

Pengembangan Aplikasi Mobile SkinCure Dengan Integrasi Cloud-Enabled AI untuk Deteksi Penyakit Kulit

Andrian Maulana¹, Nur Latifah Dwi Mutiara Sari¹

^{1,2}Program Studi Informatika, Universitas Persatuan Guru Republik Indonesia Semarang, Kota Semarang

*Email: andrianmaulana5612@gmail.com

Abstract.

Skin diseases have a high prevalence in Indonesia and have a significant impact on people's quality of life, but access to dermatology services is still limited, especially in remote areas. This research aims to develop the SkinCure Android application using the Rapid Application Development (RAD) method, cloud-enabled AI integration, and MVVM architecture. The RAD method is applied through the stages of needs analysis, prototyping, testing, and iterative refinement, followed by application development using Kotlin in Android Studio. Blackbox testing showed that all the main features such as authenticating, uploading skin photos, detecting skin conditions in real-time, and storing history and health information, ran smoothly and were successfully implemented. These results confirm that the RAD method is effective for the development of health applications with limited resources. Although the app already facilitates all the features in skin disease detection, further research for the development of better machine learning models, as well as more in-depth performance and security testing could be the right steps to improve the speed, privacy, and quality of the app in the future.

Keywords: Mobile Application; Skin Disease Detection; Rapid Application Development; Cloud-Enabled AI

Abstrak

Penyakit kulit memiliki prevalensi tinggi di Indonesia dan berdampak signifikan terhadap kualitas hidup masyarakat, namun akses ke layanan dermatologi masih terbatas, terutama di wilayah terpencil. Penelitian ini bertujuan mengembangkan aplikasi Android SkinCure menggunakan metode *Rapid Application Development (RAD)*, integrasi *cloud-enabled AI*, dan arsitektur *MVVM*. Metode *RAD* diterapkan melalui tahapan analisis kebutuhan, pembuatan prototipe, pengujian, dan penyempurnaan iteratif, diikuti pengembangan aplikasi menggunakan *Kotlin* di *Android Studio*. Pengujian *Blackbox* menunjukkan semua fitur utama seperti melakukan autentikasi, mengunggah foto kulit, mendeteksi kondisi kulit secara real-time, serta menyimpan riwayat dan informasi kesehatan, berjalan dengan lancar dan berhasil di implementasikan. Hasil ini menegaskan bahwa metode *RAD* efektif untuk pengembangan aplikasi kesehatan dengan sumber daya terbatas. Meskipun aplikasi sudah memfasilitasi semua fitur dalam deteksi penyakit kulit, perlu ada penelitian selanjutnya untuk pengembangan model *machine learning* yang lebih baik, serta pengujian performa dan keamanan aplikasi lebih mendalam dapat menjadi langkah tepat untuk meningkatkan kecepatan, privasi, dan kualitas aplikasi di masa depan.

Kata kunci: Aplikasi Seluler; Deteksi Penyakit Kulit; *Rapid Application Development*; *Cloud-Enabled AI*

1. Pendahuluan

Penyakit kulit merupakan salah satu masalah kesehatan yang signifikan di Indonesia, dengan prevalensi yang tinggi dan dampak yang luas terhadap kualitas hidup masyarakat. Menurut data dari RSUD Jagakarsa Jakarta dalam penelitian oleh Alfadli [2] selama periode Februari 2023 hingga Januari 2024, tercatat 1.066 kasus penyakit kulit, dengan 62,2% di antaranya merupakan penyakit non-infeksi seperti dermatitis atopik dan liken simpleks kronis, serta 39,6% merupakan penyakit infeksi seperti *tinea corporis* dan skabies [3]. Temuan ini

menegaskan perlunya strategi pencegahan terintegrasi, termasuk peningkatan edukasi kesehatan masyarakat dan perbaikan kebersihan lingkungan, untuk menurunkan beban penyakit kulit di Indonesia.

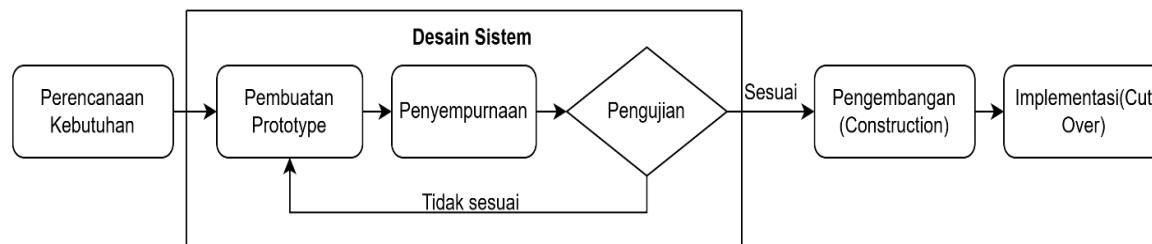
Studi lain menunjukkan bahwa penyakit kulit juga umum terjadi di lingkungan sekolah. Penelitian yang dilakukan di dua sekolah asrama di Jawa Barat pada tahun 2018 menemukan bahwa 93% siswa mengalami penyakit kulit, dengan pedikulosis 50,14% dan skabies 40,51% sebagai kondisi yang paling umum [4] [1]. Faktor lingkungan seperti kepadatan hunian dan sanitasi yang kurang memadai berkontribusi terhadap tingginya angka kejadian ini. Selain itu, penyakit kulit juga berdampak signifikan pada kelompok pekerja tertentu. Penelitian oleh Febriana [5] melakukan cross-sectional terhadap 222 pekerja batik tradisional di Yogyakarta melaporkan prevalensi *Occupational Skin Diseases* (OSD) sebesar 26,9–28,13%, dengan *Occupational Contact Dermatitis* (OCD) sebagai tipe paling dominan, mencapai 90% dari seluruh kasus OSD yang disebabkan oleh paparan bahan kimia tanpa perlindungan yang memadai. Meskipun beban penyakit kulit di Indonesia cukup tinggi, akses ke layanan dermatologi masih sangat terbatas di daerah terpencil [6]. Sebagai perbandingan rata-rata jumlah dokter spesialis kulit di Kota Jakarta mencapai 60,2 per 100.000 penduduk, sedangkan di provinsi Nusa Tenggara Timur salah satu wilayah timur Indonesia angkanya hanya 6,9 per 100.000 penduduk jauh di bawah rata-rata nasional.

Tujuan penelitian berfokus pada pengembangan aplikasi deteksi penyakit kulit yang mengintegrasikan *cloud computing* dan *machine*. Dalam penilitian mengambil konteks dari proyek capstone, tim pengembang terdiri dari tiga divisi utama seperti Machine Learning, Cloud Computing, dan Mobile Developer. Namun, ruang lingkup penelitian difokuskan pada pengembangan aplikasi Android, tanpa mencakup proses pengembangan model Machine Learning dan infrastruktur Cloud yang ditangani oleh divisi terkait. Untuk mencapai tujuan pengembangan aplikasi yang efisien dan tepat sasaran, diperlukan metode yang mampu mengakomodasi kebutuhan secara iteratif dan cepat. Salah satu metode yang digunakan dalam penelitian ini adalah metode *Rapid Application Development (RAD)*, yang dikenal efektif dalam pengembangan aplikasi. Menurut Parlika metode RAD mempermudah pengembangan karena memiliki tahapan-tahapan yang sistematis dan cepat [9]. Dijelaskan oleh Fauzi metode ini mampu menghemat waktu dan biaya karena dilakukan dalam siklus iteratif yang mencakup pembuatan prototipe, pengujian, dan penyempurnaan secara berulang hingga diperoleh hasil optimal [8]. Metode ini telah efektif digunakan dalam berbagai pengembangan aplikasi Android seperti pada penilitian oleh Subhiyakto yang mengembangkan aplikasi evaluasi kegiatan berbasis Android [10], serta oleh Nugroho dalam pengembangan game edukatif Molemash untuk platform Android [11]. Hal ini menunjukkan bahwa pendekatan RAD relevan dan adaptif terhadap pengembangan aplikasi.

2. Metode

2.1 Metode Pengembangan

Penelitian ini menggunakan metode *Rapid Application Development (RAD)*. Metode RAD dipilih karena sesuai dengan tujuan penelitian yang berfokus pada pengembangan aplikasi Android, memungkinkan proses pengembangan yang lebih fleksibel dan hanya membutuhkan sedikit waktu dalam prosesnya. Menurut Bahari dan Pramudwiyatmoko, tahapan metode RAD meliputi perencanaan kebutuhan, pembuatan prototipe, penyempurnaan, dan pengujian dalam siklus desain sistem yang dilanjutkan dengan pembangunan dan implementasi aplikasi [12]. Seperti yang diilustrasikan pada Gambar 1.



Gambar 1. Tahapan *Rapid Application Development (RAD)*

Pada Gambar 1 menggambarkan proses pengembangan perangkat lunak dengan metode *Rapid Application Development (RAD)*. Tahapan dimulai dari perencanaan kebutuhan melalui workshop internal dan *focus group* teknis. Berdasarkan hasil workshop internal dan *focus group* difokuskan ke fase *Design System* yang bersifat iteratif dengan iterasi kembali ke desain jika belum sesuai dan disetujui, dan mencakup tiga sub-tahapan: (1) *Prototyping*(pembuatan sistem), (2) *Test*(pengujian fitur), dan (3) *Refine*(penyempurnaan berdasarkan hasil uji). Tahapan dilanjutkan dengan pengembangan yang dijalankan secara cepat, dan tahap *Implementasi* dilakukan pengujian aplikasi yang sudah dibuat. Dengan metode *RAD*, dalam prosesnya dapat merespon masukan secara fleksibel dalam tahap *Design System* sebelum masuk ke tahap Pengembangan(*Construction*), dan memastikan produk akhir sesuai kebutuhan dengan melakukan pengujian dalam tahap *Implementasi*(*Cutover*).

2.2 Bahasa Pemrograman dan Alat

a) Bahasa Pemrograman

Aplikasi SkinCure dikembangkan menggunakan Kotlin karena sintaksnya yang ringkas, aman terhadap null, dan interoperable dengan Java. Hal ini mempercepat proses pengembangan dan mengurangi potensi kesalahan.

b) *Integrated Development Environment (IDE)*

Pengembangan dilakukan di Android Studio versi “LadyBug 24.2.1” yang menyediakan editor, *debugger*, dan alat build otomatis. Fitur *Virtual Device Manager* memungkinkan pengujian aplikasi tanpa perangkat fisik.

c) *Database*

Aplikasi menggunakan *SQLite* dengan *Room* untuk manajemen *database* lokal secara efisien. Untuk sinkronisasi *cloud*, digunakan *Firebase Realtime Database* yang menyediakan layanan *real-time* dan hosting terkelola.

3. Hasil dan Pembahasan

3.1 Perencanaan Kebutuhan

3.1.1 Analisis Kebutuhan Fungsional

Tabel 1 Kebutuhan Fungsional

ID Kebutuhan	Deskripsi Kebutuhan	Modul Target
KF-01	Pengguna dapat melakukan registrasi akun baru.	Autentikasi
KF-02	Pengguna dapat masuk ke aplikasi dengan akun terdaftar.	Autentikasi
KF-03	Pengguna dapat mengunggah gambar kulit untuk deteksi.	Fitur Scan
KF-04	Aplikasi menampilkan hasil deteksi penyakit kulit.	Fitur Scan
KF-05	Aplikasi menampilkan berita kesehatan kulit terkini.	Informasi Utama
KF-06	Aplikasi menyimpan riwayat deteksi pengguna.	Riwayat
KF-07	Pengguna dapat menyimpan informasi kesehatan kulit.	Bookmark
KF-08	Pengguna dapat mengedit foto, nama, dan umur.	Profile

Pada Tabel 1 menyajikan daftar kebutuhan fungsional yang menggambarkan tugas-tugas utama yang harus dapat dilakukan oleh pengguna melalui aplikasi. Setiap kebutuhan diidentifikasi dengan ID unik, disertai deskripsi yang menjelaskan aktivitas spesifik pengguna, serta modul atau fitur aplikasi yang bertanggung jawab dalam memenuhinya. Kebutuhan fungsional mencakup interaksi langsung pengguna dengan sistem, kebutuhan-kebutuhan ini

dirancang agar aplikasi dapat memberikan fungsionalitas utama yang dibutuhkan sesuai dengan tujuan pengembangan.

3.1.2 Analisis Kebutuhan Non Fungsional

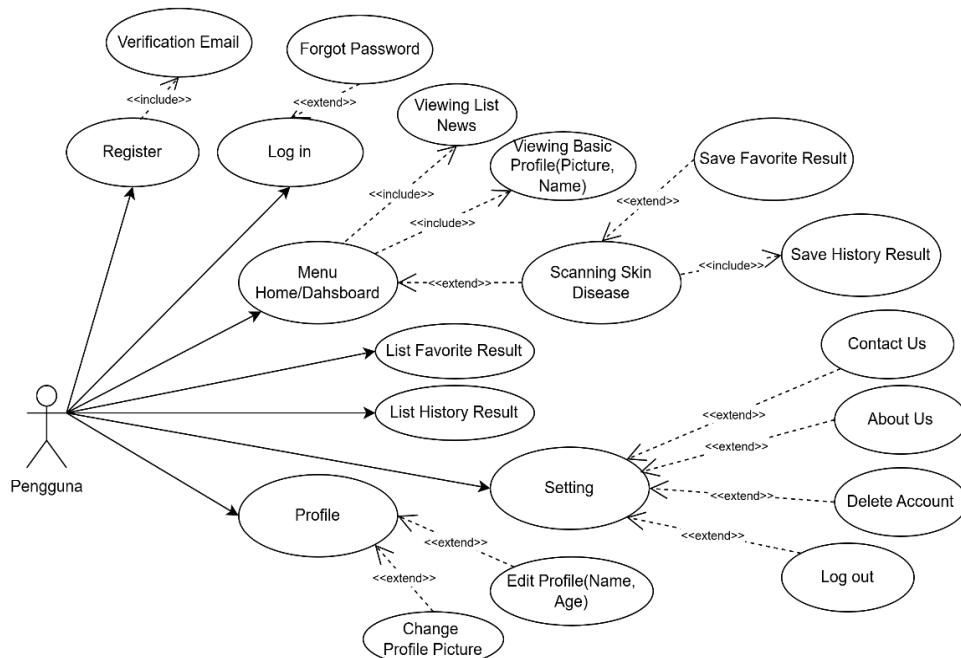
Tabel 2 Kebutuhan Non Fungsional

Aspek	Deskripsi Kebutuhan	Metrik/Target
Performa	Aplikasi harus responsif dan cepat dalam deteksi.	Waktu respons deteksi < 5 detik.
Keamanan	Data pengguna dan gambar harus aman.	Enkripsi data, otentikasi dua faktor (opsional).
Usability	Antarmuka mudah digunakan dan intuitif.	Tidak ada bug dalam tampilan UI di xml
Kompatibilitas	Aplikasi berjalan di berbagai versi Android.	Mendukung Android 7.0 (Nougat) ke atas.

Pada tabel 2 berisi kebutuhan non fungsional yang bertujuan untuk memastikan bahwa aplikasi tidak hanya berfungsi secara teknis, tetapi juga mampu bekerja secara optimal dalam aspek performa, keamanan, dan kemudahan penggunaan (*usability*).

3.2 Desain Sistem

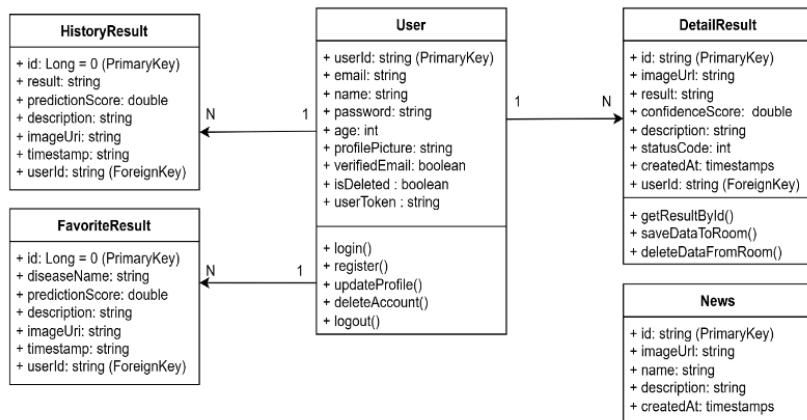
3.2.1 Usecase Diagram



Gambar 2. Usecase Diagram

Gambar 2 menunjukkan diagram use case dari aplikasi yang akan dikembangkan. Dalam diagram ini, pengguna dapat mengakses berbagai fitur yang tersedia setelah melakukan proses registrasi atau login terlebih dahulu. Diagram ini menunjukkan hubungan antara pengguna dan semua fitur utama yang mendukung fungsionalitas menyeluruh.

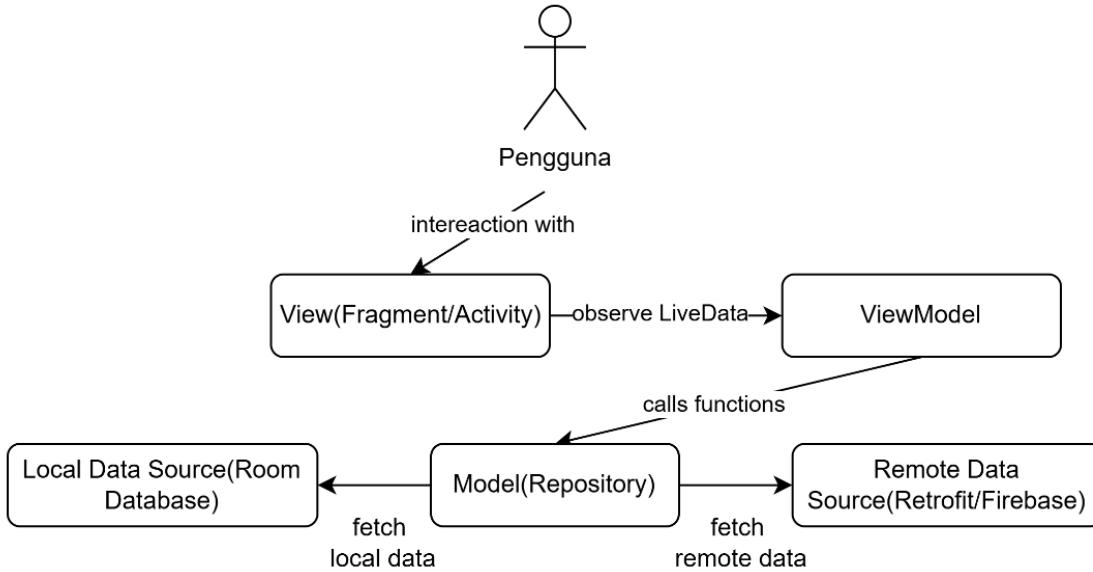
3.2.2 Class Diagram



Gambar 3 Class Diagram

Gambar 3 memperlihatkan struktur entitas dan relasi utama pada rancangan awal sistem. Entitas *User* berelasi *one-to-many* dengan *HistoryResult* (menyimpan riwayat hasil) dan *FavoriteResult* (menyimpan hasil favorit). Setiap *HistoryResult* kemudian memiliki relasi *one-to-many* dengan *DetailResult* untuk rincian data. Sementara itu, entitas *News* berdiri terpisah sebagai penyimpanan konten berita yang dapat dihubungkan ke *User* sesuai kebutuhan aplikasi.

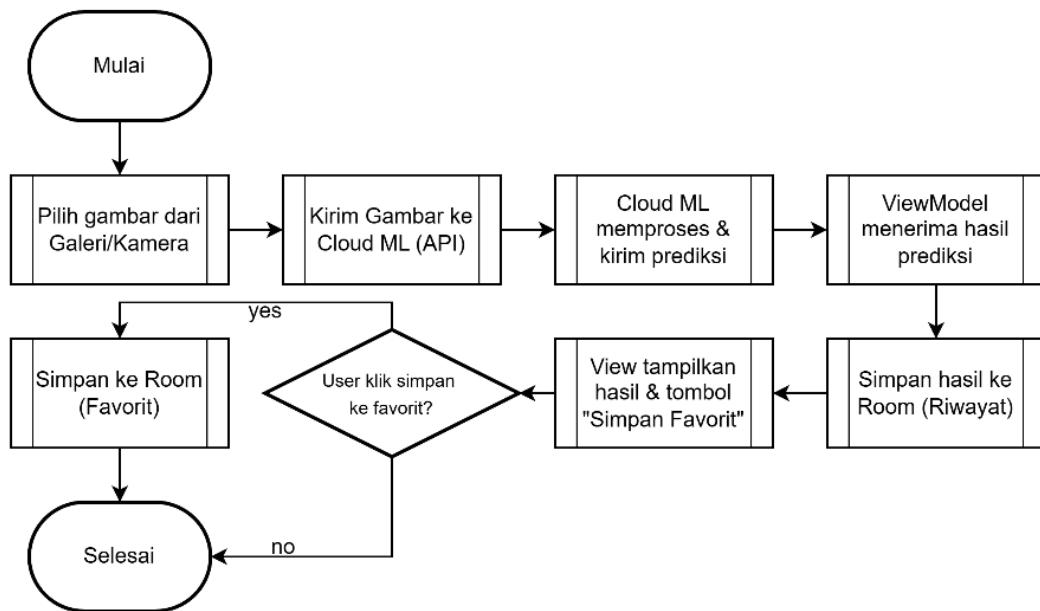
3.2.3 Arsitektur MVVM



Gambar 4 Desain Arsitektur MVVM

Gambar 4 menunjukkan arsitektur MVVM(*Model View View Model*) yang digunakan dalam pengembangan aplikasi. *View* (Fragment/Activity) berperan sebagai antarmuka pengguna yang mengamati data dari *ViewModel* melalui *LiveData*. *ViewModel* menjadi penghubung antara *View* dan *Repository*, serta mengelola *state UI*. *Repository* mengakses data dari dua sumber, yaitu *Remote Data Source* seperti *API Cloud* atau *Firebase* dan *Local Data Source* menggunakan *Room Library*, sehingga pemisahan logika dan pengelolaan data lebih terstruktur.

3.2.4 Diagram Alur Fitur Utama

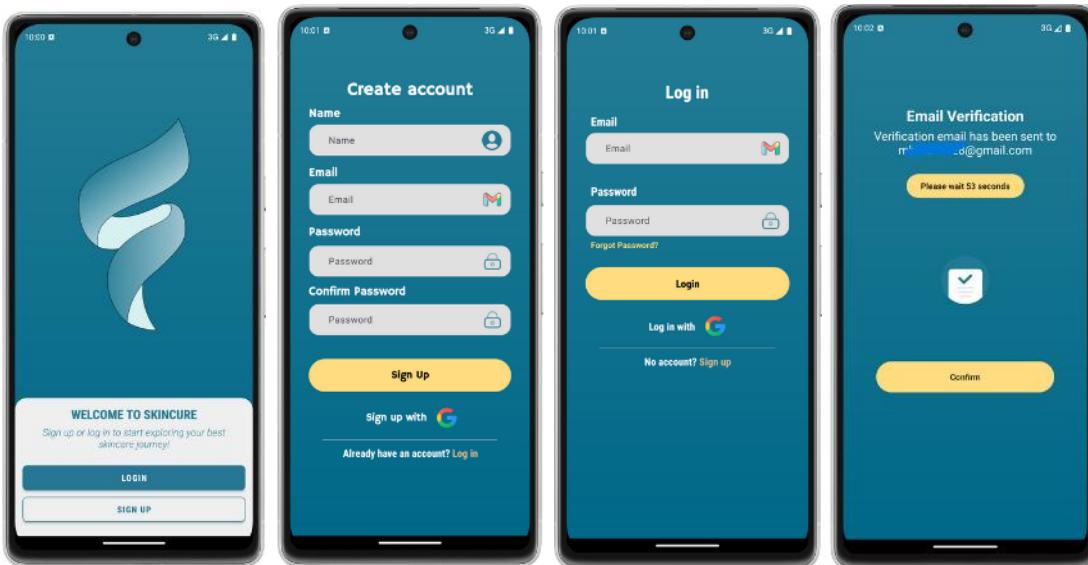


Gambar 5. Diagram Alur Fitur Utama

Pada Gambar 5 diagram menggambarkan alur proses pengguna saat menggunakan fitur utama aplikasi, yaitu mendeteksi penyakit kulit berdasarkan gambar dari galeri atau kamera yang di upload oleh pengguna.

3.3 Pengembangan (Construction)

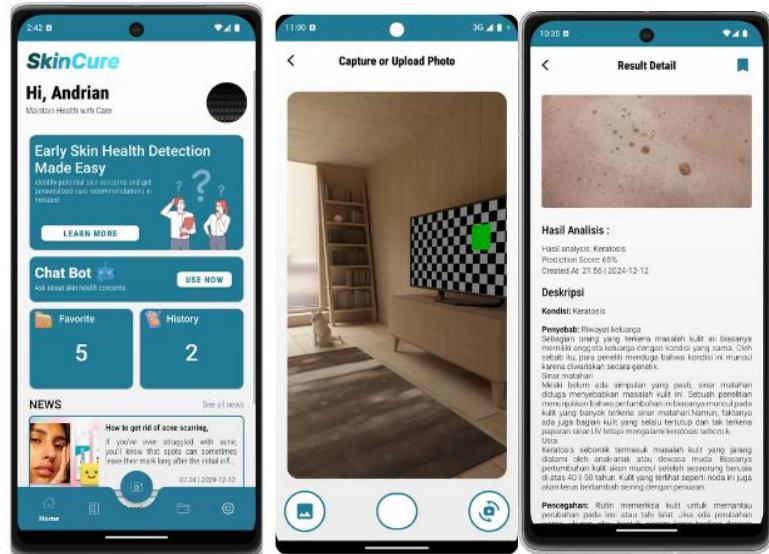
Setelah proses *Design System* dilanjutkan di fase pengembangan yang mengimplementasikan tampilan antarmuka yang telah dibuat di Figma dan kode Kotlin ke dalam aplikasi yang akan dapat digunakan oleh pengguna atau penguji. Desain antarmuka high-fidelity yang telah disusun di Figma berhasil direalisasikan dalam bentuk XML(*Extensible Markup Language*) pada Android Studio. Pengembangan tampilan XML dilakukan pada android studio. Berikut hasil pengembangan dari aplikasi deteksi kulit.



Gambar 6. Halaman Welcome App, Registrasi, Login, dan Verifikasi Email

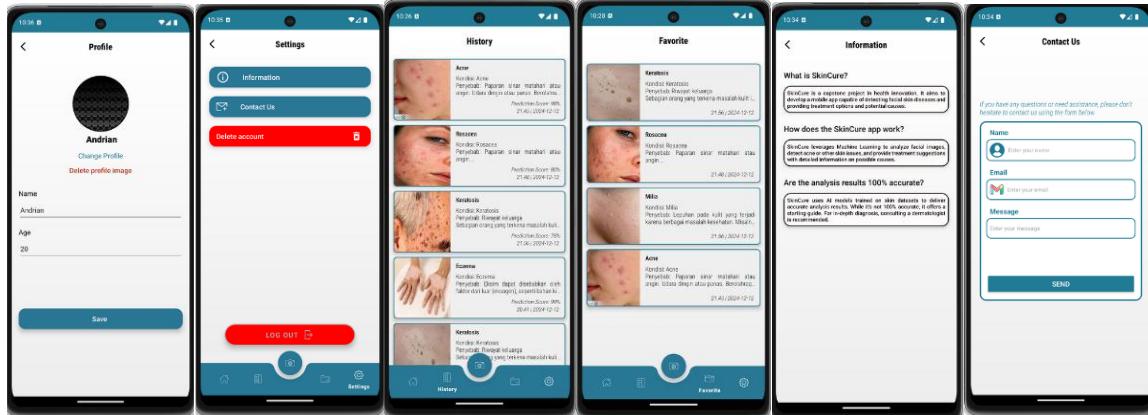
Pada Gambar 6 halaman login, registrasi, dan verifikasi email digunakan oleh pengguna untuk melakukan autentikasi pada aplikasi deteksi penyakit kulit, metode registrasi dan login juga terintregasi dengan *Google Authentication* dari Firebase menggunakan akun google. Fitur

lupa password dan verifikasi email juga ditambahkan untuk memudahkan dan mengamankan autentikasi pengguna.



Gambar 7. Dahboard Home, Halaman Scan, Hasil Deteksi

Pada Gambar 7 terdapat dashboard utama, fitur scan deteksi penyakit kulit, dan hasil deteksi, bagian tersebut adalah fitur utama dari aplikasi yang dimana user melakukan upload foto dari galeri atau dari kamera kemudian mengeluarkan hasil yang di tampilkan pada halaman hasil deteksi.



Gambar 8 Halaman Profile, Setting, History, Favorite, About Us, Contact Us

Terdapat pada Gambar 8 bagian-bagian yang mendukung seluruh fitur aplikasi. Berdasarkan hasil evaluasi awal dengan divisi lain, dilakukan perbaikan pada elemen navigasi dan kontras warna untuk aksesibilitas.

3.4 Implementasi (Cutover)

Pada tahap ini, dilakukan pengujian menggunakan metode pengujian *Blackbox*. Pengujian *Blackbox* untuk menilai fungsi implementasi aplikasi. Tes ini berfokus pada masukan dan keluaran perangkat lunak guna memastikan aplikasi berfungsi sebagaimana yang telah ditentukan. Berikut ini, Tabel 1 menunjukkan hasil dari pengujian *Blackbox*.

Tabel 3 Hasil Pengujian *Blackbox*

Interaksi pengguna	Hasil yang diharapkan	Hasil Tes
Memilih Login atau Registrasi pada Halaman Welcome App	Dapat berpindah halaman Login dan Registrasi	✓
Melakukan Login dan Registrasi	Setelah berhasil login atau registrasi, berpindah halaman ke Dashboard	✓

Klik tombol Scan(icon kamera) pada dashboard	Pindah halaman ke preview kamera dan muncul tombol galeri	✓
Mengunggah gambar kulit untuk deteksi	Pindah ke halaman hasil deteksi penyakit dan menyimpan ke History	✓
Di halaman hasil deteksi pengguna klik tombol Favorite(bookmark)	Data hasil deteksi tersimpan Favorite ke Room(database lokal)	✓
Klik Profile dan mengedit foto, nama, umur	Muncul halaman profile dan data berhasil di update	✓
Memilih menu setting, history, favorite	Menampilkan halaman setting, list history, favorite	✓

Berdasarkan pengujian black box pada tabel 3 yang dilakukan, dari 7 skema pengujian didapatkan hasil yang sesuai dan berhasil. Artinya keseluruhan sistem fungsional berjalan dengan baik dan sesuai skema pengujian.

4. Kesimpulan

Penelitian ini menunjukkan bahwa pengembangan aplikasi Android untuk deteksi penyakit kulit menggunakan metode *Rapid Application Development (RAD)* telah berhasil mencapai tujuannya dengan pengembangan menggunakan Kotlin di Android Studio. Aplikasi ini menyediakan fitur seperti autentikasi pengguna(login dan registrasi), verifikasi email, *dashboard/menu home, profile, favorite hasil, setting*(logout dan hapus akun), unggah dan deteksi gambar kulit, penyimpanan riwayat deteksi, dan berita terkini tentang kesehatan kulit. Arsitektur *MVVM* yang digunakan memudahkan pemeliharaan kode dan pengujian. Pengujian *Blackbox* menunjukkan fitur utama berfungsi baik dan siap untuk uji lapangan. Keberhasilan ini menegaskan bahwa metode *RAD* memungkinkan desain yang membutuhkan sedikit waktu dan efektif.

Beberapa bug pada aplikasi saat fase pengembangan(*construction*) seperti pada fitur penyimpanan gambar di perangkat Android 12, yang berhasil diatasi dengan penyesuaian *permission* dan bug hasil analisis yang disimpan tersimpan 2 kali pada halaman riwayat hasil yang berhasil diatasi dengan menggunakan *database* lokal *Room SQLite*. Uji coba dengan pengguna terbatas(penguji) menunjukkan bahwa pengguna merasa terbantu dengan kehadiran fitur diagnosis cepat. Langkah penelitian selanjutnya dapat meliputi pengembangan *machine learning* yang lebih baik, uji coba lapangan, serta pengujian performa dan keamanan yang lebih detail.

Ucapan terimakasih dari peneliti kepada anggota tim yang telah berkontribusi pada proyek Skincure, karena telah berkolaborasi membangun aplikasi yang bermanfaat untuk kesehatan. Ucapan terimakasih juga kepada para mentor dan penguji yang telah banyak memberi bimbingan dan masukan terhadap proyek ini, sehingga aplikasi ini berhasil dibuat sampai selesai.

5. Referensi

- [1]. S. Widaty et al., "Characteristic of Skin Diseases in Two Public Boarding Schools Occupants in West Java 2018," *EJournal Kedokteran Indonesia*, vol. 11, no. 1, p. 14, 2023. <https://doi.org/10.23886/ejki.11.217.14>
- [2]. R. Alfadli and S. Khairunisa, "Prevalensi penyakit kulit infeksi dan non-infeksi di poliklinik kulit dan kelamin RSUD Jagakarsa periode Februari 2023–Januari 2024," *J. Kedokteran Meditek*, vol. 30, no. 3, Art. no. 3254, 2024. <https://ejournal.ukrida.ac.id/index.php/Meditek/article/view/3254>
- [3]. A. Gunawan, F. A. Syahril, M. Sari, and A. Triyani, "Gambaran Penyakit Kulit di RSUD Jagakarsa Jakarta Februari 2023–Januari 2024," *Meditek: J. Ilm. Kesehatan*, vol. 10, no. 1, pp. 1–7, 2024. <https://ejournal.ukrida.ac.id/index.php/Meditek/article/view/3254>
- [4]. R. Faradz, D. Indriani, and B. A. Pribadi, "Prevalensi Penyakit Kulit Menular pada Anak Sekolah Asrama di Jawa Barat," *eJ. Kedokteran Indonesia (EJKI)*, vol. 10, no. 2, pp. 78–85, 2022. <https://ejki.fk.ui.ac.id/index.php/jurnal/article/view/217>

-
- [5]. S. A. Febriana et al., "Risk factors of occupational skin diseases among traditional batik manufacturing workers in Yogyakarta, Indonesia," *BMC Res. Notes*, vol. 16, no. 2, 2023. <https://doi.org/10.1186/s13104-022-06105-0>
 - [6]. F. J. Adella et al., "Teledermatology to improve access to and quality of skin care in Eastern Indonesia," *Am. J. Trop. Med. Hyg.*, vol. 110, no. 2, pp. 364–369, 2024. <https://doi.org/10.4269/ajtmh.23-0218>
 - [7]. N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," in *Ind. Res. Workshop Nat. Sem.*, vol. 9, 2018. <https://jurnal.polban.ac.id/proceeding/article/view/1116>
 - [8]. M. A. Fauzi, H. Tribiakto, A. Moniva, F. Amir, I. K. Ilyas, and E. Utami, "Systematic Literature Reviews on Rapid Application Development Information System," *Bull. Comput. Sci. Electr. Eng.*, vol. 4, no. 1, pp. 57–64, 2023. <https://bcsee.org/index.php/bcsee/article/view/1181/39>
 - [9]. R. Parlika, M. Afifudin, I. A. Pradana, Y. D. W. Wiratama, and M. N. Holis, "Studi literatur efisiensi model Rapid Application Development dalam pengembangan perangkat lunak (2014–2022)," *Positif: J. Sistem dan Teknol. Inform.*, vol. 9, no. 1, pp. 1–10, 2023. <https://pdfs.semanticscholar.org/a899/8425b84ffbb866105208bd6bd64f3da57570.pdf>
 - [10]. E. R. Subhiyakto, F. Agustina, and C. V. Reswara, "Pengembangan Aplikasi Evaluasi Kegiatan Berbasis Android menggunakan Metode RAD (Rapid Application Development)," *J. Ilm. Intech: Information Technology Journal of UMUS*, vol. 5, no. 1, pp. 49–59, 2023. <https://doi.org/10.46772/intech.v5i1.1082>
 - [11]. A. C. Nugroho, "Pengembangan Game Molemash di Android Menggunakan Metode Rapid Application Development," *J. Appl. Comput. Sci. Technol.*, vol. 1, no. 1, pp. 1–6, 2020. <https://doi.org/10.52158/jacost.v1i1.6>
 - [12]. A. F. Bahari and A. Pramudwiatmoko, "Implementation of Rapid Application Development (RAD) Method for Mobile-Based Ice Cream Ordering Application," *Malcom: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 1, pp. 283–291, 2025. <https://doi.org/10.57152/malcom.v5i1.1747>