

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Transformasi website klinik gigi dari arsitektur monolitik ke arsitektur microservice dapat mengatasi berbagai masalah seperti skalabilitas terbatas, pemeliharaan sulit, dan risiko downtime tinggi. Dalam sistem monolitik, seluruh komponen terikat dalam satu kesatuan, membuat perubahan kecil sekalipun berdampak luas dan memerlukan deploy ulang seluruh aplikasi. Hal ini sering menyebabkan downtime dan menghambat perkembangan fitur baru karena kompleksitas kode yang terus meningkat.

Dengan memisahkan frontend dan backend, sistem dibagi menjadi dua bagian utama: tampilan dan pemrosesan data. Frontend bertanggung jawab atas antarmuka pengguna dan interaksi, sedangkan backend menangani logika bisnis dan pengelolaan data. Setiap fungsi backend seperti penjadwalan janji, manajemen pasien, dan penagihan dipecah menjadi layanan yang lebih kecil serta independen. Setiap layanan saling berinteraksi melalui API yang telah dirancang secara jelas, sehingga masing-masing komponen berkembang, diuji, dan diimplementasikan secara sendiri tanpa mengganggu layanan lainnya.

3.1.1 Identifikasi Masalah

Penelitian ini mengidentifikasi sejumlah permasalahan yang muncul selama ini, di antaranya adalah:

Tabel 3.1 Identifikasi Masalah

Masalah	Faktor	Titik Keputusan
Proses pendaftaran pasien dilakukan menggunakan kartu.	Terjadi rusak dan hilangnya kartu kontrol.	Proses pendaftaran dan pengolahan data pasien.
Setelah terdaftar, kartu itu berfungsi menjadi identitas	pasien karena kartu dari kertas.	

Masalah	Faktor	Titik Keputusan
wajib pasien yang harus dibawa setiap kontrol.		
Pencatatan rekam medis dilakukan melalui buku tulis.	Penumpukan berkas menyebabkan proses pencarian data menjadi lambat dan kurang efisien.	Proses data rekam medis pasien.
Penjadwalan pemeriksaan pasien dilakukan terlebih dahulu, kemudian didokumentasikan secara manual dalam buku.	Jadwal pemeriksaan pasien kerap mengalami konflik atau tumpang tindih antar pasien.	Proses penjadwalan pemeriksaan.
Transaksi pembayaran belum tercatat dalam sistem pembukuan.	Proses perhitungan pendapatan masih rentan terhadap kesalahan.	Proses pembayaran.
Proses perhitungan profit dilakukan secara manual dan belum didukung oleh sistem pencatatan data.	Perhitungan pembagian profit antara owner, alat, dan dokter masih kurang terperinci.	Proses perhitungan profit
Sistem monolitik sulit diskalakan karena semua komponen terikat erat satu sama lain. Hal ini menyebabkan kesulitan dalam	Dalam arsitektur monolitik, semua komponen saling bergantung satu sama lain. Perubahan kecil pada satu bagian dapat mempengaruhi	Mengubah arsitektur menjadi microservices memungkinkan setiap layanan dibuat, diuji, dan dideploy secara independen. Ini

Masalah	Faktor	Titik Keputusan
menambah kapasitas atau mengoptimalkan performa untuk bagian tertentu dari aplikasi tanpa mempengaruhi keseluruhan sistem.	bagian lain, yang memerlukan pengujian dan deployment ulang seluruh aplikasi.	dapat meningkatkan fleksibilitas dan efisiensi pengembangan.
Pengembangan dan pengujian fitur baru memerlukan pengujian ulang seluruh aplikasi, yang memakan waktu dan sumber daya. Ini menghambat inovasi dan respons cepat terhadap kebutuhan bisnis yang berubah.	Monolitik tidak memungkinkan skalabilitas selektif. Artinya, jika satu komponen membutuhkan lebih banyak sumber daya, seluruh aplikasi harus diskalakan, yang tidak efisien.	Dengan microservices, setiap layanan dapat diskalakan secara independen sesuai kebutuhan, yang lebih efisien dan hemat biaya .
Ketika ada bug atau masalah dalam satu bagian aplikasi, seluruh sistem bisa terpengaruh. Ini membuat pemeliharaan dan perbaikan menjadi lebih kompleks dan memakan waktu .	Siklus pengembangan dan deployment menjadi lambat karena setiap perubahan memerlukan pengujian dan deployment ulang seluruh aplikasi.	Masalah dalam satu layanan tidak akan mempengaruhi layanan lain, sehingga meningkatkan stabilitas dan keandalan sistem secara keseluruhan.

3.1.2 Pemecahan Masalah

Pemecahan masalah diperlukan untuk merancang langkah-langkah secara sistematis yang akan digunakan dalam menyelesaikan

suatu permasalahan. Dengan adanya kerangka kerja pemecahan masalah ini, diharapkan proses yang dijalankan dan hasil yang diperoleh dapat selaras dengan tujuan yang telah ditetapkan. Adapun langkah dalam penyelesaian masalah sebagai berikut:

Tabel 3.2 Penyelesaian masalah

Fitur	Solusi
Proses registrasi pasien serta pengelolaan data pasien secara terstruktur.	Sistem menyediakan proses pencatatan data pasien secara dinamis untuk memudahkan manajemen pasien.
Proses pengolahan dan manajemen data rekam medis pasien.	Sistem secara dinamis mengelola data rekam medis pasien dengan pengelompokan berdasarkan nama pasien, sehingga memudahkan pemantauan riwayat tindakan medis yang telah dilakukan.
Prosedur pengaturan jadwal pemeriksaan pasien.	Sistem menyediakan penjadwalan pemeriksaan yang dinamis untuk mempermudah pasien dan admin dalam mengelola pemeriksaan.
Proses Pembayaran.	Sistem menyediakan pembayaran yang variatif, sehingga pasien dapat dengan mudah mengakses riwayat transaksinya dan administrasi dapat memantau pendapatan secara efisien.
Proses perhitungan profit	Proses perhitungan profit dirancang secara dinamis dalam sistem, sehingga memudahkan

	pihak administrasi dalam melakukan sharing profit, serta mempermudah dokter dalam memantau pendapatan mereka.
Proses pengelolaan stok alat dan bahan medis	Mengimplementasikan microservice manajemen inventaris yang terintegrasi untuk mempermudah pemantauan stok alat dan bahan medis, mengurangi risiko kehabisan stok, dan memastikan ketersediaan alat dan bahan yang diperlukan untuk tindakan medis
Proses pembukuan	Mengintegrasikan microservice pembayaran elektronik dan otomatisasi pembukuan untuk mengurangi kesalahan dalam pencatatan pembayaran dan mempermudah proses audit keuangan, serta memberikan berbagai opsi pembayaran kepada pasien .

3.2 Perancangan

3.2.1 Perancangan Sistem

Hasil dari analisa kebutuhan sistem diperoleh 3 pengguna sistem sebagai berikut :

Tabel 3.3 Penggunaan Aplikasi

Admin	Admin diberi hak akses menyeluruh ke seluruh informasi sistem, meliputi profile dokter, pasien, jadwal, riwayat medis, pembayaran, dan profit.
Dokter	Fitur yang tersedia untuk dokter mencakup akses ke

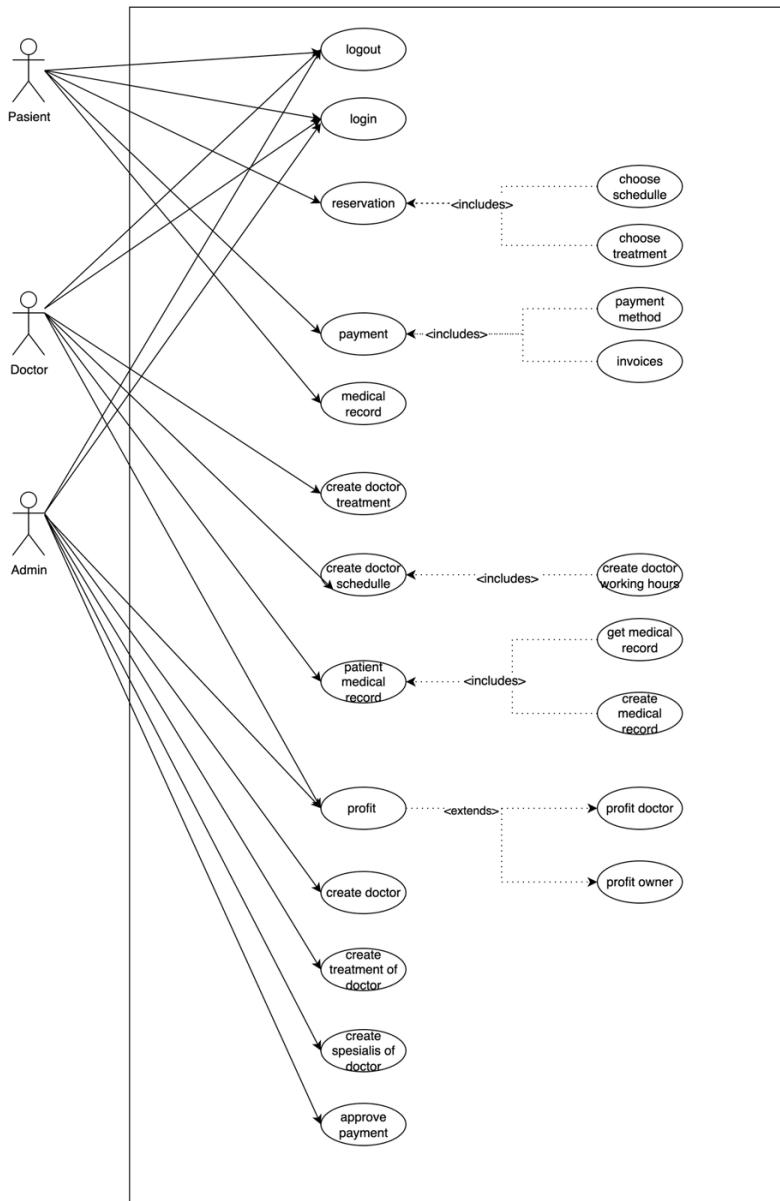
rekam medis pasien yang mereka tangani, pengelolaan jadwal pemeriksaan, serta informasi mengenai penghasilan tiap bulan.

Pasien	Hak akses pasien dibatasi pada fungsi pendaftaran, informasi jadwal dokter, data rekam medis milik sendiri, penjadwalan pemeriksaan, dan riwayat transaksi pembayaran.
--------	--

3.2.2 Perancangan Proses

3.2.2.1 Skema Ucase

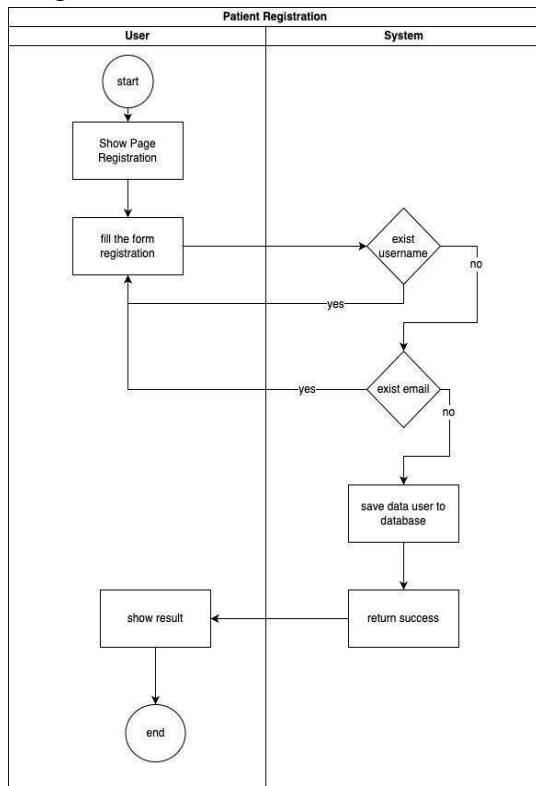
1. Ucase Diagram



Gambar 3.1 Ucase Actor

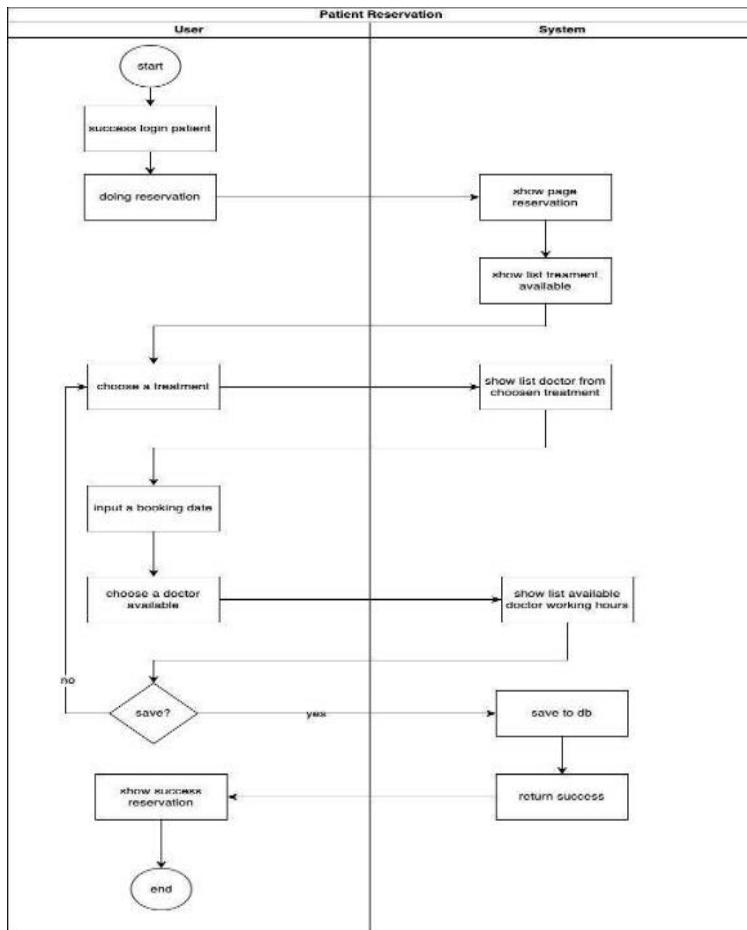
3.2.2.2 Activity

1. Patient Registration



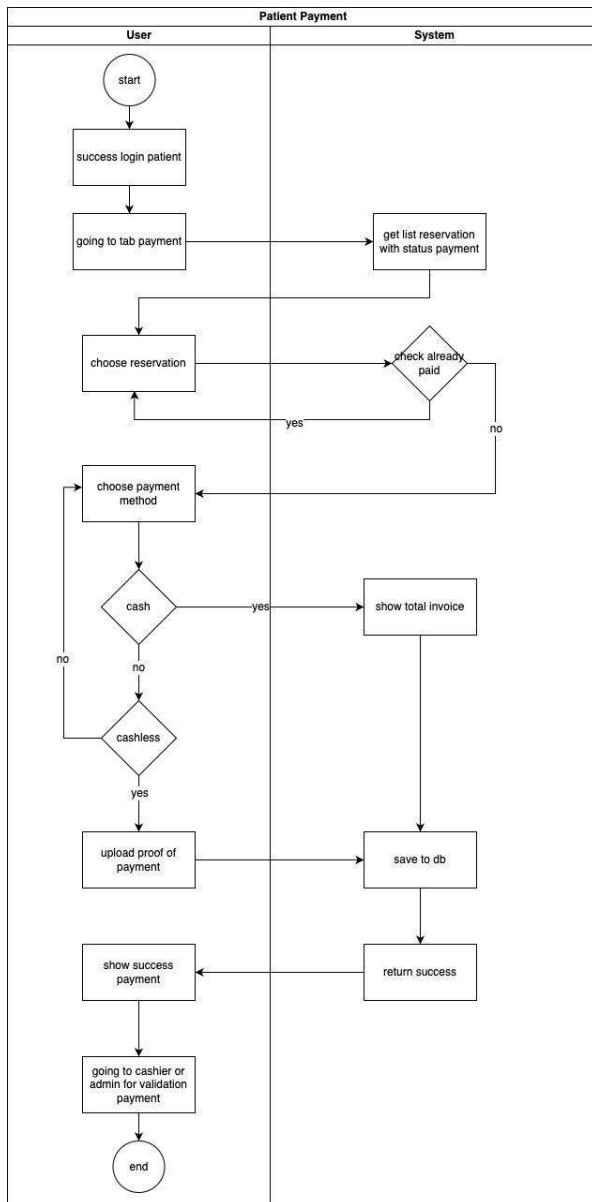
Gambar 3.2 Activity Patient Registration

2. Patient Reservation



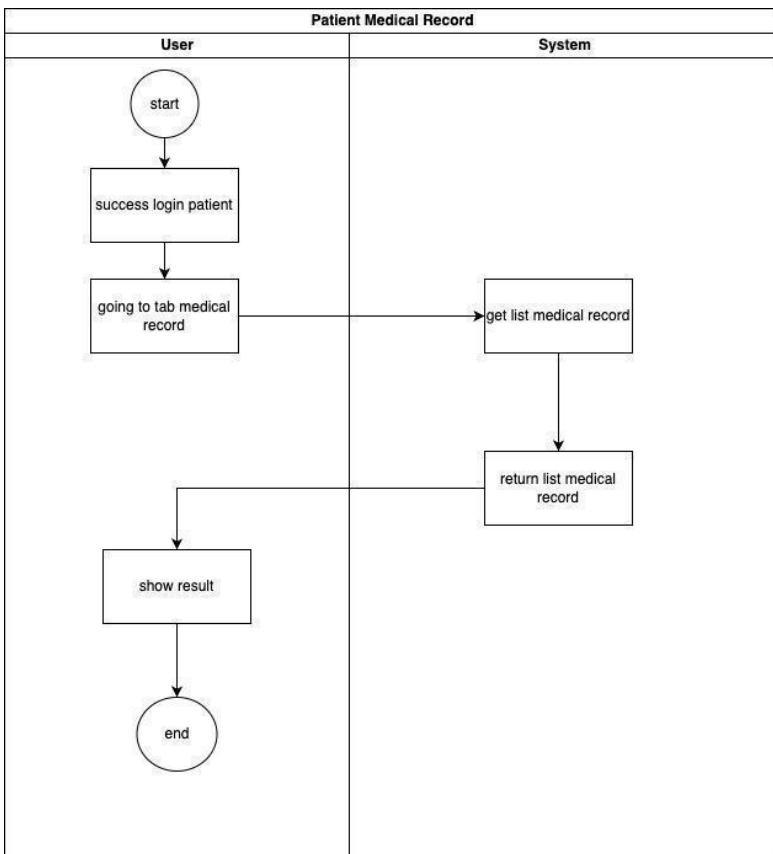
Gambar 3.3 Activity Patient Reservation

3. Patient Payment



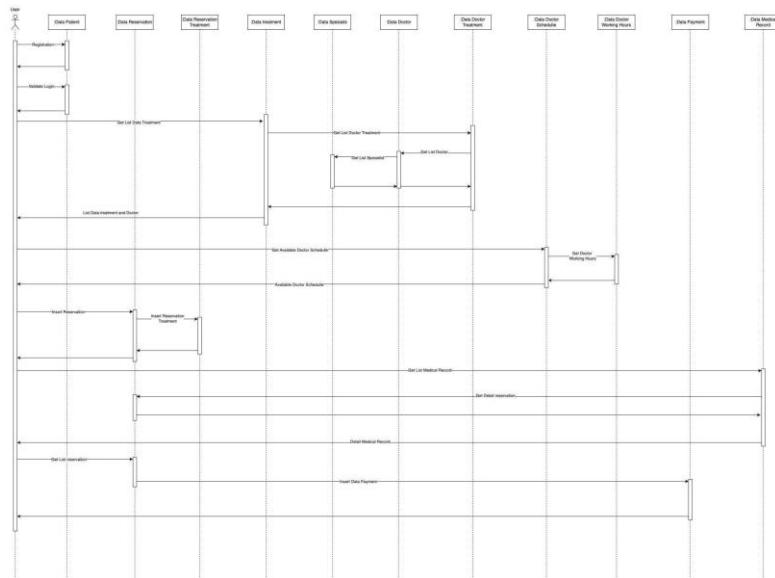
Gambar 3.4 Activity Patient Payment

4. Patient Medical Recor



Gambar 3.5 Activity Patient Medical Record

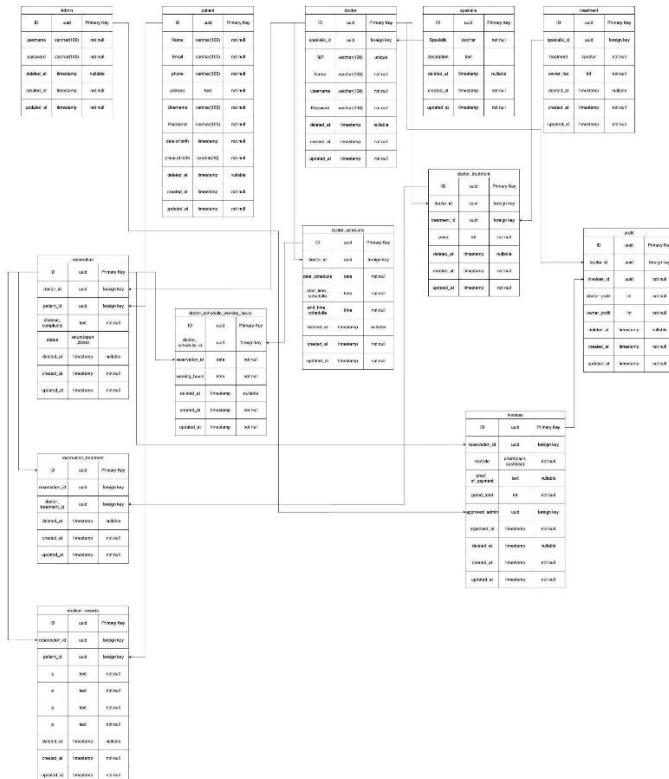
3.2.2.3 Sequence



Gambar 3.6 Sequence

3.2.3 Perancangan Data

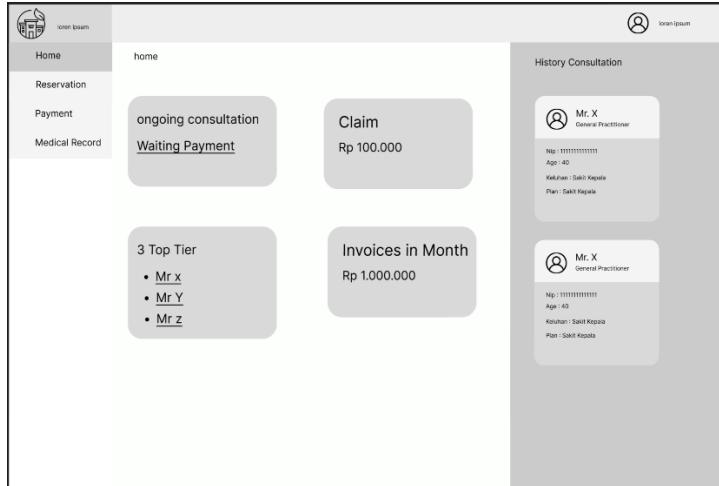
3.2.3.1 Skema Database



Gambar 3.7 Skema Database

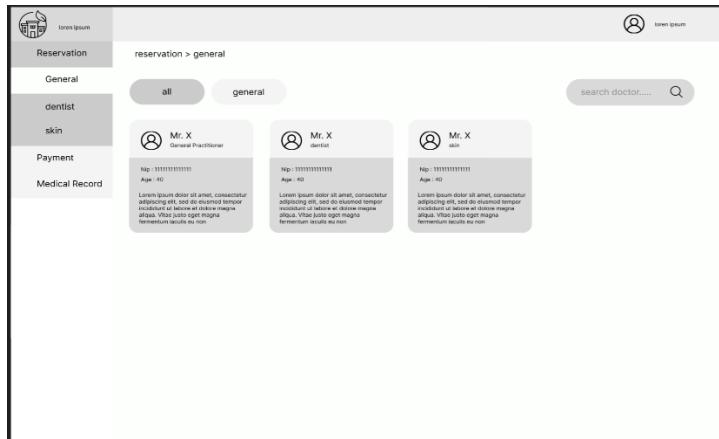
3.2.4 Perancangan User Interface

1. Home Dashboard



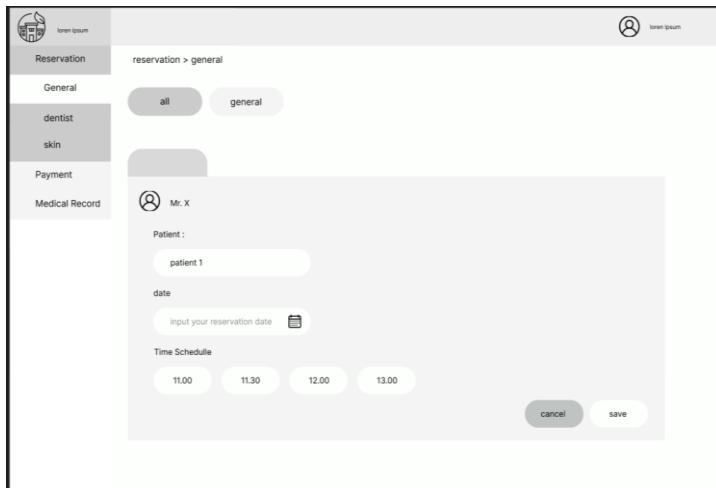
Gambar 3.8 User Interface Home Dashboard

2. Pilih dokter



Gambar 3.9 User Interface Pilih Dokter

3. Reservasi



Gambar 3.10 User Interface Reservasi

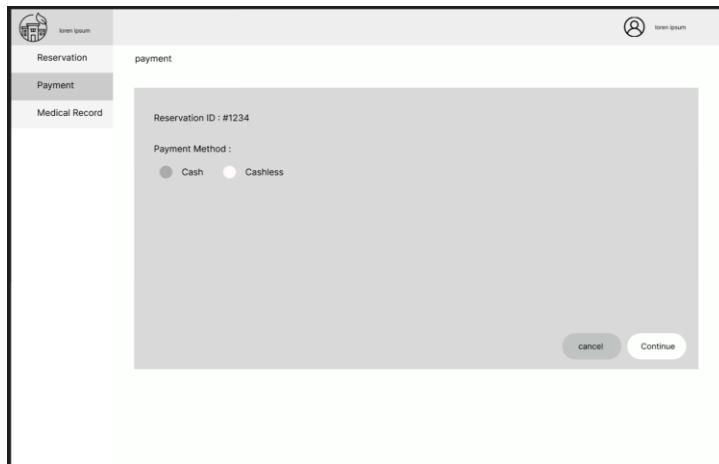
4. List Pembayaran

The screenshot shows a payment list interface. On the left is a sidebar with icons for Reservation, Payment, and Medical Record. The 'Payment' icon is highlighted. The main area has a header 'payment' and a table with columns: No, ReservationID, Date, and Status. The table data is as follows:

No	ReservationID	Date	Status
1	#1234	22 Mei 2014	Waiting
2	#2222	23 Mei 2014	Processing
2	#3333	24 Mei 2014	Completed

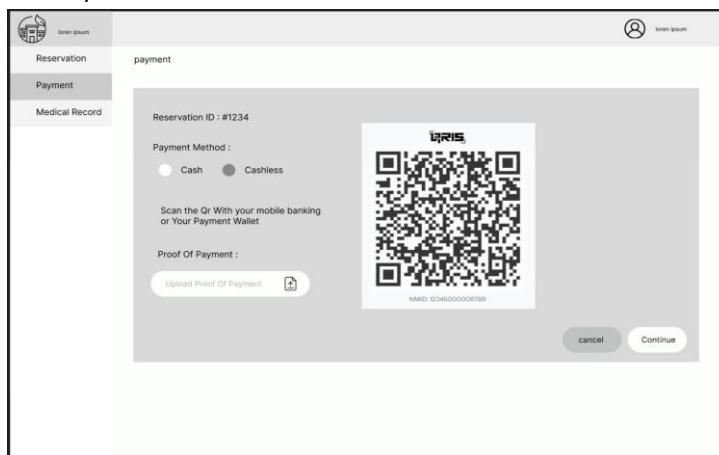
Gambar 3.11 User Interface List Pembayaran

5. Pembayaran Secara Tunai



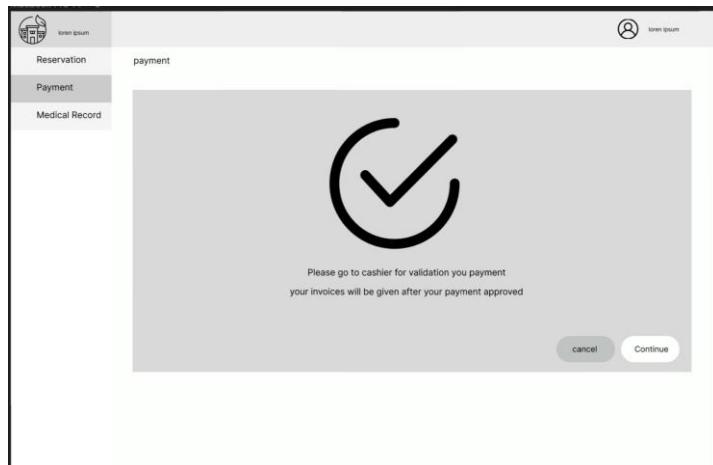
Gambar 3.12 User Interface Pembayaran Secara Tunai

6. Pembayaran Secara Non Tunai



Gambar 3.13 User Interface Pembayaran Secara Non Tunai

7. Pembayaran Sukses



Gambar 3.14 User Interface Pembayaran Sukses

8. List Medical Record

A screenshot of a mobile application interface. The top navigation bar has icons for user profile and settings, with placeholder text 'lorem ipsum'. The left sidebar menu has four items: 'Reservation' (disabled), 'Payment' (disabled), and 'Medical Record' (selected). The main content area is titled 'payment' and displays a table of medical records. The table has columns: 'No', 'ReservationID', 'Date', and 'Time'. The data is as follows:

No	ReservationID	Date	Time
1	#1234	22 Mei 2014	11:30
2	#2222	23 Mei 2014	11:30
2	#3333	24 Mei 2014	11:30

Gambar 3.15 User Interface List Medical Record

9. Detail Medical Record



Gambar 3.16 User Interface Detail Medical Record

3.3 Rancangan Pengujian

Black Box Testing, sebagai salah satu teknik dasar dalam pengujian perangkat lunak, mampu mengidentifikasi sejumlah proses yang tidak dapat terdeteksi oleh White Box Testing. Metode ini menitikberatkan pada pengujian terhadap fungsi-fungsi sistem sesuai dengan kebutuhan pengguna (Rahadi & Vikasari, 2020). Umumnya, Black Box Testing dapat mengungkap beberapa hal berikut:

1. Hilangnya fungsi yang seharusnya tersedia dalam sistem.
2. Kesalahan dalam perancangan antarmuka pengguna
3. Kesalahan dalam perancangan struktur data maupun pada penentuan field-field dalam basis data.
4. Terjadi kegagalan atau malfungsi selama proses eksekusi sistem.
5. Error yang muncul disebabkan oleh proses instalasi sistem yang tidak berjalan dengan semestinya.

Selain *Black Box Testing*, *Integration Testing* juga dapat menjadi rancangan pengujian yang dilakukan dengan memeriksa interaksi antara berbagai microservice. Hal ini penting untuk memastikan bahwa microservice dapat bekerja bersama secara harmonis dan data dapat

dipertukarkan dengan benar di antara layanan. Tools seperti Postman yang dapat digunakan untuk menjalankan pengujian ini.

End-to-End Testing juga dapat dilakukan dengan menguji seluruh sistem dari awal hingga akhir untuk memastikan bahwa semua komponen microservice berfungsi dengan baik secara keseluruhan. Selenium atau Cypress bisa digunakan untuk menjalankan pengujian ini, terutama untuk aplikasi berbasis web.