

**PENGEMBANGAN APLIKASI *MOBILE* UII *SKIN ANALYZER*
DENGAN FLUTTER UNTUK MEMBANTU ANALISIS
PERMASALAHAN KULIT WAJAH**



Disusun Oleh:

N a m a : Nurastuti Wijareni
NIM : 19523064

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2023**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI *MOBILE* UII *SKIN ANALYZER*
DENGAN FLUTTER UNTUK MEMBANTU ANALISIS
PERMASALAHAN KULIT WAJAH**

TUGAS AKHIR

Yogyakarta, 5 Mei 2023

Pembimbing 1,

Pembimbing 2,

(Izzati Muhimmah, S.T., M.Sc., Ph.D.)

(Septia Rani, S.T., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN APLIKASI *MOBILE UII SKIN ANALYZER* DENGAN FLUTTER UNTUK MEMBANTU ANALISIS PERMASALAHAN KULIT WAJAH

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 16 Mei 2023

Tim Penguji

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Anggota 1

Chanifah Indah Ratnasari, S.Kom., M.Kom.

Anggota 2

Zainudin Zukhri, S.T., MIT.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nurastuti Wijareni

NIM : 19523064

Tugas akhir dengan judul:

PENGEMBANGAN APLIKASI *MOBILE* UII *SKIN ANALYZER* DENGAN FLUTTER UNTUK MEMBANTU ANALISIS PERMASALAHAN KULIT WAJAH

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Mei 2023



(Nurastuti Wijareni)

HALAMAN PERSEMBAHAN



Alhamdulillah rabbil'alamin segala puji dan syukur kepada Allah Subhanahu wa Ta'ala atas segala limpahan rahmat, taufiq, hidayah, inayah, dan kasih sayang-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.

Selawat serta salam selalu tercurahkan kepada Rasulullah Shallallahu 'Alaihi Wasallam. Semoga kebaikan selalu mengiringi Beliau, keluarga, sahabat, serta para pengikutnya hingga hari kiamat kelak.

Bapak Budi Hartono dan Ibu Isnurul Kusnaeni

Terima kasih yang sebesar-besarnya kepada kedua orang tua penulis yang senantiasa selalu memberikan dukungan baik berupa nasihat, motivasi, doa, material dan dalam bentuk lainnya sehingga penulis dapat menyelesaikan tugas akhir dan menjalani masa studi dengan penuh tanggung jawab.

Edwin Aditya Irawan, S.Psi. dan Muhammad Satrio Wicaksono

Terima kasih kepada Mas Edwin dan Adik Satrio selaku saudara sekaligus teman atas segala dukungan, pengingat, nasihat, canda-tawa agar menjalankan kuliah dan mengerjakan tugas akhir dengan sungguh-sungguh.

Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D. dan Ibu Septia Rani, S.T., M.Cs.

Terima kasih kepada Bu Izzati dan Bu Rani selaku dosen pembimbing tugas akhir penulis atas segala bimbingan, arahan, nasihat, doa, dan ilmu yang telah diberikan kepada penulis dalam pengembangan aplikasi mobile bernama UII Skin Analyzer yang bertujuan untuk membantu masyarakat untuk menganalisis permasalahan kulit wajahnya.

HALAMAN MOTO

“Dan sesungguhnya telah Kami muliakan anak-anak Adam, Kami angkut mereka di daratan dan di lautan, Kami beri mereka rezeki dari yang baik-baik dan Kami lebihkan mereka dengan kelebihan yang sempurna atas kebanyakan makhluk yang telah Kami ciptakan”

(Q.S. Al-Isra' : 70)

“Tunjukilah kami jalan yang lurus, (yaitu) jalan orang-orang yang telah Engkau beri nikmat kepadanya, bukan (jalan) mereka yang dimurkai, dan bukan (pula jalan) mereka yang sesat”

(Q.S. Al-Fatihah : 6-7)

“..Boleh jadi kamu tidak menyenangi sesuatu, padahal itu baik bagimu, dan boleh jadi kamu menyukai sesuatu, padahal itu tidak baik bagimu. Allah Maha Mengetahui, sedang kamu tidak mengetahui”

(Q.S. Al-Baqarah : 216)

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(Q.S. Al-Baqarah : 286)

“Barang siapa yang menempuh suatu jalan untuk menuntut ilmu, Allah akan memudahkan baginya jalan ke surga”

(HR. Muslim)

“Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia”

(HR. Ahmad, ath-Thabrani, ad-Daruqutni)

“Mencari ilmu itu hukumnya wajib bagi musliman dan muslimat”

(HR. Ibnu Abdil Bari)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah Rabbil'alamin, segala puji syukur saya panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penyusunan laporan Tugas Akhir (TA) dapat diselesaikan dengan baik dan tepat waktu. Selawat serta salam tak lupa saya haturkan kepada Nabi Muhammad SAW yang telah membawa kita dari zaman kegelapan menuju zaman yang terang seperti saat ini.

Laporan ini disusun untuk memenuhi persyaratan akademik dalam menyelesaikan studi jenjang Strata 1 (S1). Selanjutnya, laporan Tugas Akhir yang berjudul “Pengembangan Aplikasi *Mobile* “*UII Skin Analyzer*” dengan Flutter Untuk Membantu Analisis Permasalahan Kulit Wajah” dan aplikasi *UII Skin Analyzer* yang fungsional pada perangkat Android dapat diselesaikan dengan baik atas bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah melimpahkan rahmat, taufiq, hidayah, dan inayah sehingga penulis dapat menyelesaikan laporan tugas akhir ini
2. Bapak, ibu, dan keluarga yang senantiasa memberikan doa, dukungan, semangat, dan kasih sayang.
3. Bapak Prof. Fathul Wahid, S.T., M.S.c., Ph.D. selaku Rektor Universitas Islam Indonesia.
4. Bapak Prof. Dr. Hari Purnomo., M.T., IPU., ASEAN, Eng., selaku dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Program Sarjana Universitas Islam Indonesia.
7. Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D. dan Ibu Septia Rani, S.T., M.Cs. selaku dosen pembimbing tugas akhir yang selalu memberikan bimbingan, arahan dan doa dalam menyelesaikan tugas akhir dengan sangat baik hingga selesai.
8. Ibu Sri Mulyati, S.Kom., M.Kom. selaku dosen pembimbing akademik yang senantiasa membantu dan memberikan saran yang dapat memperlancar proses perkuliahan.
9. Segenap dosen dan tendik di Jurusan Informatika UII yang telah memberikan banyak pengetahuan dan mengajarkan banyak hal.

10. Mas July Arifianto yang telah membuat model untuk deteksi dan membantu penulis dalam mengintegrasikan server dengan perangkat *mobile* yang penulis buat.
11. Pemilik NIM 19523170 yang telah membantu, memberikan kritik, saran, dan dukungan kepada penulis dalam rangka menyelesaikan tugas akhir ini.
12. Teman-teman jurusan Informatika dan jurusan lainnya angkatan 2019 di lingkungan Universitas Islam Indonesia yang telah bersedia menjadi tempat berbagi suka dan duka selama masa perkuliahan.
13. Seluruh pihak yang tidak dapat disebutkan satu per satu, terima kasih atas segala bantuan yang telah diberikan dalam bentuk apapun.

Semoga penelitian ini dapat bermanfaat dimasa sekarang dan akan datang, serta memberikan pahala jariyah bagi seluruh pihak yang telah membantu dan berkontribusi dalam pelaksanaan penelitian ini. *Aamiin Ya Rabbal'aalamiin.*

Yogyakarta, 5 Mei 2023



(Nurastuti Wijareni)

SARI

Kulit adalah organ tubuh paling kompleks, terluar, terbesar, dan berfungsi sebagai pelindung organ-organ di bawahnya. Kulit wajah memiliki kelenjar minyak yang lebih banyak, sehingga lebih mudah mengalami berbagai permasalahan dibandingkan dengan kulit pada bagian tubuh lainnya. Oleh karena itu, perawatan kulit wajah yang tepat sangat penting untuk menjaga kesehatan dan kecantikan kulit wajah. Pada saat ini perawatan kulit wajah menggunakan *skincare* telah menjadi hal yang familier di tengah masyarakat tetapi perlu disadari bahwa tidak hanya memberikan dampak positif saja, ternyata penggunaan *skincare* yang tidak tepat justru semakin memperparah kondisi kulit wajah seseorang. Pada umumnya untuk mengetahui kondisi wajah dan *skincare* yang tepat adalah dengan melakukan konsultasi kepada dokter kulit. Akan tetapi pada faktanya tidak semua orang melakukan hal tersebut.

Dalam rangka membantu mengatasi permasalahan tersebut, pada penelitian sebelumnya telah dibuat sebuah sistem analisis permasalahan kulit wajah yang menerapkan teknologi *deep learning* berbasis *website* tetapi masih belum berjalan dengan baik pada perangkat *mobile*. Padahal gawai yang digunakan sehari-hari oleh sebagian besar masyarakat adalah perangkat *mobile* atau *smartphone*. Sehingga pada penelitian ini penulis melakukan pengembangan aplikasi *mobile* yang diberi nama *UII Skin Analyzer* menggunakan *framework* Flutter untuk membantu pengguna mempercepat proses analisis permasalahan kulit wajah. Pada proses pengembangannya aplikasi ini menerapkan metode *Rapid Application Development* (RAD). Metode ini menggunakan prinsip *agile* yang lebih fleksibel dan melibatkan pengguna secara langsung sehingga dapat menghasilkan sebuah luaran secara cepat dan tepat. Pada penelitian ini, penulis melakukan dua iterasi yang tiap iterasinya dilakukan pengujian untuk memastikan apakah luaran yang diharapkan sudah sesuai atau belum. Hasil akhir dari penelitian ini adalah sebuah aplikasi fungsional berbasis *mobile* dengan sistem operasi Android dengan tampilan yang menarik dan memperhatikan kenyamanan pengguna.

Aplikasi *UII Skin Analyzer* ini telah melalui tahap pengujian usabilitas menggunakan *System Usability Scale* (SUS) dan diperoleh nilai sebesar 82.6%. Dari hasil tersebut dapat ditarik kesimpulan bahwa aplikasi *UII Skin Analyzer* ini berada pada nilai A dan menurut skala Likert memiliki kategori Sangat Baik.

Kata kunci: kulit wajah, *skincare*, *smartphone*, *UII Skin Analyzer*, Flutter, *Rapid Application Development* (RAD), *System Usability Scale* (SUS).

GLOSARIUM

Agile	metodologi pengembangan perangkat lunak yang menekankan pada pengiriman produk cepat dan adaptif kolaborasi tim, pengiriman bertahap, dan responsif terhadap perubahan permintaan pelanggan atau kebutuhan bisnis.
API	<i>Application Programming Interface</i> , antarmuka perangkat lunak yang memungkinkan aplikasi atau sistem berinteraksi dan bertukar data melalui jaringan atau internet.
Array	struktur data dalam pemrograman komputer yang terdiri dari kumpulan elemen dengan jenis data yang sama dan diidentifikasi oleh satu atau beberapa indeks.
BPMN	<i>Business Process Model and Notation</i> , notasi grafis yang digunakan untuk memodelkan dan mendokumentasikan proses bisnis secara visual dalam satu diagram.
Codebase	istilah yang merujuk pada seluruh kode sumber atau kode program yang membentuk suatu aplikasi atau sistem.
Compile	proses konversi kode sumber dari bahasa pemrograman manusia ke dalam bentuk yang dapat dieksekusi oleh mesin atau komputer.
Cross-Platform	istilah yang mengacu pada kemampuan suatu program atau aplikasi untuk berjalan di berbagai sistem operasi atau platform yang berbeda tanpa perlu dimodifikasi ulang.
Deep Learning	sub-bidang <i>machine learning</i> yang menggunakan jaringan saraf tiruan (<i>neural network</i>) dengan beberapa lapisan untuk mempelajari fitur-fitur kompleks dari data.
Flutter	kerangka kerja open-source untuk pengembangan aplikasi seluler dan desktop dengan UI yang kaya dan menarik menggunakan bahasa Dart dari Google.
Framework	kerangka kerja yang menyediakan aturan, prinsip, dan alat-alat terorganisir untuk memudahkan pengembangan aplikasi secara efisien.
Hosting	layanan penyimpanan dan akses internet untuk situs web, aplikasi, atau data agar dapat diakses dan digunakan secara online melalui jaringan internet.

IDE	<i>Integrated Development Environment</i> , perangkat lunak yang menyediakan fitur untuk memudahkan pengembangan dan debugging aplikasi, termasuk editor kode, debugger, dan pengaturan proyek.
Informed Consent	persetujuan sukarela setelah memahami informasi lengkap dan jelas tentang prosedur medis atau penelitian.
JSON	<i>JavaScript Object Notation</i> format pertukaran data ringan dan mudah dibaca serta dipahami, sering digunakan untuk mentransmisikan data di aplikasi web.
Key-value Pairs	konstruksi atau struktur data dalam pemrograman yang terdiri dari dua elemen terkait, yaitu "key" yang berfungsi sebagai identifier dan "value" yang merupakan nilai yang terkait dengan identifier tersebut.
Machine Learning	cabang dari kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan pengalaman tanpa harus secara eksplisit diprogram.
Mobile	perangkat elektronik portabel yang dirancang untuk digunakan saat bepergian atau berpindah-pindah, seperti <i>smartphone</i> , tablet, laptop.
Mobile App	<i>mobile application</i> , program komputer yang dirancang untuk dijalankan pada perangkat <i>mobile</i> .
Native Code	kode yang ditulis dalam bahasa mesin atau bahasa pemrograman tingkat rendah yang dapat langsung dijalankan oleh CPU tanpa memerlukan interpretasi atau kompilasi tambahan.
Open Source	model pengembangan perangkat lunak dengan kode sumber terbuka untuk umum dan dapat diakses untuk dilihat, dimodifikasi, atau dikembangkan kembali oleh siapa saja.
Outline wajah	sketsa atau garis besar dari wajah seseorang, yang biasanya digunakan sebagai panduan untuk menggambar atau melukis potret. singkatnya, sketsa wajah.
Parse	proses mengurai atau memecahkan data yang terkandung dalam suatu format tertentu, seperti JSON, menjadi bentuk data yang lebih mudah diproses oleh program komputer.
Platform	suatu lingkungan yang menyediakan infrastruktur dan sumber daya yang diperlukan untuk membangun dan menjalankan aplikasi/layanan, baik itu dalam bentuk perangkat keras maupun perangkat lunak.

RAD	<i>Rapid Application Development</i> adalah suatu metode pengembangan perangkat lunak yang memungkinkan pengembang untuk membuat aplikasi dengan cepat dan efisien dengan memanfaatkan alat dan kerangka kerja yang telah tersedia.
Request	permintaan yang dilakukan oleh sebuah aplikasi untuk mengambil atau mengubah data dari suatu API.
Response	balasan atau hasil yang diterima setelah mengirimkan permintaan API ke server.
SDK	<i>Software Development Kit</i> , kumpulan alat dan perpustakaan perangkat lunak yang membantu pengembang dalam membuat aplikasi untuk suatu platform atau sistem operasi tertentu.
Server	sebuah komputer atau sistem yang menyediakan layanan atau sumber daya, seperti data atau aplikasi, kepada komputer atau perangkat lain di dalam jaringan.
Stakeholder	individu, kelompok, atau organisasi yang memiliki kepentingan terkait dengan suatu proyek, perusahaan, atau inisiatif tertentu.
SUS	<i>System Usability Scale</i> , kuesioner standar yang digunakan untuk mengukur tingkat kegunaan atau usability suatu produk atau sistem.
Usability Testing	Usability testing adalah suatu metode untuk mengevaluasi seberapa mudah pengguna dapat menggunakan suatu produk atau sistem, dan mendeteksi masalah potensial yang perlu diperbaiki untuk meningkatkan pengalaman pengguna.
Source Code	kumpulan instruksi dalam bahasa pemrograman yang digunakan untuk membuat program komputer. Digunakan untuk mengembangkan, memperbaiki, dan menambahkan fitur pada program.
UI	<i>User Interface</i> , antarmuka pengguna yang mencakup elemen-elemen seperti grafis, tombol, dan menu untuk berinteraksi dengan sistem atau aplikasi yang melibatkan estetika, kemudahan, fungsionalitas, dll.
UX	<i>User Experience</i> , pengalaman keseluruhan pengguna saat berinteraksi dengan suatu produk, layanan, atau sistem, termasuk emosi, persepsi, dan respons terhadap interaksi tersebut.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvi
DAFTAR GAMBAR.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Metodologi Secara Umum	5
1.7 Rancangan Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Kulit Wajah Manusia	8
2.2 Perawatan Kulit Wajah	8
2.3 Permasalahan Kulit Wajah.....	9
2.3.1 Jerawat	9
2.3.2 Keriput	9
2.3.3 Kemerahan	10
2.3.4 Bercak Hitam	10
2.3.5 Jenis Kulit	11
2.4 Purwarupa UII <i>Skin Analyzer</i>	11
2.5 UI/UX	11
2.6 API.....	12

2.7	JSON	13
2.8	<i>Smartphone</i>	14
2.9	<i>Mobile App</i>	15
2.10	Android Studio	15
2.11	Flutter	16
2.11.1	Dart	16
2.11.2	Arsitektur Flutter	17
2.11.3	Flutter <i>Widget</i>	17
2.11.4	Flutter <i>Packages</i>	18
2.11.5	<i>Framework</i> untuk <i>Mobile Application</i>	18
2.12	SQLite	20
2.13	Rapid Application Development	20
2.14	Review Aplikasi Sejenis	21
2.14.1	Face Mapping Dermalogica	21
2.14.2	Olay Skin Advisor	21
2.14.3	L'Oréal Paris Skin Genius	21
BAB III METODOLOGI PENELITIAN		23
3.1	<i>Rapid Application Development (RAD)</i>	23
3.2	Fase Perencanaan	23
3.2.1	Permasalahan Saat ini	24
3.2.2	<i>Review Aplikasi Sejenis</i>	24
3.2.3	Analisis Kebutuhan	24
3.3	Fase Perancangan	25
3.3.1	Iterasi Ke-1	26
3.3.2	<i>Feedback Calon Pengguna</i>	46
3.3.3	Iterasi Ke-2	50
3.4	Fase Implementasi	66
3.5	Fase Pengujian	66
BAB IV HASIL DAN PEMBAHASAN		67
4.1	Hasil	67
4.1.1	Analisis Hasil Desain	67
4.1.2	Halaman Logo	68
4.1.3	Halaman <i>Onboarding</i>	69
4.1.4	<i>Pop-up Informed Consent</i>	70

4.1.5	Pembungkus Menu Utama	71
4.1.6	Halaman Beranda	71
4.1.7	Halaman Tutorial Menu Analisis	72
4.1.8	Halaman Menu Analisis	73
4.1.9	Mekanisme Pengolahan Gambar Pada Halaman Menu Analisis	74
4.1.10	<i>Pop-up</i> Bantuan dan Panduan Pengambilan Foto	78
4.1.11	Halaman Kamera	78
4.1.12	Mekanisme Analisis	79
4.1.13	Halaman Hasil Analisis	81
4.1.14	Mekanisme Penanda Gambar Hasil Analisis	82
4.1.15	Mekanisme Menyimpan Gambar Hasil Analisis	87
4.1.16	Halaman Riwayat	90
4.1.17	Mekanisme Kalender Hasil Analisis	92
4.1.18	Mekanisme Grafik Hasil Analisis	96
4.1.19	Halaman Profil	100
4.2	Pembahasan	104
4.2.1	Pengujian <i>BlackBox</i>	104
BAB V KESIMPULAN DAN SARAN		111
5.1	Kesimpulan	111
5.2	Saran	112
DAFTAR PUSTAKA		113
LAMPIRAN		116

DAFTAR TABEL

Tabel 2.1 Perbandingan aplikasi sejenis UII <i>Skin Analyzer</i>	22
Tabel 3.1 Kebutuhan fungsional	25
Tabel 3.2 <i>Feedback</i> rancangan aplikasi UII <i>Skin Analyzer</i>	46
Tabel 3.3 Instrumen pengujian <i>System Usability Scale</i> (SUS)	46
Tabel 3.4 Pilihan jawaban dan skor responden	47
Tabel 3.5 Hasil dan rata-rata <i>system usability scale</i> (SUS)	47
Tabel 3.6 Rangkuman temuan aplikasi UII <i>Skin Analyzer</i>	50
Tabel 4.1 Pengujian <i>BlackBox</i> pada saat membuka aplikasi	104
Tabel 4.2 Pengujian <i>BlackBox</i> pada halaman beranda	105
Tabel 4.3 Pengujian <i>BlackBox</i> pada halaman menu analisis jerawat	106
Tabel 4.4 Pengujian <i>BlackBox</i> pada halaman menu analisis selain jerawat	108
Tabel 4.5 Pengujian <i>BlackBox</i> pada halaman riwayat	108
Tabel 4.6 Pengujian <i>BlackBox</i> pada halaman profil	109

DAFTAR GAMBAR

Gambar 1.1 Statistik rata-rata penggunaan aplikasi <i>smartphone</i> harian di dunia.....	3
Gambar 2.1 Ilustrasi mekanisme API	12
Gambar 2.2 Respon JSON UII <i>Skin Analyzer</i>	14
Gambar 2.3 Ilustrasi Flutter <i>Widget</i>	17
Gambar 2.4 Flutter <i>Packages Dependencies</i> UII <i>Skin Analyzer</i>	18
Gambar 2.5 Fase pengembangan pada metode <i>Rapid Application Development</i>	20
Gambar 3.1 Diagram BPMN (iterasi ke-1).....	26
Gambar 3.2 Diagram Aktivitas (iterasi ke-1)	28
Gambar 3.3 <i>Use Case Diagram</i> (iterasi ke-1)	29
Gambar 3.4 Halaman Logo (iterasi ke-1)	30
Gambar 3.5 Halaman <i>Onboarding</i> (iterasi ke-1)	31
Gambar 3.6 <i>Pop-up Informed Consent</i> (iterasi ke-1).....	32
Gambar 3.7 Halaman Tutorial (iterasi ke-1).....	33
Gambar 3.8 Halaman Selamat Datang (iterasi ke-1)	34
Gambar 3.9 Halaman <i>Pop-up</i> Tombol Bantuan Halaman Selamat Datang (iterasi ke-1)	35
Gambar 3.10 Halaman <i>pop-up</i> tombol panduan pengambilan foto (iterasi ke-1)	36
Gambar 3.11 Halaman kamera (iterasi ke-1).....	37
Gambar 3.12 Halaman beranda (iterasi ke-1)	38
Gambar 3.13 Halaman Tutorial Menu Analisis (iterasi ke-1)	39
Gambar 3.14 Halaman Menu Analisis (iterasi ke-1)	40
Gambar 3.15 Halaman <i>Pop-up</i> Tombol Bantuan Halaman Analisis (iterasi ke-1)	41
Gambar 3.16 Halaman <i>Loader</i> Saat Analisis (iterasi ke-1)	42
Gambar 3.17 Halaman Hasil Analisis (iterasi ke-1)	43
Gambar 3.18 Halaman Riwayat (iterasi ke-1)	44
Gambar 3.19 Halaman Profil (iterasi ke-1).....	45
Gambar 3.20 Kriteria <i>System Usability Scale</i> (SUS).....	49
Gambar 3.21 Diagram BPMN	51
Gambar 3.22 Diagram Aktivitas	52
Gambar 3.23 <i>Use Case Diagram</i>	53
Gambar 3.24 Halaman Logo (iterasi ke-2)	54
Gambar 3.25 Halaman <i>Onboarding</i> (iterasi ke-2)	55
Gambar 3.26 Halaman <i>Pop-up Informed Consent</i> (iterasi ke-2)	56

Gambar 3.27 Halaman Tutorial Menu Analisis (iterasi ke-2)	57
Gambar 3.28 Halaman Menu Analisis (iterasi ke-2)	58
Gambar 3.29 Halaman <i>Pop-up</i> Pesan dan Panduan Pengambilan Foto (iterasi ke-2)	59
Gambar 3.30 Halaman Hasil Analisis (iterasi ke-2)	60
Gambar 3.31 Halaman Riwayat (iterasi ke-2)	61
Gambar 3.32 Halaman Riwayat (iterasi ke-2)	62
Gambar 3.33 Halaman Detail Riwayat (iterasi ke-2)	63
Gambar 3.34 Halaman Profil (iterasi ke-2)	64
Gambar 3.35 Halaman Masuk (iterasi ke-2)	65
Gambar 4.1 Hasil halaman Logo	69
Gambar 4.2 Hasil halaman <i>Onboarding</i>	70
Gambar 4.3 <i>Pop-up Informed Consent</i>	71
Gambar 4.4 Hasil halaman Beranda	72
Gambar 4.5 Hasil halaman Tutorial Menu Analisis	73
Gambar 4.6 Hasil halaman Menu Analisis	74
Gambar 4.7 Fungsi <i>_fetchCachedImage</i>	75
Gambar 4.8 Fungsi <i>_pickImage</i>	76
Gambar 4.9 Fungsi <i>_cachingImage</i>	77
Gambar 4.10 <i>Pop-up</i> Bantuan dan Panduan Pengambilan Foto	78
Gambar 4.11 Halaman Kamera	79
Gambar 4.12 Fungsi <i>_uploadImage</i>	81
Gambar 4.13 Halaman Hasil Analisis	82
Gambar 4.14 <i>Widget CustomPaint</i>	83
Gambar 4.15 Fungsi <i>paint</i> Pada Kelas <i>DrawInCanvas</i>	85
Gambar 4.16 Halaman Hasil Analisis Berdasarkan Fitur	86
Gambar 4.17 <i>Pop-up</i> Simpan Hasil Analisis	87
Gambar 4.18 Fungsi <i>_saveToDB</i>	89
Gambar 4.19 Kelas <i>DBHelper</i>	90
Gambar 4.20 Halaman Riwayat	91
Gambar 4.21 Fungsi <i>_fetchAndSetAnalysisHistory</i>	92
Gambar 4.22 <i>Widget</i> Kalender	93
Gambar 4.23 Halaman Detail Hasil Analisis	93
Gambar 4.24 Kode <i>CalendarCardWidget</i>	95
Gambar 4.25 Kode <i>CalendarWidget</i>	96

Gambar 4.26 <i>ChartCardWidget</i>	97
Gambar 4.27 Kode <i>ChartCardWidget</i>	100
Gambar 4.28 Halaman Profil	101
Gambar 4.29 Halaman Profil	102
Gambar 4.30 Fungsi <i>_onPressedSignIn</i>	104

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kulit pada manusia memiliki peranan vital sebagai pelindung organ tubuh di bawahnya serta menyumbang sekitar 12-15% dari total berat badan (Martini & Nath, 2009). Kulit merupakan organ tubuh yang paling besar dan salah satu yang paling kompleks serta memiliki banyak peran dalam mempertahankan kehidupan serta kesehatan, tetapi juga dapat memiliki banyak potensi masalah yaitu dengan lebih dari 3.000 kemungkinan kelainan kulit. Kulit adalah organ yang unik dan membutuhkan perhatian besar baik dalam hal penyakit yang mungkin terjadi maupun kesehatannya.

Seiring dengan hal itu, banyak orang yang saat ini memberikan perhatian lebih terhadap kondisi kulitnya khususnya di bagian wajah. Wajah adalah organ pusat ekspresi, kognisi, dan komunikasi manusia. Terlebih lagi, wajah adalah bagian tubuh yang pertama kali dilihat oleh orang lain. Akan tetapi, pada faktanya beberapa orang memiliki masalah dengan kulit wajah mereka, hal ini dapat mempengaruhi mereka secara psikologis (Salman dkk., 2016), salah satunya dapat mempengaruhi kepercayaan diri mereka.

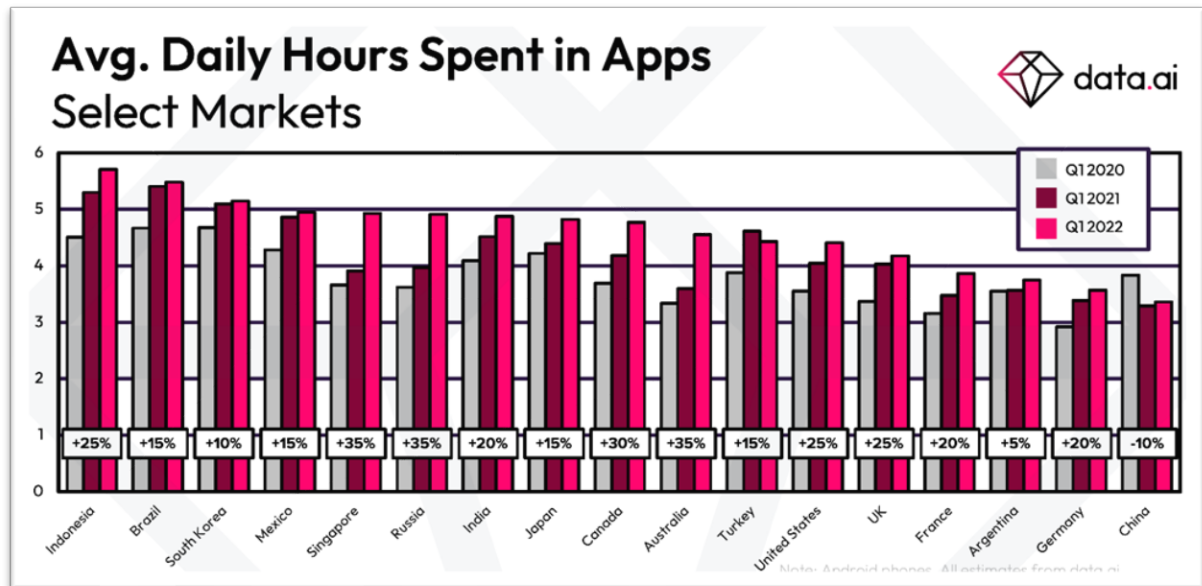
Menurut survei yang dilakukan pada tahun 2021 oleh MarkPlus Inc kepada sekitar 6000 responden diketahui bahwa kesadaran akan perawatan kesehatan kulit ini cukup tinggi pada wanita Indonesia. Tercatat bahwa 67,1% dari responden mengatakan bahwa kecantikan adalah memiliki kulit yang bersih dan mulus. Selanjutnya, sekitar 70,3% mengatakan bahwa manfaat skincare yang dicari adalah yang mampu mencerahkan kulit. Di samping itu, makeup yang merupakan salah satu cara untuk mempercantik diri, ternyata menjadi tidak terlalu penting bagi wanita Indonesia saat ini. Hanya sekitar 8,7% wanita saja yang mengatakan bahwa cantik itu jika menggunakan make-up. Istilah perawatan kulit sangat penting untuk dilakukan oleh setiap orang, istilah ini tidak hanya cenderung melekat pada wanita saja tetapi pria juga perlu melakukan perawatan kulit untuk memelihara kesehatan dan mencegah permasalahan kulit karena pada dasarnya perawatan kulit tidak terbatas hanya pada *gender* tertentu serta manfaat yang dirasakan akan sama pentingnya untuk pria. Survei menunjukkan bahwa hampir seluruh wanita Indonesia (98,8%) setuju bahwa pria juga memerlukan perawatan kulit wajah. (MarkPlus Inc, 2021).

Jika ingin mendapatkan hasil perawatan yang maksimal, bahan-bahan yang digunakan harus sesuai dan tepat dengan kondisi kulit wajah pengguna. Penggunaan bahan *skincare* dan kosmetik yang tidak tepat, dapat menjadi pemicu berbagai efek yang berbahaya. Menurut Dr. Fajar Waskito, Sp.K.K(K), pada kasus yang ditemuinya menunjukkan bahwa kelainan kulit dapat terjadi akibat penggunaan kosmetik yang tidak sesuai dengan jenis kulit pengguna sehingga timbul reaksi alergi (Ratnadita, 2016). Selain itu, untuk pemakaian yang salah selama bertahun-tahun dapat mengakibatkan zat-zat yang tidak diinginkan mengendap di dalam kulit dan memicu timbulnya kanker (Pangaribuan, 2017). Oleh sebab itu, sebelum menggunakan bahan *skincare*/kosmetik tertentu sangat disarankan untuk melakukan pengecekan untuk mengetahui kondisi kulit wajah. Pada umumnya praktik yang benar dalam melakukan proses *assessment*/penilaian untuk mengetahui kondisi kulit wajah yaitu dengan berkonsultasi ke dokter kulit. Namun, hanya sebagian kecil orang yang dapat melakukan hal tersebut.

Pada saat ini telah dibuat sebuah purwarupa sistem analisis kulit wajah (*Skin Analyzer*) berdasarkan citra wajah. Purwarupa ini berperan sebagai alat bantu yang menerapkan *machine learning*. Alat ini nantinya akan belajar dari pengetahuan yang dimiliki oleh dokter kulit dalam melakukan *assessment*/penilaian masalah kulit dan jenis kulit wajah. Purwarupa ini dibuat untuk membantu pengguna dalam mengidentifikasi masalah kulit yang mungkin terjadi di antaranya jerawat, keriput, kemerahan, bintik hitam, hingga deteksi jenis kulit wajah pengguna. Saat ini, sistem telah tersedia secara *online* dan dapat diakses melalui pemanggilan dari API sistem tersebut.

Smartphone telah menjadi hal yang sangat lumrah pada saat ini. Di Indonesia sendiri, pada tahun 2020, pengguna *smartphone* mencapai 61,7% dari total populasi atau sekitar 170,4 juta pengguna. Adapun di seluruh dunia, pengguna *smartphone* berada di kisaran 4,3 miliar pengguna (Pusparisa, 2021). Hal tersebut disebabkan oleh berbagai kemudahan yang ditawarkan oleh *smartphone* di antaranya dapat mengakses informasi di mana pun dan kapan pun, ukuran yang kecil sehingga mudah untuk dibawa, kuat dan dapat diandalkan serta konsumsi daya yang rendah dan sebagainya. Pernyataan tersebut diperkuat dengan adanya data yang menunjukkan bahwa setiap tahunnya rata-rata penggunaan aplikasi *smartphone* harian di dunia terus mengalami peningkatan. Di Indonesia sendiri rata-rata penggunaan pada kuartal pertama tahun 2020 menunjukkan angka sekitar 4,5 jam perhari. Kemudian pada kuartal pertama tahun 2021 mengalami kenaikan menjadi sekitar 5,25 jam perhari. Selanjutnya pada kuartal pertama tahun 2022 juga mengalami kenaikan menjadi hampir 6 jam perhari.

Hal ini dapat terlihat pada Gambar 1.1 yang menunjukkan rata-rata penggunaan aplikasi *smartphone* harian di dunia (Bikker, 2022).



Gambar 1.1 Statistik rata-rata penggunaan aplikasi *smartphone* harian di dunia

API atau *Application Programming Interface* merupakan *interface* yang dapat menjadi penghubung antara aplikasi satu dengan aplikasi lainnya. Dengan kata lain, API adalah perantara antar berbagai aplikasi yang berbeda, baik dalam satu *platform* yang sama bahkan lintas *platform* (Jacobson dkk., 2011). Sistem *Skin Analyzer* yang telah dikembangkan berfungsi sebagai sistem yang akan menganalisis kulit wajah. Dengan bantuan dari API, sistem tersebut dapat terhubung pada aplikasi *mobile* yang nantinya akan dikembangkan. Pengguna dapat memasukkan gambar yang ingin dianalisis melalui dua cara, yaitu memilih dari galeri atau mengambil dari kamera. Selanjutnya, data berupa gambar yang sudah dimasukkan ke aplikasi akan dikirimkan ke API. Kemudian API yang menjembatani proses pendeteksian objek akan meneruskan ke sistem *Skin Analyzer*. Akhirnya, sistem *Skin Analyzer* menganalisis gambar dan mengirimkan hasil analisis dalam bentuk JSON yang sudah tertata dan mudah dipahami, sehingga aplikasi *mobile* dapat menampilkan hasil analisis yang sudah diformat dengan baik.

Flutter adalah *open-source software development kit* yang dikembangkan oleh Google. Flutter dapat mengembangkan aplikasi *cross-platform* atau lintas *platform* artinya dengan satu kali penulisan kode dapat dihasilkan atau di-*compile* ke berbagai *platform*. Hingga saat ini, Flutter dapat membuat aplikasi pada enam platform yang berbeda dengan *codebase* yang sama.

Flutter juga memiliki kemampuan untuk membuat aplikasi dengan user interface yang menarik dengan lebih mudah, tanpa mengurangi performa dari aplikasi itu sendiri.

Berdasarkan uraian di atas, dalam penelitian ini akan dibuat sebuah aplikasi berbasis *mobile* menggunakan Flutter untuk mendeteksi dan menghitung lesi masalah kulit yang mungkin terjadi, yaitu jerawat, keriput, kemerahan, bintik hitam, hingga deteksi jenis kulit wajah pengguna pada suatu citra digital. Oleh karena itu, peneliti membuat penelitian yang berjudul **“Pengembangan Aplikasi *Mobile* UI *Skin Analyzer* dengan Flutter Untuk Membantu Analisis Permasalahan Kulit Wajah”**. Harapannya dengan penelitian ini mampu mengembangkan aplikasi berbasis *mobile* yang dapat digunakan untuk melakukan asesmen perawatan kulit secara mandiri dengan mengirimkan sebuah gambar melalui *smartphone*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana mengembangkan aplikasi bergerak UI *Skin Analyzer* menggunakan Flutter untuk membantu proses asesmen analisis permasalahan kulit wajah?
- b. Bagaimana merancang aplikasi bergerak UI *Skin Analyzer* dengan *user interface* dan *user experience* yang baik?
- c. Bagaimana mengevaluasi aplikasi bergerak UI *Skin Analyzer* terkait fungsionalitas dan penerimaan pengguna dengan metode *usability testing* menggunakan *system usability scale*?
- d. Bagaimana menyediakan fitur riwayat dan profil yang memperhatikan privasi pengguna agar pengguna tertarik untuk terus menggunakan aplikasi dalam jangka panjang?

1.3 Batasan Masalah

Untuk memperjelas lingkup permasalahan sehingga pengerjaan lebih terarah, maka batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Bahasa pemrograman yang digunakan adalah dart dengan *framework* Flutter.
- b. Aplikasi yang dikembangkan hanya untuk perangkat mobile dengan sistem Android.
- c. Aplikasi menggunakan algoritma *deep learning* yang sudah dikembangkan oleh tim lain dan dihosting sehingga fokus pada tugas akhir ini adalah pengembangan aplikasi *mobile* yang dapat melakukan pemanggilan API yang sudah tersedia untuk melakukan analisis permasalahan kulit wajah.

- d. Pengambilan gambar langsung melalui kamera yang memiliki outline bentuk wajah.
- e. Mengunggah gambar dari galeri pengguna dengan resolusi minimal 400x400px.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Membuat rancangan aplikasi bergerak UII *Skin Analyzer* dengan *user interface* dan *user experience* yang baik.
- b. Mengembangkan aplikasi bergerak UII *Skin Analyzer* menggunakan Flutter untuk membantu proses asesmen analisis permasalahan kulit wajah.
- c. Mengetahui tingkat kesesuaian fungsionalitas yang telah dirancang dan tingkat penerimaan pengguna terkait adanya aplikasi bergerak UII *Skin Analyzer* dengan metode *usability testing* menggunakan *system usability scale*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut: Tujuan dari penelitian ini adalah sebagai berikut:

- a. Memudahkan pengguna untuk menganalisis kulit wajah hanya melalui aplikasi UII *Skin Analyzer* pada perangkat bergerak/*smartphone*.
- b. Dengan kemudahan menganalisis permasalahan kulit wajah diharapkan dapat mengurangi resiko kesalahan dalam pemilihan dan penggunaan bahan *skincare* atau kosmetik yang tidak tepat.
- c. Hasil penelitian ini dapat dijadikan sebagai acuan untuk penelitian lebih lanjut terkait penggunaan framework Flutter untuk mengembangkan aplikasi bergerak.

1.6 Metodologi Secara Umum

Dalam penelitian ini, terdapat beberapa langkah yang dapat digunakan untuk mencapai tujuan yang diinginkan sebagai berikut:

a. Pengumpulan Data

1. Studi Literatur

Mencari sumber referensi yang berkaitan dengan penelitian terkait pengembangan aplikasi mobile dengan Flutter.

2. Perbandingan Aplikasi Sejenis

Mencari dan membandingkan kelebihan dan kekurangan dari aplikasi sejenis yang berfungsi untuk mendeteksi permasalahan kulit wajah dengan citra digital.

b. Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem agar memudahkan peneliti dalam proses implementasi sistem. Rancangan yang dibuat yaitu berupa diagram alir dengan flowchart, BPMN (*Business Process Modelling Notation*), dan desain antar muka pengguna (*user interface*).

c. Implementasi Sistem

Pada tahap ini dilakukan implementasi hasil studi literatur yang diaplikasikan sebagai solusi atas permasalahan yang ditemukan. Tahap ini meliputi pengembangan aplikasi sesuai dengan rancangan antarmuka yang telah dibuat dan menghubungkan dengan API yang sudah tersedia.

d. Pengujian Sistem

Pada tahap ini dilakukan pengujian sistem terkait fungsionalitas dengan metode *usability testing* menggunakan *system usability scale*.

1.7 Rancangan Sistematika Penulisan

Dalam penelitian ini terdapat beberapa bab yang saling terkait untuk memudahkan pembaca dalam memahaminya. Deskripsi singkat dari bab ini adalah: sebagai berikut:

a. Bab I Pendahuluan

Bab ini berisi tentang latar belakang masalah yang mendasari dilakukannya penelitian mengenai pembuatan aplikasi *mobile* UI *Skin Analyzer* dengan Flutter, rumusan masalah yang menjadi acuan dalam pemecahan masalah yang ada, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

b. Bab II Landasan Teori

Bab ini berisikan penjelasan dasar teori yang berhubungan dengan permasalahan kulit wajah, konsep aplikasi *mobile*, analisis perbandingan dengan aplikasi sejenis, teori dasar mengenai metode perancangan system, dan pengujian sistem yang akan digunakan.

c. Bab III Metodologi Penelitian

Bab ini berisi tentang metodologi dan perancangan aplikasi *mobile* UI *Skin Analyzer* dengan Flutter yang meliputi metodologi perancangan aplikasi *mobile*, analisis kebutuhan, diagram perancangan sistem serta desain antarmuka sistem.

d. Bab IV Hasil dan Pembahasan

Bab ini berisi hasil dan pembahasan pada penelitian yang terdiri dari setiap proses dan fungsionalitas dalam sistem, pengujian sistem, kelebihan dan kekurangan sistem.

e. Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan tentang hasil penelitian sebagai penanda dari penelitian ini sudah sesuai atau belum dan saran agar peneliti lain yang ingin memperdalam penelitian dapat melakukan penelitian kembali lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Kulit Wajah Manusia

Kulit merupakan organ tubuh manusia yang terbesar. Kulit memiliki luas sekitar 1.5 m² pada manusia dewasa, dan memiliki berat sekitar 15% dari berat badan. Kulit memiliki fungsi sebagai barier fisik, perlindungan terhadap agen infeksius, termoregulasi, sensasi, proteksi terhadap sinar ultraviolet (UV), serta regenerasi dan penyembuhan luka (Chu, 2012). Kulit adalah organ tubuh yang kompleks, elastis, sensitif, serta memiliki banyak variasi yang dapat dipengaruhi oleh keadaan iklim, umur, ras, seks, dan lokasi pada tubuh. Kulit adalah organ unik yang membutuhkan perhatian besar baik dalam hal penyakit maupun kesehatannya. Tercatat, kulit memiliki banyak potensi masalah yaitu dengan lebih dari 3.000 kemungkinan kelainan kulit.

Wajah adalah organ pusat ekspresi, kognisi, dan komunikasi manusia. Terlebih lagi, wajah adalah bagian tubuh yang pertama kali dilihat oleh orang lain. Setiap orang memiliki jenis kulit wajah yang berbeda-beda dan dapat berubah seiring bertambahnya usia. Selain itu, jenis kulit wajah juga dapat berubah berdasarkan faktor lingkungan seperti paparan sinar matahari, debu, dan polusi yang berkepanjangan. Setiap jenis kulit wajah memiliki kebutuhan yang berbeda. Suatu jenis kulit wajah dapat membutuhkan perhatian dan perawatan yang lebih dibanding jenis kulit wajah yang lain. Akan tetapi, pada faktanya beberapa orang memiliki masalah dengan kulit wajah mereka, hal ini dapat mempengaruhi mereka secara psikologis (Salman dkk., 2016), salah satunya dapat mempengaruhi kepercayaan diri mereka.

2.2 Perawatan Kulit Wajah

Perawatan kulit wajah merupakan suatu proses atau metode perawatan yang berfungsi untuk menjaga dan memelihara kesehatan kulit dan bertujuan untuk mencegah hingga mengatasi berbagai masalah pada kulit wajah. Perawatan kulit wajah ini sebaiknya dilakukan secara rutin baik perawatan mandiri di rumah maupun menggunakan bantuan dokter di rumah sakit atau di klinik kecantikan atau yang biasa dikenal dengan klinik estetika.

Pada saat memasuki usia 30an kulit pada umumnya mulai terasa kering dan terlihat kusam. Hal ini disebabkan karena kemampuan tubuh untuk memproduksi sel-sel kulit yang baru mulai melambat. Keriput muncul karena adanya penurunan produksi kolagen dan

akumulasi elastin abnormal (Anindita & AF, 2017). Karena itu, perawatan kecantikan kulit wajah menjadi hal yang sangat penting bagi wanita usia 30an, terlebih pada rentang usia tersebut sudah semakin dekat dengan menurunnya vitalitas dan kemampuan regenerasi kulit. Oleh sebab itu, perlu dilakukan perawatan secara teratur serta menyeluruh untuk menjaga dan mempertahankan kecantikan serta kesehatan kulit wajah.

Perawatan kulit wajah dapat dilakukan dengan berbagai macam jenis kosmetika di antaranya adalah susu pembersih (*milk cleanser*), penyegar (*toner*), pelembab (*moisturizer*), tabir surya (*sunscreen*), dan masih banyak lagi. Kemudian, terdapat perawatan yang menggunakan kosmetik *peeling*, *massage* dan *face mask* yang jika dilakukan secara rutin setiap hari dapat mengurangi masalah kelainan yang terjadi pada kulit wajah dengan berbagai macam teknik dan metode yang sesuai dengan kondisi kulit wajah. Hingga saat ini, banyak sekali jenis perawatan kulit wajah yang berkembang dan telah banyak digunakan. Oleh sebab itu, mengetahui kondisi kulit wajah merupakan kunci dalam memilih perawatan kulit yang tepat.

2.3 Permasalahan Kulit Wajah

Pada penelitian ini aplikasi *mobile* mampu mengakses API yang dapat melakukan analisis permasalahan kulit wajah sebagai berikut: Cucu Subbab tidak perlu penomoran tetapi dicetak tebal. Cucu Subbab tidak perlu penomoran tetapi dicetak tebal.

2.3.1 Jerawat

Jerawat adalah salah satu penyakit kulit yang selalu mendapat perhatian bagi para remaja dan dewasa muda (Sampelan dkk., 2017). Kondisi ini berhubungan erat dengan produksi minyak (sebum) secara berlebihan. Padahal di setiap pori-pori kulit, terdapat folikel yang terdiri dari kelenjar minyak dan folikel rambut. Jerawat sering terjadi pada kulit wajah, leher dan punggung, baik pada laki-laki maupun perempuan (Susanto, 2013). Kulit yang berminyak menyebabkan pori-pori tersumbat, sehingga bakteri *anaerobic* seperti *Staphylococcus aureus* akan berkembang biak dengan cepat dan menyebabkan timbulnya jerawat (Mumpuni, 2011).

2.3.2 Keriput

Penuaan kulit secara umum dapat dibagi menjadi dua kategori, yakni penuaan intrinsik atau penuaan kronologis yang terkait dengan semakin bertambahnya usia dan penuaan ekstrinsik yang terkait dengan paparan faktor-faktor luar (Sjerobabski Masnec & Situm, 2010). Faktor ekstrinsik yang paling utama sebagai penyebab dalam mempercepat proses penuaan

kulit yaitu, paparan sinar matahari yang mengandung sinar ultraviolet (UV), sehingga penuaan kulit ekstrinsik sering disebut juga sebagai *photoaging* (Helfrich dkk., 2008). Faktor intrinsik yang ditimbulkan oleh pertambahan usia adalah dikarenakan menurunnya kemampuan tubuh untuk melakukan regenerasi, dalam hal ini khususnya adalah produksi kolagen dan elastin yang menurun. Kolagen dan elastin merupakan komponen utama matriks ekstraselular yang menentukan kekuatan jaringan (Word dkk., 2009). Kolagen merupakan protein yang terdapat di kulit, tendon, tulang rawan, dan organ dengan kandungan sekitar 30% atau lebih dari protein total (Rahman dkk., 2021).

2.3.3 Kemerahan

Kemerahan pada kulit dapat disebabkan oleh peningkatan jumlah hemoglobin jenuh, peningkatan diameter atau jumlah sebenarnya dari kapiler kulit, atau kombinasi dari faktor-faktor ini (Fitzpatrick & Wolff, 2008). Kemerahan atau kondisi sebagian kulit menjadi merah terjadi merupakan akibat dari adanya peningkatan aliran darah. Apabila ada aliran darah yang meningkat dalam jumlah yang banyak menuju area kulit terutama wajah (pipi), pembuluh darah akan secara otomatis membesar untuk mengimbangnya. Pembesaran inilah yang memberikan efek "memerah" pada kulit. Seringkali, kemerahan pada wajah dapat disebabkan oleh alergi, iritasi, tidak cocok dengan kandungan *skincare*, hormon, sering melupakan menghapus riasan wajah dalam waktu yang lama dan lain sebagainya. Adanya kondisi kemerahan pada wajah dapat menimbulkan efek yang negatif terhadap kualitas hidup (Lee dkk., 2021).

2.3.4 Bercak Hitam

Bercak hitam atau flek hitam adalah salah satu masalah kulit yang ditandai dengan munculnya bintik hitam pada wajah, lengan, bahu, dan bagian tubuh lainnya. Bercak hitam atau flek hitam sendiri merupakan kumpulan melanin atau pigmen yang berisi *melanosome*. Flek hitam muncul karena meningkatnya produksi melanin, terutama setelah terpapar oleh sinar matahari atau sinar ultraviolet (UV) (Djuanda, 2007) dan kondisi kulit tertentu atau efek samping dari obat juga bisa memicu timbulnya bercak hitam atau flek hitam. Bercak hitam atau flek hitam dapat muncul dikarenakan oleh berbagai sebab, yakni paparan sinar ultraviolet, terjadi masalah kulit seperti *hiperpigmentasi pasca-inflamasi*, *melasma*, *linea nigra*, *melanosis riehl*, *poikiloderma of civatte*, hingga faktor genetik.

2.3.5 Jenis Kulit

Secara sederhana, terdapat beberapa jenis kulit yaitu kulit kering, berminyak, kombinasi dan sensitif. Disampaikan oleh dr. Flandiana Yogianti, Ph.D., Sp.DV (Satria, 2021), Awam menyebut beberapa jenis kulit, yaitu kulit kering (seperti kusam, tertarik, sedikit mengelupas), kulit berminyak (banyak berminyak pada area-area tertentu), kulit kombinasi (ada kalanya berminyak dan ada kalanya kulitnya kering), dan kulit sensitif. Kulit sensitif ada beberapa jenis, seperti lebih mudah untuk berjerawat, memerah, gatal, mudah alergi dan mengelupas karena pemakaian produk tertentu. Berdasarkan warnanya, dr. Flandiana menjelaskan terdapat 6 tipe kulit yang dituliskan dalam bentuk *range* mulai dari 1-6 di mana 1 adalah tipe kulit yang paling cerah dan 6 adalah tipe kulit yang paling gelap. Untuk pemeriksaan lebih detail mengenai jenis dan tipe kulit secara objektif menggunakan alat ukur melalui dokter spesialis dermatologi dan *venereology*. dr. Flandiana menjelaskan perawatan pada setiap jenis kulit pada dasarnya sama, yaitu memilih pelembab (*moisturizer*) dan pembersih (*cleanser*) yang tepat sesuai dengan jenis kulit masing-masing.

2.4 Purwarupa UII *Skin Analyzer*

Pada saat ini telah dibuat sebuah purwarupa sistem analisis kulit wajah (UII *Skin Analyzer*) berdasarkan citra wajah (Muhimmah dkk., 2023). Purwarupa ini berperan sebagai alat bantu dan telah menerapkan *machine learning*. *Machine learning* adalah sub bidang ilmu komputer yang berkaitan dengan pembuatan algoritma yang mengandalkan kumpulan contoh dari beberapa fenomena (Burkov, 2019). Dengan adanya *machine learning*, sebuah program komputer dapat menirukan kemampuan manusia untuk belajar dari pengalaman. Purwarupa ini nantinya akan belajar dari pengetahuan yang dimiliki oleh dokter kulit dalam melakukan *assessment*/penilaian masalah kulit dan jenis kulit wajah. Purwarupa ini dibuat untuk membantu pengguna dalam mengidentifikasi masalah kulit yang mungkin terjadi di antaranya jerawat, keriput, kemerahan, bintik hitam, hingga deteksi jenis kulit wajah pengguna. Saat ini, sistem telah tersedia secara online dan dapat diakses melalui pemanggilan dari API sistem tersebut.

2.5 UI/UX

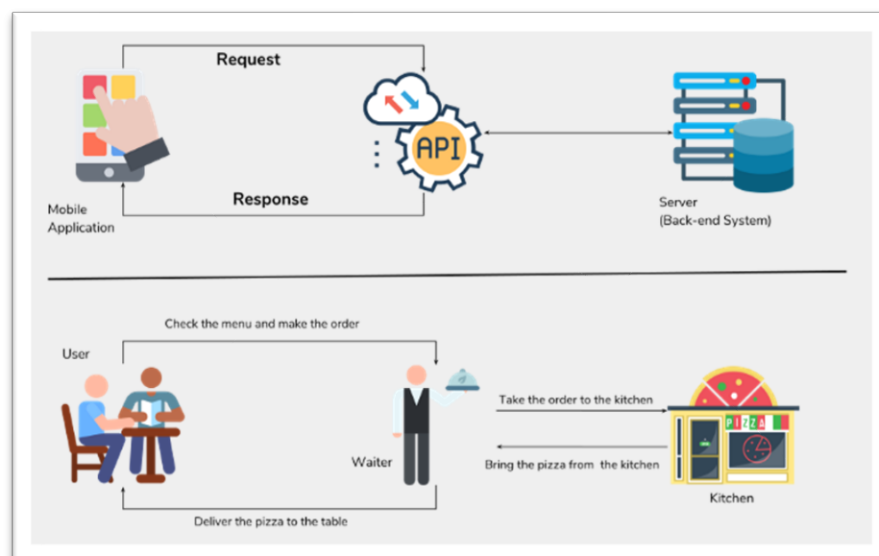
UI (*User Interface*) adalah bagian dari desain aplikasi atau sistem yang berhubungan langsung dengan interaksi antara pengguna dan perangkat lunak. UI meliputi segala komponen visual, seperti tombol, menu, ikon, dan tata letak halaman, yang digunakan untuk

menghubungkan pengguna dengan fungsionalitas aplikasi. UI yang baik dirancang dengan mempertimbangkan kebutuhan dan preferensi pengguna. Desain UI yang efektif harus memudahkan pengguna untuk berinteraksi dengan aplikasi dengan cara yang intuitif, mudah dipahami, dan menyenangkan. Tujuan utama dari UI adalah menciptakan pengalaman pengguna yang baik dan memastikan pengguna dapat mencapai tujuan mereka dengan efisien.

UX (*User Experience*) merujuk pada keseluruhan interaksi dan persepsi pengguna terhadap suatu produk, layanan, atau sistem dalam konteks penggunaannya. Hal ini meliputi aspek-aspek seperti kemudahan penggunaan, kepuasan pengguna, efisiensi, aksesibilitas, estetika visual, dan emosi yang dirasakan oleh pengguna saat berinteraksi dengan produk atau layanan tersebut. Pentingnya UX terletak pada upaya untuk menciptakan pengalaman yang positif, efektif, dan memuaskan bagi pengguna. Dengan mendesain UX yang baik, pengguna dapat lebih mudah dan efisien dalam menggunakan produk, serta merasakan kepuasan dan nilai tambah dari pengalaman tersebut. UX juga berperan dalam membangun loyalitas pengguna, meningkatkan retensi, dan mendukung keberhasilan produk atau layanan di pasar.

2.6 API

API atau *Application Programming Interface* merupakan interface yang dapat menjadi penghubung antara satu aplikasi dengan aplikasi lainnya. Dengan kata lain, API adalah perantara antar berbagai aplikasi yang berbeda, baik dalam satu platform yang sama ataupun lintas platform (Jacobson dkk., 2011).



Gambar 2.1 Ilustrasi mekanisme API

Sumber: Kothalawala (2018)

Seperti pada ilustrasi di atas (lihat Gambar 2.1), API dapat dimisalkan sebagai *waiter* atau pelayan pada sebuah restoran. Pengguna dalam hal ini adalah pelanggan restoran melakukan pemesanan/*request* kepada pelayan. Pelayan akan mengirimkan pesanan tersebut ke dapur agar diproses, dalam hal ini dapur adalah sebuah server. Setelah pesanan selesai dibuat, maka pelayan akan mengantar pesanan/*response* kepada pelanggan.

Sistem purwarupa UII *Skin Analyzer* yang telah dikembangkan berfungsi sebagai sistem yang akan menganalisis kulit wajah. Dengan bantuan dari API, sistem tersebut dapat terhubung pada aplikasi *mobile*. Pengguna mengirimkan gambar dari aplikasi *mobile* di mana gambar tersebut akan dikirimkan ke purwarupa UII *Skin Analyzer* melalui API. Setelah proses deteksi oleh purwarupa selesai, purwarupa UII *Skin Analyzer* akan mengirimkan hasil analisis dalam bentuk JSON yang sudah tertata dan siap untuk digunakan sehingga aplikasi *mobile* dapat menampilkan hasil analisis dengan baik.

2.7 JSON

JSON atau JavaScript Object Notation adalah format file dan format pertukaran data yang ringan dan mudah di-*parse* dan digunakan oleh komputer untuk menyimpan dan mengirimkan data yang terdiri dari *key-value pairs*. Tiap *key-value pairs* dipisahkan oleh koma dan urutan dari *key-value pairs* tersebut tidak mempengaruhi apa-apa. Sintaks JSON dapat dibaca oleh manusia (Nurseitov dkk., 2009). JSON adalah format data yang banyak digunakan dalam *electronic data interchange*, termasuk seperti aplikasi *mobile* dengan server. JSON adalah format data yang tidak bergantung pada bahasa. JSON sendiri berasal dari Bahasa pemrograman Javascript, tetapi banyak bahasa pemrograman modern menyertakan kode untuk menghasilkan dan mengolah data dengan format JSON. Nama file JSON menggunakan ekstensi *.json*.

```

1  {
2    "deteksi_objek": {
3      "jerawat": [
4        {
5          "score": 0.745612681,
6          "xmax": 0.374266744,
7          "xmin": 0.337818826,
8          "ymax": 0.662555754,
9          "ymin": 0.615534246
10        },
11        {
12          "score": 0.714348435,
13          "xmax": 0.498758836,
14          "xmin": 0.4529742,
15          "ymax": 0.181156278,
16          "ymin": 0.13687712
17        },
18        {
19          "score": 0.689856291,
20          "xmax": 0.399748445,
21          "xmin": 0.359186864,
22          "ymax": 0.782688211,
23          "ymin": 0.749757111
24        }
25      ]
26    }
27  }

```

Gambar 2.2 Respon JSON UII *Skin Analyzer*

JSON adalah format dari *response* yang diberikan oleh server purwarupa aplikasi UII *Skin Analyzer*. Pada contoh *response* di atas (perhatikan Gambar 2.2), akan diberikan sebuah *array* jerawat yang berisi data-data akan jerawat yang terdeteksi pada gambar yang dikirimkan ke server. Data-data tersebut adalah data *score* kemungkinan hal tersebut adalah jerawat dan sama seperti koordinat di mana jerawat tersebut terdeteksi. Data tersebut selanjutnya digunakan untuk menampilkan gambar dengan lingkaran penanda pada area yang terdeteksi jerawat, sesuai dengan respon yang diberikan oleh server.

2.8 Smartphone

Smartphone adalah perangkat komputer *portable* yang menggabungkan fungsi telepon seluler dan komputasi menjadi satu kesatuan. *Smartphone* berbeda dengan *feature phone*, yaitu jenis ponsel yang mempertahankan bentuk ponsel generasi sebelumnya dengan input berbasis tombol fisik dan layar non-sentuh yang kecil. *Smartphone* memiliki kemampuan *hardware* dan sistem operasi yang lebih baik. *Smartphone* dilengkapi dengan kemampuan untuk menampilkan foto, bermain *game*, memutar video, navigasi, kamera, memutar dan merekam audio/video, mengirim/menerima email, menjelajahi web, *wireless* internet dan masih banyak lagi (Miftah dkk., 2013). *Smartphone* biasanya berisi sejumlah *metal-oxide-semiconductor* (MOS) *integrated circuit* (IC) *chips*, berbagai sensor yang dapat dimanfaatkan oleh software modern seperti magnetometer, sensor jarak, barometer, giroskop, accelerometer dan lainnya, dan mendukung protokol komunikasi nirkabel seperti Bluetooth, Wi-Fi, atau navigasi satelit.

2.9 Mobile App

Mobile App atau aplikasi bergerak adalah program komputer atau aplikasi perangkat lunak yang dirancang untuk berjalan di perangkat bergerak seperti ponsel, tablet, atau jam tangan. *Mobile app* berbeda dengan aplikasi desktop yang dirancang untuk berjalan di komputer desktop dan aplikasi web yang berjalan di browser. *Mobile app* awalnya ditujukan untuk menambah produktivitas seperti email, kalender, dan kontak, tetapi dengan meningkatnya kemampuan komputasi dan semakin lumrahnya perangkat *mobile*, permintaan akan aplikasi saat ini mulai masuk ke sektor yang lain seperti *game*, *factory automation*, GPS dan *location-based service*, *order tracking*, dan pembelian tiket, sehingga saat ini terdapat jutaan aplikasi *mobile* yang tersedia. Aplikasi *mobile* umumnya diunduh dari *store* pada masing-masing sistem operasi. Aplikasi *mobile* memainkan peran yang terus meningkat dalam perawatan kesehatan, dan ketika dirancang dan diintegrasikan dengan benar dapat menghasilkan banyak sekali manfaat (Lee Ventola, 2014).

Aplikasi *mobile* dapat diklasifikasikan menjadi empat jenis: aplikasi native, aplikasi hybrid, aplikasi web, dan aplikasi multi-platform. Aplikasi native dirancang khusus untuk sistem operasi tertentu, seperti iOS atau Android. Aplikasi web dapat berjalan di tiap sistem operasi yang mendukung koneksi internet dan dapat menjalankan browser. Aplikasi hybrid adalah aplikasi web yang dimasukkan ke dalam native *app shell*. Contoh aplikasi hybrid adalah aplikasi yang dikembangkan menggunakan React Native. Saat ini, telah dikembangkan jenis aplikasi *mobile* yang baru, yaitu aplikasi multi-platform. Aplikasi multi-platform dapat menghasilkan aplikasi pada beberapa platform yang berbeda, dengan satu codebase yang sama. Contoh aplikasi multi-platform adalah aplikasi yang dikembangkan menggunakan Flutter.

2.10 Android Studio

Android Studio adalah *Integrated Development Enviroment* (IDE) resmi dari Google untuk mengembangkan aplikasi berbasis sistem operasi Android, yang dibangun di atas *software* JetBrains oleh IntelliJ IDEA (Google, 2013). Android merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya adalah IDE utama untuk mengembangkan aplikasi Android. Android studio pertama kali diumumkan di Google I/O conference pada tanggal 16 Mei 2013. Pada saat itu, Android Studio masih dalam tahap preview, yaitu versi 0.1, dan memasuki tahap beta pada versi 0.8 pada Juni 2014. Versi rilis stabil pertama kali dirilis pada December 2014, yaitu versi 1.0.

Pada instalasi Android Studio terdapat Android SDK. Di mana SDK tersebut diperlukan untuk meng-compile kode Flutter yang ada ke versi Android. Selain itu, pada Android Studio dapat dilakukan instalasi emulator yang digunakan pada saat development agar memudahkan untuk melihat hasil pengembangan tanpa harus menggunakan perangkat Android secara fisik.

2.11 Flutter

Flutter adalah open-source UI software development kit (SDK) yang dibuat oleh Google (Google, 2018a). Versi pertama Flutter dikenal sebagai "Sky" dan berjalan di sistem operasi Android. Versi pertama tersebut diumumkan pada Dart Developer Summit. Pada *keynote* Google Developer Days di Shanghai pada September 2018, Google mengumumkan Flutter *Release Preview 2*, rilis besar terakhir sebelum Flutter 1.0. Pada tanggal 4 Desember 2018, Flutter 1.0 dirilis di acara Flutter Live, yang menunjukkan versi stabil pertama dari Framework Flutter. Flutter digunakan untuk mengembangkan aplikasi pada berbagai platform (multiplatform/*cross platform*) dengan hanya satu *codebase* saja. Hingga saat ini, Flutter telah mendukung pengembangan aplikasi pada platform mulai dari *mobile*, hingga *embedded device*. Pada platform *mobile*, Flutter dapat digunakan untuk mengembangkan aplikasi Android dan iOS. Flutter juga dapat digunakan untuk mengembangkan aplikasi desktop dan web. Pada *embedded device*, Flutter dapat digunakan untuk mengembangkan aplikasi pada *smart device* dan *mobile*.

2.11.1 Dart

Flutter menggunakan bahasa pemrograman Dart sebagai bahasa utama. Dart adalah bahasa pemrograman yang dikembangkan oleh Google (Google, 2011). Dart adalah bahasa pemrograman dikembangkan oleh google sejak tahun 2007 dipimpin oleh Lars Bak dan Kasper Lund. Dart diresmikan pada konferensi GOTO di Aarhus, Denmark pada 10-12 Oktober 2011. Dart 1.0 dirilis pada 14 November 2013. Dart berfokus untuk optimalisasi sisi client. Dart dapat digunakan untuk mengembangkan berbagai macam aplikasi seperti aplikasi *mobile*, web, micro service, desktop, dan aplikasi yang menggunakan teknologi Internet of Things (IOT). Dart merupakan bahasa pemrograman berorientasi objek (OOP) dengan syntax yang hampir sama dengan C++, Java dan Javascript. Oleh karena itu, Dart mendukung pengembangan aplikasi *mobile* secara native untuk dua platform yang berbeda, yaitu Android dan iOS. Dart juga bahasa pemrograman yang dinamis. Dengan Dart VM, kode dapat langsung dijalankan tanpa harus melakukan kompilasi terlebih dahulu.

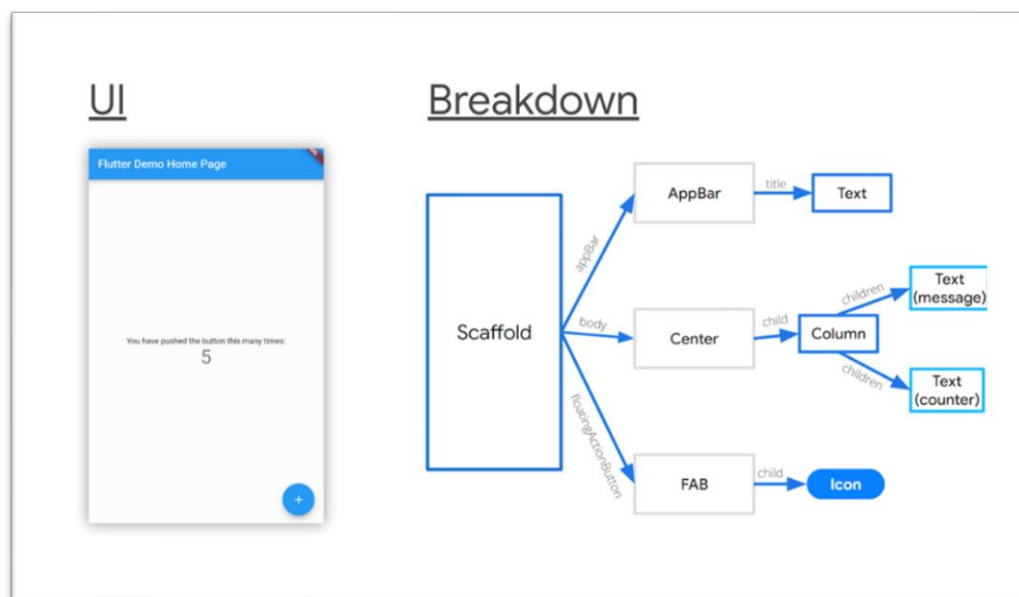
2.11.2 Arsitektur Flutter

Flutter menjalankan sebagian besar *framework* dan kode aplikasi di dalam Dart virtual machine yang ringan. Kode *Framework* Flutter ditulis pada bahasa Dart sedangkan *rendering engine* ditulis pada bahasa C++. Dart *source code* dikompilasi ke *native code* menggunakan fitur kompilasi AoT (*Ahead of Time*) Dart.

Pada Android, kode *engine* C/C++ dikompilasi menggunakan Android NDK, dan seluruh kode Dart dikompilasikan secara AoT ke dalam kode *native*. Pada iOS, kode *engine* C/C++ akan dikompilasi dengan LLVM (Low Level Virtual Machine), dan semua kode Dart dikompilasi secara AoT ke dalam kode *native* (Fayzullaev, 2018).

2.11.3 Flutter *Widget*

Semua hal dalam Flutter adalah *widget*. *Widget* (Google, 2018b) adalah potongan-potongan UI atau komponen yang dapat dikombinasikan untuk membuat sebuah aplikasi. Membuat aplikasi dengan Flutter mirip halnya dengan sebuah lego, menyusun bagian-bagian kecil hingga menjadi suatu susunan aplikasi.



Gambar 2.3 Ilustrasi Flutter *Widget*

Sumber: (Nwabude, 2022)

Pada gambar di atas (pada Gambar 2.3) diperlihatkan breakdown semua *widget* yang ada untuk menampilkan halaman tersebut. Mulai dari scaffold yang berfungsi sebagai *parent* dari semua *widget*, hingga *widget text* yang berfungsi untuk menampilkan teks ke layar.

2.11.4 Flutter Packages

Packages (Google, 2018c) merupakan sebuah paket kode program/fungsi yang dapat digunakan untuk membantu membuat aplikasi secara cepat tanpa membuat program atau fungsi tertentu dari awal. Tiap *package* yang digunakan didaftarkan pada file `pubspec.yaml` pada *section dependencies*. File `pubspec.yaml` mirip seperti `package.json`. File ini berisi metadata mengenai suatu proyek Flutter, *dependencies* atau *package-package*, serta asset-asset yang digunakan oleh proyek tersebut. *Package* yang digunakan oleh aplikasi *mobile* UII *Skin Analyzer* dapat dilihat pada Gambar 2.4 di bawah ini.



Gambar 2.4 Flutter Packages Dependencies UII Skin Analyzer

2.11.5 Framework untuk Mobile Application

Framework yang dapat digunakan untuk membuat aplikasi berbasis *mobile* di antaranya adalah sebagai berikut:

- a. React Native.
- b. Flutter.
- c. Xamarin.
- d. Ionic.
- e. NativeScript.
- f. Kotlin Multiplatform.

Dari 6 *framework* di atas peneliti membandingkan 2 *framework* yaitu Kotlin Multiplatform dan Flutter. Komputer yang digunakan oleh peneliti mengalami masalah seperti *hang*, *lag* dan *crash* sehingga menghambat proses pembuatan aplikasi mobile. Keunggulan penggunaan *framework* Flutter adalah sebagai berikut (Freeborn, 2022) (Ashraf, 2022):

- a. Performa tinggi: Flutter menggunakan mesin *rendering* sendiri yang disebut Flutter *Engine*. Dibangun dengan menggunakan bahasa pemrograman Dart, Flutter memungkinkan aplikasi untuk menjalankan kode langsung pada level rendah menghasilkan performa yang tinggi dan responsif.
- b. Antarmuka pengguna yang konsisten: Flutter menyediakan kumpulan *widget* yang kaya dan sepenuhnya dapat disesuaikan (*customizable*). Pengembang dapat membuat antarmuka pengguna yang menarik dan responsif secara konsisten di berbagai platform seperti iOS, Android, dan desktop.
- c. Pengembangan lintas platform: Dengan menggunakan satu basis kode, Flutter memungkinkan pengembang untuk membuat aplikasi *mobile* lintas platform. Hal ini menghemat waktu dan upaya pengembangan karena pengembang tidak perlu menulis ulang kode untuk setiap platform yang berbeda.
- d. *Hot reload*: Fitur *Hot reload* di Flutter memungkinkan pengembang untuk melihat perubahan yang diterapkan secara instan pada aplikasi bahkan saat aplikasi sedang berjalan. Hal ini mempercepat proses pengembangan dan memungkinkan eksperimen dan iterasi yang cepat.
- e. Komunitas dan ekosistem yang kuat: Flutter memiliki komunitas yang aktif dan dukungan yang luas dari Google. Komunitas yang kuat ini berarti tersedia banyak sumber daya, tutorial, dan paket ekstensi (*packages*) yang dapat digunakan oleh pengembang untuk mempercepat proses pengembangan.
- f. Animasi yang kaya: Flutter menyediakan *library* animasi yang kuat sehingga pengembang dapat membuat animasi yang kompleks dan menarik dengan mudah. Animasi yang mulus (*fluid/seamless*) dan interaktif dapat memberikan pengalaman pengguna yang lebih baik.
- g. Pembuatan antarmuka kustom: Flutter memungkinkan pengembang untuk membuat antarmuka pengguna yang kustom dengan mudah. Fleksibilitas yang tinggi dalam desain antarmuka pengembang dapat mengimplementasikan desain yang kompleks dan kreatif.

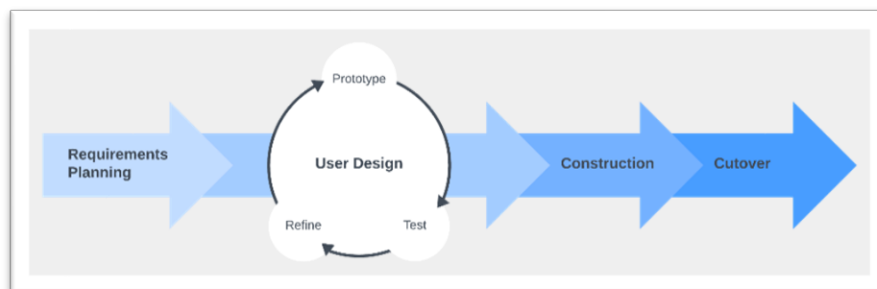
2.12 SQLite

SQLite adalah *database engine* yang ditulis dalam bahasa pemrograman C (SQLite Consortium, 2000). SQLite bukan sebuah aplikasi yang *standalone*, melainkan sebuah *library*. Dengan demikian, SQLite termasuk ke dalam keluarga *embedded databases*. SQLite adalah *database engine* yang paling banyak digunakan, dengan SQLite digunakan oleh beberapa web browser populer, sistem operasi, *smartphone*, dan *embedded device* lainnya. SQLite pada dasarnya adalah sistem manajemen basis data relasional (RDBMS), seperti halnya SQL.

Pada Flutter, tersedia *package* bernama *sqflite*, yang membuat aplikasi Flutter dapat menggunakan *database* SQLite (Tekartik, 2018). *Package* ini telah mendukung sistem operasi Android, iOS, dan MacOS. SQLite digunakan pada aplikasi *mobile* UI *Skin Analyzer* dikarenakan masalah privasi dan keamanan. Dengan SQLite, data-data pengguna yang relatif *private* seperti gambar wajah pengguna, akan aman tersimpan pada perangkat pengguna. Selain itu, *local storage* memiliki kecepatan akses yang lebih andal dibanding *cloud storage*.

2.13 Rapid Application Development

Rapid Application Development (RAD) adalah strategi manajemen proyek *agile* yang populer dalam pengembangan perangkat lunak. Manfaat utama dari pendekatan RAD adalah perputaran proyek yang cepat, menjadikannya pilihan yang menarik bagi pengembang yang bekerja di lingkungan yang bergerak cepat. Perputaran proyek yang cepat ini mungkin dilakukan karena fokus RAD ada pada meminimalkan tahap perencanaan dan memaksimalkan pengembangan prototipe. Secara umum, terdapat empat fase pengembangan di dalam metode RAD, seperti terlihat pada Gambar 2.5 berikut:



Gambar 2.5 Fase pengembangan pada metode *Rapid Application Development*

2.14 Review Aplikasi Sejenis

Terdapat beberapa aplikasi sejenis dengan yang coba dibuat pada penelitian ini. Aplikasi-aplikasi tersebut umumnya dibuat oleh perusahaan kosmetika, sehingga aplikasi tersebut cenderung digunakan sebagai alat promosi produk. Berikut beberapa aplikasi tersebut.

2.14.1 Face Mapping Dermalogica

Dermalogica adalah perusahaan perawatan pribadi Amerika yang berkantor pusat di Carson, California. Produk Dermalogica meliputi pembersih, pengelupasan kulit, toner, masker, perawatan mata, dan pelembab, serta rangkaian perawatan jerawat untuk remaja. Dermalogica beroperasi di Amerika Serikat, Inggris, Kanada, Australia, India, Pakistan dan Irlandia, serta telah dijual di lebih dari 80 negara di seluruh dunia.

Face Mapping Analysis dari Dermalogica menggunakan *machine learning* untuk menunjukkan, mengukur, dan merawat masalah-masalah yang biasa muncul pada kulit hanya dengan mengambil dan mengirimkan foto ke sistem (Dermalogica, 2015).

2.14.2 Olay Skin Advisor

Olay berasal dari Afrika Selatan dengan nama sebelumnya yaitu "Oil of Olay". Graham Wulff, mantan ahli kimia Unilever dari Durban, memulai Olay pada tahun 1952. Ia memilih nama "Oil of Olay" sebagai pengganti kata "lanolin", bahan utama dari produknya. Produk tersebut unik pada masa itu dikarenakan wujudnya yang berupa cairan merah muda, dan dikemas dalam botol kaca yang berat. Wulff dan mitra pemasarannya, Jack Lowe, mantan copywriter, telah menguji produk itu pada istri dan teman-teman mereka dan yakin akan keunikan dan kualitasnya.

Olay Skin Advisor adalah platform analisis kulit berbasis web yang memberikan analisis kulit yang tepat dan dipersonalisasi serta rekomendasi produk sesuai dengan hasil analisis. Olay Skin Advisor memberikan rekomendasi rejimen produk yang dipersonalisasi tentang produk yang paling cocok untuk jenis kulit hasil analisis (Olay, 2017).

2.14.3 L'Oréal Paris Skin Genius

L'Oréal adalah perusahaan perawatan pribadi Prancis yang berkantor pusat di Clichy, Hauts-de-Seine dengan kantor utama di Paris. L'Oréal adalah perusahaan kosmetik terbesar di dunia dan telah mengembangkan produk di bidang yang berkonsentrasi pada warna rambut, perawatan kulit, perlindungan matahari, make-up, parfum, dan perawatan rambut.

L'Oréal Paris Skin Genius menyediakan analisis perawatan kulit seperti dari ahli, serta memberikan rutinitas skin care yang sesuai dengan kebutuhan kulit. Didukung dengan teknologi AI yang dikembangkan oleh L'Oréal Research berdasarkan data klinis selama 30 tahun dan didukung oleh publikasi ilmiah dermatologis, hasil analisis Skin Genius memiliki tingkat akurasi hingga 95% jika dibandingkan dengan evaluasi dokter kulit. Setelah mendapatkan analisis, temukan rekomendasi rutinitas skin care yang disesuaikan dengan jenis dan kebutuhan kulit, divalidasi oleh *International Board of Dermatologists of L'Oréal Paris* (L'Oréal Paris, 2020).

Tabel 2.1 Perbandingan aplikasi sejenis UII *Skin Analyzer*

Nama Aplikasi	Platform	Proses Analisis	Hasil Analisis	Hal yang Dianalisis
Face Mapping Dermalogica	Web-based, hanya dapat digunakan pada desktop	Memerlukan pengguna menjawab pertanyaan; Memerlukan koneksi internet yang andal; Posisi gambar yang harus tepat	Diberikan hasil analisis beserta rekomendasi produk dari Dermalogica	Tidak ada deteksi jerawat
Olay Skin Advisor	Web-based, hanya dapat digunakan pada desktop	Memerlukan pengguna menjawab pertanyaan; Pengambilan gambar tidak disertai dengan alat bantu berupa outline wajah	Hanya rekomendasi produk dari Olay	Tidak menganalisis langsung kondisi kulit wajah
L'Oréal Paris Skin Genius	Web-based, hanya dapat digunakan pada <i>mobile</i>	Memerlukan pengguna menjawab pertanyaan;	Diberikan hasil analisis beserta rekomendasi produk dari L'Oréal Paris	Tidak ada deteksi jerawat, kemerahan, dan jenis kulit

BAB III

METODOLOGI PENELITIAN

3.1 *Rapid Application Development (RAD)*

Rapid Application Development adalah salah satu metode pengembangan yang mengarah pada filosofi *agile*. Prinsip utama RAD adalah keterlibatan secara erat dengan pengguna akhir serta pembuatan prototipe secara iteratif dan cepat berdasarkan umpan balik dari pengguna. Setelah pengguna telah mencapai *satisfaction*, fokus dari RAD adalah berpindah ke pengiriman perangkat lunak yang siap untuk dilakukan produksi.

Rapid Application Development adalah metode pengembangan perangkat lunak dengan waktu yang relatif singkat. Pada metode RAD, dilakukan pengembangan perangkat lunak atau dalam penelitian ini adalah aplikasi *mobile* dengan melakukan pembuatan prototipe, penerimaan *feedback* dari pengguna, pengembangan kembali prototipe, menerima kembali *feedback* dari pengguna, dan begitu seterusnya, sehingga aplikasi akan sesuai dengan apa yang diharapkan oleh pengguna. Terdapat 4 pada metode RAD, yaitu fase perencanaan, fase perancangan, fase implementasi, dan fase *finishing*.

Meskipun *Rapid Application Development* merupakan konsep yang sudah lama namun terus mengalami kebangkitan dengan tren transformasi digital dan *time-to-market* yang lebih cepat. Dengan jenis proyek dan tim yang tepat, metodologi RAD sangat membantu mencapai kepuasan pengguna yang lebih baik dengan risiko dan waktu yang lebih singkat.

3.2 Fase Perencanaan

Fase perencanaan dapat dikatakan sebagai fase *project scoping*. Fase perencanaan pada metode RAD dilakukan dengan lebih ringkas dibandingkan pada metodologi pengembangan perangkat lunak lainnya, akan tetapi tetap saja, fase ini adalah langkah yang penting dalam menentukan kesuksesan dari keseluruhan proses pengembangan perangkat lunak.

Pada fase ini dilakukan analisis terkait masalah yang terjadi saat ini, yaitu dengan semakin maraknya permasalahan kulit wajah seperti jerawat dan keriput, serta semakin meningkatnya keinginan untuk memiliki kulit yang sehat. Pada fase ini juga dilakukan *review* aplikasi sejenis. Selain itu, pada fase ini dilakukan juga analisis terkait kebutuhan untuk mengembangkan aplikasi untuk menyelesaikan permasalahan tersebut.

3.2.1 Permasalahan Saat ini

Saat ini, betapa familiarnya masyarakat dengan *skin care*. Mulai dari anak-anak, orang dewasa, hingga orang tua sekalipun memiliki keinginan yang kuat untuk merawat kulitnya. Selain itu, bukan hanya wanita saja yang saat ini melakukan perawatan kulit, pria pun banyak yang melakukannya. Perawatan kulit dapat dilakukan mulai dari menjalani perilaku hidup bersih dan sehat (PHBS), melakukan perawatan yang dibantu dengan berbagai produk kecantikan, hingga melakukan perawatan di klinik kecantikan yang dibantu langsung oleh dokter kecantikan. Berbagai komplikasi akan kesalahan penggunaan produk kecantikan sudah banyak terjadi, sehingga untuk melakukan perawatan kulit melalui produk kecantikan, diperlukan adanya pengetahuan akan jenis kulit dan produk kecantikan mana yang sesuai. Untuk itu, dibuatlah aplikasi UII *Skin Analyzer* yang bertujuan untuk membantu pengguna dalam memahami kondisi kulitnya.

3.2.2 Review Aplikasi Sejenis

Hingga saat ini sudah terdapat beberapa aplikasi yang memiliki tujuan untuk menyelesaikan permasalahan yang ada. Aplikasi-aplikasi ini secara khusus dibuat oleh perusahaan produk kecantikan yang sudah dikenal di dunia. Aplikasi-aplikasi tersebut yaitu Facemapping oleh Dermalogica, Skin Advisor oleh Olay, Skin Genius oleh L'Oréal Paris. Setelah melakukan *review* terhadap aplikasi-aplikasi tersebut didapatkan kesimpulan bahwa yang coba dilakukan atau dieksekusi oleh ketiga aplikasi tersebut yaitu memberikan panduan bagi calon pembeli terhadap produk yang mereka jual, dikarenakan hasil analisis disertai dengan rekomendasi produk-produk yang sesuai dengan kondisi kulit sesuai dengan gambar yang telah dianalisis.

3.2.3 Analisis Kebutuhan

Setelah melakukan diskusi dengan *stakeholder/project owner*, maka didapatkan hasil analisis kebutuhan sebagai berikut:

- a. Aplikasi berbasis *mobile*.
- b. Aplikasi harus menghormati privasi pengguna.
- c. Aplikasi harus mudah digunakan.
- d. Pengguna dapat menyimpan hasil analisis.
- e. Pengguna dapat melihat riwayat hasil analisis.

- f. Aplikasi dapat melakukan analisis permasalahan kulit berupa jerawat.
- g. Aplikasi dapat melakukan analisis permasalahan kulit berupa keriput.
- h. Aplikasi dapat melakukan analisis permasalahan kulit berupa kemerahan.
- i. Aplikasi dapat melakukan analisis permasalahan kulit berupa bercak hitam.
- j. Aplikasi dapat melakukan analisis jenis kulit wajah pengguna.

Berdasarkan poin-poin terkait hasil dari wawancara soal kebutuhan-kebutuhan apa saja yang kira-kira diperlukan oleh pengguna, maka didapatkan kebutuhan fungsional dari aplikasi ini adalah seperti Tabel 3.1 berikut:

Tabel 3.1 Kebutuhan fungsional

No	Kebutuhan Fungsional
1	Sistem dapat memasukkan gambar dari galeri
2	Sistem dapat mengambil gambar dari kamera
3	Sistem mampu melakukan analisis gambar
4	Sistem dapat menyimpan hasil analisis ke aplikasi <i>UII Skin Analyzer</i>
5	Sistem dapat menyimpan hasil analisis ke galeri
6	Sistem dapat menghapus hasil analisis
7	Sistem dapat mengakses riwayat yang tersimpan tanpa koneksi internet
8	Sistem dapat melakukan pendaftaran dan penggantian akun

3.3 Fase Perancangan

Pada fase ini, akan dilakukan perancangan yang melibatkan calon pengguna secara langsung sehingga dinamakan *user design*. Nantinya akan dibuat prototipe, di mana prototipe ini akan ditunjukkan ke calon pengguna dan calon pengguna memberikan *feedback*. Lalu, berdasarkan *feedback* tersebut maka dimulai lagi proses perancangan prototipe dan begitu seterusnya. Dilakukan iterasi atau kegiatan yang berulang antara perancangan prototipe dengan pengujian oleh calon pengguna. Fase ini dapat dikatakan sebagai keunikan dari metode RAD.

Pada fase ini, dibuat beberapa rancangan aplikasi yang berbentuk BPMN atau *Business Process Model Notation*, diagram aktivitas, *use case diagram*, dan pembuatan *wireframe* pada iterasi ke-2 dari aplikasi. Berbagai rancangan aplikasi tersebut dibuat untuk memudahkan calon pengguna dalam memahami aplikasi yang nantinya akan dibuat. Pada pengembangan aplikasi

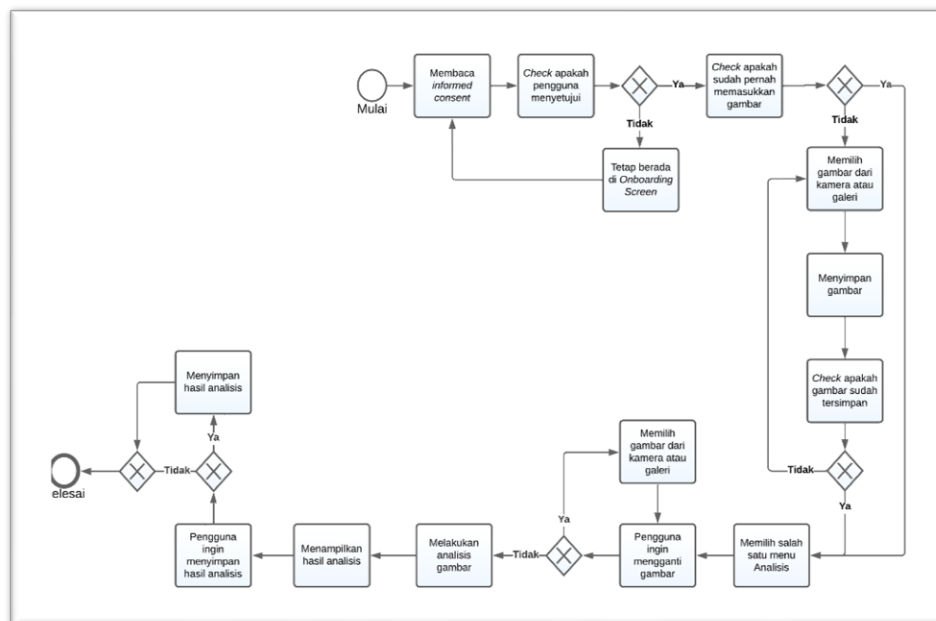
mobile UII *Skin Analyzer*, terdapat dua kali iterasi, dengan melibatkan 30 orang calon untuk memberikan *feedback* terkait rancangan yang telah dibuat.

3.3.1 Iterasi Ke-1

Pada iterasi ke-1, dibuat rancangan dari keseluruhan aplikasi *mobile* UII *Skin Analyzer*. Rancangan ini mulai dari awal pengguna membuka aplikasi, hingga pada tombol-tombol seperti tombol bantuan. Rancangan dibuat selengkap mungkin agar calon pengguna yang memberikan *feedback* dapat langsung merasakan menggunakan aplikasi walaupun masih dalam berbentuk prototipe.

Business Process Model Notation (BPMN)

BPMN adalah metode diagram alur yang memungkinkan untuk membuat diagram yang mudah untuk dipahami. BPMN dapat menjelaskan langkah proses bisnis secara menyeluruh. BPMN dipilih agar dapat memvisualisasikan bagaimana rancangan langkah-langkah dari aplikasi yang nantinya akan dibuat dan dapat dimengerti dengan mudah oleh calon pengguna. Pada Gambar 3.1 ditampilkan diagram BPMN awal dari aplikasi *mobile* UII *Skin Analyzer*. Diagram ini nantinya dapat berubah jika mendapat *feedback* yang buruk dari calon pengguna.



Gambar 3.1 Diagram BPMN (iterasi ke-1)

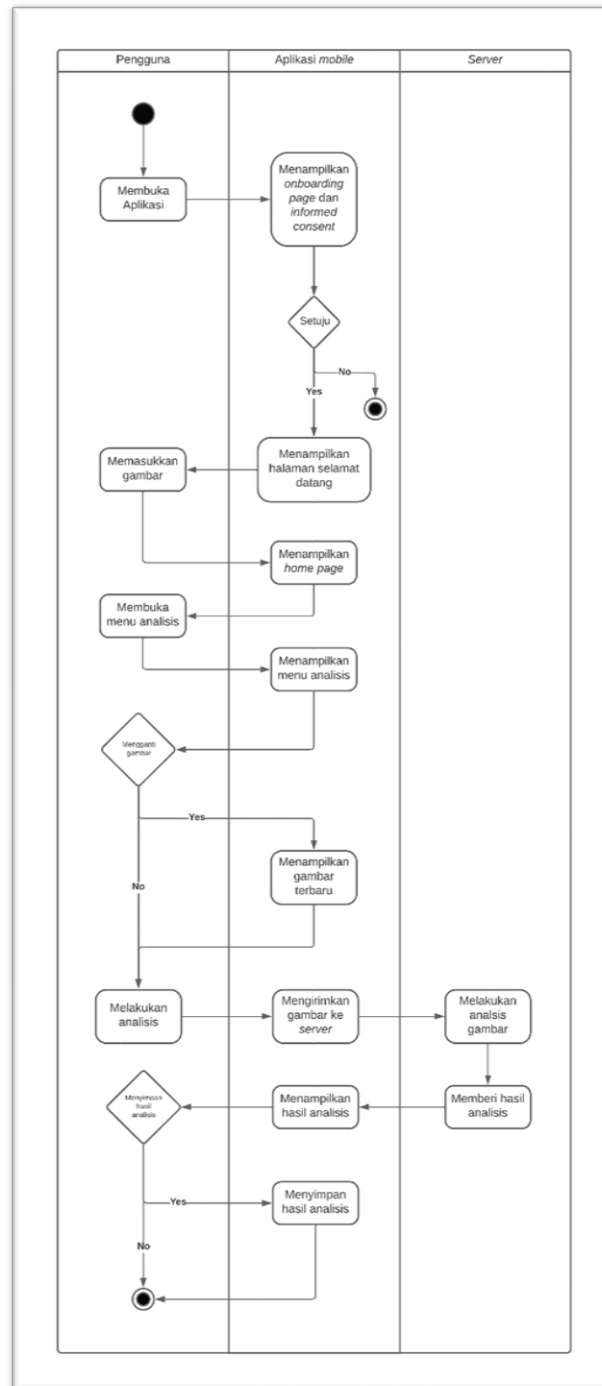
Berdasarkan gambar di atas (perhatikan Gambar 3.1) langkah atau alur yang dilakukan oleh pengguna dimulai dengan pengguna membaca *informed consent*, jika pengguna tidak menyetujui maka akan tetap berada di halaman *onboarding screen*. Sebaliknya jika pengguna menyetujui maka akan diarahkan ke halaman berikutnya yaitu mengecek apakah pengguna sudah pernah memasukkan gambar atau tidak. Jika sudah pernah memasukkan gambar maka pengguna menuju halaman utama dan dapat memilih salah satu menu analisis. Jika tidak maka pengguna harus memilih gambar dari kamera atau galeri, menyimpan gambar, dan mengecek apakah gambar sudah tersimpan atau belum. Jika belum pengguna akan kembali ke halaman memilih gambar. Pengguna yang pertama kali menggunakan aplikasi ini hanya dapat melakukan analisis terhadap gambar yang sudah berhasil disimpan atau dimasukkan. Selanjutnya setelah memilih salah satu menu analisis pengguna masih diberikan kesempatan untuk mengubah gambar yang sudah disimpan pertama kali. Setelah gambar sudah final maka pengguna dapat melakukan analisis terhadap gambar tersebut. Saat proses analisis sudah selesai, akan tertampil hasil analisis berdasarkan citra atau gambar yang telah dipilih. Pengguna dapat memilih menyimpan hasil analisis atau tidak untuk dilihat pada lain waktu. Alur penggunaan aplikasi UII *Skin Analyzer* selesai.

Diagram Aktivitas

Selain BPMN, dibuat juga diagram aktivitas. Diagram aktivitas dibuat untuk lebih menjelaskan akan alur aplikasi yang nantinya dibuat. Diagram aktivitas ini nantinya juga akan diperlihatkan kepada calon pengguna dan jika mendapatkan *feedback* yang buruk, diagram ini juga akan dapat berubah.

Berdasarkan gambar di bawah (perhatikan Gambar 3.2), diagram aktivitas aplikasi *mobile* UII *Skin Analyzer* memiliki 3 *swimlanes*, yakni Pengguna, Aplikasi *Mobile*, dan Server. Aktivitas dimulai oleh pengguna yang membuka aplikasi, lalu aplikasi akan menampilkan halaman *onboarding* dan *pop-up informed consent*. Jika pengguna menekan ‘Tidak Setuju’ maka aktivitas akan selesai, dan jika pengguna menekan tombol ‘Setuju’ maka aplikasi akan menampilkan halaman selamat datang. Setelah itu, pengguna memasukkan gambar di mana gambar ini nantinya akan tersimpan sampai pengguna mengganti gambar tersebut. Hal ini ditujukan untuk memudahkan pengguna jika ingin melakukan analisis pada gambar yang sama. Setelah memasukkan gambar, akan ditampilkan halaman beranda atau *home*, di mana saat pengguna menekan tombol kondisi kulit yang ingin dianalisis maka aplikasi akan membuka menu analisis. Dikarenakan pengguna telah memasukkan gambar, maka pengguna dapat

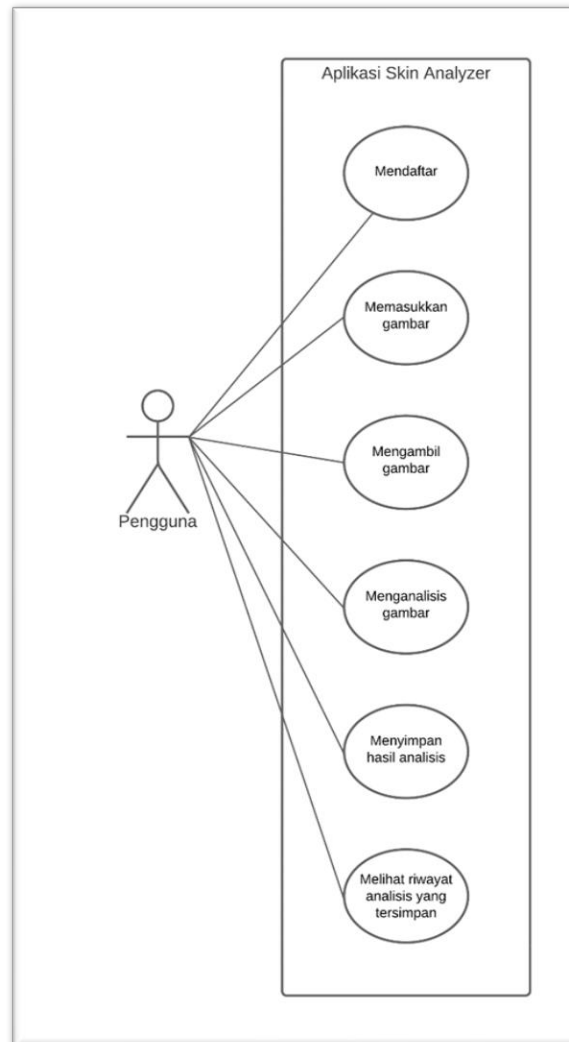
langsung melakukan analisis dengan menekan tombol ‘Analisis’. Aplikasi akan mengirimkan gambar tersebut ke *server*, *server* melakukan analisis pada gambar, *server* mengirimkan hasil analisis, dan aplikasi menampilkan hasil analisis. Pengguna dapat menyimpan hasil hasil analisis tersebut.



Gambar 3.2 Diagram Aktivitas (iterasi ke-1)

Use Case Diagram

Selain BPMN dan diagram aktivitas, juga dibuat *use case diagram*. *Use case diagram* ini dibuat untuk menunjukkan apa-apa saja yang dapat dilakukan oleh pengguna pada aplikasi UII *Skin Analyzer*.



Gambar 3.3 *Use Case Diagram* (iterasi ke-1)

Pada *use case diagram* di atas (Gambar 3.3), ditampilkan apa-apa saja yang dapat dilakukan oleh pengguna aplikasi *mobile UII Skin Analyzer*. Pengguna dapat mendaftar, memasukkan gambar, mengambil gambar dari kamera, menganalisis gambar, menyimpan hasil analisis, dan melihat hasil analisis yang tersimpan.

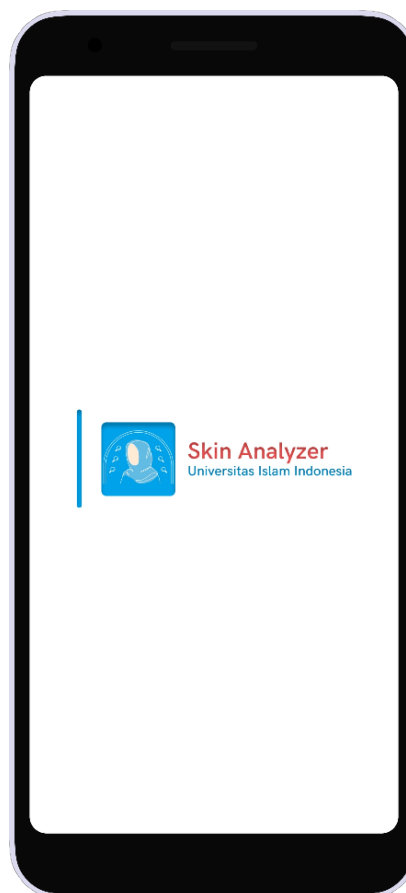
Desain Aplikasi

Setelah membuat diagram BPMN, diagram aktivitas, dan *use case* diagram terkait aplikasi UII *Skin Analyzer* ini tahap selanjutnya adalah membuat desain aplikasi. Desain aplikasi ini sudah menggunakan *high-fidelity* agar tampilan, gaya visual, dan perilaku interaktif yang dihasilkan dapat semirip mungkin dengan produk final.

Berikut adalah desain pertama aplikasi *mobile* UII *Skin Analyzer* berdasarkan kebutuhan fungsional yang ada serta diagram-diagram yang telah dibuat sebelumnya:

a. Halaman Logo (*Splash Screen*)

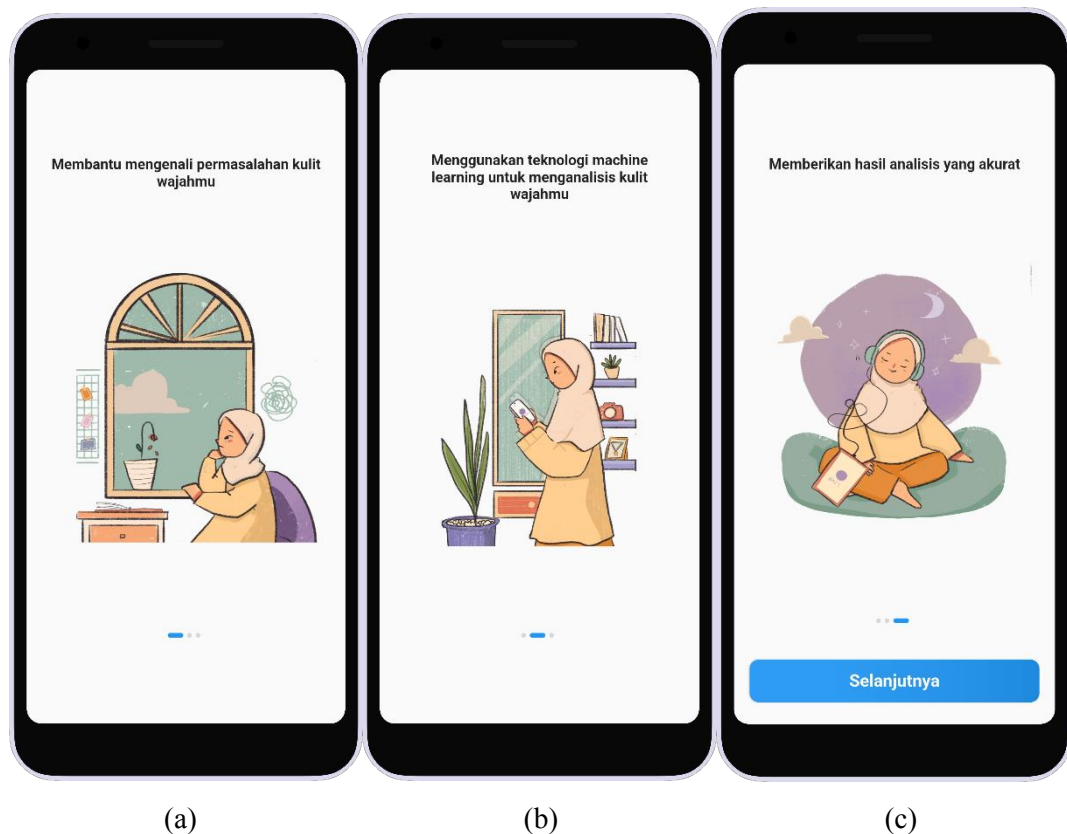
Halaman logo akan ditampilkan saat pertama membuka aplikasi. Seperti pada Gambar 3.4 halaman *splash screen* ini terdiri dari logo di sebelah kiri dan teks di sebelah kanan yang merupakan nama aplikasi tersebut yaitu “*Skin Analyzer*” dan “Universitas Islam Indonesia”. Logo akan ditampilkan dengan animasi *appear* dari kanan ke kiri.



Gambar 3.4 Halaman Logo (iterasi ke-1)

b. Halaman *Onboarding*

Halaman *onboarding* (pada Gambar 3.5), akan menampilkan tiga halaman yang berisi pengenalan akan aplikasi UII *Skin Analyzer* berupa satu kalimat singkat dan ilustrasi pada tiap halamannya. Pada tampilan *onboarding* yang pertama (pada 3.5 bagian a), berisi kalimat singkat yang merupakan tujuan dari aplikasi UII *Skin Analyzer*. Selanjutnya pada tampilan *onboarding* yang kedua (pada Gambar 3.5 bagian b), berisi jenis teknologi yang digunakan pada aplikasi UII *Skin Analyzer*. Terakhir, pada tampilan *onboarding* yang ketiga (pada Gambar 3.5 bagian c), berisi kalimat yang meyakinkan pengguna bahwa hasil analisis yang diberikan adalah hasil yang akurat.

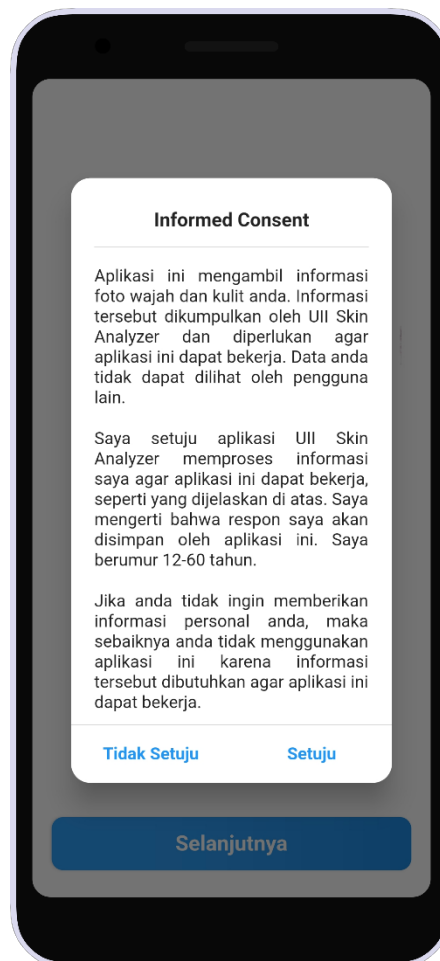


Gambar 3.5 Halaman *Onboarding* (iterasi ke-1)

c. *Pop-up Informed Consent*

Pop-up Informed Consent (perhatikan Gambar 3.6), akan muncul saat pengguna menekan tombol selanjutnya pada halaman *onboarding* yang terakhir. *Informed Consent* ini berisi persetujuan akan apa saja yang pengguna harus lakukan untuk dapat menggunakan aplikasi terutama mengirimkan gambar wajahnya ke server *machine learning* aplikasi UII *Skin Analyzer*. Pada *pop-up* ini juga diberikan pernyataan bahwa gambar yang pengguna

unggah hanya akan digunakan untuk keperluan aplikasi *machine learning* untuk terus belajar atau sebagai data *training* agar aplikasi dapat menganalisis gambar dengan lebih akurat. Pengguna dapat memilih untuk menekan tombol “Setuju” atau “Tidak Setuju”. Jika menekan tombol “Tidak Setuju” pengguna tidak bisa menggunakan aplikasi dan kembali ke halaman *onboarding*. Sebaliknya, jika menekan tombol “Setuju” pengguna dapat menggunakan aplikasi UII *Skin Analyzer* dan diarahkan ke halaman berikutnya.

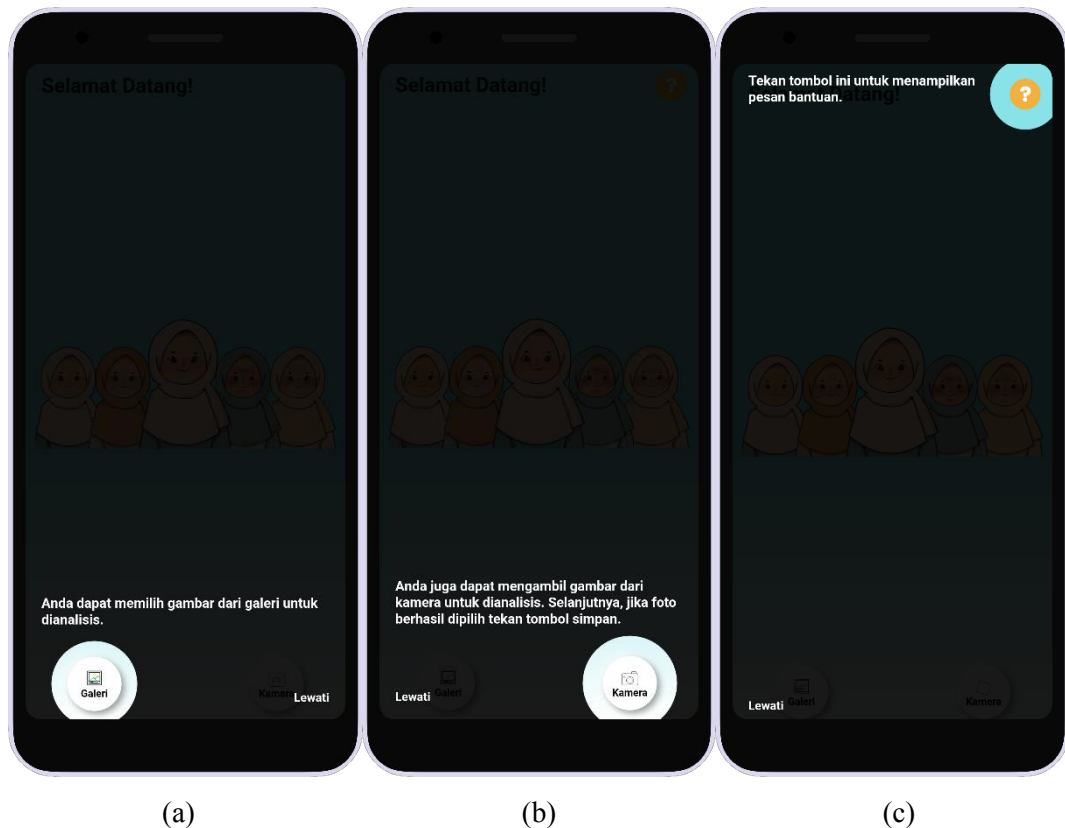


Gambar 3.6 *Pop-up Informed Consent* (iterasi ke-1)

d. Halaman Tutorial

Halaman tutorial (pada Gambar 3.7), berisi tutorial terkait tombol-tombol yang kerap muncul pada aplikasi UII *Skin Analyzer* seperti mengambil gambar dari galeri, kamera, dan tombol bantuan. Halaman ini akan muncul saat pengguna pertama kali menggunakan aplikasi dan setelah menyetujui *informed consent*. Tutorial akan ditampilkan dengan menggelapkan layar dan meng-*highlight* tombol yang dijelaskan. Tutorial yang pertama

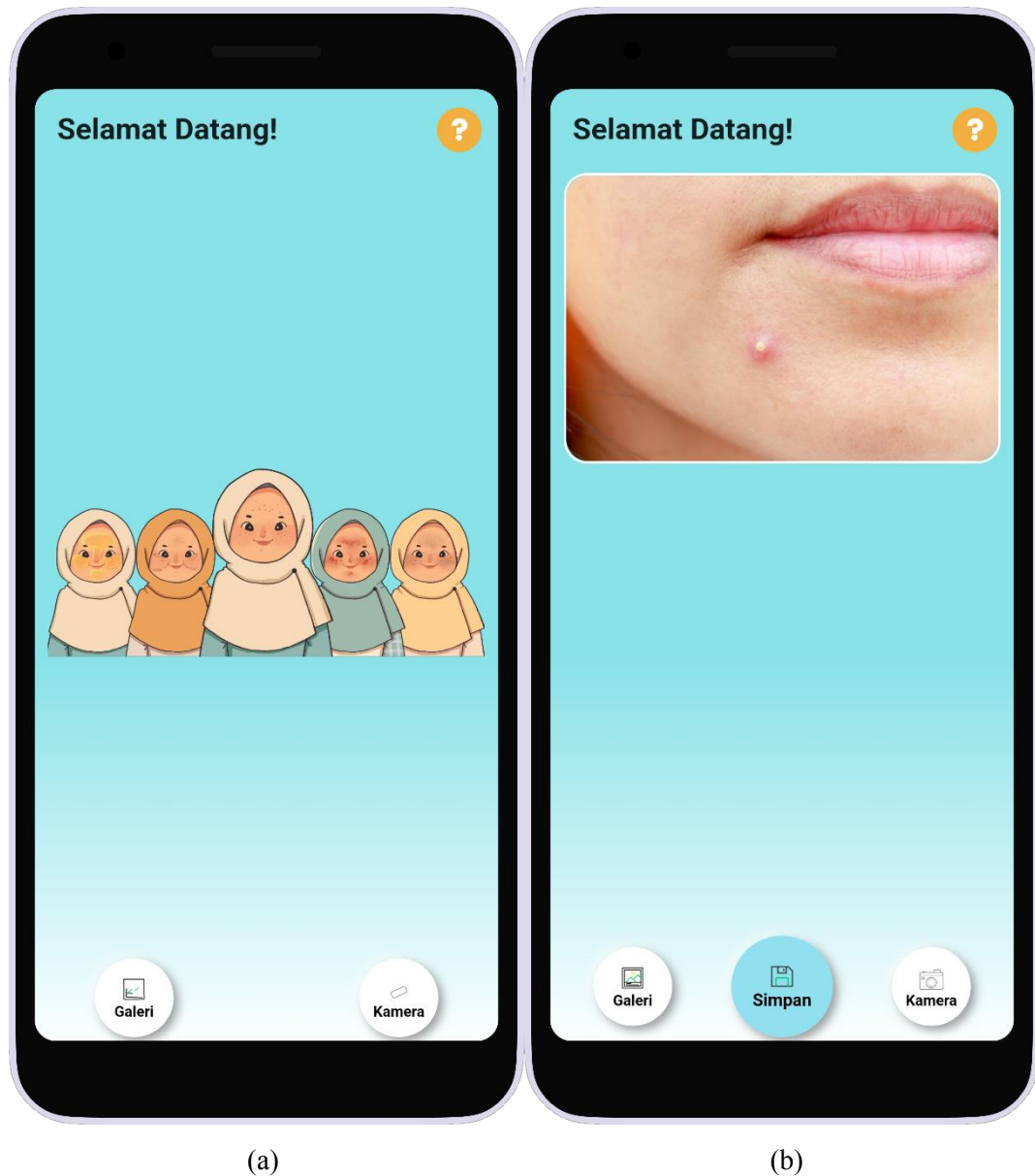
(a) adalah tombol ambil gambar dari galeri. Tutorial yang kedua (b) adalah tombol ambil gambar dari kamera. Tutorial yang ketiga (c) adalah tombol bantuan. Pengguna dapat mengikuti ketiga langkah tersebut atau dapat langsung menekan tombol “Lewati” untuk melewati tutorial tersebut.



Gambar 3.7 Halaman Tutorial (iterasi ke-1)

e. Halaman Selamat Datang

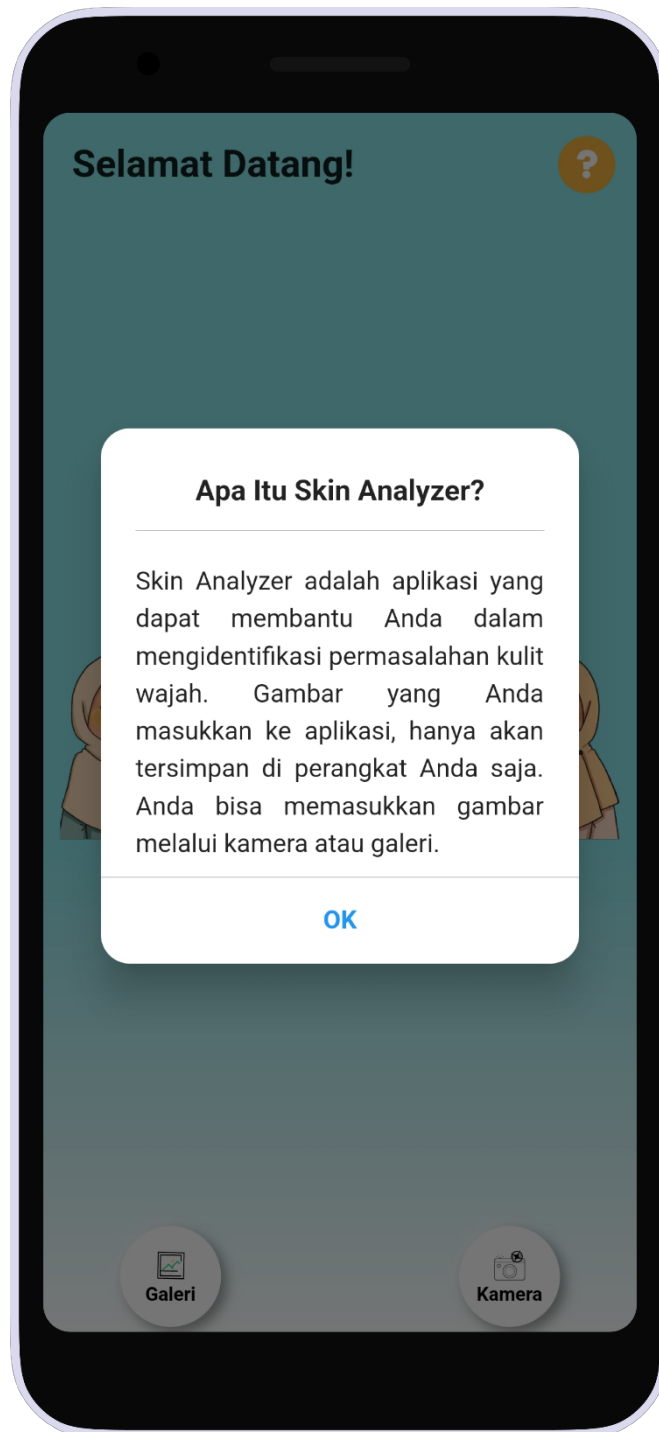
Halaman selamat datang (lihat Gambar 3.8), berisi halaman untuk mengambil gambar pertama dari pengguna. Halaman ini ditampilkan setelah tutorial selesai atau pengguna melewati tutorial yang ada. Pada gambar (a) pengguna dapat mengunggah gambar dari galeri maupun mengambil langsung dari kamera. Setelah pengguna memasukkan gambar akan muncul tombol simpan untuk menyimpan gambar pada aplikasi (b). Gambar hanya tersimpan pada aplikasi *mobile* atau pada perangkat pengguna saja. Gambar yang tersimpan nantinya dapat langsung dianalisis tanpa pengguna harus memasukkan kembali gambar kecuali pengguna ingin mengganti gambar yang ingin dianalisis. Halaman ini hanya akan muncul jika pengguna belum menyimpan gambar.



Gambar 3.8 Halaman Selamat Datang (iterasi ke-1)

f. Halaman *Pop-up* Tombol Bantuan Halaman Selamat Datang

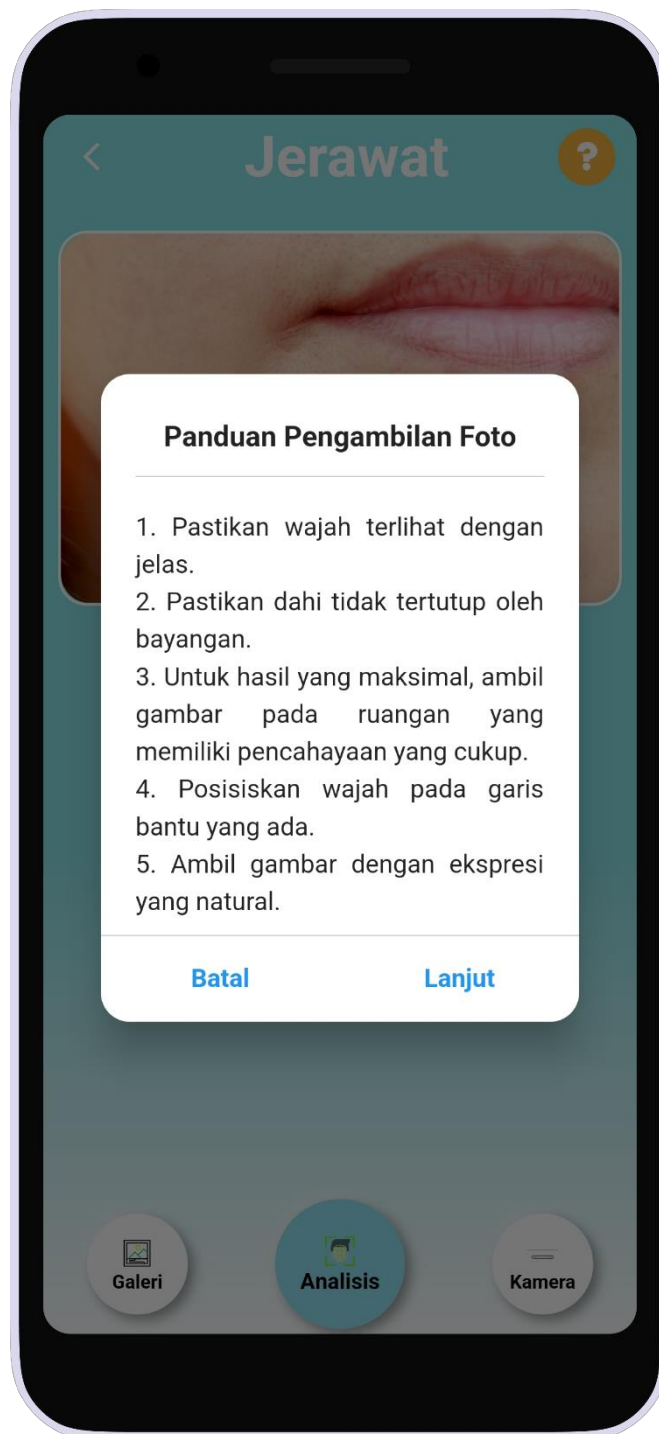
Halaman *Pop-up* Tombol Bantuan Halaman Selamat Datang (perhatikan Gambar 3.9), ini berisi penjelasan singkat terkait “Apa itu UUI *Skin Analyzer*?”. Pada halaman ini pengguna hanya diberikan satu pilihan tombol yaitu tombol “OK”.



Gambar 3.9 Halaman *Pop-up* Tombol Bantuan Halaman Selamat Datang (iterasi ke-1)

g. Halaman *Pop-up* Tombol Panduan Pengambilan Foto

Halaman *Pop-up* Tombol Panduan Pengambilan Foto (perhatikan Gambar 3.10), ini berisi penjelasan singkat terkait bagaimana cara untuk mengambil foto yang benar sehingga mendapatkan hasil analisis yang maksimal. Pengguna dapat memilih tombol “Batal” atau “Lanjut” pada halaman ini.

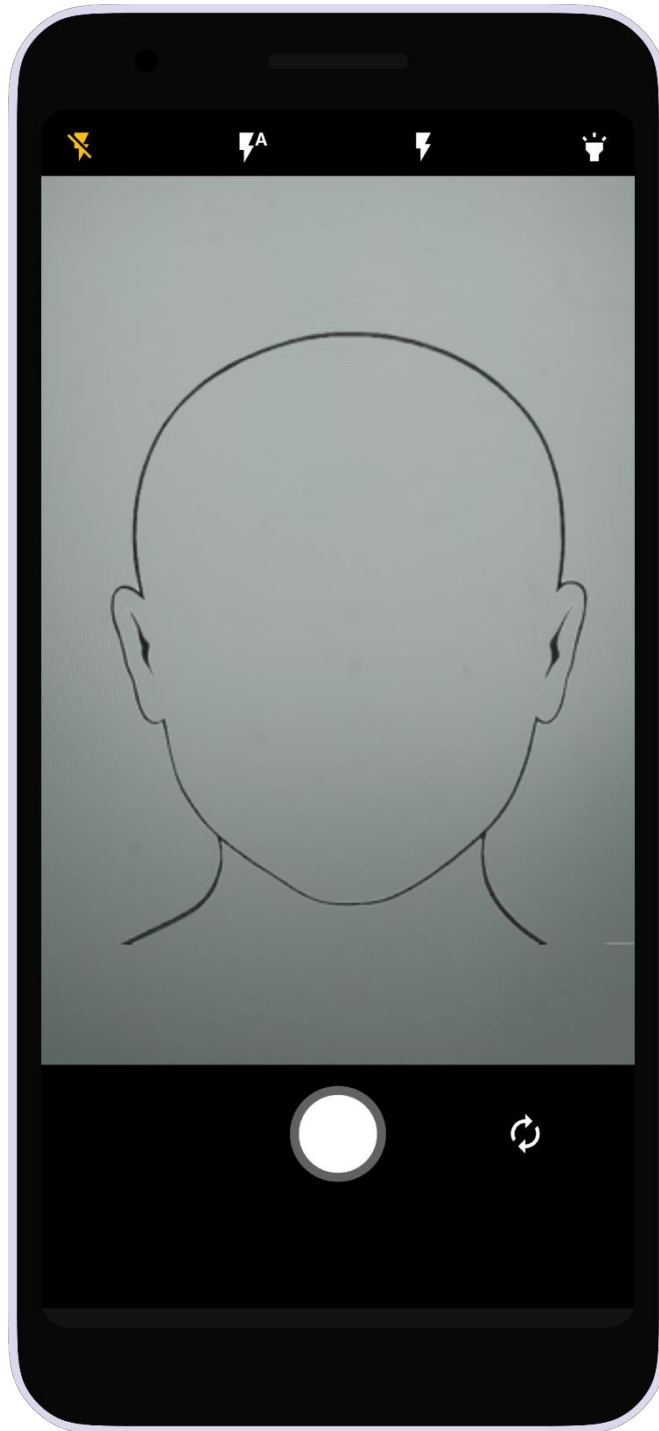


Gambar 3.10 Halaman *pop-up* tombol panduan pengambilan foto (iterasi ke-1)

h. Halaman Kamera

Pada halaman kamera (pada Gambar 3.11), akan ditampilkan suatu *overlay* dengan bentuk wajah menghadap ke depan yang bertujuan untuk memudahkan pengguna saat

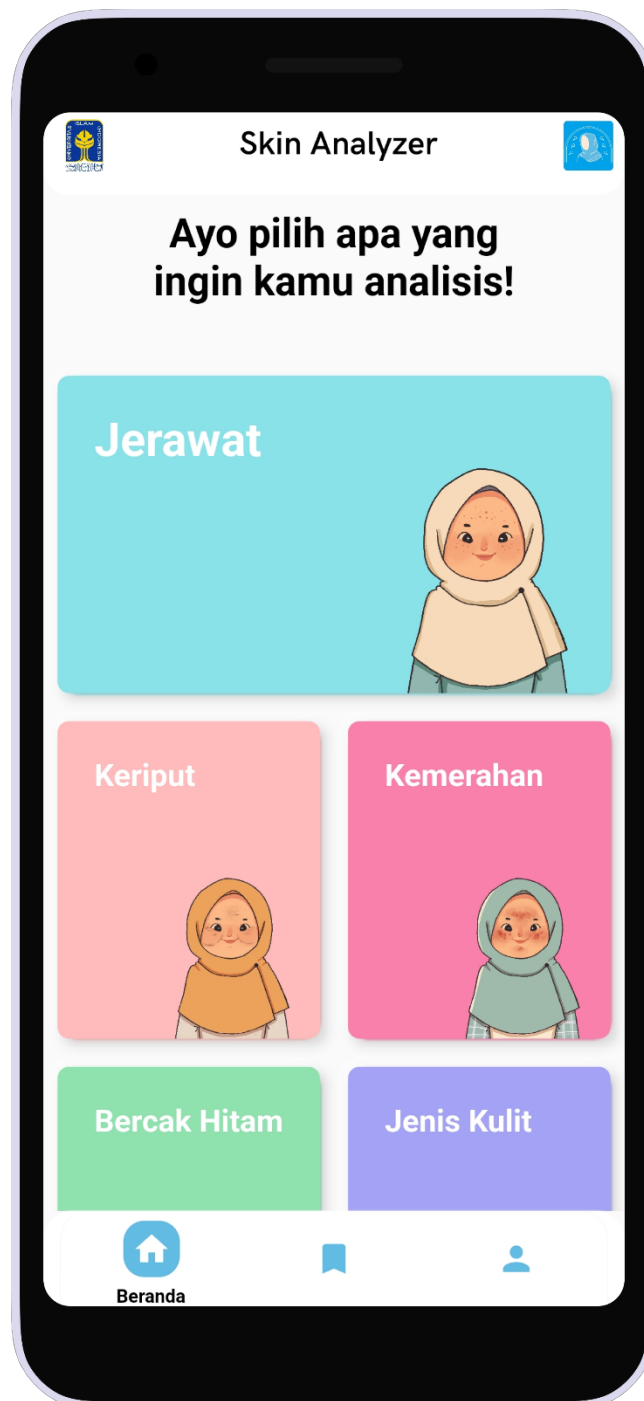
pengambilan gambar dengan posisi yang lebih baik dan sesuai dengan *overlay* agar mendapatkan hasil analisis yang maksimal.



Gambar 3.11 Halaman kamera (iterasi ke-1)

i. Halaman Beranda

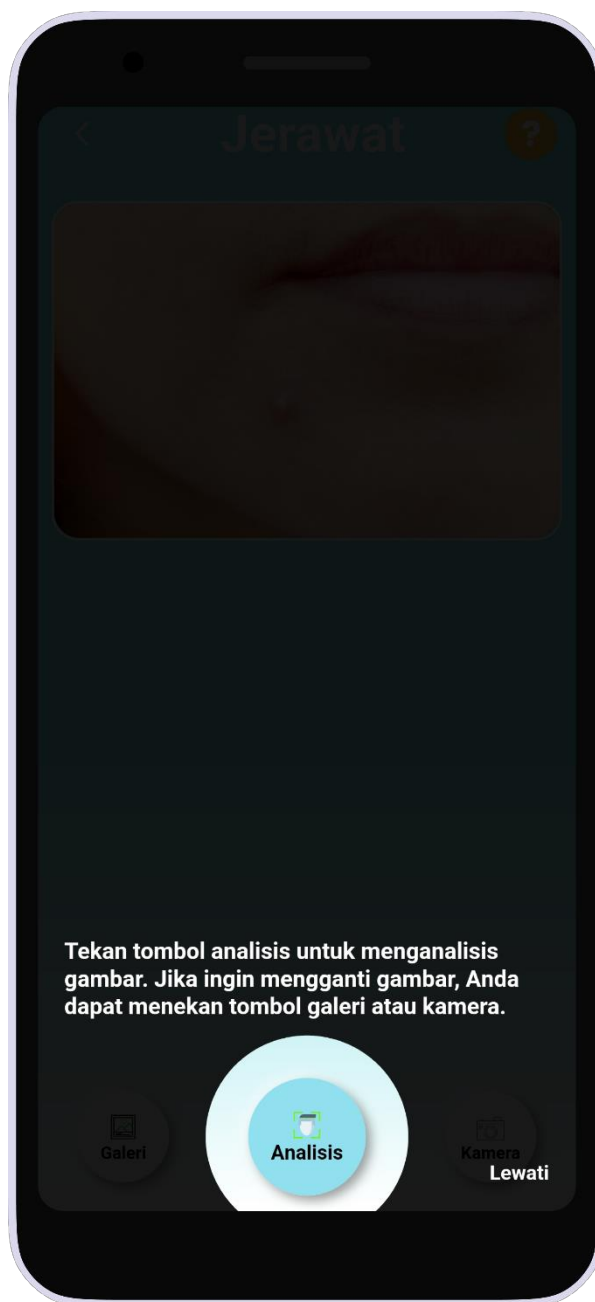
Halaman beranda (perhatikan Gambar 3.12), berisi menu-menu analisis antara lain jerawat, keriput, kemerahan, bercak hitam, dan jenis kulit. Tentu saja menu analisis ini dapat bertambah di kemudian hari. Pada bagian bawah terdapat navigasi yang terdiri dari menu beranda, riwayat, dan profil.



Gambar 3.12 Halaman beranda (iterasi ke-1)

j. Halaman Tutorial Menu Analisis

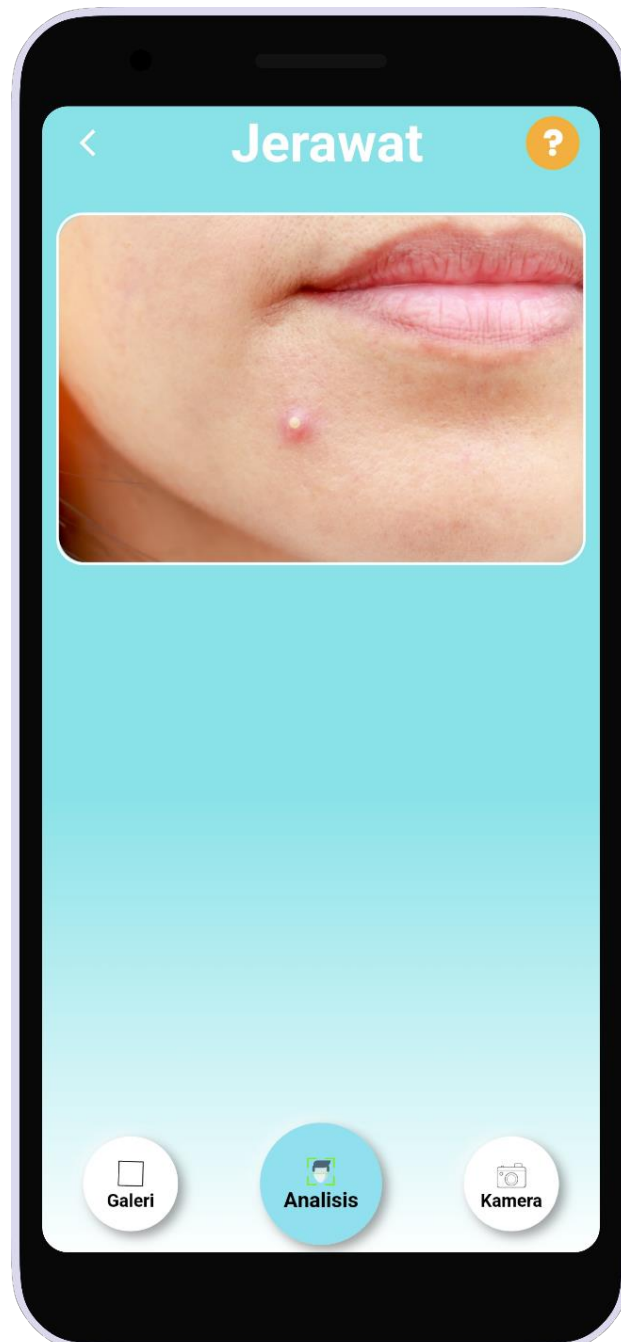
Saat pertama kali masuk pada menu analisis maka akan ditampilkan halaman tutorial yang sama dengan tutorial pada halaman selamat datang sebelumnya. Halaman tutorial ini (pada Gambar 3.13), menjelaskan akan tombol analisis artinya pengguna dapat menekan tombol analisis untuk menganalisis gambar yang telah tersimpan pada halaman selamat datang sebelumnya. Selain itu, pengguna dapat mengganti gambar yang ingin dianalisis dengan menekan tombol ambil dari galeri atau kamera.



Gambar 3.13 Halaman Tutorial Menu Analisis (iterasi ke-1)

k. Halaman Menu Analisis

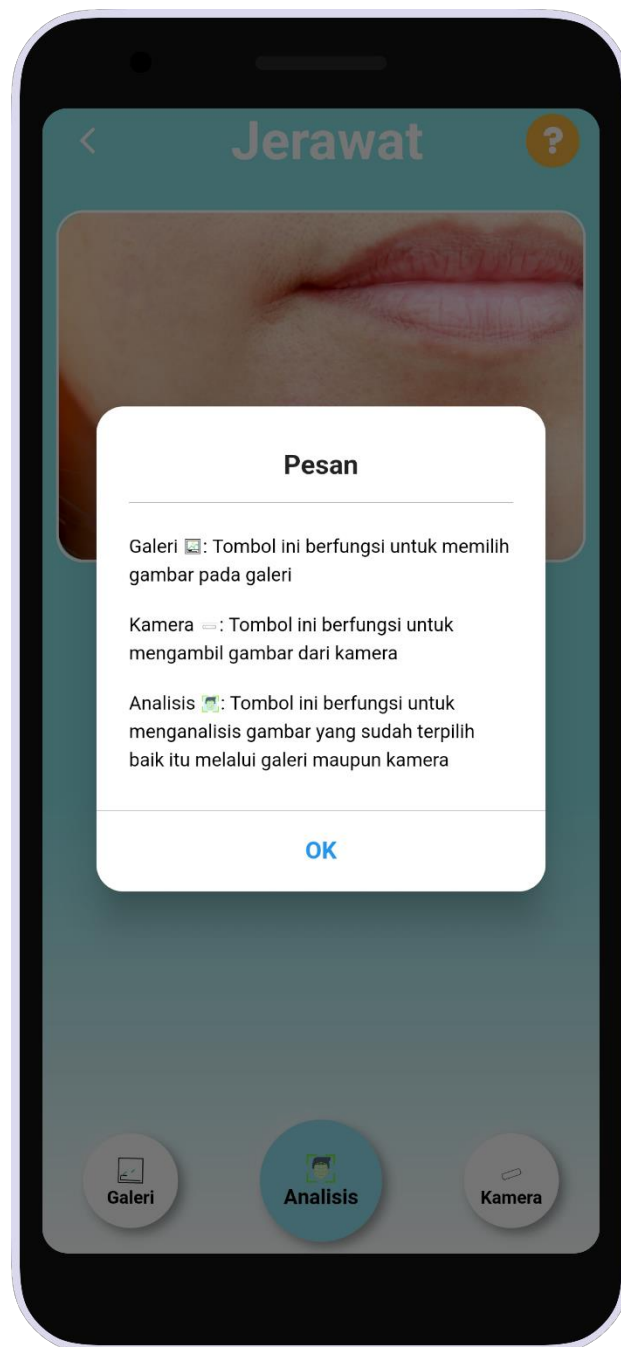
Pada halaman menu analisis (pada Gambar 3.14), berisi gambar yang sudah tersimpan sebelumnya dan tombol untuk menganalisis, mengganti gambar baik memilih gambar lain di galeri maupun mengambil gambar baru melalui kamera.



Gambar 3.14 Halaman Menu Analisis (iterasi ke-1)

1. Halaman *Pop-up* Tombol Bantuan Halaman Analisis

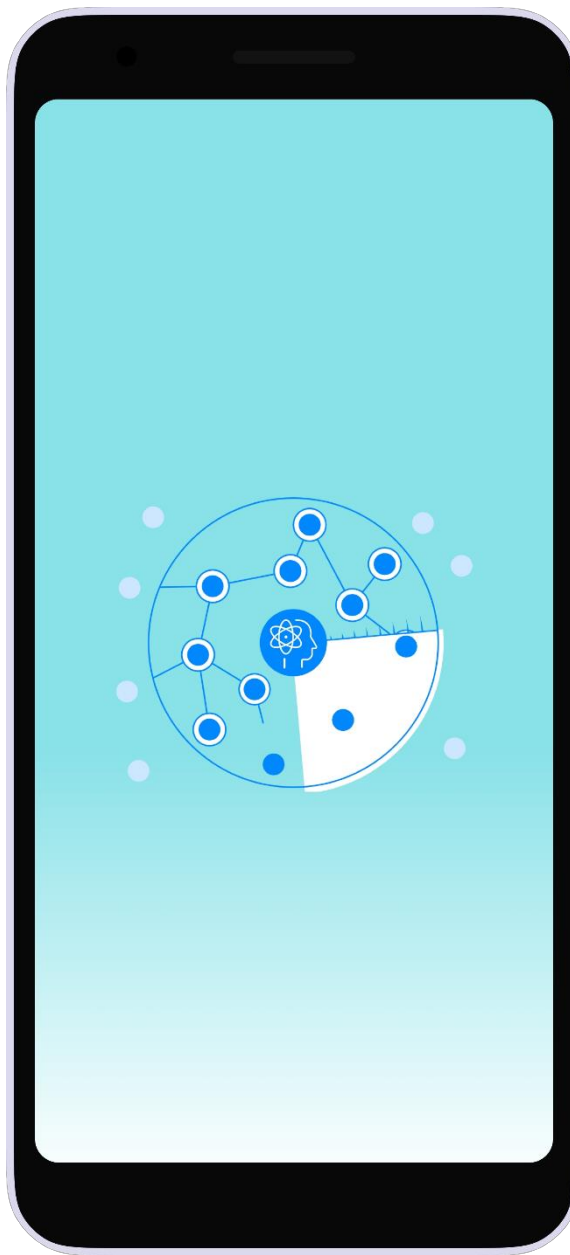
Halaman *pop-up* tombol bantuan halaman analisis ini seperti pada Gambar 3.15 berisi penjelasan singkat terkait kegunaan tombol-tombol yang ada pada menu analisis. Pada *pop-up* ini pengguna diberikan satu tombol “OK” untuk ditekan jika sudah mengerti dan ingin menutup *pop-up* yang ada.



Gambar 3.15 Halaman *Pop-up* Tombol Bantuan Halaman Analisis (iterasi ke-1)

m. Halaman *Loader* Saat Analisis

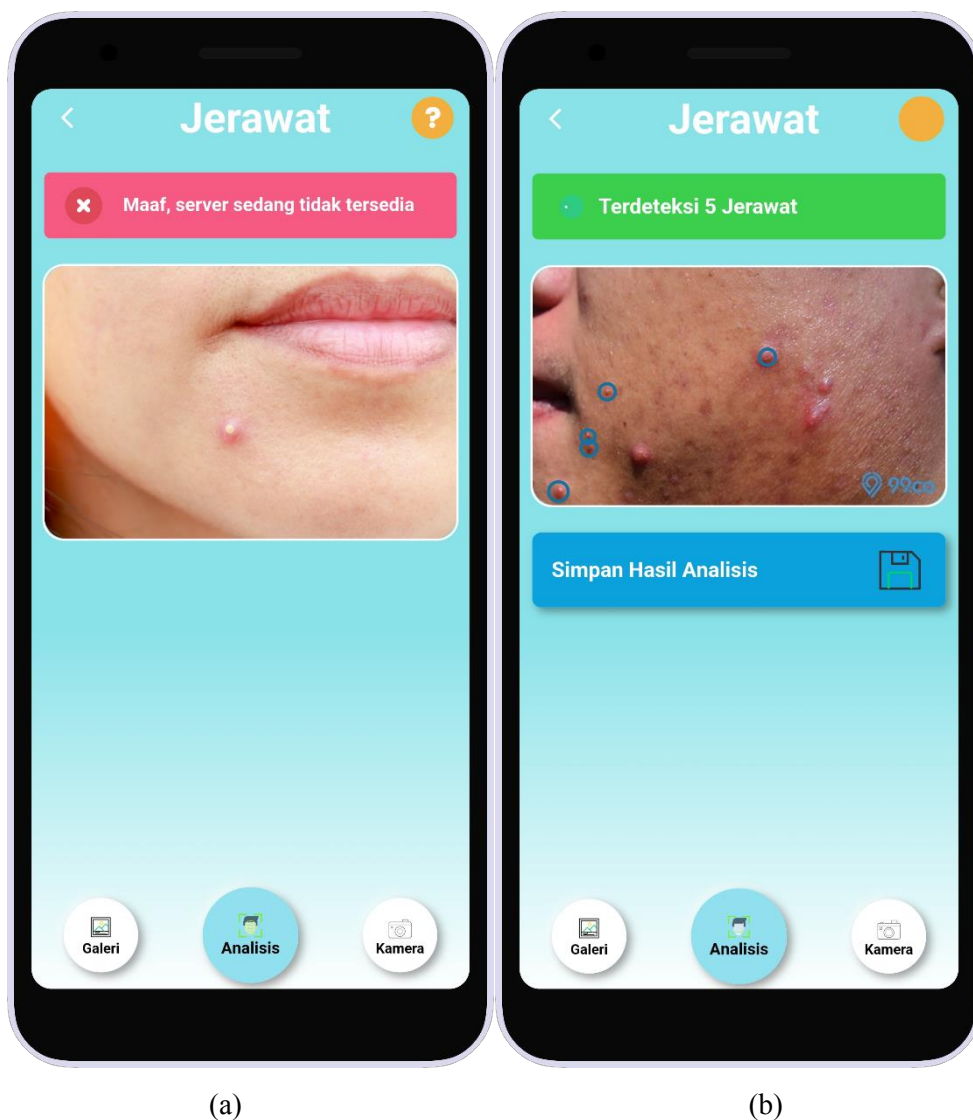
Halaman *loader* saat analisis ini (perhatikan Gambar 3.16) akan ditampilkan setelah pengguna menekan tombol analisis. Analisis dilakukan dengan mengirimkan gambar ke *server* aplikasi *machine learning* untuk mendeteksi gambar. Pengguna tetap berada pada halaman ini selama melakukan proses analisis. Proses analisis membutuhkan waktu yang sangat bervariasi dan sangat bergantung dengan kecepatan internet pengguna serta kemampuan atau keandalan *server*.



Gambar 3.16 Halaman *Loader* Saat Analisis (iterasi ke-1)

n. Halaman Hasil Analisis

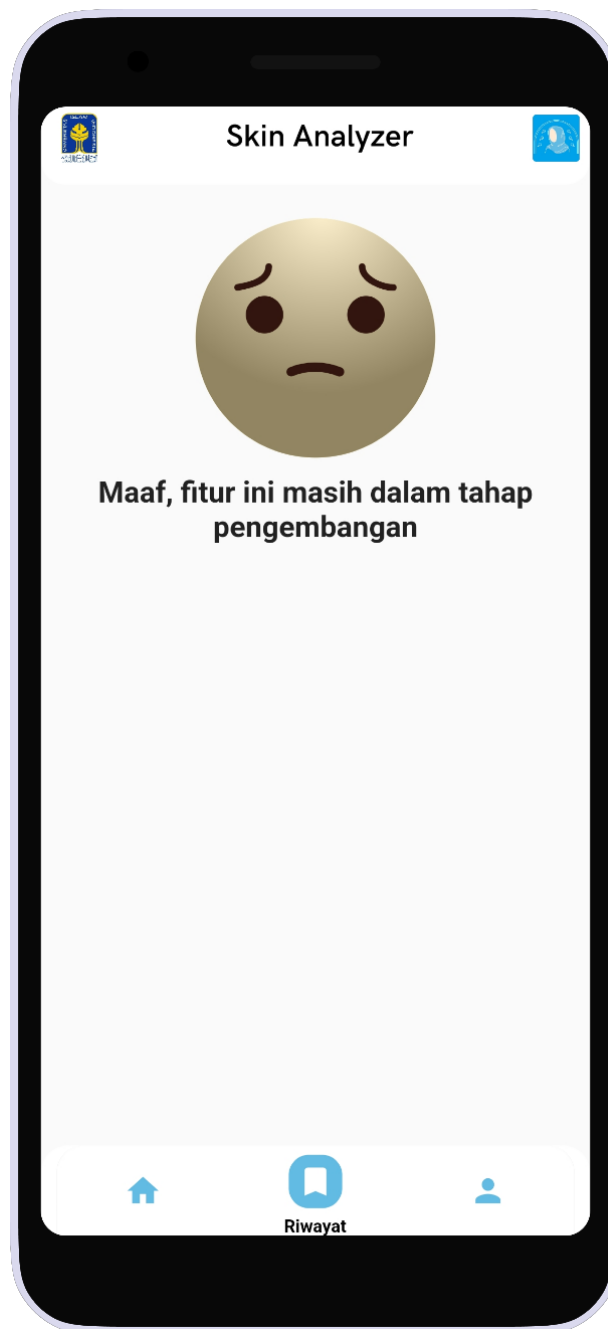
Setelah analisis dilakukan oleh *server*, *server* akan mengembalikan data hasil analisis. Data tersebut kemudian diolah dan ditampilkan pada halaman analisis (perhatikan Gambar 3.17). Jika terdapat gangguan pada *server* akan ditampilkan notifikasi seperti pada gambar pertama (a), dan jika analisis berhasil dilakukan maka akan ditampilkan notifikasi seperti gambar kedua (b). Jika analisis berhasil dilakukan dan ditemukan adanya jerawat seperti contoh gambar kedua, maka akan ditampilkan *highlight* jerawat pada gambar yang dianalisis. Selain itu, jika analisis berhasil dilakukan, juga akan muncul tombol untuk menyimpan hasil analisis tersebut. Hasil analisis tersebut tersimpan hanya pada perangkat pengguna.



Gambar 3.17 Halaman Hasil Analisis (iterasi ke-1)

o. Halaman Riwayat

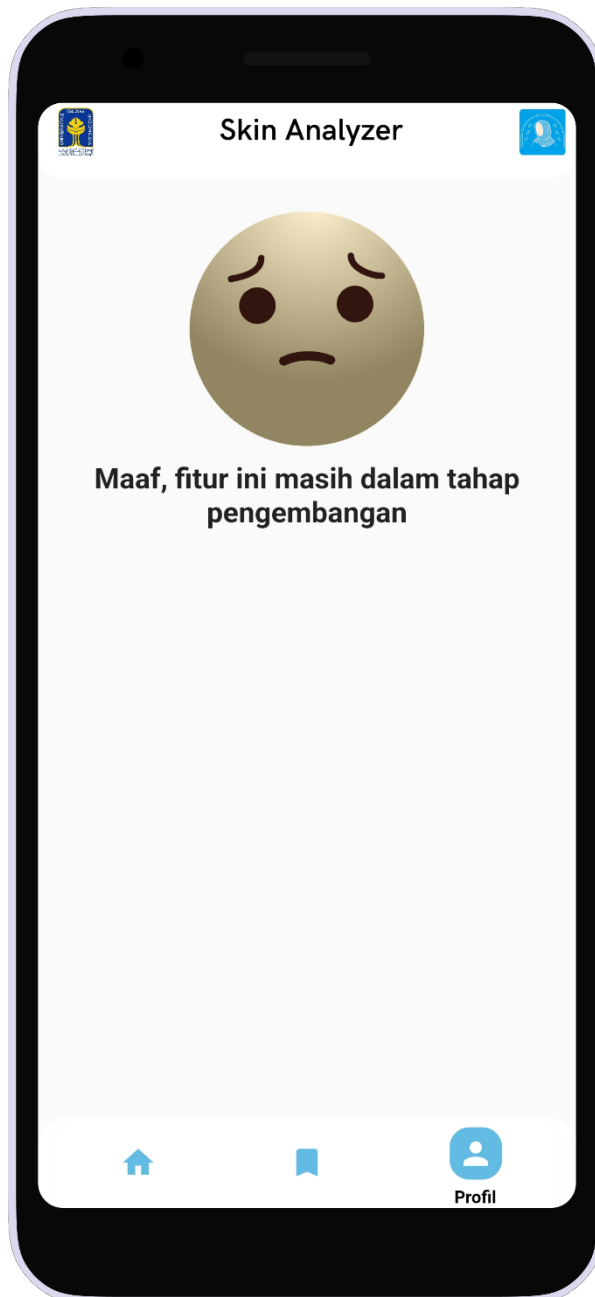
Pengguna nantinya dapat menyimpan hasil analisis untuk satu hari sekali pada setiap menu analisis. Halaman ini masih dalam tahap pengembangan seperti pada Gambar 3.18 ditampilkan tulisan “Maaf, fitur ini masih dalam tahap pengembangan” dan di atasnya diberikan animasi bergerak berupa emoji dengan ekspresi sedih.



Gambar 3.18 Halaman Riwayat (iterasi ke-1)

p. Halaman Profil

Pengguna nantinya dapat mendaftarkan akun ke aplikasi ini. Namun, halaman ini masih dalam tahap pengembangan seperti pada Gambar 3.19 ditampilkan tulisan “Maaf, fitur ini masih dalam tahap pengembangan” dan di atasnya diberikan animasi bergerak berupa emoji dengan ekspresi sedih.



Gambar 3.19 Halaman Profil (iterasi ke-1)

3.3.2 *Feedback Calon Pengguna*

Setelah melakukan perancangan, selanjutnya adalah mencari calon pengguna untuk memberikan *feedback* terkait rancangan tersebut. Calon pengguna dipilih berdasarkan kriteria yaitu laki-laki atau perempuan dengan usia mulai dari 12 sampai 60 tahun. *Feedback* yang didapatkan yaitu terdapat beberapa hal yang masih dapat ditingkatkan, terutama dalam hal *user interface* dan *user experience*. *Feedback* calon pengguna ini diperoleh menggunakan pengujian kuantitatif dengan *System Usability Scale* (SUS) dan melakukan diskusi dengan UI/UX *Enthusiast*.

Feedback Rancangan

Rancangan aplikasi UII *Skin Analyzer* yang telah dibuat mendapatkan beberapa *feedback* seperti yang tersedia pada Tabel 3.2 berikut:

Tabel 3.2 *Feedback* rancangan aplikasi UII *Skin Analyzer*

No	<i>Feedback</i>
1	Halaman logo terlalu kosong.
2	Tulisan seharusnya berada di bawah gambar ilustrasi.
3	Tombol simpan hasil analisis taruh di atas.
4	Penggunaan tombol berbentuk lingkaran terlalu banyak.
5	Desain <i>home page</i> dan halaman selamat datang disamakan.
6	Halaman selamat datang lebih baik ditiadakan.

System Usability Scale (SUS)

Pengujian *system usability scale* memiliki 10 instrumen pernyataan yang menjadi tolak ukur pengujian. Instrumen pengujian ini dapat digunakan ke berbagai macam jenis *software* (*mobile app*, *website*) dan *hardware*. Pengujian *system usability scale* memiliki 10 instrumen pernyataan yang tersedia pada Tabel 3.3 berikut:

Tabel 3.3 Instrumen pengujian *System Usability Scale* (SUS)

No	Pernyataan
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.

3	Saya merasa sistem ini mudah digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Seperti yang terlihat pada Tabel 3.3, dalam instrumen pengujian *system usability scale* (SUS) terdiri dari 10 pernyataan. Dari 10 pernyataan di atas telah berisi 5 pernyataan positif dan 5 pernyataan negatif. Setiap pernyataan memiliki skala penilaian yang menjadi ukuran pembobotan dalam pengujian. Responden diberikan lima pilihan jawaban yang dapat dilihat pada Tabel 3.4 berikut:

Tabel 3.4 Pilihan jawaban dan skor responden

No	Pilihan Jawaban	Skor
1	Sangat Tidak Setuju (STS)	1
2	Tidak Setuju (TS)	2
3	Ragu-Ragu (RG)	3
4	Setuju (S)	4
5	Sangat Setuju (SS)	5

Pada pengujian ini dilaksanakan secara *offline* dan dilaksanakan pada rentang 4 sampai 11 Oktober 2022. Hasil pengujian *system usability scale* terhadap aplikasi UII *Skin Analyzer* dengan 30 responden dan rata-rata nilai yang dihasilkan tersedia pada Tabel 3.5 berikut:

Tabel 3.5 Hasil dan rata-rata *system usability scale* (SUS)

No	Pernyataan	Skor
1	Saya berpikir akan menggunakan sistem ini lagi.	3.0
2	Saya merasa sistem ini rumit untuk digunakan.	3.7
3	Saya merasa sistem ini mudah digunakan.	4.0
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.	3.7
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.	3.7

6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).	2.9
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.	3.6
8	Saya merasa sistem ini membingungkan.	3.8
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.	3.5
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.	3.0

Pengujian terhadap aplikasi UII *Skin Analyzer* dengan *system usability scale* (SUS) dilakukan kepada 30 responden menggunakan bantuan Google form yang berisi 10 pernyataan dan responden terdiri dari rentang usia 12-60 tahun. Para responden diminta untuk mengisi data diri yang terdiri dari: nama, usia, dan jenis kelamin. Profil 30 responden dapat dilihat pada Lampiran Profil Responden. Selanjutnya, responden diminta untuk melakukan pengujian terhadap aplikasi UII *Skin Analyzer* dengan mengisi 10 pernyataan yang terdapat pada Tabel 3.3. Dari pengujian ini dihasilkan nilai rata-rata seperti yang tersedia pada Tabel 3.4. Perhitungan hasil pengujian menggunakan metode *system usability scale* (SUS) dilakukan dengan cara berikut:

1. Pernyataan instrumen ganjil (positif) adalah skala jawaban instrumen dikurangi 1.
2. Pernyataan instrumen genap (negatif) adalah 5 dikurangi skala jawaban instrumen.
3. Hasil penilaian berada pada rentang skala 0 – 4 (4 merupakan jawaban terbaik).
4. Melakukan penjumlahan jawaban kemudian dikali dengan 2.5.
5. Menentukan hasil rata-rata jawaban instrumen pengujian dari semua responden.

Persamaan perhitungan *system usability scale* (SUS) dilihat pada persamaan (3.1), (3.2), (3.3).

$$Y_{ganjil} = X_{ganjil} - 1 \quad (3.1)$$

$$Y_{genap} = 5 - X_{ganjil} \quad (3.2)$$

$$Skor = Y_{ganjil} + Y_{genap} \cdot 2.5 \quad (3.3)$$

Skala Likert memiliki lima penilaian, yaitu: Sangat Setuju (SS), Setuju (S), Ragu-ragu(R), Tidak Setuju (TS), Sangat Tidak Setuju (STS). Dengan menggunakan penilaian tersebut, maka standar penilaian akhirnya adalah:

- a) 0 – 19.99% = Sangat Kurang.
- b) 20 – 39.99% = Kurang.
- c) 40 – 59.99% = Cukup.
- d) 60 – 79.99% = Baik.
- e) 80 – 100% = Sangat Baik.

Berikut ketentuan penentuan penilaian pada SUS score percentile rank(Bangor dkk., 2009).

- a) Nilai A: skor $\geq 80,3$
- b) Nilai B: skor ≥ 74 dan $< 80,3$
- c) Nilai C: skor ≥ 68 dan < 74 .
- d) Nilai D: skor ≥ 51 dan < 68 .
- e) Nilai F: skor < 51 .



Gambar 3.20 Kriteria *System Usability Scale* (SUS)

Pada Gambar 3.21 terlihat kriteria dari *system usability scale* yang menunjukkan bahwa *acceptably ranges* terdiri dari *not acceptable*, *low*, *high*, dan *acceptable*. *Grade scale* terdiri dari A, B, C, D, dan E. *Adjective ratings* terdiri dari *worst imaginable*, *poor*, *OK*, *good*, *excellent*, *best imaginable*. Setelah melakukan pengumpulan data hasil dari responden, data tersebut dihitung menggunakan rumus yang tertera di atas. Pengujian terhadap aplikasi UII *Skin Analyzer* menghasilkan nilai *Y* pada pernyataan nomor ganjil (pernyataan positif) sebesar 17.27. Selanjutnya nilai *Y* pada pernyataan nomor genap (pernyataan negatif) dijumlahkan menghasilkan nilai sebesar 15.77. Untuk mendapatkan skor akhir digunakan Persamaan (3). Dari perhitungan tersebut, UII *Skin Analyzer* mendapatkan nilai sebesar 82.6%. Sehingga dapat

disimpulkan bahwa aplikasi UII *Skin Analyzer* berdasarkan rentang kriteria *system usability scale* (SUS) pada Gambar 3.21 menunjukkan bahwa aplikasi memiliki nilai A dan menurut skala *Likert* berada pada kategori Sangat Baik. Rekapitulasi hasil SUS dapat dilihat pada Lampiran Rekapitulasi perhitungan SUS.

Selama pengujian berlangsung terdapat beberapa responden yang memberikan *feedback* terhadap permasalahan, kesulitan, dan kendala yang dialami saat menggunakan aplikasi UII *Skin Analyzer* dalam bentuk verbal. Selain itu, penulis juga berdiskusi dengan salah satu UI/UX *Enthusiast* terkait tampilan antarmuka dan pengalaman pengguna saat menggunakan aplikasi ini. Berdasarkan *feedback* dan hasil diskusi maka temuan-temuan tersebut dirangkum dan disajikan pada Tabel 3.6 berikut:

Tabel 3.6 Rangkuman temuan aplikasi UII *Skin Analyzer*

No	Temuan-temuan
1	Halaman <i>Onboarding</i> selalu muncul pada saat pengguna membuka aplikasi.
2	Tombol ‘analisis’ pada halaman deteksi masih belum intuitif.
3	Halaman <i>home page</i> tidak terdapat perbedaan yang signifikan antara fitur yang masih dalam pengembangan dan fitur yang sudah bisa digunakan.
4	Pada halaman <i>home page</i> terlalu banyak animasi bergerak/ <i>gif</i> sehingga <i>user</i> menjadi fokus ke animasi tersebut.
5	Tidak ada tombol “Lewati” pada <i>onboarding</i> sehingga pengguna harus <i>swipe-left</i> dan membaca tulisan pada halaman ini.

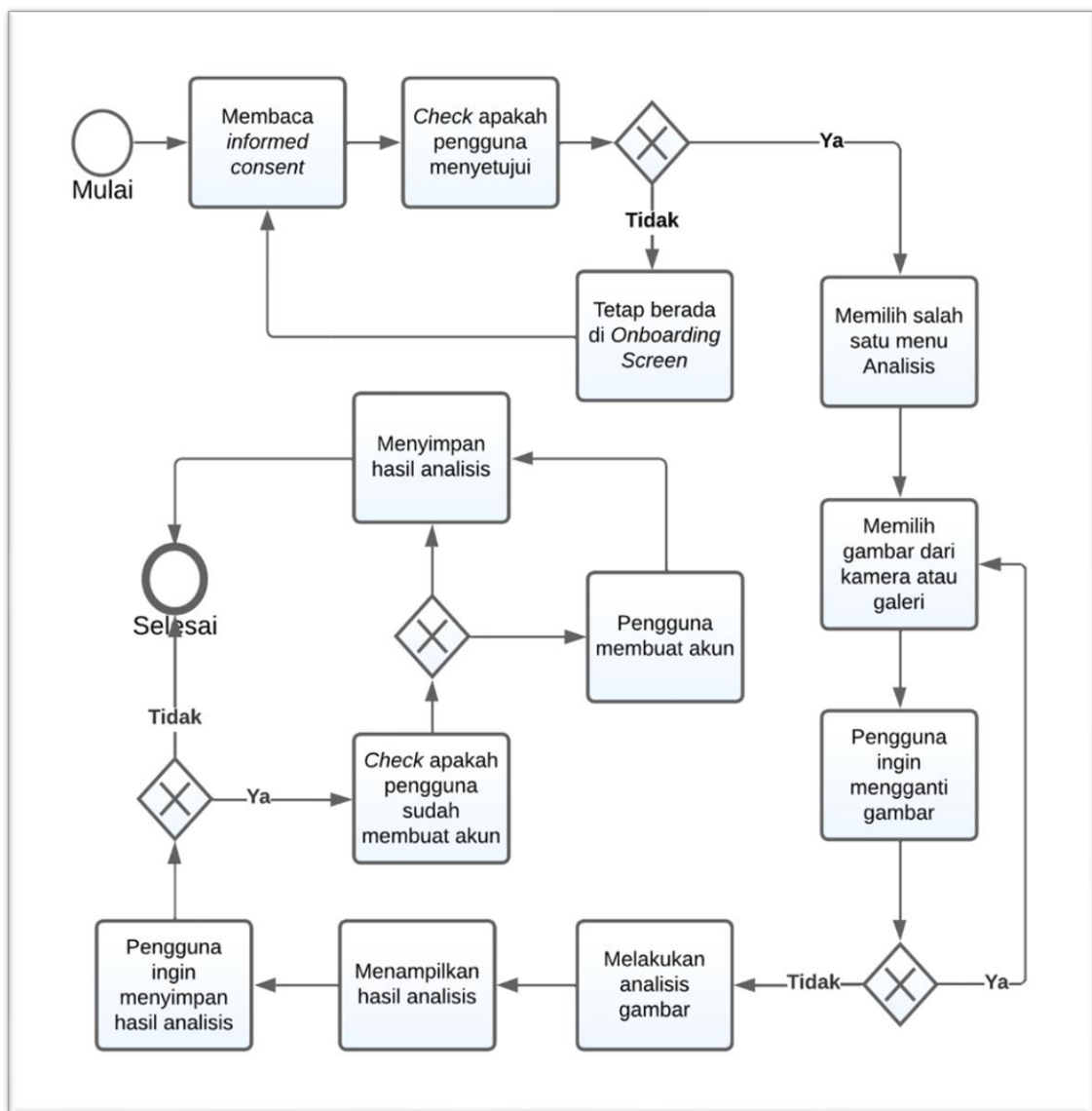
3.3.3 Iterasi Ke-2

Setelah menerima *feedback* dari calon pengguna, dilakukannya kembali fase perancangan dengan iterasi yang kedua. Proses perancangan iterasi kedua ini menekankan pada apa-apa saja yang menjadi bahasan dari calon pengguna pada saat melihat prototipe dari aplikasi. Dengan berdasarkan pada hal tersebut, maka dibuat kembali rancangan aplikasi mulai dari BPMN hingga prototipe.

Business Process Model Notation (BPMN)

Dengan memperhatikan *feedback* dari calon pengguna, maka dirancang kembali proses bisnis dari aplikasi, yang dituangkan ke dalam diagram BPMN. Perubahan proses bisnis ada pada dihilangkannya *welcome page*, dan penambahan proses pendaftaran akun saat ingin

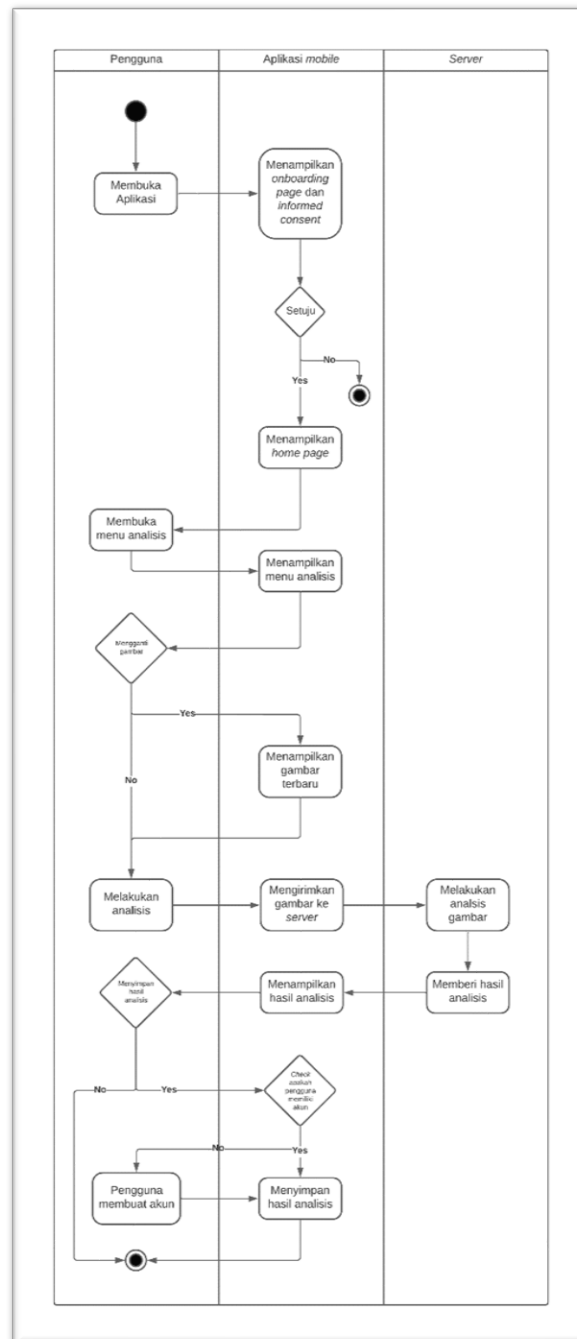
menyimpan hasil analisis. Pada iterasi sebelumnya, *welcome page* berguna untuk menyimpan gambar, di mana gambar tersebut dapat langsung pengguna analisis pada tiap-tiap menu analisis yang ada. Langkah ini membuat calon pengguna bingung dan dirasa tidak perlu, dikarenakan pengguna dapat langsung memasukkan gambar pada tiap-tiap menu analisis. Untuk fitur akun, pada iterasi sebelumnya terdapat kesalahan yang cukup fatal, yaitu pengguna hanya dapat membuat akun, dan tidak dapat masuk. Sehingga dengan bantuan dari *feedback* yang didapat, fitur akun dapat difinalisasi proses bisnisnya. Diagram BPMN dapat dilihat pada Gambar 3.22.



Gambar 3.21 Diagram BPMN

Diagram Aktivitas

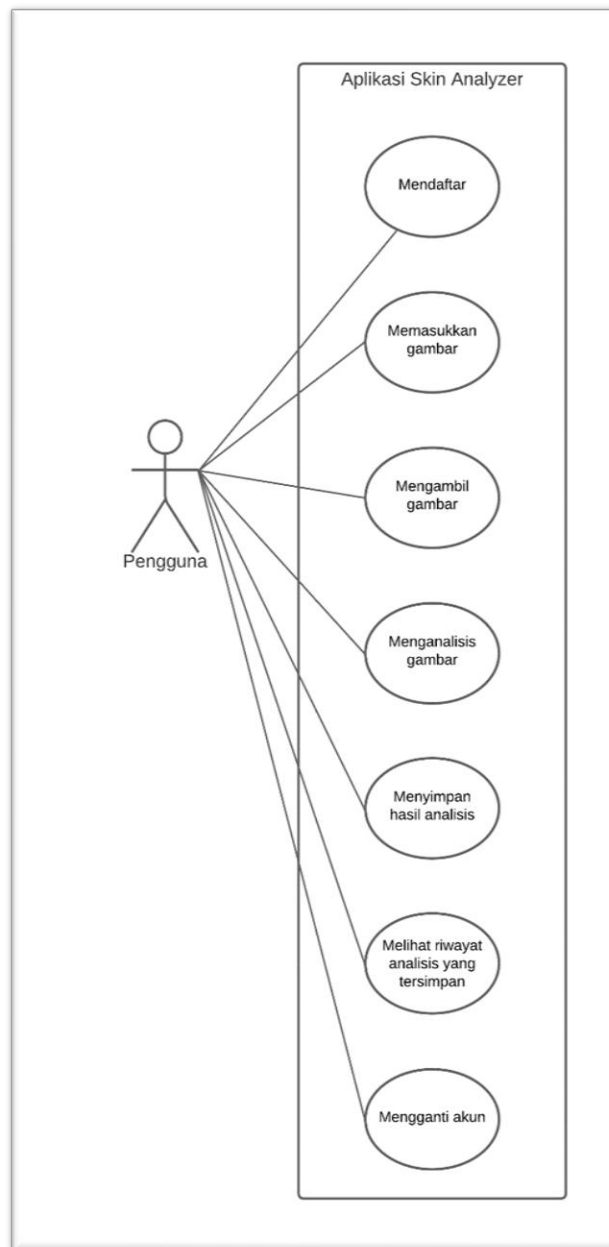
Dengan perubahan alur proses bisnis, diagram aktivitas juga mengalami perubahan. Perubahan tersebut kurang lebih sama dengan perubahan yang ada pada diagram BPMN, yaitu dihilangkannya *welcome page*, dan perbaikan pada fitur akun untuk menyimpan hasil analisis. Diagram aktivitas dapat dilihat pada Gambar 3.23.



Gambar 3.22 Diagram Aktivitas

Use Case Diagram

Untuk *use case diagram*, perubahan yang terjadi ada pada tambahan *case* masuk atau *login*, dikarenakan pada iterasi sebelumnya belum dibuatnya kemampuan aplikasi untuk masuk ke akun yang telah dibuat pengguna. *Use case diagram* dapat dilihat pada Gambar 3.24.



Gambar 3.23 *Use Case Diagram*

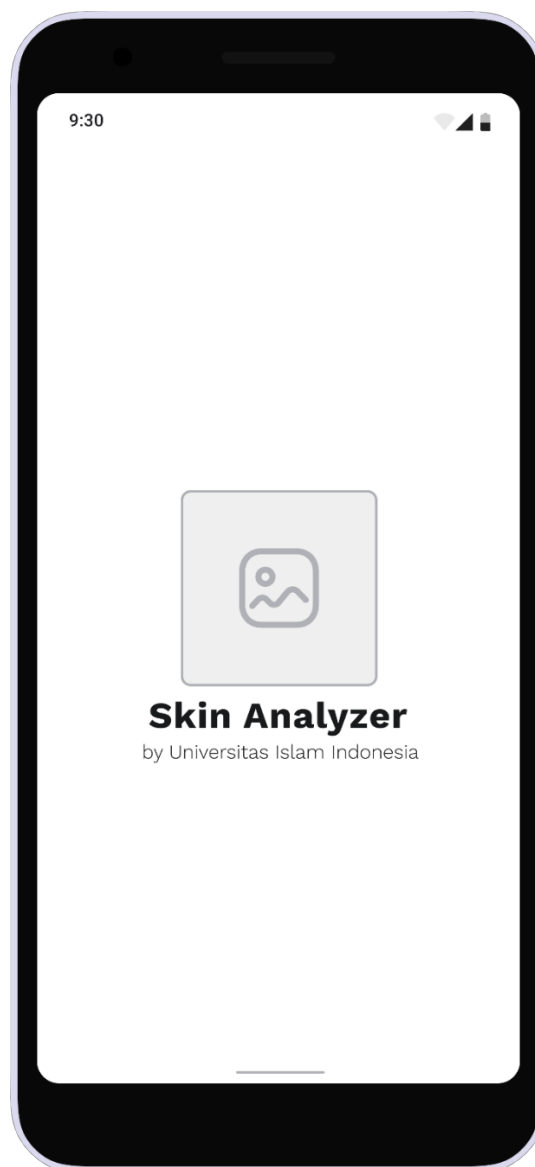
Wireframe Aplikasi

Untuk *wireframe* aplikasi, perubahan yang terjadi bukan hanya pada aspek proses atau alur aplikasi, tetapi juga pada aspek UI/UX. Hal tersebut dikarenakan pada iterasi pertama, terdapat aspek-aspek *design* yang masih dapat ditingkatkan seperti pada penggunaan tombol

berbentuk lingkaran yang tidak memperhatikan hirarki, sampai dengan posisi teks yang lebih dulu ditampilkan daripada gambar ilustrasi, yang membuat *experience* menjadi kurang sempurna.

a. Halaman Logo

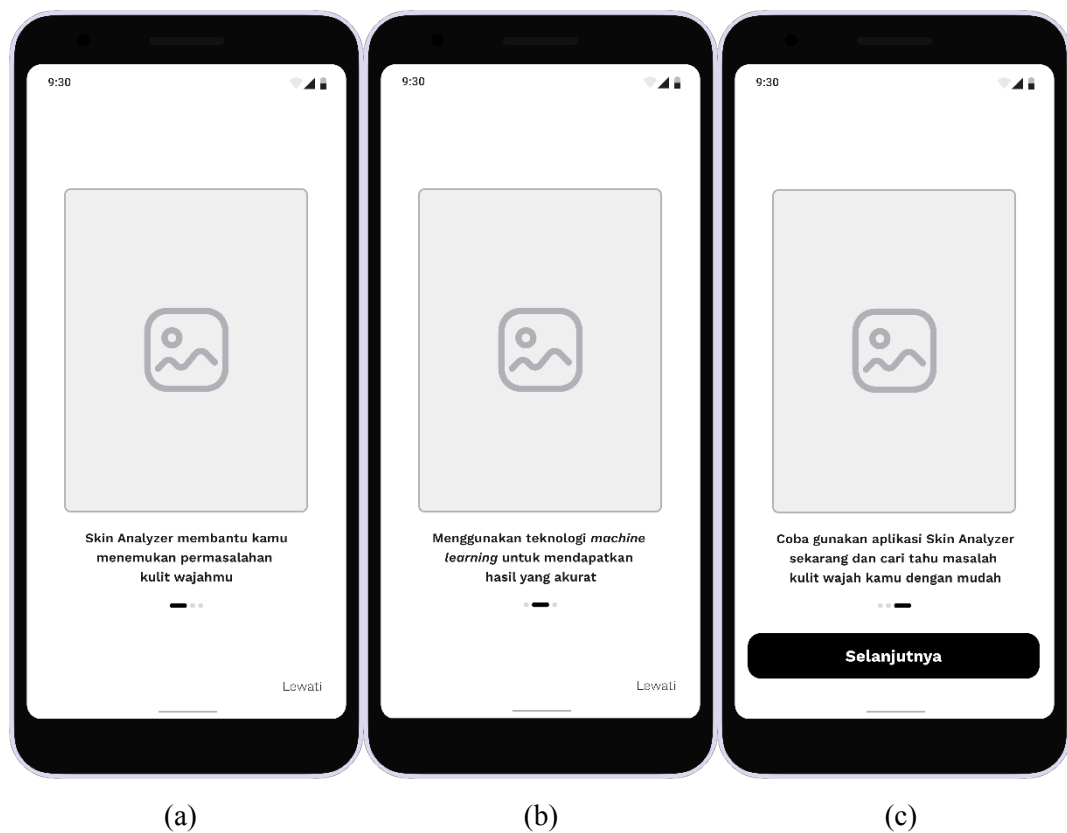
Halaman logo terdapat perubahan pada komposisinya (perhatikan Gambar 3.24). Pada iterasi 1, logo aplikasi dan nama aplikasi terletak bersebelahan yang menyebabkan banyaknya ruang kosong yang tidak terpakai serta kurangnya *highlight* dari logo dan nama aplikasi. Sehingga, dibuat logo dengan nama aplikasi disusun secara vertikal serta memperbesar ukuran logo dan teks yang ada.



Gambar 3.24 Halaman Logo (iterasi ke-2)

b. Halaman *Onboarding*

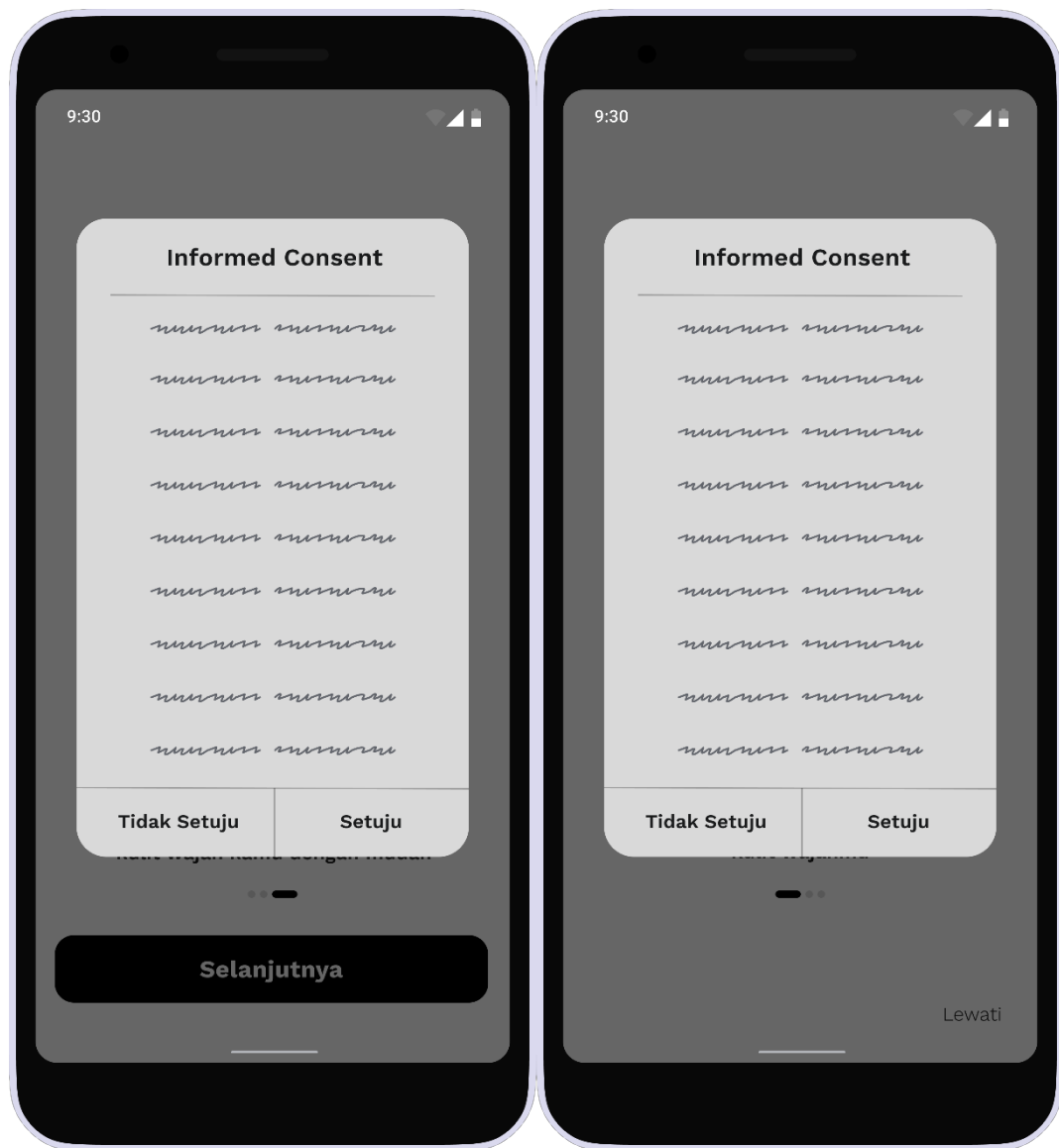
Pada halaman *onboarding* ditambahkan tombol lewati, diubahnya posisi gambar ilustrasi dan deskripsi singkat, serta menambahkan teks deskripsi menjadi sedikit lebih panjang (seperti pada Gambar 3.25). Tombol lewati ditambahkan agar pengguna memiliki opsi untuk langsung menggunakan aplikasi. Diubahnya posisi gambar dan teks dikarenakan pengguna akan melihat aplikasi dari atas ke bawah dan gambar yang cenderung lebih dominan sebaiknya ditempatkan di atas agar dapat langsung terlihat oleh pengguna. Pada tampilan *onboarding* yang pertama (seperti pada Gambar 3.26 bagian a), kalimatnya berubah menjadi “*Skin Analyzer* membantu kamu menemukan permasalahan kulit wajahmu”. Selanjutnya pada tampilan *onboarding* yang kedua (lihat Gambar 3.26 bagian b), kalimatnya berubah menjadi “Menggunakan teknologi *machine learning* untuk mendapatkan hasil yang akurat”. Terakhir, pada tampilan *onboarding* yang ketiga (perhatikan Gambar 3.26 bagian c), kalimatnya berubah menjadi “Coba gunakan aplikasi *Skin Analyzer* sekarang dan cari tahu masalah kulit wajah kamu dengan mudah”.



Gambar 3.25 Halaman *Onboarding* (iterasi ke-2)

c. Halaman *Pop-up Informed Consent*

Pop-up Informed Consent tidak memiliki perbedaan dengan iterasi yang sebelumnya (lihat Gambar 3.26), hanya jika pengguna menekan tombol lewati maka *pop up* juga akan muncul, dan sebelum pengguna menekan setuju, pengguna tidak dapat masuk ke tampilan berikutnya di aplikasi UII *Skin Analyzer* ini.



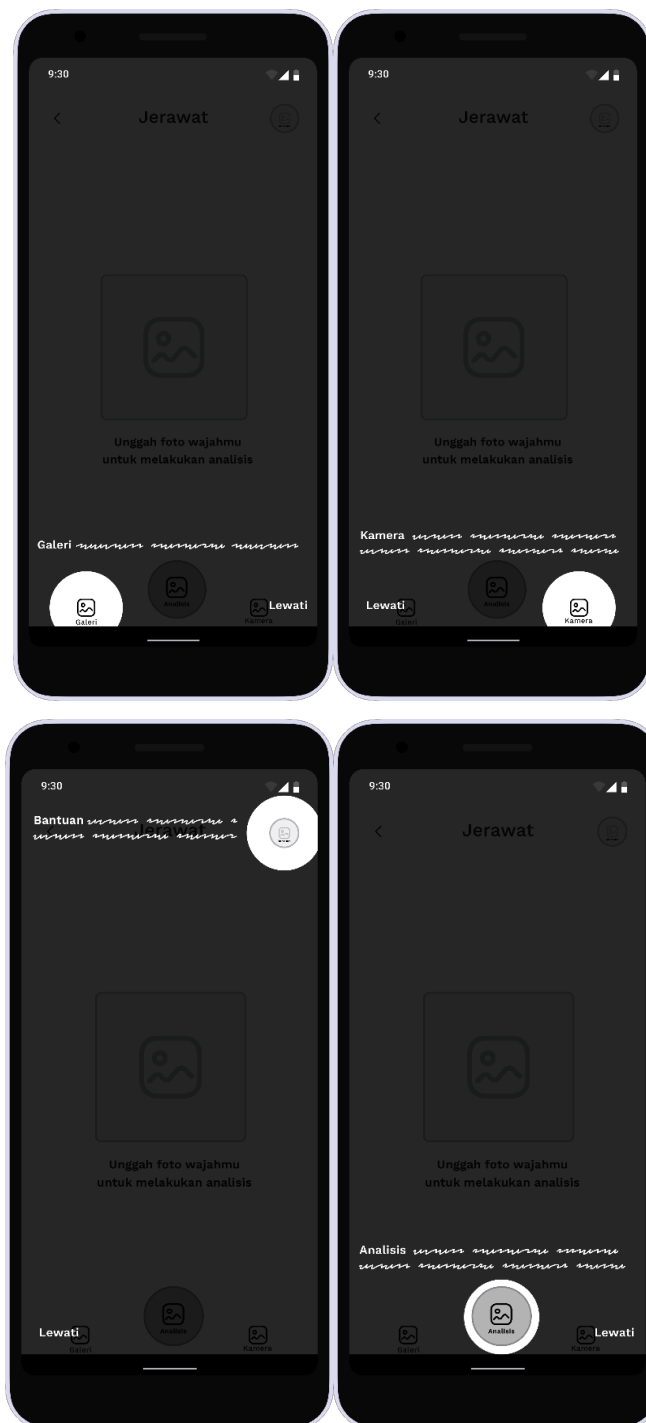
Gambar 3.26 Halaman *Pop-up Informed Consent* (iterasi ke-2)

d. Halaman Beranda

Halaman beranda tidak mengalami perubahan dari iterasi ke-1 (seperti pada Gambar 3.12). Hal ini disebabkan karena menu yang ditampilkan masih sama dan pengguna telah mendapatkan *experience* yang sangat baik saat berada pada halaman beranda.

e. Halaman Tutorial Menu Analisis

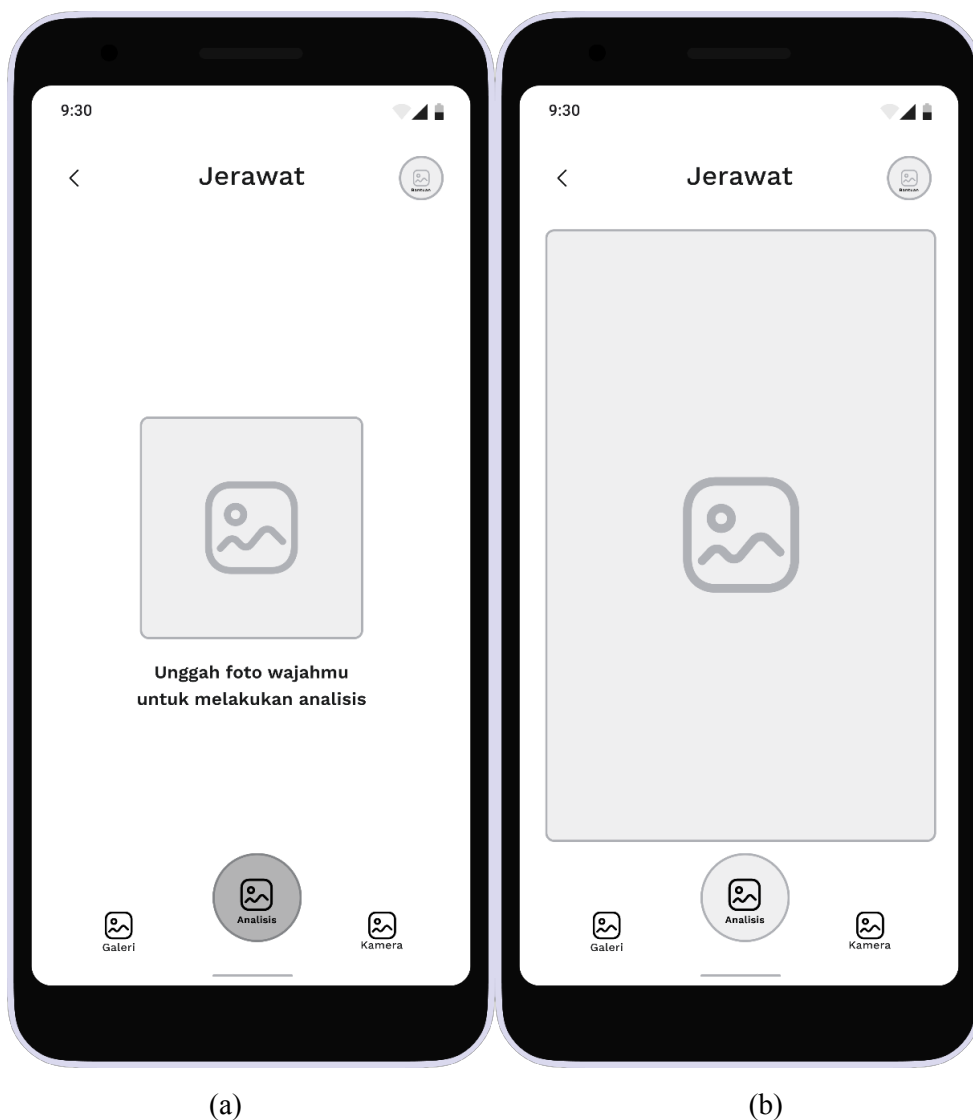
Halaman tutorial menu analisis mengalami perubahan karena halaman selamat datang sudah tidak ada lagi pada iterasi ke-2 ini. Sehingga halaman tutorial menu analisis ini dimulai pada saat pengguna baru pertama kali membuka halaman menu analisis (periksa Gambar 3.27).



Gambar 3.27 Halaman Tutorial Menu Analisis (iterasi ke-2)

f. Halaman Menu Analisis

Pada iterasi sebelumnya, saat pertama kali menggunakan aplikasi akan ditampilkan *welcome page* yang mengharuskan pengguna untuk memasukkan gambar sebelum menggunakan aplikasi. Proses tersebut dinilai tidak efektif karena pengguna dapat langsung memasukkan gambar pada halaman menu analisis. Sehingga pada halaman menu analisis (seperti pada Gambar 3.28), saat baru menggunakan aplikasi dan belum pernah memasukkan gambar maka tombol analisis tidak bisa ditekan (a), pengguna harus memasukkan gambar terlebih dahulu. Tombol analisis dapat ditekan jika pengguna telah memasukkan gambar (b). Selain itu, tombol-tombol interaksi yang ada seperti mengambil gambar dari galeri, mengalami perubahan.



Gambar 3.28 Halaman Menu Analisis (iterasi ke-2)

g. Halaman *Pop-up* Pesan dan Panduan Pengambilan Foto

Halaman *pop-up* pesan dan panduan pengambilan foto ini (lihat Gambar 3.29) tidak memiliki perbedaan dengan tampilan yang ada pada iterasi pertama.



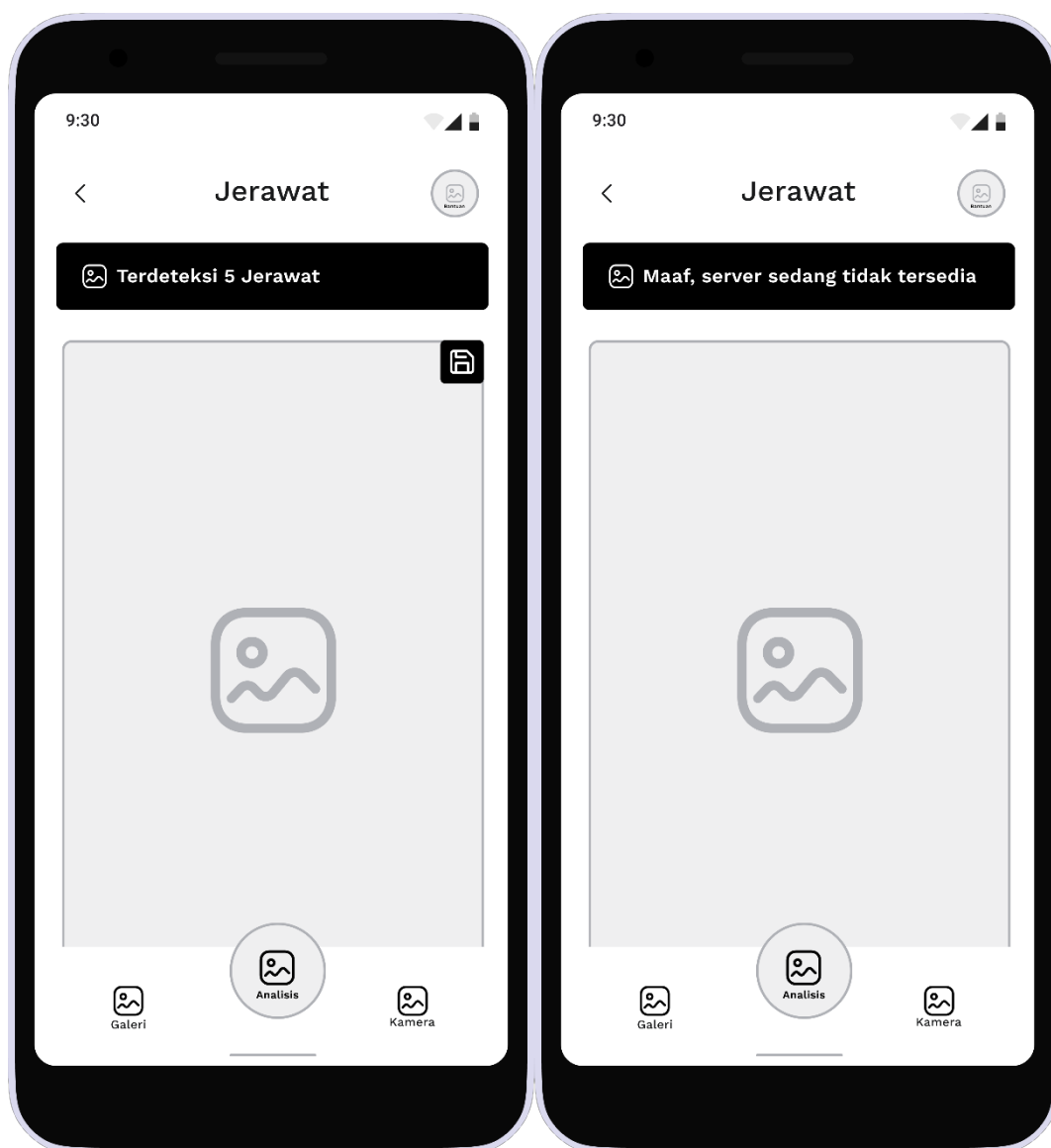
Gambar 3.29 Halaman *Pop-up* Pesan dan Panduan Pengambilan Foto (iterasi ke-2)

h. Halaman Kamera dan *Loader* Saat Analisis

Halaman kamera (lihat Gambar 3.11) dan halaman *loader* saat analisis (lihat Gambar 3.16) tidak mengalami perubahan dari iterasi ke-1. Hal ini disebabkan karena komposisi halaman yang ditampilkan masih sama dan pengguna tidak mengalami kendala saat berada pada kedua halaman ini.

i. Halaman Hasil Analisis

Pada iterasi sebelumnya, tombol-tombol interaksi seperti mengambil gambar dari galeri memiliki bentuk lingkaran yang terpisah-pisah. Hal tersebut memiliki kekurangan pada aspek UI/UX seperti yang terdapat pada Google *Material Design*. Kekurangan tersebut disebabkan karena tombol lingkaran memiliki hirarki yang paling tinggi. Sehingga tombol galeri, kamera, dan analisis diganti menjadi sebuah navigasi. Selain itu, tombol untuk menyimpan hasil analisis dibuat menjadi lebih sederhana dan *icon* yang digunakan pada tombol kamera dan galeri berupa gambar tidak berupa GIF (lihat Gambar 3.30).



Gambar 3.30 Halaman Hasil Analisis (iterasi ke-2)

j. Halaman Riwayat Saat Belum Login

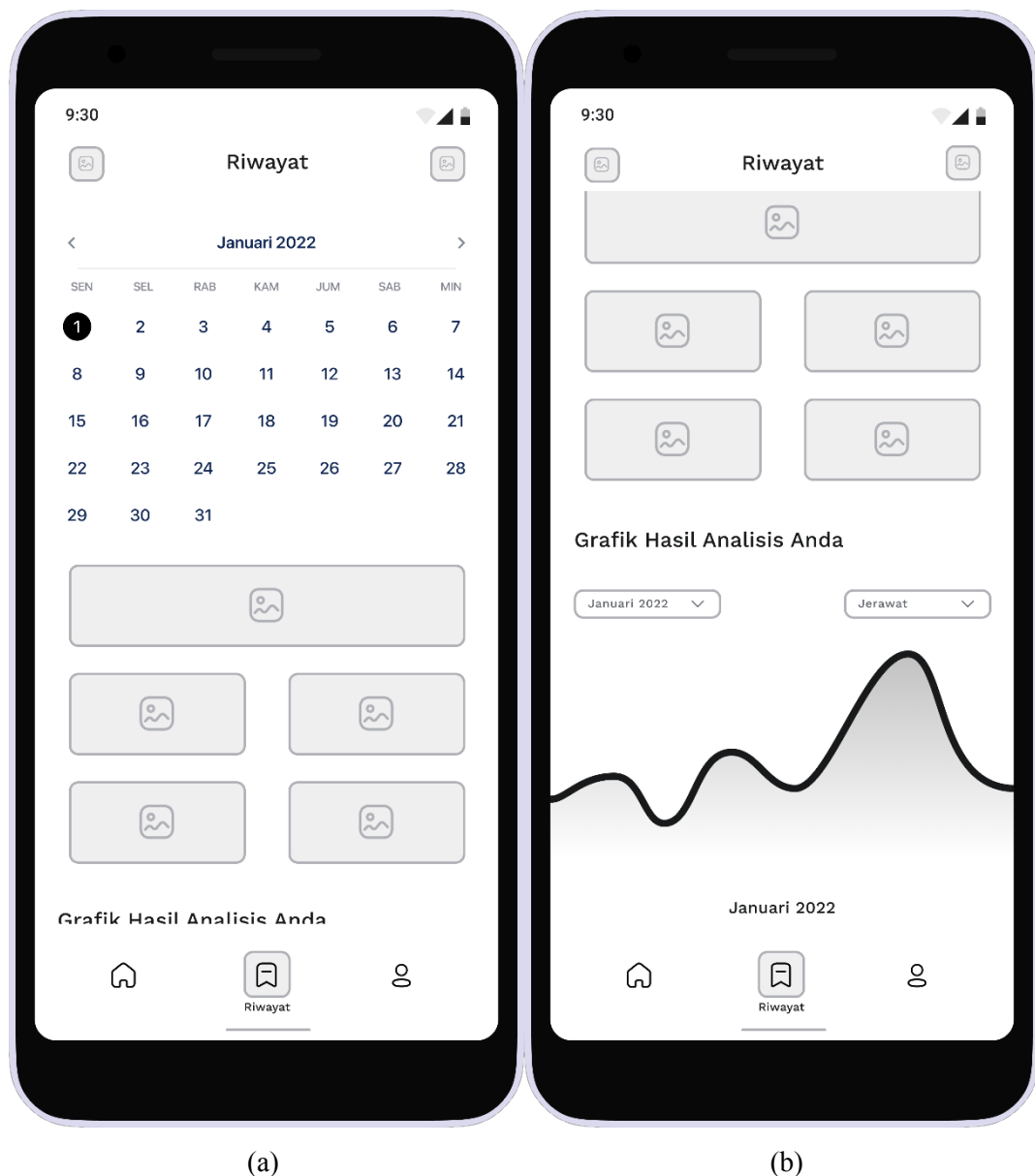
Pada halaman riwayat (lihat Gambar 3.31) tidak terdapat perbedaan yang signifikan dengan interaksi ke-1. Terdapat perubahan pada saat pengguna belum ada menyimpan hasil analisis maka halaman akan menampilkan pesan yang berisi “Simpan hasil analisis untuk menampilkan riwayat”.



Gambar 3.31 Halaman Riwayat (iterasi ke-2)

k. Halaman Riwayat

Pengguna dapat menyimpan hasil analisisnya untuk satu hari sekali pada setiap menu analisis. Pada halaman riwayat (lihat Gambar 3.32) akan ditampilkan kalender beserta grafik riwayat hasil analisis tiap bulannya (a). Di bawah kalender akan ditampilkan tombol-tombol untuk tiap-tiap menu analisis yang ada. Pengguna dapat menekan tombol tersebut untuk masuk ke tiap halaman riwayat hasil analisis pada hari sesuai yang dipilih di kalender. Pada Grafik hasil analisis (b), pengguna dapat melihat grafik hasil analisis tiap bulan serta untuk tiap-tiap menu analisis yang ada.



Gambar 3.32 Halaman Riwayat (iterasi ke-2)

1. Halaman Detail Riwayat

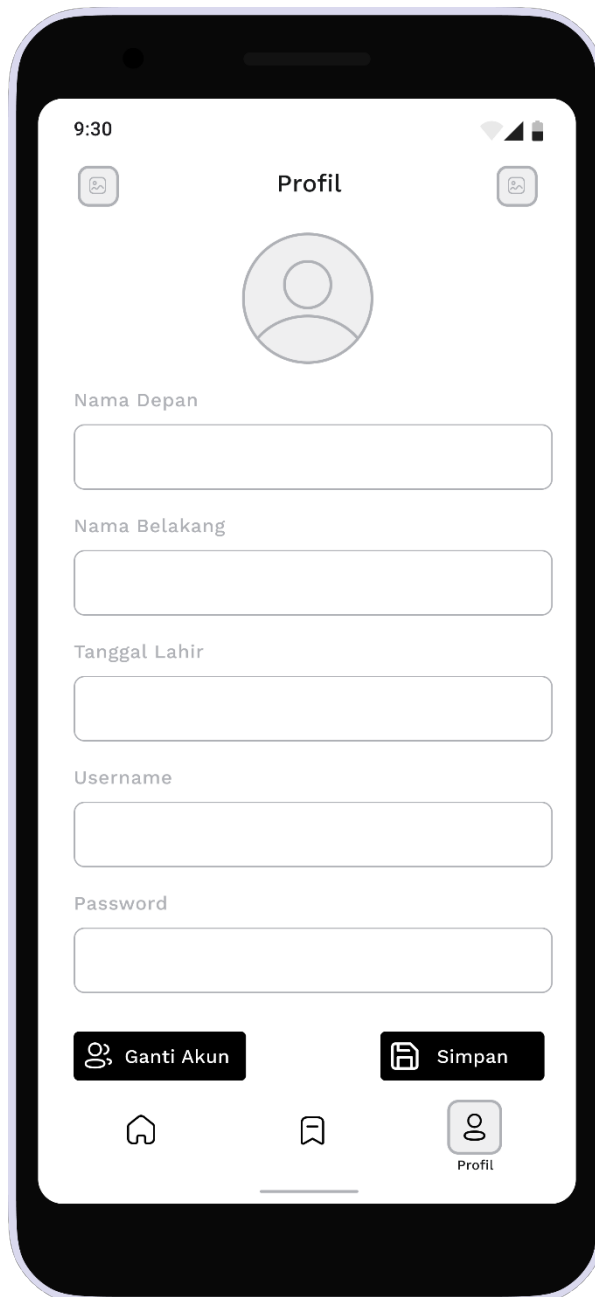
Halaman detail riwayat (lihat Gambar 3.33) akan ditampilkan setelah pengguna menekan tombol yang ada di bawah kalender. Halaman ini akan menampilkan gambar dengan tambahan hasil analisis yang sudah disimpan oleh pengguna sebelumnya. Halaman ini juga menampilkan kalender dengan format horizontal agar pengguna dapat langsung melihat ke hasil analisis yang tersimpan pada hari sebelum maupun setelahnya.



Gambar 3.33 Halaman Detail Riwayat (iterasi ke-2)

m. Halaman Profil

Pada halaman profil (perhatikan Gambar 3.34) tidak terdapat perbedaan yang signifikan dengan interaksi ke-1. Hanya ditambahkan tombol untuk mengganti akun agar pengguna dapat masuk ke dalam akun berbeda yang telah dibuat sebelumnya.



9:30

Profil

Nama Depan

Nama Belakang

Tanggal Lahir

Username

Password

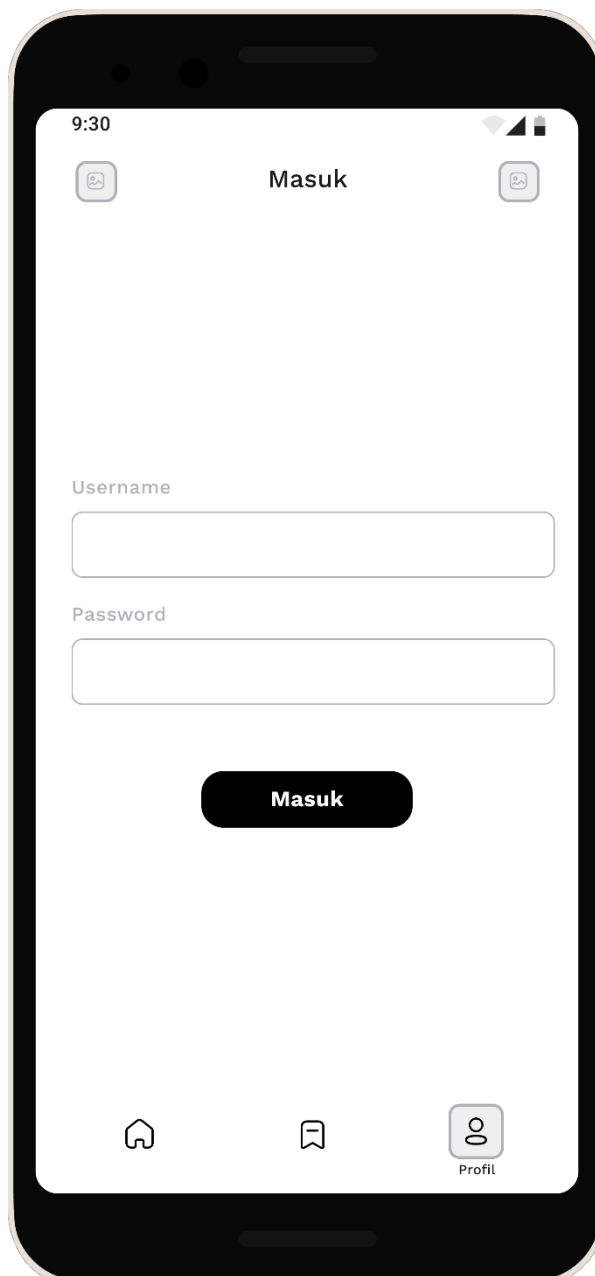
 Ganti Akun  Simpan

   Profil

Gambar 3.34 Halaman Profil (iterasi ke-2)

n. Halaman Masuk

Halaman masuk ini (pada Gambar 3.35) akan ditampilkan saat pengguna ingin masuk ke akun atau mengganti akun. Pada halaman ini pengguna diminta untuk mengisi *username* dan *password* yang telah dibuat. Jika sudah terisi, pengguna dapat menekan tombol “Masuk”.



Gambar 3.35 Halaman Masuk (iterasi ke-2)

Feedback Calon Pengguna

Pada iterasi kedua ini, dilakukan beberapa perbaikan untuk mengatasi kekurangan-kekurangan yang ada pada iterasi pertama. *Feedback* didapatkan dari calon pengguna yang pada iterasi pertama juga memberikan *feedback*. Hasil dari *feedback* tersebut adalah aplikasi dinilai sudah siap untuk digunakan.

3.4 Fase Implementasi

Fase implementasi adalah fase di mana rancangan yang telah dibuat sebelumnya, diimplementasikan ke dalam aplikasi yang sesungguhnya. Di sini juga dapat terlihat alasan yang membuat metode pengembangan perangkat lunak RAD memiliki waktu yang lebih singkat, yaitu dikarenakan pada saat perancangan sudah melibatkan calon pengguna, masalah-masalah serta kebutuhan-kebutuhan dari pengguna sudah jelas sehingga pengembang perangkat lunak dapat langsung mengembangkan perangkat lunak sesuai rancangan yang sudah disetujui oleh calon pengguna sebelumnya.

Pada aplikasi *mobile* UII *Skin Analyzer*, proses implementasi atau pengembangan menggunakan bahasa pemrograman Dart dan *framework* Flutter. *Database* yang digunakan untuk menyimpan hasil analisis dan akun yang dapat dibuat oleh pengguna adalah *database* lokal, sesuai dengan kebutuhan akan privasi pada calon pengguna aplikasi UII *Skin Analyzer*. *Database* yang digunakan yang *sqlite*, dengan dibantu penerapannya pada Flutter oleh *package* *sqlite*.

3.5 Fase Pengujian

Setelah melewati fase implementasi, fase yang terakhir yaitu fase pengujian. Fase pengujian merupakan fase terakhir dalam *System Development Life Cycle (SDLC)*. Fase ini meliputi konversi data, pengujian, dan pelatihan kepada pengguna. Pada fase ini, dilakukan *blackbox testing* yang menguji apakah fungsionalitas aplikasi sudah berjalan sebagaimana yang diharapkan atau belum. Jika dibandingkan dengan metode tradisional seluruh fase/proses dikompresi atau dipersingkat. Sehingga hasilnya dibangunnya sistem atau aplikasi baru kemudian dikirimkan ke pengguna dan dapat dioperasikan lebih cepat.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Berdasarkan rancangan yang telah dibuat, selanjutnya akan dikembangkan aplikasi *mobile* UII *Skin Analyzer*. Hasil dari proses pengembangan yang telah dilakukan adalah kode Flutter yang dapat di-*build* menjadi aplikasi *mobile* pada berbagai macam sistem operasi. Dalam proses pengembangan dan pengujiannya, aplikasi yang di-*build* berupa aplikasi berbasis Android. Sehingga luaran yang dihasilkan berupa aplikasi *mobile* UII *Skin Analyzer* yang sudah dapat di-*install* dan digunakan pada perangkat Android.

4.1.1 Analisis Hasil Desain

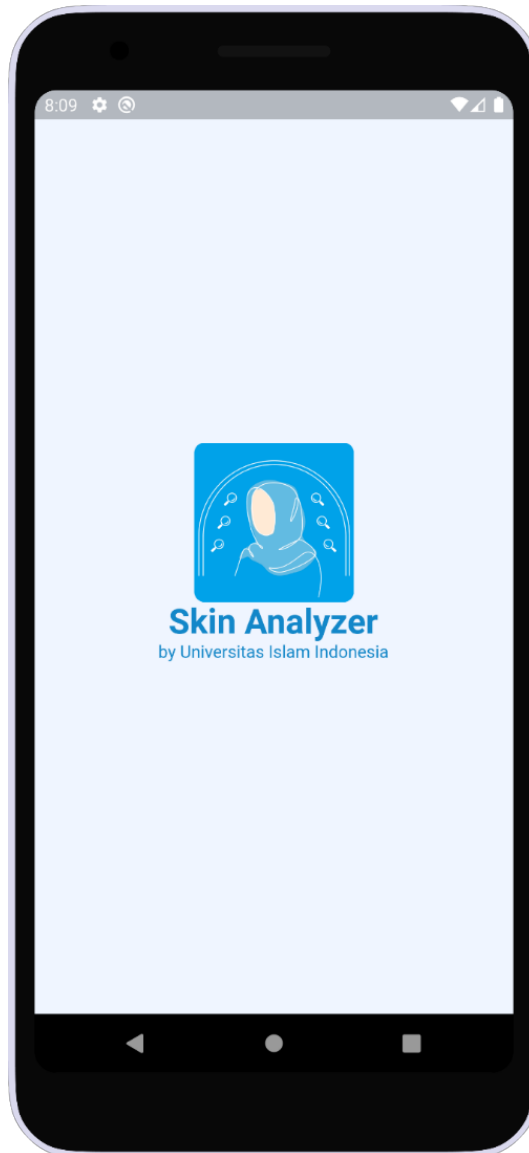
Berdasarkan dari hasil iterasi, *feedback* rancangan, dan temuan saat pengujian *System Usability Scale* (SUS) sehingga diperoleh analisis hasil desain seperti berikut:

- a. Aplikasi UII *Skin Analyzer* mengacu pada peraturan UII Nomor 10 Tahun 2018 tentang penggunaan merek UII.
- b. Aplikasi UII *Skin Analyzer* menggunakan logo yang representatif berupa ilustrasi wajah perempuan menggunakan jilbab.
- c. Aplikasi UII *Skin Analyzer* disarankan untuk menggunakan tema pastel dan feminin.
- d. Aplikasi UII *Skin Analyzer* menggunakan ilustrasi yang sama di setiap fitur analisis permasalahan kulitnya.
- e. Aplikasi UII *Skin Analyzer* menggunakan ilustrasi yang sesuai syariat bahwa perempuan menggunakan jilbab.
- f. Aplikasi UII *Skin Analyzer* menggunakan bahasa Indonesia yang sesuai dengan KBBI dan PUEBI.
- g. Pada halaman *splash screen* logo disusun secara vertikal. Nama aplikasi berada di bawah logo agar *space* bisa dimaksimalkan.
- h. Teks sebaiknya berada di bawah ilustrasi. Pengguna secara natural akan *scan* dari atas ke bawah sedangkan gambar yang relatif dominan berada di bawah teks.
- i. Kurangi penggunaan ikon bergerak (GIF). Gunakan ikon statis karena jika terlalu banyak ikon bergerak akan jadi terlalu ramai. Ikon bergerak (GIF) bisa dipakai untuk tombol yang ingin ditonjolkan seperti tombol Analisis.

- j. Penggunaan *Circular Button* yang terlalu banyak. Di dalam Google Material Design, *Circular Button* mempunyai hierarki tertinggi sedangkan di halaman utama terdapat tiga *Circular Button* berarti tidak sesuai panduan. Dari segi *target size* juga relatif lebih kecil dibandingkan *rectangular rounded corner* sehingga dapat menyulitkan pengguna dalam *tapping*.
- k. Sebaiknya disamakan desain *Home Page* dan desain selamat datang. Pengguna awal mungkin akan bingung ketika *Home Page* pertama kali dibuka dibanding *Home Page* setelah digunakan berbeda.
- l. Beritahu pengguna jika ada layanan yang belum aktif/masih dikembangkan.
- m. Tombol Simpan Hasil analisis sebaiknya taruh di atas. Hal tersebut dikarenakan jika pengguna mengambil *selfie potrait*, tombol Simpan Hasil tidak terlihat. Pengguna tidak akan melihat jika ada tombol Simpan Hasil di bawah.
- n. Bedakan *Text Copy* antara “Terdeteksi 0 jerawat” dengan “Gagal deteksi jerawat”. Pengguna akan bingung dengan maksudnya. (Tidak terdeteksi jerawat karena error atau memang tidak ada jerawat ditemukan).

4.1.2 Halaman Logo

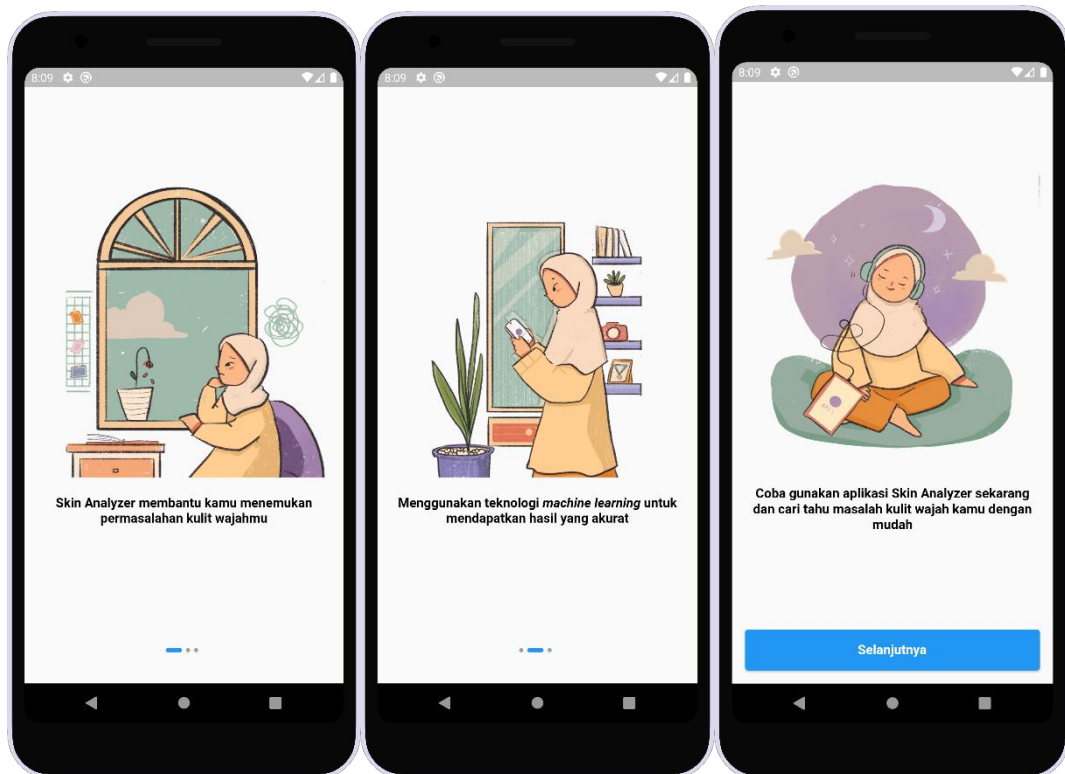
Halaman logo akan menampilkan animasi logo beserta nama aplikasi secara *fade in* dengan *background* berwarna biru muda (lihat Gambar 4.1). Animasi dilakukan dengan total waktu selama 3 detik. 1 detik pertama tidak menampilkan apapun, 1 detik selanjutnya menampilkan animasi *fade in*, dan 1 detik terakhir menampilkan logo dan nama aplikasi. Halaman ini sering diistilahkan sebagai *splash screen*. Dalam pengembangannya, halaman ini menggunakan fitur *StatefulWidget*, *Future*, dan *AnimatedOpacity*. *Future* akan menjalankan kode setelah durasi yang ditentukan, *StatefulWidget* mengubah *state* atau keadaan dari aplikasi, serta *AnimatedOpacity* adalah *Widget* yang digunakan untuk menganimasikan *opacity*. Secara garis besar alur dari kode pada halaman ini adalah ditampilkannya halaman yang kosong terlebih dahulu dengan *opacity* sama dengan 0, setelah satu detik ubah *state* aplikasi sehingga *opacity* menjadi 1, *widget AnimatedOpacity* akan menganimasikan perubahan *opacity* dari yang semula 0 menjadi 1, tunggu selama satu detik, lalu pindah ke halaman selanjutnya.



Gambar 4.1 Hasil halaman Logo

4.1.3 Halaman *Onboarding*

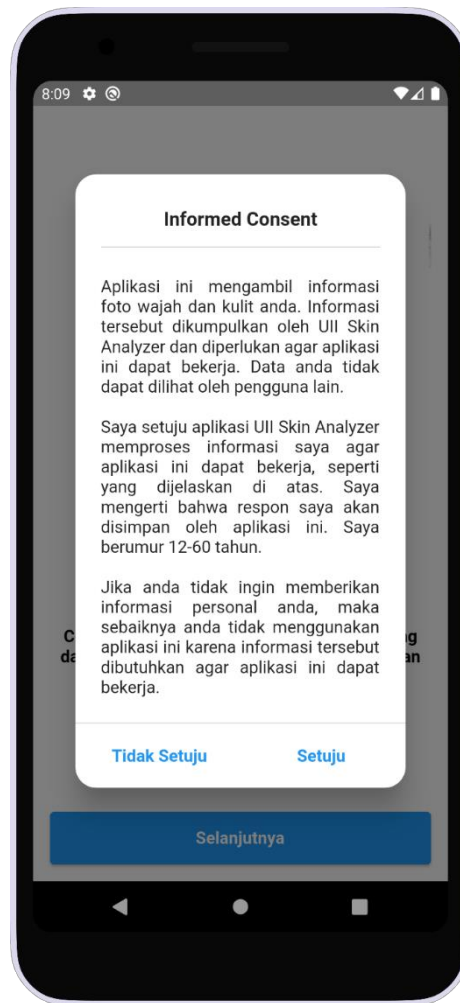
Halaman *onboarding* (seperti pada Gambar 4.2) akan selalu ditampilkan selama pengguna belum pernah memasukkan gambar pada halaman analisis. Halaman ini adalah satu halaman yang terdiri dari 3 tampilan yang berbeda. Pada halaman ini digunakan fitur *StatefulWidget* dan *PageView* pada Flutter. *PageView* digunakan untuk menampilkan tulisan pendek dan gambar di bawahnya, sedangkan *StatefulWidget* digunakan untuk mengubah indikator halaman menjadi tombol pada halaman yang terakhir.



Gambar 4.2 Hasil halaman *Onboarding*

4.1.4 *Pop-up Informed Consent*

Pop-up informed consent ini (lihat Gambar 4.3) menggunakan fitur *showGeneralDialog*, *Transform.scale*, dan *AlertDialog* pada Flutter. *Transform.scale* berguna untuk membuat efek *scale* pada dialog, *AlertDialog* adalah salah satu *widget pop-up* atau *dialog* pada Flutter, dan *showGeneralDialog* adalah fungsi yang dipanggil untuk menampilkan *dialog* pada saat pengguna menekan tombol 'Selanjutnya'.



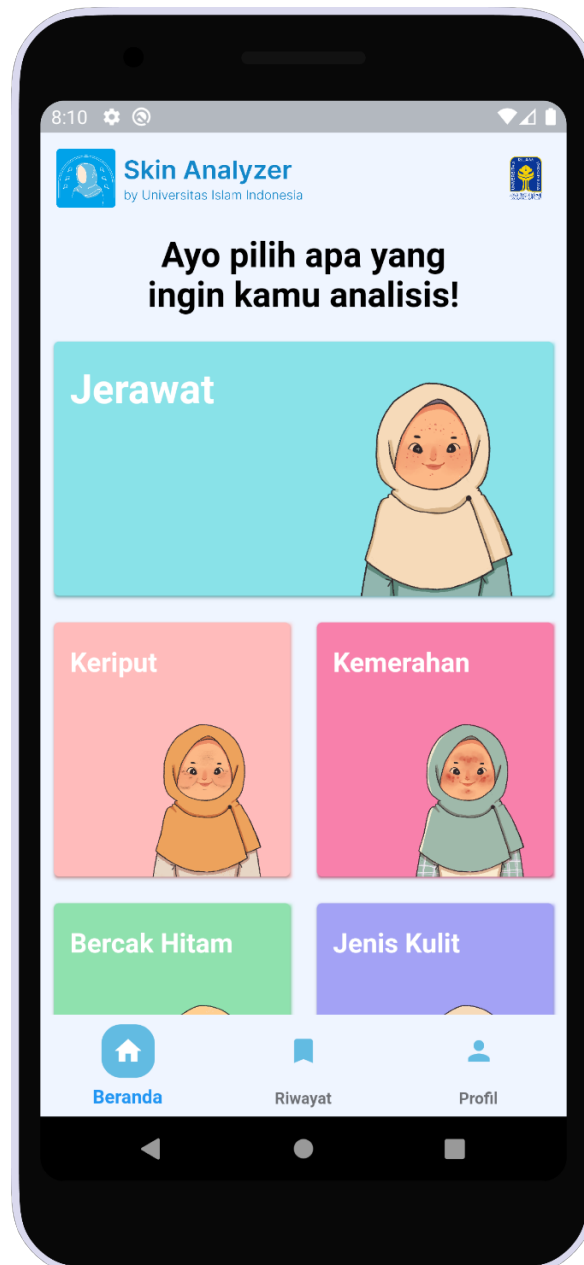
Gambar 4.3 *Pop-up Informed Consent*

4.1.5 Pembungkus Menu Utama

Pembungkus ini terdiri dari *AppBar* yang berisi logo, nama aplikasi, logo UII, *body* yang dapat berubah berdasarkan menu yang dipilih, serta *BottomNavigationBar* untuk mengganti menu yang ditampilkan pada *body*. Untuk mengubah halaman yang ditampilkan pada *body*, digunakan fitur *state* pada Flutter. Halaman menu utama terdiri dari 3 pilihan yaitu Beranda, Riwayat, dan Profil.

4.1.6 Halaman Beranda

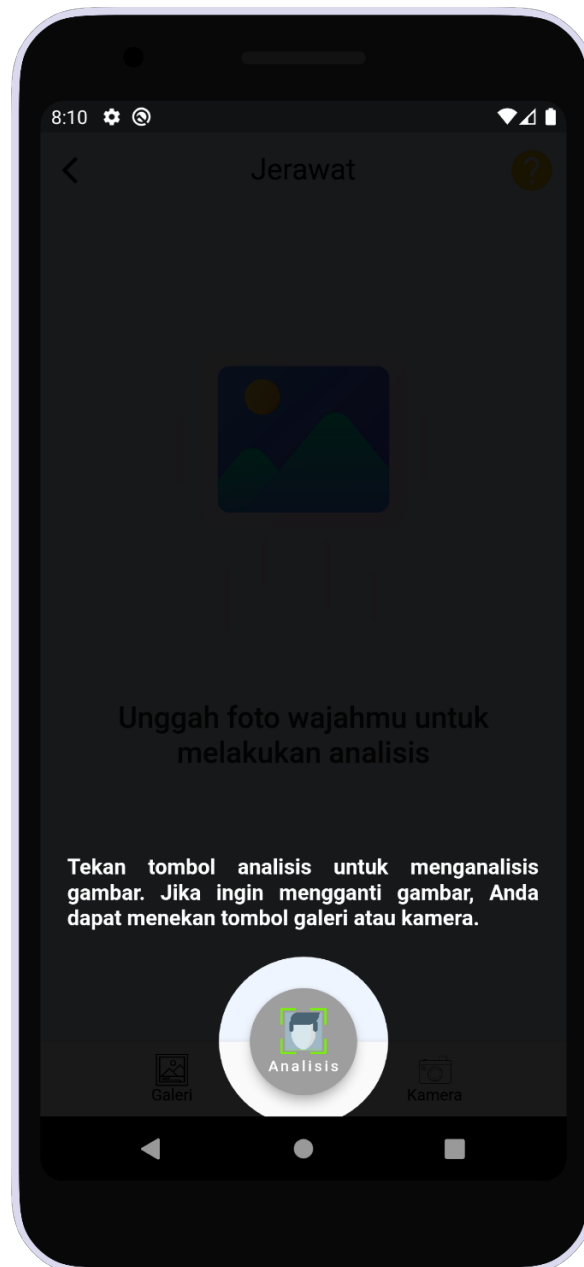
Halaman beranda berisi teks judul serta tombol-tombol menu yang menuju halaman analisis (lihat Gambar 4.4). Pada Flutter, untuk membuat *layout* tampilan digunakan *widget* seperti *Column* dan *Row*. *Column* adalah *widget* yang menampilkan *widget-widget* di dalamnya secara vertikal. Sebaliknya *Row* adalah *widget* yang menampilkan *widget-widget* di dalamnya secara horizontal. Pada halaman beranda digunakan kedua *layout* tersebut.



Gambar 4.4 Hasil halaman Beranda

4.1.7 Halaman Tutorial Menu Analisis

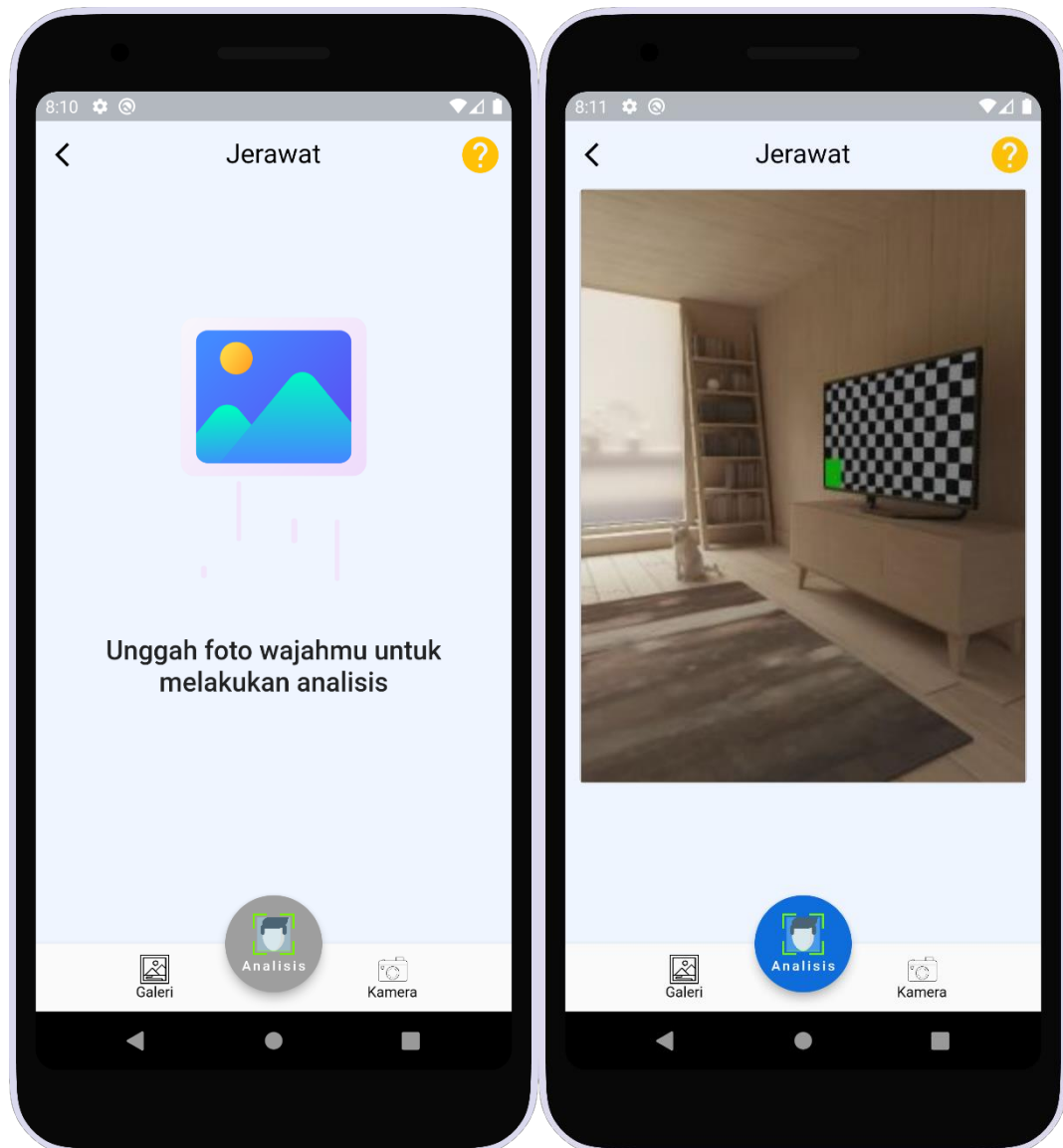
Halaman ini akan ditampilkan saat pengguna belum pernah memasukkan gambar di halaman analisis (lihat Gambar 4.5). Untuk membuat halaman ini, digunakan *package tutorial_coach_mark* oleh RafaelBarbosatec. *Package* tersebut dapat digunakan dengan menginstalnya setelah didaftarkan pada *pubspec.yaml*. Pada halaman ini akan ditampilkan panduan atau tutorial dalam bentuk tulisan dan *highlight* tombol serta terdapat animasi saat proses *highlight* tombol. Pengguna dapat menekan layar di bagian manapun untuk melanjutkan tutorial di halaman ini.



Gambar 4.5 Hasil halaman Tutorial Menu Analisis

4.1.8 Halaman Menu Analisis

Halaman ini memiliki struktur yaitu *AppBar*, *body*, dan *BottomNavigationBar* (lihat Gambar 4.6). Pada *AppBar* ditampilkan tombol kembali, judul halaman, dan tombol bantuan. Pada *body*, akan ditampilkan salah satu hal berikut yaitu pesan untuk memilih gambar atau gambar terakhir yang telah dipilih oleh pengguna. Untuk menentukan apa yang akan ditampilkan dari keduanya, terdapat suatu mekanisme yang akan dijelaskan dengan lebih detail pada subbab selanjutnya. *BottomNavigationBar* berisi tiga tombol di antaranya tombol galeri, analisis, dan kamera.



Gambar 4.6 Hasil halaman Menu Analisis

4.1.9 Mekanisme Pengolahan Gambar Pada Halaman Menu Analisis

Berikut ditampilkan penjelasan serta kode dari proses pengolahan gambar pada halaman menu analisis (lihat Gambar 4.7). Pertama dimulai saat halaman menu analisis dibuka. Pada halaman ini terdapat suatu variabel global dengan nama *_cachedImage*. Variabel ini nantinya yang akan menentukan apakah halaman menu analisis menampilkan gambar terakhir yang dipilih oleh pengguna atau menampilkan pesan untuk memilih gambar. Variabel tersebut akan di-assign saat fungsi *initState* dijalankan, di mana fungsi *initState* akan otomatis dijalankan saat halaman dibuka. Di dalam fungsi *initState* akan dipanggil fungsi *_fetchCachedImage* yang berguna untuk meng-assign variabel global *cachedImage*.

```

File? _cachedImage;

@Override
void initState() {
    _fetchCachedImage();

    super.initState();
}

Future<void> _fetchCachedImage() async {
    final prefs = await SharedPreferences.getInstance();
    final imagePath = prefs.getString('cachedImage');
    if (imagePath == null) {
        return;
    }

    setState(() {
        _cachedImage = File(imagePath);
    });
}

```

Gambar 4.7 Fungsi *_fetchCachedImage*

Pada fungsi *_fetchCachedImage* akan dilakukan pengecekan apakah terdapat gambar yang sudah tersimpan. Pengecekan dilakukan dengan bantuan *package shared_preference* di mana *package* tersebut berguna untuk menyimpan suatu *key value pair*. Pada fungsi *_fetchCachedImage* akan diakses *value* dari *key* 'cachedImage' yang tersimpan pada *package shared_preference*. Kemudian *value* tersebut akan dicek apakah sama dengan *null*, jika *true* maka fungsi akan *return* dan tidak lanjut mengeksekusi baris di bawahnya. Jika *false* yang artinya *value* dari *key* 'cachedImage' tidak sama dengan *null*, *value* tersebut berupa *String path* dari gambar yang dipilih oleh pengguna. Variabel global *_cachedImage* berupa *File* yang ada pada *path* yang sudah tersimpan pada *shared_preference*.

Selanjutnya terdapat fungsi *_pickImage* (seperti pada Gambar 4.8) yang akan dipanggil saat salah satu tombol galeri atau kamera ditekan. Saat menekan tombol galeri, variabel *index* pada fungsi sama dengan 0, sehingga pada blok kode di bawah, akan masuk ke *if* yang pertama. Pada blok kode *if* tersebut, variabel *imageFile* yang sudah dideklarasikan di-*assign* menjadi data yang dikembalikan saat fungsi *picker.pickImage* selesai dieksekusi. Fungsi tersebut

berasal dari *package image_picker*. Kode *await* berfungsi sebagai penanda agar kode setelah baris tersebut tidak dijalankan terlebih dahulu sebelum fungsi *pickImage* selesai dieksekusi.

```
void _pickImage(int index, BuildContext context) async {
    final ImagePicker picker = ImagePicker();
    XFile? imageFile;

    if (index == 0) {
        imageFile = await picker.pickImage(source: ImageSource.gallery);
    } else if (index == 1) {
        imageFile = await Navigator.push<XFile>(
context, MaterialPageRoute(builder: (_) => const CameraScreen()));
    }

    if (imageFile == null) {
        return;
    }

    _cachingImage(imageFile);
}
```

Gambar 4.8 Fungsi *_pickImage*

Saat menekan tombol kamera, variabel *index* sama dengan 1, artinya pada blok kode di atas akan memindahkan tampilan ke *CameraScreen* dikarenakan pada aplikasi ini terdapat fitur *guide* yang akan menampilkan *overlay* yang dapat membantu pengguna saat mengambil gambar melalui kamera. Pada *package image_picker* tidak dapat ditambahkan suatu *overlay*, sehingga digunakan *package camera* oleh pengembang Flutter agar dapat membuat halaman kamera secara lebih leluasa.

Setelah proses pengambilan gambar selesai, maka akan dicek apakah variabel *imageFile* sama dengan *null* yang artinya pengguna tidak memilih gambar, jika *true* maka fungsi akan berhenti sampai di sini. Jika *imageFile* tidak sama dengan *null* yang artinya *false*, maka akan dipanggil fungsi *_cachingImage* yang berguna untuk menyimpan gambar yang telah dipilih oleh pengguna.

```
void _cachingImage(XFile imageFile) async {
    if (_cachedImage != null) {
```

```

        await _cachedImage!.delete();
        _cachedImage = null;
    }

    final appDir = await syspaths.getApplicationDocumentsDirectory();
    final fileName = path.basename(imageFile.path);
    final finalPath = '${appDir.path}/${fileName}';

    File? convertedFile = File(imageFile.path);
    File? savedImage = await convertedFile.copy(finalPath);

    await convertedFile.delete();
    convertedFile = null;

    final prefs = await SharedPreferences.getInstance();
    await prefs.setString('cachedImage', savedImage.path);

    setState(() {
        _cachedImage = savedImage;
        _isServerError = false;
        _isUploadingImage = false;
        _jerawatData = [];
        _jerawatCount = 0;
    });
}

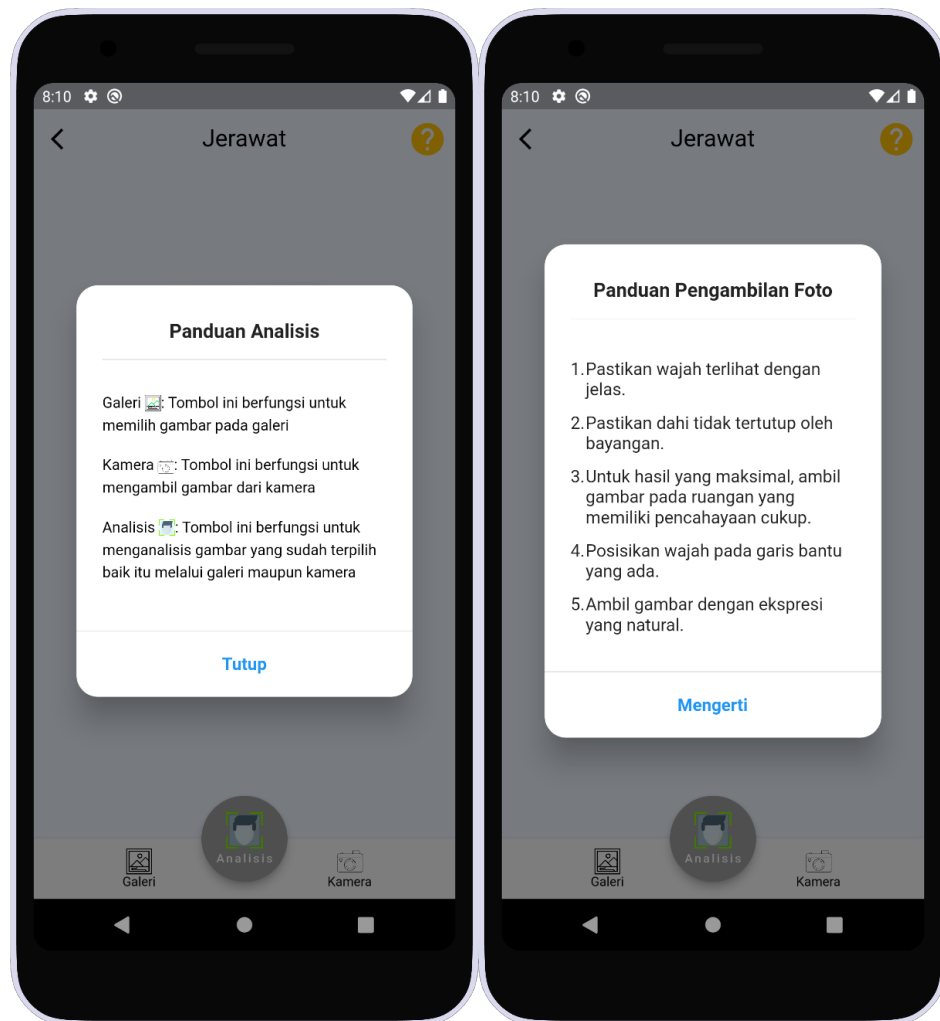
```

Gambar 4.9 Fungsi `_cachingImage`

Pada fungsi `_cachingImage` (perhatikan Gambar 4.9) pertama-tama akan dilakukan pengecekan apakah variabel global `_cachedImage` tidak sama dengan `null`. Jika `true` maka `File` gambar tersebut akan dihapus dan selanjutnya akan diganti oleh `File` gambar yang terbaru. Setelah pengecekan, kode akan mendapatkan `path` dari gambar yang telah dipilih oleh pengguna, di mana akan dijalankan fungsi `copy` yang akan meng-copy `File` pada `path` yang telah ditentukan. Proses peng-copy-an ini dilakukan untuk menyimpan gambar yang dipilih pada perangkat pengguna, tidak hanya pada variabel `File` yang akan hilang saat aplikasi dikeluarkan. Setelah proses peng-copy-an selesai, maka akan disimpan `path` dari gambar tersebut dengan key 'cachedImage' pada `shared_preference`.

4.1.10 *Pop-up* Bantuan dan Panduan Pengambilan Foto

Pop-up bantuan dan panduan pengambilan foto ini (lihat Gambar 4.10) menggunakan fitur *showGeneralDialog*, *Transform.scale*, dan *AlertDialog* pada Flutter. *Transform.scale* berguna untuk membuat efek *scale* pada dialog, *AlertDialog* adalah salah satu *widget pop-up* atau *dialog* pada Flutter, dan *showGeneralDialog* adalah fungsi yang dipanggil untuk menampilkan *dialog* ‘Pesan’ pada saat tombol ‘Bantuan’ ditekan dan menampilkan *dialog* ‘Panduan’ pada saat tombol kamera ditekan.

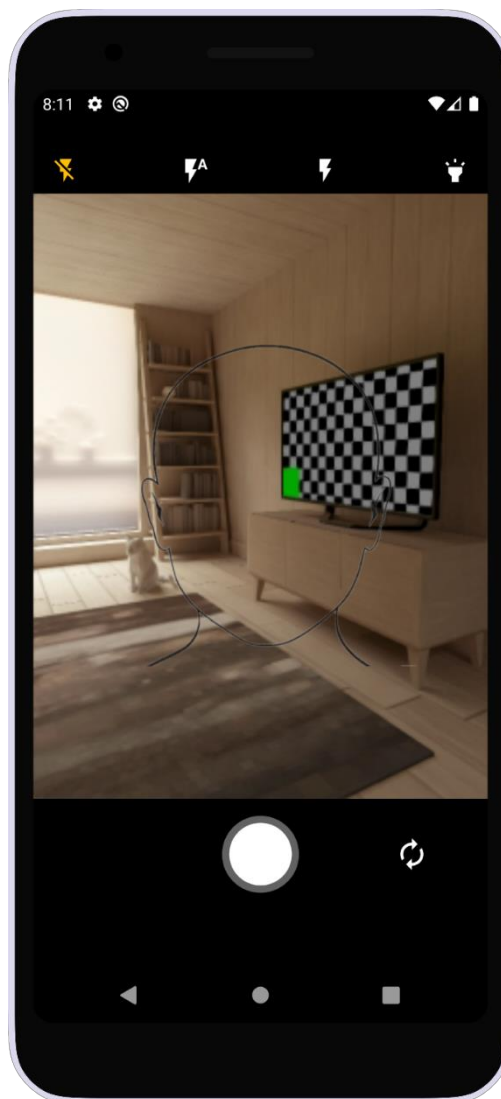


Gambar 4.10 *Pop-up* Bantuan dan Panduan Pengambilan Foto

4.1.11 Halaman Kamera

Halaman kamera akan ditampilkan setelah pengguna menekan tombol ‘Mengerti’ pada *pop-up* panduan pengambilan gambar. Halaman kamera akan menampilkan tombol pengaturan *flash*, kamera itu sendiri, tombol ambil gambar, dan *rotate* kamera (lihat Gambar 4.11).

Untuk memudahkan pengguna dalam mengambil gambar dan menaikkan tingkat akurasi hasil analisis maka ditambahkan suatu *overlay* pada saat pengguna ingin mengambil gambar melalui kamera. Untuk membuat hal tersebut, digunakan *package camera* yang dikembangkan oleh pengembang Flutter. Pada *package* tersebut diberi keleluasaan untuk membuat suatu *custom* halaman kamera. Untuk proses pengambilan gambar dengan *package* tersebut terinspirasi dari salah satu artikel pada [LogRocket](#). Pada artikel tersebut juga dicontohkan bagaimana jika ingin menambahkan suatu gambar di atas tampilan kamera yang dapat berfungsi sebagai *overlay*.



Gambar 4.11 Halaman Kamera

4.1.12 Mekanisme Analisis

Saat variabel global `_cachedImage` tidak sama dengan `null` atau sudah terdapat gambar yang dimasukkan maka pengguna dapat melakukan analisis. Untuk melakukan analisis,

pengguna dapat langsung menekan tombol analisis. Pada saat tombol analisis ditekan, fungsi `_uploadImage` akan dijalankan (seperti pada Gambar 4.12). Untuk mengirim gambar ke API digunakan *package http*. Fungsi `_uploadImage` melakukan beberapa hal di antaranya, mengubah *state* `_isUploadingImage` menjadi *true* untuk menampilkan *loader* dan menonaktifkan tombol analisis dan memilih gambar, mengirim data gambar yang ada pada variabel global `_cachedImage` ke API untuk dianalisis, menjalankan fungsi yang mengolah data kembalian dari API menjadi objek agar mudah digunakan, menghitung jumlah jerawat yang terdeteksi, dan *men-set state* dari data-data yang didapatkan di antaranya data jerawat, data jumlah jerawat, data apakah *server error* dan data selesainya proses *upload*. Data-data tersebut adalah data global, yang nantinya digunakan untuk menampilkan sesuatu di layar.

```
void _uploadImage() async {
  setState(() {
    _isUploadingImage = true;
  });

  File convertedImage = File(_cachedImage!.path);
  var jerawatCount = 0;

  var request = http.MultipartRequest(
    'POST',
    Uri.parse("acne-api/deteksi_jerawat"),
  );

  request.files.add(
    http.MultipartFile(
      'file[]',
      convertedImage.readAsBytes().asStream(),
      convertedImage.lengthSync(),
      filename: convertedImage.path,
    ),
  );

  try {
    var res = await request.send();
    http.Response response = await http.Response.fromStream(res);

    var jsonData = json.decode(response.body);
```

```

JSONResult result = JSONResult.fromJson(jsonData[0]);

for (int i = 0; i < result.deteksiObjek!.jerawat!.length; i++) {
    if (result.deteksiObjek!.jerawat![i].score! > 0.3) {
        jerawatCount++;
    }
}

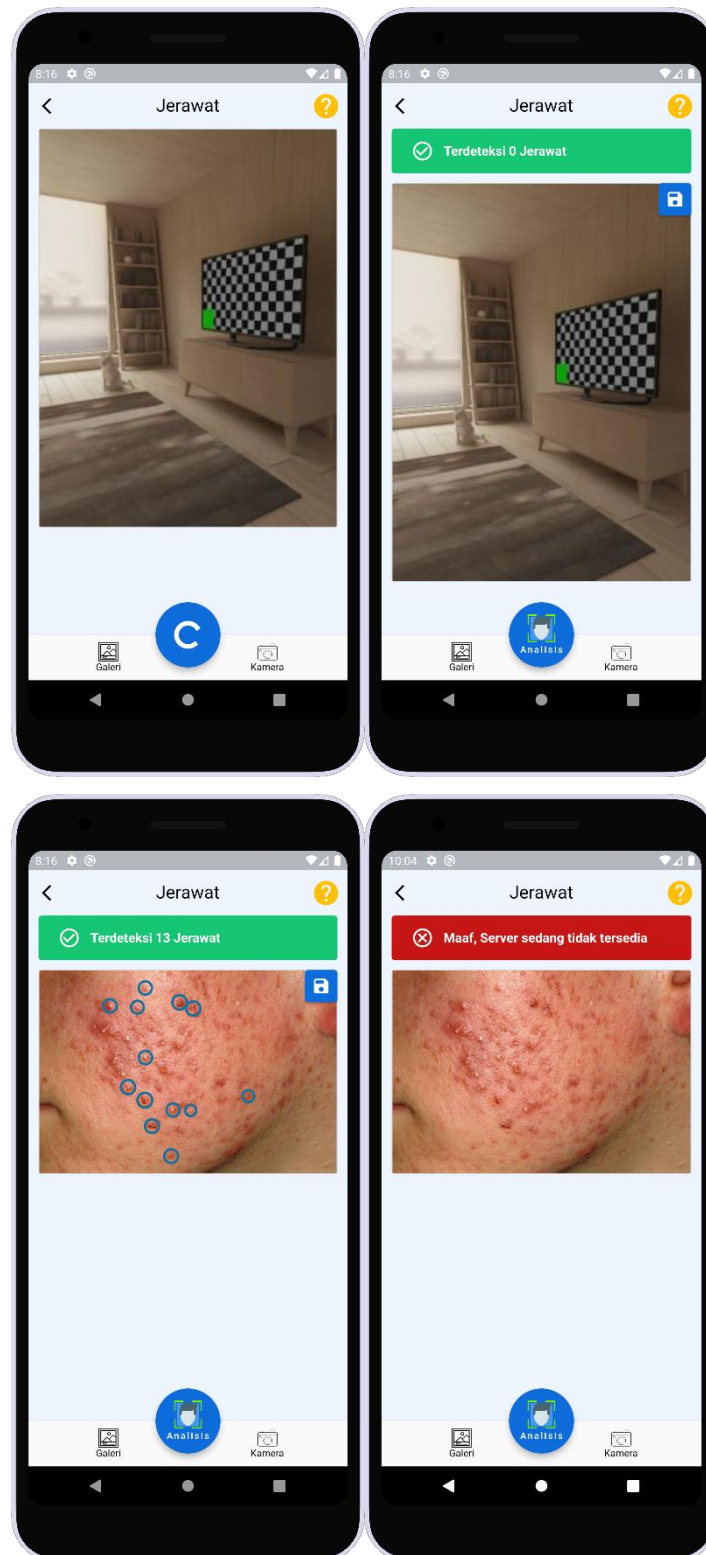
setState(() {
    _isUploadingImage = false;
    _isServerError = false;
    _jerawatData = [...(result.deteksiObjek!.jerawat)!];
    _jerawatCount = jerawatCount;
});
} catch (e) {
    setState(() {
        _isUploadingImage = false;
        _isServerError = true;
        _jerawatData = [];
        _jerawatCount = 0;
    });
}
}

```

Gambar 4.12 Fungsi `_uploadImage`

4.1.13 Halaman Hasil Analisis

Halaman hasil analisis akan ditampilkan setelah pengguna menekan tombol analisis. Sesaat setelah pengguna menekan tombol analisis, tombol-tombol akan dinonaktifkan, dan akan ditampilkan loader seperti pada Gambar 4.13. Setelah hasil analisis didapatkan dari *server*, maka aplikasi akan men-*set state* seperti pada subbab sebelumnya yang akan mengubah tampilan berdasarkan hasil analisis. Jika terdapat error pada *server* saat proses analisis, maka akan ditampilkan pesan *error*, dan jika berhasil maka akan ditampilkan pesan berhasil serta akan ditampilkan suatu tanda pada gambar yang dianalisis, seperti saat menganalisis jerawat maka akan ditampilkan lingkaran di sekitar area jerawat yang terdeteksi. Selain itu, jika analisis berhasil dilakukan akan ditampilkan tombol simpan seperti pada gambar.



Gambar 4.13 Halaman Hasil Analisis

4.1.14 Mekanisme Penanda Gambar Hasil Analisis

Gambar yang sudah dideteksi akan ditandai dengan penanda sesuai menu yang dipilih. Untuk membuat fitur tersebut, digunakan *widget CustomPaint* pada Flutter. Pada *widget*

CustomPaint (lihat Gambar 4.14) terdapat atribut *foregroundPainter* yang berfungsi sebagai penanda. *ForegroundPainter* menerima objek dengan tipe *CustomPainter* di mana pada kode di bawah kelas *DrawInCanvas* meng-extend *CustomPainter*. Dibuatnya kelas tersebut agar dapat digunakan sebagai penanda hasil analisis jerawat, keriput, kemerahan, dan bercak hitam dalam satu kelas. Kelas *DrawInCanvas* menerima warna, lebar penanda, dan objek yang berisi data-data koordinat yang harus ditandai.

```
CustomPaint(
  foregroundPainter: DrawInCanvas(
    color: canvasColor,
    strokeWidth: strokeWidth,
    objects: objectData,
  ),
  child: Stack(
    children: [
      Container(
        width: double.infinity,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(4),
          border: Border.all(
            color: const Color(0xFFCACACA),
          ),
        ),
        child: Image.file(
          imageFile!,
          fit: BoxFit.contain,
        ),
      ),
      saveResultButton,
    ],
  ),
)
```

Gambar 4.14 *Widget CustomPaint*

Berikut ditampilkan isi dari fungsi *paint* pada kelas *CustomPainter* (lihat Gambar 4.15), fungsi ini yang berguna untuk menampilkan penanda. Di dalam fungsi ini, akan dicek apakah objek memiliki points atau titik-titik. Data kembalian dari API untuk fitur analisis keriput dan

kemerahan berupa titik-titik, sedangkan untuk fitur analisis jerawat dan bitnik hitam berupa titik-titik untuk membuat lingkaran. Pada kode di bawah, jika kelas *DrawInCanvas* dijalankan pada halaman analisis keriput dan kemerahan maka variabel *objects* akan memiliki data *points* sehingga akan masuk ke *if* blok yang pertama. Berdasarkan data *points* tersebut akan ditampilkan titik-titik yang menandakan area keriput atau kemerahan yang terdeteksi. Sedangkan pada fitur analisis jerawat dan bintik hitam, variabel *objects* tidak memiliki data *points* sehingga akan masuk ke *if* blok yang kedua dan berdasarkan rumus yang didapatkan pada penelitian sebelumnya akan ditampilkan lingkaran pada sekeliling jerawat atau bintik hitam yang terdeteksi.

```
@override
void paint(Canvas canvas, Size size) {
  var paint1 = Paint()
    ..color = color
    ..strokeWidth = strokeWidth
    ..style = PaintingStyle.stroke;

  for (var object in objects) {
    if (object.points != null) {
      List<Offset> points = [];

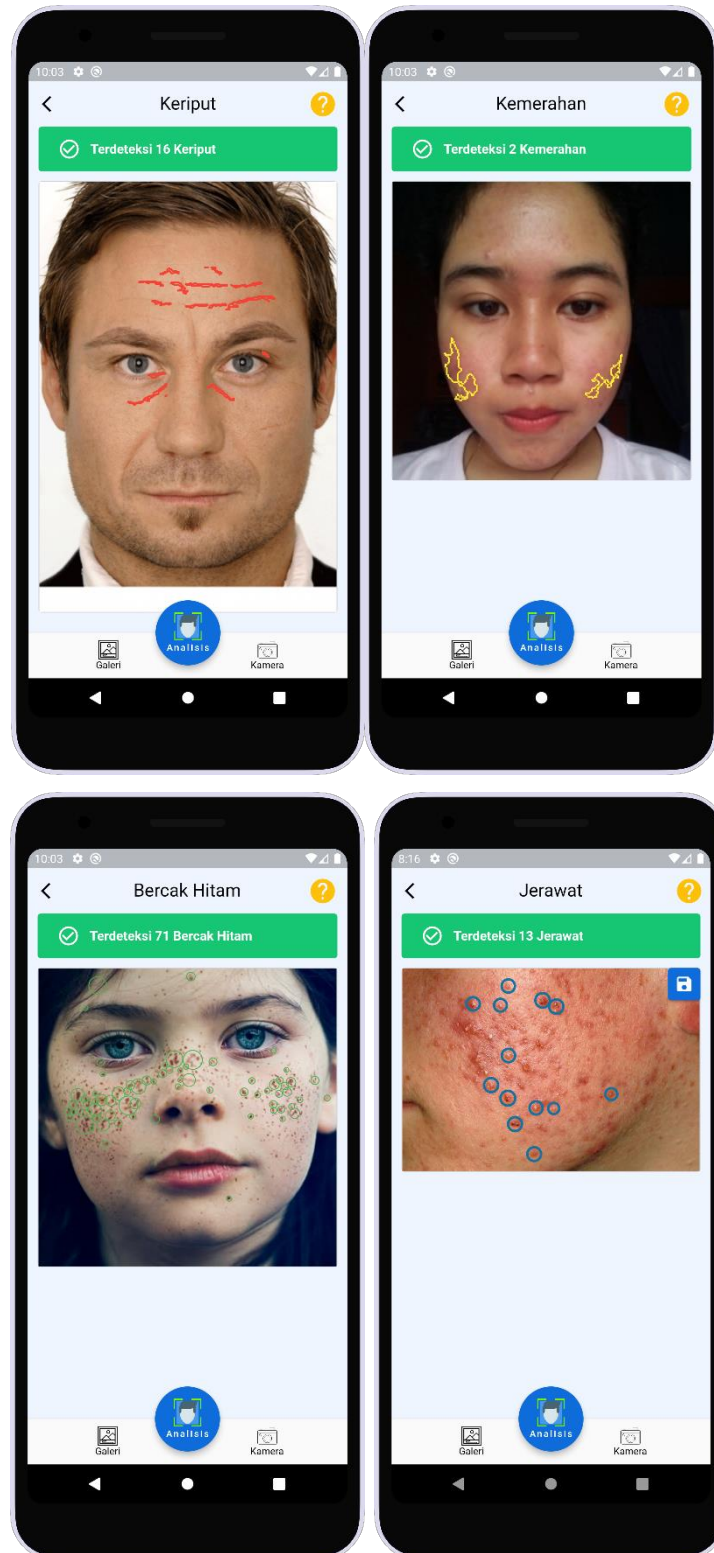
      for (var point in object.points!) {
        var dx = size.width * point[0];
        var dy = size.height * point[1];
        points.add(Offset(dx, dy));
      }

      canvas.drawPoints(PointMode.points, points, paint1);
    } else if (object.score! > 0.3) {
      var ymin = object.ymin;
      xmin = object.xmin;
      ymax = object.ymax;
      xmax = object.xmax;
      var left = xmin! * size.width;
      var right = xmax! * size.width;
      var top = ymin! * size.height;
      var bottom = ymax! * size.height;
      var boxH = bottom - top, boxW = right - left;
```

```
var radius = sqrt(pow(boxH, 2) + pow(boxW, 2)) / 2;

canvas.drawCircle(
    Offset(
        left + boxW / 2,
        top + boxH / 2,
    ),
    radius,
    paint1
);
}
}
```

Gambar 4.15 Fungsi *paint* Pada Kelas *DrawInCanvas*

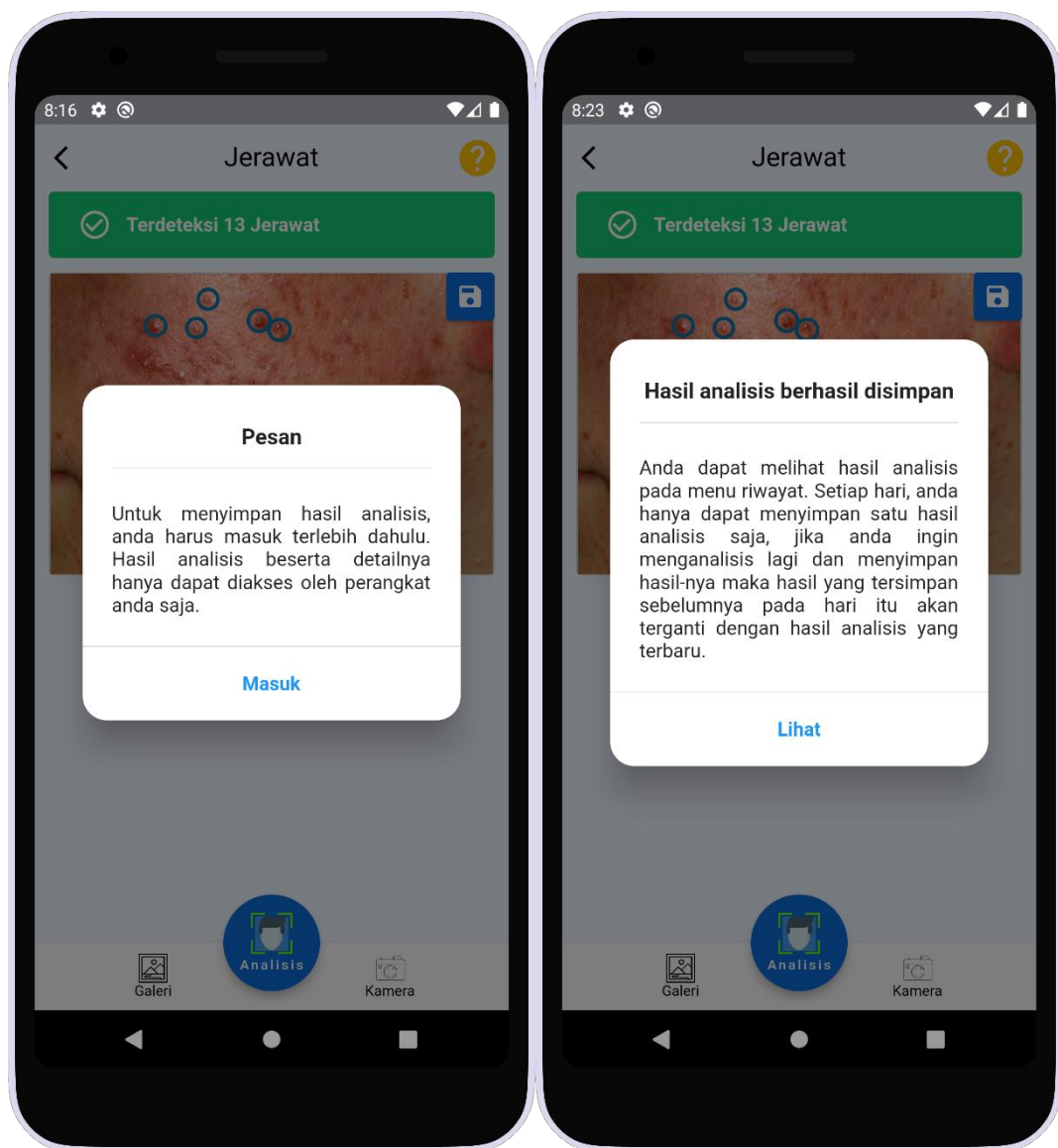


Gambar 4.16 Halaman Hasil Analisis Berdasarkan Fitur

4.1.15 Mekanisme Menyimpan Gambar Hasil Analisis

Saat pengguna menekan tombol simpan setelah analisis berhasil dilakukan, maka akan dilakukan pengecekan terlebih dahulu apakah pengguna sudah *login* pada aplikasi. Jika belum maka akan ditampilkan *pop-up* yang berisi pesan untuk *login* dan tombol ‘Masuk’ yang akan mengarahkan langsung ke halaman profil seperti yang dapat dilihat pada Gambar 4.17.

Jika pengguna sudah *login* maka proses penyimpanan gambar baru dapat dilakukan. Dalam menjaga privasi pengguna, gambar hasil analisis hanya tersimpan pada aplikasi UII *Skin Analyzer*, pengguna sendiri tidak dapat melihat hasil analisisnya pada galeri perangkat masing-masing. Untuk membuat fitur tersebut, digunakan *package sqflite* oleh Tekartik sebagai *package* yang memungkinkan untuk menggunakan *database sql* pada aplikasi Flutter.



Gambar 4.17 *Pop-up* Simpan Hasil Analisis

Di bawah ditampilkan kode fungsi `_saveToDB` (lihat Gambar 4.18) yang akan dijalankan saat pengguna sudah *login* dan menekan tombol simpan hasil analisis. Secara garis besar alur dari fungsi di bawah adalah fungsi akan melakukan pengecekan apakah pada hari tersebut pengguna sudah pernah menyimpan hasil analisisnya. Jika sudah pernah menyimpan maka data hasil analisis sebelumnya akan dihapus karena diasumsikan bahwa pengguna hanya dapat menyimpan satu hasil analisis dalam satu hari. Fitur tersebut ada karena untuk membuat grafik hasil analisis yang lebih valid pada halaman Riwayat nantinya. Setelah menghapus data sebelumnya jika hari itu sudah pernah menyimpan maka selanjutnya adalah menyimpan data hasil analisis pada hari itu. Mengambil data untuk pengecekan dan menyimpan data pada *database sql* dilakukan dengan bantuan kelas *DBHelper* (lihat Gambar 4.19) yang pada kelas tersebut terdapat berbagai macam fungsi untuk berinteraksi dengan *database sql*.

```
Future<void> _saveToDB() async {
  setState(() {
    _isUploadingImage = true;
  });
  final dateNow = DateTime(
    DateTime.now().year, DateTime.now().month, DateTime.now().day);
  final date = '
    ${DateTime.now().year}${DateTime.now().month}${DateTime.now().day}';

  final dateFormat = DateFormat.yMd().format(dateNow);
  final jsonData = json.encode(_jerawatData);

  final appDir = await syspaths.getApplicationDocumentsDirectory();
  final Directory appDocDirFolder = Directory(
    '${appDir.path}/jerawat_history/');
  final Directory appDocDirNewFolder = await appDocDirFolder.create();

  final fileName = path.basename(_imageFile!.path);
  final finalPath = '${appDocDirNewFolder.path}$fileName$date';

  File savedImage = await _imageFile!.copy(finalPath);

  var searchDay = await DBHelper.getSingleData(
    'analysis_results',
    ['id'],
```

```

        '$dateFormat${auth.currentUser!.email!}',
    );

    if (searchDay.isNotEmpty) {
        var searchFile = await DBHelper.getSingleData(
            'analysis_results',
            ['id', 'image_path'], '$dateFormat${auth.currentUser!.email!}');
        var searchedImage = searchFile[0]['image_path'];

        if (searchedImage != savedImage.path
            && File(searchedImage).existsSync()) {
            File? file = File(searchedImage);
            await file.delete();
            file = null;
        }

        await DBHelper.insert('analysis_results', {
            'id': '$dateFormat${auth.currentUser!.email!}',
            'email': auth.currentUser!.email!,
            'image_path': savedImage.path,
            'jerawat_result': jsonData,
            'date': dateNow.toString(),
        });

        setState(() {
            _isUploadingImage = false;
        });
    }
}

```

Gambar 4.18 Fungsi `_saveToDB`

```

import 'package:sqflite/sqflite.dart' as sql;
import 'package:path/path.dart' as path;
import 'package:sqflite/sqlite_api.dart';

class DBHelper {
    static Future<Database> database() async {
        final dbPath = await sql.getDatabasesPath();
        return sql.openDatabase(
            path.join(dbPath, 'skin_analyzer.db'), onCreate: (db, version) {
                return db.execute(

```

```

        'CREATE TABLE analysis_results(id TEXT PRIMARY KEY NOT NULL,
            email TEXT NOT NULL, image_path TEXT NOT NULL,
            jerawat_result TEXT, keriput_result TEXT,
            kemerahan_result TEXT, bercak_hitam_result TEXT,
            jenis_kulit_result TEXT, date TEXT NOT NULL) ');
    }, version: 1);
}

static Future<void> insert(String table,
    Map<String, Object> data) async {
    final db = await DBHelper.database();
    db.insert(
        table, data, conflictAlgorithm: ConflictAlgorithm.replace,
    );
}

static Future<List<Map<String, dynamic>>> getData(String table) async {
    final db = await DBHelper.database();
    return db.query(table);
}

static Future<List<Map<String, dynamic>>> getSingleData(
    String table, List<String> column, String whereArgs) async {
    final db = await DBHelper.database();
    return db.query(table, columns: column,
        where: '${column[0]} = ?', whereArgs: [whereArgs]);
}
}

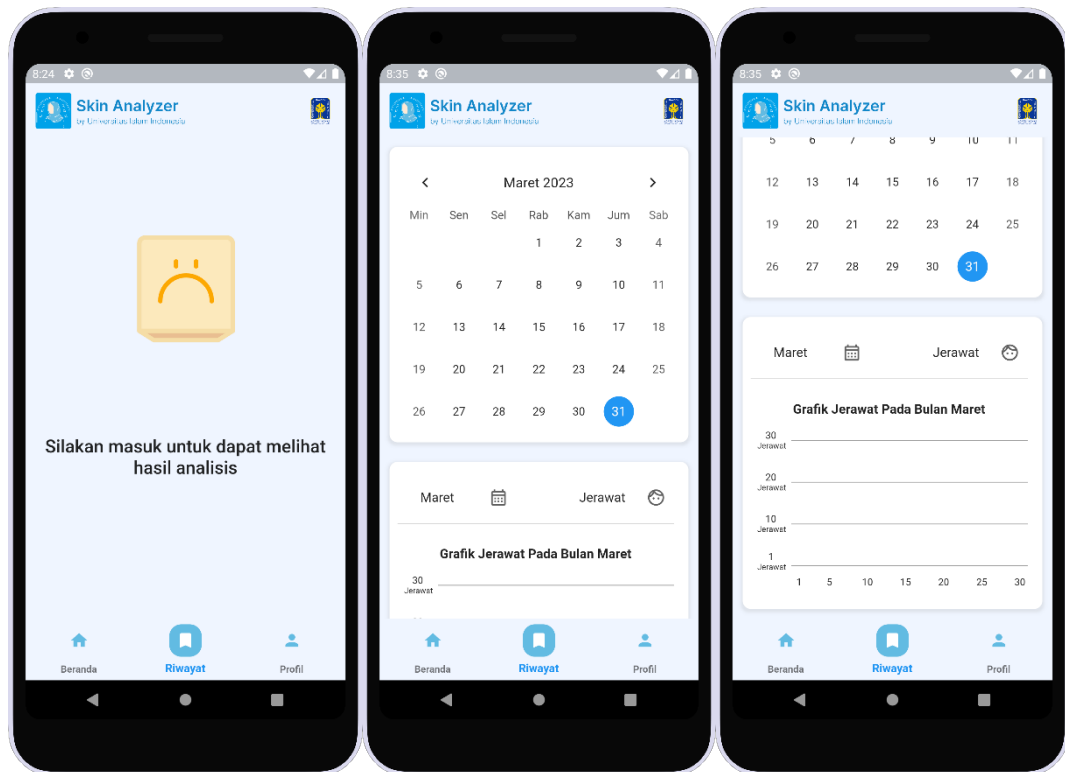
```

Gambar 4.19 Kelas *DBHelper*

4.1.16 Halaman Riwayat

Pada halaman pembungkus yang berisi tiga halaman utama salah satunya ialah riwayat (seperti pada Gambar 4.20). Jika ingin ke halaman riwayat maka pengguna dapat langsung menekan tombol ‘Riwayat’ pada *BottomNavigationBar* sehingga *body* akan terganti dan menampilkan halaman Riwayat. Halaman ini berisi kalender perbulan dan grafik hasil analisis. Untuk fitur kalender digunakan *package table_calendar* oleh [aleksanderwozniak](#) dan fitur grafik menggunakan *package fl_chart* oleh [imaNNeo](#). Pada halaman ini dilakukan *fetch* atau pengambilan data hasil analisis yang tersimpan pada *database* yang kemudian akan diolah oleh

widget kalender dan *widget* grafik (pada Gambar 4.21). Sebelum diolah, data hasil analisis harus di-*filter* terlebih dahulu agar pengguna hanya dapat melihat riwayat hasil analisis yang sesuai dengan akun yang *login* saja. Fitur ini akan dibahas lebih lanjut pada subbab halaman profil.



Gambar 4.20 Halaman Riwayat

```
void _fetchAndSetAnalysisHistory() async {
  if (auth.currentUser == null) {
    return;
  }

  setState(() {
    isFetching = true;
  });

  final rawData = await DBHelper.getData('analysis_results');

  List<AnalysisHistory> rawAnalysisHistory = rawData
    .map(
      (data) => AnalysisHistory(
```

```

        id: data['id'],
        email: data['email'],
        imagePath: data['image_path'],
        jerawatResult: data['jerawat_result'],
        keriputResult: data['keriput_result'],
        kemerahanResult: data['kemerahan_result'],
        bercakHitamResult: data['bercak_hitam_result'],
        jenisKulitResult: data['jenis_kulit_result'],
        date: data['date'],
    ),
)
.toList();

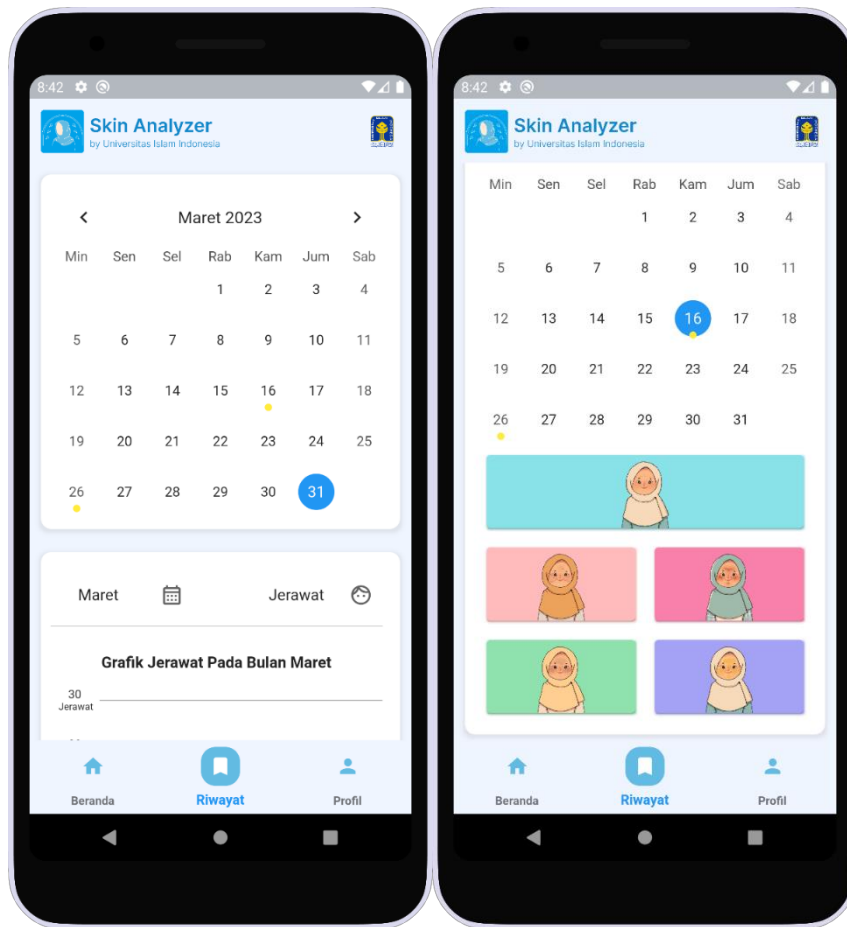
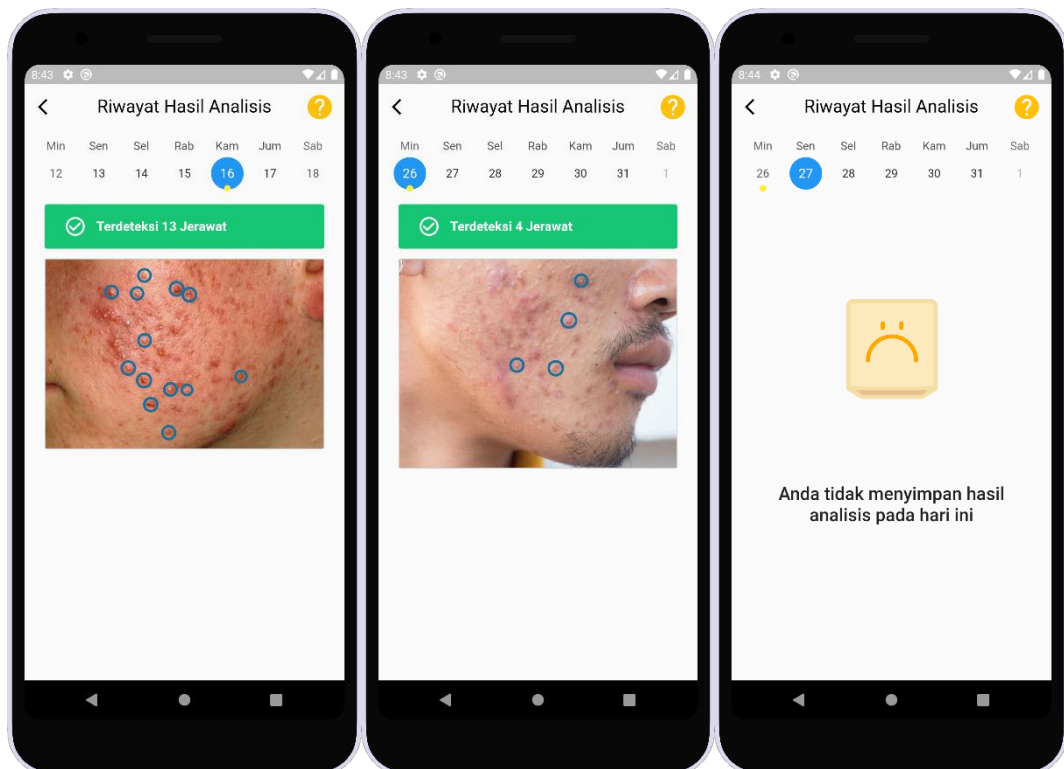
setState(() {
    isFetching = false;
    analysisHistory = rawAnalysisHistory.where(
        (obj) => obj.email == auth.currentUser!.email).toList();
});
}

```

Gambar 4.21 Fungsi `_fetchAndSetAnalysisHistory`

4.1.17 Mekanisme Kalender Hasil Analisis

CalendarCardWidget adalah *widget* yang menampilkan kalender hasil analisis (lihat Gambar 4.22). Di dalamnya terdapat *widget Container* sebagai pembungkus sekaligus untuk mengatur *padding*, *margin*, *border radius*, dan *box shadow* agar menampilkan efek seperti *card*. Di dalam *Container* terdapat *Column* yang berisi *widget* kalender itu sendiri dan *widget ValueListenableBuilder* yang dapat mengawasi atau *listen* saat apakah pada hari yang dipilih oleh pengguna di *widget* kalender terdapat hasil analisis. Jika *true* maka akan ditampilkan tombol yang mengarah ke halaman detail hasil analisis. Jika *false* maka tidak akan ditampilkan apa-apa. Untuk saat ini fitur Riwayat hanya terdapat pada menu analisis jerawat (lihat Gambar 4.23), dikarenakan menu analisis lainnya hanya dapat menganalisis satu gambar saja. Jika tombol jerawat ditekan, maka pengguna akan diarahkan ke halaman yang berisi gambar yang sudah ditandai serta *card* yang berisi jumlah jerawat yang terdeteksi pada hari yang dipilih pada kalender. Pada halaman detail juga terdapat kalender horizontal di mana pengguna dapat langsung melihat hasil analisis pada hari lainnya tanpa perlu keluar halaman terlebih dahulu.

Gambar 4.22 *Widget Kalender*

Gambar 4.23 Halaman Detail Hasil Analisis

Pada Gambar 4.24 di bawah ditampilkan kode utama dari *CalendarCardWidget*. Terdapat widget *ValueListenableBuilder* yang mendengarkan perubahan dari *value* atau variabel *_selectedEvent*. Variabel *_selectedEvent* akan berubah pada saat fungsi *_onChangeDate* dipanggil, di mana fungsi tersebut dijadikan atribut pada *CalendarWidget* sehingga nantinya pada widget kalender tersebut akan dilakukan pemanggilan fungsi *_onChangeDate* pada saat pengguna menekan hari yang berbeda. Saat fungsi *_onChangeDate* berjalan maka variabel *_selectedEvent* akan berubah menjadi nilai kembalian dari fungsi *getEventForDay*, fungsi ini terdapat pada *CalendarWidget* yang akan mengembalikan data hasil analisis pada hari yang dipilih pada kalender.

```
void _onChangeDate(DateTime selectedDay,
    List<AnalysisHistory> Function(DateTime) getEventForDay) {
    _selectedEvent.value = getEventForDay(selectedDay);
}

Container(
    width: MediaQuery.of(context).size.width,
    margin: const EdgeInsets.all(12.0),
    padding: const EdgeInsets.all(12.0),
    decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: const BorderRadius.all(Radius.circular(10)),
        boxShadow: [
            BoxShadow(
                color: Colors.grey.withOpacity(0.2),
                spreadRadius: 2,
                blurRadius: 2,
                offset: const Offset(0, 2),
            ),
        ],
    ),
    child: Column(
        children: [
            CalendarWidget(
                analysisHistory: analysisHistory,
                onChangeDate: _onChangeDate,
                calendarFormat: CalendarFormat.month,
```

```

    ),
    ValueListenableBuilder<List>(
        valueListenable: _selectedEvent,
        builder: (ctx, value, _) {
            return (value.isEmpty)
                ? const SizedBox()
                : Column(
                    children: [
                        _buildButton(
                            imagePath:
                                "assets/images/home_illustrations/jerawat.png",
                            color: const Color(0xFF89E2E8),
                            onPressed: () {
                                Navigator.of(context).push(MaterialPageRoute(
                                    builder: (_) => JerawatHistoryScreen(
                                        selectedAnalysisHistory:
                                            _selectedEvent.value,
                                        analysisHistory: analysisHistory,
                                    )),
                                );
                            },
                            buttonWidth: 1,
                            context: context,
                        ),
                    ],
                ),
    ),
    ..
    ..
    ..

```

Gambar 4.24 Kode *CalendarCardWidget*

Kode di bawah (lihat Gambar 4.25) menampilkan kode utama dari *CalendarWidget*. Untuk menampilkan kalender digunakan *widget TableCalendar* yang berasal dari *package table_calendar*. Pada *widget* tersebut terdapat atribut *onDaySelected*, fungsi di atribut ini akan dijalankan saat pengguna menekan hari pada kalender. Pada fungsi ini juga akan dijalankan fungsi *onChangeDate* yang berasal dari atribut yang ada pada *CalendarWidget*, di mana fungsi tersebut berguna untuk mengubah nilai dari variabel *selectedEvent*.

```

List<AnalysisHistory> _getEventForDay(DateTime day) {
    List<AnalysisHistory> analysisHistory = [];
    for (var element in widget.analysisHistory) {

```

```

        if (!isSameDay(DateTime.parse(element.date), day)) continue;

        analysisHistory.add(element);
    }

    return analysisHistory;
}

TableCalendar(
  onDaySelected: (selectedDay, focusedDay) {
    if (!isSameDay(_selectedDay, selectedDay)) {
      setState(() {
        _selectedDay = selectedDay;
        _focusedDay = focusedDay;
      });

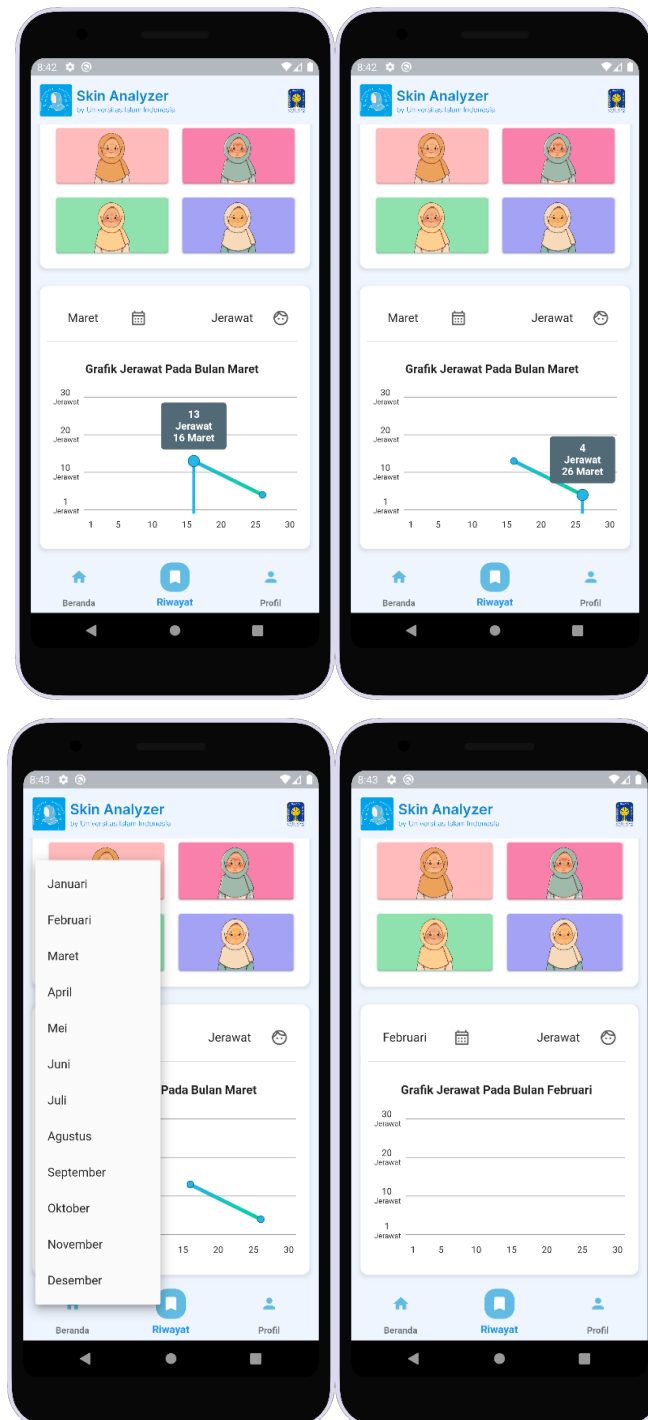
      widget.onChangeDate(selectedDay, _getEventForDay);
    }
  },
  ..
  ..
  ..

```

Gambar 4.25 Kode *CalendarWidget*

4.1.18 Mekanisme Grafik Hasil Analisis

ChartCardWidget adalah *widget* yang menampilkan grafik hasil analisis (pada Gambar 4.26). Di dalamnya terdapat *widget Container* sebagai pembungkus sekaligus untuk mengatur *padding*, *margin*, *border radius*, dan *box shadow* agar menampilkan efek seperti card. Di dalam *Container* terdapat *Column* berisi *Row* dengan dua *widget dropdown* untuk mengatur bulan ataupun fitur pada grafik yang ditampilkan, *widget* teks yang berisi judul, serta *ChartWidget*.



Gambar 4.26 *ChartCardWidget*

ChartCardWidget akan melakukan *filtering* agar menampilkan data hasil analisis berdasarkan bulan dan menu analisis yang terpilih pada *dropdown* (perhatikan Gambar 4.27). Akan tetapi, dikarenakan menu analisis lainnya belum bisa melakukan penyimpanan hasil analisis maka *dropdown* pemilihan fitur menu analisis dinonaktifkan. *Filtering* dilakukan dengan fitur dari *dart* yaitu *where* yang akan mengembalikan nilai pada suatu *list* yang sesuai dengan kondisi tertentu. Dalam hal ini pada *list analysisHistory* hanya akan dikembalikan

analysisHistory yang bulan dari data *analysisHistory* sama dengan bulan yang dipilih pada *dropdown*. Pada *ChartWidget*, akan ditampilkan *widget LineChart* yang berasal dari *package fl_chart*.

```
Widget _buildDropdown(String dropdownType, String value, Icon icon,
List<DropdownMenuItem<String>> items) {
  return Container(
    width: MediaQuery.of(context).size.width / 2 - 50,
    padding: const EdgeInsets.symmetric(horizontal: 10),
    child: DropdownButtonHideUnderline(
      child: DropdownButton(
        alignment: Alignment.center,
        onChanged: (newValue) {
          setState(() {
            if (dropdownType == 'month') {
              selectedMonth = newValue!;
              filteredAnalysisHistory =
                widget.analysisHistory.where((obj) =>
                  DateFormat.MMMM('id_ID').format(DateTime.parse(obj.date))
                    == newValue).toList();
            }
            // if (dropdownType == 'type') {
            //   selectedType = newValue!;
            // }
          });
        },
        icon: icon,
        value: value,
        items: items,
      ),
    ),
  );
}
```

```
Container(
  width: MediaQuery.of(context).size.width,
  margin: const EdgeInsets.all(12.0),
```

```

padding: const EdgeInsets.only(left: 10,
                              top: 20, bottom: 18, right: 18),
decoration: BoxDecoration(
  color: Colors.white,
  borderRadius: const BorderRadius.all(Radius.circular(10)),
  boxShadow: [
    BoxShadow(
      color: Colors.grey.withOpacity(0.2),
      spreadRadius: 2,
      blurRadius: 2,
      offset: const Offset(0, 2),
    ),
  ],
),
child: Column(
  children: [
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        _buildDropdown('month', selectedMonth,
          const Icon(Icons.calendar_month_outlined), _monthItems),
        _buildDropdown('type', selectedType,
          const Icon(Icons.face_outlined), _typeItems),
      ],
    ),
    const Divider(
      color: Colors.grey,
    ),
    Padding(
      padding: const EdgeInsets.only(top: 20.0, bottom: 24.0),
      child: Text(
        'Grafik $selectedType Pada Bulan $selectedMonth',
        style: const TextStyle(fontSize: 16,
                              fontWeight: FontWeight.bold),
      ),
    ),
    AspectRatio(
      aspectRatio: 1.75,
      child: ChartWidget(analysisHistory: filteredAnalysisHistory),
    ),
  ],
)

```

```

    ],
  ),
);

```

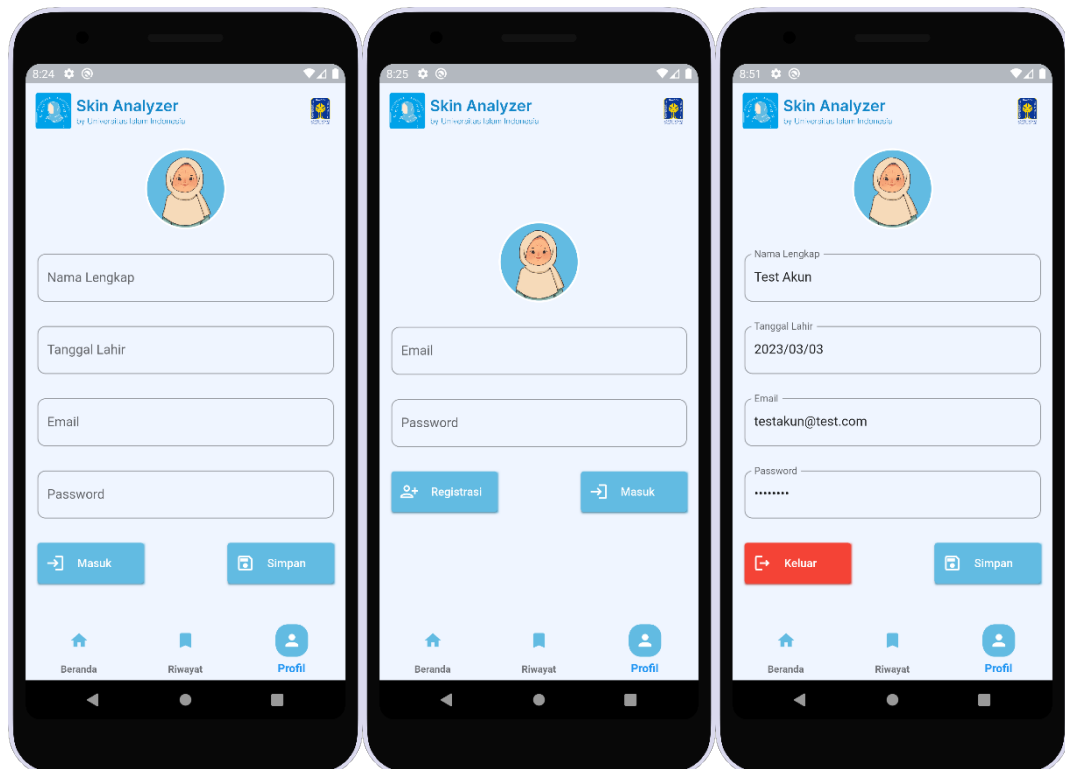
Gambar 4.27 Kode *ChartCardWidget*

4.1.19 Halaman Profil

Pada subbab halaman riwayat sebelumnya sudah dijelaskan bahwa pengguna hanya dapat melihat data hasil analisis sesuai dengan akun yang *login* pada aplikasi saat itu. Data hasil analisis sejatinya disimpan pada *database* lokal. Akan tetapi data akun pada profil disimpan pada *firebase*. *Firebase* adalah *backend-as-a-service* yang proses integrasinya dengan Flutter sangat mudah. Mulai dari proses *register* hingga manajemen *user* yang sedang *login* dapat dilakukan dengan mudah menggunakan *firebase*.

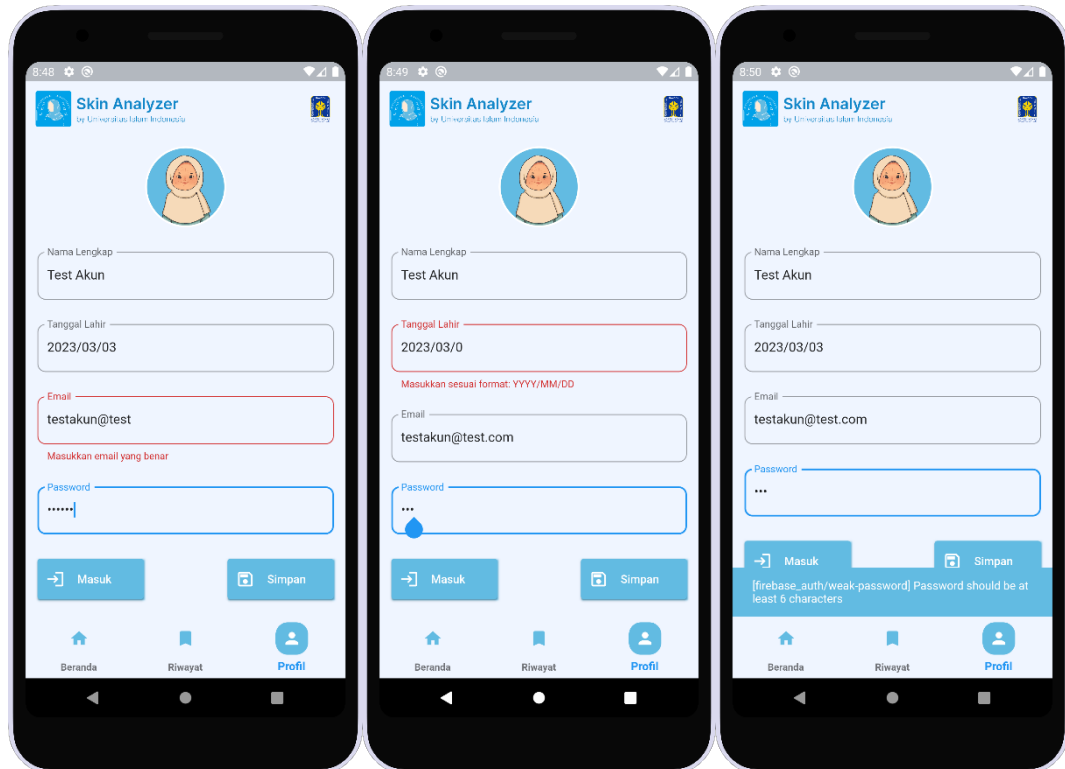
Fitur profil dibuat untuk mengakomodir kebutuhan terhadap privasi pengguna. Sehingga data hasil analisis yang disimpan akan memiliki atribut email yang berasal dari email yang terdaftar pada *firebase*, jadi pengguna dapat dengan mudah untuk tidak menampilkan hasil analisis yang tersimpan, hanya dengan menekan tombol keluar pada aplikasi, maka hasil analisis yang tersimpan tidak akan ditampilkan. Hal tersebut dikarenakan pada saat *fetch* data pada halaman Riwayat dilakukan *filtering* terlebih dahulu, apakah email pada data sesuai dengan email pengguna yang *login* saat itu. Selain itu, saat proses penyimpanan hasil analisis pada menu analisis, pengguna diharuskan untuk *login* terlebih dahulu untuk dapat menyimpan hasil analisis dikarenakan email dari pengguna akan dijadikan sebagai tanda pengenalan data hasil analisis tersebut.

Pada halaman profil itu sendiri hanya terdiri dari beberapa *form input* di antaranya nama, tanggal lahir, email, dan password (pada Gambar 4.28). Data-data tersebut akan tersimpan pada *firebase*. Selain itu, pada *firebase* nantinya akan tersimpan data *analysisCount*, di mana data ini akan merepresentasikan seberapa banyak pengguna menyimpan hasil analisisnya sehingga pada saat pengguna menekan tombol simpan pada halaman hasil analisis, data *analysisCount* pada *firebase* akan bertambah satu. Untuk itu, selain sebagai fitur privasi, fitur profil yang menyimpan data-data tersebut pada *firebase* ini harapannya dapat digunakan sebagai tolak ukur akan seberapa banyak pengguna yang menggunakan aplikasi *UII Skin Analyzer* ini secara berkelanjutan.



Gambar 4.28 Halaman Profil

Pada halaman ini juga sudah terdapat validasi untuk form pengisian (lihat Gambar 4.29). Fitur validasi ini berasal dari *widget input* yang ada pada Flutter. Pada gambar di bawah ditampilkan beberapa validasi yang ada pada form pengisian. Selain yang ditampilkan di gambar, validasi sederhana seperti tidak boleh adanya *input* yang kosong juga sudah diimplementasikan. Selain itu untuk validasi tanggal lahir dan email menggunakan fitur *regex* pada Flutter, di mana *pattern* yang digunakan berasal dari internet. Untuk menambah interaktifitas, ditambahkan *snackbar* untuk menampilkan pesan singkat, seperti pada gambar di bawah menampilkan pesan yang didapatkan dari *firebase* bahwa *password* yang dimasukkan harus lebih dari 6 karakter.



Gambar 4.29 Halaman Profil

Penggunaan *firebase* pada Flutter dibantu dengan beberapa *package* di antaranya *cloud_firestore*, *firebase_auth*, dan *firebase_core*. *Package* tersebut dikembangkan langsung oleh pengembang *firebase*. Sebelum menggunakan *firebase* pada aplikasi Flutter juga terdapat beberapa tahap instalasi terlebih dahulu, di mana tahap-tahap tersebut dijelaskan pada halaman *firebase*, pengguna dapat langsung mengikuti tahapan-tahapan yang ada.

Berikut contoh kode untuk melakukan *sign in* pada aplikasi UII *Skin Analyzer*. Hal yang pertama dilakukan adalah mendeklarasikan *instance* dari fitur *firebase* yang ingin digunakan. Pada proses *sign in* digunakan fitur *auth* sehingga perlu dilakukan deklarasi sebelum dapat melakukan proses *sign in*. *Instance FirebaseAuth* ini juga dapat digunakan untuk melakukan pengecekan apakah pengguna sedang login pada aplikasi atau tidak. Hanya dengan mendeklarasikan *instance*, maka dapat langsung mengecek dengan memanggil atribut *currentUser* yang akan mengembalikan data pengguna yang sedang *login* atau mengembalikan *null* jika pengguna belum *login*. Metode *auth* ini digunakan di seluruh aplikasi UII *Skin Analyzer* yang membutuhkan data pengguna yang sedang *login*. Pada fungsi *_onPressedSignIn* di bawah (perhatikan Gambar 4.30) dijalankan fungsi *signInWithEmailAndPassword* dan

memasukkan parameter email serta *password* maka pengguna akan langsung *login* dan keseluruhan aplikasi akan otomatis diberitahu akan pengguna yang sedang *login*.

```

FirebaseAuth auth = FirebaseAuth.instance;

void _onPressedSignIn() async {
  try {
    setState(() {
      isLoading = true;
    });

    await auth.signInWithEmailAndPassword(
      email: emailController.text,
      password: passwordController.text,
    );

    setState(() {
      isLoading = false;
    });

    _changeSignedInState();

    if (!mounted) return;

    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Selamat Datang Kembali'),
        backgroundColor: Color(0xFF62BBE2),
        duration: Duration(seconds: 2),
      ),
    );
  } on FirebaseAuthException catch (e) {
    setState(() {
      isLoading = false;
    });

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text('${e.message}'),

```

```

        backgroundColor: const Color(0xFF62BBE2),
        duration: const Duration(seconds: 2),
    ),
);
}
}

```

Gambar 4.30 Fungsi `_onPressedSignIn`

4.2 Pembahasan

Pada subbab ini akan dijelaskan mengenai hasil dari pengujian *BlackBox* dari aplikasi *UII Skin Analyzer*. Selain itu juga akan diberikan pemaparan akan kelebihan dan kekurangan dari aplikasi *UII Skin Analyzer*.

4.2.1 Pengujian *BlackBox*

Pengujian *blackbox* adalah pengujian yang tanpa mengetahui dan menjalankan *source code* dari aplikasinya. Pengujian *blackbox* juga dapat disebut sebagai pengujian untuk menguji *behaviour* atau perilaku dari sistemnya dan apa saja yang dapat dilakukan oleh sistem tersebut. Jika diberikan suatu kondisi tertentu sistem yang diujikan akan menghasilkan perilaku atau tampilan yang seperti apa. Pengujian *blackbox* dilakukan untuk menguji apakah aplikasi sudah berjalan sebagaimana yang diharapkan atau belum. Pada kesempatan ini pengujian *blackbox* dilakukan oleh Muhammad Satrio Wicaksono pelajar SMA melalui *Google Meet* (dokumentasi dapat dilihat pada lampiran Dokumentasi Pengujian Blackbox).

Saat Membuka Aplikasi

Berikut ditampilkan Tabel 4.1 yang berisi pengujian *BlackBox* aplikasi *UII Skin Analyzer* saat membuka aplikasi.

Tabel 4.1 Pengujian *BlackBox* pada saat membuka aplikasi

No	Test Case	Expected Result	Test Result
1	Membuka aplikasi <i>UII Skin Analyzer</i>	Ditampilkan animasi <i>fade in</i> selama 3 detik	Berhasil
2	Pengguna belum pernah memasukkan gambar pada aplikasi	Ditampilkan halaman <i>onboarding</i> setelah animasi <i>splash screen</i>	Berhasil

3	Pengguna sudah pernah memasukkan gambar pada aplikasi	Ditampilkan halaman beranda	Berhasil
4	Pengguna menggeser layar ke kiri atau ke kanan pada halaman <i>onboarding</i>	Konten pada halaman <i>onboarding</i> berubah	Berhasil
5	Pengguna menggeser layar hingga konten terakhir pada halaman <i>onboarding</i>	Ditampilkan tombol ‘Selanjutnya’	Berhasil
6	Pengguna menekan tombol ‘Selanjutnya’ pada konten terakhir pada halaman <i>onboarding</i>	Ditampilkan <i>pop-up</i> dengan animasi <i>scale</i> yang berisi <i>informed consent</i>	Berhasil
7	Pengguna menekan ‘Tidak Setuju’ pada <i>pop-up informed consent</i>	Tetap ditampilkan halaman <i>onboarding</i>	Berhasil
8	Pengguna menekan ‘Setuju’ pada <i>pop-up informed consent</i>	Ditampilkan halaman beranda	Berhasil

Halaman Beranda

Berikut ditampilkan Tabel 4.2 yang berisi pengujian *BlackBox* aplikasi UII *Skin Analyzer* pada halaman beranda.

Tabel 4.2 Pengujian *BlackBox* pada halaman beranda

No	<i>Test Case</i>	<i>Expected Result</i>	<i>Test Result</i>
1	Pengguna menekan tombol navigasi ‘Beranda’ pada bagian bawah halaman	Ditampilkan halaman beranda	Berhasil
2	Pengguna menekan tombol ‘Jerawat’ pada halaman beranda	Ditampilkan halaman menu analisis untuk jerawat	Berhasil
3	Pengguna menekan tombol ‘Keriput’ pada halaman beranda	Ditampilkan halaman menu analisis untuk keriput	Berhasil
4	Pengguna menekan tombol ‘Kemerahan’ pada halaman beranda	Ditampilkan halaman menu analisis untuk kemerahan	Berhasil
5	Pengguna menekan tombol ‘Bercak Hitam’ pada halaman beranda	Ditampilkan halaman menu analisis untuk bercak hitam	Berhasil
6	Pengguna menekan tombol ‘Jenis Kulit’ pada halaman beranda	Ditampilkan <i>pop-up</i> yang berisi fitur analisis jenis kulit masih dalam pengembangan	Berhasil

Halaman Menu Analisis Jerawat

Berikut ditampilkan Tabel 4.3 yang berisi pengujian *BlackBox* aplikasi *UII Skin Analyzer* pada halaman menu analisis jerawat.

Tabel 4.3 Pengujian *BlackBox* pada halaman menu analisis jerawat

No	<i>Test Case</i>	<i>Expected Result</i>	<i>Test Result</i>
1	Pengguna belum pernah memasukkan gambar pada aplikasi	Ditampilkan tutorial bagaimana melakukan analisis	Berhasil
2	Pengguna menekan bagian layar mana saja saat tutorial berlangsung	Tutorial selesai dan menampilkan halaman menu analisis jerawat	Berhasil
3	Pengguna belum pernah memasukkan gambar pada aplikasi dan telah menyelesaikan tutorial	Ditampilkan pesan untuk memasukkan gambar sebelum memulai analisis pada halaman menu analisis	Berhasil
4	Pengguna menekan tombol ‘Bantuan’	Ditampilkan <i>pop-up</i> bantuan dengan animasi <i>scale</i>	Berhasil
5	Pengguna menekan tombol ‘Analisis’ saat belum pernah memasukkan gambar	Tombol tidak dapat ditekan	Berhasil
6	Pengguna menekan tombol ‘Galeri’	Diarahkan ke <i>file manager</i> yang hanya dapat memilih <i>file</i> gambar saja	Berhasil
7	Pengguna memilih gambar	Pesan untuk memilih gambar akan digantikan oleh gambar yang terpilih	Berhasil
8	Pengguna menekan tombol ‘Galeri’ saat sudah ada gambar yang terpilih dan memilih gambar	Gambar yang terpilih sebelumnya akan digantikan oleh gambar yang terpilih saat ini	Berhasil
9	Pengguna menekan tombol ‘Galeri’ dan menekan tombol <i>back</i> sehingga tidak memilih gambar	Gambar pada halaman menu analisis tidak berubah	Berhasil
10	Pengguna menekan tombol ‘Kamera’	Ditampilkan <i>pop-up</i> panduan pengambilan gambar	Berhasil

11	Pengguna menekan tombol 'Mengerti' pada <i>pop-up</i> panduan pengambilan gambar	Ditampilkan halaman kamera yang sudah terdapat <i>overlay</i> bantuan posisi wajah	Berhasil
12	Pengguna menekan tombol <i>flashlight</i>	<i>Flashlight</i> akan menyala atau mati sesuai dengan tombol yang dipilih	Berhasil
13	Pengguna menekan tombol <i>rotate</i>	Kamera akan berubah menjadi kamera belakang jika sebelum ditekan <i>rotate</i> adalah kamera depan dan begitu pula sebaliknya	Berhasil
14	Pengguna menekan tombol ambil gambar	Kamera akan mengambil gambar dan mengembalikan ke halaman menu analisis saat gambar selesai diambil. Gambar yang terpilih akan berubah sesuai dengan gambar yang diambil pada kamera	Berhasil
15	Pengguna menekan tombol analisis	Melakukan analisis dan menonaktifkan tombol-tombol pada halaman. Saat analisis selesai akan menampilkan hasil analisis yang berasal dari <i>server</i>	Berhasil
16	Pengguna menekan tombol 'Simpan' saat tidak ada <i>error</i> pada hasil analisis	Ditampilkan <i>pop-up</i> pengguna harus masuk pada aplikasi terlebih dahulu	Berhasil
17	Pengguna menekan tombol 'Simpan' saat tidak ada <i>error</i> pada hasil analisis dan pengguna sudah masuk pada aplikasi	Hasil analisis disimpan dan ditampilkan <i>pop-up</i> apakah penyimpanan berhasil atau gagal	Berhasil

Halaman Menu Analisis Keriput, Kemerahan, dan Bercak Hitam

Berikut ditampilkan Tabel 4.4 yang berisi pengujian *BlackBox* aplikasi *UII Skin Analyzer* pada halaman menu analisis keriput, kemerahan, dan bercak hitam.

Tabel 4.4 Pengujian *BlackBox* pada halaman menu analisis selain jerawat

No	<i>Test Case</i>	<i>Expected Result</i>	<i>Test Result</i>
1	Pengguna menekan tombol analisis	Ditampilkan hasil analisis	Berhasil
2	Pengguna menekan tombol galeri atau kamera	Ditampilkan <i>pop-up</i> pesan tidak bisa mengganti gambar	Berhasil

Halaman Riwayat

Berikut ditampilkan Tabel 4.5 yang berisi pengujian *BlackBox* aplikasi *UII Skin Analyzer* pada halaman riwayat.

Tabel 4.5 Pengujian *BlackBox* pada halaman riwayat

No	<i>Test Case</i>	<i>Expected Result</i>	<i>Test Result</i>
1	Pengguna menekan tombol ‘Riwayat’ pada <i>navigation bar</i> saat belum masuk ke aplikasi	Ditampilkan pesan yang berisi pengguna harus masuk terlebih dahulu	Berhasil
2	Pengguna menekan tombol ‘Riwayat’ pada <i>navigation bar</i> saat sudah masuk ke aplikasi	Ditampilkan kalender dan grafik hasil analisis yang tersimpan sesuai dengan akun yang <i>login</i> saat itu	Berhasil
3	Pengguna menekan hari yang tidak memiliki titik pada kalender	Ditampilkan perubahan <i>highlight</i> hari yang terpilih dan jika sebelumnya terdapat 5 tombol pada kalender maka tombol akan menghilang	Berhasil
4	Pengguna menekan hari yang memiliki tanda titik	Ditampilkan perubahan <i>highlight</i> hari yang terpilih dan 5 tombol di bawah kalender yang jika ditekan akan menuju halaman detail riwayat pada hari tersebut	Berhasil

5	Pengguna menekan titik pada grafik	Ditampilkan <i>tooltip</i> informasi akan berapa jerawat yang terdeteksi dan tanggal kapan analisis tersebut dilakukan	Berhasil
6	Pengguna menekan <i>dropdown</i> bulan pada grafik dan memilih bulan	Grafik akan menampilkan data hasil analisis yang tersimpan pada bulan yang dipilih	Berhasil
7	Pengguna menekan tombol ‘Jerawat’ yang ada di bawah kalender	Ditampilkan halaman detail hasil analisis	Berhasil
8	Pengguna menekan tombol ‘Keriput’, ‘Kemerahan’, ‘Bercak Hitam’, dan ‘Jenis Kulit’ yang ada di bawah kalender	Ditampilkan <i>pop-up</i> pesan bahwa fitur riwayat belum tersedia pada menu analisis tersebut	Berhasil
9	Pengguna memilih hari pada kalender horizontal di menu detail hasil analisis	Ditampilkan detail hasil analisis pada hari yang dipilih, jika tidak ada hasil analisis yang tersimpan, akan menampilkan pesan tidak ada hasil analisis yang tersimpan	Berhasil

Halaman Profil

Berikut ditampilkan Tabel 4.6 yang berisi pengujian *BlackBox* aplikasi *UII Skin Analyzer* pada halaman profil.

Tabel 4.6 Pengujian *BlackBox* pada halaman profil

No	<i>Test Case</i>	<i>Expected Result</i>	<i>Test Result</i>
1	Pengguna menekan tombol ‘Profil’ pada <i>navigation bar</i> saat belum masuk ke aplikasi	Ditampilkan halaman profil	Berhasil
2	Pengguna menekan tombol ‘Profil’ pada <i>navigation bar</i> saat sudah masuk ke aplikasi	Ditampilkan halaman profil dengan <i>input</i> data yang sudah terisi	Berhasil
3	Pengguna menekan tombol masuk	Ditampilkan <i>input</i> email dan <i>password</i> serta tombol masuk berubah menjadi tombol registrasi	Berhasil

		dan tombol masuk berpindah ke sebelah kanan	
4	Pengguna menekan tombol simpan saat belum memasukkan data pada <i>input</i> yang tersedia	Ditampilkan <i>error</i> yang mengharuskan pengguna untuk memasukkan data terlebih dahulu	Berhasil
5	Pengguna menekan tombol simpan dengan data tanggal lahir yang tidak sesuai format, contoh: 2001/13/15	Ditampilkan <i>error</i> yang berisi format tanggal lahir yang benar, yaitu: YYYY/MM/DD	Berhasil
6	Pengguna menekan tombol simpan dengan data email yang tidak benar, contoh: email@email	Ditampilkan <i>error</i> yang berisi pesan masukkan email yang benar	Berhasil
7	Pengguna menekan tombol simpan saat semua data sudah benar	Ditampilkan <i>loader</i> dan <i>snackbar</i> yang berisi pesan dari <i>firebase</i> jika terdapat <i>error</i> atau pesan berhasil jika tidak terdapat <i>error</i>	Berhasil
8	Pengguna menekan tombol simpan setelah mengubah data nama dan tanggal lahir	Ditampilkan <i>loader</i> dan <i>snackbar</i> yang berisi pesan dari <i>firebase</i> jika terdapat <i>error</i> atau pesan berhasil jika tidak terdapat <i>error</i>	Berhasil
9	Pengguna menekan tombol masuk dan mengisi data email dan <i>password</i>	Ditampilkan <i>loader</i> dan <i>snackbar</i> yang berisi pesan dari <i>firebase</i> jika terdapat <i>error</i> atau pengguna akan dilempar ke halaman beranda jika <i>login</i> berhasil dan pesan <i>snackbar</i> 'Selamat Datang Kembali'	Berhasil
10	Pengguna menekan tombol keluar	Pengguna dipindahkan ke halaman beranda	Berhasil

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Aplikasi *mobile* UII *Skin Analyzer* adalah aplikasi yang menghubungkan sistem analisis wajah dari penelitian sebelumnya dengan aplikasi *mobile* agar memudahkan pengguna dalam penggunaannya dan membantu pengguna untuk menganalisis permasalahan kulit wajahnya. Selain sebagai penghubung, aplikasi UII *Skin Analyzer* sangat memperhatikan privasi pengguna sehingga pada fitur riwayat yang memungkinkan pengguna untuk menyimpan dan melihat hasil analisisnya secara pribadi karena menggunakan *database* yang ada pada perangkat pengguna. Aplikasi ini dikembangkan dengan tampilan antarmuka yang menarik agar memberikan pengalaman yang menyenangkan saat menggunakan aplikasi ini. Pada fitur riwayat terdapat grafik garis sehingga pengguna dapat dengan mudah melihat perkembangan terhadap kondisi wajahnya. Aplikasi UII *Skin Analyzer* dikembangkan dengan metode RAD (*Rapid Application Development*) dan menggunakan *framework* Flutter untuk implementasinya. Kemudian dari hasil penelitian aplikasi UII *Skin Analyzer* dapat diambil kesimpulan sebagai berikut:

- a. Aplikasi UII *Skin Analyzer* dibuat dan dikembangkan dengan metode *Rapid Application Development* (RAD) dan menggunakan bantuan *framework* Flutter.
- b. Aplikasi UII *Skin Analyzer* sangat memperhatikan user interface dan user experience yang baik. Hal ini ditandai dengan diadakannya pengujian usability testing dengan SUS yang bertemu langsung dengan pengguna akhir serta mendapatkan feedback positif dari para responden. Selanjutnya, feedback tersebut dijadikan sebagai acuan dalam perbaikan di iterasi kedua.
- c. Aplikasi UII *Skin Analyzer* telah dievaluasi terkait fungsionalitas dan penerimaan pengguna dengan metode usability testing menggunakan SUS dan dibuktikan dengan hasil pengujian dengan skor akhir 82.6% yang berada pada nilai A dan menurut Skala Likert berada pada kategori Sangat Baik.
- d. Aplikasi UII *Skin Analyzer* menjaga dan memperhatikan privasi pengguna dengan melakukan penyimpanan seluruh data hanya pada perangkat pengguna saja. Sehingga dapat dipastikan bahwa yang dapat mengakses data yang telah dimasukkan hanya pengguna tersebut. Dengan demikian, aplikasi ini dapat menjadi handheld application dan

dapat digunakan dalam jangka panjang tanpa perlu khawatir mengenai kebocoran data pengguna.

5.2 Saran

Berdasarkan kekurangan dan keterbatasan yang ditemukan pada tahap implementasi maka untuk pengembangan lebih lanjut terhadap aplikasi UII *Skin Analyzer* ini ke depannya dapat disarankan sebagai berikut:

- a. Menerapkan metode *deep learning* terhadap fitur analisis keriput, kemerahan, bercak hitam, dan jenis kulit agar mampu menganalisis berbagai macam citra.
- b. Mengembangkan fitur riwayat untuk hasil analisis dari fitur keriput, kemerahan, bercak hitam, dan jenis kulit.
- c. Dikembangkannya fitur untuk menghapus hasil analisis yang tersimpan.
- d. Menambahkan fitur untuk menambahkan atau mengganti foto profil.
- e. Melakukan pengujian menggunakan berbagai jenis perangkat atau sistem operasi yang berbeda-beda.

DAFTAR PUSTAKA

- Anindita, A. H., & AF, S. M. (2017). Formulasi Masker Alami Berbahan Dasar Rumput Laut Dan Cokelat Mengurangi Keriput Dan Bintik Noda Pada Kulit Wajah. *Care : Jurnal Ilmiah Ilmu Kesehatan*, 5(2), 205–219.
<https://jurnal.unitri.ac.id/index.php/care/article/view/540>
- Ashraf, H. (2022). *Flutter Pros & Cons*. <https://flutterdesk.com/flutter-pros-cons/>
- Bangor, A., Kortum, P., & Miller, J. (2009). *Determining What Individual SUS Scores Mean: Adding an Adjective Rating ScaleJUX*. <https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/>
- Bikker, G. (2022). *Daily Time on Mobile Passes 5 Hours in Q1 2022 Among Top 10 Mobile-First Markets*. <https://www.data.ai/en/insights/market-data/q1-2022-market-pulse-regional/>
- Burkov, A. (2019). *The Hundred-Page Machine Learning*.
- Chu, D. H. (2012). Chapter 7. Development and Structure of Skin. Dalam L. A. Goldsmith, S. I. Katz, B. A. Gilchrest, A. S. Paller, D. J. Leffell, & K. Wolff (Ed.), *Fitzpatrick's Dermatology in General Medicine*, 8e. The McGraw-Hill Companies.
accessmedicine.mhmedical.com/content.aspx?aid=56021404
- Dermalogica. (2015). *face mapping* — *Dermalogica*.
<https://www.dermalogica.com/pages/face-mapping>
- Djuanda, A. (2007). *Ilmu Penyakit Kulit Dan Kelamin*.
[//opac.stikesmucis.ac.id%2Findex.php%3Fp%3Dshow_detail%26id%3D567](https://opac.stikesmucis.ac.id%2Findex.php%3Fp%3Dshow_detail%26id%3D567)
- Fayzullaev, J. (2018). *Native-like Cross-Platform Mobile Development*.
- Fitzpatrick, T. B., & Wolff, K. (2008). *Fitzpatrick's dermatology in general medicine. Published in 2008 in New York by McGraw-Hill*.
<https://lib.ugent.be/catalog/rug01:001251477>
- Freeborn, C. (2022). *Pros and cons of Flutter app development*.
<https://blog.logrocket.com/pros-cons-flutter-app-development/>
- Google. (2011). *Dart programming language | Dart*. <https://dart.dev/>
- Google. (2013). *Download Android Studio & App Tools - Android Developers*.
<https://developer.android.com/studio>
- Google. (2018a). *Flutter - Build apps for any screen*. <https://flutter.dev/>

- Google. (2018b). *Introduction to widgets* | *Flutter*.
<https://docs.flutter.dev/development/ui/widgets-intro>
- Google. (2018c). *Using packages* | *Flutter*. <https://docs.flutter.dev/development/packages-and-plugins/using-packages>
- Helfrich, Y., Sachs, D., & Voorhees, J. (2008). Overview of skin aging and photoaging. *Dermatology nursing / Dermatology Nurses' Association*, 20, 177–183; quiz 184.
- Jacobson, Daniel., Brail, Greg., & Woods, Dan. (2011). *APIs : a strategy guide*. 134.
- Kothalawala, A. (2018). *What is an API? How does it work?*
<https://medium.com/@ama.thanu/what-is-an-api-how-does-it-work-f4ea552d741f>
- Lee, E., Cho, C., & Ha, J. (2021). Biophysical properties of redness-prone skin in Korean women. *Journal of Cosmetic Dermatology*, 21. <https://doi.org/10.1111/jocd.14722>
- Lee Ventola, C. (2014). Mobile Devices and Apps for Health Care Professionals: Uses and Benefits. *Pharmacy and Therapeutics*, 39(5), 356. /pmc/articles/PMC4029126/
- L'Oréal Paris. (2020). *Analisa kulitmu selayaknya seorang ahli - L'Oréal Paris*.
<https://www.loreal-paris.co.id/skin-genius>
- MarkPlus Inc. (2021). *ZAP Clinic | Klinik Kesehatan khusus wanita terbaik di indonesia*. MarkPlus, Inc. <https://zapclinic.com/zapbeautyindex>
- Martini, Frederic., & Nath, J. Lindsley. (2009). *Fundamentals of anatomy & physiology*. 1123.
- Miftah, B., Olusewa, A., Damilare, A., Sarwar, M., & Rahim Soomro, T. (2013). Impact of Smartphone's on Society. *European Journal of Scientific Research*, 98(2), 216–226.
<http://www.europeanjournalofscientificresearch.com>
- Muhimmah, I., Prasetyo, D., Kurniawardhani, A., & Rani, S. (2023). *Prototype for Automatically Detecting Acne in Face Images Using Digital Image Processing Techniques*.
- Mumpuni, Y. (2011). *Cara jitu mengatasi jerawat*. Andi Offset.
- Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). *Comparison of JSON and XML Data Interchange Formats: A Case Study*.
- Nwabude, O. (2022). *How to Implement Any UI in Flutter*.
<https://www.freecodecamp.org/news/how-to-implement-any-ui-in-flutter/>
- Olay. (2017). *Olay Skin Advisor*. <https://www.olay.com/skin-advisor>
- Pangaribuan, L. (2017). PUSDIBANG-KS UNIMED 20 Efek Samping Kosmetik dan Penanganannya Bagi Kaum Perempuan. *Jurnal Keluarga Sehat Sejahtera*, 15(2).

- Pusparisa, Y. (2021). *Daftar Negara Pengguna Smartphone Terbanyak, Indonesia Urutan Berapa?* | *Databoks*. <https://databoks.katadata.co.id/datapublish/2021/07/01/daftar-negara-pengguna-smartphone-terbanyak-indonesia-urutan-berapa>
- Rahman, V. R., Bratadiredja, M. A., & Saptarini, M.Si, Apt., N. M. (2021). Artikel Review: Potensi Kolagen sebagai Bahan Aktif Sediaan Farmasi. *Majalah Farmasetika*, 6(3), 253–286. <https://doi.org/10.24198/MFARMASETIKA.V6I3.33621>
- Ratnadita, A. (2016). *IAI - Artikel Hampir 50% Kasus Penyakit Kulit Disebabkan Produk Kosmetik*. <https://iai.id/news/artikel/hampir-50-kasus-penyakit-kulit-disebabkan-produk-kosmetik>
- Salman, A., Kurt, E., Topcuoglu, V., & Demircay, Z. (2016). Social Anxiety and Quality of Life in Vitiligo and Acne Patients with Facial Involvement: A Cross-Sectional Controlled Study. *American journal of clinical dermatology*, 17(3), 305–311. <https://doi.org/10.1007/S40257-016-0172-X>
- Sampelan, M. G., Pangemanan, D., Kundre, R. M., Studi, P., Keperawatan, I., & Kedokteran, F. (2017). *Hubungan Timbulnya Acne Vulgaris dengan Tingkat Kecemasan pada Remaja di SMPN 1 Likupang Timur*. 5(1).
- Satria. (2021). *Mengenal Jenis Kulit Kita* | *Universitas Gadjah Mada*. <https://ugm.ac.id/id/berita/22064-mengenal-jenis-kulit-kita>
- Sjerobabski Masnec, I., & Situm, M. (2010). Skin aging. *Acta clinica Croatica*, 49, 515–518.
- SQLite Consortium. (2000). *SQLite Home Page*. <https://www.sqlite.org/index.html>
- Susanto, R. C. (2013). *Penyakit Kulit dan Kelamin*. Nuha Medika.
- Tekartik. (2018). *sqflite | Flutter Package*. <https://pub.dev/packages/sqflite>
- Word, A., Pathi, S., & Schaffer, J. (2009). Pathophysiology of Pelvic Organ Prolapse. *Obstetrics and gynecology clinics of North America*, 36, 521–539. <https://doi.org/10.1016/j.ogc.2009.09.001>

LAMPIRAN

A. Hasil Kuesioner pengujian *System Usability Scale* (SUS) pada iterasi ke-1

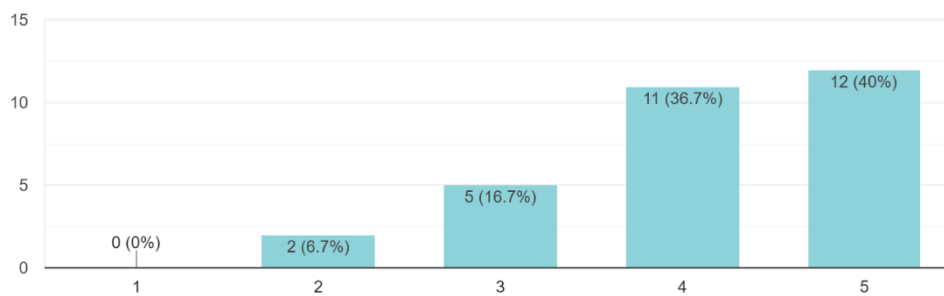
1. Pilihan jawaban dan skor

No	Pilihan Jawaban	Skor
1	Sangat Tidak Setuju (STS)	1
2	Tidak Setuju (TS)	2
3	Ragu-Ragu (RG)	3
4	Setuju (S)	4
5	Sangat Setuju (SS)	5

2. Pernyataan positif 1

Saya berpikir akan menggunakan sistem ini lagi.

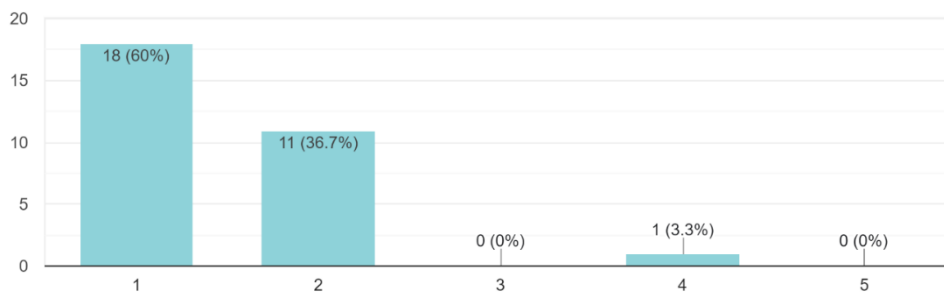
30 responses



3. Pernyataan negatif 1

Saya merasa sistem ini rumit untuk digunakan.

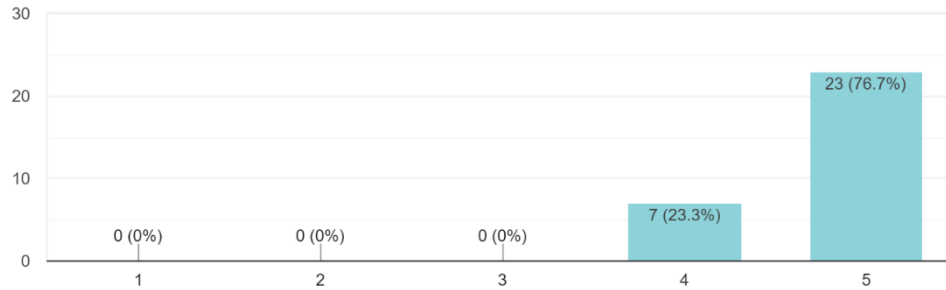
30 responses



4. Pernyataan positif 2

Saya merasa sistem ini mudah digunakan.

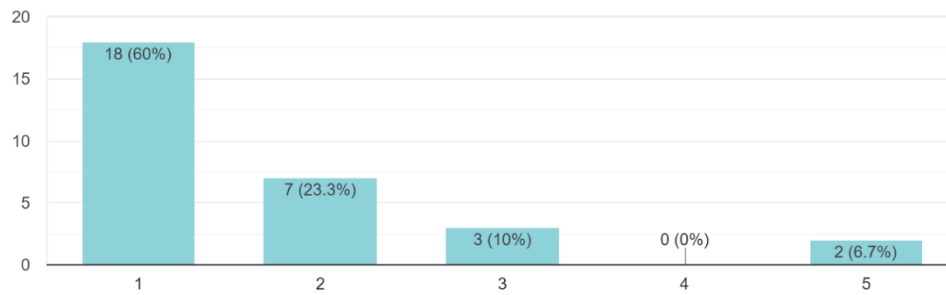
30 responses



5. Pernyataan negatif 2

Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.

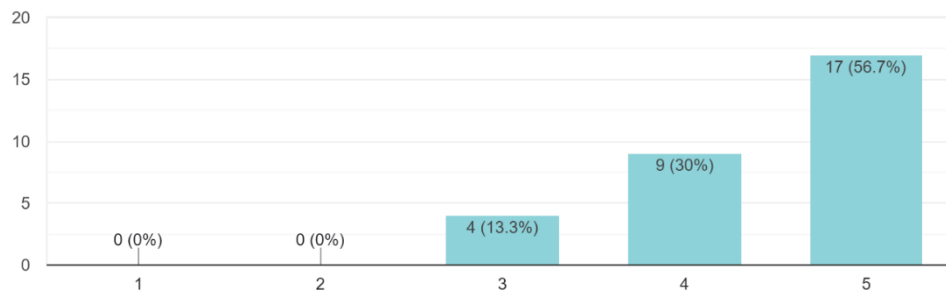
30 responses



6. Pernyataan positif 3

Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

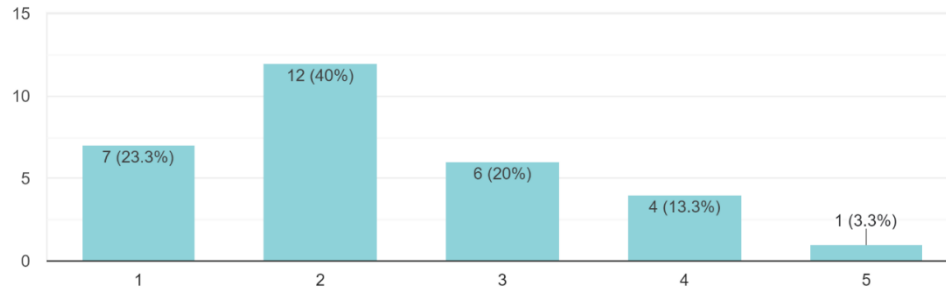
30 responses



7. Pernyataan negatif 3

Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).

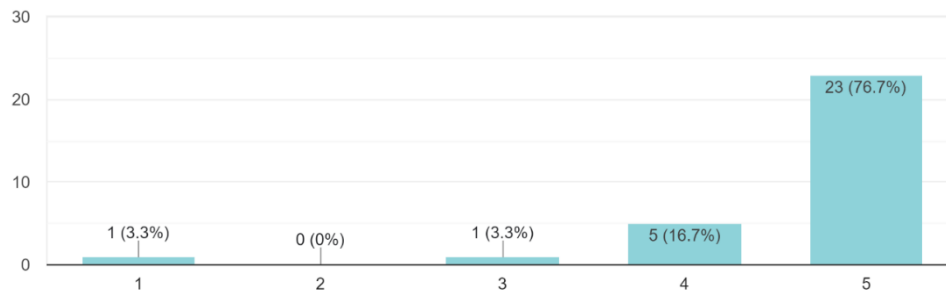
30 responses



8. Pernyataan positif 4

Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

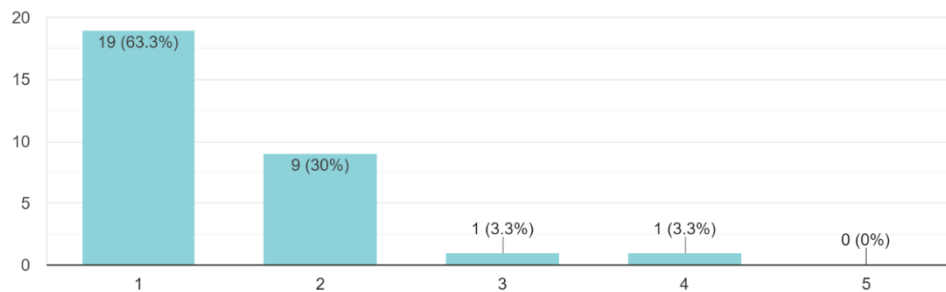
30 responses



9. Pernyataan negatif 4

Saya merasa sistem ini membingungkan.

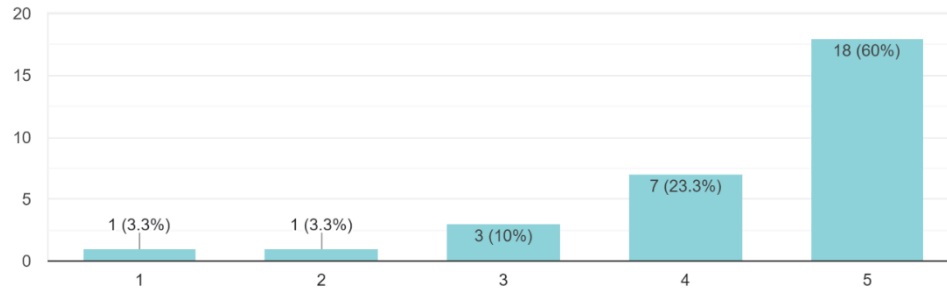
30 responses



10. Pernyataan positif 5

Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

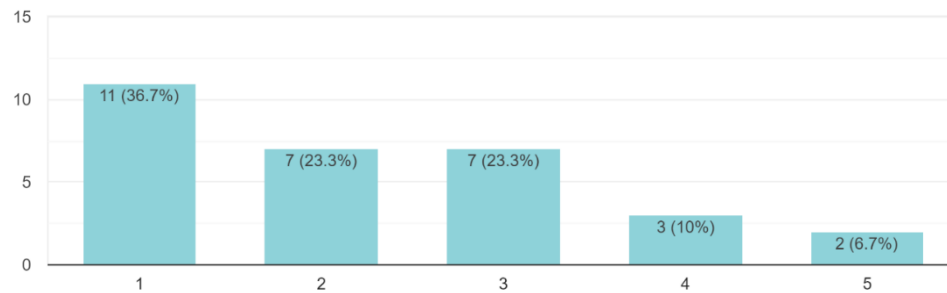
30 responses



11. Pernyataan negatif 5

Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

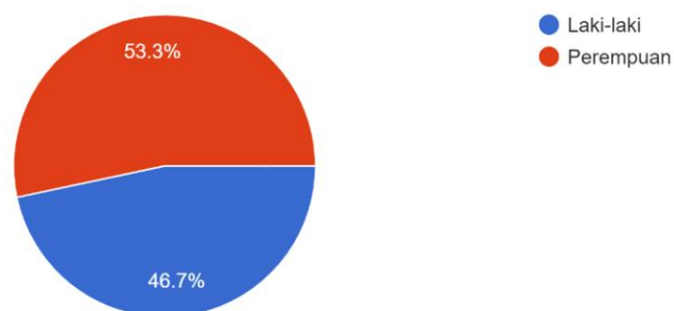
30 responses



12. Jenis Kelamin

Jenis Kelamin

30 responses



13. Lampiran Profil Responden

Nama	Usia	Jenis Kelamin
Raynord Rasendriya Risatama	20	Laki-laki
Satrio	15	Laki-laki
Anwaruddin Ridho Novianto	22	Laki-laki
Desy Kusuma	21	Perempuan
Sonya	22	Perempuan
Ghina Fathia	19	Perempuan
Sutta Wira Admojo	21	Laki-laki
Hafizh Qhorieq	20	Laki-laki
Muhammad Suriansyah	22	Laki-laki
Rafly	21	Laki-laki
Ariennova	20	Perempuan
Nanda	20	Perempuan
Zikriandhini	21	Perempuan
Istiqomah	20	Perempuan
Umam	21	Laki-laki
Ahmad Fathannafi	21	Laki-laki
Allaam Muhammad Abdillah	21	Laki-laki
Eka Aprilianti	20	Perempuan
Isnurul Kusnaeni	52	Perempuan
Rini	19	Perempuan
Vania	15	Perempuan
Yufita Dwi Marlina	21	Perempuan
Nurafni	20	Perempuan
Renita Sendya Fadhillah	20	Perempuan
Syifa	22	Perempuan
Muhammad Riyadi AM	21	Laki-laki
Dzakwan Anas Young Anhar	15	Laki-laki
Rasyid	15	Laki-laki
Nina Indria	21	Perempuan
Naufal Yusran	21	Laki-laki

14. Dokumentasi Pengujian Blackbox

The image shows a Google Meet window on the left and a Microsoft Word document on the right. The Meet window displays a presentation titled 'Informed Consent' by Nurastuti Wijani. The Word document, titled 'Blackbox testing...', contains a table with 8 rows of test cases. Each row includes a step number, a description of the action, the expected result, and the actual result.

2	Pengguna belum pernah memasukkan gambar pada aplikasi	Ditampilkan halaman onboarding setelah animasi splash screen	Berhasil
3	Pengguna sudah pernah memasukkan gambar pada aplikasi	Ditampilkan halaman beranda	Berhasil
4	Pengguna menggeser layar ke kiri atau ke kanan pada halaman onboarding	Konten pada halaman onboarding berubah	Berhasil
5	Pengguna menggeser layar hingga konten terakhir pada halaman onboarding	Ditampilkan tombol 'Selanjutnya'	Berhasil
6	Pengguna menekan tombol 'Selanjutnya' pada konten terakhir pada halaman onboarding	Ditampilkan pop up dengan animasi scale yang berisi informed consent	
7	Pengguna menekan 'Tidak Setuju' pada pop-up informed consent	Tetap ditampilkan halaman onboarding	
8	Pengguna menekan 'Setuju' pada pop-up informed consent	Ditampilkan halaman beranda	

15. Lampiran rekapitulasi perhitungan SUS

Hasil Pengujian System Usability Scale (SUS) - UII Skin Analyzer - ...																						Search		Nurastuti Wijarani		NW															
File Home Insert Page Layout Formulas Data Review View Help																						Share																			
O47																																									
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y																
	Nama	Usia	Jenis Kelamin	Saya berpikir akan menggunakan sistem ini lagi.	Result	Saya merasa sistem ini rumit untuk digunakan.	Result	Saya merasa sistem ini mudah digunakan.	Result	Saya membutuhkan bantuan dari orang lain atau	Result	Saya merasa fitur sistem ini berjalan	Result	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)	Result	Saya merasa orang lain akan memahami cara menggunakan	Result	Saya merasa sistem ini membingungkan.	Result	Saya merasa tidak ada hambatan dalam menggunakan	Result	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.	Result	TOTAL	TOTAL X 2,5																
1	Raynord Rasendriya Risatama	20	Laki-laki	2	1	1	4	5	4	1	4	5	4	2	3	1	0	1	4	1	0	1	4																		
2	Satrio	15	Laki-laki	2	1	1	4	5	4	2	3	5	4	2	3	5	4	1	4	5	4	1	4	28	70																
3	Anwaruddin Ridho Novianto	22	Laki-laki	5	4	1	4	5	4	1	4	5	4	3	2	5	4	2	3	5	4	4	1	35	87.5																
4	Desy Kusuma	21	Perempuan	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	34	85																
5	Sonya	22	Perempuan	4	3	2	3	5	4	1	4	4	3	4	1	4	3	2	3	4	3	2	4	40	100																
6	Ghina Fathia	19	Perempuan	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	30	75																
7	Sutta Wira Admojo	21	Laki-laki	5	4	1	4	5	4	1	4	5	4	1	4	4	3	1	4	5	4	2	4	40	100																
8	Hafizh Qhorieq	20	Laki-laki	5	4	1	4	5	4	5	0	5	4	5	0	5	4	1	4	5	4	3	3	38	95																
9	Muhammad Suriansyah	22	Laki-laki	4	3	1	4	5	4	1	4	5	4	2	3	5	4	1	4	5	4	1	4	30	75																
10	Rafly	21	Laki-laki	5	4	4	1	5	4	5	0	4	3	4	1	4	3	4	1	3	2	5	0	38	95																
11	Ariennova	20	Perempuan	3	2	1	4	5	4	1	4	4	3	4	1	5	4	2	3	3	2	1	4	19	47.5																
12	Nanda	20	Perempuan	4	3	2	3	4	3	1	4	3	2	2	3	5	4	3	2	5	4	1	4	31	77.5																
13	Zikriandhini	21	Perempuan	4	3	1	4	5	4	1	4	4	3	2	3	5	4	1	4	5	4	1	4	32	80																
14	Istiqomah	20	Perempuan	4	3	1	4	5	4	3	2	5	4	2	3	5	4	1	4	5	4	1	4	37	92.5																
15	Ummam	21	Laki-laki	3	2	2	3	4	3	2	3	4	3	3	2	3	2	2	3	4	3	2	3	36	90																
16	Ahmad Fathannafi	21	Laki-laki	4	3	2	3	5	4	1	4	3	2	2	3	5	4	1	4	4	3	2	3	27	67.5																
17	Allaam Muhammad Abdillah	21	Laki-laki	4	3	1	4	5	4	1	4	5	4	3	2	5	4	1	4	5	4	1	4	33	82.5																
18	Eka Aprilanti	20	Perempuan	5	4	2	3	5	4	2	3	4	3	2	3	5	4	1	4	5	4	3	4	37	92.5																
19	Isnurul Kusnaeni	52	Perempuan	5	4	2	3	5	4	2	3	5	4	2	3	5	4	1	4	5	4	3	2	34	85																
20	Rini	19	Perempuan	5	4	2	3	5	4	2	3	5	4	1	4	5	4	2	3	5	4	5	0	35	87.5																
21	vania	15	Perempuan	3	2	1	4	5	4	1	4	4	3	3	2	5	4	1	4	4	3	4	1	33	82.5																
22	Yufita Dwi Marlana	21	Perempuan	4	3	2	3	4	3	2	3	4	3	2	3	4	3	2	3	4	3	2	3	31	77.5																
23	Nurafni	20	Perempuan	4	3	2	3	4	3	2	3	4	3	2	3	5	4	2	3	3	2	3	2	30	75																
24	Renita Sendya Fadhillah	20	Perempuan	4	3	2	3	4	3	1	4	5	4	2	3	5	4	2	3	4	3	3	2	29	72.5																
25	Syifa	22	Perempuan	5	4	1	4	5	4	1	4	3	2	3	2	5	4	1	4	5	4	2	3	32	80																
26	Muhammad Riyadi AM	21	Laki-laki	3	2	2	3	4	3	1	4	5	4	1	4	5	4	1	4	5	4	3	2	35	87.5																
27	Dzakwan Anas Young Anhar	15	Laki-laki	4	3	1	4	5	4	3	2	5	4	3	2	5	4	1	4	4	3	4	1	34	85																
28	Rasyid	15	Laki-laki	3	2	1	4	4	3	3	2	3	2	4	1	4	3	2	3	2	1	3	2	31	77.5																
29	Nina Indria	21	Perempuan	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	2	3	23	57.5																
30	Naufal Yusran	21	Laki-laki	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	39	97.5																
31																								40	100																
32			Total	93			106		113		99		103		80		109		106		100		82	NILAI TOTAL	2477.5																
33			Rata-Rata	3.1			3.533		3.767		3.3		3.433		2.667		3.633		3.533		3.333		2.733	NILAI RATA-RATA	82.583333																
34																								NILAI RATA-RATA (%)	82.60%																
35						Pernyataan Positif		17.27																																	
36						Pernyataan Negatif		15.77																																	
37																																									
38																																									
39																																									

Responden 30

Ready Accessibility: Investigate

