

TUGAS AKHIR

**RANCANG BANGUN APLIKASI SISTEM *ELECTRONIC
PRESCRIBING* DOKTER BERBASIS *WEBSITE* DI
KLINIK ASY-SYIFA DESA RANDUDONGKAL**



MUHAMMAD ARIF ALIFAN SULTONI

20102036

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

TUGAS AKHIR

**RANCANG BANGUN APLIKASI SISTEM *ELECTRONIC
PRESCRIBING DOKTER BERBASIS WEBSITE DI
KLINIK ASY-SYIFA DESA RANDUDONGKAL***

***DESIGN AND DEVELOPMENT OF A WEBSITE BASED
DOCTOR'S ELECTRONIC PRESCRIBING SYSTEM
APPLICATION AT ASY-SYIFA CLINIC IN
RANDUDONGKAL VILLAGE***



MUHAMMAD ARIF ALIFAN SULTONI

20102036

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

LEMBAR PERSETUJUAN PEMBIMBING

RANCANG BANGUN APLIKASI SISTEM *ELECTRONIC PRESCRIBING* DOKTER BERBASIS WEBSITE DI KLINIK ASY-SYIFA DESA RANDUDONGKAL

DESIGN AND DEVELOPMENT OF A WEBSITE BASED DOCTOR'S ELECTRONIC PRESCRIBING SYSTEM APPLICATION AT ASY-SYIFA CLINIC IN RANDUDONGKAL VILLAGE

Dipersiapkan dan Disusun Oleh
Muhammad Arif Alifan Sultoni
20102036

Usulan penelitian Tugas Akhir telah disetujui
Pada tanggal 5 Juni 2024

Pembimbing Utama,

(Abednego Dwi Septiadi, S.Kom., M.Kom.)

NIDN. 0616098901

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI SISTEM *ELECTRONIC PRESCRIBING* DOKTER BERBASIS WEBSITE DI KLINIK ASY-SYIFA DESA RANDUDONGKAL

*DESIGN AND DEVELOPMENT OF A WEBSITE BASED DOCTOR'S
ELECTRONIC PRESCRIBING SYSTEM APPLICATION AT ASY-SYIFA
CLINIC IN RANDUDONGKAL VILLAGE*

Dipersiapkan dan Disusun oleh

Muhammad Arif Alisan Sultoni

20102036

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tugas Akhir Pada Hari
Kamis, Tanggal 20 Juni 2024

Penguji 1

(Arief Rais Bahtiar, S.Kom., M.Kom.)

NIDN. 0604119101

Penguji 2

(Novanda Alim Setya Nugraha, S.S., M.Hum.)

NIDN. 0627119002

Pembimbing Utama,

(Abednego Dwi Septiadi, S.Kom., M.Kom.)

NIDN. 0616098901



Dekan,

(Auliya Burhanuddin, S.S., M.Kom.)

NIK.19820008

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertandatangan di bawah ini,

Nama Mahasiswa : Muhammad Arif Alifan Sulton

NIM : 20102036

Program Studi : Teknik Informatika

Menyatakan bahwa Tugas Akhir dengan judul berikut :

**RANCANG BANGUN APLIKASI SISTEM ELECTRONIC PRESCRIBING
DOKTER BERBASIS WEBSITE DI KLINIK ASY-SYIFA DESA
RANDUDONGKAL**

Dosen Pembimbing Utama : Abednego Dwi Septiadi, S.Kom., M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Institut Teknologi Telkom Purwokerto maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan, dan penelitian Saya Sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam Karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggungjawab Saya, bukan tanggungjawab Institut Teknologi Telkom Purwokerto.
5. Pernyataan ini Saya buat dengan sesungguhnya, apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka Saya bersedia menerima Sanksi Akademik dengan pencabutan gelar yang sudah diperoleh serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Purwokerto, 5 Juni 2024,

Yang Menyatakan,



(Muhammad Arif Alifan Sulton)

KATA PENGANTAR

Dengan rasa syukur dan puji, penulis ingin mengucapkan terima kasih kepada Allah SWT. atas berkat dan rahmat-Nya yang memungkinkan penulis menyelesaikan skripsi ini dengan lancar sebagai syarat lulus di Institut Teknologi Telkom Purwokerto. Penulis juga mengakui dukungan moral dan material yang diberikan oleh berbagai pihak, terutama kepada semua individu dan kelompok yang turut membantu dalam penyusunan tugas akhir ini, terutama kepada:

1. Tuhan Yang Maha Esa (Tuhan YME), penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Ibunda tercinta, Ibu Asmaiyyah dan Ayahanda Pahlawan saya, Bapak Beni Mustofa yang menjadi tempat curhat berbagai macam hambatan. Terima kasih atas doa, dukungan serta motivasinya.
3. Segenap Keluarga Besar Bapak dan Ibu saya yang memberikan dukungan moril sehingga lelah ini tidak terasa.
4. Mas Dan Mbaku sekalian serta Keponakan-keponakanku yang harus panggil saya om. Terimakasih atas dukungan dan penyemangat dalam perjalanan saya mendapatkan gelar sarjana.
5. Dr. Tenia Wahyuningrum, S.Kom., M.T. selaku Rektor Institut Teknologi Telkom Purwokerto.
6. Auliya Burhanuddin, S.Si., M.Kom. selaku Dekan Fakultas Informatika Institut Teknologi Telkom Purwokerto
7. Amalia Beladinna Arifa, S.Pd., M.Cs. selaku Ketua Program Studi Teknik Informatika.
8. Abednego Dwi Septiadi, S.Kom., M.Kom. Selaku Dosen Pembimbing yang telah memberikan saran dan masukan luar biasa dalam penyusunan Tugas Akhir ini.
9. Viky Indika Tintatiana yang telah membantu dan selalu mendoakan dari pembuatan Tugas Akhir sampai mendapatkan gelar sarjana.
10. Teman-teman seperjuangan HAM yang telah menemaniku untuk bersama-sama memperjuangkan gelar S.Kom.

Penulis menyadari bahwa tugas akhir ini jauh dari sempurna karena keterbatasan pengalaman dan pengetahuan penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran dan pendapat dari berbagai pihak bahkan kritik yang membangun. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca dan semua pihak yang terlibat khususnya dalam bidang teknologi informasi.

Purwokerto, 5 Juni 2024

Penulis,



Muhammad Arif Alifan Sulton

DAFTAR ISI

TUGAS AKHIR.....	i
LEMBAR PERSETUJUAN PEMBIMBING	ii
LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiv
ABSTRAK.....	xv
<i>ABSTRACT</i>	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Pertanyaan Penelitian	5
1.4 Batasan Masalah.....	5
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Tinjauan Pustaka	7
2.2 Landasan Teori.....	14
2.2.1 Rancang Bangun	14
2.2.2 <i>Website</i>	14
2.2.3 Sistem <i>Electronic Prescribing</i>	14
2.2.4 <i>User Interface</i>	15

2.2.5	<i>Database MySQL</i>	15
2.2.6	UML	16
2.2.9	Pemrograman <i>Web</i>	21
2.2.10	<i>JavaScript</i>	21
2.2.11	<i>Framework</i>	22
2.2.12	Metode <i>Extreme Programming</i>	22
2.2.13	<i>Blackbox Testing</i>	26
2.2.14	Rasio Efisiensi.....	27
	BAB III METODE PENELITIAN.....	29
3.1	Subjek Dan Objek Penelitian	29
3.1.1	Subjek Penelitian.....	29
3.1.2	Objek Penelitian	29
3.2	Alat Dan Bahan	29
3.2.1	Alat.....	29
3.2.2	Bahan.....	30
3.3	Diagram Alir Penelitian	30
3.3.1	Identifikasi Masalah.....	31
3.3.2	Studi Literatur	31
3.3.3	Perencanaan.....	32
3.3.4	Perancangan	33
3.3.5	Pengkodean	34
3.3.6	Pengujian.....	34
3.3.7	Hasil Analisis	34
	BAB IV HASIL DAN PEMBAHASAN	36
4.1	Hasil	36
4.2	Pembahasan.....	48

4.2.1	Perencanaan/ <i>Planning</i>	48
4.2.2	Perancangan/ <i>Design</i>	51
4.2.3	Pengkodean/ <i>Coding</i>	63
4.2.4	Pengujian/ <i>Testing</i>	73
4.2.5	Hasil Analisis	95
BAB V KESIMPULAN DAN SARAN.....		99
5.1	Kesimpulan	99
5.2	Saran.....	99
DAFTAR PUSTAKA		100
LAMPIRAN		105

DAFTAR GAMBAR

Gambar 1.1 Wawancara Dengan Dokter	2
Gambar 2.1 Contoh Diagram <i>Use Case</i> [23].....	17
Gambar 2.2 Contoh <i>Activity Diagram</i> [24]	19
Gambar 2.3 Contoh <i>Class Diagram</i> [25].....	21
Gambar 2.4 Metode <i>Extreme Programming</i> [32]	23
Gambar 3.1 Diagram Alir Penelitian.....	31
Gambar 4.1 Halaman <i>Login User</i>	36
Gambar 4.2 Halaman <i>Dashboard</i> Dokter	37
Gambar 4.3 Tampilan Awal Halaman Resep Obat.....	38
Gambar 4.4 Tampilan Awal Pilih Pasien Baru.....	38
Gambar 4.5 Tampilan Awal Pilih Pasien Lama.....	39
Gambar 4.6 Konfirmasi Data Pasien Sudah Benar	39
Gambar 4.7 Tampilan Kedua Halaman Resep Obat	40
Gambar 4.8 Tambah Resep Pada Form Resep Obat.....	40
Gambar 4.9 Konfirmasi Simpan Resep.....	41
Gambar 4.10 Halaman <i>Dashboard</i> Apoteker.....	41
Gambar 4.11 Halaman Stok Resep	42
Gambar 4.12 Tampilan <i>Pop Up</i> Pada Tambah Resep.....	42
Gambar 4.13 Tampilan <i>Pop Up</i> Pada Edit Resep	43
Gambar 4.14 Konfirmasi Data Resep Dihapus	43
Gambar 4.15 Halaman Riwayat Resep	44
Gambar 4.16 Tampilan <i>Pop Up</i> Buat Resep.....	44
Gambar 4.17 Hasil Cetak Resep Obat Pasien	45
Gambar 4.18 Halaman <i>Dashboard</i> Role Admin	45
Gambar 4.19 Halaman Pasien	46
Gambar 4.20 Tampilan Edit Pasien	46
Gambar 4.21 Halaman <i>Users</i>	47
Gambar 4.22 Tampilan <i>Add User</i>	47
Gambar 4.23 Tampilan Edit <i>User</i>	48

Gambar 4.24 Konfirmasi Nonaktifkan <i>User</i>	48
Gambar 4.25 <i>Use Case Diagram</i>	52
Gambar 4.26 <i>Activity Diagram</i> Halaman <i>Login Dokter</i>	53
Gambar 4.27 <i>Activity Diagram</i> Halaman Resep Obat Dokter.....	54
Gambar 4.28 <i>Activity Diagram</i> Halaman <i>Dashboard</i>	55
Gambar 4.29 <i>Activity Diagram</i> Halaman <i>Login Apoteker</i>	56
Gambar 4.30 <i>Activity Diagram</i> Halaman Stok Resep	57
Gambar 4.31 <i>Activity Diagram</i> Halaman Riwayat Resep Apoteker	58
Gambar 4.32 <i>Activity Diagram</i> Halaman <i>Login Admin</i>	59
Gambar 4.33 <i>Activity Diagram</i> Halaman Pasien <i>Admin</i>	60
Gambar 4.34 <i>Activity Diagram</i> Halaman <i>Users Admin</i>	61
Gambar 4.35 <i>Class Diagram</i>	62
Gambar 4.36 Kode Fungsi Pada Halaman <i>Login</i>	63
Gambar 4.37 Kode Fungsi Program Pada Halaman <i>Dashboard</i>	64
Gambar 4.38 Kode Fungsi <i>Chart</i> Pada Halaman <i>Dashboard</i>	65
Gambar 4.39 Kode Fungsi Pada Halaman Resep Obat.....	66
Gambar 4.40 Kode Fungsi Program Tampilan Obat	67
Gambar 4.41 Kode Fungsi Program Tampilan Resep Obat.....	67
Gambar 4.42 Kode Program Halaman Riwayat Resep.....	68
Gambar 4.43 Kode Fungsi Program Proses Resep	69
Gambar 4.44 Kode Fungsi Program Halaman Pasien.....	70
Gambar 4.45 Kode Fungsi Program Tampilan Edit.....	70
Gambar 4.46 Kode Program Halaman <i>Users</i>	71
Gambar 4.47 Kode Fungsi Program Edit <i>User</i>	72
Gambar 4.48 Kode Fungsi Program Konfirmasi Status <i>User</i>	72

DAFTAR TABEL

Tabel 1.1 Waktu proses pembuatan resep sampai jadi obat secara manual	3
Tabel 2.1 Ringkasan Penelitian Terdahulu.....	11
Tabel 2.2 <i>Use-case</i> Diagram [23]	16
Tabel 2.3 <i>Activity</i> Diagram [24].....	18
Tabel 2.4 <i>Class</i> Diagram [25]	19
Tabel 2.5 Contoh Kebutuhan Pengguna [34].....	24
Tabel 2.6 Contoh <i>User Story</i> [33]	24
Tabel 2.7 Contoh Pengujian Fitur Pada Halaman <i>Login</i> [36].....	26
Tabel 2.8 Contoh <i>Test Case</i> [38]	27
Tabel 3.1 Spesifikasi <i>Hardware</i>	29
Tabel 3.2 Daftar <i>Software</i>	30
Tabel 3.3 Pertanyaan yang Diajukan	32
Tabel 4.1 <i>User Stories</i> Iterasi Pertama.....	50
Tabel 4.2 <i>User Stories</i> Iterasi Kedua	51
Tabel 4.3 <i>Blackbox Testing</i> Dokter Sebelum <i>login</i>	73
Tabel 4.4 <i>Blackbox Testing</i> Dokter Halaman <i>Login</i>	74
Tabel 4.5 <i>Blackbox Testing</i> Dokter Halaman <i>Dashboard</i>	75
Tabel 4.6 <i>Blackbox Testing</i> Dokter Halaman Resep Obat.....	77
Tabel 4.7 <i>Blackbox Testing</i> Apoteker Sebelum <i>Login</i>	79
Tabel 4.8 <i>Blackbox Testing</i> Apoteker Halaman <i>Login</i>	79
Tabel 4.9 <i>Blackbox Testing</i> Apoteker Halaman <i>Dashboard</i>	81
Tabel 4.10 <i>Blackbox Testing</i> Apoteker Halaman Stok Resep.....	82
Tabel 4.11 <i>Blackbox Testing</i> Apoteker Halaman Riwayat Resep.....	84
Tabel 4.12 <i>Blackbox Testing</i> Admin Sebelum <i>Login</i>	86
Tabel 4.13 <i>Blackbox Testing</i> Admin Halaman <i>Login</i>	87
Tabel 4.14 <i>Blackbox Testing</i> Admin Halaman <i>Dashoard</i>	89
Tabel 4.15 <i>Blackbox Testing</i> Admin Halaman Pasien.....	90
Tabel 4.16 <i>Blackbox Testing</i> Admin Halaman <i>Users</i>	92
Tabel 4.17 Waktu proses pembuatan resep sampai jadi obat secara manual.....	95

Tabel 4.18 Waktu Proses menggunakan sistem *electronic prescribing* 96

DAFTAR LAMPIRAN

Lampiran 1. Surat Permohonan Pengambilan Data Penelitian	105
Lampiran 2. Daftar Pertanyaan Wawancara	106
Lampiran 3. Hasil Wawancara	107
Lampiran 4 Rancangan <i>User Interface</i>	109
Lampiran 5 Penyerahan <i>Website</i>	110

ABSTRAK

RANCANG BANGUN APLIKASI SISTEM *ELECTRONIC PRESCRIBING* DOKTER BERBASIS *WEBSITE* DI KLINIK ASY-SYIFA DESA RANDUDONGKAL

Oleh
Muhammad Arif Alifan Sultoni
20102036

Kemajuan teknologi yang cepat telah menyebar ke beragam bidang kehidupan manusia, termasuk perannya dalam transformasi dunia perpustakaan sebagai penyebaran informasi. Kemajuan teknologi sangat canggih telah membutuhkan perkakas teknologi untuk membangun sistem, salah satunya adalah membuat sistem *electronic prescribing* oleh seorang pengembang *web*. *Electronic prescribing* adalah metode pengiriman resep obat melalui *platform* digital yang menghubungkan dokter, perangkat pembuat resep elektronik dan apoteker. Sistem ini telah dianjurkan digunakan dalam praktik peresepan dokter di rumah sakit dan klinik untuk mengurangi kesalahan pembuatan resep obat secara manual, tidak semua layanan kesehatan mempunyai sistem tersebut salah satunya adalah klinik Asy-Syifa, masih menggunakan proses pembuatan resep obat secara manual. Pengembangan aplikasi sistem *electronic prescribing* dokter dibuat berbasis *website* untuk dokter dan apoteker di Klinik Asy-Syifa Desa Randudongkal, metode yang diterapkan dalam pengembangan sistem ini adalah pendekatan *Extreme Programming*, sebuah metodologi pengembangan perangkat lunak yang menekankan pengembangan berurutan dan berulang. Sistem ini dibangun menggunakan bahasa pemrograman *JavaScript* dengan *framework ReactJS* untuk *frontend*, *ExpressJS* untuk *backend*, *MySQL* untuk *database*, dan *Black box* untuk pengujian sistem. Tujuan utamanya memudahkan pengguna sistem ini bagi dokter dan apoteker, sehingga mereka dapat memanfaatkannya dengan mudah dan efisien. Pengujian akhir dari *website* ini menunjukkan bahwa keseluruhan sistem berfungsi dengan baik, semua fitur dari halaman *dashboard*, halaman input data pasien sampai fungsi untuk buat resep sudah berjalan dengan semestinya.

Kata kunci : *Black box*, Klinik As-Syifa, Sistem *electronic prescribing*, *Website*

ABSTRACT

DESIGN AND DEVELOPMENT OF A WEBSITE BASED DOCTOR'S ELECTRONIC PRESCRIBING SYSTEM APPLICATION AT ASY-SYIFA CLINIC IN RANDUDONGKAL VILLAGE

Oleh
Muhammad Arif Alifan Sulton
20102036

The rapid advancement of technology has permeated various aspects of human life, including its role in transforming the library world as a means of information dissemination. This sophisticated technological progress necessitates tools to build systems, one of which is the development of an electronic prescribing system by a web developer. Electronic prescribing is a method of transmitting drug prescriptions through a digital platform that connects doctors, electronic prescription devices, and pharmacists. This system is recommended for use in medical prescription practices in hospitals and clinics to reduce errors associated with manual prescription creation. However, not all healthcare services have implemented this system; for instance, the Asy-Syifa Clinic still uses a manual prescription process. The development of the electronic prescribing application for doctors is web-based and specifically designed for doctors and pharmacists at the Asy-Syifa Clinic in Randudongkal Village. The methodology applied in the development of this system is the Extreme Programming approach, a software development methodology that emphasizes sequential and iterative development. The system is built using the JavaScript programming language with the ReactJS framework for the frontend, ExpressJS for the backend, MySQL for the database, and Black box for system testing. The primary goal is to facilitate the use of this system for doctors and pharmacists, enabling them to utilize it easily and efficiently. Final testing of this website indicates that the overall system functions well, with all features from the dashboard page, patient data input page, to the prescription creation function operating as intended.

Keywords : Asy-Syifa clinic, Black box, Electronic prescribing system, Website

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada perkembangan teknologi saat ini, kemajuan yang signifikan di berbagai bidang ilmu pengetahuan tercermin dalam produk-produk teknologi. Kemajuan ini memberikan dampak besar dalam memenuhi kebutuhan individu maupun lembaga. Teknologi yang terus berkembang dimanfaatkan dalam berbagai konteks, termasuk juga untuk meningkatkan efisiensi kerja dengan cara yang sesuai dengan kebutuhan masing-masing individu atau institusi [1].

Perkembangan teknologi yang sangat cepat telah merasuki berbagai kehidupan manusia, termasuk peranannya dalam dunia perpustakaan sebagai tempat penyebaran informasi. Kemajuan teknologi yang semakin canggih telah membuat informasi menjadi lebih mudah diakses dan telah membantu mempermudah berbagai aspek aktivitas manusia [2]. Dari pesatnya perkembangan teknologi, kebutuhan akan alat-alat teknologi yang akan digunakan dalam pembangunan sistem semakin meningkat. Perangkat yang disebut termasuk perangkat keras seperti *laptop* dan perangkat lunak seperti *Sublime Text*, *Xampp* dan *Chrome*. Penggunaan teknologi berbasis *website* yang terkoneksi secara *online* dapat memberikan kemudahan dalam mengolah dan memproses data [2].

Sebuah *website* merupakan sekumpulan informasi atau halaman yang dapat dijangkau lewat *internet*, memungkinkan akses bagi siapapun, di manapun, dan kapanpun, asal terhubung dengan *internet*. Dari segi teknik, *website* terdiri dari berbagai halaman yang tergabung dibawah sebuah domain atau subdomain tertentu [3]. Situs *website* dapat berfungsi sebagai indikator yang meningkatkan komunikasi antara berbagai pihak. Bahkan di lokasi dan waktu yang berbeda, siapapun bisa terhubung dengan *internet* [4].

Pada *website* terdapat seorang pengembang *web* yang memiliki kapabilitas tersendiri untuk membuat sebuah *website*, seperti contohnya sistem *electronic prescribing* pada situs *web*. Sistem *electronic prescribing* merupakan sistem

pengiriman resep obat melalui media digital yang mengaitkan informasi antara para dokter, sistem pembuatan resep elektronik, apotek dan departemen keuangan dalam ranah kesehatan. Sistem preskripsi elektronik seperti ini tidak hanya mengirim informasi antara dokter dan sistem resep elektronik, tetapi juga memfasilitasi aliran informasi yang komprehensif [5].

Di Indonesia, penggunaan *electronic prescribing* sudah banyak diterapkan dan dianjurkan dalam praktik peresepan dokter di rumah sakit dan klinik karena dapat mengurangi kesalahan pengobatan yang disebabkan oleh resep manual. Namun, tidak semua layanan kesehatan menerapkan sistem *electronic prescribing* [6]. Saat ini proses pembuatan resep obat dokter pada beberapa klinik masih secara manual. Sehingga terkadang terdapat tulisan tangan yang sulit dibaca atau kesalahan dalam penulisan.

Sedangkan proses pembuatan resep obat manual masih diberlakukan pada klinik, termasuk di klinik Asy-Syifa. Sebagai salah satu layanan kesehatan yang bertujuan untuk meningkatkan kesehatan, serta menyembuhkan dan memulihkan kondisi kesehatan pasien. Pelayanan kesehatan juga ingin meningkatkan efisiensi dan keberhasilan dalam proses peresepan obat di klinik. Salah satunya adalah saat pembuatan resep obat dan juga keselamatan pada pasien.



Gambar 1.1 Wawancara Dengan Dokter

Berdasarkan hasil observasi yang dilakukan dan hasil wawancara dengan dokter di klinik Asy-Syifa yang terdapat pada gambar 1.1, bahwa di klinik Asy-Syifa masih menggunakan proses manual saat pembuatan resep obat. Dokter menuliskan resep obat secara manual pada kertas, seperti nama obat, dosis, bentuk obat, penggunaan dan intruksi penggunaan lainnya, lalu dokter memberikan resep obat kepada apoteker yang ada di dalam klinik setelah meyelesaikan penulisannya.

Dan apoteker meracik resep obat sesuai instruksi yang ditulis oleh dokter sebelumnya, setelah obat siap, apoteker akan menyerahkan obat kepada pasien, untuk lebih lengkapnya terdapat pada lampiran 3.

Dikarenakan pembuatan resep obat secara manual di klinik Asy-Syifa pada kertas menimbulkan beberapa masalah yang pertama yaitu kesalahan saat penulisan resep obat oleh dokter yang mengakibatkan nantinya kesalahan meracik obat yang tidak sesuai dan harus mengulang pembuatan resep obat, dan waktu tunggu pasien yang cukup lama, beberapa pasien mengeluh dikarenakan waktu tunggu yang lama memberikan pasien tersebut jadi kurang senang dalam layanan di klinik Asy-Syifa. Adapun salah satu penyebab pada masalah tersebut yaitu pembuatan resep obat sampai menjadi obat secara manual membutuhkan waktu cukup lama untuk rata-rata untuk satu pasien yaitu 19 menit, tergantung berapa banyak obat yang dibuat. Pada tabel 1.1 Merupakan tahapan dan waktu yang diperlukan dari dokter menuliskan resep obat secara manual sampai apoteker menyerahkan obat ke pasien.

Tabel 1.1 Waktu proses pembuatan resep sampai jadi obat secara manual

No	Tahapan	Waktu yang dibutuhkan (menit)
1	Dokter menulis data pasien dan menulis resep obat	3-4
2	Dokter menyerahkan resep obat ke apoteker	2
3	Apoteker melihat resep dari dokter lalu meracik obat sesuai intruksi yang ditulis oleh dokter	10-15
4	Apoteker menyerahkan obat ke pasien dan menjelaskan intruksi penggunaan	2
	Total waktu minimum	16 menit
	Total waktu maksimum	23 menit

Proses waktu pasien sampai mendapatkan obat memakan waktu yang cukup lama karena dokter harus menuliskan data pasien dan data resep obat secara berkali-

kali dengan menuliskan intruksi penggunaanya, terlihat pada tabel 1.1 waktu dokter menulis data pasien dan menulis resep memakan waktu 3-4 menit tergantung resep yang ditulis, setelah itu dokter menyerahkan resep yang ditulis kepada apoteker yang memiliki waktu 2 menit, setelah itu apoteker akan meracik obat sesuai intruksi yang ditulis oleh dokter dengan waktu 10-15 menit tergantung obat yang dibuat, dari waktu pembuatan resep obat menjadi obat yang sudah dijabarkan melalui tabel 1.1 menggunakan metode manual dalam pembuatan resep itu kurang efisien, dan dokter juga pernah mengalami kesalahan penulisan resep yang nantinya mengakibatkan waktu pembuatan obat semakin lama.

Dalam mengurangi kesalahan saat pembuatan resep obat dan mengurangi waktu tunggu pasien dikarenakan pembuatan resep obat yang dilakukan menggunakan proses secara manual dengan tujuan untuk meningkatkan efisiensi dalam seluruh pelayanan kesehatan, mulai dari penulisan resep obat, perlu adanya sistem yang membuat resep obat tersebut secara elektronik atau sistem *electronic prescribing*, adanya *electronic prescribing* diharapkan bisa mengurangi kesalahan seperti penulisan resep obat sehingga meningkatkan akurasi dalam pengobatan pasien dan mempercepat proses peresepan dan pembuatan obat pada pada klinik, serta mengurangi waktu tunggu pasien [7].

Tujuan dari penelitian ini adalah mengembangkan sebuah *website* sistem *electronic prescribing* dokter di klinik Asy-Syifa, yang menggabungkan teknologi *web* dan *database* untuk mengurangi terjadinya kesalahan dalam penulisan resep obat. Metode yang akan diterapkan didalam penelitian ini adalah metode *Extreme Programming*, yang merupakan metodologi yang digunakan untuk pengembangan perangkat lunak yang terjadi secara berkelanjutan dan berulang. Dengan menerapkan metode *Extreme Programming*, diharapkan sistem *electronic prescribing* tersebut dapat dikembangkan dengan langkah-langkah yang terstruktur dan memberikan hasil yang lebih baik. *Extreme Programming* merupakan suatu metode dalam pengembangan perangkat lunak yang menekankan pada penyederhanaan prosesnya, bertujuan untuk membuat proses pengembangan perangkat lunak lebih efisien [8].

Dalam upaya pengembangan situs *web* ini, untuk menekankan kualitas dan mempermudah perbaikan jika ada *bug* dapat dilakukan perbaikan sejak awal. Platform *electronic prescribing* dokter ini dikembangkan menggunakan bahasa pemrograman *Javascript*, bagian *frontend* dari aplikasi ini dikembangkan dengan bahasa pemrograman *JavaScript* menggunakan *framework ReactJS*, yang bertugas untuk mengelola antarmuka pengguna. Sementara itu, bagian *backend* dikembangkan dengan bahasa pemrograman *JavaScript* menggunakan *Express.js* pada runtime *Node.js*, yang memfasilitasi pengelolaan *server* dan *API*, dan untuk *database* menggunakan *MySQL*. Sistem ini akan memungkinkan pembuatan sistem *website* yang dapat digunakan oleh dokter dan apoteker dalam proses sistem *electronic prescribing*. Sistem ini juga dirancang untuk mempermudah penggunaan oleh dokter dan apoteker, sehingga mereka dapat memanfaatkan sistem ini dengan mudah dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, penulis menyajikan rumusan masalah yang bisa diajukan dalam penelitian ini adalah proses pembuatan resep obat kurang efisien dalam segi waktu di klinik Asy-Syifa.

1.3 Pertanyaan Penelitian

Dari rumusan masalah yang telah disajikan, maka pertanyaan peneliti dalam penelitian ini adalah bagaimana mengatasi pembuatan resep obat di klinik Asy-Syifa yang kurang efisien?

1.4 Batasan Masalah

Dari rumusan masalah dan tujuan penelitian, didapat batasan-batasan dalam penelitian sebagai berikut:

1. Sistem ini akan digunakan khusus di Klinik Asy-Syifa Desa Randudongkal.
2. Sistem ini tidak mencakup pelayanan medis lainnya, seperti jadwal dokter dan manajemen keuangan atau aspek non-medis lainnya.

3. Pengembangan sistem menggunakan metode *Extreme Programming*.
4. *Website* dikembangkan menggunakan Bahasa pemrograman *JavaScript* dan *database MySQL*.

1.5 Tujuan Penelitian

Berdasarkan pertanyaan penelitian, tujuan dari penelitian ini dapat diuraikan sebagai berikut meningkatkan efisiensi waktu saat pembuatan resep obat di klinik Asy-Syifa.

1.6 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat yang dengan menyediakan solusi efisien untuk implementasi sistem *electronic prescribing* dokter di Klinik Asy-Syifa Desa Randudongkal. Manfaat adanya sistem *electronic prescribing* adalah terjadi peningkatan efisiensi dalam pengelolaan resep di klinik Asy-Syifa Desa Randudongkal. Penelitian ini dapat membantu Klinik Asy-Syifa untuk mengurangi risiko kesalahan dalam penulisan resep. Dengan penggunaan *electronic prescribing*, diharapkan risiko kesalahan dalam pemberian resep seperti tulisan yang sulit terbaca atau kesalahan dosis dapat diminimalkan. Hal ini diharapkan meningkatkan kesehatan pasien dan memberikan manfaat mengenai pelayanan kesehatan bagi Klinik Asy-Syifa di Desa Randudongkal.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pertama, penelitian dengan judul “Perancangan Sistem Informasi Resep Pulang Farmasi Rawat Inap Dengan Metode *E-Prescribing* Pada RS XYZ” oleh Tb. Dedi fuadi, Ahmad Surahmat, Rizki Fatullah [9]. Dalam penelitian ini, peneliti menggunakan metode pengembangan sistem *waterfall* yang menyusun langkah-langkah terstruktur dalam pembuatan perangkat lunak, dimulai dari tahap analisis, desain, pembuatan kode, pengujian, hingga tahap dukungan secara berurutan menggunakan metode *E-Prescribing* untuk memperluas ketersediaan layanan medis dan meningkatkan standar serta efisiensi dari layanan yang telah disediakan. Sistem yang dikembangkan oleh peneliti mencakup penggunaan elektronik dalam penulisan resep untuk obat yang bertujuan untuk meningkatkan kualitas serta memperbaiki efektifitas layanan kesehatan yang diberikan. Hasil peneliti dimana menunjukkan adanya perbedaan signifikan dalam waktu tunggu pasien sebelum dan setelah penggunaan resep elektronik. Ini menandakan bahwa perubahan menuju sistem peresepan elektronik memiliki dampak positif yang signifikan pada kualitas pelayanan terkait terhadap obat yang mereka butuhkan. Hal ini berpengaruh pada pelayanan pasien karena obat yang diberikan lebih cepat dan juga menghindari kesalahan dalam membaca resep.

Kedua penelitian dengan judul “Peneran Sistem *E-Prescribing* dan *Barcode System* di Rumah Sakit Pertamina Menggunakan *PowerBuilder* Dengan *Arsitektur Client - Server System*” oleh Try Viananda Nova M, S. Kom., M. Kom, dan Rina Alfah, S. Kom., M. Kom [10]. Hasil dari penelitian terkait pengembangan di Rumah Sakit Pertamina Tanjung, aplikasi *E-Prescribing* dan sistem *Barcode* dibuat menggunakan *Power Builder v12.6* dan memanfaatkan *database stbase* menunjukkan bahwa aplikasi *E-Prescribing* tersebut memberikan manfaat yang signifikan bagi instalasi farmasi di rumah sakit tersebut. Aplikasi ini menampilkan kenyamanan dalam proses menerima resep yang diberikan oleh dokter kepada para

pasien yang ditangani, dan juga mengurangi risiko kesalahan dalam pemberian obat yang mungkin terjadi. Peneliti menggunakan metode *Rapid Application Development (RAD)* dalam pengembangan aplikasi, pendekatan yang digunakan juga memiliki kecepatan secara visual, kekuatan yang signifikan, dan kemudahan penggunaan yang optimal. Penelitian yang dilakukan melibatkan eksperimen dan penerapan guna menciptakan sistem yang menyatukan dokter dengan asisten atau apoteker, mengatur informasi tentang ketersediaan stok obat, memberikan informasi tentang indikasi dan kegunaan obat.

Ketiga, penelitian dengan judul “Rancang Bangun Aplikasi Elektronik Resep yang Terhubung ke Kasir Menggunakan *Website*” oleh Helfy Susilawati, Tri Arif Wiharso, Teddy Mulyadi Hidayat [11]. Penelitian ini menggunakan resep elektronik yang terhubung secara langsung ke sistem kasir, merupakan langkah peningkatan dalam layanan kesehatan karena memungkinkan pasien untuk melakukan pembayaran hanya sekali saja. Dalam praktik konvensional, pasien hanya memberikan resep ke kasir, menunggu perhitungan jumlah pembayaran, dan kemudian melakukan pembayaran ke kasir. Tetapi, melalui penerapan resep elektronik yang saling terkait secara langsung ke kasir, pasien hanya perlu menunggu hingga nomor antrian atau nomor pasien mereka dipanggil sebelum melakukan pembayaran untuk obatnya. Peneliti menerapkan metode pembuatan aplikasi tersebut menggunakan metode *prototype*, yang sesuai untuk sistem atau perangkat yang akan mengalami perkembangan lebih lanjut. *Prototype* akan berkembang sesuai dengan keinginan pengguna dan akan membantu pengembangan oleh para pengembang.

Keempat, penelitian dengan judul “Rancang Bangun Sistem *Electronic Prescribing* Dokter dengan Menggunakan *Codeigniter*” oleh Dian Rakasiwi, Ria Arafiyah, Fariani Hermin Indiyah [12]. Tujuan dari penelitian ini adalah menerapkan sistem *electronic prescribing* guna meningkatkan aksesibilitas layanan kesehatan serta meningkatkan mutu dan efisiensi layanan yang disediakan pada Rumah Sakit Gigi dan Mulut *TNI-AU* Jakarta. Pengembangan sistem tersebut melibatkan pendekatan metode *waterfall*, yang melibatkan langkah-langkah dari analisis kebutuhan, perancangan sistem, penulisan kode program, implementasi

perangkat lunak, dan pemeliharaan sistem. Peneliti memanfaatkan Bahasa pemrograman *PHP* dengan framework *Codeigniter* yang mengadopsi pola *MVC* (*Model View Controller*). Penelitian ini bertujuan untuk mengalihkan cara tradisional pencatatan resep dokter menjadi sistem resep elektronik bertujuan untuk meningkatkan efektivitas dan efisiensi, terutama dengan meminimalkan waktu tunggu saat pengambilan obat dan menghindari kesalahan dalam proses pemberian obat.

Kelima, penelitian dengan judul “Sistem Informasi Klinik Berbasis *Website* Menggunakan Metode *Extreme Programming* Pada Klinik Karunia Bunda” oleh Donda Banjarnahor, Nurulfaizah, Kurnia gusti Ayu [13]. Berdasarkan penelitian yang dilakukan, peneliti membangun sistem informasi klinik berbasis *web* dengan menerapkan metode *Extreme Programming*. Hasil dari aplikasi yang dikembangkan oleh peneliti adalah aplikasi yang mampu menghasilkan laporan dan informasi sesuai kebutuhan, memungkinkan manajemen untuk mengambil keputusan yang mendukung perkembangan klinik. Aplikasi yang dikembangkan juga mempermudah seluruh petugas dalam memberikan layanan optimal kepada pasien, terutama dalam pengelolaan iniformasi medis pasien. Peneliti memanfaatkan bahasa pemrograman *PHP* dan menggunakan *database MySQL* untuk mengembangkan aplikasi tersebut.

Keenam, penelitian dengan judul “*Impact of the e-prescribing system on the incidence and nature of drug-related problems in children in a Saudi hospital*” oleh Aeshah A. Alazmia, Hani Alhamdana, Omaima Ahmeda, Stephen Tomlin, Asia N. Rashed [14]. Berdasarkan penelitian yang dilakukan oleh peneliti penggunaan sistem *electronic prescribing* berpengaruh pada anak-anak dan saat menggunakan sistem *electronic prescribing* ini mengurangi angka kesalahan peresepan obat pada anak-anak khususnya pada rumah sakit saudi dan menggunakan sistem *CPOE* untuk menurunkan insiden *DRP* pada anak-anak yang dirawat di *KAMC-J*. Menggabungkan panduan klinis berbasis komputer yang lebih khusus untuk pediatrik ke dalam *CPOE* diperlukan untuk lebih mengurangi masalah ini dan meningkatkan perawatan yang diberikan kepada anak-anak.

Ketujuh, penelitian dengan judul “*Mixed methods study of medication-related decision support alerts experienced during electronic prescribing for inpatients at an English hospital*” oleh Helen Bell, Sara Garfield, Sonia Khosla, Chimnay Patel, Bryony Dean Franklin [15]. Berdasarkan penelitian yang dilakukan oleh peneliti menemukan bahwa sebagian besar peringatan dukungan keputusan diabaikan, namun peringatan untuk *VTE (venous thromboembolism)* jauh lebih kecil kemungkinannya untuk diabaikan ketika muncul di luar putaran *ward*. Temuan ini menunjukkan bahwa peringatan memiliki kemungkinan lebih kecil untuk diabaikan jika mereka terintegrasi dalam alur kerja yang tepat bagi para penulis resep, baik dari segi waktu peringatan maupun para profesional kesehatan yang dituju. Temuan juga menyarankan bahwa peringatan seharusnya mencakup tautan ke tindakan yang diperlukan. Maka dari itu, peneliti menggunakan sistem *electronic prescribing* untuk mengurangi kesalahan penulisan resep obat dari dokter. Bertujuan untuk membantu para penulis resep dalam melakukan preskripsi secara aman, seperti kamus obat, saran dosis default, peringatan interaksi obat-obatan, peringatan alergi obat, dan peringatan terhadap hasil laboratorium yang relevan. Peringatan pop-up sering digunakan untuk menyoroti peringatan kepada penulis resep.

Pada Tabel 2.1 berisikan ringkasan penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan oleh peneliti.

Tabel 2.1 Ringkasan Penelitian Terdahulu

No	Judul	Penulis (tahun)	Hasil	Perbedaan
1	Perancangan Sistem Informasi Resep Pulang Farmasi Rawat Inap Dengan Metode <i>E-Prescribing</i> Pada RS XYZ [9]	Tb. Dedifuadi, Ahmad Surahmat, Rizki Fatullah	Mengembangkan sebuah sistem informasi untuk resep obat di farmasi rawat inap menggunakan pendekatan/metode <i>E-Prescribing</i> dengan bertujuan meningkatkan aksebilitas layanan kesehatan serta meningkatkan efisiensi dari pelayanan yang disediakan.	<i>Website</i> kali ini akan dibuat dengan tampilan yang lebih hidup, dengan menggunakan <i>framework react.js</i>
2	Penerapan Sistem <i>E-Prescribing</i> dan <i>Barcode system</i> di Rumah Sakit Pertamina menggunakan <i>PowerBuilder</i> dengan <i>Arsitektur Client - Server System</i> [10]	Try Viananda Nova M, S.Kom.,M.Kom, Rina Alfah, S.Kom.,M.Kom	Menerapkan sistem <i>E-Prescribing</i> dan <i>Barcode System</i> di Rumah Sakit Pertamina yang bertujuan mempermudah instalasi farmasi di rumah sakit agar lebih efisien dalam menerima dan memproses resep dari dokter dan mengurangi risiko <i>medician error</i> .	Penelitian sebelumnya menggunakan pengembangan sistem <i>Rapid Application Development (RAD)</i> , sedangkan penelitian kali ini menggunakan pengembangan sistem yaitu <i>extreme programming</i>

No	Judul	Penulis (tahun)	Hasil	Perbedaan
3	Rancang Bangun Aplikasi Elektronik Resep yang Terhubung ke Kasir Menggunakan <i>Website</i> [11]	Helfy Susilawati, Tri Arif Wiharso, Teddy Mulyadi Hidayat	Membuat aplikasi elektronik resep yang terhubung kasir dikembangkan dengan metode <i>prototype</i> , tujuan dari penelitian ini adalah meminimalkan waktu tunggu pasien untuk membayar, sehingga hanya dibutuhkan satu kali proses pembayaran setelah panggilan di kasir. Selain itu, resep obat juga didesain agar mudah dibaca oleh pasien dan apoteker.	Pengembangan sistem sebelumnya menggunakan metode <i>prototype</i> , pada penelitian kali ini menggunakan metode <i>extreme programming</i>
4	Rancang Bangun Sistem <i>Electronic Prescribing</i> Dokter dengan Menggunakan <i>Codeigniter</i> [12]	Dian Rakasiwi, Ria Arafiyah, Fariani Hermin Indiyah	Pembuatan sistem <i>electronic prescribing</i> dokter dengan menggunakan <i>framework</i> <i>Codeigniter</i> yang bertujuan meningkatkan ketersediaan layanan kesehatan dan meningkatkan standar serta efisiensi yang diberikan oleh Rumah Sakit.	Penelitian ini sebelumnya menggunakan <i>framework</i> <i>codeigniter</i> , sedangkan penelitian kali ini menggunakan bahasa <i>javascript</i> dengan <i>framework</i> <i>express.js</i>

No	Judul	Penulis (tahun)	Hasil	Perbedaan
5	Sistem Informasi Klinik Berbasis Website Menggunakan Metode <i>Extreme Programming</i> Pada Klinik Karunia Bunda [13]	Donda Banjarnahor, Nurulfaizah, Kurnia Gusti Ayu	Pengembangan sistem informasi klinik berbasis website yang menerapkan metode <i>extreme programming</i> dengan tujuan untuk menyediakan kemudahan bagi semua petugas dalam memberikan layanan yang optima kepada pasien di klinik tersebut.	Peneliti sebelumnya menggunakan bahasa pemrograman <i>PHP</i> , sedangkan Penlitian kali ini menggunakan bahasa pemrograman <i>javascript</i>
6	<i>Impact of the e-prescribing system on the incidence and nature of drug-related problems in children in a Saudi Hospital</i> [14]	Aeshah A. AlAzmia, Hani AlHamdana, Omaima Ahmeda, Stephen Tomlin, Asia N. Rashed	Penelitian dengan peresepan obat dengan sistem <i>electronic prescribing</i> pada anak-anak dengan menggabungkan panduan klinis berbasis komputer yang lebih khusus untuk pediatrik ke dalam <i>CPOE</i> .	Penelitian kali ini menggunakan <i>javascript</i> untuk bahasa pemrogramannya dan menggunakan <i>react.js</i> untuk bagian <i>frontend</i>
7	<i>Mixed methods study of medication-related decision support alerts experienced during electronic prescribing for inpatients at an English hospital</i> [15]	Helen Bell, Sara Garfield, Sonia Khosla, Chimnay Patel, Bryony Dean Franklin	Membangun sistem <i>electronic prescribing</i> yang dilakukan oleh peneliti bertujuan untuk membantu para penulis resep dalam melakukan preskripsi secara aman, saran dosis default, peringatan interaksi obat-obatan, dan peringatan alergi obat.	Penelitian kali ini membuat sistem <i>electronic prescribing</i> menggunakan <i>express.js</i> untuk bagian <i>backend</i> dan <i>MySQL</i> untuk <i>database</i>

2.2 Landasan Teori

Pada penelitian kali ini, diperlakukan pemahaman mengenai beberapa teori yang relevan dan akan digunakan selama proses pengembangan aplikasi sistem *electronic prescribing* dalam penelitian ini.

2.2.1 Rancang Bangun

Rancang bangun merupakan proses konstruksi sistem yang berorientasi pada pembuatan atau perbaikan sistem yang sudah ada, baik secara menyeluruh maupun sebagian, bertujuan untuk meningkatkan efisiensi atau kinerja secara keseluruhan. Rancang bangun merupakan langkah pertama dalam menciptakan gambaran atau sketsa sistem yang belum pernah ada sebelumnya, dan kemudian mengelolanya hingga menciptakan representasi visual atau sketsa yang sesuai dengan kebutuhan fungsional yang diinginkan [16].

2.2.2 Website

Website merupakan koleksi halaman-halaman yang memuat informasi dalam bentuk teks, gambar yang bisa diam atau bergerak, animasi, serta elemen suara, dan kombinasi dari semua elemen tersebut, yang bisa bersifat tetap atau dinamis. Ini juga membentuk suatu struktur bangunan yang saling terkait. Setiap bagian dari struktur ini dihubungkan dengan jaringan halaman, yang terdiri dari sejumlah halaman *web* yang saling terhubung dengan topik yang berhubungan satu sama lain. Di beberapa kasus, halaman *web* juga bisa memuat berkas-berkas seperti gambar, video, atau jenis berkas lainnya [17].

2.2.3 Sistem *Electronic Prescribing*

Sistem *electronic prescribing* atau sistem peresepan elektronik adalah sistem untuk membuat resep elektronik yang dapat langsung dikirimkan ke apotek atau sistem farmasi terkait. Sistem ini menggunakan teknologi informasi untuk menggantikan atau mengurangi penggunaan resep kertas dalam proses pemberian resep obat kepada pasien. Tujuan utama dari penerapan *electronic prescribing* yaitu mengurangi kesalahan medis, mengurangi biaya yang terkait dengan apotek, meningkatkan efisiensi

penulisan resep dan apotek, menghilangkan risiko kehilangan dalam menginterpretasikan tulis tangan, mengurangi komunikasi melalui telepon antara apoteker dan dokter, mengurangi pekerjaan *input* data, serta mempercepat permintaan salinan resep [18].

2.2.4 User Interface

User Interface (UI) adalah suatu cara atau antarmuka yang digunakan untuk memfasilitasi interaksi antara manusia dan sistem komputer. UI sering kali dianggap sebagai pengganti *Human Computer Interaction* (HCI), mencakup semua bentuk interaksi yang dilakukan oleh manusia terhadap komputer. Dalam konteks ini, UI mencakup elemen-elemen seperti tata letak, elemen grafis, dan kontrol yang memungkinkan pengguna berkomunikasi dengan sistem komputer secara efektif. Inti dari UI adalah memberikan pengalaman pengguna yang intuitif dan efisien, memastikan bahwa interaksi antara manusia dan komputer berjalan dengan lancar dan memenuhi kebutuhan pengguna [19].

2.2.5 Database MySQL

Database adalah kumpulan informasi data yang tersimpan dengan terstruktur di dalam komputer untuk memungkinkan pengambilan informasi. Istilah “basis data” pertama kali muncul dalam domain ilmu komputer. Artikel ini membahas basis data komputer, meskipun signifikansinya kemudian berkembang untuk mencakup hal-hal di luar konteks elektronik. Basis data telah ada sebelum periode revolusi industri, bentuk awal dari basis data terdapat dalam buku, kwitansi, dan koleksi informasi bisnis yang terstruktur [20]. *MySQL* ialah sebuah program manajemen basis data yang *open source*. Ini berasal dari konsep utama dalam manajemen basis data yang disebut *SQL (Structured Query Language)*. *SQL* merupakan sebuah konsep yang digunakan untuk mengatur dan mengoperasikan basis data, terutama dalam pemilihan serta pengelompokkan data, sehingga memungkinkan pengoperasian data secara otomatis dengan mudah [21].

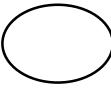
2.2.6 UML

UML (Unified Modeling Language) adalah sebuah bahasa berbasis gambar yang digunakan untuk visualisasi, penjelasan, perancangan, dan dokumentasi sistem perangkat lunak yang menggunakan pendekatan berorientasi objek. UML juga memberikan standar untuk membuat panduan sistem, termasuk konsep proses bisnis, perancangan kelas dalam bahasa pemrograman tertentu, struktur basis data, serta elemen-elemen yang diperlukan dalam pengembangan perangkat lunak [22].

2.2.5.1 Use Case Diagram

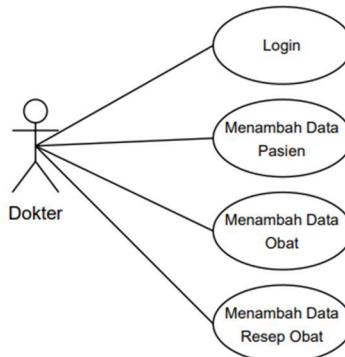
Diagram use case merupakan alat pemodelan yang digunakan untuk menggambarkan bagaimana satu atau lebih aktor berinteraksi dengan sistem informasi yang sedang dikembangkan. Diagram *use case* dipakai untuk menjelaskan bagaimana interaksi antara *actor* (pengguna atau *entitas ekternal*) dengan sistem yang sedang dibangun. Dengan menggunakan *Use case* diagram, proses pemetaan kebutuhan fungsional yang diperoleh dalam tahap analisis menjadi lebih mudah [23]. Nama simbol beserta fungsinya terdapat pada tabel 2.3 berikut ini:

Tabel 2.2 Use-case Diagram [23]

Gambar	Nama	Deskripsi
	Asosiasi	Asosiasi berguna sebagai penghubung antara objek satu dengan objek lainnya.
	<i>Use case</i>	Deskripsi dari suatu fungsi atau tindakan yang dapat dilakukan oleh sistem.
	<i>System</i>	Menampilkan sistem yang sedang dikembangkan atau sedang berjalan.

Gambar	Nama	Deskripsi
	<i>Actor</i>	Aktor merupakan entitas yang berhubungan secara aktif dengan sistem, seperti pengguna atau sistem lainnya.
	<i>Include</i>	Hubungan antara <i>use case</i> yang memasukkan fungsionalitas dari <i>use case</i> lain.
	<i>Extend</i>	Hubungan antara <i>use case</i> yang menambahkan fungsionalitas ke <i>use case</i> lain.

Berikut merupakan contoh penggunaan *use case* diagram yang terdapat pada gambar 2.1.



Gambar 2.1 Contoh Diagram *Use Case* [23]

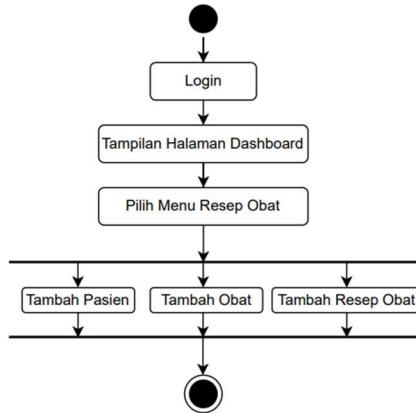
2.2.5.2 Activity Diagram

Activity diagram merupakan suatu bentuk pemodelan yang diterapkan pada sistem yang menggambarkan bagaimana aktivitas dalam sistem berjalan. *Activity* diagram dipakai untuk menguraikan kegiatan dalam sebuah program tanpa harus memeriksa kode atau tampilannya. Diagram ini merepresentasikan proses bisnis dan urutan kegiatan dalam suatu proses. Seperti *flowchart*, diagram ini menggambarkan aliran kerja dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status tertentu dalam sistem [24]. Nama simbol beserta fungsinya terdapat pada tabel 2.4 berikut ini:

Tabel 2.3 Activity Diagram [24]

Simbol	Nama	Keterangan
	Aktivitas	Umumnya dimulai dengan kata kerja, menggambarkan aktivitas yang sedang dilakukan oleh sistem
	<i>Initial Node</i>	Mengawali suatu aktivitas.
	Status Akhir	Mengakhiri alur sistem.
	Penggabungan	Penggabungan yang mana lebih dari satu aktivitas akan digabungkan menjadi satu.
	<i>Decision</i>	Menggambarkan terjadinya percabangan atau ada pilihan aktivitas yang lebih dari satu.
	<i>Swimlane</i>	<i>Swimlane</i> memperlihatkan tanggung jawab pelaksanaan aktivitas dalam suatu diagram atau aktivitas yang terjadi.

Berikut merupakan contoh penggunaan *activity* diagram yang terdapat pada gambar 2.2.



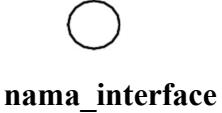
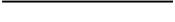
Gambar 2.2 Contoh *Activity Diagram* [24]

2.2.5.3 Class Diagram

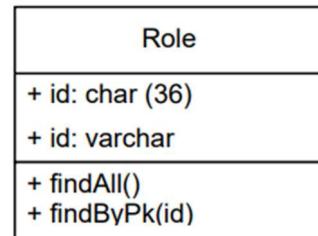
Class diagram merupakan tipe diagram dalam *UML* dipakai untuk menggambarkan struktur kelas dan paket yang terdapat dalam sistem yang direncanakan digunakan. Dalam kata lain, *UML* digunakan untuk mengilustrasikan kelas-kelas dan kelompok-kelompok yang ada dalam sistem tersebut, diagram ini memberikan representasi visual mengenai struktur sistem serta interaksi-interaksi yang ada di dalamnya. *Class* diagram merupakan cara untuk menunjukan kelas-kelas yang ada didalam sistem dan hubungan logis di antara mereka, dan ini memberikan gambaran tentang struktur statis sistem tersebut [25]. Nama simbol beserta fungsinya terdapat pada tabel 2.5 berikut ini:

Tabel 2.4 *Class Diagram* [25]

Gambar	Nama	Deskripsi
	Kelas	Entitas yang merepresentasikan objek dalam suatu sistem atau proses

Gambar	Nama	Deskripsi
	<i>Interface</i>	Mirip dengan ide dari antarmuka dalam paradigma pemrograman berbasis objek
	Asosiasi berarah	Relasi antar kelas yang bermakna generalisasi - spesialisasi (umum khusus)
	Asosiasi	Hubungan antar kelas yang memiliki arti yang umum, sering kali terkait dengan asosiasi yang juga menyertakan konsep <i>multiplicity</i>
	Generalisasi	Hubungan antar kelas bisa terjadi melalui asosiasi dan generalisasi-spesialisasi
	<i>Aggregation</i>	Hubungan yang berarti antara kelas yang mencakup keseluruhan dan bagian-bagiannya (<i>whole-part</i>)
	<i>Dependency</i>	Ketergantungan antarkelas

Berikut merupakan contoh penggunaan *class diagram* yang terdapat pada gambar 2.3.



Gambar 2.3 Contoh *Class Diagram* [25]

2.2.9 Pemrograman *Web*

Pemrograman *web* merupakan hasil penggabungan dua kata, yaitu pemrograman dan *web*. Pemrograman merujuk pada proses atau metode pembuatan program. Sedangkan, *web* merujuk pada suatu rangkaian komputer yang terhubung di jaringan, terdiri dari berbagai situs *internet* yang menyuguhkan teks, gambar, dan sumber daya animasi menggunakan protokol transfer *hypertext*. Istilah umum yang digunakan untuk *web* adalah “*WWW*” (*World Wide Web*), yang merupakan kumpulan halaman *website* yang terhubung satu sama lain melalui hyperlink. *WWW* beroperasi dengan menggunakan *protocol HyperText Transfer Protocol (HTTP)*. Sebuah halaman *web* adalah dokumen teks yang berisi kode *HTML* dan bisa diakses, dilihat atau ditampilkan oleh peramban *internet*. Kode *HTML* memfasilitasi penampilan konten seperti teks, gambar, audio, video, dan animasi [26].

2.2.10 *JavaScript*

JavaScript adalah bahasa pemrograman *web* yang sering dipakai untuk pengembangan di sisi klien (*Client-Side Programming Language*). Merupakan jenis Bahasa pemrograman yang dilakukan oleh *client*, pada konteks ini mengacu pada perangkat seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini*, dan lainnya. Memungkinkan banyak orang untuk melihat dan mengakses isi *Source Code JavaScript*. Seiring dengan perkembangan waktu, *JavaScript* telah berkembang lebih lanjut melalui

kontribusi para pemrogram dan pengembang, sehingga saat ini *JavaScript* dapat digunakan tidak hanya pada *web browser*, tetapi juga dalam *server* ataupun *game* [27].

2.2.11 Framework

Framework adalah sekumpulan instruksi yang dikelompokkan dalam kelas-kelas dan fungsi-fungsi dengan tujuan untuk mempermudah cara penggunaannya oleh para pengembang. Hal ini memungkinkan para pengembang untuk tidak mengulangi penulisan sintaks program serupa secara berulang, menghemat waktu dan usaha [28]. Terdapat jenis-jenis *framework* yang digunakan untuk pengembangan *website*, terdapat dua yang sering digunakan oleh *web developer* adalah *ReactJS* dan *ExpressJS*.

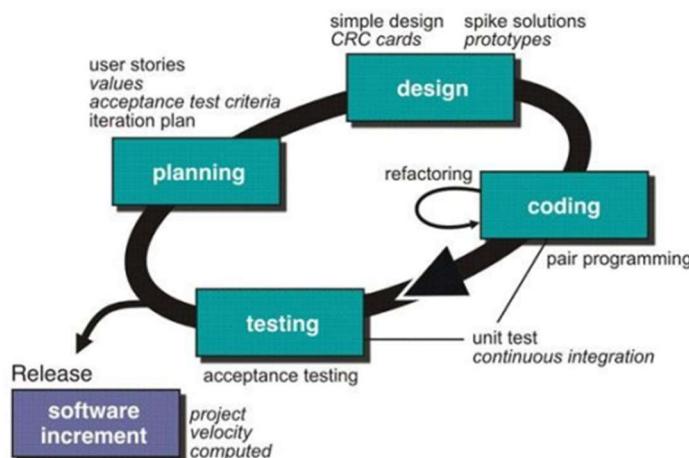
Dalam pembuatan situs *web*, penggunaan *framework* dapat menjadi sarana untuk mempermudah proses pengembangan. Dalam konteks ini, penulis memanfaatkan *framework ReactJS* sebagai alat bantu untuk mengelompokkan *code* program. *ReactJS* merupakan sebuah perpustakaan *JavaScript* sumber terbuka yang diciptakan oleh *Facebook*, yang digunakan untuk konstruksi antarmuka pengguna (*User Interface*). *ReactJS* difokuskan pada pengelolaan seluruh aspek yang terkait dengan tampilan, memungkinkan penggunaannya dalam pembangunan serta pengembangan aplikasi berbasis *web* [29].

ExpressJS adalah sebuah *framework* aplikasi *web* untuk *NodeJS* yang ditulis dalam Bahasa pemrograman *javascript*. *ExpressJS* digunakan untuk membuat aplikasi sisi belakang (*backend*) secara efisien dan optimal. Sebagai *backend web* aplikasi *framework* yang bersifat *open soure* dibawah lisensi *IMT* untuk *NodeJS*, *ExpressJS* didesain khusus untuk membangun aplikasi *web* dan *API* [30].

2.2.12 Metode *Extreme Programming*

Extreme Programming (XP) adalah sebuah metode pengembangan perangkat lunak yang sederhana dan melibatkan praktik-praktik dari metode yang fleksibel yang dipelipori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. *XP* dikenal sebagai salah satu pendekatan metode

tangkas yang sangat terkenal dan umum digunakan. Metode ini menerapkan pendekatan rekayasa perangkat lunak yang cenderung menggunakan pendekatan yang berfokus pada objek [31]. Metode *Extreme Programming* adalah metode yang berfokus pada pembuatan kode aplikasi dengan tujuan menyederhanakan pengembangan sistem agar lebih fleksibel [32]. Pada gambar 2.4 merupakan tahapan dari *extreme programming* yang terdiri dari empat tahapan dan juga penjelasan dari tahapan-tahapan tersebut.



Gambar 2.4 Metode *Extreme Programming* [32]

1. Planning

Tahap *planning* atau perencanaan dimulai dengan langkah awal dalam pembangunan sistem dengan metode *XP*, dimana beberapa kegiatan perencanaan dilakukan, seperti identifikasi permasalahan, analisis kebutuhan, hingga penetapan jadwal pelaksanaan pembangunan sistem [32].

Pada tahap *planning*, perencanaan dilakukan melalui *planning meeting*. Pertemuan dimulai dengan meninjau kebutuhan pengguna, selanjutnya, akan dilakukan penetapan estimasi waktu penyelesaian. Di akhir pertemuan, pengguna membuat keputusan mengenai fitur mana yang akan diimplementasikan [33]. Luaran yang diharapkan dari tahap *planning meeting* dapat berupa *list* masalah ataupun daftar

kebutuhan fitur pengguna [32]. Pada tabel 2.5 merupakan contoh *output* dari tahapan *planning meeting*.

Tabel 2.5 Contoh Kebutuhan Pengguna [34]

Jenis Kebutuhan	
Kebutuhan Pengguna	Kebutuhan Sistem
Pengguna dapat mengakses halaman <i>website</i> dan mengisi <i>form login</i>	Sistem mengalihkan ke halaman <i>dashboard</i> setelah <i>login</i>
Pengguna dapat menambahkan data pasien	Sistem menampilkan data pasien yang ada
Pengguna dapat mencetak resep	Sistem akan kembali halaman <i>login</i> jika menekan tombol keluar

Dalam setiap iterasi, tim pengembangan dan pelanggan bertemu untuk mendefinisikan pekerjaan yang akan diselesaikan melalui pengembangan “*user stories*”. *User Story* merupakan deskripsi pengalaman atau aksi pengguna pada situs *web*. Setiap deskripsi *user story* umumnya terdiri dari satu hingga tiga kalimat. Kegiatan ini merupakan kelanjutan dari perencanaan yang dilakukan pada tahap awal, dimana *user stories* memberikan landasan konkret untuk pekerjaan yang akan dilakukan dalam setiap iterasi. Berikut contoh kalimat *user story* yang terdapat pada tabel 2.6.

Tabel 2.6 Contoh *User Story* [33]

No	User Story
1	Sistem yang dibuat harus ada validasi atau konfirmasi setelah data pasien atau resep obat ditambahkan
2	Sistem yang dibuat harus mampu update otomatis jika proses pembuatan resep obat selesai

2. *Design*

Pada tahap *design* dilakukan perencanaan representasi sistem yang akan digunakan untuk memudahkan *developer* dalam proses pengembangan aplikasi yang umumnya dengan bantuan *Unified Modelling Language (UML)* sebagai modelnya, berbagai jenis diagram seperti diagram *use case*, *activity* diagram dan *class* diagram digunakan untuk mewakili informasi secara visual mengenai bagaimana sistem atau proses bekerja, sementara untuk memodelkan basis data digunakan *class* diagram [35]. Contoh *use case* diagram terdapat pada gambar 2.1, contoh *class* diagram pada gambar 2.2 dan contoh *class* diagram pada gambar 2.3.

3. *Coding*

Tahapan *coding* adalah tahapan yang melibatkan langsung *developer* sistem untuk langsung mengimplementasikan hasil dari desain yang telah sebelumnya telah dibuat. Aktivitas yang terlibat dalam tahap ini mencakup pembuatan basis data dan antarmuka pengguna [34].

Tahapan *coding* dalam pengembangan suatu aplikasi biasanya sistem akan dibagi menjadi dua, yaitu untuk *frontend* dan *backend*. Pada tahapan pengkodean dari sisi *frontend* atau coding akan dilakukan proses penerjemahan dari perancangan *UML* menjadi sebuah *user interface* dan dari sisi *backend* akan merancang basis data menggunakan bahasa pemrograman. Sehingga *output* dari tahapan ini akan berupa tampilan dan fitur dari sebuah sistem yang akan dibangun [36].

4. *Testing*

Pada tahap ini, aplikasi yang dibuat akan dilakukan pengujian atau *testing*. Tahap ini dipengaruhi oleh pengguna sistem yang memprioritaskan fitur dan fungsi keseluruhan sistem, serta dinilai oleh mereka sendiri. Metode yang digunakan untuk menguji

aplikasi sistem *electronic prescribing* dokter berbasis *website* adalah *Blackbox Testing*.

Ada beberapa cara untuk melakukan pengujian sistem, salah satunya dengan menggunakan metode *blackbox*. Keuntungan dari metode ini adalah penentuan penguji atau *tester* tidak harus memiliki pengetahuan akan bahasa pemrograman [36]. Pada tabel 2.7 merupakan contoh halaman dan fitur yang akan dilakukan pengujian menggunakan *blackbox testing*.

Tabel 2.7 Contoh Pengujian Fitur Pada Halaman *Login* [36]

No	Proses	Expected Output
1	Mengakses halaman <i>website</i> di <i>browser</i>	Menampilkan Halaman <i>Login</i>
2	Mengisi <i>form login</i> dan klik tombol <i>login</i>	Mengalihkan pada halaman <i>login</i>

2.2.13 *Blackbox Testing*

Blackbox Testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa memerlukan pemahaman mendalam tentang struktur internal atau bagaimana aplikasi bekerja didalamnya. Dalam pengujian ini, informasi yang digunakan untuk mengembangkan uji kasus berasal dari deskripsi eksternal perangkat lunak, seperti spesifikasi, dan desain. Pengujian ini dapat mencakup baik pengujian fungsional. Perancang uji akan memilih input yang valid dan invalid serta menetapkan hasil yang diharapkan. Pendekatan pengujian ini dapat diterapkan pada berbagai tahapan pengujian perangkat lunak, mulai dari pengujian unit, integrasi, fungsional, sistem, hingga tahap penerimaan [37].

Salah satu metode *Blackbox* yang cukup popular adalah *Equivalence partitioning*, metode ini didasarkan pada pemisahan masukan dan keluaran suatu komponen sesuai spesifikasi. Asumsinya adalah masukan yang sama akan menghasilkan respons yang sama [38]. Pada tabel 2.8 merupakan

contoh *test case* pada pengujian dengan *Equivalence partitioning* pada pengujian *blackbox*.

Tabel 2.8 Contoh *Test Case* [38]

No	Skenario	Ekspektasi Output	Hasil Pengujian
1	Mengakses halaman <i>website</i> di <i>browser</i>	Menampilkan halaman <i>login</i>	Berhasil
2	Mengisi <i>form login</i> dan klik tombol <i>login</i>	Mengalihkan ke halaman <i>dashboard</i>	Berhasil

Hasil dari metode ini akan melihat apakah fungsi pada suatu halaman sudah bekerja dengan dan sesuai harapan. Dengan demikian akan diketahui fitur apa saja yang belum sesuai dengan ekspektasi yang diharapkan dan akan diperbaiki [39]. Pada Persamaan 2.1 merupakan perhitungan nilai kesuksesan dalam pengujian diperoleh dengan perhitungan sebagai berikut [38].

$$\text{Nilai Kesuksesan} = \frac{\text{Pengujian sukses}}{\text{Total pengujian}} \times 100 \quad (2.1)$$

Berdasarkan ketentuan perhitungan diatas akan diketahui presentase kesuksesan pengujian halaman dan fitur didalamnya dengan cara membandingkan jumlah pengujian sukses dengan banyaknya pengujian yang dilakukan [38].

2.2.14 Rasio Efisiensi

Pada perhitungan hasil efisiensi tahap pertama menggunakan perhitungan rata rata dari setiap waktu atau barang yang digunakan untuk menghitung rata rata efisiensi yang akan dihitung nantinya. Berikut untuk membandingkan rasio efisiensi dengan menghitung rata-rata sebelum dan

rata-rata setelah digunakan [40]. Yang pertama menghitung rata-rata waktu yang dapat dilihat pada Persamaan 2.2.

$$Rata - rata waktu = \frac{Total\ waktu\ minimum + Total\ waktu\ maksimum}{2} \quad (2.2)$$

Setelah menghitung rata-rata waktu, selanjutnya bisa menghitung perbedaan rata-rata untuk melihat perbedaan sesuai rata-rata yang ada dengan melihat Persamaan 2.3.

$$Perbedaan\ rata - rata = rata - rata\ sebelum - rata - rata\ setelah \quad (2.3)$$

Setelah itu bisa menghitung untuk persentase perbedaan untuk melihat persentase yang didapatkan yang sesuai dengan melihat Persamaan 2.4.

$$Persentase\ perbedaan = \left(\frac{Perbedaan\ rata - rata}{Rata - rata\ sebelum} \right) \times 100\% \quad (2.4)$$

Setelah mendapatkan hasilnya, selanjutnya bisa menghitung rasio efisiensi untuk melihat rasio efisiensi yang sesuai dengan persamaan 2.5

$$Rasio\ efisiensi = \frac{Rata - rata\ sebelum}{rata - rata\ setelah} \quad (2.5)$$

Setelah menemukan rasio efisiensi maka selanjutnya menghitung persentase peningkatan efisiensi, pada Persamaan 2.6 untuk menghitung persentase peningkatan efisiensi dengan rasio efisiensi dan rasio efisiensi sebelum untuk melihat berapa persen meningkat untuk tingkat efisiensinya [40].

$$persentase\ peningkatan\ efisiensi = \left(\frac{rasio\ efisiensi - rasio\ efisiensi\ sebelum}{rasio\ efisiensi\ sebelum} \right) \times 100\% \quad (2.6)$$

Setelah dihitung maka terlihat apakah persentase meningkat atau menurun dalam perhitungan persentase peningkatan efisiensi dan peningkatan akan terlihat secara signifikan menggunakan rumus tersebut [40].

BAB III

METODE PENELITIAN

3.1 Subjek Dan Objek Penelitian

3.1.1 Subjek Penelitian

Subjek dalam yang dalam penelitian ini adalah dokter dan apoteker di klinik Asy-Syifa Desa Randudongkal yang berlokasi di Jl. Randudongkal - Belik, Desa Randudongkal, Kabupaten Pemalang.

3.1.2 Objek Penelitian

Berdasarkan informasi latar belakang yang dijelaskan, dapat disimpulkan bahwa objek yang dimanfaatkan dalam penelitian ini adalah sistem *electronic prescribing* dokter untuk klinik Asy-Syifa Desa Randudongkal.

3.2 Alat Dan Bahan

Dalam penelitian ini, beragam peralatan dan bahan digunakan untuk memastikan bahwa penelitian dapat berjalan dengan lancar dan sukses. Jenis alat dan bahan akan dijabarkan dengan rincian sebagai berikut:

3.2.1 Alat

Alat yang ada dalam penelitian ini dikelompokkan kedalam dua jenis, yaitu *hardware* dan *software*.

Tabel 3.1 Spesifikasi Hardware

No	Nama <i>Hardware</i>	Spesifikasi
1	<i>Processor</i>	<i>Intel Pentium Gold 5405U</i>
2	<i>RAM</i>	<i>8 GB</i>
3	<i>ROM</i>	<i>1TB HDD, 512GB SSD</i>

Tabel 3.2 Daftar *Software*

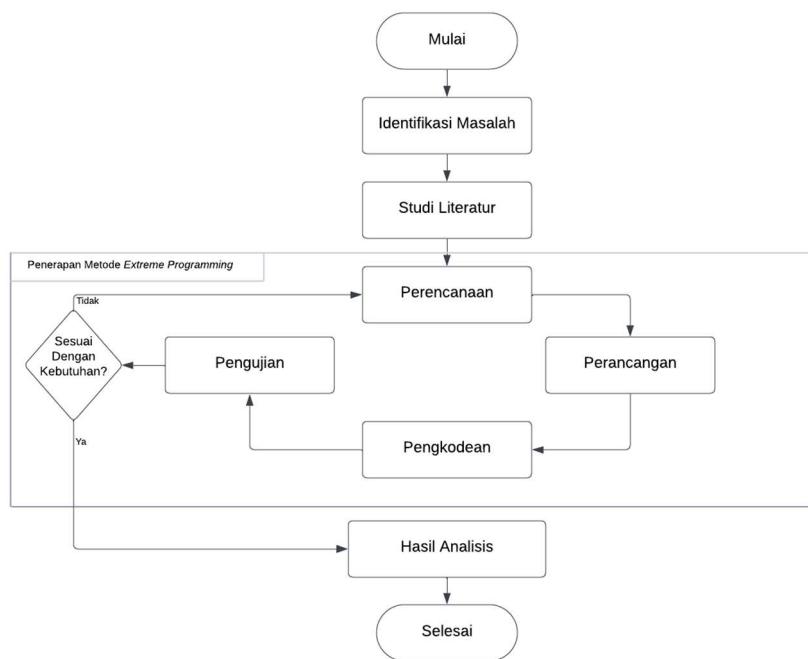
No	Nama	Versi	Manfaat
1	<i>Microsoft Word</i>	365	Untuk menulis laporan
2	<i>Google Scholar</i>	-	Untuk mencari referensi
3	<i>Visual Studio Code</i>	1.90	Sebagai <i>text editor</i>
4	<i>Draw.io</i>	24.4.13	Untuk membuat diagram
5	<i>Brave</i>	1.66.118	Untuk melihat kode program
6	<i>Web server Apache</i>	2.48.58	Untuk <i>server apache</i> pada <i>database</i>
7	<i>Database Mysql</i>	10.4.32	Membuat <i>database</i>

3.2.2 Bahan

Bahan dari penelitian ini berasal dari hasil wawancara dengan dokter yang ada di klinik Asy-Syifa Desa Randudongkal. Wawancara yang sudah dilakukan dapat berfungsi menjadi informasi utama, dan kebutuhan yang perlu diatasi oleh peneliti melalui pembuatan sistem *electronic prescribing* dokter berbasis *website*.

3.3 Diagram Alir Penelitian

Dalam tahap ini, penelitian akan dijelaskan secara detail seluruh kegiatan yang akan dilakukan oleh penulis selama proses penelitian, dan menjadi panduan utama dalam melaksanakan penelitian guna memastikan bahwa hasil dicapai sesuai dengan tujuan yang telah ditetapkan dan tidak menyimpang. Pada gambar 3.1 merupakan tahapan diagram alir penelitian yang dilakukan oleh peneliti.



Gambar 3.1 Diagram Alir Penelitian

3.3.1 Identifikasi Masalah

Pada tahapan ini penulis telah melakukan identifikasi masalah yang terjadi di klinik Asy-Syifa Desa Randudongkal. Hal ini dilakukan dengan cara melakukan observasi dan wawancara untuk mengumpulkan data sebagai pendukung pembuatan sistem *electronic prescribing* dokter sesuai kebutuhan dan juga untuk mengetahui gambaran permasalahan yang dihadapi oleh Klinik Asy-Syifa. Pertanyaan yang disampaikan penulis kepada dokter sebagai bentuk identifikasi masalah dapat dilihat pada lampiran 2.

3.3.2 Studi Literatur

Pada tahapan studi literatur, penulis mengumpulkan informasi dan teori secara ilmiah untuk mengatasi permasalahan yang telah diidentifikasi sebelumnya. Proses studi literatur dilakukan untuk memperkuat fakta dalam penelitian dan dilakukan dengan membaca jurnal penelitian yang sesuai dengan topik penelitian.

3.3.3 Perencanaan

Pada tahapan perencanaan atau *planning*, peneliti akan melakukan identifikasi kebutuhan sistem yang ada dengan mengumpulkan informasi dan menganalisis data yang ada. Dalam konteks pembangunan sistem *electronic prescribing* di klinik Asy-Syifa Desa Randudongkal. Peneliti menyusun daftar kebutuhan yang mencakup aspek-aspek yang diperlukan oleh pengguna, setelah mengetahui kebutuhan pengguna, peneliti menentukan prioritas dari setiap kebutuhan. Pada tabel 3.3 merupakan beberapa pertanyaan yang akan ditanyakan penulis pada tahap perencanaan:

Tabel 3.3 Pertanyaan yang Diajukan

Pertanyaan	Tujuan
Apa kebutuhan utama dari sistem <i>electronic prescribing</i> yang dibutuhkan?	Memastikan bahwa aplikasi yang dikembangkan dapat memenuhi kebutuhan pengguna terutama dalam sistem penambahan data resep obat.
Bagaimana apoteker melihat data resep obat yang ditambahkan oleh dokter?	Menentukan beberapa fitur yang diperlukan untuk meningkatkan kemudahan saat membaca data resep obat.
Selain pencatatan data dari dokter adakah fitur lain yang dibutuhkan dan dapat mempermudah melihat data dari pasien atau resep obat?	Mencari kebutuhan pengguna selain fitur utama untuk mempermudah melihat data dari pasien atau resep obat.

Setelah mendapatkan jawaban dari pihak dokter dan apoteker yang ada di klinik, selanjutnya dibuat *user story* seperti contoh pada tabel 2.6 dan menyepakati lama penggerjaan fitur yang dibutuhkan.

3.3.4 Perancangan

Pada tahap ini akan dibuat *Unified Modeling Language* (UML) berdasarkan kebutuhan yang ada pada tahap perencanaan. Setelah UML dibuat, maka penulis mempersiapkan tampilan *user interface* aplikasi. UML yang akan digunakan pada tahap perancangan adalah *use case* diagram, *activity* diagram dan *class* diagram. Hasil diagram dan tampilan *user interface* yang dibuat akan dijadikan sebagai pedoman selama proses coding.

1. *Use Case Diagram*

Use case dapat digunakan untuk menggambarkan bagaimana pengguna berinteraksi dengan sistem dengan menjelaskan bagaimana sistem tersebut digunakan. Diagram *use case* ini terdiri dari aktor serta aksi yang dilakukan di dalam sebuah sistem. Diagram ini dibuat untuk memenuhi kebutuhan sistem dalam pembuatan sistem *electronic prescribing* dokter pada Klinik Asy-Syifa Desa Randudongkal.

2. *Activity Diagram*

Activity diagram dapat menggambarkan aktivitas dari suatu sistem yang sedang dikembangkan. Diagram ini memberikan gambaran tentang bagaimana aktivitas-aktivitas dimulai, berbagai kemungkinan arah yang bisa terjadi selama proses, dan cara dimana proses tersebut bisa selesai atau berakhir. Diagram ini digunakan untuk mempermudah perancangan sistem *website electronic prescribing* dokter di Klinik Asy-Syifa Desa Randudongkal.

3. *Class Diagram*

Pada *class* diagram digunakan untuk memvisualisasikan struktur sistem dengan menentukan kelas-kelas yang dibuat untuk membentuk struktur basis data dalam program. *Database* yang digunakan disesuaikan dengan kebutuhan sistem *electronic prescribing* dokter di klinik Asy-Syifa Desa randudongkal.

3.3.5 Pengkodean

Pada tahapan pengkodean atau *coding*, peneliti menggunakan teknologi untuk mengembangkan aplikasi *sistem electronic prescribing* dokter. Dalam penelitian ini menggunakan bahasa pemrograman *javascript* dengan *framework react.js* sebagai *frontend* dan *express.js* sebagai *backend framework*. Proses pengkodean atau *coding* dilakukan dengan menggunakan *Visual Studio Code* sebagai *text editor*, dengan sistem manajemen data yang menggunakan *MySQL*.

3.3.6 Pengujian

Pada tahapan ini, peneliti mempersiapkan skenario pengujian yang akan dilakukan dengan *blackbox testing* yang melibatkan perencanaan dan perancangan kasus uji, pelaksanaan pengujian untuk berbagai fitur aplikasi, pemantauan dan pelaporan masalah, serta uji ulang dan pengujian ulang setelah perbaikan. Tahapan ini juga memastikan bahwa aplikasi memenuhi kebutuhan pengguna, dan dapat memberikan pengalaman yang baik kepada pengguna dengan fokus pada fungsionalitas yang berorientasi pada pengguna. Jika pada tahapan pengujian ada yang tidak sesuai, maka akan dilakukan iterasi. Daftar pengujian akan disesuaikan dengan format tabel 2.8.

3.3.7 Hasil Analisis

Pada tahapan hasil analisis, penulis melakukan perhitungan untuk membandingkan waktu yang dibutuhkan pada saat pembuatan resep obat dibuat sampai obat diserahkan kepada pasien dengan sistem *electronic prescribing* dokter yang baru diimplementasikan. Penulis mengidentifikasi waktu yang dibutuhkan untuk setiap tahapan proses pembuatan resep obat, mulai dokter mencatat resep hingga obat tersebut diserahkan kepada pasien. Data waktu yang diperoleh dari masing masing tahapan digunakan untuk menghitung total waktu minimum dan waktu maksimum yang digunakan dan mendapatkan rata rata waktu dalam pembuatan obat untuk satu pasien yang menjadi perhitungan dalam penelitian kali ini. Perbandingan waktu antara sistem pembuatan

resep obat secara manual dan sistem *electronic prescribing* dokter disusun dengan mempertimbangkan efisiensi masing-masing tahapan, dengan penekanan pada penggunaan teknologi untuk mengurangi waktu tunggu pasien, mempercepat pengiriman resep obat ke apoteker, dan mengurangi kesalahan penulisan resep.

BAB IV

HASIL DAN PEMBAHASAN

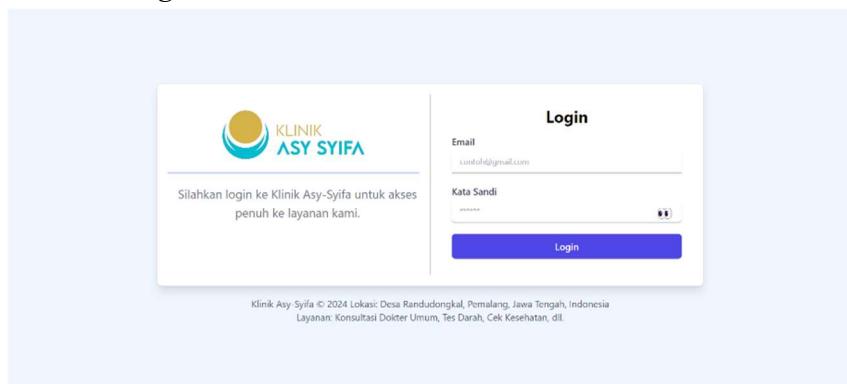
4.1 Hasil

Pada bab hasil merupakan hasil dari tahapan metode penelitian yang telah dijelaskan pada bab sebelumnya. Metode yang digunakan adalah *Extreme Programming* dalam perancangan *website* sistem *electronic prescribing*. Pengujian sistem *website* dilakukan menggunakan *BlackBox Testing* untuk mengevaluasi fungsionalitas fitur-fitur yang ada. Hasil penelitian ini dapat dilihat sebagai berikut:

1. Hasil Website

Hasil akhir tampilan *website* aplikasi sistem *electronic prescribing* yang melalui pengujian sebagai berikut:

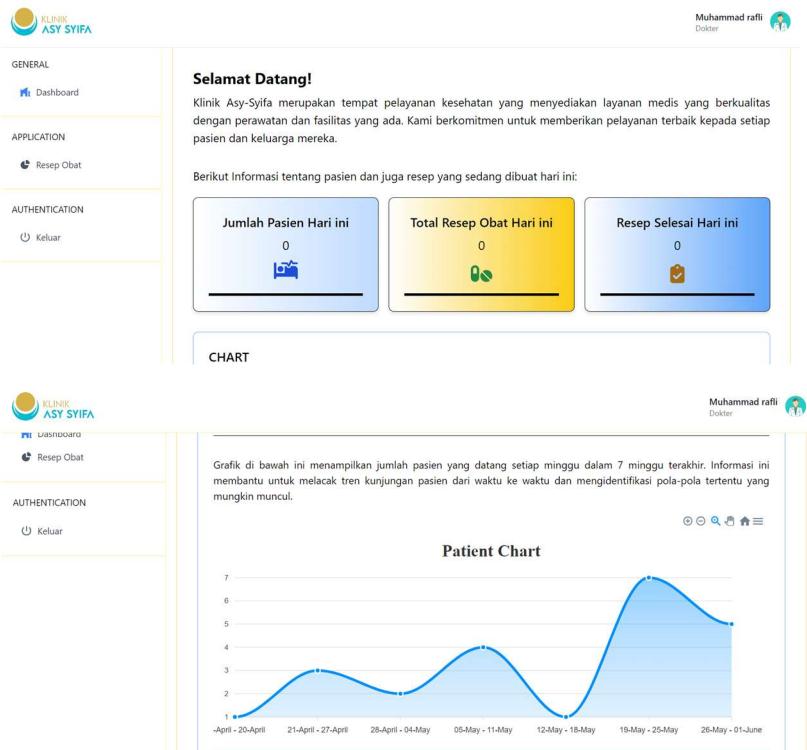
a. Halaman *Login*



Gambar 4.1 Halaman *Login User*

Pada gambar 4.1 merupakan halaman untuk *login* semua role yaitu dokter, apoteker dan *admin*. Sistem akan menampilkan *form* untuk pengisian *email* dan kata sandi, kalau *email* dan kata sandi berhasil akan masuk ke halaman *dashboard* masing-masing role, tapi jika ada yang salah maka sistem akan menampilkan pesan kesalahan.

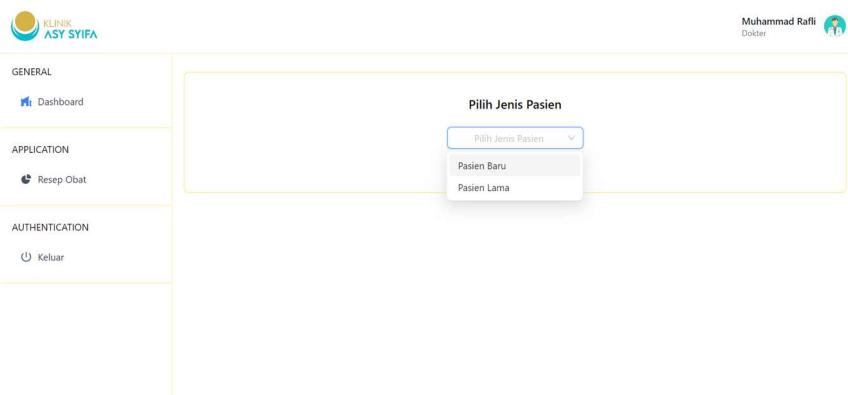
b. Halaman *Dashboard* untuk role dokter



Gambar 4.2 Halaman *Dashboard* Dokter

Pada gambar 4.2 merupakan halaman *dashboard*, dimana pada halaman *dashboard* dokter terlihat yaitu informasi mengenai pasien hari ini, total resep obat hari ini, dan resep selesai hari ini. Dan dibawahnya juga terdapat grafik atau *chart* untuk informasi mengenai pasien dalam kurun waktu tujuh minggu ke belakang dan bisa memilih pada minggu tertentu untuk mendapatkan data pasien per-hari pada satu minggu yang di pilih. Fungsi *dashboard* tersebut adalah agar bisa melihat data pasien yang berkunjung pada klinik dan bisa rekap data untuk keperluan pihak klinik.

c. Halaman Resep Obat Dokter



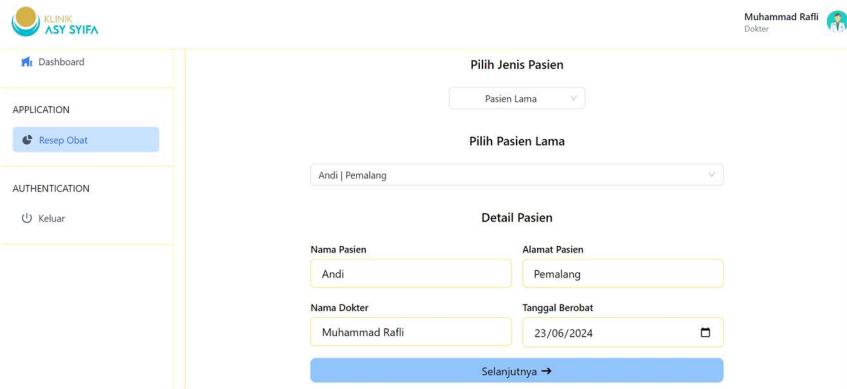
Gambar 4.3 Tampilan Awal Halaman Resep Obat

Pada gambar 4.3 merupakan Tampilan awal dari halaman Resep Obat, pada tampilan awal sistem menampilkan pilihan jenis pasien yaitu ada pasien baru dan pasien lama.

This screenshot shows the 'Pilih Pasien Baru' (Select New Patient) screen. It features a 'Form Data Pasien Baru' section with fields for 'Nama Pasien' (Zidan) and 'Alamat Pasien' (Bantarbolan), and another section for 'Nama Dokter' (Muhammad Rafli) and 'Tanggal Berobat' (26/06/2024). A large blue button at the bottom right contains the text 'Selanjutnya →'.

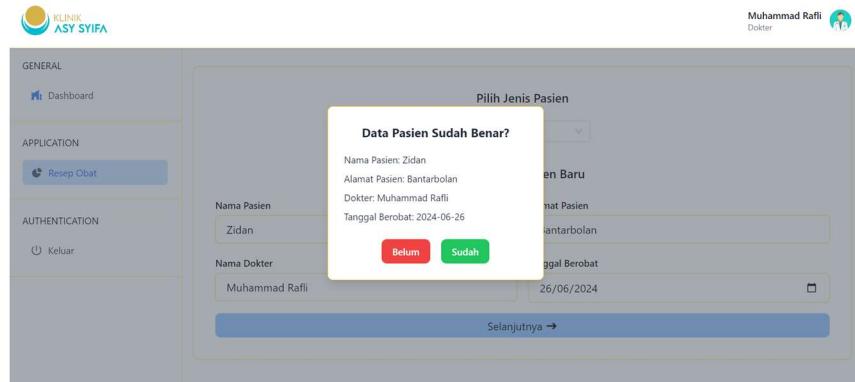
Gambar 4.4 Tampilan Awal Pilih Pasien Baru

Pada gambat 4.4 merupakan tampilan saat dokter memilih pasien baru terdapat *form* yang masih harus diisi oleh dokter seperti nama pasien, alamat pasien, dan tanggal berobat



Gambar 4.5 Tampilan Awal Pilih Pasien Lama

Pada gambar 4.5 merupakan tampilan awal pada halaman Resep Obat jika dokter memilih pasien lama, nantinya dokter bisa mencari pasien dan setelah sudah dipilih maka bisa melihat detail pasien lama dan nantinya bisa menekan selanjutnya untuk lanjut ke halaman berikutnya.



Gambar 4.6 Konfirmasi Data Pasien Sudah Benar

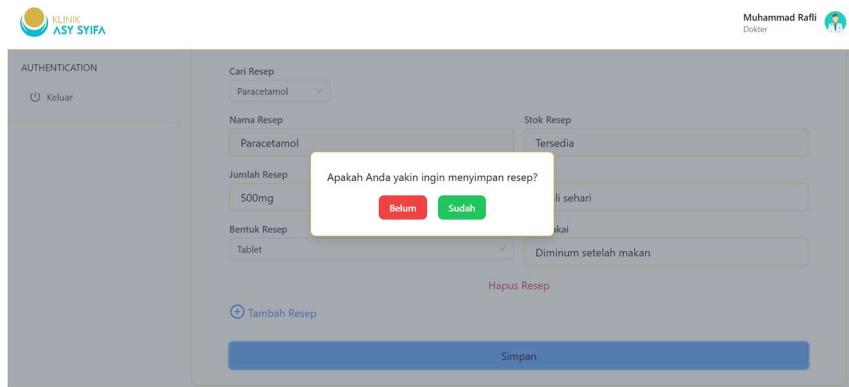
Pada gambar 4.6 merupakan Tampilan konfirmasi apakah data yang diinputkan sudah benar, jika belum maka akan tetap di tampilan *form* data pasien, jika tekan sudah maka akan ke tampilan berikutnya.

Gambar 4.7 Tampilan Kedua Halaman Resep Obat

Pada gambar 4.7 merupakan tampilan kedua pada halaman resep obat, dimana sistem menampilkan *form* resep obat yang pertama berisi nama obat dan jumlah obat, lalu dokter bisa menambah resep untuk menambahkan resep bagian obatnya.

Gambar 4.8 Tambah Resep Pada *Form* Resep Obat

Pada gambar 4.8 merupakan Tampilan tambah resep untuk form resep obat dokter bisa menambahkan resep obat sesuai obatnya dan setelah mengisi beberapa form tersebut dokter bisa klik simpan.



Gambar 4.9 Konfirmasi Simpan Resep

Pada gambar 4.9 merupakan tampilan konfirmasi apakah data siap untuk disimpan atau belum jika dokter menekan sudah maka akan kembali tampilan awal pada halaman Resep Obat.

d. Halaman *Dashboard* untuk role apoteker



Gambar 4.10 Halaman *Dashboard* Apoteker

Pada gambar 4.10 merupakan halaman *dashboard* untuk apoteker, sistem akan menampilkan *dashboard* yang nanti bisa dilihat oleh apoteker. Untuk *dashboard* apoteker sama seperti *dashboard* pada dokter, yang memperlihatkan informasi mengenai jumlah pasien hari ini, total resep obat hari ini, resep selesai hari ini, dan juga ada *chart* atau grafik pasien.

e. Halaman Stok Resep

The screenshot shows the 'Stok Resep' (Stock Prescription) page. At the top right, there's a logo for 'Viky Indika Tintatiana Apoteker'. The main area has a table with columns: No, Nama Resep, Stok, Status Stok, Tanggal Pembaruan, and Aksi. The table contains three rows of data:

No	Nama Resep	Stok	Status Stok	Tanggal Pembaruan	Aksi
1	Salicylamide	900 mg	Tersedia	26/6/2024	Edit Hapus
2	Chlorpheniramine	900 mg	Tersedia	26/6/2024	Edit Hapus
3	Paracetamol	400 mg	Hampir Habis	26/6/2024	Edit Hapus

A modal window titled 'Tambah Resep' (Add Prescription) is overlaid on the table. It contains fields for 'Nama Resep' (Paracetamol), 'Stok' (400), 'Satuan' (mg), and 'Tanggal Pembaruan' (2024-06-26). There are 'Cancel' and 'OK' buttons at the bottom of the modal.

Gambar 4.11 Halaman Stok Resep

Pada gambar 4.11 merupakan halaman stok resep nantinya apoteker bisa melihat data resep dan melihat stok yang ada pada, nantinya apoteker bisa menambah data resep, mengedit dan menghapus data resep.

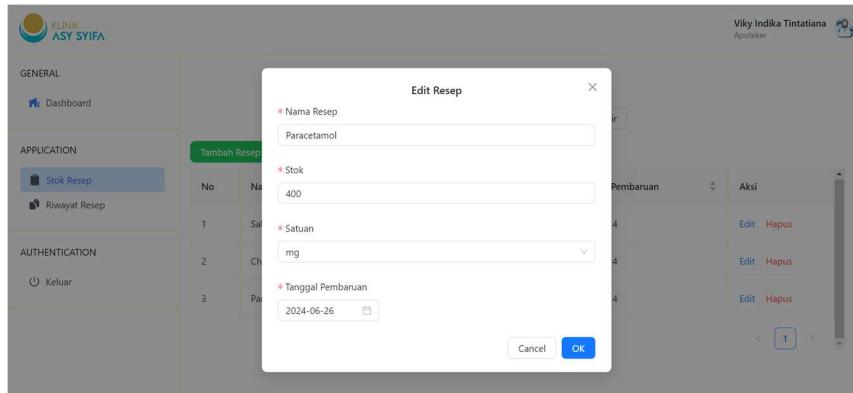
The screenshot shows a 'Tambah Resep' (Add Prescription) dialog box. It contains the following fields:

- * Nama Resep: Paracetamol
- * Stok: 400
- * Satuan: mg
- * Tanggal Pembaruan: 2024-06-26

At the bottom of the dialog are 'Cancel' and 'OK' buttons. The 'OK' button is highlighted with a blue border.

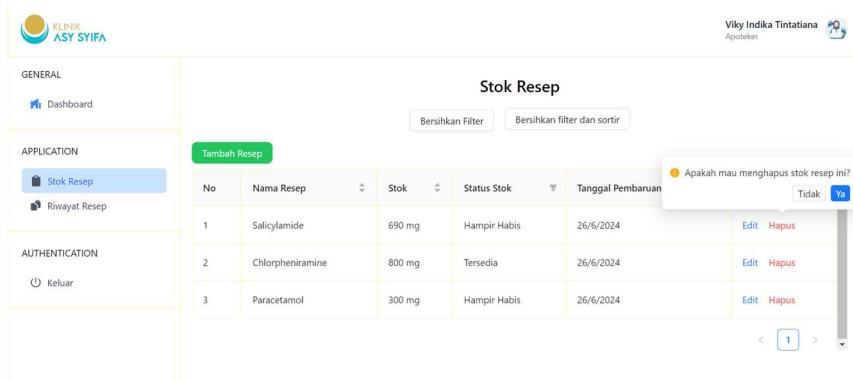
Gambar 4.12 Tampilan Pop Up Pada Tambah Resep

Pada gambar 4.12 merupakan tampilan *pop up* pada tambah resep, nantinya apoteker bisa menambahkan resep yang ada, setelah mengisi semua *form* bisa menekan ok untuk menyimpan datanya.



Gambar 4.13 Tampilan Pop Up Pada Edit Resep

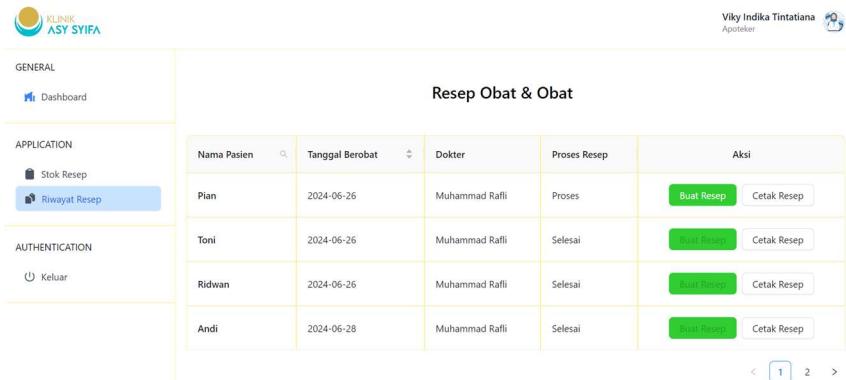
Pada gambar 4.13 merupakan tampilan edit pada resep dimana nantinya apoteker bisa mengedit seperti stok ataupun nama resep. Jika sudah selesai bisa menekan ok untuk mengupdate data resep tersebut.



Gambar 4.14 Konfirmasi Data Resep Dihapus

Pada gambar 4.14 merupakan tampilan saat ada data resep yang mau dihapus pada halaman stok resep, saat sudah menekan hapus pada nama resep yang dipilih maka terdapat konfirmasi “Apakah mau menghapus stok resep ini?”. Saat menekan Ya maka ada notif “Stok resep berhasil dihapus”, dan data nantinya akan terhapus.

f. Halaman Riwayat Resep



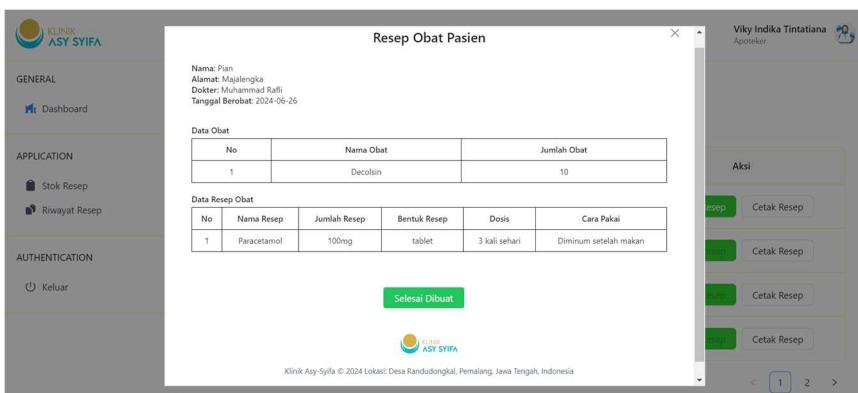
The screenshot shows a web-based application interface for a pharmacy. At the top left is the logo 'KLINIK ASY SYIFA'. At the top right, it says 'Vicky Indika Tintatiana Apoteker' with a profile icon. The main title 'Resep Obat & Obat' is centered above a table. The table has columns for 'Nama Pasien', 'Tanggal Berobat', 'Dokter', 'Proses Resep', and 'Aksi'. There are four rows of data:

Nama Pasien	Tanggal Berobat	Dokter	Proses Resep	Aksi
Pian	2024-06-26	Muhammad Raffi	Proses	<button>Buat Resep</button> <button>Cetak Resep</button>
Toni	2024-06-26	Muhammad Raffi	Selesai	<button>Buat Resep</button> <button>Cetak Resep</button>
Ridwan	2024-06-26	Muhammad Raffi	Selesai	<button>Buat Resep</button> <button>Cetak Resep</button>
Andi	2024-06-28	Muhammad Raffi	Selesai	<button>Buat Resep</button> <button>Cetak Resep</button>

At the bottom right of the table area, there are navigation buttons: '<' (left), '1' (page 1), '2' (page 2), and '>' (right).

Gambar 4.15 Halaman Riwayat Resep

Pada gambar 4.15 merupakan halaman riwayat resep untuk apoteker, pada halaman tersebut apoteker bisa melihat data pasien yang sudah diinputkan oleh dokter. Dan nantinya apoteker membuat obat dengan melihat data tersebut saat sudah tekan Buat Resep dan juga terdapat tombol cetak resep untuk mencetak apabila data tersebut dibutuhkan.



Gambar 4.16 Tampilan Pop Up Buat Resep

Pada gambar 4.16 merupakan tampilan *pop up* pada saat apoteker menekan “Buat Resep”, dimana apoteker bisa melihat data pasien, data obat, dan data resep obat, saat muncul *pop up* proses resep yang sebelumnya “Belum Diproses” menjadi “Proses”, dan setelah apoteker sudah siap dengan obatnya maka apoteker bisa tekan Selesai Dibuat untuk menyelesaikan proses

resepnya dan proses resep pada tabel akan berubah menjadi “Selesai”.

Resep Obat Pasien

Nama: Pian
 Alamat: Majalengka
 Dokter: Muhammad Rafli
 Tanggal Berobat: 2024-06-26

Data Obat

No	Nama Obat	Jumlah Obat
1	Decolsin	10

Data Resep Obat

No	Nama Resep	Jumlah Resep	Bentuk Resep	Dosis	Cara Pakai
1	Paracetamol	100mg	tablet	3 kali sehari	Diminum setelah makan



Klinik Asy-Syifa © 2024 Lokasi: Desa Randudongkal, Pemalang, Jawa Tengah, Indonesia

Gambar 4.17 Hasil Cetak Resep Obat Pasien

Pada gambar 4.17 merupakan hasil cetak resep obat pasien yang mungkin dibutuhkan oleh apoteker maupun dokter untuk data ataupun untuk rekap data resep obat pada pasien.

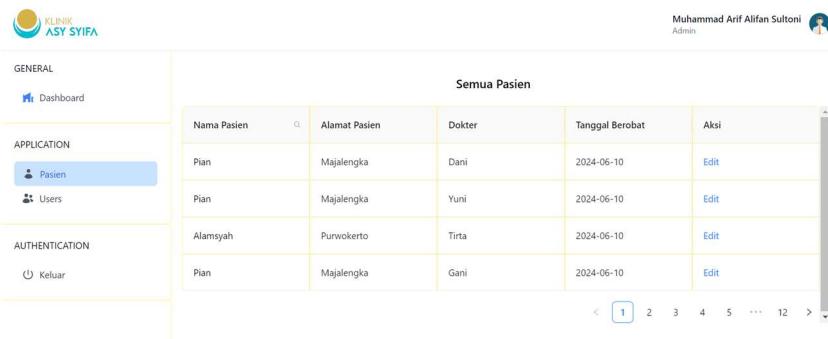
g. Halaman *Dashboard* untuk role *admin*

The dashboard features a sidebar with navigation links for General (Dashboard), Application (Pasien, Users), and Authentication (Keluar). The main area displays a welcome message "Selamat Datang!" and information about the clinic's commitment to quality service. It also shows three summary boxes: "Jumlah Pasien Hari ini" (0), "Total Resep Obat Hari ini" (0), and "Resep Selesai Hari ini" (0). A chart section is partially visible at the bottom.

Gambar 4.18 Halaman *Dashboard* Role *Admin*

Pada gambar 4.18 merupakan halaman *dashboard admin*, yang sama seperti halaman *dashboard* di dokter dan apoteker, yang berisi informasi mengenai data pasien dan juga resep obat.

h. Halaman Pasien

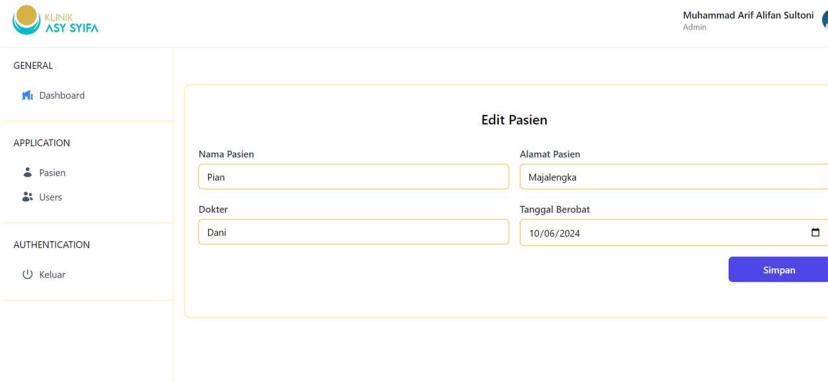


The screenshot shows a web application interface for managing patients. At the top right, there is a user profile for 'Muhammad Arif Alifan Sultan' (Admin). On the left, a sidebar menu includes 'GENERAL' (Dashboard), 'APPLICATION' (Pasien, highlighted in blue), 'Users', and 'AUTHENTICATION' (Keluar). The main content area is titled 'Semua Pasien' and displays a table with columns: Nama Pasien, Alamat Pasien, Dokter, Tanggal Berobat, and Aksi (Edit). The table contains four rows of data. At the bottom right of the table, there is a pagination control with buttons for 1, 2, 3, 4, 5, ..., 12, >.

Semua Pasien				
Nama Pasien	Alamat Pasien	Dokter	Tanggal Berobat	Aksi
Pian	Majalengka	Dani	2024-06-10	Edit
Pian	Majalengka	Yuni	2024-06-10	Edit
Alamsyah	Purwokerto	Tirta	2024-06-10	Edit
Pian	Majalengka	Gani	2024-06-10	Edit

Gambar 4.19 Halaman Pasien

Pada gambar 4.19 merupakan halaman pasien untuk *admin*, pada halaman tersebut, *admin* bisa melihat semua *list* untuk pasien yang sudah diinputkan oleh dokter dan bisa *update* pasien jika diperlukan.



The screenshot shows a modal dialog titled 'Edit Pasien'. It contains input fields for 'Nama Pasien' (Pian), 'Alamat Pasien' (Majalengka), 'Dokter' (Dani), and 'Tanggal Berobat' (10/06/2024). At the bottom right of the dialog is a blue 'Simpan' button.

Gambar 4.20 Tampilan Edit Pasien

Pada gambar 4.20 merupakan tampilan edit pasien di halaman pasien saat admin tekan edit akan muncul halaman baru dan admin bisa edit pasien, setelah admin menginputkan data pasien yang baru maka admin hanya tekan simpan saja. Edit pasien ini hanya jika diperlukan saja.

i. Halaman *Users*

The screenshot shows a web application interface for managing users. On the left, there's a sidebar with 'GENERAL' (Dashboard), 'APPLICATION' (Pasien, Users - highlighted in blue), and 'AUTHENTICATION' (Keluar). The main area is titled 'Semua Pengguna' and contains a table with three rows of user data:

Nama	Email	Role	Active	Aksi
Muhammad Arif Alifan ...	admin@gmail.com	Admin	Active	Edit Nonaktifkan
Muhammad raffi	dokter@gmail.com	Dokter	Active	Edit Nonaktifkan
Viky Indika Tintatiana	apoteker@gmail.com	Apoteker	Active	Edit Nonaktifkan

At the bottom right of the table, there are navigation buttons: '<' (left), '1' (center), and '>' (right).

Gambar 4.21 Halaman *Users*

Pada gambar 4.21 merupakan halaman *users* dimana *admin* nantinya bisa melihat semua pengguna yaitu dokter dan juga apoteker, *admin* juga bisa menambahkan dokter dan apoteker jika diperlukan.

The screenshot shows the 'Add User' form. It has fields for 'Nama' (with placeholder 'Contoh: Dokter'), 'Email' (with placeholder 'Contoh: dokter@gmail.com'), 'Kata Sandi' (with placeholder '*****'), and 'Role' (a dropdown menu with 'Pilih Role'). A 'Simpan' (Save) button is at the bottom right.

Gambar 4.22 Tampilan *Add User*

Pada gambar 4.22 merupakan tampilan *add user*, admin nanti bisa menambahkan sesuai role yang dibutuhkan oleh pengguna dan saat sudah menginputkan, *admin* hanya tekan simpan untuk menambahkan pengguna sesuai rolenya.

The screenshot shows the 'Edit User' page. On the left, there's a sidebar with 'GENERAL' (Dashboard), 'APPLICATION' (Pasien, Users), and 'AUTHENTICATION' (Keluar). At the top right, it says 'Muhammad Arif Alifan Sultan' and 'Admin'. The main area has a title 'Edit User' and fields for 'Nama' (Muhammad rafli), 'Email' (dokter@gmail.com), 'Kata Sandi Baru', 'Role' (Dokter), and a 'Pilih Role' dropdown. A 'Simpan' button is at the bottom right.

Gambar 4.23 Tampilan Edit User

Pada gambar 4.23 merupakan tampilan *add user*, admin nanti bisa menambahkan sesuai role yang dibutuhkan oleh pengguna dan saat sudah menginputkan admin hanya tekan simpan untuk menambahkan pengguna sesuai rolenya.

The screenshot shows a table titled 'Semua Pengguna' with columns: Nama, Email, Role, Active, and Aksi. There are three rows of data. In the 'Aksi' column for the first row, a modal dialog box appears asking 'Apakah mau nonaktifkan pengguna ini?' with options 'Tidak' and 'Ya'. The 'Ya' button is highlighted. Navigation buttons <, 1, and > are at the bottom.

Semua Pengguna					
Bersihkan Filter Bersihkan filter dan sortir					
Add User					
Nama	Email	Role	Active	Aksi	
Muhammad Arif Alifan ...	admin@gmail.com	Admin	Active	Edit Nonaktifkan	! Apakah mau nonaktifkan pengguna ini? <input style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 10px;" type="button" value="Tidak"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 10px;" type="button" value="Ya"/>
Muhammad rafli	dokter@gmail.com	Dokter	Active	Edit Nonaktifkan	
Viky Indika Tintatiana	apoteker@gmail.com	Apoteker	Active	Edit Nonaktifkan	

Gambar 4.24 Konfirmasi Nonaktifkan User

Pada gambar 4.24 merupakan konfirmasi jika user tersebut mau dinonaktifkan, setelah admin klik “Ya” maka akan muncul notifikasi bahwa pengguna sukses dinonaktifkan, dan saat sudah *inactive* maka *user* tersebut tidak akan bisa akses *website* klinik lagi, kecuali jika *user* tersebut diaktifkan kembali.

4.2 Pembahasan

4.2.1 Perencanaan/*Planning*

Pada tahap perencanaan iterasi pertama, dilakukan *planning meeting* dengan dokter dan apoteker untuk membahas kebutuhan

pengguna. Berdasarkan pertanyaan yang terdapat di tabel 3.3, maka diperoleh jawaban dari dokter dan apoteker sebagai berikut:

4.2.1.1 Kebutuhan Utama Sistem

Pembuatan aplikasi sistem *electronic prescribing* berbasis *website* memiliki fitur untuk menambahkan data pasien, data obat dan data resep obat. Selain itu pihak klinik juga menyarankan untuk memiliki halaman untuk bisa melihat halaman riwayat untuk apoteker bisa melihat yang nantinya terhubung terkait data pasien, dan data obat serta data resep obat.

4.2.1.2 Tampilan Riwayat Resep

Dari pihak klinik menyampaikan agar pada tampilan riwayat resep memiliki tampilan yang mudah dilihat dan bisa dipahami dengan mudah sama apoteker saat membuat resep obat nantinya, dan memiliki fitur cetak resep jika diperlukan untuk apoteker.

4.2.1.3 Fitur Pendukung Aplikasi

Selain halaman input data untuk pasien dan resep obat, dari pihak klinik juga membutuhkan tampilan *dashboard* yang menampilkan data pasien hari ini dan resep yang sudah dibuat, agar bisa melihat data pada hari itu sesuai resep yang dibutuhkan dan resep yang sudah dibuat bisa terukur oleh data.

Berdasarkan hasil tersebut, maka diperoleh *user stories* yang akan digunakan sebagai landasan dalam perancangan aplikasi pada tahap berikutnya. Pada tabel 4.1 merupakan *user stories* pada iterasi pertama

Tabel 4.1 User Stories Iterasi Pertama

No	User Story
1	Sistem yang dibuat mampu menambahkan data pasien, data obat, dan data resep obat yang nantinya bisa terlihat jelas agar dapat mudah dibaca oleh apoteker
2	Sistem yang dibuat mampu mencetak resep untuk rekap data yang nantinya untuk simpan data resep jika diperlukan oleh klinik
3	Sistem yang dibuat mampu menampilkan data pasien serta resep obat yang ditambahkan oleh dokter
4	Sistem yang dibuat membuat pihak dari klinik dapat memantau berapa pasien yang datang dan juga resep obat yang sudah selesai dibuat lewat <i>dashboard</i>

Berdasarkan *user stories* yang diperoleh, maka hasil analisis kebutuhan sistem yang dibuat adalah sebagai berikut:

- a. Fitur *login* untuk membatasi akses penggunaan aplikasi yang hanya akan digunakan oleh pihak klinik Asy-Syifa Desa Randudongkal.
- b. Fitur resep obat untuk menambah atau mencari data pasien, menambah data obat, dan menambah data resep obat.
- c. Fitur riwayat resep untuk melihat data pasien, data obat dan data resep obat yang ditambahkan oleh dokter.
- d. Fitur *dashboard* sebagai pemantau pasien yang datang dan resep obat yang sudah dibuat.
- e. Fitur *users* untuk menambah dokter dan apoteker jika terdapat dokter baru atau apoteker baru pada klinik.

Setelah seluruh tahapan dilaksanakan, penulis melakukan pengujian kepada pihak dokter dan apoteker yang ada pada klinik Asy-Syifa kemudian didapatkan terdapat fitur baru yang dibutuhkan oleh dokter dan apoteker di klinik Asy-Syifa yaitu pengelolalan stok obat dan juga validasi penambahan data resep obat saat dokter

menambahkan data resep obat. Pada tahapan ini, diperoleh user stories baru sebagai tambahan dari user stories sebelumnya. Pada iterasi kedua, diperoleh user stories seperti pada tabel 4.2.

Tabel 4.2 User Stories Iterasi Kedua

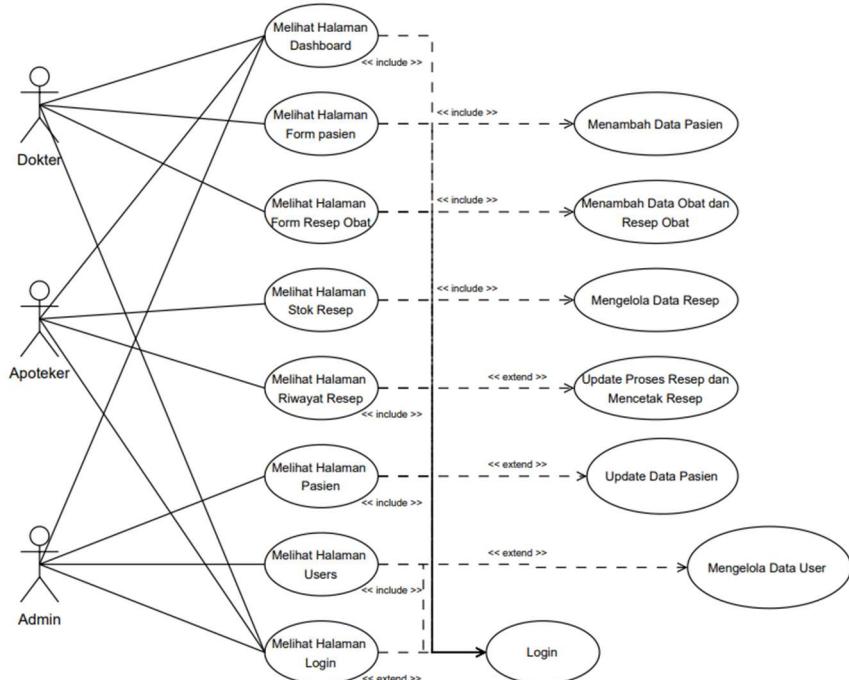
No	User Story
1	Penambahan fitur untuk melihat stok obat dan juga manajemen stok obat pada bagian apoteker
2	Saat menambahkan data pasien baru atau menambahkan data resep obat perlu ada konfirmasi bahwa data yang ditambahkan sudah benar atau belum

Sehingga, pada iterasi kedua kebutuhan sistem bertambah dari tahapan iterasi pertama dengan fitur yaitu manajemen stok obat dan juga konfirmasi data pada saat menambahkan data pasien atau data resep obat

4.2.2 Perancangan/*Design*

Langkah yang dilakukan pada tahapan ini adalah pembuatan *Unified Modeling Language* berdasarkan kebutuhan dan analisis fitur pada tahapan sebelumnya. Pada tahap perancangan, penulis menggunakan *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*. Hasil analisis dari ketiga *UML* dijadikan rancangan *user interface* yang akan digunakan pada proses pengkodean.

4.2.2.1 Use Case Diagram



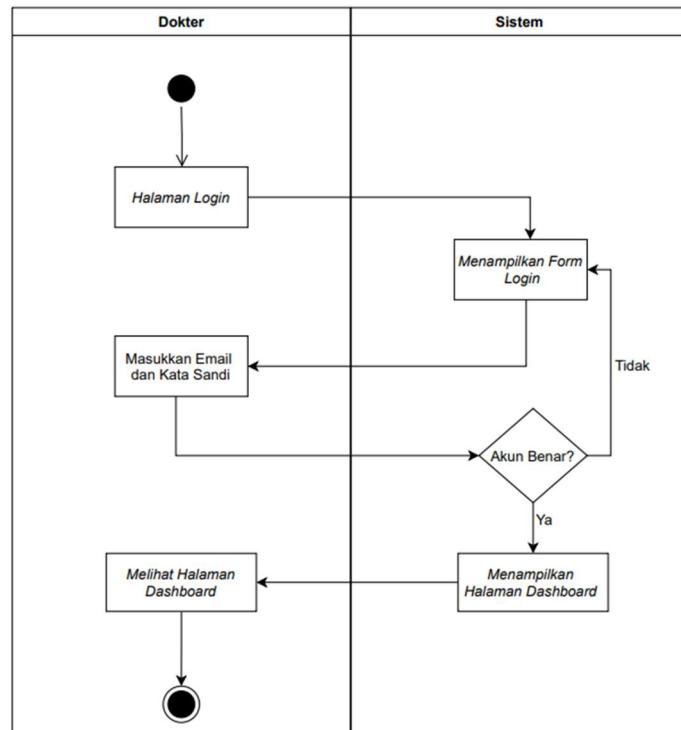
Gambar 4.25 Use Case Diagram

Pada gambar 4.25 di atas, ditampilkan diagram *use case* dari penelitian ini yang melibatkan tiga aktor: dokter, apoteker, dan *admin*. Dalam sistem ini, aktor dokter setelah *login* dapat mengakses berbagai halaman seperti *dashboard*, resep obat, obat, dan resep. Aktor apoteker, setelah *login*, dapat melihat halaman *dashboard* dan riwayat resep. Sedangkan aktor *admin*, setelah *login*, memiliki akses yaitu beberapa fitur khusus yang hanya dapat diakses oleh *admin*, seperti halaman pasien dan *users*. *Admin* juga memiliki kemampuan untuk menambahkan, mengubah, dan menonaktifkan *users*, serta memiliki akses sendiri ke halaman pasien.

4.2.2.2 Activity Diagram

Activity diagram dalam pembuatan sistem yang dilakukan dalam penelitian ini dapat dilihat pada gambar berikut:

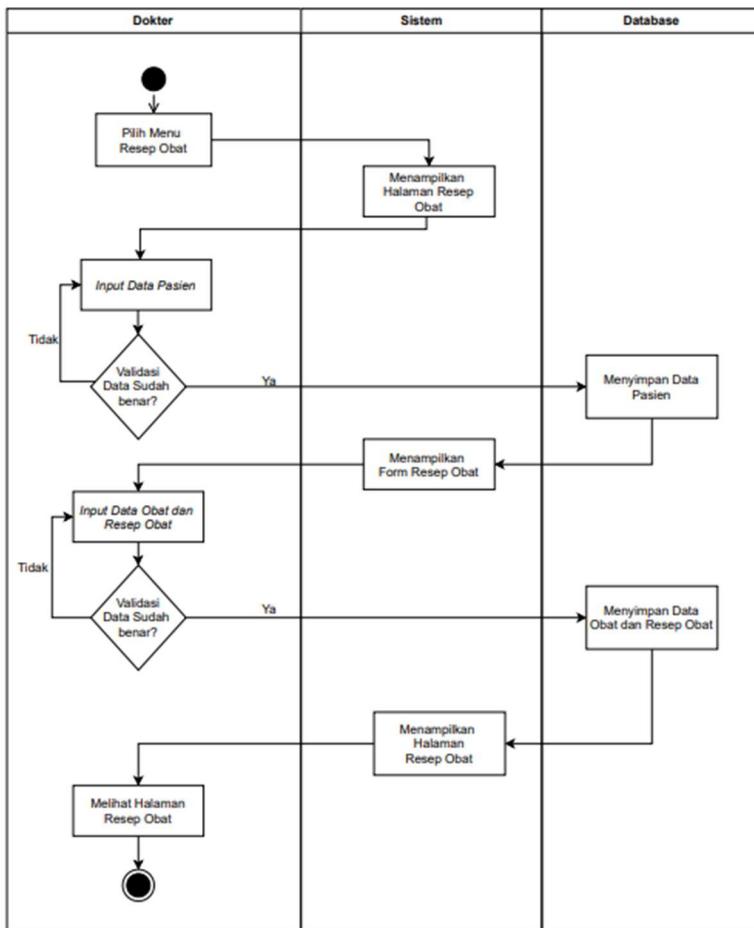
Halaman *Login* Dokter



Gambar 4.26 Activity Diagram Halaman *Login* Dokter

Pada gambar 4.26 merupakan halaman *login* dokter, dimana dokter nanti memasukkan *email* dan kata sandi yang nantinya akan dilihat dengan data yang sudah ada jika akun benar maka masuk ke halaman *dashboard*, jika salah memasukkan email dan kata sandi akan tetap di halaman *login*.

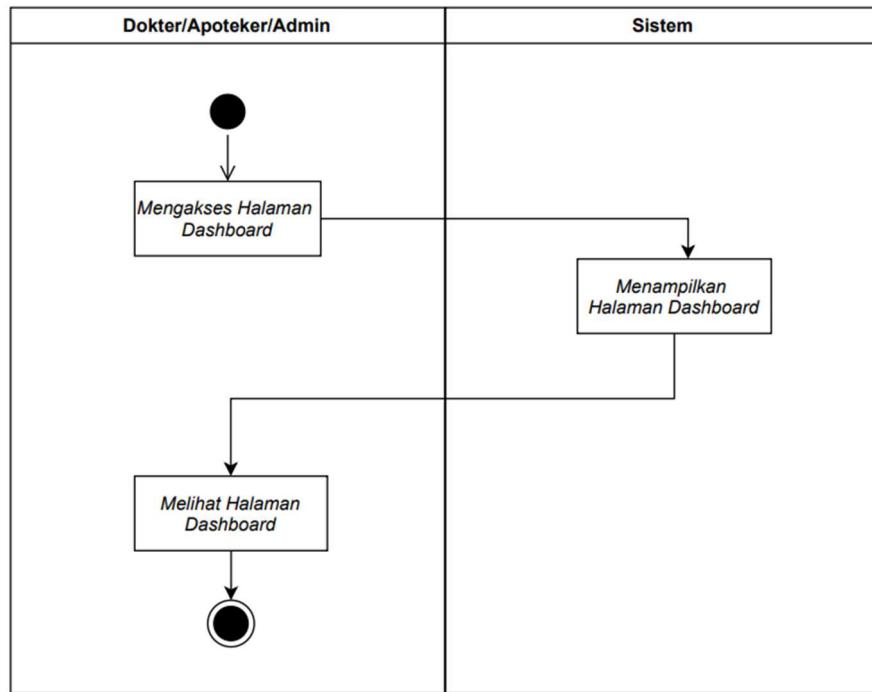
Halaman Resep Obat



Gambar 4.27 Activity Diagram Halaman Resep Obat Dokter

Pada gambar 4.27 merupakan halaman resep obat untuk dokter, tampilan awal dokter menginputkan data pasien setelah selesai maka data pasien akan tersimpan ke *database*, setelah itu dokter menginputkan data obat pada pasien tersebut setelah itu data tersimpan di *database*, dan terakhir dokter menginputkan data resep obat untuk obat yang sudah diinputkan, setelah itu data resep obat akan tersimpan dalam *database*.

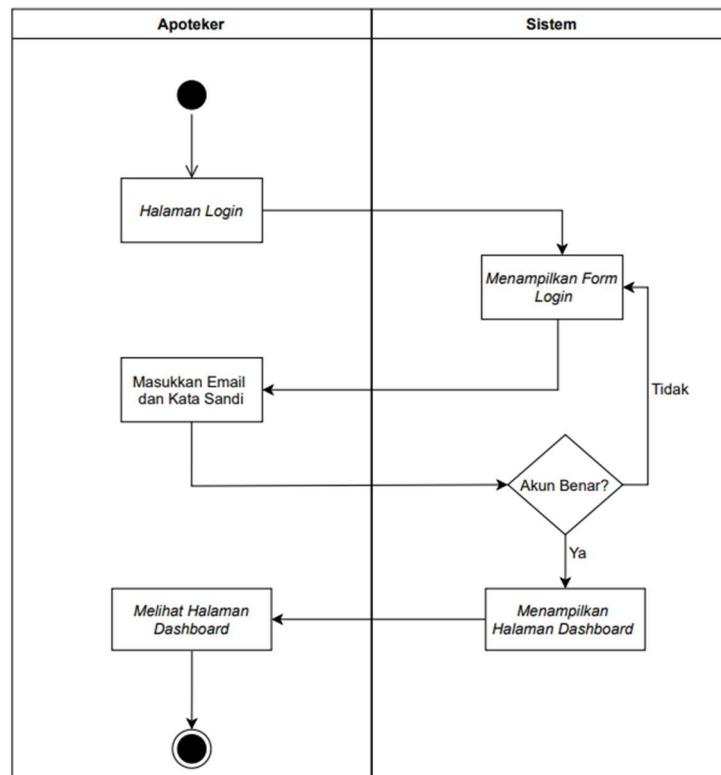
Halaman *Dashboard*



Gambar 4.28 Activity Diagram Halaman *Dashboard*

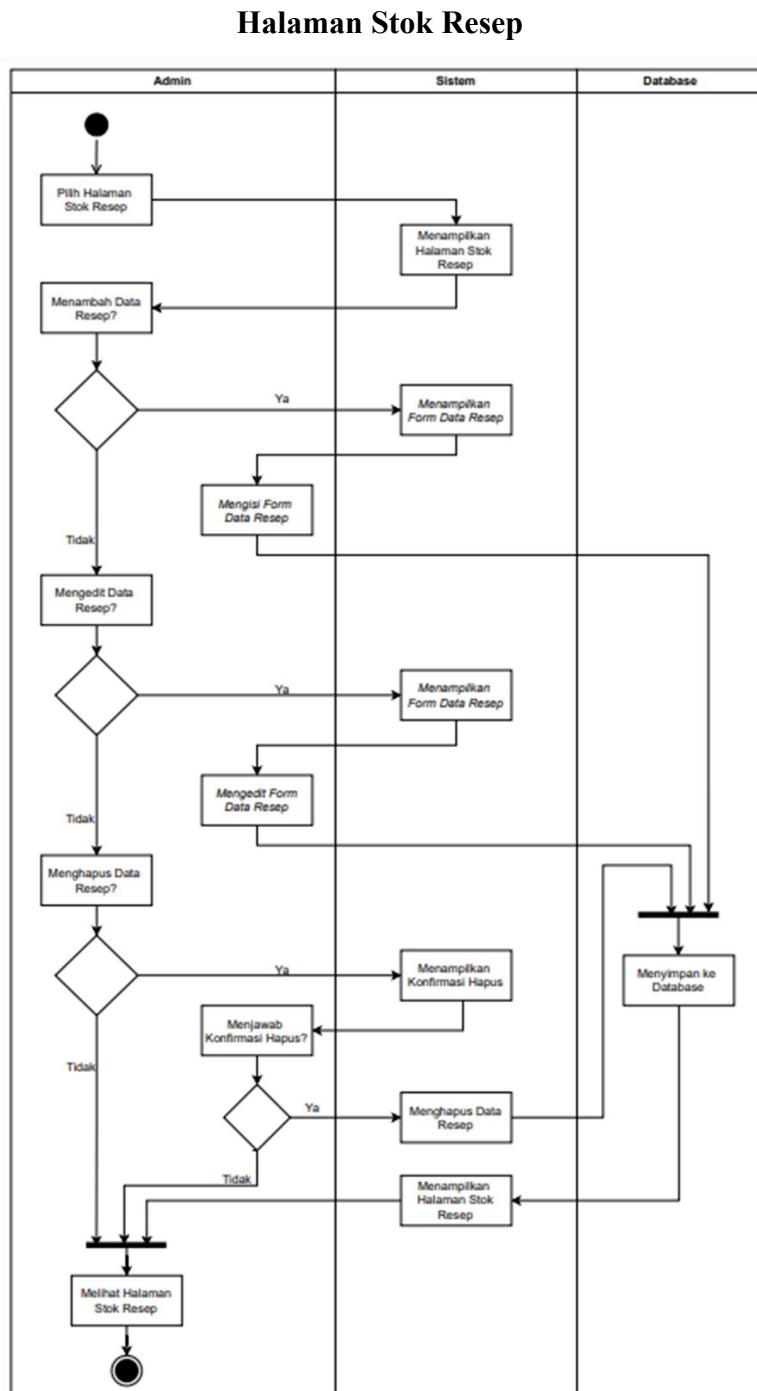
Pada gambar 4.28 menampilkan halaman *dashboard* yang digunakan oleh tiga jenis pengguna dengan peran yang berbeda dalam sistem: dokter, apoteker, dan admin. *Dashboard* ini dirancang untuk memberikan akses yang mudah ke berbagai fitur dan informasi penting yang relevan dengan masing-masing peran. *Dashboard* ini juga menyediakan informasi tentang pasien dan juga resep obat. Secara keseluruhan, halaman *dashboard* yang ditampilkan memberikan antarmuka yang disesuaikan untuk memenuhi kebutuhan operasional dan informasi dari dokter, apoteker, dan *admin* dalam satu *platform* yang terintegrasi.

Halaman Login Apoteker



Gambar 4.29 Activity Diagram Halaman Login Apoteker

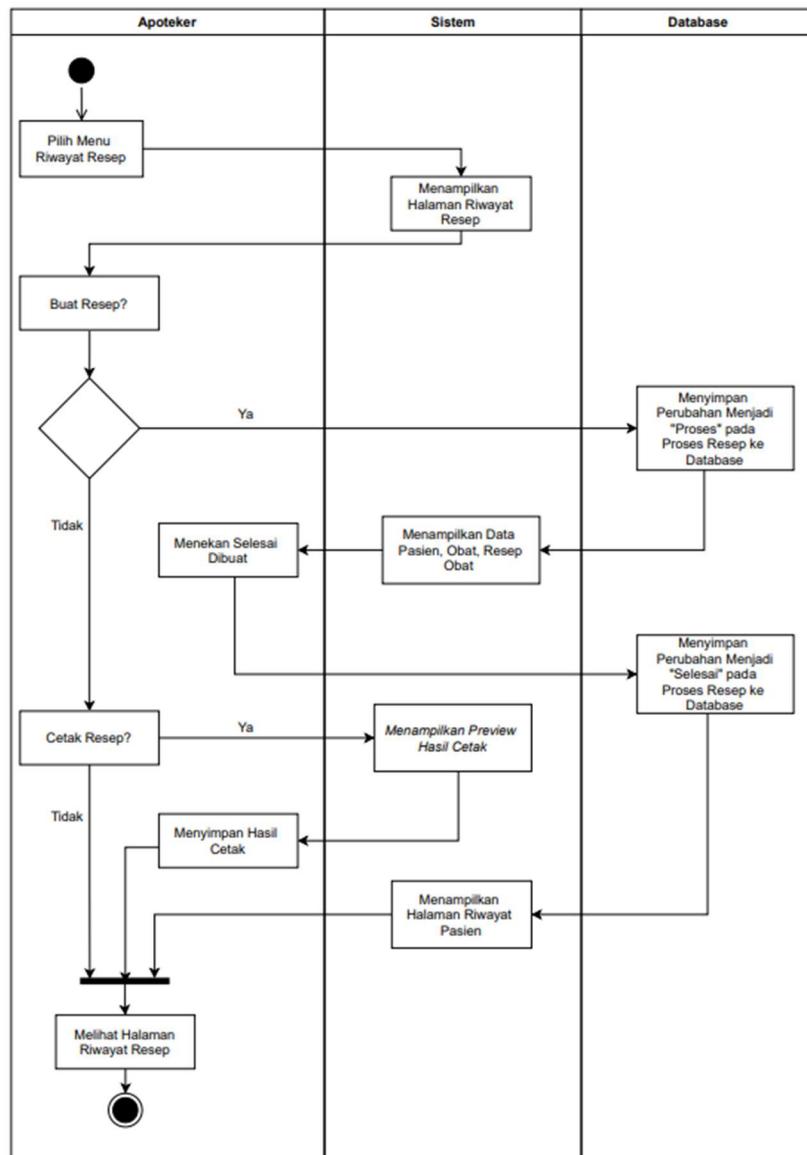
Pada gambar 4.29 menunjukkan halaman *login* apoteker, di mana mereka harus memasukkan *email* dan kata sandi untuk mengakses sistem. Setelah kredensial ini dimasukkan, sistem akan memverifikasinya dengan mencocokkan data yang ada di *database*. Jika kombinasi *email* dan kata sandi valid, apoteker akan diarahkan ke halaman *dashboard*. Namun, jika kredensial tidak sesuai, sistem akan menolak akses dan apoteker akan tetap berada di halaman login sambil menerima pesan kesalahan yang menunjukkan bahwa informasi yang dimasukkan salah atau user tidak ditemukan. Hal ini memastikan bahwa hanya pengguna yang sah dan berwenang yang dapat mengakses informasi sensitif dan operasional dalam sistem.



Gambar 4.30 Activity Diagram Halaman Stok Resep

Pada gambar 4.30 merupakan halaman stok resep dimana nantinya apoteker bisa mengelola data dari resep pada halaman tersebut, seperti menambah, mengedit dan menghapus data resep.

Halaman Riwayat Resep

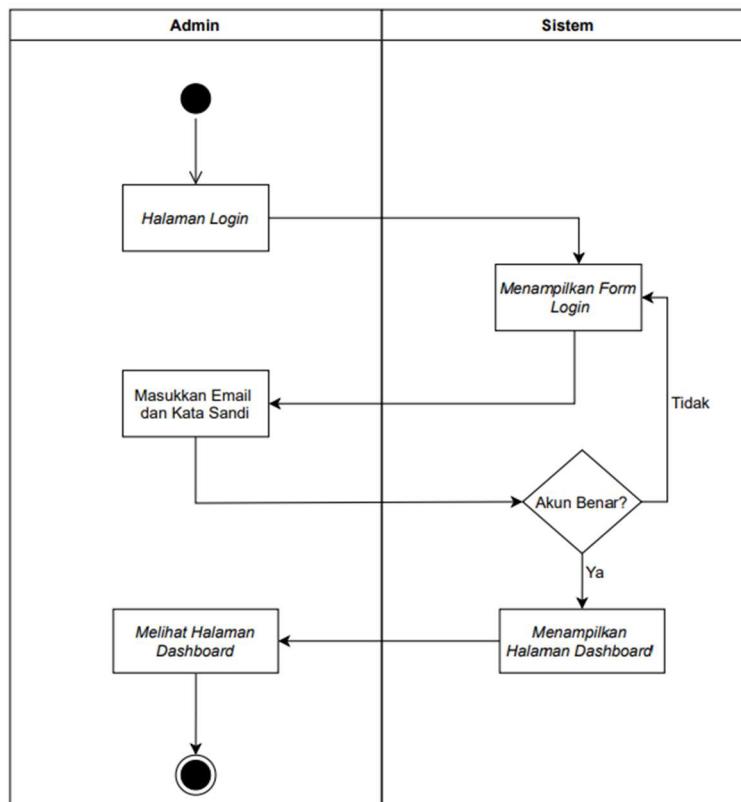


Gambar 4.31 Activity Diagram Halaman Riwayat Resep Apoteker

Pada gambar 4.31 merupakan halaman riwayat resep untuk apoteker, dimana apoteker memilih pasien terlebih dahulu untuk dibuatkan obatnya, lalu apoteker buat resep pada pasien tersebut, pada *database* pada proses resep berubah menjadi Proses lalu menampilkan data pasien, data obat dan resep obat. Setelah apoteker selesai membuat obatnya dengan melihat resep tersebut maka apoteker bisa selesai dibuat untuk selesai maka *database* akan

menyimpan data pada proses resep menjadi selesai dan obat pasien telah selesai dibuat. Dan apoteker juga bisa mencetak resep pada halaman riwayat resep dimana berisi data pasien, obat, dan resep obat pada pasien yang dipilih.

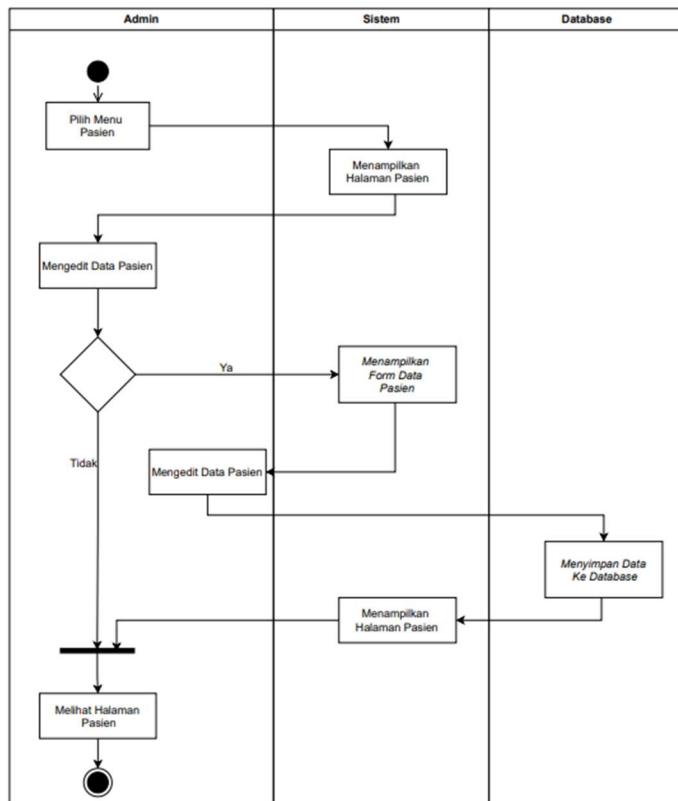
Halaman Login Admin



Gambar 4.32 Activity Diagram Halaman Login Admin

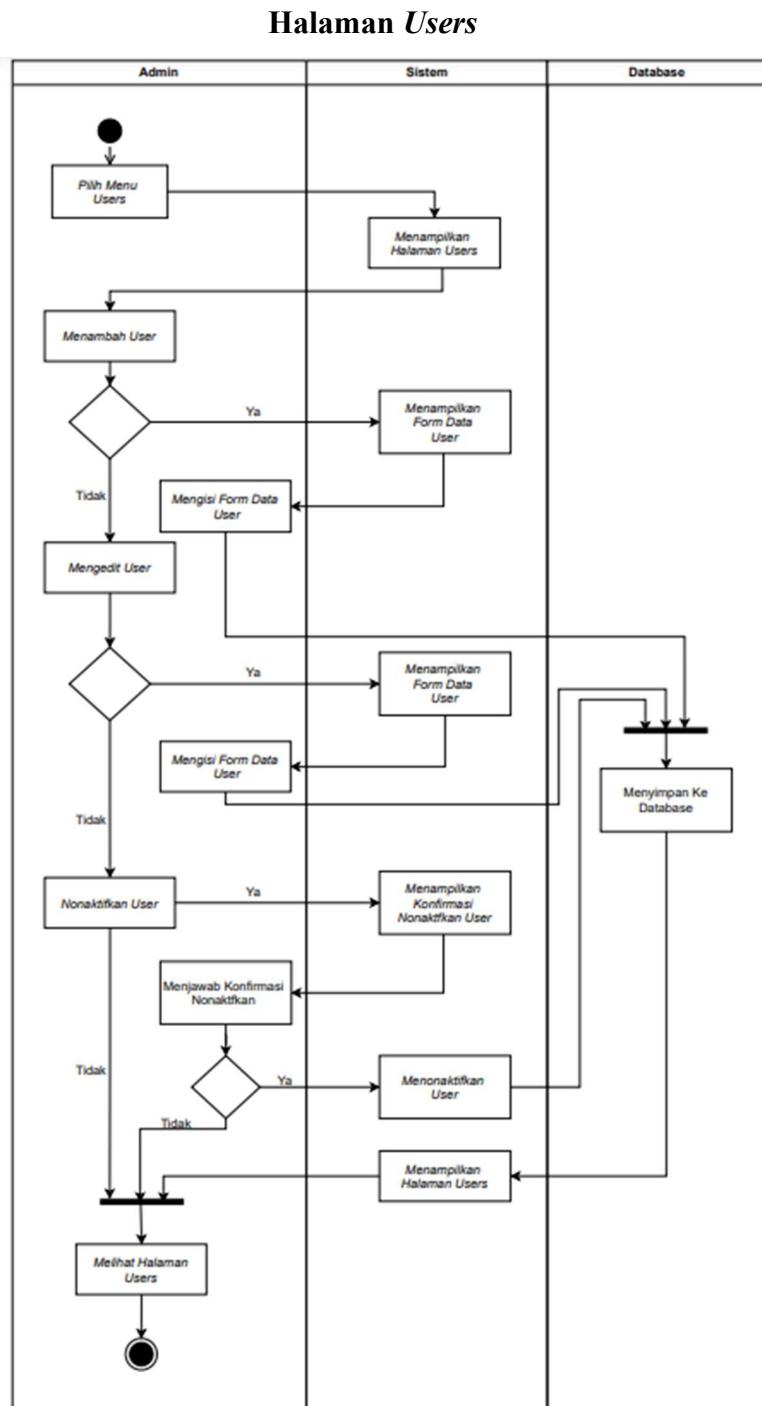
Pada gambar 4.32 menampilkan tampilan halaman *login* yang ditujukan untuk *admin* sistem. Di halaman ini, *admin* diminta untuk memasukkan kredensial *login*, yakni *email* dan kata sandi. Ketika *admin* memasukkan *email* dan kata sandi, sistem akan memeriksa data tersebut dengan membandingkannya dengan informasi akun yang telah terdaftar dalam *database*. Jika kombinasi *email* dan kata sandi yang dimasukkan sesuai dengan data yang ada, *admin* akan berhasil masuk dan dialihkan ke halaman *dashboard*. Namun, jika *email* atau kata sandi yang dimasukkan tidak sesuai dengan data yang ada, sistem akan menolak *login* tersebut.

Halaman Pasien



Gambar 4.33 Activity Diagram Halaman Pasien Admin

Pada gambar 4.33 merupakan halaman pasien *admin*, dimana *admin* nanti memilih untuk mengedit atau tidak jika *admin* memilih tidak maka akan melihat halaman pasien, jika *admin* memilih untuk mengedit pasien maka akan menampilkan *form* data pasien, setelah *admin* mengedit data pasien maka akan menyimpan data ke *database* setelah itu maka akan menampilkan halaman pasien.

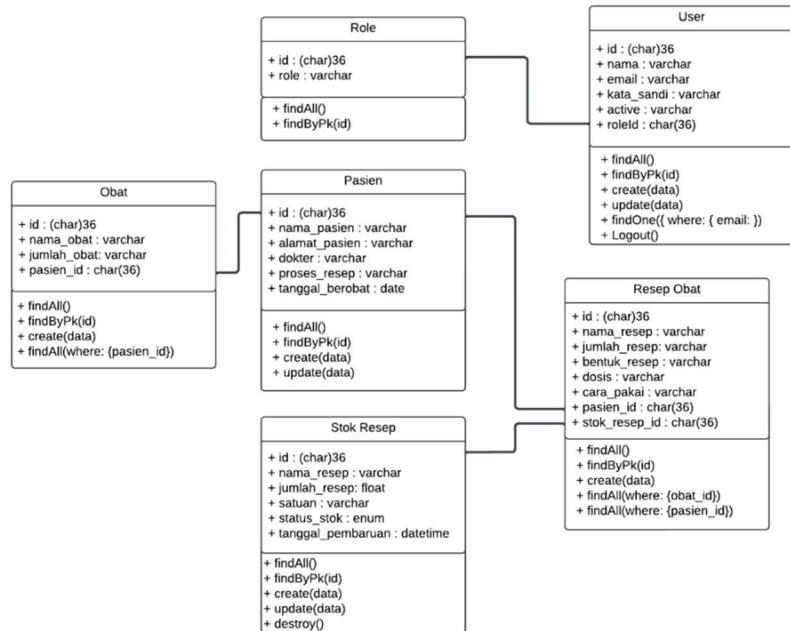


Gambar 4.34 Activity Diagram Halaman Users Admin

Pada gambar 4.34 merupakan halaman *user admin*, dimana *admin* nanti memilih untuk menambahkan *user* atau tidak jika *admin* memilih tidak maka akan melihat halaman *user*, jika *admin* memilih untuk menambahkan *user* maka akan menampilkan *form data user*,

setelah itu admin bisa mengisi *form* data *user*, setelah itu data tersimpan di *database*, dan admin bisa melakukan itu juga pada edit *user*, dan nonaktifkan *user*.

4.2.2.3 Class Diagram



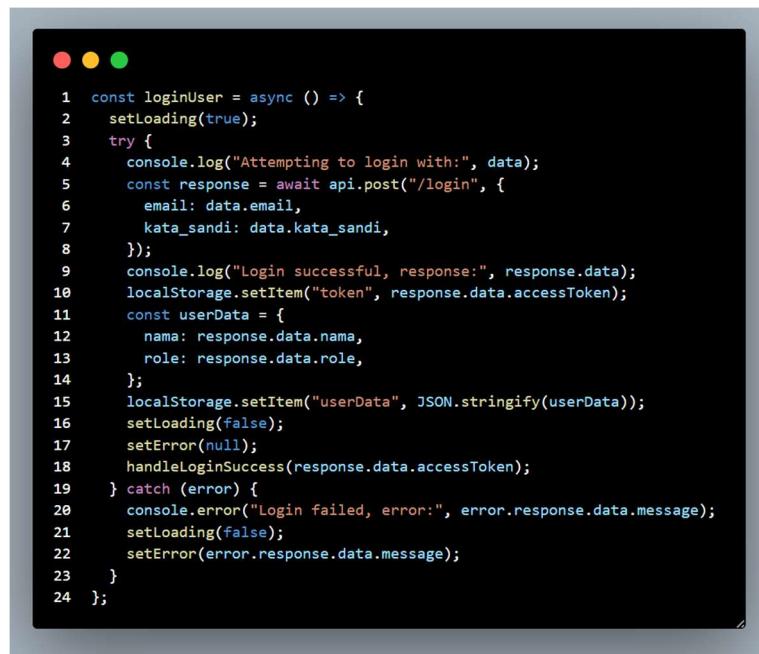
Gambar 4.35 Class Diagram

Pada Gambar 4.35 di atas, ditampilkan *class diagram* yang merupakan hasil dari penelitian yang dilakukan oleh peneliti dalam tugas akhir ini. *Class diagram* tersebut digunakan untuk menggambarkan hubungan antara objek-objek yang ada pada sistem. Dalam konteks pengembangan sistem *electronic prescribing* di Klinik Asy-Syifa, *class diagram* ini menjadi salah satu komponen penting yang membantu memahami struktur dan interaksi antar objek dalam sistem. Diagram ini mencakup berbagai kelas dan hubungan di antara mereka, termasuk atribut dan metode yang dimiliki setiap kelas.

4.2.3 Pengkodean/ *Coding*

Selama tahap pengkodean, penulis mengembangkan kode program berdasarkan rancangan *UML* seperti *use case* diagram, *activity* diagram, dan *class* diagram yang telah dibuat sebelumnya. Penulis menggunakan kerangka kerja *ReactJS* untuk *frontend* dan *ExpressJS* untuk bagian *backend* dalam memenuhi kebutuhan sistem. Sepanjang proses pengkodean, penulis berkolaborasi dengan pihak yang mengelola *API* dan *database*. Hasil kode program mencakup fungsionalitas pada setiap halaman dan fitur-fitur yang telah dikembangkan sebagai berikut:

4.2.3.1 Halaman Login



```

1  const loginUser = async () => {
2    setLoading(true);
3    try {
4      console.log("Attempting to login with:", data);
5      const response = await api.post("/login", {
6        email: data.email,
7        kata_sandi: data.kata_sandi,
8      });
9      console.log("Login successful, response:", response.data);
10     localStorage.setItem("token", response.data.accessToken);
11     const userData = {
12       nama: response.data.nama,
13       role: response.data.role,
14     };
15     localStorage.setItem("userData", JSON.stringify(userData));
16     setLoading(false);
17     setError(null);
18     handleLoginSuccess(response.data.accessToken);
19   } catch (error) {
20     console.error("Login failed, error:", error.response.data.message);
21     setLoading(false);
22     setError(error.response.data.message);
23   }
24 };

```

Gambar 4.36 Kode Fungsi Pada Halaman *Login*

Pada gambar 4.36 merupakan halaman *login*, halaman tersebut dilakukan pengiriman dengan alamat *API* yang tersedia, jadi dokter, apoteker dan admin akan login menggunakan email dan kata sandi agar bisa di proses datanya melalui *API*.

4.2.3.2 Halaman Dashboard



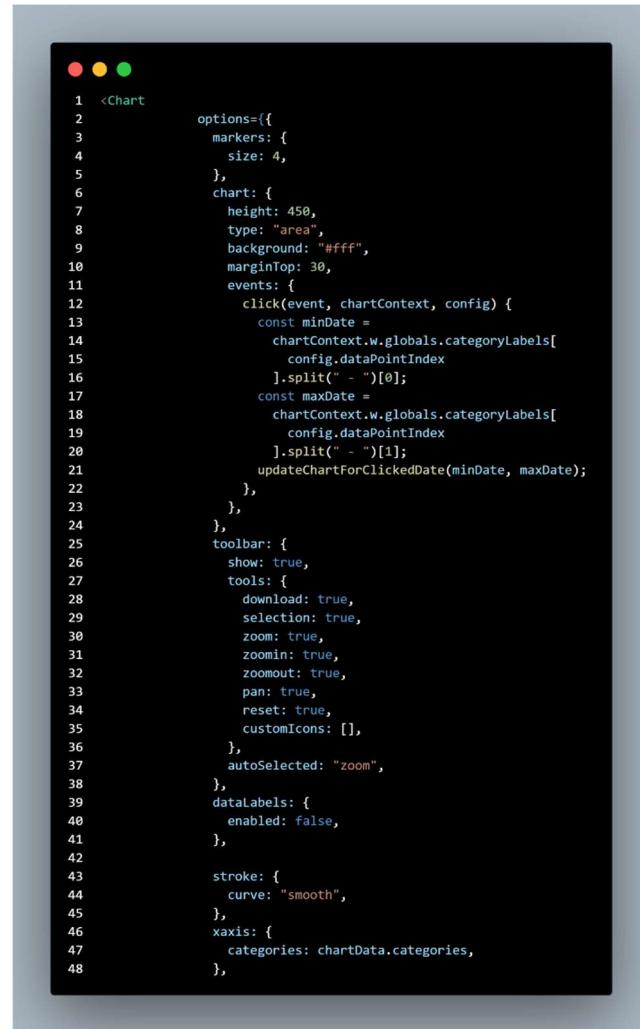
```

1  useEffect(() => {
2    const fetchPatientsToday = async () => {
3      try {
4        const response = await api.get("/pasien/today/date");
5        const pasienList = response.data;
6        setPatientsToday(pasienList.length);
7      } catch (error) {
8        console.error("Error fetching patients today:", error);
9      }
10    };
11
12    const fetchTotalResepHariIni = async () => {
13      try {
14        const response = await api.get("/resep_obat/today/date");
15        const totalResep = response.data.totalResepObatHariIni;
16        setTotalResepHariIni(totalResep);
17      } catch (error) {
18        console.error("Error fetching total resep hari ini:", error);
19      }
20    };
21
22    const fetchPatientsSelesaiHariIni = async () => {
23      try {
24        const response = await api.get("/pasien/today/obat/selesai");
25        const pasienList = response.data;
26        setPatientsSelesaiHariIni(pasienList);
27      } catch (error) {
28        console.error(
29          "Error fetching patients with finished prescriptions today:",
30          error
31        );
32      }
33    };
34    fetchPatientsToday();
35    fetchTotalResepHariIni();
36    fetchPatientsSelesaiHariIni();
37  }, []);
38
39  const updateChartForClickedDate = async (minDate, maxDate) => {
40    console.log(minDate);
41    console.log(maxDate);
42    try {
43      // Kirim permintaan ke server untuk mendapatkan data untuk tanggal yang diklik
44      const response = await api.get("/pasien/perhari-by-week/hari", {
45        params: {
46          startOfWeek: minDate,
47          endOfWeek: maxDate,
48        },
49      });

```

Gambar 4.37 Kode Fungsi Program Pada Halaman *Dashboard*

Pada gambar 4.37 merupakan halaman *dashboard* yang berisi data pasien hari ini, resep obat hari ini, dan resep yang sudah selesai hari ini, data tersebut didapatkan lewat *API* yang sudah disediakan, jadi pengguna dapat melihat informasi yang sudah didapatkan dari *API* yang ada, *API* tersebut mengambil dari *database* setelah dokter menginputkan data pasien sampai resep obat sampai data yang diinputkan tersimpan, setelah itu terbaca lagi oleh *API* untuk menampilkan data pasien dan resep obat tersebut.



```

1  <Chart
2    options={{
3      markers: {
4        size: 4,
5      },
6      chart: {
7        height: 450,
8        type: "area",
9        background: "#fff",
10       marginTop: 30,
11       events: {
12         click(event, chartContext, config) {
13           const minDate =
14             chartContext.w.globals.categoryLabels[
15               config.dataPointIndex
16             ].split(" - ")[0];
17           const maxDate =
18             chartContext.w.globals.categoryLabels[
19               config.dataPointIndex
20             ].split(" - ")[1];
21           updateChartForClickedDate(minDate, maxDate);
22         },
23       },
24     },
25     toolbar: {
26       show: true,
27       tools: {
28         download: true,
29         selection: true,
30         zoom: true,
31         zoomin: true,
32         zoomout: true,
33         pan: true,
34         reset: true,
35         customIcons: [],
36       },
37       autoSelected: "zoom",
38     },
39     dataLabels: {
40       enabled: false,
41     },
42     stroke: {
43       curve: "smooth",
44     },
45   },
46   xaxis: {
47     categories: chartData.categories,
48   },

```

Gambar 4.38 Kode Fungsi *Chart* Pada Halaman *Dashboard*

Pada gambar 4.38 merupakan data *chart* yang ada pada *dashboard*, kode diatas menggunakan *chart* dari *library apex chart*, untuk mendapatkan data pasien yaitu tujuh minggu sebelumnya dengan *API* yang sudah ada, lalu dari data pasien yang sebelumnya perminggu akan bisa memecah lagi membaca datanya jadi data perharian di satu minggu yang dipilih yang didapatkan dari *API* yang sudah disediakan.

4.2.3.3 Halaman Resep Obat

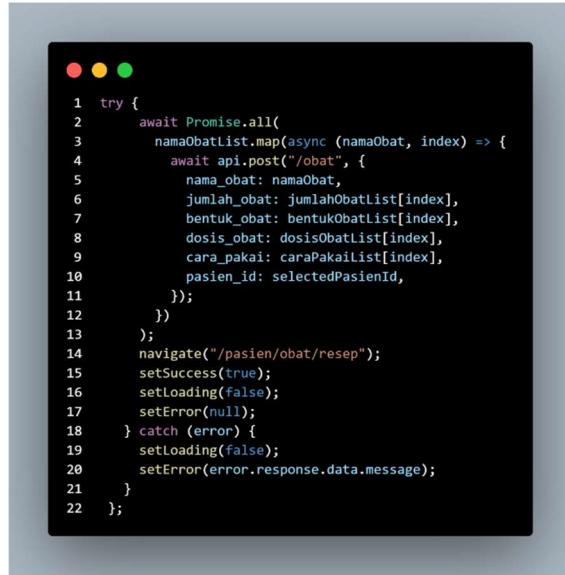
```

1  const handleSubmit = async (e) => {
2    e.preventDefault();
3    if (selectedOption === "existing" && existingPatientSelected) {
4      navigate("/pasien/obat/");
5    } else if (selectedOption === "new") {
6      if (isValid()) {
7        setErrorMessage("");
8        setLoading(true);
9        try {
10          await api.post("/pasien", formData);
11          setSuccess(true);
12          setLoading(false);
13          setErrorMessage("");
14          setFormData({
15            nama_pasien: "",
16            alamat_pasien: "",
17            dokter: "",
18            tanggal_berobat: "",
19          });
20          navigate("/pasien/obat");
21        } catch (error) {
22          setLoading(false);
23          setSuccess(false);
24          setErrorMessage("Gagal menyimpan data. Silakan coba lagi.");
25          console.error("Error saving data: ", error);
26        }
27      } else {
28        setErrorMessage("Silakan isi semua bidang sebelum melanjutkan.");
29      }
30    };
31  };
32
33  const isValid = () => {
34    return Object.values(formFieldsValid).every((value) => value);
35  };

```

Gambar 4.39 Kode Fungsi Pada Halaman Resep Obat

Pada gambar 4.39 merupakan kode program untuk mengirim data saat dokter sudah menginputkan data pasien, setelah data pasien sudah benar maka akan lanjut ke halaman berikutnya serta mengirim data melalui *API* yang tersedia dan data pasien tersebut akan tersimpan dalam *database*. Setelah data tersimpan, maka baru lanjut ke tampilan berikutnya.



```

1 try {
2     await Promise.all(
3         namaObatList.map(async (namaObat, index) => {
4             await api.post("/obat", {
5                 nama_obat: namaObat,
6                 jumlah_obat: jumlahObatList[index],
7                 bentuk_obat: bentukObatList[index],
8                 dosis_obat: dosisObatList[index],
9                 cara_pakai: caraPakaiList[index],
10                pasien_id: selectedPasienId,
11            });
12        })
13    );
14    navigate("/pasien/obat/resep");
15    setSuccess(true);
16    setLoading(false);
17    setError(null);
18 } catch (error) {
19     setLoading(false);
20     setError(error.response.data.message);
21 }
22 };

```

Gambar 4.40 Kode Fungsi Program Tampilan Obat

Pada gambar 4.40 merupakan kode program dari tampilan obat disaat dokter sudah menginputkan data obat maka program tersebut bekerja melalui *API* yang sudah tersedia.



```

1 try {
2     await Promise.all(
3         data.map(async (resep, index) => {
4             await api.post("/resep_obat", {
5                 nama_resep: resep.nama_resep,
6                 jumlah_resep: resep.jumlah_resep,
7                 bentuk_resep: resep.bentuk_resep,
8                 obat_id: selectedObatIds[index],
9             });
10        })
11    );
12    navigate("/pasien");
13    setSuccess(true);
14    window.location.reload();
15    setLoading(false);
16    setError(null);
17 } catch (error) {
18     setLoading(false);
19     setError(error.response.data.error);
20 }
21 };

```

Gambar 4.41 Kode Fungsi Program Tampilan Resep Obat

Pada gambar 4.41 merupakan kode program dari tampilan Resep Obat disaat dokter sudah menginputkan data resep obat maka program tersebut bekerja melalui *API* yang sudah tersedia.

4.2.3.4 Halaman Riwayat Resep



```

1 const response = await api.get(`/pasien/${record.id}`);
2 const pasienData = response.data;
3
4 const obatResponse = await api.get(`/obat/${record.id}`);
5 const obatData = obatResponse.data;
6
7 const obatIds = obatData.map((obat) => obat.id);
8
9 const resepObatPromises = obatIds.map((obatId) =>
10   api.get(`/resep_obat/obat/${obatId}`)
11 );
12 const resepObatResponses = await Promise.all(resepObatPromises);
13 const allResepObatData = resepObatResponses.map(
14   (response) => response.data
15 );
16
17 const mergedResepObatData = allResepObatData.flat();
18 setSelectedPasienData(pasienData);
19 setSelectedObatData(obatData);
20 setSelectedResepObatData(mergedResepObatData);
21
22 const selectedResepObatGroupedByObatId = mergedResepObatData.reduce(
23   (acc, resep_obat) => {
24     if (!acc[resep_obat.id_obat]) {
25       acc[resep_obat.id_obat] = [];
26     }
27     acc[resep_obat.id_obat].push(resep_obat);
28     return acc;
29   },
30   {}
31 );
32 setSelectedResepObatByObatId(selectedResepObatGroupedByObatId);
33
34 await api.put(`/pasien/${record.id}`, { proses_resep: "Proses" });

```

Gambar 4.42 Kode Program Halaman Riwayat Resep

Pada gambar 4.32 merupakan kode program untuk melihat data pasien, data obat dan data resep obat, dimana data tersebut didapatkan dari *API* yang tersedia, yang pertama ada data pasien dari semua data pasien mendapatkan dari *API* yang ada, lalu ada data obat yang didapatkan dari *API* yang ada. Dan terakhir yaitu data resep obat didapatkan oleh *API* yang mendapatkan semua data tersebut dari *database*.



```
● ● ●
1 const handleCloseModal = async () => {
2   try {
3     if (!resepProcessed) {
4       await api.put('/pasien/${selectedPasienId}', {
5         proses_resep: "Proses",
6       });
7
8     setResepProcessed(true);
9
10    const updatedPasienList = pasienList.map((pasien) =>
11      pasien.id === selectedPasienId
12        ? { ...pasien, proses_resep: "Proses" }
13        : pasien
14    );
15    setPasienList(updatedPasienList);
16  }
17
18  setModalVisible(false);
19  setSelectedPasienId(null);
20  setSelectedPasienData(null);
21  setSelectedResepObatData(null);
22  message.success("Status resep berhasil diperbarui");
23  window.location.reload();
24 } catch (error) {
25   console.error("Error handling modal close:", error);
26   message.error("Gagal menutup modal");
27 }
28 };
29 const handleProcessCompleted = () => {
30   console.log("Proses telah selesai");
31 };
```

Gambar 4.43 Kode Fungsi Program Proses Resep

Pada gambar 4.43 memperlihatkan kode program yang bertanggung jawab untuk proses pembaruan data saat apoteker membuat obat dalam sistem. Pada awalnya, data yang sebelumnya belum diproses akan diperbarui melalui *API* yang tersedia yang terhubung dengan *backend* saat apoteker memulai proses pembuatan obat pada proses resep akan berubah menjadi “Proses”. Setelah selesai membuat obat, apoteker mengirimkan status proses resep menjadi “selesai” melalui *API* yang sama. Kode ini memungkinkan sistem untuk secara otomatis memperbarui dan melacak status proses resep, memastikan data sepanjang proses pembuatan obat.

4.2.3.5 Halaman Pasien



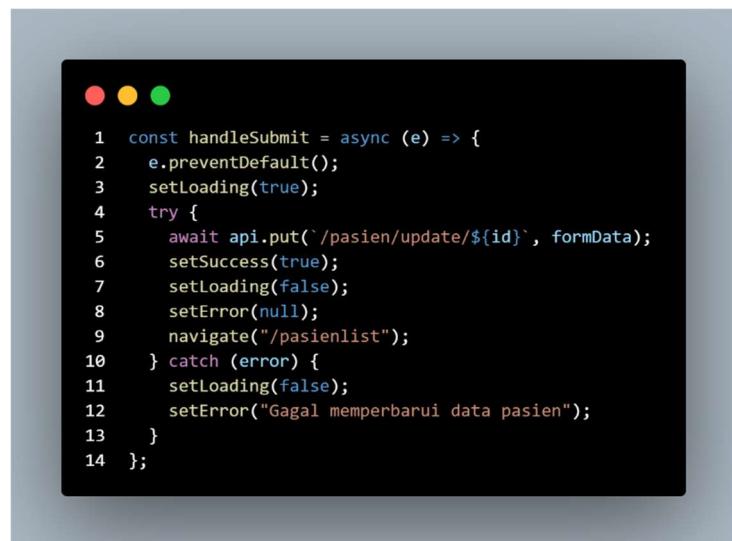
```

1 const fetchPatients = async () => {
2   setLoading(true);
3   try {
4     const response = await api.get("/pasien");
5     const sortedPatients = response.data.sort(
6       (a, b) => new Date(b.tanggal_berobat) - new Date(a.tanggal_berobat)
7     );
8     setPatients(sortedPatients);
9     setInitialPatients(sortedPatients);
10    setLoading(false);
11  } catch (error) {
12    console.error("Error fetching data:", error);
13    setLoading(false);
14  }
15 };
16 fetchPatients();
17 }, []);

```

Gambar 4.44 Kode Fungsi Program Halaman Pasien

Pada gambar 4.44 merupakan kode program pada halaman pasien, dimana dalam kode tersebut data semua pasien didapatkan oleh *API* yang ada pada program tersebut, setelah *API* mendapatkan data dari *database* maka sistem menampilkan semua data dari semua pasien.



```

1 const handleSubmit = async (e) => {
2   e.preventDefault();
3   setLoading(true);
4   try {
5     await api.put(` /pasien/update/${id}`, formData);
6     setSuccess(true);
7     setLoading(false);
8     setError(null);
9     navigate("/pasienlist");
10  } catch (error) {
11    setLoading(false);
12    setError("Gagal memperbarui data pasien");
13  }
14 };

```

Gambar 4.45 Kode Fungsi Program Tampilan Edit

Pada gambar 4.45 merupakan kode program untuk *update* data pasien, setelah menginputkan semua data maka program berjalan dan mengirim data sesuai *API* yang tersedia.

4.2.3.6 Halaman *Users*



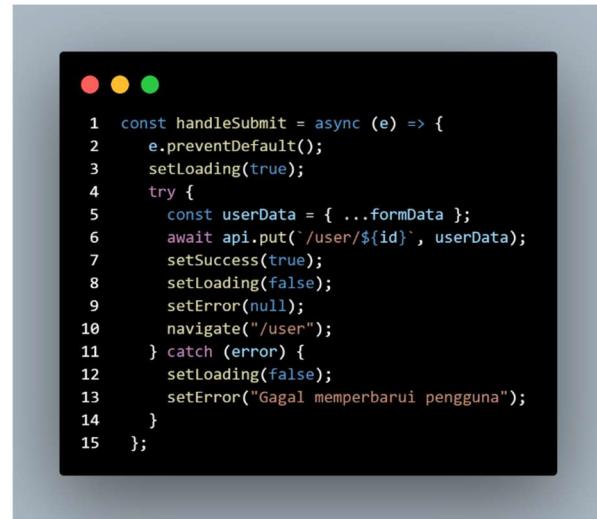
```

1  useEffect(() => {
2    const fetchData = async () => {
3      setLoading(true);
4      try {
5        const [userData, roleData] = await Promise.all([
6          api.get('/user'),
7          api.get('/role'),
8        ]);
9
10       const updatedUsers = userData.data.map(user => {
11         const role = roleData.data.find(role => role.id === user.role.id);
12         return {
13           ...user,
14           role: role ? role.role : "Unknown",
15           active: user.active,
16         };
17       });
18
19       setUsers(updatedUsers);
20       setRoles(roleData.data);
21       setLoading(false);
22     } catch (error) {
23       console.error("Error fetching data:", error);
24       setLoading(false);
25     }
26   };
27   fetchData();
28 }, []);
29
30 const handleChange = (pagination, filters, sorter) => {
31   console.log("Various parameters", pagination, filters, sorter);
32   setFilteredInfo(filters);
33   setSortedInfo(sorter);
34 };
35
36 const clearFilters = () => {
37   setFilteredInfo({});
38 };
39
40 const clearAll = () => {
41   setFilteredInfo({});
42   setSortedInfo({});
43 };
44
45 const handleEdit = (record) => {
46   console.log("Edit user:", record);
47   setEditClicked(true);
48 };
49
50 const handleDisable = async (record) => {
51   console.log("Disable user:", record);
52   try {
53     await api.put(`/user/${record.id}/disable`);
54     const updatedUsers = users.map(user =>
55       user.id === record.id ? { ...user, active: "inactive" } : user
56     );
57     setUsers(updatedUsers);
58     message.success("Sukses dinonaktifkan pengguna");
59   } catch (error) {
60     console.error("Error disabling user:", error);
61     message.error("Gagal memnonaktifkan pengguna");
62   }
63 };
64
65 const handleEnable = async (record) => {
66   console.log("Enable user:", record);
67   try {
68     await api.put(`/user/${record.id}/enable`);
69     const updatedUsers = users.map(user =>
70       user.id === record.id ? { ...user, active: "active" } : user
71     );
72     setUsers(updatedUsers);
73     message.success("Sukses mengaktifkan kembali pengguna");
74   } catch (error) {
75     console.error("Error enabling user:", error);
76     message.error("Gagal mengaktifkan kembali pengguna");
77   }
78 };

```

Gambar 4.46 Kode Program Halaman *Users*

Pada gambar 4.46 merupakan kode program bagian halaman *users*, dimana untuk mendapatkan data *user* dan juga role yang ada di *database* melalui *API* yang ada, selain itu juga mendapatkan bahwa *user* tersebut masih *active* atau *inactive* dengan *API*.



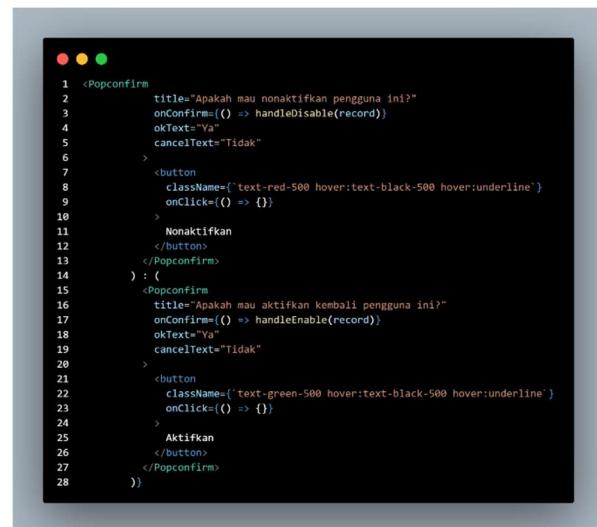
```

1 const handleSubmit = async (e) => {
2   e.preventDefault();
3   setLoading(true);
4   try {
5     const userData = { ...formData };
6     await api.put(`/user/${id}`, userData);
7     setSuccess(true);
8     setLoading(false);
9     setError(null);
10    navigate("/user");
11  } catch (error) {
12    setLoading(false);
13    setError("Gagal memperbarui pengguna");
14  }
15};

```

Gambar 4.47 Kode Fungsi Program Edit *User*

Pada gambar 4.47 merupakan kode program untuk *update* data pasien dimana pada kode tersebut setelah *admin* edit *user* maka data tersebut mengirimkan *update* baru ke *database* melalui *API* yang tersedia.



```

1 <Popconfirm
2   title="Apakah mau nonaktifkan pengguna ini?"
3   onConfirm={() => handleDisable(record)}
4   okText="Ya"
5   cancelText="Tidak"
6   >
7   <button
8     className="text-red-500 hover:text-black-500 hover:underline"
9     onClick={() => {}}
10    >
11      Nonaktifkan
12    </button>
13  </Popconfirm>
14  : (
15    <Popconfirm
16      title="Apakah mau aktifkan kembali pengguna ini?"
17      onConfirm={() => handleEnable(record)}
18      okText="Ya"
19      cancelText="Tidak"
20      >
21      <button
22        className="text-green-500 hover:text-black-500 hover:underline"
23        onClick={() => {}}
24        >
25          Aktifkan
26        </button>
27      </Popconfirm>
28    )}}

```

Gambar 4.48 Kode Fungsi Program Konfirmasi Status *User*

Pada gambar 4.48 merupakan kode program untuk konfirmasi saat *user* tersebut apakah menonaktifkan *user* atau aktifkan *user*, setelah itu *update* tersebut dikirimkan melalui *API* yang tersedia dan akan ada *update* pada *database*.

4.2.4 Pengujian/ *Testing*

Pengujian *website* sdilakukan dengan metode *black box testing* yang dilakukan secara langsung oleh dokter dan apoteker untuk mengevaluasi apakah fitur-fitur yang telah dibuat berfungsi dengan baik atau tidak. Proses pengujian dibagi menjadi 3 sistem, yaitu pengujian dengan peran sebagai dokter, peran sebagai apoteker, dan peran sebagai admin yang tertera pada tabel berikut:

4.2.4.1 *BlackBox Testing* Dengan Role Dokter

Pada pengujian *blackbox* testing pada role dokter berjalan sesuai dengan yang diharapkan. Adapun hasil pengujian yang telah dilakukan dapat dilihat pada tabel-tabel yang sudah diujikan.

a. Dokter Belum *Login*

Pada pengujian dokter saat belum *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.3.

Tabel 4.3 *Blackbox Testing* Dokter Sebelum *login*.

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
DR01	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i>	Berhasil
DR02	Mengakses Halaman <i>Dashboard</i>	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil
DR03	Mengakses Halaman Resep Obat	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil

b. Dokter Pada Halaman *Login*

Pada pengujian dokter pada halaman *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.4.

Tabel 4.4 Blackbox Testing Dokter Halaman *Login*

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
DR04	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i> dengan <i>form email</i> dan kata sandi	Berhasil
DR05	Mengisi email <u>dokter@gmail.com</u> dengan kata sandi “dokter” dan menekan tombol <i>login</i>	Sistem akan memproses masukan dari <i>user</i> dan mengarahkan ke halaman <i>dashboard</i>	Berhasil
DR06	Mengisi email dan kata sandi yang tidak terdaftar	Sistem akan menampilkan pesan <i>error</i> “User tidak ditemukan”	Berhasil
DR07	Mengisi email yang terdaftar dan mengisi kata sandi yang salah	Sistem akan menampilkan pesan <i>error</i> “Kata Sandi Salah”	Berhasil
DR08	Hanya mengisikan email dan mengosongkan kata sandi	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini”	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		dibawah input kata sandi	
DR09	Hanya Mengisikan kata sandi dan mengosongkan email	Sistem akan menampilkan pesan error "Harap isi bidang ini" dibawah input email	Berhasil

c. Dokter Pada Halaman *Dashboard*

Pada pengujian dokter pada halaman *dashboard* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.5.

Tabel 4.5 Blackbox Testing Dokter Halaman Dashboard

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
DR10	Mengakses Halaman <i>Dashboard</i>	Sistem akan menampilkan halaman <i>dashboard</i> yang menunjukkan deskripsi klinik, jumlah pasien hari ini, jumlah resep obat hari ini, resep obat yang sudah dibuat, dan grafik <i>patient chart</i>	Berhasil
DR11	Meletakkan cursor pada titik <i>chart</i> atau grafik pada tanggal	Sistem akan menampilkan tanggal dan bulan	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
	satu minggu yang dipilih	yang di pilih dan menampilkan semua pasien dalam satu minggu	
DR12	Menekan <i>icon</i> untuk <i>zoom in</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran yang lebih tinggi dari tampilan normal	Berhasil
DR13	Menekan <i>icon</i> untuk <i>zoom out</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran yang lebih rendah dan kembali normal setelah di <i>zoom</i>	Berhasil
DR14	Menekan titik pada grafik pasien satu minggu pada data <i>chart patient</i>	Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih	Berhasil

d. Dokter Pada Halaman Resep Obat

Pada pengujian dokter pada halaman resep obat berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.6.

Tabel 4.6 Blackbox Testing Dokter Halaman Resep Obat

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
DR15	Mengakses Halaman Resep Obat	Sistem akan menampilkan halaman resep obat dengan tampilan awal <i>form</i> data pasien	Berhasil
DR16	Mengisi seluruh <i>input form</i> data pasien dan menekan selanjutnya	Sistem akan menampilkan konfirmasi “Data Pasien Sudah Benar?”	Berhasil
DR17	Menekan tombol Sudah setelah ada konfirmasi “Data Pasien sudah Benar?”	Sistem akan menampilkan <i>form</i> tampilan kedua yaitu <i>form</i> obat	Berhasil
DR18	Mengosongkan atau mengisi salah satu <i>input form</i> data pasien	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini” dibawah input yang belum terisi	Berhasil
DR19	Mengisi seluruh <i>input form</i> data obat	Sistem akan menampilkan	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
	dan menekan selanjutnya	konfirmasi “Data Obat Sudah Benar?”	
DR20	Menekan tombol sudah setelah ada konfirmasi “Data Obat sudah Benar?”	Sistem akan menampilkan <i>form</i> tampilan ketiga yaitu <i>form</i> resep obat	Berhasil
DR21	Mengosongkan atau mengisi salah satu <i>input form</i> data resep obat	Sistem akan menampilkan pesan <i>error</i> “Harap isi semua bidang yang diperlukan” diatas tombol selanjutnya	Berhasil
DR22	Mengisi seluruh <i>input form</i> data resep obat dan menekan selanjutnya	Sistem akan menampilkan konfirmasi “Data Resep Obat Sudah Benar?”	Berhasil
DR23	Menekan tombol Sudah setelah ada konfirmasi “Data Resep Obat sudah Benar?”	Sistem akan menampilkan <i>form</i> awal di halaman resep obat	Berhasil

4.2.4.2 *Black Box Testing* Dengan Role Apoteker

Pada pengujian *blackbox* testing pada role apoteker berjalan sesuai dengan yang diharapkan. Adapun hasil pengujian yang telah dilakukan dapat dilihat pada tabel-tabel yang sudah diujikan.

a. Apoteker Belum *Login*

Pada pengujian apoteker saat belum *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.7.

Tabel 4.7 Blackbox Testing Apoteker Sebelum *Login*

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP01	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i>	Berhasil
AP02	Mengakses Halaman <i>Dashboard</i>	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil
AP03	Mengakses Halaman Riwayat Resep	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil

b. Apoteker Pada Halaman *Login*

Pada pengujian dokter pada halaman *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.8.

Tabel 4.8 Blackbox Testing Apoteker Halaman *Login*

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP04	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i> dengan <i>form email</i> dan kata sandi	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP05	Mengisi email <u>apoteker@gmail.com</u> dengan kata sandi “apoteker” dan menekan tombol <i>login</i>	Sistem akan memproses masukan dari <i>user</i> dan mengarahkan ke halaman <i>dashboard</i>	Berhasil
AP06	Mengisi email dan kata sandi yang tidak terdaftar	Sistem akan menampilkan pesan <i>error</i> “User tidak ditemukan”	Berhasil
AP07	Mengisi email yang terdaftar dan mengisi kata sandi yang salah	Sistem akan menampilkan pesan <i>error</i> “Kata Sandi Salah”	Berhasil
AP08	Hanya mengisikan email dan mengosongkan kata sandi	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini” dibawah input kata sandi	Berhasil
AP09	Hanya Mengisikan kata sandi dan mengosongkan email	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini” dibawah input email	Berhasil

c. Apoteker Pada Halaman *Dashboard*

Pada pengujian apoteker pada halaman *dashboard* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.9.

Tabel 4.9 Blackbox Testing Apoteker Halaman *Dashboard*

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP10	Mengakses Halaman <i>Dashboard</i>	Sistem akan menampilkan halaman <i>dashboard</i> yang menunjukkan deskripsi klinik, jumlah pasien hari ini, jumlah resep obat hari ini, resep obat yang sudah dibuat, dan grafik <i>patient chart</i>	Berhasil
AP11	Meletakkan kursor pada titik <i>chart</i> atau grafik pada tanggal satu minggu yang dipilih	Sistem akan menampilkan tanggal dan bulan yang di pilih dan menampilkan semua pasien dalam satu minggu	Berhasil
AP12	Menekan <i>icon</i> untuk <i>zoom in</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran yang lebih tinggi dari tampilan normal	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP13	Menekan <i>icon</i> untuk <i>zoom out</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran yang lebih rendah dan kembali normal setelah di <i>zoom</i>	Berhasil
AP14	Menekan titik pada grafik pasien satu minggu pada data <i>patient chart</i>	Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih	Berhasil

d. Apoteker Pada Halaman Stok Resep

Pada pengujian apoteker pada halaman riwayat resep berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.10.

Tabel 4.10 Blackbox Testing Apoteker Halaman Stok Resep

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP15	Mengakses Halaman Stok Resep	Sistem akan menampilkan stok resep dengan tabel yang berisi data resep dan juga stok resep	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP16	Menekan <i>icon filter</i> sebelah status stok lalu memilih “Tersedia” dan menekan Ok	Sistem akan menampilkan data resep hanya yang Tersedia pada status stok	Berhasil
AP17	Menekan tombol Bersihkan Filter	Sistem akan mereset filter yang dipilih dan mengembalikan semua data seperti semula	Berhasil
AP18	Menekan tombol Bersihkan filter dan sortir	Sistem akan mereset filter dan sortir yang dipilih dan mengembalikan semua data seperti semula	Berhasil
AP19	Menekan tombol Tambah Resep	Sistem akan menampilkan <i>pop up model</i> dengan menampilkan <i>form</i> untuk tambah resep	Berhasil
AP20	Mengisi form pada tambah resep dan menekan tombol OK	Sistem akan menutup <i>pop up</i> dan terdapat notifikasi “Stok resep berhasil ditambahkan”	Berhasil
AP21	Menekan tombol Edit pada data resep yang dipilih	Sistem akan menampilkan <i>pop up model</i> dengan menampilkan <i>form</i>	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		yang berisi data resep yang dipilih	
AP22	Mengedit pada <i>form</i> edit resep	Sistem akan menutup <i>pop up</i> dan terdapat notifikasi “Stok resep berhasil diperbarui”	Berhasil
AP23	Menekan tombol Hapus pada data resep yang dipilih	Sistem akan menampilkan konfirmasi “Apakah mau menghapus stok resep ini?”	Berhasil
AP24	Menekan “Ya” pada konfirmasi	Maka sistem akan menghapus data resep yang dipilih dan terdapat notifikasi “Data resep berhasil dihapus”	Berhasil

e. Apoteker Pada Halaman Riwayat Resep

Pada pengujian apoteker pada halaman riwayat resep berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.11.

Tabel 4.11 Blackbox Testing Apoteker Halaman Riwayat Resep

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP25	Mengakses Halaman Riwayat Resep	Sistem akan menampilkan halaman riwayat	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		resep dengan tabel data pasien dan terdapat proses resep	
AP26	Menekan <i>icon search</i> sebelah nama pasien	Sistem akan menampilkan pencarian nama dibawah nama pasien	Berhasil
AP27	Mengisi nama “Dika” dan menekan tombol <i>search</i>	Sistem akan menampilkan data pasien Dika	Berhasil
AP28	Mengosongkan pada pencarian nama pasien	Sistem akan menampilkan isi tabel kosong dan ada tulisan “No data”	Berhasil
AP29	Menekan tombol reset setelah tidak ada pencarian nama	Sistem akan menampilkan tampilan seperti awal data pasien	Berhasil
AP30	Menekan tombol Buat Resep pada aksi	Sistem akan menampilkan <i>pop up model</i> dengan data resep obat pasien, dan pada proses resep tabel pasien berubah menjadi “Proses”	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AP31	Menekan tombol Selesai Dibuat pada <i>pop up model</i>	Sistem akan menutup <i>pop up model</i> dan proses resep pada tabel pasien berubah menjadi “Selesai”	Berhasil
AP32	Menekan tombol Cetak Resep pada <i>pop up model</i>	Sistem akan menampilkan lembar halaman untuk mencetak	Berhasil
AP33	Menekan tombol cetak pada halaman untuk di cetak	Sistem akan menampilkan pilih folder mana untuk disimpan setelah mencetak	Berhasil

4.2.4.3 *Black Box Testing* Dengan Role Admin

Pada pengujian *blackbox* testing pada role *admin* berjalan sesuai dengan yang diharapkan. Adapun hasil pengujian yang telah dilakukan dapat dilihat pada tabel-tabel yang sudah diujikan.

a. Admin Sebelum Login

Pada pengujian *admin* saat belum *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.12.

Tabel 4.12 *Blackbox Testing Admin Sebelum Login*

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD01	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i>	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD02	Mengakses Halaman <i>Dashboard</i>	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil
AD03	Mengakses Halaman Pasien	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil
AD04	Mengakses Halaman <i>Users</i>	Sistem akan menolak akses dan mengarahkan ke halaman <i>login</i>	Berhasil

b. *Admin Pada Halaman Login*

Pada pengujian *admin* pada halaman *login* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.13.

Tabel 4.13 Blackbox Testing Admin Halaman Login

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD05	Mengakses Halaman <i>Login</i>	Sistem akan menampilkan halaman <i>login</i> dengan <i>form email</i> dan kata sandi	Berhasil
AD06	Mengisi email <u>admin@gmail.com</u> dengan kata sandi “admin” dan	Sistem akan memproses masukan dari <i>user</i> dan mengarahkan ke halaman <i>dashboard</i>	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
	menekan tombol <i>login</i>		
AD07	Mengisi email dan kata sandi yang tidak terdaftar	Sistem akan menampilkan pesan <i>error</i> “User tidak ditemukan”	Berhasil
AD08	Mengisi email yang terdaftar dan mengisi kata sandi yang salah	Sistem akan menampilkan pesan <i>error</i> “Kata Sandi Salah”	Berhasil
AD09	Hanya mengisikan email dan mengosongkan kata sandi	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini” dibawah input kata sandi	Berhasil
AD10	Hanya Mengisikan kata sandi dan mengosongkan email	Sistem akan menampilkan pesan <i>error</i> “Harap isi bidang ini” dibawah input email	Berhasil

c. Admin Pada Halaman *Dashboard*

Pada pengujian *admin* pada halaman *dashboard* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.14.

Tabel 4.14 Blackbox Testing Admin Halaman Dashboard

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD11	Mengakses Halaman <i>Dashboard</i>	Sistem akan menampilkan halaman dashboard yang menunjukkan deskripsi klinik, jumlah pasien hari ini, jumlah resep obat hari ini, resep obat yang sudah dibuat, dan grafik <i>patient chart</i>	Berhasil
DR12	Meletakkan cursor pada titik <i>chart</i> atau grafik pada tanggal satu minggu yang dipilih	Sistem akan menampilkan tanggal dan bulan yang di pilih dan menampilkan semua pasien dalam satu minggu	Berhasil
AD13	Menekan icon untuk <i>zoom in</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran yang lebih tinggi dari tampilan normal	Berhasil
AD14	Menekan icon untuk <i>zoom out</i> pada <i>patient chart</i>	Sistem akan menampilkan data <i>chart</i> dengan tingkat perbesaran	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		yang lebih rendah dan kembali normal setelah di <i>zoom</i>	
AD15	Menekan titik pada grafik pasien satu minggu pada data <i>patient chart</i>	Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih	Berhasil

d. *Admin* Pada Halaman Pasien

Pada pengujian *admin* pada halaman pasien berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.15.

Tabel 4.15 Blackbox Testing Admin Halaman Pasien

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD16	Mengakses Halaman Pasien	Sistem akan menampilkan halaman Pasien dengan semua data pasien	Berhasil
AD17	Menekan <i>icon search</i> sebelah nama pasien	Sistem akan menampilkan pencarian nama dibawah nama pasien	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD18	Mengisi nama “Dika” dan menekan tombol search	Sistem akan menampilkan data pasien Dika	Berhasil
AD19	Menekan tombol reset pada <i>search</i> nama	Sistem akan mengembalikan nama jadi kosong yang diinputkan pada pencarian nama	Berhasil
AD20	Menekan tombol Edit pada aksi	Sistem akan menampilkan halaman baru untuk mengedit data pasien	Berhasil
AD21	Mengisi semua data pasien dan menekan Simpan	Sistem akan kembali ke data semua pasien pada halaman pasien	Berhasil
AD22	Mengisi salah satu atau mengosongkan semua form edit pasien	Sistem menampilkan pesan <i>error</i> “Harap isi bidang ini.” Pada inputan kosong	Berhasil

e. Admin Pada Halaman *Users*

Pada pengujian admin pada halaman *users* berjalan dengan lancar sesuai yang diharapkan, adapun hasil pengujian dapat dilihat pada tabel 4.16.

Tabel 4.16 Blackbox Testing Admin Halaman Users

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD23	Mengakses Halaman <i>Users</i>	Sistem akan menampilkan halaman <i>Users</i> dengan tampilan tabel untuk semua <i>user</i>	Berhasil
AD24	Menekan <i>icon</i> untuk <i>sort ascending</i> atau <i>descending</i>	Sistem akan menampilkan nama yang teratas jadi paling bawah dan nama terbawah menjadi paling atas	Berhasil
AD25	Menekan <i>icon</i> untuk filter role menjadi dokter saja	Sistem akan menampilkan data pengguna yaitu dokter saja yang terlihat	Berhasil
AD26	Menekan <i>icon</i> untuk filter <i>Active</i> menjadi <i>Inactive</i> saja	Sistem akan menampilkan data pengguna yitu hanya yang <i>inactive</i> saja	Berhasil
AD27	Menekan Bersihkan filter	Sistem akan mengembalikan filter yang sudah dipilih menjadi seperti semula	Berhasil
AD28	Bersihkan filter dan sortir	Sistem akan mengembalikan filter yang sudah dipilih dan sortir	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		yang dipilih menjadi seperti semula	
AD29	Menekan tombol <i>Add User</i>	Sistem akan menampilkan halaman baru <i>form add user</i>	Berhasil
AD30	Input semua isi <i>form add user</i> dan menetakan simpan	Sistem akan menampilkan tampilan awal ke data semua pengguna	Berhasil
AD31	Mengisi salah satu atau mengosongkan semua <i>form add user</i>	Sistem menampilkan pesan <i>error</i> “Harap isi bidang ini.” Pada inputan kosong	Berhasil
AD32	Menekan tombol Edit pada aksi	Sistem akan menampilkan halaman baru untuk <i>form edit user</i>	Berhasil
AD33	Mengisi semua data <i>user</i> dan menekan Simpan	Sistem akan kembali ke data semua pengguna pada halaman <i>users</i>	Berhasil
AD34	Mengisi salah satu atau mengosongkan semua <i>form edit pasien</i>	Sistem menampilkan pesan <i>error</i> “Harap isi bidang ini.” Pada inputan kosong	Berhasil

ID	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
AD35	Menekan Nonaktifkan pada Aksi	Sistem akan menampilkan konfirmasi “Apakah mau nonaktifkan pengguna ini?” dan ada pilihan “Ya” dan “Tidak”	Berhasil
AD36	Menekan Ya pada konfirmasi “Apakah mau nonaktifkan pengguna ini?”	Sistem akan menampilkan status pesan” Sukses menonaktifkan pengguna”	Berhasil
AD35	Menekan Aktifkan pada Aksi	Sistem akan menampilkan konfirmasi “Apakah mau aktifkan kembali pengguna ini?” dan ada pilihan “Ya” dan “Tidak”	Berhasil
AD36	Menekan Ya pada konfirmasi “Apakah mau aktifkan kembali pengguna ini?”	Sistem akan menampilkan status pesan” Sukses mengaktifkan kembali pengguna”	Berhasil

4.2.4.4 Hasil Pengujian

Pengujian fungsionalitas yang dilakukan pada pengembangan aplikasi sistem *electronic prescribing* dokter pada penelitian ini adalah pengujian menggunakan *blackbox testing*. Pengujian fungsionalitas aplikasi yang dilaksanakan dilakukan oleh dokter dan apoteker yang didampingi oleh penulis memperoleh persentase

kesuksesan sebesar 100% yang berarti pada setiap skenario pengujian bernilai berhasil. Dengan hasil ini berarti sistem dapat berfungsi dengan baik.

4.2.5 Hasil Analisis

Analisis ini bertujuan untuk membandingkan waktu yang dibutuhkan dalam setiap pembuatan resep obat, mulai dari dokter menuliskan resep obat, dokter menyerahkan obat ke apoteker, apoteker meracik obat sampai dokter menyerahkan obat kepada pasien. Data dari proses pembuatan resep obat secara manual akan dibandingkan dengan data dari sistem electronic prescribing untuk menentukan perbedaan rata rata waktu, persentase perbedaan, dan rasio efisiensi. Hasil analisis ini akan memberikan gambaran tentang seberapa besar peningkatan efisiensi yang dicapai melalui implementasi sistem electronic prescribing, serta manfaat yang diperoleh dalam hal penghematan waktu dan sumber daya. Pada tabel 4.17 merupakan waktu yang dibutuhkan untuk melakukan pembuatan resep sampai menjadi obat untuk pasien sebelum diimplementasikan sistem *electronic prescribing*.

Tabel 4.17 Waktu proses pembuatan resep sampai jadi obat secara manual

No	Tahapan	Waktu yang dibutuhkan (menit)
1	Dokter menulis data pasien dan menulis resep obat	3-4
2	Dokter menyerahkan resep obat ke apoteker	2
3	Apoteker melihat resep dari dokter lalu meracik obat sesuai intruksi yang ditulis oleh dokter	10-15
4	Apoteker menyerahkan obat ke pasien dan menjelaskan intruksi penggunaan	2

No	Tahapan	Waktu yang dibutuhkan (menit)
	Total waktu minimum	17 menit
	Total waktu maksimum	23 menit

Pada tabel 4.18 merupakan data waktu yang dibutuhkan untuk melakukan pembuatan resep dengan sistem *electronic prescribing*.

Tabel 4.18 Waktu Proses menggunakan sistem *electronic prescribing*

No	Tahapan	Waktu yang dibutuhkan (menit)
1	Dokter mencari/menambah data pasien dan mencari resep obat dalam sistem	1-2
2	Apoteker melihat data resep dari sistem dan meracik obat sesuai data dari sistem	10-12
3	Apoteker menyerahkan obat ke pasien dan menjelaskan intruksi penggunaan	2
	Total waktu minimum	13 menit
	Total waktu maksimum	16 menit

Selanjutnya penulis akan menganalisis tingkat efisiensi proses pembuatan resep setelah implementasi sistem *electronic prescribing*, penulis menghitung perbedaan rata rata waktu, persentase perbedaan, dan rasio efisiensi, berikut ini adalah perhitungan yang dilakukan oleh penulis:

1. Rata-rata waktu sebelum dilakukan implementasi

Rata-rata waktu sebelum dilakukan implementasi yaitu menghitung rata-rata waktu sesuai dengan Persamaan 2.2.

$$Rata - rata waktu = \frac{17 \text{ menit} + 23 \text{ menit}}{2}$$

$$Rata - rata waktu = \frac{38 \text{ menit}}{2}$$

$$Rata - rata waktu = 20 \text{ menit/pasien}$$

2. Rata-rata waktu setelah dilakukan implementasi

Rata-rata dilakukan untuk menghitung rata-rata waktu setelah dilakukan implementasi sesuai dengan Persamaan 2.2.

$$\text{Rata - rata waktu} = \frac{13 \text{ menit} + 16 \text{ menit}}{2}$$

$$\text{Rata - rata waktu} = \frac{31 \text{ menit}}{2}$$

$$\text{Rata - rata waktu} = 14,5 \text{ menit/pasien}$$

Setelah berhasil menghitung rata-rata waktu, selanjutnya menghitung perbedaan rata-rata waktu.

3. Perbedaan rata-rata waktu antara sebelum dan setelah implementasi sistem *electronic prescribing*.

Dibawah ini untuk menghitung perbedaan rata-rata yang sesuai dengan Persamaan 2.3.

$$\text{Perbedaan rata - rata} = 20 \text{ menit} - 14,5 \text{ menit}$$

$$\text{Perbedaan rata - rata} = 5,5 \text{ menit/pasien}$$

Setelah menghitung perbedaan rata-rata, selanjutnya menghitung rasio persentase perbedaan.

4. Rasio persentase perbedaan

Pada rasio persentase perbedaan, hal ini untuk menghitung rasio persentase perbedaan, sesuai dengan Persamaan 2.4.

$$\text{Persentase perbedaan} = \left(\frac{5,5 \text{ menit/pasien}}{20 \text{ menit/pasien}} \right) \times 100\%$$

$$\text{Persentase perbedaan} \approx 27,5 \%$$

5. Rasio efisiensi

Pada rasio efisiensi, hal ini untuk menghitung rasio efisiensi yang sesuai dengan Persamaan 2.5.

$$\text{Rasio efisiensi} = \frac{20 \text{ menit}}{14,5 \text{ menit}}$$

$$\text{Rasio efisiensi} \approx 1,38$$

Setelah menghitung rasio efisiensi, selanjutnya menghitung persentase peningkatan efisiensi.

6. Persentase peningkatan efisiensi

Persentase peningkatan efisiensi ini untuk menghitung persentase peningkatan efisiensi yang sesuai dengan Persamaan 2.6.

$$\text{Persentase peningkatan efisiensi} = \left(\frac{1,38 - 1}{1} \right) \times 100\%$$

$$\text{Persentase peningkatan efisiensi} = 0,38 \times 100\% = 38\%$$

Setelah dilakukan analisis terhadap implementasi dengan sistem *electronic prescribing*, terjadi peningkatan persentase efisiensi dalam proses pembuatan resep. Waktu yang dibutuhkan setiap pasien setelah dilakukan perhitungan dengan rata rata yaitu berkurang sebesar 5 menit, hal ini juga menunjukkan penurunan sebesar 27,5% waktu yang dibutuhkan dalam pembuatan resep sampai menjadi obat. Menunjukkan efisiensi sistem yang lebih baik dengan sebelumnya yaitu dengan persentase peningkatan efisiensi 38% yang artinya sistem lebih baik dari sebelumnya saat pembuatan resep obat sampai menjadi obat masih menggunakan cara manual. Implementasi sistem *electronic prescribing* juga terbukti tidak hanya mengurangi waktu tunggu pasien tetapi juga secara signifikan meningkatkan efisiensi dalam manajemen data resep obat di Klinik Asy-Syifa.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan dari proses identifikasi masalah hingga hasil analisis, serta pengembangan *website* Sistem *Electronic Prescribing* di Klinik Asy-Syifa, dapat disimpulkan bahwa pengembangan aplikasi sistem *Electronic Prescribing* dokter menggunakan bahasa pemrograman *JavaScript* berhasil diselesaikan. Proses ini melibatkan seluruh tahapan *Extreme Programming* dan analisis data yang ada. Hasil analisis menunjukkan adanya pengurangan waktu yang dibutuhkan untuk setiap pasien sebesar 5,5 menit atau sekitar 27,5%. Selain itu, terdapat peningkatan efisiensi sebesar 38% dibandingkan dengan penggunaan sistem manual sebelumnya. Hasil ini menunjukkan bahwa implementasi sistem *Electronic Prescribing* tidak hanya mengurangi waktu dalam proses, tetapi juga secara signifikan meningkatkan efisiensi dalam manajemen data resep obat di Klinik Asy-Syifa.

5.2 Saran

Terdapat sedikit banyak kekurangan dari penelitian ini yang masih perlu dibenahi agar penelitian selanjutnya lebih bagus lagi, penulis merekomendasikan beberapa masukan yang dapat digunakan penulisan selanjutnya. Saran dan masukan yang diberikan penulis adalah agar pengujian juga dilakukan dengan metode lain selain *blackbox*. Penulis juga memberikan masukan agar penelitian selanjutnya menyempurnakan tampilan dan juga mengoptimalkan aplikasi bisa menjadi *responsive* menyesuaikan dengan kebutuhan pengguna nantinya. Dengan demikian, diharapkan perbaikan ini dapat meningkatkan kemudahan pengguna aplikasi sistem *electronic prescribing* dokter di Klinik Asy-Syifa Desa Randudongkal.

DAFTAR PUSTAKA

- [1] M. Mulyadi, I. Zulkarnain, and N. Laugu, “Adaptasi pustakawan dalam menghadapi kemajuan teknologi,” *Berkala Ilmu Perpustakaan dan Informasi*, vol. 15, no. 2, pp. 163–174, Nov. 2019, doi: 10.22146/bip.39843.
- [2] Mahisataruna, Arman, Rajab, and Nelfira, “Perancangan Sistem Informasi Koperasi Syariah Berbasis Web Pada Bundo Saiyo Padang,” *Jurnal IndraTech*, vol. 2, no. 2, pp. 1–10, 2021.
- [3] M. Hamdan Romadhon and Y. Yudhistira, “Sistem Informasi Rental Mobil Berbsasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri,” 2021. [Online]. Available: www.journal.peradaban.ac.id
- [4] A. Amarulloh, “Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django, dan Node JS Pada Aplikasi Berbasis Website,” *Jurnal Teknik Informatika STMIK Antar Bangsa*, vol. 09, no. 01, pp. 12–17, 2023.
- [5] D. Arta and E. Yulianingsih, “Sistem Informasi E-Prescribing (Stusi kasus : Instalasi Rawat Jalan Pada RS TK II DR AK Gani Palembang),” *Bina Darma Conference on Computer Science*, vol. 3, no. 1, pp. 688–697, 2019.
- [6] K. Pengobatan Kholidatul Ulum and I. Laily Hilmi, “Review Artikel : Implementasi dan Evaluasi Persepsi Elektronik Dalam Upaya Menurunkan Kesalahan Pengobatan,” *Journal of Pharmaceutical and Sciences*, vol. 6, no. 1, pp. 191–198, 2023, [Online]. Available: <https://www.jurnal-jps.com>
- [7] M. L. Chresna Merry P., “Gambaran Resep Elektronik Terhadap Waktu Tunggu Obat Jadi Pada Instalansi Farmasi Rawat Jalan di Rumah Sakit X Surabaya Periode 20 – 27 Februari 2019,” *Jurnal Farmasi Indonesia*, vol. 1, no. 1, pp. 34–43, 2019.

- [8] Z. Halim, U. Muhammadiyah HAMKA Jl Tanah Merdeka No, K. Rambutan, P. Rebo, and J. Timur, “Penerapan Sistem Informasi Akademik Dengan Metode Extreme Programming,” *Sistem Informasi* |, vol. 8, no. 1, pp. 66–74, 2021.
- [9] T. Dedifuadi, A. Surahmat, and R. Fatullah, “Perancangan Sistem Informasi Resep Pulang Farmasi Rawat Inap Dengan Metode E-Prescribing Pada RS XYZ,” 2021.
- [10] S. Kom. ,M. K. R. A. S. Kom. ,M. K. Try Viananda Nova M, “Penerapan Sistem E-Prescribing dan Barcode system di Rumah Sakit Pertamina menggunakan PowerBuilder dengan Arsitektur Client - Server System,” *Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin*, vol. 3, no. 1, pp. 1–81, 2020.
- [11] H. Susilawati, T. A. Wiharso, and T. M. Hidayat, “Rancang Bangun Aplikasi Elektronik Resep yang Terhubung ke Kasir Menggunakan Website,” *Jurnal Teknologi Informasi dan Terapan*, vol. 7, no. 2, pp. 86–91, Dec. 2020, doi: 10.25047/jtit.v7i2.167.
- [12] D. Rakasiwi *et al.*, “Rancang Bangun Sistem Electronic Prescribing Dokter dengan Menggunakan Codeigniter,” 2019.
- [13] N. A. K. G. BANJARNAHOR DONDA, “Sistem Informasi Klinik Berbasis Website Menggunakan Metode Extreme Programming (Studi Kasus Klinik Karunia Bunda),” *PETIR*, vol. 14, no. 2, pp. 223–234, Sep. 2021, doi: 10.33322/petir.v14i2.1155.
- [14] A. A. AlAzmi, H. AlHamdan, O. Ahmed, S. Tomlin, and A. N. Rashed, “Impact of the e-prescribing system on the incidence and nature of drug-related problems in children in a Saudi hospital,” *International Journal of Pharmacy Practice*, vol. 27, no. 6, pp. 578–581, Dec. 2019, doi: 10.1111/ijpp.12548.
- [15] H. Bell, S. Garfield, S. Khosla, C. Patel, and B. D. Franklin, “Mixed methods study of medication-related decision support alerts experienced during electronic prescribing for inpatients at an English

- hospital,” *European Journal of Hospital Pharmacy* , vol. 26, no. 6, pp. 318–322, Nov. 2019, doi: 10.1136/ejhp pharm-2017-001483.
- [16] P. Marlina Ariansyah and K. Wijaya Sekolah Tinggi Manajemen Informatika dan Komputer Prabumulih, “Rancang Bangun Sistem Informasi Akademik Berbasis Web: Studi Kasus: SD Negeri 18 Tanah Abang,” 2021.
 - [17] M. Ronaldo and D. Pasha, “Sistem Informasi Pengelolaan Data Santri Pondok Pesantren AN-AHL Berbasis Website,” 2021.
 - [18] M. Zaini, N. Isnani, and P. Studi D-III Farmasi Politeknik Unggulan Kalimantan, “Pandangan Apoteker Dalam Penerapan e-Prescribing DI Rumah Sakit dan Farmasi Komunitas : Literatur Review,” *jurnal.polanka.ac.id/index.php/JKI KT*, vol. 5, no. 1, pp. 9–17, May 2023.
 - [19] P. A. C. Jamilah Yulianti Siti, “Perancangan User Interface Dan User Experience Aplikasi SAY.CO,” *TANRA*, vol. 9, no. 1, pp. 73–88, 2022.
 - [20] M. Aswiputri and K. Penulis, “Literature Review Determinasi Sistem Informasi Manajemen: Database, CCTV dan Brainware,” *JEMSI*, vol. 3, no. 3, pp. 312–322, 2022, doi: 10.31933/jemsi.v3i3.
 - [21] I. Maharani Safira, A. Rakhman, T. Elektro, P. Studi Telekomunikasi, P. Negeri Sriwijaya Jl Srijaya Negara, and B. Besar Palembang, “Rancang Bangun Aplikasi E-Ticketing Travel Antar Kota Berbasis Android Dengan Menggunakan Metode Waterfall,” 2019.
 - [22] A. Mubarak, J. J. Metro, and K. T. Selatan, “Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (UNIFIED MODELING LANGUAGE) Dan Bahasa Pemrograman PHP (PHP HYPERTEXT PREPROCESSOR) Berorientasi Objek,” 2019.
 - [23] F. Rozi, K. Khomsatun, and P. Teknologi Informasi STKIP PGRI Tulungagung Jl Mayor Sujadi Timur No, “Rancang Bangun Game Edukasi Pengenalan Warna Untuk Pendidikan Anak Usia Dini Menggunakan Adobe Flash Berbasis Android,” 2019.

- [24] H. Kurniawan, W. Apriliah, I. Kurniawan, and D. Firmansyah, “Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang,” *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 14, no. 4, pp. 13–23, Jan. 2020, doi: 10.35969/interkom.v14i4.58.
- [25] R. Aditya, V. Handrianus Pranatawijaya, P. Bagus Adidyana Anugrah Putra, J. Hendrik Timang, K. Palangkaraya, and K. Tengah, “Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype,” 2021.
- [26] A. Abdullah, A. Achmad, S. Sahibu, P. Studi Pascasarjana Sistem Komputer, and S. Handayani, “Media Pembelajaran Interaktif Mata Kuliah Pemrograman Web Berbasis Android,” 2021.
- [27] N. Sianturi and K. Puspita, “Keamanan Source Code Java Script Menggunakan Metode Stream Cipher Dengan Verifikasi MD5,” 2020.
- [28] A. F. Sallaby and I. Kanedi, “Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter,” 2020.
- [29] A. S. Sari and R. Hidayat, “Designing website vaccine booking system using golang programming language and framework react JS,” *Journal of Information System, Informatics and Computing Issue Period*, vol. 6, no. 1, pp. 22–39, 2022, doi: 10.5236/jisicom.v6i1.760.
- [30] R. Lisgiani and S. Nurmajid, “Implementasi Autentikasi Dari Sisi Backend Pada Arsitektur Microservices Menggunakan Express JS,” *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 7, no. 1, pp. 27–32, Jun. 2022, doi: 10.32897/infotronik.2022.7.1.1330.
- [31] R. Marta Syakira, “Rancang Bangun Data Storage System berbasis Web Dengan Metode Extreme Programming,” *Jurnal TEKNO KOMPAK*, vol. 16, no. 1, pp. 123–135, 2022.
- [32] Dedy Armady, “Pengembangan Aplikasi Berbasis Responsive Website Menggunakan Metode Extreme Programming,” *Jurnal TIKA*, vol. 7, no. 2, pp. 149–157, 2022.

- [33] I. Carolina, A. Supriyatna, J. Kamal Raya No, and R. Barat Cengkareng Jakarta Barat, “Penerapan Metode EXTREME PROGRAMMING Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen,” 2019.
- [34] N. A. Septiani and F. Y. Habibie, “Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 3, pp. 341–349, Mar. 2022, doi: 10.30865/json.v3i3.3931.
- [35]) Hardiyanto,) Abdussomad, E. Haryadi, R. Sopandi, and) Asep, “Penerapan Model WATERFALL dan UML Dalam Rancang Bangun Program Pembelian Barang Berorientasi Objek Pada PT. FUJITA Indonesia,” 2019.
- [36] A. Arif, “Penerapan Metode Extreme Programming Pada E-Voting Pemilihan Ketua Unit Kegiatan Mahasiswa (UKM) Sekolah Tinggi Teknologi XYZ,” *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 9, no. 2, pp. 234–240, Apr. 2021, doi: 10.26418/justin.v9i2.44266.
- [37] A. Pradana Putra, F. Andriyanto, T. Dewi Muji Harti, and W. Puspitasari, “Pengujian Aplikasi Point of Sale Berbasis Web Menggunakan Blackbox Testing,” 2020.
- [38] E. Novalia and A. Voutama, “Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah,” 2022.
- [39] F. A. Fauzi, G. E. Putra, S. Supriyanto, N. A. Saputra, and T. Desyani, “Pengujian Terhadap Aplikasi Parking Management Menggunakan Metode Black-Box Berbasis Equivalence Partitions,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 2, pp. 64–68, Apr. 2020, doi: 10.32493/jtsi.v3i2.4685.
- [40] K. H. Iin Ivanda Listari, “Analisis Rasio Efektivitas, Rasio Efisiensi, Dan Rasio Kemandirian Dalam Pengelolaan Keuanga Alokasi Dana Desa (2018-2021),” *Jurnal Mahasiswa* , vol. 4, no. 3, pp. 129–140, 2022.

LAMPIRAN

Lampiran 1. Surat Permohonan Pengambilan Data Penelitian

Nomor : IT Tel10043/AKA-000/DKN-02/XI/2023
Lampiran : -



Kepada Yth.
Dokter
Klinik Asy-Syifa
Jl. Randudongkal - Belik, Desa Randudongkal, Kabupaten Pemalang, Jawa Tengah 52356

Perihal : Permohonan pengambilan data untuk penelitian Tugas Akhir / Skripsi

Dengan hormat,

1. Diberitahukan bahwa untuk memenuhi salah satu persyaratan kelulusan di Institut Teknologi Telkom Purwokerto, mahasiswa program Sarjana diwajibkan melaksanakan Tugas Akhir / Skripsi dalam bentuk penulisan karya ilmiah..
2. Dalam rangka hal tersebut, adapun mahasiswa yang akan mengambil data sebagai berikut :
NIM : 20102036
Nama : Muhammad Arif Alifan Sulton
Program Studi : Program Studi Teknik Informatika (S1)
Telepon : 088225431200
3. Adapun data yang akan diambil adalah **Observasi dan Wawancara**
4. Sehubungan dengan hal tersebut, apabila data informasi yang dimaksud bukan merupakan data yang konfidensial, mohon kiranya kepada mahasiswa kami dapat diberikan bantuan seperlunya.
5. Demikian kami sampaikan permohonan ini. Atas kebijaksanaan dan kerjasamanya kami ucapkan terima kasih.

Purwokerto, 27 November 2023

Dekan Fakultas Informatika



Auliya Burhanuddin, S.Si., M.Kom
NIK : 19810008



Lampiran 2. Daftar Pertanyaan Wawancara

Daftar Pertanyaan Wawancara

Dalam pertanyaan wawancara ini berfungsi untuk menjawab rumusan masalah pada penelitian ini. Berikut daftar pertanyaan wawancara:

1. Bagaimana proses pembuatan resep obat di klinik dari dokter hingga apoteker?
2. Berapa lama waktu yang diperlukan untuk pembuatan resep obat?
3. Adakah kendala saat pembuatan resep obat hingga menjadi obat?
4. Apa yang diharapkan jika pembuatan *website sistem electronic prescribing* ini dibuat?
5. Apa saja fitur yang diharapkan pada pembuatan *wesbite sistem electronic prescribing*?
6. Apakah pihak klinik memberi dukungan pada pengembangan dan implementasi pada sistem ini?

Lampiran 3. Hasil Wawancara

Jawaban Hasil Wawancara

Nama Responden : Dr. Ahmad Alifin

Waktu Wawancara : 13 November 2023

1. Proses pembuatan resep obat di klinik melibatkan beberapa langkah yang melibatkan dokter yang meresepkan obat, peran tenaga medis lainnya jika diperlukan, dan apoteker yang mengisi dan menyediakan obat kepada pasien, atau sebagai berikut untuk langkah-langkahnya:
 - Dokter akan melakukan pemeriksaan fisik, mendiagnosa kondisi pasien, dan menuliskan resep obat yang dianggap tepat sesuai dengan kondisi tersebut.
 - Dokter akan menuliskan resep obat yang mencakup informasi tentang obat yang diresepkan, dosis, frekuensi konsumsi, dan instruksi penggunaan lainnya.
 - Apoteker membawa resep yang ditulis oleh dokter untuk dibuat obat sesuai instruksi dokter sebelumnya.
 - Setelah resep diisi, apoteker akan memberikan obat kepada pasien. Apoteker juga memberikan penjelasan mengenai dosis yang benar, penggunaan obat, interaksi obat, efek samping yang mungkin terjadi, serta instruksi lain yang relevan untuk penggunaan obat tersebut.
2. Waktu yang perlukan untuk pembuatan resep obat sampai menjadi obat itu kurang lebih 20 menit.
3. Kendala saat pembuatan resep obat yaitu terkadang dokter bisa membuat kesalahan dalam menulis resep, seperti kesalahan penulisan nama obat, dosis, frekuensi penggunaan, atau instruksi penggunaan yang kurang jelas. Dan beberapa obat mungkin memerlukan proses pembuatan atau persiapan khusus yang membutuhkan waktu lebih lama.

4. Dengan adanya sistem ini diharapkan membantu klinik dari mengurangi kesalahan manusia dalam penulisan resep, mengurangi kemungkinan kesalahan dosis, obat yang salah, atau instruksi yang ambigu.
5. Sistem *login* dengan fitur untuk otentikasi pengguna dengan *login* yang aman untuk dokter dan apoteker. Fitur kemampuan dokter untuk membuat resep obat secara elektronik dengan mengisi detail obat, dosis, instruksi penggunaan.
6. Dari pihak klinik akan mendukung penuh dalam pembuatan sistem ini dan juga akan memberi dukungan teknis maupun pemeliharaan sistem ini nantinya.

Lampiran 4 Rancangan User Interface

Tautan Halaman:

<https://www.figma.com/design/w2eGqTpsN3D8zerlrEYLAY/Klinik-Asy-Syifa?node-id=0-1&t=oJyQGpVpJ3e6DxM2-0>

Lampiran 5 Penyerahan Website