BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pelayanan gawat darurat adalah tindakan medis yang dibutuhkan oleh pasien gawat darurat dalam waktu segera untuk menyelamatkan nyawa dan pencegahan kecacatan. Namun pendataan pasien IGD pada Rumah Sakit Fatima masih diselenggara manual dimana setiap pasien yang masuk ke IGD didata dalam *logbook* atau buku yang mana kemudian diarsipkan. Ini menyebabkan banyaknya permasalahan yang berlaku seperti salah pendataan, pencarian pasien dalam arti kata lain pasien pernah memasuki IGD Rumah Sakit Fatima sebelumnya ataupun baru pertama kali pada laporan bulanan ke pimpinan direksi. Staf terpaksa menelusuri buku atau *logbook* lama untuk melihat data pasien dan ini memakan waktu baik untuk karyawan maupun untuk pasien yang memerlukan perawatan segera. Oleh karena itu adanya sebuah sistem pendataan komputerisasi mampu untuk mengefisienkan waktu dan membuat kinerja lebih efektif.

Perancangan sebuah sistem pendataan IGD pada rumah sakit ini sangatlah penting untuk meningkatkan efektifitas dan efisiensi baik untuk karyawan dan juga pasien yang datang berkunjung ke rumah sakit serta mengurangi tingkat kesalahan *(error)* yang terjadi pada penginputan data pasien dan pencarian data. Berdasarkan masalah yang dihadapi oleh pihak Rumah Sakit Fatima, penulis mengusulkan sebuah sistem pendataan untuk rumah sakit ini yang mana memudahkan karyawan yang menggunakan serta memberikan laporan yang lebih

lengkap mengenai pasien yang datang berkunjung. Disini penulis mengusulkan judul "**Penerapan Sistem Pendataan IGD (Instalasi Gawat Darurat) Pada Rumah Sakit Fatima Berbasis Client Server**".

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat diambilkan perumusan masalah yaitu :

1. Bagaimana merancang sebuah sistem pendataan pada IGD Rumah Sakit Fatima yang mudah digunakan untuk karyawan pendataan yang bertugas?
2. Bagaimana merancang sebuah sistem pendataan yang memberikan laporan secara otomatis berdasarkan pada apa yang diperlukan oleh pihak direksi Rumah Sakit Fatima efektif dan efisien?
3. Bagaimana merancang sebuah sistem pendataan yang mana staf yang bertugas juga dapat melihat pasien yang dirawat maupun yang telah dirawat sebelumnya?

C. Tujuan Masalah

Berdasarkan latar belakang dan tujuan masalah yang telah diuraikan maka dapat disimpulkan tujuan penelitian adalah seperti berikut:

1. Merancang sebuah sistem pendataan pada IGD (Instalasi Gawat Darurat) Rumah Sakit Fatima yang mudah digunakan untuk karyawan pendataan yang bertugas.
2. Merancang sebuah sistem pendataan yang memberikan laporan secara otomatis berdasarkan apa yang diperlukan oleh pihak rumah sakit efektif dan efisien.
3. Merancang sebuah sistem pendataan yang mana doktor yang bertugas juga dapat melihat pasien yang dirawat maupun yang telah dirawat sebelumnya.

D. Batasan Masalah

Batasan masalah diperlukan agar penelitian yang dilakukan dapat lebih terfokus, maka penulis membatasi masalah pokok bahasan sebagai berikut:

1. Lokasi penelitian dilakukan di IGD Rumah Sakit Fatima.
2. Sistem yang dibuat dibatasi kepada sistem pendataan pasien yang berkunjung dan rawat inap pada IGD Rumah Sakit Fatima.
3. Laporan yang dibuat hanya berdasarkan informasi pendataan yang diinput oleh sistem pendataan pasien saja.

E. Manfaat Penelitian

Beberapa manfaat dari penelitian ini sebagai berikut :

1. Aspek Akademis

Sebagai langkah awal bagi peneliti untuk dapat mengembangkan teori-teori yang didapat selama ini di bangku perkuliahan khususnya dalam mengenai sistem berbasis *client server.*

2. Aspek Praktis

a. Menjadi pertimbangan dalam pembuatan sistem yang praktis dalam pendataan bagi IGD (Instalasi Gawat Darurat) Rumah Sakit Fatima.
b. Menjadi saran masukan informasi dan perbaikan sistem pendataan bagi IGD (Instalasi Gawat Darurat) Rumah Sakit Fatima.
c. Menginspirasi bagi mahasiswa peneliti lebih lanjut untuk menerapkan ilmu pengetahuan dan mengembangkan pengalaman yang telah diterima.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

"Sistem Informasi Pengolahan Data Pasien Rawat Inap Berbasis *SMS Gateway*" oleh (Haryanto & Priyatmoko, 2015). Implementasi sistem yang dirancang berhasil mempermudah pengolahan data pasien dan menginformasikan data pasien kepada keluarga yang bertanggung jawab. Sistem ini menggunakan *UML (Unified Modeling Language), Microsoft Visual Basic 6.0* sebagai bahasa pemrograman, databasenya menggunakan *Microsoft SQL Server 2000* dan laporannya dibuat menggunakan *Crystal Report 8.5*. Sistem ini memiliki fasilitas-fasilitas yang berhubungan medis seperti pendataan dokter, pendataan perawat, pendataan ruangan, pendataan pasien, pendaftaran rawat jalan, pendaftaran rawat inap, SMS pihak keluarga pasien, pendataan obat, pembayaran serta laporan tersaji cepat dan akurat. Perbedaan antara penelitian ini penelitian penulis terletak pada pemilihan bahasa program dan *database* yang digunakan karena penulis menggunakan bahasa pemrograman *Flutter* untuk *frontend* dan *PHP* untuk *backend* dan *MYSQL* sebagai *database* sistem.

"Perancangan *Database Sistem Informasi Rumah Sakit Bhayangkara Palembang Pada Citra Lidah*" oleh (Erwin, Zellah, Facrurrozi, & Primanita, 2016). Penelitian ini menyimpulkan bahwa telah menghasilkan *database* aplikasi analisis penyakit melalui citra lidah yang dapat membantu aplikasi dalam penyimpanan data hasil pemeriksaan penyakit berdasarkan citra lidah pada pasien.

Perbedaan penelitian ini yang dilakukan oleh penulis adalah obyek penelitian dimana penelitian sebelumnya terfokus kepada satu penyakit manakala penulis terfokus kepada sistem pendataan pada *IGD* (Instalasi Gawat Darurat) pada Rumah Sakit Fatima.

B. Rumah Sakit

Rumah sakit adalah institusi perawatan kesehatan khusus yang dilayani oleh dokter, perawat, dan profesional perawatan kesehatan lainnya. Tanggung jawab dan fungsi rumah sakit adalah:

1. Perumusan kebijakan teknis di bidang pelayanan kesehatan
2. Penyelenggaraan kegiatan pelayanan keperawatan sesuai dengan standar rumah sakit
3. Penyelenggaraan kegiatan pelayanan pengobatan medis dan pemulihan kesehatan
4. Pemeliharaan dan peningkatan kesehataan perorangan melalui pelayanan kesehatan medis
5. Penyelenggaraan kegiatan pelayanan kesehatan medis tradisional dan pelayanan medis di bidang pengobatan khusus
6. Melaksanakan pelayanan rujukan kesehatan, pelayanan penunjang medis dan non medis
7. Penyelenggaraan kegiatan pelayanan rawat jalan atau rawat darurat dan rawat inap (observasi) serta penyuluhan kesehatan sosial
8. Membantu kegiatan penelitian dan pengembangan kesehatan

9. Penyelenggaraan pengelolaan keuangan dan akuntansi serta perencanaan dan pengembangan sistem manajemen rumah sakit
10. Penyelenggaraan pelayanan rawat inap, administratif, pendidikan para medis, pendidikan tenaga medis umum, pendidikan tenaga medis spesialis

Tanggung jawab pelayanan tersebut terkait dengan jenis dan kelas rumah sakit umum dan rumah sakit khusus, yang terdiri dari kelas A, B, C, dan D. Sebagai unit pelaksana teknis daerah yang berbentuk kelembagaan. Perubahan peringkat rumah sakit dapat terjadi karena berkurangnya efisiensi rumah sakit sebagaimana ditetapkan oleh Kementerian Kesehatan RI.

C. *IGD* (Instalasi Gawat Darurat)

IGD (Instalasi Gawat Darurat) adalah unit dalam rumah sakit yang menyediakan penanganan awal kepada pasien sesuai dengan tingkat kegawatan. Pelayanan pada instalasi gawat darurat adalah pelayanan medis pada pasien dengan kelainan medis atau kondisi medis yang memerlukan perawatan secepat mungkin untuk menyembukan atau menyelamatkan nyawa pasien.

D. Sistem

Sistem adalah unit komponen atau elemen-elemen yang dihubungkan bersama untuk memfasilitasi aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk menggambarkan satu set objek yang berinteraksi dari mana model matematika dapat dibuat.

E. *Electron*

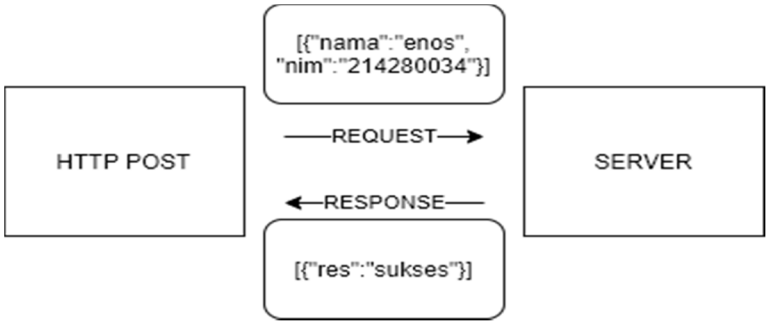
Electron merupakan *framework* yang digunakan untuk membuat aplikasi *desktop* dimana aplikasi yang dihasilkan dapat berjalan di sistem operasi apapun

seperti *Windows*, *Linux*, dan *MacOS*. *Elektron* dibuat oleh *Cheng Zhao* dan kemudian dibangun, dikembangkan, dan dikendalikan di *GitHub*. *Elektron* pertama kali dirilis pada 15 Juli 2013.

F. *Javascript*

Javascript adalah galat pada bahasa pemrograman *web* yang berfungsi untuk menciptakan laman *website* agar lebih dinamis dan interaktif. Bahasa pemrograman *javascript* itu sendiri berjalan pada perangkat lunak *website* atau suatu *situs*, bukan berjalan pada *server*, jadi tidak sinkron berdasarkan menggunakan *PHP*, *HTML*, dan *CSS*. *HTML* berfungsi untuk mengatur tampilan halaman konten dan *CSS* berfungsi untuk mengelola layout, jadi ketiganya tidaklah sinkron. Bahasa pemrograman biasa di gambarkan seperti *HTML* sebagai bangunan rumah, *CSS* sebagai perabotan di dalamnya, dan *javascript* sebagai lampu, jendela, dan pintu-pintu. Tetapi sekarang, *javascript* pula mulai beralih pada *cross platform* yaitu pelaksanaan *multi platform* misalnya ke sistem operasi *Android* & *IOS* selain daripada *WEB* misalnya sebelumya. Perkembangan *javascript* pula memungkinkan programmer buat menciptakan pelaksanaan *desktop*.

G. *JSON (Javascript Object Notation)*



Gambar 2.1 Penggunaan JSON REST

Gambar 2.1 adalah penggunaan *JSON REST* method *POST*. Data yang dikirimkan oleh *client* adalah nama : enos dan nim : 214280034. Server kemudian menjalankan fungsi '/service/mahasiswa' untuk memasukkan data mahasiswa ke dalam database. Fungsi tersebut kemudian akan me-returnkan data res : sukses.

H. *Node JS*

Node.js merupakan platform perangkat lunak untuk menjalankan kode *JavaScrip* pada sisi server dan aplikasi jaringan. Ia bertugas untuk mengeksekusi kode *JavaScrip* sebelum halaman website ditampilkan di browser. *Node.js* ditulis dalam *JavaScript* dan berjalan di *Windows, Mac OS X*, dan *Linux* tanpa perubahan kode. *Node.js* hadir pustaka *HTTP*nya sendiri yang memungkinkan anda menjalankan pemuat web tanpa program *web* seperti *Apache* atau *Lighttpd.*

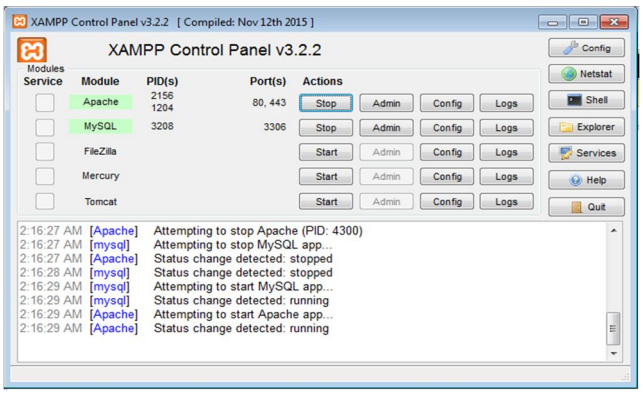
I. *MySQL*

MySQL adalah sistem manajemen basis data relasional (RDBMS) berbasis *SQL (Structured Query Language)* yang *multithread* dan *multi-user*. *MySQL* adalah sebuah perangkat lunak yang bersifat open-source dengan model client-server yang digunakan untuk membuat sebuah database. Berdasarkan pendapat yang di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu software atau program yang digunakan untuk membuat sebuah database yang bersifat open source.

J. *XAMPP*

XAMPP adalah perangkat lunak yang berfungsi sebagai server yang berdiri sendiri dan mendukung berbagai jenis sistem operasi komputer, dan

merupakan kompilasi dari banyak program (XAMPP, 2019) *XAMPP* berfungsi sebagai server yang mandiri (*localhost*) dan terdiri dari bahasa pemrograman yang ditulis dengan *PHP* dan *Perl*, serta program server database *MySQL*, dan *Apache HTTP*. Nama *XAMPP* diambil dari singkatan *X* (tempat sistem operasi dari semua jenis), *Apache*, *MySQL*, *PHP*, dan *Perl*. Di bawah adalah tampilan dari aplikasi *XAMPP*



Gambar 2.1 Tampilan *XAMPP*

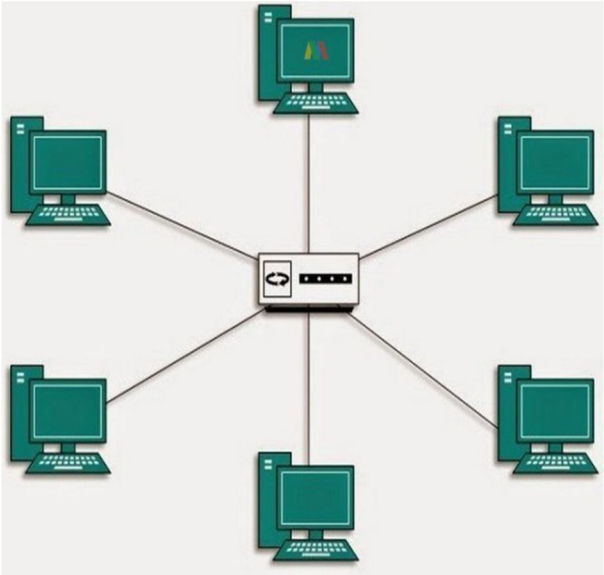
K. *Database*

Database (Nugroho, 2008), adalah kumpulan/kumpulan data yang disimpan dalam memori komputer (biasanya permanen) yang dirancang dan diatur agar pengguna dapat mudah menemukan, mengakses, dan mengedit (memodifikasi, menambah, menghapus). Data ini dapat berupa teks, angka, atau format grafik, dan definisi data (diadopsi oleh *Oracle Corp.*) saat ini didefinisikan untuk tipe data lain, audio, gambar, foto, dan video. Dapat diperluas.

L. *Client Server*

Client Server menurut (Klien-server, 2018) merupakan sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak: pihak klien dan pihak server.

M. Topologi Star



Gambar 2.3 Ilustrasi Topologi Star

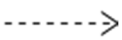
Menurut (Adiputra, 2017), dalam topologi star, setiap komputer host memiliki kabel sendiri yang terhubung langsung ke *Multistation Access Unit* (*MAU*) melalui hub pusat, sakelar, pengulang *multipoint*, atau sistem *point-to-point*.

Topologi star umumnya dipakai dalam jaringan komputer kantor atau rumah. Jenis kabel yang dipakai pada topologi jaringan ini biasanya jenis *unshielded twisted pair* (*UTP*), namun beberapa kabel menggunakan kabel *coaxial* atau *fiber optic*.

N. UML *(Unified Modeling Language)*

UML *(Unified Modeling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Menurut Adi nugroho (2010:6), "*Unified Modeling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang beraparadigma berorientasi objek". Pemodelan sebenarnya digunakan untuk menyederhanaan permasalahan kompleks yang dianalisis dengan sedemikian rupa sehingga mudah untuk dipelajari dan dipahami.

Table 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Mewakili peran orang, sistem yang lain, alat, atau himpunan peran yang pengguna mainkan ketika berinteraksi dan berkomunikasi dengan *use case*.
2		*Dependency*	Hubungan antara elemen mandiri *(independent)* dan elemen yang tidak mandiri *(non independent)*, dimana jika terjadi perubahan pada elemen yang mandiri maka juga akan mengubah elemen yang tidak mandiri.
3		*Generalization*	Rangkaian dimana objek bawah (*descendent*) berbagi tindakan struktur data dengan objek utama (*ancestor*) yang ada di atasnya.
4		*Include*	Menunjukkan dimana suatu *use case* asal secara pasti.
5		*Extend*	Menunjukkan dimana suatu *use case* tujuan memperluas tindakan dari *use case* asal pada suatu titik yang diberikan

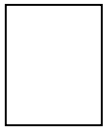
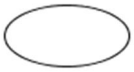
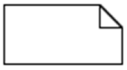
6		*System*	Pengkhususan bagian yang menampilkan suatu operasi yang secara terbatas.
7		*Use Case*	Uraian rangkaian tindakan yang menampilkan hasil suatu sistem yang menunjukkan hubungan dan interaksi antara sistem dan aktor.
8		*Note*	Menunjukkan bagian fisik pada saat aplikasi dijalankan yang menggambarkan sumber daya komputasi.

Table 2.2 Simbol *Class Diagram*

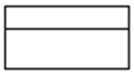
NO	GAMBAR	NAMA	KETERANGAN
1		*Nary Association*	Menunjukkan suatu upaya untuk menghindari asosiasi lebih dari 2 objek.
2		*Class*	Himpunan berbagai objek yang berbagi atribut serta aktivitas yang sama.
3		*Collaboration*	Proses interaksi objek, aturan, atau elemen yang bekerja sama untuk menyelesaikan tindakan yang lebih besar dari elemen-elemennya.
4		*Realization*	Menunjukkan proses aktivitas yang benar-benar sedang dilaksanakan oleh objek.
5		*Dependency*	Hubungan antara elemen mandiri *(independent)* dan elemen yang tidak mandiri *(non independent)*, dimana jika terjadi perubahan pada elemen yang mandiri maka juga akan mengubah elemen yang tidak mandiri.
6		*Association*	Menunjukkan hubungkan interaksi antara objek satu dengan objek lainnya.

Table 2.3 Simbol *Sequence Diagram*

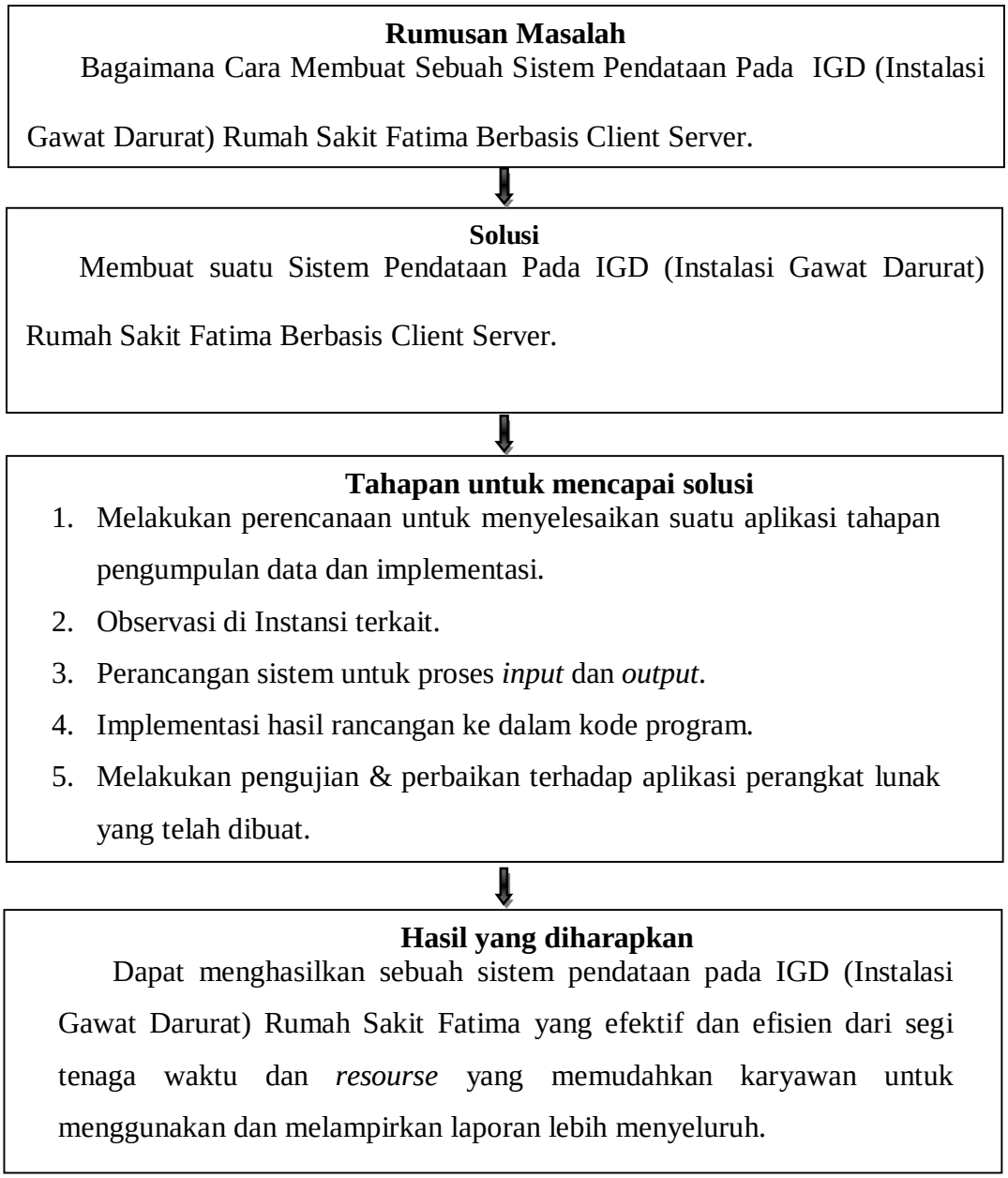
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Hubungan antara suatu objek atau *entitas*, dimana menampilkan antarmuka yang saling berkomunikasi atau berinteraksi.
2		*Message*	spesifikasi komunikasi antara suatu objek yang mengandung data dan informasi tentang aktivitas yang sedang terjadi.
3		*Message*	spesifikasi komunikasi antara suatu objek yang mengandung data dan informasi tentang aktivitas yang sedang terjadi.

Tabel 2.4 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Menunjukkan aktivitas, pekerjaan, atau masing-masing kelas antarmuka yang dijalankan dalam aliran kerja yang aling berinteraksi satu sama lain.
2		*Action*	Menunjukkan status dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana suatu Awal dimulainya aliran kerja pada suatu aktivitas diagram dan pada sebuah aktivitas diagram hanya terdapat satu final state
4		*Activity Final Node*	Bagaimana suatu Akhir dari aliran kerja pada suatu aktivitas diagram dan pada sebuah aktivitas diagram bisa terdapat lebih satu final state
5		*Fork Node*	Menunjukkan bagaimana satu aliran kerja yang pada tahap tertentu berubah menjadi beberapa tahap aliran kerja.

O. Kerangka Berpikir

Untuk memahami kerangka pikir penelitian ini dapat dilihat pada gambar:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

1. Penelitian Kepustakaan

Yaitu Pengumpulan data membaca buku mengenai literatur dan buku lain yang bersifat ilmiah yang berkaitan materi pembahasan seperti

a. *Flutter (Frontend)*
b. *NodeJs* dan *Express (Backend)*
c. *Database MySql*
d. Topologi *Star*
e. Sistem Informasi *Client Server*

2. Penelitian Lapangan

Yaitu kegiatan yang dilakukan cara mengumpulkan data secara langsung dari objek penelitian melalui wawancara dan pengamatan langsung.

B. Waktu Dan Tempat Penelitian

1. Lokasi

Kegiatan penelitian ini dilakukan di *IGD* (Instalasi Gawat Darurat) Rumah Sakit Fatima, Parepare yang beralamat di Jl. Ilham No. 5, Mallusetasi, Kecamatan Ujung, 91114.

2. Waktu

Waktu yang dipergunakan untuk pelaksanaan penelitian ini adalah kurang lebih dua (2) bulan perencanaan sebagai berikut.

Tabel 2.1 Waktu Penelitian

Pengumpulan data dan wawancara	2 (dua) Minggu
Perancangan dan pembuatan *frontend*	2 (dua) Minggu
Perancangan dan pembuatan *database*	2 (dua) Minggu
Perancangan dan pembuatan *backend*	1 (dua) Minggu
Pengujian dan finalisasi aplikasi	1 (dua) Minggu

C. Alat Dan Bahan

Agar dapat melaksanakan penelitian dengan baik, maka yang harus dibutuhkan adalah beberapa alat serta bahan penelitian yang lengkap, guna mendukung proses pelaksanaan penelitian tersebut. Adapun beberapa alat dan bahan yang diperlukan sebagai berikut:

1. Perangkat Keras (Komputer)

Tabel 2.2 Perangkat Keras

Sistem Operasi	*Windows* 10 *Enterprise* 64 *Bit operating system*
Prosesor	*Intel Core* i7-6700HQ
RAM	*Memory 8 GB*
Harddisk	*1 TB HDD 7200 RPM*

Display	*Asus X550VX 15,6”HD, Resolution 1366x768 pixels, Flatron 16 inches*

2. Perangkat Lunak (Software)

Tabel 3.3 Perangkat Lunak

IDE	*Microsoft Visual Studio Code*
Server	*XAMPP versi 3.2.4*
UML	*Diagram Desktop*
Browser	*Google Chrome*

D. Metode Pengumpulan Data

1. Studi Literatur

Metode pengumpulan data yang dilakukan adalah dengan cara mengumpulkan jurnal paper, literatur, skripsi dan bacaan-bacaan yang ada kaitannya judul penelitian.

2. Observasi

Metode pengumpulan data yang dilaksanakan dengan melakukan pengamatan langsung pada *IGD* (Instalasi Gawat Darurat) Rumah Sakit Fatima untuk mengumpulkan informasi dan data yang dibutuhkan dalam penelitian.

3. Wawancara

Metode pengumpulan data yang dilaksanakan dengan mengandalkan wawancara pada staf dan karyawan di *IGD* (Instalasi Gawat Darurat) Rumah Sakit Fatima yang ada kaitannya topik yang diambil. Adapun beberapa pertanyaan yang diajukan untuk memperoleh hasil penelitian.

4. Dokumentasi

Metode pengumpulan data yang dilaksanakan dengan mengambil data yang dibutuhkan pada *IGD* (Instalasi Gawat Darurat) Rumah Sakit Fatima berupa informasi tertulis maupun gambar pada lokasi penelitian.

E. Teknik Analisis Data

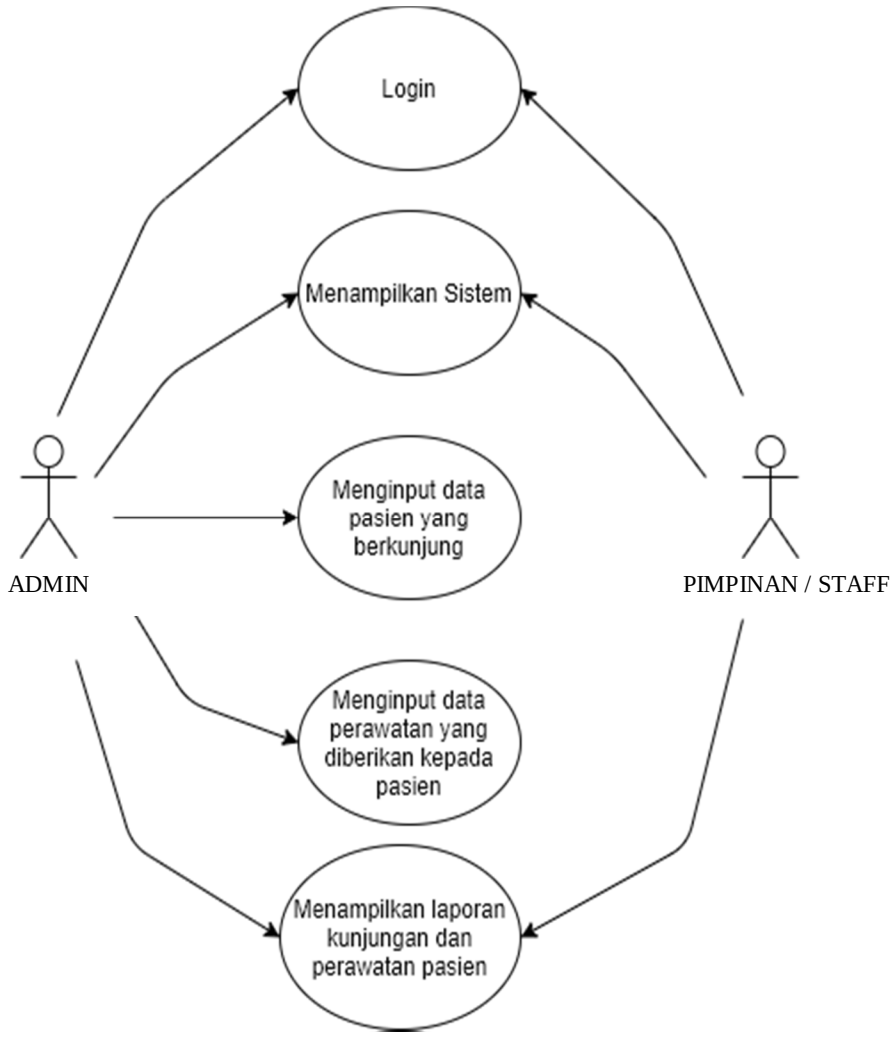
Beberapa tahap-tahap analisis data yang dilakukan dalam proses pelaksanaan berkenaan penelitian ini adalah:

1. Tahap pengumpulan data. Tahap ini dilaksanakan dengan mengumpulan data serta informasi yang diperlukan untuk merancangan Sistem Pendataan Pada Rumah Sakit Fatima Berbasis *Client Server*.
2. Tahap analisis data. Tahap ini dilaksanakan dengan menganalisis terhadap data dan informasi yang telah dikumpulkan pada tahap pengumpulan data, menganalisis cara kerja sistem yang akan dirancang, mengidentifikasi masalah, dan menganalisa kebutuhan sistem.
3. Tahap perancangan. Tahap ini dari hasil analisis data dan informasi, dilaksanakan perancangan diagram *UML* dan hasil output dari aplikasi yang akan dibuat
4. Tahap pembuatan aplikasi. Tahap ini aplikasi yang dibuat menggunakan text editor *Sublime Text*. *Bootstrap* (*Template*) yang telah didownload, dan juga penulisan syntax menggunakan bahasa pemrograman, *Javascript, Electron, PHP, Node.JS*. Aplikasi dibangun menggunakan *framework Flutter* menggunakan *database MYSQL*.
5. Tahap Pengujian aplikasi. Tahap ini aplikasi yang telah dibuat diuji menggunakan metode *Black Box* dan *White Box* umtuk mengetahui tingkat kesalahan dan keberhasilannya.
6. Tahap Implementasi. Tahap ini dilaksanakan setelah tidak adanya kesalahan pada pengunjian aplikasi, sehingga aplikasi siap untuk diterapkan, dengan ini seluruh tahap selesai dilaksanakan.

F. Rancangan Penelitian

Use Case Diagram

Dibawah merupakan sistem yang diusulkan oleh penulis. Dimana karyawan yang bertugas menginput data pasien yang berkunjung dan bisa menampilkan laporan yang diproses oleh sistem secara otomatis berdasarkan inputan yang dimasukkan oleh karyawan yang bertugas.



Gambar 2.1 *Use Case Diagram*

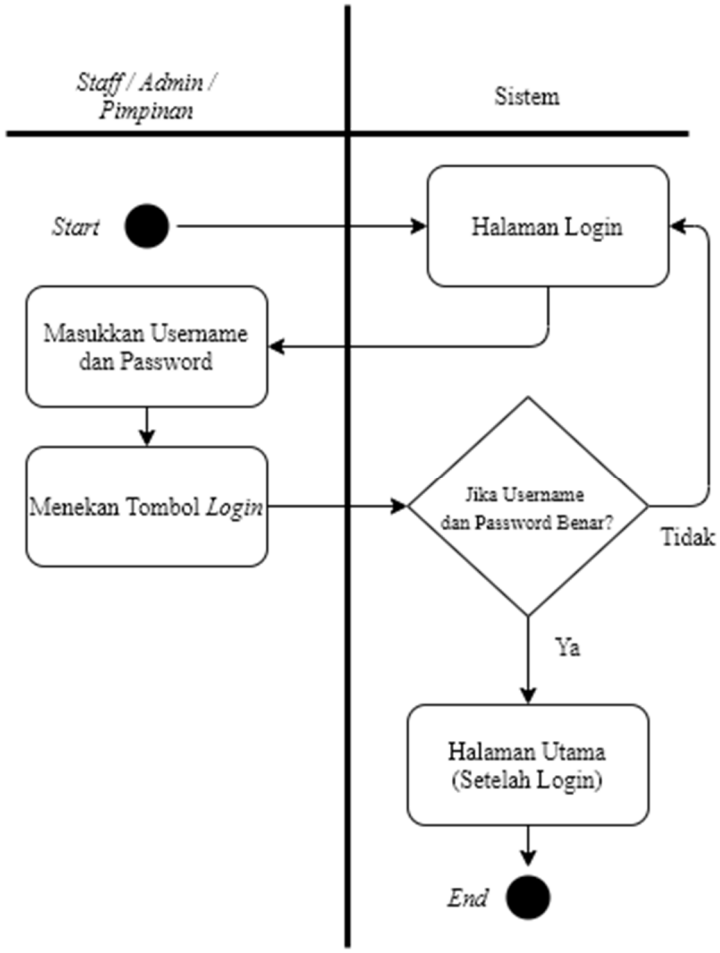
Tabel 2.4 Rancangan Penelitian

Nama Use Case	Deskripsi Use Case
Login	Use Case ini menjelaskan bahwa admin dan pimpinan harus login untuk mengakses sistem.
Menampilkan sistem	Use Case ini menjelaskan bahwa admin dan pimpinan dapat menampilkan sistem setelah login berhasil
Menginput data pasien yang berkunjung	Use Case ini menjelaskan bahwa admin bertugas untuk menginput data setiap pasien yang berkunjung ke *IGD* rumah sakit
Menginput data perawatan yang diberikan kepada pasien	Use Case ini menjelaskan bahwa admin bertugas untuk menginput data perawatan yang diberikan kepada setiap pasien yang berkunjung
Menampilkan laporan kunjungan dan perawatan pasien	Use Case ini menjelaskan bahwa admin dan pimpinan dapat menampilkan laporan data kunjungan yang telah diproses serta data perawatan setiap pasien.

G. Diagram Alir

1. Activity Diagram

a. *Activity Diagram Login*

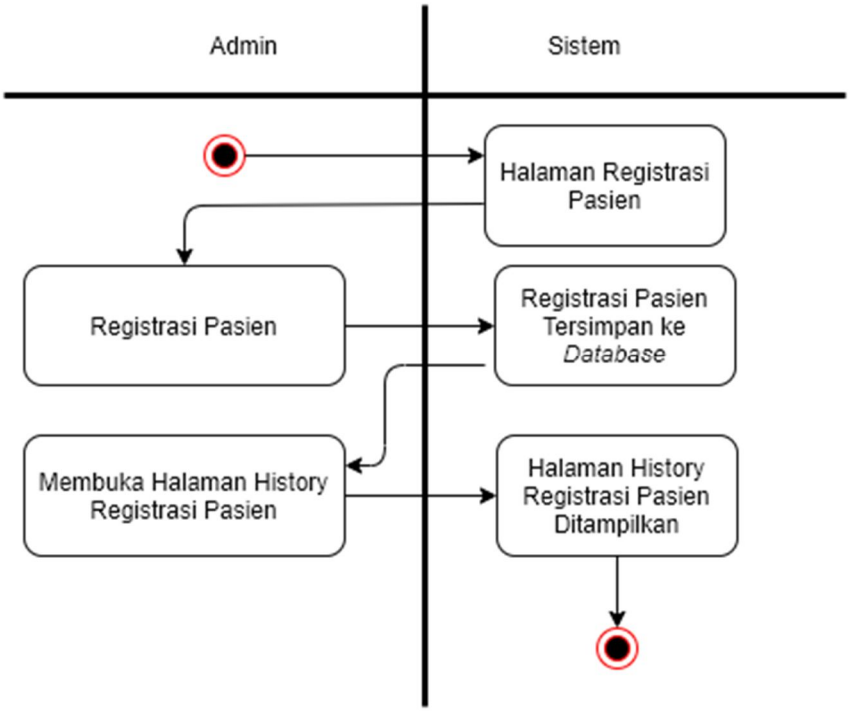


Gambar 2.2 *Activity Diagram Login*

Activity Diagram diatas menjelaskan bahwa *staff, admin* maupun pimpinan akan dihadapkan ke halaman *login.* Kemudian disini *staff, admin* maupun admin akan mengisi *username* dan *password* dan kemudian menekan tombol *login.* Sistem akan memeriksa *username* dan *password* yang dimasukkan jika benar maka akan dihadapkan ke halaman utama *staff, admin* maupun *pimpinan* setelah sukses *login.* Namun jika

username dan *password* salah maka *user* maupun admin akan dihadapkan kembali ke tampilan login

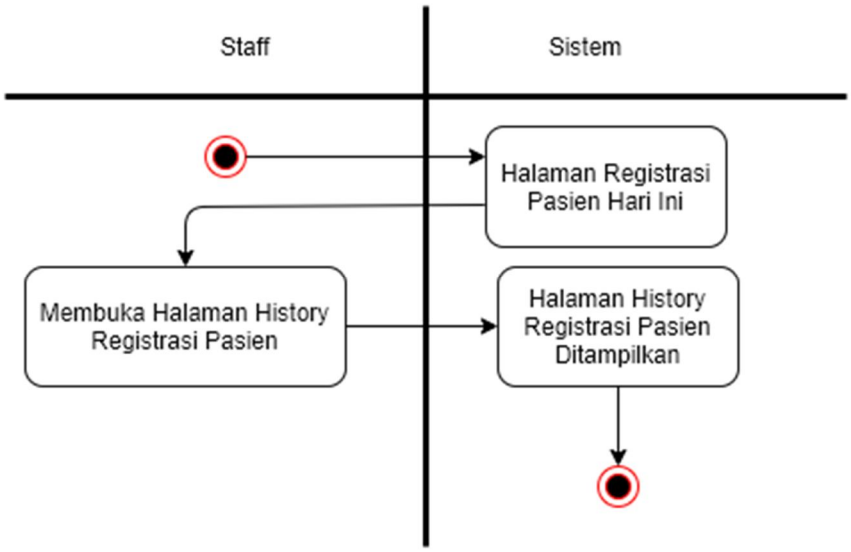
b. *Activity Diagram Admin*



Gambar 2.3 *Activity Diagram Admin*

Activity Diagram diatas menjelaskan bahwa admin setelah sukses login akan dihadapkan ke tampilan registrasi pasien. Kemudian admin dapat meregistrasi pasien. Kemudian registrasi pasien akan tersimpan ke *database* sistem. Admin juga dapat menekan sub menu Halaman history registrasi pasien, kemudian aplikasi akan menampilkan halaman history *registrasi* pasien dimana di halaman ini admin dapat melihat *history* registrasi pasien yang telah berlaku sebelumnya.

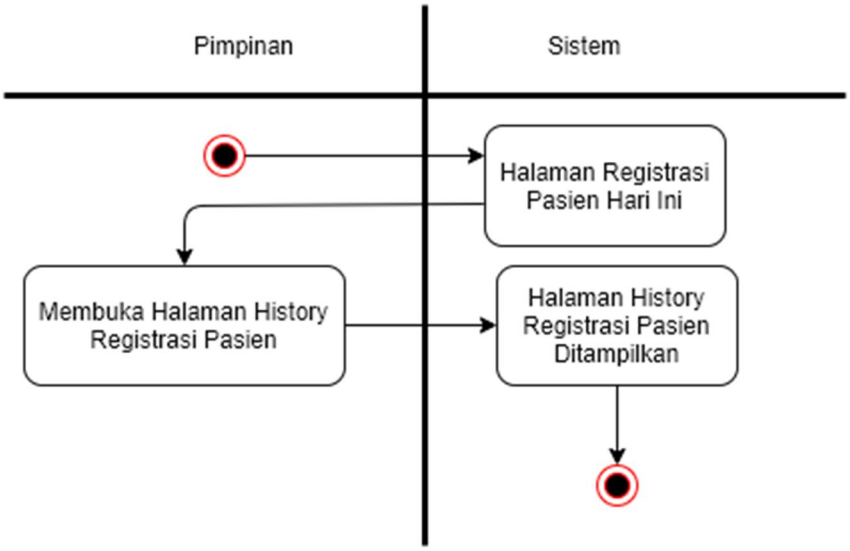
c. *Activity Diagram Staff*



Gambar 2.4 *Activity Diagram Staff*

Activity Diagram diatas menjelaskan bahwa staf setelah sukses login akan dihadapkan ke tampilan registrasi pasien untuk hari ini. Staf juga dapat menekan sub menu Halaman history registrasi pasien, kemudian aplikasi akan menampilkan halaman history *registrasi* pasien dimana di halaman ini admin dapat melihat *history* registrasi pasien yang telah berlaku sebelumnya berdasarkan jenis kasus dari staf tersebut.

d. *Activity Diagram Pimpinan*

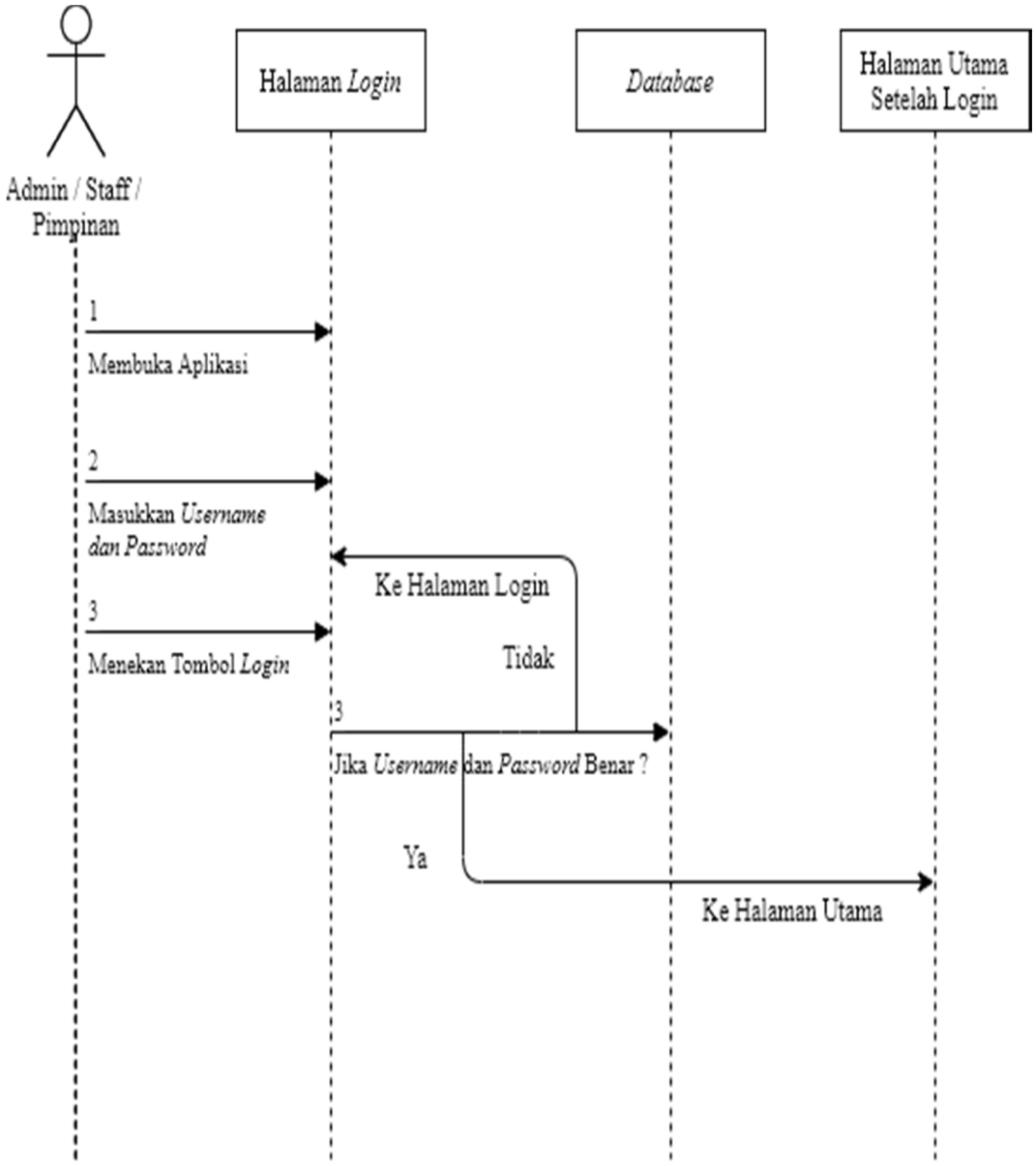


Gambar 2.5 *Activity Diagram Pimpinan*

Activity Diagram diatas menjelaskan bahwa staf setelah sukses login akan dihadapkan ke tampilan registrasi pasien untuk hari ini. Pimpinan juga dapat menekan sub menu Halaman history registrasi pasien, kemudian aplikasi akan menampilkan halaman history *registrasi* pasien dimana di halaman ini admin dapat melihat *history* registrasi pasien yang telah berlaku sebelumnya.

2. *Sequence Diagram*

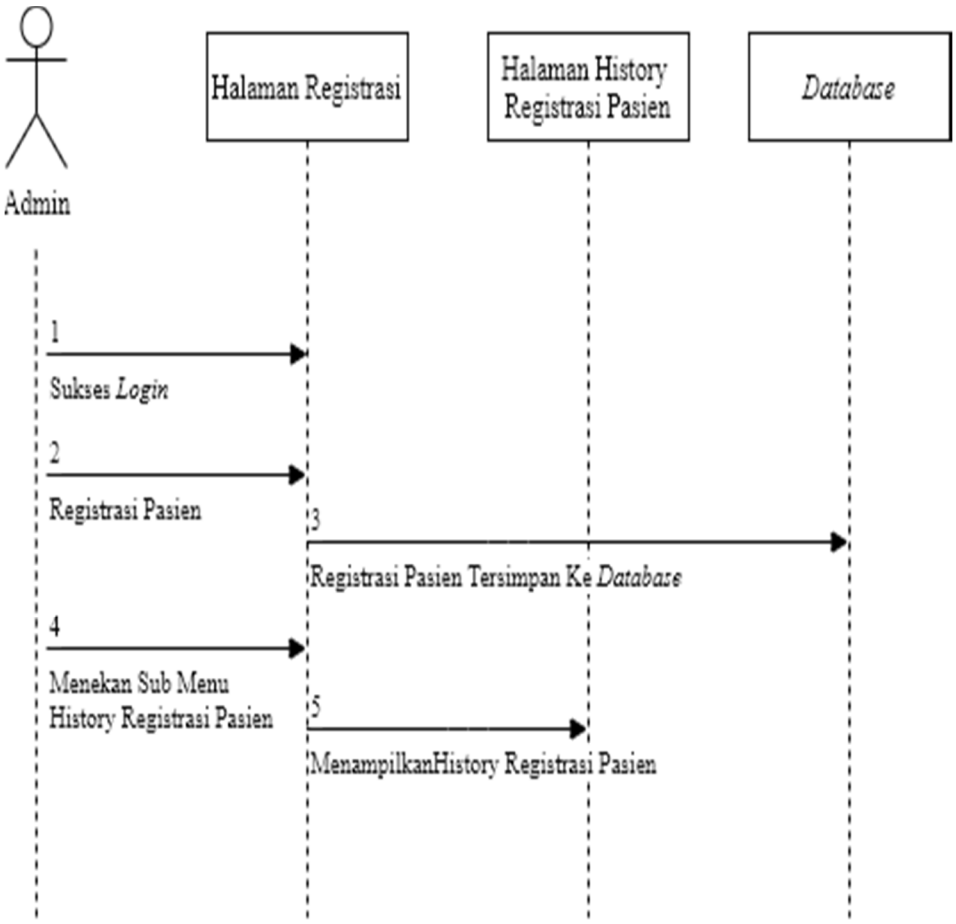
a. *Sequence Diagram Login*



Gambar 2.6 Sequence Diagram Login

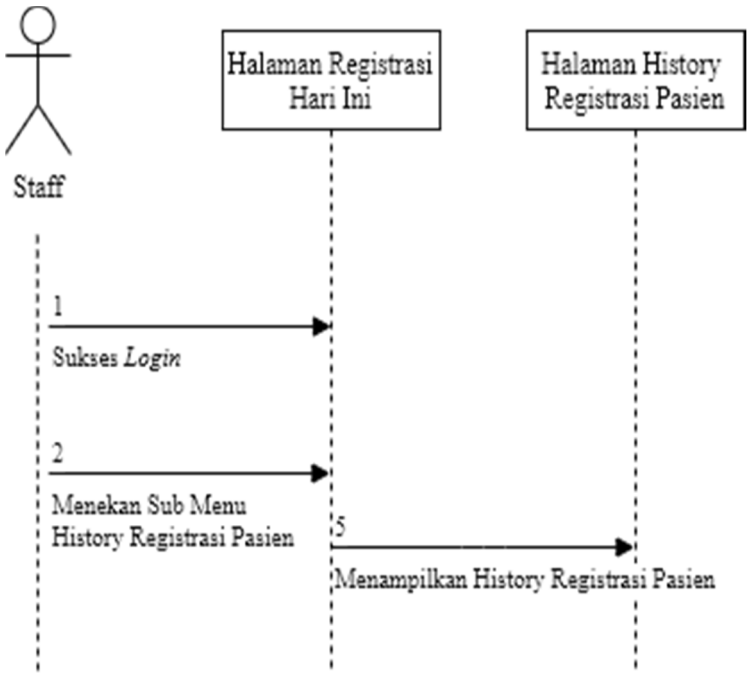
.

b. Sequence Diagram Admin



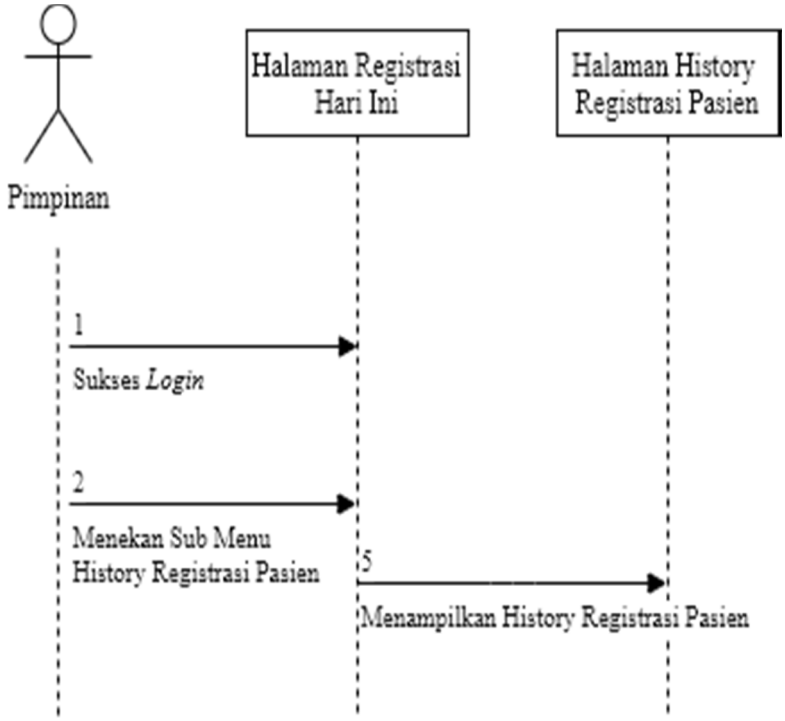
Gambar 2.7 *Sequence Diagram Admin*

c. Sequence Diagram Staff



Gambar 2.8 *Sequence Diagram Staff*

d. Sequence Diagram Pimpinan



Gambar 2.9 *Sequence Diagram Pimpinan*

BAB IV

HASIL DAN PEMBAHASAN

A. Metode Pengujian

Pengujian aplikasi dilakukan dengan dua cara pengujian: metode pertama yaitu secara *Black Box* yang menguji pada fungsionalitas *input* dan *output*, dan metode kedua yaitu secara *White Box* yang menganalisa kode dan struktur internal aplikasi.

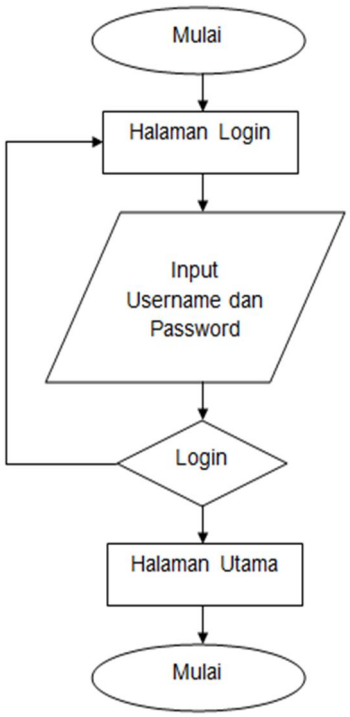
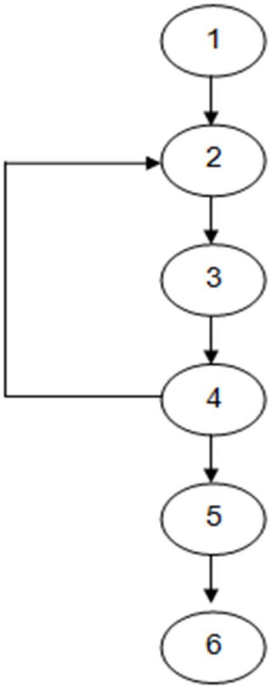
1. Pengujian *Black box* yaitu menguji pada fungsionalitas *input* dan *output* dan antarmuka perangkat lunak pada aplikasi.
2. pengujian *White Box* semua berfokus pada efektifitas kode dan struktur internal aplikasi yang dirancang.

Meskipun dirancang untuk mendekteksi kesalahan, pengujian *Black Box* dan *White Box* digunakan untuk menunjukkan bahwa semua fungsi perangkat lunak beroperasi dengan benar, input yang diterima dan output yang dihasilkan benar, dan integritas informasi external dipertahankan. Pengujian *Black Box* memeriksa beberapa aspek dasar aplikasi dengan sedikit berfokus pada struktur logis internal aplikasi tersebut, sedangkan pengujian *White Box* semua berfokus pada menguji kode sumber dan struktur internal aplikasi yang telah dibuat.

B. Pengujian *White Box*

1. White Box Login

Tabel 3.1 *White Box Login*

Flowchart	*Flowgraph*
	

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

Independent Path.

1) *Path* 1 = 1-2-3-4-5-6
2) *Path* 2 = 1-2-3-4-2

Cyclomatic Complexity.

V(G) = E - N + 2

= 6 - 6 + 2 = 2

V(G) = P + 1

= 1 + 1

= 2

Region = 2

Grafik Matriks

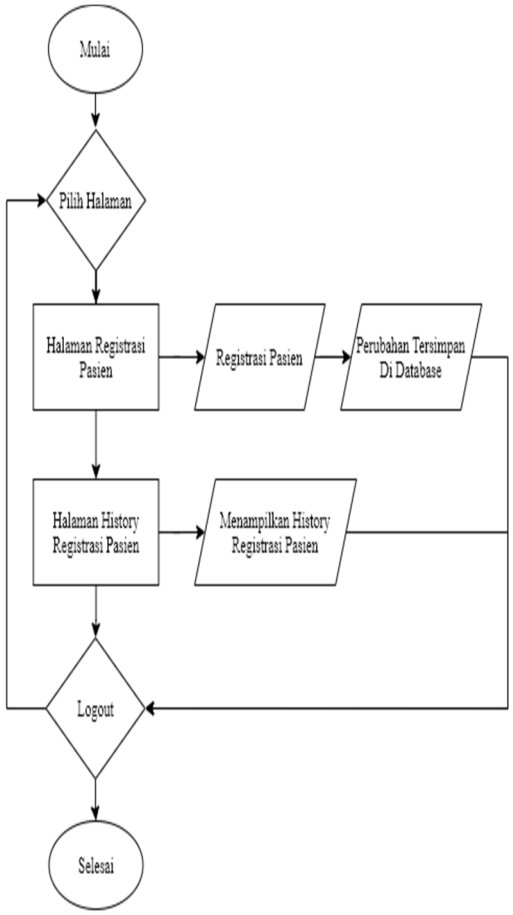
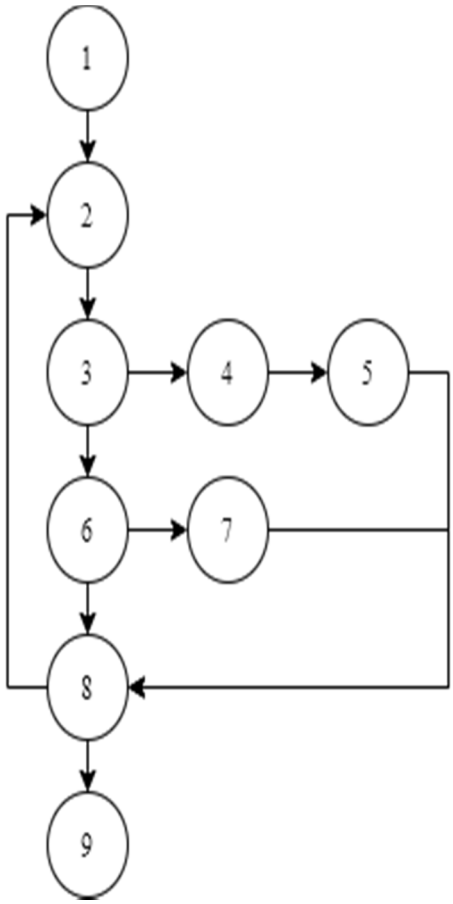
Tabel 3.2 Grafik Matriks L*ogin*

	1	2	3	4	5	6	E-1
1		1					1 - 1 = 0
2			1				1 - 1 = 0
3				1			1 - 1 = 0
4		1			1		2 - 1 = 1
5						1	1 - 1 = 0
6							
SUM (E + 1)							1 + 1 = 2

Berdasarkan perhitungan diatas, Region = 2, Independent Path = 2, Cyclomatic Complexity = 2, maka form Login dinyatakan bebas dari kesalahan.

2. White Box Admin

Tabel 3.3 White Box Admin

Flowchart	*Flowgraph*
	

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

Independent Path.

1) *Path* 1 = 1-2-3-6-8-9
2) *Path* 2 = 1-2-3-4-5-8-9
3) *Path* 3 = 1-2-3-6-7-8-9

4) *Path* 3 = 1-2-3-6-7-8-2

Cyclomatic Complexity.

V(G) = E - N + 2

= 11 - 9 + 2 = 4

V(G) = P + 1

= 3 + 1

= 4

Region = 4

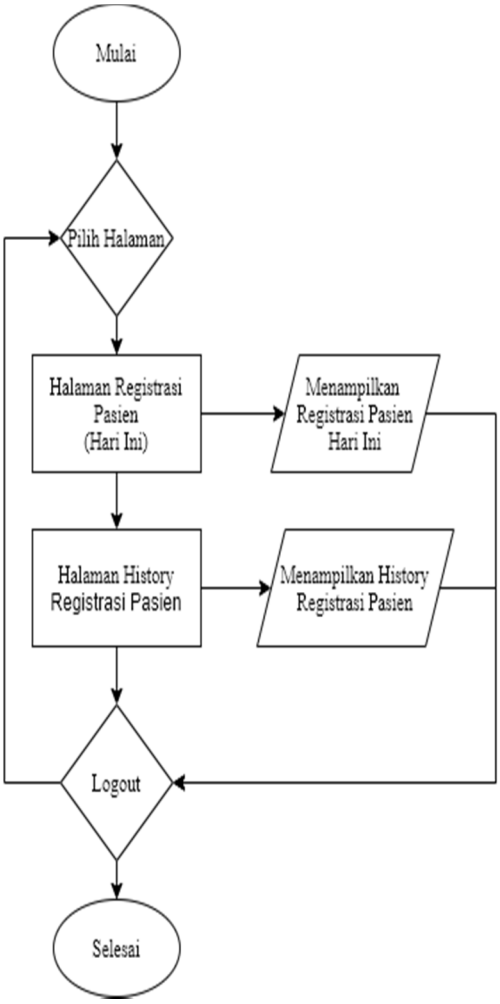
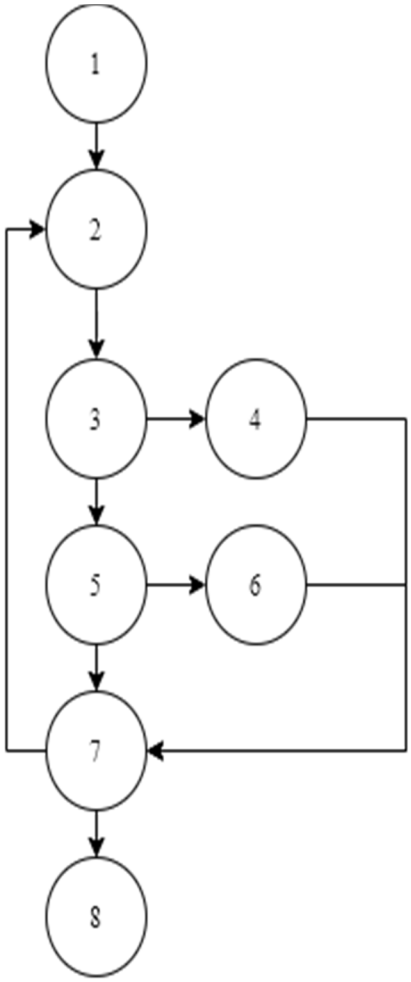
Tabel 3.4 Grafik Matriks Admin

	1	2	3	4	5	6	7	8	9	E-1
1		1								1-1=0
2			1							1-1=0
3				1		1				2-1=1
4					1					1-1=0
5								1		1-1=0
6							1	1		2-1=1
7								1		1-1=0
8		1							1	2-1=1
9										
SUM (E + 1)										3 + 1 = 4

Berdasarkan perhitungan diatas, Region = 4, Independent Path = 4 Cyclomatic Complexity = 4, maka *White Box* Admin dinyatakan bebas dari kesalahan.

3. White Box Staff

Tabel 3.5 *White Box Staff*

Flowchart	*Flowgraph*
	

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

Independent Path.

1) *Path* 1 = 1-2-3-5-7-8
2) *Path* 2 = 1-2-3-4-7-8
3) *Path* 3 = 1-2-3-5-6-7-8

4) *Path* 3 = 1-2-3-5-6-7-2

Cyclomatic Complexity.

V(G) = E - N + 2

= 10 - 8 + 2 = 4

V(G) = P + 1

= 3 + 1

= 4

Region = 4

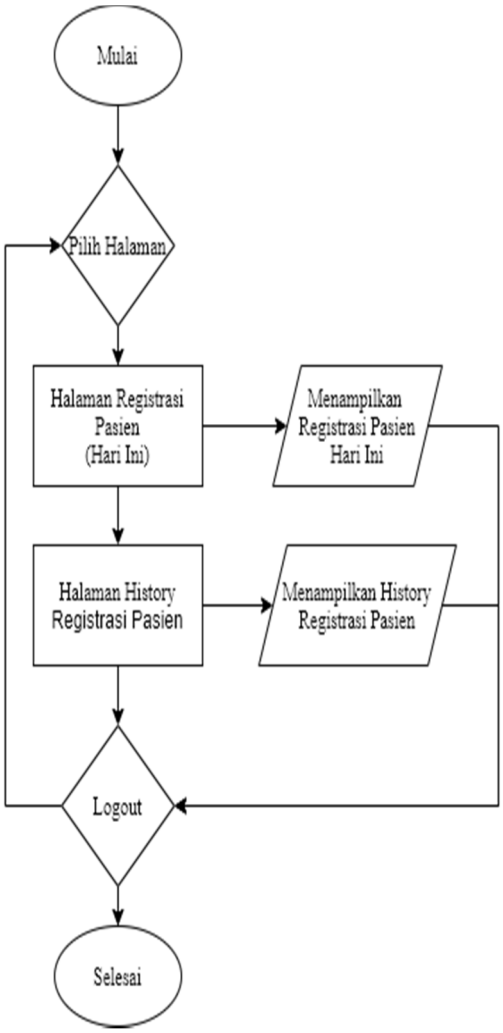
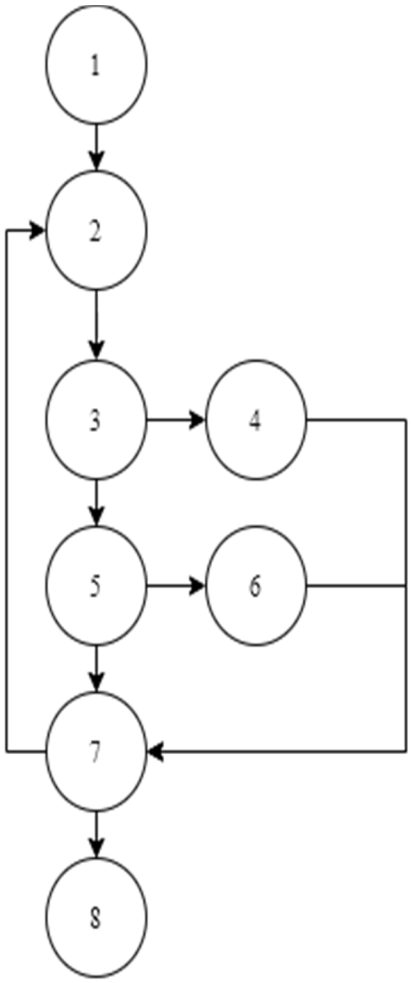
Tabel 3.6 Grafik Matriks Staf

	1	2	3	4	5	6	7	8	E-1
1		1							1-1=0
2			1						1-1=0
3				1	1				2-1=1
4							1		1-1=0
5						1	1		2-1=1
6							1		1-1=0
7		1						1	2-1=1
8									
SUM (E + 1)									3 + 1 = 4

Berdasarkan perhitungan diatas, Region = 4, Independent Path = 4 Cyclomatic Complexity = 4, maka *White Box* Staf dinyatakan bebas dari kesalahan.

4. White Box Pimpinan

Tabel 3.7 *White Box* Pimpinan

Flowchart	*Flowgraph*
	

Menghitung jumlah *Region, Cyclomatic Complexity* dan *Independent Path.*

Independent Path.

1) *Path* 1 = 1-2-3-5-7-8

2) *Path* 2 = 1-2-3-4-7-8

3) *Path* 3 = 1-2-3-5-6-7-8

4) *Path* 3 = 1-2-3-5-6-7-2

Cyclomatic Complexity.

V(G) = E - N + 2

= 10 - 8 + 2 = 4

V(G) = P + 1

= 3 + 1

= 4

Region = 4

Tabel 3.8 Grafik Matriks Pimpinan

	1	2	3	4	5	6	7	8	E-1
1		1							1-1=0
2			1						1-1=0
3				1	1				2-1=1
4							1		1-1=0
5						1	1		2-1=1
6							1		1-1=0
7		1						1	2-1=1
8									
SUM (E + 1)									3 + 1 = 4

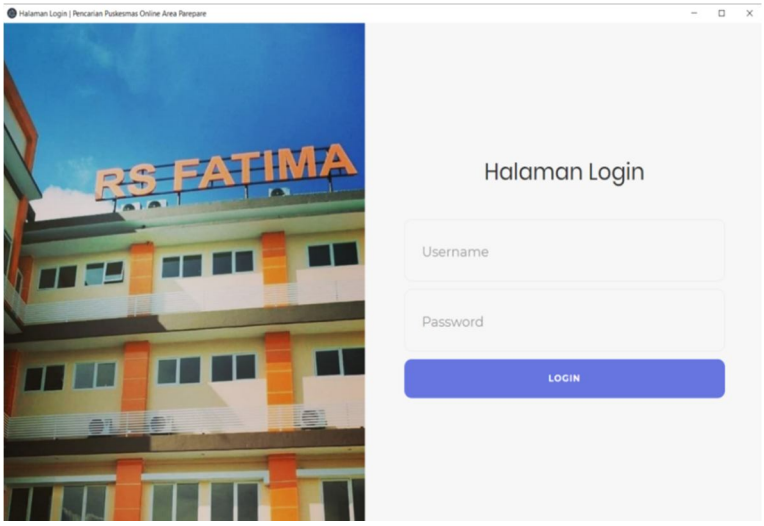
Berdasarkan perhitungan diatas, Region = 4, Independent Path = 4 Cyclomatic Complexity = 4, maka *White Box* Pimpinan dinyatakan bebas dari kesalahan.

C. Pengujian *Black Box*

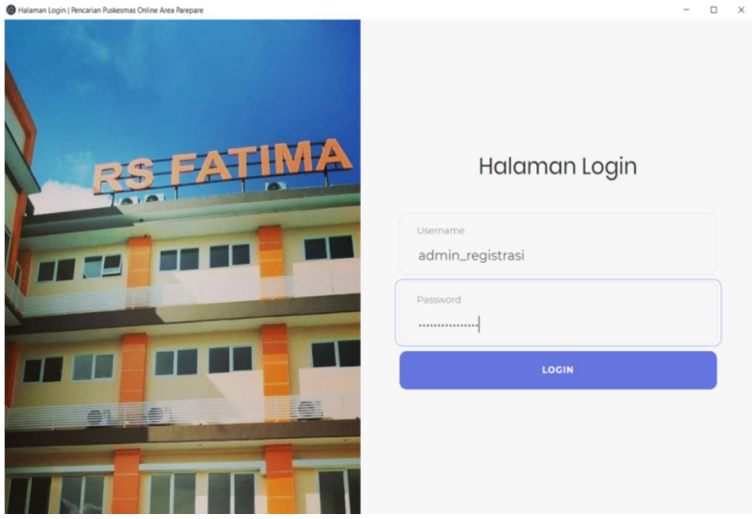
1. Pengujian Form *Login*

Tabel 3.9 *Black Box* Login

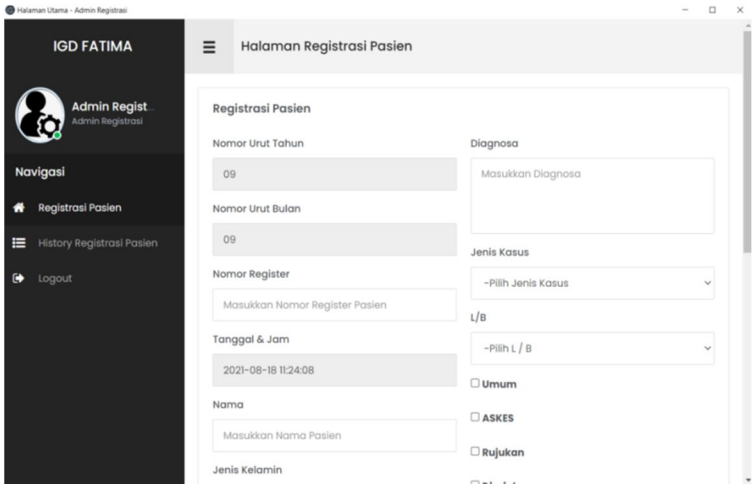
Test Factor	Hasil	Keterangan
Login dengan memasukkan *username* dan *Password*	Berhasil	Jika *Username* dan *Password* benar, maka akan login ke halaman utama registrasi pasien



Gambar 4.1 Tampilan Halaman Login



Gambar 4.2 Masukkan Username dan Password

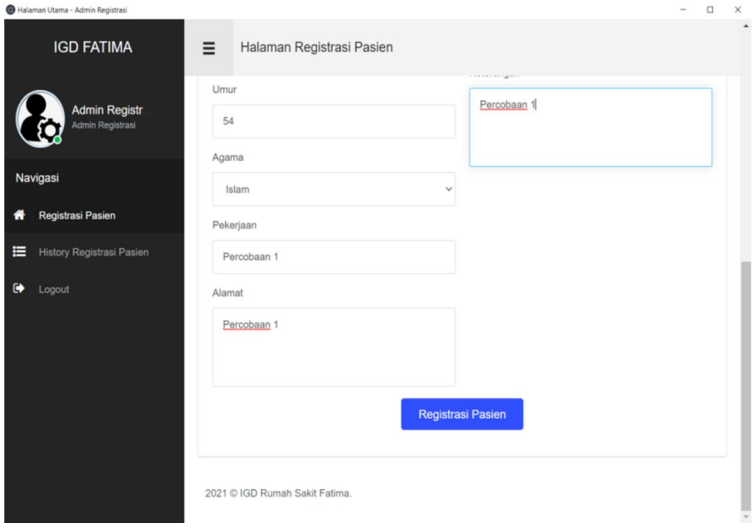


Gambar 4.3 Berhasil Masuk Ke Halaman Registrasi Pasien

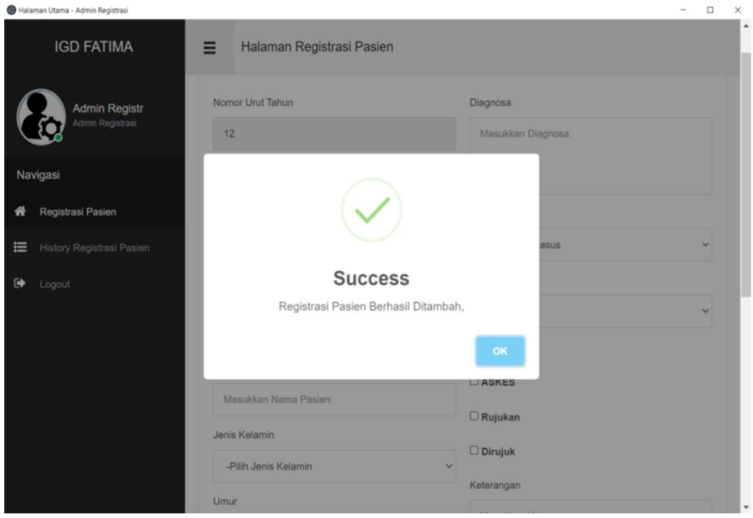
2. Pengujian Form Registrasi Pasien

Tabel 3.10 *Black Box* Registrasi Pasien

Test Factor	Hasil	Keterangan
Masukkan semua *data* pasien dengan benar	Berhasil	Jika semua *data* pasien yang dimasukkan benar, maka pada halaman registrasi pasien akan terdapat laporan *Success*



Gambar 4.4 Menambah Data Pasien Pada Halaman Registrasi

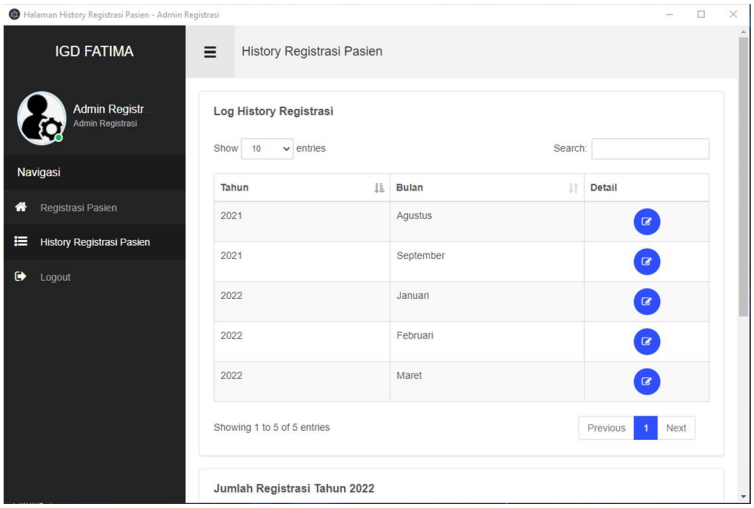


Gambar 4.5 Berhasil Menambahkan Data Pasien

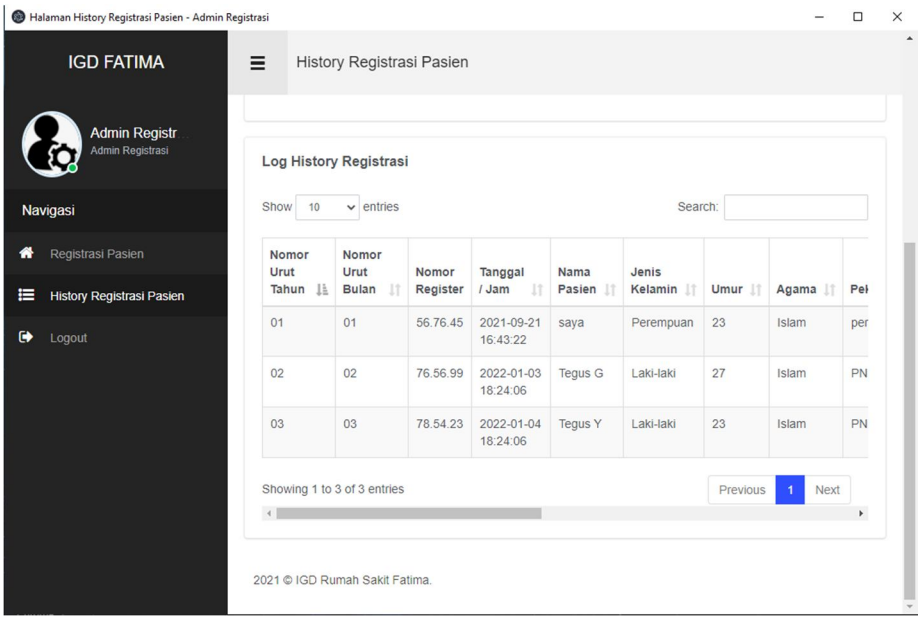
3. Pengujian Form History Registrasi Pasien

Tabel 3.11 *Black Box* History Registrasi Pasien

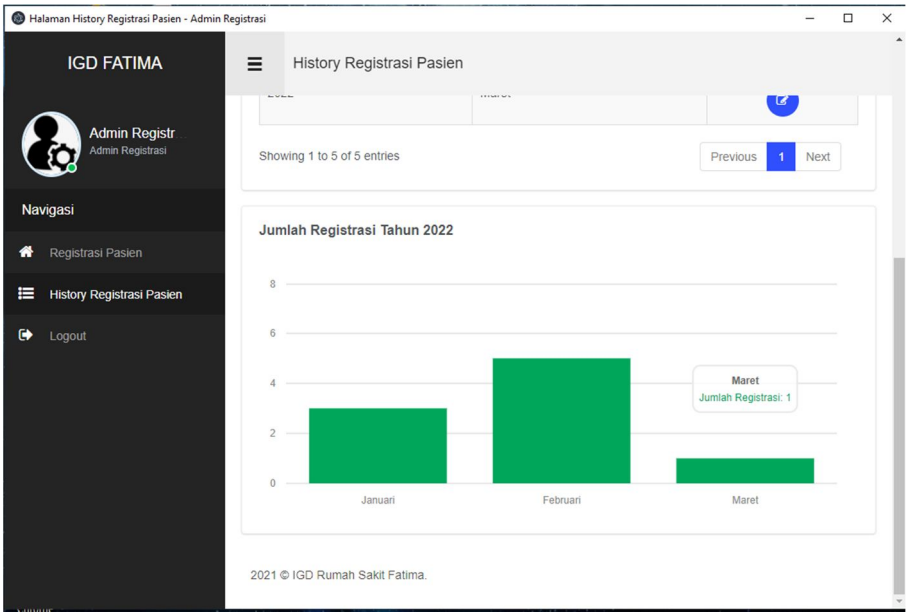
Test Factor	Hasil	Keterangan
Masukkan semua *data* pasien dengan benar	Berhasil	Jika semua *data* pasien yang dimasukkan benar, maka jumlah *data* dan statistik pada halaman registrasi pasien akan bertambah



Gambar 4.6 History Registrasi Pasien



Gambar 4.7 Detail Jumlah Registrasi Pasien



Gambar 4.8 Statistik Jumlah Registrasi Pasien

BAB V

PENUTUP

A. Kesimpulan

Dari hasil penelitian yang telah dilakukan serta pembahasan pada bab sebelumnya, maka peneliti dapat menyimpulkan sebagai berikut; Telah berhasil membuat sebuah aplikasi Penerapan Sistem Pendataan Pada *IGD* (Instalasi Gawat Darurat) Rumah Sakit Fatima Berbasis Client Server, menggunakan bahasa pemrograman *Javascript, Electron, HTML* dan *CSS* serta *database* menggunakan *MYSQL*. Dimana admin pada sistem ini dapat mengelola pendataan registrasi pasien pada *IGD* Rumah Sakit Fatima, dimana staf dan pimpinan dapat melihat registrasi pasien yang dilakukan oleh admin sebelumnya untuk dijadikan laporan.

Aplikasi yang dibuat telah melewati dua metode pengujian dan bebas dari kesalahan, metode pengujian pertama yaitu secara *Black Box* dan metode kedua secara *White Box*.

B. Saran

Setelah melakukan penelitian di *IGD* Rumah Sakit Fatina, maka penulis mengajukan saran. Untuk pengembangan lebih lanjut sebaiknya aplikasi ini dikembangkan menjadi *WEB* bagi pimpinan agar dapat diakses bagi pimpinan di mana saja.

DAFTAR PUSTAKA

Adiputra, Y. R. (2017). *Topologi Jaringan Komputer.* Semarang.

Agung, L. (2010). *Aplikasi Pemrograman Javascript untuk Halaman Web.* Yogyakarta: Andi Offset.

Dharwiyanti, S. (2003). *Pemodelan Visual UML.* Yogyakarta: Graha Ilmu.

Erwin, Zellah, A. A., Facrurrozi, & Primanita, A. (2016). Perancangan Database Sistem Informasi Rumah Sakit Bhayangkara Palembang Pada Citra Lidah. *Perancangan Database Sistem Informasi Rumah Sakit Bhayangkara Palembang Pada Citra Lidah* .

Haryanto, & Priyatmoko, B. (2015). Sistem Informasi Pengolahan Data Pasien Rawat Inap Berbasis SMS Gateway. *Sistem Informasi Pengolahan Data Pasien Rawat Inap Berbasis SMS Gateway* , 34-42.

Jubilee. (2011). *HTML 5 Cara cepat dan mudah menguasai script HTML 5.* Yogyakarta: Jubilee Enterprise.

Klien-server. (2018, 08 17). Retrieved 08 10, 2019, from Wikipedia: https://id.wikipedia.org/wiki/Klien-server

Munawar. (2005). *Permodelan Visual UML.*Yogyakarta: Pustaka Pelajar.

Nugroho, A. (2008). *Menjadi Administrator Basis Data ORACLE 10 g.* Bandung: Informatika Bandung.

Setiawan, A. T., & Permadani, R. P. (2016). Sistem Informasi Manajemen Rumah Sakit Welasasih Ambarawa. *Sistem Informasi Manajemen Rumah Sakit Welasasih Ambarawa* , 52-60.

Solichin, A. (2010). *MYSQL 5 Dari Pemula Hingga Mahir*. Jakarta.

XAMPP. (2019, 06 24). Retrieved 08 10, 2019, from Wikipedia: https://id.wikipedia.org/wiki/XAMPP