

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Beberapa penelitian sebelumnya menunjukkan bahwa penerapan sistem tiket layanan bantuan berbasis *web* mampu meningkatkan efektivitas pengelolaan keluhan layanan teknologi informasi. Sistem *ticketing* memungkinkan setiap permasalahan yang dilaporkan oleh pengguna tercatat secara sistematis, dipantau status penanganannya, serta terdokumentasi dengan baik sebagai bahan evaluasi layanan. Selain itu, penggunaan sistem berbasis *web* juga dinilai mampu memperbaiki alur komunikasi antara pengguna dan tim *IT Support* dibandingkan dengan metode pelaporan konvensional.

Temuan tersebut diperkuat oleh sejumlah penelitian yang secara khusus mengkaji perancangan dan implementasi *Helpdesk Ticketing System* berbasis *web*. Effendi dan Assegaff (2022) mengembangkan sistem *Helpdesk ticketing* pada lingkungan layanan TI perusahaan yang berfokus pada pengelolaan tiket secara terintegrasi. Hasil penelitian tersebut menunjukkan bahwa sistem yang dirancang mampu mempermudah proses pelaporan dan penanganan masalah TI secara terstruktur, sehingga meningkatkan efisiensi layanan internal organisasi [3].

Penelitian lain yang dilakukan oleh Zakaria dan Ardiansyah (2022) menerapkan sistem *Helpdesk ticketing* berbasis *web* dengan metode *Rapid Application Development* untuk menangani permasalahan pencatatan gangguan perangkat keras dan perangkat lunak. Penelitian ini menunjukkan bahwa penggunaan mekanisme *ticketing* dapat mempercepat respon layanan *IT Support* serta memudahkan pemantauan status penyelesaian keluhan secara *real-time* [4].

Selain itu, Christina dan Ziveria (2024) merancang aplikasi *support ticketing* berbasis *web* yang mendukung proses pembuatan tiket, pemantauan status, serta pendokumentasian permasalahan pengguna. Hasil penelitian tersebut menyimpulkan bahwa sistem *ticketing* berbasis *web* mampu meningkatkan efektivitas penanganan masalah serta menyediakan riwayat layanan yang dapat dimanfaatkan sebagai dasar evaluasi dan pengembangan sistem di masa mendatang [5].

Berdasarkan kajian terhadap penelitian-penelitian tersebut, dapat disimpulkan bahwa pengembangan sistem tiket layanan bantuan berbasis *web* merupakan pendekatan yang relevan dan telah terbukti efektif dalam mendukung kinerja tim *IT Support*. Oleh karena itu, penelitian ini difokuskan pada perancangan *Helpdesk Ticketing System* berbasis *web* yang disesuaikan dengan kebutuhan operasional tim *IT*

Support, sehingga diharapkan mampu memberikan kontribusi praktis dan akademis yang *optimal*.

Tabel 1.1 Perbandingan Penelitian Terdahulu

Nama Peneliti	Tahun Penelitian	Metode Pengembangan	Fitur Utama sistem	Teknologi yang digunakan
Effendi & Assegaf	2022	Prototype	Pengelolaan tiket gangguan SAP, pembagian tugas key <i>User</i> , laporan kinerja	Berbasis Web
Cristina & Ziveria	2024	Prototyping	Manajemen akun (<i>Admin, Helpdesk, Programmer, User</i>), riwayat tiket, lampiran gambar kendala	<i>Web</i> , MySQL, PHP, Visual Studio, UML
Zakaria & Ardiansyah	2022	<i>Rapid Application Development (RAD)</i>	Penomoran tiket otomatis (<i>request ticket</i>), pelaporan kendala	<i>Web</i> , PHP, MySQL

			hardware/software, cek status laporan	
--	--	--	--	--

Penelitian yang dilakukan oleh Effendi & Assegaf (2022), Christina & Ziveria (2024), serta Zakaria & Ardiansyah (2022) secara kolektif menegaskan bahwa transformasi sistem pelaporan dari konvensional ke berbasis digital sangat krusial untuk meningkatkan efisiensi operasional dan akuntabilitas dalam tata kelola layanan teknologi informasi. Namun studi-studi tersebut umumnya masih berfokus pada efisiensi pelayanan secara makro. Sejalan dengan hal tersebut, penelitian ini dikembangkan dengan orientasi yang lebih spesifik, yakni pada penguatan fungsionalitas sistem melalui pengembangan fitur-fitur teknis yang lebih komprehensif dan detail guna mengoptimalkan interaksi antara pengguna dan *IT Support*.

Berbeda dengan sistem *Helpdesk* standar, sistem ini mengedepankan beberapa fitur unggulan yang dirancang untuk mengoptimalkan interaksi dan akurasi penanganan kendala. Keunggulan tersebut mencakup mekanisme notifikasi otomatis ke *email Administrator*, manajemen antrean yang memprioritaskan laporan terbaru, serta fasilitas pelampiran dokumentasi multimedia (gambar dan video) untuk diagnosis masalah yang lebih presisi. Selain itu, sistem ini juga menyediakan fitur

komunikasi interaktif lanjutan (*follow-up chat*) guna menjamin penyelesaian masalah yang tuntas antara pengguna dan teknisi. Secara keseluruhan, penelitian ini berfokus pada pengembangan arsitektur fitur yang lebih komprehensif untuk menciptakan sistem *Helpdesk* yang lebih responsif dan informatif.

2.1 Landasan Teori

2.1.1 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan diaasipilin ilmu yang mempelajari metode, teknik, dan pendekatan sistematis dalam pengembangan perangkat lunak. Ruang lingkup rekayasa perangkat lunak mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan perangkat lunak. Tujuan utama rekayasa perangkat lunak adalah menghasilkan perangkat lunak yang berkualitas, dapat memenuhi kebutuhan pengguna, serta mudah dikembangkan dan dipelihara di masa mendatang [6].

2.1.2 Sistem Informasi

Rekayasa perangkat lunak merupakan diaasipilin ilmu yang mempelajari metode, teknik, dan pendekatan sistematis dalam pengembangan perangkat lunak. Ruang lingkup rekayasa perangkat lunak mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan perangkat lunak.

Tujuan utama rekayasa perangkat lunak adalah menghasilkan perangkat lunak yang berkualitas, dapat memenuhi kebutuhan pengguna, serta mudah dikembangkan dan dipelihara di masa mendatang. Rekayasa perangkat lunak merupakan disiplin ilmu yang mempelajari metode, teknik, dan pendekatan sistematis dalam pengembangan perangkat lunak. Ruang lingkup rekayasa perangkat lunak mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan perangkat lunak. Tujuan utama rekayasa perangkat lunak adalah menghasilkan perangkat lunak yang berkualitas, dapat memenuhi kebutuhan pengguna, serta mudah dikembangkan dan dipelihara di masa mendatang [7].

2.1.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan dan mendokumentasikan sistem perangkat lunak. UML menyediakan berbagai jenis diagram, seperti use case diagram, class diagram, *Sequence Diagram*, dan *Activity Diagram*, yang digunakan untuk menggambarkan struktur sistem serta interaksi antar komponen. Penggunaan UML membantu pengembang dalam memahami kebutuhan sistem secara menyeluruh sebelum proses implementasi dilakukan [8].

2.1.4 Basis Data

Basis data merupakan kumpulan data yang disimpan secara terorganisasi dan saling berelasi sehingga dapat diakses, dikelola, dan diperbarui dengan mudah. Basis data digunakan dalam sistem informasi untuk menjaga konsistensi, integritas, dan keamanan data. Dengan basis data, sistem mampu menyimpan informasi dalam jumlah besar dan mendukung proses pencarian serta pengolahan data secara efisien [9].

2.1.5 Helpdesk dalam Layanan Teknologi Informasi

Helpdesk merupakan layanan yang berfungsi sebagai pusat kontak antara pengguna dan penyedia layanan teknologi informasi. *Helpdesk* bertugas menerima laporan gangguan, mencatat permasalahan, serta mengoordinasikan proses penanganan masalah oleh tim teknis. Keberadaan *Helpdesk* membantu organisasi dalam menjaga kualitas layanan TI melalui pengelolaan keluhan yang terstruktur dan terdokumentasi [10].

2.1.6 Ticketing System

Ticketing system merupakan mekanisme pengelolaan layanan yang digunakan untuk mencatat dan melacak permintaan layanan atau gangguan dalam bentuk tiket. Setiap tiket memuat

informasi mengenai permasalahan, status penanganan, prioritas, serta riwayat penyelesaian. *Ticketing system* membantu tim teknis dalam mengatur alur kerja, memantau progres penyelesaian masalah, serta menyediakan dokumentasi layanan sebagai bahan evaluasi [11].

2.1.7 Metode FCFS

Metode *First Come First Served* (FCFS) merupakan teknik penjadwalan dalam sistem antrian yang mengutamakan proses berdasarkan urutan kedatangan, di mana proses pertama kali masuk akan dilayani terlebih dahulu hingga selesai. Algoritma ini memproses keadilan dan transparansi dalam pengelolaan antrian pengaduan atau job, seperti pada sistem *Helpdesk* di mana keluhan pertama diprioritaskan [12].

2.1.8 Metode Waterfall

Metode Waterfall merupakan kerangka pengembangan sistem yang bersifat linier dan sekuensial, dimulai dari pengumpulan kebutuhan, desain implementasi, hingga pengujian, tanpa kembali ke tahap sebelumnya. Pendekatan ini digunakan untuk memandu proses pengembangan sistem informasi layanan pengaduan server secara terstruktur, memastikan setiap tahap selesai sebelum lanjut ke tahap berikutnya. Metode ini efektif untuk proyek dengan kebutuhan

yang jelas dan stabil, seperti dalam pengembangan sistem monitoring siswa berbasis *web* [13].

2.1.9 PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman *scripting* sisi server yang banyak digunakan untuk pengembangan aplikasi *web* dinamis karena bersifat *Open source*, fleksibel, dan memiliki dukungan komunitas yang luas [14].

2.1.10 Laravel

Laravel merupakan *framework* PHP modern yang dirancang untuk mempermudah dan mempercepat pengembangan aplikasi *web* melalui struktur arsitektur yang teratur, seperti pola MVC, *routing*, *middleware*, dan ORM Eloquent yang memudahkan interaksi dengan basis data tanpa menulis perintah SQL secara manual [15].

Penggunaan Laravel dalam pengembangan aplikasi PHP telah terbukti meningkatkan efisiensi penulisan kode, mempermudah pemeliharaan sistem, serta mendukung pengembangan aplikasi yang lebih terstruktur dan dapat diskalakan [16].

2.2.11 MySQL

MySQL merupakan sistem manajemen basis data relasional yang bersifat *Open-source* dan banyak digunakan dalam pengembangan aplikasi berbasis web karena mampu mengelola data secara terstruktur, efisien, aman, dan konsisten. Penerapan MySQL dalam sistem informasi mendukung proses penyimpanan, pengelolaan, dan akses data secara terintegrasi, sehingga data seperti keluhan, riwayat penanganan, maupun informasi transaksi dapat tersusun dengan baik dan mudah diolah sesuai dengan kebutuhan sistem yang ada [17].

2.1.12 UI/UX

User interface (UI) dan *User experience (Ux)* merupakan dua aspek yang sangat penting dalam proses pengembangan suatu aplikasi karena berperan dalam membentuk kesan awal serta mempengaruhi kualitas interaksi pengguna dengan sistem. Oleh karena itu, perancangan UI dan UX perlu disesuaikan dengan kebutuhan pengguna agar aplikasi yang dikembangkan dapat digunakan secara efektif dan memberikan pengalaman yang baik. Salah satu komponen utama dalam perancangan tersebut adalah *User interface (UI)*, yaitu seluruh komponen antarmuka yang ditampilkan pada layer sebagai media interaksi antara pengguna dengan sistem. Melalui antarmuka tersebut, sistem dapat

menyampaikan informasi kepada pengguna sekaligus menerima masukan dari pengguna sehingga setiap fungsi pada aplikasi dapat dijalankan dengan baik [18].

Selain *User interface (UI)*, *User experience (Ux)* juga menjadi aspek yang tidak kalah penting dalam pengembangan suatu aplikasi. UX berkaitan dengan bagaimana pengguna dapat berinteraksi secara efektif dan nyaman Ketika menggunakan antarmuka yang telah dirancang. Penerapan UX bertujuan untuk meningkatkan kepuasan pengguna selama menggunakan aplikasi, baik pada platform website, perangkat bergerak (*mobile*), maupun desktop. Oleh karena itu, perancangan UX perlu memperhatikan kemudahan dan kenyamanan pengguna agar interaksi dengan sistem dapat berjalan secara lebih *optimal* [19].

2.2.13 Black-Box Testing

Black-Box Testing merupakan metode pengujian perangkat lunak yang dilakukan dengan menguji fungsi-fungsii sistem berdasarkan dengan spesifikasi kebutuhan tanpa melihat struktur maupun kode program yang ada di dalamnya. Pengujian difokuskan pada kesesuaian antara data masukan dan data keluaran yang dihasilkan sistem, sehingga setiap fitur dapat dipastikan berfungsi sesuai dengan yang diharapkan. Metode ini bertujuan untuk mendeteksi kesalahan pada semua aspek mulai dari aspek

fungsional, antarmuka, struktur data, performa, serta proses inisialisasi dan terminasi sistem. Dengan demikian, *Black-Box Testing* menjadi salah satu metode yang efektif untuk mengevaluasi kualitas perangkat lunak dari sudut pandang pengguna sebelum sistem diimplementasikan [20].