



KERJA PRAKTIK - EF234603

Pengembangan Aplikasi Mobile SWiDS

Klinik Utama Eagles HEAD Medical Centre
Jl. Seruni No.38, Ketabang, Kec. Genteng, Kota Surabaya, Jawa
Timur 60275

Periode: 14 April 2024 - 30 September 2024

Oleh:

Afiq Fawwaz Haidar	5025211246
Syarifah Talitha Erfany	5025211175

Pembimbing Departemen

Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

Pembimbing Lapangan

Dr. Shoffi Izza Sabilla, S.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024



KERJA PRAKTIK - EF234603

Pengembangan Aplikasi Mobile SWiDS

Klinik Utama Eagles HEAD Medical Centre

Jl. Seruni No.38, Ketabang, Kec. Genteng, Kota Surabaya, Jawa Timur
60275

Periode: 14 April 2024 - 30 September 2024

Oleh:

Afiq Fawwaz Haidar 5025211246

Syarifah Talitha Erfany 5025211175

Pembimbing Departemen

Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc.

Pembimbing Lapangan

Dr. Shoffi Izza Sabilla, S.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
LEMBAR PENGESAHAN	xii
KATA PENGANTAR	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik	3
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur	3
1.6.3. Analisis dan Perancangan Sistem	4
1.6.4. Implementasi Sistem	4
1.6.5. Pengujian dan Evaluasi	4
1.6.6. Kesimpulan dan Saran	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan	4
1.7.2. Bab II Profil Perusahaan	5

1.7.3.	Bab III Tinjauan Pustaka	5
1.7.4.	Bab IV Infrastruktur Sistem	5
1.7.5.	Bab V Implementasi Sistem.....	5
1.7.6.	Bab VI Pengujian dan Evaluasi.....	5
1.7.7.	Bab VII Kesimpulan dan Saran	5
BAB II PROFIL PERUSAHAAN		7
2.1.	Profil Klinik Utama Eagles HEAD Medical Centre	7
2.2.	Lokasi	8
BAB III TINJAUAN PUSTAKA.....		9
3.1.	Pemrograman Mobile	9
3.2.	Dart	9
3.3.	Flutter	10
3.4.	Firestore.....	10
3.5.	Vertabelo.....	11
3.6.	Figma.....	11
BAB IV INFRASTRUKTUR SISTEM.....		13
4.1.	Desain Sistem	13
4.2.	Fungsionalitas Aplikasi SWiDS	15
BAB V IMPLEMENTASI SISTEM.....		19
5.1.	Fitur Login dan Forgot Password	19
5.2.	Fitur Integrasi Stetoskop	22
5.3.	Fitur Rekam Medis	27
5.4.	Fitur Histori Rekam Medis.....	32

5.5.	Fitur Daftar Pasien	36
5.6.	Fitur Profil	42
BAB VI PENGUJIAN DAN EVALUASI.....		45
6.1.	Tujuan Pengujian	45
6.2.	Kriteria Pengujian	45
6.3.	Skenario Pengujian.....	45
6.4.	Evaluasi Pengujian	47
BAB VII KESIMPULAN DAN SARAN		49
7.1.	Kesimpulan	49
7.2.	Saran	49
DAFTAR PUSTAKA		51
BIODATA PENULIS I.....		55

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Klinik	7
Gambar 4.1 Desain Arsitektur Aplikasi SWiDS	13
Gambar 5.1 Tampilan Halaman Login.....	20
Gambar 5.2 Tampilan Halaman Forgot Password.....	21
Gambar 5.3 Tampilan Halaman Scan QR.....	24
Gambar 5.4 Tampilan Halaman Mengaktifkan Stetoskop	26
Gambar 5.5 Tampilan Halaman Menonaktifkan Stetoskop	27
Gambar 5.6 Tampilan Halaman Memilih Pasien saat Rekam Medis	29
Gambar 5.7 Tampilan Halaman Memasukan Keluhan saat Rekam Medis	31
Gambar 5.8 Tampilan Halaman Mendengar dan Menyimpan Rekam Medis.....	32
Gambar 5.9 Tampilan Halaman Histori Rekam Medis.....	34
Gambar 5.10 Tampilan Halaman Detail Rekam Medis	36
Gambar 5.11 Tampilan Halaman Daftar Pasien	38
Gambar 5.12 Tampilan Halaman Detail Pasien	40
Gambar 5.13 Tampilan Halaman Mendaftarkan Pasien Baru	42
Gambar 5.14 Tampilan Halaman Profil dan Edit Profil.....	44

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 6.1 Hasil Evaluasi Pengujian	47
--	----

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

Pengembangan Aplikasi Mobile SWiDS

Oleh:

Afiq Fawwaz Haidar
Syarifah Talitha Erfany

5025211246
5025211175

Disetujui oleh Pembimbing Kerja Praktik:

1. Ir. Adhatus Solichah
Ahmadiyah, S.Kom, M.Sc.
NIP. 198508262015042002



(Pembimbing Departemen)

2. Dr. Shoffi Izza Sabilla,
S.Kom.
NIP. 2022199412054



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Pengembangan Aplikasi Mobile SWiDS

Nama Mahasiswa : Afiq Fawwaz Haidar
NRP : 5025211246
Nama Mahasiswa : Syarifah Talitha Erfany
NRP : 5025211175
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Ir. Adhatus Solichah Ahmadiyah,
S.Kom, M.Sc.
Pembimbing Lapangan : Dr. Shoffi Izza Sabilla, S.Kom.

ABSTRAK

Aplikasi SWiDS dikembangkan untuk membantu tenaga kesehatan dalam pengelolaan data medis pasien. Aplikasi ini terintegrasi dengan stetoskop digital nirkabel, memungkinkan tenaga kesehatan untuk merekam, menyimpan, dan menganalisis suara medis dari jantung, paru-paru, dan abdomen secara real-time. Data yang dihasilkan disimpan dalam Firebase Storage dan Firestore, sehingga mudah diakses oleh tenaga kesehatan kapan saja.

Proyek ini menggunakan framework Flutter dan arsitektur Model-View-Controller (MVC) untuk menangani bagian frontend dan sebagian backend, lalu digunakan Firebase sebagai backend utama. Beberapa microservice dikembangkan menggunakan Python untuk memenuhi kebutuhan khusus yang tidak dapat ditangani oleh Firebase. Fitur utama aplikasi meliputi perekaman suara stetoskop, pengelolaan riwayat medis, manajemen daftar pasien, serta integrasi dengan perangkat medis lainnya.

Kata Kunci : Mobile, Flutter, Firebase, Stetoskop Digital, Kesehatan

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Pengembangan Aplikasi Mobile SWiDS

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Ibu Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc. selaku dosen pembimbing departemen selama kerja praktik.
3. Dr. Shoffi Izza Sabilla, S.Kom. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 7 Oktober 2024

Afiq Fawwaz Haidar dan Syarifah Talitha Erfany

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Aplikasi *mobile* menjadi perangkat komunikasi pribadi yang sering dibawa dan dapat secara aktif atau pasif memiliki data terkait kesehatan pasien. Cara para profesional kesehatan berinteraksi dengan aplikasi kesehatan telah berkembang seiring waktu (Butcher & Hussain, 2022). Salah satu bentuk interaksi aplikasi *mobile* adalah dengan stetoskop yang terintegrasi, sebuah stetoskop digital nirkabel yang memungkinkan pemantauan pasien. Alat ini juga dapat digunakan bersamaan dengan *wearable wireless electrocardiogram* (EKG) nirkabel yang dapat dipakai dan monitor denyut nadi untuk memberikan informasi kesehatan yang lebih lengkap tentang pasien. Sinyal yang direkam dari stetoskop digital berkabel dan nirkabel dibandingkan berdasarkan kesalahan kuadrat rata-rata dalam domain waktu, distribusi spektralnya dalam domain frekuensi, serta analisis dalam domain waktu-frekuensi menggunakan *short-time Fourier transform* (STFT) (Shi et al., 2015).

Aplikasi *mobile* untuk kesehatan dirancang untuk menyediakan layanan untuk tenaga kesehatan, seperti bantuan dalam diagnosis pada *wearable wireless electrocardiogram*. Oleh karena itu, aplikasi *mobile* sangat penting, di mana informasi dan layanan dapat diakses oleh tenaga kesehatan dimana saja dan kapan saja (Gagnon et al., 2016).

Proyek kami berfokus pada pengembangan aplikasi SWiDS, sebuah solusi inovatif untuk membantu tenaga kesehatan dalam pemantauan dan pengelolaan data medis. Aplikasi ini dirancang untuk mengintegrasikan

teknologi stetoskop digital nirkabel dengan sistem mobile, memungkinkan pengguna untuk merekam dan menyimpan data kesehatan secara real-time. SWiDS menghubungkan berbagai stakeholder, termasuk tenaga medis, pasien, dan fasilitas kesehatan, guna menciptakan ekosistem pelayanan kesehatan yang lebih efektif dan efisien. Dengan dukungan fitur seperti perekaman suara stetoskop, pengelolaan riwayat medis, dan integrasi dengan perangkat medis lainnya, SWiDS bertujuan meningkatkan kualitas diagnosis dan perawatan pasien secara lebih komprehensif.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan membantu Klinik Utama Eagles HEAD Medical Centre untuk mengembangkan aplikasi SWiDS terhadap permintaan yang ada dari pada stakeholder aplikasi.

1.3. Manfaat

Manfaat aplikasi SWiDS adalah mempermudah tenaga kesehatan untuk berkonsultasi dengan dokter spesialis pada bagian paru-paru, jantung, dan abdomen terkait keluhan pasien. Aplikasi ini memiliki rekaman dari stetoskop yang terintegrasi, sehingga dokter dapat membantu tenaga kesehatan dalam memberikan diagnosis kepada pasien.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara merancang aplikasi SWiDS agar sesuai dengan kebutuhan tenaga kesehatan dan di lapangan?

2. Bagaimana mengintegrasikan stetoskop dengan aplikasi SWiDS agar dapat merekam dan menyimpan data medis secara akurat?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilakukan secara *remote* dengan waktu yang *fleksibel*. Untuk rapat pada saat kerja praktik, dilaksanakan secara *online* di Zoom dan *offline* di Fakultas Kedokteran dan Kesehatan

Adapun kerja praktik dimulai pada tanggal 14 April 2024 hingga 30 September 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan dari aplikasi, kami mengikuti rapat bulanan bersama tim dari Fakultas Kedokteran dan Kesehatan yang memiliki permintaan untuk pembuatan aplikasi yang dapat disambungkan dengan stetoskop yang terintegrasi dengan aplikasi. Pada saat rapat, kami melakukan diskusi mengenai alur aplikasi dan fitur-fitur yang akan diterapkan pada aplikasi yang akan dibuat. Setelah itu, kami merancang desain aplikasi hingga desain tersebut diterima, dan kemudian memulai tahap pengembangan aplikasi.

1.6.2. Studi Literatur

Setelah memahami bagaimana aplikasi akan bekerja, kami diberikan tinjauan mengenai hal-hal yang akan diimplementasikan untuk membuat aplikasi dapat beroperasi. Tinjauan tersebut meliputi Firebase, Flutter, lain-lain. Kami juga mengeksplorasi berbagai library dari Flutter yang membantu dalam pengembangan aplikasi. Pada Firebase, kami menggunakan fitur

Authentication, Firestore Database, Realtime Database, dan Storage.

1.6.3. Analisis dan Perancangan Sistem

Langkah selanjutnya adalah merancang sistem dengan desain arsitektur yang baik. Aplikasi kami menggunakan pendekatan arsitektur MVC (Model-View-Controller) pada framework Flutter. Flutter menangani semua bagian frontend (mobile) dan sebagian besar fungsi backend, memanfaatkan Firebase sebagai backend utama karena telah menyediakan SDK yang mempermudah integrasi.

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini, kami melakukan pembuatan aplikasi sesuai dengan fitur-fitur dan alur yang diinginkan. Setelah tahap development selesai, kami melakukan *deployment* terhadap *service* yang digunakan.

1.6.5. Pengujian dan Evaluasi

Setelah aplikasi sudah jadi, dilakukan evaluasi pengujian apakah aplikasi sesuai dengan harapan *client*. Jika masih belum sesuai atau menambah fitur, maka akan dilakukan penambahan atau perbaikan fitur.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Klinik Utama Eagles HEAD Medical Centre mulai dari profil, lokasi klinik.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Infrastruktur Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses pengembangan aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

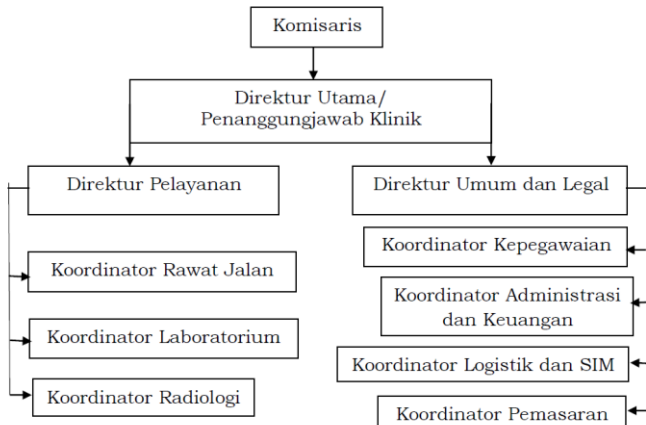
[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Klinik Utama Eagles HEAD Medical Centre

Klinik Utama Eagles HEAD Medical Centre, yang didirikan pada tahun 2018, mengedepankan pelayanan kesehatan berkualitas tinggi dengan pendekatan kekeluargaan dan persahabatan. Dengan visi menjadi klinik utama yang unggul dalam mendukung kesehatan pelanggan, Eagles HEAD Medical Centre terus berupaya memberikan pelayanan terbaik. Moto yang diusung, “Melayani Dengan Hati dan Ketrampilan Tinggi,” menggambarkan komitmen kuat seluruh tenaga medis dalam melayani pasien dengan penuh perhatian dan profesionalisme.



Gambar 2.1 Struktur Organisasi Klinik

Klinik ini menawarkan berbagai layanan medis, termasuk pelayanan rawat jalan yang mencakup spesialis

saraf, spesialis penyakit dalam, serta konsultasi dengan spesialis patologi klinik. Selain itu, tersedia fasilitas treadmill dan ECG, laboratorium klinik dengan pemeriksaan kimia dan hematologi, serta layanan radiologi dengan peralatan modern seperti X-ray dan USG. Klinik ini didukung oleh sumber daya manusia yang mumpuni, terdiri dari empat dokter, satu perawat, tiga analis kesehatan, dua radiografer, dan satu tenaga administrasi serta teknologi informasi. Semua ini menjamin pelayanan kesehatan yang efisien dan akurat.

Eagles HEAD Medical Centre berkomitmen pada peningkatan mutu layanan dengan memantau indikator nasional seperti kepatuhan kebersihan tangan, penggunaan alat pelindung diri, identifikasi pasien yang tepat, dan tingkat kepuasan pasien. Indikator mutu internal juga menjadi perhatian, termasuk angka tidak terjadinya pengulangan foto rontgen, kesalahan hasil laboratorium, dan kesalahan pembayaran pasien. Penyakit-penyakit yang paling sering ditangani di klinik ini meliputi diabetes melitus, bronkitis, dispepsia, dislipidemia, dan lainnya, menunjukkan keahlian klinik dalam menangani berbagai kasus medis.

Klinik ini fokus pada edukasi kesehatan pasien dan menjaga hubungan erat berbasis kepercayaan. Beroperasi setiap hari dari pukul 6 pagi hingga 9 malam, klinik ini memastikan akses mudah bagi pasien.

2.2. Lokasi

Jl. Seruni No.38, Ketabang, Kec. Genteng, Kota Surabaya, Jawa Timur 60272

BAB III

TINJAUAN PUSTAKA

3.1. Pemrograman Mobile

Mobile programming atau pemrograman aplikasi bergerak telah berkembang pesat seiring meningkatnya penggunaan smartphone. Mobile programming adalah proses pengembangan aplikasi yang dirancang untuk perangkat mobile seperti smartphone atau tablet, umumnya untuk sistem operasi Android atau iOS. Bahasa pemrograman yang populer dalam mobile programming termasuk Java dan Kotlin untuk Android, serta Swift untuk iOS. Selain itu, bahasa seperti C#, JavaScript, dan Dart digunakan dalam pengembangan aplikasi lintas platform dengan kerangka kerja seperti Xamarin, React Native, dan Flutter. Pilihan bahasa tergantung pada platform yang ditargetkan dan preferensi pengembang, dengan setiap bahasa memiliki kelebihan dan kekurangan. Dengan pemahaman yang baik tentang mobile programming, programmer dapat menciptakan aplikasi mobile yang inovatif dan efisien (*Mobile Accessibility - Learn Web Development / MDN, 2024*).

3.2. Dart

Dart adalah bahasa pemrograman yang dikembangkan oleh Google. Dart dirancang untuk pembangunan aplikasi modern, baik di mobile, web, desktop, maupun server. Dalam pengembangan mobile, Dart menjadi bahasa utama untuk Flutter, kerangka kerja yang memungkinkan pembuatan aplikasi lintas platform. Dart memiliki sintaks yang sederhana dan mendukung pemrograman berorientasi objek, menjadikannya pilihan

yang efektif untuk membangun aplikasi dengan kinerja tinggi (*Dart Programming Language / Dart*, n.d.).

3.3. Flutter

Flutter adalah kerangka kerja pengembangan aplikasi yang dikembangkan oleh Google untuk membangun aplikasi mobile, web, dan desktop dari satu basis kode. Flutter menggunakan bahasa pemrograman Dart dan memungkinkan pengembang untuk membuat antarmuka pengguna yang responsif dan dinamis dengan widget yang dapat disesuaikan. Flutter menawarkan kinerja yang tinggi dengan kemampuan rendering native dan akses langsung ke API platform, memungkinkan pengembang untuk menciptakan aplikasi yang mulus dan interaktif (*Flutter - Build Apps for Any Screen*, n.d.).

3.4. Firebase

Firebase adalah platform pengembangan aplikasi berbasis cloud yang disediakan oleh Google. Firebase menyediakan berbagai layanan seperti autentikasi pengguna, database real-time, analitik, penyimpanan file, dan layanan notifikasi push yang memudahkan pengembang dalam membangun dan mengelola aplikasi. Firebase mendukung integrasi dengan berbagai platform seperti Android, iOS, dan web, menjadikannya solusi yang fleksibel dan scalable untuk aplikasi modern.

Firebase menyediakan SDK yang lengkap dan terintegrasi dengan Flutter, yang memungkinkan pengembang untuk memanfaatkan berbagai layanan Firebase secara langsung dalam aplikasi Flutter. SDK ini mencakup autentikasi pengguna, integrasi dengan Firestore dan Realtime Database, penyimpanan file menggunakan Firebase Storage, serta fitur analitik dan

notifikasi push. Integrasi dengan Flutter memastikan bahwa layanan-layanan tersebut dapat digunakan dengan efisien dan memberikan performa optimal pada aplikasi lintas platform (*Dokumentasi Firebase*, n.d.).

3.5. Vertabelo

Vertabelo adalah alat desain database berbasis web yang memungkinkan pengembang untuk membuat, memvisualisasikan, dan mengelola struktur database secara efisien. Alat ini mendukung berbagai jenis database, termasuk SQL Server, PostgreSQL, MySQL, dan lainnya. Vertabelo memungkinkan kolaborasi tim dalam desain database dan memudahkan pembuatan diagram entitas-relasi yang diperlukan untuk perancangan sistem (*Vertabelo*, n.d.).

3.6. Figma

Figma adalah alat desain antarmuka pengguna berbasis cloud yang populer di kalangan pengembang dan desainer UI/UX. Figma memungkinkan kolaborasi secara real-time, yang memudahkan tim untuk merancang dan memodifikasi desain aplikasi secara bersama-sama. Alat ini mendukung pembuatan prototipe, wireframe, dan desain antarmuka yang interaktif, serta mempermudah pengembang dalam menerapkan desain ke dalam kode aplikasi (*Figma*, n.d.).

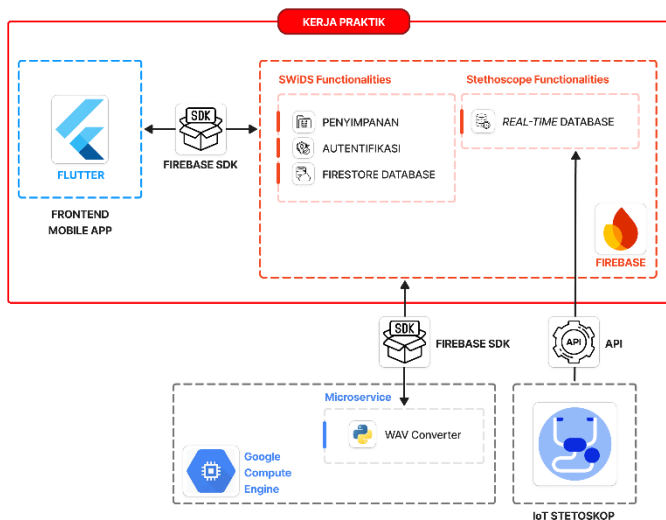
[Halaman ini sengaja dikosongkan]

BAB IV

INFRASTRUKTUR SISTEM

4.1. Desain Sistem

Desain arsitektur pada aplikasi SWiDS ini terbagi menjadi beberapa komponen utama, sebagaimana dijelaskan berikut.



Gambar 4.2 Desain Arsitektur Aplikasi SWiDS

Aplikasi ini menggunakan Flutter untuk pengembangan antarmuka pengguna lintas platform dan Firebase untuk memenuhi berbagai kebutuhan backend. Selain itu, terdapat microservice WAV Converter dan IoT Stethoscope yang berfungsi mendukung aplikasi SWiDS, namun kedua komponen ini berada di luar scope pengerjaan KP.

Pada Scope pengerjaan KP, aplikasi mobile dikembangkan menggunakan Flutter, sebuah framework

pengembangan aplikasi lintas platform yang mendukung Android dan iOS. Flutter dipilih karena kemampuannya yang memungkinkan pembuatan antarmuka pengguna yang interaktif dan responsif, serta dukungan SDK yang luas. Fitur hot-reload pada Flutter juga memungkinkan pengembang untuk melihat perubahan kode secara langsung, sehingga mempercepat proses pengembangan.

Backend aplikasi ini menggunakan Firebase, yang menyediakan berbagai fungsionalitas backend yang terintegrasi dengan Flutter melalui SDK yang tersedia. Beberapa fitur Firebase yang digunakan mencakup Authentication, Firestore Database, Realtime Database, dan Storage. Firebase Authentication digunakan untuk otentikasi pengguna menggunakan email sebagai metode login dan pendaftaran. Ini memberikan keamanan yang terintegrasi dan memudahkan pengguna untuk mengakses aplikasi. Firebase Storage berfungsi untuk menyimpan file yang berkaitan dengan aplikasi, termasuk data rekaman dari stetoskop.

Aplikasi ini menggunakan dua jenis database, yaitu Firestore Database dan Realtime Database. Penggunaan dua database ini dilakukan karena kebutuhan yang berbeda: Realtime Database digunakan untuk streaming data langsung dari stetoskop, memungkinkan data audio diproses dan disimpan secara real-time. Sebaliknya, Firestore Database berfungsi sebagai penyimpanan data utama yang digunakan oleh aplikasi mobile untuk menyimpan informasi pengguna dan data aplikasi lainnya. Kedua database ini adalah tipe NoSQL, yang menggunakan struktur dokumen dan memungkinkan fleksibilitas dalam penyimpanan data yang tidak terstruktur.

Pada bagian di luar scope pengerjaan, terdapat microservice backend dan IoT stetoskop yang berperan dalam pengiriman data medis. Microservice backend

dikembangkan menggunakan Python dan dideploy pada Google Compute Engine (GCE) dengan spesifikasi standar berbasis Debian dan disk persisten 10GB. Microservice ini bertugas untuk mengonversi data byte yang diterima dari Realtime Database menjadi file audio berformat WAV, yang kemudian disimpan di Firebase Storage. Di sisi lain, perangkat IoT stetoskop berfungsi untuk mengirimkan data bit secara real-time, yang kemudian diproses oleh microservice untuk keperluan penyimpanan dan analisis lebih lanjut.

Arsitektur ini memastikan aplikasi SWiDS dapat menangani berbagai fungsionalitas dengan lancar, mulai dari streaming data medis, penyimpanan data, hingga autentikasi pengguna.

4.2. Fungsionalitas Aplikasi SWiDS

Aplikasi SWiDS dirancang untuk membantu tenaga kesehatan dalam merekam dan mengelola data suara stetoskop secara efisien, dengan tujuan mendukung diagnosis medis berbasis suara. Fungsionalitas utamanya memungkinkan pengguna untuk mencatat, menyimpan, dan mengakses rekam jejak pemeriksaan medis, sehingga dapat melacak perkembangan kondisi pasien dengan lebih terstruktur. Setiap pengguna dapat mengelola rekaman stetoskop yang telah dibuat untuk menyimpan arsip pemeriksaan. Aplikasi ini memiliki beberapa fitur utama, yaitu integrasi dengan stetoskop, perekaman suara stetoskop, pengelolaan rekam medis, daftar pasien, serta pengaturan profil pengguna.

Untuk mengakses aplikasi, pengguna wajib login menggunakan akun yang terdaftar. Jika pengguna mengalami masalah dalam mengakses akun karena lupa kata sandi, aplikasi menyediakan fitur pemulihan melalui

opsi "lupa password" yang memudahkan pengguna untuk mengatur ulang akses akun mereka.

Pada fitur integrasi dengan stetoskop, pengguna dapat menambahkan perangkat stetoskop dengan cara memindai QR code yang terdapat pada alat. Setelah terdaftar, stetoskop akan memiliki status yang mencerminkan ketersediaannya: "available" menandakan perangkat siap digunakan, "unavailable" menunjukkan perangkat sedang dipakai oleh pengguna lain, dan "active" berarti stetoskop saat ini aktif digunakan. Pengguna diberi kebebasan untuk menambahkan lebih dari satu stetoskop ke dalam akun mereka, namun hanya satu perangkat yang dapat diaktifkan pada satu waktu.

Untuk merekam suara medis, pengguna harus terlebih dahulu mengaktifkan stetoskop yang ingin digunakan. Aplikasi menyediakan tiga kategori perekaman, yaitu jantung, abdomen, dan paru-paru, untuk memfasilitasi pemeriksaan medis yang spesifik. Setelah memilih kategori, pengguna dapat memilih pasien dari daftar yang sudah ada atau menambahkan pasien baru, serta mencatat keluhan yang dialami pasien sebagai bagian dari dokumentasi. Selama proses perekaman, suara stetoskop akan dikonversi secara otomatis oleh aplikasi dan hasilnya dapat disimpan dalam sistem.

Histori rekam medis yang telah selesai dapat diakses melalui fitur riwayat, yang menyediakan filter berdasarkan jenis rekaman (jantung, abdomen, atau paru-paru). Pengguna dapat menelusuri detail dari setiap rekaman, termasuk mendengarkan kembali suara yang telah direkam dan mengakses informasi tambahan terkait pemeriksaan.

Selain itu, aplikasi juga menyediakan fitur pengelolaan daftar pasien, di mana pengguna dapat menambah atau memperbarui informasi pasien yang terdaftar. Informasi yang dapat disimpan meliputi data

personal pasien dan riwayat pemeriksaan. Pada bagian profil, pengguna dapat mengelola informasi akun mereka, seperti mengubah foto profil dan memperbarui data pribadi yang terkait dengan akun.

[Halaman ini sengaja dikosongkan]

BAB V

IMPLEMENTASI SISTEM

Bab ini membahas implementasi sistem yang telah kami buat, mencakup pengembangan aplikasi SWiDS menggunakan Flutter dan Firebase. Implementasi aplikasi SWiDS terbagi menjadi beberapa fitur utama, yaitu fitur Login dan Forgot Password, integrasi dengan stetoskop, perekaman suara medis, pengelolaan riwayat rekam medis, pengaturan daftar pasien, serta pengaturan profil pengguna.

5.1. Fitur Login dan Forgot Password

Pada fitur ini, digunakan SDK Firebase Authentication yang berfungsi untuk menangani proses login dan pemulihan kata sandi secara aman. Implementasi login akan ditampilkan pada Kode Sumber 5.1 dan Gambar 5.1, sedangkan implementasi fitur Forgot Password akan ditampilkan pada Kode Sumber 5.2 dan Gambar 5.2.

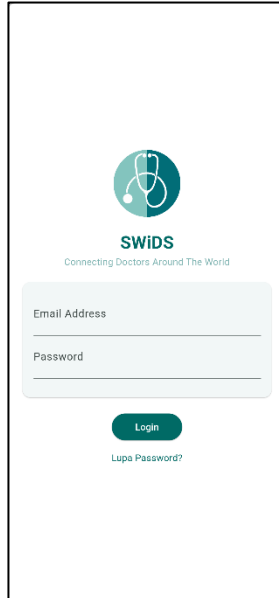
```
void _submit() async {
  final isValid = _form.currentState!.validate();

  if (!isValid) {
    return;
  }
  _form.currentState!.save();

  try {
    await _firebase.signInWithEmailAndPassword(
      email: _enteredEmail, password: _enteredPassword);
  } on FirebaseAuthException catch (error) {
    ScaffoldMessenger.of(context).clearSnackBars();
    ScaffoldMessenger.of(context).showSnackBar(SnackBar(
      content: Text(error.message!),
    ));
  }
}
```

```
}
```

Kode Sumber 5.1 Implementasi Service Login



Gambar 5.1 Tampilan Halaman Login

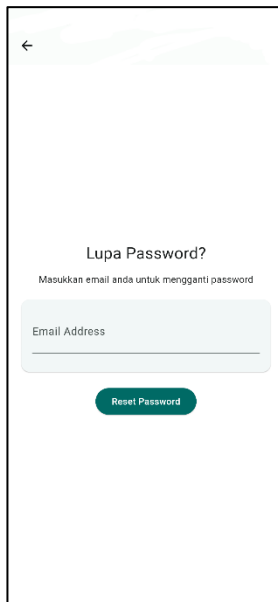
```
void _submit() async {  
  final isValid = _form.currentState!.validate();  
  
  if (!isValid) {  
    return;  
  }  
  
  _form.currentState!.save();  
  
  try {  
    await _firebaseAuth.sendPasswordResetEmail(email:  
      _enteredEmail);  
  }  
}
```

```

Navigator.pop(context);
ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(
    content:
      Text('Link for password reset has been sent to
your email!')),
  );
} catch (e) {
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('Failed to send reset email:
$e')),
  );
}
}

```

Kode Sumber 5.2 Implementasi Service Forgot Password



Gambar 5.2 Tampilan Halaman Forgot Password

5.2. Fitur Integrasi Stetoskop

Fitur ini menggunakan library QR code reader untuk memindai dan mendaftarkan stetoskop ke dalam sistem serta mengelola statusnya. Kode Sumber 5.3 menunjukkan implementasi pemindaian QR stetoskop, dan Gambar 5.3 menampilkan antarmuka untuk pemindaian QR. Setelah pemindaian berhasil, sistem mengaktifkan stetoskop pada Kode Sumber 5.4 dan Gambar 5.4 atau menonaktifkannya pada Kode Sumber 5.5 dan Gambar 5.5

```
QRViewController? controller;

@Override
void reassemble() {
    super.reassemble();
    if (controller != null) {
        controller!.pauseCamera();
        controller!.resumeCamera();
    }
}

void _onQRViewCreated(QRViewController controller) {
    this.controller = controller;
    controller.scannedDataStream.listen((scanData) {
        final decryptedData = decryptData(scanData.code ?? '');
        _processScanData(decryptedData);
    });
}

void _processScanData(String scanData) {
    if (scanData.startsWith('swids-st:')) {
        // If the scan data starts with the expected prefix
        controller?.pauseCamera(); // Pause the camera for dialog
interaction
        String scanResult =
            scanData.substring('swids-st:'.length); // Extract the
```

```

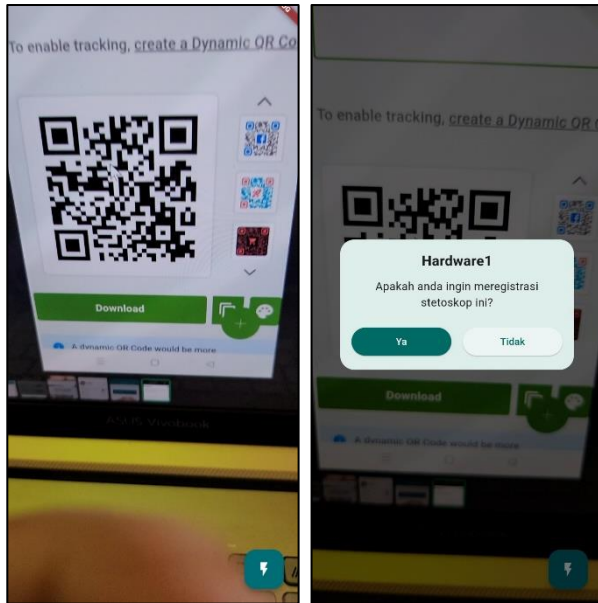
actual data
    _showScanResultDialog(scanResult);
} else {
    // If the scan data does not match the expected format,
    continue scanning
    controller?.resumeCamera();
}
}

void _showScanResultDialog(String scanResult) {
    showDialog(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) {
            return ScanResultDialog(
                scanResult: scanResult,
                onNoPressed: () {
                    _handleDialogButtonPressed();
                },
            );
        },
    );
}

void _handleDialogButtonPressed() {
    Navigator.of(context).pop(); // Close the dialog
    Navigator.of(context).pop(); // Navigate back to the
previous screen
}

```

Kode Sumber 5.3 Implementasi Scan QR Stetoskop



Gambar 5.3 Tampilan Halaman Scan QR

```
final usedStethoscope = ref.read(usedStethoscopeProvider);  
  
try {  
  if (usedStethoscope.stethoscopeId == null) {  
    await ref  
      .read(usedStethoscopeProvider.notifier)  
      .toggleStethoscopeAvailability(stethoscopeId,  
"unavailable");  
  
    ref  
      .read(usedStethoscopeProvider.notifier)  
      .turnOn(stethoscopeId, stethoscopeName);  
  } else if (usedStethoscope.stethoscopeId != stethoscopeId)  
{  
    await ref  
      .read(usedStethoscopeProvider.notifier)
```

```

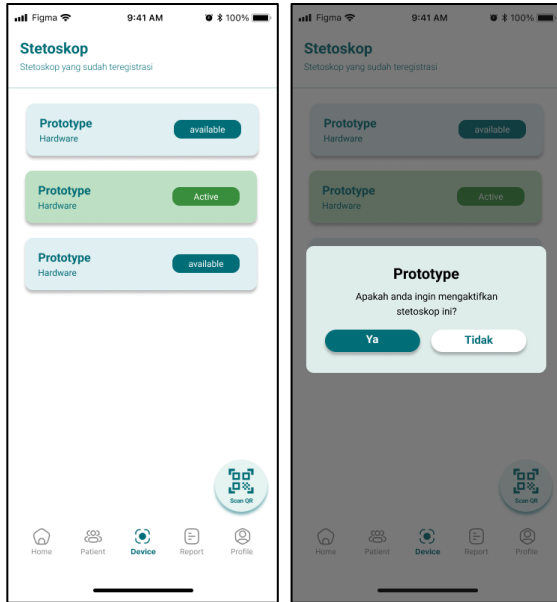
        .toggleStethoscopeAvailability(
            usedStethoscope.stethoscopeId!, "available");

    await ref
        .read(usedStethoscopeProvider.notifier)
        .toggleStethoscopeAvailability(stethoscopeId,
"unavailable");

    ref
        .read(usedStethoscopeProvider.notifier)
        .turnOn(stethoscopeId, stethoscopeName);
    }
} catch (e) {
    showDialog(
        context: context,
        builder: (context) => AlertDialog(
            title: const Text('Error'),
            content: Text('An error occurred: $e'),
            actions: [
                TextButton(
                    onPressed: () => Navigator.of(context).pop(),
                    child: const Text('OK'),
                ),
            ],
        ),
    );
} finally {
    Navigator.of(context).pop(); // Pop the loading modal
    Navigator.of(context).pop(); // Close the main dialog
}

```

Kode Sumber 5.4 Implementasi Service Mengaktifkan Stetoskop



Gambar 5.4 Tampilan Halaman Mengaktifkan Stetoskop

```

try {
  await ref
    .read(usedStethoscopeProvider.notifier)
    .toggleStethoscopeAvailability(stethoscopeId,
  "available");

  ref.read(usedStethoscopeProvider.notifier).turnOff();
} catch (e) {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text('Error'),
      content: Text('An error occurred: $e'),
      actions: [
        TextButton(
          onPressed: () => Navigator.of(context).pop(),

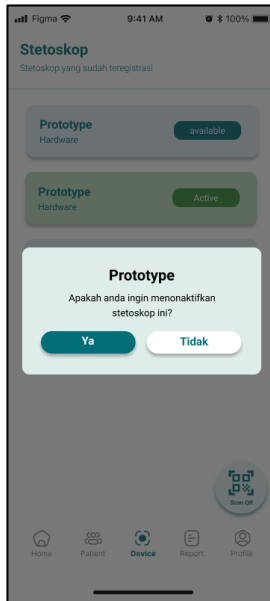
```

```

        child: const Text('OK'),
      ),
    ],
  ),
);
} finally {
  Navigator.of(context).pop(); // Pop the loading modal
  Navigator.of(context).pop(); // Close the main dialog
}

```

Kode Sumber 5.5 Implementasi Service Menonaktifkan Stetoskop



Gambar 5.5 Tampilan Halaman Menonaktifkan Stetoskop

5.3. Fitur Rekam Medis

Fitur ini memungkinkan pengguna merekam suara stetoskop berdasarkan jenis pemeriksaan (jantung, paru-paru, atau abdomen), serta menyimpan data di Firestore

dan Firebase Storage. Proses meliputi penambahan informasi pasien, keluhan, dan penyimpanan rekaman. Implementasi penyimpanan data pasien ditunjukkan pada Kode Sumber 5.6 dan Gambar 5.6, yang menampilkan halaman memilih pasien. Kode Sumber 5.7 dan Gambar 5.7 menunjukkan cara menyimpan keluhan, sementara Kode Sumber 5.8 dan Gambar 5.8 memperlihatkan proses mendengar dan menyimpan rekam medis.

```
async {
    // Update the patientId in the progress
    report when "Selanjutnya" is pressed
    final NetworkService _networkService =
    NetworkService();
    bool isConnected = await _networkService
    .checkInternetAndShowSnackBar(context)
;

    if (!isConnected) {
        return;
    }
    final progressReport =
    ref.read(progressReportProvider);
    if (progressReport != null) {
        try {
            await FirebaseFirestore.instance
            .collection('reports')
            .doc(progressReport.id)
            .update({'patientId':
            currentPatient.id});

            _navigateToMedicalCheckup();
        } catch (e) {

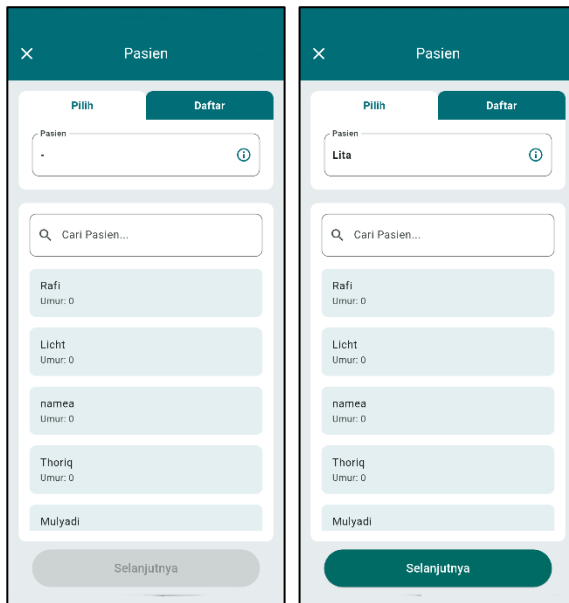
            ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(
            content: Text('Failed to update
```

```

patientId.'),
    },
    );
    }
    }
    }

```

Kode Sumber 5.6 Implementasi Save Patient pada Rekam Medis



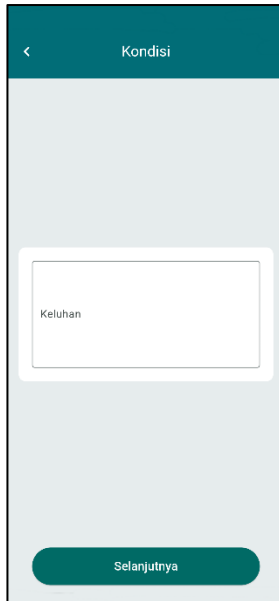
Gambar 5.6 Tampilan Halaman Memilih Pasien saat Rekam Medis

```

void _submitComplaint() async {
    bool isConnected =
        await
        _networkService.checkInternetAndShowSnackBar(context);

    if (!isConnected) {

```

Gambar 5.7 Tampilan Halaman Memasukan Keluhan saat Rekam Medis

```
Future<void> _saveReport(BuildContext context) async {
  try {
    // Reference to the Firestore document
    DocumentReference reportRef =
    FirebaseFirestore.instance.collection('reports').doc(reportId);

    // Update the report
    await reportRef.update({
      'status': 'finished',
    });

    // Show a success message
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Rekam medis berhasil
    disimpan!')),
```

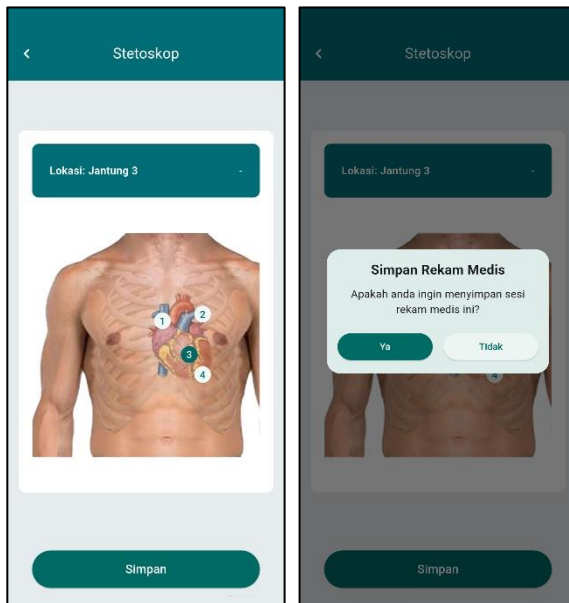


```

    );
  } catch (e) {
    // Show an error message in case of failure
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Gagal menyimpan rekam medis:
    $e')),
    );
  }
}
}
}

```

Kode Sumber 5.8 Implementasi Menyimpan Rekam Medis



Gambar 5.8 Tampilan Halaman Mendengar dan Menyimpan Rekam Medis

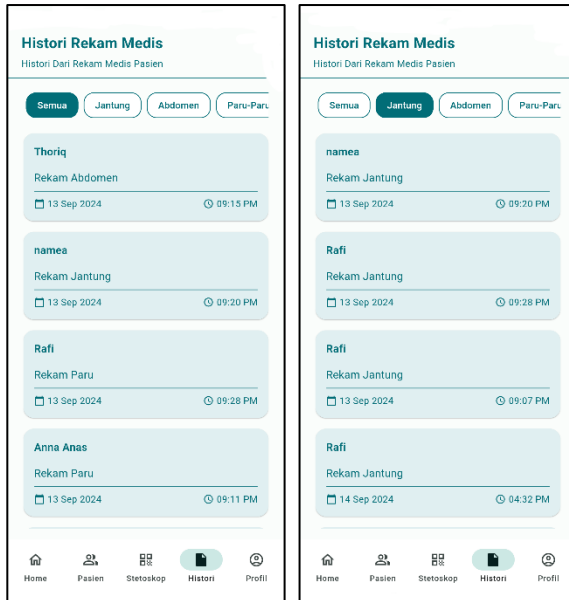
5.4. Fitur Histori Rekam Medis

Histori rekam medis memungkinkan pengguna untuk melihat rekam medis yang tersimpan secara kronologis, difilter berdasarkan kategori pemeriksaan.

Pengguna dapat melihat detail informasi, termasuk rekaman suara medis. Implementasi daftar histori rekam medis akan ditampilkan pada Kode Sumber 5.10 dan Gambar 5.9. Implementasi halaman detail rekam medis akan ditampilkan pada Kode Sumber 5.11 dan Gambar 5.10.

```
Scaffold(  
  appBar: const PreferredSize(  
    preferredSize: Size.fromHeight(80.0),  
    child: HistoryHeader(),  
  ),  
  body: Padding(  
    padding: const EdgeInsets.symmetric(horizontal: 23.0,  
vertical: 16),  
    child: Column(  
      crossAxisAlignment: CrossAxisAlignment.start,  
      children: [  
        SingleChildScrollView(  
          scrollDirection: Axis.horizontal,  
          child: Row(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: _buildChoiceChips(),  
          ),  
        ),  
        const SizedBox(height: 10),  
        Expanded(  
          child: Center(  
            child: _buildContent(reports, patientNameMap),  
          ),  
        ),  
      ],  
    ),  
  ),  
);  
}
```

Kode Sumber 5.9 Implementasi List Histori Rekam Medis



Gambar 5.9 Tampilan Halaman Histori Rekam Medis

```

 Scaffold(
    appBar: AppBar(
      toolbarHeight: 80,
      flexibleSpace: Container(
        color: const Color(0xFF016D77),
      ),
      leading: IconButton(
        icon: const Icon(Icons.chevron_left_rounded),
        onPressed: () => Navigator.pop(context),
        color: Colors.white,
        iconSize: 30,
      ),
      title: const Text(
        "Rekam Medis",
        style: TextStyle(
          color: Colors.white,

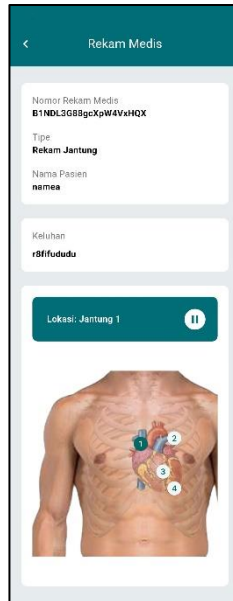
```

```

        fontSize: 22,
        fontWeight: FontWeight.w400,
    ),
),
centerTitle: true,
),
body: SingleChildScrollView(
  child: Container(
    width: double.infinity,
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 30,
    ),
    color: const Color(0xFFE6ECED),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        DetailReportDataBox(widget.report),
        const SizedBox(height: 25),
        DetailComplaintBox(widget.report.complaint),
        const SizedBox(height: 25),
        DetailRecordBox(
          recordPath: widget.report.recordPath,
          type: widget.report.type,
        ),
      ],
    ),
  ),
),
);

```

Kode Sumber 5.10 Implementasi Detail Rekam Medis



Gambar 5.10 Tampilan Halaman Detail Rekam Medis

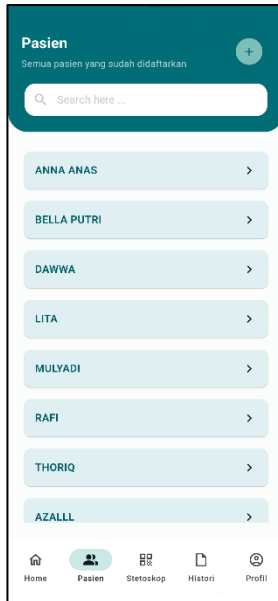
5.5. Fitur Daftar Pasien

Fitur ini menyediakan manajemen daftar pasien, di mana pengguna dapat menambahkan dan melihat informasi pasien. Implementasi melibatkan interaksi dengan SDK Firestore untuk menyimpan data pasien baru. Implementasi daftar pasien akan ditampilkan pada Kode Sumber 5.12 dan Gambar 5.11, yang memperlihatkan halaman daftar pasien dan fitur pencarian. Implementasi halaman detail pasien akan ditampilkan pada Kode Sumber 5.13 dan Gambar 5.12, yang mencakup informasi pasien dan daftar laporan medis terkait. Terakhir, implementasi layanan penambahan pasien baru akan ditampilkan pada Kode Sumber 5.14 dan Gambar 5.13,

yang menampilkan form pendaftaran pasien baru dengan validasi dan penyimpanan ke Firestore.

```
Scaffold(  
  appBar: PreferredSize(  
    preferredSize: const Size.fromHeight(160.0),  
    child: PatientHeader(  
      onTap: goToAddPatientPage,  
      patientNameController: searchController,  
    ),  
  ),  
  body: Padding(  
    padding: const EdgeInsets.symmetric(  
      horizontal: 20,  
      vertical: 25,  
    ),  
    child: ListView.separated(  
      itemCount: filteredPatients.length,  
      itemBuilder: (context, index) => InkWell(  
        onTap: () => goToDetailPatient(index),  
        child: PatientCard(  
          name: filteredPatients[index].name,  
          id: filteredPatients[index].id,  
        ),  
      ),  
      separatorBuilder: (context, index) => const  
    ),  
  ),  
  ),  
  ),  
  );
```

Kode Sumber 5.11 Implementasi List Pasien



Gambar 5.11 Tampilan Halaman Daftar Pasien

```

Scaffold(
  appBar: AppBar(
    toolbarHeight: 80,
    flexibleSpace: Container(
      color: const Color(0xFF016D77),
    ),
    leading: IconButton(
      icon: const Icon(Icons.chevron_left_rounded),
      onPressed: () => Navigator.pop(context),
      color: Colors.white,
      iconSize: 30,
    ),
    title: const Text(
      "Pasien",
      style: TextStyle(
        color: Colors.white,

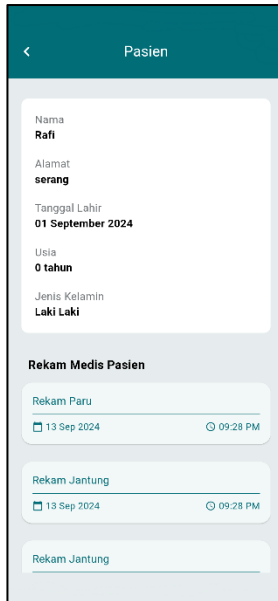
```

```

        fontSize: 22,
        fontWeight: FontWeight.w400,
    ),
),
centerTitle: true,
),
body: Container(
    height: double.infinity,
    width: double.infinity,
    padding: const EdgeInsets.symmetric(
        horizontal: 20,
        vertical: 30,
    ),
    color: const Color(0xFFE6E6ED),
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            DetailPatientBox(widget.patient),
            Expanded(
                child: PatientReportsList(
                    patientId: widget.patient.id,
                ),
            ),
        ],
    ),
),
);

```

Kode Sumber 5.12 Implementasi Detail Pasien



Gambar 5.12 Tampilan Halaman Detail Pasien

```
submitForm() async {
  if (_key.currentState?.validate() ?? false) {
    bool isConnected =
      await
_networkService.checkInternetAndShowSnackbar(context);

    if (!isConnected) {
      // If there's no internet connection, return early
      return;
    }

    // Proceed with Firebase operation if connected to the
internet
    final userId = FirebaseAuth.instance.currentUser?.uid;
    if (userId != null) {
      final birthDate = DateTime.parse(_birthController.text);
```

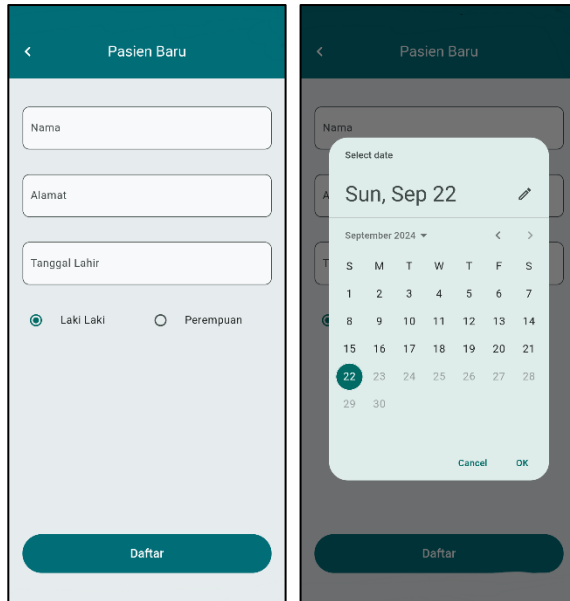
```

        final newPatient = Patient(
            id: '',
            name: _nameController.text,
            userId: userId,
            birth: Timestamp.fromDate(birthDate),
            address: _addressController.text,
            gender: _gender!,
        );

        try {
            await
            ref.read(patientProvider.notifier).addPatient(newPatient);
            Navigator.pop(context);
        } catch (e) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(content: Text('An error occurred:
                ${e.toString()}')),
            );
        }
    }
}

```

Kode Sumber 5.13 Implementasi Service Menambah Pasien



Gambar 5.13 Tampilan Halaman Mendaftarkan Pasien Baru

5.6. Fitur Profil

Pada fitur profil, pengguna dapat memperbaiki informasi akun seperti nama, nomor telepon, dan foto profil. Firebase Storage digunakan untuk menyimpan foto profil yang diunggah, sedangkan data pengguna disimpan dan diperbarui di Firestore. Implementasi pembaruan profil pengguna akan ditampilkan pada Kode Sumber 5.15 dan Gambar 5.14, yang mencakup halaman profil dan fitur edit profil dengan proses unggah foto dan pembaruan data di Firestore.

```
Future<void> _updateUserProfile() async {
```

```

final user = FirebaseAuth.instance.currentUser;
if (user != null) {
  String? imageUrl;

  if (_selectedImage != null) {
    setState(() {
      _isUploading = true;
    });

    final storageRef = FirebaseStorage.instance
      .ref()
      .child('profile_pictures')
      .child('${user.uid}.jpg');

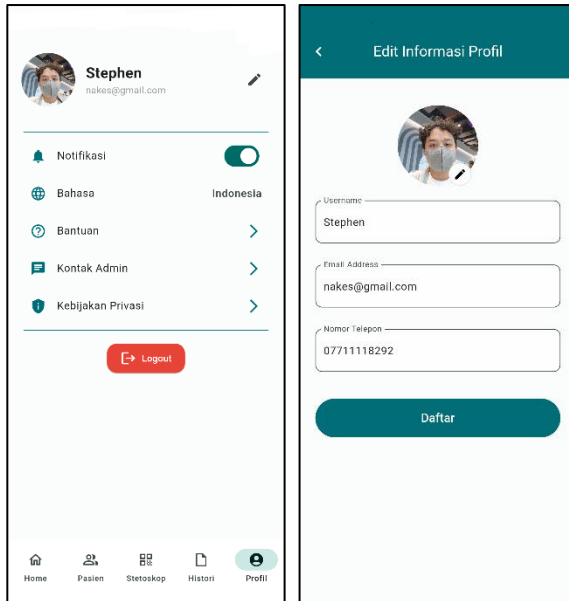
    final uploadTask = storageRef.putFile(_selectedImage!);
    final snapshot = await uploadTask.whenComplete(() {});
    imageUrl = await snapshot.ref.getDownloadURL();

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

  final userDoc =
  FirebaseFirestore.instance.collection('users').doc(user.uid);
  await userDoc.update({
    'name': nameController.text,
    // 'email': emailController.text,
    'phoneNumber': phoneController.text,
    if (imageUrl != null) 'profilePicture': imageUrl,
  });
}
}

```

Kode Sumber 5.14 Implementasi Service Update Profil



Gambar 5.14 Tampilan Halaman Profil dan Edit Profil

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap Aplikasi SWiDS. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap Aplikasi SWiDS guna menguji kemampuan arsitektur dalam melayani permintaan sistem aplikasi.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

- a. Kemampuan aplikasi untuk melayani tampilan UI.
- b. Kemampuan aplikasi untuk melakukan data fetching dari database ke aplikasi.
- c. Kemampuan aplikasi untuk menyimpan data yang diinput melalui aplikasi ke database.
- d. Kemampuan aplikasi untuk melayani upload file dari aplikasi ke storage.
- e. Kemampuan aplikasi untuk melakukan melakukan proses rekam data dengan integrasi stetoskop.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Membuka Aplikasi SWiDS

- User dapat membuka aplikasi SWiDS di perangkat mobile.
 - Aplikasi dapat menampilkan halaman splash dan berlanjut ke halaman login.
2. Login ke Aplikasi SWiDS
 - User dapat memasukkan email dan kata sandi untuk login.
 - Aplikasi dapat memverifikasi kredensial dan mengarahkan pengguna ke halaman utama.
 - User dapat menggunakan Forgot Password jika lupa kata sandi.
 3. Integrasi dengan Stetoskop
 - User dapat memindai QR code pada stetoskop untuk mendaftarkan perangkat.
 - User dapat mengaktifkan atau menonaktifkan stetoskop sesuai kebutuhan.
 4. Perekaman Medis dengan Stetoskop
 - User dapat memilih jenis pemeriksaan (jantung, paru-paru, atau abdomen).
 - User dapat melakukan pengisian data rekam medis
 - Nakes dapat mendengar suara stetoskop dan menyimpan hasil rekam medis di sistem.
 5. Manajemen Daftar Pasien
 - User dapat memasukkan informasi pasien baru dan menyimpan datanya.
 - Aplikasi memverifikasi kelengkapan data sebelum menyimpan ke database.

- Aplikasi dapat menampilkan daftar pasien yang berhasil ditambahkan.
 - User dapat melihat informasi detail dari pasien.
6. Melihat Riwayat Rekam Medis
 - User dapat melihat daftar rekam medis berdasarkan kategori pemeriksaan.
 - User dapat membuka detail rekam medis dan memutar kembali rekaman suara.
 7. Mengelola Profil User
 - User dapat memperbarui informasi profil seperti nama, foto, dan nomor telepon.

Pengujian lain dilakukan bersama para stakeholder aplikasi dalam rapat bulanan untuk mengevaluasi implementasi fitur dan kemajuan pengembangan aplikasi SWiDS.

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku aplikasi SWiDS terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 6.1 Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Aplikasi dapat melayani tampilan UI	Terpenuhi
Aplikasi dapat melakukan data	Terpenuhi

fetching dari database	
Aplikasi dapat menyimpan data input pada database	Terpenuhi
Aplikasi dapat menyimpan file pada storage	Terpenuhi
Aplikasi dapat melakukan integrasi dengan stetoskop untuk proses rekam medis	Terpenuhi

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pembangunan aplikasi SWiDS adalah sebagai berikut :

- a. Aplikasi SWiDS dengan fitur utama aplikasi, termasuk integrasi dengan stetoskop, perekaman suara medis, manajemen daftar pasien, dan riwayat rekam medis, telah berhasil dikembangkan sesuai dengan permintaan stakeholder.
- b. Aplikasi SWiDS telah memudahkan tenaga kesehatan dalam mengelola data rekam medis, mendukung pemeriksaan berbasis suara, dan menyimpan catatan medis pasien secara terstruktur, sehingga meningkatkan efisiensi layanan kesehatan.

7.2. Saran

Saran untuk pengembangan aplikasi SWiDS adalah sebagai berikut :

- a. Optimalisasi antarmuka pengguna (UI) untuk memberikan pengalaman yang lebih baik dan intuitif
- b. Menambahkan enkripsi pada data rekam medis yang disimpan di Firebase untuk meningkatkan keamanan dan melindungi privasi data pasien.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Butcher, C. J., & Hussain, W. (2022). Digital healthcare: The future. *Future Healthcare Journal*, 9(2), 113–117. <https://doi.org/10.7861/fhj.2022-0046>
- Compute Engine documentation | Compute Engine Documentation*. (n.d.). Google Cloud. Retrieved October 28, 2024, from <https://cloud.google.com/compute/docs>
- Dart programming language | Dart*. (n.d.). Retrieved October 27, 2024, from <https://dart.dev/>
- Dokumentasi Firebase*. (n.d.). Firebase. Retrieved October 27, 2024, from <https://firebase.google.com/docs?hl=id>
- Figma: The Collaborative Interface Design Tool*. (n.d.). Figma. Retrieved October 27, 2024, from <https://www.figma.com/>

Flutter—Build apps for any screen. (n.d.). Retrieved

October 27, 2024, from <https://flutter.dev/>

Gagnon, M.-P., Ngangue, P., Payne-Gagnon, J., &

Desmartis, M. (2016). m-Health adoption by

healthcare professionals: A systematic review.

Journal of the American Medical Informatics

Association: JAMIA, 23(1), 212–220.

<https://doi.org/10.1093/jamia/ocv052>

Mobile accessibility—Learn web development | MDN.

(2024, August 2). [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Mobile)

[US/docs/Learn/Accessibility/Mobile](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Mobile)

Sharma, M. (2023, July 31). Microservices in Python:

Django, RabbitMQ and Pika. *Medium.*

[https://medium.com/@mansha99/microservices-in-](https://medium.com/@mansha99/microservices-in-python-django-rabbitmq-and-pika-fe1adb0c6a1a)

[python-django-rabbitmq-and-pika-fe1adb0c6a1a](https://medium.com/@mansha99/microservices-in-python-django-rabbitmq-and-pika-fe1adb0c6a1a)

Shi, W. Y., Mays, J., & Chiao, J.-C. (2015). A wireless

stethoscope. *2015 IEEE MTT-S 2015 International*

Microwave Workshop Series on RF and Wireless

*Technologies for Biomedical and Healthcare
Applications (IMWS-BIO)*, 197–198.

<https://doi.org/10.1109/IMWS-BIO.2015.7303845>

Vertabelo. (n.d.). Vertabelo Data Modeler. Retrieved

October 27, 2024, from <https://vertabelo.com/>

Welcome to Python.org. (n.d.). Python.Org. Retrieved

October 27, 2024, from <https://www.python.org/doc/>

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Afiq Fawwaz Haidar
Tempat, Tanggal Lahir : Cilegon, 3 November 2002
Jenis Kelamin : Laki-laki
Telepon : +6282298239360
Email : afiq.haidar12@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)

BIODATA PENULIS II

Nama : Syarifah Talitha Erfany
Tempat, Tanggal Lahir : Jakarta, 31 Desember 2002
Jenis Kelamin : Perempuan
Telepon : +6281808242937
Email : syarifah.talithae@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)