

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Rancang Bangun

Rancang bangun mencakup beberapa elemen penting, seperti perancangan basis data, perancangan antarmuka pengguna (user interface), perancangan arsitektur sistem (client-server, web-based, cloud), serta logika bisnis dan alur proses yang akan dijalankan sistem. Proses ini juga melibatkan pemilihan teknologi yang akan digunakan seperti bahasa pemrograman, framework, dan database. Semua elemen ini dirancang untuk memastikan sistem yang dibangun dapat berjalan secara optimal, efisien, dan mudah digunakan oleh pengguna akhir.

Ruang lingkup dalam proses rancang bangun sistem informasi mencakup keseluruhan aktivitas mulai dari identifikasi kebutuhan, analisis sistem, desain sistem, implementasi, pengujian, hingga dokumentasi dan pemeliharaan sistem. Jogiyanto (2021) menyatakan rancang bangun tidak hanya berfokus pada sisi teknis pembuatan aplikasi, tetapi juga mencakup pemahaman terhadap proses bisnis organisasi yang akan diotomatisasi., ruang lingkup rancang bangun tidak sebatas “membangun sistem”, melainkan bagaimana sistem tersebut menyatu dengan kebutuhan dan struktur organisasi.

A. Analisis perancangan

Analisis perancangan sistem merupakan proses sistematis dalam mengidentifikasi, memahami, dan merancang kebutuhan sistem informasi yang akan dikembangkan. Pratama (2020) menyatakan analisis sistem adalah tahapan awal dalam pengembangan sistem informasi yang bertujuan untuk memahami kebutuhan pengguna dan proses bisnis yang ada, sehingga dapat merancang sistem yang sesuai dengan kebutuhan tersebut. Proses ini melibatkan pengumpulan data, identifikasi masalah, dan penentuan solusi yang tepat untuk meningkatkan efisiensi dan efektivitas sistem yang akan dibangun.

Tahapan dalam analisis perancangan sistem mencakup beberapa

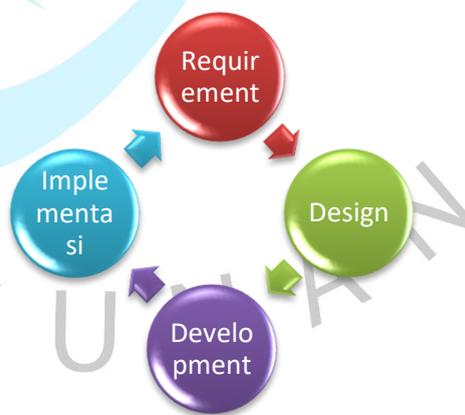
langkah penting, yaitu:

- 1) Identifikasi Masalah: Menentukan permasalahan yang ada dalam sistem saat ini.
- 2) Pengumpulan Data: Mengumpulkan informasi melalui wawancara, observasi, dan studi dokumentasi.
- 3) Analisis Kebutuhan: Menentukan kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan.
- 4) Perancangan Sistem: Membuat model sistem menggunakan alat bantu
- 5) Evaluasi dan Validasi: Memastikan bahwa desain sistem memenuhi kebutuhan pengguna dan dapat diimplementasikan dengan baik.

Menurut Utami (2024), tahapan-tahapan ini penting untuk memastikan bahwa sistem yang dikembangkan dapat memenuhi kebutuhan pengguna dan mendukung proses bisnis secara efektif.

B. System Development Life Cycle (SDLC)

Rukmana dkk. Mengungkapkan (2023) dalam buku berjudul *Pengantar Sistem Informasi: Panduan Praktis Pengenalan Sistem Informasi & Penerapannya*, SDLC terbagi ke dalam empat tahapan utama, yaitu perencanaan (planning), analisis (analysis), desain (design), dan implementasi (implementation).



Gambar 2.1 SDLC (Sumber : Rukmana, Dkk buku berjudul Pengantar Sistem Informasi(2023)

Tahap perencanaan, tim pengembang dan pengguna bersama-sama menentukan ruang lingkup sistem, tujuan utama dari aplikasi yang akan dikembangkan, serta menilai kelayakan dari sisi teknis maupun bisnis.

Tahap ini merupakan fondasi dari keseluruhan siklus SDLC karena menentukan arah dan batasan sistem.

Tahap analisis, di mana dilakukan identifikasi lebih dalam terhadap kebutuhan pengguna dan sistem yang sedang berjalan. Dalam tahap ini, dilakukan pengumpulan data melalui observasi, wawancara, dan dokumentasi. Tujuannya adalah untuk menggambarkan dengan jelas seperti apa sistem baru yang dibutuhkan oleh pengguna.

Tahap desain, yang bertujuan untuk menyusun rancangan sistem secara menyeluruh, baik dari sisi antarmuka pengguna (UI), alur proses sistem, hingga rancangan basis data. Desain ini menjadi acuan utama dalam proses pengkodean program nantinya.

Tahap implementasi, dilakukan penerapan desain yang telah dibuat ke dalam sistem nyata. Proses ini meliputi penulisan kode program, pengujian aplikasi, pelatihan pengguna, hingga instalasi sistem ke lingkungan operasional. Tahapan ini memastikan bahwa sistem benar-benar siap digunakan sesuai dengan kebutuhan pengguna.

C. Metodologi Pengembangan Sistem Informasi

Metodologi pengembangan sistem informasi adalah kerangka kerja atau pendekatan sistematis yang digunakan untuk merancang, mengembangkan, dan mengelola sistem informasi. Menurut Prabowo (2020)

Jenis-Jenis Metodologi Pengembangan Sistem Informasi

- 1) Waterfall: Pendekatan berurutan dan linear di mana setiap tahap harus diselesaikan sebelum melanjutkan. Sangat cocok untuk proyek yang memiliki persyaratan yang jelas dan tidak berubah.
- 2) Agile Development: Pendekatan iteratif dan inkremental yang mempromosikan kolaborasi antara tim pengembang dan pengguna. Agile memungkinkan fleksibilitas dan adaptasi terhadap perubahan selama proses pengembangan.
- 3) Rapid Application Development (RAD): Fokus pada pengembangan cepat dengan menggunakan prototipe dan keterlibatan pengguna secara intensif. Metodologi ini memungkinkan penyesuaian yang

cepat terhadap perubahan kebutuhan.

Menurut Chadijah dkk. (2023) dalam bukunya *Pengantar Sistem Informasi*, model *prototyping* merupakan pendekatan yang digunakan untuk memvisualisasikan sistem sebelum dibangun secara penuh, sehingga dapat meminimalisir kesalahan pemahaman antara pengguna dan pengembang. Dua jenis *prototyping* yang dijelaskan dalam buku ini adalah *Throwaway Prototyping* dan *Evolutionary Prototyping*. Pendekatan yang digunakan penulis termasuk dalam *evolutionary prototyping*, di mana desain awal dibangun dengan fungsi minimal, kemudian dievaluasi dan disempurnakan secara berulang berdasarkan umpan balik pengguna hingga sistem akhir terbentuk.

Berikut ini adalah tahapan RAD *Prototyping* yang diterapkan dalam penelitian ini, serta korelasinya dengan pengembangan aplikasi yang dilakukan penulis:

1. Tahap Perencanaan

Tahap ini merupakan langkah awal untuk menentukan ruang lingkup sistem serta mengidentifikasi kebutuhan dasar dari aplikasi yang akan dibangun. Dalam konteks penelitian ini, perencanaan dilakukan dengan cara memahami permasalahan utama di perusahaan, yaitu tidak adanya SOP pelaporan aduan, pelaporan yang tidak seragam, dan tidak adanya dokumentasi aduan yang terpusat.

2. Tahap Analisis

Tahap analisis dilakukan untuk menggali lebih dalam proses bisnis yang berjalan dan kebutuhan pengguna terhadap sistem yang akan dibangun. Pada tahap ini, penulis melakukan wawancara serta analisis proses bisnis yang ada saat ini, dan perumusan kebutuhan sistem baik secara fungsional maupun non-fungsional.

3. Tahap Workshop Desain

Tahap ini merupakan inti dari metode RAD berbasis *prototyping*. Pada tahap ini, penulis menyusun desain awal sistem berupa use case diagram dan mockup antarmuka, lalu dilakukan sesi diskusi atau

workshop desain bersama pengguna untuk mendapatkan masukan secara langsung. Tujuannya agar desain sistem benar-benar sesuai dengan kebutuhan pengguna di lapangan.

4. Tahap Desain

Setelah desain dari workshop pertama dan kedua disempurnakan, penulis menyusun desain teknis sistem dalam bentuk diagram.

5. Tahap Implementasi

Setelah desain disetujui dan final, penulis mulai membangun aplikasi menggunakan teknologi

Metode RAD sangat cocok digunakan dalam situasi-situasi berikut:

- a) Proyek dengan Batasan Waktu Ketat: Ketika proyek memerlukan penyelesaian dalam waktu singkat, RAD memungkinkan pengembangan yang cepat melalui prototipe dan iterasi.
- b) Kebutuhan yang Tidak Jelas atau Berubah-ubah: RAD memungkinkan penyesuaian cepat terhadap perubahan kebutuhan selama proses pengembangan.
- c) Keterlibatan Pengguna yang Tinggi: RAD menekankan kolaborasi antara pengembang dan pengguna, sehingga cocok ketika pengguna tersedia untuk terlibat secara aktif.
- d) Proyek dengan Sumber Daya yang Memadai: Karena RAD memerlukan tim yang berpengalaman dan sumber daya yang cukup, metode ini cocok ketika perusahaan memiliki kapasitas tersebut.

Hubungan Metodologi dengan SDLC dan Rancang Bangun Metodologi pengembangan sistem informasi berperan sebagai panduan dalam menerapkan tahapan-tahapan dalam System Development Life Cycle (SDLC). SDLC sendiri adalah kerangka kerja yang menggambarkan proses pengembangan sistem dari awal hingga akhir, termasuk perencanaan, analisis, desain, implementasi, dan pemeliharaan. Dengan

menggunakan metodologi yang sesuai, setiap tahap dalam SDLC dapat dilaksanakan secara efektif dan efisien.

D. Object-Oriented Analysis and Design OOAD

Menurut Gerardus Blokdyk dalam bukunya *Object Oriented Analysis and Design: A Complete Guide 2020 Edition*, OOAD membantu dalam mengidentifikasi peluang investasi dan tren yang muncul dalam pengembangan sistem berorientasi objek. Buku ini juga menekankan pentingnya proses yang dirancang oleh individu dengan perspektif kompleks untuk mengajukan pertanyaan yang tepat dan melihat masalah dari sudut pandang yang berbeda .

A. Tahapan OOAD

OOAD terdiri dari dua tahap utama:

1. Analisis Berorientasi Objek (Object-Oriented Analysis - OOA): Tahap ini fokus pada identifikasi kebutuhan fungsional sistem dan pemodelan domain masalah menggunakan objek. Tujuannya adalah memahami apa yang harus dilakukan oleh sistem tanpa memikirkan bagaimana cara melakukannya.
2. Desain Berorientasi Objek (Object-Oriented Design - OOD): Setelah analisis, tahap desain mengembangkan model analisis menjadi spesifikasi implementasi. Ini mencakup perancangan struktur kelas, antarmuka, dan interaksi antar objek untuk memenuhi kebutuhan yang telah diidentifikasi. Buku *Systems Analysis and Design: An Object-Oriented Approach with UML, 6th Edition* oleh Dennis, Wixom, dan Tegarden juga membahas pendekatan ini, menekankan pentingnya penggunaan UML dalam memodelkan sistem berorientasi objek .

E. Unified Modeling Language (UML)

Menurut (Muharni, 2021) menjelaskan bahwa diperlukan adanya terminologi untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Maka pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa

pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML juga memungkinkan untuk menggabungkan berbagai perspektif pengembangan perangkat lunak dalam satu model yang komprehensif, memudahkan pengembang dalam memvisualisasikan hubungan antar komponen sistem dan interaksi yang terjadi.

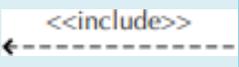
UML memiliki beberapa jenis diagram yang digunakan untuk menggambarkan berbagai aspek sistem. Berikut adalah beberapa diagram yang paling umum digunakan menurut para ahli di atas: meliputi:

a) Use Case Diagram

Use Case Diagram adalah salah satu komponen penting dalam Unified Modeling Language (UML) yang biasa digunakan dalam menggambarkan interaksi antara aktor dan sistem yang sedang dianalisis. Dalam konteks pengembangan perangkat lunak, diagram ini memiliki peran yang sangat penting untuk memvisualisasikan fungsionalitas sistem dari sudut pandang pengguna. Aktor dalam diagram ini dapat berupa pengguna manusia atau sistem lain yang berinteraksi dengan sistem, sementara "use cases" menggambarkan fungsi atau layanan yang disediakan oleh sistem untuk memenuhi kebutuhan aktor tersebut. Dengan demikian, Use Case Diagram memberikan gambaran yang jelas tentang apa yang dapat dilakukan oleh sistem dan bagaimana aktor berinteraksi dengan sistem tersebut.

Tabel 2.1 Simbol Use Case Diagram (Sumber : Rukmana, Dkk (2023))

NO	Nama	Simbol	Keterangan
1	Actor	 Actor/Role	Yang menggambarkan manusia atau sistem yang mendapatkan manfaat namun berada di luar subjek (sistem).
2	UseCase	 Use Case	Yang menunjukkan fungsionalitas utama dalam sistem.
3	Asociation		Abstraksi dari koneksi antara aktor

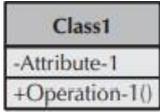
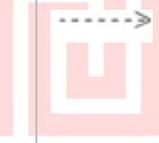
NO	Nama	Simbol	Keterangan
			dan <i>use case</i> .
4	<i>Extend</i>		Menunjukkan bahwa suatu <i>use case</i> adalah tambahan fungsional dari <i>use case</i> lain jika suatu kondisi tertentu terpenuhi.
5	Include		Yang merepresentasikan inklusi fungsional dari satu <i>Use Case</i> menuju <i>Use Case</i> lainnya.
6	Generalisasi		Menunjukkan bagaimana aktor disesuaikan untuk berpartisipasi dalam <i>use case</i> tertentu.

Tujuan utama dari Use Case Diagram adalah untuk menyediakan dokumentasi yang mudah dipahami mengenai fungsionalitas sistem yang dibangun. Diagram ini membantu para pengembang, pemangku kepentingan, dan klien untuk lebih mudah memahami dan mendiskusikan sistem yang sedang dirancang, bahkan bagi mereka yang tidak memiliki latar belakang teknis yang mendalam. Diagram ini juga berfungsi untuk mendokumentasikan kebutuhan fungsional sistem

b) Class Diagram

Class diagram adalah salah satu elemen penting dalam pengembangan perangkat lunak, khususnya dalam fase desain sistem. Diagram ini menggambarkan struktur statis sistem yang menggambarkan kelas-kelas yang ada dalam sebuah aplikasi serta hubungan antar kelas tersebut. Setiap kelas pada class diagram memiliki atribut (data) dan metode (fungsi) yang menjelaskan karakteristik dan perilaku objek yang dapat dibuat dari kelas tersebut.

Tabel 2. 2 Simbol Class Diagram (Sumber : Rukmana, Dkk (2023))

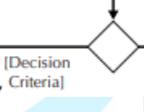
NO	Nama	Simbol	Keterangan
1	Class		Kumpulan objek yang memiliki atribut dan operasi yang serupa.
2	<i>Realization</i>		Tindakan yang sepenuhnya dilakukan oleh sebuah objek.
3	Asociation		Hubungan yang menghubungkan satu objek dengan objek lainnya.
4	Generalization		Hubungan di mana objek turunan berbagi sifat dan struktur data dari objek di atasnya, yaitu objek yang lebih umum.
5	Depedency		Hubungan di mana perubahan pada satu elemen yang mandiri akan berdampak pada elemen yang tergantung padanya.

Activity Diagram ini menampilkan langkah-langkah yang terlibat dalam suatu aktivitas atau proses, serta bagaimana aliran kontrol bergerak antar langkah tersebut. Dalam konteks perangkat lunak, activity diagram sering kali digunakan untuk memodelkan prosedur atau bisnis proses yang menggambarkan urutan tindakan yang harus diambil, keputusan yang perlu dibuat, serta interaksi antar komponen sistem.

Pada activity diagram, setiap aktivitas atau langkah kerja digambarkan dengan simbol berbentuk bulat atau oval, sementara alur proses diwakili oleh panah yang menghubungkan aktivitas satu ke aktivitas lainnya. Selain itu, diagram ini juga dapat menunjukkan kondisi atau keputusan dengan menggunakan simbol berlian, yang

menunjukkan cabang alur berdasarkan kondisi yang dipenuhi. berdasarkan kondisi yang ada.

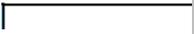
Tabel 2.3 Simbol Activity Diagram (Sumber : Rukmana, Dkk (2023))

NO	Nama	Simbol	Keterangan
1	Initial Node		Menandai titik permulaan dari alur proses. Digunakan untuk menunjukkan di mana aktivitas dimulai. Dalam setiap diagram aktivitas hanya ada satu initial node.
2	Activity		Mewakili suatu tindakan atau pekerjaan yang dilakukan oleh sistem atau aktor. Ini bisa berupa proses manual atau otomatis, dan biasanya menjadi inti dari alur kerja sistem.
3	Control Flow (Alur Kendali)		Menunjukkan urutan dan arah eksekusi antar aktivitas. Panah ini menghubungkan node satu ke node lainnya, menggambarkan bagaimana proses berpindah dari satu aktivitas ke aktivitas berikutnya.
4	Decision Node		Digunakan untuk membuat percabangan alur berdasarkan kondisi tertentu. Setiap panah keluar dari decision node diberi label kondisi, dan hanya satu jalur yang diambil berdasarkan hasil evaluasi kondisi tersebut.
5	Final Node		Menunjukkan akhir dari alur aktivitas. Semua proses berhenti ketika mencapai node ini. Diagram aktivitas hanya memiliki satu final node utama.
6	Swimlane		Yang digunakan untuk membagi diagram aktivitas menjadi baris dan kolom, di mana ini membantu menetapkan individu atau objek yang bertanggungjawab dalam menjalankan aktivitas tertentu.

c) Sequence Diagram

Sequence diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang menggambarkan interaksi antar objek atau komponen dalam sistem seiring berjalannya waktu. Diagram ini menunjukkan urutan pesan atau perintah yang dikirimkan antara objek-objek yang terlibat dalam suatu proses atau skenario tertentu. Sequence diagram sangat berguna untuk memodelkan bagaimana objek berinteraksi dan berkomunikasi satu sama lain dalam rangka mencapai tujuan tertentu dalam sistem.

Tabel 2.4 Simbol Sequence Diagram (Sumber : Rukmana, Dkk (2023))

NO	Nama	Simbol	Keterangan
1	Actor	 Actor	Mewakili entitas yang berinteraksi dalam skenario, bisa berupa pengguna, sistem, atau objek. Ditempatkan secara horizontal sebagai pelaku komunikasi.
2	Lifeline		Menunjukkan eksistensi (keberadaan) partisipan selama alur interaksi berlangsung. Digunakan untuk menyusun kronologi dari pesan-pesan yang dikirim dan diterima.
3	Activation Bar		Menunjukkan kapan suatu objek/aktor sedang aktif menjalankan operasi atau memproses pesan. Panjang bar ini mencerminkan durasi aktivitas.
4	Object		Berfungsi untuk menerima atau mengirimkan sebuah pesan sebagai interaksi antar object atau class.
5	Message		

Di sequence diagram, objek yang saling terkait digambarkan

dalam bentuk kolom vertikal, dengan garis horizontal yang mewakili waktu. Setiap objek akan menerima pesan yang digambarkan dengan panah horizontal, yang menunjukkan interaksi atau komunikasi antara objek tersebut. Pesan ini bisa berupa pemanggilan metode atau pengiriman data, yang mengarah pada perubahan status atau hasil tertentu. Sequence diagram juga dapat menggambarkan kondisi atau keputusan, di mana urutan alur akan bercabang sesuai dengan hasil dari kondisi yang diuji.

2.1.2 Aplikasi

Aplikasi Menurut Nur Azis dkk. (2020) tunjukkan hingga kata "aplikasi" berasal dari kata "use," yang berarti "aplikasi." Aplikasi seringkali merupakan program yang sudah jadi yang dirancang untuk memenuhi tujuan tertentu bagi pengguna atau aplikasi lain. Aplikasi tersebut dapat digunakan sesuai dengan tujuan yang diinginkan. Wulandari (2021).

Aplikasi komputer adalah perangkat lunak yang dirancang untuk membantu pengguna dalam menyelesaikan tugas-tugas tertentu, seperti pengolahan data, pengelolaan informasi, komunikasi, hingga hiburan. Menurut Ciptayasa (2021) dalam *Buku Aplikasi Komputer Dasar*, aplikasi merupakan alat bantu dalam pengolahan data yang digunakan untuk mempermudah pekerjaan manusia di berbagai bidang. Aplikasi tidak hanya sekadar alat bantu, tetapi juga menjadi komponen penting dalam transformasi digital, baik di sektor pendidikan, pemerintahan, bisnis, hingga layanan publik.

a. Klasifikasi Aplikasi Komputer

Aplikasi komputer dapat diklasifikasikan berdasarkan platform tempat aplikasi dijalankan dan fungsi utamanya, sebagai berikut:

1. Berdasarkan Platform:

- a. Aplikasi Desktop: Dijalankan di sistem operasi komputer seperti Windows, Linux, atau macOS. Biasanya digunakan untuk pekerjaan administratif atau teknis. Contohnya: Microsoft Word, Excel, dan Adobe Photoshop.
- b. Aplikasi Web: Diakses melalui browser dan tidak memerlukan

instalasi. Cocok untuk penggunaan yang fleksibel dan kolaboratif secara daring. Contohnya: Google Docs, Gmail, dan Canva.

2. Berdasarkan Fungsi:

- a. Aplikasi Produktivitas: Membantu aktivitas sehari-hari dalam dunia kerja atau pendidikan, seperti pengolah kata dan spreadsheet.
- b. Aplikasi Multimedia: Digunakan untuk membuat, mengedit, atau memutar file media. Contoh: VLC Media Player, Adobe Premiere.
- c. Aplikasi Komunikasi: Memfasilitasi komunikasi antarindividu maupun kelompok. Contoh: Zoom, Skype, Microsoft Teams.
- d. Aplikasi Pendidikan: Digunakan untuk proses pembelajaran interaktif atau mandiri. Contoh: Ruangguru, Khan Academy, Quipper.
- e. Aplikasi Bisnis: Meliputi sistem POS, manajemen inventaris, dan CRM seperti Salesforce atau SAP.

B. Hubungan Aplikasi dengan Komponen Sistem Komputer

Dalam sistem komputer, aplikasi berinteraksi dengan beberapa komponen penting agar dapat berjalan optimal. Pertama, aplikasi membutuhkan perangkat keras (hardware) seperti CPU, RAM, dan penyimpanan untuk mendukung proses eksekusi data dan pemrosesan. Kedua, aplikasi berjalan di atas sistem operasi seperti Windows, Android, atau iOS, yang bertugas sebagai penghubung antara perangkat keras dan perangkat lunak. Ketiga, banyak aplikasi saat ini yang terintegrasi dengan jaringan komputer (LAN atau internet), yang memungkinkan fitur seperti penyimpanan cloud, kolaborasi daring, hingga komunikasi real-time.

2.1.3 Website

Web merupakan halaman-halaman web yang terhubung satu sama lain dan bisa diakses melalui browser. World Wide Web, yang diciptakan oleh Tim Berners-Lee pada tahun 1989, telah menjadi bagian integral dari internet dan membantu orang di seluruh dunia berbagi dan berbagi informasi. Halaman web dibuat dengan menggunakan bahasa seperti HTML, CSS, dan JavaScript, kemudian dihosting di server web, yang kemudian mengirimkan

konten ke peramban pengguna saat diminta.

Website juga merupakan kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet, yang berisi berbagai jenis informasi seperti teks, gambar, video, dan lainnya.

Menurut beberapa ahli, website adalah fasilitas yang menghubungkan dokumen secara online atau kumpulan halaman interaktif yang saling terkait. Hal ini sesuai dengan pendapat dalam buku "Pengantar Teknologi Informasi" oleh Rudi Haryanto (2021), yang menjelaskan bahwa website adalah sekumpulan halaman informasi yang disajikan melalui jalur internet sehingga dapat diakses di seluruh dunia selama terhubung dengan jaringan internet.

Website dapat diklasifikasikan berdasarkan fungsinya, seperti:

- a) Blog: Situs pribadi atau kelompok yang berisi artikel, opini, atau catatan harian.
- b) E-commerce: Situs yang digunakan untuk kegiatan jual beli secara online, seperti Tokopedia dan Shopee.
- c) Pemerintah: Situs resmi instansi pemerintah yang menyediakan informasi dan layanan publik.

Klasifikasi ini didukung oleh buku "*Desain dan Implementasi Website*" oleh Andi Prasetyo (2022), yang menjelaskan berbagai jenis website berdasarkan fungsi dan tujuannya. Seiring perkembangan teknologi, website mengalami evolusi dari versi statis hingga dinamis:

- 1) Single Page Application (SPA): Website yang memuat satu halaman utama dan memperbarui konten secara dinamis tanpa memuat ulang seluruh halaman.
- 2) Progressive Web App (PWA): Website yang memiliki fitur seperti aplikasi mobile, dapat diakses offline, dan memiliki performa tinggi.

Penjelasan ini sesuai dengan buku "*Web Programming: Konsep dan Implementasi*" oleh Budi Raharjo (2023), yang membahas berbagai versi dan evolusi website.

2.1.4 E-ticket Helpdesk

Sistem Pelayanan aduan merupakan sebuah solusi berbasis teknologi yang digunakan untuk mempermudah pengelolaan dan

pemrosesan tiket pengaduan, layanan, atau permasalahan yang diajukan oleh pengguna.

E-ticketing helpdesk adalah sistem yang dirancang untuk mengelola dan mendokumentasikan permintaan bantuan atau keluhan dari pengguna secara digital. Sistem ini memungkinkan tim dukungan untuk melacak, mengatur, dan menyelesaikan masalah pelanggan dengan lebih efisien.

Beberapa fitur utama dalam sistem e-ticketing helpdesk meliputi:

1. Pelacakan Tiket: Memungkinkan tim dukungan untuk melacak status setiap tiket dari awal hingga selesai.
2. Prioritas Masalah: Menentukan tingkat urgensi dari setiap tiket untuk penanganan yang lebih efektif.
3. Integrasi Komunikasi: Menyediakan platform komunikasi antara pelanggan dan agen dukungan langsung dalam tiket, termasuk integrasi email dan fitur chat.
4. Basis Pengetahuan: Menyediakan akses ke sumber daya bantuan bagi pelanggan dan agen dukungan untuk menyelesaikan masalah secara mandiri.
5. Laporan dan Analitik: Menyediakan alat untuk menghasilkan laporan dan menganalisis data terkait volume tiket, waktu penyelesaian, kinerja agen, dan kepuasan pelanggan.

2.1.5 Helpdesk

Menurut Menurut Manulu, W. U. S. , helpdesk adalah sebuah layanan atau bagian dalam organisasi yang bertugas memberikan bantuan teknis dan dukungan kepada pengguna akhir atau pelanggan mengenai penggunaan produk atau layanan yang disediakan oleh organisasi tersebut. Tugas utama helpdesk adalah menjawab pertanyaan, menyelesaikan masalah, atau memenuhi permintaan bantuan teknis yang diajukan oleh pengguna, baik melalui telepon, email, maupun sistem tiket online. Helpdesk biasanya menjadi titik pertama yang dihubungi oleh pengguna yang mengalami kesulitan atau memiliki pertanyaan mengenai produk atau layanan yang digunakan.

Helpdesk adalah layanan yang dirancang untuk membantu pengguna dalam menyelesaikan masalah teknis atau administratif yang mereka hadapi. Menurut Arini dkk (2020), helpdesk merupakan layanan teknologi informasi yang mencakup dukungan dan pemeliharaan, serta berisi laporan permintaan dari pengguna. antara pengguna dan penyedia layanan.

Dalam praktiknya, helpdesk dapat dikategorikan berdasarkan beberapa aspek:

1. Berdasarkan Target Pengguna:

- a. Internal Helpdesk: Melayani karyawan atau anggota organisasi dalam menyelesaikan masalah internal, seperti gangguan sistem atau permintaan akses.
- b. External Helpdesk: Berfokus pada pelanggan atau pengguna eksternal yang membutuhkan bantuan terkait produk atau layanan perusahaan.

2. Berdasarkan Metode Layanan:

- a) Helpdesk Manual: Menggunakan metode tradisional seperti telepon atau email untuk menerima dan menangani permintaan bantuan.
- b) Helpdesk Otomatis (Ticketing System): Menggunakan perangkat lunak khusus untuk mengelola dan melacak permintaan bantuan secara sistematis.

3. Berdasarkan Waktu Operasional:

- a) Helpdesk 24/7: Menyediakan layanan bantuan sepanjang waktu tanpa henti.
- b) Helpdesk Jam Kerja: Beroperasi sesuai dengan jam kerja perusahaan atau organisasi.

Dalam konteks perusahaan, helpdesk memiliki peran vital dalam menjaga kelancaran operasional dan kepuasan pengguna. Beberapa fungsi utama helpdesk di perusahaan antara lain

- 1) Titik Kontak Utama: Helpdesk berfungsi sebagai pintu pertama bagi pengguna yang mengalami masalah atau memiliki

pertanyaan, memastikan bahwa setiap permintaan ditangani dengan cepat dan efisien.

- 2) Pencatatan dan Pelacakan Masalah: Dengan sistem ticketing, helpdesk dapat mencatat setiap permintaan bantuan, memantau status penyelesaiannya, dan memastikan bahwa tidak ada permintaan yang terlewatkan.
- 3) Analisis dan Pelaporan: Data yang dikumpulkan oleh helpdesk dapat dianalisis untuk mengidentifikasi tren masalah, memungkinkan perusahaan untuk melakukan perbaikan sistematis dan proaktif.
- 4) Peningkatan Kepuasan Pengguna: Dengan memberikan respons yang cepat dan solusi yang efektif, helpdesk berkontribusi pada peningkatan kepuasan pengguna, baik internal maupun eksternal.

2.1.6 Database

Menurut Rahimi Fitri (2020), database merujuk pada kumpulan data yang terorganisir, biasanya disimpan dan diakses secara elektronik dalam suatu sistem komputer. Helmud (2021) juga menekankan bahwa database merupakan sekumpulan informasi yang disimpan secara sistematis agar bisa diakses oleh aplikasi untuk mendapatkan informasi yang dibutuhkan.

1. Relational Database (RDBMS): Menyimpan data dalam bentuk tabel dengan baris dan kolom. Cocok untuk data terstruktur. Contoh: MySQL, PostgreSQL, Oracle.
2. NoSQL Database: Dirancang untuk data tidak terstruktur atau semi-terstruktur. Cocok untuk big data dan aplikasi real-time. Contoh: MongoDB, Cassandra, Redis.
3. Object-Oriented Database: Menyimpan data dalam bentuk objek, mirip dengan konsep OOP. Cocok untuk aplikasi yang kompleks. Contoh: db4o, ObjectDB.
4. Hierarchical Database: Data disusun dalam struktur pohon dengan hubungan parent-child. Cocok untuk sistem yang memerlukan struktur data hierarkis.
5. Network Database: Memungkinkan hubungan many-to-many antar data. Cocok untuk aplikasi yang memerlukan

fleksibilitas dalam hubungan data.

Jenis-jenis database ini dijelaskan secara mendalam dalam buku "Modern Database Management" yang membahas perkembangan dan implementasi berbagai model database.

MySQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang menggunakan bahasa SQL (Structured Query Language) untuk mengelola data. MySQL banyak digunakan dalam pengembangan aplikasi web karena kemudahan dan fleksibilitasnya.

Normalisasi Basis Database. Dalam merancang sistem informasi yang efisien, proses normalisasi menjadi salah satu langkah penting untuk memastikan struktur basis data terbentuk dengan baik dan bebas dari redundansi. Menurut Sonedika dan Pedia (2023), normalisasi adalah proses mengorganisir struktur tabel dalam basis data relasional dengan tujuan untuk menghindari duplikasi data, mengurangi inkonsistensi, serta menjaga integritas data.

Normalisasi dilakukan dengan membagi data ke dalam beberapa tabel yang lebih kecil, di mana setiap tabel hanya menyimpan informasi yang berhubungan dengan satu entitas atau hubungan saja. Proses ini dilakukan secara bertahap berdasarkan bentuk normal (normal form) tertentu. Adapun tahapan normalisasi menurut Sonedika & Pedia (2023) meliputi:

1. Bentuk Normal Pertama (1NF)

Tahap ini memastikan bahwa setiap sel dalam tabel hanya berisi satu nilai (nilai atomik) dan tidak ada data yang berulang. Hal ini bertujuan agar struktur tabel menjadi lebih terstruktur dan efisien.

2. Bentuk Normal Kedua (2NF)

Pada tahap ini, setiap atribut non-kunci dalam tabel harus memiliki dependensi fungsional penuh terhadap kunci utama. Jika terdapat atribut yang hanya bergantung pada sebagian dari kunci utama, maka atribut tersebut harus dipindahkan ke tabel terpisah.

3. Bentuk Normal Ketiga (3NF)

Tahap ini memastikan tidak adanya dependensi transitif antar atribut

non-kunci. Jika ditemukan hubungan tidak langsung antar atribut non-kunci, maka perlu dipisahkan lagi ke tabel baru agar struktur basis data menjadi lebih logis.

4. Bentuk Normal Selanjutnya (BCNF, 4NF, 5NF)

Digunakan untuk menangani ketergantungan yang lebih kompleks, seperti ketergantungan multivalued atau join dependencies, terutama pada sistem dengan basis data skala besar.

2.2 Tinjauan Studi

1. Jurnal Penelitian yang dilakukan oleh Anton, R. dan Sari, L. pada tahun 2022 berjudul "*Rancang Bangun Aplikasi Pelayanan Aduan untuk Pelaporan Keluhan Pelanggan di Perusahaan*" membahas tentang pengembangan aplikasi berbasis web yang dirancang untuk mempermudah pelanggan dalam menyampaikan keluhan kepada perusahaan. Tujuan dari aplikasi ini adalah menyediakan media pelaporan yang cepat, transparan, dan mudah diakses. Dalam proses pengembangannya, penulis menggunakan pendekatan desain berfokus pada pengguna, dengan pengumpulan data dilakukan melalui survei dan wawancara guna memahami kebutuhan serta ekspektasi pelanggan. Fitur utama aplikasi meliputi pengiriman keluhan secara langsung, pelacakan status keluhan, serta notifikasi penyelesaian masalah. Metode pengembangan sistem yang digunakan SDLC meliputi tahapan analisis kebutuhan, desain antarmuka, implementasi, dan pengujian. Hasil dari penelitian menunjukkan bahwa aplikasi yang dibangun mampu mempercepat proses penanganan keluhan, meningkatkan respons perusahaan, serta meningkatkan kepuasan pelanggan.

Perbedaan antara penelitian tersebut dengan penelitian yang sedang dilakukan oleh penulis saat ini yang berjudul "*Rancang Bangun Aplikasi Pelayanan Aduan Berbasis Web pada PT. FORCE Quality Solution Menggunakan Metode RAD*" terletak pada beberapa aspek. Penelitian terdahulu berfokus pada aduan dari pelanggan, sedangkan

penelitian ini berfokus pada pengelolaan aduan internal dari karyawan perusahaan. Dari sisi metode pengembangan sistem, penelitian terdahulu menggunakan pendekatan umum berbasis tahapan sistematis, sedangkan penulis menggunakan metode Rapid Application Development (RAD) yang lebih cepat dan iteratif. Dalam pengumpulan data, penelitian terdahulu menggunakan dua teknik yaitu survei dan wawancara, sedangkan penulis hanya menggunakan metode wawancara. Dari sisi fitur, meskipun memiliki kesamaan fungsi utama, aplikasi yang penulis kembangkan dilengkapi dengan fitur tambahan berupa laporan bulanan yang tidak terdapat dalam penelitian sebelumnya.

2. Dalam Penelitian yang dilakukan oleh Fitriansyah dan Hasibuan pada tahun 2022 berjudul "*Penerapan Service Request Management pada Tata Kelola Layanan Teknologi Informasi*" dan dipublikasikan dalam *JRIS: Jurnal Rekayasa Informasi Swadharma, Vol. 2, No. 1* bertujuan untuk meningkatkan efektivitas proses permintaan layanan (service request) pada divisi teknologi informasi di PT. Mitra Integrasi Informatika. Penelitian ini menggunakan metode kualitatif dengan teknik pengumpulan data melalui observasi langsung di departemen IT, wawancara dengan pihak terkait, serta analisis terhadap database internal perusahaan. Sistem yang dirancang dalam penelitian ini mengacu pada kerangka kerja ITIL (Information Technology Infrastructure Library) sebagai pedoman dalam tata kelola layanan TI. Aplikasi yang diusulkan mencakup fitur notifikasi melalui email, pengamanan data, pemantauan oleh top-level management, serta akses melalui website dan perangkat Android. Tujuan akhirnya adalah menggantikan proses manual berbasis kertas menjadi sistem digital yang efisien dan terintegrasi untuk permintaan hardware dan software.

Perbedaan dengan penelitian yang sedang dilakukan oleh penulis saat ini yang berjudul "*Rancang Bangun Aplikasi Pelayanan Aduan*"

Berbasis Web pada PT. FORCE Quality Solution Menggunakan Metode RAD”, terdapat beberapa persamaan dan perbedaan yang cukup signifikan. Dari segi metode, keduanya sama-sama menggunakan pendekatan kualitatif, namun teknik pengumpulan datanya berbeda; penelitian terdahulu menggunakan observasi, wawancara, dan analisis database, sementara penulis hanya menggunakan wawancara. Dari sisi tujuan, keduanya berfokus pada pengelolaan layanan dalam konteks internal perusahaan, namun topik dan fokus utama berbeda penelitian terdahulu menangani permintaan layanan IT, sedangkan penelitian ini berfokus pada pelayanan aduan internal dari karyawan. Selain itu, penelitian terdahulu menggunakan kerangka kerja ITIL sebagai dasar pengembangan sistem, sedangkan penelitian ini menggunakan pendekatan RAD (Rapid Application Development) sebagai metode pengembangan sistem. Dari sisi platform, keduanya sama-sama berbasis web, namun sistem yang dikembangkan oleh Fitriansyah & Hasibuan juga mencakup platform mobile Android, sementara sistem dalam penelitian ini difokuskan untuk penggunaan berbasis web dengan fitur tambahan berupa laporan bulanan yang tidak dijumpai dalam penelitian sebelumnya.

3. Penelitian yang dilakukan oleh Mazia dkk. pada tahun 2021 berjudul *“Rancang Bangun Sistem Informasi Helpdesk Ticketing Berbasis Web pada PT Mitra Tiga Berlian Bekasi”*, dipublikasikan dalam *Jurnal Informatika Universitas Pamulang, Vol. 6, No. 1*. Penelitian ini dilatarbelakangi oleh permasalahan yang muncul saat aktivitas kerja berbasis teknologi di perusahaan sering mengalami kendala yang tidak terdokumentasi dengan baik dan menimbulkan miskomunikasi di divisi IT. Untuk mengatasi hal tersebut, penulis merancang sistem Helpdesk berbasis web yang digunakan untuk mencatat dan melaporkan gangguan secara sistematis. Pengumpulan data dilakukan melalui observasi, wawancara, serta studi pustaka. Metode pengembangan sistem yang digunakan adalah SDLC dengan model waterfall, yang mencakup tahapan analisis kebutuhan, perancangan sistem

menggunakan ERD, implementasi, dan pengujian. Hasil akhir dari penelitian ini adalah sistem Helpdesk Ticketing berbasis web yang mampu memfasilitasi proses pelaporan gangguan teknologi informasi dengan lebih cepat, terdokumentasi, dan terstruktur.

Perbandingan dengan penelitian penulis yang berjudul "*Rancang Bangun Aplikasi Pelayanan Aduan Berbasis Web pada PT. FORCE Quality Solution Menggunakan Metode RAD*", keduanya memiliki persamaan dari sisi jenis sistem yang dikembangkan, yaitu sistem berbasis web untuk menangani aduan atau gangguan yang berkaitan dengan operasional perusahaan, serta sama-sama menggunakan pendekatan kualitatif dalam pengumpulan data. Namun terdapat beberapa perbedaan, terutama pada metode pengembangan sistem; penelitian Mazia menggunakan model waterfall, sedangkan penulis menggunakan Rapid Application Development (RAD) yang lebih iteratif dan fleksibel. Selain itu, penelitian Mazia menggabungkan observasi, wawancara, dan studi pustaka, sedangkan penulis hanya menggunakan metode wawancara. Dari sisi fokus pengguna, sistem yang dikembangkan Mazia lebih ditujukan untuk divisi IT dalam menangani gangguan teknis, sedangkan sistem yang dikembangkan oleh penulis mencakup pelayanan aduan secara umum dari seluruh karyawan, dan dilengkapi dengan fitur tambahan seperti laporan bulanan, yang tidak dijumpai dalam penelitian Mazia.

4. Jurnal Penelitian yang dilakukan oleh Budi, Y. dan Rahma, T. pada tahun 2021 berjudul "*Implementasi Sistem Pelayanan Aduan untuk Mengelola Keluhan Pelanggan di Sektor Keluarga*" membahas pengembangan sistem pelayanan aduan yang dirancang untuk mempermudah pelanggan dalam menyampaikan keluhan mereka secara online. Sistem ini tidak hanya memungkinkan pengiriman keluhan secara cepat dan efisien, tetapi juga menyediakan kemudahan bagi penyedia layanan dalam menanggapi dan menyelesaikan masalah

pelanggan. Fitur yang dikembangkan mencakup pengiriman aduan, pelacakan status keluhan, serta notifikasi yang diberikan kepada pelanggan mengenai perkembangan penanganan aduan. Metodologi yang digunakan dalam penelitian ini meliputi analisis kebutuhan, desain sistem, implementasi, dan evaluasi. Untuk mendukung perancangan sistem, penulis melakukan survei guna memahami kebutuhan serta kendala yang sebelumnya dihadapi dalam pengelolaan keluhan.

Perbandingan dengan penelitian yang sedang dilakukan oleh penulis dengan judul "*Rancang Bangun Aplikasi Pelayanan Aduan Berbasis Web pada PT. FORCE Quality Solution Menggunakan Metode RAD*", terdapat sejumlah kesamaan dan perbedaan. Dari sisi kesamaan, keduanya sama-sama merancang sistem pelayanan aduan berbasis web dengan fitur utama seperti pengiriman keluhan, pelacakan status, dan notifikasi. Perbedaannya terletak pada konteks dan kelengkapan fitur. Penelitian Budi dan Rahma ditujukan untuk sektor layanan masyarakat atau keluarga, sedangkan penelitian ini fokus pada pengelolaan aduan internal karyawan di lingkungan perusahaan. Selain itu, aplikasi yang dikembangkan dalam penelitian ini memiliki fitur tambahan berupa laporan bulanan, yang tidak tersedia pada penelitian sebelumnya. Dari sisi metodologi, Budi dan Rahma menggunakan tahapan pengembangan sistem secara umum, sedangkan penelitian ini secara khusus menggunakan metode Rapid Application Development (RAD) yang menekankan pada kecepatan pengembangan dan iterasi yang fleksibel.

5. Jurnal Penelitian yang dilakukan oleh Dhivi Rizki Ramadhani berjudul "*Rancang Bangun Aplikasi Helpdesk Ticketing Berbasis Web*" dilakukan di Eurokars Group Indonesia dengan tujuan membangun sistem Helpdesk internal yang menggantikan sistem sebelumnya, yaitu Spiceworks dan PowerApps, yang memiliki banyak keterbatasan. Permasalahan yang diangkat dalam penelitian

ini meliputi proses pelacakan tiket yang tidak transparan, komunikasi yang tidak efektif antara pengguna dan tim IT, serta penggunaan formulir hardcopy yang memperlambat proses pengajuan perbaikan dan permintaan perangkat IT. Penelitian ini menggunakan pendekatan kualitatif, dengan metode pengumpulan data berupa observasi dan wawancara, dan metode pengembangan sistem yang digunakan adalah Waterfall. Aplikasi yang dikembangkan berbasis web dan terintegrasi dengan modul e-form untuk pengajuan akun karyawan baru (New Joinee), permintaan perangkat rusak atau baru, pelacakan status tiket, serta notifikasi status. Penulis juga melakukan analisis PIECES dan elisitasi kebutuhan sistem untuk merancang solusi yang tepat bagi permasalahan yang ada di perusahaan.

Perbandingan dengan penelitian yang dilakukan oleh penulis saat ini berjudul "*Rancang Bangun Aplikasi Pelayanan Aduan Berbasis Web pada PT. FORCE Quality Solution Menggunakan Metode RAD*", terdapat beberapa persamaan dan perbedaan. Kesamaan keduanya terletak pada fokus pengembangan aplikasi berbasis web untuk mendukung layanan aduan internal perusahaan, penggunaan pendekatan kualitatif, serta tujuan meningkatkan efisiensi dan kecepatan layanan. Namun, perbedaannya cukup signifikan, terutama pada metode pengembangan sistem: Dhivi menggunakan metode Waterfall, sedangkan penulis menggunakan metode Rapid Application Development (RAD). Dari sisi pengumpulan data, Dhivi menggunakan observasi dan wawancara, sementara penulis hanya menggunakan wawancara. Selain itu, aplikasi yang dikembangkan Dhivi lebih ditujukan untuk mendukung proses permintaan perangkat dan pengelolaan akun oleh divisi IT, sedangkan aplikasi yang dikembangkan penulis lebih umum untuk pelayanan aduan seluruh karyawan di berbagai divisi. Penelitian ini juga melengkapi sistem dengan fitur laporan bulanan, yang tidak dijumpai pada sistem yang dikembangkan oleh Dhivi.