

BAB I

PENDAHULUAN

A. Latar Belakang

Pusat Kesehatan Masyarakat (Puskesmas) adalah fasilitas pelayanan kesehatan di Indonesia yang bertujuan menyelenggarakan upaya kesehatan masyarakat dan perawatan *individu* tingkat pertama. Puskesmas mengedepankan upaya *promotif* dan *preventif* guna mencapai derajat kesehatan masyarakat yang optimal. Upaya kesehatan tersebut diselenggarakan dengan menitikberatkan kepada pelayanan untuk masyarakat luas guna mencapai derajat kesehatan yang optimal, tanpa mengabaikan mutu pelayanan kepada perorangan. Puskesmas dipimpin oleh seorang kepala Puskesmas yang bertanggung jawab kepada Dinas Kesehatan Kabupaten/Kota. Puskesmas Buntu Batu merupakan salah satu puskesmas yang berada di kabupaten Enrekang, Kecamatan Buntu Batu, Desa Pasui. Puskesmas tersebut merupakan satu-satunya puskesmas yang ada di Kecamatan Buntu Batu, beberapa desa di kecamatan Buntu Batu memiliki jarak yang lumayan jauh untuk ke puskesmas sehingga hampir setiap hari banyak pasien yang datang untuk berobat sementara itu fasilitas yang ada kurang memadai seperti tempat duduk untuk menunggu sehingga banyak pasien memilih untuk keluar sementara untuk mencari tempat istirahat.

Dalam era *digitalisasi* kesehatan yang terus berkembang, pelayanan kesehatan perlu terus *berinovasi* untuk memberikan pengalaman yang lebih *efisien* dan nyaman bagi pasien. Dalam dunia pelayanan kesehatan, kenyamanan pasien

dan *efisiensi* layanan menjadi faktor *krusial* yang memengaruhi kualitas suatu fasilitas kesehatan. Salah satu aspek penting dalam pengelolaan pelayanan kesehatan adalah sistem antrian pasien yang dapat memastikan proses pendaftaran dan penanganan pasien berlangsung dengan baik. Saat ini, masih banyak terjadi kendala dalam pengelolaan antrian pasien yang dapat memakan waktu dan menimbulkan ketidaknyamanan. Oleh karena itu, dirasa penting untuk mengembangkan Sistem Antrian Pasien berbasis *web* yang memanfaatkan *framework CodeIgniter*. Dengan memanfaatkan teknologi *web*, pasien dapat melakukan pendaftaran antrian secara *online*, mengakses informasi antrian secara *real-time*, dan meminimalkan waktu menunggu di lokasi pelayanan kesehatan.

Selain itu, untuk meningkatkan keterjangkauan informasi kepada pasien, perlu diterapkan pula sistem notifikasi otomatis melalui aplikasi pesan instan seperti *WhatsApp*. Dengan integrasi ini, pasien akan menerima notifikasi secara langsung mengenai status antrian mereka, memungkinkan mereka untuk lebih efisien mengatur waktu kedatangan ke fasilitas kesehatan. Melalui penerapan Sistem Antrian Pasien *Web* dan Notifikasi Otomatis, diharapkan dapat meningkatkan efisiensi pelayanan kesehatan dan memberikan pengalaman positif kepada pasien, sejalan dengan semangat terus berkembangnya teknologi dalam sektor kesehatan.

Oleh karena itu, penulis mengangkat judul proposal "*Sistem Antrian Pasien berbasis Web dan Notifikasi Otomatis Melalui WhatsApp dengan Framework CodeIgniter*". Diharapkan dengan adanya sistem antrian ini mempermudah

masyarakat yang memiliki tempat tinggal jauh untuk mendaftar antrian di puskesmas tersebut sehingga mereka tidak mengantri terlalu lama di puskesmas.

B. Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana cara membuat antrian pasien puskesmas dengan notifikasi whatsapp berbasis web pada Puskesmas Buntu Batu ?

C. Batasan Masalah

1. Aplikasi ini hanya digunakan untuk mengambil nomor antrian, tidak ada ketersediaan obat.
2. Aplikasi tidak dibuat berbasis IOS.
3. Aplikasi ini hanya bisa di gunakan di Puskesmas Buntu Batu.

D. Tujuan Penelitian

Penelitian ini bertujuan untuk mengetahui cara membuat aplikasi antrian dengan nontifikasi whatsapp berbasis web.

E. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini yaitu:

1. Bagi Penulis

Dapat mengembangkan wawasan keilmuan dan meningkatkan pemahaman tentang struktur, sistem kerja dan perancangannya dalam pembuatan aplikasi.

2. Bagi Pengguna

Dengan adanya sistem antrian yang terintegrasi secara online, pasien dapat mengurangi waktu tunggu mereka di fasilitas kesehatan. Pendaftaran antrian

yang lebih efisien dan notifikasi otomatis melalui WhatsApp akan membantu pasien untuk lebih baik mengelola waktu kedatangan mereka, mengurangi ketidaknyamanan dan meningkatkan kepuasan pasien.

3. Bagi Instansi

Meningkatkan efisiensi dan efektivitas proses antrian dan meningkatkan kualitas layanan kesehatan.

F. Sistematika Penulisan

Berdasarkan latar belakang yang dikemukakan diatas, sistematika penulisan dari penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Dalam bab ini dijelaskan mengenai hal yang terdiri dari Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian , Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Dalam tinjauan pustaka ini berisi tentang teori-teori yang berhubungan dengan Penelitian Proposal Skripsi dari berbagai sumber mulai dari buku dan jurnal.

BAB III METODE PENELITIAN

Dalam bab ini dijelaskan tentang rencana penelitian, objek penelitian, teknik pengumpulan data dengan observasi dan kondisi tempat penelitian, bab ini juga berisi metode pengembang *Prototype* yang digunakan untuk Sistem Antrian Pasien Berbasis Web Dan Notifikasi Otomatis Melalui *Whatsapp Framework Codeigniter*, dibab ini juga berisi tentang perbandingan penelitian terdahulu dan analisis masalah yang ada pada proses bisnis yang sedang berjalan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi usulan proses Sistem Antrian Pasien Berbasis *Web* Dan *Notifikasi* Otomatis Melalui *Whatsapp Framework Codeigniter* usulan proses setelah diterapkan dalam bentuk rancangan diagram *use case diagram*, *class diagram* , *activity diagram*, dan *sequence diagram*.

BAB V KESIMPULAN

Pada bab ini berisi kesimpulan yang telah didapat dari penelitian terkait laporan tugas akhir.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Antrian

Antrian adalah situasi barisan tunggu dimana sejumlah kesatuan pendatang sedang berusaha untuk menerima pelayanan dari fasilitas terbatas (pemberi pelayanan), sehingga pendatang harus menunggu beberapa waktu dalam barisan agar dilayani (Abbas et al., 2022).

Antrian adalah konsep dalam dunia teknologi dan manajemen yang mengacu pada rangkaian entitas yang menunggu untuk diproses. Entitas-entitas ini bisa berupa objek, data, atau tugas-tugas yang perlu diproses oleh suatu sistem dalam urutan tertentu. Antrian digunakan di berbagai konteks, mulai dari sistem komputer dan jaringan, layanan pelanggan, hingga industri manufaktur.

2. Notifikasi

Notification atau Notifikasi adalah pemberitahuan mengenai informasi atau pengumuman dari pihak tertentu kepada pihak yang dituju yang dilakukan melalui media seperti email, sms, maupun aplikasi *chatting* (*line, whatsapp, BBM, wechat, dsb*). Notifikasi sangat penting bagi penerima informasi, karena dengan notifikasi, informasi tersebut dapat langsung diketahui oleh penerima. Notifikasi diperlukan ketika informasi yang akan disampaikan kepada pihak terkait tidak memungkinkan untuk diberitahukan secara langsung atau tatap muka. Notifikasi memungkinkan keakuratan atau ketepatan informasi yang diberikan. (James W, Elston D, 20 C.E.)

3. *WhatsApp*



Gambar 2. 1 Logo *WhatsApp*
(Sumber : *Google.com*)

WhatsApp berperan sebagai aplikasi *messenger online* yang saat ini banyak digunakan untuk berbagai hal seperti *chat* personal, komunitas, seminar *online*, dan sebagainya. *WhatsApp* merupakan sebuah aplikasi perpesanan (*messenger*) instan dan lintas *platform* pada *smartphone* yang memungkinkan pengguna mengirim dan menerima pesan seperti SMS tanpa menggunakan pulsa melainkan koneksi internet. *WhatsApp* memiliki *basic* yang mirip dengan *BlackBerry Messenger*. *WhatsApp* dirilis pada January 2009. Tetapi, pada tahun 2014, *WhatsApp* resmi menjadi milik *Facebook* setelah melalui proses akuisisi selama 8 bulan (Maulidda & Jaya, 2021).

Fitur-Fitur *WhatsApp*:

a) Pesan Teks

Fitur utama dari *WhatsApp* adalah kemampuannya untuk mengirim pesan teks. Pengguna dapat mengirim pesan ke pengguna lain yang memiliki aplikasi *WhatsApp* tanpa biaya apapun. Selain itu, pesan teks di *WhatsApp* juga dilengkapi dengan fitur seperti teks cetak tebal, miring, dan garis bawah, serta kemampuan untuk mengirim emotikon dan stiker.

b) Panggilan Suara dan Video

Selain pesan teks, *WhatsApp* juga memungkinkan pengguna untuk melakukan

panggilan suara dan video secara gratis dengan pengguna lain yang memiliki aplikasi *WhatsApp*. Fitur ini sangat membantu untuk mengurangi biaya komunikasi jarak jauh, terutama bagi pengguna yang berada di luar negeri.

c) *File* *Sharing*

WhatsApp juga memungkinkan pengguna untuk berbagi file dengan pengguna lain. Pengguna dapat mengirim gambar, video, dokumen, dan berbagai jenis file lainnya melalui aplikasi ini. Fitur ini sangat berguna bagi pengguna yang ingin mengirim file tanpa menggunakan email atau layanan file sharing lainnya.

d) *Grup* *Chat*

WhatsApp juga memungkinkan pengguna untuk membuat grup *chat* dengan beberapa orang sekaligus. Pengguna dapat membuat grup untuk keluarga, teman, atau bahkan untuk pekerjaan. Fitur ini memudahkan pengguna untuk berkomunikasi dengan kelompok orang yang sama tanpa harus mengirim pesan satu per satu.

e) *Keamanan*

WhatsApp telah menambahkan fitur keamanan yang kuat dalam aplikasinya, termasuk *end-to-end encryption*. Fitur ini memastikan bahwa pesan yang dikirimkan antara pengguna tidak dapat diakses oleh pihak ketiga, termasuk *WhatsApp* sendiri. Hal ini memberikan keamanan yang lebih baik bagi pengguna ketika berkomunikasi melalui aplikasi ini.

f) *WhatsApp* *Web*

WhatsApp juga tersedia sebagai aplikasi *web* yang memungkinkan pengguna untuk mengakses pesan mereka melalui *browser web* mereka. Ini

memungkinkan pengguna untuk mengirim dan menerima pesan dari komputer mereka tanpa perlu membuka aplikasi pada ponsel mereka. Fitur ini sangat berguna bagi pengguna yang ingin menggunakan *WhatsApp* pada layar yang lebih besar atau saat sedang bekerja di depan komputer.

4. Nontifikasi *Whatsapp*

Notifikasi *WhatsApp* adalah pesan atau pemberitahuan yang diterima oleh pengguna saat ada aktivitas baru di aplikasi *WhatsApp* mereka. Ini bisa berupa pesan teks, gambar, video, atau panggilan masuk. Notifikasi tersebut biasanya muncul di layar ponsel atau perangkat lainnya, serta bisa disertai dengan suara atau getar, tergantung pada pengaturan preferensi notifikasi pengguna.

5. *WhatsApp API Gateway*

WhatsApp API Gateway adalah aplikasi kirim pesan *WhatsApp* yang dimodifikasi menggunakan *API (application programming interface)* sehingga *whatsapp* tersebut bisa dicustom untuk membuat *chatbot*, *notifikasi* tambahan,dll. *WhatsApp API* merupakan aplikasi berbasis *Windows*, *macOS* dan *Linux* yang berjalan dengan mengotomatisasikan *Browser* untuk menjalankan *WhatsApp Web*. *WhatsApp Gateway* tentu saja bukan produk resmi yang pihak *WhatsApp*, namun produk ini merupakan ciptaan seseorang pengembang aplikasi.

Whatsapp Gateway mampu mengirim pesan ke ratusan nomor secara otomatis dalam waktu yang singkat (*broadcast* pesan), dan langsung terhubung dengan database nomor kontak yang ada di *Whatsapp*, tanpa mengetik ratusan nomor dan pesan di ponsel sebab semua nomor akan diambil secara otomatis dari

database, sehingga hal tersebut bisa menghemat waktu bekerja anda. Bukan hanya itu, *whatsapp api business* juga bisa difungsikan untuk notifikasi, *customer service*, *chatbot*, *verifikasi*, dll.

Kelebihan *WhatsApp API Gateway*

Mungkin sebagian dari anda bingung apa keuntungan yang akan didapatkan apabila menggunakan layanan *Whatsapp Api* ini. Nah, untuk memudahkan anda dalam memahami lebih dalam apa itu *WA Gateway*, berikut saya jabarkan beberapa hal yang bisa dilakukan oleh pengguna *WA Gateway*.

- a) Anda bisa mendapatkan *notifikasi* serta melakukan *verifikasi* apapun langsung melalui aplikasi *WhatsApp* anda, bukan melalui SMS maupun email
- b) Sebagai Pengguna, Anda bisa memperoleh beberapa informasi yang anda butuhkan melalui *WhatsApp*.
- c) *WA Gateway* memiliki fitur *bot WhatsApp* atau yang biasa dikenal dengan fitur auto respon (membalas pesan otomatis) yang mampu bekerja selama 24 jam non stop.
- d) Selayaknya aplikasi *Whatsapp Api* pada umumnya, *aplikasi WA Gateway* ini juga menawarkan fitur *Whatsapp Blast* yaitu fitur *API Whatsapp Business* yang memungkinkan Anda mengirim banyak pesan teks maupun gambar dalam satu waktu sekaligus. Anda juga bisa mengirim pesan tanpa menyimpan nomor pengguna di kontak. Fitur ini merupakan cara yang sangat *efektif* dan *efisien* bagi bisnis anda untuk menjangkau lebih banyak pelanggan.

- e) Bukan itu saja, aplikasi ini juga bisa memungkinkan Anda untuk bisa memberikan informasi yang akurat dan cepat untuk pengguna lainnya, sehingga menjadi pemasukan pasif bagi anda.

Kelebihan *WhatsApp Api Gateway - Whapi*

WhatsApp Business API - Whapi memiliki fitur menarik dan selalu update yang bisa memudahkan anda dalam menerima serta mengirim pesan bisnis

- a. *Aplikasi API* yang cepat, lengkap dan responsif
- b. Mengirim pesan ke berbagai grup hanya dalam hitungan detik tanpa harus menyimpan nomor maupun kontak yang akan dituju
- c. Laporan notifikasi status pengiriman pesan seperti notifikasi chat masuk, grup, baterai, status koneksi, dan masih banyak lagi
- d. *Rest API* yang terstruktur dan mudah di pahami, dan juga lengkap, mulai dari kirim pesan teks, media, lokasi, button, list, dokumen, dan masih banyak lagi.
- e. *Whatsapp Api multi device*, merupakan fitur terbaru dari *whapi* yang membuat *whatsapp gateway* anda selalu aktif 24 jam nonstop meski perangkat *handphone* Anda offline, Anda juga bisa membuka beberapa *whatsapp web* dengan fitur terbaru ini.
- f. *Fitur Auto Reply* anda bisa membuat *chatbot* lebih *interaktif* dengan cara membalas pesan secara otomatis sesuai dengan *keyword* tertentu.
- g. Panel admin yang bisa memantau seluruh aktivitas *Whatsappapi* secara *realtime* serta mengontrol penuh koneksi antara perangkat dan juga aplikasi.

- h. *Whapi* merupakan aplikasi yang *dedicated*, hal yang dimaksud adalah anda bisa menginstall *API WhatsApp Gateway* di *PC, RDP, VPS*, maupun *hosting*. *Source WHAPI* bisa Anda gunakan sepenuhnya, dan tidak perlu dibagi.
- i. Aplikasi ini dapat diinstal di berbagai sistem operasi, baik itu *Windows, Linux*, maupun *MacOS*. *Instalasi* dan *maintenance WHAPI* sangat mudah dan dilengkapi dengan tutorial yang mudah dipahami.
- j. Anda tidak perlu khawatir lagi dengan kuota pengiriman pesan teks, media, lokasi, maupun pesan lainnya. Sebab aplikasi *WHAPI* tidak terbatas, artinya anda bisa mengirimkan berapa jumlah pesannya tanpa ada batasan
- k. Dokumentasi *API* dan Tutorial Instalasi *API* lengkap
- l. Admin diskusi tanya-jawab *fast respon*

6. *Fonnte*

Fonnte.com adalah platform layanan *WhatsApp API Gateway* yang digunakan untuk mengirim pesan otomatis melalui *WhatsApp*. Platform ini mendukung berbagai kebutuhan bisnis, seperti *blast message, autoresponder*, dan integrasi dengan layanan lain seperti *Google Forms* dan *spreadsheet*.

Fitur Utama:

- a) **Pengiriman Pesan Otomatis:** Anda dapat mengirim pesan ke nomor individu atau grup, dengan opsi penjadwalan dan pengulangan pesan.
- b) **Autoreply dan Chatbot:** Fitur ini memungkinkan Anda membuat balasan otomatis berdasarkan kata kunci tertentu. Sangat berguna untuk layanan pelanggan.

- c) **Integrasi API:** Fonnte mendukung penggunaan API untuk menghubungkan layanan eksternal seperti aplikasi berbasis PHP atau platform lainnya.
- d) **Pesan Berbasis Data:** Dapat mengirim pesan dari Google Forms atau spreadsheet langsung ke WhatsApp menggunakan skrip kustom **【6】 【7】 【8】** .

Cara Memulai:

1. **Daftar Akun:** Daftarkan akun di Fonnte.
2. **Hubungkan Perangkat:** Tambahkan perangkat melalui menu *device* di dashboard.
3. **Kirim Pesan:** Akses menu pengiriman pesan, isi formulir, dan tentukan target pengiriman.
4. **Buat Chatbot (Opsional):** Jika diperlukan, Anda bisa membuat chatbot dengan fitur *autoreply* **【6】 【7】** .

7. Framework

Framework merupakan suatu kerangka kerja yang dipakai dalam pengembangan situs *web*. Kerangka kerja ini dirancang untuk memberikan dukungan kepada pengembang *web* dalam menyusun baris kode. Penggunaan *framework* mempermudah, mempercepat, dan mengorganisir penulisan kode secara sistematis.

Framework adalah kerangka kerja atau struktur konseptual yang menyediakan fondasi untuk membangun sesuatu, seperti aplikasi perangkat lunak. Ini adalah kumpulan alat, aturan, dan komponen yang tersusun secara sistematis untuk membantu pengembang dalam membuat, mengelola, dan mengoptimalkan

aplikasi dengan lebih efisien. *Framework* menyediakan abstraksi yang berguna untuk menangani tugas-tugas umum, memungkinkan pengembang untuk fokus pada logika aplikasi yang khusus.

Ada berbagai jenis *framework*, termasuk:

- a. **Framework Web:** Digunakan untuk membangun aplikasi *web*. Contohnya adalah *Angular*, *React*, dan *Vue.js*.
- b. **Framework Backend:** Digunakan untuk membangun bagian belakang aplikasi *web* (*server-side*). Contohnya adalah *Express.js* untuk *Node.js*, *Django* untuk *Python*, dan *Laravel* untuk *PHP*.
- c. **Framework Frontend:** Berfokus pada pengembangan antarmuka pengguna (*UI*) aplikasi *web*. Contohnya adalah *Bootstrap* dan *Foundation*.
- d. **Framework Mobile:** Digunakan untuk membangun aplikasi *mobile*. Contohnya adalah *React Native* untuk aplikasi *mobile* berbasis *JavaScript* dan *Flutter* untuk aplikasi *mobile* berbasis *Dart*.
- e. **Framework Desktop:** Digunakan untuk membangun aplikasi *desktop*. Contohnya adalah *Electron*, yang memungkinkan pengembang menggunakan *HTML*, *CSS*, dan *JavaScript* untuk membangun aplikasi *desktop* lintas *platform*.

Keuntungan menggunakan *framework* meliputi:

- a. **Productivity:** *Framework* menyediakan alat dan struktur yang sudah jadi sehingga pengembang dapat membangun aplikasi lebih cepat.
- b. **Consistency:** *Framework* mempromosikan standar pengembangan yang konsisten, membantu dalam memelihara dan memperbarui aplikasi.

- c. **Security:** Banyak *framework* memiliki fitur keamanan bawaan atau praktik terbaik yang membantu melindungi aplikasi dari serangan keamanan.
- d. **Scalability:** Dengan pola desain yang baik, *framework* dapat memudahkan skalabilitas aplikasi saat kebutuhan meningkat.

8. *CodeIgniter*



Gambar 2. 2 Logo *CodeIgniter*
(Sumber : *Google.com*)

CodeIgniter merupakan aplikasi *open source* berupa *framework PHP* untuk membuat *website* lebih dinamis menggunakan *PHP* dengan pola *MVC* (Model, View, Controller). *CodeIgniter* memudahkan pengembang *web* untuk membuat dan membangun aplikasi *web* dengan cepat dari awal. Selain membuat *web* menjadi lebih dinamis, proses ini juga dapat membantu pengembang membangun aplikasi *web* yang ringan dan cepat. *CodeIgniter* mempunyai dokumentasi dengan contoh implementasi kode yang sangat lengkap, Dokumentasi yang lengkap ini menjadi salah satu alasan kuat mengapa banyak orang memilih *CodeIgniter* sebagai pilihannya. Terlepas dari kelebihan *CodeIgniter*, banyak orang yang belum menggunakannya karena mereka tidak cukup tahu tentang *framework CodeIgniter* (Ikhsan et al., 2024).

9. Framework *CodeIgniter*

Framework *Codeigniter* merupakan sebuah *web application network* yang bersifat *open source*. Dimana *Codeigniter* merupakan salah satu *Framework*

pemrograman *PHP (Hypertext Preprocessor)* dengan model *MVC (Model, View, Controller)* untuk membangun *website* dinamis yang dapat mempercepat proses pembuatan dan pengembangan aplikasi *web*. Selain ringan dan cepat, *Codeigniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. (Ahmad Martani et al., 2022).

10. Web

Website merupakan sebuah media informasi yang ada di internet. *Website* tidak hanya dapat digunakan untuk penyebaran informasi saja melainkan bisa digunakan untuk membuat toko *online*. *Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau *subdomain*, yang tempatnya berada di dalam *World Wide Web (WWW)* di *Internet*. Sebuah halaman *web* adalah dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)*, yang hampir selalu bisa diakses melalui *HTTP*, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar (Yunita Trimarsiah & Muhajir Arafat, 2017).

Website pada awalnya adalah sebuah platform yang memberikan informasi kepada pengguna internet dengan menggunakan konsep *Hyperlink*, sehingga mempermudah pengguna untuk menjelajahi informasi di dunia maya. Informasi yang dipersembahkan melalui internet menggunakan media yang beragam, seperti tulisan, foto, animasi, audio, atau video.

Pengertian *website* menurut para ahli :

- a. Wahana Komputer, *web* merupakan jenis komunikasi *interaktif* yang dipakai dalam sistem komputer yang terhubung.
- b. Taufiq Hidayatullah menyatakan bahwa *web* merupakan komponen yang paling terlihat dari jaringan terbesar di dunia, yaitu internet.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Belajar HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Gambar 2. 3 Pemrograman Web
(Sumber : *Google.com*)

11. PHP (*HypertextPre-processor*)



Gambar 2. 4 PHP (*HypertextPre-processor*)
(Sumber : *Google.com*)

PHP adalah sebuah bahasa pemrograman berbasis *web* yang mampu mengolah data secara dinamis. *PHP* dikenal sebagai bahasa skrip yang disertakan di sisi *server*, yang berarti semua *sintaks* dan perintahnya dieksekusi oleh *server* namun dapat dimasukkan ke dalam halaman *HTML* biasa. Aplikasi yang dikembangkan menggunakan *PHP* biasanya menampilkan hasilnya melalui *web browser*, tetapi proses sebagian besar dijalankan oleh *server*.

PHP dikenalkan oleh *Rasmus Lerdof* pada tahun 1995. Pada mulanya, *PHP* disebut *Personal Homepage*. Alasannya adalah karena *PHP* awalnya adalah sebuah program kecil yang dirancang untuk menambah fitur pada situs personal di *Rasmus*.

```
<html>
<head>
  <title> Contoh Sederhana </title>
</head>
<body>
  <?php
    echo("Hallo apa kabar? Nama saya PHP
script");
  ?>
</body>
</html>
```

Gambar 2. 5 Personal di *Rasmus*
(Sumber : *Google.com*)

12. *MySQL (My Structured Structured Query Language)*



Gambar 2. 6 Logo *MySQL*
(Sumber : *Google.com*)

Menurut Rulianto Kurniawan (2010 :16) *MySQL* merupakan suatu jenis *database server* yang sangat terkenal. *MySQL* termasuk jenis *RDBMS (Relational Database Manajement System)*. *MySQL* mendukung bahasa pemrograman *PHP*, bahasa permintaan yang terstruktur, karena pada penggunaannya *SQL* memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama *ANSI*. *MySQL* merupakan *RDBMS (Relational Database Management System) server*.

RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model *relational*.

Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari *MySQL* yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya. *MySQL* lebih empat tiga sampai empat kali dari pada *database server komersial* yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL*.
- b. Didukung oleh berbagai bahasa *Database Server MySQL* dapat memberikan pesan Error dalam berbagai bahasa seperti *Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia*.
- c. Mampu membuat tabel berukuran sangat besar. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
- d. Lebih murah *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX platform, OS/2 dan Windows Platform*. Melekatnya integrasi *PHP* dengan *MySQL*. Keterikatan antara *PHP* dengan *MySQL* yang sama-sama *Software Open-Source* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya. Modul *MySQL* di *PHP* telah dibuat *Built-in* sehingga tidak memerlukan konfigurasi tambahan pada File konfigurasi *Php* ini (Hermiati et al., 2021).

13. *Cascading Style Sheets (CSS)*

CSS adalah bahasa *style sheet* dan teknologi *web* yang menggambarkan *presentasi visual situs web*. Dengan internet yang berkembang pesat teknologi dalam dekade terakhir dan perkembangan teknologi yang konstan memungkinkan terciptanya *situs web* yang mengesankan dan membantu mengurangi jumlah pekerjaan. Seiring berkembangnya teknologi, kinerja harus selalu dipertimbangkan. Kinerja dalam konteks ini berarti berapa lama dibutuhkan untuk memuat konten halaman *web*, kapan konten menjadi *responsif* dan bagaimana caranya dapat diakses dan memperlancar halaman *web* selama penggunaan. Kinerja CSS sangat berarti dari total waktu buka yang diperlukan perhitungan gaya dan tata letak selain pembuatan *skrip*, pemuatan, dan lukisan (Kuparinen, 2023). Sedangkan Menurut Bunafit Nugroho (2013:1), *Cascading Style Sheet* adalah *style sheet* bahasa yang digunakan untuk mengatur tampilan suatu dokumen tertulis dalam bahasa markup. CSS berfungsi sebagai pelengkap elemen *HTML* yang semuanya dapat dikontrol dengan menggunakan bahasa *scripting CSS*. Penggunaan CSS dilakukan untuk memperluas kemampuan *HTML* dalam memformat dokumen *website* atau mempercantik tampilan *website*. Penulisan kode CSS disisipkan dalam *HTML* bukan. Kode CSS ditulis tanpa `<style>` dan `</style>` dengan mendefinisikan *style* baru maka dapat digunakan berulang kali. (Annafri et al., 2023).

Salah satu contoh CSS adalah untuk mengatur format paragraf. Misalkan ingin agar semua paragraf dalam sebuah halaman *website* berukuran 120% dan berwarna abu tua, Anda cukup menambahkan kode di bawah ini:

```
p { font-size: 120%; color: dimgray; }.
```

Anda juga bisa mengubah warna link dengan CSS. Sebagai catatan, warna link yang ditentukan dengan CSS ada empat, yaitu:

Normal: warna link yang belum pernah dibuka dan tidak diklik

Visited: warna link yang sudah pernah dibuka

Hover: warna ketika Anda meletakkan kursor di atas link

Active: warna link ketika Anda klik

Untuk menentukan keempat warna itu, gunakan kode di bawah ini dan ketikkan warna yang Anda inginkan setelah **color:**.

```
a:link { color: gray; } a:visited { color: green; } a:hover { color: purple; } a:active { color: teal; }
```

14. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain (dikutip dari Dosen Online, 2017). Fungsi *ERD* adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat. (Afifah et al., 2022).

Di dalam *ERD* terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi.

1. Entitas

Entitas merupakan objek yang akan menjadi perhatian dalam suatu database.

Entitas dapat berupa manusia, tempat, benda, atau kondisi mengenai data yang dibutuhkan. Simbol dari entitas berbentuk persegi panjang.

2. Atribut

Atribut merupakan informasi yang terdapat dalam entitas. Sebuah entitas

harus memiliki primary key sebagai ciri khas entitas dan atribut deskriptif. Atribut biasanya terletak dalam tabel entitas atau dapat juga terpisah dari tabel. Simbol dari atribut berbentuk elips.

3. Relasi

Relasi di dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat. Relasi yang dapat dimiliki oleh ERD ada beberapa macam, yaitu:

a) *One* *to* *One*

Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.

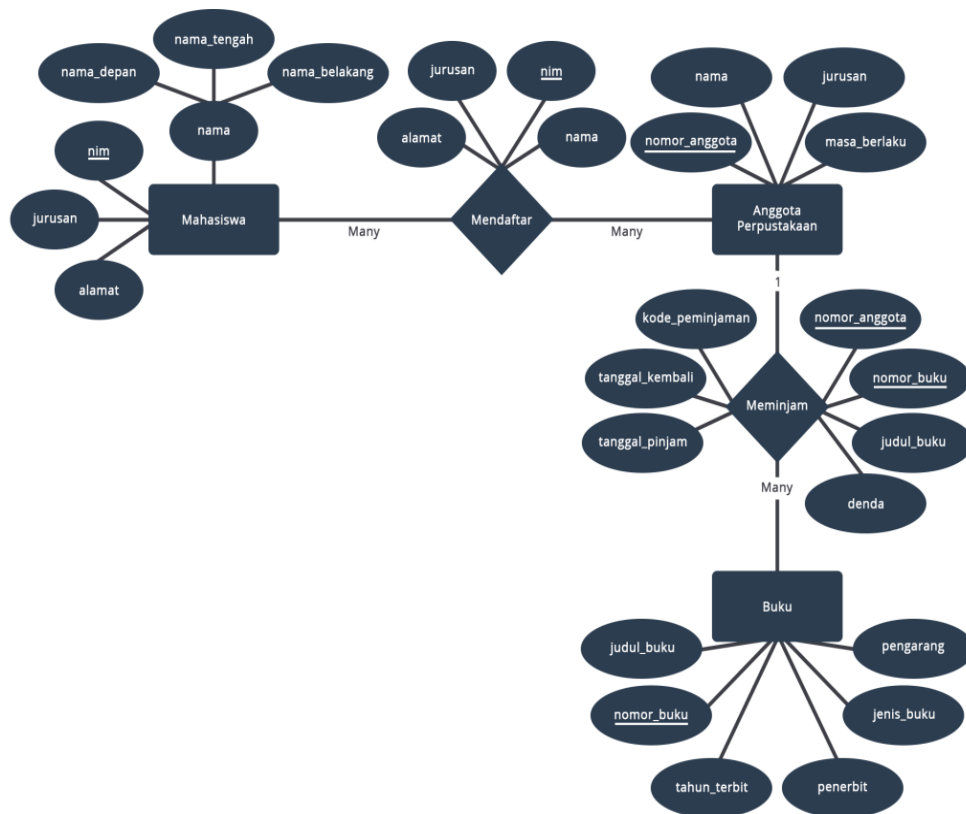
b) *One* *to* *Many*

Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

c) *Many* *to* *Many*

Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain ('Afiifah et al., 2022).

Contoh dari ER diagram sistem perpustakaan. Berikut adalah contohnya:



Gambar 2. 7 Contoh ERD
(Sumber : Google.com)

Jadi pada contoh ER diagram di atas menjelaskan terdapat entitas, atribut, dan proses apa saja yang terjadi pada saat mahasiswa ingin meminjam buku dari perpustakaan. ER diagram tersebut dapat dijadikan gambaran untuk membuat sebuah sistem database perpustakaan.

Dapat dilihat pada diagram perpustakaan di atas terdapat empat entitas di dalamnya. Keempat entitas tersebut adalah mahasiswa, anggota perpustakaan, buku, dan denda. Setiap entitas memiliki atributnya masing-masing. Berikut adalah atribut dari masing-masing entitas.

Tabel 2. 1 Contoh *EDR*

Entitas	Atribut
Mahasiswa	Nim, nama_jurusan, alamat
Anggota Perpustakaan	Nomor_anggota, nama, jurusan, masa_berlaku
Buku	Nomor_buku, judul_buku, pengarang, penerbit, tahu_terbit, jenis_buku

Pada setiap relasi dari ER diagram di atas juga mengandung beberapa atribut yang digunakan dalam prosesnya. Seperti contohnya saat mahasiswa ingin mendaftar menjadi anggota perpustakaan maka proses yang terjadi adalah mahasiswa diminta datanya seperti NIM, nama, jurusan, dan alamat untuk didaftarkan menjadi anggota perpustakaan.

15. *Unified Modelling Language (UML)*

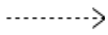
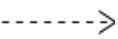
UML adalah bahasa pemodelan populer yang memiliki visualisasi sistem dan kinerja dokumentasi yang baik. Pemodelan UML bahkan dapat menghasilkan kode-kode pemrograman yang siap diimplementasikan. UML memiliki fungsi untuk membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. UML diciptakan dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir tahun 1980-an dan awal tahun 1990-an. Penggunaan UML tidak terbatas untuk metodologi tertentu, walaupun pada kenyataannya UML ini paling banyak digunakan pada metodologi berorientasi objek.

Jenis-jenis Diagram *Unified Model Language (UML)* yaitu *use case* diagram berfungsi untuk mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibangun, dan dapat menggambarkan

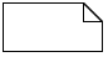
fungsi apa saja yang ada pada sebuah sistem informasi, *class* diagram merupakan pembentuk utama dari sistem berorientasi objek yang mempresentasikan suatu class beserta dengan atribut dan operasinya, *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Perbedaan *use case* diagram dan *activity* diagram adalah untuk *use case* menggambarkan bagaimana aktif berjalan atau *actor* Ketika menggunakan sistem untuk melakukan aktivitas. Sedangkan *activity* diagram menggambarkan aktifitas sistem bukan apa yang dilakukan oleh aktor. (Nistrina & Sahidah, 2022).

Adapun daftar simbol *UML* yaitu :

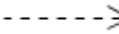
Tabel 2. 2 Simbol *Use Case Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Mendefinisikan serangkaian tindakan pengguna saat berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Suatu hubungan di mana perubahan pada elemen tertentu mempengaruhi elemen dependen.
3		<i>Generalization</i>	Hubungan dimana objek anak mewarisi perilaku dan struktur data dari objek yang berada di level hierarki di atasnya objek induk..
4		<i>Include</i>	Mengklarifikasi bahwa sumber <i>use case</i> secara jelas ditentukan.
5		<i>Extend</i>	Menjelaskan bahwa <i>use case</i> target menambahkan perilaku dari <i>use case</i> sumber pada titik tertentu.
6		<i>Association</i>	Yang menggabungkan dua objek menjadi satu.
7		<i>System</i>	Mengidentifikasi paket yang memiliki tampilan sistem terbatas.
8		<i>Use Case</i>	Penjelasan mengenai langkah-langkah yang diambil oleh sistem untuk

Lanjutan Tabel 2.2

No	GAMBAR	NAMA	KETERANGAN
			menyebabkan seorang <i>actor</i> mengalami hasil yang dapat diukur
9		<i>Collaboration</i>	Cara hukum dan komponen lainnya berinteraksi untuk menghasilkan perilaku yang lebih kompleks daripada gabungan bagian-bagiannya (sinergi).
10		<i>Note</i>	Komponen fisik yang berfungsi sebagai sumber daya komputer dan hadir ketika suatu program dijalankan.

Tabel 2. 3 Simbol *Class Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Mencoba menjauhi koneksi dengan lebih dari dua objek.
3		<i>Class</i>	Pengelompokan item dengan karakteristik dan fungsi yang serupa.
4		<i>Collaboration</i>	Penjelasan mengenai tindakan-tindakan yang terurut yang dilakukan oleh sistem untuk menghasilkan hasil yang dapat diukur untuk seorang pelaku
5		<i>Realization</i>	Operasi yang sesungguhnya dijalankan oleh suatu entitas.
6		<i>Dependency</i>	Suatu hubungan dimana perubahan yang terjadi pada suatu unsur yang berdiri sendiri mempengaruhi unsur-unsur yang bergantung pada unsur yang tidak bebas tersebut.
7		<i>Association</i>	Sesuatu yang menghubungkan satu objek dengan objek lainnya

Tabel 2. 4 Simbol *Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi komunikasi antar item yang menyimpan data tentang peristiwa yang terjadi
3		<i>Message</i>	Spesifikasi komunikasi antar item yang menyimpan data tentang peristiwa yang terjadi

Tabel 2. 5 Simbol *State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut yang dimiliki suatu objek pada saat tertentu dan nilai tautan.
2		<i>Initial Pseudo State</i>	Bagaimana segala sesuatunya berasal atau terbentuk
3		<i>Final State</i>	Bagaimana sesuatu diciptakan dan dimusnahkan
4		<i>Transition</i>	Peristiwa yang mengubah satu atau lebih nilai atribut suatu objek sehingga menyebabkan objek tersebut berubah statusnya
5		<i>Association</i>	Apa yang menggabungkan dua objek menjadi satu.
6		<i>Node</i>	Komponen fisik yang berfungsi sebagai sumber daya komputer dan hadir ketika suatu program dijalankan.

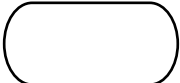
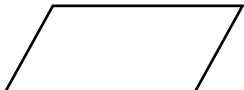
Tabel 2.6 Simbol *Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	mendemonstrasikan interaksi antara setiap kelas antarmuka.
2		<i>Action</i>	Status sistem seperti yang muncul setelah tindakan selesai.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau dimulai.
4		<i>Activity Final Node</i>	Bagaimana sesuatu diciptakan dan dimusnahkan
5		<i>Fork Node</i>	Satu aliran yang pada akhirnya terpecah menjadi beberapa aliran

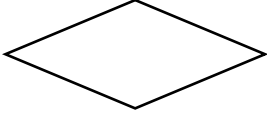
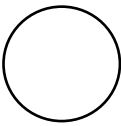
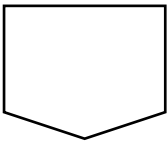
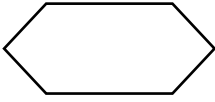
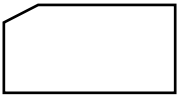
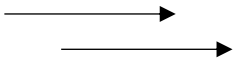
16. Flowchart

Menurut (Arifianto) “*Flowchart* adalah representasi grafis menggunakan simbol-simbol tertentu yang menunjukkan hubungan antara suatu instruksi dan operasi lain dalam suatu program serta urutan proses secara menyeluruh.

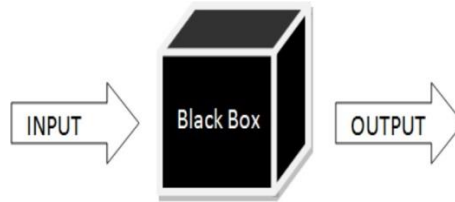
Tabel 2.7 Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		Terminal	Menyatakan kapan suatu program dimulai dan dihentikan.
2		<i>Input/Output</i>	Menunjukkan proses masukan atau keluaran, apa pun jenis peralatannya.
3		<i>Process</i>	Menjelaskan tindakan atau prosedur yang dihasilkan komputer.

Lanjutan Tabel 2.7

No	Simbol	Nama	Fungsi
4		<i>Decision</i>	menunjukkan keadaan tertentu yang akan menghasilkan salah satu dari dua tanggapan alternatif: ya/tidak
5		<i>Connector</i>	Menunjukkan bagaimana satu aktivitas terhubung ke aktivitas lain di halaman yang sama.
6		<i>Offline Connector</i>	Mengidentifikasi hubungan antara satu proses dan proses lainnya pada halaman terpisah.
7		<i>Prefefned Process</i>	Menyatakan ketersediaan ruang penyimpanan untuk pemrosesan guna menetapkan biaya awal.
8		<i>Punched card</i>	Menunjukkan apakah masukan ditulis ke kartu atau berasal dari kartu.
9		<i>Document</i>	Dengan menggunakan printer, cetak hasilnya dalam bentuk dokumen.
10		<i>Flow</i>	Menyatakan bagaimana suatu prosedur berjalan

17. *Blackbox testing*



Tabel 2. 8 *Black Box Testing*
(Sumber : *Google.com*)

Pengujian *black box* merupakan metode pengujian yang melibatkan pengamatan hasil dari eksekusi perangkat lunak dengan menggunakan data uji, tanpa membuka dan memeriksa secara detail struktur internal dari perangkat lunak tersebut. Metode ini lebih fokus pada pengujian *fungsi* dari perangkat lunak. Kita bisa membandingkannya dengan melihat sebuah kotak yang dilapisi warna hitam. Kita hanya bisa melihat bagian luar kotaknya tanpa mengetahui apa yang ada di dalamnya. Sama halnya dengan uji *black box*, mengevaluasi berdasarkan hanya pada tampilan luar (antarmuka), *fungsi* tanpa mengetahui proses *internal* yang sebenarnya terjadi (hanya mengetahui masukan dan keluaran).

Berikut dari bagian-bagian dalam teknik yang digunakan dalam *Black Box Testing* antara lain :

- a. Berfungsi untuk menguji fitur-fitur khusus dari perangkat lunak..
- b. Ketika menguji perangkat lunak, kebenaran hanya dinilai berdasarkan hasil yang dihasilkan olehnya.

Kemampuan sebuah program untuk memenuhi kebutuhan pengguna dapat dinilai sambil juga mengidentifikasi kesalahan-kesalahannya.

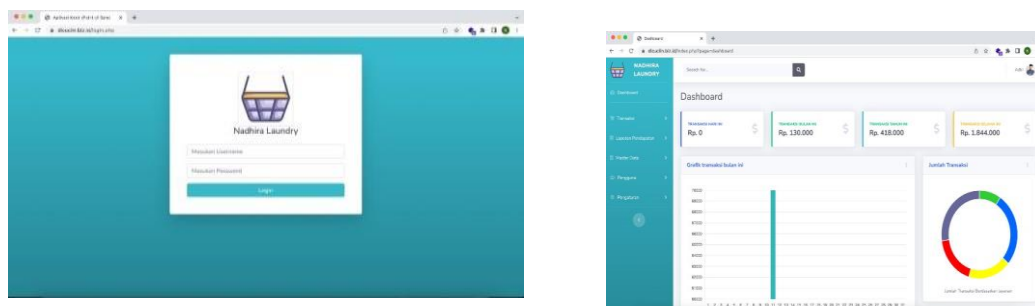
Berikut adalah contoh *blackbox testing* Studi Kasus Bisnis Jasa Laundry .Sistem informasi *login* dan *logout* diimplementasikan untuk tujuan pengujian guna mengetahui keberadaan sistem sebelum di-*deploy* sistem. Hasil pengujian Jika kerentanan ditemukan di bagian sistem, pengguna dapat mengembangkan lebih lanjut bagian sistem yang tergolong lemah. Pengujian dilakukan berdasarkan pengujian metode *black box* yaitu: (Rahayu, 2020).

1. Memasukkan data yang benar, misalnya sistem menerima data input yang tersimpan di *database*.
2. Data *input* adalah data randomyg berguna memastikan bahwa sistem menolak untuk menyimpan data *input* ke *database*.

Hasil pengujian memiliki peta kendali untuk dapat menyimpulkan apakah sistem berhasil diuji jenis ini atau tidak. Berikut beberapa penjelasan dari tabel uji yang akan digunakan:

1. *Input* adalah interpretasi untuk memasukkan data *real* atau data acak.
2. Hasil yang diharapkan adalah hasil yang ingin dicapai selama tes.
3. Hasil adalah hasil pengujian setelah sistem diuji.
4. Kesimpulan apakah sistem lulus atau tidak.

Pengujian Form *Login*

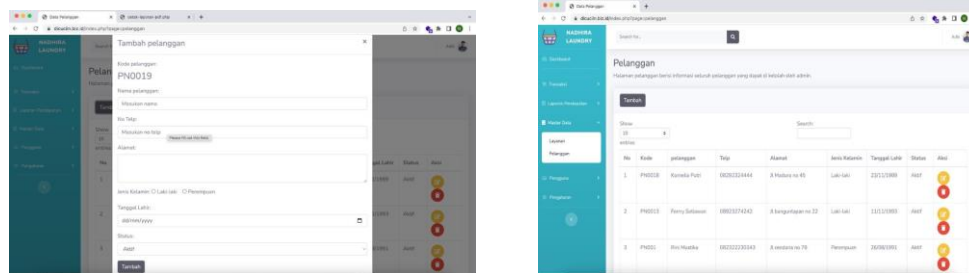


Gambar 2. 8 Contoh *Blackbox Testig* Pengujian Form *Login*
(Sumber : Google.com)

Tabel 2. 9 Contoh *Blackbox Testig* Pengujian *Form Login*

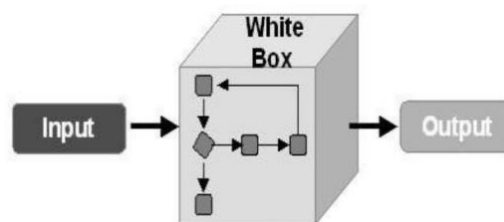
No	Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1.	Saat Klik Tombol <i>Login</i>	<i>Username</i> : super admin laundry <i>Password</i> : 12345	Masuk Ke Menu	Valid

Pengujian Tambah Data Pelanggan

**Gambar 2. 9** Contoh *Blackbox Testig* Pengujian Tambah Data
(Sumber : Google.com)**Tabel 2. 10** Contoh *Blackbox Testig* Pengujian Tambah Data

No	Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1.	Saat Klik Tambah Pelanggan	Memasukkan Nama, No telpon, alamat, jenis kelamin, tanggal lahir, status	Berhasil Tambah Pelanggan	Valid
2.	Saat klik Tambah Pelanggan	Tidak memasukkan data salah satu form	Gagal menambah data pelanggan	Tidak Valid

18. Whitebox testing

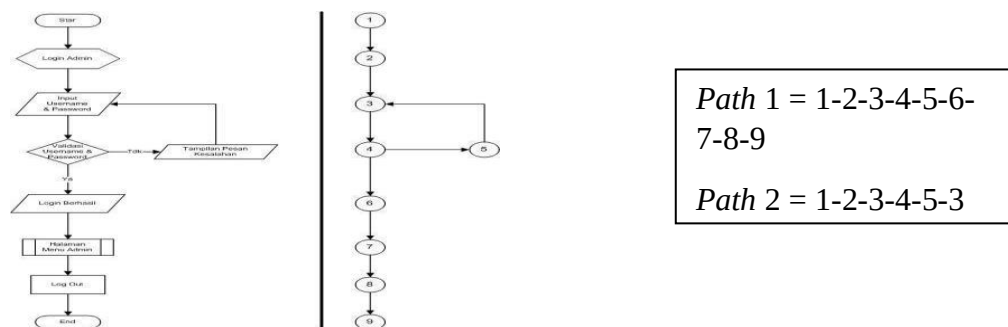
**Tabel 2. 11** *White Box Testing*
(Sumber : Google.com)

Uji coba *white box testing* adalah metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk

meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di test dengan kondisi dan perulangan secara fisik.

Dalam studi kasus pengujian *white box* ini, pengguna diperingatkan ketika mereka memasukkan kata sandi pengguna yang salah dan diberikan informasi tentang kesalahan yang mereka buat.

Berikut ini adalah contoh pengujian *white box* pada sistem *login* yang terdapat pada Sistem Informasi laporan Data Statistik Desa Pananrang berbasis Web. (Asmil. (2020)), yaitu :



Gambar 2. 10 Contoh Whitebox Testig
(Sumber : Google.com)

Perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki region =

2. menghitung *cyclomatic complexity* dari *edge* dna *node*, dengan rumus :

$$V(G) = E - N + 2$$

$$E(\text{edge}) = 9, N(\text{node}) = 9$$

$$\text{Penyelesaian : } V(G) = 9 - 9 + 2$$

$$V(G) = 2$$

Jadi jumlah path pada *flowgraph* diatas adalah 2. *Independent path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

Path 2 = 1 - 2 - 3 - 4 - 5 - 3

19. Hosting

Hosting adalah tempat fisik dimana semua isi atau *content* sebuah *website* disimpan di dalamnya. *Hosting* membutuhkan harddisk yang terletak pada pemilik penyedia *hosting*. *Hosting* dapat diakses dan digunakan oleh pemilik *website*. *Hosting* digunakan oleh pemilik *website* untuk mengelola *CMS* atau *Content Management System*.

Hosting adalah jasa layanan internet yang menyediakan sumber daya *server-server* untuk disewakan sehingga memungkinkan organisasi atau individu menempatkan informasi di internet berupa *http*, *ftp*, email, (seluruh file yang telah dibuat dan akan *diupload*)

Ketika membeli *hosting*, yang sebenarnya si pembeli adalah menyewa tempat atau *space* di *server* untuk menempatkan atau menyimpan file-file *CMS* dan data-data lainnya sehingga *website* dapat menampilkan objek-objek yang sesuai dengan keinginan pemilik *website*.

CMS atau *Content Management System* adalah aplikasi untuk *website* yang digunakan dalam mengelola file seperti *upload*, edit, menambahkan, dan lain-lain). Beragam *CMS* atau *Content Management System* adalah *Wordpress*, *Joomla*, *Plone*, *textpattern*, *movabletype*, *commerce*, *yp03*, *Prestashop*, *Oxid*, *DruaMo*, *dx*, *Zikula*, *Magento*, *Rails*, *Simplemachines*, *Bulletin*, *Mediawiki*, *phpBB*, *SPIP*, *ZendFramework*, *RoundCube*, *Symfony*, *Mantis*, *Sonar*, *Status.net*, *Silverlight*, *trac*, *Jenkins*, *CodeIgniter*, *django*, *Java*, *PHP.C*

Content Management System yang digunakan dalam pembahasan ini adalah *Wordpress*. *Wordpress* adalah sebuah aplikasi sumber terbuka (*open source*) yang sangat populer digunakan sebagai mesin *blog (blog engine)*. *Wordpress* dibangun dengan bahasa pemrograman *PHP* dan basis data (*database*) *MySQL*, *PHP* dan *MySQL* (Yosli, 2021).

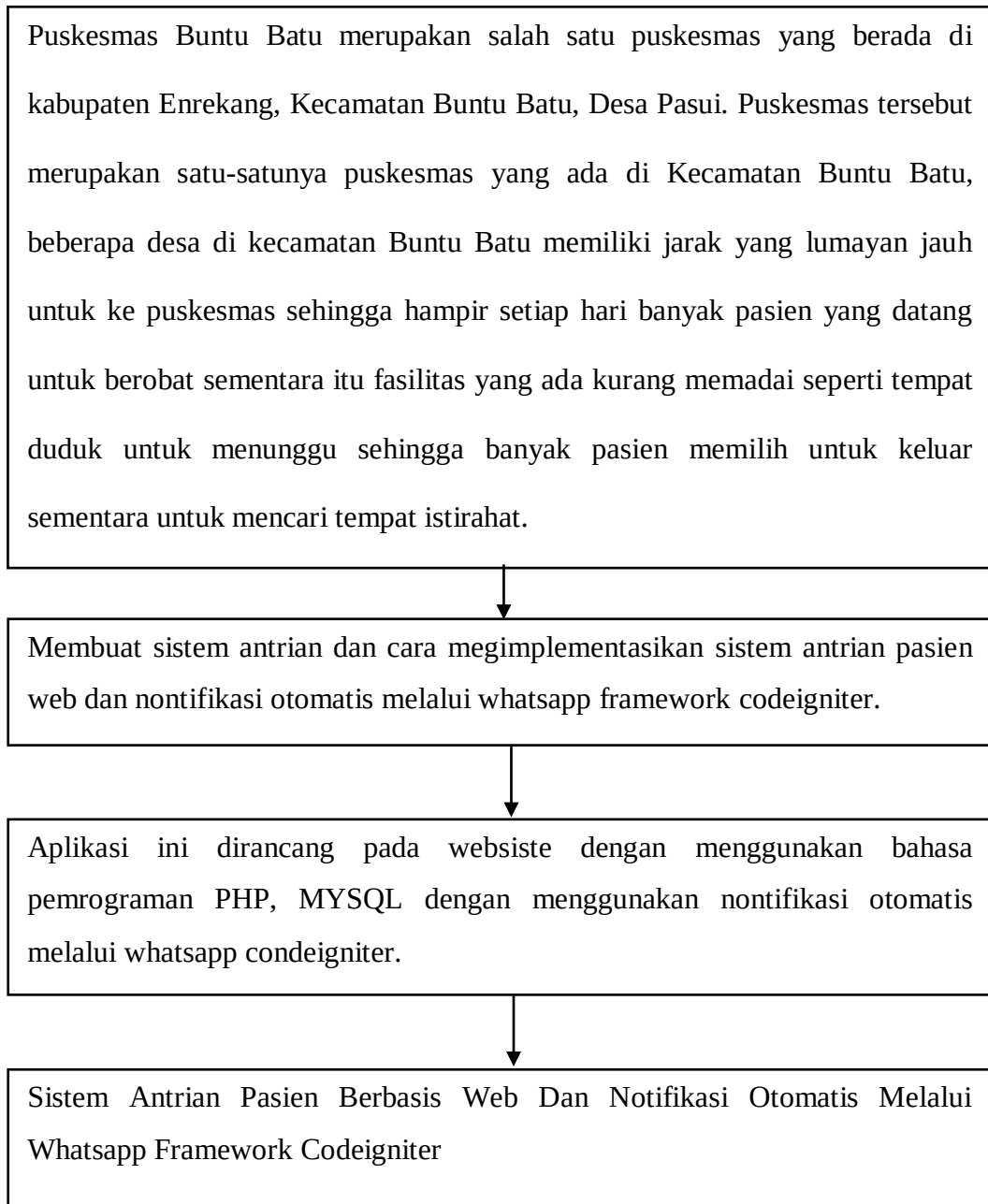
B. Kajian Hasil Penelitian Terdahulu

Beberapa penelitian terkait dengan tugas akhir dari penulis antara lain sebagai berikut:

- 1) (Febriyanto et al., 2023) dengan judul “Penerapan *Whatsapp Notification* Pada Sistem Pendaftaran *Online* Klinik Pku Muhammadiyah Kitamura Pontianak”. Telah berhasil dibangun Sistem informasi pendaftaran di klinik PKU Muhammadiyah Kitamura Pontianak dengan menerapkan *Whatsapp notification*. *Whatsapp notification* digunakan sebagai media oleh pasien untuk melakukan pendaftaran dengan memberikan *notifikasi real time* ketika admin operator klinik sudah *memverifikasi* pendaftaran yang diajukan pasien selain itu pasien bisa mengetahui secara langsung proses pendaftaran yang telah diajukan. Pengujian antarmuka sistem dilakukan oleh pasien dan *admin operator* untuk melihat kesesuaian Sistem yang dibangun dan kebutuhan pengguna. Berdasarkan hasil pengujian sistem dikategorikan sangat baik dengan nilai akhir 88,42%.
- 2) (Putri & Ratnasari, 2021) dengan judul “Perancangan Aplikasi Mobile untuk Sistem Antrean Rumah Sakit Puri Husada Yogyakarta”. Menyatakan sistem antrean merupakan layanan antrean yang dapat membantu pasien dalam

melakukan pendaftaran dan mendapatkan nomor antrean. Pada literatur yang diulas, mayoritas berupa sistem berbasis *mobile* dan memiliki fitur *registrasi*, nomor antrean, jadwal dokter/poliklinik, dan cetak nomor antrean. Fitur yang paling sedikit digunakan pada literatur terkait adalah fitur pemilihan jaminan kesehatan, jumlah pegawai, dan jumlah pengguna. Berdasarkan hasil analisis dan kajian literatur yang penulis lakukan, fitur yang akan dibangun pada sistem antrean Rumah Sakit Puri Husada Yogyakarta adalah fitur *registrasi*, *login*, *perjanjian/appointment* (pemilihan poliklinik, pemilihan jaminan kesehatan), fitur dokter yang berisikan jadwal dokter, dan daftar poliklinik, daftar antrean, serta notifikasi antrean. Penelitian ini menggunakan metode *waterfall*, yang mana berkembang secara sistematis dan berurutan. Sistem yang akan dibangun menggunakan *Visual Studio Code*, *MySQL* sebagai basis data, *Figma* sebagai pembuatan rancangan antarmuka, dan menggunakan *framework Flutter*.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian kualitatif merupakan metode penelitian yang bertujuan memahami fenomena sosial secara mendalam melalui data deskriptif, seperti wawancara, observasi, dan analisis dokumen. Pendekatan ini bersifat subjektif, *kontekstual*, dan *fleksibel*, dengan analisis yang dilakukan secara induktif.

B. Lokasi dan Waktu Penelitian

1. Lokasi : Puskesmas BuntuBatu,Provinsi sulawesi Selatan, Kabupaten Enrekang,Kecamatan Buntu Batu,Desa Pasui.
2. Waktu Penelitian : 2 (dua) Bulan.

C. Meode Pengumpulan Data

1. Studi Kasus

Pengujian studi kasus adalah metode penelitian yang digunakan untuk mengevaluasi suatu sistem, proses, atau fenomena dalam konteks nyata.

2. Observasi

Merupakan teknik pengumpulan data yang dilakukan dengan mengadakan, pengamatan secara langsung terhadap objek penelitian.

D. Alat dan Bahan Penelitian

Bahan dan instrumen penelitian diperlukan untuk membantu kegiatan penelitian pada saat melakukan penelitian. Perlengkapan dan perlengkapan berikut ini diperlukan:

1. Alat

a. Laptop ACER Aspire A514-54 dengan spesifikasi :

- 1) *Processor* : Intel Core i3 4 CPU
- 2) *RAM* : 8GB DDR4
- 3) *Harddisk* : 300GB
- 4) *Monitor* : 12 Inci
- 5) *Printer* : *EpsonL3110*

b. Dalam membuat aplikasi, *software* yang digunakan adalah :

- 1) *Windows 11 Home Single Language*
- 2) *WhatsApp*
- 3) *PHP*
- 4) *XAMPP*
- 5) *MySQL*
- 6) *Framework CodeIgniter*
- 7) *Browser*

2. Bahan

Data yang digunakan dalam penelitian ini berasal dari hasil upaya pengumpulan data para peneliti, yang kemudian akan dilakukan analisis tambahan untuk dijadikan landasan perancangan sistem.

E. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *blackbox testing* dan *whitebox testing*:

1. *Blackbox testing*

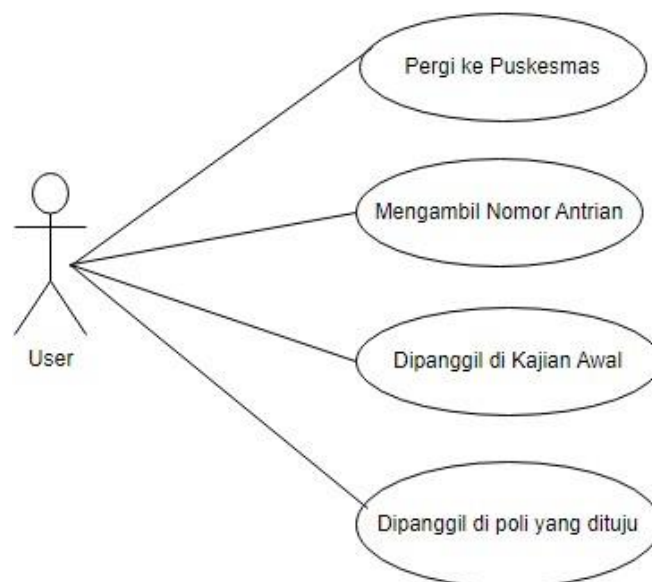
Blackbox testing yaitu untuk menguji program yang sedang berjalan apakah aplikasi Antrian Puskesmas yang dibuat sudah berhasil atau tidak.

2. *Whitebox testing*

Whitebox testing yaitu untuk mengetahui struktur dan alur pada aplikasi Sistem Antrian Pasien Berbasis Web Dan Notifikasi Otomatis Melalui *Whatsapp* Framework *CodeIgniter* apakah sudah sesuai dengan ketentuan kemudian menguji aplikasi tersebut untuk memperoleh hasil.

F. Desain Sistem

1. Desain Sistem yang Berjalan



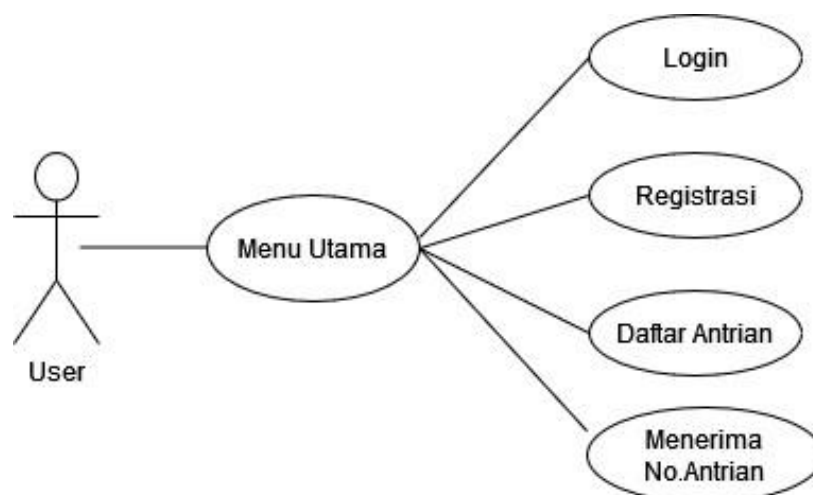
Gambar 3. 1 Desain Sistem yang Berjalan

Pada gambar di atas sistem yang berjalan pada saat ini, *user* langsung ke puskesmas untuk mengambil antrian, kemudian di panggil menuju kajian awal untuk pemeriksaan awal kemudian diberikan kembali antrian untuk ke poli yang di tuju, kemudian baru dipanggil menuju poli.

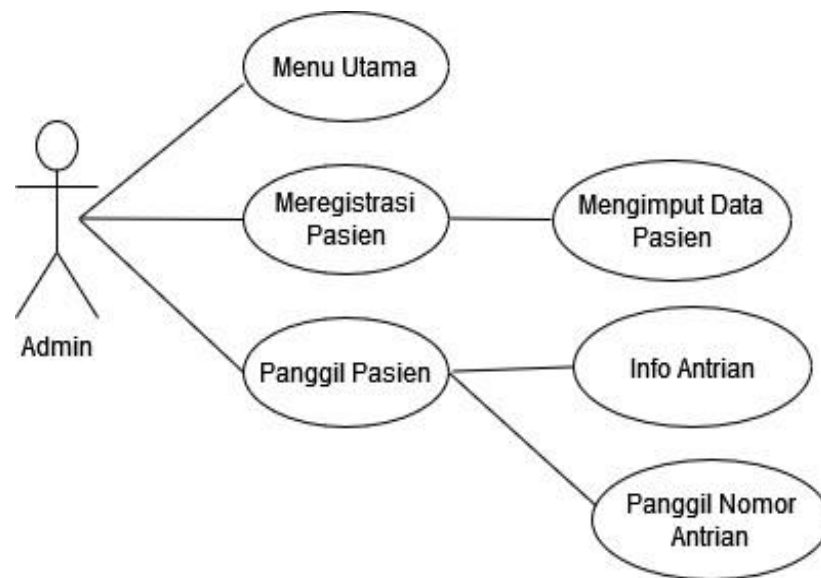
Tabel 3. 1 Penjelasan Desain Sistem yang Berjalan

Nama <i>use case</i>	Deskripsi
<i>User</i> ke Puskesmas	<i>Use case</i> ini menjelaskan tentang sistem yang berjalan bahwa <i>user</i> langsung ke puskesmas.
Mengambil Antrian	<i>Use case</i> ini menjelaskan bahwa <i>user</i> mengambil antrian secara manual di puskesmas
Kajian Awal	<i>Use case</i> ini menjelaskan bahwa <i>user</i> sebelum masuk ke poli akan di panggil untuk kajian awal dan akan di berikan antrian kembali untuk ke poli.
Dipanggil di Poli yang dituju	<i>Use Case</i> ini menjelaskan bahwa <i>user</i> setelah kajian awal akan di panggil menuju poli sesuai dengan keluhan pasien.

2. Desain Sistem yang Diusulkan



Gambar 3. 2 Desain Sistem *User*



Gambar 3. 3 Desain Sistem Admin

Pada gambar di atas, terdapat dua peran dalam aplikasi. Pertama, admin, yang memiliki tanggung jawab untuk mengelola aplikasi, termasuk data materi dan informasi terkait *user* atau pasien. Admin juga dapat memperbarui, menghapus, serta menambah informasi yang dibutuhkan dalam sistem. Selain itu, admin bertugas memastikan bahwa data yang tersimpan tetap aman dan sesuai dengan kebutuhan puskesmas. Kedua, *user/pasien*, yang dapat mengakses aplikasi untuk melakukan pendaftaran di Puskesmas Buntubatu. Aplikasi ini mempermudah pasien dalam mengambil nomor antrian, karena dapat diakses dari rumah, sehingga pasien tidak perlu datang langsung ke puskesmas hanya untuk mengambil nomor antrian. Dengan adanya sistem ini, proses pendaftaran menjadi lebih efisien dan mengurangi kepadatan antrian di lokasi. Pasien juga dapat melihat estimasi waktu pelayanan, sehingga mereka bisa datang sesuai dengan jadwal yang diperkirakan.

Selain itu, aplikasi ini dirancang untuk memberikan pengalaman yang lebih nyaman bagi pasien dengan fitur notifikasi sebagai pengingat jadwal kunjungan. Hal ini

membantu pasien dalam mengatur waktu dengan lebih baik dan menghindari keterlambatan. Keamanan data pasien juga menjadi prioritas utama, sehingga informasi yang diberikan tetap terjaga kerahasiaannya.

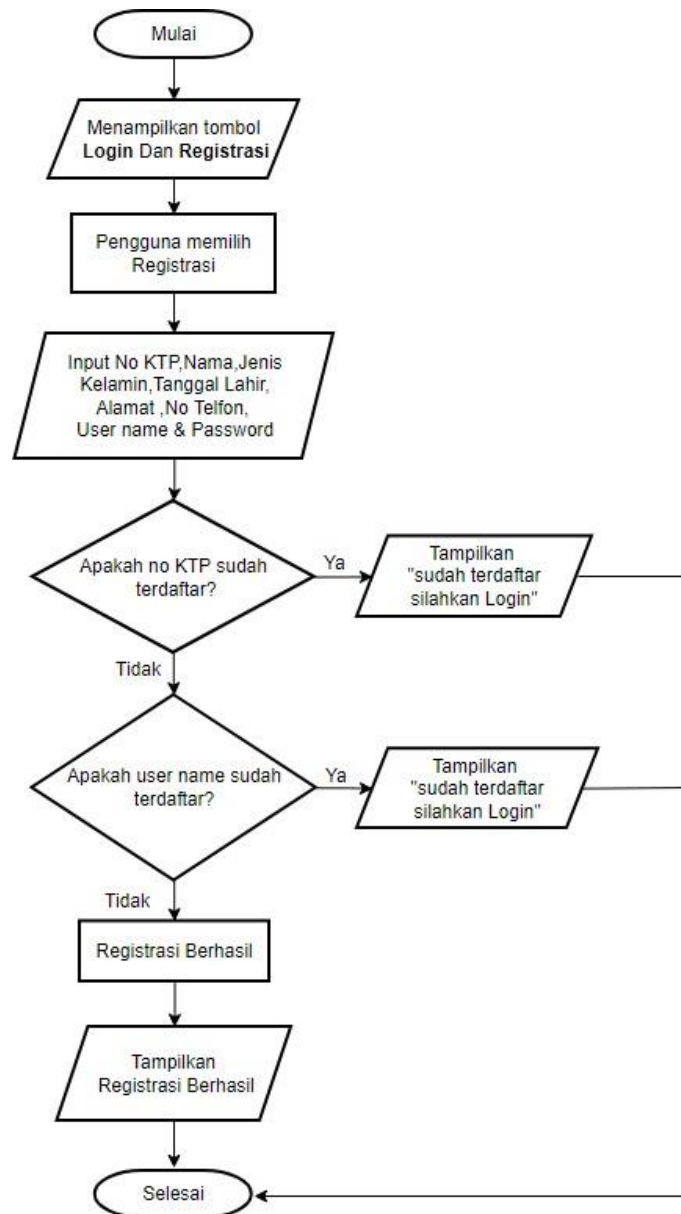
Tabel 3. 2 Penjelasan Desain Sistem Admin

Nama <i>use case</i>	Deskripsi
<i>Login</i>	<i>Use case</i> ini menjelaskan proses dimana admin <i>login</i> terlebih dahulu
Registrasi Pasien	Admin menginput data yang telah di isi oleh pasien sebelumnya
Panggil Pasien	Admin akan menginfokan nomor antrian dan memanggil pasien sesuai dengan nomor antrian

Tabel 3. 3 Penjelasan Desain Sistem User

Nama <i>use case</i>	Deskripsi
<i>Login</i>	<i>Use case</i> ini menjelaskan proses dimana <i>user login</i> terlebih dahulu
Registrasi	<i>Use case</i> ini menjelaskan bahwa <i>user</i> akan melakukan registrasi
Daftar Antrian	<i>Use case</i> ini menjelaskan bahwa <i>user</i> mendaftar terlebih dahulu
Menerima Nomor Antrian	<i>Use case</i> menjelaskan bahwa <i>user</i> akan menerima nomor antrian di <i>whatsapp</i>

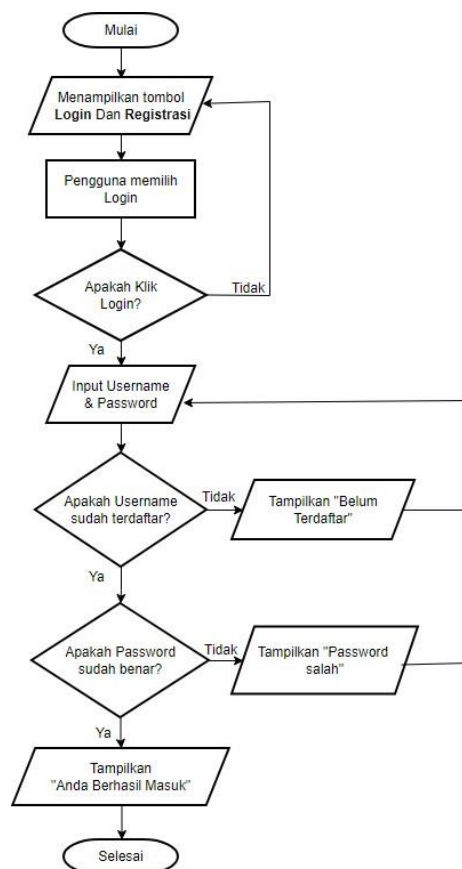
G. Flowchart System



Gambar 3. 4 Flowchart Registrasi User

Pada gambar diatas menjelaskan mengenai alur dari Registrasi User dimana, User atau pasien memilih menu registrasi yang artinya belum pernah mendaftar, kemudian user atau pasien akan menginput nomor KTP, jenis kelamin, tanggal lahir, alamat, no handphone dan juga akan memasukkan username dan

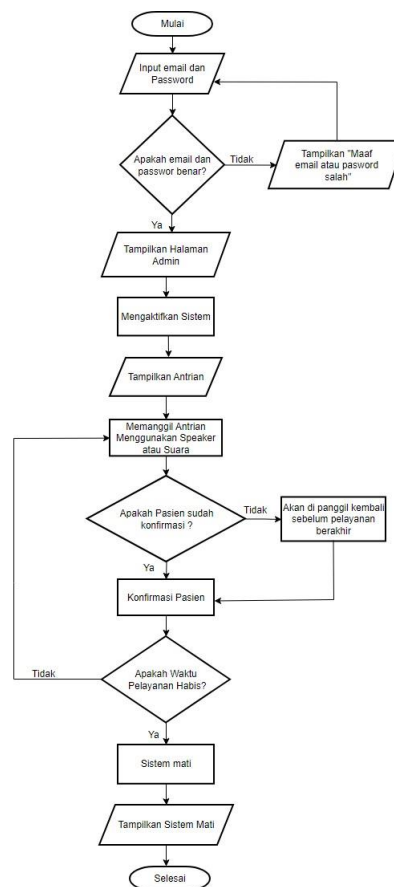
password. Pastikan no KTP yang dimasukkan belum pernah didaftar sebelumnya apabila sudah maka akan tampil “Sudah terdaftar silahkan *Login*” dan juga pastikan *user name* belum terdaftar jika sudah akan tampil “sudah terdaftar silahkan *login*”, *user name* yang di gunakan adalah no KTP pasien setelah itu tampil registrasi berhasil dan selesai.



Gambar 3. 5 *Flowchart Login User*

Pada gambar diatas menjelaskan mengenai alur dari *login user* yaitu *user* akan di tampilkan dengan halaman *login* kemudian *user* mengklik tombol *login* dan apabila *login* berhasil akan masuk ke *input username* dan *password* dan apabila tidak maka kembali ke halaman *login* dan setelah menginput *username* dan *password*. Patikan *username* dan *password* yang di masukkan sudah benar karna

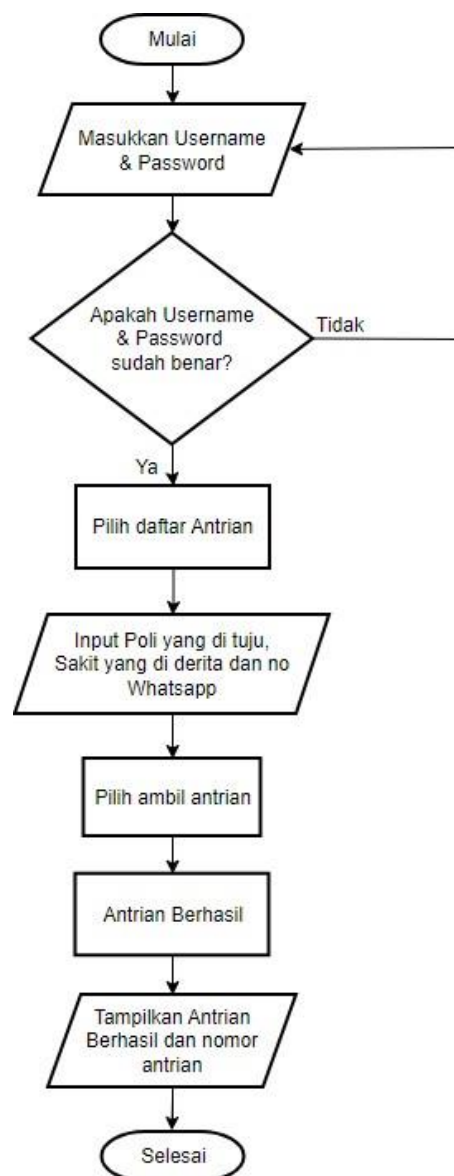
apabila *username* tidak sesuai maka akan menampilkan belum terdaftar dan jika sudah terdaftar maka akan tampil *password* sudah benar jika tidak maka akan tampil *password* salah. Jika *username* dan *password* sudah benar maka akan tampil anda berhasil masuk dan selesai.



Gambar 3. 6 *Flowchart Login Admin*

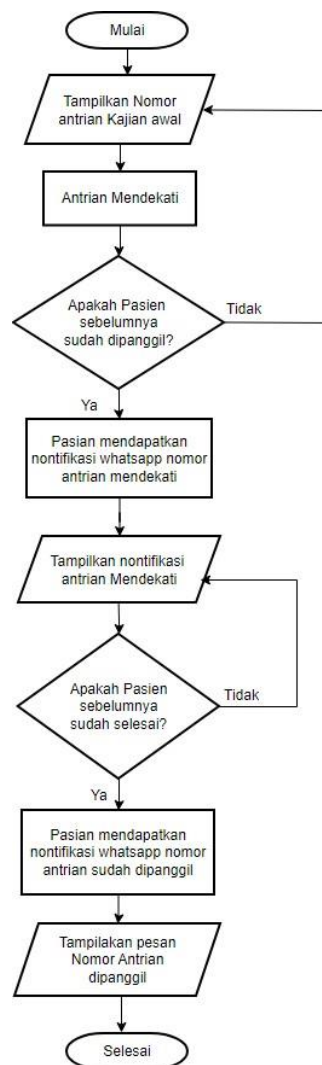
Pada gambar diatas menjelaskan mengenai alur *login admin* yaitu *admin* akan mengimput *email* dan *password* jika *email* dan *password* tidak sesuai maka akan tampil maaf email atau password salah dan akan kembali ke awal, dan jika sudah benar maka akan tampil ke halaman admin dan admin akan mengaktifkan sistem dan akan tampil antrian kemudian akan memanggil antrian menggunakan speaker, kemudian jika pasien tidak konfirmasi sebelum maka akan dipanggil nanti

setelah semua antrian selesai jika masih ada waktu dan jika pasien sudah konfirmasi dan menuju ke tempat yang telah ditentukan. Jika waktu pelayanan belum habis maka akan memanggil antrian selanjutnya dan antrian yang dilewati sebelumnya dan jika waktu habis sistem mati dan menampilkan sistem mati dan selesai.



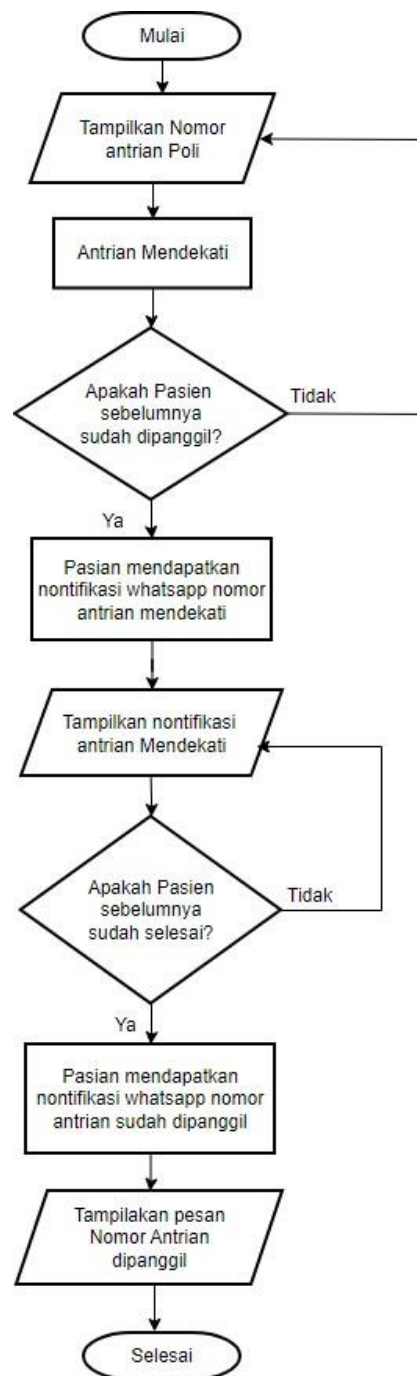
Gambar 3. 7 *Flowchart Daftar Antrian*

Pada gambar menjelaskan alur mendaftar antrian yaitu user atau pasien memasukkan *username* dan *password* yang sudah dibuat pastikan *username* dan *password* sudah benar karena jika tidak maka akan kembali ke awal. Setelah berhasil masuk aplikasi dan memilih daftar antrian kemudian *input* poli yang di tuju, sakit yang diderita, dan nomor *whatsapp* dan memilih ambil antrian dan akan akan tampil antrian berhasil dan nomor antrian dan selesai.



Gambar 3. 8 Flowchart Notifikasi Kajian Awal

Pada gambar *Nonfikasi* kajian awal menjelaskan alur *nontifikasi* kajian awal yaitu aplikasi akan menampilkan nomor antrian untuk kajian awal setelah nomor antrian mendekati, pastikan pasien sebelumnya sudah di panggil jika tidak maka akan tampil nomor antrian, jika sudah di panggil maka akan mendapat pesan nomor antrian mendekati dengan *deskripsi* pesan “Halo [Nama Pasien], Ini adalah pemberitahuan penting dari [Puskesmas Buntu Batu]. Nomor antrian Anda,[Nomor Antrian], akan segera dipanggil di kajian awal. Mohon bersiap-siap dan pastikan Anda berada di [Puskesmas Buntu Batu] dalam beberapa waktu ke depan. Terima kasih atas perhatiannya. Salam sehat,[Puskesmas Buntu Batu]”. Pastikan Pasien sebelumnya sudah selesai jika tidak maka masih tampil antrian mendekati dan jika sudah selesai maka akan mendapatkan pesan dengan deskripsi pesan “Halo [Nama Pasien], Kami dari [Puskesmas Buntu Batu] ingin memberitahu bahwa nomor antrian Anda, [Nomor Antrian], telah dipanggil. Mohon datang ke [Puskesmas Buntu Batu]. Terima kasih atas kerjasamanya. Salam sehat, [Puskesmas Buntu Batu], nomor antrian dipanggil menggunakan speaker dan pasien menuju kajian awal dan jika pasien tidak datang maka akan beralih ke nomor antrian berikutnya dan jika masi ada waktu maka nomor antrian akan kembali di panggil dan selesai.



Gambar 3. 9 *Flowchart Notifikasi Poli*

Pada gambar *Nonfikasi poli* menjelaskan alur *notifikasi poli* yaitu aplikasi akan menampilkan nomor antrian untuk kajian awal setelah nomor antrian mendekati, pastikan pasien sebelumnya sudah di panggil jika tidak maka akan

tampil nomor antrian, jika sudah di panggil maka akan mendapat pesan nomor antrian mendekati dengan *deskripsi* pesan “Halo [Nama Pasien], Ini adalah pemberitahuan penting dari [Puskesmas Buntu Batu]. Nomor antrian Anda,[Nomor Antrian], akan segera dipanggil di poli. Mohon bersiap-siap dan pastikan Anda berada di [Puskesmas Buntu Batu] dalam beberapa waktu ke depan. Terima kasih atas perhatiannya. Salam sehat,[Puskesmas Buntu Batu]”. Pastikan Pasien sebelumnya sudah selesai jika tidak maka masih tampil antrian mendekati dan jika sudah selesai maka akan mendapatkan pesan dengan deskripsi pesan “Halo [Nama Pasien], Kami dari [Puskesmas Buntu Batu] ingin memberitahu bahwa nomor antrian Anda, [Nomor Antrian], telah dipanggil. Mohon datang ke [Puskesmas Buntu Batu]. Terima kasih atas kerjasamanya. Salam sehat, [Puskesmas Buntu Batu], nomor antrian dipanggil menggunakan speaker dan pasien menuju kajian awal dan jika pasien tidak datang maka akan beralih ke nomor antrian berikutnya dan jika masi ada waktu maka nomor antrian akan kembali di panggil dan selesai.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

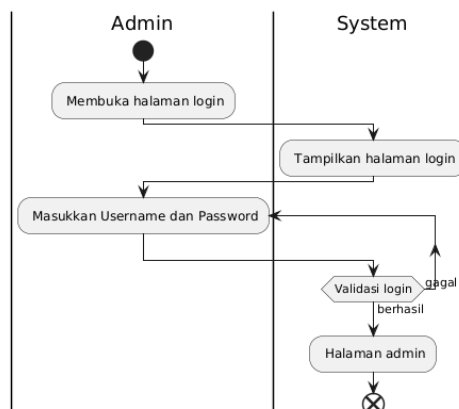
Untuk menganalisis dan merancang sistem antrian pasien, peneliti menggunakan beberapa diagram UML (*Unified Modeling Language*) yang akan membantu dalam menggambarkan aliran data dan interaksi sistem. Diagram-diagram ini meliputi *Activity Diagram*, dan *Sequence Diagram*.

1. Activity Diagram

Activity Diagram adalah alat spesifikasi standar dalam pemodelan perangkat lunak yang digunakan untuk mendokumentasikan dan membangun proses sistem secara *visual*. Diagram ini menggambarkan alur kerja dan aktivitas yang terjadi dalam sistem, serta bagaimana aktivitas-aktivitas tersebut saling berhubungan.

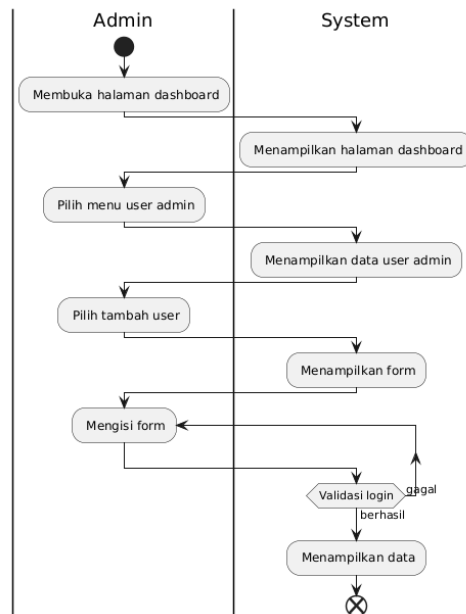
a. Activity Diagram Admin

1) Activity diagram login



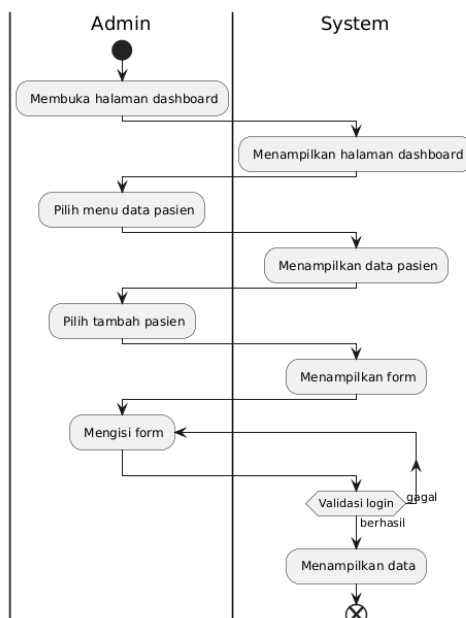
Gambar 4. 1 Activity diagram login

2) *Activity diagram tambah user admin*



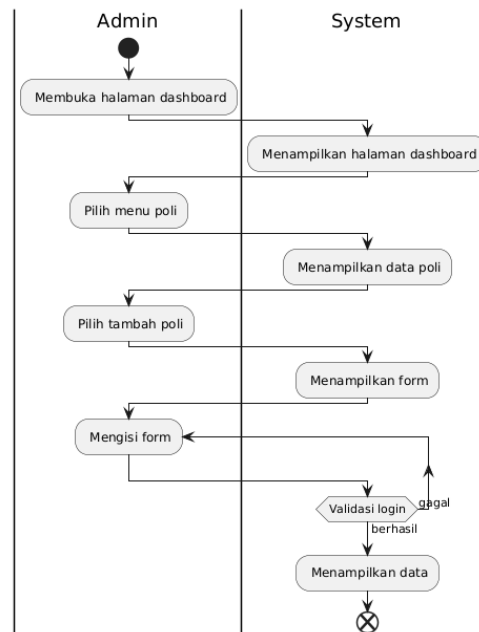
Gambar 4. 2 *Activity diagram tambah user admin*

3) *Activity diagram tambah pasien*



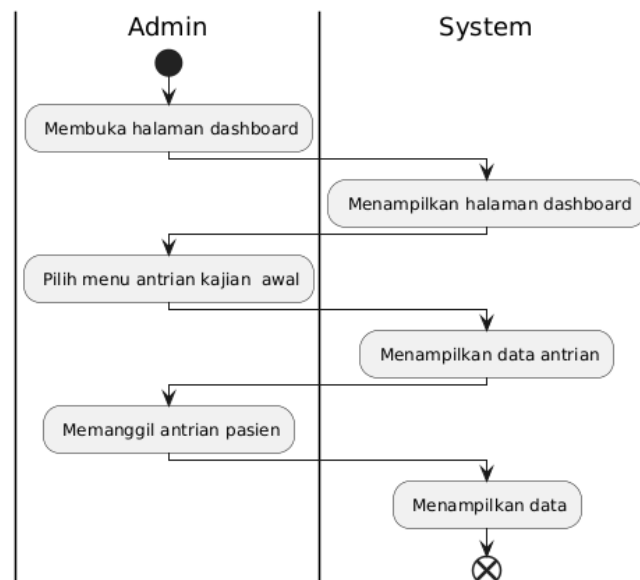
Gambar 4. 3 *Activity diagram tambah pasien*

4) *Activity diagram poli*



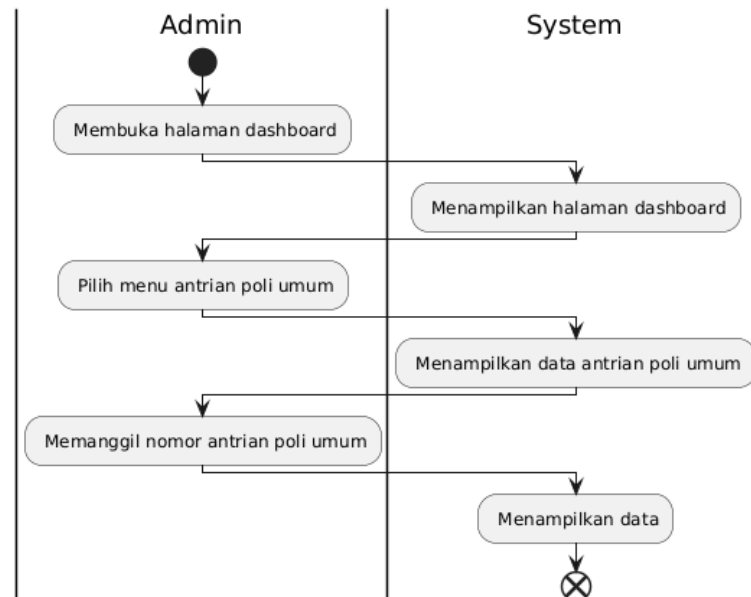
Gambar 4. 4 *Activity diagram poli*

5) *Activity diagram antrian kajian awal*



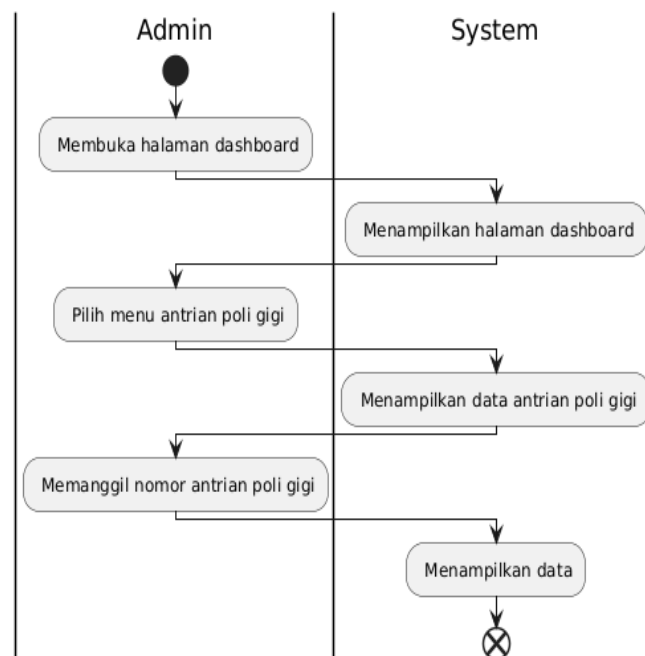
Gambar 4. 5 *Activity diagram antrian kajian awal*

6) *Activity diagram* antrian poli umum



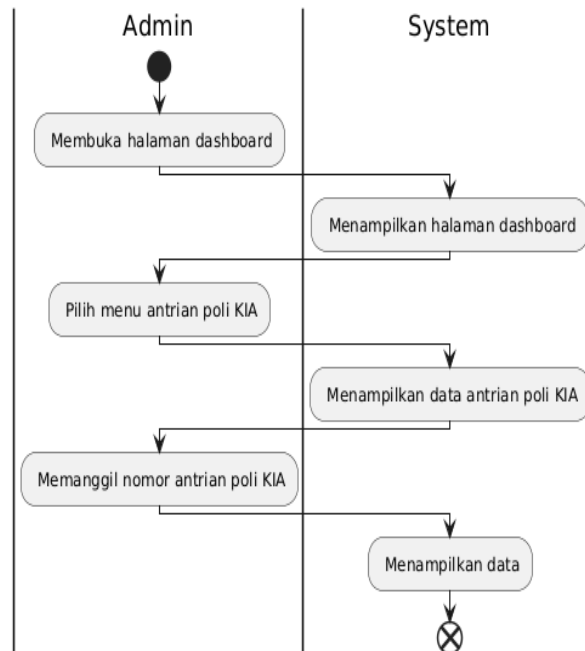
Gambar 4. 6 *Activity diagram* antrian poli umum

7) *Activity diagram* antrian poli gigi



Gambar 4. 7 *Activity diagram* antrian poli gigi

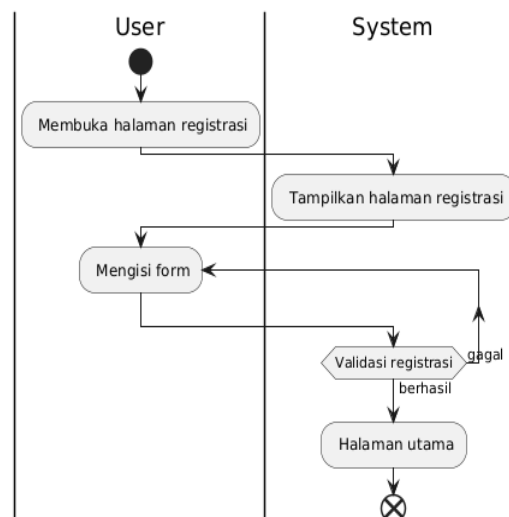
8) *Activity diagram* antrian poli KIA



Gambar 4. 8 *Activity diagram* antrian poli KIA

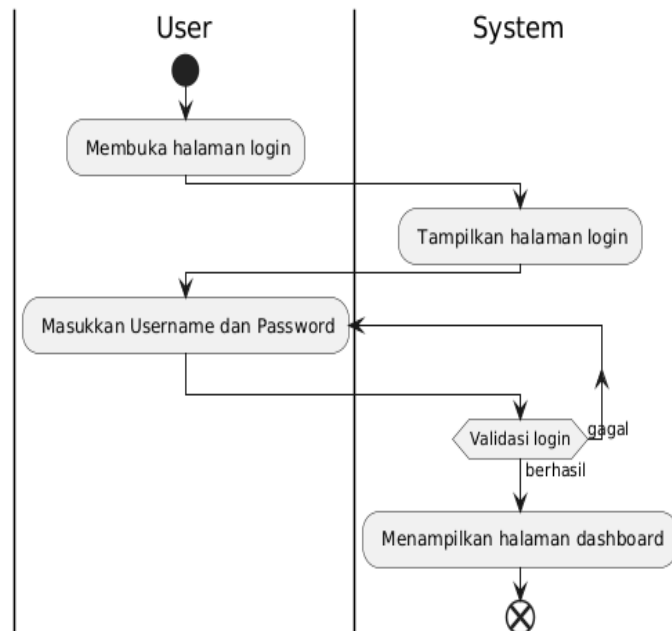
b. *Activity Diagram User*

1) *Activity diagram* registrasi



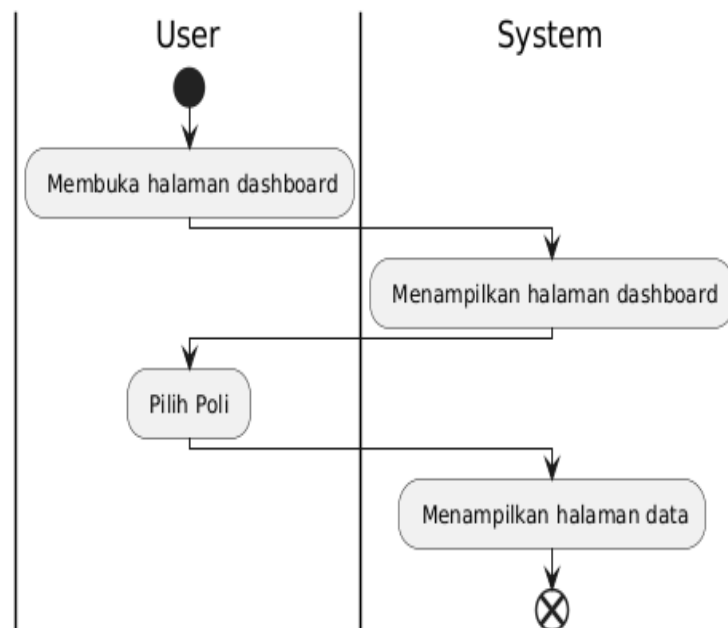
Gambar 4. 9 *Activity diagram* registrasi

2) *Activity diagram login*



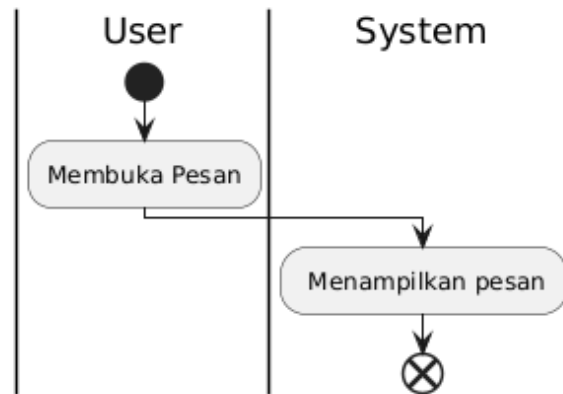
Gambar 4. 10 *Activity diagram login*

3) *Activity diagram ambil antrian*



Gambar 4. 11 *Activity diagram ambil antrian*

4) *Activity diagram* terima pesan



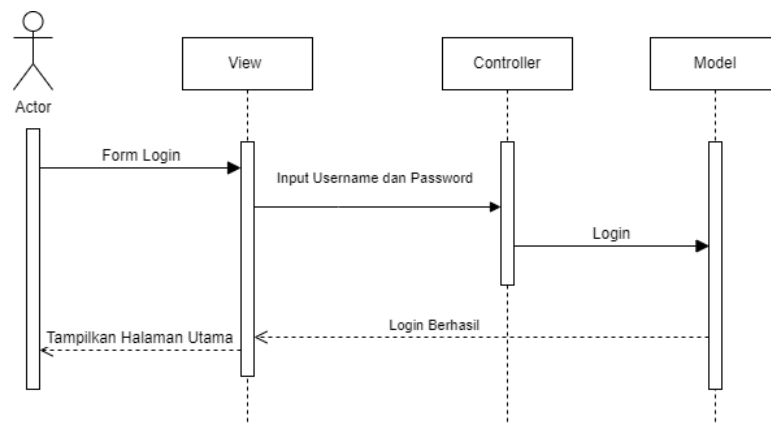
Gambar 4. 12 *Activity diagram* terima pesan

2. *Sequence Diagram*

Sequence diagram merupakan diagram yang menggambarkan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu. Diagram ini menunjukkan bagaimana objek-objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case*.

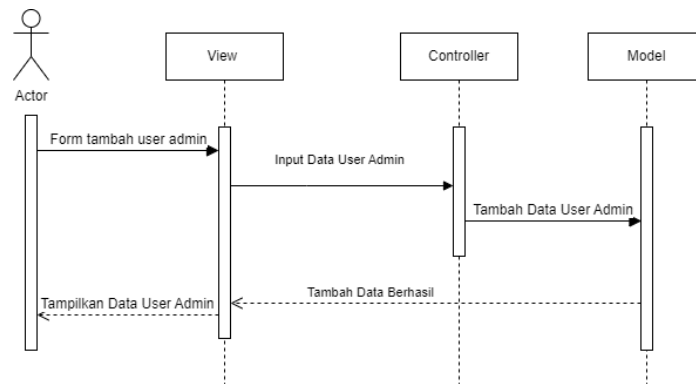
a. *Sequence Diagram* Admin

1) *Sequence diagram* login admin



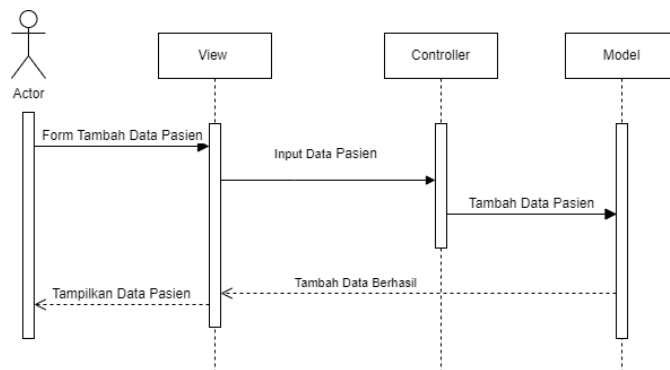
Gambar 4. 13 *Sequence diagram* login admin

2) *Sequence diagram tambah user admin*



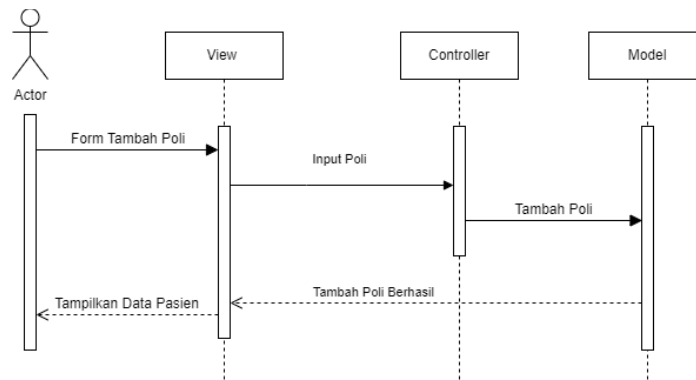
Gambar 4. 14 *Sequence diagram tambah user admin*

3) *Sequence diagram tambah pasien*



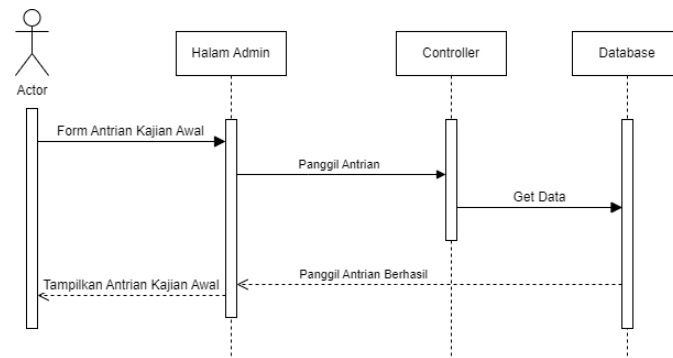
Gambar 4. 15 *Sequence diagram tambah pasien*

4) *Sequence diagram poli*



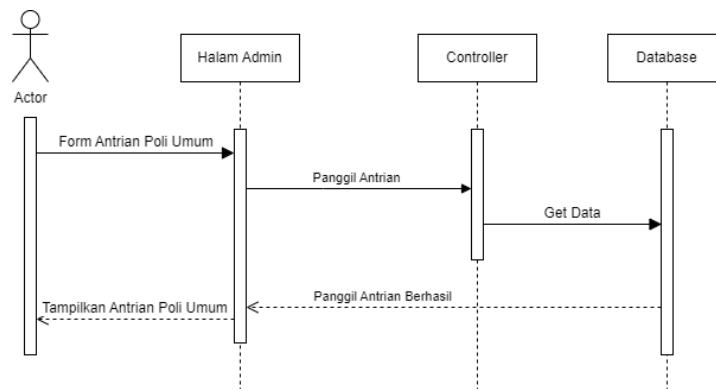
Gambar 4. 16 *Sequence diagram poli*

5) *Sequence diagram antrian kajian awal*



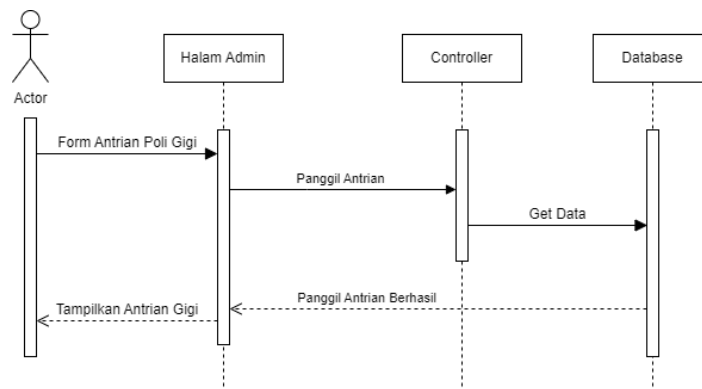
Gambar 4. 17 *Sequence diagram antrian kajian awal*

6) *Sequence diagram antrian poli umum*



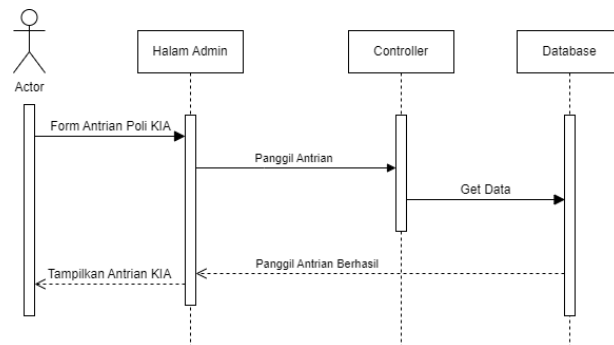
Gambar 4. 18 *Sequence diagram antrian poli umum*

7) *Sequence diagram antrian poli gigi*



Gambar 4. 19 *Sequence diagram antrian poli gigi*

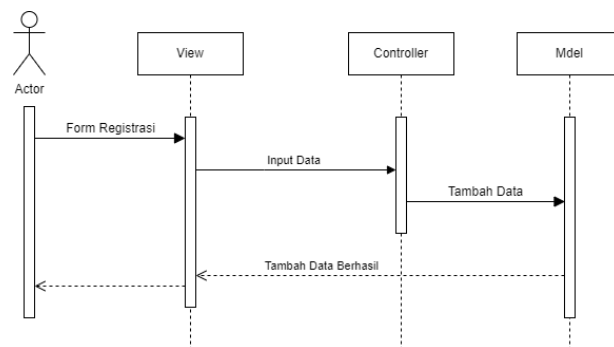
8) *Sequence diagram antrian poli KIA*



Gambar 4. 20 *Sequence diagram antrian poli KIA*

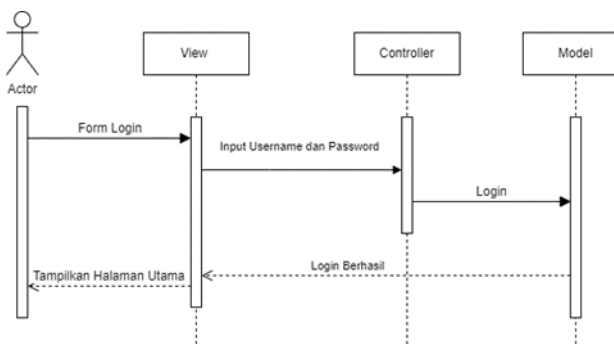
b. *Sequence Diagram User*

1) *Sequence diagram registrasi user*



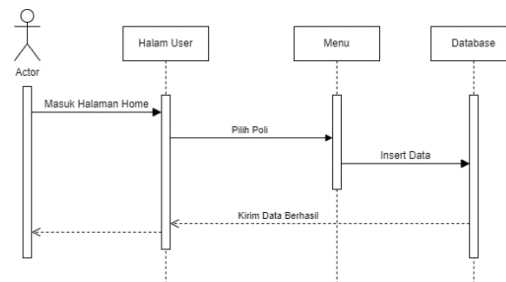
Gambar 4. 21 *Sequence diagram registrasi user*

2) *Sequence diagram login user*



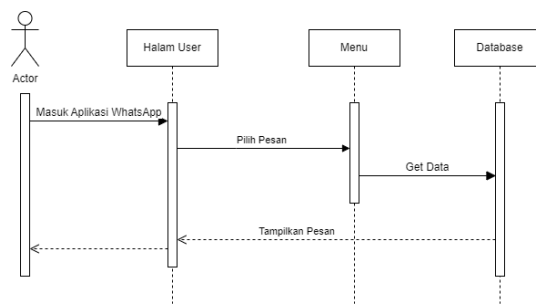
Gambar 4. 22 *Sequence diagram login user*

3) *Sequence* diagram ambil antrian



Gambar 4. 23 *Sequence* diagram ambil antrian

4) *Sequence* diagram terima pesan



Gambar 4. 24 *Sequence* diagram terima pesan

B. Perancangan Database

1. Tabel Admin

Tabel 4. 1 Tabel Admin

No	Nama	Jenis	Ukuran	Ket
1	id	Int		<i>AUTO_INCREMENT</i>
2	email	Varchar	50	
3	password	Varchar	40	
4	nama	Varchar	40	
5	status	Tinyint		
6	gambar	Varchar	255	
7	theme	Varchar	20	

2. Tabel Antrian

Tabel 4. 2 Tabel Antrian

No	Nama	Jenis	Ukuran	Ket
1	id_antrian	Int		<i>AUTO_INCREMENT</i>
2	id_pasien	Int		
3	tgl_antrian	Date		
4	no_antrian	Varchar	10	
5	proses	Int		

3. Tabel Antrian Poli

Tabel 4. 3 Tabel Antrian Poli

No	Nama	Jenis	Ukuran	Ket
1	id_antrian_poli	Int		<i>AUTO_INCREMENT</i>
2	id_antrian	Int		
No	Nama	Jenis	Ukuran	Ket
3	id_pasien	Int		
4	id_poli	Int		
5	no_antrian_poli	Varchar	10	
6	tgl_antrian_poli	Date		
7	proses	Int		

4. Tabel Kategori Poli

Tabel 4. 4 Tabel Kategori Poli

No	Nama	Jenis	Ukuran	Ket
1	id_poli	Int		<i>AUTO_INCREMENT</i>
2	kode_poli	Varchar	5	
3	nama_poli	Varchar	100	
4	jumlah_maksimal	Varchar	2	

5. Tabel Pasien

Tabel 4. 5 Tabel Pasien

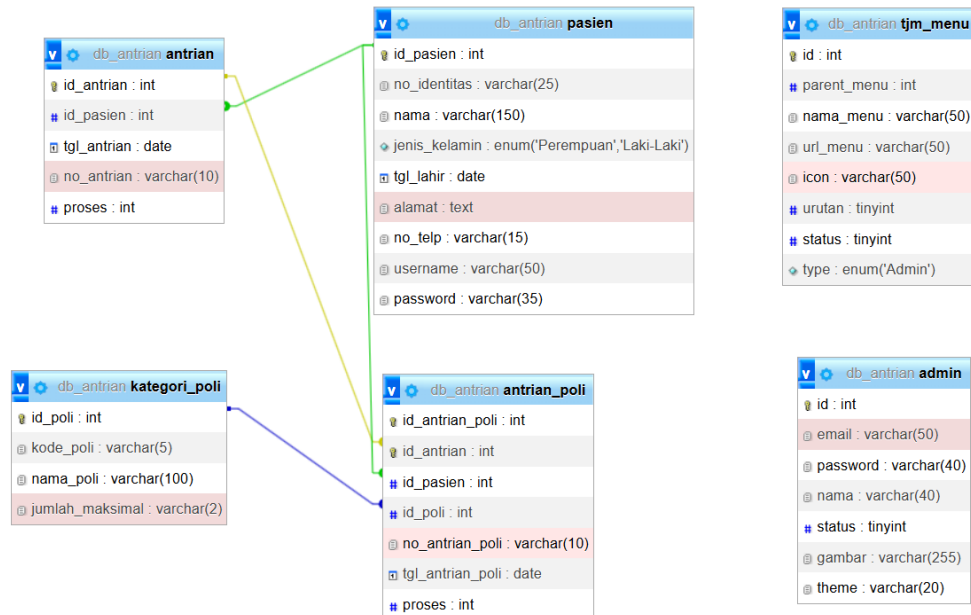
No	Nama	Jenis	Ukuran	Ket
1	id_pasien	Int		<i>AUTO_INCREMENT</i>
2	no_identitas	Varchar	25	
3	nama	Varchar	150	
4	jenis_kelamin	Enum		<i>‘Perempuan’, ‘Laki-laki’</i>
5	tgl_lahir	Date		
6	alamat	Text		
7	no_telp	Varchar	15	
8	username	Varchar	50	
9	password	Varchar	35	

6. Tabel Tjm Menu

Tabel 4. 6 Tabel Tjm Menu

No	Nama	Jenis	Ukuran	Ket
1	id	Int		<i>AUTO_INCREMENT</i>
2	parent_menu	Int		
3	nama_menu	Varchar	50	
4	url_menu	Varchar	50	
5	icon	Varchar	50	
6	urutan	Tinyint		
7	status	Tinyint		
8	type	Enum		<i>‘Admin’</i>

7. Relasi Tabel Database



Gambar 4. 25 Relasi Database

Pada gambar 4.25 *relasi database* terdapat beberapa *foreign key* (kunci asing) yang digunakan untuk membangun keterkaitan antar tabel, beberapa *foreign key* tersebut adalah kolom nama pasien.

Tabel admin berfungsi untuk menyimpan data admin setiap poli, yang didalamnya terdapat data yaitu id (sebagai *Primary Key*), email, *password*, nama, status, gambar, dan *theme*.

Tabel antrian berfungsi untuk menyimpan data antrian pasien, yang didalamnya terdapat data id_antrian (sebagai *Primary Key*), id_pasien, tgl_antrian, no_antrian dan proses. Relasi antar tabel saat pasien mengisi data untuk mendapatkan antrian akan masuk ke tabel data pasien dan juga ke tabel data antrian poli.

Tabel Antrian Poli untuk menyimpan data pasien sesuai dengan poli yang di tuju, yang didalamnya terdapat data *id_antrian_poli* (sebagai *Primary Key*), *id_antian*, *id_pasien*, *id_poli*, *no_antrian_poli*, *tgl_antrian_poli* dan *proses*. Relasi antar tabel yaitu data pasien tersimpan juga di tabel data pasien keseluruhan.

Tabel kategori poli terdapat data *Id_poli* (sebagai *Primary Key*), *kode_poli*, *nama_poli* dan *jumlah_maksimal*.

Tabel Pasien berfungsi untuk menyimpan data-data pasien yang telah mendaftar, yang didalamnya terdapat 6 data, *id_pasien* (sebagai *primary key*), *no_idntitas*, *nama*, *jenis_kelamin*, *tgl_lahir*, *alamat*, *no_telp*, *username* dan *password*. Relasi antrar tabel yaitu pasien yang sudah mengisi data maka otomatis data akan tersimpan di tabel antrian dan tabel antrian poli.

Tabel *tjm_menu* berfungsi untuk menyimpan informasi mengenai item menu yang digunakan, yang didalamnya terdapat 3 data yaitu *parent_menu* (sebagai *primary key*), *nama_menu* dan *icon*.

8. *Fonnte.com*

Berikut adalah langkah-langkah untuk mendaftar di *Fonnte.com*:

1. Kunjungi Situs *Fonnte*

Buka fonnte.com melalui *browser* anda.

2. Klik Tombol “Daftar”

Cari tombol “**Daftar**” di pojok kanan atas halaman utama, lalu klik.

3. Isi Formulir Pendaftaran

Masukkan informasi yang diperlukan, seperti:

- a) **Nama Lengkap:** Isikan nama Anda.

- b) **Email:** Gunakan email aktif.
- c) **Kata Sandi:** Buat kata sandi yang kuat.

Setelah itu, klik **“Daftar”** atau **“Sign Up”**.

4. Konfirmasi Email

Periksa email Anda untuk pesan konfirmasi dari *Fonnte*. Klik tautan yang diberikan untuk memverifikasi akun.

5. Login ke Dashboard

Setelah verifikasi, kembali ke situs *Fonnte* dan login menggunakan email serta kata sandi yang telah dibuat.

6. Pilih Paket

- a) Setelah masuk, Anda dapat memilih paket layanan sesuai kebutuhan. Ada opsi **Gratis** maupun **Berbayar**.
- b) Untuk berlangganan paket berbayar, Anda perlu melakukan pembayaran melalui metode yang tersedia.

7. Integrasi dengan *WhatsApp*

- a) Ikuti panduan yang tersedia di dashboard untuk menghubungkan akun *WhatsApp* Anda ke *Fonnte*.
- b) Biasanya, Anda akan diberikan QR code untuk dipindai melalui aplikasi *WhatsApp*.

Tips:

- a) Pastikan menggunakan nomor *WhatsApp* yang aman, terutama jika Anda ingin mengirim pesan dalam jumlah besar.
- b) Baca syarat dan ketentuan untuk menghindari risiko akun diblokir.

Jika ada kendala, Anda dapat menghubungi layanan pelanggan *Fonnte* melalui menu **Bantuan** di *dashboard*.

Fonnte.com mengirim pesan otomatis menggunakan layanan **WhatsApp API Gateway** yang memungkinkan integrasi sistem Anda dengan *WhatsApp*. Berikut adalah langkah-langkah untuk mengatur dan menggunakan fitur pengiriman pesan otomatis di *Fonnte*:

1. Buat Akun dan *Login*

Daftar dan *login* ke *dashboard Fonnte*.

2. Hubungkan Nomor *WhatsApp*

Masuk ke menu **Dashboard** dan pilih **QR Code** untuk menghubungkan nomor *WhatsApp* Anda. Pindai QR code menggunakan aplikasi *WhatsApp* di ponsel Anda (mirip dengan *WhatsApp Web*).

3. Siapkan Pesan Otomatis

- a) Pergi ke menu **Pesan Otomatis**.
- b) Pilih jenis pesan yang ingin Anda kirim:
- c) **Teks**: Untuk pesan biasa.
- d) **Gambar atau Dokumen**: Lampirkan file seperti foto, PDF, atau dokumen lainnya.

4. Atur Penerima

- a) Ada beberapa opsi untuk mengatur penerima:
- b) **Manual**: Masukkan nomor secara langsung.
- c) **Grup atau Daftar Kontak**: Upload file berisi daftar nomor menggunakan format CSV atau TXT.

5. Buat Jadwal (Opsional)

- a) Jika ingin mengatur pesan terjadwal:
- b) Pilih menu **Penjadwalan**.
- c) Atur waktu dan tanggal pengiriman pesan.

6. Integrasi API (Opsional untuk Developer)

- a) Jika Anda ingin mengintegrasikan pengiriman pesan dengan aplikasi atau sistem lain:
- b) Masuk ke menu **API** di dashboard.
- c) Ambil **API Key** Anda.
- d) Gunakan endpoint API yang disediakan Fonnte untuk mengirim pesan.

Contoh permintaan API menggunakan cURL:

```
curl -X POST https://api.fonnte.com/send \
-H "Authorization: Bearer {api_key}" \
-F "target=6281234567890" \
-F "message=Halo, ini pesan otomatis dari Fonnte!" \
-F "url=https://example.com/image.jpg" # Opsional untuk media
```

- e) API Fonnte mendukung pesan teks, media, dan dokumen.

7. Kirim Pesan

- a) Setelah semua pengaturan selesai, klik **Kirim Sekarang**.
- b) Pesan akan dikirimkan secara otomatis sesuai pengaturan Anda.

Fitur Tambahan

- a) **Pesan Respon Otomatis**: Atur balasan otomatis untuk pesan masuk.
- b) **Notifikasi Bisnis**: Kirimkan notifikasi seperti konfirmasi pesanan atau pengingat pembayaran.

Tips Penting

Hindari mengirim pesan ke kontak yang tidak mengenal Anda untuk meminimalkan risiko nomor Anda diblokir.

Gunakan nomor khusus untuk aktivitas ini jika mengirim pesan dalam jumlah besar.

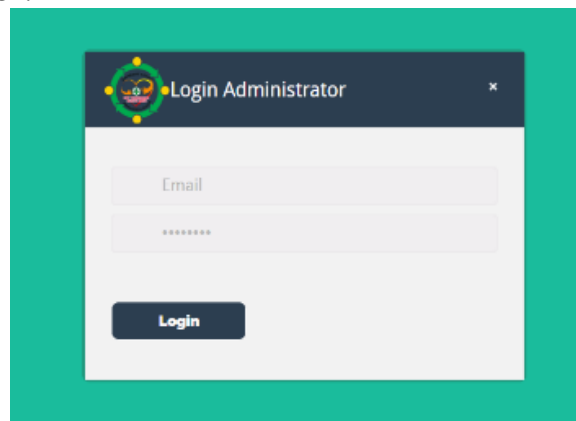
Pantau statistik pengiriman pesan di dashboard Fonnte untuk melihat keberhasilannya.

C. Detail Sistem

1. Admin

a. Halaman *Login Admin*

Halaman *Login* ini merupakan sebagai pintu gerbang yang memungkinkan admin untuk masuk ke dalam sistem dengan mengidentifikasi identitas mereka, memastikan bahwa hanya admin yang berwenang yang dapat mengakses dan mengelola data serta fungsi sistem antrian pasien.



Gambar 4. 26 Halaman *Login Admin*

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Login extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('M_loginadmin');
    }

    public function index()
    {
        // halaman login, kalau sudah login, maka pindah ke halaman dashboard admin
        if ($this->session->userdata('login_admin'))
            redirect('admin/dashboard');

        if (isset($_POST['login'])) {
            $data['result'] = $this->M_loginadmin->login();
            $this->load->view('admin/v_login', $data);
        } else {
            $this->load->view('admin/v_login');
        }
    }

    public function logout()
    {
        $this->session->sess_destroy();
        redirect('admin/login');
    }
}

```

Gambar 4. 27 Coding Tampilan Halaman *Login* Admin

```

public function proses_login()
{
    print_r($_POST);
    $username = $this->input->post('username');
    $password = md5($this->input->post('password'));

    $this->db->where('username', $username);
    $this->db->where('password', $password);
    $getpasien = $this->db->get('pasien')->row();

    if ($getpasien) {
        $this->session->set_userdata('id_pasien', $getpasien->id_pasien);
        $this->session->set_userdata('nama', $getpasien->nama);

        $this->session->set_flashdata("notif", true);
        $this->session->set_flashdata("pesan", 'Login Berhasil');
        $this->session->set_flashdata("type", 'success');
        redirect(base_url());
    } else {
        $this->session->set_flashdata("notif", true);
        $this->session->set_flashdata("pesan", 'Username atau Password Salah');
        $this->session->set_flashdata("type", 'warning');
        redirect(base_url());
    }
}
}

```

Gambar 4. 28 Coding *Login* Admin

```

public function proses_login()
{
    print_r($_POST);
    $username = $this->input->post('username');
    $password = md5($this->input->post('password'));

    $this->db->where('username', $username);
    $this->db->where('password', $password);
    $getpasien = $this->db->get('pasien')->row();

    if ($getpasien) {
        $this->session->set_userdata('id_pasien', $getpasien->id_pasien);
        $this->session->set_userdata('nama', $getpasien->nama);

        $this->session->set_flashdata("notif", true);
        $this->session->set_flashdata("pesan", 'Login Berhasil');
        $this->session->set_flashdata("type", 'success');
        redirect(base_url());
    } else {
        $this->session->set_flashdata("notif", true);
        $this->session->set_flashdata("pesan", 'Username atau Password Salah');
        $this->session->set_flashdata("type", 'warning');
        redirect(base_url());
    }
}
}*
```

Gambar 4. 29 Coding Proses *Login* Admin

b. Halaman Tambah *User* Admin

Halaman Tambah *User* Admin adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin utama menambahkan pengguna admin baru ke dalam sistem.

The screenshot shows a web application interface. On the left is a dark sidebar with a menu. The main content area is titled 'Admin - User'. Below the title is a breadcrumb 'Dashboard / Admin'. The 'Add User' form contains the following fields: 'Email' (text input), 'Password' (password input), 'Name' (text input), and 'Status' (dropdown menu with 'Select Status' as the placeholder). At the bottom of the form are three buttons: 'Save', 'Save and go back to list', and 'Cancel'.

Gambar 4. 30 Halaman Tambah *User* Admin


```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
include('Super.php');

class Useradmin extends Super
{
    function __construct()
    {
        parent::__construct();
        $this->language = 'english'; /** Indonesian / english **/
        $this->tema = "datatables"; /** datatables / flexigrid **/
        $this->tabel = "admin";
        $this->active_id_menu = "user";
        $this->nama_view = "User";
        $this->status = true;
        $this->field_tambah = array('email', 'password', 'nama', 'status');
        $this->field_edit = array('email', 'password', 'nama', 'status');
        $this->field_tampil = array('email', 'nama');
        $this->folder_upload = 'assets/uploads/files';
        $this->add = true;
        $this->edit = true;
        $this->delete = true;
        $this->crud;
    }

    function index(){
        $data = [];
        /** Bagian GROCERY CRUD USER**/

        /** Relasi Antar Tabel
         * @parameter (nama_field ditabel ini, tabel_relasi, field dari tabel_relasinya)
         **/
        // $this->crud->set_relation('user_id','users','username');

        /** Upload **/
        // $this->crud->set_field_upload('nama_field_upload',$this->folder_upload);

        /** Ubah Nama yang akan ditampilkan**/
        // $this->crud->display_as('nama','Nama Setelah di Edit')
        // ->display_as('email','Email Setelah di Edit');

        /** Akhir Bagian GROCERY CRUD Edit Oleh User**/
        $data = array_merge($data,$this->generateBreadcumbs());
        $data = array_merge($data,$this->generateData());
        $this->generate();
        $data['output'] = $this->crud->render();
        $this->load->view('admin/'.$this->session->userdata('theme').'/v_index',$data);
    }

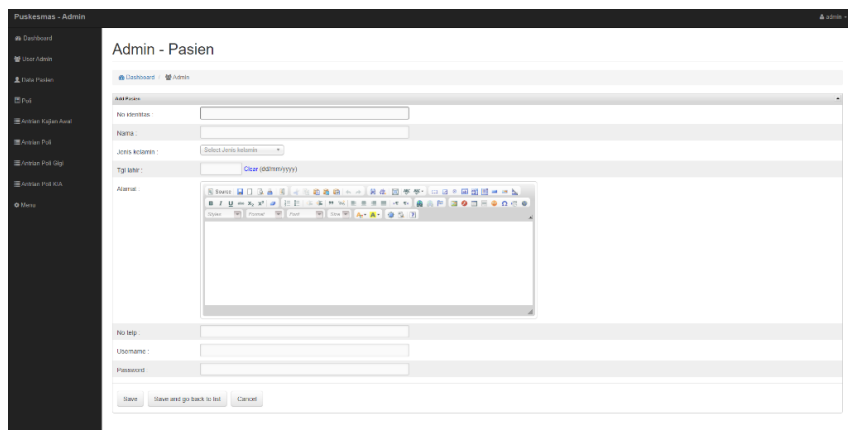
    private function generateBreadcumbs(){
        $data['breadcumbs'] = array(
            array(
                'nama'=>'Dashboard',
                'icon'=>'fa fa-dashboard',
                'url'=>'admin/dashboard'
            ),
            array(
                'nama'=>'Admin',
                'icon'=>'fa fa-users',
                'url'=>'admin/useradmin'
            ),
        );
        return $data;
    }
}

```

Gambar 4. 31 Coding Halaman Tambah *User* Admin

c. Halaman Tambah Pasien

Halaman Tambah Pasien adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin memasukkan data pasien baru ke dalam sistem antrian.



Gambar 4. 32 Halaman Tambah Pasien

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
include('Super.php');

class Pasien extends Super
{
    function __construct()
    {
        parent::__construct();
        $this->language = 'english'; /** Indonesian / english **/
        $this->tema = "fiexigrid"; /** datatables / fiexigrid **/
        $this->tabel = "pasien";
        $this->active_id_menu = "Pasien";
        $this->nama_view = "Pasien";
        $this->status = true;
        $this->field_tambah = array();
        $this->field_edit = array();
        $this->field_tampil = array();
        $this->folder_upload = 'assets/uploads/files';
        $this->add = true;
        $this->edit = true;
        $this->delete = true;
        $this->crud;
    }

    function index(){
        $data = [];
        if($this->crud->getState() == "save")
            redirect(base_url('admin/pasien/save_data'));
        /** Bagian GROCERY CRUD USER**/

        /** Relasi Antar Tabel
         * @parameter (nama_field_ditabel_ini, tabel_relasi, field_dari_tabel_relasinya)
         **/
        // $this->crud->set_relation('id_kategori', 'kategori', 'nama_kategori');
        // $this->crud->set_relation_n_n('warna', 'relasi_warna', 'warna', 'id_produk', 'id_warna', 'warna');

        /** Upload **/
        // $this->crud->set_field_upload('nama_field_upload', $this->folder_upload);
        // $this->crud->set_field_upload('gambar', $this->folder_upload);

        /** Ubah Nama yang akan ditampilkan**/
        // $this->crud->display_as('nama', 'Nama Setelah di Edit')
        // ->display_as('email', 'Email Setelah di Edit');

        /** Akhir Bagian GROCERY CRUD Edit Oleh User**/
        $data = array_merge($data, $this->generateBreadcrumbs());
        $data = array_merge($data, $this->generateData());
        $this->crud->callback_before_insert(array($this, 'encrypt_password_callback'));
        $this->generate();
        $data['output'] = $this->crud->render();
        $this->load->view('admin/'. $this->session->userdata('theme'). '/v_index', $data);
    }

    private function generateBreadcrumbs(){
        $data['breadcrumbs'] = array(
            array(
                'nama' => 'Dashboard',
                'icon' => 'fa fa-dashboard',
                'url' => 'admin/dashboard'
            ),
            array(
                'nama' => 'Admin',
                'icon' => 'fa fa-users',
                'url' => 'admin/useradmin'
            ),
        );
        return $data;
    }

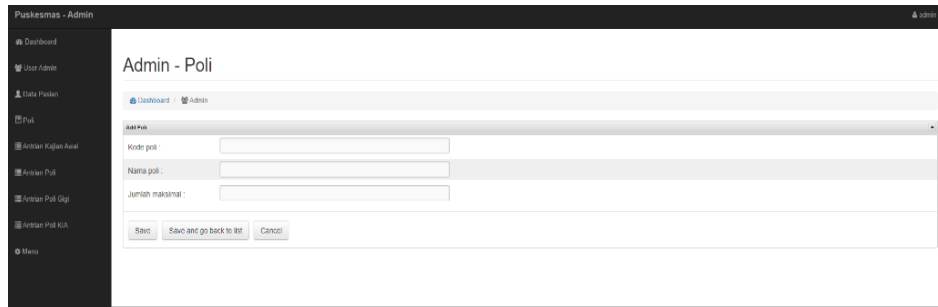
    function encrypt_password_callback($post_array) {
        // $this->load->library('encrypt');
        // $key = 'super-secret-key';
        // $post_array['password'] = $this->encrypt->encode($post_array['password'], $key);
        $post_array['password'] = md5($post_array['password']);

        return $post_array;
    }
}
```

Gambar 4. 33 Coding Halaman Tambah Pasien

d. Halaman Tambah Poli

Halaman Tambah Poli adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin menambahkan data poli atau unit pelayanan baru ke dalam sistem antrian pasien.



Gambar 4. 34 Halaman Tambah Poli

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
include('Super.php');

class Poli extends Super
{
    function __construct()
    {
        parent::__construct();
        $this->language = 'english'; /** Indonesian / english **/
        $this->tema = "flexigrid"; /** datatables / flexigrid **/
        $this->tabel = "kategori_poli";
        $this->active_id_menu = "poli";
        $this->nama_view = "poli";
        $this->status = true;
        $this->field_tambah = array();
        $this->field_edit = array();
        $this->field_tampil = array();
        $this->folder_upload = 'assets/uploads/files';
        $this->add = true;
        $this->edit = true;
        $this->delete = true;
        $this->crud;
    }

    function index(){
        $data = [];
        /** Bagian GROCERY CRUD USER**/

        /** Relasi Antar Tabel
         * @parameter (nama_field_ditabel_ini, tabel_relasi, field_dari_tabel_relasinya)
         **/
        // $this->crud->set_relation('id_kategori', 'kategori', 'nama_kategori');
        // $this->crud->set_relation_n_n('warna', 'relasi_warna', 'warna', 'id_produk', 'id_warna', 'warna');

        /** Upload **/
        // $this->crud->set_field_upload('nama_field_upload', $this->folder_upload);
        // $this->crud->set_field_upload('gambar', $this->folder_upload);

        /** Ubah Nama yang akan ditampilkan**/
        // $this->crud->display_as('nama', 'Nama Setelah di Edit')
        // ->display_as('email', 'Email Setelah di Edit');

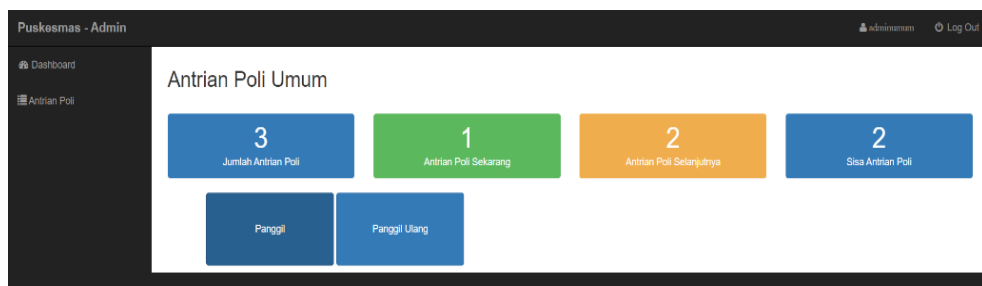
        /** Akhir Bagian GROCERY CRUD Edit Oleh User**/
        $data = array_merge($data, $this->generateBreadcrumbs());
        $data = array_merge($data, $this->generateData());
        $this->generate();
        $data['output'] = $this->crud->render();
        $this->load->view('admin/'.$this->session->userdata('theme').'/v_index', $data);
    }

    private function generateBreadcrumbs(){
        $data['breadcrumbs'] = array(
            array(
                'nama'=>'Dashboard',
                'icon'=>'fa fa-dashboard',
                'url'=>'admin/dashboard'
            ),
            array(
                'nama'=>'Admin',
                'icon'=>'fa fa-users',
                'url'=>'admin/useradmin'
            ),
        );
        return $data;
    }
}
```

Gambar 4. 35 Coding Halaman Tambah Poli

f. Halaman Antrian Poli Umum

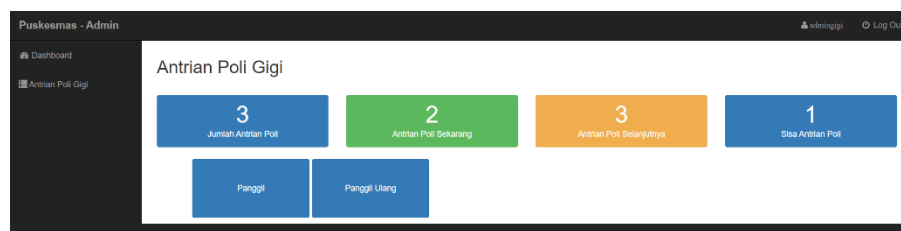
Halaman Antrian Poli Umum adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin mengelola dan memantau antrian pasien yang akan dilayani di poli umum.



Gambar 4. 38 Halaman Antrian Poli Umum

g. Halaman Antrian Poli Gigi

Halaman Antrian Poli Gigi adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin mengelola dan memantau antrian pasien yang akan dilayani di poli gigi.



Gambar 4. 39 Halaman Antrian Poli Gigi

h. Halaman Antrian Poli KIA

Halaman Antrian Poli KIA adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin mengelola dan memantau antrian pasien yang akan dilayani di poli Kesehatan Ibu dan Anak (KIA).



Gambar 4. 40 Halaman Antrian Poli KIA

2. *User*

a. Halaman Registrasi *User*

Halaman Registrasi untuk *User* adalah bagian dari sistem yang dirancang khusus untuk memungkinkan admin membuat akun baru bagi pengguna sistem.

Gambar 4. 41 Halaman Registrasi *User*

```

<!-- Modal -->
<div class="modal fade bd-example-modal-lg" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Registrasi Home</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <span style="display: none;" id="generated-code"></span>
        <form action="{?php echo base_url('Index/registrasi') }?" method="post">
          <div>
            <label>Nomor KTP <sup>*</sup></label>
            <input type="text" id="no_identitas" name="no_identitas" class="form-control"
              value="" placeholder="Nomor KTP" required=""
              pattern="\d{16}" title="Harus 16 digit angka" maxlength="16"
              oninput="syncUsername()" />
          </div>

          <div>
            <label>Nama <sup>*</sup></label>
            <input type="text" id="nama" name="nama" class="form-control" value="" placeholder="Nama" required="" />
          </div>

          <div>
            <label>Jenis Kelamin</label>
            <select id="jenis_kelamin" name="jenis_kelamin" class="form-control">
              <option value="">Pilih</option>
              <option value="Laki-Laki">Laki-Laki</option>
              <option value="Perempuan">Perempuan</option>
            </select>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

Gambar 4. 42 Coding Halaman Registrasi *User*

```

<div align="right">
  <!-- <a href="{?php //echo base_url() }?" >Registrasi</a> -->
  <button type="button" class="btn btn-secondary" data-dismiss="modal">Tutup</button>
  <button type="submit" class="btn btn-primary">Simpan</button>
</div>

```

Gambar 4. 43 Coding Simpan dan Tutup

```

public function registrasi()
{
    $no_identitas = $this->input->post('no_identitas'); //langkah 2
    $nama = $this->input->post('nama');
    $jenis_kelamin = $this->input->post('jenis_kelamin');
    // $usia = $this->input->post('usia');
    $tgl_lahir = $this->input->post('tgl_lahir');
    $alamat = $this->input->post('alamat');
    $no_telp = $this->input->post('no_telp');
    $username = $this->input->post('username');
    $password = md5($this->input->post('password'));

    $this->db->set('no_identitas', $no_identitas); //langkah ke 3
    $this->db->set('nama', $nama);
    $this->db->set('jenis_kelamin', $jenis_kelamin);
    // $this->db->set('usia', $usia);
    $this->db->set('tgl_lahir', $tgl_lahir);
    $this->db->set('alamat', $alamat);
    $this->db->set('no_telp', $no_telp);
    $this->db->set('username', $username);
    $this->db->set('password', $password);

    $this->db->insert('pasien');

    $this->session->set_flashdata("notif", true);
    $this->session->set_flashdata("pesan", 'Registrasi Berhasil');
    $this->session->set_flashdata("type", 'success');

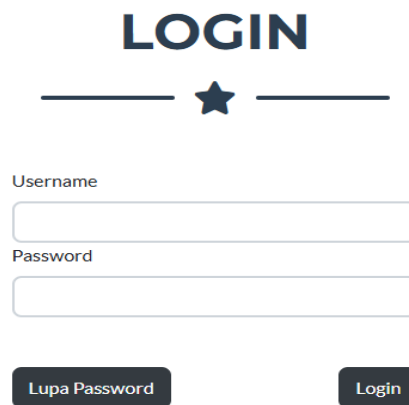
    redirect(base_url());
}

```

Gambar 4. 44 Coding Proses Simpan

b. Halaman *Login User*

Halaman *Login User* adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna masuk ke dalam sistem dengan memasukkan *username* dan *password* mereka.



LOGIN

★

Username

Password

[Lupa Password](#)
[Login](#)

Gambar 4. 45 Halaman *Login User*

```
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col-md-4" style="margin-top: 20px">
      <!-- <h1 align="center">Login </h1> -->
      <form action="<?php echo base_url('Index/proses_login') ?>" method='post'>
        <label>Username</label>
        <input type="username" name="username" autocomplete="off" class="form-control">

        <label>Password</label>
        <input type="password" name="password" autocomplete="off" class="form-control">
        <br><br>

        <div style="display: flex; justify-content: space-between; width: 100%;">
          <a href="lupapassword.php" target="_blank" class="btn btn-dark">Lupa Password</a>
          <button type="submit" class="btn btn-dark">Login</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

Gambar 4. 46 Coding Halaman *Login User*


```

<!-- Portfolio Grid Items -->
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col-md-4" style="margin-top: 20px">
      <!-- <h1 align="center">Login </h1> -->
      <form action="{?php echo base_url('Index/proses_login')} ?}" method='post'>
        <label>Username</label>
        <input type="username" name="username" autocomplete="off" class="form-control">

        <label>Password</label>
        <input type="password" name="password" autocomplete="off" class="form-control">
        <br><br>

        <div style="display: flex; justify-content: space-between; width: 100%;">
          <a href="lupapassword.php" target="_blank" class="btn btn-dark">Lupa Password</a>
          <button type="submit" class="btn btn-dark">Login</button>
        </div>

      </form>
    </div>
  </div>
</div>

```

Gambar 4. 47 Coding Login User

c. Halaman Ambil Antrian

Halaman Ambil Antrian adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pasien atau admin mengambil nomor antrian untuk layanan medis.

Gambar 4. 48 Halaman Ambil Antrian

```

<div class="container">
  <h2 class="page-section-heading text-center text-uppercase text-secondary mb-0">Ambil Antrian</h2>
  <div class="divider-custom">
    <div class="divider-custom-line"></div>
    <div class="divider-custom-icon">
      <i class="fas fa-star"></i>
    </div>
  </div>
</div>
<div class="row">
  <div class="container">
    <div class="row justify-content-md-center">
      <div class="col-md-12" style="margin-top: 20px">
        <form action="?php echo base_url('index/saveAntrian') ?>" method="post">
          <div class="row">
            <div class="col-md-2">
              <h6><label>Pilih Poli</label></h6>
            </div>
            <div class="col-md-5">
              <select name="id_poli" id="id_poli" class="form-control" onchange="noAntrian(this.value)">
                <option value="">Pilih</option>
                <?php foreach ($getPoli as $row) {
                  <option value="<?php echo $row->id_poli; ?>"><?php echo $row->kode_poli; ?> </option>
                <?php } ?>
              </select>
            </div>
          </div>
          <div class="row">
            <div class="col-md-2">
              <h6><label>No Antrian Poli</label></h6>
            </div>
            <div class="col-md-5">
              <input type="text" name="no_antrian_poli2" id="no_antrian_poli2" value="" disabled="" class="form-control">
              <input type="hidden" name="no_antrian_poli" id="no_antrian_poli" value="" class="form-control">
              <input type="hidden" name="no_antrian" value="<?php echo $no_antrian ?>">
            </div>
          </div>
          <div class="row text-right">
            <div class="col-md-7">
              <input type="submit" name="simpan" id="simpan" value="Ambil Antrian" class="btn btn-primary">
            </div>
          </div>
        </form>
      </div>
      <div class="row text-justify">
        <h6>Keterangan :</h6>
        <ol>
          <li>Poli Umum (PLUM)</li>
          <li>Poli Gigi (PLGG)</li>
          <li>Poli Kesehatan Ibu & Anak (PLKIA)</li>
        </ol>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>

```

Gambar 4. 49 Coding Halaman Ambil Antrian

```

public function saveAntrian()
{
    $id_pasien = $this->session->userdata('id_pasien');
    $id_poli = $this->input->post('id_poli');
    $no_antrian_poli_input = $this->input->post('no_antrian_poli');
    $tanggal = date("Y-m-d");

    // Mengambil hanya angka dari no_antrian_poli
    $no_antrian_poli = preg_replace('/\D/', '', $no_antrian_poli_input);

    // Menyimpan data ke tabel antrian terlebih dahulu
    $no_antrian = $this->input->post('no_antrian');
    $this->db->set('id_pasien', $id_pasien);
    $this->db->set('tgl_antrian', $tanggal);
    $this->db->set('no_antrian', $no_antrian + 1);
    $this->db->insert('antrian');

    // Mengambil id_antrian yang baru saja dimasukkan
    $id_antrian = $this->db->insert_id();

    // Simpan data ke tabel antrian_poli
    $this->db->set('id_antrian', $id_antrian);
    $this->db->set('id_pasien', $id_pasien);
    $this->db->set('id_poli', $id_poli);
    $this->db->set('no_antrian_poli', $no_antrian_poli);
    $this->db->set('tgl_antrian_poli', $tanggal);
    $this->db->insert('antrian_poli');

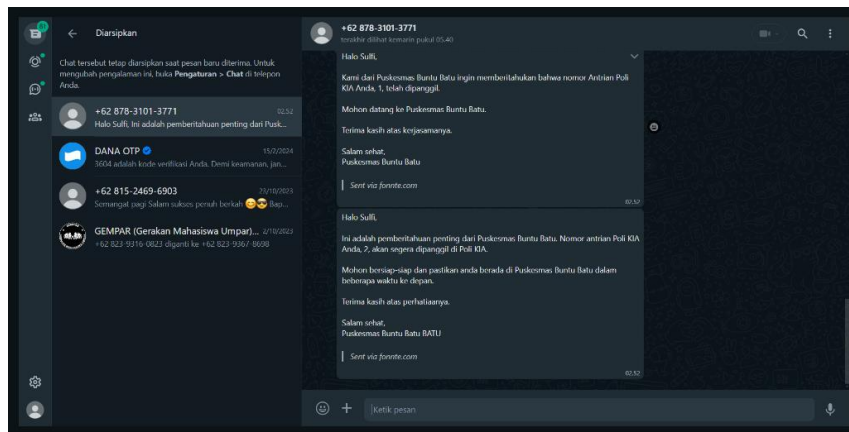
    redirect(base_url());
}

```

Gambar 4. 50 Coding Proses Ambil Antrian

d. Halaman Terima Pesan

Halaman Terima Pesan adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna menerima dan melihat pesan masuk dari system.



Gambar 4. 51 Halaman Terima Pesan

```
<?php
if (isset($_POST['kirimPesan'])) {
    // Token API
    $token = "jp+DZ6Hpkxy8_D#UP-y";

    // Ambil nomor telepon dan nama pasien dari form
    $target = $_POST['no_telp'];
    $namaPasien = $_POST['nama_pasien'];
    $noAntrian = $_POST['no_antrian'];

    $pesan = "Halo $namaPasien,\n\n"
        . "Kami dari Puskesmas Buntu Batu ingin memberitahukan bahwa nomor antrian Anda, $noAntrian, telah dipanggil.\n\n"
        . "Mohon datang ke Puskesmas Buntu Batu.\n\n"
        . "Terima kasih atas kerjasamanya.\n\n"
        . "Salam sehat,\n\n"
        . "Puskesmas Buntu Batu";

    $curl = curl_init();

    // Konfigurasi CURL untuk permintaan POST
    curl_setopt_array($curl, array(
        CURLOPT_URL => 'https://api.fonnte.com/send',
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => '',
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 0,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'POST',
        CURLOPT_POSTFIELDS => array(
            'target' => $target,
            'message' => $pesan,
        ),
        CURLOPT_HTTPHEADER => array(
            "Authorization: $token"
        ),
    ));
}
```

Gambar 4. 52 Coding kirim Pesan

```

// Eksekusi permintaan CURL
curl_exec($curl);
curl_close($curl);

// Redirect ke halaman lain setelah berhasil mengirim
// header("Location: antrian_kajian_awal");
// exit;
}
if (isset($_POST['kirimPesan'])) {
    // Token API
    $token = "jpiD7GHpkxy8_D#lP y";

    $target = $noTelpSelanjutnya;
    $namaPasien = $namaSelanjutnya;
    $noAntrian = $nomorAntrianSelanjutnya;

    $pesan = "Halo $namaPasien,\n\n"
        . "Ini adalah pemberitahuan penting dari Puskesmas Runtu Batu. Nomor antrian Anda, $noAntrian, akan segera dipanggil di kajian Awal.\n\n"
        . "Mohon bersiap-siap dan pastikan anda berada di Puskesmas Buntu Batu dalam beberapa waktu ke depan.\n\n"
        . "Terima kasih atas perhatiannya.\n\n"
        . "Salam sehat,\n\n"
        . "Puskesmas Buntu Batu BATU";
    $curl = curl_init();

    // Konfigurasi CURL untuk permintaan POST
    curl_setopt_array($curl, array(
        CURLOPT_URL => 'https://api.fonnte.com/send',
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => '',
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 0,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'POST',
        CURLOPT_POSTFIELDS => array(
            'target' => $target,
            'message' => $pesan,
        ),
        CURLOPT_HTTPHEADER => array(
            "Authorization: $token"
        ),
    ));

    // Eksekusi permintaan CURL
    curl_exec($curl);
    curl_close($curl);

    // Redirect ke halaman lain setelah berhasil mengirim
    header("Location: antrian_kajian_awal");
    // exit;
}
?>

```

Lanjutan Gambar 4.52

D. Pengujian Sistem

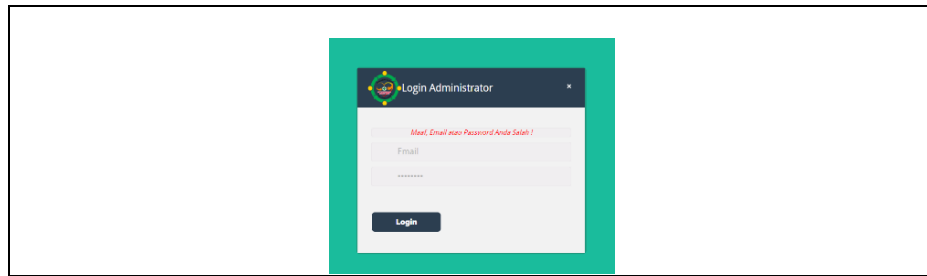
1. Black Box

a. *Black box testing* kesalahan *email* dan *password* login admin

Tabel 4. 7 *Black box testing* kesalahan *email* dan *password* login

Tes Faktor	Hasil	Keterangan
Memasukkan <i>email</i> atau <i>password</i> yang tidak sesuai	✓	Berhasil, ketika <i>email</i> atau <i>password</i> tidak sesuai maka <i>login failed</i>
Screenshot		

Lanjutan Tabel 4.7



b. *Black box testing login admin berhasil*

Tabel 4. 8 *Black box testing login admin berhasil*

Tes Faktor	Hasil	Keterangan
Memasukkan <i>email</i> atau <i>password</i> yang sesuai	✓	Berhasil, ketika <i>email</i> atau <i>password</i> sesuai maka tampil Halaman <i>Dashboard</i>
Screenshot		

c. *Black box testing tambah user admin*

Tabel 4. 9 *Black box testing tambah user admin*

Tes Faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol simpan	✓	Berhasil, maka tampil pemberitahuan tambah <i>user</i> admin berhasil
Screenshot		

Lanjutan Tabel 4.9

d. *Black box testing* tambah data pasien

Tabel 4. 10 *Black box testing* tambah data pasien

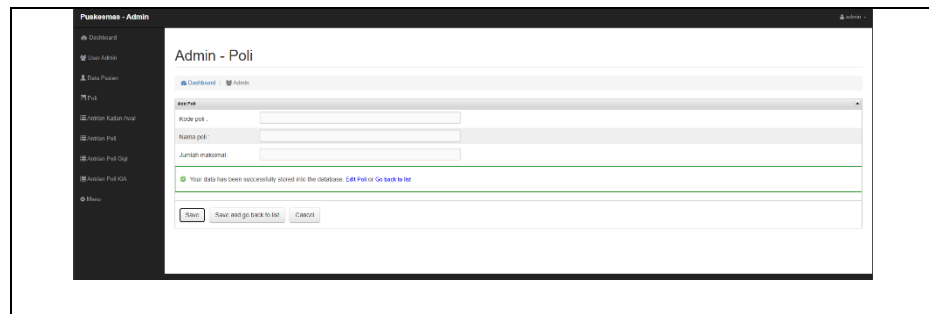
Tes Faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol simpan	✓	Berhasil,maka tampil pesan pemberitahuan data berhasil di tambahkan
Screenshot		

e. *Black box testing* tambah poli

Tabel 4. 11 *Black box testing* tambah poli

Tes Faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol simpan	✓	Berhasil,maka tampil pesan pemberitahuan data berhasil di tambahkan
Screenshot		

Lanjutan Tabel 4.11



f. *Black box testing* panggil antrian kajian awal

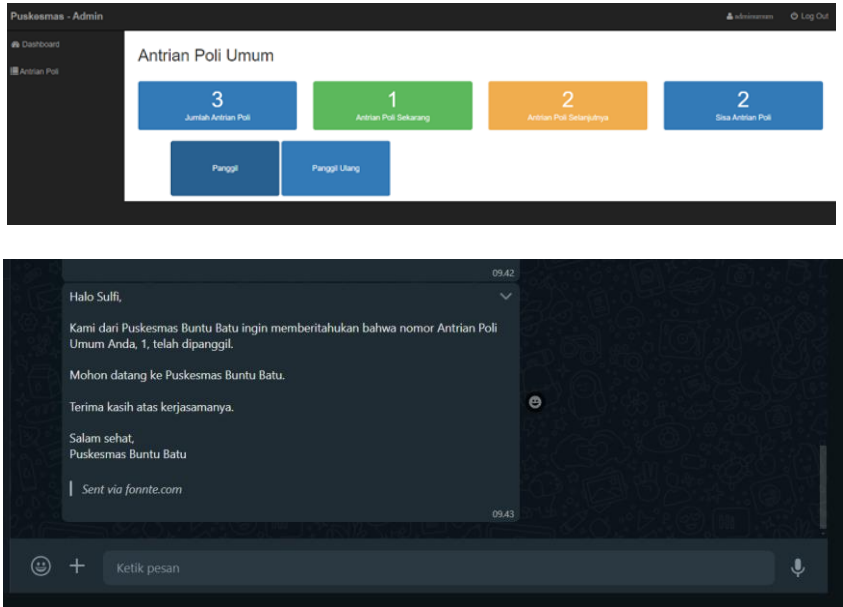
Tabel 4. 12 *Black box testing* panggil antrian kajian awal

Tes Faktor	Hasil	Keterangan
<i>Admin</i> menekan tombol Call	✓	Berhasil, maka akan mengirim pesan nomor antrian ke pasien
Screenshot		

g. *Black box testing* panggil antrian poli umum

Tabel 4. 13 *Black box testing* panggil antrian poli umum

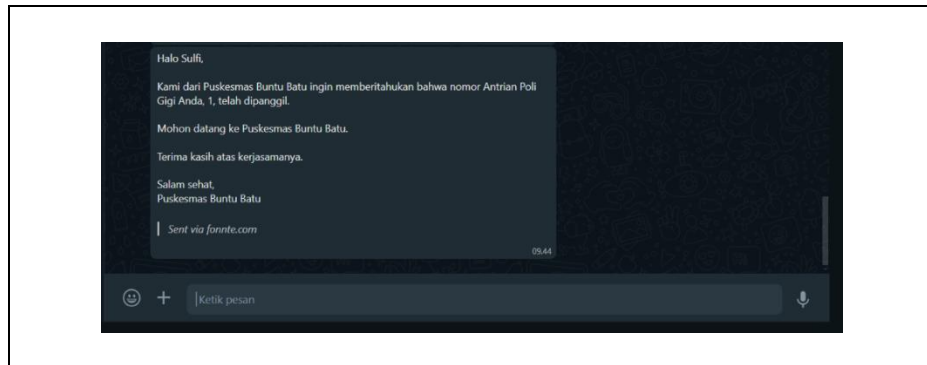
Tes Faktor	Hasil	Keterangan
------------	-------	------------

<i>Admin</i> menekan tombol Call	✓	Berhasil, maka akan mengirim pesan nomor antrian ke pasien
Screenshot		
		

h. *lack box testing* panggil antrian poli gigi

Tabel 4. 14 *Black box testing* panggil antrian poli gigi

Tes Faktor	Hasil	Keterangan
<i>Admin</i> menekan tombol Call	✓	Berhasil, maka akan mengirim pesan nomor antrian ke pasien
Screenshot		
		



i. **Black box testing** panggil antrian poli KIA

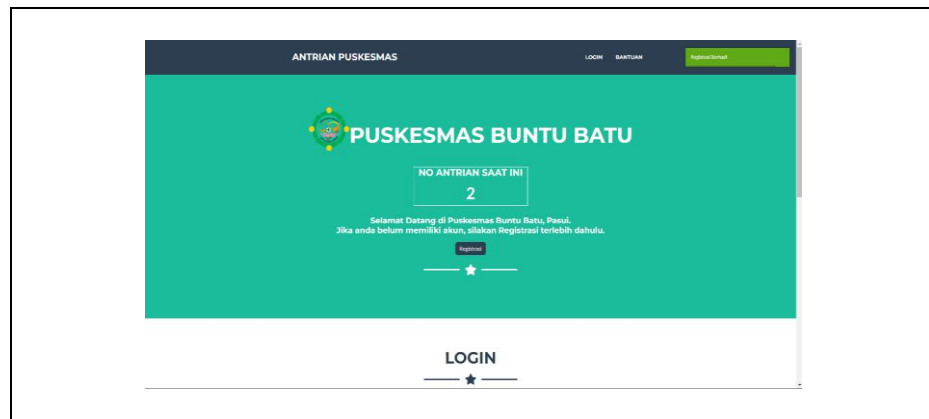
Tabel 4. 15 *Black box testing* panggil antrian poli KIA

Tes Faktor	Hasil	Keterangan
<i>Admin</i> menekan tombol Call	✓	Berhasil, maka akan mengirim pesan nomor antrian ke pasien
Screenshot		

j. **Black box testing** registrasi user

Tabel 4. 16 *Black box testing* registrasi user

Tes Faktor	Hasil	Keterangan
Memasukkan Data sesuai yang tercantum di format <i>registrasi</i>	✓	Berhasil, maka tampil pesan Registrasi Berhasil
Screenshot		



k. *Black box testing login berhasil*

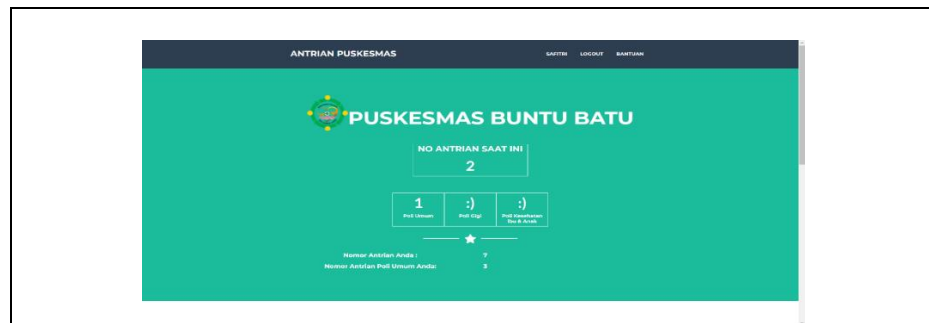
Tabel 4. 17 *Black box testing login berhasil*

Tes Faktor	Hasil	Keterangan
Memasukkan <i>email</i> atau <i>password</i> yang tidak sesuai	✓	Berhasil, ketika <i>email</i> atau <i>password</i> sesuai maka tampil Halaman <i>Dashboard</i>
Screenshot		

l. *Black box testing ambil antrian*

Tabel 4. 18 *Black box testing ambil antrian*

Tes Faktor	Hasil	Keterangan
<i>User</i> memilih poli dan menekan tombol ambil antrian	✓	Berhasil, maka tampil nomor antrian poli yang dipilih pasien
Screenshot		

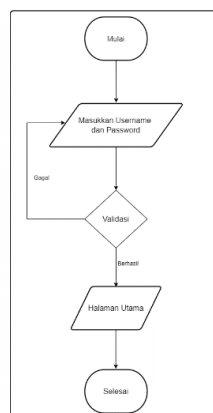


2. *White Box*

Pada metode pengujian ini akan ditampilkan *flowchart* dan *flowgraph* dari sistem yang telah dibuat. Berikut dibawah ini merupakan hasil pengujian menggunakan metode *white box testing*.

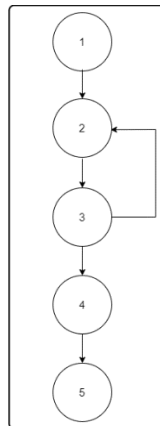
a. *White Box* kesalahan email dan password login admin

1) *Flowchart*



Gambar 4. 53 *Flowchart* kesalahan email dan password login

2) *Flowgraph*



Gambar 4. 54 *Flowgraph* kesalahan email dan *password* login

Berdasarkan gambar di atas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 1

Penyelesaian :

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

$\text{Predikat } (P) = P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari

flowgraph di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks kesalahan *username* atau *password*

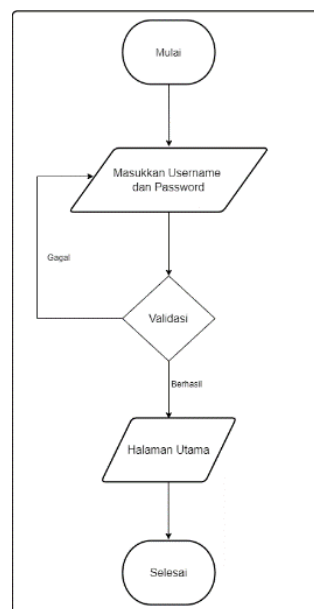
Tabel 4. 19 Grafik matriks kesalahan *username* atau *password*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$

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

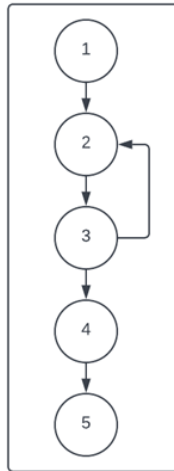
b. *White Box login admin berhasil*

1) *Flowchart*



Gambar 4. 55 *Flowchart login admin berhasil*

2) *Flowgraph*



Gambar 4. 56 *Flowgraph login admin berhasil*

Berdasarkan gambar di atas dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 1

Penyelesaian :

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

$\text{Predikat (P)} = P + 1$

$= 1 + 1$

$= 2$

- (2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 2

- (3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

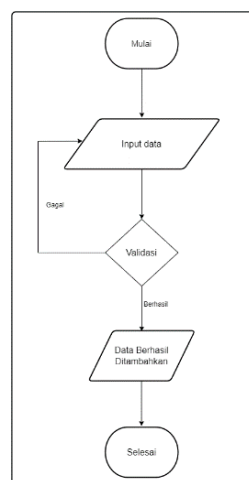
- (4) Grafik matriks *login* berhasil

Tabel 4. 20 Grafik matriks *login* berhasil

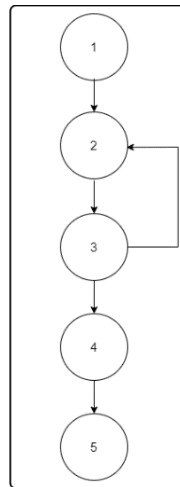
	1	2		3	4	5	E-1
1		1					$1 - 1 = 0$
2				1			$1 - 1 = 0$
3		1			1		$2 - 1 = 1$
4						1	$1 - 1 = 0$
5							0
		SUM (E + 1)					$1 + 1 = 2$

c. *White Box* tambah *user* admin

1) *Flowchart*

**Gambar 4. 57** *Flowchart* tambah *user* admin

2) *Flowgraph*



Gambar 4. 58 *Flowgraph* tambah *user* admin

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari

flowgraph di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

(4) Grafik matriks tambah *user* admin

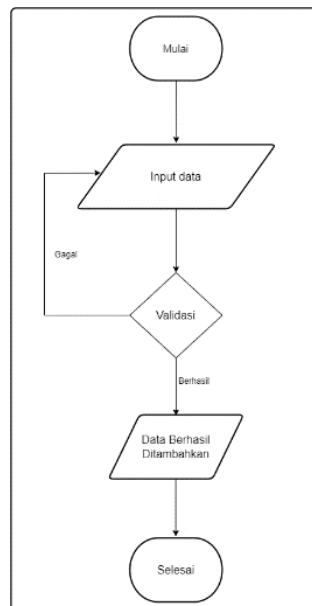
Tabel 4. 21 Grafik matriks tambah *user* admin

	1	2	3	4	5	E-1
--	---	---	---	---	---	-----

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

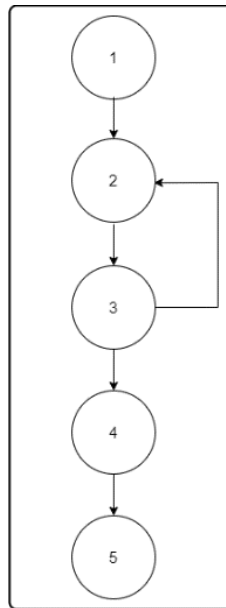
d. *White Box* tambah data pasien

1) *Flowchart*



Gambar 4. 59 *Flowchart* tambah data pasien

2) *Flowgraph*



Gambar 4. 60 *Flowgraph* tambah data pasien

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari

flowgraph di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

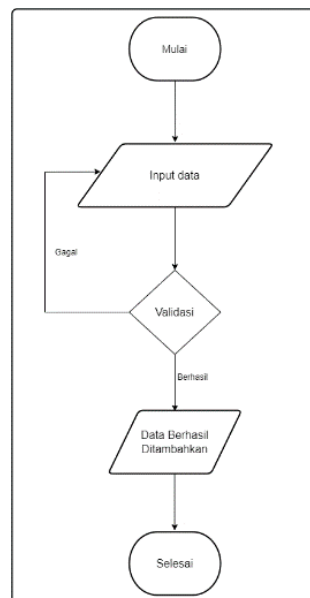
(4) Grafik matriks tambah data pasien

Tabel 4. 22 Grafik matriks tambah data pasien

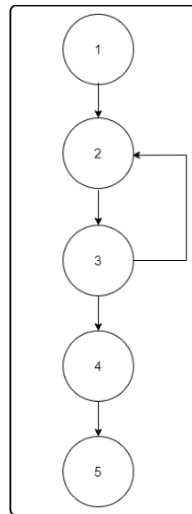
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

e. *White Box* tambah poli

1) *Flowchart*

**Gambar 4. 61** *Flowchart* tambah poli

2) *Flowgraph*



Gambar 4. 62 *Flowgraph* tambah poli

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

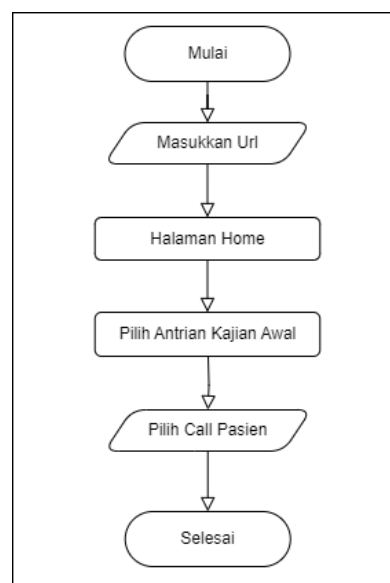
(4) Grafik matriks tambah poli

Tabel 4. 23 Grafik matriks tambah poli

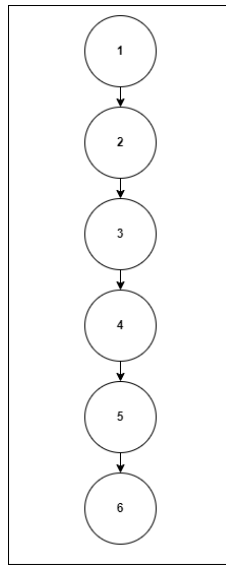
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

f. *White Box* panggil antrian kajian awal

1) *Flowchart*

**Gambar 4. 63** *Flowchart* panggil antrian kajian awal

2) *Flowgraph*



Gambar 4. 64 *Flowgraph* panggil antrian kajian awal

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

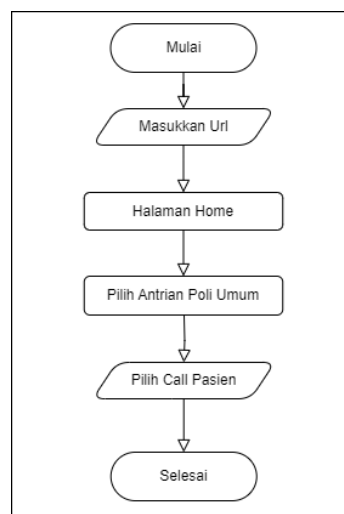
(4) Grafik matriks panggil antrian kajian awal

Tabel 4. 24 Grafik matriks panggil antrian kajian awal

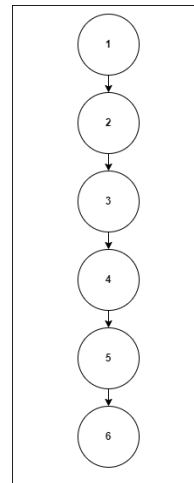
	1	2	3	4	5	6	E-1
1		1					$1 - 1 = 0$
2			1				$1 - 1 = 0$
3				1			$1 - 1 = 0$
4					1		$1 - 1 = 0$
5						1	$1 - 1 = 0$
6							0
	SUM (E + 1)						$0 + 1 = 1$

g. *White Box* panggil antrian poli umum

1) *Flowchart*



Gambar 4. 65 *Flowchart* panggil antrian poli umum

2) *Flowgraph***Gambar 4. 66** *Flowgraph* panggil antrian poli umum

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

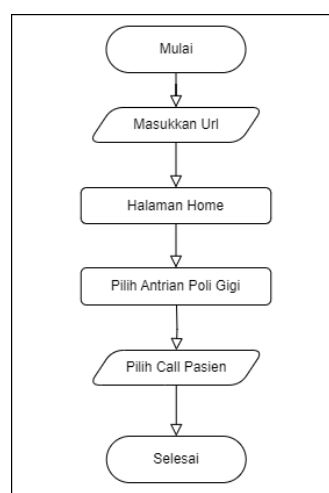
(4) Grafik matriks panggil antrian poli umum

Tabel 4. 25 Grafik matriks panggil antrian poli umum

	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		$1 - 1 = 0$
5						1	$1 - 1 = 0$
6							0
	SUM (E + 1)						$0 + 1 = 1$

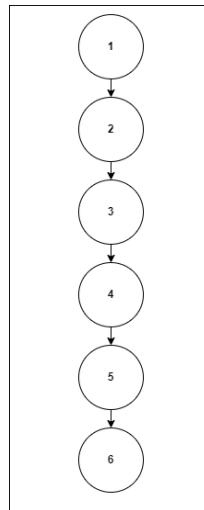
h. *White Box* panggil antrian poli gigi

1) *Flowchart*



Gambar 4. 67 *Flowchart* panggil antrian poli gigi

2) Flowgraph



Gambar 4. 68 Flowgraph panggil antrian poli gigi

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

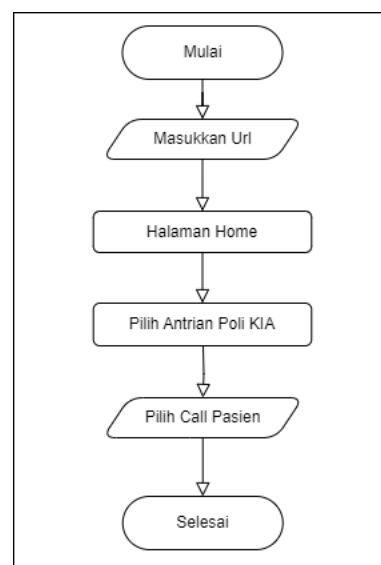
(4) Grafik matriks panggil antrian poli gigi

Tabel 4. 26 Grafik matriks panggil antrian poli gigi

	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		$1 - 1 = 0$
5						1	$1 - 1 = 0$
6							0
	SUM (E + 1)						$0 + 1 = 1$

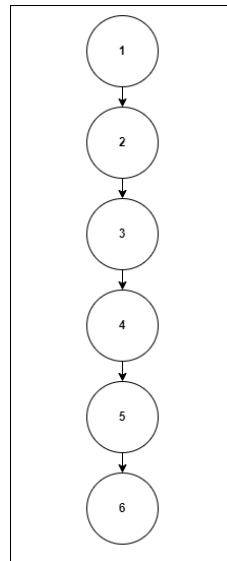
i. *White Box* panggil antrian poli KIA

1) *Flowchart*



Gambar 4. 69 *Flowchart* panggil antrian poli KIA

2) Flowgraph



Gambar 4. 70 Flowgraph panggil antrian poli KIA

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

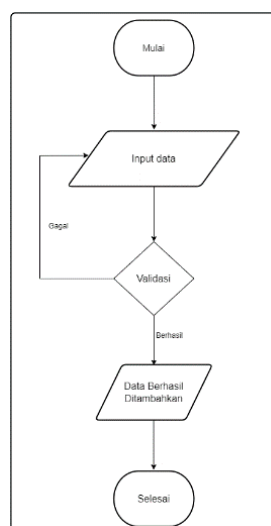
(4) Grafik matriks panggil antrian poli KIA

Tabel 4. 27 Grafik matriks panggil antrian poli KIA

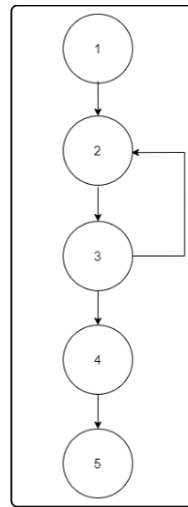
	1	2	3	4	5	6	E-1
1		1					$1 - 1 = 0$
2			1				$1 - 1 = 0$
3				1			$1 - 1 = 0$
4					1		$1 - 1 = 0$
5						1	$1 - 1 = 0$
6							0
	SUM (E + 1)						$0 + 1 = 1$

j. *White Box* registrasi user

1) *Flowchart*



Gambar 4. 71 *Flowchart* registrasi user

2) *Flowgraph***Gambar 4. 72** *Flowgraph* registrasi *user*

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari

flowgraph di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

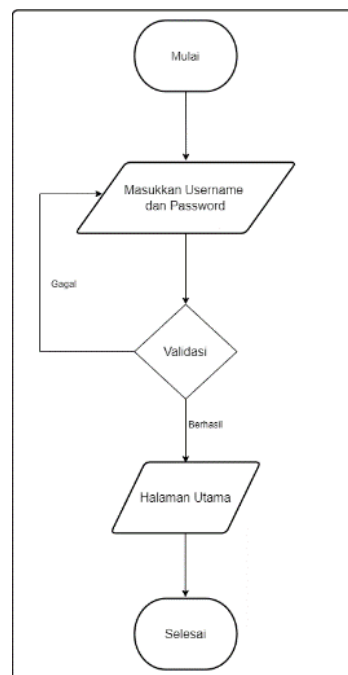
(4) Grafik matriks registrasi *User*

Tabel 4. 28 Grafik matriks registrasi *user*

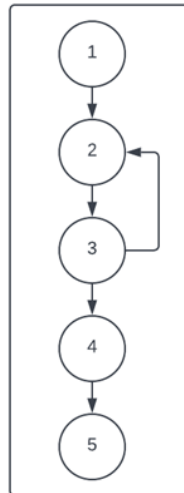
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

k. *White Box login user* berhasil

1) *Flowchart*

**Gambar 4. 73** *Flowchart login user* berhasil

2) *Flowgraph*



Gambar 4. 74 *Flowgraph login user*

Berdasarkan gambar di atas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 1

Penyelesaian :

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

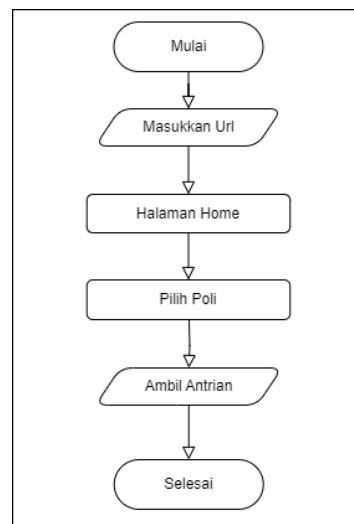
(4) Grafik matriks *login user* berhasil

Tabel 4. 29 Grafik matriks *login user* berhasil

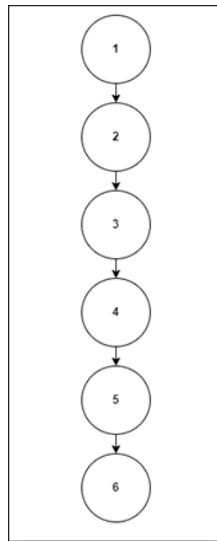
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

I. *White Box* ambil antrian

1) Flowchart

**Gambar 4. 75** *Flowchart* ambil antrian

2) Flowgraph



Gambar 4. 76 *Flowgraph* ambil antrian

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

BAB V

PENUTUP

A. Kesimpulan

1. Aplikasi Sistem Antrian Pasien Berbasis *Web* Dan Notifikasi Otomatis Melalui *Whatsapp* ini dibuat dengan menggunakan bahasa pemrograman *visual studio code* dan *mysql* sebagai *database* management sistem, telah melalui pengujian dengan baik menggunakan metode *blackbox testing* dan *whitebox testing*.
2. Aplikasi ini memiliki fitur nontifikasi melalui *whatsapp* ketika nomor antrian mendekati dan nomor antrian di panggil.
3. Fitur *notifikasi* otomatis melalui *WhatsApp* memberikan pemberitahuan terkait jadwal atau giliran antrian secara langsung kepada pasien.
4. Aplikasi ini menggunakan *fonnte.com* dalam menghubungkan *whatsapp*

B. Saran

Sebagai saran, tambahkan fitur untuk mengubah atau membatalkan pendaftaran antrian secara online dan pertimbangkan pengembangan aplikasi *mobile* untuk akses lebih mudah. Lakukan pengujian sistem secara berkala dan integrasikan dengan sistem manajemen Puskesmas lainnya untuk meningkatkan efisiensi. Berikan pelatihan kepada staf mengenai penggunaan sistem, dan rutin kumpulkan umpan balik pengguna untuk evaluasi dan perbaikan.

DAFTAR PUSTAKA

- 'Afiifah, K., Azzahra, Z. F., & Anggoro, A. D. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review. *Intech*, 3(2), 18–22. <https://doi.org/10.54895/intech.v3i2.1682>
- Abbas, H., Sajiah, Aresky, A. R., & Bhayangkara, A. (2022). Rancang Bangun Aplikasi Pemanggil Nomor Antrian Loker Pelayanan Bank Dan Akademik Universitas Islam Makassar. *Jurnal Teknologi Dan Komputer (JTEK)*, 2(01). <https://doi.org/10.56923/jtek.v2i01.61>
- Ahmad Martani, Saripuddin M, & Nurul Ikhsan. (2022). Rancang Bangun Website Company Profile Berbasis Framework Bootstrap dan Framework Codeigniter Pada Yayasan Khalifah Cendekia Mandiri. *Jurnal Multidisiplin Madani*, 2(6), 2895–2912. <https://doi.org/10.55927/mudima.v2i6.510>
- Annafri, M., Bijaksana Putra N, A., Priyanto, H., Yulianti, Sasty Pratiwi, H., & Candraningrum, N. (2023). Making a Website Using HTML and CSSfor SMKStudents. *Tijdessa*, Vol4(2), 9–13. <https://doi.org/10.26418/tijdessa.v4i2.41>
- Febriyanto, F., Sari, R. P., & Rasimin, E. (2023). *PENERAPAN WHATSAPP NOTIFICATION PADA SISTEM PENDAFTARAN ONLINE KLINIK PKU MUHAMMADIYAH KITAMURA PONTIANAK*. 27(1), 279–286. <https://doi.org/10.46984/sebatik.v27i1.1999>
- Hermiati, R., Asnawati, A., & Kanedi, I. (2021). Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database

Mysql.*JurnalMediaInfotama*17(1),54–66.

<https://doi.org/10.37676/jmi.v17i1.1317>

Ikhsan, M., Akbar, Z., Dani, R., & Ediansa, O. (2024). *Sosialisasi dan Pelatihan Framework Codeigniter Untuk Mahasiswa Universitas Muhammadiyah Jambi*. 2(1), 70–76. <https://doi.org/10.61132/aspirasi.v2i1.138>

James W, Elston D, T. J. et al. (20 C.E.). 済無No Title No Title No Title. *Andrew's Disease of the Skin Clinical Dermatology*.

Kuparinen, S. (2023). *Improving Web Performance by Optimizing Cascading StyleSheets(CSS):LiteratureReviewandEmpiricalFindings*.

<https://helda.helsinki.fi/items/abdda622-ffb8-4daf-a55a-a6d9deec4f91>

Maulidda, T. S., & Jaya, S. M. (2021). Perancangan Sistem Informasi Berbasis Web Melalui Whatsapp Gateway Studi Kasus Sekolah Luar Biasa-Bc Nurani. *JurnalTeknologiInformasiDanKomunikasi*,11(1),38–44.

<https://doi.org/10.56244/fiki.v11i1.421>

Nistrina, K., & Sahidah, L. (2022). Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil. *Jurnal Sistem Informasi, J-SIKA*, 4(1), 17–23.

Putri, D. H., & Ratnasari, C. I. (2021). Perancangan Aplikasi Mobile untuk Sistem Antrean Rumah Sakit Puri Husada Yogyakarta. *Automata*, 2(2).

<https://journal.uui.ac.id/AUTOMATA/article/view/19542>

Rahayu, D. K. P. (2020). *Unit Testing Pada Aplikasi Web Mobile(Studi Kasus BisnisJasaLaundry)*.1(1),1–79.

<https://dspace.uui.ac.id/bitstream/handle/123456789/28341/13523202Dwi>

Kusumastuti Puji Rahayu.pdf?sequence=1&isAllowed=y

- Yosli, R. (2021). Meningkatkan Kapasitas Hosting, Mengelola Content Management System Untuk Kenyamanan Memakai Website Berbayar. *JAVIT :JurnalVokasiInformatika*, 31–37. <https://doi.org/10.24036/javit.v1i2.6>
- Yunita Trimarsiah, & Muhajir Arafat. (2017). Analisis Dan Perancangan Website Sebagai Sarana Informasi Pada Lembaga Bahasa Kewirausahaan Dan Komputer Akmi Baturaja. *Jurnal Ilmiah MATRIK*, 19 No 1, 1–10.

