

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Sistem informasi berbasis web telah banyak dikembangkan oleh peneliti maupun pengembang sebelumnya. Tujuan dari sistem ini umumnya adalah untuk meningkatkan efisiensi layanan serta memberikan kemudahan akses informasi bagi pengguna. Dalam konteks layanan jaringan, pengembangan sistem informasi Wi-Fi berbasis web bertujuan agar pengelolaan jaringan, pemantauan pengguna, dan penyampaian informasi dapat dilakukan secara lebih terstruktur dan *real-time*. Penelitian terkait ini menjadi landasan dalam perancangan sistem yang dibangun supaya lebih relevan dan sesuai dengan kebutuhan pengguna. Penelitian oleh Yanto et al. (2021) juga merancang sistem informasi berbasis web untuk pemasangan dan pengaduan gangguan Wi-Fi di Telkom Sungai Dareh yang bertujuan meningkatkan pelayanan serta efisiensi pengaduan pelanggan.[9].

Berdasarkan penelitian oleh Tri Adi *et al.* (2021) yang berjudul "*Pengembangan Aplikasi Manajemen Tagihan Internet Berbasis Web pada Penyedia Layanan RT/RW-Net*", penelitian ini membahas perancangan sistem informasi yang bertujuan untuk mempermudah proses penagihan pelanggan secara otomatis. Sistem dikembangkan menggunakan metode *waterfall* dan dibangun dengan bahasa pemrograman PHP serta basis data MySQL. Hasil penelitian Total waktu proses bisnis menggunakan sistem adalah 244.69 detik

sedangkan proses bisnis dengan cara manual memakan waktu 477.83 detik. Ini menunjukkan kenaikan efisiensi waktu secara keseluruhan sebesar 48.78%. Dari hasil tersebut, dapat disimpulkan bahwa menggunakan sistem lebih efisien dari segi waktu dibandingkan dengan mengelola tagihan dan data pelanggan secara manual. Penelitian ini relevan dengan yang penulis lakukan karena sama-sama berfokus pada pengembangan sistem informasi untuk pengelolaan tagihan di lingkungan penyedia layanan internet skala lokal [4].

Selanjutnya, penelitian oleh Setiawan dan Widiyastuti (2024) Penelitian ini mengimplementasikan jaringan RT/RW-Net menggunakan metode NDLC di wilayah RT 01 Sragen Permai. Hasil dari penelitian ini menunjukkan *bandwidth* yang diberikan kepada pelanggan sebesar 3 Mbps sampai 10 Mbps. Penelitian ini lebih menekankan pada aspek infrastruktur dan konfigurasi jaringan menggunakan perangkat Mikrotik, dengan fokus pada efisiensi *bandwidth* dan kemudahan *monitoring*. Meskipun fokusnya bukan pada aplikasi manajemen, namun penelitian ini relevan sebagai referensi teknis dalam mengatur jaringan yang mendasari sistem informasi pelanggan RT/RW-Net [10].

Penelitian oleh Pratama (2023) berjudul "*Sistem Informasi Manajemen Hotspot Berbasis Web Menggunakan Framework Laravel (Studi Kasus CV AzzahraNet)*" membahas pengembangan sistem informasi manajemen *hotspot* berbasis web yang dirancang untuk mengatasi permasalahan dalam pengelolaan layanan *hotspot* di CV.AzzahraNet. Sistem ini dikembangkan menggunakan framework Laravel dan dikombinasikan

dengan router yang terhubung melalui API. Hasil penelitian menunjukkan bahwa sistem ini dapat meningkatkan efisiensi dalam pengelolaan layanan *hotspot* serta mempermudah *monitoring* jaringan. Penelitian ini relevan dengan proyek yang penulis kembangkan karena sama-sama menggunakan Laravel untuk membangun sistem informasi layanan Wi-Fi yang terintegrasi [11].

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan dan bekerja sama untuk menjalankan suatu aktivitas guna menyelesaikan permasalahan yang memiliki manfaat atau nilai guna bagi pihak yang membutuhkan. Menurut Tysara (2021), sistem informasi tidak hanya mencakup aspek teknologi, tetapi juga melibatkan prosedur serta unsur manusia yang saling terhubung dalam proses pengolahan informasi. Sementara itu, menurut Erni Widiarti dan tim penulis dalam buku Pengantar Sistem Informasi, sistem informasi memiliki peran penting dalam memperluas serta mempermudah akses terhadap data, meningkatkan efektivitas perencanaan, serta mendorong produktivitas dalam pengembangan dan pemeliharaan sistem [12].

2.2.2 Jaringan Komputer

Jaringan komputer adalah sekumpulan dua atau lebih perangkat yang saling terhubung untuk saling berbagi data, sumber

daya, dan layanan melalui media kabel (*wired*) maupun nirkabel (*wireless*) (*Buku Ajar Jaringan Komputer*). Komponen jaringan umumnya terdiri dari dua bagian, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras meliputi server, router, switch, hub, dan modem yang berperan dalam proses pengiriman serta pengelolaan data. Sementara itu, perangkat lunak meliputi sistem operasi jaringan, perangkat manajemen seperti SNMP, dan protokol komunikasi seperti TCP/IP [13].

2.2.3 Wi-Fi dan Hotspot

Menurut M. Agus J. Alam (2019), Wi-Fi atau *Wireless Fidelity* adalah kumpulan standar yang digunakan dalam jaringan lokal nirkabel (*Wireless Local Area Network* – WLAN). Sedangkan *hotspot* merupakan area jangkauan sinyal yang dipancarkan oleh access point sebagai titik akses jaringan nirkabel [13]. Cara kerja Wi-Fi diawali dengan perangkat yang mencari SSID (*Service Set Identifier*), kemudian melakukan otentikasi menggunakan protokol keamanan seperti WPA atau WPA2 untuk menjamin koneksi tetap aman. Teknologi ini memberikan banyak manfaat, seperti memudahkan akses internet di area publik maupun privat, mendukung mobilitas pengguna, serta memungkinkan berbinterinteragai perangkat untuk berbagi koneksi secara bersamaan [14].

2.2.4 *Laravel Framework*

Laravel adalah salah satu *framework* PHP yang paling populer dan banyak digunakan oleh pengembang web di seluruh dunia. Laravel pertama kali dirilis pada tahun 2011 oleh Taylor Otwell, seorang pengembang asal Amerika Serikat.

Framework ini menawarkan kemudahan penggunaan dan berbagai fitur lengkap yang mendukung percepatan pengembangan aplikasi web, seperti *routing*, *controller*, *view*, *model*, *basis data*, *Eloquent ORM*, *middleware*, *autentikasi*, *testing*, dan *debugging* [15].

2.2.5 *Model Waterfall*

Model *Waterfall* didefinisikan sebagai rangkaian aktivitas proyek dalam *System Development Life Cycle* (SDLC) yang dilakukan secara linier dan berurutan (R. Kurniawan, 2023). Setiap tahapannya bergantung pada hasil dan dokumentasi dari tahap sebelumnya. Model ini memiliki keunggulan dalam hal struktur yang jelas, pembagian tugas yang sistematis, dan *deliverables* yang terukur. Karena sifatnya yang terstruktur, *Waterfall* digolongkan sebagai *predictive model*, di mana waktu dan sumber daya dapat diperkirakan dengan lebih akurat sejak awal proyek. Dokumentasi yang dihasilkan pun lengkap dan konsisten, sehingga dapat menjadi referensi bersama antar tim pengembang. Namun demikian, model ini juga memiliki kelemahan, yaitu kurang fleksibel terhadap perubahan kebutuhan sistem ketika proses pengembangan sudah berjalan. Hal ini menjadi tantangan,

terutama dalam proyek-proyek berskala menengah hingga besar, karena kesalahan pada tahap awal seperti *requirement* atau desain baru dapat teridentifikasi di tahap akhir, yang membuat proses revisi menjadi lebih sulit dan mahal [16].

2.2.6 UML (*Unified Modeling Language*)

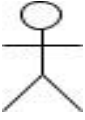

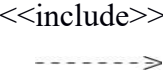
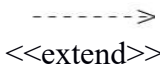


UML, atau *Unified Modeling Language*, adalah bahasa yang digunakan untuk menggambarkan desain perangkat lunak. karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma object oriented. UML juga memberikan standar penulisan sebuah sistem blueprint, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem software. UML terdiri dari beberapa diagram, yaitu *use case diagram*, *class diagram*, *state diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram* [17].



Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. *Use Case Diagram*: merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor yang merupakan sebuah gambaran entitas dari

manusia atau sebuah sistem yang melakukan pekerjaan di sistem.


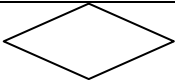
Tabel 2.1 Simbol *Usecase Diagram*





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 dapat berdiri sendiri.
5		<i>Extend</i>	Relasi tambahan yang Menunjukkan <i>use case</i> lain memerlukan <i>use case</i> . syarat untuk dijalankan atau agar fungsinya dapat 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 interaksi dengan aktor.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

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

2. *Activity Diagram*: Sebuah diagram aktivitas UML menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem melalui aliran kontrol antara aksi yang dilakukan sistem [18].

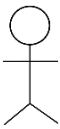




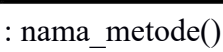
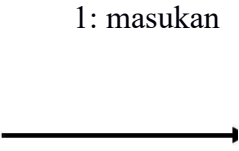
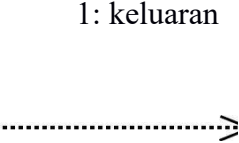
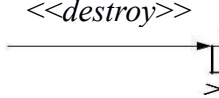
Tabel 2.2 Simbol *Activity Diagram*

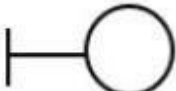

No.	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana kelas saling berinteraksi
2		<i>Decision</i>	Asosiasi percabangan jika ada pilihan yang lebih dari satu

No.	Gambar	Nama	Keterangan
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
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 atau kelompok

3. *Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek serta pesan yang dikirimkan dan diterima antar objek. Gambaran *sequence diagram* dibuat minimal sebanyak pendefinisian use case yang memiliki proses tersendiri atau yang penting. Semua *use case* yang telah didefinisikan harus mencakup interaksi jalannya pesan pada *sequence diagram*. Oleh karena itu, semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga akan semakin banyak [19].

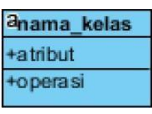


Tabel 2. 3 Simbol *Sequence Diagram*





No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang , proses, atau sistem lain yang berinteraksi dengan system 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 ada pada objek lain atau dirinya sendiri
7		Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/infor masi ke objek lain
8		Pesan tipe <i>Return</i>	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu
9		Pesan tipe <i>Destroy</i>	Menyatakan suatu objek telah

No.	Gambar	Nama	Keterangan
			menjalankan suatu operasi
10		Boundary Class	Berupaan tepi dari sistem, seperti user <i>interface</i> atau alat yang berinteraksi dengan sistem lain
11		Control Class	Mengatur aliran dari informasi untuk sebuah skenario

4. *Class Diagram* digunakan untuk memodelkan kelas-kelas dalam sistem, atributnya, metodenya, dan hubungan antar kelas. Penerapan class diagram dalam perancangan sistem informasi perpustakaan diharapkan dapat menghasilkan model sistem yang jelas, terstruktur, dan mudah dipahami. Model sistem yang baik akan menjadi dasar yang kuat untuk pengembangan dan implementasi sistem informasi perpustakaan yang efektif [20].

Tabel 2. 4.Simbol *Class Diagram*

No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem
2		Antarmuka/ <i>Interface</i>	Sama dengan konsepn <i>interface</i> 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
6		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna whole- part

2.2.7 Basis Data (*database*)

Menurut Ahmad Tabrani (2024), basis data (*database*) adalah kumpulan informasi yang terstruktur dan disimpan dalam sistem komputer sehingga dapat dikelola dan diakses dengan mudah. Istilah “basis data” berasal dari kata “basis” yang berarti tempat penyimpanan atau gudang, dan “data” yang mencakup semua bentuk representasi dari objek-objek di dunia nyata. Keuntungan penggunaan basis data meliputi kemudahan akses, efisiensi biaya, keamanan informasi, dan integrasi data yang lebih baik. Hal ini membuatnya menjadi solusi efektif untuk pengelolaan informasi baik dalam skala organisasi kecil maupun besar. Dalam proses perancangannya, digunakan model konseptual data berupa *Entity-Relationship Diagram* (ERD). ERD merupakan alat visual yang digunakan untuk merancang dan memodelkan struktur data dalam sistem informasi.

Tujuan utamanya adalah menyederhanakan pemahaman dan desain basis data dengan cara memetakan entitas-entitas dunia nyata ke dalam struktur yang sistematis dan terkelola. Selain itu, ERD juga menjadi sarana komunikasi yang penting antara pengembang dan pengguna sistem [18].