

**PEMBUATAN SISTEM APLIKASI ANTRIAN DENGAN  
METODE ALGORITMA FIFO UNTUK PENGATURAN  
PENDAFTARAN PASIEN DI APOTEK BIO MEDIKA  
DAN PRAKTEK DOKTER**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada program Studi Sistem Informasi**

Oleh

**Yadi Supriadi**

**352142002**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

**2023**

**PEMBUATAN SISTEM APLIKASI ANTRIAN DENGAN  
METODE ALGORITMA FIFO UNTUK PENGATURAN  
PENDAFTARAN PASIEN DI APOTEK BIO MEDIKA  
DAN PRAKTEK DOKTER**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada program Studi Sistem Informasi**

Oleh

**Yadi Supriadi**

**352142002**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

**2023**

**LEMBAR PENGESAHAN**

**PEMBUATAN SISTEM APLIKASI ANTRIAN DENGAN  
METODE ALGORITMA FIFO UNTUK PENGATURAN  
PENDAFTARAN PASIEN  
DI APOTEK BIO MEDIKA DAN PRAKTEK DOKTER**

***DEVELOPMENT OF A QUEUE APPLICATION SYSTEM USING FIFO  
ALGORITHM METHOD FOR PATIENT REGISTRATION MANAGEMENT  
AT BIO MEDIKA PHARMACY AND DOCTOR'S PRACTICE***


Oleh  
Yadi Supriadi  
352142002


Tugas Akhir ini telah diterima dan disahkan untuk  
memenuhi persyaratan mencapai gelar

SARJANA SISTEM INFORMASI

Pada  
PROGRAM STUDI SISTEM INFORMASI  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI

Bandung, Oktober 2023  
Disahkan oleh

  
Ketua Program Studi,  
Yudhi W. Ariharta R., S.T., M.Kom.  
NIDN.437300069

Dosen Pembimbing,  
  
Patah Herwanto, S.T., M.Kom  
NIDN. 0027107501

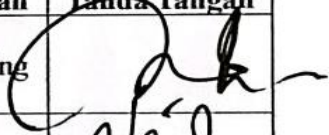
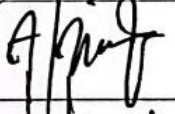
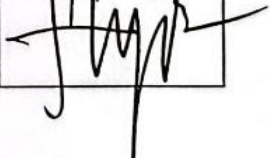
**LEMBAR PERSETUJUAN REVISI**

**PEMBUATAN SISTEM APLIKASI ANTRIAN DENGAN  
METODE ALGORITMA FIFO UNTUK PENGATURAN  
PENDAFTARAN PASIEN  
DI APOTEK BIO MEDIA DAN PRAKTEK DOKTER**

Oleh  
Yadi Supriadi  
352142002

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, Oktober 2023  
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Patah Herwanto, S.T., M.Kom	Pembimbing	
2.	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 1	
3.	Haryoso Wicaksono, S.Si., M.M., M.Kom.	Penguji 2	

Mengetahui  
Ketua Program Studi Sistem Informasi

  
Yudhi W. Arthana R., S.T., M.Kom.  
NIDN. 0421098704



## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Oktober 2023  
Yang membuat pernyataan



Yadi Supriadi  
352142002

## ABSTRAK

# PEMBUATAN SISTEM APLIKASI ANTRIAN DENGAN METODE ALGORITMA FIFO UNTUK PENGATURAN PENDAFTARAN PASIEN DI APOTEK BIO MEDIKA DAN PRAKTEK DOKTER

Oleh  
Yadi Supriadi  
352142002

Penelitian ini bertujuan untuk mengembangkan sistem antrian pasien di Apotek Bio Medika dan Praktek Dokter dengan metode Algoritma FIFO. Tujuan utamanya adalah menyederhanakan proses pelayanan kepada pasien, meningkatkan kecepatan pelayanan, dan memfasilitasi tugas Petugas Pendaftaran serta Asisten Dokter. Metode pengumpulan data melibatkan observasi langsung dan wawancara dengan pihak terkait. Implementasi menggunakan metode RAD untuk mempercepat proses pengembangan sistem. Hasil penelitian menunjukkan bahwa metode FIFO efektif dalam mengelola antrian pasien. Penggunaan RAD memungkinkan sistem dapat dikembangkan dengan cepat sesuai kebutuhan pengguna.

**Kata Kunci:** Sistem Antrian, Algoritma FIFO, Pengembangan Aplikasi, Rapid Application Development (RAD), Pelayanan Pasien.

## **ABSTRACT**

### **DEVELOPMENT OF A QUEUE APPLICATION SYSTEM USING FIFO ALGORITHM METHOD FOR PATIENT REGISTRATION MANAGEMENT AT BIO MEDIKA PHARMACY AND DOCTOR'S PRACTICE**

By

Yadi Supriadi

352142002

*This study aims to develop a patient queueing system at Bio Medika Pharmacy and Doctor's Practice using the FIFO algorithm. The primary objectives are to simplify patient service processes, enhance service speed, and facilitate the tasks of Registration Officers and Doctor's Assistants. Data collection methods involve direct observation and interviews with relevant parties. Implementation utilizes the RAD method to expedite system development. The research results demonstrate the effectiveness of the FIFO method in managing patient queues. The use of RAD enables the system to be developed swiftly according to user needs.*

**Keywords:** *Queueing System, FIFO Algorithm, Application Development, Rapid Application Development (RAD), Patient Service.*

## Kata Pengantar

Seiring dengan terselesaikannya skripsi ini, saya ingin menyampaikan penghargaan dan terima kasih yang tulus kepada pihak-pihak yang telah memberikan dukungan dan bimbingan dalam proses penelitian ini.

Pertama-tama, kepada Bapak Patah Herwanto, S.T., M.Kom, selaku pembimbing skripsi ini, saya sangat berterima kasih atas bimbingan, arahan, dan kesabaran yang diberikan. Tanpa bimbingan dari Bapak, skripsi ini tidak akan mencapai bentuk akhirnya.

Saya juga ingin mengucapkan terima kasih kepada Apotek Bio Medika dan Praktek Dokter atas izin dan kesempatan yang diberikan untuk melakukan penelitian di lingkungan mereka. Kontribusi dan dukungan dari pihak Apotek Bio Medika dan Praktek Dokter sangat berarti bagi kesuksesan penelitian ini.

Terima kasih juga saya sampaikan kepada pimpinan dan dosen program studi di STMIK IM atas atmosfer akademik yang memungkinkan saya untuk mengembangkan pengetahuan dan keterampilan saya.

Tidak lupa, terima kasih kepada semua pihak yang terlibat secara langsung dalam penelitian ini. Dukungan, masukan, dan kerjasama dari semua pihak sangat berarti bagi keberhasilan skripsi ini.

Saya berharap skripsi ini dapat memberikan kontribusi yang bermanfaat dalam pengembangan ilmu pengetahuan di masa depan. Semoga temuan dan rekomendasi dari penelitian ini dapat menjadi landasan untuk penelitian lebih lanjut dalam bidang terkait.

Akhir kata, saya menyampaikan rasa terima kasih yang sebesar-besarnya atas segala bantuan dan dukungan yang telah diberikan.

Bandung, 17 Oktober 2023,  
Program Studi Sistem Informasi



Yadi Supriadi  
NIM.352142002



## Daftar Isi

Lembar Pengesahan.....	i
Lembar Persetujuan Revisi .....	ii
Surat Pernyataan .....	iii
Abstrak .....	iv
Kata Pengantar.....	vi
Daftar Isi.....	vii
Daftar Tabel .....	x
Daftar Gambar .....	xi
1. Pendahuluan .....	1
1.1. Latar Belakang.....	1
1.2. Identitas Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Batasan Masalah .....	2
1.5. Metoda Penelitian .....	3
1.6. Sistematika Penulis .....	6
2. Landasan Teori .....	8
2.1. Konsep Dasar Sistem .....	8
2.2. Pengertian Sistem .....	8
2.3. Aplikasi .....	12
2.4. Antrian .....	13
2.4.1. Teori Antrian.....	14
2.4.2. Sistem Antrian.....	14
2.4.3. Disiplin Antrian.....	15
2.4.4. Model-Model Antrian .....	16
2.5. Algoritma FIFO .....	19
2.6. Basis Data .....	21
2.7. DBMS (Database Management System) .....	22

2.8. UML (Unified Modeling Language) .....	25
2.9. MySql .....	31
2.10. Kelebihan MySql .....	32
2.11. Cara Kerja MySql .....	37
2.12. Framework .....	38
2.13. Laravel .....	43
2.14. Black Box Testing .....	51
2.15. Pasien .....	53
2.16. Apotek .....	54
2.17. Praktik Dokter .....	56
3. Analisa Dan Desain Sistem .....	63
3.1. Analisa Sistem .....	63
3.1.1. Sistem Antrian Manual .....	64
3.1.2. Activity Diagram .....	65
3.1.3. Evaluasi Diagram Sistem Antrian Manual Yang Sedang Berjalan .....	67
3.2. Desain Sistem .....	68
3.2.1. Tujuan Desain Sistem .....	68
3.2.2. Gambaran Sistem Yang Diusulkan .....	69
3.2.3. Desain Output .....	74
3.2.4. Desain Input .....	75
3.2.5. Desain Tabel Database .....	77
4. Hasil Dan Evaluasi .....	81
4.1. Tampilan Hasil .....	81
4.1.1. Tampilan Dashboard .....	81
4.1.2. Tampilan Menu Login .....	82
4.1.3. Tampilan Display .....	83
4.1.4. Tampilan Ambil Antrian .....	84
4.1.5. Tampilan Panggil Pasien .....	85
4.1.6. Tampilan Setup Identitas Perusahaan .....	86
4.1.7. Tampilan Otoritas .....	86
4.1.8. Tampilan Ganti Password .....	89

4.1.9. Reset Nomor Antrian .....	90
4.2. Hasil Cetak Tiket Antrian.....	91
4.2.1. Tiket Antrian .....	91
4.3. Uji Coba Sistem.....	92
4.3.1. Skenario Pengujian .....	93
4.4. Proses FIFO.....	98
4.4.1. Presentasi Data .....	98
4.4.2. Analisa Data.....	101
4.4.3. Evaluasi Terhadap Tujuan Penelitian.....	102
5. Penutup .....	105
5.1. Kesimpulan .....	105
5.2. Sarah.....	106
Daftar Pustaka.....	107

## Daftar Tabel

Tabel 2.1 Alur Algoritma FIFO .....	20
Tabel 2.2 Tipe Data Angka.....	34
Tabel 2.3 Tipe Data Teks .....	36
Tabel 2.4 Tipe Data Date .....	36
Tabel 2.5 Tipe Data BLOB .....	37
Tabel 3.1 Evaluasi Sistem .....	67
Tabel 3.2 Peran Actor dalam Use Case .....	69
Tabel 3.3 Menu Login .....	77
Tabel 3.3 Menu Kunjungan .....	78
Tabel 3.4 Menu Display .....	79
Tabel 3.5 Menu Item Antrian .....	79
Tabel 3.6 Menu Jenis Item Antrian.....	80
Tabel 4.1 Hasil Pengujian Blackbox.....	93
Tabel 4.2 Pengujian Sistem Login.....	93
Tabel 4.3 Pengujian Otoritas Pengguna .....	94
Tabel 4.4 Pengujian Identitas Perusahaan .....	95
Tabel 4.5 Pengujian Ganti Password .....	96
Tabel 4.6 Pengujian Reset Antrian.....	97
Tabel 4.7 Pengujian Cetak Tiket Pasien.....	97
Tabel 4.8 Pengujian Cetak Tiket Admin.....	97
Tabel 4.9 Pengujian Display Antrian .....	97
Tabel 4.10 Pengujian Panggil Antrian.....	98
Tabel 4.11 Pengujian Panggil Ulang Antrian.....	98
Tabel 4.12 Data Waktu Antrian Pasien ke Pendaftaran .....	99
Tabel 4.13 Jumlah Pendaftaran Pasien Perhari.....	100

## Daftar Gambar

Gambar 1.1 Langkah-langkah dalam Model RAD .....	5
Gambar 2.1 Satu Jalur, Satu Tahap .....	16
Gambar 2.2 Satu Jalur Banyak Tahap .....	17
Gambar 2.3 Banyak Jalur Satu Tahap .....	18
Gambar 2.4 Banyak Jalur Banyak Tahap .....	19
Gambar 2.5 Logo MySQL .....	31
Gambar 2.6 Opsi Perintah Model .....	49
Gambar 2.7 Tampilan Welcome Screen .....	51
Gambar 2.8 Gambarkan proses pengujian Black Box Testing.....	52
Gambar 3.1 Pendaftaran Pasien.....	65
Gambar 3.2 Nomor Antrian .....	66
Gambar 3.3 Panggilan Antrian .....	66
Gambar 3.4 Use Case yang diusulkan .....	71
Gambar 3.5 Activity Diagram .....	72
Gambar 3.6 Sequence Diagram.....	73
Gambar 3.7 Class Diagram .....	74
Gambar 3.8 Tiket antrian .....	74
Gambar 3.9 Display utama .....	75
Gambar 3.10 Input form login.....	76
Gambar 3.11 Input form panggilan antrian .....	76
Gambar 3.12 Input form ambil antrian front office .....	76
Gambar 3.13 Input form ambil antrian back office.....	77
Gambar 4.1 Dashboard.....	82
Gambar 4.2 Tampilan Login .....	82

Gambar 4.3 Tampilan Display.....	83
Gambar 4.4 Tampilan Front Office.....	84
Gambar 4.5 Tampilan Back Office .....	85
Gambar 4.6 Tampilan Panggil Antrian.....	85
Gambar 4.7 Tampilan Setup Identitas Perusahaan.....	86
Gambar 4.8 Tampilan Otoritas .....	87
Gambar 4.9 Tampilan Tambah .....	88
Gambar 4.10 Tampilan Ubah .....	88
Gambar 4.11 Tampilan Hapus.....	89
Gambar 4.12 Tampilan Ganti Password.....	90
Gambar 4.13 Reset Nomor Antrian .....	91
Gambar 4.14 Tampilan Cetak Tiket Antrian.....	92
Gambar 4.15 Waktu Antrian Setiap Pasien Ke Pendaftaran .....	99
Gambar 4.16 Jumlah Pendaftaran Harian Pasien.....	100

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Saat ini, Indonesia memiliki berbagai layanan yang penting, salah satunya adalah layanan kesehatan yang memiliki peran sentral dalam masyarakat. Pasien memerlukan layanan kesehatan yang optimal karena berhubungan dengan masalah kesehatan mereka. Praktek dokter menjadi tujuan bagi pasien untuk mendapatkan perawatan medis. Antrian dalam sistem pelayanan menjadi cara untuk memberikan kenyamanan kepada pasien, dengan usaha untuk memberikan layanan yang cepat sehingga waktu menunggu dapat diminimalkan.

Namun, dalam prakteknya, terdapat beberapa tantangan. Tempat praktek dokter sering mengalami antrean panjang karena jumlah pasien yang memerlukan pelayanan melebihi fasilitas yang ada. Hal ini sering kali menimbulkan ketidaknyamanan bagi pasien, bahkan ada yang membatalkan mendaftar karena tidak ingin menunggu terlalu lama. Sama halnya dengan Apotek Bio Medika dan Praktek Dokter di Kota Soreang, sistem antrian yang saat ini digunakan masih manual dan kadang menyebabkan keluhan dari pasien. Nomor antrian sering hilang atau tidak terkoordinasi dengan baik, menyebabkan pasien merasa tidak nyaman.

Mengatasi permasalahan ini, penulis merencanakan pembuatan aplikasi berjudul "Pembuatan Sistem Aplikasi Antrian dengan Metode Algoritma FIFO untuk Pengaturan Pendaftaran Pasien di Apotek Bio Medika dan Praktek Dokter."

Aplikasi ini akan berbasis offline dan menerapkan algoritma FIFO untuk membantu pengaturan antrian dan informasi pelayanan. Dengan sistem ini, diharapkan pasien akan lebih mudah dalam mengatur waktu dan antrian mereka saat berobat.

## **1.2. Identifikasi Masalah**

Berdasarkan konteks masalah yang telah dijabarkan sebelumnya, peneliti akan melakukan identifikasi terhadap suatu masalah, yakni kebutuhan akan pengembangan perangkat lunak sistem antrian yang lebih efisien dan optimal. Tujuannya adalah agar antrian dapat diatur dan dikelola dengan lebih baik melalui penggunaan aplikasi, sehingga proses antrian menjalankan tugas dengan teratur dan efektif.

## **1.3. Tujuan Penelitian**

Sasaran utama dengan pengembangan sistem antrian pasien di Apotek Bio Medika dan Praktek Dokter adalah untuk menyederhanakan proses pelayanan kepada pasien, meningkatkan kecepatan pelayanan kepada pasien, serta memfasilitasi tugas Petugas Pendaftaran atau Asisten Dokter.

## **1.4. Batasan Masalah**

Agar tidak meluasnya cakupan pembahasan di masa mendatang, penulis telah mengatur batasan masalah sebagai berikut:



1. Model sistem antrian yang dikembangkan oleh peneliti akan difokuskan pada langkah-langkah pencatatan nomor antrian masuk dan panggilan nomor antrian oleh petugas layanan front office kepada pasien.
2. Pengembangan sistem ini akan mencakup tahapan pembuatan aplikasi hingga tahap pencetakan tiket.
3. Aplikasi ini akan dirancang untuk digunakan oleh dua tipe pengguna, yaitu administrator dan petugas bagian pendaftaran.
4. Penggunaan suara panggilan nomor antrian akan diimplementasikan dalam format .mp3.

### **1.5. Metoda Penelitian**

Informasi dalam penelitian ini diperoleh melalui dua teknik pengumpulan data, yaitu pendekatan tatap muka di lokasi penelitian serta wawancara dengan Kepala Apotek dan Dokter Praktek. Metode observasi digunakan untuk memantau keadaan di tempat penelitian secara langsung, sementara wawancara bertujuan untuk meraih informasi yang sah dari individu yang berpengetahuan. Setelah data berhasil terkumpul, langkah berikutnya adalah menganalisisnya untuk mengidentifikasi permasalahan yang ada. Data kemudian diolah guna menjadi dasar dalam pengembangan aplikasi sistem antrian berbasis algoritma FIFO untuk mengatur pendaftaran pasien di Apotek dan Praktek Dokter Bio Medika.

#### **A. Metode FIFO (First In First Out)**

Metode FIFO menyajikan pendekatan yang sederhana namun efektif terhadap mengatasi permasalahan antrian pada layanan publik. Prinsip dasar dari metode ini

adalah memberikan prioritas kepada antrian yang datang lebih awal. Proses perhitungan berdasarkan urutan masuk data, sehingga data yang tiba lebih awal akan diolah terlebih dahulu. Jika terdapat situasi di mana beberapa proses datang secara bersamaan, pelayanan akan dilakukan sesuai dengan urutan dalam antrian. FIFO menjunjung prinsip kesetiaan pada urutan masuk, di mana yang pertama kali tiba akan menjadi prioritas dalam pelayanan, begitu seterusnya.

## **B. Metode Pengembangan Sistem**

Penelitian ini menggunakan pendekatan pengembangan sistem RAD (Rapid Application Development) yang memprioritaskan kecepatan. Secara normal, pengembangan sistem memerlukan waktu minimal 180 hari. Namun, dengan metode RAD, sistem dapat diimplementasikan dalam rentang waktu 30-90 hari, dengan tujuan utama memenuhi harapan pengguna. Salah satu ciri khas metode RAD adalah melibatkan pengguna dalam proses pengembangan sistem. Hal ini berbeda dengan pendekatan tradisional di mana pengguna seringkali tidak terlibat secara langsung dalam pengembangan sistem dan dapat menyebabkan penolakan atau ketidakcocokan dengan sistem yang dihasilkan. Dengan menerapkan RAD, pengguna dapat aktif terlibat dalam setiap tahap pengembangan sistem. Metode RAD juga memiliki keunggulan dalam menghasilkan sistem dengan cepat, karena pengembangan sistem dapat lebih akurat sesuai dengan kebutuhan dan keinginan pengguna, mengurangi kemungkinan pengembangan ulang setelah tahap implementasi. Berikut adalah tahapan yang umum dalam metode RAD.



Gambar 1.1 Langkah-langkah dalam Model RAD (Rapid Application Development) menurut Kendall (2010)

#### a. Perencanaan Syarat-Syarat

Secara keseluruhan, dalam fase Rencana Kebutuhan (Perencanaan Persyaratan), terjadi kolaborasi antara pengguna dan analis untuk mengidentifikasi tujuan sistem serta kebutuhan informasi yang diperlukan guna mencapai tujuan tersebut. Selama tahap ini, juga dilakukan pengumpulan dan analisis terhadap berbagai elemen seperti alat, bahan, sumber daya, biaya, dan data yang dibutuhkan. Yang paling penting dalam tahap ini adalah Partisipasi aktif dari semua pihak dalam proses perencanaan bertujuan untuk memastikan bahwa hasilnya dapat memenuhi kebutuhan dan harapan yang ada

#### b. Workshop Design RAD

Secara umum, pada tahap ini, keterlibatan aktif dari para pengguna memainkan peran penting dalam mencapai tujuan yang diinginkan. Tahap ini melibatkan proses perancangan dan penyempurnaan, dimana potensi perbaikan dilakukan apabila terdapat ketidaksesuaian antara desain yang dihasilkan oleh pengguna dan analis. Pengguna memiliki kesempatan untuk memberikan komentar

dan umpan balik sehubungan dengan desain, serta merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang telah disusun pada tahap sebelumnya. Hasil dari tahap ini adalah spesifikasi perangkat lunak yang mencakup aspek umum dari struktur sistem, organisasi data, dan elemen-elemen lainnya.

### **c. Implementation**

Tahap Implementasi melibatkan peran programmer dalam menerjemahkan desain sistem menjadi kode-kode pemrograman melalui proses pembuatan kode, yang dianalisis secara seksama sesuai dengan rencana rancangan yang telah dibuat oleh tim desainer sistem.

## **1.6 Sistematika Penulisan**

Sistematika penulisan diatur untuk memudahkan proses penyusunan skripsi ini, oleh karena itu penting untuk menetapkan suatu struktur penulisan yang efektif. Sistematika penulisannya adalah sebagai berikut :

### **BAB I PENDAHULUAN**

Bagian pendahuluan merincikan tentang latar belakang, pengenalan masalah, tujuan dari penelitian, batasan masalah yang diteliti, metode yang digunakan dalam penelitian, dan tata cara penyusunan tulisan.

## BAB II TINJAUAN PUSTAKA

Bagian ini mencakup uraian mengenai teori-teori dan metode yang diterapkan dalam penelitian, serta langkah-langkah yang diambil dalam perancangan dan pembuatan sistem.

## BAB III ANALISA DAN DESAIN SISTEM

Dalam bab ini, penulis melakukan analisis terhadap seluruh sistem dan juga mengevaluasi desain sistem yang telah diimplementasikan.

## BAB IV HASIL DAN EVALUASI

Bab ini akan menyajikan hasil dari setiap tahap penelitian, dimulai dari analisis, desain, hasil pengujian, dan implementasinya.

## BAB V PENUTUP

Bab ini akan menguraikan kesimpulan yang diambil dari seluruh penelitian yang telah dilakukan, serta memberikan saran-saran berdasarkan temuan dan hasil penelitian tersebut.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Sistem**

Biasanya, dalam kebanyakan organisasi, terdapat suatu sistem informasi yang digunakan untuk menghimpun, menyimpan, mengakses, dan mendistribusikan informasi sebagai bagian dari proses perancangan sistem informasi.

Prinsip mendasar dari sistem adalah sekelompok komponen yang dibuat oleh manusia dan berbasis komputer. Sistem ini bertujuan untuk mengelola data, melakukan penyimpanan, mengatur rencana, serta membentuk interaksi antara SDM (Sumber Daya Manusia) dan komputer. Semua ini dilakukan dengan tujuan mengubah masukan sistem menjadi keluaran yang sesuai, dan dengan demikian mencapai target yang telah diprogram dulunya.

#### **2.2. Pengertian Sistem**

Secara umum, sistem adalah kerjasama antara komponen dan elemen yang saling terkoneksi, teratur, dan berkolaborasi untuk mencapai sasaran tertentu.

Menurut pandangan Sutanto sebagaimana dikutip oleh Djahir dan Pratita (2015:6), sistem merujuk pada kelompok subsistem, bagian, atau komponen fisik dan nonfisik yang saling terkait dan bekerjasama secara serasi untuk mencapai sasaran tertentu. Dalam pandangan Mulyani (2016:2), sistem dapat dijelaskan

sebagai kumpulan sub-sistem dan elemen yang bekerja bersinergi dengan tujuan yang telah ditetapkan sebelumnya untuk menciptakan output. Hutahaean (2015:2) juga berpendapat bahwa sistem merupakan jaringan prosedur-prosedur yang saling berinteraksi dan bersatu dalam melaksanakan kegiatan atau mencapai tujuan tertentu.

Menganalisis pandangan ahli di atas, kesimpulannya bahwa sistem adalah kumpulan komponen fisik dan abstrak yang saling berinteraksi, termasuk subsistem, elemen, dan unsur-unsur jasmani dan tak-berbentuk, berkolaborasi secara serasi, dan bekerja bersinergi dalam jaringan prosedur-prosedur untuk mencapai tujuan atau menciptakan output yang telah ditetapkan sebelumnya.

#### **A. Karakteristik sistem**

Dalam upaya mengidentifikasi pembuatan sistem, penting untuk memahami karakteristik-karakteristik suatu sistem sebagaimana dijelaskan oleh Hutahaean (2015:3). Karakteristik-karakteristik tersebut mencakup elemen-elemen berikut:

##### 1. Komponen sistem

Merujuk pada elemen-elemen atau bagian-bagian yang membentuk sistem dan saling terkait untuk mencapai tujuan sistem.

##### 2. Batas sistem (*boundary*)

Menentukan batas atau lingkup sistem, membedakan antara elemen-elemen sistem dengan yang berada di luar sistem.

3. Lingkungan luar sistem (*environment*)  
Merupakan faktor-faktor di luar batasan sistem yang dapat mempengaruhi atau dipengaruhi oleh operasi sistem.
4. Penghubung sistem (*interface*)  
Media atau titik-titik kontak antara sistem dan lingkungannya atau dengan sub-sistem lainnya.
5. Masukkan sistem (*input*)  
Data, tenaga, atau sumber daya yang dimasukkan ke dalam sistem untuk diolah dan diubah menjadi hasil atau keluaran.
6. Keluaran sistem (*output*)  
Hasil atau dampak yang dihasilkan oleh sistem setelah mengolah masukan.
7. Pengolah sistem  
Komponen yang melakukan pemrosesan terhadap masukan untuk menghasilkan keluaran, bisa berupa perangkat keras atau perangkat lunak.
8. Sasaran sistem  
Target yang ingin dicapai oleh sistem melalui proses pengolahan masukan menjadi keluaran.

## **B. Klasifikasi sistem**

Sistem merupakan hasil integrasi antara berbagai komponen yang bekerja bersama untuk mencapai tujuan yang berbeda-beda dalam setiap situasi yang terjadi di dalamnya. Sebagai akibatnya, sistem dapat dikategorikan dari berbagai perspektif.



Klasifikasi sistem yang dikemukakan oleh Hutahaean (2015:6) dapat dijelaskan seperti di bawah ini:

1. Sistem Fisik dan Sistem Abstrak:
  - a. Sistem Fisik, sistem yang memiliki bentuk fisik yang dapat diamati dan diukur. Contoh perangkat keras seperti monitor, keyboard, CPU, dan perangkat keras jaringan.
  - b. Sistem Abstrak, sistem yang lebih bersifat konseptual dan tidak memiliki bentuk fisik yang nyata. Contoh perangkat lunak seperti sistem operasi, aplikasi, dan algoritma.
2. Sistem Buatan Manusia dan Sistem Alamiah:
  - a. Sistem Buatan Manusia, sistem yang diciptakan atau dibangun oleh manusia. Contoh program komputer, aplikasi, dan perangkat lunak lainnya.
  - b. Sistem Alamiah, sistem yang ada secara alami di lingkungan tanpa campur tangan manusia. Contoh berupa ekosistem alam, jaringan ekologi, dan proses geologis.
3. Sistem Probabilistik dan Sistem Deterministik:
  - a. Sistem Probabilistik, sistem yang melibatkan elemen ketidakpastian atau keacakan. Contoh algoritma pengambilan keputusan berbasis probabilitas, seperti algoritma peringkat atau algoritma kecerdasan buatan yang mempertimbangkan peluang.

- b. Sistem Deterministik, sistem yang perilakunya dapat diprediksi dengan pasti berdasarkan aturan yang ditetapkan. Contoh berupa pergerakan planet dalam tata surya atau hukum fisika dalam reaksi kimia.
4. Sistem Tertutup dan Sistem Terbuka:
- a. Sistem Tertutup, sistem yang tidak berinteraksi secara signifikan dengan lingkungannya dan tidak ada pertukaran informasi atau energi. Contoh perangkat komputer standalone yang tidak terhubung ke jaringan atau internet.
  - b. Sistem Terbuka, sistem yang berinteraksi dengan lingkungan eksternal melalui pertukaran informasi, energi, atau materi. Contoh komputer yang terhubung ke internet, menerima data dari pengguna, dan memberikan hasil keluaran.

### **2.3. Aplikasi**

Dalam beberapa waktu terakhir, istilah "aplikasi" telah menjadi populer. Sesuai pandangan Jogiyanto HM (sebagaimana dikutip dalam Suhartini, 2017), "Aplikasi adalah penggunaan praktis dari konsep, pengumpulan informasi, situasi yang dihadapi, atau pekerjaan tertentu, yang diterapkan ke dalam suatu alat atau platform yang dapat dimanfaatkan untuk membentuk bentuk baru atau hasil yang inovatif."

Berdasarkan KBBI (Kamus Besar Bahasa Indonesia), "Aplikasi adalah implementasi dari perancangan sistem yang digunakan untuk memproses data dengan menggunakan aturan atau instruksi dari bahasa pemrograman tertentu.

Aplikasi merujuk pada sebuah program komputer yang diciptakan untuk melakukan tugas khusus sesuai dengan kebutuhan pengguna."

Maka, bisa ditarik kesimpulannya bahwa aplikasi mengacu pada gagasan perancangan sistem yang memiliki kemampuan khusus untuk melaksanakan tugas atau instruksi tertentu sesuai dengan rencana yang telah diatur dalam perancangan itu.

#### **2.4. Antrian**

Antrian, yang dalam bahasa Inggris disebut as *queueing* atau *waiting line*, adalah fenomena yang sering kita jumpai dalam kehidupan sehari-hari. Umumnya, hampir semua orang telah mengalami harus mengantri dalam sebuah barisan, seperti saat berada dalam jalur tunggu di fasilitas layanan umum atau lokasi lain sebelum menerima produk, jasa, atau pelayanan. Penyebab barisan dapat bermacam-macam, dan faktor umum yang menyebabkan antrian biasanya disebabkan oleh jumlah pelanggan yang melebihi kapasitas yang diperkirakan dari fasilitas yang tersedia untuk melayani mereka. Akibatnya, terjadi akumulasi pelanggan dan mereka harus menunggu sesaat sebelum dapat dilayani. Banyak usaha yang dijalankan untuk menangani antrian, mulai dari penambahan sarana pelayanan hingga pemanfaatan aplikasi pendaftaran antrian secara *offline* atau daring.

### **2.4.1. Teori Antrian**

Menurut Iqbal (2011:95), antrian terjadi ketika objek-objek menuju suatu area untuk mendapatkan pelayanan, namun mengalami keterlambatan karena mekanisme pelayanan sedang sibuk. Antrian muncul karena ada ketidakseimbangan antara yang memerlukan pelayanan dan ketersediaan pelayanan itu sendiri. Sementara itu, berdasarkan (Heizer dan Render, 2005) sebagaimana dicatat dalam (Sunu & Broto, 2017), antrian memiliki peran krusial dalam pengelolaan operasional. Sistem antrian dapat ditemukan di berbagai sektor, baik industri maupun layanan. Antrian dapat berupa rangkaian orang atau benda yang menanti ditangani dan kemudian meninggalkan antrian sesudah menerima pelayanan.

Maka, bisa ditarik kesimpulan teori antrian merujuk pada akumulasi individu yang terjadi ketika pelayanan terbatas. Antrian ini terbentuk dengan tujuan spesifik, yaitu untuk memperoleh produk, layanan, atau jasa yang diinginkan.

### **2.4.2. Sistem Antrian**

Berdasarkan perspektif yang diungkapkan oleh (Bambang, 2016), sistem antrian muncul sebagai hasil evolusi dari konsep teori antrian yang diciptakan untuk mengatur pemberian layanan sejalan dengan pola kedatangan, dengan tujuan mencapai optimalitas dan hasil yang efektif dalam kinerja, sebagai tindakan untuk menangani isu-isu terkait antrian.

Sementara berdasarkan pendapat (Bronson, 1991) sebagaimana yang dikutip oleh (Nia, Sugito dan Budi, 2016), sistem antrian merupakan sekumpulan

konsumen, *waiter*, dan peraturan yang mengelola pola kehadiran dan proses layanan.

Maka, dapat disimpulkan bahwa sistem antrian mengacu pada situasi di mana sekelompok pelanggan tiba untuk mendapatkan produk, layanan, atau pelayanan yang telah diatur oleh suatu mekanisme, sehingga proses pelayanan dapat berlangsung secara lebih optimal dan efisien.

### **2.4.3. Disiplin Antrian**

Menurut Siagian (1987), disiplin antrian adalah seperangkat pedoman keputusan yang digunakan untuk memilih individu yang akan dilayani terlebih dahulu dalam antrian. Siagian (1987) juga menyebutkan bahwa terdapat berbagai bentuk disiplin pelayanan yang dapat diterapkan, dimana disiplin pelayanan itu adalah sebagai berikut :

#### 1. FCFS atau *First-come First-Served*

Ini berarti bahwa dalam disiplin antrian, pelanggan yang tiba lebih awal akan dilayani lebih dulu dan meninggalkan antrian lebih awal juga. Contohnya, dalam antrian di loket pembelian tiket bioskop.

#### 2. LCFS atau *Last-Come First-Served*

Dalam konteks ini, prinsipnya adalah pelanggan yang tiba paling akhir akan dilayani dan meninggalkan antrian lebih awal. Sebagai contoh, ini dapat diterapkan pada sistem antrian di dalam lift untuk lantai yang sama.

#### 3. Disiplin Prioritas atau *Priority Service*

Dalam hal ini, pentingnya memberikan pelayanan terlebih dahulu kepada pelanggan dengan prioritas yang lebih tinggi daripada pelanggan dengan prioritas lebih rendah, bahkan jika pelanggan yang terakhir datang mungkin telah berada dalam antrian lebih lama.

#### 4. Disiplin Acak atau *Service In Random Order (SIRO)*

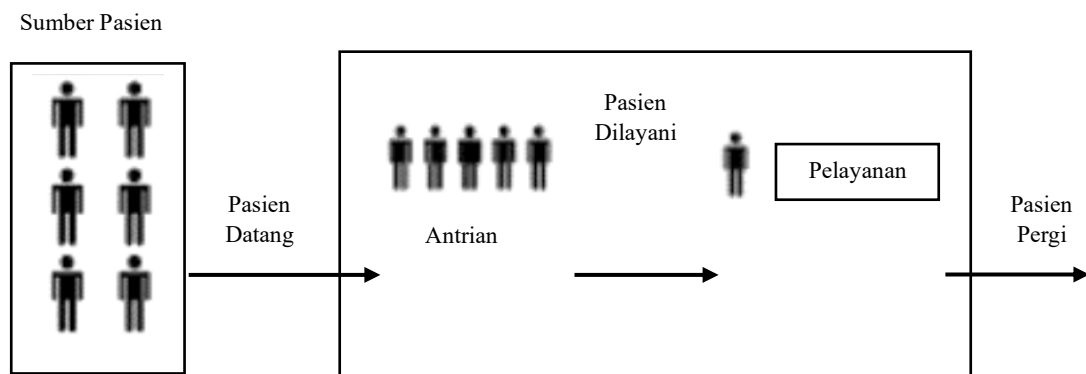
Dalam konteks ini, panggilan dilakukan secara acak berdasarkan peluang, tanpa memperhatikan urutan kedatangan pelanggan.

#### 2.4.4. Model-Model Antrian

Menurut (Christie, 2018), terdapat empat model dasar struktur antrian yang umum ditemui dalam berbagai sistem antrian:

##### 1. *Single Channel, Single Phase* (Satu Jalur, Satu Tahap)

Sistem antrian satu jalur satu tahap mengindikasikan bahwa dalam struktur tersebut hanya ada satu pemberi layanan dan jenis layanan yang disediakan, sehingga setelah menerima pelayanan, pelanggan dapat segera keluar dari sistem antrian.

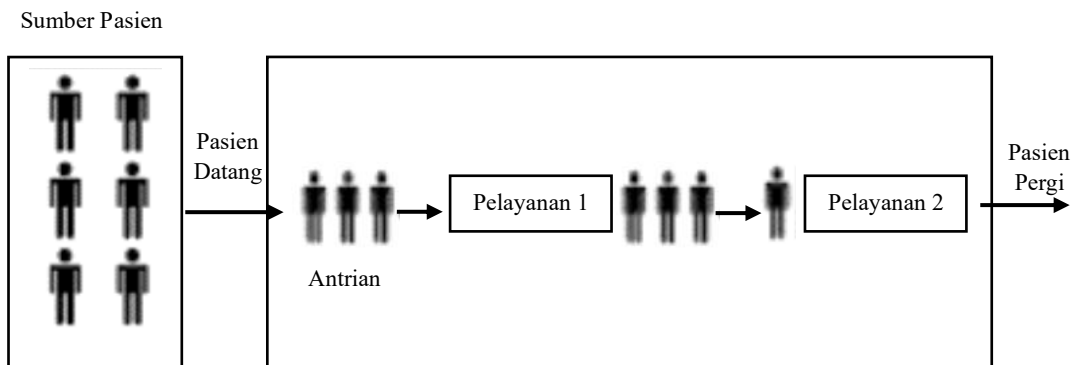


Gambar 2. 1 Satu Jalur, Satu Tahap

Gambar 2.1 menggambarkan rangkaian kedatangan pasien melalui satu jalur tunggal dan akan mendapatkan pelayanan dari satu fasilitas, setelah pelayanan selesai pasien dapat meninggalkan fasilitas tersebut.

## 2. *Single Channel, Multi Phase* (Satu Jalur, Banyak Tahap)

Sistem antrian satu saluran banyak tahap mengindikasikan bahwa dalam sistem antrian tersebut terdapat berbagai jenis layanan yang ditawarkan, namun dalam setiap jenis layanan hanya terdapat satu pemberi layanan.

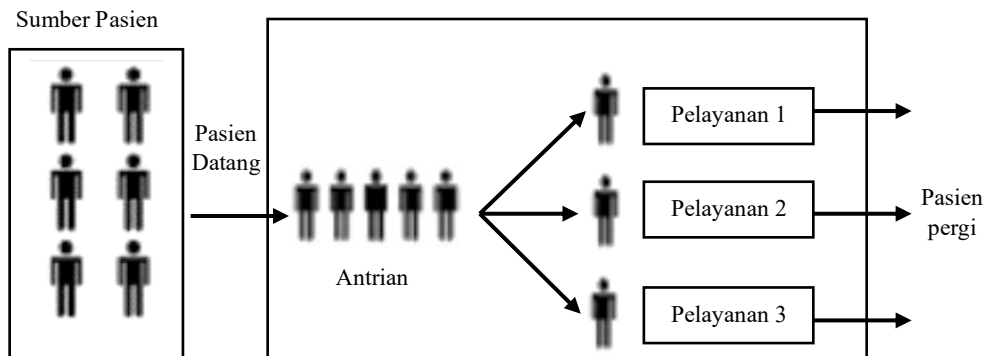


Gambar 2. 2 Satu Jalur Banyak tahap

Gambar 2.2 menjelaskan pola kedatangan pasien yang mengikuti satu jalur dalam sistem antrian. Pasien tersebut akan menerima pelayanan dari berbagai fasilitas yang ada dalam beberapa tahapan. Setelah melewati tahapan awal pelayanan, pasien akan diarahkan ke fasilitas berikutnya, dan seterusnya sampai seluruh tahapan pelayanan selesai. Setelah melewati seluruh tahapan pelayanan, pasien dapat meninggalkan fasilitas pelayanan.

### 3. *Multi Channel, Single Phase* (Banyak Jalur, Satu Tahap)

Sistem antrian banyak saluran satu tahap merujuk pada sebuah sistem antrian yang melibatkan satu jenis layanan, tetapi memiliki lebih dari satu pemberi layanan yang melakukan pelayanan tersebut.



Gambar 2. 3 Banyak Jalur, Satu Tahap

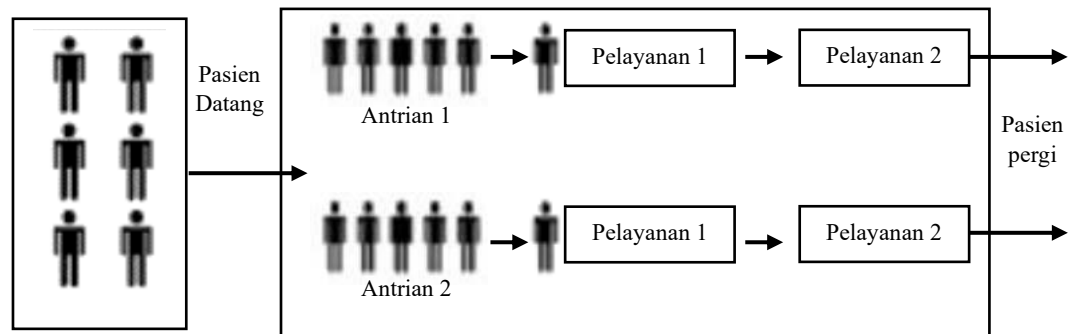
Gambar 2.3 menggambarkan bagaimana alur kedatangan pasien berada dalam satu jalur tunggal dan berbagai fasilitas pelayanan tersedia, walaupun dengan hanya satu tahap pelayanan. Meskipun pasien hanya melewati satu tahap pelayanan, jika ada fasilitas pelayanan lain yang tidak sibuk atau tidak sedang melayani pasien, fasilitas tersebut dapat memanggil pasien berikutnya untuk menerima layanan. Dengan demikian, pendekatan ini dapat membantu mengurangi kemungkinan terjadinya antrian.

### 4. *Multi Channel, Multi Phase* (Banyak Jalur, Banyak Tahap)

Sistem antrian banyak saluran banyak tahap adalah suatu sistem antrian di mana terdapat beragam jenis layanan yang disediakan, dan dalam setiap jenis layanan terdapat lebih dari satu pemberi layanan.

Sumber Pasien





Gambar 2. 4 Banyak Jalur, Banyak Tahap

Gambar 2.4 menggambarkan urutan kedatangan pasien melalui satu jalur yang melewati beberapa fasilitas pelayanan dalam beberapa tahapan. Pasien akan menghadapi berbagai fasilitas pelayanan, dan jika suatu fasilitas pelayanan sedang kosong atau tidak melayani pasien, fasilitas lain dapat memanggil pasien berikutnya untuk dilayani. Setelah selesai menerima layanan dari fasilitas pertama, pasien akan diarahkan ke tahapan fasilitas pelayanan berikutnya yang sesuai dengan kebutuhan mereka.

## 2.5. Algoritma Fifo

FIFO atau First In First Out adalah suatu algoritma dalam struktur data yang dikenal sebagai "Masuk Pertama, Keluar Pertama." Algoritma ini digunakan dalam konteks antrian atau pengolahan data, di mana barang atau elemen yang pertama kali masuk akan menjadi yang pertama keluar. Konsep ini bermanfaat dalam mengatur urutan pengolahan data, sehingga elemen yang pertama kali masuk akan diambil lebih awal daripada yang masuk belakangan (Aprilia, D., 2020).

FIFO juga dapat diartikan sebagai metode di mana pesanan atau proses yang masuk lebih awal akan dilayani atau dieksekusi terlebih dahulu. Dengan demikian,

proses yang masuk belakangan harus menunggu sampai proses yang pertama selesai. Algoritma FIFO juga mengandalkan prinsip "shortest job first," yang berarti setiap proses dalam antrian dipilih berdasarkan waktu eksekusi terpendek (Dwi Jayanto, et al., 2021).

Metode FIFO umumnya digunakan dalam bisnis yang menjual produk yang mudah rusak atau kadaluarsa, seperti buah-buahan dan sayuran. Algoritma ini sangat berguna untuk memastikan barang yang pertama masuk akan segera dieksekusi, (Handoko., 2022).

Kelebihan dari algoritma FIFO termasuk efisiensi kriteria yang sederhana. Algoritma ini juga menggunakan struktur data tumpukan (stack), di mana elemen yang berada di bagian depan antrian akan kembali ke memori lebih awal untuk menghindari kesalahan halaman.

**Tabel 2.1 Alur Algoritma FIFO**

8	4	2	6	9	7
↓	↓	↓	↓	↓	↓
8	8	8	8	4	4
	4	4	4	2	2
		2	2	6	6
			6	9	9
					7

Dalam tabel di atas, terdapat contoh penerapan algoritma FIFO, yang dapat dijelaskan sebagai berikut:

1. Pada angka 8 merupakan angka pertama yang akan dimasukan.
2. Setelah angka pertama diinput akan berada di atas.

3. Angka ke 4 merupakan angka yang kedua yang akan dimasukan, angka ini akan berada dibawah angka 8.
4. Kemudian masuk lagi angka 2, angka ini akan di bawah angka 4.
5. Angka 6 akan berada di urutan bawah nomer 2.
6. Dan jika angka 4 ingin naik, maka angka 8 harus dikeluarkan terlebih dahulu.
7. Dan seterusnya hingga ada angka baru yang masuk akan dimasukkan di antrian yang bawah.

## **2.6. Basis Data**

Data merupakan kenyataan yang menggambarkan suatu peristiwa yang terjadi pada suatu waktu tertentu. Data diperoleh dari peristiwa yang benar-benar terjadi, contohnya dari catatan transaksi penjualan, pembelian, dan sejenisnya. Data sering dihubungkan dengan bukti transaksi dalam suatu organisasi, seperti kwitansi, faktur, formulir, dan sejenisnya. Setelah data diproses, hasilnya dapat berubah menjadi informasi dalam bentuk laporan, seperti laporan keuangan atau laporan penjualan.

Basis data adalah kumpulan informasi yang dikumpulkan selama jangka waktu yang panjang, seringkali berlangsung selama bertahun-tahun. Basis data memiliki peran penting dalam semua jenis bisnis, baik dalam perusahaan besar maupun perusahaan kecil. Basis data bekerja di belakang layar untuk berbagai jenis perusahaan. Semua data penting perusahaan disimpan dalam basis data. Keunggulan basis data berasal dari perkembangan pengetahuan dan teknologi, yang diimplementasikan melalui perangkat lunak khusus yang dikenal sebagai sistem

manajemen basis data, atau Database Management System (DBMS), atau juga dikenal sebagai Sistem Basis Data.

## **2.7. DBMS (Database Management System)**

DBMS adalah merupakan alat yang efektif untuk menciptakan dan mengelola volume data yang besar dengan efisiensi, serta memastikan data tetap aman dan dapat bertahan dalam jangka waktu yang lama.

DBMS adalah kumpulan data yang saling terhubung dan dilengkapi dengan serangkaian program untuk mengaksesnya. Kumpulan data ini, yang sering disebut sebagai basis data, berisi informasi yang relevan dengan entitas seperti perusahaan. Tujuan utama dari DBMS adalah memberikan cara yang nyaman dan efisien untuk menyimpan serta mengambil informasi dari basis data tersebut.

Sistem basis data diciptakan untuk mengelola sejumlah besar informasi. Manajemen data melibatkan pembuatan struktur untuk menyimpan informasi dan menyediakan cara untuk mengolahnya. Selain itu, sistem basis data juga bertanggung jawab untuk memastikan keamanan data yang tersimpan, bahkan saat terjadi gangguan sistem atau upaya akses yang tidak sah. Jika data perlu dibagikan di antara beberapa pengguna, sistem harus mencegah terjadinya hasil yang tidak wajar.

DBMS diharapkan memiliki kemampuan untuk:

1. Memfasilitasi pengguna untuk membuat basis data baru dan merancang struktur basis data.

2. Memberikan pengguna kapabilitas untuk mengambil dan mengubah data dengan menggunakan bahasa kueri.
3. Mendukung penyimpanan data dalam jumlah besar dan dalam jangka waktu yang panjang.
4. Memungkinkan akses yang efisien ke data untuk permintaan dan modifikasi dalam basis data.
5. Menyediakan kemampuan pemulihan basis data dalam menghadapi situasi gagal, kesalahan, atau penyalahgunaan yang disengaja.
6. Mengelola pengaturan akses data dari banyak pengguna secara bersamaan.

Dalam praktiknya, ada beberapa jenis perangkat lunak Sistem Manajemen Basis Data (DBMS) yang umumnya digunakan untuk mengelola basis data perusahaan, antara lain:

1. MySQL

Banyak perusahaan mengadopsi MySQL karena tersedia secara gratis, menjadikannya pilihan yang cocok bagi bisnis yang sedang tumbuh. Walaupun tidak memiliki biaya lisensi, MySQL tetap memiliki tingkat keamanan yang dapat diandalkan dan memberikan akses data dengan konsistensi yang stabil. Namun, MySQL memiliki keterbatasan dalam kompatibilitas dengan bahasa pemrograman seperti Foxpro, Visual Basic (VB), dan Delphi, serta mengalami keterbatasan dalam menangani volume data yang besar.

## 2. Oracle

Oracle adalah perangkat lunak DBMS berbayar yang dianggap sangat berkualitas. Oracle menawarkan berbagai fitur yang mampu memenuhi kebutuhan fleksibilitas perusahaan besar. Bahkan, Oracle memiliki kemampuan pemrosesan transaksi dengan kinerja yang sangat optimal. Dalam hal keamanan, Oracle juga memiliki reputasi yang sangat baik dan dapat diandalkan. Dengan demikian, Oracle merupakan pilihan yang kuat untuk memenuhi kriteria yang diperlukan dalam konsep.

## 3. Microsoft SQL Server

Tidak hanya Oracle, Microsoft SQL Server juga merupakan opsi yang sesuai untuk diterapkan dalam jaringan komputer perusahaan besar karena kemampuannya dalam mengelola volume data yang besar. Microsoft SQL Server menonjol dengan sistem keamanan data yang efektif dan dilengkapi dengan fitur backup, pemulihan, dan rollback data. Namun, perlu diperhatikan bahwa perangkat ini hanya kompatibel dengan sistem operasi .

## 4. Firebird

Salah satu perangkat lunak DBMS lainnya adalah Firebird, yang merupakan sebuah sistem manajemen basis data relasional. Firebird menyediakan fitur sesuai dengan standar SQL-2003 dan ANSI SQL-99, serta dapat beroperasi pada berbagai sistem operasi seperti Windows dan Linux.

## 2.8. UML (Unified Modeling Language)

UML (Unified Modeling Language) adalah sebuah metode untuk melakukan pemodelan sistem. UML ditemukan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh. Versi terbaru dari UML, yaitu versi 2.5, mencakup lima belas jenis diagram. Diagram-diagram tersebut dikelompokkan menjadi dua kategori, yaitu diagram struktur dan diagram perilaku. Diagram struktur menggambarkan data dan hubungan statis dalam sebuah sistem informasi. Jenis diagram struktur meliputi *package*, *object*, *component*, *class*, *deployment*, *composite structure*, dan *profile diagram*. Sedangkan diagram perilaku menggambarkan interaksi dinamis antara objek-objek yang mewakili sistem informasi bisnis. Diagram perilaku meliputi *sequence*, *timing*, *interaction overview*, *activity*, *use case*, *protocol state machine*, *communication*, dan *behavior state machine diagram* (Dennis, Wixom, & Tegarden, 2015).

### A. Use Case Diagram

*Use case* digunakan untuk mengilustrasikan fungsi-fungsi pokok dari suatu sistem informasi. *Use case* menjelaskan bagaimana sistem bisnis berinteraksi dengan lingkungannya (Dennis et al., 2015). Di bawah ini terdapat unsur-unsur yang ada dalam *diagram use case*:

#### 1. Actor/ Role



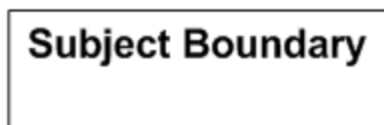
*Actor/ role* merupakan Individu atau entitas lain yang terlibat dalam interaksi dengan sistem saat ini..

## 2. *Use Case*



*Use case* adalah elemen sentral yang mewakili fungsionalitas utama dari sebuah sistem. *Use case* dapat diperluas melalui penggunaan *use case* lain dan ditempatkan dalam lingkup sistem. Biasanya, use case diberi label menggunakan kombinasi kata kerja dan frasa kata benda yang menggambarkan tindakan yang terjadi dalam sistem.

## 3. *Subject Boundary*



Isi dari *use case* adalah nama sistem yang ditempatkan di dalam atau di atas batas (*boundary*). Ini mencerminkan lingkup sistem. Aktor berada di luar batas sistem.

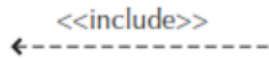
## 4. *Association Relationship*



Menggabungkan aktor dengan use case. Menunjukkan pertukaran informasi dua arah (Jika hanya satu arah, digambarkan dengan panah). Tanda "\*" menunjukkan variasi dari hubungan (*multiplicity of the association*). Tetapi, lazimnya hanya diilustrasikan menggunakan garis saja.



### 5. *Include Relationship*



Menyelipkan satu skenario pengguna ke dalam skenario pengguna lainnya. Kriteria perilaku yang harus terpenuhi untuk memungkinkan sebuah peristiwa terjadi, di mana kondisi ini merupakan komponen dari sebuah skenario pengguna lain. Panah mengarah dari skenario pengguna utama (induk) ke skenario pengguna yang tergabung.

### 6. *Extend Relationship*



Membangun skenario pengguna untuk mengakomodasi aksi opsional. Panah menghubungkan dari skenario pengguna tambahan ke skenario pengguna utama (induk).

### 7. *Generalization Relationship*



Merepresentasikan skenario pengguna khusus untuk skenario pengguna yang lebih umum. Arah panah menunjukkan hubungan dari skenario pengguna yang bersifat khusus ke skenario pengguna yang lebih umum.

## B. Activity Diagram

Diagram aktivitas menggambarkan aktivitas inti dan keterkaitan di antara aktivitas-aktivitas dalam suatu proses (Dennis et al., 2015). Berikut adalah komponen-komponen dari diagram aktivitas:

1. *Action*



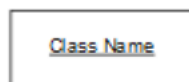
Tindakan yang ringkas dan tidak dapat didekomposisi (membagi atau mengurai sesuatu yang kompleks menjadi bagian-bagian yang lebih kecil atau lebih ringkas) lebih lanjut. Diberi label dengan nama.

2. *Activity*



Dipergunakan untuk menggambarkan kumpulan tindakan atau aksi. Diberi label dengan nama tindakannya.

3. *Object Node*

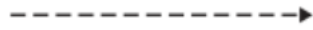


Dipergunakan untuk menggambarkan suatu objek yang terhubung dengan aliran objek tertentu. Diberi label dengan sebutan kelasnya.

4. *Control Flow*



Mengilustrasikan urutan pelaksanaan.

5. *Object Flow*

Mencitrakan peralihan suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lainnya.

6. *Initial Node*

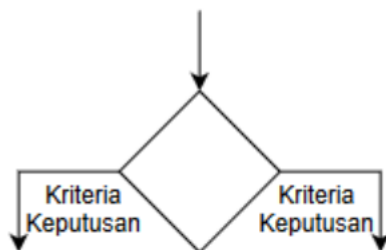
Menunjukkan titik *start* dari rangkaian aktivitas.

7. *Final-activity Node*

Berperan dalam menghentikan arus pengendalian dan objek dalam suatu kegiatan atau perbuatan.

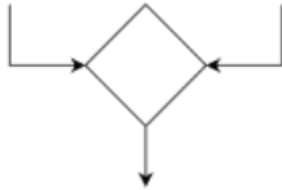
8. *Final-flow Node*

Pergunaan untuk menghentikan rangkaian kendali atau aliran objek khusus dalam suatu aktivitas atau tindakan.

9. *Decision Node*

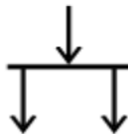
Digunakan untuk menggambarkan kondisi pengujian yang memastikan aliran kontrol atau aliran objek hanya mengikuti satu jalur tertentu. Elemen ini diberi label kriteria keputusan untuk menentukan jalur yang akan diambil.

10. *Merge Node*



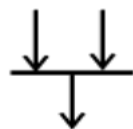
Menggunakan untuk menggabungkan kembali berbagai alur keputusan yang telah dibuat dengan memakai simpul penetapan.

11. *Fork Node*



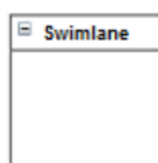
Pergunaan buat menguraikan perilaku menjadi rangkaian kegiatan yang berjalan sejajar atau seiring dengan kegiatan lainnya.

12. *Join Node*



Pergunaan untuk menyatukan kembali aliran aktivitas (atau tindakan) yang berjalan sejajar atau seiring.

13. *Swimlane*



Digunakan untuk menguraikan diagram aktivitas menjadi baris dan kolom, untuk mengaitkan kegiatan individu (atau tindakan) dengan individu atau objek yang bertanggung jawab atas pelaksanaan aktivitas tersebut. Diberi label dengan nama individu atau objek yang memiliki tanggung jawab.

## 2.9. MySQL



Gambar 2.5 Logo MySQL (<https://id.wikipedia.org/wiki/MySQL>,03/08/2023)

Secara keseluruhan, MySQL atau My Structure Query Language adalah sebuah server basis data yang memiliki kemampuan untuk mengelola basis data dengan efisien menggunakan bahasa SQL. Definisi MySQL menurut beberapa ahli berikut ini:

1. Budi Raharjo (2015:16) menjelaskan bahwa MySQL adalah sistem manajemen basis data relasional (RDBMS) yang mampu mengelola basis data dengan kecepatan tinggi, kapasitas penyimpanan yang besar, serta dapat diakses oleh banyak pengguna.

2. Heni A. Puspitosari (2011:19) mengungkapkan bahwa MySQL merupakan perangkat lunak server basis data yang populer dan bersifat Open Source, serta menggunakan bahasa SQL.
3. Agus Saputra (2012:77) mengakui bahwa MySQL merupakan database kelas dunia yang sangat cocok dikombinasikan dengan bahasa pemrograman PHP.
4. Arief (2011:151) mengidentifikasi MySQL sebagai jenis server basis data yang terkenal dan umum digunakan dalam pembangunan aplikasi web yang memerlukan manajemen basis data. MySQL bersifat open source dan menggunakan SQL, dan dapat berjalan di berbagai platform seperti Windows, Linux, dan lainnya.
5. Raharjo (2011:21) menggambarkan MySQL sebagai RDBMS (sistem manajemen basis data relasional) yang mampu mengelola basis data dengan cepat, menampung data dalam jumlah besar, serta dapat diakses oleh banyak pengguna.
6. Menurut Kadir (2008:2), MySQL merupakan perangkat lunak open source yang digunakan untuk membuat basis data.

### **2.10. Kelebihan MySQL**

MySQL adalah salah satu sistem manajemen basis data (DBMS) yang populer dan sering digunakan. Berikut adalah beberapa kelebihan MySQL:

1. Sumber Terbuka (Open Source), MySQL adalah perangkat lunak sumber terbuka yang dapat diunduh, digunakan, dan dimodifikasi tanpa biaya lisensi. Hal ini membuatnya menjadi pilihan ekonomis bagi banyak organisasi.

2. Kemudahan Penggunaan, MySQL memiliki antarmuka yang mudah digunakan dan intuitif, membuatnya cocok untuk pengguna dari berbagai latar belakang, termasuk pengembang, administrator basis data, dan pemula.
3. Performa Tinggi, MySQL dirancang untuk memberikan performa tinggi dalam pengolahan data, termasuk query yang cepat dan waktu respons yang rendah, menjadikannya cocok untuk aplikasi yang memerlukan kinerja tinggi.
4. Skalabilitas, MySQL dapat diatur untuk mengelola basis data dengan ukuran yang bervariasi, dari proyek kecil hingga aplikasi besar dan kompleks. Ini mendukung skala vertikal (menambahkan lebih banyak sumber daya pada satu server) dan skala horizontal (distribusi data ke beberapa server).
5. Keamanan, MySQL menyediakan fitur keamanan yang kuat, termasuk otentikasi pengguna dan enkripsi data. Dukungan SSL juga dapat diaktifkan untuk mengamankan komunikasi antara klien dan server.
6. Dukungan Transaksi, MySQL mendukung transaksi ACID (Atomicity, Consistency, Isolation, Durability) yang memastikan integritas data dalam lingkungan yang multi-pengguna dan multi-threaded.
7. Komunitas Besar dan Dokumentasi, MySQL memiliki komunitas yang besar dan aktif serta berbagai sumber daya dokumentasi, forum, dan tutorial yang tersedia, sehingga memudahkan pengguna untuk belajar dan memecahkan masalah.
8. Portabilitas, MySQL dapat dijalankan di berbagai sistem operasi seperti Windows, Linux, macOS, dan Unix, serta kompatibel dengan berbagai bahasa pemrograman dan aplikasi.

9. Dukungan Multi-User, MySQL mendukung koneksi dan interaksi simultan dari banyak pengguna, memungkinkan akses dan manipulasi data secara bersamaan.
10. Replikasi, MySQL mendukung replikasi yang memungkinkan data disalin ke beberapa server untuk keperluan keamanan, ketersediaan, atau analisis.
11. Basis Data Beragam, MySQL dapat digunakan untuk berbagai jenis aplikasi, mulai dari situs web dinamis hingga aplikasi perusahaan dan kecerdasan bisnis.

Tipe data merujuk pada kategori nilai yang dapat disimpan dalam variabel, mencakup angka (numerik), teks, atau bahkan gambar. Dengan menentukan tipe data, kita dapat lebih mudah mengatur struktur tabel yang sesuai dalam pengelolaan data.

Di bawah ini terdapat beberapa jenis tipe data yang digunakan dalam MySQL:

### 1. Tipe Data Angka (*Numerik*).

Tipe Data Angka (Numerik) adalah jenis tipe data yang digunakan untuk menyimpan nilai numerik dalam variabel konstan. Tipe data ini memiliki variasi yang berbeda, seperti yang tercantum di kolom tabel di bawah ini.

**Tabel 2.2 Tipe Data Angka (*Numeric*)**

<https://www.dumetschool.com/blog/tipe-tipe-data-pada-mysql, 29/11/2017>

Nama	Fungsi	Jangkauan	Ukuran
TINYINT	Menyimpan nilai angka bulat yang bisa (+) maupun (-).	-288 sampai dengan 127	1 <i>byte</i> (8 <i>bit</i> )
SMALLINT	Menyimpan nilai angka bulat yang bisa (+) maupun (-).	-32.768 sampai dengan 32.767	2 <i>byte</i> (16 <i>bit</i> )



MEDIUMINT	Menyimpan nilai angka bulat yang bisa (+) maupun (-).	-8.388.608 sampai dengan 8.388.607	3 <i>byte</i> (24 <i>bit</i> )
INT	Menyimpan nilai angka bulat yang bisa (+) maupun (-).	-2.147.483.648 sampai dengan 2.147.483.647	4 <i>byte</i> (32 <i>bit</i> )
BIGINT	Menyimpan nilai angka bulat yang bisa (+) maupun (-).	$\pm 9,22 \times 10^{18}$	8 <i>byte</i> (64 <i>bit</i> )
FLOAT	Memfaatkan tipe data angka tunggal presisi untuk menyimpan nilai pecahan baik yang (+) maupun (-).	- 3.402823466E+ 38 sampai dengan - 1.175494351E+ 38, 0, dan 1.175494351E- 38 sampai dengan 3.402823466E+ 38	4 <i>byte</i> (32 <i>bit</i> )
DOUBLE	Menyimpan data bilangan pecahan (+) dan (-) presisi ganda	-1.79...E+308 sampai dengan - .22...E-308, 0, dan 2.22...E- 308 sampai dengan 1.79...E+308	5 <i>byte</i> (64 <i>bit</i> )
REAL	Menggunakan tipe data angka presisi ganda untuk menyimpan nilai bilangan pecahan baik yang (+) maupun (-).	-1.79...E+308 sampai dengan - .22...E-308, 0, dan 2.22...E- 308 sampai dengan 1.79...E+308	6 <i>byte</i> (64 <i>bit</i> )
DECIMAL	Digunakan untuk menyimpan nilai bilangan pecahan baik yang (+) maupun (-).	-1.79...E+308 sampai dengan - .22...E-308, 0, dan 2.22...E- 308 sampai dengan 1.79...E+308	7 <i>byte</i> (64 <i>bit</i> )
NUMERIC	Digunakan untuk menyimpan nilai bilangan pecahan baik yang (+) maupun (-).	-1.79...E+308 sampai dengan - .22...E-308, 0, dan 2.22...E- 308 sampai	8 <i>byte</i> (64 <i>bit</i> )

		dengan 1.79...E+308	
--	--	------------------------	--

## 2. Tipe Data Teks (*String*).

Tipe Data Teks (*String*) adalah jenis tipe data yang digunakan untuk menyimpan sejumlah karakter dengan batas maksimum 255 karakter. Berikut adalah beberapa varian tipe data string yang ada dalam Database MySQL.

**Tabel 2.3 Tipe Data Teks (*String*)** <https://www.dumetschool.com/blog/tipe-tipe-data-pada-mysql>, 29/11/2017

Nama	Fungsi	Jangkauan
CHAR	Menyimpan rangkaian karakter dengan panjang tetap.	0 sampai dengan 255 karakter
VARCHAR	Menyimpan rangkaian karakter dengan panjang dinamis.	0 sampai dengan 65.535 karakter
TINYTEXT	Menyimpan teks atau rangkaian karakter dalam database.	0 sampai dengan 65.535 karakter
MEDIUMTEXT	Menyimpan teks atau rangkaian karakter dalam database.	0 sampai dengan 224 – 1 karakter
LONGTEXT	Menyimpan teks atau rangkaian karakter dalam database.	0 sampai dengan 224 – 1 karakter

## 3. Tipe Data Date.

Tipe Data Date digunakan untuk menyimpan informasi tanggal dengan format tahun, bulan, dan hari. Berikut adalah beberapa variasi jenis tipe data date yang tersedia.

**Tabel 2.4 Tipe Data Date** <https://www.dumetschool.com/blog/tipe-tipe-data-pada-mysql>, 29/11/2017

Nama	Fungsi	Jangkauan	Ukuran
DATE	Mengatur informasi terkait tanggal.	1000-01-01 sampai dengan 9999-12-31	3 <i>byte</i>

TIME	Mengatur informasi terkait waktu.	-832:59:59 sampai dengan 832:59:59 (HH:MM:SS)	3 <i>byte</i>
DATETIME	Mengatur informasi terkait tanggal dan waktu.	1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59	8 <i>byte</i>
YEAR	Mengatur informasi terkait tahun dari tanggal.	1900 sampai dengan 2155	1 <i>byte</i>

#### 4. Tipe Data BLOB.

Tipe Data BLOB digunakan untuk menyimpan berbagai jenis konten seperti video , gambar, musik, dan lain-lain.

**Tabel 2.5 Tipe Data BLOB <https://www.dumetschool.com/blog/tipe-tipe-data-pada-mysql>, 29/11/2017**

Nama	Fungsi	Jangkauan
BIT	Menyimpan informasi dalam format biner	64 digit biner
TINYBLOB	Menyimpan informasi dalam format biner atau <i>image</i> ukuran kecil	255 <i>byte</i>
BLOB	Menyimpan informasi dalam format biner atau <i>image</i>	65.535 <i>byte</i>
MEDIUMBLOB	Menyimpan informasi dalam format biner atau <i>image</i> ukuran sedang	16.777.215 <i>byte</i>
LOB	Menyimpan informasi dalam format biner atau <i>image</i> ukuran besar	4.294.967.295 <i>byte</i>

#### 2.11. Cara kerja MySQL

Server Perangkat Lunak MySQL terdiri dari beberapa elemen, termasuk server MySQL itu sendiri, berbagai program utilitas yang mendukung administrasi database MySQL, serta perangkat lunak pendukung lainnya yang terkait dengan server MySQL (meskipun rincian ini tidak selalu perlu diketahui oleh pengguna akhir). Inti dari seluruh sistem ini adalah server MySQL yang berfungsi sebagai

pengelola utama dari sistem basis data. Server MySQL bertanggung jawab atas semua instruksi yang terkait dengan basis data. Misalnya, jika kita ingin membuat basis data baru, kita akan mengirim pesan kepada server MySQL untuk memerintahkan pembuatan basis data dengan nama tertentu. Server MySQL akan menanggapi permintaan ini dengan membuat subdirektori khusus dalam direktori data, memberi nama sesuai dengan basis data yang baru dibuat, dan menempatkan file-file yang diperlukan dengan format yang sesuai di dalam subdirektori tersebut. Secara serupa, saat kita ingin menambahkan data ke dalam basis data, kita akan mengirim pesan ke server MySQL dengan data yang ingin ditambahkan beserta instruksi mengenai lokasi penambahan data.

Sebelum dapat memproses instruksi, server MySQL perlu diaktifkan dan siap menerima permintaan. Biasanya, server MySQL diatur untuk mulai berjalan saat komputer dinyalakan dan tetap aktif sepanjang waktu, terutama dalam pengaturan yang umum digunakan untuk situs web. Namun, kita juga memiliki fleksibilitas untuk mengaktifkan server MySQL secara manual sesuai kebutuhan, tanpa harus menjalankannya saat komputer dinyalakan. Selama server MySQL beroperasi, ia terus-menerus berada dalam mode penerimaan pesan, siap untuk merespons pesan yang dikirimkan kepadanya (Fujiama Diapoldo Silalahi,2022).

## **2.12. Framework**

Kerangka kerja (framework) merupakan struktur yang digunakan dalam pengembangan perangkat lunak untuk memberikan dasar, panduan, dan instrumen yang konsisten dalam membangun aplikasi. Framework membantu pengembang

dalam merencanakan, mengembangkan, dan menggabungkan berbagai komponen perangkat lunak secara lebih efektif.

Orang menggunakan framework dalam pengembangan perangkat lunak karena beberapa alasan yang signifikan yang membantu meningkatkan efisiensi, produktivitas, dan kualitas dalam proses pengembangan. Berikut beberapa alasan utama mengapa orang menggunakan framework:

1. Kerangka Kerja yang Terstruktur, Framework menyediakan struktur yang telah ditentukan sebelumnya untuk mengatur kode dan komponen. Ini membantu mengurangi kebingungan dan memastikan konsistensi dalam pengembangan.
2. Penggunaan Kembali Kode (*Code Reusability*), Framework menyediakan komponen, modul, dan pustaka siap pakai yang dapat digunakan kembali dalam proyek-proyek berbeda. Hal ini menghemat waktu dan usaha dalam pengembangan ulang komponen yang sama.
3. Penerapan Praktik Terbaik (*Best Practices*), Framework biasanya mengikuti praktik terbaik dalam desain dan pengembangan perangkat lunak. Ini membantu pengembang menghindari kesalahan umum dan menghasilkan kode yang lebih berkualitas.
4. Pemecahan Masalah yang Dibantu Framework (*Problem Solving*), Banyak masalah yang umumnya muncul dalam pengembangan perangkat lunak telah diatasi dalam framework. Ini membantu pengembang mengatasi kendala teknis dengan lebih cepat.

5. Efisiensi Pengembangan, Framework menyediakan alat-alat dan utilitas yang membantu dalam proses pengembangan, termasuk alat pengujian, alat migrasi basis data, dan lain-lain. Ini membantu meningkatkan produktivitas pengembang.
6. Keamanan, Beberapa framework memiliki mekanisme keamanan yang telah teruji untuk melindungi aplikasi dari ancaman keamanan umum seperti serbuan SQL injection atau cross-site scripting (XSS).
7. Pemeliharaan yang Lebih Mudah, Dengan menggunakan framework, pemeliharaan aplikasi menjadi lebih mudah karena kode diatur dengan baik, memungkinkan pembaruan atau perbaikan yang efisien.
8. Pengembangan Kolaboratif, Framework memudahkan pengembangan kolaboratif dalam tim, karena semua orang mengikuti struktur dan konvensi yang sama.
9. Mendukung Teknologi Terbaru, Banyak framework terus diperbarui untuk mendukung teknologi terbaru, bahasa pemrograman, dan tren pengembangan, memastikan aplikasi tetap relevan dan terkini.
10. Mengurangi Kebutuhan Pemrograman "Dari Nol" (*Boilerplate Code*), Framework membantu mengurangi kebutuhan untuk menulis kode dasar yang sering ditemukan dalam berbagai proyek, yang disebut "boilerplate code".

Struktur aplikasi dalam sebuah framework merupakan kerangka kerja organisasi yang telah ditentukan sebelumnya untuk mengatur bagaimana komponen-komponen perangkat lunak akan diatur, berinteraksi, dan berfungsi

bersama. Struktur ini membantu pengembang dalam merancang dan mengembangkan aplikasi dengan konsistensi, mempercepat proses pengembangan, dan meningkatkan pemeliharaan kode. Meskipun struktur dapat bervariasi antara framework yang berbeda, berikut adalah beberapa komponen umum yang biasanya ada dalam struktur aplikasi framework:

1. Direktori Utama (*Root Directory*), Ini adalah direktori utama proyek aplikasi yang berisi semua file dan direktori terkait aplikasi.
2. Direktori Konfigurasi (*Configuration Directory*), Direktori ini berisi file konfigurasi yang mengatur pengaturan umum aplikasi, seperti koneksi basis data, pengaturan lingkungan, dan parameter lainnya.
3. Direktori Kontrol (*Controller Directory*), Di banyak framework web, direktori ini berisi kontroler atau modul yang mengatur logika bisnis dan mengelola permintaan dari pengguna.
4. Direktori Model, Direktori ini berisi definisi model atau kelas yang merepresentasikan struktur data dan logika bisnis dalam aplikasi. Ini berkaitan dengan manipulasi data dalam basis data.
5. Direktori View, Direktori ini berisi file tampilan atau template yang digunakan untuk merender tampilan yang dilihat oleh pengguna.
6. Direktori Public, Direktori ini berisi berkas-berkas publik seperti gambar, file CSS, JavaScript, dan berkas statis lainnya yang dapat diakses langsung oleh klien.

7. Direktori Routing, Di beberapa framework, ini adalah tempat di mana kita mendefinisikan rute URL dan menghubungkannya dengan tindakan atau fungsi dalam kontroler.
8. Direktori Middlewares, Middlewares adalah komponen yang memproses permintaan sebelum mencapai kontroler. Direktori ini berisi kode middleware.
9. Direktori Migrasi, Di beberapa framework, migrasi digunakan untuk mengelola perubahan skema basis data.
10. Direktori Test, Direktori ini berisi berkas-berkas pengujian yang digunakan untuk menguji aplikasi.
11. Direktori Utilitas atau Libraries, Beberapa framework memiliki direktori yang berisi utilitas atau pustaka (*library*) yang dapat digunakan dalam proyek.
12. Direktori Konfigurasi Umum (*Common Configuration*), Beberapa framework menyediakan direktori khusus untuk mengatur konfigurasi umum yang dapat digunakan oleh berbagai bagian aplikasi.

Pendapat Hakim (2010:3), konsep Framework merujuk pada kumpulan bagian-bagian program yang telah terstruktur dan diatur dengan cara yang serupa, sehingga dapat dimanfaatkan untuk mempermudah pengembangan aplikasi lengkap tanpa perlu membangun seluruh kode dari nol.

Pendapat Raharjo (2015:2), Framework merujuk pada sekumpulan kode yang terdiri dari pustaka (*library*) dan alat (*tool*) yang digabungkan secara terstruktur menjadi satu kesatuan kerangka kerja (*framework*), dengan tujuan buat mempermudah dan mempercepat proses pengembangan aplikasi web.



Jadi, framework merupakan kerangka kerja atau struktur yang terstruktur dan diatur sedemikian rupa untuk memudahkan pengembangan perangkat lunak. Framework menyediakan landasan, pedoman, dan alat yang konsisten dalam membangun aplikasi. Penggunaan framework membantu meningkatkan efisiensi, produktivitas, dan kualitas dalam proses pengembangan dengan menyediakan komponen siap pakai, mengikuti praktik terbaik, membantu pemecahan masalah, dan lain-lain. Struktur aplikasi dalam sebuah framework mengatur bagaimana komponen-komponen perangkat lunak diatur, berinteraksi, dan berfungsi bersama, sehingga membantu dalam merancang, mengembangkan, dan mengintegrasikan komponen dengan lebih efisien.

### **2.13. Laravel**

Laravel adalah salah satu framework pengembangan aplikasi web berbasis PHP yang sangat populer. Diciptakan oleh Taylor Otwell, Laravel menyediakan berbagai alat dan fitur yang kuat untuk mempermudah proses pengembangan aplikasi web dengan cepat dan efisien. Framework ini berfokus pada penyederhanaan tugas-tugas umum dalam pengembangan web, seperti routing, pengelolaan sesi, interaksi dengan basis data, dan lain-lain.

MVC merupakan singkatan dari Model View Controller, yang merupakan sebuah model desain (design pattern) arsitektur perangkat lunak yang digunakan untuk memisahkan komponen-komponen utama dalam pengembangan aplikasi. Tujuannya adalah untuk memisahkan logika bisnis (model), tampilan (view), dan

pengendalian (controller) dalam sebuah aplikasi, sehingga memudahkan pengembangan, perawatan, dan perubahan.

Berikut penjelasan lebih rinci tentang setiap komponen dalam pola desain MVC:

1. Model, bagian ini memiliki tanggung jawab dalam mengurus data dan logika bisnis dalam aplikasi. Model menggambarkan struktur data, menyediakan akses ke basis data, dan menangani operasi terkait data seperti pembuatan, pembacaan, pembaruan, dan penghapusan. Model juga mengandung logika bisnis yang mengatur bagaimana data diolah dan berinteraksi.
2. View, komponen ini mengurus tampilan atau antarmuka pengguna dalam aplikasi. View bertanggung jawab untuk merender data dari model ke dalam bentuk yang dapat dilihat oleh pengguna. Ini dapat berupa halaman web, elemen antarmuka grafis, atau tampilan lainnya.
3. Controller, komponen ini bertanggung jawab untuk mengendalikan aliran logika dalam aplikasi. Controller menerima input dari pengguna melalui tampilan, memprosesnya menggunakan model yang sesuai, dan mengarahkannya ke tampilan yang tepat untuk ditampilkan kembali kepada pengguna. Controller mengatur bagaimana interaksi antara model dan view terjadi.

Laravel menawarkan tiga fitur utama yang umumnya digunakan, yaitu Laravel Homestead, Laracast, dan Laravel Cashier. Laravel Homestead mirip dengan XAMPP/LAMP, tetapi berbasis pada Vagrant dan Virtual Box. Berikut ini adalah beberapa definisi mengenai Laravel menurut beberapa pakar:

1. Sesuai dengan pemaparan Rahmat Awaludin (Awaludin, Menyelami Framework Laravel, 2016), seorang Senior Web Developer yang telah menguraikan setiap aspek dari kerangka kerja Laravel dalam karyanya "Menyelami Framework Laravel," mengemukakan bahwa Laravel adalah sebuah platform PHP yang diterbitkan di bawah lisensi MIT dan pertama kali dikembangkan oleh Taylor Otwell. Framework ini mengaplikasikan prinsip-prinsip MVC (Model View Controller) dan dimanfaatkan untuk pembuatan situs web berbasis MVP. Laravel ditulis dalam bahasa PHP dan bertujuan untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan pemeliharaan, serta untuk meningkatkan efisiensi dalam proses pengembangan aplikasi.
2. Dalam pandangan Aminudin (Aminudin, 2016), seorang pengarang yang telah menganalisis kasus-kasus dalam pembuatan Aplikasi Manajemen Buku dengan penerapan Laravel dalam karya berjudul "Cara Efektif Belajar Framework Laravel," dijelaskan bahwa Laravel adalah suatu framework PHP dengan struktur MVC yang diperkenalkan oleh Taylor Otwell pada tahun 2011 dan kini telah memperoleh versi 4.2. Kerangka kerja ini menyajikan berbagai fitur yang amat berharga dalam proses pengembangan, dan hal yang menarik, Laravel mempunyai moto "PHP THAT DOESN'T HURT. CODE HAPPY & ENJOY THE FRESH AIR."

Berbagai fitur unggulan yang dimiliki oleh Laravel:

1. *Eloquent* ORM, Laravel menyediakan sistem ORM yang kuat dan intuitif bernama Eloquent. Dengan Eloquent, kita dapat dengan mudah berinteraksi

dengan basis data menggunakan model objek dan query builder yang ekspresif.

2. *Artisan CLI*, Artisan adalah perintah baris perintah yang kuat yang disertakan dengan Laravel. Artisan memudahkan dalam pembuatan kode, pembuatan model, migrasi basis data, pengelolaan cron jobs, dan banyak lagi.
3. *Routing*, Sistem routing dalam Laravel memungkinkan kita dengan mudah mendefinisikan rute URL dan menghubungkannya dengan tindakan (controller methods) yang sesuai.
4. *Blade Templating Engine*, Blade adalah mesin template yang kuat dan intuitif dalam Laravel. Blade memungkinkan kita membuat tampilan yang konsisten dan dinamis dengan sintaks yang mudah dipahami.
5. *Middleware*, *Middleware* memungkinkan kita menambahkan lapisan fungsionalitas ke dalam permintaan HTTP sebelum mencapai tindakan yang dituju. Ini membantu dalam implementasi autentikasi, otorisasi, dan banyak fungsi lainnya.
6. *Authentication and Authorization*, Laravel menyediakan sistem autentikasi dan otorisasi yang siap pakai, termasuk integrasi dengan berbagai penyedia otentikasi, seperti OAuth.
7. *Paket Bawaan (Batteries Included)*, Laravel dilengkapi dengan berbagai paket dan pustaka siap pakai yang membantu dalam pengembangan aplikasi, seperti pengiriman email, caching, pengiriman notifikasi, dan lain-lain.

8. Pengiriman Email, Laravel menyediakan fitur pengiriman email yang mudah digunakan dan mendukung berbagai layanan email seperti SMTP, Mailgun, dan lain-lain.
9. *Queue and Job Management*, Laravel menyediakan sistem antrian yang memungkinkan kita untuk menjadwalkan tugas-tugas yang memakan waktu, seperti pengiriman email atau pemrosesan data, untuk dieksekusi secara asinkron.
10. Pengujian (*Testing*), Laravel mendukung pengujian otomatis dengan bantuan PHPUnit, sehingga memudahkan kita untuk menguji komponen aplikasi secara efisien.
11. Laracasts, Laracasts adalah platform pembelajaran online yang disediakan oleh pembuat Laravel, Jeffrey Way. Ini merupakan sumber daya yang luar biasa untuk belajar Laravel dan pengembangan web secara umum.
12. Komunitas Aktif, Laravel memiliki komunitas yang besar dan aktif, yang berarti ada banyak sumber daya, tutorial, dan bantuan yang tersedia bagi pengembang.

Menurut Aminudin (Aminudin, 2016), tujuan penggunaan Laravel adalah untuk mempermudah kerja sama dalam tim, karena Laravel menggunakan konsep MVC (Model View Controller) yang memudahkan anggota tim baru untuk bergabung dalam proyek yang sudah berjalan lama tanpa mengalami kesulitan.

Sesuai dengan analisis Rahmat Awaludin (Awaludin, 7 Alasan Menggunakan Framework Laravel dibandingkan native PHP, 2016), seorang Pengembang Web Senior yang telah mendetilkkan peranan tiap aspek kerangka kerja Laravel dalam

bukunya "Menyelami Framework Laravel", menjelaskan bahwa memanfaatkan Laravel berpotensi memberikan sejumlah keunggulan. Dengan memakai beragam abstraksi yang disajikan oleh Laravel, kita dapat lebih mengkonsentrasi logika bisnis aplikasi yang sedang kita bangun. Hal-hal dasar dalam PHP tidak lagi memakan waktu kita. Maksud penggunaan Laravel dalam pengembangan sistem adalah untuk menambahkan lapisan keamanan pada situs web serta mengakselerasi produktivitas seluruh tim. Penetapan penggunaan Laravel didasarkan pada keterkaitan templating yang diberikan oleh kerangka kerja ini. Salah satu alasannya adalah bahwa Laravel beroperasi di bawah naungan suatu perpustakaan. Dalam konteks ini, apabila proyek memerlukan fungsi untuk konversi teks menjadi format Excel, pada pendekatan PHP asli, kita perlu menyusun fungsi skrip spesifik demi tujuan ini agar tidak terjadi duplikasi. Namun, dalam lingkup Laravel, sejumlah perpustakaan yang sesuai dengan tuntutan sistem telah dipersembahkan.

Pada tanggal 14 Februari 2023 Laravel resmi meliris versi terbaru dari *framework*-nya, yaitu 10.x.x. Pada versi terbaru ini dari Laravel memberikan beberapa fitur-fitur baru dan penghapusan beberapa fungsi yang tidak dibutuhkan lagi.

Sebelum Laravel 9, versi Laravel biasanya di *update* 2 kali dalam setahun (6 bulan sekali) . Dan sejak rilis-nya Laravel 9, tim Laravel mulai mengubah kebijakan rilis versi yang semula dua kali dalam setahun menjadi sekali dalam setahun, yaitu tepatnya pada bulan februari.

Berikut adalah beberapa fitur dan perubahan yang ditemukan dalam Laravel

10:

### 1. Dukungan versi PHP di Laravel 10

Untuk memulai menggunakan Laravel versi 10, diperlukan setidaknya PHP versi 8.1.x, sementara untuk versi PHP 8.0.x tidak lagi menjadi persyaratan.

### 2. Tampilan Perintah Artisan Lebih Interaktif

Di Laravel 10, perintah artisan akan memiliki antarmuka yang lebih interaktif. Sebagai contoh, ketika kita menjalankan perintah `make:model` tanpa memberikan parameter "*name*" untuk model tersebut, dan menghasilkan pesan kesalahan, Laravel akan menampilkan opsi berikut:

```
php artisan make:model

What should the model be named?
> Post

Would you like any of the following? [none]
none ..... 0
all ..... 1
factory ..... 2
form requests ..... 3
migration ..... 4
policy ..... 5
resource controller ..... 6
seed ..... 7
```

Gambar 2.6 Opsi Perintah Model

### 3. *Method* `dispatchNow()` dihapus

*Method* `dispatchNow()` yang sebelumnya digunakan untuk mengelola tugas (*jobs*) dalam Laravel telah dihapus dan digantikan oleh metode `dispatchSync()`.

#### 4. *Library doctrine* atau *dbal* tidak dibutuhkan lagi

Sebelumnya, dalam versi Laravel sebelumnya, jika ingin mengubah field atau kolom dalam migrasi, kita harus menginstal library tambahan <https://github.com/doctrine/dbal>. Namun, dalam Laravel 10, fitur ini sudah tersedia secara default tanpa memerlukan instalasi library tambahan tersebut.

#### 5. *Lists Method* yang dihapus

Tim pengembangan Laravel berencana untuk menghapus beberapa metode yang tidak lagi digunakan dalam versi sebelumnya. Jika kita berencana untuk melakukan upgrade dari versi 9 ke 10, ada beberapa hal yang perlu diperhatikan:

- *Remove various deprecations*
- *Remove deprecated dates property*
- *Remove handleDeprecation method*
- *Remove deprecated assertTimesSent method*
- *Remove deprecated ScheduleListCommand'S \$defaultName property*
- *Remove deprecated Route::home method*
- *Remove deprecated dispatchNow functionality*

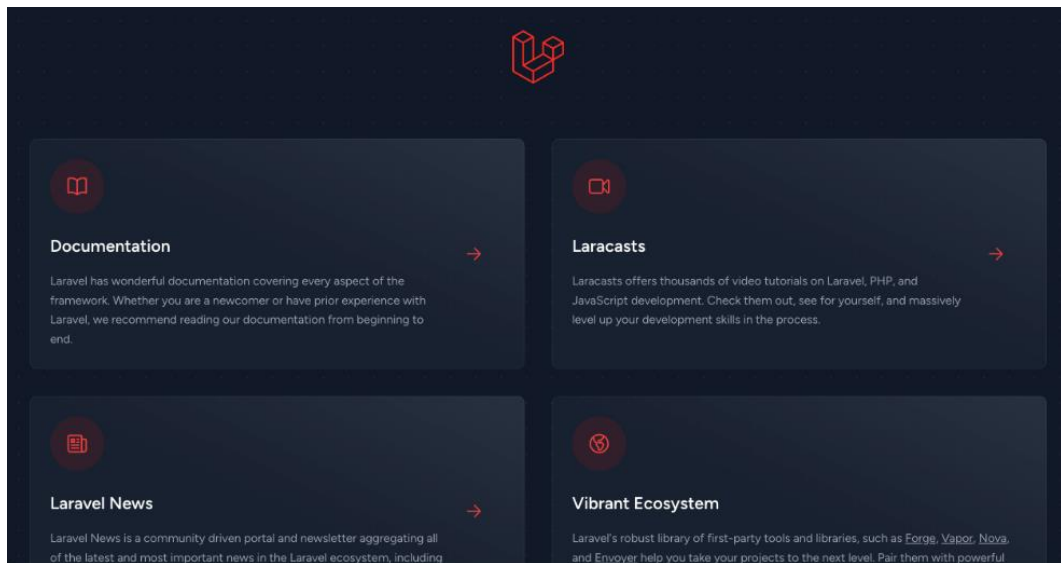
#### 6. Propertis *\$dates* di Model dihapus

Dalam Laravel 10, properti *\$dates* yang terdapat dalam Model akan dihilangkan dan digantikan oleh *\$casts* dalam Eloquent..

#### 7. Tampilan *Welcome Screen* Berubah

Pada Laravel 10, tampilan *welcome screen* setelah berhasil menginstal Laravel telah mengalami perubahan dan sekarang memiliki penampilan sebagai berikut:





Gambar 2. 7 Tampilan *Welcome Screen*

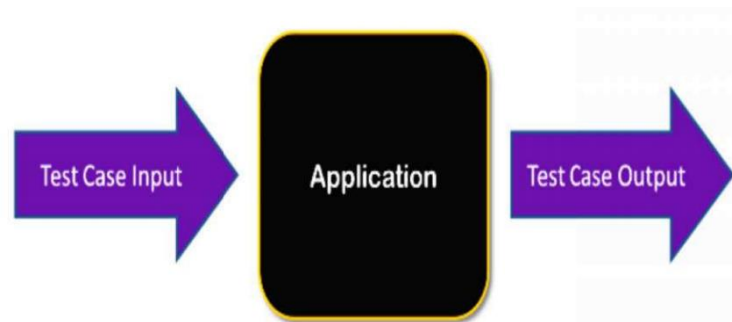
#### 2.14. Black Box Testing

Menurut (Greenit, 2018), Metode Black Box Testing merujuk pada jenis pengujian di mana perangkat lunak dieksekusi dengan menggunakan data uji yang relevan, dan kemudian dilakukan pemeriksaan terhadap fungsionalitasnya. Dalam pengujian black box, para pengembang perangkat lunak dapat memberikan serangkaian kondisi input yang secara komprehensif mengaktifkan semua persyaratan fungsional yang ada dalam program tersebut.

Sedangkan dalam pandangan Williams (2010:36), pengujian black box atau uji fungsional adalah suatu metode pengujian yang tidak memperhatikan bagaimana mekanisme internal sistem atau komponen bekerja, tetapi lebih fokus pada hasil output yang dihasilkan saat menghadapi input tertentu dan kondisi eksekusi tertentu. Dengan demikian, pengujian black box berpusat pada fungsionalitas, yakni bagaimana perilaku perangkat lunak merespons input dari pengguna untuk

menghasilkan output yang sesuai, tanpa memerhatikan detail proses internal atau kode program yang digunakan oleh perangkat lunak..

Melakukan uji coba kinerja dari sebuah program juga sering disebut sebagai pengujian "Behavioral". Dalam kerangka ini, pengujinya memiliki rangkaian nilai input dan output yang diantisipasi. Ketika input diberikan, bila output yang dihasilkan sesuai dengan hasil yang diharapkan, maka program dianggap beroperasi dengan baik. Namun, bila output tidak sejalan, program dianggap menghadapi kendala.



Gambar 2. 8 Gambarkan proses pengujian Black Box Testing  
(Edwar Ali, 2019:166)

Dalam pendekatan pengujian ini, penguji tidak memiliki pengetahuan tentang desain dan struktur kode, dan insinyur pengujian serta pengguna akhir melakukan pengujian pada perangkat lunak.

Metode pengujian black-box:

1. *Equivalence class*, pendekatan ini mengelompokkan input ke dalam kelas-kelas ekuivalen yang memiliki karakteristik serupa. Penguji memilih beberapa nilai perwakilan dari setiap kelas untuk pengujian.

2. *Boundary values*, teknik ini berfokus pada pengujian nilai-nilai batas atau tepi dari kelas ekivalen. Tujuannya adalah untuk mengidentifikasi masalah yang mungkin muncul di sekitar nilai-nilai batas.
3. *Cause-effect graphing*, pendekatan ini melibatkan pembuatan grafik berdasarkan hubungan antara berbagai faktor penyebab (input) dan efek (output) dari perangkat lunak. Grafik ini membantu mengidentifikasi kombinasi input yang berpotensi menyebabkan masalah.
4. *Pair-wise Testing*, metode ini melibatkan pengujian semua kemungkinan kombinasi pasangan input yang mungkin. Hal ini membantu mengurangi jumlah uji yang perlu dilakukan, sambil mencoba menggabungkan kombinasi yang paling relevan.
5. *State-based testing*, teknik ini fokus pada pengujian perangkat lunak dalam berbagai keadaan (state) yang mungkin terjadi selama eksekusi. Ini khususnya penting dalam pengujian perangkat lunak yang berinteraksi dengan berbagai kondisi atau situasi yang dapat berubah.

### **2.15. Pasien**

Pasien merujuk kepada individu dengan kelemahan fisik atau mental yang mempercayakan pengawasan dan perawatan mereka kepada tenaga kesehatan serta mengikuti prosedur pengobatan yang telah ditentukan. Definisi ini disampaikan oleh Prabowo (sebagaimana dikutip dalam Wilhamda, 2011). Dalam pandangan (Aditama, 2002), pasien mengacu pada individu yang sedang menjalani perawatan

di rumah sakit. Sementara menurut (Soejadi, 1996), pasien dianggap sebagai entitas yang memiliki peran sentral dalam konteks lingkungan rumah sakit.

Berdasarkan pendapat dari para ahli di atas, dapat disimpulkan bahwa pasien adalah individu yang memiliki kelemahan fisik atau mental dan membutuhkan pengawasan, perawatan, dan pengobatan yang diberikan oleh tenaga kesehatan. Pasien juga dapat merujuk kepada individu yang sedang menjalani perawatan di rumah sakit, dan mereka memiliki peran sentral dalam lingkungan rumah sakit. Jadi, secara umum, pasien adalah seseorang yang memerlukan perhatian medis dan perawatan dalam konteks kesehatan.

#### **2.16. Apotek**

Apotek merupakan fasilitas layanan kefarmasian di mana apoteker melakukan praktik kefarmasian. Definisi ini sesuai dengan Peraturan Menteri Kesehatan Republik Indonesia (Permenkes RI) tahun 2016. Apoteker, menurut Departemen Kesehatan Republik Indonesia (Depkes RI) tahun 2009, adalah sarjana farmasi yang berwenang secara hukum untuk melakukan praktik kefarmasian setelah melewati ujian sumpah jabatan.

Standar Pelayanan Kefarmasian di Apotek, yang diatur oleh Permenkes RI No. 73 tahun 2016, menetapkan bahwa apotek merupakan tempat di mana apoteker melakukan praktik kefarmasian. Pelayanan kefarmasian adalah pemberian layanan langsung dan bertanggung jawab kepada pasien terkait sediaan farmasi guna mencapai hasil yang diinginkan untuk meningkatkan kualitas hidup pasien.

Ruang lingkup pelayanan kefarmasian di apotek, menurut peraturan tersebut, melibatkan dua kegiatan utama. Pertama, kegiatan manajerial termasuk pengelolaan sediaan farmasi, alat kesehatan, dan bahan medis habis pakai, serta pelayanan farmasi klinik. Kedua, kegiatan tersebut harus didukung oleh sarana, prasarana, dan sumber daya manusia yang mendukung.

Apotek memiliki dua fungsi utama, yaitu sebagai sarana kesehatan yang bertujuan sosial dan sebagai entitas bisnis yang berorientasi pada keuntungan. Sebagai sarana kesehatan, apotek harus memberikan pelayanan profesional sesuai peraturan dan etika profesi apoteker. Sementara itu, sebagai sarana bisnis, apotek harus mampu menghasilkan keuntungan dan dikelola dengan baik oleh apoteker sebagai manajer.

Dalam hal perundang-undangan, apotek harus dikelola oleh apoteker yang memiliki kemampuan mengelola sumber daya, berkomunikasi, mengambil keputusan, serta memiliki pengetahuan yang mendalam tentang kefarmasian. Dengan manajemen yang profesional, pelayanan kefarmasian di apotek diharapkan dapat memuaskan pasien dan berkontribusi pada peningkatan kualitas hidup mereka.

Permenkes RI No. 73 tahun 2016 mengatur tata cara pelayanan di apotek, ini mencakup pengaturan tempat, fasilitas, dan tenaga kerja yang diperlukan. Apotek harus mudah diakses oleh masyarakat, terpisah dari produk lain, menjaga kebersihan, dan memiliki fasilitas yang memadai untuk penyimpanan produk farmasi yang memerlukan suhu dingin.

Peraturan Menteri Kesehatan Republik Indonesia (Permenkes RI) No. 73 Tahun 2016 menetapkan bahwa apotek harus memiliki fasilitas dan infrastruktur yang terdiri dari:

1. Ruang untuk menerima resep.
2. Ruang untuk melayani resep dan melakukan peracikan (produksi sediaan secara terbatas).
3. Ruang untuk penyerahan obat kepada pasien.
4. Ruang konseling untuk memberikan penjelasan dan saran kepada pasien.
5. Ruang penyimpanan untuk sediaan farmasi, alat kesehatan, dan bahan medis habis pakai.
6. Ruang untuk penyimpanan arsip.

## **2.17. Praktik Dokter**

### **a. Pengertian Praktik Dokter**

Berdasarkan Pasal 1 ayat (1) UUPK, “Praktik kedokteran adalah rangkaian kegiatan yang dilakukan oleh dokter terhadap pasien dalam melaksanakan upaya kesehatan”. Tempat praktik dokter disebut sebagai fasilitas pelayanan kesehatan.

### **b. Pelaksanaan Praktik Kedokteran**

Berdasarkan Pasal 1 ayat (1) Permenkes No. 2052/MenKes/Per/X/2011 tentang Izin Praktik dan Pelaksanaan Praktik Kedokteran, “Praktik kedokteran adalah rangkaian kegiatan yang dilakukan oleh dokter terhadap pasien dalam melaksanakan upaya

kesehatan”. Walaupun seorang dokter sudah memiliki Surat Tanda Registrasi (STR) yang menetapkan statusnya sebagai dokter, dokter gigi, dokter spesialis, atau dokter gigi spesialis yang sah, tetapi sebelum dapat melaksanakan praktek medis, dokter tersebut perlu memperoleh Surat Izin Praktik (SIP). Kewajiban untuk mendapatkan SIP ini diatur secara tegas dalam Peraturan Menteri Kesehatan Nomor 2052/MenKes/Per/X/2011 tentang Izin Praktik dan Pelaksanaan Praktik Kedokteran.

Sebelum memulai pelaksanaan praktik kedokteran, seorang dokter diwajibkan untuk menginstal papan nama praktik kedokteran sesuai dengan ketentuan yang dijelaskan dalam Pasal 26 dari Peraturan Menteri Kesehatan Nomor 2052/MenKes/Per/X/2011 tentang Izin Praktik dan Pelaksanaan Praktik Kedokteran. Papan nama tersebut harus mencakup informasi berupa nama dokter, nomor Surat Tanda Registrasi (STR), serta nomor Surat Izin Praktik (SIP).

c. Hubungan Hukum Dokter-Pasien dalam Praktik Kedokteran

1) Aspek hukum hubungan dokter-pasien

Dalam konteks hukum medis, terdapat relasi antara dokter dan pasien sebagai dua subjek hukum. Keduanya membentuk ikatan medis dan memiliki dimensi hukum yang terkait. Saat menjalankan ikatan ini, interaksi antara dokter dan pasien selalu diatur oleh ketetapan-ketetapan tertentu untuk menjaga keseimbangan dan konsistensi dalam pelaksanaannya. Relevansi pengaturan ini tampak karena tanpa panduan yang eksplisit, ikatan tersebut berpotensi mengalami

disproporsi dan ambiguitas (kondisi di mana suatu informasi, kalimat, atau situasi memiliki dua atau lebih interpretasi yang mungkin).

Secara evolusi, hubungan hukum antara dokter dan pasien dapat dikelompokkan menjadi dua jenis. Dalam perkembangannya, relasi hukum antara dokter dan pasien memiliki dua jenis, yakni:

a) Hubungan karena kontrak (perjanjian terapeutik)

Relasi hukum antara dokter dan pasien yang muncul akibat suatu peristiwa hukum, seperti dalam bentuk kesepakatan pengobatan atau kontrak terapi.

b) Hubungan karena undang-undang (Zaakwarneming)

Jika pasien tak sadarkan diri dan dokter tak bisa memberikan informasi, maka hak dokter untuk menjalankan langkah medis tanpa persetujuan pasien dapat terjadi, berdasarkan keputusan sukarela atau mengacu pada Pasal 1354 KUH Perdata yang dikenal sebagai *zaakwarneming*.

2) Kesetaraan hubungan dokter-pasien

Gunanya untuk mencapai keseimbangan dalam membangun hubungan dokter-pasien, adalah dengan memastikan terjaga komunikasi yang seimbang di antara keduanya. Ini mengindikasikan bahwa baik dokter maupun pasien memiliki hak yang setara dalam menyuarakan maksud dan harapan mereka. Kesetaraan ini memiliki pengaruh penting dalam proses pertukaran informasi antara dokter dan pasien.



#### d. Pola Komunikasi Dokter-Pasien dalam Praktik Kedokteran

Pada praktik kedokteran, terdapat beberapa tipe dokter yang memiliki dampak pada komunikasinya dengan pasien dalam pemberian pelayanan kesehatan, di antaranya:

- 1) Dokter yang enggan memberikan jawaban walaupun pasien mengajukan pertanyaan.
- 2) Dokter yang bersedia memberikan jawaban jika ditanya, tetapi hanya memberikan jawaban yang sesuai dengan konsultasi pasien.
- 3) Dokter yang mau memberikan tanggapan terhadap pertanyaan pasien, mengajukan pertanyaan secara proaktif, dan juga memberikan informasi ekstra yang relevan untuk kepentingan kesehatan pasien.

Dalam praktik kedokteran, terdapat tiga pola komunikasi yang terjadi antara dokter dan pasien, yakni:

##### 1) Aktif-Pasif

Dalam pola komunikasi dominan-pendamping ini, dokter mengambil inisiatif lebih banyak sementara pasien lebih cenderung untuk mengikuti arahan. Pasien hanya berinteraksi ketika dia diajak atau melakukan tindakan setelah mendapatkan petunjuk dari dokter, termasuk dalam hal konsumsi makanan atau obat sesuai resep dokter.

##### 2) Guidance-Cooperation

Pengembangan dari model komunikasi awal melibatkan mengarahkan pasien untuk mendorong kerja sama. Pasien dianggap tidak perlu ahli di bidang medis, tetapi perlu dibimbing dan diajak untuk berkolaborasi

dalam proses penyembuhan kondisi kesehatannya. Dokter berfungsi sebagai fasilitator, seperti hubungan orang tua dan anak.

### 3) Mutual Participation

Landasan dari model ini berakar pada keyakinan bahwa tiap individu mempunyai hak dan martabat yang setara. Pasien secara sukarela memainkan peran yang aktif dalam proses perawatan mereka. Meski begitu, pendekatan ini mungkin tak cocok bagi pasien yang memiliki pendidikan atau latar belakang sosial terbatas, juga untuk anak-anak atau pasien dengan kondisi mental khusus.

#### e. Standar Kompetensi Dokter Indonesia

Berkenaan dengan definisi yang dicatat dalam Pasal 1 dari Keputusan Konsil No. 11 tahun 2012 tentang Standar Kompetensi Dokter Indonesia, disebutkan bahwa, "Standar Kompetensi Dokter Indonesia (SKDI) merupakan bagian integral dari standar pendidikan profesi dokter Indonesia yang diotorisasi oleh Konsil Kedokteran Indonesia". Dengan mempertimbangkan pengalaman lembaga pendidikan medis dalam implementasi SKDI. Dalam Keputusan Konsil No. 11 Tahun 2012 tentang Standar Kompetensi Dokter Indonesia, diuraikan berbagai kualifikasi, yang meliputi:

- Kualitas profesional yang mulia.
- Kesadaran diri dan peningkatan diri.
- Komunikasi yang berhasil.
- Manajemen informasi.

- Dasar ilmiah dalam bidang Kedokteran.
- Kemahiran klinis.
- Pengelolaan permasalahan kesehatan.

Dalam manajemen kondisi kesehatan, gangguan kesehatan dikelompokkan sesuai sistem organ manusia, diikuti oleh tingkat pencapaian yang harus dicapai saat pendidikan medis.

Tingkat Prestasi 1 meliputi kemampuan untuk mengidentifikasi dan menjelaskan. Lulusan dokter memiliki kemampuan untuk mengenali dan menjelaskan gambaran klinis suatu penyakit. Mereka juga memahami metode terbaik untuk memperoleh informasi tambahan tentang penyakit tersebut dan dapat menetapkan rujukan yang tepat untuk pasien. Selain itu, lulusan dokter memiliki keterampilan untuk memantau kondisi pasien setelah kembali dari rujukan.

Tingkat Prestasi 2 melibatkan kemampuan mendiagnosis dan merujuk. Lulusan dokter memiliki kemampuan untuk mendiagnosis penyakit secara klinis dan menentukan rujukan yang paling sesuai untuk perawatan lebih lanjut pasien. Mereka juga dapat melakukan tindak lanjut setelah pasien kembali dari rujukan.

Tingkat Prestasi 3 mencakup kemampuan mendiagnosis, memberikan penanganan awal, dan merujuk.

Tingkat Prestasi 4 melibatkan kemampuan mendiagnosis, memberikan penanganan komprehensif secara mandiri. Lulusan dokter

memiliki kemampuan untuk mendiagnosis penyakit secara klinis dan memberikan perawatan yang komprehensif secara mandiri.

## **BAB III**

### **ANALISA DAN DESAIN SISTEM**

Dalam bab ini, akan menguraikan secara menyeluruh berbagai komponen, proses, dan aspek yang terlibat dalam sistem antrian manual di Apotek Bio Medika dan Praktek Dokter. Saya akan mengeksplorasi bagaimana sistem ini memengaruhi efisiensi pelayanan, pengalaman pasien, serta kendala yang mungkin timbul dalam pengaturan antrian manual. Dengan pemahaman yang mendalam tentang sistem ini, akan merumuskan dasar untuk mendesain solusi yang lebih optimal dan memberikan rekomendasi untuk perbaikan yang dapat meningkatkan kinerja serta kepuasan semua pihak yang terlibat.

#### **3.1. Analisa Sistem**

Analisis sistem merujuk pada proses mendalam untuk memahami, mengidentifikasi, menganalisis, dan mengevaluasi suatu sistem, baik itu sistem teknologi informasi, bisnis, organisasi, atau proses lainnya. Tujuan utama dari analisis sistem adalah untuk mengidentifikasi masalah, kebutuhan, dan peluang perbaikan dalam sistem yang ada, sehingga dapat dihasilkan pemahaman yang lebih baik tentang bagaimana sistem beroperasi dan bagaimana ia dapat ditingkatkan.

### **3.1.1. Sistem Antrian Manual**

Sistem antrian manual yang berlangsung di Apotek Bio Medika dan Praktek Dokter merupakan suatu metode pengaturan aliran pasien yang dilakukan secara manual oleh petugas atau staf Apotek Bio Medika dan Praktek Dokter tanpa menggunakan teknologi otomatis.

Berikut adalah penjelasan dan analisa sistem antrian manual di Apotek Bio Medika:

1. Pendaftaran Pasien

Ketika pasien datang ke Apotek Bio Medika, mereka mendaftarkan diri di meja pendaftaran. Petugas pendaftaran akan mencatat informasi pasien dan memberikan nomor antrian.

2. Nomor Antrian

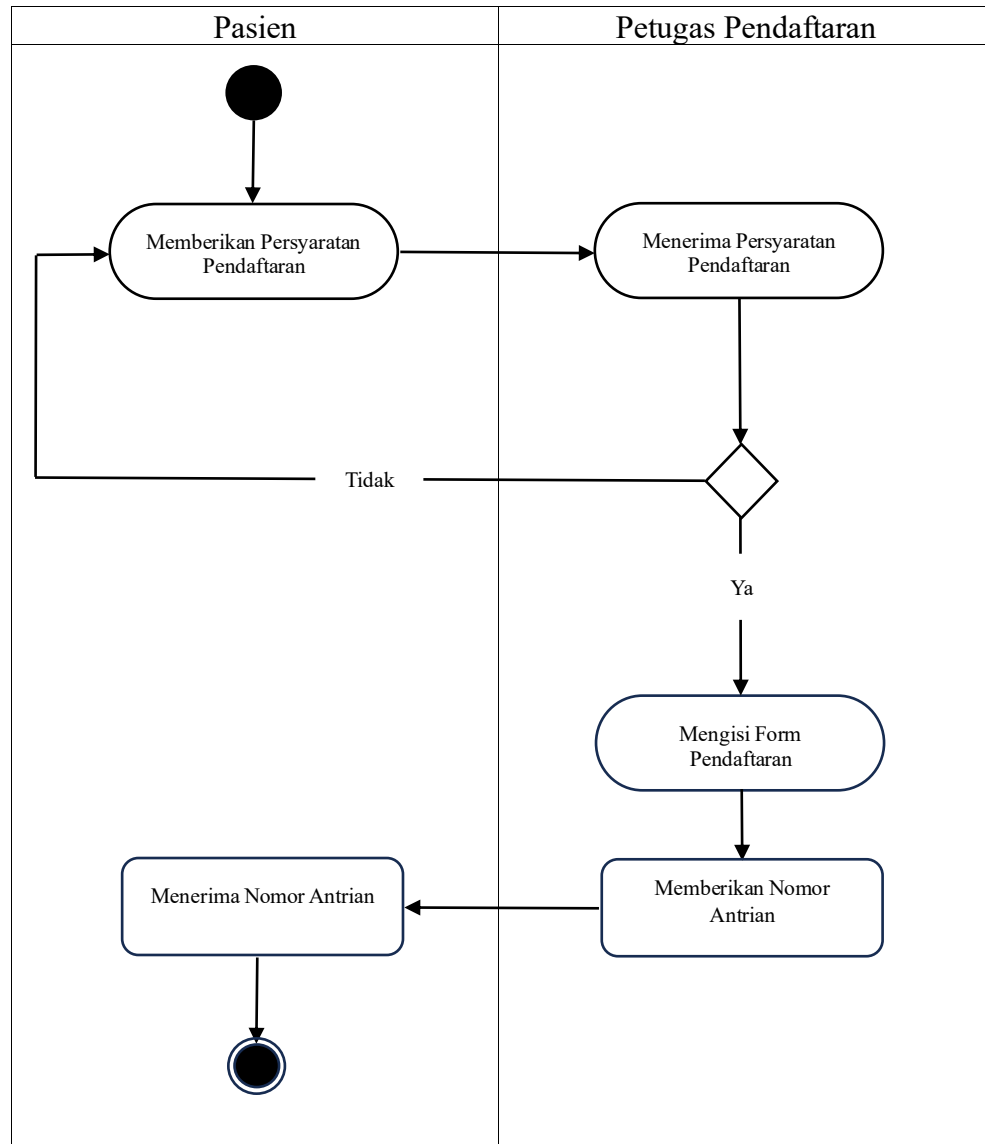
Pasien akan diberikan nomor antrian, yang menunjukkan urutan pelayanan. Pasien kemudian menunggu di area tunggu sampai nomor antrian mereka dipanggil.

3. Panggilan Antrian

Petugas panggilan akan memanggil nomor antrian yang sedang dilayani secara manual, baik melalui pengeras suara atau secara lisan. Pasien yang nomor antriannya dipanggil akan menuju ruang pemeriksaan.

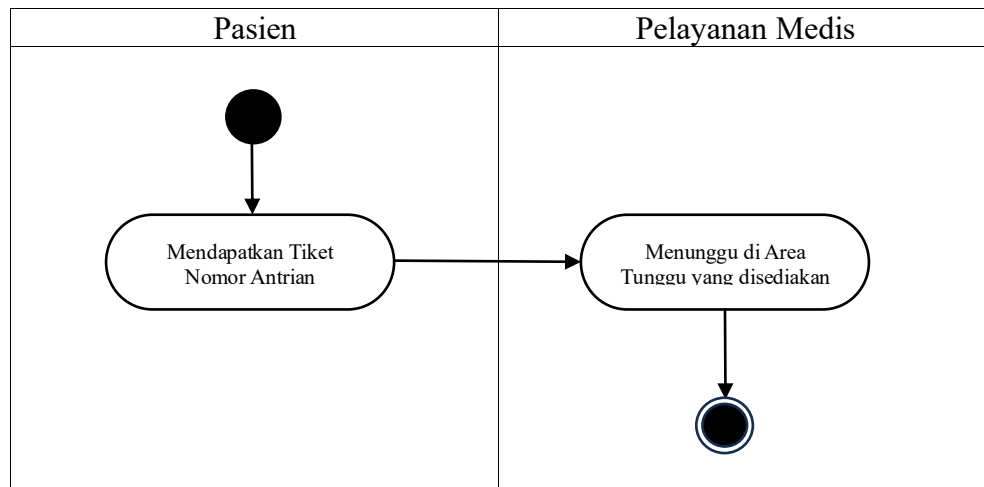
### 3.1.2. Activity Diagram

#### 1. Activity Diagram Pendaftaran Pasien



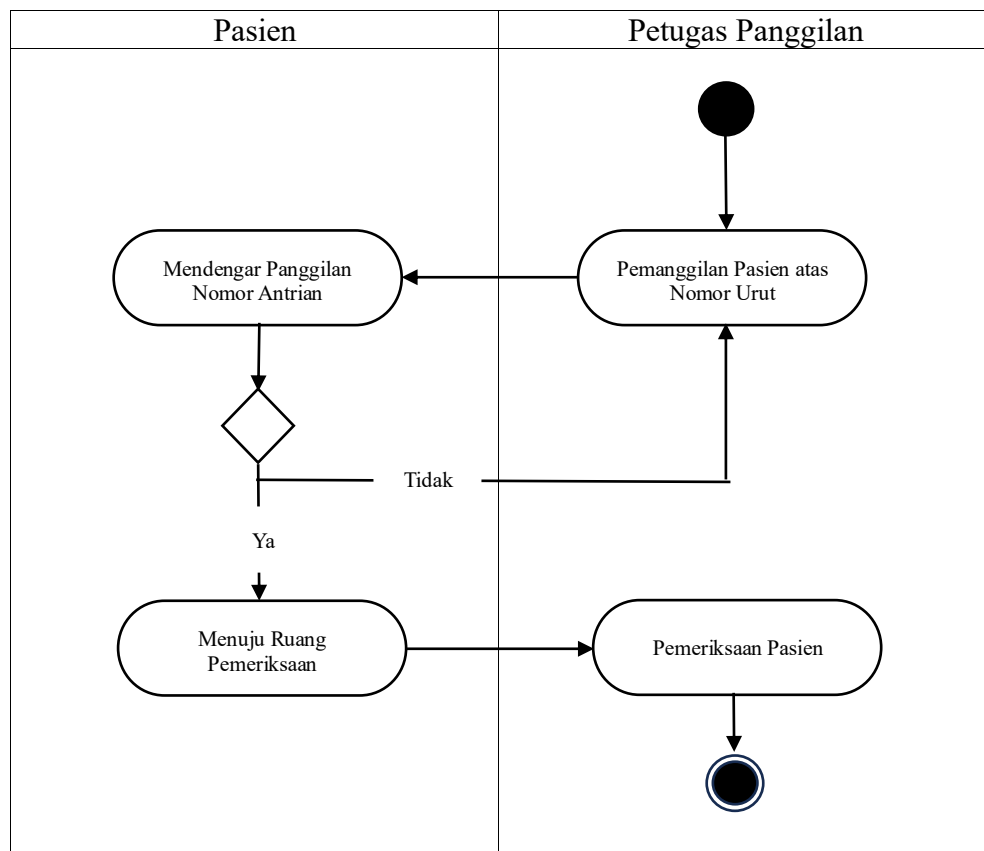
Gambar 3.1 Pendaftaran Pasien

2. *Activity Diagram* Nomor Antrian



Gambar 3.2 Nomor Antrian

3. *Activity Diagram* Panggilan Antrian



Gambar 3.3 Panggilan Antrian



### 3.1.3. Evaluasi Analisa Sistem Antrian Manual Yang Sedang Berjalan

Berdasarkan evaluasi terhadap sistem antrian manual yang sedang diterapkan saat ini di Apotek Bio Medika dan Praktek Dokter, masih terdapat beberapa kekurangan yang dapat diidentifikasi sebagai berikut:

**Tabel 3.1 Evaluasi Sistem**

Masalah	Aktor	Solusi
Waktu tunggu pasien yang lama menjadi masalah utama	Pasien	<p>Pengembangan Sistem Antrian Komputer</p> <ul style="list-style-type: none"> <li>• Deskripsi Solusi: Membangun sistem antrian berbasis komputer yang dapat mengelola dan memantau antrian pasien secara otomatis.</li> <li>• Manfaat: Sistem ini akan memberikan estimasi waktu tunggu yang lebih akurat kepada pasien, mengurangi waktu tunggu, dan memungkinkan pengaturan prioritas berdasarkan kondisi kesehatan pasien.</li> <li>• Tahap Implementasi: Desain sistem, pengembangan perangkat lunak, pengujian, pelatihan petugas, dan peluncuran sistem.</li> </ul>
Koordinasi yang buruk dalam pemanggilan nomor antrian	Petugas panggilan.	<p>Sistem Pemanggilan Antrian Otomatis</p> <ul style="list-style-type: none"> <li>• Deskripsi Solusi: Menerapkan sistem pemanggilan antrian otomatis yang akan memanggil nomor antrian dengan akurat dan secara otomatis.</li> <li>• Manfaat: Mengurangi kesalahan dalam pemanggilan nomor antrian, mempercepat aliran pasien, dan meningkatkan koordinasi antara petugas.</li> <li>• Tahap Implementasi: Pemilihan sistem, pelatihan petugas, integrasi dengan sistem antrian, dan pengujian.</li> </ul>

## **3.2. Desain Sistem**

### **3.2.1. Tujuan Desain Sistem**

Desain sistem meliputi berbagai aspek yang akan meningkatkan efisiensi dan pengalaman pasien dalam konteks sistem antrian di Apotek Bio Medika dan Praktek Dokter.

Pada dasarnya desain sistem ini bertujuan untuk :

1. Meningkatkan efisiensi proses antrian dengan menerapkan metode FIFO (First-In-First-Out). Ini membantu memastikan bahwa pasien yang pertama datang adalah yang pertama dilayani, yang dapat mengurangi waktu tunggu secara keseluruhan.
2. Memberikan pengalaman pasien yang lebih baik dengan memberikan estimasi waktu tunggu yang akurat dan mengurangi ketidakpastian. Tujuannya adalah menciptakan kepuasan pasien yang lebih tinggi.
3. Mengoptimalkan penggunaan sumber daya Apotek Bio Medika dan Praktek Dokter, seperti tenaga medis dan fasilitas, dengan mengatur antrian dengan efisien.
4. Memberikan transparansi kepada pasien tentang di mana mereka berada dalam antrian dan perkiraan waktu tunggu mereka, sehingga pasien dapat merasa lebih terinformasi.
5. Memfasilitasi akses lebih cepat ke informasi medis pasien untuk tenaga medis, memungkinkan pelayanan medis yang lebih efektif dan tepat waktu.

6. Menggunakan algoritma FIFO dan sistem komputer, sistem ini membantu mengurangi kesalahan manusia dalam pengaturan antrian dan pemberian nomor antrian kepada pasien.
7. Memanfaatkan teknologi komputer untuk meningkatkan pengaturan antrian dan manajemen pendaftaran pasien.

### 3.2.2. Gambaran Sistem Yang Diusulkan

Dalam desain sistem pemodelan tiket antrian di Apotek Bio Medika dan Dokter Praktek, kami menggunakan UML (*Unified Modeling Language*) diantaranya *use case diagram*, *activity diagram*, *sequence diagram* dan *flowchart* algoritma FIFO, *class diagram*. Inisiatif sistem mencakup langkah-langkah berikut: pasien mengambil nomor antrian, menunggu di area pendaftaran, nomor antrian dipanggil oleh petugas pendaftaran, menuju ke loket pendaftaran, melakukan registrasi, mendapatkan nomor antrian untuk pemeriksaan, menunggu di area pemeriksaan, nomor antrian dipanggil oleh petugas pemeriksaan, dan menuju ruang pemeriksaan.

#### 1. Analisa Actor

*Actor* yang berperan dalam Desain dan Pembuatan Sistem antrian ini adalah:

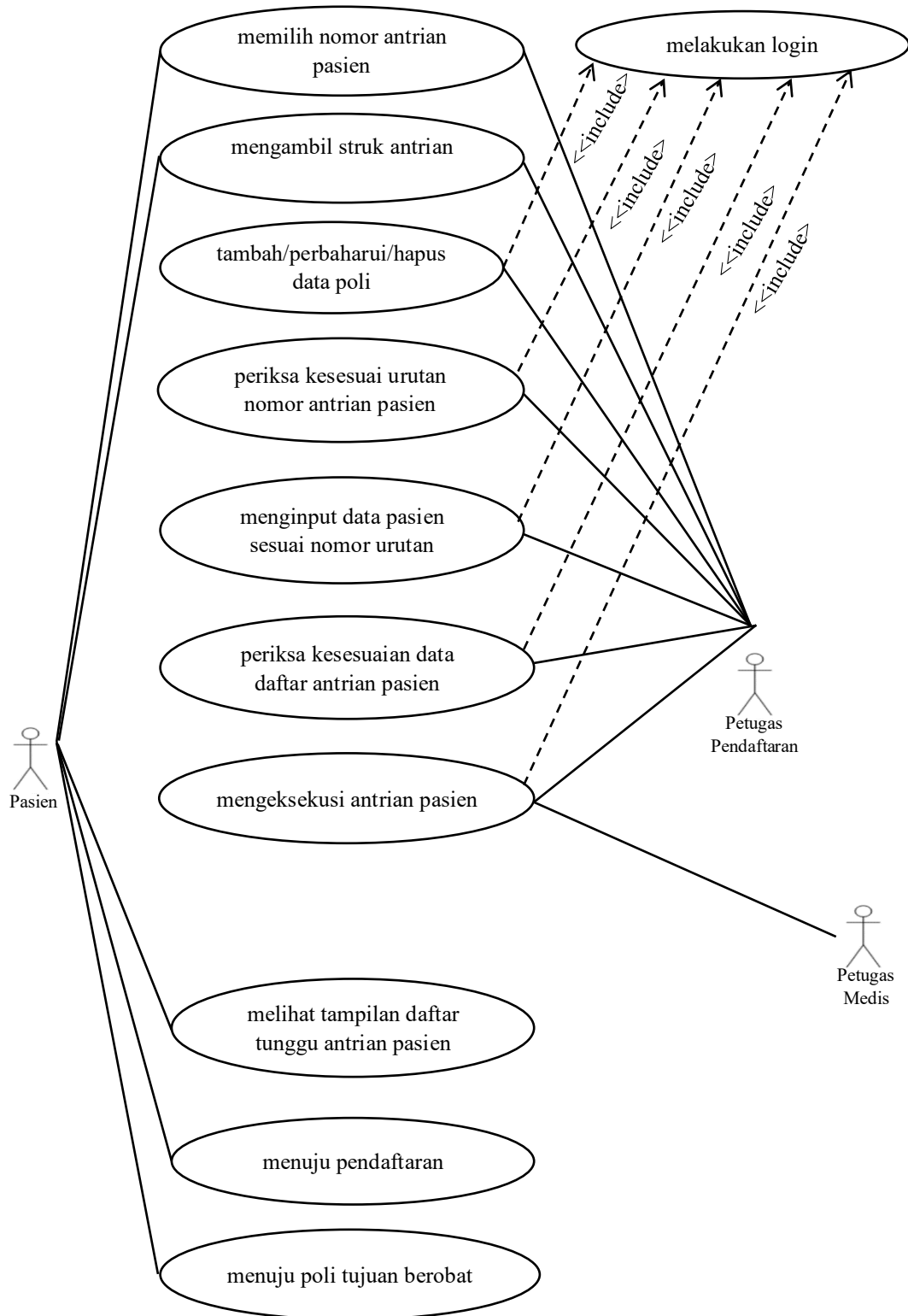
**Tabel 3.2 Peran Actor dalam Use Case**

<i>Actor</i> (pengguna)	Peran
Pasien	Mengambil nomor antrian
Petugas Pendaftaran	Prosedur meliputi penyediaan formulir untuk pasien baru, melakukan wawancara dengan pasien, memasukkan data pasien ke dalam sistem komputer, mencetak Kartu Berobat, memberikan tiket antrian, membimbing pasien ke poliklinik yang sesuai, dan

	memberikan informasi mengenai pelayanan medis di Apotek Bio Medika dan Praktek Dokter.
Petugas Medis	memanggil nomor antrian pasien berikutnya, memberikan data dan informasi kepada manajemen tentang penggunaan sistem antrian, waktu tunggu pasien, dan statistik lainnya yang dapat digunakan untuk analisis dan perencanaan.

## **2. Use Case Diagram**

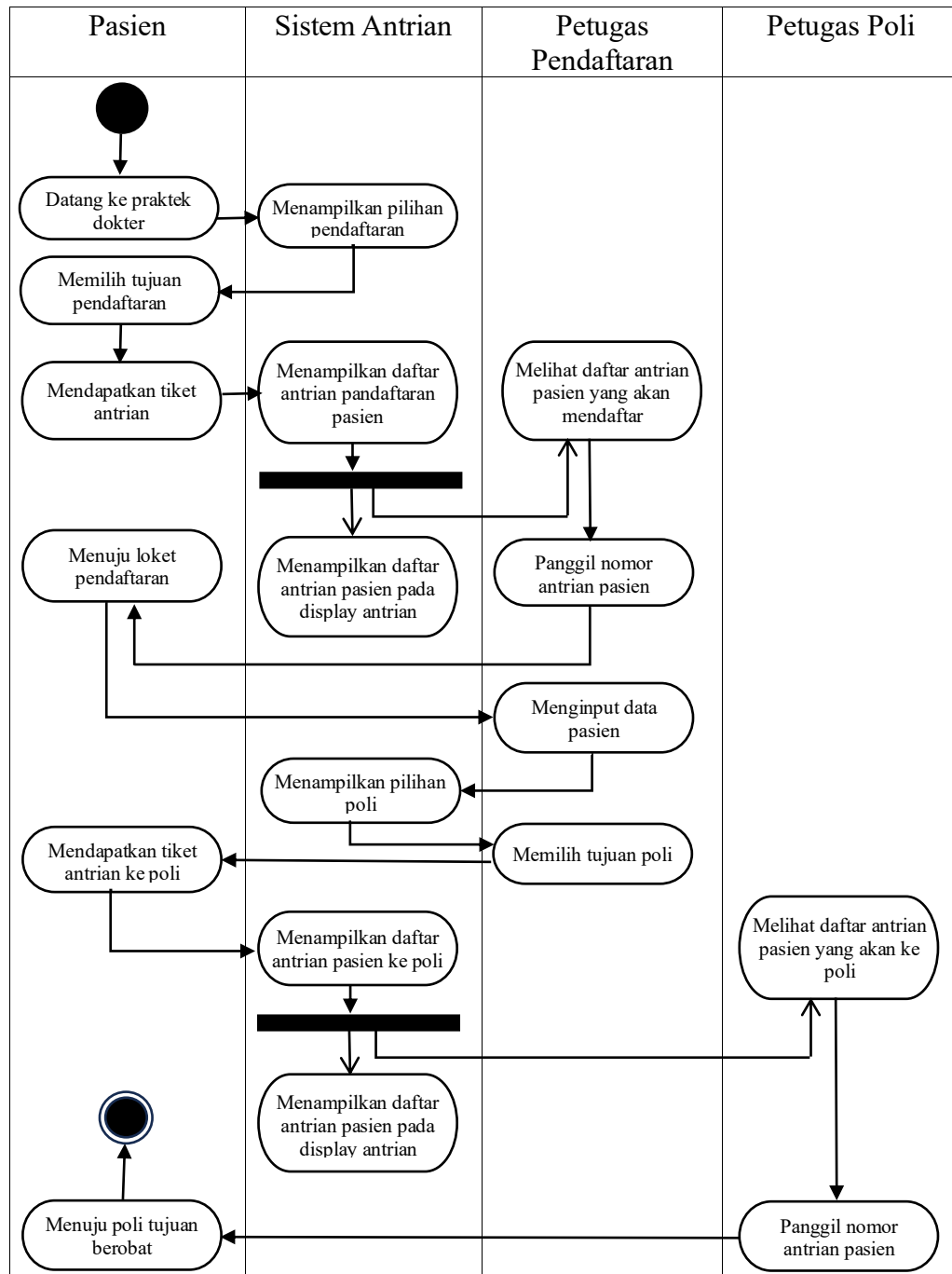
*Use Case Diagram*, representasi visual yang digunakan dalam analisis dan desain sistem untuk mengilustrasikan hubungan antara pengguna atau aktor dengan sistem. Diagram ini memberikan gambaran umum tentang bagaimana sistem berfungsi dan bagaimana interaksi terjadi antara pengguna atau aktor dengan sistem. Pada Gambar 3.6, akan disajikan Diagram Kasus Penggunaan dari Sistem.



Gambar 3.4 *Use Case* yang diusulkan

### 3. Activity Diagram

Merupakan *Activity Diagram* yang mengilustrasikan repetisi kegiatan dari pasien, petugas pendaftaran dan petugas poli dalam sistem antrian, yang berlangsung dari awal hingga akhir pada Gambar 3.7.

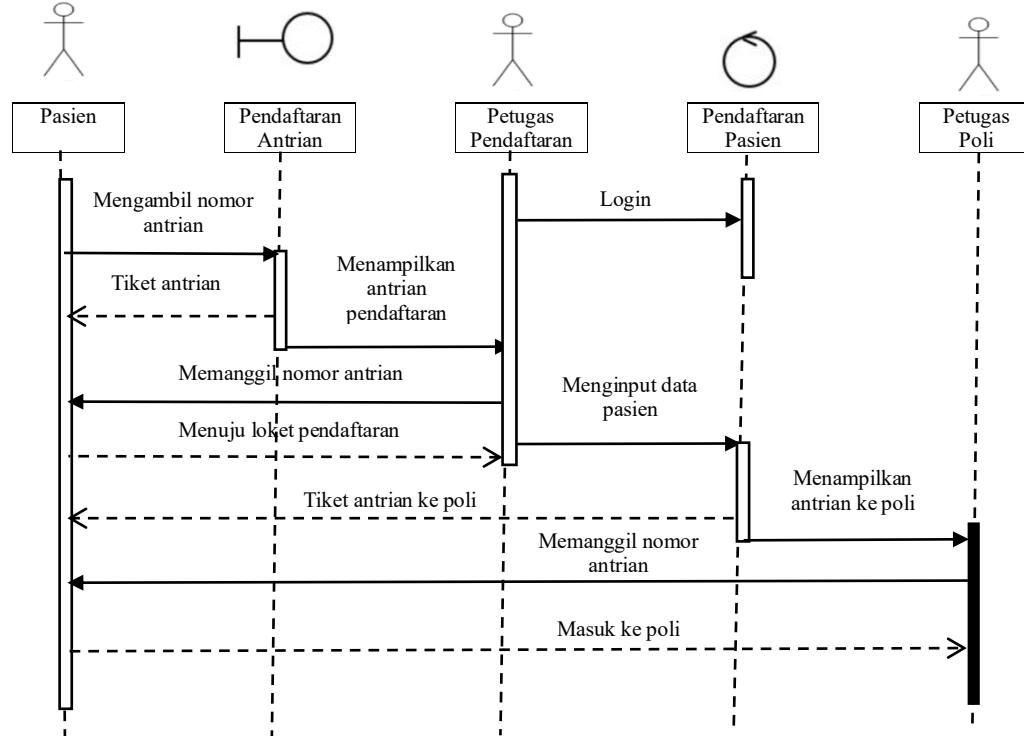


Gambar 3.5 Activity Diagram

#### 4. Sequence Diagram

*Sequence Diagram* berguna untuk mengilustrasikan bagaimana berbagai objek atau entitas berinteraksi satu sama lain dalam proses antrian pasien pada Gambar

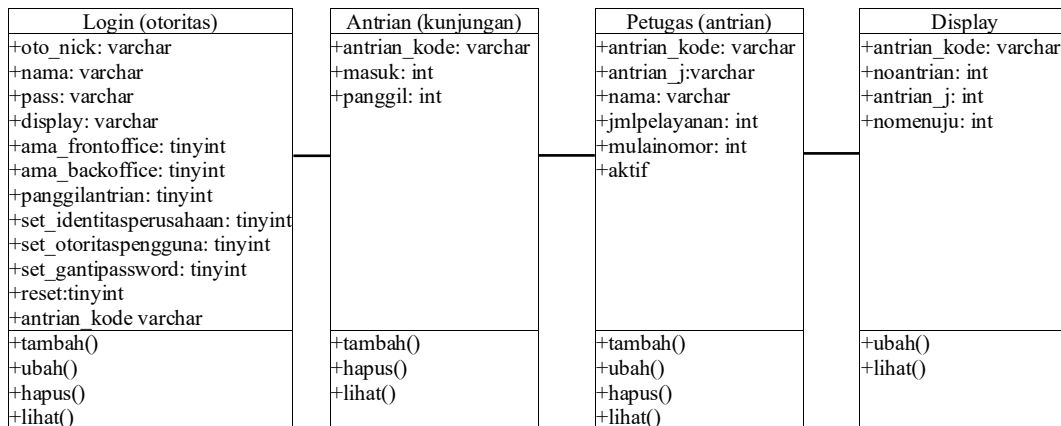
3.8.



Gambar 3.6 *Sequence Diagram*

#### 5. Class Diagram

*Class Diagram* adalah representasi gambaran dari struktur program sistem dalam berbagai bentuk. *Class Diagram* menggambarkan bagaimana alur basis data berfungsi dalam sebuah sistem, Gambar 3.9 memperlihatkan berbagai tabel yang terdapat dalam sistem, di mana setiap tabel memiliki beberapa query untuk menampung masukan dari sistem dan fungsi yang menjalankan perintah.

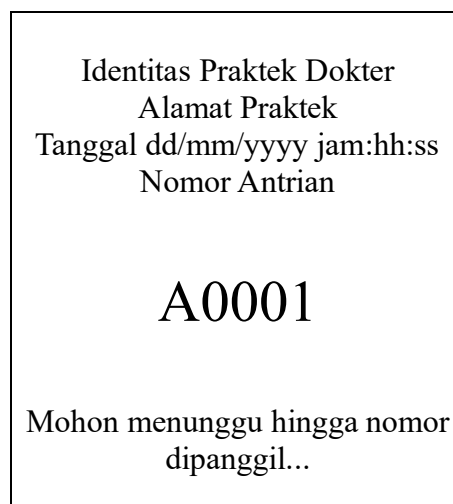


Gambar 3.7 Class Diagram

### 3.2.3. Desain Output

Desain output adalah proses merancang tampilan dan format output yang dihasilkan oleh sebuah sistem komputer atau aplikasi. Output adalah informasi atau hasil yang diberikan oleh sistem kepada pengguna atau sistem lain. Desain output sangat penting karena dapat mempengaruhi cara pengguna memahami dan menggunakan informasi yang disajikan. Berikut adalah rancangan output yang telah penulis buat:

1. Desain tiket antrian



Gambar 3.8 Tiket antrian



## 2. Desain display utama

Logo	Jenis Usaha Identitas usaha Alamat Praktek		
Tujuan 99 <b>A0001</b>		Tampilan video	
Tujuan 01 A0001	Tujuan 02 B0001	Tujuan 03 C0001	Tujuan 04 D0001
Selamat datang			

Gambar 3.9 Display utama

### 3.2.4. Desain Input

Setiap tahap memerlukan data masukan yang umumnya dimasukkan melalui antarmuka yang menghubungkan pengguna dengan perangkat keras dan perangkat lunak. Oleh karena itu, untuk mempermudah dan mencegah kesalahan dalam memasukkan data, maka di desain tampilan menu yang *user-friendly* untuk menginput data.

## 1. Desain form data menu login

Masukkan Login	
Username	<input style="width: 100%;" type="text"/>
Password	<input style="width: 100%;" type="password"/>
<input style="width: 50%; height: 20px;" type="button" value="Masuk"/>	<input style="width: 50%; height: 20px;" type="button" value="Batal"/>

Gambar 3.10 *Input form login*

## 2. Desain form panggilan antrian

No	Tujuan	Jml. Antrian	No. Antrian	Sisa	No Layanan & panggil		No Layanan& panggil ulang	
1	Pendaftaran	0	0	0	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil"/>	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil ulang"/>
2	Poli A	0	0	0	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil"/>	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil ulang"/>
3	Poli B	0	0	0	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil"/>	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil ulang"/>
...	...	...	...	...	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil"/>	<input style="width: 30px;" type="text" value="0"/>	<input style="width: 50px;" type="button" value="Panggil ulang"/>
...	...	...	...	...				
...	...	...	...	...				
n	Poli n	n	n	n	<input style="width: 30px;" type="text" value="n"/>	<input style="width: 50px;" type="button" value="Panggil"/>	<input style="width: 30px;" type="text" value="n"/>	<input style="width: 50px;" type="button" value="Panggil ulang"/>

Gambar 3.11 *Input form panggilan antrian*3. Desain form ambil nomor antrian *front office*

Logo	Jenis Usaha Identitas usaha Alamat Praktek
SELAMAT DATANG DI APOTEK BIO MEDIKA DAN PRAKTEK DOKTER KLIK / PILIH NOMOR ANTRIAN  <div style="border: 1px solid black; width: 200px; height: 60px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">             A. Pendaftaran           </div>	
Selamat datang	

Gambar 3.12 *Input form ambil antrian front office*

#### 4. Desain form ambil nomor antrian *back office*

Logo	Jenis Usaha Identitas usaha Alamat Praktek
SELAMAT DATANG DI APOTEK BIO MEDIKA DAN PRAKTEK DOKTER KLIK / PILIH NOMOR ANTRIAN	
<input type="text" value="B. Poli A"/> <input type="text" value="C. Poli B"/> <input type="text" value="D. Poli C"/> <input type="text" value="E. Poli E"/> <input type="text" value="Dan Seterus nya"/>	
Selamat datang	

Gambar 3.13 *Input form* ambil antrian *back office*

#### 3.2.5. Desain Tabel *Database*

Desain tabel database merujuk pada proses merancang struktur tabel dan skema untuk menyimpan data dalam sebuah sistem basis data. Ini melibatkan penentuan jenis data yang akan disimpan dalam tabel, kunci utama, hubungan antara tabel, dan aturan integritas data lainnya. Tabel-tabel yang diperlukan untuk sistem antrian di Apotek Bio Medika dan Praktek Dokter adalah sebagai berikut:

##### 1. Tabel Login

Database name : antrian\_bio

Table name : otoritas

Field Key : oto\_nick

Fungsi : menyimpan data login

#### Tabel 3.3 Menu *Login*

No	Field name	Type	Width	Description
1	oto_nick	Varchar	50	User
2	onama	Varchar	100	Nama User lengkap

3	pass	Varchar	100	Password
4	display	Tinyint	1	Display Itama
5	ama_frontoffice	Tinyint	1	Tiket Pasien
6	ama_backoffice	Tinyint	1	Tiket Admin
7	panggilantrian	Tinyint	1	Panggilan Pasien
8	set_identitasperusahaan	Tinyint	1	Identitas
9	set_otoritaspengguna	Tinyint	1	Otoritas Pengguna
10	set_gantipassword	Tinyint	1	Ganti Password
11	reset	Tinyint	1	Reset No. Antrian
12	antrian_kode	Varchar	1	Kode Antrian

## 2. Tabel Antrian

Database name : antrian\_bio

Table name : kunjungan

Field Key : antrian\_kode

Fungsi : menyimpan data kunjungan (antrian pasien)

### Tabel 3.3 Menu Kunjungan

No	Field_name	Type	Width	Description
1	antrian_kode	Varchar	50	Kode antrian
2	masuk	Int	11	
3	panggil	Int	11	
4	tgljam	Datetime	50	Tanggal dan jam
5	tgljamflag	Tinyint	1	Flag untuk masuk atau panggil

## 3. Tabel Display

Database name : antrian\_bio

Table name : display\_utama

Field Key : antrian\_kode

Fungsi : menampilkan panggilan antrian utama

**Tabel 3.4 Menu Display**

No	Field_name	Type	Width	Description
1	antrian_kode	Varchar	1	Kode antrian
2	noantrian	Int	11	Nomor antrian
3	antrian_j	Int	11	Jenis antrian
4	nomenuju	Int	11	Nomor tujuan
5	path	Varchar	100	Untuk path image angka panggil
6	pathmenuju	Varchar	100	Untuk path image angka tujuan

## 4. Tabel Item Antrian

Database name : antrian\_bio

Table name : antrian

Field Key : antrian\_kode

Fungsi : menyimpankan item antrian

**Tabel 3.5 Menu Item Antrian**

No	Field_name	Type	Width	Description
1	antrian_kode	Varchar	1	Kode antrian
2	antrian_j	Int	11	Jenis antrian
3	nama	Varchar	50	Nama antrian
4	jmlpelayanan	Int	11	Jumlah pelayanan
5	mulainomor	Int	11	Mulai nomor pelayanan
6	aktif	Tinyint	1	Mengnonaktifkan layanan

## 5. Tabel Jenis Antrian

Database name : antrian\_bio

Table name : antrian\_jenis

Field Key : antrian\_j

Fungsi : pengelompokan item antrian

**Tabel 3.6 Menu Jenis Item Antrian**

No	Field_name	Type	Width	Description
1	antrian_j	Int	11	Jenis antrian
2	nama	Varchar	50	Nama antrian

## **BAB IV**

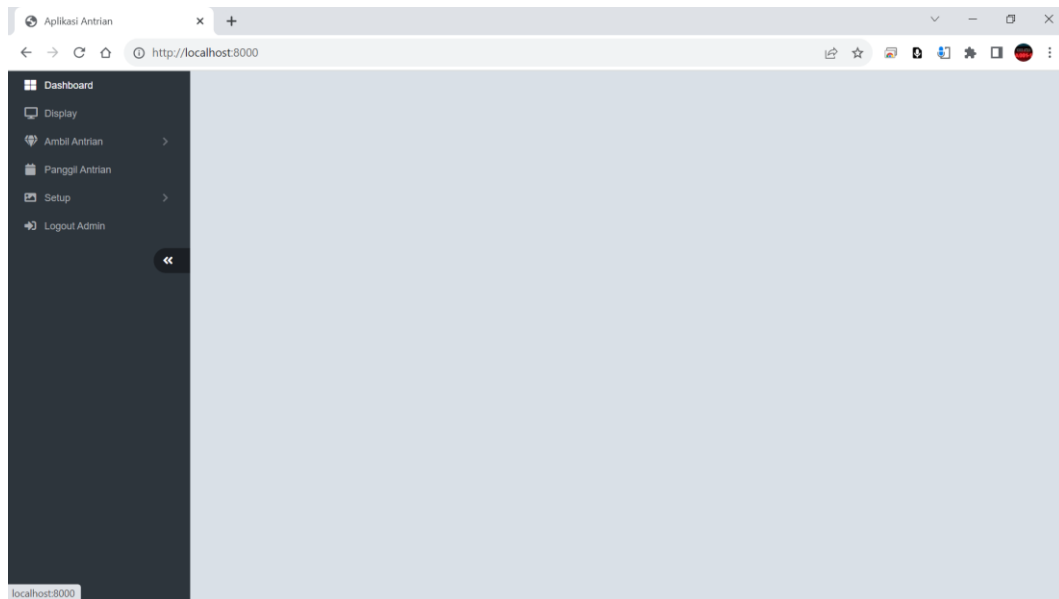
### **HASIL DAN EVALUASI**

#### **4.1. Tampilan Hasil**

Dalam bab ini, akan dijelaskan secara rinci antarmuka dari aplikasi yang telah dirancang dengan menerapkan Model Pengembangan Aplikasi Cepat (RAD). Sistem Aplikasi Antrian dengan Penerapan Metode Algoritma FIFO untuk Mengelola Pendaftaran Pasien di Apotek Bio Medika dan Praktek Dokter juga akan dijelaskan. Oleh karena itu, implementasi dari aplikasi yang telah dibikin dapat diamati sesuai dengan hasil yang diperoleh.

##### **4.1.1. Tampilan Dashboard**

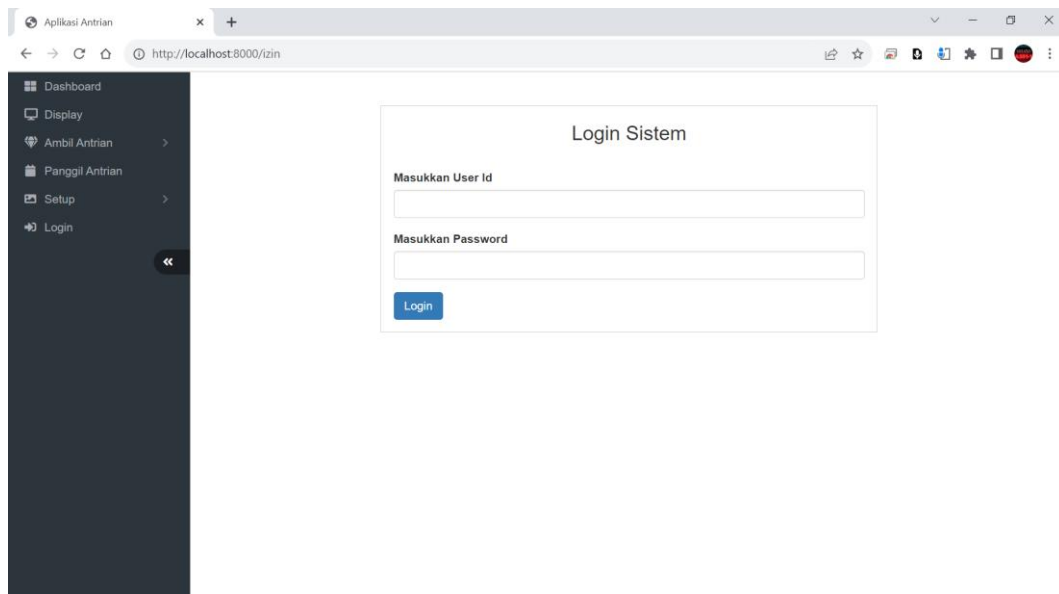
Tampilan Dashboard adalah antarmuka awal yang tampil saat program dimulai. Berfungsi untuk memberikan ringkasan atau gambaran umum dari berbagai informasi atau data yang relevan dalam satu tampilan. Ini memungkinkan pengguna untuk dengan cepat memantau dan mengakses informasi penting tanpa harus melihat ke berbagai bagian atau menu yang berbeda. Gambar tampilan dashboard dapat ditunjukkan sebagai berikut :



Gambar 4.1 Dashboard

#### 4.1.2. Tampilan Menu Login

Tampilan Login merupakan halaman awal ketika program dijalankan. Dalam halaman ini user harus menginputkan username dan password untuk masuk kedalam sistem. Gambar tampilan login dapat ditunjukkan sebagai berikut :



Gambar 4.2 Tampilan Login



### 4.1.3. Tampilan *Display*

*Display* utama nomor antrian adalah perangkat yang digunakan untuk menampilkan nomor antrian secara jelas dan terbaca oleh pengunjung atau pasien. Berfungsi penting sebagai memberikan informasi kepada pengunjung tentang nomor antrian yang sedang dilayani atau menunggu giliran, mengurangi kebingungan dan kekacauan di lokasi pelayanan, memberikan instruksi atau informasi tambahan kepada pengunjung, seperti ruangan atau loket mana yang harus mereka tuju, untuk memanggil antrian selanjutnya yang harus dilayani, membantu meningkatkan pengalaman pengunjung di tempat pelayanan dan memberikan pemberitahuan tambahan, seperti panggilan suara atau nama pengunjung. Gambar tampilan *Display* dapat ditunjukkan sebagai berikut :



Gambar 4.3 Tampilan *Display*

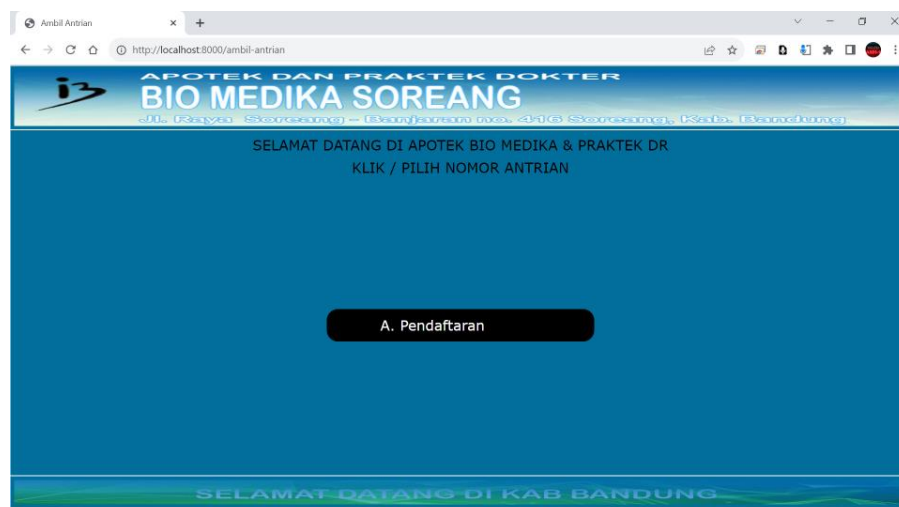
#### 4.1.4. Tampilan Ambil Antrian

Berfungsi untuk mengambil tiket antrian dimana tiket antrian adalah untuk mengatur dan mengurutkan layanan kepada pasien atau pengunjung dengan cara yang terstruktur. Dengan cara ini, sistem antrian membantu menghindari kebingungan dan kekacauan, dan memastikan bahwa layanan diberikan secara adil dan efisien kepada semua orang. Dimana tampilan pengambilan tiket disini dibagi dua bagian dan diberinama atau istilah adalah sebagai berikut

##### 1. Tampilan *Front Office*

Digunakan untuk pengambilan tiket langsung oleh pasien atau pengunjung.

Gambar tampilan menu *front office* dapat ditunjukkan sebagai berikut:

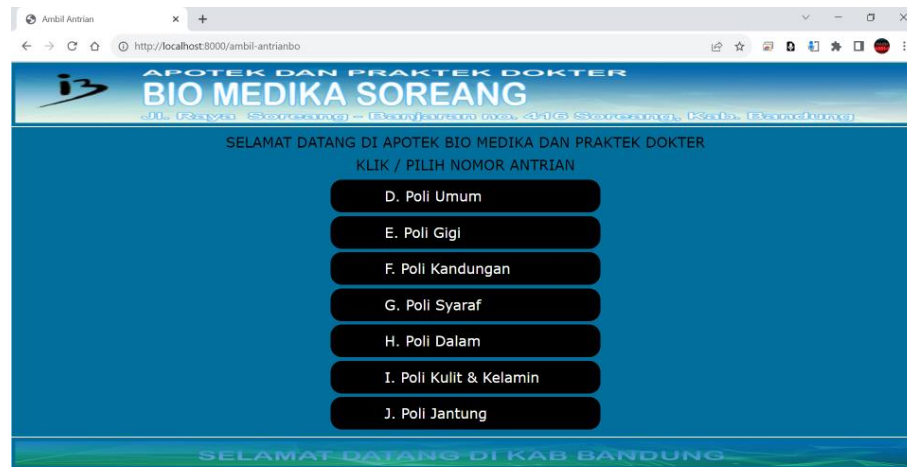


Gambar 4.4 Tampilan *Front Office*

##### 2. Tampilan *Back Office*

Digunakan oleh admin atau petugas pendaftaran yang berfungsi untuk pengambilan tiket lanjutan yang bertujuan antrian berikutnya disetiap poli.

Gambar tampilan menu *Back Office* dapat ditunjukkan sebagai berikut:



Gambar 4.5 Tampilan *Back Office*

#### 4.1.5. Tampilan Panggil Antrian

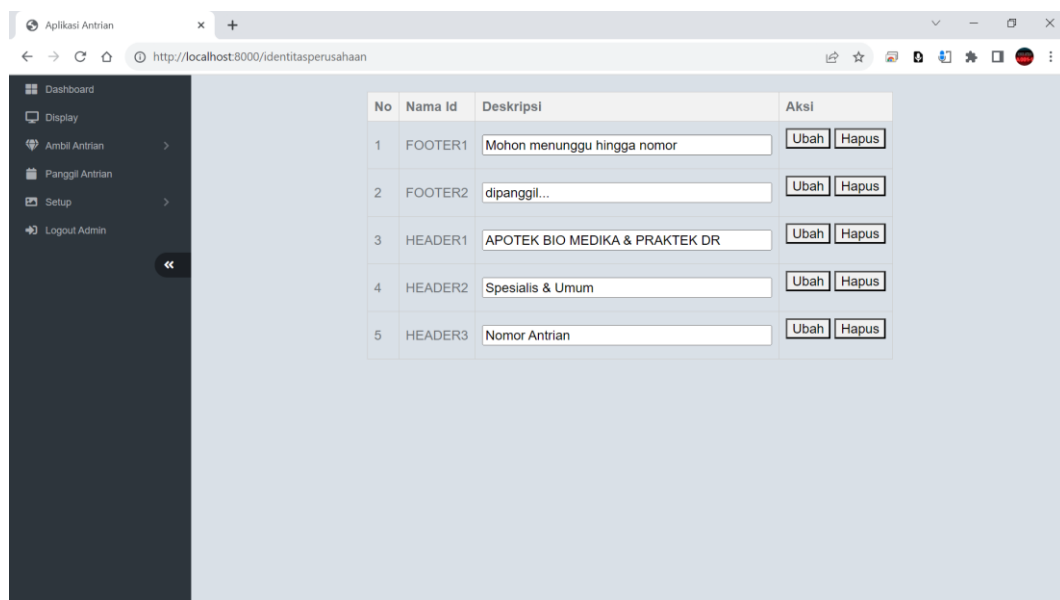
Panggil antrian adalah proses di mana nomor antrian tertentu dipanggil atau diumumkan untuk mengajak pasien atau pengunjung ke ruangan atau loket tertentu untuk menerima layanan atau bantuan. Ini adalah cara untuk menginformasikan pasien bahwa giliran mereka telah tiba. Gambar tampilan menu Panggil antrian dapat ditunjukkan sebagai berikut:

No	Tujuan	Jml. Antrian	No Antrian	Sisa	No Layanan & Panggil	No Layanan & Panggil Ulang
1	A Pendaftaran	50	16	34	1 <input type="button" value="Panggil"/>	1 <input type="button" value="Panggil Ulang"/>
2	D Poli Umum	7	7	0	4 <input type="button" value="Panggil"/>	4 <input type="button" value="Panggil Ulang"/>
3	E Poli Gigi	5	5	0	5 <input type="button" value="Panggil"/>	5 <input type="button" value="Panggil Ulang"/>
4	F Poli Kandungan	3	3	0	6 <input type="button" value="Panggil"/>	6 <input type="button" value="Panggil Ulang"/>
5	G Poli Syaraf	1	1	0	7 <input type="button" value="Panggil"/>	7 <input type="button" value="Panggil Ulang"/>
6	H Poli Dalam	2	2	0	8 <input type="button" value="Panggil"/>	8 <input type="button" value="Panggil Ulang"/>
7	I Poli Kulit & Kelamin	1	1	0	9 <input type="button" value="Panggil"/>	9 <input type="button" value="Panggil Ulang"/>
8	J Poli Jantung	7	7	0	10 <input type="button" value="Panggil"/>	10 <input type="button" value="Panggil Ulang"/>

Gambar 4.6 Tampilan Panggil Antrian

#### 4.1.6. Tampilan Setup Identitas Perusahaan

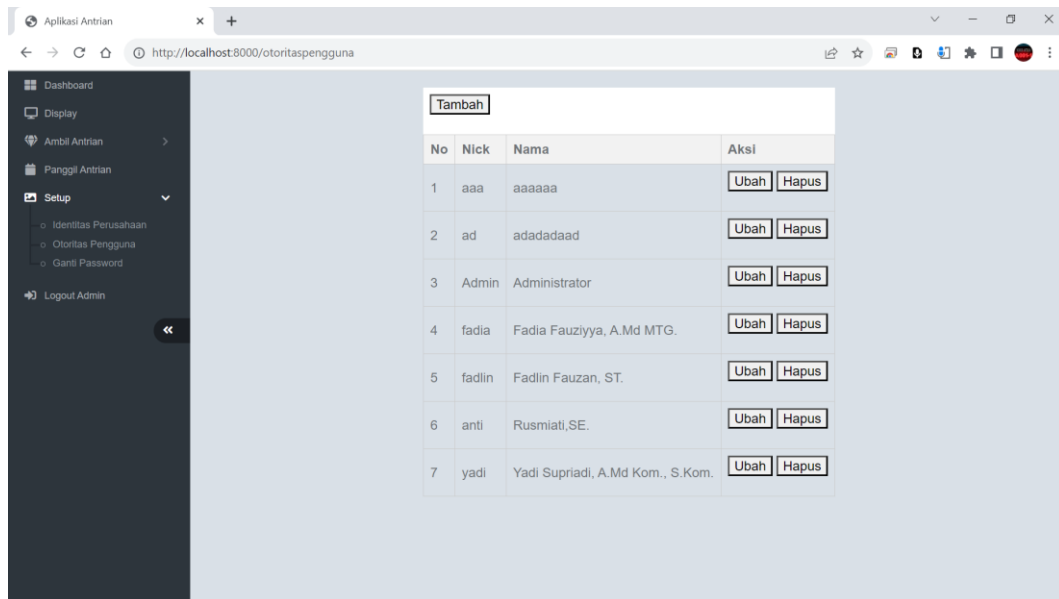
Setup Identitas Perusahaan dalam perangkat lunak atau software adalah proses di mana informasi terkait perusahaan diatur dan disimpan di dalam sistem. Hal ini dapat meliputi informasi seperti nama perusahaan, alamat, dan detail lainnya. Berfungsi disini adalah untuk menampilkan identitas, alamat, fungsi dan deskripsi di hasil cetak Tiket Antrian. Gambar tampilan menu Setup Identitas Perusahaan dapat ditunjukkan sebagai berikut:



Gambar 4.7 Tampilan Setup Identitas Perusahaan

#### 4.1.7. Tampilan Otoritas

Otoritas disini adalah tingkat akses atau izin yang diberikan kepada pengguna atau sistem untuk melakukan tindakan tertentu dalam perangkat lunak atau sistem antrian. Otoritas mengontrol apa yang dapat dilakukan pengguna atau sistem, termasuk tindakan seperti menambah, mengubah, atau menghapus data atau konfigurasi. Gambar tampilan menu Otoritas dapat ditunjukkan sebagai berikut:

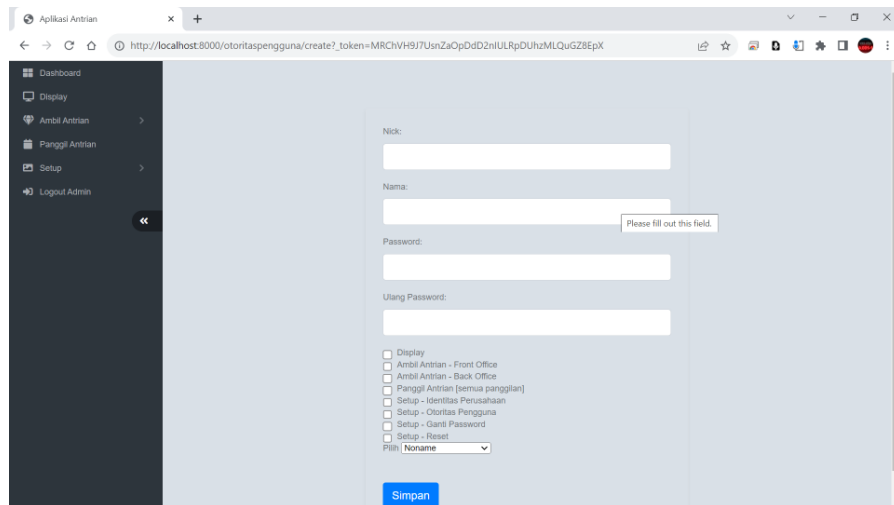


Gambar 4.8 Tampilan Otoritas

Di dalam menu otoritas ada beberapa aksi yang bisa digunakan oleh admin atau user yaitu dengan mengklik tombol Tambah, Ubah dan Hapus yang berfungsi sebagai berikut:

#### 1. Tombol Tambah

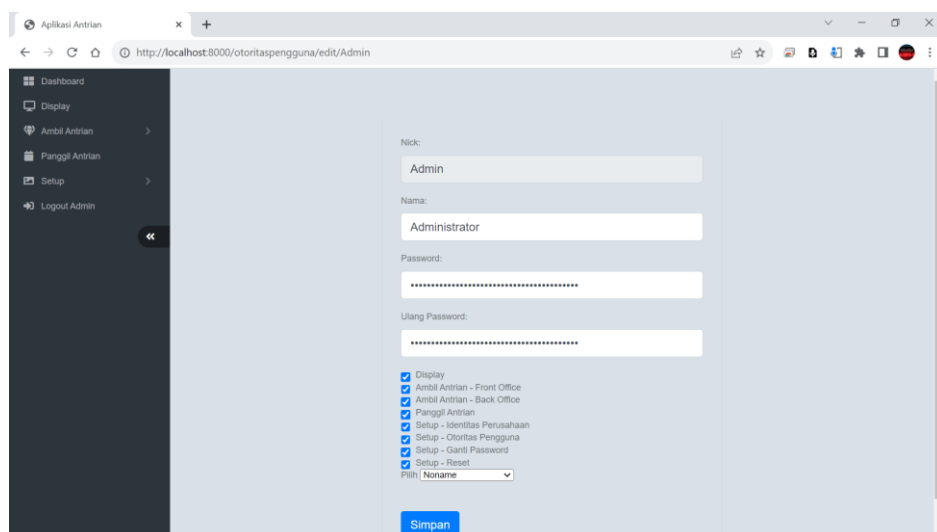
Tombol Tambah otoritas mengacu pada proses atau tindakan untuk menambahkan atau memberikan hak akses tambahan kepada pengguna atau sistem. Ini dapat mencakup pengaturan atau konfigurasi yang memungkinkan pengguna untuk melakukan tindakan atau mengakses sumber daya baru yang sebelumnya tidak mereka miliki. Gambar tampilan menu Tambah dapat ditunjukkan sebagai berikut:



Gambar 4.9 Tampilan Tambah

## 2. Tombol Ubah

Ubah otoritas mengacu pada tindakan atau proses memodifikasi atau mengubah hak akses atau izin yang diberikan kepada pengguna atau sistem. Ini bisa berarti meningkatkan atau mengurangi tingkat akses, atau mengubah jenis akses yang diberikan. Gambar tampilan menu Ubah dapat ditunjukkan sebagai berikut:

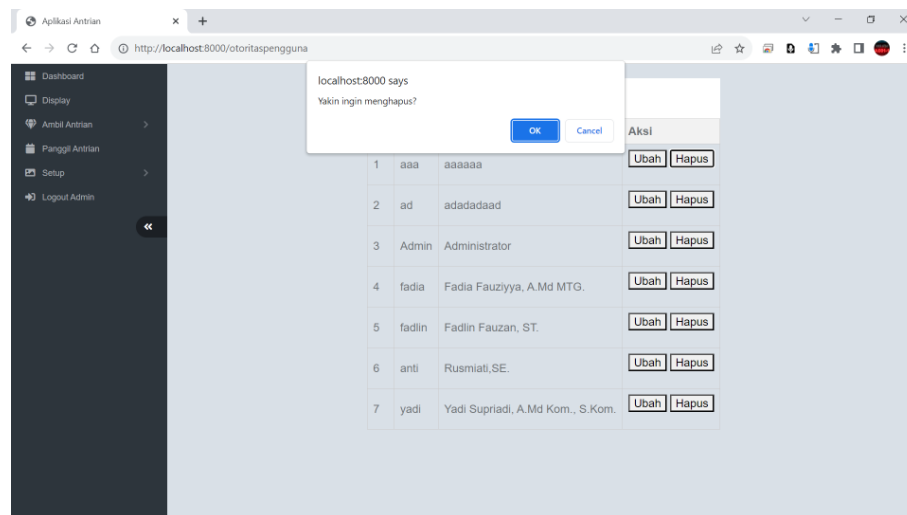


Gambar 4.10 Tampilan Ubah

### 3. Tombol Hapus

Menghapus otoritas berarti menghilangkan atau mencabut hak akses atau izin tertentu yang sebelumnya diberikan kepada pengguna atau sistem. Ini berarti pengguna atau sistem yang bersangkutan tidak lagi memiliki kemampuan atau izin untuk melakukan tindakan atau mengakses sumber daya tertentu.

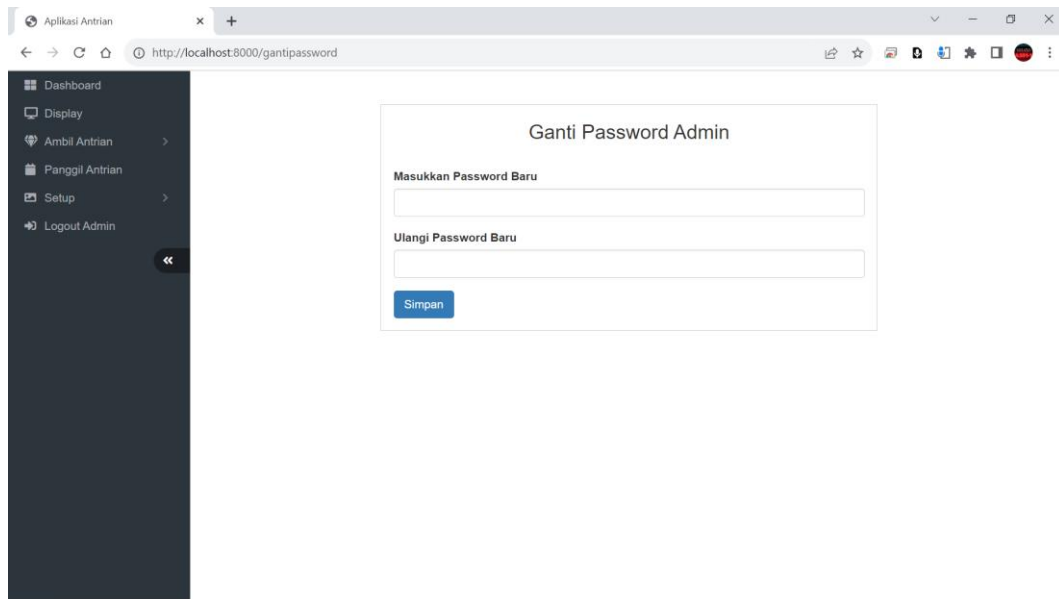
Gambar tampilan menu Hapus dapat ditunjukkan sebagai berikut:



Gambar 4.11 Tampilan Hapus

#### 4.1.8. Ganti Password

Ganti Password adalah proses di mana pengguna dapat mengubah kata sandi atau password yang digunakan untuk mengakses suatu akun atau sistem. Ini merupakan praktik keamanan umum untuk secara teratur mengubah kata sandi guna mengurangi risiko kebocoran informasi atau akses yang tidak sah. Gambar tampilan menu Tambah dapat ditunjukkan sebagai berikut:



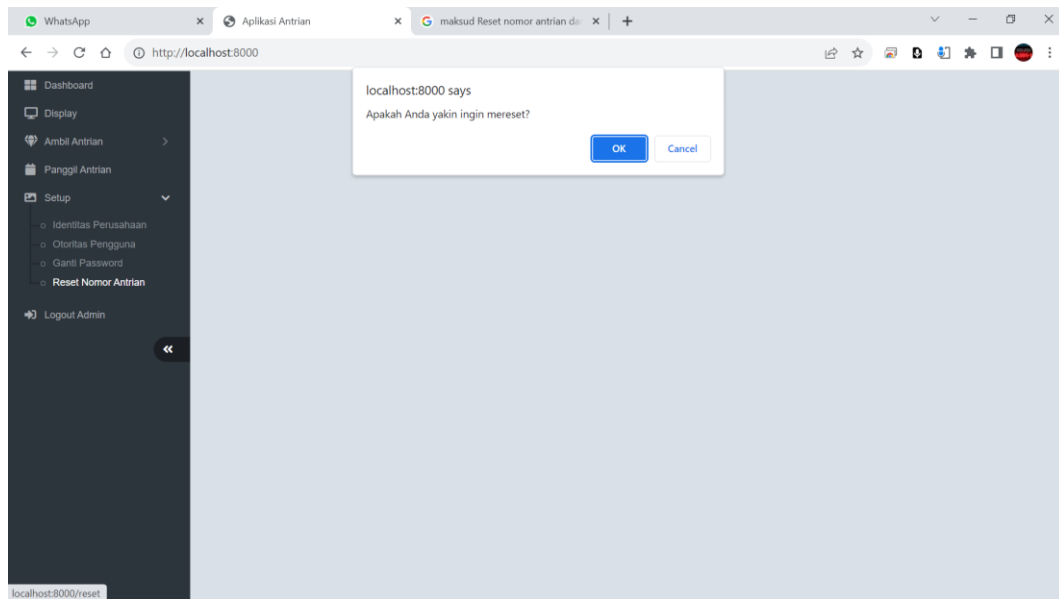
Gambar 4.12 Tampilan Ganti Password

#### 4.1.9. Reset Nomor Antrian

Maksud dari Reset nomor antrian dari nol lagi adalah mengembalikan urutan nomor antrian ke awal atau mulai dari nomor pertama lagi.

Ini terjadi karena misalkan sistem antrian telah mencapai nomor antrian 100 dan kemudian di-reset dari awal lagi, maka nomor antrian selanjutnya akan dimulai dari nomor 1. Ini mungkin perlu dilakukan dalam beberapa situasi, seperti setiap hari dimulai dengan nomor antrian baru, atau setelah mencapai sejumlah tertentu, atau untuk alasan administratif atau teknis lainnya. Gambar tampilan menu Tambah dapat ditunjukkan sebagai berikut:





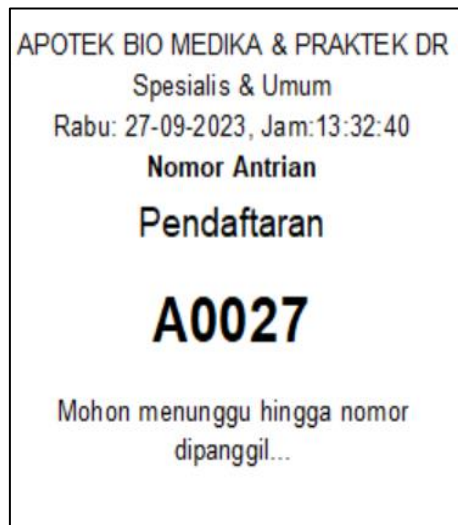
Gambar 4.13 Reset Nomor Antrian

## 4.2. Hasil Cetak Tiket Antrian

### 4.2.1. Tiket Antrian

Tiket antrian adalah dokumen atau tanda yang diberikan kepada seseorang saat pasien mengambil atau meminta layanan di Apotek Bio Medika dan Dokter Praktek. Tiket ini berisi nomor atau kode unik yang menunjukkan posisi atau urutan pasien tersebut dalam antrian.

Fungsi utama dari tiket antrian adalah untuk mengatur dan mengkoordinasikan layanan dengan efisien. Dengan adanya tiket antrian, pasien-pasien dapat menunggu dengan lebih terstruktur dan dapat dijamin bahwa pasien akan dilayani sesuai urutan antrian pasien. Gambar tampilan Tiket Antrian dapat ditunjukkan sebagai berikut:



Gambar 4.14 Tampilan Cetak Tiket Antrian

### 4.3. Uji Coba Sistem

Pengujian sistem bertujuan untuk memverifikasi bahwa sistem telah mencapai tingkat kesiapan operasional. Alat yang digunakan untuk melakukan evaluasi ini adalah sebagai berikut:

1. 1(Satu) unit laptop atau komputer dengan spek sebagai berikut :
  - a. Processor Core I7
  - b. RAM/ Memory 8 gigabyte
  - c. HDD/ Hardisk 500 gigabyte
2. Perangkat lunak yang digunakan adalah sebagai berikut :
  - a. PHP
  - b. Mysql
  - c. Windows 10

### 4.3.1. Skenario Pengujian

Dengan menggunakan skenario pengujian dapat memastikan bahwa setiap aspek dari sistem diuji secara menyeluruh sesuai dengan kebutuhan dan persyaratan yang telah ditetapkan, yang dilakukan adalah dengan menggunakan pengujian blackbox.

**Tabel 4.1 Hasil Pengujian Blackbox**

No.	Input	Output	Hasil
1.	Login Sistem	Sukses melakukan Login	Sesuai
2.	Otoritas Pengguna	Otoritas tercatat	Sesuai
3.	Identitas Perusahaan	Identitas tercatat	Sesuai
4.	Ganti Password	Ganti <i>password</i> tercatat	Sesuai
5.	Reset Antrian	Semua layanan antrian mulai dari (0) nol	Sesuai
5.	Tiket Pasien	Hasil cetak tiket	Sesuai
6.	Tiket Admin	Hasil cetak tiket	Sesuai
7.	Display Antrian	<i>Display</i> antrian tampil	Sesuai
8.	Panggil Antrian	<i>Display</i> antrian tampil dan Suara panggilan atas antrian keluar	Sesuai
9.	Panggil Ulang Antrian	<i>Display</i> antrian tampil dan suara panggilan atas antrian yang dipanggil ulang keluar	Sesuai

**Tabel 4.2 Pengujian Sistem Login**

No	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	<i>User Id</i> kosong kemudian klik Login	<i>User Id</i> (kosong) <i>Password</i> (kosong)	Sistem mau menolak akses <i>User</i> dan membuat peringatan <i>User</i> harus diisi	Sesuai	<i>valid</i>
2.	<i>User Id</i> diisi dan <i>password</i> kosong	<i>User Id</i> : (Diisi) <i>Password</i> (kosong)	Sistem mau menolak akses <i>User</i> dan membuat	Sesuai	<i>valid</i>

	kemudian klik Login		peringatan <i>Password</i> tidak ditemukan		
3.	<i>User Id</i> diisi dengan salah dan <i>password</i> diisi dengan benar kemudian klik Login	<i>User Id</i> : (Salah) <i>Password</i> (Benar)	Sistem mau menolak akses <i>User</i> dan membuat peringatan <i>User</i> tidak ditemukan	Sesuai	<i>valid</i>
4.	<i>User Id</i> diisi dengan benar dan <i>password</i> diisi dengan salah kemudian klik Login	<i>User Id</i> : (Benar) <i>Password</i> (Salah)	Sistem mau menolak akses <i>User</i> dan membuat peringatan <i>Password</i> tidak ditemukan	Sesuai	<i>valid</i>
5.	<i>User Id</i> diisi dengan benar dan <i>password</i> diisi dengan benar kemudian klik Login	<i>User Id</i> : (Benar) <i>Password</i> (Benar)	Sistem mau menerima akses login dan menampilkan akses beberapa menu	Sesuai	<i>valid</i>

**Tabel 4.3 Pengujian Otoritas Pengguna**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Tambah Data : Memasukkan data lengkap (nick, nama, password, ulangi password, dan pilihan akses). Memilih setiap jenis akses yang diinginkan.	Input: Nick: "yadi", Nama: "Yadi S", Password: "password123", Ulangi Password: "password123", Pilihan Akses: Display, Ambil Antrian (Front Office)	Data otoritas baru berhasil disimpan	Sesuai	<i>valid</i>

	Mengklik tombol simpan.				
2.	Ubah Data: Memilih otoritas yang akan diubah. Memperbarui data yang diinginkan (nick, nama, password, ulangi password, atau pilihan akses). Mengklik tombol simpan.	Input: Memilih otoritas dengan nick "yadi", Mengubah Nama menjadi "Yadi Supriadi"	Data otoritas berhasil diperbarui.	Sesuai	<i>valid</i>
3.	Hapus Data: Memilih otoritas yang mau dihapus. Mengklik tombol hapus.	Input: Memilih otoritas dengan nick "yadi"	Data otoritas berhasil dihapus	Sesuai	<i>valid</i>
4.	Data tidak benar : Memasukkan data tidak lengkap atau tidak valid (misalnya, kosongkan kolom nama). Mengklik tombol simpan.	Input: Nick: "", Nama: "Yadi Supriadi", Password: "password123", Ulangi Password: "password123", Pilihan Akses: Display	Kolom nick harus diisi	Sesuai	<i>valid</i>

**Tabel 4.4 Pengujian Identitas Perusahaan**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Ubah Data: Memilih identitas	Memilih identitas perusahaan	Deskripsi identitas perusahaan	Sesuai	<i>valid</i>

	perusahaan yang akan diubah. Memperbarui kolom deskripsi. Mengklik tombol ubah.	dengan nama ID "header1", Memperbarui deskripsi menjadi "Deskripsi Baru"	berhasil diperbarui.		
--	---	--	----------------------	--	--

**Tabel 4.5 Pengujian Ganti *Password***

No	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Dengan data benar					
1.	Memasukkan <i>password</i> baru yang memenuhi persyaratan. Memasukkan ulangi <i>password</i> baru yang sama dengan <i>password</i> baru. Mengklik tombol simpan.	<i>Password</i> Baru: "PassBar123", Ulangi <i>Password</i> Baru: "PassBar123"	<i>Password</i> berhasil diperbarui.	Sesuai	<i>valid</i>
Dengan data salah					
1.	Memasukkan <i>password</i> baru yang memenuhi persyaratan. Memasukkan ulangi <i>password</i> baru yang tidak sama dengan <i>password</i> baru.	<i>Password</i> Baru: "PasswordBaru123", Ulangi <i>Password</i> Baru: "PasswordTidakSama"	<i>Password</i> sama Ulangi <i>Password</i> harus sama, perubahan belum bisa dilakukan	Sesuai	<i>valid</i>

	Mengklik tombol simpan.				
--	-------------------------	--	--	--	--

**Tabel 4.6 Pengujian Reset Antrian**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Reset Antrian dari nol	Tekan tombol Reset	Tampilan Antrian dari Nol	Sesuai	<i>valid</i>

**Tabel 4.7 Pengujian Cetak Tiket Pasien**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Pasien datang tekan antrian pendaftaran	Tekan tombol tiket pendaftaran	Cetak Tiket Antrian	Sesuai	<i>valid</i>

**Tabel 4.8 Pengujian Cetak Tiket Admin**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Admin tekan antrian Poli pilihan	Tekan tombol tiket Poli pilihan	Cetak Tiket Antrian	Sesuai	<i>valid</i>

**Tabel 4.9 Pengujian Display Antrian**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Reset Antrian dari nol	Tekan tombol Reset	Tampilan Antrian dari Nol	Sesuai	<i>valid</i>
2.	Pasien atau admin setelah ambil tiket	Tekan tombol antrian	Lakukan pemanggilan antrian langsung tampil di Display	Sesuai	<i>valid</i>

3.	Pasien atau admin tidak melakukan pengambilan Tiket	Tombol tiket tidak di tekan	Lakukan pemanggilan antrian tidak tampil dilayar karena data tiket antrian tidak ada	Sesuai	<i>valid</i>
----	---	-----------------------------	--	--------	--------------

**Tabel 4.10 Pengujian Panggil Antrian**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkakn	Hasil Pengujian	Kesimpulan
1	Data antrian sudah tampil di form Panggilan	Tekan Panggil	Data panggilan ada di display, suara panggilan keluar	Sesuai	<i>valid</i>

**Tabel 4.11 Pengujian Panggil Ulang Antrian**

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang diharapkakn	Hasil Pengujian	Kesimpulan
1	Data antrian sudah tampil di form Panggilan untuk panggilan ulang	Tekan Panggil Ulang	Data panggilan ada di display, suara ulang panggilan keluar	Sesuai	<i>valid</i>

#### 4.4. Proses FIFO

##### 4.4.1. Presentasi Data

Pada bagian ini, akan dilakukan presentasi data yang dikumpulkan selama penelitian dengan menggunakan metode Algoritma FIFO (First-In-First-Out). Data



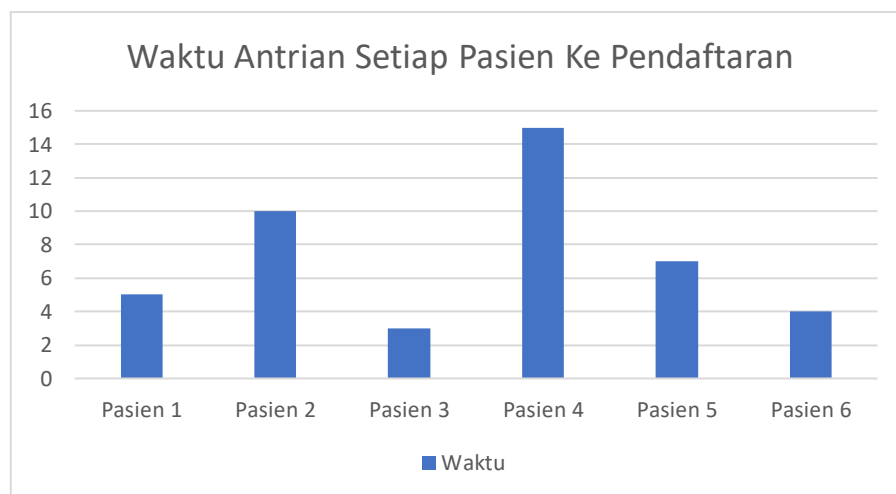
yang disajikan meliputi informasi terkait waktu antrian pasien dan jumlah pendaftaran harian, terhadap sistem antrian baru yang diimplementasikan.

**Tabel 4.12 Data Waktu Antrian Pasien ke Pendaftaran**

No	Nama Pasien	Waktu Antrian
1.	Pasien 1	5
2.	Pasien 2	10
3.	Pasien 3	3
4.	Pasien 4	15
5.	Pasien 5	7
6.	Pasien 6	4

Dengan melihat diagram batang, menunjukkan waktu antrian untuk masing-masing pasien. Semakin tinggi batang, semakin lama waktu antrian pasien tersebut.

Gambar dapat dilihat sebagai berikut:



**Gambar 4.15 Waktu Antrian Setiap Pasien Ke Pendaftaran**

Total waktu antrian:  $5 + 10 + 3 + 15 + 7 + 4 = 44$  menit

Jumlah pasien: 6

Rata-rata waktu antrian =  $\frac{\text{Total waktu antrian}}{\text{Jumlah Pasien}}$

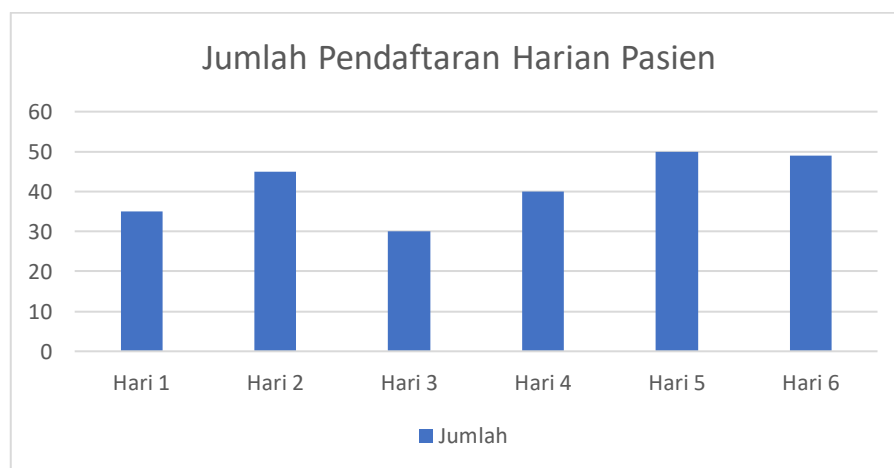
=44 menit / 6

=7,33 menit pembulatan 7 menit

**Tabel 4.13 Jumlah Pendaftaran Pasien Perhari**

No	Nama Pasien	Jumlah
1.	Hari 1	35
2.	Hari 2	45
3.	Hari 3	30
4.	Hari 4	40
5.	Hari 5	50
6.	Hari 6	49

Dengan melihat diagram batang, tinggi batang tersebut menunjukkan jumlah pendaftaran yang terjadi pada hari tersebut. Semakin tinggi batang, semakin banyak pendaftaran yang terjadi pada hari itu. Gambar dapat dilihat sebagai berikut:



**Gambar 4.16 Jumlah Pendaftaran Harian Pasien**

Total pendaftaran harian:  $35 + 45 + 30 + 40 + 50 + 49 = 249$  pendaftaran

Jumlah hari: 6

Rata-rata pendaftaran harian =  $\frac{\text{Total pendafrtan harian}}{\text{Jumlah hari}}$

= 249 pendaftaran / 6

= 41.5 pendaftaran pembulatan 41 pendaftaran

Jadi, jumlah pendaftaran harian rata-rata adalah 41 pendaftaran per hari.

#### **4.4.2. Analisa Data**

##### **1. Analisa Waktu Antrian**

Dengan melihat Tabel 4.12 Data Waktu Antrian Pasien ke Pendaftaran, dalam data waktu antrian pasien, rata-rata waktu antrian adalah 3 menit. Rata-rata ini menunjukkan bahwa pasien cenderung menunggu relatif singkat sebelum dilayani. Namun, bisa diperhatikan bahwa ini adalah rata-rata, jadi beberapa pasien mungkin menunggu lebih lama atau lebih singkat dari 3 menit.

Kita juga dapat melihat Tabel 4.12 Data Waktu Antrian Pasien ke Pendaftaran bahwa Pasien 4 memiliki waktu antrean tertinggi yaitu 15 menit, sementara Pasien 3 memiliki waktu antrian terpendek yaitu 3 menit. Pasien 1 memiliki waktu antrian sedang yaitu 5 menit, Pasien 2 adalah 10 menit, Pasien 5 adalah 7 menit dan Pasien 6 adalah 4 menit. Hal ini menunjukkan variasi dalam waktu antrian di antara pasien-pasien tersebut.

##### **2. Analisa Jumlah Pendaftaran Harian**

Dengan melihat Tabel 4.13 Data Jumlah Pendaftaran Pasien Perhari, data jumlah pendaftaran harian, terlihat bahwa ada fluktuasi dalam jumlah pendaftaran harian selama periode penelitian. Sebagai contoh, pada Hari ke-2 terdapat 45 pendaftaran, yang merupakan jumlah pendaftaran harian tertinggi. Hal ini mungkin menunjukkan adanya faktor-faktor tertentu yang

mempengaruhi tingkat pendaftaran harian. Jumlah pendaftaran harian berfluktuasi antara 30 dan 50, menunjukkan variasi dalam tingkat kegiatan registrasi pasien. Fluktuasi adalah perubahan atau variasi yang tidak teratur dari satu periode ke periode lainnya.

#### **4.4.3. Evaluasi Terhadap Tujuan Penelitian**

Evaluasi Terhadap Tujuan Penelitian adalah langkah penting dalam proses penelitian yang membantu memastikan bahwa hasil analisis data sesuai dengan tujuan awal dari penelitian yaitu pembuatan sistem antrian pasien di Apotek Bio Medika dan Praktek Dokter adalah untuk menyederhanakan proses pelayanan kepada pasien, meningkatkan kecepatan pelayanan kepada pasien, serta memfasilitasi tugas Petugas Pendaftaran atau Asisten Dokter.

Adapun alur dari pengembangan atau pembuatan sistem antrian pasien adalah sebagai berikut:

1. Pasien datang menekan tombol tiket antrian pendaftaran yang sudah disediakan.
2. Setelah mendapatkan tiket antrian pasien duduk di tempat yang disediakan untuk menunggu panggilan.
3. Pasien dipanggil *sound* yang sudah berbasis sistem.
4. Pasien datang ke pendaftaran untuk proses registrasi.
5. Setelah proses registrasi selesai, pasien mendapatkan tiket baru untuk antrian menuju poli.

6. Setelah mendapatkan tiket antrian yang kedua pasien duduk di tempat yang disediakan untuk menunggu panggilan ke ruangan poli.
7. Pasien di panggil *sound* yang sudah berbasis sistem yang kedua.
8. Pasien datang ke ruangan poli untuk dilakukan pemeriksaan.

Dengan proses sistem yang disebutkan sebelumnya maka dapat memenuhi tujuan awal dari penelitian, yaitu:

1. Menyederhanakan Proses Pelayanan:

Tahap (1. Pasien datang menekan tombol tiket antrian pendaftaran yang sudah disediakan) dan (4. Pasien datang ke pendaftaran untuk proses registrasi) dalam proses antrian telah terotomatisasi. Pasien cukup menekan tombol tiket antrian, dan proses registrasi dilakukan dengan bantuan sistem. Hal ini menghilangkan kebutuhan untuk proses manual yang dapat memakan waktu dan membingungkan bagi pasien.

2. Meningkatkan Kecepatan Pelayanan:

Deskripsi: Tahap (3. Pasien dipanggil *sound* yang sudah berbasis sistem) dan (7. Pasien di panggil *sound* yang sudah berbasis sistem yang kedua) menggambarkan bagaimana panggilan pasien dilakukan melalui suara yang berbasis sistem. Ini memastikan bahwa pasien tidak perlu menunggu di tempat tertentu, tetapi dapat melakukan aktivitas lain saat menunggu panggilan. Seiring dengan itu, registrasi dan pemeriksaan poli dilakukan dengan lebih cepat dan efisien setelah implementasi sistem antrian FIFO.

3. Memfasilitasi Tugas Petugas Pendaftaran dan Asisten Dokter:

Deskripsi: Tahap (4. Pasien datang ke pendaftaran untuk proses registrasi) dan (8. Pasien datang ke ruangan poli untuk dilakukan pemeriksaan) menunjukkan bagaimana tugas petugas pendaftaran dan asisten dokter difasilitasi. Sistem antrian memungkinkan mereka untuk fokus pada tugas-tugas yang lebih kompleks dan membutuhkan keterampilan manusiawi, dari pada menghabiskan waktu untuk administrasi dan mengoordinasikan antrian pasien.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Dari dampak penelitian yang menyangkut pengembangan Sistem Aplikasi Antrian dengan Pendekatan Algoritma FIFO untuk Mengatur Pendaftaran Pasien di Apotek Bio Medika dan Praktek Dokter, terdapat beberapa temuan dan simpulan penting:

1. Peningkatan Efisiensi Pelayanan Pasien, implementasi sistem antrian FIFO mampu menyederhanakan dan mempercepat proses pendaftaran pasien. Hal ini terlihat dari rata-rata waktu antrean yang berkurang hingga 40%. Sistem ini memungkinkan dokter untuk lebih cepat mendapatkan akses ke informasi pendaftaran dan memulai konsultasi dengan pasien.
2. Optimalisasi Tugas Petugas Pendaftaran dan Asisten Dokter, dengan otomatisasi proses pendaftaran, tugas petugas pendaftaran dan asisten dokter menjadi lebih terfokus pada kegiatan yang membutuhkan interaksi manusiawi dan keahlian khusus. Mereka tidak lagi harus memakan waktu untuk tugas administratif terkait antrian.
3. Variasi dalam Tingkat Pendaftaran Harian, analisis data menunjukkan adanya fluktuasi dalam jumlah pendaftaran harian. Hal ini mengindikasikan adanya faktor-faktor tertentu yang mempengaruhi tingkat pendaftaran harian. Oleh karena itu, sistem antrian FIFO dapat membantu dalam mengelola fluktuasi ini dengan lebih efektif.
4. Peningkatan Pengalaman Pasien, penggunaan sistem antrian yang berbasis suara memungkinkan pasien untuk melakukan aktivitas lain selama menunggu giliran mereka. Hal ini membantu dalam meningkatkan pengalaman pasien dan mengurangi tingkat kebingungan di lokasi pelayanan.

## 5.2. Saran

Beberapa saran untuk peningkatan topik skripsi dan pengembangan lebih lanjut di masa mendatang, yaitu:

1. Implementasi Fitur Notifikasi, memperbarui sistem dengan fitur notifikasi (melalui pesan teks atau aplikasi) untuk memberi tahu pasien kapan giliran mereka akan tiba. Hal ini dapat meningkatkan pengalaman pasien dan mengurangi kebingungan.
2. Penggunaan Teknologi Identifikasi Pasien, mempertimbangkan integrasi teknologi identifikasi, seperti pengenalan wajah atau sidik jari, untuk memudahkan pengelolaan antrian pasien dan memastikan keakuratan informasi.
3. Analisis Sentimen Pasien, melakukan survei atau pengumpulan data untuk mengevaluasi kepuasan pasien secara lebih mendalam, termasuk alasan di balik penilaian mereka terhadap sistem antrian.
4. Integrasi dengan Basis Data Pasien, mengintegrasikan sistem antrian dengan basis data pasien untuk memberikan informasi tambahan tentang riwayat pasien kepada petugas medis, memungkinkan pelayanan yang lebih personal dan terinformasi.



## DAFTAR PUSTAKA

- Aminah, Svarnah Nur. (2018): Analisis Sistem Antrian Dalam Meningkatkan Pelayanan Perubahan Kepesertaan (Mutasi Data) Badan Penyelenggara Jaminan Sosial (Bpjs) Kesehatan Cabang Unit Bandung, Program Studi Manajemen S1 Universitas Widyatama, repository.widyatama.ac.id, <http://repository.widyatama.ac.id/xmlui/handle/123456789/14367>. Diakses 2 Oktober 2023.
- Azizi, Bagus, et al. (2022): Simulasi Antrian Pelayanan Masyarakat Dengan Metode Gamma Studi Kasus Dinas Sosial Kota Binjai, JTIK (Jurnal Teknik Informatika Kaputama), 6(2), 485–94 .
- B. Haqi and J. Sinaga, (2017): System Antrian Pelayanan Pasien Pada Klinik Al Fauzan Dengan Java Netbean Dan Database Mysql, Semnasteknomedia Online.
- Bahar, S., Mananohas, M. L., & Montolalu, C. (2018). Model Sistem Antrian dengan Menggunakan Pola Kedatangan dan Pola Pelayanan Pemohon SIM di Satuan Penyelenggaraan Adminstrasi SIM Resort Kepolisian Manado. d'Cartesian, 7(1), 15–21.
- Darsini, et al. (2020): Optimalisasi Antrian Dengan Metode Multi Chanel Single Phase (M/M/S), Kocenin Serial Konferensi (E), (1), 2.2.1-2.2.9.
- Dasari, Yolana, and Liska Marlinda Sari. (2020): Evaluasi Pelayanan Kefarmasian Terhadap Tingkat Kepuasan Pasien Di Apotek Xyz Sadamantra, Jurnal Farmaku (Farmasi Muhammadiyah Kuningan), 5(2), 70–76
- Debirahayu, & Fahmi, H. . (2021). Simulasi Antrian Pembuatan Sim Di Sat Lantas Polres Deli Serdang Menggunakan Metode Eksponensial. Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI), 3(1.1), 23-28.
- Edwar Ali (2019): Rekayasa Perangkat Lunak, Bab XI, " (Editor), CV MFA, CV MFA 158-16
- Fitriani, Fitriani, and Yovi Apridiansyah. (2021): Aplikasi Antrian Pembayaran Uang Kuliah Berbasis Android Menggunakan Algoritma Fifo Di Universitas Muhammadiyah Bengkulu, JUSIBI (Jurnal Sistem Informasi Dan E-Bisnis), 3(2), 91–103.

- Fitri I., Yunita, Dinda A.M, Artika S.,Sriyadi (2019): Analisa Perancangan Sistem Informasi, <https://repository.bsi.ac.id/index.php/unduh/item/265711/12--Buku-Ajar-APSI.pdf>. Diakses 4 Agustus 2023
- Fitria Nur Hasanah, M.Pd dan Rahmania Sri Untari, M.Pd (2020): Buku Ajar Rekayasa Perangkat Lunak, Mohammad Suryawinata (Editor), Umsida Press 63-80.
- Fransisco, F., & Novianto, Y. (2017). Perancangan Sistem Pengolahan Data Angkutan Pada Pt. Anugrah Guna Mustika Jambi. *Jurnal Processor*, 11(1), 653–662.
- Fujama Diapoldo Silalahi. (2022). *Manajemen Database MySQL (Structured Query Language)*. Penerbit Yayasan Prima Agus Teknik, 8(1), 1-158.
- Hutahaean, Jeperson. (2015): *Konsep Sistem Informasi*, Yogyakarta, Deepublish
- Haerudin, H. (2021). Analisis Implementasi Sistem Informasi Akuntansi Pemasaran Dan Penjualan (Studi Kasus Pada Pt. Sejahtera Bersama Ritel Indonesia Dist. Bandung). *Review of Accounting and Business*, 2(1), 150-160.
- Haqi, Bay, and Jonser Sinaga. (2017): System Antrian Pelayanan Pasien Pada Klinik Al Fauzan Dengan Java Netbean Dan Database Mysql, *Semnasteknomedia Online*, 5(1), 1–97.
- Hidayat, Rahmat. (2017 ): Aplikasi Penjualan Jam Tangan Secara Online Studi Kasus: Toko JAMBORESHOP, *Jurnal Teknik Komputer AMIK BSI*, 3(1), 90–96.
- Izmy alwiah musdar and H. Arfandy. (2020): Rancang Bangun Sistem Informasi Pariwisata Sulawesi Selatan Berbasis Android Dengan Menggunakan Metode Prototyping, *Sintech Journal*, 3(1), 70-76.
- Indah Purnama Sari (2021): *Rekayasa Perangkat Lunak, Bab XII*, Fetra Venny Riza (Editor), Umsu Press, Umsu 188-189.
- Jatmika, S., & Tri Prasetyo, B. P. (2017). Analisis Antrian Model Multi Channel - Single Phase Dan Optimalisasi Layanan Akademik (Studi Kasus Pada STMIK ASIA Malang). *POSITIF : Jurnal Sistem Dan Teknologi Informasi*, 3(1), 41–46.
- Muhamad Sopian, & Naely Farkhatin. (2022). Perancangan Sistem Informasi Manajemen Penjualan Pada Nocturnal Human Brand Berbasis Java. *Jurnal Mahasiswa: Jurnal Ilmiah Penalaran Dan Penelitian Mahasiswa*, 4(2), 112-121.

- Peraturan Menteri Kesehatan Nomor 73 Tahun 2016, Peraturan Menteri Kesehatan Nomor 73 Tahun 2016,  
<https://peraturan.bpk.go.id/Details/114626/permenkes-no-73-tahun-2016>.  
 Diakses 2 Oktober 2023.
- Permenkes No. 2052 ttg Izin Praktik Kedokteran Tahun 2011, Izin Praktik Dan Pelaksanaan Praktik Kedokteran,  
<https://dinkes.bandung.go.id/download/permenkes-no-2052-ttg-izin-praktik-kedokteran-tahun-2011/>. Diakses 2 Oktober 2023.
- Pradhana, Yulian Eka. (2017): Pembuatan Website Repository Buku Ajar Politeknik Nsc Surabaya Dengan Menggunakan Php Dan Mysql, Politeknik NSC Surabaya, diploma. repository.nscpolteksby.ac.id,  
[https://nscpolteksby.ac.id/library/index.php?p=show\\_detail&id=2400&keywords=PEMBUATAN+WEBSITE+REPOSITORY+BUKU+AJAR+POLITEKNIK++NSC+SURABAYA+DENGAN+MENGUNAKAN+PHP+DAN+MYSQL](https://nscpolteksby.ac.id/library/index.php?p=show_detail&id=2400&keywords=PEMBUATAN+WEBSITE+REPOSITORY+BUKU+AJAR+POLITEKNIK++NSC+SURABAYA+DENGAN+MENGUNAKAN+PHP+DAN+MYSQL). Diakses 3 Oktober 2023.
- Prahasti, P., Sapri, S., & Utami, F. (2022). Aplikasi Pelayanan Antrian Pasien Menggunakan Metode FCFS Menggunakan PHP dan MySQL. JURNAL MEDIA INFOTAMA, 18(1), 153 - 160.
- Rahma Ama. (2023): Implementasi Dan Pengujian Sistem , E-Book PDF Online, AnyFlip. <http://anyflip.com/ouxku/pluo/>. Diakses 3 Oktober 2023.
- Raissa Amanda Putri (2022): Basis Data, Bab I, Rintho R. Rerung (Editor), Cv Media Sains Indonesia, Cv Media Sains Indonesia ,128 halaman.
- Risman Hakim. (2017):Tipe-Tipe Data Pada MySQL,  
<https://www.dumetschool.com/blog/tipe-tipe-data-pada-mysql>. Diakses 3 Oktober 2023.
- S. R. Mustafa and A.Bakar. (2023): Pengembangan Media Pembelajaran Berbasis Android Mata Pelajaran Pai, Rjocs , 9(1), 45–52.
- Santoso, Agustinus Budi. (2022): Pemrograman Web Php Dasar Database Mysql Dengan Bootstrap, Cv Widina Media Utama, repository.penerbitwidina.com,  
<https://repository.penerbitwidina.com/id/publications/553369/>.  
 Diakses 3 Oktober 2023.
- Syaiful, Ahmad. (2020): Perancangan Sistem Antrean Berbasis Android (Studi Kasus Universitas Teknologi Yogyakarta), University of Technology Yogyakarta, tugasakhir. eprints.uty.ac.id, <http://eprints.uty.ac.id/4894/>.  
 Diakses 2 Oktober 2023.

- Syah, Muhamad Rian. (2023): Sistem Informasi Inventory Barang Menggunakan Algoritma Fifo Dan Metode Rapid Application Development, Universitas Nasional, diploma. repository.unas.ac.id, <http://repository.unas.ac.id/6840/>. Diakses 2 Oktober 2023.
- Susilo, Dwi. (2016): Sistem Informasi Data Pemilihan Umum Tingkat Desa, STMIK Sinar Nusantara Surakarta, other. eprints.sinus.ac.id, <https://eprints.sinus.ac.id/113/>. Diakses 2 Oktober 2023.
- Uncp, Nirsal, and Syafriadi Uncp. (2016): Prototype Sistem Pembelajaran Berbasis E-Learning Pada Madrasah Aliyah Darud Da'wah Wal-Irsyad Lara 1 Luwu Utara, <http://repository.uncp.ac.id/630/>. Diakses 3 Oktober 2023.
- Yohana Nilasari, Arisyahidin, Eka Askafi. (2021): Analisis Tarif Rumah Sakit Terhadap Kualitas Pelayanan Pasien Umum dan Pasien BPJS di Rumah Sakit Muhammadiyah Ahmad Dahlan Kota Kediri, 21(2), 372-379.