# TUGAS AKHIR

**RANCANG BANGUN APLIKASISISTEM *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 APLIKASISISTEM *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**

# A close-up of a document Description automatically generatedLEMBAR PERSETUJUAN PEMBIMBING

Icon

Description automatically generated

**RANCANG BANGUN APLIKASISISTEM *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

Icon

Description automatically generated**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**

Penguji 1 Penguji 2

(Arief Rais Bahtiar, S.Kom., M.Kom.) (Novanda Alim Setya Nugraha, S.S., M.Hum.)

NIDN. 0604119101 NIDN. 0627119002

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

Pembimbing Utama,

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

NIDN. 0616098901

Dekan,

NIK.19820008

# HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertandatangan di bawah ini,

**Nama Mahasiswa : Muhammad Arif Alifan Sultoni**

**NIM : 20102036**

**Program Studi : Teknik Informatika**

Menyatakan bahwa Tugas Akhir dengan judul berikut:

**RANCANG BANGUN APLIKASI SISTEM E*LECTRONIC 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,

(Materai)

(Muhammad Arif Alifan Sultoni)

# KATA PENGANTAR

Dengan rasa syukur dan pujian, 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 Asmaiyah 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 menemani 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.

A black background with a black text

Description automatically generatedPurwokerto, 5 Juni 2024

Penulis,

Muhammad Arif Alifan Sultoni

# DAFTAR ISI

[TUGAS AKHIR i](#_Toc170923798)

[LEMBAR PERSETUJUAN PEMBIMBING ii](#_Toc170923799)

[LEMBAR PENGESAHAN iii](#_Toc170923800)

[HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv](#_Toc170923801)

[KATA PENGANTAR v](#_Toc170923802)

[DAFTAR ISI vii](#_Toc170923803)

[DAFTAR GAMBAR x](#_Toc170923804)

[DAFTAR TABEL xii](#_Toc170923805)

[DAFTAR LAMPIRAN xiv](#_Toc170923806)

[ABSTRAK xv](#_Toc170923807)

[*ABSTRACT* xvi](#_Toc170923808)

[BAB I PENDAHULUAN 1](#_Toc170923809)

[1.1 Latar Belakang 1](#_Toc170923810)

[1.2 Rumusan Masalah 5](#_Toc170923811)

[1.3 Pertanyaan Penelitian 5](#_Toc170923812)

[1.4 Batasan Masalah 5](#_Toc170923813)

[1.5 Tujuan Penelitian 6](#_Toc170923814)

[1.6 Manfaat Penelitian 6](#_Toc170923815)

[BAB II TINJAUAN PUSTAKA 7](#_Toc170923816)

[2.1 Tinjauan Pustaka 7](#_Toc170923817)

[2.2 Landasan Teori 14](#_Toc170923818)

[2.2.1 Rancang Bangun 14](#_Toc170923819)

[2.2.2 *Website* 14](#_Toc170923820)

[2.2.3 Sistem *Electronic Prescribing* 14](#_Toc170923821)

[2.2.4 *User Interface* 15](#_Toc170923822)

[2.2.5 *Database MySQL* 15](#_Toc170923823)

[2.2.6 UML 16](#_Toc170923824)

[2.2.9 Pemrograman *Web* 21](#_Toc170923825)

[2.2.10 *JavaScript* 21](#_Toc170923826)

[2.2.11 *Framework* 22](#_Toc170923827)

[2.2.12 Metode *Extreme Programming* 22](#_Toc170923828)

[2.2.13 *Blackbox Testing* 26](#_Toc170923829)

[2.2.14 Rasio Efisiensi 27](#_Toc170923830)

[BAB III METODE PENELITIAN 29](#_Toc170923831)

[3.1 Subjek Dan Objek Penelitian 29](#_Toc170923832)

[3.1.1 Subjek Penelitian 29](#_Toc170923833)

[3.1.2 Objek Penelitian 29](#_Toc170923834)

[3.2 Alat Dan Bahan 29](#_Toc170923835)

[3.2.1 Alat 29](#_Toc170923836)

[3.2.2 Bahan 30](#_Toc170923837)

[3.3 Diagram Alir Penelitian 30](#_Toc170923838)

[3.3.1 Identifikasi Masalah 31](#_Toc170923839)

[3.3.2 Studi Literatur 31](#_Toc170923840)

[3.3.3 Perencanaan 32](#_Toc170923841)

[3.3.4 Perancangan 33](#_Toc170923842)

[3.3.5 Pengkodean 34](#_Toc170923843)

[3.3.6 Pengujian 34](#_Toc170923844)

[3.3.7 Hasil Analisis 34](#_Toc170923845)

[BAB IV HASIL DAN PEMBAHASAN 36](#_Toc170923846)

[4.1 Hasil 36](#_Toc170923847)

[4.2 Pembahasan 48](#_Toc170923848)

[4.2.1 Perencanaan/*Planning* 48](#_Toc170923849)

[4.2.2 Perancangan/*Design* 51](#_Toc170923850)

[4.2.3 Pengkodean/ *Coding* 63](#_Toc170923851)

[4.2.4 Pengujian/ *Testing* 73](#_Toc170923852)

[4.2.5 Hasil Analisis 95](#_Toc170923853)

[BAB V KESIMPULAN DAN SARAN 99](#_Toc170923854)

[5.1 Kesimpulan 99](#_Toc170923855)

[5.2 Saran 99](#_Toc170923856)

[DAFTAR PUSTAKA 100](#_Toc170923857)

[LAMPIRAN 105](#_Toc170923858)

# DAFTAR GAMBAR

[Gambar 1.1 Wawancara Dengan Dokter 2](#_Toc170924161)

[Gambar 2.1 Contoh Diagram *Use Case* [23]……………………………………..17](#_Toc170924174)

[Gambar 2.2 Contoh *Activity* Diagram [24] 19](#_Toc170924175)

[Gambar 2.3 Contoh *Class* Diagram [25] 21](#_Toc170924176)

[Gambar 2.4 Metode *Extreme Programming* [32] 23](#_Toc170924177)

[Gambar 3.1 Diagram Alir Penelitian……………………………………………..31](#_Toc170924191)

[Gambar 4.1 Halaman *Login User*………………………………………………...36](#_Toc170924224)

[Gambar 4.2 Halaman *Dashboard* Dokter 37](#_Toc170924225)

[Gambar 4.3 Tampilan Awal Halaman Resep Obat 38](#_Toc170924226)

[Gambar 4.4 Tampilan Awal Pilih Pasien Baru 38](#_Toc170924227)

[Gambar 4.5 Tampilan Awal Pilih Pasien Lama 39](#_Toc170924228)

[Gambar 4.6 Konfirmasi Data Pasien Sudah Benar 39](#_Toc170924229)

[Gambar 4.7 Tampilan Kedua Halaman Resep Obat 40](#_Toc170924230)

[Gambar 4.8 Tambah Resep Pada Form Resep Obat 40](#_Toc170924231)

[Gambar 4.9 Konfirmasi Simpan Resep 41](#_Toc170924232)

[Gambar 4.10 Halaman *Dashboard* Apoteker 41](#_Toc170924233)

[Gambar 4.11 Halaman Stok Resep 42](#_Toc170924234)

[Gambar 4.12 Tampilan *Pop Up* Pada Tambah Resep 42](#_Toc170924235)

[Gambar 4.13 Tampilan *Pop Up* Pada Edit Resep 43](#_Toc170924236)

[Gambar 4.14 Konfirmasi Data Resep Dihapus 43](#_Toc170924237)

[Gambar 4.15 Halaman Riwayat Resep 44](#_Toc170924238)

[Gambar 4.16 Tampilan *Pop Up* Buat Resep 44](#_Toc170924239)

[Gambar 4.17 Hasil Cetak Resep Obat Pasien 45](#_Toc170924240)

[Gambar 4.18 Halaman *Dashboard* Role *Admin* 45](#_Toc170924241)

[Gambar 4.19 Halaman Pasien 46](#_Toc170924242)

[Gambar 4.20 Tampilan Edit Pasien 46](#_Toc170924243)

[Gambar 4.21 Halaman *Users* 47](#_Toc170924244)

[Gambar 4.22 Tampilan *Add* *User* 47](#_Toc170924245)

[Gambar 4.23 Tampilan Edit *User* 48](#_Toc170924246)

[Gambar 4.24 Konfirmasi Nonaktifkan *User* 48](#_Toc170924247)

[Gambar 4.25 *Use Case* Diagram 52](#_Toc170924248)

[Gambar 4.26 *Activity* Diagram Halaman *Login* Dokter 53](#_Toc170924249)

[Gambar 4.27 *Activity* Diagram Halaman Resep Obat Dokter 54](#_Toc170924250)

[Gambar 4.28 *Activity* Diagram Halaman *Dashboard* 55](#_Toc170924251)

[Gambar 4.29 *Activity* Diagram Halaman *Login* Apoteker 56](#_Toc170924252)

[Gambar 4.30 *Activity* Diagram Halaman Stok Resep 57](#_Toc170924253)

[Gambar 4.31 *Activity* Diagram Halaman Riwayat Resep Apoteker 58](#_Toc170924254)

[Gambar 4.32 *Activity* Diagram Halaman *Login Admin* 59](#_Toc170924255)

[Gambar 4.33 *Activity* Diagram Halaman Pasien *Admin* 60](#_Toc170924256)

[Gambar 4.34 *Activity* Diagram Halaman *Users Admin* 61](#_Toc170924257)

[Gambar 4.35 *Class* Diagram 62](#_Toc170924258)

[Gambar 4.36 Kode Fungsi Pada Halaman *Login* 63](#_Toc170924259)

[Gambar 4.37 Kode Fungsi Program Pada Halaman *Dashboard* 64](#_Toc170924260)

[Gambar 4.38 Kode Fungsi *Chart* Pada Halaman *Dashboard* 65](#_Toc170924261)

[Gambar 4.39 Kode Fungsi Pada Halaman Resep Obat 66](#_Toc170924262)

[Gambar 4.40 Kode Fungsi Program Tampilan Obat 67](#_Toc170924263)

[Gambar 4.41 Kode Fungsi Program Tampilan Resep Obat 67](#_Toc170924264)

[Gambar 4.42 Kode Program Halaman Riwayat Resep 68](#_Toc170924265)

[Gambar 4.43 Kode Fungsi Program Proses Resep 69](#_Toc170924266)

[Gambar 4.44 Kode Fungsi Program Halaman Pasien 70](#_Toc170924267)

[Gambar 4.45 Kode Fungsi Program Tampilan Edit 70](#_Toc170924268)

[Gambar 4.46 Kode Program Halaman *Users* 71](#_Toc170924269)

[Gambar 4.47 Kode Fungsi Program Edit *User* 72](#_Toc170924270)

[Gambar 4.48 Kode Fungsi Program Konfirmasi Status *User* 72](#_Toc170924271)

# DAFTAR TABEL

[Tabel 1.1 Waktu proses pembuatan resep sampai jadi obat secara manual 3](#_Toc170924472)

[Tabel 2.1 Ringkasan Penelitian Terdahulu………………………………………11](#_Toc170924484)

[Tabel 2.2 *Use-case* Diagram [23] 16](#_Toc170924485)

[Tabel 2.3 *Activity* Diagram [24] 18](#_Toc170924486)

[Tabel 2.4 *Class* Diagram [25] 19](#_Toc170924487)

[Tabel 2.5 Contoh Kebutuhan Pengguna [34] 24](#_Toc170924488)

[Tabel 2.6 Contoh *User Story* [33] 24](#_Toc170924489)

[Tabel 2.7 Contoh Pengujian Fitur Pada Halaman *Login* [36] 26](#_Toc170924490)

[Tabel 2.8 Contoh *Test Case* [38] 27](#_Toc170924491)

[Tabel 3.1 Spesifikasi *Hardware*………………………………………………….29](#_Toc170924495)

[Tabel 3.2 Daftar *Software* 30](#_Toc170924496)

[Tabel 3.3 Pertanyaan yang Diajukan 32](#_Toc170924497)

[Tabel 4.1 *User Stories* Iterasi Pertama……………………………………………50](#_Toc170924503)

[Tabel 4.2 *User Stories* Iterasi Kedua 51](#_Toc170924504)

[Tabel 4.3 *Blackbox Testing* Dokter Sebelum *login*. 73](#_Toc170924505)

[Tabel 4.4 *Blackbox Testing* Dokter Halaman *Login* 74](#_Toc170924506)

[Tabel 4.5 *Blackbox Testing* Dokter Halaman *Dashboard* 75](#_Toc170924507)

[Tabel 4.6 *Blackbox Testing* Dokter Halaman Resep Obat 77](#_Toc170924508)

[Tabel 4.7 *Blackbox Testing* Apoteker Sebelum *Login* 79](#_Toc170924509)

[Tabel 4.8 *Blackbox Testing* Apoteker Halaman *Login* 79](#_Toc170924510)

[Tabel 4.9 *Blackbox Testing* Apoteker Halaman *Dashboard* 81](#_Toc170924511)

[Tabel 4.10 *Blackbox Testing* Apoteker Halaman Stok Resep 82](#_Toc170924512)

[Tabel 4.11 *Blackbox Testing* Apoteker Halaman Riwayat Resep 84](#_Toc170924513)

[Tabel 4.12 *Blackbox Testing Admin* Sebelum *Login* 86](#_Toc170924514)

[Tabel 4.13 *Blackbox Testing Admin* Halaman *Login* 87](#_Toc170924515)

[Tabel 4.14 *Blackbox Testing Admin* Halaman *Dashoard* 89](#_Toc170924516)

[Tabel 4.15 *Blackbox Testing Admin* Halaman Pasien 90](#_Toc170924517)

[Tabel 4.16 *Blackbox Testing Admin* Halaman *Users* 92](#_Toc170924518)

[Tabel 4.17 Waktu proses pembuatan resep sampai jadi obat secara manual 95](#_Toc170924519)

[Tabel 4.18 Waktu Proses menggunakan sistem *electronic prescribing* 96](#_Toc170924520)

# DAFTAR LAMPIRAN

[Lampiran 1. Surat Permohonan Pengambilan Data Penelitian 105](#_Toc170924693)

[Lampiran 2. Daftar Pertanyaan Wawancara 106](#_Toc170924694)

[Lampiran 3. Hasil Wawancara 107](#_Toc170924695)

[Lampiran 4 Rancangan *User Interface* 109](#_Toc170924696)

[Lampiran 5 Penyerahan *Website* 110](#_Toc170924697)

# 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 Sultoni

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

## **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 prescbing* 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 *JavaScrip*t 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.

## **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.

## **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?

## **Batasan Masalah**

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

Sistem ini akan digunakan khusus di Klinik Asy-Syifa Desa Randudongkal.

Sistem ini tidak mencakup pelayanan medis lainnya, seperti jadwal dokter dan manajemen keuangan atau aspek non-medis lainnya.

Pengembangan sistem menggunakan metode *Extreme Programming*.

*Website* dikembangkan menggunakan Bahasa pemrograman *JavaScript* dan *database* *MySQL*.

## **Tujuan Penelitian**

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

## **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

## **Tinjauan Pustaka**

Pertama, penelitian dengan judul “Perancangan Sistem Informasi Resep Pulang Farmasi Rawat Inap Dengan Metode *E-Prescribing* Pada *RS XYZ*” oleh Tb. Dedifuadi, 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 menunjukan 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 Aplication 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 *E-*  *Presscribing* Pada *RS XYZ* [9] | Tb. Dedifuadi, Ahmad Surahmat, Rizki Fatullah | Mengembangkan sebuah sistem informasi untuk resep obat di farmasi rawat inap menggunakan pendekatan/metode *E-Prescribing* dengan bertujuan meningkatkan aksebilitas layanan kesehatan serta meningkatkan efisiensi dari pelayanan yang disediakan. | *Website* kali ini akan dibuat dengan tampilan yang lebih hidup, dengan menggunakan *framework react.js* |
| 2 | Penerapan Sistem *E-Prescribing* dan *Barcode system* di  Rumah Sakit Pertamina menggunakan *PowerBuilder*  dengan *Arsitektur Client - Server System* [10] | Try Viananda Nova M, S.Kom.,M.Kom, Rina Alfah, S.Kom.,M.Kom | Menerapkan sistem *E-Prescribing* dan *Barcode System* 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 *medician error*. | Penelitian sebelumnya menggunakan pengembangan sistem *Rapid Aplication Development (RAD)*, sedangkan penelitian kali ini menggunakan pengembangan sistem yaitu *extreme programming* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul** | **Penulis (tahun)** | **Hasil** | **Perbedaan** |
| 3 | Rancang Bangun Aplikasi Elektronik Resep yang Terhubung ke Kasir Menggunakan *Website* [11] | Helfy Susilawati, Tri Arif Wiharso, Teddy Mulyadi Hidayat | Membuat aplikasi elektronik resep yang terhubung kasir dikembangkan dengan metode *prototype,* 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. | Pengembangkan sistem sebelumnya menggunaan metode *prototype*, pada penelitian kali ini menggunakan metode *extreme programming* |
| 4 | Rancang Bangun Sistem *Electronic Prescribing*  Dokter dengan Menggunakan *Codeigniter* [12] | Dian Rakasiwi, Ria Arafiyah, Fariani Hermin Indiyah | Pembuatan sistem e*lectronic prescribing* dokter dengan menggunakan *framework Codeigniter* yang bertujuan meningkatkan ketersediaan layanan kesehatan dan meningkatkan standar serta efisiensi yang diberikan oleh Rumah Sakit. | Penelitian ini sebelumnya menggunakan *framework codeigniter*, sedangkan penelitian kali ini menggunakan bahasa *javascript* dengan *framework express.js* |

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

## **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.

### **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].

### ***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].

### **Sistem *Electronic Prescribing***

Sistem *electronic prescribing* atau sistem peresepan electronik 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].

### ***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].

### ***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 Languange)*. *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].

### **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].

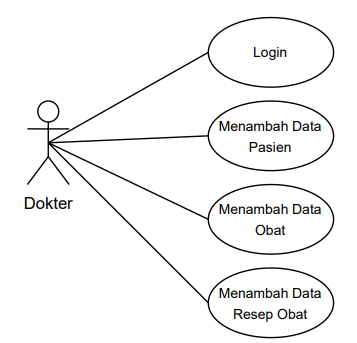
#### **Use Case Diagram**

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. |
|  | *Use case* | Deskripsi dari suatu fungsi atau tindakan yang dapat dilakukan oleh sistem. |
| A black rectangular frame with a white background  Description automatically generated | *System* | Menampilkan sistem yang sedang dikembangkan atau sedang berjalan. |
|  | *Actor* | Aktor merupakan entitas yang berhubungan secara aktif dengan sistem, seperti pengguna atau sistem lainnya. |
| A black text with a dotted line  Description automatically generated | *Include* | Hubungan antara *use case* yang memasukkan fungsionalitas dari *use case* lain*.* |
| A black text with a dotted line  Description automatically generated | *Extend* | Hubungan antara *use case* yang menambahkan fungsionalitas ke *use case lain.* |

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



**Gambar 2.1** Contoh Diagram Use Case [23]

#### **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 |
| A black circle with white border  Description automatically generated | *Initial Node* | Mengawali suatu aktivitas. |
|  | Status Akhir | Mengakhiri alur sistem. |
|  | Penggabungan | Penggabungan yang mana lebih dari satu  aktivitas akan digabungkan menjadi satu. |
|  | *Decision* | Menggambarkan terjadinya percabangan atau ada pilihan aktivitas  yang lebih dari satu. |
|  | *Swimlane* | *Swimlane* memperlihatkan tanggung jawab pelaksanaan aktvitas dalam suatu diagram  atau aktivitas yang terjadi. |

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

A diagram of a software company

Description automatically generated

**Gambar 2.2** Contoh Activity Diagram [24]

#### **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 sistemserta 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** |
| --- | --- | --- |
| A close up of words  Description automatically generated | Kelas | Entitas yang merepresentasikan objek dalam suatu sistem atau proses |
| A black circle with white lines  Description automatically generated  **nama\_interface** | *Interface* | 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 *multiplicity* |
|  | Generalisasi | Hubungan antar kelas bisa terjadi melalui asosiasi dan generalisasi-spesialisasi |
|  | *A ggregation* | Hubungan yang berarti antara kelas yang mencakup keseluruhan dan bagian-bagiannya *(whole-part)* |
|  | *Dependecy* | Ketergantungan antar kelas |

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

A white background with black text

Description automatically generated

**Gambar 2.3** Contoh Class Diagram [25]

### **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].

### ***JavaScript***

*JavaScript* adalah bahasa pemrograman *web* yang sering dipakai untukpengembangan 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].

### ***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].

### **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.

A diagram of software testing

Description automatically generated

**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 *website* dan mengisi *form login* | | Sistem mengalihkan ke halaman *dashboard* seteleh *login* |
| Pengguna dapat menambahkan data pasien | | Sistem menampilkan data pasien yang ada |
| Pengguna dapat mencetak resep | | Sistem akan kembali halaman *login* 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 |

* 1. *Design*

Pada tahap *design* dilakukanperencanaan 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.

* 1. *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 biasannya 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].

* 1. *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 menggunaan *blackbox testing.*

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

|  |  |  |
| --- | --- | --- |
| **No** | **Proses** | ***Expexted Output*** |
| 1 | Mengakses halaman *website* di *browser* | Menampilkan Halaman *Login* |
| 2 | Mengisi *form* *login* dan klik tombol *login* | Mengalihkan pada halaman *login* |

### ***Blackbox Testing***

*Blackbox Testing* adalah metode pegujian 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 pupular adalah *Equivalance 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 *Equivalance partitioning* pada pengujian *blackbox*.

**Tabel 2.8** Contoh Test Case [38]

| **No** | **Skenario** | **Ekspektasi Output** | **Hasil Pengujian** |
| --- | --- | --- | --- |
| 1 | Mengakses halaman *website* di *browser* | Menampilkan halaman *login* | Berhasil |
| 2 | Mengisi *form* *login* dan klik tombol *login* | Mengalihkan ke halaman *dashboard* | 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].

(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].

### **Rasio Efisiensi**

Pada perhitungan hasil efisiensi tahap pertama menggunkan 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.

(2. 2)

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

(2. 3)

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

(2. 4)

Setelah mendapatkan hasilanya, selanjutnya bisa menghitung rasio efisiensi untk melihat rasio efisiensi yang sesuai dengan persamaan 2.5

(2. 5)

Setelah menemukan rasio efisiensi maka selanjutnya menghitung persentase peningkatan efisensi, 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].

(2. 6)

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

# BAB III METODE PENELITIAN

## **Subjek Dan Objek Penelitian**

### **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.

### **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.

## **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:

### **Alat**

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

**Tabel 3.1** Spesifikasi Hardware

|  |  |  |
| --- | --- | --- |
| **No** | **Nama *Hardware*** | **Spesifikasi** |
| 1 | *Processor* | *Intel Pentium Gold 5405U* |
| 2 | *RAM* | *8 GB* |
| 3 | *ROM* | *1TB HDD, 512GB SSD* |

**Tabel 3.2** Daftar Software

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

### **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*.

## **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.

A diagram of a process

Description automatically generated

**Gambar 3.1** Diagram Alir Penelitian

### **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.

### **Studi Literatur**

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

### **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 *electronic prescribing* 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 pengerjaan fitur yang dibutuhkan.

### **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.

1. ***Activity* Diagram**

*Activity* diagram dapat mengambarkan 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 berkhir. Diagram ini digunakan untuk mempermudah perancangan sistem *website electronic prescribing* dokter di Klinik Asy-Syifa Desa Randudongkal.

1. ***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.

### **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 yangmenggunakan *MySQL.*

### **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.

### **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

## **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:

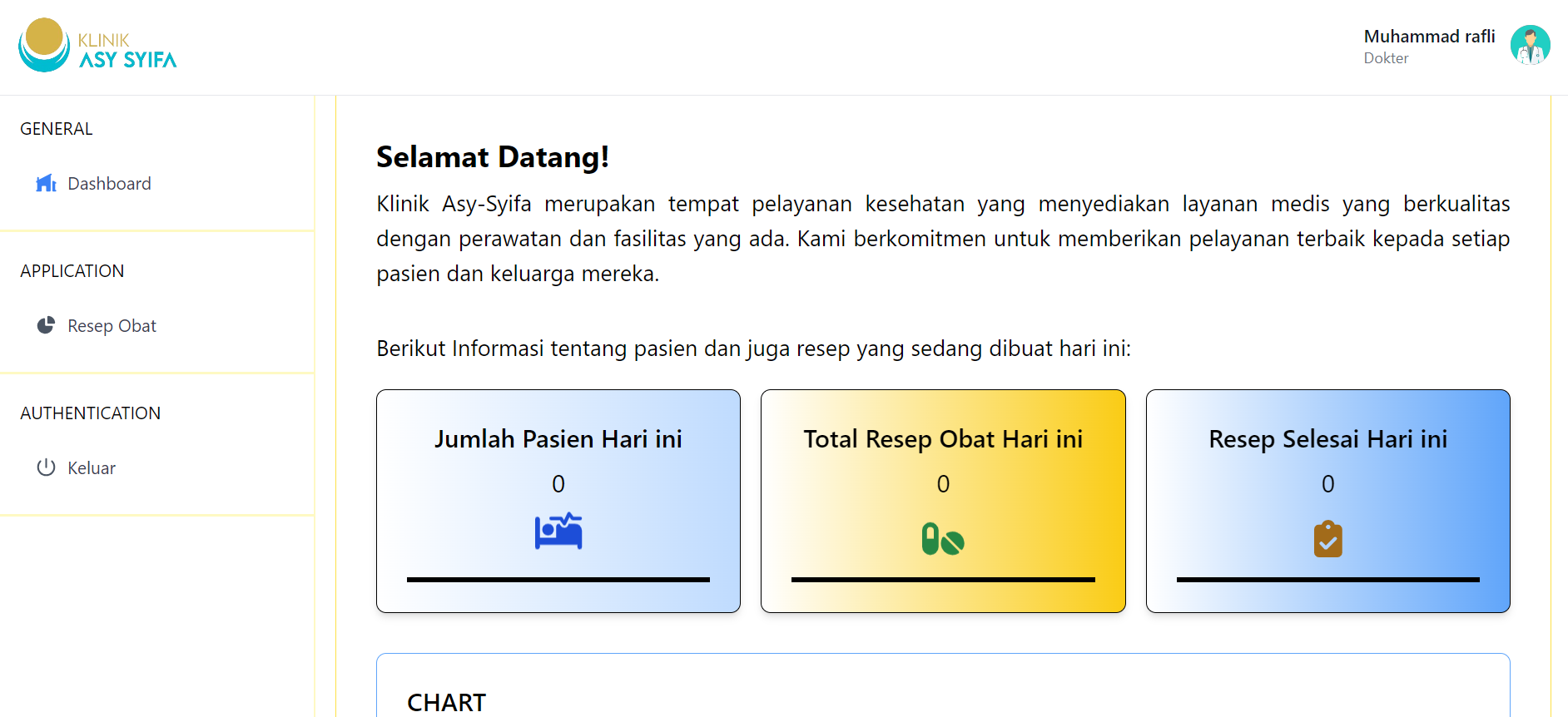
A screenshot of a login page

Description automatically generatedHalaman *Login*

**Gambar 4.1** Halaman Login User

Pada gambar 4.1 merupakan halaman untuk *login* semua role yaitu dokter, apoteker dan *admin*. Sistem akan menampilan *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.

Halaman *Dashboard* untuk role dokter



A graph with blue lines and black text

Description automatically generated

**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.

Halaman Resep Obat Dokter

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**Gambar 4.8** Tambah Resep Pada Form Resep Obat

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

A screenshot of a computer

Description automatically generated

**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.

Halaman *Dashboard* untuk role apoteker

A screenshot of a computer

Description automatically generated

**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.

Halaman Stok Resep

A screenshot of a computer

Description automatically generated

**Gambar 4.11** Halaman Stok Resep

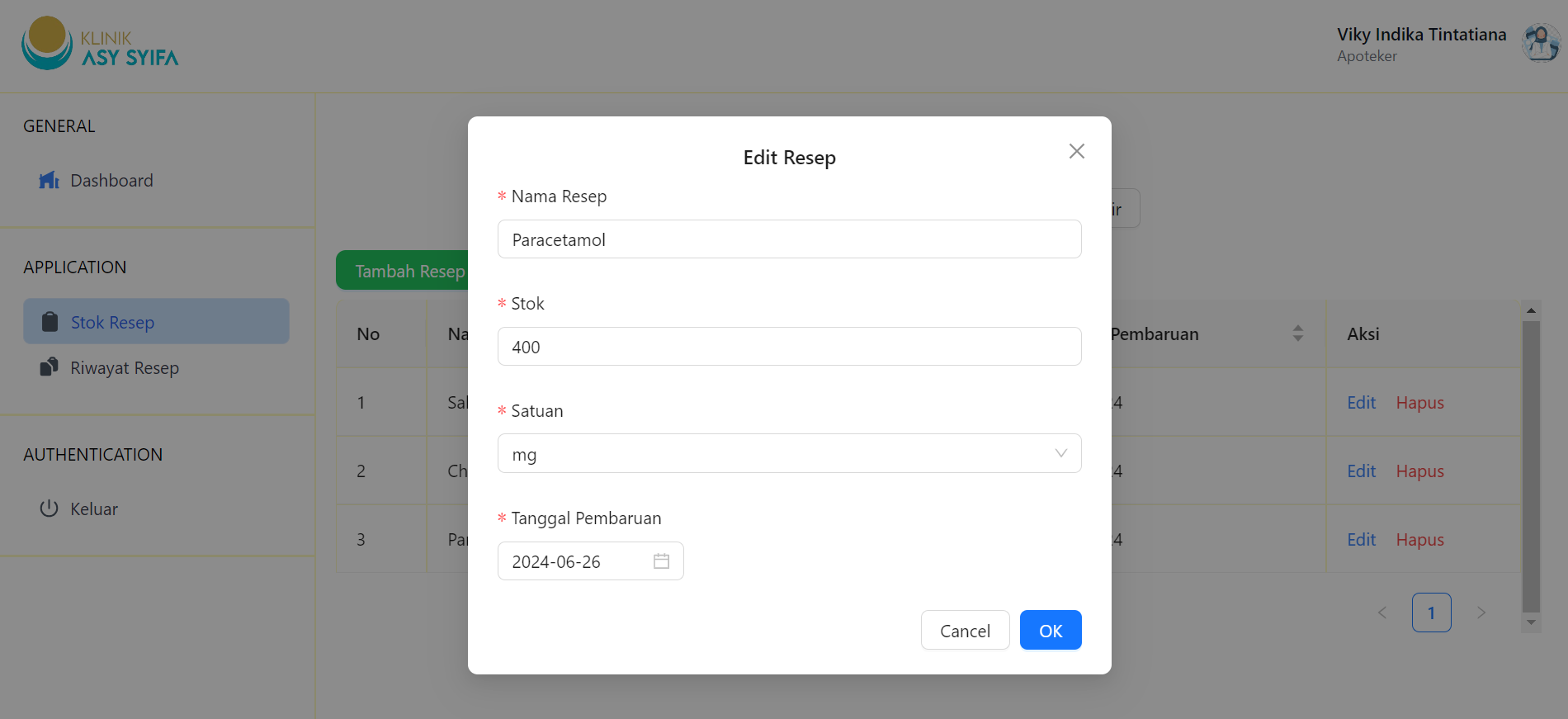
Pada gambar 4.11 merupakan halaman stok resep nantinya apoteker bisa melihat data resep dan melihat stok yang ada pada, nantinya apoteker bisa menambah data resep, mengedit dan menghapus data resep.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

Halaman Riwayat Resep

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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”.

A screenshot of a computer

Description automatically generated

**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.

Halaman *Dashboard* untuk role *admin*

A screenshot of a computer

Description automatically generated

**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.

Halaman Pasien

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

Halaman *Users*

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a login form

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

## **Pembahasan**

### **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:

#### **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.

#### **Tampilan Riwayat Resep**

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

#### **Fitur Pendukung Aplikasi**

Selain halaman input data untuk pasien dan resep obat, dari pihak klinik juga membutuhkan tampilan *dashboard* yang menampikan 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 *dashboard* |

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

* 1. Fitur *login* untuk membatasi akses penggunaan aplikasi yang hanya akan digunakan oleh pihak klinik Asy-Syifa Desa Randudongkal.
  2. Fitur resep obat untuk menambah atau mencari data pasien, menambah data obat, dan menambah data resep obat.
  3. Fitur riwayat resep untuk melihat data pasien, data obat dan data resep obat yang ditambahkan oleh dokter.
  4. Fitur *dashboard* sebagai pemantau pasien yang datang dan resep obat yang sudah dibuat.
  5. 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 terdapatt fitur baru yang dibutuhkan oleh dokter dan apoteker di klinik Asy-Syifa yaitu pengelolalan stok obat dan juga vaidasi 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

### **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.

#### **Use Case Diagram**

A diagram of a data flow

Description automatically generated

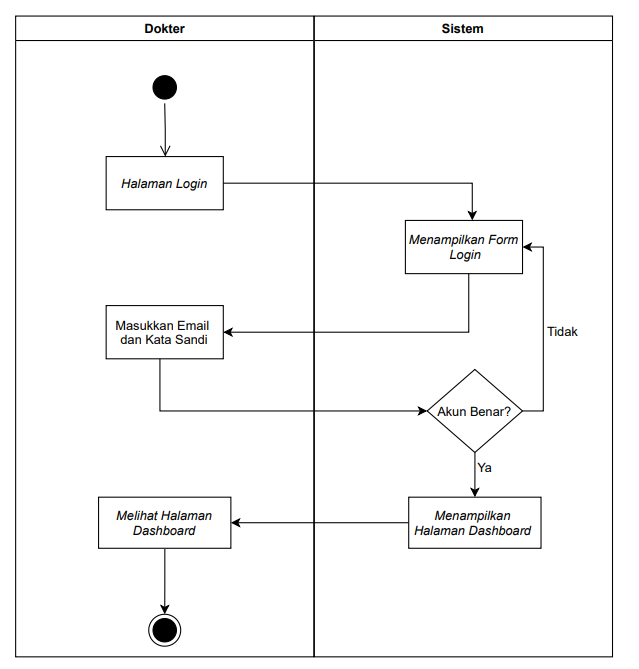
**Gambar 4.25** Use Case Diagram

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.

#### **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**

A diagram of a data flow

Description automatically generated

**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*.

**A diagram of a system

Description automatically generatedHalaman *Dashboard***

**Gambar 4.28** Activity Diagram Halaman Dashboard

Pada gaambar 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**

**A diagram of a software system

Description automatically generated**

**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 ditemmukan. Hal ini memastikan bahwa hanya pengguna yang sah dan berwenang yang dapat mengakses informasi sensitif dan operasional dalam sistem.

**Halaman Stok Resep**

**A diagram of a flowchart

Description automatically generated**

**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.

A diagram of a process flow

Description automatically generated**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.

A diagram of a software application

Description automatically generated with medium confidence**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**

**A diagram of a data flow

Description automatically generated**

**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.

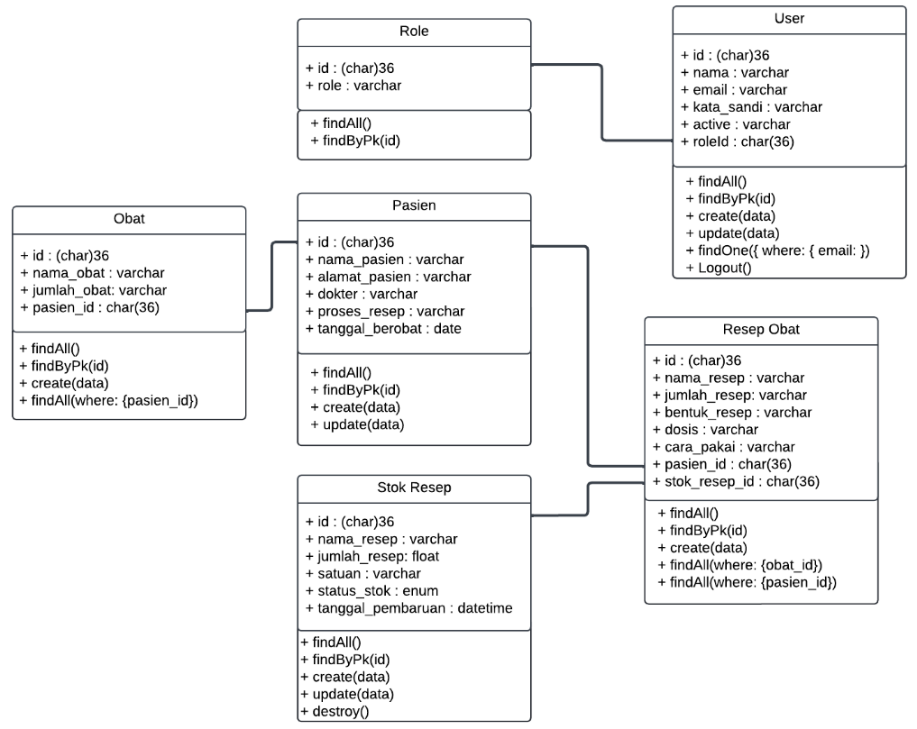
**A diagram of a software flowchart

Description automatically generatedHalaman *Users***

**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*.

#### **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.

### **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:

#### **A screen shot of a computer screen Description automatically generatedHalaman Login**

**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*.

#### **Halaman Dashboard**

**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.



**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 dipiih yang didapatkan dari *API* yang sudah disediakan.

#### **Halaman Resep Obat**



**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.



**Gambar 4.40** Kode Fungsi Program Tampilan Obat

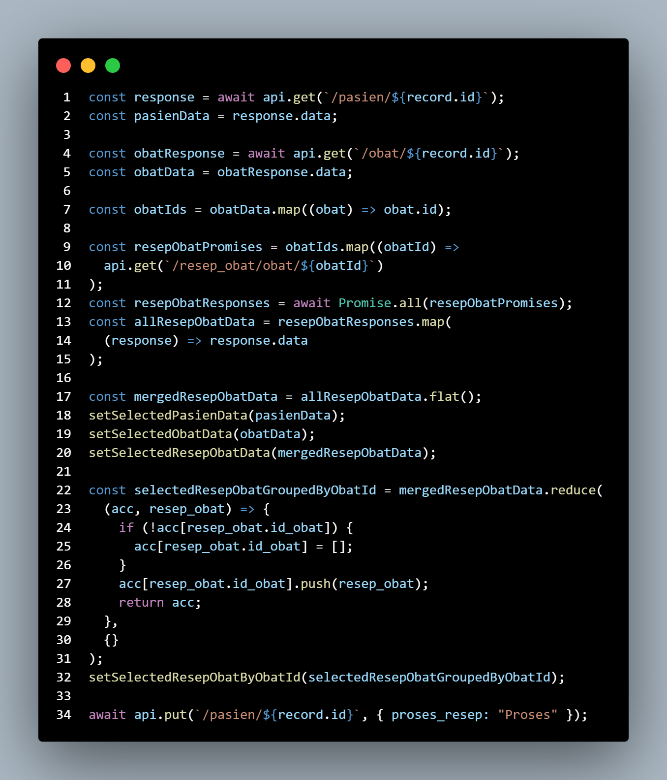
Pada gambar 4.40 merupakan kode program dari tampilan obat disaat dokter sudah menginputkan data obat maka program tersebut bekerja melalui *API* yang sudah tersedia.



**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.

#### **Halaman Riwayat Resep**



**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*.



**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.

#### **Halaman Pasien**

**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* mendapatakan data dari *database* maka sistem menampilkan semua data dari semua pasien.



**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.

#### **Halaman Users**



**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*.



**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.



**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.*

### **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:

#### **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.

* + - * 1. 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 *Login* | Sistem akan menampilkan halaman *login* | Berhasil |
| DR02 | Mengakses Halaman *Dashboard* | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |
| DR03 | Mengakses Halaman Resep Obat | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |

* + - * 1. 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 *Login* | Sistem akan menampilkan halaman *login* dengan *form* email dan kata sandi | Berhasil |
| DR05 | Mengisi email [dokter@gmail.com](mailto:dokter@gmail.com) dengan kata sandi “dokter” dan menekan tombol *login* | Sistem akan memproses masukan dari *user* dan mengarahkan ke halaman *dashboard* | Berhasil |
| DR06 | Mengisi email dan kata sandi yang tidak terdaftar | Sistem akan menampilkan pesan *error* “User tidak ditemukan” | Berhasil |
| DR07 | Mengisi email yang terdaftar dan mengisi kata sandi yang salah | Sistem akan menampilkan pesan *error*” Kata Sandi Salah” | Berhasil |
| DR08 | Hanya mengisikan email dan mengosongkan kata sandi | Sistem akan menampilkan pesan *error*” Harap isi bidang ini” dibawah input kata sandi | Berhasil |
| DR09 | Hanya Mengisikan kata sandi dan mengosongkan email | Sistem akan menampilkan pesan error” Harap isi bidang ini” dibawah input email | Berhasil |

* + - * 1. 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 *Dashboard* | 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 *patient chart* | Berhasil |
| DR11 | Meletakkan kursor pada titik *chart* 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 |
| DR12 | Menekan *icon* untuk *zoom in* pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih tinggi dari tampilan normal | Berhasil |
| DR13 | Menekan *icon* untuk *zoom out* pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih rendah dan kembali normal setelah di *zoom* | Berhasil |
| DR14 | Menekan titik pada grafik pasien satu minggu pada data *chart patient* | Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih | Berhasil |

* + - * 1. 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 *form* data pasien | Berhasil |
| DR16 | Mengisi seluruh *input* *form* 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 *form* tampilan kedua yaitu *form* obat | Berhasil |
| DR18 | Mengosongkan atau mengisi salah satu *input* *form* data pasien | Sistem akan menampilkan pesan *error* “Harap isi bidang ini” dibawah input yang belum terisi | Berhasil |
| DR19 | Mengisi seluruh input *form* data obat dan menekan selanjutnya | Sistem akan menampilkan konfirmasi “Data Obat Sudah Benar?” | Berhasil |
| DR20 | Menekan tombol sudah setelah ada konfirmasi “Data Obat sudah Benar?” | Sistem akan menampilkan *form* tampilan ketiga yaitu *form* resep obat | Berhasil |
| DR21 | Mengosongkan atau mengisi salah satu *input* *form* data resep obat | Sistem akan menampilkan pesan *error* “Harap isi semua bidang yang diperlukan” diatas tombol selanjutnya | Berhasil |
| DR22 | Mengisi seluruh *input* *form* 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 *form* awal di halaman resep obat | Berhasil |

#### **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.

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 *Login* | Sistem akan menampilkan halaman *login* | Berhasil |
| AP02 | Mengakses Halaman *Dashboard* | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |
| AP03 | Mengakses Halaman Riwayat Resep | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |

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 *Login* | Sistem akan menampilkan halaman *login* dengan *form* email dan kata sandi | Berhasil |
| AP05 | Mengisi email [apoteker@gmail.com](mailto:apoteker@gmail.com) dengan kata sandi “apoteker” dan menekan tombol *login* | Sistem akan memproses masukan dari *user* dan mengarahkan ke halaman *dashboard* | Berhasil |
| AP06 | Mengisi email dan kata sandi yang tidak terdaftar | Sistem akan menampilkan pesan *error* “User tidak ditemukan” | Berhasil |
| AP07 | Mengisi email yang terdaftar dan mengisi kata sandi yang salah | Sistem akan menampilkan pesan *error*” Kata Sandi Salah” | Berhasil |
| AP08 | Hanya mengisikan email dan mengosongkan kata sandi | Sistem akan menampilkan pesan *error*” Harap isi bidang ini” dibawah input kata sandi | Berhasil |
| AP09 | Hanya Mengisikan kata sandi dan mengosongkan email | Sistem akan menampilkan pesan error” Harap isi bidang ini” dibawah input email | Berhasil |

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 *Dashboard* | 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 patient *chart* | Berhasil |
| AP11 | Meletakkan kursor pada titik *chart* 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 *icon* untuk *zoom* in pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih tinggi dari tampilan normal | Berhasil |
| AP13 | Menekan *icon* untuk *zoom* *out* pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih rendah dan kembali normal setelah di *zoom* | Berhasil |
| AP14 | Menekan titik pada grafik pasien satu minggu pada data *patient chart* | Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih | Berhasil |

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 |
| AP16 | Menekan *icon filter* 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 *pop up model* dengan menampilkan *form* untuk tambah resep | Berhasil |
| AP20 | Mengisi form pada tambah resep dan menekan tombol OK | Sistem akan menutup *pop up* dan terdapat notifikasi “Stok resep berhasil ditambahkan” | Berhasil |
| AP21 | Menekan tombol Edit pada data resep yang dipilih | Sistem akan menampilkan *pop up model* dengan menampilkan *form* yang berisi data resep yang dipilih | Berhasil |
| AP22 | Mengedit pada *form* edit resep | Sistem akan menutup *pop up* 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 |

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 resep dengan tabel data pasien dan terdapat proses resep | Berhasil |
| AP26 | Menekan *icon search* sebelah nama pasien | Sistem akan menampilkan pencarian nama dibawah nama pasien | Berhasil |
| AP27 | Mengisi nama “Dika” dan menekan tombol *search* | 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 *pop up model* dengan data resep obat pasien, dan pada proses resep tabel pasien berubah menjadi “Proses” | Berhasil |
| AP31 | Menekan tombol Selesai Dibuat pada *pop up model* | Sistem akan menutup *pop up* *model* dan proses resep pada tabel pasien berubah menjadi “Selesai” | Berhasil |
| AP32 | Menekan tombol Cetak Resep pada *pop up model* | 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 |

#### **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.

* 1. *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 *Login* | Sistem akan menampilkan halaman *login* | Berhasil |
| AD02 | Mengakses Halaman *Dashboard* | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |
| AD03 | Mengakses Halaman Pasien | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |
| AD04 | Mengakses Halaman *Users* | Sistem akan menolak akses dan mengarahkan ke halaman *login* | Berhasil |

* 1. *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 *Login* | Sistem akan menampilkan halaman *login* dengan *form* email dan kata sandi | Berhasil |
| AD06 | Mengisi email [admin@gmail.com](mailto:admin@gmail.com) dengan kata sandi “admin” dan menekan tombol *login* | Sistem akan memproses masukan dari *user* dan mengarahkan ke halaman *dashboard* | Berhasil |
| AD07 | Mengisi email dan kata sandi yang tidak terdaftar | Sistem akan menampilkan pesan *error* “User tidak ditemukan” | Berhasil |
| AD08 | Mengisi email yang terdaftar dan mengisi kata sandi yang salah | Sistem akan menampilkan pesan *error*” Kata Sandi Salah” | Berhasil |
| AD09 | Hanya mengisikan email dan mengosongkan kata sandi | Sistem akan menampilkan pesan *error*” Harap isi bidang ini” dibawah input kata sandi | Berhasil |
| AD10 | Hanya Mengisikan kata sandi dan mengosongkan email | Sistem akan menampilkan pesan error” Harap isi bidang ini” dibawah input email | Berhasil |

* 1. *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 Dashoard

| **ID** | **Skenario Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** |
| --- | --- | --- | --- |
| AD11 | Mengakses Halaman *Dashboard* | 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 *patient chart* | Berhasil |
| DR12 | Meletakkan kursor pada titik *chart* 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 *zoom* in pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih tinggi dari tampilan normal | Berhasil |
| AD14 | Menekan icon untuk *zoom* *out* pada *patient chart* | Sistem akan menampilkan data *chart* dengan tingkat perbesaran yang lebih rendah dan kembali normal setelah di *zoom* | Berhasil |
| AD15 | Menekan titik pada grafik pasien satu minggu pada data *patient chart* | Sistem akan menampilkan data pasien dengan grafik yang baru dalam satu minggu yang dipilih | Berhasil |

* 1. *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 *icon* search sebelah nama pasien | Sistem akan menampilkan pencarian nama dibawah nama pasien | Berhasil |
| AD18 | Mengisi nama “Dika” dan menekan tombol search | Sistem akan menampilkan data pasien Dika | Berhasil |
| AD19 | Menekan tombol reset pada *search* 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 *error* “Harap isi bidang ini.” Pada inputan kosong | Berhasil |

* 1. *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 *Users* | Sistem akan menampilkan halaman *Users* dengan tampilan tabel untuk semua *user* | Berhasil |
| AD24 | Menekan *icon* untuk *sort accending* atau *descending* | Sistem akan menampilkan nama yang teratas jadi paling bawah dan nama terbawah menjadi paling atas | Berhasil |
| AD25 | Menekan *icon* untuk filter role menjadi dokter saja | Sistem akan menampilkan data pengguna yaitu dokter saja yang terlihat | Berhasil |
| AD26 | Menekan *icon* untuk filter *Active* menjadi *Inactive* saja | Sistem akan menampilkan data pengguna yitu hanya yang *inactive* 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 yang dipilih menjadi seperti semula | Berhasil |
| AD29 | Menekan tombol *Add User* | Sistem akan menampilkan halaman baru *form* *add user* | Berhasil |
| AD30 | Input semua isi *form add user* dan menetakan simpan | Sistem akan menampilkan tampilan awal ke data semua pengguna | Berhasil |
| AD31 | Mengisi salah satu atau mengosongkan semua *form add user* | Sistem menampilkan pesan *error* “Harap isi bidang ini.” Pada inputan kosong | Berhasil |
| AD32 | Menekan tombol Edit pada aksi | Sistem akan menampilkan halaman baru untuk *form* edit user | Berhasil |
| AD33 | Mengisi semua data *user* dan menekan Simpan | Sistem akan kembali ke data semua pengguna pada halaman *users* | Berhasil |
| AD34 | Mengisi salah satu atau mengosongkan semua *form* edit pasien | Sistem menampilkan pesan error “Harap isi bidang ini.” Pada inputan kosong | Berhasil |
| 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 mengaktfikan kembali pengguna” | Berhasil |

#### **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.

### **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 precribing untuk menentukan perbedaan rata rata waktu, presentase 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 penghemeatan 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 |
|  | **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, presentase perbedaan, dan rasio efisiensi, berikut ini adalah perhitungan yang dilakukan oleh penulis:

1. Rata-rata waktu sebelum dilakukan implementasi

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

1. Rata-rata waktu setelah dilakukan implementasi

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

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

1. 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.

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

1. Rasio persentase perbedaan

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

1. Rasio efisiensi

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

Setalah menghitung rasio efisiensi, selanjutnya menghitung persentase peningkatan efisiensi.

1. Persentase peningkatan efisiensi

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

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 menunjukan 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 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

## **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.

## **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 Kholifatul Ulum and I. Laily Hilmi, “Review Artikel : Implementasi dan Evaluasi Peresepan Elektronik Dalam Upaya Menurunkan Kesalahan Pengobatan,” *Journal of Pharmaceutical and Sciences*, vol. 6, no. 1, pp. 191–198, 2023, [Online]. Available: https://www.journal-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/ejhpharm-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/JKIKT*, 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: 10.52362/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 Armiady, “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.

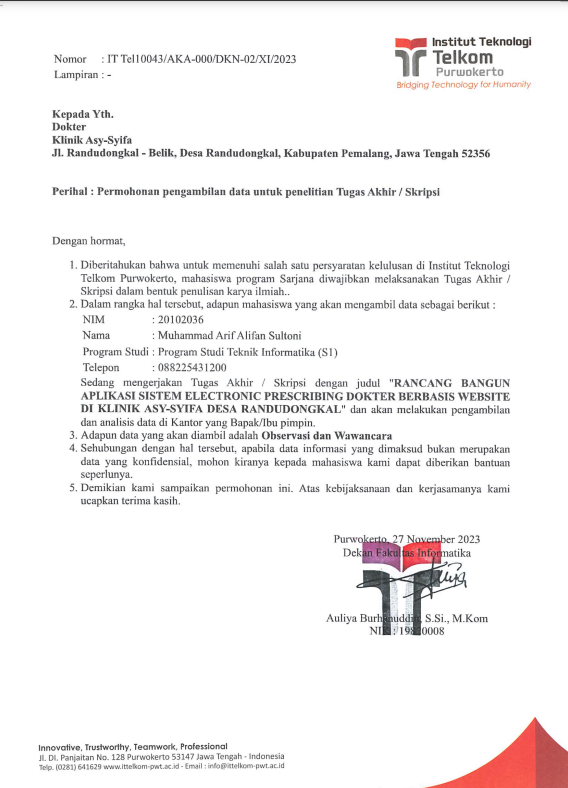
[38] E. Novalia and A. Voutama, “Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah,” 2022.

[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**

****

**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 ditulisakn dokter untuk dibuat obat sesuai instrukti 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.

1. Waktu yang di perlukan untuk pembuatan resep obat sampai menjadi obat itu kurang lebih 20 menit.
2. 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.
3. 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.
4. 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.
5. 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**

