



PENYUSUNAN DOKUMENTASI SISTEM PADA PERANCANGAN WEB BACKOFFICE APLIKASI MOBILE TELEMEDICINE

Laily Maghfira¹, Esi Putri Silmina²

^{1,2} Teknologi Informasi, Universitas 'Aisyiyah Yogyakarta
Kabupaten Sleman, Daerah Istimewa Yogyakarta, Indonesia 55592

lailymaghfira@gmail.com, esiputrisilmina@unisayogya.ac.id

Abstract

Documentation is a means of conveying information about software. A computer program cannot be considered software without documentation of the software. The compiled documentation can provide many benefits for developers. Proper documentation can shorten the time required to design, create, test, and utilize the software. Designing the back-office web for this mobile telemedicine application requires a formal and structured system design process and meets standards regarding software design. This can be achieved by compiling system documentation as a first step in the ongoing system development. Based on these problems, system documentation is needed in the backoffice web design process for mobile telemedicine applications from the beginning to the end of the design process. The research method used in this research will focus on preparing documentation for designing the back-office web system for mobile telemedicine applications. The application development model used is the waterfall model. The stages of preparing this system documentation are explaining the project background, user persona, user flow, wireframe, prototype, testing, and developing design. The testing process uses the black box testing method. The document preparation process adapts to the project being undertaken. This research results in system documentation, which the developer can later use on duty for further development purposes.

Keywords: Backoffice, Drafting, Software System Documentation, Telemedicine, Web Design

Abstrak

Dokumentasi adalah sarana untuk menyampaikan informasi tentang sebuah perangkat lunak. Suatu program komputer belum dapat dianggap sebagai perangkat lunak tanpa adanya dokumentasi dari perangkat lunak tersebut. Dokumentasi yang disusun dapat memberikan banyak manfaat bagi para *developer*. Penyusunan dokumentasi yang tepat dapat mempersingkat waktu yang diperlukan untuk melakukan perancangan, pembuatan, pengujian dan pemanfaatan sebuah perangkat lunak. Perancangan *web backoffice* aplikasi *mobile telemedicine* ini memerlukan proses perancangan sistem secara formal dan terstruktur dan memenuhi standar mengenai perancangan perangkat lunak (*software*). Hal ini dapat dicapai dengan menyusun dokumentasi sistem sebagai langkah awal dari pengembangan sistem yang akan berlangsung. Berdasarkan masalah tersebut, maka dibutuhkan sebuah dokumentasi sistem dalam proses perancangan *web backoffice* aplikasi *mobile telemedicine* sejak awal hingga akhir proses perancangan. Metode penelitian yang dilakukan adalah penelitian ini akan terfokus dalam penyusunan dokumentasi perancangan sistem *web backoffice* aplikasi *mobile telemedicine*. Model pengembangan aplikasi yang digunakan adalah model *waterfall*. Tahapan dari penyusunan dokumentasi sistem ini, yaitu dengan menjelaskan *background project*, *user persona*, *user flow*, *wireframe*, *prototype*, *testing* dan *develop design*. Proses *testing* menggunakan metode *black box testing*. Proses penyusunan dokumen menyesuaikan dengan proyek yang dikerjakan. Hasil dari penelitian ini adalah sebuah dokumentasi sistem yang nantinya dapat digunakan oleh *developer* yang bertugas untuk keperluan pengembangan selanjutnya.

Kata kunci: Backoffice, Dokumentasi Sistem Perangkat Lunak, Penyusunan, Perancangan Web, Telemedicine

1. PENDAHULUAN

Teknologi informasi saat ini berkembang sangat pesat sesuai dengan perubahan karakteristik masyarakat *modern* yang memiliki mobilitas yang tinggi. Hal ini dapat terlihat dari penggunaan teknologi informasi secara menyeluruh

dalam segala aspek kehidupan. Salah satu aspek yang terkena dampak yaitu dalam hal akses layanan kesehatan. Penggunaan teknologi informasi dalam di bidang kesehatan yang sudah cukup sering ditemui yaitu aplikasi *telemedicine* sebagai media pelayanan medis jarak jauh.

Menurut survei Katadata Insight Center (KIC), layanan *telemedicine* & fasilitas kesehatan (faskes) yang paling banyak digunakan di Indonesia sejak pandemi COVID-19, yaitu Halodoc dengan persentase sebanyak 46,5%, *Telemedicine* RS/Klinik sebanyak 41,8%, Alodokter sebanyak 35,7%, Konsul *online* dokter sebanyak 20,3%, KlikDokter sebanyak 15,5%, isoman.kemkes.go.id sebanyak 10,2%, Good Doctor sebanyak 5,4%, LinkSehat sebanyak 4,4%, Lekasehat sebanyak 2% dan Lainnya sebanyak 1,1% [1].

Christina Rawis, Stanley D. S. Karouw, Sherwin R. U. A. Sompie dari Universitas Sam Ratulangi, Manado, Indonesia melakukan penelitian yang berfokus pada pembuatan dokumentasi SRS berdasarkan sistem informasi akademik yang sudah ada dengan menggunakan gabungan dari 3 model *template* SRS (IEEE, ACM, dan KOMINFO). Hasil penelitian ini adalah sebuah dokumentasi *Software Requirement Specification* Sistem Informasi Akademik UNSRAT dengan menggunakan gabungan 3 *template* [2].

Richi Andrew Serafino dan Meme Susilowati dari Universitas Machung, Malang, Indonesia melakukan penelitian yang berfokus pada pembuatan dokumen *Software Requirement Specification* (SRS) Perancangan Sistem Informasi *Marketing* dengan standar ISO/IEC/IEEE 29148:2018. Metode yang digunakan dalam penelitian ini adalah 3 *Major Phases* untuk melakukan tahapan analisis dan desain. Hasil dari penelitian ini berupa dokumen SRS yang berisi tentang gambaran kebutuhan sistem pada perusahaan tersebut [3].

Penelitian ini berbeda dengan penelitian sebelumnya. Hal ini dapat terlihat pada lokasi penelitian dan hasil penelitian yang dilakukan berupa struktur penulisan dokumentasi sistem pada *web backoffice*. Institusi XYZ telah mengembangkan sebuah aplikasi *mobile telemedicine* yang bertujuan untuk membantu mahasiswa kedokteran tingkat akhir dalam proses pembelajaran sehingga mampu beradaptasi dengan teknologi informasi khususnya dibidang kesehatan. Aplikasi *mobile* tersebut belum memiliki aplikasi *backoffice* untuk melakukan monitoring data, sehingga diperlukannya sebuah perancangan aplikasi *backoffice* dalam bentuk *web*. Proses perancangan *web backoffice* aplikasi *mobile telemedicine* yang sedang berlangsung ini belum terdapat dokumentasi sistem yang menggambarkan tahapan dari proses perancangan.

Berdasarkan latar belakang yang telah dijelaskan, penulisan penelitian ini bertujuan untuk menyusun sebuah dokumentasi sistem berdasarkan pada proyek yang sedang berlangsung. Dokumentasi tersebut akan menyajikan informasi-informasi sejak awal hingga akhir proses perancangan. Rumusan masalah penelitian berupa bagaimana menyajikan informasi terkait proses perancangan dalam bentuk dokumen yang terstruktur agar dapat membantu *developer* untuk pengembangan selanjutnya. Prosedur penelitian dilakukan melalui beberapa

tahap dengan menjelaskan setiap tahapannya (*background project, user persona, user flow, wireframe, prototype, testing dan develop design*). Berdasarkan penjelasan sebelumnya, maka penelitian ini diberi judul “Penyusunan Dokumentasi Sistem Pada Perancangan *Web Backoffice* Aplikasi *Mobile Telemedicine*”

Sistem Informasi

Maydianto dan Ridho berpendapat bahwa sistem informasi merupakan kerangka terorganisir yang mencakup perangkat lunak, perangkat keras, prosedur, data, dan orang-orang yang bekerja bersama untuk mengumpulkan, menyimpan, mengelola, mengolah, dan menyampaikan informasi yang relevan dan bermanfaat kepada pengguna dalam sebuah organisasi [4].

Telemedicine

Waller dan Stotler berpendapat bahwa definisi *telemedicine* merupakan perpaduan antara kata *medicine* dan *tele*, yang berarti seni dan ilmu pemeliharaan kesehatan, serta pencegahan dan penyembuhan penyakit yang dilakukan melalui jarak jauh [5].

Aplikasi *Backoffice*

Menurut Saputra, Prasetyo, dan Khairudin, Aplikasi *backoffice* merupakan salah satu bagian penting dari sebuah bisnis ataupun produk. Aplikasi *backoffice* tidak berjalan sendiri, biasanya aplikasi ini memiliki suatu objek untuk di monitor. *User* yang akan menggunakan *backoffice* nantinya adalah operator/admin yang sudah ditunjuk oleh perusahaan/pihak berwenang untuk memonitoring maupun membuat laporan dari hasil kinerja sebuah produk/jasa. Pembuatan aplikasi *backoffice* bertujuan untuk membantu memonitoring kinerja dari aplikasi utamanya (aplikasi *mobile*) dimana aplikasi utamanya digunakan oleh *end user* [6].

Dokumentasi Perangkat Lunak

Dokumentasi adalah sebuah artefak yang disusun dengan tujuan untuk menyampaikan informasi lengkap dan akurat tentang sistem perangkat lunak yang berjalan. Penyusunan dokumentasi yang baik dapat dilihat dari hasil penulisan, apakah dokumentasi yang disusun telah lengkap dan akurat atau belum. Hasil dokumentasi yang bagus dapat memberikan manfaat bagi para *developer*. Sebaliknya, dokumentasi yang buruk sejak awal proses perancangan perangkat lunak (*software*) bisa jadi akan mempengaruhi baik atau tidaknya kualitas *software* yang dibuat.

Perancangan

Menurut N. Rizqya, perancangan merupakan pemilihan proses maupun data dalam proses pembuatan sistem yang baru. Tujuan perancangan sistem ialah untuk menjelaskan gambaran rancangan atau *blue print* secara lengkap guna menjadi bahan panduan pengembangan dan *developer* yang

tentunya telah disesuaikan dengan perangkat lunak yang ada [7].

Pengujian Sistem

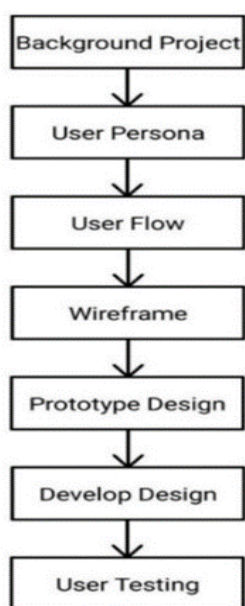
Pengujian sistem dilakukan bertujuan untuk melakukan evaluasi terhadap *software* serta menentukan apakah *software* yang dirancang atau dikembangkan sudah sesuai dengan fungsionalitas yang diharapkan [8].

2. METODE PENELITIAN

Penelitian ini akan terfokus dalam penyusunan dokumentasi perancangan sistem *web backoffice* aplikasi *mobile telemedicine*. Prosedur penyusunan dokumentasi sistem akan menyesuaikan dengan proses perancangan.

2.1 Tahapan Penelitian

Terdapat 7 tahapan dalam proses penelitian diantaranya: (1) tahapan *background project* yang dilakukan dengan menjelaskan tujuan perancangan *web backoffice* aplikasi *mobile telemedicine*; (2) tahapan *user persona* yang dilakukan dengan menjelaskan siapa pengguna akhir web; (3) tahapan *user flow* yang dilakukan dengan menjelaskan alur pengguna saat akan menggunakan produk (*web*) yang terbagi menjadi beberapa bagian; (4) tahapan *wireframe* yang dilakukan dengan menyajikan konsep desain dan fungsi dasar; (5) tahapan *prototype design* yang dilakukan dengan menyajikan tampilan sistem yang telah diubah kedalam sistem yang bekerja (*working system*) dan memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem; (6) tahapan *user testing* yang dilakukan dengan menyajikan informasi pengujian sistem dengan menggunakan metode *black box testing*; (7) *develop design* yang dilakukan dengan menyajikan informasi terkait *environment* dalam perancangan *web backoffice* aplikasi *mobile telemedicine*. Detail dari tahapan yang telah dijelaskan dapat terlihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.2 Model Pengembangan Aplikasi

Model pengembangan yang digunakan pada proses perancangan *web backoffice* adalah *waterfall*. Model *waterfall* memiliki 5 tahapan diantaranya: (1) tahapan *requirement analysis and definition* dilakukan dengan menganalisis kebutuhan *user* serta masalah yang dihadapi; (2) tahapan *system and software design* dilakukan dengan implementasi spesifikasi sistem yang telah diperoleh pada tahap 1 ke dalam bentuk desain *interface* pengembangan sistem; (3) tahapan *implementation and unit testing* dilakukan dengan mengimplementasikan hasil rancangan sistem ke dalam rangkaian program dengan bahasa pemrograman yang sudah ditentukan; (4) tahapan *integration and unit testing* dilakukan dengan memeriksa dan menguji sistem untuk mengetahui apakah terdapat *error* pada sistem tersebut [9].

2.3 Metode Pengujian

Pada fase *user testing* metode pengujian yang digunakan untuk *web backoffice* aplikasi *mobile telemedicine* institusi XYZ adalah *Black Box Testing*. *Black Box Testing* merupakan sebuah metode yang digunakan bertujuan untuk menguji sebuah *software* tanpa harus memperhatikan detail *software* tersebut [10].

Pengujian *Black Box* ialah kondisi pengujian perangkat lunak (*software*) yang berfokus pada spesifikasi fungsionalnya, penguji akan mendefinisikan kondisi dari input dan spesifikasi fungsional perangkat lunak [11].

2.4 Dokumentasi Requirement Engineering

Requirements engineering merupakan cabang dari *software engineering* yang berkaitan dengan tujuan tujuan yang diinginkan, fungsi, dan kendala pada sistem perangkat lunak [12].

Requirements engineering adalah serangkaian aktivitas terstruktur yang membantu *developer* memahami masalah yang coba dipecahkan serta mendokumentasikan spesifikasi sistem untuk para *stakeholders* dan *engineers* yang terlibat dalam pengembangan sistem. Beberapa aktivitas yang merupakan dasar dari semua proses RE (*requirements engineering*) adalah (1) elisitasi; (2) analisis; (3) validasi; (4) negosiasi; (5) dokumentasi dan (6) manajemen. Hasil dari proses RE adalah sebuah dokumen persyaratan yang mendefinisikan apa yang akan diimplementasikan [13].

3. HASIL DAN PEMBAHASAN

3.1 Background Project

Perancangan *web backoffice* ini bertujuan untuk mengatasi masalah berupa dibutuhkannya sebuah *software* yang dapat membantu kinerja dari aplikasi *mobile telemedicine* yang sudah dibuat/dirancang sebelumnya. *Web* ini memiliki fitur-fitur seperti: *login*, *register*, *dashboard*, *pasien*, *konsultasi*, *dokter*, *spesialisasi*, *activity log* dan *logout*. Harapannya *web* ini nanti dapat digunakan agar dapat mempermudah

admin dalam hal mengelola data admin, pasien, dokter dan data lainnya.

3.2 User Persona

Menurut Szerovay, *user persona* dibuat bertujuan untuk mewakili sekelompok pengguna penting yang mencerminkan pola berdasarkan perilaku, tujuan, dan sikap [14].

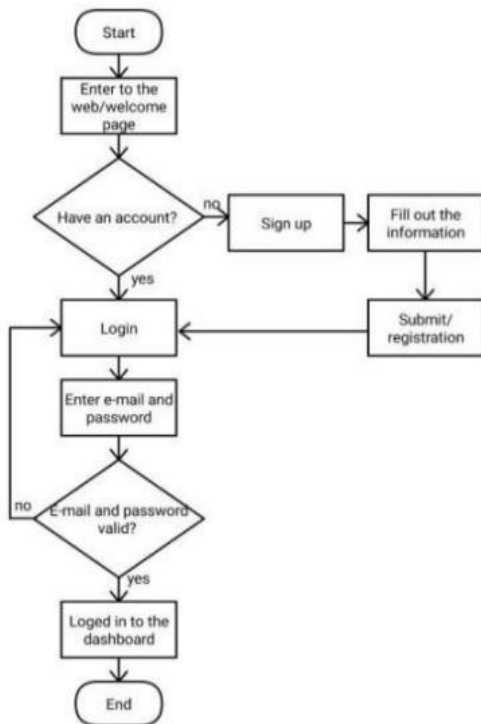
Tabel 1 menjelaskan tentang *user persona* bertujuan menggambarkan *user type* dan *user expectation* dari pengguna *web*.

Tabel 1. Tabel User Persona

User type	User Expectation
Admin	User admin dapat memiliki hak akses untuk mengelola seluruh data yang ada seperti: melihat data konsultasi pasien, spesialisasi dokter, <i>log activity</i> , pasien, mengubah data pasien, serta melihat/ menambah/ mengubah/menghapus data dokter.

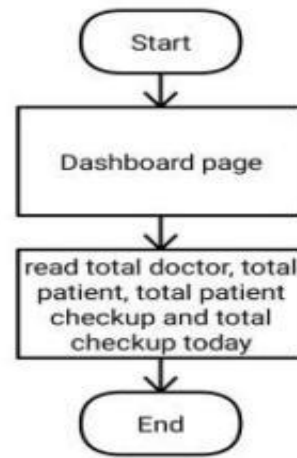
3.3 User flow

Menurut Maulana, *user flow* merupakan urutan penyelesaian tugas yang dilakukan oleh pengguna pada sebuah aplikasi. merupakan alur yang dilalui oleh pengguna [15]. Alur pengguna (*user flow*) biasanya divisualisasikan dalam bentuk diagram *flow chart* guna mempermudah setiap proses yang dilalui oleh *user* saat berinteraksi dengan sistem [16]. Berikut beberapa *user flow* dari *web backoffice* yang ditunjukkan pada Gambar 2 hingga Gambar 9.



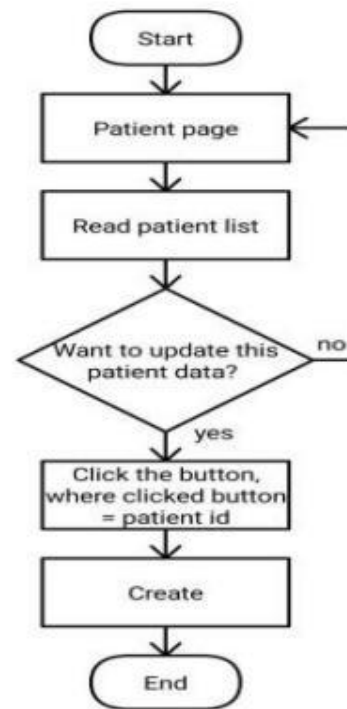
Gambar 2. User Flow Login dan Register

Gambar 2 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* akan melakukan *login* dan *register*.



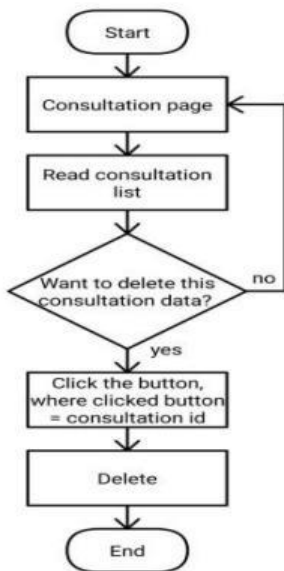
Gambar 3. User Flow Dashboard

Gambar 3 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* berhasil melakukan *login* dan *register* yang kemudian *user* akan diarahkan menuju ke halaman *dashboard*.



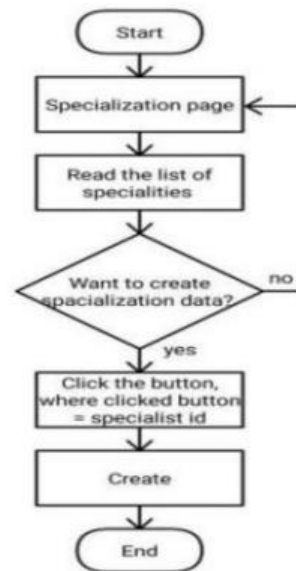
Gambar 4. User Flow Pasien

Gambar 4 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin beralih menuju menu/halaman pasien serta melakukan *update* status pasien (*active* atau *suspend*).



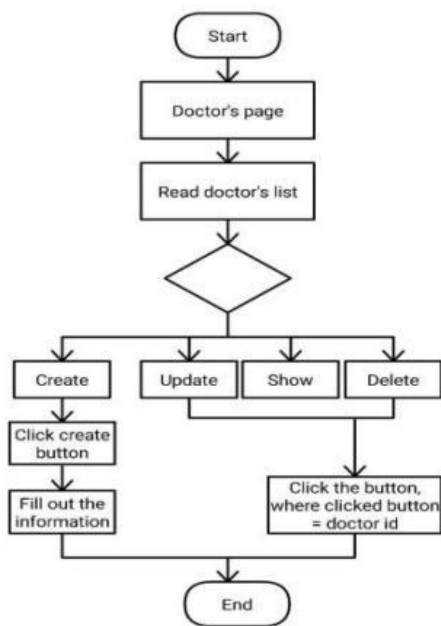
Gambar 5. User Flow Konsultasi Pasien

Gambar 5 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin beralih menuju menu/halaman konsultasi pasien serta melakukan hapus data konsultasi yang ingin dihapus.



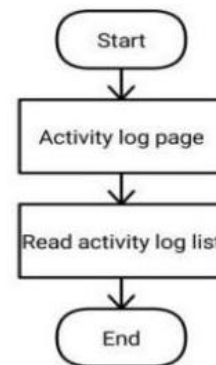
Gambar 7. User Flow Spesialisasi

Gambar 7 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin beralih menuju menu/halaman spesialisasi serta melakukan tambah data spesialisasi dokter.



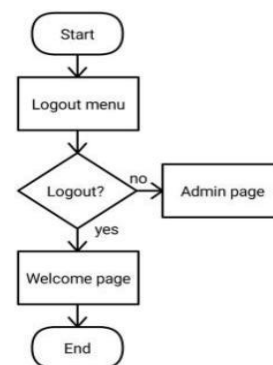
Gambar 6. User Flow Dokter

Gambar 6 yang diperlihatkan di atas menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin beralih menuju menu/halaman dokter serta melakukan tambah, *update*, lihat dan hapus data dokter yang diinginkan.



Gambar 8. User Flow Log Activity

Gambar 8 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin beralih menuju menu/halaman *Log Activity* untuk melihat aktivitas apa saja yang telah dilakukan.



Gambar 9. User Flow Logout

Gambar 9 menjelaskan bagaimana alur pengguna (*user flow*) dari proses ketika *user* ingin keluar dari *web* yang kemudian beralih kembali menuju halaman *welcome*.

3.4 Wireframe

Wireframe merupakan suatu konsep awal dalam pengembangan desain *website* maupun aplikasi *mobile* yang lebih banyak menentukan tata letak yang tepat untuk posisi tombol, fitur, dan konten dari sebuah aplikasi [17].

Menurut Segara, *wireframing* merupakan langkah penting dalam proses desain atau perancangan sebuah media digital (*screen design process*). *Wireframe* diterapkan dengan tujuan mempermudah penyusunan sebuah konten dan *user experience* yang akan diterapkan [18]. Berikut merupakan tampilan *wireframe* dari *web backoffice* yang ditujukan pada Gambar 10 hingga Gambar 25.



Gambar 10. Halaman Welcome

Gambar 10 memvisualisasikan rancangan *wireframe* halaman *welcome*.



Gambar 11. Halaman Login

Gambar 11 memvisualisasikan rancangan *wireframe* halaman *login*.



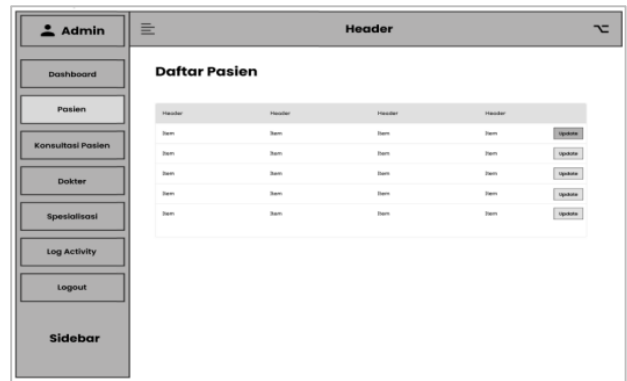
Gambar 12. Halaman Register

Gambar 12 yang diperlihatkan di atas memvisualisasikan rancangan *wireframe* halaman *register*.



Gambar 13. Halaman Dashboard

Gambar 13 memvisualisasikan rancangan *wireframe* halaman *dashboard*.



Gambar 14. Halaman Pasien

Gambar 14 memvisualisasikan rancangan *wireframe* halaman pasien.



Gambar 15. Form update Pasien

Gambar 15 yang diperlihatkan di atas memvisualisasikan rancangan *wireframe* halaman pasien pada bagian *form update* status pasien.



Gambar 16. Halaman Konsultasi Pasien

Gambar 16 memvisualisasikan rancangan *wireframe* halaman konsultasi pasien.



Gambar 17. Form Hapus Konsultasi Pasien

Gambar 17 memvisualisasikan rancangan *wireframe* halaman konsultasi pasien pada bagian *form* hapus data konsultasi.



Gambar 18. Halaman Dokter

Gambar 18 memvisualisasikan rancangan *wireframe* halaman dokter.



Gambar 19. Form Tambah Dokter

Gambar 19 memvisualisasikan rancangan *wireframe* halaman dokter pada bagian *form* tambah dokter.



Gambar 20. Form Update Dokter

Gambar 20 memvisualisasikan rancangan *wireframe* halaman dokter pada bagian *form update* dokter.



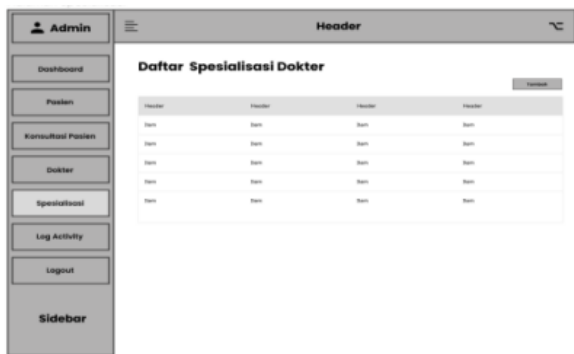
Gambar 21. Form Lihat Dokter

Gambar 21 memvisualisasikan rancangan *wireframe* halaman *welcome*.



Gambar 22. Form Hapus Dokter

Gambar 22 memvisualisasikan rancangan *wireframe* halaman dokter pada bagian *form* hapus dokter.



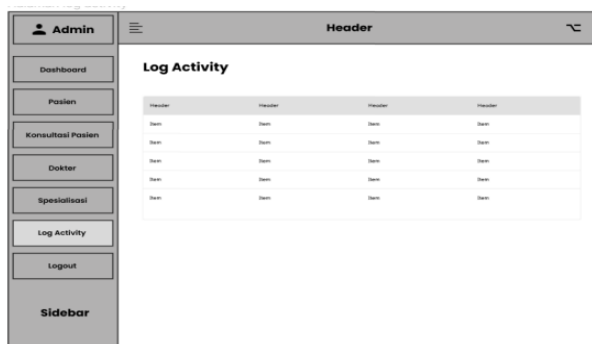
Gambar 23. Halaman Spesialisasi

Gambar 23 yang diperlihatkan diatas memvisualisasikan rancangan *wireframe* halaman spesialisasi.



Gambar 24. Form Tambah Spesialisasi

Gambar 24 memvisualisasikan rancangan *wireframe* halaman spesialisasi pada bagian *form* tambah spesialisasi.



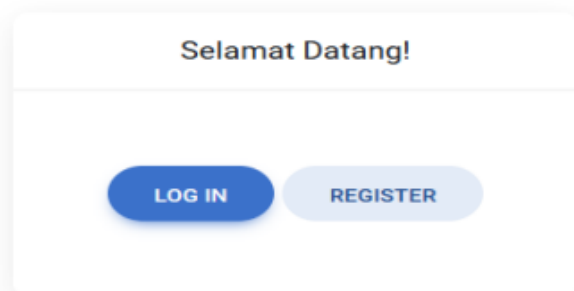
Gambar 25. Halaman Log Activity

Gambar 25 memvisualisasikan rancangan *wireframe* halaman *log activity*.

3.5 Prototype Design

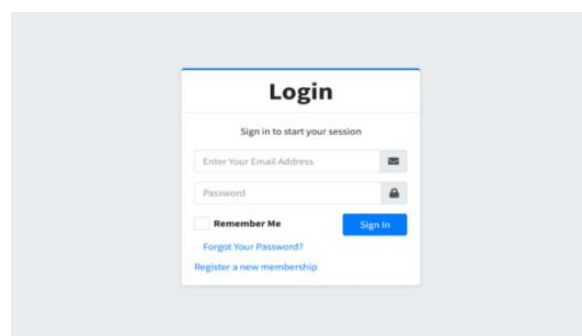
Proses menghasilkan sebuah *prototype* dikenal sebagai *prototyping*. *Prototyping* merupakan proses perancangan *prototype* yang merupakan model dari sebuah *prototype* produk yang mungkin belum terdapat semua fitur produk yang sebenarnya, tetapi sudah memiliki beberapa fitur utama dari produk sebenarnya dan biasa digunakan untuk keperluan pengujian atau bahan uji coba sebelum melanjutkan ke fase pembuatan produk asli [19]. Adanya *prototyping* ini, *developer* maupun *user* dapat saling

berinteraksi selama proses perancangan atau pembuatan perangkat lunak. Berikut merupakan tampilan *prototype* dari *web backoffice* yang ditujukan pada Gambar 26 hingga Gambar 41.



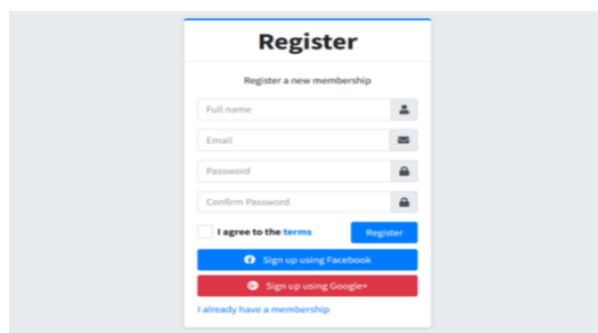
Gambar 26. Halaman Welcome

Gambar 26 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman *welcome*.



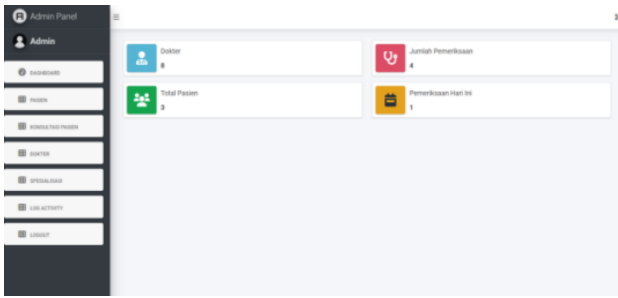
Gambar 27. Halaman Login

Gambar 27 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman *login*.



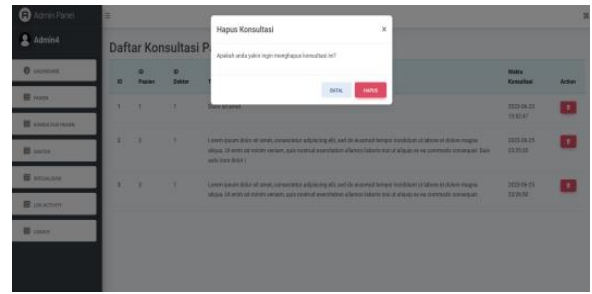
Gambar 28. Halaman Register

Gambar 28 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman *register*.



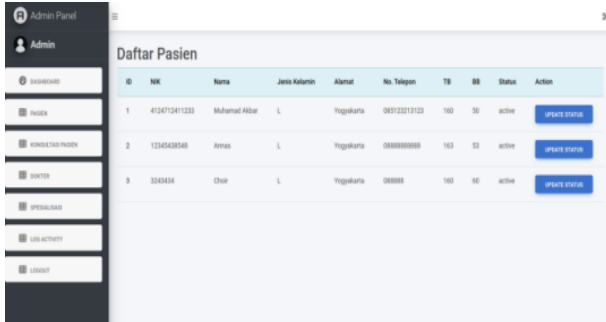
Gambar 29. Halaman Dashboard

Gambar 29 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman *dashboard*.



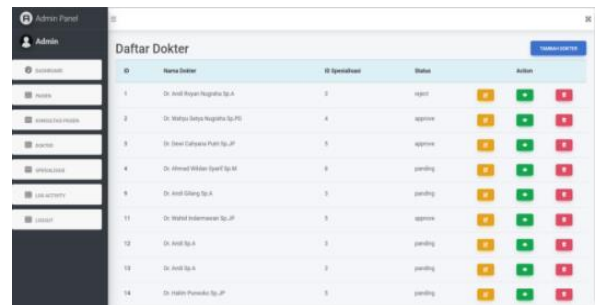
Gambar 33. Form Hapus Konsultasi Pasien

Gambar 33 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman konsultasi pasien pada bagian *form* hapus konsultasi.



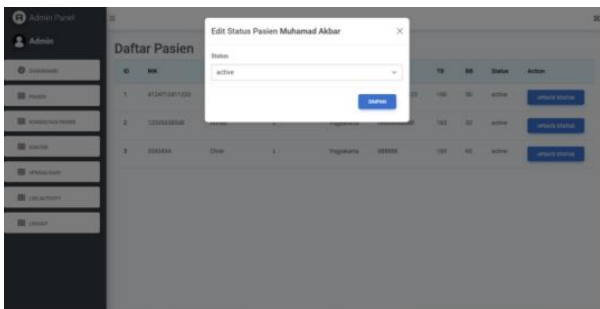
Gambar 30. Halaman Pasien

Gambar 30 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman pasien.



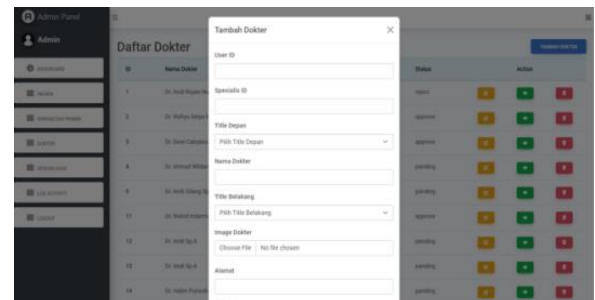
Gambar 34. Halaman Dokter

Gambar 34 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman dokter.



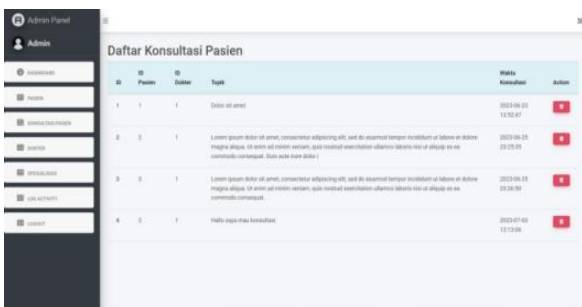
Gambar 31. Form Update Pasien

Gambar 31 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman pasien pada bagian *form* update pasien.



Gambar 35. Form Tambah Dokter

Gambar 35 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman dokter pada bagian *form* tambah dokter.



Gambar 32. Halaman Konsultasi Pasien

Gambar 32 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman konsultasi pasien.



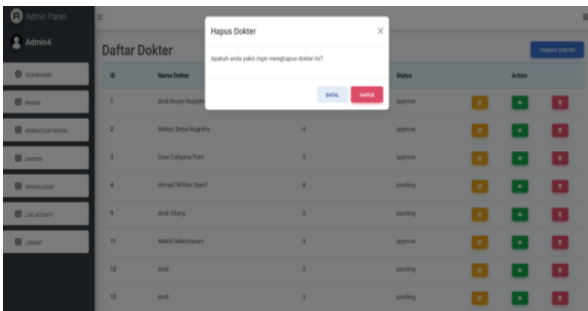
Gambar 36. Form Update Dokter

Gambar 36 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman dokter pada bagian *form* update dokter.



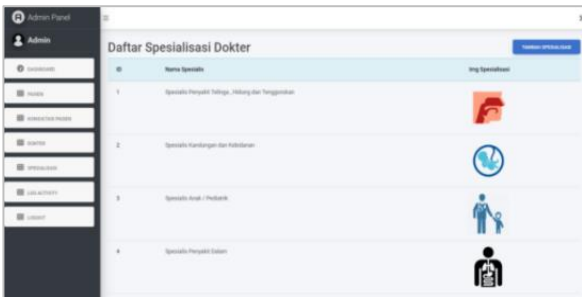
Gambar 37. Form Lihat Dokter

Gambar 37 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman dokter pada bagian *form* lihat dokter.



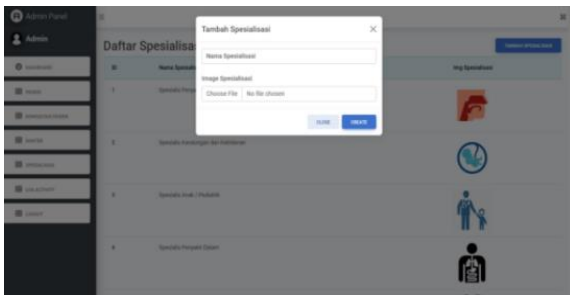
Gambar 38. Form Hapus Dokter

Gambar 38 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman dokter pada bagian *form* hapus dokter.



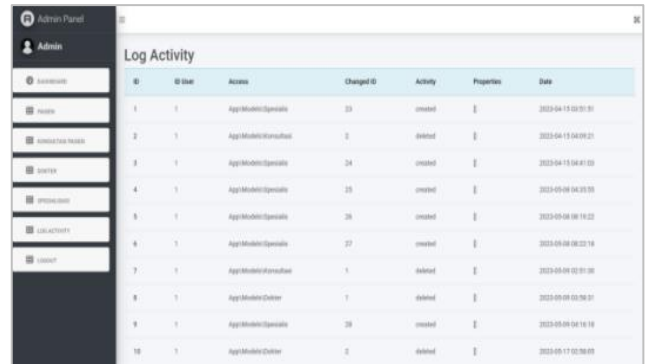
Gambar 39. Halaman Spesialisasi

Gambar 39 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman spesialisasi.



Gambar 40. Form Tambah Spesialisasi

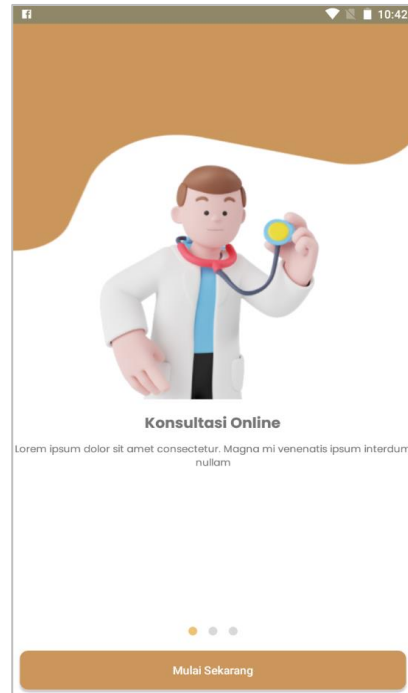
Gambar 40 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman spesialisasi pada bagian *form* tambah spesialisasi.



Gambar 41. Halaman Log Activity

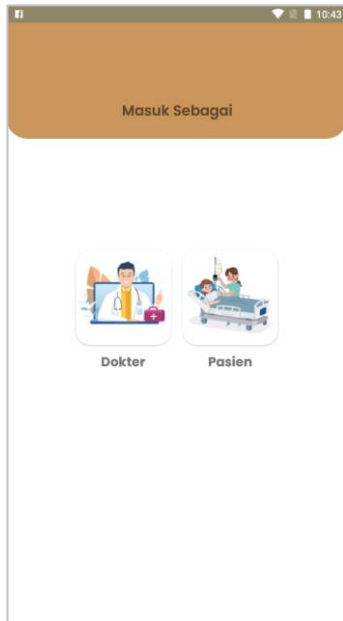
Gambar 41 memvisualisasikan hasil implementasi dari rancangan *wireframe* halaman *log activity*.

Tampilan *prototype* dari aplikasi *mobile telemedicine* yang ditujukan pada Gambar 42 hingga Gambar 55.



Gambar 42. Halaman Welcome

Gambar 42 memvisualisasikan halaman *welcome* aplikasi *mobile telemedicine*.



Gambar 43. Halaman *Login*

Gambar 43 memvisualisasikan halaman *login* aplikasi *mobile telemedicine*.



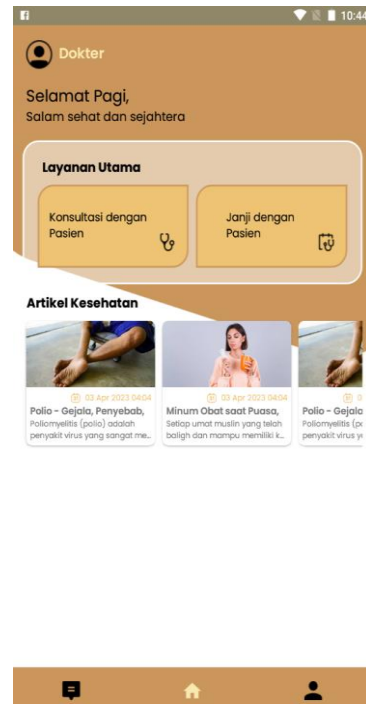
Gambar 45. Halaman *Login Pasien*

Gambar 45 memvisualisasikan halaman *login* pasien aplikasi *mobile telemedicine*.



Gambar 44. Halaman *Login Dokter*

Gambar 44 memvisualisasikan halaman *login* dokter aplikasi *mobile telemedicine*.



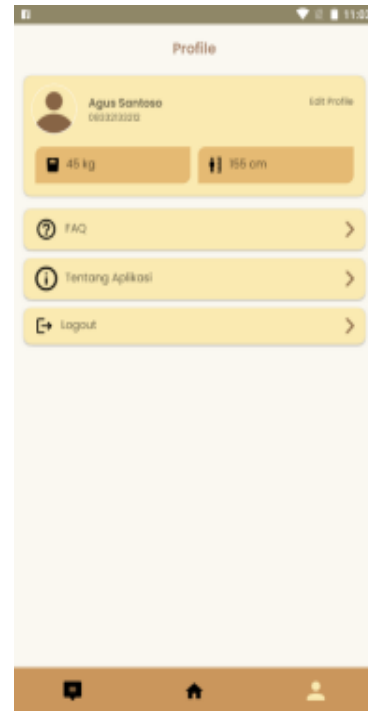
Gambar 46. Halaman *Utama Dokter*

Gambar 46 memvisualisasikan halaman *utama* dokter aplikasi *mobile telemedicine*.



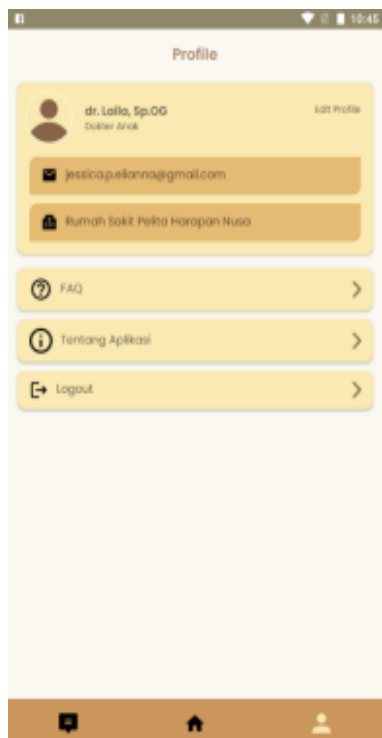
Gambar 47. Halaman Utama Pasien

Gambar 47 memvisualisasikan halaman utama pasien aplikasi *mobile telemedicine*.



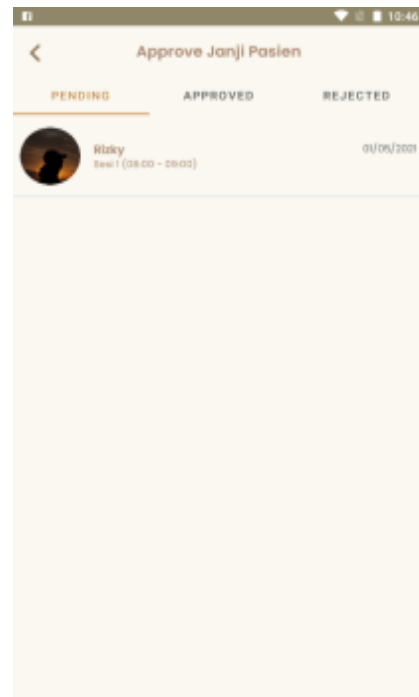
Gambar 49. Halaman Profil Pasien

Gambar 49 memvisualisasikan halaman profil pasien aplikasi *mobile telemedicine*.



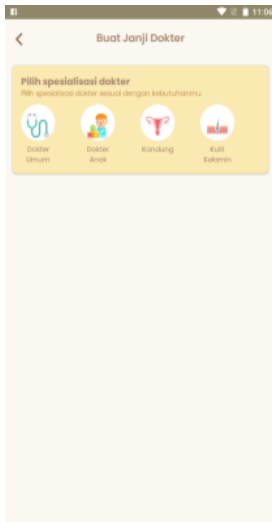
Gambar 48. Halaman Profil Dokter

Gambar 48 memvisualisasikan halaman profil dokter aplikasi *mobile telemedicine*.



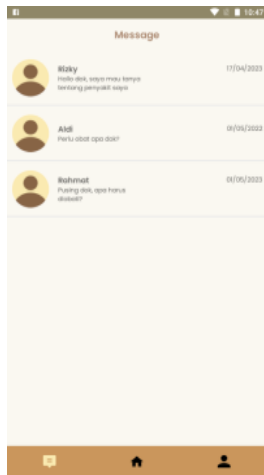
Gambar 50. Halaman Approve Janji Pasien

Gambar 50 memvisualisasikan halaman *approve janji* pasien pada menu dokter aplikasi *mobile telemedicine*.



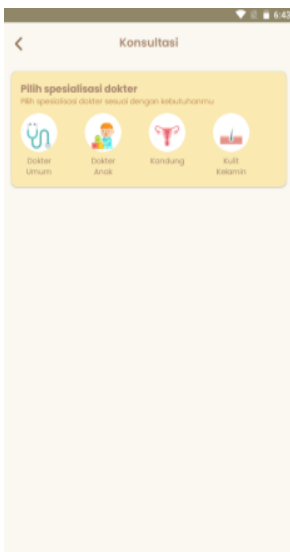
Gambar 51. Halaman Buat Janji Dokter

Gambar 51 memvisualisasikan halaman buat janji dokter pada menu pasien aplikasi *mobile telemedicine*.



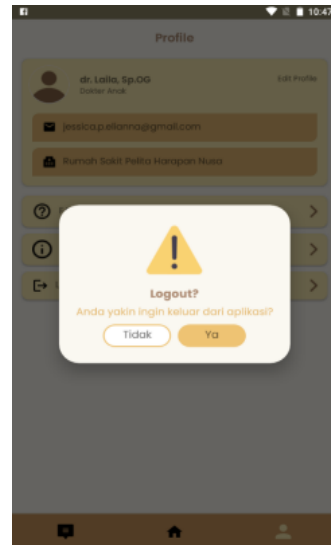
Gambar 52. Halaman Konsultasi Dokter

Gambar 52 memvisualisasikan halaman konsultasi pada menu dokter aplikasi *mobile telemedicine*.



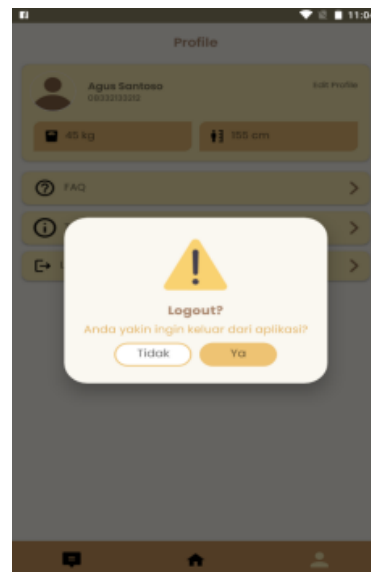
Gambar 53. Halaman Konsultasi Pasien

Gambar 53 memvisualisasikan halaman konsultasi pada menu Pasien aplikasi *mobile telemedicine*.



Gambar 54. Halaman Logout Dokter

Gambar 54 memvisualisasikan halaman *logout* dokter aplikasi *mobile telemedicine*.



Gambar 55. Halaman Logout Pasien

Gambar 55 memvisualisasikan halaman *logout* Pasien aplikasi *mobile telemedicine*.

3.6 User Testing

Bagian ini menjelaskan tentang hasil yang didapatkan dari pengujian *web backoffice* aplikasi *mobile telemedicine* menggunakan metode *black box testing* berfokus pada uji fungsionalitas. Berdasarkan pengujian fitur-fitur *web* yang sudah dilakukan, didapatkan bahwa semua fitur berjalan dengan baik sesuai yang diharapkan. Visualisasi detail hasil pengujian akan tampak pada Tabel 2 hingga Tabel 11.

Hasil *Test Scenario*: Verifikasi Halaman *Welcome* dapat dilihat pada Tabel 2.

Tabel 2. *Test Scenario:* Verifikasi Halaman *Welcome*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	Menekan tombol <i>login</i>	Beralih menuju halaman <i>login</i>	Beralih menuju halaman <i>login</i>	<i>Pass</i>
2	Menekan tombol <i>register</i>	Beralih menuju halaman <i>register</i>	Beralih menuju halaman <i>register</i>	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Login* dapat dilihat pada Tabel 3.

Tabel 3. *Test Scenario:* Verifikasi Halaman *Login*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	<i>E-mail:</i> admin1@gmail.com <i>Password:</i> Admin1	Berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	Berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	<i>Pass</i>
2	<i>E-mail:</i> admin@gmail.com <i>Password:</i> Admin1	Berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	Tidak berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Register* dapat dilihat pada Tabel 4.

Tabel 4. *Test Scenario:* Verifikasi Halaman *Register*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	<i>Fullname:</i> Admin <i>E-mail:</i> admin1@gmail.com <i>Password:</i> Admin1 <i>Confirm Password:</i> Admin1	Berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	Berhasil <i>login</i> dan menampilkan halaman <i>dashboard</i>	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Dashboard* dapat dilihat pada Tabel 5.

Tabel 5. *Test Scenario:* Verifikasi Halaman *Dashboard*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	Ketika berhasil <i>login</i> atau <i>register</i> , akan diarahkan menuju halaman <i>dashboard</i>	Dapat menampilkan data total dokter, pasien, dan jumlah pemeriksaan hari ini	Dapat menampilkan data total dokter, pasien, dan jumlah pemeriksaan hari ini	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Pasien* dapat dilihat pada Tabel 6.

Tabel 6. *Test Scenario:* Verifikasi Halaman *Pasien*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	Menekan tombol menu <i>pasien</i>	Beralih menuju halaman <i>pasien</i> dan menampilkan seluruh data <i>pasien</i>	Beralih menuju halaman <i>pasien</i> dan menampilkan seluruh data <i>pasien</i>	<i>Pass</i>
2	Menekan tombol <i>update</i> pada menu <i>pasien</i>	Dapat melakukan <i>update</i> status data <i>pasien</i>	Dapat melakukan <i>update</i> status data <i>pasien</i>	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Konsultasi Pasien* dapat dilihat pada Tabel 7.

Tabel 7. *Test Scenario:* Verifikasi Halaman *Konsultasi Pasien*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	Menekan tombol menu <i>konsultasi pasien</i>	Beralih menuju halaman <i>konsultasi pasien</i> dan menampilkan seluruh data <i>konsultasi</i>	Beralih menuju halaman <i>pasien</i> dan menampilkan seluruh data <i>pasien</i>	<i>Pass</i>
2	Menekan tombol <i>delete</i> pada menu <i>konsultasi pasien</i>	Dapat melakukan <i>delete</i> data <i>konsultasi</i>	Dapat melakukan <i>delete</i> data <i>konsultasi</i>	<i>Pass</i>

Hasil *Test Scenario:* Verifikasi Halaman *Dokter* dapat dilihat pada Tabel 8.

Tabel 8. *Test Scenario:* Verifikasi Halaman *Dokter*

<i>Test Case</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Pass/fail?</i>
1	Menekan tombol menu <i>dokter</i>	Beralih menuju halaman <i>dokter</i> dan menampilkan seluruh data <i>dokter</i>	Beralih menuju halaman <i>dokter</i> dan menampilkan seluruh data <i>dokter</i>	<i>Pass</i>
2	Menekan tombol <i>tambah</i> pada menu <i>dokter</i>	Dapat menambah data <i>dokter</i>	Dapat menambah data <i>dokter</i>	<i>Pass</i>
3	Menekan tombol <i>update</i> pada menu <i>dokter</i>	Dapat melakukan <i>update</i> data <i>dokter</i>	Dapat melakukan <i>update</i> data <i>dokter</i>	<i>Pass</i>
4	Menekan tombol <i>show</i> pada	Dapat melihat data <i>dokter</i>	Dapat melihat data <i>dokter</i>	<i>Pass</i>

Test Case	Input	Expected Output	Actual Output	Pass/fail?
5	menu dokter Menekan tombol <i>delete</i> pada menu dokter	Dapat melakukan <i>delete</i> data dokter	Dapat melakukan <i>delete</i> data dokter	Pass

Hasil *Test Scenario*: Verifikasi Halaman Spesialisasi dapat dilihat pada Tabel 9.

Tabel 9. *Test Scenario*: Verifikasi Halaman Spesialisasi

Test Case	Input	Expected Output	Actual Output	Pass/fail?
1	Menekan tombol menu spesialisasi	Beralih menuju halaman spesialisasi dan menampilkan seluruh data spesialisasi dokter	Beralih menuju halaman spesialisasi dan menampilkan seluruh data spesialisasi dokter	Pass
2	Menekan tombol tambah pada menu spesialisasi	Dapat menambah data spesialisasi dokter	Dapat menambah data spesialisasi dokter	Pass

Hasil *Test Scenario*: Verifikasi Halaman *Log Activity* dapat dilihat pada Tabel 10.

Tabel 10. *Test Scenario*: Verifikasi Halaman *Log Activity*

Test Case	Input	Expected Output	Actual Output	Pass/fail?
1	Menekan tombol menu <i>log activity</i>	Beralih menuju halaman <i>log activity</i> dan menampilkan seluruh data <i>log activity</i>	Beralih menuju halaman <i>log activity</i> dan menampilkan seluruh data <i>log activity</i>	Pass

Hasil *Test Scenario*: Verifikasi Halaman *Logout* dapat dilihat pada Tabel 11.

Tabel 11. *Test Scenario*: Verifikasi Halaman *Logout*

Test Case	Input	Expected Output	Actual Output	Pass/fail?
1	Menekan tombol <i>logout</i>	Berhasil <i>logout</i> dan kembali menuju halaman <i>welcome</i>	Berhasil <i>logout</i> dan kembali menuju halaman <i>welcome</i>	Pass

3.7 Develop Design

Bagian ini menjelaskan tentang *environment* yang digunakan pada perancangan *web backoffice* aplikasi *mobile telemedicine*. Perancangan *web* menggunakan *Framework* Laravel dengan pendekatan arsitektur MVC dan penerapan PHP MVC *Framework* berbasis *web* pada perancangan.

Penjelasan detail *environment API* dapat dilihat pada Tabel 12.

Tabel 12. *API*

Language	Framework	Architecture
PHP	Laravel	MVC (Model - View - Controller)

Penjelasan detail *environment Front-end dan Back-end* dapat dilihat pada Tabel 13.

Tabel 13. Tabel *Front-end* dan *Back-end*

Language	Framework	Architecture	Database
PHP 8.2.0	Laravel 10.13.2	MVC (Model - View - Controller)	MySQL

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan:

- Tersusunnya sebuah dokumentasi sistem perancangan *web backoffice* aplikasi *mobile telemedicine* pada institusi XYZ sesuai dengan yang diharapkan.
- Berdasarkan pengujian dengan *black box testing* yang telah dilakukan, diketahui bahwa fungsional *web backoffice* aplikasi *mobile telemedicine* pada institusi XYZ berjalan sesuai kebutuhan pengguna dengan persentase keberhasilan sebesar 100%.

Harapannya dokumen yang telah tersusun dapat menjadi panduan *developer* ketika akan melakukan pengembangan selanjutnya.

DAFTAR PUSTAKA

- C. M. Annur, "Layanan Telemedicine yang Paling Banyak Digunakan di Indonesia, Apa Saja?." databoks, 2022. <https://databoks.katadata.co.id/> (accessed Jul. 25, 2023).
- C. Rawis, S. D. S. Karouw, and Sherwin R. U. A. Sompie, "Software Requirement Specification Academic Information System of Sam Ratulangi University," J. Tek. Elektro dan Komput., vol. 10, no. 2, pp. 107–118, 2021, [Online]. Available: <https://ejournal.unsrat.ac.id/index.php/elekdankom/article/download/34108/32441>.
- R. A. Serafintino, M. Susilowati, S. Informasi, and U. Machung, "Dokumen Software Requirement

- Spesification (SRS) Sistem Informasi Pemasaran Usaha Jasa Percetakan Dan Iklan,” Kurawal J. Teknol. Inf. dan Ind., vol. 5, no. 2, pp. 117–128, 2022.
- [4] Maydianto and M. R. Ridho, “Rancang Bangun Sistem Informasi Point of Sale Dengan Framework Codeigniter Pada Cv Powershop,” J. Comasie, vol. 04, no. 02, pp. 50–59, 2021.
- [5] M. Waller and C. Stotler, “Telemedicine: a Primer,” *Curr. Allergy Asthma Rep.*, vol. 18, no. 10, p. 54, 2018, doi: 10.1007/s11882-018-0808-4.
- [6] S. Saputra and A. B. Prasetyo, “Scientia Sacra : Jurnal Sains , Teknologi dan Masyarakat Pembuatan dan Implementasi Aplikasi Backoffice untuk Menopang Kebutuhan Monitoring dan Reporting dari Aplikasi PPOB Mobile,” *J. Sci. Sacra J. Sains, Teknol. dan Masy.*, vol. 2, no. 3, pp. 263–271, 2022.
- [7] N. Rizqya, “Perancangan Sistem Informasi Akuntansi Laporan Posisi Keuangan Pada UMKM Berbasis WEB (Studi Kasus UMKM Home Catering),” *J. Ilm. Komputasi*, vol. 19, no. 3, pp. 383–390, 2020, doi: 10.32409/jikstik.19.3.65.
- [8] M. Sholeh, I. Gisfas, Cahiman, and M. A. Fauzi, “Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods,” *J. Phys. Conf. Ser.*, vol. 1823, no. 1, 2021, doi: 10.1088/1742-6596/1823/1/012029.
- [9] A. Saputra, C. F. I. Safitri, F. Fitriyani, Y. Gulo, and T. Desyani, “Pengembangan Aplikasi Kasir Menggunakan Model Waterfall,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 4, no. 2, p. 86, 2021, doi: 10.32493/jtsi.v4i2.10167.
- [10] N. M. D. Febriyanti, A. A. K. O. Sudana, and I. N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” *J. Ilm. Teknol. dan Komput.*, vol. 2, no. 3, pp. 535–544, 2021.
- [11] T. Hidayat and M. Muttaqin, “Pengujian sistem informasi pendaftaran dan pembayaran wisuda online menggunakan black box testing dengan metode equivalence partitioning dan boundary value analysis,” *J. Tek. Inform. UNIS*, vol. 6, no. 1, pp. 2252–5351, 2018. [9]
- [12] M. K. Phillip A. Laplante, *Requirements Engineering for Software and Systems*, 4th ed. New York: Auerbach Publications, 2022. doi: <https://doi.org/10.1201/9781003129509>.
- [13] I. Sommerville, “Integrated Requirements Engineering: A tutorial,” *IEEE Softw.*, pp. 16–23, 2005, doi: 10.1109/MS.2005.13.
- [14] K. Szerovay, “Creating Personas — Part 1,” *UX Knowledge Base Sketch*, 2017. <https://uxknowledgebase.com/> (accessed Aug. 25, 2023).
- [15] R. T. Maulana, “Perancangan User Interface User Experience Dengan Metode User Centered Design Pada Aplikasi Mobile Auctentik,” *Informatics Eng.*, p. 60, 2020, [Online]. Available: <https://dspace.uui.ac.id/123456789/28891>.
- [16] R. P. Sutanto, “Analisis User Flow pada Website Pendidikan: Studi Kasus Website DKV UK Petra,” *J. Desain Komun. Vis. Nirmana*, vol. 22, no. 1, pp. 41–51, 2022, doi: 10.9744/nirmana.22.1.41-51.
- [17] A. Agung, A. Wiryya, R. I. Rokhmawati, and H. M. Az-zahra, “Perancangan dan Evaluasi User Experience Aplikasi Microteaching Pendukung Standar Kompetensi Mahasiswa Calon Guru berbasis Mobile dengan Pendekatan Human Centered Design dan Remote Usability Testing,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 1, pp. 402–410, 2022.
- [18] A. Segara, “Penerapan Pola Tata Letak (Layout Pattern) pada Wireframing Halaman Situs Web,” *J. Magenta, STMK Trisakti*, vol. 3, no. 1, pp. 452–464, 2019.
- [19] F. N. Hasanah and R. S. Untari, *Buku Ajar Rekayasa Perangkat Lunak*, 1st ed. Sidoarjo: UMSIDA Press, 2020