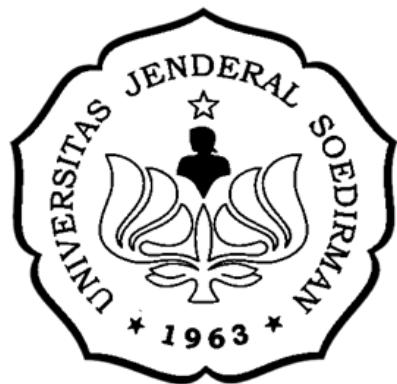


**RANCANG BANGUN SISTEM REGISTRASI *ONLINE* UNTUK PASIEN
UMUM/ASURANSI BERBASIS *WEBSITE* MENGGUNAKAN LARAVEL
DI RSU ST. ELISABETH PURWOKERTO**

LAPORAN KERJA PRAKTIK



Oleh:

BRIAN CAHYA PURNAMA

H1D022009

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS TEKNIK
JURUSAN INFORMATIKA
PURWOKERTO
2024**

ABSTRAK

Sistem registrasi *online* merupakan komponen penting dalam meningkatkan efisiensi pelayanan kesehatan modern. Laporan ini menjabarkan pengembangan sistem registrasi *online* untuk RSU St. Elisabeth Purwokerto menggunakan *framework* Laravel, dengan integrasi Filament untuk sistem *admin*. Metodologi pengembangan mengadaptasi prinsip Agile, khususnya Scrum, yang disesuaikan untuk pengerjaan individu. Sistem ini dirancang untuk mengakomodasi kebutuhan pasien umum dan asuransi, memungkinkan pemilihan poliklinik dan jadwal dokter secara mandiri. Proses pengembangan meliputi analisis kebutuhan, perancangan menggunakan diagram UML, implementasi kode, dan desain UI/UX melalui *wireframing* dan *prototyping*. Fitur-fitur utama mencakup manajemen poliklinik dan dokter, *generate booking code*, penyimpanan data registrasi, pembuatan bukti pendaftaran PDF, serta penampilan jadwal dan info kuota. Pengujian komprehensif menggunakan metode *Black Box* menunjukkan bahwa sistem memenuhi semua kriteria fungsional yang ditetapkan. Implementasi sistem ini diharapkan dapat meningkatkan efisiensi operasional rumah sakit dan kepuasan pasien.

Kata kunci: sistem registrasi *online*, Laravel, Filament, Agile, pelayanan kesehatan

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : BRIAN CAHYA PURNAMA

NIM : H1D022009

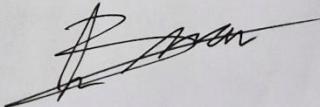
Menyatakan dengan sebenar-benarnya bahwa laporan kerja praktik saya yang berjudul:

**RANCANG BANGUN SISTEM REGISTRASI *ONLINE* UNTUK PASIEN
UMUM/ASURANSI BERBASIS *WEBSITE* MENGGUNAKAN LARAVEL
DI RSU ST. ELISABETH PURWOKERTO**

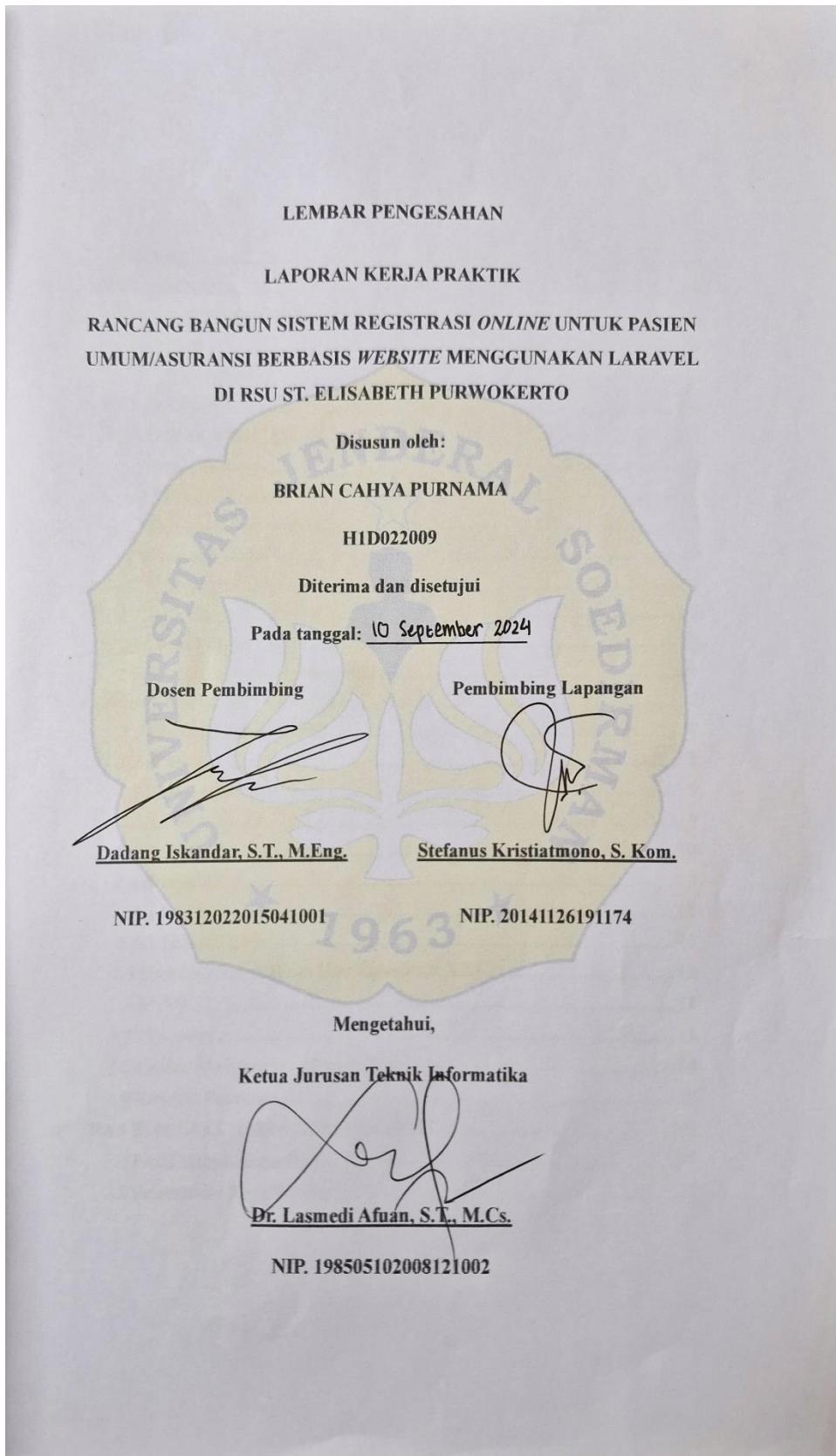
Adalah hasil karya sendiri dan bukan jiplakan hasil karya orang lain.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Jika dikemudian hari terbukti bahwa laporan kerja praktik saya merupakan hasil jiplakan, maka saya bersedia menerima sanksi apapun yang diberikan.

Purwokerto, 9 Agustus 2024



BRIAN CAHYA PURNAMA



DAFTAR ISI

ABSTRAK	i
PERNYATAAN	ii
LEMBAR PENGESAHAN	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Pertanyaan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Tujuan Kerja Praktik	4
1.6 Kegunaan Kerja Praktik	5
1.7 Tempat Kerja Praktik.....	6
1.8 Waktu Pelaksanaan Kerja Praktik.....	6
BAB II TINJAUAN PUSTAKA	7
2.1 Sistem Registrasi <i>Online</i>	7
2.2 <i>Website</i>	8
2.3 Bahasa Pemrograman	9
2.3.1 PHP	9
2.3.2 JavaScript.....	10
2.4 <i>Framework</i>	10
2.4.1 Laravel.....	11
2.4.2 Tailwind CSS.....	12
2.5 <i>User Interface</i> (UI) dan <i>User Experience</i> (UX)	12
2.6 MySQL	13
2.7 <i>Flowchart</i>	13
2.8 <i>Unified Modeling Language</i> (UML)	14
2.9 Keaslian Penelitian	14
BAB III PELAKSANAAN KERJA PRAKTIK	16
3.1 Profil Tempat Kerja Praktik.....	16
3.2 Pelaksanaan Kerja Praktik	17

3.2.1 Tahap Persiapan.....	17
3.2.2 Tahap Pelaksanaan.....	18
3.3 Metode Kerja Praktik	18
BAB IV PEMBAHASAN.....	21
4.1 <i>Product Backlog</i>	21
4.1.1 Identifikasi Pengguna	21
4.1.2 Kebutuhan Pengguna.....	21
4.1.3 Kebutuhan Sistem.....	22
4.2 <i>Sprint Planning</i>	25
4.2.1 <i>Flowchart</i>	25
4.2.2 <i>Use Case Diagram</i>	26
4.2.3 <i>Sequence Diagram</i>	27
4.2.4 <i>Class Diagram</i>	59
4.2.5 Desain UI/UX.....	59
4.3 <i>Sprint Backlog</i>	87
4.3.1 <i>Item Backlog</i>	88
4.3.2 Pembagian <i>Sprint</i>	91
4.4 <i>Sprint</i>	92
4.4.1 Pemilihan Teknologi.....	92
4.4.2 Rincian Pengembangan	93
4.4.3 <i>Package Filament</i>	110
4.5 Finished Work.....	113
4.5.1 Pengujian Sistem	113
4.5.2 Penyerahan <i>Source Code</i>	118
BAB V PENUTUP.....	117
5.1 Kesimpulan.....	117
5.2 Saran	117
DAFTAR PUSTAKA	118
LAMPIRAN 1 RINCIAN KERJA PRAKTIK.....	123
LAMPIRAN 2 SERTIFIKAT KELULUSAN.....	130
LAMPIRAN 3 SURAT PENERIMAAN INSTANSI	131
LAMPIRAN 4 PENILAIAN PELAKSANAAN KERJA PRAKTIK	132
LAMPIRAN 5 LEMBAR PRESENSI KERJA PRAKTIK	133
LAMPIRAN 6 CURRICULUM VITAE.....	135

DAFTAR GAMBAR

Gambar 1.	Diagram Alur Metode Scrum.....	18
Gambar 2.	<i>Flowchart</i> Sistem.....	26
Gambar 3.	<i>Use Case Diagram</i> Sistem	27
Gambar 4.	<i>Sequence Diagram</i> Registrasi	28
Gambar 5.	<i>Sequence Diagram</i> Jadwal Praktik Dokter	29
Gambar 6.	<i>Sequence Diagram</i> Info Kuota Besok	30
Gambar 7.	<i>Sequence Diagram</i> <i>Login Admin</i>	31
Gambar 8.	<i>Sequence Diagram</i> Lihat Data Registrasi	32
Gambar 9.	<i>Sequence Diagram</i> Ubah Status Data Registrasi	33
Gambar 10.	<i>Sequence Diagram</i> Hapus Data Registrasi	34
Gambar 11.	<i>Sequence Diagram</i> <i>Create Data Pasien</i>	35
Gambar 12.	<i>Sequence Diagram</i> Lihat Data Pasien.....	36
Gambar 13.	<i>Sequence Diagram</i> Ubah Data Pasien.....	37
Gambar 14.	<i>Sequence Diagram</i> Hapus Data Pasien	38
Gambar 15.	<i>Sequence Diagram</i> <i>Create Data Poliklinik</i>	39
Gambar 16.	<i>Sequence Diagram</i> Lihat Data Poliklinik	40
Gambar 17.	<i>Sequence Diagram</i> Ubah Data Poliklinik	41
Gambar 18.	<i>Sequence Diagram</i> Hapus Data Poliklinik.....	42
Gambar 19.	<i>Sequence Diagram</i> <i>Create Data Dokter</i>	43
Gambar 20.	<i>Sequence Diagram</i> Lihat Data Dokter	44
Gambar 21.	<i>Sequence Diagram</i> Ubah Data Dokter.....	45
Gambar 22.	<i>Sequence Diagram</i> Hapus Data Dokter	46
Gambar 23.	<i>Sequence Diagram</i> <i>Create Data Jadwal Dokter</i>	47
Gambar 24.	<i>Sequence Diagram</i> Lihat Data Jadwal Dokter	48
Gambar 25.	<i>Sequence Diagram</i> Ubah Data Jadwal Dokter	49
Gambar 26.	<i>Sequence Diagram</i> Hapus Data Jadwal Dokter	50
Gambar 27.	<i>Sequence Diagram</i> <i>Create Data Jadwal Khusus Dokter</i>	51
Gambar 28.	<i>Sequence Diagram</i> Lihat Data Jadwal Khusus Dokter	52
Gambar 29.	<i>Sequence Diagram</i> Ubah Data Jadwal Khusus Dokter	53
Gambar 30.	<i>Sequence Diagram</i> Hapus Data Jadwal Khusus Dokter	54
Gambar 31.	<i>Sequence Diagram</i> <i>Create Data Cuti Dokter</i>	55
Gambar 32.	<i>Sequence Diagram</i> Lihat Data Cuti Dokter	56
Gambar 33.	<i>Sequence Diagram</i> Ubah Data Cuti Dokter	57
Gambar 34.	<i>Sequence Diagram</i> Hapus Data Cuti Dokter	58
Gambar 35.	<i>Class Diagram</i> Sistem.....	59
Gambar 36.	<i>Wireframe</i> Halaman <i>Home/Beranda</i>	61
Gambar 37.	<i>Wireframe</i> Halaman Memilih Pasien Lama	62
Gambar 38.	<i>Wireframe</i> Halaman Memilih Pasien Baru	63
Gambar 39.	<i>Wireframe</i> Halaman Jadwal Praktik Dokter	64
Gambar 40.	<i>Wireframe</i> Halaman Info Kuota Besok.....	65
Gambar 41.	<i>Wireframe</i> Halaman Selesai Daftar.....	66
Gambar 42.	<i>Prototype</i> Halaman <i>Home/Beranda</i>	67

Gambar 43. <i>Prototype Halaman Memilih Pasien Lama</i>	68
Gambar 44. <i>Protoype Halaman Memilih Pasien Baru</i>	69
Gambar 45. <i>Prototype Halaman Jadwal Praktik Dokter</i>	70
Gambar 46. <i>Prototype Halaman Info Kuota Besok.....</i>	71
Gambar 47. <i>Prototype Halaman Selesai Daftar.....</i>	72
Gambar 48. Tampilan Halaman <i>Admin Login</i>	73
Gambar 49. Tampilan Halaman <i>Admin Dashboard.....</i>	73
Gambar 50. Tampilan Halaman <i>Admin Data Registrasi.....</i>	74
Gambar 51. Tampilan Halaman <i>Admin Detail Registrasi.....</i>	74
Gambar 52. Tampilan Halaman <i>Admin Ubah Registrasi.....</i>	75
Gambar 53. Tampilan Halaman <i>Admin Hapus Registrasi</i>	75
Gambar 54. Tampilan Halaman <i>Admin Data Pasien</i>	76
Gambar 55. Tampilan Halaman <i>Admin Detail Pasien</i>	76
Gambar 56. Tampilan Halaman <i>Admin Create Pasien</i>	77
Gambar 57. Tampilan Halaman <i>Admin Ubah Pasien</i>	77
Gambar 58. Tampilan Halaman <i>Admin Hapus Pasien</i>	77
Gambar 59. Tampilan Halaman <i>Admin Data Jadwal Dokter.....</i>	78
Gambar 60. Tampilan Halaman <i>Admin Create Jadwal Dokter.....</i>	78
Gambar 61. Tampilan Halaman <i>Admin Ubah Jadwal Dokter.....</i>	79
Gambar 62. Tampilan Halaman <i>Admin Hapus Jadwal Dokter</i>	79
Gambar 63. Tampilan Halaman <i>Admin Data Jadwal Khusus Dokter.....</i>	80
Gambar 64. Tampilan Halaman <i>Admin Create Jadwal Khusus Dokter.....</i>	80
Gambar 65. Tampilan Halaman <i>Admin Ubah Jadwal Khusus Dokter.....</i>	81
Gambar 66. Tampilan Halaman <i>Admin Hapus Jadwal Khusus Dokter</i>	81
Gambar 67. Tampilan Halaman <i>Admin Data Dokter.....</i>	82
Gambar 68. Tampilan Halaman <i>Admin Create Dokter.....</i>	82
Gambar 69. Tampilan Halaman <i>Admin Ubah Dokter.....</i>	83
Gambar 70. Tampilan Halaman <i>Admin Hapus Dokter</i>	83
Gambar 71. Tampilan Halaman <i>Admin Data Cuti Dokter</i>	84
Gambar 72. Tampilan Halaman <i>Admin Create Cuti Dokter.....</i>	84
Gambar 73. Tampilan Halaman <i>Admin Ubah Cuti Dokter</i>	85
Gambar 74. Tampilan Halaman <i>Admin Hapus Cuti Dokter</i>	85
Gambar 75. Tampilan Halaman <i>Admin Data Poliklinik</i>	86
Gambar 76. Tampilan Halaman <i>Admin Create Poliklinik</i>	86
Gambar 77. Tampilan Halaman <i>Admin Ubah Poliklinik</i>	87
Gambar 78. Tampilan Halaman <i>Admin Hapus Poliklinik</i>	87
Gambar 79. Bukti Pendaftaran PDF	104

DAFTAR TABEL

Tabel 1. Identifikasi Pengguna	21
Tabel 2. Kebutuhan <i>Admin</i>	22
Tabel 3. Kebutuhan Pasien	22
Tabel 4. Kebutuhan Sistem.....	23
Tabel 5. <i>Item Backlog</i>	88
Tabel 6. Pengujian Sistem	114

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem registrasi pasien merupakan komponen penting dalam operasional rumah sakit, yang berperan penting dalam meningkatkan kepuasan pasien dan mengoptimalkan alur kerja staf (Rismawan & Renaningtyas, 2024). Perkembangan teknologi informasi telah membawa inovasi signifikan dalam bidang pelayanan kesehatan, termasuk sistem registrasi *online* untuk pasien (Susilo et al., 2023). Sistem ini tidak hanya mempermudah proses pendaftaran, tetapi juga meningkatkan efisiensi manajemen data pasien. RSU St. Elisabeth Purwokerto, dalam upayanya meningkatkan kualitas pelayanan, menyadari perlunya pembaruan sistem registrasi yang dapat mengakomodasi kebutuhan pasien umum maupun pasien asuransi secara lebih efektif.

Pengembangan sistem registrasi *online* berbasis *website* merupakan solusi yang tepat untuk memenuhi kebutuhan tersebut. Dengan memanfaatkan *framework* Laravel, sistem ini dapat dibangun dengan struktur yang kuat dan fitur-fitur modern (Sinlae et al., 2024). Laravel menawarkan berbagai keunggulan, seperti *syntax* yang elegan, sistem migrasi *database* yang handal, dan dukungan terhadap pola arsitektur MVC (*Model-View-Controller*). Selain itu, penggunaan Tailwind CSS memungkinkan pengembangan antarmuka yang responsif dan estetis, sementara Alpine.js menyediakan interaktivitas yang ringan pada sisi klien.

Salah satu fitur unggulan dari sistem registrasi *online* ini adalah kemampuan pasien untuk memilih poliklinik dan jadwal dokter secara mandiri. Pasien dapat dengan mudah melihat daftar poliklinik yang tersedia di RSU St. Elisabeth Purwokerto, beserta informasi dokter yang bertugas

dan jadwal praktiknya. Fitur ini memungkinkan pasien untuk merencanakan kunjungan mereka dengan lebih baik, memilih waktu yang paling sesuai dengan kebutuhan mereka, tanpa harus datang langsung ke rumah sakit untuk melakukan pendaftaran. Dengan demikian, sistem ini tidak hanya menghemat waktu dan tenaga pasien, tetapi juga mengurangi kepadatan antrean di loket pendaftaran rumah sakit, sehingga meningkatkan efisiensi pelayanan secara keseluruhan.

Untuk memastikan pengelolaan data yang efisien, sistem *admin* dirancang menggunakan Filament, sebuah paket *admin panel* untuk Laravel yang memanfaatkan Livewire. Filament memungkinkan pembuatan antarmuka *admin* yang kuat dan mudah digunakan, dengan dukungan untuk operasi CRUD (*Create, Read, Update, Delete*) yang *seamless/mulus*. Integrasi Livewire dalam Filament juga memungkinkan pembaruan komponen secara *real-time* tanpa perlu *me-refresh* halaman *admin* secara keseluruhan, meningkatkan responsivitas dan pengalaman pengguna (Restaldo & Beeh, 2022).

Dalam proses perancangan sistem, penggunaan diagram alur (*flowchart*) dan UML (*Unified Modeling Language*) sangat penting untuk memvisualisasikan dan mendokumentasikan berbagai aspek sistem (P & Abdillah, 2024). Diagram *use case* membantu dalam mengidentifikasi interaksi antara pengguna dan sistem, sementara diagram *sequence* menggambarkan urutan interaksi objek dalam skenario tertentu. Diagram kelas, di sisi lain, memberikan gambaran struktur statis sistem dengan menunjukkan kelas-kelas, atribut-atribut, dan hubungan antar kelas.

Dengan kombinasi teknologi modern dan perancangan yang matang, sistem registrasi *online* ini diharapkan dapat meningkatkan efisiensi operasional RSU St. Elisabeth Purwokerto. Sistem ini tidak hanya akan memudahkan proses pendaftaran bagi pasien, tetapi juga membantu staf rumah sakit dalam mengelola data pasien dengan lebih baik. Selain itu, integrasi sistem untuk pasien umum dan asuransi dalam satu platform akan

menyederhanakan proses administrasi dan meningkatkan akurasi data. Melalui implementasi sistem ini, RSU St. Elisabeth Purwokerto dapat meningkatkan kualitas pelayanannya dan memberikan pengalaman yang lebih baik bagi pasien.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah disampaikan di atas, terdapat beberapa rumusan masalah, diantaranya adalah:

- a. RSU St. Elisabeth Purwokerto membutuhkan pembaruan sistem registrasi yang dapat mengakomodasi kebutuhan pasien umum dan pasien asuransi secara lebih efektif.
- b. Perlu adanya sistem yang memungkinkan pasien memilih poliklinik dan jadwal dokter secara mandiri untuk meningkatkan efisiensi dan kepuasan pasien.
- c. Perlu adanya sistem yang memungkinkan *admin* memanajemen data pasien dan rumah sakit yang lebih efisien untuk meningkatkan kualitas pelayanan.

1.3 Pertanyaan Penelitian

Berdasarkan rumusan masalah yang sudah disampaikan di atas, terdapat beberapa pertanyaan penelitian, diantaranya adalah:

- a. Bagaimana cara merancang dan membangun sistem registrasi *online* yang dapat membantu pasien umum/asuransi untuk dapat melakukan pendaftaran secara mandiri di RSU St. Elisabeth Purwokerto dengan menggunakan *framework* Laravel?
- b. Bagaimana cara merancang dan membangun sistem *admin* yang efisien menggunakan Filament dan Livewire untuk pengelolaan data pasien dan manajemen sistem registrasi *online*?

1.4 Batasan Masalah

Agar pelaksanaan kerja praktik lebih terarah dan tidak menyimpang atau melebar dari pokok rumusan masalah yang sudah ditentukan, maka

diterapkan beberapa batasan masalah. Batasan masalah dalam merancang dan membangun sistem registrasi *online* untuk pasien umum/asuransi di RSU St. Elisabeth Purwokerto di antaranya adalah:

- a. Sistem registrasi *online* ini dikembangkan khusus untuk RSU St. Elisabeth Purwokerto.
- b. Pengembangan sistem menggunakan *framework* Laravel sebagai basis utama.
- c. Antarmuka pengguna dikembangkan menggunakan Tailwind CSS dan Alpine.js untuk responsivitas dan interaktivitas.
- d. Sistem *admin* dirancang menggunakan Filament dengan integrasi Livewire untuk pengelolaan data.
- e. Sistem hanya mencakup proses registrasi *online* untuk pasien umum dan asuransi, tidak termasuk proses lain seperti untuk pasien BPJS dan pembayaran atau rekam medis.
- f. Fitur pemilihan poliklinik dan jadwal dokter dibatasi pada informasi yang tersedia di RSU St. Elisabeth Purwokerto.
- g. Perancangan sistem menggunakan diagram alur (*flowchart*) dan UML (*Unified Modeling Language*) terbatas pada *use case*, *sequence*, dan *class* diagram.
- h. Integrasi akhir dengan sistem yang ada di rumah sakit akan dilakukan oleh tim IT rumah sakit, bukan bagian dari proyek kerja praktik ini.
- i. Pengujian sistem dilakukan dalam lingkungan pengembangan, terbatas pada fungsionalitas dan *user experience*, tidak termasuk pengujian keamanan atau *stress testing*, dan belum mencakup implementasi langsung di infrastruktur rumah sakit.

1.5 Tujuan Kerja Praktik

Tujuan - tujuan yang ingin dicapai dari kegiatan kerja praktik di RSU St. Elisabeth Purwokerto antara lain adalah:

- a. Memenuhi persyaratan kelulusan sebagai Sarjana Informatika, Universitas Jenderal Soedirman.
- b. Merancang dan membangun sistem registrasi *online* berbasis *website* untuk pasien umum dan asuransi di RSU St. Elisabeth Purwokerto dengan menggunakan *framework* Laravel.
- c. Merancang dan mengimplementasikan sistem *admin* menggunakan Filament dan Livewire untuk pengelolaan data pasien dan manajemen sistem registrasi *online* secara efisien.

1.6 Kegunaan Kerja Praktik

Pelaksanaan kerja praktik dan penulisan laporan kerja praktik ini memiliki beberapa kegunaan, yakni:

- a. Bagi Mahasiswa
 - Melaksanakan mata kuliah wajib yaitu Kerja Praktik.
 - Memenuhi beban SKS yang harus diambil sebagai persyaratan untuk mengambil Tugas Akhir.
 - Menambah pengalaman dan wawasan mahasiswa dalam mengaplikasikan ilmu informatika pada lingkungan kerja dunia nyata, khususnya dalam konteks sistem registrasi *online* rumah sakit.
- b. Bagi Instansi/Perusahaan
 - Menjalin hubungan yang baik antara pihak Universitas Jenderal Soedirman dengan RSU St. Elisabeth Purwokerto.
 - Mempermudah proses registrasi mandiri pasien umum/asuransi RSU St. Elisabeth Purwokerto melalui implementasi sistem registrasi *online* yang efisien dan *user-friendly*.
 - Mempermudah staf/*admin* rumah sakit dalam mengelola data pasien dan jadwal kunjungan melalui sistem *admin* yang terintegrasi.

c. Bagi Universitas

- Membangun dan memperkuat kerjasama dengan industri, khususnya dalam sektor kesehatan.
- Sebagai umpan balik dari industri mengenai kualitas dan relevansi kurikulum pendidikan yang diterapkan.
- Meningkatkan reputasi universitas melalui kontribusi nyata mahasiswa dalam proyek-proyek di dunia kerja.

1.7 Tempat Kerja Praktik

Kerja praktik ini dilaksanakan di Rumah Sakit Umum Saint Elisabeth Purwokerto, yakni di Teknologi Informasi. Lokasi kantor berada di Ruang IT di sebelah gedung baru RSU St. Elisabeth Purwokerto yang beralamatkan Jl. Dr. Angka No. 40, Purwokerto. Kontak yang bisa dihubungi yaitu Kepegawaian RSE (0895-3233-34836).

1.8 Waktu Pelaksanaan Kerja Praktik

Kerja praktik berlangsung selama 1 bulan. Periode Kerja Praktik dilaksanakan pada 10 Juli sampai dengan 9 Agustus 2024 dengan jam kerja dimulai dari jam 7 pagi hingga jam 2 siang. Ketentuan tambahan mengenai waktu pelaksanaan kerja praktik adalah sebagai berikut:

- a. Batas wajar keterlambatan pada saat berangkat adalah 5 menit.
- b. Jika ingin berangkat siang, peserta kerja praktik dapat menghubungi divisi kepegawaian terlebih dahulu.
- c. Total jam kerja harus mencapai 7 jam per hari.
- d. Waktu istirahat yang diberikan adalah 30 menit, kecuali pada hari Jumat.

Ketentuan-ketentuan ini bertujuan untuk memastikan kedisiplinan dan produktivitas selama pelaksanaan kerja praktik, sembari tetap memberikan fleksibilitas yang wajar kepada peserta.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Registrasi *Online*

Sistem adalah sekumpulan elemen yang saling terkait dan terorganisir yang bekerja bersama untuk mencapai tujuan tertentu (Febriani et al., 2020). Dalam konteks teknologi informasi, sistem merujuk pada kombinasi perangkat keras, perangkat lunak, data, prosedur, dan sumber daya manusia yang berinteraksi untuk mengolah *input* menjadi *output* yang bermanfaat. Sistem dapat bervariasi dalam kompleksitasnya, mulai dari yang sederhana hingga yang sangat rumit, namun semua sistem memiliki tujuan, komponen yang saling terhubung, dan batas yang jelas.

Registrasi adalah proses pencatatan atau pendaftaran informasi ke dalam suatu sistem pencatatan. Dalam konteks layanan atau organisasi, registrasi melibatkan pengumpulan dan penyimpanan data pribadi atau informasi penting lainnya dari individu atau entitas. Tujuan registrasi dapat beragam, seperti untuk keanggotaan, partisipasi dalam suatu acara, akses ke layanan tertentu, atau pemenuhan persyaratan administratif. Proses registrasi tradisional sering melibatkan pengisian formulir fisik, namun dengan perkembangan teknologi, metode registrasi telah berkembang menjadi lebih efisien dan praktis (Novianti Putri et al., 2024).

Sistem registrasi *online* adalah penerapan teknologi informasi untuk mengotomatisasi dan memfasilitasi proses registrasi melalui internet (Nabyla & Sigitta, 2019). Sistem ini menggabungkan konsep sistem dan registrasi ke dalam platform digital yang memungkinkan pengguna untuk mendaftar atau memberikan informasi tanpa perlu hadir secara fisik (Suci Widya Primadhani, Yaslis Ilyas, 2021). Komponen utama sistem registrasi online meliputi antarmuka pengguna berbasis web, *database* untuk menyimpan informasi, dan sistem *back-end* untuk memproses data. Manfaat

utama sistem ini termasuk efisiensi waktu, pengurangan kesalahan input, aksesibilitas yang dapat diakses setiap saat, dan kemudahan dalam pengelolaan data. Namun, implementasinya juga menghadapi tantangan seperti kebutuhan infrastruktur teknologi yang memadai, keamanan data, dan desain antarmuka yang ramah pengguna.

2.2 *Website*

Website atau situs web, adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet menggunakan *browser* web (Noviantoro et al., 2022). Web atau lengkapnya www (*world wide web*) adalah sebuah koleksi keterhubungan dokumen-dokumen yang disimpan diinternet dan diakses menggunakan *protocol* (*Hyper Text Transfer Protocol*) (WULANDARI, 2020). *Website* terdiri dari berbagai elemen seperti teks, gambar, video, dan file multimedia lainnya yang disusun menggunakan bahasa markup seperti HTML (*Hypertext Markup Language*). Tampilan dan tata letak website umumnya diatur menggunakan CSS (*Cascading Style Sheets*), sementara interaktivitas dan fungsionalitas dinamis sering ditambahkan menggunakan bahasa pemrograman seperti JavaScript.

Website dapat dikategorikan berdasarkan tujuan dan fungsinya, seperti situs informasi, *e-commerce*, media sosial, atau aplikasi web. Dalam pengembangan modern, *website* tidak lagi terbatas pada tampilan statis, tetapi telah berkembang menjadi platform dinamis dan interaktif yang dapat menyesuaikan konten berdasarkan input pengguna atau data *real-time*. *Responsive web design* telah menjadi standar industri, memungkinkan website untuk beradaptasi dengan berbagai ukuran layar dan perangkat, dari *desktop* hingga *smartphone* (Aprilia et al., 2024). *Website* dapat diakses baik menggunakan internet jika tersimpan di server dan dapat diakses secara *offline* jika tersimpan di jaringan *local*.

2.3 Bahasa Pemrograman

Bahasa pemrograman adalah sistem notasi formal yang digunakan untuk menulis instruksi yang dapat dieksekusi oleh komputer. Ini merupakan alat fundamental dalam pengembangan perangkat lunak, memungkinkan *programmer* untuk menentukan dengan tepat tindakan yang harus dilakukan oleh komputer. Bahasa pemrograman terdiri dari sintaks (aturan tentang bagaimana simbol dan kata-kata dapat disusun) dan semantik (makna dari susunan tersebut), yang bersama-sama mendefinisikan struktur dan makna program komputer. Bahasa-bahasa ini dapat dikategorikan menjadi beberapa jenis, seperti bahasa tingkat rendah (misalnya bahasa mesin dan assembly) yang lebih dekat dengan instruksi *hardware*, dan bahasa tingkat tinggi (seperti Python, Java, atau C++) yang lebih mudah dibaca dan ditulis oleh manusia (Geni et al., 2023). Dalam konteks pengembangan web, bahasa seperti PHP dan JavaScript memainkan peran penting, di mana PHP umumnya digunakan untuk pemrograman sisi server, sementara JavaScript digunakan untuk meningkatkan interaktivitas pada sisi klien. Pada pengembangan sistem registrasi *online* ini menggunakan bahasa pemrograman PHP dan JavaScript.

2.3.1 PHP

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman sisi server yang dirancang khusus untuk pengembangan web. diciptakan oleh Rasmus Lerdorf pada tahun 1994, PHP telah berkembang menjadi salah satu bahasa pemrograman web paling populer dan banyak digunakan (Purba & Rahmat, 2014). PHP bersifat *open-source*, yang berarti dapat digunakan secara gratis dan memiliki komunitas pengembang yang besar dan aktif. Bahasa ini terutama digunakan untuk menghasilkan konten web dinamis, memproses formulir, mengelola sesi pengguna, dan berinteraksi dengan *database*. PHP dapat disisipkan langsung ke dalam kode HTML, memungkinkan pengembang untuk dengan mudah mencampur logika *server-side* dengan *markup* HTML.

Salah satu kekuatan utama PHP adalah kompatibilitasnya yang luas dengan berbagai sistem manajemen *database*, terutama MySQL. PHP juga mendukung beragam protokol internet dan menyediakan banyak fungsi bawaan untuk tugas-tugas umum dalam pengembangan web. Dengan evolusi bahasa ini melalui berbagai versi, PHP telah mengadopsi fitur-fitur pemrograman berorientasi objek, peningkatan performa, dan perbaikan keamanan. Meskipun menghadapi persaingan dari bahasa pemrograman web lainnya, PHP tetap menjadi pilihan populer karena kemudahan penggunaannya, dukungan *hosting* yang luas, dan ekosistem yang baik (Sugito & Shafira, 2023).

2.3.2 JavaScript

JavaScript adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan diproses di sisi *client*. JavaScript adalah bahasa pemrograman berbentuk kumpulan *script* yang berjalan pada suatu dokumen HTML (Sahi, 2020). JavaScript digunakan dalam pembuatan website agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. JavaScript dapat menyempurnakan tampilan dan sistem pada halaman *web-based application* yang dikembangkan. Adapun karakteristik dari bahasa pemrograman JavaScript yaitu, bahasa pemrograman berjenis *high-level programming*, bersifat *client-side*, berorientasi pada objek, dan bersifat *loosely typed*. Aplikasi atau *tools* standar yang digunakan dalam mengimplementasikan JavaScript adalah, *software text-editor* (Notepad++, Adobe Dreamweaver, dan NetBeans), *web browser* (Mozilla Firefox, Google Chrome, Internet Explorer, Opera Mini, Safari), dan HTML (Mariko, 2019).

2.4 Framework

Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan

sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau class dari awal (Ambriani & Nurhidayat, 2020). Ada banyak sekali jenis *framework* pemrograman, tetapi biasanya dibagi ke dalam dua kelas, yaitu untuk *backend* dan *frontend*. Contoh *framework* yang termasuk dalam kategori *framework frontend* adalah Bootstrap, React, Vue.js, Angular, Tailwind CSS, dan lain-lain. Untuk *framework* yang masuk dalam kategori *backend*, ada Laravel, CodeIgniter, Django, dan lain-lain. Dalam pengembangan sistem registrasi *online* ini, *framework* yang digunakan adalah *backend* yaitu Laravel dan untuk *frontend* atau tampilan menggunakan Tailwind CSS.

2.4.1 Laravel

Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework* Laravel. Laravel adalah *framework* berbasis PHP yang sifatnya *open source* dan menggunakan konsep MVC atau *model – view – controller* (Ambriani & Nurhidayat, 2020). Laravel berada di bawah lisensi MIT License dengan menggunakan Github sebagai tempat berbagi kode. Dalam penggunaannya, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library* Laravel dan PHP minimal versi 8.2 untuk menjalankan Laravel versi 11.

Laravel adalah *framework* PHP yang dirilis sejak Juni 2011 dan dirancang untuk mengembangkan aplikasi *website* dengan cara yang lebih terstruktur dan lebih rapi. Laravel adalah sebuah *framework* PHP yang menekankan pada 27 kesederhanaan dan fleksibilitas pada desainnya. Laravel diluncurkan dibawah lisensi MIT dengan menggunakan Github sebagai tempat berbagi kode. Sama halnya dengan *framework* PHP lainnya, Laravel dikembangkan dengan basis MVC (*Model – View – Controller*). Laravel menyediakan fitur *database seeder* yang memungkinkan pengembang untuk mengisi *database* dengan data uji atau data awal secara otomatis. *Database seeder* ini sangat berguna dalam tahap pengembangan dan pengujian aplikasi, memungkinkan pengembang untuk dengan cepat

mengatur lingkungan *database* yang konsisten. Selain itu, Laravel dilengkapi *command line tool* yang bernama “Artisan” yang dapat digunakan untuk *packaging bundle* dan instalasi *bundle* (Muthia Kansha, 2023).

2.4.2 Tailwind CSS

Tailwind CSS adalah *framework* CSS utility-first yang menyediakan berbagai kelas utilitas yang bisa langsung digunakan untuk mengatur tampilan elemen HTML. Tailwind memungkinkan developer untuk mengatur desain langsung di markup HTML tanpa perlu menulis CSS custom (H. Saputra et al., 2024). Dirilis pada tahun 2017, Tailwind CSS menawarkan pendekatan yang berbeda dalam *styling web* dibandingkan dengan *framework* CSS tradisional. Framework ini menggunakan sistem konfigurasi yang fleksibel, memungkinkan penyesuaian berbagai aspek seperti palet warna, ukuran *font*, dan *breakpoint* responsif. Tailwind CSS juga menerapkan teknik "purge" untuk efisiensi, menghasilkan file CSS yang kecil. Meskipun memiliki kurva pembelajaran awal, Tailwind CSS populer karena mempercepat pengembangan *front-end*, menawarkan konsistensi desain, dan fleksibilitas dalam menciptakan antarmuka yang unik dan responsif. *Framework* ini cocok untuk berbagai skala proyek dan sering digunakan bersama *framework* JavaScript modern seperti Alpine.js, React, atau Vue.

2.5 User Interface (UI) dan User Experience (UX)

User Interface (UI) merujuk pada tampilan visual dan interaktif dari sebuah produk digital yang memungkinkan pengguna untuk berinteraksi dengan sistem. UI mencakup elemen-elemen seperti tombol, menu, ikon, dan tata letak yang dirancang untuk memudahkan interaksi pengguna dengan produk (D. Saputra et al., 2024). Desain UI yang efektif harus memenuhi prinsip-prinsip seperti konsistensi, visibilitas status sistem, dan pencegahan kesalahan (Kumoro & Hasanah, 2020).

Sementara itu, *User Experience (UX)* memiliki cakupan yang lebih luas, meliputi keseluruhan pengalaman dan perasaan pengguna saat berinteraksi dengan produk atau sistem (Wiwesa, 2021). UX sebagai semua aspek interaksi pengguna akhir dengan perusahaan, layanan, dan produknya (Fitriana et al., 2021). UX yang baik tidak hanya tentang memberikan fungsi yang pengguna butuhkan, tetapi juga tentang menciptakan pengalaman yang menyenangkan, mudah, dan memuaskan.

2.6 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang populer, dikenal karena kecepatan, keandalan, dan fleksibilitasnya dalam mengelola data terstruktur (Sugito & Shafira, 2023). Dikembangkan oleh Oracle Corporation, MySQL mendukung bahasa *query* standar SQL dan dapat digunakan pada berbagai platform, menjadikannya pilihan utama untuk banyak aplikasi web dan sistem manajemen konten (Aqilah et al., 2023). Dalam konteks pengembangan sistem registrasi *online*, MySQL menawarkan solusi yang ideal untuk menyimpan dan mengelola data pasien, jadwal dokter, dan informasi pendaftaran dengan aman dan efisien.

2.7 Flowchart

Flowchart atau diagram alur adalah representasi grafis dari sebuah algoritma atau proses, menunjukkan langkah-langkah dalam bentuk kotak berbagai jenis yang dihubungkan dengan panah (Surojudin, 2019). *Flowchart* sangat efektif dalam menggambarkan alur logika suatu program atau sistem, memudahkan pemahaman dan komunikasi antara pengembang, pengguna, dan pemangku kepentingan lainnya (Prasetyo & Chernovita, 2023). Dalam konteks sistem registrasi *online* rumah sakit, *flowchart* dapat digunakan untuk menggambarkan alur pendaftaran pasien, proses pemilihan jadwal dokter, dan langkah-langkah verifikasi data, memastikan bahwa setiap aspek sistem terencana dengan baik dan efisien.

2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual standar yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak-artefak sistem perangkat lunak. UML mencakup berbagai jenis diagram, seperti *use case* diagram untuk menggambarkan interaksi antara sistem dan aktor, *class* diagram untuk merepresentasikan struktur statis sistem, dan *sequence* diagram untuk mengilustrasikan interaksi dinamis antar objek (Friadi et al., 2023). Dalam konteks pengembangan sistem registrasi *online* rumah sakit, UML dapat digunakan untuk memodelkan alur pendaftaran pasien, struktur data pasien dan dokter, serta interaksi antara berbagai komponen sistem, memastikan bahwa semua aspek sistem terdokumentasi dengan baik dan dapat dipahami oleh semua pihak yang terlibat dalam pengembangan.

2.9 Keaslian Penelitian

Rahman et al. (2022) mengembangkan sistem informasi registrasi pasien berbasis web untuk Rumah Sakit Harum Sisma Medika. Tujuannya adalah mengatasi masalah antrian dan meningkatkan efisiensi pendaftaran pasien. Sistem ini memungkinkan pasien baru dan lama melakukan registrasi *online*, melihat jadwal dokter, serta mendapatkan kode *booking* dan nomor antrian. Peneliti menggunakan metode *System Development Life Cycle* (SDLC) dengan pendekatan *prototype*. Teknologi yang digunakan meliputi PHP, JavaScript, dan MySQL, dengan pemodelan sistem menggunakan *Unified Modeling Language* (UML). Pengujian dilakukan dengan metode *Black Box testing*, dan hasilnya menunjukkan bahwa sistem dapat mempercepat proses pendaftaran dan mengurangi antrian di rumah sakit.

Fitriani & Voutama (2024) merancang sistem registrasi pasien berbasis web untuk Rumah Sakit Hermina Bekasi menggunakan UML dan metode SDLC. Tujuannya adalah meningkatkan efisiensi layanan dengan mengatasi masalah antrian panjang dan kompleksitas registrasi manual.

Sistem dirancang menggunakan diagram UML seperti *use case*, *activity*, dan *class diagram*. Implementasi melibatkan pengembangan antarmuka untuk *admin* dan pasien menggunakan teknologi web. Pengujian dilakukan dengan metode *blackbox testing*, dan hasil penelitian menunjukkan peningkatan aksesibilitas dan efisiensi proses registrasi pasien.

Yuli Fitriyah (2021) melakukan perancangan dan evaluasi aplikasi sistem informasi registrasi *online* (SIREGOL) berbasis *website* untuk fasilitas pelayanan kesehatan. Peneliti menggunakan metode *action research* untuk mengidentifikasi kebutuhan, merancang sistem, dan membuat prototipe. Teknologi yang digunakan meliputi *system flow*, *menu design*, *use case diagram*, *activity diagram*, *sequence diagram*, *entity relationship diagram* (ERD), dan *wireframe*. Evaluasi usability dilakukan menggunakan metode *System Usability Scale* (SUS), dengan hasil menunjukkan skor 75, mengindikasikan aplikasi dapat diterima dengan baik oleh pengguna dan masuk dalam kategori "Good".

Penelitian yang telah dilakukan mengembangkan sistem registrasi *online* untuk RSU St. Elisabeth Purwokerto dengan beberapa perbedaan dan penambahan dibandingkan penelitian sebelumnya. Pertama, penggunaan *framework* Laravel memberikan struktur yang lebih kuat dan fitur modern dalam pengembangan. Kedua, implementasi Tailwind CSS dan Alpine.js untuk antarmuka yang lebih responsif dan interaktif. Ketiga, penggunaan Filament untuk sistem *admin* yang lebih efisien dengan dukungan Livewire untuk pembaruan *real-time*. Keempat, sistem ini mencakup fitur khusus seperti pengelolaan jadwal khusus dokter, cuti dokter, informasi kuota yang lebih dinamis, dan integrasi QR *code* untuk verifikasi pendaftaran. Terakhir, pendekatan pengembangan mengadaptasi prinsip Agile-Scrum untuk fleksibilitas dan iterasi cepat dan pengujian menggunakan *Black Box testing*.

BAB III

PELAKSANAAN KERJA PRAKTIK

3.1 Profil Tempat Kerja Praktik

Pada awal mula RS St. Elisabeth didirikan berupa Klinik Bersalin kemudian berkembang menjadi Rumah Sakit Khusus Ibu dan Anak dan kini menjadi Rumah Sakit Umum. Mengingat bahwa dengan semakin berkembangnya teknologi kedokteran dan jenis penyakit yang ditangani menuntut Rumah Sakit Umum St. Elisabeth untuk dapat meyediakan pelayanan medis yang lengkap dan memadai bagi masyarakat kota Purwokerto dan sekitarnya. Oleh karena itu Rumah Sakit Umum St. Elisabeth yang saat ini sudah beroperasional keseluruhannya berupaya memberikan pelayanan kepada masyarakat sesuai dengan ketentuan atau peraturan yang berlaku dalam bidang perumahsakitan. Secara umum saat ini Rumah Sakit Umum St Elisabeth memberikan pelayanan rawat inap umum dan anak-anak, rawat jalan umum, poli gigi dan spesialis serta pelayanan penunjang.

Visi dari Rumah Sakit Umum Santa Elisabeth Purwokerto adalah: Menjadi Rumah Sakit Pilihan untuk Setiap Lapisan Masyarakat Teristimewa yang Terlupakan. Rumah Sakit Umum Santa Elisabeth Purwokerto juga telah menetapkan misi sebagai berikut:

- a. Menyelenggarakan pelayanan kesehatan yang menyeluruh.
- b. Menyelenggarakan pelayanan kesehatan yang bermutu dan profesional.
- c. Mengutamakan keselamatan pasien, menjadikan pasien dan keluarganya sebagai pusat penyelenggaraan pelayanan kesehatan.
- d. Memberi peluang kepada karyawan untuk selalu mengubah diri agar lebih bermutu.

3.2 Pelaksanaan Kerja Praktik

Pelaksanaan Kerja Praktik adalah bagian tak terpisahkan dari struktur kurikulum di pendidikan tinggi yang memberikan kesempatan kepada mahasiswa untuk menerapkan pengetahuan teoritis mereka dalam situasi kerja nyata. Selama periode ini, mahasiswa aktif terlibat dalam aktivitas sehari-hari di lembaga atau perusahaan yang relevan, dibimbing oleh seorang supervisor, yang memungkinkan mereka untuk mengembangkan keterampilan praktis, memahami dinamika dunia industri, dan meningkatkan kemampuan beradaptasi. Selain itu, pelaksanaan kerja praktik juga memiliki peran penting dalam membentuk jaringan profesional, meningkatkan keterampilan interpersonal, dan membentuk sikap etika kerja, sehingga mempersiapkan mahasiswa untuk menghadapi tantangan di dunia kerja setelah menyelesaikan pendidikan tinggi.

3.2.1 Tahap Persiapan

- a. Mengajukan Surat Pengantar Kerja Praktik pada Sistem Informasi Kerja Praktik (SIKAP).
- b. Mengajukan izin kerja praktik kepada pihak RSU St. Elisabeth Purwokerto dengan membawa surat pengantar dari fakultas.
- c. Mendapat surat balasan dari pihak instansi berupa kesediaan menerima pelaksanaan kegiatan kerja praktik.
- d. Berdiskusi dengan calon pembimbing lapangan terkait topik dan teknis kerja praktik.
- e. Membuat proposal kerja praktik dengan topik yang telah disetujui dengan pihak instansi dan berkonsultasi dengan dosen pembimbing.
- f. Mengajukan permohonan kerja praktik pada Sistem Informasi Kerja Praktik (SIKAP) dengan melampirkan proposal dan surat penerimaan permohonan kegiatan kerja praktik dari instansi terkait.
- g. Komisi menyatakan kelayakan permohonan kerja praktik.
- h. Mendapatkan Surat Perintah Kerja Praktik beserta berkas-berkas pendukung lainnya dari fakultas.

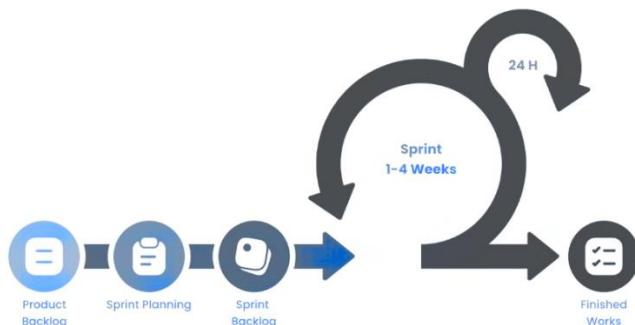
- i. Pelaksanaan kerja praktik berdasarkan kesepakatan dengan instansi terkait.

3.2.2 Tahap Pelaksanaan

Kerja praktik dilaksanakan selama 1 bulan dimulai sejak tanggal 10 Juli 2024 sampai 10 Agustus 2024. Waktu pelaksanaan kerja praktik mengikuti aturan jam operasional instansi yaitu setiap hari Senin sampai Sabtu pukul 07.00 WIB sampai pukul 14.10 WIB. Rincian kerja praktik dapat dilihat pada Lampiran 1 Rincian Kerja Praktik.

3.3 Metode Kerja Praktik

Dalam pelaksanaan kerja praktik ini, pengembangan sistem registrasi *online* untuk RSU St. Elisabeth Purwokerto dilakukan secara individu dengan mengadaptasi prinsip-prinsip metode Agile, khususnya kerangka kerja Scrum. Meskipun tidak bekerja dalam tim, pendekatan ini dipilih karena memungkinkan pengembangan yang fleksibel dan iteratif, dengan fokus pada prioritas fitur yang paling bernilai bagi sistem (Hayat et al., 2019). Metode ini sangat cocok untuk pengerjaan dalam jangka waktu singkat seperti kerja praktik 1 bulan ini, karena memungkinkan pencapaian hasil yang terukur dalam periode yang terbatas, serta memfasilitasi kolaborasi yang efektif dengan pembimbing lapangan.



Gambar 1. Diagram Alur Metode Scrum

Diagram alur metode kerja yang diadaptasi dalam kerja praktik ini dapat dilihat pada Gambar 1. Adaptasi metode Scrum dalam konteks kerja praktik individu ini melibatkan tahapan-tahapan berikut:

a. *Product Backlog*

Bersama dengan pembimbing lapangan, dibuat daftar fitur dan kebutuhan sistem yang diprioritaskan. Daftar ini mencakup semua fungsionalitas yang diinginkan untuk sistem registrasi *online* RSU St. Elisabeth, termasuk analisis kebutuhan pengguna dan kebutuhan sistem (Hanifudin et al., 2024).

b. *Sprint Planning*

Pada tahap ini, dilakukan perancangan menyeluruh untuk menerjemahkan kebutuhan yang telah diidentifikasi dalam *Product Backlog* menjadi desain sistem yang terstruktur (Nurzaman, 2020). Proses ini meliputi pembuatan berbagai diagram dan rancangan visual untuk memberikan gambaran komprehensif tentang sistem yang akan dikembangkan. Secara spesifik, tahap ini mencakup perancangan *flowchart*, *use case diagram*, *sequence diagram*, dan *class diagram*. Selain itu, juga dilakukan perancangan desain *wireframe* dan desain *prototype* untuk memvisualisasikan antarmuka pengguna dan alur interaksi sistem.

c. *Sprint Backlog*

Berdasarkan *sprint planning*, dibuat daftar tugas spesifik yang harus diselesaikan dalam *sprint* ini. Tugas-tugas ini diuraikan secara detail dengan estimasi waktu penyelesaian dan kriteria "selesai" yang jelas, sehingga memudahkan *tracking progres* selama *sprint* berlangsung (Hanifudin et al., 2024).

d. *Sprint*

Periode kerja aktif di mana pengembangan fitur-fitur yang telah direncanakan dilaksanakan. Selama *sprint*, dilakukan komunikasi rutin dengan pembimbing lapangan untuk *review* membahas progres dan hambatan. Proses pengembangan ini

mencakup implementasi *database* dan implementasi kode (Nurzaman, 2020).

e. *Finished Work*

Di akhir *sprint*, fitur-fitur yang telah selesai dikembangkan di-review bersama pembimbing lapangan. Pada tahap ini, dilakukan juga pengujian menyeluruh untuk memastikan bahwa fitur-fitur tersebut berfungsi dengan baik dan sesuai dengan kebutuhan yang telah ditentukan. Setelah proses pengembangan dan pengujian selesai, langkah terakhir adalah penyerahan *source code* kepada pihak IT RSU St. Elisabeth Purwokerto. Jika ada penyesuaian atau perbaikan yang diperlukan, hal tersebut dicatat dan dimasukkan ke dalam *product backlog* untuk ditindaklanjuti pada *sprint* berikutnya (Arnomo & Kurniawan, 2024).

BAB IV

PEMBAHASAN

4.1 Product Backlog

Product Backlog untuk sistem registrasi *online* RSU St. Elisabeth Purwokerto disusun berdasarkan hasil diskusi dan pengamatan dari sistem registrasi sebelumnya dengan pembimbing lapangan. Hal ini bertujuan untuk menjelaskan secara rinci dan menggambarkan sistem yang akan dibuat supaya sistem dapat memenuhi kebutuhan pengguna.

4.1.1 Identifikasi Pengguna

Berdasarkan hasil diskusi dan pengamatan yang telah dilakukan, sistem ini memiliki dua pengguna di antaranya:

Tabel 1. Identifikasi Pengguna

No	Pengguna	Peran
1	<i>Admin</i>	Melihat dan mengelola data – data dalam sistem secara keseluruhan.
2	Pasien	Melakukan registrasi mandiri dan melihat informasi data jadwal dokter.

4.1.2 Kebutuhan Pengguna

Setelah mengetahui calon pengguna yang akan menggunakan sistem, tahap selanjutnya adalah menentukan kebutuhan masing-masing pengguna. Tabel 2 dan Tabel 3 adalah hasil analisis kebutuhan pengguna berdasarkan hasil diskusi dan pengamatan yang telah dilakukan.

a. *Admin*

Tabel 2. Kebutuhan *Admin*

Kode	Kebutuhan
[URS-A-001]	<i>Admin</i> dapat melihat data registrasi pasien dan mengubah status registrasinya.
[URS-A-002]	<i>Admin</i> dapat melihat dan mengelola data pasien.
[URS-A-003]	<i>Admin</i> dapat melihat dan mengelola data jadwal dokter.
[URS-A-004]	<i>Admin</i> dapat melihat dan mengelola data jadwal khusus dokter.
[URS-A-005]	<i>Admin</i> dapat melihat dan mengelola data cuti dokter.
[URS-A-006]	<i>Admin</i> dapat melihat dan mengelola data poliklinik.
[URS-A-007]	<i>Admin</i> dapat melihat dan mengelola data dokter.

b. Pasien

Tabel 3. Kebutuhan Pasien

Kode	Kebutuhan
[URS-P-001]	Pasien dapat melakukan registrasi mandiri.
[URS-P-002]	Pasien dapat melihat data daftar poliklinik serta data jadwal dokternya yang tersedia.
[URS-P-003]	Pasien dapat melihat data informasi kuota yang tersisa untuk hari berikutnya.

4.1.3 Kebutuhan Sistem

Selanjutnya, dilakukan analisis untuk mengetahui kebutuhan sistem berdasarkan hasil analisis pada Tabel 2 dan Tabel 3. Hal ini bertujuan untuk merinci fitur-fitur yang ada pada sistem. Tabel 4 adalah hasil analisis kebutuhan sistem registrasi *online* RSU St. Elisabeth Purwokerto.

Tabel 4. Kebutuhan Sistem

Kode	Kebutuhan
[SRS-001]	Sistem menyediakan akses <i>login</i> untuk <i>admin</i> .
[SRS-002]	Ketika <i>admin login</i> dengan nama pengguna atau kata sandi tidak valid, sistem akan menampilkan pesan kesalahan.
[SRS-003]	Sistem dapat menampilkan halaman <i>dashboard</i> yang berisi <i>widget-widget</i> informatif yang menampilkan ringkasan data penting rumah sakit.
[SRS-004]	Sistem menyediakan fitur untuk lihat, hapus data registrasi, dan ubah status registrasi, serta <i>filter</i> data registrasi.
[SRS-005]	Sistem menyediakan fitur untuk tambah, ubah, hapus, lihat, dan <i>filter</i> data pasien.
[SRS-006]	Sistem menyediakan fitur untuk tambah, ubah, hapus, lihat, dan <i>filter</i> data dokter.
[SRS-007]	Sistem menyediakan fitur untuk tambah, ubah, hapus, lihat, dan <i>filter</i> data poliklinik.
[SRS-008]	Sistem menyediakan fitur untuk tambah banyak data jadwal dokter untuk satu dokter, ubah, hapus, lihat, dan <i>filter</i> data jadwal dokter.
[SRS-009]	Sistem menyediakan fitur untuk tambah banyak data jadwal khusus dokter untuk satu dokter, ubah, hapus, lihat, dan <i>filter</i> data jadwal khusus dokter.
[SRS-010]	Sistem menyediakan fitur untuk tambah, ubah, hapus, lihat, dan <i>filter</i> data cuti dokter.
[SRS-011]	Ketika <i>email</i> dan nomor kartu yang di- <i>inputkan</i> sudah terdaftar, sistem akan menampilkan peringatan.
[SRS-012]	Ketika jam selesai yang di- <i>inputkan</i> lebih awal dari jam mulai pada jadwal dokter dan jadwal khusus dokter, sistem akan menampilkan peringatan.
[SRS-013]	Ketika tanggal selesai yang di- <i>inputkan</i> lebih awal dari tanggal mulai pada cuti dokter, sistem akan menampilkan peringatan.
[SRS-014]	Sistem dapat menampilkan tabel informasi sisa kuota dokter untuk hari berikutnya.
[SRS-015]	Ketika tidak ada dokter yang tersedia, sistem akan menampilkan label pemberitahuan berwarna merah untuk poliklinik tersebut pada tabel jadwal praktik dan informasi sisa kuota.

[SRS-016]	Ketika sisa kuota seorang dokter habis, sistem akan menampilkan tulisan "Kuota Penuh" untuk dokter tersebut pada tabel informasi sisa kuota.
[SRS-017]	Ketika seluruh dokter dalam suatu poliklinik sisa kuotanya habis, sistem akan menampilkan label pemberitahuan berwarna kuning untuk poliklinik tersebut pada tabel informasi sisa kuota.
[SRS-018]	Ketika dokter mengambil cuti, sistem akan menghapus nama jadwal dokter tersebut dari tabel jadwal praktik dan informasi sisa kuota, serta pada pilihan opsi dokter pada <i>form</i> registrasi.
[SRS-019]	Ketika dokter memiliki jadwal khusus, sistem akan menggantikan jadwal reguler dengan jadwal khusus tersebut di tabel jadwal praktik, informasi sisa kuota, dan pilihan opsi dokter pada <i>form</i> registrasi.
[SRS-020]	Sistem dapat menampilkan tabel jadwal praktik dokter untuk satu minggu penuh, dari Senin hingga Minggu.
[SRS-021]	Sistem menyediakan <i>navigation bar</i> berupa logo, <i>home</i> , jadwal praktik, dan info kuota untuk pasien.
[SRS-022]	Sistem menyediakan <i>footer</i> yang memuat informasi dan media sosial rumah sakit.
[SRS-023]	Sistem menyediakan dua tombol untuk memilih antara pasien lama atau pasien baru.
[SRS-024]	Sistem menyediakan tiga <i>step progress bar</i> untuk menunjukkan tahapan proses pendaftaran <i>online</i> .
[SRS-025]	Ketika pasien memilih jenis pasien lama, sistem akan menampilkan <i>form</i> pasien lama.
[SRS-026]	Ketika pasien lama salah <i>input</i> data pada <i>form</i> pasien lama, sistem akan menampilkan pesan peringatan.
[SRS-027]	Ketika pasien lama benar <i>input</i> data pada <i>form</i> pasien lama, sistem akan menampilkan <i>form</i> data diri pasien lama tersebut dan registrasi.
[SRS-028]	Ketika pasien memilih jenis pasien baru, sistem akan menampilkan <i>form</i> data diri dan registrasi.
[SRS-029]	Sistem membatasi tanggal kunjungan selama tujuh hari ke depan pada <i>form</i> registrasi.
[SRS-030]	Ketika pasien mendaftar, sistem akan menghasilkan kode <i>booking</i> dan QR <i>code</i> serta menampilkan informasi pendaftaran.

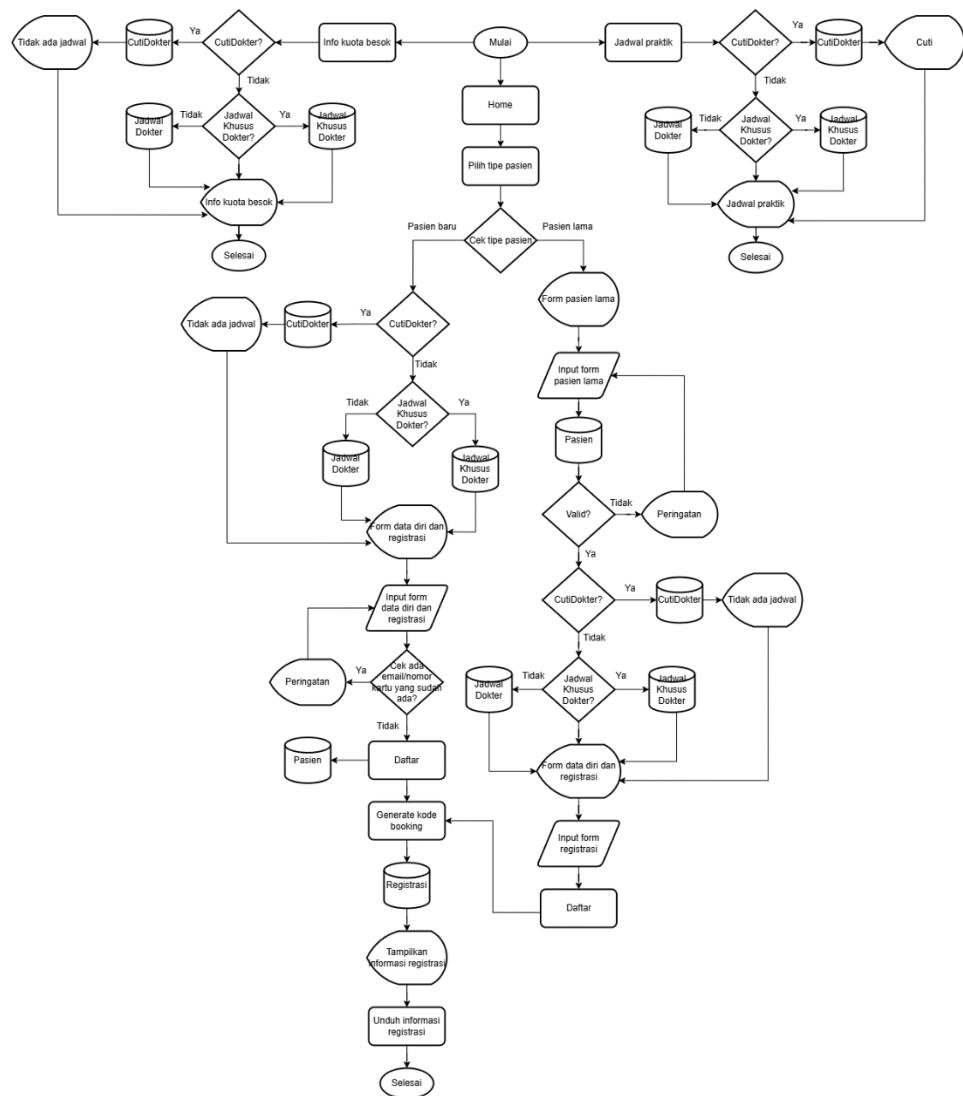
[SRS-031]	Sistem menyediakan tombol untuk mengunduh bukti pendaftaran yang dapat disimpan pada perangkat dalam bentuk berkas PDF yang di dalamnya terdapat informasi pendaftaran.
[SRS-032]	Sistem <i>responsive</i> ketika berada di resolusi layar <i>mobile</i> atau <i>handphone</i> .

4.2 Sprint Planning

Tahap *Sprint Planning* merupakan tahap yang penting dalam pengembangan sistem registrasi *online* RSU St. Elisabeth Purwokerto. Pada tahap ini, dilakukan perancangan menyeluruh untuk menerjemahkan kebutuhan yang telah diidentifikasi dalam *product backlog* menjadi desain sistem yang terstruktur. Proses ini meliputi pembuatan berbagai diagram dan rancangan visual untuk memberikan gambaran komprehensif tentang sistem yang akan dikembangkan.

4.2.1 Flowchart

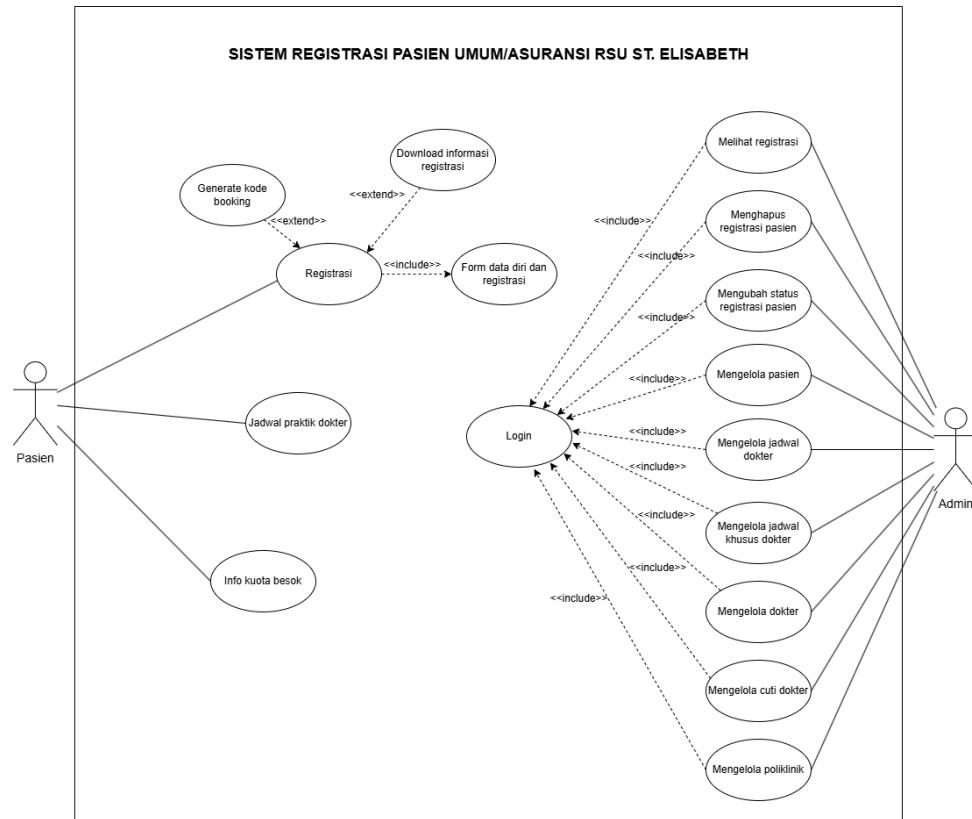
Flowchart atau diagram alur merupakan representasi visual dari alur proses pasien dalam sistem registrasi *online* RSU St. Elisabeth Purwokerto. Penggunaan *flowchart* membantu dalam memahami logika sistem secara keseluruhan. Adapun rancangan *flowchart* sistem ini dapat dilihat pada Gambar 2.



Gambar 2. Flowchart Sistem

4.2.2 Use Case Diagram

Use case diagram digunakan untuk menvisualisasikan interaksi antara pengguna (aktor) dengan sistem registrasi *online* RSU St. Elisabeth Purwokerto. Diagram ini menggambarkan fungsionalitas sistem dari perspektif pengguna, membantu dalam mengidentifikasi dan mendefinisikan kebutuhan fungsional sistem. Adapun rancangan *use case* diagram sistem ini dapat dilihat pada Gambar 3.

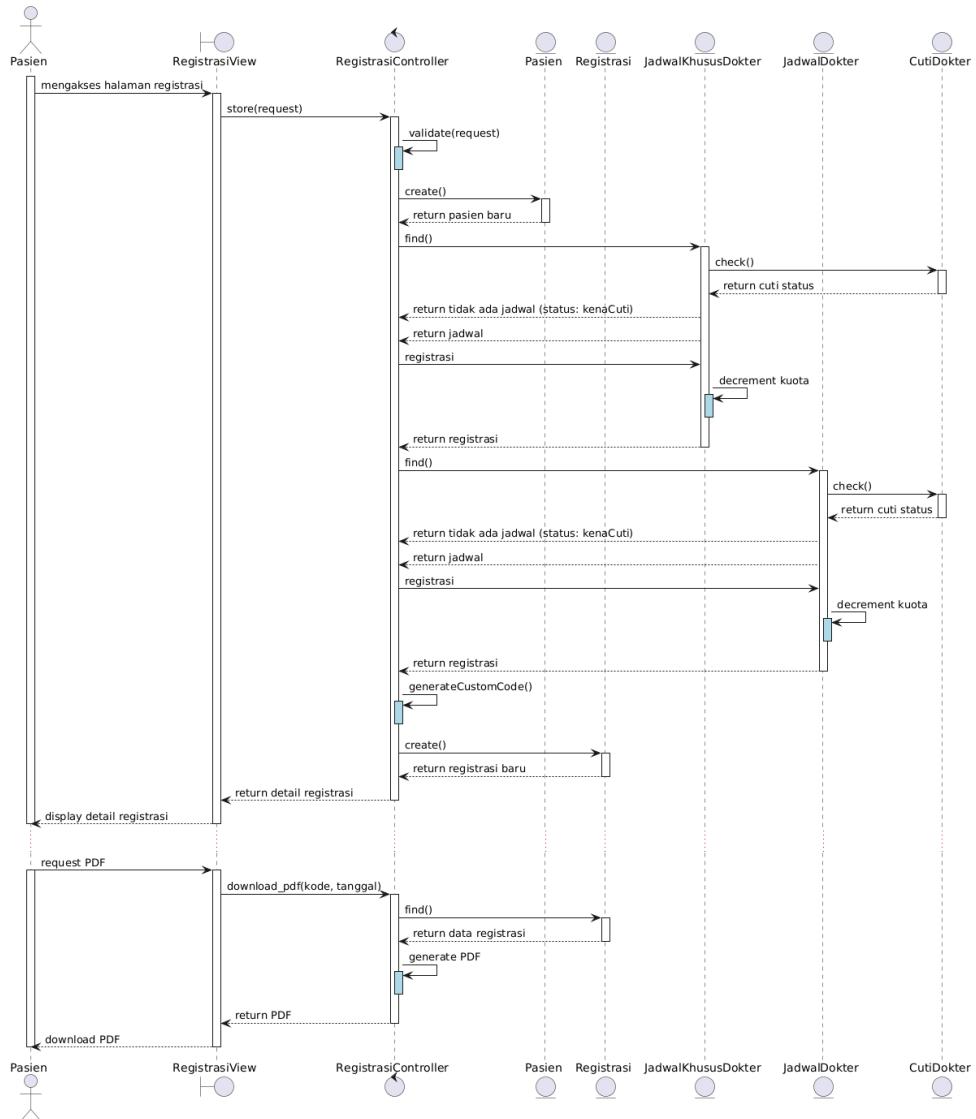


Gambar 3. Use Case Diagram Sistem

4.2.3 Sequence Diagram

Sequence diagram digunakan untuk mengilustrasikan interaksi antar objek dalam sistem registrasi *online* RSU St. Elisabeth Purwokerto secara kronologis. Diagram ini menunjukkan urutan pesan yang dikirim antar objek serta interaksi antar komponen sistem. *Sequence* diagram membantu dalam memahami alur kerja sistem secara lebih detail dan mengidentifikasi potensi masalah dalam alur komunikasi antar objek. Adapun rancangan *sequence* diagram sistem ini dapat dilihat pada Gambar 4 hingga Gambar 34.

- Registrasi

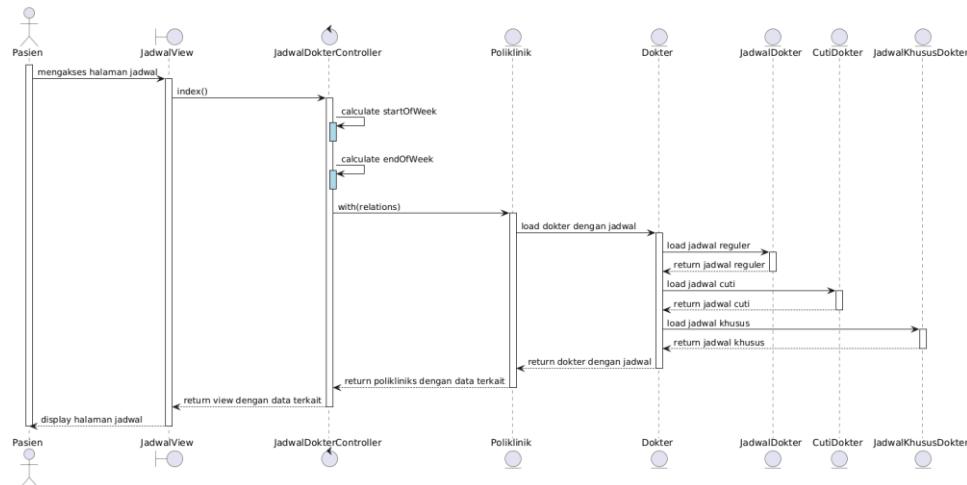


Gambar 4. Sequence Diagram Registrasi

Sequence diagram ini menggambarkan proses registrasi pasien. Alurnya dimulai dari pasien mengakses halaman registrasi, dilanjutkan dengan RegistrasiController memproses permintaan, melakukan validasi, dan membuat data pasien baru. Sistem kemudian memeriksa jadwal dan status cuti dokter, melakukan registrasi, mengurangi kuota, dan menghasilkan kode *custom*. Setelah registrasi selesai, detail ditampilkan ke

pasien. Diagram juga menunjukkan proses tambahan di mana pasien dapat meminta dan mengunduh PDF dari data registrasi.

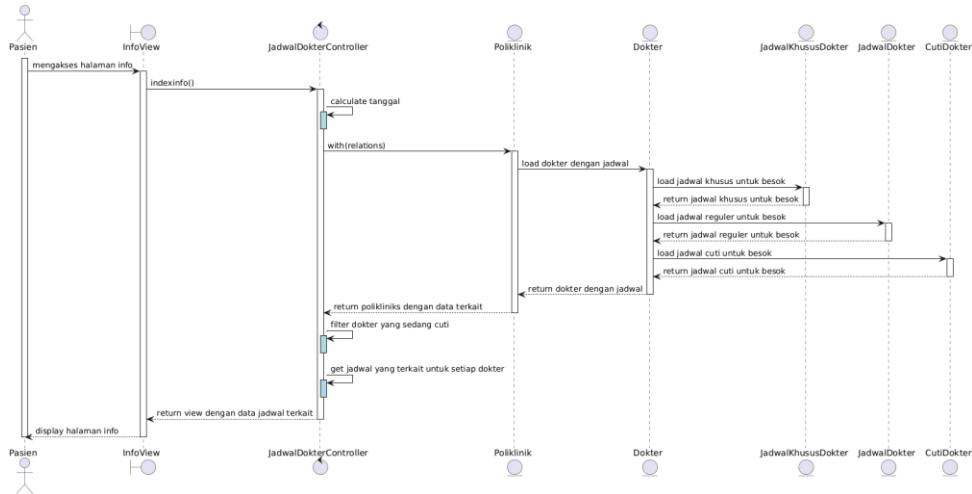
- Jadwal Praktik Dokter



Gambar 5. Sequence Diagram Jadwal Praktik Dokter

Sequence diagram ini menggambarkan proses pengaksesan jadwal dokter oleh pasien dalam sistem poliklinik. Alurnya dimulai dari pasien mengakses halaman jadwal, kemudian JadwalView memanggil JadwalDokterController. Controller ini menghitung awal dan akhir minggu, lalu mengambil data dari Poliklinik yang terhubung dengan Dokter. Sistem kemudian memuat jadwal reguler, jadwal cuti, dan jadwal khusus dokter dari model terkait (JadwalDokter, CutiDokter, JadwalKhususDokter). Setelah mengumpulkan semua informasi, data dikembalikan melalui controller ke view, yang akhirnya menampilkan halaman jadwal lengkap kepada pasien.

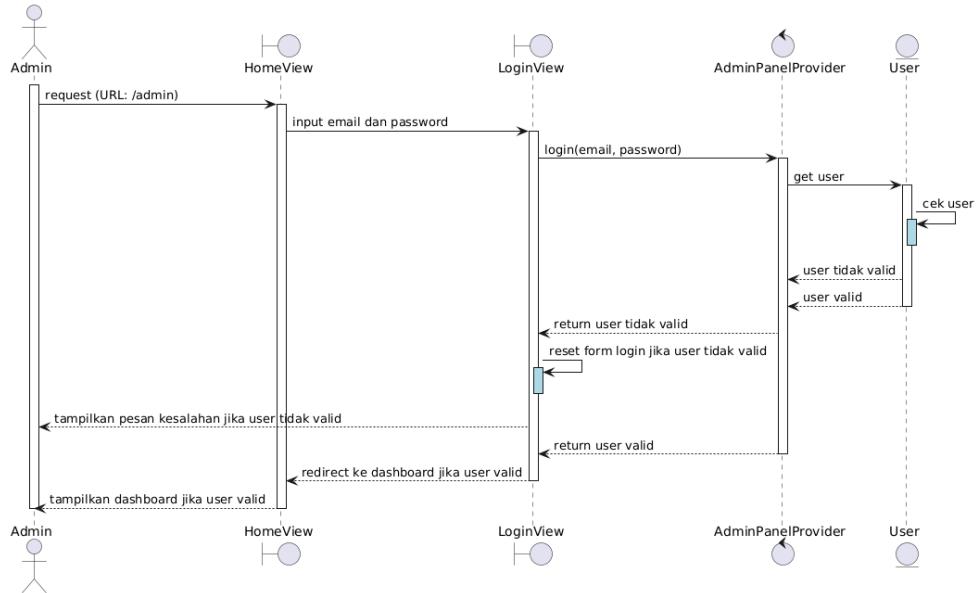
- Info Kuota Besok



Gambar 6. Sequence Diagram Info Kuota Besok

Sequence diagram ini mengilustrasikan proses pengaksesan informasi jadwal dokter oleh pasien. Alurnya dimulai dari pasien mengakses halaman info, kemudian InfoView memanggil JadwalDokterController. Controller ini menghitung tanggal besok dan mengambil data dari Poliklinik, termasuk informasi dokter beserta jadwalnya. Sistem memuat jadwal khusus, reguler, dan cuti dokter untuk keesokan harinya dari *model* terkait. Setelah mengumpulkan data, *controller* melakukan *filter* dokter yang sedang cuti dan mengatur jadwal terkait untuk setiap dokter. Akhirnya, *view* menampilkan halaman info dengan data jadwal yang telah diproses kepada pasien.

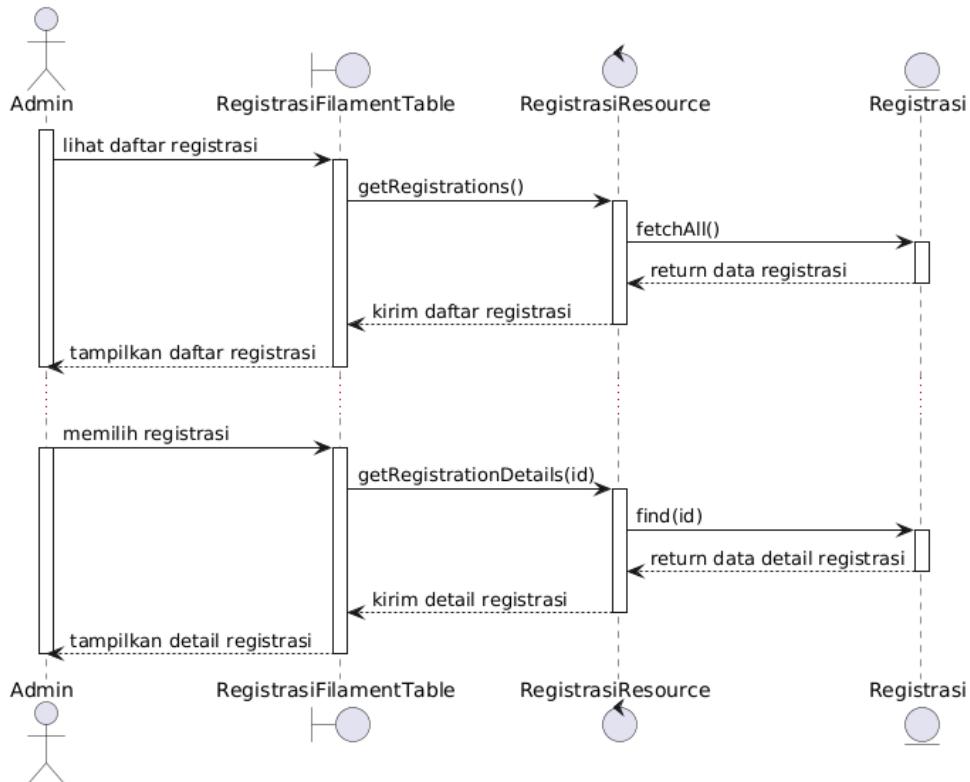
- Login Admin



Gambar 7. Sequence Diagram Login Admin

Sequence diagram ini menggambarkan proses *login admin* ke *panel administrasi*. Alurnya dimulai dari *Admin* mengakses URL *admin*, kemudian memasukkan *email* dan *password* di *HomeView*. Data *login* diteruskan ke *LoginView* yang meminta validasi ke *AdminPanelProvider* (Filament). *Provider* ini memeriksa kevalidan *user* melalui *model User*. Jika *user* tidak valid, sistem mengembalikan pesan *error*, mereset *form login*, dan menampilkan pesan kesalahan. Jika valid, sistem mengembalikan konfirmasi dan mengarahkan *admin* ke *dashboard*.

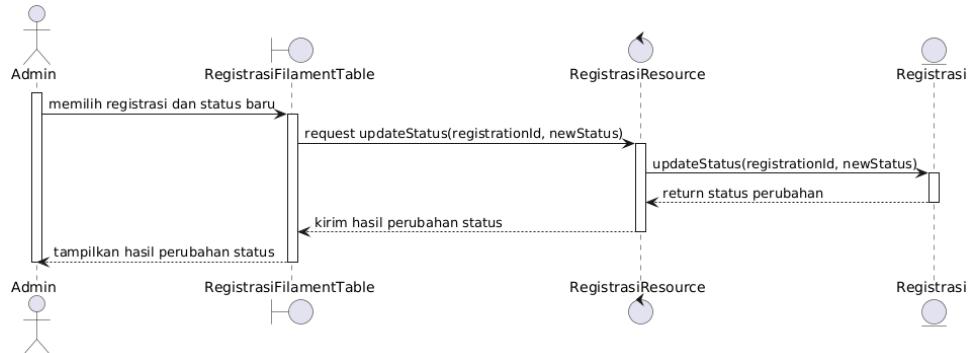
- Lihat Data Registrasi



Gambar 8. Sequence Diagram Lihat Data Registrasi

Sequence diagram ini menggambarkan proses melihat daftar registrasi dan detail registrasi oleh seorang *Admin*. Pertama, *Admin* meminta untuk melihat daftar registrasi, yang memicu RegistrasiFilamentTable untuk memanggil *getRegistrations()* pada RegistrasiResource. RegistrasiResource kemudian mengambil semua data registrasi dari *model Registrasi* menggunakan *fetchAll()*. Setelah daftar ditampilkan, *Admin* dapat memilih registrasi tertentu untuk melihat detailnya. Ini mengakibatkan pemanggilan *getRegistrationDetails(id)* pada RegistrasiFilamentTable, yang selanjutnya memanggil *find(id)* pada *model Registrasi* untuk mendapatkan data detail registrasi spesifik. Akhirnya, detail registrasi yang dipilih ditampilkan kepada *Admin*.

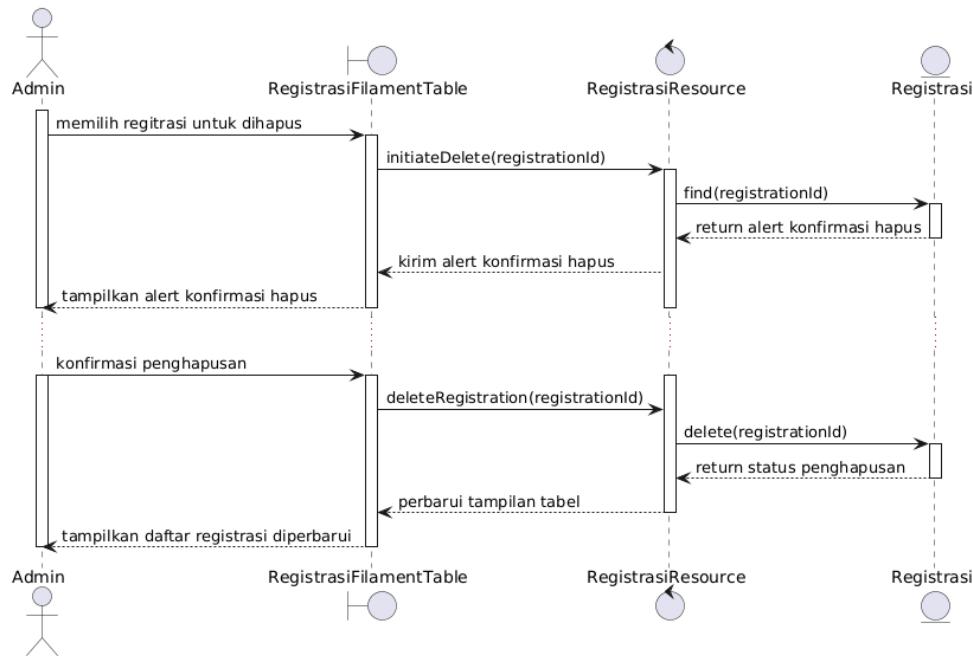
- Ubah Status Data Registrasi



Gambar 9. Sequence Diagram Ubah Status Data Registrasi

Sequence diagram ini mengilustrasikan proses perubahan status registrasi oleh *Admin*. Proses dimulai ketika *Admin* memilih registrasi tertentu dan status baru yang ingin ditetapkan. *RegistrasiFilamentTable* kemudian mengirim permintaan *updateStatus* ke *RegistrasiResource* dengan menyertakan ID registrasi dan status baru. *RegistrasiResource* meneruskan permintaan ini ke *model Registrasi* untuk melakukan pembaruan status. Setelah status diperbarui, hasilnya dikembalikan melalui *RegistrasiResource* ke *RegistrasiFilamentTable*, yang kemudian menampilkan hasil perubahan status kepada *Admin*.

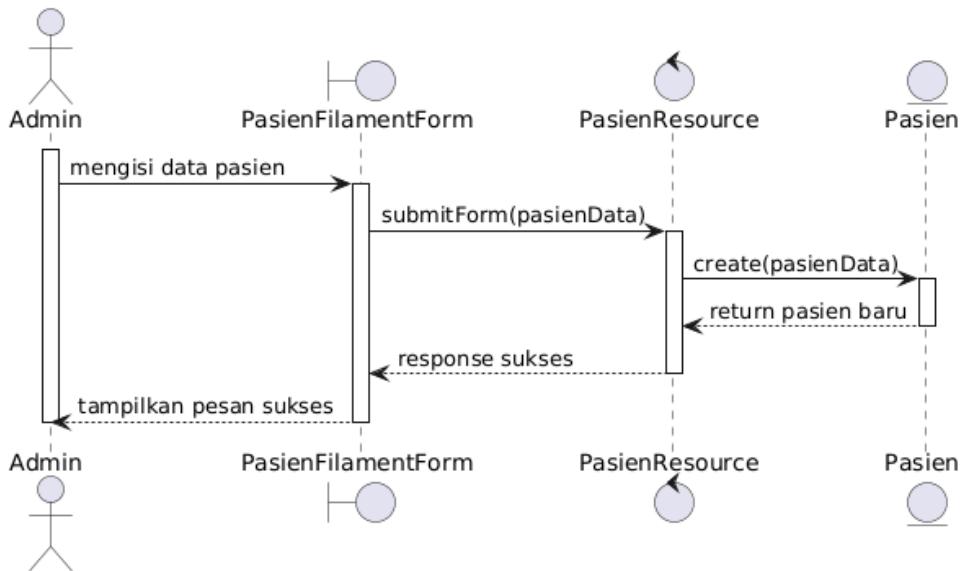
- Hapus Data Registrasi



Gambar 10. Sequence Diagram Hapus Data Registrasi

Sequence diagram ini menggambarkan proses penghapusan registrasi oleh *Admin*. Awalnya, *Admin* memilih registrasi untuk dihapus, memicu RegistrasiFilamentTable untuk menginisiasi proses penghapusan. Sistem mencari registrasi yang dipilih dan menampilkan *alert* konfirmasi kepada *Admin*. Jika *Admin* mengonfirmasi penghapusan, RegistrasiFilamentTable memanggil fungsi `deleteRegistration` pada RegistrasiResource, yang kemudian menghapus data dari *model* Registrasi. Setelah penghapusan berhasil, status penghapusan dikembalikan, dan RegistrasiFilamentTable memperbarui tampilan tabel, menampilkan daftar registrasi yang telah diperbarui kepada *Admin*.

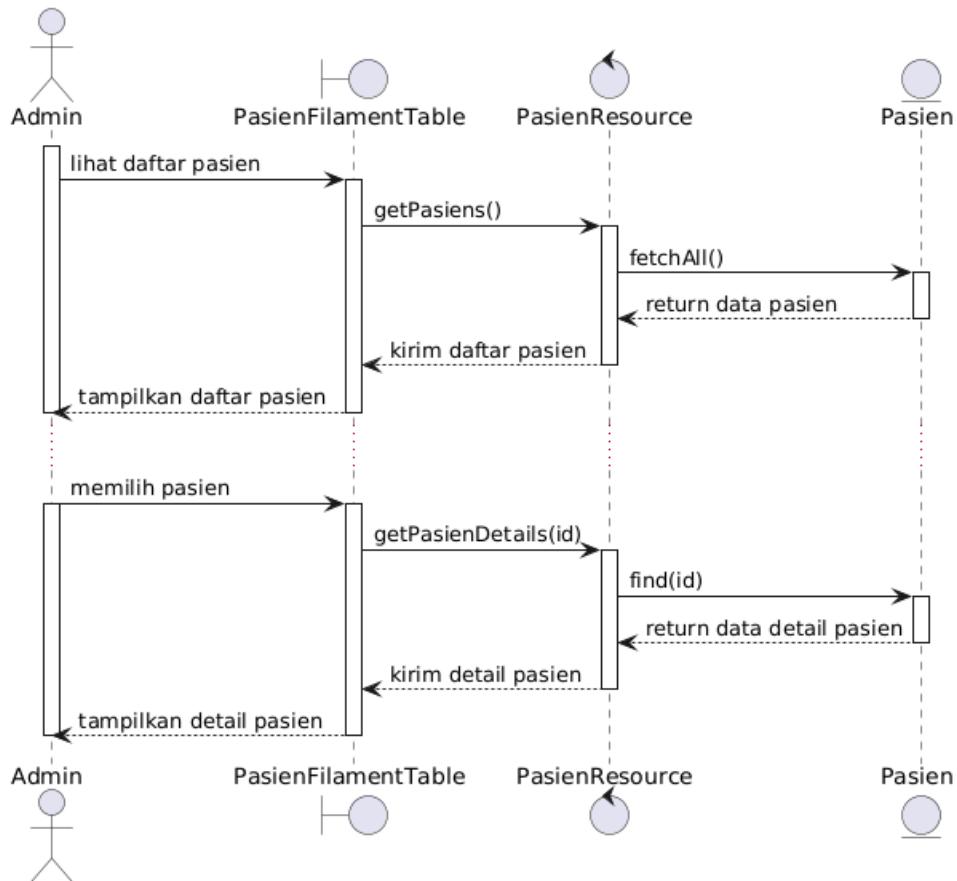
- *Create Data Pasien*



Gambar 11. Sequence Diagram *Create Data Pasien*

Sequence diagram ini mengilustrasikan proses penambahan data pasien baru oleh *Admin*. Proses dimulai ketika *Admin* mengisi data pasien melalui *PasienFilamentForm*. *Form* ini kemudian mengirimkan data (`submitForm`) ke *PasienResource*. *PasienResource* menggunakan metode `create` untuk menyimpan data pasien baru ke dalam *model* *Pasien*. Setelah data berhasil disimpan, *model* *Pasien* mengembalikan informasi pasien baru, yang diteruskan kembali ke *PasienFilamentForm* sebagai respons sukses. Akhirnya, *PasienFilamentForm* menampilkan pesan sukses kepada *Admin*, mengonfirmasi bahwa data pasien baru telah berhasil ditambahkan ke sistem.

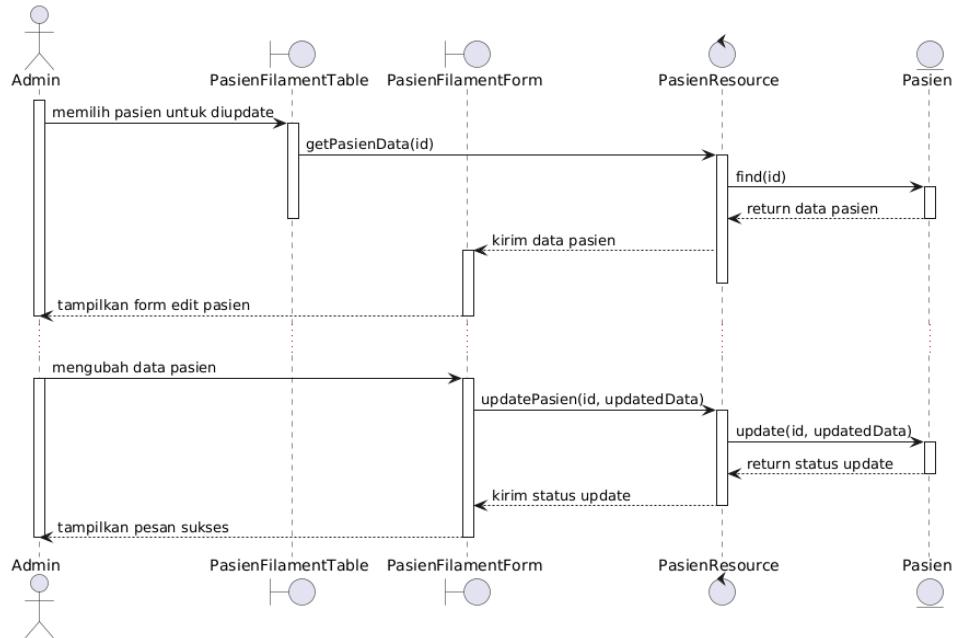
- Lihat Data Pasien



Gambar 12. Sequence Diagram Lihat Data Pasien

Sequence diagram ini menggambarkan proses melihat daftar pasien dan detail pasien oleh *Admin*. Pertama, *Admin* meminta untuk melihat daftar pasien, yang memicu *PasienFilamentTable* memanggil `getPasiens()` pada *PasienResource*. *PasienResource* kemudian mengambil semua data pasien dari *model* *Pasien* menggunakan `fetchAll()`. Setelah daftar ditampilkan, *Admin* dapat memilih pasien tertentu untuk melihat detailnya. Ini mengakibatkan pemanggilan `getPasienDetails(id)` pada *PasienFilamentTable*, yang selanjutnya memanggil `find(id)` pada *model* *Pasien* untuk mendapatkan data detail pasien spesifik. Akhirnya, detail pasien yang dipilih ditampilkan kepada *Admin*.

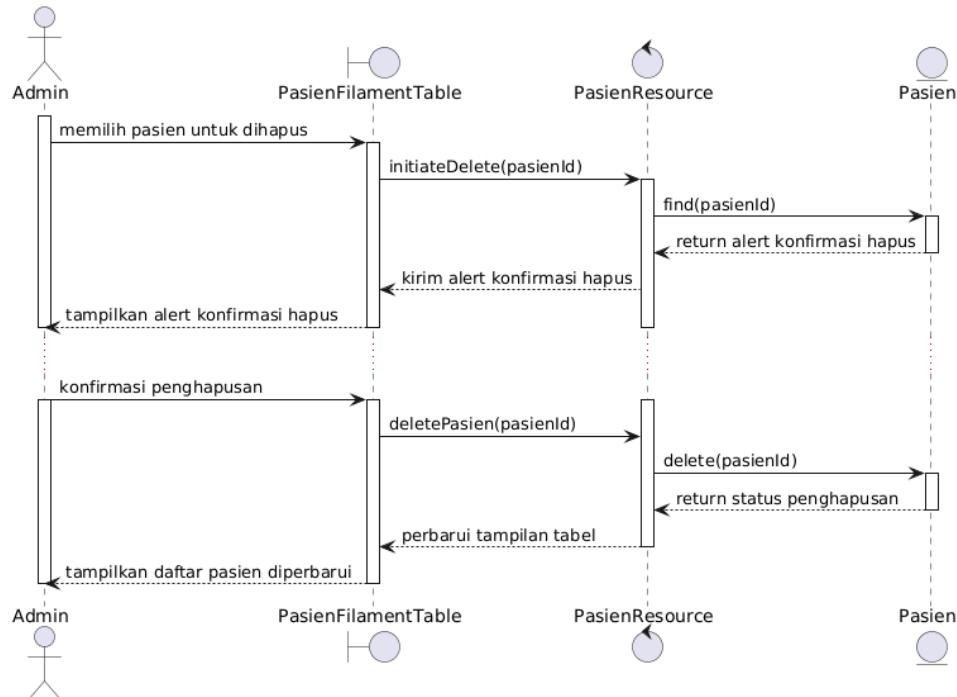
- Ubah Data Pasien



Gambar 13. Sequence Diagram Ubah Data Pasien

Sequence diagram ini menggambarkan proses pembaruan data pasien oleh *Admin*. Dimulai dengan *Admin* memilih pasien untuk *di-update*, *PasienFilamentTable* meminta data pasien dari *PasienResource*, yang mengambilnya dari *model Pasien*. Data ini kemudian ditampilkan dalam *form edit*. Setelah *Admin* mengubah data, *PasienFilamentForm* mengirim permintaan *update* ke *PasienResource*, yang meneruskannya ke *model Pasien*. *Pasien* memperbarui data dan mengembalikan status *update*, yang diteruskan kembali ke *PasienFilamentForm*. Akhirnya, pesan sukses ditampilkan kepada *Admin*, mengkonfirmasi bahwa pembaruan data pasien telah berhasil dilakukan.

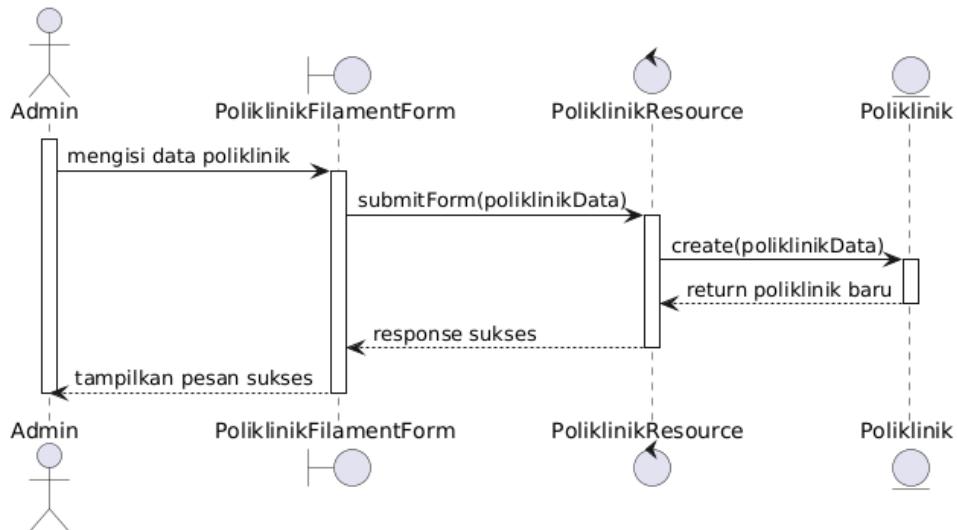
- Hapus Data Pasien



Gambar 14. Sequence Diagram Hapus Data Pasien

Sequence diagram ini mengilustrasikan proses penghapusan data pasien oleh *Admin*. Proses dimulai ketika *Admin* memilih pasien untuk dihapus, memicu *PasienFilamentTable* untuk menginisiasi proses penghapusan. Sistem mencari data pasien dan menampilkan *alert* konfirmasi kepada *Admin*. Jika *Admin* mengonfirmasi, *PasienFilamentTable* memanggil fungsi `deletePasien` pada *PasienResource*, yang kemudian menghapus data dari *model Pasien*. Setelah penghapusan berhasil, status penghapusan dikembalikan, dan *PasienFilamentTable* memperbarui tampilan, menampilkan daftar pasien yang telah diperbarui kepada *Admin*.

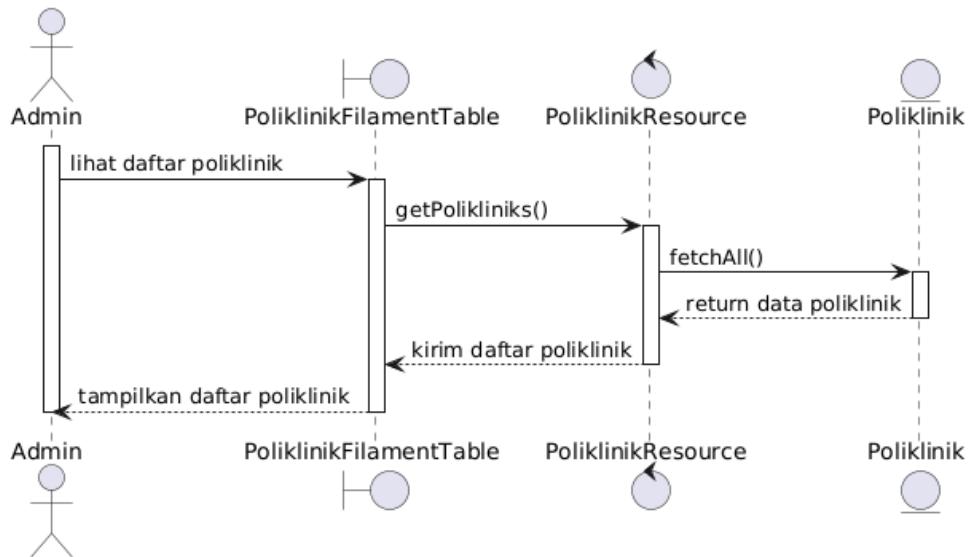
- *Create Data Poliklinik*



Gambar 15. Sequence Diagram *Create Data Poliklinik*

Sequence diagram ini menggambarkan proses penambahan data poliklinik baru oleh *Admin*. *Admin* memulai dengan mengisi data poliklinik melalui *PoliklinikFilamentForm*. *Form* ini kemudian mengirimkan data (*submitForm*) ke *PoliklinikResource*. *PoliklinikResource* menggunakan metode *create* untuk menyimpan data poliklinik baru ke dalam *model* *Poliklinik*. Setelah data berhasil disimpan, *model* *Poliklinik* mengembalikan informasi poliklinik baru, yang diteruskan kembali ke *PoliklinikFilamentForm* sebagai respons sukses. Akhirnya, *PoliklinikFilamentForm* menampilkan pesan sukses kepada *Admin*, mengonfirmasi bahwa data poliklinik baru telah berhasil ditambahkan ke sistem.

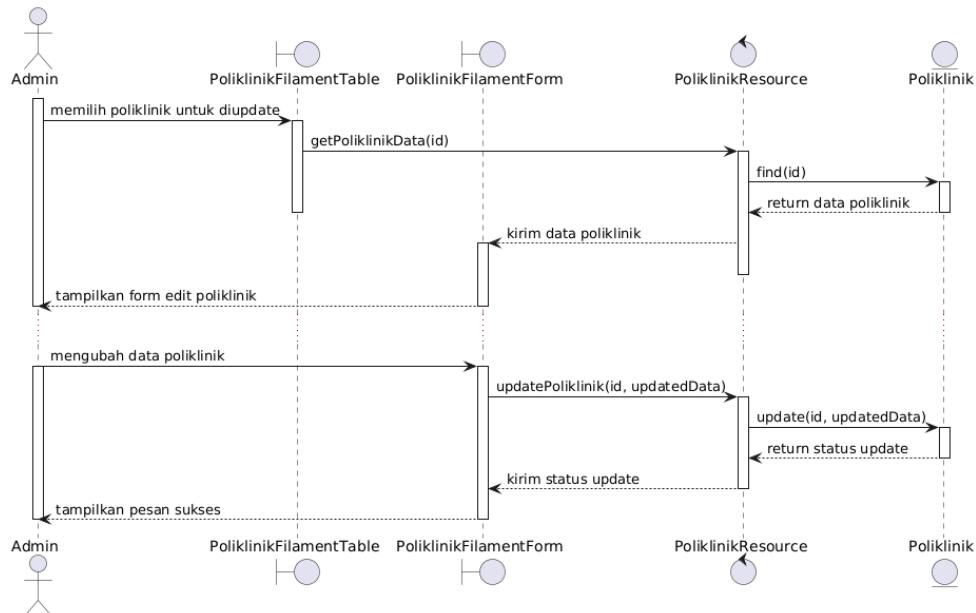
- Lihat Data Poliklinik



Gambar 16. Sequence Diagram Lihat Data Poliklinik

Sequence diagram ini menggambarkan proses melihat daftar poliklinik oleh *Admin*. Proses dimulai ketika *Admin* meminta untuk melihat daftar poliklinik. *PoliklinikFilamentTable* merespons dengan memanggil metode *getPolikliniks()* pada *PoliklinikResource*. *PoliklinikResource* kemudian menggunakan metode *fetchAll()* untuk mengambil semua data poliklinik dari *model Poliklinik*. Data yang diperoleh dikirimkan kembali melalui *PoliklinikResource* ke *PoliklinikFilamentTable*, yang akhirnya menampilkan daftar poliklinik kepada *Admin*.

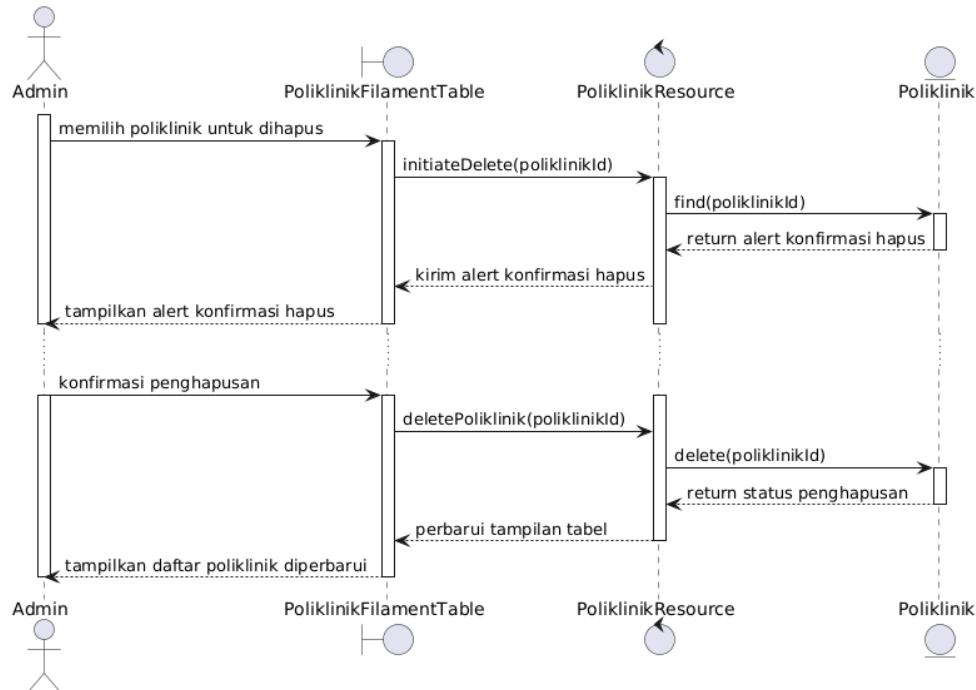
- Ubah Data Poliklinik



Gambar 17. Sequence Diagram Ubah Data Poliklinik

Sequence diagram ini menggambarkan proses pembaruan data poliklinik oleh *Admin*. Prosesnya dimulai dengan *Admin* memilih poliklinik untuk diperbarui, kemudian sistem mengambil data poliklinik tersebut dari *database*. *Form edit* poliklinik ditampilkan, dan *Admin* mengubah data yang diperlukan. Setelah perubahan dilakukan, sistem memperbarui data di *database* melalui *PoliklinikResource*. Status pembaruan dikirim kembali ke *form*, dan akhirnya sistem menampilkan pesan sukses kepada *Admin*, menandakan bahwa proses pembaruan data poliklinik telah berhasil dilakukan.

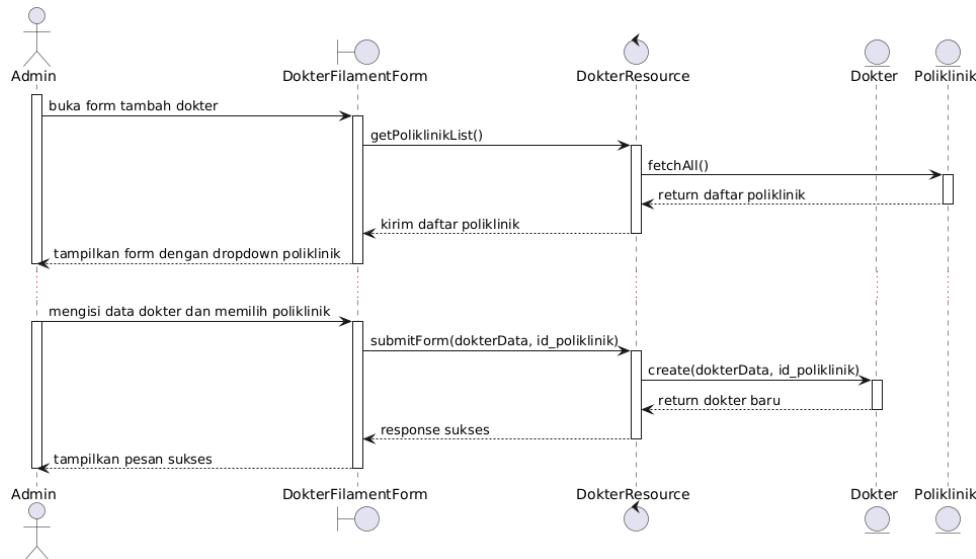
- Hapus Data Poliklinik



Gambar 18. Sequence Diagram Hapus Data Poliklinik

Sequence diagram ini mengilustrasikan proses penghapusan data poliklinik oleh *Admin*. Prosesnya dimulai dengan *Admin* memilih poliklinik yang akan dihapus. Sistem kemudian menginisiasi proses penghapusan dengan mencari data poliklinik tersebut dan menampilkan *alert* konfirmasi kepada *Admin*. Setelah *Admin* mengkonfirmasi penghapusan, sistem melakukan penghapusan data dari *database* melalui *PoliklinikResource*. Setelah penghapusan berhasil, sistem memperbarui tampilan tabel dan menampilkan daftar poliklinik yang telah diperbarui kepada *Admin*, menandakan bahwa proses penghapusan telah selesai dilakukan.

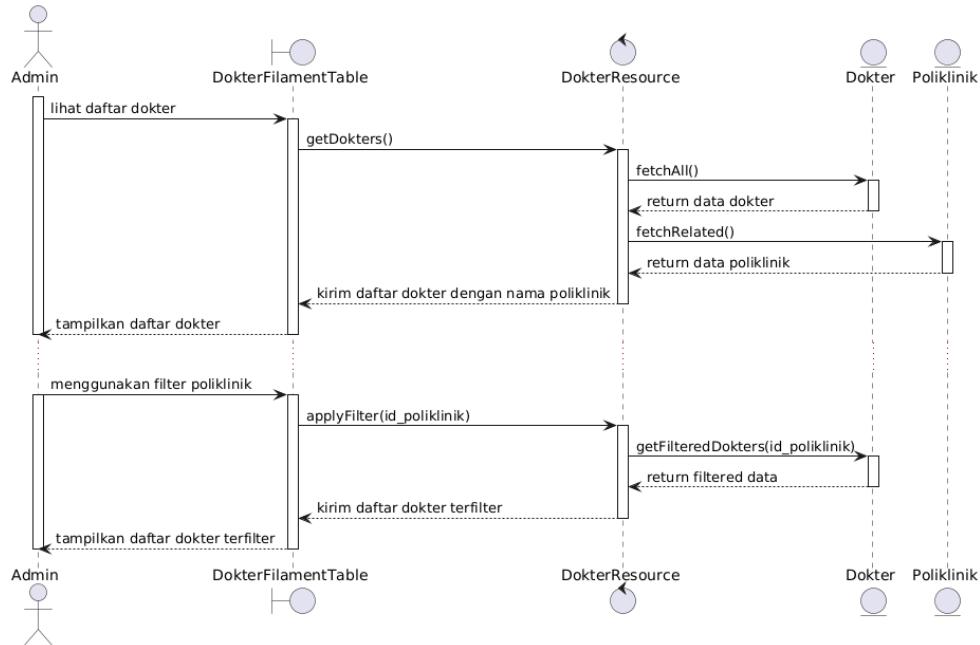
- *Create Data Dokter*



Gambar 19. Sequence Diagram *Create Data Dokter*

Sequence diagram ini menggambarkan proses penambahan dokter baru oleh *Admin*. Proses dimulai dengan *Admin* membuka *form* tambah dokter. Sistem kemudian mengambil daftar poliklinik yang tersedia dan menampilkannya dalam bentuk *dropdown* pada *form*. *Admin* mengisi data dokter dan memilih poliklinik terkait. Setelah data di-*submit*, sistem meneruskan informasi ke *DokterResource* untuk membuat entri dokter baru dalam *database*. Setelah dokter baru berhasil ditambahkan, sistem mengirimkan respons sukses dan menampilkan pesan konfirmasi kepada *Admin*, menandakan bahwa proses penambahan dokter telah berhasil dilakukan.

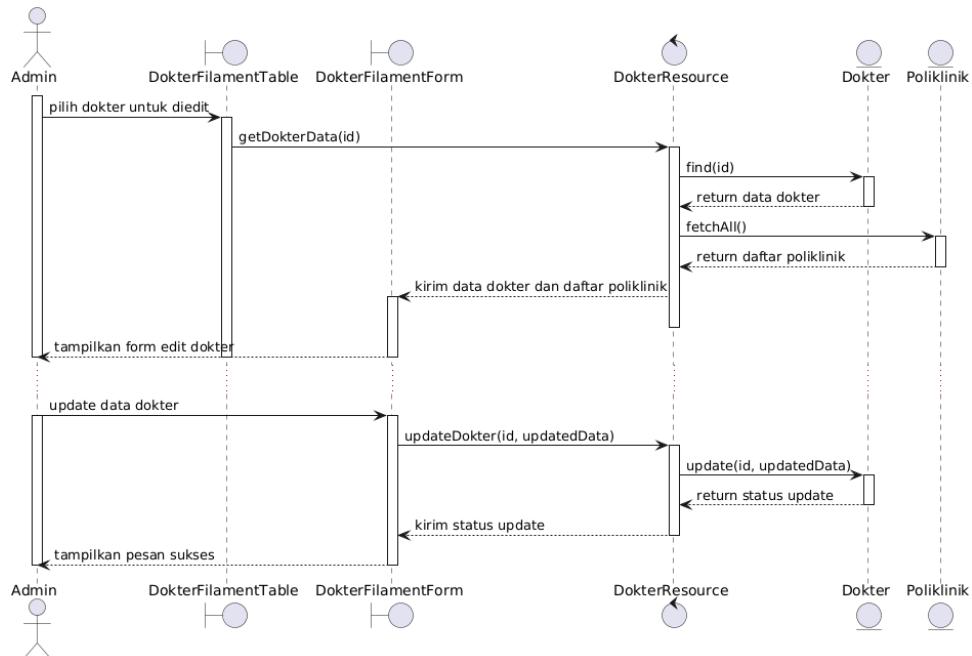
- Lihat Data Dokter



Gambar 20. Sequence Diagram Lihat Data Dokter

Sequence diagram ini menjelaskan proses melihat dan memfilter daftar dokter oleh *Admin*. Awalnya, *Admin* meminta untuk melihat daftar dokter. Sistem mengambil data dokter dan poliklinik terkait melalui *DokterResource*. Daftar dokter lengkap dengan nama poliklinik ditampilkan. Selanjutnya, *Admin* menggunakan filter poliklinik untuk menyaring daftar. Sistem menerapkan filter ini melalui *DokterResource*, yang mengembalikan data dokter yang telah difilter. Akhirnya, sistem menampilkan daftar dokter terfilter kepada *Admin*.

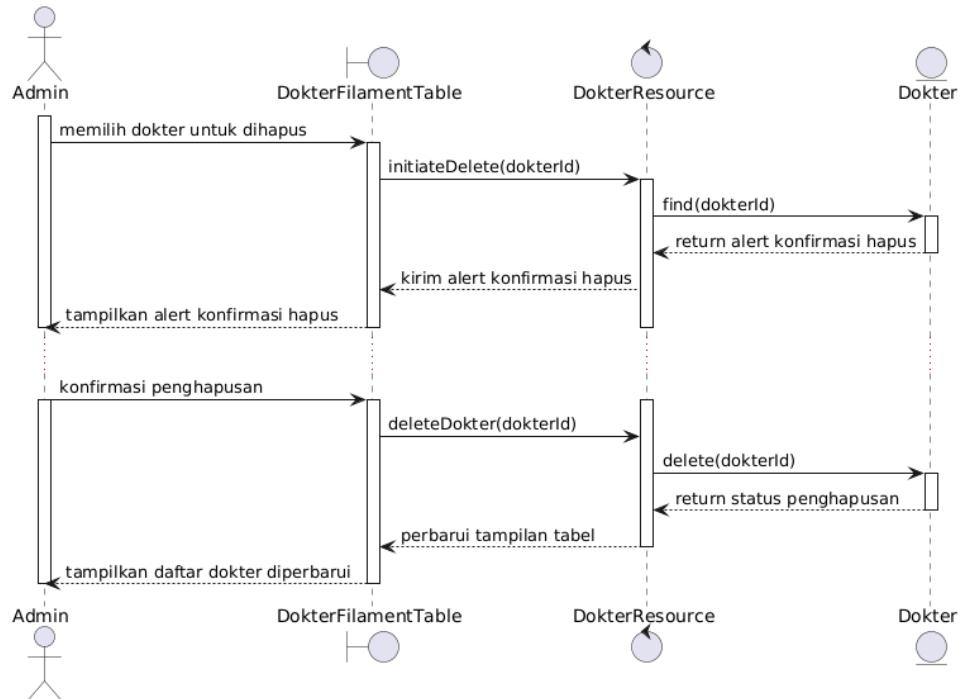
- Ubah Data Dokter



Gambar 21. Sequence Diagram Ubah Data Dokter

Sequence diagram ini menggambarkan proses *edit* data dokter oleh *Admin*. Proses dimulai dengan *Admin* memilih dokter untuk di-*edit*. Sistem mengambil data dokter tersebut dan daftar poliklinik dari *database* melalui *DokterResource*. *Form edit* dokter ditampilkan dengan data yang ada dan daftar poliklinik. *Admin* kemudian mengubah data dokter pada *form*. Setelah perubahan dilakukan, sistem mengirim data yang diperbarui ke *DokterResource* untuk diupdate di *database*. Status *update* dikirim kembali melalui sistem, dan akhirnya pesan sukses ditampilkan kepada *Admin*, menandakan bahwa proses *edit* data dokter telah berhasil dilakukan.

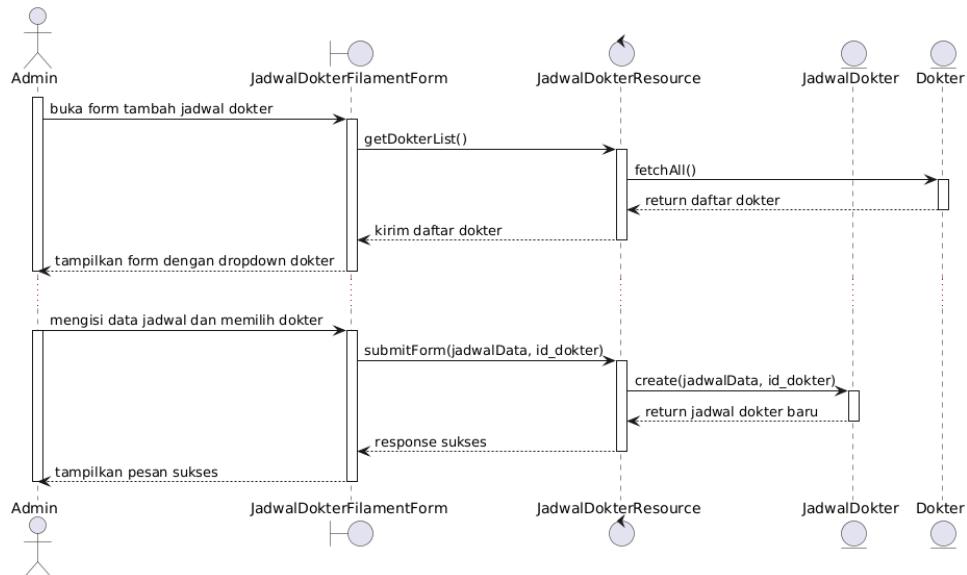
- Hapus Data Dokter



Gambar 22. Sequence Diagram Hapus Data Dokter

Sequence diagram ini menggambarkan proses penghapusan data dokter oleh *Admin*. Prosesnya dimulai dengan *Admin* memilih dokter yang akan dihapus. *DokterFilamentTable* kemudian menginisiasi penghapusan dengan mengirim permintaan ke *DokterResource*. *DokterResource* mencari data dokter dan mengembalikan konfirmasi penghapusan. Setelah *Admin* mengonfirmasi, *DokterFilamentTable* mengirim perintah penghapusan ke *DokterResource*, yang kemudian menghapus data dari *model Dokter*. Setelah penghapusan berhasil, tampilan tabel dokter diperbarui dan daftar dokter yang telah diperbarui ditampilkan kepada *Admin*.

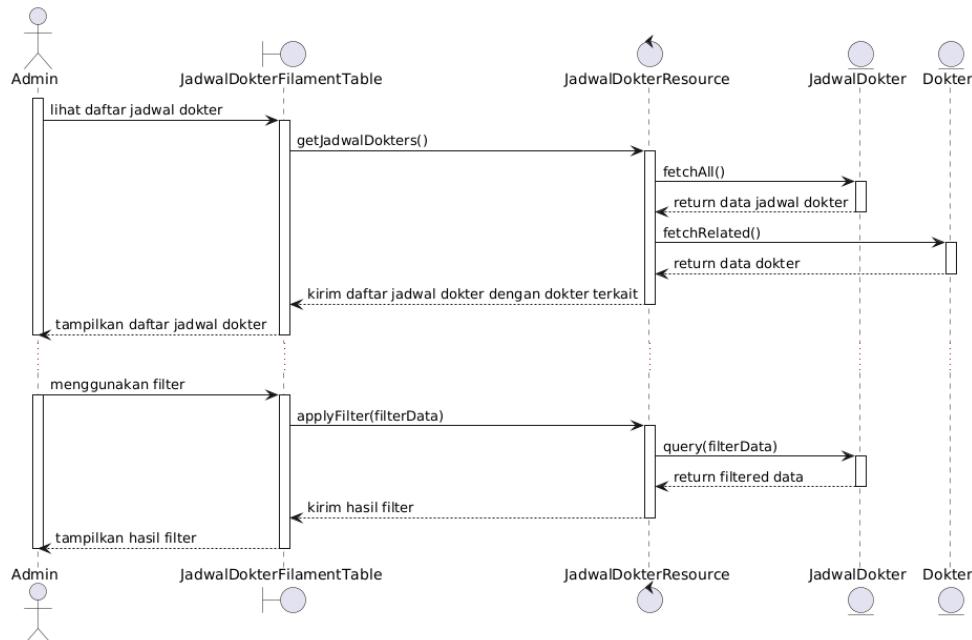
- *Create Data Jadwal Dokter*



Gambar 23. Sequence Diagram *Create Data Jadwal Dokter*

Sequence diagram ini mengilustrasikan proses penambahan jadwal dokter oleh *Admin*. Prosesnya dimulai ketika *Admin* membuka *form* tambah jadwal dokter. *JadwalDokterFilamentForm* kemudian meminta daftar dokter dari *JadwalDokterResource*, yang mengambil data dari *model* *Dokter*. *Form* ditampilkan dengan *dropdown* berisi daftar dokter. *Admin* mengisi data jadwal dan memilih dokter, lalu mengirimkan *form*. *JadwalDokterFilamentForm* meneruskan data ke *JadwalDokterResource* untuk membuat jadwal baru di *model* *JadwalDokter*. Setelah jadwal berhasil dibuat, respons sukses dikirim kembali melalui *JadwalDokterFilamentForm*, dan pesan sukses ditampilkan kepada *Admin*.

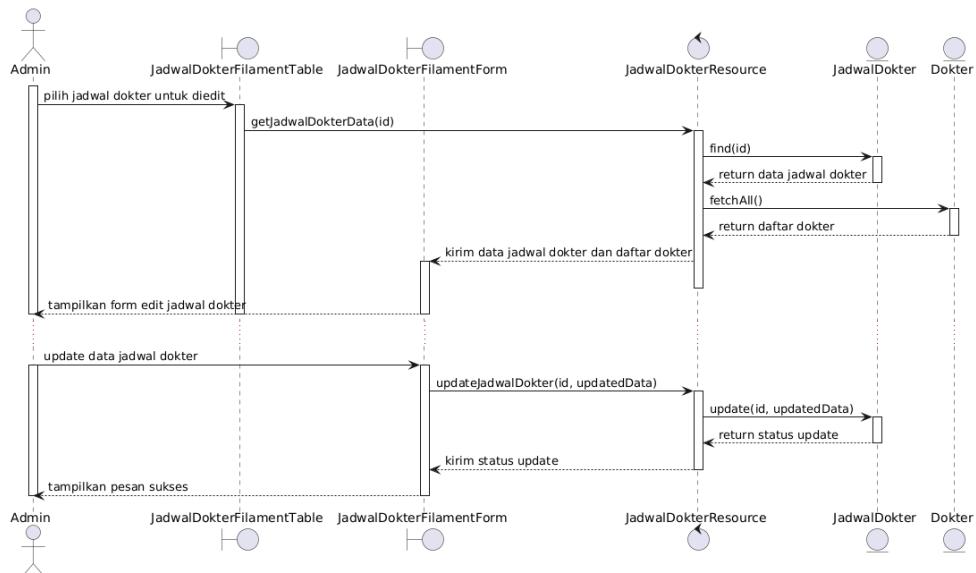
- Lihat Data Jadwal Dokter



Gambar 24. Sequence Diagram Lihat Data Jadwal Dokter

Sequence diagram ini menggambarkan proses melihat dan memfilter daftar jadwal dokter oleh *Admin*. Awalnya, *Admin* meminta untuk melihat daftar jadwal dokter. *JadwalDokterFilamentTable* meminta data dari *JadwalDokterResource*, yang mengambil data jadwal dokter dan data dokter terkait. Daftar jadwal dokter lengkap ditampilkan kepada *Admin*. Selanjutnya, *Admin* dapat menggunakan fitur filter. Ketika filter diterapkan, *JadwalDokterFilamentTable* mengirim permintaan filter ke *JadwalDokterResource*, yang kemudian melakukan *query* berdasarkan kriteria filter pada *model JadwalDokter*. Hasil filter dikirimkan kembali dan ditampilkan kepada *Admin*, memungkinkan mereka melihat jadwal dokter yang sesuai dengan kriteria yang dipilih.

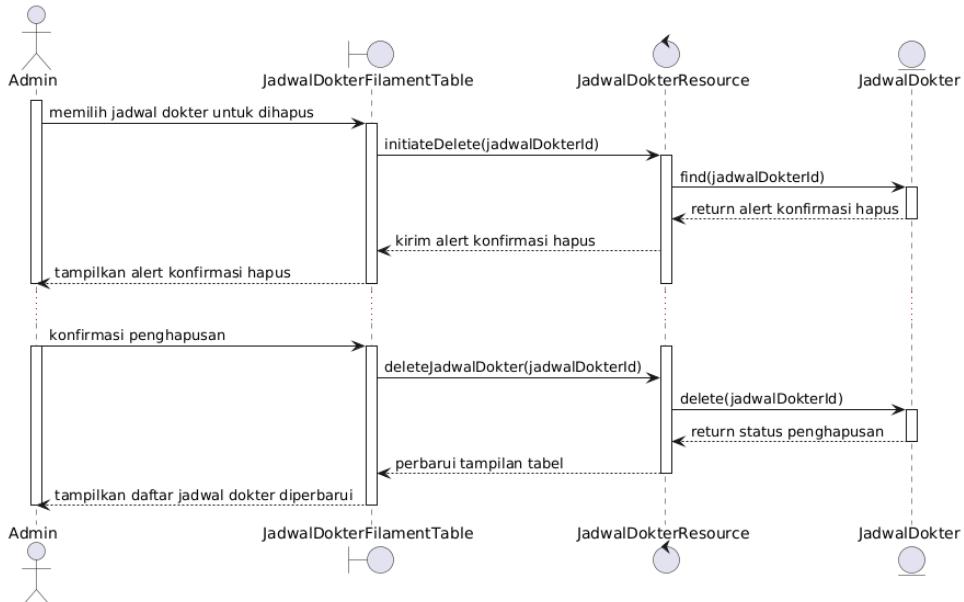
- Ubah Data Jadwal Dokter



Gambar 25. Sequence Diagram Ubah Data Jadwal Dokter

Sequence diagram ini menggambarkan proses *edit* jadwal dokter oleh *Admin*. Proses dimulai saat *Admin* memilih jadwal untuk di-*edit*. *JadwalDokterFilamentTable* meminta data jadwal dan daftar dokter dari *JadwalDokterResource*, yang mengambilnya dari *model* *JadwalDokter* dan *Dokter*. *Form edit* ditampilkan dengan data yang ada. *Admin* melakukan perubahan dan mengirimkan *update*. *JadwalDokterFilamentForm* meneruskan data *update* ke *JadwalDokterResource*, yang memperbarui *model* *JadwalDokter*. Setelah *update* berhasil, status dikirim kembali melalui *JadwalDokterFilamentForm*, dan pesan sukses ditampilkan kepada *Admin*.

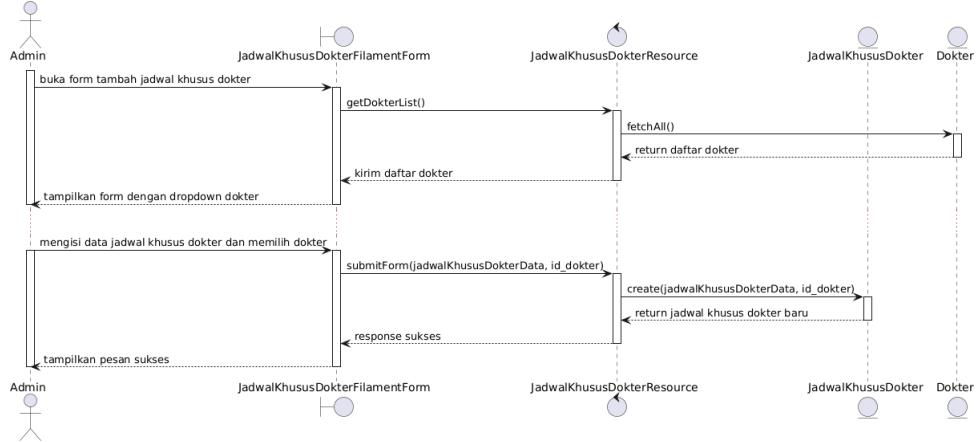
- Hapus Data Jadwal Dokter



Gambar 26. Sequence Diagram Hapus Data Jadwal Dokter

Sequence diagram ini menggambarkan proses penghapusan jadwal dokter oleh *Admin*. Prosesnya dimulai ketika *Admin* memilih jadwal dokter untuk dihapus. *JadwalDokterFilamentTable* menginisiasi penghapusan dengan mengirim permintaan ke *JadwalDokterResource*, yang mencari data jadwal dan mengembalikan konfirmasi penghapusan. Setelah *Admin* mengonfirmasi, *JadwalDokterFilamentTable* mengirim perintah penghapusan ke *JadwalDokterResource*, yang kemudian menghapus data dari *model JadwalDokter*. Setelah penghapusan berhasil, tampilan tabel jadwal dokter diperbarui dan daftar jadwal yang telah diperbarui ditampilkan kepada *Admin*.

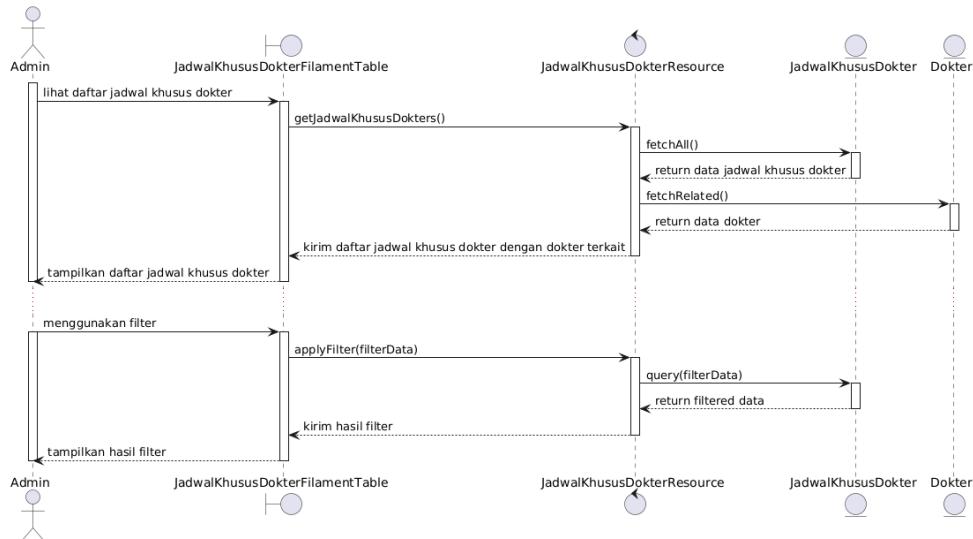
- *Create Data Jadwal Khusus Dokter*



Gambar 27. Sequence Diagram Create Data Jadwal Khusus Dokter

Sequence diagram ini menunjukkan proses penambahan jadwal khusus dokter oleh *Admin*. Dimulai dengan *Admin* membuka *form* tambah jadwal khusus, *JadwalKhususDokterFilamentForm* meminta daftar dokter dari *JadwalKhususDokterResource*, yang mengambilnya dari *model Dokter*. *Form* ditampilkan dengan *dropdown* berisi daftar dokter. *Admin* mengisi data jadwal khusus dan memilih dokter, lalu mengirimkan *form*. Data diteruskan ke *JadwalKhususDokterResource* untuk membuat jadwal khusus baru di *model JadwalKhususDokter*. Setelah berhasil dibuat, respons sukses dikirim kembali dan pesan sukses ditampilkan kepada *Admin*.

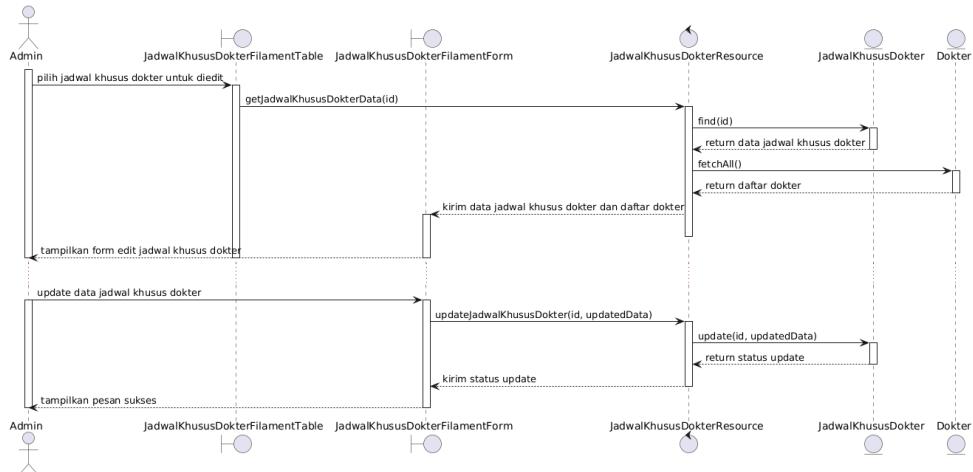
- Lihat Data Jadwal Khusus Dokter



Gambar 28. Sequence Diagram Lihat Data Jadwal Khusus Dokter

Sequence diagram ini menggambarkan proses melihat dan memfilter jadwal khusus dokter oleh *Admin*. Awalnya, *Admin* meminta daftar jadwal khusus dokter, yang kemudian diproses melalui beberapa objek (JadwalKhususDokterFilamentTable, JadwalKhususDokterResource, JadwalKhususDokter, dan Dokter) untuk mengambil dan menampilkan data. Setelah itu, *Admin* dapat menggunakan filter, yang memicu proses *filtering* data melalui objek-objek yang sama, hingga akhirnya hasil filter ditampilkan kembali kepada *Admin*.

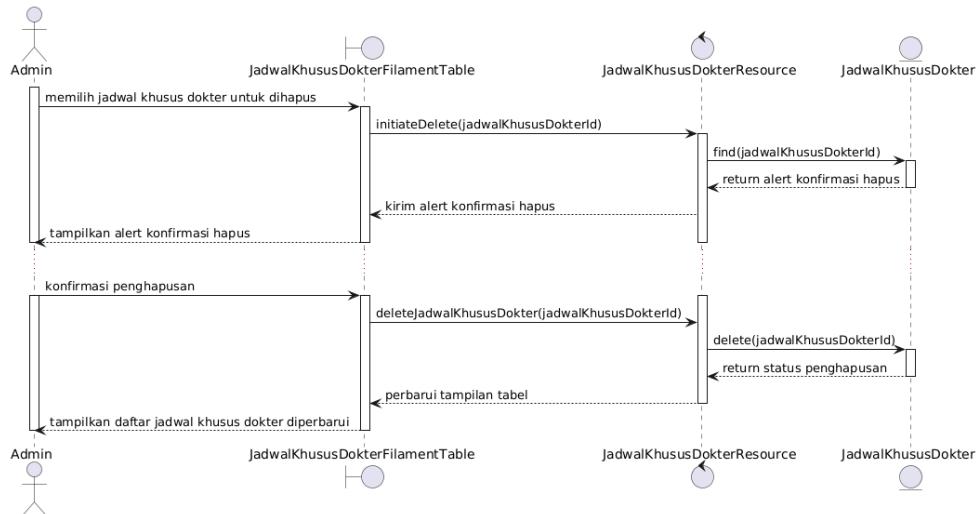
- Ubah Data Jadwal Khusus Dokter



Gambar 29. Sequence Diagram Ubah Data Jadwal Khusus Dokter

Sequence diagram ini mengilustrasikan proses pengeditan jadwal khusus dokter oleh *Admin*. Dimulai dengan *Admin* memilih jadwal untuk di-*edit*, sistem kemudian mengambil data jadwal dan daftar dokter melalui beberapa objek (JadwalKhususDokterFilamentTable, JadwalKhususDokterResource, JadwalKhususDokter, dan Dokter). Setelah *form edit* ditampilkan, *Admin* melakukan pembaruan data. Proses *update* ini melalui JadwalKhususDokterFilamentForm ke JadwalKhususDokterResource, lalu ke JadwalKhususDokter untuk menyimpan perubahan. Akhirnya, status *update* dikembalikan melalui alur yang sama, dan *Admin* menerima pesan sukses.

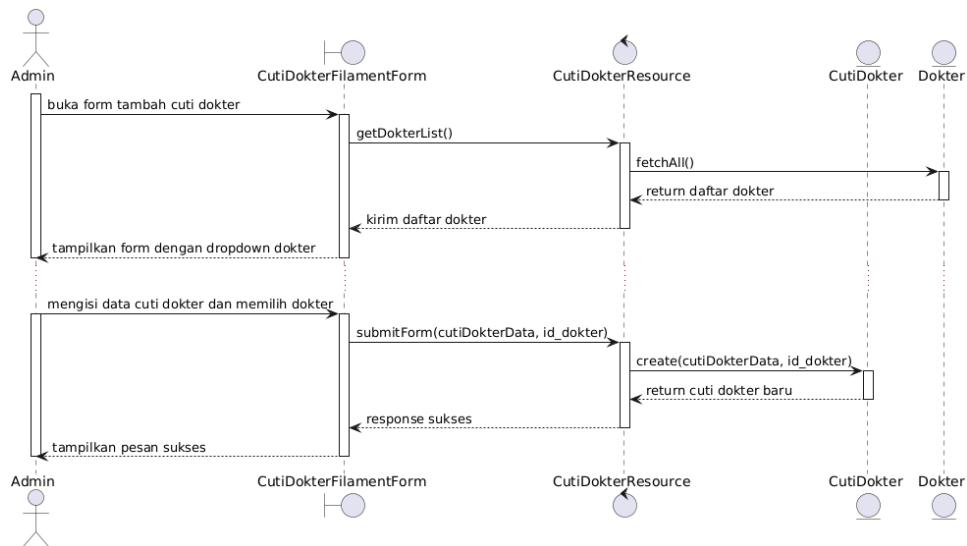
- Hapus Data Jadwal Khusus Dokter



Gambar 30. Sequence Diagram Hapus Data Jadwal Khusus Dokter

Sequence diagram ini menggambarkan proses penghapusan jadwal khusus dokter oleh *Admin*. Proses dimulai ketika *Admin* memilih jadwal untuk dihapus, yang memicu *JadwalKhususDokterFilamentTable* untuk menginisiasi penghapusan. *JadwalKhususDokterResource* kemudian mencari jadwal yang dimaksud dan mengembalikan *alert* konfirmasi. Setelah *Admin* mengkonfirmasi penghapusan, perintah *delete* diteruskan melalui *JadwalKhususDokterFilamentTable* ke *JadwalKhususDokterResource*, yang kemudian menghapus data dari *JadwalKhususDokter*. Setelah penghapusan selesai, status dikembalikan, tampilan tabel diperbarui, dan *Admin* melihat daftar jadwal khusus dokter yang telah diperbarui.

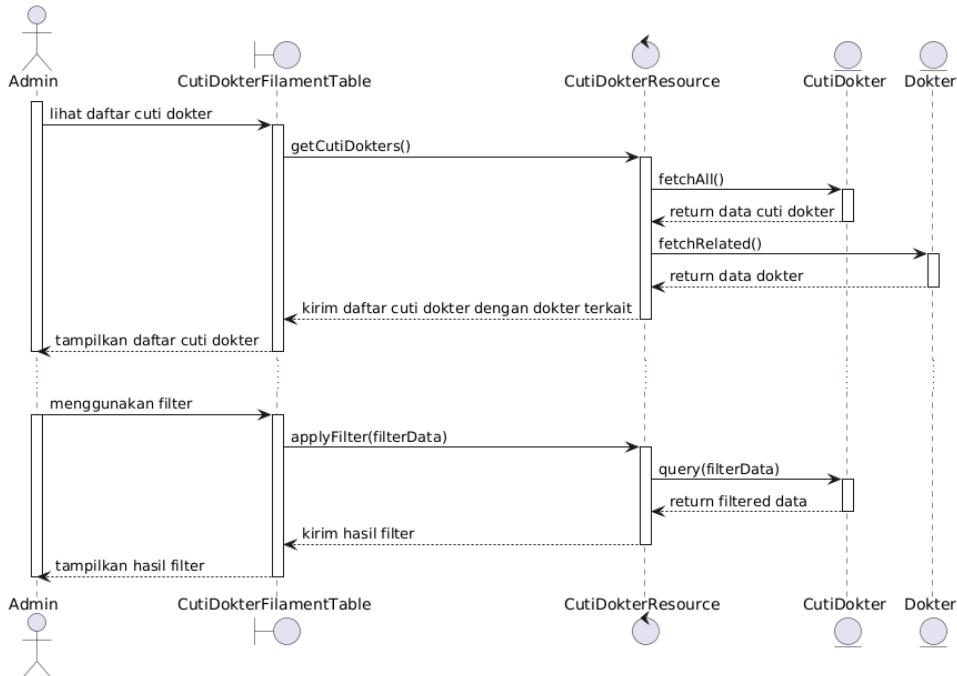
- *Create Data Cuti Dokter*



Gambar 31. Sequence Diagram *Create Data Cuti Dokter*

Sequence diagram ini mengilustrasikan proses penambahan cuti dokter oleh *Admin*. Dimulai dengan *Admin* membuka *form* tambah cuti, sistem mengambil daftar dokter melalui *CutiDokterFilamentForm* dan *CutiDokterResource*. *Form* ditampilkan dengan *dropdown* daftar dokter. *Admin* kemudian mengisi data cuti dan memilih dokter. Data ini dikirim melalui *submitForm* ke *CutiDokterResource*, yang menciptakan entri cuti dokter baru di *CutiDokter*. Setelah berhasil, sistem mengirim respons sukses kembali ke *form*, dan *Admin* melihat pesan sukses.

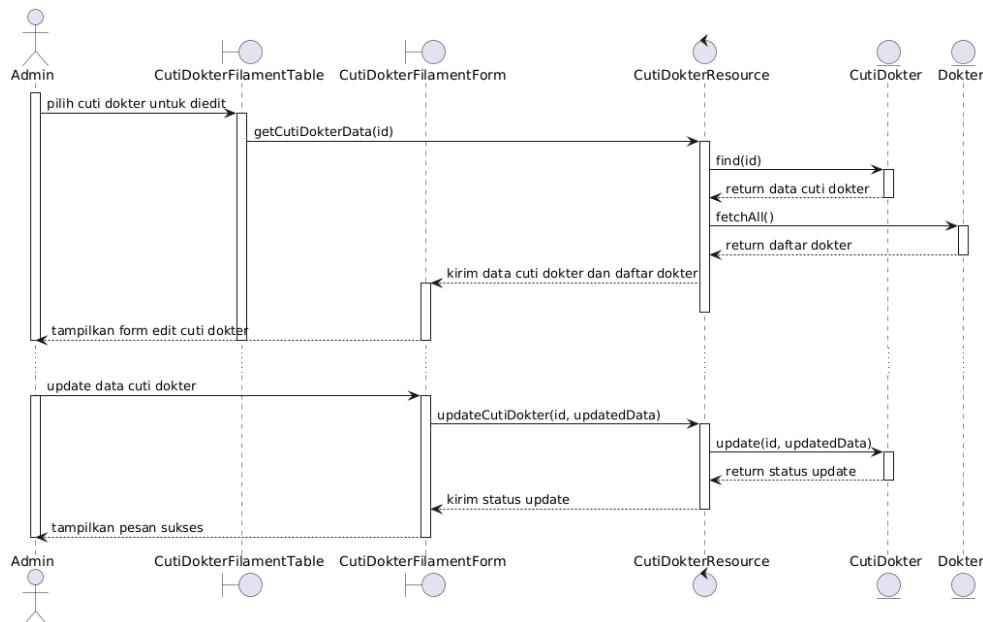
- Lihat Data Cuti Dokter



Gambar 32. Sequence Diagram Lihat Data Cuti Dokter

Sequence diagram ini menggambarkan proses melihat dan memfilter daftar cuti dokter oleh *Admin*. Awalnya, *Admin* meminta untuk melihat daftar cuti dokter, yang memicu *CutiDokterFilamentTable* untuk mengambil data melalui *CutiDokterResource*. Data cuti dokter dan data dokter terkait diambil dari *CutiDokter* dan *Dokter*, lalu ditampilkan ke *Admin*. Selanjutnya, *Admin* dapat menggunakan fitur filter, yang mengirimkan permintaan filter melalui *CutiDokterFilamentTable* ke *CutiDokterResource*. *CutiDokterResource* melakukan *query* berdasarkan filter yang diberikan, dan hasilnya dikirimkan kembali untuk ditampilkan kepada *Admin*.

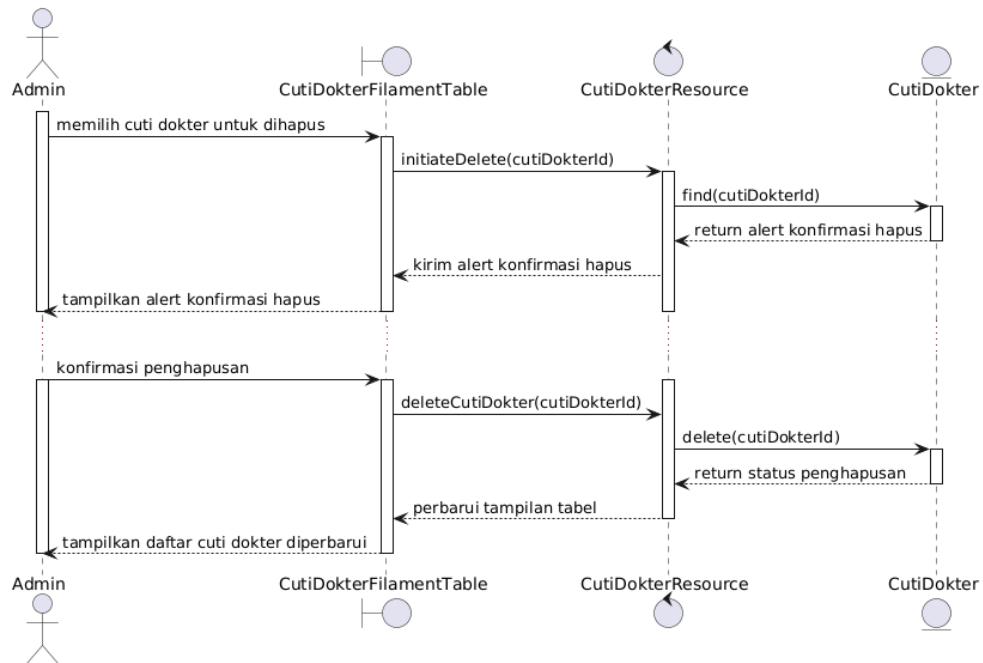
- Ubah Data Cuti Dokter



Gambar 33. Sequence Diagram Ubah Data Cuti Dokter

Sequence diagram ini mengilustrasikan proses pembaruan data cuti dokter oleh *Admin*. *Admin* memilih cuti dokter untuk di-edit, memicu *CutiDokterFilamentTable* untuk meminta data dari *CutiDokterResource*. Resource ini mengambil data cuti dokter spesifik dan daftar semua dokter. Data ini ditampilkan dalam *form edit*. Setelah *Admin* mengubah data, *CutiDokterFilamentForm* mengirim permintaan *update* ke *CutiDokterResource*, yang meneruskannya ke *model CutiDokter*. Setelah pembaruan berhasil, status *update* dikembalikan melalui *resource* ke *form*, yang kemudian menampilkan pesan sukses kepada *Admin*.

- Hapus Data Cuti Dokter

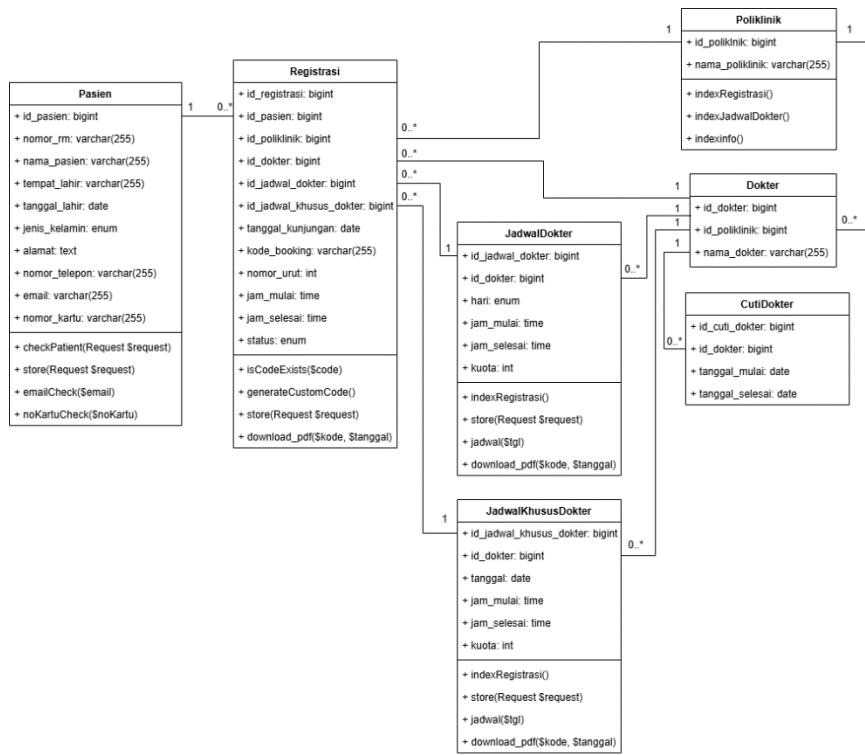


Gambar 34. Sequence Diagram Hapus Data Cuti Dokter

Sequence diagram ini menggambarkan proses penghapusan data cuti dokter oleh *Admin*. Proses dimulai ketika *Admin* memilih cuti dokter untuk dihapus, memicu *CutiDokterFilamentTable* untuk menginisiasi penghapusan. *CutiDokterResource* mencari data yang akan dihapus dan mengembalikan *alert* konfirmasi. Setelah *Admin* mengonfirmasi, *CutiDokterFilamentTable* memanggil fungsi `deleteCutiDokter` pada *CutiDokterResource*, yang kemudian menghapus data dari *model* *CutiDokter*. Status penghapusan dikembalikan, dan *CutiDokterFilamentTable* memperbarui tampilan, menunjukkan daftar cuti dokter yang telah diperbarui kepada *Admin*.

4.2.4 Class Diagram

Class diagram merepresentasikan struktur statis dari sistem registrasi *online* RSU St. Elisabeth Purwokerto. Diagram ini menggambarkan kelas-kelas dalam sistem, atribut-atributnya, metode-metodenya, serta hubungan antar kelas. Class diagram sangat penting dalam perancangan berorientasi objek, membantu dalam mendefinisikan arsitektur sistem dan menjadi dasar untuk implementasi kode. Adapun rancangan class diagram sistem ini dapat dilihat pada Gambar 35.



Gambar 35. Class Diagram Sistem

4.2.5 Desain UI/UX

Desain UI/UX merupakan aspek penting dalam pengembangan sistem registrasi *online* RSU St. Elisabeth Purwokerto, bertujuan untuk menciptakan antarmuka yang intuitif dan pengalaman pengguna yang optimal. Proses desain ini melibatkan pendekatan yang berbeda untuk tampilan pasien dan tampilan *admin*.

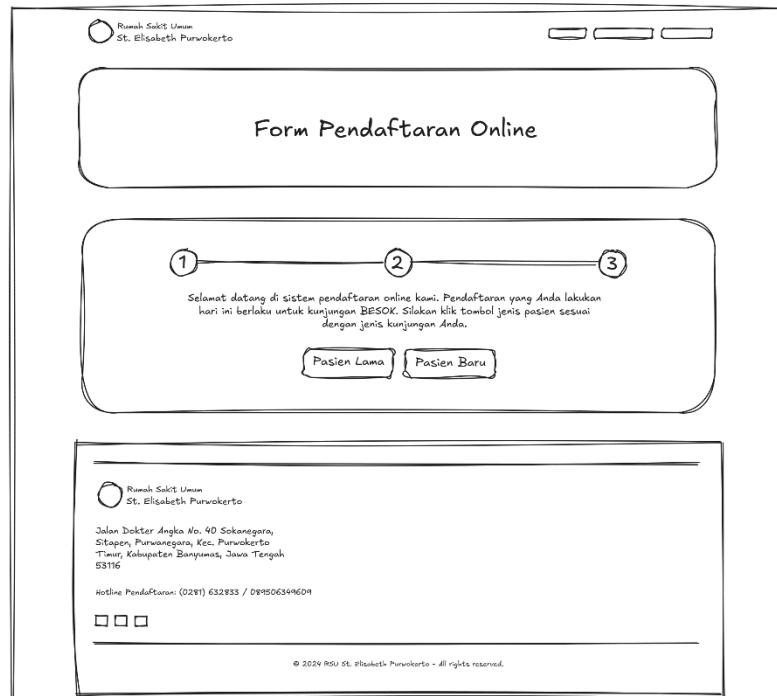
Untuk tampilan pasien, proses desain melibatkan dua tahap utama yaitu *wireframing* dan *prototyping*. Tahapan ini memungkinkan pengembangan antarmuka yang disesuaikan secara khusus dengan kebutuhan pengguna rumah sakit atau pasien.

Sementara itu, untuk tampilan *admin*, sistem ini memanfaatkan Filament, sebuah *panel admin* Laravel yang *powerful* dan mudah dikustomisasi. Penggunaan Filament memungkinkan pengembangan yang lebih cepat dan efisien untuk bagian *backend* sistem, terutama mengingat bahwa ini adalah proyek individu. Pendekatan ini meniadakan kebutuhan untuk *wireframing* dan *prototyping* pada sisi *admin*, karena Filament menyediakan komponen UI yang sudah teroptimasi dan siap pakai.

Dalam aspek visual, baik tampilan pasien maupun *admin* dirancang dengan skema warna yang didominasi oleh biru, biru muda, dan putih. Pemilihan palet warna ini bertujuan untuk menciptakan keselarasan visual dengan logo Rumah Sakit St. Elisabeth Purwokerto, sekaligus memperkuat identitas merek rumah sakit dalam sistem registrasi *online*.

a. *Wireframe*

Wireframe adalah kerangka dasar atau *blueprint* dari antarmuka sistem. Tahap ini fokus pada tata letak elemen-elemen UI, struktur navigasi, dan alur informasi tanpa mempertimbangkan aspek visual detail. Adapun rancangan *wireframe* sistem ini dapat dilihat pada Gambar 36 hingga Gambar 41.



Gambar 36. Wireframe Halaman Home/Beranda

Wireframe ini menampilkan desain dasar halaman pendaftaran online Rumah Sakit Umum St. Elisabeth Purwokerto. Terdapat *header* dengan logo rumah sakit, judul "Form Pendaftaran *Online*", dan konten utama berupa pesan selamat datang serta pilihan untuk pasien lama atau baru. *Footer* menyediakan informasi kontak rumah sakit.

The wireframe shows a mobile application interface for patient registration. At the top, there is a header with the logo of Rumah Sakit Umum St. Elisabeth Purwokerto and three small icons. Below the header is a large rounded rectangle containing the title "Form Pendaftaran Online". Inside this main form, there is a horizontal sequence of three numbered circles (1, 2, 3) connected by lines, indicating a process flow. Below the circles are two input fields: "Nomor Rekam Medis *" and "Tanggal Lahir *". At the bottom of this section are two buttons: "Kembali" and "Berikutnya". At the very bottom of the screen, there is a footer bar containing the hospital's address, contact number, and a copyright notice.

Rumah Sakit Umum
St. Elisabeth Purwokerto

Form Pendaftaran Online

① ② ③

Nomor Rekam Medis *

Tanggal Lahir *

Kembali Berikutnya

Rumah Sakit Umum
St. Elisabeth Purwokerto

Jalan Sultan Agung No. 40 Sokoanggrono,
Sukapura, Purworejo, Kec. Purwokerto
Timur, Kabupaten Banjarnas, Jawa Tengah
53116

Hotline Pendaftaran: (0291) 632833 / 081506349609

© 2024 RSU St. Elisabeth Purwokerto - All rights reserved.

Gambar 37. Wireframe Halaman Memilih Pasien Lama

Wireframe ini menunjukkan langkah kedua dalam proses pendaftaran *online* untuk pasien lama di Rumah Sakit Umum St. Elisabeth Purwokerto. Halaman ini meminta pasien memasukkan Nomor Rekam Medis dan Tanggal Lahir, dengan tombol "Kembali" dan "Berikutnya" di bawahnya.

Rumah Sakit Umum
St. Elisabeth Purwokerto

Form Pendaftaran Online

① ② ③

Nama Lengkap *

Tempat Lahir *

Tanggal Lahir *
dd/mm/yyyy

Jenis Kelamin *

Laki-laki Perempuan

Alamat

Nomor Telepon *

Email *

Nomor Kartu

Tanggal Kunjungan *
dd/mm/yyyy

Poliklinik *

Dokter *

Kembali Daftar

Rumah Sakit Umum
St. Elisabeth Purwokerto

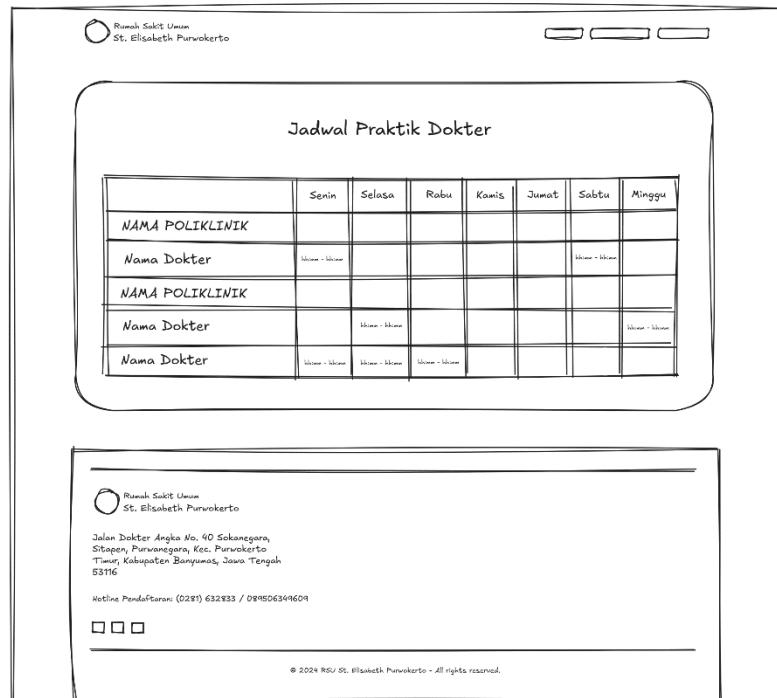
Jalan Dokter Angka No. 40 Sokanggara,
Sitapar, Purwokerto, Kec. Purwokerto
Timur, Kabupaten Banyumas, Jawa Tengah
53116

Hotline Pendaftaran: (0281) 632833 / 089506349609

© 2024 RSU St. Elisabeth Purwokerto - All rights reserved.

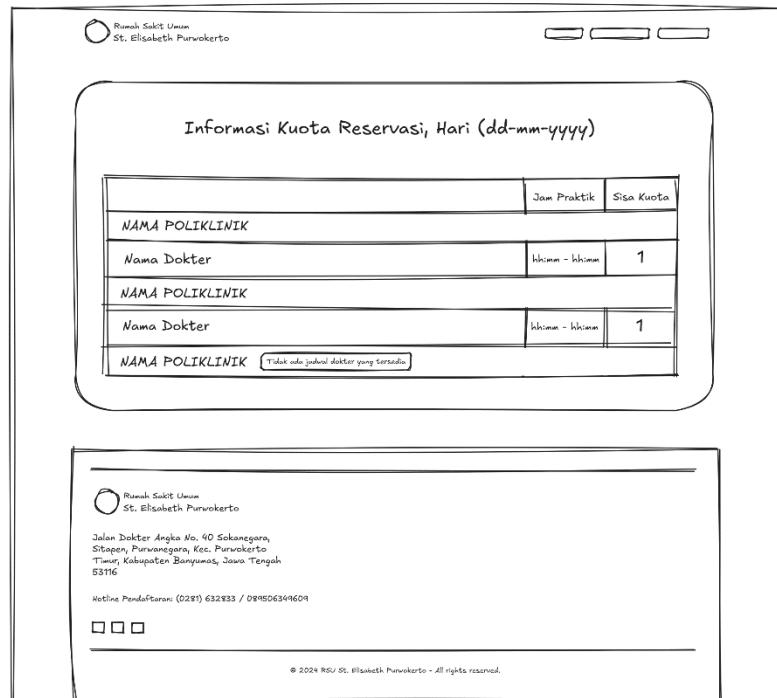
Gambar 38. Wireframe Halaman Memilih Pasien Baru

Wireframe ini menampilkan formulir pendaftaran lengkap untuk pasien baru di Rumah Sakit Umum St. Elisabeth Purwokerto. Formulir ini mencakup berbagai *field* seperti nama lengkap, tempat dan tanggal lahir, jenis kelamin, alamat, kontak, serta pilihan poliklinik dan dokter.



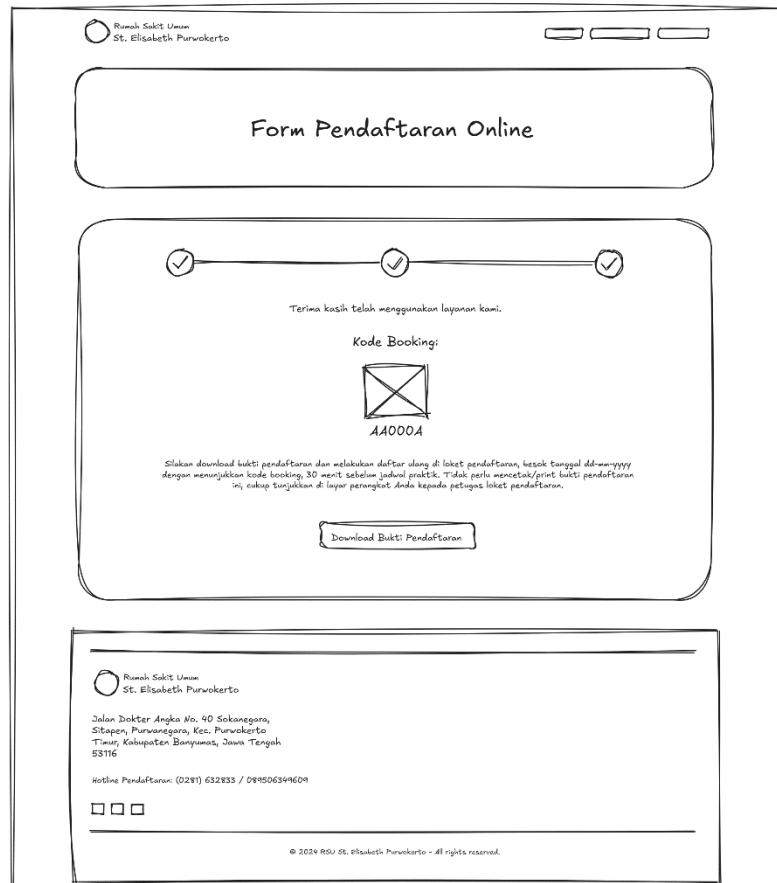
Gambar 39. Wireframe Halaman Jadwal Praktik Dokter

Wireframe ini menampilkan desain halaman Jadwal Praktik Dokter untuk Rumah Sakit Umum St. Elisabeth Purwokerto. Elemen utamanya meliputi *header* dengan logo rumah sakit, judul halaman, tabel jadwal mingguan yang menampilkan nama poliklinik dan dokter beserta waktu praktiknya, serta *footer* yang berisi informasi kontak dan alamat rumah sakit.



Gambar 40. Wireframe Halaman Info Kuota Besok

Wireframe ini menampilkan halaman Informasi Kuota Reservasi untuk Rumah Sakit Umum St. Elisabeth Purwokerto. Halaman ini mencakup tabel dengan informasi nama poliklinik, nama dokter, jam praktik, dan sisa kuota untuk hari besok. Terdapat juga keterangan untuk poliklinik tanpa jadwal dokter. Footer menyediakan informasi kontak rumah sakit.

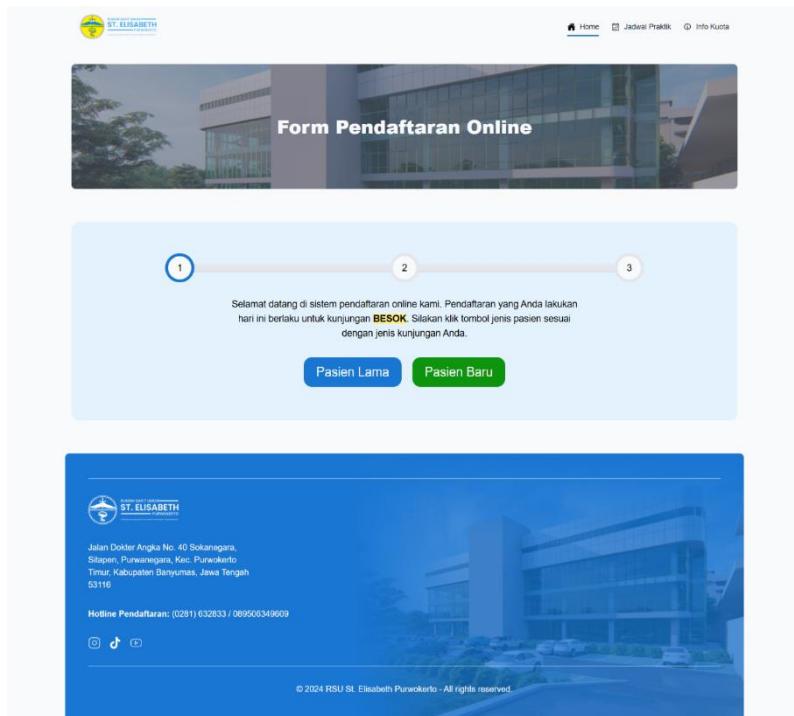


Gambar 41. Wireframe Halaman Selesai Daftar

Wireframe ini menunjukkan halaman konfirmasi setelah pendaftaran *online* selesai di Rumah Sakit Umum St. Elisabeth Purwokerto. Halaman ini menampilkan *progress bar*, kode booking (AA000A) serta QR *code*, instruksi untuk *check-in*, dan tombol untuk mengunduh bukti pendaftaran.

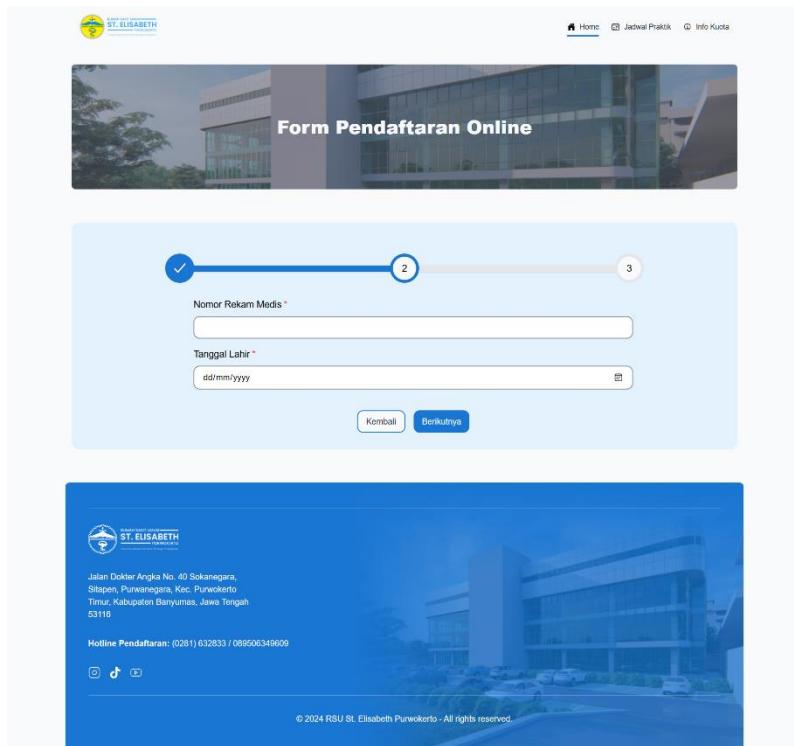
b. Prototype

Prototype merupakan versi lebih detail dan interaktif dari *wireframe*. Pada tahap ini, elemen-elemen visual seperti warna, tipografi, dan gambar ditambahkan untuk memberikan gambaran yang lebih realistik tentang tampilan akhir sistem. Adapun rancangan *prototype* sistem ini dapat dilihat pada Gambar 42 hingga Gambar 47.



Gambar 42. *Prototype Halaman Home/Beranda*

Gambar 42 menunjukkan *prototype* halaman beranda untuk sistem pendaftaran *online* Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 36. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.



Gambar 43. *Prototype Halaman Memilih Pasien Lama*

Gambar 43 menunjukkan *prototype* halaman langkah kedua dalam proses pendaftaran *online* untuk pasien lama Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 37. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.

ST. ELISABETH
PURWOKERTO

Form Pendaftaran Online

Home Jadwal Praktik Info Kuda

Nama Lengkap *

Tempat Lahir *

Tanggal Lahir *

Jenis Kelamin *

Laki-laki Perempuan

Alamat

Nomor Telepon *

Email *

Nomor Kartu

Tanggal Kunjungan *

dd/mm/2024

Poliklinik *

Pilih Poliklinik

Dokter *

Pilih Dokter

Kembali Daftar

RSU St. Elisabeth Purwokerto

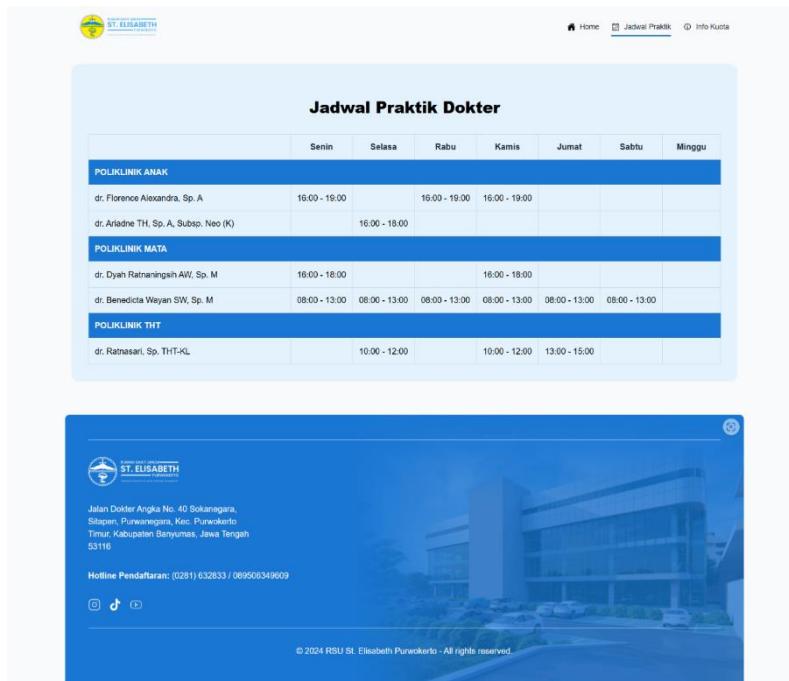
Jalan Dokter Angka No. 40 Sokanegara,
Slipan, Purworejo, Kec. Purwokerto
Timur, Kabupaten Banyumas, Jawa Tengah
53116

Hotline Pendaftaran: (0281) 632833 / 089506349609

© 2024 RSU St. Elisabeth Purwokerto - All rights reserved.

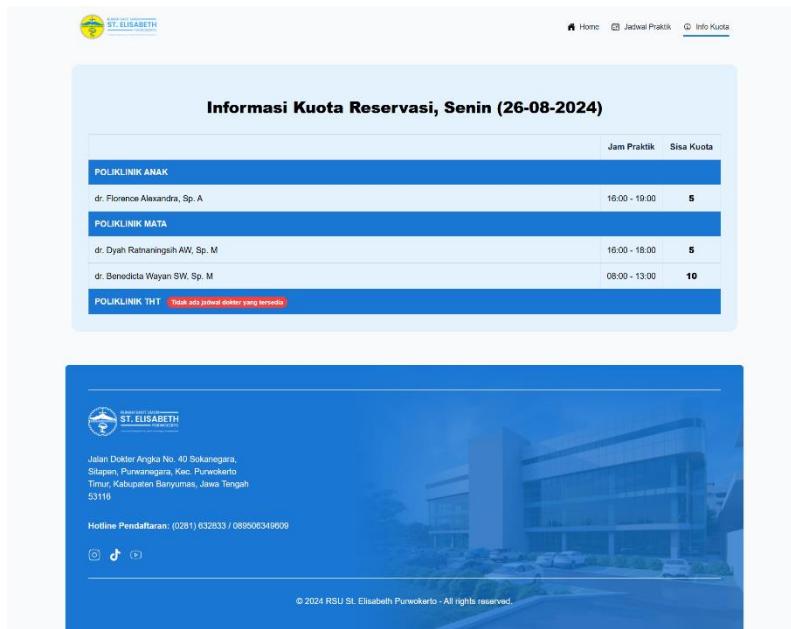
Gambar 44. Prototype Halaman Memilih Pasien Baru

Gambar 44 menunjukkan *prototype* halaman langkah kedua dalam proses pendaftaran *online* untuk pasien baru Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 38. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.



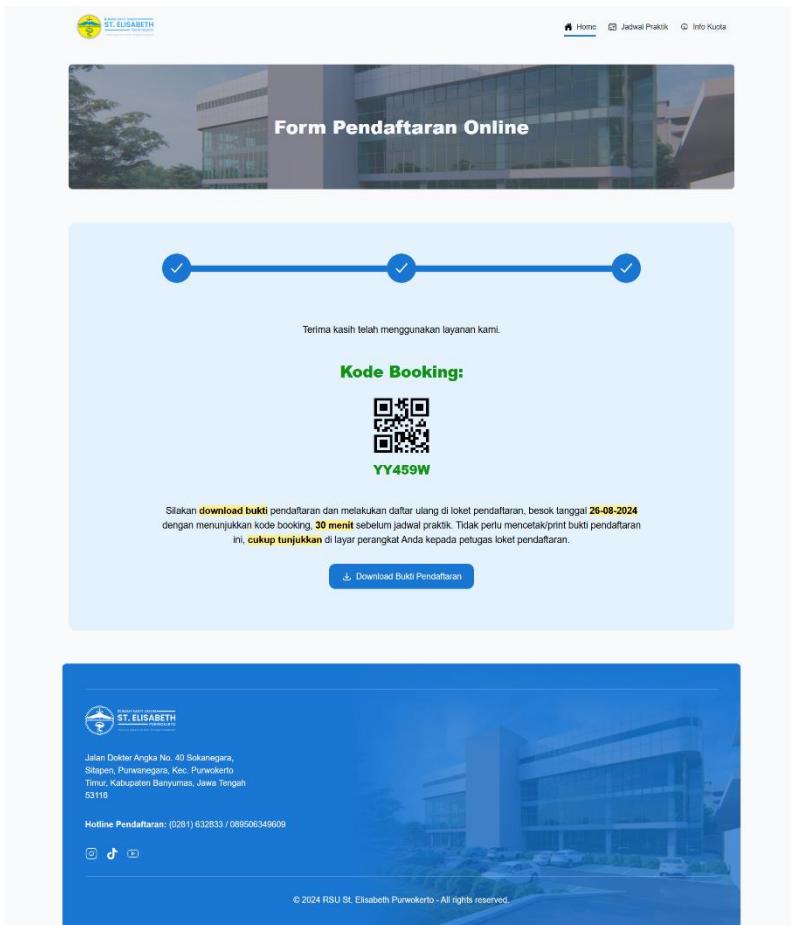
Gambar 45. *Prototype Halaman Jadwal Praktik Dokter*

Gambar 45 menunjukkan *prototype* halaman jadwal praktik dokter untuk pasien Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 39. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.



Gambar 46. *Prototype Halaman Info Kuota Besok*

Gambar 46 menunjukkan *prototype* halaman informasi kuota hari besoknya untuk pasien Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 40. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.



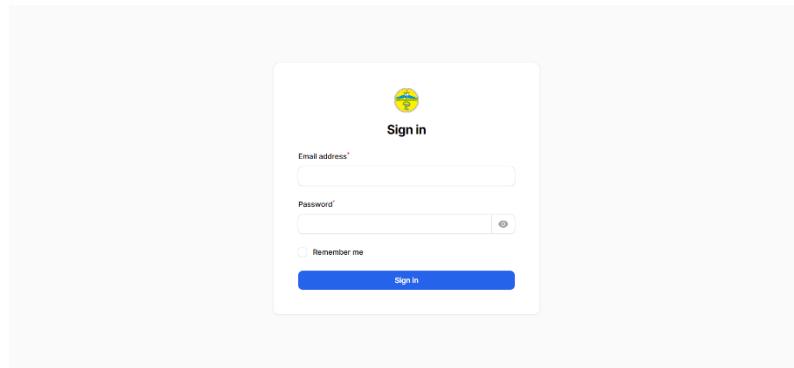
Gambar 47. Prototype Halaman Selesai Daftar

Gambar 47 menunjukkan *prototype* halaman langkah ketiga/terakhir dalam proses pendaftaran *online* untuk pasien baik lama maupun baru Rumah Sakit Umum St. Elisabeth Purwokerto. *Prototype* ini merupakan pengembangan dari *wireframe* pada Gambar 41. Desain ini menawarkan antarmuka yang jelas dan mudah digunakan bagi pasien.

c. Tampilan Admin Filament

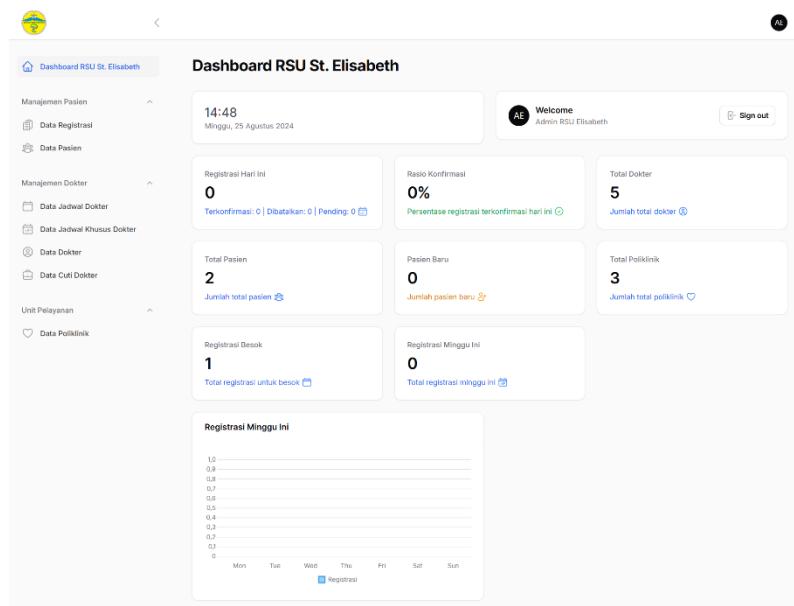
Untuk antarmuka *admin*, sistem ini memanfaatkan Filament, sebuah *panel admin* Laravel yang *powerful* dan mudah dikustomisasi. Filament dipilih karena kemampuannya menyediakan antarmuka *admin* yang kompleks dan fungsional dengan usaha pengembangan yang minimal, sangat sesuai untuk proyek individu ini. Penggunaan Filament memungkinkan fokus lebih besar pada pengembangan fitur-fitur kritis

sistem, sambil tetap menyediakan antarmuka *admin* yang *robust* dan intuitif. Berikut adalah beberapa tampilan dari antarmuka *admin* yang dihasilkan menggunakan Filament (lihat pada Gambar 48 hingga Gambar 78).



Gambar 48. Tampilan Halaman *Admin Login*

Tampilan ini merupakan tampilan ketika *admin* baru saja mengakses halaman *admin*. *Admin* akan *login* dengan mengisi *email* dan *password* *admin* agar dapat masuk pada sistem *admin*.



Gambar 49. Tampilan Halaman *Admin Dashboard*

Tampilan ini merupakan tampilan ketika *admin* baru saja masuk ke sistem *admin*. *Admin* akan dapat melihat tampilan *dashboard* terlebih

dahulu, di mana terdapat *widget-widget* informatif yang menampilkan ringkasan data penting rumah sakit.

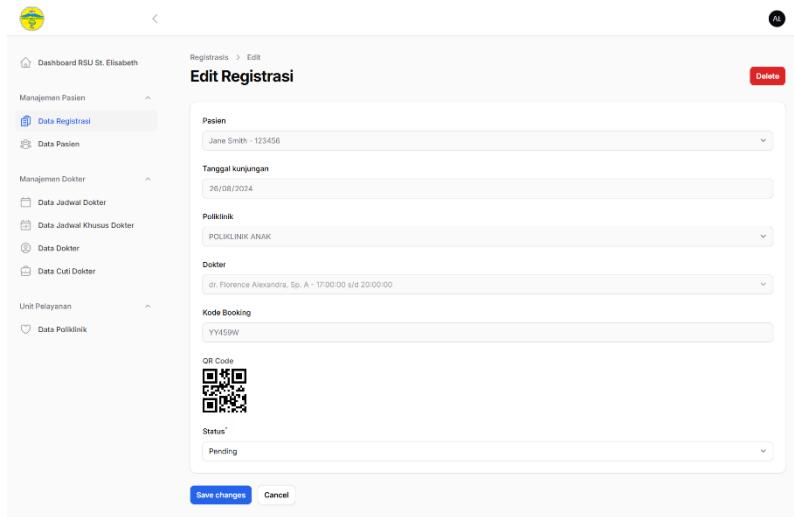
No	Nama Pasien	Tanggal kunjungan	PoliKlinik	Dokter	Jam Praktik	Kode Booking	No Urut	Status
1	Jane Smith	Agt 26, 2024	POLIKLINIK ANAK	dr. Florence Alexandra, Sp. A	16:00 - 19:00	YY459W	1	Pending

Gambar 50. Tampilan Halaman *Admin* Data Registrasi

Tampilan ini merupakan tampilan ketika *admin* mengakses data registrasi. *Admin* akan dapat melihat *detail* data registrasi apabila menekan salah satu data registrasi, *admin* juga bisa *filter* status data registrasi yang *pending*, *confirmed*, atau *canceled*.

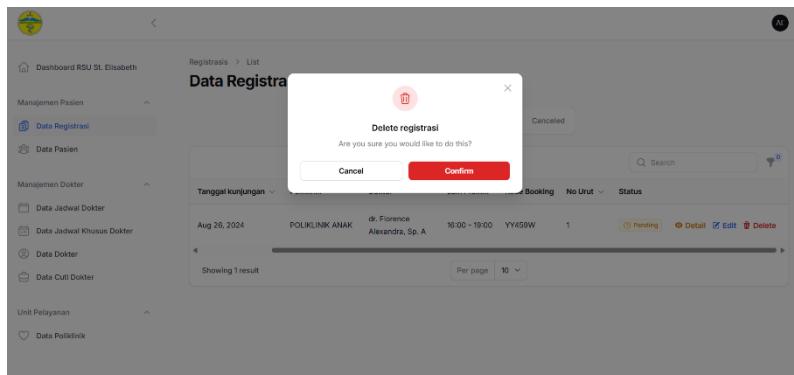
Gambar 51. Tampilan Halaman *Admin* Detail Registrasi

Tampilan ini merupakan tampilan *detail* registrasi. Seperti yang sudah dijelaskan sebelumnya, ketika *admin* menekan salah satu data registrasi maka akan muncul *modal* seperti gambar. *Admin* akan dapat melihat informasi data registrasi lebih lanjut pada tampilan ini.



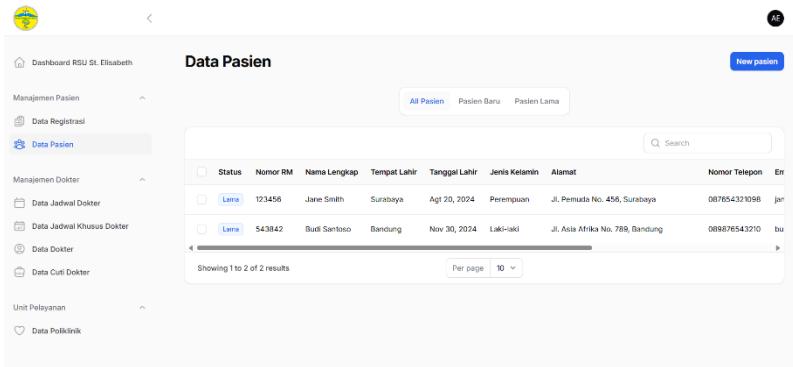
Gambar 52. Tampilan Halaman *Admin* Ubah Registrasi

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data registrasi. *Admin* akan dapat mengubah statusnya saja, namun untuk nama pasien, tanggal kunjungan, poliklinik, dokter, dan kode *booking* tidak bisa diubah atau *disable*.



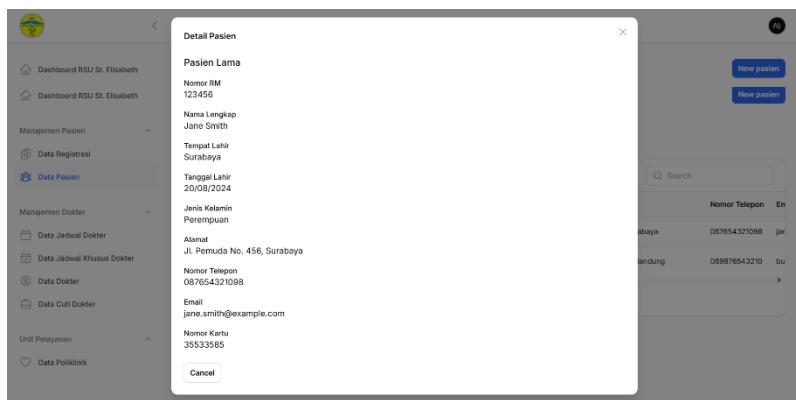
Gambar 53. Tampilan Halaman *Admin* Hapus Registrasi

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada registrasi. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data registrasi tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



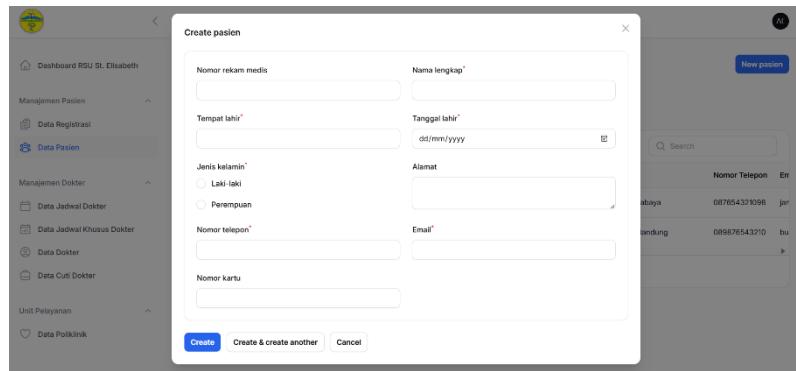
Gambar 54. Tampilan Halaman *Admin* Data Pasien

Tampilan ini merupakan tampilan ketika *admin* mengakses data pasien. *Admin* akan dapat melihat *detail* data pasien apabila menekan salah satu data pasien, *admin* juga bisa *filter* status data pasien yang merupakan tipe pasien baru atau tipe pasien lama.



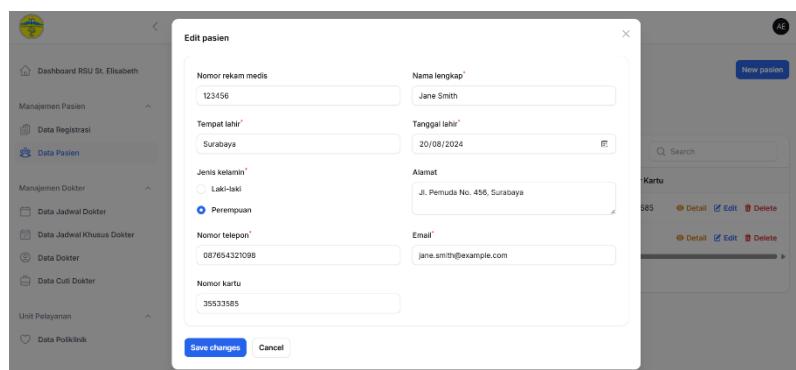
Gambar 55. Tampilan Halaman *Admin* Detail Pasien

Tampilan ini merupakan tampilan *detail* pasien. Seperti yang sudah dijelaskan sebelumnya, ketika *admin* menekan salah satu data pasien maka akan muncul *modal* seperti gambar. *Admin* akan dapat melihat informasi data pasien lebih lanjut pada tampilan ini.



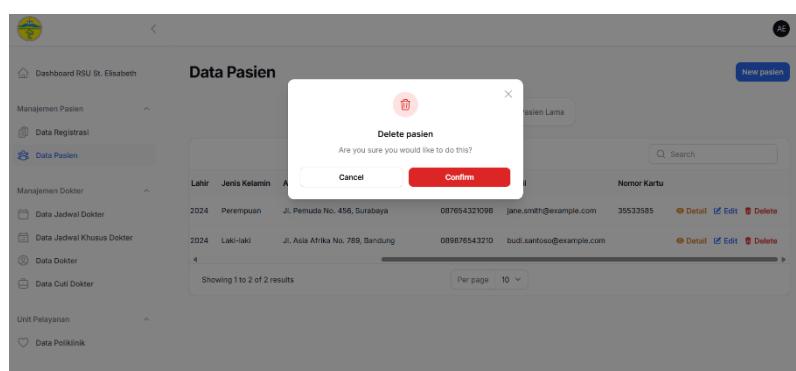
Gambar 56. Tampilan Halaman *Admin Create Pasien*

Tampilan ini merupakan tampilan *create* pasien. *Admin* akan mengisi data pasien pada *field* yang tersedia, lalu dapat menambahkan pasien dengan menekan tombol *create* berwarna biru.



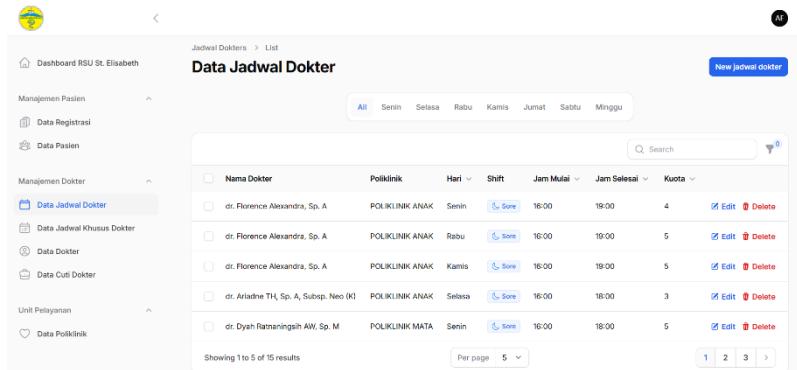
Gambar 57. Tampilan Halaman *Admin Ubah Pasien*

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data pasien. *Admin* akan dapat mengubah seluruh data-data pasien yang diinginkan.



Gambar 58. Tampilan Halaman *Admin Hapus Pasien*

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada pasien. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data pasien tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



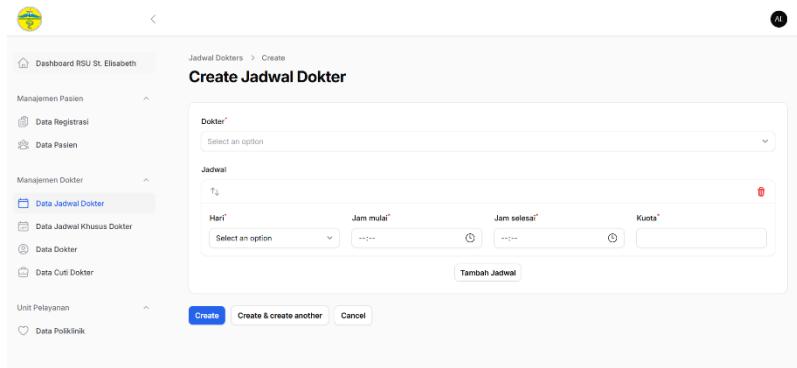
The screenshot shows a table titled "Data Jadwal Dokter" with the following data:

Nama Dokter	PoliKlinik	Hari	Shift	Jam Mulai	Jam Selesai	Kuota	Actions
dr. Florence Alexandra, Sp. A	POLIKLINIK ANAK	Senin	<input type="button" value="Sore"/>	16:00	19:00	4	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete
dr. Florence Alexandra, Sp. A	POLIKLINIK ANAK	Rabu	<input type="button" value="Sore"/>	16:00	19:00	5	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete
dr. Florence Alexandra, Sp. A	POLIKLINIK ANAK	Kamis	<input type="button" value="Sore"/>	16:00	19:00	5	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete
dr. Arladine TH, Sp. A, Subsp. Neo (KI)	POLIKLINIK ANAK	Selasa	<input type="button" value="Sore"/>	16:00	18:00	3	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete
dr. Dyah Ratnawisingh AW, Sp. M	POLIKLINIK MATA	Senin	<input type="button" value="Sore"/>	16:00	18:00	5	<input checked="" type="checkbox"/> Edit <input type="checkbox"/> Delete

Showing 1 to 5 of 15 results

Gambar 59. Tampilan Halaman Admin Data Jadwal Dokter

Tampilan ini merupakan tampilan ketika *admin* mengakses data jadwal dokter. *Admin* akan dapat melihat seluruh data jadwal dokter, *admin* juga bisa *filter* hari jadwal dokter dari hari Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, dan Minggu.



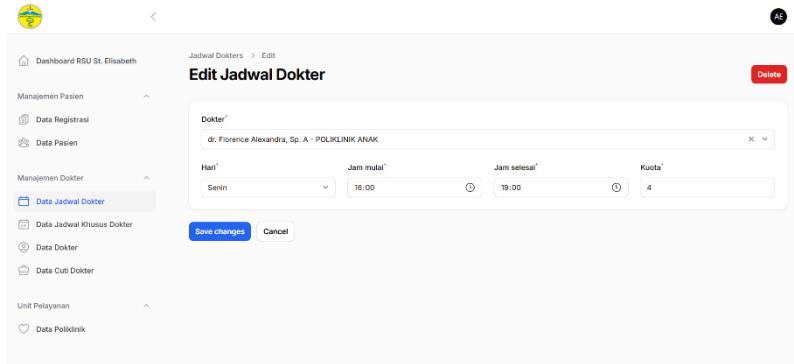
The screenshot shows a form titled "Create Jadwal Dokter" with the following fields:

- Dokter***: A dropdown menu labeled "Select an option".
- Jadwal**: A table with columns "Hari*", "Jam mulai*", "Jam selesai*", and "Kuota*". It has a "Tambah Jadwal" button at the bottom.

At the bottom, there are three buttons: "Create", "Create & create another", and "Cancel".

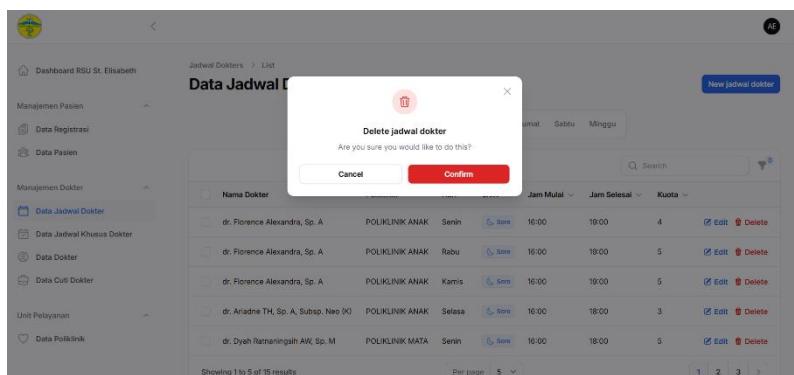
Gambar 60. Tampilan Halaman Admin Create Jadwal Dokter

Tampilan ini merupakan tampilan *create* jadwal dokter. *Admin* akan mengisi data jadwal dokter pada *field* yang tersedia dan *admin* juga dapat menambahkan lebih dari satu jadwal dokter untuk satu dokter, lalu dapat menambahkan jadwal dokter dengan menekan tombol *create* berwarna biru.



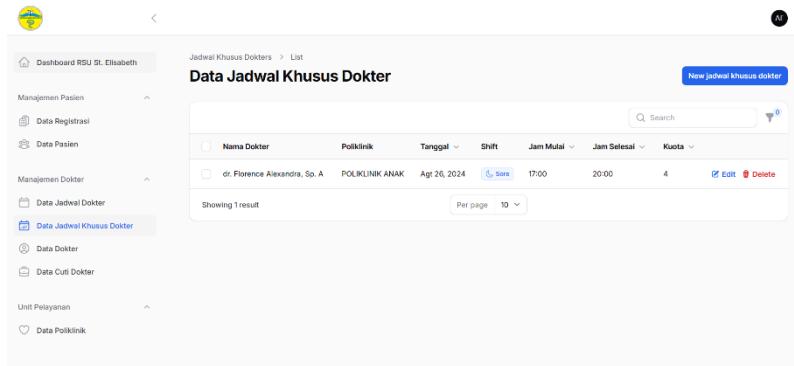
Gambar 61. Tampilan Halaman *Admin* Ubah Jadwal Dokter

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data jadwal dokter. *Admin* akan dapat mengubah seluruh data-data jadwal dokter yang diinginkan.



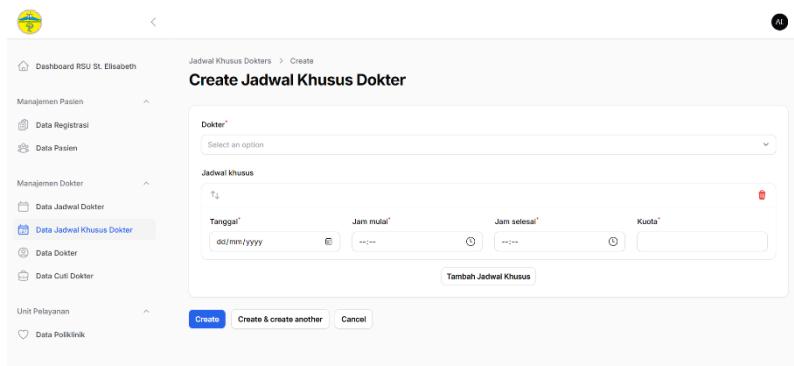
Gambar 62. Tampilan Halaman *Admin* Hapus Jadwal Dokter

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada jadwal dokter. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data jadwal dokter tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



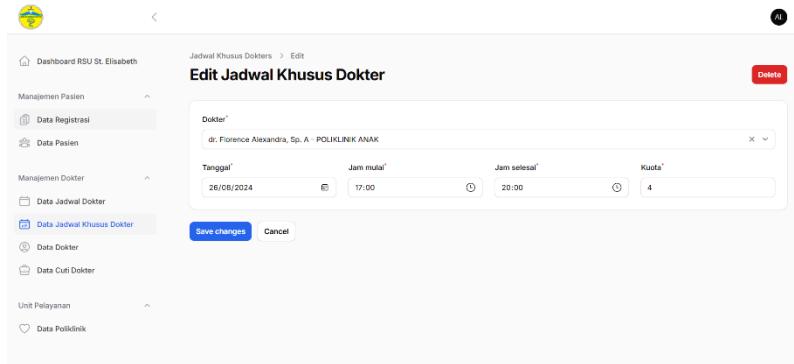
Gambar 63. Tampilan Halaman *Admin* Data Jadwal Khusus Dokter

Tampilan ini merupakan tampilan ketika *admin* mengakses data jadwal khusus dokter. *Admin* akan dapat melihat seluruh data jadwal khusus dokter, *admin* juga bisa *filter* tanggal jadwal khusus dokter yang dipilih.



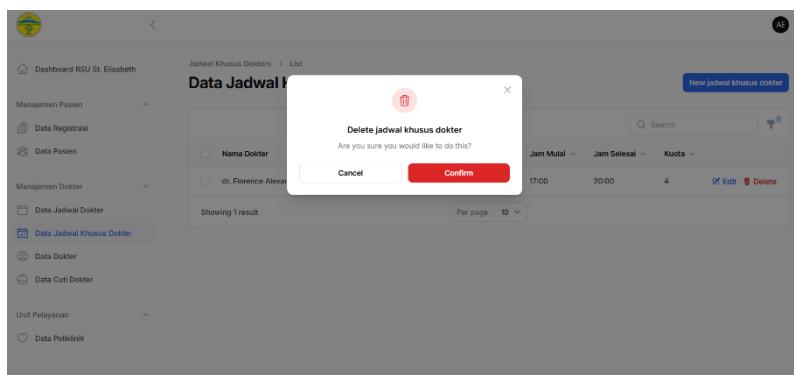
Gambar 64. Tampilan Halaman *Admin* Create Jadwal Khusus Dokter

Tampilan ini merupakan tampilan *create* jadwal khusus dokter. *Admin* akan mengisi data jadwal khusus dokter pada *field* yang tersedia dan *admin* juga dapat menambahkan lebih dari satu jadwal khusus dokter untuk satu dokter, lalu dapat menambahkan jadwal khusus dokter dengan menekan tombol *create* berwarna biru.



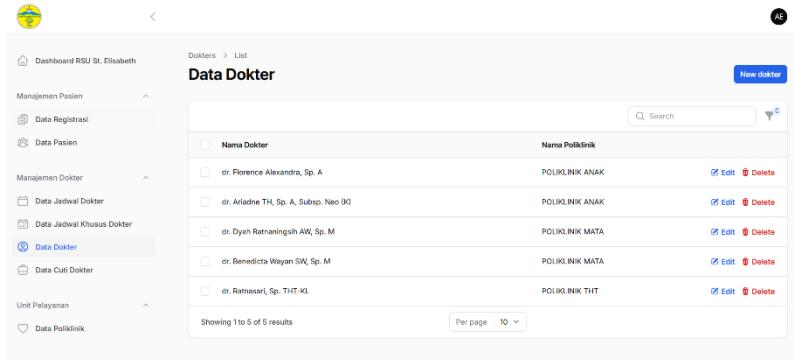
Gambar 65. Tampilan Halaman *Admin* Ubah Jadwal Khusus Dokter

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data jadwal khusus dokter. *Admin* akan dapat mengubah seluruh data-data jadwal khusus dokter yang diinginkan.



Gambar 66. Tampilan Halaman *Admin* Hapus Jadwal Khusus Dokter

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada jadwal khusus dokter. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data jadwal khusus dokter tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



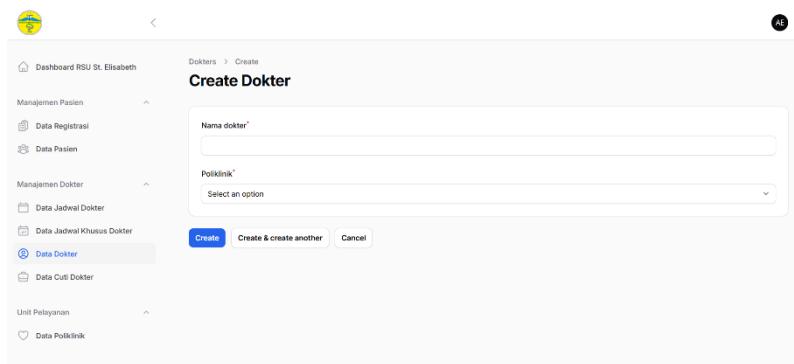
The screenshot shows a table titled "Data Dokter" under the "Dokters > List" section. The table has two columns: "Nama Dokter" and "Nama Poliklinik". There are five rows of data:

Nama Dokter	Nama Poliklinik
dr. Florence Alexandra, Sp. A	POLIKLINIK ANAK
dr. Arladne TH, Sp. A, Subsp. Neo (K)	POLIKLINIK ANAK
dr. Dyah Rathaningsih AW, Sp. M	POLIKLINIK MATA
dr. Benedicta Wayan SW, Sp. M	POLIKLINIK MATA
dr. Ratnasari, Sp. THT-KL	POLIKLINIK THT

At the bottom left, it says "Showing 1 to 5 of 5 results". At the bottom right, there are buttons for "Per page" (set to 10) and "Search".

Gambar 67. Tampilan Halaman *Admin* Data Dokter

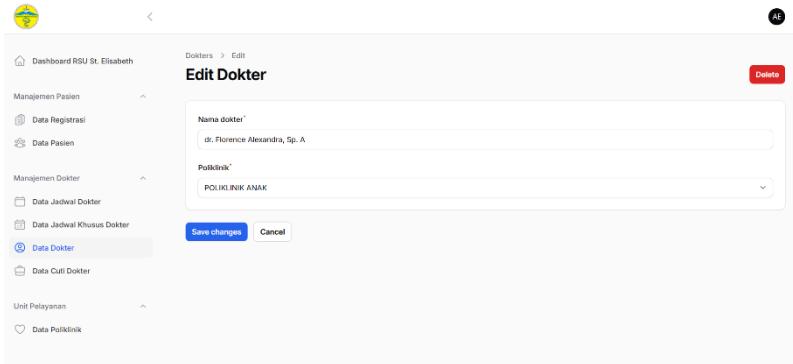
Tampilan ini merupakan tampilan ketika *admin* mengakses data dokter. *Admin* akan dapat melihat seluruh data dokter, *admin* juga bisa *filter* dokter berdasarkan poliklinik yang dipilih.



The screenshot shows a form titled "Create Dokter" under the "Dokters > Create" section. It has two input fields: "Nama dokter*" and "Poliklinik*". Below the fields are three buttons: "Create" (highlighted in blue), "Create & create another", and "Cancel".

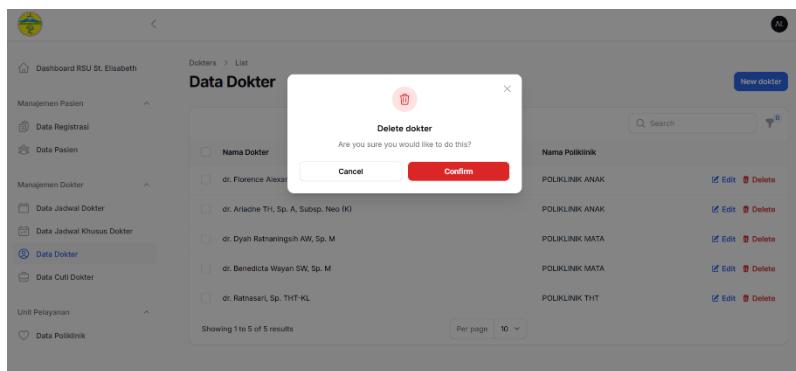
Gambar 68. Tampilan Halaman *Admin* Create Dokter

Tampilan ini merupakan tampilan *create* dokter. *Admin* akan mengisi data dokter pada *field* yang tersedia, lalu dapat menambahkan dokter dengan menekan tombol *create* berwarna biru.



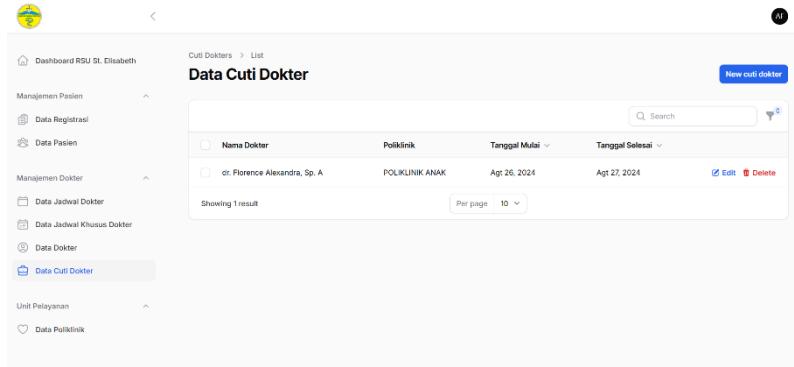
Gambar 69. Tampilan Halaman *Admin* Ubah Dokter

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data dokter. *Admin* akan dapat mengubah seluruh data-data dokter yang diinginkan.



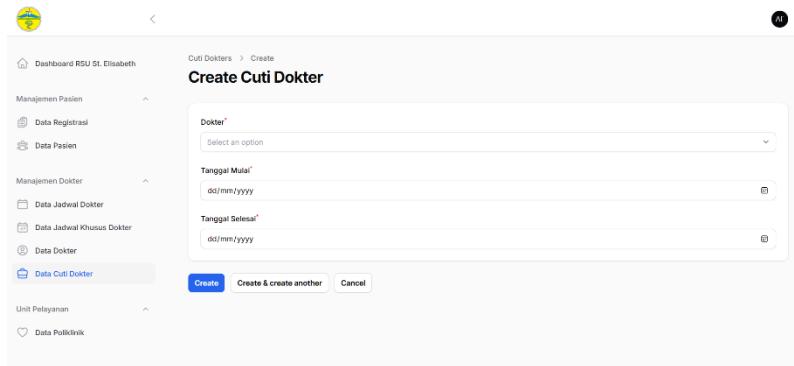
Gambar 70. Tampilan Halaman *Admin* Hapus Dokter

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada dokter. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data dokter tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



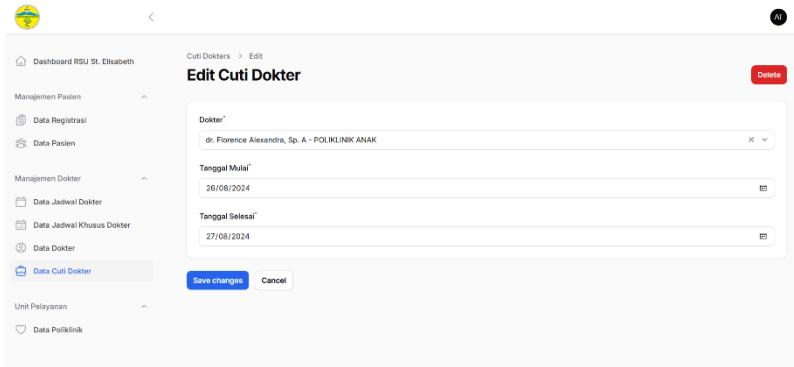
Gambar 71. Tampilan Halaman *Admin* Data Cuti Dokter

Tampilan ini merupakan tampilan ketika *admin* mengakses data cuti dokter. *Admin* akan dapat melihat seluruh data cuti dokter, *admin* juga bisa filter cuti dokter berdasarkan tanggal yang dipilih.



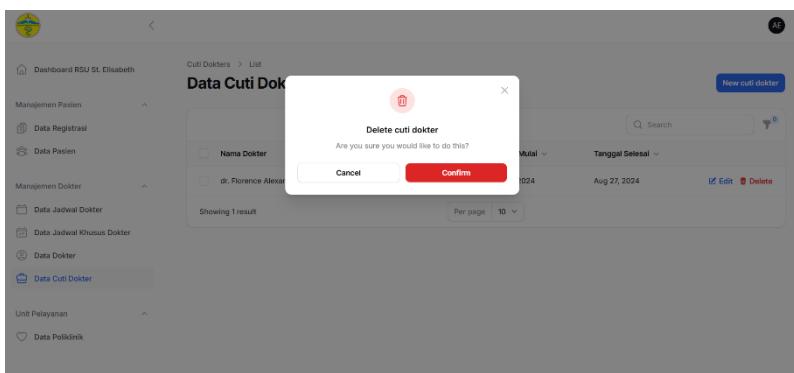
Gambar 72. Tampilan Halaman *Admin Create* Cuti Dokter

Tampilan ini merupakan tampilan *create* cuti dokter. *Admin* akan mengisi data cuti dokter pada *field* yang tersedia, lalu dapat menambahkan cuti dokter dengan menekan tombol *create* berwarna biru.



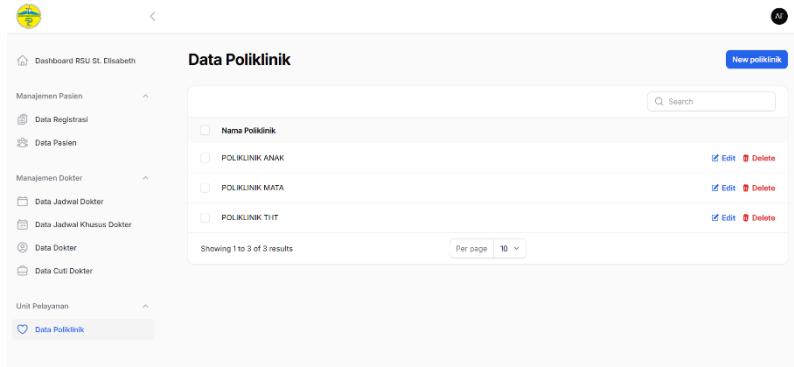
Gambar 73. Tampilan Halaman *Admin* Ubah Cuti Dokter

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data cuti dokter. *Admin* akan dapat mengubah seluruh data-data cuti dokter yang diinginkan.



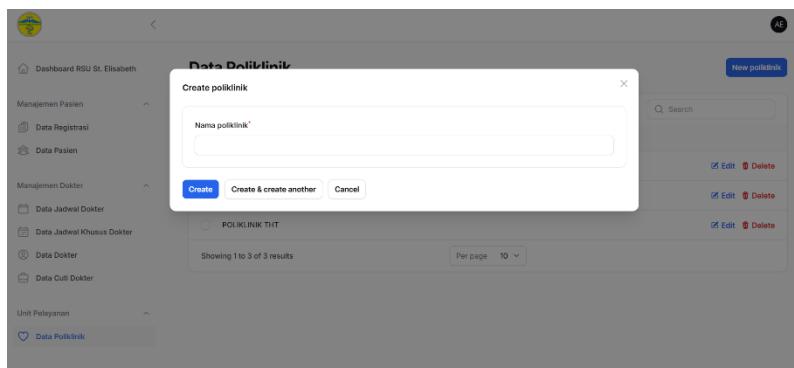
Gambar 74. Tampilan Halaman *Admin* Hapus Cuti Dokter

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada cuti dokter. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data cuti dokter tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.



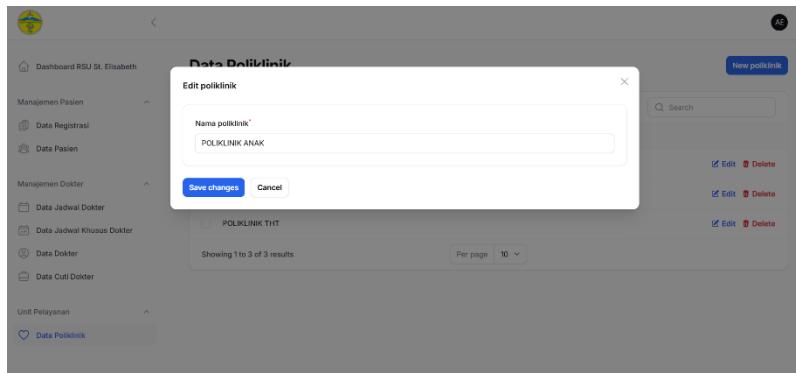
Gambar 75. Tampilan Halaman *Admin* Data Poliklinik

Tampilan ini merupakan tampilan ketika *admin* mengakses data poliklinik. *Admin* akan dapat melihat seluruh data poliklinik, *admin* juga bisa *filter* pencarian poliklinik berdasarkan nama poliklinik yang dicari.



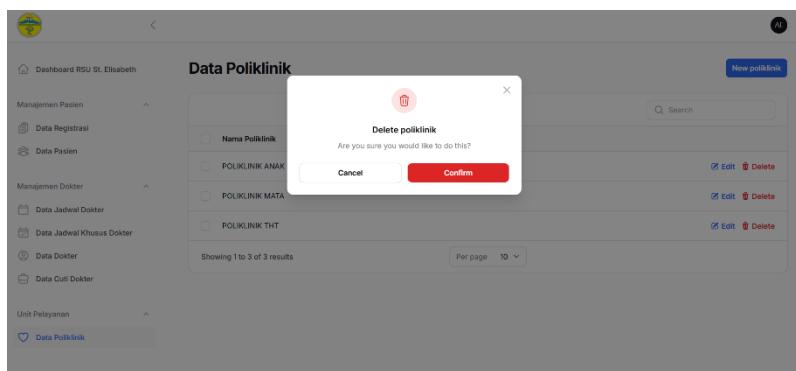
Gambar 76. Tampilan Halaman *Admin Create* Poliklinik

Tampilan ini merupakan tampilan *create* poliklinik. *Admin* akan mengisi data poliklinik pada *field* yang tersedia, lalu dapat menambahkan poliklinik dengan menekan tombol *create* berwarna biru.



Gambar 77. Tampilan Halaman *Admin* Ubah Poliklinik

Tampilan ini merupakan tampilan ketika *admin* ingin mengubah data poliklinik. *Admin* akan dapat mengubah data poliklinik yang diinginkan.



Gambar 78. Tampilan Halaman *Admin* Hapus Poliklinik

Tampilan ini merupakan tampilan ketika *admin* akan menghapus salah satu data pada poliklinik. *Admin* akan mendapatkan peringatan apakah *admin* yakin untuk menghapus data poliklinik tersebut, jika iya, maka bisa tekan tombol *confirm* berwarna merah.

4.3 Sprint Backlog

Sprint Backlog untuk sistem registrasi *online* RSU St. Elisabeth Purwokerto merupakan daftar tugas yang direncanakan untuk diselesaikan dalam satu *sprint*. *Backlog* ini disusun berdasarkan prioritas dan kompleksitas fitur yang telah diidentifikasi dalam *Product Backlog*. Dalam konteks proyek ini, *Sprint Backlog* mencakup berbagai tugas

pengembangan, mulai dari implementasi fitur dasar *admin* hingga fitur-fitur lanjutan untuk pasien.

4.3.1 Item Backlog

Item Backlog dalam proyek ini merupakan komponen individual dari *Sprint Backlog*, yang mewakili fitur atau fungsi spesifik yang perlu dikembangkan. Setiap *item* diformulasikan dalam format "Saya ingin [fitur] sehingga [manfaat]", yang membantu dalam memahami perspektif pengguna dan tujuan dari setiap fitur. Setiap *item* juga dilengkapi dengan informasi prioritas, *sprint* di mana *item* tersebut direncanakan untuk diselesaikan, dan status penyelesaiannya, memungkinkan pelacakan progres yang efektif selama proses pengembangan.

Tabel 5. Item Backlog

No	Saya ingin bisa...	Sehingga...	Prioritas	Sprint	Status
1	Mengetahui hal yang perlu disiapkan sebelum memulai	Saya siap menggunakan sistem	Tinggi	1	Selesai
2	Melakukan <i>login</i> sebagai <i>admin</i>	Saya dapat mengakses fitur <i>admin</i>	Tinggi	1	Selesai
3	Melihat pesan kesalahan saat <i>login admin</i> gagal	Saya dapat memperbaiki data <i>input login admin</i> saya	Tinggi	1	Selesai
4	Melihat <i>dashboard</i> dengan <i>widget</i> waktu dan informasi	Saya dapat melihat waktu sekarang dan ringkasan data penting rumah sakit	Sedang	1	Selesai
5	Mengelola data registrasi pasien	Saya dapat melihat, menghapus data registrasi, dan mengubah status registrasi	Tinggi	1	Selesai

6	Mengelola data pasien	Saya dapat menambah, mengubah, menghapus, dan melihat data pasien	Tinggi	1	Selesai
7	Mengelola data dokter	Saya dapat menambah, mengubah, menghapus, dan melihat data dokter	Tinggi	1	Selesai
8	Mengelola data poliklinik	Saya dapat menambah, mengubah, menghapus, dan melihat data poliklinik	Tinggi	2	Selesai
9	Mengelola data jadwal dokter	Saya dapat menambah, mengubah, menghapus, dan melihat data jadwal dokter	Tinggi	2	Selesai
10	Mengelola data jadwal khusus dokter	Saya dapat menambah, mengubah, menghapus, dan melihat data jadwal khusus dokter	Sedang	2	Selesai
11	Mengelola data cuti dokter	Saya dapat menambah, mengubah, menghapus, dan melihat data cuti dokter	Sedang	2	Selesai
12	Melihat peringatan untuk <i>email</i> dan nomor kartu yang sudah terdaftar	Saya dapat memastikan keunikan data pasien	Sedang	2	Selesai
13	Menggunakan fitur pencarian dan <i>filter</i> untuk data	Saya dapat dengan mudah menemukan	Sedang	2	Selesai

informasi yang dibutuhkan					
14	Melihat peringatan untuk <i>input</i> jadwal dokter, jadwal khusus dokter, dan jadwal cuti dokter yang tidak valid	Saya dapat menghindari kesalahan dalam penjadwalan	Sedang	2	Selesai
15	Melihat tabel informasi sisa kuota dokter	Saya dapat melihat informasi ketersediaan dokter untuk hari berikutnya	Tinggi	3	Selesai
16	Melihat label pemberitahuan untuk ketersediaan dokter dan kuota	Saya dapat dengan cepat mengidentifikasi ketersediaan layanan	Sedang	3	Selesai
17	Melihat perubahan jadwal otomatis untuk jadwal khusus dan saat dokter cuti	Saya dapat memastikan informasi jadwal selalu akurat	Sedang	3	Selesai
18	Melihat tabel jadwal praktik dokter untuk satu minggu	Saya dapat melihat jadwal praktik dokter dari Senin hingga Minggu	Tinggi	3	Selesai
19	Menggunakan <i>navigation bar</i>	Saya dapat dengan mudah mengakses fitur-fitur utama sistem	Rendah	3	Selesai
20	Melihat <i>footer</i> dengan informasi rumah sakit	Saya dapat mengakses informasi kontak dan media sosial rumah sakit	Rendah	3	Selesai
21	Memilih antara pasien lama atau baru saat registrasi	Saya dapat menggunakan alur pendaftaran yang sesuai	Sedang	3	Selesai

22	Melihat <i>progress bar</i> saat melakukan pendaftaran	Saya dapat mengetahui tahapan proses pendaftaran	Rendah	3	Selesai
23	Melakukan registrasi sebagai pasien lama atau baru	Saya dapat mendaftar untuk layanan rumah sakit secara <i>online</i>	Tinggi	4	Selesai
24	Memilih tanggal kunjungan dalam rentang 7 hari	Saya dapat mendaftar untuk kunjungan dalam waktu dekat	Tinggi	4	Selesai
25	Mendapatkan kode booking dan <i>QR code</i>	Saya dapat memiliki bukti pendaftaran yang valid	Tinggi	4	Selesai
26	Mengunduh bukti pendaftaran dalam format PDF	Saya dapat menyimpan dan menunjukkan bukti pendaftaran saya	Tinggi	4	Selesai
27	Mengakses sistem melalui perangkat <i>mobile</i>	Saya dapat mendaftar kapan saja dan di mana saja	Rendah	4	Selesai

4.3.2 Pembagian *Sprint*

Berdasarkan prioritas dan kompleksitas fitur, pekerjaan dibagi menjadi empat sprint:

- Sprint 1:* Fokus pada fitur-fitur dasar admin dan manajemen data utama. (*Item backlog:* 1 – 7)
- Sprint 2:* Fokus pada manajemen jadwal, poliklinik, dan validasi *input*. (*Item backlog:* 8 – 14)
- Sprint 3:* Fokus pada fitur-fitur untuk pasien, tampilan jadwal, dan ketersediaan layanan. (*Item backlog:* 15 – 22)

- d. *Sprint 4*: Fokus pada fitur-fitur lanjutan untuk pasien, finalisasi sistem, dan pengalaman pengguna. (*Item backlog*: 23 – 27)

4.4 Sprint

Sprint pada proyek pengembangan sistem registrasi *online* RSU St. Elisabeth Purwokerto mencakup serangkaian tahapan kritis yang meliputi pemilihan teknologi, rincian pengembangan, dan implementasi *package*. Tahap ini dimulai dengan pemilihan *framework* Laravel dan *database* MySQL berdasarkan kebutuhan spesifik proyek. Selanjutnya, rincian pengembangan diuraikan melalui implementasi berbagai fitur utama seperti manajemen poliklinik dan dokter, validasi data pasien, *generate booking code*, penyimpanan data registrasi, pembuatan bukti pendaftaran PDF, serta penampilan jadwal dan info kuota.

4.4.1 Pemilihan Teknologi

Pemilihan teknologi untuk sistem registrasi *online* RSU St. Elisabeth Purwokerto didasarkan pada kebutuhan spesifik proyek, efisiensi pengembangan, dan kemampuan teknologi dalam memenuhi persyaratan sistem. Teknologi yang dipilih harus mendukung pengembangan sistem yang responsif, aman, dan mudah dikelola, serta memungkinkan integrasi yang mulus dengan infrastruktur yang ada di rumah sakit. Pertimbangan juga diberikan pada skalabilitas teknologi untuk mengakomodasi pertumbuhan dan perubahan kebutuhan di masa depan.

a. Pemilihan *Framework*

Untuk pengembangan sistem registrasi *online* RSU St. Elisabeth Purwokerto, *framework* Laravel dipilih sebagai basis utama. Berikut adalah poin-poin *plus* mengapa Laravel cocok untuk proyek ini:

- Struktur MVC untuk organisasi kode yang baik
- Eloquent ORM untuk manajemen *database* yang mudah
- Fitur keamanan bawaan (CSRF, XSS protection)
- Artisan CLI untuk otomatisasi tugas

- *Blade Templating Engine* untuk UI dinamis
 - Ekosistem yang kaya dengan *package* tambahan
 - Integrasi Filament dan Livewire untuk sistem *admin* interaktif
 - Dokumentasi komprehensif
 - Komunitas besar untuk dukungan
 - *Database Seeding* untuk pengujian dan setup awal
- b. Integrasi *Database*

Untuk integrasi *database*, MySQL dipilih sebagai sistem manajemen *database* untuk sistem registrasi *online* RSU St. Elisabeth Purwokerto. Berikut adalah poin-poin *plus* dari penggunaan MySQL:

- Kompatibilitas tinggi dengan Laravel
- Performa tinggi untuk menangani data besar
- Fitur keamanan kuat
- Skalabilitas baik
- Fitur *backup* dan *recovery*
- Mendukung berbagai tipe data
- Kompatibilitas lintas platform
- Gratis dan *open source*

4.4.2 Rincian Pengembangan

Rincian pengembangan berisi potongan kode fungsi yang diambil dari beberapa *controller* untuk fitur utama sistem registrasi *online* ini yang meliputi:

- a. Opsi Poliklinik dan Dokter untuk Pasien Registrasi

Kode ini menangani konversi hari ke format Carbon, mengambil dan memproses jadwal khusus dan reguler dokter dengan mempertimbangkan kuota dan cuti, serta menyiapkan data untuk ditampilkan pada halaman registrasi pasien.

```

public function translateDayToEnglish($dayInIndonesian)
{
    $days = [
        'Minggu' => Carbon::SUNDAY,
        'Senin' => Carbon::MONDAY,
        'Selasa' => Carbon::TUESDAY,
        'Rabu' => Carbon::WEDNESDAY,
        'Kamis' => Carbon::THURSDAY,
        'Jumat' => Carbon::FRIDAY,
        'Sabtu' => Carbon::SATURDAY,
    ];

    return $days[$dayInIndonesian] ?? null;
}

public function index()
{
    $polikliniks = Poliklinik::all();
    $today = Carbon::now();
    $tomorrow = Carbon::tomorrow();

    // Fetch special schedules with quota > 0
    $jadwalKhusus =
        JadwalKhususDokter::with(['dokter', 'dokter.cutiDokter'])
            ->whereDate('tanggal', '>=', $today)
            ->where('kuota', '>', 0)
            ->get();

    foreach ($jadwalKhusus as $value) {
        $value->kenaCuti = false;
    }
}

```

```

        foreach($value->dokter->cutiDokter as
$value2){
    if(Carbon::parse($value->tanggal)-
>startOfDay() <= Carbon::parse($value2->tanggal_selesai)-
>startOfDay() && Carbon::parse($value->tanggal)-
>startOfDay() >= Carbon::parse($value2->tanggal_mulai)-
>startOfDay() ){
        $value->kenaCuti = true;
    }
}
}

$jadwalKhusus = $jadwalKhusus-
>where('kenaCuti', false)->values();

// Get all special schedules (including zero
quota) for blocking regular schedules of the same doctor

$allJadwalKhusus =
JadwalKhususDokter::whereDate('tanggal', '>=', $today)-
>get();

$specialSchedules = $allJadwalKhusus-
>groupBy(function ($item) {
    return $item->tanggal . '-' . $item-
>id_dokter;
})->map(function ($group) {
    return $group->first();
});

// Fetch regular schedules

```

```

$jadwal = JadwalDokter::with(['poliklinik',
'dokter', 'dokter.cutiDokter'])
->where('kuota', '>', 0)
->get();

foreach ($jadwal as $value) {
    $value->tanggal = Carbon::today()-
>next($this->translateDayToEnglish($value->hari));
    $value->kenaCuti = false;

    foreach($value->dokter->cutiDokter as
$value2){
        if($value->tanggal <=
Carbon::parse($value2->tanggal_selesai)->startOfDay() &&
$value->tanggal >= Carbon::parse($value2->tanggal_mulai)-
>startOfDay() ){
            $value->kenaCuti = true;
        }
    }
}

$jadwal = $jadwal->where('kenaCuti', false)-
>values();

return view('regis.index', compact('polikliniks',
'jadwal', 'jadwalKhusus', 'specialSchedules'));
}

```

b. Validasi Data Pasien Lama

Kode ini memeriksa keberadaan pasien dalam *database* berdasarkan nomor rekam medis dan tanggal lahir. Jika ditemukan,

fungsi mengembalikan data pasien dalam format JSON; jika tidak, mengembalikan indikasi bahwa pasien tidak ditemukan.

```
public function checkPatient(Request $request)
{
    $patient = Pasien::where('nomor_rm', $request->nomor_rm)
        ->where('tanggal_lahir', $request->tanggal_lahir)
        ->first();

    if ($patient) {
        return response()->json([
            'exists' => true,
            'id_pasien' => $patient->id_pasien,
            'nama_pasien' => $patient->nama_pasien,
            'tempat_lahir' => $patient->tempat_lahir,
            'jenis_kelamin' => $patient->jenis_kelamin,
            'alamat' => $patient->alamat,
            'nomor_telepon' => $patient->nomor_telepon,
            'email' => $patient->email,
            'nomor_kartu' => $patient->nomor_kartu,
            'tanggal_lahir' => $patient->tanggal_lahir,
        ]);
    } else {
        return response()->json(['exists' => false]);
    }
}
```

c. *Generate Booking Code*

Kode ini menghasilkan kode booking unik dengan format dua huruf, tiga angka, dan satu huruf. Fungsi memeriksa keunikan kode yang dihasilkan dan akan terus menghasilkan kode baru hingga mendapatkan kode yang belum ada dalam *database*.

```

private function isCodeExists($code)
{
    return Registrasi::where('kode_booking', $code)->exists();
}

public function generateCustomCode()
{
    $letters = 'ABCDEFGHIJKLMNPQRSTUVWXYZ';
    $numbers = '0123456789';

    do {
        $firstTwoLetters = $letters[random_int(0,
strlen($letters) - 1)] . $letters[random_int(0,
strlen($letters) - 1)];
        $threeNumbers = '';
        for ($i = 0; $i < 3; $i++) {
            $threeNumbers .= $numbers[random_int(0,
strlen($numbers) - 1)];
        }
        $lastLetter = $letters[random_int(0,
strlen($letters) - 1)];
    }
}
```

```

    $customCode = $firstTwoLetters .
    $threeNumbers . $lastLetter;

} while ($this->isCodeExists($customCode));

return $customCode;
}

```

d. *Store Data Registrasi Pasien*

Kode ini menangani proses penyimpanan data registrasi pasien. Fungsi ini melakukan validasi *input*, memeriksa dan memperbarui jadwal dokter (khusus atau reguler), menyimpan data pasien baru jika diperlukan, dan mencatat registrasi kunjungan. Setelah penyimpanan berhasil, fungsi mengembalikan *view* dengan informasi kode *booking*, tanggal kunjungan, dan nomor urut.

```

public function store(Request $request)
{
    $validatedData = $request->validate([
        'nama_pasien' => 'required|string',
        'tempat_lahir' => 'required|string',
        'tanggal_lahir' => 'required|date',
        'jenis_kelamin' => 'required|in:Laki-
laki,Perempuan',
        'alamat' => 'nullable|string',
        'nomor_telepon' => 'required|string',
        'email' => 'required|email',
        'nomor_kartu' => 'nullable|string',
        'tanggal_kunjungan' => 'required|date',
        'poliklinik' => 'required',
    ]);
}

```

```

        'dokter' => 'required',
    ]);

$tanggalKunjungan =
Carbon::parse($validatedData['tanggal_kunjungan']);
$hariKunjungan = $tanggalKunjungan->locale('id')-
>dayName;

$jamMulai = $request->jam_mulai;
$jamSelesai = $request->jam_selesai;

$id_jadwal =null;
$id_jadwal_khusus = null;

$jadwalKhusus =
JadwalKhususDokter::where('id_dokter',
$validatedData['dokter'])
->whereDate('tanggal', $tanggalKunjungan)
->where('jam_mulai', $jamMulai)
->where('jam_selesai', $jamSelesai)
->first();

if ($jadwalKhusus) {
    $jadwalKhusus->kuota = $jadwalKhusus->kuota -
1;
    $jadwalKhusus->save();
    $no_urut =
Registrasi::where('id_jadwal_khusus_dokter',
$jadwalKhusus->id_jadwal_khusus_dokter)->count())+1;
}

```

```

        $id_jadwal_khusus = $jadwalKhusus-
>id_jadwal_khusus_dokter;

    } else {

        $jadwal = JadwalDokter::where('id_dokter',
$validatedData['dokter'])

            ->where('hari', $hariKunjungan)
            ->where('jam_mulai', $jamMulai)
            ->where('jam_selesai', $jamSelesai)
            ->first();

        if ($jadwal) {

            $jadwal->kuota = $jadwal->kuota - 1;
            $jadwal->save();
            $no_urut =
Registrasi::where('id_jadwal_dokter', $jadwal-
>id_jadwal_dokter)->count())+1;
            $id_jadwal = $jadwal->id_jadwal_dokter;
        }

    }

    $kode = $this->generateCustomCode();

    if ($request->id_pasien === null) {

        $pasien = new Pasien();
        $pasien->nomor_rm = null;
        $pasien->nama_pasien =
$validatedData['nama_pasien'];
        $pasien->tempat_lahir =
$validatedData['tempat_lahir'];
    }
}

```

```

$pasien->tanggal_lahir =
$validatedData['tanggal_lahir'];

$pasien->jenis_kelamin =
$validatedData['jenis_kelamin'];

$pasien->alamat = $validatedData['alamat'];

$pasien->nomor_telepon =
$validatedData['nomor_telepon'];

$pasien->email = $validatedData['email'];

$pasien->nomor_kartu =
$validatedData['nomor_kartu'];

$pasien->save();

$registrasi = new Registrasi();

$registrasi->id_pasien = $pasien->id_pasien;
$registrasi->tanggal_kunjungan =
$validatedData['tanggal_kunjungan'];
$registrasi->id_poliklinik =
$validatedData['poliklinik'];
$registrasi->id_dokter =
$validatedData['dokter'];
$registrasi->kode_booking = $kode;
$registrasi->jam_mulai = $request->jam_mulai;
$registrasi->jam_selesai = $request-
>jam_selesai;
$registrasi->nomor_urut = $no_urut;
$registrasi->id_jadwal_dokter = $id_jadwal;
$registrasi->id_jadwal_khusus_dokter =
$id_jadwal_khusus;
$registrasi->save();
} else {

```

```

$registrasi = new Registrasi();

$registrasi->id_pasien = $request->id_pasien;
$registrasi->tanggal_kunjungan =
$validatedData['tanggal_kunjungan'];

$registrasi->id_poliklinik =
$validatedData['poliklinik'];

$registrasi->id_dokter =
$validatedData['dokter'];

$registrasi->kode_booking = $kode;
$registrasi->jam_mulai = $request->jam_mulai;
$registrasi->jam_selesai = $request-
>jam_selesai;

$registrasi->nomor_urut = $no_urut;
$registrasi->id_jadwal_dokter = $id_jadwal;
$registrasi->id_jadwal_khusus_dokter =
$id_jadwal_khusus;

$registrasi->save();
}

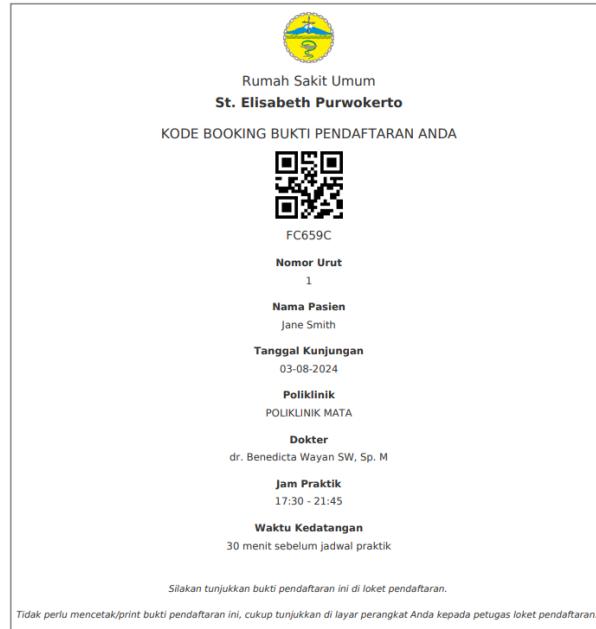
$besok = Carbon::parse($request-
>tanggal_kunjungan)->format('d-m-Y');

return view('regis.index', compact('kode',
'besok', 'no_urut'));
}

```

e. *Download PDF*

Kode ini menghasilkan dan mengunduh bukti pendaftaran dalam format PDF yang berisi detail informasi pendaftaran. Detail bukti pendaftaran dapat dilihat pada Gambar 79.



Gambar 79. Bukti Pendaftaran PDF

Fungsi ini mengambil data registrasi, memeriksa jadwal dokter, menyiapkan logo dan QR *code* dengan menggunakan *package* ‘composer require milon/barcode’, kemudian me-*render template* HTML dengan data tersebut. Akhirnya, HTML dikonversi menjadi PDF dengan menggunakan *package* ‘composer require barryvdh/laravel-dompdf’ dan diunduh dengan nama “bukti-pendaftaran-rsu-elisabeth-purwokerto.pdf”.

```
public function download_pdf($kode, $tanggal)
{
    $registrasi = Registrasi::with(['pasien',
    'poliklinik', 'dokter'])
        ->where('kode_booking', $kode)
        ->first();

    if (!$registrasi) {
        abort(404, 'Registration not found');
    }
}
```

```

    }

    $tanggalKunjungan = Carbon::parse($registrasi-
>tanggal_kunjungan);

    $hariKunjungan = $tanggalKunjungan->locale('id')-
>dayName;

    // Cek apakah ada jadwal khusus

    $jadwalKhusus =
JadwalKhususDokter::where('id_dokter', $registrasi-
>id_dokter)
    ->whereDate('tanggal', $tanggalKunjungan)
    ->first();

    if ($jadwalKhusus) {
        $jadwal = $jadwalKhusus;
    } else {
        $jadwal = JadwalDokter::where('id_dokter',
$registrasi->id_dokter)
    ->where('hari', $hariKunjungan)
    ->first();
    }

    $jamMulai = $registrasi->jam_mulai;
    $jamSelesai = $registrasi->jam_selesai;
    $no_urut = $registrasi->nomor_urut;

    $logoPath = public_path('logo.png');
    $logoData =
base64_encode(file_get_contents($logoPath));

```

```

$dns2d = new DNS2D();

$qrcode = $dns2d->getBarcodePNG($kode, 'QRCODE',
5, 5);

$html = view('pdf.bukti-pendaftaran',
compact('registrasi', 'kode', 'tanggal', 'jadwal',
'logoData', 'qrcode', 'jamMulai',
'jamSelesai','no_urut'))->render();

$pdf = PDF::loadHtml($html);

$pdf->setPaper('A4', 'portrait');

return $pdf->download('bukti-pendaftaran-rsu-
elisabeth-purwokerto.pdf');

}

```

f. Validasi *Email* dan Nomor Kartu Pasien

Kode ini terdiri dari dua fungsi: *emailCheck* dan *noKartuCheck*. Keduanya memeriksa keberadaan *email* atau nomor kartu dalam database pasien. Jika ditemukan, fungsi mengembalikan respons JSON dengan nilai *true*, jika tidak ditemukan, mengembalikan *false*. Fungsi-fungsi ini berguna untuk validasi data pasien secara *real-time* pada sisi klien.

```

public function emailCheck($email){
    $email_check = Pasien::where('email', $email)-
>first();
    if($email_check){
        return response()->json(['data' => true]);
    }
}

```

```

        }else{
            return response()->json(['data' => false]);
        }
    }

public function noKartuCheck($noKartu){
    $no_kartu_check = Pasien::where('nomor_kartu',
$noKartu)->first();
    if($no_kartu_check){
        return response()->json(['data' => true]);
    } else{
        return response()->json(['data' => false]);
    }
}

```

g. Menampilkan Jadwal

Kode ini menampilkan jadwal dokter untuk seminggu. Dengan mengambil data poliklinik beserta dokter dan jadwalnya (termasuk jadwal reguler, cuti, dan khusus) untuk rentang waktu dari awal hingga akhir minggu saat ini. Hasilnya ditampilkan dalam *view* 'jadwal.index' bersama dengan data rentang waktu tersebut.

```

public function index()
{
    $startOfWeek = now()->startOfWeek();
    $endOfWeek = now()->endOfWeek();

    $polikliniks = Poliklinik::with(['dokter' =>
function($query) {
    $query->has('jadwalDokters');
}

```

```

        }, 'dokter.jadwalDokters', 'dokter.cutiDokters',
'dokter.jadwalKhususDokters' => function($query) use
($startOfWeek, $endOfWeek) {
    $query->whereBetween('tanggal',
[$startOfWeek, $endOfWeek]);
}])->get();

return view('jadwal.index',
compact('polikliniks', 'startOfWeek', 'endOfWeek'));
}

```

h. Menampilkan Info Kuota Besok

Kode ini menampilkan informasi kuota dokter untuk besok. Dengan mengambil data poliklinik beserta dokter yang memiliki jadwal (baik jadwal khusus atau reguler) untuk hari berikutnya. Kode ini juga memperhitungkan cuti dokter, menghapus dokter yang sedang cuti dari daftar. Untuk setiap dokter, ditentukan jadwalnya (khusus atau reguler) yang berlaku besok. Hasilnya ditampilkan dalam view 'info.index' bersama dengan informasi tanggal dan nama hari besok.

```

public function indexinfo()
{
    Carbon::setLocale('id');

    $today = now();

    $tomorrow = now()->addDay();
    $dayName = $tomorrow->translatedFormat('1');
    $dayOfWeek = $tomorrow->dayOfWeekIso;
}

```

```

$polikliniks = Poliklinik::with(['dokter' =>
function($query) use ($tomorrow, $dayOfWeek) {
    $query->whereHas('jadwalKhususDokters',
function($subQuery) use ($tomorrow) {
    $subQuery->where('tanggal', $tomorrow-
>toDateString());
})->orWhereHas('jadwalDokters',
function($subQuery) use ($dayOfWeek) {
    $subQuery->where('hari', $dayOfWeek);
});
}, 'dokter.jadwalKhususDokters' =>
function($query) use ($tomorrow) {
    $query->where('tanggal', $tomorrow-
>toDateString());
}, 'dokter.jadwalDokters' => function($query) use
($dayOfWeek) {
    $query->where('hari', $dayOfWeek);
}, 'dokter.cutiDokters'])->get();

$polikliniks->each(function ($poliklinik) use
($today, $tomorrow) {
    $poliklinik->dokter = $poliklinik->dokter-
>filter(function ($dokter) use ($today, $tomorrow) {
        $onLeave = $dokter->cutiDokters-
>where('tanggal_mulai', '<=', $tomorrow)
        -
        >where('tanggal_selesai', '>=', $today)
        -
        >isNotEmpty();
        return !$onLeave;
    });
});

```

```

    });

    $poliklinik->dokter->each(function ($dokter)
use ($tomorrow) {
    $dokter->effectiveSchedule = $dokter-
>jadwalKhususDokters->where('tanggal', $tomorrow-
>toDateString())->isNotEmpty()
    ? $dokter->jadwalKhususDokters-
>where('tanggal', $tomorrow->toDateString())
        : $dokter->jadwalDokters;
});

return view('info.index', compact('polikliniks',
'tomorrow', 'dayName')));
}

```

4.4.3 Package Filament

Sistem ini mengimplementasikan integrasi *package* Filament ke dalam aplikasi Laravel. Hal ini dibuktikan dengan adanya kelas AdminPanelProvider yang mengkonfigurasi *panel admin* Filament. Konfigurasi tersebut mencakup pengaturan tema, logo, rute, sumber daya, halaman, dan *widget* kustom. Penggunaan Filament memungkinkan pembangunan antarmuka *admin* yang *robust* dan fleksibel, memanfaatkan berbagai fitur bawaan Filament seperti manajemen sumber daya, *widget*, dan sistem autentikasi.

```
<?php
```

```
namespace App\Providers\Filament;

use Filament\Http\Middleware\Authenticate;
use Filament\Http\Middleware\DisableBladeIconComponents;
use Filament\Http\Middleware\DispatchServingFilamentEvent;
use Filament\Pages;
use Filament\Panel;
use Filament\PanelProvider;
use Filament\Support\Colors\Color;
use Filament\Widgets;
use Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse;
use Illuminate\Cookie\Middleware\EncryptCookies;
use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken;
use Illuminate\Routing\Middleware\SubstituteBindings;
use Illuminate\Session\Middleware\AuthenticateSession;
use Illuminate\Session\Middleware\StartSession;
use Illuminate\View\Middleware\ShareErrorsFromSession;

class AdminPanelProvider extends PanelProvider
{
    public function panel(Panel $panel): Panel
    {
        return $panel
            ->default()
            ->sidebarCollapsibleOnDesktop(true)
            ->id('admin')
            ->path('admin')
            ->login()
            ->brandLogo(asset('logo.png'))
            ->brandLogoHeight('3rem')
```

```

->colors([
    'primary' => Color::Blue,
])

->discoverResources(in:
app_path('Filament/Resources'), for:
'App\\Filament\\Resources')

->discoverPages(in: app_path('Filament/Pages'),
for: 'App\\Filament\\Pages')

->pages([
    // Pages\Dashboard::class,
])

->discoverWidgets(in: app_path('Filament/Widgets'),
for: 'App\\Filament\\Widgets')

->widgets([
    Widgets\AccountWidget::class,
    \App\Filament\Widgets\WeeklyRegistrationChart::
class,
    \App\Filament\Widgets\LiveClockWidget::class,
])

->middleware([
    EncryptCookies::class,
    AddQueuedCookiesToResponse::class,
    StartSession::class,
    AuthenticateSession::class,
    ShareErrorsFromSession::class,
    VerifyCsrfToken::class,
    SubstituteBindings::class,
    DisableBladeIconComponents::class,
    DispatchServingFilamentEvent::class,
])

])

```

```

    ->authMiddleware([
        Authenticate::class,
    ]);
}

}

```

4.5 Finished Work

Bagian ini menjabarkan tahap-tahap akhir dari proyek pengembangan sistem pendaftaran *online* untuk RSU St. Elisabeth Purwokerto. Pertama, proses pengujian sistem yang komprehensif menggunakan metode *Black Box* dan kedua, prosedur penyerahan kode sumber (*source code*) kepada tim IT rumah sakit. Pengujian sistem dilakukan secara menyeluruh untuk memastikan bahwa seluruh fungsionalitas berjalan sesuai dengan spesifikasi yang telah ditetapkan, sementara penyerahan kode sumber merupakan langkah final yang memungkinkan tim IT rumah sakit untuk mengimplementasikan dan mengelola sistem secara mandiri di masa mendatang.

4.5.1 Pengujian Sistem

Pengujian sistem merupakan tahap yang penting dalam pengembangan perangkat lunak untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Dalam penelitian ini, pengujian dilakukan menggunakan metode *Black Box*, yang berfokus pada pengujian fungsionalitas sistem tanpa melihat struktur kode *internal*. Proses ini melibatkan serangkaian tes yang dirancang untuk memverifikasi kesesuaian antara hasil yang diharapkan dan hasil aktual dari berbagai fitur sistem. Pengujian *Black Box* ini mencakup berbagai skenario penggunaan, mulai dari proses *login*, manajemen data, hingga fungsionalitas spesifik seperti penanganan jadwal dokter dan kuota pasien. Hasil pengujian kemudian didokumentasikan secara rinci dalam bentuk tabel, seperti yang dapat dilihat pada Tabel 6 berikut.

Tabel 6. Pengujian Sistem

Kode SRS	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
[SRS-001]	Mencoba <i>login</i> sebagai <i>admin</i> dengan nama pengguna dan kata sandi yang benar	<i>Admin</i> berhasil masuk dan mengakses sistem	<i>Admin</i> berhasil masuk dan mengakses sistem	Valid
[SRS-002]	Mencoba <i>login</i> sebagai <i>admin</i> dengan nama pengguna atau kata sandi yang tidak valid	Sistem menampilkan pesan kesalahan ketika nama pengguna atau kata sandi tidak valid	Sistem menampilkan pesan kesalahan ketika nama pengguna atau kata sandi tidak valid	Valid
[SRS-003]	Mengakses <i>dashboard</i> setelah <i>login</i>	<i>Dashboard</i> ditampilkan dengan <i>widget</i> informatif yang menunjukkan waktu sekarang dan ringkasan data penting rumah sakit	<i>Dashboard</i> ditampilkan dengan <i>widget</i> informatif yang menunjukkan waktu sekarang dan ringkasan data penting rumah sakit	Valid
[SRS-004]	Melihat, menghapus data registrasi, dan mengubah status registrasi, serta memfilter data registrasi	Semua operasi berhasil dilakukan pada data registrasi	Semua operasi berhasil dilakukan pada data registrasi	Valid
[SRS-005]	Menambah, mengubah, menghapus, melihat, dan memfilter data pasien	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data pasien	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data pasien	Valid
[SRS-006]	Menambah, mengubah, menghapus, melihat, dan	Semua operasi CRUD dan <i>filtering</i> berhasil	Semua operasi CRUD dan <i>filtering</i> berhasil	Valid

	memfilter data dokter	dilakukan pada data dokter	dilakukan pada data dokter	
[SRS-007]	Menambah, mengubah, menghapus, melihat, dan memfilter data poliklinik	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data poliklinik	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data poliklinik	Valid
[SRS-008]	Menambah banyak jadwal dokter untuk satu dokter, mengubah, menghapus, melihat, dan memfilter data jadwal dokter	Semua operasi berhasil dilakukan pada data jadwal dokter	Semua operasi berhasil dilakukan pada data jadwal dokter	Valid
[SRS-009]	Menambah banyak jadwal khusus dokter untuk satu dokter, mengubah, menghapus, melihat, dan memfilter data jadwal khusus dokter	Semua operasi berhasil dilakukan pada data jadwal khusus dokter	Semua operasi berhasil dilakukan pada data jadwal khusus dokter	Valid
[SRS-010]	Menambah, mengubah, menghapus, melihat, dan memfilter data cuti dokter	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data cuti dokter	Semua operasi CRUD dan <i>filtering</i> berhasil dilakukan pada data cuti dokter	Valid
[SRS-011]	Memasukkan <i>email</i> dan nomor kartu yang sudah terdaftar	Sistem menampilkan pesan peringatan	Sistem menampilkan pesan peringatan	Valid
[SRS-012]	Memasukkan jam selesai lebih awal dari jam mulai pada jadwal dokter dan jadwal khusus dokter	Sistem menampilkan pesan peringatan	Sistem menampilkan pesan peringatan	Valid

[SRS-013]	Memasukkan tanggal selesai lebih awal dari tanggal mulai pada cuti dokter	Sistem menampilkan pesan peringatan	Sistem menampilkan pesan peringatan	Valid
[SRS-014]	Melihat tabel informasi sisa kuota dokter untuk hari berikutnya	Tabel ditampilkan dengan benar dan informasi kuota akurat	Tabel ditampilkan dengan benar dan informasi kuota akurat	Valid
[SRS-015]	Memeriksa tampilan ketika tidak ada dokter yang tersedia untuk suatu poliklinik	Label pemberitahuan berwarna merah muncul untuk poliklinik tersebut pada tabel jadwal dan kuota	Label pemberitahuan berwarna merah muncul untuk poliklinik tersebut pada tabel jadwal dan kuota	Valid
[SRS-016]	Memeriksa tampilan ketika kuota seorang dokter penuh	Teks "Kuota Penuh" muncul untuk dokter tersebut pada tabel informasi kuota	Teks "Kuota Penuh" muncul untuk dokter tersebut pada tabel informasi kuota	Valid
[SRS-017]	Memeriksa tampilan ketika semua dokter dalam suatu poliklinik kuotanya penuh	Label pemberitahuan berwarna kuning muncul untuk poliklinik tersebut pada tabel informasi kuota	Label pemberitahuan berwarna kuning muncul untuk poliklinik tersebut pada tabel informasi kuota	Valid
[SRS-018]	Memeriksa perilaku sistem ketika seorang dokter mengambil cuti	Nama dokter dihapus dari tabel jadwal, tabel kuota, dan pilihan pada formulir registrasi	Nama dokter dihapus dari tabel jadwal, tabel kuota, dan pilihan pada formulir registrasi	Valid
[SRS-019]	Memeriksa perilaku sistem ketika seorang dokter memiliki jadwal khusus	Jadwal khusus menggantikan jadwal reguler di tabel jadwal, tabel kuota,	Jadwal khusus menggantikan jadwal reguler di tabel jadwal, tabel kuota,	Valid

		dan pilihan pada formulir registrasi	dan pilihan pada formulir registrasi	
[SRS-020]	Melihat tabel jadwal praktik dokter untuk satu minggu penuh	Tabel ditampilkan dengan benar dari Senin hingga Minggu	Tabel ditampilkan dengan benar dari Senin hingga Minggu	Valid
[SRS-021]	Memeriksa <i>navigation bar</i> untuk pasien	<i>Navigation bar</i> menampilkan logo, beranda, jadwal praktik, dan info kuota	<i>Navigation bar</i> menampilkan logo, beranda, jadwal praktik, dan info kuota	Valid
[SRS-022]	Memeriksa konten <i>footer</i>	<i>Footer</i> menampilkan informasi rumah sakit dan tautan media sosial	<i>Footer</i> menampilkan informasi rumah sakit dan tautan media sosial	Valid
[SRS-023]	Memeriksa ketersediaan tombol untuk pasien baru dan lama	Dua tombol ditampilkan untuk pemilihan jenis pasien	Dua tombol ditampilkan untuk pemilihan jenis pasien	Valid
[SRS-024]	Memeriksa <i>progress bar</i> selama proses pendaftaran	<i>Progress bar</i> tiga langkah ditampilkan dan diperbarui dengan benar	<i>Progress bar</i> tiga langkah ditampilkan dan diperbarui dengan benar	Valid
[SRS-025]	Memilih opsi pasien lama	Sistem menampilkan formulir untuk pasien lama	Sistem menampilkan formulir untuk pasien lama	Valid
[SRS-026]	Memasukkan data yang salah pada formulir pasien lama	Sistem menampilkan pesan peringatan	Sistem menampilkan pesan peringatan	Valid
[SRS-027]	Memasukkan data yang benar pada formulir pasien lama	Sistem menampilkan data pasien lama dan formulir registrasi	Sistem menampilkan data pasien lama dan formulir registrasi	Valid
[SRS-028]	Memilih opsi pasien baru	Sistem menampilkan formulir data	Sistem menampilkan formulir data	Valid

		pasien baru dan registrasi	pasien baru dan registrasi	
[SRS-029]	Memeriksa rentang tanggal kunjungan untuk pendaftaran janji temu	Sistem hanya mengizinkan pemesanan untuk tujuh hari ke depan	Sistem hanya mengizinkan pemesanan untuk tujuh hari ke depan	Valid
[SRS-030]	Menyelesaikan pendaftaran pasien	Sistem menghasilkan kode <i>booking</i> , kode QR, dan menampilkan informasi pendaftaran	Sistem menghasilkan kode <i>booking</i> , kode QR, dan menampilkan informasi pendaftaran	Valid
[SRS-031]	Menguji fungsi unduh bukti pendaftaran	File PDF dengan informasi pendaftaran berhasil diunduh	File PDF dengan informasi pendaftaran berhasil diunduh	Valid
[SRS-032]	Mengakses sistem dari perangkat <i>mobile</i>	Sistem ditampilkan secara responsif pada resolusi layar <i>mobile</i>	Sistem ditampilkan secara responsif pada resolusi layar <i>mobile</i>	Valid

4.5.2 Penyerahan *Source Code*

Setelah menyelesaikan tahap pengembangan dan pengujian sistem secara menyeluruh, langkah terakhir dalam proyek ini adalah penyerahan kode sumber (*source code*) kepada tim IT RSU St. Elisabeth Purwokerto. Proses ini diawali dengan pembuatan repositori GitHub (<https://github.com/buriane/hospital-regist>) yang berfungsi sebagai sentral untuk menyimpan dan mengelola seluruh kode proyek secara terorganisir. Repositori ini tidak hanya berisi kode inti aplikasi, tetapi juga mencakup dokumentasi teknis, panduan pengguna, dan catatan perubahan (*changelog*) yang komprehensif. Untuk memastikan aksesibilitas, pihak RSU St. Elisabeth Purwokerto secara spesifik meminta agar tautan repositori GitHub

dikirimkan melalui *email* resmi pihak IT RSU St. Elisabeth Purwokerto dan kontak WhatsApp pembimbing lapangan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pelaksanaan kerja praktik di RSU St. Elisabeth Purwokerto, dapat disimpulkan bahwa:

- a. Sistem registrasi *online* RSU St. Elisabeth Purwokerto telah berhasil dirancang dan dibangun menggunakan framework Laravel, dengan integrasi Filament untuk sistem *admin*, sebagaimana ditunjukkan oleh hasil pengujian *Black Box* yang memberikan hasil valid untuk setiap parameter pengujian.
- b. Sistem registrasi *online* ini mampu mengakomodasi kebutuhan pasien umum dan asuransi dalam melakukan pendaftaran secara mandiri, dengan fitur pemilihan poliklinik dan jadwal dokter yang intuitif.
- c. Sistem *admin* yang dikembangkan menggunakan Filament dan Livewire berhasil menyediakan antarmuka yang efisien untuk pengelolaan data pasien, jadwal dokter, dan manajemen sistem registrasi *online* secara keseluruhan.
- d. Penggunaan metode Agile yang diadaptasi untuk pengembangan individual telah memungkinkan pengembangan sistem yang fleksibel dan iteratif, memenuhi kebutuhan spesifik RSU St. Elisabeth Purwokerto dalam waktu yang terbatas.

5.2 Saran

Beberapa rekomendasi untuk meningkatkan sistem registrasi *online* RSU St. Elisabeth Purwokerto, disarankan mengimplementasikan notifikasi otomatis via SMS/*email* untuk pengingat jadwal kunjungan dan perubahan jadwal dokter. Selain itu, penerapan sistem antrian digital terintegrasi akan memungkinkan pasien memantau waktu tunggu secara *real-time*.

DAFTAR PUSTAKA

- Ambriani, D., & Nurhidayat, A. I. (2020). Rancang Bangun Repository Publikasi Ilmiah Dosen Berbasis Web Menggunakan Framework Laravel. *Jurnal Manajemen Informatika*, 10(01), 58–66.
- Aprilia, K., Nuzry, P., & Suryani, E. (2024). Evaluasi Pengalaman Pengguna Website Responsif Beyondsultra Evaluation of the User Experience of the Beyondsultra Responsive Website. *Journal of Information Technology and Computer Science (INTECOMS)*, 7(3), 738–751.
- Aqilah, A. al afif fadhil, Bustamin, S., & Sultan sahrir, S. (2023). Sistem Informasi Manajemen Persediaan Berbasis Web di CV. Makmur Sejahtera Palopo. *Jurnal PROCESSOR*, 18(2), 227–240. <https://doi.org/10.33998/processor.2023.18.2.1385>
- Armono, S. A., & Kurniawan, D. E. (2024). Metode Agile Scrum Dalam Pengembangan Sistem Pengendali Stok Barang. *Desain Dan Analisis Teknologi*, 3(2), 169–177.
- Febriani, O. M., Putra, A. S., & Prayogie, R. P. (2020). Rancang Bangun Sistem Monitoring Sirkulasi Obat Pada Pedagang Besar Farmasi (PBF) Di Kota Bandar Lampung Berbasis Web. *Jurnal Darmajaya*, 1(1), 122–132. <https://jurnal.darmajaya.ac.id/index.php/PSND/article/view/2472>
- Fitriana, K., Priharsari, D., & Kariyoto. (2021). Perancangan User Experience (UX) Aplikasi Manajemen Waktu Berbasis Mobile dengan Metode Design Thinking dan Human-Centered Design. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 5(5), 2036–2044.
- Fitriani, K., & Voutama, A. (2024). Perancangan Uml Sistem Registrasi Pasien Pada Rumah Sakit Bekasi Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 2626–2633. <https://doi.org/10.36040/jati.v8i3.9527>
- Friadi, J., Yani, D. P., Zaid, M., & Sikumbang, A. (2023). Perancangan Pemodelan

- Unified Modeling Language Sistem Antrian Online Kunjungan Pasien Rawat Jalan pada Puskesmas. *Jurnal Ilmu Siber Dan Teknologi Digital*, 1(2), 125–133. <https://doi.org/10.35912/jisted.v1i2.2298>
- Geni, B. Y., Kurnia, O., Putra, R. R. S., & Gunawan, R. (2023). Perancangan Sistem Informasi Dashboard Bpjs Berbasis Visual basic. *Open Journal Systems*, 17(1978), 2837–2850.
- Hanifudin, R., Rokhmayati, P., Fadhlly Noor Rizqi, M., & Fitriana Masitoh, L. (2024). Rancang Bangun Sistem Enterprise Resource Planning (ERP) Berbasis Web pada Pt Sainsgo Karya Indonesia Menggunakan Metode Scrum. *Syntax Idea*, 6(6), 2857–2871. <https://doi.org/10.46799/syntax-idea.v6i6.3889>
- Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K., & Abbas, M. (2019). The Influence of Agile Methodology (Scrum) on Software Project Management. *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 145–149. <https://doi.org/10.1109/SNPD.2019.8935813>
- Kumoro, D. T., & Hasanah, U. (2020). Tinjauan Desain Interface Website E-Commerce Wisata Mototravel.id Menggunakan Evaluasi Heuristik. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, 2(1), 43–49. <https://doi.org/10.35746/jtim.v2i1.84>
- Mariko, S. (2019). Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus. *Jurnal Inovasi Teknologi Pendidikan*, 6(1), 80–91. <https://doi.org/10.21831/jitp.v6i1.22280>
- Muthia Kansha, W. (2023). Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application. *Jurnal Teknik Informatika Stmik Antar Bangsa*, 09(01), 25–30.
- Nabyla, F., & Sigitta, R. C. (2019). Desain Aplikasi Sistem Pendaftaran Online Menggunakan Smartphone Untuk Meningkatkan Mutu Pelayanan Pada Rumah Sakit. *JOINS (Journal of Information System)*, 4(2), 168–177.

<https://doi.org/10.33633/joins.v4i2.3078>

Novianti Putri, S., Voutama Sistem Informasi, A., Singaperbangsa Karawang Jl Ronggo Waluyo, U. H., & Telukjambe Timur, K. (2024). Penerapan Uml Pada E-Reservasi Perawatan Kulit Wajah Berbasis Website (Studi Kasus Rumah Cantik Alisyah). *Jurnal Mahasiswa Teknik Informatika*, 8(2), 2310–2317.

Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103. <https://doi.org/10.56127/jts.v1i2.108>

Nurzaman, F. (2020). Pengembangan Sistem Otomatisasi Tagihan Menggunakan Metode Agile Software 1 Development. IKRA-ITH Informatika: Jurnal Komputer Dan Informatika,. *Jurnal IKRA-ITH Informatika*, 4(1), 46–57.

P, I. G. S., & Abdillah, M. Z. (2024). ANALISIS DAN PERANCANGAN SISTEM INFORMASI MANAJEMEN GEREJA MENGGUNAKAN UML (UNIFIED MODELLING LANGUAGE). *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3), 2634–2641.

Prasetiyo, V., & Chernovita, H. P. (2023). Perancangan Sistem Informasi Manajemen Ruangan pada Dinas Pendidikan Kota Salatiga. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, 7(3), 352–358. <https://doi.org/10.35870/jtik.v7i3.810>

Purba, M. M., & Rahmat, C. (2014). Perancangan Sistem Informasi Stok Barang Berbasis Web Di Pt Mahesa Cipta. *Jurnal Sistem Informasi Universitas Suryadarma*, 9(2), 123–158. <https://doi.org/10.35968/jsi.v9i2.923>

Rahman, R. A., Ziveria, M., Rumah, A., Harum, S., Medika, S., & Umum, S. (2022). Sistem Informasi Registrasi Pada Rumah Sakit Harum Sisma Medika Berbasis WEB. *Sains Dan Teknologi*, 9(1), 49–61.

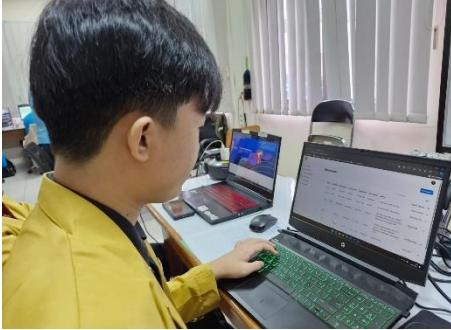
Restaldo, A., & Beeh, Y. R. (2022). Penerapan Framework Laravel pada Sistem Informasi Arsip Dosen Fakultas Keguruan dan Ilmu Pendidikan. *JATISI*

- (*Jurnal Teknik Informatika Dan Sistem Informasi*), 9(1), 785–797.
- Rismawan, I., & Renaningtyas, N. (2024). Analisis Efektifitas Penggunaan E-Medical Record Terhadap Waktu Tunggu Pasien Di Rumah Sakit. *Jurnal ...*, 5(1), 2437–2451.
<http://journal.universitaspahlawan.ac.id/index.php/jkt/article/view/16588>
- Sahi, A. (2020). Aplikasi Test Potensi Akademik Seleksi Saringan Masuk LP3I Berbasis Web Online menggunakan Framework Codeigniter. *Tematik*, 7(1), 120–129. <https://doi.org/10.38204/tematik.v7i1.386>
- Saputra, D., Ivania, A., Sherill, A., & Anindya, Q. M. (2024). Analisis UI dan UX Aplikasi Senayan Library Management System (SLiMS) dalam Pengelolaan Perpustakaan Digital. *Journal of Librarianship and Information Science*, 4(1), 10–30.
- Saputra, H., Sasi Kirana, F., & Sunandar, D. (2024). *Pengembangan Website Official Sekolah Dan PPDB Online Study Kasus Di SMP Mitra Bintaro Islamic School*. 1(4), 131–139.
<https://ojs.jurnalmahasiswa.com/ojs/index.php/bin>
- Sinlae, F., Irwanda, E., Maulana, Z., & Syahputra, V. E. (2024). *Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP*. 2(2), 119–132.
- Suci Widya Primadhani, Yaslis Ilyas, A. I. A. (2021). Sistem Pendaftaran Online sebagai Suatu Strategi Peningkatan Layanan Rumah Sakit. *Citizen-Based Marine Debris Collection Training: Study Case in Pangandaran*, 2(1), 56–61.
- Sugito, B., & Shafira, A. (2023). Design of a Fuel Sales and Distribution Business Application Using a Website. *Prosiding Seminar Nasional Manajemen Industri Dan Rantai Pasok*, 4(1), 59–66.
- Surojudin, N. (2019). Sistem Informasi Persediaan Bahan Baku Pada Pt. Takahashi Spring Indonesia Dengan Menggunakan Metode Waterfall. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.

- Susilo, H., Abdillah, N., Ikhsan, M., & Diana Morika, H. (2023). Analisa Dan Perancangan Sistem Informasi Booking Antrian Pelayan Pada Klinik Medika SaintikaBerbasis Website. *Jurnal Kesehatan Medika Saintika*, 14(Nomor 1), 344–352.
- Wiwesa, N. R. (2021). Jurnal Sosial Humaniora Terapan USER INTERFACE DAN USER EXPERIENCE UNTUK MENGELOLA. *Jurnal Sosial Humaniora Terapan*, 3(2), 2.
- WULANDARI, L. (2020). *Rancang Bangun Website Pengolahan Data Jemaat Gereja Imanuel Terpedo Di Sabbang*. 02(01), 1–6.
<http://repository.uncp.ac.id/id/eprint/369>
- Yuli Fitriyah. (2021). Perancangan dan Evaluasi Aplikasi Sistem Informasi Registrasi Online (SIREGOL) Berbasis Website Menggunakan System Usability Scale (SUS). *Jurnal Ilmu Kesehatan Bhakti Setya Medika*, 6(1), 53–63. <https://doi.org/10.56727/bsm.v6i1.51>

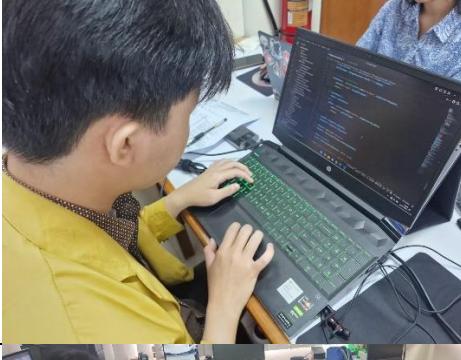
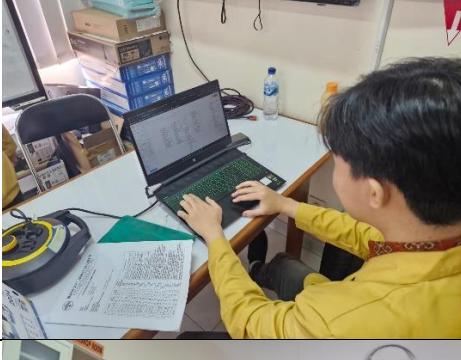
LAMPIRAN 1
RINCIAN KERJA PRAKTIK

No	Waktu	Uraian Pekerjaan	Bukti Dokumentasi
1	10/07/24 [06:49 – 14:10]	Administrasi, registrasi, perkenalan staf, dan perencanaan proyek.	
2	11/07/24 [06:53 – 14:10]	Membuat tampilan UI dan berdiskusi.	
3	12/07/24 [06:51 – 14:10]	Mengaplikasikan tampilan UI pada kode, dan mengambil ID card PKL.	
4	13/07/24 [06:53 – 14:10]	Membuat diagram flowchart dan melanjutkan tampilan UI pada kode.	

5	15/07/24 [06:53 – 14:10]	Membuat gambaran ERD dan menyelesaikan tampilan UI pada kode, serta mengikuti apel pagi.	
6	16/07/24 [06:52 – 14:10]	Membuat <i>migration</i> , relasi tabel, dan implementasi pada kode.	
7	17/07/24 [06:56 – 14:10]	Melanjutkan implementasi pada kode.	
8	18/07/24 [06:56 – 14:10]	Melanjutkan implementasi kode dan membuat <i>dashboard admin</i> dengan Filament.	
9	19/07/24 [06:57 – 14:10]	Melanjutkan <i>admin panel</i> Filament.	

10	20/07/24 [06:53 – 14:10]	Menyelesaikan <i>generate</i> kode <i>booking</i> untuk pasien baru dan lama, serta melanjutkan <i>admin panel</i> Filament.	
11	22/07/24 [06:56 – 14:10]	Membuat bukti pendaftaran sebagai PDF dan melanjutkan <i>dashboard admin</i> Filament, serta mengikuti apel pagi.	
12	23/07/24 [06:58 – 14:10]	Mengubah <i>repeater</i> jadwal dokter dengan kebutuhan, menambah <i>widget</i> pada tampilan <i>dashboard admin</i> , dan fitur cuti dokter.	
13	24/07/24 [06:58 – 14:10]	Memberikan <i>alert</i> pada <i>dashboard admin</i> ketika tanggal/jam awal melebihi tanggal/jam akhir dan modifikasi bukti pendaftaran PDF.	

14	25/07/24 [06:59 – 14:10]	Membuat QR <i>code</i> untuk kode <i>booking</i> dan menambahkan animasi <i>loading</i> pada <i>website</i> .	
15	26/07/24 [06:59 – 14:10]	Konsultasi dan revisi fitur jadwal dokter pada <i>dashboard admin</i> .	
16	27/07/24 [06:56 – 14:10]	Merevisi fitur jadwal dokter pada <i>dashboard</i> dan menambah fitur jadwal khusus dokter.	
17	29/07/24 [07:00 – 14:10]	Mengikuti apel pagi, <i>take</i> video keperluan RS, dan melanjutkan implementasi kode.	

18	30/07/24 [06:58 – 14:10]	Revisi layout jadwal pada <i>dashboard admin panel</i> , memperbaiki beberapa <i>bug</i> sistem.	
19	31/07/24 [06:53 – 14:10]	Membuat no urut antrian pada registrasi dan membuat <i>modal</i> untuk peringatan jika <i>input</i> email/nomor kartu pasien sudah ada di <i>database</i> .	
20	01/08/24 [06:59 – 14:10]	Membuat <i>seeder</i> dan membenarkan <i>footer</i> .	
21	02/08/24 [06:58 – 14:10]	Memperbaiki opsi jadwal khusus dokter pada opsi dokter dan membuat <i>class diagram</i> .	
22	03/08/24 [07:00 – 14:10]	Konsultasi mengenai selesainya proyek dan membuat <i>use case diagram</i> serta memulai pembuatan laporan.	

- 23 04/08/24 Membantu pindahan gedung baru, membuat kabel LAN (*crimping*), dan memasang pc pada setiap ruang poli gedung baru.



- 24 06/08/24 Melanjutkan laporan dan membuat *sequence* diagram.



- 25 07/08/24 Menata kursi bangku untuk suatu acara dan melanjutkan *sequence* diagram laporan.



26	08/08/24 [06:57 – 14:10]	Membuat PPT untuk presentasi akhir di RSU St. Elisabeth dan melanjutkan <i>sequence</i> diagram.	
27	09/08/24 [06:58 – 14:10]	Menyelesaikan <i>sequence</i> diagram dan melanjutkan laporan.	
28	10/08/24 [07:52 – 11:45]	Presentasi laporan proyek yang dikerjakan selama 1 bulan ini.	

LAMPIRAN 2
SERTIFIKAT KELULUSAN



LAMPIRAN 3
SURAT PENERIMAAN INSTANSI



RUMAH SAKIT UMUM ST. ELISABETH

Jl. Dr. Angka No. 40 Purwokerto 53116

Tel. (0281) 625857, 627384, 632833, 623771, Fax. 627824

E-mail : rs_elisabeth@yahoo.com

No : 0695/E/PWT/IV/2024

Purwokerto, 18 April 2024

Lamp :

Hal : Balasan Permohonan Ijin Praktik Kerja

Kepada Yth.
 Wakil Dekan Fakultas Teknik
 Universitas Jenderal Soedirman Purwokerto
 di TEMPAT

Menanggapi surat nomor 1017/UN 23.12.1/PP.05.02/2024, tanggal 03 April 2024, hal surat :
 Permohonan Ijin Praktik Kerja Program Studi S1 Informatika Fakultas Teknik Universitas Jenderal Soedirman Purwokerto, adapun rincian data mahasiswa PKL adalah sebagai berikut :

No	NIM	Nama	Jurusan
1	H1D022009	Brian Cahya Purnama	Informatika

Dengan ini kami sampaikan bahwa Direktur Rumah Sakit Umum St. Elisabeth pada prinsipnya tidak keberatan dan mengizinkan mahasiswa tersebut melakukan Praktik Kerja pada 10 Juli 2024 s.d 09 Agustus 2024.

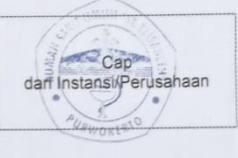
Demikian kami sampaikan agar menjadi periksa. Atas perhatian dan kerjasamanya kami ucapkan terimakasih.

Direktur,

Dr. Rudianto, M.H., M.M.

LAMPIRAN 4

PENILAIAN PELAKSANAAN KERJA PRAKTIK

	FS-KP15 Penilaian Pembimbing Teknis Lapangan																								
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI UNIVERSITAS JENDERAL SOEDIRMAN FAKULTAS TEKNIK Jalan Mayor Jenderal Sungkono KM 5 Blater Purbalingga 53371 Faks. (0281) 6596700, Telp.(0281) 6596801 Laman : www.ft.unsoed.ac.id E-mail : ft@unsoed.ac.id																									
LAMPIRAN PENILAIAN PELAKSANAAN KERJA PRAKTIK																									
Telah dilakukan seminar/ujian Kerja Praktik atas nama mahasiswa sebagai berikut,																									
Nama : BRIAN CAHYA PURNAMA NIM : H1D022009 Jurusan : Informatika Hari/Tanggal seminar :	dengan perincian nilai sebagai berikut :																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">No.</th> <th style="width: 60%;">Komponen Penilaian</th> <th style="width: 30%;">Nilai</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Kesesuaian dengan rencana kerja</td> <td>90</td> </tr> <tr> <td>2.</td> <td>Kehadiran di lokasi Kerja Praktik</td> <td>100</td> </tr> <tr> <td>3.</td> <td>Kedisiplinan,Sikap,Etika dan Tingkah Laku</td> <td>87</td> </tr> <tr> <td>4.</td> <td>Keaktifan dan kreatifitas</td> <td>90</td> </tr> <tr> <td>5.</td> <td>Kecermatan</td> <td>85</td> </tr> <tr> <td>6.</td> <td>Tanggung jawab</td> <td>86</td> </tr> <tr> <td></td> <td style="text-align: center;">HURUF MUTU</td> <td style="text-align: center;">A AB B BC C CD D E</td> </tr> </tbody> </table>		No.	Komponen Penilaian	Nilai	1.	Kesesuaian dengan rencana kerja	90	2.	Kehadiran di lokasi Kerja Praktik	100	3.	Kedisiplinan,Sikap,Etika dan Tingkah Laku	87	4.	Keaktifan dan kreatifitas	90	5.	Kecermatan	85	6.	Tanggung jawab	86		HURUF MUTU	A AB B BC C CD D E
No.	Komponen Penilaian	Nilai																							
1.	Kesesuaian dengan rencana kerja	90																							
2.	Kehadiran di lokasi Kerja Praktik	100																							
3.	Kedisiplinan,Sikap,Etika dan Tingkah Laku	87																							
4.	Keaktifan dan kreatifitas	90																							
5.	Kecermatan	85																							
6.	Tanggung jawab	86																							
	HURUF MUTU	A AB B BC C CD D E																							
Keterangan : A. Nilai \geq 80 BC. $65 \leq$ Nilai $<$ 69,99 D. $46 \leq$ Nilai $<$ 55,99 AB. $75 \leq$ Nilai $<$ 79,99 C. $60 \leq$ Nilai $<$ 64,99 E. $<$ 46 B. $70 \leq$ Nilai $<$ 74,99 CD. $56 \leq$ Nilai $<$ 59,99																									
Nilai di atas akan sah sebagai nilai matakuliah jika yang bersangkutan telah melengkapi kekurangannya yang tercantum dalam "Berita Acara Seminar Kerja Praktik" beserta menyerahkan laporan akhir Kerja Praktik yang telah dijilid dan disahkan. Ditetapkan di																									
Nama Pembimbing Teknis Lapangan : Stefanus Kristiantono, S.Kom	TTD :  <div style="text-align: center; border: 1px solid black; border-radius: 50%; width: 50px; height: 50px; margin: 10px auto;">  </div>																								
<i>[Rangkap 3:dosen pembimbing, bapendik program]</i>																									

LAMPIRAN 5
LEMBAR PRESENSI KERJA PRAKTIK



RUMAH SAKIT UMUM ST. ELISABETH

Jl. Dr. Angka No. 40 Purwokerto 53116
 Tel. (0281) 625857, 627384, 632833, 623771, Fax. 627824
 E-mail : rs_elisabeth@yahoo.com

LEMBAR PRESENSI KERJA PRAKTIK

Nama : Biran Cahya P
 NIM : H10022009
 Jurusan : Informatika
 Topik Kerja Praktik : Rancang Bangun Website

Nomor SPK : 1584 / UN.23.08.6 / PP.05.02 / 2024
 Dimulai Sejak : 10 Juli 2024
 Nama Pembimbing Lapangan : Stefanus Kris Bratmono, S.Kom
 NIP / NIK : 20141126191179

No	Tanggal	Jam Masuk	Jam Keluar	Tanda Tangan Pembimbing	Uraian Pekerjaan
1	10/07/24	06 : 49	14 : 10		Administrasi, registrasi, pencadangan stasi, dan perencanaan proyek
2	11/07/24	06 : 53	14 : 10		Membuat tampilan UI dan berdiskusi mengambil ID Card PKL
3	12/07/24	06 : 51	14 : 10		Mengaplikasikan tampilan UI pada code, dan menyelesaikan 10 Card PKL
4	13/07/24	06 : 53	14 : 10		menulis program HomeWork dan melanjutkan tampilan UI pada code
5	15/07/24	06 : 53	14 : 10		membuat gambaran ERD dan menyelesaikan tampilan UI pada code, serta opel pagi
6	16/07/24	06 : 52	14 : 10		membuat migrasi, tabel relasi, dan implementasi code
7	17/07/24	06 : 56	14 : 10		melanjutkan implementasi code
8	18/07/24	06 : 56	14 : 10		melanjutkan implementasi code dan membuat dashboard admin dengan filament
9	19/07/24	06 : 57	14 : 10		Melanjutkan admin panel filament
10	20/07/24	06 : 53	14 : 10		menyelesaikan generate kode booking untuk patient baru & lama, serta perbaikan admin
11	22/07/24	06 : 56	14 : 10		memberi bukti pendukung sebagai PDF dan melanjutkan dashboard admin filament dan opel pagi
12	23/07/24	06 : 58	14 : 10		menambahkan repeater jadwal dokter dengan kebutuhan, menambahkan tampilan dashboard admin dan fitur cuti dokter
13	24/07/24	06 : 58	14 : 10		menambahkan alere pada dashboard ketika tanggal /jam awal nihil/tidak ada. Dan mendefinisikan bukti pendukung PDF
14	25/07/24	06 : 59	14 : 10		Membuat QR code untuk kode booking dan menambahkan animasi loading pada website
15	26/07/24	06 : 59	14 : 10		Konsultasi dan revisi fitur jadwal dokter pada dashboard admin
16	27/07/24	06 : 56	14 : 10		menambahkan fitur jadwal dokter pada dashboard dan menambah fitur jadwal khusus dokter
17	29/07/24	07 : 00	14 : 10		menambahkan opel post, faile video keperluan RS, dan melanjutkan implementasi code
18	30/07/24	06 : 58	14 : 10		Revisi layout jadwal pada dashboard admin panel, memperbaiki beberapa bug sistem
19	31/07/24	06 : 53	14 : 10		Membuat no urut aman pada registrasi, dan membuat modal UI peningatan jika input email/no not konsisten mengisi seeder dan memberikan footer sudah adارد
20	01/08/24	06 : 59	14 : 10		Mempersiapkan opsi jadwal kantor dokter pada opel dokter dan membuat class drrogram
21	02/08/24	06 : 58	14 : 10		



RUMAH SAKIT UMUM ST. ELISABETH

Jl. Dr. Angka No. 40 Purwokerto 53116

Tel. (0281) 625857, 627384, 632833, 623771, Fax. 627824

E-mail : rs_elisabeth@yahoo.com

22	03/08/24	07 : 00	14 : 10	<i>RS</i>	konseptasi mengenai selesainya proyek dan membuat use case diagram. Serta memulai pembuatan laporan membantu pindahan gedung baru, membuat kabel LAN (cabling), dan memasang PC pada setiap ruang gedung baru melanjutkan laporan dan membuat sequence diagram.
23	04/08/24	09 : 58	18 : 20	<i>RS</i>	
24	06/08/24	06 : 57	14 : 10	<i>RS</i>	
25	07/08/24	07 : 00	14 : 10	<i>RS</i>	menulis surat bantuan untuk bantuan dan melanjutkan Sequence diagram laporan
26	08/08/24	06 : 57	14 : 10	<i>RS</i>	membuat ppt untuk presentasi akhir di RSU St. Elisabeth dan melanjutkan sequence diagram
27	09/08/24	06 : 58	14 : 10	<i>RS</i>	menyelesaikan sequence diagram dan melanjutkan laporan.
28	10/08/24	07 : 52	11 : 45	<i>RS</i>	Presentasi laporan proyek yg dikergakan selama 1 bulan
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					

Mengetahui,

(Setefanus Kristiamono, S.Kom)

LAMPIRAN 6

CURRICULUM VITAE

BRIAN CAHYA PURNAMA

brian.purnama@mhs.unsoed.ac.id | +62 822 6803 8586



PROFIL

NIM	: H1D022009
Tempat, Tanggal Lahir	: Purwokerto, 14 Desember 2003
Agama	: Kristen
Alamat	: Penatusan I No. 319, Purwokerto Timur, Purwokerto Wetan, Kabupaten Banyumas 53111
Keahlian	: Web Developer (Basis PHP/Laravel, Basis Wordpress), Data Analyst (SQL, Basis Rapid Miner, Basis Python), UI/UX Designer (Figma), Multimedia (CapCut, Basis After Effects), Microsoft Office (Word, PowerPoint, Basis Excel), Graphic Design (Canva, Basis CorelDRAW/Illustrator)

EDUKASI

**Universitas Jenderal Soedirman
Informatika**

**Jul 2022 - Sekarang
IP: 3.96/4.0**

Mata Kuliah Relevan: Algoritma Pemrograman, Struktur Data, Matematika Diskrit, Logika Informatika, Basis Data, Pemrograman Web, Rekayasa Perangkat Lunak, Sistem Operasi, Penambangan Data, Kecerdasan Buatan, Sistem Informasi, Audit Sistem Informasi, Analisis dan Desain Sistem, Probabilitas dan Statistika, Pemrograman Berorientasi Objek, Jaringan Komputer, Interaksi Manusia Komputer, Teknologi Multimedia.

Aktivitas: Menjadi anggota aktif Himpunan Mahasiswa Informatika Unsoed.

Penghargaan: Best Project II pada Praktikum Pemrograman Web (*Jujutsu Kaisen Hub*).

**SMA Negeri 1 Purwokerto
MIPA**

**Jul 2019 – Mei 2022
Nilai Akhir: 92,94**

Aktivitas: Menjadi anggota aktif organisasi Kerohanian 2019 - 2021 dan OSIS 2020 - 2021.

Penghargaan: Lulus SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) 2022 memasuki program studi Informatika, Universitas Jenderal Soedirman.

PENGALAMAN

**Staf Keilmuan dan Teknologi
Himpunan Mahasiswa Informatika Unsoed**

Feb 2024 - Sekarang

- Menjadi penanggung jawab agenda kerja Pengelolaan Web divisi Keilmuan dan Teknologi.
- Menghadiri rapat untuk membahas progres dan berbagai kegiatan himpunan.
- Terlibat dalam beberapa kegiatan, baik sebagai anggota atau koordinator divisi.
- Berinteraksi dan membentuk hubungan baik dengan teman-teman satu himpunan melalui kegiatan sosial dan kegiatan.

PROYEK

Soeara - Website

Apr 2024 - Jul 2024

- Soeara (Soedirman Berbicara) adalah platform petisi yang bertujuan untuk memberikan suara kepada komunitas Unsoed dan mempromosikan perubahan positif dalam lingkungan kampus.
- Bertujuan untuk memperjuangkan perubahan dan hak mahasiswa, staf, dan alumni.
- Teknologi: VPS, Laravel, Tailwind CSS, Alpine JS, Livewire, dan Filament.
- Proyek ini merupakan tugas akhir dari mata kuliah Rekayasa Perangkat Lunak, Pemrograman Web II, dan Interaksi Manusia Komputer.

Connected - Website

Okt 2023 - Des 2023

- Connected adalah website manajemen proyek sederhana untuk perusahaan fiktif.
- Mengimplementasikan konsep CRUD (*Create, Read, Update, Delete*).
- Profesi sebagai *Frontend* dan *Backend Engineer* dalam tim yang beranggotakan 3 orang.
- Teknologi: Laravel 10, Bootstrap, MySQL, JavaScript, dan HTML/CSS.
- Proyek ini dikembangkan untuk Tugas Akhir mata kuliah Pemrograman Web dan menjadi Proyek Terbaik.

Jujutsu Kaisen Hub - Website

Nov 2023 - Des 2023

- Jujutsu Kaisen Hub adalah website yang dibuat untuk menjadi sumber utama informasi, artikel, episode, dan season terkait anime “Jujutsu Kaisen”.

- Profesi sebagai *Backend* dan *UI/UX Engineer* dalam tim yang beranggotakan 2 orang.
- Teknologi: *MySQL, PHP, JavaScript*, dan *HTML/CSS*.
- Design Tool: *Figma*.
- Proyek ini dikembangkan untuk Ujian Praktikum Pemrograman Web dan memenangkan penghargaan Proyek Terbaik II.

Glorious Web Survey - Website**Nov 2023 - Des 2023**

- Glorious Web Survey adalah website pengelolaan, pembuatan, dan pengisian survei.
- Mengimplementasikan konsep CRUD (*Create, Read, Update, Delete*).
- Profesi sebagai *Backend* dan *UI/UX Engineer* dalam tim yang beranggotakan 5 orang.
- Teknologi: *PHP, JavaScript, MySQL*, dan *HTML/CSS*.
- Design Tool: *Figma*.
- Proyek ini dikembangkan untuk Tugas Akhir mata kuliah Analisis dan Desain Sistem.

Workshop Registration - Website**Nov 2023 - Des 2023**

- Workshop Registration adalah website yang dibuat untuk pendaftaran workshop sederhana.
- Mengimplementasikan konsep CRUD (*Create, Read, Update, Delete*).
- Profesi sebagai *Frontend* dan *Backend Engineer* dalam tim yang beranggotakan 5 orang.
- Teknologi: *Play Framework 1.6, Java, MySQL*, dan *HTML/CSS*.
- Proyek ini dikembangkan untuk Tugas Akhir mata kuliah Pemrograman Berorientasi Objek.

RELAWAN**Pengajar Microsoft Office
Himpunan Mahasiswa Informatika Unsoed****9 Sep 2023**

- Program pengabdian masyarakat bernama Informatika Mengajar di SD Negeri 5 Serang, Purbalingga.
- Mengajarkan materi Microsoft Office seperti fungsi Word, Excel, dan PowerPoint pada 20 siswa/siswi kelas 5.
- Melakukan permainan terkait materi yang dipaparkan dan siswa/siswi yang salah diberi hukuman berupa bernyanyi dan menari.

**Panitia Bakti Sosial
Organisasi Siswa Intra Sekolah****20 Mar 2021**

- Program kerja OSIS bernama Bakti Sosial di Desa Sambirata, Cilongok.
- Kegiatan dilaksanakan saat pandemi Covid-19.
- Melakukan penyemprotan disinfektan pada rumah-rumah warga dan pembagian sembako untuk warga Desa Sambirata, sebagai bagian dari upaya pencegahan penyebaran penyakit dan bantuan sosial dalam situasi darurat.

KETERAMPILAN NON-TEKNIS

1. Kolaborasi dalam tim
2. Manajemen waktu
3. Penyelesaian masalah
4. Kejujuran
5. Berpikir kritis
6. Komunikasi