

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut beberapa penelitian yang dijadikan sebagai sumber referensi yang berhubungan dengan Laporan Tugas Akhir ini adalah :

Penelitian yang dilakukan oleh Kusumawati (2020) membahas pengembangan sistem antrian pelayanan publik berbasis *website* sebagai solusi atas permasalahan pelayanan yang masih dilakukan secara manual di instansi pemerintahan. Permasalahan yang dihadapi meliputi penumpukan masyarakat, keterbatasan ruang tunggu, serta pencatatan antrian yang masih menggunakan metode konvensional sehingga menyebabkan ketidakteraturan dan lamanya waktu tunggu pelayanan. Dalam penelitian tersebut, sistem antrian dikembangkan menggunakan metode waterfall dengan menerapkan konsep *First In First Out* (FIFO) untuk memastikan urutan pelayanan berjalan secara adil dan terstruktur. Sistem ini menyediakan fitur pengambilan nomor antrian secara online, informasi nomor antrian yang sedang dilayani, serta pengelolaan antrian oleh petugas secara digital. Hasil penelitian menunjukkan bahwa penerapan sistem antrian berbasis *website* mampu meningkatkan efisiensi dan ketertiban proses pelayanan, mengurangi penumpukan antrian, serta memberikan kenyamanan yang lebih baik bagi masyarakat dibandingkan dengan sistem manual[4].

Penelitian yang dilakukan oleh Raharjo dan Aji (2024) membahas pengembangan sistem informasi antrian online pada instansi pelayanan publik, khususnya di Dinas Kependudukan dan Pencatatan Sipil (Dukcapil). Permasalahan yang dihadapi meliputi sistem antrian yang masih dilakukan secara konvensional, sehingga menyebabkan antrian panjang, ketidakteraturan pelayanan, serta kesulitan masyarakat dalam memantau urutan antrian. Selain itu, sistem manual juga dinilai kurang efisien dan berpotensi menimbulkan ketidaktransparanan dalam proses pelayanan. Dalam penelitian tersebut, sistem dikembangkan menggunakan pendekatan *User Centered Design (UCD)* yang berfokus pada kebutuhan pengguna, khususnya masyarakat sebagai penerima layanan. Sistem ini menyediakan fitur pengambilan nomor antrian secara online, pemantauan antrian secara *real-time*, serta pengelolaan data antrian secara terkomputerisasi. Hasil penelitian menunjukkan bahwa sistem antrian online yang dikembangkan mampu meningkatkan efisiensi dan transparansi pelayanan, mempermudah masyarakat dalam mengakses informasi antrian, serta mengurangi waktu tunggu dalam proses pelayanan publik[5].

Penelitian yang dilakukan oleh Handayani (2025) membahas pengembangan sistem antrian online sebagai solusi terhadap permasalahan antrian manual yang masih banyak digunakan pada instansi pelayanan publik. Permasalahan yang diangkat meliputi antrian yang tidak terorganisir, penumpukan masyarakat di area pelayanan, serta ketidakpastian waktu tunggu yang berdampak pada menurunnya kualitas pelayanan. Dalam

penelitian tersebut, sistem antrian dikembangkan secara terkomputerisasi dan dilengkapi dengan fitur notifikasi *real-time* yang memungkinkan pengguna memperoleh informasi pemanggilan antrian secara langsung tanpa harus menunggu di lokasi pelayanan. Selain itu, sistem juga memberikan kemudahan bagi masyarakat untuk melakukan pendaftaran antrian secara online serta memantau progres antrian melalui perangkat yang digunakan. Hasil penelitian menunjukkan bahwa penerapan sistem antrian online dengan notifikasi *real-time* mampu meningkatkan efisiensi waktu pelayanan, mengurangi kepadatan antrian di lokasi, serta memberikan kenyamanan dan kepuasan yang lebih baik bagi masyarakat dalam memperoleh layanan publik[6].

2.2 Landasan Teori

2.2.1 *Waterfall*

Metode penelitian *waterfall* adalah konsep pengembangan yang menekankan pada langkah sistematis. Sehingga, proses penciptaan sebuah sistem harus dilakukan secara berurutan, mulai dari tahapan identifikasi kebutuhan sampai ke proses perawatan.

2.2.2 Antrian

Antrian merupakan suatu kondisi di mana pengguna layanan harus menunggu untuk memperoleh pelayanan karena keterbatasan fasilitas atau petugas pelayanan. Pengelolaan antrian diperlukan agar proses pelayanan dapat berjalan secara tertib, teratur, dan efisien.

2.2.3 Website

Website disebut juga *site*, situs, situs *web* atau portal. Merupakan kumpulan halaman *web* yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah *home page*, sedangkan halaman demi halamannya secara mandiri disebut *web page*, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. *Website* adalah situs yang dapat diakses dan dilihat oleh para pengguna *Internet*. Pengguna *Internet* semakin hari semakin bertambah banyak, sehingga hal ini adalah potensi besar yang berkembang terus[7].

2.2.4 Laravel



Gambar 2.1 *Laravel*

Sumber : <https://www.stickpng.com/id/img/agama/ikon-logo-emoji/perusahaan-teknologi/logo-hitam-laravel>

Laravel adalah *framework PHP* berlisensi *MIT* yang mengadopsi konsep *Model-View-Controller (MVC)*. *Framework* ini dirancang untuk pengembangan aplikasi berbasis *MVP* menggunakan *PHP*, dengan tujuan meningkatkan kualitas perangkat lunak. *Laravel* membantu mengurangi biaya awal pengembangan dan pemeliharaan,

sekaligus memberikan pengalaman kerja yang lebih efisien melalui sintaks yang jelas, ekspresif, dan hemat waktu[8].

2.2.5 *Visual Studio Code*



Gambar 2.2 *Visual Studio Code*

Sumber : <https://www.stickpng.com/id/img/agama/ikon-logo-emoji/liburan/objek/natal/logo-lengkap-visual-studio-code>

Visual Studio Code (VS Code) adalah editor teks multiplatform yang lengkap dan andal, dikembangkan oleh *Microsoft*. Selain tersedia untuk *Windows*, *VS Code* juga mendukung *Linux* dan *macOs*. Editor ini kompatibel dengan berbagai bahasa pemrograman, seperti *JavaScript*, *TypeScript*, dan *Node.js*[9].

2.2.6 *Bootstrap*



Gambar 2.3 *Bootstrap*

Sumber : <https://getbootstrap.com/docs/5.0>

HTML, *CSS*, dan *JavaScript* yang dirancang untuk mempercepat proses pengembangan *web responsive* dan *mobile first*

(memprioritaskan perangkat seluler). Selain bisa digunakan untuk mengembangkan *website* dengan lebih cepat, *Bootstrap* juga merupakan *framework* gratis yang bersifat *open-source*. Skrip dan *syntax* yang disediakan *Bootstrap* bisa diterapkan untuk berbagai komponen dalam desain *web*.

2.2.7 Cascading Style Sheet (CSS)



Gambar 2.4 Cascading Style Sheet (CSS)

Sumber : <https://pngdownload.io/png-image/css-logo-png-symbol-for-web-development-transparent>

CSS merupakan bahasa pemrograman dalam membuat *web* yang digunakan untuk komponen dalam sebuah *web* agar lebih terlihat terstruktur dan lebih menarik. Cara CSS bekerja adalah dengan memodifikasi *HTML* dengan memilih komponen dalam *HTML* kemudian diatur dan diberikan property sesuai dengan tampilan yang diinginkan. Dalam memberikan aturan dalam komponen *HTML*, CSS terdiri dari 3 bagian skrip yaitu *selector* untuk memilih komponen yang diberikan aturan, *property* adalah aturan yang diberikan dan *value* sebagai nilai dari aturan yang diberikan[10].

2.2.8 HTML



Gambar 2.5 HTML

Sumber : <https://pngdownload.io/png-image/html5-logo-png-symbol-for-web-development-transparent/>

HTML adalah kependekan dari *Hypertext Markup Language* yang merupakan sebuah bahasa *markup*. *HTML* adalah kode untuk membuat struktur halaman suatu *website* yang menarik, saling terhubung satu dengan yang lainnya, dan yang pasti dapat diakses melalui *internet*.

Menurut Anhar (2010:40), *HTML* adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada *web browser*. Tag-tag *HTML* selalu diawali dengan `<` dan diakhiri dengan `>` dimana `x` tag *HTML* itu seperti `b`, `i`, `u` dll.

Menurut Ardhana (2012:42), *HTML* merupakan suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video[11].

2.2.9 Laragon



Gambar 2.6 Laragon

Sumber : <https://copysymbol.org/developer/apa-itu-laragon-pengenalan-dan-manfaatnya-untuk-pengembangan-web/>

Laragon adalah *software* yang berfungsi untuk membuat *server* lokal di komputer, yang sangat membantu pengembang dalam mengelola proyek *web* mereka dengan lebih efisien. *Laragon* mendukung berbagai teknologi seperti *PHP*, *MySQL*, *Apache*, *Nginx* dan lainnya, serta menyediakan lingkungan pengembangan yang stabil, cepat dan mudah diakses[12].

2.2.10 MySQL



Gambar 2.7 MySQL

Sumber : https://wikitech.wikimedia.org/wiki/File:Mysql_logo.png

MySQL adalah salah satu jenis *database server* yang sangat populer. Kepopulerannya disebabkan oleh pengguna *SQL* sebagai bahasa utama untuk mengakses database. *MySQL* bersifat *open-source*

dan dilengkapi dengan *source code* (kode yang digunakan untuk mengembangkan *MYSQL*)[13].

2.2.11 Database

Database adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap *database* mempunyai *API* tertentu untuk membuat, mengakses, mengatur, mencari, menyalin data yang ada di dalamnya. *Database* yaitu kumpulan file-file yang berhubungan satu dengan yang lainnya, diatur sedemikian rupa sehingga dapat digunakan oleh beberapa program aplikasi *database*[14].

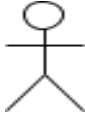
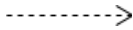
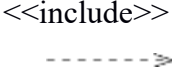
2.2.12 UML (Unified Modeling Language)

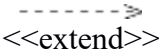
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek[15]. *UML* tersusun atas sejumlah elemen grafis membentuk 9 diagram-diagram. Dalam penelitian ini melakukan desain hanya 4 diagram yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. Berikut pengertian dan bentuk simbol-simbol dari diagram:

a. *Usecase diagram*

Usecase diagram menggambarkan hubungan antara pengguna dengan sistem. Diagram ini menunjukkan aktivitas yang dapat dilakukan oleh setiap aktor admin dan *user*[16]. Berikut simbol- simbol *usecase*:

Tabel 2.1 Simbol *Usecase diagram*

No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses atau sistem yang berinteraksi dengan sistem lain.
2		<i>Dependency</i>	Hubungan perubahan yang terjadi pada suatu elemen mandiri
3		<i>Generalization</i>	Hubungan generalisasi (umum-khusus) antar dua buah <i>use case</i> .
4		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan

No.	Gambar	Nama	Keterangan
			dapat berdiri sendiri.
5	 <<extend>>	<i>Extend</i>	Relasi tambahan yang Menunjukkan <i>use case</i> lain memerlukan <i>use case</i> . syarat untuk dijalankan atau agar fungsinya berjalan dengan baik.
6		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki

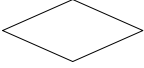
No.	Gambar	Nama	Keterangan
			interaksi dengan aktor.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan Sistem.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari

No.	Gambar	Nama	Keterangan
			jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya Komputasi.

b. *Activity diagram*

Activity diagram menggambarkan berbagai alur aktivitas atau proses kerja pada sistem. Melalui *activity diagram*, alur sistem dapat dipahami dengan lebih jelas karena menggambarkan proses yang dilakukan oleh *user*, *admin*, dan sistem secara terstruktur[17]. Berikut simbol-simbol dari *activity diagram*:

Tabel 2.2 Simbol *Activity diagram*

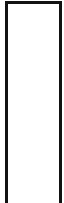
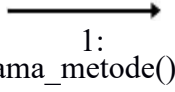
No.	Simbol	Nama	Penjelasan
1		<i>Activity</i>	Memperlihatkan bagaimana kelas saling berinteraksi.
2		<i>Decision</i>	Asosiasi percabangan jika ada pilihan yang lebih dari satu.
3		<i>Initial Node</i>	Bagaimana objek diawali. Hanya ada satu initial dalam satu diagram.
4		<i>Activity Final Node</i>	Bagaimana objek diakhiri. Dalam satu diagram minimal ada satu final state.

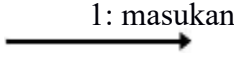
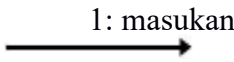
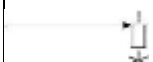
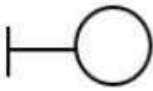
No.	Simbol	Nama	Penjelasan
5		<i>Fork Node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran
6		<i>Control Flow</i>	Menunjukkan urutan aliran aktivitas.
7		<i>Partition</i>	Simbol yang membatasi aktivitas antar orang, organisasi, sistem

c. *Sequence diagram*

Sequence Diagram digunakan untuk menggambarkan urutan interaksi antara pengguna dan sistem. Diagram ini menunjukkan aliran pesan atau proses yang terjadi dari awal hingga akhir sesuai fungsi sistem yang dijalankan. Diagram ini membantu memahami urutan proses dan komunikasi antar objek dalam sistem secara lebih detail[18]. Berikut simbol-simbol dari *sequence diagram*:

Tabel 2.3 Simbol *Sequence diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		<i>Object</i>	Menyatakan objek yang berinteraksi pesan.
4		<i>Timelife</i>	Menyatakan objek dalam berinteraksi pesan.
5		Pesan tipe <i>Create</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi.
6		Pesan tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi/metode yang

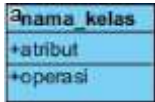
No	Gambar	Nama	Keterangan
			ada pada objek lain atau dirinya sendiri.
7		Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain.
8		Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain.
9		Pesan tipe <i>Destroy</i>	Menyatakan suatu objek telah menjalankan suatu operasi.
10		Boundar y Class	Berupa tepi dari sistem, seperti <i>user interface</i> atau alat yang berinteraksi dengan

No	Gambar	Nama	Keterangan
			sistem lain.
11		Control Class	Mengatur aliran dari informasi untuk sebuah skenario.

d. *Class diagram*

Class Diagram digunakan untuk menggambarkan struktur kelas, atribut, metode, serta hubungan antar kelas pada sistem. Diagram ini membantu dalam memahami struktur data dan rancangan sistem yang akan dibangun. Diagram ini digunakan sebagai acuan dalam proses pembuatan *database* dan pengembangan program agar sistem berjalan sesuai dengan kebutuhan[19]. Berikut simbol-simbol dari *class diagram*:

Tabel 2.4 Simbol *Class diagram*

No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem.
2		Antarmuka/ <i>Interface</i>	Sama dengan konsepn interface dalam pemrograman berorientasi objek.
3		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum.
4		Asosiasi berarah/ <i>Directed Association</i>	Relasi antar kelas dengan makna kelas.
5		Generalisasi	Relasi antar kelas dengan makna generalisasi.

No.	Gambar	Nama	Keterangan
6		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna <i>whole- part</i> .