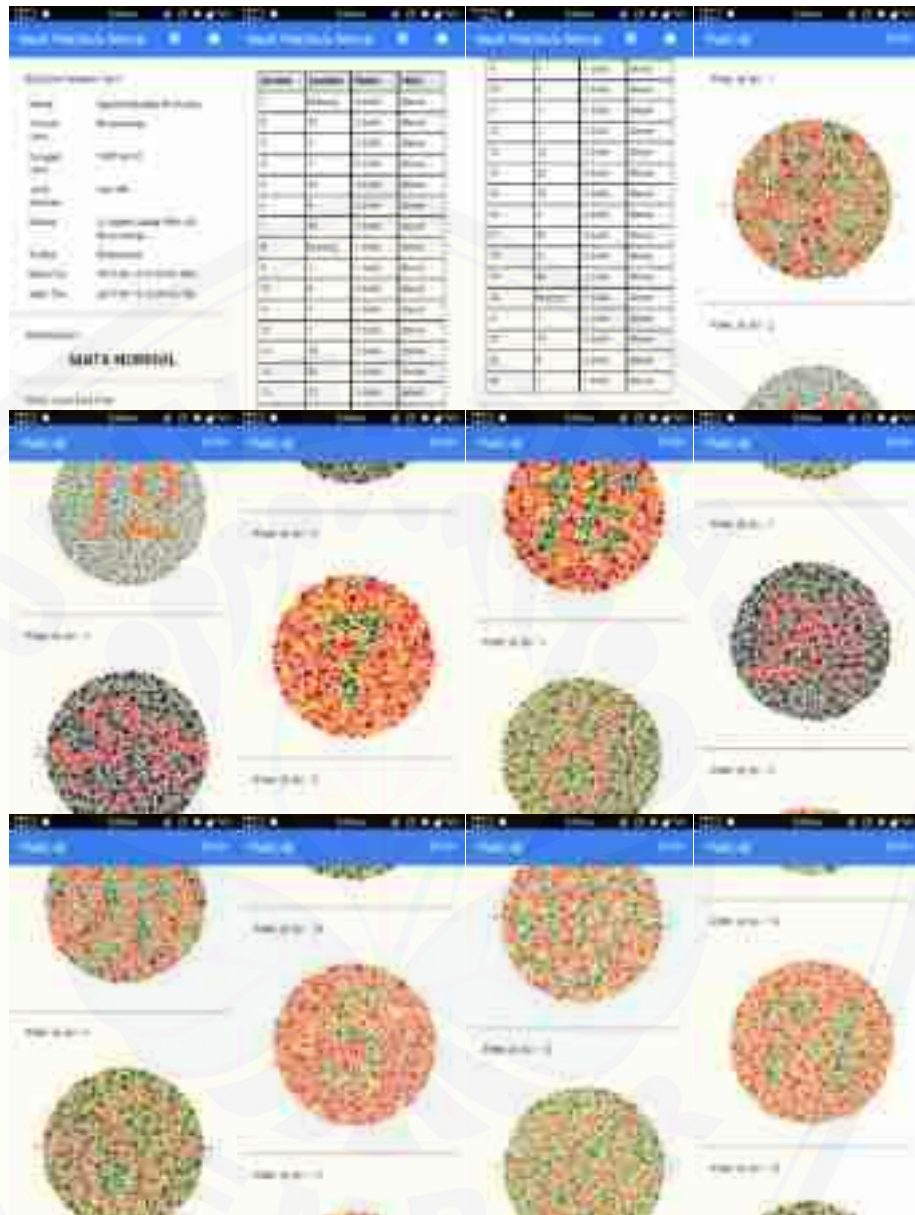


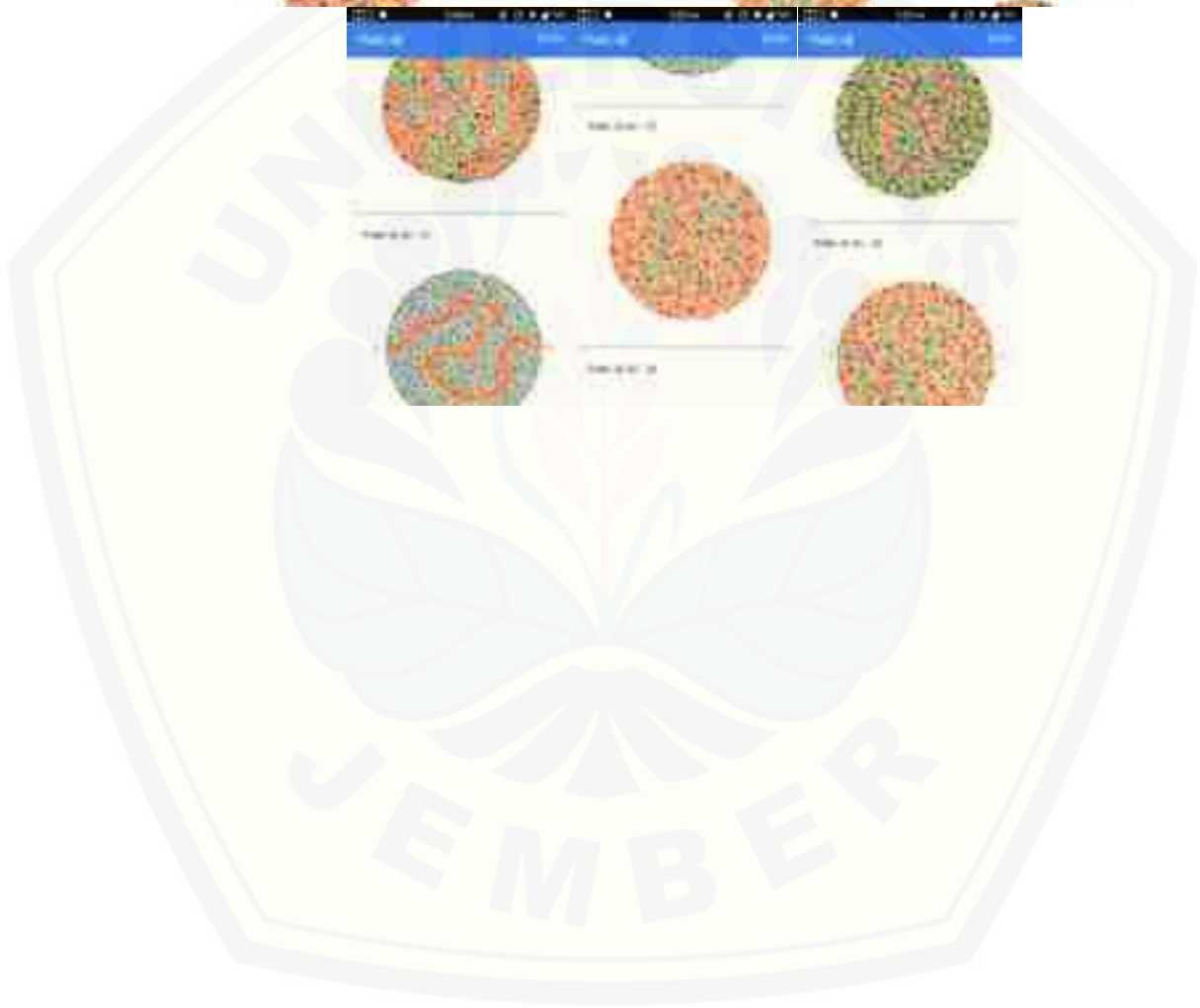
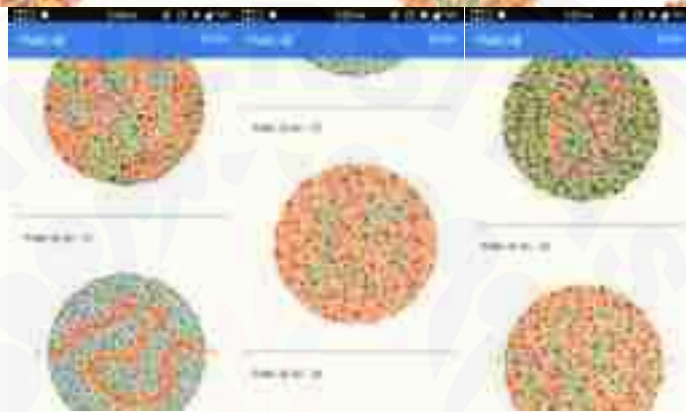
2. Bagas Di Saputra



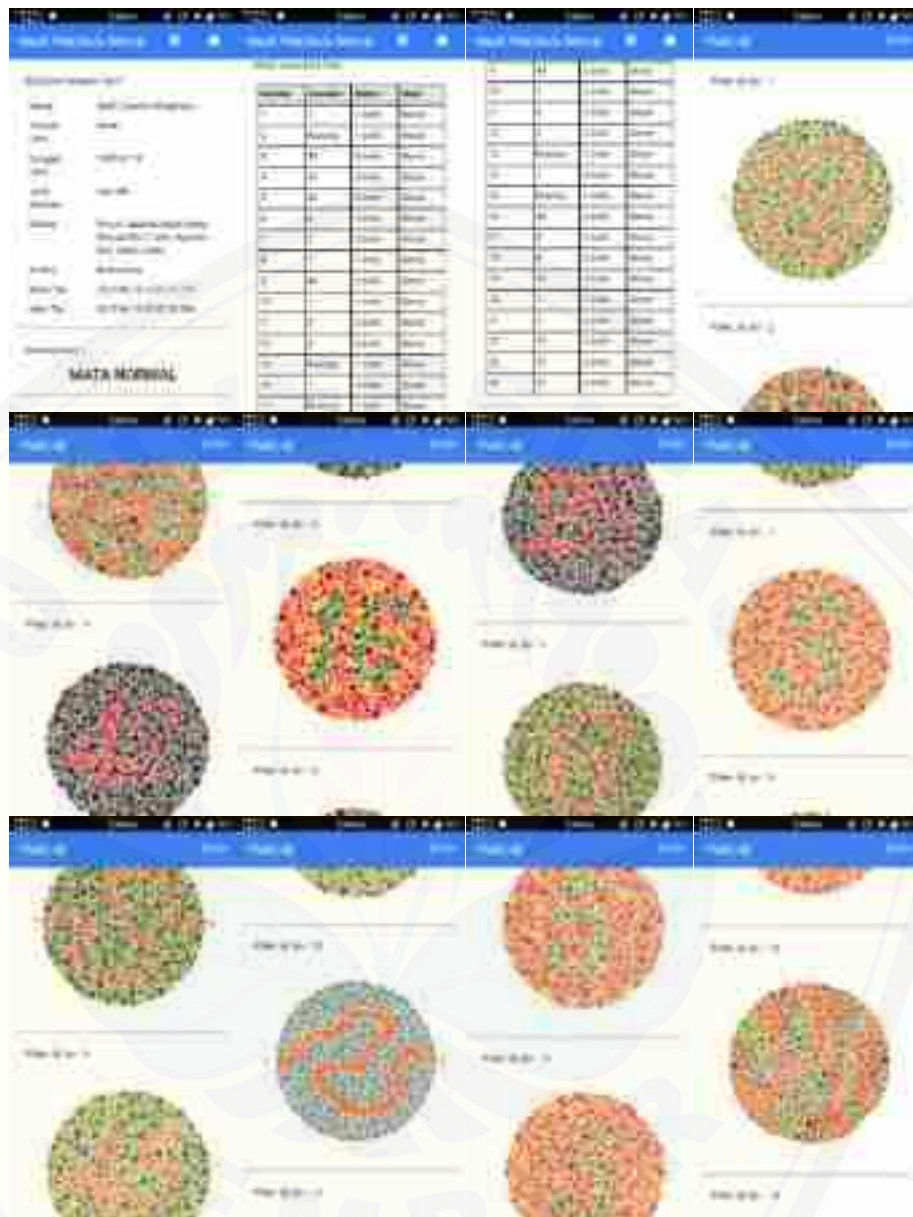


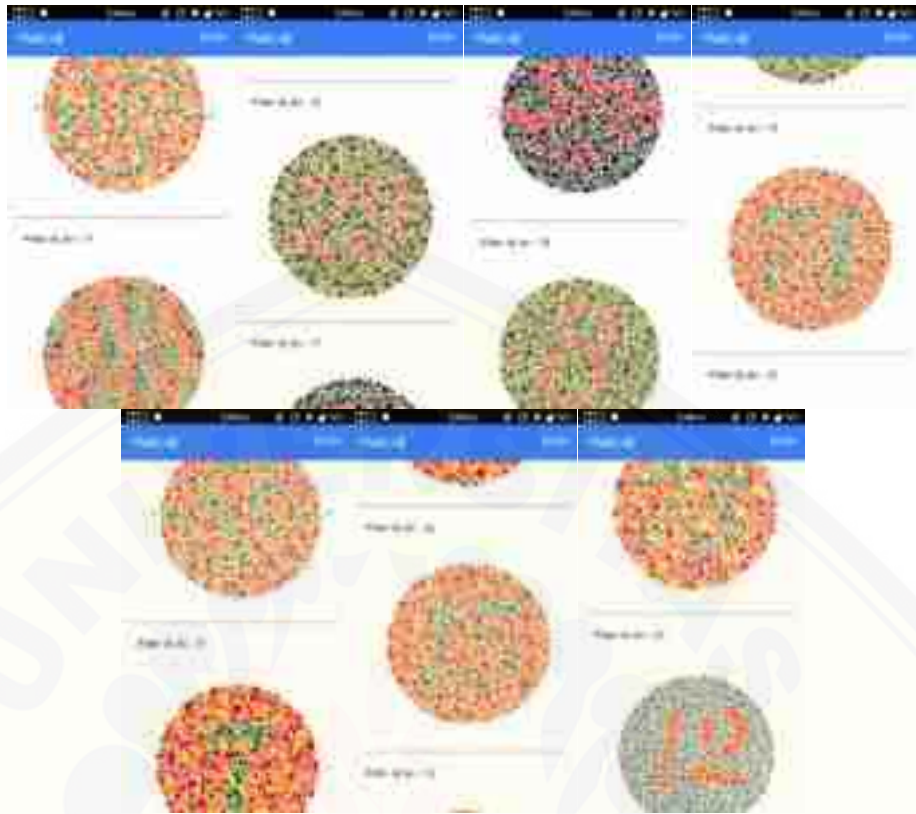
3. Agusty Maulana Bramasta



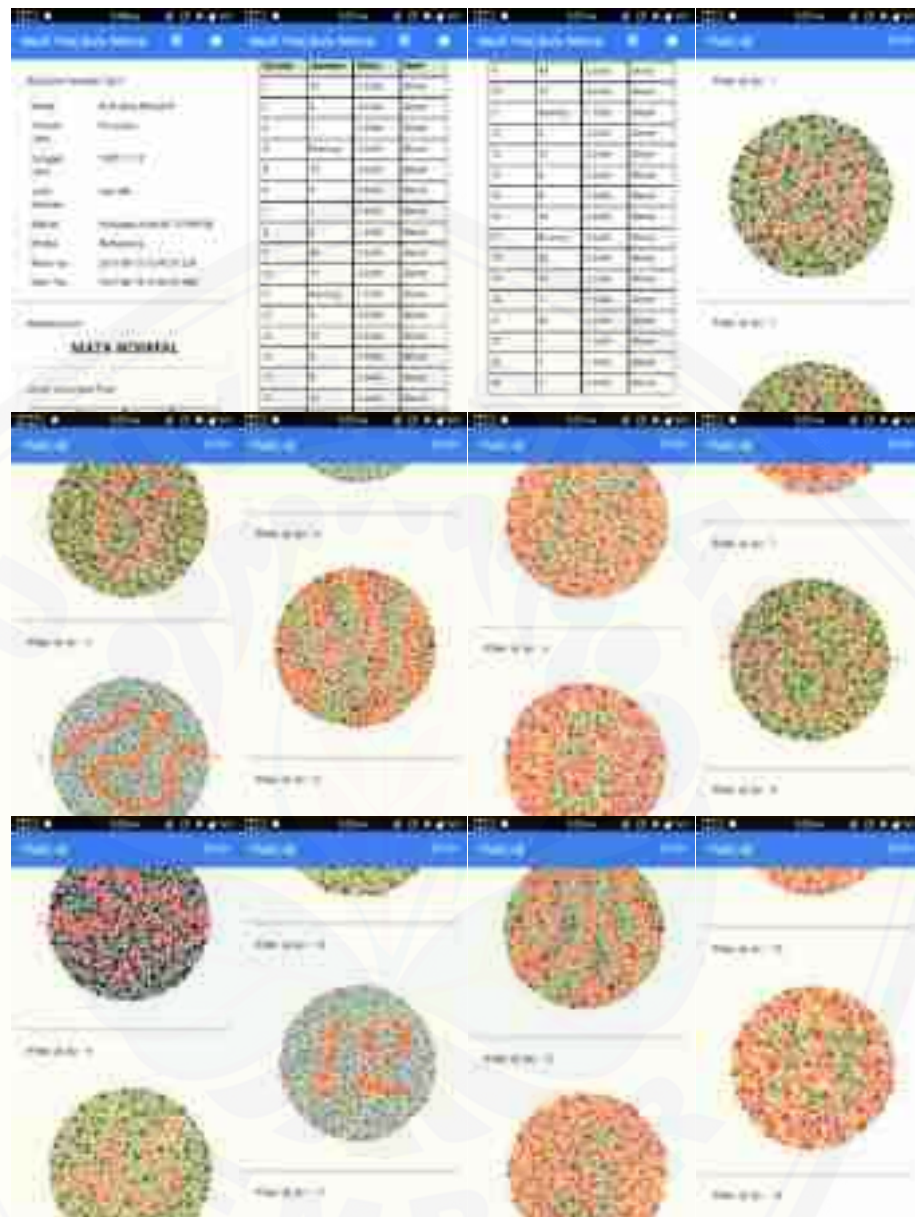


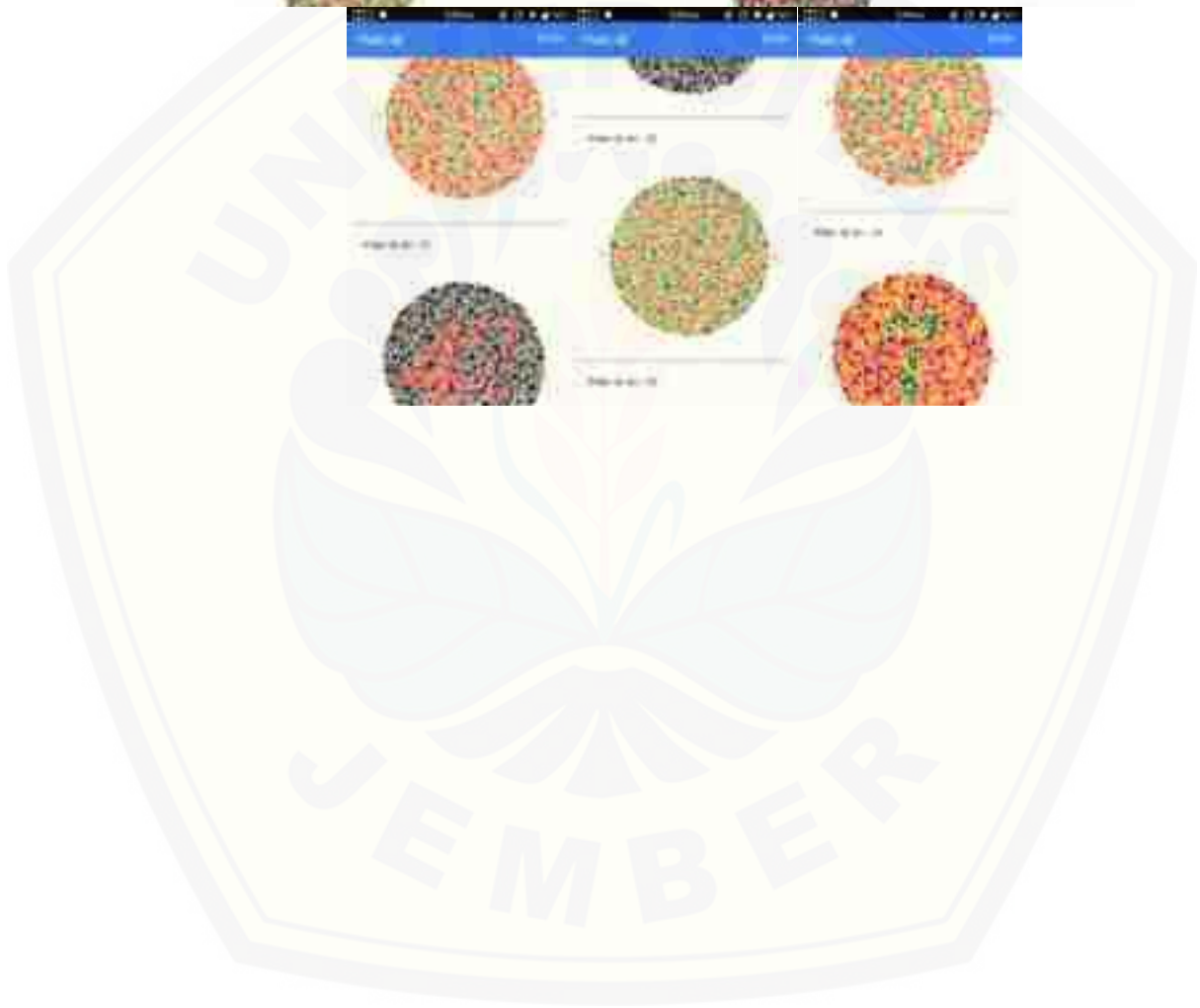
4. Galih Chandra Wiradhika



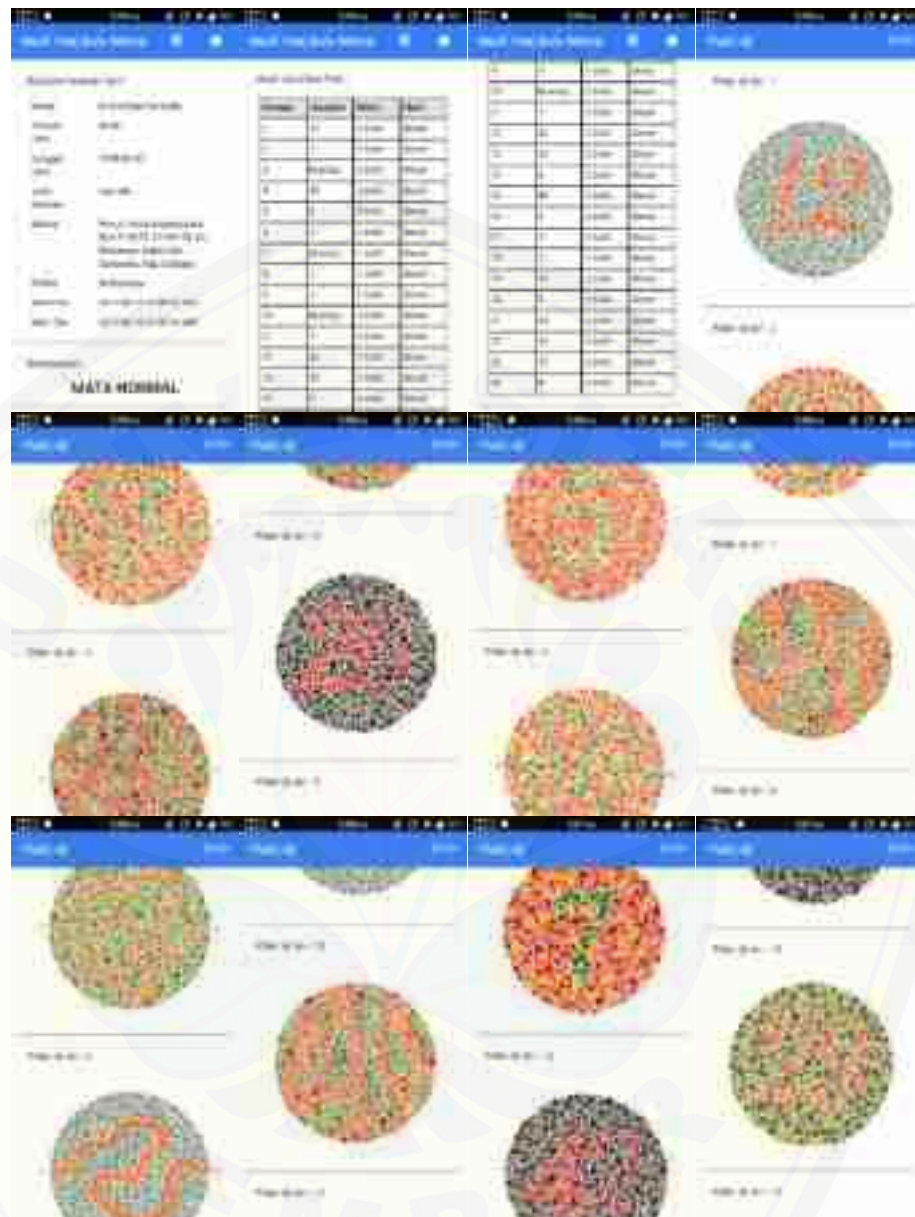


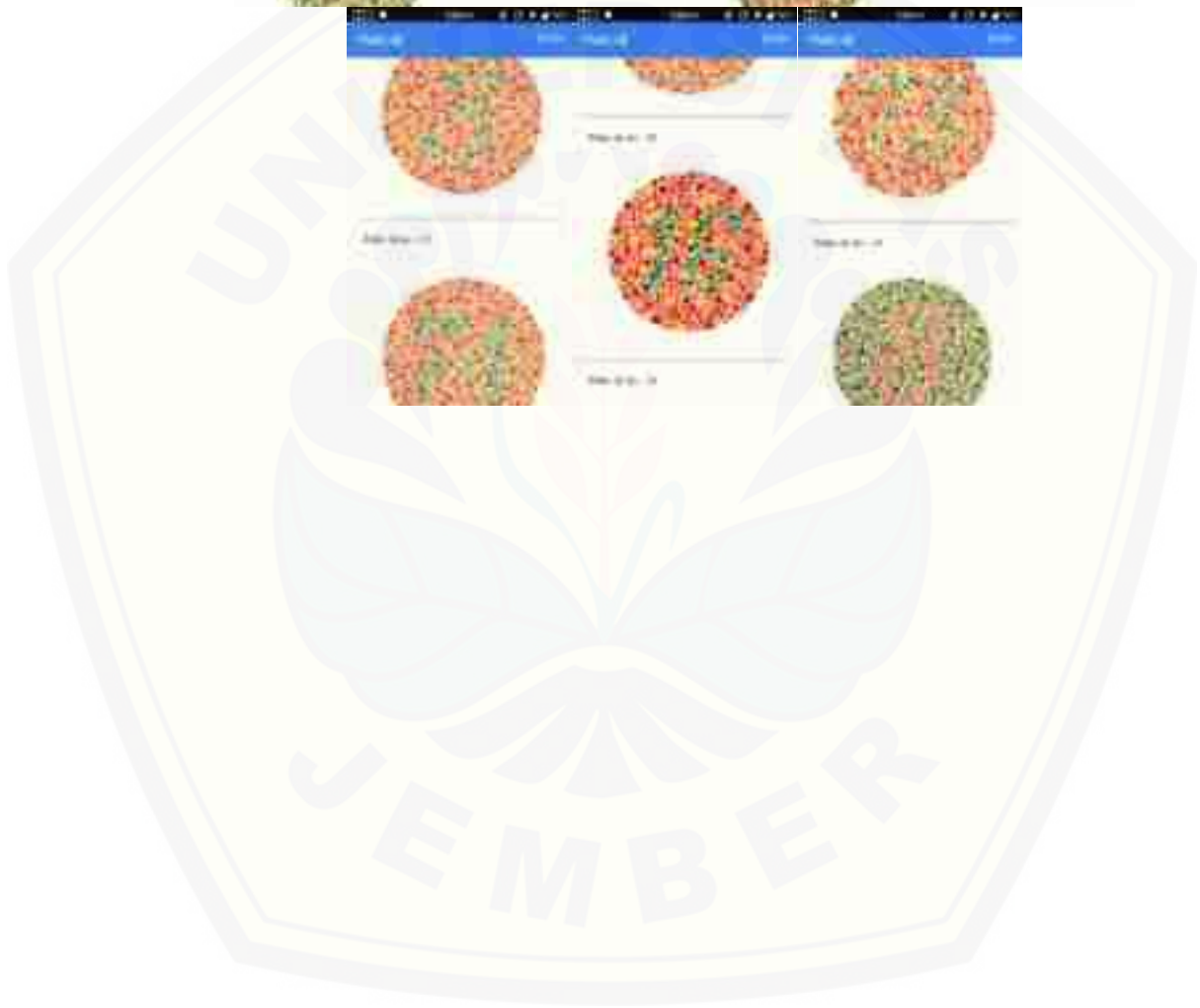
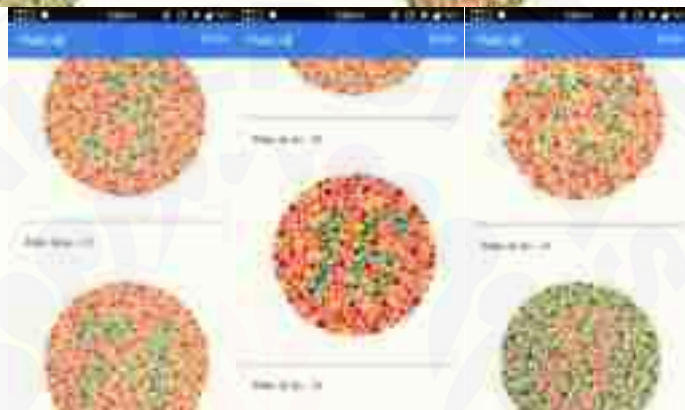
5. Kharisma Ahmad P.



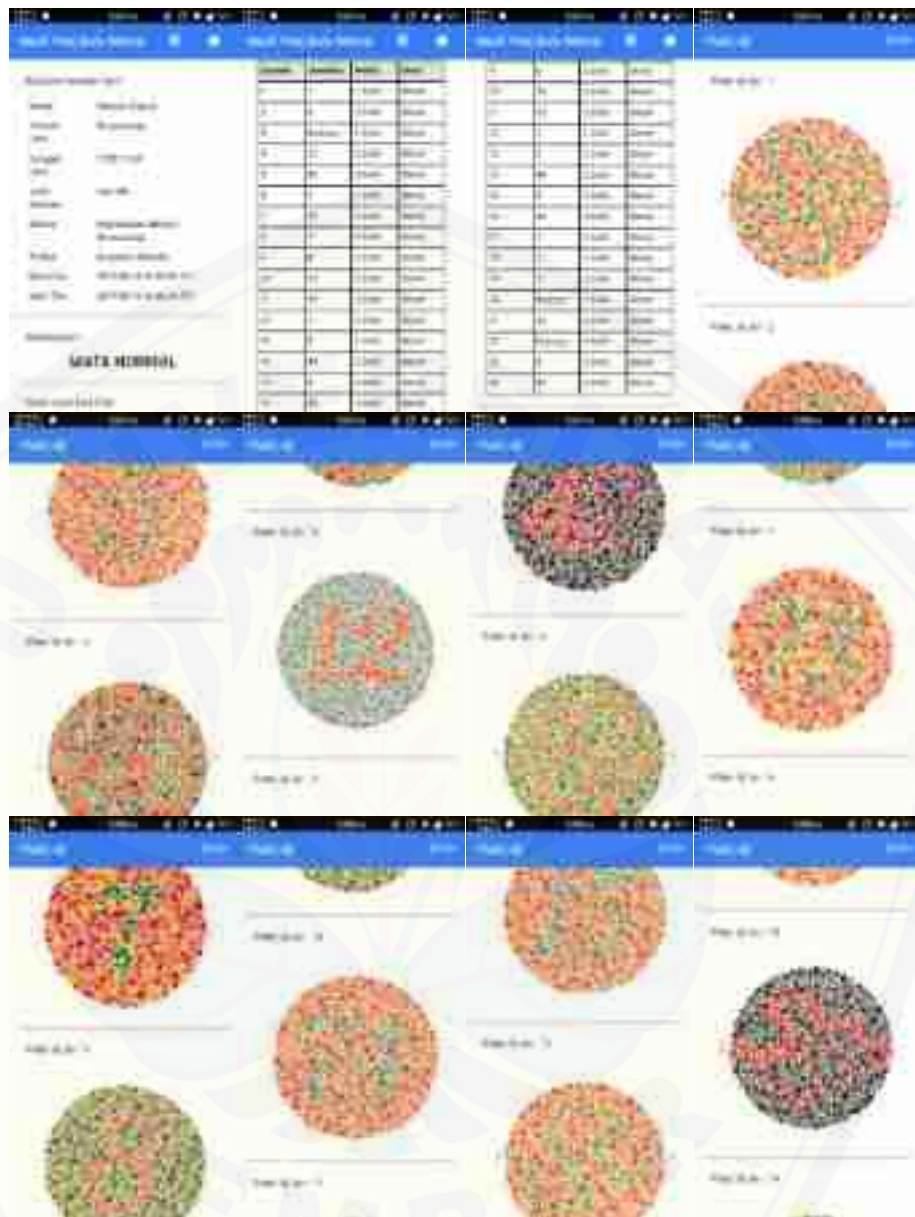


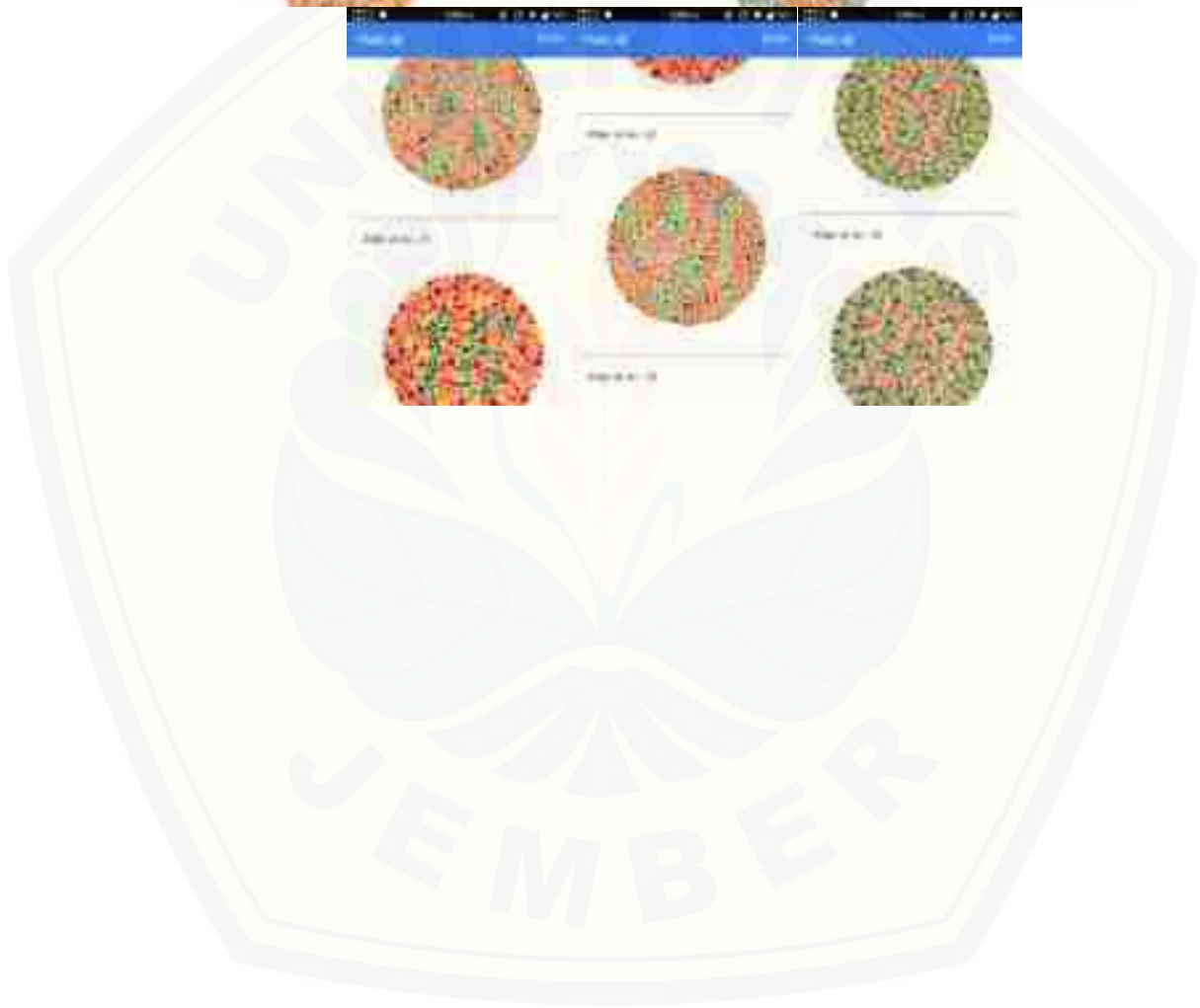
6. Emha Dinar Samudro



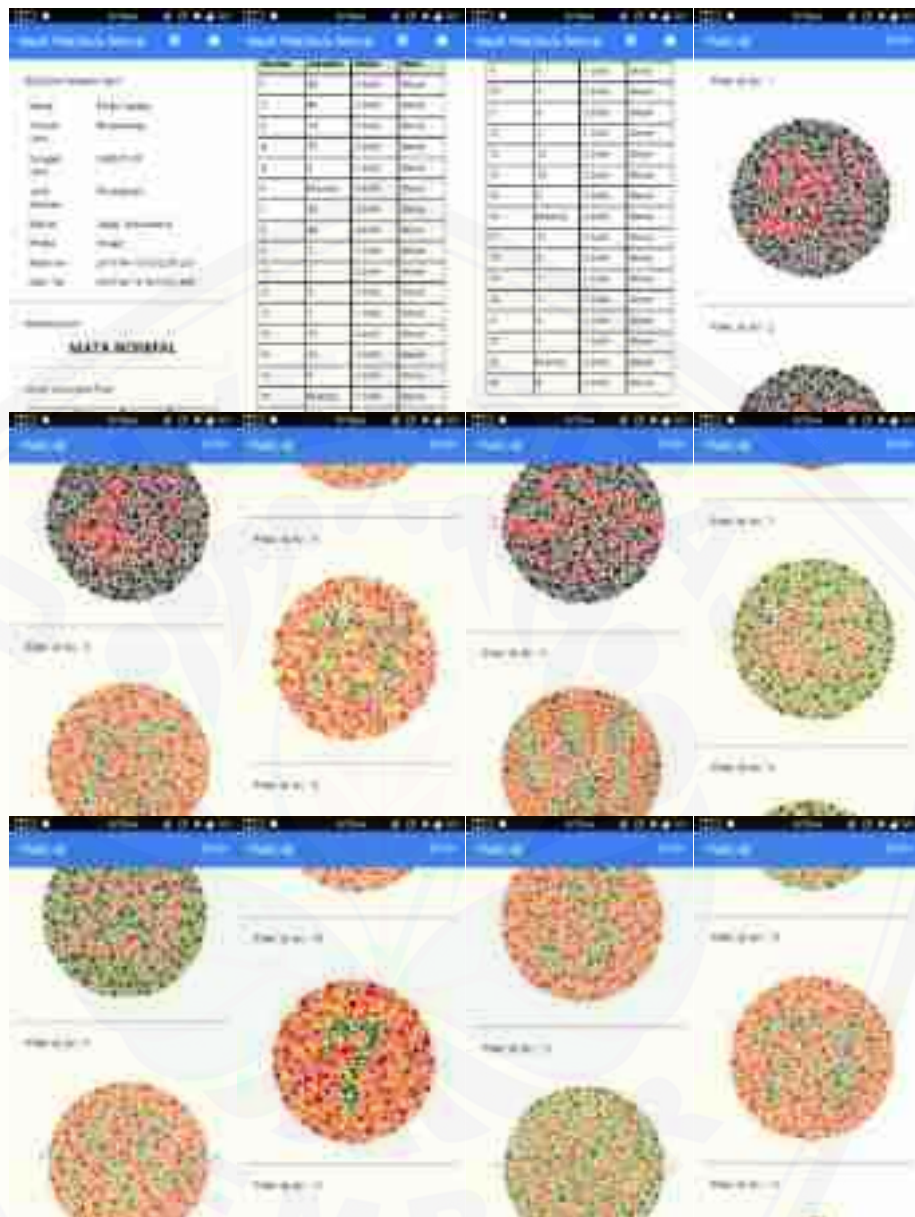


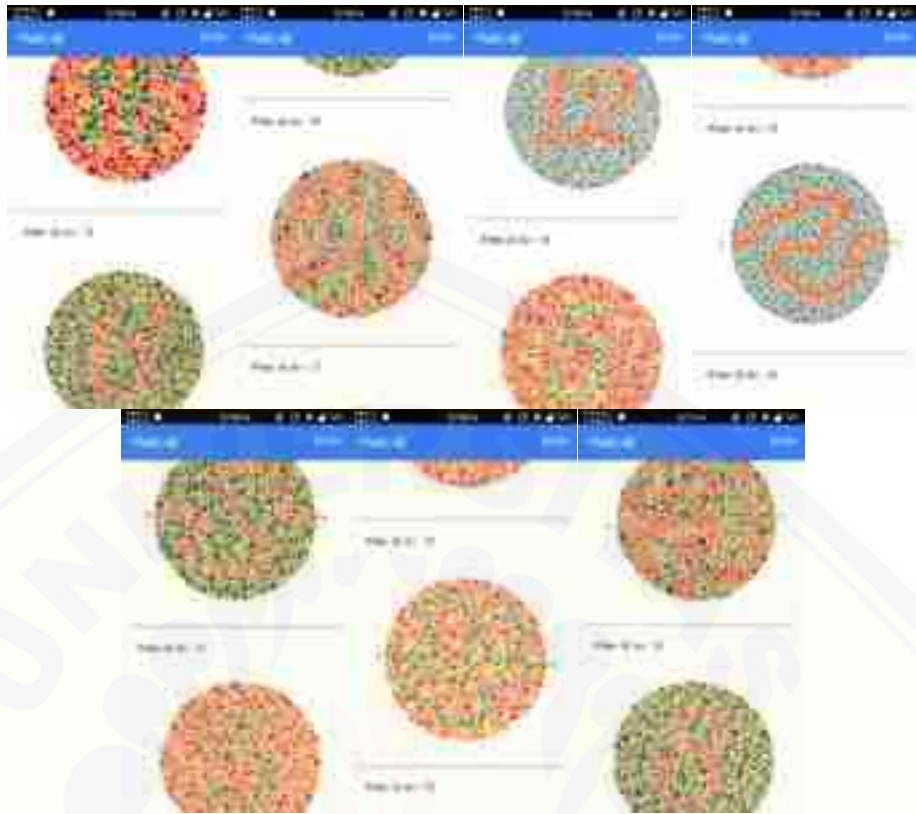
7. Novado Caesar



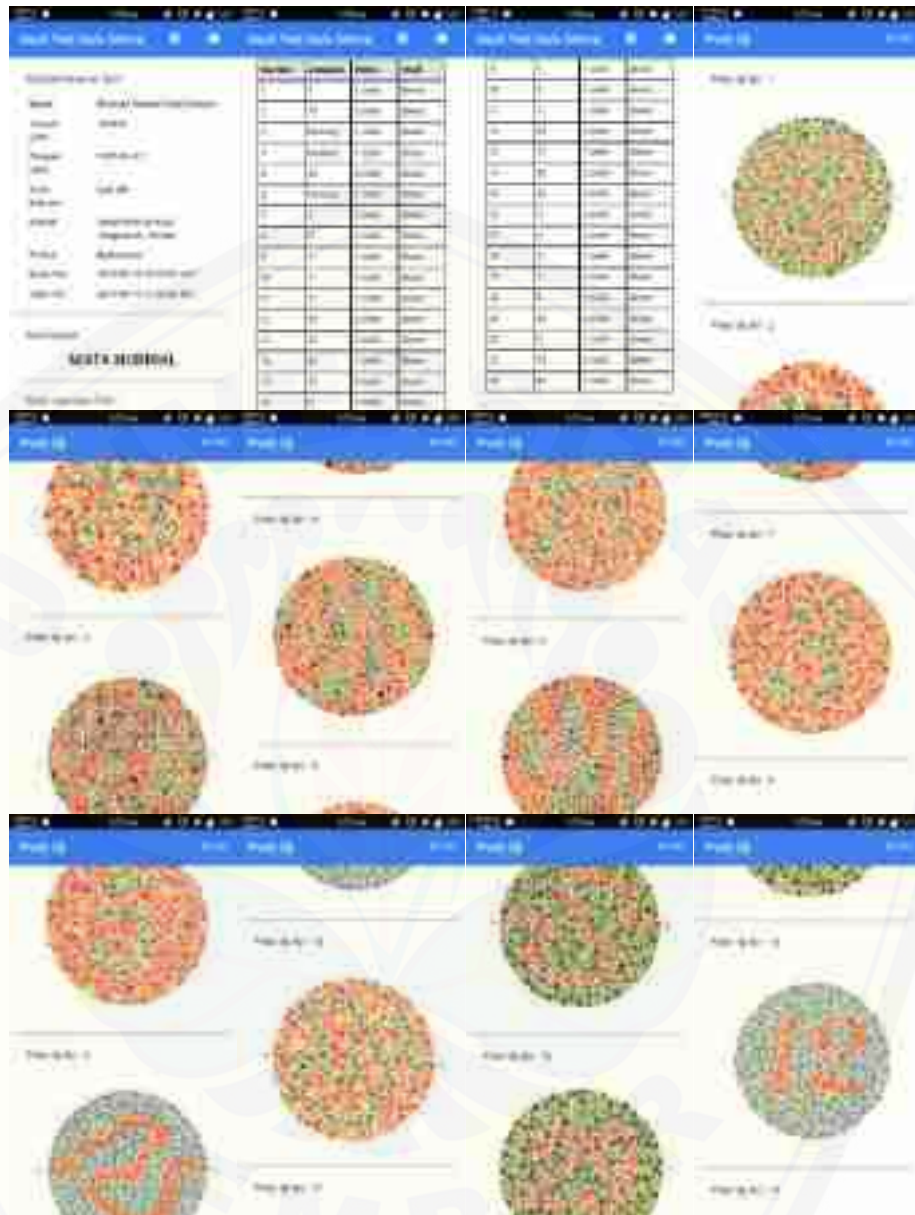


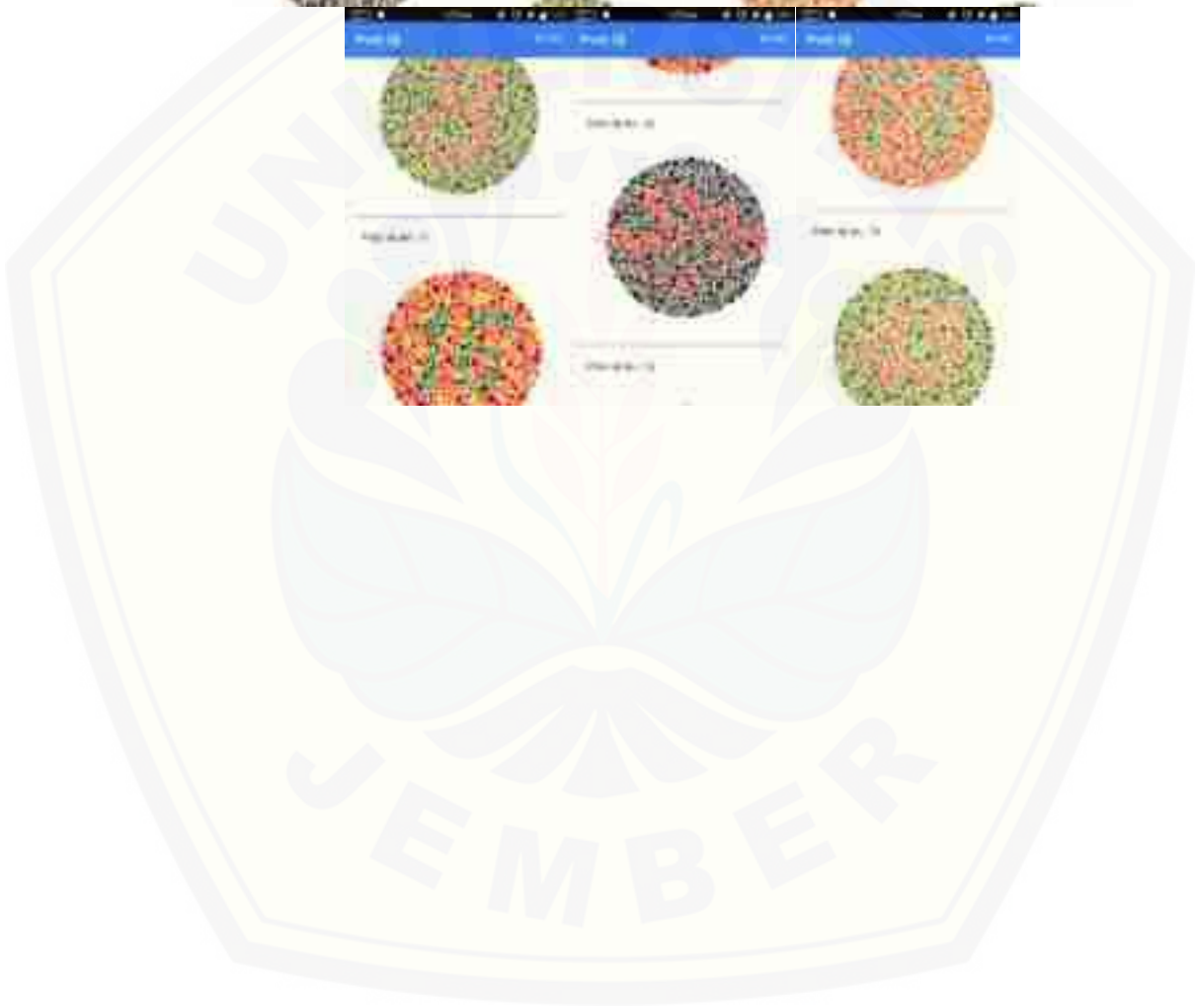
8. Ferika Sandra

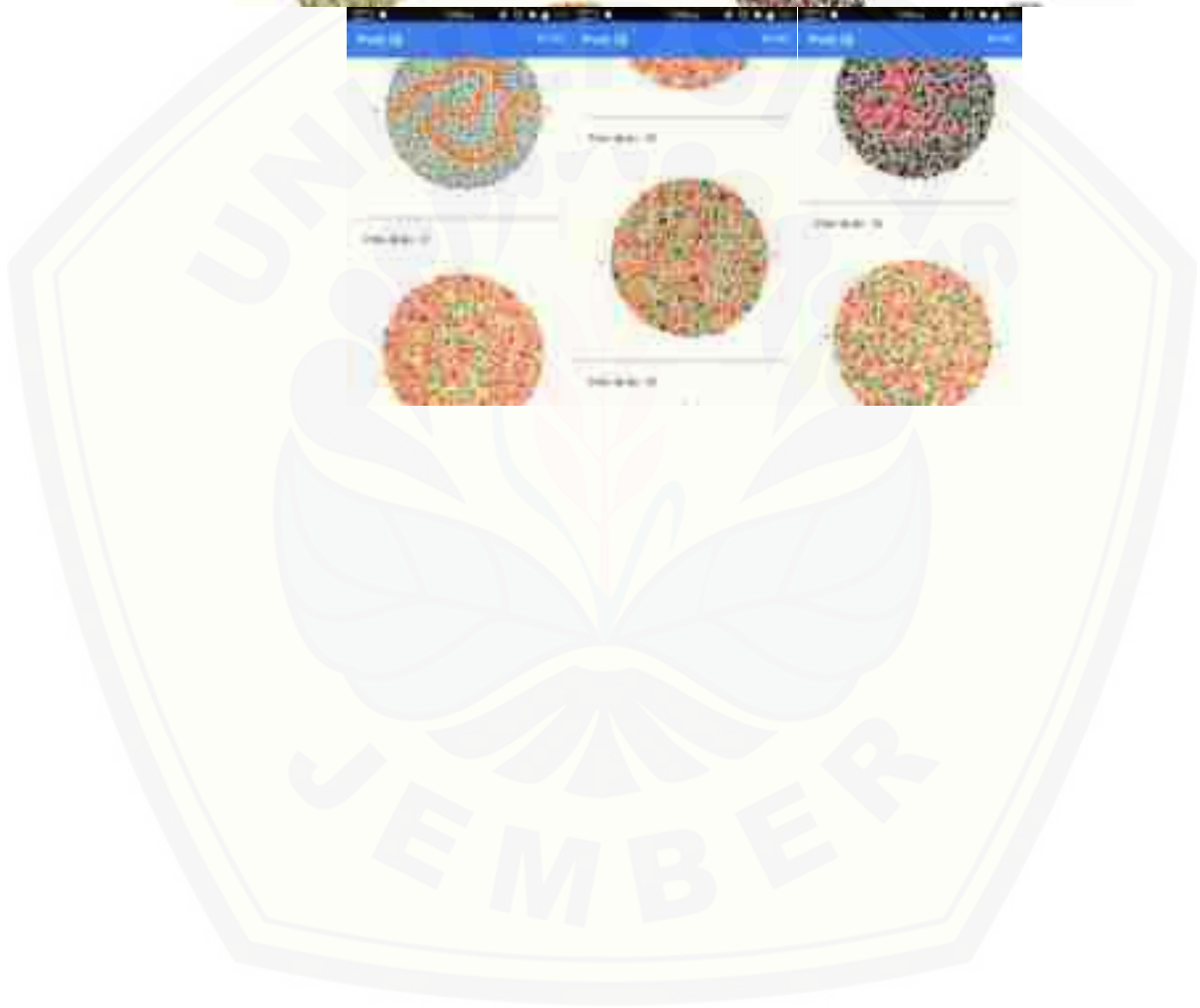


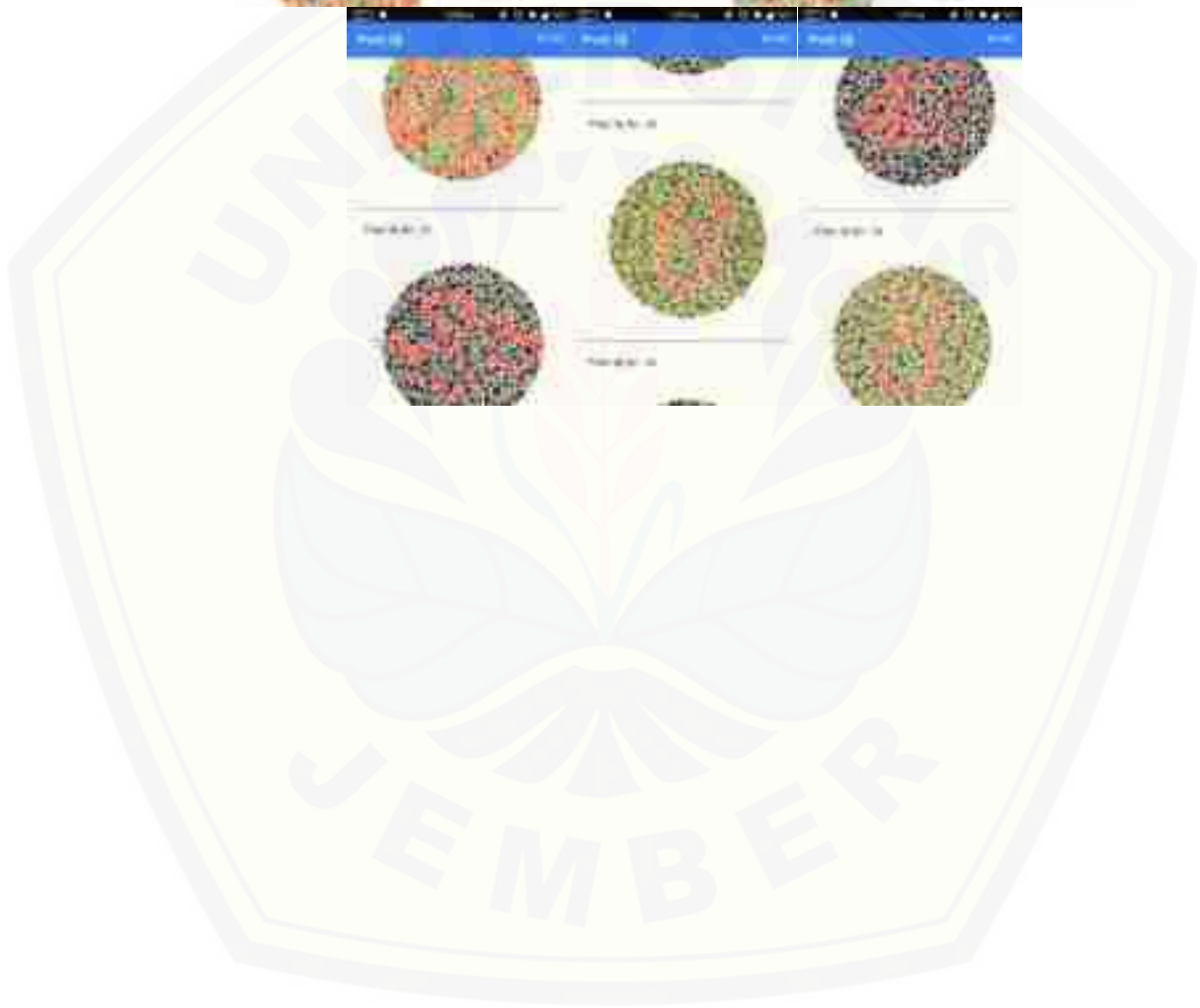
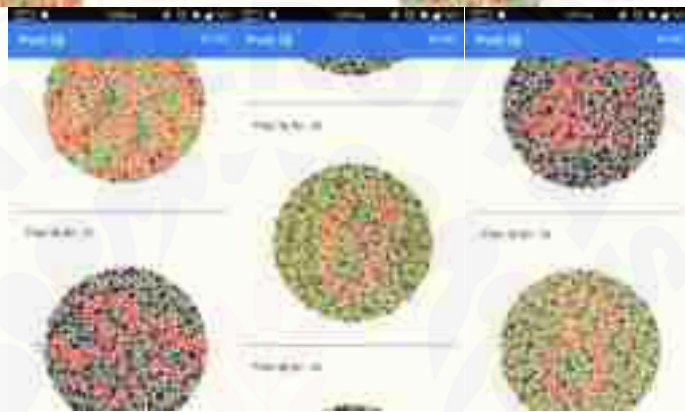


9. Rahmad Hidayat Hadi Subroto

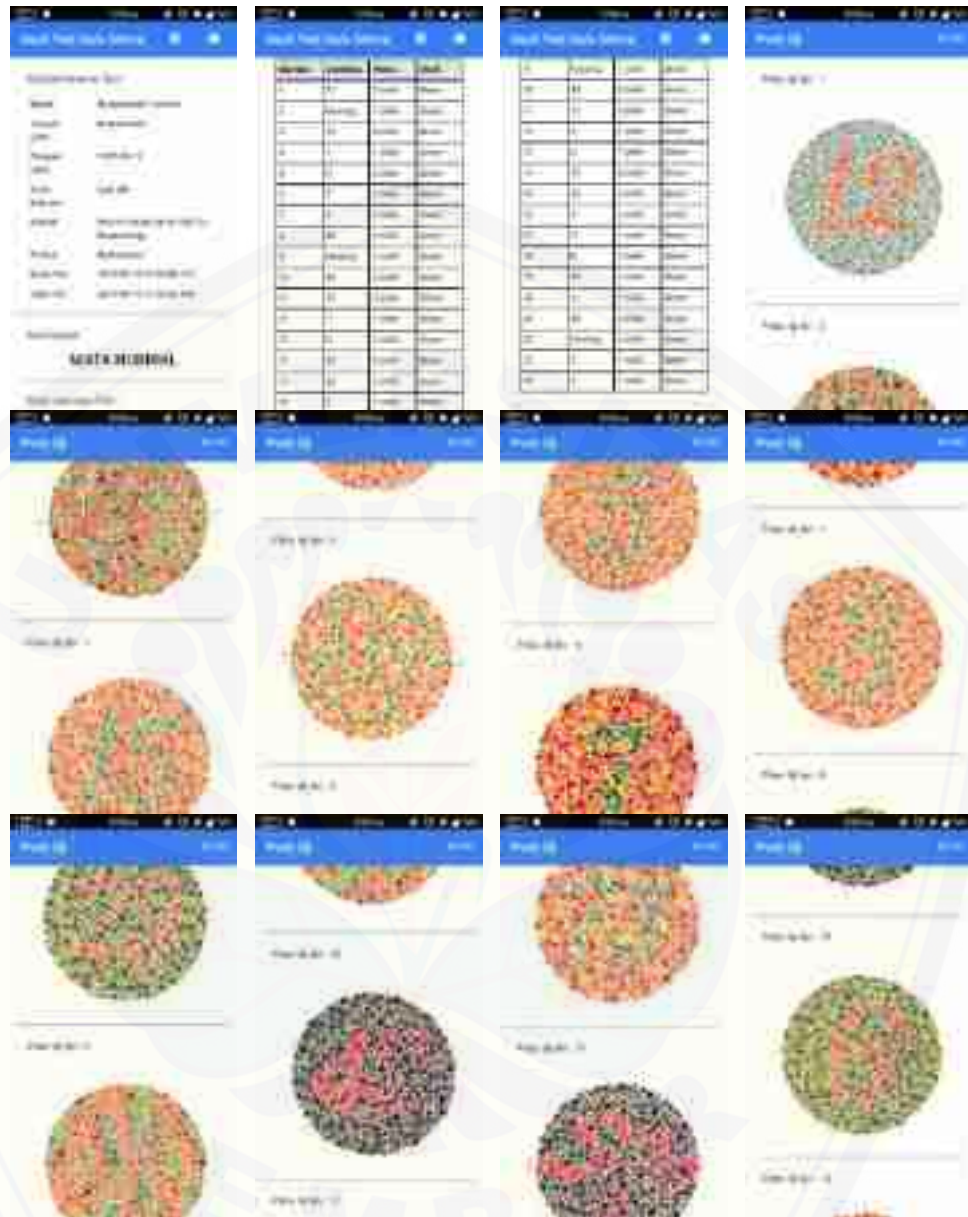


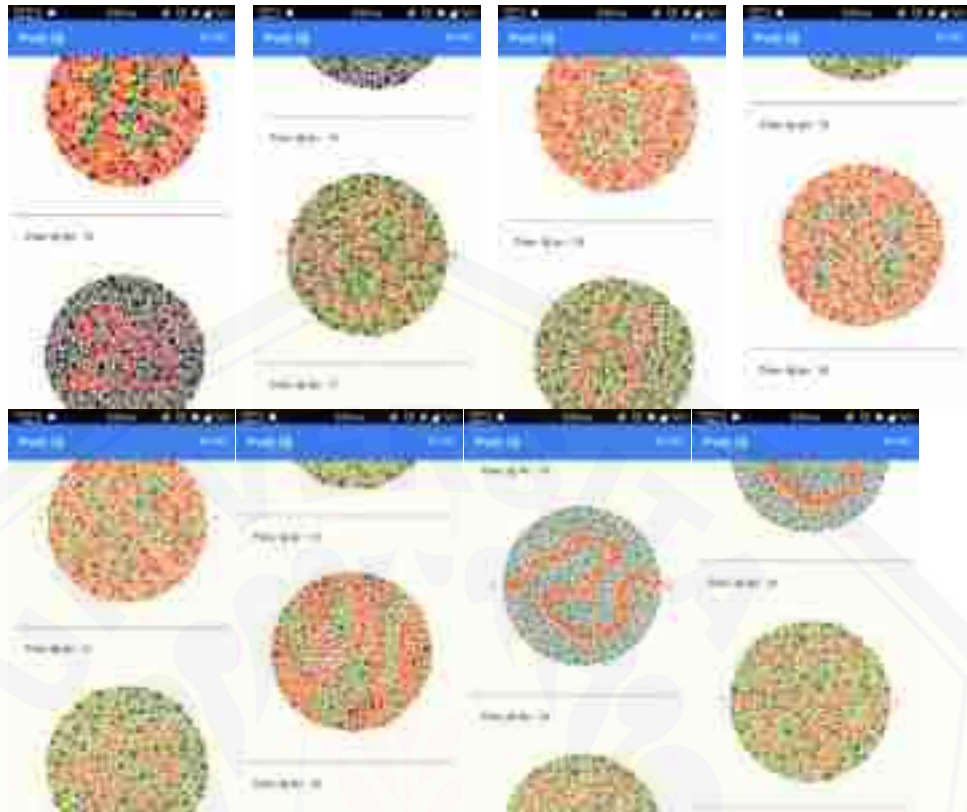




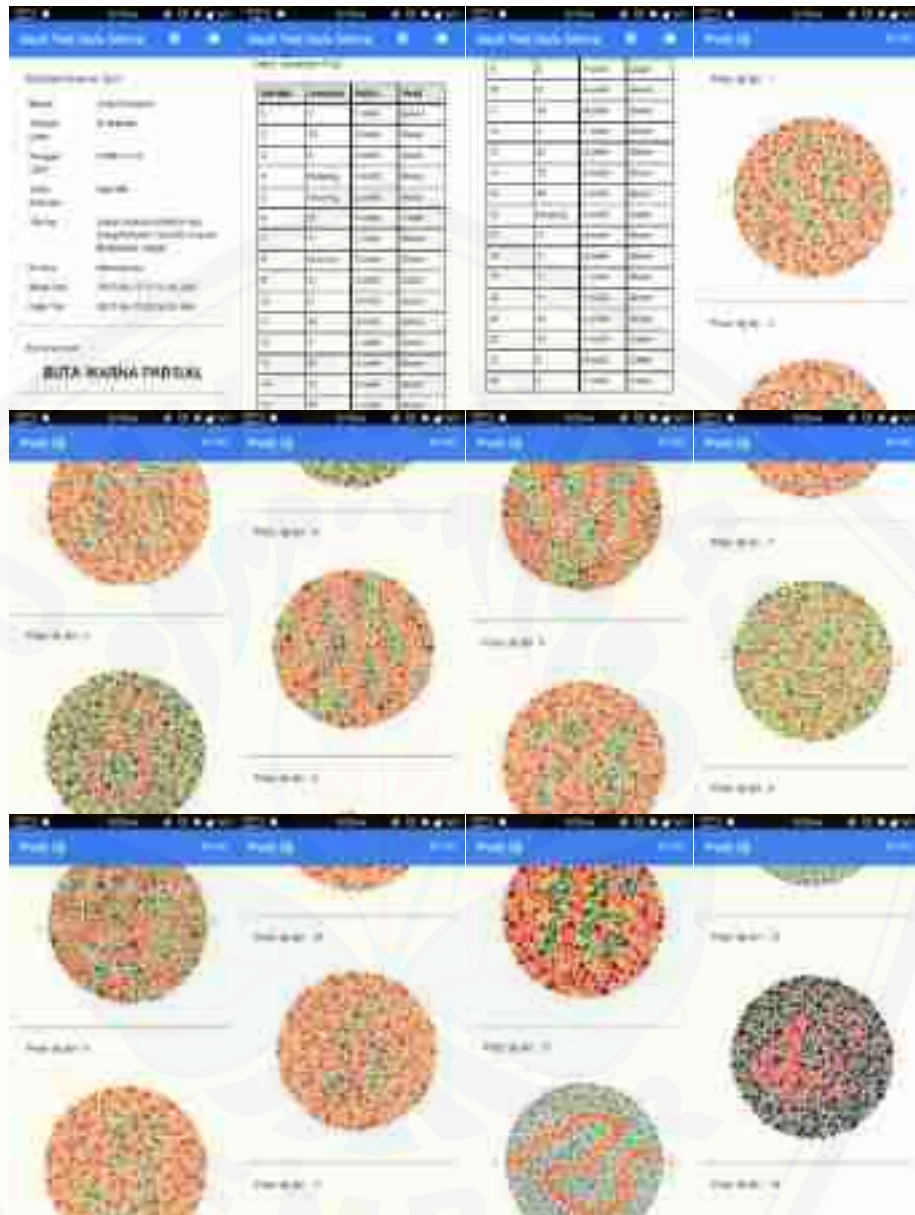


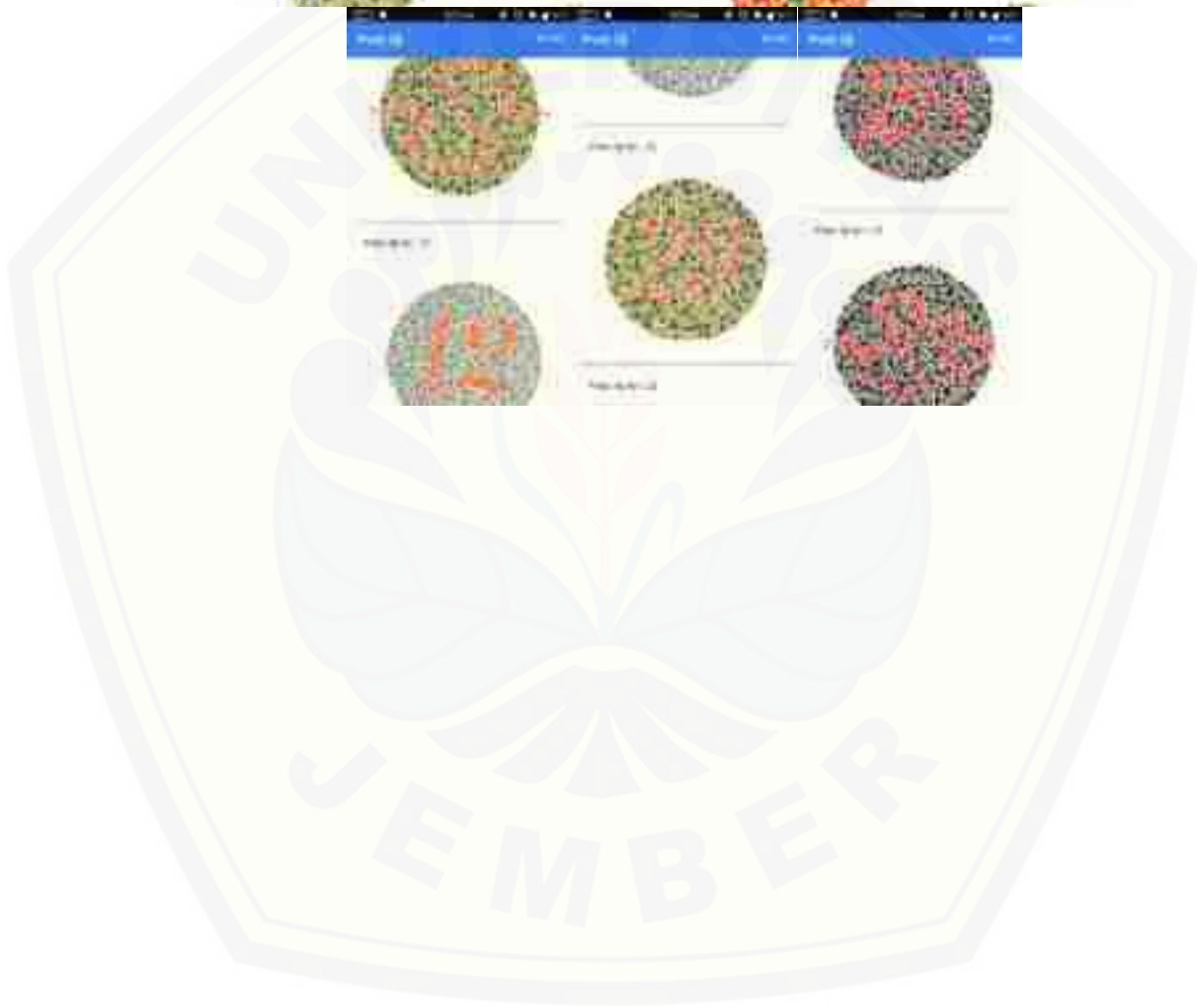
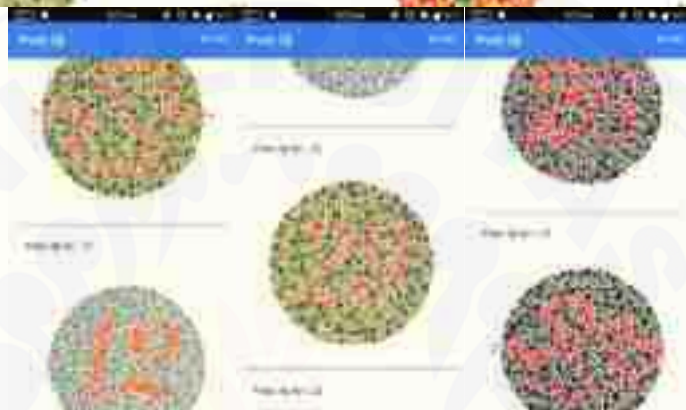
12. Muliyanayah Suciono



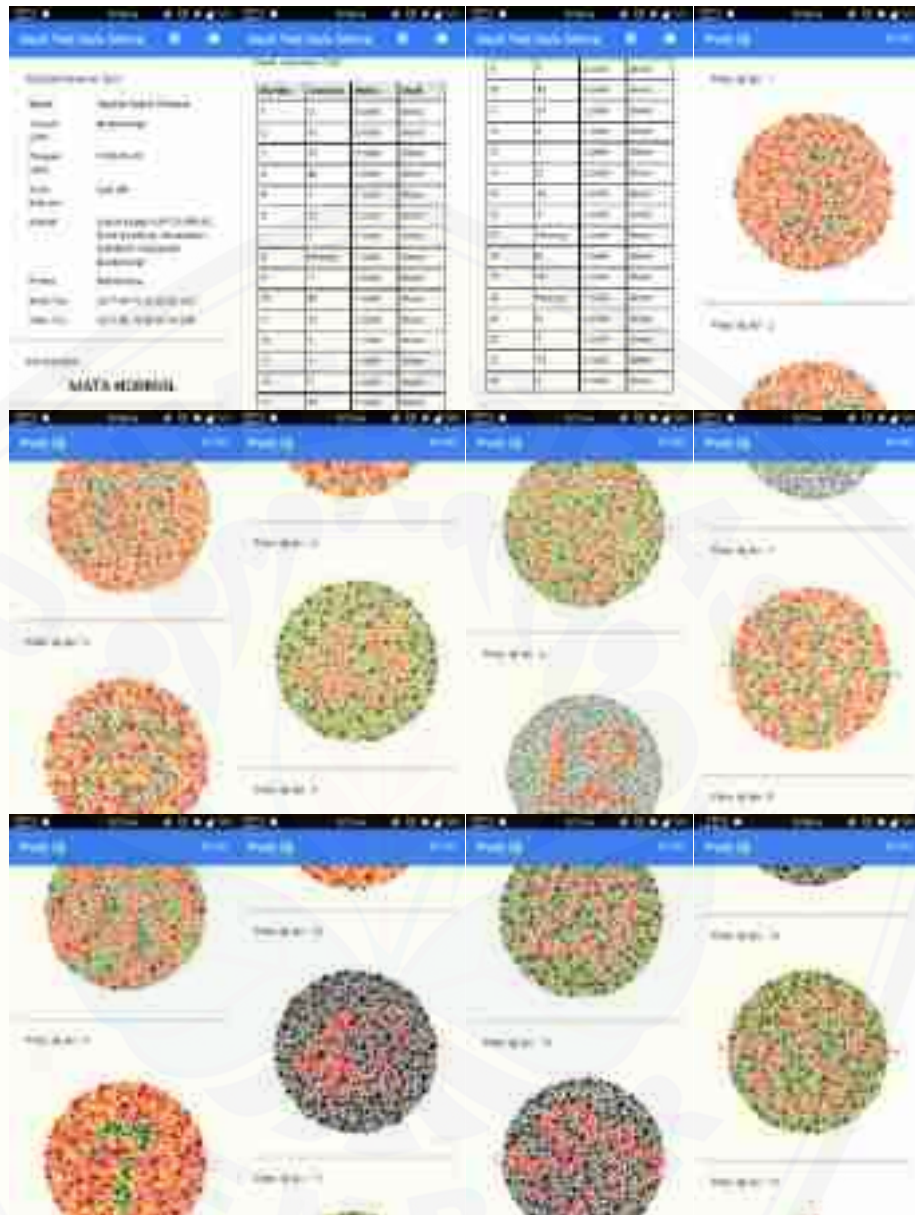


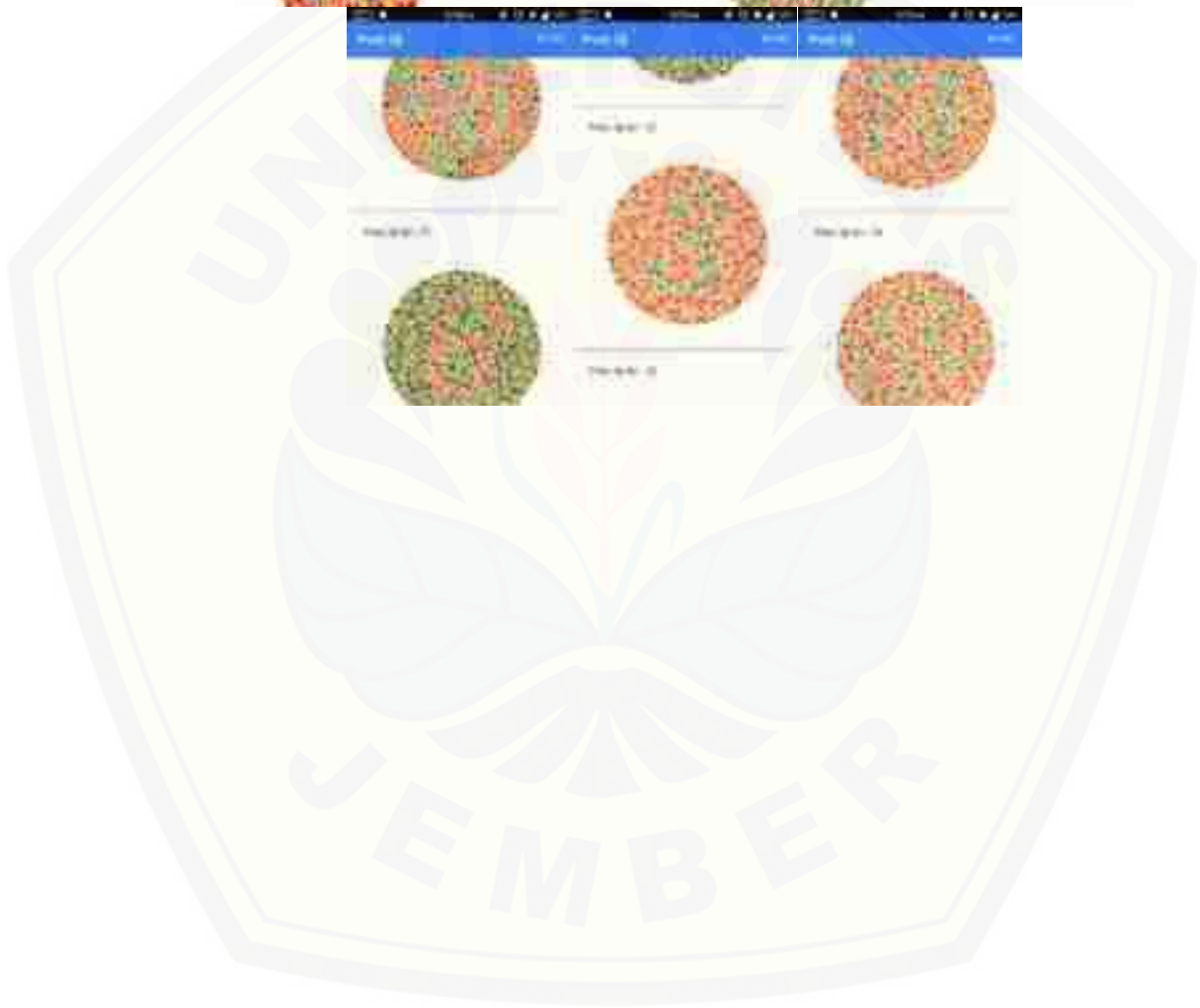
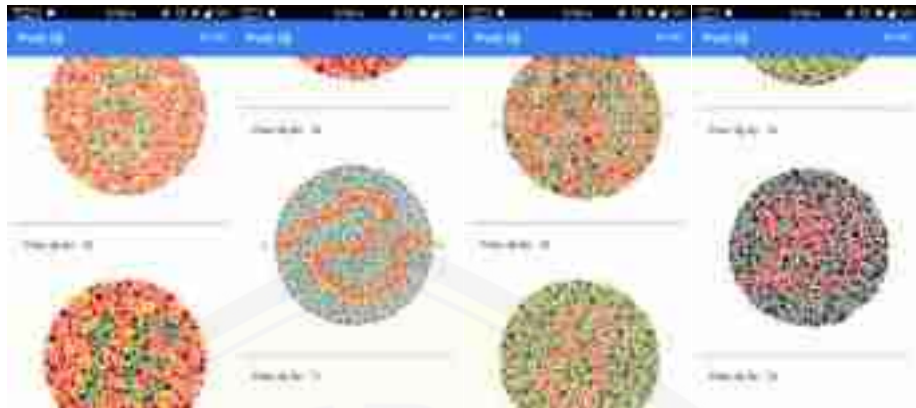
13. Andri Armanto



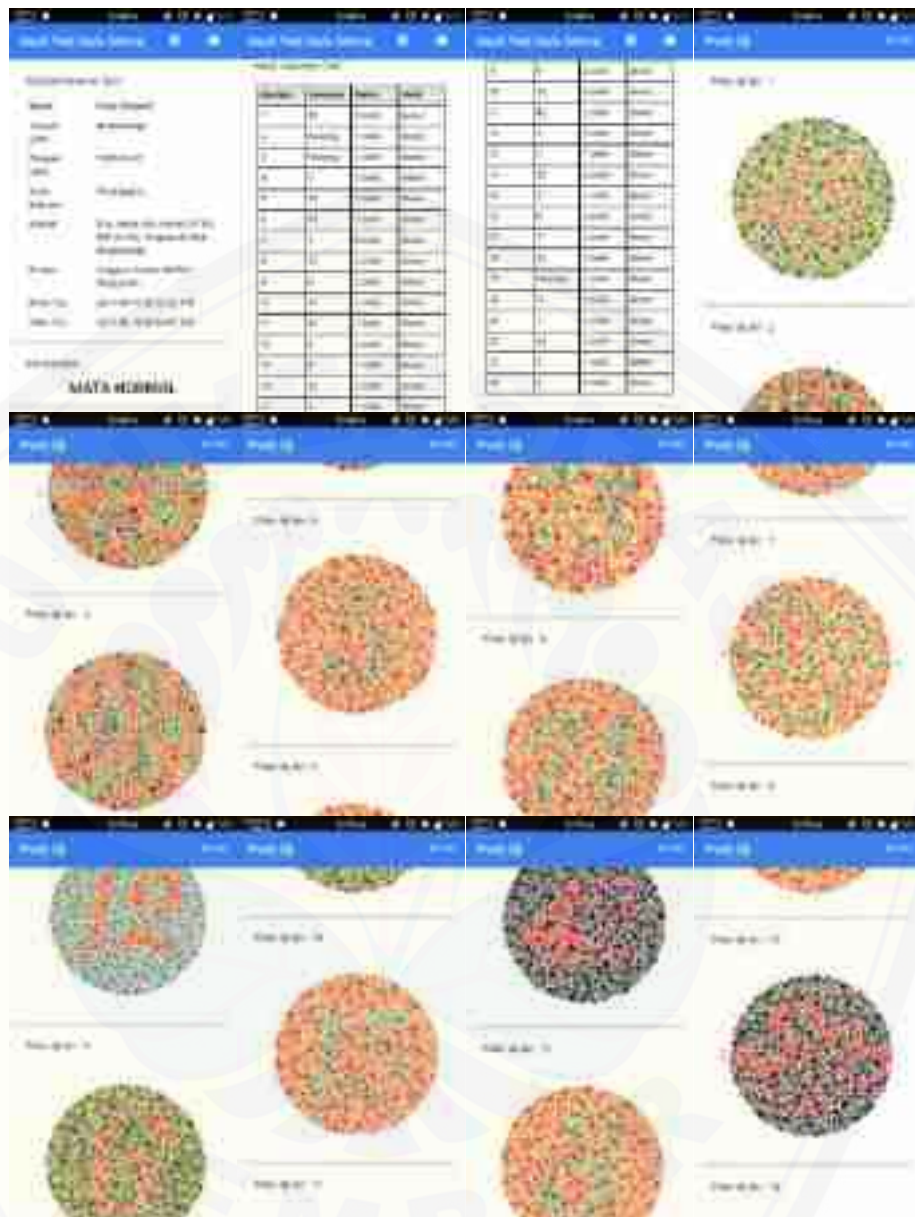


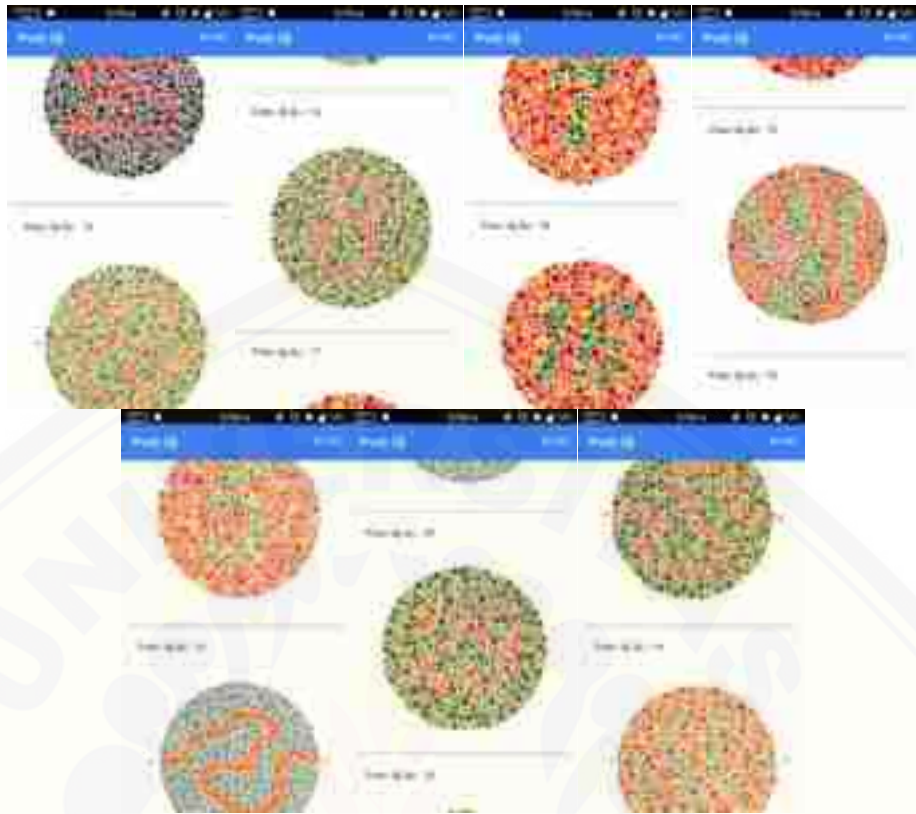
14. Septian Nanda Perdana



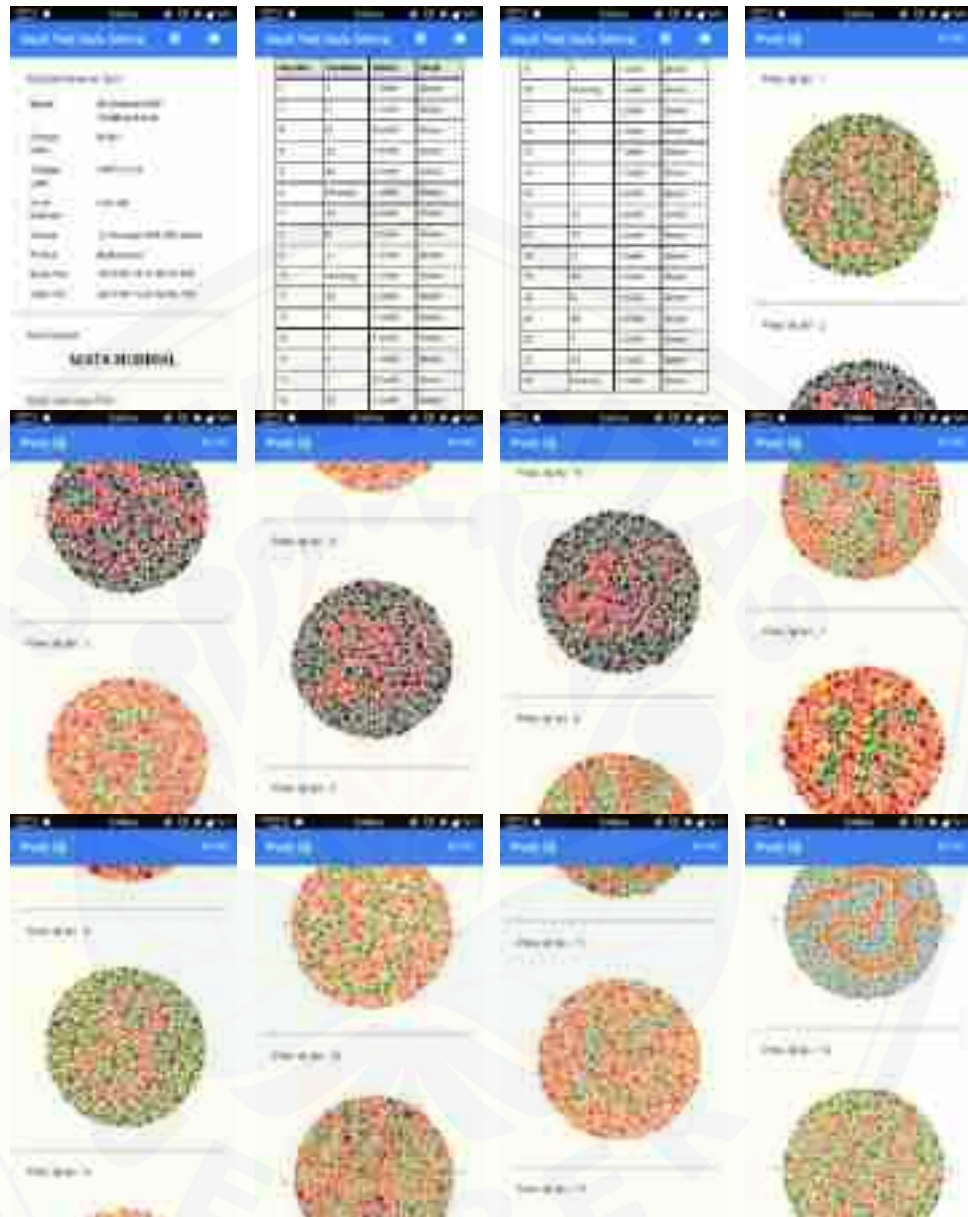


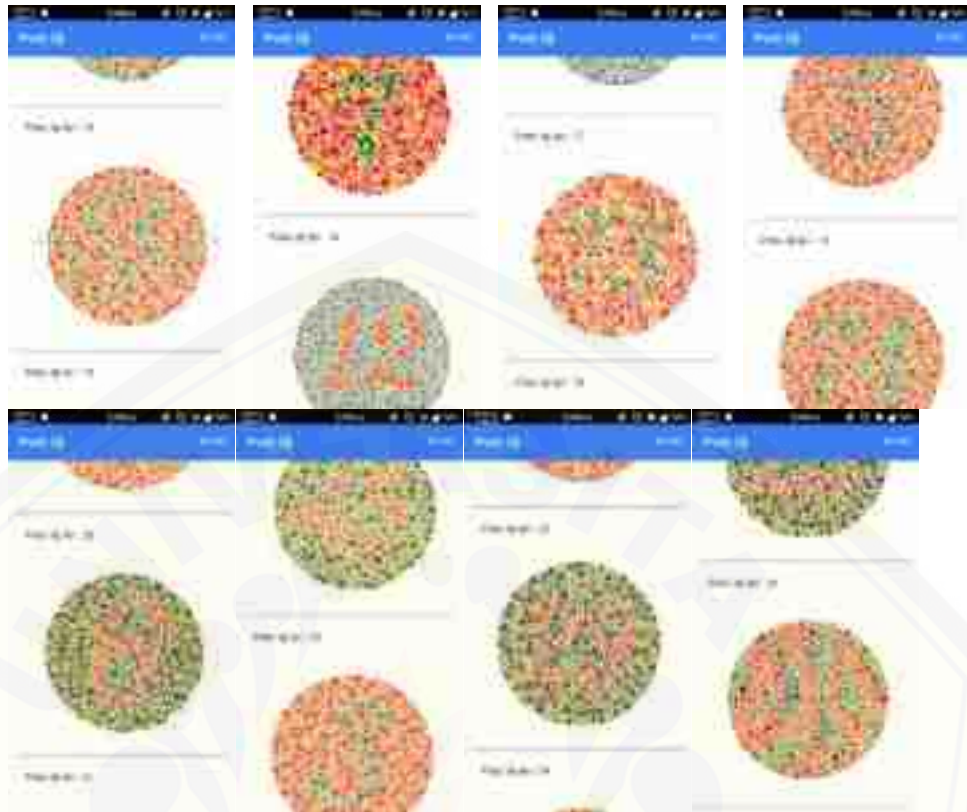
15. Pipin Setiyanti



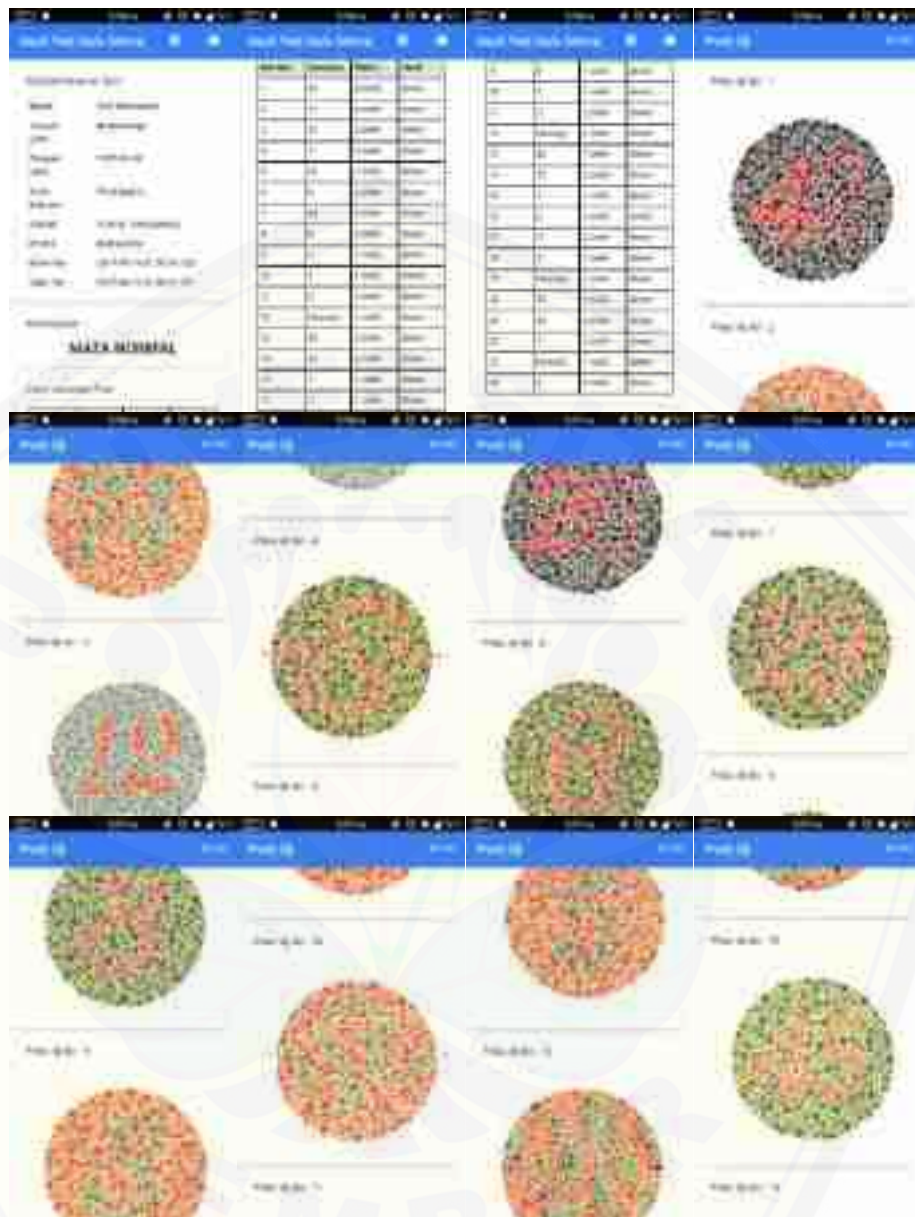


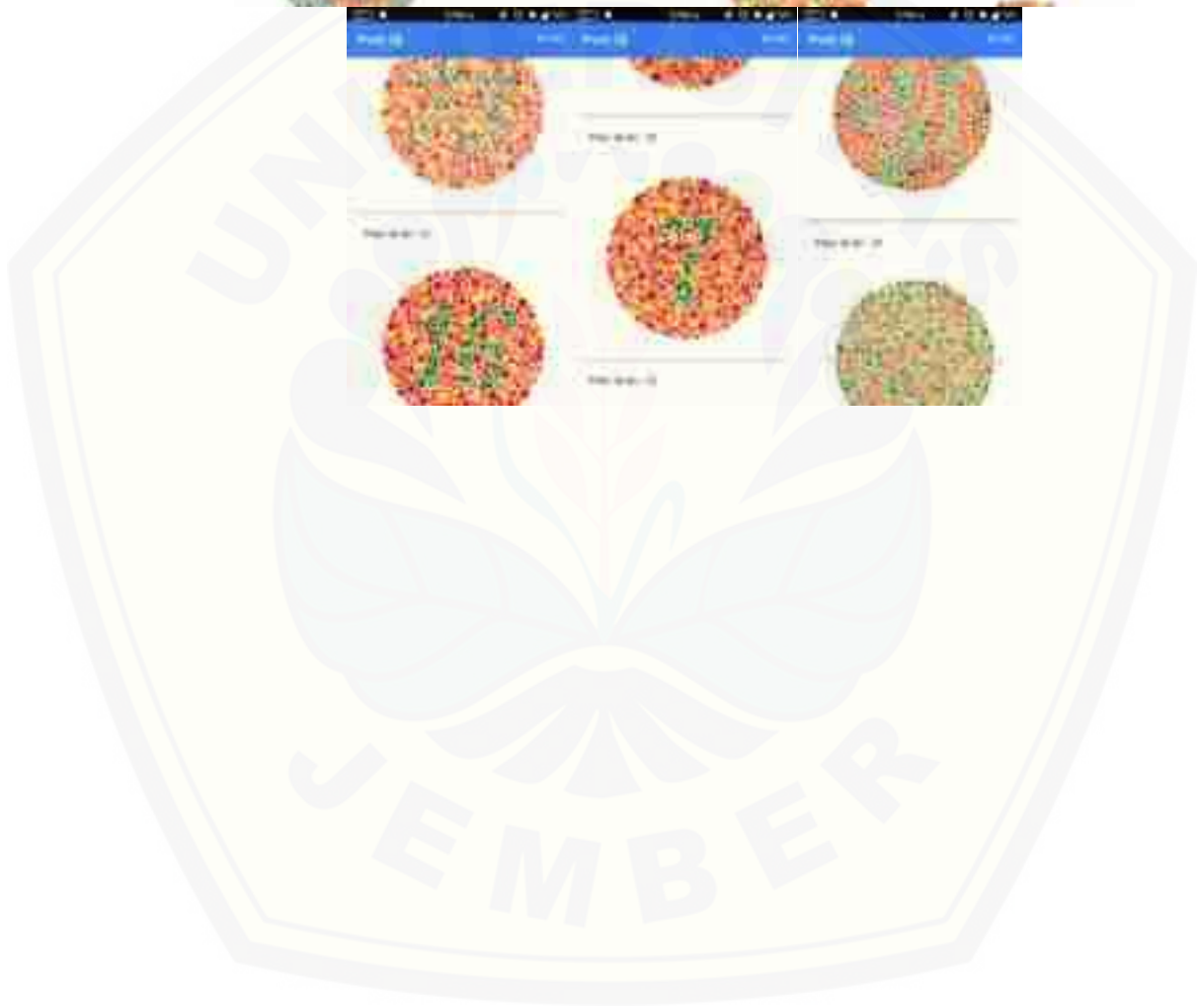
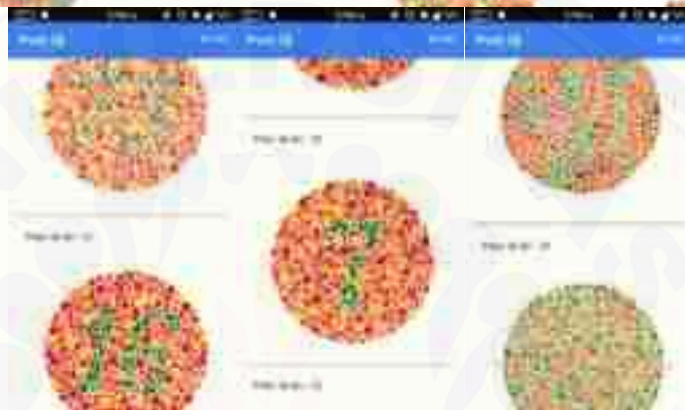
16. Mochamad Rafli Tuafikurrohman



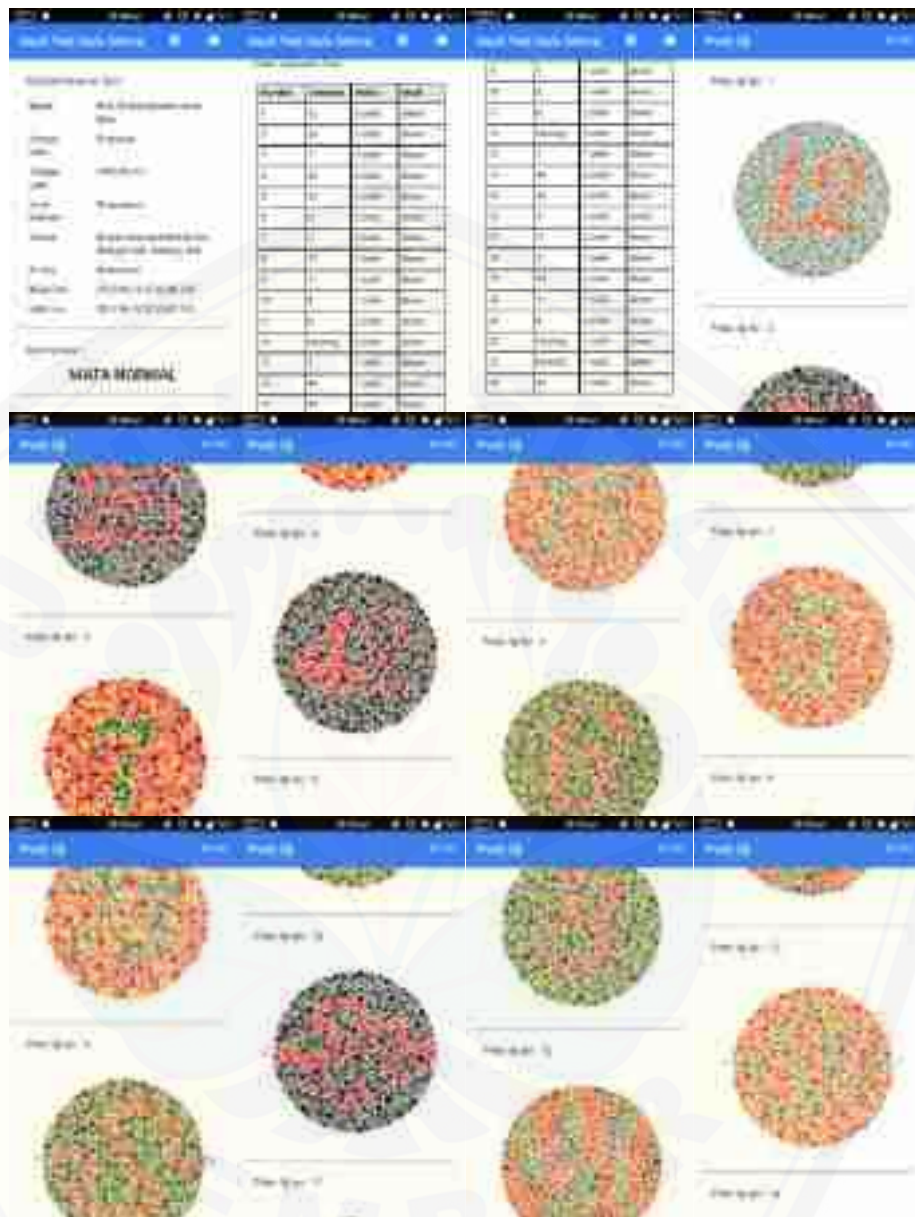


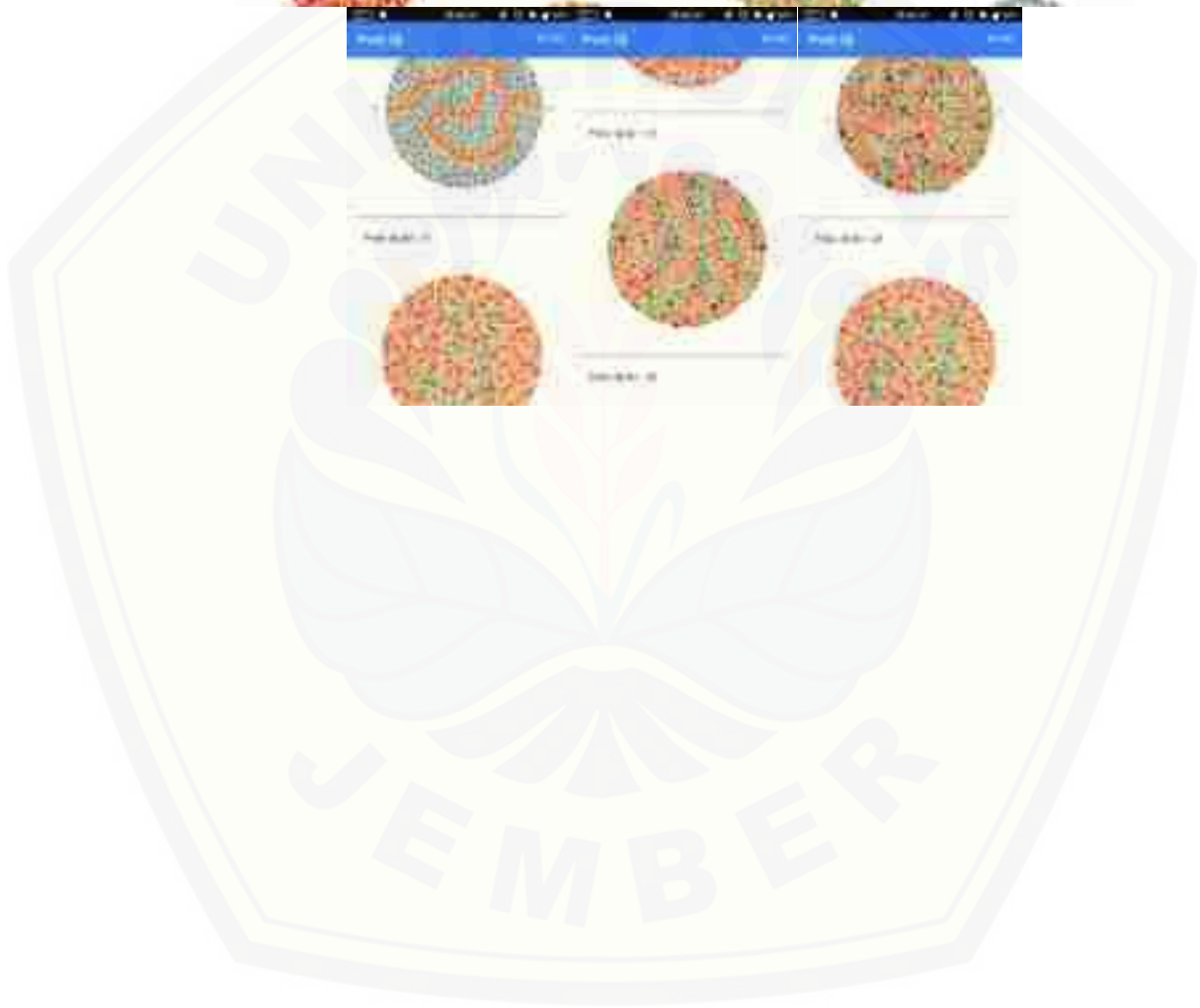
17. Ulfa Rahmawati



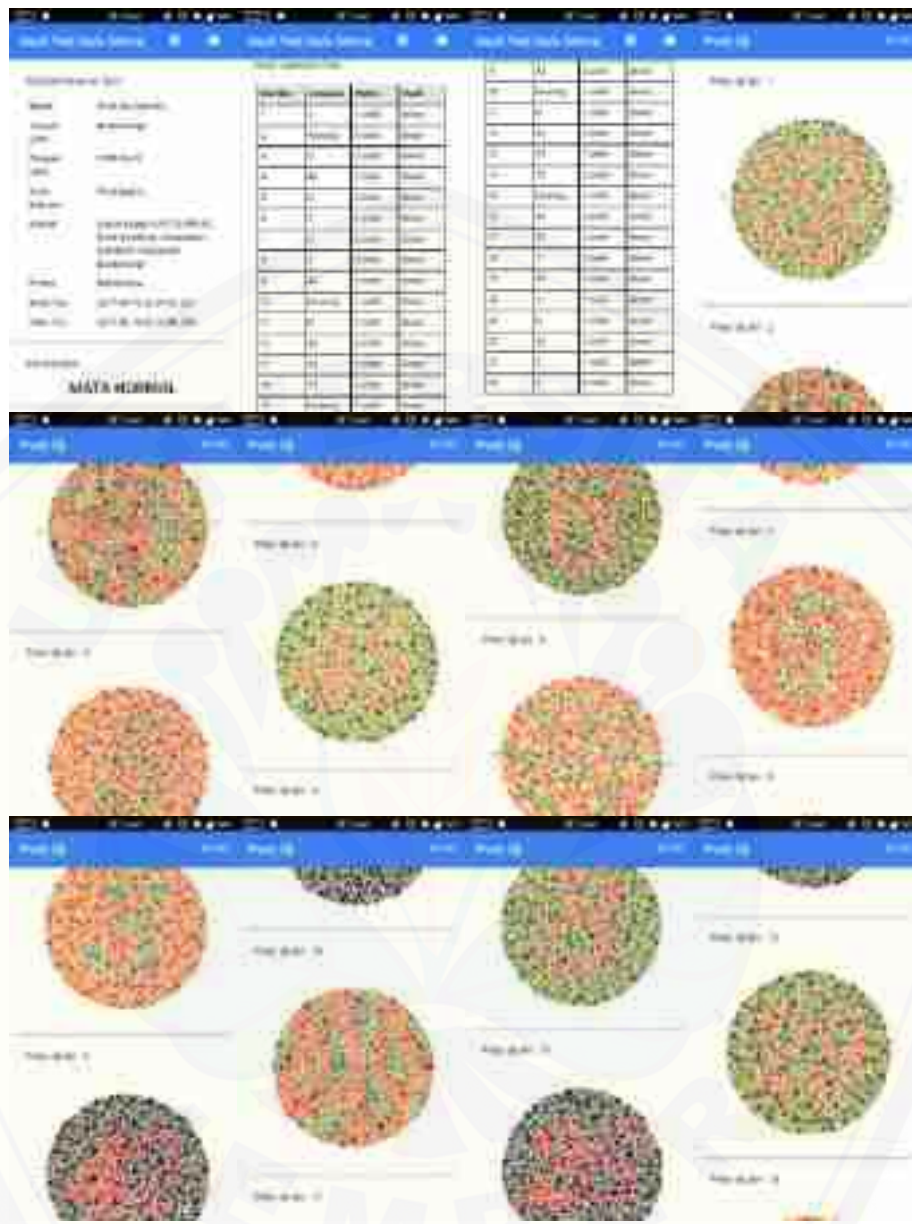


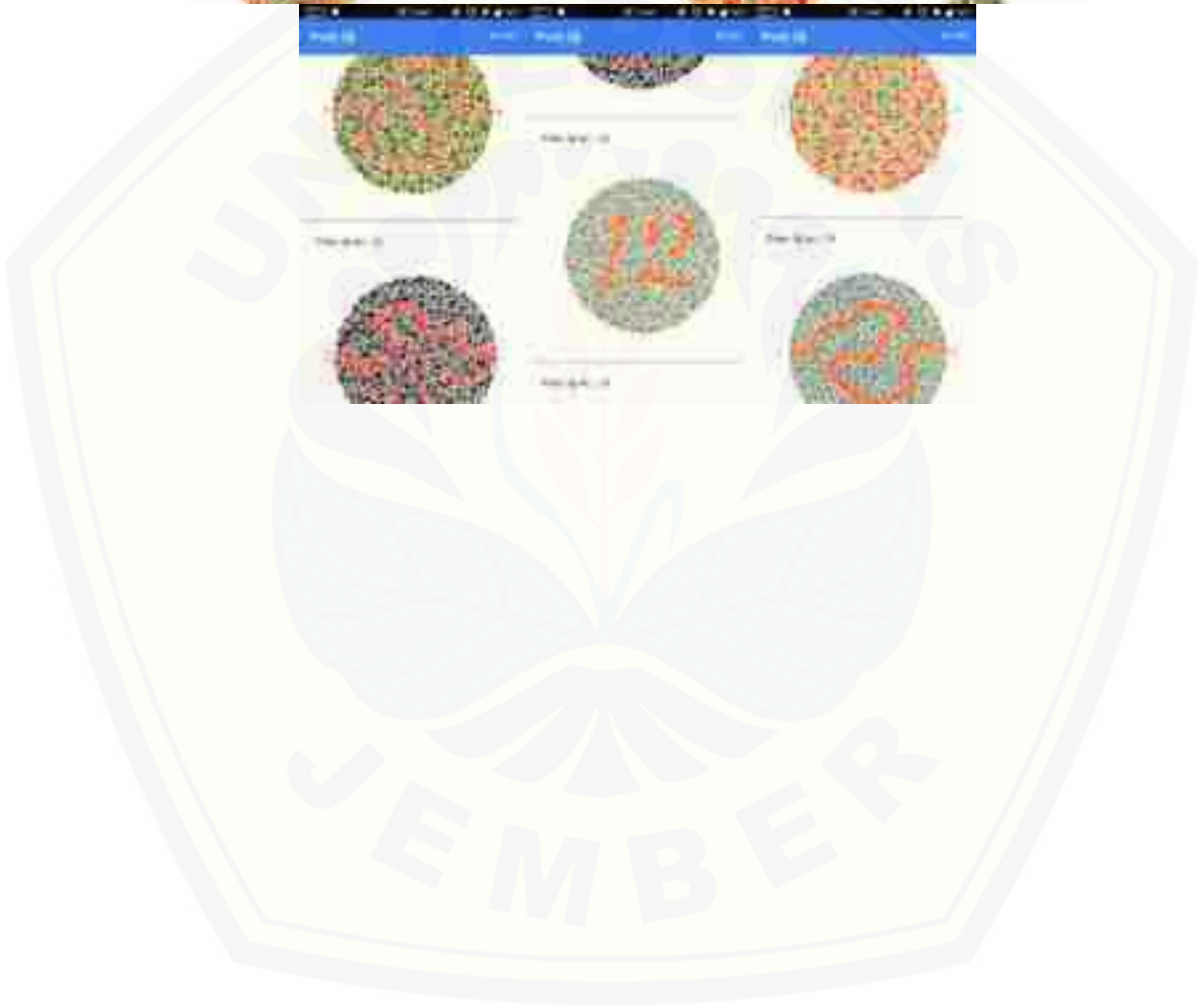
18. Putu Shintia Apriana Sucia Dewi



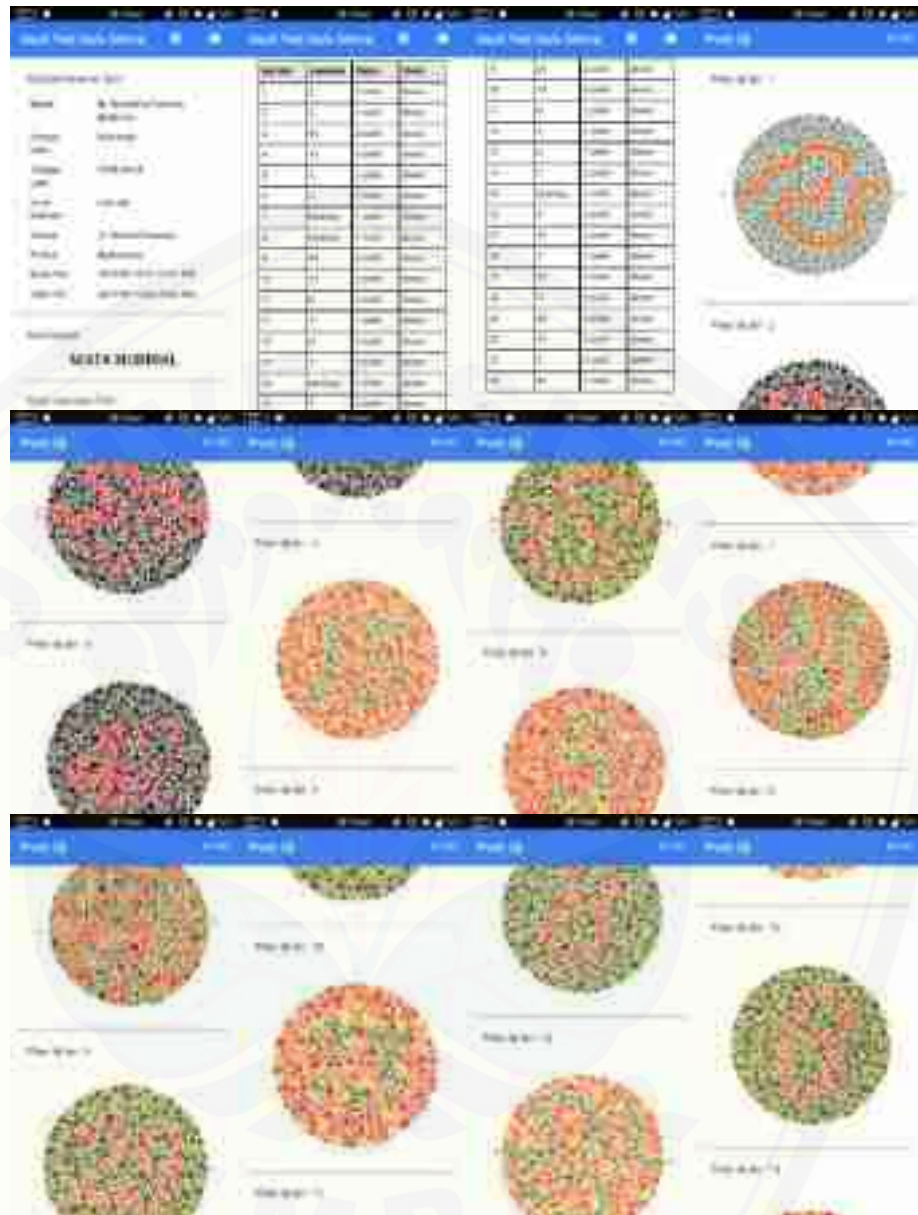


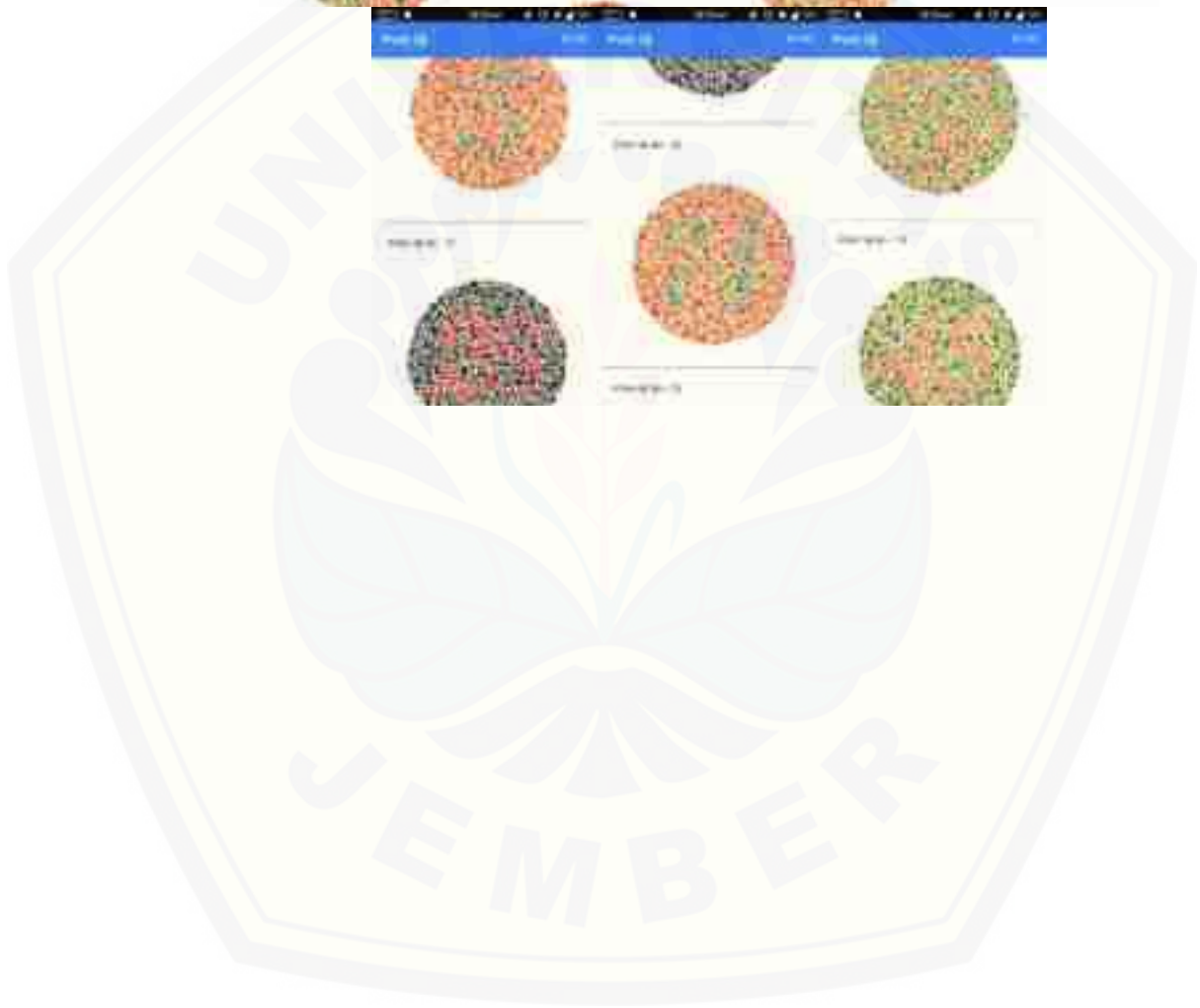
19. Sindi Dwi Agustin





20. M, Khuzaimy Rurroziq Basthoni





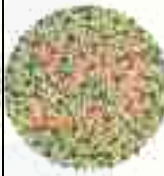
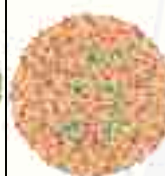


C. Pengumpulan Hasil Tes 20 Responden





Pada bagian ini memaparkan hasil jawaban yang didapat berdasarkan *screenshot* hasil tes aplikasi buta warna pada Lampiran B. Kemudian dari hasil jawaban responden tersebut diurutkan sesuai *plate* Ishihara. Hal ini dilakukan karena tampilan hasil tes yang diperoleh dari aplikasi bersifat acak. Agar memudahkan peneliti dalam menganalisis, maka diurutkanlah sesuai *plate* Ishihara. Jawaban benar diwakilkan dengan 1, sedangkan jawaban salah diwakilkan dengan 0. Selanjutnya, hasil pengumpulan jawaban tes buta warna yang sudah diurutkan tersebut akan digunakan dalam pengujian validitas dan reliabilitas.

C.1 Tes Pertama



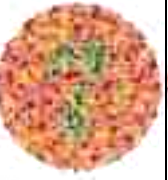

Hasil pengumpulan tes buta warna (bagian 1)

No.	Responden	Plate Ishihara dan Jawaban Responden			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	0	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hamman I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	0	0	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 2)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	0	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hamman I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	0	1	0	0
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 3)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	0	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 4)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	0
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	1	1	0
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1

Hasil pengumpulan tes buta warna (bagian 5)



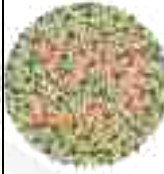
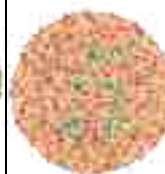
No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	0	0	1	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	0	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	0	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1

Hasil pengumpulan tes buta warna (bagian 6)





No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	1	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1

C.2 Tes Kedua



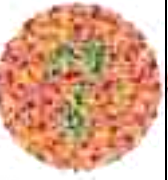

Hasil pengumpulan tes buta warna (bagian 1)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	0	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	0	0	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 2)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	0	0
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	0	1	0	0
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 3)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	1	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1





Hasil pengumpulan tes buta warna (bagian 4)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	0
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	1	1	0
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1

Hasil pengumpulan tes buta warna (bagian 5)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	0	0	0	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	0	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1

Hasil pengumpulan tes buta warna (bagian 6)

No.	Responden	<i>Plate Ishihara dan Jawaban Responden</i>			
					
1.	Abrian Eko P.	1	1	1	1
2.	Bagas Di Saputra	1	1	1	1
3.	Agusty Maulana B.	1	1	1	1
4.	Galih Chandra W.	1	1	1	1
5.	Kharisma Ahmad P.	1	1	1	1
6.	Emha Dinar S.	1	1	1	1
7.	Novado Caesar	1	1	1	1
8.	Ferika Sandra	1	1	1	1
9.	Rahmad Hidayat H.	1	1	1	1
10.	Hammam I.	1	1	1	1
11.	Hariawan	1	1	1	1
12.	Muliyansyah S.	1	1	1	1
13.	Andri Armanto	1	1	1	1
14.	Septian Nanda P.	1	1	1	1
15.	Pipin Setiyanti	1	1	1	1
16.	Mochamad Rafli T.	1	1	1	1
17.	Ulfa Rahmawati	1	1	1	1
18.	Putu Shintia A. S. D.	1	1	1	1
19.	Sindi Dwi Agustin	1	1	1	1
20.	M. Khuzaimy R. B.	1	1	1	1