

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian Penerapan sistem informasi manajemen proyek pengelolaan stok obat berbasis *website* juga dibahas oleh Anggy dan Yahfizham (2023). Mereka mengusulkan sistem yang membantu pelanggan melihat informasi obat secara *real-time*, mengatasi masalah ketersediaan obat, dan meningkatkan kualitas layanan kefarmasian di apotek [5].

Penelitian yang menunjukkan bahwa sistem informasi berbasis *website* mampu meningkatkan efisiensi dan akurasi dalam pengelolaan stok obat. Fitri et al. (2023), misalnya, mengembangkan sistem *REDASELESE* untuk Apotek Semoga Lekas Sembuh, yang memudahkan pencatatan stok dan menjaga sinkronisasi data antara *admin*, pelanggan, dan pemilik apotek [2].

Penelitian yang dilakukan oleh Marfani dan Utomo (2023) berjudul “Rancang Bangun Sistem Informasi Apotek Berbasis *Web* Menggunakan *Laravel*” juga menunjukkan adanya permasalahan pada sistem pencatatan stok obat secara manual di Apotek Wahyu Farma. Penggunaan buku atau kertas dalam pencatatan transaksi dinilai tidak efisien dan rawan kesalahan. Sebagai solusi, mereka membangun sistem pengelolaan stok berbasis *website* dengan *Framework Laravel* untuk meningkatkan akurasi dan efisiensi operasional apotek [3].

Permasalahan pengelolaan stok obat pada apotek yang memiliki lebih

dari satu cabang juga menjadi perhatian dalam penelitian oleh Widhiartini et al. (2025), yang berjudul “Pengelolaan Pengadaan Logistik Farmasi pada Empat Apotek Jaringan di Wilayah Denpasar dan Badung”. Penelitian tersebut mengidentifikasi kendala seperti ketidakseimbangan stok antar cabang, keterlambatan pengadaan, dan rendahnya kepatuhan terhadap regulasi. Studi ini menegaskan bahwa pemanfaatan sistem digital dapat meningkatkan efisiensi dan akurasi dalam pengelolaan stok di apotek jaringan [4].

2.2 Landasan Teori

Landasan teori merupakan konsep, teori, prinsip dan pendapat yang mendukung proses pengembangan sistem dan sumber daya yang digunakan dalam pengembangan sistem yang dibuat. Berikut dijelaskan teori-teori yang menjadi dasar atau pedoman dalam Sistem Informasi Pengelolaan Stok Obat Apotek Saditan, diantaranya adalah pengolah data yaitu *PHPMyAdmin*, *MYSQL*, *Visual Studio Code* dan lain-lain.

2.2.1 Website

Website merupakan kumpulan komponen yang terdiri dari teks, gambar, suara animasi, sehingga merupakan media informasi yang menarik dan sangat diminati untuk dipergunakan sebagai media berbagi informasi [6].

2.2.2 MYSQL

MySQL, merupakan aplikasi *database server*.

Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. *MySQL* dapat digunakan untuk membuat dan mengelola *database* beserta isinya. *Mysql* dapat dimanfaatkan untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database* [7].



Gambar 2. 1. Logo *Mysql*

2.2.3 Basis Data

basis data adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Basis data meliputi spesifikasi dari tipe data, struktur, dan batasan dari data atau informasi yang akan disimpan. Basis data merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user* [8].

2.2.4 *PHPMysqlAdmin*

PHPMysqlAdmin adalah sebuah aplikasi atau perangkat lunak bebas (*open source*) yang ditulis dalam bahasa pemrograman PHP

yang digunakan untuk menangani administrasi *database MySQL* melalui jaringan lokal maupun internet. Perbedaan *PhpMyAdmin* dengan *MySQL* terletak pada fungsi. *PhpMyAdmin* merupakan alat untuk memudahkan dalam mengoperasikan *database MySQL*, sedangkan *MySQL* adalah *database* tempat penyimpanan data.



Gambar 2. 2. Logo *PHPMYAdmin*

2.2.5 *Node Js*

Node.js adalah *runtime JavaScript* yang berbasis pada mesin V8 milik *Google Chrome*. Dimaksudkan untuk menjalankan kode *JavaScript* di sisi *server*, memungkinkan pengembang membangun aplikasi *server* yang cepat, ringan, dan efisien.

2.2.6 *Laravel*

Laravel adalah suatu kerangka kerja sumber terbuka yang menyediakan sejumlah alat dan arsitektur aplikasi, termasuk *bundle*, migrasi, dan antarmuka baris perintah (CLI) yang dikenal sebagai *Artisan*. Kerangka kerja ini menggabungkan fitur terbaik dari berbagai kerangka kerja seperti *Codeigniter*, *Yii*, *ASP.NET MVC*, *Ruby on Rails*, *Sinatra*, dan lainnya. Kelebihan *Laravel* terletak pada performa

yang lebih cepat, stabilitas *reload* data, keamanan data, pemanfaatan fitur canggih seperti *Blade* dengan konsep *HMVC* (*Hierarchical Model View Controller*), ketersediaan *library* siap pakai, dan fitur pengelolaan migrasi untuk pembuatan skema tabel dalam basis data [9].



Gambar 2. 3 Logo *Laravel*

2.2.7 *Cascading Style Sheets* (CSS)

Cascading Style Sheets(CSS) adalah bahasa yang dirancang untuk menambahkan gaya visual, seperti jenis huruf, warna, dan tata letak, pada halaman *web* [10]. Dengan CSS dapat membuat halaman *web* lebih menarik secara visual dan mudah dibaca bagi pengguna. penulisan kode CSS biasanya dilakukan dengan cara membuat *style sheet* terpisah dalam *file* CSS atau memasukkan CSS tersebut langsung ke dalam *tag* HTML menggunakan atribut *style*.



Gambar 2. 4. Logo CSS

2.2.8 HTML

HTML (*Hypertext Markup Language*) adalah bahasa markup

standar yang digunakan untuk membuat dan merancang halaman *web*. *HTML* digunakan untuk mengatur struktur dan konten halaman *web*, seperti memformat bagian teks, gambar, video, tabel, *audio*, dan elemen lainnya. Penulisan kode *HTML* biasanya dilakukan dengan cara mengetikkan *tag-tag HTML* pada *text editor* [11].



Gambar 2. 5 Logo *HTML*

2.2.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah alat pemodelan yang banyak digunakan dalam pengembangan sistem berbasis perangkat lunak. UML membantu menggambarkan spesifikasi, struktur, dan perilaku sistem dengan menggunakan diagram-diagram tertentu [12]. UML memvisualisasikan konstruksi sistem dalam beberapa diagram yang sering digunakan yaitu diagram *use case*, *activity diagram*, dan *class diagram*.

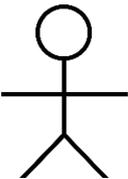
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

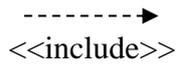
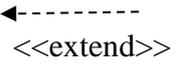
1. Use Case Diagram

Use case diagram adalah sebuah alat visual yang digunakan untuk menggambarkan interaksi antara sistem komputer dengan user

atau pengguna. Dalam use case digambarkan sebuah skenario penggunaan yang menggambarkan tindakan atau aktivitas yang dilakukan oleh pengguna. Simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2. 1. *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Actor/Aktor</i>	<i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. aktor berinteraksi.

No	Simbol	Nama	Keterangan
3.		<i>Association/ Assosiasi</i>	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4.		<i>Generalization/ Generalisasi</i>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk Mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
5.		<i>Include/Uses</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
6.		<i>Extend/Ekstensi</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan atau memodelkan berbagai aktivitas aliran dalam sistem yang sedang dikembangkan, serta bagaimana setiap aliran berawal dari keputusan yang mungkin terjadi, dan bagaimana setiap aktivitas berakhir. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

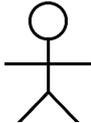
Tabel 2. 2. *Activity Diagram*

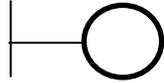
No	Simbol	Pengertian	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Activity Final Mode</i>	bagaimana objek dibentuk dan dihancurkan.
3.		<i>Initial Node</i>	bagaimana objek dibentuk atau diawali.
4.		<i>Fork Node</i>	Satu aliran yang tahap tertentu berubah menjadi beberapa aliran.

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan interaksi antar objek berupa pesan (*message*) yang digambarkan terhadap waktu. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 2. 3. *Sequence Diagram*

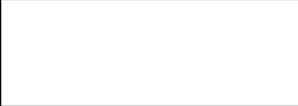
No.	Simbol	Pengertian	Keterangan
1.		<i>Aktor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		<i>Entity Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan pengiriman pesan.
4.		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

No.	Simbol	Pengertian	Keterangan
5.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem.
6.		<i>Focus Of Control & A Life Line</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

4. Class Diagram

Class Diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* menjelaskan basis data yang digunakan dari segi pendefinisian tabel apa saja yang digunakan dalam pembuatan pengembangan sistem.

Tabel 2. 4. *Class Diagram*

No	Simbol	Deskripsi
1		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
2		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
3		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
4		Relasi antar kelas dengan Makna keberuntungan antar kelas
5		Relasi antar kelas dengan makna <i>whole-part</i>

2.2.10 Visual Studio Code

Visual Studio Code adalah aplikasi *code editor* buatan *Microsoft* yang dapat dijalankan di semua perangkat desktop secara gratis. Kelengkapan fitur dan ekstensi membuat *code editor* ini menjadi pilihan utama para pengembang. *Visual Studio Code* bahkan mendukung hampir semua sistem operasi seperti *Windows*, *Mac OS*, *Linux*, dan lain sebagainya.

Visual Studio Code dapat digunakan untuk berbagai bahasa pemrograman seperti JavaScript, HTML, CSS, PHP, Python, C++, dan masih banyak lagi. Selain itu, *Visual Studio Code* menyediakan fitur *Live Share* yang memungkinkan beberapa pengembang bekerja pada satu proyek yang sama secara bersamaan dari lokasi yang berbeda [13].